AN INTERACTIVE MATRIX INTERPRETIVE SYSTEM FOR THE IBM 360/67

Robert Douglas Little

United States
Naval Postgraduate School

THESIS

AN INTERACTIVE MATRIX INTERPRETIVE SYSTEM
FOR THE IBM 360/67

by

Robert Douglas Little

June 1970

An Interactive Matrix Interpretive System
For the IBM 360/67

by

Robert Douglas Little
Lieutenant, United States Navy
B.S., United States Naval Academy, 1963

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1970

ABSTRACT

The Matrix Interpretive System devised by Wilson and modified by Cantin has been transformed into an interactive program operable under a time sharing system. Several new commands have been added to extend the usefulness of the code and give it the capability to offline read, write and punch data, to plot graphs and contour maps and to integrate simple polynomial expressions.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to Professor E. L. Wilson of the University of California, the originator of the program. He is also grateful to Professor Gilles Cantin of the Naval Postgraduate School, his advisor, for his constructive supervision and encouragement throughout this work.

The Naval Postgraduate School Computer Center provided facilities and technical advice for this work.

I. INTRODUCTION

A. GENERAL DESCRIPTION

Many engineering problems in such fields as steady-state temperature distribution [Ref. 1], stress in elastic structures [Ref. 2], and fluid flows [Ref. 3] can be formulated as problems of numerical linear algebra. Until the advent of modern digital computers, all but the simplest of these problems had to be reserved to specialists for solutions. Even problems of moderate size required a considerable computational effort. The computer changed all of this. All of the simple matrix operations like multiplication, transposition, solution of eigen systems can be coded to solve individual problems. The computer program described in this report integrates a number of standard subprograms of linear algebra [Refs. 4,5,6,7] into a problem oriented computer language. This program, heretofore called: Matrix Interpretive System was coded in FORTRAN IV. The resulting matrix operations sub-language has a syntax and rules of its own as well as a certain number of diagnostic messages that can be automatically returned to the user of the language. The basic specification for the design of the language structure was that it had to be simple enough to be used by persons unfamiliar with computer programming. As written, the language could be used to solve stacks of problems sequentially; if an error is found in one of the problems a message is printed and a solution for the next problem is attempted. To be effective in a conversational mode at a remote station, the program had to be heavily modified so as

to leave effective control of the flow of operations to the user. This version of the code is called: Interactive Matrix Interpretive System.

The Interactive Matrix Interpetive System is a general program written in FORTRAN IV [Ref. 8], it is a problem oriented sub-language. The entire program is coded in double precision arithmetic. The only subroutines which are used but are not contained in the program are a square root and a natural log subroutine. The program uses 160,000 bytes of core storage.

B. HISTORICAL BACKGROUND

The original program was developed by Professor E. L. Wilson of the University of California in April 1963 for an IBM 7094. It was later adapted by Professor G. Cantin of the Naval Postgraduate School for a CDC 1604 in February 1966 and for the IBM 360/67 in August 1968.

C. OBJECTIVES

The objectives of the author's work have been to:

1. transform the program into an interactive system operable under time sharing from a remote typewriter station;

2. add operations to extend the capabilities of the program;

3. create a "USER'S MANUAL" to facilitate the use of the program.

II. GENERAL PROGRAM OPERATION

The Interactive Matrix Interpretive System will be referred to as IMIS in this and following sections of this report. IMIS is written in FORTRAN IV language for the IBM 360/67 time sharing system available at the Naval Postgraduate School (CP/CMS). A different version called DSMIS is used for batch processing. All operations are performed in double precision arithmetic. Matrices are stored internally in a single dimensional array, A, that is 10,000 double words in size. Matrix names are stored in an array called NAME. Arrays, NROW and NCOL, are used to store the number of rows and columns of each matrix.

The following is a brief description of the function and operation of most of the subroutines of IMIS. Some of the subroutines are straightforward and will not be discussed. A flow diagram of the program is presented in Figure 1.

A. MAIN PROGRAM AND SERVICE SUBROUTINES

1. Main Program

The Main program is used only to initialize internal variables and to call the main service subroutine, SMIS1.

2. Main Working Subroutine (SMIS1)

SMIS1 is the general working subroutine. Some of the operations such as duplication of a matrix and envelope of a matrix are carried out in SMIS1. The other operations are carried out in subroutines which are called from SMIS1. Most of the error messages are called from SMIS1.

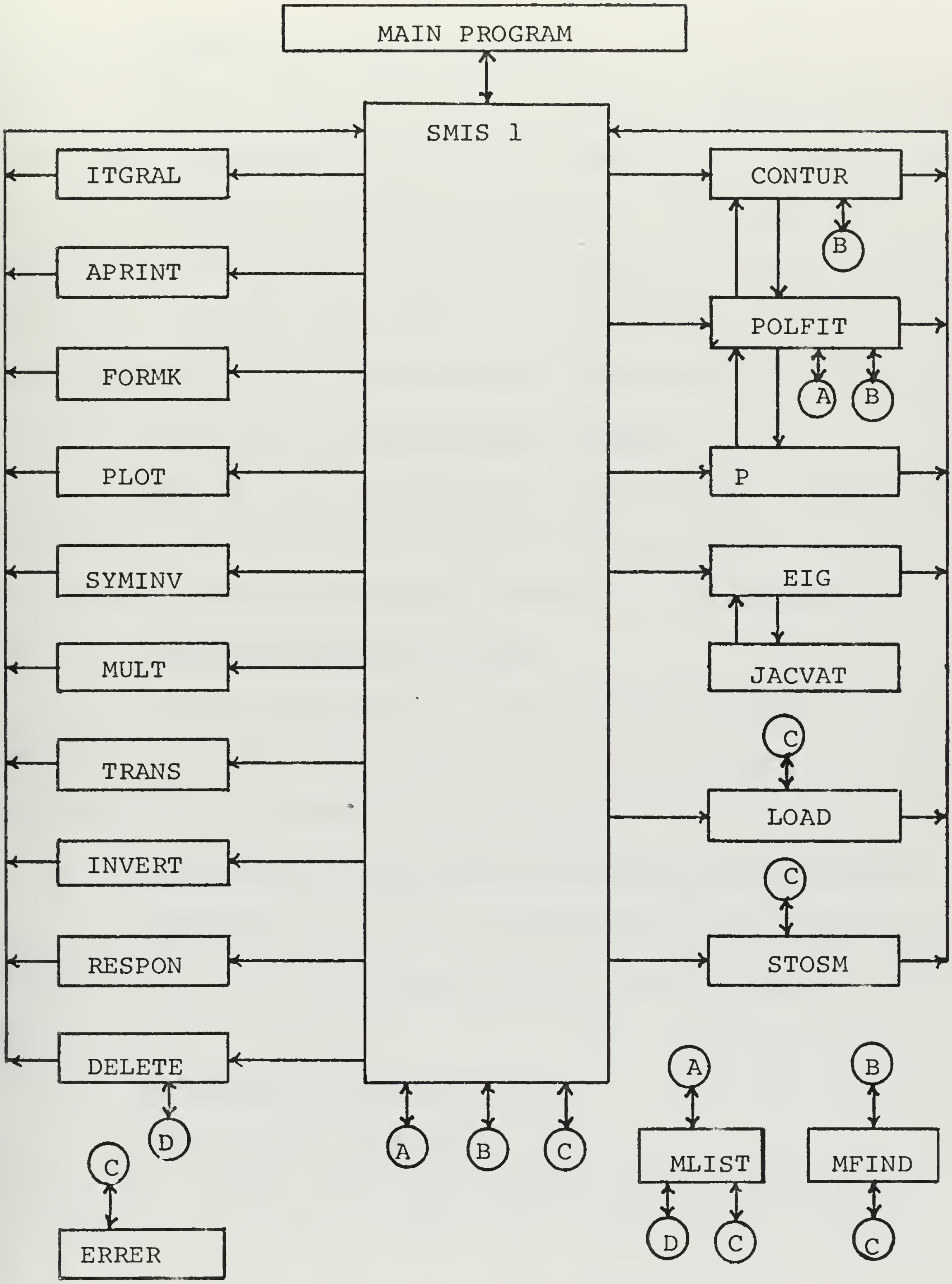


Figure 1. Functional Flow Diagram

3. <u>Listing of a Matrix (MLIST)</u>

MLIST stores each matrix by name, number of rows and columns and starting position in the array, A. It checks the name of a new matrix to be stored against the names of previously stored matrices to see if the name has been used before. If there is another matrix with the same name, the previously stored matrix will be deleted before the new matrix is stored.

4. <u>Printing of Error Messages (ERRER)</u>

When an error is made in an operational command, ERRER is called. It prints out an error message, deletes the current operational command and returns control to the typewriter.

5. <u>Locating a Matrix (MFIND)</u>

Locates the matrix or matrices to be used in an operation. It finds the number of rows and columns of a matrix and the starting position in the array A.

6. <u>Calculating Eigenvalues and Eigenvectors (JACVAT)</u>

Determines all of the eigenvalues and eigenvectors of a matrix. It uses the threshold cyclic Jacobi rotation method [Ref. 4].

7. <u>Formation of Matrix N (P)</u>

The bicubic polynomial,

$$P=C_1X^3Y^3+C_2X^3Y^2+C_3X^3Y+C_4X^3+C_5X^2Y^3+C_6X^2Y^2+C_7X^2Y+C_8X^2+$$

$$C_9XY^3+C_{10}XY^2+C_{11}XY+C_{12}X+C_{13}Y^3+C_{14}Y^2+C_{15}Y+C_{16}$$

is used in the polynomial fit operation. This polynomial can be written as

$$P = \langle F_i \rangle \{C_i\}_{i=1,2,\ldots,16}$$

where $\langle F_i \rangle = \langle X^3Y^3, X^3Y^2, X^3Y, X^3, X^2Y^3, X^2Y^2, X^2Y, X^2,$

$XY^3, XY^2, XY, X, Y^3, Y^2, Y, 1 \rangle$

and $\{C_i\} = \langle C_1, C_2, \ldots, C_{16} \rangle^T$. In the POLFIT operation, values of P at all of the sixteen nodal points of the grid shown in Figure 2 must be used to solve for all of the unknown coefficients of the polynomial using the following matrix equation:

$$\{P_i\} = [N] \{C_i\}_{i=1,2,\ldots,16}$$

where

$$[N] = \begin{bmatrix} \langle F_i^{(1)} \rangle \\ \langle F_i^{(2)} \rangle \\ \vdots \\ \langle F_i^{(16)} \rangle \end{bmatrix}$$

Subroutine P forms matrix N for use in the POLFIT operation. The bicubic polynomial, P, will be referred to as CUPOL in the following sections.

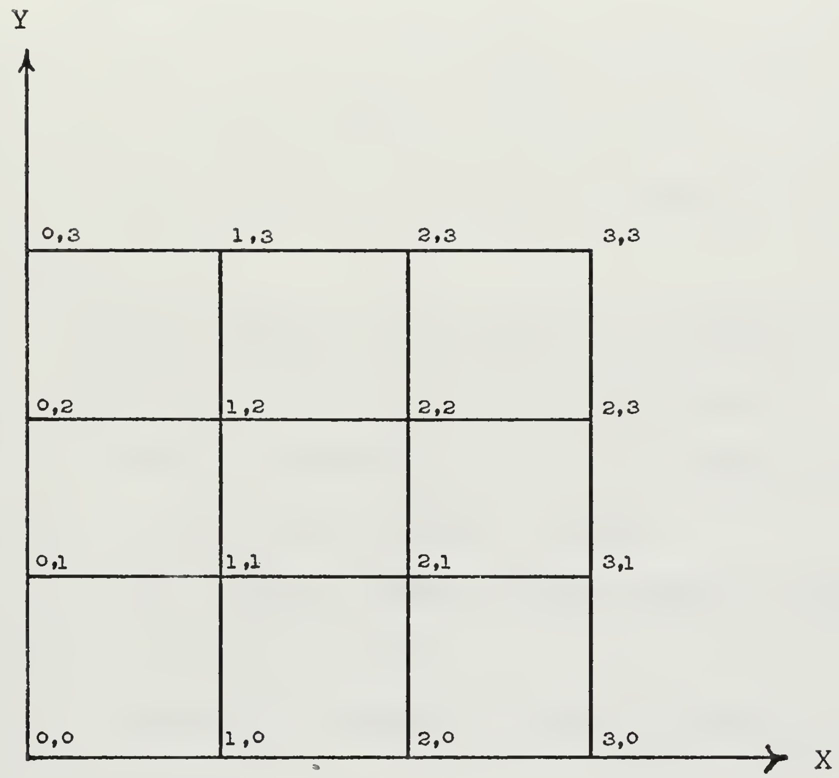


Figure 2. Grid For Subroutine P

B. OPERATION SUBROUTINES

1. Inverting a Symmetric Matrix (SYMINV)

SYMINV inverts a symmetric matrix, A, by Gauss elimination [Ref. 4]. If the matrix is not symmetric, it takes the average of the two numbers, A(i,j),A(j,i), and replaces both by the average value and prints a warning that the matrix was not symmetric.

2. Adding, Removing and Storing a Submatrix (STOSM)

STOSM is called to do three operations.

a) Store a submatrix at an indicated position in a matrix when called by the command, STOSM.

b) The command, RMVSM, calls STOSM to remove a submatrix from the original matrix.

c) ADDSM calls STOSM to add a submatrix to a matrix at an indicated position.

3. Function Generation (FUNGN)

This subroutine takes a function described by a set of points $(X_i,Y_i)_{i=1,2,...,n}$, not necessarily equally spaced, and calculates, by straight line interpolation, ordinates at equally spaced values of X.

4. Printing of Graphs (PLOT)

PLOT prints a 10 inch by 10 inch graph of 1 to 4 curves. It automatically scales the graph to the maximum and minimum values of X and Y. The axes are plotted as

a) X------X

b) Y------Y

with the characters, X and Y, appearing at every inch on their respective axes. When all of the points are plotted on one side of either axis, an offset axis replaces the real axis and is indicated by

a) A------A offset X axis

b) B------B offset Y axis

The origin of the axes and the X and Y scale factors are calculated and printed below the graph.

5. Determining the Coefficients of a Bicubic Polynomial (POLFIT)

The 16 unknown coefficients of a bicubic polynomial surface can be determined in terms of 16 known values of the polynomial at 16 known values of the independent variables.

POLFIT solves the matrix equation,

$$[N] \{C\} = [P]$$

for the matrix of unknown coefficients, C, where matrix N is formed by the subroutine P. Matrix P is a matrix of the values of CUPOL at each nodal point and is the input to POLFIT.

6. Printing Contour Maps (CONTUR)

CONTUR prints a contour map of CUPOL. There are 21 possible contour lines which are represented by the alphabetic letters, A through J, O, Q through Z. The input is either the coefficients of CUPOL with the upper and lower limits of X and Y or the values of the function at the grid points used for POLFIT. In the latter case, POLFIT is called and the coefficients of CUPOL are calculated.

CUPOL is then scanned over the ranges of X and Y to determine the maximum and minimum Z ordinate. The range of Z is divided into 42 equal parts represented by alphabetic characters and blanks assigned to alternating contour values. CUPOL is then evaluated at each point on a line, the correct character inserted and the line printed.

7. Formation of a 2 by 2 Stiffness Matrix (FORMK)

FORMK forms the stiffness matrix [Ref. 9],

$$[K] = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

for a simply supported beam having length, L, Young's Modulus, E, moment of inertia, I, shear modulus, G, and effective shear area, $\bar{A}$.

The stiffness coefficients are:

$$k_{11} = \frac{2EI}{L}\left(\frac{2+\beta}{1+2\beta}\right) = k_{22}$$

$$k_{12} = \frac{2EI}{L}\left(\frac{1-\beta}{1+\beta}\right) = k_{21}$$

where

$$\beta = \frac{6EI}{L^2\bar{A}G}$$

The matrix equation,

$$\begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix} = [K] \begin{Bmatrix} \theta_1 \\ \theta_2 \end{Bmatrix}$$

relates end moments, M_1, M_2, to their corresponding rotations, θ_1, θ_2 as shown in Figure 3.

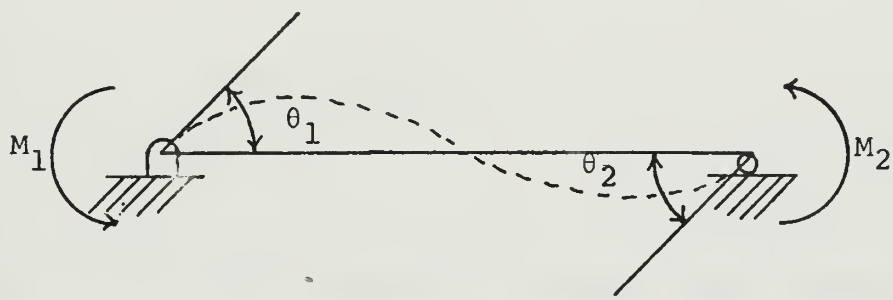


Figure 3. Simple Beam

8. Numerical Integration (ITGRAL)

The subroutine, ITGRAL, is capable of performing numerical integration of one and two dimensional functions described by discrete sets of points, either by Simpson's rule, or, if the function to be integrated is known, by Gaussian quadrature.

9. Numerical Integration of a Set of Differential Equations (RESPON)

In the study of vibrations of systems with many degrees of freedom, the following system of ordinary differential equations governs the problem:

$$[M]\ \{\ddot{X}\} + [C]\ \{\dot{X}\} + [K]\ \{X\} = \{P\} \qquad (1)$$

where

[M] is the mass matrix,

[C] is the damping matrix

and

[K] is the stiffness matrix.

In general, if

$$\{X\} = [V]\ \{x\} \qquad (2)$$

where matrix [V] is the matrix of undamped mode shapes and {x} is the vector of the time dependent amplitude function for each mode, we get after substitution of (2) into (1):

$$[M]\ [V]\ \{\ddot{x}\} + [C]\ [V]\ \{\dot{x}\} + [K]\ [V]\ \{x\} = \{P\} \qquad (3)$$

After premultiplication by $[V]^T$ and the use of the orthogonality properties of normal modes:

$$[V]^T\ [M]\ [V] = [\ulcorner m \lrcorner]$$

$$[V]^T\ [K]\ [V] = [\ulcorner k \lrcorner] = [\ulcorner \omega^2 m \lrcorner]$$

$$[V]^T\ [C]\ [V] = [\ulcorner c \lrcorner] = [\ulcorner 2\lambda\omega m \lrcorner]\ ^{(a)}$$

(a) Such an orthogonality condition imposes some restrictions on the form of the damping matrix [C].

where

$$\lambda = \frac{c_i}{2m_i\omega_i}$$

and using the notation

$$[V]^T \{P\} = \{p\}$$

we get

$$[\ulcorner m \lrcorner]\ \{\ddot{x}\} + [\ulcorner 2\lambda\omega m \lrcorner]\ \{\dot{x}\} + [\ulcorner \omega^2 m \lrcorner]\ \{x\} = \{p\} \qquad (4)$$

or

$$\{\ddot{x}\} + [\ulcorner 2\lambda\omega \lrcorner]\ \{\dot{x}\} + [\ulcorner \omega^2 \lrcorner]\ \{x\} = \{p^*\} \qquad (5)$$

where $p_i^* = p_i/m_i$.

This last set of equations is completely decoupled and can easily be numerically integrated. The subroutine RESPON performs this task.

III. OPERATIONAL COMMANDS

The following is a list of the operations that IMIS contains. The operations must conform to the standard format shown in Figure 4. OPERATION stands for any of the symbolic command names mentioned below. A, B, C and D are matrices used as arguments of the operation. N1, N2, N3 and N4 are four integer arguments that must always be right justified. S1 and S2 are two floating point arguments. The arguments can be input or output values and are not all necessary for each command.

There are two versions of the matrix interpretive system in use at the Naval Postgraduate School. One is for time sharing and one for batch processing. The commands denoted by a "*" can only be used in conjunction with the time sharing option. The READ, WRITE and PUNCH commands make use of sequential input/output files which are available with the FORTRAN IV language [Ref. 8].(b)

The READ command uses the file FT04F001. This file is used to load matrices into IMIS from punched cards and is read into the computer prior to executing IMIS. Any data stored in the file is automatically erased when new data is read into the file.

File FT08F001 is used to store any output which the user wants to print offline. This file is not automatically erased

---

(b)A certain knowledge of the CP/CMS time sharing system is needed when using a remote terminal of the computer. It is presumed that the reader is already familiar with this system [Ref. 10].

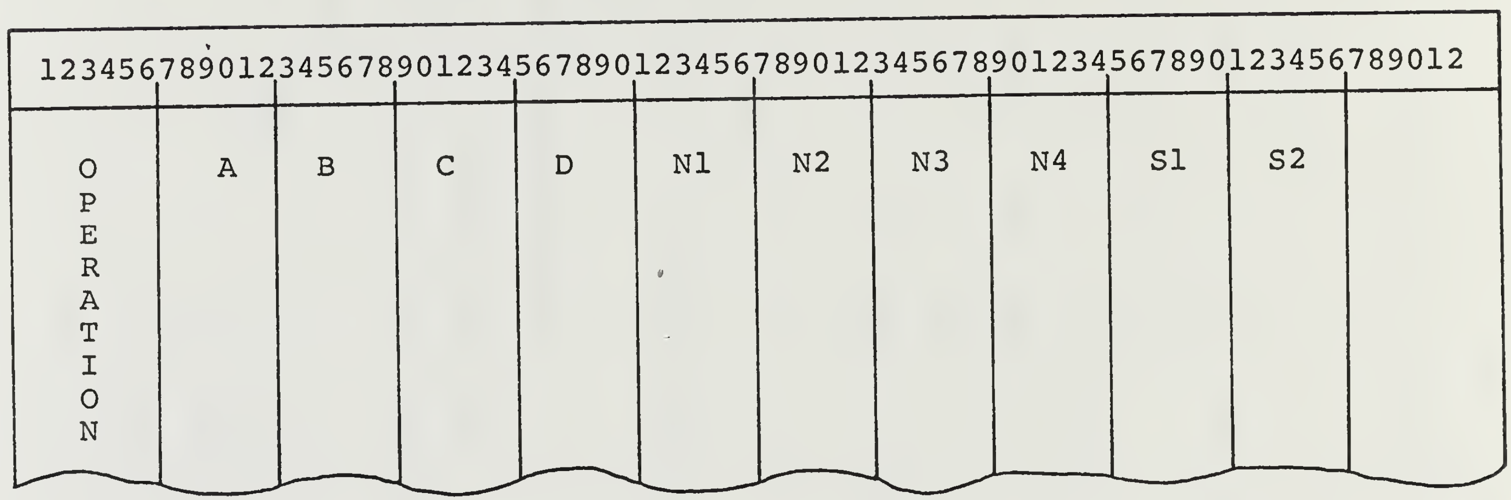


Figure 4. Standard Format

when new data is read into the file. New data is appended to the data that is already stored in the file. Therefore, the file must be erased before executing IMIS. To erase the file, the command,

ERASE$_b$FILE$_b$FT08F001

is used. After exiting from IMIS, the command,

OFFLINE$_b$PRINTCC$_b$FILE$_b$FT08F001

will cause the file to be printed on the offline printer.

The command, PUNCH, uses file FT07F001 to store data to be punched onto cards. This file appends new data at the end of old data so the file must be erased before using IMIS. The command,

ERASE$_b$FILE$_b$FT07F001

erases the file. To offline punch the data stored in the file, the command,

OFFLINE$_b$PUNCH$_b$FILE$_b$FT07F001

is issued after exiting from IMIS. Examples of the use of these files are given in the sample problems. The operational commands are:

START   This command causes the elimination of all matrices which are in core storage and must be the first command used in the solution of a problem. If an error is found during execution of a command, an error message is printed on the

typewriter, the current command eliminated and control returned to the typewriter. If an error is found during the execution of a problem with DSMIS, the current problem is automatically deleted and the remaining cards scanned until a new START or STOP command is found.

LOAD — Matrix Load:

A = Name of matrix to be loaded.

N1 = Number of rows of matrix A.

N2 = Number of columns of matrix A.

N3 = Format control indicator for the data.

The elements of the matrix are typed or punched on cards row-wise after the LOAD command according to one of the options determined by the value of N3:

a. if N3= 1 or blank, the format is (12F6.0). A maximum of twelve numbers per line, using as many lines as required to write the entire matrix must be used. If the decimal point is omitted, it is assumed to be at the extreme right of the field.

b. if N3= 2, (6F12.0), 6 numbers per line in fields of 12 columns.

c. if N3= 3, (4F18.0), 4 numbers per line in fields of 24 columns.

d. if N3= 4, (3F24.0), 3 numbers per line in fields of 24 columns.

PRINT — Matrix Print:

A = Name of matrix to be printed.

N1= Number of lines, which follow the PRINT command, to be printed as a label for the matrix. Column 13 of the command card is used for carriage control in DSMIS only. A one in column 13 will cause the matrix to be printed on a separate output page.

REMARK — This operation causes the number of lines designated by N1, which follow the operation, to be printed as remarks.

STOP — This operation ends a run of problems and is the last command to be issued in a stack of problems. It can be used only once per run.

ZERO — Formation of a Null Matrix:

A = Name of null matrix.

	N1= Number of rows of matrix A.
	N2= Number of columns of matrix A.
ADD	Matrix Addition - [A] + [B] = [A]
	Matrix A is replaced by the sum of matrices A and B.
SUB	Matrix Subtraction - [A] - [B] = [A]
	Matrix A is replaced matrix A minus matrix B.
MULT	Matrix Multiplication - [A] • [B] = [C]
	The product of matrix A times matrix B is generated and defined by matrix C.
TRANS	Matrix Transpose - $[B] = [A]^T$
	The transpose of matrix A is generated and defined by matrix B.
INVERT	Matrix Inversion - $[A] = [A]^{-1}$
	Matrix A is replaced by the inverse of itself.
SYMINV	Symmetric Matrix Inversion - $[A] = [A]^{-1}$
	Symmetric matrix A is replaced by the inverse of itself.
STOSM	Store Submatrix in a Matrix:
	A = Name of large matrix.
	B = Name of submatrix to be stored.
	N1= Row number in large matrix of first element of submatrix.
	N2= Column number in large matrix of first element of submatrix.

Elements of A are replaced by elements of B in designated area.

STODG	Store Row Matrix on Diagonal of Matrix: A = Name of square matrix. B = Name of row matrix to be stored on diagonal of matrix A. Diagonal elements of A are replaced by elements of B.
SCALE	Scalar Multiplication - [A] = S1·[A]
ADDSM	Add Submatrix to a Large Matrix: A = Name of large matrix. B = Name of submatrix to be added into large matrix. N1= Row number in large matrix of first element of submatrix. N2= Column number in large matrix of first element of submatrix.
RMVSM	Extracts Submatrix [B] from [A]: A = Name of large matrix. B = Name of submatrix defined as matrix. N1= Row number in large matrix of first element of submatrix. N2= Column number in large matrix of first element of submatrix. N3= Number of rows in submatrix. N4= Number of columns in submatrix. Matrix A is not modified.

LOG — The Log of each Element

Each element in the matrix A is replaced by the natural log of the element.

MSCALE — Scalar Multiplication

Each element in the matrix A is replaced by B times the element where B has previously been defined as a 1 by 1 matrix.

RMVDG — Extracts Row Matrix B from Diagonal of Square matrix A.

A = Name of square matrix.

B = Name of row matrix composed of the diagonal elements of matrix A.

Matrix A is not modified.

EIGEN — Eigenvalues and Eigenvectors:

The eigenvalues and eigenvectors of the system, $[A][X]=\lambda[B][X]$, are determined where [A] is a symmetrical matrix and [B] is a diagonal matrix of positive elements stored as a row matrix.

A = Name of matrix A.

B = Name of matrix B.

C = Name of matrix of eigenvectors stored row-wise.

D = Name of row matrix of eigenvalues, λ.

N1= The number of eigenvectors to be calculated. The ordering of eigenvectors and eigenvalues is determined by the sign of N1 as follows:

a. If N1 is positive, eigenvalues are arranged in descending order.

b. If N1 is negative, eigenvalues are arranged in ascending order.

N2= Eigenvector normalizing.

a. If N2 is blank:
Eigenvectors are normalized so that $[C][B][C]^T = [I]$

b. If N2 is 1:
Eigenvalues are normalized so that $[A] = [C]^T [\ulcorner\lambda\lrcorner][C]$

SQREL Square Root of Each Element
Each element in the matrix A is replaced by the square root of itself.

INVEL Inversion of Each Element
Each element in the matrix A is replaced by the reciprocal of itself.

DELETE Matrix Deletion
The matrix A is eliminated from core storage.

DUPL — Matrix Duplication:

A = Name of matrix to be duplicated.

B = Name of matrix defined to be identical with matrix A.

ENVEL — Envelope Value of Matrix

The maximum absolute value in each row of matrix A is printed along with its column number times an interval.

A = Name of matrix A.

S1= Interval between columns.

RESPON — Numerical Integration of a Set of Differential Equations:

$$\{\ddot{x}\}+2\lambda[\lceil\omega\rfloor]\{\dot{x}\}+[\lceil\omega^2\rfloor]\{x\}=\{S\}P(t)$$

with initial conditions, $x(0)=\dot{x}(0)=0$.

A = Name of a row matrix containing the diagonal elements of matrix $[\lceil\omega\rfloor]$. The dimension of A must be (1xn) where n is the number of equations in the set.

B = Name of a column matrix containing the elements of {S}. The dimension of B must be (nx1).

C = Name of a row matrix containing the values of the function P(t) evaluated at equal intervals, Δt:

$$P(0), P(\Delta t), P(2\Delta t), \ldots, P(m\Delta t)$$

The dimension of C depends on the choice of Δt which must be small for accurate results. The size of Δt should be determined by actual numerical experimentation with the command.

D = Name of the output matrix containing the calculated values of the variables as shown in Figure 5.

N1= Output interval i. The values of the n unknowns, $\{x\}$, are printed at $i\Delta t, 2i\Delta t, 3i\Delta t, \ldots$; if i=1, all of the values are printed.

S1= Value of the damping ratio, λ.

S2= Time increment, Δt, used in the numerical integration.

$$\begin{bmatrix} x_1(i\Delta t), x_1(2i\Delta t), x_1(3i\Delta t), \ldots \\ x_2(i\Delta t), x_2(2i\Delta t), x_2(3i\Delta t), \ldots \\ \vdots \\ x_n(i\Delta t), x_n(2i\Delta t), x_n(3i\Delta t), \ldots \end{bmatrix}$$

Fig. 5. OUTPUT MATRIX, D, FOR RESPON

FUNGN Function Generation:

A = Name of matrix which defines a function in terms of line segments $X_1, Y_1; X_2, Y_2; \cdots; X_n, Y_n$. (Nx2)

B = Name of matrix to be formed which is composed of the y ordinates at equal x ordinate intervals of the function defined by matrix A.

N1= Number of y ordinates to be generated. The first ordinate will be equal to Y_1.

S1= X ordinate interval.

FORMK Forms a 2x2 Element Stiffness Matrix:

A = Name of matrix to be formed.

The line following the command is as follows:

a. Columns 1-12 Moment of inertia of member, I.

b. Columns 13-24 Effective shear area of member, $\bar{A}$.

c. Columns 25-36 Length of member, L.

d. Columns 37-48 Modulus of elasticity of member, E.

The shear modulus, G, is assumed to be equal to E/2.4.

	PUNCH	Punches Matrix onto Cards:
		A = Name of matrix to be punched.
		N1= Number of comment cards to be punched as a heading for matrix A.
		After the PUNCH command is issued, the next line contains the format elected by the user as follows:
		(NNTRR.SS)
		where
		NN= Number of elements per card.
		RR= Field width per element.
		SS= Number of significant digits.
		T = Type of data field. T must be a standard F or D Fortran format.
*	READ	Loads Matrices from file FT04F001
		The load operation card must have a 1. in S1. The last card must be a Continue Operation card.
*	CONT	Continue Operation
		S1= 1.
		Indicates that all matrices in file FT04F001 have been loaded into IMIS.
*	WRITE	Writes Matrix into File FT08F001:
		A = Name of matrix to be written.
		N1= Number of lines of remarks to be

written as a label for the matrix A.

N2 must be equal to 1.

PLOT

Plots One to Four Curves Described by Sets of Points Contained in Matrices A, B, C, D:

A = First matrix to be plotted on one graph. Plotted as a "+".

B = Second matrix to be plotted on one graph. Plotted as a "*".

C = Third matrix to be plotted on one graph. Plotted as a "O".

D = Fourth matrix to be plotted on one graph. Plotted as a ".".

* N3= 1. Offline print indicator. Graph will be stored in file FT08F001

N4= Number of lines of remarks to be printed at bottom of graph.

Matrices to be plotted do not have to have the same number of points. They have to be entered as N by 2 matrices with the first column the X co-ordinates and the second column the Y co-ordinates. When plotting more than one curve on the same graph, a conflict may arise if more than one curve passes

through the same point. The symbol printed at that point is determined by the following order: D, C, B and A.

POLFIT Determines the Coefficients of CUPOL:

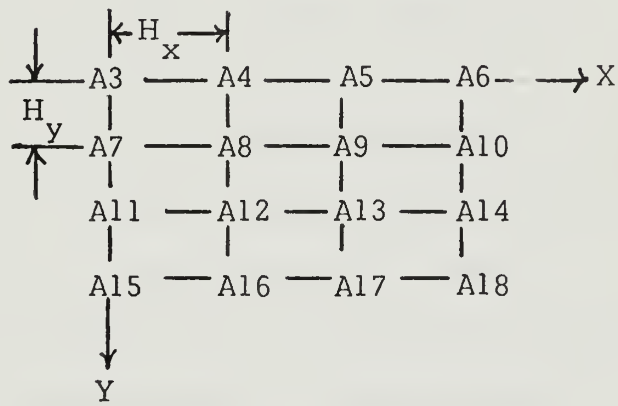
A = Input matrix. (18x1)

B = Output matrix of coefficients of CUPOL, $(C_1, C_2, \ldots, C_{16})$

where

$A(1) = H_x$

$A(2) = H_y$



and

$$CUPOL = C_1X^3Y^3 + C_2X^3Y^2 + C_3X^3Y + C_4X^3 +$$

$$C_5X^2Y^3 + C_6X^2Y^2 + C_7X^2Y + C_8X^2 +$$

$$C_9XY^3 + C_{10}XY^2 + C_{11}XY + C_{12}X +$$

$$C_{13}Y^3 + C_{14}Y^2 + C_{15}Y + C_{16}$$

CONTUR Contour Maps CUPOL:

N2= Number of lines of comments.

N3= Input type indicator.

a. N3= Blank or 1.

A = Input matrix. Same as for POLFIT command. (18x1)

B = Output matrix of coefficients of CUPOL. (16x1)

b. N3= 2

A = Input matrix. (20x1)

A(1) = Lower bound of X.

A(2) = Upper bound of X.

A(3) = Lower bound of Y.

A(4) = Upper bound of Y.

A(5) through A(20) are the coefficients of CUPOL.

* N4= Offline print indicator. Stores contour map in file FT09F001.

ITGRAL Integration of Functions Represented by Discrete Values or Polynomials:

A = Input matrix. (Rx1)

N3= Integration type indicator:

a. If N3= 1, Simpson's Rule for one dimension. R must be even.

$A(1) = H_x$

A(2) through A(R) are evenly spaced values of function.

b. If N3 = 2, Simpson's Rule for two dimensions.

$A(1) = H_x$

$A(2) = H_y$

A(3) through A(R) are values of function entered row-wise as for POLFIT.

$R = N^2+2$, N must be odd.

There must be the same number of values in both directions. H_x and H_y can be unequal.

c. If N3= 3, Gauss Quadrature for one dimension.

A(1) = Lower bound of X.

A(2) = Upper bound of X.

A(3) through A(14) are the coefficients of an 11th degree polynomial:

$$A_3X^{11}+A_4X^{10}+A_5X^9+A_6X^8+A_7X^7+A_8X^6+$$

$$A_9X^5+A_{10}X^4+A_{11}X^3+A_{12}X^2+A_{13}X+A_{14}$$

d. If N3 = 4, Gauss Quadrature for two dimensions.

A(1) = Lower bound of X.

A(2) = Upper bound of X.

A(3) = Lower bound of Y.

A(4) = Upper bound of Y.

A(5) through A(20) are coefficients of CUPOL.

IV. LIMITATIONS

A. SIZE AND NUMBER OF MATRICES

The storage available for all of the matrices in any problem is 80,000 bytes or 10,000 double words. This limit is arbitrary and controlled only by the total core storage available at any installation. For the naval Postgraduate School installation, 10,000 double words was selected in order to reduce turn around time in batch processing and reduce the size of the data set required for time sharing. The total number of matrices which can be stored is 100.

B. MATRIX NAMES

Each matrix is referred to by its name composed of one to six alphameric characters including blanks. The name is selected by the user when the matrix is first loaded or generated by an operation. If the name for a previously defined matrix is used later as a name of a new matrix, the former matrix is deleted before the operation is executed and is replaced by the new matrix. This is done so that there is only one matrix with a certain name at any one time.

C. PLOT

There are 200 points in the Y direction on the graph. The X direction has 130 points for the remote terminal printing and 160 points for the offline printer. The difference is due to the different number of lines per inch. The accuracy of the position of a point depends on the range in both the X and Y direction. The maximum error in the Y direction is the range of Y divided by 200. The maximum error in the X direction is

the range of X divided by either 130 for the remote terminal or 160 for the offline printer. If the X and Y ordinates of two or more points fall on the same nodal point on the graph, only one point will be plotted. If the points are from different matrices, the matrix hierarchy (see page 34) determines which will be plotted.

D. INTEGRATION

Integration by Simpson's Rule is restricted to equally spaced points in any one direction. The error is of the order of H_x^2 for integration in one dimension and of the order of H_x^2 times H_y^2 for two dimensions. H_x does not have to equal H_y but there has to be an odd number of points in any one direction. Integration by Gauss Quadrature for six Gauss ordinates is exact for all polynomials of degree eleven or less in one dimension and for CUPOL in two dimensions [Ref. 11]

E. POLFIT AND CONTUR

The polynomial, CUPOL, will pass through the values of the function at the nodal points of the grid but may not be a true representation in the space between nodal points. If the range of values of the function is less than 10^{-9}, a contour map will not be printed.

V. HOW TO CALL THE MATRIX INTERPRETIVE SYSTEM

A. TIME SHARING (IMIS VERSION)

At the Naval Postgraduate School, IMIS is stored in a library (AED) with the name, LIT, so that any number of users can use the system at any one time. In order to call LIT, it must be called as a subroutine. A Fortran program called IMIS has to be written as follows:

```
CALL LIT
STOP
END
```

There are two other libraries which must be loaded along with AED library in order to use LIT. This can be done with an executive program called SMIS as follows:

```
FORTRAN IMIS
GLOBAL LOADER TXTLIB AEDLIB SYSLIB
LOAD IMIS
START
```

SMIS is then executed with the command, $ SMIS. The computer will then type out the executive program with each command preceded by a time. Next it will type

EXECUTION BEGINS ...

At this time you are in IMIS and ready to begin the first problem. The first command will be START.

B. BATCH PROCESSING (DSMIS VERSION)

The first card must be a job card for the batch processing system as follows:

//aaaznnnn$_b$JOB(nnnn,ssssFP,TTTT), 'IDENTIFICATION'

where

aaa = The first three letters of the user's last name.

z = Job sequence number

nnnn= User's number.

ssss= Project number.

TTTT= Student section or faculty number.

The next six cards are control cards. Card five is used only when using the punch command. The cards are as follows stating in column one:

//JOBLIB$_b$DD$_b$DSNAME=DSMIS.F59CI,UNIT=2314,DISP=(OLD,KEEP),

//bbbbbbbbbbbbVOLUME=SER=MARY

//$_{bb}$EXEC$_b$PGM=DSMIS,REGION=160K

//FT06F001$_b$DD$_b$SYSOUT=A

//FT07F001$_b$DD$_b$SYSOUT=B

//FT05F001$_b$DD$_b$*

The "$_b$" is where blanks are left.

The problem deck follows with the first command a START command and the last a STOP command. The last card of the deck is a standard delimiter card:

/*

On the job request card, check the item marked DISK PACK under the SPECIAL INPUT heading and write in "MARY". If the punch command is being used, under the heading SPECIAL OUTPUT check the item marked PUNCHED CARDS.

VI. SAMPLE PROBLEMS

A. THE EIGEN SYSTEM

The matrix equation

$$[A]\{X\} = \lambda [B]\{X\}$$

where

$$[A] = \begin{bmatrix} 0.0 & -1.0 & 0.0 \\ -1.0 & 2.0 & -1.0 \\ 0.0 & -1.0 & 2.0 \end{bmatrix}$$

and

$$[B] = \begin{bmatrix} 1.5 & 0.0 & 0.0 \\ 0.0 & 2.0 & 0.0 \\ 0.0 & 0.0 & 2.5 \end{bmatrix}$$

is to be solved. The entire sequence of events is as shown below. Everything written in lower case was typed by the operator. The computer response is in upper case.

```
start

    START
load  a                               3     3
    LOAD  A
    3 ROWS,     3 COLUMNS
0.0   -1.0   0.0  -1.0  2.0   -1.0  0.0   -1.0   2.0
load  b                               1     3
    LOAD  B
    1 ROWS,     3 COLUMNS
1.5   2.0   2.5
eigen a     b     c     d             3     1
    EIGEN A     B     C     D
    3
print c
    PRINT C


          1             2             3
      1  -3.774271D-01 1.136139D 00-8.056459D-01
      2  -4.626117D-01 5.490667D-01 1.329088D 00
      3   1.069364D 00 6.385240D-01 2.906204D-01


print d
    PRINT D


          1             2             3
      1   1.505111D 00 5.934423D-01-2.985532D-01
```

```
trans c      ct
    TRANS C      CT
zero  x
    ZERO  X
    3 ROWS,      3 COLUMNS
stodg x      d
    STODG X      D
mult  ct     x      ctx
    MULT  CT     X      CTX
mult  ctx    c      a
    MULT  CTX    C      A
print a
    PRINT A


              1             2             3
    1  -1.471323D-13-1.000000D 00 1.902783D-13
    2  -1.000000D 00 2.000000D 00-1.000000D 00
    3   1.902228D-13-1.000000D 00 2.000000D 00


stop
    STOP
```

B. CONTOUR MAP

A contour map for a surface described by the ordinates at the sixteen nodal points of the grid shown on page 47 is desired.

$$[A] = \begin{bmatrix} 10.0 & 12.0 & 8.0 & 10.0 \\ 12.0 & 11.0 & 9.0 & 8.0 \\ 8.0 & 9.0 & 11.0 & 12.0 \\ 10.0 & 8.0 & 12.0 & 10.0 \end{bmatrix}$$

and

$H_x=H_y=1$

The results are as follows:

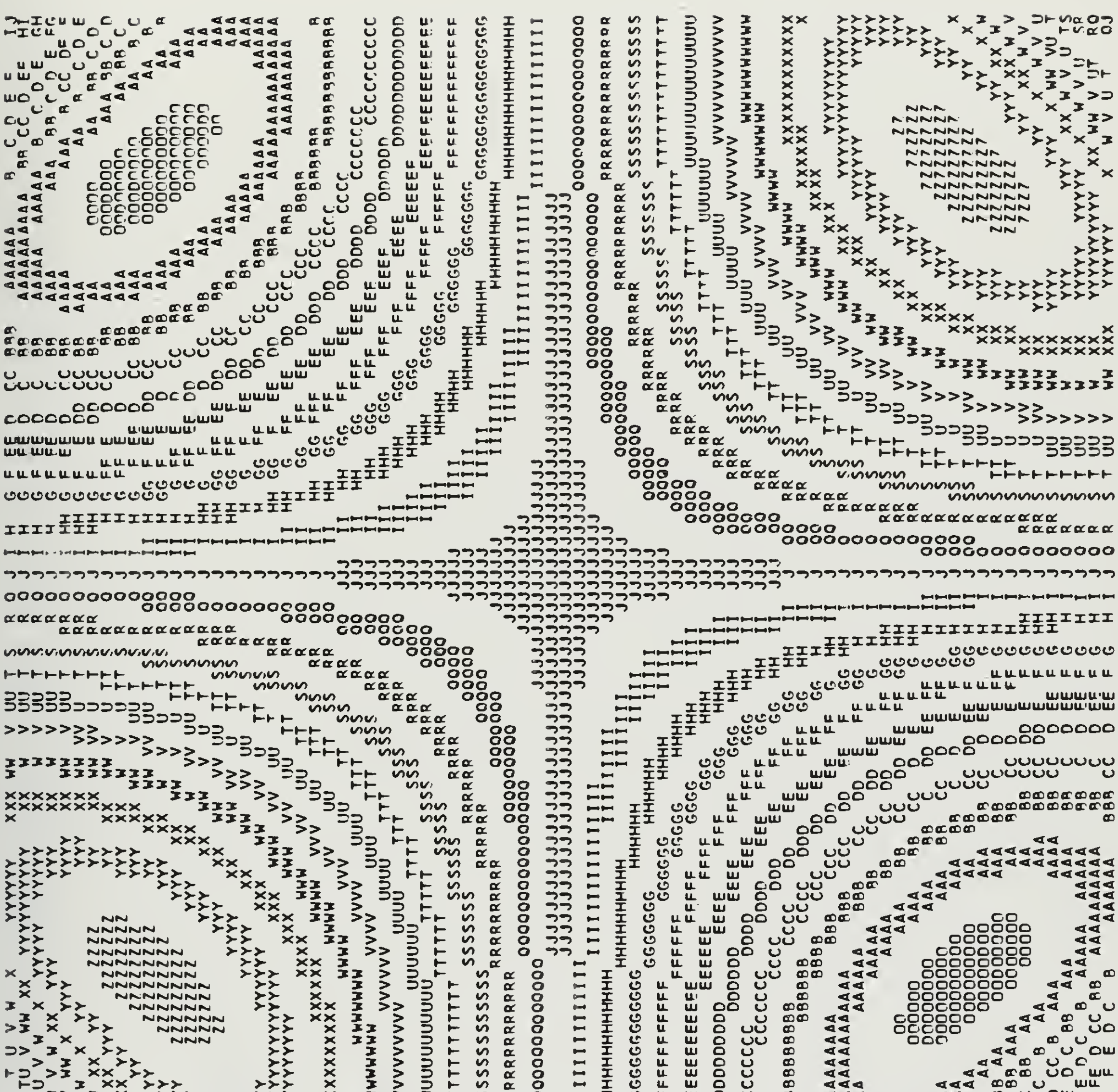
```
start

    START
load  a                                   18    1
    LOAD  A
   18 ROWS,      1 COLUMNS
1.     1.     10.    12.    8.     10.    12.     11.  9.     8.    8.      9.
11.    12.    10.    8.     12.    10.

contura     b
    CONTURA     B
```

THIS IS THE FIRST PAGE OUTPUT USING GRID POINTS AS INPUT.

```
U1-------U2--------U3---------U4
|        |         |          |
|        |         |          |
|        |         |          |
U5-------U6--------U7---------U8
|        |         |          |
|        |         |          |
|        |         |          |
U9-------U10-------U11--------U12
|        |         |          |
|        |         |          |
|        |         |          |
U13------U14-------U15--------U16
```

```
U1 =   10.000000 U2 =   12.000000 U3 =    8.000000 U4 =   10.000000
U5 =   12.000000 U6 =   11.000000 U7 =    9.000000 U8 =    8.000000
U9 =    8.000000 U10=    9.000000 U11=   11.000000 U12=   12.000000
U13=   10.000000 U14=    8.000000 U15=   12.000000 U16=   10.000000
```

```
O=   7.0814463545      RANGE(   0.708145D 01        0.129186D 02 )
A=       7.3733     B=       7.6652     C=       7.9570     D=       8.2489     E=       8.5407
F=       8.8326     G=       9.1244     H=       9.4163     I=       9.7081     J=      10.0000
Q=      10.2919     R=      10.5837     S=      10.8756     T=      11.1674     U=      11.4593
V=      11.7511     W=      12.0430     X=      12.3348     Y=      12.6267     Z=      12.9186
```



```
punch b
    PUNCH B
(4f15.8)


stop
    STOP
```

The command

offline punch file ft07f001

will cause the matrix B to be punched onto cards with the indicated format of four numbers per card with a field width of 15 and 8 significant figures.

If the input for the contour map had been the coefficients of CUPOL, the first page of output would have been as follows:

```
THIS IS THE FIRST PAGE OUTPUT COEFFICIENTS AS INPUT.



 -0.333300 X3Y3     1.500000 X3Y2    -2.833000 X3Y      2.000000 X3
  1.500000 X2Y3    -6.750000 X2Y2    12.750000 X2Y     -9.000000 X2
 -2.833000 XY3     12.750000 XY2    -18.750000 XY       9.000000 X
  2.000000 Y3      -9.000000 Y2       9.000000 Y       10.000000



O=   7.0837699378       RANGE(   0.708377D 01          0.129594D 02 )
A=       7.3776    B=       7.6713    C=       7.9651    D=       8.2589    E=       8.5527
F=       8.8465    G=       9.1402    H=       9.4340    I=       9.7278    J=      10.0216
Q=      10.3154    R=      10.6092    S=      10.9029    T=      11.1967    U=      11.4905
V=      11.7843    W=      12.0781    X=      12.3719    Y=      12.6656    Z=      12.9594
```

C. GRAPH

A plot of the function

$$x = e^{-15t} \sin(54.99t)$$

is desired from t = 0.0 to t = 0.3. The values were calculated in a separate program and the values of X were punched onto cards. They were prepared for offline reading into file FT04F001.

The first card is a standard CP67 offline read card as follows starting in column one.

$CP67USERID_{bb}GUxxssss_{bbbbbb}TTTT$

where

xx = Terminal number user will be using.

ssss= User's account number.

TTTT= User's name.

The second card is as follows:

$OFFLINE_{b}READ_{b}FILE_{b}FT04F001$

The next card is a LOAD command card with a 1. in S1.

```
LOAD A                 80   2                     1.
```

The matrix is punched onto cards row-wise with the same format as one of the options under IMIS and these cards follow the LOAD card. As many matrices as desired may be loaded in this manner. The last card is a continue card as follows:

```
CONT                                              1.
```

This card indicates that all matrices have been read from file FT04F001. The deck is then loaded into the computer.

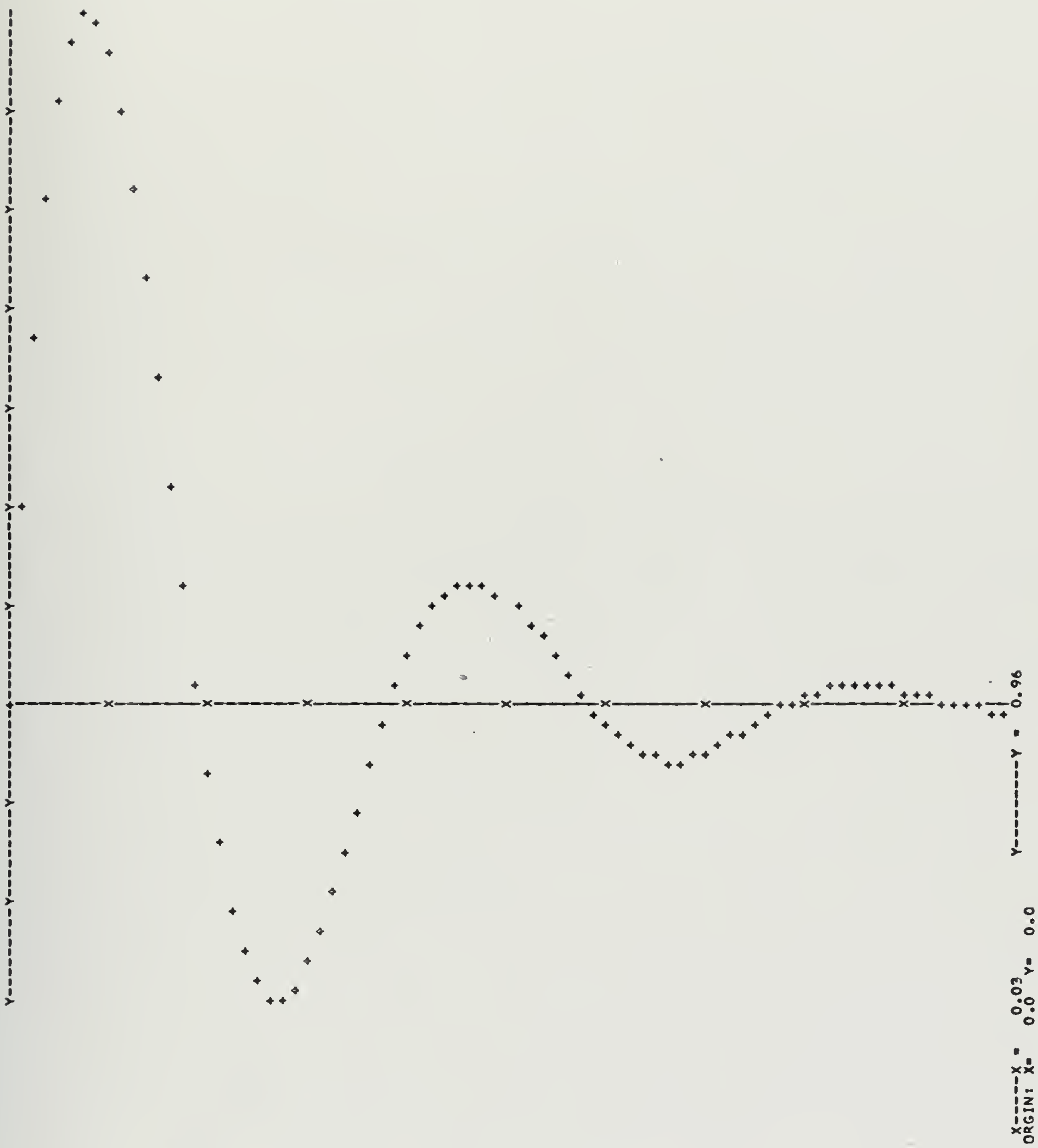
After logging onto the time sharing system, the command to read the file is as follows:

offline read *

OFFLINE READ READ FILE FT04F001

IMIS is then executed and the sequence is as follows:

```
start

    START
read


    LOAD  A
   80 ROWS,      2 COLUMNS
write a                           1
    WRITE A
plot  a
    PLOT  A
    .
```


X-----X = 0.03
Y-------Y = 0.96
ORGIN: X= 0.0 Y= 0.0

$\text{stop}_{\text{STOP}}$

The command

```
OFFLINE PRINTCC FILE FT08F001
```

will cause the matrix A to be printed on the offline printer.

VII. IMPROVEMENTS

This system is an open ended system, i.e., any matrix operation with fewer than four matrix arguments, four integer arguments and two scalar arguments can be coded as a new command and added to the system. As coded this program is sequential. Each command is executed once after it is read, no facilities for automatic repetitive execution of one or more commands is included. Such a feature would be very useful if it could be implemented without introducing extraneous complexities to the language.

APPENDIX A COMPUTER PROGRAM

```
C     MAIN PROGRAM
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON        XNAME(100),A(10000),W1(80),W2(80),W3(80),W4(80)
     1              ,NSIZE,NUM,NX,NR,NC,NNR,NNC,NDUM,NROW(100),NCOL(100)
     2,NN(100),NW1(80)
C
C     INITIALIZATION
C
      NSIZE=4
      WRITE (6,99)
      NUM=0
      CALL SMIS1
   99 FORMAT (1H1 )
      STOP 1
      END


      SUBROUTINE MLIST(MM,*)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE TO LIST MATRIX
C
      REAL*8 NAME,MM
      COMMON         NAME(100),A(10000),W1(80),W2(80),W3(80),W4(80)
     1              ,NSIZE,NUM,NX,NR,NC,NNR,NNC,NDUM,NROW(100),NCOL(100)
     2,NN(100),NW1(80)
      I=1
   15 IF (NUM-I) 30,10,10
   10 IF (NAME(I)-MM) 20,25,20
   20 I=I+1
      GO TO 15
   25 CALL DELETE(MM,&5)
   30 NUM=NUM+1
      NROW(NUM)=NR
      NCOL(NUM)=NC
      NAME(NUM)=MM
      NN(NUM)=NSIZE
      NX=NSIZE
      NSIZE=NSIZE+NR*NC
      IF (NUM-101) 45,40,45
   40 CALL ERRER(6,&5)
   45 IF(10001-NSIZE) 50,60,60
   50 CALL ERRER(5,&5)
   60 RETURN
    5 RETURN 1
      END
```

```
      SUBROUTINE SMIS1
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON      XNAME(100),A(10000),W1(80),W2(80),W3(80),W4(80)
     1           ,NSIZE,NUM,NX,NR,NC,NNR,NNC,NDUM,NROW(100),NCOL(100)
     2,NN(100),NW1(80)
      REAL*8 NOPER,NSTART,LISTOP
      DIMENSION OPLIST(50),LISTOP(50),HED(10),FMT(10)
      EQUIVALENCE (OPLIST(1),LISTOP(1)),(OPER,NOPER),(NSTART,START)
C
C     LIST OF OPERATIONS
C
      DATA OPLIST/8HZZZZZZZ  ,8HPRINT   ,8HZERO    ,8HDELETE  ,8HDUPL
     1,8HADD     ,8HSUB     ,8HSCALE   ,8HMULT    ,8HINVERT  ,8HTRANS
     2,8HSTOSM   ,8HRMVSM   ,8HWRITE   ,8HREAD    ,8HSTART   ,8HREWIND
     3,8HEIGEN   ,8HRESPON  ,8HFUNGN   ,8HSQREL   ,8HINVEL   ,8HSTOP
     4,8HENVEL   ,8HADDSM   ,8HREMARK  ,8HSYMINV  ,8HFORMK   ,8HLOG
     5,8HSTODG   ,8HRMVDG   ,8HMSCALE  ,8HLOAD    ,8HCONT    ,8HPLOT
     6,8HITGRAL  ,8HCONTUR  ,8HPOLFIT  ,8HPUNCH   /,START/8HSTART   /,
     7XZR/8H        /
    9 CONTINUE
      OPER=XZR
      XA=XZR
      XB=XZR
      XC=XZR
      XD=XZR
      DO 1 I=1,80
      W1(I)=0.0D0
      W2(I)=0.0D0
      W3(I)=0.0D0
      W4(I)=0.0D0
    1 NW1(I)=0
      DO 2 I=1,10000
    2 A(I)=0.0D0
      DO 3 I=1,100
      NROW(I)=0
      NCOL(I)=0
      NN(I)=0
C
C     READ AND PRINT OF CONTROL INFORMATION
C
    3 XNAME(I)=XZR
      NUMOP=39
    4 CONTINUE
    5 READ(5,101)  OPER,XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2
      IF (NSTART-NOPER) 7,6,7
   27 READ(4,101) OPER,XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2
      IF(SCAL1-1.0) 28,6,28
   28 CALL ERRER(10,&5)
```

```
    6 WRITE(6,99)
    7 WRITE(6,100) OPER,XA,XB,XC,XD
C
C     DETERMINATION OF OPERATION
C
      I=1
   10 IF(NUMOP-I) 20,30,30
   20 CALL ERRER(3,&5)
   30 IF (LISTOP(I)-NOPER) 40,50,40
   40 I=I+1
      GO TO 10
   50 GO TO ( 20,120,130,140,150,160,170,180,190,200,210,220,230,240,
     1  250,260,270,280,290,300,310,320,330,340,350,360,370,380,390,
     2  400,410,420,430,440,450,460,470,480,490),I
C
C     MATRIX LOAD
C
  430 WRITE(6,102) NR,NC
      CALL MLIST(XA,&5)
      NA=NX
      CALL LOAD(A(NA),NR,NC,NNR,SCAL1,&5)
      IF(SCAL1.EQ.1.0) GO TO 27
      GO TO 5
C
C      CONTINUE OPERATION
C
  440 GO TO 5
C
C     MATRIX PRINT
C
  120 IF(XB.EQ.1.0D0) WRITE(6,99)
  123 WRITE(6,109)
      K=1
      GO TO 361
  121 CALL MFIND(XA,&5)
  126 NA=NX
      CALL APRINT(A(NA),NR,NC,NNR)
      WRITE(6,109)
      GO TO 5
C
C     MATRIX ZERO
C
  130 WRITE(6,102) NR,NC
      CALL MLIST(XA,&5)
      DO 135 I=NX,NSIZE
  135 A(I)=0.0D0
      GO TO 5
C
```

```
C       DELETE OPERATION                                                LIT01410
C                                                                       LIT01420
  140 CALL DELETE(XA,&5)                                                LIT01430
      GO TO 5                                                           LIT01440
C                                                                       LIT01450
C       DUPLICATION OF MATRIX                                           LIT01460
C                                                                       LIT01470
  150 CALL MFIND(XA,&5)                                                 LIT01480
      CALL MLIST(XB,&5)                                                 LIT01490
      NB=NX-1                                                           LIT01500
      CALL MFIND(XA,&5)                                                 LIT01510
      NA=NX-1                                                           LIT01520
      NS=NR*NC                                                          LIT01530
      DO 155 I=1,NS                                                     LIT01540
      J=I+NB                                                            LIT01550
      K=I+NA                                                            LIT01560
  155 A(J)=A(K)                                                         LIT01570
      GO TO 5                                                           LIT01580
C                                                                       LIT01590
C       ADDITION OR SUBTRACTION OF MATRICES                             LIT01600
C                                                                       LIT01610
  160 TAG=1.0                                                           LIT01620
      GO TO 171                                                         LIT01630
  170 TAG=-1.0                                                          LIT01640
  171 CALL MFIND(XA,&5)                                                 LIT01650
      NA=NX-1                                                           LIT01660
      NNR=NR                                                            LIT01670
      NNC=NC                                                            LIT01680
      CALL MFIND(XB,&5)                                                 LIT01690
      NB=NX-1                                                           LIT01700
      IF(NR-NNR) 173,172,173                                            LIT01710
  172 IF(NC-NNC) 173,174,173                                            LIT01720
  173 CALL ERRER(1,&5)                                                  LIT01730
  174 NS=NR*NC                                                          LIT01740
  175 DO 176 I=1,NS                                                     LIT01750
      J=I+NA                                                            LIT01760
      K=I+NB                                                            LIT01770
  176 A(J)=A(J)+TAG*A(K)                                                LIT01780
      GO TO 5                                                           LIT01790
C                                                                       LIT01800
C       SCALAR MULTIPLICATION                                           LIT01810
C                                                                       LIT01820
  180 WRITE(6,103) SCAL1                                                LIT01830
      CALL MFIND(XA,&5)                                                 LIT01840
      NA=NX                                                             LIT01850
      NB=NX-1+NR*NC                                                     LIT01860
      DO 185 I=NA,NB                                                    LIT01870
  185 A(I)=SCAL1*A(I)                                                   LIT01880
```

```
      GO TO 5
C
C     MATRIX MULTIPLICATION
C
  190 CALL MFIND(XA,&5)
      NRA=NR
      NCA=NC
      CALL MFIND(XB,&5)
      NRB=NR
      NCB=NC
      IF(NCA-NRB) 191,192,191
  191 CALL ERRER(1,&5)
  192 NR=NRA
      CALL MLIST(XC,&5)
      ND=NX
      CALL MFIND(XA,&5)
      NA=NX
      CALL MFIND(XB,&5)
      NB=NX
      CALL MULT(NRA,A(NA),NCA,A(NB),NCB,A(ND))
      GO TO 5
C
C     MATRIX INVERSION
C
  200 CALL MFIND(XA,&5)
      NA=NX
      IF(NR-NC) 201,202,201
  201 CALL ERRER(1,&5)
  202 CALL INVERT(A(NA),W3(1),NW1(1),NR)
      GO TO 5
C
C     MATRIX TRANSPOSE
C
  210 CALL MFIND(XA,&5)
      NRA=NR
      NR=NC
      NC=NRA
      CALL MLIST(XB,&5)
      NB=NX
      CALL MFIND(XA,&5)
      NA=NX
      CALL TRANS(A(NA),A(NB),NRA,NC)
      GO TO 5
C
C     STORE SUBMATRIX
C
  220 NTAG=1
  221 J=NR
```

```
      K=NC
      CALL MFIND(XB,&5)
      GO TO 235
C
C     REMOVE SUBMATRIX
C
  230 J=NR
      K=NC
      NR=NNR
      NC=NNC
      WRITE(6,102) NR,NC
      NTAG=0
      CALL MLIST(XB,&5)
  235 WRITE(6,104) J,K
      NB=NX
      NRB=NR
      NCB=NC
      CALL MFIND(XA,&5)
      NA=NX
      NRA=NR
      NCA=NC
      CALL STOSM(A(NA),A(NB),NRA,NCA,NRB,NCB,J,K,NTAG)
      GO TO 5
C
C     WRITE MATRIX IN FILE FT08F001
C
  240 IF(NNR-1) 122,124,122
  122 CALL ERRER(11,&5)
  124 WRITE(8,111) XA
      GO TO 123
C
C     READ MATRIX FROM FILE FT04F001
C
  250 GO TO 27
C
C     START
C
  260 NSIZE=4
      NUM=0
      GO TO 9
C
C     REWIND TAPE NR
C
  270 CALL ERRER(9,&5)
C
C     EIGENVALUES AND EIGENVECTORS
C
  280 M=NR
```

```
      N2=NC
      CALL MFIND(XA,&5)
      IF (NR-NC) 281,282,281
  281 CALL ERRER(1,&5)
  282 N=NR
      NEV=N
      NR=1
      CALL MLIST(XD,&5)
      NR=IABS (M)
      CALL MLIST(XC,&5)
      NE=NX
      CALL MFIND(XA,&5)
      NA=NX
      CALL MFIND(XB,&5)
      NB=NX
      CALL MFIND(XD,&5)
      ND=NX
      WRITE(6,107) M
      CALL EIG(A(NA),A(NB),A(NE),A(ND),NEV,M,N2,N)
      GO TO 5
C
C     RESPONSE CALCULATION
C
  290 N=NR
      WRITE(6,107) N,SCAL1,SCAL2
      CALL MFIND(XA,&5)
      M=NC
      CALL MFIND(XC,&5)
      NC=NC/N
      NR=M
      CALL MLIST(XD,&5)
      ND=NX
      CALL MFIND(XA,&5)
      NA=NX
      CALL MFIND(XB,&5)
      NB=NX
      CALL MFIND(XC,&5)
      NNC=NX
      NTAG=1
      IF(NR-1) 396,394,396
  394 NTAG=0
  396 L=NC
      CALL RESPON(A(NA),A(NB),A(NNC),A(ND),M,N,L,SCAL1,SCAL2,NTAG)
      GO TO 5
C
C     FUNCTION GENERATION
C
  300 N=NR
```

```
      WRITE(6,107) N,SCAL1
      NR=1
      NC=N
      CALL MLIST(XB,&5)
      K=NX
      NH=NX+N-1
      CALL MFIND(XA,&5)
      L=NX
      TIM =A(L)
      GO TO 302
  301 IF (A(L)-TIM) 302,302,306
  302 DT=A(L+2)-A(L)
      DP=A(L+3)-A(L+1)
      L=L+2
      IF (DT) 304,302,305
  304 CALL ERRER(1,&5)
  305 SLOPE=DP/DT
  306 A(K)=A(L-1)+(TIM -A(L-2))*SLOPE
  308 TIM =TIM +SCAL1
      K=K+1
        IF (NH-K) 5,301,301
C
C     SQUARE ROOT OF EACH ELEMENT
C
  310 CALL MFIND(XA,&5)
      NA=NX
      NB=NA+NR*NC-1
      DO 315 K=NA,NB
  315 A(K)=DSQRT(A(K))
      GO TO 5
C
C     INVERSION OF EACH ELEMENT
C
  320 CALL MFIND(XA,&5)
      NA=NX
      NB=NA+NR*NC-1
      DO 325 K=NA,NB
  325 A(K)=1./A(K)
      GO TO 5
  330 RETURN
C
C     ENVELOPE VALUES OF MATRIX
C
  340 WRITE(6,101) XB
      K=2
      GO TO 361
  341 CALL MFIND(XA,&5)
      NA=NX
```

```
      NR=NR                                                             LIT03810
      NC=NC                                                             LIT03820
      K=NA                                                              LIT03830
      DO 347 I=1,NR                                                     LIT03840
      AMAX=0.0                                                          LIT03850
      TIM =SCAL1                                                        LIT03860
      DO 346 J=1,NC                                                     LIT03870
      XA=DABS(A(K))                                                     LIT03880
      IF(AMAX-XA) 344,345,345                                           LIT03890
  344 AMAX=XA                                                           LIT03900
      TMAX=TIM                                                          LIT03910
  345 K=K+1                                                             LIT03920
  346 TIM =TIM +SCAL1                                                   LIT03930
  347 WRITE(6,108) I,TMAX,AMAX                                          LIT03940
       WRITE(6,101) XB                                                  LIT03950
      GO TO 5                                                           LIT03960
C                                                                       LIT03970
C     ADD SUBMATRIX                                                     LIT03980
C                                                                       LIT03990
  350 NTAG=-1                                                           LIT04000
      GO TO 221                                                         LIT04010
C                                                                       LIT04020
C     READ AND PRINT OF REMARK CARDS                                    LIT04030
C                                                                       LIT04040
  360 K=3                                                               LIT04050
  361 IF(NR) 366,366,364                                                LIT04060
  364 IF(NNR.EQ.1) GO TO 367                                            LIT04070
      DO 365 I=1,NR                                                     LIT04080
      READ(5,105) HED                                                   LIT04090
  365 WRITE(6,110) HED                                                  LIT04100
  366 GO TO (121,341,5),K                                               LIT04110
  367 DO 368 I=1,NR                                                     LIT04120
      READ(5,105) HED                                                   LIT04130
  368 WRITE(8,110) HED                                                  LIT04140
      GO TO 366                                                         LIT04150
C                                                                       LIT04160
C     INVERSION OF SYMMETRICAL MATRIX                                   LIT04170
C                                                                       LIT04180
  370 CALL MFIND(XA,&5)                                                 LIT04190
      NA=NX                                                             LIT04200
      IF (NR-NC) 372,375,372                                            LIT04210
  372 CALL ERRER(1,&5)                                                  LIT04220
  375 CALL SYMINV(A(NA),NR,W1(1),W3(1))                                 LIT04230
      GO TO 5                                                           LIT04240
C                                                                       LIT04250
C     FORM 2X2 MATRIX                                                   LIT04260
C                                                                       LIT04270
  380 NR=2                                                              LIT04280
```

```
      NC=2                                                              LIT04290
      CALL MLIST(XA,&5)                                                 LIT04300
      NA=NX                                                             LIT04310
      CALL FORMK(A(NA))                                                 LIT04320
      GO TO 5                                                           LIT04330
C                                                                       LIT04340
C     TAKES A LOG OF A MATRIX                                           LIT04350
C                                                                       LIT04360
  390 CALL MFIND(XA,&5)                                                 LIT04370
      IH=NX-1+NR*NC                                                     LIT04380
      IL=NX                                                             LIT04390
      DO 395 I=IL,IH                                                    LIT04400
  395 A(I)= DLOG(A(I))                                                  LIT04410
      GO TO 5                                                           LIT04420
C                                                                       LIT04430
C     STORES ROW (B) ON DIAGONAL OF (A)                                 LIT04440
C                                                                       LIT04450
  400 CALL MFIND(XA,&5)                                                 LIT04460
      K=NX                                                              LIT04470
      CALL MFIND(XB,&5)                                                 LIT04480
      IL=NX                                                             LIT04490
      IH=NX-1+NC                                                        LIT04500
      DO 405 I=IL,IH                                                    LIT04510
      A(K)=A(I)                                                         LIT04520
  405 K=K+NC+1                                                          LIT04530
      GO TO 5                                                           LIT04540
C                                                                       LIT04550
C     REMOVES ROW (B) FROM DIAGONAL OF (A)                              LIT04560
C                                                                       LIT04570
  410 CALL MFIND(XA,&5)                                                 LIT04580
      NR=1                                                              LIT04590
      CALL MLIST(XB,&5)                                                 LIT04600
      IL=NX                                                             LIT04610
      IH=NX-1+NC                                                        LIT04620
      CALL MFIND(XA,&5)                                                 LIT04630
      K=NX                                                              LIT04640
      DO 415 I=IL,IH                                                    LIT04650
      A(I)=A(K)                                                         LIT04660
  415 K=K+NC+1                                                          LIT04670
      GO TO 5                                                           LIT04680
C                                                                       LIT04690
C     MULTIPLIES A BY B(1,1)                                            LIT04700
C                                                                       LIT04710
  420 CALL MFIND(XB,&5)                                                 LIT04720
      I=NX                                                              LIT04730
      SCAL1=A(I)                                                        LIT04740
      GO TO 180                                                         LIT04750
C                                                                       LIT04760
```

```
C       PLOT OPERATION
C
  450 INR=NNR
      NAA=0.0
      NAB=0.0
      NAC=0.0
      NBA=0.0
      NBB=0.0
      NBC=0.0
      CALL MFIND(XA,&5)
      NA=NX
      NB=NA+NC*NR-1
      IF(XB.EQ.XZR) GO TO 451
      CALL MFIND(XB,&5)
      NAA=NX
      NBA=NAA+NC*NR-1
      IF(XC.EQ.XZR) GO TO 451
      CALL MFIND(XC,&5)
      NAB=NX
      NBB=NAB+NC*NR-1
      IF(XD.EQ.XZR) GO TO 451
      CALL MFIND(XD,&5)
      NAC=NX
      NBC=NAC+NC*NR-1
  451 CALL PLOT(NA,NB,NAA,NAB,NAC,NBA,NBB,NBC,INR)
      IF(NNC) 455,455,454
  454 IF(INR) 456,456,457
  456 DO 458 JJ=1,NNC
  458 READ(5,105) HED
      GO TO 5
  457 DO 459 JJ=1,NNC
      READ(5,105) HED
  459 WRITE(8,110) HED
  455 GO TO 5
C
C        INTEGRATION OPERATION
C
  460 CALL MFIND(XA,&5)
      NA=NX
      INR=NNR
      CALL ITGRAL(NA,INR)
      GO TO 5
C
C        CONTUR OPERATION
C
  470 NOC=NC
      IF(NNR.EQ.0) NNR=1
      IF(NNC.EQ.0) NNC=1
```

```
      NAR=NNR                                                           LIT05250
      NAC=NNC                                                           LIT05260
      IF(NOC.LT.1) GO TO 474                                            LIT05270
      DO 471 I=1,NOC                                                    LIT05280
      READ(5,105) HED                                                   LIT05290
      GO TO (471,472),NAC                                               LIT05300
  472 WRITE(8,110) HED                                                  LIT05310
  471 CONTINUE                                                          LIT05320
  474 CALL CONTUR(XA,XB,NAR,NAC,NOC)                                    LIT05330
      GO TO 5                                                           LIT05340
C                                                                       LIT05350
C       POLYNOMINAL FIT OPERATION                                       LIT05360
C                                                                       LIT05370
  480 CALL POLFIT(XA,XB,&5)                                             LIT05380
      GO TO 5                                                           LIT05390
C                                                                       LIT05400
C     PUNCH OPERATION                                                   LIT05410
C                                                                       LIT05420
  490 READ(5,105) FMT                                                   LIT05430
      IF(NR.EQ.0) GO TO 491                                             LIT05440
      DO 492 J=1,NR                                                     LIT05450
      READ(5,105) HED                                                   LIT05460
  492 WRITE(7,105) HED                                                  LIT05470
  491 CALL MFIND(XA,&5)                                                 LIT05480
      NA=NX                                                             LIT05490
      NB=NA-1+NR*NC                                                     LIT05500
      WRITE(7,FMT) (A(I),I=NA,NB)                                       LIT05510
      GO TO 5                                                           LIT05520
   99 FORMAT (1H1 )                                                     LIT05530
  100 FORMAT(5X,5A6)                                                    LIT05540
  101 FORMAT(5A6,4I6,2F6.0)                                             LIT05550
  102 FORMAT (1I6,6H ROWS, 1I6, 8H COLUMNS )                            LIT05560
  103 FORMAT (9H  SCALAR= 1PD15.7)                                      LIT05570
  104 FORMAT (12H  ROW NUMBER I4,18H,     COLUMN NUMBER I4)             LIT05580
  105 FORMAT(10A8)                                                      LIT05590
  106 FORMAT (11H  TAPE UNIT  I3)                                       LIT05600
  107 FORMAT(1I6,2F12.5)                                                LIT05610
  108 FORMAT (1I6,1F15.5,1PD18.7)                                       LIT05620
  109 FORMAT(1H0)                                                       LIT05630
  110 FORMAT(2X,10A8)                                                   LIT05640
  111 FORMAT(10X,1A6)                                                   LIT05650
      END                                                               LIT05660
```

```
      SUBROUTINE APRINT( A ,NR,NC,NNR)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(1),NCOL(8)
      DOUBLE PRECISION A
C
C     PRINT OPERATION
C
      IF(NNR.EQ.1) GO TO 60
      DO 50 M=1,NC,8
      NN=NC-M
      IF (NN-7) 40,40,35
   35 NN=7
   40 MM=NN+1
      DO 45 N=1,MM
   45 NCOL(N)=M-1+N
      WRITE(6,101) (NCOL(N),N=1,MM)
      DO 50 N=1,NR
      NL=M+(N-1)*NC
      NH=NL+NN
   50 WRITE(6,100) (N,(A(I),I=NL,NH))
      RETURN
   60 DO 61 M=1,NC,8
      NN=NC-M
      IF(NN-7) 63,63,62
   62 NN=7
   63 MM=NN+1
      DO 64 N=1,MM
   64 NCOL(N)=M-1+N
      WRITE(8,101) (NCOL(N),N=1,MM)
      DO 61 N=1,NR
      NL=M+(N-1)*NC
      NH=NL+NN
   61 WRITE(8,100) (N,(A(I),I=NL,NH))
      WRITE(8,102)
      RETURN
  100 FORMAT(2X,1I6,2X,8(1PD13.6     ))
  101 FORMAT(1H0,2X,8I13)
  102 FORMAT(1H0)
  103 FORMAT(T10,1A6)
      END

      SUBROUTINE ERRER(N,*)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 NSTART,NOPER
      COMMON       XNAME(100),A(10000),W1(80),W2(80),W3(80),W4(80)
     1            ,NSIZE,NUM,NX,NR,NC,NNR,NNC,NDUM,NROW(100),NCOL(100)
     2,NN(100),NW1(80)
```

```
      GO TO (1,2,3,4,5,6,7,8,9,10,11),N                                 LIT06130
    1 WRITE(6,101)                                                      LIT06140
      GO TO 100                                                         LIT06150
    2 WRITE(6,102)                                                      LIT06160
      GO TO 100                                                         LIT06170
    3 WRITE(6,103)                                                      LIT06180
      GO TO 100                                                         LIT06190
    4 WRITE(6,104)                                                      LIT06200
      GO TO 100                                                         LIT06210
    5 WRITE(6,105)                                                      LIT06220
      GO TO 100                                                         LIT06230
    6 WRITE(6,106)                                                      LIT06240
      GO TO 100                                                         LIT06250
    7 WRITE(6,107)                                                      LIT06260
      GO TO 100                                                         LIT06270
    8 WRITE(6,108)                                                      LIT06280
      GO TO 100                                                         LIT06290
    9 WRITE(6,109)                                                      LIT06300
      GO TO 100                                                         LIT06310
                                                                        LIT06320
   10 WRITE(6,110)                                                      LIT06330
      GO TO 100                                                         LIT06340
   11 WRITE(6,111)                                                      LIT06350
  100 RETURN 1                                                          LIT06360
   50 FORMAT (1A6)                                                      LIT06370
   99 FORMAT(1H1,4X,6H START)                                           LIT06380
  101 FORMAT (27H MATRICES INCOMPATIBLE      )                          LIT06390
  102 FORMAT (27H MATRIX UNDEFINED           )                          LIT06400
  103 FORMAT (27H OPERATION UNDEFINED        )                          LIT06410
  104 FORMAT (27H MATRIX PREVIOUSLY DEFINED  )                          LIT06420
  105 FORMAT (27H STORAGE EXCEEDED           )                          LIT06430
  106 FORMAT (27H OVER 100 MATRICES IN CORE  )                          LIT06440
  107 FORMAT(27H ERROR IN EIGEN SUBROUTINE    )                         LIT06450
  108 FORMAT(27H ERROR IN LOAD OPERATION     )                          LIT06460
  109 FORMAT (27H OPERATION NOT AVAILABLE    )                          LIT06470
  110 FORMAT (27H ERROR IN READ OPERATION    )                          LIT06480
  111 FORMAT (27H ERROR IN WRITE OPERATION   )                          LIT06490
      END                                                               LIT06500

      SUBROUTINE FORMK(A)                                               LIT06510
      IMPLICIT REAL*8(A-H,O-Z)                                          LIT06520
      DIMENSION A(1)                                                    LIT06530
      DOUBLE PRECISION A                                                LIT06540
      READ(5,100) XI,XA,XL,XE                                           LIT06550
      WRITE(6,200) XI,XA,XL,XE                                          LIT06560
      B=14.4*XI/(XA*XL**2)                                              LIT06570
      S=2.*XE*XI/((1.+2.*B)*XL)                                         LIT06580
```

```
      A(1)=S*(2.+B)                                                     LIT06590
      A(2)=S*(1.-B)                                                     LIT06600
      A(3)=A(2)                                                         LIT06610
      A(4)=A(1)                                                         LIT06620
      RETURN                                                            LIT06630
  100 FORMAT (4F12.1)                                                   LIT06640
  200 FORMAT (3H I=F12.1,5H,  A=F12.1,5H,  L=F12.1,5H,  E=F12.1)       LIT06650
      END                                                               LIT06660

      SUBROUTINE LOAD( A ,NR,NC,NFMT,SCAL1,*)                           LIT06670
      IMPLICIT REAL*8(A-H,O-Z)                                          LIT06680
      DIMENSION A(1)                                                    LIT06690
      DOUBLE PRECISION A                                                LIT06700
C                                                                       LIT06710
C     LOAD OPERATION                                                    LIT06720
C                                                                       LIT06730
      IF(NFMT.GT.4) CALL ERRER(8,&5)                                    LIT06740
      NS=NR*NC                                                          LIT06750
      IF(NFMT.EQ.0) NFMT=1                                              LIT06760
      IF(SCAL1.EQ.1.0)GO TO 14                                          LIT06770
      GO TO (10,11,12,13),NFMT                                          LIT06780
   10 READ(5,100) (A(I),I=1,NS)                                         LIT06790
      GO TO 15                                                          LIT06800
   11 READ(5,101) (A(I),I=1,NS)                                         LIT06810
      GO TO 15                                                          LIT06820
   12 READ(5,102) (A(I),I=1,NS)                                         LIT06830
   13 READ(5,103) (A(I),I=1,NS)                                         LIT06840
      GO TO 15                                                          LIT06850
   14 GO TO(20,21,22,23),NFMT                                           LIT06860
   20 READ(4,100) (A(I),I=1,NS)                                         LIT06870
      GO TO 15                                                          LIT06880
   21 READ(4,101) (A(I),I=1,NS)                                         LIT06890
      GO TO 15                                                          LIT06900
   22 READ(4,102) (A(I),I=1,NS)                                         LIT06910
      GO TO 15                                                          LIT06920
   23 READ(4,103) (A(I),I=1,NS)                                         LIT06930
   15 RETURN                                                            LIT06940
    5 RETURN 1                                                          LIT06950
  100 FORMAT (12F6.0)                                                   LIT06960
  101 FORMAT(6F12.0)                                                    LIT06970
  102 FORMAT(4F18.0)                                                    LIT06980
  103 FORMAT(3F24.0)                                                    LIT06990
      END                                                               LIT07000
```

```
      SUBROUTINE MFIND(MM,*)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON        NAME(100),A(10000),W1(80),W2(80),W3(80),W4(80)
     1            ,NSIZE,NUM,NX,NR,NC,NNR,NNC,NDUM,NROW(100),NCOL(100)
     2,NN(100),NW1(80)
C
C     SUBROUTINE TO FIND LOCATION AND SIZE OF MATRIX XX
C
      REAL*8 NAME,MM
      I=1
   15 IF((NUM-I).LT.0) CALL ERRER(2,&5)
   10 IF ( NAME(I)-MM) 20,30,20
   20 I=I+1
      GO TO 15
   30 NR=NROW(I)
      NC=NCOL(I)
      NX=NN(I)
      RETURN
    5 RETURN 1
      END


      SUBROUTINE DELETE(MM,*)
      IMPLICIT REAL*8(A-H,O-Z)
C
C      SUBROUTINE TO DELETE MATRIX MM
C
      REAL*8 NAME,MM
      COMMON        NAME(100),A(10000),W1(80),W2(80),W3(80),W4(80)
     1            ,NSIZE,NUM,NX,NR,NC,NNR,NNC,NDUM,NROW(100),NCOL(100)
     2,NN(100),NW1(80)
      I=1
   15 IF (NUM-I) 40,10,10
   40 CALL ERRER(2,&5)
   10 IF ( NAME(I)-MM) 20,30,20
   20 I=I+1
      GO TO 15
   30 NX=NN(I)
      NUM=NUM-1
      NS=NROW(I)*NCOL(I)
      NSIZE=NSIZE-NS
      DO 50 J=I,NUM
      NAME(J)=NAME(J+1)
      NROW(J)=NROW(J+1)
      NCOL(J)=NCOL(J+1)
   50 NN(J)=NN(J+1)-NS
      DO 60 J=NX,NSIZE
      I=J+NS
```

```
   60 A(J)=A(I)                                         LIT07470
       RETURN                                           LIT07480
    5 RETURN 1                                          LIT07490
      END                                               LIT07500

      SUBROUTINE MULT(NRA,A,NCA,B,NCB,C)                LIT07510
      IMPLICIT REAL*8(A-H,O-Z)                          LIT07520
      DIMENSION A(1),B(1),C(1)                          LIT07530
      DOUBLE PRECISION A,B,C                            LIT07540
C                                                       LIT07550
C       MATRIX MULTIPLICATION                           LIT07560
C                                                       LIT07570
      K=1                                               LIT07580
      DO 200 I=1,NRA                                    LIT07590
      NN=(I-1)*NCA+1                                    LIT07600
      NH=NN-1+NCA                                       LIT07610
      DO 200 J=1,NCB                                    LIT07620
      MM=J                                              LIT07630
      C(K)=0.0D0                                        LIT07640
      DO 100 L=NN,NH                                    LIT07650
      C(K)=C(K)+A(L)*B(MM)                              LIT07660
  100 MM=MM+NCB                                         LIT07670
  200 K=K+1                                             LIT07680
      RETURN                                            LIT07690
      END                                               LIT07700

      SUBROUTINE TRANS(A,B,NR,NC)                       LIT07710
      IMPLICIT REAL*8(A-H,O-Z)                          LIT07720
      DIMENSION A(1),B(1)                               LIT07730
      DOUBLE PRECISION A,B                              LIT07740
C                                                       LIT07750
C      MATRIX TRANSPOSE                                 LIT07760
C                                                       LIT07770
      K=1                                               LIT07780
      DO 100 I=1,NC                                     LIT07790
      L=I                                               LIT07800
      DO 100 J=1,NR                                     LIT07810
      B(K)=A(L)                                         LIT07820
      L=L+NC                                            LIT07830
  100 K=K+1                                             LIT07840
      RETURN                                            LIT07850
      END                                               LIT07860
```

```
      SUBROUTINE INVERT(A,C,M,N)                                        LIT07870
      IMPLICIT REAL*8(A-H,O-Z)                                          LIT07880
      DIMENSION A(1),C(1),M(1)                                          LIT07890
      DOUBLE PRECISION A,C                                              LIT07900
C                                                                       LIT07910
C             GENERAL MATRIX INVERSION SUBROUTINE                       LIT07920
C                                                                       LIT07930
      DO 90 I=1,N                                                       LIT07940
   90 M(I)=-I                                                           LIT07950
      DO 140 I=1,N                                                      LIT07960
C                                                                       LIT07970
C     LOCATE LARGEST ELEMENT                                            LIT07980
C                                                                       LIT07990
      D=0.0D0                                                           LIT08000
      DO 112 L=1,N                                                      LIT08010
      IF (M(L)) 100,100,112                                             LIT08020
  100 J=L                                                               LIT08030
      DO 110 K=1,N                                                      LIT08040
      IF (M(K)) 103,103,108                                             LIT08050
  103 IF (DABS(D)-DABS(A(J))) 105,105,108                               LIT08060
  105 LD=L                                                              LIT08070
      KD=K                                                              LIT08080
      D=A(J)                                                            LIT08090
  108 J=J+N                                                             LIT08100
  110 CONTINUE                                                          LIT08110
  112 CONTINUE                                                          LIT08120
C                                                                       LIT08130
C     INTERCHANGE COLUMNS                                               LIT08140
C                                                                       LIT08150
      TEMP=-M(LD)                                                       LIT08160
      M(LD)=M(KD)                                                       LIT08170
      M(KD)=TEMP                                                        LIT08180
      L=LD                                                              LIT08190
      K=KD                                                              LIT08200
      DO 114 J=1,N                                                      LIT08210
      C(J)=A(L)                                                         LIT08220
      A(L)=A(K)                                                         LIT08230
      A(K)=C(J)                                                         LIT08240
      L=L+N                                                             LIT08250
  114 K=K+N                                                             LIT08260
C                                                                       LIT08270
C     DIVIDE ROW BY LARGEST ELEMENT                                     LIT08280
C                                                                       LIT08290
      NR=(KD-1)*N+1                                                     LIT08300
      NH=NR+N-1                                                         LIT08310
      DO 115 K=NR,NH                                                    LIT08320
  115 A(K)=-A(K)/D                                                      LIT08330
C                                                                       LIT08340
```

```
C       REDUCE REMAINING ROWS AND COLUMNS
C
        L=1
        DO 135 J=1,N
        IF (J-KD) 130,125,130
  125   L=L+N
        GO TO 135
  130   DO 134 K=NR,NH
        A(L)=A(L)+C(J)*A(K)
  134   L=L+1
  135   CONTINUE
C
C       REDUCE COLUMN
C
        C(KD)=1.0D0
        J=KD
        DO 140 K=1,N
        A(J)=C(K)/D
  140   J=J+N
C
C       INTERCHANGE ROWS
C
        DO 200 I=1,N
        L=0
  150   L=L+1
        IF(M(L)-I) 150,160,150
  160   K=(L-1)*N+1
        J=(I-1)*N+1
        M(L)=M(I)
        M(I)=I
        DO 200 LL=1,N
        TEMP=A(K)
        A(K)=A(J)
        A(J)=TEMP
        J=J+1
  200   K=K+1
        RETURN
        END


        SUBROUTINE SYMINV(A,N,B,C)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION A(1),B(1),C(1)
        DOUBLE PRECISION A,B,C
C
C        SYMMETRICAL MATRIX INVERSION
C
        LP=N
```

```
      DO 10  I=2,LP
      DO 10  J=I,LP
      ICOL=J+LP*(I-2)
      IROW=I+(J-1)*LP-1
      IF(A(ICOL).NE.A(IROW))  WRITE(6,1000)
      A(ICOL)=0.5D0*(A(ICOL)+A(IROW))
   10 A(IROW)=A(ICOL)
 1000 FORMAT(/,54H *****************************************************
     1       /,54H WARNING THE INPUT MATRIX TO SYMINV WAS NOT SYMMETRIC
     2    /,54H ***************************************************** )
      DO 140  I=1,N
      NR=(I-1)*N
      DO 100  J=1,N
      K=NR+J
  100 B(J)=A(K)
      D=B(I)
      DO 110  J=1,N
  110 C(J)=-B(J)/D
      L=1
      DO 130  J=1,N
      M=L
      DO 120  K=J,N
      A(L)=A(L)+B(J)*C(K)
      A(M)=A(L)
      M=M+N
  120 L=L+1
  130 L=L+J
      C(I)=-1.0D0/D
      M=I
      DO 140  J=1,N
      K=NR+J
      A(K)=C(J)
      A(M)=C(J)
  140 M=M+N
      NS=N*N
      DO 150  J=1,NS
  150 A(J)=-A(J)
      RETURN
      END


      SUBROUTINE STOSM(A,B,NRA,NCA,NRB,NCB,NR,NC,NTAG,*)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(1),B(1)
      DOUBLE PRECISION A,B
C
C      SUBROUTINE TO STORE OR REMOVE A SUBMATRIX
C
```

```
      IF (NRA+1-NR-NRB) 60,50,50
   50 IF (NCA+1-NC-NCB) 60,70,70
   60 CALL ERRER(1,&5)
   70 DO 100 I=1,NRB
      K=(NC-1)+(NR+I-2)*NCA
      L=(I-1)*NCB
      DO 100 J=1,NCB
      N=K+J
      M=L+J
      IF (NTAG) 75,90,80
   75 A(N)=A(N)+B(M)
      GO TO 100
   80 A(N)=B(M)
      GO TO 100
   90 B(M)=A(N)
  100 CONTINUE
      RETURN
    5 RETURN 1
      END


      SUBROUTINE RESPON(W,XM,P,X,NM,NT,L,DAMP,DT,NTAG)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION W(1),XM(1),P(1),X(1)
      DOUBLE PRECISION W,XM,P,X,DAMP,DT
C
C       RESPONSE PROGRAM
C
      K=1
      C1=DT/2.0D0
      C2=C1*DT/3.0D0
      C3=C2*2.0D0
      DO 160 N=1,NM
      NOUT=NT+1
      C4=W(N)**2
      C5=   DAMP*W(N)*2.0D0
      F=   C1*C5+C2*C4+1.0D0
      DISP=0.0D0
      VEL=0.0D0
      IL=L*(N-1)*NTAG+1
      ACEL=P(IL)*XM(N)
      IL=IL+1
      IH=IL+L-2
      DO 160 I=IL,IH
      A=VEL+C1*ACEL
      B=DISP+DT*VEL+C3*ACEL
      ACEL=(P(I)*XM(N)-C5*A-C4*B)/F
      VEL=A+C1*ACEL
```

```
      DISP=B+C2*ACEL
      IF(NOUT-I) 160,150,160
  150 X(K)=DISP
      K=K+1
      NOUT=NOUT+NT
  160 CONTINUE
      RETURN
      END


      SUBROUTINE EIG(R,XM,V,E,NEV,M,N2,NM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(1),XM(1),V(1),E(1)
      DOUBLE PRECISION R,XM,V,E
      LP=NM
      IF(NM.EQ.1) RETURN
      NVEC=IABS(M)
    1 DO 2 I=1,LP
    2 XM(I)=1.0D0/(DSQRT(XM(I)))
      DO 3 I=1,LP
      DO 3 J=1,LP
      JJ=I+(J-1)*NM
    3 R(JJ)=XM(I)*R(JJ)*XM(J)
      DO 104 I=2,LP
      DO 104 J=I,LP
      ICOL=J+LP*(I-2)
      IROW=I+(J-1)*LP-1
      IF(R(ICOL).NE.R(IROW))  WRITE(6,1000)
      R(ICOL)=0.5D0*(R(ICOL)+R(IROW))
  104 R(IROW)=R(ICOL)
 1000 FORMAT(/,54H ******************************************************
     1        /,54H WARNING THE INPUT MATRIX TO EIGEN WAS NOT SYMMETRIC
     2     /,54H ****************************************************** )
      CALL JACVAT(R,NM,1,E,V,NM)
      NEL=NVEC*NM
      DO 400 J=1,NVEC
      ANORM=0.0D0
      NL=1+(J-1)*NM
      NH=NL+NM-1
      DO 200 I=NL,NH
  200 ANORM=ANORM+V(I)*V(I)
      ANORM=DSQRT(ANORM)
      DO 300 I=NL,NH
  300 V(I)=V(I)/ANORM
  400 CONTINUE
C
C     MODIFY VECTORS
C
```

```
      DO 500 I=1,LP                                                     LIT10190
      DO 500 J=1,NVEC                                                   LIT10200
      JJ=I+(J-1)*NM                                                     LIT10210
      IF(N2.EQ.1) GO TO 450                                             LIT10220
      V(JJ)=V(JJ)*XM(I)                                                 LIT10230
      GO TO 500                                                         LIT10240
  450 V(JJ)=V(JJ)/XM(I)                                                 LIT10250
  500 CONTINUE                                                          LIT10260
      IF(M.GT.0) RETURN                                                 LIT10270
      NLIM=NEV/2                                                        LIT10280
      DO 600 I=1,NLIM                                                   LIT10290
      ATEMP=E(I)                                                        LIT10300
      II=NEV-I+1                                                        LIT10310
      E(I)=E(II)                                                        LIT10320
  600 E(II)=ATEMP                                                       LIT10330
      NLIM=NVEC/2                                                       LIT10340
      DO 700 I=1,NLIM                                                   LIT10350
      DO 700 J=1,NM                                                     LIT10360
      JJ=J+(I-1)*NM                                                     LIT10370
      JJJ=NEL-NM*I+J                                                    LIT10380
      ATEMP=V(JJ)                                                       LIT10390
      V(JJ)=V(JJJ)                                                      LIT10400
  700 V(JJJ)=ATEMP                                                      LIT10410
      RETURN                                                            LIT10420
      END                                                               LIT10430


      SUBROUTINE JACVAT(A,N,NOYES,EIVU,EIVR,NDIM)                       LIT10440
      IMPLICIT REAL*8(A-H,O-Z)                                          LIT10450
      DIMENSION A(NDIM,NDIM),EIVU(NDIM),EIVR(NDIM,NDIM)                 LIT10460
      IF(N-1)20,23,21                                                   LIT10470
   20 PRINT 22,N                                                        LIT10480
   22 FORMAT(4H N= ,I3,68H IS TOO SMALL.  LIMIT IS 1.  RETURN TO CALLINGLIT10490
     1 ROUTINE FRCM JACVAT. )                                           LIT10500
      RETURN                                                            LIT10510
   23 PRINT 24, A(1,1)                                                  LIT10520
   24 FORMAT (24H IN JACVAT , MATRIX A = ,E14.6)                        LIT10530
      RETURN                                                            LIT10540
   21 IF(N-160)1,1,3                                                    LIT10550
    3 PRINT 2,N                                                         LIT10560
    2 FORMAT(3H N=,I5,70H IS TOO LARGE.  LIMIT IS 160.  RETURN TO CALLI-LIT10570
     1NG ROUTINE FROM JACVAT. )                                         LIT10580
      RETURN                                                            LIT10590
    1 IF(NOYES)99,102,99                                                LIT10600
   99 CONTINUE                                                          LIT10610
      DO 101 J=1,N                                                      LIT10620
      DO 100 I=1,N                                                      LIT10630
  100 EIVR(I,J)=0.0                                                     LIT10640
```

```
101 EIVR(J,J)=1.0
102 ATOP=0.
    DO 112 J=1,N
    DO 111 I=1,J
    IF(A(I,J)-A(J,I))90,103,90
 90 PRINT 106,N,N
106 FORMAT(14H IN JACVAT (A(,I3,1H,,I3,3H)),)
    PRINT 108,I,J,J,I
108 FORMAT(3H A(,I3,1H,,I3,10H)  AND  A(,I3,1H,,I3,54H) WERE UNEQUAL,
   1SO THEY WERE REPLACED WITH THEIR MEAN )
    A(I,J)=.5*(A(I,J)+A(J,I))
    A(J,I)=A(I,J)
103 CONTINUE
    IF(ATOP-DABS(A(I,J)))104,111,111
104 ATOP=DABS(A(I,J))
111 CONTINUE
112 EIVU(J)=A(J,J)
    IF(ATOP)109,109,113
109 PRINT 110
110 FORMAT(26H IN JACVAT, MATRIX A  = 0 )
    RETURN
113 AVGF=DFLOAT(N*(N-1))*.55
    D=0.0
    DO 114 JJ=2,N
    DO 114 II=2,JJ
    S=A(II-1,JJ)/ATOP
114 D=S*S+D
    DSTOP=(1.D-08)*D
    THRSH =DSQRT (D/AVGF)*ATOP
115 IFLAG=0
    DO 130 JCOL=2,N
    JCOL1=JCOL-1
    DO 130 IROW=1,JCOL1
    AIJ=A(IROW,JCOL)
    IF(DABS(AIJ)-THRSH)130,130,117
117 AII=A(IROW,IROW)
    AJJ=A(JCOL,JCOL)
    S=AJJ-AII
    IF(DABS(AIJ)-1.D-09*DABS(S))130,130,118
118 IFLAG=1
    IF(1.D-10*DABS(AIJ)-DABS(S))116,119,119
119 S=.707106781
    C=S
    GO TO 120
116 T=AIJ/S
    S=0.25/DSQRT(0.25+T*T)
    C=DSQRT(0.5+S)
    S=2.*T*S/C
```

```
  120 DO 121 I=1,IROW
      T=A(I,IROW)
      U=A(I,JCOL)
      A(I,IROW)=C*T-S*U
  121 A(I,JCOL)=S*T+C*U
      I2=IROW+2
      IF(I2-JCOL)127,127,123
  127 CONTINUE
      DO 122 I=I2,JCOL
      T=A(I-1,JCOL)
      U=A(IROW,I-1)
      A(I-1,JCOL)=S*U+C*T
  122 A(IROW,I-1)=C*U-S*T
  123 A(JCOL,JCOL)=S*AIJ+C*AJJ
      A(IROW,IROW)=C*A(IROW,IROW)-S*(C*AIJ-S*AJJ)
      DO 124 J=JCOL,N
      T=A(IROW,J)
      U=A(JCOL,J)
      A(IROW,J)=C*T-S*U
  124 A(JCOL,J)=S*T+C*U
      IF(NOYES)131,126,131
  131 CONTINUE
      DO 125 I=1,N
      T=EIVR(I,IROW)
      EIVR(I,IROW)=C*T-EIVR(I,JCOL)*S
  125 EIVR(I,JCOL)=S*T+EIVR(I,JCOL)*C
  126 CONTINUE
      S=AIJ/ATOP
      D=D-S*S
      IF(D-DSTOP)1260,129,129
 1260 D=0.
      DO 128 JJ=2,N
      DO 128 II=2,JJ
      S=A(II-1,JJ)/ATOP
  128 D=S*S+D
      DSTOP=(1.D-08)*D
  129 THRSH=DSQRT(D/AVGF)*ATOP
  130 CONTINUE
      IF(IFLAG)115,134,115
  134 T=A(1,1)
      A(1,1)=EIVU(1)
      EIVU(1)=T
      DO 132 J=2,N
      T=A(J,J)
      A(J,J)=EIVU(J)
      EIVU(J)=T
      DO 132 I=2,J
  132 A(I-1,J)=A(J,I-1)
```

```
C   CRDER VECTORS AND VALUES
      DO 200 I=1,N
       DO 200 J=I,N
      IF(EIVU(I).GE.EIVU(J)) GO TO 200
      TEMP=EIVU(I)
      EIVU(I)=EIVU(J)
      EIVU(J)=TEMP
      DO 195 K=1,N
      TEMP=EIVR(K,I)
      EIVR(K,I)=EIVR(K,J)
  195 EIVR(K,J)=TEMP
  200 CONTINUE
  133 RETURN
      END


      SUBROUTINE PLOT(NA,NB,NAA,NAB,NAC,NBA,NBB,NBC,INR)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON          NAME(100),A(10000),W1(80),W2(80),W3(80),W4(80)
     1          ,NSIZE,NUM,NX,NR,NC,NNR,NNC,NDUM,NROW(100),NCOL(100)
     2,NN(100),NW1(80)
      REAL*8 NAME
      INTEGER*2 IGRID(101)
      INTEGER*4 ICHAR(11)/'  + ','   ',' - ',' | ',' Y ',' X ',
     *' A ',' B ',' * ',' O ',' . '/
      NNN=6
      KKK=65
      IF(INR.EQ.1) NNN=8
      IF(INR.EQ.1) KKK=81
      KKJ=KKK-1
      PNN=NNN
      XMAX=A(NA)
      XMIN=XMAX
      YMAX=A(NA+1)
      YMIN=YMAX
      IF(INR) 48,48,49
   49 WRITE(8,99)
   48 DO 10 K=NA,NB,2
      IF(A(K).GT.XMAX) XMAX=A(K)
      IF(A(K).LT.XMIN) XMIN=A(K)
      IF(A(K+1).GT.YMAX) YMAX=A(K+1)
   10 IF(A(K+1).LT.YMIN) YMIN=A(K+1)
      IF(NAA.EQ.0) GO TO 60
      DO 50 K=NAA,NBA,2
      IF(A(K).GT.XMAX) XMAX=A(K)
      IF(A(K).LT.XMIN) XMIN=A(K)
      IF(A(K+1).GT.YMAX) YMAX=A(K+1)
   50 IF(A(K+1).LT.YMIN) YMIN=A(K+1)
```

```
      IF(NAB.EQ.0) GO TO 60                                             LIT12070
      DO 51 K=NAB,NBB,2                                                 LIT12080
      IF(A(K).GT.XMAX) XMAX=A(K)                                        LIT12090
      IF(A(K).LT.XMIN) XMIN=A(K)                                        LIT12100
      IF(A(K+1).GT.YMAX) YMAX=A(K+1)                                    LIT12110
   51 IF(A(K+1).LT.YMIN) YMIN=A(K+1)                                    LIT12120
      IF(NAC.EQ.0) GO TO 60                                             LIT12130
      DO 52 K=NAC,NBC,2                                                 LIT12140
      IF(A(K).GT.XMAX) XMAX=A(K)                                        LIT12150
      IF(A(K).LT.XMIN) XMIN=A(K)                                        LIT12160
      IF(A(K+1).GT.YMAX) YMAX=A(K+1)                                    LIT12170
   52 IF(A(K+1).LT.YMIN) YMIN=A(K+1)                                    LIT12180
   60 XRANGE=XMAX-XMIN                                                  LIT12190
      IF(XRANGE.EQ.0.0) XRANGE=1.0                                      LIT12200
      YRANGE=YMAX-YMIN                                                  LIT12210
      IF(YRANGE.EQ.0.0) YRANGE=1.0                                      LIT12220
      XT=XMAX*XMIN                                                      LIT12230
      YT=YMAX*YMIN                                                      LIT12240
      IF(YT.LT.0.0) IXAX=100.0*(-YMIN)/YRANGE+1.5                       LIT12250
      IF(XT.LT.0.0) IYAX=KKJ- KKJ*XMAX/XRANGE+1.5                       LIT12260
      IF(XMIN.GE.0.0) IYAX=1                                            LIT12270
      IF(XMAX.LE.0.0) IYAX=KKK                                          LIT12280
      IF(YMIN.GE.0.0) IXAX=1                                            LIT12290
      IF(YMAX.LE.0.0) IXAX=101                                          LIT12300
      YINCR=YRANGE/10.0                                                 LIT12310
      XINCR=PNN*(XRANGE/KKJ)                                            LIT12320
      DO  11 LINE=1,KKK                                                 LIT12330
      DO 12 J=1,101                                                     LIT12340
   12 IGRID(J)=ICHAR(2)                                                 LIT12350
      IF(LINE-IYAX) 13,14,13                                            LIT12360
   14 DO 15 K=1,101                                                     LIT12370
   15 IGRID(K)=ICHAR(3)                                                 LIT12380
      DO 25 LL=1,10                                                     LIT12390
      LY=IXAX-10*LL                                                     LIT12400
      IF(LY.LT.0.0) GO TO 26                                            LIT12410
      IGRID(LY)=ICHAR(5)                                                LIT12420
      IF(XT.GT.0.0) IGRID(LY)=ICHAR(8)                                  LIT12430
   26 LY=IXAX+10*LL                                                     LIT12440
      IF(LY.GT.100) GO TO 25                                            LIT12450
      IGRID(LY)=ICHAR(5)                                                LIT12460
      IF(XT.GT.0.0) IGRID(LY)=ICHAR(8)                                  LIT12470
   25 CONTINUE                                                          LIT12480
   13 IGRID(IXAX)=ICHAR(4)                                              LIT12490
      DO 27 JK=1,10                                                     LIT12500
      JX=IYAX-PNN*JK                                                    LIT12510
      IF(JX.LT.0.0) GO TO 28                                            LIT12520
      IF(JX-LINE) 28,29,28                                              LIT12530
   29 IGRID(IXAX)=ICHAR(6)                                              LIT12540
```

```
      IF(YT.GT.0.0) IGRID(IXAX)=ICHAR(7)                                LIT12550
   28 JX=IYAX+PNN*JK                                                    LIT12560
      IF(JX.GT.KKJ) GO TO 27                                            LIT12570
      IF(JX-LINE) 27,30,27                                              LIT12580
   30 IGRID(IXAX)=ICHAR(6)                                              LIT12590
      IF(YT.GT.0.0) IGRID(IXAX)=ICHAR(7)                                LIT12600
   27 CONTINUE                                                          LIT12610
      DO 16 I=NA,NB,2                                                   LIT12620
      IPTX=KKJ-KKJ*(XMAX-A(I))/XRANGE+1.5                               LIT12630
      IF(IPTX.LT.1) IPTX=1                                              LIT12640
      IF(IPTX.GT.KKJ) IPTX=KKK                                          LIT12650
      IF(IPTX-LINE) 16,17,16                                            LIT12660
   17 IPTY=100.0*(A(I+1)-YMIN)/YRANGE+1.5                               LIT12670
      IGRID(IPTY)=ICHAR(1)                                              LIT12680
   16 CONTINUE                                                          LIT12690
      IF(NAA.EQ.0) GO TO 97                                             LIT12700
      DO 53 IA=NAA,NBA,2                                                LIT12710
      IPTX=KKJ-KKJ*(XMAX-A(IA))/XRANGE+1.5                              LIT12720
      IF(IPTX.LT.1) IPTX=1                                              LIT12730
      IF(IPTX.GT.KKJ) IPTX=KKK                                          LIT12740
      IF(IPTX-LINE) 53,67,53                                            LIT12750
   67 IPTY=100.0*(A(IA+1)-YMIN)/YRANGE+1.5                              LIT12760
      IGRID(IPTY)=ICHAR(9)                                              LIT12770
   53 CONTINUE                                                          LIT12780
      IF(NAB.EQ.0) GO TO 97                                             LIT12790
      DO 54 IB=NAB,NBB,2                                                LIT12800
      IPTX=KKJ-KKJ*(XMAX-A(IB))/XRANGE+1.5                              LIT12810
      IF(IPTX.LT.1) IPTX=1                                              LIT12820
      IF(IPTX.GT.KKJ) IPTX=KKK                                          LIT12830
      IF(IPTX-LINE) 54,66,54                                            LIT12840
   66 IPTY=100.0*(A(IB+1)-YMIN)/YRANGE+1.5                              LIT12850
      IGRID(IPTY)=ICHAR(10)                                             LIT12860
   54 CONTINUE                                                          LIT12870
      IF(NAC.EQ.0) GO TO 97                                             LIT12880
      DO 55 IC=NAC,NBC,2                                                LIT12890
      IPTX=KKJ-KKJ*(XMAX-A(IC))/XRANGE+1.5                              LIT12900
      IF(IPTX.LT.1) IPTX=1                                              LIT12910
      IF(IPTX.GT.KKJ) IPTX=KKK                                          LIT12920
      IF(IPTX-LINE) 55,68,55                                            LIT12930
   68 IPTY=100.0*(A(IC+1)-YMIN)/YRANGE+1.5                              LIT12940
      IGRID(IPTY)=ICHAR(11)                                             LIT12950
   55 CONTINUE                                                          LIT12960
   97 IF(INR) 42,42,41                                                  LIT12970
   42 WRITE(6,20) (IGRID(I),I=1,101)                                    LIT12980
      GO TO 11                                                          LIT12990
   41 WRITE(8,20) (IGRID(I),I=1,101)                                    LIT13000
   11 CONTINUE                                                          LIT13010
      IF(XT.LE.0.0) YYY=0.0                                             LIT13020
```

```
      IF(YT.LE.0.0) XXX=0.0
      IF(XMIN.GE.0.0) XXX=XMIN
      IF(YMAX.LE.0.0) YYY=YMAX
      IF(YMIN.GE.0.0) YYY=YMIN
      IF(XMAX.LE.0.0) XXX=XMIN+XRANGE+XINCR/PNN
      IF(INR) 43,43,44
   43 WRITE(6,22) XINCR,YINCR
      WRITE(6,23) XXX,YYY
      GO TO 45
   44 WRITE(8,22) XINCR,YINCR
      WRITE(8,23) XXX,YYY
   45 RETURN
   20 FORMAT(15X,101A1)
   99 FORMAT(1H1)
   22 FORMAT(12H  X-----X = ,F6.2,10X,15H  Y---------Y = ,F6.2)
   23 FORMAT(11H ORGIN: X= ,F6.2,3H Y=,F6.2)
      END


      SUBROUTINE ITGRAL(NA,INR)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON       XNAME(100),A(10000),W1(80),W2(80),W3(80),W4(80)
     1            ,NSIZE,NUM,NX,NR,NC,NNR,NNC,NDUM,NROW(100),NCOL(100)
     2,NN(100),NW1(80)
      DIMENSION C(16),X(6),W(6)
      SUM=0.0D0
      SUM2=0.0D0
      SUM4=0.0D0
      SUM8=0.0D0
      SUM16=0.0D0
      DO 9 I=1,16
    9 C(I)=0.0D0
      X(1)=-.932469514203152D0
      X(2)=-.661209386466264 5D0
      X(3)=-.238619186083196 9D0
      X(4)=-X(3)
      X(5)=-X(2)
      X(6)=-X(1)
      W(1)=.1713244923791703D0
      W(2)=.3607615730481386D0
      W(3)=.4679139345726910D0
      W(4)=W(3)
      W(5)=W(2)
      W(6)=W(1)
      GO TO (1,2,3,4),INR
    1 H=A(NA)
      XF=A(NA+1)
      XL=A(NA+NR-1)
```

```
      NV=NR-1
      NNV=NR-2
      IF(NV.LT.3) GO TO 23
      DO 20 I=3,NV,2
   20 SUM4=SUM4+A(NA+I-1)
      IF(NNV.LT.4) GO TO 23
      DO 21 I=4,NNV,2
   21 SUM2=SUM2+A(NA+I-1)
   23 SUM=(H/3.0)*(XF+XL+4.0*SUM4+2.0*SUM2)
      WRITE(6,61) SUM
      GO TO 99
    2 H=A(NA)
      HH=A(NA+1)
      IJ=NR-2
      ZZ=IJ
      IK=DSQRT(ZZ)
      NB=(IK-1)/2
      NC=NB-1
      ND=IK-2
      NE=ND-1
      NF=IK-3
      SUM=A(NA+2)+A(NA+IK+1)+A(NA+NR-IK)+A(NA+NR-1)
      DO 31 I=1,ND,2
   31 SUM4=SUM4+A(NA+I+2)+A(NA+NR-IK+1)
      IF(NF.LT.1) GO TO 30
      DO 38 I=1,NF,2
   38 SUM2=SUM2+A(NA+I+3)+A(NA+NR-IK+2)
   30 DO 32 J=1,ND,2
      DO 32 JJ=1,ND,2
   32 SUM16=SUM16+A(NA+J*IK+JJ+2)
      DO 33 K=1,ND,2
      DO 33 KK=1,NE,2
   33 SUM8=SUM8+A(NA+K*IK+KK+3)
      DO 34 L=2,NF,2
      DO 34 LL=1,ND,2
   34 SUM8=SUM8+A(NA+L*IK+LL+2)
      DO 35 M=2,NF,2
      DO 35 MM=1,NE,2
   35 SUM4=SUM4+A(NA+M*IK+MM+3)
      DO 36 MN=1,ND,2
   36 SUM4=SUM4+A(NA+MN*IK+2)+A(NA+MN*IK+IK+1)
      IF(NF.LT.2) GO TO 39
      DO 37 MNN=2,NF,2
   37 SUM2=SUM2+A(NA+MNN*IK+2)+A(NA+MNN*IK+IK+1)
   39 SUM=(SUM+2*SUM2+4*SUM4+8*SUM8+16*SUM16)*(H*HH)/9.0D0
      WRITE(6,62) SUM
      GO TO 99
    3 NB=NR-2
```

```
      XL=A(NA)
      XU=A(NA+1)
      DO 41 I=1,NB
   41 C(I)=A(NA+I+1)
      RANGE=XU-XL
      XM=(XU+XL)/2.0D0
      DO 42 I=1,6
      PT=X(I)*(XU-XM)+XM
      FX=C(1)*PT**11+C(2)*PT**10+C(3)*PT**9+C(4)*PT**8+C(5)*PT**7+
     1C(6)*PT**6+C(7)*PT**5+C(8)*PT**4+C(9)*PT**3+C(10)*PT**2+C(11)*PT
     2+C(12)
   42 SUM=FX*W(I)+SUM
      SUM=SUM*RANGE/2.0D0
      WRITE(6,61) SUM
      GO TO 99
    4 NB=NR-4
      XL=A(NA)
      XU=A(NA+1)
      YL=A(NA+2)
      YU=A(NA+3)
      XRANGE=XU-XL
      YRANGE=YU-YL
      XM=(XU+XL)/2.0D0
      YM=(YU+YL)/2.0D0
      DO 51 I=1,NB
   51 C(I)=A(NA+I+3)
      DO 53 I=1,6
      DO 53 J=1,6
      PTX=X(I)*(XU-XM)+XM
      PTY=X(J)*(YU-YM)+YM
      FX=C(1)*PTX**3*PTY**3+C(2)*PTX**3*PTY**2+C(3)*PTX**3*PTY+
     1C(4)*PTX**3+C(5)*PTX**2*PTY**3+C(6)*PTX**2*PTY**2+C(7)*PTX**2*PTY
     2+C(8)*PTX**2+C(9)*PTX*PTY**3+C(10)*PTX*PTY**2+C(11)*PTX*PTY+
     3C(12)*PTX+C(13)*PTY**3+C(14)*PTY**2+C(15)*PTY+C(16)
   53 SUM=FX*W(I)*W(J)+SUM
      SUM=SUM*XRANGE*YRANGE/4.0D0
      WRITE(6,62) SUM
   99 RETURN
   61 FORMAT(6H AREA=,1D25.16)
   62 FORMAT(8H VOLUME=,1D25.16)
   25 FORMAT(12F6.4)
      END
```

```
      SUBROUTINE POLFIT(XA,XB,*)                                        LIT14390
      IMPLICIT REAL *8 (A-H,O-Z)                                        LIT14400
      COMMON        XNAME(100),A(10000),W1(80),W2(80),W3(80),W4(80)     LIT14410
     1              ,NSIZE,NUM,NX,NR,NC,NNR,NNC,NDUM,NROW(100),NCOL(100) LIT14420
     2,NN(100),NW1(80)                                                  LIT14430
      DIMENSION CM(256),COEF(16),CM1(16),PV(16),CV(15),GSP(16)          LIT14440
      NC=1                                                              LIT14450
      NR=16                                                             LIT14460
      CALL MFIND(XA,&5)                                                 LIT14470
      NA=NX                                                             LIT14480
      HX=A(NA)                                                          LIT14490
      HY=A(NA+1)                                                        LIT14500
      DO 10 I=1,16                                                      LIT14510
   10 PV(I)=A(NA+1+I)                                                   LIT14520
      L=0                                                               LIT14530
      DO 100 I=1,4                                                      LIT14540
      Y=I-1                                                             LIT14550
      DO 100 J=1,4                                                      LIT14560
      X=J-1                                                             LIT14570
      DO 50 II=1,16                                                     LIT14580
   50 COEF(II)=0.0D0                                                    LIT14590
      CALL P(X,Y,COEF)                                                  LIT14600
      DO 70 K=1,16                                                      LIT14610
   70 CM(L*16+K)=COEF(K)                                                LIT14620
      L=L+1                                                             LIT14630
  100 CONTINUE                                                          LIT14640
      CALL INVERT(CM,CM1,GSP,16)                                        LIT14650
      CALL MULT(16,CM,16,PV,1,CV)                                       LIT14660
      CV(1)=CV(1)/(HX**3*HY**3)                                         LIT14670
      CV(2)=CV(2)/(HX**3*HY**2)                                         LIT14680
      CV(3)=CV(3)/(HX**3*HY)                                            LIT14690
      CV(4)=CV(4)/(HX**3)                                               LIT14700
      CV(5)=CV(5)/(HX**2*HY**3)                                         LIT14710
      CV(6)=CV(6)/(HX**2*HY**2)                                         LIT14720
      CV(7)=CV(7)/(HX**2*HY)                                            LIT14730
      CV(8)=CV(8)/(HX**2)                                               LIT14740
      CV(9)=CV(9)/(HX*HY**3)                                            LIT14750
      CV(10)=CV(10)/(HX*HY**2)                                          LIT14760
      CV(11)=CV(11)/(HX*HY)                                             LIT14770
      CV(12)=CV(12)/(HX)                                                LIT14780
      CV(13)=CV(13)/(HY**3)                                             LIT14790
      CV(14)=CV(14)/HY**2                                               LIT14800
      CV(15)=CV(15)/HY                                                  LIT14810
      NR=16                                                             LIT14820
      NC=1                                                              LIT14830
      CALL MLIST(XB,&5)                                                 LIT14840
      CALL MFIND(XB,&5)                                                 LIT14850
      NA=NX                                                             LIT14860
```

```
      DO 200 I=1,16
  200 A(NA-1+I)=CV(I)
      RETURN
    5 RETURN 1
      END

      SUBROUTINE P(X,Y,COEF)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COEF(16)
      IF(X.EQ.0.0D0.AND.Y.EQ.0.0D0) GO TO 10
      GO TO 15
   10 COEF(16)=1.0D0
      RETURN
   15 CONTINUE
      IF(X.EQ.0.0D0) GO TO 20
      GO TO 25
   20 COEF(13)=Y**3
      COEF(14)=Y**2
      COEF(15)=Y
      COEF(16)=1.0D0
      RETURN
   25 CONTINUE
      IF(Y.EQ.0.0D0) GO TO 30
      GO TO 35
   30 COEF(4)=X**3
      COEF(8)=X**2
      COEF(12)=X
      COEF(16)=1.0D0
      RETURN
   35 CONTINUE
      COEF(1)=(X**3)*(Y**3)
      DO 100 I=2,5
      IM=I-1
      IF(I.GT.4) GO TO 50
      COEF(I)=COEF(IM)/Y
   50 CONTINUE
      DO 100 J=2,4
      JJ=(J-1)*4+IM
      JM=JJ-4
      COEF(JJ)=COEF(JM)/X
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONTUR(XA,XB,NAR,NAC,NOC,*)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON       XNAME(100),A(10000),W1(80),W2(80),W3(80),W4(80)
     1            ,NSIZE,NUM,NX,NR,NC,NNR,NNC,NDUM,NROW(100),NCOL(100)
     2,NN(100),NW1(80)
      DIMENSION  ISYMB(42),BV(21),C(16),IGRID(101),U(16)
      INTEGER*2 GRAPH(16),GRAPH1(16),GRAPH2(16),GRAPH3(16),GRAPH4(16)
      DATA ISYMB/1H ,1HO,1H ,1HA,1H ,1HB,1H ,1HC,1H ,1HD,1H ,1HE
     1,1H ,1HF,1H ,1HG,1H ,1HH,1H ,1HI,1H ,1HJ,1H ,1HQ,1H ,1HR,1H ,1HS,
     21H ,1HT,1H ,1HU,1H ,1HV,1H ,1HW,1H ,1HX,1H ,1HY,1H ,1HZ/
      DATA IBLK/1H /
      DATA GRAPH /'U1--------U2---------U3---------U4 '/
      DATA GRAPH1/'U5--------U6---------U7---------U8 '/
      DATA GRAPH2/'U9--------U10--------U11--------U12'/
      DATA GRAPH3/'U13-------U14--------U15--------U16'/
      DATA GRAPH4/'|         |          |          |  '/
      GO TO (474,475),NAR
  474 CALL POLFIT(XA,XB,P5)
      CALL MFIND(XA,&5)
      NA=NX
      HX=A(NA)
      HY=A(NA+1)
      XL=0.0D0
      YL=0.0D0
      XRANGE=3.0D0*HX
      YRANGE=3.0D0*HY
      CALL MFIND(XB,&5)
      NB=NX
      DO 476 I=1,16
      U(I) =A(NA+1+I)
  476 C(I)=A(NB+I-1)
      GO TO 480
  475 CALL MFIND(XA,&5)
      NA=NX
      XL=A(NA)
      XU=A(NA+1)
      YL=A(NA+2)
      YU=A(NA+3)
      XRANGE=XU-XL
      YRANGE=YU-YL
      DO 477 I=1,16
  477 C(I)=A(NA+I+3)
  480 IJK=65
      IF(NAC.GT.1) IJK=80
      PJK=IJK-1.0
      GO TO (20,21),NAC
   20 GO TO (1,2),NAR
    1 WRITE(6,101)
```

```
      WRITE(6,300)  GRAPH
      DO 6 I=1,6
    6 WRITE(6,300)  GRAPH4
      WRITE(6,300)  GRAPH1
      DO 7 I=1,6
    7 WRITE(6,300)  GRAPH4
      WRITE(6,300)  GRAPH2
      DO 8 I=1,6
    8 WRITE(6,300)  GRAPH4
      WRITE(6,300)  GRAPH3
      WRITE(6,101)
      WRITE(6,301)  U(1),U(2),U(3),U(4),U(5),U(6),U(7),U(8)
      WRITE(6,302)  U(9),U(10),U(11),U(12),U(13),U(14),U(15),U(16)
      JMB=4-NOC
      DO 60 L=1,JMB
   60 WRITE(6,100)
      GO TO 4
    2 WRITE(6,103)
      WRITE(6,303)  C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8)
      WRITE(6,304)  C(9),C(10),C(11),C(12),C(13),C(14),C(15),C(16)
      JMB=20-NOC
      DO 61 L=1,JMB
   61 WRITE(6,100)
      GO TO 4
   21 GO TO (22,23),NAR
   22 WRITE(8,101)
      WRITE(8,300)  GRAPH
      DO 24 I=1,8
   24 WRITE(8,300)  GRAPH4
      WRITE(8,300)  GRAPH1
      DO 25 I=1,8
   25 WRITE(8,300)  GRAPH4
      WRITE(8,300)  GRAPH2
      DO 26 I=1,8
   26 WRITE(8,300)  GRAPH4
      WRITE(8,300)  GRAPH3
      WRITE(8,101)
      WRITE(8,301)  U(1),U(2),U(3),U(4),U(5),U(6),U(7),U(8)
      WRITE(8,302)  U(9),U(10),U(11),U(12),U(13),U(14),U(15),U(16)
      JMB=6-NOC
      DO 62 L=1,NCC
   62 WRITE(8,100)
      GO TO 4
   23 WRITE(8,105)
      WRITE(8,303)  C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8)
      WRITE(8,304)  C(9),C(10),C(11),C(12),C(13),C(14),C(15),C(16)
      JMB=9-NOC
      DO 63 L=1,JMB
```

```
63 WRITE(8,100)
 4 PMAX=-1.0D70
   PMIN= 1.0D70
   DO 10 I=1,IJK
   DO 10 J=1,101
   X1=J-1
   Y1=I-1
   X=XL+(XRANGE*X1)/100.0D0
   Y=YL+(YRANGE*Y1)/PJK
   PP=C(1)*X**3*Y**3+C(2)*X**3*Y**2+C(3)*X**3*Y+C(4)*X**3 +
  1C(5)*X**2*Y**3+C(6)*X**2*Y**2+C(7)*X**2*Y+C(8)*X**2+C(9)*X*Y**3+
  2C(10)*X*Y**2+C(11)*X*Y+C(12)*X+C(13)*Y**3+C(14)*Y**2+C(15)*Y+C(16)
   IF(PP.GT.PMAX) PMAX=PP
10 IF(PP.LT.PMIN) PMIN=PP
   RANGE=PMAX-PMIN
   IF(DABS(RANGE).GT.1.0D-10) GO TO 11
   WRITE(6,1000)
   WRITE(6,1400) RANGE,PMAX,MIN
   GO TO 1500
11 SCF=40./RANGE
   DO 12 I=1,21
   RR=I-1
12 BV(I)=2*RR/SCF+PMIN
   GO TO (30,31),NAC
30 WRITE(6,1300) BV(1),BV(1),BV(21)
   WRITE(6,1100) BV(2),BV(3),BV(4),BV(5),BV(6)
   WRITE(6,1200) BV(7),BV(8),BV(9),BV(10),BV(11)
   WRITE(6,1210) BV(12),BV(13),BV(14),BV(15),BV(16)
   WRITE(6,1220) BV(17),BV(18),BV(19),BV(20),BV(21)
   WRITE(6,102)
   GO TO 40
31 WRITE(8,1300) BV(1),BV(1),BV(21)
   WRITE(8,1100) BV(2),BV(3),BV(4),BV(5),BV(6)
   WRITE(8,1200) BV(7),BV(8),BV(9),BV(10),BV(11)
   WRITE(8,1210) BV(12),BV(13),BV(14),BV(15),BV(16)
   WRITE(8,1220) BV(17),BV(18),BV(19),BV(20),BV(21)
   WRITE(8,104)
40 CONTINUE
   DO 13 I=1,IJK
   DO 14 K=1,101
14 IGRID(K)=IBLK
   DO 15 J=1,101
   X1=J-1
   Y1=I-1
   X=XL+(XRANGE*X1)/100.0D0
   Y=YL+(YRANGE*Y1)/PJK
   PP=C(1)*X**3*Y**3+C(2)*X**3*Y**2+C(3)*X**3*Y+C(4)*X**3 +
  1C(5)*X**2*Y**3+C(6)*X**2*Y**2+C(7)*X**2*Y+C(8)*X**2+C(9)*X*Y**3+
```

```
   2C(10)*X*Y**2+C(11)*X*Y+C(12)*X+C(13)*Y**3+C(14)*Y**2+C(15)*Y+C(16)LIT16730
    KK=((PP-PMIN)*SCF+2.50001)                                          LIT16740
 15 IGRID(J)=ISYMB(KK)                                                  LIT16750
    GO TO (50,51),NAC                                                   LIT16760
 50 WRITE(6,99) (IGRID(II),II=1,101)                                    LIT16770
    GO TO 13                                                            LIT16780
 51 WRITE(8,99) (IGRID(II),II=1,101)                                    LIT16790
 13 CONTINUE                                                            LIT16800
    IF(NAC.EQ.1) WRITE(6,102)                                           LIT16810
    IF(NAC.EQ.2) WRITE(8,104)                                           LIT16820
 99 FORMAT(15X,101A1)                                                   LIT16830
100 FORMAT(/)                                                           LIT16840
101 FORMAT(////)                                                        LIT16850
102 FORMAT(//////)                                                      LIT16860
103 FORMAT(////)                                                        LIT16870
104 FORMAT(1H1)                                                         LIT16880
105 FORMAT(//////////////)                                              LIT16890
300 FORMAT(25X,16A2)                                                    LIT16900
301 FORMAT(15X,5H U1 =,F12.6,5H U2 =,F12.6,5H U3 =,F12.6,5H U4 =,      LIT16910
   1F12.6,//,15X,5H U5 =,F12.6,5H U6 =,F12.6,5H U7 =,F12.6,5H U8 =,    LIT16920
   2F12.6,/)                                                            LIT16930
302 FORMAT(15X,5H U9 =,F12.6,5H U10=,F12.6,5H U11=,F12.6,5H U12=,      LIT16940
   1F12.6,//,15X,5H U13=,F12.6,5H U14=,F12.6,5H U15=,F12.6,5H U16=,    LIT16950
   2F12.6,////)                                                         LIT16960
303 FORMAT(15X,F12.6,5H X3Y3,F12.6,5H X3Y2,F12.6,5H X3Y ,F12.6,        LIT16970
   15H X3  ,//,15X,F12.6,5H X2Y3,F12.6,5H X2Y2,F12.6,5H X2Y ,F12.6,    LIT16980
   25H X2  ,/)                                                          LIT16990
304 FORMAT(15X,F12.6,5H XY3 ,F12.6,5H XY2 ,F12.6,5H XY  ,F12.6,        LIT17000
   15H X   ,//,15X,F12.6,5H Y3  ,F12.6,5H Y2  ,F12.6,5H Y   ,F12.6)    LIT17010
1400 FORMAT(//,5X,13H THE RANGE IS,E15.6,/,5X,13H   MAXIMUM IS,E15.6,/,LIT17020
   15X,13H   MINIMUM IS,E15.6)                                          LIT17030
1000 FORMAT(1H1)                                                        LIT17040
1050 FORMAT(//)                                                         LIT17050
1100 FORMAT(5X,3H A=,F12.4,3X,3H B=,F12.4,3X,3H C=,F12.4,3X,3H D=,F12.4LIT17060
   1,3X,3H E=,F12.4,/)                                                  LIT17070
1200 FORMAT(5X,3H F=,F12.4,3X,3H G=,F12.4,3X,3H H=,F12.4,3X,3H I=,F12.4LIT17080
   1,3X,3H J=,F12.4,/)                                                  LIT17090
1210 FORMAT(5X,3H Q=,F12.4,3X,3H R=,F12.4,3X,3H S=,F12.4,3X,3H T=,F12.4LIT17100
   1,3X,3H U=,F12.4,/)                                                  LIT17110
1220 FORMAT(5X,3H V=,F12.4,3X,3H W=,F12.4,3X,3H X=,F12.4,3X,3H Y=,F12.4LIT17120
   1,3X,3H Z=,F12.4,/)                                                  LIT17130
1300 FORMAT(5X,3H O=,F15.10,5X,7H RANGE(,E15.6,5X,E15.6,2H ),/)        LIT17140
1500 RETURN                                                             LIT17150
   5 RETURN 1                                                           LIT17160
     END                                                                LIT17170
```

LIST OF REFERENCES

1. Holman, J. P., Heat Transfer, 2nd ed., p.55, McGraw-Hill, 1968.

2. Crandall, S. H., Engineering Analysis, pp.1-405, McGraw-Hill, 1956.

3. Schlichting, H., Boundary Layer Theory, 6th ed., pp.181-184, McGraw-Hill, 1968.

4. Wilkinson, J. P., The Algebraic Eigenvalue Problem, pp.189-483, Oxford University Press, 1965.

5. Clenshaw, C. W., and others, Modern Computing Methods, 2nd ed., pp.1-136, John Wright & Sons, 1961.

6. Forsythe, G. E. and Moler, C. B., Computer Solution of Linear Algebraic Systems, pp.1-136, Prentice-Hall, 1967.

7. Fox, L., An Introduction to Numerical Linear Algebra, pp.60-99, Oxford University Press, 1965.

8. IBM System/360, "FORTRAN IV LANGUAGE," Form C-28-6515-5, IBM Corp., Poughkeepsie, N. Y.

9. Przemieniecki, J. S., Theory of Matrix Structural Analysis, pp.70-82, McGraw-Hill, 1968.

10. USERS MANUAL, 1st ed., pp.4-1 - 4-9, Naval Postgraduate School Computer Center, 1970.

11. Stroud, A. H. and Secrest, D., Gaussian Quadrature Formulas, p.7, Prentice-Hall, 1966.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Director, Computer Facility, Code 0211 Naval Postgraduate School Monterey, California 93940	5
4.	Professor Gilles Cantin, Code 59Ci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	5
5.	LT R. D. Little, USN Ship Repair Facility, Guam FPO San Francisco, California 96630	2

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

An Interactive Matrix Interpretive System For the IBM 360/67

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Master's Thesis; June 1970

5. AUTHOR(S) (First name, middle initial, last name)

Robert Douglas Little

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1970	97	11

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

The Matrix Interpretive System devised by Wilson and modified by Cantin has been transformed into an interactive program operable under a time sharing system. Several new commands have been added to extend the usefulness of the code and give it the capability to offline read, write and punch data, to plot graphs and contour maps and to integrate simple polynomial expressions.

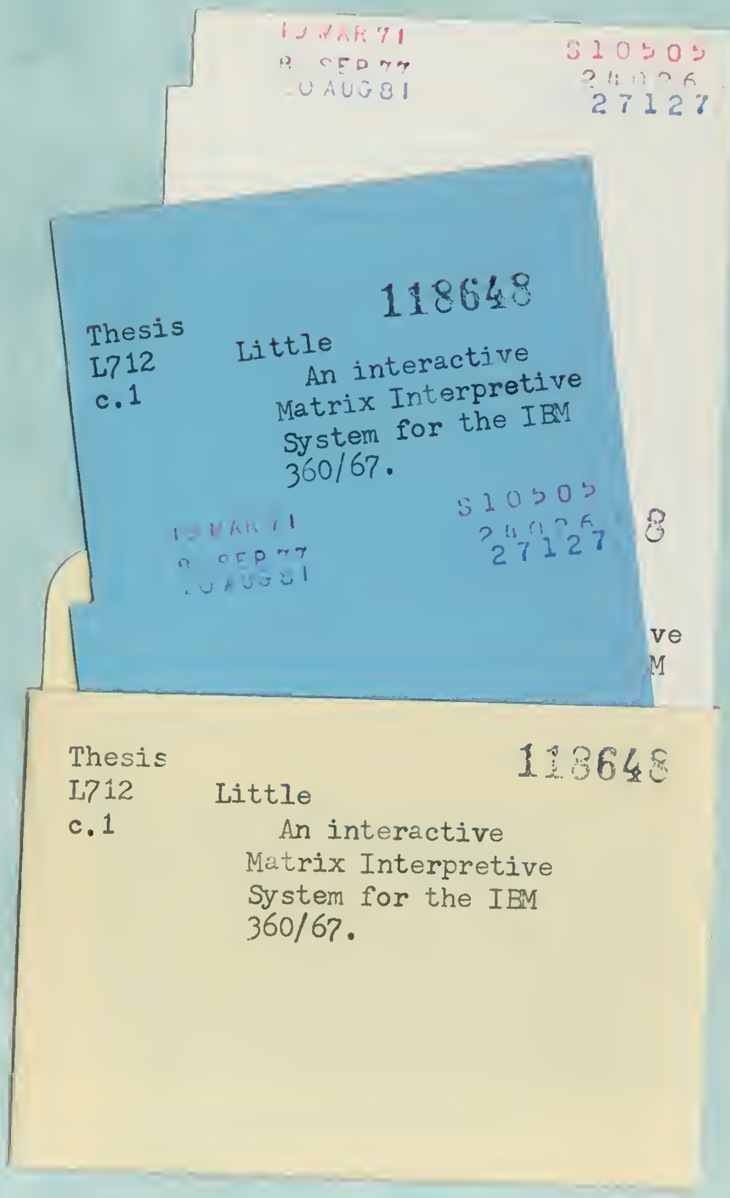
DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

A-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Problem Oriented Computer Sublanguage						
Matrix Interpretive System						
Numerical Linear Algebra						



118648
Thesis
L712
c.1
Little
An interactive
Matrix Interpretive
System for the IBM
360/67.
S10505
27127
Thesis
L712
c.1
Little
An interactive
Matrix Interpretive
System for the IBM
360/67.
118648