NBS TECHNICAL NOTE 1126

U.S. DEPARTMENT OF COMMERCE / National Bureau of Standards

Problems Used in Testing the Efficiency and Accuracy of the Modified Gram-Schmidt Least Squares Algorithm

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, and the Institute for Computer Sciences and Technology.

THE NATIONAL MEASUREMENT LABORATORY provides the national system of physical and chemical and materials measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; conducts materials research leading to improved methods of measurement, standards, and data on the properties of materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

Absolute Physical Quantities[2] — Radiation Research — Thermodynamics and Molecular Science — Analytical Chemistry — Materials Science.

THE NATIONAL ENGINEERING LABORATORY provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

Applied Mathematics — Electronics and Electrical Engineering[2] — Mechanical Engineering and Process Technology[2] — Building Technology — Fire Research — Consumer Product Technology — Field Methods.

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

Programming Science and Technology — Computer Systems Engineering.

[1]Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Washington, DC 20234.
[2]Some divisions within the center are located at Boulder, CO 80303.

Problems Used in Testing the Efficiency and Accuracy of the Modified Gram-Schmidt Least Squares Algorithm

Roy H. Wampler

Center for Applied Mathematics
National Engineering Laboratory
National Bureau of Standards
Washington, D.C. 20234

U.S. DEPARTMENT OF COMMERCE, Philip M. Klutznick, Secretary

Luther H. Hodges, Jr., Deputy Secretary

Jordan J. Baruch, Assistant Secretary for Productivity, Technology and Innovation

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Issued August 1980

National Bureau of Standards Technical Note 1126
Nat. Bur. Stand. (U.S.), Tech. Note 1126, 83 pages (Aug. 1980)
CODEN: NBTNAE

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1980

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
Price $4.00
(Add 25 percent for other than U.S. mailing)

TABLE OF CONTENTS

Problems Used in Testing the Efficiency and Accuracy of the Modified Gram-Schmidt Least Squares Algorithm

Roy H. Wampler

Center for Applied Matematics
National Engineering Laboratory
National Bureau of Standards
Washington, DC 20234

ABSTRACT

In preparing "Algorithm 544: L2A and L2B, Weighted Least Squares Solutions by Modified Gram-Schmidt with Iterative Refinement" for publication in ACM Transactions on Mathematical Software (Vol. 5, 1979), the Fortran computer program was extensively tested. This note describes the various types of problems which were used to explore the efficiency and accuracy of this algorithm. The Fortran subprograms which performed the various tests are listed in an appendix. Also listed are the data used in executing the testing routines as well as typical output from several different types of problems. Among the testing routines is one which is suitable for handling general linear least squares problems. Here, the user has the option of scaling his raw data in order to mitigate the effects of ill-conditioning.

Key words: Algorithms; curve fitting; least squares; modified Gram-Schmidt; pseudoinverse; regression; statistics; test problems; test results.

1. INTRODUCTION

In preparing "Algorithm 544: L2A and L2B, Weighted Least Squares Solutions by Modified Gram-Schmidt with Iterative Refinement" [23] for publication in ACM Transactions on Mathematical Software, the Fortran computer program was subjected to extensive testing. Section 5 of the companion paper [22] which appeared in the same issue of ACM Transactions on Mathematical Software gave a very brief account of some of the test results and also cited previously published relevant findings of Wampler [21] and other authors. The purpose of the present paper is to present a more detailed description of the various types of problems which were used to explore the efficiency and accuracy of the Gram-Schmidt algorithm.

Attention is given to the execution time required to solve problems of various sizes, to how the choice of a tolerance parameter can affect the computed rank of a matrix, to the effect of heavy weights on computed solutions, and to how one can enlarge the class of problems which this algorithm solves by scaling the raw data. The Fortran subroutines which performed the various tests are listed in an appendix, together with a driver program and input data. Several examples of output generated by the computer program are displayed.

The results described in this paper were obtained chiefly on the Univac 1108 at the National Bureau of Standards under the 1100 Operating System, Versions 33.R2 and 33R3.A (Exec 8). Some results obtained from an IBM 360 Model 65 using the Fortran IV G level 21 compiler are also included. The testing routines, data, and sample output listed in the appendices may prove to be useful in implementing algorithms L2A and L2B on other computing systems.

The reader is referred to [22] and [23] for a complete formulation of the least squares problem, a description of the method of solution, notes on the calling sequences, and a listing of the introductory comments of subroutines L2A and L2B.

The linear least squares problem may be briefly defined [22, p. 458] as follows: Given a real M × N matrix A of rank N1 $\leq$ min(M,N), and given a real M-vector b of observations, find the N-vector $\hat{x}$ of coefficients which minimizes the sum of squares of the residual vector $\delta = b - Ax$. If M $\geq$ N and N1 = N, the solution is unique. If M < N or if N1 < N a unique solution can be obtained by imposing the condition that the vector $\hat{x}$ be of minimal Euclidean norm. In some problems, the solution vector $\hat{x}$ is subject to M1 linear equality constraints. In others, b is a vector of observations whose components have unequal variances; here an M-vector W of weights enters the calculations. (It should be noted that the parameters M, N, N1 and M1 which are here printed in capital letters were printed in lower case italics with subscripts ($\underline{m}$, $\underline{n}$, $\underline{n}_1$, $\underline{m}_1$) in Sections 1, 2 and 3 of [22].)

When matrix A is of rank N and $M \geq N$, algorithms L2A and L2B furnish the same solutions. When matrix A is of rank $N1 < N$, the two algorithms follow different paths and furnish different solutions. Properties of both algorithms are explored in the testing routines discussed in this paper.

In algorithm L2A, if the original matrix A is found to be of rank $N1 < N$, the original matrix A is replaced by a smaller matrix A^* which consists of N1 columns of A found to be linearly independent. A solution is sought for the smaller system of equations.

In algorithm L2B, if the rank of the $M \times N$ matrix A is found to be $N1 < N$, a unique solution is obtained by imposing the condition that $\hat{x}$, the $N \times 1$ solution vector, be of minimal Euclidean norm.

The L2A/L2B algorithms are structured so that solutions can be found for several right-hand sides corresponding to one given matrix A. It is sometimes convenient to refer to the set of one or more b-vectors as a matrix B, and to the set of x-vectors as a matrix X.

2. TESTING ROUTINES AND TEST RESULTS

The Fortran program listed in Appendix A consists of a main (driver) program, nine subroutines and one function. Depending upon two user-furnished input parameters, the main program calls one of five testing subroutines (named TEST1, TEST2, TEST3, TEST4, and TEST5) each of which calls either L2A or L2B for computing solutions after generating or reading input data. Subroutines ORTHO and HADMAR and function GEN are called to construct certain types of input matrices. Subroutine SCALE is called by TEST4 and TEST5 to scale the raw input data when this option is requested by the user. Subroutine OUTPUT prints computed solutions.

Subroutines L2A, L2B and the seven subroutines which they call are not listed here. Introductory comments explaining these algorithms appear in [23]. The complete listing is available in "Collected Algorithms from ACM" [6]. It is also available (as listing, card deck, and magnetic tape) from ACM Algorithms Distribution Service, c/o International Mathematical and Statistical Libraries, Inc., GNB Building, Sixth Floor, 7500 Bellaire Boulevard, Houston, Texas 77036.

The driver program reads in two input parameters, KIND and MODE. The value of KIND (1, 2, 3, 4 or 5) determines which of the five testing subroutines shall be called -- TEST1, TEST2, TEST3, TEST4, or TEST5. The value of MODE (1 or 2) determines whether subroutine L2A or L2B shall be called. MODE 1 corresponds to L2A, and MODE 2 corresponds to L2B.

The first four testing subroutines generate data matrices for certain specific types of least squares problems. Whereas subroutine TEST2 requires no input from the user, subroutine TEST1, TEST3 and TEST4 require some input parameters and/or data in order to construct the matrices A and B which are input to L2A or L2B.

Subroutine TEST5 is a general-purpose testing routine which reads in matrices A and B and other quantities from card images. Weights are read in when the user wishes to assign unequal weights to different observations. TEST5 is suitable for use in production runs as well as for testing and development purposes.

A description will now be given of the various types of problems which were considered and of the results which were obtained from the five testing routines.

2.1 TEST1: Problems Constructed from Orthonormal Vectors

Subroutines TEST1 and ORTHO construct input matrices A and B for least squares problems through the use of a set of orthonormal vectors. The method for generating these vectors is described in Example 2 of Hastings [9]. These problems were used to obtain data on the time required to solve various-sized problems. The timing data is summarized in Table I for a set of 60 problems run on the Univac 1108. Besides giving the total time required for each problem, the table shows the time spent in subroutines DECOMP, SOLVE, and COVAR. ("DECOMP" here denotes DECOM1 in the case of L2A and DECOM2 in the case L2B; "SOLVE" denotes SOLVE1 in the case of L2A, and SOLVE2 and SOLVE3 in the case of L2B.) Most of the remaining time not otherwise accounted for in the table was consumed by generating the data and printing the results. These operations required from 36% (large-sized problems) to 77% (small-sized problems) of the total time.

In all 60 problems the rank was found to equal N. All but two problems converged to a solution after two iterations; the two cases where M = 150, N = 80, M1 = 0 both required three iterations.

The data of element DATA1A were used as input to TEST1 for the 44 problems which called L2A (MODE 1). Element DATA1B gives the data for the 16 problems which called L2B (MODE2). These two sets of data were processed in two separate runs. In each run it was necessary to modify the DIMENSION statements and DATA statements in the driver program (lines 141-145 and 152-158) to handle problems as large as M = 200, N = 80.

In the run which called L2A we used:

```
INTEGER IPIVOT(80)
REAL A(200,81), B(200,1), C(1682)
REAL Q(200,80), R(80,80), QR(1,1)
REAL RES(200,1), SDX(80), SF(81), W(200), X(80,1)
```

```
      LOGICAL FAIL(1)
      DATA MM, NN, LL, MMPNN, NNP1, NNPLL, MNL
     * / 200, 80, 1, 1, 81, 81, 1682 /
```

The run which called L2B used the same dimensions and dimensioning parameters as those used in the L2A run except for

```
      REAL Q(1,1), R(1,1), QR(280,80)
```

and in the DATA statement MMPNN was set equal to 280 rather than 1.

The timing data given in Table I were obtained from two batch runs (one for MODE 1, one for MODE 2) made on February 16, 1978, at night under N (lowest) priorty on the Univac 1108 at NBS. These two jobs were repeated on two other occasions in a similar environment, and the details of the timing were quite similar to those given in Table I. Runs made during the day (prime time) in batch mode and runs made in interactive mode could be expected to require different amounts of time -- probably greater.

The chart below gives the total run time and the central processor unit (CPU) time required for execution of these two jobs on three different occasions.

		MODE 1 (L2A) 44 Problems	MODE 2 (L2B) 16 Problems
Run 1	Date	2-16-78	2-16-78
	Total time	4 min. 11.268 sec.	2 min. 54.677 sec.
	CPU	3 min. 40.461 sec.	2 min. 32.982 sec.
Run 2	Date	2-27-80	8-28-79
	Total time	4 min. 1.003 sec.	3 min. 0.761 sec.
	CPU	3 min. 28.232 sec.	2 min. 37.009 sec.
Run 3	Date	2-29-80	2-27-80
	Total time	4 min. 5.124 sec.	2 min. 53.016 sec.
	CPU	3 min. 28.907 sec.	2 min. 29.238 sec.

2.2 TEST2: Problems Having Rank Deficiencies

Subroutine TEST2 and function GEN generate two sets of test problems as described by Lawson and Hanson [13, pp. 252, 277]. The first set (18 cases) was constructed from a sequence of integers having a short period so that some of the matrices are mathematically rank-deficient. The second set of 18 cases is the same as the first set except that "noise," simulating data uncertainty, has been added to all data values.

These two sets of problems were run using subroutine L2B with two different values of the tolerance parameter TOL, namely 0.0 and 0.5. The results obtained from L2B were compared with results from three of the programs given in [13] which furnish least squares solutions: PROG1, PROG2 and PROG3. PROG1 and PROG2 both use Householder transformation algorithms, whereas PROG3 uses a singular value decomposition. The results summarized below were all obtained on a Univac 1108.

For the problems of the first set, whether TOL was set equal to 0.0 or 0.5, the computed rank agreed with that given by PROG2 and PROG3. In six of the 18 cases, $N1 < \min(M, N)$. The agreement of computed coefficients obtained by the two programs L2B and PROG2 ranged from 5 to 8 digits. Agreement between L2B and PROG3 was also 5 to 8 digits. The setting of TOL = 0.5 in L2B corresponds to TAU = 0.5 in PROG2 and PROG3.

For the problems of the second set, where noise was added to all elements of A and B, when TOL was set equal to 0.5, the ranks computed by L2B again agreed with those computed by PROG2 and PROG3. When TOL was set equal 0.0, the rank in all cases was computed to be min(M, N). This was in agreement with PROG1 in all problems for which PROG1 gave results. The agreement of coefficients obtained by the two programs L2B and PROG2 ranged from 6 to 8 digits. Between L2B and PROG3, as well as between L2B and PROG1, agreement ranged from 1 to 8 digits.

Table II lists the solutions, computed ranks and Euclidean norms of residuals for the test case M = 7, N = 6, with the two different noise levels. These results are very similar to those reported in Table C.3, page 252 of [13]. (It should be noted that 6 × 6 matrix of coefficients in Table C.3 of [13] gives incorrect signs in three instances. The (3,3), (2,5) and (3,5) entries should read, respectively, +0.0833, -0.0217 and +0.0846.)

No input data is required for subroutine TEST2. One can construct additional problems of this type by changing one or more of the parameters NOISE, ANOISE, MN1 and MN2 in TEST2.

2.3. TEST3: Problems Constructed from Hadamard Matrices

Subroutines TEST3 and HADMAR construct least squares problems from Hadamard matrices by a method due to Richard J. Hanson [8] which makes use of the singular value decomposition of a rectangular matrix. In these examples, the problem size dimensions were varied, the condition number of the matrix A was varied and the data uncertainty in the right-hand sides was varied. Such problems can be devised in the following manner:

(a) Let $A = U S^* V^T$

where A is M × N, U is M × M, S^* is M × N, V^T is N × N, with $M = 2^{k_1}$, $k_1 = 1,2,3,4$, and $N = 2^{k_2}$, $k_2 = 1,\ldots, k_1$.

(b) The matrices U and V are symmetric Hadamard matrices. (An Hadamard matrix H of order n is a matrix of +1's and -1's such that $HH^T = nI$; any two rows of H are orthogonal. Cf. Hall [7, pp. 204-221]. For order 2 one can take

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Hadamard matrices of orders 4, 8 and 16 can be constructed from lower order Hadamard matrices. Let $H_4 = H_2 \times H_2$, $H_8 = H_2 \times H_4$ and $H_{16} = H_4 \times H_4$, where $\times$ denotes the Kronecker product of two matrices.

(c) Let k_3 be a positive integer, and let

$$S^* = \begin{bmatrix} S \\ 0 \end{bmatrix}$$

where S is a diagonal $N \times N$ matrix whose elements are uniformly distributed as integers on $[1, 2^{k_3}]$. Set one of the diagonal elements equal to 2^{k_3} and another equal to 1. The largest and smallest singular values of S are thus 2^{k_3} and 1, and hence the condition number of S is 2^{k_3}. It is easily shown that the condition number of A is also equal to 2^{k_3}. Values chosen for k_3 depend upon the computer precision available.

(d) Let $\tilde{x} = (1,1,\ldots,1)^T$ and $\hat{b} = A\,\tilde{x}$. Next, compute a residual vector

$$\delta = U \begin{bmatrix} 0 \\ \rho \end{bmatrix} \begin{matrix} \}\, N \\ \}\, M-N \end{matrix}$$

where ρ is a vector of integers uniformly distributed on $[0, ||\hat{b}||]$, and $||\hat{b}||$ denotes the Euclidean norm of $\hat{b}$. With A defined as above, we have $\hat{b} = (N,N,\ldots, N)^T$ and $||\hat{b}|| = NM^{1/2}$.

(e) Finally, compute

$$b = \hat{b} + \alpha\delta \text{ for four values of } \alpha, \text{ namely } 0,\ 0.01,\ 1,\ 100$$

The true solution for all problems constructed in this manner is $\hat{x} = (1,1,\ldots,1)^T$.

Sets of Hadamard-type problems were run on the Univac 1108 and the IBM 360/65. On the Univac 1108, where the machine-dependent parameter ETA was set equal to 2^{-26}, we chose k_3 = 12, 16, 20, 24, 25. On the IBM 360/65, with ETA set equal to 16^{-5}, we chose k_3 = 12, 16, 20. (ETA is the smallest positive real number such that 1.0 + ETA > 1.0 in floating-point arithmetic.) In all cases we used subroutine L2A with TOL = 0. The values of ρ, having been obtained from [15] and [16], were read in from data elements (DATA3A, DATA3B, DATA3C, DATA3D, DATA3E). DATA3A was used with k_3 = 12; DATA3B with k_3 = 16; DATA3C with k_3 = 20; DATA3D with k_3 = 24; DATA3E with k_3 = 25.

Tables III and IV summarize the results for four values of k_3 on the Univac. In Table III, results for ten combinations of M and N are condensed on each row of the table, whereas Table IV gives greater details for the cases where k_3 = 24. That the error in the computed residuals was sometimes as great as 8 units in the seventh significant digit when α = 0.01 is explained by the fact that the elements of b in these cases were not integers and could not, in general, be exactly represented within the computer as octal fractions. In all other examples covered by Tables III and IV, the elements of A and b were integers which could be converted exactly to octal fractions.

For the problems where k_3 = 25, in seven of the ten (M,N)-combinations, the solution failed to converge for all values of α. For M = 16, N = 16 the computed rank was found to be 15, and a solution was then given for the problem of reduced size. In the cases where M = 8, N = 8 and M = 16, N = 8, solutions were computed, but here the matrices A as represented in the computer were perturbed versions of the precise problems defined in steps (a) through (e) above. The largest element of A in both these cases, when computed without rounding error, is 134217729. By printing out matrix A in octal format, it could be seen that this number was represented in the Univac as 134217728. The exact solution for this perturbed problem has all elements of $\hat{x}$ equal to 8/7. The computed coefficients (1.1428571) for M = 8, N = 8 and for M = 16, N = 8, α = 0, 0.01, 1 were thus correct to 8 digits; for M = 16, N = 8, α = 100 the computed coefficients were in error by 5 units in the eighth significant digit.

Table V summarizes the results for problems run on an IBM 360/65, with k_3 = 12, 16, 20. In all cases where k_3 = 20, the matrix A was found to be rank deficient, with N1 = N -1. Solutions were then computed for the problems of reduced size.

For one of the Hadamard-type problems run on the Univac 1108, further details are now given. Here, M = 8, N = 4; k_3 = 24; the diagonals of S were $s_1 = 1$, $s_2 = 5592406$, $s_3 = 11184811$, $s_4 = 16777216$; $\rho = (10, 7, 2, 10)^T$; and $\alpha = 1$.

We obtain

$$A = \begin{bmatrix} 33554434 & -11184810 & -22369620 & 0 \\ -11184810 & 33554434 & 0 & -22369620 \\ -22369620 & 0 & 33554434 & -11184810 \\ 0 & -22369620 & -11184810 & 33554434 \\ 33554434 & -11184810 & -22369620 & 0 \\ -11184810 & 33554434 & 0 & -22369620 \\ -22369620 & 0 & 33554434 & -11184810 \\ 0 & -22369620 & -11184810 & 33554434 \end{bmatrix}, \quad b = \begin{bmatrix} 33 \\ -1 \\ 9 \\ 15 \\ -25 \\ 9 \\ -1 \\ -7 \end{bmatrix}.$$

With ETA = 2^{-26} , TOL = 0.0, the computed rank was N1 = 4 and a solution was obtained after 8 iterations.

Computed results were:

$$\hat{x} = \begin{bmatrix} .99999999 \\ .99999999 \\ .99999999 \\ .99999999 \end{bmatrix}, \quad \text{diagonals of covariance matrix} = \begin{bmatrix} .0071733 \\ .0071733 \\ .0071733 \\ .0071733 \end{bmatrix},$$

$$\delta = \begin{bmatrix} 29.000000 \\ -5.0000000 \\ 5.0000000 \\ 11.000000 \\ -29.000000 \\ 5.0000000 \\ -5.0000000 \\ -11.000000 \end{bmatrix}.$$

Accurately calculated to five significant digits the diagonal terms of the unscaled covariance matrix are .0078125; the computed values, .0071733, are correct to only 1.09 digits. This accuracy is close to the value of DIGITX reported for this problem, 1.33. Recall that DIGITX [22, p. 463] is an estimate of the accuracy of the initial solution of $\hat{x}$, before iterative refinement. There is no iterative refinement in the computation of the unscaled covariance matrix.

2.4 TEST4: Problems with Heavy Weights, Unscaled and Scaled

In the problems considered thus far, all components of the observation vectors were assumed to have equal variances. We now consider problems where some observations are assumed to have smaller variances than others, and hence have greater weights. These examples were constructed by Björck [3] from the polynomial data of Wampler [20].

(a) Let $A = (a_{ij})$ be the 21×6 matrix given by

$$a_{ij} = 1 \text{ if } i = j = 1,$$

$$a_{ij} = (i-1)^{j-1} \text{ otherwise.}$$

This corresponds to fitting a fifth-degree polynomial to observations at the points 0,1,2,..., 20.

(b) Let $A_w = D_w A$ where

$$D_w = \text{diag}\ (w,1,1,\ldots,1,w,1,\ldots,1,w)$$

is a diagonal matrix which scales the three rows 1, 11, and 21 of A with the weight $w > 1$.

(c) Table 1 of [20] lists a vector of observations called Y2 and a vector of residuals called Δ. Let $y = 10^5 \cdot Y2$. The vector of observations b_w is chosen as

$$b_w = D_w y + w D_w^{-1} \Delta.$$

(d) Let $w = 2^{i-1}$, $i = 1,2,\ldots,p$, where the choice of p depends upon the computer precision available. We thus have a sequence of problems $A_w x \cong b_w$ with moderate to large residuals (depending on the magnitude of w), the exact solution of which is

$$\hat{x} = (10^5, 10^4, 10^3, 10^2, 10, 1)^T.$$

(e) Consider also modifications of the problem defined thus far in which the columns of A_w are scaled in the following manner.

(i) By $G = \text{diag}(10^6, 10^5, 10^4, 10^3, 10^2, 1)$.

Then $(A_w\ G)\ x_G \cong b_w$ has the solution

$$\hat{x}_G = (10^{-1}, 10^{-1}, 10^{-1}, 10^{-1}, 10^{-1}, 1)^T$$

from which one can calculate $\hat{x} = G\ \hat{x}_G$.

(ii) By $H = \text{diag}(||a_1||^{-1}, ||a_2||^{-1},\ldots, ||a_6||^{-1})$ where $||a_j||$ denotes the Euclidean norm of the jth column of A_w. From $\hat{x}_H$, the solution to $(A_wH)\ x_H \cong b_w$, one can obtain $\hat{x} = H\ \hat{x}_H$. Under this scaling all columns of A_wH have unit length. (Cf. Lawson and Hanson [13, pp. 185-188] and Björck [3].)

Note that for w = 1 the elements of A_w and b_w are integers having up to seven significant digits. As w increases, the number of significant digits, especially in the weighted elements of b_w, increases.

Subroutine TEST4 was used to construct the matrices A and b_w by reading in the data of element DATA4 and applying the formulas given in (a), (b), (c), and (d) above. Subroutine SCALE was used to apply the scaling described in (e) above.

For the problem defined in steps (a) through (d) above, solutions from L2A obtained on a Univac 1108 with TOL = 0.0 reported the computed rank N1 to be 6 for w = 1, 2 and 4, but $N1 \leq 5$ was reported for $w \geq 8$. With scaling (i) of step (e) introduced, solutions with N1 = 6 were obtained for $w \leq 2^{14}$. With scaling (ii), solutions with N1 = 6 were obtained for $w \leq 2^{16}$.

Example: $w = 2^{14} = 16384$, scaling (ii), TOL = 0. Two kinds of perturbations occurred in expressing the data of this problem on the Univac. Firstly, there were two instances of integers having too many significant digits to be exactly expressible in octal represenation. These were the first and last elements of b_w, equal to 1,638,400,759 and 103,219,200,759, which were converted to 1,638,400,752 and 103,219,200,000, respectively, in octal representation. Secondly, the scaling caused the elements of A_wH to be decimal fractions between zero and one. Such numbers cannot, in general, be converted exactly to octal fractions. The exact solution to the perturbed problem, obtained with the aid of a double precision program, is listed below, together with the solution to which L2A converged after four iterations.

Exact solution (to 8 digits)	Solution computed by L2A
100000.00	99999.998
10035.918	10035.917
948.16813	948.16848
102.26114	102.26109
9.8705418	9.8705446
1.0025746	1.0025745

In discussing weighted problems such as the example considered here, Björck [3] cautioned that the iterative refinement procedure would fail to converge for extremely large values of w. In this particular example we note, however, that rank-deficiency was reported for the larger values of w, and reports of failure to converge were not encountered.

2.5 TEST5: General Least Squares Problems

Subroutine TEST5 is a general-purpose routine which reads in matrices A and B as well as vector W if the observations are to be weighted. Input parameters which are read in by TEST5 are:

ISCALE	Denotes whether the raw data are to be scaled (by calling subroutine SCALE) and if so, what type of scaling is to be applied.
M	Total number of equations.
N	Number of unknown coefficients.
M1	Number of linear constraints. (If M1 > 0, the first M1 equations are to be satisfied exactly.)
L	Number of right-hand sides (vectors of observations).
ITYPE	Parameter which specifies whether a polynomial type fit or a multiple regression fit is to be performed.
IWGHT	Parameter which specifies whether or not weights are to be read in.
TOL	Parameter determining the type of tolerance to be used in determining rank.

Further remarks on these input parameters are given in the listing of TEST5 in Appendix A as well as in Wampler [22; 23].

Appendix B lists data elements DATA5A, DATA5B, DATA5C, DATA5D and DATA5E which were used in conjunction with TEST5.

Through the use of Sande's Fortran execution profiler [17] it was found that on the Univac 1108 each executable statement of L2A and L2B (and the seven subroutines called by them) was used at least once in executing the 24 problems listed in DATA5A and DATA5B.

Element DATA5C contains five problems, a subset of DATA5A, for which the results from L2B are different from those of L2A.

Elements DATA5B, DATA5D and DATA5E illustrate that problems with zero weights can be easily handled by this algorithm, and that the computed rank of the system of equations can be affected by scaling the raw data.

The first example of DATA5D, taken from Bennett and Franklin [2, pp 379-385] illustrates how orthogonal designs with missing observations can be treated by assigning weight zero to the missing cells. Whenever the I-th weight W(I) = 0, the I-th rows of A and B are ignored in computing the solution vector and the residual sum of squares. Residuals, however, are computed corresponding to observations given zero weights.

The second example of DATA5D, taken from Hogben, Peavy and Varner [10, pp. 134-135], is one where 5 of the 12 observations are given zero weight. It was reported in [10] that the fact that the system of equations is singular was not detected by OMNITAB II in connection with a twoway analysis of variance. OMNITAB 80 [11], however, which uses an adaptation of algorithm L2A for least squares fits and for twoway analysis of variance detects and reports to the user the fact that this system is singular.

The last two examples (Nos. 27 and 28) of DATA5D are, respectively, sixth and fifth degree polynomial-type problems. On the Univac 1108, using unscaled data (ISCALE = 0), both problems were reported to be rank-deficient. The computed ranks were 4 and 3, respectively. If the user specifies that input matrices A and B should be automatically scaled (ISCALE = 2), algorithm L2A reports for both problems that matrix A is of full rank. Computed results for the fifth degree polynomial, with and without scaling, are shown in Appendix C.

Another example of computed output in Appendix C is problem No. 22 from element DATA5B. This illustrates several features of the L2A/L2B algorithms:

(1) Linear constraints: The first 3 of the 14 equations are to be satisfied exactly.

(2) Unequal weights: The observations are assumed to have different variances, hence different weights are assigned.

(3) Zero weights: The last two observations are given zero weights. Hence the last two rows of A and B are ignored in computing the solution vector of coefficients, but residuals are computed for these two observations.

3. STRUCTURE CHART

A structure chart showing subprograms and data elements used by the driver program and each of the five testing subroutines is given in Figure 1.

4. HOW TO OMIT PIVOTING

In order to increase the accuracy of computations, pivoting is used in the decomposition subroutines (DECOM1 and DECOM2) which are called by L2A and L2B. In certain applications, however, the user may wish to omit pivoting. Pivoting was omitted in the adaptation of L2A prepared for OMNITAB 80 [11] so that the user can control the order in which variables enter the regression equation and thus obtain the proper sequence of certain sums of squares used in an analysis of variance.

Pivoting can be omitted by making the following changes in DECOM1 and DECOM2. (DECOM1 and DECOM2 are not listed in the present paper, but complete listings of these subroutines are given in "Collected Algorithms from ACM" [6].)

DECOM1:

(1) Delete statements DC1 630 through DC1 990, inclusive.
(2) Replace DC1 1000 by:
```
130 IK = NS
```
(3) After DC1 1060 insert:
```
      IF (NS.EQ.1 .AND. DS.EQ.0.0) RETURN
```

DECOM2:

(1) Delete statements DC2 670 through DC2 1020, inclusive.
(2) Replace DC2 1030 by:
```
140 IK = NS
```
(3) After statement DC2 1160 insert:
```
      IF (NS.EQ.1 .AND. DS.EQ.0.0)RETURN
```

5. COMPUTING A PSEUDOINVERSE

One can compute the pseudoinverse of an $M \times N$ matrix A through the use of subroutine L2B. If one chooses MODE = 2, L = M, and sets B equal to the identity matrix of order M, the $N \times M$ pseudoinverse of A will be returned by L2B in array X. That is, the K-th coefficient vector will be the K-th column of the desired pseudoinverse.

Problem 9 of elements DATA5A and DATA5C is an example given by Albert [1 ; p. 63] in which A is the 3 by 4 matrix

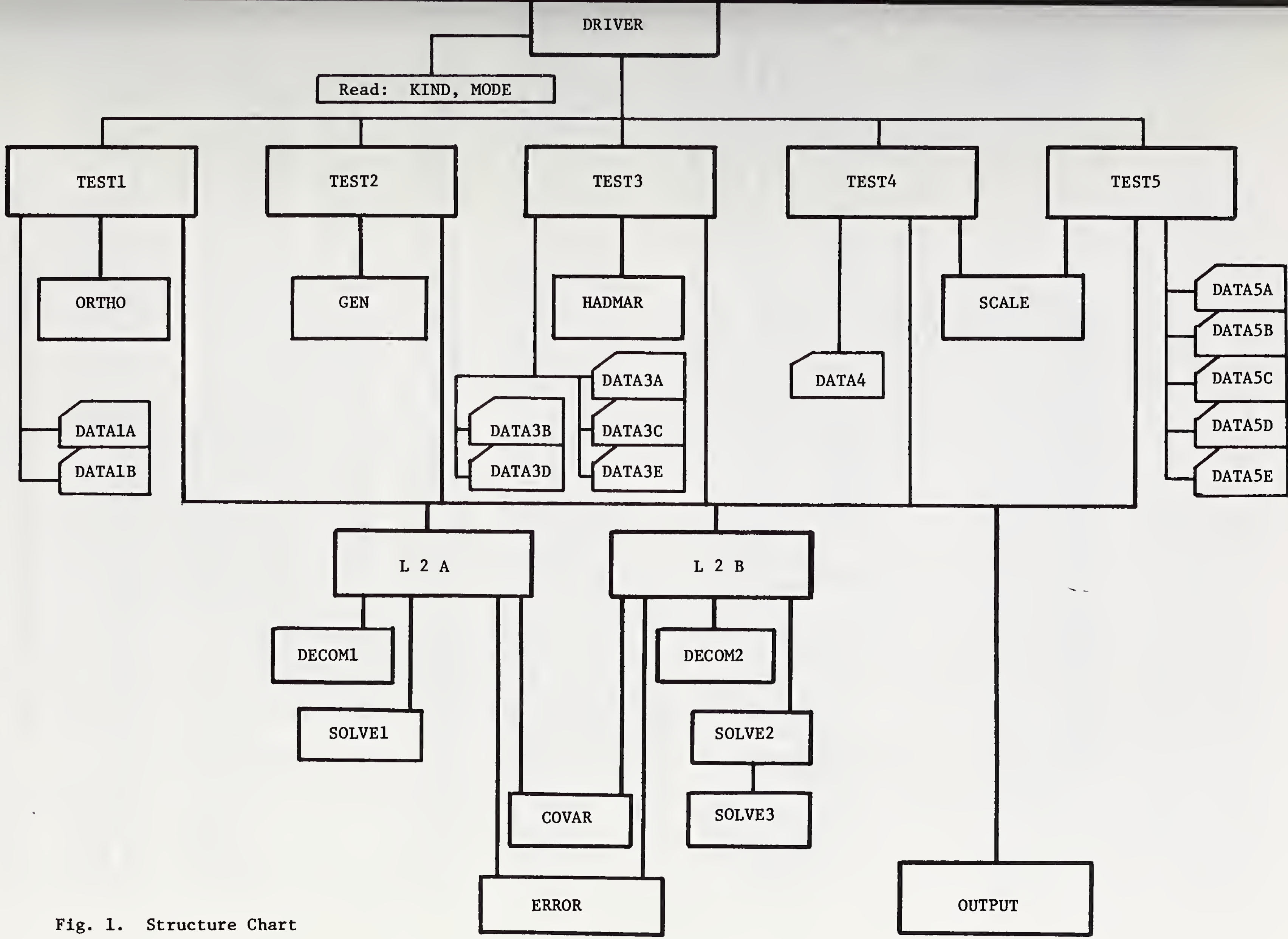

Fig. 1. Structure Chart

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} .$$

This matrix has rank 2; its pseudoinverse is

$$A^+ = \frac{1}{15} \begin{bmatrix} 3 & 0 & 3 \\ -1 & 5 & 4 \\ 4 & -5 & -1 \\ 3 & 0 & 3 \end{bmatrix} .$$

Appendix C gives computed results for this problem from both L2A (MODE 1) and L2B (MODE 2). Both algorithms find the matrix A to be of rank 2 and give the same sum of squared residuals. In L2A, columns 3 and 4 of A are ignored in computing the solutions since only columns 1 and 2 were found to be linearly independent. In L2B, however, all four columns of A are used in obtaining the solutions. Uniqueness is achieved by requiring that the solution vectors be of minimal norm.

6. MISCELLANEOUS REMARKS

In using the scaling option in connection with subroutine L2B, one should note that if the computed rank N1 is less than N, the solution vector of coefficients depends upon the choice of scaling. (Recall that when $N1 < N$, a unique solution is not obtained until we impose the condition that the solution vector be of minimal norm.).

It was mentioned previously that an adaptation of algorithm L2A was prepared for OMNITAB 80 [11]. Special features of this version which differ from L2A as described in [22] are:

(1) Pivoting is omitted in the decomposition subroutine.
(2) Sums of squares useful in analysis of variance are computed.
(3) Standard deviations of predicted values are computed.
(4) An estimate of the number of correctly computed digits in each coefficient is obtained by following the user-specified fit by a fit to predicted values and comparing how well the two sets of coefficients agree. See Wampler [19].

(5) If the desired fit using raw data fails to converge to a solution, or if the system of equations is found to be singular, the data are automatically scaled and another attempt is made to obtain a solution.

(6) All two-dimensional arrays were treated as one-dimensional vectors.

(7) The feature of allowing M1 linear constraints was omitted.

As was mentioned in Wampler [22], double precision versions of L2A and L2B (in which the iterative refinement of the solution is omitted) have been successfully implemented on a Univac 1108 and an IBM 360/65.

Additional results of test problems run on L2A are given in Wampler's "Test Procedures and Test Problems for Least Squares Algorithms" [24]. This paper considers two types of test problems: (1) Data due to Longley [14]; (2) Wampler's modification of a problem due to Läuchli [12]. Comparative results are given from running these problems on five different algorithms:

(1) Cholesky decomposition.
(2) Givens transformations.
(3) Modified Gram-Schmidt orthogonalization.
(4) Householder transformations.
(5) Singular value decomposition.

Wampler [24] also discusses and classifies some of the useful least squares test problems which have appeared in the literature during the past twenty years.

Table I

Time Required to Solve Hastings-type Problems on the Univac 1108

Values of parameters				Time required (seconds)			
M	N	M1	Subroutine	DECOMP	SOLVE	COVAR	TOTAL
10	2	0	L2A	.004	.008	.003	.349
10	3	0	L2A	.006	.010	.004	.251
10	4	0	L2A	.007	.011	.005	.268
10	6	0	L2A	.009	.015	.006	.300
10	8	0	L2A	.013	.018	.009	.337
10	8	0	L2B	.014	.037	.008	.495
10	8	4	L2A	.010	.018	.008	.333
10	8	4	L2B	.011	.036	.008	.376
20	2	0	L2A	.005	.012	.004	.317
20	5	0	L2A	.012	.019	.005	.365
20	10	0	L2A	.028	.033	.012	.472
20	15	0	L2A	.056	.048	.024	.635
20	15	0	L2B	.057	.104	.025	.734
20	15	5	L2A	.039	.047	.024	.617
20	15	5	L2B	.042	.102	.025	.709
30	5	0	L2A	.014	.024	.005	.410
30	10	0	L2A	.038	.043	.011	.558
30	15	0	L2A	.078	.064	.024	.752
30	20	0	L2A	.131	.087	.047	1.002
30	20	0	L2B	.134	.188	.047	1.160
30	20	10	L2A	.086	.087	.047	.942
30	20	10	L2B	.091	.189	.048	1.109
40	5	0	L2A	.016	.030	.005	.470
40	10	0	L2A	.048	.055	.012	.642
40	20	0	L2A	.169	.107	.047	1.141
40	30	0	L2A	.368	.165	.134	1.895
40	30	0	L2B	.372	.371	.134	2.209
40	30	10	L2A	.246	.161	.133	1.752
40	30	10	L2B	.263	.362	.134	2.068
50	10	0	L2A	.058	.065	.011	.723
50	20	0	L2A	.207	.128	.047	1.284
50	30	0	L2A	.451	.195	.134	2.136
50	40	0	L2A	.791	.272	.292	3.293
50	40	0	L2B	.796	.646	.308	3.850
50	40	10	L2A	.551	.262	.292	2.984
50	40	10	L2B	.604	.615	.300	3.590
100	10	0	L2A	.108	.120	.011	1.144
100	20	0	L2A	.396	.231	.048	2.007
100	40	0	L2A	1.524	.471	.292	4.950

Table I

Time Required to Solve Hastings-type Problems on the Univac 1108

Values of parameters				Time required (seconds)			
M	N	M1	Subroutine	DECOMP	SOLVE	COVAR	TOTAL
100	60	0	L2A	3.389	.746	.915	9.792
100	80	0	L2A	6.156	1.034	2.083	16.912
100	80	0	L2B	6.122	2.404	2.125	18.152
100	80	10	L2A	4.840	1.001	2.115	15.093
100	80	10	L2B	4.986	2.288	2.112	16.619
150	10	0	L2A	.159	.174	.011	1.562
150	20	0	L2A	.585	.334	.047	2.730
150	40	0	L2A	2.258	.672	.291	6.634
150	60	0	L2A	5.058	1.232	1.044	13.276
150	80	0	L2A	10.255	2.295	2.298	24.691
150	80	0	L2B	8.976	4.426	2.084	24.563
150	80	10	L2A	7.458	1.448	2.190	20.842
150	80	10	L2B	7.473	3.074	2.086	21.580
200	10	0	L2A	.213	.234	.012	2.009
200	20	0	L2A	.840	.496	.054	3.665
200	40	0	L2A	3.227	.918	.295	8.789
200	60	0	L2A	7.382	1.625	1.049	17.684
200	80	0	L2A	12.879	1.917	2.102	29.626
200	80	0	L2B	11.901	3.963	2.091	29.278
200	80	10	L2A	11.848	1.823	2.299	28.129
200	80	10	L2B	12.419	4.221	2.157	30.669

Table II

Solutions, Computed Rank, and Residual Norm for the Test Case M = 7, N = 6, using L2B

	Noise Level = 0.0		Relative Noise Level = 10^{-4}	
	TOL = 0.0	TOL = 0.5	TOL = 0.0	TOL = 0.5
x_1	0.1458	0.1458	-4543.7	0.1470
x_2	-0.0208	-0.0208	4544.4	-0.0217
x_3	0.0833	0.0833	5063.6	0.0846
x_4	-0.0833	-0.0833	-5063.8	-0.0841
x_5	-0.1042	-0.1042	-520.8	-0.1030
x_6	-0.2708	-0.2708	520.5	-0.2717
Computed rank	4	4	6	4
Norm of residuals	530.33	530.33	54.25	530.34

Table III. Hadamard-type Problems Run on Univac 1108
Machine precision approximately 8 digits. ETA = 2^{-26}.

		Max. error in 8th sig. digit of coefficients				Residual norm	Max. error in 8th sig. digit of residuals		
k 2^3	No. of iterations	α=0	α=.01	α=1	α=100	α=0	α=.01	α=1	α=100
2^{12}	2 to 4	0	0	0	0	$.95\times10^{-17}$ to $.14\times10^{-10}$	20 to 31	0 or 1	0 or 1
2^{16}	2 to 5	0	0	0	1	$.94\times10^{-14}$ to $.44\times10^{-9}$	10 to 48	0 or 1	0 or 1
2^{20}	4 to 5	0	0	0	0	$.29\times10^{-11}$ to $.75\times10^{-8}$	26 to 48	0 or 1	0 or 1

Table IV. Hadamard-type Problems Run on Univac 1108
Machine precision approximately 8 digits. ETA = 2^{-26}. $2^{3^k} = 2^{24}$.

		No. of iterations				Max. error in 8th sig. digit of coefficients				Residual norm	Max. error in 8th sig. digit of residuals		
M	N	α=0	α=.01	α=1	α=100	α=0	α=.01	α=1	α=100	α=0	α=.01	α=1	α=100
2	2	8				0				$.21\times10^{-8}$			
4	2	8	8	7	5	0	0	0	5	$.30\times10^{-8}$	29	0	1
4	4	8				0*				$.56\times10^{-8}$			
8	2	8	8	7	7	0	0	0	1	$.43\times10^{-8}$	30	1	1
8	4	8	8	8	6	0*	0*	0*	3	$.79\times10^{-8}$	53	0	0
8	8	9				0*				$.21\times10^{-7}$			
16	2	8	8	7	5	0	0	0	1	$.61\times10^{-8}$	28	1	1
16	4	8	8	8	6	0*	0	0	2	$.11\times10^{-7}$	30	0	0
16	8	9	9	8	7	0*	0*	0	0	$.29\times10^{-7}$	82	1	1
16	16	9				0				$.11\times10^{-7}$			

*Computed coefficients were 0.99999999.

Table V. Hadamard-type Problems Run on IBM 360/65
Machine precision approximately 6 digits. ETA = 16^{-5}.

		Max. error in 6th sig. digit of coefficients				Residual norm	Max. error in 6th sig. digit of residuals		
2^{3k}	No. of iterations	α=0	α=.01	α=1	α=100	α=0	α=.01	α=1	α=100
2^{12}	2 to 4	0	0	0	0	$.26\times10^{-11}$ to $.33\times10^{-7}$	5	0	0
2^{16}	2 to 5	0	0	0	7	$.11\times10^{-9}$ to $.72\times10^{-7}$	4	0	0
2^{20}	Computed rank was N-1 in all cases.								

ACKNOWLEDGMENTS

The author wishes to thank Prentice-Hall, Inc., Englewood Cliffs, New Jersey, for permission to reproduce Function GEN and portions of Program PROG2 (pages 311, 281-282), from Charles L. Lawson and Richard J. Hanson, SOLVING LEAST SQUARES PROBLEMS, copyright 1974.

Thanks are also due to Richard J. Hanson for permission to use the problems constructed from Hadamard matrices, and to Joseph M. Cameron and James W. Longley for helpful collaborations.

REFERENCES

1. Albert, Arthur. Regression and the Moore-Penrose Pseudoinverse. Academic Press, New York, 1972.

2. Bennett, Carl A., and Franklin, Norman L. Statistical Analysis in Chemistry and the Chemical Industry. Wiley, New York, 1954.

3. Björck, Åke. Comment on the iterative refinement of least-squares solutions. Journal of the American Statistical Association 73 (1978), 161-166.

4. Björck, Åke, and Golub, Gene. ALGOL Programming Contribution No. 22: Iterative refinement of linear least square solutions by Householder transformation. BIT 7 (1967), 322-337.

5. Businger, Peter, and Golub, Gene H. Linear least squares by Householder transformations. Numerische Mathematik 7 (1965), 269-276.

6. Collected Algorithms from ACM. Association for Computing Machinery, 1133 Avenue of the Americas, New York, N.Y. 10036.

7. Hall, Marshall, Jr. Combinatorial Theory. Blaisdell, Waltham, Massachusetts, 1967.

8. Hanson, Richard J. Personal communication.

9. Hastings, W. Keith. Test data for statistical algorithms: Least squares and ANOVA. Journal of the American Statistical Association 67 (1972), 874-879.

10. Hogben, David, Peavy, Sally T., and Varner, Ruth N. OMNITAB II User's Reference Manual. National Bureau of Standards Technical Note 552, U. S. Government Printing Office, Washington, D.C., 1971.

11. Hogben, David, and Peavy, Sally T. OMNITAB 80: A High-Quality Integrated Programming Language and Statistical Software Computing System. NBS Handbook, in preparation.

12. Läuchli, Peter. Jordan-Elimination und Ausgleichung nach kleinsten Quadraten. Numerische Mathematik 3 (1961), 226-240.

13. Lawson, Charles L., and Hanson, Richard J. Solving Least Squares Problems. Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

14. Longley, James W. An appraisal of least squares programs for the electronic computer from the point of view of the user. Journal of the American Statistical Association 62 (1967), 819-841.

15. Moses, L. E., and Oakford, R. V. Tables of Random Permutations. Stanford University Press, Stanford, California, 1963.

16. The RAND Corporation. A Million Random Digits with 100,000 Normal Deviates. The Free Press, Glencoe, Illinois, 1955.

17. Sande, G. Program execution profiles. Proceedings of the Computer Science and Statistics Eighth Annual Symposium on the Interface, ed. by James W. Frane, University of California, Los Angeles (1975), 325-326.

18. Seber, G. A. F. Linear Regression Analysis. Wiley, New York, 1977.

19. Wampler, Roy H. An evaluation of linear least squares computer programs. Journal of Research of the National Bureau of Standards 73B (1969), 59-90.

20. Wampler, Roy H. A report on the accuracy of some widely used least squares computer programs. Journal of the American Statistical Association 65 (1970), 549-565.

21. Wampler, Roy H. Some recent developments in linear least-squares computations. Proceedings of the Computer Science and Statistics Sixth Annual Symposium on the Interface, ed. by Michael E. Tarter, University of California, Berkeley (1972), 94-110.

22. Wampler, Roy H. Solutions to weighted least squares problems by modified Gram-Schmidt with iterative refinement. ACM Transactions on Mathematical Software, Vol. 5, No. 4 (Dec. 1979), 457-465.

23. Wampler, Roy H. Algorithm 544: L2A and L2B, Weighted least squares solutions by modified Gram-Schmidt with iterative refinement. ACM Transactions on Mathematical Software, Vol. 5, No. 4 (Dec. 1979), 494-499.

24. Wampler, Roy H. Test procedures and test problems for least squares algorithms. Journal of Econometrics 12 (1980), 3-22.

APPENDIX A. LISTING OF PROGRAM

```
C THIS MAIN PROGRAM CALLS FIVE TESTING SUBROUTINES FOR SOLVING LINEAR
C LEAST SQUARES TEST PROBLEMS.  THE TESTING SUBROUTINES, IN TURN, CALL
C SUBROUTINES WHICH USE A MODIFIED GRAM-SCHMIDT ALGORITHM WITH
C ITERATIVE REFINEMENT IN ORDER TO OBTAIN THE SOLUTIONS.
C
C VERSION OF FEBRUARY 25, 1980.
C
C WRITTEN BY ROY H. WAMPLER, STATISTICAL ENGINEERING DIVISION,
C NATIONAL BUREAU OF STANDARDS, WASHINGTON, D. C. 20234.
C
C REFERENCES --
C    1. ROY H. WAMPLER, SOLUTIONS TO WEIGHTED LEAST SQUARES PROBLEMS BY
C       MODIFIED GRAM-SCHMIDT WITH ITERATIVE REFINEMENT,
C       ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE, VOL. 5, 1979,
C       PP. 457-465.
C    2. ROY H. WAMPLER, ALGORITHM 544, L2A AND L2B, WEIGHTED LEAST
C       SQUARES SOLUTIONS BY MODIFIED GRAM-SCHMIDT WITH ITERATIVE
C       REFINEMENT (F4), ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE,
C       VOL. 5, 1979, PP. 494-499.
C    3. ROY H. WAMPLER, PROBLEMS USED IN TESTING THE EFFICIENCY AND
C       ACCURACY OF THE MODIFIED GRAM-SCHMIDT LEAST SQUARES ALGORITHM,
C       N.B.S. TECHNICAL NOTE, NATIONAL BUREAU OF STANDARDS,
C       WASHINGTON, D. C. 20234.  (THIS PUBLICATION)
C
C THIS MAIN PROGRAM READS IN TWO INPUT PARAMETERS -- KIND AND MODE.
C
C DEPENDING ON THE VALUE OF KIND (1, 2, 3, 4 OR 5), ONE OF FIVE TESTING
C SUBROUTINES IS CALLED TO EITHER GENERATE DATA OR READ IN DATA FOR
C LEAST SQUARES PROBLEMS.  THESE FIVE SUBROUTINES ARE NAMED TEST1,
C TEST2, TEST3, TEST4 AND TEST5.  THE FIRST FOUR GENERATE SPECIFIC TYPES
C OF TEST PROBLEMS, WHEREAS TEST5 READS IN DATA FOR GENERAL LEAST
C SQUARES PROBLEMS.
C
C EACH OF THESE FIVE SUBROUTINES WILL CALL EITHER SUBROUTINE L2A OR L2B
C TO COMPUTE THE LEAST SQUARES SOLUTIONS.  SUBSEQUENTLY, EACH OF THE
C FIVE TESTING SUBROUTINES CALLS SUBROUTINE OUTPUT FOR PRINTING
C RESULTS.
C
C THE VALUE OF THE PARAMETER MODE (1 OR 2) DETERMINES WHETHER SUBROUTINE
C L2A OR L2B SHALL BE CALLED.  THESE TWO SUBROUTINES WILL FURNISH THE
C SAME SOLUTIONS WHENEVER THE COMPUTED RANK OF THE SYSTEM OF M EQUATIONS
C IN N UNKNOWNS EQUALS N AND M IS GREATER THAN OR EQUAL TO N.  IN CASES
C WHERE THE COMPUTED RANK N1 IS LESS THAN N, THE USER SPECIFIES THE TYPE
C OF SOLUTION TO BE COMPUTED ACCORDING TO WHETHER MODE = 1 OR MODE = 2.
C MATRIX A IS THE GIVEN MATRIX OF A SYSTEM OF M LINEAR EQUATIONS IN N
C UNKNOWNS.  MATRIX W IS A GIVEN DIAGONAL MATRIX OF WEIGHTS WITH ALL
C DIAGONAL ELEMENTS NONNEGATIVE.  LET H = (SQRT(W))*A.
C
C MODE = 1 INDICATES THAT IF N1.LT.N THE ORIGINAL MATRIX (M BY N)
C          IS TO BE REPLACED BY A SMALLER MATRIX (M BY N1) WHOSE
C          COLUMNS ARE LINEARLY INDEPENDENT, AND A SOLUTION IS TO BE
C          SOUGHT FOR THE SMALLER SYSTEM OF EQUATIONS.  THUS N - N1
C          COLUMNS OF THE ORIGINAL MATRIX H ARE DELETED, AND COEFFICIENTS
C          CORRESPONDING TO THESE N - N1 DELETED COLUMNS WILL BE SET
C          EQUAL TO ZERO.  (MODE 1 CORRESPONDS TO SUBROUTINE L2A.)
C MODE = 2 INDICATES THAT A SOLUTION IS SOUGHT FOR A LEAST SQUARES
C          PROBLEM HAVING N ELEMENTS IN THE SOLUTION VECTOR.  IN ORDER
```

```
C         TO OBTAIN A UNIQUE SOLUTION, THE CONDITION THAT THE
C         SOLUTION VECTOR BE OF MINIMAL EUCLIDEAN NORM IS IMPOSED.
C         (MODE 2 CORRESPONDS TO SUBROUTINE L2B.)
C
C THE ARGUMENT LISTS IN THE CALLING SEQUENCES FOR THE FIVE SUBROUTINES
C CALLED BY THIS MAIN PROGRAM ARE ALL ALIKE.  OF THE 21 PARAMETERS
C APPEARING IN THESE ARGUMENT LISTS, ONE IS THE INPUT PARAMETER MODE
C ALREADY DESCRIBED, 13 ARE DIMENSIONED ARRAYS, AND 7 ARE DIMENSIONING
C PARAMETERS.  THE DIMENSIONING PARAMETERS ARE USED TO TRANSMIT
C DIMENSIONS OF ARRAYS TO SUBROUTINES.  FIVE ADDITIONAL INPUT QUANTITIES
C (M, N, M1, L AND TOL) NEEDED IN THE CALLING SEQUENCES FOR L2A AND L2B
C WILL BE GIVEN VALUES IN THE TESTING SUBROUTINES.  SEE SUBROUTINE TESTS
C FOR COMMENTS ON THESE PARAMETERS.
C
C DESCRIPTION OF DIMENSIONED ARRAYS --
C A       TWO-DIMENSIONAL ARRAY OF SIZE (MM,NNP1).  ON ENTRY TO L2A OR
C         L2B, THIS ARRAY CONTAINS THE GIVEN MATRIX OF A SYSTEM OF M
C         LINEAR EQUATIONS IN N UNKNOWNS, WHERE THE FIRST M1 EQUATIONS
C         ARE TO BE SATISFIED EXACTLY.
C B       TWO-DIMENSIONAL ARRAY OF SIZE (MM,LL).  ON ENTRY TO L2A OR L2B,
C         B CONTAINS THE L GIVEN RIGHT-HAND SIDES (VECTORS OF
C         OBSERVATIONS).
C W       ONE-DIMENSIONAL ARRAY OF LENGTH (MM).  ON ENTRY TO L2A OR L2B,
C         W CONTAINS THE DIAGONAL ELEMENTS OF A GIVEN DIAGONAL MATRIX OF
C         WEIGHTS, ALL NONNEGATIVE.  (THE FIRST M1 ELEMENTS OF W ARE SET
C         EQUAL TO 1.0 BY THE PROGRAM WHEN M1 IS GREATER THAN ZERO.)  ON
C         EXIT FROM L2A OR L2B, THE ORIGINAL ELEMENTS OF W HAVE BEEN
C         REPLACED BY THEIR SQUARE ROOTS.
C X       TWO-DIMENSIONAL ARRAY OF SIZE (NN,LL).  ON EXIT FROM L2A OR
C         L2B, X CONTAINS THE SOLUTION VECTORS.
C RES     TWO-DIMENSIONAL ARRAY OF SIZE (MM,LL).  ON EXIT FROM L2A OR
C         L2B, RES CONTAINS THE RESIDUAL VECTORS.
C R       TWO-DIMENSIONAL ARRAY OF SIZE (NN,NN).  ON EXIT FROM L2A, R
C         CONTAINS THE LOWER TRIANGULAR PORTION OF THE SYMMETRIC
C         UNSCALED COVARIANCE MATRIX.
C Q       TWO-DIMENSIONAL ARRAY OF SIZE (MM,NN).  USED INTERNALLY ONLY
C         IN SUBROUTINE L2A AND RELATED SUBROUTINES.
C QR      TWO-DIMENSIONAL ARRAY OF SIZE (MMPNN,NN).  ON EXIT FROM L2B,
C         QR CONTAINS THE LOWER TRIANGULAR PORTION OF THE SYMMETRIC
C         UNSCALED COVARIANCE MATRIX.
C IPIVOT  ONE-DIMENSIONAL ARRAY OF LENGTH (NN).  ON EXIT FROM L2A OR
C         L2B, THIS ARRAY RECORDS THE ORDER IN WHICH THE COLUMNS OF H
C         WERE SELECTED BY THE PIVOTING SCHEME IN THE COURSE OF THE
C         ORTHOGONAL DECOMPOSITION.
C C       ONE-DIMENSIONAL ARRAY OF LENGTH (MNL).  IF L2A IS TO BE CALLED,
C         MNL MUST EQUAL AT LEAST 4*(M+N)+2*L.  IF L2B IS TO BE CALLED,
C         MNL MUST EQUAL AT LEAST 6*(M+N)+2*L.  THIS VECTOR IS USED (1)
C         FOR INTERNAL WORK SPACE AND (2) FOR RETURNING INFORMATION ON
C         THE BEHAVIOR OF THE ITERATIVE REFINEMENT PROCEDURE.
C         (A) NUMIT IS THE NUMBER OF ITERATIONS CARRIED OUT DURING THE
C             ITERATIVE REFINEMENT IN ATTEMPTING TO OBTAIN A SOLUTION
C             FOR THE K-TH RIGHT-HAND SIDE.  ON EXIT FROM L2A OR L2B,
C             C(K) = +NUMIT IF THE SOLUTION CONVERGED, AND
C             C(K) = -NUMIT IF THE SOLUTION FAILED TO CONVERGE.
C         (B) DIGITX GIVES AN ESTIMATE OF THE NUMBER OF CORRECT DIGITS
C             IN THE INITIAL SOLUTION OF THE COEFFICIENTS FOR THE K-TH
C             RIGHT-HAND SIDE.  ON EXIT FROM L2A OR L2B, C(K+L) = DIGITX.
C FAIL    ONE-DIMENSIONAL ARRAY OF LENGTH (LL).  USED IN SUBROUTINE
```

```
C         OUTPUT IN CONNECTION WITH INFORMATION ON THE SUCCESS OR
C         FAILURE OF THE ITERATIVE REFINEMENT PROCEDURE.
C SDX     ONE-DIMENSIONAL ARRAY OF LENGTH (NN).  USED FOR STANDARD
C         DEVIATIONS OF COEFFICIENTS IN SUBROUTINE OUTPUT.
C SF      ONE-DIMENSIONAL ARRAY OF LENGTH (NNPLL).  USED FOR SCALE
C         FACTORS IN SUBROUTINES TEST4, TEST5 AND SCALE.
C
C DIMENSIONING PARAMETERS --
C MM      MUST SATISFY MM.GE.M.
C NN      MUST SATISFY NN.GE.N.
C LL      MUST SATISFY LL.GE.L.
C MMPNN   MUST SATISFY MMPNN.GE.M+N.
C NNP1    MUST SATISFY NNP1.GE.N+1.
C NNPLL   MUST SATISFY NNPLL.GE.N+L.
C MNL     MUST SATISFY MNL.GE.K*(M+N)+2*L, WHERE K EQUALS 4 IF
C         SUBROUTINE L2A IS TO BE CALLED, AND K EQUALS 6 IF
C         SUBROUTINE L2B IS TO BE CALLED.
C
C DIMENSIONED ARRAYS ARRANGED  BY TYPE --
C     INTEGER IPIVOT(NN)
C     REAL A(MM,NNP1), B(MM,LL), C(MNL)
C     REAL Q(MM,NN), R(NN,NN), QR(MMPNN,NN)
C     REAL RES(MM,LL), SDX(NN), SF(NNPLL), W(MM), X(NN,LL)
C     LOGICAL FAIL(LL)
C
      INTEGER IPIVOT(20)
      REAL A(30,21), B(30,4), C(308)
      REAL Q(30,20), R(20,20), QR(50,20)
      REAL RES(30,4), SDX(20), SF(24), W(30), X(20,4)
      LOGICAL FAIL(4)
C
C IN THE FOLLOWING DATA STATEMENT, NR IS THE CARD READER DEVICE
C AND NW IS THE PRINTER DEVICE NUMBER.
C
      DATA NR,NW / 5, 6 /
C
      DATA MM    / 30 /
      DATA NN    / 20 /
      DATA LL    /  4 /
      DATA MMPNN / 50 /
      DATA NNP1  / 21 /
      DATA NNPLL / 24 /
      DATA MNL   /308 /
C
      READ (NR,80000) KIND, MODE
      IF (KIND.LT.1 .OR. KIND.GT.5) WRITE (NW,90000)
      IF (KIND.LT.1 .OR. KIND.GT.5) STOP
      IF (MODE.NE.2) MODE = 1
      WRITE (NW,90010) KIND,KIND,MODE
      GO TO (10,20,30,40,50), KIND
C
C     TEST1 -- PROBLEMS DUE TO HASTINGS.
C
   10 CALL TEST1 (A, B, C, FAIL, IPIVOT, LL, MM, MMPNN, MNL, MODE,
     * NN, NNPLL, NNP1, Q, QR, R, RES, SDX, SF, W, X)
      GO TO 60
C
C     TEST2 -- PROBLEMS DUE TO LAWSON AND HANSON.
```

```
C
   20 CALL TEST2 (A, B, C, FAIL, IPIVOT, LL, MM, MMPNN, MNL, MODE,
     * NN, NNPLL, NNP1, Q, QR, R, RES, SDX, SF, W, X)
      GO TO 60
C
C     TEST3 -- PROBLEMS DUE TO HANSON.
C
   30 CALL TEST3 (A, B, C, FAIL, IPIVOT, LL, MM, MMPNN, MNL, MODE,
     * NN, NNPLL, NNP1, Q, QR, R, RES, SDX, SF, W, X)
      GO TO 60
C
C     TEST4 -- PROBLEMS DUE TO BJORCK AND WAMPLER.
C
   40 CALL TEST4 (A, B, C, FAIL, IPIVOT, LL, MM, MMPNN, MNL, MODE,
     * NN, NNPLL, NNP1, Q, QR, R, RES, SDX, SF, W, X)
      GO TO 60
C
C     TEST5 - GENERAL PROBLEMS.
C
   50 CALL TEST5 (A, B, C, FAIL, IPIVOT, LL, MM, MMPNN, MNL, MODE,
     * NN, NNPLL, NNP1, Q, QR, R, RES, SDX, SF, W, X)
C
   60 STOP
C
80000 FORMAT (2I5)
C
90000 FORMAT (60H0***  ERROR -- KIND MUST EQUAL 1, 2, 3, 4 OR 5.  HERE K
     *IND =,I5)
90010 FORMAT (1H1,6HKIND =,I2/10H CALL TEST,I1,34H TO OBTAIN LEAST SQUAR
     *ES SOLUTIONS/7H MODE =,I2)
C
      END


      SUBROUTINE TEST1 (A, B, C, FAIL, IPIVOT, LL, MM, MMPNN, MNL, MODE,
     * NN, NNPLL, NNP1, Q, QR, R, RES, SDX, SF, W, X)
C
C SUBROUTINE TEST1 CALLS SUBROUTINE ORTHO TO GENERATE INPUT DATA
C (MATRICES A AND B) FOR TESTING LEAST SQUARES PROGRAMS.  THE ALGORITHM
C IN ORTHO IS BASED ON EXAMPLE 2 IN --
C    TEST DATA FOR STATISTICAL ALGORITHMS -- LEAST SQUARES AND ANOVA,
C    BY W. K. HASTINGS, JOURNAL OF THE AMERICAN STATISTICAL
C    ASSOCIATION, VOL. 67, 1972, PP. 874-879.
C
C HASTINGS DID NOT GIVE FORMULAS FOR THE COEFFICIENTS IN THE CASE OF
C PROBLEMS HAVING LINEAR CONSTRAINTS, BUT SOME PROBLEMS OF THIS TYPE
C ARE INCLUDED IN ORDER TO OBTAIN DATA ON TIMING.
C
C THIS SUBROUTINE CALLS A MACHINE-DEPENDENT EXTERNAL ROUTINE (CLOCKT)
C WHICH PRINTS THE NUMBER OF SECONDS ELAPSED SINCE EXECUTION OF THE
C RUN BEGAN.  TIME IS PRINTED IN SECONDS, TO THE CLOSEST THOUSANDTH
C OF A SECOND.
C
C THE SEQUENCE OF INPUT CARDS (OR CARD IMAGES) FOR THIS SUBROUTINE IS --
C 1. CARD IN (I5) FORMAT GIVING THE VALUE OF NPROB, THE NUMBER OF
C    PROBLEMS TO BE SOLVED.
C 2. FOR EACH OF THE NPROB PROBLEMS, A CARD IN (3I5) FORMAT GIVING THE
C    VALUES OF M, N AND M1.
```

```
C     M    TOTAL NUMBER OF EQUATIONS.  M MUST BE EVEN.
C     N    NUMBER OF UNKNOWN COEFFICIENTS.
C     M1   NUMBER OF LINEAR CONSTRAINTS (0.LE.M1.LE.M AND M1.LE.N).
C
      INTEGER IPIVOT(NN)
      REAL A(MM,NNP1), B(MM,LL), C(MNL)
      REAL Q(MM,NN), R(NN,NN), QR(MMPNN,NN)
      REAL RES(MM,LL), SDX(NN), SF(NNPLL), W(MM), X(NN,LL)
      REAL AC, TOL
      LOGICAL FAIL(LL)
C
C IN THE FOLLOWING DATA STATEMENT, NR IS THE CARD READER DEVICE
C AND NW IS THE PRINTER DEVICE NUMBER.
C
      DATA NR,NW /5,6/
C
      CALL CLOCKT(AC)
C
      READ (NR,80000) NPROB
      IPROB = 0
   10 IPROB = IPROB + 1
      WRITE (NW,90000) IPROB
C
      CALL CLOCKT(AC)
C
      READ (NR,80010) M,N,M1
      L = 1
      TOL = 0.0
      IWGHT = 1
C
C IWGHT = 1 DENOTES THAT ALL WEIGHTS ARE EQUAL TO 1.0.
C
      WRITE (NW,90010)
      WRITE (NW,90020) M,N,M1,L,IWGHT,MODE,TOL
C
C SET PARAMETERS WHICH ALLOCATE VECTOR C (TO BE USED AS WORK AREA IN
C SUBROUTINE ORTHO).
C
      K1 = 1
      K2 = K1 + N + 1
      K3 = K2 + N
      K4 = K3 + N
      K5 = K4 + N
      K6 = K5 + N
C
C C(K6) IS A VECTOR OF LENGTH N.
C
      CALL CLOCKT(AC)
C
      IF (MODE.EQ.1) CALL ORTHO (M, N, MM, MM, Q, B, C(K1), C(K2),
     * C(K3), C(K4), C(K5), C(K6), A)
C
      IF (MODE.EQ.2) CALL ORTHO (M, N, MM, MMPNN, QR, B, C(K1), C(K2),
     * C(K3), C(K4), C(K5), C(K6), A)
C
      CALL CLOCKT(AC)
C
      NNN = M/2
```

```
C
C M MUST BE EVEN.
C
      IF (2*NNN .NE. M) GO TO 90
      GO TO (20,50), MODE
   20 DO 40 I=1,M
        DO 30 J=1,N
          A(I,J) = Q(I,J)
   30   CONTINUE
        W(I) = 1.0
   40 CONTINUE
C
      CALL CLOCKT(AC)
C
C MATRIX A, MATRIX B AND VECTOR OF WEIGHTS COULD BE PRINTED HERE.
C
      CALL L2A (M, N, M1, L, A, B, W, TOL, MM, NN,
     * N1, IPIVOT, X, RES, R, Q, C, IFAULT)
C
      CALL CLOCKT(AC)
C
      GO TO 80
   50 DO 70 I=1,M
        DO 60 J=1,N
          A(I,J) = QR(I,J)
   60   CONTINUE
        W(I) = 1.0
   70 CONTINUE
C
      CALL CLOCKT(AC)
C
C MATRIX A, MATRIX B AND VECTOR OF WEIGHTS COULD BE PRINTED HERE.
C
      CALL L2B (M, N, M1, L, A, B, W, TOL, MM, NN, MMPNN,
     * N1, IPIVOT, X, RES, QR, C, IFAULT)
C
      CALL CLOCKT(AC)
C
   80 CALL OUTPUT (B, C, IFAULT, IPIVOT, L, M, MODE, M1, N, N1, QR, R,
     * RES, W, X, LL, MM, MMPNN, MNL, NN, FAIL, SDX)
C
   90 CALL CLOCKT(AC)
C
      IF (IPROB.LT.NPROB) GO TO 10
      RETURN
C
80000 FORMAT (I5)
80010 FORMAT (3I5)
C
90000 FORMAT (1H1,115(1H*),4X,7HPROBLEM,I4)
90010 FORMAT (1H0,3X,1HM,4X,1HN,3X,2HM1,4X,1HL,5X,5HIWGHT,6X,4HMODE,7X,
     * 3HTOL)
90020 FORMAT (4I5,2I10,G15.8)
C
      END


      SUBROUTINE TEST2 (A, B, C, FAIL, IPIVOT, LL, MM, MMPNN, MNL, MODE,
```

```
     * NN, NNPLL, NNP1, Q, QR, R, RES, SDX, SF, W, X)
C
C PORTIONS OF THIS SUBROUTINE ARE TAKEN FROM PROGRAM PROG2,
C CHARLES L. LAWSON AND RICHARD J. HANSON,
C SOLVING LEAST SQUARES PROBLEMS, COPYRIGHT 1974, PAGES 281-282.
C REPRINTED BY PERMISSION OF PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, N.J.
C
C SUBROUTINE TEST2 CALLS FUNCTION GEN TO GENERATE LEAST SQUARES DATA
C MATRICES A AND B.
C
C FOUR SETS OF PROBLEMS ARE GENERATED WITH 18 PROBLEMS IN EACH SET.
C VALUES OF M AND N ARE SELECTED FROM THE SET (1, 2, 3, 6, 7, 8).
C THE TOLERANCE PARAMETER, TOL, IS SET EITHER TO 0.0 OR 0.5.
C
C SET 1.  DATA MATRICES A AND B ARE INTEGERS BETWEEN -500 AND +500.
C         SOME OF THE A-MATRICES ARE RANK-DEFICIENT.
C         TOL = 0.0.
C SET 2.  DATA OF SET 1 MODIFIED BY ADDING NOISE TO THE INTEGERS.
C         TOL = 0.0.
C SET 3.  DATA AS IN SET 1.
C         TOL = 0.5.
C SET 4.  DATA AS IN SET 2.
C         TOL = 0.5.
C
C THIS SUBROUTINE REQUIRES NO INPUT FROM THE USER.
C
      INTEGER IPIVOT(NN)
      REAL A(MM,NNP1), B(MM,LL), C(MNL)
      REAL Q(MM,NN), R(NN,NN), QR(MMPNN,NN)
      REAL RES(MM,LL), SDX(NN), SF(NNPLL), W(MM), X(NN,LL)
      REAL ANOISE, ANORM, DUMMY, GEN, TAU, TOL
      LOGICAL FAIL(LL)
C
C IN THE FOLLOWING DATA STATEMENT, NW IS THE PRINTER DEVICE NUMBER.
C
      DATA NW / 6 /
C
      IPROB = 0
      DO 90 ITOL=1,2
        DO 80 NOISE=1,2
          ANORM = 500.0
          ANOISE = 0.0
          TAU = 0.5
          IF (NOISE.EQ.1) GO TO 10
          ANOISE = 1.E-4
          TAU = ANORM * ANOISE * 10.0
   10     CONTINUE
C
C INITIALIZE THE DATA GENERATION FUNCTION.
C
          DUMMY = GEN(-1.0)
          WRITE (NW,90000) ANOISE,ANORM,TAU
          DO 70 MN1=1,6,5
            MN2 = MN1 + 2
            DO 60 M=MN1,MN2
              DO 50 N=MN1,MN2
C
C GENERATE DATA FOR MATRICES A AND B AND SET WEIGHTS EQUAL TO UNITY.
```

```
C
                DO 30 I=1,M
                  DO 20 J=1,N
                    A(I,J) = GEN(ANOISE)
   20             CONTINUE
                  B(I,1) = GEN(ANOISE)
                  W(I) = 1.0
   30           CONTINUE
                M1 = 0
                L = 1
                IWGHT = 1
                TOL = 0.0
                IF (ITOL.EQ.2) TOL = TAU
                IPROB = IPROB + 1
                WRITE (NW,90010) IPROB
                WRITE (NW,90020)
                WRITE (NW,90030) M,N,M1,L,IWGHT,MODE,TOL
C
C PRINT A, B AND W.
C
                WRITE (NW,90040)
                DO 40 I=1,M
                  WRITE (NW,90050) (A(I,J),J=1,N),(B(I,K),K=1,L),W(I)
   40           CONTINUE
C
                IF (MODE.EQ.1) CALL L2A (M, N, M1, L, A, B, W, TOL, MM,
     *           NN, N1, IPIVOT, X, RES, R, Q, C, IFAULT)
C
                IF (MODE.EQ.2) CALL L2B (M, N, M1, L, A, B, W, TOL, MM,
     *           NN, MMPNN, N1, IPIVOT, X, RES, QR, C, IFAULT)
C
                CALL OUTPUT (B, C, IFAULT, IPIVOT, L, M, MODE, M1, N,
     *          N1, QR, R, RES, W, X, LL, MM, MMPNN, MNL, NN, FAIL, SDX)
C
   50         CONTINUE
   60       CONTINUE
   70     CONTINUE
   80   CONTINUE
   90 CONTINUE
      RETURN
C
90000 FORMAT (1H0,54HTHE RELATIVE NOISE LEVEL OF THE GENERATED DATA WILL
     * BE,G15.8/33H0THE MATRIX NORM IS APPROXIMATELY,G15.8/43H0THE ABSOL
     *UTE PSEUDORANK TOLERANCE, TAU, IS,G15.8)
90010 FORMAT (1H1,115(1H*),4X,7HPROBLEM,I4)
90020 FORMAT (1H0,3X,1HM,4X,1HN,3X,2HM1,4X,1HL,5X,5HIWGHT,6X,4HMODE,7X,
     * 3HTOL)
90030 FORMAT (4I5,2I10,G15.8)
90040 FORMAT (41H0MATRIX A, MATRIX B AND VECTOR OF WEIGHTS/)
90050 FORMAT (1X,8G15.8)
C
      END


      SUBROUTINE TEST3 (A, B, C, FAIL, IPIVOT, LL, MM, MMPNN, MNL, MODE,
     * NN, NNPLL, NNP1, Q, QR, R, RES, SDX, SF, W, X)
C
C SUBROUTINE TEST3 GENERATES LEAST SQUARES TEST PROBLEMS CONSTRUCTED
```

```
C FROM HADAMARD MATRICES.  THE METHOD OF CONSTRUCTING THESE PROBLEMS IS
C DUE TO RICHARD J. HANSON.  THIS SUBROUTINE CALLS SUBROUTINE HADMAR
C WHICH FURNISHES THE HADAMARD MATRICES.
C
C THE SEQUENCE OF INPUT CARDS (OR CARD IMAGES) FOR THIS SUBROUTINE IS --
C 1. CARD IN (I5) FORMAT GIVING THE VALUE OF NK3, A PARAMETER WHICH
C    SPECIFIES HOW MANY SETS OF PROBLEMS ARE TO BE CONSTRUCTED AND
C    SOLVED.  EACH SET WILL CONSIST OF 10 PROBLEMS USING THE FOLLOWING
C    COMBINATIONS OF M AND N.
C    (M, N) = (2,2), (4,2), (4,4), (8,2), (8,4), (8,8), (16,2), (16,4),
C             (16,8), (16,16).
C 2. FOR EACH OF THE NK3 SETS OF PROBLEMS, A CARD IN (I5) FORMAT GIVING
C    A VALUE OF K3.  K3, A NONNEGATIVE INTEGER, IS TO BE USED AS A POWER
C    OF 2 WHICH WILL DETERMINE THE CONDITION NUMBER OF MATRIX A.
C    (CONDITION NUMBER OF A EQUALS 2**K3.)
C 3. FOR EACH OF THE NK3 SETS OF PROBLEMS, 6 CARDS IN (14I3) FORMAT
C    GIVING VALUES OF IRHO(I), I=1,...,MMN, WHERE MMN = M - N (FOR
C    PROBLEMS HAVING M.GT.N).  THESE ARE TO BE FURNISHED IN THE
C    FOLLOWING SEQUENCE --
C       CARD 1   M = 4    N = 2   MMN = 2
C       CARD 2   M = 8    N = 2   MMN = 6
C       CARD 3   M = 8    N = 4   MMN = 4
C       CARD 4   M = 16   N = 2   MMN = 14
C       CARD 5   M = 16   N = 4   MMN = 12
C       CARD 6   M = 16   N = 8   MMN = 8
C    THE ELEMENTS OF IRHO ARE TO BE INTEGERS WHICH ARE UNIFORMLY
C    DISTRIBUTED ON THE CLOSED INTERVAL (0, BNORM), WHERE
C    BNORM = N*SQRT(M).
C
      INTEGER IPIVOT(NN)
      INTEGER IRHO(16)
      REAL A(MM,NNP1), B(MM,LL), C(MNL)
      REAL Q(MM,NN), R(NN,NN), QR(MMPNN,NN)
      REAL RES(MM,LL), SDX(NN), SF(NNPLL), W(MM), X(NN,LL)
      REAL BHAT(16), RHO(16), S(16,16), TOL, U(16,16), V(16,16)
      REAL Z(16,16)
      LOGICAL FAIL(LL)
C
C IN THE FOLLOWING DATA STATEMENT, NR IS THE CARD READER DEVICE
C AND NW IS THE PRINTER DEVICE NUMBER.
C
      DATA NR, NW / 5, 6 /
C
C COMPUTE AND PRINT HADAMARD MATRICES OF ORDER 2, 4, 8 AND 16.
C
      N = 2
      WRITE (NW,90000)
C
   10 CALL HADMAR (N, V)
C
      WRITE (NW,90010) N
      DO 20 I=1,N
        WRITE (NW,90020) (V(I,J),J=1,N)
   20 CONTINUE
      N = 2*N
      IF (N.LE.16) GO TO 10
C
      READ (NR,80000) NK3
```

```
      WRITE (NW,90030) NK3
      IPROB = 0
      KA = 0
   30 KA = KA + 1
      READ (NR,80000) K3
      KK = 2**K3
C
C K3 CONTROLS VALUE OF S(N,N).
C S(N,N) MAY BE SET EQUAL TO THE FOLLOWING POWERS OF 2 --
C     2**12 =     4096
C     2**16 =    65536
C     2**20 =  1048576
C     2**24 = 16777216
C     2**25 = 33554432
C
      M = 2
   40 N = 2
   50 DO 70 I=1,M
        W(I) = 1.0
        DO 60 J=1,N
          S(I,J) = 0.0
   60   CONTINUE
   70 CONTINUE
      S(1,1) = 1.0
      S(N,N) = FLOAT(KK)
      IF (N.EQ.2) GO TO 90
C
C IF N.GT.2 SET S(I,I) EQUAL TO NUMBERS BETWEEN 1.0 AND S(N,N), FOR
C 1.LT.I.LT.N.  THE S(I,I) ARE EITHER EQUALLY SPACED OR ALMOST EQUALLY
C SPACED.  HERE THE S(I,I) ARE SET ARBITRARILY.  THESE VALUES COULD BE
C CHOSEN RANDOMLY.
C
      NM1 = N - 1
      IFACT = (KK-1)/(N-1)
      DO 80 I=2,NM1
        S(I,I) = 1.0 + FLOAT(IFACT*(I-1))
   80 CONTINUE
C
C CALL SUBROUTINE HADMAR TO OBTAIN V (N BY N).
C
   90 CALL HADMAR (N, V)
C
C MULTIPLY MATRIX S (M BY N) BY V (N BY N) TO OBTAIN MATRIX Z (M BY N).
C
      DO 120 I=1,M
        DO 110 J=1,N
          Z(I,J) = 0.0
          DO 100 K=1,N
            Z(I,J) = Z(I,J) + S(I,K)*V(K,J)
  100     CONTINUE
  110   CONTINUE
  120 CONTINUE
C
C CALL SUBROUTINE HADMAR TO OBTAIN U (M BY M).
C
      CALL HADMAR (M, U)
C
C MULTIPLY MATRIX U (M BY M) BY Z (M BY N) TO OBTAIN MATRIX A (M BY N).
```

```
C
      DO 150 I=1,M
        DO 140 J=1,N
          A(I,J) = 0.0
          DO 130 K=1,M
            A(I,J) = A(I,J) + U(I,K)*Z(K,J)
  130     CONTINUE
  140   CONTINUE
  150 CONTINUE
      DO 160 I=1,M
        BHAT(I) = FLOAT(N)
        RHO(I) = 0.0
  160 CONTINUE
C
C READ IRHO EXCEPT WHEN N = M.
C THE DATA USED HERE AS IRHO WERE TAKEN FROM --
C (1) A MILLION RANDOM DIGITS BY THE RAND CORPORATION, THE FREE PRESS,
C     GLENCOE, ILLINOIS, 1955.
C (2) TABLES OF RANDOM PERMUTATIONS, L. E. MOSES AND R. V. OAKFORD,
C     STANFORD UNIVERSITY PRESS, STANFORD, CALIFORNIA, 1963.
C AS AN ALTERNATIVE TO READING IN DATA FOR IRHO, ONE COULD CALL A
C SUBROUTINE TO GENERATE RANDOM INTEGERS.
C
      IF (N.EQ.M) GO TO 180
      MMN = M - N
C
      READ (NR,80010) (IRHO(I),I=1,MMN)
C
      DO 170 I=1,MMN
        IPN = I + N
        RHO(IPN) = FLOAT(IRHO(I))
  170 CONTINUE
C
C MULTIPLY MATRIX U (M BY M) BY RHO (M BY 1) TO OBTAIN R (M BY 1).
C
  180 DO 210 I=1,M
        DO 200 J=1,1
          R(I,J) = 0.0
          DO 190 K=1,M
            R(I,J) = R(I,J) + U(I,K)*RHO(K)
  190     CONTINUE
  200   CONTINUE
  210 CONTINUE
C
C FOR ALPHA = 0.0, SET B = BHAT.
C
      DO 220 I=1,M
        B(I,1) = BHAT(I)
  220 CONTINUE
      IF (N.EQ.M) GO TO 240
C
C COMPUTE B(I,2), B(I,3), B(I,4) FOR ALPHA = 0.01, 1.0, 100.0 WHEN
C M.GT.N.
C
      DO 230 I=1,M
        B(I,2) = BHAT(I) + 0.01 * R(I,1)
        B(I,3) = BHAT(I) + 1.00 * R(I,1)
        B(I,4) = BHAT(I) + 100. * R(I,1)
```

```
  230 CONTINUE
  240 IPROB = IPROB + 1
      WRITE (NW,90040) IPROB
      IF (N.EQ.M) L = 1
      IF (N.LT.M) L = 4
      M1 = 0
      IWGHT  = 1
      TOL = 0.0
      WRITE (NW,90050)
      WRITE (NW,90060) M,N,M1,L,IWGHT,MODE,TOL
      IF (N.EQ.M) GO TO 250
      WRITE (NW,90070) (IRHO(I),I=1,MMN)
  250 WRITE (NW,90080) K3,KK
      DO 260 I=1,N
        RHO(I) = S(I,I)
  260 CONTINUE
      WRITE (NW,90090)
      WRITE (NW,90100) (RHO(I),I=1,N)
C
C PRINT A, B AND W.
C
      WRITE (NW,90110)
      DO 270 I=1,M
        WRITE (NW,90100) (A(I,J),J=1,N),(B(I,K),K=1,L),W(I)
  270 CONTINUE
C
      IF (MODE.EQ.1) CALL L2A (M, N, M1, L, A, B, W, TOL, MM, NN,
     * N1, IPIVOT, X, RES, R, Q, C, IFAULT)
C
      IF (MODE.EQ.2) CALL L2B (M, N, M1, L, A, B, W, TOL, MM, NN, MMPNN,
     * N1, IPIVOT, X, RES, QR, C, IFAULT)
C
      CALL OUTPUT (B, C, IFAULT, IPIVOT, L, M, MODE, M1, N, N1, QR, R,
     * RES, W, X, LL, MM, MMPNN, MNL, NN, FAIL, SDX)
C
      IF (N.EQ.M) GO TO 280
      N = 2*N
      GO TO 50
  280 M = 2*M
      IF (M.LE.16) GO TO 40
      IF (KA.LT.NK3) GO TO 30
C
      RETURN
C
80000 FORMAT (I5)
80010 FORMAT (14I3)
C
90000 FORMAT (1H1)
90010 FORMAT (25H0HADAMARD MATRIX OF ORDER,I3/)
90020 FORMAT (16F6.0)
90030 FORMAT (1H0//23H NUMBER OF K3 VALUES IS,I5)
90040 FORMAT (1H1,115(1H*),4X,7HPROBLEM,I4)
90050 FORMAT (1H0,3X,1HM,4X,1HN,3X,2HM1,4X,1HL,5X,5HIWGHT,6X,4HMODE,7X,
     * 3HTOL)
90060 FORMAT (4I5,2I10,G15.8)
90070 FORMAT (5H0IRHO,24I5)
90080 FORMAT (13H0K3 AND 2**K3,I5,I10)
90090 FORMAT (23H0DIAGONAL ELEMENTS OF S/)
```

```
90100 FORMAT (1X,8G15.8)
90110 FORMAT (41HOMATRIX A, MATRIX B AND VECTOR OF WEIGHTS/)
C
      END


      SUBROUTINE TEST4 (A, B, C, FAIL, IPIVOT, LL, MM, MMPNN, MNL, MODE,
     * NN, NNPLL, NNP1, Q, QR, R, RES, SDX, SF, W, X)
C
C SUBROUTINE TEST4 GENERATES DATA FOR WEIGHTED POLYNOMIALS.  THE METHOD
C IS BASED ON --
C    1. COMMENT ON THE ITERATIVE REFINEMENT OF LEAST-SQUARES SOLUTIONS,
C       BY AKE BJORCK, JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION,
C       VOL. 73, 1978, PP. 161-166.
C    2. A REPORT ON THE ACCURACY OF SOME WIDELY USED LEAST SQUARES
C       COMPUTER PROGRAMS, BY ROY H. WAMPLER, JOURNAL OF THE AMERICAN
C       STATISTICAL ASSOCIATION, VOL. 65, 1970, PP. 549-565.
C
C THE SEQUENCE OF INPUT CARDS (OR CARD IMAGES) FOR THIS SUBROUTINE IS --
C 1. CARD IN (I5) FORMAT GIVING VALUE OF THE SCALING PARAMETER ISCALE.
C    ISCALE = 0 MEANS THAT DATA ARE NOT TO BE SCALED.
C    ISCALE = 1 MEANS DATA MATRIX A IS TO BE SCALED, BUT MATRIX B
C       IS NOT TO BE SCALED.  SCALE FACTORS WILL BE AUTOMATICALLY
C       COMPUTED FROM THE DATA.  THE J-TH SCALE FACTOR, SF(J), IS
C       BASED ON THE EUCLIDEAN NORM OF THE J-TH COLUMN OF A.
C    ISCALE = 2 MEANS DATA MATRICES A AND B ARE TO BE SCALED.
C       MATRIX A IS SCALED IN THE SAME MANNER AS WHEN ISCALE = 1,
C       AND MATRIX B IS SCALED IN A SIMILAR MANNER.
C    ISCALE = 3 MEANS DATA MATRICES A AND B ARE TO BE SCALED.  THE
C       SCALE FACTORS ARE FURNISHED BY THE USER.  THEY ARE READ IN
C       AS ITEM 4, DESCRIBED BELOW, ONLY WHEN ISCALE = 3.
C 2. CARD IN (2I5) FORMAT GIVING VALUES OF MIN AND MAX.
C    MIN AND MAX WILL BE USED IN DETERMINING WEIGHTS TO BE APPLIED TO
C    THE FIRST, MIDDLE AND LAST OBSERVATIIONS.  THESE OBSERVATIONS ARE
C    GIVEN WEIGHT W = 2**(IP-1) WHERE IP = MIN,MIN+1,MIN+2,....,MAX.
C 3. CARDS IN (F3.0,F9.0,F7.0) FORMAT GIVING VALUES FOR AA(I,2),
C    Y2(I) AND DEL(I), I=1,2,....,21.  FROM THESE QUANTITIES AND
C    FROM THE WEIGHTS W, THE MATRICES A AND B ARE CONSTRUCTED.
C 4. CARD(S) GIVING SCALE FACTORS IN (8F10.0) FORMAT, TO BE READ ONLY
C    IF ISCALE = 3.
C
      INTEGER IPIVOT(NN)
      REAL A(MM,NNP1), B(MM,LL), C(MNL)
      REAL Q(MM,NN), R(NN,NN), QR(MMPNN,NN)
      REAL RES(MM,LL), SDX(NN), SF(NNPLL), W(MM), X(NN,LL)
      REAL AA(21,6), DEL(21), T, TOL, U, Y2(21)
      LOGICAL FAIL(LL)
C
C IN THE FOLLOWING DATA STATEMENT, NR IS THE CARD READER DEVICE
C AND NW IS THE PRINTER DEVICE NUMBER.
C
      DATA NR, NW / 5, 6 /
      DATA IONE, ITWO, ITHREE / 1, 2, 3 /
C
      READ (NR,80000) ISCALE
      READ (NR,80010) MIN,MAX
      IF (MODE.NE.2) MODE = 1
      IPROB = 0
```

```
      M = 21
      N = 6
      M1 = 0
      L = 1
      IWGHT = 2
      TOL = 0.0
      NM1 = N - 1
      WRITE (NW,90000)
      WRITE (NW,90010) M,N,M1,L,IWGHT,MODE,TOL,MIN,MAX
      WRITE (NW,90020) ISCALE
      DO 20 I=1,M
        READ (NR,80020) AA(I,2),Y2(I),DEL(I)
        W(I) = 1.0
        AA(I,1) = 1.0
        DO 10 J=2,NM1
          JP1 = J + 1
          AA(I,JP1) = AA(I,J) * AA(I,2)
   10   CONTINUE
   20 CONTINUE
C
      NPL = N + L
      IF (ISCALE.NE.3) GO TO 40
      READ (NR,80030) (SF(J),J=1,NPL)
      DO 30 J=1,NPL
        IF (SF(J).EQ.0.0) SF(J) = 1.0
   30 CONTINUE
C
   40 DO 80 IP=MIN,MAX
        DO 60 I=1,M
          IF (I.EQ.1 .OR. I.EQ.11 .OR. I.EQ.21) W(I) = 2.**(IP-1)
          DO 50 J=1,N
            A(I,J) = AA(I,J) * W(I)
   50     CONTINUE
          B(I,1) = Y2(I) * W(I)
          T = (2.**(IP-1)) * (1.0/W(I))
          T = T * DEL(I)
          B(I,1) = B(I,1) + T
          W(I) = 1.0
   60   CONTINUE
        IPROB = IPROB + 1
C
C PRINT A, B, W, IP AND PROBLEM NUMBER.
C
        WRITE (NW,90030) IP,IPROB
        DO 70 I=1,M
          WRITE (NW,90040) (A(I,J),J=1,N),(B(I,K),K=1,L),W(I)
   70   CONTINUE
C
        IF (ISCALE.GT.0) CALL SCALE (ISCALE, ITWO, M, N, L, W, MM, NN,
     *   NN, A, B, R, RES, SF, U, X, IFAULT)
C
        IF (ISCALE.GT.0) WRITE (NW,90050)
        IF (ISCALE.GT.0) WRITE (NW,90040) (SF(J),J=1,NPL)
C
        IF (MODE.EQ.1) CALL L2A (M, N, M1, L, A, B, W, TOL, MM, NN,
     *   N1, IPIVOT, X, RES, R, Q, C, IFAULT)
C
        IF (MODE.EQ.2) CALL L2B (M, N, M1, L, A, B, W, TOL, MM, NN,
```

```
     *   MMPNN, N1, IPIVOT, X, RES, QR, C, IFAULT)
C
        IF (ISCALE.GT.0 .AND. MODE.EQ.1) CALL SCALE (ISCALE, ITHREE,
     *   M, N, L, W, MM, NN, NN, A, B, R, RES, SF, U, X, IFAULT)
C
        IF (ISCALE.GT.0 .AND. MODE.EQ.2) CALL SCALE (ISCALE, ITHREE,
     *   M, N, L, W, MM, MMPNN, NN, A, B, QR, RES, SF, U, X, IFAULT)
C
        CALL OUTPUT (B, C, IFAULT, IPIVOT, L, M, MODE, M1, N, N1, QR, R,
     *   RES, W, X, LL, MM, MMPNN, MNL, NN, FAIL, SDX)
C
   80 CONTINUE
      RETURN
C
80000 FORMAT (I5)
80010 FORMAT (2I5)
80020 FORMAT (F3.0,F9.0,F7.0)
80030 FORMAT (8F10.0)
C
90000 FORMAT (1H0,3X,1HM,4X,1HN,3X,2HM1,4X,1HL,5X,5HIWGHT,6X,4HMODE,7X,
     * 3HTOL,7X,3HMIN,2X,3HMAX)
90010 FORMAT (4I5,2I10,G15.8,2I5)
90020 FORMAT (9H0ISCALE =,I2)
90030 FORMAT (41H1MATRIX A, MATRIX B AND VECTOR OF WEIGHTS,5X,4HIP =,I4,
     * 30X,7HPROBLEM,I4/)
90040 FORMAT (1X,8G15.8)
90050 FORMAT (14H0SCALE FACTORS/)
C
      END


      SUBROUTINE TEST5 (A, B, C, FAIL, IPIVOT, LL, MM, MMPNN, MNL, MODE,
     * NN, NNPLL, NNP1, Q, QR, R, RES, SDX, SF, W, X)
C
C SUBROUTINE TEST5 READS AND PRINTS INPUT DATA FOR LEAST SQUARES
C PROBLEMS, CALLS SUBROUTINE SCALE TO SCALE THE DATA (IF SCALING IS
C REQUESTED), CALLS EITHER SUBROUTINE L2A OR L2B TO COMPUTE SOLUTIONS,
C AND CALLS SUBROUTINE OUTPUT TO PRINT COMPUTED RESULTS.
C
C THE USER INDICATES THAT DATA ARE TO BE SCALED BY GIVING A POSITIVE
C VALUE (1, 2 OR 3) TO THE PARAMETER ISCALE.  THE PURPOSE OF SCALING
C IS TO MITIGATE, IF POSSIBLE, ROUNDING ERROR PROBLEMS WHICH CAN OCCUR
C IN CONNECTION WITH SOLVING ILL-CONDITIONED SYSTEMS OF EQUATIONS.
C
C SEE MAIN PROGRAM FOR A DESCRIPTION OF THE 21 PARAMETERS APPEARING IN
C THE ARGUMENT LIST OF SUBROUTINE TEST5.
C
C THE SEQUENCE OF INPUT CARDS (OR CARD IMAGES) FOR THIS SUBROUTINE IS --
C 1. CARD IN (I5) FORMAT GIVING VALUE OF THE SCALING PARAMETER ISCALE.
C    THIS VALUE OF ISCALE PERTAINS TO THE COMPLETE SET OF PROBLEMS WHICH
C    FOLLOWS.  IT IS READ IN ONLY ONCE FOR ANY SET OF PROBLEMS.
C    ISCALE = 0 MEANS THAT DATA ARE NOT TO BE SCALED.
C    ISCALE = 1 MEANS DATA MATRIX A IS TO BE SCALED, BUT MATRIX B
C       IS NOT TO BE SCALED.  SCALE FACTORS WILL BE AUTOMATICALLY
C       COMPUTED FROM THE DATA.  THE J-TH SCALE FACTOR, SF(J), IS
C       BASED ON THE EUCLIDEAN NORM OF THE J-TH COLUMN OF A.
C    ISCALE = 2 MEANS DATA MATRICES A AND B ARE TO BE SCALED.
C       MATRIX A IS SCALED IN THE SAME MANNER AS WHEN ISCALE = 1,
```

```
C        AND MATRIX B IS SCALED IN A SIMILAR MANNER.
C     ISCALE = 3 MEANS DATA MATRICES A AND B ARE TO BE SCALED.  THE
C        SCALE FACTORS ARE FURNISHED BY THE USER.  THEY ARE READ IN
C        AS ITEM 6, DESCRIBED BELOW, ONLY WHEN ISCALE = 3.
C 2. PROBLEM HEADING CARD, IN (80A1) FORMAT.
C 3. PARAMETER CARD IN (6I5,5X,F10.0) FORMAT, GIVING VALUES OF THE
C    PARAMETERS M, N, M1, L, ITYPE, IWGHT, TOL.
C    M      TOTAL NUMBER OF EQUATIONS.
C    N      NUMBER OF UNKNOWN COEFFICIENTS.
C    M1     NUMBER OF LINEAR CONSTRAINTS (0.LE.M1.LE.M AND M1.LE.N).
C    L      NUMBER OF RIGHT-HAND SIDES (VECTORS OF OBSERVATIONS).
C    ITYPE PARAMETER WHICH SPECIFIES WHETHER OR NOT DATA FOR A
C          POLYNOMIAL TYPE FIT ARE TO BE READ IN.
C          ITYPE = 1 INDICATES POLYNOMIAL TYPE.
C          ITYPE = 2 INDICATES NON-POLYNOMIAL TYPE.
C    IWGHT PARAMETER WHICH SPECIFIES WHETHER OR NOT WEIGHTS ARE TO
C          BE READ IN.
C          IWGHT = 1 INDICATES WEIGHTS ARE NOT TO BE READ IN.  (THE
C                    PROGRAM SETS ALL WEIGHTS EQUAL TO 1.0.)
C          IWGHT = 2 INDICATES WEIGHTS ARE TO BE READ IN.
C    TOL   PARAMETER USED IN DETERMINING THE RANK OF MATRIX H,
C          WHERE H = (SQRT(W))*A.
C          NOTE --
C          (1) IF TOL EQUALS ZERO, THE TOLERANCE USED IN THE
C              DECOMPOSITION SUBROUTINE WILL BE BASED ON MACHINE
C              PRECISION.
C          (2) IF TOL IS GREATER THAN ZERO, THIS VALUE OF TOL WILL BE
C              USED IN SETTING AN ABSOLUTE TOLERANCE FOR COMPARISON
C              WITH DIAGONAL ELEMENTS OF THE TRIANGULAR MATRIX OBTAINED
C              IN THE DECOMPOSITION SUBROUTINE.  THE VALUE OF TOL CAN
C              BE BASED ON KNOWLEDGE CONCERNING THE ACCURACY OF THE
C              DATA.
C 4. CARD GIVING FORMAT OF THE DATA CARDS (CONTAINING A, B AND
C    POSSIBLY W) WHICH FOLLOW.  THIS FORMAT CARD IS IN (80A1)
C    FORMAT.
C 5. DATA CARDS FOR THE ARRAYS A, B AND POSSIBLY W.  THERE ARE FOUR
C    POSSIBLE CONFIGURATIONS FOR THE DATA, DEPENDING ON THE VALUES
C    OF ITYPE AND IWGHT.  (FOR POLYNOMIAL FITS, THE FIRST POWER OF A
C    IS READ IN AND HIGHER POWERS ARE COMPUTED BY THE PROGRAM WHEN
C    N.GT.2.)  THE FOUR CONFIGURATIONS ARE ILLUSTRATED BELOW BY
C    SHOWING WHAT THE CARD (OR CARDS) FOR THE I-TH ROW OF DATA
C    CONTAINS.
C   A. ITYPE = 1, IWGHT = 1.
C      POLYNOMIAL TYPE FIT.  EQUAL WEIGHTS, NOT TO BE READ IN.
C     A(I,2)  B(I,1) B(I,2) ... B(I,L)
C   B. ITYPE = 1, IWGHT = 2.
C      POLYNOMIAL TYPE FIT.  UNEQUAL WEIGHTS, TO BE READ IN.
C     A(I,2)  B(I,1) B(I,2) ... B(I,L)  W(I)
C   C. ITYPE = 2, IWGHT = 1.
C      NON-POLYNOMIAL TYPE FIT.  EQUAL WEIGHTS, NOT TO BE READ IN.
C     A(I,1) A(I,2) ... A(I,N)  B(I,1) B(I,2) ... B(I,L)
C   D. ITYPE = 2, IWGHT = 2.
C      NON-POLYNOMIAL TYPE FIT.  UNEQUAL WEIGHTS, TO BE READ IN.
C     A(I,1) A(I,2) ... A(I,N)  B(I,1) B(I,2) ... B(I,L)  W(I)
C 6. *** OMIT THIS ITEM UNLESS ISCALE = 3. ***
C        WHEN ISCALE = 3, SCALE FACTORS ARE TO BE READ IN,
C        IN (8F10.0) FORMAT.  A TOTAL OF N + L SCALE FACTORS ARE
C        TO BE FURNISHED, THE FIRST N PERTAINING TO THE N COLUMNS OF
```

```
C          MATRIX A AND THE LAST L PERTAINING TO THE L COLUMNS OF B.
C          EACH SCALE FACTOR IS USED TO MULTIPLY THE CORRESPONDING
C          COLUMN OF A OR B BEFORE THE LEAST SQUARES FIT IS PEFORMED.
C 7. CARD IN (I5) FORMAT GIVING VALUE OF THE PARAMETER IFDONE.  IF
C    THE PROBLEM AT HAND IS TO BE FOLLOWED BY ANOTHER PROBLEM,
C    IFDONE = 1.  OTHERWISE, IFDONE EQUALS ANY INTEGER EXCEPT 1.
C    IF IFDONE = 1, GO TO ITEM 2 ABOVE FOR THE NEXT HEADING CARD.
C
      INTEGER IPIVOT(NN)
      INTEGER IFMT(80), IHEAD(80)
      REAL A(MM,NNP1), B(MM,LL), C(MNL)
      REAL Q(MM,NN), R(NN,NN), QR(MMPNN,NN)
      REAL RES(MM,LL), SDX(NN), SF(NNPLL), W(MM), X(NN,LL)
      REAL TOL, U
      LOGICAL FAIL(LL)
C
C IN THE FOLLOWING DATA STATEMENT, NR IS THE CARD READER DEVICE
C AND NW IS THE PRINTER DEVICE NUMBER.
C
      DATA NR, NW / 5, 6 /
C
      DATA IONE, ITWO, ITHREE / 1, 2, 3 /
C
      READ (NR,80000) ISCALE
      IF (ISCALE.LT.0 .OR. ISCALE.GT.3) ISCALE = 0
      WRITE (NW,90000) ISCALE
      IPROB = 0
   10 READ (NR,80010) IHEAD
      IPROB = IPROB + 1
      IFAULT = 0
      U = 0.0
      WRITE (NW,90010) IPROB
      WRITE (NW,90020) IHEAD
      READ (NR,80020) M,N,M1,L,ITYPE,IWGHT,TOL
      WRITE (NW,90030)
      WRITE (NW,90040) M,N,M1,L,ITYPE,IWGHT,MODE,ISCALE,TOL
      READ (NR,80010) IFMT
      WRITE (NW,90050) IFMT
      GO TO (20,100), ITYPE
C
C TYPE 1.  POLYNOMIAL FIT.
   20 GO TO (30,50), IWGHT
C       A.  EQUAL WEIGHTS, NOT TO BE READ IN.
   30 DO 40 I=1,M
        READ (NR,IFMT) A(I,2),(B(I,K),K=1,L)
        W(I) = 1.0
   40 CONTINUE
      GO TO 70
C       B.  UNEQUAL WEIGHTS, TO BE READ IN.
   50 DO 60 I=1,M
        READ (NR,IFMT) A(I,2),(B(I,K),K=1,L),W(I)
   60 CONTINUE
C
C CALL SUBROUTINE SCALE TO COMPUTE MEAN OF VECTOR A(I,2), IF DATA ARE
C TO BE SCALED.
C
   70 IF (ISCALE.GT.0 .AND.N.GT.1) CALL SCALE (ISCALE, IONE, M, N,
     * L, W, MM, NN, NN, A, B, R, RES, SF, U, X, IFAULT)
```

```
C
C IF DATA ARE SCALED, PRINT U.  IN SCALED POLYNOMIAL-TYPE PROBLEMS, U
C EQUALS THE MEAN OF THE INDEPENDENT VARIABLE (VECTOR A(I,2)).
C U IS SUBTRACTED FROM EACH ELEMENT OF A(I,2) BEFORE POWERS OF
C THIS VECTOR ARE GENERATED.
C
      IF (ISCALE.GT.0) WRITE (NW,90060) U
      DO 90 I=1,M
        A(I,1) = 1.0
        IF (N.EQ.1) GO TO 90
        A(I,2) = A(I,2) - U
        IF (N.EQ.2) GO TO 90
        DO 80 J=3,N
          A(I,J) = A(I,2)**(J-1)
   80   CONTINUE
   90 CONTINUE
      GO TO 150
C
C TYPE 2.  NON-POLYNOMIAL FIT.
  100 GO TO (110,130), IWGHT
C       C.  EQUAL WEIGHTS, NOT TO BE READ IN.
  110 DO 120 I=1,M
        READ (NR,IFMT) (A(I,J),J=1,N),(B(I,K),K=1,L)
        W(I) = 1.0
  120 CONTINUE
      GO TO 150
C       D.  UNEQUAL WEIGHTS, TO BE READ IN.
  130 DO 140 I=1,M
        READ (NR,IFMT) (A(I,J),J=1,N),(B(I,K),K=1,L),W(I)
  140 CONTINUE
  150 IF (M1.EQ.0 .OR. IWGHT.EQ.1) GO TO 170
C
C INSURE THAT WEIGHTS EQUAL 1.0 FOR THE FIRST M1 EQUATIONS WHEN M1
C IS GREATER THAN ZERO.
C
      DO 160 I=1,M1
        W(I) = 1.0
  160 CONTINUE
C
C PRINT A, B AND W.
C
  170 WRITE (NW,90070)
      KZ = 0
      DO 180 I=1,M
        IF (W(I).EQ.0.0) KZ = KZ + 1
        WRITE (NW,90080) (A(I,J),J=1,N),(B(I,K),K=1,L),W(I)
  180 CONTINUE
C
      NPL = N + L
      IF (ISCALE.NE.3) GO TO 200
      READ (NR,80030) (SF(J),J=1,NPL)
      DO 190 J=1,NPL
        IF (SF(J).EQ.0.0) SF(J) = 1.0
  190 CONTINUE
C
C CALL SUBROUTINE SCALE TO SCALE DATA MATRICES A AND B, IF REQUESTED.
C SCALE FACTORS WILL BE COMPUTED IF ISCALE EQUALS 1 OR 2.
C
```

```
  200 IF (ISCALE.GT.0) CALL SCALE (ISCALE, ITWO, M, N, L, W, MM,
     * NN, NN, A, B, R, RES, SF, U, X, IFAULT)
C
      IF (IFAULT.EQ.4) GO TO 240
C
C PRINT SCALE FACTORS (SF) IF DATA WERE SCALED.
C
      IF (ISCALE.GT.0) WRITE (NW,90090)
      IF (ISCALE.GT.0) WRITE (NW,90080) (SF(J),J=1,NPL)
      GO TO (210,220), MODE
C
C CALL SUBROUTINE L2A OR L2B TO COMPUTE LEAST SQUARES SOLUTIONS.
C
  210 CALL L2A (M, N, M1, L, A, B, W, TOL, MM, NN,
     * N1, IPIVOT, X, RES, R, Q, C, IFAULT)
C
      GO TO 230
C
  220 CALL L2B (M, N, M1, L, A, B, W, TOL, MM, NN, MMPNN,
     * N1, IPIVOT, X, RES, QR, C, IFAULT)
C
  230 IF (IFAULT.GE.1 .AND. IFAULT.LE.4) GO TO 240
C
C IF DATA WERE SCALED, CALL SUBROUTINE SCALE TO ADJUST COMPUTED RESULTS
C (COEFFICIENTS, RESIDUALS AND COVARIANCE MATRIX) FOR SCALING.
C
      IF (ISCALE.GT.0 .AND. MODE.EQ.1) CALL SCALE (ISCALE, ITHREE,
     * M, N, L, W, MM, NN, NN, A, B, R, RES, SF, U, X, IFAULT)
C
      IF (ISCALE.GT.0 .AND. MODE.EQ.2) CALL SCALE (ISCALE, ITHREE,
     * M, N, L, W, MM, MMPNN, NN, A, B, QR, RES, SF, U, X, IFAULT)
C
C CALL SUBROUTINE OUTPUT TO PRINT COMPUTED RESULTS.
C
  240 CALL OUTPUT (B, C, IFAULT, IPIVOT, L, M, MODE, M1, N, N1, QR, R,
     * RES, W, X, LL, MM, MMPNN, MNL, NN, FAIL, SDX)
C
      READ (NR,80000) IFDONE
      IF (IFDONE.EQ.1) GO TO 10
C
      RETURN
C
80000 FORMAT (I5)
80010 FORMAT (80A1)
80020 FORMAT (6I5,5X,F10.0)
80030 FORMAT (8F10.0)
C
90000 FORMAT (9H0ISCALE =,I5)
90010 FORMAT (1H1,113(1H*),4X,7HPROBLEM,I4)
90020 FORMAT (1H0,80A1)
90030 FORMAT (1H0,3X,1HM,4X,1HN,3X,2HM1,4X,1HL,5X,5HITYPE,5X,5HIWGHT,6X,
     * 4HMODE,4X,6HISCALE,7X,3HTOL)
90040 FORMAT (4I5,4I10,G15.8)
90050 FORMAT (8H0FORMAT ,80A1)
90060 FORMAT (28H0U = MEAN OF VECTOR A(I,2) =,G15.8/72H U IS SUBTRACTED
     *FROM EACH ELEMENT OF A(I,2) IN CONNECTION WITH SCALING.)
90070 FORMAT (41H0MATRIX A, MATRIX B AND VECTOR OF WEIGHTS/)
90080 FORMAT (1X,8G15.8)
```

```
90090 FORMAT (14HOSCALE FACTORS/)
C
      END


      SUBROUTINE OUTPUT (B, C, IFAULT, IPIVOT, L, M, MODE, M1, N, N1,
     * QR, R, RES, W, X, LL, MM, MMPNN, MNL, NN, FAIL, SDX)
C
C SUBROUTINE OUTPUT PRINTS THE SOLUTIONS OF LEAST SQUARES PROBLEMS
C WHICH WERE OBTAINED FROM EITHER SUBROUTINE L2A OR SUBROUTINE L2B.
C
C INPUT VARIABLES --
C B       TWO-DIMENSIONAL ARRAY OF SIZE (MM,LL).  SEE MAIN PROGRAM FOR
C         FURTHER DETAILS.
C C       VECTOR OF LENGTH MNL.  SEE MAIN PROGRAM.
C IFAULT  FAULT INDICATOR WHICH IS ZERO IF NO ERRORS WERE ENCOUNTERED
C         AND POSITIVE IF ERRORS WERE DETECTED OR IF EVIDENCE OF SEVERE
C         ILL-CONDITIONING WAS FOUND.  DIAGNOSTIC MESSAGES ARE PRINTED
C         FROM SUBROUTINE ERROR.  IF IFAULT IS SET EQUAL TO 1, 2, 3, 4,
C         5, 6 OR 7, EXECUTION IS TERMINATED.  EXECUTION CONTINUES WHEN
C         IFAULT IS SET EQUAL TO 8, 9 OR 10 PROVIDED THAT A SOLUTION
C         WAS OBTAINED FOR AT LEAST ONE RIGHT-HAND SIDE.  THE VALUE OF
C         IFAULT IS USED TO INDICATE THE FOLLOWING --
C         0 = NO ERRORS ENCOUNTERED.
C         1 = BAD INPUT PARAMETER (M, N OR L).
C         2 = BAD INPUT PARAMETER (M1).
C         3 = BAD DIMENSION.  EITHER M.GT.MM, N.GT.NN OR M+N.GT.MMPNN.
C         4 = AT LEAST ONE WEIGHT IS NEGATIVE.
C         5 = EITHER MATRIX H OR MATRIX OF CONSTRAINTS EQUALS ZERO.
C         6 = CONSTRAINTS ARE LINEARLY DEPENDENT.
C         7 = ALL SOLUTIONS FAILED TO CONVERGE.
C         8 = SOLUTION FAILED TO CONVERGE FOR AT LEAST ONE RIGHT-HAND
C             SIDE.
C         9 = LARGE NUMBER OF ITERATIONS REQUIRED FOR CONVERGENCE.
C        10 = ESTIMATED NUMBER OF DIGITS IN INITIAL SOLUTION OF
C             COEFFICIENTS IS SMALL.
C        11 = DIAGONAL ELEMENT OF COVARIANCE MATRIX WAS COMPUTED TO BE
C             NEGATIVE OWING TO ROUNDING ERROR.
C IPIVOT  VECTOR OF LENGTH NN.  SEE MAIN PROGRAM.
C L       NUMBER OF RIGHT-HAND SIDES (VECTORS OF OBSERVATIONS).
C M       TOTAL NUMBER OF EQUATIONS.
C MODE    PARAMETER WHICH INDICATES WHETHER SOLUTION WAS OBTAINED FROM
C         L2A OR L2B.  SEE MAIN PROGRAM FOR FURTHER DETAILS.
C M1      NUMBER OF LINEAR CONSTRAINTS.
C N       NUMBER OF UNKNOWN COEFFICIENTS.
C N1      COMPUTED RANK OF THE SYSTEM OF EQUATIONS.
C QR      TWO-DIMENSIONAL ARRAY OF SIZE (MMPNN,NN).  SEE MAIN PROGRAM.
C R       TWO-DIMENSIONAL ARRAY OF SIZE (NN,NN).  SEE MAIN PROGRAM.
C RES     TWO-DIMENSIONAL ARRAY OF SIZE (MM,LL).  SEE MAIN PROGRAM.
C W       VECTOR OF LENGTH MM.  SEE MAIN PROGRAM.
C X       TWO-DIMENSIONAL ARRAY OF SIZE (NN,LL).  SEE MAIN PROGRAM.
C LL      DIMENSIONING PARAMETER.  SEE MAIN PROGRAM.
C MM      DIMENSIONING PARAMETER.  SEE MAIN PROGRAM.
C MMPNN   DIMENSIONING PARAMETER.  SEE MAIN PROGRAM.
C MNL     DIMENSIONING PARAMETER.  SEE MAIN PROGRAM.
C NN      DIMENSIONING PARAMETER.  SEE MAIN PROGRAM.
C
C INTERNAL VARIABLES --
```

```
C FAIL    VECTOR OF LENGTH LL.  IF FAIL(K) = .TRUE. THE K-TH SOLUTION
C         FAILED TO CONVERGE.  IF FAIL(K) = .FALSE. THE K-TH SOLUTION
C         CONVERGED.
C SDX     VECTOR OF LENGTH NN.  STANDARD DEVIATIONS OF COEFFICIENTS (X).
C
      INTEGER IPIVOT (NN)
      REAL B(MM,LL), C(MNL), QR(MMPNN,NN), R(NN,NN), RES(MM,LL)
      REAL W(MM), X(NN,LL), SDX(NN)
      REAL SD, SNORM, SS, WSQ, Z
      DOUBLE PRECISION SUM
      LOGICAL FAIL(LL)
C
C IN THE FOLLOWING DATA STATEMENT, NW IS THE PRINTER DEVICE NUMBER.
C
      DATA NW / 6 /
C
      KZ = 0
      DO 10 I=1,M
        IF (W(I).EQ.0.0) KZ = KZ + 1
   10 CONTINUE
C
C PRINT COMPUTED RESULTS.
C
      WRITE (NW,90000)
      WRITE (NW,90010) MODE,IFAULT
      IF (IFAULT.GE.1 .AND. IFAULT.LE.4) GO TO 190
      WRITE (NW,90020) N1
      IF (N1.EQ.0) GO TO 190
      IF (N1.LT.M1) GO TO 190
      WRITE (NW,90030)
      WRITE (NW,90040) (IPIVOT(J),J=1,N1)
      IF (N1.EQ.N) GO TO 20
      N1P1 = N1 + 1
      WRITE (NW,90050)
      WRITE (NW,90040) (IPIVOT(J),J=N1P1,N)
   20 NDF = M - N1 - KZ
      WRITE (NW,90060) KZ,NDF
      WRITE (NW,90070)
      DO 40 K=1,L
        K2 = L + K
        IF (C(K).LT.0.0) GO TO 30
        FAIL(K) = .FALSE.
        WRITE (NW,90080) K,C(K),C(K2)
        GO TO 40
   30   FAIL(K) = .TRUE.
        C(K) = -C(K)
        WRITE (NW,90090) K,C(K),C(K2)
   40 CONTINUE
      IF (IFAULT.EQ.7) GO TO 190
      DO 150 K=1,L
C
C COMPUTE SUM OF SQUARED RESIDUALS, NORM OF RESIDUALS, RESIDUAL
C STANDARD DEVIATION, STANDARD DEVIATIONS OF COEFFICIENTS, AND
C PREDICTED VALUES.  PRINT THESE QUANTITIES, TOGETHER WITH
C COEFFICIENTS, OBSERVED VALUES, RESIDUALS AND WEIGHTS.  WEIGHTS ARE
C RESTORED TO THEIR ORIGINAL VALUES BEFORE BEING PRINTED.  (WITHIN
C SUBROUTINE L2A OR L2B, THE ORIGINAL WEIGHTS WERE REPLACED BY THEIR
C SQUARE ROOTS.)
```

```
C
      IF (FAIL(K)) GO TO 150
      WRITE (NW,90100) K
      SS = 0.0
      SUM = 0.0
      IF (M.EQ.M1) GO TO 110
      M1P1 = M1 + 1
      DO 50 I=M1P1,M
        SUM = SUM + DBLE(RES(I,K))**2 * DBLE(W(I)**2)
   50 CONTINUE
      SS = SUM
      IF (M.LE.N1+KZ) GO TO 110
      IF (MODE.EQ.2 .AND. N1.LT.N) GO TO 110
      SD = SQRT(SS/FLOAT(NDF))
      IF (MODE.EQ.2) GO TO 70
      DO 60 J=1,N
        IF (R(J,J).LT.0.0) R(J,J) = 0.0
        SDX(J) = SD*SQRT(R(J,J))
   60 CONTINUE
      GO TO 90
   70 DO 80 J=1,N
        IF (QR(J,J).LT.0.0) QR(J,J) = 0.0
        SDX(J) = SD*SQRT(QR(J,J))
   80 CONTINUE
   90 WRITE (NW,90110)
      DO 100 J=1,N
        WRITE (NW,90120) J,X(J,K),SDX(J)
  100 CONTINUE
      GO TO 130
  110 WRITE (NW,90130)
      DO 120 J=1,N
        WRITE (NW,90120) J,X(J,K)
  120 CONTINUE
  130 WRITE (NW,90140)
      DO 140 I=1,M
        Z = B(I,K) - RES(I,K)
        WSQ = W(I)*W(I)
        WRITE (NW,90120) I,B(I,K),Z,RES(I,K),WSQ
  140 CONTINUE
      SNORM = SQRT(SS)
      WRITE (NW,90150) SS,SNORM
      IF ((MODE.EQ.2 .AND. N1.LT.N) .OR. (M.EQ.N1+KZ)) GO TO 150
      WRITE (NW,90160) SD
  150 CONTINUE
C
C PRINT LOWER TRIANGULAR PORTION OF SYMMETRIC UNSCALED COVARIANCE
C MATRIX.
C
    IF (MODE.EQ.2 .AND. N1.LT.N) GO TO 190
    WRITE (NW,90170)
    IF (MODE.EQ.2) GO TO 170
    DO 160 I=1,N
      WRITE (NW,90180) (R(I,J),J=1,I)
  160 CONTINUE
    GO TO 190
  170 DO 180 I=1,N
      WRITE (NW,90180) (QR(I,J),J=1,I)
  180 CONTINUE
```

```
  190 RETURN
C
C FORMAT STATEMENTS.
C
90000 FORMAT (17HOCOMPUTED RESULTS)
90010 FORMAT (7HOMODE =,I4,5X,8HIFAULT =,I4)
90020 FORMAT (44HON1 = COMPUTED RANK OF SYSTEM OF EQUATIONS =,I4)
90030 FORMAT (87HOCOLUMNS OF H = (SQRT(W))*A WERE SELECTED BY THE PIVOTI
     *NG SCHEME IN THE FOLLOWING ORDER/)
90040 FORMAT (30I4)
90050 FORMAT (98HOTHE FOLLOWING COLUMNS OF H ARE LINEARLY DEPENDENT.  IF
     * MODE 1, THEY DID NOT ENTER THE REGRESSION./30H IF MODE 2, THEY EN
     *TERED LAST./)
90060 FORMAT (25HONUMBER OF ZERO WEIGHTS =,I3,5X,17HDEG. OF FREEDOM =,I3
     *)
90070 FORMAT (1H0,15X,9HREPORT ON,12X,9HNUMBER OF,4X,27HESTIMATED NUMBER
     * OF CORRECT/13H B-VECTOR NO.,3X,11HCONVERGENCE,10X,10HITERATIONS,3
     *X,26HDIGITS IN INITIAL SOLUTION/)
90080 FORMAT (I7,9X,9HCONVERGED,14X,F4.0,10X,G15.8)
90090 FORMAT (I7,9X,18HFAILED TO CONVERGE,5X,F4.0,10X,G15.8)
90100 FORMAT (26HOSOLUTION FOR B-VECTOR NO.,I3)
90110 FORMAT (1H0,27X,18HSTANDARD DEVIATION/5X,1HJ,4X,14HCOEFFICIENT(J),
     * 5X,17HOF COEFFICIENT(J)/)
90120 FORMAT (I6,G17.8,3G21.8)
90130 FORMAT (1H0,4X,1HJ,4X,14HCOEFFICIENT(J)/)
90140 FORMAT (1H0,4X,1HI,4X,11HOBSERVED(I),9X,12HPREDICTED(I),10X,11HRES
     *IDUAL(I),12X,9HWEIGHT(I)/)
90150 FORMAT (30HOSUM OF SQUARED RESIDUALS    =,G15.8/1X,17HNORM OF RESI
     *DUALS,11X,1H=,G15.8)
90160 FORMAT (30H RESIDUAL STANDARD DEVIATION =,G15.8)
90170 FORMAT (27HOUNSCALED COVARIANCE MATRIX/)
90180 FORMAT (1X,8G15.8)
C
      END


      SUBROUTINE ORTHO (N, IR, MM, MMM, X, Y, A, B, C, D, E, BETA, Z)
C
C SUBROUTINE ORTHO GENERATES DATA (X AND Y) FOR TESTING LEAST SQUARES
C PROGRAMS.  THE ALGORITHM IS BASED ON EXAMPLE 2 IN --
C    TEST DATA FOR STATISTICAL ALGORITHMS -- LEAST SQUARES AND ANOVA,
C    BY W. K. HASTINGS, JOURNAL OF THE AMERICAN STATISTICAL
C    ASSOCIATION, VOL. 67, 1972, PP. 874-879.
C
C FIRST A SET OF ORTHONORMAL VECTORS IS CONSTRUCTED IN THE ARRAY Z.
C THEN A SET OF X-VECTORS AND A Y-VECTOR ARE GENERATED FROM Z, AND
C THESE ARE USED AS INPUT FOR LEAST SQUARES PROBLEMS.  HASTINGS GAVE
C FORMULAS FOR THE LEAST SQUARES SOLUTIONS AND CERTAIN RELATED
C QUANTITIES.
C
C N AS USED HERE CORRESPONDS TO LOWER CASE N IN HASTINGS.
C NN AS USED HERE CORRESPONDS TO UPPER CASE N IN HASTINGS.
C X AND Y IN THIS SUBROUTINE CORRESPOND TO A AND B, RESPECTIVELY,
C IN SUBROUTINE TEST2 (THE CALLING PROGRAM).
C
C INPUT VARIABLES --
C N      LENGTH OF THE X-, Y- AND Z-VECTORS.  N MUST BE EVEN.
C IR     NUMBER OF X-VECTORS TO BE GENERATED.
```

```
C MM      DIMENSIONING PARAMETER SPECIFYING MAXIMUM NUMBER OF ROWS IN
C         ARRAYS Y AND Z.  N.LE.MM.
C MMM     DIMENSIONING PARAMETER SPECIFYING MAXIMUM NUMBER OF ROWS IN
C         ARRAY X.  N.LE.MMM.
C
C OUTPUT VARIABLES --
C X       TWO-DIMENSICNAL ARRAY OF SIZE (MMM,IR).
C Y       TWO-DIMENSIONAL ARRAY OF SIZE (MM,1).
C
C INTERNAL VARIABLES --
C A       VECTOR (WORK-SPACE) OF LENGTH N + 1.
C B       VECTOR (WORK-SPACE) OF LENGTH N.
C C       VECTOR (WORK-SPACE) OF LENGTH N.
C D       VECTOR (WORK-SPACE) OF LENGTH N.
C E       VECTOR (WORK-SPACE) OF LENGTH N.
C BETA    VECTOR (WORK-SPACE) OF LENGTH N.
C Z       TWO-DIMENSIONAL ARRAY OF SIZE (MM,IR+1).  THE IR + 1 COLUMNS
C         OF Z ARE ORTHONORMAL.
C
C CONSTANTS A(K), B(J) AND C(J) REQUIRED BY THIS ALGORITHM CAN BE SET
C BY THE USER.  HERE WE USE THE FOLLOWING VALUES --
C     A(K) = K*K (K=1,2,...,IR+1)
C     B(J) = 1000 (J=1,2,...,IR-1)
C     B(IR) = 1
C     C(J) = 1 (J=1,2,...,IR)
C
      REAL A(1), B(1), BETA(1), C(1), D(1), E(1), X(MMM,1), Y(MM,1)
      REAL Z(MM,1)
      REAL ALPHA, ENN, ES, PI, SSREG, SSTOT, SUMYY, T
C
C IN THE FOLLOWING DATA STATEMENT, NW IS THE PRINTER DEVICE NUMBER.
C
      DATA NW / 6 /
C
      PI = 4.0 * ATAN(1.0)
C
      M = IR + 1
      NN = N/2
      WRITE (NW,90000)
      WRITE (NW,90010) N,NN,IR,M
      IF (2*NN .NE. N) WRITE (NW,90020)
      IF (2*NN .NE. N) RETURN
      IF (IR.LT.2) WRITE (NW,90030)
      IF (IR.LT.2) RETURN
      ENN = FLOAT(NN)
      DO 40 J=1,M
        IF (J.GT.NN+1) GO TO 20
        IS = J - 1
        ES = IS
        ALPHA = 1.0/SQRT(ENN)
        IF (IS.EQ.0 .OR. IS.EQ.NN) ALPHA = 1.0/SQRT(2.0*ENN)
        DO 10 I=1,N
          T = I
          Z(I,J) = ALPHA * COS(T*PI*ES/ENN)
   10   CONTINUE
        GO TO 40
   20   IS = J - NN - 1
        ES = IS
```

```
        ALPHA = 1.0/SQRT(ENN)
        DO 30 I=1,N
          T = I
          Z(I,J) = ALPHA * SIN(T*PI*ES/ENN)
   30   CONTINUE
   40 CONTINUE
      DO 50 J=1,IR
        B(J) = 1000.0
        C(J) = 1.0
   50 CONTINUE
      B(IR) = 1.0
C
C TO CONSTRUCT AN ILL-CONDITIONED SYSTEM, ONE OR MORE OF THE B VALUES
C SHOULD BE SMALL RELATIVE TO THE OTHER B VALUES.
C
C FORM X-VECTORS.
C
      DO 80 I=1,N
        DO 70 J=1,IR
          X(I,J) = 0.0
          DO 60 K=1,J
            X(I,J) = X(I,J) + B(K)*Z(I,K)
   60     CONTINUE
          X(I,J) = C(J)*X(I,J)
   70   CONTINUE
   80 CONTINUE
C
C FORM Y-VECTOR.
C
      DO 90 K=1,M
        A(K) = FLOAT(K*K)
   90 CONTINUE
      SUMYY = 0.0
      DO 120 I=1,N
        DO 110 L=1,1
          Y(I,L) = 0.0
          DO 100 K=1,M
            Y(I,L) = Y(I,L) + A(K)*Z(I,K)
  100     CONTINUE
          SUMYY = SUMYY + Y(I,L)*Y(I,L)
  110   CONTINUE
  120 CONTINUE
C
C COMPUTE D, E AND BETA.
C
      DO 140 J=1,IR
        D(J) = 0.0
        E(J) = 0.0
        DO 130 K=1,J
          D(J) = D(J) + A(K)*B(K)
          E(J) = E(J) + B(K)*B(K)
  130   CONTINUE
  140 CONTINUE
      IR1 = IR - 1
      BETA(1) = (1.0/C(1)) * (D(1)/E(1) - (D(2)-D(1))/(E(2)-E(1)))
      BETA(IR) = (1.0/C(IR)) * ((D(IR)-D(IR1))/(E(IR)-E(IR1)))
      IF (IR1.LT.2) GO TO 160
      DO 150 J=2,IR1
```

```
        BETA(J) = (1.0/C(J))*((D(J)-D(J-1))/(E(J)-E(J-1)) -
     *    (D(J+1)-D(J))/(E(J+1)-E(J)))
  150 CONTINUE
  160 SSREG = 0.0
      DO 170 J=1,IR
        SSREG = SSREG + C(J)*D(J)*BETA(J)
  170 CONTINUE
      SSTOT = 0.0
      DO 180 K=1,M
        SSTOT = SSTOT + A(K)*A(K)
  180 CONTINUE
C
C PRINT REGRESSION SUM OF SQUARES, TOTAL SUM OF SQUARES, AND SUM OF
C Y-SQUARED.
C
      WRITE (NW,90040)
      WRITE (NW,90050) (BETA(J),J=1,IR)
      WRITE (NW,90060) SSREG,SSTOT,SUMYY
      RETURN
C
90000 FORMAT (20H0    N    NN    IR     M)
90010 FORMAT (4I5)
90020 FORMAT (48H0***  ERROR -- N (LENGTH OF VECTOR) IS NOT EVEN./)
90030 FORMAT (40H0***  ERROR -- IR CANNOT BE LESS THAN 2./)
90040 FORMAT (46H0BETA COEFFICIENTS CALCULATED FROM C, D AND E./)
90050 FORMAT (1X,8G15.8)
90060 FORMAT (20H0SSREG, SSTOT, SUMYY,5X,3G15.8)
C
      END


      FUNCTION   GEN(ANOISE)
C CHARLES L. LAWSON AND RICHARD J. HANSON,
C SOLVING LEAST SQUARES PROBLEMS, COPYRIGHT 1974, PAGE 311.
C REPRINTED BY PERMISSION OF PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, N.J.
C           GENERATE NUMBERS FOR CONSTRUCTION OF TEST CASES.
      IF (ANOISE) 10,30,20
   10 MI=891
      MJ=457
      I=5
      J=7
      AJ=0.
      GEN=0.
      RETURN
C
C     THE SEQUENCE OF VALUES OF J  IS BOUNDED BETWEEN 1 AND 996
C     IF INITIAL J = 1,2,3,4,5,6,7,8, OR 9, THE PERIOD IS 332
   20 J=J*MJ
      J=J-997*(J/997)
      AJ=J-498
C     THE SEQUENCE OF VALUES OF I  IS BOUNDED BETWEEN 1 AND 999
C     IF INITIAL I = 1,2,3,6,7, OR 9,  THE PERIOD WILL BE 50
C     IF INITIAL I = 4 OR 8   THE PERIOD WILL BE 25
C     IF INITIAL I = 5        THE PERIOD WILL BE 10
   30 I=I*MI
      I=I-1000*(I/1000)
      AI=I-500
      GEN=AI+AJ*ANOISE
```

```
      RETURN
      END


      SUBROUTINE HADMAR (N, H)
C
C SUBROUTINE HADMAR GENERATES HADAMARD MATRICES OF ORDER 2, 4, 8 AND 16.
C THESE WILL BE USED IN THE CONSTRUCTION OF LEAST SQUARES TEST PROBLEMS.
C
C INPUT VARIABLE --
C N       ORDER OF THE DESIRED HADAMARD MATRIX.
C
C OUTPUT VARIABLE --
C H       HADAMARD MATRIX OF ORDER N.
C
      INTEGER N2(2,2), N4(4,4), N8(8,8), N16(16,16)
      REAL H(16,16)
C
C IN THE FOLLOWING DATA STATEMENT, NW IS THE PRINTER DEVICE NUMBER.
C
      DATA NW / 6 /
C
C DEFINE N2, AN HADAMARD MATRIX OF ORDER 2.
C
      N2(1,1) =  1
      N2(1,2) =  1
      N2(2,1) =  1
      N2(2,2) = -1
      IF (N.EQ.2) GO TO 200
C
C DEFINE N4, AN HADAMARD MATRIX OF ORDER 4.
C
      DO 20 I=1,4
        DO 10 J=1,4
          N4(I,J) = 1
   10   CONTINUE
   20 CONTINUE
      N4(2,2) = -1
      N4(2,4) = -1
      N4(3,3) = -1
      N4(3,4) = -1
      N4(4,2) = -1
      N4(4,3) = -1
      IF (N.EQ.4) GO TO 400
C
C COMPUTE N8 AS THE TENSOR PRODUCT OF N2 AND N4.
C
      DO 40 I=1,4
        DO 30 J=1,4
          N8(I,J)      =  N4(I,J)
          N8(I+4,J)    =  N4(I,J)
          N8(I,J+4)    =  N4(I,J)
          N8(I+4,J+4) = -N4(I,J)
   30   CONTINUE
   40 CONTINUE
      IF (N.EQ.8) GO TO 800
C
C COMPUTE N16 AS THE TENSOR PRODUCT OF N4 AND N4.
```

```
C
      DO 60 I=1,4
        DO 50 J=1,4
          N16(I,J)        =  N4(I,J)
          N16(I,J+4)      =  N4(I,J)
          N16(I,J+8)      =  N4(I,J)
          N16(I,J+12)     =  N4(I,J)
          N16(I+4,J+4)    =  N4(I,J)
          N16(I+4,J+8)    = -N4(I,J)
          N16(I+4,J+12)   = -N4(I,J)
          N16(I+8,J+8)    =  N4(I,J)
          N16(I+8,J+12)   = -N4(I,J)
          N16(I+12,J+12)  =  N4(I,J)
   50   CONTINUE
   60 CONTINUE
      DO 80 I=1,16
        DO 70 J=I,16
          N16(J,I) = N16(I,J)
   70   CONTINUE
   80 CONTINUE
      IF (N.EQ.16) GO TO 1600
      WRITE (NW,90000) N
      RETURN
C
  200 DO 220 I=1,N
        DO 210 J=1,N
          H(I,J) = N2(I,J)
  210 CONTINUE
  220 CONTINUE
      RETURN
C
  400 DO 420 I=1,N
        DO 410 J=1,N
          H(I,J) = N4(I,J)
  410   CONTINUE
  420 CONTINUE
      RETURN
C
  800 DO 820 I=1,N
        DO 810 J=1,N
          H(I,J) = N8(I,J)
  810   CONTINUE
  820 CONTINUE
      RETURN
C
 1600 DO 1620 I=1,N
        DO 1610 J=1,N
          H(I,J) = N16(I,J)
 1610   CONTINUE
 1620 CONTINUE
      RETURN
C
90000 FORMAT (52H0***  ERROR -- N MUST EQUAL 2, 4, 8 OR 16.  HERE N =,
     * I5/)
C
      END
```

```
      SUBROUTINE SCALE (ISCALE, NCALL, M, N, L, W, MM, MMNN, NN, A, B,
     * R, RES, SF, U, X, IFAULT)
C
C SUBROUTINE SCALE SCALES THE DATA (A AND B) FOR LEAST SQUARES PROBLEMS
C IN ORDER TO MITIGATE, IF POSSIBLE, ROUNDING ERROR PROBLEMS WHICH CAN
C OCCUR IN CONNECTION WITH SOLVING ILL-CONDITIONED SYSTEMS OF EQUATIONS.
C
C IN THE CASE OF POLYNOMIAL TYPE PROBLEMS, THE MEAN OF THE INDEPENDENT
C VARIABLE (VECTOR A(I,2)) IS COMPUTED SO THAT IT CAN BE SUBTRACTED FROM
C EACH ELEMENT OF A(I,2) BEFORE POWERS OF THIS VECTOR ARE GENERATED.
C
C AFTER A SOLUTION IS OBTAINED FOR A SCALED PROBLEM, THE COEFFICIENTS
C (X), RESIDUALS (RES), AND COVARIANCE MATRIX (R) MUST BE ADJUSTED TO
C ACCOUNT FOR SCALING.
C
C INPUT VARIABLES --
C ISCALE PARAMETER WHICH INDICATES THE TYPE OF SCALING TO BE DONE.
C          ISCALE = 1 MEANS DATA MATRIX A IS TO BE SCALED, BUT MATRIX B
C             IS NOT TO BE SCALED.  SCALE FACTORS WILL BE AUTOMATICALLY
C             COMPUTED FROM THE DATA.  THE J-TH SCALE FACTOR, SF(J), IS
C             BASED ON THE EUCLIDEAN NORM OF THE J-TH COLUMN OF A.
C          ISCALE = 2 MEANS DATA MATRICES A AND B ARE TO BE SCALED.
C             MATRIX A IS SCALED IN THE SAME MANNER AS WHEN ISCALE = 1,
C             AND MATRIX B IS SCALED IN A SIMILAR MANNER.
C          ISCALE = 3 MEANS DATA MATRICES A AND B ARE TO BE SCALED.  THE
C             SCALE FACTORS ARE FURNISHED BY THE USER AND ARE ENTERED
C             AS INPUT DATA IN THE CALLING PROGRAM.
C NCALL    PARAMETER INDICATING WHICH OPERATIONS ARE TO BE PERFORMED.
C          NCALL = 1 MEANS THAT THE MEAN OF THE VECTOR A(I,2) IS TO BE
C             COMPUTED IN POLYNOMIAL TYPE PROBLEMS.
C          NCALL = 2 MEANS --
C             (A) SCALE FACTORS ARE TO BE COMPUTED IF ISCALE = 1 OR 2.
C             (B) MATRIX A IS TO BE SCALED.
C             (C) MATRIX B IS TO BE SCALED IF ISCALE = 2 OR 3.
C          NCALL = 3 MEANS THAT COEFFICIENTS (X), RESIDUALS (RES), AND
C             COVARIANCE MATRIX (R) ARE TO BE ADJUSTED TO ACCOUNT FOR
C             SCALING.  THE MATRIX OF OBSERVATIONS (B) WILL ALSO BE
C             ADJUSTED SO THAT IT IS RESTORED TO ITS ORIGINAL FORM (BEFORE
C             SCALING).
C M        TOTAL NUMBER OF EQUATIONS.
C N        NUMBER OF UNKNOWN COEFFICIENTS.
C L        NUMBER OF VECTORS OF OBSERVATIONS.
C W        VECTOR OF WEIGHTS (OF LENGTH M).
C MM       DIMENSIONING PARAMETER SPECIFYING MAXIMUM NUMBER OF ROWS IN
C          ARRAYS A, B AND RES.
C MMNN     DIMENSIONING PARAMETER SPECIFYING MAXIMUM NUMBER OF ROWS IN
C          ARRAY R.
C NN       DIMENSIONING PARAMETER SPECIFYING MAXIMUM NUMBER OF ROWS IN
C          ARRAY X.
C
C INPUT AND OUTPUT VARIABLES --
C A        TWO-DIMENSIONAL ARRAY OF SIZE (MM,N).  SEE MAIN PROGRAM FOR
C          FURTHER DETAILS.
C B        TWO-DIMENSIONAL ARRAY OF SIZE (MM,L).  SEE MAIN PROGRAM.
C R        TWO-DIMENSIONAL ARRAY OF SIZE (MMNN,N).  SEE MAIN PROGRAM.
C RES      TWO-DIMENSIONAL ARRAY OF SIZE (MM,L).  SEE MAIN PROGRAM.
C SF       VECTOR OF SCALE FACTORS (OF LENGTH N + L).
C U        IN POLYNOMIAL TYPE PROBLEMS U IS THE MEAN OF THE VECTOR
```

```
C         A(I,2).  IN NON-POLYNOMIAL TYPE PROBLEMS, U = 0.
C X       TWO-DIMENSICNAL ARRAY OF SIZE (NN,L).  SEE MAIN PROGRAM.
C
C OUTPUT VARIABLE --
C IFAULT FAULT INDICATOR WHICH ON EXIT EQUALS EITHER 0 OR 4.
C         IFAULT = 4 MEANS THAT AT LEAST ONE WEIGHT WAS FOUND TO BE
C            NEGATIVE.  (ALL WEIGHTS SHOULD BE NONNEGATIVE.)
C         IFAULT = 0 MEANS ALL WEIGHTS ARE NONNEGATIVE.
C
      REAL A(MM,N), B(MM,L), R(MMNN,N), RES(MM,L), SF(1), U, W(M)
      REAL X(NN,L)
      DOUBLE PRECISION SUM
C
      IF (NCALL.EQ.2) GO TO 20
      IF (NCALL.EQ.3) GO TO 120
C
C NCALL = 1.
C
      SUM = 0.0
      DO 10 I=1,M
        SUM = SUM + DBLE(A(I,2))
   10 CONTINUE
      U = SUM
      U = U / FLOAT(M)
      RETURN
C
C NCALL = 2.
C
   20 IFAULT = 0
      DO 70 J=1,N
        IF (ISCALE.EQ.3) GO TO 50
        SUM = 0.0
        SF(J) = 1.0
        DO 40 I=1,M
          IF (W(I).GE.0.0) GO TO 30
          IFAULT = 4
          CALL ERROR (IFAULT, I, W(I))
   30     SUM = SUM + DBLE(A(I,J))*DBLE(A(I,J))*DBLE(W(I))
   40   CONTINUE
        IF (IFAULT.EQ.4) RETURN
        IF (SUM.EQ.0.0) GO TO 70
        SF(J) = 1.0/DSQRT(SUM)
   50   DO 60 I=1,M
          A(I,J) = A(I,J)*SF(J)
   60   CONTINUE
   70 CONTINUE
C
      DO 110 K=1,L
        NPK = N + K
        IF (ISCALE.EQ.3) GO TO 90
        SUM = 0.0
        SF(NPK) = 1.0
        IF (ISCALE.EQ.1) GO TO 110
        DO 80 I=1,M
          SUM = SUM + DBLE(B(I,K))*DBLE(B(I,K))*DBLE(W(I))
   80   CONTINUE
        IF (SUM.EQ.0.0) GO TO 110
        SF(NPK) = 1.0/DSQRT(SUM)
```

```
   90   DO 100 I=1,M
          B(I,K) = B(I,K)*SF(NPK)
  100   CONTINUE
  110 CONTINUE
      RETURN
C
C NCALL = 3.
C
  120 DO 170 K=1,L
        NPK = N + K
        DO 130 J=1,N
          X(J,K) = (X(J,K)*SF(J)) / SF(NPK)
  130   CONTINUE
        DO 140 I=1,M
          RES(I,K) = RES(I,K) / SF(NPK)
          B(I,K) = B(I,K) / SF(NPK)
  140   CONTINUE
        IF (K.GT.1) GO TO 170
        DO 160 I=1,N
          DO 150 J=1,I
            R(I,J) = R(I,J)*SF(I)*SF(J)
  150     CONTINUE
  160   CONTINUE
  170 CONTINUE
      IF (U.EQ.0.0) RETURN
C
C IN SCALED POLYNOMIAL PROBLEMS, ADJUST COEFFICIENTS (X) AND UNSCALED
C COVARIANCE MATRIX FOR SUBTRACTION OF MEAN (U).
C REFERENCE --
C   G. A. F. SEBER, LINEAR REGRESSION ANALYSIS (1977), THEOREM
C   1.4 AND COROLLARIES, PAGES 10-11.
C
C
C FILL OUT THE SYMMETRIC COVARIANCE MATRIX SO THAT ALL ELEMENTS ARE
C PRESENT.
C
      DO 190 I=1,N
        DO 180 J=I,N
          R(I,J) = R(J,I)
  180   CONTINUE
  190 CONTINUE
      DO 260 I=1,N
        SF(I) = 1.0
        IP1 = I + 1
        IF (IP1.GT.N) GO TO 210
        DO 200 J=IP1,N
          SF(J) = -DBLE(FLOAT(J-1))/DBLE(FLOAT(J-I)) * DBLE(SF(J-1))
     *      * DBLE(U)
  200   CONTINUE
  210   DO 230 K=1,L
          SUM = 0.0
          DO 220 J=I,N
            SUM = SUM + DBLE(SF(J)) * DBLE(X(J,K))
  220     CONTINUE
          X(I,K) = SUM
  230   CONTINUE
        DO 250 J=I,N
          SUM = 0.0
```

```
          DO 240 K=I,N
            SUM = SUM + DBLE(SF(K))*DBLE(R(K,J))
  240     CONTINUE
          R(I,J) = SUM
  250   CONTINUE
  260 CONTINUE
C
      DO 310 J=1,N
        SF(J) = 1.0
        IP1 = J + 1
        IF (IP1.GT.N) GO TO 280
        DO 270 I=IP1,N
          SF(I) = -DBLE(FLOAT(I-1))/DBLE(FLOAT(I-J)) * DBLE(SF(I-1))
     *     * DBLE(U)
  270   CONTINUE
  280   DO 300 I=1,J
          SUM = 0.0
          DO 290 K=J,N
            SUM = SUM + DBLE(SF(K))*DBLE(R(I,K))
  290     CONTINUE
          R(J,I) = SUM
  300   CONTINUE
  310 CONTINUE
      RETURN
      END
```

APPENDIX B. LISTING OF DATA

DATA1A

DATA1B

DATA3A

DATA3B

DATA3C

DATA3D

DATA3E

DATA4

DATA5A

DATA5B

DATA5C

DATA5D

DATA5E

***** DATA1A *****

44		
10	2	0
10	3	0
10	4	0
10	6	0
10	8	0
10	8	4
20	2	0
20	5	0
20	10	0
20	15	0
20	15	5
30	5	0
30	10	0
30	15	0
30	20	0
30	20	10
40	5	0
40	10	0
40	20	0
40	30	0
40	30	10
50	10	0
50	20	0
50	30	0
50	40	0
50	40	10
100	10	0
100	20	0
100	40	0
100	60	0
100	80	0
100	80	10
150	10	0
150	20	0
150	40	0
150	60	0
150	80	0
150	80	10
200	10	0
200	20	0
200	40	0
200	60	0
200	80	0
200	80	10

***** DATA1B *****

16		
10	8	0
10	8	4
20	15	0
20	15	5
30	20	0

```
  30   20   10
  40   30    0
  40   30   10
  50   40    0
  50   40   10
 100   80    0
 100   80   10
 150   80    0
 150   80   10
 200   80    0
 200   80   10
```

```
*****   TEST.DATA3A   *****

  1  4
  5  0  4  1  2  4
  2  9  7  4
  1  2  4  1  6  2  0  6  5  6  2  3  3  4
  5 10  9 14  0 14  5  6  3  2  3 12
  5  9  6 25 19 27 28  0


*****   TEST.DATA3B   *****

  2  3
  3  4  0  2  5  4
  0  3  1 11
  0  6  3  5  3  4  1  2  8  5  8  7  6  7
  4  1  6 12  4 10  9  6 11  1  5 16
 29 26  3  4  6 15  8 32


*****   TEST.DATA3C   *****

  0  3
  2  1  5  3  0  3
  8 10  5  8
  8  7  5  8  7  0  7  4  3  1  4  0  0  1
 11  8 12 13 14 10  4 16  6  8 12 11
 32 16  4  7 24 24 31 15


*****   TEST.DATA3D   *****

  4  2
  0  1  4  1  1  0
 10  7  2 10
  6  3  7  2  2  2  3  4  6  1  0  7  5  8
 14 13  9 12  3 12  1 11  6  7  6 11
  0 15 28 16 12 14  9 16


*****   TEST.DATA3E   *****

  2  1
  1  2  0  5  3  5
  6 11  7  5
```

```
  7  4  5  5  2  1  7  8  5  4  3  5  0  8
 16  3 15  1  8  7  3  9  0 12  8  4
  0 18 30 29 22 14 21  2


*****  TEST.DATA4  *****

 0.  100000.   759.
 1.  111111. -2048.
 2.  124992.  2048.
 3.  142753. -2048.
 4.  165984.  2523.
 5.  196875. -2048.
 6.  238336.  2048.
 7.  294117. -2048.
 8.  368928.  1838.
 9.  468559. -2048.
10.  600000.  2048.
11.  771561. -2048.
12.  992992.  1838.
13. 1275603. -2048.
14. 1632384.  2048.
15. 2078125. -2048.
16. 2629536.  2523.
17. 3305367. -2048.
18. 4126528.  2048.
19. 5116209. -2048.
20. 6300000.   759.


***** TEST.DATA5A  *****

(1) WAMPLER, J.AMER.STAT.ASSN. 1970, P.549, 5TH DEG. POLYNOMIALS, EQUAL WEIGHTS.
   21     6     0     2     1     1     1          0.
(F2.0,2F8.0)
 0       1     760
 1       6   -2042
 2      63    2111
 3     364   -1684
 4    1365    3888
 5    3906    1858
 6    9331   11379
 7   19608   17560
 8   37449   39287
 9   66430   64382
10  111111  113159
11  177156  175108
12  271453  273291
13  402234  400186
14  579195  581243
15  813616  811568
16 1118481 1121004
17 1508598 1506550
18 2000719 2002767
19 2613660 2611612
20 3368421 3369180
    1
(2) FIRST DEGREE POLYNOMIAL, UNEQUAL WEIGHTS.
```

```
    6    2    0    1    1    2    1         0.
(3F3.0)
 1. 2. 2.
 2. 2. 1.
 3. 5. 1.
 4. 4. 1.
 5. 7. 1.
 6. 7. 2.
    1
(3) DATA WITH UNEQUAL WEIGHTS, TWO COLUMNS LINEARLY DEPENDENT.
    7    6    0    2    2    2    1         0.
(2F3.0,F3.1,F3.0,F4.2,F3.0,F5.1,F4.0,F2.0)
  1  1 .5  2 .25  2 13.0 130 2
  1  2 .5  2 .25  3 17.0 170 2
  0  3 .0  3 .00  3 18.2 182 1
  0  2 .0  1 .00  1  8.8  88 1
  0  1 .0 -3 .00  0 -3.0 -30 1
  0  1 .0  0 .00  0  2.8  28 1
  0  0 .0  1 .00  0  2.1  21 1
    1
(4) EXAMPLE WITH WEIGHTS AND CONSTRAINTS.
   12    6    3    1    2    2    1         0.
(8F3.0)
  1  1  1  1  1  1  6  1
  1  1  1  0  0  0  3  1
  1  1  0  0  0  0  2  1
  1 -1  0  0  0  0  1  3
  1  0 -1  0  0  0 -1  3
  1  0  0 -1  0  0  1  3
  1  0  0  0  0 -1 -1  2
  0  1 -1  0  0  0  1  2
  0  1  0  0 -1  0 -1  2
  0  1  0  0  0 -1  1  1
  0  0  1 -1  0  0 -1  1
  0  0  1  0 -1  0  1  1
    1
(5) INVERSE OF HILBERT MATRIX OF ORDER 4.  M = 4, N = 4, M1 = 0.
    4    4    0    1    2    1    1         0.
  (5F7.0)
    16.  -120.   240.  -140.     -4.
  -120.  1200. -2700.  1680.     60.
   240. -2700.  6480. -4200.   -180.
  -140.  1680. -4200.  2800.    140.
    1
(6) INVERSE OF HILBERT MATRIX OF ORDER 4.  M = 4, N = 4, M1 = 4.
    4    4    4    1    2    1    1         0.
  (5F7.0)
    16.  -120.   240.  -140.     -4.
  -120.  1200. -2700.  1680.     60.
   240. -2700.  6480. -4200.   -180.
  -140.  1680. -4200.  2800.    140.
    1
(7) BUSINGER-GOLUB, NUM. MATH. 1965, P.269, INVERSE OF HILBERT MATRIX, ORDER 6.
    6    5    1    2    2    1    1         0.
(5F10.0,10X,2F10.0)
       36.      -630.      3360.     -7560.      7560.                      463.       463.
     -630.     14700.    -88200.    211680.   -220500.                   -13860.    -17820.
     3360.    -88200.    564480.  -1411200.   1512000.                    97020.     93555.
```

```
   -7560.   211680. -1411200.  3628800. -3969000.                -258720.  -261800.
    7560.  -220500.  1512000. -3969000.  4410000.                 291060.   288288.
   -2772.    83160.  -582120.  1552320. -1746360.                -116424.  -118944.
    1
(8) EXAMPLE WITH X = 0 (HENCE XNORM = 0).  TOL = -1 ON ENTRY TO L2A OR L2B.
    1    1    0    1    2    1    0        -1.
(2F3.0)
 1. 0.
    1
(9) ALBERT, REGRESSION AND THE MOORE-PENROSE INVERSE, 1972, P. 63.
    3    4    0    3    2    1    2         0.
(7F4.0)
  1.  0.  1.  1.  1.  0.  0.
  0.  1. -1.  0.  0.  1.  0.
  1.  1.  0.  1.  0.  0.  1.
    1
(10) FIFTH DEGREE POLYNOMIAL WITH HEAVY WEIGHTS, MATRIX A SCALED.      IFAULT=11
   21    6    0    1    2    2    1         0.
(F8.0,3F9.0,2F10.0,F13.2,F10.0)
1000000.       0.        0.        0.         0.         0.  100000.1853 16777216.
1000000.  100000.    10000.     1000.       100.         1.  -8277497.00        1.
1000000.  200000.    40000.     8000.      1600.        32.   8513600.00        1.
1000000.  300000.    90000.    27000.      8100.       243.  -8245855.00        1.
1000000.  400000.   160000.    64000.     25600.      1024.  10500192.00        1.
1000000.  500000.   250000.   125000.     62500.      3125.  -8191733.00        1.
1000000.  600000.   360000.   216000.    129600.      7776.   8626944.00        1.
1000000.  700000.   490000.   343000.    240100.     16807.  -8094491.00        1.
1000000.  800000.   640000.   512000.    409600.     32768.   7897376.00        1.
1000000.  900000.   810000.   729000.    656100.     59049.  -7920049.00        1.
1000000. 1000000.  1000000.  1000000.   1000000.    100000.     600000.50 16777216.
1000000. 1100000.  1210000.  1331000.   1464100.    161051.  -7617047.00        1.
1000000. 1200000.  1440000.  1728000.   2073600.    248832.   8521440.00        1.
1000000. 1300000.  1690000.  2197000.   2856100.    371293.  -7113005.00        1.
1000000. 1400000.  1960000.  2744000.   3841600.    537824.  10020992.00        1.
1000000. 1500000.  2250000.  3375000.   5062500.    759375.  -6310483.00        1.
1000000. 1600000.  2560000.  4096000.   6553600.   1048576.  12963744.00        1.
1000000. 1700000.  2890000.  4913000.   8352100.   1419857.  -5083241.00        1.
1000000. 1800000.  3240000.  5832000.  10497600.   1889568.  12515136.00        1.
1000000. 1900000.  3610000.  6859000.  13032100.   2476099.  -3272399.00        1.
1000000. 2000000.  4000000.  8000000.  16000000.   3200000. 6300000.1853 16777216.
    1
(11) LAWSON-HANSON, SOLVING LEAST SQUARES PROBLEMS, 1974, SET 1 EX.16. IFAULT=10
    8    6    4    1    2    1    1         0.
(7F6.0)
  155.  105. -445. -495.  -45.  -95. -245.
  355.  305. -245. -295.  155.  105. -295.
 -445. -495.  -45.  -95.  355.  305.  155.
 -245. -295.  155.  105. -445. -495.  105.
  -45.  -95.  355.  305. -245. -295. -445.
  155.  105. -445. -495.  -45.  -95. -495.
  355.  305. -245. -295.  155.  105.  -45.
 -445. -495.  -45.  -95.  355.  305.  -95.
    1
(12) LAWSON-HANSON, SOLVING LEAST SQUARES PROBLEMS, P.252, SET 1, EX.16, TOL=.5.
    8    6    4    1    2    1    2        0.5
(7F6.0)
  155.  105. -445. -495.  -45.  -95. -245.
  355.  305. -245. -295.  155.  105. -295.
```

```
 -445. -495.   -45.   -95.  355.  305.  155.
 -245. -295.  155.  105. -445. -495.  105.
  -45.   -95.  355.  305. -245. -295. -445.
  155.  105. -445. -495.   -45.   -95. -495.
  355.  305. -245. -295.  155.  105.   -45.
 -445. -495.   -45.   -95.  355.  305.   -95.
    1
(13) BJORCK-GOLUB, BIT 1967, P.322, HILBERT MATRIX INVERSE, ORDER 8.  IFAULT=8,9
    8    6    2    3    2    1    1         0.
(6F12.0)
      20160.      -92400.      221760.     -288288.      192192.      -51480.
        945.         945.     8400945.
    -952560.     4656960.   -11642400.    15567552.   -10594584.     2882880.
     -40320.      -40320.     4159680.
   11430720.   -58212000.   149688000.  -204324120.   141261120.   -38918880.
     456120.     3256120.     3256120.
  -58212000.   304920000.  -800415000.  1109908800.  -776936160.   216216000.
    -2236080.     -136080.     -136080.
  149688000.  -800415000.  2134440000.-2996753760.  2118916800.  -594594000.
    5599440.     7279440.     7279440.
 -204324120.  1109908800.-2996753760.  4249941696.-3030051024.   856215360.
    -7495488.    -6095488.    -6095488.
  141261120.  -776936160.  2118916800.-3030051024.  2175421248.  -618377760.
    5105100.     6305100.     6305100.
  -38918880.   216216000.  -594594000.   856215360.  -618377760.   176679360.
   -1389960.     -339960.     -339960.
    1
(14) LAWSON-HANSON, SOLVING LEAST SQUARES PROBLEMS, 1974, SET 1, EX.12. IFAULT=7
    6    8    6    1    2    1    1         0.
(9F6.0)
 -245. -295.  155.  105. -445. -495.   -45.   -95.  355.     1.
  355.  305. -245. -295.  155.  105. -445. -495.  305.     1.
  -45.   -95.  355.  305. -245. -295.  155.  105. -245.     1.
 -445. -495.   -45.   -95.  355.  305. -245. -295. -295.     4.
  155.  105. -445. -495.   -45.   -95.  355.  305.  155.     9.
 -245. -295.  155.  105. -445. -495.   -45.   -95.  105.    16.
    1
(15) EXAMPLE WITH SINGULAR MATRIX OF CONSTRAINTS.  M1 = 3, N1 = 2.      IFAULT=6
    6    3    3    1    2    1    1         0.
(4F2.0)
 1 1 1 1
 2 2 2 1
 1 0 0 1
 1 2 4 1
 1 3 9 1
 1 4 9 1
    1
(16) EXAMPLE WITH MATRIX A EQUAL TO ZERO (HENCE RANK EQUALS ZERO).       IFAULT=5
    3    2    0    1    2    1    1         0.
(3F2.0)
 0 0 1
 0 0 1
 0 0 1
    1
(17) EXAMPLE WITH ZERO AND NEGATIVE WEIGHTS.                            IFAULT=4
    2    1    0    1    2    2    1         0.
(2F3.0,F4.0)
 1. 1.  0.
```

```
 1. 1. -1.
    1
(18) EXAMPLE WHERE N EXCEEDS THE CORRESPONDING DIMENSION LIMIT.          IFAULT=3
    1   21    0    1    2    1    2          0.
   (22F2.0)
 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 1
    1
(19) EXAMPLE WHERE M1 EXCEEDS M AND N.                                   IFAULT=2
    1    1    2    1    2    1    1          0.
(2F3.0)
 1. 1.
    1
(20) EXAMPLE WITH M = 0.                                                 IFAULT=1
    0    1    0    1    2    1    1          0.
(2F3.0)
 1. 1. 1.
    0


*****  TEST.DATA5B  *****

(21) FIRST DEGREE POLYNOMIAL, POSITIVE AND ZERO WEIGHTS.  COMPARE EXAMPLE (2).
    8    2    0    1    1    2               0.
(3F3.0)
 1. 2. 2.
 2. 2. 1.
 3. 5. 1.
 4. 4. 1.
 5. 7. 1.
 6. 7. 2.
 7. 9. 0.
 8. 6. 0.
    1
(22) EXAMPLE WITH WEIGHTS AND CONSTRAINTS.  COMPARE EXAMPLE (4).
   14    6    3    1    2    2               0.
(8F3.0)
  1  1  1  1  1  1  6  1
  1  1  1  0  0  0  3  1
  1  1  0  0  0  0  2  1
  1 -1  0  0  0  0  1  3
  1  0 -1  0  0  0 -1  3
  1  0  0 -1  0  0  1  3
  1  0  0  0  0 -1 -1  2
  0  1 -1  0  0  0  1  2
  0  1  0  0 -1  0 -1  2
  0  1  0  0  0 -1  1  1
  0  0  1 -1  0  0 -1  1
  0  0  1  0 -1  0  1  1
  1  0  0  0 -1  0 -1  0
  0  1  0 -1  0  0  1  0
    1
(23) EXAMPLE WITH ZERO WEIGHTS, WHERE RANK A = RANK H = 2.  H = (SQRT(W))*A.
    4    2    0    1    2    2               0.
(4F3.0)
 1. 0. 1. 1.
 0. 1. 2. 1.
 0. 0. 3. 0.
 0. 0. 4. 0.
```

```
    1
(24) EXAMPLE WITH ZERO WEIGHTS, WHERE RANK A = 2, RANK H = 1.  H = (SQRT(W))*A.
    4    2    0    1    2    2              0.
(4F3.0)
 1. 0. 1. 1.
 0. 1. 2. 0.
 0. 0. 3. 0.
 0. 0. 4. 1.
    0


*****  TEST.DATA5C  *****

(3) DATA WITH UNEQUAL WEIGHTS, TWO COLUMNS LINEARLY DEPENDENT.
    7    6    0    2    2    2    1         0.
(2F3.0,F3.1,F3.0,F4.2,F3.0,F5.1,F4.0,F2.0)
  1  1 .5  2 .25  2 13.0 130 2
  1  2 .5  2 .25  3 17.0 170 2
  0  3 .0  3 .00  3 18.2 182 1
  0  2 .0  1 .00  1  8.8  88 1
  0  1 .0 -3 .00  0 -3.0 -30 1
  0  1 .0  0 .00  0  2.8  28 1
  0  0 .0  1 .00  0  2.1  21 1
    1
(9) ALBERT, REGRESSION AND THE MOORE-PENROSE INVERSE, 1972, P. 63.
    3    4    0    3    2    1    2         0.
(7F4.0)
  1.  0.  1.  1.  1.  0.  0.
  0.  1. -1.  0.  0.  1.  0.
  1.  1.  0.  1.  0.  0.  1.
    1
(12) LAWSON-HANSON, SOLVING LEAST SQUARES PROBLEMS, P.252, SET 1, EX.16, TOL=.5.
    8    6    4    1    2    1    2         0.5
(7F6.0)
  155.  105. -445. -495.  -45.  -95. -245.
  355.  305. -245. -295.  155.  105. -295.
 -445. -495.  -45.  -95.  355.  305.  155.
 -245. -295.  155.  105. -445. -495.  105.
  -45.  -95.  355.  305. -245. -295. -445.
  155.  105. -445. -495.  -45.  -95. -495.
  355.  305. -245. -295.  155.  105.  -45.
 -445. -495.  -45.  -95.  355.  305.  -95.
    1
(14) LAWSON-HANSON, SOLVING LEAST SQUARES PROBLEMS, 1974, SET 1, EX.12. IFAULT=7
    6    8    6    1    2    1    1         0.
(9F6.0)
 -245. -295.  155.  105. -445. -495.  -45.  -95.  355.     1.
  355.  305. -245. -295.  155.  105. -445. -495.  305.     1.
  -45.  -95.  355.  305. -245. -295.  155.  105. -245.     1.
 -445. -495.  -45.  -95.  355.  305. -245. -295. -295.     4.
  155.  105. -445. -495.  -45.  -95.  355.  305.  155.     9.
 -245. -295.  155.  105. -445. -495.  -45.  -95.  105.    16.
    1
(18) EXAMPLE WHERE N EXCEEDS THE CORRESPONDING DIMENSION LIMIT.          IFAULT=3
    1   21    0    1    2    1    2         0.
   (22F2.0)
 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 1
    0
```

```
*****  TEST.DATA5D  *****

(25) BENNETT AND FRANKLIN, 1954, PP. 379-385.  MISSING DATA.
   20    8    0    1    2    2              0.
(10F4.0)
  1.  1.  0.  0.  0.  1.  0.  0. 28.  1.
  1.  1.  0.  0.  0.  0.  1.  0. 20.  1.
  1.  1.  0.  0.  0.  0.  0.  1. 11.  1.
  1.  1.  0.  0.  0. -1. -1. -1. 10.  1.
  1.  0.  1.  0.  0.  1.  0.  0.  0.  0.
  1.  0.  1.  0.  0.  0.  1.  0. 16.  1.
  1.  0.  1.  0.  0.  0.  0.  1. 15.  1.
  1.  0.  1.  0.  0. -1. -1. -1.  8.  1.
  1.  0.  0.  1.  0.  1.  0.  0. 29.  1.
  1.  0.  0.  1.  0.  0.  1.  0. 13.  1.
  1.  0.  0.  1.  0.  0.  0.  1. 16.  1.
  1.  0.  0.  1.  0. -1. -1. -1.  0.  0.
  1.  0.  0.  0.  1.  1.  0.  0. 27.  1.
  1.  0.  0.  0.  1.  0.  1.  0. 10.  1.
  1.  0.  0.  0.  1.  0.  0.  1. 18.  1.
  1.  0.  0.  0.  1. -1. -1. -1. 11.  1.
  1. -1. -1. -1. -1.  1.  0.  0. 28.  1.
  1. -1. -1. -1. -1.  0.  1.  0. 11.  1.
  1. -1. -1. -1. -1.  0.  0.  1. 15.  1.
  1. -1. -1. -1. -1. -1. -1. -1. 10.  1.
   1
(26) NBS TECHNICAL NOTE 552, TWOWAY ANOVA, PAGES 134-135 -- WITH ZERO WEIGHTS.
   12    6    0    1    2    2              0.
(8F3.0)
  1  1  0  1  0  0  8  1
  1  1  0  0  1  0  2  0
  1  1  0  0  0  1  1  0
  1  1  0 -1 -1 -1  3  0
  1  0  1  1  0  0  4  0
  1  0  1  0  1  0  0  1
  1  0  1  0  0  1  3  1
  1  0  1 -1 -1 -1  1  1
  1 -1 -1  1  0  0  2  0
  1 -1 -1  0  1  0  1  1
  1 -1 -1  0  0  1  0  1
  1 -1 -1 -1 -1 -1  4  1
   1
(27) 6TH DEGREE POLYNOMIAL FITTED TO 5TH DEGREE DATA -- SCALING IS NOT USED.
   21    7    0    2    1    1              0.
(F2.0,2F8.0)
 0       1     760
 1       6   -2042
 2      63    2111
 3     364   -1684
 4    1365    3888
 5    3906    1858
 6    9331   11379
 7   19608   17560
 8   37449   39287
 9   66430   64382
10  111111  113159
```

```
11  177156  175108
12  271453  273291
13  402234  400186
14  579195  581243
15  813616  811568
16 1118481 1121004
17 1508598 1506550
18 2000719 2002767
19 2613660 2611612
20 3368421 3369180
    1
(28) WAMPLER, J.AMER.STAT.ASSN. 1970, P.549, 5TH DEG. POLYNOMIAL, EQUAL WEIGHTS.
   21    6    0    1    1    1    1         0.
(F3.0,8X,F8.0)
100        1      760
101        6    -2042
102       63     2111
103      364    -1684
104     1365     3888
105     3906     1858
106     9331    11379
107    19608    17560
108    37449    39287
109    66430    64382
110   111111   113159
111   177156   175108
112   271453   273291
113   402234   400186
114   579195   581243
115   813616   811568
116  1118481  1121004
117  1508598  1506550
118  2000719  2002767
119  2613660  2611612
120  3368421  3369180
    0


*****  TEST.DATASE  *****

(27) 6TH DEG. POLYNOMIAL FITTED TO 5TH DEG. DATA.  SCALING USED (ISCALE = 3).
   21    7    0    2    1    1              0.
(F2.0,2F8.0)
 0       1      760
 1       6    -2042
 2      63     2111
 3     364    -1684
 4    1365     3888
 5    3906     1858
 6    9331    11379
 7   19608    17560
 8   37449    39287
 9   66430    64382
10  111111   113159
11  177156   175108
12  271453   273291
13  402234   400186
14  579195   581243
```

```
15  813616  811568
16 1118481 1121004
17 1508598 1506550
18 2000719 2002767
19 2613660 2611612
20 3368421 3369180
 1000000.0 100000.00 10000.000 1000.0000 100.00000 10.000000 1.0000000 1.0000000
 1.0000000
    0
```

APPENDIX C. SAMPLE OUTPUT FROM FIVE PROBLEMS USING TEST5

Fig. 2. Example 9 from DATA5C, MODE = 1, ISCALE = 0.

Fig. 3. Example 9 from DATA5C, MODE = 2, ISCALE = 0.

Fig. 4. Example 22 from DATA5B, MODE = 1, ISCALE = 0.

Fig. 5. Example 28 from DATA5D, MODE = 1, ISCALE = 0.

Fig. 6. Example 28 from DATA5D, MODE = 1, ISCALE = 2.

The five examples given in this appendix illustrate various features of the L2A and L2B algorithms. Input parameters and input data (matrix A, matrix B, and vector of weights) are printed. If data are to be scaled before solutions are computed, scale factors are printed. Computed results include the computed rank of the system of equations; the sequence in which columns were selected by the pivoting scheme; information on the behavior of the iterative refinement procedure; coefficients and their standard deviations; observations, predicted values and residuals; sum of squared residuals, norm of residuals, and residual standard deviation; and unscaled covariance matrix.

```
(9) ALBERT, REGRESSION AND THE MOORE-PENROSE INVERSE, 1972, P. 63.

   M    N   M1    L     ITYPE     IWGHT      MODE    ISCALE       TOL
   3    4    0    3         2         1         1         0   .00000000
FORMAT (7F4.0)

MATRIX A, MATRIX B AND VECTOR OF WEIGHTS

 1.0000000    .00000000    1.0000000    1.0000000    1.0000000    .00000000    .00000000    1.0000
 .00000000    1.0000000   -1.0000000    .00000000    .00000000    1.0000000    .00000000    1.0000
 1.0000000    1.0000000    .00000000    1.0000000    .00000000    .00000000    1.0000000    1.0000

COMPUTED RESULTS

MODE =   1      IFAULT =   0

N1 = COMPUTED RANK OF SYSTEM OF EQUATIONS =   2

COLUMNS OF H = (SQRT(W))*A WERE SELECTED BY THE PIVOTING SCHEME IN THE FOLLOWING ORDER

  1   2

THE FOLLOWING COLUMNS OF H ARE LINEARLY DEPENDENT.  IF MODE 1, THEY DID NOT ENTER THE REGRESSION.
IF MODE 2, THEY ENTERED LAST.

  3   4

NUMBER OF ZERO WEIGHTS =  0      DEG. OF FREEDOM =  1

               REPORT ON               NUMBER OF    ESTIMATED NUMBER OF CORRECT
B-VECTOR NO.   CONVERGENCE             ITERATIONS   DIGITS IN INITIAL SOLUTION

     1         CONVERGED                   1.              7.8267799
     2         CONVERGED                   2.              7.8267799
     3         CONVERGED                   1.              7.8267799

SOLUTION FOR B-VECTOR NO.  1

                             STANDARD DEVIATION
    J     COEFFICIENT(J)      OF COEFFICIENT(J)

    1      .66666666            .47140452
    2     -.33333333            .47140452
    3      .00000000            .00000000
    4      .00000000            .00000000

    I     OBSERVED(I)          PREDICTED(I)           RESIDUAL(I)            WEIGHT(I)

    1      1.0000000            .66666667              .33333334             1.0000000
    2      .00000000           -.33333333              .33333333             1.0000000
```

```
SUM OF SQUARED RESIDUALS     =   .33333334
NORM OF RESIDUALS            =   .57735027
RESIDUAL STANDARD DEVIATION  =   .57735027

SOLUTION FOR B-VECTOR NO.  2

                            STANDARD DEVIATION
   J     COEFFICIENT(J)      OF COEFFICIENT(J)

   1     -.33333333            .47140452
   2      .66666666            .47140452
   3      .00000000            .00000000
   4      .00000000            .00000000

   I     OBSERVED(I)         PREDICTED(I)          RESIDUAL(I)          WEIGHT(I)

   1      .00000000           -.33333333            .33333333          1.0000000
   2     1.0000000             .66666667            .33333334          1.0000000
   3      .00000000            .33333333           -.33333333          1.0000000

SUM OF SQUARED RESIDUALS     =   .33333333
NORM OF RESIDUALS            =   .57735027
RESIDUAL STANDARD DEVIATION  =   .57735027

SOLUTION FOR B-VECTOR NO.  3

                            STANDARD DEVIATION
   J     COEFFICIENT(J)      OF COEFFICIENT(J)

   1      .33333333            .47140452
   2      .33333333            .47140452
   3      .00000000            .00000000
   4      .00000000            .00000000

   I     OBSERVED(I)         PREDICTED(I)          RESIDUAL(I)          WEIGHT(I)

   1      .00000000            .33333334           -.33333334          1.0000000
   2      .00000000            .33333333           -.33333333          1.0000000
   3     1.0000000             .66666667            .33333334          1.0000000

SUM OF SQUARED RESIDUALS     =   .33333334
NORM OF RESIDUALS            =   .57735027
RESIDUAL STANDARD DEVIATION  =   .57735027

UNSCALED COVARIANCE MATRIX

   .66666666
  -.33333333      .66666666
   .00000000      .00000000      .00000000
   .00000000      .00000000      .00000000      .00000000
```

Fig. 2. Example 9 from DATA5C, using MODE 1. The 3 × 4 matrix A is found to be of rank 2. Columns 3 and 4 of A are ignored in obtaining the solution.

```
(9) ALBERT, REGRESSION AND THE MOORE-PENROSE INVERSE, 1972, P. 63.

   M    N   M1    L      ITYPE     IWGHT      MODE    ISCALE        TOL
   3    4    0    3          2         1         2         0   .00000000

FORMAT (7F4.0)

MATRIX A, MATRIX B AND VECTOR OF WEIGHTS

 1.0000000    .00000000   1.0000000   1.0000000   1.0000000    .00000000    .00000000   1.0000
 .00000000   1.0000000   -1.0000000    .00000000   .00000000   1.0000000    .00000000   1.0000
 1.0000000   1.0000000    .00000000   1.0000000    .00000000    .00000000   1.0000000   1.0000

COMPUTED RESULTS

MODE =   2      IFAULT =   0

N1 = COMPUTED RANK OF SYSTEM OF EQUATIONS =   2

COLUMNS OF H = (SQRT(W))*A WERE SELECTED BY THE PIVOTING SCHEME IN THE FOLLOWING ORDER

  1   2

THE FOLLOWING COLUMNS OF H ARE LINEARLY DEPENDENT.  IF MODE 1, THEY DID NOT ENTER THE REGRESSION.
IF MODE 2, THEY ENTERED LAST.

  3   4

NUMBER OF ZERO WEIGHTS =  0      DEG. OF FREEDOM =  1

                 REPORT ON               NUMBER OF    ESTIMATED NUMBER OF CORRECT
B-VECTOR NO.     CONVERGENCE             ITERATIONS   DIGITS IN INITIAL SOLUTION

     1           CONVERGED                   2.              7.7096676
     2           CONVERGED                   2.              7.8267799
     3           CONVERGED                   1.              7.8267799

SOLUTION FOR B-VECTOR NO.  1

    J     COEFFICIENT(J)     (Column 1 of Pseudoinverse)

    1      .20000000
    2     -.66666667-01
    3      .26666667
    4      .20000000
```

```
  I     OBSERVED(I)        PREDICTED(I)        RESIDUAL(I)         WEIGHT(I)

  1      1.0000000          .66666667           .33333334          1.0000000
  2      .00000000         -.33333333           .33333333          1.0000000
  3      .00000000          .33333334          -.33333334          1.0000000

SUM OF SQUARED RESIDUALS   =   .33333334
NORM OF RESIDUALS          =   .57735027

SOLUTION FOR B-VECTOR NO.  2

  J     COEFFICIENT(J)    (Column 2 of Pseudoinverse)

  1      .00000000
  2      .33333334
  3     -.33333333
  4      .30531133-16

  I     OBSERVED(I)        PREDICTED(I)        RESIDUAL(I)         WEIGHT(I)

  1      .00000000         -.33333333           .33333333          1.0000000
  2      1.0000000          .66666667           .33333334          1.0000000
  3      .00000000          .33333333          -.33333333          1.0000000

SUM OF SQUARED RESIDUALS   =   .33333333
NORM OF RESIDUALS          =   .57735027

SOLUTION FOR B-VECTOR NO.  3

  J     COEFFICIENT(J)    (Column 3 of Pseudoinverse)

  1      .20000000
  2      .26666667
  3     -.66666665-01
  4      .20000000

  I     OBSERVED(I)        PREDICTED(I)        RESIDUAL(I)         WEIGHT(I)

  1      .00000000          .33333334          -.33333334          1.0000000
  2      .00000000          .33333333          -.33333333          1.0000000
  3      1.0000000          .66666667           .33333334          1.0000000

SUM OF SQUARED RESIDUALS   =   .33333334
NORM OF RESIDUALS          =   .57735027
```

Fig. 3. Example 9 from DATA5C, using MODE 2. The three solution vectors computed by subroutine L2B give the pseudoinverse of the 3×4 matrix A.

(22) EXAMPLE WITH WEIGHTS AND CONSTRAINTS. COMPARE EXAMPLE (4).

M	N	M1	L	ITYPE	IWGHT	MODE	ISCALE	TOL
14	6	3	1	2	2	1	0	.00000000

FORMAT (8F3.0)

MATRIX A, MATRIX B AND VECTOR OF WEIGHTS

1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	6.0000000	1.0000
1.0000000	1.0000000	1.0000000	.00000000	.00000000	.00000000	3.0000000	1.0000
1.0000000	1.0000000	.00000000	.00000000	.00000000	.00000000	2.0000000	1.0000
1.0000000	-1.0000000	.00000000	.00000000	.00000000	.00000000	1.0000000	3.0000
1.0000000	.00000000	-1.0000000	.00000000	.00000000	.00000000	-1.0000000	3.0000
1.0000000	.00000000	.00000000	-1.0000000	.00000000	.00000000	1.0000000	3.0000
1.0000000	.00000000	.00000000	.00000000	.00000000	-1.0000000	-1.0000000	2.0000
.00000000	1.0000000	-1.0000000	.00000000	.00000000	.00000000	1.0000000	2.0000
.00000000	1.0000000	.00000000	.00000000	-1.0000000	.00000000	-1.0000000	2.0000
.00000000	1.0000000	.00000000	.00000000	.00000000	-1.0000000	1.0000000	1.0000
.00000000	.00000000	1.0000000	-1.0000000	.00000000	.00000000	-1.0000000	1.0000
.00000000	.00000000	1.0000000	.00000000	-1.0000000	.00000000	1.0000000	1.0000
1.0000000	.00000000	.00000000	.00000000	-1.0000000	.00000000	-1.0000000	.00000
.00000000	1.0000000	.00000000	-1.0000000	.00000000	.00000000	1.0000000	.00000

COMPUTED RESULTS

MODE = 1 IFAULT = 0

N1 = COMPUTED RANK OF SYSTEM OF EQUATIONS = 6

COLUMNS OF H = (SQRT(W))*A WERE SELECTED BY THE PIVOTING SCHEME IN THE FOLLOWING ORDER

1 3 4 2 6 5

NUMBER OF ZERO WEIGHTS = 2 DEG. OF FREEDOM = 6

B-VECTOR NO.	REPORT ON CONVERGENCE	NUMBER OF ITERATIONS	ESTIMATED NUMBER OF CORRECT DIGITS IN INITIAL SOLUTION

1 CONVERGED 2. 7.5180585

SOLUTION FOR B-VECTOR NO. 1

J	COEFFICIENT(J)	STANDARD DEVIATION OF COEFFICIENT(J)
1	1.0512821	.35744484
2	.94871794	.35744484
3	.99999999	.00000000
4	.48717948	.73885909
5	1.2307692	.81367340
6	1.2820513	.76207496

I	OBSERVED(I)	PREDICTED(I)	RESIDUAL(I)	WEIGHT(I)
1	6.0000000	6.0000000	.00000000	1.0000000
2	3.0000000	3.0000000	.00000000	1.0000000
3	2.0000000	2.0000000	.00000000	1.0000000
4	1.0000000	.10256411	.89743590	3.0000000
5	-1.0000000	.51282048-01	-1.0512820	3.0000000
6	1.0000000	.56410258	.43589744	3.0000000
7	-1.0000000	-.23076923	-.76923077	2.0000000
8	1.0000000	-.51282048-01	1.0512820	2.0000000
9	-1.0000000	-.28205130	-.71794871	2.0000000
10	1.0000000	-.33333334	1.3333333	1.0000000
11	-1.0000000	.51282051	-1.5128205	1.0000000
12	1.0000000	-.23076923	1.2307692	1.0000000
13	-1.0000000	-.17948717	-.82051283	.00000000
14	1.0000000	.46153846	.53846154	.00000000

SUM OF SQUARED RESIDUALS = 16.307692
NORM OF RESIDUALS = 4.0382783
RESIDUAL STANDARD DEVIATION = 1.6486202

UNSCALED COVARIANCE MATRIX

.47008546-01					
-.47008546-01	.47008546-01				
.58207661-10	-.58207661-10	.00000000			
.29914529-01	-.29914529-01	.00000000	.20085470		
-.38461538-01	.38461538-01	.00000000	-.11538462	.24358974	
.85470086-02	-.85470086-02	.00000000	-.85470084-01	-.12820513	.21367521

Fig. 4. Example 22 from DATA5B, using MODE 1. The first three equations are linear constraints. Observations are given unequal weights. The last two observations, given zero weights, are ignored in obtaining the solution, but residuals are computed for these observations.

(28) WAMPLER, J.AMER.STAT.ASSN. 1970, P.549, 5TH DEG. POLYNOMIAL, EQUAL WEIGHTS.

M	N	M1	L	ITYPE	IWGHT	MODE	ISCALE	TOL
21	6	0	1	1	1	1	0	.00000000

FORMAT (F3.0,8X,F8.0)

MATRIX A, MATRIX B AND VECTOR OF WEIGHTS

1.0000000	100.00000	10000.000	1000000.0	.10000000+09	.10000000+11	760.00000	1.0000
1.0000000	101.00000	10201.000	1030301.0	.10406040+09	.10510100+11	-2042.0000	1.0000
1.0000000	102.00000	10404.000	1061208.0	.10824322+09	.11040808+11	2111.0000	1.0000
1.0000000	103.00000	10609.000	1092727.0	.11255088+09	.11592741+11	-1684.0000	1.0000
1.0000000	104.00000	10816.000	1124864.0	.11698586+09	.12166529+11	3888.0000	1.0000
1.0000000	105.00000	11025.000	1157625.0	.12155062+09	.12762816+11	1858.0000	1.0000
1.0000000	106.00000	11236.000	1191016.0	.12624770+09	.13382256+11	11379.000	1.0000
1.0000000	107.00000	11449.000	1225043.0	.13107960+09	.14025517+11	17560.000	1.0000
1.0000000	108.00000	11664.000	1259712.0	.13604890+09	.14693281+11	39287.000	1.0000
1.0000000	109.00000	11881.000	1295029.0	.14115816+09	.15386239+11	64382.000	1.0000
1.0000000	110.00000	12100.000	1331000.0	.14641000+09	.16105100+11	113159.00	1.0000
1.0000000	111.00000	12321.000	1367631.0	.15180704+09	.16850582+11	175108.00	1.0000
1.0000000	112.00000	12544.000	1404928.0	.15735194+09	.17623417+11	273291.00	1.0000
1.0000000	113.00000	12769.000	1442897.0	.16304736+09	.18424352+11	400186.00	1.0000
1.0000000	114.00000	12996.000	1481544.0	.16889602+09	.19254146+11	581243.00	1.0000
1.0000000	115.00000	13225.000	1520875.0	.17490063+09	.20113572+11	811568.00	1.0000
1.0000000	116.00000	13456.000	1560896.0	.18106394+09	.21003417+11	1121004.0	1.0000
1.0000000	117.00000	13689.000	1601613.0	.18738872+09	.21924480+11	1506550.0	1.0000
1.0000000	118.00000	13924.000	1643032.0	.19387778+09	.22877577+11	2002767.0	1.0000
1.0000000	119.00000	14161.000	1685159.0	.20053392+09	.23863537+11	2611612.0	1.0000
1.0000000	120.00000	14400.000	1728000.0	.20736000+09	.24883200+11	3369180.0	1.0000

COMPUTED RESULTS

MODE = 1 IFAULT = 0

N1 = COMPUTED RANK OF SYSTEM OF EQUATIONS = 3

COLUMNS OF H = (SQRT(W))*A WERE SELECTED BY THE PIVOTING SCHEME IN THE FOLLOWING ORDER

6 5 4

THE FOLLOWING COLUMNS OF H ARE LINEARLY DEPENDENT. IF MODE 1, THEY DID NOT ENTER THE REGRESSION.
IF MODE 2, THEY ENTERED LAST.

3 2 1

NUMBER OF ZERO WEIGHTS = 0 DEG. OF FREEDOM = 18

REPORT ON NUMBER OF ESTIMATED NUMBER OF CORRECT

SOLUTION FOR B-VECTOR NO. 1

J	COEFFICIENT(J)	STANDARD DEVIATION OF COEFFICIENT(J)
1	.00000000	.00000000
2	.00000000	.00000000
3	.00000000	.00000000
4	109.06503	9.5677701
5	-2.0588078	.17304154
6	.97053145-02	.78063607-03

I	OBSERVED(I)	PREDICTED(I)	RESIDUAL(I)	WEIGHT(I)
1	760.00000	237398.82	-236638.82	1.0000000
2	-2042.0000	133278.79	-135320.79	1.0000000
3	2111.0000	43223.706	-41112.706	1.0000000
4	-1684.0000	-31129.579	29445.579	1.0000000
5	3888.0000	-88071.863	91959.863	1.0000000
6	1858.0000	-125825.35	127683.35	1.0000000
7	11379.000	-142539.90	153918.90	1.0000000
8	17560.000	-136291.81	153851.81	1.0000000
9	39287.000	-105085.04	144372.04	1.0000000
10	64382.000	-46849.543	111231.54	1.0000000
11	113159.00	40571.487	72587.513	1.0000000
12	175108.00	159392.79	15715.206	1.0000000
13	273291.00	311930.34	-38639.343	1.0000000
14	400186.00	500558.44	-100372.44	1.0000000
15	581243.00	727752.62	-146509.63	1.0000000
16	811568.00	996051.75	-184483.75	1.0000000
17	1121004.0	1308095.9	-187091.89	1.0000000
18	1506550.0	1666590.9	-160040.92	1.0000000
19	2002767.0	2074348.9	-71581.909	1.0000000
20	2611612.0	2534250.2	77361.786	1.0000000
21	3369180.0	3049276.4	319903.59	1.0000000

SUM OF SQUARED RESIDUALS = .42885020+12
NORM OF RESIDUALS = 654866.55
RESIDUAL STANDARD DEVIATION = 154353.53

UNSCALED COVARIANCE MATRIX

.00000000					
.00000000	.00000000				
.00000000	.00000000	.00000000			
.00000000	.00000000	.00000000	.38422742-08		
.00000000	.00000000	.00000000	-.69468797-10	.12568042-11	
.00000000	.00000000	.00000000	.31310574-12	-.56680776-14	.25577855-16

Fig. 5. Example 28 from DATA5D, using MODE 1 and ISCALE = 0. Without scaling, the rank of the 21 × 6 matrix A is found to be 3.

(28) WAMPLER, J.AMER.STAT.ASSN. 1970, P.549, 5TH DEG. POLYNOMIAL, EQUAL WEIGHTS.

M	N	M1	L	ITYPE	IWGHT	MODE	ISCALE	TOL
21	6	0	1	1	1	1	2	.00000000

FORMAT (F3.0,8X,F8.0)

U = MEAN OF VECTOR A(I,2) = 110.00000
U IS SUBTRACTED FROM EACH ELEMENT OF A(I,2) IN CONNECTION WITH SCALING.

MATRIX A, MATRIX B AND VECTOR OF WEIGHTS

1.0000000	-10.000000	100.00000	-1000.0000	10000.000	-100000.00	760.00000	1.0000
1.0000000	-9.0000000	81.000000	-729.00000	6561.0000	-59049.000	-2042.0000	1.0000
1.0000000	-8.0000000	64.000000	-512.00000	4096.0000	-32768.000	2111.0000	1.0000
1.0000000	-7.0000000	49.000000	-343.00000	2401.0000	-16807.000	-1684.0000	1.0000
1.0000000	-6.0000000	36.000000	-216.00000	1296.0000	-7776.0000	3888.0000	1.0000
1.0000000	-5.0000000	25.000000	-125.00000	625.00000	-3125.0000	1858.0000	1.0000
1.0000000	-4.0000000	16.000000	-64.000000	256.00000	-1024.0000	11379.000	1.0000
1.0000000	-3.0000000	9.0000000	-27.000000	81.000000	-243.00000	17560.000	1.0000
1.0000000	-2.0000000	4.0000000	-8.0000000	16.000000	-32.000000	39287.000	1.0000
1.0000000	-1.0000000	1.0000000	-1.0000000	1.0000000	-1.0000000	64382.000	1.0000
1.0000000	.00000000	.00000000	.00000000	.00000000	.00000000	113159.00	1.0000
1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	175108.00	1.0000
1.0000000	2.0000000	4.0000000	8.0000000	16.000000	32.000000	273291.00	1.0000
1.0000000	3.0000000	9.0000000	27.000000	81.000000	243.00000	400186.00	1.0000
1.0000000	4.0000000	16.000000	64.000000	256.00000	1024.0000	581243.00	1.0000
1.0000000	5.0000000	25.000000	125.00000	625.00000	3125.0000	811568.00	1.0000
1.0000000	6.0000000	36.000000	216.00000	1296.0000	7776.0000	1121004.0	1.0000
1.0000000	7.0000000	49.000000	343.00000	2401.0000	16807.000	1506550.0	1.0000
1.0000000	8.0000000	64.000000	512.00000	4096.0000	32768.000	2002767.0	1.0000
1.0000000	9.0000000	81.000000	729.00000	6561.0000	59049.000	2611612.0	1.0000
1.0000000	10.000000	100.00000	1000.0000	10000.000	100000.00	3369180.0	1.0000

SCALE FACTORS

.21821789	.36037498-01	.44426458-02	.50272143-03	.54598147-04	.57900585-05	.19248482-06

COMPUTED RESULTS

MODE = 1 IFAULT = 0

N1 = COMPUTED RANK OF SYSTEM OF EQUATIONS = 6

COLUMNS OF H = (SQRT(W))*A WERE SELECTED BY THE PIVOTING SCHEME IN THE FOLLOWING ORDER

1 2 5 6 3 4

NUMBER OF ZERO WEIGHTS = 0 DEG. OF FREEDOM = 15

REPORT ON NUMBER OF ESTIMATED NUMBER OF CORRECT

1	CONVERGED	2.	7.5303056

SOLUTION FOR B-VECTOR NO. 1

J	COEFFICIENT(J)	STANDARD DEVIATION OF COEFFICIENT(J)
1	-.99009996+10	.17929694+10
2	.49603022+09	81793223.
3	-9940306.9	1491159.5
4	99601.073	13580.127
5	-499.00033	61.781344
6	1.0000006	.11232484

I	OBSERVED(I)	PREDICTED(I)	RESIDUAL(I)	WEIGHT(I)
1	759.99999	1.0016937	758.99830	1.0000000
2	-2042.0000	5.9954834	-2047.9955	1.0000000
3	2111.0000	63.000946	2047.9990	1.0000000
4	-1684.0000	364.00348	-2048.0034	1.0000000
5	3888.0000	1364.9996	2523.0003	1.0000000
6	1858.0000	3906.0017	-2048.0018	1.0000000
7	11379.000	9330.9982	2048.0016	1.0000000
8	17560.000	19607.998	-2047.9979	1.0000000
9	39287.000	37448.999	1838.0014	1.0000000
10	64382.000	66429.998	-2047.9994	1.0000000
11	113159.00	111111.00	2047.9999	1.0000000
12	175108.00	177156.00	-2048.0021	1.0000000
13	273291.00	271453.00	1837.9978	1.0000000
14	400186.00	402234.00	-2048.0014	1.0000000
15	581242.99	579195.00	2047.9991	1.0000000
16	811567.99	813615.98	-2047.9965	1.0000000
17	1121004.0	1118481.0	2523.0036	1.0000000
18	1506550.0	1508598.0	-2047.9991	1.0000000
19	2002767.0	2000719.0	2047.9958	1.0000000
20	2611612.0	2613660.0	-2048.0016	1.0000000
21	3369180.0	3368421.0	759.00190	1.0000000

SUM OF SQUARED RESIDUALS = 83554260.
NORM OF RESIDUALS = 9140.8019
RESIDUAL STANDARD DEVIATION = 2360.1449

UNSCALED COVARIANCE MATRIX

.57712308+12					
-.26326542+11	.12010396+10				
.47989197+09	-21895005.	399181.93			
-4369464.1	199373.66	-3635.2276	33.107800		
19872.339	-906.83134	16.535917	-.15061385	.68523159-03	
-36.115814	1.6482107	-.30057462-01	.27379578-03	-.12457675-05	.22650318-08

Fig. 6. Example 28 from DATA5D, using MODE 1 and ISCALE = 2. With automatic scaling, the rank of the 21 × 6 matrix A is found to be 6.

U. S. GOVERNMENT PRINTING OFFICE : 1980 311-046/168

NBS-114A (REV. 9-78)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS TN 1126	2. Gov't. Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

Problems Used in Testing the Efficiency and Accuracy of the Modified Gram-Schmidt Least Squares Algorithm

5. Publication Date

August 1980

6. Performing Organization Code

7. AUTHOR(S)

Roy H. Wampler

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, DC 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

Same as item 9.

13. Type of Report & Period Covered

Final

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

In preparing "Algorithm 544: L2A and L2B, Weighted Least Squares Solutions by Modified Gram-Schmidt with Iterative Refinement" for publication in ACM Transactions on Mathematical Software (Vol. 5, 1979), the Fortran computer program was extensively tested. This note describes the various types of problems which were used to explore the efficiency and accuracy of this algorithm. The Fortran subprograms which performed the various tests are listed in an appendix. Also listed are the data used in executing the testing routines as well as typical output from several different types of problems. Among the testing routines is one which is suitable for handling general linear least squares problems. Here, the user has the option of scaling his raw data in order to mitigate the effects of ill-conditioning.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Algorithms; curve fitting; least squares; modified Gram-Schmidt; pseudoinverse; regression; statisics; test problems; test results.

18. AVAILABILITY ☒ Unlimited	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PRINTED PAGES
☐ For Official Distribution. Do Not Release to NTIS	UNCLASSIFIED	83
☒ Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402.	20. SECURITY CLASS (THIS PAGE)	22. Price
☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161	UNCLASSIFIED	$4.00

USCOMM-DC

NBS TECHNICAL PUBLICATIONS

PERIODICALS

URNAL OF RESEARCH—The Journal of Research of the tional Bureau of Standards reports NBS research and developnt in those disciplines of the physical and engineering sciences in ich the Bureau is active. These include physics, chemistry, gineering, mathematics, and computer sciences. Papers cover a ad range of subjects, with major emphasis on measurement thodology and the basic technology underlying standardization. o included from time to time are survey articles on topics sely related to the Bureau's technical and scientific programs. a special service to subscribers each issue contains complete ations to all recent Bureau publications in both NBS and non-S media. Issued six times a year. Annual subscription: domestic ; foreign $16.25. Single copy, $3 domestic; $3.75 foreign.

TE: The Journal was formerly published in two sections: Secn A "Physics and Chemistry" and Section B "Mathematical ences."

MENSIONS/NBS—This monthly magazine is published to inm scientists, engineers, business and industry leaders, teachers, dents, and consumers of the latest advances in science and hnology, with primary emphasis on work at NBS. The magazine hlights and reviews such issues as energy research, fire protecn, building technology, metric conversion, pollution abatement, lth and safety, and consumer product performance. In addin, it reports the results of Bureau programs in measurement ndards and techniques, properties of matter and materials, gineering standards and services, instrumentation, and tomatic data processing. Annual subscription: domestic $11; eign $13.75.

NONPERIODICALS

onographs—Major contributions to the technical literature on rious subjects related to the Bureau's scientific and technical acities.

ndbooks—Recommended codes of engineering and industrial actice (including safety codes) developed in cooperation with inested industries, professional organizations, and regulatory dies.

ecial Publications—Include proceedings of conferences sponred by NBS, NBS annual reports, and other special publications propriate to this grouping such as wall charts, pocket cards, and liographies.

plied Mathematics Series—Mathematical tables, manuals, and dies of special interest to physicists, engineers, chemists, logists, mathematicians, computer programmers, and others gaged in scientific and technical work.

ational Standard Reference Data Series—Provides quantitative ta on the physical and chemical properties of materials, comled from the world's literature and critically evaluated. eveloped under a worldwide program coordinated by NBS under e authority of the National Standard Data Act (Public Law -396).

NOTE: The principal publication outlet for the foregoing data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Services, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services, Springfield, VA 22161, in paper copy or microfiche form.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

he following current-awareness and literature-survey bibliographies e issued periodically by the Bureau:

ryogenic Data Center Current Awareness Service. A literature surey issued biweekly. Annual subscription: domestic $25; foreign 30.

iquefied Natural Gas. A literature survey issued quarterly. Annual bscription: $20.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $30. Please send subscription orders and remittances for the preceding bibliographic services to the National Bureau of Standards, Cryogenic Data Center (736) Boulder, CO 80303.

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-215

SPECIAL FOURTH-CLASS RATE
BOOK