**Title:** Fortran Subroutines for M Estimators in the Linear Model

**Author:** Alfio Marrazi

**Performing Organization Name and Address:**
Princeton University
Department of Statistics
Princeton, New Jersey 08540

**Controlling Office Name and Address:**
U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709

**Report Date:** April 1979

**Distribution Statement (of this Report):**
Approved for public release; distribution unlimited.

**Distribution Statement (of the abstract entered in Block 20, if different from Report):**
TR-147, SER-2

**Supplementary Notes:**
The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

**Key Words:**
- Fortran subroutines
- Computational methods
- Estimators
- Least squares problems
- Linear models

**Abstract:**
This report describes the computational methods used in programming a set of subroutines to solve linear least squares problems and their robust versions by use of M estimators.
FORTAN SUBROUTINES FOR M ESTIMATORS
IN THE LINEAR MODEL

by

Alfio Marrazi
Princeton University

Technical Report No. 147, Series 2
Department of Statistics
Princeton University
April 1979

Research was supported in part by a contract with the U. S. Army Research Office, No. DAAG29-76-G-0298, awarded to the Department of Statistics, Princeton University, Princeton, New Jersey, and in part by a grant from the Swiss National Science Foundation, No. 2.569-0.76.

*On leave from the Fachgruppe für Mathematische Statistik, Eidgenössische Technische Hochschule Zürich, Zürich, Switzerland.
1. **INTRODUCTION**

This report describes the computational methods used in programming a set of subroutines to solve linear least squares problems and their robust versions by use of $M$ estimators. That is, we will concern ourselves with the following problems:

**Problem A**

Given a real $m = n$ matrix $A$ of rank $k \leq \min(m,n)$, and given a real $m$-vector $\mathbf{b}$, find a real $m$-vector $\hat{\mathbf{x}}$ minimizing the Euclidean length of $A\hat{\mathbf{x}} - \mathbf{b}$.  
(We shall use the symbolism $A\hat{\mathbf{x}} = \mathbf{b}$ to denote Problem A.)

**Problem B**

Solve the following system for $\mathbf{z} = (z_1, \ldots, z_n)^T$ and $\phi$:

$$
\begin{align*}
\sum_{i=1}^{m} \left( \frac{z_i}{w_i} \right) a_{ij} z_j &= 0, \quad j = 1 \ldots n \quad (1.1) \\
\sum_{i=1}^{m} \left( \frac{z_i}{w_i} \right) w_i^2 &= \text{CONST} \quad (1.2)
\end{align*}
$$

where

$$
(a_{ij}) = A, \\
z_i = b_i - \sum_{j=1}^{n} a_{ij} z_j
$$

the $w_i$'s are given weights

$$
\phi(x) = \phi'(x); \quad x(x) = x\phi(x) - \phi(x)
$$

and $\phi$ is a given function, which should satisfy

$$
\phi \geq 0, \quad \phi \text{ convex, } \phi(0) = 0 \\
0 \leq \lim \frac{\phi(x)}{|x|} = L < \infty
$$

When all the $w_i$'s are equal to 1, the solution $\hat{\mathbf{x}}$ of Problem B is an $M$ estimate of the parameter vector $\mathbf{a} = (a_1 \ldots a_n)$ in a linear model $\mathbf{b} = A\mathbf{a} + \text{error}$, as defined in L4.

The solution of Problem A may be used as starting value for an iterative algorithm, based on the algorithm $M$ of L4, to solve Problem B.

Problem A is solved by triangularizing $A$. We use Householder transformations as defined in L6. The basic algorithm for their computation and application is algorithm H12 of that book. By this method we can compute the "minimal length solution" when $A$ is not a full rank matrix.

The introduction of the weights $w_i$'s in Problem B allows us to compute $M$ estimates of $\mathbf{a}$ which are optimal for a kind of extremal problem proposed in L1: minimize the trace of the asymptotic covariance matrix over all Fisher-consistent ($M$-) estimators of $\mathbf{a}$ with the same bound on the influence curve considered as a function of $(b,\mathbf{a})$ ("robustness in factor space"). Here $\mathbf{a}$ denotes a general row of $A$, and $b$ the corresponding component of $\mathbf{b}$. We will briefly discuss how the $w_i$'s should be chosen, but the subroutines for their computation as well as for the computation of the estimate asymptotic covariance matrix have still to be written.

The subroutines are written in FORTRAN following the guidelines in L7 in order to facilitate transportability, readability and
reliability. Particular attention has been given to modularity for the following reasons:

- Obtain easily several computational schemes (e.g., perform the initial iterations of the algorithm of Problem 8 using a monotone \( \phi \) and the last ones with a descending \( \phi \), etc.).
- Facilitate further changes in the code.
- Facilitate further additions and particularly the development of a package.

The subroutines are written as parts of a library and belong to three chapters:

1. Main subroutines

   MTRF: upper triangularization of \( A \) and determination of its pseudorank.

   CLLS: solution of Problem A ("Classical Least Squares Solution").

   NREG: iterative algorithm for the solution of Problem 3.

   UCYY: computation of the Unscaled Covariance matrix of the parameter estimates in a regression problem with upper triangular design matrix.

   SCYY: computation of the Scaled Covariance matrix of the parameter estimates.

   RLLS: examples of some possible combination of MTRF, CLLS, NREG, UCYY and SCYY.

2 and 3. Auxiliary and utility subroutines

These subroutines have very specific purposes and will not be described in this report. Only a brief guide in Appendix 1 indicates their purposes.

The code of MTRF, CLLS and UCYY is very similar to the code given in L6 for the same purposes. For this reason the description of MTRF, CLLS and UCYY is brief and need only introduce NREG.

The reader who is already familiar with the mentioned statistical and numerical problems can go directly to the documentation on comment cards. This is available on tape together with the programs.
2. SOLVING THE LEAST SQUARES PROBLEM

The computational methods used in the subroutines MTRF and CLLS are those of L6. The following parts of this book contain the theoretical background: Chapters 1, 2, 3, 10 (p. 53-57), 11, 13, 14, and part of Chapter 25. We consider the principal steps.

(i) Construction of an \( m \times m \) orthogonal matrix \( Q \), of an \( n \times n \) permutation matrix \( P \), and of an \( m \times n \) matrix \( R \) such that

- \( QAP = R \).
- \( R \) is upper triangular (or upper trapezoidal when \( n < m \)).
- the diagonal elements of \( R \) are non-increasing in magnitude.

(ii) Determination of the pseudorank of \( A \). This is the largest index \( j \) such that the absolute value of the element \( (j,j) \) of \( R \) is greater than a given value \( \tau \). (The appropriate selection of the parameter \( \tau \) is discussed in L6, ch. 25. We will set \( \tau = 0 \) if the rank of \( A \) is complete.)

Let \( k \) be the pseudorank of \( A \). Then we write

\[
R = \begin{bmatrix} R1 & R2 \\ 0 & R3 \end{bmatrix} \in \mathbb{R}^{m \times n} \tag{2.1}
\]

with \( R1 \) upper triangular \((k \times k)\)

\( R2 \) \((k \times (n-k)) \)

\( R3 \) \((n-k) \times (m-k)) \).

If \( k = \min(m,n) \) the matrix \( R3 \) does not appear in (2.1). If \( k < \min(m,n) \) only the matrix \( R1 \) will appear in (2.1). In this case, the computations in (iii) are skipped and we set \( W = R1, \ V = I_n \) (here \( I_n \) is the \( n \) by \( n \) identity matrix).

(iii) The choice of a pseudorank \( k < \min(m,n) \) means that \( R3 \) may be ignored. In this case we consider a new least squares problem

\[
Ax = b \quad \text{replacing the original } \ Ax = b, \text{ where}
\]

\[
A1 = Q^T \begin{bmatrix} R1 & R2 \\ 0 & 0 \end{bmatrix} P^T \tag{2.2}
\]

is a new design matrix replacing \( A = Q^T R P^T \). We compute then an \( n \times n \) orthogonal matrix \( V \) (denoted by \( K \) in L6) and a \( k \times k \) matrix \( W \) such that

\[
\begin{bmatrix} R1 & R2 \end{bmatrix} V = \begin{bmatrix} W & 0 \end{bmatrix}, \tag{2.3}
\]

\( W \) is non-singular and upper triangular.

(iv) Compute

\[
k \begin{bmatrix} ab1 \\ ab2 \end{bmatrix} = Qb \tag{2.4}
\]

\( m-k \begin{bmatrix} ab1 \\ ab2 \end{bmatrix} = Qb \tag{2.5} \)

(v) The solutions of the least squares problem

\[
\begin{bmatrix} u & 0 \\ 0 & 0 \end{bmatrix} x^T \begin{bmatrix} ab1 \\ ab2 \end{bmatrix} \tag{2.6}
\]

are of the form

\[
\begin{bmatrix} x1 \\ x2 \end{bmatrix} = \begin{bmatrix} u^{-1} ab1 \ (k \text{ components}) \\ \text{arbitrary} \ (n-k \text{ components}) \end{bmatrix}
\]

Our subroutines set \( x2 = 0 \) so that

\[
\hat{x} = \begin{bmatrix} x1 \\ x2 \end{bmatrix}
\]
is the “minimal length solution” of (2.5).

(vi) Compute

\[ \hat{x} = PV \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = PV x \]

(2.7)

This is the minimal length solution for the problem \( A \hat{x} \approx b \).

(vii) Residuals. Let \( \hat{x} \) be a given vector (e.g. the minimal length solution (2.7)) and let \( y = (PV)^{-1} \). We may be interested in two residual vectors:

\[ z_1 = b - A \hat{x} \quad \text{and} \quad z_2 = b - A x \]

After the transformations described in (iv)-(iv) have been performed, the matrix \( A \) (or \( A_1 \)) and the vector \( b \) are not available in memory anymore. They are essentially replaced by the matrix \( RV \) and by the vector \( Qb \). Therefore, for the computation of \( z_1 \) and \( z_2 \) we must use the following relations. Write

\[ RV = \begin{bmatrix} R1 & R2 \\ m-k & 0 \end{bmatrix}, \quad V = \begin{bmatrix} b_k \\ 0 \end{bmatrix} \]

Hence,

\[ RV \hat{x} = \begin{bmatrix} W_k \\ (R3V) \end{bmatrix} \]

and the components \( n+1, n+2, \ldots, m \) of \( RV \hat{x} \) are all equal to 0. We have

\[ Qb = \begin{bmatrix} W_k \\ (R3V) \end{bmatrix} = \begin{cases} Qz_2 \text{ if } (R3V) \text{ is not ignored} \quad (2.8) \\ Qz_1 \text{ if } (R3V) \text{ is ignored} \quad (2.9) \\ \text{(i.e. if we set } R3=0) \end{cases} \]

\( z_1 \) and \( z_2 \) are computed by inverting these equations.

Particularly:

- If \( y = \hat{x} \), where \( \hat{x} \) is computed as in (2.6), then the first \( k \) components of \( Qz_2 \) are all equal to 0 and

\[ Qb = (0 \ldots 0, gb^T) \]

- If \( k = n \), then

\[ Qb = Qz_2 = \begin{bmatrix} sb_1 - wz \\ gb_2 \end{bmatrix} \]

(viii) The error variance in the model \( A \hat{x} \approx \text{error} \approx b \) may be estimated by

\[ s^2 = \frac{1}{m-k} \begin{bmatrix} (z_1)^T \\ z_1 \end{bmatrix} = \frac{1}{m-k} \begin{bmatrix} (gb_1)^T \\ gb_2 \end{bmatrix} \]

(2.10)

where \( z_1 \) is the residual vector corresponding to \( \hat{x} \) in (2.7).

The matrices \( Q \) and \( V \) are obtained as a product of elementary Householder transformations of the form \( Q_j = I_m + b_j^{-1} u_j u_j^T \) where the vectors \( u_j \) and the scalars \( b_j \) are conveniently chosen.

The determination and the application of these transformations are obtained by the algorithm HFTI:

“For the construction of the \( j \)th Householder transformation, one considers columns \( j \) through \( m \) and selects that one, say, column \( \lambda \), whose sum of squares of components in rows \( j \) through \( m \) is greatest. The contents of columns \( j \) and \( \lambda \) are then interchanged and the \( j \)th Householder transformation is constructed to zero the elements stored in \( a_{ij}, i=j+1 \ldots m \).”

The code of the subroutines HTRF and CLLS is very similar to the one of the subroutine HFTI in L6, p. 290-291. The principal modifications are:
9.

- The introduction of an input parameter check: these parameters must belong to a meaningful range.
- The program parts corresponding to the steps i), ii) and iii) are grouped in MTRF and the others in CLLS.
- Residuals are computed following vii).
- The simultaneous handling of several vectors $b$ is eliminated.
The memory organization of the matrices $A, R, W, Q, V, P$ (and of some auxiliary arrays: $G, H, IP$) is the same as it is described in LG, p. 80.

3. ALGORITHM FOR THE COMPUTATION OF THE ROBUST SOLUTION

We consider now the subroutine HREG to compute the solution of Problem B. HREG implements a modification of the algorithm $H$ of L4 in the special case of a linear problem. The modification consists of the introduction of the weights $w_1$-s and of some minor changes due to the use of the relations (2.1)-(2.7). We will discuss at the end of this section the reason of the introduction of the $w_1$-s and how they should be chosen.

We assume that
- $x$ and $v$ initially have some starting values (these may be given by the minimal length solution of $Alg \ X = b$).
- The transformations $Q, P$ and $V$ are previously computed (and applied to $A$ whose pseudorank is $k$).
- $c$ is a given tolerance and MAXIT a given integer.
- The unscaled covariance matrix (COV) of the estimate $x$ (see below, steps 7 and 8) is previously computed (using UCTV) and given to HREG as a parameter.

HREG performs the following steps:
- Computation of $x = (PV)\cdot^{-1}b$
- Computation of $ab = QB$ (only if $ab$ was not previously computed)
- Computation of $BETA = \left\{ \phi(x)/2 \right\}$ with $\phi(x) = min(-c, min(c, x))$ (c a given parameter), unless the user prefers a different function $\phi$ or a different value of BETA.

In this case the new $\phi$ has to be externally defined together with the new BETA. (See also the remark 3 at the end of this section.)
- Computation of
  
  \[ \text{CONST} = (m-k) \text{BETA}. \]

Then an iterative algorithm is started:

**Step 1.** Set \( \text{NIT} = 1 \)

**Step 2.** Compute

\[
\mathbf{z}_1 = \mathbf{w} - \mathbf{Hy}_k
\]

and \( \mathbf{z}_1 = Q^{-1} \mathbf{z}_1 \) following (2.9)

**Step 3.** Compute a new value for \( \mathbf{a} \) from:

\[
s^2 = \frac{1}{\text{CONST}} \sum_{i=1}^{n} \left( \frac{z_{1i}^2}{w_i^0} \right) (w_i^0)^{-2}
\]

and \( \mathbf{a} = \sqrt{s^2} \)

**Step 4.** "Winsorise" the residuals

\[
z_{1i} = \min\{z_{1i}, 2\text{a}^2 \} (w_i^0)
\]

**Steps 5 and 6.** Compute

\[
\mathbf{z}_1 = Q \mathbf{z}_1 \quad \text{and the vector } \mathbf{z}_1
\]

such that:

\[
\begin{bmatrix}
 z_{11} \\
 \vdots \\
 z_{1n}
\end{bmatrix} = W^{-1}
\begin{bmatrix}
 q_1 \\
 \vdots \\
 q_n
\end{bmatrix}
\]

\[
\begin{bmatrix}
 z_{21} \\
 \vdots \\
 z_{2n}
\end{bmatrix} = W^{-1}
\begin{bmatrix}
 q_{21} \\
 \vdots \\
 q_{2n}
\end{bmatrix}
\]

\[
\begin{bmatrix}
 z_{k1} \\
 \vdots \\
 z_{kn}
\end{bmatrix} = W^{-1}
\begin{bmatrix}
 0 \\
 \vdots \\
 0
\end{bmatrix}
\]

**Step 7.** Set \( \mathbf{x} = \mathbf{x} + \mathbf{z}_1 \)

**Step 8.** Stop iterations and go to 9 if the (transformed)
parameter estimates change by less than \( \epsilon \) times their standard
deviation, i.e., if for \( j = 1 \ldots k \)

\[
|z_{ij}| < \epsilon \sqrt{(\text{COV})_{jj}} \sigma
\]

and the absolute change of \( \mathbf{a} \) is less than \( \epsilon \) or if \( \text{NIT} = \text{MAXIT} \);
otherwise put \( \text{NIT} = \text{NIT} + 1 \) and go to Step 2.

**Step 9.** Compute

\[
\mathbf{x} = PK \mathbf{x}
\]

\( \mathbf{a}_1 \) and \( \mathbf{a}_2 \) following (2.8) and (2.9)

\[\mathbf{a}_2 = PK \mathbf{a}_1 \quad (= \text{last parameter change in the original}
\text{coordinate system}).\]

**Remark 1.** Optimal choice of the weights \( w_i \)s.

We now discuss briefly how the \( w_i \)s have to be chosen in
order to get an estimator which is optimal for an Hampel-extremal
problem. See L1, L2, LS and LB for further information. We consider
the linear model

\[
\mathbf{A} \mathbf{x} + \mathbf{r} = \mathbf{b}
\]

where

\[\mathbf{b} \in \mathbb{R}^n \text{ is a vector of parameters},\]
\[\mathbf{A} \text{ is a matrix of random row vectors in } \mathbb{R}^n \text{ with a}
given distribution (we denote the } i\text{th row of } \mathbf{A} \text{ by}
\]
\[\mathbf{a}_i = (a_{i1}, \ldots, a_{im})^T, \]
\[\mathbf{b} \in \mathbb{R}^n \text{ is the (so called) "observation vector"}, \]
\[\mathbf{r} \in \mathbb{R}^n \text{ is the "error vector"}, \text{ and}
\]
\[\mathbf{r} = \mathbf{N}(0, \sigma^2 I_m), \sigma \text{ is known} \quad (4.1)\]
\[\mathbf{r} \text{ is independent of } \mathbf{A}\]
H-estimators of \( \theta \) are given by a family of functions

\[
g(b,\theta) \in \text{R}^n, \quad g \in \text{R}^n, \quad \theta \in \text{R}^n
\]

into the \( \text{R}^n \) which define the estimate \( \hat{\theta} \) as a selected solution of

\[
\int g(b,\theta) \, d\mathbf{P}_n(b,\theta) = \mathbf{0}
\]

(4.2)

where \( \mathbf{P}_n \) is the empirical measure defined by the "observations" \( (b_1, b_2, \ldots, b_n) \). Fisher consistency means

\[
\int g(b,\theta) \, d\mathbf{P}_n(b,\theta) = \mathbf{0} \quad \text{(for all } \theta \text{)}
\]

(4.3)

where \( \mathbf{P}_n(b,\theta) \) is the family of distributions given by (4.1).

We will now restrict our attention to the family of Fisher consistent H-estimators defined by the functions \( \hat{\theta} \) of the form

\[
\hat{\theta}(b,\theta) = \frac{\theta - \theta_0}{\hat{w}(\theta_0)}
\]

where \( \hat{\theta} \) is a function of \( \theta \) into \( \text{R} \) and \( \theta \) a function of \( \text{R}^n \)

(4.4)

Into this class the influence curve of an estimator \( \hat{\theta} \) is

\[
\hat{\mathbf{I}}C(b,\theta; \hat{\varphi}) = \mathbf{E} \left[ \frac{\partial \hat{\mathbf{g}}}{\partial \mathbf{w}^T} \left( \frac{\partial \mathbf{w}}{\partial \mathbf{w}^T} \right) \right] T \mathbf{w}(\theta) \mathbf{w}(\theta)
\]

where \( \mathbf{E} \) denotes expectation taken for the measure \( \mathbf{P}_n \) and

\[
\varphi(x) = \frac{\partial}{\partial x} \theta(x). \quad \text{Our aim is to minimize the trace of the asymptotic covariance matrix } \mathbf{E} \left[ \hat{\mathbf{I}}C \cdot \hat{\mathbf{I}}C^T \right] \text{ of } \hat{\theta}, \text{ under the condition } \hat{\mathbf{I}}C \leq \mathbf{C}
\]

where \( \mathbf{C} \) is an \( n \times n \) symmetric and non-singular matrix, we get

\[
\hat{\mathbf{I}}C(b,\theta; \hat{\varphi}) = \mathbf{H}^T \mathbf{E} \left[ \frac{\partial \mathbf{w}}{\partial \mathbf{w}^T} \left( \frac{\partial \mathbf{w}}{\partial \mathbf{w}^T} \right) \right] T \mathbf{w}(\theta) \mathbf{w}(\theta) \mathbf{H}^T
\]

(4.5)

where \( \mathbf{w}(\theta) = \mathbf{w}(\mathbf{w}^T \mathbf{H}^{-1}) = \mathbf{w}(\theta) \).

We can now choose \( \mathbf{H} \) so that

\[
\mathbf{H}^T \mathbf{E} \left[ \frac{\partial \mathbf{w}}{\partial \mathbf{w}^T} \left( \frac{\partial \mathbf{w}}{\partial \mathbf{w}^T} \right) \right] T \mathbf{w}(\theta) \mathbf{w}(\theta) \mathbf{H}^T = \mathbf{I}_n
\]

(4.6)

and then it is easy to see that the solution of the extremal problem is given by

\[
\varphi(x) = \varphi_c(x) = \begin{cases} -c & x \leq -c \\ x & -c < x < c \\ c & c \leq x \end{cases}
\]

(4.7)

and

\[
\mathbf{w}(\theta) = 1 \mathbf{w}(\mathbf{w}^T \mathbf{H}^{-1})^{1/2}
\]

(4.8)

If the expectation in (4.6) is taken for the product of the empirical distribution of \( \hat{\mathbf{g}}_1, \ldots, \hat{\mathbf{g}}_n \) with the multinormal distribution of \( \mathbf{g} \), we obtain that the weights \( \mathbf{w}_i = \mathbf{w}(\hat{\mathbf{g}}_i) \) have to be determined by solving for \( \mathbf{H} \):

\[
\mathbf{w}_i = 1 / \left( \mathbf{e}_i \mathbf{m}^T \mathbf{H}_i \mathbf{m} \right)^{1/2}
\]

(4.9)

where \( \mathbf{Diag}(\mathbf{A}) \) denotes a diagonal matrix with elements \( d_1 \ldots d_n \) and

\[
\mathbf{E}_{\varphi} \left[ \mathbf{w}(\hat{\mathbf{g}}_i) \right] = 2 \mathbf{e}(\mathbf{w}_i \mathbf{c}) - 1
\]

(4.10)
with \( g(x) = \int_{2}^{\infty} \exp(-r^2/2) 1/\sqrt{2\pi} \, dr \).  

(4.11)

The system (4.9) has to be solved numerically.

For a slightly different extremal problem this system is replaced by a simpler one. Let \( \hat{g} = (H^T)^{-1}g \) where \( H \) is chosen so that

\[
E \left[ \hat{g}^T \left\{ \frac{r}{w'(\hat{g}')} \right\} \hat{g} \right] = I_n .
\]

(4.12)

Then the influence function of an "M-estimator" \( \hat{\theta} = (H^T)^{-1} \theta \) of \( \theta \) is

\[
I(\theta, \hat{\theta} : \hat{\theta}) = \hat{\theta}^T \left\{ \frac{r}{w'(\hat{g}')} \right\} \hat{\theta} \right) .
\]

(4.13)

In order to minimize the trace of \( E[I(\theta, \hat{\theta} : \hat{\theta})] \) we have to choose \( \phi \) and \( w'(\hat{g}') \) as in (4.7) - (4.8) and (4.9) becomes

\[
A^T \text{Diag} \left[ \frac{c}{\hat{g}' \hat{g}'} \right] - A = I_n .
\]

(4.14)

(4.14) has exactly the form of equation (15.3), p. 41 in L4 by which affine invariant M-estimators of the covariance matrix of \( \hat{\theta} \) are defined. A program for their computation may then be used for the numerical determination of the \( w_i \)'s.

Remark 2.

By combining the main subroutines in different ways we can get several different computational schemes. For example,

- Given starting values \( \hat{x}^0 \) and \( \sigma^0 \), compute a certain number of iterations using a monotone \( \phi \) and a few ones with a re-descending \( \phi \) function (see L3 for a motivation).

- As we assume that the triangularization of \( A \) is computed separately before the call of HREG, it is possible to solve problems economically with the same design matrix \( A \) but different right sides \( \hat{b} \).

With our tools, the simplest starting values, computationally, for \( \hat{x}^0 \) and \( \sigma^0 \), are those produced by CLLS, despite their poor robustness properties. ("Most customers will want to see the least squares result anyway!", see L4.). Some of the possible computational schemes are given as examples in the subroutine RLLS.
Remark 3

When the $w_i$'s are all equal to 1, the value of $\sigma$, computed in step 3 of HREG, is a robust estimate of the error standard deviation which is asymptotically unbiased for normal errors. This value is used to standardize the residuals. If $(m-k)/m$ is not close to 1, it seems better to standardize using $\sqrt{(m-k)/m} \sigma$ instead of $\sigma$. This may be easily done as follows:

- Put the starting value of $\sigma$ (i.e., SIGMA in the parameter list of HREG) equal to an estimate of the residual standard error.
- Use externally defined BETA and $\phi$ such that

$$BETA = \frac{m}{m-k} E(x) = \frac{m}{m-k} \int x(x) \exp(-x^2/2)/\sqrt{2\pi} \, dx.$$ 

In this way the quantities computed in step 3 are

$$s^2 := \frac{1}{m-k} \sum_{i=1}^{m} \left( \frac{x_i}{\sigma} \right)^2 \sigma^2$$

and

$$\sigma := \sqrt{s^2}.$$

(Remember to multiply the final value of $\sigma$ by $\sqrt{m/(m-k)}$ for estimating the error standard deviation).

Another possible standardization of the residuals uses weights

$$w_i = \sqrt{1-h_{ii}}$$

where $h_{ii}$, $i = 1 \ldots m$, are the diagonal elements of the Hat matrix

$$(H) = A(A^T A)^{-1} A^T.$$

The code of a FORTRAN subroutine for the computation of the $h_{ii}$'s is given in Appendix 2.

When the $w_i$'s are not all equal to 1, it may be appropriate to use an externally defined BETA such that

$$BETA = \int x(x/w(x)) w(x)^2 \exp(-x^2/2)/\sqrt{2\pi} \, dx$$

with an appropriate function $w$. 

4. COVARIANCE MATRIX OF THE PARAMETER ESTIMATES

In this section we restrict our discussion to problems with all the \( w_i = s \) in (1.1) - (1.2) equal to 1. Correspondent subroutines for the more general case are still to be written.

A. Unscaled covariance matrix of the parameter estimates in a problem with upper triangular or upper trapezoidal design matrix.

We have seen that the unscaled covariance matrix \((\text{COV})\) of the estimate \( \hat{\beta} \), defined at step 7 of HREG, has to be computed before calling HREG. In the case \( w_i = 1 \) for all \( i \), this matrix coincides with the unscaled covariance matrix of the least squares estimate \( \hat{\beta} \) defined in Section 2, (v). As \( \hat{\beta} \) is set equal to \( 0 \) by CLLS, we define the unscaled covariance matrix of \( \hat{\beta} \) as follows:

\[
(\text{COV}) = \begin{bmatrix}
(w^T w)^{-1} & 0 \\
0 & 0 \\
\end{bmatrix}
\]

The subroutine UCV computes \((\text{COV})\). Its code is taken from the algorithm COV in L6, p. 69.

B. Scaled covariance matrix of the parameter estimates

The parameter estimates in the original coordinate system are \( \hat{\beta} = PV \bar{z} \) (see (2.7)) or \( \hat{\beta} = PV \hat{z} \) (see step 9 in HREG).

Hence their unscaled covariance matrix is

\[
(\text{COV}) = PV(\text{COV})P^T .
\]

The subroutine SCV computes this matrix given the matrices \((\text{COV})\), \( P \) and \( V \) and multiplies it by a scale factor. This may be given as a parameter (e.g., using the value of \( \sigma \) computed by CLLS from (2.10)); otherwise, it is computed as

\[
\sigma^2 = \frac{1}{\hat{m}} \cdot \frac{\text{var} \hat{z} / \left( \text{ave} \hat{z} \right)^2}{\left( \text{ave} \hat{z} \right)^2}
\]

where \( \hat{z} \) is the residual vector computed in step 9 of HREG.

\( \sigma \) is a correction factor (see L4, p. 40):

\[
\sigma = \left(1 + (k/n)\right) \text{var} \hat{z} / \left( \text{ave} \hat{z} \right)^2 / \left( \text{ave} \hat{z} \right)^2
\]

\[
\text{ave} \hat{z} = \frac{1}{\hat{m}} \sum \hat{z} = \frac{\sum \hat{z}}{\hat{m}}
\]

\[
\text{var} \hat{z} = \frac{1}{\hat{m}} \sum \left( \hat{z} - \text{ave} \hat{z} \right)^2
\]

\( \sigma \) is computed in step 3 of HREG.
APPENDIX 1. Guide for the Auxiliary and Utility Subroutines

Auxiliary:
RES: computes residuals following (vii) in Section 2.
R3V: applies Householder transformations with the special
purpose to compute the matrix (R3V) of (vii), Section 2.
SOLVE: solves a triangular system of linear equations (back
substitution) (see 12.6 and Steps 5-6 in HREG).
PVNIX: given x, this subroutine computes \( x = (P\lambda)^{-1}x \)
(see preliminary steps in HREG).
NEWSIG: computes a new value for the robust estimate of the
error standard deviation (see Step 3 in HREG).
HUB: computes "Winorised" residuals (see Step 4 in HREG).
FACR: computes the robust scale factor for the estimate
 covariance matrix.
ADIF: exchanges the contents of two arrays.
DIFF (Function): computes a difference (see L6, p.278).
PSI (Function): computes the Huber function
\[ \phi_c(x) = \max(-c, \min(c,x)) \]
CHI (Function): computes \( \chi(x) = \phi_c(x)^2 \).
PSIPRM (Function): computes the first derivative of \( \phi_c(x) \).
PSIU (Function): computes the function \( \phi(x) \) (see introduction)
and has to be completed by the user.
CHIU (Function): computes \( \chi(x) \) (see introduction) and has to
be completed by the user.
PSIPRZ (Function): computes the function \( \frac{d}{dx} \phi(x) \) and has to be
completed by the user.

Utility:
H12: construction and/or application of a single elementary
Householder transformation (see L6, p.308).
EXCH: exchanges two rows and two columns of a symmetric matrix.
VSV: computes the matrix VSV, where \( V \) is an elementary
Householder transformation and \( S \) a symmetric matrix
stored columnwise in a one-dimensional array.
PHI (Function): cumulative normal distribution.

APPENDIX 2. FORTRAN Subroutine for the Computation of the Diagonal
Elements of the Hat Matrix.

This subroutine uses the output (arrays A and H) of MTRF and
the subroutine H12 to compute the diagonal elements \( HT(1) \ldots HT(m) \)
of the hat matrix

\[ (HT) = A(A^T)\^{-1}A^T. \]

It is supposed that the pseudorank of \( A \) is \( m \). SC is an auxiliary
array. The integer parameters M, N, MDA have the same meaning as in
MTRF.
Method: From \( A = Q^T R P^T \) we get

\[ (HT) = Q^T \begin{bmatrix} I_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} Q \]

Computer Listing:

```
SUBROUTINE HAT (A,HT,H,SC,M,N,MDA)
DIMENSION A(MDA,N),HT(N),H(N),SC(M)
DOUBLE PRECISION SM,DZERO
DZERO = 0.0
DO 100 I = 1,N
DO 100 J = 1,M
20 SC(J) = 0.
SC(I) = 1.
DO 50 JJ = 1,M
J = JJ
50 CALL H12(2,J,J+1,M,A(1,J),1,N(J),SC,1,N,1,1,1)
SM = DZERO
DO 70 J = 1,M
70 SM = SM+SC(J)*DBLE(5SC(J))
100 HT(I) = SM
RETURN
END
```
REFERENCES


L2. F.R. Hampel (1978a), Optimally bounding the Gross-error-sensitivity and the influence of position in factor space. (Invited paper read at the ASA/IMS Meeting in San Diego, Ca., August 16, 1978); Research report No. 18, Fachgruppe fuer Statistik, ETH Zuerich.


