AN ABSTRACT OF THE THESIS OF

Crystal L. Hollingsworth for the Master of Science in Mathematics presented on December 1977.

Title: Generalized Inverses Calculations and Relations

Abstract approved: George D. Poole

The mathematical content of this thesis is normally considered numerical linear algebra. It is assumed that the reader has some knowledge of matrix theory. Matrix entries are restricted to the real numbers.

The subject of generalized inverses is introduced with a discussion of the inadequacies of the usual matrix inverse in solving systems of linear equations when rectangular or singular matrices are involved. The various types of generalized inverses (mainly A', A^{+}, A_d, and A_g are then defined and methods of calculation are presented. Computer programs (written in Fortran IV) are made available for the calculation of A', solutions in particular and general form, and A^{+}. Lastly, a look is taken at some additional relations among the different generalized inverses; specifically, the possibility of a relationship between A_d and A^{+}.

GENERALIZED INVERSES

CALCULATIONS AND RELATIONS

A Thesis

Presented to

the Department of Mathematics

EMPORIA STATE UNIVERSITY

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Crystal L. Hollingsworth

December 1977

Approved for the Major Department

Approved for the Graduate Council

PREFACE

The mathematical content of this thesis is normally considered numerical linear algebra. It has been written with the assumption that the reader has some knowledge of matrix theory. Matrix entries are restricted to the real numbers although many theorems and properties are also true if complex numbers are used. A discussion of the inadequacies of the usual matrix inverse in solving systems of linear equations when rectangular or singular matrices are involved introduces the generalized inverse. The various types of generalized inverses are then defined and methods of calculation are presented. A few of the methods are suitable for programming and computer programs are given for those that are. Lastly, a look is taken at some additional relations among the different generalized inverses; specifically, the possibility of a relationship between A_d and A^+.

Table of Contents

Chapter | Page

I. Introductory Material

1.1 Introduction 1

1.2 Some Generalized Inverses 4

1.3 Applications 8

II. Calculations of Generalized Inverses

2.1 Calculate A' 10

2.2 Calculate A^+ 13

2.3 Finding Jordan Form J and P Such That $PAP^{-1} = J$ ... 15

2.4 Calculate A_d 23

2.5 Calculate A_g 25

III. Computer Programs

3.1 Introduction 26

3.2 Program for A' (particular solution) 28

3.3 Program for A' (general solution) 36

3.4 Program for A^+ 44

IV. Relations Between Inverses

4.1 Conditions for A_d And A^+ To Have The Same Index ... 48

V. Conclusion

Bibliography

CHAPTER I: INTRODUCTORY MATERIAL

1.1 INTRODUCTION

Anyone associated with mathematics to some degree has come in contact with the usual matrix inverse; that is, for every nonsingular square matrix A, there exists a unique inverse, A^{-1}, such that $AA^{-1} = A^{-1}A = I$ where I is the identity matrix. No such matrix A^{-1} exists when A is rectangular or singular. (Notice that from now on, nonsingular will mean square and nonsingular.)

A very common application of matrices is in the solution of systems of linear equations. If A is a nonsingular matrix and $A\bar{x} = \bar{b}$ is a system of linear equations ($\bar{x}$ and $\bar{b}$ are column vectors), then the solution is unique and easily found.

$$A\bar{x} = \bar{b}$$

$$A^{-1}A\bar{x} = A^{-1}\bar{b}$$

$$\bar{x} = A^{-1}\bar{b}$$

If the system to be solved is singular or rectangular, A^{-1} can't be used in finding the solution. Some new type of matrix inverse is needed, one that would exist for rectangular or singular matrices as well as nonsingular ones. It should have some of the properties of the usual inverse and reduce to the usual inverse when A is nonsingular. A matrix has been defined that has these qualities, and is useful in solving systems of linear equations. It's called the generalized inverse.

<u>Definition 1.1</u> The generalized inverse of a matrix A is any matrix A' such that $AA'A = A$.

Recall that any matrix A can be written as

$$A = P \begin{bmatrix} B & 0 \\ 0 & N \end{bmatrix} P^{-1}$$

where B and P are nonsingular, 0 denotes a null matrix, and N is nilpotent. $\begin{bmatrix} B & 0 \\ 0 & N \end{bmatrix}$ is a Jordan canonical form for A.

The existence of A' can be seen in that if

$$A = P \begin{bmatrix} B & 0 \\ 0 & N \end{bmatrix} P^{-1} \text{ then}$$

$$A' = P \begin{bmatrix} B^{-1} & 0 \\ 0 & N^T \end{bmatrix} P^{-1}.$$

It is easily seen that if A is nonsingular,

$$AA'A = A$$

$$A^{-1}(AA'A)A^{-1} = A^{-1}AA^{-1}$$

$$A' = A^{-1}$$

In comparing the definitions of A^{-1} and A', observe that A^{-1} is unique whereas A' is not, except in some special cases.

Now in systems of linear equations ($A\bar{x} = \bar{b}$), when A is singular or rectangular there could be no solutions or infinitely many solutions or a unique solution. Consider the system $A\bar{x} = \bar{b}$ and suppose that $\bar{b}$ is a linear combination of the columns of A (consistent). Then A' can be used to obtain a solution for $A\bar{x} = \bar{b}$. For example, let $\bar{b} = A\bar{y}$. Then if $\bar{x} = A'\bar{b}$,

$$A\bar{x} = A(A'\bar{b}) = AA'(A\bar{y}) = A\bar{y} = \bar{b}.$$

The above produces a particular solution. If a general solution is desired, then $A\bar{x} = \bar{b}$ has a solution if and only if $AA'\bar{b} = \bar{b}$. The general solution is then $\bar{x} = A'\bar{b} + (I - A'A)\bar{y}$ where $\bar{y}$ is arbitrary. [2, p. 2-3].

Note: If the above system is inconsistent, other generalized inverses are used to obtain "a solution".

1.2 SOME GENERALIZED INVERSES

In addition to A', there is a variety of generalized inverses which exist. The following equations (where X denotes the generalized inverse) are grouped in various ways to define the different generalized inverses. Here T denotes transpose and could be replaced by the conjugate transposed if complex numbers were considered.

rectangular or square matrices

(1) $AXA=A$

(2) $XAX=X$

(3) $(AX)^T=AX$

(4) $(XA)^T=XA$

square matrices only

(1^k) $A^kXA=A^k$

(5) $AX=XA$

(5^k) $A^kX=XA^k$

(6^k) $AX^k=X^kA$

In the equation for square matrices only the k is a given positive integer, actually the index of A, which will be discussed later in this section along with the Drazin inverse.

Definition 1.2 For any element A of R^{mxn}, let $A^{(i,j,\ldots,l)}$ denote the set of matrices in R^{mxn} which satisfies equations i, j, ..., l above. $A^{(i,j,\ldots,l)}$ is called an $\{i,j,\ldots,l\}$ - inverse of A.

Example 1.1 $\{1,2\}$ - inverse is a generalized inverse that satisfies equations (1) and (2) and is represented by $A^{(1,2)}$.

The notation $\{i,j,\ldots,l\}$ - inverse or $A^{(i,j,\ldots,l)}$ will generally be used for clarification purposes or when the inverse being discussed does not have its own name and special symbol as most inverses here will have. The following definitions and theorems will present the most important generalized inverses.

Definition 1.3 A' is a $\{1\}$ - inverse and exists for each matrix A (as seen in section 1.1).

Theorem 1.1 If A is nonsingular, $A' = A^{-1}$.

Proof: Done in section 1.1.

Definition 1.4 A^+, the Moore-Penrose inverse, is a unique $\{1,2,3,4\}$ - inverse. (The proof that A^+ is unique can be found in [2, p.8] and existence [2, p.21-22].)

Theorem 1.2 If A is nonsingular, $A^+ = A^{-1}$.

Proof: A^+ is a $\{1\}$ - inverse. Hence $A^+ = A^{-1}$.

Definition 1.5 A_d, the Drazin inverse, is a unique $\{1^k,2,5\}$ - inverse of a square matrix A. (The proof for uniqueness and existence can be found in [2, p. 172]. Also if $A=P\begin{bmatrix} B & 0 \\ 0 & N \end{bmatrix}P^{-1}$, then $A_d = P\begin{bmatrix} B^{-1} & 0 \\ 0 & 0 \end{bmatrix}P^{-1}$.)

Lemma 1.1 (2) $A_dAA_d = A_d$ is equivalent to $AA_d{}^2 = A_d$

Proof: Recall (5) $AA_d = A_dA$, then $A_dAA_d = AA_dA_d = AA_d{}^2 = A_d$.

Lemma 1.2 $(1^k)A^kA_dA = A^k$ is equivalent to $A^{k+1}A_d = A^k$

Proof: Using (5), $A^kA_dA = A^kAA_d = A^{k+1}A_d = A^k$.

The two above equivalencies are sometimes used instead of (2) and (1^k) in proofs or derivations.

Definition 1.6 The index of a square matrix A is the smallest nonnegative integer k such that rank A^k = rank A^{k+1} (where $A^\circ = I$).

Theorem 1.3 Let A be an element of R^{nxn}. Then the following are equivalent:

(a) The index of A is k.

(b) k is the smallest positive exponent for which $A^{k+1}A_d = A^k$.

(c) If A is singular and $m(\lambda)$ is its minimum polynomial, k is the multiplicity of $\lambda = 0$ as a zero of $m(\lambda)$.

Proof: [2, p.170-171].

Theorem 1.4 If A is a nonsingular matrix, then $A_d = A^{-1}$.

Proof: Observe that A^{-1} satisfies the definition of A_d and since A_d is unique, $A^{-1} = A_d$.

Definition 1.7 A_g, the group inverse, is unique $\{1,2,5\}$ - inverse. When it does exist, A_g is A_d (the Drazin inverse).

Theorem 1.5 A square matrix A has a group inverse if and only if its index is 1 or 0 or if and only if rank A = rank A^2. When the group inverse exists, it is unique.

Proof: If A has index 0, A is nonsingular and since A_g is a $\{1\}$ - inverse, $A_g = A^{-1}$. When A has index 1, by definition of index, rank A = rank A^2. The index of A cannot be greater than 1 because by definition $A_g = A_d$ when A_g exist. A_d is a $\{1^k,2,5\}$ - inverse whereas A_g is a $\{1,2,5\}$ - inverse. k clearly must be 1 (unless it is 0).

To prove uniqueness, let X and A_g be $\{1,2,5\}$ - inverses of A. Let $E = AX = XA$ and $F = AA_g = A_gA$. Then

$$E = AX = AA_gAX = FE$$

$$F = AgA = A_gAXA = FE$$

Therefore, $F = E$ and $X = EX = FX = A_gE = A_gF = A_g$.

If A_g exists and $A = P \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} P^{-1}$, then $A_g = P \begin{bmatrix} B^{-1} & 0 \\ 0 & 0 \end{bmatrix} P^{-1}$.

Theorem 1.6 If A is nonsingular, $A_g = A^{-1}$.

Proof: Done in proof of Theorem 1.5.

In independent theorems, each generalized inverse (A', A^+, A_d, and A_g) was shown to be the usual matrix inverse when A is nonsingular. And, therefore, the following theorem results.

Theorem 1.7 If A is nonsingular, $A^{-1} = A' = A^+ = A_d = A_g$.

1.3 APPLICATIONS

The generalized inverses (A', A^+, and A_d) just defined in the last section have applications in different areas of applied mathematics. In this section, a list of some of the applications of each inverse will be given along with some possible references.

<u>A'</u>

Estimation from Linear Models: [11]

Interval Linear Programming: [2], [11], [12]

Least Squares: [6], [11]

Minimum norm: [11]

Network Theory: [11]

Solutions of Linear Systems: [2], [6]

<u>A^+</u>

Algebraic Eigenvalue Problem: [3]

Blue's: [1]

Conditional Expectation: [1]

Constrained Least Squares: [1]

Distribution Theorem: [3]

Fixed Point Probability Vector of Regular or Ergodic Transition Matrices: [3]

General Linear Hypothesis: [1]

Incidence Matrices: [3]

Independence of Quadratic Forms in Normal Variates: [3]

Kalman Filtering: [1]

Least Squares and Best Linear Unbiased Estimation: [12]

Linear Programming: [1]

$\underline{A^+}$

Markov Chains: [1]

Minimum Norm and Least Squares: [1], [2], [8], [11]

Multivariate Analysis Discrimination Problems: [12]

Network Theory: [11]

Nonlinear Analysis: [12]

Nonnegative Definitive Matrices: [1]

Optimum Experimental Designs: [12]

Penalty Functions: [1]

Regression Analysis: [12]

Sequential Least Squares and Recursive Estimation: [12]

Sequential Least Squares Parameter Estimation: [3]

Singular Multivariate Normal Distributions and Associated Predictive Problems: [12]

Stepwise Regression: [1]

Stochastic Matrices: [3]

Variance Component Problems: [12]

$\underline{A_d}$

Cesaro-Neumann Iterations: [5]

Linear Systems of Differential Equations: [4]

Spectral Inequality of Marcus, Minc, and Moyls: [9]

CHAPTER II CALCULATIONS OF GENERALIZED INVERSES

2.1 CALCULATE A' (Method from [2, p.8-14])

Definition 2.1 A mxn matrix H with rank r is in Hermite normal form (row echelon form) if:

(a) Each of the first r rows contains at least one nonzero element; the remaining rows contain only zeros.

(b) The first r columns of the identity matrix I_m appear in columns $c_1, c_2, \ldots, c_r$.

H can be put into a partitioned form by a permutation of its columns

$$R = \begin{bmatrix} I_r & K \\ 0 & 0 \end{bmatrix}$$

where 0 is a null matrix. This permutation of columns can be accomplished by multiplying H by a permutation matrix P on the right. If P_j is the j^{th} column of P, e_j the j^{th} column of I_n, and c_j the j^{th} column of H, then $P_j = e_k$ where $k = c_j$ $(j = 1,2, \ldots, r)$. The remaining columns of P are the remaining unit vectors $\{e_k: \ k \neq c_j \ (j = 1, 2, \ldots, r)\}$ in any order.

For any $(n - r)x(m - r)$ matrix L, the nxm matrix

$$S = \begin{bmatrix} I_r & 0 \\ 0 & L \end{bmatrix}$$

is a $\{1\}$ - inverse of R. Now given a mxn matrix A, recall that every elementary row operation can be interpreted as a premultiplication of A by a suitable nonsingular elementary row matrix. Any matrix can be put in Hermite normal form by a finite sequence of elementary row operations. Hence, for any mxn matrix A there is a nonsingular mxm matrix E (the product of the elementary row matrices) and a permutation matrix P (nxn) such that

$$EAP = \begin{bmatrix} I_r & K \\ 0 & 0 \end{bmatrix}$$

Theorem 2.1 Let A be an mxn matrix, E a mxm matrix (product of elementary row matrices), and P a nxn permutation matrix such that

$$EAP = \begin{bmatrix} I_r & K \\ 0 & 0 \end{bmatrix}$$

Then, for any (n - r)x(m - r) matrix L, the nxm matrix

$$X = P \begin{bmatrix} I_r & 0 \\ 0 & L \end{bmatrix} E$$

is a $\{1\}$ - inverse of A.

Proof: From

$$EAP = \begin{bmatrix} I_r & K \\ 0 & 0 \end{bmatrix}, \quad A = E^{-1} \begin{bmatrix} I_r & K \\ 0 & 0 \end{bmatrix} P^{-1}.$$

Now notice that X does satisfy AXA = A.

Example 2.1

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

(a) Transform A into Hermite normal form. Augment ${}_mA_n$ with I_m and then do Gaussian elimination. (Here ${}_mA_n$ denotes an mxn matrix A.)

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \quad \text{(augmented form)}$$

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{array}\right] \quad \text{(after Gaussian elimination)}$$

It is in the partitioned form $\left[EA \,\vdots\, E \right]$ where EA is the Hermite normal form and E is the matrix needed (product of the elementary row operations). Notice that rank A = 2.

Now find P such that

$$EAP = \begin{bmatrix} I_2 & K \\ 0 & 0 \end{bmatrix} .$$

P is $e_1\ e_2\ e_3$ or

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

From Theorem 2.1,

$$X = P \begin{bmatrix} I_2 & 0 \\ 0 & L \end{bmatrix} E$$

where L is a 1x1 matrix. Let $L = [\alpha]$, where α is an element of R.

$$x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\alpha & -\alpha & \alpha \end{bmatrix} .$$

This is a general form for ${}_nX_m$. A particular form would be obtained by replacing α with a particular real number.

Note: A program that calculates a particular A' by this method is found in section 3.1.

2.2 CALCULATE A^+ (An iterative technique from [3, p.74-75])

Let a_k denote the k^{th} column of A, and let A_k denote the matrix consisting of the first k columns. Consider A_k in the partitioned form $[A_{k-1}, a_k]$. Compute $d_k = A^+_{k-1} a_k$ and $c_k = a_k - (A_{k-1} d_k)$. If $c_k \neq 0$, let $b_k = c^+_k$. If $c_k = 0$, compute $b_k = (1 + d^T_k d_k)^{-1} d^T_k A^+_{k-1}$. Then

$$A^+_k = \begin{bmatrix} A_{k-1} - d_k b_k \\ b_k \end{bmatrix} .$$

To initiate the process, take $A^+_1 = 0$ if $a_1 = \overline{0}$; otherwise $A^+_1 = (a^T_1 a_1)^{-1} a^T_1$. The value of K starts at 1 and after A^+_k has been calculated $K = K + 1$.

Example 2.2

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$A^+_1 = (a^T_1 a_1)^{-1} a^T_1 = ([1 \quad 0 \quad 1] \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix})^{-1} [1 \quad 0 \quad 1] = [1/2 \quad 0 \quad 1/2]$$

$$d_2 = A^+_1 a_2 = [1/2 \quad 0 \quad 1/2] \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 1/2$$

$$c_2 = a_2 - A_1 d_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} = \begin{bmatrix} -1/2 \\ 1 \\ 1/2 \end{bmatrix} \neq \overline{0}$$

$$b_2 = c^+_2 = ([-1/2 \quad 1 \quad 1/2] \begin{bmatrix} -1/2 \\ 1 \\ 1/2 \end{bmatrix})^{-1} [-1/2 \quad 1 \quad 1/2] = [-1/3 \quad 2/3 \quad 1/3]$$

$$A_2^+ = \begin{bmatrix} [1/2 \quad 0 \quad 1/2] - 1/2\,[-1/3 \quad 2/3 \quad 1/3] \\ \\ -1/3 \quad 2/3 \quad 1/3 \end{bmatrix} = \begin{bmatrix} 2/3 & -1/3 & 1/3 \\ -1/3 & 2/3 & 1/3 \end{bmatrix}$$

$$d_3 = \begin{bmatrix} 2/3 & -1/3 & 1/3 \\ -1/3 & 2/3 & 1/3 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$c_3 = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \overline{0}$$

$$b_3 = (1 + [1 \quad -1] \begin{bmatrix} 1 \\ -1 \end{bmatrix})^{-1} [1 \quad -1] \begin{bmatrix} 2/3 & -1/3 & 1/3 \\ -1/3 & 2/3 & 1/3 \end{bmatrix} = [1/3 \quad -1/3 \quad 0]$$

$$A_3^+ = \begin{bmatrix} \begin{bmatrix} 2/3 & -1/3 & 1/3 \\ -1/3 & 2/3 & 1/3 \end{bmatrix} - \begin{bmatrix} 1/3 & -1/3 & 0 \\ -1/3 & 1/3 & 0 \end{bmatrix} \\ 1/3 \qquad -1/3 \qquad 0 \end{bmatrix} = \begin{bmatrix} 1/3 & 0 & 1/3 \\ 0 & 1/3 & 1/3 \\ 1/3 & -1/3 & 0 \end{bmatrix} = A^+.$$

Note: There is a program in section 3.3 to calculate A^+ by this method.

2.3 FINDING JORDAN FORM J AND P SUCH THAT $PAP^{-1} = J$

Definition 2.3 Let A be an nxn matrix whose distinct eigenvalues are $\lambda_1, \ldots \lambda_s$ with multiplicities $m_1, \ldots, m_s$. Then J is the Jordan form of A if and only if

$$J = \begin{bmatrix} J_1 & 0 & \ldots & 0 \\ 0 & J_2 & \ldots & 0 \\ \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \ldots & \cdot \\ 0 & 0 & \ldots & J_s \end{bmatrix}$$

where J_i is an $m_i x m_i$ matrix of the form

$$J_i = \begin{bmatrix} \lambda_i & * & 0 & \ldots & 0 & 0 & 0 \\ 0 & \lambda_i & * & \ldots & 0 & 0 & 0 \\ 0 & 0 & \lambda_i & \ldots & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \ldots & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \ldots & 0 & \lambda_i & * \\ 0 & 0 & 0 & \ldots & 0 & 0 & \lambda_i \end{bmatrix}$$

with each * equal to 0 or 1. Some may equal 0 while others equal 1.

If A has distinct eigenvalues, s = n and all $m_j = 1$, then J = diag $(\lambda_1, \ldots, \lambda_n)$. If A has multiple eigenvalues but has n linearly independent eigenvectors, then every * in J is equal to 0. (The number of 0's in J_i depends on the number of linearly independent eigenvectors corresponding to λ_i.)

From the definition, a method for calculating the Jordan form J of a matrix A is readily apparent. Find the eigenvalues $(\lambda_1, \ldots, \lambda_s)$ and multiplicities $(m_1, \ldots, m_s)$ from the characteristic equation. Then calculate the eigenvectors associated with each eigenvalue. The number of independent eigenvectors will determine the number of 0's in each J_i.

Another method of arriving at the Jordan form is as follows:

Definition 2.4 Let A be an mxn matrix and $1 \leq k \leq \min \{m,n\}$, the determinantal divisors of A are $d_k(A)$ = g.c.d. $\{\det A[\alpha/\beta], \alpha \in Q_{k,m}, \beta \in Q_{k,n}\}$ [8, p.10] if rank $A \geq k$, and $d_k(A) = 0$ if rank $A \leq k$ where g.c.d. denotes the greatest common divisor. (In other words $d_k(A)$ is found by taking the g.c.d. of all kxk submatrices of A.) Take $d_0 = 1$. [5, p.300]

Definition 2.5 The invariant factors of A are $q_k(A) = d_k(A)/d_{k-1}(A)$ $k = 1, \ldots, r$ (r = rank A).

$q_k(\lambda I - A)$ will look something like the following: $q_k = (\lambda - \lambda_i)^{ni}(\lambda - \lambda_j)^{nj} \ldots (\lambda - \lambda_t)^{nt}$ where the n's are positive integers. Then $(\lambda - \lambda_i)^{ni}$ is an elementary divisor of $\lambda I - A$. The list of elementary divisors of $\lambda I - A$ is the totality of elementary divisors each counted the number of times it appears among the factorizations of the q_k's.

The *'s in the Jordan form J are determined by whether $(\lambda - \lambda_i)^{mi}$ is found in the elementary divisors raised to a power less than m_i. For each power less, a * = 0. Then the rest of the *'s in J_i are 1.

Definition 2.6 The minimum polynomial of A is q_r where r = rank A when q_k is determined from $\lambda I - A$.

Definition 2.7 A matrix S is the Smith normal form of A if it has 1's on the diagonal except for the last diagonal element which is the minimum polynomial of A. The rest of the elements are 0.

$$S = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ 0 & 0 & \dots & q_r \end{bmatrix}$$

Example 2.3

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \end{bmatrix} \qquad \lambda I - A = \begin{bmatrix} \lambda-1 & 0 & -1 \\ 0 & \lambda-1 & 1 \\ -1 & -1 & \lambda \end{bmatrix}$$

$$|\lambda I - A| = \lambda(\lambda - 1)^2$$

Characteristic equation: $\lambda(\lambda - 1)^2 = 0.$

Eigenvalues: $\lambda_1 = 0 \quad \lambda_2 = 1$

Multiplicities: $m_1 = 1 \quad m_2 = 2$

$$(\lambda_1 I - A)\bar{c} = \begin{bmatrix} -1 & 0 & -1 \\ 0 & -1 & 1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \begin{aligned} -c_1 + 0c_2 - c_3 &= 0 \\ 0c_1 - c_2 + c_3 &= 0 \\ -c_1 - c_2 + 0c_3 &= 0 \end{aligned}$$

Eigenvector for $\lambda_1 = 0$: $\begin{bmatrix} \alpha \\ -\alpha \\ -\alpha \end{bmatrix} \quad \alpha \in R$

$$(\lambda_2 I - A)\bar{c} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 1 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \begin{aligned} 0c_1 + 0c_2 - c_3 &= 0 \\ 0c_1 + 0c_2 + c_3 &= 0 \\ -c_1 - c_2 + c_3 &= 0 \end{aligned}$$

Eigenvector for $\lambda_2 = 1$: $\begin{bmatrix} \beta \\ -\beta \\ 0 \end{bmatrix} \quad \beta \in R$

$$J(A) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$ (Any 2 eigenvectors of λ_2 are dependent.)

By the second method,

$d_0 = 1 \quad d_1 = 1 \quad d_2 = 1 \quad d_3 = \lambda(\lambda - 1)^2$

$q_1 = 1 \quad q_2 = 1 \quad q_3 = \lambda(\lambda - 1)^2$

$e_1 = \lambda \quad e_2 = (\lambda - 1)^2$

The power of $(\lambda - 1)$ is not reduced in e_2 so the * = 1.

$$J(A) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Minimum polynomial: $q_3 = m(\lambda) = \lambda(\lambda - 1)^2$

$$S(A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \lambda(\lambda - 1)^2 \end{bmatrix}$$

Now that methods have been presented for finding the Jordan form J of A, we need to be able to find a matrix P such that $PAP^{-1} = J$ since P is used in the calculation of A_d and A_g.

Given any square matrix A, there exists a nonsingular matrix P such that $A = P^{-1}JP$ and $PAP^{-1} = J$. [3, p.20-21]

The following is a method of finding P [7, p.300-345]:

(a) Find matrix Q_1 by starting with $\lambda I-A$ and using elementary row and column operations, keeping track of column operations only, to get the Smith normal form of A. Q_1 is the product of matrices representing column operations (matrices are multiplied in order used).

(b) Find Q_1^{-1}. (Gaussian elimination)

(c) Find the matrix Q_2. Start with $\lambda I-J$ and go through same procedure as in (a) to get Smith normal form of A.

(d) Find matrix $Q = Q_2 Q_1^{-1}$. (Q is a matrix whose elements are polynomials in λ.)

(e) Write $Q(\lambda)$ over matrices. (A is an nxn matrix. Then looking at each element in Q, write down the coefficients of λ^n in another nxn matrix. This matrix is multiplied on the right by λ^n. Now add the product of the matrix generated when λ^{n-1} is considered and λ^{n-1}. Continue until the matrix for λ is generated and multiplied by λ, then add on the constant matrix.)

(f) Express P as a polynomial in A, Q(A).

(g) Do the operations (matrix multiplication and addition) to obtain P.

(h) Find P^{-1}.

(i) Can perform check to see if $PAP^{-1} = J$.

<u>Example 2.3</u> (continued)

$$J(A) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(a) Find Q_1: $\begin{bmatrix} \lambda-1 & 0 & -1 \\ 0 & \lambda-1 & 1 \\ -1 & -1 & \lambda \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & -\lambda \\ 0 & -1 & 1 \\ \lambda-1 & 0 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & -\lambda \\ 0 & \lambda-1 & 1 \\ 0 & -(\lambda-1) & \lambda(\lambda-1)-1 \end{bmatrix}$

$$\rightarrow \begin{bmatrix} 1 & 1 & -\lambda \\ 0 & \lambda-1 & 1 \\ 0 & 0 & \lambda(\lambda-1) \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\lambda & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1+\lambda^2 & -\lambda \\ 0 & -1 & 1 \\ 0 & -\lambda^2(\lambda-1) & \lambda(\lambda-1) \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 1 & 1+\lambda^2 & -\lambda \\ 0 & 1 & -1 \\ 0 & -\lambda^2(\lambda-1) & \lambda(\lambda-1) \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & -\lambda+1+\lambda^2 \\ 0 & 1 & -1 \\ 0 & 0 & -\lambda(\lambda-1)^2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & \lambda^2-\lambda+1 \\ 0 & 1 & -1 \\ 0 & 0 & \lambda(\lambda-1)^2 \end{bmatrix}$$

$$\bullet \begin{bmatrix} 1 & 0 & -\lambda^2+\lambda-1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \lambda(\lambda-1)^2 \end{bmatrix}$$

$$Q_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\lambda & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\lambda^2+\lambda-1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\lambda^2+\lambda-1 \\ 0 & 1 & 1 \\ 0 & -\lambda & 1-\lambda \end{bmatrix}$$

(b) Find Q_1^{-1}. (Gaussian elimination)

$$Q_1^{-1} = \begin{bmatrix} 1 & \lambda^3-\lambda^2+\lambda & \lambda^2-\lambda+1 \\ 0 & 1-\lambda & -1 \\ 0 & \lambda & 1 \end{bmatrix}$$

(c) Find Q_2:

$$\begin{bmatrix} \lambda-1 & -1 & 0 \\ 0 & \lambda-1 & 0 \\ 0 & 0 & \lambda \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} -1 & \lambda-1 & 0 \\ \lambda-1 & 0 & 0 \\ 0 & 0 & \lambda \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1-\lambda & 0 \\ \lambda-1 & 0 & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 1 & 1-\lambda & 0 \\ \lambda & 1-\lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1-\lambda & 0 \\ 0 & (\lambda-1)^2 & 0 \\ 0 & 0 & \lambda \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1-\lambda & 0 \\ 0 & 0 & \lambda \\ 0 & (\lambda-1)^2 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 1 & 0 & 1-\lambda \\ 0 & \lambda & 0 \\ 0 & 0 & (\lambda-1)^2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & \lambda-1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & (\lambda-1)^2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & \lambda & \lambda^2-2\lambda+1 \\ 0 & 0 & (\lambda-1)^2 \end{bmatrix}$$

$$\cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2-\lambda \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & (\lambda-1)^2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \lambda \\ 0 & (\lambda-1)^2 & 0 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \lambda \\ 0 & 0 & -\lambda(\lambda-1)^2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\lambda \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & (\lambda-1)^2 \end{bmatrix}$$

$$Q_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & \lambda-1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2-\lambda \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\cdot\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & -\lambda\\ 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}0 & 1 & -\lambda\\ 1 & \lambda-1 & \lambda-\lambda^2\\ 0 & 2-\lambda & (\lambda-1)^2\end{bmatrix}$$

(d) $Q = Q_2\, Q_1^{-1} = \begin{bmatrix}0 & 1 & -\lambda\\ 1 & \lambda-1 & \lambda-\lambda^2\\ 0 & 2-\lambda & (\lambda^2-2\lambda+1)\end{bmatrix} \begin{bmatrix}1 & \lambda^3-\lambda^2+\lambda & \lambda^2-\lambda+1\\ 0 & 1-\lambda & -1\\ 0 & \lambda & 1\end{bmatrix}$

$$= \begin{bmatrix}0 & -\lambda^2-\lambda+1 & -1-\lambda\\ 1 & -\lambda^2+3\lambda-1 & -\lambda+2\\ 0 & \lambda^3-\lambda^2-2\lambda+2 & \lambda^2-\lambda-1\end{bmatrix}$$

(e) $Q(\lambda) = \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\lambda^3 + \begin{bmatrix}0 & -1 & 0\\ 0 & -1 & 0\\ 0 & -1 & 1\end{bmatrix}\lambda^2 + \begin{bmatrix}0 & -1 & -1\\ 0 & 3 & -1\\ 0 & -2 & -1\end{bmatrix}\lambda + \begin{bmatrix}0 & 1 & -1\\ 1 & -1 & 2\\ 0 & 2 & -1\end{bmatrix}$

(f) Find Q(A):

$A^3 = \begin{bmatrix}3 & 2 & 1\\ -2 & -1 & -1\\ 1 & 1 & 0\end{bmatrix} \qquad A^2 = \begin{bmatrix}2 & 1 & 1\\ -1 & 0 & -1\\ 1 & 1 & 0\end{bmatrix} \qquad A = \begin{bmatrix}1 & 0 & 1\\ 0 & 1 & -1\\ 1 & 1 & 0\end{bmatrix}$

$$\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}A^3 + \begin{bmatrix}0 & -1 & 0\\ 0 & -1 & 0\\ 0 & -1 & 1\end{bmatrix}A^2 + \begin{bmatrix}0 & -1 & -1\\ 0 & 3 & -1\\ 0 & -2 & -1\end{bmatrix}A + \begin{bmatrix}0 & 1 & -1\\ 1 & -1 & 2\\ 0 & 2 & -1\end{bmatrix}$$

(g) $P = \begin{bmatrix}0 & -1 & 1\\ 1 & 1 & 0\\ -1 & -1 & 1\end{bmatrix}$

(h) Find P^{-1}. (Gaussian elimination)

$$P^{-1} = \begin{bmatrix}1 & 0 & -1\\ -1 & 1 & 1\\ 0 & 1 & 1\end{bmatrix}$$

(i) Check:

$$PAP^{-1} = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = J.$$

2.4 CALCULATE A_d

This is the first of two methods presented for the calculation of A_d. This method uses J, P, and P^{-1} calculated in the last section.

If $A = P^{-1}JP = P^{-1}\begin{bmatrix} B & 0 \\ 0 & N \end{bmatrix} P$

where B is a nonsingular matrix, then $A_d = P^{-1}\begin{bmatrix} B^{-1} & 0 \\ 0 & 0 \end{bmatrix} P$ [9, p.504].

Example 2.4

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \end{bmatrix} \quad J = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{so } B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \text{ and } B^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

$$A_d = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 1 \\ 1 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -2 & 1 \\ 2 & 3 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

For more details on this second method see reference [10,p.504-506].

Given a square matrix A, find the minimum polynomial of A, $m(\lambda) = c\lambda^i(\lambda-\lambda_1)^{i1} \ldots (\lambda-\lambda_n)^{in}$. Then the minimum polynomial of B (in J(A)) is $t(\lambda) = (\lambda-\lambda_1)^{i1} \ldots (\lambda-\lambda_n)^{in}$. From this find $g(\lambda)$ such that $g(B) = B^{-1}$. To do this, set $t(B) = 0$ and solve for B^{-1}. B^{-1} is a polynomial in B. So let $g(B) = B^{-1}$. $g(\lambda)$ is found by replacing B with λ and I with 1. The index k of A can be found from $m(\lambda)$, the minimum polynomial, as previously described. Now having k, set $f(\lambda) = \lambda^k g^{k+1}(\lambda)$ and $f(A) = A_d$.

Example 2.4 (continued)

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

$m(\lambda) = \lambda(\lambda-1)^2$ index = 1

$t(\lambda) = (\lambda-1)^2 = \lambda^2 - 2\lambda+1$

$t(B) = B^2 - 2B + I = 0$

$B^2 - 2B = -I$

$B^{-1}(B^2 - 2B) = -B^{-1}$

$B - 2I = -B^{-1}$

$-B + 2I = B^{-1}$

$g(\lambda) = -\lambda + 2$

$f(\lambda) = \lambda g^2(\lambda)$

$f(A) = A \cdot g^2(A) = A(-A + 2I)^2$

$$f(A) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \end{bmatrix} \left(\begin{bmatrix} -1 & 0 & -1 \\ 0 & -1 & 1 \\ -1 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \right)^2 = \begin{bmatrix} -1 & -2 & 1 \\ 2 & 3 & -1 \\ 1 & 1 & 0 \end{bmatrix} = A_d$$

2.5 CALCULATE A_g

Recall that A_g is the Drazin inverse, A_d, when it exists. Therefore, the two methods presented in the previous section for calculating A_d will also calculate A_g if A has index 1.

Note: $A_g = P^{-1}J^{+}P$ [2, p.164].

But $J = \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix}$ and B is nonsingular so $B^{-1} = B^{+}$, and $A_g = P^{-1}J^{+}P =$

$P^{-1} \begin{bmatrix} B^{-1} & 0 \\ 0 & 0 \end{bmatrix} P.$

This is the same as one of the methods already presented.

Example 2.5

Since $A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \end{bmatrix}$ has index 1, A_g does exist and

$$A_g = A_d = \begin{bmatrix} -1 & -2 & 1 \\ 2 & 3 & -1 \\ 1 & 1 & 0 \end{bmatrix}.$$

CHAPTER III COMPUTER PROGRAMS

3.1 INTRODUCTION

Many of the methods for calculating the different generalized inverses are rather long and involved. And certainly it would help if computer programs could be written to compute either some or all of these generalized inverses. Not all render themselves easily to programming, but A' and A^+ do. Two programs to compute A' are included. One computes a particular solution while the other gives the means to compute a general solution. A program to compute A^+ is also included.

In all three programs the data is entered in much the same manner. Each program will handle a number of matrices all in the same run. The exact number depends on the amount of time allowed on the computer. In each run, all matrices must be of the same size. If the matrices you wish to enter are of bigger dimension than the program allows, simply change the dimension statement. The only statements that need to be changed on a regular basis are the format statements.

The first card in a data deck should be one listing the size of the matrices to be entered, row then column, in 2I2 format. (Program 3.3 is an exception as it deals with square matrices. Only one number indicating size need be listed in I2 format.) The second card will give the number of matrices in I3 format. The rest of the cards will be the matrices you wish to enter listed by row. The format used is F5.0 with the appropriate number in front (depending on the number of elements in a row), ie. 3F5.0 or 4F5.0.

The programs are written in Fortran IV (with modifications) and contain many comment statements to aid in following the flow of the program.

3.2 PROGRAM FOR A' (a particular solution)

The following program uses the method of computation presented in section 2.2 to calculate a particular A'. It yields only a particular solution because the computer can't do calculations (addition, subtraction, multiplication, etc.) with variables. They must have a numeric value. Therefore, set any variables in the matrix containing L equal to 10.123. This number could very easily be changed and if more than one variable is present, each could be assigned a different value with minor changes in the program.

As far as output, each matrix A is printed along with the matrices E, P, matrix containing L, and A'. The format used in the examples is E16.8 (again with the appropriate number in front).

For the matrices entered in this program, no machine error has resulted. But if it should occur, it can most likely be handled as in programs 3.3 and 3.4.

```
C COMPUTING A 1-INVERSE
C X SUCH THAT AXA=A
      DIMENSION A(10,10),B(10),C(10),P(10,10),D(10,10),F(10,10),
     XX(10,10)
C READ THE SIZE OF THE MATRIX A
      READ (5,2)M,N
    2 FORMAT (2I2)
C READ THE NUMBER OF MATRICES
C MATRICES MUST BE THE SAME SIZE
      READ (5,1)MSZ
    1 FORMAT (I3)
      DO 70 II=1,MSZ
C READ THE MATRIX A
      DO 6 I=1,M
    6 READ (5,4)(A(I,J),J=1,N)
    4 FORMAT (4F5.0)
      WRITE (6,55)
   55 FORMAT ('1',5X,'MATRIX A',/)
      WRITE (6,58)(A(I,J),J=1,N),I=1,M)
   58 FORMAT (5X,4E16.8)
C TACK ON THE IDENTITY MATRIX
      NP1=N+1
```

```
      NPM=N+M
      DO 8 I=1,M
      DO 8 J=NP1,NPM
      IF (I+N.EQ.J)GO TO 7
      A(I,J)=0.0
      GO TO 8
    7 A(I,J)=1.0
    8 CONTINUE
C MOVE THE IDENTITY MATRIX INTO P
      DO 10 I=1,N
      DO 10 J=1,N
      IF (I.EQ.J)GO TO 9
      P(I,J)=1.0
      GO TO 10
    9 P(I,J)=1.0
   10 CONTINUE
C FIND THE HERMITE NORMAL FORM OF A
      IR=0
      MN=M
      IF (M.GT.N)MN=N
      DO 12 I=1,MN
   11 IF (A(I,I).EQ.0.0)GO TO 24
C HAVE PIVOT ELEMENT
   13  IR=IR+1
      IF (A(I,I).EQ.1.0)GO TO 15
      R=A(I,I)
      DO 14 K=1,NPM
   14 A(I,K)=A(I,K)/R
   15 DO 17 L=1,M
      IF (L.EQ.I)GO TO 17
      IF (A(L,I).EQ.0.0)GO TO 17
      S=A(L,I)
      DO 16 K=I,NPM
   16 A(L,K)=A(L,K)-S*A(I,K)
   17 CONTINUE
      GO TO 12
C FIND PIVOT ELEMENT
   24 IF (I.EQ.MN)GO TO 12
C IS THERE A NONZERO ELEMENT IN COLUMN I
      IP1=I+1
      DO 26 K=IP1,M
      IF (A(K,I).NE.0.0)GO TO 28
   26 CONTINUE
C COLUMN I IS 0
      KK=I
      KKP1=I+1
   25 DO 27 L=1,M
      C(L)=A(L,KK)
      A(L,KK)=A(L,KKP1)
      A(L,KKP1)=C(L)
   27 CONTINUE
      IF (KKP1.EQ.N)GO TO 20
      KK=KK+1
      KKP1=KKP1+1
      GO TO 25
```

```
C INTERCHANGE COL OF P
   20 KK=I
      KKP1=I+1
   19 DO 18 L=1,N
      C(L)=P(L,KK)
      P(L,KK)=(PL,KKP1)
      P(L,KKP1)=C(L)
   18 CONTINUE
      IF (KKP1.EQ.N)GO TO 11
      KK=KK+1
      KKP1=KKP1+1
      GO TO 19
C INTERCHANGE ROWS
   28 DO 30 L=1,NPM
   30 B(L)=A(I,L)
      DO 32 L=1,NPM
   32 A(I,L)=A(K,L)
      DO 34 L=1,NPM
   34 A(K,L)=B(L)
      GO TO 13
   12 CONTINUE
C GENERATE MATRIX CONTAINING L
      DO 36 I=1,N
      DO 36 J=1,M
      IF (I.GT.IR)GO TO 42
      IF (J.GT.IR)GO TO 40
      IF (I.EQ.J)GO TO 38
      D(I,J)=0.0
      GO TO 36
   38 D(I,J)=1.0
      GO TO 36
   40 D(I,J)=0.0
      GO TO 36
   42 IF (J.GT.IR)GO TO 44
      D(I,J)=0.0
      GO TO 36
   44 D(I,J)=10.123
   36 CONTINUE
C MULTIPLY P,L,E
C CLEAR F
      DO 50 I=1,N
      DO 50 J=1,N
   50 F(I,J)=0.0
C MULTIPLY P*L=F
      DO 52 I=1,N
      DO 52 J=1,M
      DO 52 K=1,N
   52 F(I,J)=F(I,J)+P(I,K)*D(K,J)
C MULTIPLY F*E=X
      DO 54 I=1,N
      DO 54 J=1,M
   54 X(I,J)=0.0
      DO 56 I=1,N
      DO 56 J=1,M
```

```
      DO 56 K=1,M
   56 X(I,J)=X(,J)+F(I,K)*A(K,N+J)
      WRITE (6,57)
   57 FORMAT (///,5X,'MATRIX E',/)
      WRITE (6,71)((A(I,J),J=NP1,NPM),I=1,M)
   71 FORMAT (5X,3E16.8)
      WRITE (6,59)
   59 FORMAT (///,5X,'MATRIX P',/)
      WRITE (6,58)((P(I,J),J=1,N),I=1,N)
      WRITE (6,60)
   60 FORMAT (///,5X,'MATRIX CONTAINING L',/)
      WRITE (6,71)((D(I,J),J=1,M),I=1,N)
      WRITE (6,61)
   61 FORMAT (///,5X,'MATRIX X',/)
      WRITE (6,71)((XI,J),J=1,M),I=1,N)
   70 CONTINUE
      STOP
      END
```

Example 3.2 (some output)

First Run (new page for print out)

```
     MATRIX A

      0.10000000E 01  0.0             0.10000000E 01
      0.0             0.10000000E 01 -0.10000000E 01
      0.10000000E 01  0.10000000E 01  0.0


     MATRIX E

      0.10000000E 01  0.0             0.0
      0.0             0.10000000E 01  0.0
     -0.10000000E 01 -0.10000000E 01  0.10000000E 01


     MATRIX P

      0.10000000E 01  0.0             0.0
      0.0             0.10000000E 01  0.0
      0.0             0.0             0.10000000E 01


     MATRIX CONTAINING L

      0.10000000E 01  0.0             0.0
      0.0             0.10000000E 01  0.0
      0.0             0.0             0.10122999E 02
```

```
     MATRIX X

      0.10000000E 01  0.0             0.0
      0.0             0.10000000E 01  0.0
     -0.10122999E 02 -0.10122999E 02  0.10122999E 02

(new page)

     MATRIX A

      0.10000000E 01  0.0             0.10000000E 01
      0.0             0.0             0.0
      0.0             0.10000000E 01 -0.20000000E 01

     MATRIX E

      0.10000000E 01  0.0             0.0
      0.0             0.0             0.10000000E 01
      0.0             0.10000000E 01  0.0

     MATRIX P

      0.10000000E 01  0.0             0.0
      0.0             0.10000000E 01  0.0
      0.0             0.0             0.10000000E 01

     MATRIX CONTAINING L

      0.10000000E 01  0.0             0.0
      0.0             0.10000000E 01  0.0
      0.0             0.0             0.10122999E 02

     MATRIX X

      0.10000000E 01  0.0             0.0
      0.0             0.0             0.10000000E 01
      0.0             0.10122999E 02  0.0

(new page)

     MATRIX A

      0.60000000E 01  0.20000000E 01 -0.20000000E 01
     -0.20000000E 01  0.20000000E 01  0.20000000E 01
      0.20000000E 01  0.20000000E 01  0.20000000E 01
```

MATRIX E

-0.59604645E-07	-0.25000000E 00	0.25000006E 00
0.25000000E 00	0.50000006E 00	-0.25000012E 00
-0.25000006E 00	-0.25000006E 00	0.50000024E 00

MATRIX P

0.10000000E 01	0.0	0.0
0.0	0.10000000E 01	0.0
0.0	0.0	0.10000000E 01

MATRIX CONTAINING L

0.10000000E 01	0.0	0.0
0.0	0.10000000E 01	0.0
0.0	0.0	0.10000000E 01

MATRIX X

-0.59604645E-07	-0.25000000E 00	0.25000006E 00
0.25000000E 00	0.50000006E 00	-0.25000012E 00
-0.25000006E 00	-0.25000006E 00	0.50000024E 00

(Any number as close to being 0.0 as -0.59604645E-07, can be considered 0. 0.)

Second Run (new page)

MATRIX A

0.10000000E 01	0.0	0.10000000E 01	0.10000000E 01
0.0	0.10000000E 01	-0.10000000E 01	0.0
0.10000000E 01	0.10000000E 01	0.0	0.10000000E 01

MATRIX E

0.10000000E 01	0.0	0.0
0.0	0.10000000E 01	0.0
-0.10000000E 01	-0.10000000E 01	0.10000000E 01

MATRIX P

0.10000000E 01	0.0	0.0	0.0
0.0	0.10000000E 01	0.0	0.0

0.0	0.0	0.10000000E 01	0.0
0.0	0.0	0.0	0.10000000E 01

MATRIX CONTAINING L

0.10000000E 01	0.0	0.0
0.0	0.10000000E 01	0.0
0.0	0.0	0.10122999E 02
0.0	0.0	0.10122999E 02

MATIX X

0.10000000E 01	0.0	0.0
0.0	0.10000000E 01	0.0
-0.10122999E 02	-0.10122999E 02	0.10122999E 02
-0.10122999E 02	-0.10122999E 02	0.10122999E 02

(new page)

MATRIX A

0.10000000E 01	0.0	0.0	0.20000000E 01
0.30000000E 01	0.0	-0.10000000E 01	0.0
-0.40000000E 01	0.0	0.20000000E 01	0.0

MATRIX E

0.0	0.10000000E 01	0.50000000E 00
0.0	0.20000000E 01	0.15000000E 01
0.50000000E 00	-0.50000000E 00	-0.25000000E 00

MATRIX P

0.10000000E 01	0.0	0.0	0.0
0.0	0.0	0.0	0.10000000E 01
010	0.10000000E 01	0.0	0.0
0.0	0.0	0.10000000E 01	0.0

MATRIX CONTAINING L

0.10000000E 01	0.0	0.0
0.0	0.10000000E 01	0.0
0.0	0.0	0.10000000E 01
0.0	0.0	0.0

MATRIX X

0.0	0.10000000E 01	0.50000000E 00
0.0	0.0	0.0
0.0	0.20000000E 01	0.15000000E 01
0.50000000E 00	-0.50000000E 00	-0.25000000E 00

3.3 PROGRAM FOR A' (general solution)

Since program 3.2 didn't produce the general form A', another method was needed. Suppose A is square. Recall that A' satisfies AA'A=A. Write A and A' using variables as the elements. Let's look at the specific case when A is a 3x3 matrix.

Let $A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$ and $A' = \begin{bmatrix} x & y & z \\ r & s & t \\ u & v & w \end{bmatrix}$. Then AA'A=A would look like

$$\begin{bmatrix} ax+br+cu & ay+bs+cv & az+bt+cw \\ dx+er+fu & dy+es+fv & dz+et+fw \\ gx+hr+iu & gy+hs+iv & gz+ht+iw \end{bmatrix} \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}.$$

Write the result of the above multiplication with respect to the variables in A' by columns and augment with the column of constants, k (elements of A' in row order).

x	r	u	y	s	v	z	t	w	k
a^2	ab	ac	da	db	dc	ga	gb	gc	a
ba	b^2	bc	ea	eb	ec	ha	hb	hc	b
ca	cb	c^2	fa	fb	fc	ia	ib	ic	c
ad	ae	af	d^2	de	df	gd	de	df	d
bd	be	bf	ed	e^2	ef	hd	he	hf	e
cd	ce	cf	fd	fe	f^2	id	ie	if	f
ag	ah	ai	dg	dh	di	g^2	gh	gi	g
bg	bh	bi	eg	eh	ei	hg	h^2	hi	h
cg	ch	ci	fg	fh	fi	ig	ih	i^2	i

The resulting 9 X 10 matrix represents a 9 X 9 system of linear equations in x,r,u,...w.

Observe from the table above that a pattern is present and is as follows:

(a) To generate the first row, take the first element of the first column of A times each element in the first row of A individually (will have items under x,r, and u). Then take the second element of the first column times the first row (will have items under y,s, and v). Do the same with the third element of the first column and tack on the constant to get the entire first row of the generated matrix.

(b) To get the second, take the second column of A times the first row of A as done in (a).

(c) For the third row, third column and first row.

(d) For the fourth row of the generated matrix, take the first column of A times the second row as before.

(e) Fifth row, second column and second row.

(f) Sixth row, third column and second row.

(g) Seventh row, first column and third row.

(h) etc.

If this generated system of equations is solved by Gaussian elimination, the resulting matrix can be used to obtain the elements of A' in general form. Program 3.3 takes a given matrix A, generates the needed system, and then uses Gaussian elimination to reduce it to row echelon form. The output consists of the original matrix, the generated matrix, and the generated matrix after Gaussian elimination (printed in 1P3E12.4 format, the 3 being replaced by an appropriate number depending on the number of elements in a row). From this information the general form of A' can be easily calculated.

The program is written to do 5x5 matrices and smaller but the print-out for 4x4 and 5x5's must be broken up as they generate matrices too large (16x17 and 25x26) to be printed in one piece. This is taken care of in the program itself and by a set of inserts. If A is a 2x2 or 3x3, certain inserts are put in to write the generated matrix before and after Gaussian elimination. If A is a 4x4 or 5x5, two different inserts are put in.

A problem in machine error did occur in this program. When a number should have been, say, an integer, the computer was producing the integer part right but was somehow placing a digit in the 6th or 7th decimal place. (Example: 1.000004 instead of 1.000000) When Gaussian elimination took place, instead of obtaining a zero from subtraction, the machine would get something like 4×10^{-6}. It would then try to use this as a pivot element causing quite a bit of error in the resulting matrix. This problem was handled by putting in checks for closeness to zero at various points in the program. If the absolute value of an element of the generated matrix (during Gaussian elimination) was less than or equal to .000001, that number was set equal to zero.

The following is the program with the inserts for a 3x3 (or less) matrix in. The inserts for 4x4 matrices or bigger will be given and it will be pointed out where to put them.

```
C COMPUTING THE GENERAL FORM OF A 1-INVERSE OF A SQUARE MATRIX A
C BY USING GAUSSIAN ELIMINATION ON THE SYSTEM OF EQUATIONS GENERATED BY
C AXA=A
      DIMENSION A(5,5),B(25,26),BB(26)
C READ THE NUMBER OF MATRICES (MUST BE SAME SIZE)
      READ (5,2)N
```

```
    1 FORMAT (I3)
      NNM=1
C READ IN SIZE OF MATRIX A
      READ (5,2)N
    2 FORMAT (I2)
C READ IN MATRIX A
    7 DO 3 I=1,N
    3 READ (5,4)(A(I,J),J=1,N)
    4 FORMAT (3F5.0)
C WRITE MATRIX A
      WRITE (6,6)
    6 FORMAT (///,5X,'MATRIX A',/)
      WRITE (6,8)((A(I,J),J=1,N),I=1,N)
    8 FORMAT (5X,1P3E12.4)
C GENERATE MATRIX B (ALL EXCEPT LAST COL.)
      NN=N*N
      II=1
      M=0
      DO 14 I=1,NN
      J=1
      M=M+1
      IF (M.EQ.N+1)M=1
      IF (I.EQ.N*II+1)II=II+1
      DO 12 K=1,N
      DO 10 L=1,N
      B(I,J)=A(K,M)*A(II,L)
      J=J+L
   10 CONTINUE
   12 CONTINUE
   14 CONTINUE
C TACK ON LAST COLUMN
      NNP1=NN+1
      I=1
      DO 15 J=1,N
      DO 13 K=1,N
      B(I,NNP1)=A(J,K)
      I=I+1
   13 CONTINUE
   15 CONTINUE
C WRITE MATRIX B
      WRITE (6,16)
   16 FORMAT (///,5X,1P10E12.4)
C VARIABLES ARE IN COLUMN ORDER X11,X21,...,X12,X22,...,X13,X23,...
C INSERT FOR 3x3 AND LESS
      WRITE (6,18)((B(I,J),J=1,NNP1),I=1,NN)
   18 FORMAT (5X,1P10E12.4)
C END OF INSERT
C DO GAUSSIAN ELIMINATION ON MATRIX B
      I=1
      DO 20 J=1,NN
      DO 100 K=1,NN
      DO 100 L=1,NNP1
      IF (ABS(B(K,L)).LE..000001)B(K,L)=0.
```

```
  100 CONTINUE
   21 IF (B(I,J).EQ.0.0)GO TO 28
C HAVE PIVOT ELEMENT
      IF (B(I,J).EQ.1.0)GO TO 23
C DIVIDE ROW I BY B(I,J)
      R=B(I,J)
      DO 22 K=1,NNP1
   22 B(I,K)=B(I,K)/R
C GET ZEROS IN COLUMN J
   23 DO 24 L=1,NN
      IF (L.EQ.I)GO TO 24
      IF (B(L,J).EQ.0.0)GO TO 24
      S=B(L,J)
      DO 26 K=I,NNP1
   26 B(L,K)=B(L,K)-S*B(I,K)
   24 CONTINUE
      I=I+1
      GO TO 20
C FIND PIVOT ELEMENT
   28 IF (I.EQ.NN)GO TO 20
C IS THERE A NONZERO ELEMENT IN COLUMN J
      IP1=I+1
      DO 30 K=IP1,NN
      IF (B(K,J).NE.0.0)GO TO 32
   30 CONTINUE
C NO PIVOT ELEMENT
      GO TO 20
C INTERCHANGE ROWS
   32 DO 34 L=1,NNP1
   34 BB(L)=B(I,L)
      DO 36 L=1,NNP1
   36 B(I,L)=B(K,L)
      DO 38 L=1,NNP1
   38 B(K,L)=BB(L)
      GO TO 21
   20 CONTINUE
C WRITE MATRIX AFTER GAUSSIAN ELIMINATION
      DO 101 I=1,NN
      DO 101 J=1,NNP1
      IF (ABS(B(I,J)).LE..000001)B(I,J)=0.
  101 CONTINUE
      WRITE (6,40)
   40 FORMAT (///,5X,'MATRIX AFTER GAUSSIAN ELIMINATION',//)
C INSERT FOR 3x3 AND LESS
      WRITE (6,18)((B(I,J),J=1,NNP1),I=1,NN)
C END OF INSERT
C CHECK TO SEE IF DONE
      IF (NNM.EQ.NUM)GO TO 41
      NNM=NNM+1
      GO TO 7
   41 STOP
      END
```

The following is the insert for writing matrix B when 4x4 or larger.

```
C INSERT FOR 4x4 AND LARGER
      KK=1
   17 KK10=KK+9
      WRITE (6,120)((B(I,J),J=KK,KK10),I=1,NN)
  120 FORMAT (5X,1P10E12.4)
      KK=KK+10
      NK=NNP1-KK10
      WRite (6,121)
  121 FORMAT (//,'CONTINUED',/)
      IF (NK.GT.10)GO TO 17
      WRITE (6.122)((B(I,J),J=KK,NNP1),I=1,NN)
  122 FORMAT (5X,1P6E12.4)                          (6 would be 5 if 5x5)
C END OF INSERT
```

This is the insert for writing matrix B after Gaussian elimination when 4x4 or larger.

```
C INSERT FOR 4x4 AND LARGER
      KK=1
   42 KK10=KK+9
      WRITE (6,120)((B(I,J),J=KK,KK10),I=1,NN)
      KK=KK+10
      NK=NNP1-KK10
      WRITE (6,121)
      IF (NK.GT.10)GO to 42
      WRITE (6,122)((B(I,J),J=kk,NNP1),I=1,NN)
C END OF INSERT
```

To use these, just take out the inserts for 3x3 and less and put these in their place.

<u>Example 3.3</u> (some output)

```
     MATRIX A

        1.0000E 00  0.0          1.0000E 00
        0.0         1.0000E 00  -1.0000E 00
        1.0000E 00  1.0000E 00   0.0
```

MATRIX OF GENERATED SYSTEM

1.0000E 00	0.0	1.0000E 00	0.0	0.0
0.0	0.0	0.0	1.0000E 00	0.0
1.0000E 00	0.0	1.0000E 00	-1.0000E 00	0.0
0.0	1.0000E 00	-1.0000E 00	0.0	0.0
0.0	0.0	0.0	0.0	1.0000E 00
0.0	1.0000E 00	-1.0000E 00	0.0	-1.0000E 00
1.0000E 00	1.0000E 00	0.0	0.0	0.0
0.0	0.0	0.0	1.0000E 00	1.0000E 00
1.0000E 00	1.0000E 00	0.0	-1.0000E 00	-1.0000E 00

(continuing matrix here as too large for space)

0.0	1.0000E 00	0.0	1.0000E 00	1.0000E 00
1.0000E 00	1.0000E 00	0.0	1.0000E 00	0.0
-1.0000E 00	0.0	0.0	0.0	1.0000E 00
0.0	0.0	1.0000E 00	-1.0000E 00	0.0
-1.0000E 00	0.0	1.0000E 00	-1.0000E 00	1.0000E 00
1.0000E 00	0.0	0.0	0.0	-1.0000E 00
0.0	1.0000E 00	1.0000E 00	0.0	1.0000E 00
0.0	1.0000E 00	1.0000E 00	0.0	1.0000E 00
0.0	0.0	0.0	0.0	0.0

MATRIX AFTER GAUSSIAN ELIMINATION

1.0000E 00	0.0	1.0000E 00	0.0	0.0
0.0	1.0000E 00	-1.0000E 00	0.0	0.0
0.0	0.0	0.0	1.0000E 00	0.0
0.0	0.0	0.0	0.0	1.0000E 00
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0

(continuing matrix here as too large for space)

0.0	1.0000E 00	0.0	1.0000E 00	1.0000E 00
0.0	0.0	1.0000E 00	-1.0000E 00	0.0
1.0000E 00	1.0000E 00	0.0	1.0000E 00	0.0
-1.0000E 00	0.0	1.0000E 00	-1.0000E 00	1.0000E 00
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0

MATRIX A

1.0000E 00	0.0	1.0000E 00
0.0	0.0	0.0
0.0	1.0000E 00	-2.0000E 00

MATRIX OF GENERATED SYSTEM

1.0000E 00	0.0	1.0000E 00	0.0	0.0
0.0	0.0	0.0	0.0	0.0
1.0000E 00	0.0	1.0000E 00	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	1.0000E 00	-2.0000E 00	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	1.0000E 00	-2.0000E 00	0.0	0.0

(continuing matrix here as too large for space)

0.0	0.0	0.0	0.0	1.0000E 00
0.0	1.0000E 00	0.0	1.0000E 00	0.0
0.0	-2.0000E 00	0.0	-2.0000E 00	1.0000E 00
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0000E 00	-2.0000E 00	1.0000E 00
0.0	0.0	-2.0000E 00	4.0000E 00	-2.0000E 00

MATRIX AFTER GAUSSIAN ELIMINATION

1.0000E 00	0.0	1.0000E 00	0.0	0.0
0.0	1.0000E 00	-2.0000E 00	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0

(continuing matrix here as too large for space)

0.0	0.0	0.0	0.0	1.0000E 00
0.0	0.0	0.0	0.0	0.0
0.0	1.0000E 00	0.0	1.0000E 00	0.0
0.0	0.0	1.0000E 00	-2.0000E 00	1.0000E 00
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0

3.4 PROGRAM FOR A^+

This program is done using the method presented in 2.2. After the matrix is read in, calculations are made that are necessary to compute A_2^+. Then a check is made to see if the program is done, that is, if A^+ has been calculated. If not, it performs the steps to calculate A_3^+ and then goes back to the check.

The output consists of printing the matrix A and then A^+ which is called matrix X in this program. The format used is 1P3E16.6 (3 can be replaced by a different number depending on the number of elements in a row).

This program also had a problem with machine error as before. This time the problem was in the check to see if c_k was zero (not in Gaussian elimination as before). Again a check was made that resulted in causing numbers that are closed to zero (within .000001 of zero) to actually be zero. This ended the problem.

```
C FINDING THE PSEUDO-INVERSE OF A MATRIX
      DIMENSION A(5,5),AA(5,5),D(5),C(5),B(5),DB(5,5)
C READ IN NUMBER OF MATRICES (MUST BE SAME SIZE)
      READ (5,1)NUM
    1 FORMAT (I3)
      NNM=1
C READ IN MATRIX SIZE AND MATRIX
      READ (5,2)M,N
    2 FORMAT (2I2)
    3 DO 4 I=1,M
    4 READ (5,6)(A(I,J),J=1,N)
    6 FORMAT (3F5.0)
C GET A1+
C CHECK TO SEE IF 1ST COL OF A IS THE 0-VECTOR
      DO 8 I=1,M
      IF (A(I,1).NE.0.)GO TO 12
    8 CONTINUE
C 1ST COL IS 0-VECTOR
      DO 10 I=1,M
   10 AA(1,I)=0.
      GO TO 17
C 1ST COL IS NOT 0-VECTOR
   12 P=0
      DO 14 I=1,M
```

```
   14 P=P+A(I,1)*A(I,1)
      DO 16 I=1,M
   16 AA(1,I)=A(I,1)/P
C CALCULATE D,C,B, AND A2+
   17 D(1)=D(1)+AA(1,I)*A(I,2)
      DO 18 I=1,M
   18 D(L)=D(1)+AA(1,I)*(I,2)
      DO 22 I=1,M
   22 C(I)=A(I,2)-D(L)*A(I,1)
C CHECK TO SEE IF C IS 0-VECTOR
      DO 24 I=1,M
      IF (C(I).NE.0.)GO TO 27
   24 CONTINUE
C C IS THE 0-VECTOR
      P=D(1)*D(1)+1
      P=D(1)/P
      DO 26 I=1,M
   26 B(I)=P*AA(1,I)
      GO TO 31
C C IS NOT THE 0-VECTOR
   27 P=0
      DO 28 I=1,M
   28 P=P+C(I)*C(I)
      DO 30 I=1,M
   30 B(I)=C(I)/P
C CALCULATE A2+
   31 DO 32 I=1,M
   32 AA(1,I)=AA(1,I)-D(1)*B(I)
      DO 34 I=1,M
   34 AA(2,I)=B(I)
C CHECK TO SEE IF DONE
      K=2
   35 IF (K.EQ.N)GO TO 62
C NOT DONE
C CALCULATE D,C,B, AND AK+
      K=K+1
      KM1=K-1
C CALCULATE D
      DO 36 I=1,KM1
   36 D(I)=0.
      DO 38 I=1,KM1
      DO 38 J=1,M
   38 D(I)=D(I)+AA(I,J)*A(J,K)
C CALCULATE C
      DO 40 I=1,M
   40 C(I)=0.
      DO 42 I=1,M
      DO 42 J=1,KM1
   42 C(I)=C(I)+A(I,J)*D(J)
      DO 44 I=1,M
   44 C(I)=A(I,K)-C(I)
      DO 45 I=1,M
      IF (ABS(C(I)).LE..000001)C(I)=0.
   45 CONTINUE
```

```
C CHECK TO SEE IF C IS THE 0-VECTOR
      DO 46 I=1,M
      IF (C(I).NE.0.)GO TO 53
   46 CONTINUE
C C IS THE 0-VECTOR
      P=0.
      DO 48 I=1,KM1
   48 P=P+D(I)*D(I)
      P=1/(P+1)
      DO 49 I=1,M
   49 B(I)=0.
      DO 50 J=1,M
      DO 50 I=1,KM1
   50 B(J)=B(J)+D(I)*AA(I,J)
      DO 51 I=1,M
   51 B(I)=B(I)*P
      GO TO 55
C C IS NOT THE 0-VECTOR
   53 P=0
      DO 52 I=1,M
   52 P=P+C(I)*C(I)
      DO 54 I=1,M
   54 B(I)=C(I)/P
C CALCULATE AK+
   55 DO 56 I=1,KM1
      DO 56 J=1,M
   56 DB(I,J)=D(I)*B(J)
      DO 58 K=1,KM1
      DO 58 J=1,M
   58 AA(I,J)=AA(I,J)-DB(I,J)
      DO 60 J=1,M
   60 AA(K,J)=B(J)
      DO 61 I=1,N
      DO 61 J=1,M
      IF (ABS(AA(I,J)).LE..000001)AA(I,J)=0.
   61 CONTINUE
      GO TO 35
C WRITE MATRICES A AND X
   62 WRITE (6,64)
   64 FORMAT (///,5X,'MATRIX A',/)
      WRITE (6,66)((A(I,J),J=1,N),I=1,M)
   66 FORMAT (5X,1P3E16.6)
      WRITE (6,68)
   68 FORMAT (///,5X,'MATRIX X' ,/)
      WRITE (6,66)((AA(I,J),J=1,M),I=1,N)
      IF (NNM.EQ.NUM)GO TO 69
      NNM=NNM+1
      GO TO 3
   69 STOP
      END
```

Example 3.4 (some output)

```
MATRIX A

    1.000000E 00  0.0           1.000000E 00
    0.0           1.000000E 00 -1.000000E 00
    1.000000E 00  1.000000E 00  0.0


MATRIX X

    3.333334E-01  0.0           3.333333E-01
    0.0           3.333334E-01  3.333333E-01
    3.333333E-01 -3.333333E-01  0.0


MATRIX A

    1.000000E 00  0.0           1.000000E 00
    0.0           0.0           0.0
    0.0           1.000000E 00 -2.000000E 00


MATRIX X

    8.333334E-01  0.0           3.333333E-01
    3.333333E-01  0.0           3.333335E-01
    1.666666E-01  0.0          -3.333333E-01


MATRIX A

    6.000000E 00  2.000000E 00 -2.000000E 00
   -2.000000E 00  2.000000E 00  2.000000E 00
    2.000000E 00  2.000000E 00  2.000000E 00


MATRIX X

    0.0          -2.499996E-01  2.499999E-01
    2.500005E-01  4.999996E-01 -2.500001E-01
   -2.500006E-01 -2.499995E-01  5.000002E-01
```

CHAPTER IV RELATIONS BETWEEN INVERSES

Sometimes, it is important to know some relationships between the different generalized inverses. Say, for example, that a given relationship exists for two generalized inverses, A_s and A_t. Suppose that A_s is harder to calculate than A_t. Then there is a possibility that A_t could be used along with the given relationship to obtain A_s (resulting in a much easier calculation of A_s). For example, if A is a nonsingular square matrix, then $A^{-1} = A' = A^{+} = A_d = A_g$. And if A is a square matrix not equal to the zero matrix 0, then $\lim_{\lambda\to 0} (\lambda I_n + A^T A)^{-1} A^T = A^{+}$ where $\lambda\to 0$ means $\lambda\to 0$ through any neighborhood of 0 in R (reals) which excludes the nonzero eigenvalues of $-A$ [2,p.174]. Again let A be square. Then (a) A_d can be expressed as a polynomial in A ($A_d = A^k(qIA))^{k+1}$ where q(A) is an appropriately defined polynomial) and (b) A^{+} is expressable as a polynomial in A if and only if A is range-Hermitian [2,p.172-173]. It is known [12,p.50-51] that $A_d = A^k(A^{2k+1})^{+}A^k$ and $A_d^{+} = (A^k)^{+}A^{2k+1}(A^k)^{+}$. But is it possible to relate A_d to A^{+}?

Many relationships among the various generalized inverses are known and used, but there is an open question as to whether a relationship exists between A_d and A^{+}. The following results were obtained in an attempt to find such a relationship.

4.1 CONDITIONS FOR A_d AND A^{+} TO HAVE THE SAME INDEX

In order to find the conditions for A_d and A^{+} to have the same index, it is necessary to decide when A and A^{+} have the same index. Notice that when A is nonsingular, it has index 0 and $A_d = A^{+} = A^{-1}$ which also has index 0. So if A is nonsingular, the index A = index A^+ = index A_d = 0. Hence, there is need only to consider the case when the index is greater than 0. The following theorems help in finding when A and A^+ have the same index.

Theorem 4.1 rank A = rank A^+A = rank A^+.

Proof: Recall that rank AB $\leq$ rank A or rank AB $\leq$ rank B. Rank $A^+A \leq$ rank A = rank $AA^+A \leq$ rank A^+A, so rank A = rank A^+A. Now, rank $A^+A \leq$ rank A^+ = rank $A^+AA^+ \leq$ rank A^+A, so rank A^+ = rank A^+A. Therefore, rank A = rank A^+A = rank A^+.

Theorem 4.2 $(A^T)^+ = (A^+)^T$

Proof: Notice that $(A^+)^T$ satisfies the definition of $(A^T)^+$:

(1) $A^T(A^+)^TA^T = (AA^+A)^T = A^T$

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

(3) $[(A^+)^TA^T]^T = [(AA^+)^T]^T = [AA^+]^T = (A^+)^TA^T$

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

Theorem 4.3 $(AA^T)^+ = (A^+)^TA^+$ and $(A^TA)^+ = A^+(A^T)^+$

Proof: Observe $(A^+)^TA^+$ satisfies the definition of $(AA^T)^+$:

(1) $$\begin{aligned} AA^T[(A^+)^TA^+]AA^T &= A(A^+A)^TA^+AA^T \\ &= AA^+AA^+AA^T \\ &= AA^+AA^T \\ &= AA^T \end{aligned}$$

(2) $$\begin{aligned} [(A^+)^TA^+]AA^T[(A^+)^TA^+] &= (A^+)^TA^+A(A^+A)^TA^+ \\ &= (A^+)^TA^+AA^+AA^+ \\ &= (A^+)^TA^+AA^+ \\ &= (A^+)^TA^+ \end{aligned}$$

(3) $[(A^+)^T A^+ AA^T]^T = AA^T(A^+)^T A^+$

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

$= AA^+AA^+$

$= AA^+$

$= (AA^+)^T$

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

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

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

(4) $[AA^T(A^+)^T A^+]^T = [A(A^+A)^T A^+]^T$

$= [AA^+AA^+]^T$

$= [AA^+]^T$

$= AA^+$

$= AA^+AA^+$

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

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

The second part, $(A^TA)^+ = A^+(A^T)^+$, is done similarly.

<u>Theorem 4.4</u> $A^+ = (A^TA)^+A^T$

Proof: $(A^TA)^+A^T = A^+(A^T)^+A^T$ (Theorem 4.3)

$= A^+(A^+)^T A^T$ (Theorem 4.2)

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

$= A^+AA^+$

$= A^+$

<u>Theorem 4.5</u> $(AA^T)^+AA^T = AA^+$

Proof: $(AA^T)^+AA^T = [(AA^T)^+AA^T]^T$

$= [(A^+)^T A^+ AA^T]^T$ (Theorem 4.3)

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

$= AA^+AA^+$

$= AA^+$

Definition 4.1 A square matrix is normal if $AA^T = A^TA$.

Theorem 4.6 $A^+A = AA^+$ if A is normal.

Proof: $A^+A = (A^TA)^+A^TA$ (Theorem 4.4)

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

$= AA^+$ (Theorem 4.5)

Theorem 4.7 $(A^k)^+ = (A^+)^k$ if A is normal and k is a positive integer.

Proof: Notice that $(A^+)^k$ satisfies the definition of $(A^k)^+$:

(1) $$A^k(A^+)^kA^k = \underbrace{A\cdot A\cdots A}_{k\text{ times}}\ \underbrace{A^+\cdot A^+\cdots A^+}_{k\text{ times}}\ \underbrace{A\cdot A\cdots A}_{k\text{ times}}$$

$$= \underbrace{AA^+A\cdot AA^+A\cdots AA^+A}_{k\text{ times}}$$

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

$$= A^k$$

(2) $(A^+)^kA^k(A^+)^k = (A^+)^k$ is easily shown in a manner similar to that used in (1).

(3) $$[(A^+)^k]^T = [\underbrace{A^+\cdot A^+\cdots A^+}_{k\text{ times}}\cdot\underbrace{A\cdot A\cdots A}_{k\text{ times}}]^T$$

$$= [\underbrace{A^+A\cdot A^+A\cdots A^+A}_{k\text{ times}}]^T$$

$$= \underbrace{(A^+A)^T\cdot(A^+A)^T\cdots(A^+A)^T}_{k\text{ times}}$$

$$= \underbrace{(A^+A)\cdot(A^+A)\cdots(A^+A)}_{k\text{ times}}$$

$= (A^{+}A)^{k}$

$= (A^{+})^{k}A^{k}$

(4) $[A^{k}(A^{+})^{k}]^{T} = A^{k}(A^{+})^{k}$ is shown in a manner similar to that used in(3).

Theorem 4.8 If A is normal, index A = index A^{+}.

Proof: Let the index of A be k. Then rank A^{k} = rank A^{k+1}. Rank $(A^{k})^{+}$ = rank A^{k} and rank $(A^{k+1})^{+}$ = rank A^{k+1} by Theorem 4.1. But rank $(A^{k})^{+}$ = rank $(A^{+})^{k}$ and rank $(A^{k+1})^{+}$ = rank $(A^{+})^{k+1}$ by theorem 4.7. So rank $(A^{+})^{k}$ = rank $(A^{+})^{k+1}$.

Assume that $t < k$ and that rank $(A^{+})^{t}$ = rank $(A^{+})^{t+1}$. By Theorem 4.7, rank $(A^{t})^{+}$ = rank $(A^{t+1})^{+}$. And by Theorem 4.1, rank A^{t} = rank A^{t+1}. But this contradicts that k is the index of A. Therefore, $t \geq k$ and the index of A^{+} is k by definition

Therefore, if A is normal, A and A^{+} have the same index. Now find the conditions for A and A_d to have the same index.

Theorem 4.9 If A is singular, the index of A_d is 1.

Proof: rank ${A_d}^{2} \leq$ rank A_d = rank $AA_d^{2} \leq$ rank A_d^{2} from Lemma 1.1. Hence, rank A_d = rank A_d^{2} and index A_d = 1.

So it is only necessary to determine when A has index 1 since A_d always has index 1.

Definition 4.2 A square matrix A is range-Hermitian if $R(A^{T}) = R(A)$ where R(A) denotes range of A.

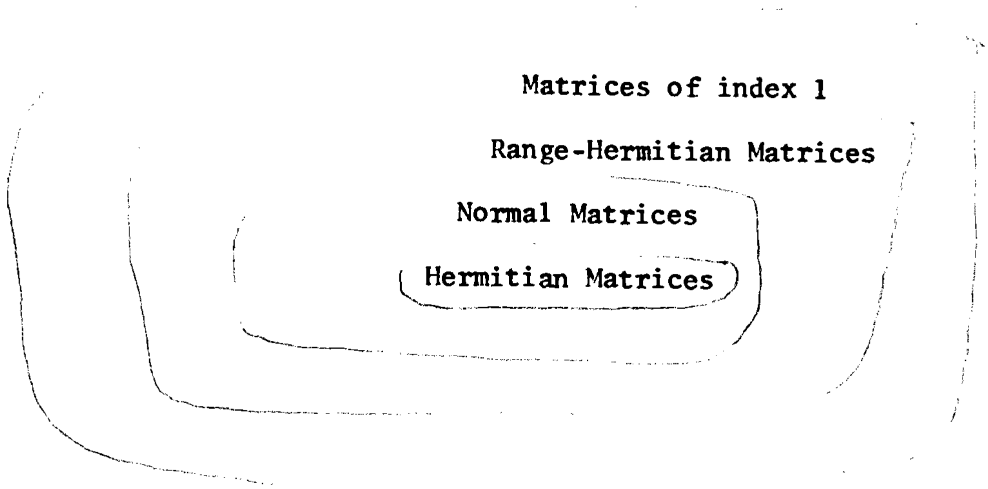
A range-Hermitian matrix also has the property that $N(A^{T}) = N(A)$ where N(A) denotes the null space of A. [2,p.163]

<u>Definition 4.3</u> A square matrix is Hermitian if $A = A^T$.

<u>Theorem 4.10</u> Every range-Hermitian matrix has index 1.

Proof: [2,p.165]

The following diagram drawn from information given in [2,p.166] is useful.



It can be concluded that if A is range-Hermitian, A and A_d have the same index which is 1. And if A is normal, A^+ and A_d have the same index which is also 1. (All three have the same index (1) when A is normal.)

Another interesting fact, found in [2,p.163], is that $A_d = A_g = A^+$ if and only if A is range-Hermitian.

CHAPTER V CONCLUSION

What has been presented here is mainly an introduction to the subject of generalized inverses. The main inverses (A', A^+, A_d, and A_g) were presented along with at least one method of calculation. Computer programs were made available for A', solutions in particular and general form, and A^+. A few relations among the different generalized inverses were mentioned, and the possibility of a relationship between A_d and A^+ was considered. The concept of generalized inverses is fairly new so there are areas for further study. Certainly it would be interesting to look into some applications of the generalized inverses presented here. Section 1.3 gives some references that should be helpful.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Albert, Arthur. Regression and the Moore-Penrose Pseudoinverse: New York, Academic Press, Vol. 94, 1972.

2. Ben-Israel, Adi and Greville, Thomas N. E. Generalized Inverses, Theory and Application: New York, John Wiley and Sons (Interscience Publishers), 1971.

3. Boullion, Thomas L. and Odell, Patrick L. Generalized Inverse Matrices: New York, John Wiley and Sons (Interscience Publishers), 1971.

4. Campbell, Stephen L., Meyer, Carl D., and Rose, Nicholas J. Applications of the Drazin Inverse to Linear Systems of Differential Equations: North Carolina State University, Raleigh, North Carolina.

5. Hartwig, R. E. and Hall, F. J. Applications of the Drazin Inverse to Cesaro-Neumann Iterations: North Carolina State University and Pembroke State University, North Carolina.

6. Kruskal, William. The Geometry of Generalized Inverses; reprint from The Journal of the Royal Statistical Society, Series B (Methodological), Vol. 37, No. 2, 1975, 272-283.

7. Marcus, Marvin. Notes on Modern Algebra: University of California, Santa Barbara, 1972.

8. Marcus, Marvin and Mic, Henryk. A Survey of Matrix Theory and Matrix Inequalities: Boston, Allyn and Bacon, Inc., 1964.

9. Poole, G. Geometry of Minimum Norm and Least Squares Solutions to Matrix Equations, Delta, Fall 1973, 12-17.

10. Poole, G. and Boullion, T. The Drazin Inverse and a Spectral Inequality of Marcus, Mic, and Moyls, Journal of Optimization Theory and Applications, Vol. 15, No. 5, 1975, 503-508.

11. Rao, C. R. and Mitra, S. K. Generalized Inverse of Matrices and Its Applications: New York, John Wiley and Sons, Inc., 1971.

12. ______________________. Theory and Application of Generalized Inverses of Matrices: Texas Technological College, Symposium Proceedings, March 1968.