CONF-770636--1

THE LEVENBERG-MARQUARDT ALGORITHM: IMPLEMENTATION AND THEORY

Jorge J. More

NOTICE
This report was prepared as an account of work sponsored by the United States Government Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights

Prepared for

Conference on Numerical Analysis
University of Dundee, Scotland
June 28- July 1, 1977

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

U of C-AUA-USERDA

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

operated under contract W-31-109-Eng-38 for the
U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

The facilities of Argonne National Laboratory are owned by the United States Government Under the terms of a contract (W-31-109-Eng-38) between the U S Energy Research and Development Administration, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona
Carnegie-Mellon University
Case Western Reserve University
The University of Chicago
University of Cincinnati
Illinois Institute of Technology
University of Illinois
Indiana University
Iowa State University
The University of Iowa
Kansas State University
The University of Kansas
Loyola University
Marquette University
Michigan State University
The University of Michigan
University of Minnesota
University of Missouri
Northwestern University
University of Notre Dame
The Ohio State University
Ohio University
The Pennsylvania State University
Purdue University
Saint Louis University
Southern Illinois University
The University of Texas at Austin
Washington University
Wayne State University
The University of Wisconsin

NOTICE

This report was prepared as an account of work sponsored by the United States Government Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights Mention of commercial products, their manufacturers, or their suppliers in this publication does not imply or connote approval or disapproval of the product by Argonne National Laboratory or the U S Energy Research and Development Administration

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

THE LEVENBERG-MARQUARDT ALGORITHM: IMPLEMENTATION AND THEORY*

Jorge J. Moré

1. Introduction

Let $F: R^n \to R^m$ be continuously differentiable, and consider the nonlinear least squares problem of finding a local minimizer of

$$\Phi(x) = \frac{1}{2} \sum_{i=1}^{m} f_i^2(x) = \frac{1}{2} \|F(x)\|^2 . \tag{1.1}$$

Levenberg [1944] and Marquardt [1963] proposed a very elegant algorithm for the numerical solution of (1.1). However, most implementations are either not robust, or do not have a solid theoretical justification. In this work we discuss a robust and efficient implementation of a version of the Levenberg-Marquardt algorithm, and show that it has strong convergence properties. In addition to robustness, the main features of this implementation are the proper use of implicitly scaled variables, and the choice of the Levenberg-Marquardt parameter via a scheme due to Hebden [1973]. Numerical results illustrating the behavior of this implementation are also presented.

Notation. In all cases $\|\cdot\|$ refers to the ℓ_2 vector norm or to the induced operator norm. The Jacobian matrix of F evaluated at x is denoted by $F'(x)$, but if we have a sequence of vectors $\{x_k\}$, then J_k and f_k are used instead of $F'(x_k)$ and $F(x_k)$, respectively.

2. Derivation

The easiest way to derive the Levenberg-Marquardt algorithm is by a linearization argument. If, given $x \in R^n$, we could minimize

$$\Psi(p) = \|F(x+p)\|$$

as a function of p, then x+p would be the desired solution. Since Ψ is usually a nonlinear function of p, we linearize F(x+p) and obtain the linear least squares problem

$$\psi(p) = \|F(x) + F'(x)p\| .$$

Of course, this linearization is not valid for all values of p, and thus we consider the constrained linear least squares problem

*Work performed under the auspices of the U.S. Energy Research and Development Administration

(2.1) $$\min\{\psi(p): \|Dp\| \le \Delta\} .$$

In theory D is any given nonsingular matrix, but in our implementation D is a diagonal matrix which takes into account the scaling of the problem. In either case, p lies in the hyperellipsoid

(2.2) $$E = \{p: \|Dp\| \le \Delta\} ,$$

but if D is diagonal, then E has axes along the coordinate directions and the length of the ith semi-axis is Δ/d_i.

We now consider the solution of (2.1) in some generality, and thus the problem

(2.3) $$\min\{\|f+Jp\|: \|Dp\| \le \Delta\}$$

where $f \in R^m$ and J is any m by n matrix. The basis for the Levenberg-Marquardt method is the result that if p^* is a solution to (2.3), then $p^* = p(\lambda)$ for some $\lambda \ge 0$ where

(2.4) $$p(\lambda) = -(J^TJ + \lambda D^TD)^{-1}J^Tf .$$

If J is rank deficient and $\lambda = 0$, then (2.4) is defined by the limiting process

$$Dp(0) \equiv \lim_{\lambda\to 0^+} Dp(\lambda) = -(JD^{-1})^{\dagger}f .$$

There are two possibilities: Either $\lambda = 0$ and $\|Dp(0)\| \le \Delta$, in which case p(0) is the solution to (2.3) for which $\|Dp\|$ is least, or $\lambda > 0$ and $\|Dp(\lambda)\| = \Delta$, and then $p(\lambda)$ is the unique solution to (2.3).

The above results suggest the following iteration.

(2.5) <u>Algorithm</u>

(a) Given $\Delta_k > 0$, find $\lambda_k \ge 0$ such that if

$$(J_k{}^TJ_k + \lambda_kD_k{}^TD_k)p_k = -J_k{}^Tf_k ,$$

then either $\lambda_k = 0$ and $\|D_kp_k\| \le \Delta_k$, or $\lambda_k > 0$ and $\|D_kp_k\| = \Delta_k$.

(b) If $\|F(x_k+p_k)\| < \|F(x_k)\|$ set $x_{k+1} = x_k+p_k$ and evaluate J_{k+1}; otherwise set $x_{k+1} = x_k$ and $J_{k+1} = J_k$.

(c) Choose Δ_{k+1} and D_{k+1}.

In the next four sections we elaborate on how (2.5) leads to a very robust and efficient implementation of the Levenberg-Marquardt algorithm.

3. Solution of a Structured Linear Least Squares Problem

The simplest way to obtain the correction p is to use Cholesky decomposition on the linear system

$$(3.1) \qquad (J^TJ + \lambda D^TD)p = -J^Tf \ .$$

Another method is to recognize that (3.1) are the normal equations for the least squares problem

$$(3.2) \qquad \begin{pmatrix} J \\ \lambda^{\frac{1}{2}}D \end{pmatrix} p \cong -\begin{pmatrix} f \\ 0 \end{pmatrix} ,$$

and to solve this structured least squares problem using QR decomposition with column pivoting.

The main advantage of the normal equations is speed; it is possible to solve (3.1) twice as fast as (3.2). On the other hand, the normal equations are particularly unreliable when $\lambda = 0$ and J is nearly rank deficient. Moreover, the formation of J^TJ or D^TD can lead to unnecessary underflows and overflows, while this is not the case with (3.2). We feel that the loss in speed is more than made up by the gain in reliability and robustness.

The least squares solution of (3.2) proceeds in two stages. These stages are the same as those suggested by Golub (Osborne [1972]), but modified to take into account the pivoting.

In the first stage, compute the QR decomposition of J with column pivoting. This produces an orthogonal matrix Q and a permutation π of the columns of J such that

$$(3.3) \qquad QJ\pi = \begin{pmatrix} T & S \\ 0 & 0 \end{pmatrix}$$

where T is a nonsingular upper triangular matrix of rank (J) order. If $\lambda = 0$, then a solution of (3.2) is

$$p = \pi \begin{pmatrix} T^{-1} & 0 \\ 0 & 0 \end{pmatrix} Qf \equiv J^{-}f$$

where J^{-} refers to a particular symmetric generalized inverse of J in the sense that JJ^{-} is symmetric and $JJ^{-}J = J$. To solve (3.2) when $\lambda > 0$ first note that (3.3) implies that

$$(3.4) \qquad \begin{pmatrix} Q & 0 \\ 0 & \pi^T \end{pmatrix} \begin{pmatrix} J \\ \lambda^{\frac{1}{2}}D \end{pmatrix} \pi = \begin{pmatrix} R \\ 0 \\ D_\lambda \end{pmatrix}$$

where $D_\lambda = \lambda^{\frac{1}{2}}\pi^TD\pi$ is still a diagonal matrix and R is a (possibly singular) upper triangular matrix of order n.

In the second stage, compute the QR decomposition of the matrix on the right of (3.4). This can be done with a sequence of n(n+1)/2 Givens rotations. The result is an orthogonal matrix W such that

$$(3.5) \qquad W\begin{pmatrix} R \\ 0 \\ D_\lambda \end{pmatrix} = \begin{pmatrix} R_\lambda \\ \\ 0 \end{pmatrix}.$$

where R_λ is a nonsingular upper triangular matrix of order n. The solution to (3.2) is then

$$p = -\pi R_\lambda^{-1} u$$

where $u \in R^n$ is determined from

$$W\begin{pmatrix} Qf \\ \\ 0 \end{pmatrix} = \begin{pmatrix} u \\ \\ v \end{pmatrix}.$$

It is important to note that if λ is changed, then only the second stage must be redone.

4. Updating the Step Bound

The choice of Δ depends on the ratio between the actual reduction and the predicted reduction obtained by the correction. In our case, this ratio is given by

$$(4.1) \qquad \rho(p) = \frac{\|F(x)\|^2 - \|F(x+p)\|^2}{\|F(x)\|^2 - \|F(x)+F'(x)p\|^2}.$$

Thus (4.1) measures the agreement between the linear model and the (nonlinear) function. For example, if F is linear then $\rho(p) = 1$ for all p, and if $F'(x)^T F(x) \neq 0$, then $\rho(p) \to 1$ as $\|p\| \to 0$. Moreover, if $\|F(x+p)\| \geq \|F(x)\|$ then $\rho(p) \leq 0$.

The scheme for updating Δ has the objective of keeping the value of (4.1) at a reasonable level. Thus, if ρ(p) is close to unity (i.e. $\rho(p) \geq 3/4$), we may want to increase Δ, but if ρ(p) is not close to unity (i.e. $\rho(p) \leq 1/4$), then Δ must be decreased. Before giving more specific rules for updating Δ, we discuss the computation of (4.1). For this, write

$$(4.2) \qquad \rho = \frac{\|f\|^2 - \|f_+\|^2}{\|f\|^2 - \|f+Jp\|^2}$$

with an obvious change in notation. Since p satisfies (3.1),

$$(4.3) \qquad \|f\|^2 - \|f+Jp\|^2 = \|Jp\|^2 + 2\lambda\|Dp\|^2 ,$$

and hence we can rewrite (4.2) as

(4.4)
$$\rho = \frac{1 - \left(\frac{\|f_+\|}{\|f\|}\right)^2}{\left(\frac{\|Jp\|}{\|f\|}\right)^2 + 2\left(\frac{\lambda^{\frac{1}{2}}\|Dp\|}{\|f\|}\right)^2} .$$

Since (4.3) implies that

$$\|Jp\| \le \|f\|, \quad \lambda^{\frac{1}{2}}\|Dp\| \le \|f\| ,$$

the computation of the denominator will not generate any overflows, and moreover, the denominator will be non-negative regardless of roundoff errors. Note that this is not the case with (4.2). The numerator of (4.4) may generate overflows if $\|f_+\|$ is much larger than $\|f\|$, but since we are only interested in positive values of ρ, if $\|f_+\| > \|f\|$ we can just set $\rho = 0$ and avoid (4.4).

We now discuss how to update Δ. To increase Δ we simply multiply Δ hy a constant factor not less than one. To decrease Δ we follow Fletcher [1971] and fit a quadratic to $\delta(0)$, $\delta'(0)$ and $\delta(1)$ where

$$\delta(\theta) = \frac{1}{2}\|F(x+\theta p)\|^2 .$$

If μ is the minimizer of the resulting quadratic, we decrease Δ by multiplying Δ by μ, but if $\mu \notin \left[\frac{1}{10}, \frac{1}{2}\right]$, we replace μ by the closest endpoint. To compute μ safely, first note that (3.1) implies that

$$\gamma \equiv \frac{p^T J^T f}{\|f\|^2} = -\left[\left(\frac{\|Jp\|}{\|f\|}\right)^2 + \left(\lambda^{\frac{1}{2}} \frac{\|Dp\|}{\|f\|}\right)^2\right] ,$$

and that $\gamma \in [-1,0]$. It is now easy to verify that

(4.5)
$$\mu = \frac{\frac{1}{2}\gamma}{\gamma + \frac{1}{2}\left[1 - \left(\frac{\|f_+\|}{\|f\|}\right)^2\right]} .$$

If $\|f_+\| \le \|f\|$ we set $\mu = 1/2$. Also note that we only compute μ by (4.5) if say, $\|f_+\| \le 10\|f\|$, for otherwise, $\mu \le 1/10$.

5. The Levenberg-Marquardt Parameter

In our implementation $\alpha > 0$ is accepted as the Levenberg-Marquardt parameter if

(5.1)
$$|\phi(\alpha)| \le \sigma\Delta ,$$

where

(5.2)
$$\phi(\alpha) = \|D(J^TJ+\alpha D^TD)^{-1}J^Tf\| - \Delta ,$$

and $\sigma \in (0,1)$ specifies the desired relative error in $\|Dp(\alpha)\|$. Of course, if

$\phi(0) \leq 0$ then $\alpha = 0$ is the required parameter, so in the remainder of this section we assume that $\phi(0) > 0$. Then ϕ is a continuous, strictly decreasing function on $[0,+\infty)$ and $\phi(\alpha)$ approaches $-\Delta$ at infinity. It follows that there is a unique $\alpha^* > 0$ such that $\phi(\alpha^*) = 0$. To determine the Levenberg-Marquardt parameter we assume that an initial estimate $\alpha_0 > 0$ is available, and generate a sequence $\{\alpha_k\}$ which converges to α^*.

Since ϕ is a convex function, it is very tempting to use Newton's method to generate $\{\alpha_k\}$, but this turns out to be very inefficient -- the particular structure of this problem allows us to derive a much more efficient iteration due to Hebden [1973]. To do this, note that

$$\phi(\alpha) = \|(\tilde{J}^T\tilde{J}+\alpha I)^{-1}\tilde{J}^T f\| - \Delta, \quad \tilde{J} = JD^{-1} , \tag{5.3}$$

and let $\tilde{J} = U\Sigma V^T$ be the singular value decomposition of $\tilde{J}$. Then

$$\phi(\alpha) = \left(\sum_{i=1}^{n} \frac{\sigma_i^2 z_i^2}{(\sigma_i^2+\alpha)^2}\right)^{\frac{1}{2}} - \Delta$$

where $z = U^T f$ and $\sigma_1,\ldots,\sigma_n$ are the singular values of $\tilde{J}$. Hence, it is very natural to assume that

$$\phi(\alpha) \doteq \frac{a}{b+\alpha} - \Delta \equiv \tilde{\phi}(\alpha) ,$$

and to choose a and b so that $\tilde{\phi}(\alpha_k) = \phi(\alpha_k)$ and $\tilde{\phi}'(\alpha_k) = \phi'(\alpha_k)$. Then $\tilde{\phi}(\alpha_{k+1}) = 0$ if

$$\alpha_{k+1} = \alpha_k - \left(\frac{\phi(\alpha_k) + \Delta}{\Delta}\right)\left[\frac{\phi(\alpha_k)}{\phi'(\alpha_k)}\right] . \tag{5.4}$$

This iterative scheme must be safeguarded if it is to converge. Hebden [1973] proposed using upper and lower bounds u_k and ℓ_k, and that (5.4) be applied with the restriction that no iterate may be within $(u_k-\ell_k)/10$ of either endpoint. It turns out that this restriction is very detrimental to the progress of the iteration since in a lot of cases u_k is much larger than ℓ_k. A much more efficient algorithm can be obtained if (5.4) is only modified when α_{k+1} is outside of (ℓ_{k+1},u_{k+1}). To specify this algorithm we first follow Hebden [1973] and note that (5.3) implies that

$$u_0 = \frac{\|(JD^{-1})^T f\|}{\Delta}$$

is a suitable upper bound. If J is not rank deficient, then $\phi'(0)$ is defined and the convexity of ϕ implies that

$$\ell_0 = -\frac{\phi(0)}{\phi'(0)}$$

is a lower bound; otherwise let $\ell_0 = 0$.

(5.5) *Algorithm*

(a) If $\alpha_k \notin (\ell_k, u_k)$ let $\alpha_k = \max\{0.001\, u_k, (\ell_k u_k)^{\frac{1}{2}}\}$.

(b) Evaluate $\phi(\alpha_k)$ and $\phi'(\alpha_k)$. Update u_k by letting $u_{k+1} = \alpha_k$ if $\phi(\alpha_k) < 0$ and $u_{k+1} = u_k$ otherwise. Update ℓ_k by

$$\ell_{k+1} = \max\left\{\ell_k,\ \alpha_k - \frac{\phi(\alpha_k)}{\phi'(\alpha_k)}\right\} .$$

(c) Obtain α_{k+1} from (5.4).

The role of (5.5)(a) is to replace α_k by a point in (ℓ_k, u_k) which is biased towards ℓ_k; the factor $0.001\, u_k$ was added to guard against exceedingly small values of ℓ_k, and in particular, $\ell_k = 0$. In (5.5)(b), the convexity of ϕ guarantees that the Newton iterate can be used to update ℓ_k.

It is not too difficult to show that algorithm (5.5) always generates a sequence which converges quadratically to α^*. In practice, less than two iterations (on the average) are required to satisfy (5.1) when $\sigma = 0.1$.

To complete the discussion of the Hebden algorithm, we show how to evaluate $\phi'(\alpha)$. From (5.2) it follows that

$$\phi'(\alpha) = -\frac{(D^T q(\alpha))^T (J^T J + \alpha D^T D)^{-1} (D^T q(\alpha))}{\| q(\alpha) \|}$$

where $q(\alpha) = Dp(\alpha)$ and $p(\cdot)$ is defined by (2.4). From (3.4) and (3.5) we have

$$\pi^T (J^T J + \alpha D^T D) \pi = R_\alpha^{\ T} R_\alpha ,$$

and hence,

$$\phi'(\alpha) = - \| q(\alpha) \| \left\| R_\alpha^{-T} \left(\frac{\pi^T D^T q(\alpha)}{\| q(\alpha) \|} \right) \right\|^2 .$$

6. Scaling

Since the purpose of the matrix D_k in the Levenberg-Marquardt algorithm is to take into account the scaling of the problem, some authors (e.g. Fletcher [1971]) choose

$$D_k = \text{diag}(d_1^{(k)}, \ldots, d_n^{(k)}) \tag{6.1}$$

where

$$d_i^{(k)} = \| \partial_i F(x_0) \|, \quad k \geq 0 . \tag{6.2}$$

This choice is usually adequate as long as $\| \partial_i F(x_k) \|$ does not increase with k. However, if $\| \partial_i F(x_k) \|$ increases, this requires a decrease in the length ($= \Delta / d_i$) of the i^{th} semi-axis of the hyperellipsoid (2.2), since F is now changing faster along the

i^{th} variable, and therefore, steps which have a large i^{th} component tend to be unreliable. This argument leads to the choice

(6.3)
$$d_i^{(0)} = \|\partial_i F(x_0)\|$$
$$d_i^{(k)} = \max\left\{d_i^{(k-1)}, \|\partial_i F(x_k)\|\right\} \quad , \quad k \geq 1 \; .$$

Note that a decrease in $\|\partial_i F(x_k)\|$ only implies that F is not changing as fast along the i^{th} variable, and hence does not require a decrease in d_i. In fact, the choice

(6.4)
$$d_i^{(k)} = \|\partial_i F(x_k)\| \quad , \quad k \geq 0 \; ,$$

is computationally inferior to both (6.2) and (6.3). Moreover, our theoretical results support choice (6.3) over (6.4), and to a lesser extent, (6.2).

It is interesting to note that (6.2), (6.3), and (6.4) make the Levenberg-Marquardt algorithm scale invariant. In other words, for all of the above choices, if D is a diagonal matrix with positive diagonal elements, then algorithm (2.5) generates the same iterates if either it is applied to F and started at x_0, or if it is applied to $\tilde{F}(x) = F(D^{-1}x)$ and started at $\tilde{x}_0 = Dx_0$. For this result it is assumed that the decision to change Δ is only based on (4.1), and thus is also scale invariant.

7. Theoretical Results

It will be sufficient to present a convergence result for the following version of the Levenberg-Marquardt algorithm.

(7.1) Algorithm

(a) Let $\sigma \in (0,1)$. If $\|D_k J_k^- f_k\| \leq (1+\sigma)\Delta_k$, set $\lambda_k = 0$ and $p_k = -J_k^- f_k$. Otherwise determine $\lambda_k > 0$ such that if

$$\begin{pmatrix} J_k \\ \lambda_k^{\frac{1}{2}} D_k \end{pmatrix} p_k \cong - \begin{pmatrix} f_k \\ 0 \end{pmatrix}$$

then

$$(1-\sigma)\Delta_k \leq \|D_k p_k\| \leq (1+\sigma)\Delta_k \; .$$

(b) Compute the ratio ρ_k of actual to predicted reduction.

(c) If $\rho_k \leq 0.0001$, set $x_{k+1} = x_k$ and $J_{k+1} = J_k$.
If $\rho_k > 0.0001$, set $x_{k+1} = x_k + p_k$ and compute J_{k+1}.

(d) If $\rho_k \leq 1/4$, set $\Delta_{k+1} \in \left[\frac{1}{10}\Delta_k, \frac{1}{2}\Delta_k\right]$.
If $\rho_k \in \left(\frac{1}{4}, \frac{3}{4}\right)$ and $\lambda_k = 0$, or if $\rho_k \geq 3/4$, set $\Delta_{k+1} = 2\|D_k p_k\|$.

(e) Update D_{k+1} by (6.1) and (6.3).

The proof of our convergence result is somewhat long and will therefore be presented elsewhere.

<u>Theorem</u>. Let $F: R^n \to R^m$ be continuously differentiable on R^n, and let $\{x_k\}$ be the sequence generated by algorithm (7.1). Then

$$\liminf_{k \to +\infty} \|(J_k D_k^{-1})^T f_k\| = 0 \ . \tag{7.2}$$

This result guarantees that eventually a <u>scaled</u> gradient will be small enough. Of course, if $\{J_k\}$ is bounded then (7.2) implies the more standard result that

$$\liminf_{k \to +\infty} \|J_k^T f_k\| = 0 \ . \tag{7.3}$$

Furthermore, we can also show that if F' is uniformly continuous then

$$\lim_{k \to +\infty} \|J_k^T f_k\| = 0 \ . \tag{7.4}$$

Powell [1975] and Osborne [1975] have also obtained global convergence results for their versions of the Levenberg-Marquardt algorithm. Powell presented a general algorithm for unconstrained minimization which as a special case contains (7.1) with $\sigma = 0$ and $\{D_k\}$ constant. For this case Powell obtains (7.3) under the assumption that $\{J_k\}$ is bounded. Osborne's algorithm directly controls $\{\lambda_k\}$ instead of $\{\Delta_k\}$, and allows $\{D_k\}$ to be chosen by (6.1) and (6.3). For this case he proves (7.4) under the assumptions that $\{J_k\}$ and $\{\lambda_k\}$ are bounded.

8. <u>Numerical Results</u>

In our numerical results we would like to illustrate the behavior of our algorithm with the three choices of scaling mentioned in Section 6. For this purpose, we have chosen four functions.

1) <u>Fletcher and Powell</u> [1963] n=3, m=3

$$f_1(x) = 10[x_3 - 10\theta(x_1,x_2)]$$

$$f_2(x) = 10[(x_1^2+x_2^2)^{\frac{1}{2}} - 1]$$

$$f_3(x) = x_3$$

where

$$\theta(x_1,x_2) = \begin{cases} \frac{1}{2\pi} \arctan\ (x_2/x_1), & x_1 > 0 \\ \frac{1}{2\pi} \arctan\ (x_2/x_1) + 0.5, & x_1 < 0 \end{cases}$$

$$x_0 = (-1,0,0)^T$$

2. Kowalik and Osborne [1968]   n=4, m=11

$$f_i(x) = y_i - \frac{x_1[u_i^2 + x_2u_i]}{(u_i^2 + x_3u_i + x_4)}$$

where u_i and y_i are specified in the original paper.

$$x_0 = (0.25, 0.39, 0.415, 0.39)^T$$

3. Bard [1970]   n=3, m=15

$$f_i(x) = y_i - \left[x_1 + \frac{u_i}{x_2v_i + x_3w_i}\right]$$

where $u_i = i$, $v_i = 16-i$, $w_i = \min\{u_i,v_i\}$, and y_i is specified in the original paper.

$$x_0 = (1,1,1)^T$$

4. Brown and Dennis [1971]   n=4, m=20

$$f_i(x) = [x_1 + x_2t_i - \exp(t_i)]^2 + [x_3 + x_4\sin(t_i) - \cos(t_i)]^2$$

where $t_i = (0.2)i$.

$$x_0 = (25, 5, -5, 1)^T$$

These problems have very interesting features. Problem 1 is a helix with a zero residual at $x^* = (1,0,0)$ and a discontinuity along the plane $x_1 = 0$; note that the algorithm must cross this plane to reach the solution. Problems 2 and 3 are data fitting problems with small residuals, while Problem 4 has a large residual. The residuals are given below.

1. $\|F(x^*)\| = 0.0$
2. $\|F(x^*)\| = 0.0175358$
3. $\|F(x^*)\| = 0.0906359$
4. $\|F(x^*)\| = 292.9542$

Problems 2 and 3 have other solutions. To see this, note that for Kowalik and Osborne's function,

$$(8.1) \qquad \lim_{\alpha\to\infty} f_i(\alpha,x_2,\alpha,\alpha) = y_i - \left(\frac{u_i}{u_i+1}\right)(x_2+u_i) \ ,$$

while for Bard's function,

$$(8.2) \qquad \lim_{\alpha\to\infty} f_i(x_1,\alpha,\alpha) = y_i - x_1 \ .$$

These are now linear least squares problems, and as such, the parameter x_2 in (8.1) and x_1 in (8.2) are completely determined. However, the remaining parameters only need to be sufficiently large.

In presenting numerical results one must be very careful about the convergence criteria used. This is particularly true of the Levenberg-Marquardt method since, unless $F(x^*) = 0$, the algorithm converges linearly. In our implementation, an approximation x to x^* is acceptable if either x is close to x^* or $\|F(x)\|$ is close to $\|F(x^*)\|$. We attempt to satisfy these criteria by the convergence tests

$$(8.3) \qquad \Delta \le \mathrm{XTOL}\ \|Dx\| ,$$

and

$$(8.4) \qquad \left(\frac{\|Jp\|}{\|f\|}\right)^2 + 2\left(\lambda^{\frac{1}{2}} \frac{\|Dp\|}{\|f\|}\right)^2 \le \mathrm{FTOL} .$$

An important aspect of these tests is that they are scale invariant in the sense of Section 6. Also note that the work of Section 4 shows that (8.4) is just the relative error between $\|f+Jp\|^2$ and $\|f\|^2$.

The problems were run on the IBM 370/195 of Argonne National Laboratory in double precision (14 hexadecimal digits) and under the FORTRAN H (opt=2) compiler. The tolerances in (8.3) and (8.4) were set at FTOL = 10^{-8} and XTOL = 10^{-8}. Each problem is run with three starting vectors. We have already given the starting vector x_0 which is closest to the solution; the other two points are $10x_0$ and $100x_0$. For each starting vector, we have tried our algorithm with the three choices of $\{D_k\}$. In the table below, choices (6.2), (6.3) and (6.4) are referred to as initial, adaptive, and continuous scaling, respectively. Moreover, NF and NJ stands for the number of function and Jacobian evaluations required for convergence.

		x_0		$10x_0$		$100x_0$	
PROBLEM	SCALING	NF	NJ	NF	NJ	NF	NJ
1	Initial	12	9	34	29	FC	FC
	Adaptive	11	8	20	15	19	16
	Continuous	12	9	14	12	176	141
2	Initial	19	17	81	71	365	315
	Adaptive	18	16	79	71	348	307
	Continuous	18	16	63	54	FC	FC
3	Initial	8	7	37	36	14	13
	Adaptive	8	7	37	36	14	13
	Continuous	8	7	FC	FC	FC	FC
4	Initial	268	242	423	400	FC	FC
	Adaptive	268	242	57	47	229	207
	Continuous	FC	FC	FC	FC	FC	FC

Interestingly enough, convergence to the minimizer indicated by (8.1) only occurred for starting vector $10x_0$ of Problem 2, while for Problem 3 starting vectors $10x_0$ and $100x_0$ led to (8.2). Otherwise, either the global minimizer was obtained, or the algorithm failed to converge to a solution; the latter is indicated by FC in the table.

It is clear from the table that the adaptive strategy is best in these four examples. We have run other problems, but in all other cases the difference is not as dramatic as in these cases. However, we believe that the above examples adequately justify our choice of scaling matrix.

Acknowledgments. This work benefited from interaction with several people. Beverly Arnoldy provided the numerical results for several versions of the Levenberg-Marquardt algorithm, Brian Smith showed how to use pivoting in the two-stage process of Section 3, and Danny Sorensen made many valuable comments on an earlier draft of this paper. Finally, I would like to thank Judy Beumer for her swift and beautiful typing of the paper.

References

1. Bard, Y. [1970]. Comparison of gradient methods for the solution of nonlinear parameter estimation problem, SIAM J. Numer. Anal. 7, 157-186.

2. Brown, K. M. and Dennis, J. E. [1971]. New computational algorithms for minimizing a sum of squares of nonlinear functions, Department of Computer Science report 71-6, Yale University, New Haven, Connecticut.

3. Fletcher, R. [1971]. A modified Marquardt subroutine for nonlinear least squares, Atomic Energy Research Establishment report R6799, Harwell, England.

4. Fletcher, R. and Powell, M.J.D. [1963]. A rapidly convergent descent method for minimization, Comput. J. 6, 163-168.

5. Hebden, M. D. [1973]. An algorithm for minimization using exact second derivatives, Atomic Energy Research Establishment report TP515, Harwell, England.

6. Kowalik, J. and Osborne, M. R. [1968]. Methods for Unconstrained Optimization Problems, American Elsevier.

7. Levenberg, K. [1944]. A method for the solution of certain nonlinear problems in least squares, Quart. Appl. Math. 2, 164-168.

8. Marquardt, D. W. [1963]. An algorithm for least squares estimation of nonlinear parameters, SIAM J. Appl. Math. 11, 431-441.

9. Osborne, M. R. [1972]. Some aspects of nonlinear least squares calculations, in Numerical Methods for Nonlinear Optimization, F. A. Lootsma, ed., Academic Press.

10. Osborne, M. R. [1975]. Nonlinear least squares - the Levenberg algorithm revisited, to appear in Series B of the Journal of the Australian Mathematical Society.

11. Powell, M. J. D. [1975]. Convergence properties of a class of minimization algorithms, in Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press.