SU326 P30-15

THE DIFFERENTIATION OF PSEUDOINVERSES AND NONLINEAR LEAST SQUARES PROBLEMS WHOSE VARIABLES SEPARATE

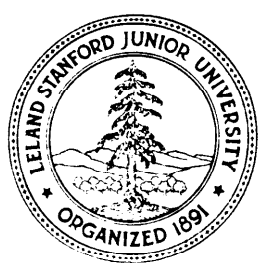
BY

G. H. GOLUB AND V. PEREYRA

STAN-CS-72-261

FEBRUARY 1972

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY



The Differentiation of Pseudoinverses and Nonlinear Least Squares Problems Whose Variables-Separate

G. H. Golub*

and

V. Pereyra**

*Computer Science Department, Stanford University, Stanford, California 94305.
This work was in part supported by the Atomic Energy Commission.

**Departamento de Computacion, Universidad Central de Venezuela, Caracas, Venezuela.

ABSTRACT

For given data (t_i , y_i) , $i=1,\ldots,m$, we consider the least squares fit of nonlinear models of the form

$$F(\underset{\sim}{a} , \underset{\sim}{\alpha} ; t) = \sum_{j=1}^{n} g_j(\underset{\sim}{a})\, \varphi_j(\underset{\sim}{\alpha} ; t) , \quad \underset{\sim}{a} \in \mathbb{R}^s , \quad \underset{\sim}{\alpha} \in \mathbb{R}^k .$$

For this purpose we study the minimization of the nonlinear functional

$$r(\underset{\sim}{a} , \underset{\sim}{\alpha}) = \sum_{i=1}^{m} (y_i - F(\underset{\sim}{a}, \underset{\sim}{\alpha}, t_i))^2 .$$

It is shown that by defining the matrix $\{\Phi(\underset{\sim}{\alpha})\}_{i,j} = \varphi_j(\underset{\sim}{\alpha} ; t_i)$, and the modified functional $r_2(\underset{\sim}{\alpha}) = \| \underset{\sim}{y} - \Phi(\underset{\sim}{\alpha})\Phi^{+}(\underset{\sim}{\alpha})\, \underset{\sim}{y} \|_2^2$, it is possible to optimize first with respect to the parameters $\underset{\sim}{\alpha}$, and then to obtain, a posteriori, the optimal parameters $\hat{\underset{\sim}{a}}$. The matrix $\Phi^{+}(\underset{\sim}{\alpha})$ is the Moore-Penrose generalized inverse of $\Phi(\underset{\sim}{\alpha})$, and we develop formulas for its Fréchet derivative under the hypothesis that $\Phi(\underset{\sim}{\alpha})$ is of constant (though not necessarily full) rank. From these formulas we readily obtain the derivatives of the orthogonal projectors associated with $\Phi(\underset{\sim}{\alpha})$, and also that of the functional $r_2(\underset{\sim}{\alpha})$. Detailed algorithms are presented which make extensive use of well-known reliable linear least squares techniques, and numerical results and comparisons are given. These results are generalizations of those of H. D. Scolnik[20].

1. Introduction

The least squares fit of experimental data is a common tool in many applied sciences and in engineering problems. Linear problems have been well-studied, and stable and efficient methods are available (see for instance: Björck and Golub [3], Golub[8]).

Methods for the nonlinear problems fall mainly in two categories: (a) general minimization techniques; (b) methods of Gauss-Newton type. The latter type of method takes into consideration the fact that the functional to be minimized is a sum of squares of functions (cf. Daniel [5], Osborne [14], Pereyra [15]). The well-known reliable linear techniques have been used mainly in connection with the successive linearization of the nonlinear models. Very recently it has been noticed that by restricting the class of models to be treated, a much more significant use of linear techniques can be made (cf. [2, 9, 12, 13, 17, 20]).

In this paper we consider the following problem. Given data (t_i, y_i), $i=1,\ldots,m$, find optimal parameters $\hat{\underset{\sim}{a}} = (\hat{a}_1, \ldots, \hat{a}_s)^T$, $\hat{\underset{\sim}{\alpha}} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_k)^T$ that minimize the nonlinear functional

$$r(\underset{\sim}{a}, \underset{\sim}{\alpha}) = \sum_{i=1}^{m} [y_i - \sum_{j=1}^{n} g_j(\underset{\sim}{a})\, \varphi_j(\underset{\sim}{\alpha}; t_i)]^2 . \tag{1.1}$$

Throughout this paper a lower case letter in bold face will indicate a column vector, while the same letter with a subscript will indicate a component of the vector. Matrices which are not vectors are denoted by capital letters, and the (i,j) element of (say) a matrix A will be indicated by either a_{ij} or $\{A\}_{i,j}$. The transpose of a vector $\underset{\sim}{u}$ is indicated by $\underset{\sim}{u}^T$. Given a function $f(t)$, we shall denote by $\underset{\sim}{f}$ the vector whose components are

$(f(t_1), f(t_2), \ldots, f(t_m))^T$. The scalar product of two vectors $\underset{\sim}{u}$ and $\underset{\sim}{v}$ is indicated by

$$\langle \underset{\sim}{u}, \underset{\sim}{v} \rangle = \underset{\sim}{v}^T \underset{\sim}{u} .$$

The only norm which will be used is the Euclidean norm, $\| \underset{\sim}{v} \|^2 = \langle \underset{\sim}{v}, \underset{\sim}{v} \rangle$. Given a matrix A and a vector $\underset{\sim}{b}$, then we say

$$A\underset{\sim}{x} \cong \underset{\sim}{b}$$

if $\underset{\sim}{x} = A^+ \underset{\sim}{b}$ where A^+ is the Moore-Penrose pseudoinverse.

We shall use the symbol **D** for the Fréchet derivative of a mapping and V for the gradient of a functional. We assume that the reader has some familiarity with pseudoinverses and Fréchet derivatives and their properties. A useful reference for the pseudoinverse is [19]; for details on the formalism and manipulation of Fréchet derivatives, we suggest [6, chapter 8].

Let

$$\{\Phi\}_{i,j} = \varphi_j(\underset{\sim}{\alpha} ; t_i) \quad (i=1,\ldots,m ; j=1, 2,\ldots,n) ,$$

and

$$\underset{\sim}{g}(\underset{\sim}{a}) = (g_1(\underset{\sim}{a}), g_2(\underset{\sim}{a}), \ldots, g_n(\underset{\sim}{a}))^T .$$

With the given notation, we can rewrite (1.1) as

$$(1.1') \qquad r(a, \underset{\sim}{\alpha}) = \| \underset{\sim}{y} - \Phi(\underset{\sim}{\alpha}) \underset{\sim}{g}(\underset{\sim}{a}) \|^2 .$$

Our approach to finding a critical point or a minimum of the functional (1.1') requires two additional hypotheses:

H-1. For any vector $\underset{\sim}{b} \in \mathbb{R}^n$, the system of nonlinear equations

$$(1.2) \qquad \underset{\sim}{g}(\underset{\sim}{a}) = \underset{\sim}{b} ,$$

has a solution (not necessarily unique).

H-2. The matrix $\Phi(\underset{\sim}{\alpha})$ has constant rank, $r \leq \min(m, n)$ for $\underset{\sim}{\alpha} \in \Omega \subset \mathbb{R}^k$, Ω being an open set containing the desired solution.

Our aim is to be able to deal separately with the parameters $\underset{\sim}{\alpha}$, and then proceed to obtain the parameters $\underset{\sim}{a}$, as it was done in [9, 20] whose results this paper generalize. The reader should also note the independent results obtained by Pérez and Scolnik [17], who in addition deal with nonlinear constraints.

In order to obtain this separation of variables, we consider, as in [9, 17, 20], the modified functional

$$r_2(\underset{\sim}{\alpha}) = \| \underset{\sim}{y} - \Phi(\underset{\sim}{\alpha})\Phi^+(\underset{\sim}{\alpha})\, \underset{\sim}{y}\|^2 , \tag{1.3}$$

which will be called the variable projection functional. Once optimal parameters $\hat{\underset{\sim}{\alpha}}$ have been obtained by minimizing (1.3), then auxiliary parameters $\hat{\underset{\sim}{b}}$ are obtained as $\hat{\underset{\sim}{b}} = \Phi^+(\hat{\underset{\sim}{\alpha}})\, \underset{\sim}{y}$, and finally we take $\hat{\underset{\sim}{a}}$ as any solution of the system of equations (1.2).

We shall show in Theorem 2.1 the relationship between critical or minimal points encountered considering the original functional $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$ and those obtained from the functional $r_2(\underset{\sim}{\alpha})$ and $\Phi^+(\underset{\sim}{\alpha})\underset{\sim}{y}$. Both for our proof and for the numerical algorithms of Section 5, we need to develop formulas for the Fréchet derivative of the pseudoinverse of a matrix function. In Section 4, we develop these formulas and obtain the derivatives of the projectors and the Jacobian of the residual vector. The only hypothesis necessary on the rank of the matrix is that it should be constant on an open neighborhood of the point in which the derivative has to be calculated. This is necessary since otherwise the pseudoinverse is not a continuous function, and therefore it could hardly be differentiable. Our proof is coordinate-free. For the full rank case,

similar formulas have been obtained by Fletcher and Lill [7] (without proof), by Hanson and Lawson [10], and by Pérez and Scolnik [17]. In [7] and [17] this is used to deal with constraints via penalty functions. In [17] the authors choose to work with components, and also obtain a formula for the rank deficient case which is given in terms of the factors of a certain decomposition of the original matrix. Our formulas, besides being coordinate-free and thus much more convenient for algebraic manipulation, are given exclusively in terms of the original matrix, its derivative, and its pseudoinverse. The formula for the rank deficient case seems to be new.

In Section 5 we give a detailed explanation of how to implement the method in an efficient way and in Section 6 we present some numerical examples and comparisons. Extensive use is made of linear least squares techniques.

The authors wish to thank Professor Olof Widlund of the Courant Institute for his careful reading of this manuscript, and to Miss Godela Scherer of the Instituto Venezolano de Investigaciones Científicas for programming assistance. We are also pleased to acknowledge the kind hospitality and stimulating conversations with Dr. H. D. Scolnik of the Bariloche Foundation where this work was initiated in July 1971. Several helpful suggestions were made by Miss Linda Kaufman and Mr. Michael Saunders.

2. A class of nonlinear least square-s problems whose parameters separate.

We are going to consider in this paper models of the form:

$$(2.1) \qquad h(\underset{\sim}{a}, \underset{\sim}{\alpha}; t) = \sum_{j=1}^{n} g_j(\underset{\sim}{a})\, \varphi_j(\underset{\sim}{\alpha}; t) ,$$

where $\underset{\sim}{a} \in \mathbb{R}^s$, $\alpha \in \mathbb{R}^k$, and the functions g_j, φ_j, are continuously differentiable with respect to $\underset{\sim}{a}$, and $\underset{\sim}{\alpha}$ respectively. We shall call the functions g_j autonomous, to distinguish them from the φ_j which are dependent on t. We remark that the parameters $\underset{\sim}{a}$ and $\underset{\sim}{\alpha}$ form two completely disjoint sets.

The independent variable t could be a vector itself as in [9, 17]. This requires only small notational changes and we shall not pursue it here.

Given the data (t_i, y_i), $i=1,\ldots, m$, $m \geq s + k$, our task is to find the values of the parameters $\underset{\sim}{a}$, $\underset{\sim}{\alpha}$, that minimize the nonlinear functional:

$$(2.2) \qquad r(\underset{\sim}{a}, \underset{\sim}{\alpha}) = \| \underset{\sim}{y} - \underset{\sim}{h}(\underset{\sim}{a}, \underset{\sim}{\alpha}) \|^2 = \sum_{i=1}^{m} (y_i - h(\underset{\sim}{a}, \underset{\sim}{\alpha}; t_i))^2 .$$

The approach to the solution of this problem is, as in [9, 17, 20], to modify the functional $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$, in such a way that consideration of the autonomous parameters $\underset{\sim}{a}$ is deferred.

In what follows we shall call $\Phi(\underset{\sim}{\alpha})$ the matrix function

$$(2.3) \qquad \Phi(\underset{\sim}{\alpha}) = [\varphi_1(\underset{\sim}{\alpha}), \ldots, \varphi_n(\underset{\sim}{\alpha})] .$$

For each fixed $\underset{\sim}{\alpha}$, the linear operator

$$(2.4) \qquad P_{\Phi(\underset{\sim}{\alpha})} = \underset{\sim}{\Phi}(\alpha)\, \underset{\sim}{\Phi}^{+}(\alpha) ,$$

is the orthogonal projector on the linear space spanned by the columns of the matrix $\Phi(\underset{\sim}{\alpha})$. We shall denote the linear operator $(I - P_{\Phi(\alpha)})$ by $P^{\perp}_{\Phi(\underset{\sim}{\alpha})}$.

$P^{\perp}_{\Phi(\underset{\sim}{\alpha})}$ is the projector on the orthogonal complement of the column space of $\Phi(\underset{\sim}{\alpha})$. Similarly,

(2.4') $\quad {}_{\Phi}P = \Phi^{+}\Phi$

is the orthogonal projector on the row space of Φ, and ${}_{\Phi}P^{\perp} = I - {}_{\Phi}P$.

When there is no possibility of confusion we shall omit either the matrix subindex or the arguments in projections and functions, or both.

Taking $\underset{\sim}{b}$ as a new parameter vector, we consider the following auxiliary model:

$$(2.5) \qquad n_1(\underset{\sim}{b}, \underset{\sim}{\alpha}; t) = \sum_{j=1}^{n} b_j \varphi_j(\underset{\sim}{\alpha}; t) .$$

We define similarly the functional $r_1(\underset{\sim}{b}, \underset{\sim}{a}) = \| \underset{\sim}{y} - \underset{\sim}{n}_1 \|^2$.

For any given $\underset{\sim}{\alpha}$ we have the minimal least squares solution

$$(2.6) \qquad \underset{\sim}{b}^{*} = \Phi^{+}(\underset{\sim}{\alpha})\, \underset{\sim}{y} .$$

Thus,

$$(2.7) \qquad \min_{\underset{\sim}{b}} r_1(\underset{\sim}{b}, \underset{\sim}{\alpha}) = r_1(\underset{\sim}{b}^{*}, \underset{\sim}{\alpha}) = \| \underset{\sim}{y} - \Phi(\underset{\sim}{\alpha})\Phi^{+}(\underset{\sim}{\alpha})\, \underset{\sim}{y} \|^2 = \| P^{\perp}_{\Phi(\underset{\sim}{\alpha})}\, \underset{\sim}{y} \|^2 .$$

The modified functional is then the variable projection functional that we mentioned earlier and can be rewritten as:

$$(2.8) \qquad r_2(\underset{\sim}{\alpha}) = \| P^{\perp}_{\Phi(\underset{\sim}{\alpha})}\, \underset{\sim}{y} \|^2 .$$

Once a critical point (or a minimizer) $\hat{\underset{\sim}{\alpha}}$ is found for this functional, then $\hat{\underset{\sim}{b}}$ is obtained by replacing $\underset{\sim}{\alpha}$ by $\hat{\underset{\sim}{\alpha}}$ in (2.6). Finally, by hypothesis H-1, $\hat{\underset{\sim}{a}}$ is obtained as any solution of the system of nonlinear equations

$$(2.9) \qquad \underset{\sim}{g}(\underset{\sim}{a}) = \hat{\underset{\sim}{b}} .$$

The justification for employing this procedure is given by the following theorem:

Theorem 2.1 *Let* $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$ *and* $r_2(\underset{\sim}{\alpha})$ *be defined as above in the open set* $\Omega \subset R^k$, $\Phi(\underset{\sim}{\alpha})$ *has constant rank* $r \leq \min(m, n)$.

(a) *If* $\hat{\underset{\sim}{\alpha}}$ *is a critical point (or a global minimizer in* Ω*) of* $r_2(\underset{\sim}{\alpha})$, *and* $\hat{\underset{\sim}{a}}$ *satisfies:*

$$\underset{\sim}{g}(\hat{\underset{\sim}{a}}) = \Phi^+(\hat{\underset{\sim}{\alpha}})\,\underset{\sim}{y}, \tag{2.10}$$

then $\hat{\underset{\sim}{\alpha}}$ *is a critical point of* $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$ *(or a global minimizer for* $\underset{\sim}{\alpha} \in \Omega$*) and* $r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}}) = r_2(\hat{\underset{\sim}{\alpha}})$.

(b) *If* $(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$ *is a global minimizer of* $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$ *for* $\underset{\sim}{\alpha} \in \Omega$, *then* $\hat{\underset{\sim}{\alpha}}$ *is a global minimizer of* $r_2(\underset{\sim}{\alpha})$ *in* Ω *and* $r_2(\hat{\underset{\sim}{\alpha}}) = r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$. *Furthermore, if there is an unique* $\hat{\underset{\sim}{a}}$ *among the minimizing pairs of* $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$, *then* $\hat{\underset{\sim}{a}}$ *must satisfy* (2.10).

We shall postpone the proof of this Theorem until the end of Section 4, where we obtain a convenient expression for the gradient of the functional $r_2(\underset{\sim}{\alpha})$.

3. Algorithmia I. Residual calculation.

One of our main points in the algorithmic part of this paper is to emphasize, when possible and appropriate, the use of stable and efficient linear least squares techniques. Thus it is convenient to review some of the tools and introduce the necessary notation.

If Q is an orthogonal matrix then, for every vector $\underset{\sim}{z}$,

$$\|Q\underset{\sim}{z}\| = \|\underset{\sim}{z}\| .$$

It is well-known (cf. [8, 10, 18]) that every $m \times n$ matrix Φ ($m \geq n$) of rank $r \leq n$, can be orthogonally transformed into "triangular" form, viz., there exist Q, Z orthogonal, such that

$$(3.1) \qquad Q \Phi Z^T = \left[\begin{array}{c|c} \tilde{T} & 0 \\ \hline 0 & 0 \end{array}\right] \equiv T ,$$

where $\tilde{T}$ is an $r \times r$ upper triangular and nonsingular matrix. Then

$$\Phi^+ = Z^T T^+ Q = Z^T \left[\begin{array}{c|c} \tilde{T}^{-1} & 0 \\ \hline 0 & 0 \end{array}\right] Q ,$$

and consequently,

$$P_\Phi = \Phi\Phi^+ = Q^T \left[\begin{array}{c|c} I_r & 0 \\ \hline 0 & 0 \end{array}\right] Q ; \quad P_\Phi^\perp = Q^T \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I_{m-r} \end{array}\right] Q .$$

(Similarly, though we don't use it in our calculations:

$$_\Phi P = \Phi^+\Phi = Z^T \left[\begin{array}{c|c} I_r & 0 \\ \hline 0 & 0 \end{array}\right] Z ; \quad _\Phi P^\perp = Z^T \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I_{n-r} \end{array}\right] Z .)$$

Due to the isometric properties of the orthogonal transformation Q ,

the least squares problem can be expressed as

$$\min_{\underset{\sim}{b}} \| \underset{\sim}{y} - \Phi \underset{\sim}{b} \|^2 = \min_{\underset{\sim}{b}} \| Q\underset{\sim}{y} - Q\Phi \underset{\sim}{b} \|^2 .$$

Calling $\bar{\underset{\sim}{y}} = Q\underset{\sim}{y}$ and partitioning it as $\bar{y} = \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} \begin{matrix} \} r \\ \} (m-r) \end{matrix}$, we obtain

$$(3.2) \qquad \hat{\underset{\sim}{b}} = Z^T \begin{bmatrix} T^{-1}\bar{y}_1 \\ \hline 0 \end{bmatrix} .$$

A simple computation shows that:

$$(3.3) \qquad \| \underset{\sim}{y} - \Phi \hat{\underset{\sim}{b}} \|^2 = \| P_{\Phi}^{\perp} \underset{\sim}{y} \|^2 = \| \bar{\underset{\sim}{y}}_2 \|^2 .$$

Therefore, one can evaluate the nonlinear functional $r_2(\underset{\sim}{\alpha})$ of (2.8) for any value of $\underset{\sim}{\alpha}$ in the following way: First the orthogonal matrix $Q(\underset{\sim}{\alpha})$ that is used in the reduction of $\Phi(\underset{\sim}{\alpha})$ is determined; simultaneously, $\bar{\underset{\sim}{y}} = Q\underset{\sim}{y}$ is computed, and finally

$$(3.4) \qquad r_2(\underset{\sim}{\alpha}) = \| \bar{\underset{\sim}{y}}_2 \|^2$$

is evaluated.

For minimization techniques not requiring derivatives this is all that is needed. For iterative techniques using the gradient of the functional or the Jacobian of the residual. vector function $P_{\Phi(\underset{\sim}{\alpha})}^{\perp} \underset{\sim}{y}$, we shall provide in the next section formulas which will also be useful in the proof of Theorem 2.1.

4. Fréchet derivatives of pseudoinverses, projectors, and residual vectors.

In this section we develop formulas for the Fréchet derivative of the pseudoinverse of a matrix function. This leads to expressions for the derivatives of the associated orthogonal projectors, and for the residual vector function

$$(4.1) \qquad \underset{\sim}{r}_2(\underset{\sim}{\alpha}) = P^{\perp}_{\Phi(\underset{\sim}{\alpha})}\, \underset{\sim}{y} .$$

As an aid to those readers not familiar with these concepts, we observe that an m x n matrix function $A(\underset{\sim}{\alpha})$ is a nonlinear mapping between the linear space of parameters $\underset{\sim}{\alpha} \in \mathbb{R}^k$ and the space of linear transformations $\mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$. Consequently, $\mathbf{D}A(\underset{\sim}{\alpha})$ will be, for each $\underset{\sim}{\alpha}$, an element of $\mathcal{L}(\mathbb{R}^k, \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m))$. Thus, $\mathbf{D}A(\underset{\sim}{\alpha})$ could be interpreted as a tridimensional tensor, formed with k (m X n) matrices (slabs), each one containing the partial derivatives of the elements of A with respect to one of the variables α_i. Still in another way, each column in the k-direction is the gradient of the corresponding matrix element.

Since all dimensions involved are different, it will be always clear in the algebraic manipulations how the different vectors, matrices, and tensors interact.

Lemma 4.1. For any $\underset{\sim}{\alpha} \in \Omega$, an open set of $\mathbb{R}^k$, let $B(\underset{\sim}{\alpha})$ be an m x n full column rank matrix function, and $C(\underset{\sim}{\alpha})$ an n x m full row rank matrix function. If $B(\underset{\sim}{\alpha})$ and $C(\underset{\sim}{\alpha})$ are Fréchet differentiable in Ω, then

$$(4.2) \qquad \mathbf{D}(B^+) = -B^+ \mathbf{D}B\, B^+ + (B^T B)^{-1}\, \mathbf{D}B^T\, P^{\perp}_{B} ,$$

$$(4.3) \qquad \mathbf{D}(C^+) = -C^+ \mathbf{D}C\, C^+ + {}_{C}P^{\perp}\, \mathbf{D}C^T (CC^T)^{-1} .$$

proof. Since B has full column rank, then $B^+ = (B^T B)^{-1} B^T$, and

$$D(B^+) = D(B^T B)^{-1} B^T + (B^T B)^{-1} DB^T .$$

But,

$$D(B^T B)^{-1} = -(B^T B)^{-1} D(B^T B) (B^T B)^{-1} .$$

Therefore,

$$(4.4) \qquad D(B^+) = (B^T B)^{-1} [DB^T - D(B^T B) B^+] .$$

Developing $D(B^T B)$ and regrouping, we obtain (4.2). Since C^T has full column rank, (4.3) follows readily from (4.2). ∎

Since $P_{A(\alpha)} = AA+$, $P^{\perp}_{A(\alpha)} = I - AA+$, it follows that

$$(4.5) \qquad DP_A = D_A A^+ + A D(A^+) ,$$

and

$$(4.6) \qquad DP^{\perp}_A = -DP_A .$$

If $A(\alpha)$ has full column rank, then from (4.5) and Lemma 4.1 we obtain

$$(4.7) \qquad D_{P_A} P^{\perp}_A D_A A^+ + (P^{\perp}_A D_A A^+)^T .$$

Similarly, if A has full row rank:

$$(4.8) \qquad D_A P = A^+ D_A {}_A P^{\perp} + (A^+ D_A {}_A P^{\perp})^T .$$

We shall prove now that formulas (4.7) and (4.8) are valid in the rank deficient case. For this purpose we shall prove first an auxiliary Lemma, and then obtain the derivative of the pseudoinverse of an arbitrary matrix function.

Let $A(\alpha)$ be an m X n matrix function, Fréchet differentiable,

and with constant rank $r \leq \min(m, n)$, on an open set $\Omega \subset \mathbb{R}^k$. Let $B(\underset{\sim}{\alpha})$ be a maximal set of independent columns of $A(\underset{\sim}{\alpha})$ in Ω, and let $C = B^+A$. It is well-known (see, for instance, [16]): (1) C has full row rank, (2) $A = BC$, (3) $A^+ = C^+B^+$. Due to our hypothesis, $B(\underset{\sim}{\alpha})$ can be formed with the same columns of $A(\underset{\sim}{\alpha})$ on a neighborhood of every $\underset{\sim}{\alpha} \in \Omega$, Other useful relations that we shall use below are

$$AA^+ = P_A = BB^+ = P_B \; ; \; B^+P_A^\perp = 0 \; ;$$

$$CA^+ = B^+AA^+ = B^+ \; ; \; P_A^\perp B = 0 \; ; \; P_A^\perp A^{+T} = 0 \; .$$

Lemma 4.2. With A, B, and C defined as above, the following formula is valid in Ω :

$$(4.9) \qquad B\,\mathbf{D}\,B^+ \; P_A^\perp = (\mathbf{D}A\,A^+)^T P_A^\perp \; .$$

Proof: From Lemma 4.1 we get

$$B\,\mathbf{D}\,B^+ = -P_B\,\mathbf{D}B\;B^+ + (\mathbf{D}B\;B^+)^T P_B^\perp$$

$$= -P_A\,\mathbf{D}B\;B^+ + (\mathbf{D}B\;B^+)^T P_A^\perp \; .$$

Therefore,

$$(4.10) \qquad B\mathbf{D}\,B^+ \; P_A^\perp = (DB\;B^+)^T P_A^\perp \; ,$$

On the other hand, since $A = BC$,

$$D_A\;A^+ = D_B\;CA^+ + B\mathbf{D}C\;A^+ = D_B\;B^+ + BDC\;A^+ \; .$$

Thus,

$$P_A^{\perp} \mathbf{D}A \; A^{+} = P_A^{\perp} \mathbf{D}B \; B^{+} ,$$

or

$$(\mathbf{D}A \; A^{+})^{T} \; P_A^{\perp} = (\mathbf{D}B \; B^{+})^{T} \; P_A^{\perp} ,$$

and this last expression together with (4.10) proves the Lemma.∎

Theorem 4.3. Let $\Omega \subset \mathbb{R}^k$ be an open set and for $\underset{\sim}{\alpha} \in \Omega$ let $A(\underset{\sim}{\alpha})$ be an $m \times n$ Fréchet differentiable matrix function having fixed rank $r \leq \min(m, n)$. Then for any $\underset{\sim}{\alpha} \in \Omega$:

$$(4.11) \quad \mathbf{D} \; A+ = -A^{+}\mathbf{D}A \; A^{+} + A^{+}A^{+T}\mathbf{D}A^{T} \; P_A^{\perp} + {}_AP^{\perp}\mathbf{D}A^{T} \; A+ \; {}^{T}A+ .$$

proof: With B and C as above, we have that

$$\mathbf{D} \; A^{+} = \mathbf{D}(C^{+}B^{+}) = \mathbf{D}C^{+} \; B^{+} + C^{+}\mathbf{D}B^{+} ,$$

and hence by (4.3)

$$\mathbf{D} \; A^{+} = -C^{+}\mathbf{D}C \; C^{+}B^{+} + {}_CP^{\perp}\mathbf{D}C^{T} \; C^{+}B^{+} + C^{+}\mathbf{D}B^{+} ,$$

since

$$(CC^{T})^{-1} = C^{+T} \; C^{+} .$$

Substituting

$$\mathbf{D}C = \mathbf{D}(B^{+}A) = \mathbf{D}B^{+} \; A + B^{+}\mathbf{D}A , \; C^{+}B^{+} = A^{+} , \; {}_CP^{\perp} = {}_AP^{\perp},$$

in the last expression we get:

$$(4.12)\quad \mathbf{D}A^{+} = -A^{+}\mathbf{D}A\,A^{+} + C^{+}\mathbf{D}B^{+} - C^{+}\mathbf{D}B^{+}\,AA^{+} + {}_{A}P^{\perp}\,\mathbf{D}C^{T}\,C^{+T}A^{+}$$

$$= -A^{+}\mathbf{D}A\,A^{+} + C^{+}\mathbf{D}B^{+}\,P_{A}^{\perp} + {}_{A}P^{\perp}\,\mathbf{D}C^{T}\,C^{+T}A^{+}$$

But,

$$(4.13)\quad {}_{A}P^{\perp}\,\mathbf{D}C^{T}\,C^{+T}A^{+} = {}_{A}P\mathbf{D}A^{T}\,A^{+T}A^{+} + {}_{A}P\,A^{T}\,\mathbf{D}B^{+T}C^{+T}A^{+}$$

$$= {}_{A}P\mathbf{D}A^{T}\,\tilde{A}^{+T}A+ ,$$

since ${}_{A}PA^{T} = 0$.

Substituting (4.13) into (4.12) and using the relationship $C^{+}\mathbf{D}B^{+}\,P_{A} = A^{+}B\mathbf{D}B^{+}\,P_{A} = A^{+}A^{+T}\mathbf{D}\,A^{T}\,P_{A}$, given by Lemma 4.2, we finally obtain the desired result. ∎

Corollary 4.3. Let $A(\underset{\sim}{a})$ be as in Theorem 4.2. Then, for any $\underset{\sim}{\alpha} \in \Omega$.

$$(4.14a)\quad \mathbf{D}P_{A} = \mathbf{D}(AA^{+}) = P_{A}^{\perp}\mathbf{D}A\,A^{+} + (P_{A}^{\perp}\mathbf{D}A\,A^{+}) ,$$

$$(4.14b)\quad \mathbf{D}_{A}P = \mathbf{D}(A^{+}A) = A^{+}\mathbf{D}A\,{}_{A}P^{\perp} + (A^{+}\mathbf{D}A\,{}_{A}P^{\perp})^{T} .$$

Proof: Obvious. I

From this result it is now easy to derive an expression for the gradient of the functional $r_{2}(\underset{\sim}{\alpha})$ (see (2.8)), provided the matrix $\Phi(\underset{\sim}{\alpha})$ has constant rank on an open neighborhood of the point in which the gradient is calculated. In fact:

$$(4.15)\quad r_{2}(\underset{\sim}{\alpha}) = \| P_{\Phi(\underset{\sim}{\alpha})}^{\perp}\,\underset{\sim}{y} \|^{2} = \langle P^{\perp}\underset{\sim}{y} , P^{\perp}\underset{\sim}{y} \rangle ,$$

and

$$\frac{1}{2}\nabla r_2(\underset{\sim}{\alpha}) = -\underset{\sim}{y}^T P^{\perp} \left[P^{\perp} D\Phi \ \Phi^{+} + \Phi^{+T} D\Phi^T \ P^{\perp}\right]\underset{\sim}{y} .$$

Since $P^{\perp}_{\Phi}\Phi^{+T} = 0$, we finally obtain:

$$\frac{1}{2}\nabla r_2(\underset{\sim}{\alpha}) = \underset{\sim}{y}^T P^{\perp}_{\Phi} D\Phi \ \Phi^{+}\underset{\sim}{y} . \tag{4.16}$$

Now we have the elements for proving Theorem 2.1.

Proof of Theorem 2.1.

From (2.2) we have that $r(\underset{\sim}{a}, \underset{\sim}{\alpha}) = \| \underset{\sim}{y} - \Phi(\underset{\sim}{\alpha})g(\underset{\sim}{a})\|^2$.

Therefore,

$$\frac{1}{2}\nabla r(\underset{\sim}{a}, \underset{\sim}{\alpha}) = -(\underset{\sim}{y} - \Phi g)^T \ (D\Phi \ g + \Phi D g) . \tag{4.17}$$

Assume now that $\hat{\underset{\sim}{\alpha}}$ is a critical point of $r_2(\underset{\sim}{\alpha})$, and that $\hat{\underset{\sim}{a}}$ satisfies

$$\underset{\sim}{g}(\hat{\underset{\sim}{a}}) = \Phi^{+}(\hat{\underset{\sim}{\alpha}})\underset{\sim}{y} . \tag{4.18}$$

Then,

$$\frac{1}{2}\nabla r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}}) = -(P^{\perp}_{\Phi}\underset{\sim}{y})^T \ (D\Phi \ \Phi^{+}\underset{\sim}{y} + \Phi D\underset{\sim}{g}) \tag{4.19}$$

$$= \frac{1}{2}\nabla r_2(\hat{\underset{\sim}{\alpha}}) ,$$

since $\underset{\sim}{y}^T P^{\perp}_{\Phi}\Phi D\underset{\sim}{g} = \underset{\sim}{0}$. Thus $(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$ is a critical point of $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$,

Assume now that $\hat{\underset{\sim}{\alpha}}$ is a global minimizer of $r_2(\underset{\sim}{\alpha})$ in Ω , and $\hat{\underset{\sim}{a}}$ satisfies (4.18). Then clearly, $r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}}) = r_2(\hat{\underset{\sim}{\alpha}})$. Assume that there exists $(\underset{\sim}{a}^*, \alpha^*)$, $\underset{\sim}{\alpha}^* \in \Omega$, such that $r(\underset{\sim}{a}^*, \underset{\sim}{\alpha}^*) < r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$. Since for any $\underset{\sim}{\alpha}$ we have $r_2(\underset{\sim}{\alpha}) \leq r(\underset{\sim}{a}, \underset{\sim}{\alpha})$, then it follows that $r_2(\underset{\sim}{\alpha}^*) \leq r(\underset{\sim}{a}^*, \underset{\sim}{\alpha}^*) < r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}}) = r_2(\hat{\underset{\sim}{\alpha}})$, which is a contradiction to the fact that $\underset{\sim}{\alpha}$ was a global

minimizer of $r_2(\underset{\sim}{\alpha})$ in Ω . Therefore $(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$ is a global minimizer of $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$ in Ω and part (a) of the Theorem is proved.

Conversely, suppose that $(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$ is a global minimizer of $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$ in Ω , then as above

$$r_2(\hat{\underset{\sim}{\alpha}}) \leq r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}}) .$$

Now let $\underset{\sim}{a}^*$ be a solution of $g(\underset{\sim}{a}) = \Phi^+(\hat{\underset{\sim}{\alpha}})y$.

Then we have

$$r_2(\hat{\underset{\sim}{\alpha}}) = r(\underset{\sim}{a}^*, \hat{\underset{\sim}{\alpha}}) \leq r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}}) ,$$

but since $(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$ was a global minimizer we must have equality. If there was an unique $\underset{\sim}{a}$ among the minimizers of $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$, then $\underset{\sim}{a}^* \equiv \hat{\underset{\sim}{a}}$. We still have to show that $\hat{\underset{\sim}{\alpha}}$ is a global minimizer of $r_2(\underset{\sim}{\alpha})$. Assume that it is not. Thus, there will be $\bar{\underset{\sim}{\alpha}} \in \Omega$, such that $r_2(\bar{\underset{\sim}{\alpha}}) < r_2(\hat{\underset{\sim}{\alpha}})$. Let $\bar{\underset{\sim}{a}}$ be a solution of $g(\underset{\sim}{a}) = \Phi^+(\bar{\underset{\sim}{\alpha}})y$. Then $r_2(\bar{\underset{\sim}{\alpha}}) = r(\bar{\underset{\sim}{a}}, \bar{\underset{\sim}{\alpha}}) < r_2(\hat{\underset{\sim}{\alpha}}) = r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$, which is a contradiction to the fact that $(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$ was a global minimizer of $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$. I

5. Algorithmia II. Detailed implementation of the Gauss-Newton-Marquardt algorithm.

We shall now explain in detail how to apply the results of Section 4 to the Marquardt modification of the Gauss-Newton iterative procedure; we make extensive use of linear least squares techniques. We shall include an economical implementation of the Marquardt algorithm devised earlier by Golub (see also [11,14]).

We define the vector

$$\underset{\sim}{r}_2(\underset{\sim}{\alpha}) = P^{\perp}_{\Phi(\underset{\sim}{\alpha})}\underset{\sim}{y} .$$

The generalized Gauss-Newton iteration-with step control for the nonlinear least squares problem

$$(5.1) \qquad \min_{\underset{\sim}{\alpha}} r_2(\underset{\sim}{\alpha}) = \min_{\underset{\sim}{\alpha}} \| \underset{\sim}{r}_2(\underset{\sim}{\alpha}) \| = \min_{\underset{\sim}{\alpha}} \| P^{\perp}_{\Phi(\underset{\sim}{\alpha})}\underset{\sim}{y} \|$$

is given by

G.N. Starting from an arbitrary $\underset{\sim}{\alpha}^0$:

$$(5.2) \qquad \underset{\sim}{\alpha}^{\ell+1} = \underset{\sim}{\alpha}^{\ell} - t_{\ell}[D\underset{\sim}{r}_2(\underset{\sim}{\alpha}^{\ell})]^{+}\underset{\sim}{r}_2(\underset{\sim}{\alpha}^{\ell}) , \ (\ell=0,\ldots) .$$

The parameters $t_{\ell} > 0$, which control the size of the step, are used to prevent divergence. Usually $t_a = 1$, unless $r_2(\underset{\sim}{\alpha}^{\ell+1}) > r_2(\underset{\sim}{\alpha}^{\ell})$, in which case t_{ℓ} is reduced. Another use of the parameters t_{ℓ} is to minimize $r_2(\underset{\sim}{\alpha}^{\ell+1})$ along the direction $[D\underset{\sim}{r}_2(\underset{\sim}{\alpha}^{\ell})]^{+}\underset{\sim}{r}_2(\underset{\sim}{\alpha}^{\ell})$.

Marquardt's modification calls for the introduction of a sequence of non negative auxiliary parameters $\nu_{\ell} > 0$.

G.N.M. Define

$$K_{\ell} \equiv \begin{bmatrix} D\underset{\sim}{r}_2(\underset{\sim}{\alpha}^{\ell}) \\ \hline \nu_{\ell}F_{\ell} \end{bmatrix} , \quad \underset{\sim}{r}_{\ell} \equiv \left.\begin{bmatrix} \underset{\sim}{r}_2(\underset{\sim}{\alpha}^{\ell}) \\ \hline 0 \end{bmatrix}\right\} n ,$$

where for each ℓ, F_{ℓ} is the upper triangular Cholesky factor of an $n \times n$

symmetric positive definite matrix M_ℓ. Then the Gauss-Newton-Marquardt iteration is given by

$$\underset{\sim}{\alpha}^{\ell+1} = \underset{\sim}{\alpha}^{\ell} - K_\ell^{+} \underset{\sim}{r}_\ell , \qquad \ell \geq 0 .$$

Reasons for this modification are well-known. For more details and an interesting study of the convergence of this method we refer to [14]. We wish to make explicit now the "two-stage orthogonal factorization" given in [11] and [14], in order to show how to take advantage of the special structure of the problem.

Calling $\underset{\sim}{h} = \underset{\sim}{\alpha}^{\ell+1} - \underset{\sim}{\alpha}^{\ell}$, $DP = D\underset{\sim}{r}_2(\underset{\sim}{\alpha}^{\ell}) = DP_{\mathcal{L}}^{\perp}$

and dropping the superscript ℓ from here on in, one step of the Marquardt algorithm is equivalent to solving the linear least squares problem

$$K\underset{\sim}{h} \cong \left[\begin{array}{c} -\underset{\sim}{r}_2(\underset{\sim}{\alpha}) \\ \hline \underset{\sim}{0} \end{array}\right] .$$

In the first stage of the orthogonal factorization of K an $m \times n$ orthogonal matrix Q is chosen so that

$$Q\,DP = R_1 = \left[\begin{array}{c} \text{upper triangular } n \\ \hline 0 \end{array}\right] \equiv \left[\begin{array}{c} R_1' \\ \hline 0 \end{array}\right] .$$

Thus,

$$\left[\begin{array}{c|c} Q & 0 \\ \hline 0 & I_n \end{array}\right] \cdot \left[\begin{array}{c} DP \\ \hline \nu F \end{array}\right] \equiv Q_1 \left[\begin{array}{c} DP \\ \hline \nu F \end{array}\right] = \left[\begin{array}{c} R_1 \\ \hline \nu F \end{array}\right] ,$$

$$Q_1 \left[\begin{array}{c} -\underset{\sim}{r}_2(\underset{\sim}{\alpha}) \\ \hline \underset{\sim}{0} \end{array}\right] \equiv \left[\begin{array}{c} \bar{\underset{\sim}{r}} \\ \hline \underset{\sim}{0} \end{array}\right] .$$

R_1' and $\bar{\underset{\sim}{r}}$ are saved for future use.

In the second stage we choose an $(m+n)\times(m+n)$ orthogonal matrix Q_2 to reduce

$$A \equiv \begin{bmatrix} R_1' \\ \hline \nu F \end{bmatrix}$$ to "triangular" form.

For this purpose we shall use successive Householder transformations as in [3], from where we adopt the notation.*

On reducing the first column of A, which is of the form:

$$\underset{\sim}{a}_1^{(1)} = \begin{bmatrix} a_{11}^{(1)} \\ 0 \\ \vdots \\ 0 \\ \hline \mu \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Bigg\} m \quad , \qquad \mu = \nu f_{11}$$

we use $Q^{(1)} = I - \beta_1 \underset{\sim}{u}^{(1)} \underset{\sim}{u}^{(1)T}$,

where

$$u_1^{(1)} = \text{sign}\,(a_{11}^{(1)})(\sigma_1 + |a_{11}^{(1)}|) \ ,$$

$$\sigma_1 = (a_{11}^2 + \mu^2)^{\frac{1}{2}} \ ,$$

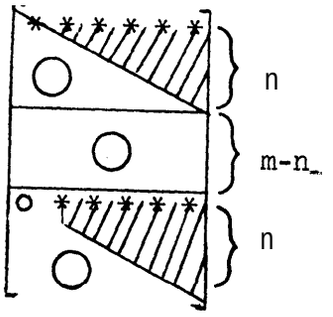
$$u_{m+1}^{(1)} = \mu \ ,$$

$$u_1^{(1)} = 0 \ , \text{ otherwise,}$$

$$\beta_1 = 2/\underset{\sim}{u}^{(1)T} \underset{\sim}{u}^{(1)} \ .$$

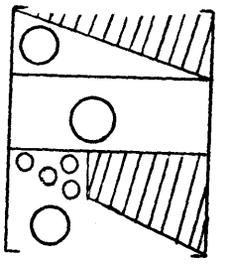
Now we observe that when $Q^{(1)}$ is applied to a vector, any component corresponding to a zero component of $\underset{\sim}{u}^{(1)}$ is left unchanged. In particular,

the band of zeros in A is preserved. Thus, in this first step we only need to transform the elements of rows number 1 and m+1 . Consequently, $A^{(2)} = Q^{(1)}A$ will have the schematic form:

$A^{(2)} =$ 

where the asterisks indicate the modified elements.

It is now clear that at step k , $A^{(k)}$ will have the form

$A^{(k)} =$ 

The matrix $A^{(k+1)}$, k= 1,...,n , is obtained as follows:

i) $ok = \left((a_{kk}^{(k)})^2 + \sum_{i=1}^{k} (a_{m+i,k}^{(k)})^2 \right)^{\frac{1}{2}}$,

ii) $\beta_k = (\sigma_k(\sigma_k + |a_{kk}^{(k)}|))^{-1}$,

iii) $u_i^{(k)} = 0$ for $i<k$, $k+1 \leq i \leq m$, $m+k<i$;

$u_k^{(k)} = \text{sign}\,(a_{kk}^{(k)})(\sigma_k + |a_{kk}^{(k)}|)$;

$u_i^{(k)} = a_i^{(k)}$, $m+1 \leq i \leq m+k$.

iv) $y^{I} = \beta_k \underset{\sim}{u}^{(k)T} A^{(k)}$,

$$y_j = \beta_k [u_k^{(k)} a_{kj}^{(k)} + \sum_{i=1}^{k} a_{m+i,k}^{(k)} a_{m+i,j}^{(k)}] \quad , \quad j=k+1,\ldots,n \ .$$

Finally,

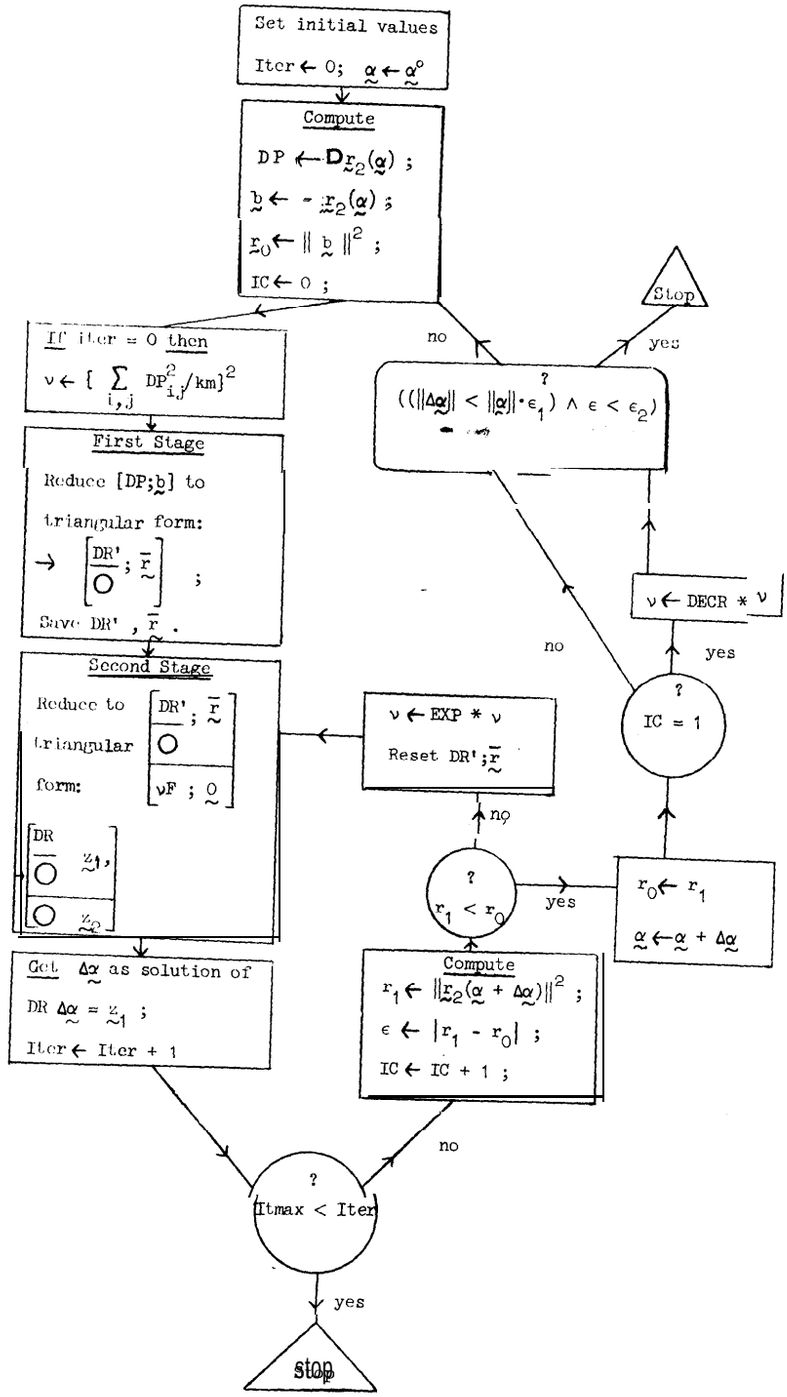
v) $a_{ij}^{(k+1)} = a_{ij}^{(k)} - u_1^{(k)} y_j$, $\quad i=k;\ m+1,\ldots,m+k;\quad j=k+1,\ldots,n;$

$a_{kk}^{(k+1)} = - \text{sign}\,(a_{kk}^{(k)})\sigma_k$.

These formulas are similar to those given in [3], but are modified to take advantage of the structure of the matrix A .

Osborne's version of Marquardt's algorithm, modified for our present problem, is presented in the detailed flow-chart of Figure 1. The parameters DECR and EXP are the factors by which ν is either decreased or increased.

Figure: Marquardt's algorithm (Osborne [14])



We will evaluate $\mathbf{D}\underset{\sim}{r}_2(\alpha) = \mathbf{D}P^{\perp}_{\Phi(\alpha)}\underset{\sim}{y}$ for a given α, according to

$$\mathbf{D}P^{\perp}_{\Phi(\alpha)}\underset{\sim}{y} = -(P^{\perp}_{\Phi}\ \mathbf{D}\,a\,)\ \Phi^{+}\underset{\sim}{y} - \Phi^{+T}\,(P^{\perp}_{\Phi}\mathbf{D}\Phi)^{T}\underset{\sim}{y}\,, \tag{5.4}$$

which is readily obtained from (4.14a).

In many applications, each component function φ_j depends only upon a few of the parameters $\{\alpha_t\}_{t=1}^{k}$, and therefore its derivatives with respect to the other parameters will vanish. Those vanishing derivatives will produce m-columns of zeros in the tensor $\mathbf{D}\Phi$. In order to avoid a waste of storage and useless computation with zeros it is convenient to introduce from the outset the k x n incidence matrix $E = (e_{jt})$. This matrix will be defined as follows:

$e_{jt} = 1$ iff parameter α_t appears in function φ_j ;

$e_{jt} = 0$ otherwise.

We shall also call p the number of nonzero derivatives in $\mathbf{D}\Phi$: $p = \sum_{t,j} e_{jt}$. The nonzero derivative vectors can then be stored sequentially in a bidimensional array $B(m\times p)$. In our implementation we chose to store the nonzero m-columns varying first the index corresponding to the different differentiations, and then that corresponding to the different functions. This information can then be decoded for use in algebraic manipulations by means of the incidence matrix E ,

We now introduce some notation in order to describe the compressed storage of the nonzero columns of the tensor $\mathbf{D}\Phi$ in a more explicit fashion. We define, for t= l,...,k ,

$$S_t = \{\text{ set of ordered indices for which } e_{jt} \neq 0\ ,\ j=1,\ldots,n\};$$

$$\psi_{tj}(\underset{\sim}{\alpha}) = \frac{\partial \varphi_j(\underset{\sim}{\alpha})}{\partial \alpha_t} \quad , \qquad j=1,\ldots,n \ , \quad t=1,\ldots,k \ .$$

We write the matrix B in partitioned form

$$B = [B_1, B_2, \ldots, B_k] \ ,$$

where

$$B_t = [\psi_{tj_1}, \psi_{tj_2}, \ldots, \psi_{tj_t}]_{j_i \in S_t} \ .$$

A step-by-step description of the computation of $\mathbf{D}P^{\perp}_{\Phi}\underset{\sim}{y}$ follows. We assume that the rank of $\Phi(\underset{\sim}{\alpha})$ is computationally determined and equal to $r \leq \min(m,n)$.

a) Compute $\Phi(\underset{\sim}{\alpha})$, $\mathbf{D}\Phi(\underset{\sim}{\alpha})$.

b) Form the m x (n+p+1) array

$$G \equiv [\Phi(\underset{\sim}{\alpha}) \ ; \ \underset{\sim}{y} \ ; \ \mathbf{D}\Phi(\underset{\sim}{\alpha})] = [A \ ; \ \underset{\sim}{y} \ ; \ B] \ .$$

c) Obtain the complete orthogonal factorization of A (cf. Section 3):

$$QAZ^T = T = \left[\begin{array}{c|c} \tilde{T} & 0 \\ \hline 0 & 0 \end{array}\right] ; \quad \tilde{T} = \left[\begin{array}{c} \\ \\ \\ \end{array}\right]_{r \times r} \ .$$

Also $\underset{\sim}{v} = Q\underset{\sim}{y}$; $C = QB$

($\tilde{T}$, $\underset{\sim}{v}$, and C will be stored in the array G). Note again that (see Section 3):

$$P^{\perp}_{\Phi(\underset{\sim}{\alpha})} = Q^T \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I_{m-r} \end{array}\right] Q \ .$$

d) Get the intermediary values:

$$\bar{D} = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I_{m-r} \end{array}\right] \cdot C$$

(i.e., Remember that the nonzero information of $\bar{D}$ is stored in the last p columns and last m-r rows of G);

$$\underset{\sim}{x} = A^{+}\underset{\sim}{y} = Z^{T}\left[\begin{array}{c} \tilde{T}^{-1}\underset{\sim}{v}_1 \\ \hline \underset{\sim}{0} \end{array}\right] \quad \left(\underset{\sim}{v} = \left[\begin{array}{c} \underset{\sim}{v}_1 \\ \hline \underset{\sim}{v}_2 \end{array}\right] \begin{array}{l} \} r \\ \} m-r \end{array}\right).$$

e) $$\underset{nxk}{U} = (P^{-1}D\Phi)^{T}\underset{\sim}{y} = D\Phi^{T}\, Q^{T}\left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I_{m-r} \end{array}\right] Q\underset{\sim}{y} = \bar{D}^{T}\underset{\sim}{v} = C^{T}\left[\begin{array}{c} \underset{\sim}{0} \\ \hline \underset{\sim}{v}_2 \end{array}\right]$$

(transposition in the tensor $D\Phi$ refers to transposition within the "slabs" corresponding to the different derivatives, and must be interpreted adequately when decoding the information from the compressed storage array G ; the appropriate ALGOL-60 code for computing U with our storage convention would be (assuming that C = QB is stored in the same place B is :

```
n1 ← n+1 ;
L ← n1 ;
for t ← 1 step 1 until k do
for j ← 1 step 1 until n do
if E[j,t] = 0 then U[j,t] ← 0 else
begin L ← L+1 ; acum ← 0 ;
for i ← n1 step 1 until m do
acum ← acum + G[i,L] X G[i,n1] ;
U[j,t] ← acum
end ; ) .
```

f) Compute $S = Z \cdot U$.
nxk

Solve the k , rxr lower triangular systems:

$\tilde{T}^{T}W = \tilde{S}$, where $\underset{rxk}{\tilde{S}}$ contains the first r rows of S .

Store W in the first r rows of the mxk array B . Compute $\bar{D}\otimes\underset{\sim}{x}$ and store the nonzero information in the last m-r rows of B .

g) Finally, the mxk matrix B is obtained as:

$$B \leftarrow D P^{-1}_{\Phi(\underset{\sim}{\alpha})} \underset{\sim}{y} = \underset{m\times m}{-Q^{T}} \left\{ \begin{bmatrix} W \\ \hline O \end{bmatrix} + \bar{D}\otimes\underset{\sim}{x} \right\} = -Q^{T} B \quad .$$

We emphasize the systematic use made of the triangular orthogonal decomposition of the matrix $\Phi(\underset{\sim}{\alpha})$. We also warn the reader about the correct interpretation of the algebraic operations in which any tridimensional tensor intervene, as we exemplified in (e) .

6. Numerical experiments.

We have implemented three different algorithms based on the developments of the previous sections for the case $\underset{\sim}{g}(\underset{\sim}{a}) = \underset{\sim}{a}$ and rank $\Phi = n$.

The methods minimize the variable projection functional $r_2(\underset{\sim}{\alpha}) = \|P^{\perp}_{\Phi(\alpha)} y\|^2$ first, in order to obtain the optimal parameters $\hat{\underset{\sim}{\alpha}}$, and then complete the optimization according to our explanation in Section 2. The algorithms differ in the procedure used for the minimization of $r_2(\underset{\sim}{\alpha})$.

A1. Minimization without derivatives. We use PRAXIS, a FORTRAN version of a 'program developed by R. Brent [4], who very kindly made it available to us. All that PRAXIS essentially requires from the user is the value of the functional for any $\underset{\sim}{\alpha}$. This is computed using the results of Section 3. In fact, the user has only to give code for filling the matrix Φ for any $\underset{\sim}{\alpha}$, and our program will effect the triangular reduction and so on. It turns out that many times (see the examples) the models have some terms which are exclusively linear, i.e., functions φ_j which are independent of $\underset{\sim}{\alpha}$. Those functions produce columns in $\Phi(\underset{\sim}{\alpha})$ which are constant throughout the process. If they are considered first, then it is possible to reduce them once and for all, saving the repetition of computation. This is done in our program.

A2. Minimization by Gauss-Newton with control of step (see (5.2)).

The user is required to provide the incidence matrix E and the array of functions φ_j and non-vanishing partial derivatives: G. See Section 5 for a more detailed description.

A3. Minimization by Marquardt's modification, as explained in Section 5 with

$F_\ell \equiv I$. User supplied information is the same as in A2.

Test problems. Problems 1 and 2 are taken from Osborne [14], where the necessary data can be found.

P1. Exponential fitting. The model is of the form:

$$\eta_1(\underset{\sim}{a}, \underset{\sim}{\alpha}; t) = a_1 + a_2 e^{-\alpha_1 t} + a_3 e^{-\alpha_2 t} .$$

The functions φ_i are obviously $\varphi_1(\underset{\sim}{\alpha}; t) \equiv 1$, $\varphi_{j+1}(\underset{\sim}{\alpha}; t) = e^{-\alpha_j t}$, $j=1,2$.

So the different constants, in the notation of Section 2 are: $n=3$, $s=3$, $k=2$. For the problem considered, $m=33$. The number of constant functions: NCF = 1 . The number of non-vanishing partial derivatives: $p=2$.

In Table I we compare our results for methods A1, A2, A3, and those obtained by minimizing the full functional $r(\underset{\sim}{a}, \underset{\sim}{\alpha})$.

P2. Fitting Gaussians with an exponential background.

$$\eta_2(\underset{\sim}{a}, \underset{\sim}{\alpha}; t) = a_1 e^{-\alpha_1 t} + a_2 e^{-\alpha_2 (t-\alpha_5)^2} + a_3 e^{-\alpha_3 (t-\alpha_6)^2} + a_4 e^{-\alpha_4 (t-\alpha_7)^2} .$$

The functions φ_j are:

$$\varphi_1(\underset{\sim}{\alpha}; t) = e^{-\alpha_1 t} ; \quad \varphi_j(\underset{\sim}{\alpha}; t) = e^{-\alpha_j (t-\alpha_{j+3})^2} , \quad j=2,3,4 .$$

Thus: $n=4$, $s=4$, $k=7$, $m=65$, $p=7$.

Results for this problem appear in Table II.

P3. Iron Mössbauer Spectrum with two sites of different electric field gradient and one single line [21].

The model here is the following:

$$h_3(\underset{\sim}{a}, \underset{\sim}{\alpha}; t) = a_1 + a_2 t + a_3 t^2$$

$$+ 4\left[\frac{1}{1 + \left(\frac{\alpha_1 + 0.5\alpha_2 - t}{\alpha_3}\right)^2} + \frac{1}{1 + \left(\frac{\alpha_1 - 0.5\alpha_2 - t}{\alpha_3}\right)^2}\right]$$

$$- a_5\left[\frac{1}{1 + \left(\frac{\alpha_4 + 0.5\alpha_5 - t}{\alpha_6}\right)^2} + \frac{1}{1 + \left(\frac{\alpha_4 - 0.5\alpha_5 - t}{\alpha_6}\right)^2}\right]$$

$$- a_6\left[\frac{1}{1 + \left(\frac{\alpha_7 - t}{\alpha_8}\right)^2}\right] .$$

Clearly, $\varphi_j(\underset{\sim}{\alpha}; t) = t^j$, $j=1,2,3$; and φ_4 , φ_5 , φ_6 are the functions inside the square brackets.

Here: $n=6$, $k=8$, $NCF=3$, $p=8$, $m=188$, $s=6$.

For this example we wish to thank Dr. J. C. Travis of NBS who kindly supplied the problem and results from his own computer program.

Comparisons are offered in Table III.

The qualitative behavior of the three different minimization procedures used in our computation follows the pattern that have been expounded in recent comparisons (Bard [1]). Gauss-Newton is fastest whenever it converges from a good initial estimate. As is shown in the fitting of Gaussians (Table II), if the problem is troublesome, then a more elaborate strategy is called for. Brent's program has the advantage of not needing derivatives, which in this case leads to a big simplification. On the other hand, it is a very conservative program which really tries to obtain rigorous results. This, of course, can lead to a long search in cases where it is not entirely justified.

As a consequence of our Theorem 2.1, and of our numerical experience, we strongly recommend, even in the case when our procedure is not used, to obtain

initial values for the linear parameters when $g_j(\underset{\sim}{a}) = a_j$ by setting $\underset{\sim}{a}^\circ = \Phi^+(\underset{\sim}{\alpha}^\circ)y$. This is done in our program for the full functional and in the program of Travis with excellent results.

The computer times shown in Table I and Table II correspond to the CPU times (execution of the object code) on an IBM 360/50. All calculations were performed in long precision; viz. 14 hexadecimal digits in the mantissa of each number. We compare the results of minimizing the reduced functional when the Variable Projection (VP) technique is used with that of minimizing the full functional (FF) for various minimization algorithms. In order to eliminate the coding aspect, we have used essentially the same code for minimizing the two functionals. The only difference was in the subroutine DPA which computes in both cases the Jacobian of the residual vector.

In the FF approach, the subroutine DPA computed the m x (n+k) matrix B as follows: the first n columns consisted of the vectors $\underset{\sim}{\varphi}_j(\underset{\sim}{\alpha})$ while the remaining columns were the partial derivatives

$$\frac{\partial}{\partial \alpha_\ell}(y - \Phi(\underset{\sim}{\alpha})\underset{\sim}{a}) = -\sum_{j=1}^{n} a_j \frac{\partial \underset{\sim}{\varphi}_j(\underset{\sim}{\alpha})}{\partial \alpha_\ell}, \quad (\ell=1,2,\ldots,k) \quad .$$

These derivatives were constructed using the same information provided by the user subroutine ADA. We also obtained from DPA in the FF case, the automatic initialization of the linear parameters, viz. $\underset{\sim}{a}^\circ = \Phi^+(\underset{\sim}{\alpha}^\circ)\underset{\sim}{y}$.

For the numerical examples given here, the cost per iteration was somewhat higher for the VP functional. However, we see that in some cases there has been a dramatic decrease in the number of iterations; this has been observed previously (cf. [12]). Thus, in these cases the total computing time is much more favorable for the VP approach. This was especially true for all three

methods of minimization when the exponential fit was made and when Marquardt's method was used in the Mössbauer spectrum problem,

For the Mossbauer spectrum problem, we used two sets of initial values. We used those given by Travis [21], (say) $\underset{\sim}{\beta}^{\circ}$, and also $\underset{\sim}{\hat{\beta}}^{\circ} \approx \beta^{\circ} \pm 0.05\, \underset{\sim}{\beta}^{\circ}$. For $\underset{\sim}{\beta}^{\circ}$, the value of the functional is 3.04467×10^{8} while for $\underset{\sim}{\hat{\beta}}^{\circ}$, the value of the functional is 6.405×10^{8}; the final estimates of the parameters yielded a residual sum of squares less than 3.0444×10^{8}. When Brent's method was used on the full functional, the method did not seem to converge, but for the reduced functional, Brent's method converged reasonably well, In fact, after twenty minutes Brent's algorithm applied to the full functional with $\underset{\sim}{\beta}^{\circ}$ did not achieve the desired reduction in the functional.

The results we have obtained in minimizing the full functional for the first two problems using the Marquardt method, and those of problem 3 with Newton's method and $\underset{\sim}{\beta}^{\circ}$, are consistent with the results reported by Osborne and Travis.

From a rough count of the number of arithmetic operations (function and derivative evaluation per step are the same for both procedures, so that the work they do can be disregarded), it seems that for almost no combination of the parameters (m, n, k, p) the VP procedure will require fewer operations per iteration than the FF procedure. It is an open problem then to determine a priori under what conditions the VP procedure will converge more quickly than the FF procedure when minimization algorithms using derivatives are used.

Another important problem is that of stability, The numerical stability of the process and of the attained solution must be studied. By insisting on the use of stable linear techniques, we have tried to achieve an overall numerically stable procedure for this nonlinear situation. Since the standards

of stability for non-linear problems are ill-defined at' this time, it is hard to say whether we have succeeded in obtaining our goal.

Table I

Exponential fit.

Method	Functional	Number of Function Evaluations	Number of Derivative Evaluations	Time (seconds)
A_1	FF	1832	—	191.00
	VP	100	—	9.00
A_2	FF	11	11	5.05
	VP	4	4	3.20
A_3	FF	32	26	12.55
	VP	4	4	3.12

$r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$, $r_2(\hat{\underset{\sim}{\alpha}}) \leq 0.5465 \times 10\text{-}4$

Table II

Gaussian fit.

Method	Functional	Number of Function Evaluations	Number of Derivative Evaluations	Time (seconds)
A_3	FF	11	9	23.35
	VP	10	8	26.82

$r(\hat{\underset{\sim}{a}}, \hat{\underset{\sim}{\alpha}})$, $r_2(\hat{\underset{\sim}{\alpha}}) \leq 0.048$

Methods A_1 and A_2 were either slowly convergent or non-convergent.

Table III

Mössbauer Iron Spectrum.

Method	Functional	Initial Values	Number of Function Evaluations	Number of Derivative Evaluations	Time (seconds)
A_1	FF	$\tilde{\beta}^0$			*
	VP	$\tilde{\beta}^0$	65	0	70.00
A_2	FF	$\tilde{\beta}^0$	4	4	34.34
	VP	$\tilde{\beta}^0$	4	4	41.64
	FF	$\tilde{\beta}^0$	7	7	52.27
	VP	$\tilde{\beta}^0$	6	6	59.60
A_3	FF	$\tilde{\beta}^0$	16	16	118.22
	VP	$\tilde{\beta}$	3	3	35.35
	FF	$\tilde{\beta}^0$	18	18	130.50
	VP	$\tilde{\beta}^0$	6	6	61.92

$r(\hat{\tilde{a}}, \hat{\tilde{\alpha}})$, $r_2(\hat{\tilde{\alpha}}) \leq 3.0444 \times 10^8$

$(\tilde{\beta}^0 = (80, 49, 5, 81, 24, 9.5, 100, 4)^T)$

* Did not converge in finite amount of time.

REFERENCES

1. Bard, Yonathan, "Comparison of gradient methods for the solution of nonlinear parameter estimation problems", SIAM J. Numer. Anal. 7, pp. 157-186 (1970).

2. Barrodale, I., F. D. K. Roberts, and C. R. Hunt, "Computing best ℓ_p approximations by functions nonlinear in one parameter", Comp. J. 13, pp. 382-386 (1970).

3. Björck, A., and G. H. Golub, "Iterative refinement of linear least squares solutions by Householder transformations", BIT 7, pp. 322-337, (1967).

4. Brent, Richard P., "Algorithms for finding zeros and extrema of functions without calculating derivatives", Stanford University, Computer Science Report STAN-CS-71-198 (1971).

5. Daniel, J. W., The Approximate Minimization of Functionals, Prentice Hall, New York, (1971).

6. Dieudonné, J., Foundations of Modern Analysis, Academic Press, New York, (1960).

7. Fletcher, R. and Shirley A. Lill, "A class of methods for non-linear programming. II computational experience", in Nonlinear Programming (ed. by J. B. Rosen, 0. L. Mangasarian, and K. Ritter), pp. 67-92, Academic Press, New York (1970).

8. Golub, Gene H., "Matrix decompositions and statistical calculations", in Statistical Computation (edited by Roy C. Milton and John A. Nelder), pp. 365-397. Academic Press, New York (1969).

9. Guttman, I., V. Pereyra, and H. D. Scolnik, "Least squares estimation for a class of nonlinear models", Centre de Rech. Math., U. de Montreal, (Jan. 1971). To appear in Technometrics,

10. Hanson, Richard, J. and Charles-L. Lawson, "Extensions and applications of the Householder algorithm for solving linear least squares problems", Math. Comp. 23 > pp. 787-812 (1969).

11. Jennings, L. S., and M. R. Osborne, "Applications of orthogonal matrix transformations to the solution of systems of linear and nonlinear equations", Techn. Rep. No. 37, Computer C., Australian Nat. Univ. (1970).

12. Lawton, William H. and E. A. Sylvestre, 'Elimination of linear parameters in nonlinear regression", Technometrics 13, pp. 461-467 (1971).

13. Osborne, M. R., "A class of nonlinear regression problems" in Data Representation, (R. S. Anderssen and M. R. Osborne, editors), pp. 94-101 (1970).

14. Osborne, M. R., "Some aspects of nonlinear least squares calculations', unpublished manuscript (Nov. 1971).

15. Pereyra, V., "Iterative methods for solving nonlinear least squares problems" , SIAM J. Numer. Anal. 4, pp. 27-36 (1967).

16. Pereyra, V., "Stability of general systems of linear equations', Aequationes Math. 2, pp. 194-206 (1969).

17. Pérez, A., and H. D. Scolnik, "Derivatives of pseudoinverses and constrained non-linear regression problems", to appear in Numerische Mathematik.

18. Peters, G. and J. H. Wilkinson, "The least squares problem and pseudo-inverses" , The Comp. J. 13, pp. 309-316 (1970).

19. Rao, Radhakrishna C., and Sujit Kumar Mitra, Generalized Inverse of Matrices and its Applications, Wiley, New York (1971).

20. Scolnik, H. D., "On the solution of nonlinear least squares problems', Proc. IFIP-71, Numerical Math., pp. 18-23 (1971), North-Holland Pub. Co., Amsterdam. Also Ph.D. thesis, U. of Zurich (1970).

21. Travis, J. C., Radiochemical Analysis Section, Tech. Note No. 501, Nat. Bureau of Standards, pp. 19-33, Washington, D.C. (1970).

Key Words

Pseudoinverse

Nonlinear Least Squares

Fréchet Derivative

Projectors

Orthogonalization

FORTRAN Program

```
C*****DRIVER PROGRAM FOR USE OF  VARPRO  IN PROBLEM 1: FITTING OF TWO
C       EXPONENTIALS AND ONE CONSTANT TERM.
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION Y(200),T(200),ALF(20),AC(20)
        EXTERNAL ADA
        CALL USER(N,M,KG,NCFUN,Y,T,ALF)
        CALL VARPRO(N,M,KG,NCFUN,Y,T,ALF,AC,ADA)
        CALL EXIT
        END

C-------------------------------------------------------------------------------

        SUBROUTINE VARPRO(N,M,KG,NCFUN,Y,T,ALF,AC,ADA)
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON A(200,20),AA(200,10),E(20,20),B(220,20),UKK(200),
       * BETA(20),P
        INTEGER P
        DIMENSION UK1(20),BET1(20),Z(20),DR(20,20),ZPR(200),DEL (20)
       * ,ALF(KG),ALF1(20),AC(20),Y(M),T(M)
        EXTERNAL ADA
        NONLINEAR LEAST SQUARES PROGRAM FOR LINEAR COMBINATIONS OF NONLINEAR
C       FUNCTIONS.
C       WRITTEN IN FORTRAN 4  - LEVEL G. IN THIS SUBROUTINE THERE ARE WRITE
C       STATEMENTS USING UNIT 3 AS OUTPUT. THAT UNIT NUMBER IS INSTALLATION
C       DEPENDENT.
C       MINIMIZATION BY OSBORNE-MARQUARDT ALGORITHM (OR GAUSS-NEWTON WITH STEP
C       CONTROL BY MAKING THE SMALL CHANGES INDICATED IN THE SECOND LINE AFTER
C       INSTRUCTION LABELED 5, AND AFTER LABEL 61).
C       SEE 'THE DIFFERENTIATION OF PSEUDOINVERSES AND NONLINEAR LEAST SQUARES
C       PROBLEMS WHOSE VARIABLES SEPARATE' BY GENE H. GOLUB AND V. PEREYRA,
C       STANFORD U. TECHN. REP. 261, MARCH 1 9 7 2 .
C       M = NUMBER OF OBSERVATIONS.
C       N = NUMBER OF FUNCTIONS.
C       KG = NUMBER OF NONLINEAR VARIABLES.
C       NCFUN = NUMBER OF CONSTANT FUNCTIONS, I.E. FUNCTIONS PHI WHICH DO NOT
                DEPEND UPON ANY PARAMETERS ALPHA. THEY SHOULD APPEAR FIRST.
C       Y = M- VECTOR OF OBSERVATIONS.
C       T = M-VECTOR OF INDEPENDENT VARIABLE,
C       E=(N*KG) INCIDENCE MATRIX. E(I,J) = 1 IFF VARIABLE 3 APPEARS IN
                  FUNCTION I.  P= SUMO FE(I,J).
        ALF = KG-VECTOR OF INITIAL VALUES. ON OUTPUT IT WILL CONTAIN
                THE OPTIMAL VALUES OF THE NONLINEAR PARAMETERS.
        AC = N -VECTOR OF LINEAR PARAMETERS (OUTPUT).
      ********************************************************************
        CONTINUE
C       THE USER MUST PROVIDE A SUBROUTINE THAT FOR GIVEN ALF WILL EVALUATE
C       THE FUNCTIONS PHI AND THEIR PARTIAL DERIVATIVES D PHI(I)/D ALF(J), AT THE
        SAMPLE POINTSST . THE VECTOR SAMPLED FUNCTION PHI (I) SHOULD BE STORED IN
C       THE  I-TH COLUMN OF THE  (M X (P+N+1)) MATRIX A. THE NONZERO
C       DERIVATIVES COLUMN VECTORS SHOULD BE STORED SEQUENTIALLY IN THE MATRIX A
        STARTING IN THE COLUMN N+2. IF ITER=0 (THE FIRST TIME THIS SUBROUTINE IS
        CALLED), THE MATRIX E SHOULD BE FILLED. WITH THIS MATRIX THE STORAGE OF
        THE DERIVATIVES IS EXPLAINED IN THE FOLLOWING CODE:
        L = N+1
        DO 10 J=1,KG
```

```
C       DO 10 I=1,N
C       IF (E(I,J))10,10,11
c   11  L=L+1
c       DO 10 K=1,M
C       A(K,L) '='DPHI(I)/ D ALF(J)(T(K))
c   10  CONTINUE
C       THE  N+1-TH COLUMN OF  A IS RESERVED FOR THE VECTOR OF OBSERVATIONS Y.
C       THE SUBROUTINE HEADING SHOULD BE: ( ISEL =0 : FUNCT. AND DER. MUST B  E COMP
C       ISEL = -1 : ONLY FUNCTIONS MUST BE COMP. ISEL =1 : ONLY DER. NECESSARY)
C
C
C       SUBROUTINE ADA(N,M,KG,A,E,ITER,P,T,ALF,ISEL)
C
C
C       (ITER IS AN ITERATION COUNTER PROVIDED BY VARPRO).
C       If IS ASSUMED THAT THE MATRIX PHI (ALPHA) HAS ACWAYS FULL COLUMN RANK
C  ********************************************************************************
        ITER=0
C*****THE THREE FOLLOWING PARAMETERS ARE USED IN THE CONVERGENCE TEST (BETWEEN
c       INSTRUCTIONS NUMBER 200 A N D 400): EPS1 I S  A RELATIVE TOLERANCE FOR
c       DIFFERENCE BETWEEN TWO CONSECUTIVE RESIDUALS; ITMAX IS THE MAXIMUM
C       THE SIZE OF THE CORRECTION. EPS2 IS A RELATIVE TOLERANCE FOR THE
C       NUMBER OF FUNCTION AND DERIVATIVE EVALUATIONS ALLOWED.
        ITMAX=50
        EPS1=100-4
        EPS2=50D-6
C*****
        KG1=KG+1
        DO 10 I=1,M
   10   A(I,N+1)=Y(I)
    2   CALL  DPA(N,M,KG,NCFUN,ITER,ITER,R,Y,T,ALF,ADA)
        CT=1.DO
        IF(ITER.EQ.0)
       *WRITE(3,104)ITER,R
        IC=0
        IF(ITER)3,5,3
    5   CONTINUE
        XNU=0.
C*****
C*****IF GAUSS-NEWTON IS DESIRED REMOVE THE NEXT FOUR (4) STATEMENTS (SEE
C       ALSO LABEL 61) .
        DO 4 I=1,M
        DO 4 J=1,KG
    4   XNU=XNU+B(I,J)**2
        XNU= DSQRT(XNU/(M*KG))
        WRITE(3,105)XNU
C*****REDUCTION OF  B  TO TRIANGULAR FORM,
    3   DO 30 I=1,KG
        SGMA=0
        DO 11 I 1=I ,M
   11   SGMA= SGMA+B(I1,I)**2
        SGMA=DSQRT(SGMA)
        IF(B(I,I))12,12,13
   12   SG=-1.
        GO TO 14
```

```
   13   SG=1.
   14   UK1(I)=SG*(SGMA+DABS(B(I,I)))
        BET1(I)=1./(SGMA*(SGMA+DABS(B(I,I))))
        B(I,I)=-SG*SGMA
        I1=I+1
   23   DO 15 I2=I1,KG1
        ACUM=UK1(I)*B(I,I2)
        DO 16 I3=I1,M
   16   ACUM=ACUM+B(I3,I)*B(I3,I2)
   15   Z(I2)=BET1(I)*ACUM
        DO 30 J=I1,KG1
        B(I,J)=B(I,J)-UK1(I)*Z(J)
        DO 30 I2=I1,M
   30   B(I2,J)=B(I2,J)-B(I2,I)*Z(J)
C*****SAVE TRIANGULAR FORM AND Z'.
        DO 40 I=1,KG
        DO 40 J=I,KG
   40   DR(I,J)=B(I,J)
        DO 41 I=1,M
   41   ZPR(I)=B(I,KG1)
C*****REDUCTION. SECOND PHASE.
   50   IF(XNU .EQ. 0.D0)GO TO 300
        DO 60 K=1,KG
        K1=K+1
        B(M+K,K)=XNU
        DO 42 J=K1,KG1
   42   B(M+K,J)=0.D0
        SGMA=B(K,K)**2
        DO 51 J=1,K
   51   SGMA=SGMA+B(J+M,K)**2
        SGMA=DSQRT(SGMA)
        IF(B(K,K))52,52,53
   52   SG=-1.
        GO TO 54
   53   SG=1.
   54   UK1(K)=SG*(SGMA+DABS(B(K,K)))
        BET1(K)=1./(SGMA*(SGMA+DABS(B(K,K))))
        B(K,K)=-SG*SGMA
        DO 55  J=K1,KG1
        ACUM=UK1(K)*B(K,J)
        IF(K .EQ. 1)GO TO 55
        K2=K-1
        DO 56 I=1,K2
        IM=I+M
   56   ACUM=ACUM+B(IM,K)*B(IM,J)
   55   Z(J)=BET1(K)*ACUM
        DO 57 J=K1,KG1
        B(K,J)=B(K,J)-UK1(K)*Z(J)
        DO 57 I=1,K
        MI=M+I
   57   B(MI,J)=B(MI,J)-B(MI,K)*Z(J)
   60   CONTINUE
C*****SOLVE FOR DELTA-ALF.
  300   NZ=KG-1
        DEL(KG)=B(KG,KG1)/B(KG,KG)
```

```
      ALF1(KG)=ALF(KG)+DEL(KG)
      DO 58 I=1,N2
      I1=KG-I
      I2=I1+1
      ACUM=B(I1,KG1)
      DO 59 J=I2,KG
   59 ACUM=ACUM-B(I1,J)*DEL(J)
      DEL(I1)=ACUM/B(I1,I1)
   58 ALF1(I1)=ALF(I1)+DEL(I1)
C*****GET NEW RESIDUAL.
  310 ITER=ITER+1
      ISEL=-1
      WRITE(3,103)ITER
      WRITE(3,800)(ALF1(I),I=1,KG)
      IF(ITER .GT. ITMAX) GO TO 400
      DO 900 I=1,M
  900 A(I,N+1)=Y(I)
      CALL DPA(N,M,KG,NCFUN,ITER,ISEL,R1,Y,T,ALF1,ADA)
      IC=IC+1
      WRITE(3,107)IC,R1
      IF(R-R1)61,60,60
   61 CONTINUE
C*****IF GAUSS-NEWTON IS DESIRED REMOVE THE  C  FROM THE NEXT SIX (6) STATEMENTS
C     IF(XNU)110,111,110
C111  TT=0.5*TT
C     IF(TT .LT. 5.D-4)GO TO 400
C     DO 112 I=1,KG
C112  ALF1(I)=ALF(I)+TT*DEL(I)
C     GO TO 310
C*****
  110 XNU=1.5*XNU
      WRITE(3,106)XNU
C*****RETRIEVE TRIANGULAR FORM OF FIRST PHASE.
      DO 62 I=1,KG
      DO 62 J=I,KG
   62 B(I,J)=DR(I,J)
      DO 63 I=1,M
   63 B(I,KG1)=ZPR(I)
      GO TO 50
   60 EPS=R-R1
      R=R1

      ACC=0.
      DAC=0.
      DO 65 I=1,KG
      ALF(I)=ALF1(I)
      ACC=ACC+ALF(I)**2
   65 DAC=DAC+DEL(I)**2
C*****IF  IC IS GREATER THAN 1 THEN NU HAS BEEN INCREASED DURING THIS
C     ITERATION.
      IF(IC .EQ. 1 ) XNU=0.5*XNU
      WRITE(3,200)IC,XNU
      ACC=DSQRT(ACC)
      DAC=DSQRT(DAC)
      AC1=DAC/ACC
      WRITE(3,108)AC1
```

```
      IF ((DAC .LE. ACC*EPS1 .AND. EPS .LE. R*EPS2))
     *  GOTO 400
      ISEL=1
      GO TO 2
  410 N1=N-1
      AC(N)=A(N,N+1)/A(N,N)
      IF(N .EQ. 1)GOTO 135
      DO 130 I=1,N1
      I1=N-I
      I2=I1+1
      ACUM=A(I1,N+1)
      DO 120 J=I2,N
  120 ACUM=ACUM-A(I1,J)*AC(J)
  130 AC(I1)=ACUM/A(I1,I1)
  135 WRITE(3,209)
      WRITE(3,210)(AC(I),I=1,N)
      WRITE(3,215)(ALF(I),I=1,KG)
      WRITE(3,209)
  500 RETURN
  103 FORMAT(1HO,'   ITER=',I3,'         PARAMETERS')
  104 FORMAT(1HO,'  RESIDUAL',I5,D15.7)
  105 FORMAT(1HO,'  NU=',D15.7)
  106 FORMAT(1HO,'  N U W A S INCREASED TO',D15.7)
  137 FORMAT(1HO,I5,'  NEW RESIDUAL',D15.7)
  103 FORMAT(1HO,'  THENORM OF THE RELATIVE CORRECTION IS=',D15.3)
  200 FORMAT(1HO,IS,'  N U IS',D15.7)
  209 FORMAT(1HO,50('*'))
  210 FORMAT(1HO,'  WEIGHTS'//(4D15.7))
  215 FORMAT(1HO,'  NONLINEAR PARAMETERS'//(4D15.7))
  800 FORMAT(1HO,4D20.10)
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE DPA(N,M,KG,NCFUN,ITER,ISEL,R,Y,T,ALF,ADA)
C*****COMPUTATION OF THE DERIVATIVE OF THE VARIABLE PROJECTION.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON A(200,20),AA(200,10),E(20,20),B(220,20),UKK(200),
     * BETA(20),P
      INTEGER P
      DIMENSION ALF(KG),Z(120),X(20),U(20,20),Y(M),T(M)
      EXTERNAL ADA
      CALL ADA(N,M,KG,A,E,ITER,P,T,ALF,ISEL)
      N1=N+1
      N2=1
      IF(ISEL.GT.0)GO TO 111
      IF(ITER .GT. 0)N2=NCFUN+1
      DO 110 I=1,M
      DO 110 J=N2,N
  110 AA(I,J)=A(I,J)
C*****REDUCTION OF  A  TO TRIANGULAR FORM, COMPUTATION OF V=QY, AND
C     SELECTIVE COMPUTATION OF QB ACCORDING TO VALUE OF ISEL.
  111 DO 30 I=1,N
      I11=I+1
      IF(ISEL.GT.0)GO TO 22
```

```
      IF(ITER .GT. 0 .AND. I .LE. NCFUN)GO T O  7
      SGMA=0.
      DO 1  1 I1=I,M
   11 SGMA=SGMA+A( 11, I )**2
      SGMA=DSQRT(SGMA)
      IF(A(I,I))12,12,13
   12 SG=-1.
      GO TO  14
    7 I1=NCFUN+1
   13 GO TO 20
   14 SG=1.
      UKK(I)=SG*(SGMA+DABS(A(I,I)))
      BETA(I)=1./(SGMA*(SGMA+DABS(A(I,I))))
      A(I,I)=-SG*SGMA
      I1=I11
    8 IF(ISEL)20,21,22
   20 NN=N1
      GO TO 23
   21 NN=N1+P
      GO TO 2 3
   22 NN=N1+P
      I1=N+2
   23 DO 1  5 I2=I1,NN
      ACUM=UKK(I)*A(I,I2)
      DO 1  4 I3=I11,M
   16 ACUM=ACUM+A(I3,I)*A( 13, I2)
   15 Z(I2)=BETA(I)*ACUM
      DO 1 7 J=I1,NN
      A(I,J)=A(I,J)-UKK(I)*Z(J)
      DO 1 7 I2=I11,M
   17 A(I2,J)=A(I2,J)-A(I2,I)*Z(J)
   30 CONTINUE
      IF(ISEL.GT.0)GO T O  5 0
      R=0.
      DO 40 I=N1,M
   40 R=R+A(I,N1)**2
      IF( ISEL .LT. 0)RETURN
C*****D-SNAKE IS CONTAINED NOM IN A(I,J),I=N+1,...,M; J=N+2,...,N+P+1.
C     COMPUTATION O F X.
   50 N2=N-1
      X(N)=A(N,N1)/A(N,N)
      IF(N .EQ. 1) G  O TO 310
      DO 3 0 0 I=1,N2
      I1=N-I
      I2=I1+1
      ACUM=A(I1,N1)
      DO 2 0 0 J=I2,N
  200 A C U M =ACUM-A(I1,J)*X(J)
  300 X(I1)=ACUM/A(I1,I1)
C*****COMPUTATION O  F U.
  310 L=N1
      DO 60 J=1,KG
      DO 60 I=1,N
      IF(E(I,J))70,70,71
   70 U(I,J)=0.
```

```
      GO TO 60
   71 L=L+1
      ACUM=0.
      DO 600 K=N1,M
  600 ACUM=ACUM+A(K,L)*A(K,N1)
      U(I,J)=ACUM
   60 CONTINUE
C*****COMPUTATION OF  W(STORED IN UPPER PART O   F B).
      DO 80 J=1,KG
      B(1,J)=U(1,J)/A(1,1)
      DO 80 I=2,N
      ACUM=U(I,J)
      I1=I-1
      DO 7 9 L=1,I1
   7 9ACUM=ACUM-A(L,I )*B(L,J)
   80 B(I,J)=ACUM/A(I,I)
C*****COMPUTATION OF D-SNAKE* X(STORED IN LOWER PART OF B).
      DO 9  0 I=N1,M
      L=N1
      DO 9  0 J=1,KG
      ACUM=0.
      0  0 900 K=1,N
      IF(E(K,J))900,900,92
    3 2 L=L+1
      ACUM=ACUM+A(I,L)*X(K)
  900 CONTINUE
   90 B(I,J)=ACUM
C*****FINALLY,DPA(ALF)*Y IS PRODUCED AS QTB.
      DO 9  5 K1=1,N
      K=N-K1+1
      DO 93 I=1,KG
      K2=K+1
      ACUM=UKK(K)*B(K,I)
      DO 9 4 J=K2,M
    3 4 ACUM=ACUM+A(J,K)*B(J,I)
    93 Z(I)=BETA(K)*ACUM
      DO 9 6 J=1,KG
      B(K,J)=B(K,J)-UKK(K)*Z(J)
      DO 9  6 I=K2,M
    9 6 B(I,J)=B(I,J)-A(I,K)*Z(J)
    9 5 CONTINUE
C*****COMPUTATION OF ETA=ORTOGONAL COMPONENT OF Y, RESPECT OF A.
      DO 1 2 0 I=1,M
      ACUM= Y(I)
      DO 119 J=1,N
  119 ACUM=ACUM-AA( I,J)*X(J)
  120 B( I,KG+1)=ACUM
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE ADA(N,M,KG,A,E,ITER,P,T,ALF,ISEL)
C     OSBORNE'S EXPONENTIAL FITTING.TWO EXPONENTIALS AND CONSTANT TERM.
      IMPLICIT REAL*8(A-H,O-Z)
```

```
      INTEGER P
      DIMENSION A(200,20 ),E(20,20),ALF(KG),T(M)
      L=0
      IF(ITER .GT. 0)GO TO 5
C*****IN THIS CASE THE INCIDENCE MATRIX  E  IS:
C
C        ⎛ 0  0 ⎞
C        ⎜ 1  0 ⎟
C        ⎝ 0  1 ⎠
C
      E(1,1)=0.
      E(1,2)=0.
      E(2,1)=1.
      E(2,2)=0.
      E(3,1)=0.
      E(3,2)=1.
      P=2
      DO 4 I=1,M
    4 A(I,1)=1.0D0
    5 IF(ISEL  .GT.  0)GO TO 16
      DO 1 0 I=1,M
      A(I,2)=DEXP(-ALF(L+1)*T(I))
   1 0 A(I,3)=DEXP(-ALF(L+2)*T(I))
      IF(ISEL)14,15,16
   1 b DO 17 I=1,M
      A(I,5)=-T(I)*DEXP(-ALF(L+1)*T(I))
    1 7 A(I,6)=-T(I)*DEXP(-ALF(L+2)*T(I))
   14 RETURN
   15 DO 20 I=1,M
      A(I,5)=-T(I)*A(I,2)
   2 0 A(I,6)=-T(I)*A(I,3)
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE LEER(N,M,KG,NCFUN,Y,T,ALF)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(200),T(200),ALF(20)
C*****LEER  READS THE DATA.SEE FORMATS 100,102.
    1   READ(1,100,END=500)N,M,KG,NCFUN,(T(I),Y(I),I=1,M)
   100 FORMAT(4I5/(2D15.7))
      WRITE(3,101)N,M,KG,NCFUN,(I,T(I),Y(I),I=1,M)
   101 FORMAT(1H1,'  NON LINEAR LEAST SQUARES PROBLEM'//'  NUMBER Of- FUNC
     *TIONS=',13,3X                    ,                    'NUMBER OF OBS
     *ERVATIONS=',I3// '  NUMBER OF VARIABLES =',I3,'NUMBER OF CONSTANT
     * FUNCTIONS=',I3//                                    '     |
     *T(I)                  Y(I)'//(I5,2D20.7))
      N1=1
      READ(1,102)(ALF(I),I=N1,KG)
  102  FORMAT(4D20.7)
      WRITE(3,103)(ALF(I),I=N1,KG)
  103  FORMAT(1H0,'  INITIAL NONLINEAR PARAMETERS'//(4D20.7))
      WRITE(3,104)
  104 FORMAT (  1H0,50('*'))
```

```
      RETURN
500   CALL EXIT
      END
```

NUN LINEAR LEASTSQUARES PROBLEM

NUMBER OF FUNCTIONS= 3 NUMBER OF OBSERVATIONS= 33

NUMBER OF VARIABLES = 2 NUMBER OF CONSTANT FUNCTIONS= 1

1	T(I)	Y(I)
1	0.0	0.84400000D 00
2	0.10000000D 02	0.90800000D 00
3	0.20000000D 02	0.93200000 00
4	0.30000000D 02	0.93600000 00
5	0.40000000D 02	0.92500000 00
6	0.50000000D 02	0.90800000 00
7	0.60000000 02	0.88100005 00
8	0.70000000D 02	0.85000000D 00
9	0.80000000D 02	0.81800000 00
10	0.90000000 02	0.78400000D 00
11	0.10000000D 03	0.75100000 00
12	0.11000000 03	0.71800000D 00
13	0.12000000 03	0.68500000D 00
14	0.13000000 03	0.65800000D 00
15	0.14000000D 03	0.62800000D 00
16	0.13000000 03	0.60300000D 00
17	0.16000000 03	0.58000000 00
18	0.17000000D 03	0.55800000D 00
19	0.18000000 03	0.53800000D 00
20	0.19000000D 03	0.52200000D 00
21	0.20000000 03	0.50600000 00
22	0.21000000 03	0.49000000D 00
23	0.22000000 03	0.47800000D 00
24	0.23000000 03	0.46700000D 00
25	0.24000000D 03	0.45700000D 00
26	0.25000000D 03	0.44800000 00
27	0.26000000 03	0.43800000D 00
28	0.27000000D 03	0.43100000D 00
29	0.28000000D 03	0.42400000D 00
30	0.29000000D 03	0.42000000D 00
31	0.30000000 03	0.41400000 00
32	0.31000000 03	0.41100000D 00
33	0.32000000D 03	0.40600000D 00

INITIAL NONLINEAR PARAMETERS

0.10000000D-01 0.20000000D-01

****

RESIDUAL C 0.4917861D-02

NU= 0.2444940D 01

ITER= 1 PARAMETERS

0.12950688730-01 0.21832093270-01

1 NEW RESIDUAL 0.5609383D-04

1 NU IS 0.1222470D 01

THE NORM OF THE RELATIVE CORRECTION IS= 0.1370 00

```
 ITER=  2          PARAMETERS

 0.1292835923D-01     0.2199967360D-01

 I  NEW RESIDUAL  0.5468443D-04

 1  NU IS  0.6112350D 00

THE NORM OF THE RELATIVE CORRECTION IS=        0.663D-02

 ITER=  3          PARAMETERS

 0.1287837647D-01     0.2210022751D-01

 1  NEW RESIDUAL 0.5465016D-04

 1 N U  I S  0 . 3 0 5 6 1 7 5 0 0 0

THE NORM OF THE RELATIVE CORRECTION IS=        0.439D-02

 ITER= 4  ,        PARAMETERS

 0.1286831632D-01     0.2212108054D-01

 1  NEW RESIDUAL 0.5464895D-04

 1  N U  I S 0.1528088D 0  0

THE NORM OF THE RELATIVE CORRECTION IS=        0.905D-03

**************************************************

 WEIGHTS

0.37541320  00  0.1936239D 01 -0.1465082D 01

 NONLINEAR PARAMETERS

0.1286832D-01  0.2212108D-01

**************************************************
```