Optimization in Action

Proceedings of the Conference on Optimization in Action held at the University of Bristol in January 1975, organised by the Institute of Mathematics and its Applications

Edited by

L. C. W. DIXON

Hatfield Polytechnic
Hatfield, England

ACADEMIC PRESS INC. (LONDON) LTD.
24/28 Oval Road,
London NW1

United States Edition published by
ACADEMIC PRESS INC.
111 Fifth Avenue
New York, New York 10003

Library of Congress Catalog Card Number: 76–25697
ISBN: 0-12-218550-1

Printed by photolithography in Great Britain by
T. & A. Constable Ltd, Edinburgh

CONTENTS

A Survey of Methods for Minimizing Sums of Squares of Nonlinear Functions

Shirley A. Lill

(University of Liverpool)

SUMMARY

The survey classifies the types of sums of squares problems that arise, according to the complexity of the function to be minimized and the available level of function information. An outline of methods for each class of problem is given with an indication of readily available algorithms.

1. INTRODUCTION

The problem of minimizing a nonlinear function that has the form of a sum of squares of other functions is one of the most commonly occurring types of minimization problem. It frequently arises in the fields of engineering and applied science where the theory predicts that a certain process should satisfy some functional relationship or model and the experimenter obtains data in order to ascertain the values of the variable parameters of this model. Such problems are essentially curve fitting problems, where the form of the function is known. Other problems which can result in a sum of squares formulation are the solution of simultaneous nonlinear equations and the more general parameter estimation problems such as those described by Bard (1970).

Once a problem has been posed as finding the minimum of a sum of squares it can be tackled by either using a straightforward minimization tech-

nique (Murray (1972)) or the minimization process can be adapted to exploit the special nature of the function. It is this latter approach which is described in this paper.

Section 2 introduces notation and shows how a sum of squares function arises from a curve fitting problem. It also indicates points for consideration when minimizing sums of squares functions. Section 3 describes the basic approaches for solving this type of problem, whilst section 4 examines the central issue of solving the linear least squares equations at each iteration. In section 5 the question of use of derivatives of the function is considered and in section 6 some special problems are discussed. Finally, some suggestions for choosing methods are given in section 7.

Surveys of methods for minimizing sums of squares problems are also given by Powell (1972), Bard (1970), Dennis (1972) and Brown (1972). Some comparative numerical results are given by Bard (1970), Box (1966), Brown and Dennis (1972) and McKeown (1974).

2. A DISCUSSION OF THE PROBLEM

Consider determining the values of the parameters $\underset{\sim}{x} = (x_1, x_2, x_3, \ldots, x_n)^T$ which satisfy the relationship:

$$y = F(\underset{\sim}{t}, \underset{\sim}{x}), \tag{2.1}$$

where y is the dependent variable, and $\underset{\sim}{t} = (t_1, t_2, \ldots, t_K)^T$ are the independent variables for a certain process modelled by the function F. A set of m experiments or observations is made to obtain values of y for different values of $\underset{\sim}{t}$, and these satisfy the equations:

$$y_i = F(\underset{\sim}{t}_i, \underset{\sim}{x}) + \varepsilon_i, \quad i = 1(1)m, \tag{2.2}$$

where the ε_i are randomly distributed independent experimental errors. The problem is to find those values of $\underset{\sim}{x}$ which give the experimental data the best fit to (2.1).

The best fit in the least squares sense is obtained by defining residuals:

$$f_i(\underset{\sim}{x}) = F(\underset{\sim}{t}_i, \underset{\sim}{x}) - y_i, \quad i = 1(1)m, \qquad (2.3)$$

and minimizing the sum of squares of these residuals:

$$S(\underset{\sim}{x}) = \sum_{i=1}^{m} f_i(\underset{\sim}{x})^2 = \underset{\sim}{f}(\underset{\sim}{x})^T \underset{\sim}{f}(\underset{\sim}{x}), \qquad (2.4)$$

where $\underset{\sim}{f}(\underset{\sim}{x}) = (f_1(\underset{\sim}{x}), f_2(\underset{\sim}{x}), \ldots f_m(\underset{\sim}{x}))^T$, with respect to $\underset{\sim}{x}$.

Note that it is possible to solve this type of problem by obtaining the best fit in some other sense such as by minimizing the maximum residual (for example Osborne (1971)) or by using a combination of least squares and minimax. It is also possible to fit smooth curves to experimental data when there is no known model function and thus interpolate intermediate values. Such techniques known as data fitting are described by Cox and Hayes (1973).

When m is greater than n, that is, there are more observations than parameters, the nonlinear least squares problem is said to be over-determined, and in many curve fitting applications for example, m will be significantly larger than n. However a very important class of problems is that of finding the solution of a set of n simultaneous equations in n unknowns (*i.e.*, $m = n$). This particular problem is covered here only as a special case of the over-determined type, and the reader is referred to Ortega and Rheinbolt (1970) for a full exposition. Similarly, the solution when the model is linear is not explicitly discussed here, except that the methods are essentially the same as those described in section 4. Finally, when m is less than n the system is said to be under-determined and only certain of the methods can be applied.

Points for consideration when choosing an algorithm for solving nonlinear least squares problems are introduced as appropriate in the text.

However, certain key points are listed here because of their underlying importance in the discussion of the methods:

(*a*) Can the problem be expressed as a simple sum of squares, or are there further considerations such as constraints, errors in the t_i and so on? Can it be broken down into a simpler form?

(*b*) Is the model a good one and are the experimental errors ε_i in equations (2.2) small so that the minimum of $S(\underset{\sim}{x})$ is zero (*i.e.*, are the equations (2.3) consistent)?

(*c*) Can analytical partial derivatives (no more than second order) of the functions $f_i(\underset{\sim}{x})$ be evaluated, and at what cost, in terms of effort in obtaining the formulae and computer time in calculation?

(*d*) Is a good estimate of the solution available? Is the sum of squares function well-behaved in the region of search, that is, are there other local minima, is the function singular and so on?

(*e*) Are the residuals $f_i(\underset{\sim}{x})$ expensive to calculate, in terms of computer time? Are any of them linear?

(*f*) What are the sizes of m and n?

3. METHODS OF SOLUTION

3.1 Gauss-Newton method

The problem is to find the least value of the function:

$$S(\underset{\sim}{x}) = \underset{\sim}{f}(\underset{\sim}{x})^T \underset{\sim}{f}(\underset{\sim}{x}), \qquad m \geqslant n.$$

Now it is well known that a stationary point of any function $S(\underset{\sim}{x})$ occurs when the gradient $\nabla S(\underset{\sim}{x}) = \underset{\sim}{0}$ and for that stationary point to also be a local minimum of the function the Hessian matrix of second derivatives $\nabla^2 S(\underset{\sim}{x})$ must be positive definite. One way of locating such a point is to use Newton's

classical minimization method (Barnes (1965)) where, starting from an initial estimate of the minimum $\underset{\sim}{x}_0$, a correction $\underset{\sim}{d}_K$ is applied iteratively, $K = 0,1,2,\ldots$, until convergence:

$$\underset{\sim}{x}_{K+1} = \underset{\sim}{x}_K + \underset{\sim}{d}_K \ . \tag{3.1}$$

The correction $\underset{\sim}{d}_K$ is the solution to the equations:

$$\underset{\sim}{\nabla}^2 S(\underset{\sim}{x}_K)\underset{\sim}{d}_K = - \underset{\sim}{\nabla} S(\underset{\sim}{x}_K) \ ,$$

derived from the Taylor series expansion of the function about $\underset{\sim}{x}_K$.

When $S(\underset{\sim}{x})$ is a sum of squared terms and $\underset{\sim}{f}(\underset{\sim}{x})$ is twice differentiable then the gradient can be expressed in terms of $\underset{\sim}{f}(\underset{\sim}{x})$ and its derivatives as:

$$\underset{\sim}{\nabla} S(\underset{\sim}{x}) = 2J(\underset{\sim}{x})^T \underset{\sim}{f}(\underset{\sim}{x}) , \tag{3.2}$$

where $J(\underset{\sim}{x})$ is the $m \times n$ Jacobian matrix with ijth element:

$$J_{ij}(\underset{\sim}{x}) = \frac{\partial f_i(\underset{\sim}{x})}{\partial x_j} \ .$$

The Hessian of $S(\underset{\sim}{x})$ is given by:

$$\underset{\sim}{\nabla}^2 S(\underset{\sim}{x}) = 2J(\underset{\sim}{x})^T J(\underset{\sim}{x}) + 2 \sum_{i=1}^{m} \underset{\sim}{\nabla}^2 f_i(\underset{\sim}{x}) f_i(\underset{\sim}{x}) , \tag{3.3}$$

where $\underset{\sim}{\nabla}^2 f_i(\underset{\sim}{x})$ is the second derivative matrix of $f_i(\underset{\sim}{x})$.

A justification for using special methods to exploit the form of $S(\underset{\sim}{x})$, rather than carrying out a straightforward minimization, can at once be seen from these equations since a substantial part of the Hessian (3.3) is obtained by using only first derivatives of the residuals (*i.e.*, $J(\underset{\sim}{x})^T J(\underset{\sim}{x})$) thus removing the need for explicit second derivatives of the residuals which may be expensive to calculate. This observation, and the fact that near the solution, if the residuals are small (which they are for many practical problems) or nearly linear, then the second term in (3.3) is

negligible, has led to many algorithms using the approximation:

$$\nabla^2 S(\underset{\sim}{x}) \simeq 2J(\underset{\sim}{x})^T J(\underset{\sim}{x}) = 2A(\underset{\sim}{x}), \qquad (3.4)$$

Substituting (3.4) and (3.3) Newton's method gives a correction $\underset{\sim}{d}_K$ which is calculated from equations conventionally known as the *normal equations*:

$$J(\underset{\sim}{x}_K)^T J(\underset{\sim}{x}_K)\underset{\sim}{d}_K = -J(\underset{\sim}{x}_K)^T f(\underset{\sim}{x}_K). \qquad (3.5)$$

This algorithm may be derived directly by expanding the residuals $\underset{\sim}{f}$ in a Taylor series about $\underset{\sim}{x}_K$ (Kowalik and Osborne (1968)), and was first put forward by Gauss (1809). It is usually referred to as the Gauss-Newton method.

The solution of the normal equations obviously breaks down if any $J_K{}^T J_K$ is *singular*. However, in practice, and especially when $m >> n$ the eigenvalues of $J^T J$ are usually bounded away from zero so this problem does not arise and convergence for a region near a solution can be proved (Fox (1964), Meyer (1970) and Pereyra (1967)). Now the rate of convergence for Newton's method is second order, but in the Gauss-Newton method the error incurred by neglecting the term $\nabla^2 f$ in (3.3) reduces the rate of convergence to no more than *linear* unless $S(\underset{\sim}{x}^*)=0$ (Brown and Dennis (1972), Meyer (1970) and Osborne (1972)). Osborne (1972) shows that the reduction in $\underset{\sim}{d}_K$ at each iteration is given by:

$$\|\underset{\sim}{d}_{K+1}\| = \|(J_{K+1}^T J_{K+1})^{-1}\frac{d\bar{J}}{dt}\underset{\sim}{f}_K\| \; \|\underset{\sim}{d}_K\| + O(\|\underset{\sim}{d}_K\|^2) \qquad (3.6)$$

where d/dt denotes differentiation with respect to any direction t and $\bar{J}$ indicates mean values. This demonstrates that for all $K > K_0$, if the series is convergent and $0 < \gamma \leq 1$ where:

$$\|(J^T J)^{-1}\frac{dJ}{dt}\underset{\sim}{f}\| = \gamma$$

the rate of convergence is linear, whereas if $\gamma > 1$ the method is actually *divergent*. Obviously if $S(\underset{\sim}{x}^*) = 0, \gamma$ is zero so that convergence is ultimately second order. Powell (1972) and Osborne (1972) give

simple numerical examples demonstrating the consequences of different values of γ.

This deficiency has, in practice, led to a wide variety of modifications and improvements to the basic Gauss-Newton algorithm. Several suggested alternatives are discussed below. They are introduced in the sense of development from the basic algorithm rather than in chronological order.

3.2 Modified Gauss-Newton or Hartley method

The Gauss-Newton is often modified to prevent divergence by using $\underset{\sim}{d}_K$ as a direction along which to search for a lower value of S, so that (3.1) becomes

$$\underset{\sim}{x}_{K+1} = \underset{\sim}{x}_K + \alpha_K \underset{\sim}{d}_K. \qquad (3.7)$$

α_K is a scalar which may be chosen so as to minimize $S(\underset{\sim}{x}_K + \alpha_K \underset{\sim}{d}_K)$ with respect to α_K, or simply to ensure that $S(\underset{\sim}{x}_{K+1}) \leq S(\underset{\sim}{x}_K)$. Kowalik and Osborne (1968) argue that since the cost involved in calculating $\underset{\sim}{d}_K$ is substantial, the further cost in function evaluations of carrying out an accurate linear search is justified, and for J^TJ bounded above and below, Hartley (1961) proves convergence for this version of the method. However, it is often the case that function evaluations are at a premium and several efficient schemes for calculating a suitable α_K to reduce the sum of squares have been suggested. Efficient searches for nonlinear least squares in particular are investigated by Bard (1970) and Osborne (1972) who also supplies a proof of convergence. However near the solution α_K set to unity is usually successful so the convergence is ultimately that of the Gauss-Newton method.

Although these modifications do prevent divergence they do not overcome the problem of solving the normal equation when $J_K^TJ_K$ is singular, nor do they ensure convergence to a solution. In fact they may appear to converge to a point which is not a local minimum of S and Powell (1970) gives

such an example. The method apparently converges to a point where J^TJ is singular so that $J^Tf = 0$, even though it may not be singular elsewhere. The difficulty is that near such points the directions $\underset{\sim}{d}_K$ are almost orthogonal to the descent direction of $S(-\nabla S)$ so that little reduction in the value of S can be made.

3.3 Methods interpolating between Gauss-Newton and steepest descent

Powell's hybrid method (1970) was devised to solve this problem by introducing a search along the steepest descent direction whenever the Gauss-Newton correction is unsuccessful, so that:

$$\underset{\sim}{x}_{K+1} = \underset{\sim}{x}_{K+1} + \alpha_K \underset{\sim}{p}_K,$$

where

$$\underset{\sim}{p}_K = \beta_K \underset{\sim}{d}_K - \gamma_K \underset{\sim}{\nabla S}_K ,$$

$\underset{\sim}{d}_K$ is the solution to (3.2), and α_K, β_K and γ_K are suitable step lengths. The algorithm is briefly as follows.

The Gauss-Newton correction, d_K, is calculated, but if this is deemed too large or does not give a reduction in the sum of squares S, then the predicted minimum of S along the steepest descent direction $\underset{\sim}{\nabla S}$ is calculated. A search for a reduced S is then made along $-\underset{\sim}{\nabla S}$ and the line joining the predicted minimum to the end point of the Gauss-Newton correction, as in Fig. 1.

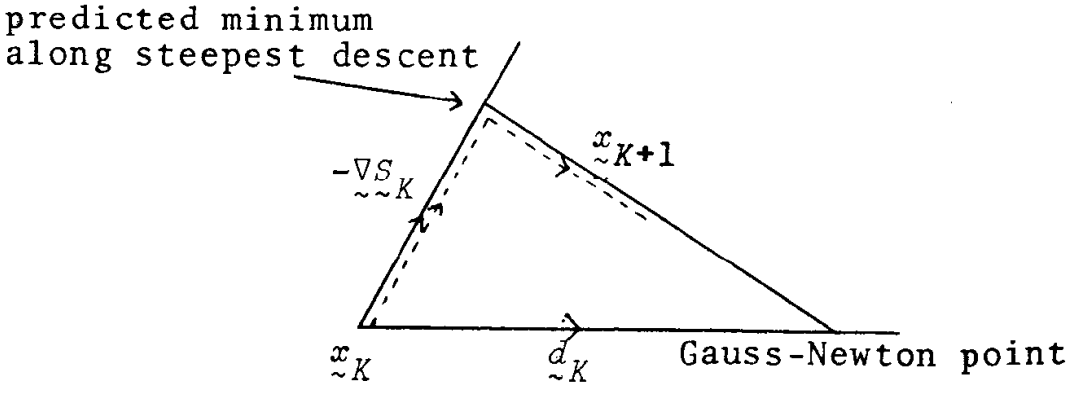

Fig. 1. Powell's hybrid algorithm

Thus, the method interpolates very simply between the Gauss-Newton and steepest descent directions.

An alternative approach, due to Levenberg (1944) and Marquardt (1963), of introducing a bias towards steepest descent whilst at the same time guaranteeing the calculation of $\underset{\sim}{d}_K$ is to add some positive definite matrix D_K to $J_K^T J_K$ in (3.5) and solve:

$$(J_K^T J_K + \lambda_K D_K)\underset{\sim}{d}_K = - J_K^T \underset{\sim}{f}_K, \qquad (3.8)$$

where $\lambda_K > 0$ is some variable parameter.

Providing λ_K is chosen large enough the composite matrix will be positive definite so that $\underset{\sim}{d}_K$ can be calculated and in addition, as $\lambda_K D_K$ is increased, $\underset{\sim}{d}_K$ is forced towards the descent direction so that a reduction in S can always be achieved. $\lambda_K D_K$ can also be considered as an approximation to the term in $\nabla^2 f$ which is ignored in (3.3). For simplicity D_K is usually chosen to be a constant diagonal matrix, either the unit matrix or a matrix which reflects the scaling of the variables (Marquardt (1963)).

The methods suggested by Levenberg and Marquardt differ slightly but the main idea behind them is that at each iteration, given a value λ_K, the equations (3.8) are successively solved for increasing values of λ_K until a $\underset{\sim}{d}_K$ is obtained such that $S(\underset{\sim}{x}_{K+1}) < S(\underset{\sim}{x}_K)$ (Levenberg actually suggests finding the minimum S with respect to λ_K) when $\underset{\sim}{x}_{K+1}$ is accepted as the new iterate and λ_K is decreased by some constant factor. It is easy to show that, as $\lambda_K \to 0$, $\underset{\sim}{d}_K$ tends to the Gauss-Newton correction $-(J_K^T J_K)^{-1} J_K^T \underset{\sim}{f}_K$, and as $\lambda_K \to \infty$, $\underset{\sim}{d}_K$ tends to the descent direction $-\nabla S$ and the trajectory of the end point of $\underset{\sim}{d}_K$ for varying λ_K is as shown in Fig. 2.

From Fig. 2 it can be seen that increasing λ_K has the effect of reducing the size of $\underset{\sim}{d}_K$ as well as altering its direction, thus α_K may in theory be set to unity in calculating $\underset{\sim}{x}_{K+1}$ (3.7).

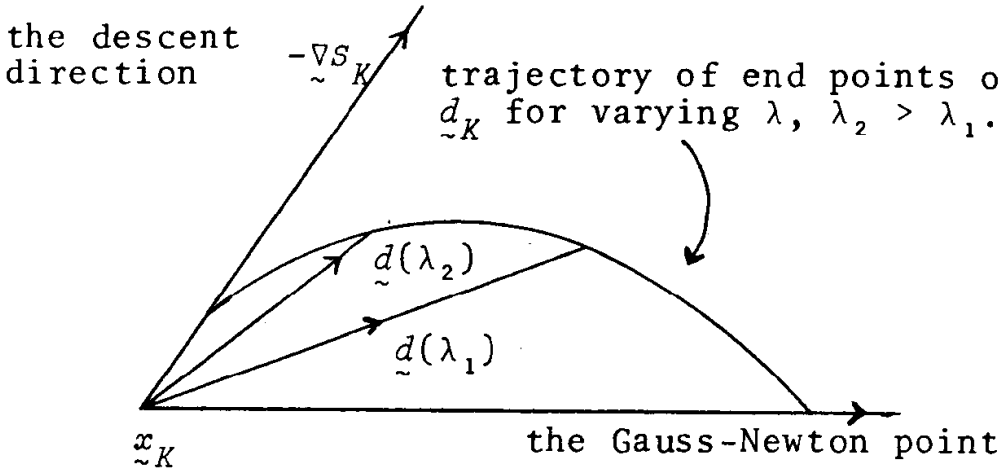

Fig. 2. Levenberg-Marquardt algorithm

However against this simplification must be measured the cost of re-solving (3.8) every time λ_K is altered, and so it is usually more efficient to vary α_K. A sensible strategy for choosing λ_K is to let λ_0 be some significant value, thus ensuring a reduction in S at points far removed from the solution, and to employ an over-all reduction philosophy (although λ may be temporarily increased) so that $\lambda \to 0$ as $x \to x^*$ and the second order convergence of Gauss-Newton (for $S(x^*) = 0$) can be attained (Brown and Dennis (1972), Meyer (1970) and Osborne (1972)). Computational schemes for calculating λ_K are given by Fletcher (1971), Meyer (1970) and Osborne (1972) together with limited numerical results.

To avoid the work involved in the recalculation of d_K when λ_K is changed Bard (1970) calculates the eigenvalues of $(J_K^T J_K + \lambda_K D_K)$ at each iteration and uses them to evaluate d_K. This allows the smallest possible λ_K to be chosen initially so that the condition number of the composite matrix is limited to lie within specified bounds and ensures that it can be adjusted at little extra cost if the resulting sum of squares is not acceptable. However the cost of the eigensolutions is high and a simpler method is suggested by Jones (1970) where the trajectory of Levenberg and Marquardt is replaced by a spiral which has the same

end points, but is of constant slope so that points along it are simple to obtain. See Fig. 3.

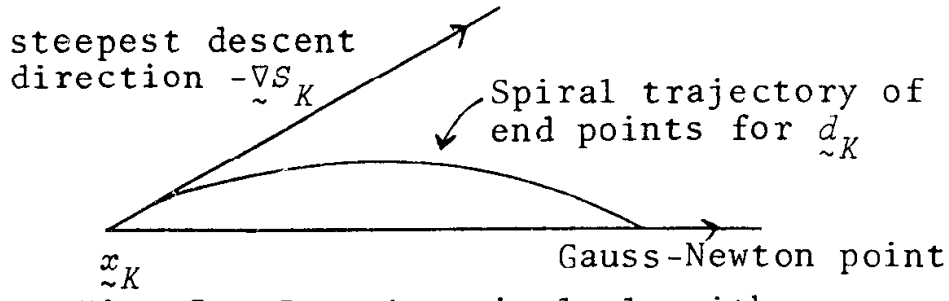

Fig. 3. Jones's spiral algorithm

Once the spiral is set up discrete points along it can be calculated simply by vector additions.

4. SOLUTION OF THE NORMAL EQUATIONS

This section outlines possible approaches to the solution of the normal equations which are central to most of the nonlinear least squares techniques discussed so far. Dropping the subscript for convenience, equation (3.5) becomes:

$$J^T J d = - J^T f . \qquad (4.1)$$

The efficiency and accuracy of the solution of these equations are crucial since they have to be solved at every iteration of the nonlinear least squares algorithm.

Traditionally (4.1) is solved by evaluating $J^T J$ and performing a Cholesky factorization (see Fox (1964)) to obtain LL^T where L is a lower triangular matrix and then carrying out back substitutions to obtain d. However, forming the product $J^T J$ worsens the conditioning of the problem and leads to a loss of accuracy (Businger and Golub (1965)), and recently algorithms have been given (Gill and Murray (1972)) which avoid the explicit calculation of $J^T J$ by performing instead an orthogonal triangularization of J.

An alternative method which avoids the compu-

tation of J^TJ is due to Businger and Golub (1965). Reposing the problem as solving:

$$J\underset{\sim}{d} = -\underset{\sim}{f} \qquad (4.2)$$

in the least squares sense and factorizing J into $Q\begin{bmatrix} R \\ \cdots \\ 0 \end{bmatrix}$ where Q is an $m \times m$ orthogonal matrix and R is $n \times n$ right triangular, reduces (4.2) to:

$$\begin{bmatrix} R \\ \cdots \\ 0 \end{bmatrix}\underset{\sim}{d} = -Q^T\underset{\sim}{f}$$

since the problem is invariant to orthogonal transforms. The solution is obtained by setting $\underset{\sim}{b}$ to the first n elements of $-Q^T\underset{\sim}{f}$ and then solving the triangular system $R\underset{\sim}{d} = \underset{\sim}{b}$ to obtain $\underset{\sim}{d}$. The factorization of J into QR is very stable and an important implementation detail is that Q, which may be very large, need not be stored since $\underset{\sim}{b}$ can be built up simultaneously.

A third method, which should be used when the matrix J is of rank less than n, or cannot be guaranteed to be of full rank, is the singular value decomposition due to Golub and Reinsch (1970). Again this uses (4.2), but J is factorized into UDV^T where:

U - $m \times n$ orthogonal matrix made up of the eigenvectors of the n largest eigenvalues of JJ^T.

D - $n \times n$ diagonal matrix of singular values of J^TJ (non-negative square roots of the eigenvalues).

V - $n \times n$ orthogonal matrix made up of the eigenvectors of J^TJ.

so that $\underset{\sim}{d}$ is obtained by setting:

$$\underset{\sim}{d} = -VD^+U^T\underset{\sim}{f} \quad ,$$

where $D^+_{ii} = 1/D_{ii}$, $D_{ii} > 0$,
$= 0$, otherwise.

In addition to dealing with J of rank $< n$, this method is also useful if information on the