A new method for minimising a sum of squares without calculating gradients

G. Peckham

University of Reading, J. J. Thomson Physical Laboratory, Whiteknights, Reading, Berkshire

A new method for minimising a sum of squares of non-linear functions is described and is shown to be more efficient than other methods in that fewer function values are required.

(Received March 1969)

1. Introduction

The problems of finding the parameters of a physical theory from experimental results, adjusting the parameters in a design to obtain the best possible performance, or solving sets of non-linear simultaneous equations may frequently be reduced to the problem of finding the minimum of a sum of squares of non-linear functions. If gradients are available there are highly efficient and simple methods, but in many cases it is extremely difficult to calculate gradients and sometimes almost impossible. It is easy to make programming errors and the procedure has to be carefully checked, for instance by numerical differentiation of the function. There are considerable advantages to the programmer in methods which do not use gradients, and in many cases these methods are faster.

2. The problem

The vector of variables $x = x_1, x_2, \ldots, x_n$ which gives a minimum value for S is to be determined, where

$$S = \sum_{k=1}^{m} \{f_k(x)\}^2 \tag{1}$$

In a linear approximation we may write

$$f_k = h_k + \sum_{i=1}^{n} g_{ki}x_i$$

or in a matrix notation

$$f = h + Gx \tag{2}$$

The value of x at the minimum is y given by

$$G^TGy = - G^Th \tag{3}$$

If the gradients g_{ki} are available, these equations can be solved for y. Since the f_k are not generally linear functions, y will not be the true minimum, but may be used as a starting value for the next iteration. Powell (1965) gives a method in which the gradients are evaluated numerically, the component in the direction of y being re-evaluated at each iteration. He shows his method to be comparable in efficiency with the least squares method using gradients, and very much more efficient than methods designed to minimise functions which are not in general sums of squares. (See also Box, 1966.) This latter result is not surprising, for if we assume S to have the form given by equations (1) and (2), the unknown coefficients h_k and g_{ki} could be determined by evaluating the sets of functions f_k at $n + 1$ points, whereas, if S is not known to be a sum of squares, the simplest assumption we can make is that it is quadratic in the neighbourhood of the minimum and the number of function values needed to determine the coefficients of a quadratic form is $\frac{1}{2}(n + 1)$ $(n + 2)$, that is 66 for $n = 10$ compared with 11 for the sum of squares.

3. The new method

Spendley *et al.* (1962) describe a method for finding the minimum of a function (not necessarily a sum of squares) in which the function is evaluated at $n + 1$ points forming a simplex in n dimensional space. An iteration consists of replacing the point with highest function value by its reflection in the hyperplane containing the other points. Nelder and Mead (1965) and Box (1965) describe methods based on this in which the 'simplex' becomes irregular and may consist of a set of more than the minimum of $n + 1$ points necessary to span n dimensions. However the *ad hoc* rules of these methods do not lead to as rapid convergence as can be achieved by methods based on the properties of quadratic forms, at least for suitably well behaved functions. (Powell, 1964 and Box, 1966.)

In the case of functions which are sums of squares, the above discussion suggests that the function values at a set of $n + 1$ or more points might be used to estimate values for the coefficients h_k and g_{ki} and hence the position of the minimum y from equation (3). An iteration would consist of replacing the point of the set with highest function value by this estimate of the position of the minimum. Convergence should be rapid, as, if the f_k were strictly linear functions, the minimum would be found in one iteration.

Assume that we have function values f_{kl} for a set of p points x_{il} where $p \geqslant n + 1$ and $l = 1, 2, \ldots, p$. The linear approximation is obtained by choosing h and G to minimise the m expressions

$$\sum_{l=1}^{p} w_l^2\left(h_k + \sum_{i=1}^{n} g_{ki}x_{il} - f_{kl}\right)^2 \tag{4}$$

where $k = 1, 2, \ldots, m$ and w_l is a weighting factor. It is convenient to choose the weighted mean as origin for the x_{il} so that $\sum_{l=1}^{p} w_l^2x_{il} = 0$ and to define $x'_{il} = w_lx_{il}$ and $f'_{kl} = w_lf_{kl}$. The values of h and G which minimise (4) are given by

$$X'X'^TG^T = X'F'^T \qquad h = \frac{1}{\Omega}F'w$$

where an obvious matrix notation has been used and $\Omega = \sum_{l=1}^{p} w_l^2$.

If these values are substituted in equation (3), the following expression is obtained for y

$$y = -\frac{1}{\Omega}(X'X'^T)(X'F'^TF'X'^T)^{-1}(X'F'^TF'w) \qquad (5)$$

4. Some details of the new method

The functions f_k are evaluated at a starting point and at points displaced by given distances along each of the axes in turn. These points should span the region in which the minimum is expected to lie. The number of points p in the set is increased from $n + 1$ to the largest integer less than $n + 3 + \frac{1}{3}n$, the new points being obtained by repeated application of equation (5). Thereafter each new point replaces the point of the set with largest sum of squares. The upper bound for p was chosen empirically and is of order the number of function values needed to determine the minimum in a simple test case (see **Table 1**).

The values of w_l were chosen to give function values near the minimum more weight in determining h and G.

$$w_l^2 = \frac{1}{S_l} \text{ where } S_l = \sum_{k=1}^{m} (f_{kl})^2$$

Although convergence was entirely satisfactory for the test functions described below, it has been found advisable to ensure convergence by limiting the step made in any one iteration and by not accepting a new point if S for this point is larger than all in the set. If a point x_l is unacceptable, a new point x_l', may be generated by

$$x_l' = \frac{w_lx_l + w_0x_0}{w_l + w_0}$$

where x_0 is the point with smallest S. This rule is applied several times if necessary to give an acceptable point. The average value of S for the set is now reduced by each iteration.

To identify a good numerical method for solving equation (5), it is rewritten as:

$$y = -\frac{1}{\Omega}X'X'^Tz \qquad (6)$$

where $$(X'F'^TF'X'^T)z = X'F'^TF'w \qquad (7)$$

Equation (7) will be recognised as the normal equation of a linear least squares problem. The euclidean norm $||F'w - F'X'^Tz||$ has a minimum value when z satisfies (7). This problem is best solved by the use of orthogonal transformations and methods have been described by Golub (1965) and Bauer (1965). The ALGOL procedure 'Ortholin 2' described by Bauer was used, but without iterative refinement of the solution.

The number of points in the set, p, has been chosen to be greater than the minimum number, $n + 1$, necessary to span n dimensions to reduce the probability that the set may collapse into a subspace of less than n dimensions. However, this can still happen, an example being the case where the functions f_k are linear in one of the variables so that each iteration gives the same value for this variable (i.e. the value at the minimum). In this case equation (5), will be ill conditioned (this can be detected during application of the orthogonal transformations) and no attempt is made to obtain a solution. The coordinates of a new point in the neighbourhood of the point with smallest sum of squares, S, are generated by a pseudo-random number procedure. This new point replaces the point with largest sum of squares and the new set will in general span the full n dimensions. If not the procedure will be repeated at the next iteration. It is important for the user of the algorithm to ensure that there are at least $n + 1$ independent functions f_k.

The work required per iteration, namely of order mn^2 operations is much greater than that required by other methods (e.g. Powell, 1965). It is claimed, however, that fewer function evaluations are needed and that in many cases, the time required to find the minimum will be less.

5. Numerical examples and comparison with other methods

Powell (1965) and Box (1966) compare the performance of various methods in solving the simultaneous equations

$$\sum_{i=1}^{n} (a_{ki} \sin x_i + b_{ki} \cos x_i) = e_k \qquad k = 1, \ldots, m \qquad (9)$$

by defining

$$f_k = \sum_{i=1}^{n} (a_{ki} \sin x_i + b_{ki} \cos x_i) - e_k$$

and minimising

$$S = \sum_{k=1}^{m} (f_k)^2$$

a_{ki} and b_{ki} are random numbers in the interval $[-100, 100]$ and the solutions x_i are random in the interval $[-\pi, \pi]$. Starting values differ from known solutions by random numbers in the range $\left[-\frac{\pi}{10}, \frac{\pi}{10}\right]$. The comparison shows Powell's (1965) method to be the most effective of the methods not requiring gradients. In **Table 1**, m has been taken equal to n and the number of function values needed to obtain values of x_i within 0·0001 of the true solution is compared with the numbers needed by the least squares method (with gradients) and Powell's method as given in Table 1 of his paper. (The two values for each n were obtained with different random number sequences.)

Table 1

Number of function values to solve equations (9)

n	LST. SQS.	POWELL	NEW METHOD
5	5	24	11
5	10	24	11
10	5	38	15
10	8	34	20*
20	6	46	31
20	9	65	32

* Converged to another solution close to that originally chosen.

To obtain figures comparable with Powell's (1965) Table 2, m was increased to $2n$ and a disturbance in the range $[-\delta, \delta]$ applied to the values of e_k so that S was no longer zero at the minimum.

Table 2

Number of function values to minimise a sum of squares that does not tend to zero

n	$\delta = 0{\cdot}1$		$\delta = 1$		$\delta = 10$	
	POWELL	NEW	POWELL	NEW	POWELL	NEW
5	17	8	37	18	33	24
5	20		29		34	
10	26	15	47	27	78	34
10	29		47		86	
20	42	26	118	48	175	55
20	36		88		93	

The differences in behaviour between the new method and Powell's are clearly shown in the two dimensional case first introduced by Rosenbrock (1960):

$$S = f_1^2 + f_2^2$$

where $$f_1 = 10(x_2 - x_1^2) \qquad f_2 = 1 - x_1$$

The function S has the form of a parabolic valley descending to a minimum of zero at the point (1, 1). The starting value is at $(-1{\cdot}2, 1)$. Since f_2 is a linear function, the first iteration of the new method gave a point with $f_2 = 0$. Further iterations gave points along the line $x_1 = 1$ (except for an occasional small deviation to regain the 2 dimensions), the minimum being found after only 12 evaluations of f_1 and f_2. Final convergence was rapid, the last three values for S being 2×10^{-4}, 6×10^{-5}, 4×10^{-18} (zero within rounding error).

In Powell's method, after solving equation (3) for the estimated position of the minimum, y, a search is made for the true minimum of S along the direction y. This means that each iteration gives a point on the valley floor so that progress is along the parabolic curve of the valley, and far more function evaluations are needed (70).

After writing this paper, the author's attention was drawn to a paper by Spendley (1969) in which he describes an improvement to the simplex search method (Spendley *et al.*, 1962; Nelder and Mead, 1965) which is applicable to functions which are sums of squares. As in the present paper, an estimate of the minimum is made by means of a quadratic approximation determined by a number of function values. However, there are important differences:

(*a*) The quadratic approximation is used only rarely (e.g., every $3n$ iterations in the example quoted in Table 2).

(*b*) The quadratic approximation is determined by the minimum number of function values $(n + 1)$ instead of by the larger number (p) used in the present work.

Although the improved method is shown to need fewer iterations than the original simplex search method, many more iterations are needed than for Powell's (1965) method or for the present method.

6. Conclusions

A new method for finding the minimum of a sum of squares of nonlinear functions has been described and has been shown to be significantly more efficient than other methods in that it requires fewer function evaluations. It is easy to program as gradients of the functions are not required. The economy in the number of function values required will in most cases lead to considerable time saving, for, although the number of operations needed per iteration is greater than in other methods, most of the time will be spent in computing function values.

An ALGOL procedure incorporating the new method has been used to fit observed crystal lattice vibrational frequencies by various theoretical models with up to twelve adjustable parameters and in the design of optical filters. Convergence has been satisfactory in all cases so far tried. Copies of the ALGOL procedure written for an Elliott 4130 computer may be obtained from the author.

References

Bauer, F. L. (1965). Elimination with weighted row combinations for solving linear equations and least squares problems, *Numerische Mathematik*, Vol. 7, pp. 338–352.

Box, M. J. (1965). A new method of constrained optimisation and a comparison with other methods, *The Computer Journal*, Vol. 8, pp. 42–52.

Box, M. J. (1966). A comparison of several current optimisation methods, and the use of transformations in constrained problems, *The Computer Journal*, Vol. 9, pp. 67–77.

Golub, G. (1965). Numerical Methods for solving linear least squares problems, *Numerische Mathematik*, Vol. 7, pp. 206–216.

Nelder, J. A., and Mead, R. (1965). A simplex method for function minimisation, *The Computer Journal*, Vol. 7, pp. 308–313.

Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives, *The Computer Journal*, Vol. 7, pp. 155–162.

Powell, M. J. D. (1965). A method for minimising a sum of squares of non-linear functions without calculating derivatives, *The Computer Journal*, Vol. 7, pp. 303–307.

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

Spendley, W. (1969). Nonlinear Least squares fitting using a modified simplex minimization method, *Minimization* Ed. R Fletcher, Academic Press: London and New York, pp. 259–270.

Spendley, W., Hext, G. R., and Himsworth, F. R. (1962). Sequential application of simplex designs in optimisation and evolutionary operation, *Technometrics*, Vol. 4, pp. 441–461.