General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

GE AUTHORIZATION
Control No. ______
Date: ______

378

MSC-IN-66-ED-41

MSC INTERNAL NOTE

EFFICIENCY OF GENERALIZED MATRIX INVERSION METHODS

By

Fred C. Delaney

Gary G. Gaffney

and

Fred M. Speed

AUG 1970
RECEIVED
NASA STI FACILITY
INPUT BRANCH

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

September 1, 1966

N70-35743
(ACCESSION NUMBER)
38
(PAGES)
TMX-64453
(NASA CR OR TMX OR AD NUMBER)
1
(THRU)
(CODE)
19
(CATEGORY)
FACILITY FORM 602

MSC INTERNAL NOTE NO. 66-ED-41

EFFICIENCY OF GENERALIZED MATRIX INVERSION METHODS

Prepared by: Fred C. Delaney
Fred C. Delaney
Lockheed Electronics Company

Prepared by: Gary G. Gaffney
Gary G. Gaffney
Theory and Analysis Office
Aerospace Writing and Analysis Program

Prepared by: Fred M. Speed
Fred M. Speed
Theory and Analysis Office

Approved by: Eugene L. Davis
Eugene L. Davis, Jr.
Chief, Theory and Analysis Office

Approved by: Eugene H. Brock
Eugene H. Brock
Chief, Computation and Analysis Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

September 1, 1966

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

EFFICIENCY OF GENERALIZED MATRIX INVERSION METHODS

By Fred C. Delaney, Gary G. Gaffney, and Fred M. Speed

SUMMARY

Generalized matrix inversion has become an increasingly important concept in matrix theory as well as a useful tool in engineering, statistics, control theory, and space mission design. For this reason, the need arises for an efficient (i.e., fast and accurate) method of computing the generalized inverse of an arbitrary n x m matrix. The purpose of this paper is to present the results of computer tests used to compare the relative efficiency of several computer programs designed to calculate the generalized inverse of an arbitrary real matrix.

INTRODUCTION

The concept of matrix inversion was first generalized by E. H. Moore (ref. 1) in 1920. In the 1950's R. Penrose (ref. 2) and A. Bjerhammer (ref. 3), working independently, formulated equivalent definitions of the generalized inverse of an arbitrary complex matrix. The most common definition, given by Penrose (ref. 2) is a consequence of the following theorem.

THEOREM 1: For any real n x m matrix A, there is a unique real m x n matrix A^+ (the generalized inverse of A) such that:

(1) $AA^+A = A$

(2) $A^+AA^+ = A^+$

(3) $(AA^+)^T = AA^+$

(4) $(A^+A)^T = A^+A$

The use of the generalized inverse in engineering problems, statistics, and control theory gave rise, naturally enough, to the development of several different computational methods. Some of these methods were developed and programed by researchers at NASA-MSC to solve problems requiring generalized matrix inversion.

This paper presents the results of an examination of the various programs for overall efficiency, comparing them in terms of accuracy and speed.

SYMBOLS

Capital letters denote matrices with real entries.

a_n is a row vector.

p_n is a column vector.

A^T denotes the transpose of the matrix A.

A^+ denotes the generalized inverse of the matrix A.

$(A)_{ij}$ is the entry in the i^{th} row and j^{th} column of the matrix A.

$|| \quad ||$ is a matrix norm.

Trace (A) is the trace of the matrix A.

I is the identity matrix.

I_k is the k x k identity matrix.

$(A \vdots B)$ is a matrix partitioned into the matrix A and the matrix B.

METHODS STUDIED

Although there are numerous mathematical methods for calculating the generalized inverse of a matrix, the purpose of this study was to determine the most efficient (that is, fastest and most accurate) computer method for calculating this inverse for arbitrary real matrices. A preliminary survey of the existing algorithms for generalized matrix inversion showed that some of them were not readily adaptable to computer programing or were more suitable only to theoretical investigations and required no further consideration. One of the Penrose methods (ref. 4) was discarded because it first requires a type of matrix partitioning that is time consuming on the computer. The Ben-Israel and Wersan method (ref. 5) was eliminated because it depends on the exact determination of rank, which depends on round-off and approximation errors. The Householder method (ref. 6) was rejected because it depends on predetermining the rank of the matrix. Since the Ben-Israel and Charnes method (ref. 7) uses the Lagrange-Sylvester interpolation polynomial, which is sensitive to error in the computer, it, too, was discarded. Finally, the Decell method based on the Cayley-Hamilton theorem (ref. 8), which requires the calculation of powers of a matrix, was eliminated because of the error which such a calculation causes.

The following five methods, having satisfied this preliminary requirement of computer adaptability, were then examined because they showed promise of being efficient generalized inverse programs:

(1) the computer program PEN2, based on another of the Penrose (ref. 2) methods,

(2) the program SEQINV, based on a method by H. P. Decell, Jr., (ref. 9) of NASA-MSC,

(3) JPLUS, a method developed and programed by two of the authors, F. M. Speed of NASA-MSC, and F. C. Delaney of LEC,

(4) APLUS, taken from an iterative method devised by H. P. Decell, Jr., and S. W. Kahng (ref. 10),

(5) GINV2, an algorithm developed by B. Rust, W. R. Burras, and C. Schneeberger (ref. 11).

(For the mathematics underlying these methods see Appendix A.)

These methods were programed in FORTRAN IV, if they had not already been, and were then tested very extensively on the UNIVAC 1108 to determine efficiency.

Since generalized matrix inversion is applicable to arbitrary matrices, some preliminary mention should be made of the variety of matrices used in testing the programs. The generalized inverses of singular and nonsingular square matrices and of nonsquare matrices of full and less than full rank were computed by each of the five methods. These matrices were, for the most part, randomly generated and differed in size from order 2 x 2 to order 45 x 40. As the results will indicate, the type and size of the matrix, whether due to round-off error in the computer, or to the increased computer time required by larger matrices, or to peculiarities in the method of generalized inversion, often had significant effects on the speed and accuracy of the program.

Accuracy Determinations - Methods and Results

Before any test results can be presented, a description of the methods for determining and comparing the accuracies of the above five programs is necessary.

The four identities of THEOREM 1, which define the generalized inverse, suggest a means for testing the accuracy of a program designed to calculate it. In the case of real matrices, norms based on these identities can be defined in the following way:

Let A be an n x m matrix with real entries, and let $\hat{A}$ denote the generalized inverse as calculated by computer. Then $\hat{A}$ is an m x n matrix also with real entries.
Define:

$$\text{NORM 1} = ||A\hat{A}A - A|| = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\left[(A\hat{A}A)_{ij} - (A)_{ij}\right]^2}{nm}}$$

$$\text{NORM 2} = ||\hat{A}A\hat{A} - \hat{A}|| = \sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left[(\hat{A}A\hat{A})_{ij} - (\hat{A})_{ij}\right]^2}{nm}}$$

$$\text{NORM 3} = \|(A\hat{A})^T - A\hat{A}\| = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left[((A\hat{A})^T)_{ij} - (A\hat{A})_{ij}\right]^2}{nm}}$$

$$\text{NORM 4} = \|(\hat{A}A)^T - \hat{A}A\| = \sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{m}\left[((\hat{A}A)^T)_{ij} - (\hat{A}A)_{ij}\right]^2}{nm}}$$

These norms provide a satisfactory test for accuracy since, first, each is, in fact, the root mean square of an element in the difference matrix for that norm and, second, each norm is equal to zero if and only if $\hat{A}$ is equal to A^+, the true generalized inverse of A. Note, however, that these norms will not, as a rule, be zero due to round-off error in the computer.

In order for a generalized inversion program to be considered dependable and to have wide application, it must be capable of computing the generalized inverse of all types of matrices with a consistent and predictable accuracy. The results of the tests in this study showed that not all of the five programs mentioned above could meet this demand.

Two programs, APLUS and GINV2, did, however, perform with more than satisfactory accuracy for all of the matrix types used in testing. The norms evaluated using the generalized inverses computed by these two programs ranged nearly always between 1×10^{-4} and 1×10^{-12}, averaging between 1×10^{-7} and 1×10^{-9}. (For a more detailed comparison see Table I.)

In order to determine if there was a significant difference in accuracy between APLUS and GINV2, an analysis of variance was performed (see Appendix B for the details of the analysis). The results of the analysis showed that, for full rank matrices, it is not possible to reject the hypothesis that the accuracies of the two methods are equal. However, for matrices of nonfull rank (excluding n x m matrices of rank 1), the hypothesis that the accuracies of the two methods were equal was rejected in favor of the hypothesis that GINV2 was more accurate than APLUS.

The remaining three methods, PEN2, SEQINV, and JPLUS, could not consistently meet the demands on accuracy, and therefore will certainly have restrictions--in varying degrees--on their application.

PEN2 gives acceptable norms for some small matrices and for matrices of very low rank but gives very poor results for all other types of matrices. This program is not very dependable and should find little, if any, application.

SEQINV yields good results for nonsingular matrices and matrices with full rank or low rank, but as the order of the matrix increases past 30 x 30 it begins to fail noticeably for singular matrices and matrices with less than full rank. Even when SEQINV performs well, its accuracy does not exceed--and usually lags behind--that of APLUS and GINV2. For completeness, it should be noted here that SEQINV contains a zero-test whose epsilon value, when increased slightly, causes significant improvement in some norms which had previously indicated that the program had failed. This epsilon value was not experimented with in detail since varying it caused no significant change in those norms for which SEQINV formerly gave good results.

Of the above three methods, JPLUS is, by far, the most consistently accurate and dependable. It yields poor norms in only a small number of cases; namely where the matrices were singular and of large rank and order. But despite its rather satisfactory performance, its accuracy is not as great as that of APLUS or GINV2, which limits its application. JPLUS, like SEQINV, also has a zero-test, whose epsilon value, when varied, causes changes in the norms with results very similar to those observed for SEQINV.

Results of Speed Determination

In order to obtain a sample of the relative speed of each program, a test block consisting of one hundred 10 x 10 nonsingular matrices was generated randomly. The computer time required by each program (except PEN2) for calculating the generalized inverse of each matrix in the block was then determined for comparison purposes. (See Table II.) Because the levels of accuracy for PEN2, SEQINV, and JPLUS were not entirely satisfactory, no further time tests were made on these programs.

Since APLUS and GINV2 were the only programs which met the demands on accuracy, and since they were found to have nearly equal accuracy, time was the deciding factor in determining which was the most efficient computer program examined. For this reason, time samplings were run for blocks of 20 x 20, 30 x 30, and 40 x 40 nonsingular matrices of the type described above. (To conserve computer time, the number of matrices per block was reduced as the matrix order increased.) The results of these samplings showed that GINV2 is considerably faster than APLUS. (See Table III.)

The GINV2 program must do an additional set of operations for any dependent column in a matrix whose generalized inverse is to be computed. For this reason, the times required by APLUS and GINV2 were also compared for singular matrices. Various blocks, each of 15 matrices of the same rank and order, were again generated randonly. The matrix types tested were order 10 x 10 matrices of ranks 1 through 10 and order 20 x 20 matrices of ranks 1 through 20. (See Table IV.)

It was observed that for matrices of rank 1, both of order 10 x 10 and of order 20 x 20, APLUS is slightly faster than GINV2. It seems reasonable to conclude that for the rank 1 case APLUS makes a sufficiently accurate initial guess at the generalized inverse and the iteration process stops immediately. However, for matrices of rank 2, APLUS becomes considerably slower while GINV2 becomes slightly faster so that the times for the two methods compare much as they did in the nonsingular case. As the rank increases to full rank, APLUS becomes generally slower while GINV2 becomes increasingly faster. For a fixed matrix size, GINV2 is at its fastest when the rank is maximum. It was observed that GINV2 averages 10 to 15 times faster than APLUS.

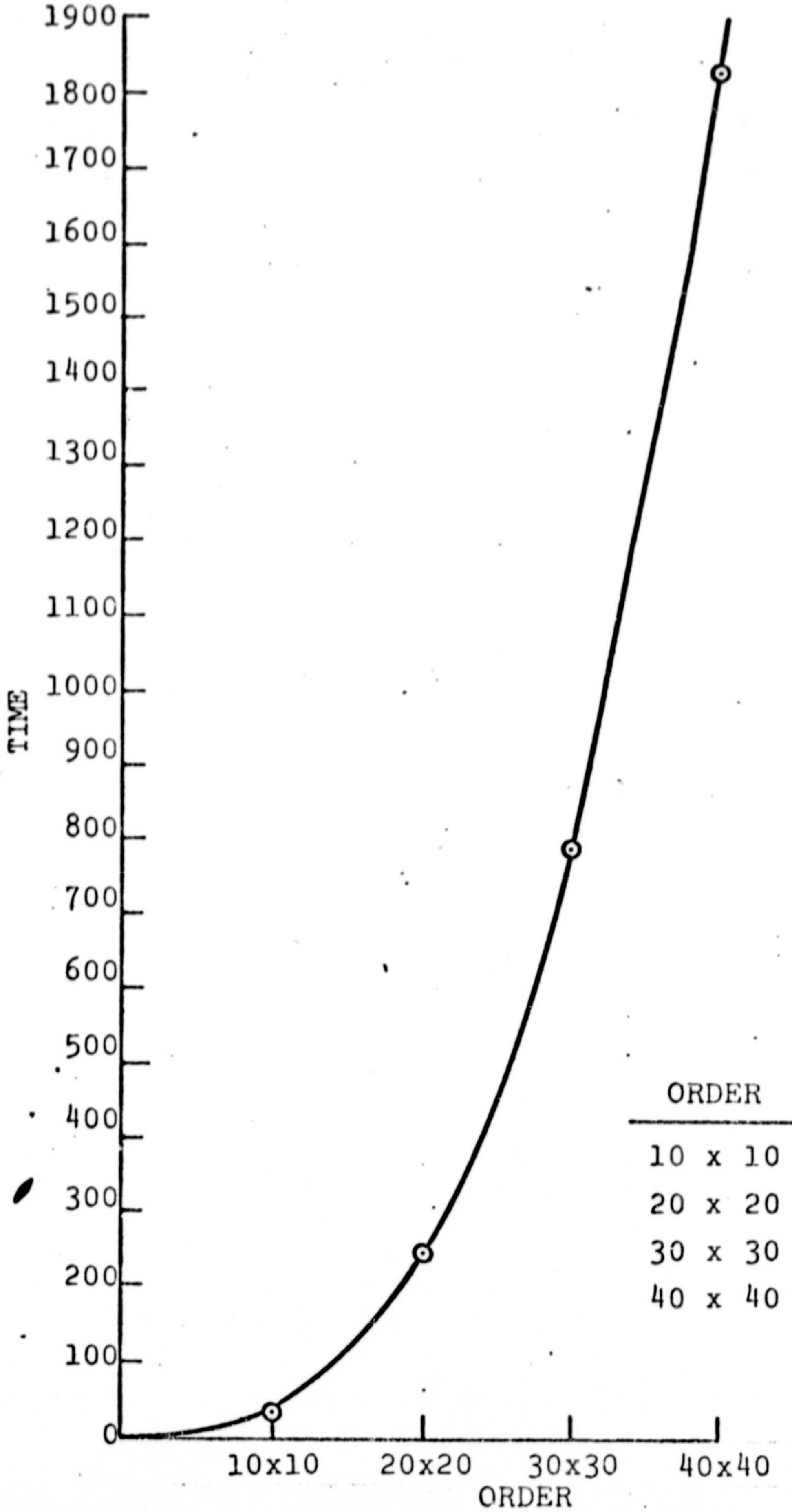
CONCLUSION

The results of this study clearly indicate that GINV2 is the most efficient program (among the computer subroutines studied) for calculating the generalized inverse of a matrix. (See Figure 1.) Both APLUS and GINV2 are dependable methods in terms of accuracy, but GINV2 is considerably faster. APLUS is more efficient than GINV2 in only one special case--matrices (other than m x 1 matrices) of rank 1; and in this case the following simple formula exists for computing the generalized inverse:

$$A^{+} = \frac{1}{\mathrm{trace}(A^{T}A)} A^{T}.$$

Further information as well as copies of the computer programs can be obtained from:

F. M. Speed
Theory and Analysis Office
National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas



ORDER	TIME (MILLI-SECONDS)
10 x 10	34.9
20 x 20	245.8
30 x 30	784.1
40 x 40	1825

FIGURE 1
GRAPH OF TIME VS. ORDER FOR GINV2

Table I

Norm Values For GINV2 and APLUS
For Selected Randomly Generated Matrices
(The norm values for APLUS appear first for each matrix type)

ORDER	RANK	NORM 1	NORM 2	NORM 3	NORM 4
2 x 2	1	2.9×10^{-8}	1.1×10^{-10}	0	0
		1.4×10^{-7}	5.7×10^{-10}	2.6×10^{-9}	2.6×10^{-8}
2 x 2	2	1.6×10^{-6}	8.7×10^{-9}	4.0×10^{-9}	4.0×10^{-9}
		3.8×10^{-7}	1.1×10^{-9}	0	0
2 x 3	1	4.5×10^{-7}	5.5×10^{-11}	0	2.2×10^{-9}
		2.4×10^{-8}	1.2×10^{-11}	0	4.3×10^{-8}
2 x 3	2	4.4×10^{-6}	1.7×10^{-7}	4.3×10^{-9}	1.0×10^{-8}
		1.2×10^{-7}	2.0×10^{-9}	1.1×10^{-9}	9.9×10^{-9}
4 x 5	1	1.6×10^{-6}	1.7×10^{-11}	6.6×10^{-10}	0
		1.5×10^{-6}	3.3×10^{-11}	1.3×10^{-9}	6.9×10^{-8}
4 x 5	2	3.0×10^{-7}	1.1×10^{-8}	1.3×10^{-8}	1.2×10^{-8}
		1.4×10^{-6}	4.2×10^{-10}	6.6×10^{-9}	1.7×10^{-7}
4 x 5	4	2.1×10^{-7}	8.4×10^{-10}	7.3×10^{-9}	1.1×10^{-8}
		2.2×10^{-6}	6.0×10^{-9}	2.4×10^{-7}	1.3×10^{-7}
10 x 8	1	2.2×10^{-6}	2.9×10^{-11}	7.1×10^{-10}	9.2×10^{-10}
		4.6×10^{-6}	8.1×10^{-11}	9.2×10^{-10}	4.8×10^{-7}

Table I (Continued)

ORDER	RANK	NORM 1	NORM 2	NORM 3	NORM 4
10 x 8	2	3.5×10^{-6} 3.9×10^{-6}	5.4×10^{-9} 2.6×10^{-10}	1.4×10^{-8} 7.1×10^{-9}	4.1×10^{-8} 1.2×10^{-7}
10 x 8	5	7.6×10^{-7} 1.7×10^{-6}	1.0×10^{-8} 3.0×10^{-10}	1.7×10^{-8} 6.7×10^{-9}	1.7×10^{-8} 2.1×10^{-7}
10 x 8	8	1.0×10^{-6} 4.2×10^{-7}	2.1×10^{-8} 9.6×10^{-10}	1.4×10^{-8} 1.3×10^{-8}	1.8×10^{-8} 1.5×10^{-8}
8 x 10	1	2.9×10^{-6} 5.7×10^{-6}	2.7×10^{-11} 4.8×10^{-11}	5.4×10^{-10} 1.1×10^{-9}	4.5×10^{-10} 3.8×10^{-7}
8 x 10	2	2.9×10^{-6} 5.8×10^{-6}	3.9×10^{-9} 8.4×10^{-11}	9.5×10^{-9} 4.2×10^{-9}	1.1×10^{-8} 6.9×10^{-7}
8 x 10	5	1.8×10^{-6} 1.4×10^{-6}	3.1×10^{-8} 6.2×10^{-10}	1.4×10^{-8} 7.2×10^{-9}	2.7×10^{-8} 3.5×10^{-7}
8 x 10	8	3.4×10^{-7} 5.5×10^{-7}	1.0×10^{-9} 1.2×10^{-9}	9.4×10^{-9} 2.8×10^{-8}	2.0×10^{-8} 6.2×10^{-8}
10 x 10	1	3.1×10^{-6} 9.4×10^{-6}	2.0×10^{-11} 5.7×10^{-11}	5.4×10^{-10} 1.3×10^{-9}	4.5×10^{-10} 3.8×10^{-7}
10 x 10	2	5.9×10^{-6} 7.3×10^{-6}	8.3×10^{-9} 3.2×10^{-10}	1.2×10^{-8} 1.7×10^{-9}	3.4×10^{-8} 4.5×10^{-7}

Table I (Continued)

ORDER	RANK	NORM 1	NORM 2	NORM 3	NORM 4
10 x 10	5	5.8×10^{-6} 3.1×10^{-6}	2.6×10^{-8} 3.4×10^{-10}	1.4×10^{-8} 5.9×10^{-9}	1.3×10^{-7} 2.9×10^{-7}
10 x 10	10	1.0×10^{-6} 9.0×10^{-7}	1.6×10^{-8} 1.1×10^{-8}	4.1×10^{-8} 5.6×10^{-8}	1.7×10^{-7} 9.6×10^{-8}
20 x 20	1	8.4×10^{-5} 8.7×10^{-5}	1.8×10^{-11} 1.9×10^{-11}	9.0×10^{-10} 8.5×10^{-10}	4.5×10^{-10} 1.6×10^{-6}
20 x 20	2	5.0×10^{-5} 5.4×10^{-5}	7.2×10^{-9} 2.9×10^{-10}	1.3×10^{-8} 7.0×10^{-9}	8.1×10^{-8} 1.7×10^{-6}
20 x 20	5	3.0×10^{-5} 3.5×10^{-5}	6.0×10^{-9} 7.1×10^{-10}	1.9×10^{-8} 6.3×10^{-9}	5.2×10^{-8} 1.7×10^{-6}
20 x 20	10	2.6×10^{-5} 3.3×10^{-5}	3.4×10^{-8} 7.0×10^{-10}	2.8×10^{-8} 9.1×10^{-9}	1.7×10^{-7} 1.6×10^{-6}
20 x 20	15	8.2×10^{-6} 3.2×10^{-6}	1.0×10^{-7} 8.1×10^{-10}	2.2×10^{-8} 1.1×10^{-8}	1.5×10^{-7} 5.2×10^{-7}
20 x 20	20	1.2×10^{-6} 1.1×10^{-6}	1.3×10^{-7} 5.7×10^{-9}	2.7×10^{-8} 5.4×10^{-8}	1.0×10^{-7} 6.6×10^{-8}
30 x 10	1	3.0×10^{-5} 2.9×10^{-5}	2.8×10^{-11} 2.8×10^{-11}	9.9×10^{-10} 1.0×10^{-9}	3.4×10^{-10} 4.1×10^{-7}

Table I (Continued)

ORDER	RANK	NORM 1	NORM 2	NORM 3	NORM 4
30 x 10	2	2.0×10^{-5} 1.8×10^{-5}	2.9×10^{-9} 1.6×10^{-10}	1.6×10^{-9} 2.9×10^{-9}	2.1×10^{-8} 6.2×10^{-7}
30 x 10	5	1.4×10^{-5} 1.5×10^{-5}	3.0×10^{-9} 2.4×10^{-10}	2.0×10^{-8} 2.8×10^{-9}	1.8×10^{-8} 4.1×10^{-7}
30 x 10	10	7.5×10^{-7} 5.4×10^{-7}	2.8×10^{-10} 2.2×10^{-10}	1.4×10^{-8} 5.8×10^{-9}	3.5×10^{-9} 5.0×10^{-9}
10 x 30	1	7.2×10^{-5} 9.8×10^{-5}	4.9×10^{-12} 1.1×10^{-11}	3.5×10^{-10} 1.1×10^{-9}	3.8×10^{-10} 3.4×10^{-6}
10 x 30	2	3.5×10^{-5} 5.0×10^{-5}	2.1×10^{-8} 1.7×10^{-10}	2.4×10^{-8} 4.6×10^{-9}	1.2×10^{-8} 3.8×10^{-6}
10 x 30	5	3.4×10^{-5} 6.0×10^{-5}	4.8×10^{-8} 1.3×10^{-9}	4.2×10^{-8} 1.1×10^{-8}	4.3×10^{-7} 5.6×10^{-6}
10 x 30	10	6.5×10^{-7} 9.1×10^{-7}	2.6×10^{-10} 3.4×10^{-10}	1.9×10^{-9} 1.7×10^{-8}	7.7×10^{-9} 7.1×10^{-7}
30 x 30	1	1.6×10^{-4} 1.6×10^{-4}	1.7×10^{-11} 1.8×10^{-11}	7.2×10^{-10} 5.8×10^{-10}	3.1×10^{-10} 4.2×10^{-6}
30 x 30	2	2.4×10^{-4} 2.6×10^{-4}	1.7×10^{-8} 1.4×10^{-10}	1.9×10^{-8} 2.0×10^{-9}	5.8×10^{-8} 4.3×10^{-6}

Table I (Continued)

ORDER	RANK	NORM 1	NORM 2	NORM 3	NORM 4
30 x 30	5	1.2×10^{-4} 1.2×10^{-4}	9.8×10^{-9} 3.5×10^{-10}	2.5×10^{-8} 9.6×10^{-9}	7.2×10^{-8} 3.7×10^{-6}
30 x 30	10	1.0×10^{-4} 1.1×10^{-4}	2.0×10^{-8} 7.4×10^{-10}	2.9×10^{-8} 1.1×10^{-8}	2.2×10^{-7} 4.8×10^{-6}
30 x 30	15	3.4×10^{-5} 3.8×10^{-5}	8.6×10^{-8} 1.4×10^{-9}	4.2×10^{-8} 1.1×10^{-8}	3.3×10^{-7} 5.3×10^{-6}
30 x 30	20	3.0×10^{-5} 2.9×10^{-5}	1.4×10^{-7} 1.2×10^{-9}	3.5×10^{-8} 1.2×10^{-8}	7.7×10^{-7} 2.8×10^{-6}
30 x 30	30	1.3×10^{-6} 1.4×10^{-6}	3.5×10^{-9} 3.0×10^{-9}	1.7×10^{-8} 4.7×10^{-8}	7.7×10^{-8} 5.0×10^{-8}
35 x 35	1	3.9×10^{-4} 3.8×10^{-4}	1.5×10^{-11} 1.5×10^{-11}	8.7×10^{-10} 4.9×10^{-10}	2.6×10^{-10} 7.9×10^{-6}
35 x 35	2	2.6×10^{-4} 2.7×10^{-4}	7.9×10^{-9} 1.5×10^{-10}	1.2×10^{-8} 6.2×10^{-9}	1.7×10^{-7} 5.2×10^{-6}
35 x 35	5	1.8×10^{-4} 2.0×10^{-4}	1.8×10^{-8} 6.0×10^{-10}	2.4×10^{-8} 9.3×10^{-9}	2.1×10^{-7} 6.7×10^{-6}
35 x 35	10	1.5×10^{-4} 2.6×10^{-4}	3.5×10^{-6} 8.1×10^{-10}	7.5×10^{-8} 7.3×10^{-9}	2.7×10^{-7} 6.0×10^{-6}

Table I (Continued)

ORDER	RANK	NORM 1	NORM 2	NORM 3	NORM 4
35 x 35	15	1.5×10^{-4} 1.6×10^{-4}	6.1×10^{-8} 7.3×10^{-10}	4.5×10^{-8} 1.0×10^{-8}	4.7×10^{-7} 5.1×10^{-6}
35 x 35	20	2.4×10^{-5} 1.1×10^{-5}	1.2×10^{-7} 8.8×10^{-10}	3.3×10^{-8} 1.2×10^{-8}	3.1×10^{-7} 4.0×10^{-6}
35 x 35	30	1.8×10^{-5} 1.3×10^{-5}	2.6×10^{-7} 1.6×10^{-9}	2.6×10^{-8} 2.0×10^{-8}	2.7×10^{-7} 1.5×10^{-6}
35 x 35	35	1.8×10^{-6} 1.8×10^{-6}	3.4×10^{-9} 4.0×10^{-9}	2.3×10^{-8} 6.5×10^{-8}	1.5×10^{-7} 7.0×10^{-8}
40 x 15	1	7.7×10^{-5} 7.8×10^{-5}	2.5×10^{-11} 2.5×10^{-11}	9.0×10^{-10} 6.5×10^{-10}	4.0×10^{-10} 1.2×10^{-6}
40 x 15	2	7.0×10^{-5} 7.1×10^{-5}	6.5×10^{-9} 1.6×10^{-10}	1.8×10^{-8} 3.1×10^{-9}	6.4×10^{-8} 5.1×10^{-7}
40 x 15	5	5.8×10^{-5} 5.6×10^{-5}	3.9×10^{-9} 2.4×10^{-10}	2.3×10^{-8} 4.6×10^{-9}	3.6×10^{-8} 4.3×10^{-7}
40 x 15	10	2.1×10^{-5} 2.0×10^{-5}	5.3×10^{-9} 2.2×10^{-10}	2.0×10^{-8} 4.8×10^{-9}	1.9×10^{-8} 5.1×10^{-7}
40 x 15	15	8.9×10^{-7} 9.0×10^{-7}	2.7×10^{-10} 2.9×10^{-10}	1.4×10^{-8} 6.7×10^{-9}	3.1×10^{-9} 6.4×10^{-9}

Table I (Continued)

ORDER	RANK	NORM 1	NORM 2	NORM 3	NORM 4
40 x 40	1	4.4×10^{-4} 4.3×10^{-4}	1.2×10^{-11} 1.2×10^{-11}	6.0×10^{-10} 4.8×10^{-10}	2.4×10^{-10} 1.1×10^{-5}
40 x 40	2	4.1×10^{-4} 4.1×10^{-4}	1.2×10^{-8} 9.2×10^{-11}	1.6×10^{-8} 5.7×10^{-9}	1.6×10^{-8} 9.4×10^{-6}
40 x 40	5	2.0×10^{-4} 3.2×10^{-4}	3.1×10^{-6} 1.6×10^{-10}	7.4×10^{-8} 1.1×10^{-8}	1.7×10^{-7} 7.0×10^{-6}
40 x 40	10	1.4×10^{-4} 1.9×10^{-4}	3.0×10^{-6} 6.5×10^{-10}	9.1×10^{-8} 1.4×10^{-8}	2.8×10^{-7} 8.5×10^{-6}
40 x 40	15	1.2×10^{-4} 2.1×10^{-4}	2.6×10^{-6} 9.7×10^{-10}	8.1×10^{-8} 1.3×10^{-8}	2.6×10^{-7} 7.2×10^{-6}
40 x 40	20	1.3×10^{-4} 1.4×10^{-4}	6.7×10^{-8} 1.0×10^{-9}	4.3×10^{-8} 1.1×10^{-8}	4.6×10^{-7} 4.7×10^{-6}
40 x 40	30	3.7×10^{-5} 3.5×10^{-5}	6.2×10^{-8} 1.2×10^{-9}	3.0×10^{-8} 1.2×10^{-8}	5.3×10^{-7} 4.5×10^{-6}
40 x 40	35	1.1×10^{-5} 1.0×10^{-5}	5.5×10^{-7} 1.9×10^{-9}	3.1×10^{-8} 2.7×10^{-8}	1.0×10^{-6} 4.3×10^{-6}
40 x 40	39	5.4×10^{-6} 7.2×10^{-6}	1.1×10^{-7} 2.1×10^{-9}	2.6×10^{-8} 3.8×10^{-8}	2.4×10^{-7} 3.8×10^{-7}

Table I (Concluded)

ORDER	RANK	NORM 1	NORM 2	NORM 3	NORM 4
40 x 40	40	2.6×10^{-6} 2.5×10^{-6}	7.1×10^{-9} 8.2×10^{-9}	3.7×10^{-8} 9.3×10^{-8}	2.6×10^{-7} 1.3×10^{-7}
45 x 40	30	7.2×10^{-5} 6.9×10^{-5}	6.7×10^{-8} 7.9×10^{-10}	3.0×10^{-8} 1.1×10^{-8}	2.8×10^{-7} 3.7×10^{-6}
45 x 40	35	2.2×10^{-5} 1.8×10^{-5}	1.2×10^{-7} 9.4×10^{-10}	2.8×10^{-8} 1.4×10^{-8}	1.5×10^{-7} 3.1×10^{-6}
45 x 40	39	2.5×10^{-5} 5.7×10^{-6}	4.8×10^{-7} 9.0×10^{-10}	1.9×10^{-8} 1.7×10^{-8}	1.3×10^{-7} 1.3×10^{-7}

Table II

Computer Times Required to Calculate the Generalized Inverse of a 10 x 10 Nonsingular Matrix

PROGRAM	MEAN TIME (MILLI-SECONDS)
GINV2	34.9
SEQINV	50.3
JPLUS	390.5
APLUS	417.2

Table III

Computer Times Required by GINV2 and APLUS for Selected Randomly Generated Matrices.

MATRIX		MEAN TIME (MILLI-SECONDS)	
TYPE	ORDER	GINV2	APLUS
NONSINGULAR	10 x 10	34.9	417.2
NONSINGULAR	20 x 20	245.8	3391 *
NONSINGULAR	30 x 30	784.1	10981 **
NONSINGULAR	40 x 40	1825.0	27383 ***

(Note: The mean is, in general, calculated for a block of 100 matrices)

*Calculated for a block of only 50 matrices
**Calculated for a block of only 25 matrices
***Calculated for a block of only 15 matrices

Table IV

Computer Times Required by GINV2 and APLUS for Selected Randomly Generated Matrices.

MATRIX TYPE		MEAN TIME (MILLI-SECONDS)	
ORDER	RANK	GINV2	APLUS
10 x 10	1	43	29
10 x 10	2	43	377
10 x 10	3	42	395
10 x 10	4	42	404
10 x 10	5	41	414
10 x 10	6	40	436
10 x 10	7	39	442
10 x 10	8	38	461
10 x 10	9	37	477
10 x 10	10	35	416
20 x 20	1	301	266
20 x 20	2	300	3243
20 x 20	3	300	3204
20 x 20	4	299	3354
20 x 20	5	294	3254
20 x 20	6	293	3390
20 x 20	7	291	3409

Table IV (Continued)

MATRIX TYPE		MEAN TIME (MILLI-SECONDS)	
ORDER	RANK	GINV2	APLUS
20 x 20	8	287	3468
20 x 20	9	283	3461
20 x 20	10	282	3556
20 x 20	11	282	3533
20 x 20	12	278	3612
20 x 20	13	277	3730
20 x 20	14	275	3776
20 x 20	15	269	3794
20 x 20	16	264	3710
20 x 20	17	259	3943
20 x 20	18	254	3857
20 x 20	19	249	3993
20 x 20	20	243	3354

REFERENCES

(1) Moore, E. H.: Bull. Amer. Math. Soc., 26(1920), pp. 394-395.

(2) Penrose, R.: A Generalized Inverse for Matrices. Proc. Camb. Philos. Soc., 51(1955), pp. 406-413.

(3) Bjerhammer, A.: Application of Calculus of Matrices to the Method of Least Squares with Special References to Geodetic Calculations. Trans. Roy. Inst. Tech. Stockholm, 49(1951), pp. 1-86.

(4) Penrose, R.: On Best Approximate Solutions of Linear Matrix Equations. Proc. Camb. Philos. Soc., 52(1956), pp. 17-19.

(5) Ben-Israel, A.; Wersan, S. J.: An Elimination Method for Computing the Generalized Inverse of an Arbitrary Complex Matrix. Journal of the Association of Computing Machinery, 10(October, 1963), pp. 532-537.

(6) Householder, A.: The Theory of Matrices for Numerical Analysis. Blaisdell Publishing Company, (New York) 1964.

(7) Ben-Israel, A., and Charnes, A.: Contributions to the Theory of Generalized Inverses. Journal of SIAM, 11(September, 1963), pp. 667-699.

(8) Decell, H. P., Jr.: An Application of the Cayley-Hamilton Theorem to Generalized Matrix Inversion. SIAM Review, 7(October, 1965), pp. 526-528.

(9) Decell, H. P., Jr.: An Application of Generalized Matrix Inversion to Sequential Least Squares Parameter Estimation. NASA TN D-2830 (May, 1965).

(10) Decell, H. P., Jr., and Kahng, S. W.: An Iterative Method for Computing the Generalized Inverse of a Matrix. NASA TN D-3464, (June, 1966).

(11) Burras, W. R.; Rust, B., and Schneeberger, C.: A Simple Algorithm for Computing the Generalized Inverse of a Matrix. Communications of the A.C.M., 9(May, 1966), pp. 381-387.

APPENDIX A

THEORETICAL BACKGROUND

The following is a brief description of the mathematics used in each of the five programs examined in this study.

(1) PEN2 is based on a method, devised by R. Penrose (ref. 2), which computes the generalized inverse of a matrix A using the formula

(a) $A^{+} = DA^{T}$

where D is any matrix satisfying

(b) $A^{T}A = D(A^{T}A)^{2}$

(Note that multiplication of (a) on the right by $A^{+}(A^{+})^{T}A^{+}$ yields (b).)

Define a sequence of matrices C_k, $k = 1,2,...$ by

$$C_1 = I$$

$$C_2 = I \cdot \frac{1}{1}\,\mathrm{trace}\,(C_1B) - C_1B$$

$$\vdots$$

$$C_{j+1} = I \cdot \frac{1}{j}\,\mathrm{trace}\,(C_jB) - C_jB$$

where $B = A^{T}A$.

If r is the rank of A, then $C_{r+1}B = 0$ and trace $C_rB \neq 0$.

Then D can be calculated by the formula

$$D = \frac{rC_r}{\text{trace }(C_rB)} .$$

(2) SEQINV is based on a sequential method for computing the generalized inverse (ref. 9). Let A_{n-1} be the matrix containing the first (n-1) rows of the matrix A and a_n be the n^{th} row of A. The generalized inverse of A is then calculated sequentially by the formula

$$A_n^+ = \left(A_{n-1}^+ - p_n a_n A_{n-1}^+ \ \vdots \ p_n\right)$$

$$\text{with } p_n = \begin{cases} \left(a_n(I - A_{n-1}^+ A_{n-1})\right)^+ & \text{if } a_n \neq a_n A_{n-1}^+ A_{n-1} \\ \left(1 + a_n(A_{n-1}^+)^T a_n^T\right)^{-1} A_{n-1}^+ (A_{n-1}^+)^T a_n^T & \text{if } a_n = a_n A_{n-1}^+ A_{n-1} \end{cases}$$

$A_n^+ A_n$ and $A_n^+(A_n^+)^T$ are computed sequentially as follows:

$$A_n^+ A_n = \left(A_{n-1}^+ - p_n a_n A_{n-1}^+ \ \vdots \ p_n\right) \begin{pmatrix} A_{n-1} \\ a_n \end{pmatrix}$$

$$= A_{n-1}^+ A_{n-1} - p_n a_n A_{n-1}^+ A_{n-1} + p_n a_n .$$

If $a_n = a_n A_{n-1}^+ A_{n-1}$, then $A_n^+ A_n = A_{n-1}^+ A_{n-1}$.

If $a_n \neq a_n A_{n-1}^{+} A_{n-1}$ then $A_n^{+} A_n = A_{n-1}^{+} A_{n-1} + p_n p_n^{+}$.

$$A_n\left(A_n^{+}\right)^T = \left(A_{n-1}^{+} - p_n a_n A_{n-1}^{+} \;\vdots\; p_n\right) \begin{pmatrix} \left(A_{n-1}^{+}\right)^T - \left(A_{n-1}^{+}\right)^T a_n^T \, p_n^T \\ p_n^T \end{pmatrix}$$

$$= A_{n-1}^{+}\left(A_{n-1}^{+}\right)^T - p_n a_n A_{n-1}^{+}\left(A_{n-1}\right)^T$$

$$+ p_n\left(a_n A_{n-1}^{+} (A_{n-1}^{+})^T a_n^T + 1\right) p_n^T$$

$$- A_{n-1}^{+}\left(A_{n-1}^{+}\right)^T a_n^T p_n^T$$

If $a_n = a_n A_{n-1}^{+} A_{n-1}$ then

$$A_n^{+}\left(A_n^{+}\right)^T = A_{n-1}^{+}\left(A_{n-1}^{+}\right)^T - p_n a_n A_{n-1}^{+}\left(A_{n-1}^{+}\right)^T$$

(3) The method JPLUS uses the property that if A is an n x n symmetric matrix, there exists an n x n matrix P such that

$$PAP^T = D$$

where D is the diagonal matrix whose diagonal elements are the eigenvalues of A. It can be shown that

$$A^{+} = P^T D^{+} P \; .$$

If A is an n x m matrix, then let $B = A^T A$. Then, by the above, there is an m x m matrix P such that

$$D = PBP^T$$

where D is again diagonal with the eigenvalues of B as its diagonal elements.

Then

$$B^+ = P^T D^+ P \ .$$

Using the fact that

$$A^+ = (A^T A)^+ A \ ,$$

it follows that

$$A^+ = B^+ A.$$

(4) APLUS in an iterative method (ref. 10) based on the following formula

$$X_n = X_{n-1}(2I - AX_{n-1})$$

where A is the matrix whose generalized inverse is to be computed. After initially setting $X_o = \dfrac{A^T}{||AA^T||}$, the iteration process is continued until a near-zero test indicates that X_m can be assumed to be A^+ for some m. (Note that here, $||AA^T|| = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} \left(AA^T\right)_{ij}^2\right)^{1/2}$.)

(5) GINV2 is based on an extension of the orthogonalization process for computing the inverse of nonsingular matrices (ref. 11). The problem of computing the generalized inverse of an n x m matrix A can be reduced to the problem of computing the inverse of an n x m matrix $(R \vdots S)$, partitioned so that R is the matrix of all linearly independent columns (say there are k of them) of A and S is the matrix consisting of the remaining (m-k) dependent columns. (Note that $(R \vdots S)$ can be obtained from A by a finite number of permutations of columns of A.)

A Gram-Schmitt orthogonalization process is performed on $(R \vdots S)$ and the same operations performed simultaneously on the n x n identity matrix. The result of the Gram-Schmitt on $(R \vdots S)$ is a matrix of the form $(Q \vdots 0)$ where Q is an n x k matrix and 0 is the n x (m-k) zero matrix. The identity matrix will become of the form

$$\begin{pmatrix} Z & -U \\ 0 & I_{n-k} \end{pmatrix}$$ where Z is k x k

0 is the (n-k) x k zero matrix, -U is k x (n-k) and I_{n-k} is the (n-k) identity matrix.

Performing a Gram-Schmitt orthogonalization on

$$\begin{pmatrix} -U \\ I_{n-k} \end{pmatrix}$$ yields a matrix of the

form

$$\begin{pmatrix} -UP \\ P \end{pmatrix} \text{ where P is (n-k) x (n-k) and}$$

-UP is k x n-k.

After these orthogonalizations are completed, all of the matrices necessary for the computation of A^+ have been calculated.

It can be shown that A^+ is given by

$$A^+ = \begin{pmatrix} Z & -UP \\ 0 & P \end{pmatrix} \begin{pmatrix} Q^T \\ (UP)^T ZQ^T \end{pmatrix} = \begin{pmatrix} ZQ^T - (UP)(UP)^T ZQ^T \\ P(UP)^T ZQ^T \end{pmatrix}.$$

APPENDIX B

ANALYSIS OF VARIANCE

This appendix describes the analysis of variance used to test the hypothesis that the accuracies of GINV2 and APLUS are equal. The observations used in the analysis were the absolute values of the logarithms of the norms, which were defined on page 6 of the report.

CASE I. Full rank.

A three-way full factorial was assumed as the model, i.e.

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + (\beta\gamma)_{jk} + \rho_{ijk}$$

where:

$i = 1,2$

$j = 1,2,\ldots,15$

$k = 1,2,3,4$

α_1: effect due to APLUS

α_2: effect due to GINV2

β_j: effect due to order (orders considered were— (2 x 2), (2 x 3), (3 x 2), (4 x 5), (10 x 8), (8 x 10), (10 x 10), (20 x 20), (30 x 10), (10 x 30), (30 x 30), (35 x 35), (40 x 15), (40 x 40), (45 x 40))

γ_k: effect due to norms

$(\alpha\beta)_{ij}$: effect of interaction due to α_i and β_j

$(\alpha\gamma)_{jk}$: effect of interaction due to α_i and γ_k

$(\beta\gamma)_{jk}$: effect of interaction due to β_j and γ_k

ρ_{ijk}: effect due to random error

The null hypothesis is as follows:

H_o: accuracy of GINV2 = accuracy of APLUS

The results of the analysis are tabulated below:

Analysis of Variance

Source of Variation	D.F.	SS	MS	F
Method	1	0.01391	0.01397	0.1405
Order	14	19.3749	1.3837	13.91
Norm	3	94.9728	31.6576	318.89
Method x Order	14	7.0123	0.5009	
Method x Norm	3	1.4788	0.4930	
Order x Norm	42	12.2357	0.2912	
Error	42	4.1762	0.0994	

Since the F ratio for the methods is .1405 it is not possible to reject H_o based on the data so far collected.

CASE II. Nonfull rank.

The model assumed in this case was a four-way partially-nested factorial, in which rank is nested in any order. The results of this case are tabulated below.

H_o: accuracy of GINV2 = accuracy of APLUS

Analysis of Variance

Source of Variation	D.F.	SS	MS	F
Method	1	2.596	2.596	60
Order	9	22.525	2.503	
Rank	1	0.633	0.033	
Norm	3	324.595	108.198	
⋮ Interactions	⋮	⋮	⋮	
Error		1.1496	0.0425	

Since the F ratio for the method is 60, the hypothesis that the two methods are equal must be rejected in favor of the hypothesis that the accuracy of GINV2 is better than that of APLUS.