A COMPUTER PROGRAM FOR CALCULATING THE MOORE-PENROSE GENERALIZED INVERSE OF A RECTANGULAR OR SINGULAR MATRIX

SUSY PHILIPOSE

University of Bombay, Bombay

(*Received* 18 *October* 1968 ; *revised* 6 *March* 1969)

The inverse of a nonsingular matrix is often used to solve the system of simultaneous equations when the number of equations are the same as the number of unknowns. The inverse of a singular or rectangular matrix can also be defined such that it can be used to solve the system of equations when the number of equations are not equal to the number of unknowns. This paper deals with some of the results of the above inverse and a computer program to calculate the same.

An inverse of a nonsingular matrix is used to solve the system of consistent linear equations $AX=Y$ where X and Y are column vectors and A is a $n \times n$ nonsingular matrix. If solution to the above equation is $X=A^{-1}Y$, than A^{-1} is called the inverse of the matrix A. The possibilities of a similar representation were explored in the case of singular matrix and rectangular matrix so that for an equation $AX = Y$, $X = BY$ is a solution. If such a representation is possible B can be called the generalized inverse of a matrix A.

This generalized inverse is of common interest in these days as it is giving solution to any consistent system of linear equations. Generalized inverse can be used to solve the linear programing problems and many other statistical problems. The two important applications of generalized inverse are the minimum approximate solution and the basic approximate solution. In minimum approximate solution we minimize Euclidean norm and in basic approximate solution the components of X are at the most r where r is the rank of the matrix A.

The notion of generalized inverse (G-inverse) was first introduced by Moore[1] and afterwards by Penrose[2]. Even though Penrose has defined this inverse different from Moore both the inverses are one and the same. Rao[3] also contributed to this theory but this inverse is different from Moore-Penrose inverse. Moore inverse is unique for a matrix while Rao's generalized inverse is not unique.

DEFINITION AND PROPERTIES OF G-INVERSE

Rao[3] gives the definition of G-inverse as follows :

Definition : Consider an $m \times n$ matrix A of any rank. A generalized inverse (or a G-inverse) of A is a $n \times m$ matrix denoted by $\bar{A}$ such that for any vector Y for which $AX=Y$ is a consistent equation, $X = \bar{A}Y$ is a solution. Moore[1] defined a generalized inverse as follows ;

Definition : A generalized inverse of a matrix A denoted by A^+ exists satisfying the following properties.

(a) $AA^+A=A$ (b) $A^+AA^+ = A^+$

(c) $(AA^+)' = AA^+$ (d) $(A^+A)' = A^+A$

and A^+ is unique.

The relation (a) of Moore-Penrose inverse will hold for Rao's G-inverse. In fact Moore-Penrose inverse is a particular case of Rao's inverse satisfying three more conditions. In both the cases the general solution to the system of equation $AX=Y$ is given by

$$X = BY + (H - I) Z$$

where B is the G-inverse of A, $H = BA$, I is the identity matrix and Z is an arbitrary vector.

In this paper we deal with some of the properties of Moore-Penrose G-inverse and computation program in FORTRAN language to calculate the same. The G-inverse A^+ is unique and it is equal to A^{-1} when the matrix is nonsingular. Consider an equation $AX=Y$ where A is a $m \times n$ matrix of rank r and X, Y are 2 column vectors. Then the minimum approximate solution for X is unique and is $X_m = A^+ Y$. Similarly the basic approximate solution which is not unique is $X_b = A^* Y$ where A^* is related to A^+. When $r=n=m$, that is when A is nonsingular, $A^*=A^+=A^{-1}$ and the solution $X_b = X_m = A^{-1} Y$. When $r=n \leqslant m$, $A^*=A^+$ and $X_b = X_m = A^+ Y$. The definition and the relations of A^+ will hold good in complex field also if we define the transpose as the conjugate transpose.

COMPUTER PROGRAM

The computer program adopted in this paper is based on the method of computation by Boot[4] and for completeness the proof is quoted here.

Proof: The G-inverse A^+ is obtained as the solution to the extremum problem: Minimize $tr\ YY'$ subject to

$$A'AY = A' \tag{1}$$

Since (1) is consistent, we have to consider only r of the n equations in (1). Let the first r rows be independent. If the first r rows of (1) are not independent we can interchange the rows and make the first r rows as the first independent rows. Let B' $(r \times n)$ be formed by the first r rows of $A'A$ and $C'(r \times m)$ is formed by the first r rows of A'. Let Y_i be the ith column of Y and C_i the ith column of C'. The vector Y_i minimizing $Y'_i Y_i$ subject to $B' Y_i = C_i$ is, using Langrangian technique, given by the solution of the system.

$$\begin{bmatrix} I & B \\ B' & 0 \end{bmatrix} \cdot \begin{bmatrix} Y_i \\ U_i \end{bmatrix} = \begin{bmatrix} 0 \\ C_i \end{bmatrix}$$

where U_i is a column vector of r Langrangians

Since $$\begin{bmatrix} I - B(B'B)^{-1}B' & B(B'B)^{-1} \\ (B'B)^{-1}B' & -(B'B)^{-1} \end{bmatrix} \begin{bmatrix} I & B \\ B' & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

The solution is given by $\quad Y_i = B(B'B)^{-1} C_i$

Hence $\quad Y = B(B'B)^{-1} C'$.

This computation necessitates inversion of a matrix of order $r \times r$ whereas the method described by Greville[5] requires two such inversions.

FORTRAN PROGRAM

A FORTRAN program which works on IBM 1620 computer is described in this paper. This program is suitable for any matrix of order (12×12) or less than this. The maximum position occupied by any element of the matrix, including the decimal point and the sign, is 6 or less. The program also works for matrix of higher order and bigger elements, if necessary alterations are made in the Dimension statement and in the Format statement. Special care has been taken to see that the program takes minimum number of memory spaces. This single program can be used to find the G-inverse of a large number of matrices by a single compilation.

The program consists of the main program and three small subroutine subprograms. The subroutine *ANMT* is used to find $A'A$ where A is a matrix of order $m \times n$ and A' is the transpose of A. Subroutine *AVERT* is used to find out the regular inverse A^{-1} of A where A is a nonsingular matrix. Subroutine *AMULT* is used to find $A \times B$ where A is a matrix of order $(1 \times m)$ and B is a matrix of order $(m \times n)$. In the main program the input and output are handled.

In this paper Moore-Penrose G-inverse has been worked out for a matrix of order (10×12) on 1620 IBM computer. To get a greater accuracy a control card* FANDK 1004 is put so that all calculations are done in 10 significant digits for decimal figures and 4 digits for whole figures. If the accuray is still to be increased an appropriate control card can be put instead of the former one. The computer as such calculates 8 significant digits for decimal figures and 4 digits for whole figures without this control card. First the 3 subroutine subprograms are compiled and afterwards the main program.

Subroutine ANMT : This subroutine is used to calculate $A'A$, where $A(M \times N)$ is a matrix. As the matrix $A'A$ is symmetric the program calculates the elements below the principal diagonal and any element a_{ji} is put as equivalent to a_{ij}. The calculated matrix $A'A$ occupies the space of the original matrix A and so the subprogram requires very little memory space. X is a one dimensional array which is used for transferring some of the calculated values.

```
SUBROUTINE ANMT (A,X,M,N)
      DIMENSION A (12, 12), X (13)
      J=1
15    DO 10 L = J,N
      X (L) = 0·0
      DO 10 K = 1,M
10    X (L) = X (L) + A (K,J)*A (K,L)
      DO 14 L = J,N
14    A (L,J) = X (L)
      IF (J—N) 12, 11, 11
12    J = J + 1
      GO TO 15
11    CONTINUE
      K = 1
      N1 = N — 1
      DO 16 J = 1, N1
      K = K + 1
      DO 16 I = K, N
16    A (J, I) = A (I, J)
      RETURN
      END
```

Subroutine AMULT : This program is used to multiply two matrices A (L,M) and B (M,N) and the result occupies the space C (L,N).

```
SUBROUTINE AMULT (A,B,C,L,M,N)
      DIMENSION A (12, 12), B (12, 12), C (12, 12)
      DO 10 I = 1, L
      DO 10 J = 1, N
      C (I, J) = 0
      DO 10 K = 1, M
10    C (I, J) = C (I, J)
           + A (I, K)* B (K, J)
      RETURN
      END
```

Subroutine AVERT: This subroutine is used to calculate the regular inverse of the matrix A. It is programed to calculate the inverse by sweep out method. The order of the matrix is N and the inverse occupies the space B.

```
SUBROUTINE AVERT (A,N,B)
      DIMENSION A (12, 24), B (12, 12)
      M = 2*N
      MM = N+1
      DO 20 I = 1, N
      DO 20 J = MM, M
20    A (I, J) = 0·0
      DO 21 I = 1,N
      II = I+N
21    A (I, II) = 1·0
      DO 10 J = 1,N
      JJ = J+1
      DIAG = A (J, J)
      DO 14 I = JJ, M
14    A (J, I) = A (J,I)/DIAG
      DO 10 K = 1, N
      IF (K—J) 13, 10, 13
13    DO 10 I = JJ, M
      A (K, I) = A (K,I) —A (J,I)* A (K,J)
10    CONTINUE
      DO 90 I = 1,N
      DO 90 J = MM, M
      LLL = J—N
90    B (I,LLL) = A (I, J)
      RETURN
      END
```

MAIN PROGRAM

To find the independent rows, the matrix A' is reduced by sweep out method. As the pivoted rows are the independent rows the matrix A' is rearranged with the pivoted rows as the first rows. Further calculations are done as is given by Boot[4]. After it has calculated the G-inverse of the first set of data, the computer comes to the PAUSE statement and it halts temporarily. The MANUL light will be on in the console. If we have another set of data to calculate the G-inverse, the START key on the console is pressed and the next set of INPUT data is put in the READER. By pressing the START key of the READER, computer will work for the next set of data and so on. If it has calculated the *G*-inverse of all the matrices stop the computer by pressing the STOP key.

```
      C TO CALCULATE MOORE-PENROSE G-INVERSE
      DIMENSION A (12, 12), CDASH (12, 12), X (13), B (12,12), BA (12, 12),
         BB (12, 24)
      C TO READ THE ORDER OF THE MATRIX
50    READ 16, M, N
16    FORMAT (2 I 2)
      M1 = M+1
      C TO READ THE MATRIX
      READ 27, ( (A (I,J), J = 1, 12), I = 1, 12)
27    FORMAT (12F6·0)
      DO 75 J = 1, N
      C TO NUMBER THE ROWS OF A-DASH
      BB (J, 1) = J
      DO 75 I = 1, M
      BB (J, I+1) = A(I, J)
      C TO CALCULATE A-DASH
75    CDASH (J, I) = A (I, J)
      C TO CALCULATE A-DASH-A
      CALL ANMT (A,X,M,N)
      C TO REDUCE THE MATRIX BY SWEEP OUT METHOD
      I = 1
      C TO FIND THE RANK OF THE MATRIX
      L1 = 0
      I1 = 1
5     I1 = I1+1
      L = I
62    IF (ABSF(BB(L, I1))—0·1E—06) 3, 3, 2
3     IF (L—N) 12, 63, 63
63    IF (I—N) 71, 9, 9
71    IF (I1—M1) 5, 9, 9
12    L = L+1
      GO TO 62
2     L1 = L1+1
      IF (I—L) 60, 61, 61
```

```
60   DO 4 K = 1, M1
 4   X (K) = BB (L,K)
55   L2 = L—1
     DO 52 K = 1, M1
52   BB (L,K) = BB (L2, K)
     IF (L2—I) 54, 54, 53
53   L = L—1
     GO TO 55
54   DO 56 K = 1, M1
56   BB (I,K) = X (K)
61   DO 40 K = 1, N
40   X (K) = BB (K, I1)
     DO 72 L = I1, M1
72   BB (I,L) = BB (I,L)/X (I)
     IF (I—N) 8, 9, 9
 8   L3 = I+1
     DO 7 K = L3, N
     DO 7 L = I1, M1
 7   BB (K,L) = BB (K,L)—BB (I,L)* X (K)
     IF (I1—M1) 51, 9, 9
51   I = I+1
     GO TO 5
 9   L = L1
     PRINT 26, L
26   FORMAT (/2X, 22HRANK OF THE MATRIX IS = 13/)
     C TO FIND THE FIRST R INDEPENDENT ROWS
     J = 1
     DO 22 K = 1, N
     DO 22 I = 1, L
     L1 = BB (I,1)
     IF (K—L1) 22, 28, 22
28   X (J) = K
     J = J+1
22   CONTINUE
     C TO REARRANGE THE ROWS OF A-DASH AND A-DASH-A SO THAT
       C-DASH IS FORMED BY
     C THE FIRST R ROWS OF A-DASH AND B IS FORMED BY THE
       TRANSPOSE OF THE FIRST R
```

```
C INDEPENDENT ROWS OF A-DASH-A
   DO 23 I = 1, L
   DO 23 J = 1, N
   DO 23 K = 1, M
   L1 = X(I)
   CDASH (I,K) = CDASH (L1, K)
23 B (J,I) = A(L1, J)
   DO 14 I = 1, N
   DO 14 J = 1, L
   C TO CALCULATE B-DASH-B
14 A (I,J) = B (I,J)
   CALL AMNT (A,X,N,L)
   DO 17 I = 1, L
   DO 17 J = 1, L
17 BB (I,J) = A(I,J)
   C TO CALCULATE THE INVERSE OF B-DASH-B
   CALL AVERT (BB,L,A)
   C TO CALCULATE B (B-DASH-B) INVERSE C-DASH
   CALL AMULT (B,A, BA,N,L,L)
   CALL AMULT (BA, CDASH, A,N,L,M)
   PRINT 37
37 FORMAT (/5X, 9HG-INVERSE/)
38 PRINT 39
39 FORMAT (/5X, 1HI, 5X, 1HJ, 10X, 6HA (I,J)/)
   DO 38 I = 1, N
   DO 38 J = 1, M
   PRINT 40, I,J,A (I,J)
40 FORMAT (/3X, I3, 3X, I3, 40, 3X, E20·10/)
   PAUSE 3333
   C TO CALCULATE THE G-INVERSE FOR THE NEXT SET OF DATA
   GO TO 50
   END
```

INPUT

There are 13 cards as data cards (see Table 1). The first card is allotted to specify the order of the matrix. If the matrix is of order 10 × 12, the first 4 columns of the first card are punched as 1012 (if it is 4 × 4 matrix it should be punched as 0404) one figure in one column each. In the next m cards (if the matrix is of order $m \times n$) m rows of the matrix are punched. Six columns are allotted to each element of the matrix so that an element can take at the most 6 columns. So each card will have n elements of that particular row. For example if $a_{4,10}$ is an element in the matrix it is punched in the 5th card of the input cards from column 55 to column 60 (both inclusive). The remaining 13—$(m+1)$ cards of the input are blank.

TABLE 1

INPUT

MATRIX $A(I,J)$ 10 × 12

Card 1 Col. 1–4
1012

Columns / Cards	1–6	7–12	13–18	19–24	25–30	31–36	37–42	43–48	49–54	55–60	61–66	67–72	73–80
2	23·4	–18·3	0·09	0	–231·0	78·3	–12·3	81·9	98·32	4·31	–21·8	19·5	
3	41·9	21·81	–12·5	3·4	33·5	0	13·3	29·38	–0·34	88·5	15·3	1·28	
4	92·3	43·1	3·43	–1·32	9·14	7·98	–12·3	9·94	7·94	73·6	9·38	4·10	
5	5·12	0	–1·32	8·19	12·10	41·39	0	91·4	45·10	8·96	8·43	–1·78	
6	12·18	2·38	64·14	–5·2	32·1	91·3	4·89	0	–7·38	7·94	–14·3	9·14	
7	1·39	20·1	0	3·4	9·14	4·51	10·42	26·1	33·19	120·3	0	81·34	
8	8·97	–7·2	14·88	0	–11·3	8·24	–3·49	–8·2	42·91	0	23·8	19·32	
9	91·32	49·1	–2·39	7·38	4·47	–91·3	506·6	39·61	–10·1	4·32	71·3	0	
10	39·9	10·4	–1·38	6·1	0	41·98	18·32	201·8	34·7	1·98	0·09	48·3	
11	6·84	5·84	8·82	–2·75	39·61	9·94	6·14	37·1	39·1	–2·34	1·28	–1·38	
12													
13													

TABLE 2

OUTPUT

I	*J*	*A (I,J)*
1	1	0·9489939470E–03
1	2	0·4847050140E–02
1	3	–0·1256023952E–01
1	4	0·4951871250E–02
1	5	0·9334097150E–03
1	6	–0·5520526470E–02
1	7	0·6073084102E–02
1	8	–0·1025874900E–03
1	9	0·4160125583E–02
1	10	0·4887907754E–02
2	1	–0·6531954308E–02
2	2	–0·4746753120E–01
2	3	0·4344721567E–01
2	4	0·2934706910E–01
2	5	–0·5867980124E–02
2	6	0·1379439590E–01
2	7	–0·2021174951E–01
2	8	0·2118244640E–02
2	9	–0·9913279126E–02
2	10	–0·2012695486E–01
3	1	0·4218625663E–02
3	2	0·2194724787E–01
3	3	–0·3256024813E–01
3	4	–0·1292443112E–01
3	5	0·7790790347E–02
3	6	–0·6831888520E–02
3	7	0·1028378923E–01
3	8	–0·1043706385E–02
3	9	0·7134008578E–02
3	10	0·1977579399E–01
4	1	–0·7649737060E–02
4	2	–0·3454344057E–01
4	3	0·4694712553E–01
4	4	0·1952202214E–01
4	5	–0·4830386411E–02
4	6	0·9530312300E–02
4	7	–0·5814348110E–02
4	8	0·1752507094E–02
4	9	–0·8699660152E–02
4	10	–0·3456583370E–01
5	1	–0·3146108328E–02
5	2	–0·1078242600E–02
5	3	0·2041897735E–02
5	4	0·3774487480E–03
5	5	0·6927511540E–03
5	6	0·7297917100E–03
5	7	–0·4718317840E–03
5	8	–0·1450682907E–03
5	9	–0·3933635256E–03
5	10	0·5511805001E–02
6	1	–0·2869079912E–02
6	2	–0·1670215006E–01
6	3	0·2912190878E–01
6	4	0·9282074580E–02
6	5	0·4282599574E–02
6	6	0·4282827920E–02
6	7	–0·4421591432E–02
6	8	0·8686962610E–03
6	9	–0·6489156672E–02
6	10	–0·2058966189E–01
7	1	0·7262605636E–03
7	2	–0·1487432470E–03
7	3	0·8858875500E–03
7	4	–0·1813571856E–02
7	5	0·1418459583E–02
7	6	0·1093522891E–02
7	7	–0·2972927690E–02
7	8	0·1908915936E–02
7	9	–0·1349322082E–02
7	10	0·1677483525E–02
8	1	0·1502862612E–02
8	2	0·1086878717E–01
8	3	–0·8717082410E–02
8	4	–0·6637956390E–02
8	5	–0·1403858400E–04
8	6	–0·3463763740E–02
8	7	–0·3566912000E–04
8	8	–0·5965631740E–03
8	9	0·6128516877E–02
8	10	0·5359568278E–02
9	1	0·1944547330E–02
9	2	–0·5452658660E–02

(*contd.*)

(*Table 2 contd.*)

I	J	A (I,J)	I	J	A (I,J)
9	3	0·3817242769E–02	11	2	0·2400052051E–02
9	4	0·2474299730E–02	11	3	0·1720635121E–01
9	5	–0·3228579382E–02	11	4	0·3842166690E–03
9	6	0·2770713180E–02	11	5	–0·1608016683E–02
9	7	0·7851396800E–03	11	6	–0·3603987484E–02
9	8	0·1998348916E–03	11	7	0·2268264820E–01
9	9	–0·4846896040E–02	11	8	0·4494958005E–03
9	10	0·1609843896E–01	11	9	–0·1548039638E–03
10	1	0·3595341719E–02	11	10	–0·2795350703E–01
10	2	0·2175601326E–01	12	1	–0·3886328032E–02
10	3	–0·1217347106E–01	12	2	–0·1660952416E–01
10	4	–0·9958678430E–02	12	3	0·2658602810E–02
10	5	0·2094241063E–02	12	4	0·5739417070E–02
10	6	–0·8978985600E–03	12	5	–0·6398676450E–03
10	7	0·4301312280E–03	12	6	0·8578888960E–02
10	8	–0·8030223240E–03	12	7	0·4859023399E–02
10	9	0·7215698080E–04	12	8	0·2795421330E–03
10	10	0·7713404066E–02	12	9	0·4730555008E–02
11	1	–0·5198265867E–02	12	10	–0·1173628923E–01

OUTPUT

Rank of the matrix is 10 G-inverse (see Table 2).

The time taken for the compilation and execution of the program is 5 minutes on 1620 IBM computer and the memory space required is about 16000.

ACKNOWLEDGEMENT

My sincere thanks to Professor M. C. Chakrabarti for his kind guidance in preparing this paper.

REFERENCES

1. MOORE, E. H., *Mem. Amer. Philosophical Soc.*, I (1935).
2. PENROSE, R., *Proc. Cambridge Philosophical Soc.*, **51** (1955), 406.
3. RAO, C. R., *J. Royal Statist. Soc.*, **B24** (1962), 152.
4. BOOT, J. C. G., *Amer. Mathematical Monthly*, **70** (1963), pp 302-303.
5. GREVILLE, T. N. E., *J. Soc. Industr. Appl. Maths. Rev.*, 1 (1959), pp 38–43.