Generalized inverse methods for the best least squares solution of systems of non-linear equations

By R. Fletcher*

It is shown how many previous methods for the exact solution (or best least squares solution) of systems of non-linear equations are all based upon simple cases of the generalized inverse of the matrix of first derivatives of the equations. The general case is given and algorithms for its application are suggested, especially in the case where the matrix of first derivatives cannot be calculated. Numerical tests confirm that these algorithms extend the range of practical problems which can be solved.

1. Introduction

The solution of a number of equations in as many variables occurs frequently in scientific problems. The derivation of best approximations by minimizing the sum of squares of differences between two functions (residuals) occurs even more widely. Both can be posed as the problem of obtaining the best least squares approximation to a set of m simultaneous equations in n variables.

Methods of solution are well known for linear equations. However, in the case where the equations are ill-conditioned or even singular in some way, methods described in this paper may be used to advantage. Primarily, though, we shall be concerned with the far more general, and more difficult, problem of non-linear equations.

There is a whole group of methods available which are based on approximating to the non-linear situation by a linear one and solving the problem iteratively. They all involve in some way an inverse connected with the matrix of first derivatives of the equations. When there are as many equations as unknowns then Newton's method (see Broyden, 1965) is best known and may be used if derivatives of the equations can be evaluated. If this is not so, or is inconvenient, then methods such as the Secant method (Wolfe, 1959) or those of Barnes (1965) and Broyden (1965) can be used. When there are more equations than unknowns, a best least squares solution may be obtained using the Generalized Least Squares method (sometimes called the Gauss or Gauss–Newton method: see Powell (1965)); unless derivatives cannot be evaluated in which case the method given by Powell (*loc. cit.*) can be used.

One aim of this paper is to describe a method involving the "generalized inverse" of the matrix of first derivatives of the equations, and also to show how the methods mentioned in the previous paragraph are all special cases of this new method. The generalized inverse is a concept which has been developed rapidly in recent times and which is often linked with the solution of linear least squares problems—see for example Greville (1959). A brief introduction to the generalized inverse is given at the beginning of Section 3, followed by formulation of the method for the straightforward case when derivatives can be evaluated. An important property of the method which implies stability is also presented. In Section 4 it is shown that when derivatives cannot be evaluated, then the generalized inverse can be approximated using differences and used in a very similar way. In both cases the connection with previous methods is explained. Numerical results on a representative range of problems are then presented and discussed. Finally an appendix is given on how a simple algorithm for computation of the generalized inverse might readily be deduced from the ideas of Section 4.

A further objective has been to examine the convergence of inverse type methods. Previous methods often fail when the matrix of first derivatives becomes of non-maximum rank and the required inverses cannot be calculated. Problems do occur which exhibit this behaviour in various ways and unfortunately it is not generally possible to tell beforehand that this is likely to happen. The effect of loss of rank on these methods is discussed in the last section and reasons are given as to why convergence breaks down. It is shown how the generalized inverse formulation caters automatically for these problems. There is only one proviso, namely that in methods such as these, where stability is obtained by ensuring that the sum of squares is decreased at each iteration, convergence may take place to a local rather than a global minimum of the sum of squares.

2. Notation

Conventional matrix notation will be used with A, b and c representing matrix, column vector and scalar respectively. The transpose of A will be denoted by A'. A set of m non-linear simultaneous algebraic equations in n variables $x \equiv (x_1, x_2 \dots x_n)'$, can be written $f_1(x) = 0$, $f_2(x) = 0$, . . . ,$f_m(x) = 0$, or collectively $f(x) = 0$. The vector of function values (residuals) for any particular x will be denoted by $f(x)$ or more often f. The $m \times n$ matrix of first partial derivatives J (Jacobian) has elements $J_{ij} = \partial f_i / \partial x_j$ and continuity of J and f will be assumed. A special case of these are linear equations

$$f(x) = Jx - b = 0 \quad (1)$$

in which J is a constant matrix of coefficients.

* *Electronic Computing Laboratory, University of Leeds, Leeds,* 2.

Given certain conditions, an exact unique solution of a set of non-linear equations will only occur if there are n equations in n unknowns. If $m > n$ then there are usually too many equations to determine an exact solution (overdetermined equations) and if $m < n$ then too few equations, so that the solution is not unique (underdetermined). In all cases, however, we can talk sensibly of a solution as that x which minimizes a scalar function F (a norm) of the residuals. In particular the least squares norm $F(x) = f'f = (f_1^2 + f_2^2 + \ldots + f_m^2)$ will be used. In what follows "solution" will refer to a best least squares solution, and "exact solution" to one with zero sum of squares. Finally the gradient of F with respect to x will be denoted by $g(x)$, and can be calculated using the relation $g = 2J'f$.

In the case of linear equations we can use the property that the gradient will be zero at the solution, and together with (1) we get

$$J'Jx = J'b \qquad (2)$$

as a necessary and sufficient condition that x is a solution.

Methods for solution of non-linear equations are generally iterative, and subscripts will be used to identify members of an iterative sequence, e.g. $K_1, K_2, \ldots, K_i$. If $x_1, x_2, \ldots, x_i$ are a sequence of n-dimensional column vectors they will be denoted collectively as the $n \times i$ matrix $[x_i]$. A projection matrix P (see for example Householder, 1953), can be defined for a set of vectors $[x_i]$, such that Py is the projection of any vector y on the sub-space $[x_i]$, and $y - Py$ the component of y orthogonal to $[x_i]$. P is idempotent and symmetric and if the vectors $[x_i]$ are linearly independent we have the relationship $P = [x_i]([x_i]'[x_i])^{-1}[x_i]'$. A complementary projection matrix $\tilde{P} = I - P$ can also be defined so that $\tilde{P}y$ becomes the component of y orthogonal to $[x_i]$. If the x_i are linearly independent, we can consider $\tilde{P}$ as being derived from a set of vectors $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n-i}$ or $[\tilde{x}_{n-i}]$ which span the sub-space orthogonal to $[x_i]$.

Finally, differences between vectors are often required, and the notation $\Delta x_i = x_{i+1} - x_i$ is used. If the equations are linear then the relation

$$\Delta f_i = J \Delta x_i \qquad (3)$$

is of importance.

3. The generalized inverse method

The generalized inverse is an extension of the concept of an inverse for matrices which are singular or rectangular. If A is a real $m \times n$ matrix, then the generalized inverse of A is a real $n \times m$ matrix, denoted by A^+ which satisfies the equations

$$AA^+A = A \qquad A^+AA^+ = A^+ \qquad (4a, b)$$

$$(A^+A)' = A^+A \qquad (AA^+)' = AA^+ \qquad (4c, d)$$

In the more general problem of complex matrices, Penrose (1955) showed that the solution of these equations for A^+ is unique. If A is square non-singular, then $A^+ = A^{-1}$ (the ordinary inverse) and if $m > n$ and $\operatorname{rank}(A) = n$ then $A^+ = (A'A)^{-1}A'$. Both these can be verified by substitution in (4), as also can $0^+ = 0$ for null matrices, and $A^+ = A'/\operatorname{trace}(A'A)$ for matrices of unit rank. More complicated formulae, however, can be derived for all cases. A simple method suitable for computation is described in the Appendix. Finally A^+A is the projection matrix for rows of A, and AA^+ for columns of A.

If we have any set of linear equations with matrix J as in (1) above, then the associated generalized inverse J^+ enables us to find the best least squares solution directly. If we are given any x, and calculate

$$\Delta x = -J^+f(x) \qquad (5)$$

then $x + \Delta x$ is a solution of (1). This we can readily show, as from (5) and (1),

$$x + \Delta x = x - J^+Jx + J^+b$$

so

$$\begin{aligned} J'J(x + \Delta x) &= J'Jx - J'JJ^+Jx + J'JJ^+b \\ &= J'JJ^+b \text{ from } (4a) \\ &= J'(JJ^+)'b \text{ from } (4d) \\ &= (JJ^+J)'b. \end{aligned}$$

Thus using (4a) again we get

$$J'J(x + \Delta x) = J'b$$

showing that $x + \Delta x$ satisfies the conditions (2) for a best least squares solution. If $m \geqslant n$ and $\operatorname{rank}(J) = n$ then we obtain the well known

$$x = (J'J)^{-1}J'b$$

as a unique solution. If $\operatorname{rank}(J) < n$, then by virtue of (5), the solution (not unique) obtained is that whose component outside the column space of J^+ (row space of J) is the same as that of the initial approximation.

This form of a solution of a set of linear simultaneous equations (i.e. (5)) is readily adapted to an iterative method for non-linear equations, as it requires only that the residuals and the matrix of first partial derivatives be evaluated. For reasons of stability we can consider (5) as not defining a difference, but a direction s thus

$$s = -J^+f.$$

We can then ensure that the sum of squares of residuals $F(x)$ is reduced at each iteration by minimizing it along the direction s through the current approximation x.

Thus we might have:

(i) given x, set $i = 1$;

(ii) compute f_i, J_i and $s_i = -J_i^+f_i$;

(iii) set $x_{i+1} = x_i + \alpha_i s_i$ choosing α_i so that $\alpha_i > 0$ and $F(x_{i+1})$ is the minimum of $F(x)$ in the direction s_i through x_i;

(iv) set $i = i + 1$ and repeat from (ii) until convergence;

as the basis of a suitable algorithm.

It will now be shown that this is in fact an extension of two other well known methods. In the case where $m \geqslant n$ and $\text{rank}(J) = n$ then J^+ becomes $(J'J)^{-1}J'$ and the algorithm reduces to the generalized least squares method in which s_i is obtained from $-(J_i'J_i)^{-1}J_i'f_i$ or equivalently by solving $(J_i'J_i)s_i = -J_i'f_i$. If the further restriction is made that $m = n$ (i.e. as many equations as unknowns) then J^+ becomes J^{-1} and the algorithm becomes Newton's method, where s_i is obtained from $-J_i^{-1}f_i$ or again by solving $J_is_i = -f_i$.

The convergence of the generalized least squares and Newton methods is limited to regions of x-space for which the condition that $\text{rank}(J) = n$ is satisfied, for otherwise the required inverses could not be calculated. Use of the generalized inverse, however, extends this method to all situations. This is not to say that we expect to solve many linear problems in which $\text{rank}(J) < n$, that is when the solution is not unique. However, there exist non-linear problems where J evaluated at the solution is not of maximum rank. Of equal importance, J evaluated at any approximation x to the solution may not be of maximum rank, even though it may be at the solution, especially if that approximation is poor. Thus the case of $\text{rank}(J) < n$ cannot be neglected from a practical viewpoint. (The problem of underdetermined non-linear equations is also covered by the theory, although these problems do only occur rarely.)

A most important property of the generalized inverse formulation is that in all circumstances, even when the generalized least squares method would fail, the directions of search generated are downhill, and so an improvement can always be made to the sum of squares (assuming that the approximation is not already a stationary point). For as the gradient $g = 2J'f$ and the direction of search $s = -J^+f$, so $-g's = f'JJ^+f$. But JJ^+ is a projection matrix for columns of J and is therefore positive semi-definite, showing $-g's$ to be non-negative. As the approximation is not stationary, so $g \neq 0$ and hence $J'f \neq 0$. Thus f is not orthogonal to all columns of J and the projection matrix cannot annihilate f. Hence $-g's$ is strictly positive, showing that s has a positive component along the negative gradient and so is downhill.

Apart from computation of the generalized inverse, described in the Appendix, the only other practical points in setting up an algorithm concern the linear search for a minimum, and testing for convergence. When derivatives are available a process of cubic interpolation can be used, described for example by Fletcher and Powell (1963). Two different approaches to this can be seen in ALGOL procedures by Wells (1965) and Fletcher (1966). In the case to be studied below where derivatives are not available, a method based on quadratic interpolation can be used. Further references can be found in Fletcher (1965). This is more efficient than the Fibonnaci search (see for example Spang (1962)) when high accuracy is not required. Convergence can be tested for either by finding no improvement in the sum of squares in an iteration (n iterations when the inverse is being formed by differencing) or by finding the elements of $s = -J^+f$ to be less in absolute magnitude than some preassigned vector ϵ which measures the tolerance allowed in x at the solution. These have been tried and found suitable although other strategies also suggest themselves.

4. Difference formulation

Frequently it is required to solve systems of non-linear equations when the matrix of first derivatives J and hence J^+ cannot be computed directly. An expression for J^+ in terms of differences between function values could then be used. One approach would be to take differences about the current approximation, calculate J^+, and proceed as in the previous section. This would be an inefficient way to use function evaluations and it is worthwhile to consider other possibilities.

Previous work has proceeded by taking an approximate matrix and updating it at each iteration in accordance with information about the function obtained during that iteration. Powell's method, which is related to the generalized least squares method, essentially updates approximations to $(J'J)^{-1}$ and J', and finds directions of search from $s = (J'J)^{-1}J'f$. Related to Newton's method are the Secant method and Barnes' and Broyden's methods, differing chiefly in how the approximating matrix is handled. In the Secant method, an augmented form of J^{-1} is calculated from points $x_1 \ldots x_{n+1}$ and corresponding residuals $f_1 \ldots f_{n+1}$, and is subsequently updated at each iteration; in Barnes' method an approximation to J is updated using differences in x and f; and in Broyden's method an approximation to J^{-1} is updated using differences. In all three methods, however, the approximating matrix is used in the appropriate way at each iteration to calculate directions of search from $s = -J^{-1}f$.

This section shows how an approximating matrix can be used with the generalized inverse formulation described in the previous section. An approximation to J^+ rather than J will be considered, as this limits the amount of computation at each iteration to order mn rather than mn^2. Formulae for J^+ in terms of differences are given and it is also shown how an arbitrary matrix of the correct dimensions can be updated, so that after n iterations of an iterative process it has become the generalized inverse of the differences $\Delta f_1 \ldots \Delta f_n$ obtained from the steps $\Delta x_1 \ldots \Delta x_n$ taken at each iteration. Finally details of how this approximating matrix can be used to calculate directions of search from $s = -J^+f$ in a general iterative scheme are described.

Consider therefore the matrices $[\Delta f_n]$ and $[\Delta x_n]$ (of order $m \times n$ and $n \times n$) whose columns are the differences above, and assume that they are related by (3) with a matrix of first derivatives J. Of necessity we can only consider linear equations (i.e. (1)) in this analysis but the results are in suitable form for iterative use in the non-linear case. The aim is to state the formula for J which these differences imply, and hence that for J^+. Assume that the Δx_i are linearly independent so that $[\Delta x_n]$ is non-singular. Then by virtue of (3) we have

$$J = [\Delta f_n]\,[\Delta x_n]^{-1}.$$

If the rank of J and hence $[\Delta f_n]$ is n, and $m \geqslant n$, then

$$J^+ = [\Delta x_n]\,([\Delta f_n]'\,[\Delta f_n])^{-1}\,[\Delta f_n]'.$$

which can be verified by substitution in (4a–d).

This has properties

$$J^+\Delta f_j = \Delta x_j \qquad j \leqslant n \qquad (6a)$$

$$J^+\widetilde{\Delta f}_j = 0 \qquad j \leqslant m - n. \qquad (6b)$$

(6a) is complementary to (3), and (6b) shows how J^+ multiplies with vectors $\tilde{\Delta} f_i$ from any basis orthogonal to $[\Delta f_n]$. (See notation.)

This, however, is only a special case of the general problem in which the rank of J and hence $[\Delta f_n]$ is r, where $r \leqslant n$. The importance of this case when we extend to non-linear equations was discussed in the previous section and applies equally here. In this case we can pick r vectors $[\Delta x_r]$ for which the corresponding $[\Delta f_r]$ are linearly independent (if necessary by permuting the columns of the original $[\Delta x_n]$, $[\Delta f_n]$). Then

$$[\Delta f_r] = J[\Delta x_r]. \qquad (7)$$

From the remaining $n - r$ columns of $[\Delta x_n]$ we can, by removing the appropriate linear combinations of the columns of $[\Delta x_r]$, obtain a set of linearly independent column vectors $[\eta_{n-r}]$, for which the corresponding changes in f would be zero (remember this analysis assumes linearity), and these can further be considered mutually orthonormal without loss of generality. Thus

$$[0_{n-r}] = J[\eta_{n-r}] \qquad (8)$$

and as the partitioned matrix $([\Delta x_r] \mid [\eta_{n-r}])$ is still non-singular, we have

$$J = ([\Delta f_r] \mid [0_{n-r}])\,([\Delta x_r] \mid [\eta_{n-r}])^{-1}.$$

If N is a projection matrix (see notation) for this "null" sub-space spanned by columns of $[\eta_{n-r}]$ and $\tilde{N} = I - N$ is a complementary projection matrix which removes the components in this sub-space, then we can write J without partitions as

$$J = [\Delta f_r]([\Delta x_r]'\tilde{N}[\Delta x_r])^{-1}[\Delta x_r]'\tilde{N}$$

which can be verified by substitution in (7) and (8). The generalized inverse is then

$$J^+ = \tilde{N}[\Delta x_r]([\Delta f_r]'[\Delta f_r])^{-1}[\Delta f_r]'$$

which can be verified by substitution in (4a–d), with

$$JJ^+ = [\Delta f_r]([\Delta f_r]'[\Delta f_r])^{-1}[\Delta f_r]'$$

the projection matrix for columns of $[\Delta f_r]$, and

$$J^+J = \tilde{N}[\Delta x_r]([\Delta x_r]'\tilde{N}[\Delta x_r])^{-1}[\Delta x_r]'\tilde{N}$$

the projection matrix for columns of $\tilde{N}[\Delta x_r]$ (from which we can further deduce that $J^+J = \tilde{N}$).

Finally J^+ has the following properties (which are sufficient to define it uniquely)

$$J^+\Delta f_j = \tilde{N}\Delta x_j \qquad j \leqslant n \qquad (9a)$$

$$(J^+)'\eta_j = 0 \qquad j \leqslant n - r \qquad (9b)$$

$$J^+\tilde{\Delta} f_j = 0 \qquad j \leqslant m - r \qquad (9c)$$

with regard to individual vectors of differences. All these formulae reduce to those for the simpler case by setting $r = n$, when there is no "null" sub-space and $\tilde{N} = I$.

It will now be shown how an arbitrary $n \times m$ matrix say K_1 can be taken and updated to give K_2, K_3 etc., using differences obtained during an iteration, so that after the nth iteration K_{n+1} will have all the properties (9a–c) with regard to the differences obtained. Denote by K_i the matrix at the beginning of the ith iteration, and by r_j the rank of $[\Delta f_j]$. Let the "null" subspace at the beginning of the ith iteration be spanned by orthogonal vectors η_j $(1 \leqslant j < i - r_{i-1})$ with projection matrix N_i and let $\tilde{N}_i = I - N_i$. Finally assume that K_i satisfies properties (9a, b) with respect to the first $i - 1$ differences, that is

$$K_i\Delta f_j = \tilde{N}_i\Delta x_j \qquad j < i \qquad (10a)$$

$$K_i'\eta_j = 0 \qquad j < i - r_{i-1}. \qquad (10b)$$

Then it will be shown that K_{i+1} satisfies these properties with $i + 1$ replacing i. We shall need vectors z_i and y_i obtained by orthogonalization of Δx_i with respect to $[\Delta x_{i-1}]$ and Δf_i to $[\Delta f_{i-1}]$. Either of two possibilities may then occur.

If Δf_i is linearly independent of columns $[\Delta f_{i-1}]$ (i.e. $y_i \neq 0$) then $N_{i+1} = N_i$ (no change in the "null" sub-space), $r_i = r_{i-1} + 1$, and we can update K_i by

$$K_{i+1} = K_i + \frac{\tilde{N}_i\Delta x_i v'}{(v'\Delta f_i)} - \frac{K_i\Delta f_i w'}{(w'\Delta f_i)}$$

where v and w are non-trivial m-vectors orthogonal to $[\Delta f_{i-1}]$ but not to Δf_i. It can readily be verified that K_{i+1} satisfies the properties (10a, b). The simplest choice for v and w which satisfies the required conditions is $v = w = y_i$. Then the simpler formula

$$K_{i+1} = K_i + \frac{(\tilde{N}_i\Delta x_i - K_i\Delta f_i)y_i'}{(y_i'\Delta f_i)} \qquad (12)$$

is used to update K_i.

If Δf_i is dependent on columns of $[\Delta f_{i-1}]$, (i.e. $y_i = 0$) then there is a new vector $\eta_{i-r_{i-1}} = \tilde{N}_i\Delta x_i - K_i\Delta f_i$ which extends the null sub-space and is orthogonal to previous vectors in the null sub-space. K_i is then updated from

$$K_{i+1} = K_i - \frac{\eta\eta' K_i}{(\eta'\eta)} \qquad (13)$$

where η refers to $\eta_{i-r_{i-1}}$. The rank of $[\Delta f_i]$ is unchanged so $r_i = r_{i-1}$ so it can be verified that K_{i+1} satisfies (10a, b) in this case.

By an inductive argument, as ($10a$, b) imply no conditions on K_1, so K_{n+1} satisfies ($10a$, b) and hence ($9a$, b). As yet ($9c$) is not satisfied by K_{n+1}. However, after the nth iteration we can readily calculate a sequence of orthonormal vectors $\widetilde{\Delta f}_1 \ldots \widetilde{\Delta f}_{m-r}$ which complete the space of rank r spanned by columns of $[\Delta f_n]$ and are orthogonal to it (e.g. by using suitably projected unit vectors). Then K_{n+1} can be corrected by

$$K_{n+1} = K_{n+1} - \sum_{i=1}^{m-r} K_{n+1}\widetilde{\Delta f}_i \widetilde{\Delta f}_i' \qquad (14)$$

and as this process does not have any effect on ($9a$, b), so K_{n+1} now satisfies all of ($9a$, b, c) and is the generalized inverse corresponding to the differences obtained.

We are now in a position to state an algorithm for the solution of sets of non-linear equations for which the derivatives are not available. It is based on the repetition of the above updating process in cycles each of n iterations. An approximation x to the solution and an approximation K to the generalized inverse of the matrix of first derivatives are assumed available at the start of each cycle. These are initially arbitrary but approximations at the end of one cycle will naturally be used to start the next. Following Section 3, at each iteration in the cycle, K will be used to generate a direction of search by $s_i = -K_i f_i$. As $[\Delta x_n]$ must be non-singular, however, the linear independence of s_i must be tested. This concept of linear independence is fundamental and must be preserved, even in the linear case: for example consider solving $f(x) \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} x$ from an initial approximation $x_1 = (0, 1)$, with K_1 as the unit matrix. [It is also wise to act when s_i becomes nearly dependent on $[\Delta x_{i-1}]$ in order to keep the computation well-conditioned. An empirical rule has been used, namely that if the length of s_i is reduced to less than 10% when the projection on to $[\Delta x_{i-1}]$ is removed, a component is added to s_i to restore this degree of independence.]

The iteration then involves estimating x_{i+1} as the point at which the sum of squares of residuals F is minimized along s_i, then calculating Δf_i and Δx_i and finally updating K_i by either (12) or (13). Operations with projection matrices, represented conveniently for the theoretical discussion by $\tilde{N}_i \Delta x_i$ for instance, are most readily accomplished by removing components of the η_j by which $\tilde{N}_i$ is defined, using the Gram–Schmidt orthogonalization. After the nth iteration, the cycle is completed by applying (14) to give K all the properties ($9a$–c) of the generalized inverse. If the equations are linear as in (1) then K becomes J^+ after the first cycle and convergence occurs on the next iteration. In the general non-linear case, the cycle is repeated until convergence occurs.

This is the simplest, but by no means the only, way in which differences can be used in a generalized inverse type method. The ultimate method to aim at is one based not on a cyclic method, but on always having K to be the generalized inverse of the differences from the last n iterations. This could be done by calculating K afresh at each iteration, of course, but the advantage of the method described above is that only updating of a matrix is needed, rather than the complete calculation which requires of order n times as much calculation. It seems likely that any "ideal" updating formula, even if possible, would be far more complicated than the simple formulae (12), (13) and (14) given here.

5. Numerical examples

Although algorithms for both derivative and non-derivative problems have been described, numerical tests have been restricted for the present to the more difficult problem of the non-derivative case. This is the more important in the sense that any problem can be solved in this way. It is expected that as results become available, the assessment of generalized inverse algorithms for the derivative case will parallel that for the non-derivative case. With this in mind, an algorithm of the type described in Section 4 was tested on various problems from the literature.

The well known "parabolic valley" equations (Powell, 1965)

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

with solution $x = (1, 1)$ were considered, from the usual initial approximation $x_1 = (-1{\cdot}2, 1{\cdot}0)$. To obtain the solution as accurately as possible, required from 30 to 90 function evaluations, depending upon how K_1 was chosen, and on how the theory was implemented. A second problem chosen was the best least squares estimation of parameters a and b in the function ae^{by} to data values d taken at five different values of y. Here there are five equations

$$f_i = ae^{by_i} - d_i \qquad 1 \leqslant i \leqslant 5$$

in the two unknowns a and b. This problem was readily solved from a good initial guess.

The minimization test functions "Chebyquad" introduced by Fletcher (1965) were used as test functions for non-linear equations in the form

$$\Delta_i(x) = 0 \qquad 1 \leqslant i \leqslant n$$

from the initial estimate $x_1 = (1, 2, \ldots n)/(n + 1)$, (loc. cit., p. 36). These equations have the property that an exact solution exists only for $n = 1(1)7$ and 9. For other values of n the best least squares solution is not exact, and hence has a singular Jacobian. This is the type of problem which is covered by the theory given in this paper, and particular interest is centred in the case $n = 8$. The initial choice of K was the zero matrix, the program being so arranged that this caused co-ordinate directions to be used in the first cycle of the iteration. The result of solving these equations to 4 and 6 decimal places accuracy in x (corresponding to about 8 and 12 decimal places in the sum of squares of residuals F) is shown in the **Table 1.**

Table 1

Number of function evaluations required when solving Chebyquad test problems

n	ACCURACY OF SOLUTION	
	4 DECIMAL PLACES	6 DECIMAL PLACES
2	15	19
4	40	40
6	73	92
8	340	838
9	174	181

Finally the equations given by Freudenstein and Roth (1963)

$$f_1 = x_1 - x_2^3 + 5x_2^2 - 2x_2 - 13$$

$$f_2 = x_1 + x_2^3 + x_2^2 - 14x_2 - 29$$

from $x_1 = (15, -2)$, were considered. These again exhibit the behaviour that a local best least squares solution with singular Jacobian is reached by descending from the given approximation. With K chosen initially as the zero matrix, 121 function evaluations were required to obtain six significant figures accuracy in x. The location of this solution is at $x = [53 - 4\sqrt{(22)}, 2 - \sqrt{22}]/3$, or about $(11.4, -0{\cdot}9)$.

Consideration of the table shows that the Chebyquad $n = 8$ case is clearly anomalous in that very many more function evaluations are required for solution than would be expected from the other cases. Similarly the 121 function evaluations required to solve the Freudenstein and Roth equations from what is quite a good initial approximation, compare badly with those required to solve the parabolic valley equations, which have a far worse initial approximation. It would therefore seem that we can divide problems into well-behaved and badly-behaved classes, according to whether or not the solution has a Jacobian matrix of maximum rank (or more generally, has continuity in rank at the solution).

Only problems which fall into the well-behaved class have been solved by previous methods of inverse type. Results both there (see source papers and also a comparison by Box (1966)) and also in this paper, show that methods based on an inverse type formulation are considerably superior to those based solely on a minimization approach. Badly behaved problems (in the above sense) have not previously been reported as having been solved and tests with a program based on Broyden's method did indeed fail on the Chebyquad $n = 8$ problem. I understand also that a version of Powell's method also failed on this problem. Results given here show that a program based on the generalized inverse of a Jacobian does converge in this sort of problem, but that the rate of convergence is much less impressive. It seems likely that this is due to the discontinuity which exists in the generalized inverse in these cases.

An interesting practical point which occurs is the numerical recognition of the linear dependence of one vector upon others. The approach used when programming has been that of assuming linear independence in Δf_i unless its orthogonal component y_i is exactly zero. It could be argued, however, that because of both rounding and truncation errors, one should not look for zero, but rather for quantities less than some error bound. This can make a considerable difference in the elements of K, as the elements of a generalized inverse are discontinuous with regard to variation from near singular to exactly singular. Two further numerical tests were therefore made using Freudenstein and Roth's equations, looking for zero in y_i to 6 decimal places and then to 3 decimal places. Although the elements of K varied from test to test, little variation in the overall rate of convergence was noticed. Conversely a corresponding reduction was found in the accuracy to which the solution could be located, so study of these possibilities was discontinued.

6. Summary and discussion

The problem of solving systems of m non-linear algebraic equations in n variables has two aspects, depending upon whether or not J, the $m \times n$ matrix of first derivatives of the equations, can be calculated readily from an algebraic formula. The aim of this paper has been to show that methods for finding a best least squares solution of either problem can be based on the concept of the generalized inverse of a matrix and to suggest possible algorithms whereby these ideas might be implemented. In particular J^+ the generalized inverse of J is used to generate directions of search s from the vector of residuals of the equations f by $s = -J^+f$. The current approximation x can then be improved by searching along s for a minimum of the sum of squares of residuals.

Previous work on both aspects of the problem has used one of two types of method, both of which involve special cases of J^+. When there are n equations in n unknowns and J is non-singular, then the generalized inverse J^+ becomes the ordinary inverse J^{-1}, and the above scheme becomes Newton's method when derivatives can be evaluated. Based on this in the non-derivative case is the Secant method, in which an augmented form of the inverse is updated at each iteration; also Barnes' method, in which an approximation to J is updated using differences, and Broyden's method, in which an approximation to J^{-1} is updated using differences. For the more general case of m equations in n unknowns (with $m \geqslant n$) then the generalized inverse becomes $(J'J)^{-1}J'$ if J is of rank n. The generalized least squares method uses this version in the case when derivatives are available, and Powell's method is based on this in the non-derivative case, essentially by updating approximations to both $(J'J)^{-1}$ and J' at each iteration using differences.

All previous methods then, are based on the condition that J shall be of maximum rank. In practice, however, this is by no means always so and cases have been described in Section 5 where J is singular at a solution. Furthermore, even if the solution is well-behaved, J evaluated at a poor approximation may well not be, and there is a distinct inadequacy in previous methods to cater for these cases. The derivative methods break down directly because the inverses which they require cannot be calculated. The analysis of Section 3 shows how use of the true generalized inverse in these cases circumvents this problem, and furthermore, causes downhill directions of search to be chosen, thus implying that the method is stable. In non-derivative methods loss of rank in an approximation to J can be fatal for this same reason, as also can loss of rank of an approximation K to an inverse. For then directions generated by $s = -Kf$ (or its equivalent) span only a sub-space of x-space, in which the solution will not generally lie.

However, the adverse effects of loss of rank can be alleviated to some extent in these non-derivative methods by forcing the approximating matrix to have maximum rank even if the true matrix does not. Powell ensures that of the differences Δx_i and Δf_i which are obtained in the iterative process, those which are retained are linearly independent in both cases. Hence the special case of the generalized inverse which he uses can always be calculated. Broyden tackles the problem empirically, showing that when updating an approximate matrix K by a formula of type (11), with $v = w = z_i$, then choice of $z_i = K_i'\Delta x_i$ leads to a reliable method whereas $z_i = \Delta f_i$ does not. This reliability can be accounted for theoretically, however, by showing that it implies that K can never become singular. For singularity, K_{i+1} in (11) must have the property $K_{i+1}\,q = 0$ for some non-trivial vector q which, from (11), must therefore satisfy

$$q + (K_i^{-1}\Delta x_i - \Delta f_i)(z_i'q)/(z_i'\Delta f_i) = 0.$$

For such a q to exist implies that both $q = \Delta f_i - K_i^{-1}\Delta x_i$ and $z_i'q = z_i'\Delta f_i$. Hence z_i must satisfy $z_i'K_i^{-1}\Delta x_i = 0$. It can be seen that Broyden's choice of $z_i = K_i'\Delta x_i$ can never cause this condition to be satisfied and hence K can never become singular. Unfortunately this choice of z_i does not satisfy the conditions on v and w in (11) which cause convergence for linear equations from arbitrary K in $n + 1$ iterations.

This device of forcing K to have maximum rank covers, to a large extent, cases where J is badly-behaved remote from the solution, but still fails when J at the solution has not maximum rank. The reason is seen by considering that at the solution $J'f = 0$ by virtue of the minimum sum of squares. However K being of maximum rank causes $s = -Kf$ *not* to be zero at the solution. In fact a "spurious" component is introduced into the direction of search s which causes convergence to break down. The generalized inverse always has the property that $s = -J^+f = 0$ at the solution. Previous methods do not allow K to converge to this generalized inverse. Of course, allowing K to have non-maximum rank means that an alternative means of generating directions must be available to ensure that differences Δx do not approach linear dependence. So long as $s = -Kf$ is used whenever possible, no serious effect on rates of convergence is likely.

A further inadequacy of previous non-derivative methods is that they use differencing techniques, followed by inversion to obtain the initial approximation to K. Unfortunately this wastes function evaluations which could be used in making progress to the solution and, more seriously, might well lead to failure on inversion. Most methods could be improved to include the feature that initially K be chosen arbitrarily, and subsequently updated in the appropriate way to take differences obtained into account. This means that crude approximations to derivatives can be used if available and furthermore, that if a sequence of related problems is to be solved, then K from the first solution can be used to start the next, and so on. For this device to be successful, however, the method must have the property that linear equations are solved in $n + 1$ iterations, else a realistic inverse may never be obtained from a poor initial K. The generalized inverse formulation in Section 4 has this property.

The techniques used by various authors to effect the linear search and also to test for convergence seem to be satisfactory. In no method is the obtaining of an exact minimum of the sum of squares of residuals F in the direction of search necessary to the successful updating of K. Thus alternatives such as discussed by Broyden, in which a minimum is not sought, are equally valid and their merit will depend upon empirical evaluation. It seems essential, however, that one must insist that F is not increased in any iteration. Although this may lead to no progress for some directions of search, information obtained about changes in f along that direction can and should be used to update K.

Speed of convergence of the generalized inverse method is neither significantly better nor worse than for other inverse methods, for well-behaved problems where these other methods work satisfactorily. However, the tests on Freudenstein and Roth's equations and Chebyquad $n = 8$, in which consideration of rank is critical, show that convergence can be obtained with a generalized inverse method. The only regret is that the rate of convergence is less rapid, probably on account of the discontinuity in the elements of the generalized inverse.

Appendix

Computation of the generalized inverse

The algorithm of Section 4 can be adapted to calculate the generalized inverse of an arbitrary $m \times n$ matrix A. We can assume $m \geqslant n$: if this is not so we can transpose A and use $(A')^+ = (A^+)'$. The method then involves the updating of an arbitrary matrix K_1 as though the equations $f = Ax = 0$ were being solved. The matrix

K_{n+1} after one cycle of n iterations is then the generalized inverse A^+. For simplicity each difference Δx_i is taken as the ith unit vector with jth element δ_{ij}. Then Δf_i becomes the ith column of A. Also no storage need be assigned to the η sub-space because Δx_i is automatically orthogonal to $[\Delta x_{i-1}]$, so $\tilde{N}_i \Delta x_i = \Delta x_i$. Further simplicity accrues by taking K_1 as the zero matrix. The computation to force $K\widetilde{\Delta f} = 0$ is then not required because this condition holds automatically. Furthermore after the ith iteration, K_{i+1} will have rows $i + 1$ to n with all zero elements, which can be taken into account when making the matrix multiplications. An interesting feature is that K_{i+1} is the generalized inverse for the first i columns of A and the updating formula is therefore a recurrence relation for the generalized inverses of matrices related in this way. Finally if rank$(A) = n$, then the computation requires approximately $1 \cdot 5mn(n + 1)$ multiplications. If the rank is less than n then this figure is an overestimate of the computation required.

References

Barnes, J. G. P. (1965). An algorithm for solving non-linear equations based on the secant method, *Computer Journal*, Vol. 8, p. 66.

Box, M. J. (1966). A comparison of several current optimization methods and the use of transformations in constrained problems, *Computer Journal*, Vol. 9, p. 67.

Broyden, C. G. (1965). A class of methods for solving non- linear simultaneous equations, *Mathematics of Computation*, Vol. 19, p. 577.

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

Fletcher, R. (1965). Function minimization without evaluating derivatives—a review, *Computer Journal*, Vol. 8, p. 33.

Fletcher, R. (1966). Certification of Algorithm 251, Function Minimization, *Communications A.C.M.* Vol. 9, p. 686.

Freudenstein, F., and Roth, B. (1963). Numerical solutions of systems of non-linear equations, *Journal A.C.M.*, Vol. 10, p. 550.

Greville, T. N. E. (1959). The pseudoinverse of a rectangular or singular matrix and its application to the solution of sytems of linear equations, *S.I.A.M. Review*, Vol. 1, p. 38.

Householder, A. S. (1963). *Principles of Numerical Analysis*, McGraw-Hill, New York.

Penrose, R. (1955). A generalized inverse for matrices, *Proc. Camb. Phil. Soc.*, Vol. 51, p. 406.

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

Spang, H. A. (1962). A review of minimization techniques for non-linear functions, *S.I.A.M., Review*, Vol. 4, p. 343.

Wells, M. (1965). Algorithm 251, Function minimization, *Communications A.C.M.*, Vol. 8, p. 169.

Wolfe, P. (1959). The secant method for simultaneous non-linear equations, *Communications A.C.M.*, Vol. 2, p. 12.

Book Review

Approximation of Functions: Theory and Numerical Methods, by G. Meinardus (Translated by L. L. Schumaker), 1967; 198 pages. (Berlin, Heidelberg, New York: *Springer*, DM.54.0, US $13.50.)

This is a slightly expanded translation of a book that was first published in German three years ago. The author's aim is "to collect essential results of approximation theory which on the one hand makes possible a fast introduction to the modern development of this area, and on the other hand provides a certain completeness to the problem area of Tchebycheff approximation". In this Professor Meinardus has succeeded brilliantly and Dr. Schumaker's excellent translation now makes this text available to the English-speaking world.

The whole concern of the book is the study of "best" approximations to functions by simpler functions (usually polynomials or rational functions). In recent years this topic has received considerable stimulus from the widespread use of computers: it has also undergone a revolution through the introduction of the methods of functional analysis. It is this elegant application of the most powerful tools of pure mathematics to problems of considerable practical concern that makes modern numerical analysis so attractive.

Part I treats Linear Approximation. The first five chapters deal with the theoretical problem, covering uniqueness, the minimax properties, error estimates, and a host of related problems. The last two chapter of this section go into more detail about approximation by polynomials and the numerical determination of best approximations. Points of interest for the computer user are the derivation of very good starting approximations for the exchange methods of Remez and Stiefel, and the derivation that the Remez algorithm has second order convergence. Non-linear approximation is the subject of Part II. This is mostly concerned with approximation by rational functions although there is also a brief section on exponential approximation.

All in all, this is a book that can be recommended without reservation to those interested in approximation theory.

P. A. Samet (London)