C/325

ORNL-3430 MASTER
UC-32 - Mathematics and Computers

A COMPARISON OF SEVERAL METHODS FOR
INVERTING LARGE SYMMETRIC POSITIVE
DEFINITE MATRICES

M. H. Lietzke
R. W. Stoughton
Marjorie P. Lietzke

OAK RIDGE NATIONAL LABORATORY
operated by
UNION CARBIDE CORPORATION
for the
U.S. ATOMIC ENERGY COMMISSION

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Facsimile Price \$ 3.60

Microfilm Price \$ 1.22

Available from the
Office of Technical Services
Department of Commerce
Washington 25, D. C.

ORNL-3430

Contract No. W-7405-eng-26

CHEMISTRY DIVISION

A COMPARISON OF SEVERAL METHODS FOR INVERTING
LARGE SYMMETRIC POSITIVE DEFINITE MATRICES

M. H. Lietzke
R. W. Stoughton
Marjorie P. Lietzke

Date Issued

MAY 23 1963

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee
operated by
UNION CARBIDE CORPORATION
for the
U. S. ATOMIC ENERGY COMMISSION

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

A Comparison of Several Methods for Inverting Large
Symmetric Positive Definite Matrices

M. H. Lietzke, R. W. Stoughton, and Marjorie P. Lietzke

ABSTRACT

Four methods for inverting large symmetric positive definite matrices have been compared. These are the Gauss-Jordan, Choleski, congruent transformation, and rank annihilation methods. An attempt has been made to consider the effects of both condition and order of the matrix to be inverted on the closeness of the computed inverse to the exact inverse. Each method was programmed in FORTRAN using only single precision arithmetic. Then the error indicators were computed using double precision arithmetic so that the latter calculation was not a limiting factor. It was found that any one of the methods is satisfactory for a well-conditioned least squares matrix of order 10-30. However, when the matrix becomes ill-conditioned then the Gauss-Jordan method appears to be clearly superior (at least for the matrices studied).

Algorithms and FORTRAN subroutines for each method are presented in the appendices.

In curve fitting and in the quantitative treatment of experimental data the method of least squares is widely used. At the present time attempts are being made, for example in the fitting of spectral data, to determine ever larger numbers of parameters. In many cases the underlying mathematical models are highly non-linear, and convergence difficulties may be encountered. In any least squares fit involving the determination of a large number of parameters, however, the accumulation of round-off error during the course of the computation, resulting in a loss of significance, is one of the most serious restrictions. Since the evaluation of a large number of parameters and their variances by a least squares procedure on a high speed computer involves the inversion of a symmetric, positive-definite matrix it becomes important to choose an inversion scheme in which the effects of the accumulation of round-off error are minimized. This is especially true if limited memory space for storage precludes the use of double precision arithmetic. In this report a comparison of several direct methods for inverting such matrices is given. An attempt has been made to consider the effects of both condition and order of the matrix to be inverted on the closeness of the computed inverse to the exact inverse. Since the method of least squares has been described elsewhere¹, it will not be repeated here.

The matrix inversion methods compared are the Gauss-Jordan², Choleski³, congruent transformation⁴, and rank annihilation⁵ schemes. To give a fair comparison of the methods each was programmed in IBM 7090 FORTRAN II (Version 2) using only single precision arithmetic (good to 26 binary or about 8 decimal digits). The error indicators were then computed using double precision arithmetic (good to about 16 decimal

digits) so that the latter calculation was not a limiting factor. All the methods used are well known and need not be described here. In the following discussion the symbols for matrices will be underlined while other symbols will not be.

The first matrix chosen for testing the inversion routines was $\underline{A}^*$ where the elements a_{ij} of $\underline{A}$ are given by

$$a_{ij} = \begin{cases} 2 & (i=j) \\ -1 & (|i-j| = 1) \\ 0 & (|i-j| > 1) \end{cases}$$

The inverse of $\underline{A}$ is given by $\frac{\underline{C}}{(n+1)}$ where the elements c_{ij} of $\underline{C}$ are given by

$$c_{ij} = \begin{cases} i(n-i+1) & (i = j) \\ c_{i,j-1}^{-1} & (j > i) \\ c_{ji} = c_{ij} & (j < i) \end{cases}$$

and n is the order of the matrix. $\underline{A}$ is a symmetric positive definite matrix; it has a P-condition number⁶ of approximately $\frac{4}{\pi^2} n^2$. In respect to both form and condition the matrix $\underline{A}$ is analogous to a least squares matrix derived from a well designed experiment. In addition to $\underline{A}$ the matrices $\underline{A}^2$ and $\underline{A}^3$ were also inverted. The inverse of $\underline{A}^2$ is given by $\frac{\underline{C}^2}{(n+1)^2}$, while the inverse of $\underline{A}^3$ is given by $\frac{\underline{C}^3}{(n+1)^3}$. The latter two matrices have P-condition numbers of $\frac{6}{\pi^4} n^4$ and $\frac{64}{\pi^6} n^6$ respectively and are hence progressively more ill-conditioned. In this respect they correspond to least squares matrices derived from increasingly more poorly designed experiments.

* The negative of this matrix was discussed by Newman and Todd⁶.

In addition to the three matrices just mentioned two other test matrices were used. They are B, whose elements b_{ij} are given by

$$b_{ij} = \begin{cases} 2 & (i = j) \\ 1 & (i \neq j), \end{cases}$$

and D, whose elements d_{ij} are given by

$$d_{ij} = n - |i - j|.$$

The exact inverse of B is given by BINV whose elements binv_{ij} are

$$\text{binv}_{ij} = \begin{cases} \frac{n}{n+1} & \text{if } i = j \\ -\frac{1}{n+1} & \text{if } i \neq j, \end{cases}$$

while the exact inverse of D is given by DINV whose elements dinv_{ij} are

$$\text{dinv}_{ij} = \begin{cases} \frac{n+2}{2n+2} & \text{if } i=j = 1 \text{ or } n \\ 1 & \text{if } i=j \neq 1 \text{ or } n \\ -1/2 & \text{if } |i-j| = 1 \text{ and } n \neq 2 \\ -1/3 & \text{if } |i-j| = 1 \text{ and } n=2 \\ \frac{1}{2n+2} & \text{if } |i-j| = n-1 = 1 \\ 0 & \text{if } 1 < |i-j| < n-1 \end{cases}$$

In carrying out the comparison of the various inversion methods the test matrix was generated and inverted by each method to give $(\underline{M})_{\text{approx}}^{-1}$. Then the exact inverse was generated (using the above formulas and double precision arithmetic) and as a measure of error the quantity Q defined as

$$Q = \frac{1}{n^2} \sum_{ij} |(\underline{M})_{\text{exact}}^{-1} - (\underline{M})_{\text{approx}}^{-1}|_{ij} \quad (1)$$

was computed for each method, where $\underline{M}$ represents the matrix inverted. The computations were performed with matrices of order 10, 15, 20, 25, and 30. In addition to Q two other quantities, recommended as error measures by Newman and Todd⁶,

$$a = \frac{1}{n^2} \sum_{ij} |r_{ij}| \quad (2)$$

and

$$f = \frac{1}{n} \left\{ \sum_{ij} r_{ij}^2 \right\}^{1/2} \quad (3)$$

were calculated, where $\underline{R} = (r_{ij})$ is the error matrix taken as

$$\underline{R} = (\underline{M})_{\text{approx}}^{-1} (\underline{M}) - \underline{I} \quad (4)$$

where $\underline{I}$ is the identity matrix. The number Q obviously reflects the per term average of the overall difference between the approximate inverse matrix and the exact inverse matrix.

Results of the Computations

The A Matrix

All the methods tried gave an approximate inverse that was very close to the exact inverse of the $\underline{A}$ matrix for all orders from 10 to 30 except the rank annihilation method, in which case the routine used failed to invert the $\underline{A}$ matrix. The reason for this failure was evidently peculiar to the exact integers in the matrix and the particular computer procedure for the rank annihilation method; the procedure could undoubtedly be rewritten to obviate this particular failure although it was not considered warranted in the current comparative study. Hence any one of the methods would be suitable for a well-conditioned least squares matrix in this range. All elements of the inverse matrix

produced by the congruent transformation method were good to one in the eighth significant figure even for order 30. The Gauss-Jordan method and the Choleski method were close behind in that order. Not only did the congruent transformation method produce an inverse closest to the exact inverse of the $\underline{A}$ matrix but the values of a and f (as defined by equations 2 and 3) computed from the approximate inverse were lower than the corresponding values derived from the Gauss-Jordan and Choleski inversion routines as can be seen in Tables IA-ID. All the methods produced inverse matrices, all elements of which were smaller in magnitude than the exact inverse for all orders from 10 to 30. (The reason for this is unknown.)

The A^2 Matrix

In the inversion of the $\underline{A^2}$ matrix the effect of the condition of the matrix starts to become apparent. This matrix is not as well conditioned as the $\underline{A}$ matrix; in fact, its condition number is the square of the condition number for the $\underline{A}$ matrix and hence is proportional to the fourth power of the order.

In the inversion of the $\underline{A^2}$ matrix for all orders from 10 to 30 the Gauss-Jordan method produced an inverse matrix closest to the exact inverse. The congruent transformation, the Choleski, and the rank annihilation methods gave very similar results and were somewhat inferior to the Gauss-Jordan method. All elements of the inverse matrix produced by the latter method were good to one part in 600,000 for order 10 to one part in 20,000 for order 30. The corresponding figures for the other methods were one part in about 150,000 for $n = 10$ to one part in about 3000 for $n = 30$. The values of Q , a , and f for the $\underline{A^2}$ matrix are shown in Table I.

The A^3 Matrix

An examination of the values of Q for the A^3 matrix (Tables IA-ID) indicates that the Gauss-Jordan routine produced an inverse matrix closest to the exact inverse for all orders from 10 to 30. All elements of the inverse matrix produced by this routine are good to one part in 80,000 for $n = 10$, to one part in 1100 for $n = 20$ and one part in 400 for $n = 30$. Close behind the Gauss-Jordan routine for $n = 10$ is the congruent transformation method; however, for $n = 15$ to 30 the Choleski method produced a more nearly exact inverse. At $n = 30$ the rank annihilation method appears slightly better than the congruent transformation method, producing an inverse each element of which is good to one part in 16.

In addition to the above observations concerning the closeness of the approximate inverse of the A^3 matrix to the exact inverse it should be noted that the Gauss-Jordan and congruent transformation routines produced inverse matrices whose elements were in all cases smaller in magnitude than those of the exact inverse, while the inverse matrices produced by the rank annihilation method were for all orders greater than the exact inverse. The Choleski routine produced an inverse matrix at orders 10 and 15 with elements smaller than those of the exact inverse, while at orders 20, 25 and 30 the inverse matrix elements were greater than those of the exact inverse. The low value of Q at $n = 15$ for the Choleski method reflects the fact that the elements of the approximate inverse are passing through a transition from all being too small to all being too large.

The B Matrix

The B matrix is apparently so well conditioned that all four methods produced inverse matrices very close to the true inverse as can be seen in Table I. In fact the inverse matrix produced by the rank annihilation routine varied at most by one in the eighth significant figure from the exact inverse, and the values of Q computed were in all cases 0.0 to eight decimal digits.

The D Matrix

The D matrix is also well conditioned and all the methods except the rank annihilation method produced inverse matrices close to the exact inverse. The rank annihilation routine failed to invert properly the D matrix, undoubtedly because of the peculiar form of this matrix and the peculiarities of the computer program as written. The values of a , f , and Q for the D matrix are shown in Tables IA-ID.

A Comparison of the Indices of Error a , f and Q

The following observations may be made concerning the results presented in Tables IA-ID:

- 1) The value of a is in all cases lower than the value of f .
- 2) The values of a and f are not, by themselves, reliable estimates of the closeness of the computed inverse to the exact inverse. For example, if the a and f values alone were used to compare the approximate inverses of the A³ matrix produced by the various inversion methods then the congruent transformation method would appear to be as good as the Gauss-Jordan method. However, the Gauss-Jordan routine produced an inverse for $n = 30$ good to one part in 400 while the inverse produced by the congruent transformation method was good only to one part in 15.

Thus the Q values (which follow this pattern) reflect the observed relation between the exact and approximate inverse matrices while the a and f values may not. A possible explanation in the latter case may be in the fact that the $\underline{A}^3$ matrix contains elements of alternating sign while all the elements of the inverse matrix are positive. Hence there may be some cancellation of errors in forming the $(\underline{M})_{\text{approx}}^{-1} \underline{M}$ product in equation (4), this effect becoming more pronounced as the matrix becomes more ill-conditioned. Also the $\underline{A}^3$ matrix contains many zeros (in fact a higher proportion the larger the order) and hence the errors in some of the inverse matrix elements do not show up in the a and f values.

Table IA

A Comparison of the a, f, and Q Values as a Function of Matrix Order and Inversion Method

n	Matrix	<u>Gauss Jordan Method</u>				
		<u>A</u>	<u>A²</u>	<u>A³</u>	<u>B</u>	<u>D</u>
10	a	1.3×10^{-8}	9.6×10^{-6}	6.7×10^{-4}	2.2×10^{-8}	7.3×10^{-8}
	f	1.9×10^{-8}	1.4×10^{-5}	1.1×10^{-3}	2.7×10^{-8}	9.1×10^{-8}
	Q	4.8×10^{-8}	2.7×10^{-5}	2.1×10^{-3}	5.5×10^{-9}	3.3×10^{-8}
15	a	2.3×10^{-8}	3.6×10^{-5}	3.0×10^{-2}	3.0×10^{-8}	1.5×10^{-7}
	f	3.6×10^{-8}	7.1×10^{-5}	4.7×10^{-2}	3.8×10^{-8}	1.9×10^{-7}
	Q	7.8×10^{-8}	8.3×10^{-5}	2.6×10^{-2}	6.4×10^{-9}	4.8×10^{-8}
20	a	3.7×10^{-8}	1.1×10^{-4}	5.0×10^{-1}	4.0×10^{-8}	2.2×10^{-7}
	f	6.1×10^{-8}	1.6×10^{-4}	8.1×10^{-1}	5.3×10^{-8}	2.8×10^{-7}
	Q	1.2×10^{-7}	1.1×10^{-3}	3.4	6.6×10^{-9}	7.4×10^{-7}
25	a	4.3×10^{-8}	4.3×10^{-4}	3.4	5.0×10^{-8}	3.7×10^{-7}
	f	7.1×10^{-8}	7.2×10^{-4}	5.8	6.4×10^{-8}	4.8×10^{-7}
	Q	2.0×10^{-7}	5.2×10^{-3}	27.	6.6×10^{-9}	9.2×10^{-8}
30	a	5.1×10^{-8}	1.3×10^{-3}	13.	6.1×10^{-8}	5.3×10^{-7}
	f	8.2×10^{-8}	2.0×10^{-3}	23.7	7.8×10^{-8}	6.9×10^{-7}
	Q	3.9×10^{-7}	1.9×10^{-2}	69.	6.5×10^{-9}	1.2×10^{-7}

Table IB

A Comparison of the a, f, and Q Values as a Function of Matrix Order and Inversion Method

n	Matrix	<u>Congruent Transformation Method</u>				
		<u>A</u>	<u>A²</u>	<u>A³</u>	<u>B</u>	<u>D</u>
10	a	1.0×10^{-8}	1.2×10^{-6}	5.6×10^{-5}	1.5×10^{-8}	6.8×10^{-8}
	f	1.5×10^{-8}	1.4×10^{-6}	7.2×10^{-5}	2.5×10^{-8}	1.1×10^{-7}
	Q	5.9×10^{-9}	1.0×10^{-4}	1.3×10^{-2}	7.9×10^{-9}	2.6×10^{-8}
15	a	0.0	4.3×10^{-6}	5.2×10^{-4}	1.8×10^{-8}	1.1×10^{-7}
	f	0.0	5.6×10^{-6}	7.1×10^{-4}	3.0×10^{-8}	1.7×10^{-7}
	Q	0.0	1.2×10^{-3}	1.3	9.2×10^{-9}	2.9×10^{-8}
20	a	1.6×10^{-8}	9.1×10^{-6}	2.3×10^{-3}	2.1×10^{-8}	1.4×10^{-7}
	f	2.6×10^{-8}	1.2×10^{-5}	3.2×10^{-3}	3.9×10^{-8}	2.5×10^{-7}
	Q	1.0×10^{-8}	7.4×10^{-3}	29.	1.0×10^{-8}	5.4×10^{-8}
25	a	1.8×10^{-8}	1.5×10^{-5}	5.7×10^{-3}	2.3×10^{-8}	1.7×10^{-7}
	f	3.1×10^{-8}	2.0×10^{-5}	7.5×10^{-3}	4.3×10^{-8}	3.1×10^{-7}
	Q	6.2×10^{-8}	2.7×10^{-2}	229.	1.0×10^{-8}	6.4×10^{-8}
30	a	1.9×10^{-8}	2.9×10^{-5}	1.5×10^{-2}	2.4×10^{-8}	2.1×10^{-7}
	f	3.2×10^{-8}	4.0×10^{-5}	2.3×10^{-2}	4.6×10^{-8}	4.3×10^{-7}
	Q	2.2×10^{-8}	1.1×10^{-1}	1885.	1.0×10^{-8}	7.3×10^{-8}

Table IC

A Comparison of the a, f, and Q Values as a Function of Matrix Order and Inversion Method

n	Matrix	<u>Choleski Method</u>				
		<u>A</u>	<u>A²</u>	<u>A³</u>	<u>B</u>	<u>D</u>
10	a	2.0×10^{-8}	1.6×10^{-6}	8.7×10^{-5}	2.7×10^{-8}	8.0×10^{-8}
	f	3.1×10^{-8}	2.2×10^{-6}	1.1×10^{-4}	3.7×10^{-8}	1.0×10^{-7}
	Q	1.3×10^{-7}	1.1×10^{-4}	2.1×10^{-2}	7.8×10^{-9}	4.5×10^{-8}
15	a	4.0×10^{-8}	8.9×10^{-6}	1.6×10^{-3}	3.7×10^{-8}	1.6×10^{-7}
	f	5.8×10^{-8}	1.4×10^{-5}	2.2×10^{-3}	4.8×10^{-8}	2.2×10^{-7}
	Q	2.4×10^{-7}	1.7×10^{-3}	1.3×10^{-1}	9.2×10^{-9}	4.6×10^{-8}
20	a	5.7×10^{-8}	1.8×10^{-5}	1.4×10^{-2}	4.3×10^{-8}	3.3×10^{-7}
	f	8.8×10^{-8}	2.7×10^{-5}	1.9×10^{-2}	5.8×10^{-8}	4.5×10^{-7}
	Q	2.4×10^{-7}	1.1×10^{-2}	8.3	9.9×10^{-9}	8.4×10^{-8}
25	a	7.4×10^{-8}	4.0×10^{-5}	6.4×10^{-2}	5.2×10^{-8}	5.9×10^{-7}
	f	1.2×10^{-7}	5.5×10^{-5}	9.6×10^{-2}	6.8×10^{-8}	7.9×10^{-7}
	Q	3.7×10^{-7}	4.5×10^{-2}	130.	1.2×10^{-8}	1.3×10^{-7}
30	a	8.9×10^{-8}	1.2×10^{-4}	2.2×10^{-1}	6.1×10^{-8}	9.6×10^{-7}
	f	1.4×10^{-7}	1.7×10^{-4}	3.7×10^{-1}	7.9×10^{-8}	1.2×10^{-6}
	Q	6.4×10^{-7}	1.4×10^{-1}	751.	1.5×10^{-8}	1.6×10^{-7}

Table ID

A Comparison of the a, f, and Q Values as a Function of Matrix Order and Inversion Method

n	Matrix	<u>Rank Annihilation Method</u>		
		<u>A²</u>	<u>A³</u>	<u>B</u>
10	a	2.2×10^{-5}	1.5×10^{-3}	1.3×10^{-8}
	f	2.6×10^{-5}	2.6×10^{-3}	1.6×10^{-8}
	Q	8.8×10^{-5}	2.6×10^{-1}	0.0
15	a	4.2×10^{-5}	6.4×10^{-3}	0.0
	f	5.9×10^{-5}	1.1×10^{-2}	0.0
	Q	1.5×10^{-3}	9.1	0.0
20	a	7.5×10^{-5}	5.2×10^{-2}	3.1×10^{-8}
	f	1.0×10^{-4}	9.4×10^{-2}	4.2×10^{-8}
	Q	1.2×10^{-2}	97.	0.0
25	a	1.9×10^{-4}	2.6×10^{-1}	3.6×10^{-8}
	f	2.7×10^{-4}	4.7×10^{-1}	5.0×10^{-8}
	Q	3.9×10^{-2}	440.	0.0
30	a	4.0×10^{-4}	9.1×10^{-1}	4.8×10^{-8}
	f	6.4×10^{-4}	1.4	7.1×10^{-8}
	Q	9.0×10^{-2}	1386.	0.0

As shown above, the value of Q, defined by equation (1), reflects the overall difference between the approximate inverse matrix and the exact inverse matrix. This quantity can be computed only if the exact inverse matrix is known, as it is in the case of the test matrices used in this report. In general, however, this will not be the case and yet it is desirable to have some method for estimating at least an upper bound of the difference which does not depend on knowing the exact inverse matrix. Such an upper bound is given by the following inequality⁷

$$|| \underline{A}^{-1} - \underline{C} || \leq \frac{|| \underline{CH} ||}{1 - || \underline{H} ||} \quad (5)$$

where $\underline{A}^{-1}$ is the exact inverse of the matrix $\underline{A}$, $\underline{C}$ is the computed inverse, and $\underline{H}$ is a matrix defined by

$$\underline{H} = \underline{I} - \underline{AC}$$

where $\underline{I}$ is the identity matrix. (Enclosure within the double vertical lines indicates that the maximum row sum of magnitudes of the elements of the appropriate matrix is taken.)

For comparison purposes values of both the left L and right R side of the inequality (5) were computed for all the test matrices used and the results are presented in Table II.

As shown in Table II the right side of the inequality (5) is indeed usually greater than the term on the left, which is the maximum row sum of the matrix formed by subtracting the computed inverse from the exact inverse. In the few cases shown in Table II where the inequality does not seem to hold the apparent violation can probably be attributed to the particular method of rounding used in the arithmetic sub-routines of the program and the consequent accumulation of round-off error.

Table II
Values of the Two Sides of the Inequality

$$\| \underline{A}^{-1} - \underline{C} \| \leq \frac{\| \underline{C} \underline{H} \|}{1 - \| \underline{H} \|}$$

n	Matrix	<u>Gauss-Jordan</u>				
		<u>A</u>	<u>A</u> ²	<u>A</u> ³	<u>B</u>	<u>D</u>
10	L	8.3 x 10 ⁻⁷	4.1 x 10 ⁻⁴	3.1 x 10 ⁻²	1.2 x 10 ⁻⁷	5.4 x 10 ⁻⁷
	R	9.6 x 10 ⁻⁷	5.8 x 10 ⁻⁴	4.7 x 10 ⁻¹	2.1 x 10 ⁻⁷	8.9 x 10 ⁻⁷
15	L	2.5 x 10 ⁻⁶	2.4 x 10 ⁻³	6.7 x 10 ⁻¹	2.0 x 10 ⁻⁷	1.5 x 10 ⁻⁶
	R	3.7 x 10 ⁻⁶	9.9 x 10 ⁻³	34.5	3.2 x 10 ⁻⁷	1.9 x 10 ⁻⁶
20	L	4.7 x 10 ⁻⁶	3.5 x 10 ⁻²	103.	2.5 x 10 ⁻⁷	2.8 x 10 ⁻⁶
	R	8.7 x 10 ⁻⁶	6.7 x 10 ⁻²	871.	4.9 x 10 ⁻⁷	4.1 x 10 ⁻⁶
25	L	7.9 x 10 ⁻⁶	2.0 x 10 ⁻¹	1034.	3.5 x 10 ⁻⁷	4.4 x 10 ⁻⁶
	R	2.0 x 10 ⁻⁵	3.9 x 10 ⁻¹	31107.	6.4 x 10 ⁻⁷	6.4 x 10 ⁻⁶
30	L	1.7 x 10 ⁻⁵	8.6 x 10 ⁻¹	--	4.0 x 10 ⁻⁷	6.2 x 10 ⁻⁶
	R	3.4 x 10 ⁻⁵	2.0	--	7.7 x 10 ⁻⁷	8.6 x 10 ⁻⁶

Table II (continued)

n	<u>Matrix</u>	<u>Congruent Transformation</u>				
		<u>A</u>	<u>A²</u>	<u>A³</u>	<u>B</u>	<u>D</u>
10	L	1.2×10^{-7}	1.4×10^{-3}	1.9×10^{-1}	1.4×10^{-7}	4.6×10^{-7}
	R	4.9×10^{-7}	1.1×10^{-3}	3.2×10^{-1}	1.9×10^{-7}	9.1×10^{-7}
15	L	0.0	2.7×10^{-2}	27.7	2.2×10^{-7}	5.4×10^{-7}
	R	0.0	3.0×10^{-2}	41.9	3.2×10^{-7}	2.1×10^{-8}
20	L	4.4×10^{-7}	2.2×10^{-1}	873.	3.4×10^{-7}	1.9×10^{-8}
	R	4.6×10^{-6}	2.3×10^{-1}	1106.	4.8×10^{-7}	3.8×10^{-6}
25	L	2.8×10^{-6}	1.0	8678.	4.5×10^{-7}	3.0×10^{-8}
	R	6.5×10^{-6}	1.1	15416.	6.5×10^{-7}	5.4×10^{-6}
30	L	1.6×10^{-6}	5.2	--	5.6×10^{-7}	4.0×10^{-8}
	R	8.7×10^{-6}	5.8	--	7.9×10^{-7}	8.6×10^{-6}

Table II (continued)

n	<u>Matrix</u>	<u>Choleski</u>				
		<u>A</u>	<u>A²</u>	<u>A³</u>	<u>B</u>	<u>D</u>
10	L	1.9×10^{-6}	1.6×10^{-3}	3.0×10^{-1}	1.2×10^{-7}	7.9×10^{-7}
	R	1.2×10^{-6}	1.5×10^{-3}	6.1×10^{-1}	3.1×10^{-7}	9.2×10^{-7}
15	L	6.2×10^{-6}	4.0×10^{-2}	2.9	2.3×10^{-7}	9.8×10^{-7}
	R	5.2×10^{-6}	4.4×10^{-2}	107.5	4.9×10^{-7}	2.8×10^{-6}
20	L	8.5×10^{-6}	3.4×10^{-1}	249.	3.4×10^{-7}	2.3×10^{-6}
	R	1.4×10^{-5}	3.4×10^{-1}	25966.	7.0×10^{-7}	5.9×10^{-6}
25	L	1.5×10^{-5}	1.7	--	4.8×10^{-7}	6.0×10^{-6}
	R	2.9×10^{-5}	1.7	--	9.1×10^{-7}	1.1×10^{-5}
30	L	3.5×10^{-5}	6.5	--	6.7×10^{-7}	7.6×10^{-6}
	R	5.9×10^{-5}	7.6	--	1.2×10^{-6}	1.6×10^{-5}

Table II (continued)

n	<u>Matrix</u>	<u>A</u>	<u>Rank Annihilation</u>		<u>B</u>	<u>D</u>
			<u>A²</u>	<u>A³</u>		
10	L	--	1.3×10^{-3}	3.8	0.0	--
	R	--	1.3×10^{-2}	15.1	9.1×10^{-8}	--
15	L	--	3.6×10^{-2}	202.	0.0	--
	R	--	2.4×10^{-1}	2729.	0.0	--
20	L	--	3.6×10^{-1}	--	0.0	--
	R	--	1.4	--	3.8×10^{-7}	--
25	L	--	1.5	--	0.0	--
	R	--	5.1	--	5.0×10^{-7}	--
30	L	--	4.2	--	0.0	--
	R	--	19.6	--	7.1×10^{-7}	--

Hence it is possible to get an estimate at least of an upper bound of the term on the left of the inequality without knowing the exact inverse of a matrix.

Conclusions

From the foregoing calculations it appears that any one of the matrix inversion methods tried will invert a well-conditioned least squares matrix of order 10 to 30. When the matrix becomes ill-conditioned then the Gauss-Jordan method appears to be clearly superior (at least for the matrices studied). However, if memory capacity must be considered and only a triangular array can be tolerated, then the Choleski and congruent transformation methods appear to offer an advantage for large n . Only in the case of the relatively poorly conditioned A^3 matrix did the Choleski method appear to be the better of these two for evaluating the inverse matrix. From a study involving the inversion of several well-conditioned least squares matrices of orders to $n = 29$ all the methods tried gave α and f values about two to three orders of magnitude smaller than those obtained with the A^3 matrix and consistent with those obtained with the A matrix.

Acknowledgement

The authors wish to express their appreciation to Drs. H. A. Levy and A. S. Householder for interesting and helpful discussions concerning this work.

References

- (1) H. Margenau and G. M. Murphy, "The Mathematics of Physics and Chemistry", Chap. 13, D. Van Nostrand Co., Inc., New York (1943);
M. P. Lietzke, "Polfit II, an IBM 7090 Program for Polynomial Least Squares Fitting", ORNL-3132, April 1961.
- (2) Anthony Ralston and Herbert S. Wilf, "Mathematical Methods for Digital Computers", John Wiley and Sons, Inc. (1960), p. 39.
- (3) "Contributions to the Solution of Systems of Linear Equations and the Determination of Eigenvalues", U. S. Department of Commerce, National Bureau of Standards Applied Mathematics Series 39 (1954), p. 31.
- (4) William R. Busing and Henri A. Levy, "A Procedure for Inverting Large Symmetric Matrices", Comm. ACM 5, 445 (1962).
- (5) Anthony Ralston and Herbert S. Wilf, loc. cit., p. 73.
- (6) Morris Newman and John Todd, "The Evaluation of Matrix Inversion Programs", J. Soc. Indust. Appl. Math 6, 466 (1958).
- (7) Alston S. Householder, "The Theory of Matrices in Numerical Analysis", Ginn and Co., in press, Chapter 4.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX I
Algorithms for the Inversion Methods*

* These algorithms are written in the current 7090 Algol hardware representation. Private communication, Share Algol Project.

```

COMMENT MATRIX INVERSION BY GAUSS-JORDAN ELIMINATION$
PROCEDURE INVERT(N,A,EPSILON,ALARM,DELTA)$
  VALUE N$ ARRAY A$ INTEGER N$
  REAL EPSILON,DELTA$ BOOLEAN ALARM$
  BEGIN ARRAY B,C(/1..N/)$ INTEGER I,J,K,L,P$
  INTEGER ARRAY Z(/1..N/)$ REAL W,Y$
  DELTA#1.0$ALARM# FALSE $
  FOR J#1 STEP 1 UNTIL N DO
    Z(/J/)#J$
    FOR I#1 STEP 1 UNTIL N DO
      BEGIN K#I$Y#A(/I,I/)$L#I-1$P#I+1$
      FOR J#P STEP 1 UNTIL N DO
        BEGIN W#A(/I,J/)$ IF ABS(W) GR ABS(Y) THEN
          BEGIN K#J$Y#W$
          END
        END $
      DELTA#DELTA*Y$
      IF ABS(Y) LS EPSILON THEN
        BEGIN ALARM# TRUE $ GOTO RETURN END $
        FOR J#1 STEP 1 UNTIL N DO
          BEGIN C(/J/)#A(/J,K/)$A(/J,K/)$A(/J,I/)$
            A(/J,I/)#-C(/J/)/Y$B(/J/)$A(/I,J/)$A(/I,J/)/Y
          END $
          A(/I,I/)#I/Y$J#Z(/I/)$Z(/I/)$Z(/K/)$Z(/K/)$J$
          FOR K#1 STEP 1 UNTIL L,P STEP 1 UNTIL N DO
            FOR J#1 STEP 1 UNTIL L,P STEP 1 UNTIL N DO
              A(/K,J/)$A(/K,J/)-B(/J/)*C(/K/)
            END $
          FOR I#1 STEP 1 UNTIL N DO
            BEGIN
              REPEAT..K#Z(/I/)$ IF K EQ I THEN GOTO ADVANCES$
              FOR J#1 STEP 1 UNTIL N DO
                BEGIN W#A(/I,J/)$A(/I,J/)$A(/K,J/)$A(/K,J/)$W END $
                P#Z(/I/)$Z(/I/)$Z(/K/)$Z(/K/)$P$DELTA#-DELTA$
                GOTO REPEAT$ADVANCE.. END $RETURN.. END $
            END
          END
        END
      END
    END
  END

```

```

COMMENT CONGRUENT TRANSFORMATION INVERSION ROUTINE$
PROCEDURE SYMINV(A,N,OK)$
  VALUE N$
  INTEGER N$
  BOOLEAN OK$
  REAL ARRAY A$
  BEGIN
    REAL AIJ,AJL,ALJ,AII,AIL$
    COMMENT.. OBTAIN THE INVERSE OF SYMMETRIC MATRIX A
              OF ORDER N. ONLY THE UPPER TRIANGULAR
              PART OF A IS STORED. ON EXIT THE INVERSE
              REPLACES A IN STORAGE. PROCEDURE
              BEGINS AT DIAGONALIZE$

    REAL BIGAJJ$
    INTEGER I,J,K$
    REAL ARRAY Q(/1..N,1..N/)$
    INTEGER ARRAY L(/1..N/)$
    PROCEDURE INTERCHANGE$
      BEGIN
        COMMENT.. INTERCHANGE ROW-COLUMN I WITH
                  ROW-COLUMN L(/I/). PROCEDURE BEGINS
                  AT INTERCHANGE$
        PROCEDURE EXCHANGE(X,Y)$ REAL X,Y$
          BEGIN
            COMMENT.. EXCHANGE CONTENTS OF
                      LOCATIONS X AND Y$
            REAL HOLD$
            HOLD # X$
            X # Y$
            Y # HOLD
          END EXCHANGE$
        IF L (/I/) NQ I THEN
          BEGIN
            FOR J # I STEP 1 UNTIL N DO
              IF J LS L (/I/) THEN
                BEGIN
                  AIJ#A(/I,J/)$
                  AJL#A(/J,L(/I/)/)$
                  EXCHANGE(AIJ,AJL)$
                  A(/I,J/)#AIJ$
                  A(/J,L(/I/)/)#AJL
                END
              END
            END
          END
        END INTERCHANGE..
      END
    END
  END

```

```

ELSE
  BEGIN
    AIJ#A(/I,J/)$
    ALJ#A(/L(/I/),J/)$
    EXCHANGE(AIJ,ALJ)$
    A(/I,J/)#AIJ$
    A(/L(/I/),J/)#ALJ$
    END $
    AII#A(/I,I/)$
    AIL#A(/I,L(/I/)) $
    EXCHANGE(AII,AIL)$
    A(/I,I/)#AII$
    A(/I,L(/I/)) #AIL
  END
END INTERCHANGES$
DIAGONALIZE.. FOR I # 1 STEP 1 UNTIL N DO
  BEGIN
    L(/I/) # 1$
    BIGAJJ # 0$
    FOR J # 1 STEP 1 UNTIL N DO
      BEGIN
        IF ABS(A(/J,J/)) GR BIGAJJ THEN
          BEGIN
            BIGAJJ # ABS(A(/J,J/))$
            L(/I/) # J
          END
        END$
      END$
    INTERCHANGES$
    IF A (/I,I/) EQ 0 THEN
      BEGIN
        OK # FALSE $
        COTO ENDI
      END $
    A(/I,I/) # 1/A(/I,I/) $
    FOR K # I+1 STEP 1 UNTIL N DO
      Q(/I,K/) # -A(/I,I/)*A(/I,K/)$
      FOR J # I+1 STEP 1 UNTIL N DO
        FOR K # J STEP 1 UNTIL N DO
          A(/J,K/) # A(/J,K/)+Q(/I,K/)*A(/I,J/)
        END $
      END $
    END $
  END $

```

```

RESTORE..      FOR I # N-1 STEP -1 UNTIL 1 DO
                BEGIN
                  FOR J # I+1 STEP 1 UNTIL N DO
                    A(/I,J/) # 0$
                  FOR J # I+1 STEP 1 UNTIL N DO
                    BEGIN
                      FOR K # J STEP 1 UNTIL N DO
                        BEGIN
                          A(/I,K/) # A(/I,K/)+Q(/I,J/)*A(/J,K/)$
                          IF J NQ K THEN
                            A(/I,J/) # A(/I,J/)+Q(/I,K/)*A(/J,K/)
                        END$
                      A(/I,I/) # A(/I,I/)+Q(/I,J/)*A(/I,J/)
                    END $
                  INTERCHANGE$
                END $
              OK # TRUE$
            ENDI..  END SYMINV$

```



```

COMMENT ROUTINE TO INVERT SYMMETRIC POSITIVE DEFINITE MATRICES BY
THE CHOLESKY METHOD$
PROCEDURE MATINV(A,AINV,ORDER,SINGULAR MATRIX,NEGATIVE MATRIX)$
  VALUE ORDER$ INTEGER ORDER$ ARRAY A,AINV$
  BOOLEAN SINGULAR MATRIX,NEGATIVE MATRIX$
  BEGIN ARRAY T,K(/1..ORDER,1..ORDER/)$
  REAL SUML$
  NEGATIVE MATRIX#SINGULAR MATRIX# FALSE $
  FOR L#1 STEP 1 UNTIL ORDER DO
    FOR M#L STEP 1 UNTIL ORDER DO
      BEGIN IF L EQ M THEN
        BEGIN SUML#0.0$
          FOR I#1 STEP 1 UNTIL L-1 DO
            SUML#T(/I,L/)**2+SUML$
          IF (A(/L,M/)-SUML) EQ 0 THEN BEGIN SINGULAR MATRIX# TRUE $
            GOTO RETURN END
          ELSE BEGIN IF (A(/L,M/)-SUML) LS 0.0 THEN BEGIN NEGATIVE
            MATRIX#TRUE$ GOTO RETURN END END $
            T(/L,M/)#SQRT(A(/L,M/)-SUML)
          END
          ELSE
            BEGIN SUML#0.0$
              FOR I#1 STEP 1 UNTIL L-1 DO
                SUML#T(/I,L/)*T(/I,M/)+SUML$
                T(/L,M/)#(A(/L,M/)-SUML)/T(/L,L/)
              END
            END $
          FOR L#1 STEP 1 UNTIL ORDER DO
            K(/L,L/)#1.0/T(/L,L/)$
          FOR L#1 STEP 1 UNTIL ORDER-1 DO
            FOR M#L+1 STEP 1 UNTIL ORDER DO
              BEGIN SUML#0.0$
                FOR I#L STEP 1 UNTIL M-1 DO
                  SUML#T(/I,M/)*K(/L,I/)+SUML$
                  K(/L,M/)#-SUML/T(/M,M/)
                END$
              FOR L#1 STEP 1 UNTIL ORDER DO
                FOR M#1 STEP 1 UNTIL ORDER DO
                  BEGIN SUML#0.0 $
                    FOR I#M STEP 1 UNTIL ORDER DO
                      SUML#K(/L,I/)*K(/M,I/)+SUML$
                      AINV(/L,M/)#SUML
                    END $
                  RETURN.. END $

```

```

COMMENT MATRIX INVERSION BY RANK ANNIHILATIONS$
PROCEDURE RANKAN(A,C,N)$
  INTEGER N$ VALUE N$ ARRAY A,C$
  BEGIN ARRAY ALPHA,BETA,U,V(/1..N/)$
  INTEGER I,J,K,II,JI,NI$ REAL SIG,SIGP,LAMBDA$
  BEGIN FOR I#1 STEP 1 UNTIL N DO
    FOR J#1 STEP 1 UNTIL N DO
      C(/I,J/)# IF I EQ J THEN 1.0 ELSE 0.0$
    FOR I#1 STEP 1 UNTIL N DO
      A(/I,I/)#A(/I,I/)-1.0$
    NI#1$
  50..I#NI$
    FOR K#1 STEP 1 UNTIL N DO
      BEGIN ALPHA(/K/)#A(/K,NI/)/A(/NI,NI/)$
      BETA(/K/)#A(/NI,K/)
    END $
    FOR II#1 STEP 1 UNTIL N DO
      BEGIN SIG#0.0$SIGP#0.0$
      FOR K#NI STEP 1 UNTIL N DO
        BEGIN SIG#SIG+C(/II,K/)*A(/K,NI/)$
        SIGP#SIGP+A(/NI,K/)*C(/K,II/)
      END $
      U(/II/)#SIG$V(/II/)#SIGP
    END $
    J#NI$SIG#0.0$
    FOR JI#J STEP 1 UNTIL N DO
      SIG#SIG+V(/JI/)*A(/JI,NI/)$
    LAMBDA#SIG+A(/NI,NI/)$
    FOR I#1 STEP 1 UNTIL N DO
      FOR JI#1 STEP 1 UNTIL N DO
        C(/I,J/)#C(/I,J/)-U(/I/)*V(/J/)/LAMBDA
      IF NI EQ N THEN GOTO RETURN$
    I#NI$
    FOR II#1 STEP 1 UNTIL N DO
      BEGIN J#NI$
      FOR K#J STEP 1 UNTIL N DO
        A(/II,K/)#A(/II,K/)-ALPHA(/II/)*BETA(/K/)
      END $
      NI#NI+1$ GOTO 50$
  RETURN.. END RANKAN$

```

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX II
FORTRAN Subroutines for the Inversion Methods

```

C      MATRIX INVERSION BY GAUSS-JORDAN ELIMINATION
      SUBROUTINE INVERT(A,N)
      DIMENSION A(30,30),B(30),C(30),LZ(30)
      DO 10 J#1,N
10     LZ(J)#J
      DO 20 I#1,N
      K#I
      Y#A(I,I)
      L#I-1
      LP#I+1
      IF(N-LP)14,11,11
11     DO 13 J#LP,N
      W#A(I,J)
      IF(ABSF(W)-ABSF(Y))13,13,12
12     K#J
      Y#W
13     CONTINUE
14     DO 15 J#1,N
      C(J)#A(J,K)
      A(J,K)#A(J,I)
      A(J,I)#-C(J)/Y
      A(I,J)#A(I,J)/Y
15     B(J)#A(I,J)
      A(I;I)#1.0/Y
      J#LZ(I)
      LZ(I)#LZ(K)
      LZ(K)#J
      DO 19 K#1,N
      IF(I-K)16,19,16
16     DO 18 J#1,N
      IF(I-J)17,18,17
17     A(K,J)#A(K,J)-B(J)*C(K)
18     CONTINUE
19     CONTINUE
20     CONTINUE
      DO 200 I#1,N
      IF(I-LZ(I))100,200,100
100    K#I+1
      DO 500 J#K,N
      IF(I-LZ(J))500,600,500
600    M#LZ(I)
      LZ(I)#LZ(J)
      LZ(J)#M
      DO 700 L#1,N
      C(L)#A(I,L)
      A(I,L)#A(J,L)
700    A(J,L)#C(L)
500    CONTINUE
200    CONTINUE
      RETURN
      END

```

```

C      MATRIX INVERSION BY THE CONGRUENT TRANSFORMATION METHOD
      SUBROUTINE BLMATI(A,AI,N,NERR)
      DIMENSION L(30),A(30,30),Q(30,30),AI(30,30)
      DO 100 I#1,N
        L(I)#I
        BIGA#0.0
        DO 120 J#I,N
          IF(ABSF(A(J,J))-BIGA)120,120,130
130    BIGA#ABSF(A(J,J))
          L(I)#J
120    CONTINUE
          IF(L(I)-I)140,140,150
150    CALL INTCGE(L,I,A,N)
140    IF(A(I,I))160,170,160
170    NERR#1
        GO TO 1000
160    A(I,I)#1./A(I,I)
        II#I+1
        IF(II-N)400,400,100
400    DO 180 K#II,N
180    Q(I,K)#-A(I,I)*A(I,K)
        DO 101 J#II,N
        DO 101 K#J,N
101    A(J,K)#A(J,K)+Q(I,K)*A(I,J)
100    CONTINUE
        I#N-1
300    II#I+1
        DO 200 J#II,N
200    A(I,J)#0.0
        DO 210 J#II,N
        DO 220 K#J,N
          A(I,K)#A(I,K)+Q(I,J)*A(J,K)
          IF(K-J)220,220,230
230    A(I,J)#A(I,J)+Q(I,K)*A(J,K)
220    CONTINUE
210    A(I,I)#A(I,I)+Q(I,J)*A(I,J)
          IF(L(I)-I)250,250,240
240    CALL INTCGE(L,I,A,N)
250    IF(I-1)260,260,270
270    I#I-1
        GO TO 300
260    NERR#0
        DO 310 I#1,N
        DO 310 J#I,N
310    A(J,I)#A(I,J)
        DO 320 I#1,N
        DO 320 J#I,N
320    AI(I,J)#A(I,J)
1000    CONTINUE
      RETURN
      END

```

```
SUBROUTINE INTCGE(L,I,A,N)
DIMENSION L(30),A(30,30)
DO 402 J#I,N
IF(J-L(I))400,401,401
400 X#A(I,J)
M#L(I)
Y#A(J,M)
CALL EXCGE(X,Y)
A(I,J)#X
A(J,M)#Y
GO TO 402
401 X#A(I,J)
M#L(I)
Y#A(M,J)
CALL EXCGE(X,Y)
A(I,J)#X
A(M,J)#Y
402 CONTINUE
X#A(I,I)
Y#A(I,M)
CALL EXCGE(X,Y)
A(I,I)#X
403 A(I,M)#Y
RETURN
END
SUBROUTINE EXCGE(X,Y)
HOLD#X
X#Y
Y#HOLD
RETURN
END
```

```
C      MATRIX INVERSION BY THE CHOLESKY METHOD
      SUBROUTINE MATINV(A,B,N)
      DIMENSION A(30,30),B(30,30),T(30,30),D(30,30)
      DO 20 L#1,N
      DO 20 M#L,N
      IF(L-M)15,10,15
10     SUML#0.0
      LU#L-1
      IF(LU)12,50,12
12     DO 11 I#1,LU
11     SUML#T(I,L)**2+SUML
50     T(L,M)#SQRTF(A(L,M)-SUML)
      GO TO 20
15     SUML#0.0
      LU#L-1
      IF(LU)13,60,13
13     DO 16 I#1,LU
16     SUML#T(I,L)*T(I,M)+SUML
60     T(L,M)#(A(L,M)-SUML)/T(L,L)
20     CONTINUE
      DO 25 L#1,N
25     D(L,L)#1.0/T(L,L)
      NI#N-1
      DO 30 L#1,NI
      LI#L+1
      DO 30 M#LI,N
      SUML#0.0
      MI#M-1
      DO 26 I#L,MI
26     SUML#T(I,M)*D(L,I)+SUML
      D(L,M)#-SUML/T(M,M)
30     CONTINUE
      DO 40 L#1,N
      DO 40 M#L,N
      SUML#0.0
      DO 31 I#M,N
31     SUML#D(L,I)*D(M,I)+SUML
      B(L,M)#SUML
40     CONTINUE
      RETURN
      END
```



```

C      MATRIX INVERSION BY RANK ANNIHILATION
      SUBROUTINE RANKAN(A,C,N)
      DIMENSION A(30,30),C(30,30),ALP(30),BET(30),U(30),V(30)
      DO 100 I#1,N
      DO 100 J#1,N
100    C(I,J)#0.0
      DO2I#1,N
      C(I,I)#1.0
      2  A(I,I)#A(I,I)-1.0
      NI#1
      50 I#NI
      DO 6 K#I,N
      ALP(K)#A(K,NI)/A(NI,NI)
      6  BET(K)#A(NI,K)
      DO 11 II#1,N
      SIG#0.0
      SIGP#0.0
      DO 12 K#NI,N
      SIG#SIG+C(II,K)*A(K,NI)
      12 SIGP#SIGP+A(NI,K)*C(K,II)
      U(II)#SIG
      11 V(II)#SIGP
      J#NI
      SIG#0.0
      DO20JI#J,N
      20 SIG#SIG+V(JI)*A(JI,NI)
      ELAM#SIG+A(NI,NI)
      DO 23 I#1,N
      DO24J#1,N
      24 C(I,J)#C(I,J)-U(I)*V(J)/ELAM
      23 CONTINUE
      IF(NI-N)33,32,32
      33 I#NI
      DO 36 II#I,N
      J#NI
      DO 37 K#J,N
      37 A(II,K)#A(II,K)-ALP(II)*BET(K)
      36 CONTINUE
      NI#NI+1
      GO TO 50
      32 RETURN
      END

```

ORNL-3430
UC-32 - Mathematics and Computers
TID-4500 (19th ed., Rev.)

INTERNAL DISTRIBUTION

1. Biology Library	46. R. L. Hahn
2. Reactor Division Library	47. J. Halperin
3-4. Central Research Library	48. H. W. Joy
5-6. ORNL - Y-12 Technical Library,
Document Reference Section	49. M. A. Kastenbaum
7-26. Laboratory Records Department	50. C. E. Larson
27. Laboratory Records, ORNL R.C.	51-70. M. H. Lietzke
28. Nancy Alexander	71-75. M. P. Lietzke
29. E. D. Arnold	76. F. Miller
30. J. O. Betterton	77. G. D. O'Kelley
31. R. E. Biggers	78. R. M. Rush
32. G. E. Boyd	79-80. O. W. Russ
33. H. P. Carter	81. C. S. Shoup, Jr.
34. E. L. Compere	82. M. J. Skinner
35. D. A. Costanzo	83-102. R. W. Stoughton
36. W. Davis, Jr.	103. J. A. Swartout
37. A. C. Downing	104. C. D. Susano
38. H. W. Dunn	105. E. H. Taylor
39. R. G. Edwards (K-25)	106. A. M. Weinberg
40. R. E. Funderlic (K-25)	107. P. B. Wood (K-25)
41. D. A. Gardiner	108. J. H. Zeigler (K-25)
42. L. O. Gilpatrick	109. T. H. Davies (consultant)
43. C. D. Goodman	110. D. F. Hornig (consultant)
44. D. G. Gosslee	111. I. Perlman (consultant)
45. A. A. Grau	112. J. D. Roberts (consultant)

EXTERNAL DISTRIBUTION

113. Research and Development Division, AEC, ORO
114-648. Given distribution as shown in TID-4500 (19th ed., Rev.) under
Mathematics and Computers category