N72-24210

# NASA TECHNICAL MEMORANDUM

NASA TM X- 64654



# ITERATIVE COMPUTATION OF GENERALIZED INVERSES, WITH AN APPLICATION TO CMG STEERING LAWS

By James W. Steincamp
Program Development

December 7, 1971

**NASA**

*George C. Marshall Space Flight Center*
*Marshall Space Flight Center, Alabama*

TECHNICAL REPORT STANDARD TITLE PAGE

| | | |
|---|---|---|
| 1. REPORT NO. TM X- 64654 | 2. GOVERNMENT ACCESSION NO. | 3. RECIPIENT'S CATALOG NO. |

4. TITLE AND SUBTITLE
Iterative Computation of Generalized Inverses, with an Application to CMG Steering Laws

5. REPORT DATE
December 7, 1971

6. PERFORMING ORGANIZATION CODE


7. AUTHOR (S)
James W. Steincamp


8. PERFORMING ORGANIZATION REPORT #


9. PERFORMING ORGANIZATION NAME AND ADDRESS
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812


10. WORK UNIT NO.

11. CONTRACT OR GRANT NO.

12. SPONSORING AGENCY NAME AND ADDRESS
National Aeronautics and Space Administration
Washington, D. C. 20546

13. TYPE OF REPORT & PERIOD COVERED
Technical Memorandum

14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES
Prepared by Navigation and Control Systems Branch, Electronics and Control Division, Preliminary Design Office, Program Development.


16. ABSTRACT

A cubically convergent iterative method for computing the generalized inverse of an arbitrary $M \times N$ matrix A is developed and a Fortran subroutine by which the method was implemented for real matrices on a CDC 3200 is given, with a numerical example to illustrate accuracy. Application to a redundant single-gimbal CMG assembly steering law is discussed.



17. KEY WORDS
Numerical Methods
Generalized Inverses
CMG Steering Laws


18. DISTRIBUTION STATEMENT
STAR Announcement

ERICH E. GOERNER
Director, Preliminary Design Office

| 19. SECURITY CLASSIF. (of this report) | 20. SECURITY CLASSIF. (of this page) | 21. NO. OF PAGES | 22. PRICE |
|---|---|---|---|
| Unclassified | Unclassified | 16 | $3.00 |

MSFC - Form 3292 (May 1969)

# TABLE OF CONTENTS

# LIST OF SYMBOLS

| Symbol | Definition |
|---|---|
| $A = [a_{ij}]$ | $M \times N$ matrix A |
| $A^*$ | $[\bar{a}_{ji}]$, an $N \times M$ matrix |
| $A^+$ | (Moore-Penrose) generalized inverse of the matrix A |
| $I$ | the identity matrix |
| $\lambda_1(B)$ | largest eigenvalue of the positive semidefinite matrix B |
| $\|A\|$ | $\lambda_1(AA^*)$, the spherical matrix norm, or the euclidian vector norm |
| $P_{R(A)}$ | $AA^+$, the projection onto the range space of A |
| $P_{R(A^*)}$ | $A^+A$, the projection onto the range space of $A^*$ |

TECHNICAL MEMORANDUM X- 64654

# ITERATIVE COMPUTATION OF GENERALIZED INVERSES WITH AN APPLICATION TO CMG STEERING LAWS

## INTRODUCTION

The equation

$$Ax = b$$

will possess a solution if, and only if, b lies in the coordinate space spanned by the columns of A, i.e., the range space R(A), and in this case every solution can be put in the form

$$x_g = A^+ b + (I - A^+ A) z$$

where z is some N x 1 column. That solution with z = 0, denoted by $x_b$, has the property that for all z,

$$\|x_b\| \leq \|x_g\|$$

i.e., $x_b$ is the minimum-norm solution. If no solution exists, then there is some vector in R(A) which is closest to b in the least-square sense; this vector is just $x_b = A^+b$ so that for all x

$$\|b - Ax_b\| \leq \|b - Ax\|$$

These properties establish the importance of the generalized inverse for applications. Unfortunately, there is no truly convenient generally applicable formula for its computation. Various elimination algorithms based upon the defining equation, i.e.,

$$AXA = A$$
$$XAX = X$$
$$(AX)^* = AX$$
$$(XA)^* = XA$$

have been devised, but these are sensitive to the preliminary computation of the rank of A. This report describes an iterative scheme for computation of the generalized inverse and incorporates the scheme into a FORTRAN subroutine which may also be used for computing the inverses of nonsingular square matrices.

## FORTRAN SUBROUTINE DESCRIPTION

The Fortran subroutine listed below is called by the following statement:

CALL GNVERS (Y0, T, R, M, N, A, BD, AP)

where A is an $M \times N$ matrix; BD is any number not less than $\lambda_1(AA^*)$; and AP, the output, is the $N \times M$ generalized inverse of A. Y0, T, R are matrices used by GNVERS and need not be defined by the user. A convenient value for BD is

$$\max_{1 \leq i \leq M} \sum_{j=1}^{M} \left| b_{ij} \right|$$

where

$$B = \left[ b_{ij} \right]_1^M = AA^* \quad .$$

The calling program must contain the following statement:

DIMENSION Y0( ), T( ), R( ) .

The arguments in the dimension statement must not be less than the following numbers: N*M, M*M, M*M, respectively. It is assumed that $M \leq N$. If $M > N$, the generalized inverse of $A^*$ should be computed: This stratagem is adopted to keep the dimensions of certain intermediate matrices small (thereby reducing the number of arithmetical operations and computation time) and is acceptable since $(A^*)^+ = (A^+)^*$ .

The algorithm implemented via this subroutine forms successive iterates $Y_1, Y_2, \ldots$ which will converge to $A^+$: The maximum number of iterations is denoted by ITERS, and is set to 20 in the listing below. The iterations cease when

$$E \equiv \max_{\substack{1 \le i \le N \\ 1 \le j \le M}} \left\{ \left| y_{ij}^{(n)} - y_{ij}^{(n-1)} \right| \right\} < \mathrm{TOL} \qquad (n = 2, 3, \ldots)$$

where TOL is set to $10^{-7}$ in the listing below. If convergence does not occur in ITERS iterations, an error statement is printed and the subroutine returns the last iterate to the calling program.

GNVERS makes use of a matrix-multiply subroutine whose calling statement is

CALL FMXMT (M, N, IP, A, B, AB)

which forms the $M \times IP$ product AB of the $M \times N$ matrix A and the $N \times IP$ matrix B. A listing of this subroutine is also given.

The accuracy of the GNVERS subroutine is illustrated by the following example. The $3 \times 4$ matrix A was defined as

$$A = \begin{bmatrix} 4.604359873\text{-}01 & 6.981586633\text{-}01 & 1.637202877\text{-}01 & 7.932543322\text{-}01 \\ 8.176181213\text{-}01 & -5.241646385\text{-}01 & 9.788190850\text{-}01 & 5.955607116\text{-}01 \\ 3.456868410\text{-}01 & -4.876741769\text{-}01 & -1.229181702\text{-}01 & -1.267093084\text{-}01 \end{bmatrix} .$$

A careful computation using an electronic desk calculator gave the following result:

$$A^+ = \begin{bmatrix} 7.030320327\text{-}01 & -6.471307646\text{-}02 & 1.493286880\text{+}00 \\ 5.203275028\text{-}01 & -2.903730234\text{-}01 & -5.949098340\text{-}01 \\ -6.275139883\text{-}01 & 8.483851012\text{-}01 & -1.552037349\text{+}00 \\ 5.241249280\text{-}01 & 1.180262294\text{-}01 & -2.284458009\text{-}02 \end{bmatrix} ,$$

while the subroutine GNVERS gave

$$A^+ = \begin{bmatrix} 7.03032032\text{-}01 & -6.47130762\text{-}02 & 1.49328688\text{+}00 \\ 5.20327502\text{-}01 & -2.90373023\text{-}01 & -5.94909835\text{-}01 \\ -6.27513987\text{-}01 & 8.48385101\text{-}01 & -1.55203735\text{+}00 \\ 5.24124927\text{-}01 & 1.18026230\text{-}01 & -2.28445806\text{-}02 \end{bmatrix} .$$

## PROGRAM LISTING

The main subroutine listing is given below.

```
      SUBROUTINE GNVERS(Y0,T,R,M,N,A,BD,AP)
      DIMENSION A(1),Y0(1),Y1(1),T(1),R(1)

      ITERS=20
      TOL=1.E-07
      MN=M*N
      MSQR=M*M
      ALF=1.6/BD
      DO1 I=1,N
      DO1 J=1,M
      INDEX1=(J-1)*N+I
      INDEX2=(I-1)*M+J
1     Y0(INDEX1)=ALF*A(INDEX2)
      DO2 L=1, ITERS
      CALL FMXMT(M,N,M,A,Y0,R)
      DO3 I=1, MSQR
3     R(I)=-R(I)
      DO4 I=1,M
      INDEX1=(J-1)*M+I
4     R(INDEX1)=1.0+R(INDEX1)
      DO5 I=1, MSQR
5     Y1(I)=R(I)
      DO6 I=1,M
      INDEX1=(I-1)*M+I
6     Y1(INDEX1)=1.0+Y1(INDEX1)
      CALL FMXMT(M,M,M,Y1,R,T)
      DO7 I=1,M
      INDEX1=(I-1)*M+I
7     T(INDEX1)=1.0+T(INDEX1)
      CALL FMXMT(N,M,M,Y0,T,Y1)
      E=0.0
      DO8 I=1,MN
      Y=ABSF(Y1(I)-Y0(I))
      IF(Y.GT.E)9,8
9     E=Y
8     CONTINUE
      1F(E.LT.TOL)99,10
```

```
10  DO2 I=1,MN
 2  Y0(I)=Y1(I)
    PRINT11,ITERS
11  FORMAT(1X,28HGNVERS DOES NOT CONVERGE IN,I3,11H ITERATIONS)
99  CONTINUE
    RETURN
    END
```

The FMXMT listing is

```
    SUBROUTINE FMXMT(M,N,IP,A,B,AB)
    DIMENSION A(1),B(1),AB(1)
    DO1 I=1,M
    DO1 J=1,IP
    L=(J-1)*M+J
    AB(L)=0.0
    DO1 K=1,N
    K1=(K-1)*M+I
    K2=(J-1)*N+K
 1  AB(L)=AB(L)+A(K1)*B(K2)
    RETURN
    END
```

## DERIVATION OF THE ALGORITHM

The Schulz method for inversion of a nonsingular matrix is discussed in Reference 1 and 2. The generalized form of this algorithm, called the hyperpower method of degree m, is defined by

$$\left.\begin{aligned} R_n &= I - AX_n \\ X_{n+1} &= X_n\left(I + R_n + R_n^2 + \dots + R_n^{m-1}\right) \end{aligned}\right\} n = 0, 1, 2, \dots \quad .$$

In Reference 1 it is shown that the hyperpower method of degree 3 is optimum in a certain sense, and convergence bounds are given. References 3, 4, and 5 discuss the hyperpower method of degree 2 (the usual form of Schulz's algorithm) applied to the problem of computing the generalized inverse of a matrix A. Convergence bounds and optimal initialization values are investigated in References 1 and 5. Following the techniques in Reference 4 Lemma 1 and Corollary 1 establish the hyperpower method of degree m for generalized inversion.

Lemma 1: The sequence of matrices defined by

$$\left.\begin{aligned} R_n &= P_{R(A)} - AX_n && (1)\\ X_{n+1} &= X_n\left(P_{R(A)} + R_n + R_n^{\,2} + \dots + R_n^{\,m-1}\right) && (2)\end{aligned}\right\} \quad n = 0, 1, 2, \dots$$

where

$$X_o = A^* B_o \quad (B_o \text{ is some nonsingular } M \times M \text{ matrix}) \quad , \tag{3}$$

$$X_o = C_o A^* \quad (C_o \text{ is some nonsingular } N \times N \text{ matrix}) \quad , \tag{4}$$

$$\| P_{R(A)} - AX_o \| < 1 \quad , \tag{5}$$

and

$$\| P_{R(A^*)} - X_o A \| < 1 \quad . \tag{6}$$

converges to $A^+$ as $n \rightarrow \infty$ .

Proof: Since $A^+$ is the unique solution of the equations

$$AX = P_{R(A)} \tag{7}$$

and

$$XA = P_{R(A^*)} \quad , \tag{8}$$

it sufficies to prove that

$$\lim_{n \rightarrow \infty} \| P_{R(A)} - AX_n \| = 0 \tag{9}$$

$$\lim_{n \rightarrow \infty} \| P_{R(A^*)} - X_n A \| = 0 \tag{10}$$

To do this, consider the sequences of matrices

$$B_{n+1} = B_n\left(P_{R(A)} + P_n + P_n^{\,2} + \dots + P_n^{\,m-1}\right)$$

$$C_{n+1} = C_n\left(P_{R(A)} + Q_n + Q_n^{\,2} + \dots + Q_n^{\,m-1}\right)$$

where

$$P_n = P_{R(A)} - AA^*B_n \quad ,$$

$$Q_n = P_{R(A^*)} - A^*AC_n \quad ,$$

and $(n = 0, 1, 2, \dots)$. Then it is obvious that

$$X_n = A^*B_n \quad . \tag{11}$$

Moreover,

$$X_n = C_nA^* \tag{12}$$

because if we let

$$Y_{n+1} = C_{n+1}A^*$$

and note that

$$Q_n^{\,i}A^* = Q_n^{\,i-1}\left(P_{R(A^*)}A^* - A^*AC_nA^*\right) = Q_n^{\,i-1}A^*\left(P_{R(A)} - AY_n\right)$$

$$= \dots = A^*\left(P_{R(A)} - AY_n\right)^i \quad ,$$

since

$$P_{R(A^*)}A^* = A^*P_{R(A)} \quad ,$$

then

$$Y_{n+1} = Y_n\Big[P_{R(A)} + \left(P_{R(A)} - AY_n\right) + \left(P_{R(A)} - AY_n\right)^2 + \ldots + \left(P_{R(A)} - AY_n\right)^{m-1}\Big]$$

so that $Y_n$ satisfies equation (2). Since $X_o = Y_o$, $X_n = Y_n$ for all n, which establishes equation (12).

Since

$$AX_n = P_{R(A)} AX_n = AX_n P_{R(A)} \quad ,$$

we have by an easy induction that

$$R_n^k = P_{R(A)} - AX_n\left(P_{R(A)} + R_n + R_n^2 + \ldots + R_n^{k-1}\right)$$

$$(k = 0, 1, 2, \ldots) \quad .$$

Setting k = m gives $R_{n+1} = R_n^m$, so that

$$\| P_{R(A)} - AX_{n+1} \| \leq \| P_{R(A)} - AX_n \|^m \quad ,$$

which with equation (5) proves equation (9). The proof of equation (10) is analogous.

<u>Corollary 1:</u> Let

$$\lambda_1(AA^*) \geq \lambda_2(AA^*) \geq \ldots \geq \lambda_r(AA^*)$$

denote the nonzero eigenvalues of $AA^*$. If

$$0 < \alpha < \frac{2}{\lambda_1(AA^*)} \quad , \tag{13}$$

then the sequence defined by

$$Y_o = \alpha A^* \tag{14}$$

$$\left.\begin{aligned} R_n &= I - AY_n \\ Y_{n+1} &= Y_n\left(I + R_n + R_n^{\,2} + \dots + R_n^{\,m-1}\right) \end{aligned}\right\} n = 0, 1, 2, \dots \qquad \begin{aligned}(15)\\(16)\end{aligned}$$

converges to $A'$ as $n \rightarrow \infty$.

Proof: Since

$$P_{R(A)}AA^* = AA^+AA^* = AA^* = AA^*AA^+ = AA^*P_{R(A)} \quad ,$$

$P_{R(A)}$ and $AA^*$ are commuting hermitian matrices with the same range space, so that the eigenvalues of $P_{R(A)} - AY_o$ are

$$\left.\begin{aligned} &1 - \alpha\,\lambda_i(AA^*) && (i = 1, 2, \dots, r) \\ &0 && (i = r+1, \dots, m) \end{aligned}\right\} \quad ; \qquad (17)$$

hence, by equation (13)

$$\left|\lambda_1\left(P_{R(A)} - AY_o\right)\right| < 1 \quad . \qquad (18)$$

Similarly

$$\left|\lambda_1\left(P_{R(A^*)} - Y_oA\right)\right| < 1 \quad .$$

Now the process (2), begun with equation (14), retains the form of equation (12), so

$$\begin{aligned} X_n\left(P_{R(A)} + R_n + \dots + R_n^{\,m-1}\right) &= X_nP_{R(A)} + X_n\left(R_n + \dots + R_n^{\,m-1}\right) \\ &= C_nA^*AA^+ + X_n\left(R_n + \dots + R_n^{\,m-1}\right) \\ &= X_n\left(I + R_n + \dots + R_n^{\,m-1}\right) \end{aligned}$$

Thus, the convergence of equation (16) follows from that of equation (12).

Corollary 2: The process (18) is convergent if $\alpha$ satisfies the inequality

$$0 < \alpha \leq \max_{1 \leq i \leq M} \sum_{j=1}^{M} \left| b_{ij} \right|$$

where $B = \left[ b_{ij} \right]_1^M = AA^*$.

Proof: By Gershgorin's theorem [ 2 ],

$$\lambda_1(AA^*) \leq \max_{1 \leq i \leq M} \sum_{j=1}^{M} \left| b_{ij} \right| .$$

Note that it is immaterial whether we take B as $AA^*$ or $A^*A$ since the nonzero eigenvalues of these two matrices are the same.

## APPLICATION TO REDUNDANT CMG ASSEMBLIES

The instantaneous torque output of a single-gimbal CMG is given by

$$\dot{h} = \dot{\alpha}_1 e_1 \times he_2 = h\dot{\alpha}e$$

where $\dot{\alpha}$ is the instantaneous gimbal rate, h is the CMG momentum magnitude (determined by the wheel speed), and $e_1$ and $e_2$ are unit vectors along the gimbal axis and in the direction of the momentum vector, respectively. Then e is a unit vector along the instantaneous torque vector $\dot{h}$. The torque output of N such CMGs is then

$$\dot{h} = \dot{h}_1 + \dot{h}_2 + \ldots + \dot{h}_N = h_1\dot{\alpha}_1 e_1 + h_2\dot{\alpha}_2 e_2 + \ldots h_N\dot{\alpha}_N e_N$$

where $\dot{\alpha}_i$ denotes the gimbal rate of the ith CMG and $e_i$ now denotes the unit vector in the direction of its instantaneous torque vector. This last equation can be put in the following form:

$$\dot{h} = C\dot{\alpha}$$

where C is the $3 \times N$ matrix whose columns are $h_1e_1$, $h_2e_2$, ..., $h_Ne_N$ and $\dot{\alpha}$ is the column of gimbal rates. Then, we have that

$$B = A^*A = \begin{bmatrix} h_1^2 e_1^* e_1 & h_1h_2e_1^* e_2 & \cdots & h_1h_Ne_1^*e_N \\ h_2h_1 e_2^*e_1 & h_2^2e_2^*e_2 & \cdots & h_2h_Ne_2^*e_N \\ \cdots & \cdots & \cdots & \cdots \\ h_Nh_1e_N^*e_1 & h_Nh_2e_N^*e_2 & \cdots & h_N^2e_N^*e_N \end{bmatrix}$$

so

$$\lambda_1(A^*A) \leq \max_{1 \leq i \leq N} \sum_{j=1}^{N} h_i h_j \left| e_i^* e_j \right| \leq \max_{1 \leq i \leq N} \sum_{j=1}^{N} h_i h_j \leq Nh^2$$

since $|e_i^*e_j| \leq 1$ for each i and j; h is the nominal momentum of the CMGs used.

If $T_c$ is a commanded torque output, we wish to ensure that $\dot{h} = T_c$ by choosing $\dot{\alpha}$; the procedure by which this choice is made is called a steering law. One such law, which is optimum in a certain sense, is

$$\dot{\alpha} = C^+ T_c \quad .$$

While this steering law has exhibited satisfactory performance in a number of simulations, a number of complicating factors must be considered before the law can be considered for implementation. Discussion of these, which have mainly to do with situations in which the columns $e_i$ become coplanar (a hangup condition), is beyond the scope of this report.

## REFERENCES

1. Altman, M.: An Optimum Cubically Convergent Iterative Method of Inverting a Linear Bounded Operator in Hilbert Space. Pacific Jour. Math., vol. 10, 1960, pp. 1107 - 1113.

2. Householder, Alston S.: The Theory of Matrices in Numerical Analysis. Blaisdell, 1964.

3. Ben-Israel, Adi: An Iterative Method for Computing the Generalized Inverse of an Arbitrary Matrix. Math. of Computation, vol. 19, 1965, pp. 452 - 455.

4. Ben-Israel, Adi: A Note on an Iterative Method for Generalized Inversion of Matrices. Math. of Computation, vol. 20, 1966, pp. 439 - 440.

5. Ben-Israel, Adi and Cohen, Dan: On Iterative Computation of Generalized Inverses and Associated Projections. Jour. SIAM Numer. Anal., vol. 3, 1966, pp. 410 - 419.

# APPROVAL

# ITERATIVE COMPUTATION OF GENERALIZED INVERSES, WITH AN APPLICATION TO CMG STEERING LAWS

By J. W. Steincamp

The information in this report has been reviewed for security classification. Review of any information concerning Department of Defense or Atomic Energy Commission programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

This document has also been reviewed and approved for technical accuracy.

David N. Schultz

DAVID N. SCHULTZ
Chief, Navigation and Control Systems Branch

Fred E. Digesu

FRED E. DIGESU
Chief, Electronics and Control Division

ERICH E. GOERNER
Director, Preliminary Design Office

James T. Murphy

JAMES T. MURPHY
Acting Director, Program Development



MSFC—RSA, Ala

# DISTRIBUTION

## INTERNAL

DIR
Dr. Rees
DEP-T
Dr. Lucas
AD-S
Dr. Stuhlinger
PD-DIR
Mr. Murphy
Mr. Downey
Dr. Mrazek
PD-MP-DIR
Mr. Gierow
PD-MP-A
Mr. Olivier
Mr. Emanuel
PD-DO-DIR
Mr. Goerner
Dr. Thomason
Mr. Heyer
Mrs. Andrews
PD-SO-M
Mr. Laue
PD-DO-P
Mr. Goldsby
PD-DO-S
Mr. Marshall
PD-DO-E
Mr. Digesu
Mr. Harden
Mr. Pavlick

PD-DO-EC
Mr. Arsement
PD-DO-EP
Mr. Boehme
PD-DO-ES
Mr. Schultz
Mr. Davis
Mr. Green
Dr. Steincamp (20)
S&E-AERO-DIR
Dr. Geissler
S&E-AERO-DO
Dr. Worley
S&E-AERO-DOI
Mr. Buchanan
S&E-ASTR-DIR
Mr. Moore
S&E-ASTR-A
Dr. Seltzer
Dr. Nurre
Mr. Kennel
S&E-ASTR-SG
Mr. Brooks
S&E-ASTR-SGD
Mr. Chubb
Mr. Williamson
Mr. Polites
S&E-ASTR-SGI
Mr. Davis
Mr. Fisher

S&E-ASTR-SD
Mr. Scofield
S&E-ASTR-DDI
Mr. Farmer
S&E-ASTR-SGA
Mr. Thompson
Mr. Brown
S&E-ASTR-SDA
Mr. Justice
S&E-ASTR-SDI
Mr. Farmer
S&E-CSE-DIR
Dr. Haeussermann
Mr. Mack
PM-HE-MGR
Dr. Speer
PM-HE-I
Mr. Fichtner
Mr. Hammers
Mr. Shields
Mr. Wiesenmaier
PM-HE-S
Mr. McMillan
A&TS-MS-IP
Mr. Ziak (2)
A&TS-MS-I
Mr. Remer
A&TS-MS-IL
Miss Robertson (8)

PM-PR-M
Mr. Stevens
A&TS-MS-H
Mr. Akens
A&TS-MS-D
Mr. Garrett
A&TS-PAT
Mr. Wofford
A&TS-TU
Mr. Wiggins (6)

## EXTERNAL

Scientific and Technical
Information Facility (25)
College Park, Maryland
20740
Attn: NASA Representative
(S-AK/RKT)