COMPUTER SOLUTION OF THE GENERALIZED LINEAR LEAST SQUARES PROBLEM USING MODIFIED GRAM-SCHMIDT ORTHOGONALIZATION

By

JOANNA CHAMBERLAIN HWANG

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1969

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
July, 1972

COMPUTER SOLUTION OF THE GENERALIZED LINEAR
LEAST SQUARES PROBLEM USING MODIFIED
GRAM-SCHMIDT ORTHOGONALIZATION

Thesis Approved:

John P. Chandler
Thesis Adviser

James R. Van Doren

D. Durham
Dean of the Graduate College

PREFACE

This report describes the use of modified Gram-Schmidt orthogonalization in computer routines that find a basic approximate solution and the least squares solution of minimum Euclidean norm to the system of equations AX=B, where A is an m by n matrix or rank r, X is an n by h matrix, and B is an m by h matrix. A can be treated as if it were of a user-specified rank, k.

The report includes a description of the application of the routines to (a) perform stepwise regression analysis and (b) assess the effect on the solution of decreasing the reliability of the entries in the coefficient matrix.

I am deeply grateful to Dr. John P. Chandler for his advice and support during the preparation of the report. I am also grateful to Dr. Donald Grace for his help in the organization of this paper.

TABLE OF CONTENTS

Chapter		Page
I.	INTRODUCTION	1
	Linear Least Squares Problems in Curve and Surface Fitting	1
II.	THE GENERALIZED LINEAR LEAST SQUARES PROBLEM	6
	Problem Definition	6
	Systems of Linear Equations	6
	Exactly Determined Systems of Equations (m=n)	7
	Underdetermined Systems of Equations (m<n)	7
	Overdetermined Systems of Equations (m>n)	7
	Definition of the Generalized Linear Least Squares Problem	9
	Relation of the Linear Least Squares Problem to the Generalized Inverse of a Matrix	10
	Method	12
	History	15
	Algorithms Using Householder Transformations	15
	Algorithms Using Gram-Schmidt Orthogonalization	16
III.	STEPWISE REGRESSION ANALYSIS	18
IV.	THE USE OF MODIFIED GRAM-SCHMIDT ORTHOGONALIZATION TO SOLVE THE GENERALIZED LINEAR LEAST SQUARES PROBLEM	22
	Osborne's Algorithm	23
	Phase I	23
	Phase II	28
	Phase III	29
	Mathematical Summary of the Algorithm	30
	Modification and Additions	31
	IVOR	31
	Solving the System of Equations for a Pre-specified Rank	33
	Iterative Improvement of the Initial Solutions	34
	Calculation of the Error Matrix	35
	Polynomial Fitting	35

Chapter | Page

V. RESULTS AND CONCLUSIONS 37

Test Problems and Verification 37
Using the Package to Find the Generalized Inverse of an Arbitrary Matrix 37
Using the Package to Find the Solution Vector for an Arbitrary System of Equations 38
Example 1. 38
Example 2 39
Example 3 40
Example 4 43
Using the Package to Perform IVOR 44
Using the Package to Test the Effects of Decreasing the Precision of the Entries in the Coefficient Matrix 46
Comparison of Methods 48
Number of Operations and Storage Requirements . . 49
Summary 51

A SELECTED BIBLIOGRAPHY 54

APPENDIX A - COMPUTER LISTINGS OF LLCR AND LLSQ 56

APPENDIX B - SAMPLE OUTPUT FROM THE PROBLEM CONSISTING OF THE FIRST FIVE COLUMNS OF A 6 BY 6 INVERSE HILBERT MATRIX AND A RIGHT HAND SIDE CHOSEN TO GENERATE THE SOLUTION VECTOR $(1., \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5})$ 70

APPENDIX C - SETTING THE INPUT VARIABLES FOR LLCR AND LLSQ 82

LIST OF TABLES

Table | Page

I. Solution of the System of Equations Consisting of the First Five Columns of a 6 by 6 Inverse Hilbert Matrix and a Right Hand Side Chosen to Generate the Solution Vector $(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5})$ 41

II. Solution of the System of Equations Consisting of the Last Six Columns of an 8 by 8 Inverse Hilbert Matrix and a Right Hand Side Chosen to Generate the Solution Vector $(\frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8})$ 42

III. Solution of the System of Equations Consisting of the First Five Elements of Each of the First Three Rows of a 6 by 6 Inverse Hilbert Matrix with a Right Hand Side (463,-13860,97020) 43

IV. Results of the IVOR Analysis of the Oxygen Solubility Data 45

V. Analysis of Oxygen Solubility Data Using Osborne Pivoting 48

VI. Uses of the Package Consisting of LLCR and LLSQ 52

VII. Setting the Usage Independent FORTRAN Variables Needed by the LLCR Package 83

VIII. Setting the Usage Dependent Variable Needed by the LLCR Package 85

LIST OF FIGURES

Figure | Page

1. Geometrical Representation of the Orthogonalization of 2 Vectors in 2-space 26

2. Setting the Usage Dependent FORTRAN Variables 84

CHAPTER I

INTRODUCTION

This report discusses the use of modified Gram-Schmidt orthogonalization for solving generalized linear least squares problems and performing stepwise regression analysis. Chapter I of this report describes some sources of generalized linear least squares problems. Chapter II presents a general discussion of the generalized linear least squares problem. The closely related problem of stepwise regression analysis is discussed in Chapter III. An algorithm constructed by E. E. Osborne (1) to solve generalized linear least squares problems and some modifications and additions that have been made to Osborne's algorithm are described in Chapter IV. A package consisting of two computer subroutines has been written to implement the algorithm described in Chapter IV. Chapter V describes test cases that were run using the package and the results of these test cases.

Linear Least Squares Problems in Curve and Surface Fitting

One source of a generalized linear least squares problem occurs when trying to find the function,

$$y = f(\underline{t},\underline{c}),$$

that best (to be defined) represents a set of data points,

$$\{y;\underline{t}\}_i,$$

where y is the dependent variable, the $\underline{t}$ are the independent variables, the $\underline{c}$ are the coefficients to be determined (the solution vector), $f(\underline{t},\underline{c})$ is a function from a given family of functions that is linear in $\underline{c}$, and $\{y;\underline{t}\}_i$ is the i-th observation. Note that $f(\underline{t},\underline{c})$ is not necessarily linear in $\underline{t}$.

For example, we may wish to represent the set of data points,

"$(t_i,y_i) = (2,3), (0,1), (1,1)$,"

by a function, $f(\underline{t},\underline{c})$, from some given family of functions, that gives the best fit. $\underline{c}$ is called the vector of parameters for this family. The function that gives the best fit is taken often to be the function for which

$$\sum_{i=1}^{m} (f(\underline{t}_i,\underline{c}) - y_i)^2 \qquad (1.1)$$

is minimized, where m is the number of data points and the general form of $f(\underline{t},\underline{c})$ has been pre-determined. This definition was proposed by Gauss (2) and is the most often used definition of the best fit (3). Finding the vector $\underline{c}$ that minimizes (1.1) is called a linear least squares problem. The discrepancy, or error,

$$f(\underline{t}_i,\underline{c}) - y_i$$

is called the i-th residual.

It will be shown that a necessary and sufficient condition for expression (1.1) to possess minima is that

$$\frac{\partial}{\partial c_j}\left(\sum_{i=1}^{m} (f(\underline{t}_i,\underline{c}) - y_i)^2\right) = 0$$

for $j=1,\ldots,n$, where n is the number of constants, c_j, which are to be determined.

Taking the first partial derivatives with respect to c_j, $j=1,\ldots,n$, we obtain

$$\sum_{i=1}^{m} (f(\underline{t}_i,\underline{c}) - y_i) \frac{\partial f(\underline{t}_i,\underline{c})}{\partial c_j} = 0 \qquad (1.2)$$

for $j=1,\ldots,n$.

If we attempt to fit the above data to a function of the form

$$f(\underline{t},\underline{c}) = c_1+c_2t_1,$$

equations (1.2) generate the system of equations

$$\begin{aligned} 3c_1 + 4c_2 &= 5 \\ 4c_1 + 10c_2 &= 7 \end{aligned} \qquad (1.3)$$

These are called the "normal equations."

Solving the normal equations, we obtain

$$c_1 = \frac{11}{7} \text{ and } c_2 = \frac{1}{14} .$$

An equivalent way of looking at the problem of finding the curve that best fits the data is that we are trying to find the least squares solution of the overdetermined system of equations

$$T\underline{c} = \underline{y}, \text{ where } T = \begin{bmatrix} 1 & 3 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \text{ and } \underline{y} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix},$$

or

$$\begin{aligned} c_1 + 3\cdot c_2 &= 2 \\ c_1 + 0\cdot c_2 &= 1 \\ c_1 + 1\cdot c_2 &= 1 . \end{aligned} \qquad (1.4)$$

Problems of this form can also arise from situations unrelated to curve fitting.

There is no vector $\underline{c}$ such that (1.4) is satisfied exactly since there are more equations than variables and none of the three equations is a restatement (linear superposition) of the other two. However, there is a point (c_1,c_2) such that

$$(c_1 + 3c_2 - 2)^2 + (c_1 - 1)^2 + (c_1 + c_2 - 1)^2$$

is a minimum. That point is (11/7,1/14). This corresponds to finding a vector $\underline{c}$ such that

$$||T\underline{c} - \underline{y}|| \tag{1.5}$$

is a minimum, where $||T\underline{c} - \underline{y}||$ is the length (or "Euclidean norm") of the vector $T\underline{c} - \underline{y}$. The length of the vector $\underline{w}$ is defined to be

$$||\underline{w}|| = \sqrt{\underline{w}^T \cdot \underline{w}} = \sqrt{\sum_{i=1}^{n} w_i^2}$$

and is denoted by $||\underline{w}||$.

The $\underline{c}$ that minimizes (1.5) is the linear least squares solution of equations (1.4).

The above example had only one independent variable. In general. there can be any number of independent variables in the function that is used as the mathematical model for the curve to be fitted. An example with two independent variables is the following.

The distance of penetration of a projectile into a target depends upon the thickness and hardness of the target plate (4). A simple mathematical model might include only the thickness (t_1) and hardness (t_2) and have the linear form

$$\hat{y} = c_1 + c_2 t_1 + c_3 t_2,$$

where $\hat{y}$ is an estimate of the dependent variable y (penetration), and the c_i are the coefficients to be determined.

The need to solve a generalized linear least squares problem occurs almost any time one seeks the solution of an overdetermined system of equations (one which has more equations than variables),

$$A\underline{x}=\underline{b}.$$

Only in exceptional cases can all of the equations be satisfied. We could choose a subset of the equations to be satisfied exactly. The fit of the remaining equations would be disregarded. In the least squares approach, we fit all the equations as closely as possible.

CHAPTER II

THE GENERALIZED LINEAR LEAST SQUARES PROBLEM

Problem Definition

The definition of the generalized linear least squares problem will be given after a short discussion of systems of linear equations.

Systems of Linear Equations

Consider the system of linear equations $A\underline{x}=\underline{b}$ consisting of m equations in n variables. The coefficient matrix, A, is an m by n matrix, where m may be less than, equal to, or greater than n, $\underline{b}$ is an m-component vector (the "right hand side", or vector of constants), and $\underline{x}$ is an n-component vector (the solution vector).

Let the j-th column of a matrix, A, be denoted by A^j. The column rank of the matrix, A, is defined to be the maximum number of linearly independent columns in A (5). A is linearly independent of the other columns of A if there do not exist constants α_i such that

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} \alpha_i A^i = A^j.$$

The row rank of A is the number of linearly independent rows in A (5). The row rank is equal to the column rank (6). The term rank refers to the column rank throughout the remainder of this report. The rank of a matrix is less than or equal to min(m,n). If the rank of the matrix is equal to n, the matrix is said to be of full rank.

Exactly Determined Systems of Equations (m=n)

If A is an m by n matrix of full rank n, the ordinary inverse, A^{-1}, of A, exists, and $\underline{x}=A^{-1}\underline{b}$. The solution vector $\underline{x}$ is unique (5).

Underdetermined Systems of Equations (m<n)

If the system of equations has more variables than equations (m<n), the system of equations is probably consistent. A system of equations is consistent if the rank of the coefficient matrix is equal to the rank of the augmented matrix $(A,\underline{b})$ (5). The rank of the system of equations is less than or equal to m, since any set of m+1 or more m-component vectors is linearly dependent (5). When the number of variables is greater than the number of equations, there is a linear subspace of solutions. Two types of solutions are usually of interest in this case. They are a basic approximate solution that has at most r nonzero components, where r is the rank of the coefficient matrix, and the least squares solution of minimum length (Euclidean norm). These types of solutions will be described in detail in the next section.

Overdetermined Systems of Equations (m>n)

If the matrix, A, is an m by n matrix, where m is greater than n, the system of equations $A\underline{x}=\underline{b}$ is most likely inconsistent since there are more equations than variables. A system of equations is inconsistent if the rank of the coefficient matrix is not equal to the rank of the augmented matrix $(A,\underline{b})$ (5). With m>n, if the system of equations is consistent, then m-n of the equations are restatements of other equations or combinations of other equations in the system; they provide no new information. If the system of equations is consistent, there exists a unique vector $\underline{x}$ such that $A\underline{x}=\underline{b}$.

If the system of equations is inconsistent, there does not exist a vector $\underline{x}$ such that $A\underline{x}=\underline{b}$. In this case, the conventional choice is to find a vector $\underline{x}$ that minimizes the length of the vector $A\underline{x}-\underline{b}$. As mentioned previously, the length (or Euclidean norm) of a vector $\underline{w}$ is defined to be

$$||\underline{w}|| = \sqrt{\underline{w}^T\underline{w}} = \sqrt{\sum_{i=1}^{n} w_i^2}$$

and is denoted by $||\underline{w}||$. The vector $A\underline{x}-\underline{b}$ is called the "residual vector." A vector, $\underline{x}$, that produces the minimum value for the length of the residual vector is called a "least squares solution." The problem of finding a solution vector that produces a residual vector of minimum length is called a "linear least squares problem."

If the rank, r, of A is n, where n is the number of columns in the matrix, the vector that minimizes the length of the residual vector is unique (7). If the rank, r, is less than n, there is a linear subspace (a line or hyperplane) of least squares solutions (7). The solutions can be classified by type. The two types that are usually of interest are the least squares solution of minimum length and a basic approximate solution that has at most r nonzero components, where r is the rank of the coefficient matrix (1). The former is unique (7). The latter is not unique if n>1 (7).

A basic approximate solution is defined as follows:

$\underline{x}_b$ is a basic approximate solution of $A\underline{x}=\underline{b}$ if for all vectors $\underline{x}$,

$$||A\underline{x}-\underline{b}|| \geq ||A\underline{x}_b-\underline{b}||$$

and $\underline{x}$ has at most r nonzero components (8). Let BAS stand for basic approximate solution throughout this report.

A minimum norm solution is defined as follows:

$\underline{x}_0$ is the least squares solution of minimum Euclidean norm if for all vectors $\underline{x}$ either

$$||A\underline{x}-\underline{b}|| > ||A\underline{x}_0-\underline{b}||$$

or else

$$||A\underline{x}-\underline{b}|| = ||A\underline{x}_0-\underline{b}|| \text{ and } ||\underline{x}|| > ||\underline{x}_0||.$$

The second condition holds of $\underline{x}$ is orthogonal to the null space of A; i.e., $\underline{x}$ is orthogonal to every solution of $A\underline{x}=\underline{0}$ (1).

A vector $\underline{u}$ is orthogonal to a vector $\underline{v}$ if

$$\underline{u}^T\underline{v} = \sum_{i=1}^{n} u_i v_i = 0.$$

Definition of the Generalized Linear Least Squares Problem

The problem of finding the solution vector, $\underline{x}$, that minimizes the length of the vector $A\underline{x}-\underline{b}$, where the rank of A is less than or equal to n, is called the "generalized linear least squares problem." The term "generalized linear least squares problem" is used to emphasize that the rank of A may be less than the number of columns in A. In the past, the term "linear least squares problem" was used to denote the problem of finding the vector, $\underline{x}$, that minimized the length of the vector $A\underline{x}-\underline{b}$, where A was of full rank.

As mentioned previously, the solution of the system of equations $A\underline{x}=\underline{b}$, where A is an n by n matrix of rank n, can be obtained by the pre-multiplication of the right hand side by a matrix, A^{-1}, called the inverse of A. The solution of the system of equations $A\underline{x}=\underline{b}$, where A is an m by n matrix of rank r ($r \leq \min(m,n)$), can be represented by the pre-multiplication of the right hand side, $\underline{b}$, by a matrix (to be defined) called the generalized inverse of A. It has been shown further that the generalized inverse of any complex matrix, A (not necessarily square), is unique, and, therefore, the minimum length solution is unique (7).

Relation of the Linear Least Squares Problem to the Generalized Inverse of a Matrix

Penrose (7) has shown that the least squares solution of minimum Euclidean norm is unique and is represented by $\underline{x}=A^{@}\underline{b}$, where $A^{@}$ is called the generalized inverse or pseudo-inverse of A.

$A^{@}$ is defined by the relationships

$$AA^{@}A=A$$
$$A^{@}AA^{@}=A^{@}$$
$$(AA^{@})^{*}=A^{@}A$$

and $(A^{@}A)^{*}=AA^{@}$, where A^{*} is the conjugate transpose of A.

$$A^{*}=\bar{A}^{T}=(\bar{A}_{ji}).$$

Rosen (8) has shown that the BAS (Basic Approximate Solution) can be represented by $\underline{x}=A^{\#}\underline{b}$, where $A^{\#}$ is defined below.

For $r=n\leq m$, $A^{\#}$ is equal to $A^{@}$ and $\underline{x}_b$ is also the least squares solution of minimum Euclidean norm. For $r=n=m$,

$$A^{\#}=A^{@}=A^{-1}.$$

For $r<n$, $A^{\#}$ is not necessarily unique. For this last case, $A^{\#}$ can be defined as follows:

Let A be of rank r ($r<n$). Let B consist of r linearly independent columns of A. Let $\underline{\underline{B}}$ consist of the other n-r columns of A. For simplification, assume that B consists of the first r columns of A so that

$$A=(B,\underline{\underline{B}}).$$

$$B^{@}=(B^{*}B^{-1})B^{*}$$

$$A^{\#}=\begin{pmatrix} B^{@} \\ 0 \end{pmatrix}$$

The first r rows of $A^{\#}$ consist of the matrix $B^{@}$. The remaining n-r rows are zero.

$A^{@}$ can be expressed in terms of $B^{@}$ as follows:

$$A^{@}=C^{*}(CC^{*})^{-1}B^{@}$$

where

$$C=B^{@}A$$

and C^{*} is the conjugate transpose of C (7).

It is not necessary to find $A^{@}$ or $A^{\#}$ explicitly to find the least squares solution of minimum norm or a BAS. Osborne has constructed an algorithm to find these solutions without finding $A^{@}$ or $A^{\#}$. His approach is analogous to the case of solving the system of equations $A\underline{x}=\underline{b}$ by Gaussian elimination when A is an n by n matrix of rank n.

$$\underline{x}=A^{-1}\underline{b}$$

in this case, but A^{-1} did not need to be found explicitly.

Method

The most popular practical method for finding the least squares solution of $A\underline{x}=\underline{b}$ is to solve the normal equations,

$$A^TA\underline{x}=A^T\underline{b}.$$

A derivation of the normal equations and a justification for their use follows.

The vector, $\underline{x}$, that minimizes $||A\underline{x}-\underline{b}||^2$ also minimizes $||A\underline{x}-\underline{b}||$. A necessary condition for $||A\underline{x}-\underline{b}||^2$ to possess a minimum is that

$$\frac{\partial\, ||A\underline{x}-\underline{b}||^2}{\partial\, x_j} = 0 \quad \text{for } j=1,\ldots,n, \qquad (2.1)$$

where n is the number of columns in A. Since $||A\underline{x}-\underline{b}||^2$ is a positive semidefinite quadratic form in $\underline{x}$ and is greater than or equal to zero for all $\underline{x}$, $||A\underline{x}-\underline{b}||^2$ does not contain an inflection point or a maximum in an unrestricted domain (5). Therefore, the $\underline{x}$ for which equations (2.1) are satisfied must be the point where $||A\underline{x}-\underline{b}||^2$ attains its minimum value. As mentioned above, $||A\underline{x}-\underline{b}||^2$ does not contain an inflection point or a maximum in an unrestricted domain, and therefore, it is sufficient to find a vector $\underline{x}$ that satisfies (2.1) to find a minimum to $||A\underline{x}-\underline{b}||^2$.

Since

$$||r||^2 = ||A\underline{x}-\underline{b}||^2 = (A\underline{x}-\underline{b})^T(A\underline{x}-\underline{b})$$

$$= \sum_{k=1}^{m} (b_k - \sum_{i=1}^{n} x_i a_{ki})^2, \qquad (2.2)$$

$$\frac{\partial ||A\underline{x}-\underline{b}||^2}{\partial x_j} = \sum_{k=1}^{m} (b_k - \sum_{i=1}^{n} x_i a_{ki}) a_{kj} = 0, \; j=1,\ldots,n.$$

Equation (2.2) can be rewritten as follows:

$$\sum_{k=1}^{m} b_k a_{kj} - \sum_{k=1}^{m} a_{kj} \sum_{i=1}^{n} a_{ki} x_i = 0 \quad \text{or}$$

$$\sum_{k=1}^{m} a_{kj} \sum_{i=1}^{n} a_{ki} x_i = \sum_{k=1}^{m} b_k a_{kj}, \; j=1,\ldots,n.$$

The above equations are called the normal equations. In matrix notation this is equivalent to

$$A^T A\underline{x} = A^T \underline{b}.$$

$A^T A$ is always symmetric and positive semi-definite (its determinant is nonnegative, as are all its eigenvalues).

Note from (2.2) that the residual vector, r, is orthogonal to every nonzero column of A, since

$$(b_k - \sum_{i=1}^{n} x_i a_{ki}) = r_k,$$

$$\sum_{k=1}^{m} r_k a_{kj} = 0$$

or

$$r^T A^j = 0, \; j=1,\ldots,n.$$

This will be used in the derivation of an alternate method for solving a linear least squares problem. As stated earlier, the most popular method for finding the least squares solution of minimum norm is to solve the normal equations using a method such as Gaussian elimination. There are two problems with using the normal equations to find a least squares solution of minimum norm. First, if A has rank less than n, A^TA has rank less than n. A method such as Gaussian elimination would fail to find a solution. Second, the matrix A^TA is often ill-conditioned (3). A matrix is ill-conditioned if small errors in the entries in the matrix or small errors in the solving process have a large effect on the solution obtained to the problem $A\underline{x}=\underline{b}$ for some $\underline{b}$. The degree of ill-conditioning of a matrix depends on the magnitude of the elements of the inverse of A. A quantity called the condition number is a measure of the ill-conditioning of A. The condition number is equal to $||A||\ ||A^{-1}||$, where

$$||A|| = \max_{||\underline{x}|| = 1} ||A\underline{x}|| \qquad (9).$$

The larger the condition number the greater the ill-conditioning (3). The smallest possible condition number is one. If the condition number of A is cond (A), the condition number of A^TA is $\text{cond}^2(A)$.

Longley (10) and Wampler (11) have done comparative studies of methods used to solve the generalized linear least squares problem. Both of them have shown examples where solving the normal equations has produced a solution vector with almost no correct digits.

Since the normal equations cannot easily be used to find the least squares solution when the coefficient matrix has a rank less than the number of columns in the matrix, and should not be used when A^TA is ill-conditioned, a better method is needed. In Chapter IV a description of an algorithm developed by E. E. Osborne is presented.

A brief history of some of the methods that have been developed to find the solution to the generalized linear least squares problem is given below.

History

As mentioned previously, if the coefficient matrix is of full rank, then the most popular method for finding the least squares solution is to solve the normal equations. If the system is ill-conditioned, solving the normal equations can produce a solution vector that is very inaccurate (10).

Orthogonalization techniques are the second most popular class of methods for solving the generalized linear least squares problem. Householder transformations or a form of the Gram-Schmidt method are used normally to do the orthogonalization (3).

Algorithms Using Householder Transformations

E. E. Osborne (12) first proposed using Householder transformations to do orthogonalization in 1961. The method he developed was primarily for the homogeneous case $A\underline{x}=\underline{0}$. His intent was to improve the accuracy of the solution he obtained. In 1965, Businger and Golub (13) proposed using Householder transformations for solving the nonhomogeneous case $A\underline{x}=\underline{b}$, where A is of full rank. In 1965, Golub (14) allowed the imposition of linear equalities (a subset of equations that must be satisfied exactly). In 1967, Björck and Golub (15) added iterative improvement of the solution to the algorithm proposed by Businger and Golub. In 1969, Hanson and Lawson (16) extended the Businger-Golub algorithm to solve systems of equations of the form $\underline{A}\underline{x}=\underline{b}$, where A is of rank r (r≤n).

Algorithms Using Gram-Schmidt Orthogonalization

In 1964, Bauer (17) published an algorithm using modified Gram-Schmidt orthogonalization to solve the system of equations $\underline{A}\underline{x}=\underline{b}$. This method was good for matrices of full rank only. In 1965, Osborne (1) extended the use of modified Gram-Schmidt otrhogonalization to the case where the coefficient matrix was of rank r (r≤n). In 1968, Björck (18) combined iterative improvement of the solution with the use of modified Gram-Schmidt orthogonalization to reduce the error in the solution of the system of equations $\underline{A}\underline{x}=\underline{b}$, where the rank of A is r (r≤n). Björck (18) has shown that modified Gram-Schmidt orthogonalization produces a somewhat more accurate solution vector than the use of Householder transformations for orthogonalization.

Programs implementing Björck's algorithm (18) and Bauer's algorithm (17) are available at Oklahoma State University, Stillwater, Oklahoma. The package consisting of the FORTRAN subroutines, LLCR and LLSQ, has been compared with the programs implementing Björck's and Bauer's algorithms. The LLCR package produced results that were as accurate or more accurate than the routines of Björck and Bauer. Björck's routine does allow the imposition of linear equalities. In practice, this option is not usually used and hence was omitted. The imposition of linear equalities can be approximated by multiplying those rows of A and components of $\underline{b}$ by a large weighting factor before using the package. In addition, the user of the package consisting of LLCR and

LLSQ has many options available that are not available to the user of the other routines.

CHAPTER III

STEPWISE REGRESSION ANALYSIS

Stepwise regression analysis is closely related to the generalized linear least squares problem described in Chapter II.

In stepwise regression analysis a curve is fitted to a set of data points,

$$\{y_i;\ t_1,\ \ldots,\ t_n\}_i,\ i=1,\ldots,m, \text{ where}$$

$\{y_i;\ t_1,\ldots,t_n\}_i$ is the i-th observation (19). The mathematical model for the curve is called the regression equation and has the form

$$\hat{y}_i = c_0 + c_1 t_{i1} + c_2 t_{i2} + \ldots + c_n t_{in}$$

where $\hat{y}$ is an estimate of the dependent variable, y, the t_j, j=1,...,n, are the independent variables, and the c_i,i=0,...,n, are the coefficients to be determined. The t_j,j=1,...,n, can represent functions of the form

$$t_j = g_j(\underline{z}_j),$$

where the functions, $\underline{g}(\underline{z}_j)$, do not contain the dependent variable and where the $\underline{z}_j$ are variables whose observed numerical values completely determine the numerical value of the t_j (19).

Stepwise regression is used when it is desired to represent the dependent variable in terms of as few of the independent variables as possible. When the dependent variables are highly correlated, the simple regression model may be considerably simplified by eliminating some of the variables. In the stepwise procedure one variable is added to the mathematical model at a time (19). Thus, the intermediate equations

$$\hat{y} = c_0$$

$$\hat{y} = c_0' + c_1' t_{i_1}$$

$$\hat{y} = c_0'' + c_1'' t_{i_1} + c_2'' t_{i_2}$$

$$\vdots$$

are obtained. Note the i_1 is not necessarily equal to 1, i_2 is not necessarily equal to 2, etc.

An important property of the stepwise procedure is based on the fact that a variable may be significant at an early stage but may become insignificant after several other variables are entered in the equation. A variable that is not highly correlated with the other variables in the regression equation at an early stage may be highly correlated with variables that enter the regression equation later, thereby reducing its significance. The stepwise procedure permits the insignificant variable (highly correlated variable) to be removed from the regression equation. The test to decide if any variable is to leave or enter the regression equation is a statistical test, namely the F-test. The F-test measures the degree of linear correlation among variables in the regression equation (20). If a variable is too highly correlated with the other variables in the regression equation, it will be removed or not allowed to enter.

The decision as to which variable is to enter the regression equation is a numerical decision. The variable added to the regression equation at each step is the one that makes the greatest improvement in the fit of the curve as measured by the length of the residual vector; i.e., it is the one that produces the shortest residual vector. At each stage of the stepwise procedure, the least squares solution is found for the variables entered in the regression equation at that point (19).

Stepwise regression does not necessarily produce the solution vector with the residual vector of minimum length. All that is assured is that given k variables in the regression equation, the next variable to enter the equation is the variable whose addition to the model produces the solution vector for which the length of the residual vector is minimized.

Some packages that are called stepwise regression packages do not have the ability to delete variables from the regression equation. Stepwise regression without the deletion of variables is called IVOR (Independent Variable Ordering by Regression Sum of Squares)(4) or "forward selection" (19). Some packages that include only forward selection are the IBM 360 Scientific Subroutine Package (21), the Bio-Medical (BMD) stepwise regression programs (22), and the package that implements the methods described in the next chapter.

Deletion of variables from the regression equation was not implemented because of the following reasons.

First, there is no standard statistical test that best calculates the linear correlation among variables in the regression equation for all cases. The F-test assumes that the standard deviations of all the variables are equal. If the standard deviations are not all equal, the F-test may not give an accurate calculation of the linear correlation among the variables.

Second, when a variable is deleted from the regression equation, the system of equations must be returned to the state in which it would have been if the variable had never entered the regression equation. When orthogonalization is used to do stepwise regression, this state must be constructed. The construction of this state can be inaccurate. The LLCR package was written to provide an accurate means to solve generalized linear least squares problems and to perform stepwise regression (IVOR).

Third, cycling may occur when variables are deleted from the equation. A group of variables may alternately enter and leave the regression equation. For example, variable t_{i_1} may enter the regression equation followed by t_{i_2}'s entry. t_{i_1} may be deleted from the regression equation followed by t_{i_2}'s deletion from the regression equation. t_{i_1} may reenter the regression equation followed by t_{i_2}'s reentry into the regression equation. This pattern may continue until something extra-ordinary happens to stop the process such as exceeding the time limit on the job.

An attempt will be made to implement deletion of variables from the regression equation in the future.

CHAPTER IV

THE USE OF MODIFIED GRAM-SCHMIDT ORTHOGONALIZATION TO SOLVE THE GENERALIZED LINEAR LEAST SQUARES PROBLEM

The generalized linear least squares problem consists of finding the solution vector $\underline{x}$ to the system of equations, $A\underline{x}=\underline{b}$, that minimizes the length of the residual vector $A\underline{x}-\underline{b}$.

E. E. Osborne (1) has constructed a method for solving the generalized linear least squares problem based on the fact that (a) the residual vector for a linear least squares solution is orthogonal to every nonzero column of A and (b) the least squares solution of minimum norm is orthogonal to the null space of A; i.e., orthogonal to every solution of $A\underline{x}=\underline{0}$.

Osborne's algorithm consists of three phases. During the first phase of the algorithm, the numerical rank of the system of equations is found and a decomposition of the coefficient matrix into the product of an orthogonal matrix and a permuted unit upper triangular matrix is determined. During the second phase, a BAS (Basic Approximate Solution) is found. During the third phase, the minimum norm solution is found. Before the three phases of the algorithm are discussed, a definition of numerical rank will be given.

As mentioned in Chapter II, the rank of a matrix is equal to the number of linearly independent columns in A. A^j, the j-th column of A, is linearly dependent on the other columns of A if there exist constants α_i such that

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} \alpha_i A^i = A^j. \qquad (4.1)$$

Osborne's algorithm considers a column, A^j, to be linearly dependent on other columns of A if there exist constants α_j such that

$$\frac{\left|\left|A^j - \sum\limits_{\substack{i=1 \\ i \neq j}} \alpha_i A^i\right|\right|}{\left|\left|A^j\right|\right|} < \varepsilon$$

where ε is set by the user of the algorithm, as a measure of the relative error he will tolerate. In practice, ε is $\geq \delta$, where δ is the smallest number such that

$$1. + \delta > 1.$$

in single precision real arithmetic on the computer being used. For example, on the IBM 360/65. $\delta \approx 9.6 \times 10^{-7}$.

The numerical rank of A is the number of linearly independent columns in A, where the definition of linear dependency is the numerical one given in (4.1).

Osborne's Algorithm

Phase I

Phase I of Osborne's algorithm consists largely of elementary column operations performed on the matrix,

$$\begin{pmatrix} A \\ R \end{pmatrix}$$

where R is an n by n identity matrix, that produces a decomposition of the form $A = A_N R_N^{-1}$ and determines the numerical rank of A. If the numerical rank of A if r*, r* columns of A_N will be made mutually orthogonal using modified Gram-Schmidt orthogonalization. A description of R_N is given later in this section.

The transformation of

$$\begin{pmatrix} A \\ R \end{pmatrix}$$

into the matrix

$$\begin{pmatrix} A_N \\ R_N \end{pmatrix}$$

by modified Gram-Schmidt orthogonalization will be described now.

In modified Gram-Schmidt orthogonalization of a matrix of full rank, the second column is orthogonalized with respect to the first column, the third column is orthogonalized with respect to the first and second columns, ..., the n-th column is orthogonalized with respect to all the other columns of A, where n is the number of columns in A.

If the matrix has a numerical rank less than the number of columns, the lengths of some of the columns will become $\leq \varepsilon$ during the orthogonalization process (1). No attempt should be made to orthogonalize these columns with respect to the other columns of the coefficient matrix.

In order to keep track of the columns that remain to be orthogonalized, if any, Osborne reordered the columns of the partially orthogonalized coefficient matrix so that the first k columns of the modified A matrix contain the k columns that have been made mutually orthogonal, for k=1,...,r*= numerical rank of A. A vector,

$$\rho = (||A^1||^2, ||A^2||^2, \ldots, ||A^n||^2),$$

also is set up at the beginning of the algorithm. Whenever columns of the modified matrix,

$$\begin{pmatrix} A \\ R \end{pmatrix}$$

are interchanged, corresponding components of ρ are interchanged.

The k-th step of the modified Gram-Schmidt orthogonalization procedure is described below.

For k=1,...,r*= the numerical rank of A, the quantities

$$\begin{aligned} d_k &= ||A_{k-1}^k||^2 \\ \alpha_{kj} &= (A_{k-1}^k \cdot A_{k-1}^j)/d_k \\ A_k^j &= A_{k-1}^j - \alpha_{kj}A_{k-1}^k \\ R_k^j &= R_{k-1}^j - \alpha_{kj}R_{k-1}^k, \; k+1 \le j \le n, \end{aligned}$$

where $A_0 = A$ and $R_0 = I$, are calculated.

A vector representation for the orthogonalization of two vectors in 2-space is shown in Figure 1 (23). The orthogonal projection of β on α is made. The orthogonalized vectors are β' and α, where $\beta' = \beta - \alpha$. β' is orthogonal to α.

Let

$$\begin{pmatrix} A_K \\ R_k \end{pmatrix}$$

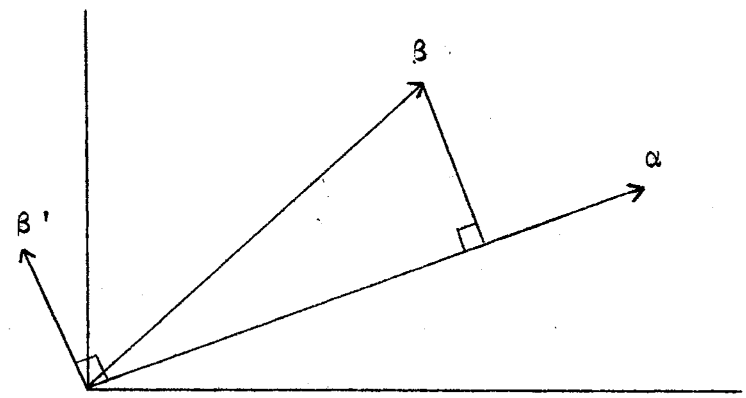

Figure 1. Geometrical Representation of the Orthogonalization of 2 vectors in 2-space.

be used to designate the state of the matrix

$$\begin{pmatrix} A \\ R \end{pmatrix}$$

after each step k, k=1,..., numerical rank of A, of the algorithm. At this point (1), k columns have been made mutually orthogonal and

$$A = A_k R_k^{-1}$$

or

$$A_k = AR_k.$$

The quantities

$$t(j) = ||A_k^j||^2/\rho(j)$$

are calculated for $j = k+1,\ldots,n$, where A^j is the j-th column of A.

If

$$t(j) \le \varepsilon \qquad \text{for } j=k+1,\ldots,n,$$

the numerical rank of A is k and $A_N = A_k$. The numerical rank is the first k for which

$$t(j) \le \varepsilon \qquad \text{for } j=k+1,\ldots,n.$$

If

$$t(j) > \varepsilon \qquad \text{for any } j,\ j=k+1,\ldots,n,$$

the j for which t(j) is the maximum is found. Column j of

$$\begin{pmatrix} A_k \\ R_k \end{pmatrix}$$

is interchanged with column k+1 of

$$\begin{pmatrix} A_k \\ R_k \end{pmatrix}.$$

The j-th component of ρ is interchanged with the (k+1)-st component of ρ. The selection of the column to become the (k+1)-st column of A_{k+1} is called Osborne pivoting throughout the remainder of the report.

Once the numerical rank, r*, of the matrix is determined, the n-r* vectors that have a length $\le \varepsilon$ are considered to be zero vectors. The last n-r* columns of R_{r*} are made mutually orthogonal. The operations described above produce the matrix

$$\begin{pmatrix} A_N \\ R_N \end{pmatrix}$$

R_N has the following properties (1):

(i) det $R_N = \pm 1$,

(ii) $A = A_N R_N$ or $A_N = AR_N^{-1}$,

(iii) R_N is obtainable by permuting rows of an upper triangular matrix all of whose diagonal elements are unity.

(iv) The vectors R_N^{r*+1}, R_N^{r*+2}, ... , R_N^n form an orthogonal basis for the null space of A (1).

Phase II

The basic approximate solution is found during phase II. The procedure to find the basic approximate solution is based on the fact that the residual vector for a linear least squares solution is orthogonal to every nonzero column of the coefficient matrix. The development of a method to find the basic approximate solution will be given now.

If the vector

$$\begin{pmatrix} -\underline{b} \\ \underline{0} \end{pmatrix}$$

is appended to the matrix

$$\begin{pmatrix} AR_N \\ R_N \end{pmatrix},$$

the matrix

$$\begin{pmatrix} AR_N & -\underline{b} \\ R_N & \underline{0} \end{pmatrix}$$

results. This matrix is post-multiplied by the (n+1) by (n+1) matrix

$$\begin{pmatrix} I & \underline{u} \\ 0 & 1 \end{pmatrix}$$

which will orthogonalize $-\underline{b}$ with respect to the first r* columns of AR_N. The matrix

$$\begin{pmatrix} AR_N & AR_N\underline{u}-\underline{b} \\ R_N & R_N\underline{u} \end{pmatrix}$$

results. Since R_N is nonsingular and the residual vector $AR_N\underline{u}-\underline{b}$ is orthogonal to every nonzero column of AR_N, $R_N\underline{u}$ is a least squares solution of $A\underline{x}=\underline{b}$. According to Rosen's definition (8), it is a basic approximate solution. This follows from properties (i) and (iii) above and from the fact that $R_N\underline{u}$ is a linear combination of the first r* columns of R_N. Therefore, $R_N\underline{u}$ has at most r* nonzero components (1).

Phase III

In phase III, the minimum length solution is found by computing a least squares solution that is orthogonal to the null space of A; i.e., orthogonal to every solution of $A\underline{x}=\underline{0}$.

The following discussion shows how the minimum length solution is found from the basic approximate solution.

If the matrix

$$\begin{pmatrix} AR_N & AR_N\underline{u}-\underline{b} \\ R_N & R_N\underline{u} \end{pmatrix}$$

is post-multiplied by the (n+1) by (n+1) matrix

$$\begin{pmatrix} I & \underline{v} \\ 0 & 1 \end{pmatrix}$$

which will orthogonalize $R_N\underline{u}$ with respect to the last n-r* columns of R_N, where r* is the numerical rank of A, the matrix

$$\begin{pmatrix} AR_N & AR_N(\underline{v}+\underline{u})-\underline{b} \\ R_N & R_N(\underline{v}+\underline{u}) \end{pmatrix}$$

is obtained. The first r* components of $\underline{v}$ are zero and the last n-r* columns of A are considered to be zero vectors. Therefore,

$$AR_N\ \underline{v}=0$$

and $AR_N(\underline{v}+\underline{u})$ is orthogonal to the nonzero columns of AR_N. Thus, $R_N(\underline{v}+\underline{u})$ is a least squares solution of $A\underline{x}=\underline{b}$. $R_N(\underline{v}+\underline{u})$ is orthogonal to the null space of A, and, therefore, is the unique least squares solution of minimum length (1).

Mathematical Summary of the Algorithm

The complete algorithm can be described mathematically as follows:

$$A_N^1 = A^1$$

$$R_N^1 = R^1$$

$$R_N^k = R^k - \sum_{j=1}^{k-1} \left(\frac{A_N^j \cdot A^k}{A_N^j \cdot A_N^j} \right) R^j$$

$$A_N^k = A^k - \sum_{j=1}^{k-1} \left(\frac{A_N^j \cdot A^k}{A_N^j \cdot A_N^j} \right) A^j \quad \text{for } k=2,\ldots,r^*,$$

where r* is the numerical rank of A.

$$R_N\underline{u} = -\sum_{j=1}^{r*} \left(\frac{A_N^{j} \cdot (-\underline{b})}{A_N^j \cdot A_N^j} \right) R_N^j$$

$$R_N(\underline{u}+\underline{v}) = R_N\underline{u} - \sum_{j=r*+1}^{n} \left(\frac{R_N^j \cdot (R_N\underline{u})}{R_N^j \cdot R_N^j} \right) R_N^j.$$

The routines that have been implemented to solve the generalized linear least squares problem employ the algorithm constructed by Osborne. Certain modifications in Osborne's method have been made and several additional features have been added.

Modification and Additions

The major addition to Osborne's algorithm was the ability to do IVOR (Independent Variable Ordering by Regression Sum of Squares)--stepwise regression without the deletion of variables from the regression equation. In addition, the coefficient matrix can be treated as if it had a pre-specified rank, the initial BAS and minimum length solutions can be iteratively refined, and the error matrix, $(A^TA)^{-1}$, is calculated for matrices of full rank.

IVOR

Earlier in the chapter it was stated that after r* steps of the algorithm constructed by Osborne, r* columns of the coefficient matrix are mutually orthogonal. After k steps of the algorithm, k columns of the coefficient matrix are mutually orthogonal. Let the state of the coefficient matrix be designated by A_k.

$$A_k = AR_k \qquad \text{or } A = A_kR_k^{-1}.$$

If

$$\begin{pmatrix} -\underline{b} \\ \underline{0} \end{pmatrix}$$

is appended to

$$\begin{pmatrix} AR_k \\ R_k \end{pmatrix}$$

the matrix

$$\begin{pmatrix} AR_k & -\underline{b} \\ R_k & \underline{0} \end{pmatrix}$$

results. If the resulting matrix is post-multiplied by the (n+1) by (n+1) matrix ,

$$\begin{pmatrix} I & \underline{u} \\ 0 & 1 \end{pmatrix},$$

that will orthogonalize $-\underline{b}$ with respect to the first k columns of AR_k, the matrix

$$\begin{pmatrix} AR_k & AR_k\underline{u}-\underline{b} \\ R_k & R_k\underline{u} \end{pmatrix}$$

results. $R_k\underline{u}$ has k nonzero components since det $R_k = \pm 1$, R_k is obtainable by permuting the rows of an unit upper triangular matrix, and $R_k\underline{u}$ is a linear combination of the first k columns of R_k. The k nonzero components of $R_k\underline{u}$ are the regression coefficients for the k variables that have entered the regression equation. The (k+1)-st variable to enter the regression equation is found as follows:

For each variable not in the regression equation, we predict the length of the residual vector that would be obtained if the variable were entered in the regression equation. The length of the residual vector can be predicted by calculating

$$||(AR_k\underline{u}-\underline{b}) - \left(\frac{A_k^t \cdot (AR_k\underline{u}-\underline{b})}{A_k^t \cdot a_k^t}\right) A_k^t||$$

for each variable t that is not in the regression equation. The variable that will produce the residual vector of the shortest length is the variable to enter the regression equation.

Solving the System of Equations for a Pre-specified Rank

The coefficient matrix can be treated as if it were of a pre-specified rank, k. If the numerical rank is less than k, the minimum norm solution is found; otherwise k columns of the coefficient matrix are made mutually orthogonal. Osborne's method (1) of column selection is used to choose those k columns. The remaining columns of the coefficient matrix are treated as if they are zero vectors. The last n-k columns of R_k then are made mutually orthogonal, where n is the number of columns in the coefficient matrix. Orthogonalizing only k columns of the coefficient matrix when the rank of the coefficient matrix is not less than k corresponds to increasing the value of ε until the numerical rank of the coefficient matrix is equal to k. This might be used on an accurate computer such as the CDC 6600 to predict the solution that could be found on a less accurate computer such as the IBM 360.

ε is the value used to determine if a column is a linear combination of other columns in the coefficient matrix. The system of equations can be solved for several ranks during one run of the implemented routines. This corresponds to solving the system of equations for a range of ε.

Iterative Improvement of the Initial Solutions

Roundoff error in the calculation of a solution vector often makes the solution vector inaccurate. If $\underline{x}_c$ is the calculated answer and $\underline{x}_t$ is the true answer,

$$\underline{x}_c + \Delta\underline{x} = \underline{x}_t$$

and

$$A\underline{x}_t = \underline{b} \quad \text{or} \quad A(\underline{x} + \Delta\underline{x}) = \underline{b}$$

or

$$A\Delta\underline{x} = \underline{b} - A\underline{x}_c$$

Iterative improvement of the initial BAS and minimum length solutions has been implemented to improve their accuracy. The interative improvement procedure is described as follows (1):

(i) Let $\underline{x}_i$ be the initial solution.

(ii) Calculate the vector $\underline{r}=\underline{b}-A\underline{x}_i$ in double precision.

(iii) Solve the system of equations

$$A\Delta\underline{x}_i = \underline{r}_i.$$

(v) If $$||\underline{x}_{i+1} - \underline{x}_i||/||\underline{x}_{i+1}|| \leq \varepsilon_1$$

where ε_1 is greater than or equal to δ , $\underline{x}_{i+1}$ is accepted as the solution to $A\underline{x}=\underline{b}$.

δ is the smallest floating point number such that $1. + \delta > 1.$ in the computer.

Calculation of the Error Matrix

The matrix R_N generated by the orthogonalization process described above can be used to obtain the error matrix, $(A^TA)^{-1}$, if A is of full rank.

The derivation of the error matrix from R follows.

$$A = A_N R_N^{-1}$$

$$\begin{aligned}(A^TA)^{-1} &= [(A_N R_N^{-1})^T (A_N R_N^{-1})]^{-1} \\ &= [(R_N^{-1})^T A_N^T A_N R_N^{-1}]^{-1} \\ &= [(R_N^{-1})^T D R_N^{-1}]^{-1} \\ &= R_N D^{-1} ((R_N^{-1})^T)^{-1} \\ &= R_N D^{-1} ((R_N^{-1})^{-1})^T \\ &= R_N D_N^{-1} R_N^T \qquad \text{D is a diagonal matrix.}\end{aligned}$$

Polynomial Fitting

The LLCR package can be used to fit a polynomial to a set of data points,

$$\{y;t\}_i$$

where y is the dependent variable, t is the independent variable, and

$\{y;t\}_i$ is the i-th observation.

The mathematical model for the curve would have the form

$$\hat{y} = c_0 + c_1 t + c_2 t^2 + \ldots + c_n t^n,$$

where $\hat{y}$ is an estimate of y and the c_j, $j=0,\ldots,n$, are the coefficients to be determined. When the package is used to fit a polynomial to a set of data points, one variable is entered into the mathematical model at a time. The variables are entered in the following order:

$$t,\ t^2,\ t^3,\ \ldots,\ t^n.$$

Entering the variables in the above order is called sequential selection. When the variables are entered sequentially, the intermediate equations

$$\begin{aligned} y &= c_0 \\ y &= c_0' + c_1' t \\ y &= c_0'' + c_1'' t + c_2'' t^2 \\ y &= c_0''' + c_1''' t + c_2''' t^2 + c_3''' t^3 \\ &\vdots \end{aligned}$$

are obtained. The user can decide if he wishes to represent the data by a polynomial of a lesser degree.

CHAPTER V

RESULTS AND CONCLUSIONS

Test Problems and Verification

A package consisting of the routines LLCR and LLSQ has been written in Standard FORTRAN (24) to implement Osborne's method (1) for solving the generalized linear least squares problem. Since the routines can solve a system of equations for multiple right hand sides during one run of the program, the generalized inverse of an arbitrary matrix can be found accurately and efficiently. In addition, the user of the package can perform IVOR (Independent Variable Ordering by Regression Sum of Squares)--stepwise regression without the deletion of variables from the regression equation. The user also can study efficiently the effects on the solution vector of decreasing the reliability of the entries in the coefficient matrix. The error matrix, $(A^TA)^{-1}$, is calculated for systems where the coefficient matrix is of full rank.

Each of the above uses has been tested on the IBM 360/65 at Oklahoma State University, Stillwater, Oklahoma. The results are listed below.

Using the Package to Find the Generalized Inverse of an Arbitrary Matrix

The generalized inverse of an arbitrary matrix, A, can be found by solving the set of equations AX=I, where A is an m by n matrix, X (the generalized inverse of A) is an n by m matrix, and I is an m by m identity matrix.

The generalized inverse of a 6 by 4 zero matrix was found exactly in one iteration. Many routines for solving an arbitrary system of equations will not handle the case where the coefficient matrix has a rank of zero.

The generalized inverse of the matrix

$$\begin{bmatrix} 1. & -1. & 3. & 1. \\ 2. & 4. & 5. & 1. \\ -1. & 2. & -1. & 1. \\ 4. & 1. & 9. & 1. \end{bmatrix}$$

was found to full single precision accuracy without iterating the solution. This example was taken from Rosen (8). The generalized inverse was found to be

$$\begin{bmatrix} -.21153 & .04487 & -.22435 & .05769 \\ -.19230 & .19230 & .03846 & -.03846 \\ .08653 & -.00320 & .01602 & .06730 \\ .50961 & -.09294 & .46474 & -.04807 \end{bmatrix}.$$

<u>Using the Package to Find the Solution Vector for an Arbitrary System of Equations</u>

<u>Example 1</u>. The first example was taken from Rosen's article (8). The system consisted of

$$\begin{bmatrix} 1. & -1. & 3. & 1. \\ 2. & 4. & 5. & 1. \\ -1. & 2. & -1. & 1. \\ 4. & 1. & 9. & 1. \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 1. \\ 3. \\ 2.5 \\ 2.5 \end{bmatrix}.$$

Using an ε of $.16 \times 10^{-5}$, the rank of the system was found to be three. Full single precision accuracy was obtained without iterating the solution. The lengths of the residual vector for the basic approximate solution (BAS) and the minimum length solution were both .5. The lengths of the BAS vector and the minimum length solution vector were 1.607 and 1.451, respectively. The BAS vector was

$$\begin{bmatrix} .07692 \\ .38462 \\ 0 \\ 1.5577 \end{bmatrix}.$$

The minimum length solution vector was

$$\begin{bmatrix} -.49359 \\ .38462 \\ .28526 \\ 1.27240 \end{bmatrix}.$$

Example 2. The second system of equations that was used to test the package had a coefficient matrix consisting of the first five columns of a 6 by 6 inverse Hilbert matrix and a right hand side chosen to generate a solution vector of $(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5})$. The matrix,

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{m-1} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{m} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{m+1} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{m-1} & \frac{1}{m} & \frac{1}{m+1} & \cdots & \frac{1}{2m-1} \end{bmatrix}$$

is the Hilbert matrix of order m.

The inverse of the Hilbert matrix was used because each of the entries in it is an integer and can be represented in a computer exactly if the precision of the computer is large enough (3). Therefore, the effect of roundoff error on the solution vector can be studied. This system of equations is fairly ill-conditioned, getting worse with larger m.

Full single precision accuracy was achieved when the solution was iterated. The results of the run are shown in Table I.

The implementation of Björck's routine (18) required five iterations to obtain this accuracy. Only three iterations were required with LLCR and LLSQ.

Example 3. The third test case consisted of the last six columns of an 8 by 8 inverse Hilbert matrix with a right hand side chosen to produce the solution vector $(\frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8})$.

This system is extremely ill-conditioned. As mentioned previously, when a system is ill-conditioned, small errors in the entries in the input coefficient matrix or in the solution process cause a large

TABLE I

SOLUTION OF THE SYSTEM OF EQUATIONS CONSISTING OF THE FIRST FIVE COLUMNS OF A 6 BY 6 INVERSE HILBERT MATRIX AND A RIGHT HAND SIDE CHOSEN TO GENERATE THE SOLUTION VECTOR $(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5})$

Iteration Number	Solution Vector	Length of the Residual Vector
0	.9558830	1.043110
	.4843262	
	.3263453	
	.2468576	
	.1988596	
1	.9997082	.4102879
	.4999225	
	.3333055	
	.2499895	
	.1999996	
2	.9999999	.1591172
	.4999998	
	.3333333	
	.2499999	
	.2000000	
3	.9999999	.2431152
	.4999999	
	.3333333	
	.2499999	
	.2000000	

change in the solution vector (3). The condition number of a matrix is a measure of the ill-conditioning of the system. The smallest possible condition number is unity. The system in the present example has a condition number of 10^8.

Using an ε of $.16 \times 10^{-5}$ and doing all calculations in single precision, the numerical rank was determined to be four. Full single precision accuracy was achieved after iteration of the solution. The rank was not determined to be six as there was considerable truncation error in forming inner products due to the low precision of the IBM 360.

Table II contains the results of the run.

TABLE II

SOLUTION OF THE SYSTEM OF EQUATIONS CONSISTING OF THE LAST SIX COLUMNS OF AN 8 BY 8 INVERSE HILBERT MATRIX AND A RIGHT HAND SIDE CHOSEN TO GENERATE THE SOLUTION VECTOR

$$(\frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8})$$

Iteration Number	Minimum Length Solution Vector	Length of the Residual Vector for the Minimum Length Solution	Basic Approximate Solution Vector	Length of the Residual Vector for the BAS
0	-.09663224	155.1612	-.1880930	97.75385
	.03279249		-.6149425	
	.05031452		-.00001407021	
	.02720612		0	
	-.008033112		0	
	-.04418335		-.005662531	
1	-.09643781	135.0647		
	.03278171			
	.05018743			
	.02709594			
	-.008037787			
	-.04403742			
2	-.09643847	133.9126		
	.03278175			
	.05018786			
	.02709632			
	-.008037772			
	-.04403792			
3	-.09643847	132.2448		
	.03278176			
	.05018786			
	.02709632			
	-.00837772			
	-.04403792			

Björck's routine failed to find a solution for this example. It must be emphasized again that obtaining a numerical rank of four was not a failure of the routines but was caused by the low precision of the computer on which the test case was run.

Example 4. The fourth test case consisted of the first five elements of each of the first three rows of a 6 by 6 inverse Hilbert matrix with the right hand side (463,-13860,97020). Both the basic approximate solution vector and the minimum norm solution vector are of interest since the number of equations is less than the number of variables.

Table III contains the results of the run.

TABLE III

SOLUTION OF THE SYSTEM OF EQUATIONS CONSISTING OF THE FIRST FIVE ELEMENTS OF EACH OF THE FIRST THREE ROWS OF A 6 BY 6 INVERSE HILBERT MATRIX WITH A RIGHT HAND SIDE (463,-13860,97020)

Iteration Number	Minimum Norm Solution Vector	Square of the Length of the Residual Vector	Basic Approximate Solution Vector
0	.02615530 -.08060956 -.002280064 .07264209 .1280568	.009011976	1.583456 .2777886 0 0 .07685214
1	.02614972 -.08058983 -.002287482 .07262659 .1280463	.0009218131	
2	.02614973 -.08058983 -.002287471 .07262659 .1280463	.0005667009	
3	.02614974 -.08058983 -.002287467 .07262659 .1280463	.0004688033	

Using the Package to Perform IVOR

The package consisting of LLCR and LLSQ was used to perform IVOR on oxygen solubility data. The mathematical model for this curve is defined below.

Let z_1 = absolute temperature,

z_2 = salinity of seawater,

y = log of the solubility of oxygen in sea water,

and the model be described by

$$\hat{y}_i = (a_{i1}+a_{i2}/z_{i1}+a_{i3}\ln(z_{i1})+a_{i4}z_{i1}+a_{i5}z_{i1}^2+\ldots) \cdot (b_{i1}+b_{i2}z_{i2}+\ldots)$$

so that

$t_{i1} \equiv 1$,	$c_1 \equiv a_1b_1$
$t_{i2} \equiv z_{i2}$,	$c_2 \equiv a_2b_2$
$t_{i3} \equiv 1/z_{i1}$,	$c_3 \equiv a_2b_1$
$t_{i4} \equiv z_{i2}/z_{i1}$,	$c_4 \equiv a_2b_2$
$t_{i5} \equiv \ln(z_{i1})$,	$c_5 \equiv a_3b_1$
$t_{i6} \equiv z_{i2}\ln(z_{i1})$,	$c_6 \equiv a_3b_2$
$t_{i7} \equiv z_{i1}$,	$c_7 \equiv a_4b_1$
$t_{i8} \equiv z_{i2}z_{i1}$,	$c_8 \equiv a_4b_2$
etc.	etc,

Table IV contains the results of the IVOR analysis. The results of a stepwise regression analysis of this data appears in an article by Weiss (25). In his analysis, only eight of the twelve variables in the model were entered in the regression equation because the sum of the squares of the residuals divided by the number of degrees of

TABLE IV

RESULTS OF THE IVOR ANALYSIS OF THE OXYGEN SOLUBILITY DATA

Number of Variables in the Equation	Number of the Variable Entered	Length of the Basic Solution Vector	Length of the Residual Vector for the Basic Solution	Length of the Minimum Norm Solution Vector	Length of Residual Vector for the Minimum Norm Solution
1	2	$.4040704 \times 10^{5}$	2419.978	$.1014893 \times 10^{5}$	7617.572
2	1	$.9049813 \times 10^{5}$	981.9752	$.1028102 \times 10^{5}$	6018.982
3	7	$.8972010 \times 10^{5}$	212.4384	$.1031885 \times 10^{5}$	7818.671
4	3	$.3168621 \times 10^{7}$	82.10176	$.1091479 \times 10^{5}$	4422.227
5	8	$.1634972 \times 10^{7}$	28.52036	$.1092203 \times 10^{5}$	4558.884
6	12	$.1634366 \times 10^{7}$	27.99304	$.1097797 \times 10^{5}$	3548.216
7	4	$.1033133 \times 10^{8}$	26.44726	$.1103234 \times 10^{5}$	3473.011
8	9	$.1005459 \times 10^{8}$	26.06484	$.1197020 \times 10^{5}$	2147.965
9	5	$.1540121 \times 10^{9}$	25.942180	$.1205483 \times 10^{5}$	454.9831
10	6	$.2658320 \times 10^{11}$	25.50685	$.1207378 \times 10^{5}$	29.94534
11	10	$.3464215 \times 10^{11}$	25.27790	$.2317320 \times 10^{8}$	42.48019
12	11	$.3202155 \times 10^{11}$	25.12197	$.3202155 \times 10^{11}$	25.12197

freedom (m-n) failed to decrease after eight variables had entered the equation. This was caused by the use of the normal equations to perform the stepwise regression analysis.

Twelve variables were entered in the regression equation by the LLCR package. The length of the residual vector continued to decrease with each variable added to the regression equation.

Note that with all twelve variables in the regression equation, the length of the solution vector was $.3202155 \times 10^{11}$ with a residual vector of length 25.12197. The length of the solution vector with seven variables in the regression equation was $.1033133 \times 10^{8}$ with a residual vector of length 26.44726. Thus, with a modest increase in the length of the residual vector, a large decrease in the length of the solution vector was obtained.

In Chapter II, it was stated that if the true rank of the coefficient matrix is less than the number of columns, there is a linear subspace of solution vectors with a residual vector of some minimum length. Among all the vectors in that subspace, there is a unique vector of minimum length (7). All the components of this vector are nonzero. An attempt was made to consider the variables not entered in the regression equation to be linear combinations of the variables represented in the regression equation. The least squares solution of minimum length was then calculated as if the coefficient matrix had a rank equal to the number of variables in the regression equation. The length of each of these solution vectors was considerably less than the length of the basic approximate solution vector (BAS) for the same rank. The lengths of the residual vectors were unacceptably high in most cases. Table IV contains the results of this analysis.

Using the Package to Test the Effects of Decreasing the Precision of the Entries in the Coefficient Matrix

As mentioned in Chapter IV, the user can request that the coefficient matrix be treated as if it had a rank equal to k. This corresponds to increasing the value of ε until the rank of the coefficient matrix is k, where ε is the value used to determine the numerical rank of the coefficient matrix. If the true numerical rank is less than the rank requested, the true minimum length solution is found. If not, n-k columns of the coefficient matrix are considered to be linear combinations of the k columns that are chosen to be made mutually orthogonal; the minimum norm solution for rank k is found.

The ability to specify a rank enables the user to test the effect on the solution vector of measuring the entries in the coefficient matrix less accurately. During one run of the package, the solution vectors for each choice of rank ranging from one to min(m,n) can be found. This corresponds to finding the solution to the generalized linear least squares problem for a range of ε. Osborne's method of column selection is used to select the columns to be made mutually orthogonal when the package is used for this purpose.

The solution vectors for a range of ranks were found for the oxygen solubility data. Table V contains the results of the analysis.

Note that for ranks ten and eleven, the length of the minimum length solution vectors greatly decreased with only a moderate increase in the length of the residual vector. For example, the minimum length solution vector's length was $.12072\text{x}10^{5}$ with a residual vector of length 26.1 for rank ten. For rank eleven, the minimum length solution bector's length was $.3462100\text{x}10^{7}$ with a residual vector of length 25.6. In contrast, when IVOR was performed and eleven variables had entered the regression equation, the length of the solution vector was $.3464215\text{x}10^{11}$ with a residual vector of length 25.3. When ten variables had entered the regression equation, the length of the solution vector was $.1033\text{x}10^{8}$ with a residual vector of length 25.5.

If the user's objective is to obtain the best trade-off between

TABLE V

ANALYSIS OF OXYGEN SOLUBILITY DATA
USING OSBORNE PIVOTING

Rank	Basic Variable Entered in the Equation	Length of the Basic Solution Vector	Length of the Residual Vector for the BAS	Length of the Minimum Norm Solution Vector	Length of the Residual Vector for the Minimum Norm Solution
1	1	$.1612785 \times 10^{5}$	$.4380520 \times 10^{4}$	$.1008446 \times 10^{5}$	$.8425207 \times 10^{4}$
2	10	$.1614233 \times 10^{5}$	$.4168454 \times 10^{4}$	$.102177 \times 10^{5}$	$.7203746 \times 10^{4}$
3	11	$.161299 \times 10^{5}$	$.4164684 \times 10^{4}$	$.102470 \times 10^{5}$	$.5419911 \times 10^{4}$
4	4	$.1649864 \times 10^{5}$	$.1013336 \times 10^{4}$	$.1045335 \times 10^{5}$	$.2508875 \times 10^{4}$
5	5	$.1652604 \times 10^{5}$	$.9690841 \times 10^{3}$	$.1051180 \times 10^{5}$	$.1671074 \times 10^{4}$
6	12	$.1652779 \times 10^{5}$	$.9588091 \times 10^{3}$	$.1051873 \times 10^{5}$	$.1564966 \times 10^{4}$
7	7	$.1724977 \times 10^{5}$	$.8021976 \times 10^{2}$	$.1118416 \times 10^{5}$	$.250944 \times 10^{4}$
8	6	$.1724570 \times 10^{5}$	$.7149151 \times 10^{2}$	$.1127042 \times 10^{5}$	$.236241 \times 10^{4}$
9	9	$.1724276 \times 10^{5}$	$.7111963 \times 10^{2}$	$.1196991 \times 10^{5}$	$.1190856 \times 10^{4}$
10	8	$.1723771 \times 10^{5}$	$.2611948 \times 10^{2}$	$.1207238 \times 10^{5}$	$.2611625 \times 10^{2}$
11	2	$.6880348 \times 10^{7}$	$.2565849 \times 10^{2}$	$.3462101 \times 10^{7}$	$.2565829 \times 10^{2}$
12	3	$.3202155 \times 10^{11}$	$.2512197 \times 10^{2}$	$.3202155 \times 10^{11}$	$.2512197 \times 10^{2}$

the length of the solution vector and the length of the residual vector, the package should be run once with IVOR and once with Osborne pivoting.

Comparison of Methods

The package consisting of LLCR and LLSQ appears to be the first accurate IVOR (stepwise regression) package for ill-conditioned systems of equations. Until this time, stepwise regression packages have solved the normal equations,

$$A^T A\underline{x} = A^T \underline{b}.$$

The normal equations are often very ill-conditioned, making double precision calculations necessary (3). Longley (10) has shown examples where essentially no correct digits were obtained when the normal equations were solved. In addition, refinement of the intermediate solutions is available with this package.

For ill-conditioned systems using modified Gram-Schmidt orthogonalization produces a much more accurate solution vector than using the normal equations to solve a linear least squares problem. For a mathematical comparison of the accuracy of the methods, see article's by Björck (26), Golub (27), and Wampler (11).

Number of Operations and Storage Requirements

If the coefficient matrix is of full rank, the package requires approximately mn^2 multiplications and $mn + m + n^2 + 2n$ storage locations to calculate the linear least squares solution of minimum norm when iterative refinement of the solution is not performed. This should be contrasted with $2mn^2 + \frac{4n^3}{3}$ single precision multiplications and n^2 storage locations needed for forming and solving the normal equations in double precision (24).

If iterative refinement is performed, another $m(n+1)$ storage locations are required. An additional $n^2 + kn$ locations are needed if the system is solved for more than one rank, where k is the number of ranks for which the system is solved.

If the coefficient matrix is of deficient rank, r^*, the number of multiplications necessary for finding the solution is $mn^2-(n-r^*-1)(m+n)$ multiplications. The storage requirements are the same as for the full rank case.

When the package is used to perform IVOR, the number of operations necessary to add the k-th variable to the regression equation is $2m(n-k+\frac{mn}{2}-3nk+n^2)$. The total number of operations would be $4m^2n^2+ mn^3-n$. Approximately $(n+2)^2(m+\frac{3n}{2})$ operations are required for stepwise regression using the normal equations (19). If double precision calculations are necessary to obtain single precision accuracy, the comparison is more favorable.

The user of the LLCR package should consider putting all the floating point variables in double precision when solving an ill-conditioned system of equations. Refinement of the initial solutions would be ineffective so no more storage would be required than would be when the calculations were done in single precision with the initial solutions being refined. δ should be chosen so that $1.D0 + \delta > 1.D0$ when doing all calculations in double precision. The solution process is slower when all calculations are done in double precision; the results should be more accurate, however.

There are advantages to being able to solve the system of equations for a range of ranks during one run of the LLCR package instead of using a routine like Björck's. Beginning with a guess, several runs might be necessary to find the ε to produce the desired rank. In addition, if the results were sought for a range of ranks, Björck's routine would require that the first h-1 columns of the coefficient matrix be orthogonalized for each rank h for which the solution was desired. The LLCR package requires that only one column of the coefficient matrix be orthogonalized after the solution vector for the first rank is found. $2n^2$ additional words are required for this feature, however.

Table VI contains a list of the various uses of the package, and the method of column selection, the ranks for which the system is solved, and the extra storage required for each use.

Summary

Routines have been written that use modified Gram-Schmidt orthogonalization to solve the generalized linear least squares problem. Both a basic approximate solution and the least squares solution of minimum Euclidean norm are found. Improvement in the accuracy of the solutions by means of iterative refinement of the initial solutions is available to the user of these routines. Full single-precision accuracy in the solutions is obtained when iterative improvement of the solutions is performed and the parameter that is used to determine the numerical rank of the system is at least as great as the relative accuracy of the computer on which the package is run. The error matrix, $(A^TA)^{-1}$, is returned for systems of full rank.

The routines can be used to determine efficiently and accurately the generalized inverse of an arbitrary matrix, A. This is accomplished by solving the system of equations

$$AX = I$$

for the matrix X. I is an m by m identity matrix, where n is the number of rows in the matrix A. The generalized inverse, X, is an n

TABLE VI

USES OF THE PACKAGE CONSISTING OF LLCR AND LLSQ

Problem	Method of Column Selection	The Solution will be found for Ranks	Amount of Extra Storage Needed
To calculate the generalized inverse of A, an m by n matrix	Osborne pivoting	the numerical rank of A	m(m-1)
To find the solution of an arbitrary system of equations, $A\underline{x} = \underline{b}$	Osborne pivoting	the numerical rank of A	none
To perform stepwise regression without the deletion of variables	IVOR	1 to the numerical rank of the coefficient matrix	$2n^2$
To study the effect of decreasing the reliability of the entries in the coefficient matrix (solve the system for a range of ranks)	Osborne pivoting	$1 \le j \le$ numerical rank of A (a range for j is chosen by the user)	$2n^2$
To find the best trade-off between the length of the solution vector and the residual vector	IVOR and Osborne pivoting	1 to the numerical rank of A	$2n^2$
To fit a polynomial to a set of data points	Sequential selection	1 to the degree of the polynomial	$2n^2$

by m matrix, where n is the number of columns in A.

IVOR, or forward selection, has been implemented. IVOR corresponds to stepwise regression without the deletion of variables from the regression equation. The package appears to be the first accurate stepwise regression package for ill-conditioned problems.

The coefficient matrix can be treated as if it had a user-specified rank, k. This corresponds to increasing the value of the parameter that is used to determine the numerical rank of the coefficient matrix until the rank is k. This facility can be used to test the sensitivity of the solution vector to decreased precision of the entries in the coefficient matrix.

A SELECTED BIBLIOGRAPHY

(1) Osborne, E. E. "Smallest Least Square Solutions of Linear Equations." J. SIAM Numer. Anal. Ser. B., Vol. 2 (1965), 300-307.

(2) Gauss, C. F. "Theoria Combinations Observationum Erroribus Minimis Obnoxiae." Werke, Vol. 4 (1873), 3-93.

(3) Forsythe, G. and C. B. Moler. Computer Solution of Linear Systems. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.; 1967.

(4) Abt, K., Gemmill, G., Herring, T. and R. Shade. "DA-MRCA: A FORTRAN IV Program for Multiple Linear Regression." Naval Weapons Laboratory Report No. 2035, 1966.

(5) Cooper, L. and D. Steinberg. Introduction to Methods of Optimization. Philadelphia: W. B. Saunders Company; 1970.

(6) Hadley, G. Linear Algebra. Reading, Massachusetts: Addison-Wesley; 1961.

(7) Penrose, R. "On Best Approximate Solutions of Linear Matrix Equations." Proc. Cambridge Philos. Soc., Vol. 52 (1956), 17-19.

(8) Rosen, J. B. "Minimum and Basic Solutions to Singular Linear Systems." J. Soc. Indust. Appl. Math., Vol. 12 (1964), 156-161.

(9) Fox, L. An Introduction to Numerical Linear Algebra. New York: Oxford University Press; 1965.

(10) Longley, J. "An Appraisal of Least Squares Problems for the Electronic Computer from the Point of View of the User." JASA, Vol. 62 (1967), 819-841.

(11) Wampler, R. "A Report on the Accuracy of Some Widely Used Least Squares Programs." JASA, Vol. 65 (1970), 528-532.

(12) Osborne, E. E. "On Least Squares Solutions Linear Equations." J. Assoc. Comput. Mach., Vol. 8 (1961), 628-636.

(13) Businger, P. and G. Golub. "Linear Least Squares Solutions by Householder Transformations." Numer. Math., Vol. 7 (1965) 269-276.

(14) Golub, G. "Numerical Methods for Solving Linear Least Squares Problems." Numer. Math., Vol. 7 (1965), 206-216.

(15) Björck, A. and G. H. Golub. "Iterative Refinement of Linear Least Squares Solutions by Householder Transformation." BIT, Vol. 7 (1967), 323-337.

(16) Hanson, R. J. and C. L. Lawson. "Extensions and Applications of the Householder Algorithms for Solving Linear Least Squares Problems." Mathematics of Computation, Vol. 23 (1969), 787-812.

(17) Bauer, F. L. "Elimination with Weighted Row Combinations for Solving Linear Equations and Least Squares Problems." Numer. Math., Vol. 7 (1964), 338-352.

(18) Björck, A. "Iterative Refinement of Linear Least Squares Solutions II." BIT, Vol. 8 (1968), 8-30.

(19) Efroymson, M. A. "Multiple Regression Analysis." in Mathematical Methods for Digital Computers (Ralston, A. and H. S. Wilfe, Ed.). New York: John Wiley and Sons, Inc.; 1960.

(20) Dixon, W. and F. Massey. Introduction to Statistical Analysis. New York: McGraw-Hill; 1957.

(21) System/360 Scientific Subroutine Package (360A-CM-03X) Version III Programmer's Manual, IBM H20-0205. White Plains, New York: IBM Corporation Technical Publications Dept.; 1971.

(22) BMD Biomedical Computer Programs (Dixon, W. J., Ed.). Berkeley, California: University of California Press; 1968.

(23) Summers, D. L. and D. D. Babb. "Unfolding Pulse-Height Distributions by Vector Analysis." in Applications of Computers to Nuclear and Radiochemistry. Washington, D. C.: Atomic Energy Commission; 1962.

(24) USA Standard FORTRAN. ANSI Standard X3.9-1966. New York: American National Standards Institute; 1966.

(25) Weiss, R. F. "The Solubility of Nitrogen, Oxygen, and Argon in Water and Seawater." Deep-Sea Research, Vol. 17 (1970), 721.

(26) Björck, A. "Solving Linear Least Squares Problems by Gram-Schmidt Orthogonalization." BIT, Vol. 7 (1967), 1-21.

(27) Golub, G. "Matrix Decompositions and Statistical Calculations." in Statistical Computations. New York: Academic Press; 1969.

APPENDIX A

COMPUTER LISTINGS OF LLCR AND LLSQ

```
274       SUBROUTINE LLCR(A,LA,R,LRE,SAVE,LSAV,X,RHO,RHOM,SA,ERR,RES,V,XOLD,
         *SALPH,LSAL,NR,NC,BASIC,NRHS,XSAVE)
C AUTHOR        JOANNA C. HWANG
C
C DATE          MARCH 1, 1972                   VERSION 1.1
C
C BRIEF DESCRIPTION OF THIS PROGRAM....
C     THIS ROUTINE WILL GIVE THE SOLUTION OF MINIMUM NORM TO THE
C GENERAL LINEAR LEAST SQUARES PROBLEM.  MODIFIED GRAM-SCHMIDT ORTHOG-
C ONALIZATION IS USED TO OBTAIN THIS SOLUTION.
C     THE FOLLOWING OPTIONS HAVE BEEN IMPLEMENTED.
C 1. THE SOLUTIONS FOR MULTIPLE RIGHT HAND SIDES WITH A SINGLE
C    COEFFICIENT MATRIX CAN BE FOUND DURING ONE CALL TO THE ROUTINE.
C 2. REFINEMENT OF THE INITIAL BASIC APPROXIMATE AND INITIAL MINIMUM
C    NORM SOLUTIONS IS AVAILABLE.
C 3. THE ERROR MATRIX, THE INVERSE OF THE PRODUCT OF THE COEFFICIENT
C    MATRIX AND ITS TRANPOSE, IS CALCULATED FOR SYSTEMS OF FULL RANK.
C 4. THE BASIC SOLUTIONS CAN BE PRINTED.
C    ASSUME THAT THE RANK OF THE COEFFICIENT MATRIX IS IRANK, WHERE
C    IRANK IS LESS THAN THE NUMBER OF COLUMNS IN THE COEFFICIENT MATRIX.
C    THE BASIC SOLUTION OBTAINED IS THE SOLUTION WITH AT MOST IRANK
C    NONZERO COMPONENTS THAT GIVES THE MINIMUM EUCLIDEAN NORM.
C 5. THE USER CAN REQUEST THAT IVOR, INDEPENDENT VARIABLE ORDERING
C    BY REGRESSION SUM OF SQUARES,  BE PERFORMED.  IVOR CORRESPONDS TO
C    PERFORMING STEPWISE REGRESSION WITHOUT REMOVING A VARIABLE FROM
C    THE REGRESSION EQUATION ONCE IT HAS ENTERED THE REGRESSION
C    EQUATION.
C 6. THE USER CAN REQUEST THAT THE COEFFICIENT MATRIX BE TREATED AS IF
C    IT HAD A RANGE OF RANKS, KRBEG THROUGH KREND.  IF THE RANK
C    REQUESTED IS GREATER THAN THE NUMERICAL RANK OF THE SYSTEM
C    (DETERMINED BY THE RELATIVE ACCURACY EPS), A MESSAGE IS PRINTED.
C     FOR A MORE COMPLETE DESCRIPTION OF THIS ROUTINE, SEE THE WRITE-UP
C IN THE AUTHOR-S M.S. REPORT (DEPARTMENT OF COMPUTING AND INFORMATION
C SCIENCES, OKLAHOMA STATE UNIVERSITY, MAY, 1972).
C
C REFERENCES....
C     E. E. OSBORNE, JOURNAL OF THE SIAM 12 (1965) 300
C     J. B. ROSEN, JOURNAL OF THE SIAM 12 (1964) 156
C     G. GOLUB, NUMERISCHE MATHEMATIK 7 (1965) 206
C     JOHN R. RICE, MATHEMATICS OF COMPUTATION 20 (1966) 325
C     A. BJORCK, BIT 7 (1967) 257
C     A. BJORCK, BIT 8 (1968) 8
C     A. BJORCK, BIT 7 (1967) 1
C
C DESCRIPTION OF SUBROUTINES CALLED....
C          LLSQ  --  THE INITIAL BASIC APPROXIMATE SOLUTION AND LEAST
C                    SQUARES SOLUTION OF MINIMUM NORM FOR EACH RIGHT HAND
C                    SIDE IS FOUND.
C
C DESCRIPTION OF VARIABLES....
C INPUT VARIABLES....
C             A  --  THE NR BY NCOLS AUGMENTED MATRIX
C                    THE NRHS RIGHT HAND SIDES ARE CONCATENATED WITH THE
C                    NR BY NC COEFFICIENT MATRIX TO FORM THE AUGMENTED
C                    MATRIX.
C                    THE FIRST IRANK COLUMNS OF A ARE MADE MUTUALLY
C                    ORTHOGONAL.  THE NEXT (NC-IRANK) COLUMNS ARE
C                    CONSIDERED TO BE ZERO VECTORS.  THIS TRANSFORMED
C                    MATRIX IS REFERRED TO BELOW AS A .
```

```
C                                                N                  LLCR0600
C          LA  --  THE FIRST DIMENSION OF THE A ARRAY               LLCR0610
C                  (LA MUST BE .GE. NR.)                            LLCR0620
C         LRE  --  THE FIRST DIMENSION OF THE R AND ERR ARRAYS      LLCR0630
C                  (LRE MUST BE .GE. NC.)                           LLCR0640
C        LSAV  --  THE FIRST DIMENSION OF THE ARRAY SAVE            LLCR0650
C                  (LSAV MUST BE .GE. NR IF ITERATIVE IMPROVEMENT OF THELLCR0660
C                  SOLUTION IS REQUESTED.  LSAV SHOULD EQUAL 1      LLCR0670
C                  OTHERWISE.)                                      LLCR0680
C        LSAL  --  THE FIRST DIMENSION OF THE ARRAY SALPH           LLCR0690
C                  (LSAL MUST BE .GE. NC IF THE PROBLEM IS TO BE SOLVED LLCR0700
C                  FOR MORE THAN ONE RANK.  LSAL SHOULD EQUAL 1     LLCR0710
C                  OTHERWISE.                                       LLCR0720
C        EPS1  --  THE CONVERGENCE CRITERION FOR THE ITERATIVE IMPROVE- LLCR0730
C                  MENT OF THE SOLUTION                             LLCR0740
C                  EACH COMPONENT IN THE FINAL SOLUTION VECTOR WILL LLCR0750
C                  DIFFER FROM THE CORRESPONDING COMPONENT IN THE   LLCR0760
C                  PREVIOUS SOLUTION VECTOR BY NO MORE THAN EPS1    LLCR0770
C                  TIMES THE COMPONENT IN THE FINAL SOLUTION VECTOR. LLCR0780
C         EPS  --  THE VALUE USED TO DETERMINE IF A COLUMN IN THE   LLCR0790
C                  COEFFICIENT MATRIX IS A LINEAR COMBINATION OF OTHER LLCR0800
C                  COLUMNS IN THE COEFFICIENT MATRIX                LLCR0810
C                  NEITHER EPS NOR EPS1 SHOULD BE LESS THAN THE PRODUCT LLCR0820
C                  OF THE BASE AND THE RELATIVE ACCURACY OF THE MACHINE LLCR0830
C                  BEING USED.                                      LLCR0840
C          NR  --  THE NUMBER OF EQUATIONS IN THE SYSTEM            LLCR0850
C                  (NUMBER OF ROWS OF A)                            LLCR0860
C          NC  --  THE NUMBER OF INDEPENDENT VARIABLES              LLCR0870
C                  (NUMBER OF COLUMNS OF A BEFORE IT IS AUGMENTED)  LLCR0880
C        NRHS  --  THE NUMBER OF RIGHT HAND SIDES                   LLCR0890
C                  (NRHS MUST BE .GE. 1.)                           LLCR0900
C        IPIV  -- =-1 NO PIVOTING IS PERFORMED.                     LLCR0910
C                 = 0 OSBORNE PIVOTING IS PERFORMED.                LLCR0920
C                 = 1 IVOR IS PERFORMED.                            LLCR0930
C                     SEE THE ARTICLE BY OSBORNE FOR A DESCRIPTION OF LLCR0940
C                     OSBORNE PIVOTING.                             LLCR0950
C         ISW  -- =1 IF THE SQUARE OF THE NORM IS TO BE RECOMPUTED USINGLLCR0960
C                    INNER PRODUCTS                                 LLCR0970
C                    OTHERWISE, THE SQUARE OF THE NORM IS RECOMPUTED LLCR0980
C                    USING THE METHOD PROPOSED BY OSBORNE.          LLCR0990
C        IREF  -- = 1 IF THE INITIAL SOLUTION IS TO BE REFINED      LLCR1000
C                 = 0 IF THE SOLUTION IS NOT TO BE REFINED BUT THE  LLCR1010
C                     RESIDUAL VECTOR IS TO BE CALCULATED           LLCR1020
C                 =-1 THE SOLUTION IS NOT TO BE REFINED AND THE RESIDUALLLCR1030
C                     VECTOR CANNOT BE CALCULATED                   LLCR1040
C                     (SAVE AND A ARE THE SAME MATRIX.)             LLCR1050
C       NTRAC  -- =-1 ERROR MESSAGES ONLY ARE PRINTED.              LLCR1060
C                 = 0 THE FINAL SOLUTION VECTORS AND THE RANK OF THE LLCR1070
C                     COEFFICIENT MATRIX ARE PRINTED IN ADDITION TO THE LLCR1080
C                     ABOVE.                                        LLCR1090
C                 = 1 THE INTERMEDIATE SOLUTION VECTORS, THE RESIDUAL LLCR1100
C                     VECTORS FOR EACH INTERMEDIATE SOLUTION AND THE LLCR1110
C                     FINAL SOLUTION, AND THE ERROR MATRIX ARE PRINTED LLCR1120
C                     IN ADDITION TO THE ABOVE.                     LLCR1130
C                 = 2 THE ORIGINAL COEFFICIENT MATRIX, THE ORIGINAL LLCR1140
C                     RIGHT HAND SIDES, AND THE DECOMPOSITION MATRIX ARELLCR1150
C                     PRINTED IN ADDITION TO THE ITEMS LISTED ABOVE. LLCR1160
C          KW  --  THE STANDARD OUTPUT UNIT NUMBER                  LLCR1170
C KRBEG,KREND  --  AN ATTEMPT WILL BE MADE TO SOLVE THE SYSTEMS OF  LLCR1180
C                  EQUATIONS AS IF THE COEFFICIENT MATRIX HAD A RANK OF LLCR1190
```

```
C                   KRBEG,KRBEG+1,....,KREND.  IF THE TRUE RANK OF THE
C                   SYSTEM IS LESS THAN THE RANK THE USER REQUESTS, A
C                   MESSAGE IS PRINTED.
C                   KRBEG MUST BE SET GREATER THAN OR EQUAL TO ONE.  IF
C                   THE RANK OF THE COEFFICIENT MATRIX IS ZERO, THE
C                   CORRECT SOLUTION WILL BE RETURNED, HOWEVER.
C                   SEE THE DEFINITION OF KRANK BELOW.
C       MAXIT  --   THE MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR THE
C                   ITERATIVE IMPROVEMENT PROCEDURE
C                   MAXIT SHOULD BE GREATER OR EQUAL TO ONE IF ITERATIVE
C                   IMPROVEMENT OF THE SOLUTION IS NOT DESIRED.
C
C OUTPUT VARIABLES....
C           R  --   AN NC BY NCOLS MATRIX THAT INITIALLY HOLDS AN
C                   IDENTITY MATRIX IN ITS FIRST NC COLUMNS AND ZERO
C                   COLUMNS FOR THE REMAINING (NCOLS-NC) COLUMNS
C                   THE FIRST NC COLUMNS OF R ARE TRANSFORMED INTO THE R
C                                             -1
C                   OF THE DECOMPOSITION A=A R   .  THE REMAINING COLUMNS
C                                          N N
C                   HOLD THE SOLUTIONS TO THE SYSTEMS OF EQUATIONS.
C XSAVE(I,J,K) --   IF IPIV.EQ.1, THE I-TH COEFFICIENT OF THE REGRESSION
C                   EQUATION FOR RIGHT HAND SIDE NUMBER J WHEN KRBEG
C                   +K-1 VARIABLES HAVE ENTERED THE REGRESSION EQUATION
C                   OTHERWISE, THE I-TH COMPONENT OF THE SOLUTION VECTOR
C                   OF MINIMUM LENGTH FOR RIGHT HAND SIDE NUMBER J WHEN
C                   RANK KRBEG+K-1
C    BASIC(J)  --   THE LENGTH OF THE BASIC APPROXIMATE SOLUTION VECTOR
C                   FOR THE J-TH RIGHT HAND SIDE
C           X  --   THE SOLUTION VECTOR FOR EACH RIGHT HAND SIDE
C                   IS PLACED IN X PRIOR TO THE ITERATIVE IMPROVEMENT OF
C                   THE SOLUTION FOR THAT RIGHT HAND SIDE.  ITERATIVE
C                   IMPROVEMENT IS PERFORMED ON THE SOLUTION VECTOR FOR
C                   ONE RIGHT HAND SIDE AT A TIME.
C         RES  --   THE RESIDUAL VECTORS FOR THE RIGHT HAND SIDES
C         ERR  --   THE INVERSE OF THE TRANSPOSE OF THE COEFFICIENT
C                   MATRIX TIMES ITS TRANSPOSE
C                   THIS IS CALCULATED ONLY IF THE COEFFICIENT MATRIX IS
C                   OF FULL RANK.  THIS IS AN NC BY NC ARRAY.
C                   (A IS THE UNAUGMENTED COEFFICIENT MATRIX IN THIS
C                   CASE.)
C       IRANK  --   THE SMALLER OF THE NUMERICAL RANK OF THE COEFFICIENT
C                   MATRIX AND KRANK
C       NFAIL  -- =0 IF ITERATIVE IMPROVEMENT FAILS TO PRODUCE RESULTS
C                     OF THE DESIRED ACCURACY WITHIN MAXIT ITERATIONS
C              -- =1 OTHERWISE
C
C INTERMEDIATE VARIABLES....
C       SALPH  --   HOLDS R IN THE STATE IT IS IN AFTER THE BASIC
C                   APPROXIMATE SOLUTION IS FOUND FOR A GIVEN RANK
C                   THIS MATRIX IS NOT NEEDED IF KRBEG.EQ.KREND.
C        SAVE  --   HOLDS THE ORIGINAL AUGMENTED MATRIX, A
C                   THIS MATRIX IS NOT REFERENCED IF ITERATIVE IMPROVE-
C                   MENT OF THE SOLUTIONS IS NOT DESIRED.
C        XOLD  --   THE SOLUTION VECTOR FOR EACH RIGHT HAND SIDE FOUND
C                   DURING THE PREVIOUS ITERATION OF THE IMPROVEMENT
C                   PROCEDURE IS PLACED IN XOLD.
C      RHO(K)  --   THE SQUARE OF THE NORM OF THE K-TH COLUMN OF A
C     RHOM(K)  --   THE SQUARE OF THE NORM OF THE K-TH COLUMN OF R
C       KRANK  --   AN ATTEMPT IS MADE TO FIND THE SOLUTION
```

```
      C                 AS IF THE COEFFICIENT MATRIX HAD A RANK OF KRANK.
      C       KROLD  --  THE PREVIOUS SOLUTION OF THE SYSTEMS OF EQUATIONS
      C                 WAS MADE AS IF THE RANK OF THE COEFFICIENT MATRIX
      C                 WAS KROLD.
      C         NCP  --  THE NUMBER OF INDEPENDENT VARIABLES PLUS ONE
      C       NCOLS  --  THE NUMBER OF INDEPENDENT VARIABLES PLUS THE NUMBER
      C                 OF RIGHT HAND SIDES
      C
      C USE OF THIS PROGRAM....
      C     TO USE THIS PROGRAM, A CALLING PROGRAM THAT CALLS THE SUBROUTINE
      C LLCR MUST BE WRITTEN.  THE CALLING PROGRAM MUST DEFINE THE VARIABLES
      C A, LA, NR, NC, NRHS, LRE, EPS1, EPS, IBAS, ISW, NTRAC, KRBEG, KREND,
      C KW, IREF, LSAV, LSAL, AND MAXIT.
      C     THE FOLLOWING STATEMENTS MUST APPEAR IN THE CALLING PROGRAM.
      C     DIMENSION A(LA,NCOLS),R(LRE,NCOLS),SAVE(LSAV,NCOLS),X(NC),RHO(NC),
      C    1SA(NC),ERR(LRE,NC),RES(NR),V(NC),XOLD(NC),SALPH(LSAL,NCOLS),
      C    2RHOM(NC),BASIC(NRHS),XSAVE(NC,NRHS,KREND-KRBEG+1)
      C     COMMON/BETA/EPS,EPS1,IPIV,ISW,NTRAC,NCOLS,KRBEG,KREND,KRANK,KROLD,
      C    1IREF,KW,NCP,IRANK,MAXIT,NFAIL,IRGT
      C
      C     (DEFINITION OF VARIABLES)
      C
      C     CALL LLCR(A,LA,R,LRE,SAVE,LSAV,X,RHO,RHOM,SA,ERR,RES,V,XOLD,SALPH,
      C    1LSAL,NR,NC,BASIC,NRHS,XSAVE)
      C     CALL EXIT
      C     END
      C
      C     THE NUMERICAL VALUES OF THE VARIABLES USED TO INDICATE THE
      C DIMENSIONS OF THE ABOVE ARRAYS MUST APPEAR IN THE DIMENSION STATEMENT
      C IN THE CALLING PROGRAM.  THE DIMENSIONS INDICATED IN THE DIMENSION
      C STATEMENT ARE MINIMUM DIMENSIONS.  LA MUST BE EQUAL TO OR GREATER THAN
      C NR.  LRE MUST BE EQUAL TO OR GREATER THAN NC.  IF ITERATIVE
      C IMPROVEMENT OF THE SOLUTION IS NOT USED, SAVE SHOULD BE DIMENSIONED
      C (1,1) IN THE CALLING PROGRAM, AND LSAV SHOULD BE SET EQUAL TO ONE.
      C OTHERWISE, LSAV SHOULD BE EQUAL TO OR GREATER THAN NR.  IF THE
      C PROBLEM IS GOING TO BE SOLVED FOR ONLY ONE RANK, SALPH SHOULD BE
      C DIMENSIONED (1,1) IN THE CALLING PROGRAM., AND LSAL SET EQUAL TO ONE.
      C OTHERWISE, LSAL SHOULD BE EQUAL TO OR GREATER THAN NC.
      C     IF THIS PROGRAM IS RUN UNDER A COMPILER THAT CHECKS SUBSCRIPTS OR
      C DOES NOT PERMIT VARIABLE SUBSCRIPTING, THE DIMENSION STATEMENTS IN THE
      C SUBROUTINES LLCR AND LLSQ WILL HAVE TO BE CHANGED.
      C     IF THE MACHINE ON WHICH THE PROGRAM IS BEING EXECUTED DOES NOT
      C HAVE LABELED COMMON, THE COMMON STATEMENT WILL HAVE TO BE CHANGED.
      C
275         DOUBLE PRECISION DRES,DINT,DINTP,DSAVE
      C
      C ....    IF SINGLE PRECISION DOES NOT GIVE THE ACCURACY DESIRED, ALL
      C ....    CALCULATIONS MUST BE DONE IN DOUBLE PRECISION.  TO DO THIS,
      C ....    REMOVE THE C IN COLUMN ONE FROM THE FOLLOWING STATEMENT.  IN
      C ....    THIS CASE, ITERATIVE IMPROVEMENT OF THE SOLUTIONS WOULD BE
      C ....    INEFFECTIVE AND SHOULD NOT BE REQUESTED.
      C     DOUBLE PRECISION A,R,RHO,SA,RHOM,SALPH,EPS,DENOM,DOT,ALPHA,
      C    1VZERO,ONE,EPS1,BASIC,QSQRT
      C
276         DIMENSION A(LA,1),R(LRE,1),SAVE(LSAV,1),X(NC),RHO(NC),SA(NC),
           *ERR(LRE,NC),RES(NR),V(NC),XOLD(NC),SALPH(LSAL,1),RHOM(NC),
           *BASIC(NRHS),XSAVE(NC,NRHS,1)
277         COMMON/BETA/EPS,EPS1,IPIV,ISW,NTRAC,NCOLS,KRBEG,KREND,KRANK,KROLD,
           *IREF,KW,NCP,IRANK,MAXIT,NFAIL,IRGT
      C
```

```
C ....                    CHANGE QSQRT(Y)=SQRT(Y) TO QSQRT(Y)=DSQRT(Y) IF LLCR2400
C ....                    ALL COMPUTATIONS ARE DONE IN DOUBLE PRECISION.  LLCR2410
C                                                                         LLCR2420
      QSQRT(Y)=SQRT(Y)                                                    LLCR2430
C                                                                         LLCR2440
      IF(NTRAC)20,10,10                                                   LLCR2450
   10 WRITE(KW,870)EPS1,EPS,NR,NC,ISW,IREF,MAXIT,KRBEG,KREND              LLCR2460
   20 NCP=NC+1                                                            LLCR2470
      IRGT=0                                                              LLCR2480
      VZERO=0.                                                            LLCR2490
      KROLD=-1                                                            LLCR2500
      ONE=1.                                                              LLCR2510
      IRANK=NCP                                                           LLCR2520
      NFAIL=1                                                             LLCR2530
      NCOLS=NC+NRHS                                                       LLCR2540
      EPSQ=EPS*EPS                                                        LLCR2550
C ....                    SAVE THE ORIGINAL COEFFICIENT MATRIX AND RIGHT  LLCR2560
C ....                    HAND SIDES.                                     LLCR2570
      IF(IREF)50,30,30                                                    LLCR2580
   30 DO 40 IR=1,NR                                                       LLCR2590
      DO 40 IC=1,NCOLS                                                    LLCR2600
   40 SAVE(IR,IC)=A(IR,IC)                                                LLCR2610
C ....                    PRINT THE ORIGINAL COEFFICIENT MATRIX AND RIGHT LLCR2620
C ....                    HAND SIDES.                                     LLCR2630
   50 IF(NTRAC-2)90,60,60                                                 LLCR2640
   60 WRITE(KW,750)                                                       LLCR2650
      DO 70 I=1,NR                                                        LLCR2660
   70 WRITE(KW,810)(A(I,J),J=1,NC)                                        LLCR2670
      WRITE(KW,760)                                                       LLCR2680
      DO 80 IC=NCP,NCOLS                                                  LLCR2690
   80 WRITE(KW,810)(A(I,IC),I=1,NR)                                       LLCR2700
C ....                    NEGATE EACH RIGHT HAND SIDE.                    LLCR2710
   90 DO 100 I=1,NR                                                       LLCR2720
      DO 100 J=NCP,NCOLS                                                  LLCR2730
  100 A(I,J)=-A(I,J)                                                      LLCR2740
C ....                    FIND THE SOLUTIONS TO THE PROBLEM               LLCR2750
C ....                    FOR RANKS KRBEG THROUGH KREND.                  LLCR2760
      DO 730 MRANK=KRBEG,KREND                                            LLCR2770
      KRANK=MRANK                                                         LLCR2780
C ....                    WRITE A MESSAGE IF THE RANK ASKED FOR IS        LLCR2790
C ....                    KNOWN TO BE LESS THAN THE RANK OF THE SYSTEM.   LLCR2800
      IF(IRANK-KROLD)110,120,120                                          LLCR2810
  110 WRITE(KW,850)                                                       LLCR2820
      GO TO 740                                                           LLCR2830
C ....                    FIND THE INITIAL SOLUTION TO THE PROBLEM.       LLCR2840
  120 CALL LLSQ(A,LA,RHO,RHOM,SA,SALPH,LSAL,NR,NC,R,LRE,BASIC,NRHS,       LLCR2850
     *SAVE,LSAV,XSAVE)                                                    LLCR2860
      IF(IRGT)740,130,740                                                 LLCR2870
  130 KROLD=KRANK                                                         LLCR2880
C ....                    PRINT THE DECOMPOSED MATRIX.                    LLCR2890
      IF(NTRAC-2)170,140,140                                              LLCR2900
  140 WRITE(KW,840)IRANK                                                  LLCR2910
      DO 150 I=1,NR                                                       LLCR2920
  150 WRITE(KW,810)(A(I,J),J=1,NC)                                        LLCR2930
      DO 160 I=1,NC                                                       LLCR2940
  160 WRITE(KW,810)(R(I,J),J=1,NC)                                        LLCR2950
C ....                    FIND THE FINAL SOLUTION FOR EACH RIGHT HAND     LLCR2960
C ....                    SIDE.                                           LLCR2970
  170 DO 670 K=NCP,NCOLS                                                  LLCR2980
      KSW=-1                                                              LLCR2990
```

```
      DO 180 I=1,NC                                                     LLCR3000
  180 X(I)=R(I,K)                                                       LLCR3010
      KNT=0                                                             LLCR3020
C ....                 CHECK IF REFINEMENT OF THE SOLUTION IS DESIRED.  LLCR3030
      IF(IREF-1)510,190,510                                             LLCR3040
  190 KNT=0                                                             LLCR3050
C ....                 CHECK TO SEE IF THE BASIC APPROXIMATE OR THE     LLCR3060
C ....                 MINIMUM NORM SOLUTION IS TO BE ITERATED.         LLCR3070
      IF(KSW)200,220,670                                                LLCR3080
  200 WRITE(KW,900)                                                     LLCR3090
      KK=K-NC                                                           LLCR3100
      NO=KRANK-KRBEG+1                                                  LLCR3110
      DO 210 I=1,NC                                                     LLCR3120
  210 X(I)=XSAVE(I,KK,NO)                                               LLCR3130
      GO TO 240                                                         LLCR3140
  220 WRITE(KW,910)                                                     LLCR3150
      DO 230 I=1,NC                                                     LLCR3160
  230 X(I)=R(I,K)                                                       LLCR3170
  240 KNT=KNT+1                                                         LLCR3180
C ....                 CHECK FOR TOO MANY ITERATIONS TO ACHIEVE THE     LLCR3190
C ....                 DESIRED ACCURACY.                                LLCR3200
      IF(KNT-MAXIT)260,260,250                                          LLCR3210
  250 NO=K-NC                                                           LLCR3220
      WRITE(KW,790)NO,MAXIT                                             LLCR3230
      NFAIL=0                                                           LLCR3240
      GO TO 740                                                         LLCR3250
C ....                 CALCULATE THE RESIDUAL VECTOR FOR THE (K-NC)-TH  LLCR3260
C ....                 RIGHT HAND SIDE.                                 LLCR3270
  260 DO 280 I=1,NR                                                     LLCR3280
      DRES=VZERO                                                        LLCR3290
      DO 270 J=1,NC                                                     LLCR3300
      DINT=X(J)                                                         LLCR3310
      DINTP=SAVE(I,J)                                                   LLCR3320
  270 DRES=DRES+DINTP*DINT                                              LLCR3330
      DSAVE=SAVE(I,K)                                                   LLCR3340
  280 RES(I)=DSAVE-DRES                                                 LLCR3350
C ....                 CALCULATE THE SQUARE OF THE NORM OF THE RESIDUALLLCR3360
C ....                 VECTOR.                                          LLCR3370
      DNORM=VZERO                                                       LLCR3380
      DO 290 I=1,NR                                                     LLCR3390
  290 DNORM=DNORM+RES(I)*RES(I)                                         LLCR3400
C ....                 CALCULATE THE LENGTH OF THE RESIDUAL VECTOR.     LLCR3410
      SNORM=DNORM                                                       LLCR3420
      DNORM=QSQRT(DNORM)                                                LLCR3430
      NO=K-NC                                                           LLCR3440
C ....                 PRINT THE RESIDUAL VECTOR AND THE LENGTH OF THE  LLCR3450
C ....                 RESIDUAL VECTOR.                                 LLCR3460
      KNTM1=KNT-1                                                       LLCR3470
      IF(NTRAC)310,300,300                                              LLCR3480
  300 WRITE(KW,860)KNTM1                                                LLCR3490
      WRITE(KW,770)NO,(RES(I),I=1,NR)                                   LLCR3500
      WRITE(KW,920)DNORM,SNORM                                          LLCR3510
C ....                 PRINT THE NEW SOLUTION VECTOR.                   LLCR3520
      WRITE(KW,800)NO,(X(IC),IC=1,NC)                                   LLCR3530
C ....                 SOLVE THE PROBLEM COMPOSED OF THE                LLCR3540
C ....                 ORIGINAL COEFFICIENT MATRIX AND THE RESIDUAL     LLCR3550
C ....                 VECTOR FOR THE RIGHT HAND SIDE.                  LLCR3560
  310 DO 320 I=1,NC                                                     LLCR3570
  320 V(I)=VZERO                                                        LLCR3580
      DO 330 IR=1,NR                                                    LLCR3590
```

```
      330 RES(IR)=-RES(IR)                                             LLCR3600
          IF(IRANK)400,400,340                                         LLCR3610
      340 DO 380 I=1,IRANK                                             LLCR3620
          DOT=VZERO                                                    LLCR3630
          DO 350 J=1,NR                                                LLCR3640
      350 DOT=DOT+A(J,I)*RES(J)                                        LLCR3650
          ALPHA=DOT/RHO(I)                                             LLCR3660
          DO 360 J=1,NR                                                LLCR3670
      360 RES(J)=RES(J)-ALPHA*A(J,I)                                   LLCR3680
          DO 370 J=1,NC                                                LLCR3690
      370 V(J)=V(J)-ALPHA*R(J,I)                                       LLCR3700
      380 CONTINUE                                                     LLCR3710
          IF(KSW)440,390,390                                           LLCR3720
      390 IF(IRANK-NC)400,440,400                                      LLCR3730
      400 IRNKP=IRANK+1                                                LLCR3740
          DO 430 I=IRNKP,NC                                            LLCR3750
          DOT=VZERO                                                    LLCR3760
          DO 410 J=1,NC                                                LLCR3770
      410 DOT=DOT+R(J,I)*V(J)                                          LLCR3780
    C ....                      RHOM(I) CAN NEVER BE ZERO THEORETICALLY OR LLCR3790
    C ....                      NUMERICALLY.                           LLCR3800
          ALPHA=DOT/RHOM(I)                                            LLCR3810
          DO 420 J=1,NC                                                LLCR3820
      420 V(J)=V(J)-ALPHA*R(J,I)                                       LLCR3830
      430 CONTINUE                                                     LLCR3840
    C ....                      CALCULATE THE NEW SOLUTION VECTOR.     LLCR3850
      440 DO 450 I=1,NC                                                LLCR3860
          XOLD(I)=X(I)                                                 LLCR3870
      450 X(I)=X(I)+V(I)                                               LLCR3880
    C ....                      CHECK FOR CONVERGENCE.                 LLCR3890
          DO 500 I=1,NC                                                LLCR3900
          DIF=X(I)-XOLD(I)                                             LLCR3910
          IF(DIF)460,470,470                                           LLCR3920
      460 DIF=-DIF                                                     LLCR3930
      470 XOLD(I)=X(I)                                                 LLCR3940
          IF(XOLD(I))480,490,490                                       LLCR3950
      480 XOLD(I)=-XOLD(I)                                             LLCR3960
      490 IF(DIF-EPS1*XOLD(I))500,500,240                              LLCR3970
      500 CONTINUE                                                     LLCR3980
          IF(KSW)540,510,540                                           LLCR3990
      510 IF(IPIV-1)520,540,520                                        LLCR4000
      520 NO=KRANK-KRBEG+1                                             LLCR4010
          KK=K-NC                                                      LLCR4020
          DO 530 KL=1,NC                                               LLCR4030
      530 XSAVE(KL,KK,NO)=X(KL)                                        LLCR4040
      540 IF(NTRAC)670,550,550                                         LLCR4050
      550 WRITE(KW,860)KNT                                             LLCR4060
          NO=K-NC                                                      LLCR4070
          IF(IREF)630,560,560                                          LLCR4080
    C ....                      CALCULATE THE RESIDUAL VECTOR.         LLCR4090
      560 DO 580 I=1,NR                                                LLCR4100
          DRES=VZERO                                                   LLCR4110
          DO 570 J=1,NC                                                LLCR4120
          DINT=X(J)                                                    LLCR4130
          DINTP=SAVE(I,J)                                              LLCR4140
      570 DRES=DRES+DINTP*DINT                                         LLCR4150
          DSAVE=SAVE(I,K)                                              LLCR4160
      580 RES(I)=DSAVE-DRES                                            LLCR4170
          IF(NTRAC-2)600,590,600                                       LLCR4180
      590 WRITE(KW,770)NO,(RES(I),I=1,NR)                              LLCR4190
```

```
C ....                    CALCULATE THE LENGTH OF THE RESIDUAL VECTOR.    LLCR4200
  600 DNORM=VZERO                                                         LLCR4210
      DO 610 I=1,NR                                                       LLCR4220
  610 DNORM=DNORM+RES(I)*RES(I)                                           LLCR4230
      SNORM=DNORM                                                         LLCR4240
      DNORM=QSQRT(DNORM)                                                  LLCR4250
      WRITE(KW,780)DNORM,SNORM                                            LLCR4260
      IF(DNORM-BASIC(NO))630,620,620                                      LLCR4270
  620 WRITE(KW,880)                                                       LLCR4280
  630 WRITE(KW,830)NO,(X(I),I=1,NC)                                       LLCR4290
      DOT=VZERO                                                           LLCR4300
      DO 640 I=1,NC                                                       LLCR4310
  640 DOT=DOT+X(I)*X(I)                                                   LLCR4320
      DOT=QSQRT(DOT)                                                      LLCR4330
      WRITE(KW,890)DOT                                                    LLCR4340
      IF(KSW)650,670,670                                                  LLCR4350
  650 DO 660 KL=1,NC                                                      LLCR4360
  660 XSAVE(KL,KK,NO)=X(KL)                                               LLCR4370
      KSW=KSW+1                                                           LLCR4380
      GO TO 190                                                           LLCR4390
  670 CONTINUE                                                            LLCR4400
C ....                    CALCULATE THE ERROR MATRIX, THE INVERSE OF A    LLCR4410
C ....                    TRANSPOSE TIMES A, IF THE RANK OF THE SYSTEM IS LLCR4420
C ....                    NC.                                             LLCR4430
      IF(IRANK-NC) 730,680,730                                            LLCR4440
  680 DO 700 I=1,NC                                                       LLCR4450
      DO 700 J=1,NC                                                       LLCR4460
      DOT=VZERO                                                           LLCR4470
      DO 690 K=1,NC                                                       LLCR4480
  690 DOT=DOT+R(I,K)*R(J,K)/RHO(K)                                        LLCR4490
  700 ERR(I,J)=DOT                                                        LLCR4500
      IF(NTRAC-1)730,710,710                                              LLCR4510
  710 WRITE(KW,820)                                                       LLCR4520
      DO 720 IR=1,NC                                                      LLCR4530
  720 WRITE(KW,810)(ERR(IR,IC),IC=1,NC)                                   LLCR4540
  730 CONTINUE                                                            LLCR4550
  740 RETURN                                                              LLCR4560
C ....   CHANGE 5E20.7 TO 4D25.14 IF USING DOUBLE PRECISION               LLCR4570
C ....   CALCULATIONS.                                                    LLCR4580
  750 FORMAT(/32H THE ORIGINAL COEFFICIENT MATRIX)                        LLCR4590
  760 FORMAT(/30H THE ORIGINAL RIGHT HAND SIDES)                          LLCR4600
  770 FORMAT(/48H THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER ,I5,3H LLCR4610
     *IS/(1X,5E20.7))                                                     LLCR4620
  780 FORMAT(/61H THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTILLCR4630
     *ON IS ,E20.7/75H THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FLLCR4640
     *OR THE FINAL SOLUTION IS ,E20.7)                                    LLCR4650
  790 FORMAT(/72H THE CONVERGENCE CRITERION FOR THE SOLUTION WITH RIGHT LLCR4660
     *HAND SIDE NUMBER ,I5/20H WAS NOT REACHED IN ,I5,12H ITERATIONS./  LLCR4670
     *85H THE VALUE OF EPS IS TOO SMALL FOR THE MACHINE ON WHICH YOU ARELLCR4680
     * RUNNING THIS ROUTINE./40H INCREASE EPS AND RUN THE ROUTINE AGAIN.LLCR4690
     *)                                                                   LLCR4700
  800 FORMAT(/55H THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER LLCR4710
     *,I5/(1X,5E20.7))                                                    LLCR4720
  810 FORMAT(/(1X,5E20.7))                                                LLCR4730
  820 FORMAT(/17H THE ERROR MATRIX)                                       LLCR4740
  830 FORMAT(/65H THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SILLCR4750
     *DE NUMBER ,I5/(1X,5E20.7))                                          LLCR4760
  840 FORMAT(/44H THE DECOMPOSITION MATRIX FOR RANK EQUAL TO ,I5)        LLCR4770
  850 FORMAT(/58H THE RANK REQUESTED IS GREATER THAN THE RANK OF THE SYSLLCR4780
     *TEM)                                                                LLCR4790


  860 FORMAT(/11H ITERATION ,I3)                                          LLCR4800
  870 FORMAT(1H1,5HEPS1=,E10.3,5X,4HEPS=,E10.3,5X,3HNR=,I5,5X,3HNC=,      LLCR4810
     *I5,5X,4HISW=,I2/1X,5HIREF=,I2,5X,6HMAXIT=,I5,5X,6HKRBEG=,I5,5X,     LLCR4820
     *6HKREND=,I5)                                                        LLCR4830
  880 FORMAT(/76H THE BASIC APPROXIMATE SOLUTION IS A BETTER SOLUTION THLLCR4840
     *AN THE FINAL SOLUTION)                                              LLCR4850
  890 FORMAT(/44H THE LENGTH OF THE FINAL SOLUTION VECTOR IS ,E20.7)      LLCR4860
  900 FORMAT(///35H START ITERATING THE BASIC SOLUTION)                   LLCR4870
  910 FORMAT(///42H START ITERATING THE MINIMUM NORM SOLUTION)            LLCR4880
  920 FORMAT(/38H THE LENGTH OF THE RESIDUAL VECTOR IS ,E20.7/            LLCR4890
     *52H THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS ,E20.7)      LLCR4900
      END                                                                 LLCR4910

$ENTRY
```

```
87        SUBROUTINE LLSQ(A,LA,RHO,RHOM,SA,SALPH,LSAL,NR,NC,R,LRE,BASIC,
         *NRHS,SAVE,LSAV,XSAVE)
     C AUTHOR       JOANNA C. HWANG
     C
     C DATE         MARCH 1, 1972                       VERSION 1.1
     C
     C BRIEF DESCRIPTION OF THIS PROGRAM....
     C      THIS ROUTINE FINDS THE INITIAL BASIC APPROXIMATE SOLUTIONS AND
     C THE SOLUTION OF MINIMUM NORM FOR A GENERAL LINEAR LEAST SQUARES
     C PROBLEM, A*X=B, WHERE A IS AN NR BY NC MATRIX, X IS AN NC BY NRHS
     C MATRIX, AND B IS AN NR BY NRHS MATRIX.  A IS CONSIDERED TO HAVE A RANK
     C SPECIFIED BY THE USER.
     C      FOR REFERENCES, DESCRIPTIONS OF METHODS USED, AND DESCRIPTIONS
     C OF VARIABLES, SEE THE SUBROUTINE LLCR.
     C
88        DOUBLE PRECISION DRES,DINTP,DINT,DSAVE
     C
     C ....   IF SINGLE PRECISION DOES NOT GIVE THE ACCURACY DESIRED, ALL
     C ....   CALCULATIONS MUST BE DONE IN DOUBLE PRECISION.  TO DO THIS,
     C ....   REMOVE THE C IN COLUMN ONE FROM THE FOLLOWING STATEMENT.  IN
     C ....   THIS CASE, ITERATIVE IMPROVEMENT OF THE SOLUTIONS WOULD BE
     C ....   INEFFECTIVE AND SHOULD NOT BE REQUESTED.
     C      DOUBLE PRECISION A,R,RHO,SA,RHOM,SALPH,EPS,DENOM,DOT,ALPHA,
     C     1VZERO,ONE,TMAX,TRY,TEMP,EPS1,BASIC,QSQRT
     C
89        DIMENSION A(LA,1),RHO(NC),RHOM(NC),SA(NC),SALPH(LSAL,1),R(LRE,1),
         *BASIC(NRHS),SAVE(LSAV,1),XSAVE(NC,NRHS,1)
90        COMMON/BETA/EPS,EPS1,IPIV,ISW,NTRAC,NCOLS,KRBEG,KREND,KRANK,KROLD,
         *IREF,KW,NCP,IRANK,MAXIT,NFAIL,IRGT
     C
     C ....                 CHANGE QSQRT(Y)=SQRT(Y) TO QSQRT(Y)=DSQRT(Y)
     C ....                 IF ALL CALCULATIONS ARE BEING DONE IN DOUBLE
     C ....                 PRECISION.
91        QSQRT(Y)=SQRT(Y)
     C
92        EPSQ=EPS*EPS
93        ONE=1.
94        HUGE=1.E50
95        VZERO=0.
     C ....                 CHECK TO SEE IF THIS IS THE FIRST TIME THE
     C ....                 SUBROUTINE HAS BEEN ENTERED.
96        IF(KROLD)30,30,10
97     10 KROP=KROLD+1
     C ....                 IF THE SUBROUTINE HAS BEEN ENTERED PREVIOUSLY,
     C ....                 RETURN THE AUGMENTED MATRIX TO THE STATE IT WAS
     C ....                 IN PRIOR TO ORTHOGONALIZING THE LAST NC-KROLD
     C ....                 COLUMNS OF R.
98        DO 20 IR=1,NC
99        DO 20 IC=KROP,NCOLS
100    20 R(IR,IC)=SALPH(IR,IC)
101       K=KROP
102       GO TO 80
     C ....                 SET UP AN NC BY NCOLS MATRIX R SUCH THAT THE
     C ....                 FIRST NC COLUMNS FORM AN NC BY NC IDENTITY
     C ....                 MATRIX AND THE LAST NRHS COLUMNS ARE ZERO
     C ....                 VECTORS.
103    30 DO 50 I=1,NC
104       DO 40 J=1,NCOLS
105    40 R(I,J)=VZERO
```

```
   50 R(I,I)=ONE                                                        LLSQ0600
C ....                   SET UP A VECTOR RHO SUCH THAT RHO(J) IS THE    LLSQ0610
C ....                   SQUARE OF THE EUCLIDEAN NORM OF THE J-TH COLUMN LLSQ0620
C ....                   OF A.                                          LLSQ0630
      DO 70 J=1,NC                                                      LLSQ0640
      DOT=VZERO                                                         LLSQ0650
      DO 60 I=1,NR                                                      LLSQ0660
   60 DOT=DOT+A(I,J)*A(I,J)                                             LLSQ0670
      RHO(J)=DOT                                                        LLSQ0680
C ....                   INITIALIZE SA(J) TO RHO(J).                    LLSQ0690
C ....                   J=1,NC.                                        LLSQ0700
   70 SA(J)=RHO(J)                                                      LLSQ0710
C ....                   INITIALIZE A POINTER K.                        LLSQ0720
      K=1                                                               LLSQ0730
C ....                   TEST FOR THE COMPLETION OF THE TRANSFORMATION OFLLSQ0740
C ....                   A INTO A MATRIX WHOSE NONZERO COLUMNS ARE      LLSQ0750
C ....                   ORTHOGONAL.                                    LLSQ0760
   80 IF(K-NC)90,90,710                                                 LLSQ0770
   90 IF(K-KRANK)100,100,300                                            LLSQ0780
  100 MAXP=K                                                            LLSQ0790
C ....                   SEARCH FOR THE PIVOTAL COLUMN OF A.            LLSQ0800
      KP1=K+1                                                           LLSQ0810
      IF(IPIV)110,110,200                                               LLSQ0820
  110 IF(SA(K))120,120,130                                              LLSQ0830
  120 TMAX=VZERO                                                        LLSQ0840
      GO TO 140                                                         LLSQ0850
  130 TMAX=RHO(K)/SA(K)                                                 LLSQ0860
C ....                   IF IPIV.EQ.-1, NO PIVOTING IS PERFORMED.       LLSQ0870
  140 IF(IPIV)290,150,150                                               LLSQ0880
  150 IF(K-NC)160,290,290                                               LLSQ0890
  160 DO 190 I=KP1,NC                                                   LLSQ0900
      IF(SA(I))190,190,170                                              LLSQ0910
  170 TRY=RHO(I)/SA(I)                                                  LLSQ0920
      IF(TRY-TMAX)190,190,180                                           LLSQ0930
  180 TMAX=TRY                                                          LLSQ0940
      MAXP=I                                                            LLSQ0950
  190 CONTINUE                                                          LLSQ0960
      GO TO 290                                                         LLSQ0970
C ....                   IF IPIV.EQ.1, IVOR, INDEPENDENT VARIABLE       LLSQ0980
C ....                   ORDERING BY REGRESSION SUM OF SQUARES, IS      LLSQ0990
C ....                   PERFORMED.                                     LLSQ1000
  200 TMAX=VZERO                                                        LLSQ1010
      TMIN=HUGE                                                         LLSQ1020
      DO 270 J=K,NC                                                     LLSQ1030
      IF(RHO(J)-SA(J)*EPSQ)270,270,210                                  LLSQ1040
  210 DOT=VZERO                                                         LLSQ1050
      DO 220 I=1,NR                                                     LLSQ1060
  220 DOT=DOT+A(I,J)*A(I,NCP)                                           LLSQ1070
      DOT=DOT/RHO(J)                                                    LLSQ1080
      SUM=VZERO                                                         LLSQ1090
      DO 230 I=1,NR                                                     LLSQ1100
      AUX=A(I,NCP)-DOT*A(I,J)                                           LLSQ1110
  230 SUM=SUM+AUX*AUX                                                   LLSQ1120
      IF(NTRAC-2)250,240,250                                            LLSQ1130
  240 WRITE(KW,840)J,SUM                                                LLSQ1140
  250 IF(SUM-TMIN)260,270,270                                           LLSQ1150
  260 TMIN=SUM                                                          LLSQ1160
      MAXP=J                                                            LLSQ1170
  270 CONTINUE                                                          LLSQ1180
      IF(SA(MAXP))300,300,280                                           LLSQ1190
```

```
  280 TMAX=RHO(MAXP)/SA(MAXP)                                          LLSQ1200
C ....                  IF THE SQUARE OF THE NORM OF THE PIVOTAL       LLSQ1210
C ....                  COLUMN DIVIDED BY THE SQUARE OF THE NORM OF    LLSQ1220
C ....                  THAT COLUMN PRIOR TO ELEMENTARY COLUMN         LLSQ1230
C ....                  OPERATIONS BEING PERFORMED ON IT IS LESS THAN  LLSQ1240
C ....                  OR EQUAL TO THE SQUARE OF EPS, THE RANK OF A IS LLSQ1250
C ....                  DETERMINED TO BE K-1.                          LLSQ1260
  290 IF(TMAX-EPSQ)300,300,490                                         LLSQ1270
  300 IRANK=K-1                                                        LLSQ1280
      NO=KRANK-KRBEG+1                                                 LLSQ1290
C ....                  PRINT THE RANK.                                LLSQ1300
      IF(NTRAC)310,310,310                                             LLSQ1310
  310 WRITE(KW,810)IRANK                                               LLSQ1320
      DO 320 I=1,NC                                                    LLSQ1330
      DO 320 J=NCP,NCOLS                                               LLSQ1340
      JJ=J-NC                                                          LLSQ1350
  320 XSAVE(I,JJ,NO)=R(I,J)                                            LLSQ1360
C ....                  CALCULATE THE LENGTH OF THE RESIDUAL VECTOR    LLSQ1370
C ....                  FOR EACH BASIC APPROXIMATE SOLUTION.           LLSQ1380
      IF(IREF)370,330,330                                              LLSQ1390
  330 DO 360 IND1=NCP,NCOLS                                            LLSQ1400
      NO=IND1-NC                                                       LLSQ1410
      DOT=VZERO                                                        LLSQ1420
      DO 350 IR=1,NR                                                   LLSQ1430
      DRES=VZERO                                                       LLSQ1440
      DO 340 IC=1,NC                                                   LLSQ1450
      DINT=R(IC,IND1)                                                  LLSQ1460
      DINTP=SAVE(IR,IC)                                                LLSQ1470
  340 DRES=DRES+DINTP*DINT                                             LLSQ1480
      DSAVE=SAVE(IR,IND1)                                              LLSQ1490
      DIF=DSAVE-DRES                                                   LLSQ1500
  350 DOT=DOT+DIF*DIF                                                  LLSQ1510
  360 BASIC(NO)=QSQRT(DOT)                                             LLSQ1520
C ....                  SAVE THE STATE OF THE SYSTEM AFTER FINDING THE LLSQ1530
C ....                  BASIC SOLUTIONS.                               LLSQ1540
  370 IF(KREND-KRBEG)400,400,380                                       LLSQ1550
  380 DO 390 IR=1,NC                                                   LLSQ1560
      DO 390 IC=K,NCOLS                                                LLSQ1570
  390 SALPH(IR,IC)=R(IR,IC)                                            LLSQ1580
  400 IF(IRANK-KROLD)420,410,420                                       LLSQ1590
  410 WRITE(KW,800)                                                    LLSQ1600
      IRGT=1                                                           LLSQ1610
      GO TO 740                                                        LLSQ1620
C ....                  CHECK IF THE BASIC SOLUTION IS TO BE PRINTED.  LLSQ1630
  420 IF(NTRAC)460,430,430                                             LLSQ1640
  430 DO 450 L=NCP,NCOLS                                               LLSQ1650
      NO=L-NC                                                          LLSQ1660
      WRITE(KW,790)NO                                                  LLSQ1670
      WRITE(KW,780)(R(IR,L),IR=1,NC)                                   LLSQ1680
      DOT=VZERO                                                        LLSQ1690
      DO 440 IC=1,NC                                                   LLSQ1700
  440 DOT=DOT+R(IC,L)*R(IC,L)                                          LLSQ1710
      DOT=QSQRT(DOT)                                                   LLSQ1720
      WRITE(KW,820)DOT                                                 LLSQ1730
      IF(IREF)460,450,450                                              LLSQ1740
  450 SQLEN=BASIC(NO)*BASIC(NO)                                        LLSQ1750
      WRITE(KW,830)NO,BASIC(NO),SQLEN                                  LLSQ1760
  460 IF(IRANK)470,470,660                                             LLSQ1770
  470 DO 480 J=1,NC                                                    LLSQ1780
  480 RHOM(J)=ONE                                                      LLSQ1790
```

```
      GO TO 720                                                         LLSQ1800
C ....                  IF MAXP, THE INDEX OF THE PIVOTAL COLUMN, IS K, LLSQ1810
C ....                  DO NOTHING.  OTHERWISE INTERCHANGE THE MAXP-TH  LLSQ1820
C ....                  AND K-TH COLUMNS OF A AND R, AND THE MAXP-TH    LLSQ1830
C ....                  AND K-TH COMPONENTS OF RHO AND SA.              LLSQ1840
  490 IF(MAXP-K)500,530,500                                             LLSQ1850
  500 DO 510 I=1,NR                                                     LLSQ1860
      TEMP=A(I,K)                                                       LLSQ1870
      A(I,K)=A(I,MAXP)                                                  LLSQ1880
  510 A(I,MAXP)=TEMP                                                    LLSQ1890
      DO 520 I=1,NC                                                     LLSQ1900
      TEMP=R(I,K)                                                       LLSQ1910
      R(I,K)=R(I,MAXP)                                                  LLSQ1920
  520 R(I,MAXP)=TEMP                                                    LLSQ1930
      TEMP=RHO(K)                                                       LLSQ1940
      RHO(K)=RHO(MAXP)                                                  LLSQ1950
      RHO(MAXP)=TEMP                                                    LLSQ1960
      TEMP=SA(K)                                                        LLSQ1970
      SA(K)=SA(MAXP)                                                    LLSQ1980
      SA(MAXP)=TEMP                                                     LLSQ1990
C ....                  USE THE MODIFIED GRAM-SCHMIDT PROCESS TO        LLSQ2000
C ....                  ORTHOGONALIZE THE NONZERO COLUMNS OF A.         LLSQ2010
  530 KP1=K+1                                                           LLSQ2020
      DO 640 I=KP1,NCOLS                                                LLSQ2030
      IF(I-NCP)540,550,550                                              LLSQ2040
  540 IF(RHO(I)-EPSQ*SA(I))640,640,550                                  LLSQ2050
  550 DOT=VZERO                                                         LLSQ2060
      DO 560 J=1,NR                                                     LLSQ2070
  560 DOT=DOT+A(J,K)*A(J,I)                                             LLSQ2080
C ....                  DIVIDE THE CALCULATED INNER PRODUCT BY THE      LLSQ2090
C ....                  SQUARE OF THE NORM OF THE K-TH COLUMN OF A.     LLSQ2100
      IF(RHO(K))300,300,570                                             LLSQ2110
  570 ALPHA=DOT/RHO(K)                                                  LLSQ2120
C ....                  SUBTRACT MULTIPLES OF THE K-TH COLUMNS OF A AND LLSQ2130
C ....                  R FROM THE I-TH COLUMNS.                        LLSQ2140
      DO 580 J=1, NR                                                    LLSQ2150
  580 A(J,I)=A(J,I)-ALPHA*A(J,K)                                        LLSQ2160
      DO 590 J=1,NC                                                     LLSQ2170
  590 R(J,I)=R(J,I)-ALPHA*R(J,K)                                        LLSQ2180
C ....                  RECALCULATE THE SQUARE OF THE NORM OF THE I-TH  LLSQ2190
C ....                  COLUMN OF A.                                    LLSQ2200
      IF(I-NCP)600,640,640                                              LLSQ2210
  600 IF(ISW-1)610,610,620                                              LLSQ2220
  610 RHO(I)=RHO(I)-ALPHA*ALPHA*RHO(K)                                  LLSQ2230
      GO TO 640                                                         LLSQ2240
  620 DOT=VZERO                                                         LLSQ2250
      DO 630 J=1,NR                                                     LLSQ2260
  630 DOT=DOT+A(J,I)*A(J,I)                                             LLSQ2270
      RHO(I)=DOT                                                        LLSQ2280
  640 CONTINUE                                                          LLSQ2290
C ....                  INCREMENT THE COLUMN COUNTER, K.                LLSQ2300
      K=K+1                                                             LLSQ2310
      GO TO 80                                                          LLSQ2320
C ....                  CALCULATE THE SQUARE OF THE NORM OF THE K-TH    LLSQ2330
C ....                  COLUMN OF R.                                    LLSQ2340
  650 IF(K-NC)660,660,740                                               LLSQ2350
  660 DENOM=VZERO                                                       LLSQ2360
      KP1=K+1                                                           LLSQ2370
      DO 670 I=1,NC                                                     LLSQ2380
  670 DENOM=DENOM+R(I,K)*R(I,K)                                         LLSQ2390
```

```
      RHOM(K)=DENOM                                                     LLSQ2400
C ....                    CALCULATE THE INNER PRODUCT OF THE I-TH COLUMN  LLSQ2410
C ....                    OF R WITH THE K-TH COLUMN OF R, I=K+1,..,NCOLS. LLSQ2420
      DO 700 I=KP1,NCOLS                                                LLSQ2430
      DOT=VZERO                                                         LLSQ2440
      DO 680 J=1,NC                                                     LLSQ2450
  680 DOT=DOT+R(J,I)*R(J,K)                                             LLSQ2460
C ....                    SUBTRACT MULTIPLES OF THE K-TH COLUMN OF R      LLSQ2470
C ....                    FROM THE I-TH COLUMN OF R.                      LLSQ2480
      ALPHA=DOT/DENOM                                                   LLSQ2490
      DO 690 J=1,NC                                                     LLSQ2500
  690 R(J,I)=R(J,I)-ALPHA*R(J,K)                                        LLSQ2510
  700 CONTINUE                                                          LLSQ2520
      K=K+1                                                             LLSQ2530
      GO TO 650                                                         LLSQ2540
C ....                    THE RANK OF A IS KRANK.                         LLSQ2550
  710 IRANK=KRANK                                                       LLSQ2560
  720 IF(NTRAC)740,730,730                                              LLSQ2570
  730 WRITE(KW,810)IRANK                                                LLSQ2580
  740 IF(KRANK-NC)770,750,770                                           LLSQ2590
  750 NO=NC-KRBEG+1                                                     LLSQ2600
      DO 760 I=1,NC                                                     LLSQ2610
      DO 760 J=1,NRHS                                                   LLSQ2620
      JNC=J+NC                                                          LLSQ2630
  760 XSAVE(I,J,NO)=R(I,JNC)                                            LLSQ2640
  770 RETURN                                                            LLSQ2650
C ....   CHANGE 5E20.7 TO 4D25.14 IF USING DOUBLE PRECISION               LLSQ2660
C ....   CALCULATIONS.                                                    LLSQ2670
  780 FORMAT(/(1X,5E20.7))                                              LLSQ2680
  790 FORMAT(/47H THE BASIC SOLUTION FOR RIGHT HAND SIDE NUMBER ,I5)    LLSQ2690
  800 FORMAT(/58H THE RANK REQUESTED IS GREATER THAN THE RANK OF THE SYSLLSQ2720
     *TEM)                                                              LLSQ2730
  810 FORMAT(/40H THE RANK OF THE SYSTEM OF EQUATIONS IS ,I10)          LLSQ2740
  820 FORMAT(/44H THE LENGTH OF THE BASIC SOLUTION VECTOR IS ,E20.7)    LLSQ2750
  830 FORMAT(/85H THE LENGTH OF THE RESIDUAL VECTOR FOR THE BASIC SOLUTILLSQ2760
     *ON FOR RIGHT HAND SIDE NUMBER ,I5,4H IS ,E20.7/75H THE SQUARE OF TLLSQ2770
     *HE LENGTH OF THE RESIDUAL VECTOR FOR THE BASIC SOLUTION IS ,      LLSQ2780
     *E20.7)                                                            LLSQ2790
  840 FORMAT(/21H RESIDUAL FOR COLUMN ,I5,4H IS ,E20.7)                 LLSQ2800
      END                                                               LLSQ2810
```

APPENDIX B

SAMPLE OUTPUT FROM THE PROBLEM CONSISTING OF THE FIRST FIVE COLUMNS OF A 6 BY 6 INVERSE HILBERT MATRIX AND A RIGHT HAND SIDE CHOSEN TO GENERATE THE SOLUTION VECTOR

$$(1., \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5})$$

EPSI= 0.160E-05 EPS= 0.160E-05 NR= 6 NC= 5 ISW= 2
IREF= 1 MAXIT= 10 KRBEG= 1 KREND= 5

THE ORIGINAL COEFFICIENT MATRIX

0.3600000E 02	-0.6300000E 03	0.3360000E 04	-0.7560000E 04	0.7560000E 04
-0.6300000E 03	0.1470000E 05	-0.8820000E 05	0.2116800E 06	-0.2205000E 06
0.3360000E 04	-0.8820000E 05	0.5644800E 06	-0.1411200E 07	0.1512000E 07
-0.7560000E 04	0.2116800E 06	-0.1411200E 07	0.3628800E 07	-0.3969000E 07
0.7560000E 04	-0.2205000E 06	0.1512000E 07	-0.3969000E 07	0.4410000E 07
-0.2772000E 04	0.8316000E 05	-0.5821200E 06	0.1552320E 07	-0.1746360E 07

THE ORIGINAL RIGHT HAND SIDES

0.4630000E 03	-0.1386000E 05	0.9702000E 05	-0.2587200E 06	0.2910600E 06
-0.1164240E 06				

THE RANK OF THE SYSTEM OF EQUATIONS IS 1

THE BASIC SOLUTION FOR RIGHT HAND SIDE NUMBER 1

0.3601010E 02	-0.0000000E 00	-0.0000000E 00	-0.0000000E 00	-0.0000000E 00

THE LENGTH OF THE BASIC SOLUTION VECTOR IS 0.3601009E 02

THE LENGTH OF THE RESIDUAL VECTOR FOR THE BASIC SOLUTION FOR RIGHT HAND SIDE NUMBER 1 IS 0.3828919E 05
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE BASIC SOLUTION IS 0.1466062E 10

THE DECOMPOSITION MATRIX FOR RANK EQUAL TO 1

0.3600000E 02	0.3942717E 03	-0.3551730E 04	0.1038025E 05	-0.1220827E 05
-0.6300000E 03	-0.3224754E 04	0.3275525E 05	-0.1022744E 06	0.1254448E 06
0.3360000E 04	0.7398688E 04	-0.8061488E 05	0.2632230E 06	-0.3330380E 06
-0.7560000E 04	-0.3417063E 04	0.4026300E 05	-0.1386520E 06	0.1823370E 06
0.7560000E 04	-0.5402938E 04	0.6053700E 05	-0.2015480E 06	0.2586630E 06
-0.2772000E 04	0.4291125E 04	-0.4991675E 05	0.1709210E 06	-0.2242040E 06
0.1000000E 01	0.2845200E 02	-0.2368622E 00	0.1341271E-01	-0.2026308E-02
0.0000000E 00	0.1000000E 01	0.6739621E 01	-0.3763733E 00	0.5461901E-01
0.0000000E 00	-0.0000000E 00	0.1000000E 01	0.2539779E 01	-0.3686066E 00
0.0000000E 00	-0.0000000E 00	-0.0000000E 00	0.1000000E 01	0.9567620E 00
0.0000000E 00	-0.0000000E 00	-0.0000000E 00	-0.0000000E 00	0.1000000E 01

START ITERATING THE BASIC SOLUTION

```
ITERATION    0

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      -0.8333635E 03       0.8826363E 04      -0.2397394E 05       0.1351636E 05       0.1882363E 05
      -0.1660400E 05

THE LENGTH OF THE RESIDUAL VECTOR IS        0.3828919E 05
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.1466062E 10

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
       0.3601010E 02      -0.0000000E 00      -0.0000000E 00      -0.0000000E 00      -0.0000000E 00

ITERATION    1

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      -0.8333652E 03       0.8826391E 04      -0.2397409E 05       0.1351671E 05       0.1882329E 05
      -0.1660387E 05

THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.3828919E 05
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.1466062E 10

THE BASIC APPROXIMATE SOLUTION IS A BETTER SOLUTION THAN THE FINAL SOLUTION

THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SIDE NUMBER      1
       0.3601015E 02       0.0000000E 00       0.0000000E 00       0.0000000E 00       0.0000000E 00

THE LENGTH OF THE FINAL SOLUTION VECTOR IS        0.3601013E 02


START ITERATING THE MINIMUM NORM SOLUTION

ITERATION    0

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      -0.6297304E 02       0.1089738E 04      -0.3764518E 04       0.2665347E 04       0.3240679E 04
      -0.3243177E 04

THE LENGTH OF THE RESIDUAL VECTOR IS        0.6594492E 04
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.4348734E 08

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
       0.6908472E-04      -0.1744074E-02       0.1176700E-01      -0.3054278E-01       0.3365493E-01

ITERATION    1

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      -0.6297322E 02       0.1089743E 04      -0.3764552E 04       0.2665438E 04       0.3240580E 04
      -0.3243138E 04

THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.6594480E 04
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.4348718E 08

THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SIDE NUMBER      1
       0.6908474E-04      -0.1744075E-02       0.1176700E-01      -0.3054279E-01       0.3365494E-01

THE LENGTH OF THE FINAL SOLUTION VECTOR IS        0.4697899E-01

THE RANK OF THE SYSTEM OF EQUATIONS IS          2
```

THE BASIC SOLUTION FOR RIGHT HAND SIDE NUMBER 1

-0.3918518E 01 -0.0000000E 00 -0.0000000E 00 -0.0000000E 00 0.7271403E-01

THE LENGTH OF THE BASIC SOLUTION VECTOR IS 0.3919192E 01

THE LENGTH OF THE RESIDUAL VECTOR FOR THE BASIC SOLUTION FOR RIGHT HAND SIDE NUMBER 1 IS 0.5536350E 03
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE BASIC SOLUTION IS 0.3065117E 06

THE DECOMPOSITION MATRIX FOR RANK EQUAL TO 2

0.3600000E 02	-0.1220827E 05	-0.6808057E 03	0.8523711E 03	0.1363816E 03
-0.6300000E 03	0.1254448E 06	0.3255375E 04	-0.4371688E 04	-0.5748318E 03
0.3360000E 04	-0.3330380E 06	-0.2296938E 04	0.3305250E 04	0.3635234E 03
-0.7560000E 04	0.1823370E 06	-0.2615789E 04	0.3651938E 04	0.4346638E 03
0.7560000E 04	0.2586630E 06	-0.2907891E 03	0.3241250E 03	0.6111328E 02
-0.2772000E 04	-0.2242040E 06	0.2807582E 04	-0.4057813E 04	-0.4450078E 03
0.1000000E 01	-0.5491187E 03	-0.6286052E 02	0.1585388E-01	0.2677550E-02
0.0000000E 00	-0.0000000E 00	-0.0000000E 00	-0.0000000E 00	0.1000000E 01
0.0000000E 00	-0.0000000E 00	0.1000000E 01	0.1109883E 01	0.1456825E 00
0.0000000E 00	-0.0000000E 00	-0.0000000E 00	0.1000000E 01	-0.1102253E 00
0.0000000E 00	0.1000000E 01	-0.2351623E 00	0.5194420E 00	-0.9915954E-01

START ITERATING THE BASIC SOLUTION

ITERATION 0

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER 1 IS
0.5434857E 02 -0.2952224E 03 0.2426062E 03 0.2579912E 03 0.1512101E 02
-0.3012573E 03

THE LENGTH OF THE RESIDUAL VECTOR IS 0.5536350E 03
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS 0.3065118E 06

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER 1
-0.3918518E 01 -0.0000000E 00 -0.0000000E 00 -0.0000000E 00 0.7271403E-01

ITERATION 1

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER 1 IS
0.5434805E 02 -0.2952134E 03 0.2425582E 03 0.2580994E 03 0.1501286E 02
-0.3012175E 03

THE LENGTH OF THE RESIDUAL VECTOR IS 0.5536348E 03
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS 0.3065116E 06

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER 1

```
      -0.3918504E 01       0.0000000E 00        0.0000000E 00        0.0000000E 00        0.7271403E-01

ITERATION   2

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
       0.5434805E 02      -0.2952134E 03        0.2425582E 03        0.2580994E 03        0.1501286E 02
      -0.3012175E 03

THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.5536348E 03
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.3065116E 06

THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SIDE NUMBER      1
      -0.3918504E 01       0.0000000E 00        0.0000000E 00        0.0000000E 00        0.7271403E-01

THE LENGTH OF THE FINAL SOLUTION VECTOR IS        0.3919178E 01


START ITERATING THE MINIMUM NORM SOLUTION

ITERATION   0

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
       0.1196937E 02      -0.9165155E 02        0.9815402E 02        0.9341266E 02       -0.2290949E 01
      -0.1245328E 03

THE LENGTH OF THE RESIDUAL VECTOR IS        0.2059540E 03
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.4241705E 05

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
      -0.1122564E-02       0.1701020E-01       -0.4927908E-01        0.7638268E-02        0.9062308E-01

ITERATION   1

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
       0.1196851E 02      -0.9162877E 02        0.9800777E 02        0.9377876E 02       -0.2683254E 01
      -0.1243818E 03

THE LENGTH OF THE RESIDUAL VECTOR IS        0.2059540E 03
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.4241707E 05

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
      -0.1122562E-02       0.1701017E-01       -0.4927897E-01        0.7638149E-02        0.9062302E-01

ITERATION   2

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
       0.1196846E 02      -0.9162712E 02        0.9799654E 02        0.9380801E 02       -0.2715523E 01
      -0.1243691E 03

THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.2059540E 03
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.4241707E 05

THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SIDE NUMBER      1
      -0.1122562E-02       0.1701017E-01       -0.4927897E-01        0.7638138E-02        0.9062302E-01

THE LENGTH OF THE FINAL SOLUTION VECTOR IS        0.1048326E 00

THE RANK OF THE SYSTEM OF EQUATIONS IS          3
```

THE BASIC SOLUTION FOR RIGHT HAND SIDE NUMBER 1

0.5984886E 01 0.5876597E 00 -0.0000000E 00 -0.0000000E 00 0.8512783E-01

THE LENGTH OF THE BASIC SOLUTION VECTOR IS 0.6014267E 01

THE LENGTH OF THE RESIDUAL VECTOR FOR THE BASIC SOLUTION FOR RIGHT HAND SIDE NUMBER 1 IS 0.7305113E 02
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE BASIC SOLUTION IS 0.5336465E 04

THE DECOMPOSITION MATRIX FOR RANK EQUAL TO 3

0.3600000E 02	-0.1220827E 05	0.1363816E 03	-0.2807053E 03	0.1331047E 03
-0.6300000E 03	0.1254448E 06	-0.5748318E 03	0.4040898E 03	-0.1751582E 03
0.3360000E 04	-0.3330380E 06	0.3635234E 03	0.2850486E 03	-0.1274695E 03
-0.7560000E 04	0.1823370E 06	0.4346638E 03	0.4069238E 02	-0.2176294E 02
0.7560000E 04	0.2586630E 06	0.6111328E 02	-0.1836123E 03	0.7392822E 02
-0.2772000E 04	-0.2242040E 06	-0.4450078E 03	-0.3606279E 03	0.1518242E 03
0.1000000E 01	-0.5491187E 03	0.1685228E 02	-0.7022729E 02	-0.1663666E 00
0.0000000E 00	-0.0000000E 00	0.1000000E 01	-0.8308134E 01	0.1486743E 01
0.0000000E 00	-0.0000000E 00	-0.0000000E 00	-0.0000000E 00	0.1000000E 01
0.0000000E 00	-0.0000000E 00	-0.0000000E 00	0.1000000E 01	0.5393688E 00
0.0000000E 00	0.1000000E 01	0.2112422E-01	0.6049420E 00	0.2171915E 00

START ITERATING THE BASIC SOLUTION

ITERATION 0

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER 1 IS
-0.2579671E 02 0.4256795E 02 0.2908449E 02 0.2302523E 01 -0.2051759E 02
-0.3983452E 02

THE LENGTH OF THE RESIDUAL VECTOR IS 0.7305113E 02
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS 0.5336469E 04

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER 1
0.5984886E 01 0.5876597E 00 -0.0000000E 00 -0.0000000E 00 0.8512783E-01

ITERATION 1

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER 1 IS
-0.2580716E 02 0.4262851E 02 0.2894385E 02 0.2535038E 01 -0.2078429E 02
-0.3970796E 02

THE LENGTH OF THE RESIDUAL VECTOR IS 0.7304846E 02
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS 0.5336078E 04

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER 1
0.5986060E 01 0.5877273E 00 0.0000000E 00 0.0000000E 00 0.8512926E-01

```
ITERATION   2

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      -0.2580716E 02       0.4262851E 02       0.2894385E 02       0.2535038E 01       -0.2078429E 02
      -0.3970796E 02

THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.7304846E 02
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.5336078E 04

THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SIDE NUMBER      1
       0.5986060E 01       0.5877273E 00       0.0000000E 00       0.0000000E 00       0.8512926E-01

THE LENGTH OF THE FINAL SOLUTION VECTOR IS        0.6015442E 01


START ITERATING THE MINIMUM NORM SOLUTION

ITERATION   0

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      -0.1412092E 01       0.7000010E 01       0.4034684E 01      -0.9714581E 00       -0.4376644E 01
      -0.7957222E 01

THE LENGTH OF THE RESIDUAL VECTOR IS        0.1227554E 02
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.1506889E 03

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
       0.1159034E-01      -0.7548279E-01       0.2888909E-01       0.1005700E 00       0.1428152E 00

ITERATION   1

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      -0.1419850E 01       0.7070168E 01       0.3861511E 01      -0.6887278E 00       -0.4759110E 01
      -0.7739984E 01

THE LENGTH OF THE RESIDUAL VECTOR IS        0.1224521E 02
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.1499453E 03

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
       0.1159722E-01      -0.7553279E-01       0.2893130E-01       0.1006204E 00       0.1428436E 00

ITERATION   2

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      -0.1419912E 01       0.7071811E 01       0.3850997E 01      -0.6624421E 00       -0.4787273E 01
      -0.7729140E 01

THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.1224554E 02
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.1499532E 03

THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SIDE NUMBER      1
       0.1159722E-01      -0.7553279E-01       0.2893132E-01       0.1006204E 00       0.1428436E 00

THE LENGTH OF THE FINAL SOLUTION VECTOR IS        0.1928872E 00

THE RANK OF THE SYSTEM OF EQUATIONS IS        4

THE BASIC SOLUTION FOR RIGHT HAND SIDE NUMBER      1
```

```
     -0.2953621E 01      -0.8268492E 00      -0.2370200E 00      -0.0000000E 00      0.1109856E 00

THE LENGTH OF THE BASIC SOLUTICN VECTOR IS        0.3078319E 01

THE LENGTH OF THE RESIDUAL VECTOR FOR THE BASIC SOLUTION FOR RIGHT HAND SIDE NUMBER     1 IS       0.8257427E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE BASIC SOLUTION IS       0.6818509E 02

THE DECOMPOSITION MATRIX FOR RANK EQUAL TO      4

      0.3600000E 02      -0.1220827E 05      0.1363816E 03      0.1331047E 03      0.2305908E 02

     -0.6300000E 03       0.1254448E 06     -0.5748318E 03     -0.1751582E 03      0.4353271E 01

      0.3360000E 04      -0.3330380E 06      0.3635234E 03     -0.1274695E 03     -0.5855225E 01

     -0.7560000E 04       0.1823370E 06      0.4346638E 03     -0.2176294E 02     -0.8973831E 01

      0.7560000E 04       0.2586630E 06      0.6111328E 02      0.7392822E 02     -0.1489728E 02

     -0.2772000E 04      -0.2242040E 06     -0.4450078E 03      0.1518242E 03     -0.1414307E 02

      0.1000000E 01      -0.5491187E 03      0.1685228E 02      0.3771204E 02      0.1583708E 02

      0.0000000E 00      -0.0000000E 00      0.1000000E 01      0.5967891E 01      0.5311466E 01

      0.0000000E 00      -0.0000000E 00     -0.0000000E 00      0.1000000E 01      0.2282146E 01

      0.0000000E 00      -0.0000000E 00     -0.0000000E 00     -0.0000000E 00      0.1000000E 01

      0.0000000E 00       0.1000000E 01      0.2112422E-01     -0.1090953E 00      0.3559704E 00


START ITERATING THE BASIC SOLUTION

ITERATION    0

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      0.5751670E 01       0.1055882E 01     -0.1063993E 01     -0.2833993E 01     -0.2997540E 01
     -0.3979918E 01

THE LENGTH OF THE RESIDUAL VECTOR IS        0.8257427E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.6818512E 02

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
     -0.2953621E 01      -0.8268492E 00      -0.2370200E 00     -0.0000000E 00      0.1109856E 00

ITERATION    1

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      0.5795353E 01       0.1082843E 01     -0.1307373E 01     -0.2626111E 01     -0.3283153E 01
     -0.3707628E 01

THE LENGTH OF THE RESIDUAL VECTOR IS        0.8239532E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.6788991E 02

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
     -0.2969888E 01      -0.8297057E 00      -0.2375268E 00      0.0000000E 00      0.1110445E 00
```

```
ITERATION   2

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER     1 IS
       0.5795730E 01       0.1074282E 01      -0.1251246E 01      -0.2769856E 01      -0.3126715E 01
      -0.3768431E 01

THE LENGTH OF THE RESIDUAL VECTOR IS         0.8243764E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS         0.6795966E 02

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
      -0.2969910E 01      -0.8297095E 00      -0.2375275E 00       0.0000000E 00       0.1110445E 00

ITERATION   3

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER     1 IS
       0.5795730E 01       0.1074282E 01      -0.1251246E 01      -0.2769856E 01      -0.3126715E 01
      -0.3768431E 01

THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS         0.8243764E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS         0.6795966E 02

THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SIDE NUMBER      1
      -0.2969910E 01      -0.8297095E 00      -0.2375275E 00       0.0000000E 00       0.1110445E 00

THE LENGTH OF THE FINAL SOLUTION VECTOR IS         0.3094758E 01


START ITERATING THE MINIMUM NORM SOLUTION

ITERATION   0

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER     1 IS
       0.1576748E 01       0.2459836E 00       0.1041412E-01      -0.1475101E 01      -0.1892567E-01
      -0.1467851E 01

THE LENGTH OF THE RESIDUAL VECTOR IS         0.2622519E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS         0.6877613E 01

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
      -0.8600903E-01       0.1348953E 00       0.1762071E 00       0.1810695E 00       0.1754409E 00

ITERATION   1

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER     1 IS
       0.1586388E 01       0.2919084E 00      -0.3293967E 00      -0.7901716E 00      -0.8221400E 00
      -0.1044050E 01

THE LENGTH OF THE RESIDUAL VECTOR IS         0.2258464E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS         0.5100661E 01

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
      -0.8656365E-01       0.1360603E 00       0.1770899E 00       0.1815757E 00       0.1756532E 00

ITERATION   2

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER     1 IS
       0.1586201E 01       0.2967024E 00      -0.3611898E 00      -0.7083106E 00      -0.9115863E 00
      -0.1009189E 01
```

THE LENGTH OF THE RESIDUAL VECTOR IS 0.2255144E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS 0.5085675E 01

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER 1
-0.8656299E-01 0.1360589E 00 0.1770889E 00 0.1815751E 00 0.1756529E 00

ITERATION 3

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER 1 IS
0.1586201E 01 0.2967024E 00 -0.3611898E 00 -0.7083106E 00 -0.9115863E 00
-0.1009189E 01

THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS 0.2255144E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS 0.5085675E 01

THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SIDE NUMBER 1
-0.8656299E-01 0.1360589E 00 0.1770889E 00 0.1815751E 00 0.1756529E 00

THE LENGTH OF THE FINAL SOLUTION VECTOR IS 0.3481222E 00

THE RANK OF THE SYSTEM OF EQUATIONS IS 5

THE DECOMPOSITION MATRIX FOR RANK EQUAL TO 5

0.3600000E 02	-0.1220827E 05	0.1363816E 03	0.1331047E 03	0.2305908E 02
-0.6300000E 03	0.1254448E 06	-0.5748318E 03	-0.1751582E 03	0.4353271E 01
0.3360000E 04	-0.3330380E 06	0.3635234E 03	-0.1274695E 03	-0.5855225E 01
-0.7560000E 04	0.1823370E 06	0.4346638E 03	-0.2176294E 02	-0.8973831E 01
0.7560000E 04	0.2586630E 06	0.6111328E 02	0.7392822E 02	-0.1489728E 02
-0.2772000E 04	-0.2242040E 06	-0.4450078E 03	0.1518242E 03	-0.1414307E 02
0.1000000E 01	-0.5491187E 03	0.1685228E 02	0.3771204E 02	0.1583708E 02
0.0000000E 00	-0.0000000E 00	0.1000000E 01	0.5967891E 01	0.5311466E 01
0.0000000E 00	-0.0000000E 00	-0.0000000E 00	0.1000000E 01	0.2282146E 01
0.0000000E 00	-0.0000000E 00	-0.0000000E 00	-0.0000000E 00	0.1000000E 01
0.0000000E 00	0.1000000E 01	0.2112422E-01	-0.1090953E 00	0.3559704E 00

START ITERATING THE BASIC SOLUTION

ITERATION 0

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER 1 IS
0.5903137E-01 -0.2641082E-01 0.2598023E 00 -0.6255484E 00 0.6785452E 00
-0.4057903E 00

THE LENGTH OF THE RESIDUAL VECTOR IS 0.1043110E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS 0.1088079E 01

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER 1

```
       0.9558830E 00        0.4843262E 00        0.3263453E 00        0.2468576E 00        0.1988596E 00

ITERATION   1

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
       0.1099586E-02       -0.1439452E-01        0.9713173E-01       -0.2567983E 00        0.2835846E 00
      -0.1110306E 00

THE LENGTH OF THE RESIDUAL VECTOR IS        0.4102879E 00
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.1683362E 00

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
       0.9997082E 00        0.4999225E 00        0.3333055E 00        0.2499895E 00        0.1999967E 00

ITERATION   2

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      -0.2093315E-03        0.5882978E-02       -0.3905296E-01        0.1000857E 00       -0.1091981E 00
       0.4262781E-01

THE LENGTH OF THE RESIDUAL VECTOR IS        0.1591172E 00
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.2531829E-01

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
       0.9999990E 00        0.4999998E 00        0.3333333E 00        0.2499999E 00        0.2000000E 00

ITERATION   3

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
      -0.3291368E-03        0.9074807E-02       -0.5993128E-01        0.1531076E 00       -0.1666510E 00
       0.6493306E-01

THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.2431152E 00
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.5910502E-01

THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SIDE NUMBER      1
       0.9999999E 00        0.4999999E 00        0.3333333E 00        0.2499999E 00        0.2000000E 00

THE LENGTH OF THE FINAL SOLUTION VECTOR IS        0.1209797E 01


START ITERATING THE MINIMUM NORM SOLUTION

ITERATION   0

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
       0.5903137E-01       -0.2641082E-01        0.2598023E 00       -0.6255484E 00        0.6785452E 00
      -0.4057903E 00

THE LENGTH OF THE RESIDUAL VECTOR IS        0.1043110E 01
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.1088079E 01

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
       0.9558830E 00        0.4843262E 00        0.3263453E 00        0.2468576E 00        0.1988596E 00

ITERATION   1

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
       0.1099586E-02       -0.1439452E-01        0.9713173E-01       -0.2567983E 00        0.2835846E 00
```

```
     -0.1110306E 00

THE LENGTH OF THE RESIDUAL VECTOR IS        0.4102879E 00
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.1683362E 00

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
       0.9997082E 00      0.4999225E 00      0.3333055E 00      0.2499895E 00      0.1999967E 00

ITERATION   2

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
     -0.2093315E-03      0.5882978E-02     -0.3905296E-01      0.1000857E 00     -0.1091981E 00
      0.4262781E-01

THE LENGTH OF THE RESIDUAL VECTOR IS        0.1591172E 00
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR IS        0.2531829E-01

THE SOLUTION TO THE SYSTEM FOR RIGHT HAND SIDE NUMBER      1
       0.9999990E 00      0.4999998E 00      0.3333333E 00      0.2499999E 00      0.2000000E 00

ITERATION   3

THE RESIDUAL VECTOR FOR RIGHT HAND SIDE NUMBER      1 IS
     -0.3291368E-03      0.9074807E-02     -0.5993128E-01      0.1531076E 00     -0.1666510E 00
      0.6493306E-01

THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.2431152E 00
THE SQUARE OF THE LENGTH OF THE RESIDUAL VECTOR FOR THE FINAL SOLUTION IS        0.5910502E-01

THE FINAL SOLUTION TO THE SYSTEM FOR THE RIGHT HAND SIDE NUMBER      1
       0.9999999E 00      0.4999999E 00      0.3333333E 00      0.2499999E 00      0.2000000E 00

THE LENGTH OF THE FINAL SOLUTION VECTOR IS        0.1209797E 01

THE ERROR MATRIX

       0.2461610E 00      0.7977712E-01      0.3363910E-01      0.1456363E-01      0.5140688E-02

       0.7977712E-01      0.2632471E-01      0.1121059E-01      0.4884373E-02      0.1731764E-02

       0.3363910E-01      0.1121059E-01      0.4800081E-02      0.2098641E-02      0.7458888E-03

       0.1456363E-01      0.4884373E-02      0.2098641E-02      0.9195909E-03      0.3273471E-03

       0.5140688E-02      0.1731764E-02      0.7458888E-03      0.3273471E-03      0.1166535E-03
```

APPENDIX C

SETTING THE INPUT VARIABLES FOR LLCR AND LLSQ

The following flowchart and tables will give the user of the LLCR package the information he needs to set the FORTRAN variables needed by the package. A complete description of the calling sequence is given in Appendix A.

The first group of variables are usage independent. They should be set at the values given below for every problem solved by the package. Table VII contains the FORTRAN variables in the group and the corresponding values.

TABLE VII

SETTING THE USAGE INDEPENDENT FORTRAN VARIABLES NEEDED BY THE LLCR PACKAGE

FORTRAN Variables	Values
KREND	$\leq n$, $\geq$ KRBEG
EPS	$\geq \delta$
EPS1	$\geq \delta$
NR	m
NC	n
LA	$\geq m$
LRE	$\geq n$

n is the number of columns in the coefficient matrix, m is the number of rows in the coefficient matrix, and δ is the relative accuracy of the computer.

The values at which the second group of variables are set depends upon the amount of extra storage available for the program, the accuracy desired for the final solutions, and the amount of extra execution time the user is willing to sacrifice. Figure 2 contains a flow chart that will show the user how to set this group of variables.

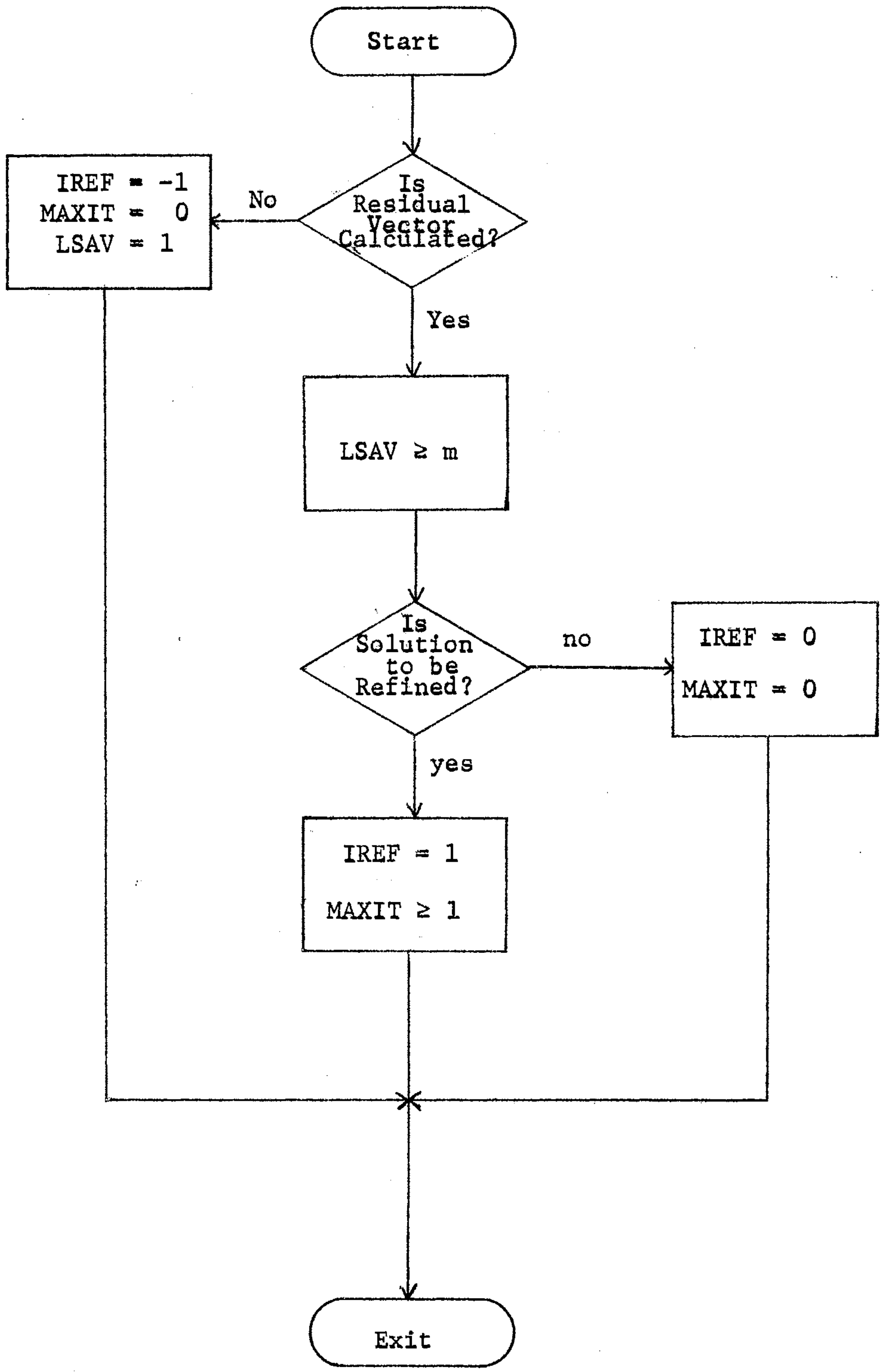

Figure 2. Setting the Usage Dependent FORTRAN Variables

The third group of variables are usage dependent; these variables will be set differently for different usages of the package. The appropriate value for each variable in this group for each usage is given in Table VIII.

TABLE VIII

SETTING THE USAGE DEPENDENT VARIABLES NEEDED BY THE LLCR PACKAGE

Program Usage	IPIV	KRBEG	NRHS	LSAL
To solve a system or systems of equations AX=B	0	n	h	1
To perform IVOR or "forward selection"	1	1	1	≥ n
To solve AX=B while treating the entries in A as if they had a variable precision	0	≥ 1	h	≥ n
To fit a polynomial to data	-1	1	1	≥ n
To calculate a generalized inverse	0	n	m	1

The tables and flowchart given above and the information given in Appendix A should should enable the user to work with LLCR and LLSQ with relative ease. The routines are also documented internally with comment cards.

VITA

Joanna Chamberlain Hwang

Candidate for the Degree of

Master of Science

Thesis: COMPUTER SOLUTION OF THE GENERALIZED LINEAR LEAST SQUARES PROBLEM USING MODIFIED GRAM-SCHMIDT ORTHOGONALIZATION

Major Field: Computing and Information Sciences

Biographical:

Personal Data: Born in New Castle, Pennsylvania, November 6, 1946, the daughter of Mr. and Mrs. Robert L. Chamberlain.

Education: Graduated from Shenango High School, New Castle, Pennsylvania, in June, 1964; attended Carnegie Institute of Technology, Pittsburgh, Pennsylvania, and the University of Houston, Houston, Texas; received a Bachelor of Science degree in mathematics from Oklahoma State University in 1969; completed requirements for the Master of Science degree at Oklahoma State University in July, 1972.

Professional Experience: graduate teaching assistant, Oklahoma State University, Department of Computing and Information Sciences, 1970-1971; instructor, Northern Oklahoma College, Tonkawa, Oklahoma, 1972-present.