SAS® Technical Report R-106

The Sweep Operator: Its Importance in Statistical Computing

5985

SAS Institute Inc.

The Sweep Operator:

Its Importance in Statistical Computing

by

J. H. Goodnight

SAS® Technical Report R-106

SAS Institute Inc.
SAS Circle, Box 8000
Cary, NC 27512-8000

The correct bibliographic citation for this technical report is as follows: SAS Institute Inc., SAS® Technical Report R-106, The Sweep Operator: Its Importance in Statistical Computing, Cary, NC: SAS Institute Inc., 1978.

SAS® Technical Report R-106, The Sweep Operator: Its Importance in Statistical Computing

Copyright© 1978 by SAS Institute Inc., Cary, NC, USA.

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

The SAS® System is an integrated system of software providing complete control over data management, analysis, and presentation. Base SAS software is the foundation of the SAS System. Products within the SAS System include SAS/ACCESS®, SAS/AF®, SAS/ASSIST®, SAS/DMI®, SAS/ETS®, SAS/FSP®, SAS/GRAPH®, SAS/IML®, SAS/IMS-DL/I®, SAS/OR®, SAS/QC®, SAS/REPLAY-CICS®, SAS/SHARE®, SAS/STAT®, SAS/CPE™, SAS/DB2™, and SAS/SQL-DS™ software. Other SAS Institute products are SYSTEM 2000® Data Management Software, with basic SYSTEM 2000, CREATE™, Multi-User™, QueX™, Screen Writer™ software, and CICS interface software; NeoVisuals™ software; JMP™ and JMP IN™ software; SAS/RTERM® software; SAS/C® and SAS/CX™ Compilers. SAS Communications®, SAS Training®, SAS Views®, and the SASware Ballot® are published by SAS Institute Inc. Plink86® and Plib86® are registered trademarks of Phoenix Technologies Ltd. All other trademarks above are registered trademarks or trademarks, as indicated by their mark, of SAS Institute Inc.

A footnote must accompany the first use of each Institute registered trademark or trademark and must state that the referenced trademark is used to identify products or services of SAS Institute Inc.

The Institute is a private company devoted to the support and further development of its software and related services.

THE SWEEP OPERATOR: ITS IMPORTANCE IN STATISTICAL COMPUTING

James H. Goodnight

ABSTRACT

The importance of the sweep operator in statistical computing is not so much that it is an inversion technique, but rather that it is a conceptual tool for understanding the least squares process itself. The sweep operator can be programmed to produce generalized inverses, and create as by-products such items as: the Forward Doolittle matrix, the Cholesky decomposition matrix, the Hermite canonical form matrix, the determinant of the original matrix, Type I sums of squares, the error sum of squares, a solution to the normal equations, and the general form of estimable functions. This tutorial paper begins by describing the use of Gauss-Jordan elimination for least squares and builds up to the description of a completely generalized sweep operator which computes and stores $(X'X)^-$, $(X'X)^-X'X$, $(X'X)^-X'Y$ and $Y'Y - Y'X(X'X)^-X'Y$ all in the space of a single upper triangular matrix.

KEY WORDS: Statistical computing; Sweep operator; Least squares; Regression analysis; Gaussian elimination.

1. INTRODUCTION

The sweep operator is perhaps the most versatile of all statistical operators. It can be adapted for use in ordinary least squares (including multiple regression and the general linear model), two- and three-stage least squares, non-linear least squares, multivariate analysis of variance, all possible regressions, regression by leaps and bounds, stepwise regression, partial correlation, and others. The most important aspect of the sweep operator with respect to its use as an instructional tool is that each element of the matrix being operated on is readily identifiable and has statistical meaning. In order to understand the significance of each element of a matrix being swept and to understand the interrelationship between Gauss-Jordan elimination, the Forward Doolittle, Cholesky decomposition, Hermite canonical forms and the sweep operator, this paper will start at the most basic level, that of Gauss-Jordan elimination.

Throughout the paper the general linear model:

$$Y = X\beta + e \tag{1.1}$$

will be used, where Y is an n x 1 vector of individual observations; X is an n x k matrix of 0's and 1's and/or continuous variables; β is a k x 1 vector of constant but unknown parameters, and e is distributed NID$(0,\sigma^2)$.

2. GAUSS-JORDAN ELIMINATION

Gauss-Jordan elimination may be used to solve directly the normal equations:

$$X'X\,b = X'Y \qquad (2.1)$$

which arise from (1.1). Gauss-Jordan elimination involves only two operations, that of multiplying equations by a constant and that of adding a multiple of one equation to another. The following simple example illustrates these points.

Given the system:

$$4b_1 + 2b_2 = 6$$

$$2b_1 + 6b_2 = 10$$

First: Multiply the first equation by 1/4

$$b_1 + 1/2\ b_2 = 3/2$$

$$2b_1 + \ 6b_2 = 10$$

Next: Add -2 times the first equation to the second

$$b_1 + 1/2\ b_2 = 3/2$$

$$0b_1 + \ 5b_2 = 7$$

Next: Multiply the second equation by 1/5

$$b_1 + 1/2\ b_2 = 3/2$$

$$0b_1 + \ b_2 = 7/5$$

Finally: Add -1/2 times the second equation to the first.

$$b_1 + 0b_2 = 4/5$$

$$0b_1 + b_2 = 7/5$$

This same sequence of operations can be carried out in a simple matrix tableau using equivalent row operations (multiplying a row by a constant and adding a multiple of one row to another. For example:

Given the tableau

$$\left[\begin{array}{cc|c} 4 & 2 & 6 \\ 2 & 6 & 10 \end{array}\right]$$

First: Multiply row 1 by 1/4

$$\left[\begin{array}{cc|c} 1 & 1/2 & 3/2 \\ 2 & 6 & 10 \end{array}\right]$$

Next: Add -2 times row 1 to row 2

$$\left[\begin{array}{cc|c} 1 & 1/2 & 3/2 \\ 0 & 5 & 7 \end{array}\right]$$

Next: Multiply row 2 by 1/5

$$\left[\begin{array}{cc|c} 1 & 1/2 & 3/2 \\ 0 & 1 & 7/5 \end{array}\right]$$

Finally: Add -1/2 times row 2 to row 1

$$\left[\begin{array}{cc|c} 1 & 0 & 4/5 \\ 0 & 1 & 7/5 \end{array}\right]$$

Thus to solve full-rank regression equations one has only to form the augmented matrix

$$\left[X'X \mid X'Y\right] \tag{2.2}$$

and use row operations to convert the left hand matrix to an identity. Once done, the right-hand matrix will be the solution. Symbolically:

$$\left[X'X \mid X'Y\right] \xrightarrow[\text{operations}]{\text{row}} \left[I \mid b\right] . \tag{2.3}$$

Note that $(X'X)^{-1}$ need not be computed in order to solve the normal equations, and that the entire process may be carried out in-place on a computer.

One of the most important aspects of performing row operations on a matrix is that it is equivalent to multiplying the matrix on the left by another matrix. Thus the row operations performed in (2.3) are equivalent to multiplying (2.2) by $(X'X)^{-1}$.

In addition to computing the b values, the error sum of squares may also be simultaneously computed by augmenting (2.2) as follows:

$$\left[\begin{array}{c|c} X'X & X'Y \\ \hline Y'X & Y'Y \end{array}\right]. \tag{2.4}$$

If (2.4) is multiplied on the left by

$$\left[\begin{array}{c|c} (X'X)^{-1} & 0 \\ \hline -Y'X(X'X)^{-1} & I \end{array}\right] \tag{2.5}$$

then the resultant product is:

$$\left[\begin{array}{c|c} I & (X'X)^{-1}X'Y \\ \hline 0 & Y'Y-Y'X(X'X)^{-1}X'Y \end{array}\right]. \tag{2.6}$$

Since (2.6) was achieved by multiplying (2.4) on the left by matrix (2.5), (2.6) may also be achieved by performing row operations on (2.4). That is,

$$\left[\begin{array}{c|c} X'X & X'Y \\ \hline Y'X & Y'Y \end{array}\right] \xrightarrow[\text{operations}]{\text{row}} \left[\begin{array}{c|c} I & (X'X)^{-1}X'Y \\ \hline 0 & Y'Y-Y'X(X'X)^{-1}X'Y \end{array}\right]. \tag{2.7}$$

Since X'X is positive definite, the row operations used in (2.3) and (2.7) (see Wilkinson, 1961), need consist only of a sequence of pivots on the diagonal elements of X'X. In fact, by restricting the row operations to pivots on the diagonal elements of X'X, much valuable statistical information may be obtained, and an easily programmed algorithm is achieved. Since pivoting on a diagonal element "adjusts" the remaining matrix elements for the variable associated with the diagonal

element, it is helpful to define this operation as an operator called the ADJUST operator.

Definition: ADJUST(K) operator

The ADJUST(K) operator performs a pivot on element a_{kk} of a matrix A as follows:

Step 1: Set $B = a_{kk}$

Step 2: Divide row k by B

Step 3: For each other row $i \neq k$

Set $B = a_{ik}$ and subtract B times Row k from Row i.

Using the ADJUST operator on a simple example shows the amount of statistical information that is available through its use. For the regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + e$$

the augmented normal equations (2.4) would be:

$$A = \begin{bmatrix} N & \Sigma X_1 & \Sigma X_2 & \Sigma Y \\ \Sigma X_1 & \Sigma X_1^2 & \Sigma X_1 X_2 & \Sigma X_1 Y \\ \Sigma X_2 & \Sigma X_1 X_2 & \Sigma X_2^2 & \Sigma X_2 Y \\ \Sigma Y & \Sigma X_1 Y & \Sigma X_2 Y & \Sigma Y^2 \end{bmatrix}$$

If an ADJUST(1) operation is performed on A then:

$$A_1 = \left[\begin{array}{c|ccc} 1 & \bar{X}_1 & \bar{X}_2 & \bar{Y} \\ \hline 0 & \Sigma x_1^2 & \Sigma x_1 x_2 & \Sigma x_1 y \\ 0 & \Sigma x_1 x_2 & \Sigma x_2^2 & \Sigma x_2 y \\ 0 & \Sigma x_1 y & \Sigma x_2 y & \Sigma y^2 \end{array}\right]$$

The lower right 3x3 in A_1 is the corrected sum of squares and cross product (SS&CP) matrix. The upper right 1x3 matrix consists of b values for the submodels: $X_1 = \beta_0' + e'$, $X_2 = \beta_0'' + e''$, and $Y = \beta_0''' + e'''$. The diagonals in the lower right 3x3 correspond to the error sum of squares for these submodels. If an ADJUST(2) operation is performed on A_1 then the following matrix results:

$$A_2 = \left[\begin{array}{cc|cc} 1 & 0 & b_0' & b_0'' \\ 0 & 1 & b_1' & b_1'' \\ \hline 0 & 0 & \Sigma x_2^2.1 & \Sigma x_2 y.1 \\ 0 & 0 & \Sigma x_2 y.1 & \Sigma y^2.1 \end{array}\right] .$$

In A_2 the upper right 2x2 matrix contains the b values for the submodels: $X_2 = \beta_0' + \beta_1' X_1 + e'$ and $Y = \beta_0'' + \beta_1'' X_1 + e''$. The lower right 2x2 matrix is the error SS&CP matrix for these two models. If an ADJUST(3) operation is performed on A_2 then the following matrix results.

$$A_3 = \left[\begin{array}{ccc|c} 1 & 0 & 0 & b_0 \\ 0 & 1 & 0 & b_1 \\ 0 & 0 & 1 & b_2 \\ \hline 0 & 0 & 0 & \Sigma y^2.1,2 \end{array}\right]$$

This last adjustment completes the solution of the normal equations and yields the error sum of squares in the bottom right position.

The Type I SS for each variable (see Barr et al., 1976), is an important by-product of the ADJUST operator. If the augmented matrix A as in (2.4) is p x p, then whenever ADJUST(K) is made, the Y'Y element a_{pp} is reduced by $(a_{kp})^2/a_{kk}$. Thus $R(X_k|$ the other variables adjusted for) = $(a_{kp})^2/a_{kk}$. An auxiliary vector can be employed to store the Type I SS prior to each use of ADJUST.

As noted above, after each use of ADJUST, the diagonals not yet ADJUSTED represent error sums of squares for submodels among the independent variables. Suppose that a matrix A as in (2.4) has been adjusted for 1,2,....,K-1. Then a_{kk} is the error SS for the model

$$X_k = \beta_0 + \beta_1 X_1 = \ldots \beta_{k-1} X_{k-1} + e.$$

The R^2 value for this model is then:

$$R_k^2 = (CSS_k - a_{kk})/CSS_k,$$

where CSS_k is the corrected SS for X_k. The R_k^2 value provides an invaluable singularity check. If $R_k^2 > .9999$ (for instance), then X_k is "statistically" at least, a linear combination of the preceding X's. The elements in column k associated with previously adjusted variables spell out what the linear combination is. Recent articles by Frane (1977) and Berk (1977) examine further the use of the tolerance, $T = 1-R^2$, for checking against singularities. The tolerance T will be discussed further in a later section concerning generalized inverses. The interested reader is referred to Hemmerle (1967) for additional discussion and numerical examples of Gauss-Jordan elimination.

In the past few years, hundreds of articles have appeared concerning the numerical accuracy and stability of regression solutions obtained using Gaussian elimination. Beaton et al. (1976) and (1977) reference some of the more important articles and conclude that in many cases, the attempt at estimating regression coefficients in highly collinear problems cannot be justified statistically. When data are highly collinear, then the technician in the lab who records the data may have more effect on the solution than does the technique used to achieve it. One of the most obvious facts about accuracy, that is often overlooked, is that when Gauss-Jordan elimination is used the mantissa length on the machine must at least be able to contain the largest element of X'X (see Golub and Klema, 1978). Any

program written in single precision is obviously headed for trouble when more than four significant digits appear in the input data. It is hoped that no least squares programs in common use are written in single precision when Gauss-Jordan elimination is used. For the very large class of least squares problems which involve only X's with 0's and 1's and whose largest X'X element is equal to the number of observations, Gauss-Jordan elimination programmed in double precision is more than adequate with respect to accuracy, and cannot be topped in terms of storage or speed.

3. OTHER ADJUST USES

For multivariate linear models, where several dependent variables have the same set of independent variables, the b values and the error SS&CP matrix may be obtained simultaneously; i.e.:

$$\left[\begin{array}{c|c} X'X & X'Y \\ \hline Y'X & Y'Y \end{array}\right] \xrightarrow[\text{Columns of } X'X]{\text{ADJUST on}}$$

$$\left[\begin{array}{c|c} I & (X'X)^{-1}\,X'Y \\ \hline 0 & Y'Y-Y'X(X'X)^{-1}\,X'Y \end{array}\right] \qquad (3.1)$$

This is equivalent to the univariate case, except that Y has dimensions N x p instead of N x 1.

Partial correlation coefficients for two sets of variables, X_1(N x p) and X_2(N x q), may be obtained using the adjust operator. Suppose the correlations among the X_2's are needed

after their adjustment for X_1, then proceed as follows (assume X_1 contains a column of 1's):

$$\left[\begin{array}{c|c} X_1'X_1 & X_1'X_2 \\ \hline X_2'X_1 & X_2'X_2 \end{array}\right] \xrightarrow[\text{Columns of } X_1'X_1]{\text{ADJUST for}}$$

$$\left[\begin{array}{c|c} I & (X_1'X_1)^{-1} X_1'X_2 \\ \hline 0 & X_2'X_2 - X_2'X_1(X_1'X_1)^{-1} X_1'X_2 \end{array}\right] . \tag{3.2}$$

Next, divide each element of $X_2'X_2 - X_2'X_1(X_1'X_1)^{-1} X_1'X_2$ by the square root of the two associated diagonals, and the partial correlation matrix is complete.

4. THE FORWARD DOOLITTLE

The basic Doolittle method (see Steel and Torrie, 1960), was introduced in 1878 while Doolittle was an engineer with the U. S. Coast and Geodetic Survey. Doolittle popularized and slightly restructured Gaussian elimination to fit the needs of the statistician. As it has evolved it is now referred to as the abbreviated Doolittle and consists of two parts. The first part, the Forward Doolittle, maps the sum of squares and cross products matrix into an upper triangular matrix. The second part, the Backward solution, computes the regression coefficients and the inverse matrix. Symbolically, the Forward Doolittle is

used to map any symmetric positive definite matrix C into an upper triangular matrix A:

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \xrightarrow[\text{operations}]{\text{row}}$$

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ 0 & C_{22.1} & C_{23.1} \\ 0 & 0 & C_{33.12} \end{bmatrix} = A$$

The Forward Doolittle matrix is constructed by adding multiples of the first row of the matrix to all following rows such that zeros are introduced below the diagonal in column one. Multiples of row two are then added to all following rows to introduce zeros below the diagonal in column two, etc. If the positive definiteness of the matrix is in question the tolerance check discussed in section three may be used. Note that if the ADJUST operator had been applied to the C matrix, and if the adjustments were made sequentially (k = 1,2,3,...), then prior to the adjustment for row k, row k of C is equal to row k of A.

When only the adjusted SS&CP matrix is needed as in (3.1) and (3.2), the Forward Doolittle affords a faster method of computing than the ADJUST operator. A further example of this is the computation of:

$$SS(HO:\ L\beta=0) = (Lb)'\ (L(X'X)^{-}L')^{-1}\ (Lb),$$

where $L\beta$ is estimable and b is any solution to (1.1). If the following matrix is formed:

$$\left[\begin{array}{c|c} L(X'X)^{-}L' & Lb \\ \hline (Lb)' & 0 \end{array}\right]$$

and a Forward Doolittle is performed on the diagonals of the upper left matrix, then the lower right matrix is transformed into:

$$0 - (Lb)'\ (L(X'X)^{-}L')^{-1}\ (Lb),$$

the negative of which is $SS(Ho:\ L\beta=0)$.

5. FACTORING A SYMMETRIC POSITIVE DEFINITE MATRIX

If the symmetric positive definite matrix C has been mapped as follows:

$$C \xrightarrow[\text{Doolittle}]{\text{Forward}} A$$

then if each row of A is divided by its diagonal to produce a matrix B, then

$$B'A = C.$$

Furthermore, if each row of A is divided by the square root of its diagonal to p·oduce a matrix U, then

$$U'U = C\ .$$

The matrix U is called the Cholesky decomposition of C, see Isaacson and Keller (1966).

6. THE DETERMINANT OF A SYMMETRIC POSITIVE DEFINITE MATRIX

Having mapped C into A by the Forward Doolittle and created B by dividing each row of A by its diagonal, then:

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 \\ b_{12} & 1 & 0 \\ b_{13} & b_{23} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}$$

$$\text{i.e.} \quad C = B'A$$

$$\text{therefore} \quad |C| = |B'| \quad |A| \cdot$$

$$= 1 \cdot \prod_i a_{ii}.$$

The determinant of an upper or lower triangular matrix is always the product of its diagonal elements, by the basic permuted products summation definition. Therefore the

determinant is easily a by-product of both the Forward Doolittle and the ADJUST operator.

7. FINDING THE INVERSE

The inverse of a symmetric positive definite matrix A, may be found by augmenting an identity matrix to the right of A and then using the ADJUST operator on the diagonals of A. Symbolically:

$$\left[A \mid I\right] \xrightarrow[\text{diagonals}]{\text{ADJUST A}} \left[I \mid A^{-1}\right] . \tag{7.1}$$

Since repetitive use of the ADJUST operator is equivalent to multiplication on the left by some matrix, (7.1) is equivalent to (7.2).

$$A^{-1}\left[A \mid I\right] = \left[I \mid A^{-1}\right] . \tag{7.2}$$

Since in (7.1) the original identity matrix keeps count of the row operations being performed, this technique has been referred to as the method of counters.

The above process is also reversible since,

$$A\left[I \mid A^{-1}\right] = \left[A \mid I\right]$$

thus

$$\left[I \mid A^{-1}\right] \xrightarrow[\text{diagonals}]{\text{ADJUST } A^{-1}} \left[A \mid I\right]$$

8. OBTAINING b VALUES, ERROR SS, AND INVERSE

By forming the following augmented matrix and adjusting the columns associated with X'X, the regression coefficients, error SS and inverse may be computed simultaneously, i.e.

$$\left[\begin{array}{c|c|c} X'X & X'Y & I \\ \hline Y'X & Y'Y & 0 \end{array}\right] \xrightarrow[X'X \text{ columns}]{\text{ADJUST}}$$

$$\left[\begin{array}{c|c|c} I & b & (X'X)^{-1} \\ \hline 0 & \text{ESS} & -b' \end{array}\right]$$

This is equivalent to the matrix multiplication,

$$\left[\begin{array}{c|c} (X'X)^{-1} & 0 \\ \hline -Y'X(X'X)^{-1} & I \end{array}\right] \left[\begin{array}{c|c|c} X'X & X'Y & I \\ \hline Y'X & Y'Y & 0 \end{array}\right] =$$

$$\left[\begin{array}{c|c|c} I & b & (X'X)^{-1} \\ \hline 0 & \text{ESS} & -b' \end{array}\right]$$

Also note that

$$\left[\begin{array}{c|c|c} I & b & (X'X)^{-1} \\ \hline 0 & \text{ESS} & -b' \end{array}\right] \xrightarrow[(X'X)^{-1} \text{ columns}]{\text{readjust}}$$

$$\left[\begin{array}{c|c|c} X'X & X'Y & I \\ \hline Y'X & Y'Y & 0 \end{array}\right]$$

The reversibility of adjustments allows one to ADJUST any of the original columns of X'X and thus "enter" that variable into

the model, and also to readjust any column in the $(X'X)^{-1}$ area and thus "remove" that variable from the model. For example, consider the following data:

X_0	X_1	X_2	Y
1	1	1	1
1	2	1	3
1	3	1	3
1	1	-1	2
1	2	-1	2
1	3	-1	1

Form $\left[\begin{array}{c|c|c} X'X & X'Y & I \\ \hline Y'X & Y'Y & 0 \end{array}\right]$

	X_0	X_1	X_2	b	X_0	X_1	X_2	
STEP 0 FORM MATRIX	6	12	0	12	1	0	0	
	12	28	0	25	0	1	0	
	0	0	6	1	0	0	1	(8.1)
	12	25	2	28	0	0	0	
STEP 1 ENTER X_0	1	2	0	2	1/6	0	0	
	0	4	0	1	-2	1	0	
	0	0	6	2	0	0	1	(8.2)
	0	1	2	4	-2	0	0	

	X_0	X_1	X_2	b	X_0	X_1	X_2	
	1	0	0	3/2	7/6	-1/2	0	
STEP 2	0	1	0	1/4	-1/2	1/4	0	(8.3)
ENTER X_1	0	0	6	2	0	0	1	
	0	0	2	15/4	-3/2	-1/4	0	

	1	0	0	3/2	7/6	-1/2	0	
STEP 3	0	1	0	1/4	-1/2	1/4	0	(8.4)
ENTER X_2	0	0	1	1/3	0	0	1/6	
	0	0	0	37/12	-3/2	-1/4	-1/3	

	1	2	0	2	1/6	0	0	
STEP 4	0	4	0	1	-2	1	0	(8.5)
REMOVE X_1	0	0	1	1/3	0	0	1/6	
	0	1	0	10/3	-2	0	-1/3	

	1	2	0	2	1/6	0	0	
STEP 5	0	4	0	1	-2	1	0	(8.6)
REMOVE X_2	0	0	6	2	0	0	1	
	0	1	2	4	-2	0	0	

	6	12	0	12	1	0	0	
STEP 6	12	28	0	25	0	1	0	(8.7)
REMOVE X_0	0	0	6	2	0	0	1	
	12	25	2	28	0	0	0	

The above tableaus represent the following models with respect to Y.

Step	Model	b values	error SS
0	$Y = 0$		28
1	$Y = b_0$	2	4
2	$Y = b_0+b_1X_1$	3/2, 1/4	15/4
3	$Y = b_0+b_1X_1+b_2X_2$	3/2, 1/4, 1/3	37/12
4	$Y = b_0+b_2X_2$	2, 1/3	10/3
5	$Y = b_0$	2	4
6	$Y = 0$		28

Observing the tableau at Step 3 (the full model), note that if X_1 is removed, the ESS will increase by (1/4 * 1/4)/1/4 = 1/4 = Type II SS due to X_1. If X_2 is removed from the full model, the ESS will increase by (1/3 . 1/3)/1/6 = 2/3 = Type II SS due to X_2. The Type II SS due to X_i (given the full model) is always $(b_i)^2/C_{ii}$, where C_{ii} is i^{th} diagonal element of $(X'X)^{-1}$. The Type II SS corresponds to the SS due to the hypothesis that $\beta_i = 0$. The reversibility of adjustments demonstrated above forms the basis for stepwise regression.

9. THE SWEEP OPERATOR

The SWEEP operator is a modification of the ADJUST operator which reduces the amount of core needed to compute the b values,

error SS, and $(X'X)^{-1}$. Whereas the ADJUST operator performs the in-place mapping:

$$\left[\begin{array}{c|c|c} X'X & X'Y & I \\ \hline Y'X & Y'Y & 0 \end{array}\right] \xrightarrow[X'X \text{ Columns}]{\text{ADJUST}} \left[\begin{array}{c|c|c} I & b & (X'X)^{-1} \\ \hline 0 & \text{ESS} & -b' \end{array}\right]$$

the sweep operator performs the in-place mapping:

$$\left[\begin{array}{c|c} X'X & X'Y \\ \hline Y'X & Y'Y \end{array}\right] \xrightarrow[X'X \text{ Columns}]{\text{SWEEP}} \left[\begin{array}{c|c} (X'X)^{-1} & b \\ \hline -b' & \text{ESS} \end{array}\right]$$

One of the first references to sweep operations may be found in Ralston (1960), but the term "sweep operator" was coined by Beaton (1964).

By observing tableaus (8.1)-(8.7), note that whenever a variable is in the model its associated X'X column is an identity column, and whenever a variable is not in the model its associated column in the original identity matrix remains unchanged. The SWEEP operator takes advantage of these features, and after adjusting the tableau for a particular variable, replaces the identity column just created by the associated column in the original identity matrix which has now been modified. By observing what happens to the i^{th} identity column when variable X_i is entered, the ADJUST operator can be modified to a SWEEP operator. When variable X_k is entered, row k is multiplied by $1/a_{kk}$, therefore its identity column will have $1/a_{kk}$ in the k^{th} row.

To zero out the remaining elements in column k, $-a_{ik}$ * row k is added to all other rows ($i \neq k$). Since the identity column has 0's in all positions other than row k, the i^{th} row of the identity column becomes $-a_{ik}/a_{kk}$ ($i \neq k$). With this in mind, the SWEEP operator is defined as follows.

SWEEP(K) defined:

Given an originally symmetric positive definite matrix A, SWEEP(K) will modify the current matrix A in the following manner:

Step 1: Let $D = a_{kk}$.

Step 2: Divide row k by D.

Step 3: For every other row $i \neq k$, let $B = a_{ik}$,
Subtract B * row k from row i, then
Set $a_{ik} = -B/D$.

Step 4: Set $a_{kk} = 1/D$.

Since the SWEEP operator is only a modification of the ADJUST operator, the reversibility of SWEEP is apparent. A useful exercise, for the interested reader, would be to use the SWEEP operator on the left most 4x4 matrix in tableau (8.1), to compute the equivalent sweep tableaus associated with (8.2)-(8.7).

It should be noted that, although the amount of core needed to compute the b values, error SS, and $(X'X)^{-1}$ has been reduced

by using SWEEP, none of the by-product information has been lost. The Type I and II SS may still be computed. The determinant of X'X may still be computed. The singularity check is still available, and all submodel information is present. For example, let

$$A = \left[\begin{array}{c|c|c} X_1'X_1 & X_1'X_2 & X_1'Y \\ \hline X_2'X_1 & X_2'X_2 & X_2'Y \\ \hline Y'X_1 & Y'X_2 & Y'Y \end{array}\right]$$

Sweeping A on the columns associated with $X_1'X_1$ yields:

$$\left[\begin{array}{c|c|c} (X_1'X_1)^{-1} & (X_1'X_1)^{-1}X_1'X_2 & (X_1'X_1)^{-1}X_1'Y \\ \hline -X_2'X_1(X_1'X_1)^{-1} & X_2'M_1X_2 & X_2'M_1Y \\ \hline -Y'X_1(X_1'X_1)^{-1} & Y'M_1X_2 & Y'M_1Y \end{array}\right] \qquad (9.1)$$

where $M_1 = I - X_1(X_1'X_1)^{-1}X_1'$.

The regression coefficients for the submodels: Model $X_2 = X_1$; and Model $Y = X_1$; are clearly available. $X_2'M_1X_2$ is the error SSCP for the first model, and $Y'M_1Y$ is the error SSCP for the second model.

The ADJUST operator and the Forward Doolittle operator are all easily programmed to operate only on the upper triangular matrix by observing their symmetry properties. The SWEEP operator is also easily modified to operate on just the upper triangular portion of the matrix.

The amount of time it takes to compute $(X'X)^{-1}$, b values, and error SS using the SWEEP operator is generally only a fraction of the time it takes to form the sum of squares and cross product matrix. If only the upper triangular portion of the sum of squares and cross product matrix is formed as the data is being read, then one SWEEP operation (modified to operate on the upper triangle) takes about the same amount of CPU time as does the reading and accumulation of two observations. This can be verified by counting the number of multiplications and additions that are performed. Thus the approximate CPU time ratio of inversion to building the X'X is approximately 2k/N where k is the total number of independent variables and N is the number of observations. Since N is usually much larger than k, inversion represents only a fraction of the cost of regression analysis. Additional references to the SWEEP operator may be found in Seber (1977).

10. COMPUTING GENERALIZED INVERSES

A large family of models, characterized by (1.1), are over-parameterized. Linear dependencies are known to exist among the columns of the X matrix. For some models, the dependencies are easily predictable. For others, the interchanging of effects (putting interactions before main effects), the absence of a cell, or the inclusion of a covariable may bring about unexpected dependencies. The ADJUST operator, the

Forward Doolittle, and the SWEEP operator, through use of the tolerance check, can detect dependencies among the columns of the X matrix. In this section, the SWEEP operator will be modified to produce a generalized inverse and the corresponding particular solution whenever a dependency is encountered. Additional references to generalized inverses may be found in Pringle and Rayner (1971) and Ben-Israel and Greville (1974).

Initially assume that the X matrix can be partitioned, i.e., $X = [X_1 \mid X_2]$, such that the columns of X_1 are linearly independent, and each column of X_2 is a linear combination of the columns of X_1. This implies that:

$$X_2 = X_1 L \qquad \text{(by definition)},$$

$$L = (X_1'X_1)^{-1}X_1'X_2 \qquad \text{(regress } X_2 \text{ on } X_1\text{)},$$

$$X_2'X_2 - X_2'X_1(X_1'X_1)^{-1}X_1'X_2 = 0 \qquad \text{(error SS\&CP)}.$$

The X'X matrix, $\begin{bmatrix} X_1'X_1 & X_1'X_2 \\ X_2'X_1 & X_2'X_2 \end{bmatrix}$ has two particular generalized inverses which are easily constructed through use of the SWEEP operator. The first, a g_1 inverse ($AA^{g1}A = A$) for X'X, is

$$\left[\begin{array}{c|c} (X_1'X_1)^{-1} & (X_1'X_1)^{-1}X_1'X_2 \\ \hline -X_2'X_1(X_1'X_1)^{-1} & 0 \end{array}\right].$$

This matrix results from SWEEPing X'X on the columns

associated with $X_1'X_1$. If the full augmented tableau,

$$\left[\begin{array}{cc|c} X_1'X_1 & X_1'X_2 & X_1'Y \\ X_2'X_1 & X_2'X_2 & X_2'Y \\ \hline Y'X_1 & Y'X_2 & Y'Y \end{array}\right] \tag{10.1}$$

is swept on the columns associated with $X_1'X_1$, the resultant tableau is:

$$\left[\begin{array}{cc|c} (X_1'X_1)^{-1} & (X_1'X_1)^{-1}X_1'X_2 & (X_1'X_1)^{-1}X_1Y \\ -X_2'X_1(X_1'X_1)^{-1} & 0 & 0 \\ \hline -Y'X_1(X_1'X_1)^{-1} & 0 & Y'Y-Y'X_1(X_1'X_1)^{-1}X_1'Y \end{array}\right] \tag{10.2}$$

Thus sweeping on the columns of $X_1'X_1$ yields a g_1 inverse and the appropriate adjusted Y'Y. However, the b values, $b_1 = (X_1'X_1)^{-1}X_1'Y$ and $b_2 = 0$, do not correspond to the b values computed using the g_1 inverse, i.e.,

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} (X_1'X_1)^{-1} & (X_1'X_1)^{-1}X_1'X_2 \\ -X_2'X_1(X_1'X_1)^{-1} & 0 \end{bmatrix} \begin{bmatrix} X_1'Y \\ X_2'Y \end{bmatrix} .$$

$$\begin{aligned} b_1 &= (X_1'X_1)^{-1}X_1'Y + (X_1'X_1)^{-1}X_1'X_2X_2'Y \\ &= (X_1'X_1)^{-1}X_1'Y + LX_2'Y . \end{aligned}$$

$$b_2 = -X_2'X_1(X_1'X_1)^{-1}X_1'Y$$

$$= -L'X_1'Y$$

$$= -X_2'Y \ .$$

Although $b_1 = (X_1'Y_1)^{-1}X_1'Y$ and $b_2 = 0$ is a solution to the normal equations this particular solution does not equal $(X'X)^{g_1}X'Y$.

Another disadvantage of computing a g_1 inverse is that additional computation is needed to compute the variance of the particular solution since:

$$\text{Var}((X'X)^{g_1}X'Y) = (X'X)^{g_1}X'X\ (X'X)^{g_1}\ \sigma^2 \ .$$

On the other hand, a g_2 inverse ($AA^{g_2}A = A$, and $A^{g_2}\ A\ A^{g_2} = A^{g_2}$) of X'X has the property that:

$$\text{Var}((X'X)^{g_2}\ X'Y) = (X'X)^{g_2}\ \sigma^2 \ .$$

It is easily verified that

$$\begin{bmatrix} (X_1'X_1)^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

is a g_2 inverse of X'X. If the full augmented tableau (10.1) is swept on the columns associated with $X_1'X_1$, then the resultant tableau is (10.2).

Introducing 0's in the two submatrices adjoining $(X_1'X_1)^{-1}$ in (10.2) yields:

$$\left[\begin{array}{cc|c} (X_1'X_1)^{-1} & 0 & (X_1'X_1)^{-1}X_1'Y \\ 0 & 0 & 0 \\ \hline -Y'X_1(X_1'X_1)^{-1} & 0 & Y'Y-Y'X_1(X_1'X_1)^{-1}X_1'Y \end{array}\right] \qquad (10.3)$$

This tableau contains $(X'X)^{g_2}$ and the adjusted Y'Y. The b values $b_1 = (X_1'X_1)^{-1}X_1'Y$ and $b_2 = 0$ are equivalent to $(X'X)^{g_2}X'Y$, and therefore $\text{Var}(b) = (X'X)^{g_2}\sigma^2$. If the g_2 inverse of (10.3) is employed, then the expected value of b_1 and b_2 are:

$$E(b_1) = E((X_1'X_1)^{-1}X_1'Y) = \beta_1 + (X_1'X_1)^{-1}X_1'X_2\beta_2$$

$$E(b_2) = 0.$$

If the submatrix containing $(X_1'X_1)^{-1}X_1'X_2$ is saved after sweeping on the columns of $X_1'X_1$, then this matrix may be used to compute all possible solutions to the normal equations since (for Z arbitrary):

$$b^* = (X'X)^{g_2}X'Y + (I-(X'X)^{g_2}X'X)Z$$

$$= \begin{bmatrix} (X_1'X_1)^{-1}X_1'Y \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -(X_1'X_1)^{-1}X_1'X_2 \\ 0 & I \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}$$

(10.4)

Therefore,

$$b_1^* = (X_1'X_1)^{-1}X_1'Y - (X_1'X_1)^{-1}X_1'X_2Z_2 = b_1 - LZ_2$$

$$b_2^* = Z_2 .$$

The foregoing discussion was simplified by assuming that the X's were grouped with X_1 being of full column rank and $X_2 = X_1L$. Naturally, this seldom happens but, as it turns out, the SWEEP operator can be modified to handle the situation in which the linearly independent columns of X are not all grouped.

For model (1.1), if it were known which columns of X formed a linearly independent basis then an index matrix M (an identity matrix with columns permuted) could be formed to rearrange the X's, and M' could be used to rearrange the β's. In other words, since $MM' = I$;

$$Y = X\beta + e$$

$$= XI\beta + e$$

$$= XMM'\beta + e.$$

Letting $Z = XM$ and $\gamma = M'\beta$, then $Y = Z\gamma + e$, where $Z = [Z_1 \mid Z_2]$ with the columns of Z_1 linearly independent and $Z_2 = Z_1L$. Thus

$$(Z'Z)^{g2} = \begin{bmatrix} (Z_1'Z_1)^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

and $\hat{\gamma} = (Z'Z)^{g2}Z'Y$.

Since $(Z'Z)^{g_2} = (M'X'XM)^{g_2} = M'(X'X)^{g_2}M$

Then $(X'X)^{g_2} = M(Z'Z)^{g_2}M'$ and $\hat{\beta} = M\hat{\gamma}$. (10.5)

In essence, the above rearrangement process selects a maximum rank subset of elements from X'X, inverts the resulting matrix, then puts the inverted subset of elements back into their original positions and zero's out the remaining elements to form $(X'X)^{g_2}$. The SWEEP operator can achieve the same results without any rearrangement of rows and columns, simply by proceeding as normal, except when encountering a variable which it finds to be linearly dependent on previously swept variables. When a depen dependency is encountered, say for variable K, the SWEEP operator should set the K^{th} row and column to zero and proceed. A g_2 inverse and corresponding solution will result. Note well, however, that once a row and column have been zeroed, the generalized SWEEP is no longer reversable. The critical information which defines that variable's dependency has been set to zero. With these points in mind, the G2SWEEP operator is defined as follows:

Given an originally symmetric positive definite or symmetric positive semi-definite matrix A as in (10.1) which has not been SWEPT on column K, G2SWEEP(K) will modify the current matrix A in the following manner:

Step 1: Let $D = a_{kk}$
If D is less than $DMIN_k$ then set row and column k to zero and stop at this step. Otherwise proceed.

Step 2: Divide row k by D.

Step 3: For every other row $i \neq k$, let $B = a_{ik}$, subtract B * row k from row i, then set $a_{ik} = -B/D$.

Step 4: Set $a_{kk} = 1/D$.

As stated above, the G2SWEEP operator should be applied only once to any given column of A, (although the order is unimportant) since information needed may have been set to zero. In addition a_{kk} once swept may be legitimately less than $DMIN_k$.

The $DMIN_k$ values used in the G2SWEEP operator are functions of the TOLERANCE value and either the corrected SS(CSS) or uncorrected SS(USS) for variable k. If no intercept is employed in the model then:

$$DMIN_k = \begin{cases} \text{Tolerance if USS} = 0 \\ \text{Tolerance} * \text{USS otherwise} \end{cases}$$

If an intercept is employed in the model then:

$$DMIN_k = \begin{cases} \text{Tolerance if } X_k \text{ does not vary or has} \\ \text{a computed CSS} <= 0. \\ \text{Tolerance} * \text{CSS otherwise} \end{cases}$$

In checking to see if X_k varies (has more than two distinct values) if the smallest and largest non zero absolute values of X_k are computed, this affords the opportunity to check the ratio of the largest to smallest non-zero absolute values of X_k. If the ratio is greater than 1E7 (or 1E8) then the chances are that the computed X'X matrix in double precision is incorrect to start with.

Establishing the value to use for tolerance is not a simple task, since it involves (as does the comparison of two means) compromising between a Type I and Type II error. The major difference between choosing a Tolerance and an α level is the lack of distributional theory. Too large a tolerance may declare variables dependent on others when they in fact are not. Too small a tolerance may let dependencies go unchecked. For variables which take on only 0 or 1 values, $(1-R^2)$ values in the range 1E-12 to 1E-14 are quite common when dependencies occur. Fifth and sixth degree polonomials may exhibit $(1-R^2)$ values as small as 1-E8 even when they are linearly independent. Thus tolerance values in the range of 1E-8 to 1E-12 seem appropriate when looking for dependencies.

11. THE UPPER TRIANGULAR G2SWEEP

As discussed by Schatzoll et al. (1968) the amount of time it takes to perform a sweep operation may be reduced by

approximately one half by taking into account the symmetry properties of the sweep tableau. The amount of core needed to store the tableau may also be reduced by almost one half by saving and operating on only the upper triangular portion of the tableau. Since operating on the upper triangular tableau, stored in a one-dimension array, is an obvious extension of what is to be discussed here, the tableau will be assumed to be stored in a rectangular array, but reference will be made only to elements on or above the diagonal.

Starting with the following tableau:

$$\begin{bmatrix} X_1'X_1 & X_1'X_2 & X_1'Y \\ X_2'X_1 & X_2'X_2 & X_2'Y \\ Y'X_1 & Y'X_2 & Y'Y \end{bmatrix} \tag{11.1}$$

perform a G2SWEEP on the $X_1'X_1$ columns. This results in:

$$\left[\begin{array}{c|c|c} (X_1'X_1)^- & (X_1'X_1)^-X_1'X_2 & (X_1'X_1)^-X_1'Y \\ \hline -X_2'X_1(X_1'X_1)^- & X_2'M_1X_2 & X_2'M_1Y \\ \hline -Y'X_1(X_1'X_1)^- & Y'M_1X_2 & Y'M_1Y \end{array}\right] \tag{11.2}$$

where $(X_1'X_1)^-$ is a symmetric g_2 inverse and $M_1 = I - X_1(X_1'X_1)^-X_1'$. Note that this tableau is symmetric, except for the sign of the submatrices below $(X_1'X_1)^-$. Letting a_{ij} denote the elements of this tableau, then the elements below the diagonal a_{ij} (with $i > j$)

may be constructed from the elements above the diagonal in the following manner:

$$a_{ij} = \begin{cases} a_{ji} & \text{if i and j have been swept.} \\ a_{ji} & \text{if neither i nor j has been swept.} \\ -a_{ji} & \text{if i or j (but not both) has been swept.} \end{cases}$$

By keeping track of which columns have been swept, the lower triangular portion of the tableau can be constructed from the upper.

The easiest way to keep track of what has been swept and what has not is through use of an auxiliary vector V. The elements of V are initially set to 1, then after sweeping on column k, V_k is set to $-V_k$. If this vector V is employed, then elements below the diagonal may be constructed from the upper triangular portion as follows:

$$a_{ij} = a_{ji} * V_i * V_j \quad (\text{for } i > j)$$

Note that the upper triangular elements are always correct as is. It is only when an element is not in the upper triangular portion of the tableau that multiplying a_{ji} by $V_i * V_j$ is necessary. The upper triangular G2SWEEP makes the same assumptions about the matrix A as does the G2SWEEP operator. Up to the point where a dependency is declared, reversibility

is possible. The vector V is assumed to have been set to 1 prior to any sweeps. UTG2SWEEP(K) modifies the upper triangular portion of the matrix A in the following manner:

Step 1: Let $D = a_{kk}$
If $V_k = 1$ and $D <= DMIN_k$ then zero elements above and to the right of a_{kk} including a_{kk}, then terminate. Otherwise proceed.

Step 2: For each value of i: i = 1,2,...,NROWS except for i = k perform Step 3. Then go to Step 5.

Step 3: If i < k then $B = a_{ik}/D$
Otherwise $B = V_i*V_k*a_{ki}/D$
Then for each value j: j = i, i+1,...,
NCOLS except for j = k perform Step 4

Step 4: If k < j then $c = a_{kj}$
Otherwise $c = V_j*V_k*a_{jk}$
Set $a_{ij} = a_{ij} - B*C$

Step 5: For each value of i: i = 1,...,k
Set $a_{ij} = -a_{ij}/D$
For each value of j: j = k,...,NCOLS
Set $a_{kj} = a_{kj}/D$
Then set $a_{kk} = 1/D$ and $V_k = -V_k$

12. THE REVERSIBLE UPPER TRIANGULAR G2SWEEP

By modifying the UTG2SWEEP such that it does not zero the k^{th} row and column when a dependency occurs, reversibility may be achieved. In addition the resulting tableau will contain both $(X'X)^{g_2}$ and $(X'X)^{g_2}X'X$ which are easily separated, along with $(X'X)^{g_2}X'Y$ and $Y'Y - Y'X(X'X)^{g_2}X'Y$. To demonstrate these properties, reconsider the case in which X was partitioned into [X1 ¦ X2] with X_1 of full column rank and $X_2 = X_1L$. Sweeping the initial tableau:

$$\begin{bmatrix} X_1'X_1 & X_1'X_2 & X_1'Y \\ X_2'X_1 & X_2'X_2 & X_2'Y \\ Y'X_1 & Y'X_2 & Y'Y \end{bmatrix} \tag{12.1}$$

on the columns of $X_1'X_1$ yields:

$$\begin{bmatrix} (X_1'X_1)^{-1} & (X_1'X_1)^{-1}X_1'X_2 & (X_1'X_1)^{-1}X_1'Y \\ -X_2'X_1(X_1'X_1)^{-1} & 0 & 0 \\ -Y'X_1(X_1'X_1)^{-1} & 0 & Y'M_1Y \end{bmatrix} \tag{12.2}$$

where $M_1 = I - X_1(X_1'X_1)^{-1}X_1'$. Sweeping (12.2) on the columns of $(X_1'X_1)^{-1}$ yields (12.1), therefore reversibility can be achieved. Note also in (12.1) and (12.2) that the same symmetry properties discussed for the UTG2SWEEP also hold.

The $(X'X)^{g_2}$ matrix can be obtained from the final tableau (in which no columns are left to sweep) by use of the V vector described earlier. Letting G represent the g_2 inverse, and A the final tableau, then

$$g_{ij} = \begin{cases} 0 & \text{if } V_i + V_j >= 0, \text{ otherwise} \\ a_{ij} & \text{if } i <= j \text{ or } a_{ji} \text{ if } i > j \end{cases}$$

Using the g_2 inverse described above,

$$(X'X)^{g_2}X'X = \begin{bmatrix} I & (X_1'X_1)^{-1}X_1'X_2 \\ 0 & 0 \end{bmatrix}$$

Letting H represent $(X'X)^{g_2}X'X$, and A the final tableau, then

$$h_{ii} = \begin{cases} 0 & \text{if } V_i = 1 \\ 1 & \text{otherwise} \end{cases}$$

$$h_{ij} = \begin{cases} 0 & \text{if } V_i + V_j \neq 0, \text{ otherwise} \\ a_{ij} & \text{if } i <= j \text{ or } -a_{ji} \text{ if } i > j \end{cases}$$

As was discussed with the G2SWEEP operator it is not necessary that the columns of X be grouped, since the tolerance check can be used to determine which variables are linearly dependent on the ones previously swept. In defining the reversable UTG2SWEEP, the same initial tableau and V vector as used in the UTG2SWEEP will be assumed.

The RUTG2SWEEP(K) modifies the upper triangular portion of the matrix A in the following manner:

Step 1: Let $D = a_{kk}$

If $V_k = 1$ and $D <= DMIN_k$ then terminate.

Column K cannot be swept at this time.

Otherwise proceed.

Step 2 - Step 5 are the same as for UTG2SWEEP.

Using the RUTG2SWEEP, the Type II SS for any effect may be computed by starting with the tableau at any stage and making the appropriate sweeps.

13. USING RUTG2SWEEP SEQUENTIALLY

By using the RUTG2SWEEP sequentially (K = 1,2,...,NX) all of the statistics described previously are available. Prior to any sweep, row k's diagonal and elements to the right of the diagonal correspond to the k'th row of the Forward Doolittle, thus $(a_{ky})^2/a_{kk}$ (provided $a_{kk} > DMIN_k$) is the Type I SS for variable k. Also the product of each diagonal (just prior to sweeping) equals the determinant of X'X. When sweeping is done sequentially then $(X'X)^{g_2}X'X$ is the Hermite canonical form H, is upper triangular and the elements of H (except for some 0's and 1's) are already in place in the final tableau. H can also be computed by sequentially pivoting on each non-zero row of X'X.

Since the expected value of the solution achieved using RUTG2SWEEP is equal $H\beta$, the nature of the bias in the solution is at hand. The H matrix is also one of many matrices whose rows can be used as a generating set to compute any estimable function. In other words, any linear combination of the rows of H produce a matrix L such that $L\beta$ is estimable (see Goodnight, 1976).

14. SUMMARY

The SWEEP operator and its extensions are indeed versatile tools which not only afford solutions to the normal equations and a gamut of additional statistics but also allow complete insight into the nature of least squares. The general concepts of the SWEEP operator, once mastered, allow the whole least squares process to be visualized. Without this conceptual tool it is extremely difficult to explain such concepts as absorption and what the R notation is testing in terms of the parameters of the model. With the SWEEP tool in mind those concepts are readily grasped.

REFERENCES

Barr, A. J. et al. (1976), A User's Guide to SAS 76, SAS Institute Inc., P. O. Box 10066, Raleigh, N. C. 27605.

Beaton, A. E. (1964), "The Use of Special Matrix Operators in Statistical Calculus," Educational Testing Service, RB-64-51.

Beaton, A. E., Rubin, D. B. and Barone, J. L. (1976), "The Acceptability of Regression Solutions: Another Look at Computational Accuracy," JASA, 71, 158-168.

_____. (1977), "Comment on More on Computational Accuracy in Regression," JASA, 72, 600.

Ben-Israel, Adi, and Greville, T. N. E. (1974), Generalized Inverses Theory and Application, John Wiley and Sons, New York.

Berk, K. N. (1977), "Tolerance and Condition in Regression Computations," JASA, 72, 863-866.

Frane, J. W. (1977), "A Note on Checking Tolerance in Matrix Inversion and Regression," Technometrics, 19, 513-514.

Golub, Gene, and Klema, Virginia. (1978), "Software for Rank Degeneracy," Proceedings of the Computer Science and Statistics: Eleventh Annual Symposium on the Interface, Institute of Statistics, North Carolina State University, Raleigh, N. C., 79-83.

Goodnight, J. H. (1976), "Computational Methods in General Linear Models," ASA 1976 Proceedings of the Statistical Computing Section, ASA, Washington, D. C.

Hemmerle, William J. (1967), Statistical Computations on a Digital Computer, Blaisdell Publishing Company, Waltham, Massachusetts.

Isaacson, E., and Keller, H. B. (1966), Analysis of Numerical Methods, John Wiley and Sons, New York.

Pringle, R. M. and Rayner, A. A. (1971), Generalized Inverse Matrices with Applications to Statistics, Hafner Publishing Company, New York.

Ralston, Anthony. (1960), Mathematical Methods for Digital Computers, p. 191, John Wiley and Sons, New York.

Schatzoff, M., Tsao, R., and Fienberg, S. (1968), "Efficient Calculation of All Possible Regressions," Technometrics, 10, 769-779.

Seber, G. A. F. (1977), Linear Regression Analysis, John Wiley and Sons, New York.

Steel, R. G. D., and Torrie, J. H. (1960), Principles and Procedures of Statistics, McGraw-Hill, New York.

Wilkinson, J. H. (1961), "Error Analysis of Direct Methods of Matrix Inversion," J. Ass. Comp. Mach., #8, 281-330.