

A COMPARISON OF LINEAR LEAST SQUARES COMPUTER PROGRAMS

by

David U. Himmelberger

Remote Sensing Group  
Research and Development Branch  
Standards and Research Division  
Statistical Reporting Service  
U.S. Department of Agriculture

June 1971

## A Comparison of Linear Least Squares Computer Programs

Two linear least squares test problems, both fifth degree polynomials, have been run on more than twenty different computer programs in order to assess their numerical accuracy. Essentially five different algorithms were used in the various programs to obtain the coefficients of the least squares fits. The tests were run on several different computers, in double precision as well as single precision. By comparing the coefficients reported, it was found that those programs using orthogonal Householder transformations or Gram-Schmidt orthonormalization were much more accurate than those using elimination algorithms. Programs using orthogonal polynomials (suitable only for polynomial fits) also proved to be superior to those using elimination algorithms. One program, using congruential methods and integer arithmetic, obtained exact solutions. In a number of programs, the coefficients reported in one test program were sometimes completely erroneous, containing not even one correct significant digit.

### Introduction:

Computational algorithms for linear least squares problems were originally designed for calculators. However, many computer solutions of the linear least squares problems are still obtained from programs using calculator algorithms. The numerical accuracy of these solutions is in many cases a tragedy.

The present study was undertaken to assess the numerical accuracy of the RAX Statistical System program polynomial regression and other representative least squares programs from a variety of sources. Many of the results summarized herein are taken from a study by Wampler. [44]

Two test problems, both fifth degree polynomials, have been run on more than twenty different programs. Included in the study were programs from the BMD Biomedical computer programs collection, the C-E-I-R Multi-Access Computing Services Library, the IBM SHARE Library, the IBM System/360 scientific subroutine package, the Univac MATH-PACK and STAT-PACK collections, and the Project MAC-7094 disk files. A detailed listing of the sources of the programs is given in Appendix A, together with a brief description of each program.

The programs included in this study used essentially five different algorithms: orthogonal Householder transformations, Gram-Schmidt orthonormalization, orthogonal polynomials, Gaussian or Jordan elimination, and a congruential method with computations in integer arithmetic.

Previous studies appraising linear least squares program and comparing the results of different algorithms have been made by Cameron [9], Freund [20], Bright and Dawkins [7], Zellner and Thornber [48], Longley [29], Jordan [27], and Wampler [44].

The linear least squares problem may be briefly stated as follows: One has  $n$  observations or measurements of a "dependent" variable  $y$  which are statistically independent with common variance  $\sigma^2$  whose expected values are given by a linear function of the corresponding

values of  $k$  "independent" variables,  $x_1, x_2, \dots, x_k, k \leq n$ . In matrix notation the  $n$  observations have expected values  $E(Y) = X\beta$ , where  $Y$  is an  $n \times 1$  vector,  $X$  is an  $n \times k$  matrix, and  $\beta$  is a  $k \times 1$  vector of unknown coefficients. Assuming that  $X$  is of rank  $k$ , the least squares estimates of the coefficients are given by  $\hat{\beta} = (X'X)^{-1} X'Y$ . Other quantities of interest are  $\hat{Y} = X \hat{\beta}$ , the vector of predicted values;  $\delta = Y - \hat{Y}$ , the vector of residuals; and

$$s^2 = \frac{1}{n-k} (Y - \hat{Y})' (Y - \hat{Y}),$$

an estimate of the variance  $\sigma^2$ .

The Test Problems:

The two main test problems which were used throughout this investigation are identified as  $Y_1$  and  $Y_2$ . Both were fifth degree polynomials, with the values of  $x$  being the integers 0, 1, 2, ..., 20. The "observations",  $Y_1$  and  $Y_2$ , were calculated from the following equations:

$$Y_1: y = 1 + x + x^2 + x^3 + x^4 + x^5, x = 0(1)20.$$

$$Y_2: y = 1 + 0.1 x + 0.01 x^2 + 0.001 x^3 + 0.0001 x^4 + 0.00001 x^5, x = 0(1)20.$$

Thus the values of  $Y_1$  were integers having one to seven digits, and those of  $Y_2$  were five decimal numbers ranging from 1.00000 to 63.00000.

If the least squares solutions were computed with no rounding error, one would obtain:

$$\beta (Y_1) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \hat{\beta} (Y_2) = \begin{bmatrix} 1. \\ 0.1 \\ .01 \\ .001 \\ .0001 \\ .00001 \end{bmatrix}$$

and for both problems the residual standard deviation would be zero.

The two test problems,  $Y_1$  and  $Y_2$ , were chosen because they are highly ill-conditioned. That is, some programs fail to obtain correct solutions while other programs succeed in obtaining reasonably accurate solutions. Polynomial problems were chosen because polynomial fitting is an important type of linear least squares problem which occurs frequently in practice.

The P-condition defined as:

$$P(A) = \left| \frac{\lambda}{\mu} \right|,$$

where  $\lambda$  is the numerically largest eigenvalue of A and  $\mu$  is the numerically smallest eigenvalue of A, is the criteria which we shall use to measure the condition of matrix A. (See Newman [34, p. 240]).

Most of the programs which were tested obtained more accurate solutions for  $Y_2$  than for  $Y_1$ . If we let A denote the 7 x 7 matrix.

$$A = \begin{bmatrix} X' X \\ Y' X \end{bmatrix} \begin{bmatrix} X' Y \\ 0 \ 0 \end{bmatrix}$$

We find that for  $Y_2$ ,  $P(A) = 4.095 \times 10^{13}$ , whereas for  $Y_1$ ,  $P(A) = 6.829 \times 10^{13}$ , indicating that the system involving  $Y_1$  is more ill-conditioned than that involving  $Y_2$ .

Summary of the Results:

Tables 1 to 6 present a brief summary of the main results. A count,  $C_j$ , of the number of correct significant digits in each computed coefficient was obtained as follows:

Let  $\beta_j$  ( $j = 1, 2, \dots, b$ ) denote the "true" value of the coefficient - that is, the value computed with no rounding error. Let  $\hat{\beta}_j$  denote the value calculated by the computer. Then

$$C_j = \begin{cases} -\log_{10} \left| \frac{\beta_j - \hat{\beta}_j}{\beta_j} \right|, & \text{if } |\beta_j - \hat{\beta}_j| \neq 0 \text{ and } \beta_j \neq 0 \\ -\log_{10} |\beta_j - \hat{\beta}_j|, & \text{if } |\beta_j - \hat{\beta}_j| \neq 0 \text{ and } \beta_j = 0 \\ D, & \text{the approximate number of decimal digits with which the} \\ & \text{machine computes, if } \beta_j - \hat{\beta}_j = 0. \end{cases}$$

The above approach to counting the number of correct digits in a computed value has been used by Jordan [27] and others.

Tables 1 to 6, in the columns headed "Average Number of Correct Digits" report

$$C = 1/6 \sum_{j=1}^6 C_j .$$

Each of the tables (1 through 6) summarizes a set of results for a particular machine precision - 8, 9, 16, 18, etc., digits. Within each table the various programs are ranked for each of the two test problems, with rank 1 denoting the best performance according to the count C.

The symbols in the Algorithm column of the tables denote the following:

- C - Congruential method, interger arithmetic
- E - Elimination method
- GS - Gram-Schmidt orthonormalization
- HT - Orthogonal Householder transformations
- OP - Orthogonal polynomials .

We shall include a discussion of only those programs currently available in SRS and the proposed program by Lautenschlager.<sup>1</sup> For a detailed discussion of the algorithms employed by the other programs summarized in this report see Wampler [44].

---

1 Lautenschlager is a mathematical statistician with the United States Department of Agriculture, Statistical Reporting Service, Washington, D. C.

Programs Using Orthogonal Polynomials:

Since the two test problems are both polynomial fits, we were able to test programs in which the algorithm used orthogonal polynomials. This method, described by Forsythe [18], is attractive because it generally requires many fewer operations than other methods.

S10OPR by Lautenschlager employs orthogonal polynomials and uses a modification of Forsythe's method [18]. From Table 3 we see that the double precision version in 16 digits of S10OPR performed best on test problem Y1, however, ranked last of the three programs on test problem Y2.

Below are listed the actual coefficients and counts obtained for S10OPR.

DOUBLE PRECISION	(16 DIGITS)
BETA-HAT (Y1)	COUNT
0.9999998617649907	6.859
1.0000002746473910	6.561
0.9999999016054061	7.007
1.0000000128933947	7.889
0.9999999992947720	9.152
1.0000000000136755	<u>10.864</u>
AVERAGE =	8.055



DOUBLE PRECISION	(16 DIGITS)
BETA-HAT (Y2)	COUNT
1.0000006249634456	6.204
0.0999986609550127	4.873
0.0100004345935624	4.362
0.0009999576813923	4.374
0.0001000011332467	4.946
0.0000100000037365	<u>6.427</u>
AVERAGE =	5.198

Programs Using Elimination Algorithms:

The majority of the programs tested in this investigation used some form of an elimination algorithm. Although this was the most popular method, it was the least successful. None of these programs performed as well as those using Householders' transformations, Gram-Schmidt orthonormalization (with iteration), or orthogonal polynomials.

Within this class of programs, there were several variations in the method of obtaining the least-squares coefficients.

The programs POLRG and DFQ02 are the polynomial regression programs of the IBM System/360 Scientific Subroutine Package [24, 25] and modified SSP for RAX [24, 25, 38], respectively. These programs call for subroutines, GDATA, ORDER, MINV, AND MULTR, in the course of obtaining the least squares coefficients and other quantities of interest. These subroutines perform the following operations:

(1) GDATA generates the powers of the independent variable, finds means and standard deviations, and sets up a correlation matrix.

(2) ORDER chooses a dependent variable and a subset of independent variables from a larger set of variables.

(3) MINV inverts the correlation matrix using the "Standard Gauss-Jordan method."

(4) MULTR computes the regression coefficients and related quantities, such as the sum of squares attributable to the regression and the sum of squares of deviations from regression.

We see from Table 1 that the single precision version of POLRG obtained rather low scores and the RAX modification DFQ02 was almost the worst of those programs included on both test problems. A double precision version of POLRG was also run, and the performance here as reported in Table 4 was comparable to other programs using similar elimination algorithms.

The user of POLRG or DFQ02 specifies  $m$ , the highest degree polynomial to be fitted, and the program automatically reports the results of fitting polynomials of successively increasing degrees, starting with the first degree. If there is no reduction in the residual sum of squares between two successive degrees of polynomials, the program stops the problem before completing the analysis for the highest degree specified. In running both programs, POLRG and DFQ02, on both test problems, Y1 and Y2, in single precision the analysis stopped after degree four, and in lieu of a fifth degree polynomial fit, the message "NO IMPROVEMENT" was printed. In order

to complete the calculations for the fifth degree, the checks on "improvement" were bypassed. In the double precision version, fifth degree results were obtained without any such alterations in POLRG, DFQ02 was not run in double precision.

The Programmer's Manual for the IBM System/360 Scientific Subroutine Package [25] contains some warnings regarding the accuracy of computations. The reader is informed that the accuracy of the computations in many of the routines is highly dependent upon the number of significant digits available for arithmetic operations. It is pointed out that matrix inversion and many of the statistical subroutines fall into this category, and that the user may, therefore, wish to use double precision versions of these routines. (The programs are so constructed that conversion to double precision is an easy matter.) An appendix of the manual classifies the subroutines of this package into three categories. They are: (1) subroutines having little or no effect on accuracy, (2) subroutines whose accuracy is dependent on the characteristics of the input data, and (3) subroutines in which definite statements on accuracy can be made. Only one of the four subroutines called by the POLRG program, namely ORDER, is in the first category. The other three subroutines, GDATA, MINV, and MULTR, fall in the second category. In connection with this second category we read that "it cannot be assumed that the results are accurate simply because execution is completed."

A more explicit statement is given in connection with the subroutine GDATA. Here there is a comment in the program stating that if in, the highest degree polynomial to be fitted, is equal to 5 or greater, single precision may not be sufficient to give satisfactory results. Since the

manual's test problem for POLRG specifies  $m = 4$  and has 15 data points, one might infer that satisfactory results would be obtained for this problem. This is not the case, however. In the solution to this problem given on page 410 of the manual, the intercept term for the polynomial regression of degree 4 is reported to be -5.26735. An accurate calculation shows that this term is actually -6.04262, so that the reported term had no correct significant digits. The four reported regression coefficients were correctly computed to only one or two digits. Furthermore, the sum of squares of deviations from the regression is reported to be 128.85156, whereas it is actually 17.67310. This error is also propagated into the calculation of the mean square, the F value, and the improvement in terms of sums of squares.

Concluding this discussion of the accuracy of the test problem accompanying the program POLRG, we note a remark given in the Programmer's Manual under "Purposes and Objectives of the Package:" "While this package may provide many of the tools necessary to solve the more commonly encountered problems in engineering and science, there is no intent to imply that these subroutines represent the current state of the art in statistics or numerical analysis."

Below we list the coefficients and counts obtained for POLRG and DFQ02. Similar results for the remaining programs listed in Tables 1-6 may be found in Wampler [44].

POLRG	DOUBLE PRECISION	(18 DIGITS)
BETA-HAT (Y1)		COUNT
1.00000011110114428		6.954
0.999999990457514712		8.020
1.00000000321449412		8.493
0.99999999544922239		9.342
1.0000000002322160		10.634
0.9999999999491835		12.294

AVERAGE = 9.290

BETA-HAT (Y2)		COUNT
1.0000000000136645		11.864
0.099999999999411990		12.231
0.010000000000305940		11.514
0.0009999999995547370		11.351
0.0001000000000227181		11.644
0.00000999999999416480		12.234

AVERAGE = 11.806

DFQ02	SINGLE PRECISION	(8 DIGITS)
BETA-HAT (Y1)		COUNT
374.3750		-3.428
-769.9102		-3.113
285.2087		-3.546
-37.12183		-2.419
3.125595		-1.672
0.9581172		1.378

AVERAGE = -2.133

BETA-HAT (Y2)	COUNT
0.9922342	2.110
0.1436781	0.360
-0.0070678	1.768
0.0032501	1.648
-0.000121566	1.915
0.00001231	0.636

AVERAGE = 1.406

Concluding Remarks:

(1) Of the four procedures using floating--point arithmetic which were included in this study, orthogonal Householder transformations and Gram-Schmidt orthonormalization proved to be the best. Orthogonal polynomials ranked next. Elimination methods were the least successful, but the most popular. The multiple precision integer arithmetic procedure using congruential methods was unique in obtaining exact solutions.

(2) Some other algorithms apparently of high quality which have been published in the last few years were not included in this study. These

include: (a) Bauer [2],

(b) Bjorck and Golub [6],

(c) Bjorck [5].

(3) The importance of accumulating inner products in double precision cannot be overstressed. A number of recent papers on least squares computations have emphasized this point. These include Businger and Golub [9], Bauer [2], Golub and Wilkinson [22], Bjorck and Golub [6], and Bjorck [5].

(4) Iterative refinement is another valuable feature of recent algorithms.

(5) In any mathematical calculation carried out on a computer, it is desirable to know whether an accurate solution has been obtained or whether the result of a calculation is contaminated by rounding error to such an extent that it is worthless. This goal has been achieved in some areas. Martin, Peters, and Wilkinson [31], in their paper giving an algorithm for solving  $Ax = b$ , where  $A$  is  $n \times n$  positive definite matrix and  $b$  is

an  $n \times p$  matrix, state that their procedure "either produces the correctly rounded solutions of the equation  $Ax = b$  or indicates that  $A$  is too ill-conditioned for this to be achieved without working to higher precision (or is possibly singular)." Similarly, Wilkinson's program [45] for the solution of an ill-conditioned system of equations  $Ax = b$ , where  $A$  is  $n \times n$ , "gives either a solution of the system which is correct to working accuracy or alternatively indicates that the system is too ill-conditioned to be solved without working to higher precision and may even be singular."

It appears that the goal set out above has now been achieved in the linear least squares program of Bjorck and Golub [6]. The authors state that their procedure may be used to compute accurate solutions and residuals to linear least squares problems, but that the procedure will fail when  $\chi$  modified by rounding errors has less than full rank, and that it will also fail if  $\chi$  is so ill-conditioned that there is no perceptible improvement in the iterative refinement. The user is easily informed of these situations.

(6) SRS users of DFQ02 should be cognizant of the restrictions outlined in the Programmer's Manual for the IBM System/360 Scientific Subroutine Package modified for RAX regarding the accuracy of computations. Whenever working with a problem similar to either of the test problems, Y1 or Y2, POLRG (available in the Center), or S100PR give more accurate solutions, than DFQ02. S100PR is now stored on the SRS Remote Access Terminal.



Table 1.--Summary of programs run in single precision--8 digits

Program	Com- puter	Algo- rithm <sup>a</sup>	Average number of correct digits		Rank	
			Y1	Y2	Y1	Y2
			ALSQ	1108	HT	4.098
BMD02R	1108	E	-0.106	1.981	12	14
BMD03R	7094	E	0.742	1.721	8	16
BMD03R	1108	E	-0.123	2.287	13	12
DAM	7094	E	1.389	2.312	7	11
DAM	1108	E	-0.264	2.622	16	9
DFQ02	360	E	-2.133	1.406	21	18
LINFIT (Miller)	7094	?	-2.756	-0.301	22	22
LSTSQ	1108	HT	4.528	5.840	1	3
MATH-PACK, ORTHLS	1108	OP	2.118	4.363	6	6
MPR3	7094	E	-0.140	1.856	14	15
OMNITAB (Invert)	7094	E	-0.607	1.460	17	19
OMNITAB (Invert)	1108	E	-0.907	1.224	19	20
OMNITAB (Ortho)	7094	GS	3.954	5.968	4	2
OMNITAB (Ortho)	1108	GS	4.137	5.464	2	4
ORTHO (no iteration)	1108	GS	-1.976	0.419	20	21
ORTHOL	1108	GS	3.593	6.197	5	1
POLRG	1108	E	-0.191	2.280	15	13
SPVMX	1108	E	-0.658	1.527	18	17
STAT-PACK, GLH	1108	E	0.066	2.767	10 $\frac{1}{2}$	7 $\frac{1}{2}$
STAT-PACK, REBSOM	1108	E	0.066	2.767	10 $\frac{1}{2}$	7 $\frac{1}{2}$
STAT-PACK, RESTEM	1108	E	0.651	2.407	9	10
WRAP	7094	E	-5.300	-2.871	23	23

<sup>a</sup> E = Elimination method; GS = Gram-Schmidt orthonormalization;  
HT = Orthogonal Householder transformations; OP = Orthogonal  
polynomials.

Table 2.--Summary of programs run in single precision--9 digits

Program	Com-puter	Algo-rithm <sup>a</sup>	Average number of correct digits		Rank	
			Y1	Y2	Y1	Y2
LINFIT	235	E	0.905	2.894	5	6
LSCF	235	E	0.308	2.483	7	7
LSFITW	235	GS	4.102	6.354	1	1
POLFIT	235	OP	3.349	5.922	2	2
SIMEX	235	E	1.402	3.213	3	3
STAT20	235	E	0.612	2.920	6	5
STAT21	235	E	1.169	3.183	4	4

<sup>a</sup> E = Elimination method; GS = Gram-Schmidt orthonormalization;  
OP = Orthogonal polynomials.

Table 3.--Summary of programs run in double precision--16 digits

Program	Com-puter	Algo-rithm <sup>a</sup>	Average number of correct digits		Rank	
			Y1	Y2	Y1	Y2
BMD05R	7094	E	6.953	6.230	3	2
DPVMIX	1107	E	7.882	9.959	2	1
S10OPR	360	OP	8.055	5.198	1	3

<sup>a</sup> E = Elimination method; OP = Orthogonal polynomials.

Table 4.--Summary of programs run in double precision--18 digits

Program	Com-puter	Algo-rithm <sup>a</sup>	Average number of correct digits		Rank	
			Y1	Y2	Y1	Y2
			ALSQ	1108	HT	12.667
BMD02R	1108	E	9.645	12.865	7	7
BMD05R	1108	E	9.368	11.791	9	10
DPVMTX	1108	E	9.744	13.484	6	6
LSTSQ	1108	HT	14.643	16.293	1	1
MATH-PACK, ORTHLS	1108	OP	12.098	14.461	5	5
ORTHO	1108	GS	13.188	15.514	3	3
ORTHO(no iteration)	1108	GS	7.963	10.354	11	11
ORTHOL	1108	GS	13.212	15.604	2	2
POLRG	1108	E	9.290	11.806	10	9
STAT-PACK, RESTEM	1108	E	9.494	12.019	8	8

<sup>a</sup> E = Elimination method; GS = Gram-Schmidt orthonormalization; HT = Orthogonal Householder transformations; OP = Orthogonal polynomials.

Table 5.--Summary of programs run in single precision (8 digits) with inner products accumulated in double precision (18 digits)

Program	Com-puter	Algo-rithm <sup>a</sup>	Average number of correct digits		Rank	
			Y1	Y2	Y1	Y2
			ALSQ	1108	HT	3.506
LSTSQ	1108	HT	8.000	6.279	1	3
ORTHO	1108	GS	3.904	6.459	2	2

<sup>a</sup> GS = Gram-Schmidt orthonormalization; HT = Orthogonal Householder transformations.

Table 6.--Summary of program run in multiple precision integer arithmetic

Program	Com-puter	Algo-rithm <sup>a</sup>		Average number of correct digits	
				Y1	Y2
SOLVER	1108	C	Rational form Floated form	18.000	17.347

<sup>a</sup> C = Congruential method.

Appendix A. Sources of the Programs, With  
Brief Descriptions.

ALSQ. A FORTRAN IV subroutine to solve the linear least squares problem, written by G. W. Stewart III, Union Carbide Corp., Oak Ridge, Tennessee (present address: The University of Texas, Austin, Texas). This program uses a modification of the Businger-Golub algorithm [8].

BMD02R, Stepwise Regression. One of the Biomedical Computer Programs, written in FORTRAN [15].

BMD03R, Multiple Regression with Case Combinations. One of the Biomedical Computer Programs, written in FORTRAN [15].

BMD05R, Polynomial Regression. One of the Biomedical Computer Programs, written in FORTRAN [15].

DAM. A general purpose computer program for data processing and multiple regression, written in FORTRAN by Rudolf R. Rhomberg, Lorette Boissonneault, and Leonard Harris, International Monetary Fund [36].

DFQ02, Polynomial Regression. One of the programs of the IBM System/360 Scientific Subroutine Package written in FORTRAN II modified for Remote Access Statistical System (RAX) [24, 25, 38].

DPVMTX. A double precision FORTRAN IV program for inverting a matrix or solving a set of linear equations. To a program from the SHARE library (7090-F1 3181INV2 Double Precision Matrix Inversion with Selective Pivot, written by A. R. Sadaka [26]), Sally T. Peavy, National Bureau of Standards, incorporated accuracy checks.

LINFIT. A program which fits a linear function to collected data via least squares. Optional constraints may be applied to the fitting coefficients to make them nonnegative, add to a constant, etc. One of eighteen statistical routines written by James R. Miller [32]. This library of routines exists in the Project.

LINFIT\*\*\*. A program written in BASIC for linear least squares curve fitting and computing correlations. Origin: Dartmouth College, Hanover, N.H. Available in the C-E-I-R Multi-Access Computer Services library [10].

LSCF--\*\*\*. A least squares polynomial curve fitting subroutine written in BASIC. Origin: Dartmouth College, Hanover, N.H. Available in the C-E-I-R Multi-Access Computer Services library [10].

LSFITW\*\*\*. A least squares curve fitting program written in BASIC. Adapted by John B. Shumaker, National Bureau of Standards, from Philip J. Walsh's ORTHO algorithm [42]. Available in the C-E-I-R Multi-Access Computer Services library [10].

LSTSQ. A FORTRAN IV subroutine which solves for X the overdetermined system  $AX = B$  of m linear equations in n unknowns for p right-hand sides. Written by Peter Businger, Computation Center, University of Texas (present address: Bell Telephone Laboratories, Murray Hill, N.J.), using the Businger-Golub algorithm [8].

MATH-PACK, ORTHLS, Orthogonal Polynomial Least-Squares Curve Fitting. One of the Univac 1108 MATH-PACK programs, written in FORTRAN V [40].

MPR3, Stepwise Multiple Regression with Variable Transformations.

A FORTRAN II program written by M. A. Efroymsen, Esso Research and Engineering Co., Madison, N.J., using the Efroymsen algorithm [16]. Available in the SHARE library: 7090-G2 3145MPR3[26].

OMNITAB, a general-purpose computer program for statistical and numerical analysis. Developed at the National Bureau of Standards by Joseph Hilsenrath et al. [23]. Now available in an A.S.A. FORTRAN version, OMNITAB allows the user to communicate with a computer in an efficient manner by means of simple English sentences.

ORTHO. A program written by Philip J. Walsh, National Bureau of Standards (present address: University Computing Co., East Brunswick, N.J.), which uses a Gram-Schmidt orthonormalization process for least squares curve fitting. ORTHO exists as an ALGOL procedure [42], a FORTRAN program, a BASIC program (see LSFITW\*\*\* on page 21), and as a routine of OMNITAB [23], where it is called by the commands FIT and POLYFIT.

ORTHOL. A modification of the Davis-Rabinowitz orthonormalization algorithm [12, 13, 14], written in FORTRAN II by James W. Longley, Bureau of Labor Statistics, Washington, D. C., and Roger A. Blau, Bureau of Labor Statistics and Carnegie-Mellon University, Pittsburgh, Pa. [30].

POLFIT. An anonymous program written in BASIC for least squares polynomial curve fitting using orthogonal polynomials.

POLRG, Polynomial Regression. One of the programs of the IBM System/360 Scientific Subroutine Package written in FORTRAN IV [24, 25].

S10OPR. A program written by Lyle Lautenschlager, Statistical Reporting Service, which uses an orthogonal polynomial approximation method, written in FORTRAN II.

SIMEX-\*\*\*. A program written in BASIC for solving  $n$  simultaneous equations in  $n$  unknowns. Origin: Naval Ordnance Laboratory, Silver Spring, Md. Available in the C-E-I-R Multi-Access Computer Services library [10].

SOLVER. A FORTRAN program written by Morris Newman, National Bureau of Standards, for obtaining the exact solution of the system  $AX = B$ , or the inverse of a matrix  $A$ , by congruential methods [35]. The elements of  $A$  and  $B$  must be integers.

SPVMIX. A single precision FORTRAN IV program for inverting a matrix or solving a set of linear equations. To a program from the SHARE library (7090-F1 3180INV1 Single Precision Matrix Inversion with Selective Pivoting written by A. R. Sadaka [25], Sally T. Peavy, National Bureau of Standards, incorporated accuracy checks.

STAT-PACK, GLH, General Linear Hypotheses. One of the Univac 1108 STAT-PACK programs, written in FORTRAN V [42].

STAT-PACK, REBSOM, Back Solution Multiple Regression. One of the Univac 1108 STAT-PACK programs, written in FORTRAN V [42].

STAT-PACK, RESTEM, Stepwise Multiple Regression. One of the Univac 1108 STAT-



PACK programs, written in FORTRAN V [42].

STAT 20\*\*\*. A program written in BASIC for stepwise multiple linear regression. Written by Thomas E. Kurtz, Dartmouth College, Hanover, N.H. Available in the C-E-I-R Multi-Access Computer Services library [10].

STAT21\*\*\*. A program written in BASIC for multiple linear regression with detailed output. Written by Gerald Childs, Dartmouth College, N.H. Available in the C-E-I-R Multi-Access Computer Services library [10].

WRAP, Weighted Regression Analysis Program. A FORTRAN II program written by M. D. Fimple, Sandia Corp., Albuquerque, New Mexico. Available in the SHARE library: 7090-G2 3231 WRAP [26].

References

- [1] Anscombe, F. J., Topics in the investigation of linear relations fitted by the method of least squares, *Journal of the Royal Statistical Society, Series B*, 29, p. 1-52, 1967.
- [2] Bauer, F. L., Elimination with weighted row combinations for solving linear equations and least squares problems, *Numerische Mathematik* 7, p. 338-352, 1965.
- [3] Björck, Ake, Solving linear least squares problems by Gram-Schmidt orthogonalization, *BIT (Nordisk tidskrift for informations-behandling)*, 7, p. 1-21, 1967.
- [4] Björck, Ake, Iterative refinement of linear least squares solutions, I, *BIT* 7, p. 257-278, 1967.
- [5] Björck, Ake, Iterative refinement of linear least squares solutions, II, *BIT* 8, p. 8-30, 1968.
- [6] Björck, Ake, and Golub, Gene, *ALGOL Programming, Contribution No. 22: Iterative refinement of linear least square solutions by Householder transformation*, *BIT* 7, p. 322-337, 1967.
- [7] Bright, J. W., and Dawkins, G. S., Some aspects of curve fitting using orthogonal polynomials, *Industrial and Engineering Chemistry Fundamentals* 4, p. 93-97, 1965.
- [8] Businger, Peter, and Golub, Gene H., Linear least squares solutions by Householder transformations, *Numerische Mathematik* 7, p. 269-276, 1965.
- [9] Cameron, J. M., Some examples of the use of high speed computers in statistics, *Proceedings of the First Conference on the Design of Experiments in Army Research, Development and Testing, Office of Ordnance Research, Report No. 57-1*, p. 129-135, 1957.
- [10] C-E-I-R Multi-Access Computer Services, Library programs documentation, *MAC 71-7-1*, Nov. 17, 1967. Addendum, *MAC 71-7-1, A 12-368*, Mar. 1, 1968.
- [11] Crisman, P. A., (ed.), *The compatible time-sharing system: A programmer's guide*, second edition, M.I.T. Computation Center, M.I.T. Press, Massachusetts Institute of Technology, Cambridge, Mass., 1965.
- [12] Davis, Philip J., Orthonormalizing codes in numerical analysis, ch. 10 in [39], 1962.
- [13] Davis, Philip, and Rabinowitz, Philip, A multiple purpose orthonormalizing code and its uses, *Journal of the Association for Computing Machinery* 1, p. 183-191, 1954.

- [14] Davis, Philip J., and Rabinowitz, Philip, Advances in orthonormalizing computation, in Franz L. Alt (ed.), Advances in Computers, p. 55-133, Vol. 2, (Academic Press, New York, 1961).
- [15] Dixon, W. J., (ed), BMD Biomedical Computer Programs, Health Sciences Computing Facility (University of California, Los Angeles, 1964. Revised 1965 and 1967).
- [16] Efrogmson, M. A., Multiple regression analysis, in Anthony Ralston and Herbert S. Wilf (editors), Mathematical Methods for Digital Computers, Ch. 17, Vol. 1, pp. 191-203 (John Wiley & Sons, New York, N.Y. 1960).
- [17] Fettis, Henry E., and Caslin, James C., Eigenvalues and eigenvectors of Hilbert matrices of order 3 through 10, Mathematics of Computation 21, p. 431-441, 1967.
- [18] Forsythe, George E., Generation and use of orthogonal polynomials for data-fitting with a digital computer, Journal of the Society for Industrial and Applied Mathematics 5, p. 74-88, 1957.
- [19] Forsythe, George E., Today's computational methods of linear algebra, SIAM Review 9, p. 489-515, 1967.
- [20] Freund, R. J., A warning of roundoff errors in regression, The American Statistician 17, p. 13-15, 1963.
- [21] Golub, G., Numerical methods for solving linear least squares problems, Numerische, Mathematik 7, p. 206-216, 1965.
- [22] Golub, G. H., and Wilkinson, J. H., Note on the iterative refinement of least squares solution, Numerische Mathematik 9, p. 139-148, 1966.
- [23] Hilsenrath, Joseph, Ziegler, Guy G., Messina, Carla G., Walsh, Philip J., and Herbold, Robert, OMNITAB, A Computer Program for Statistical and Numerical Analysis, National Bureau of Standards Handbook 101, U.S. Government Printing Office, Washington, D. C., 1966. Reissued Jan. 1968, with corrections.
- [24] IBM Application Program, System/360 Scientific Subroutine Package (360A-CM-03X) Version III, Application Description, H20-0166-5, International Business Machines Corp., 1968.
- [25] IBM Application Program, System/360 Scientific Subroutine Package (360A-CM-03X) Version III, Programmer's Manual, H20-0205-3, International Business Machines Corp., 1968.

- [26] IBM Systems Reference Library, Catalog of programs for IBM 704-709-7040-7044-7090 and 7094 Data Processing Systems, Form C20-1604-3, International Business Machines Corp., December 1965.
- [27] Jordan, T. L., Experiments on error growth associated with some linear least squares procedures, *Mathematics of Computation* 22, p. 579-588, 1968.
- [28] Kurtz, Thomas E., personal communication, 1968.
- [29] Longley, James W., An appraisal of least squares programs for the electronic computer from the point of view of the user, *Journal of the American Statistical Association* 62, p. 819-841, 1967.
- [30] Longley, James W., personal communication, 1968.
- [31] Martin, R. S., Peters, G., and Wilkinson, J. H., Iterative refinement of the solution of a positive definite system of equations, *Numerische Mathematik* 8, p. 203-216, 1966.
- [32] Miller, James R., On-line analysis for social scientists, MAC-TR-40, Project MAC, Massachusetts Institute of Technology, Cambridge, Mass., 1967.
- [33] Moler, Cleve B., Iterative refinement in floating point, *Journal of the Association for Computing Machinery* 14, p. 316-321, 1967.
- [34] Newman, Morris, Matrix computations, ch. 6 in [39], 1962.
- [35] Newman, Morris, Solving equations exactly, *J. Res. NBS* 71B (Math. and Math. Phys.), No. 4, 171-179, 1967.
- [36] Rhomberg, Rudolf R., Boissonneault, Lorette, and Harris, Leonard, A general purpose computer program for data processing and multiple regression (DAM), Revision 3, Research and Statistics Department, International Monetary Fund, Washington, D. C., Sept. 1965.
- [37] Stiefel, Eduard L., *An Introduction to Numerical Mathematics*, (Academic Press, New York, 1963).
- [38] Tauchi, H. J., IBM Corporation, Remote Statistical Access System, 360 D 13.0.003, 1930 Century Park West, Los Angeles, California 90067
- [39] Taussky, Olga, Some topics concerning bounds for eigenvalues of finite matrices, ch. 8 in [40], 1962.

- [40] Todd, John (ed.), Survey of Numerical Analysis (McGraw-Hill Book Co., New York, 1962).
- [41] Univac 1108 Multi-Processor System, MATH-PACK Programmers Reference, UP-7542, Univac Division of Sperry Rand Corporation, 1967.
- [42] Univac 1108 Multi-Processor System, STAT-PACK Programmers Reference, UP-7502, Univac Division of Sperry Rand Corporation, 1967.
- [43] Walsh, Philip J., Algorithm 127, ORTHO, Communications of the ACM 5, p. 511-513, 1962.
- [44] Wampler, R. H., An evaluation of Linear Least Squares Computer Programs, Institute for Basic Standards, National Bureau of Standards, Washington, D. C. 20234.
- [45] Wilkinson, J. H., Rounding Errors in Algebraic Processes (Prentice-Hall, Englewood Cliffs, N. J., 1963).
- [46] Wilkinson, J. H., The Algebraic Eigenvalue Problem (Clarendon Press, Oxford, 1965).
- [47] Wilkinson, J. H., The solution of ill-conditioned linear equations, in Anthony Ralston and Herbert S. Wilf (editors), Mathematical Methods for Digital Computers, ch. 3, p. 65-93, Vol. 2, (John Wiley & Sons, New York, 1967).
- [48] Zellner, A., and Thornber, H., Computational accuracy and estimation of simultaneous equation econometric models, Econometrica 34, pp. 727-729, 1966.