International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

An Adaptive Method for Minimizing a Sum of Squares of Nonlinear Functions

Nazareth, J.L.

IIASA Working Paper

WP-83-099

October 1983

Working Paper

AN ADAPTIVE METHOD FOR MINIMIZING A SUM OF SQUARES OF NONLINEAR FUNCTIONS*

Larry Nazareth

October 1983
WP-83-99

International Institute for Applied Systems Analysis
A-2361 Laxenburg, Austria

AN ADAPTIVE METHOD FOR MINIMIZING A SUM OF SQUARES OF NONLINEAR FUNCTIONS*

Larry Nazareth

October 1983
WP-83-99

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

PREFACE

The work discussed in this paper was carried out within the Adaptation and Optimization Project at IIASA. The author describes and evaluates the computational effectiveness of an adaptive nonlinear least squares method which was developed, in particular, for parameter estimation when the residual sum of squares at the optimum solution is large.

Andrzej Wierzbicki
Chairman
System and Decision Sciences Area

ABSTRACT

The Gauss-Newton and the Levenberg-Marquardt algorithms for solving nonlinear least squares problems, minimize $F(x) = \sum_{i=1}^{m} (f_i(x))^2$ for $x \in R^n$, are both based upon the premise that one term in the Hessian of $F(x)$ dominates its other terms, and that the Hessian may be approximated by this dominant term $J^T J$, where $J_{ij} = (\delta f_i / \delta x_j)$. We are motivated here by the need for an algorithm which works well when applied to problems for which this premise is substantially violated, and is yet able to take advantage of situations where the premise holds. We describe and justify a method for approximating the Hessian of $F(x)$ which uses a convex combination of $J^T J$ and a matrix obtained by making quasi-Newton updates. In order to evaluate the usefulness of this idea, we construct a nonlinear least squares algorithm which uses this Hessian approximation, and report test results obtained by applying it to a set of test problems. A merit of our approach is that it demonstrates how a single adaptive algorithm can be used to efficiently solve unconstrained nonlinear optimization problems (whose Hessians have no particular structure), small residual and large residual, nonlinear least squares problems. Our paper can also be looked upon as an investigation for one problem area, of the following more general question: how can one combine two different Hessian approximations (or model functions) which are simultaneously available? The technique suggested here may thus be more widely applicable and may be of use, for example, when minimizing functions which are only partly composed of sums of squares arising in penalty function methods.

AN ADAPTIVE METHOD FOR MINIMIZING A SUM OF SQUARES OF NONLINEAR FUNCTIONS*

Larry Nazareth

1. Introduction

We are concerned here with solving the problem

$$\underset{x \in R^n}{\text{minimize}}\ F(x) = \sum_{i=1}^{m} (f_i(x))^2 \tag{1.1}$$

The gradient of $F(x)$ is $J^T f$, where J is the $m \times n$ Jacobian matrix $J_{ij} = (\delta f_i / \delta x_j)$ at the point x, and f is the m-vector of function values $(f_1,\ldots,f_m)^T$. The $n \times n$ Hessian matrix H of the function $F(x)$ has the special form

$$H = J^T J + \sum_{i=1}^{m} f_i H_i \tag{1.2}$$

where H_i is the $n \times n$ Hessian matrix of $f_i(x)$.

---

*This paper is based upon research begun at the Argonne National Laboratory, Illinois, USA. A renewed interest in methods of this type at the 1982 Mathematical Programming Symposium held in Bonn, led us to prepare this substantially revised version of an earlier unpublished report.

Most methods for solving (1.1) are a specialized version of Newton method, and differ from one another primarily in the way in which H is approximated. The most popular of these methods are briefly summarized in Table 1.

There are numerous practical examples for which the term $\sum_{i=1}^{m} f_i H_i$ cannot be neglected in favour of $J^T J$. McKeown [14] has provided several examples where quasi-Newton optimizers which do not take account of the special structure of the Hessian (1.2) perform better than specialized routines like Gauss-Newton or Levenberg-Marquardt. The method which we propose here is motivated by the need to solve such problems efficiently and is based upon an alternative method for approximating the Hessian of $F(x)$. This is done by taking a convex combination of $J^T J$ and a matrix obtained by making quasi-Newton updates. In order to evaluate the usefulness of this idea, we construct a simple adaptive algorithm which uses this Hessian approximation, and report results obtained by applying it to a test problems. A merit of our approach is that it demonstrates how a single algorithm can be used to efficiently solve general unconstrained optimization problems, small residual and large residual nonlinear least squares problems. The technique for combining two Hessian approximations may also be applicable to other situations for which more than one model function is available.

2. The Hybrid Method for Approximating the Hessian

2.1 We approximate the Hessian by taking a convex combination of $J^T J$ and a matrix B^{QN} obtained by quasi-Newton updates. Thus

$$B^H = \alpha J^T J + (1 - \alpha) B^{QN} \tag{2.1}$$

where α is a real number such that $0 \leq \alpha \leq 1$. The idea of using a convex combination of updates has been employed elsewhere, for example, see Fletcher

Table 1.

	Hessian approx. B, search dirn. d	Method	Comments
1.	$B=I$	Gradient Method	Robust but slow
2.	$B=J^TJ$	Gauss-Newton (GN)	Rapid convergence for zero-residual problems. Not robust.
3.	$d=-J^+f$ where J^+ is the generalized inverse of J.	Fletcher [2],[1]	Useful when J is not of full column rank since d is still a direction of descent.
4.	$B=J^TJ+\lambda D, \lambda \geq 0, D$ is positive and diagonal. d is defined by solving $Bd=-J^Tf$.	Levenberg-Marquardt [3], [4], [13]	Approx. min. GN model function in a region of trust. d lies on arc A of Figure 1.
5.	B obtained by quasi-Newton updates	e.g. BFGS [12], Davidon [16]	Does not take account of special structure in Hessian.
6.	$B=J^TJ+\sum f_iB_i$ where B_i is a quasi-Newton approx. to H_i	Brown-Dennis [8]	Very expensive in terms of storage.
7.	$B=J^TJ+S$ where S is chosen so that B satisfies a quasi-Newton relation	Broyden-Dennis [11], Betts [21], Dennis et al [22]	B can become indefinite so that search dirns. may not be descent dirns. unless S is sized.
8.	d is chosen suitably in subspace spanned by negative gradient and GN or LM directions	Powell VA05A [6], Steen-Byrne [7]	For the latter, search dirns. lie on arc B of Figure 1.
9.	Augmenting GN direction in certain subspaces	Gill & Murray [17]	When J is not close to rank deficiency, method is essentially the GN method

[24] and it is a simple and natural choice when combining the Hessians of two different model functions, since a bias toward one model simultaneously reduces the contribution of the other. We also think that it is preferable to first construct a model function and then use it to determine a search direc-

tion, rather than to form search vectors from two different model functions determined by J^TJ and B^{QN} and then choose a search direction in the subspace that such vectors define, for example, by taking a convex combination.

B^H has the following properties:

a) B^H is positive semidefinite ($B^H \geq 0$) whenever $B^{QN} \geq 0$.

b) If J_0 denotes the Jacobian matrix at the initial point and B_0^{QN} is set to $J_0^TJ_0$, an algorithm based upon (2.1) converges in one step when applied to a problem (1.1) for which each f_i is linear. Furthermore, for zero residual problems, $B^{QN} \rightarrow J_m^TJ_m$, here J_m is the Jacobian matrix at the optimum and we assume that the search spans the full n-dimensional space. Thus $B^H \rightarrow J_m^TJ_m$, and this leads us to expect that an algorithm based upon (2.1) shares the rapid convergence properties of the GN method on zero residual problems.

c) Let Δx be the step just taken, so that $B^{QN}\Delta x = \Delta g$, where Δg is the change of gradient for the steps Δx.

Then

$$\begin{aligned} B^H\Delta x &= \alpha J^TJ\Delta x + (1-\alpha)B^{QN}\Delta x \\ &= \Delta g + \alpha(J^TJ\Delta x - \Delta g) \end{aligned} \tag{2.2}$$

B^H does not satisfy the quasi-Newton relation. However, when $J^TJ\Delta x$ approximately equals Δg, then $B^H\Delta x$ approximately equals Δg. Further, through α we have explicit control over the extent to which the quasi-Newton relation is violated, this being in turn determined by the extent to which known information J^TJ can be trusted as being a reasonable approximation to the Hessian.

d) If we write $B^{QN} = J^TJ + M$, then (2.1) becomes

$$B^H = J^TJ + (1-\alpha)M \tag{2.3}$$

and thus $(1-\alpha)M$ is implicitly an approximation to $\sum_{i=1}^{m} f_iH_i$. The method of

Dennis et al [22], (see also [15]), forms an explicit approximation S to $\sum_{i=1}^{m} f_i H_i$. The approximation to the Hessian is therefore $J^T J + S$, and when this is used in place of B^{QN} in (2.1) we obtain $B^D = J^T J + (1 - \alpha)S$. $(1 - \alpha)$ acts as a sizing factor and its use was found to be quite central to the success of the method of Dennis et al [22]. In particular, we can ensure that $B^D \geq 0$, by choosing α suitably. This is an example of how the idea of combining Hessian approximations may be more widely applicable.

2.2. Reasons which motivate the Levenberg-Marquardt (LM) extension of the Gauss-Newton (GN) Method apply equally well to the Hessian approximation (2.1). Perhaps the most convincing argument for the LM method is that it associates a region of trust with the GN model function and, through suitable choice of a parameter $\lambda \geq 0$, seeks an approximate minimum of the model function within the region of trust. See, for example, [13]. Solving this is equivalent to adding a term λD to $J^T J$ (where D is a positive definite diagonal matrix) and $\lambda \geq 0$ is an appropriately chosen scalar. Other related justifications for the LM approach are that we thereby improve the conditioning of $J^T J$, permit a unidimensional search in λ, and bias the associated direction of search towards the negative gradient, thus making the algorithm more likely to converge from distant starting points. We propose therefore to implement our Hessian approximation in the form

$$B = (B^H + \lambda D) = \alpha J^T J + (1 - \alpha) B^{QN} + \lambda D \tag{2.4}$$

Search directions $\boldsymbol{d}$ are defined by solving

$$B\boldsymbol{d} = -J^T f \tag{2.5}$$

and lie on the arc C of Figure 1. An algorithm which uses (2.4) can easily be specialized to the LM method by fixing α at value 1, and to a version of a quasi-Newton (QN) method by fixing α at value 0.

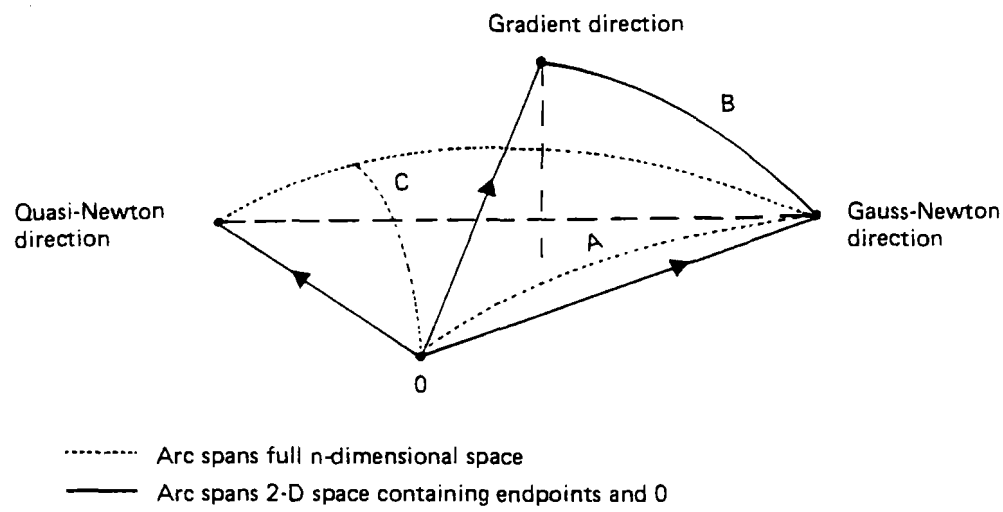

Figure 1

We can also adopt the following alternative view of (2.4) and (2.5). Suppose given the Hessian approximation B^H at a point x, with associated gradient $g = J^T f$, we make a model function

$$M(z) = g^T(z - x) + \frac{1}{2}(z - x)^T B^H (z - x) \tag{2.6}$$

and we seek to solve the problem

$$\underset{z \in R^n}{\text{minimize}}\ M(z) \tag{2.7}$$

$$\text{subject to}\quad (z - x)^T D(z - x) \leq \Delta$$

where $\Delta > 0$ and fixed. By forming the Lagrangian, it is easily seen that the solution to (2.7), say x_*, is given by

$$(x_* - x) = d_* = -(B^H + \lambda_* D)^{-1} J^T f$$

where λ_* is the vector of *optimal* Langrange multipliers. If now λ is some *approximation* to these optimal multipliers, then the step to $x_+ = x + d$, given by (2.5), solves instead the following version of (2.7):

$$\underset{z \in R^n}{\text{minimize}}\ M(z) \tag{2.8}$$

$$\text{subject to}\quad (z - x)^T D(z - x) \leq (x_+ - x) D (x_+ - x)$$

The reduction in model function value (or *predicted* reduction) for a step d is easily shown to be

$$\Delta M^H = -d^T g - \frac{1}{2} d^T B^H d \tag{2.9}$$

and from (2.8) it follows that $\Delta M^H \geq 0$. (For the *actual* reduction in function value for this step we use the notation ΔF.)

3. Implementation

The main purpose behind our implementation is to investigate whether the Hessian approximation (2.1) is viable. Therefore we formulate a quite rudimentary, (but we think also elegant) algorithm. Its main cycle consists of an inner loop for revising λ and improving the approximation to the minimum, contained within an outer loop for revising α and updating various other quantities.

We shall denote the curent estimate of the solution by x, the Jacobian at x by J, the vector of function values by f, the gradient by g, the function value by F, the current quasi-Newton approximation by B^{QN}, and the identity matrix by I. The symbol + will be attached to denote another set of such quantities at the point x_+. The algorithm is as follows:

Initialize 1A: Given an initial point x_+ form F_+, f_+, J_+, $g_+ = J_+^T f_+$;

Set $\lambda \leftarrow 0$ and $\alpha \leftarrow 0.5$;

Comment: In the absence of other information we set α to the

halfway point;

Reinitialize 1B: Set $B^{QN} \leftarrow I$;

Outerloop 2A: Make proposal x_+ the current estimate of the solution.

$$x \leftarrow x_+ \, , \, g \leftarrow g_+ \, , \, F \leftarrow F_+ \, , \, f \leftarrow f_+ \, , \, J \leftarrow J_+ ;$$

Innerloop 3A: Develop search direction d by solving $(\alpha J^T J + (1 - \alpha) B^{QN} + \lambda I) d = -J^T f$;

Comment: In our implementation we do not use the scaling matrix D.

3B: Develop next point $x_+ = x + d$ and compute F_+ and f_+. J_+ need not be evaluated yet.

3C: Revise λ, (see a) below);

3D: *If* $F_+ \geq F$ *then goto* Innerloop;

2B: Revise α, (see b) below);

2C: Compute J_+ and $g_+ = J_+^T f_+$;

2D: Check for convergence. *If* $(g_+^T g_+)^{1/2} \leq$ tol *then stop*;

2E: *If* $(g_+ - g)^T (x_+ - x) \leq 0$ *then* $\alpha \leftarrow 1$ and *goto* Reinitialize;

Comment: We reinitialize α when a positive definite update of B^{QN} cannot be ensured. This occurs so infrequently on most examples, that another strategy would make little difference.

2F: Update B^{QN}, (see c) below, and *Goto Outerloop*;

We have taken the view that if the Hessian approximation (2.1) or (2.4) is indeed worthwhile, then the above algorithm should not be unduly sensitive to strategies for choosing λ , α and B^{QN}, provided of course, they are reasonable. Therefore, we have utilized very simple strategies for choosing λ and α, drawing

up techniques already utilized elsewhere, and in particular, ones whose implementations were conveniently at hand. Details in each case are given in the Appendix. Of course, when developing a more practical implementation, careful attention must be paid to these choices, and we make some further comments to this effect in Section 5.

a) *Choice of* λ: Within each iteration of the above algorithm, λ is updated at Step 3C using the method of Fletcher [5]. There are now more effective ways of choosing λ, see [13] and [25], but for our experimental needs, Fletcher's method is quite adequate. (Details are given in the Appendix for the interested reader.) Note in particular that if $F_+ \geq F$ then λ will be increased.

b) *Choice of* α: Consider a step $(x_+ - x) = d$ determined by the solution of (2.5). Instead of the model function (2.6) with Hessian approximation (2.4) we see whether a *Gauss-Newton model* given by

$$M^{GN}(z) = g^T(z - x) + \frac{1}{2}(z - x)^T(J^T J)(z - x) \tag{3.1}$$

or a *Quasi-Newton model* given by

$$M^{QN}(z) = g^T(z - x) + \frac{1}{2}(z - x)^T B^{QN}(z - x) \tag{3.2}$$

would be more appropriate, and give predicted reductions that better match the actual redcution ΔF. Again we use a simple method, details of which are given in the Appendix.

c) *Choice of* B^{QN}: since the subroutines implementing the optimally conditional method of Davidon [16] were conveniently available, see Davidon and Nazareth [26], we used them in our implementation. Some advantages of using this method are discussed in the Appendix.

4. Test Results

Although test results on a few test problems should be viewed with caution, they help to discern certain broad trends in performance and to confirm the theoretical soundness of an algorithm. We report here the results of exercising the hybrid algorithm on a set of 11 test problems, 6 of them having zero residuals at the solution, and the remaining 5 having non-zero and usually large residuals at the solution. The test problems were as follows:

a) Zero Residual Problems

1. Rosenbrock's parabolic valley. See Brent [18] page 139.
2. Powell's quartic. See Brent [18] page 141.
3. Powell's badly scaled function. See Powell [19] page 146.
4. Fletcher's helical valley. See Brent [18] page 140.
5. Watson's function, n = 9. See Brent [18] page 142.
6. Wood's quartic. See Brent [18] page 141.

b) Non-zero Residual Problems

7. Modified Box's exponential problem obtained by adding 20 to the second function and 10 to the fourth function. This gives a residual of 0.308E3 at the solution. See Brent [18] page 140.
8. Brown-Dennis. See Brown [9] page 12.
9. Freudenstein and Roth problem. See Aird [20] page 98.
10. Davidon's large residual problem. See Davidon [23].
11. Jennrich and Sampson problem. See Aird [20] page 95.

In addition to running the hybrid algorithm on this set of test problems, we were able to obtain results for the Levenberg-Marquardt algorithm by fixing α at value 1, and for the Quasi-Newton algorithm by fixing α at value 0. For

each algorithm (under headings Hybrid, Levenberg-Marquardt and Quasi-Newton) we report a pair of numbers associated with each test problem. The upper number is the number of function calls and the lower the number of Jacobian calls. In the column headed 'Dimension', each entry is of the form $\binom{n}{m}$ where n is the number of variables and m is the number of functions f_i.

We feel that our comparison between the Hybrid, Levenberg-Marquardt and Quasi-Newton algorithms are entirely fair, since the implementations used in the comparisons differ only in the essentials. Note however that since the λ strategy used is a simple one, and since no scaling matrix D is used, our LM specialization cannot be expected to perform as well as a more sophisticated implementation. Note also that in our Quasi-Newton specialization, search directions are obtained by solving $(B^{QN} + \lambda I)d = -J^T f$. This is an alternative to using a line search, but costs $0(n^3)$ operations per iteration. It has however the advantage of approximating an optimal local step.

Our test results are shown in the next table.

On zero residual problems, we see that the Levenberg-Marquardt algorithm does very well, as compared to the other two. Only on problem 6, namely Wood's quartic, does it fare substantially worse than the hybrid, but since minor variations of initial path can lead to substantial difference in performance on this function, this negative result should be somewhat discounted. On non-zero residual problems, however, the hybrid method comes much more into its own. On problems 7 and 8 it becomes competitive with the L-M algorithm, and on problems 9 and 10 it does much better. Only on problem 11 does it fare badly, and here the results of the Q-N method indicate that the QN approximation is not behaving as expected. (However, we have not investigated this further.) Note also that the results show the hybrid method doing better than either of its specializations on non-zero residual problems, which is what

Problem	Dimension	Hybrid	Levenberg-Marquardt	Quasi-Newton
1	2	35	28	48
	(2)	21	16	39
2	4	14	9	28
	(4)	11	9	24
3	2	322	278	249
	(2)	160	143	207
4	3	21	15	38
	(3)	15	12	31
5	9	18	5	44
	(31)	11	5	39
6	4	37	97	47
	(7)	21	71	34
7	3	24	26	24
	(10)	13	13	22
8	4	32	41	50
	(6)	28	24	42
9	2	11	25	17
	(2)	11	23	13
10	4	21	36	27
	(20)	16	28	19
11	2	22	18	2^{+}
	(10)	16	13	2

+ found a different local minimum.

we had hoped to see.

We summarize the results in the next table, where figures in columns 1 and 2 are equivalent function evaluations i.e. number of function calls + dimension × number of gradient (Jacobian) calls. The third column gives for each problem the amount by which the *better* method outperforms the other, expressed as a percentage. A figure in column 3 if given in italics, and followed by a + indicates the hybrid is the better method, and if in regular script indicates the L-M did better than the hybrid. The percentage is calculated as stated in the Table.

Problem	Hybrid EQF(H)	Levenberg-Marquardt EQF(LM)	$\frac{\lvert EQF(H)-EQF(LM)\rvert * 100}{\max(EQF(H),EQF(LM))}$	
1	*77*	60	22%	
2	*58*	45	23%	
3	*642*	564	13%	
4	*66*	51	23%	
5	*117*	54	54%	
6	*121*	381	*69%*	+
7	*63*	65	*4%*	+
8	*144*	137	5%	
9	*33*	71	*54%*	+
10	*85*	148	*43%*	+
11	*54*	44	19%	

5. Conclusions and Comments on a More Practical Implementation

The test results indicate that on large residual problems the hybrid approximation may very well be worthwhile. It would also seem evident that a better strategy for choosing α which takes into account, for example, the size of the residual at the current iterate, would improve the performance of the hybrid on zero residual poblems, and make it much more competitive with the L-M method on such problems.

If one were to develop a practical implementation based upon the hybrid approximation to the Hessian, there are many potential improvements to the skeletal algorithm of Section 3. These include:

a) a better α strategy as mentioned above and a λ strategy following More [13] and Hebden [25]

b) use of a diagonal scaling matrix D and an initialization of the quasi-Newton approximation to the starting $(J_0^T J_0)$, provided it is nonsingular, of course, rather than to the identity matrix.

c) use of the currently favoured BFGS update rather than the one in [16], and maintaining B^{QN} as the product of a lower and an upper triangular matrix for

stability of the update.

d) use of the QR factorization to solve the system of equations that define the search direction. See, again, More [13].

e) reformulation of calculations to avoid destructive overflows and underflows.

f) an overall design that makes it possible to solve general nonlinear unconstrained optimization problems, small (or zero) residual nonlinear least squares problems and large residual nonlinear least squares problems within the framework of a single adaptive algorithm. Indeed, one of the goals of this paper is to promote the development of such an implementation.

REFERENCES

1. Ben-Israel, A. (1966), "A Newton-Raphson method for the solution of systems of equations," *Journal of Math. Anal. & Applics*. 15, 243-252.

2. Fletcher, R. (1968), "Generalized inverse methods for the best least squares solution of systems of nonlinear equations," *Computer Journal* 10, 392-399.

3. Levenberg, K. (1944), "A method for the solution of certain nonlinear problems in least squares," *Q. Appl. Math*. 2, 164-168.

4. Marquardt, D.W. (1963), "An algorithm for least squares estimation of nonlinear parameters," *SIAM J. Numer. Anal*. 11, 431-441.

5. Fletcher, R. (1971), "A modified Marquardt subroutine for nonlinear least squares," UKAEA Research Group Report, AERE R. 6799, Harwell, England.

6. Powell, M.J.D. (1969), "Subroutine VA05A," AERE, Harwell, England.

7. Steen, N. and Byrne, G.B. (1974), "The problem of minimizing nonlinear functionals," in *Numerical Solution of Systems of Nonlinear Equations*, Byrne & Hall (Eds.), Academic Press, New York, 185-237.

8. Brown, K.M. and Dennis, J.E. (1971), "New computational algorithms for minimizing a sum of squares of nonlinear functions," Yale University, Department of Computer Science Report No. 71-6.

9. Brown, K.M. (1972), "Computer oriented methods for fitting tabular data in the linear and nonlinear least squares sense", *Proc.FJCC*, Anaheim, California.

10. Powell, M.J.D. (1976), "Quadratic termination properties of Davidon's new variable metric algorithm," Report CSS 33, AERE, Harwell, England.

11. Dennis, J.E. (1973), "Some computational techniques for the nonlinear least square problem," in *Numerical Solution of Systems of Nonlinear Algebraic Equations*, Byrne and Hall (Eds.), Academic Press, New York.

12. Broyden, C.G. (1970), "The convergence of a class of double rank minimization algorithms Parts 1 and 2," *J. Inst. Math. Applics.* 6, 76-90, 222-231.

13. More, J.J. (1977), "The Levenberg-Marquardt method: implementation and theory," in *Numerical Analysis*, G.A. Watson (Ed.), Lecture Notes in Mathematics, P. 630, Springer-Verlag.

14. McKeown, J.J. (1975), "Specialized versus general purpose algorithms for minimizing functions that are sums of squared terms," *Math. Prog.* 9, 57-68.

15. Nazareth, J.L. (1980), "Some recent approaches to solving large residual nonlinear least squares problems," *SIAM Review*, 22, 1-11.

16. Davidon, W.C. (1975), "Optimally conditioned optimizatiion algorithms without line searches," *Math. Prog.* 9, 1-30.

17. Gill, P.E. and Murray, W. (1979), "Conjugate and gradient methods for large-scale nonlinear optimization," Tech. Rep. SOL 79-15, Department of Operations Research, Stanford University.

18. Brent, R.P. (1973), *Algorithms for Minimization without Derivatives* , Prentice Hall, New Jersey.

19. Powell, M.J.D. (1970), "A hybrid method for nonlinear equations," in *Numerical Methods for Nonlinear Algebraic Equations*, Gordon and Breach, 87-114.

20. Aird, T.J. (1973), Ph.D. thesis, Purdue University Computer Centre, Lafayette, Indiana.

21. Betts, J.T. (1976), "Solving the nonlinear least squares problem: application of a general method," *JOTA* 18, 469-484.

22. Dennis, J.E., Gay, D.M. and Welsch, R.E. (1977), "An adaptive nonlinear least squares algorithm," MIT Operations Research Center Report, No. 142, Boston.

23. Davidon, W.C. (1977), "New least squares algorithms," *JOTA* 18, 187-198.

24. Fletcher, R. (1970), "A new approach to variable metric algorithms," *Computer J.* 13, 317-322.

25. Hebden, M.D. (1973), "An algorithm for minimization using exact second derivatives," AERE Report TP515, Harwell, England.

26. Davidon, W.C. and Nazareth L. (1977), "DRVOCR - A Fortran implementation of Davidon's optimally conditioned method," TM-306, Applied Mathematics Division, Argonne National Laboratory, Illinois, USA.

6. Appendix

As mentioned earlier, the particular strategies for choosing λ, α and B^{QN} are not central to our experimental implementation. However, for completeness, we give them here.

a) Choice of λ :

When $B^H = \alpha J^T J + (1-\alpha)B^{QN}$ and $g = J^T f$, and we define $a = d^T g$, $b = d^T J^T J d$, $c = d^T B^{QN} d$, then the predicted reduction, which we denote by ΔM^H is

$$\Delta M^H = -a - \frac{1}{2}\alpha b - \frac{1}{2}(1-\alpha)c$$

Fletcher's method [5], as adapted to our needs, is based upon a comparison between ΔM^H and ΔF. If we define

$$r = \begin{cases} \Delta F / \Delta M^H & \text{if } \Delta M^H \neq 0 \\ \infty \cdot \text{sign}(\Delta F) & \text{if } \Delta M^H = 0 \end{cases}$$

the method, is as follows:

If $\rho \leq r \leq \sigma$ for certain constants $0 < \rho < \sigma < 1$ (typically $\rho = 0.25$ and $\sigma = 0.75$), λ is left unchanged.

If $r > \sigma$ then λ is decreased by a fixed factor. We use 1/10. If $\lambda < \lambda_c$, where $\lambda_c = 1/||G^{-1}||_{\text{spectral}}$, then λ is set to zero. This device increases the rate of convergence near the minimum. $||G^{-1}||_{\text{spectral}}$ is overestimated from trace (G^{-1}), and as stated above, G is given by $\alpha J^T J + (1-\alpha)B^{QN}$.

If $r < \rho$, then λ is increased to $v\lambda$. v is chosen by making a quadratic fit to $\delta(\vartheta) = F(x + \vartheta d)$ at $\delta(0)$, $\delta^1(0)$ and $\delta(1)$, subject to ensuring

that $2 \leq v \leq 10$. If λ is zero and must be increased, then $\lambda \leftarrow \lambda_c v / 2$, and λ_c is only recalculated under these circumstances, making the estimation of $||G^{-1}||_{spectral}$ very infrequent.

b) Choice of α :

(i) To determine the model towards which to bias

Step 1: $\Delta M^{GN} = -a - 1/2b$; $\Delta M^{QN} = -a - 1/2c$; $\Delta F > 0$;

Comment: This uses (3.1) and (3.2). a, b and c were defined earlier.

Step 2: *if* $(b > c)$

then

Comment: we know that $\Delta M^{GN} \geq 0$ since it is at least as large as ΔM^{H} and we know that $\Delta M^{H} \geq 0$. γ and τ are constants which we set to 0.8 and 0.2 respectively;

if $\Delta F / \Delta M^{GN} \geq \gamma$ *then* bias to GN (as discussed under (ii) below) and *return*;

if $\Delta M^{QN} \leq 0$ *then return*;

if $\Delta F / \Delta M^{GN} \leq \tau$ *then* bias to QN (see (iii) below) and *return*;

if $\Delta M^{QN} / \Delta f \geq \gamma$ *then bias* to QN and *return*;

Step 3: *else*

Comment: we now know that $\Delta M^{QN} \geq 0$;

if $\Delta M^{GN} / \Delta F \geq \gamma$ *then* bias to GN and *return*;

if $\Delta M^{GN} / \Delta F \leq \tau$ *then* bias to QN and *return*;

if $\Delta F / \Delta M^{QN} \geq \gamma$ *then* bias to QN and *return*;

(ii) Bias toward GN is carried out as follows:

if $\alpha = 0$ *then* $\alpha \leftarrow 0.05$;

if $0 < \alpha \leq 1/3$ *then* $\alpha \leftarrow 2\alpha$;

if $1/3 < \alpha < 0.95$ *then* $\alpha \leftarrow (1 + \alpha)/2$;

if $\alpha \geq 0.95$ *then* $\alpha \leftarrow 1$;

(iii) Bias toward QN is carried out by replacing α in (ii) above by $(1 - \alpha)$.

c) Choice of B^{QN}:

We use the variable metric update developed by Davidon [16]. Many variable metric methods have the property of finite termination on a function $F(x)$ defined by (1.1) with f_i linear for each i, provided line searches are exact and search directions d at each iteration are defined by $B^{QN}d = -g$, where g is the gradient at the current point. Davidon's method [16] which can be viewed as a stabilization of the symmetric rank-1 method, (see Powell [10]), does not suffer from these limitations and allows a great deal of flexibility in choice of search directions and iterates. We find this property helpful because seach directions are defined in our hybrid algorithm, by $Bd = -g$ with B given by (2.4). The use of Davidon's updates in forming B^{QN} thus ensures finite termination of the hybrid algorithm on functions for which f_i are linear, in at most n steps, regardless of the initial approximation B_0^{QN}. B^{QN} is maintained in factored form $\chi^T\chi$.

Given a step Δx along d with corresponding change in gradient Δg, B^{QN} is updated to B_+^{QN}, say, by a rank-2 matrix composed from $(\Delta g - B^{QN}\Delta x)$ and "updating vector" w, rather than the more conventional Δg. (The factored form $\chi^T\chi$ could have χ^T lower triangular.) The optimally conditional update is made at each iteration.