J. Austral. Math. Soc. Ser. B 26 (1985), 387–403

AN INEXACT LEVENBERG-MARQUARDT METHOD FOR LARGE SPARSE NONLINEAR LEAST SQUARES

S. J. WRIGHT AND J. N. HOLT[1]

(Received 19 September 1983; revised 17 January 1984)

Abstract

A method for solving problems of the form $\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} f_i^2(x)$ is presented. The approach of Levenberg and Marquardt is used, except that the linear least squares subproblem arising at each iteration is not solved exactly, but only to within a certain tolerance. The method is most suited to problems in which the Jacobian matrix is sparse. Use is made of the iterative algorithm LSQR of Paige and Saunders for sparse linear least squares.

A global convergence result can be proven, and under certain conditions it can be shown that the method converges quadratically when the sum of squares at the optimal point is zero.

Numerical test results for problems of varying residual size are given.

1. Introduction

In this paper we are concerned with the class of large scale nonlinear least squares problems for which the Jacobian is sparse. Problems of this nature arise in important practical contexts. One example is the joint inversion of first arrival times of p-waves from a set of earthquakes at a network of seismic stations. Estimates of time and space coordinates of the earthquakes are obtained, together with an estimate of the velocity structure of the earth through which the waves are propagated (Lee and Stewart [11]). Another example is the problem of large scale geodesic adjustment (Golub and Plemmons [8]).

There appears to be a genuine lack of codes specifically for large sparse nonlinear problems. A recent evaluation of software for nonlinear least squares

[1]Department of Mathematics, University of Queensland, St. Lucia, 4067, Queensland.

problems was presented by Hiebert [9]. Only one of the twelve codes tested by Hiebert, the NS03A code from Harwell, made use of sparsity in the Jacobian. The test problems used by Hiebert were small, with no more than 65 functions and 40 variables.

Of the eight codes tested by Hiebert which required the Jacobian to be supplied analytically, five used the Levenberg-Marquardt method [12, 13], and three used the augmented Gauss-Newton method (see, for example, [5]). An interesting conclusion of Hiebert's paper is that on the basis of the testing with the specific implementations of the Levenberg-Marquardt method and the augmented Gauss-Newton methods, there does not appear to be any superiority of one class of method over the other. Since about half of the test problems had nonzero residuals, this observation runs contrary to the view held by some that the Levenberg-Marquardt method is inferior on such problems.

The method described in this paper is based on the Levenberg-Marquardt strategy. The main difficulty in applying the standard implementations of this strategy to large sparse problems is the necessity of computing a decomposiiton of the Jacobian in order to solve the linearized subproblem at each iteration. In the more stable recent codes, orthogonal decompositions are used. These are impractical for our class of problems because they have poor sparsity-preserving properties and large in-core storage requirements.

There are two approaches to solving the sparse linearized subproblems which are worthy of consideration. The first is the use of direct methods based on sparsity-preserving pivoting techniques in the formation of an LU decomposition of the matrix (see, for example, Bjorck and Duff [2]). The NS03A code from Harwell, mentioned above, appears to use a direct approach to solve the subproblem exactly at each step. The second approach, adopted by this paper, is the use of an iterative solution method. We have chosen the algorithm LSQR, devised by Paige and Saunders [16] for the solution of damped linear least squares problems. This algorithm, based on the bidiagonalization procedure of Golub and Kahan [7] is analytically equivalent to the standard method of conjugate gradients, but is shown to have superior numerical properties.

The approach we adopt is to use LSQR to obtain inexact solutions of the linearized subproblem at each Levenberg-Marquardt step, by curtailing the LSQR iterations according to criteria which guaranteed overall convergence of our method. We hope to achieve a substantial reduction in the sum of squares by applying only a few iterations of LSQR to the subproblem, and hence to avoid the extra computation involved in finding an exact solution at each step.

Inexact Newton and quasi-Newton methods have received some attention in the literature in recent years. A Newton-Kantorovich theorem for damped inexact Newton methods has been proved by Altman [1]. Dembo, Eisenstat, and Steihaug

[3], and Dembo and Steihaug [4] have proved convergence results for damped inexact Newton methods applied to optimization and to the solution of non-linear systems of equations. They show that the rate of convergence of such methods can be controlled by imposing bounds on the accuracy of solution of the linear subproblem at each iteration. Gill, Murray, and Nash [6] have discussed the use of preconditioned conjugate gradient methods in this context. Steihaug [17] presents an algorithm for large-scale optimization based on an inexact quasi-Newton method. He uses a trust region approach and finds approximate solutions of the linear subproblems by use of a preconditioned conjugate gradient technique. It is shown that his algorithm has the same convergence properties as the corresponding exact methods.

In Section 2 of this paper we give an outline of the inexact Levenberg-Marquardt algorithm. A global convergence result is proved in Section 3. In Section 4, use is made of a result quoted in Dembo and Steihaug [4] to prove that the method can be made to converge quadratically when the sum of squares is zero at the minimum point. Section 4 contains some numerical results for large- and small-residual problems.

2. Outline of the algorithm

2.1 Statement of the problem

We aim to solve the problem

$$\min_{x \in \mathbf{R}^n} F(x) = \|f(x)\|_2^2 = f(x)^T f(x)$$

where $f(x) \in \mathbf{R}^m$, $m > n$, and the Jacobian of $f(x)$, denoted by

$$J(x) = \left[\frac{\partial f_i}{\partial x_j}\right]_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}}$$

is a sparse matrix.

The gradient g of F will thus be given by

$$g(x) = 2J(x)^T f(x)$$

and the Hessian G satisfies

$$\tfrac{1}{2}G(x) = J(x)^T J(x) + \sum_{l=1}^{m} f_l(x) \nabla^2 f_l(x)$$

where f_l denotes the lth component of f.

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

The standard Levenberg-Marquardt approach to solving such problems is to linearize the vector f about the current point x, and solve the linear least squares problem

$$\min_{\delta x \in \mathbb{R}^n} \|J\delta x + f\|_2^2 + \lambda^2 \|\delta x\|_2^2.$$

The second term is introduced to control the length of δx by the use of the damping parameter λ, and to ensure global convergence.

We require that f satisfy certain smoothness conditions, to be specified in later sections. Unless otherwise stated, all vector and matrix norms are 2-norms.

2.2 The inexact Levenberg-Marquardt algorithm

The algorithm we propose may be concisely expressed in the following way:

0: Given x_0, find $f(x_0)$ and $J(x_0)$.
Set $k := 0, \lambda := 0$.
1: Choose η_k with $0 < \eta_k \leqslant \eta_0$.
Set $\tau := \eta_k \|J^T f\|$.
2: Perform iterations of LSQR until $\|(J^T J + \lambda^2 I)y + J^T f\| \leqslant \tau$;
Set $\rho = (f(x) - F(x + y))(F(x) - \|f + Jy\|^2 - \lambda^2 \|y\|^2)^{-1}$
if $\rho < \pi_1$
then go to 3
else go to 4.
3: *if* $\lambda = 0$
then $\lambda := \lambda_{\min}$
else $\lambda := E\lambda$;
go to 2.
4: $x := x + y$;
if convergence *then* EXIT;
evaluate $J(x)$;
$k := k + 1$;
if $\rho > \pi_2$ *then* $\lambda := D\lambda$;
if $\lambda < \lambda_{\min}$ *then* $\lambda := 0$;
go to 1.

The parameters must satisfy the following constraints;

$$0 < \pi_1 < \pi_2 < 1, \quad D < 1,$$
$$E > 1, \quad 0 < \eta_0 < 1, \quad \lambda_{\min} > 0.$$

2.3 Implementation notes

The choice $\eta_0 = 0$ would correspond to a standard Levenberg-Marquardt implementation (see, for example, Osborne [15]). However we allow inexact solution of the linear subproblem at each iteration. Strategies for choosing η_k in

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

step 1 will be discussed in later sections. Following Dembo and Steihaug [4], the sequence $\{\eta_k\}$ is known as the forcing sequence.

Most of the computational effort in each iteration of LSQR lies in the evaluation of two matrix-vector products of the form

$$J^T p \quad \text{and} \quad Jq$$

where J is the Jacobian at the current point. We can use LSQR to solve damped linear least squares problems for a number of different values of λ at once. Each additional value of λ requires only two additional rotations per iteration, plus an extra $2n$ storage locations. Since this is a very small computational price to pay, we solve the linear subproblem in step 2 not only for λ, but also for $E\lambda$, $E^2\lambda$ up to a user-specified limit. We can thus avoid extra iterations of LSQR should it become necessary to carry out step 3.

It will be noted that damping is controlled via the damping parameter λ, rather than by a trust region bound (as in Steihaug [17], Moré [14]). Our approach is intuitively reasonable for smooth unconstrained problems since, if the step sizes are not too large, the same value of λ should produce a similar damping effect on consecutive iterations. This is not true for constrained problems where a change in the active constraint manifold may result in the same value of λ producing markedly different damping effects on consecutive iterations. In such situations, a trust region approach is more desirable (Holt and Fletcher [10], Wright and Holt [18]).

3. Convergence properties

In this section we assume that $f(x)$ is twice continuously differentiable in the set

$$S = \{ x \in \mathbf{R}^n | F(x) \leqslant F(x_0) \}.$$

and hence that the Jacobian of f satisfies

$$\|J(x)\| \leqslant M$$

for all $x \in S$, and for some constant $M > 0$. The Hessian $G(x)$ will also be bounded on S.

In a similar fashion to Osborne [15], we define

$$r_\lambda = \begin{bmatrix} J \\ \lambda I \end{bmatrix} y_\lambda + \begin{bmatrix} f \\ 0 \end{bmatrix} \tag{3.1}$$

where y_λ satisfies

$$(J^T J + \lambda^2 I)\, y_\lambda = -J^T f + t \tag{3.2}$$

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

and t is some vector satisfying

$$\frac{\|t\|}{\|J^T f\|} \leqslant \eta \tag{3.3}$$

where $0 < \eta \leqslant \eta_0 < 1$.

We first show that at a nonstationary point x, the condition

$$\|f\|^2 > \|r_\lambda\|^2$$

will hold for sufficiently large λ, and in fact that the condition in step 2, namely that $\rho > \pi_1$, will eventually hold also. We then show that $\|f\| = \|r_\lambda\| \Leftrightarrow x$ is stationary, provided λ is sufficiently large, and that $p \to \pi > 1$ as $\lambda \to \infty$.

A result like Theorem 2.1 of Osborne [15] is then applied to prove global convergence of the algorithm.

LEMMA 1. *Suppose A is a symmetric positive definite matrix with eigenvalues $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n > 0$. Then if a and b are two vectors with $\|b\| \leqslant \eta\|a\|$, and $\eta^2 < \sigma_n/\sigma_1$, then*

$$a^T A a - b^T A b > 0$$

with equality holding if and only if $a = 0$.

PROOF. Now,

$$a^T A a - b^T A b \geqslant \sigma_n \|a\|^2 - \sigma_1 \|b\|^2 \geqslant (\sigma_n - \eta^2 \sigma_1)\|a\|^2$$

and since $\eta^2 < \sigma_n/\sigma_1$, we have $(\sigma_n - \eta^2\sigma_1) > 0$, and the result follows. □

LEMMA 2. *Suppose at the current point x that $J^T J$ has largest eigenvalue s_1 and smallest eigenvalue s_n. Then if λ satisfies*

$$\lambda^2 > \frac{s_1\eta^2 - s_n}{1 - \eta^2} \tag{3.4}$$

where $\|t\| \leqslant \eta\|J^T f\|$, then

$$\|f\|^2 - \|r_\lambda\|^2 \geqslant 0$$

with strict inequality holding unless x is a stationary point of F.

PROOF. Now,

$$\begin{aligned}
\|f\|^2 - \|r_\lambda\|^2 &= -y_\lambda^T (J^T J + \lambda^2 I) y_\lambda - 2 y_\lambda^T J^T f \\
\Leftrightarrow \|f\|^2 - \|r_\lambda\|^2 &= -y_\lambda^T (J^T f + t) \\
&= (J^T f - t)^T (J^T J + \lambda^2 I)^{-1} (J^T f + t) \\
&= f^T J (J^T J + \lambda^2 I)^{-1} J^T f - t^T (J^T J + \lambda^2 I)^{-1} t. \qquad (3.5)
\end{aligned}$$

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

Noting that the maximum and minimum eigenvalues of $(J^TJ + \lambda^2 I)^{-1}$ are $(s_n + \lambda^2)^{-1}$ and $(s_1 + \lambda^2)^{-1}$ respectively, the result will apply if

$$\eta^2 < \frac{(s_1 + \lambda^2)^{-1}}{(s_n + \lambda^2)^{-1}} = \frac{s_n + \lambda^2}{s_1 + \lambda^2}$$

which is equivalent to condition (3.4). The result follows from Lemma 1. □

Since we assume that s_1 is bounded, condition (3.4) can always be satisfied for some finite λ. The inverse of $(J^TJ + \lambda^2 I)$ will always exist, since if J^TJ is singular (*i.e.* $s_n = 0$), then λ will be strictly positive by (3.4). Also note that if J has full rank at the solution x^* (*i.e.* $s_n > 0$ in some neighbourhood of x^*) then for η sufficiently small, the numerator in (3.4) will be negative and hence we can satisfy the condition (3.4) by setting $\lambda = 0$.

The following lemma characterizes a stationary point and is similar to Lemma 2.1 of Osborne [15].

LEMMA 3. *Suppose at the current point x that condition* (3.4) *holds and that λ is bounded above. Also assume that condition* (3.3) *holds and that r_λ is determined by* (3.1) *and* (3.2). *Then the following conditions are equivalent*:

(i) $[f_0] = r_\lambda$,

(ii) $\|f\| = \|r_\lambda\|$,

(iii) *x is a stationary point of the objective function.*

PROOF. Clearly (i) ⇒ (ii). Also (ii) ⇒ (iii) from Lemma 2. Assume (iii), *i.e.* $J^Tf = 0$. Then since $\|t\| \leqslant \eta\|J^Tf\|$, we have $t = 0$ also and hence from (3.2), $y_\lambda = 0$. Substitution in (3.1) gives (i). □

The next lemma shows that the ratio of actual to predicted reduction in the objective function approaches a number greater than 1 as $\lambda \to \infty$, and hence that the condition $\rho > \pi_1$ in step 2 must eventually hold.

LEMMA 4. *Suppose that the current point x is nonstationary and that r_λ and y_λ are obtained from* (3.1) *to* (3.3). *Then the ratio defined by*

$$\rho = \left(F(x) - F(x + y_\lambda)\right) / \left(F(x) - \|r_\lambda\|^2\right)$$

will satisfy $\rho \to \pi \geqslant 2(1 + \eta)^{-1}$ as $\lambda \to \infty$.

PROOF.

$$\rho = \frac{\|f(x)\|^2 - \|f(x + y_\lambda)\|^2}{\|f(x)\|^2 - \|r_\lambda\|^2}.$$

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

Dropping the subscript on y_λ, and using (3.5) and Taylor's theorem,

$$\rho = \frac{-y^T g - 1/2\, y^T \hat{G} y}{-y^T (J^T f + t)} \tag{3.6}$$

where $\hat{G}$ is the Hessian of F evaluated at some point $x + \alpha y$, $0 \leqslant \alpha \leqslant 1$. Now from (3.2),

$$y = -\lambda^{-2}(J^T f - t) + O(\lambda^{-4})$$

as $\lambda \to \infty$. Thus as $\hat{G}$ is bounded,

$$\begin{aligned}
\rho &= 2\frac{\|J^T f\|^2 - t^T J^T f}{\|J^T f\|^2 - \|t\|^2} + O(\lambda^{-2}) \\
&\geqslant 2\frac{\|J^T f\|^2 - \|t\|\,\|J^T f\|}{\|J^T f\|^2 - \|t\|^2} + O(\lambda^{-2}) \\
&= 2\frac{\|J^T f\|}{\|J^T f\| + \|t\|} + O(\lambda^{-2}) \\
&\geqslant 2(1+\eta)^{-1} + O(\lambda^{-2}). \quad \square
\end{aligned}$$

Since $\eta \leqslant \eta_0 < 1$, we have

$$2(1+\eta)^{-1} \geqslant 2(1+\eta_0)^{-1} > 1 > \pi_1$$

giving the desired result. In fact, we can prove the stronger result that $\pi \to 2$ in Lemma 4. This can be seen intuitively by noting that the condition number of $(J^T J + \lambda^2 I)$ approaches 1 as $\lambda \to \infty$. Hence for sufficiently large λ, we can solve the problem to within an arbitrary tolerance using only one iteration of LSQR. At a nonstationary point ($J^T f \neq 0$), we will have $t \to 0$ as $\lambda \to \infty$ and the result follows from (3.6).

We now state a result similar to Theorem 2.1 of Osborne:

THEOREM 5. *If for any forcing sequence* $\{\eta_k\}$, $\eta_k \leqslant \eta_0 < 1$, *we can choose a bounded sequence* $\{\lambda_k\}$ *such that*

$$\rho_k \geqslant \pi_1 > 0$$

and

$$x_{k+1} = x_k + y_{k,\lambda}$$

where

$$\rho_k = \frac{F(x_k) - F(x_{k+1})}{F(x_k) - \|r_{r,\lambda}\|^2},$$

then the sequence $\{F(x_k)\}$ *is convergent, and the limit points of the sequence* $\{x_k\}$ *are stationary points of* $F(x)$.

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

PROOF. By Lemmas 2 and 4 it is possible to choose such a sequence $\{\lambda_k\}$, and so the proof follows that of Osborne. □

Note that we have introduced iteration subscripts on the quantities x, y_λ, r_λ, and ρ. These will also be needed in parts of the following section and should not cause confusion.

4. The zero-residual problem

4.1. Introduction

In this section we prove that the algorithm can be made to converge quadratically when the minimiser x^* satisfies $F(x^*) = 0$. Firstly we adapt a result of Steihaug [17] which states that for an inexact quasi-Newton method, the sequence of iterates will converge superlinearly if $\eta_k \to 0$. Secondly, we make use of a result of Dembo and Steihaug [4] in Theorem 11, namely that if the forcing sequence satisfies

$$\eta_k = \|g(x_k)\|^s$$

and the approximate Hessian $B(x_k)$ satisfies

$$\|B(x_k) - G(x_k)\| \leqslant C\|g(x_k)\|^s$$

for all k sufficiently large, then the convergence will have order $1 + s$. In our case, $B(x_k) = 2J(x_k)^T J(x_k)$ and $s = 1$.

4.2. The eventual occurrence of Gauss-Newton steps (*i.e.* $\lambda = 0$)

We assume as before that $f(x)$ is twice continuously differentiable on S. If we denote the components of f by f_l, $l = 1,\ldots,m$, then since $f_l \in C^2$, the Hessian of f_l satisfies

$$\|\nabla^2 f_l(x)\| \leqslant \beta, \qquad l = 1,\ldots,m, \tag{4.1}$$

for $x \in S$ and some positive constant β.

Since $f(x^*) = 0$ at the minimiser, the Hessian at x^* will be

$$G(x^*) = 2J(x^*)^T J(x^*).$$

For the purpose of the proofs below, we need the additional assumption that $G(x^*)$ is positive definite, that is, $J(x^*)$ must have full rank. It can now been seen that the following property will hold:

PROPERTY 6. There is a convex neighbourhood L_0 of x^*, and an index K_0 and positive constant η_0 such that

(a) $k \geqslant K_0 \Rightarrow x_k \in L_0$,

(b) $x \in L_0 \Rightarrow \|J(x)q\|^2 \geqslant m_0\|q\|^2 \ \forall q \in \mathbf{R}^n$.

The following lemma can be used to show that the damping parameter is eventually always zero.

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

LEMMA 7. *Suppose x^* is a minimiser of $F(x)$ with $F(x^*) = 0$, and that Property 6 holds. Suppose further that $y = y_{k,\lambda}$ is determined as in (3.2) and that $\|t\| \leqslant \eta\|J^T f\|$ with $\eta = O(\|J^T f\|^s)$ for some $0 < s \leqslant 1$. Then $\rho \to 2 - \xi$ as $x \to x^*$ for some $0 < \xi \leqslant 1$.*

PROOF. From (3.6) and (3.2),

$$\begin{aligned}\rho &= \frac{-y^T g - 1/2\, y^T \hat{G} y}{-y^T(J^T f + t)} \\ &= \frac{2(-y^T J^T f - y^T t) + 2y^T t - 1/2\, y^T \hat{G} y}{-y^T(J^T f + t)} \\ &= 2 - \frac{2y^T t - 1/2\, y^T \hat{G} y}{2y^T t - y^T(J^T J + \lambda^2 I)y} \qquad (4.2)\end{aligned}$$

where, again, $\hat{G}$ is the Hessian evaluated at $x + \alpha y$, for some $\alpha \in [0, 1]$.

But from (3.2),

$$\|y\| \leqslant (m_0 + \lambda^2)^{-1}(\|J^T f\| + \|t\|)$$

and since

$$\|t\| \leqslant \eta\|J^T f\| = O(\|J^T f\|^{1+s})$$

we have

$$y^T t \leqslant O(\|J^T f\|^{2+s}).$$

Since the second terms in both numerator and denominator are $O(\|J^T f\|^2)$, we can ignore $y^T t$ as $x \to x^*$ and $\|J^T f\| \to 0$. Also $\hat{G} \to J(x^*)^T J(x^*)$ as $x \to x^*$ and so from (4.2),

$$\rho \to 2 - \frac{\|Jy\|^2}{\|Jy\|^2 + \lambda^2\|y\|^2} \Leftrightarrow \rho \to 2 - \xi,$$

where clearly $\xi \leqslant 1$, and $\xi > 0$ by boundedness of λ. □

From the above lemma, we can see that eventually the condition $\rho > \pi_2$ will hold in step 4 of the algorithm at each iteration. Hence the damping parameter λ will be reduced at each iteration, until eventually it will be set to zero. Using this observation, and the positive definiteness of the Hessian at the solution, we can quote the following property:

PROPERTY 8. There is a convex neighbourhood L_1 of x^*, positive constants μ, m_1, and m_2, and an index K_1 such that

(a) $k \geqslant K_1 \Rightarrow x_k \in L_1$,

(b) $k \geqslant K_1 \Rightarrow \lambda_k = 0$,

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

(c) $k \geqslant K_1 \Rightarrow \|y_{k,\lambda}\| = \|y_{k,0}\| \leqslant \mu \|J(x_k)^T f(x_k)\|$,
(d) $m_1\|t\|^2 \leqslant t^T G(x) t \leqslant m_2\|t\|^2$ for $x \in L_1$ and all $t \in \mathbf{R}^n$.

4.3 Quadratic convergence

We define

$$B(x) = 2J(x)^T J(x)$$

so that $B_k = B(x_k)$ is effectively our approximate Hessian when $k \geqslant K_1$ in Property 8. If we define the sequence $\{\gamma_k\}$ by

$$\gamma_k = \|B_k y_{k,\lambda} + g(x_k) - g(x_k + y_{k,\lambda})\| / \|y_{k,\lambda}\|$$

then

$$\begin{aligned} \lim_{k\to\infty} \gamma_k &= \lim_{k\to\infty} \left(\|B_k y_{k,\lambda} - \hat{G} y_{k,\lambda}\| / \|y_{k,\lambda}\|\right) \quad \text{for some } 0 \leqslant \alpha \leqslant 1, \\ &\leqslant \lim_{k\to\infty} \|B_k - \hat{G}\| \\ &= 0, \qquad \text{since } B(x^*) = G(x^*) \end{aligned}$$

and hence since $\gamma_k \geqslant 0$,

$$\lim_{k\to\infty} \gamma_k = 0. \tag{4.3}$$

Making use of this result and of Property 8, we can prove the following lemma.

LEMMA 9. *If the forcing sequence $\{\eta_k\}$ satisfies $\eta_k \to 0$ then $\{x_k\}$ converges superlinearly.*

PROOF. The proof is similar to Theorem 4.2 of Steihaug [17], but simpler because we assume that the condition

$$\frac{\|t_k\|}{\|J(x_k)^T f(x_k)\|} \leqslant \eta_k$$

holds on every iteration. We can show that for $k \geqslant K_1$,

$$\frac{\|g(x_{k+1})\|}{\|g(x_k)\|} \leqslant \mu \frac{\eta_k m_2 + \gamma_k}{2(1 - \eta_k)}$$

and hence

$$\lim_{k\to\infty} \frac{\|g(x_{k+1})\|}{\|g(x_k)\|} = 0,$$

giving the result. □

We now find an error bound for the approximate Hessian $B(x_k)$.

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

LEMMA 10. *If $x_k \to x^*$ with $F(x^*) = 0$ and $G(x^*)$ positive definite, there is a constant C such that*

$$k \geqslant K_1 \Rightarrow \|B_k - G(x_k)\| \leqslant C\|g(x_k)\| = 2C\|J(x_k)^T f(x_k)\|$$

PROOF. Since each component of $f(x)$ is twice continuously differentiable, condition (4.1) applies for some $\beta > 0$. Hence

$$\begin{aligned}\|B_k - G(x_k)\|^2 &= \left\|2\sum_{l=1}^{m} f_l(x_k)\nabla^2 f_l(x_k)\right\|^2 \\ &\leqslant 4\left(\sum_{l=1}^{m} |f_l(x_k)|\,\|\nabla^2 f_l(x_k)\|\right)^2 \\ &\leqslant 4\beta^2\left(\sum_{l=1}^{m} |f_l(x_k)|\right)^2 \\ &\leqslant 4\beta^2 m\|f(x_k)\|^2 \qquad (4.4)\end{aligned}$$

since $\sum_1^m |f_l| \leqslant \sqrt{m}\,\|f\|$.

Now F is convex in the neighbourhood L_1 of x^* and so for $k \geqslant K_1$,

$$0 = F(x^*) \geqslant F(x_k) + g_k^T(x^* - x_k).$$

Thus

$$\|f(x_k)\|^2 = F(x_k) \leqslant \|g_k\|\,\|x^* - x_k\|. \qquad (4.5)$$

Also

$$g_k = g(x^*) + \hat{G}_k(x_k - x^*)$$

and so

$$x_k - x^* = \hat{G}_k^{-1} g_k$$

where $\hat{G}_k$ is $G(x)$ evaluated at some point between x_k and x^*. Using Property 8(d),

$$\|x_k - x^*\| \leqslant m_1^{-1}\|g_k\| \qquad (4.6)$$

for $k \geqslant K_1$. Combining (4.4), (4.5), and (4.6) we obtain

$$\|B_k - G(x_k)\|^2 \leqslant 4\beta^2 m m_1^{-1}\|g_k\|^2$$

and so the result holds with $C = 2\beta(m/m_1)^{1/2}$.

THEOREM 11. *If the conditions of Lemma 10 hold, and the forcing sequence $\{\eta_k\}$ satisfies*

$$\eta_k \leqslant \|g(x_k)\| \quad \text{for } k \geqslant K_2, K_1,$$

then the sequence $\{x_k\}$ converges quadratically.

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

PROOF. The result follows directly from Dembo and Steihaug [4, page 197] and Lemma 10, page 197. □

5. Discussion and numerical results

5.1 Numerical convergence criteria

Three types of convergence are recognised: x-convergence, function convergence, and gradient convergence. These require three user-supplied parameters—tolx, tolf, and tolg respectively. Termination of the routine also occurs when λ exceeds some given maximum.

The x-convergence criterion is

$$\frac{\|y\|_\infty}{\|x+y\|_\infty + \|x\|_\infty} \leqslant \text{tolx} \tag{5.1}$$

where $\|x\|_\infty = \max_{1 \leqslant j \leqslant n} |x^{(j)}|$, $x^{(j)}$ being the jth component of the vector x. Function convergence is signalled if either

$$F(x+y) \leqslant \text{tolf} \tag{5.2}$$

or

$$\frac{F(x) - F(x+y)}{F(x+y)} \leqslant \text{tolf}. \tag{5.3}$$

The gradient convergence criterion is

$$\|g(x)\| \leqslant \text{tolg} \tag{5.4}$$

The criteria (5.1), (5.2) and (5.3) are due to Dennis, Gay, and Welsch [5].

5.2 Test problems and results

We give results for three test problems.

EXAMPLE I (see Gill, Murray, and Nash [6]).

$$f_i = a(x_i - 1), \qquad i = 1,\ldots,n,$$

$$f_{n+1} = b\left[\sum_1^n x_i^2 - \frac{1}{4}\right]$$

where a and b are constants. We use $a = 1$ and $b = 10^{-3/2}$. The Jacobian matrix is $(n+1) \times n$ and has $2n$ nonzero elements. This problem has a small residual at the optimal point.

EXAMPLE II.

$$f_i = \left(x_{i_1}^{a_i} - x_{i_2}^{b_i}\right)^{c_i}, \quad i = 1,\ldots,m,$$

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

where

$$i_1 = i \bmod \frac{n}{2} + 1,$$

$$i_2 = i_1 + \frac{n}{2},$$

$$a_i = \begin{cases} 1 & \text{for } i \leqslant \frac{m}{2}, \\ 2 & \text{for } i \geqslant \frac{m}{2}, \end{cases}$$

$$b_i = 5 - i \operatorname{div} \frac{m}{4},$$

$$c_i = i \bmod 5 + 1.$$

The Jacobian matrix is double-banded with $2m$ nonzero elements. This is a zero-residual problem.

EXAMPLE III.

$$f_i = x_{i_1}^{a_i} e^{b_i x_{i_2}} + (x_{i_2} - c_i)$$

where

$$i_1 = i \bmod \frac{n}{2} + 1,$$

$$i_2 = i_1 + \frac{n}{2},$$

$$a_i = i \operatorname{div} p + 1,$$

$$b_i = i \operatorname{div} q + 1,$$

$$c_i = i \bmod (p + q),$$

and p and q are integer constants. We use $m = 60$, $n = 12$, $p = 15$, $q = 20$ for this problem. The Jacobian has the same structure as in Example II, but this is a large-residual problem.

Two different forcing sequences were tried:

(a) $\eta_k \equiv \eta_0 = \frac{1}{2}$,

(b) $\eta_k = \min\{\frac{1}{2}, k^{-1}\}$, if $\lambda > 0$,

$\eta_k = \min\{\frac{1}{2}, k^{-1}, \|J^T f\|\}$, if $\lambda = 0$.

Values used for other numerical parameters are $\lambda_{\min} = 10^{-5}$, $E = 4$, $D = .4$. The choice of values for E and D can greatly affect the performance of the algorithm. Convergence can be very slow if they are either too close to 1, or too extreme. The descent ratio parameters π_1 and π_2 are set to .01 and .75 respectively.

Values used for the convergence parameters were tolg $= 10^{-5}$, tolx $= 10^{-6}$, tolf $= 10^{-6}$.

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

TABLE 1. Summary of test results.

Problem	I	I	I	I	II	II	II	II	III	III
m	21	21	101	101	60	60	500	500	60	60
n	20	20	100	100	12	12	100	100	12	12
Forcing Sequence	a	b	a	b	a	b	a	b	a	b
$F(x_0)$	.1071(+5)	.1071(+5)	.1148(+9)	.1148(+9)	.1774(+16)	.1774(+16)	.1389(+2)	.1389(+2)	.1921(+5)	.1921(+5)
$F(x^*)$	.3621	.3621	.7381(+1)	.7381(+1)	.2811(−6)	.6700(−8)	.4202(−6)	.8546(−7)	.7852(+4)	.7852(+4)
Function Evaluations	8	8	11	12	28	24	17	22	62	50
Jacobian Evaluations	7	7	10	11	27	23	16	12	37	29
Total LSQR Iterations	7	10	10	13	45	62	68	98	94	128
CPU Time (secs.)	.43	.49	1.33	1.42	2.34	3.04	14.77	17.38	4.23	4.39

https://doi.org/10.1017/S0334270000004604 Published online by Cambridge University Press

Test results are summarised in Table 1. Acceptable convergence was achieved on all problems. Problem III was also solved using a standard Levenberg-Marquardt code, which did not take account of sparsity, and required more than twice as much CPU time. In order to experimentally verify rates of convergence, we allowed problems II and III to run on after the numerical convergence criteria had been satisfied. The use of forcing sequence (b) indeed produced quadratic convergence in both versions of problem II. Linear convergence was noted in problem III when forcing sequence (b) was used, with

$$\lim_{k\to\infty} \frac{\|g(x_{k+1})\|}{\|g(x_k)\|} \approx .6.$$

In many cases, the value of $\lambda = 0$ sufficed throughout the computation. For the other problems, a significant number of function evaluations were wasted in raising λ to an acceptable level. This difficulty could be helped by making E and D variable, and dependent on the decrease ratio ρ. However, overall improvement in performance would probably not be very great.

Since the Jacobian matrices in problem I were well-conditioned, only one or two LSQR iterations were required to solve the linear subproblem at each step. For the other problems, an average of only three or four LSQR iterations were required, except for the last few steps of the 100-variable version of problem II, when the use of forcing sequence (b) caused η_k to become very small.

It can be seen from the results that forcing sequence (a) tends to need more function and Jacobian evaluations, and fewer LSQR iterations, than sequence (b). Since the evaluations were relatively cheap for our test problems, sequence (a) produces better results than sequence (b). In real-life applications, however, we would expect the reverse to be the case. In problems where function and Jacobian evaluations dominate the computation, a trust region approach could be used, although this would probably mean using LSQR more than once on each step.

Further testing will include application of the program to the earthquake inversion problem mentioned in the introduction.

Acknowledgement

The motivation for this work arose as a result of a period spent by J. N. Holt at the Computer Science Department, Stanford University. We wish to express our thanks to Gene Golub, and also to Willie Lee of the U. S. Geological survey. We also wish to thank one of the referees for some very valuable comments.

References

[1] M. Altman, "Iterative methods of contractor directions", *Nonlinear Anal.* 4 (1980), 761–772.

[2] A. Bjorck and I. S. Duff, "A direct method for the solution of sparse linear least squares problems", *Linear Algebra Appl.* 34 (1980), 43–67.

[3] R. Dembo, S. Eisenstat and T. Steihaug, "Inexact Newton methods", *SIAM J. Numer. Anal.* 19 (1982), 400–408.

[4] R. Dembo and T. Steihaug, "Truncated Newton algorithms for large-scale unconstrained optimization", *Math Programming* 26 (1983), 190–212.

[5] J. Dennis, D. Gay and R. Welsch, "An adaptive nonlinear least-squares algorithm", *ACM Trans. Math. Software* 7 (1981), 348–368.

[6] P. Gill, W. Murray and S. Nash, (to appear).

[7] G. Golub and W. Kahan, "Calculating the singular values and pseudoinverse of a matrix", *SIAM J. Numer. Anal.* 2 (1965), 205–224.

[8] G. Golub and R. Plemmons, "Large-scale geodetic least squares adjustment by dissection and orthogonal decomposition", *Linear Algebra Appl.* 34 (1980), 3–27.

[9] K. Hiebert, "An evaluation of mathematical software that solves nonlinear least squares problems", *ACM Trans. Math Software* 7 (1981), 1–16.

[10] J. N. Holt and R. Fletcher, "An algorithm for constrained nonlinear least squares", *J. Inst. Math. Appl.* 23 (1979), 449–463.

[11] W. Lee and S. Stewart, *Principles and applications of microearthquake networks* (Academic Press, New York, 1981).

[12] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares", *Quart. Appl. Math.* 2 (1944), 164–168.

[13] D. Marquardt, "An algorithm for least squares estimation of nonlinear parameters", *SIAM J. Appl. Math.* 11 (1963), 431–441.

[14] J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory", in *Numerical analysis* (ed. G. A. Watson), *Lecture Notes in Math.* 630 (Springer-Verlag, New York, 1977), 105–116.

[15] M. Osborne, "Nonlinear least squares—the Levenberg-Marquardt algorithm revisited", *J. Austral. Math. Soc. Ser. B* 19 (1976), 343–357.

[16] C. Paige and M. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares", *ACM Trans. Math. Software* 8 (1982), 43–71.

[17] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization", *SIAM J. Numer. Anal.* 20 (1983), 626–637.

[18] S. Wright and J. Holt, "Algorithms for nonlinear least squares with linear inequality constraints", *SIAM J. Sci. Statist. Comput.* (to appear).