SIAM J. NUMER. ANAL.
Vol. 15, No. 5, October 1978

0036-1429/78/1505-0008$01.00/0

ALGORITHMS FOR THE SOLUTION OF THE NONLINEAR LEAST-SQUARES PROBLEM*

PHILIP E. GILL† AND WALTER MURRAY†

Abstract. This paper describes a modification to the Gauss–Newton method for the solution of nonlinear least-squares problems. The new method seeks to avoid the deficiencies in the Gauss–Newton method by improving, when necessary, the Hessian approximation by specifically including or approximating some of the neglected terms. The method seeks to compute the search direction without the need to form explicitly either the Hessian approximation or a factorization of this matrix. The benefits of this are similar to that of avoiding the formation of the normal equations in the Gauss–Newton method. Three algorithms based on this method are described: one which assumes that second derivative information is available and two which only assume first derivatives can be computed.

1. Introduction. This paper is concerned with the construction of effective algorithms for finding a point $\overset{*}{x}$ which minimizes the sum of squares of nonlinear functions

$$F(x)=\sum_{i=1}^{m}[f_i(x)]^2, \qquad x\in E^n, \quad m\geqq n.$$

The gradient vector $g(x)$ and Hessian matrix $G(x)$ of $F(x)$ are given by $2J(x)^Tf(x)$ and $2(J(x)^TJ(x)+B(x))$ respectively, where $J(x)$ is the $m\times n$ Jacobian matrix of $f(x)$ whose ith row is $\nabla f_i(x)=(\partial f_i/\partial x_1, \partial f_i/\partial x_2,\cdots,\partial f_i/\partial x_n)$, $B(x)=\sum_{i=1}^{m} f_i(x)G_i(x)$ and $G_i(x)$ is the Hessian matrix of $f_i(x)$. ($F(x)$ is assumed to be twice-continuously differentiable although the methods discussed in this paper will often work when this condition does not hold.) The restriction that m is greater than or equal to n serves only to simplify the notation of later sections.

For general unconstrained minimization problems where the Hessian matrix of second derivatives can be calculated, Newton's method can be used. The method constructs a sequence of vectors $\{x^{(k)}\}$ such that

$$x^{(k+1)}=x^{(k)}+\alpha^{(k)}p_N^{(k)},$$

where $\alpha^{(k)}$ is a scalar steplength and $p_N^{(k)}$, the direction of search, satisfies the equation

$$G(x^{(k)})p_N^{(k)}=-g(x^{(k)}).$$

When minimizing a sum of squares of nonlinear functions, the special form of the Hessian matrix and gradient vector can be used in the Newton equation to give the equivalent form

$$(J(x^{(k)})^TJ(x^{(k)})+B(x^{(k)}))p_N^{(k)}=-J(x^{(k)})^Tf(x^{(k)}). \tag{1}$$

The Gauss–Newton method was the first designed to exploit the special structure of the Hessian matrix and gradient vector which occurs with least-square problems. The method computes the direction of search as the solution of

$$J(x^{(k)})^TJ(x^{(k)})p_{GN}^{(k)}=-J(x^{(k)})^Tf(x^{(k)}). \tag{2}$$

These equations are obtained by neglecting the second-derivative matrix $B(x^{(k)})$ in (1). The Gauss-Newton method is intended for problems where $\|B(x)\|$ is small

* Received by the editors January 21, 1977, and in revised form November 21, 1977.

† Division of Numerical Analysis and Computing, National Physical Laboratory, Department of Industry, Teddington, Middlesex TW11 0LW, England.

compared to $\|J(x)^TJ(x)\|$, such as the so-called "small-residual problem" where $f(x)\to 0$ as $x\to \overset{*}{x}$. For these problems the Gauss–Newton method will ultimately converge at the same rate as Newton's method despite the fact that only first derivatives are required.

Equations (2) are the so-called "normal equations" for the linear least-square problem

$$\underset{p}{\text{minimize}}\ \|J(x^{(k)})p+f(x^{(k)})\|_2.$$

When $J(x^{(k)})$ is rank deficient and consequently p is not unique, the Gauss–Newton method can be generalized by selecting that unique value of p which minimizes the linear least-square problem and has least Euclidean length (Fletcher (1968)). This so-called "minimal least-square solution" can be found using a number of modern numerically-stable techniques (see Lawson and Hanson (1974) for a survey).

During the solution of many nonlinear least-square problems at the National Physical Laboratory, it became evident that if a carefully applied Gauss–Newton method converged, it converged with surprising efficiency. Unfortunately, in common with most other nonlinear least-square methods it can be shown that it may not converge to $\overset{*}{x}$ when J is singular (see Gill and Murray (1976)). Furthermore when $J(x^{(k)})$ is near singular or $\|f(x^{(k)})\|$ is large (the so-called "large-residual case"), we may not be justified in using the Gauss–Newton method since $J(x^{(k)})^TJ(x^{(k)})$ is not an adequate approximation to $G(x^{(k)})$. (Care must be taken in defining what we mean by a "large-residual" problem since a large value of $F(x)$ may be due to the *scaling* of $F(x)$. We define a large-residual problem to be one for which $F(x)/\|J(x)^TJ(x)\|$ is large.)

In this paper we shall describe a general technique which can be viewed as a modification of the Gauss–Newton method that allows convergence for large-residual and rank-deficient problems. The proposed method is derived from an iterative technique for solving Newton's equations using the singular-value decomposition. The method gives the Gauss–Newton direction as a by-product of the computation and does not require positive definiteness of the Hessian matrix.

2. Methods for solving equations with coefficient matrix J^TJ+B: The positive definite case. In this section we shall consider a numerically-stable method for computing the Newton direction in the positive-definite case which takes advantage of the special structure arising in least-square problems. The proposed method has two important properties; (i) it can be readily extended to the indefinite case, (ii) the Gauss–Newton direction is obtained as a by-product of the computation. In order to simplify the notation we shall omit the superfix k.

We wish to determine the vector p_N which satisfies the equations

$$(J^TJ+B)p_N=-J^Tf. \tag{3}$$

Consider the singular-value decomposition of J

$$J=U\begin{bmatrix}S\\0\end{bmatrix}V^T,$$

where $S=\operatorname{diag}(s_1, s_2, \cdots, s_n)$ is the matrix of singular values with $s_{i+1}\leqq s_i$, U is an $m\times m$ orthonormal matrix and V an $n\times n$ orthonormal matrix. Substituting for J in

(3) gives

$$(VS^2V^T+B)p_N = -V[S \quad 0]U^Tf.$$

Pre-multiplying by V^T gives

$$(S^2V^T+V^TB)p_N = -[S \quad 0]U^Tf. \tag{4}$$

The vector p_N can be written as a linear combination of the n-linearly independent columns of V, with

$$p_N = Vz. \tag{5}$$

Substituting for p_N in (4) and performing some elementary rearrangement gives z as the solution of

$$(S^2+V^TBV)z = -[S \quad 0]U^Tf, \tag{6}$$

with p_N being recovered from (5). The vector z can be found by determining the LDL^T factorization of S^2+V^TBV, but this method proves unsatisfactory for repeated application unless S^2+V^TBV is well-conditioned. It is often the case with least-square problems, especially those derived from data-fitting applications, that each of the $f_i(x)$ (and hence the matrix $B(x)=\sum_{i=1}^m f_i(x)G_i(x)$) is small. Furthermore, these problems are such that J is often by nature ill-conditioned and this ill-conditioning must be reflected in the matrix S^2+V^TBV when $\|B\|$ is small compared to $\|J^TJ\|$.

These numerical problems can be avoided by computing p_N as a sum of two components, the first in the subspace spanned by the columns of V corresponding to the *larger* singular values and the second in the subspace spanned by columns of V associated with the *smaller* singular values.

Consider a partition of the diagonal matrix S such that $S_1 = \text{diag}\,(s_1, s_2, \cdots, s_r)$ and $S_2 = \text{diag}\,(s_{r+1}, s_{r+2}, \cdots, s_n)$. We shall define r as the *grade* of the matrix J. The criterion for fixing the grade of J will depend upon the progress of the optimization algorithm and will be discussed later in § 4. The value of r is generally equal to the number of dominant singular values of J and is rarely significantly less than n. For the moment we require only that S_1 be nonsingular. Unless $\|J\|=0$ a choice of $r \geqq 1$ is always possible since $s_1 > 0$. We shall define the *zero grade* of J as $S_1 = 0$, $S_2 = \text{diag}\,(s_1, s_2, \cdots, s_n)$.

The partition of S into S_1 and S_2 implies a similar partition of V such that

$$V = [V_1 \quad V_2],$$

where V_1 is an $n \times r$ matrix and V_2 an $n \times (n-r)$ matrix, we can write

$$p_N = V_1 w_N + V_2 y_N,$$

where w_N is an r vector and y_N an $(n-r)$ vector. Substituting for p_N in (4) and using the fact that

$$V^TV_1 = \begin{bmatrix} I_r \\ 0 \end{bmatrix} \quad \text{and} \quad V^TV_2 = \begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix}$$

we have

$$S^2\begin{bmatrix} I_r \\ 0 \end{bmatrix} w_N + V^TBV_1w_N + S^2\begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix} y_N + V^TBV_2y_N = -[S \quad 0]U^Tf.$$

From this equation we can obtain the two systems

$$S_1^2 w_N + V_1^T B V_1 w_N + V_1^T B V_2 y_N = -S_1 f_1 \tag{7a}$$

and

$$S_2^2 y_N + V_2^T B V_2 y_N + V_2^T B V_1 w_N = -S_2 f_2, \tag{7b}$$

where f_1, f_2 and $\bar{f}$ are r-, $(n-r)$- and $(m-n)$-vectors respectively such that $f^T U = [f_1^T \; f_2^T \; \bar{f}^T]$. The significance of these equations is best illustrated by substituting $B = \varepsilon \bar{B}$, where $\|\bar{B}\| = 1$ and ε is a scalar, and writing $p_N = V_1 w_N + V_2 y_N$ in (7a). Thus

$$S_1^2 w_N + \varepsilon V_1^T \bar{B} p_N = -S_1 f_1$$

and

$$S_2^2 y_N + \varepsilon V_2^T \bar{B} V_2 y_N + \varepsilon V_2^T \bar{B} V_1 w_N = -S_2 f_2.$$

An approximation $\bar{w}_N$ to w_N can be obtained by neglecting terms of order ε in the first set of equations. This gives $\bar{w}_N = -S_1^{-1} f_1$. If $\bar{w}_N$ is substituted for w_N in the second set of equations we obtain $\bar{y}_N$ where

$$(S_2^2 + \varepsilon V_2^T \bar{B} V_2)\bar{y}_N = -S_2 f_2 - \varepsilon V_2^T \bar{B} V_1 \bar{w}_N.$$

This idea can be extended to give the following iterative scheme for p_N:

ITERATIVE ALGORITHM FOR p_N.

(i) Define $\bar{p}_N^{(0)} = 0$;

(ii) for $j = 0, 1, 2, \cdots$, compute $\bar{w}_N^{(j+1)}$ and $\bar{y}_N^{(j+1)}$ such that

$$S_1^2 \bar{w}_N^{(j+1)} = -S_1 f_1 - V_1^T B \bar{p}_N^{(j)}, \tag{8a}$$

$$(S_2^2 + V_2^T B V_2)\bar{y}_N^{(j+1)} = -S_2 f_2 - V_2^T B V_1 \bar{w}_N^{(j+1)} \tag{8b}$$

and set $\bar{p}_1^{(j+1)} = V_1 \bar{w}_N^{(j+1)}$, $\bar{p}_2^{(j+1)} = V_2 \bar{y}_N^{(j+1)}$ and $\bar{p}_N^{(j+1)} = \bar{p}_1^{(j+1)} + \bar{p}_2^{(j+1)}$.

Since, by assumption, the matrix $J^T J + B$ is positive definite, so is the matrix $S_2^2 + V_2^T B V_2$; furthermore, $S_2^2 + V_2^T B V_2$ is not ill-conditioned because S_2 does not contain the large singular values of S. Accordingly the LDL^T factorization is an efficient and numerically-stable method for solving the equations (8b) for $\bar{y}_N^{(j+1)}$.

The following theorem gives some indication of when this scheme will converge.

THEOREM. *If $\bar{p}_N^{(j)}$ and $\bar{p}_N^{(j+1)}$ are two approximations to p_N obtained using the iterative scheme* (8), *then their relative errors satisfy the inequality*

$$\|p_N - \bar{p}_N^{(j+1)}\|/\|p_N\| \leqq \varepsilon\sigma(1+\varepsilon\eta)\|p_N - \bar{p}_N^{(j)}\|/\|p_N\|,$$

where $\varepsilon = \|B\|$, $\sigma = \|S_1^{-2}\|$ and $\eta = \|(S_2^2 + V_2^T B V_2)^{-1}\|$.

Proof. Define $p_1 = V_1 w_N$ and $p_2 = V_2 y_N$ so that we can write $p_N = p_1 + p_2$. Subtracting equations (7a) and (8a) gives

$$\begin{aligned} \|p_1 - \bar{p}_1^{(j+1)}\|/\|p_N\| &= \|V_1 S_1^{-2} V_1^T B(p_N - \bar{p}_N^{(j)})\|/\|p_N\| \\ &\leqq \|V_1\| \, \|S_1^{-2}\| \, \|V_1^T\| \, \|B\| \, \|p_N - \bar{p}_N^{(j)}\|/\|p_N\| \\ &= \varepsilon\sigma\|p_N - \bar{p}_N^{(j)}\|/\|p_N\|, \end{aligned} \tag{9}$$

since $\|V_1\| = \|V_1^T\| = 1$ and $\|B\| = \varepsilon$.

Similarly, subtracting equations (7b) and (8b) gives

$$\begin{aligned} \|p_2 - \bar{p}_2^{(j+1)}\|/\|p_N\| &= \|V_2(S_2^2 + V_2^T B V_2)^{-1} V_2^T B(p_1 - \bar{p}_1^{(j+1)})\|/\|p_N\| \\ &\leqq \varepsilon\eta\|p_1 - \bar{p}_1^{(j+1)}\|/\|p_N\| \\ &\leqq \varepsilon^2\sigma\eta\|p_N - \bar{p}_N^{(j)}\|/\|p_N\|, \end{aligned} \tag{10}$$

from (9). Since $\bar{p}_N^{(j+1)} = \bar{p}_1^{(j+1)} + \bar{p}_2^{(j+1)}$, inequalities (9) and (10) can be used to give

$$\begin{aligned}\|p_N - p_N^{(j+1)}\|/\|p_N\| &\leqq (\varepsilon\sigma + \varepsilon^2\eta\sigma)\|p_N - \bar{p}_N^{(j)}\|/\|p_N\|\\ &= \varepsilon\sigma(1+\varepsilon\eta)\|p_N - \bar{p}_N^{(j)}\|/\|p_N\|,\end{aligned}$$

which proves the theorem. □

Clearly the iterative scheme converges if $\varepsilon\sigma(1+\varepsilon\eta) < 1$. If any of the singular values of J are less than $\varepsilon^{1/2}$ then $\eta \sim O(1/\varepsilon)$ and we essentially require the partition of S to be such that $\varepsilon\sigma < 1$. If we allow $S_1 = 0$ and $S_2 = S$ in our set of allowable partitions there must exist a partition for which the iterative scheme (8) converges since if $V_2 = V$ the computation of p_N trivially reduces to the solution of the equations (6). For a given partition, if it is evident from the evaluation of the sequence $\{\bar{p}_N^{(j)}\}$ that $\bar{p}_N^{(j)}$ is not converging to p_N sufficiently quickly, more elements of S_1 can be included in S_2 and the iterative scheme restarted. This automatically ensures that $\|S_1^{-2}\|$ is reduced (since the singular values of J are arranged in order of decreasing magnitude), but makes equation (8b) less well-conditioned.

3. The indefinite case. When $J^TJ + B$ is indefinite the vector p_N is no longer satisfactory as a direction of search since it may not be a descent direction and may become arbitrarily close to a vector orthogonal to the steepest-descent direction. In order to overcome these difficulties a number of modified Newton methods have been proposed. We shall adapt the modified Newton method given by Gill and Murray (1974b) which computes a modified direction p_{MN} such that

$$(G+E)p_{MN} = -g,$$

where E is a diagonal matrix which is zero if G is positive definite. The matrix E is obtained during the computation of a modified LDL^T factorization of G such that $LDL^T = G + E$. A feature of this method is that we learn whether or not G is positive definite while evaluating the direction of search. This is useful since nothing is known about G a priori.

The iterative scheme (8) can be simply modified by applying the modified LDL^T factorization to the matrix $S_2^2 + V_2^TBV_2$. This gives a direction of search which satisfies the equations

$$(J^TJ + B + V_2EV_2^T)p = -J^Tf.$$

This may not be satisfactory for all values of r since $J^TJ + B$ may have n negative eigenvalues, in which case $J^TJ + B + V_2EV_2^T$ will not be positive definite since the addition of $V_2EV_2^T$ shifts only $n - r$ of the eigenvalues of $J^TJ + B$. As a safeguard, the vector p is recomputed with $r = 0$ if

$$(11) \qquad -g^Tp/(\|g\|\,\|p\|) < \rho$$

where ρ is a pre-assigned fixed positive scalar. (It has been shown by Gill and Murray (1974b) that if $r = 0$ is used in every iteration then there exists some scalar σ such that (11) is true for σ instead of ρ.)

We do not apply the modified LDL^T factorization directly to $J^TJ + B$ at *every* iteration for two reasons. Firstly, the only second-derivative information concerning $F(x)$ which is available at any given stage may be the matrix $V_2^TBV_2$ rather than B itself (this is the situation in the second algorithm suggested in § 4). Secondly and more importantly, we would be ignoring the "least-square" structure of the matrix G when this structure may be useful in gaining information about the problem. For example, if J^TJ is singular and $\|B\|$ is small relative to $\|J^TJ\|$ the matrix $J^TJ + B$ is close to being a positive semi-definite matrix and the round-off error made in merely forming the matrix J^TJ+B can obscure whether it is positive definite or indefinite. This is illustrated when

$$J=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} \quad\text{and}\quad B=\begin{bmatrix}\varepsilon_1 & \\ & \varepsilon_2\end{bmatrix},$$

with ε_1 and ε_2 such that $|\varepsilon_1|<2^{-t}$ and $|\varepsilon_2|<2^{-t}$ but obtained with high relative precision. Forming J^TJ+B directly gives the matrix

$$\begin{bmatrix}2 & 2\\ 2 & 2\end{bmatrix}.$$

The modified LDL^T algorithm would interpret this matrix as being singular and modify it to be positive definite. However, the suggested scheme with $\|S_2\|=0$ gives

$$V_2^TBV_2=\tfrac{1}{2}[1 \quad -1]\begin{bmatrix}\varepsilon_1 & \\ & \varepsilon_2\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}=\tfrac{1}{2}[\varepsilon_1+\varepsilon_2].$$

The determination of whether this matrix is positive definite or indefinite presents no problems. If it *is* indefinite, the correction to the diagonal will be of an order of magnitude comparable to $\|V_2^TBV_2\|$ rather than $\|J^TJ\|$.

3.1. The computation of directions of negative curvature. The sequence of points may converge to a stationary point of $F(x)$ which is not a local minimum. At such a point J^Tf is zero but J^TJ+B is not positive definite and we must compute a direction p, known as a *direction of negative curvature*, such that $p^T(J^TJ+B)p<0$. When J^Tf is zero and $S_2^2+V_2^TBV_2$ is indefinite, it can be shown that the modified LDL^T factorization can provide a direction of negative curvature. We have $LDL^T=S_2^2+V_2^TBV_2+E$, and if any element of E is positive, then J^TJ+B is indefinite and we can compute p as $p=Vz$, where z satisfies the equation

$$L^Tz=e_j. \tag{12}$$

We shall omit a rigorous proof that p is a direction of negative curvature. However, Theorem 2.3.1 given by Gill and Murray (1974b, pp. 320–321) implies that z is a direction of negative curvature in the space spanned by the columns of V and this property is unaffected by changes in the coordinate system of the form $p=Vz$.

If $S_2^2+V_2^TBV_2$ is positive definite at a stationary point, the matrix J^TJ+B could still be indefinite. The modified LDL^T factorization must immediately be computed with the grade of J fixed at zero. The matrix J^TJ+B is indefinite if and only if an element of E is nonzero. In this event a direction of negative curvature can be computed using (12).

There is no need to compute the modified LDL^T factorization at a stationary point if $F(x)$ is small (compared to $\|J^TJ\|$; see § 1) since x is almost certain to be a local minimum.

4. Algorithms. In this section we shall utilize the results of §§ 2 and 3 to derive three methods for modifying the Gauss–Newton method so that convergence is obtained for problems with rank-deficient Jacobians and an adequate rate of convergence (normally quadratic) is obtained on problems for which $F(x)$ is not small at the solution. Each method is characterized by whether or not B or some approximation to it is available. In the first the matrix B is assumed available, in the second B is approximated by finite differences along the columns of V_2 and in the third B is estimated using a quasi-Newton updating scheme. All three algorithms require the gradient vector.

Given the matrix H which approximates B, each algorithm performs the single step of the iterative scheme (8) with $B = H$. It is important to note that if the grade of J is equal to its rank, then it can be shown that $\bar{p}_1^{(1)}$ is the traditional Gauss–Newton direction. When the grade is less than the rank we can interpret $\bar{p}_1^{(1)}$ as being the Gauss–Newton direction in the subspace spanned by the columns of V_1 and we define it as the *graded Gauss–Newton direction.* With this interpretation of $\bar{p}_1^{(1)}$, $\bar{p}_2^{(1)}$ is the correction to the graded Gauss–Newton direction which we require to ensure convergence.

In all three algorithms, traditional Gauss–Newton steps are always taken while the function is being satisfactorily reduced. If the relative decrease in the function value is less than 1% a corrected graded Gauss–Newton step is taken. The grade of J used for this one step is that which approximately balances the condition numbers of S_1 and S_2. If s_l is the last nonzero singular value of J, then the grade r selected is that for which

$$s_1/s_r + s_{r+1}/s_l \leqq s_1/s_j + s_{j+1}/s_l, \qquad j = 1, 2, \cdots, l-1.$$

After the first corrected step a sequence of corrected steps are taken until the reduction in F exceeds 10%, in which case we return to the traditional Gauss–Newton technique. The grade for the repeated corrected steps again depends upon the progress made during the last iteration. If the function was reduced by at least 1% the grade is unchanged; otherwise the grade is reduced by at least one.

The method used to reduce the grade is not critical, but it is useful to include clusters of similar singular values into either S_1 and S_2 as required. On a machine with a mantissa of t binary digits, let the original grade be chosen to give a matrix S_1 such that $s_r \geqq 10^k 2^{-t}$ and $s_{r+1} < 10^k 2^{-t}$ for some integer k; if q is the smallest integer such that

$$s_{r'} \geqq 10^{k+q} 2^{-t}, \quad s_{r'+1} < 10^{k+q} 2^{-t} \quad \text{and} \quad r' < r,$$

then r' is the reduced value of the grade.

SUMMARY: GENERAL ALGORITHM FOR NONLINEAR LEAST-SQUARE PROBLEMS.

Step I Set $x^{(0)}$, $f^{(0)}$, $J^{(0)}$ and $g^{(0)} = 2J^{(0)T} f^{(0)}$.

Step II. If $x^{(k)}$ is an adequate approximation to a stationary point, the algorithm is terminated (see § 5 for the exact termination criteria used); otherwise continue at Step III.

Step III. Compute the singular-value decomposition of $J^{(k)}$:

$$J^{(k)} = U \begin{bmatrix} S \\ 0 \end{bmatrix} V^T.$$

Step IV. Select r, the grade of $J^{(k)}$ according to the rules given in § 4.

Step V. Compute the Gauss–Newton direction in the space spanned by V_1:

$$p_1 = -V_1 S_1^{-1} f_1.$$

Step VI. If a correction to the Gauss–Newton direction is not required, set $p_2 = 0$ and continue at Step VII. Otherwise compute or approximate the matrices $Y = V_2^T B^{(k)}$ and $Q = YV_2$. Use the modified LDL^T factorization to solve the equations

$$(S_2^2 + Q)y = -S_2 f_2 - Yp_1,$$

and set $p_2 = V_2 y$.

Step VII. Set $p^{(k)}=p_1+p_2$. Let σ $(\sigma>0)$ be a small pre-assigned scalar: if $-g^{(k)T}p^{(k)}/(\|g^{(k)}\|\,\|p^{(k)}\|)<\sigma$ and $r>0$ then set $r=0$ and return to Step V to recompute $p^{(k)}$.

Step VIII. Compute a steplength $\alpha^{(k)}$ such that

$$F^{(k)}-F(x^{(k)}+\alpha^{(k)}p^{(k)})>\phi\left(\frac{-g^{(k)T}p^{(k)}}{\|g^{(k)}\|\,\|p^{(k)}\|}\right),$$

where $\phi(t)$ is a function such that $\lim_{k\to\infty}\phi(t_k)=0$ implies that $\lim_{k\to\infty}t_k=0$. (Gill and Murray (1974a) give a description of one method for computing such a value of $\alpha^{(k)}$. Safeguarded cubic or quadratic interpolation is used to obtain a value of $\alpha^{(k)}$ such that $|g(x^{(k)}+\alpha^{(k)}p^{(k)})^Tp^{(k)}|\leqq-\eta g^{(k)T}p^{(k)}$, where $\eta\,(0\leqq\eta<1)$ is a fixed scalar specified by the user. If necessary, $\alpha^{(k)}$ is then reduced by a fixed multiple until

$$F^{(k)}-F(x^{(k)}+\alpha^{(k)}p^{(k)})\geqq-10^{-4}\alpha^{(k)}g^{(k)T}p^{(k)}).$$

Step IX. Compute $x^{(k+1)}=x^{(k)}+\alpha^{(k)}p^{(k)}$, $f^{(k+1)}$, $J^{(k+1)}$ and $g^{(k+1)}$; set $k=k+1$ and continue at Step II.

4.1. A second-derivative algorithm. Nonlinear least-square algorithms for problems in which the second derivatives can be provided have received little attention in the literature. This situation has probably arisen because it was thought that Newton-type methods for general unconstrained minimization were appropriate. General minimization algorithms have the unsatisfactory feature of forming the direction of search.by solving the normal equations.

When second derivatives are available, Step VI can be evaluated with $Y=V_2^TB^{(k)}$ and $Q=YV_2$. For large-residual problems, if $S_2^2+V_2^TBV_2$ is not positive definite at a stationary point, Step II can be augmented by a procedure for computing the direction of negative curvature using (12). Provided the number of stationary points is finite, this added facility gives convergence to a local minimum rather than a stationary point. (We have not included a rigorous proof of convergence due to its close similarity to Theorem 2.6.2 given by Gill and Murray (1974b, pp. 327–328). The proof depends upon establishing that the quantity $-g^{(k)T}p^{(k)}/(\|g^{(k)}\|\,\|p^{(k)}\|)$ can be uniformly bounded away from zero as in (11). This is enforced by Step VII when $r>0$. The steps for which $r=0$ do not impede the convergence of the algorithm since the subsequence of such steps is itself convergent.)

Forming the analytical second derivatives and programming them in a subroutine requires a significant amount of effort. However, the following example will serve to illustrate that second derivatives can often be provided very cheaply. The problem of minimizing a sum of squares arises naturally from the problem of determining parameters x_j, $j=1,2,\cdots,n$, in the model equation

$$y(t)=\theta(t,x)$$

from observations

$$y_i=y(t_i)+\varepsilon_i,\qquad i=1,2,\cdots,m,$$

where the ε_i are experimental errors. The appropriate maximum-likelihood analysis indicates that x should be estimated by minimizing the so-called "model problem"

$$\text{minimize}\left\{F(x)=\sum_{i=1}^{m}(y_i-\theta(t_i,x))^2\right\}.$$

These problems have two important properties. Firstly, the values of all the f_i (and their first and second derivatives) can be obtained using a *single* subroutine which evaluates $\theta(t, x)$ for a particular value of t_i and x. Secondly, $\theta(t, x)$ is often a simple linear combination of functions such as exponentials, trigonometrical functions or powers of t whose higher derivatives can be computed almost as cheaply as the value of the function itself. These properties imply that the second-derivative algorithm is particularly effective for the model problem.

4.2. An algorithm using finite-difference approximations to second derivatives. If second derivatives are not available, an algorithm using only first derivatives can be constructed by using the same iteration as that of the second-derivative algorithm but with the matrices Y and Q of Step VI approximated by finite differences.

Let v be any column of the matrix V_2. If h is a small positive scalar, then

$$\frac{1}{h}(\nabla f_i(x^{(k)}+hv)-\nabla f_i(x^{(k)}))=v^TG_i^{(k)}+O(h).$$

The vector on the left hand side of this equation is just a row of the matrix $(J(x^{(k)}+hv)-J(x^{(k)}))/h$ and consequently we can write

$$\frac{1}{h}(J(x^{(k)}+hv)-J(x^{(k)}))=\begin{bmatrix} v^TG_1^{(k)} \\ v^TG_2^{(k)} \\ \vdots \\ v^TG_m^{(k)} \end{bmatrix}+O(h).$$

Premultiplying by $f(x^{(k)})^T$ gives

$$\begin{aligned}\frac{1}{h}f^{(k)T}(J(x^{(k)}+hv)-J(x^{(k)}))&=\sum_{i=1}^{m} v^T(f_i^{(k)}G_i^{(k)})+O(h)\\ &=v^TB^{(k)}+O(h).\end{aligned}$$

If we difference $J(x)$ along each of the columns of V_2 in turn we obtain the matrices $\tilde{Y}$ and $\tilde{Q}$ which are $O(h)$ approximations to the matrices Y and Q of Step VI of the general iteration.

The value of r, the grade of the Jacobian matrix, is generally equal to the number of dominant singular values. Since the number of dominant singular values is rarely significantly less than n, the $n-r$ gradient evaluations required to approximate $V_2^TB^{(k)}V_2$ when a corrected Gauss–Newton step is taken are usually few in number.

This algorithm has very similar performance to the second-derivative algorithm and under an additional condition on the magnitude of the smallest eigenvalue of the Hessian matrix of second derivatives at the stationary points of $F(x)$, the algorithm will converge to a local minimum.

4.3 An algorithm using a quasi-Newton approximation to *B*. In the first-derivative algorithm presented in the last section, the gradient vector is computed at every point $x^{(k)}$ regardless of the type of direction of search used. These evaluations can be utilized to give a quasi-Newton approximation to $B(x)$ rather than a finite-difference approximation to $B(x)$.

Given $H^{(k)}$, the kth approximation to $B(x)$, on the completion of the kth iteration we wish to compute a matrix $H^{(k+1)}$ such that

$$(J^{(k+1)T}J^{(k+1)}+H^{(k+1)})\Delta x^{(k)}=J^{(k+1)T}f^{(k+1)}-J^{(k)T}f^{(k)}, \tag{13}$$

where $\Delta x^{(k)}=x^{(k+1)}-x^{(k)}$. This is the so-called "quasi-Newton condition" for approximating the Hessian matrix of $F(x)$ and the use of the quasi-Newton approximation $H^{(k+1)}$ is obtained by adding a correction matrix $C^{(k)}$ of rank one or two to the matrix $H^{(k)}$ giving

$$H^{(k+1)}=H^{(k)}+C^{(k)}. \tag{14}$$

The quasi-Newton scheme satisfying both (13) and (14) which was selected for use here is

$$H^{(0)}=0, \qquad C^{(k)}=\frac{1}{\alpha^{(k)}y^{(k)T}p^{(k)}}y^{(k)}y^{(k)T}-\frac{1}{p^{(k)T}W^{(k)}p^{(k)}}W^{(k)}p^{(k)}p^{(k)T}W^{(k)}, \tag{15}$$

where $W^{(k)}=J^{(k+1)T}J^{(k+1)}+H^{(k)}$ and $y^{(k)}=J^{(k+1)T}f^{(k+1)}-J^{(k)T}f^{(k)}$. This formula is based upon the complementary DFP updating rule (sometimes known as the BFGS formula) and its use for general minimization has been described by Gill, Murray and Pitfield (1972) amongst others.

The steplength algorithm can be used to ensure that $y^{(k)T}p^{(k)}>0$ (see the gradient-vector inequality given in Step VIII of the iteration of § 5), in which case the updating formula has the property that if $J^{(k+1)T}J^{(k+1)}+H^{(k)}$ is positive definite, then so is $J^{(k+1)T}J^{(k+1)}+H^{(k+1)}$. Note that unlike other methods (see Nazareth (1976) for a survey) ours does not require each $H^{(k)}$ to be positive definite, which is sensible because the matrix $B(x)$ may or may not be positive definite at any point, including the solution.

Formula (15) is one of many possible updating formulae available. However, we found that in the least-square context, the complementary DFP formula generally required fewer function evaluations than its competitors, including the symmetric rank-one formula and the optimally-conditioned formula of Davidon (1975).

In practice, the quasi-Newton algorithm did not perform as well as expected. With large-residual problems only a linear rate of convergence was achieved near the solution (this is in contrast to the superlinear rate of convergence which is often obtained for quasi-Newton methods for general unconstrained minimization). It is our view that with the quasi-Newton techniques currently available, it is not possible to achieve a superlinear rate of convergence for large-residual problems. This is because the matrix $H^{(k)}$ is a very poor approximation to $B(x)$. In *general* unconstrained optimization the quasi-Newton condition is just one of n conditions on the $(k+1)$st approximation to the Hessian matrix (for example, we may require the quasi-Newton condition to apply for some of the previous steps $\Delta x^{(j)}$ and the gradient differences $y^{(j)}$). These conditions can be used to ensure that if the objective function is quadratic and consequently we are approximating a constant matrix, then the quasi-Newton approximation converges to the exact Hessian in n steps. However, in the context of nonlinear least-square calculations $B(x)$ is only a constant matrix if it is zero. Since we are attempting to approximate a matrix which varies with x even in the quadratic case, properties such as n-step convergence have no meaning.

Recently, Betts (1976) has suggested a nonlinear least-square algorithm based upon using a quasi-Newton approximation to $B(x)$. Betts' algorithm uses the rank-one updating scheme with $J^{(k)}$ being used in the quasi-Newton condition (13) rather than $J^{(k+1)}$. This gives the following correction $C^{(k)}$ for $H^{(k)}$:

$$C^{(k)}=\frac{1}{q^{(k)T}\Delta x^{(k)}}q^{(k)}q^{(k)T}, \tag{16}$$

where $q^{(k)} = y^{(k)} - (J^{(k)T}J^{(k)} + H^{(k)})\Delta x^{(k)}$. Betts has observed a *quadratic* rate of convergence near the solution with his algorithm (see Betts (1976, p. 101)). In order to compare the alternative updating formulae (15) and (16), a number of test problems were solved (for details of the computer runs see § 5.2). Unfortunately we were unable to verify Betts' findings, the rate of convergence of (16) for large-residual problems being the same as that for (15).

5. Numerical tests. The three methods described in §§ 4.1, 4.2 and 4.3 form the theoretical basis of subroutines LSQSDN, LSQFDN and LSQFDQ respectively which are part of the NPL Algorithms Library (ref. nos E4/17/F, E4/18/F and E4/21/F). In order to demonstrate the effectiveness of the methods, numerical results were obtained for a number of test examples appearing in the literature.

Unfortunately, test problems for nonlinear least-square algorithms are not fully representative of problems arising in real situations. The most important application of nonlinear least-square algorithms is solving the model problem described in § 4.1. These problems require a large amount of data to define them and it is not helpful publishing performance results if they cannot be easily duplicated by other research workers. We do not assume therefore that the set of test problems is exhaustive or claim that the behavior exhibited will be typical for all functions.

5.1. The test problems. For brevity, the full details of the problems are not listed here. For further information the reader is referred to Gill and Murray (1976a) and the references quoted.

Each problem is referred to by the name of its originator, with the number of variables and reference following in parenthesis.

Rosenbrock ($n=2$; Rosenbrock (1960)), Helix ($n=3$; Fletcher and Powell (1963)), Singular ($n=4$; Powell (1962)), Woods ($n=4$; Colville (1968)), Zangwill ($n=3$; Zangwill (1967)), Engvall ($n=3$; Engvall (1966)), Branin ($n=2$; Branin (1971)), Beale ($n=2$; Beale (1958); Betts (1976)), Cragg and Levy ($n=4$; Cragg and Levy (1969)), Box ($n=3$; Box (1966); Betts (1976)), Davidon 1 (variable dimension; Davidon (1976)), Freudenstein and Roth ($n=2$; Freudenstein and Roth (1963)), Watson (variable dimension; Brent (1971); Gill, Murray and Pitfield (1972)), Chebyquad (variable dimension; Fletcher (1965); Gill, Murray and Pitfield (1972)), Davidon 2 ($n=4$; Davidon (1976)), Bard ($n=3$; Bard (1970)), Jennrich and Sampson ($n=2$; Jennrich and Sampson (1968)), Kowalik and Osborne ($n=4$; Kowalik and Osborne (1968)), Osborne 1 ($n=5$; Osborne (1972)), Osborne 2 ($n=11$; Osborne (1972)), and Madsen ($n=2$; Madsen (1973)).

5.2. Details of the numerical experiments. With every test problem the computation was carried out in single precision on a CDC 6500 computer with a 60 bit binary mantissa.

An iterate $x^{(k)}$ was accepted as a close approximation to a stationary point if the following criteria were satisfied:

$$\{\alpha^{(k)}\|p^{(k)}\|_2 \leqq (\text{tol}+\varepsilon)(1+\|x^{(k)}\|_2)$$

and

$$|F^{(k)} - F^{(k-1)}| \leqq (\text{tol}+\varepsilon)^2(1+F^{(k)})$$

and

$$\|g^{(k)}\|_2 \leqq \varepsilon^{1/3}(1+F^{(k)})\}$$

or

$$\|g^{(k)}\|_2^2 < \{F^{(k)}\}^{1/2}\varepsilon \quad \text{or} \quad F^{(k)} < \varepsilon^2,$$

where tol $= 10^{-6}$ and ε is the relative machine precision.

A steplength algorithm based upon safeguarded cubic interpolation was used to compute $\alpha^{(k)}$ (see Step VIII of the basic iteration) with termination criterion $\eta = 0.9$ and initial value of $\alpha^{(k)} = 1$. This choice of η and initial $\alpha^{(k)}$ implied that a unit step along $p^{(k)}$ was nearly always accepted provided that the sum of squares was sufficiently reduced.

The programs contain the facility for limiting the magnitude of $\alpha^{(k)}$ during each iteration. This bound was set at 10^5 on all the examples except Box, Jennrich and Sampson, Osborne 1 and Osborne 2, where it was set at 10 to avoid possible machine overflow while evaluating $F(x)$.

Tables I–III contain the results obtained using the algorithms described in §§ 4.1, 4.2 and 4.3 respectively. The columns headed "intermediate accuracy" give the number of iterations (n_{it}) and number of function evaluations (n_{f}) necessary to reduce

TABLE I

Results obtained using the nonlinear least-square algorithm which utilizes the exact matrix B.

Problem	*Intermediate Accuracy*		*Final Accuracy*	
	n_{it}	n_{f}	n_{it}	n_{f}
ROSENBROCK*	12	31	12	31
HELIX*	10	13	11	14
SINGULAR*	11	12	13	14
WOODS*	36	76	36	76
ZANGWILL*	1	2	1	2
ENGVALL*	9	15	9	15
BRANIN*	1	2	1	2
BEALE*	6	13	7	14
CRAGG AND LEVY*	11	13	13	15
BOX*	4	5	5	6
DAVIDON 1*	1	2	1	2
FREUDENSTEIN AND ROTH*†	7	17	8	18
WATSON, $n = 6$	5	6	6	7
WATSON, $n = 9$	4	5	5	6
WATSON, $n = 20$	4	5	4	5
DAVIDON 2	21	43	22	46
BARD	4	5	5	6
JENNRICH AND SAMPSON	9	32	10	35
KOWALIK AND OSBORNE	7	10	8	11
OSBORNE 1	7	11	8	12
OSBORNE 2	9	17	10	18
MADSEN	9	11	9	11

* Zero function value at the solution.
† Local minimum found.

the sum of squares below $10^{-10}(F(\overset{*}{x})+1)$. The columns headed "final accuracy" give the number of iterations and number of function evaluations necessary to satisfy the final convergence criterion.

Table IV contains the results obtained from an implementation of Betts' method.

6. Conclusions. We have presented the theoretical basis of a suite of algorithms for the solution of nonlinear least-square problems. The algorithms are based upon modifying the Gauss–Newton method so that it converges when the Jacobian matrix is rank deficient and the sum of squares is not small near the solution.

Nonlinear least-square problems can be classified in three categories, according to the size of the residual relative to $\|J^TJ\|$ at the solution, the first being those problems with zero residual, the second being those with small but nontrivial residuals and finally those with a large residual. It is our view that if $F(x)$ is large relative to $\|J^TJ\|$ at the solution then there is little to be gained by using a special nonlinear least-square algorithm instead of a general unconstrained minimization algorithm. If the problem is a genuine data-fitting problem then by implication the residuals should be small otherwise the solution is of no value. In data-fitting problems, what constitutes an adequate fit (say 5% agreement with observed values) may still result in a

TABLE II

Results obtained using the nonlinear least-square algorithm with finite-difference apiroximations to B.

Problem	*Intermediate Accuracy*		*Final Accuracy*	
	n_{it}	n_{f}	n_{it}	n_{f}
ROSENBROCK*	12	31	12	31
HELIX*	10	13	11	14
SINGULAR*	11	12	13	14
WOODS*	36	115	36	115
ZANGWILL*	1	2	1	2
ENGVALL*	9	15	9	15
BRANIN*	1	2	1	2
BEALE*	6	13	7	14
CRAGG AND LEVY*	11	13	13	15
BOX*	4	5	5	6
DAVIDON 1*	1	2	1	2
FREUDENSTEIN AND ROTH*†	7	21	8	24
WATSON, $n=6$	5	8	6	14
WATSON, $n=9$	4	5	5	6
WATSON, $n=20$	4	5	4	5
CHEBYQUAD, $n=8$	21	171	22	180
DAVIDON 2	21	60	22	67
BARD	4	5	5	7
JENNRICH AND SAMPSON	9	38	10	43
KOWALIK AND OSBORNE	7	16	8	21
OSBORNE 1	7	11	8	13
OSBORNE 2	9	20	10	32
MADSEN	9	16	9	16

* Zero function value at the solution.
† Local minimum found.

nonzero and nontrivial $F(x)$. If J is mildly ill-conditioned (say with condition number of the order of 100) the eigenvalues of the Hessian matrix may be different in sign to those of J^TJ and methods based on using J^TJ only are inadequate. The algorithms proposed in this paper are particularly effective in this situation.

The algorithms have been implemented in FORTRAN within the NPL Algorithms Library and a detailed description of the programs is given in the NPL Algorithms Library documents ref. nos. E4/17/F, E4/18/F and E4/21/F. Although the quasi-Newton algorithm (E4/21/F) and finite-difference algorithm (E4/18/F) require the same user-specified information, they perform differently on practical problems. In general the quasi-Newton algorithm and finite-difference algorithm require similar numbers of function and Jacobian evaluations but the finite-difference algorithm will occasionally successfully compute a solution when the quasi-Newton algorithm fails to make progress.

Acknowledgments. The authors are indebted to Nicholas Gould for obtaining the numerical results and to David Martin and Maurice Cox for their careful reading of the manuscript and a number of helpful suggestions.

TABLE III

Results obtained using the nonlinear least-square algorithm with quasi-Newton approximations to B.

Problem	*Intermediate Accuracy*		*Final Accuracy*	
	n_{it}	n_f	n_{it}	n_f
ROSENBROCK*	12	31	12	31
HELIX*	10	13	11	14
SINGULAR*	11	12	13	14
WOODS*	54	73	54	73
ZANGWILL*	1	2	1	2
ENGVALL*	9	15	9	15
BRANIN*	1	2	1	2
BEALE*	6	13	7	14
CRAGG AND LEVY*	11	13	13	15
BOX*	4	5	4	5
DAVIDON 1*	1	2	1	2
FREUDENSTEIN AND ROTH*†	14	23	18	27
WATSON, $n = 6$	15	18	20	23
WATSON, $n = 9$	4	5	5	6
WATSON, $n = 20$	4	5	4	5
CHEBYQUAD, $n = 8$	25	92	29	96
DAVIDON 2	31	53	35	57
BARD	4	5	8	9
JENNRICH AND SAMPSON	18	41	22	45
KOWALIK AND OSBORNE	10	13	18	21
OSBORNE 1	7	11	11	23
OSBORNE 2	19	28	23	32
MADSEN	17	19	22	24

* Zero function value at the solution.

† Local minimum found.

TABLE IV
Results obtained using Betts' updating rule.

Problem	Intermediate Accuracy		Final Accuracy	
	n_{it}	n_f	n_{it}	n_f
ROSENBROCK*	12	31	12	31
HELIX*	10	13	11	14
SINGULAR*	11	12	13	14
WOODS*	48	98	52	110
ZANGWILL*	1	2	1	2
ENGVALL*	9	15	9	15
BRANIN*	1	2	1	2
BEALE*	6	13	7	14
CRAGG AND LEVY*	11	13	13	15
BOX*	4	5	5	6
DAVIDON 1*	1	2	1	2
FREUDENSTEIN AND ROTH*†	15	27	19	31
WATSON, $n=6$	6	8	10	12
WATSON, $n=9$	4	5	5	6
WATSON, $n=20$	4	5	4	5
CHEBYQUAD, $n=8$	26	96	30	100
DAVIDON 2	31	53	35	57
BARD	4	5	9	11
JENNRICH AND SAMPSON	19	40	23	44
KOWALIK AND OSBORNE	12	16	19	23
OSBORNE 1	7	11	8	12
OSBORNE 2‡	—	—	68	135
MADSEN	17	19	21	23

* Zero function value at the solution.
† Local minimum found.
‡ Failed to achieve desired accuracy.

REFERENCES

Y. BARD (1970), *Comparison of gradient methods for the solution of nonlinear parameter estimation problems*, this Journal, 7, pp. 157–186.

E. M. L. BEALE (1958), *On an iterative method for finding a local minimum of a function of more than one variable*, Tech. Rep. 25, Statistical Techniques Research Group, Princeton Univ., Princeton, NJ.

J. T. BETTS (1976), *Solving the nonlinear least square problem: Application of a general method*, J. Optimization Theory Appl. 18, pp. 469–483.

M. J. BOX (1966), *A comparison of several current optimization methods and the use of transformations in constrained problems*, Comput. J., 9, pp. 67–77.

F. H. BRANIN, JR. (1972), *Widely convergent method for finding multiple solutions of simultaneous nonlinear equations*, IBM J. Res. Develop., 16, pp. 504–522.

R. P. BRENT (1973), *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, NJ.

A. R. COLVILLE, (1968), *A comparative study of nonlinear programming codes*, IBM Tech. Rep. 320–2949, IBM T. J. Watson Res. Center, Yorktown Heights, NY.

E. E. CRAGG AND A. V. LEVY (1969), *Study on a supermemory gradient method for the minimization of functions*, J. Optimization Theory Appl., 4, pp. 191–205.

W. C. DAVIDON, (1975), *Optimally conditioned optimization algorithms without line searches*, Math. Programming, 9, pp. 1–30.

——— (1976), *New least square algorithms*, J. Optimization Theory Appl., 18, pp. 187–197.

J. E. DENNIS (1973), *Some computational techniques for the nonlinear least squares problem*, Numerical Solution of Systems of Nonlinear Algebraic Equations, Byrne and Hall, eds., Academic Press, New York and London.

J. L. ENGVALL (1966), *Numerical algorithm for solving over determined systems of nonlinear equations*, NASA document N70-35600.

R. FLETCHER (1965), *Function minimization without evaluating derivatives—a review*, Comput. J., 8, pp. 33-41.

——— (1968), *Generalized-inverse methods for the best least square solution of systems of nonlinear equations*, Ibid., 10, pp. 392-399.

R. FLETCHER AND M. J. D. POWELL (1963), *A rapidly convergent descent method for minimization*, Ibid., 6, pp. 163-168.

F. FREUDENSTEIN AND B. ROTH (1963), *Numerical solution of systems of nonlinear equations*, J. Assoc. Comput. Mach., 10, pp. 550-556.

P. E. GILL, W. MURRAY AND R. A. PITFIELD, (1972), *The implementation of two revised quasi-Newton algorithms for unconstrained optimization*, Rep. NAC 11, Nat. Physical Lab., Teddington, Middlesex, England.

P. E. GILL AND W. MURRAY (1974a), *Safeguarded steplength algorithms for optimization using descent methods*, Rep. NAC 37, Nat. Physical Lab., Teddington, Middlesex, England.

——— (1974b), *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Programming, 7, pp. 311-350.

——— (1976), *Nonlinear least squares and nonlinearly constrained optimization*, Numerical Analysis, Lecture Notes in Mathematics no. 506, G. A. Watson, ed., Springer-Verlag, Berlin-Heidelberg-New York, pp. 134-147.

——— (1976a), *Algorithms for the solution of the nonlinear least squares problem*, Rep. NAC 71, Nat. Physical Lab., Teddington, Middlesex, England.

R. I. JENNRICH AND P. F. SAMPSON, (1968), *Application of stepwise regression to nonlinear estimation*, Technometrics, 10, no. 1.

J. S. KOWALIK AND M. R. OSBORNE (1968), *Methods for Unconstrained Optimization Problems*, American Elsevier, New York.

C. L. LAWSON AND J. R. HANSON (1974), *Solving Least Squares Problems*, Prentice Hall, Englewood. Cliffs, NJ.

K. MADSEN (1973), *An algorithm for minimax solution of overdetermined systems of nonlinear equations*, Rep. TP 559, AERE, Harwell, England.

J. L. NAZARETH (1976), *Some recent approaches to solving large-residual problems*, Computer Science and Statistics, 9th Annual Symposium on the Interface, Boston.

M. R. OSBORNE (1972), *Some aspects of nonlinear least squares calculations*, Numerical Methods for Nonlinear Optimization, F. Lootsma, ed., Academic Press, New York and London.

M. J. D. POWELL (1962), *An iterative method for finding stationary values of a function of several variables*, Comput. J., 5, pp. 147-151.

H. H. ROSENBROCK (1960), *An automatic method for finding the greatest or least value of a function*, Ibid., 3, pp. 175-184.

W. I. ZANGWILL (1967), *Nonlinear programming via penalty functions*, Management Sci., 13, pp. 344-358.