SIAM J. NUMER. ANAL.
Vol. 15, No. 1, February 1978

A METHOD FOR SEPARABLE NONLINEAR LEAST SQUARES PROBLEMS WITH SEPARABLE NONLINEAR EQUALITY CONSTRAINTS*

LINDA KAUFMAN† AND VICTOR PEREYRA‡

Abstract. Recently several algorithms have been proposed for solving separable nonlinear least squares problems which use the explicit coupling between the linear and nonlinear variables to define a new nonlinear least squares problem in the nonlinear variables only whose solution is the solution to the original problem. In this paper we extend these techniques to the separable nonlinear least squares problem subject to separable nonlinear equality constraints.

1. Introduction. In this paper we will consider the nonlinear least squares problem of finding $\mathbf{a}$ and $\boldsymbol{\alpha}$ which minimize

$$\|\mathbf{y}-\Phi(\boldsymbol{\alpha})\mathbf{a}\|_2^2, \tag{1.1}$$

subject to nonlinear equality constraints of the form

$$H(\boldsymbol{\alpha})\mathbf{a}=\mathbf{g}(\boldsymbol{\alpha}). \tag{1.2}$$

The abbreviated notation of (1.1) has the following meaning:

$$\Phi_{ij}(\boldsymbol{\alpha})=(\phi_j(\boldsymbol{\alpha};t_i)), \qquad i=1,\cdots,m, \quad j=1,\cdots,n,$$

$$\mathbf{a}=(a_1,\cdots,a_n)^T, \quad \mathbf{y}=(y_1,\cdots,y_m)^T, \quad \boldsymbol{\alpha}=(\alpha_1,\cdots,\alpha_k)^T.$$

In (1.2) we have p nonlinear constraints, i.e. $\mathbf{g}(\boldsymbol{\alpha})=(g_1(\boldsymbol{\alpha}),\cdots,g_p(\boldsymbol{\alpha}))^T$, and $H(\boldsymbol{\alpha})$ is a $p\times n$ nonlinear matrix function ($p\leqq n+k$). All the functions involved are assumed to be at least twice continuously differentiable, though somewhat weaker hypotheses could be employed.

In [2], Golub and Pereyra have discussed unconstrained problems of the form (1.1) which they have denoted as "separable nonlinear least squares problems".

Krogh [7] has extended those results to the more general models

$$\|\mathbf{y}-\boldsymbol{\Psi}(\boldsymbol{\alpha})-\Phi(\boldsymbol{\alpha})\mathbf{a}\|_2^2. \tag{1.3}$$

We do not need to introduce $\boldsymbol{\Psi}(\boldsymbol{\alpha})$ explicitly in our present formulation, since it can be included in $\Phi(\boldsymbol{\alpha})\mathbf{a}$ as $a_{n+1}\boldsymbol{\Psi}(\boldsymbol{\alpha})$ provided we add the constraint $a_{n+1}=1$. In [5] one of the authors has introduced more substantial modifications which simplify even further the algorithm.

Constraints of the form (1.2) appear in the applications [6] and they are considered here because of their similarity with (1.1). These problems can be reduced to unconstrained separable problems with a somewhat more complex structure. This is developed in detail in § 2. Once the reduction is performed, one could use any program available for unconstrained separable problems. However, it turns out to be considerably more efficient to devise a completely new algorithm,

* Received by the editors October 31, 1975, and in revised form August, 1976.

† Department of Computer Science, University of Colorado, Boulder, Colorado. Now at Bell Laboratories, Murray Hill, New Jersey 07974.

‡ Department of Mathematics, Universidad Central de Venezuela, Caracas, Venezuela, and Department of Mathematics, California Institute of Technology, Pasadena, California 91125.

taking into account the structure of the problem, as we have done in § 3 of this paper.

It has been shown in practice [2], [7] and there are some theoretical indications [10], that the separation of the linear variables $\mathbf{a}$ from the nonlinear variables $\boldsymbol{\alpha}$ by means of the variable projection method [2], [3], [5], [8] speeds up the convergence of iterative methods used to solve problem (1.1).

We extend in this paper the range of applicability of the variable projection method to constrained problems.

The reduction to an unconstrained problem at the beginning of § 2 was anticipated in [3].

2. The reduction to an unconstrained separable problem. In this section we consider the problem of finding vectors $\hat{\mathbf{a}}$ and $\hat{\boldsymbol{\alpha}}$ which minimize

$$r(\mathbf{a}, \boldsymbol{\alpha}) = \|\mathbf{y} - \Phi(\boldsymbol{\alpha})\mathbf{a}\|_2^2, \tag{2.1a}$$

subject to the nonlinear equality constraints

$$H(\boldsymbol{\alpha})\mathbf{a} = \mathbf{g}(\boldsymbol{\alpha}), \tag{2.1b}$$

where all the vectors and matrices are as in § 1. In what follows, an upper superscript $^+$ on a matrix will denote its Moore–Penrose generalized inverse (see [9]).

In order to guarantee the existence of feasible points we assume that there are vectors $\boldsymbol{\alpha}$ for which the resulting linear systems (2.1b) is compatible; i.e. $\mathbf{g}(\boldsymbol{\alpha}) \in$ range $(H(\boldsymbol{\alpha}))$, or equivalently, $\mathbf{g}(\boldsymbol{\alpha}) = H(\boldsymbol{\alpha})H^+(\boldsymbol{\alpha})\mathbf{g}(\boldsymbol{\alpha})$. The set of all such vectors $\boldsymbol{\alpha}$ will be denoted by A.

For each fixed $\boldsymbol{\alpha} \in A$, the general solution of the resulting system of linear equations (2.1b) is given by [8]

$$\mathbf{a} = H^+(\boldsymbol{\alpha})\mathbf{g}(\boldsymbol{\alpha}) + Y(\boldsymbol{\alpha})\mathbf{z}, \tag{2.2}$$

with $H^+(\boldsymbol{\alpha})^T Y(\boldsymbol{\alpha}) = 0$, and $\mathbf{z}$ varying over all R^{n-r} where r is the rank of H. In other words, the columns of $Y(\boldsymbol{\alpha})$ are a basis for the null space of $H(\boldsymbol{\alpha})$. The set of all pairs $(\mathbf{a}, \boldsymbol{\alpha})$, where $\boldsymbol{\alpha} \in A$, and $\mathbf{a}$ is defined in (2.2), is the feasible set for problem (2.1). Therefore, this problem is equivalent to minimizing in $\mathbf{z}$ and $\boldsymbol{\alpha}$,

$$\begin{aligned} s(\mathbf{z}, \boldsymbol{\alpha}) &= \|\mathbf{y} - \Phi(\boldsymbol{\alpha})H(\boldsymbol{\alpha})^+\mathbf{g}(\boldsymbol{\alpha}) - \Phi(\boldsymbol{\alpha})Y(\boldsymbol{\alpha})\mathbf{z}\|_2^2 \\ &\equiv \|\mathbf{y} - \boldsymbol{\zeta}(\boldsymbol{\alpha}) - G(\boldsymbol{\alpha})\mathbf{z}\|_2^2. \end{aligned} \tag{2.3}$$

where

$$\boldsymbol{\zeta}(\boldsymbol{\alpha}) = \Phi(\boldsymbol{\alpha})H(\boldsymbol{\alpha})^+\mathbf{g}(\boldsymbol{\alpha}) \quad \text{and} \quad G(\boldsymbol{\alpha}) = \Phi(\boldsymbol{\alpha})Y(\boldsymbol{\alpha}).$$

If the dimension $(n-r)$ of $\mathbf{z}$ is nonzero, then we would have a separable problem of the form (1.3). Since we want to apply the variable projection technique we will assume that $r < n$.

This problem could then be solved with any program for unconstrained separable problems by simply giving it the appropriate information. Once $\mathbf{z}$ is computed in the standard fashion, $\mathbf{a}$ is found by an application of formula (2.2). However, we would like to develop a completely new and more efficient algorithm, avoiding all redundant computation.

Let

$$\mathbf{f}(\boldsymbol{\alpha}) = V(\boldsymbol{\alpha})^T(\mathbf{y}-\boldsymbol{\zeta}(\boldsymbol{\alpha})) \tag{2.4}$$

where $V(\boldsymbol{\alpha})$ is an orthogonal basis for the null space of $G(\boldsymbol{\alpha})$. Using a proof similar to Theorem 2.1 of [2], one can show that an $\boldsymbol{\alpha}$ which minimizes $t(\boldsymbol{\alpha}) = \|\mathbf{f}(\boldsymbol{\alpha})\|_2^2$ also minimizes $s(\mathbf{z}, \boldsymbol{\alpha})$. Except for the terms involving $\boldsymbol{\zeta}(\boldsymbol{\alpha})$, the function $t(\boldsymbol{\alpha})$ is similar to $r_3(\boldsymbol{\alpha})$ of equation (4.1) in Kaufman [5]. The Marquardt–Levenberg algorithm applied to $t(\boldsymbol{\alpha})$ using the derivative formula for $r_3(\boldsymbol{\alpha})$ in [3] modified to account for $\boldsymbol{\zeta}(\boldsymbol{\alpha})$, gives the following scheme for generating the required $\boldsymbol{\alpha}$: one starts with an arbitrary $\boldsymbol{\alpha}^{(0)}$ and, until convergence is attained, generates the vectors $\boldsymbol{\alpha}^{(j)}$ by the rule

$$\boldsymbol{\alpha}^{(j+1)} = \boldsymbol{\alpha}^{(j)} - \left(\frac{B}{\nu_j I}\right)^+ \left(\frac{\mathbf{f}(\boldsymbol{\alpha}^{(j)})}{\mathbf{0}}\right)\}k, \tag{2.5}$$

where ν_j is large enough so that

$$\|\mathbf{f}(\boldsymbol{\alpha}^{(j+1)})\|_2 \leqq \|\mathbf{f}(\boldsymbol{\alpha}^{(j)})\|_2$$

and $B = V^T[-D(\boldsymbol{\zeta}) - D(G)G^-(\mathbf{y}-\boldsymbol{\zeta})]$. The operator D represents the Fréchet derivative with respect to $\boldsymbol{\alpha}$, and G^- is any matrix satisfying $GG^-G = G$ and $(GG^-)^T = GG^-$.

Once B and $\mathbf{f}(\boldsymbol{\alpha}^{(j)})$ have been computed, one may efficiently obtain trial values of $\boldsymbol{\alpha}^{(j+1)}$ for various values of ν_j using the algorithm of [2].

A vector $\mathbf{z}$ which minimizes $s(\mathbf{z}, \boldsymbol{\alpha})$ for fixed $\boldsymbol{\alpha}$ is then given by $G^-(\boldsymbol{\alpha})(\mathbf{y}-\boldsymbol{\zeta}(\boldsymbol{\alpha}))$.

The compact formula for B in (2.5) is in terms of $D(\boldsymbol{\zeta})$ and $D(G)$. A more convenient expression for B for the implementation of the algorithm is given in the following theorem:

THEOREM.

$$B = -V^T\{\Phi H^+[-D(H)\mathbf{b} + H^{+T}D(H^T)P_H^\perp \mathbf{g} + D(\mathbf{g})] + D(\Phi)\mathbf{b}\} \tag{2.6}$$

where $\mathbf{b} = YG^{-1}(\mathbf{y}-\boldsymbol{\zeta}) + H^+\mathbf{g}$.

Proof. By (2.3),

$$D(G) = \Phi D(Y) + D(\Phi)Y \tag{2.7}$$

and

$$D(\boldsymbol{\zeta}) = D(\Phi)H^+\mathbf{g} + \Phi D(H^+)\mathbf{g} + \Phi H^+ D(\mathbf{g}). \tag{2.8}$$

Thus, to obtain an expression for B, we need expressions for $D(H^+)$ and $D(Y)$.

Golub and Pereyra [2] have proved that

$$D(H^+) = -H^+D(H)H^+ + H^+H^{+T}D(H)^T P_H^\perp + {}_HP^\perp D(H)^T H^{+T}H^T \tag{2.9}$$

where

$$P_H^\perp = I - HH^+$$

and

$${}_HP^\perp = I - H^+H = YY^T$$

When this formula is inserted into (2.8) and then into (2.5), the last term of (2.9) is canceled since

$$V^T\Phi_H P^\perp = V^T\Phi YY^T = V^T GY^T = 0.$$

To obtain a formula for $D(Y)$ we'll use the orthogonal decomposition of H given by

$$H = Q^T\left(\begin{array}{c|c} T & 0 \\ \hline 0 & 0 \end{array}\right)Z^T \tag{2.10}$$

where Q and Z are orthogonal matrices and T is an $r\times r$ nonsingular upper triangular matrix where H has rank r. It is easy to verify that

$$H^+ = Z\left(\begin{array}{c|c} T^{-1} & 0 \\ \hline 0 & 0 \end{array}\right)Q$$

and that if Z is partitioned as

$$Z^T = \begin{bmatrix} Z_1^T \\ \hline Z_2^T \end{bmatrix}\begin{matrix} \}r \\ \}n-r \end{matrix},$$

then $H^{+T}Z_2 = 0$. Thus one may set $Y = Z_2$.

A formula for $D(Y)$ can be derived using the ideas of § 4 of [5]. From (2.10) we have

$$Z^T H^T Q^T = \left(\begin{array}{c|c} T^T & 0 \\ \hline 0 & 0 \end{array}\right),$$

which, according to § 4 of [5], implies that there exists a matrix M such that

$$\frac{\partial Z_2^T}{\partial \alpha_i} = -Z_2^T\frac{\partial H^T}{\partial \alpha_i}H^{T+} + MZ_2^T. \tag{2.11}$$

This means that

$$D(Y) = -H^+D(H)Z_2 + Z_2M^T.$$

The matrix Z_2 is not unique and M depends on which Z_2 is computed. Fortunately, when (2.11) is inserted into (2.7) and then into (2.5) the term with Z_2M^T is canceled since

$$V^T\Phi Z_2M^T = V^T\Phi YM^T = V^TGM^T = 0.$$

Thus one does not have to be concerned about M.

Combining (2.5), (2.7), (2.8), (2.9) and (2.10) we have

$$\begin{aligned} B &= -V^T\{D(\Phi)H^+\mathbf{g} + \Phi H^+[-D(H)H^+\mathbf{g} + H^{+T}D(H)^TP_H^\perp\mathbf{g} + D(\mathbf{g})] \\ &\qquad + [-\Phi H^+D(H)Y + D(\Phi)Y]G^-(\mathbf{y}-\boldsymbol{\zeta})\} \\ &= -V^T\{\Phi H^+[-D(H)\mathbf{b} + H^{+T}D(H^T)P_H^\perp\mathbf{g} + D(\mathbf{g})] + D(\Phi)\mathbf{b}\} \end{aligned}$$

where $\mathbf{b} = YG^-(\mathbf{y}-\boldsymbol{\zeta}) + H^+\mathbf{g}$. □

The matrix V in (2.4) and (2.6) and the matrix G^- in (2.6) can be computed using the orthogonal decomposition of G given by

$$G = U\left(\begin{array}{c|c} R & S \\ \hline 0 & 0 \end{array}\right)P$$

where U is an orthogonal matrix, P is a permutation matrix and R is a $q\times q$ nonsingular upper triangular matrix. If U is partitioned into

$$U = (\underbrace{U_1}_{q} \vdots U_2),$$

then $G^TU_2 = 0$ so V can be U_2. V can be generated using a sequence of Householder transformations as in [1]. The matrix G^- can be represented as

$$P^T\left(\begin{array}{c|c} R^{-1} & 0 \\ \hline 0 & 0 \end{array}\right)U^T.$$

3. Computational procedure. For a fixed value of $\boldsymbol{\alpha}$, the vector $\mathbf{f}(\boldsymbol{\alpha})$ of (2.4) may be constructed as follows:

1) Determine $\Phi(\boldsymbol{\alpha})$, $H(\boldsymbol{\alpha})$.

2) Determine a complete orthogonal decomposition of $H(\boldsymbol{\alpha})$ by finding orthogonal matrices H and Z such that

$$QHZ = \left(\begin{array}{c|c} T & 0 \\ \hline 0 & 0 \end{array}\right)$$

where T is an $r\times r$ nonsingular upper triangular matrix. As in Golub [1] Q and Z may be the products of Householder transformations designed to reduce H to T. The matrices Q and Z need not be explicitly formed. Only the information required to generate the Householder transformations need be saved.

3) Form the matrix $C = \Phi Z$ by applying the Householder transformations which form Z to the matrix Φ. The last $n-r$ columns of C form the matrix G in (2.3).

4) Determine the orthogonal matrix U and the permutation matrix P such that

$$UC\left(\begin{array}{c|c} I & 0 \\ \hline 0 & \underbrace{P}_{n-r} \end{array}\right) = \left(\begin{array}{c|c|c} M & R & S \\ \hline N & \underbrace{0 \quad 0}_{n-r} & \end{array}\right)$$

where R is a $q\times q$ nonsingular upper triangular matrix. Again U may be the product of Householder transformations and need not be explicitly formed.

5) Compute $\mathbf{d} = H^+\mathbf{g}$ as follows:

(a) Set $\mathbf{a} = Q\mathbf{g} = \begin{pmatrix} \mathbf{a}_1 \\ \hdashline \mathbf{a}_2 \end{pmatrix}^r.$

(b) Solve $T\mathbf{c} = \mathbf{a}_1$.

(c) Set $\mathbf{d} = Z\begin{pmatrix} \mathbf{c} \\ \hdashline 0 \end{pmatrix}.$

6) Compute $\mathbf{f}(\boldsymbol{\alpha})$ by setting

$$\mathbf{p}=U(\mathbf{y}-\Phi\mathbf{d})=\begin{pmatrix}\mathbf{p}_1\\ \hline \mathbf{p}_2\end{pmatrix}\begin{matrix}q\\ m-q\end{matrix}.$$

The vector $\mathbf{f}(\boldsymbol{\alpha})$ is contained in $\mathbf{p}_2$ and hence $\|\mathbf{f}(\boldsymbol{\alpha})\|_2$ is simply $\|\mathbf{p}_2\|_2$.

When B of (2.5) is also required, one should continue the procedure as follows:

7) Compute $D(\Phi(\boldsymbol{\alpha}))$, $D(H(\boldsymbol{\alpha}))$, and $D(\mathbf{g})$.

Usually $D(\Phi)$ and $D(H)$ are tensors with many columns that are zero. Golub and Pereyra [2] describe a scheme for storing only the nonzero columns and determining tensor by vector products using this compact storage arrangement.

8) Compute $\mathbf{b}$ of (2.6) as follows:

(a) Solve $R\mathbf{e}=\mathbf{p}_1$ where R was formed in step 4) and $\mathbf{p}_1$ in step 6).

(b) Set $\mathbf{h}=P\begin{pmatrix}\mathbf{e}\\0\end{pmatrix}$.

(c) Set $\mathbf{b}=Z\begin{pmatrix}0\\ \mathbf{h}\end{pmatrix}+\mathbf{d}$.

9) Set $R=QD(H)$ by applying the Householder transformations which form Q to the nonzero columns of the tensor $D(H)$.

10) Form $J=Q(D(H)\mathbf{b}-H^{+T}(DH)^TP_H^{\perp}\mathbf{g}-\mathrm{D}(\mathbf{g}))$.

(a) Form $(DH)^TP_H^{\perp}\mathbf{g}$ by setting

$$E=R^T\begin{pmatrix}0\\ \mathbf{a}_2\end{pmatrix}.$$

(b) Set $F=Z^TE=\begin{pmatrix}F_1\\ \hline F_2\end{pmatrix}^r$

(c) Solve the k $r\times r$ systems

$$T^TG=F_1.$$

(d) Set $J=R\mathbf{b}-\begin{bmatrix}G\\0\end{bmatrix}-QD(\mathbf{g})=\begin{pmatrix}J_1\\ \hline J_2\end{pmatrix}^r.$

11) The matrix B is finally obtained as follows:

(a) Solve the k $r\times r$ systems

$$TK=J_1.$$

(b) Set $L=UD(\Phi)\mathbf{b}=\begin{pmatrix}L_1\\ \hline L_2\end{pmatrix}\begin{matrix}q\\ m-q\end{matrix}$

(c) $B=-(NK+L_2)$.

4. Algorithm implementation and numerical results for linear constraints. Considerable simplifications arise in the aglorithm of § 3 when $H(\boldsymbol{\alpha})$ is a constant matrix. Since this is the case we have actually implemented as a computer code and for which currently we have practical applications we would like to indicate these simplifications.

Naturally $H(\boldsymbol{\alpha})$ is not evaluated in 1) each time, since H does not depend upon $\boldsymbol{\alpha}$. For the same reason 2) is done once and for all at the beginning of the process. Steps 3)–6) remain the same, while in step 7) $DH(\boldsymbol{\alpha})$ need not be calculated. Step 8) is the same, while 9) is eliminated. In 10) J simply becomes $-QD(\mathbf{g})$, so parts (a)–(d) are eliminated. Step (11) remains.

The algorithm was implemented in FORTRAN and tested on two examples, in both of which H and $\mathbf{g}$ were constant.

The first problem considered was fitting Gaussians with an expoential background, i.e., the model

$$a_1 e^{-\alpha_1 t} + a_2 e^{-\alpha_2(t-\alpha_3)^2} + a_3 e^{-\alpha_4(t-\alpha_5)^2} + a_4 e^{-\alpha_6(t-\alpha_7)^2}$$

is fitted to 65 data points. See [11] for a listing of the data and starting values for $\boldsymbol{\alpha}$. The a's were constrained to the hyperplanes

$$a_1 + 2a_2 + 3a_3 + 4a_4 = 6.27006284,$$

$$a_1 + a_3 = 1.74158318.$$

This problem was chosen in order to verify the correctness of the formulas and the corresponding code.

With these constraints the solution to the problem coincides with the solution to the unconstrained problem which is available in [2], [11].

With 9 function evaluations, 8 derivative evaluations and 3.23 seconds of computing time on a CDC 6400 computer (Run compiler) the residual $r(\mathbf{a}, \boldsymbol{\alpha})$ was reduced to .04013774.

The second problem was supplied by Peter Kirkegaard of the Atomic Energy Commission, Risö, Denmark. Kirkegaard and Eldrup [6] had devised a method for solving separable nonlinear least squares problems with linear constraints on the linear variables which arose in the analysis of positron lifetime spectra. Their algorithm used Marquardt's algorithm based on the fact that for a fixed $\boldsymbol{\alpha}$, the optimal arc could be obtained via the symmetric indefinite system

$$\begin{bmatrix} \Phi^T\Phi & H^T \\ H & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \Phi^T \mathbf{y} \\ \mathbf{g} \end{bmatrix}.$$

Kirkegaard gave the authors an example in which the Φ matrix was given by

$$\Phi_{ij} = x_i^{1/2} \sum_{p=1}^{2} w_p(z_{ijp} - z_{i+1,jp} - \operatorname{erf}\{(t_i - \alpha_4 - d_p)/\sigma_p\} + \operatorname{erf}\{(t_{i+1} - \alpha_4 - d_p)/\sigma_p\}).$$

where

$$z_{ijp} = e^{-\alpha_j(t_i - \alpha_4 - d_p - 1/4\alpha_j\sigma_p^2)}(1 - \operatorname{erf}\{\alpha_j\sigma_p/2 - (t_i - \alpha_4 - d_p)/\sigma_p\}) \quad \text{for } j = 1, 2, 3,$$

$$z_{ijp} = e^{-\lambda(t_i - \alpha_4 - d_p - 1/4\lambda\sigma_p^2)}(1 - \operatorname{erf}\{\lambda\sigma_p/2 - t_i - \alpha_4 - d_p)/\sigma_p\})),$$

and

$$w_1 = 6, \quad w_2 = 4, \quad d_1 = 25/70, \quad d_2 = 0, \quad \lambda = \frac{.07}{1.7},$$

$$\sigma_1 = \frac{.38}{.14\,[\ln(2)]^{1/2}} \quad \text{and} \quad \sigma_2 = \frac{.485}{.14\,[\ln(2)]^{1/2}}$$

and the values of x_i are given in the Table 1 below. Thus $n=4$ and $k=4$. The vector $\mathbf{t}$ was given by

$$t_i = i + 121 \quad \text{for} \quad i = 1, 2, \cdots, 379$$

and the vector $\mathbf{y}$ was given by

$$y_i = \frac{(x_i - 85)}{x_i^{1/2}} \quad \text{for } i = 1, 2, \cdots, 379.$$

TABLE 1
(*x values*)

	101	84	88	99	106	104	100	74	107
105	109	105	99	90	84	112	141	195	419
919	2,099	4,352	7,947	12,952	18,596	24,154	27,804	29,418	28,497
25,948	22,837	19,579	17,241	14,495	12,727	11,264	10,007	9,088	8,115
7,473	6,642	6,131	5,626	5,192	4,691	4,324	4,136	3,677	3,461
3,219	2,977	2,628	2,589	2,394	2,105	1,997	1,967	1,697	1,656
1,511	1,481	1,346	1,264	1,199	1,144	1,029	1,003	910	891
847	737	751	695	644	644	600	548	509	465
447	454	411	371	357	346	320	315	318	292
252	263	264	252	204	205	204	212	190	195
198	169	183	171	168	157	159	147	162	152
153	120	132	157	124	133	106	112	122	129
104	133	116	102	127	101	105	105	121	86
84	100	97	99	94	112	89	98	82	87
82	87	100	95	92	113	88	95	96	104
102	103	87	80	77	85	84	95	107	91
80	94	100	88	80	80	88	90	97	84
75	72	94	77	86	108	63	88	82	106
77	79	87	79	72	67	90	85	89	91
86	89	87	102	93	91	97	90	80	87
82	93	68	82	83	77	76	93	92	90
68	90	102	77	75	93	87	76	72	78
84	79	84	93	83	84	89	88	90	84
91	88	71	93	96	82	89	75	101	70
78	88	72	70	72	81	83	88	92	85
91	84	86	82	84	80	88	76	77	97
85	83	96	92	89	64	91	87	89	85
79	103	80	100	75	88	79	77	82	81
81	104	77	98	92	81	78	88	81	91
78	90	72	95	92	72	77	84	78	88
83	80	76	91	79	83	80	76	75	81
105	80	94	77	102	88	92	79	93	93
82	82	82	87	86	99	78	82	77	90
83	104	85	95	89	94	70	83	84	92
89	99	85	97	99	86	76	98	84	90
86	69	79	98	82	73	78	79	75	78
95	76	75	97	77	103	88	86	90	87
93	76	72	91	85	98	107	99	81	71
90	82	78	80	63	91	93	89	75	80
72	83	82	86	103	90	83	90	96	80

There was only 1 linear constraint for this particular problem:

$$.54a_1+.54a_2+.46a_3+.54a_4=0.$$

Initially $\boldsymbol{\alpha}$ was $(.54, .2, .07, 127.4)^T$ which gave a residual of 1,353.036. The residuals at successive iterations were

426.9649
359.8339
359.1253
359.0751
335.0720

Initially ν_j in (2.5) was set to $(\|B\|_2^2/(m \cdot k))$. On successive iterations ν was half of its previous value. The final $\boldsymbol{\alpha}$ was $(.53777671, .211172, .073373458, 126.92371)^T$ while the final $\mathbf{a}$ was $(32{,}783.984, 52{,}140.2229, 108{,}630.17, 7{,}612.5942)^T$.

Acknowledgment. The authors would like to thank Prof. Gene Golub, Stanford University, for his support and hospitality.

REFERENCES

[1] G. H. GOLUB, *Matrix decompositions and statistical calculations, Statistical Computation*, R. C. Milton and J. A. Nelder, eds., Academic Press, New York, 1970.

[2] G. H. GOLUB AND V. PEREYRA, *The differentiation of pseudoinverses and nonlinear least squares problems whose variables separate*, this Journal, 10, (1973), pp. 413–432.

[3] ———, *Differentiation of pseudoinverses, separable nonlinear least squares problems, and other tales*, Proc. MRC Seminar on Generalized Inverses and their applications, Z. Nashed, ed., Madison, Wisconsin (October 1973), 1976, pp. 303–324.

[4] L. S. JENNINGS AND M. R. OSBORNE, *Applications of orthogonal matrix transformations to the solution of systems of linear and nonlinear equations*, Tech. Rep. 73, Computer Centre, Australian National Univ., Canberra, 1970.

[5] L. KAUFMAN, *A variable projection method for solving separable nonlinear least squares problems*, BIT, 15 (1975), pp. 49–57.

[6] P. KIRKEGAARD AND M. ELDRUP, *Positronfit: A versatile program for analyzing positron lifetime spectra*, Computer Physics Comm., 3 (1972), pp. 240–255.

[7] F. KROGH, *Efficient implementation of a variable projection algorithm for nonlinear least squares problems*, Comm. ACM, 17 (1974), pp. 167–169.

[8] A. PEREZ AND H. D. SCOLNIK. *Derivatives of pseudoinverses and constrained nonlinear regression problems*, Numer. Math., to appear.

[9] R. C. RAO AND S. K. MITRA, *Generalized Inverses of Matrices and its Applications*, John Wiley, New York, 1971.

[10] A. RUHE AND P. A. WEDIN, *Algorithms for separable nonlinear least squares problems*, Stanford University Computer Science Rep. STAN-CS-74-434, Stanford, CA, 1974.

[11] M. R. OSBORNE, *Some aspects of nonlinear least squares calculations*. Numerical Methods for Nonlinear Optimization, F. A. Lootsma ed., Academic Press, London, 1972.