SIAM J. NUMER. ANAL.
Vol. 10, No. 2, April 1973.

THE DIFFERENTIATION OF PSEUDO-INVERSES AND NONLINEAR LEAST SQUARES PROBLEMS WHOSE VARIABLES SEPARATE*

G. H. GOLUB† AND V. PEREYRA‡

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

$$\eta(\mathbf{a}, \boldsymbol{\alpha}; t) = \sum_{j=1}^{n} a_j \varphi_j(\boldsymbol{\alpha}; t), \qquad \mathbf{a} \in \mathcal{R}^n, \quad \boldsymbol{\alpha} \in \mathcal{R}^k.$$

For this purpose we study the minimization of the nonlinear functional

$$r(\mathbf{a}, \boldsymbol{\alpha}) = \sum_{i=1}^{m} (y_i - \eta(\mathbf{a}, \boldsymbol{\alpha}, t_i))^2.$$

It is shown that by defining the matrix $\{\Phi(\boldsymbol{\alpha})\}_{i,j} = \varphi_j(\boldsymbol{\alpha}; t_i)$, and the modified functional $r_2(\boldsymbol{\alpha}) = \|\mathbf{y} - \Phi(\boldsymbol{\alpha})\Phi^+(\boldsymbol{\alpha})\mathbf{y}\|_2^2$, it is possible to optimize first with respect to the parameters $\boldsymbol{\alpha}$, and then to obtain, a posteriori, the optimal parameters $\hat{\mathbf{a}}$. The matrix $\Phi^+(\boldsymbol{\alpha})$ is the Moore–Penrose generalized inverse of $\Phi(\boldsymbol{\alpha})$. We develop formulas for the Fréchet derivative of orthogonal projectors associated with $\Phi(\boldsymbol{\alpha})$ and also for $\Phi^+(\boldsymbol{\alpha})$, under the hypothesis that $\Phi(\boldsymbol{\alpha})$ is of constant (though not necessarily full) rank. 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] and Guttman, Pereyra and Scolnik [9].

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], [23]–[26], [36]).

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

$$r(\mathbf{a}, \boldsymbol{\alpha}) = \sum_{i=1}^{m} \left[y_i - \sum_{j=1}^{n} a_j \varphi_j(\boldsymbol{\alpha}; t_i) \right]^2. \tag{1.1}$$

Throughout this paper a lower-case letter in boldface will indicate a column vector, while the same letter with a subscript will indicate a component of the

* Received by the editors April 13, 1972, and in revised form September 26, 1972.

† Computer Science Department, Stanford University, Stanford, California 94305. The work of this author was supported in part by the Atomic Energy Commission.

Gene H. Golub is Professor of Computer Science at Stanford University where, since 1962, he was a close colleague of Professor Forsythe.

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

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 $\mathbf{u}$ is indicated by $\mathbf{u}^T$. Given a function $f(t)$, we shall denote by $\mathbf{f}$ the vector whose components are $(f(t_1), f(t_2), \cdots, f(t_m))^T$. The scalar product of two vectors $\mathbf{u}$ and $\mathbf{v}$ is indicated by

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

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

$$A\tilde{\mathbf{x}} \cong \mathbf{b}$$

if $\|\mathbf{b} - A\hat{\mathbf{x}}\| \leqq \|\mathbf{b} - A\mathbf{x}\|$ for all $\mathbf{x} \in R^n$.

We shall use the symbol $\mathbf{D}$ for the Fréchet derivative of a mapping and ∇ for the gradient of a functional. We assume that the reader has some familiarity with pseudo-inverses and Fréchet derivatives and their properties. A useful reference for the pseudo-inverse is [19]; for details on the formalism and manipulation of Fréchet derivatives, we suggest [6, Chap. 8].

Let

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

and

$$\mathbf{a} = (a_1, \cdots, a_n)^T.$$

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

$$r(\mathbf{a}, \boldsymbol{\alpha}) = \|\mathbf{y} - \Phi(\boldsymbol{\alpha})\mathbf{a}\|^2. \tag{1.2}$$

Our approach to finding a critical point or a minimum of the functional (1.2) requires an important hypothesis.

H1. *The matrix $\Phi(\boldsymbol{\alpha})$ has constant rank $r \leqq \min(m, n)$ for $\boldsymbol{\alpha} \in \Omega \subset \mathscr{R}^k$, Ω being an open set containing the desired solution.*

We say that a matrix function satisfying H1 on an open neighborhood of a point $\boldsymbol{\alpha}^0$ has *local constant rank* at $\boldsymbol{\alpha}^0$.

Our aim is firstly to minimize a modified functional which depends only on the nonlinear parameters $\boldsymbol{\alpha}$, and then to proceed to obtain the linear parameters $\mathbf{a}$. In [9], [20] simpler models were treated, i.e., $\varphi_j(\boldsymbol{\alpha}; t) = t^{\alpha_j}$, and $\varphi_j(\boldsymbol{\alpha}; t) = \varphi_j(\alpha_j; t)$. A similar point of view was used but different analytic tools were employed. 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 the separation of variables we consider, as in [9], [17], [20], the modified functional

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

which will be called the *variable projection functional.* Once optimal parameters $\hat{\boldsymbol{\alpha}}$ have been obtained by minimizing (1.3), then the parameters $\hat{\mathbf{a}}$ are obtained as a solution of $\Phi(\hat{\boldsymbol{\alpha}})\mathbf{a} \cong \mathbf{y}$.

We shall show in Theorem 2.1 the relationship between critical or minimal points of the original functional $r(\mathbf{a}, \boldsymbol{\alpha})$ and those obtained from the functional $r_2(\boldsymbol{\alpha})$.

Both for our proof and for the numerical algorithms of § 5 we need to develop formulas for the gradient of the functional (1.3). In § 4 we develop these formulas and obtain the derivatives of projectors, the Jacobian of the residual vector, and the pseudo-inverse of a matrix function. The only hypothesis we make on the rank of the matrix is that it must be locally constant at the point where the derivative is calculated. This is necessary, since otherwise the pseudo-inverse is not a continuous function, and therefore it could hardly be differentiable. Our proof and final results are coordinate-free. For the full rank case similar formulas have been obtained for the pseudo-inverse 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, while in [17] the authors also obtain a formula for the rank deficient case in terms of a full rank factorization of the given matrix. Our formula in turn is given exclusively in terms of the original matrix, its derivative, and its pseudo-inverse, and it seems to be new.[1]

In § 5 we give a detailed explanation of how to implement the method in an efficient way and in § 6 we present some numerical examples and comparisons. Extensive use is made of linear least squares techniques. A FORTRAN program based on the ideas of this paper is given in [25].

2. A class of nonlinear least squares problems whose parameters separate. We are going to consider in this paper models of the form

$$\eta(\mathbf{a}, \boldsymbol{\alpha}; t) = \sum_{j=1}^{n} a_j \varphi_j(\boldsymbol{\alpha}; t), \tag{2.1}$$

where $\mathbf{a} \in \mathcal{R}^n$, $\boldsymbol{\alpha} \in \mathcal{R}^k$, and the functions φ_j are continuously differentiable with respect to $\boldsymbol{\alpha}$. We remark that the parameters $\mathbf{a}$ and $\boldsymbol{\alpha}$ 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), $i = 1, \cdots, m$, $m \geqq n + k$, our task is to find the values of the parameters $\mathbf{a}, \boldsymbol{\alpha}$, that minimize the nonlinear functional

$$r(\mathbf{a}, \boldsymbol{\alpha}) = \|\mathbf{y} - \boldsymbol{\eta}(\mathbf{a}, \boldsymbol{\alpha})\|^2 = \sum_{i=1}^{m} (y_i - \eta(\mathbf{a}, \boldsymbol{\alpha}; t_i))^2. \tag{2.2}$$

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

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

$$\Phi(\boldsymbol{\alpha}) = [\boldsymbol{\varphi}_1(\boldsymbol{\alpha}), \cdots, \boldsymbol{\varphi}_n(\boldsymbol{\alpha})]. \tag{2.3}$$

For each $\boldsymbol{\alpha}$, the linear operator

$$P_{\Phi(\boldsymbol{\alpha})} = \Phi(\boldsymbol{\alpha})\Phi^+(\boldsymbol{\alpha}) \tag{2.4}$$

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

[1] As this paper was being sent to the printers, the authors learned that Decell and Fries [34] had also obtained this result recently.

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

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

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.

For any given α we have the minimal least squares solution

$$(2.5) \qquad \hat{\mathbf{a}}(\alpha) \equiv \Phi^{+}(\alpha)\mathbf{y}.$$

Thus,

$$(2.6) \qquad \min_{\mathbf{a}} r(\mathbf{a}, \alpha) = r(\hat{\mathbf{a}}, \alpha) = \|\mathbf{y} - \Phi(\alpha)\Phi^{+}(\alpha)\mathbf{y}\|^2 = \|P^{\perp}_{\Phi(\alpha)}\mathbf{y}\|^2.$$

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

$$(2.7) \qquad r_2(\alpha) = \|P^{\perp}_{\Phi(\alpha)}\mathbf{y}\|^2.$$

Once a critical point (or a minimizer) $\hat{\alpha}$ is found for this functional, then $\hat{\mathbf{a}}$ is obtained by replacing α by $\hat{\alpha}$ in (2.5).

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

THEOREM 2.1. *Let $r(\mathbf{a}, \alpha)$ and $r_2(\alpha)$ be defined as above. We assume that in the open set $\Omega \subset \mathscr{R}^k$, $\Phi(\alpha)$ has constant rank $r \leqq \min(m, n)$.*

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

$$(2.8) \qquad \hat{\mathbf{a}} = \Phi^{+}(\hat{\alpha})\mathbf{y},$$

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

(b) *If $(\hat{\mathbf{a}}, \hat{\alpha})$ is a global minimizer of $r(\mathbf{a}, \alpha)$ for $\alpha \in \Omega$, then $\hat{\alpha}$ is a global minimizer of $r_2(\alpha)$ in Ω and $r_2(\hat{\alpha}) = r(\hat{\mathbf{a}}, \hat{\alpha})$. Furthermore, if there is an unique $\hat{\mathbf{a}}$ among the minimizing pairs of $r(\mathbf{a}, \alpha)$, then $\hat{\mathbf{a}}$ must satisfy* (2.8).

We shall postpone the proof of this theorem until the end of § 4, where we obtain a convenient expression for the gradient of the functional $r_2(\alpha)$.

Although separation of variables has been used elsewhere as indicated earlier, the correspondence between the critical points of $r(\mathbf{a}, \alpha)$ and $r_2(\alpha)$ has only been studied before in [9] for a simpler case. See also [36].

3. Algorithmia I. Residual calculation. One of our main points in the algorithmic parts 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 to introduce the necessary notation.

If Q is an orthogonal matrix, then for every vector $\mathbf{z}$, $\|Q\mathbf{z}\| = \|\mathbf{z}\|$. It is well known (cf. [8], [10], [18], [22]) that every $m \times n$ matrix Φ ($m \geqq n$) of rank $r \leqq n$

can be orthogonally transformed into "trapezoidal" form. That is, there exists an orthogonal matrix Q and a permutation matrix S such that

$$Q\Phi S = \left[\begin{array}{c|c} T_{11} & T_{12} \\ \hline 0 & 0 \end{array}\right] \equiv T_0, \tag{3.1}$$

where T_{11} is an $r \times r$ nonsingular upper triangular matrix. Naturally, $\Phi = Q^{\top} T_0 S^{\top}$.

We indicate by Φ^{-} any $n \times m$ matrix which satisfies the two properties

$$\Phi\Phi^{-}\Phi = \Phi, \qquad (\Phi\Phi^{-})^{\top} = (\Phi\Phi^{-}). \tag{3.2}$$

We observe (cf. [19], [22]) that

$$P_{\Phi} = \Phi\Phi^{-},$$

and hence Φ^{+} is not necessary for computing P_{Φ}.

From the decomposition (3.1), we can obtain a Φ^{-}. Let

$$\Phi^{B} = S\begin{bmatrix} T_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix} Q. \tag{3.3}$$

It is easy to verify that Φ^{B} satisfies (3.2) and also $\Phi^{B}\Phi\Phi^{B} = \Phi^{B}$. Hence from (3.1) and (3.3), it follows that

$$\begin{aligned} P_{\Phi} &= \Phi\Phi^{B} = Q^{\top}\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} Q, \\ P_{\Phi}^{\perp} &= I - P_{\Phi} = Q^{\top}\begin{bmatrix} 0 & 0 \\ 0 & I_{m-r} \end{bmatrix} Q. \end{aligned} \tag{3.4}$$

Due to the isometric properties of the orthogonal transformation Q, the least squares problem $\Phi\mathbf{a} \cong \mathbf{y}$ is equivalent to $Q\Phi\mathbf{a} \cong Q\mathbf{y}$. We define

$$\bar{\mathbf{y}} \equiv Q\mathbf{y} \equiv \begin{bmatrix} \bar{\mathbf{y}}_1 \\ \cdots \\ \bar{\mathbf{y}}_2 \end{bmatrix} \begin{matrix} \}r \\ \}(m-r) \end{matrix}$$

A simple computation shows that

$$\|\mathbf{y} - \Phi\hat{\boldsymbol{\alpha}}\|^2 = \|P_{\Phi}^{\perp}\mathbf{y}\|^2 = \|\bar{\mathbf{y}}_2\|^2. \tag{3.5}$$

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

$$r_2(\boldsymbol{\alpha}) = \|\bar{\mathbf{y}}_2\|^2 \tag{3.6}$$

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 $P_{\Phi(\boldsymbol{\alpha})}^{\perp}\mathbf{y}$ we shall provide in the next section formulas which will also be useful in the proof of Theorem 2.1.

4. Fréchet derivatives of projectors, residual vectors, and pseudo-inverses, and their applications. In this section we develop formulas for the Fréchet derivative of orthogonal projectors associated with differentiable matrix functions. This leads to expressions for the derivatives of the vector function

$$\mathbf{r}_2(\boldsymbol{\alpha}) = P^{\perp}_{\Phi(\boldsymbol{\alpha})}\mathbf{y}, \tag{4.1}$$

and that of the pseudo-inverse of $\Phi(\boldsymbol{\alpha})$. These expressions were developed in [25]. The arguments given here, however, are considerably simpler due to an observation of G. W. Stewart [27].

An $m \times n$ matrix function $A(\boldsymbol{\alpha})$ is a nonlinear mapping between the linear space of parameters, $\mathscr{R}^k$, and the space of linear transformations $\mathscr{L}(\mathscr{R}^n, \mathscr{R}^m)$. Consequently, $\mathbf{D}A(\boldsymbol{\alpha})$ will be, for each $\boldsymbol{\alpha}$, an element of $\mathscr{L}(\mathscr{R}^k, \mathscr{L}(\mathscr{R}^n, \mathscr{R}^m))$. Thus, $\mathbf{D}A(\boldsymbol{\alpha})$ could be interpreted as a tridimensional tensor, formed with k $(m \times n)$ 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. We expect that our efforts in not burdening the reader with a more formal but considerably heavier notation will be appreciated.

First we compute the Fréchet derivative of the orthogonal projector $P_{A(\boldsymbol{\alpha})}$ associated with a differentiable $m \times n$ matrix function $A(\boldsymbol{\alpha})$ of local constant rank r.

LEMMA 4.1. *Let $A^-(\boldsymbol{\alpha})$ be an $n \times m$ matrix function such that $AA^-A = A$ and $(AA^-)^{\mathsf{T}} = AA^-$. Then*

$$\mathbf{D}P_A = P_A^{\perp}\mathbf{D}AA^- + (P_A^{\perp}\mathbf{D}AA^-)^{\mathsf{T}}. \tag{4.2}$$

Proof. Since $P_AA = A$,

$$\mathbf{D}(P_AA) = \mathbf{D}P_AA + P_A\mathbf{D}A = \mathbf{D}A,$$

and hence,

$$\mathbf{D}P_AA = \mathbf{D}A - P_A\mathbf{D}A = P_A^{\perp}\mathbf{D}A.$$

Thus, since $P_A = AA^-$,

$$\mathbf{D}P_AP_A = \mathbf{D}P_AAA^- = P_A^{\perp}\mathbf{D}AA^-. \tag{4.3}$$

Since

$$(\mathbf{D}P_AP_A)^{\mathsf{T}} = P_A\mathbf{D}P_A \tag{4.4}$$

(the transposition being done within each symmetric slab $(\partial/\partial\alpha_i)P_A$ of the tridimensional tensor $\mathbf{D}P_A$) we finally obtain from (4.3) and (4.4)

$$\mathbf{D}P_A = \mathbf{D}(P_A^2) = \mathbf{D}P_AP_A + P_A\mathbf{D}P_A = P_A^{\perp}\mathbf{D}AA^- + (P_A^{\perp}\mathbf{D}AA^-)^{\mathsf{T}},$$

which completes the proof.

Lemma 4.1 is, of course, valid when A^- is replaced by A^+. Also

$$\mathbf{D}P_A^{\perp} = -\mathbf{D}P_A. \tag{4.5}$$

We now use the results of Lemma 4.1 to obtain the gradient of the variable projection functional $r_2(\boldsymbol{\alpha})$. Since

$$r_2(\boldsymbol{\alpha}) = \|P^{\perp}_{\Phi(\boldsymbol{\alpha})}\mathbf{y}\|^2 = \langle P^{\perp}\mathbf{y}, P^{\perp}\mathbf{y}\rangle, \tag{4.6}$$

we have immediately by (4.2) and (4.5),

$$\tfrac{1}{2}\nabla r_2(\boldsymbol{\alpha}) = -\mathbf{y}^{\mathsf{T}}P^{\perp}[P^{\perp}\mathbf{D}\Phi\Phi^{-} + (\Phi^{-})^{\mathsf{T}}\mathbf{D}\Phi^{\mathsf{T}}P^{\perp}]\mathbf{y}^{\mathsf{T}}.$$

Now we assume that Φ^{-} satisfies the additional hypothesis

$$\Phi^{-}\Phi\Phi^{-} = \Phi^{-}.$$

Then

$$\begin{aligned} P^{\perp}(\Phi^{-})^{\mathsf{T}} &= (\Phi^{-})^{\mathsf{T}} - \Phi\Phi^{-}(\Phi^{-})^{\mathsf{T}} \\ &= (\Phi^{-})^{\mathsf{T}} - (\Phi\Phi^{-})^{\mathsf{T}}(\Phi^{-})^{\mathsf{T}} \\ &= 0. \end{aligned}$$

Hence,

$$\tfrac{1}{2}\nabla r_2(\boldsymbol{\alpha}) = -\mathbf{y}^{\mathsf{T}}P^{\perp}_{\Phi(\boldsymbol{\alpha})}\mathbf{D}\Phi(\boldsymbol{\alpha})\Phi^{-}(\boldsymbol{\alpha})\mathbf{y}. \tag{4.7}$$

D. Jupp has independently developed (4.7) in the full rank case. This formula is particularly useful in association with variable metric minimization procedures (cf. [23], [24]).

Now we have the elements for proving Theorem 2.1.

Proof of Theorem 2.1. From (2.2) we have that $r(\mathbf{a}, \boldsymbol{\alpha}) = \|\mathbf{y} - \Phi(\boldsymbol{\alpha})\mathbf{a}\|^2$. Therefore,

$$\tfrac{1}{2}\nabla r(\mathbf{a}, \boldsymbol{\alpha}) = -(\mathbf{y} - \Phi\mathbf{a})^{\mathsf{T}}(\mathbf{D}\Phi\mathbf{a} \oplus \Phi), \tag{4.8}$$

where $\oplus$ stands for direct sum.

Assume that $\hat{\boldsymbol{\alpha}}$ is a critical point of $r_2(\boldsymbol{\alpha})$, and that $\mathbf{a}$ is defined by

$$\hat{\mathbf{a}} \equiv \Phi^{+}(\hat{\boldsymbol{\alpha}})\mathbf{y}. \tag{4.9}$$

Then,

$$\begin{aligned} \tfrac{1}{2}\nabla r(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}}) &= -(P^{\perp}_{\Phi}\mathbf{y})^{\mathsf{T}}(\mathbf{D}\Phi\Phi^{+}\mathbf{y} \oplus \Phi) \\ &= \tfrac{1}{2}\nabla r_2(\hat{\boldsymbol{\alpha}}) \oplus \mathbf{0} \end{aligned} \tag{4.10}$$

since $\mathbf{y}^{\mathsf{T}}P^{\perp}_{\Phi}\Phi = \mathbf{0}$. Thus $(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}})$ is a critical point of $r(\mathbf{a}, \boldsymbol{\alpha})$.

Assume now that $\hat{\boldsymbol{\alpha}}$ is a global minimizer of $r_2(\boldsymbol{\alpha})$ in Ω, and $\hat{\mathbf{a}}$ satisfies (4.9). Then clearly, $r(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}}) = r_2(\hat{\boldsymbol{\alpha}})$. Assume that there exists $(\mathbf{a}^*, \boldsymbol{\alpha}^*)$, $\boldsymbol{\alpha}^* \in \Omega$, such that $r(\mathbf{a}^*, \boldsymbol{\alpha}^*) < r(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}})$. Since for any $\boldsymbol{\alpha}$ we have $r_2(\boldsymbol{\alpha}) \leqq r(\mathbf{a}, \boldsymbol{\alpha})$, then it follows that $r_2(\boldsymbol{\alpha}^*) \leqq r(\mathbf{a}^*, \boldsymbol{\alpha}^*) < r(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}}) = r_2(\hat{\boldsymbol{\alpha}})$, which is a contradiction to the fact that $\hat{\boldsymbol{\alpha}}$ was a global minimizer of $r_2(\boldsymbol{\alpha})$ in Ω. Therefore $(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}})$ is a global minimizer of $r(\mathbf{a}, \boldsymbol{\alpha})$ in Ω, and part (a) of the Theorem is proved.

Conversely, suppose that $(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}})$ is a global minimizer of $r(\mathbf{a}, \boldsymbol{\alpha})$ in Ω. Then as above $r_2(\hat{\boldsymbol{\alpha}}) \leqq r(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}})$. Now let $\mathbf{a}^* = \Phi^{+}(\hat{\boldsymbol{\alpha}})\mathbf{y}$. Then we have $r_2(\hat{\boldsymbol{\alpha}}) = r(\mathbf{a}^*, \hat{\boldsymbol{\alpha}}) \leqq r(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}})$; but since $(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}})$ was a global minimizer we must have equality. If there was an unique $\mathbf{a}$ among the minimizers of $r(\mathbf{a}, \boldsymbol{\alpha})$, then $\mathbf{a}^* \equiv \hat{\mathbf{a}}$. We still have to

show that $\hat{\alpha}$ is a global minimizer of $r_2(\alpha)$. Assume that it is not. Thus, there will be $\bar{\alpha} \in \Omega$, such that $r_2(\bar{\alpha}) < r_2(\hat{\alpha})$. Let $\bar{\mathbf{a}}$ be equal to $\Phi^+(\bar{\alpha})\mathbf{y}$. Then $r_2(\bar{\alpha}) = r(\bar{\mathbf{a}}, \bar{\alpha}) < r_2(\hat{\alpha}) = r(\hat{\mathbf{a}}, \hat{\alpha})$, which is a contradiction to the fact that $(\hat{\mathbf{a}}, \hat{\alpha})$ was a global minimizer of $r(\mathbf{a}, \alpha)$. This completes the proof.

With Lemma 4.1 we have the machinery for obtaining the derivative of $\Phi^+(\alpha)$. Although this is a digression from the main theme of this paper, we develop the formula because of its novelty and its importance in related applications, some of which we shall mention briefly.

In order to prove Theorem 4.3 we need the following corollary to Lemma 4.1.

COROLLARY 4.2. *Let* ${}_AP = A^+A$. *Then*

$$\mathbf{D}_A P = A^+ \mathbf{D} A_A P^\perp + (A^+ \mathbf{D} A_A P^\perp)^\top. \tag{4.11}$$

Proof. Since $(P_{A^\top})^\top = (A^\top (A^\top)^+)^\top = A^+A = {}_AP$, then (4.11) follows readily from (4.2) with A replaced by $A^\top$.

THEOREM 4.3. *Let* $\Omega \subset \mathscr{R}^k$ *be an open set and for* $\alpha \in \Omega$ *let* $A(\alpha)$ *be an* $m \times n$ *Fréchet differentiable matrix function with local constant rank* $r \leqq \min(m, n)$ *in* Ω. *Then for any* $\alpha \in \Omega$,

$$\mathbf{D}A^+(\alpha) = -A^+ \mathbf{D} A A^+ + A^+ A^{+\top} \mathbf{D} A^\top P_A^\perp + {}_AP^\perp \mathbf{D} A^\top A^{+\top} A^+. \tag{4.12}$$

Proof. Since $(P_{A^\perp})^\top = (A^\top (A^\top)^+)^\top = A^+A = {}_AP$, then (4.11) follows readily

$$\mathbf{D}A^+ = \mathbf{D}({}_AP A^+) = \mathbf{D}_A P A^+ + {}_AP \mathbf{D} A^+.$$

Combining this with (4.11), and observing that ${}_AP^\perp A^+ = 0$, we obtain

$${}_AP^\perp \mathbf{D} A^+ = {}_AP^\perp \mathbf{D}_A P A^+ = {}_AP^\perp \mathbf{D} A^\top A^{+\top} A^+. \tag{4.13}$$

Now

$$\mathbf{D}A^+ = \mathbf{D}(A^+ A A^+) = \mathbf{D}A^+ P_A + A^+ \mathbf{D} A A^+ + {}_AP \mathbf{D} A^+,$$

and thus,

$$\mathbf{D}A^+ P_A = -A^+ \mathbf{D} A A^+ + {}_AP^\perp \mathbf{D} A^+.$$

Combining this last expression with (4.13), we have

$$\mathbf{D}A^+ P_A = -A^+ \mathbf{D} A A^+ + {}_AP^\perp \mathbf{D} A^\top A^{+\top} A^+. \tag{4.14}$$

But,

$$\mathbf{D}A^+ = \mathbf{D}(A^+ P_A) = \mathbf{D}A^+ P_A + A^+ \mathbf{D} P_A$$

and therefore,

$$\mathbf{D}A^+ P_A^\perp = A^+ \mathbf{D} P_A = A^+ A^{+\top} \mathbf{D} A^\top P_A^\perp, \tag{4.15}$$

since

$$A^+ P_A^\perp = 0.$$

The theorem follows from the relationship

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

and from (4.14), (4.15).

Formula (4.12) is new. Let $B = A + dA$ where dA is an arbitrary incremental matrix. Wedin [29], [30] has shown that

$$B^+ - A^+ = -B^+ dAA^+ + {}_BP^\perp dA^{\top} A^{+\top} A^+ + B^+ B^{+\top} dA^{\top} P_A^\perp. \tag{4.16}$$

Formula (4.16) can be used for deriving (4.12) by letting $dA \to 0$ as was done in the full rank case in [10]. This technique was pointed out to the authors by W. Kahan [28].

There are many potential applications for the formulas developed in this section, besides the one we explicitly give. We shall mention a few of them.

(a) *Optimization with nonlinear equality constraints.* Consider the problem

$$\min_{\mathbf{x}} f(\mathbf{x}), \qquad \mathbf{x} \in \mathscr{R}^n,$$

subject to

$$\mathbf{c}(\mathbf{x}) = \mathbf{0}, \qquad \mathbf{c}: \mathscr{R}^n \to \mathscr{R}^k, \qquad k \leqq n,$$

where $f(\mathbf{x})$ is a functional.

In Fletcher and Lill [7] methods are proposed which require the derivative of $(\mathbf{D}\mathbf{c})^+$. Our formulas would permit $(\mathbf{D}\mathbf{c})^+$ to be rank deficient, though the theory of this problem is not well understood at the present time. See also [17].

(b) *Generalized Newton's method for* $f(\mathbf{x}) = \mathbf{0}$. In [31] Ben-Israel considers the following iterative procedure:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\mathbf{D}\mathbf{f}(\mathbf{x}^k))^+ \mathbf{f}(\mathbf{x}^k). \tag{4.17}$$

Formula (4.12) could allow a direct study of the convergence properties of (4.17).

(c) *Stability of the solution of perturbed linear least squares problems.* Let $A(\varepsilon) = A + \varepsilon B$. We assume

(i) rank $(A(\varepsilon))$ = rank (A) for all $\varepsilon > 0$ sufficiently small,

(ii) $\|A\| = \|B\| = 1$.

We wish to consider the behavior of $\hat{\mathbf{x}}(\varepsilon)$ satisfying

$$\hat{\mathbf{x}}(\varepsilon) = A^+(\varepsilon)\mathbf{b}$$

as $\varepsilon \to 0$. Using Taylor's formula, we obtain

$$A^+(\varepsilon) - A^+(0) = \varepsilon(\mathbf{D}A^+)(0)B + O(\varepsilon^2). \tag{4.18}$$

Now by (4.12) we have

$$(\mathbf{D}A^+)(0) = -A^+BA^+ + A^+A^{+\top}B^{\top}P_A^\perp + {}_AP^\perp B^{\top}A^{+\top}A^+ \tag{4.19}$$

by using the fact that $(\mathbf{D}A)(0) = B$. Hence, using (4.18) and (4.19), we obtain by the usual norm argument,

$$\|\mathbf{x}(\varepsilon) - \mathbf{x}\| = \|A^+(\varepsilon)\mathbf{b} - A^+\mathbf{b}\| \leqq 2\varepsilon\|A^+\|\,\|\hat{\mathbf{x}}\| + \varepsilon\|A^+\|^2\|\hat{\mathbf{r}}\| + O(\varepsilon^2), \tag{4.20}$$

where $\hat{\mathbf{r}} = \mathbf{b} - A\hat{\mathbf{x}}$. Equation (4.20) is the rank deficient generalization of results given in [32]. A similar treatment can be used to obtain more detailed estimates of the type given in [16], [29], [30], [33].

5. Algorithmia II. Detailed implementation of the Gauss–Newton–Marquardt algorithm. Computation of the Jacobian of the variable projection vector. We shall now explain in detail how to apply the results of § 4 to the Marquardt modification of the Gauss–Newton iterative procedure; we make extensive use of linear least squares techniques. We describe an economical implementation of the Marquardt algorithm devised earlier by Golub (see also [11], [14]).

We define the vector

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

The generalized Gauss–Newton (G.N.) iteration with step control for the nonlinear least squares problem

$$\min_{\boldsymbol{\alpha}} r_2(\boldsymbol{\alpha}) = \min_{\boldsymbol{\alpha}} \|\mathbf{r}_2(\boldsymbol{\alpha})\|^2 = \min_{\boldsymbol{\alpha}} \|P^{\perp}_{\Phi(\boldsymbol{\alpha})}\mathbf{y}\|^2 \tag{5.1}$$

is given as follows:

$$\boldsymbol{\alpha}^{l+1} = \boldsymbol{\alpha}^l - t_l[\mathbf{D}\mathbf{r}_2(\boldsymbol{\alpha}^l)]^+\mathbf{r}_2(\boldsymbol{\alpha}^l), \qquad l = 0, \cdots. \tag{5.2}$$

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

Marquardt's modification calls for the introduction of a sequence of non-negative auxiliary parameters $v_l \geqq 0$.

(G.N.M.) Define

$$K_l \equiv \begin{bmatrix} \mathbf{D}\mathbf{r}_2(\boldsymbol{\alpha}^l) \\ \hline v_l F_l \end{bmatrix}, \qquad \mathbf{r}_l \equiv \begin{bmatrix} \mathbf{r}_2(\boldsymbol{\alpha}^l) \\ \hline 0 \end{bmatrix} \Big\} k'$$

where for each l, F_l is the upper triangular Cholesky factor of a $k \times k$ symmetric positive definite matrix M_l. Then the Gauss–Newton–Marquardt iteration is given by

$$\boldsymbol{\alpha}^{l+1} = \boldsymbol{\alpha}^l - K_l^+\mathbf{r}_l, \qquad l \geqq 0. \tag{5.3}$$

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

$$\mathbf{h} = \boldsymbol{\alpha}^{l+1} - \boldsymbol{\alpha}^l, \qquad DP = \mathbf{D}\mathbf{r}_2(\boldsymbol{\alpha}^l) = \mathbf{D}P^{\perp}_{\Phi}\mathbf{y}$$

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

$$K\mathbf{h} \cong \begin{bmatrix} -\mathbf{r}_2(\boldsymbol{\alpha}) \\ \hline \mathbf{0} \end{bmatrix}.$$

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

$$QDP = R_1 = \begin{bmatrix} \text{0} \\ \hline 0 \end{bmatrix} \}n \equiv \begin{bmatrix} R_1' \\ \hline 0 \end{bmatrix}.$$

Thus,

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

$$Q_1 \begin{bmatrix} -\mathbf{r}_2(\boldsymbol{\alpha}) \\ \hline \mathbf{0} \end{bmatrix} \equiv \begin{bmatrix} \bar{\mathbf{r}} \\ \hline \mathbf{0} \end{bmatrix}.$$

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

In the second stage we choose an $(m + k) \times (m + k)$ orthogonal matrix Q_2 to reduce $A \equiv \begin{bmatrix} R_1' \\ \hline vF \end{bmatrix}$ 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

$$\mathbf{a}_1^{(1)} = \left[\begin{array}{c} a_{11}^{(1)} \\ 0 \\ \vdots \\ 0 \end{array}\right] \Bigg\} m, \qquad \mu = vf_{11},$$
$$\left[\begin{array}{c} \mu \\ 0 \\ \vdots \\ 0 \end{array}\right]$$

we use $Q^{(1)} = I - \beta_1 \mathbf{u}^{(1)} \mathbf{u}^{(1)\top}$, where

$$u_1^{(1)} = \operatorname{sign}(a_{11}^{(1)})(\sigma_1 + |a_{11}^{(1)}|),$$
$$\sigma_1 = (a_{11}^2 + \mu^2)^{1/2},$$
$$u_{m+1}^{(1)} = \mu,$$
$$u_i^{(1)} = 0, \quad \text{otherwise},$$
$$\beta_1 = (\sigma_1 |u_1^{(1)}|)^{-1}.$$

Now we observe that when $Q^{(1)}$ is applied to a vector, any component corresponding to a zero component of $\mathbf{u}^{(1)}$ 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

$$A^{(2)} = \begin{bmatrix} * & * & * & * & * & * & * \\ 0 & & & & & & \\ & 0 & & & & & \\ & & 0 & & & & \\ o & * & * & * & * & * & * \\ & 0 & & & & & \end{bmatrix} \begin{matrix} \\ \\ \} k \\ \} m - k \\ \} k \\ \\ \end{matrix}$$

where the asterisks indicate the modified elements.

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

$$A^{(v)} = \begin{bmatrix} 0 & & & & & & \\ & 0 & & & & & \\ & & o & o & o & & \\ & & & & & o & \\ & 0 & & & & & \end{bmatrix}.$$

The matrix $A^{(v+1)}$, $v = 1, \cdots, k$, is obtained as follows:

(i) $$\sigma_v = ((a_{vv}^{(v)})^2 + \sum_{i=1}^{v} (a_{m+i,v}^{(v)})^2)^{1/2},$$

(ii) $$\beta_v = (\sigma_v(\sigma_v + |a_{vv}^{(v)}|))^{-1},$$

(iii) $$u_i^{(v)} = 0 \quad \text{for } i < v, \quad v + 1 \leqq i \leqq m, \quad m + v < i;$$

$$u_v^{(v)} = \operatorname{sign}(a_{vv}^{(v)})(\sigma_v + |a_{vv}^{(v)}|);$$

$$u_i^{(v)} = a_i^{(v)}, \qquad m + 1 \leqq i \leqq m + v.$$

(iv) $$\mathbf{y}^{\top} = \beta_v \mathbf{u}^{(v)\top} A^{(v)},$$

$$y_j = \beta_{(v)} \left[u_v^{(v)} a_{vj}^{(v)} + \sum_{i=1}^{v} a_{m+i,v}^{(v)} a_{m+i,j}^{(v)} \right], \qquad j = v + 1, \cdots, k.$$

Finally,

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

$$a_{vv}^{(v+1)} = -\operatorname{sign}(a_{vv}^{(v)})\sigma_v.$$

These formulas are similar to those given in [3], but are modified to take advantage of the structure of the matrix A. A FORTRAN implementation can be found in [25].

We shall evaluate $\mathbf{D}\mathbf{r}_2(\boldsymbol{\alpha}) = \mathbf{D}P_{\Phi(\boldsymbol{\alpha})}^{\perp}\mathbf{y}$ for a given $\boldsymbol{\alpha}$, according to

$$\mathbf{D}P_{\Phi(\boldsymbol{\alpha})}^{\perp}\mathbf{y} = -(P_{\Phi}^{\perp}\mathbf{D}\Phi)\Phi^{B}\mathbf{y} - (\Phi^{B})^{\top}(P_{\Phi}^{\perp}\mathbf{D}\Phi)^{\top}\mathbf{y}, \tag{5.4}$$

which is readily obtained from (4.2) and (4.5). The matrix Φ^B is constructed as in (3.3).

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 $k \times n$ incidence matrix $E = (e_{jt})$. This matrix will be defined as follows:

$$e_{jt} = \begin{cases} 1 \text{ if and only if parameter } \alpha_t \text{ appears in function } \varphi_j; \\ 0, \text{ otherwise.} \end{cases}$$

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 $t = 1, \cdots, k$,

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

$$\psi_{tj}(\boldsymbol{\alpha}) = \partial\varphi_j(\boldsymbol{\alpha})/\partial\alpha_t, \qquad j = 1, \cdots, n, \quad t = 1, \cdots, k.$$

We write the matrix B in partitioned form

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

where

$$B_t = [\boldsymbol{\psi}_{tj_1}, \boldsymbol{\psi}_{tj_2}, \cdots, \boldsymbol{\psi}_{tj_t}]_{j_i \in S_t}.$$

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

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

(b) Form the $m \times (n + p + 1)$ array

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

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

$$QAS = T_0 = \left[\begin{array}{c|c} T_{11} & T_{12} \\ \hline 0 & 0 \end{array}\right]; \qquad T_{11} = \left[\begin{array}{c} \\ 0 \end{array}\right]_{r \times r}$$

Also $\mathbf{v} = Q\mathbf{y}$; $C = QB$ (T_{11}, T_{12}, $\mathbf{v}$, and C will be stored in the array G). Note again that (see § 3)

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

(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);

$$\mathbf{x} = A^{B}\mathbf{y} = S\begin{bmatrix} T_{11}^{-1}\mathbf{v}_1 \\ \hline \mathbf{0} \end{bmatrix} \qquad \left(\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \hline \mathbf{v}_2 \end{bmatrix}\begin{matrix}\}r \\ \}m-r\end{matrix}\right).$$

$$\text{(e)} \qquad \underset{n\times k}{U} = (P^{\perp}\mathbf{D}\Phi)^{\top}\mathbf{y} = \mathbf{D}\Phi^{\top}Q^{\top}\left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & I_{m-r} \end{array}\right]Q\mathbf{y} = \bar{D}^{\top}\mathbf{v} = C^{\top}\begin{bmatrix} \mathbf{0} \\ \hline \mathbf{v}_2 \end{bmatrix}$$

(transposition in the tensor $\mathbf{D}\Phi$ 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] × G[i, n1];
U[j, t] ← acum
end;).
```

(f) Compute $H_{n\times k} = S^{\top}U$. Solve the $k, r \times r$ lower triangular systems:

$$T_{11}^{\top}W = \tilde{H},$$

where $\tilde{H}_{r\times k}$ contains the first r rows of H. Store W in the first r rows of the $m \times k$ array B. Compute $\bar{D}\mathbf{x}$ and store the nonzero information in the last $m - r$ rows of B.

(g) Finally, the $m \times k$ matrix B is obtained as

$$B \leftarrow \underset{m\times m}{\mathbf{D}P^{\perp}_{\Phi(\alpha)}}\mathbf{y} = -\underset{m\times m}{Q^{\top}}\left\{\begin{bmatrix} W \\ \hline 0 \end{bmatrix} + \bar{D}\mathbf{x}\right\} = -Q^{\top}B.$$

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

Computation of the gradient of the variable projection functional for variable metric minimization procedures. We recall (4.7):

$$\nabla r_2(\boldsymbol{\alpha}) = -2\mathbf{y}^{\top}P^{\perp}_{\Phi(\alpha)}\mathbf{D}\Phi(\boldsymbol{\alpha})\Phi^{-}(\alpha)\mathbf{y}.$$

In order to implement efficiently this formula, we proceed as in the case of $\mathbf{D}P_{\Phi}^{\perp}\mathbf{y}$, which we have just described:

(a) and (b) as before;

(c) as before, except that $C = QB$ is *not* necessary;

(d) $\mathbf{x} = T_{11}^{-1}\mathbf{v}_1$;

(e) $\mathbf{z} = P_{\Phi}^{\perp}\mathbf{y} = Q^{\top}\begin{bmatrix}\mathbf{0}\\ \cdots\\ \mathbf{v}_2\end{bmatrix}$;

(f) $\Psi = -2\mathbf{z}^{\top}\mathbf{D}\Phi$;

(g) $\nabla r_2(\boldsymbol{\alpha}) = \Psi\mathbf{x}$.

In order to use a variable metric minimization procedure such as the Davidon–Fletcher–Powell method [5], it is simply necessary to compute the residual as in § 3 and its gradient as given above.

6. Numerical experiments. We have implemented four different algorithms based on the developments of the previous sections. For each example rank $(\Phi(\boldsymbol{\alpha})) = n$. The methods minimize the variable projection functional $r_2(\boldsymbol{\alpha}) = \|P_{\Phi(\boldsymbol{\alpha})}^{\perp}\mathbf{y}\|^2$ first, in order to obtain the optimal parameters $\hat{\boldsymbol{\alpha}}$, and then complete the optimization according to our explanation in § 2. The algorithms differ in the procedure used for the minimization of $r_2(\boldsymbol{\alpha})$. We also minimize the original functional $r(\mathbf{a}, \boldsymbol{\alpha})$ and compare the results.

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 $\boldsymbol{\alpha}$. This is computed using the results of § 3. In fact, the user has only to give code for filling the matrix Φ for any $\boldsymbol{\alpha}$, 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 $\boldsymbol{\alpha}$. Those functions produce columns in $\Phi(\boldsymbol{\alpha})$ 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 nonvanishing partial derivatives G. See § 5 for a more detailed description.

A3. *Minimization by Marquardt's modification.* It is as explained in § 5 with $F_l \equiv I$. User supplied information is the same as in A2.

A4. *Variable metric procedure.* We have used a FORTRAN program of M. Osborne. The user supplied information is the same as for $A2$ and $A3$, but here only the gradient of $r_2(\boldsymbol{\alpha})$ is required and this is computed according to § 5.

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

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

The functions φ_i are obviously $\varphi_1(\boldsymbol{\alpha}; t) \equiv 1$, $\varphi_{j+1}(\boldsymbol{\alpha}; t) = e^{-\alpha_j t}$, $j = 1, 2$. So the different constants, in the notation of § 2, are $n = 3$, $k = 2$. For the

problem considered, $m = 33$. The *number of constant functions* (NCF) equals 1. The number of nonvanishing partial derivatives p equals 2.

In Table 1 we compare our results for methods A1, A2, A3, A4, and those obtained by minimizing the full functional $r(\mathbf{a}, \boldsymbol{\alpha})$.

TABLE 1
Exponential fit

Method	Functional	Number of Function Evaluations	Number of Derivative Evaluations	Time (seconds)
A1	FF	1832	—	191.00
	VP	100	—	9.00
A2	FF	11	11	5.05
	VP	4	4	3.20
A3	FF	32	26	12.55
	VP	4	4	3.12
A4	VP	27	18*	8.94

$r(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}}), r_2(\hat{\boldsymbol{\alpha}}) \leqq 0.5465 \times 10^{-4}$

* This figure corresponds to gradient evaluations.

P2. *Fitting Gaussians with an exponential background.*

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

The functions φ_j are

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

Thus, $n = 4$, $k = 7$, $m = 65$, $p = 7$. Results for this problem appear in Table 2.

TABLE 2
Gaussian fit

Method	Functional	Number of Function Evaluations	Number of Derivative Evaluations	Time (seconds)
A3	FF	11	9	23.35
	VP	10	8	26.82
A4	VP	72	65	84.34

$r(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}}), r_2(\hat{\boldsymbol{\alpha}}) \leqq 0.048$
Methods $A1$ and $A2$ were either slowly convergent or nonconvergent.

P3. *Iron Mössbauer spectrum with two sites of different electric field gradient and one single line* [21]. The model here is the following:

$$\eta_3(\mathbf{a}, \boldsymbol{\alpha}; t) = a_1 + a_2 t + a_3 t^2 - a_4\left[\frac{1}{1 + ((\alpha_1 + 0.5\alpha_2 - t)/\alpha_3)^2} + \frac{1}{1 + ((\alpha_1 - 0.5\alpha_2 - t)/\alpha_3)^2}\right] - a_5\left[\frac{1}{1 + ((\alpha_4 + 0.5\alpha_5 - t)/\alpha_6)^2} + \frac{1}{1 + ((\alpha_4 - 0.5\alpha_5 - t)/\alpha_6)^2}\right] - a_6\left[\frac{1}{1 + ((\alpha_7 - t)/\alpha_8)^2}\right].$$

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

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

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 3.

TABLE 3
Mössbauer Iron Spectrum

Method	Functional	Initial Values	Number of Function Evaluations	Number of Derivative Evaluations	Time (seconds)
A1	FF	$\boldsymbol{\beta}^0$			*
	VP	$\boldsymbol{\beta}^0$	65	0	70.00
A2	FF	$\boldsymbol{\beta}^0$	4	4	34.34
	VP	$\boldsymbol{\beta}^0$	4	4	41.64
	FF	$\tilde{\boldsymbol{\beta}}^0$	7	7	52.27
	VP	$\tilde{\boldsymbol{\beta}}^0$	6	6	59.60
A3	FF	$\boldsymbol{\beta}^0$	16	16	118.22
	VP	$\boldsymbol{\beta}^0$	3	3	35.35
	FF	$\tilde{\boldsymbol{\beta}}^0$	18	18	130.50
	VP	$\tilde{\boldsymbol{\beta}}^0$	6	6	61.92

$r(\hat{\mathbf{a}}, \hat{\boldsymbol{\alpha}}), r_2(\hat{\boldsymbol{\alpha}}) \leqq 3.0444 \times 10^8$
$(\tilde{\boldsymbol{\beta}}^0 = (80, 49, 5, 81, 24, 9.5, 100, 4)^T)$

* Did not converge in finite amount of time.

The qualitative behavior of the four different minimization procedures used in our computation follows the pattern that has 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 2), 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. The variable metric procedure did not seem to be competitive despite the simplification in the calculation of derivatives.

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 a_j, $\mathbf{a}^0 = \Phi^+(\boldsymbol{\alpha}^0)\mathbf{y}$. This is done in our program for the full functional and also, in the program of Travis with excellent results.

The computer times shown in Table 1 and Table 2 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 $m \times (n + k)$ matrix B as follows: the first n columns consisted of the vectors $\boldsymbol{\varphi}_j(\boldsymbol{\alpha})$ while the remaining columns were the partial derivatives

$$\frac{\partial}{\partial \alpha_l}(\mathbf{y} - \Phi(\boldsymbol{\alpha})\mathbf{a}) = -\sum_{j=1}^{n} a_j \frac{\partial \boldsymbol{\varphi}_j(\boldsymbol{\alpha})}{\partial \alpha_l}, \qquad l = 1, 2, \cdots, k.$$

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. $\mathbf{a}^0 = \Phi^+(\boldsymbol{\alpha}^0)\mathbf{y}$.

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], [20])), 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 Mössbauer spectrum problem, we used two sets of initial values. We used those given by Travis [21], (say) $\boldsymbol{\beta}^0$, and also $\tilde{\boldsymbol{\beta}}^0 \approx \boldsymbol{\beta}^0 \pm 0.05\,\boldsymbol{\beta}^0$. For $\boldsymbol{\beta}^0$, the value of the functional is 3.04467×10^8 while for $\tilde{\boldsymbol{\beta}}^0$, 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 $\boldsymbol{\beta}^0$ 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 P3 with Newton's method and $\boldsymbol{\beta}^0$, 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) will the VP procedure 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 the same minimization algorithm using derivatives is 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 nonlinear problems are ill-defined at this time, it is hard to say whether we have succeeded in obtaining our goal.

Acknowledgments. The authors wish to thank Professor Olof Widlund of the Courant Institute for his careful reading of this manuscript, and Miss Godela Scherer of the Institute Venezolano de Investigaciones Cientificas for programming assistance. We are also pleased to acknowledge the kind hospitality and stimulating conversations with Dr. Hugo D. Scolnik of the Bariloche Foundation, Argentina, where this work was initiated in July 1971. Several helpful suggestions were made by Ms. Linda Kaufman and Mr. Michael Saunders. The authors give special "gracias" to Professor G. W. Stewart, III, who whipped out a proof of a special case of (4.2) during a lecture of one of the authors at the University of Texas.

REFERENCES

[1] YONATHAN BARD, *Comparison of gradient methods for the solution of nonlinear parameter estimation problems*, this Journal, 7 (1970), pp. 157–186.

[2] I. BARRODALE, F. D. K. ROBERTS AND C. R. HUNT, *Computing best l_p approximations by functions nonlinear in one parameter*, Comput. J., 13 (1970), pp. 382–386.

[3] Å. BJÖRCK AND G. H. GOLUB, *Iterative refinement of linear least squares solutions by Householder transformations*, BIT, 7 (1967), pp. 322–337.

[4] RICHARD P. BRENT, *Algorithms for Finding Zeros and Extrema of Functions Without Calculating Derivatives*, Prentice-Hall, Englewood Cliffs, N.J., 1973.

[5] J. W. DANIEL, *The Approximate Minimization of Functionals*, Prentice-Hall, Englewood Cliffs, N.J., 1971.

[6] J. DIEUDONNÉ, *Foundations of Modern Analysis*, Academic Press, New York, 1960.

[7] R. FLETCHER AND SHIRLEY A. LILL, *A class of methods for non-linear programming. II: Computational experience*, Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970, pp. 67–92.

[8] GENE H. GOLUB, *Matrix decompositions and statistical calculations*, Statistical Computation, Roy C. Milton and John A. Nelder, eds., Academic Press, New York, 1969, pp. 365–397.

[9] I. GUTTMAN, V. PEREYRA AND H. D. SCOLNIK, *Least squares estimation for a class of nonlinear models*, Centre de Recherche Mathématique, Univ. de Montréal, 1971, to appear in Technometrics.

[10] RICHARD J. HANSON AND CHARLES L. LAWSON, *Extensions and applications of the Householder algorithm for solving linear least squares problems*, Math. Comp., 23 (1969), pp. 787–812.

[11] L. S. JENNINGS AND M. R. OSBORNE, *Applications of orthogonal matrix transformations to the solution of systems of linear and non-linear equations*, Tech. Rep. 37, Computer Center Australian National Univ., 1970.

[12] WILLIAM H. LAWTON AND E. A. SYLVESTRE, *Elimination of linear parameters in nonlinear regression*, Technometrics, 13 (1971), pp. 461–467.

[13] M. R. OSBORNE, *A class of nonlinear regression problems*, Data Representation, R. S. Anderssen and M. R. Osborne, eds., 1970, pp. 94–101.

[14] ———, *Some aspects of nonlinear least squares calculations*, unpublished manuscript, 1971.
[15] V. PEREYRA, *Iterative methods for solving nonlinear least squares problems*, this Journal, 4 (1967), pp. 27–36.
[16] ———, *Stability of general systems of linear equations*, Aequationes Math., 2(1969), pp. 194–206.
[17] A. PÉREZ AND H. D. SCOLNIK, *Derivatives of pseudoinverses and constrained non-linear regression problems*, Numer. Math., to appear.
[18] G. PETERS AND J. H. WILKINSON, *The least squares problem and pseudo-inverses*, Comput. J., 13 (1970), pp. 309–316.
[19] RADHAKRISHNA C. RAO AND SUJIT KUMAR MITRA, *Generalized Inverse of Matrices and its Applications*, John Wiley, New York, 1971.
[20] H. D. SCOLNIK, *On the solution of nonlinear least squares problems*, Proc. IFIP-71, Numerical Mathematics, North-Holland, Amsterdam, 1971, pp. 18–23; also Doctoral thesis, Univ. of Zurich, 1970.
[21] J. C. TRAVIS, Tech. Note 501, Radiochemical Analysis Section, National Bureau of Standards, Washington, D.C., 1970, pp. 19–33.
[22] G. GOLUB AND G. STYAN, *Numerical computations for univariate linear models*, J. Statist. Comp. and Simulation, to appear.
[23] D. JUPP, *Non-linear least square spline approximation*, Tech. Rep., The Flinders University of South Australia, Australia, 1971, 22 pp.
[24] ———, *Curve fitting by splines as an example of unconstrained non-linear optimization*, Manuscript, 1972.
[25] G. GOLUB AND V. PEREYRA, *The differentiation of pseudoinverses and nonlinear least squares problems whose variables separate*, Tech. Rep. STAN-CS-72-261, Stanford Univ., Stanford, Calif., 1972.
[26] M. KLEIN, Personal communication, Los Alamos Scientific Laboratory, N.M., 1970.
[27] G. W. STEWART, Personal communication, Univ. of Texas, Austin, 1972.
[28] W. KAHAN, Personal communication, Univ. of California, Berkeley, 1971.
[29] PER-ÅKE WEDIN, *Perturbation bounds in connection with singular value decomposition*, BIT, 12 (1972), pp. 99–111.
[30] ———, *On pseudoinverses of perturbed matrices*, Tech. Rep., Department of Computer Science, Lund Univ., Sweden, 1969, 56 pp.
[31] A. BEN-ISRAEL, *A Newton–Raphson method for the solution of systems of equations*, J. Math. Anal. Appl., 15 (1966), pp. 243–252.
[32] G. H. GOLUB AND J. WILKINSON, *Iterative refinement of least squares solutions*, Numer. Math., 9 (1966), pp. 139–148.
[33] G. W. STEWART, *On the continuity of the generalized inverse*, SIAM J. Appl. Math., 17 (1969), pp. 33–45.
[34] H. P. DECELL AND J. FRIES, *On the derivative of the generalized inverse of a matrix*, J. Linear Alg., submitted for publication (1972).
[35] R. H. BARTELS, *Nonlinear least squares without derivatives: an application of the QR matrix decomposition*, CNA-44, Center for Numerical Analysis, Univ. of Texas, Austin, 1972, 47 pp.
[36] M. R. OSBORNE, *Some special nonlinear least squares problems*, Manuscript, 1972.