A System of Subroutines for Iteratively Reweighted Least Squares Computations

DAVID COLEMAN, PAUL HOLLAND, NEIL KADEN, VIRGINIA KLEMA, and STEPHEN C. PETERS
Massachusetts Institute of Technology

A description of a system of subroutines to compute solutions to the iteratively reweighted least squares problem is presented. The weights are determined from the data and linear fit and are computed as functions of the scaled residuals. Iteratively reweighted least squares is a part of robust statistics where "robustness" means relative insensitivity to moderate departures from assumptions. The software for iteratively reweighted least squares is cast as semiportable Fortran code whose performance is unaffected (in the sense that performance will not be degraded) by the computer or operating-system environment in which it is used. An ℓ_1 start and an ℓ_2 start are provided. Eight weight functions, a numerical rank determination, a convergence criterion, and a stem-and-leaf display are included.

Key Words and Phrases: least squares, data analysis, mathematical software, portability, linear algebra, curve fitting, robust estimation, weight functions
CR Categories: 5.14, 5.5

1. INTRODUCTION

The purpose of this paper is to describe a system of Fortran subroutines written as modular mathematical software to solve the iteratively reweighted least squares problem. The software includes documentation for use and flow of control as comments in the subroutines. The specifications from which the software was written are contained in [12]. The collection of subroutines uses orthogonal factorizations by Householder transformations or the singular value decomposition from EISPACK II [7] to compute the ℓ_2 start and iterations for reweighted least squares. CL1 [2] computes the ℓ_1 start, an overdetermined solution in the ℓ_1 norm.

The computational tools that we provide include an interactive driver, eight weight functions, John Tukey's stem-and-leaf display [9, 15], and the diagonal of the "hat" matrix [10] which is the projection matrix P_A effectively computed as $U\sum\sum^{+}U^{T}$ from the singular value decomposition [7] or QQ^{T} from QR, the

This research was supported by the National Science Foundation under Grants DCR75-08802 and MCS76-11989.
Authors' present addresses: D. Coleman, Applied Mathematics, Princeton University, Princeton, NJ 08540; P. Holland, Educational Testing Service, Rosedale Road, Princeton, NJ 08540; N. Kaden, Bell Northern Research, Ltd., Ottawa, Ont., Canada K1Y 4H7; V. Klema and S. C. Peters, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139.

Householder transformations. Displaying the ℓ_2 condition of the matrix, the weights, residuals, and the convergence criterion is an option. Two forms of equilibration are also provided [16].

The usual statistical information—the number of observations, number of variables, maximum diagonal element of the "hat" matrix, condition number of the weighted data matrix, the maximum absolute value of the residuals, and the minimum weight—is optionally available. There is an option to provide the (weighted) sum of squared residuals and the sum of absolute residuals. Also available is the (weighted) R-squared statistic, the (weighted) standard error, and the (weighted) F statistic.

The software is available in the form of a basis tape suitable for use by the Fortran converter [1] from IMSL[1] to produce target Fortran code for CDC, Burroughs, Honeywell, PDP-10, and Univac machines. The source code is long precision IBM code acceptable to the Fortran converter. The PFORT verifier [14] was used to check the software.

We do not discuss the theoretical properties of iteratively reweighted least squares or the tuning constants for the weight functions in this paper. For such information we refer the reader to Holland and Welsch [11] and the references therein.

This paper is organized as follows: Section 2 defines the iteratively reweighted least squares problem, Section 3 describes the selection of rank for the data matrix and the reweighted data matrix, and Section 4 gives some numerical results. The weight functions are listed in Table I of Section 2. The subroutines for the computation are listed in Table II of Section 4.

2. ITERATIVELY REWEIGHTED LEAST SQUARES

The method of least squares has been the primary technique for fitting models to data for many years. It is versatile and numerically stable when computationally stable methods are used [13]. Despite its central role in the past, much work has been done by statisticians to improve least squares in the sense of getting more information about the data than is available from just the least squares solution or from the matrix factorizations that are used to obtain it.

The area of work that our software addresses is robust regression which is aimed at analyzing and improving the behavior of least squares estimation when the disturbances are not well behaved. We focus our attention on one of the computational procedures for robust linear regression, i.e., iteratively reweighted least squares.

Consider the model $b = Ax + r$ where b is an $m \times 1$ vector of observations, A is an $m \times n$ data or design matrix, x is an $n \times 1$ vector of parameters, and r is an $m \times 1$ vector. The notation $b = Ax + r$ corresponds to the statistical notation $y = X\beta + \epsilon$ where y is $n \times 1$, X is $n \times p$, β is $p \times 1$, and ϵ is $n \times 1$.

The ordinary least squares problem is $\min_x \sum_{i=1}^{m} ((r_i(x))/s)^2$ where r is a vector of residuals $b - Ax$, and s is a constant or fixed scale.

The weighted least squares problem is $\min \sum_{i=1}^{m} W_u((r_i(x))/s)^2$ which is solved by using ordinary least squares with $W^{1/2}A$ and $W^{1/2}b$ where W is a diagonal matrix of weights that are functions of scaled residuals.

[1] International Mathematical and Statistical Libraries, Inc., Sixth Floor, NBC Building, 7500 Bellaire, Houston, TX 77036.

Table I. Weight Functions (Where u = Scaled Residual), Range, and Default Tuning Constants

Name	$w(u)$	Range	Tuning constant[a]
Andrews	$w_A(u) = \begin{cases} \sin(u/A)/(u/A) \\ 0 \end{cases}$	$\begin{matrix} \lvert u\rvert \le \pi A \\ \lvert u\rvert > \pi A \end{matrix}$	A = 1.339
Biweight	$w_B(u) = \begin{cases} [1-(u/B)^2]^2 \\ 0 \end{cases}$	$\begin{matrix} \lvert u\rvert \le B \\ \lvert u\rvert > B \end{matrix}$	B = 4.685
Cauchy	$w_C(u) = 1/(1+(u/C)^2)$		C = 2.385
Fair	$w_F(u) = 1/(1+\lvert u/F\rvert)$		F = 1.400
Huber	$w_H(u) = \begin{cases} 1 \\ H/\lvert u\rvert \end{cases}$	$\begin{matrix} \lvert u\rvert \le H \\ \lvert u\rvert > H \end{matrix}$	H = 1.345
Logistic	$w_L(u) = \tanh(u/L)/(u/L)$		L = 1.205
Talwar	$w_T(u) = \begin{cases} 1 \\ 0 \end{cases}$	$\begin{matrix} \lvert u\rvert \le T \\ \lvert u\rvert > T \end{matrix}$	T = 2.795
Welsch	$w_R(u) = e^{-(u/R)^2}$		R = 2.985

[a] These are default values for the tuning constants which are designed to have 95 percent asymptotic efficiency with respect to ordinary least squares when the disturbances come from the normal or Gaussian distribution and the scaling function converges to the standard deviation of that disturbance distribution.

The iteratively reweighted least squares problem assumes a start $\hat{x}^{(0)}$, which can be obtained from ℓ_2, ordinary least squares, least absolute residuals, that is to say, the overdetermined solution in the ℓ_1 norm, previous iterations of iteratively reweighted least squares, or a start specified by the user. Given $\hat{x}^{(0)}$, the problem is iterated to obtain the least squares solution $\hat{x}^{(k+1)} = ((W^{(k+1)})^{1/2}A)^+ (W^{(k+1)})^{1/2}b$, where A^+ is the pseudoinverse of A and the diagonal matrix W is computed as a function of scaled residuals at the kth stage. The residual scaling function we use is the median absolute deviation, i.e., the median absolute value of the nonzero residuals. The modularity of the software makes readily possible the inclusion of additional residual scaling functions such as the interquartile range of the residuals. The software to compute the weights includes the eight weight functions listed in Table I.

To test convergence of iteratively reweighted least squares, we use the convergence criterion suggested by Dennis [4]. After the kth iteration, we compute a scale-independent measure of the gradient, $A^T r$, where r is the residuals $b - Ax$, which is

$$\frac{((W^{(k+1)})^{1/2}A_j)^T((W^{(k+1)})^{1/2}r^{(k)})}{\|(W^{(k+1)})^{1/2}A_j\|_2 \|(W^{(k+1)})^{1/2}r^{(k)}\|_2}$$

where $\|\cdot\|$ is the Euclidean norm.

The problem of iteratively reweighted least squares is a problem in optimization in the sense that one is minimizing a function of scaled residuals. The function that is being minimized determines the formula for the weight function used. In general, we minimize $\sum_{i=1}^{m} \rho(r_i(x)/s)$ so that weight function is given by $W(u) = \rho'(u)/\lvert u\rvert$.

3. SELECTING THE RANK OF THE DATA MATRIX

Starting points for the iterations include ℓ_2, ordinary least squares, and ℓ_1, the overdetermined solution in the ℓ_1 norm [2], which corresponds to least-absolute-

residuals regression. The ℓ_2 start and iterations subsequent to the ℓ_1, ℓ_2, or user-supplied start are computed by orthogonal factorizations, i.e., Householder transformation or a combination of Householder transformations and the singular value decomposition.

The way in which we decide to use the QR (Householder transformations) or a combination of QR and MINFIT [7] (least squares solution by singular value decompsition) needs some explanation. Frequently the data matrix A, in the statistical model $b = Ax + r$, has some variables (columns) that are close in the numerical sense to being linear combinations of other columns of A. Since such a situation may occur, the numerical rank [8] of A must be determined before proceeding with the least squares computation. The numerical rank *should* be determined by the user, and the determination of rank should be made with respect to the certainty of the data. Since the rank must be determined at every iteration (reweighting may downweight rows, i.e., observations, to the extent that the effective deletion of observations creates rank degeneracy), it is necessary to estimate the condition of the weighted A as inexpensively as possible. For the ℓ_2 start and for all iterations after any start we default to a QR factorization with column pivoting. Unless A is exactly singular the completion of the QR factorization of A provides the upper triangular factor R whose condition is that of A. The condition estimate using R [3] is only $O(n^2)$ operations, gives a reliable measure of the ill conditioning of A, and is used to determine whether QR is computationally sufficient or whether the computationally more expensive singular value decomposition MINFIT is necessary.

We strongly believe that the user should determine the rank of his data or design matrix by inspecting the singular values of A (which are the same as those of R). However, we provide a conservative default determination of rank associated with the condition of the matrix at each iteration relative to the square root of the precision of the computing machine that is used. Explicitly, the condition estimate of R as obtained by [3] is an estimate of the largest and smallest singular values, $\sigma_{\max}$ and $\sigma_{\min}$, of A. If the ratio $(\sigma_{\min}/\sigma_{\max}) < \epsilon^{1/2}$ where ϵ is the relative precision of the arithmetic of the computing machine, the computation is continued by using the singular value decomposition. When the singular value decomposition is used, the number k of singular values such that

$$\sigma_1 \geq \sigma_2 \geq \cdots \sigma_k > 0, \qquad \sigma_{k+1} = \cdots \sigma_n = 0,$$

is determined from the certainty of the data or from the square root of the precision of the computing machine.

4. SOME NUMERICAL RESULTS

The software to compute the solutions to the iteratively reweighted least squares problem consists of 17,500 lines of code, comments, and documentation for use. The 17,500 lines include all of the software that is required for the interactive driver and its options for use plus a selection of test matrices. The software includes "help" commands to give on-line information to users. The interactive driver is designed to operate effectively on any computer system that permits the transmission of four characters to and from a terminal. The subroutines, however, can also be used in a batch environment. All of the software is designed to be processed by the Fortran converter.

Table II

NAME	DESCRIPTION
EQ02	COLUMN (MAX. ELEMENT) EQUILIBRATION
EQ04	COLUMN (SQRT. SUM OF SQUARES) EQUILIBRATION
EUNORM	EUCLIDIAN (SQRT. SUM OF SQUARES) NORM
GETXL1	GET THE L1 START VECTOR
HMATSV	FORMS DIAG OF H-MAT=U*SIGMA*SIGMA-PSEUDO-INV*U-TRANS
HMATQR	FORMS DIAG OF H-MAT=Q*Q-TRANS
IERRIO	WRITES IERR PARAMETER ON DSET
IFLOOR	INTEGER FUNCTION FINDS FLOOR(REAL) $$
IRLSQR	SUBROUTINE TO DO ONE I R L S ITERATION (QRSOL)
IRLSSV	SUBROUTINE TO DO ONE I R L S ITERATION (MINSOL)
MATIO1	WRITES MATRIX ON DSET, OPTIONAL HEADING
MINFIT	SINGULAR VALUE DECOMPOSITION A=U*SIGMA*V-TRANSP
MINSOL	SOLVES AX=B GIVEN OUTPUT FROM MINFIT
MSG1	WRITES VARIABLE-LENGTH MESSAGE ON DSET
QR1	QR DECOMPOSITION, Q ORTHOGONAL TRANSFORMATIONS
QR1SOL	SOLVES AX=B USING QR1
RESTOR	RESTORE MATRIX WEIGHTS
RESIDL	COMPUTES RESIDUAL B-AX
SCLMAD	SCALE RESIDUALS BY SCALING FACTOR
SLDSPY	DOES STEM AND LEAF DISPLAY (CALLS OTHERS) $$
SLLEAF	DETERMINES STEMS AND LEAVES $$
SLPRNT	PRINTS STEM AND LEAF DISPLAY $$
SLSCAL	DETERMINES SCALE FACTOR AND UNIT FOR DISPLAY $$
SLSORT	SHELL SORT IN INCREASING ORDER $$
SMAD	DETERMINES MAD SCALING FACTOR
START0	USER START FOR I R L S
START1	L1 START (FROM CL1) FOR I R L S
START2	L2 START (FROM MINSOL) FOR I R L S
START3	L2 START (FROM QR1) FOR I R L S
SVMAX	ESTIMATES LARGEST S.V. OF UPPER TRIANGULAR MATRIX
SVMIN	ESTIMATES SMALLEST S.V. OF UPPER TRIANGULAR MATRIX
UNIF01	UNIFORM (0,1) RANDOM NUMBER GENERATOR FUNCTION
WANDRW	ANDREWS WEIGHTING FUNCTION $
WBIWGT	BIWEIGHT (BISQUARE) WEIGHTING FUNCTION $
WCAUCH	CAUCHY WEIGHTING FUNCTION $
WELSCH	WELSCH WEIGHTING FUNCTION $
WFAIR	FAIR WEIGHTING FUNCTION $
WGRAD2	COMPUTES SCALE INDEPENDANT MEASURE OF GRADIENT
WHUBER	HUBER WEIGHTING FUNCTION $
WLOGIS	LOGISTIC WEIGHTING FUNCTION $
WTALWR	TALWAR (ZERO-ONE) WEIGHTING FUNCTION $
WUSER	USER-DEFINED WEIGHTING FUNCTION $
CL1	DOES AN L1 START
CL1BLAS	BASIC LINEAR ALGEBRA ROUTINES FOR CL1
HELP1	PRINTS HELP FOR I R L S
IRHELP	I R L S HELP COMMAND
IRLSDR	INTERACTIVE DRIVER FOR I R L S
IROPN	GETS I1 OPTION FOR I R L S
IROPNN	GETS I2 OPTION FOR I R L S
IROPR	GETS REAL OPTION FOR I R L S
IRPRNT	I R L S PRINT COMMAND
IRPROP	I R L S OPTIONS COMMAND
IRSTAT	I R L S STATISTICS COMMAND
IRSTLF	I R L S STEM&LEAF COMMAND
MAT0A	GETS DATA MATRIX FOR I R L S
MAT0B	GETS RHS VECTOR FOR I R L S
MATRDR	READS A MATRIX AND B VECTOR FROM LOGICAL UNIT 10.
MAT04	A(I,J) = 1 / (1 + ABS(J-I))
MAT05	TLONGLEY DATA
MAT06	TLONGLEY RHS
MAT07	SLONGLEY DATA
MAT08	LONGLEY RHS
MAT09	LONGLEY DATA
OTHER MATXX ROUTINES (MAT20 - MAT47) INCLUDE DRAPER&SMITH PROBLEMS	

$ - WEIGHTING FUNCTIONS USED IN WEIGHTED LEAST SQUARES

$$ - PART OF STEM AND LEAF

We have included a selection of test matrices including those from [5]. We chose this particular collection of matrices because of the widespread use of [5] as a reference and text. Some matrices are cast in integer form and then assigned as floating point numbers to ensure that there is uniform input to a variety of computing machines.

The name and a brief description of the subroutines that are needed for all options of the iteratively reweighted least squares problem are listed in Table II.

Selected results from one of the weight functions, Biweight, and the terminal session used to compute the results applied to [6] are given at the end of this section.

The data matrix given in [6] is

$$A = \begin{bmatrix} 1 & 0.499 & 11.1 \\ 1 & 0.558 & 8.9 \\ 1 & 0.604 & 8.8 \\ 1 & 0.441 & 8.9 \\ 1 & 0.550 & 8.8 \\ 1 & 0.528 & 9.9 \\ 1 & 0.418 & 10.7 \\ 1 & 0.480 & 10.5 \\ 1 & 0.406 & 10.5 \\ 1 & 0.467 & 10.7 \end{bmatrix}, \qquad b = \begin{bmatrix} 11.14 \\ 12.74 \\ 13.13 \\ 11.51 \\ 12.38 \\ 12.60 \\ 11.13 \\ 11.70 \\ 11.02 \\ 11.41 \end{bmatrix},$$

and has singular values 31.6, 0.110, and 0.395.

The software, on tape, for the iteratively reweighted least squares problem is available from IMSL (see footnote 1). The authors of this paper are responsible for any modifications that subsequent use may show to be necessary.

Sample Terminal Session with Some Numerical Results

```
.load irlsdr (start
EXECUTION BEGINS...
I R L S ENVIRONMENT INITIALIZED
FOR LISTING OF COMMANDS TYPE HELP
  SPECIAL OPTION NUMBERS:
    9 (I1) OR 99 (I2) PRINTS HELP
    8 (I1) OR 98 (I2) KEEPS OLD OPTION
TYPE OPTIONS ON A NEW LINE
IRLS COMMAND (A4):
.ihma
OPTION NUMBER? (I1): (9 GETS HELP)
.9
 IHMAT COMMAND (IHMA):     SPECIFY WHETHER DIAGONAL OF H-MATRIX IS TO BE COMPUTED
  H = Q * Q-TRANSPOSE AFTER QRD
  H = U * SIGMA * SIGMA PSEUDO-INVERSE * U-TRANSPOSE
  AFTER SVD
 OPTIONS:
  0 - DON'T COMPUTE IT (DEFAULT)
  1 - COMPUTE IT  (AND PRINT CAN PRINT IT)
IRLS COMMAND (A4):
.ihma
OPTION NUMBER? (I1): (9 GETS HELP)
.1
IRLS COMMAND (A4):
.conv
OPTION NUMBER? (I1): (9 GETS HELP)
.3
IRLS COMMAND (A4):
.iwgt
OPTION NUMBER? (I2): (99 GETS HELP)
.04
```

```
IRLS COMMAND (A4):
.prco
OPTION NUMBER? (I1): (9 GETS HELP)
.0
IRLS COMMAND (A4):
.prco
OPTION NUMBER? (I1): (9 GETS HELP)
.1
IRLS COMMAND (A4):
.prco
OPTION NUMBER? (I1): (9 GETS HELP)
.2
IRLS COMMAND (A4):
.prco
OPTION NUMBER? (I1): (9 GETS HELP)
.3
IRLS COMMAND (A4):
.star
OPTION NUMBER? (I1): (9 GETS HELP)
.2

SIGMA =
 0.3156455D+02  0.1096286D+00  0.3952402D+00

FROM L2-START, X =
 0.1030152D+02  0.8494711D+01 -0.2663214D+00

IRLS COMMAND (A4):
.prin
AFTER ITERATION    0  X =
 0.1030152D+02  0.8494711D+01 -0.2663214D+00

WEIGHTS INITIALLY DEFAULT TO 1.00

   I    RESIDUAL(I)     WDIAG(I)       HDIAG(I)
   1 -0.4442173D+00  0.1000000D+01  0.4178935D+00
   2  0.6868776D-01  0.1000000D+01  0.2418666D+00
   3  0.4129893D-01  0.1000000D+01  0.4172806D+00
   4 -0.1674311D+00  0.1000000D+01  0.6043904D+00
   5 -0.2499867D+00  0.1000000D+01  0.2521824D+00
   6  0.4498504D+00  0.1000000D+01  0.1478688D+00
   7  0.1273257D+00  0.1000000D+01  0.2616385D+00
   8  0.1173894D+00  0.1000000D+01  0.1540321D+00
   9  0.6599797D-01  0.1000000D+01  0.3155106D+00
  10 -0.8915109D-02  0.1000000D+01  0.1873364D+00

GRADIENT (CONVERGENCE LEVEL) =
 0.0            0.0            0.0
IRLS COMMAND (A4):
.iter
** ITERATION    1 DONE
AFTER ITERATION    1  X =
 0.9547460D+01  0.8910478D+01 -0.2087326D+00
PREVIOUS X =
 0.1030152D+02  0.8494711D+01 -0.2663214D+00

   I    RESIDUAL(I)     WDIAG(I)       HDIAG(I)
   1 -0.5368568D+00  0.5553705D+00  0.2082017D+00
   2  0.7821330D-01  0.1000000D+01  0.2927932D+00
   3  0.3745804D-01  0.1000000D+01  0.4977516D+00
   4 -0.1092607D+00  0.9046122D+00  0.6607282D+00
   5 -0.2313761D+00  0.7403244D+00  0.1693049D+00
   6  0.4142602D+00  0.5518823D+00  0.6524162D-01
   7  0.9139889D-01  0.1000000D+01  0.2847991D+00
   8  0.6720272D-01  0.1000000D+01  0.2232470D+00
   9  0.4657812D-01  0.1000000D+01  0.3323426D+00
  10 -0.6521455D-01  0.1000000D+01  0.2655901D+00

GRADIENT (CONVERGENCE LEVEL) =
 0.1038461D+00  0.9706292D-01  0.1202002D+00
IRLS COMMAND (A4):
.step
OPTION NUMBER? (I2): (99 GETS HELP)
.09
IRLS COMMAND (A4):
.iter
** ITERATION    2 DONE
** ITERATION    3 DONE
** ITERATION    4 DONE
** ITERATION    5 DONE
**** CONVERGENCE ****
```

```
IRLS COMMAND (A4):
.prin
AFTER ITERATION    5  X =
 0.9155766D+01  0.9175905D+01 -0.1818402D+00
PREVIOUS X =
 0.9158178D+01  0.9174296D+01 -0.1820108D+00

   I     RESIDUAL(I)      WDIAG(I)       HDIAG(I)
   1 -0.5761166D+00  0.4876692D+00  0.1587391D+00
   2  0.8245666D-01  0.1000000D+01  0.2813957D+00
   3  0.3218103D-01  0.1000000D+01  0.4898618D+00
   4 -0.7396249D-01  0.1000000D+01  0.7124385D+00
   5 -0.2223201D+00  0.7850388D+00  0.1796539D+00
   6  0.3995740D+00  0.5855742D+00  0.7399968D-01
   7  0.8439564D-01  0.1000000D+01  0.2839598D+00
   8  0.4912152D-01  0.1000000D+01  0.2289339D+00
   9  0.4813846D-01  0.1000000D+01  0.3187368D+00
  10 -0.8522368D-01  0.1000000D+01  0.2722807D+00

GRADIENT (CONVERGENCE LEVEL) =
 0.7435491D-04  0.7578489D-04  0.8261316D-04
IRLS COMMAND (A4):
.stat
OPTION NUMBER? (I1): (9 GETS HELP)
.3

 NOBS  =   10
 NOVAR =    3
 SUMW  =   0.8858282D+01
 CONDN =  -0.1000000D+01
 MAXRI =   0.5761166D+00 (  1 )
 MINWI =   0.4876692D+00 (  1 )
 MAXHI =   0.7124385D+00 (  4 )
 SSR   =   0.5734170D+00
 WSSR  =   0.1965639D+00
 SAR   =   0.1653490D+01
 SER   =   0.2862110D+00
 WSER  =   0.1675725D+00
 RSQ   =   0.8920167D+00
 WRSQ  =   0.9558760D+00
 FSTAT =   0.2891243D+02
 WFST  =   0.7582183D+02

 IRLS COMMAND (A4):
 .stem
 OPTION NUMBER? (I1): (9 GETS HELP)
 .1
                =========
                RESIDUALS
                =========


          STEM-AND-LEAF DISPLAY,  N =    10


     1             LO I -0.576D+00


                ( UNIT =   0.1000D-01 )

     2            -2  I 2
     2            -1. I
     2            -1  I
     4            -0. I 87
     4            -0  I
     3             0  I 344
     3             0. I 88


     1             HI I  0.400D+00

 IRLS COMMAND (A4):
 .stem
 OPTION NUMBER? (I1): (9 GETS HELP)
 .2
                =========
                W-MATRIX
                =========
```

```
          STEM-AND-LEAF DISPLAY,  N =     10

               ( UNIT =    0.1000D-01 )

     1              4  I 8
     2              5  I 8
     2              6  I
     3              7  I 8
     3              8  I
     3              9  I
     7             10  I 0000000

IRLS COMMAND (A4):
.stem
OPTION NUMBER? (I1): (9 GETS HELP)
.3
               =========
               H-MATRIX
               =========


          STEM-AND-LEAF DISPLAY,  N =     10


               ( UNIT =    0.1000D-01 )

     1              0. I 7
     1              1  I
     3              1. I 57
     4              2  I 2
     3              2. I 788
     3              3  I 1
     2              3. I
     2              4  I
     2              4. I 8


     1              HI I  0.712D+00

IRLS COMMAND (A4):
.opti

PRINT VECTOR:    1   1   1   0   0   0   0

    VALUE          COMMAND       DESCRIPTION
       5          ****       CURRENT ITERATION
      10          MAXITER    MAXIMUM ITERATION
       1          PRCODE     PRINT OPTION (ON/OFF)
       6          DSET       OUTPUT DATASET
       4          IWGT       WEIGHTING FUNCTION
       1          IHMAT      COMPUTE H-MATRIX
      -1          CONVERGE   CONVERGENCE CHECKING
       2          DEVICE     FOR SAVING WEIGHTS
       2          START      START ALGORITHM
       0          ALGORITHM  ITERATION (LS) ALGORITHM
       0          MODE       CONTROLS STOPPING AFTER STEP
       9          STEPSIZE   NO. OF ITERATIONS PER STEP
       0          ASCALING   A AND B SCALING MODE

 0.1345000D+01    TUNING     TUNING CONSTANT
 0.1018682D+00    RSCALE     SCALE FOR RESIDUALS
 0.1490116D-07    ****       RANK TOLERANCE
 0.1000000D-03    CONVERGE   CONVERGENCE CONSTANT
 0.8261316D-04    ****       CONVERGENCE LEVEL

IRLS COMMAND (A4):
.quit
R; T=0.78/3.27 12:50:39
```

ACKNOWLEDGMENTS

The authors are grateful to many people for their assistance in doing this work. We especially thank John Dennis, Gene Golub, and Roy Welsch for their helpful suggestions. David Hoaglin and Stan Wasserman provided the software for the stem-and-leaf display which gives a useful summary of computational results.

Richard Bartels made available the software for the ℓ_1 start. Alan Cline, Cleve Moler, G. W. Stewart, and J. H. Wilkinson shared with us their technique for estimating the condition number of a matrix. David Gay, with valuable advice from David Hoaglin on random number generators, wrote the code for the condition estimate.

Michael Sutherland and Richard Becker helped to check the software on CDC and Honeywell computers. Douglas Raynor ran several versions of the code on the PDP-10. T. J. Aird and Ed Battiste supplied the Fortran converter from IMSL.

Paul Velleman, after using the interactive driver from its own documentation, made valuable suggestions concerning documentation for use. Maurice Herlihy and Scott Matthews independently used the software and checked parts of the program and documentation for use. Sandra Moriarty provided technical assistance throughout the period in which this work was done.

REFERENCES

1. AIRD, T.J. *The Fortran Converter User's Guide.* IMSL, Houston, Tex., 1975.
2. BARTELS, R., AND CONN, A. Linearly constrained discrete ℓ_1 problems. Tech. Rep. 248, Johns Hopkins Univ., Baltimore, Md., June 1976.
3. CLINE, A., MOLER, C., STEWART, G.W., AND WILKINSON, J.H. On an estimate for the condition number of a matrix. Informal manuscript, 1977.
4. DENNIS, J.E., JR Non-linear least squares and equations. In *The State of the Art in Numerical Analysis.* Academic Press, New York, 1977.
5. DRAPER, N.R AND SMITH, H. *Applied Regression Analysis.* Wiley, New York, 1966.
6. DRAPER, N.R., AND STONEMAN, D. Residuals and their variance patterns. *Technometrics 8* (1966), 695–699
7. GARBOW, B S., BOYLE, J.M , DONGARRA, J.J., AND MOLER, C.B. *Matrix Eigensystem Routines—EISPACK Guide Extension,* vol. 51, Lecture Notes in Computer Science, Springer-Verlag, New York, 1977.
8. GOLUB, G., KLEMA, V., AND STEWART, G.W. Rank degeneracy and least squares problems. Tech. Rep. 456, Univ. Maryland, College Park, Md , 1976, STAN-C5-76-559, Stanford Univ., Stanford, Calif., 1976; Working Paper 165, National Bureau of Economic Research, Inc., New York, N Y , 1977.
9. HOAGLIN, D.C , AND WASSERMAN, S. Automating stem-and-leaf displays. Working Paper 109, National Bureau of Economic Research, Inc., New York, N Y., 1975.
10. HOAGLIN, D.C., AND WELSCH, R.E. The hat matrix in regression and ANOVA. Memo. NS 341, Dep. Statistics, Harvard Univ., Cambridge, Mass , Dec 1976.
11. HOLLAND, P., AND WELSCH, R. Robust regression using iteratively reweighted least squares. In *Communications in Statistics: Theory and Methods, A6 (9)*(1977), pp. 813–827.
12. KADEN, N., AND KLEMA, V. Guidelines for writing semi-portable Fortran Working Paper 130, National Bureau of Economic Research, Inc., New York, N.Y , 1976
13. LAWSON, C L., AND HANSON, R.J *Solving Least Squares Problems.* Prentice-Hall, Englewood Cliffs, N.J , 1974
14. RYDER, B G *The Fortran Verifier User's Guide.* Computing Sci. Tech Rep. 12, Bell Labs., Murray Hill, N.J., 1975.
15 TUKEY, J.W. *Exploratory Data Analysis.* Addison-Wesley, Reading, Mass., 1977
16. VAN DER SLUIS, A. Condition, equilibration and pivoting in linear algebraic systems. *Numer. Math. 15* (1970), 74–86.

Received August 1977, revised December 1977 and September 1979; accepted October 1979