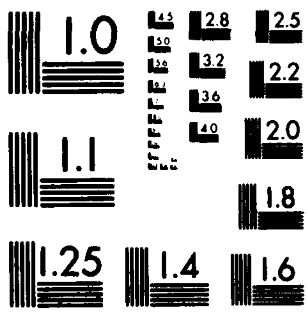
AD-A110 673   BDM CORP MONTEREY CA   F/G 20/4

PROCEDURE AND COMPUTER PROGRAM FOR THE APPROXIMATION OF DATA (W--ETC(U)

AUG 80   H ZEBNER   N00014-78-C-0204

UNCLASSIFIED   NPS-67-80-001CR   NL

1 of 2



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

(2)

NPS67-80-001CR

AD A110673

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC
ELECTE
FEB 0 9 1982
S D
E

CONTRACTOR REPORT

PROCEDURE AND COMPUTER PROGRAM
FOR THE APPROXIMATION OF DATA
(WITH APPLICATION TO MULTIPLE SENSOR PROBES)

H. ZEBNER
BDM CORPORATION
P.O. Box 2019
Monterey, CA 93940

August 1980

Interim Report for Period June 1980 - August 1980

Approved for public release; distribution unlimited

Copy available to DTIC does not permit fully legible reproduction

Prepared for:
Naval Postgraduate School
Monterey, California 93940

DTIC FILE COPY

82 02 08 186

NAVAL POSTGRADUATE SCHOOL

Monterey, California

Rear Admiral J. J. Ekelund
Superintendent

D. A. Schrady
Acting Provost

The work reported herein was carried out by the BDM Corporation under Work Order 426, Turbopropulsion Laboratory Support, under Contract Number N00014-78-C-0204. The work was part of a program entitled Transonic Compressor Investigations, funded in part by Naval Air Systems Command under Program Element 61153N, Contract Number N00019-80-WR-01199. The program was under the cognizance of Dr. H. J. Mueller, Code 310.

The author is a candidate for the degree D. Ing under Prof. H. Gallus at Rhein,-Westf. Techn. Hochschule, 51 Aachen, W. Germany.

This report was prepared by:

Hans M. Hans

BDM Corporation
P.O. Box 2019
Monterey, CA 93940

Publication of the report does not constitute approval of the sponsor for the findings or conclusions. It is published for information and for the exchange and stimulation of ideas.

Reviewed by:

R. P. Shreeve

R. P. SHREEVE, Director
Turbopropulsion Laboratory

M. F. Platzer

M. F. PLATZER, Chairman
Department of Aeronautics

Released by:

William M. Tolles

W. M. TOLLES
Dean of Research

DISCLAIMER NOTICE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS67-80-001CR	2. GOVT ACCESSION NO. AD-A110 673	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PROCEDURE AND COMPUTER PROGRAM FOR THE APPROXIMATION OF DATA (WITH APPLICATION TO MULTIPLE SENSOR PROBES).		5. TYPE OF REPORT & PERIOD COVERED CONTRACTOR REPORT June 1980 - August 1980 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) H. Zebner		8. CONTRACT OR GRANT NUMBER(s) N00014-78-C-0204
9. PERFORMING ORGANIZATION NAME AND ADDRESS BDM CORPORATION P.O. BOX 2019 MONTEREY, CALIFORNIA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Order 426
11. CONTROLLING OFFICE NAME AND ADDRESS NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA 93940		12. REPORT DATE August 1980 13. NUMBER OF PAGES 186
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Interim Report. Principal Investigator, Dr. R. P. Shreeve, Director Turbopropulsion Laboratory, Naval Postgraduate School, Monterey, California 93940

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Data Approximation
Probe Calibration
Flow Measurements

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A procedure to approximate data given at arbitrary intervals in two-, three- and four dimensions using polynominal expressions is described. The programming of the problem is explained in each case and a user manual is given for software implemented on the TPL Hewlett-Packard computer system. The method, which is general, was derived and has been applied to represent the calibration of flow probes which have multiple sensors.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 | 1

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

4101

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

ABSTRACT

A procedure to approximate data given at arbitrary intervals in two-, three- and four dimensions using polynominal expressions is described. The programming of the problem is explained in each case and a user manual is given for software implemented on the TPL Hewlett-Packard computer system. The method, which is general, was derived and has been applied to represent the calibration of flow probes which have multiple sensors.

1. INTRODUCTION

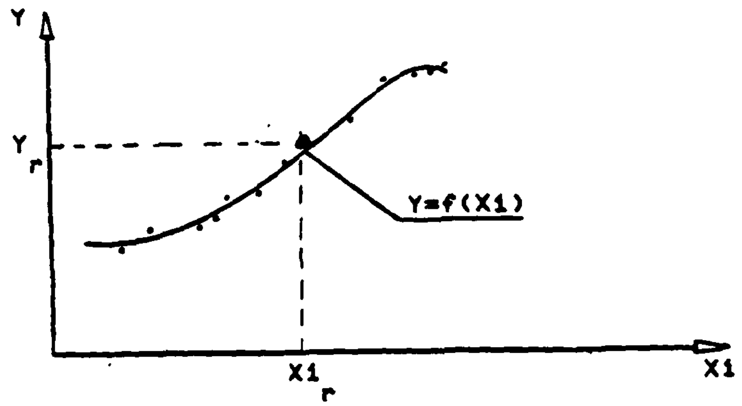
This report describes algorithms to approximate data patterns such as those which occur, for instance, when pneumatic-velocity probes are calibrated. The functional value (Y) can depend on either one (X1), two (X1 and X2) or even three (X1, X2 and X3) parameters. The approximation Y = f(X1) will be referred to as the two dimensional, the approximation Y = f(X1, X2) as the three dimensional and the approximation Y = f(X1, X2, X3) as the four dimensional approximation. These three options meet the requirements for probes used in the Turbopropulsion Laboratory.

Using the least-squares criterion to obtain the coefficients in assumed polynominals leads to a system of linear equations, which are, in principle easily solved. Numerical problems may arise however, since the software, as described herein and as implemented in the NPS Turbopropulsion Laboratory Hewlett Packard 21MX computer, uses 32-bit real constants.

The author wishes to express his thanks to Professor Ray Shreeve, whose constant and critical interest was a crucial help to solve this mathematical problem and link it to engineering application.

2. TWO DIMENSIONAL APPROXIMATION

2.1. Problem:



A data set of NPNTS1 data points is given, where Y depends on parameter X1. The data pattern is to be approximated by a function Y = f(X1), so that the error between data points and analytically determined points is lowest.

2.2. <u>Approach</u>:

Polynominals are commonly used in order to approximate data patterns. $Y = f(X1)$ is a function which approximates the data value, V, at each value of the variable X1, we look for our expression of the form

$$Y = C_1 + C_2 * X1 + C_3 * X1^2 + \ldots + C_L * X1^{(L-1)}$$

or

$$Y = \sum_{i=1}^{L} C_i * X1^{(i-1)} \qquad (2.1.)$$

in which the coefficients C_i are to be determined by the method of least squares. As Ref. 1 shows the least squares criterion leads to a linear equation system that can easily be solved. We define the error

$$R = \sum_{r=1}^{NPNTS1} [f(X1_r) - Y_r]^2 \qquad (2.2.)$$

where the index r denotes the individual data point. Using equation (2.1.) R becomes

$$R = \sum_{r=1}^{NPNTS1} [C_1 + C_2 * X1_r + C_2 * X1_r^2 + \ldots + C_L * X1_r^{(L-1)} - Y_r]^2$$

2.3. Solution:

R depends on the selection of the coefficients $C_1,\ldots,C_L$. In order to minimize the error, R is differentiated with respect to $C_1,\ldots,C_L$ and the partial derivatives are set to zero. Thus

$$\frac{\partial R}{\partial C_i} \overset{!}{=} 0 \qquad i = 1,\ldots,L$$

or

$$\frac{\partial R}{\partial C_i} = \sum_{r=1}^{NPNTS1} 2*[C_1+C_2*X1_r+C_3*X1_r^2+\ldots+C_L X1_r^{(L-1)}-Y_r]*$$

$$* \frac{\partial [C_1+C_2*X1_r+C_3*X1_r^2+\ldots+C_L X1_r^{(L-1)}-Y_r]}{\partial C_i} = 0$$

Assuming that the summations extend over all the data points, Σ should be understood to mean $\sum_{r=1}^{NPNTS1}$. Performing the differentiations and rearranging the equations, we get

$$NPNTS1*C_1 + \Sigma X1_r*C_2 + \Sigma X1_r^2*C_3 + \ldots + \Sigma X1^{(L-1)}*C_L = \Sigma Y_r$$

$$\Sigma X1_r*C_1 + \Sigma X1_r^2*C_2 + \Sigma X1_r^3*C_3 + \ldots + \Sigma X1_r^L \quad *C_1 = \Sigma Y_r*X1_r$$

$$\Sigma X1_r^2*C_1 + \Sigma X1_r^3*C_2 + \Sigma X1_r^4*C_3 + \ldots + \Sigma X1_r^{(L+1)}*C_L = \Sigma Y_r*X1_r^2$$

....

$$\Sigma X1_r^{(L-1)}*C_1 + \Sigma X1_r^L*C_2 + \Sigma X1_r^{(L+1)}*C_3+\ldots+\Sigma X1_r^{(2L-2)}*C_L = \Sigma Y_r*X1_r^{L-1}$$

In matrix notation

$$\begin{pmatrix} NPNTSi & \Sigma Xi_r & \Sigma Xi_r^2 & \dots & \Sigma Xi_r^{L-1} \\ \Sigma Xi_r & \Sigma Xi_r^2 & \Sigma Xi_r^3 & \dots & \Sigma Xi_r^{L} \\ \Sigma Xi_r^2 & \Sigma Xi_r^3 & \Sigma Xi_r^4 & \dots & \Sigma Xi_r^{L+1} \\ \dots\dots\dots & & & & \\ \Sigma Xi_r^{L-1} & \Sigma Xi_r^{L} & \Sigma Xi_r^{L+1} & \dots & \Sigma Xi_r^{2L-2} \end{pmatrix} * \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \\ c_L \end{pmatrix} = \begin{pmatrix} \Sigma Y_r \\ \Sigma Y_r * Xi_r \\ \Sigma Y_r * Xi_r^2 \\ \\ \Sigma Y_r * Xi_r^{L-1} \end{pmatrix}$$

$$\mathbb{A} \qquad\qquad * \ \mathbb{C} \ = \mathbb{B} \qquad (2.3.)$$

$\mathbb{A}$ ... System Matrix

$\mathbb{C}$ ... Coefficients vector

$\mathbb{B}$ ... Right hand side vector

2.4. Equation System

2.4.1. System Matrix A

Since all Matrix elements on diagonal lines running up from left to right (2. order, as defined in Appendix A) are identical, the elements are renamed for simplification as follows:

$$
\begin{array}{lllllll}
a_1 = a_{11} & & & & & & \\
a_2 = a_{21} = a_{12} & & & & & & \\
a_3 = a_{31} = a_{22} = a_{13} & & & & & & \\
\vdots & & & & & & \\
a_L = a_{L1} & = a_{L-1,2} & = a_{L-2,3} & = & \cdots & = a_{2,L-1} & = a_{1,L} \\
a_{L+1} & = a_{L,2} & = a_{L-1,3} & = & \cdots & = a_{3,L-1} & = a_{2,L} \\
a_{L+2} & & = a_{L,3} & = & \cdots & = a_{4,L-1} & = a_{3,L} \\
\vdots & & & & & & \\
a_{2L-1} & & & & & & = a_{L,L}
\end{array}
$$

so that the set of elements, a_k, can be written as

$$a_k = \sum_{r=1}^{NPNTS1} X1_r^{\,k-1} \qquad k = 1, \ldots, 2L-1 \tag{2.4.}$$

This operation is programmed in REAL FUNCTION S2 (NPNTS1,IPOWR1,IY). The data X1 and Y are known to this function through a common block named DTA2. So only NPNTSI, IPOWR1 (= k-1) and IY (=0) have to be passed to the function, and the value of a_k is returned through the function name S2.

SUBROUTINE MAT2 presets the system matrix in the following way:

i) Preset edge section elements (using REAL FUNCTION S2)

$$\left|\begin{array}{ccccccccc} a_1 & \Rightarrow & a_2 & \Rightarrow & a_3 & \Rightarrow & \cdots & \Rightarrow & a_L \\ &&&&&&&& \Downarrow \\ &&&&&&&& a_{L+1} \\ &&&&&&&& \Downarrow \\ &&&&&&&& a_{L+2} \\ &&&&&&&& \Downarrow \\ &&&&&&&& \cdots \\ &&&&&&&& \Downarrow \\ &&&&&&&& a_{2L-1} \end{array}\right|$$

$a_1 \rightarrow a_L$
Top edge section

$a_{L+1} \rightarrow a_{2L-1}$
right hand edge section

ii) Copy defined elements diagonally

$$\left|\begin{array}{ccccccccc} a_{11} & & a_{12} & & a_{13} & & \cdots & & a_{1L} \\ & \swarrow & & \swarrow & & \swarrow & & \swarrow & \\ &&&&&&&& a_{2L} \\ &&&&&&&& a_{3L} \\ &&&&&&&& \cdots \\ &&&&&&&& a_{LL} \end{array}\right|$$

$$\left|\begin{array}{ccccccccc} a_{11} & & a_{12} & & a_{13} & & \cdots & & a_{1L} \\ a_{21} & & a_{22} & & a_{23} & & \cdots & & a_{2L} \\ & \swarrow & & \swarrow & & \swarrow & & \swarrow & \\ &&&&&&&& a_{3L} \\ &&&&&&&& \cdots \\ &&&&&&&& a_{LL} \end{array}\right|$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1L} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2L} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3L} \\ & & & & \cdots \\ & & & & a_{LL} \end{vmatrix}$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1L} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2L} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3L} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & \end{vmatrix}$$

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1L} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2L} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3L} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{L1} & a_{L2} & a_{L3} & \cdots & a_{LL} \end{vmatrix}$$

2.4.2. <u>Right hand side vector B</u>

The elements of the right hand side vector B can be written as

$$b_k = \sum_{r=1}^{NPNTS1} (Y_r \cdot X1_r^{k-1}) \qquad k = 1, \ldots, L \tag{2.5.}$$

To calculate b_k, again REAL FUNCTION S2 is used. Data X1 and Y are available through COMMON block DTA2. NPNTS1, IPOWR1 (=k-1) and IY (=1) have to be passed to the function, and the value of b_k is returned through the function name S2.

2.5. Software:

The software to compute the coefficients for a two dimensional approximation is described in Appendix B and is implemented in the TPL HP-21MX computer system

To work correctly with these program modules, the user must conform to the following conventions:

i) Provide the data in two arrays (Type: REAL) of 256 elements through a COMMON block, named DTA2.

```
COMMON / DTA2 / X1,Y
REAL X1(256),Y(256)
```

ii) Dimension an array (Type: REAL) of 7 (seven) elements to contain the coefficients.

```
REAL COEF(7)
```

iii) Define the parameters NPNTS1, L and IPRINT (Type: all INTEGER)

NPNTS1 ... # of data points
$L \leq NPNTS1 \leq 256$

L ... (desired order of polynominal) + 1
$1 \leq L \leq 7$

IPRINT ... controls quantity of print out
- 2 ... Print system matrix and right hand side vector before and after Gauss Jordan Elimination
- 1 ... Print equation system after Gauss Jordan Elimination
- <0 ... No print out
- >0 ... Print equation (1,1) with the actual parameters

iv) When loading a program, that uses the subroutine MAT2, the binary library file has to be searched for externals.

Suppose, the source file &USER::26 contains a user program named USER. This program calls MAT2. After the compilation, the relocatable binary file for this user program is %USER::26. To load the program, the following procedure is recommended:

Type (from FMGR)

:RU,LOADR

and the loader program will respond

/LOADR: RE,%USER::26

where the underlined information already is the user's input. This causes the loader to load all program modules of the user program; a load map is listed on the terminal. Upon completion the loader prompts

/LOADR: MS,%TPLBL::26

where the underlined information already is the user's input. Now the loader conducts a search for all unsatisfied externals of the user program in the binary library file %TPLBL::26. Since some library programs have externals themselves the search has to be repeated (MS ... multiple search) as many time as is necessary for all externals to be satisfied. The loader prompts

/LOADR: END

where the underlined information is the user's input. The loader now loads all system programs, outputs a load map and generates the program. Upon completion, the loader outputs a ready message

/LOADR:USER READY AT 1:28 PM WED., 10 SEPT, 1980
/LOADR: END

Now the program can be run

:RU,USER

Again, the underlined information is the user's input.

If all these requirements are met, the correct call for the subroutine is:

CALL MAT2 (NPNTS1,L,COEF,IPRINT)

Upon completion, the array COEF contains the coefficients (COEF(1)=C_1 ,...,COEF(L)=C_L). Externals used by MAT2 are: AB2, DTA2, S2 under no circumstances may the user use any of these names for modules of his own user program. In some cases the program may not be able to perform a Gauss Jordan Elimination to the system matrix and the right hand side vector and thus cause the program to stop. If this happens, an error message is displayed.

It is highly encouraged, to use the system function FNP to calculate the value of a polynominal at one specified X1 rather than a loop such as the following:

```
      S=.0
      DO 08 I=1,L,1
   08 S=S+X**(I-1)*COEF(I)
      YCALC=S
```

To call FNP is easier and faster, because FNP is a programmed Horner Scheme. The recommended call is then;

```
      L1=L-1
      YCALC=FNP(COEF,X,L1)
```

2.6. <u>Sample User Program</u>

2.6.1. <u>Listing</u>

```
PAGE 0001  FTN.  12:40 PM  THU., 18  SEP., 1980

FTN4,L
      PROGRAM DEMO2 (3,99)
C     ..............................................................
C     .
C     .  THIS IS A DEMONSTATION PROGRAM AND IT SHOWS THE CORRECT USE
C     .  OF THE TPL BINARY LIBRARY (ON TYPE 6 FILE %TPLBL) USING
C     .  EXAMPLES.  FOR FURTHER QUESTIONS READ THE TPL-LIBRARY-BINDER
C     .  OR CONSULT THE SOFTWARE MANAGER .
C     .
C     ..............................................................
     *,     DEMONSTRATE THE USE OF THE TPL BINARY LIBRARY.

      COMMON / AFLD / A
      [redacted]

      REAL A(256)
      [redacted]

      [redacted]
      INTEGER NOLF,NOCR(2),ICLR(3)

      DATA NOLF  /006537B/
      DATA NOCR  /000033B,040433B/
      DATA ICLR  /015524B,015515B,006537B/
      DATA PI    /3.141593/

C     FORMATS DEMO2 START
  101 FORMAT (///"  HELLO! THIS IS PROGRAM DEMO AND WE'LL SEE H
     *OW TO USE THE MARVELLOUS TPL BINARY"A2/"  LIBRARY!  YOUR INTEREST
     *WILL BE GREATLY AWARDED BY EASIER PROGRAMMING."///)
  102 FORMAT ("  GENERATING DATA POINTS"/5X"I"5X"X(I)"5X"Y(I)")
  103 FORMAT (2X,I4,2F9.3,A2)
  104 FORMAT ("  DATA POINTS GENERATED!"/)
  105 FORMAT ("  INITIALIZING THE PLOTTER")
  106 FORMAT ("  PLOTTER INITIALIZED!"/)
  107 FORMAT ("  DEFINE PLOTTER AND USER AREAS AND DRAW AXES")
  108 FORMAT (9X""6X"XMIN "6X"XMAX "6X"YMIN
     *"6X"YMAX"/9X,4(""10X" ")/
     *"  HP9872",4(" "F10.3""),""/
     *"  USER  ",4(" "F10.3""),""/)
  109 FORMAT ("  PLOTTER AREAS DEFINED AND AXES DRAWN!"/)
  110 FORMAT ("  DRAWING DATA POINTS INTO COORDINATES SYSTEM")
  111 FORMAT ("  DATA POINTS DRAWN!"/)
  112 FORMAT ("  CALCULATING A CURVE FIT THROUGH THE DATA POINTS
     *")
  113 FORMAT ("  ENTER DEGREE OF POLYNOMIAL (NORDER) TO FIT THRO
     *UGH THE POINTS  "2A2)
  114 FORMAT ("  ENTER IPRINT  "2A2)
  115 FORMAT (/"  CURVE FIT DONE!"/)
  116 FORMAT ("  CURVE FIT DONE!"/)
  117 FORMAT ("  PLOTTING THE CURVE FIT")
  118 FORMAT ("  CURVE FIT DRAWN!"/)
  149 FORMAT (""((3A2)))
C     FORMATS DEMO2 STOP .


C     ..............................................................
C     .
C     .  GET THE LU OF THE TERMINAL.
C     .
C     ..............................................................
      LI = LOGLU(I)
      WRITE (LI, 101) NOLF



C     ..............................................................
C     .
C     .  GENERATE DATA POINTS SCATTERED AROUND A POLYNOMIAL.
C     .
C     ..............................................................
      CALL INITG (13)
      WRITE (LI, 102)
      XSTART = -6.
      XSTOP  = +2.
```

```
      [redacted]
      DX     = (XSTOP-XSTART)/(NPNTS1-1)
      NORDER = 4
      COEF(1) = -1.
      COEF(2) = -0.500000
      COEF(3) = -0.000340
      COEF(4) = +0.300000
      COEF(5) = +0.055000
      A       = (2.0*PI)*0.03125
      DO 01 I=1,NPNTS1,1
      X1(I)   = XSTART+(I-1)*DX
      Y(I)    = FNP(COEF,X1(I),NORDER) + 0.25*SIN(X1(I)/A)
   01 WRITE (LI, 103) I,X1(I),Y(I),NOLF
      ISTOP1 = NPNTS1-1
      DO 02 I=1,ISTOP1,2
      DUMMY  = Y(I+1)
      Y(I+1) = Y(I)
   02 Y(I)   = DUMMY
      WRITE (LI, 149) (ICLR,I=1,3,1)
      WRITE (LI, 104)



C     ..........................................................................
C     .
C     .  INITIALIZE THE PLOTTER.
C     .
C     ..........................................................................
      WRITE (LI, 105)
      LP = 13
      CALL INITG (13)
      WRITE (LI, 149) ICLR
      WRITE (LI, 106)



C     ..........................................................................
C     .
C     .  DEFINE PLOTTER AND USER AREAS;  DRAW AXES.
C     .
C     ..........................................................................
      WRITE (LI, 107)
      XPMIN  = 2.
      XPMAX  = 12.
      YPMIN  = 12.
      YPMAX  = 22.
      XUMIN  = -6.
      XUMAX  = +2.
      YUMIN  = -5.
      YUMAX  = 10.
      WRITE (LI, 108) XPMIN,XPMAX,YPMIN,YPMAX,XUMIN,XUMAX,YUMIN,YUMAX
      ALPHAX =  0.00
      XA     = (XPMAX      -XPMIN      )/(XUMAX-XUMIN)
      XB     = (XPMIN*XUMAX-XPMAX*XUMIN)/(XUMAX-XUMIN)
      XL     = (XPMAX      -XPMIN      )
      ALPHAY = 90.00
      YA     = (YPMAX      -YPMIN      )/(YUMAX-YUMIN)
      YB     = (YPMIN*YUMAX-YPMAX*YUMIN)/(YUMAX-YUMIN)
      YL     = (YPMAX      -YPMIN      )
      CALL SETSM (113,1.)
      CALL AXIS (XPMIN,YPMIN,XL,ALPHAX,2HX1,+2,XUMIN,XUMAX,4HF4.1,4,4)
      CALL AXIS (XPMIN,YPMIN,YL,ALPHAY,2H Y,-2,YUMIN,YUMAX,2HI2,2,3)
      WRITE (LI, 149) (ICLR,I=1,6,1)
      WRITE (LI, 109)



C     ..........................................................................
C     .
C     .  PLOT DATA POINTS.
C     .
C     ..........................................................................
      WRITE (LI, 110)
      CALL SETSM (113,3.)
      DO 03 I=1,NPNTS1,1
```

```
      XPLOT  = X1(I)*XA+XB
      YPLOT  = Y(I)*YA+YB
      CALL PLOT (XPLOT,YPLOT,2)
   03 CALL SYMBL (1)
      WRITE (LI, 149) ICLR
      WRITE (LI, 111)



C     ..........................................................
C     .
C     .  CALCULATE CURVE FIT THROUGH DATA POINTS.
C     .
C     ..........................................................
      WRITE (LI, 112)
   04 WRITE (LI, 113) NOCR
      READ  (LI,   *) NORDER
      WRITE (LI, 149) ICLR
      IF (NORDER .LT. 0   .OR.   NORDER  .GT.  6 ) GO TO 04
      WRITE (LI, 114) NOCR
[redacted]
      WRITE (LI, 149) (ICLR,I=1,2,1)
[redacted]
      IF (IPRINT.GE.0) WRITE (LI, 115)
      IF (IPRINT.LT.0) WRITE (LI, 116)



C     ..........................................................
C     .
C     .  PLOT POLYNOMIAL FIT.
C     .
C     ..........................................................
      WRITE (LI, 117)
      CALL SETSM (113,2.)
      DO 09 I=1,NPNTS1,1
      XPLOT  = X1(I)*XA+XB
      YPLOT  = FNP(COEF,X1(I),NORDER)*YA+YB
      IF (I.EQ.1) CALL PLOT (XPLOT,YPLOT,1)
   09 IF (I.GT.1) CALL PLOT (XPLOT,YPLOT,3)
      CALL SETSM (115,DUMMY)
      WRITE (LI, 149) ICLR
      WRITE (LI, 118)



C     ..........................................................
C     .
C     .  TERMINATE GRAPHICS.
C     .
C     ..........................................................
      CALL STOPG
      STOP 7777
      END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 01409 COMMON = 00000

2.6.2. Load map

```
DEMO2   10042 12642        DEMONSTRATE THE USE OF THE TPL BINARY LIBRARY.
FNP     12643 12746                      Calculate n-th order polynomic
```

to load these programs, type (from LOADR): MS,ZTPLBL

```
[redacted]   12747 14746   2D-Approximation  / DTA2  /         (DATA2  DTA2).
[redacted]   14747 16777   2D-Approximation arrange system matrix and vector.
[redacted]   17000 17131   2D-Approximation              compute summations.
[redacted]   17132 17311   2D-Approximation  / AB2  /
```

```
AFLD    17312 20311    GSP   / AFLD   / A(256)
AXIS    20312 22631    GSP                          Draw and label axes.
INITG   22632 23644    GSP       read control array A;  initialize plotter.
PLOT    23645 24375    GSP                     move pen to a defined point.
SETSM   24376 25336    GSP                          change plotter modes.
STOPG   25337 25401    GSP                              terminate grafics.
SYMBL   25402 26136    GSP              print symbol at location of pen.
FCTR    26137 26142    GSP   / FCTR   / FX,FY
FACTR   26143 26162    GSP                          vary size of the plot.

LOGLU   26163 26240   92067-16268 REV.1903 790228
READF   26241 27202   92067-16125 REV.1940 790719
OPEN    27203 27500   92067-16125 REV.1903 790215
CLOSE   27501 27710   92067-16125 REV.1903 781229
CLRIO   27711 27717   750701   24998-16001
OVRD.   27720 27720   92067-16125 REV.1903 780526
$SMVE   27721 30007   92067-16268 REV.1903 790202
LURQ    30010 30372   92067-16268 REV.1903 790223
.DADS   30373 30502    780818   24998-16001
.DMP    30503 30650    780818   24998-16001
.DDI    30651 31151    781021   24998-16001
SESSN   31152 31167   92067-16125 REV.1903 780413
R/W$    31170 31326   92067-16125 REV.1903 781214
P.PAS   31327 31355   92067-16125 REV.1903 740801
.DNG    31356 31365    780818   24998-16001
PAUSE   31366 31466   771122   24998-16001
$ALRN   31467 31604   92067-16268 REV.1903 770715
FMTIO   31605 33103    24998-16002 REV.1926 790417
ERR0    33104 33173    771122   24998-16001
TAN     33174 33300    780424   24998-16001
ABS     33301 33307      750701   24998-16001
.SNCS   33310 33451    780424   24998-16001
.DDE    33452 33463    780818   24998-16001
.DIN    33464 33471    780818   24998-16001
.RTOI   33472 33565    780921   24998-16001
.FPWR   33566 33627    781106   24998-16001
.SBT    33630 33670    770518   24998-16001
..FCM   33671 33705   750701   24998-16001
PAU.E   33706 33706   750701   24998-16001
ER0.E   33707 33707   750701   24998-16001
.CMRS   33710 33773    780424   24998-16001
$OPEN   33774 34150   92067-16125 REV.1903 790103
RW$UB   34151 34516   92067-16125 REV.1903 781003
RWND$   34517 34641   92067-16125 REV.1903 780801
FRMTR   34642 40277    24998-16002 REV.1926 790503
FMT.E   40300 40300    24998-16002 REV.1901 781107
IB4A2   40301 41601   59310-1X013 REV 1940 790802 1153
REIO    41602 41726   92067-16268 REV.1903 790316
RMPAR   41727 41771   781106   24998-16001
PNAME   41772 42037    771121   24998-16001
LUTRU   42040 42146   92067-16268 REV.1903 790223
IPUT    42147 42167   92067-16125 REV.1903 740801
$SETP   42170 42214    781106   24998-16001
.CFER   42215 42272   750701   24998-16001
.LBT    42273 42323    770518   24998-16001

15 PAGES RELOCATED    15 PAGES REQ'D    NO PAGES EMA       NO PAGES MSEG
LINKS:BP     PROGRAM:LB    LOAD:TE    COMMON:NC
 /LOADR:DEMO2  READY AT 12:44 PM  THU., 18  SEPT, 1980

/LOADR:$END
```

2.6.3. Results

Affect of IPRINT on quantity of print out

i) printed output

System matrix A and vector B before Gauss Jordan Elimation

	1	2	3	4	5	
1	.510E+02	-.102E+03	.487E+03	-.211E+04	.104E+05	-.578E+02
2	-.102E+03	.487E+03	-.211E+04	.104E+05	-.525E+05	.998E+02
3	.487E+03	-.211E+04	.104E+05	-.525E+05	.274E+06	-.113E+03
4	-.211E+04	.104E+05	-.525E+05	.274E+06	-.146E+07	-.944E+03
5	.104E+05	-.525E+05	.274E+06	-.146E+07	.787E+07	.113E+05
	1	2	3	4	5	

Equation system after Gauss Jordan Elimination

	1	2	3	4	5	
1	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	-.970E+00
2	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	-.416E+00
3	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	-.162E-01
4	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.275E+00
5	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.514E-01
	1	2	3	4	5	

Coefficients COEF(I)

I = 1 -.970E+00
I = 2 -.416E+00
I = 3 -.162E-01
I = 4 .275E+00
I = 5 .514E-01

IPRINT = 2

Equation system after Gauss Jordan Elimination

	1	2	3	4	5	
1	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	-.970E+00
2	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	-.416E+00
3	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	-.162E-01
4	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.275E+00
5	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.514E-01
	1	2	3	4	5	

Coefficients COEF(I)

I = 1 -.970E+00
I = 2 -.416E+00
I = 3 -.162E-01
I = 4 .275E+00
I = 5 .514E-01

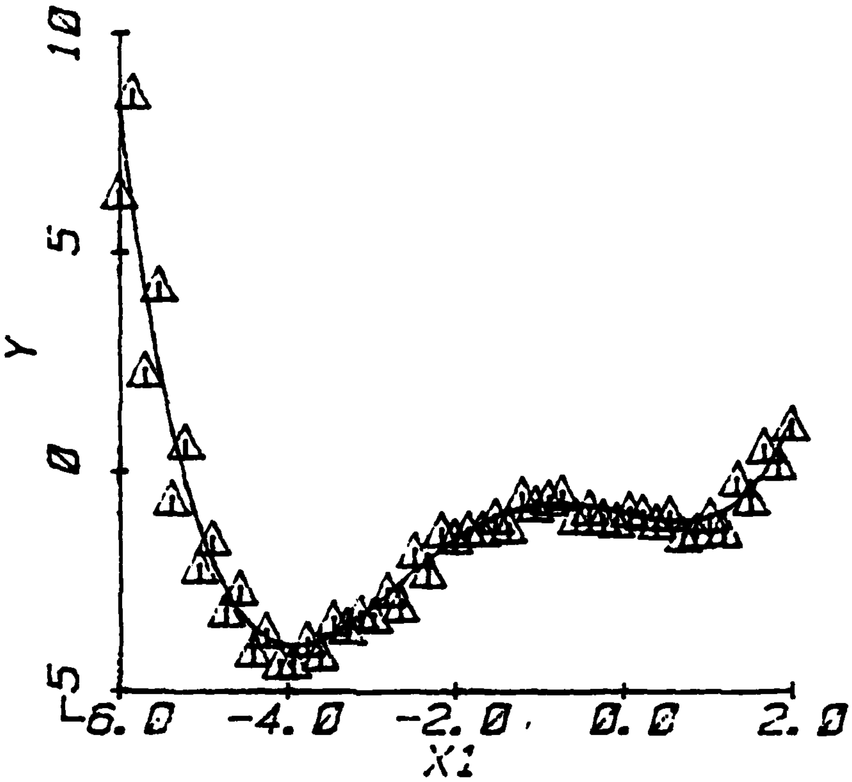
IPRINT = 1

Coefficients COEF(I)

I = 1 -.970E+00
I = 2 -.416E+00
I = 3 -.162E-01
I = 4 .275E+00
I = 5 .514E-01

IPRINT < 1
> 2

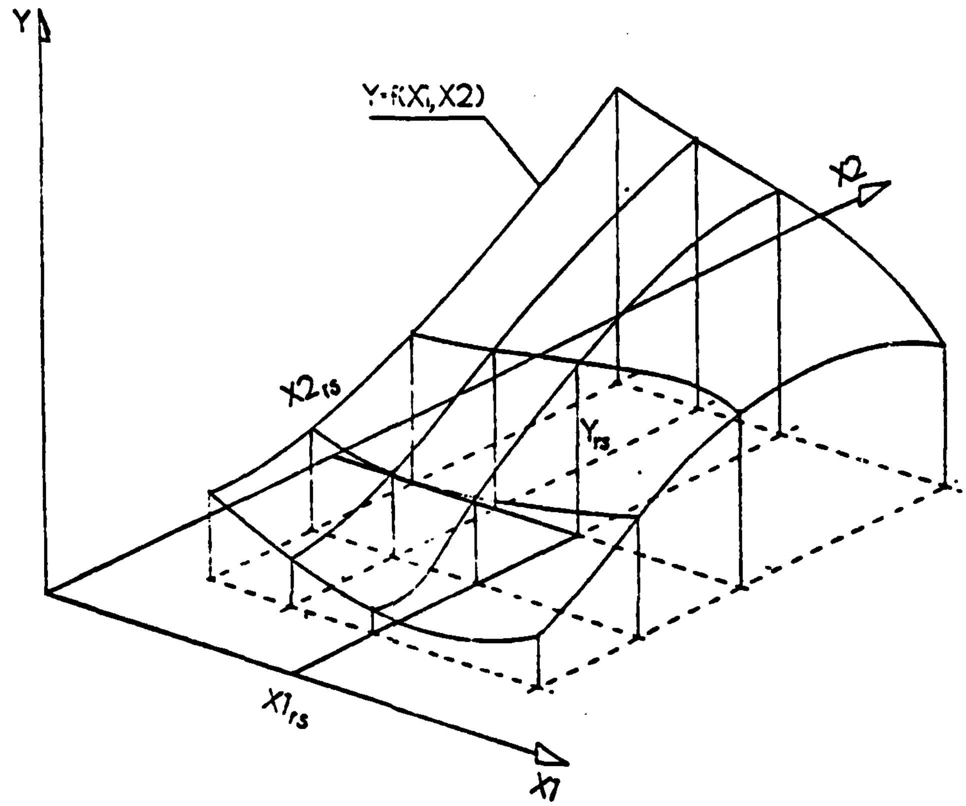
ii) graphic output



Grafic output from PROGRAM DEMO2

3. THREE DIMENSIONAL APPROXIMATION

3.1. Problem:



A data set of NPNTS1*NPNTS2 data points is given, where Y depends on the parameters X1 and X2. The data pattern is to be approximated by a function Y=f(X1, X2), so that the error between data points and analytically determined points is lowest.

3.2. Approach:

As for the two dimensional approximation, a polynominal, now with two independent variables X1 and X2, is used.

$$Y = f(X1, X2)$$

$$Y = C_{11} + C_{12} \cdot X2 + C_{13} \cdot X2^2 + \ldots + C_{1M} \cdot X2^{(M-1)} +$$

$$+ [C_{21} + C_{22} \cdot X2 + C_{23} \cdot X2^2 + \ldots + C_{2M} X2^{(M-1)}] \cdot X1 +$$

$$+ [C_{31} + C_{32} X2 + C_{33} X2^2 + \ldots + C_{3M} X2^{(M-1)}] \cdot X1^2 +$$

$$\ldots\ldots$$

$$+ [C_{L1} + C_{L2} X2 + C_{L3} X2^2 + \ldots + C_{LM} X2^{(M-1)}] \cdot X1^{(L-1)}$$

$$Y = \sum_{i=1}^{L} \{ \sum_{j=1}^{M} C_{ij} X2^{(j-1)} \} \cdot X1^{(i-1)} \qquad (3.1.)$$

The data arrangement for the three dimensional approximation already yields to the application of this method: at each constant X1 (i.e.: r = constant, too) there is a data set for various X2. This is the case for example, when a pneumatic-velocity probe is calibrated. At each Mach number, the probe is balanced in yaw and the pitch angle is varied over the range of application. The probe output (each of two separate non-dimensional pressure differences) is a function of the Mach number and the pitch angle.

The least squares criterion becomes

$$R = \sum_{r=1}^{NPNTS1} \cdot \left\{ \sum_{s=1}^{NPNTS2} \left[f(X1_{rs}, X2_{rs}) - Y_{rs} \right]^2 \right\} \quad (3.2.)$$

where the indices r and s denote the individual data points. Using equation (2.1.) R becomes

$$\begin{aligned} R = \sum_{r=1}^{NPNTS1} \cdot \Big\{ \sum_{s=1}^{NPNTS2} \Big[\; & C_{11} + C_{12} X2_{rs} + \ldots + C_{1M} X2_{rs}^{M-1} + \\ & + (C_{21} + C_{22} X2_{rs} + \ldots + C_{2M} X2_{rs}^{M-1}) \cdot X1_{rs} \\ & + (C_{31} + C_{32} X2_{rs} + \ldots + C_{3M} X2_{rs}^{M-1}) \cdot X1_{rs}^{2} + \\ & \ldots \\ & + (C_{L1} + C_{L2} X2_{rs} + \ldots + C_{LM} X2_{rs}^{M-1}) \cdot X1_{rs}^{L-1} + \\ & - Y_{rs} \Big]^2 \Big\} \end{aligned}$$

The term in the [ ] - bracket shall be called B.

$$R = \sum_{r=1}^{NPNTS1} \cdot \left\{ \sum_{s=1}^{NPNTS2} \cdot B^2 \right\}$$

3.3. <u>Solution:</u>

To minimize the error, R is partially differentiated to the coefficients C_{ij} and then the partial derivatives are set to zero

$$\frac{\partial R}{\partial C_{ij}} \overset{!}{=} 0 \qquad i = 1,\ldots,L; \qquad j = 1,\ldots,M$$

$$\frac{\partial R}{\partial C_{ij}} = \sum_{r=1}^{NPNTS1} \cdot \{ \sum_{s=1}^{NPNTS2} 2 \cdot \delta \cdot \frac{\partial \delta}{\partial C_{ij}} \} = 0$$

Assuming, that the summations extend over all data points, $\Sigma\Sigma$ should be understood to mean $\sum_{r=1}^{NPNTS1} \cdot \{ \sum_{s=1}^{NPNTS2}$. Performing the differentiations and rearranging the equations in matrix notation (which, while lengthy, is logically straight forward), we get, in matrix notation,

$$
\begin{array}{cc|cccc|cccc|}
 & & \multicolumn{4}{c|}{\overbrace{\hspace{18em}}^{1}} & \multicolumn{4}{c|}{\overbrace{\hspace{18em}}^{2}} \\
 & & (1-1)*M+1 & & & 1*M & (2-1)*M+1 & & & 2*M \\
\hline
 & (1-1)*M+1 & NPNTS1*NPNTS2 & \Sigma\Sigma X2_{rs} & \cdots & \Sigma\Sigma X2_{rs}^{M-1} & \Sigma\Sigma X1_{rs} & \Sigma\Sigma X1_{rs}\cdot X2_{rs} & \cdots & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{M-1} \\
1 & \vdots & \Sigma\Sigma X2_{rs} & \Sigma\Sigma X2_{rs}^{2} & \cdots & \Sigma\Sigma X2_{rs}^{M} & \Sigma\Sigma X1_{rs}\cdot X2 & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{2} & \cdots & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{M} \\
 & & \cdots & & & \cdots & \cdots & & & \\
 & 1*M & \Sigma\Sigma X2_{rs}^{M-1} & \Sigma\Sigma X2_{rs}^{M} & \cdots & \Sigma\Sigma X2_{rs}^{2M-2} & \Sigma\Sigma X1_{rs}\cdot X2^{M-1} & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{M} & \cdots & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{2M-2} \\
\hline
 & (2-1)*M & \Sigma\Sigma X1_{rs} & \Sigma\Sigma X1_{rs}\cdot X2_{rs} & \cdots & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{M-1} & \Sigma\Sigma X1_{rs}^{2} & \Sigma\Sigma X1_{rs}^{2}\cdot X2 & \cdots & \Sigma\Sigma X1_{rs}^{2}\cdot X2_{rs}^{M-1} \\
2 & \vdots & \Sigma\Sigma X1_{rs}\cdot X2_{rs} & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{2} & \cdots & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{M} & \Sigma\Sigma X1_{rs}^{2}\cdot X2 & \Sigma\Sigma X1_{rs}^{2}\cdot X2^{2} & \cdots & \Sigma\Sigma X1_{rs}^{2}\cdot X2_{rs}^{M} \\
 & & \cdots & & & \cdots & \cdots & & & \\
 & 2*M & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{M-1} & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{M} & \cdots & \Sigma\Sigma X1_{rs}\cdot X2_{rs}^{2M-2} & \Sigma\Sigma X1_{rs}^{2}\cdot X2^{M-1} & \Sigma\Sigma X1_{rs}^{2}\cdot X2^{M} & \cdots & \Sigma\Sigma X1_{rs}^{2}\cdot X2_{rs}^{2M-2} \\
\hline
 & \vdots & \cdots & & & & & & & \\
\hline
 & (L-1)*M & \Sigma\Sigma X1_{rs}^{L-1} & \Sigma\Sigma X1_{rs}^{L-1}\cdot X2 & \cdots & \Sigma\Sigma X1_{rs}^{L-1}\cdot X2^{M-1} & \Sigma\Sigma X1_{rs}^{L} & \Sigma\Sigma X1_{rs}^{L}\cdot X2_{rs} & \cdots & \Sigma\Sigma X1_{rs}^{L}\cdot X2_{rs} \\
L & \vdots & \Sigma\Sigma X1_{rs}^{L-1}\cdot X2 & \Sigma\Sigma X1_{rs}^{L-1}\cdot X2^{2} & \cdots & \Sigma\Sigma X1_{rs}^{L-1}\cdot X2^{M} & \Sigma\Sigma X1_{rs}^{L}\cdot X2_{rs} & \Sigma\Sigma X1_{rs}^{L}\cdot X2_{rs} & \cdots & \Sigma\Sigma X1_{rs}^{L}\cdot X2_{rs} \\
 & & \cdots & & & \cdots & \cdots & & & \\
 & L*M & \Sigma\Sigma X1_{rs}^{L-1}\cdot X2^{M-1} & \Sigma\Sigma X1_{rs}^{L-1}\cdot X2^{M} & \cdots & \Sigma\Sigma X1_{rs}^{L-1}\cdot X2^{2M-2} & \Sigma\Sigma X1_{rs}^{L}\cdot X2_{rs} & \Sigma\Sigma X1_{rs}^{L}\cdot X2_{rs} & \cdots & \Sigma\Sigma X1_{rs}^{L}\cdot X2_{rs} \\
\end{array}
$$

A

$$
\begin{array}{ccccccccc}
 & 2*M & & (L-1)*M+1 & & & L*M & & \\
\cdots & \sum\sum X1_{rs} \cdot X2_{rs}^{M-1} & & \sum\sum X1_{rs}^{L-1} & \sum\sum X1_{rs}^{L-1} \cdot X2_{rs} & \cdots & \sum\sum X1_{rs}^{L-1} \cdot X2_{rs}^{M-1} & C_{11} & \sum\sum Y_{rs} \\
\cdots & \sum\sum X1_{rs} \cdot X2_{rs}^{M} & \cdots & \sum\sum X1_{rs}^{L-1} \cdot X2_{rs} & \sum\sum X1_{rs}^{L-1} \cdot X2_{rs}^{2} & \cdots & \sum\sum X1_{rs}^{L-1} \cdot X2_{rs}^{M} & C_{12} & \sum\sum Y_{rs} \cdot X2_{rs} \\
 & & & \cdots & & & & \vdots & \vdots \\
\cdots & \sum\sum X1_{rs} \cdot X2_{rs}^{2M-2} & & \sum\sum X1_{rs}^{L-1} \cdot X2_{rs}^{M-1} & \sum\sum X1_{rs}^{L-1} \cdot X2_{rs}^{M} & \cdots & \sum\sum X1_{rs}^{L-1} \cdot X2_{rs}^{2M-2} & C_{1M} & \sum\sum Y_{rs} \cdot X2_{rs}^{M-1} \\
\hline
\cdots & \sum\sum X1_{rs}^{2} \cdot X2_{rs}^{M-1} & & \sum\sum X1_{rs}^{L} & \sum\sum X1_{rs}^{L} \cdot X2_{rs} & \cdots & \sum\sum X1_{rs}^{L} \cdot X2_{rs}^{M-1} & C_{21} & \sum\sum Y_{rs} \cdot X1_{rs} \\
\cdots & \sum\sum X1_{rs}^{2} \cdot X2_{rs}^{M} & \cdots & \sum\sum X1_{rs}^{L} \cdot X2_{rs} & \sum\sum X1_{rs}^{L} \cdot X2_{rs}^{2} & \cdots & \sum\sum X1_{rs}^{L} \cdot X2_{rs}^{M} & C_{22} & \sum\sum Y_{rs} \cdot X2_{rs} \cdot X1_{rs} \\
 & & & \cdots & & & & \vdots & \vdots \\
\cdots & \sum\sum X1_{rs}^{2} \cdot X2_{rs}^{2M-2} & & \sum\sum X1_{rs}^{L} \cdot X2_{rs}^{M-1} & \sum\sum X1_{rs}^{L} \cdot X2_{rs}^{M} & \cdots & \sum\sum X1_{rs}^{L} \cdot X2_{rs}^{2M-2} & C_{2M} & \sum\sum Y_{rs} \cdot X2_{rs}^{M-1} \cdot X1_{rs} \\
\hline
 & & & & & & & \vdots & \vdots \\
\hline
\cdots & \sum\sum X1_{rs}^{L} \cdot X2_{rs} & & \sum\sum X1_{rs}^{2L-2} & \sum\sum X1_{rs}^{2L-2} \cdot X2_{rs} & \cdots & \sum\sum X1_{rs}^{2L-2} \cdot X2_{rs}^{M-1} & C_{L1} & \sum\sum Y_{rs} \cdot X1_{rs}^{L-1} \\
\cdots & \sum\sum X1_{rs}^{L} \cdot X2_{rs} & \cdots & \sum\sum X1_{rs}^{2L-2} \cdot X2_{rs} & \sum\sum X1_{rs}^{2L-2} \cdot X2_{rs}^{2} & \cdots & \sum\sum X1_{rs}^{2L-2} \cdot X2_{rs}^{M} & C_{L2} & \sum\sum Y_{rs} \cdot X2_{rs} \cdot X1_{rs}^{L-1} \\
 & & & \cdots & & & & \vdots & \vdots \\
\cdots & \sum\sum X1_{rs}^{L} \cdot X2_{rs} & & \sum\sum X1_{rs}^{2L-2} \cdot X2_{rs}^{M-1} & \sum\sum X1_{rs}^{2L-2} \cdot X2_{rs}^{M} & \cdots & \sum\sum X1_{rs}^{2L-2} \cdot X2_{rs}^{2M-2} & C_{LM} & \sum\sum Y_{rs} \cdot X2_{rs}^{M-1} \cdot X1_{rs}^{L-1}
\end{array}
$$

$$* \ C = B \qquad (3.3.)$$

2

3.4. Equation System:

3.4.1. System Matrix A

This system matrix needs some examination. A crucial step is the introduction of submatrices 1. class into the system matrix.

$$
\mathbb{A} = \begin{vmatrix} \mathbb{A}^{*}_{11} & \mathbb{A}^{*}_{12} & & \mathbb{A}^{*}_{1L} \\ \mathbb{A}^{*}_{21} & \mathbb{A}^{*}_{22} & & \mathbb{A}^{*}_{2L} \\ & & & \\ \mathbb{A}^{*}_{L1} & \mathbb{A}^{*}_{L2} & & \mathbb{A}^{*}_{LL} \end{vmatrix}
$$

The asterisk * denotes a submatrix 1. class where

$$
\mathbb{A}^{*}_{11} = \begin{vmatrix} NPNTS1 * NPNTS2 & \sum\sum X2_{rs} & & \sum\sum X2_{rs}^{M-1} \\ \sum\sum X2_{rs} & \sum\sum X2_{rs}^{2} & \cdot & \sum\sum X2_{rs}^{M} \\ & & & \\ \sum\sum X2_{rs}^{M-1} & \sum\sum X2_{rs}^{M} & \cdot & \sum\sum X2_{rs}^{2M-2} \end{vmatrix}
$$

$$
\mathbb{A}^{*}_{12} = \begin{vmatrix} \sum\sum X1_{rs} & \sum\sum X1_{rs} X2_{rs} & & \sum\sum X1_{rs} X2_{rs}^{M-1} \\ \sum\sum X1_{rs} X2_{rs} & \sum\sum X1_{rs} X2_{rs}^{2} & & \sum\sum X1_{rs} X2_{rs}^{M} \\ & & & \\ \sum\sum X1_{rs} X2_{rs}^{M-1} & \sum\sum X1_{rs} X2_{rs}^{M} & & \sum\sum X1_{rs} X2_{rs}^{2M-2} \end{vmatrix}
$$

$$
\mathbb{A}^{*}_{13} = \begin{vmatrix} \sum\sum X1_{rs}^{2} & \sum\sum X1_{rs}^{2} X2_{rs} & & \sum\sum X1_{rs}^{2} X2_{rs}^{M-1} \\ \sum\sum X1_{rs}^{2} X2_{rs} & \sum\sum X1_{rs}^{2} X2_{rs}^{2} & & \sum\sum X1_{rs}^{2} X2_{rs}^{M} \\ & & & \\ \sum\sum X1_{rs}^{2} X2_{rs}^{M-1} & \sum\sum X1_{rs}^{2} X2_{rs}^{M} & & \sum\sum X1_{rs}^{2} X2_{rs}^{2M-2} \end{vmatrix}
$$

etc.

Now it can seem that all submatrices on the same diagonal bands 2. order are identical. The submatrices A^*_{ij} subsequently are renamed A^*_k

$$
\begin{array}{llllll}
A^*_1 = A^*_{11} \\
A^*_2 = A^*_{21} = A^*_{12} \\
A^*_3 = A^*_{31} = A^*_{22} = A^*_{13} \\
A^*_4 = A^*_{41} = A^*_{32} = A^*_{23} = A^*_{14} \\
\vdots \\
A^*_L = A^*_{L1} = A^*_{L-1\,2} = A^*_{L-2\,3} = & \cdots & = A^*_{2\,L-1} & = A^*_{1\,L} \\
A^*_{L+1} \quad = A^*_{L,2} = A^*_{L-1,3} = & \cdots & = A^*_{3\,L-1} & = A^*_{2,L} \\
A^*_{L+2} \quad\quad = A^*_{L,3} = & \cdots & = A^*_{4\,L-1} & = A^*_{3,L} \\
\vdots \\
A^*_{2L-1} & & & = A^*_{LL}
\end{array}
$$

where A^*_k can be written as

$$
A^*_K = \left|\begin{array}{cccc}
\sum\sum X1^{k-1}_{rs} X2^{0}_{rs} & \sum\sum X1^{k-1}_{rs} X2^{1}_{rs} & & \sum\sum X1^{k-1}_{rs} X2^{M-1}_{rs} \\
\sum\sum X1^{k-1}_{rs} X2^{1}_{rs} & \sum\sum X1^{k-1}_{rs} X2^{2}_{rs} & & \sum\sum X1^{k-1}_{rs} X2^{M}_{rs} \\
 & & & \\
\sum\sum X1^{k-1}_{rs} X2^{M-1}_{rs} & \sum\sum X1^{k-1}_{rs} X2^{M}_{rs} & & \sum\sum X1^{k-1}_{rs} X2^{2M-2}_{rs}
\end{array}\right|
$$

$k = 1, \ldots 2L-1$

Like the system matrix for the two dimensional approximation this submatrix 1. class is not only symmetrical to its main diagonal line 1. order, but also the elements on each diagonal line 2. order are identical. The elements $a_{k;ij}$ of the submatrix A_k therefore are renamed to $a_{k;L}$, where k specifies the place of the submatrix in the system matrix and i and j define the place of $a_{k;ij}$ in the submatrix A_k^*.

$$
\begin{array}{llllllll}
a_{k;1} & = a_{k;11} & & & & & & \\
a_{k;2} & = a_{k;21} & = a_{k;12} & & & & & \\
a_{k;3} & = a_{k;31} & = a_{k;22} & = a_{k;13} & & & & \\
\vdots & & & & & & & \\
a_{k;M} & = a_{k;M1} & = a_{k;M-1\,2} & = a_{k;M-2\,3} & = \cdots & = a_{k;2\,M-1} & = a_{k;1M} & \\
a_{k;M+1} & & = a_{k;M2} & = a_{k;M-1\,3} & = \cdots & = a_{k;3\,M-1} & = a_{k;2M} & \\
a_{k;M+2} & & & = a_{k;M3} & = \cdots & = a_{k;4\,M-1} & = a_{k;3M} & \\
\vdots & & & & & & & \\
a_{k;2M-1} & & & & & & = a_{k;MM} &
\end{array}
$$

where $a_{k;L}$ can be written as

$$
a_{k;L} = \sum_{r=1}^{NPNTS1}\sum_{s=1}^{NPNTS2} X1_{rs}^{k-1}\, X2_{rs}^{L-1} \}
$$

$$
k = 1, \ldots, 2L-1 \;;\; L = 1, \ldots, 2M-1 \qquad (3.4.)
$$

Equation (3.4.) therefore is the fundamental step towards the programming of the system matrix and is programmed in REAL FUNCTION S3. Since the data X1, X2 and Y are already provided through a COMMON block named DATA3, only NPNTS1, NPNTS2, IPOWR1 (= k-1) and IPOWR2 (= L-1) have to be passed to the function that returns the value of $a_{k;L}$ through the function name S3.

SUBROUTINE MAT31 presets a submatrix A_k^* in the following way:

i) <u>Preset edge section elements</u> (using REAL FUNCTION S3)

$$\left|\begin{array}{ccccc} a_{k;1} \Rightarrow & a_{k;2} \Rightarrow & a_{k;3} \Rightarrow & \ldots \Rightarrow & a_{k;M} \\ & & & & \Downarrow \\ & & & & a_{k;M+1} \\ & & & & \Downarrow \\ & & & & a_{k;M+2} \\ & & & & \Downarrow \\ & & & & \vdots \\ & & & & \Downarrow \\ & & & & a_{k;2M-1} \end{array}\right|$$

$a_{k,1} \rightarrow a_{k,M}$ Top edge section

$a_{k,M+1} \rightarrow a_{k,2M-1}$ right hand edge section

ii) <u>Copy defined elements diagonally</u>

$$\left|\begin{array}{ccccc} a_{k;11} & a_{k;12} & a_{k;13} & \cdots & a_{k;1M} \\ & & & & a_{k;2M} \\ & & & & a_{k;3M} \\ & & & & \cdots \\ & & & & a_{k;MM} \end{array}\right|$$

$$\left|\begin{array}{ccccc} a_{k;11} & a_{k;12} & a_{k;13} & \cdots & a_{k;1M} \\ a_{k;21} & a_{k;22} & a_{k;23} & \cdots & a_{k;2M} \\ & & & & a_{k;3M} \\ & & & & \cdots \\ & & & & a_{k;MM} \end{array}\right|$$

$$\left|\begin{array}{ccccc} a_{k;11} & a_{k;12} & a_{k;13} & \cdots & a_{k;1M} \\ a_{k;21} & a_{k;22} & a_{k;23} & \cdots & a_{k;2M} \\ a_{k;31} & a_{k;32} & a_{k;33} & \cdots & a_{k;3M} \\ & & & & \cdots \\ & & & & a_{k;MM} \end{array}\right|$$

$$
\begin{vmatrix}
a_{k;11} & a_{k;12} & a_{k;13} & \cdots & a_{k;1M} \\
a_{k;21} & a_{k;22} & a_{k;23} & \cdots & a_{k;2M} \\
a_{k;31} & a_{k;32} & a_{k;33} & \cdots & a_{k;3M} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\swarrow & \swarrow & \swarrow & \swarrow & a_{k;MM}
\end{vmatrix}
$$

$$
A_k^* = \begin{vmatrix}
a_{k;11} & a_{k;12} & a_{k;13} & \cdots & a_{k;1M} \\
a_{k;21} & a_{k;22} & a_{k;23} & \cdots & a_{k;2M} \\
a_{k;31} & a_{k;32} & a_{k;33} & \cdots & a_{k;3M} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
a_{k;M1} & a_{k;M2} & a_{k;M3} & \cdots & a_{k;MM}
\end{vmatrix}
$$

This subroutine that needs NPNTS1, NPNTS2, M and IPOWR1 (= k-1) as input parameters, returns A_k^* (SUBM1(7,7)) to SUBROUTINE MAT3, which presets the entire system matrix in the following way

i) <u>Preset edge section submatrices</u> (using SUBROUTINE MAT31)

$$
\left|\begin{array}{ccccc}
A_1^* & \Rightarrow A_2^* & \Rightarrow \cdots & \Rightarrow & A_L^* \\
 & & & & \Downarrow \\
 & & & & A_{L+1}^* \\
 & & & & \Downarrow \\
 & & & & \vdots \\
 & & & & \Downarrow \\
 & & & & A_{2L-1}^*
\end{array}\right|
$$

$A_1^* \rightarrow A_L^*$
Top edge section

$A_{L+1}^* \rightarrow A_{2L-1}^*$
right hand edge section

ii) Copy defined submatrices diagonally

$$\left|\begin{array}{cccc} A_{11}^{*} & A_{12}^{*} & \cdots & A_{1L}^{*} \\ & & & A_{2L}^{*} \\ & & & \cdots \\ & & & A_{LL}^{*} \end{array}\right|$$

$$\left|\begin{array}{cccc} A_{11}^{*} & A_{12}^{*} & \cdots & A_{1L}^{*} \\ A_{21}^{*} & A_{22}^{*} & \cdots & A_{2L}^{*} \\ & & & \cdots \\ & & & A_{LL}^{*} \end{array}\right|$$

$$\left|\begin{array}{cccc} A_{11}^{*} & A_{12}^{*} & \cdots & A_{1L}^{*} \\ A_{21}^{*} & A_{22}^{*} & \cdots & A_{2L}^{*} \\ \cdots & \cdots & \cdots & \cdots \\ & & & A_{LL}^{*} \end{array}\right|$$

$$A = \left|\begin{array}{cccc} A_{11}^{*} & A_{12}^{*} & \cdots & A_{1L}^{*} \\ A_{21}^{*} & A_{22}^{*} & \cdots & A_{2L}^{*} \\ \cdots & \cdots & \cdots & \cdots \\ A_{L1}^{*} & A_{L2}^{*} & \cdots & A_{LL} \end{array}\right|$$

The parameters to call this subroutine are NPNTS1 and NPNTS2 (# of data points), L and M (<the order of the polynominal fit + 1> in X1 and X2 - direction) and IPRINT. The coefficients C_{ij} are returned.

3.4.2. Right hand side vector B

The right hand side vector B will be divided up into L subvectors 1. class.

$$\mathcal{B} = \begin{vmatrix} \mathcal{B}_1^* \\ \mathcal{B}_2^* \\ \cdots \\ \mathcal{B}_L^* \end{vmatrix}$$

Again, the asterisk (*) denotes a subvector 1. class. Now the subvectors 1. class B_k^* can be written as

$$\mathcal{B}_k^* = \begin{vmatrix} \sum\sum Y_{rs} \cdot X1_{rs}^{k-1} \\ \sum\sum Y_{rs} \cdot X1_{rs}^{k-1} \cdot X2_{rs} \\ \cdots \\ \sum\sum Y_{rs} \cdot X1_{rs}^{k-1} \cdot X2_{rs}^{M} \end{vmatrix}$$

$$k = 1, \ldots, L$$

Finally, the elements of the subvectors 1. class B_k^* can be written as

$$b_{k;l} = \sum\sum Y_{rs} \cdot X1_{rs}^{k-1} \cdot X2_{rs}^{l-1}$$

$$k = 1, \ldots, L; \quad l = 1, \ldots, M$$

To compute $b_{k;L}$, REAL FUNCTION S3 is used. Data X1, X2 and Y are available through a COMMON block named DATA3. NPNTS1, NPNTS2, IPOWR1 (= k-1), IPOWR2 (= L-1) and IY (=1) have to be passed to the function and the value of $b_{k;L}$ is returned through the function name S3.

Using FORTRAN programming language, the allocation of $b_{k;L}$ can be performed by two stacked DO loops.

```
    I=0
    DO 08 L1-1,L,1
    IPOWR1=L1-1
    DO 08 M1=1,M,1
    IPOWR2=M1-1
    I=I+1
 08 B(I) = S3(NPNTS1,NPNTS2,IPOWR1,IPOWR2,1)
```

3.5. Software:

The software to compute the coefficients for a three dimensional approximation is described in Appendix A and is implemented in the TPL HP-21MX computer system. To work correctly with these program modules, the user has to conform to the following conventions:

i) Provide the data in three arrays (Type: REAL) of 16 * 16 elements through a COMMON block named DATA3.

```
COMMON / DATA3 / X1, X2, Y
REAL X1 (16,16), X2(16,16), Y(16,16)
```

ii) Dimension an array (Type: REAL) of 7*7 elements to contain the coefficients.

```
REAL COEF (7,7)
```

iii) Define the parameters NPNTS1, NPNTS2, L, M and IPRINT.

NPNTS1 ... # of X1 variations
$L \leq NPNTS1 \leq 16$

NPNTS2 ... # of X2 variations
$M \leq NPNTS2 \leq 16$

L ... (desired order of approximation polynominal w.r.t. X1) + 1

M ... (desired order of approximation polynominal w.r.t. X2) + 1

IPRINT ... controls quantity of print out

- 2 ... Print system matrix and right hand side vector before and after Gauss Jordan Elimination
- 1 ... Print equation system after Gauss Jordan Elimination
- <0 ... No print out
- >0 ... Display equation (3.1.) with the actual parameters

iv) If a user program uses subroutine MAT3, the software modules have to be loaded using the procedure outlined in section 2.4. iv.

If all these requirements are met, the correct call for the subroutine is:

```
CALL MAT (NPNTS1,NPNTS2,L,M,COEF,IPRINT)
```

Upon completed execution of this approximation routine the array COEF contains the coefficients. Externals used by MAT3 are: AB3, DATA3, MAT31, IEL3, S3. Under no circumstances may the user use any of these names for modules of his own program.

3.6. Sample User Program

3.6.1. FTN4 Compiler listing of sample user program

```
FTN4,L
      PROGRAM LAB (3,99)
     *,Plot calibration points and calculate coefficients.

      COMMON / AFLD / PLOTR
      COMMON / DATA3 / X1,X2,Y
      COMMON / AB3   / A,B

      REAL PLOTR(256)
      REAL X1(16,16),X2(16,16),Y(16,16)
      REAL A(49,49),B(49)

      REAL YT(16,16),R(16),COEFF(7,7)
      INTEGER IDCB(144),IFILE(3),NOCR(2),ICLR(3),YTITLE(4),ZTITLE(4)

      DATA PI /3.141593/
      DATA NOLF /006537B/
      DATA NOCR /000033B,040433B/
      DATA ICLR /015524B,015515B,006537B/

C     FORMATS LAB START
  121 FORMAT (" Enter raw data file name "2A2)
  122 FORMAT (3A2)
  123 FORMAT (" Select 1 ... Pitch angle    or  2 .
     *.. Velocity X    "2A2)
  124 FORMAT (" Select 1 ... Pitch angle v/s Gamma and Beta   (
     * = ( , ) )"/
     *      "              2 ... ",1H","      ",1H","      v/s Gamma a
     *nd Delta  (  = ( , ) )  "2A2)
  125 FORMAT (" Select 1 ... Velocity v/s Gamma and Beta   ( X=
     *X( , ) )"/
     *      "              2 ... ",1H","            v/s Gamma and Delta
     *   ( X=X( , ) )  "2A2)
  126 FORMAT (" Enter # of Mach Numbers and # of pitch angle
     *s   "2A2)
  160 FORMAT (" Enter 1 to print input data "2A2)
  131 FORMAT (/" Calibrate   = ( , )"/)
  132 FORMAT (/" Calibrate   = ( , )"/)
  133 FORMAT (/" Calibrate X=X( , )"/)
  134 FORMAT (/" Calibrate X=X( , )"/)
  127 FORMAT (24X" "9X" "9X" ")
  128 FORMAT (24X" "9X" "9X" ")
  129 FORMAT (24X" "9X" "9X"X")
  130 FORMAT (24X" "9X" "9X"X")
  150 FORMAT (" minimum values",3(1X,F9.3))
  151 FORMAT (" maximum values",3(1X,F9.3))
  152 FORMAT (" of user coordinates"3X"X1"8X"X2"9X"Y")
  153 FORMAT (" Enter 1 to redefine these data "2A2)
  154 FORMAT (" Enter  min,    max,    min, ",
     *"   max, # partitions   ,    &   ")
  155 FORMAT (" Enter  min,    max,    min, ",
     *"   max, # partitions   ,    &   ")
  156 FORMAT (" Enter    min,    max,    min,    max, X min, X ma
     *x, # partitions   ,    & X")
  157 FORMAT (" Enter    min,    max,    min,    max, X min, X ma
     *x, # partitions   ,    & X")
  158 FORMAT (1X,3(F8.3,1X,F8.3,5X),3I4)
  101 FORMAT (" Enter order M of X1 - approximation "2A2)
  102 FORMAT (" Enter oredr N of X2 - approximation "2A2)
  107 FORMAT (" Enter IPRINT "2A2)
  105 FORMAT (" Select 1 ... to calculate the absolute error"/8X
     *"2 ...  ",4H"   ",9X"the relative error    "2A2)
  103 FORMAT (" Change coefficients    ...   1"/
     *        " Change axes            ...   2  "2A2)
  104 FORMAT (" Enter ALPHAX, ALPHAY, X0 and Y0")
  149 FORMAT(""((3A2)))
  611 FORMAT ("1"////" These data from file "3A
  612 FORMAT (/16X,I2". Mach number"/)
  613 FORMAT (6X"Gamma"6X"Beta"7X"Phi"/)
  614 FORMAT (6X"Gamma"5X"Delta"7X"Phi"/)
  615 FORMAT (6X"Gamma"6X"Beta"5X"X vel"/)
  616 FORMAT (6X"Gamma"5X"Delta"5X"X vel"/)
  617 FORMAT (1X,4(1X,F9.3))
  618 FORMAT (/9X"X1"8X"X2"9X"Y")
  602 FORMAT ((3X,16(6X,I2)))
  603 FORMAT (1X,I2,16(1X,F7.3)/3(3X,16(1X,F7.3)/))
```

```
  604 FORMAT (///" Absolute error at each point   ( error = (calculated
     *value - measured value) )"/)
  644 FORMAT (///" Relative error at each point in %   ( error = (calcul
     *ated value - measured value)/measured value )"/)
  606 FORMAT (1X,I2,18(1X,F6.1)/3(3X,18(1X,F7.3)/))
 1101 FORMAT (8H  Beta   )
 1102 FORMAT (8H  Delta )
 1103 FORMAT (8H  Phi    )
 1104 FORMAT (8H  X vel )
      LI = LOGLU(I)
      LO = 6



C     ..................................................................
C     .                                                                .
C     .  read raw data file.                                           .
C     .                                                                .
C     ..................................................................
   41 WRITE (LI, 121) NOCR
      READ  (LI, 122) IFILE
      WRITE (LI, 149) ICLR
   42 WRITE (LI, 123) NOCR
      READ  (LI,   *) IPX
      WRITE (LI, 149) ICLR
      IF (IPX.LT.1 .OR. IPX.GT.2) GO TO 42
   43 CONTINUE
      IF (IPX.EQ.1) WRITE (LI, 124) NOCR
      IF (IPX.EQ.2) WRITE (LI, 125) NOCR
      READ  (LI,   *) IBD
      WRITE (LI, 149) (ICLR,I=1,2,1)
      IF (IBD.LT.1 .OR. IBD.GT.2) GO TO 43
      IF (IPX.EQ.1) IRECY=13
      IF (IPX.EQ.2) IRECY=9
      IF (IPX.EQ.1 .AND. IBD.EQ.1 ) WRITE (LI, 131)
      IF (IPX.EQ.1 .AND. IBD.EQ.2 ) WRITE (LI, 132)
      IF (IPX.EQ.2 .AND. IBD.EQ.1 ) WRITE (LI, 133)
      IF (IPX.EQ.2 .AND. IBD.EQ.2 ) WRITE (LI, 134)
      CALL OPEN (IDCB,IERR,IFILE,IOPTN,0,28,144)
      IF (IERR.LT.0) GO TO 41
      CALL READF (IDCB,IERR,X1,512,LEN,1)
      IF ( IERR  .LT.  0 ) STOP 0002
      CALL READF (IDCB,IERR,X2,512,LEN,5)
      IF ( IERR  .LT.  0 ) STOP 0003
      CALL READF (IDCB,IERR,YT,512,LEN,IRECY)
      IF ( IERR  .LT.  0 ) STOP 0004
      CALL CLOSE (IDCB,IERR,0)
   44 WRITE (LI, 126) NOCR
      READ  (LI,   *) NMACH,NPITCH
      WRITE (LI, 149) ICLR
      IF (NMACH.GT.16) GO TO 44
      IF (NPITCH.GT.16) GO TO 44
      DO 801 I=1,NMACH,1
      DO 801 J=1,NPITCH,1
  801 Y(I,J)=YT(I,J)



C     ..................................................................
C     .                                                                .
C     .  Correct input data and optional output of data, coefficient  .
C     .  are based upon.                                               .
C     .                                                                .
C     ..................................................................
      IF (IBD.EQ.2) GO TO 48
      DO 47 I=1,NMACH,1
      DO 47 J=1,NPITCH,1
   47 X2(I,J)=X2(I,J)/X1(I,J)
   48 WRITE (LI, 160) NOCR
      READ  (LI,   *) IDUM
      WRITE (LI, 149) ICLR
      IF (IDUM.NE.1) GO TO 55
      WRITE (LO, 611) IFILE
      DO 54 I=1,NMACH,1
      WRITE (LO, 612) I
      IF (IPX.EQ.1 .AND. IBD.EQ.1) WRITE (LO, 613)
```

```
      IF (IPX.EQ.1 .AND. IBD.EQ.2) WRITE (LO, 614)
      IF (IPX.EQ.2 .AND. IBD.EQ.1) WRITE (LO, 615)
      IF (IPX.EQ.2 .AND. IBD.EQ.2) WRITE (LO, 616)
      DO 54 J=1,NPITCH,1
   54 WRITE (LO, 617) X1(I,J),X2(I,J),Y(I,J)
      WRITE (LO, 618)



C     .................................................................
C     .                                                               .
C     .  Calculate minimum and maximum values for all coordinates and  .
C     .  optional redefinition of minimum and maximum values by user.  .
C     .                                                               .
C     .................................................................
   55 XUMAX=X1(1,1)
      XUMIN=X1(1,1)
      YUMAX=X2(1,1)
      YUMIN=X2(1,1)
      ZUMAX=Y(1,1)
      ZUMIN=Y(1,1)
      DO 45 I=1,NMACH,1
      DO 45 J=1,NPITCH,1
      IF (X1(I,J).GT.XUMAX) XUMAX=X1(I,J)
      IF (X1(I,J).LT.XUMIN) XUMIN=X1(I,J)
      IF (X2(I,J).GT.YUMAX) YUMAX=X2(I,J)
      IF (X2(I,J).LT.YUMIN) YUMIN=X2(I,J)
      IF (Y(I,J).GT.ZUMAX) ZUMAX=Y(I,J)
      IF (Y(I,J).LT.ZUMIN) ZUMIN=Y(I,J)
   45 CONTINUE
      IF (IPX.EQ.1 .AND. IBD.EQ.1) WRITE (LI, 127)
      IF (IPX.EQ.1 .AND. IBD.EQ.2) WRITE (LI, 128)
      IF (IPX.EQ.2 .AND. IBD.EQ.1) WRITE (LI, 129)
      IF (IPX.EQ.2 .AND. IBD.EQ.2) WRITE (LI, 130)
      WRITE (LI, 150) XUMIN,YUMIN,ZUMIN
      WRITE (LI, 151) XUMAX,YUMAX,ZUMAX
      WRITE (LI, 152)
      INCX = 4
      INCY = 4
      INCZ = 4
      WRITE (LI, 153) NOCR
      READ  (LI,   *) I
      WRITE (LI, 149) ICLR
      IF (I.NE.1) GO TO 49
      IF (IPX.EQ.1 .AND. IBD.EQ.1) WRITE (LI, 154)
      IF (IPX.EQ.1 .AND. IBD.EQ.2) WRITE (LI, 155)
      IF (IPX.EQ.2 .AND. IBD.EQ.1) WRITE (LI, 156)
      IF (IPX.EQ.2 .AND. IBD.EQ.2) WRITE (LI, 157)
      READ  (LI,   *) XUMIN,XUMAX,YUMIN,YUMAX,ZUMIN,ZUMAX,INCX,INCY,INCZ
      WRITE (LI, 158) XUMIN,XUMAX,YUMIN,YUMAX,ZUMIN,ZUMAX,INCX,INCY,INCZ



C     .................................................................
C     .                                                               .
C     .  Initialize plotter;  define plotter area;  calculate scaling  .
C     .  coefficients XA, XB, YA, YB, ZA and ZB.                       .
C     .                                                               .
C     .................................................................
   49 CALL INITG (13)
      XPMIN   = 0.
      XPMAX   = 10.
      YPMIN   = 0.
      YPMAX   = 10.
      ZPMIN   = 0.
      ZPMAX   = 10.
      ALPHAX  = 330.
      ALPHAY  =  30.
      ALPHAZ  =  90.
      XA      = (XPMAX       -XPMIN       )/(XUMAX-XUMIN)
      XB      = (XPMIN*XUMAX-XPMAX*XUMIN)/(XUMAX-XUMIN)
      XL      = (XPMAX       -XPMIN       )
      YA      = (YPMAX       -YPMIN       )/(YUMAX-YUMIN)
      YB      = (YPMIN*YUMAX-YPMAX*YUMIN)/(YUMAX-YUMIN)
      YL      = (YPMAX       -YPMIN       )
      ZA      = (ZPMAX       -ZPMIN       )/(ZUMAX-ZUMIN)
```

```
          ZB       = (ZPMIN*ZUMAX-ZPMAX*ZUMIN)/(ZUMAX-ZUMIN)
          ZL       = (ZPMAX      -ZPMIN       )
          XO       = 8.
          YO       = 12.
       85 PLOTR(61)  = (ALPHAX*PI)/180.
          PLOTR(62)  = (ALPHAY*PI)/180.
          PLOTR(63)  = (ALPHAZ*PI)/180.
          PLOTR(64)  = XO
          PLOTR(65)  = YO
          PLOTR(66)  = XA
          PLOTR(67)  = XB
          PLOTR(68)  = YA
          PLOTR(69)  = YB
          PLOTR(70)  = ZA
          PLOTR(71)  = ZB
          GO TO (86,87) IBD
       86 CALL CODE
          WRITE (YTITLE,1101)
          GO TO 88
       87 CALL CODE
          WRITE (YTITLE,1102)
       88 GO TO (301,302) IPX
      301 CALL CODE
          WRITE (ZTITLE,1103)
          GO TO 303
      302 CALL CODE
          WRITE (ZTITLE,1104)
      303 CALL SETSM (113,1.)
          CALL AXIS (XO,YO,XL,ALPHAX,6H Gamma, 8,XUMIN,XUMAX,4HF6.2,6,INCX)
          CALL AXIS (XO,YO,YL,ALPHAY,YTITLE,-8,YUMIN,YUMAX,4HF6.2,6,INCY)
          CALL AXIS (XO,YO,ZL,ALPHAZ,ZTITLE,-8,ZUMIN,ZUMAX,4HF6.2,6,INCZ)



C         ..........................................................
C         :                                                        :
C         :  Plot measured calibration surface.                    :
C         :                                                        :
C         ..........................................................
          CALL SETSM (113,2.)
          DO 01 I=1,NMACH,1
          DO 01 J=1,NPITCH,1
          CALL THRTW (XPLOT,YPLOT,X1(I,J),X2(I,J),Y(I,J))
          IF ( J   .EQ.   1 ) CALL PLOT (XPLOT,YPLOT,2)
          IF ( J   .GT.   1 ) CALL PLOT (XPLOT,YPLOT,3)
       01 CONTINUE
          DO 02 J=1,NPITCH,1
          DO 02 I=1,NMACH,1
          CALL THRTW (XPLOT,YPLOT,X1(I,J),X2(I,J),Y(I,J))
          IF ( I   .EQ.   1 ) CALL PLOT (XPLOT,YPLOT,2)
          IF ( I   .GT.   1 ) CALL PLOT (XPLOT,YPLOT,3)
       02 CONTINUE



C         ..........................................................
C         :                                                        :
C         :  Calculate calibration surface coefficients.           :
C         :                                                        :
C         ..........................................................
       91 CALL SETSM (113,0.)
          WRITE (LI, 101) NOCR
          READ  (LI,   *) MORDER
          WRITE (LI, 149) ICLR
          IF (MORDER.GT.6) GO TO 91
       92 WRITE (LI, 102) NOCR
          READ  (LI,   *) NORDER
          WRITE (LI, 149) ICLR
          WRITE (LI, 107) NOCR
          READ  (LI,   *) IPRINT
          WRITE (LI, 149) ICLR
          IF (NORDER.GT.6) GO TO 92
          M=MORDER+1
          N=NORDER+1
          CALL MAT3 (NMACH,NPITCH,M,N,COEFF,IPRINT)
```

```


C      ................................................................
C      .                                                              .
C      .  Overwrite data array with calculated data, based on the     .
C      .  just obtained coefficients.                                 .
C      .                                                              .
C      ................................................................
   411 WRITE (LI, 105) NOCR
       READ  (LI,   *) IAR
       WRITE (LI, 149) (ICLR,I=1,2)
       IF (IAR.LT.1 .OR. IAR.GT.2) GO TO 411
       IF (IAR.EQ.1) WRITE (LO, 604)
       IF (IAR.EQ.2) WRITE (LO, 644)
       WRITE (LO, 602) (J,J=1,NPITCH,1)
       DO 20 I=1,NMACH,1
       DO 19 J=1,NPITCH,1
       SUM=0.
       DO 96 I1=1,M,1
       IEXP1=I1-1
       IF (IEXP1.EQ.0) GO TO 401
       X1EXP=X1(I,J)**IEXP1
       GO TO 402
   401 X1EXP=1.
   402 CONTINUE
       DO 96 J1=1,N,1
       IEXP2=J1-1
       IF (IEXP2.EQ.0) GO TO 403
       X2EXP=X2(I,J)**IEXP2
       GO TO 404
   403 X2EXP=1.
   404 CONTINUE
    96 SUM=SUM+COEFF(I1,J1)*X1EXP*X2EXP
       GO TO (412,413) IAR
   412 R(J)=SUM-Y(I,J)
       GO TO 19
   413 R(J)=((SUM-Y(I,J))/Y(I,J))*100.
    19 Y(I,J)=SUM
    20 WRITE (6,603) I,(R(J),J=1,NPITCH,1)
       WRITE (6,602) (J,J=1,NPITCH,1)



C      ................................................................
C      .                                                              .
C      .  Plot calculated calibration surface.                        .
C      .                                                              .
C      ................................................................
       CALL SETSM (113,3.)
       DO 81 I=1,NMACH,1
       DO 81 J=1,NPITCH,1
       CALL THRTW (XPLOT,YPLOT,X1(I,J),X2(I,J),Y(I,J))
       IF (J.EQ.1) CALL PLOT (XPLOT,YPLOT,2)
       IF (J.GT.1) CALL PLOT (XPLOT,YPLOT,3)
    81 CONTINUE
       DO 82 J=1,NPITCH,1
       DO 82 I=1,NMACH,1
       CALL THRTW (XPLOT,YPLOT,X1(I,J),X2(I,J),Y(I,J))
       IF (I.EQ.1) CALL PLOT (XPLOT,YPLOT,2)
       IF (I.GT.1) CALL PLOT (XPLOT,YPLOT,3)
    82 CONTINUE



C      ................................................................
C      .                                                              .
C      .  Redefine Y.                                                 .
C      .                                                              .
C      ................................................................
       DO 802 I=1,NMACH,1
       DO 802 J=1,NPITCH
   802 Y(I,J)=YT(I,J)



C      ................................................................
```

```
C       :                                                        :
C       :  Next step ?                                           :
C       :                                                        :
C       :........................................................:
        WRITE (LI,103) NOCR
        READ  (LI,  *) IDUM
        WRITE (LI, 149) (ICLR,I=1,2,1)
        IF (IDUM.EQ.1) GO TO 91
        IF (IDUM.EQ.2) GO TO 89
        CALL STOPG
        STOP 077
     89 WRITE (LI,104)
        READ  (LI,  *) ALPHAX,ALPHAY,XO,YO
        WRITE (LI, 149) ICLR
        GO TO 85
        END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 04465 COMMON = 00000

3.6.2. Load map

```
LAB      10042 20622  Plot calibration points and calculate coefficients.

to load these program modules, enter (from LOADR):  MS,%TPLBL

DATA3    20623 23622   3D-Approximation  / DATA3 /
MAT3     23623 27455   3D-Approximation arrange system matrix and vector.
MAT31    27456 27651   3D-Approximation       arrange submatrix 1. class.
S3       27652 30070   3D-Approximation
AB3      30071 41534   3D-Approximation  / AB3   /     compute summations.

AFLD     41535 42534   GSP  / AFLD  / A(256)
AXIS     42535 45054   GSP                               Draw and label axes.
INITG    45055 46067   GSP    read control array A;  initialize plotter.
PLOT     46070 46620   GSP                      move pen to a defined point.
SETSM    46621 47561   GSP                            change plotter modes.
STOPG    47562 47624   GSP                               terminate grafics.
THRTW    47625 50111   GSP converts 3D user coordinates to plotter units.
FCTR     50112 50115   GSP  / FCTR  / FX,FY
FACTR    50116 50135   GSP                            vary size of the plot.

LOGLU    50136 50213  92067-16268 REV.1903 790228
READF    50214 51155  92067-16125 REV.1940 790719
OPEN     51156 51453  92067-16125 REV.1903 790215
CLOSE    51454 51663  92067-16125 REV.1903 781229
CLRIO    51664 51672  750701  24998-16001
OVRD.    51673 51673  92067-16125 REV.1903 780526
$SMVE    51674 51762  92067-16268 REV.1903 790202
LURQ     51763 52345  92067-16268 REV.1903 790223
.DADS    52346 52455   780818  24998-16001
.DMP     52456 52623   780818  24998-16001
.DDI     52624 53124   781021  24998-16001
SESSN    53125 53142  92067-16125 REV.1903 780413
R/WS     53143 53301  92067-16125 REV.1903 781214
P.PAS    53302 53330  92067-16125 REV.1903 740801
.DNG     53331 53340   780818  24998-16001
PAUSE    53341 53441  771122  24998-16001
$ALRN    53442 53557  92067-16268 REV.1903 770715
FMTIO    53560 55056   24998-16002 REV.1926 790417
ERR0     55057 55146   771122  24998-16001
TAN      55147 55253   780424  24998-16001
ABS      55254 55262    750701  24998-16001
.SNCS    55263 55424   780424  24998-16001
.DDE     55425 55436   780818  24998-16001
.DIN     55437 55444   780818  24998-16001
.RTOI    55445 55540   780921  24998-16001
.FPWR    55541 55602   781106  24998-16001
.SBT     55603 55643   770518  24998-16001
..FCM    55644 55660  750701  24998-16001
PAU.E    55661 55661  750701  24998-16001
ER0.E    55662 55662  750701  24998-16001
.CMRS    55663 55746   780424  24998-16001
$OPEN    55747 56123  92067-16125 REV.1903 790103
RWSUB    56124 56471  92067-16125 REV.1903 781003
RWNDS    56472 56614  92067-16125 REV.1903 780801
FRMTR    56615 62252   24998-16002 REV.1926 790503
FMT.E    62253 62253   24998-16002 REV.1901 781107
IB4A2    62254 63554  59310-1X013 REV 1940 790802 1153
REIO     63555 63701  92067-16268 REV.1903 790316
RMPAR    63702 63744  781106  24998-16001
PNAME    63745 64012   771121  24998-16001
LUTRU    64013 64121  92067-16268 REV.1903 790223
IPUT     64122 64142  92067-16125 REV.1903 740801
$SETP    64143 64167   781106  24998-16001
.CFER    64170 64245  750701  24998-16001
.LBT     64246 64276   770518  24998-16001

24 PAGES RELOCATED    24 PAGES REQ'D    NO PAGES EMA      NO PAGES MSEG
LINKS:BP    PROGRAM:LB   LOAD:TE   COMMON:NC
 /LOADR:LAB     READY AT  3:10 PM  FRI., 26  SEPT, 1980

 /LOADR:$END
```

3.6.3. Results

i) printed output

These data from file CALIAA.

1. Mach number

Gamma	Beta	Phi
-.810	-.005	-35.000
-.519	-.005	-25.000
-.280	-.005	-15.000
-.182	-.005	-10.000
-.103	-.006	-5.000
-.059	-.006	-2.500
-.018	-.006	0.000
.019	-.006	2.500
.054	-.006	5.000
.133	-.006	10.000
.197	-.006	15.000
.360	-.006	25.000
.605	-.006	35.000

2. Mach number

Gamma	Beta	Phi
-.781	-.016	-35.000
-.506	-.016	-25.000
-.280	-.016	-15.000
-.178	-.017	-10.000
-.092	-.018	-5.000
-.048	-.018	-2.500
-.004	-.018	0.000
.032	-.018	2.500
.061	-.018	5.000
.130	-.019	10.000
.190	-.019	15.000
.346	-.020	25.000
.583	-.018	35.000

3. Mach number

Gamma	Beta	Phi
-.787	-.024	-35.000
-.510	-.024	-25.000
-.276	-.024	-15.000
-.172	-.026	-10.000
-.085	-.027	-5.000
-.050	-.027	-2.500
-.003	-.027	0.000
.037	-.027	2.500
.066	-.027	5.000
.131	-.027	10.000
.191	-.029	15.000
.345	-.029	25.000
.592	-.027	35.000

4. Mach number

Gamma	Beta	Phi
-.790	-.035	-35.000
-.519	-.035	-25.000
-.276	-.035	-15.000
-.174	-.037	-10.000
-.055	-.039	-5.000
-.043	-.039	-2.500
-.003	-.038	0.000
.037	-.038	2.500
.064	-.039	5.000
.133	-.039	10.000
.196	-.041	15.000
.349	-.042	25.000
.596	-.039	35.000

5. Mach number

Gamma	Beta	Phi
-.795	-.045	-35.000
-.521	-.045	-25.000
-.282	-.045	-15.000
-.175	-.048	-10.000
-.082	-.050	-5.000
-.040	-.050	-2.500
-.010	-.050	0.000
.034	-.049	2.500
.067	-.049	5.000
.134	-.050	10.000
.198	-.053	15.000
.350	-.055	25.000
.593	-.050	35.000

6. Mach number

Gamma	Beta	Phi
-.803	-.065	-35.000
-.528	-.065	-25.000
-.284	-.065	-15.000
-.173	-.068	-10.000
-.085	-.071	-5.000
-.032	-.072	-2.500
-.004	-.071	0.000
.024	-.071	2.500
.058	-.071	5.000
.131	-.071	10.000
.199	-.075	15.000
.349	-.078	25.000
.603	-.073	35.000
X1	X2	Y

System matrix A and vector B before Gauss Jordan Elimation

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	.780E+02	-.270E+01	.129E+00	-.722E-02	.441E-03	-.323E+01	.675E-01	-.133E-02	-.272E-04	.777E-05	.933E+01	-.312E+00	.145E-01
2	-.270E+01	.129E+00	-.722E-02	.441E-03	-.205E-04	.675E-01	-.133E-02	-.272E-04	.777E-05	-.890E-06	-.312E+00	.145E-01	-.790E-03
3	.129E+00	-.722E-02	.441E-03	-.205E-04	.191E-05	-.133E-02	-.272E-04	.777E-05	-.890E-06	.852E-07	.145E-01	-.790E-03	.471E-04
4	-.722E-02	.441E-03	-.205E-04	.191E-05	-.131E-06	-.272E-04	.777E-05	-.890E-06	.852E-07	-.756E-08	-.790E-03	.471E-04	-.298E-05
5	.441E-03	-.205E-04	.191E-05	-.131E-06	.917E-08	.777E-05	-.890E-06	.852E-07	-.756E-08	.645E-09	.471E-04	-.298E-05	.195E-06
6	-.323E+01	.675E-01	-.133E-02	-.272E-04	.777E-05	.933E+01	-.312E+00	.145E-01	-.790E-03	.471E-04	-.242E+01	.709E-01	-.283E-02
7	.675E-01	-.133E-02	-.272E-04	.777E-05	-.890E-06	-.312E+00	.145E-01	-.790E-03	.471E-04	-.298E-05	.709E-01	-.283E-02	.128E-03
8	-.133E-02	-.272E-04	.777E-05	-.890E-06	.852E-07	.145E-01	-.790E-03	.471E-04	-.298E-05	.195E-06	-.283E-02	.128E-03	-.600E-05
9	-.272E-04	.777E-05	-.890E-06	.852E-07	-.756E-08	-.790E-03	.471E-04	-.298E-05	.195E-06	-.131E-07	.128E-03	-.600E-05	.274E-06
10	.777E-05	-.890E-06	.852E-07	-.756E-08	.645E-09	.471E-04	-.298E-05	.195E-06	-.131E-07	.901E-09	-.600E-05	.274E-06	-.110E-07
11	.933E+01	-.312E+00	.145E-01	-.790E-03	.471E-04	-.242E+01	.709E-01	-.283E-02	.128E-03	-.600E-05	.372E+01	-.122E+00	.561E-02
12	-.312E+00	.145E-01	-.790E-03	.471E-04	-.298E-05	.709E-01	-.283E-02	.128E-03	-.600E-05	.274E-06	-.122E+00	.561E-02	-.300E-03
13	.145E-01	-.790E-03	.471E-04	-.298E-05	.195E-06	-.283E-02	.128E-03	-.600E-05	.274E-06	-.110E-07	.561E-02	-.300E-03	.176E-04
14	-.790E-03	.471E-04	-.298E-05	.195E-06	-.131E-07	.128E-03	-.600E-05	.274E-06	-.110E-07	.274E-09	-.300E-03	.176E-04	-.108E-05
15	.471E-04	-.298E-05	.195E-06	-.131E-07	.901E-09	-.600E-05	.274E-06	-.110E-07	.274E-09	.127E-10	.176E-04	-.108E-05	.694E-07
16	-.242E+01	.709E-01	-.283E-02	.128E-03	-.600E-05	.372E+01	-.122E+00	.561E-02	-.300E-03	.176E-04	-.165E+01	.509E-01	-.219E-02
17	.709E-01	-.283E-02	.128E-03	-.600E-05	.274E-06	-.122E+00	.561E-02	-.300E-03	.176E-04	-.108E-05	.509E-01	-.219E-02	.108E-03
18	-.283E-02	.128E-03	-.600E-05	.274E-06	-.110E-07	.561E-02	-.300E-03	.176E-04	-.108E-05	.694E-07	-.219E-02	.108E-03	-.578E-05
19	.128E-03	-.600E-05	.274E-06	-.110E-07	.274E-09	-.300E-03	.176E-04	-.108E-05	.694E-07	-.455E-08	.108E-03	-.578E-05	.320E-06
20	-.600E-05	.274E-06	-.110E-07	.274E-09	.127E-10	.176E-04	-.108E-05	.694E-07	-.455E-08	.303E-09	-.578E-05	.320E-06	-.181E-07
	1	2	3	4	5	6	7	8	9	10	11	12	13

	14	15	16	17	18	19	20	
1	-.790E-03	.471E-04	-.242E+01	.709E-01	-.283E-02	.120E-03	-.600E-05	.000E+00
2	.471E-04	-.290E-05	.709E-01	-.283E-02	.120E-03	-.600E-05	.274E-06	-.234E+01
3	-.290E-05	.195E-06	-.283E-02	.120E-03	-.600E-05	.274E-06	-.110E-07	.217E+00
4	.195E-06	-.131E-07	.120E-03	-.600E-05	.274E-06	-.110E-07	.274E-09	-.179E-01
5	-.131E-07	.901E-09	-.600E-05	.274E-06	-.110E-07	.274E-09	.127E-10	.144E-02
6	.120E-03	-.600E-05	.372E+01	-.122E+00	.561E-02	-.300E-03	.176E-04	.409E+03
7	-.600E-05	.274E-06	-.122E+00	.561E-02	-.300E-03	.176E-04	-.108E-05	-.165E+02
8	.274E-06	-.110E-07	.561E-02	-.300E-03	.176E-04	-.108E-05	.694E-07	.774E+00
9	-.110E-07	.274E-09	-.300E-03	.176E-04	-.108E-05	.694E-07	-.455E-08	-.426E-01
10	.274E-09	.127E-10	.176E-04	-.108E-05	.694E-07	-.455E-08	.303E-09	.257E-02
11	-.300E-03	.176E-04	-.165E+01	.509E-01	-.219E-02	.108E-03	-.578E-05	-.844E+02
12	.176E-04	-.108E-05	.509E-01	-.219E-02	.108E-03	-.578E-05	.320E-06	.228E+01
13	-.108E-05	.694E-07	-.219E-02	.108E-03	-.578E-05	.320E-06	-.181E-07	-.810E-01
14	.694E-07	-.455E-08	.108E-03	-.578E-05	.320E-06	-.181E-07	.102E-08	.380E-02
15	-.455E-08	.303E-09	-.578E-05	.320E-06	-.181E-07	.102E-08	-.565E-10	-.927E-04
16	.108E-03	-.578E-05	.171E+01	-.621E-01	.283E-02	-.151E-03	.870E-05	.180E+03
17	-.578E-05	.320E-06	-.621E-01	.283E-02	-.151E-03	.870E-05	-.530E-06	-.595E+01
18	.320E-06	-.181E-07	.283E-02	-.151E-03	.870E-05	-.530E-06	.334E-07	.274E+00
19	-.181E-07	.102E-08	-.151E-03	.870E-05	-.530E-06	.334E-07	-.215E-08	-.140E-01
20	.102E-08	-.565E-10	.870E-05	-.530E-06	.334E-07	-.215E-08	.141E-09	.870E-03
	14	15	16	17	18	19	20	

Equation system after Gauss Jordan Elimination

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
2	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
3	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
4	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
5	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
6	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
7	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
8	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
9	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00
10	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00
11	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00
12	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00

	14	15	16	17	18	19	20	
1	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.130E+02
2	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	-.981E+02
3	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	-.838E+04
4	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	-.210E+04
5	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
6	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	-.145E+07
7	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.482E+03
8	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.430E+05
9	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.113E+07
10	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.894E+07
11	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.181E+01
12	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	-.564E+03

13	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01
	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	-.248E+05					
14	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	-.538E+06					
15	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	-.370E+07					
16	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	-.309E+02					
17	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	-.220E+04					
18	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	-.157E+06					
19	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	-.373E+07					
20	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	-.279E+08					
	1	2	3	4	5	6	7	8	9	10	11	12	13
	14	15	16	17	18	19	20						

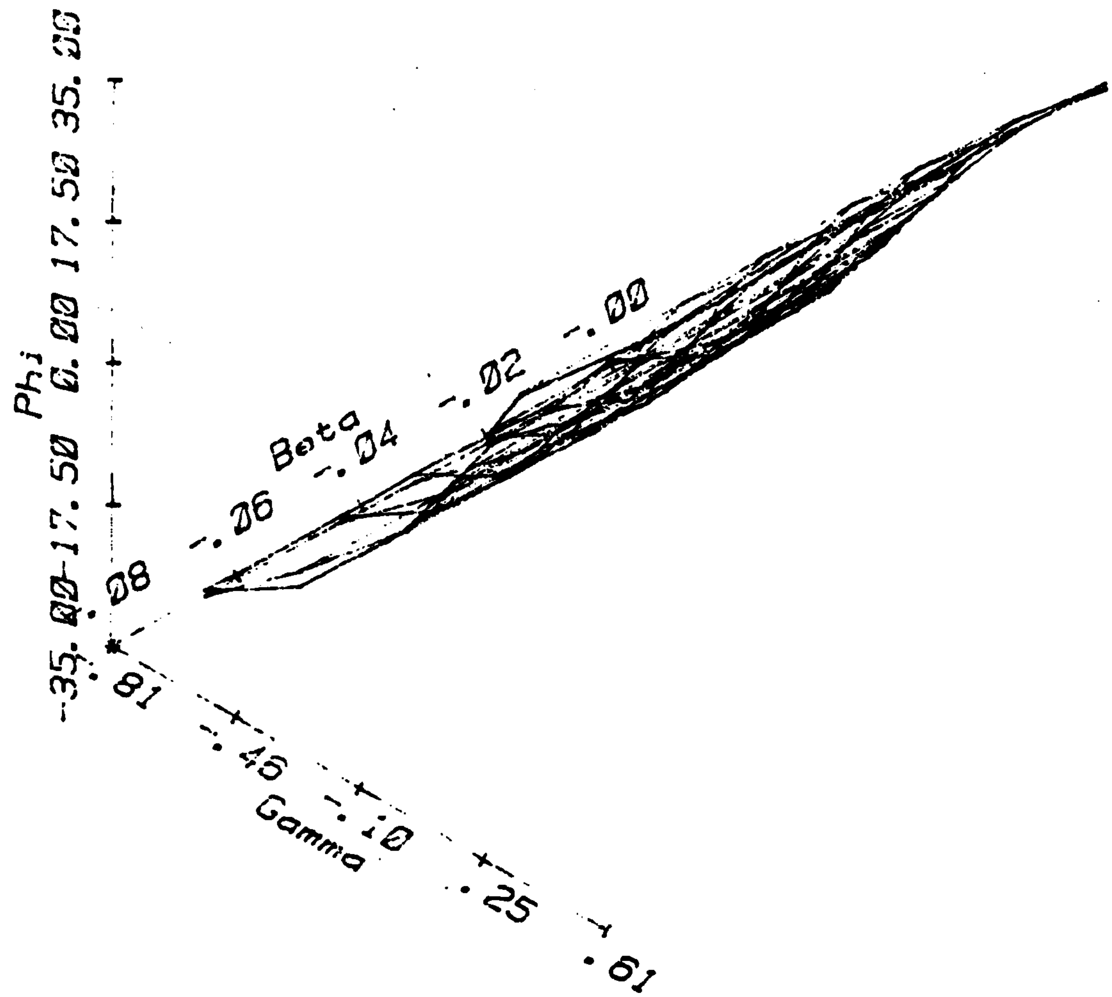
Coefficients COEF(I,J)

	J = 1	2	3	4	5
I = 1	.139E+01	-.981E+02	-.838E+04	-.200E+06	-.145E+07
I = 2	.630E+02	.482E+03	.430E+05	.113E+07	.894E+07
I = 3	.101E+01	-.564E+03	-.268E+05	-.538E+06	-.370E+07
I = 4	-.309E+02	-.220E+04	-.157E+06	-.373E+07	-.279E+08
	J = 1	2	3	4	5

Absolute error at each point (error = (calculated value - measured value))

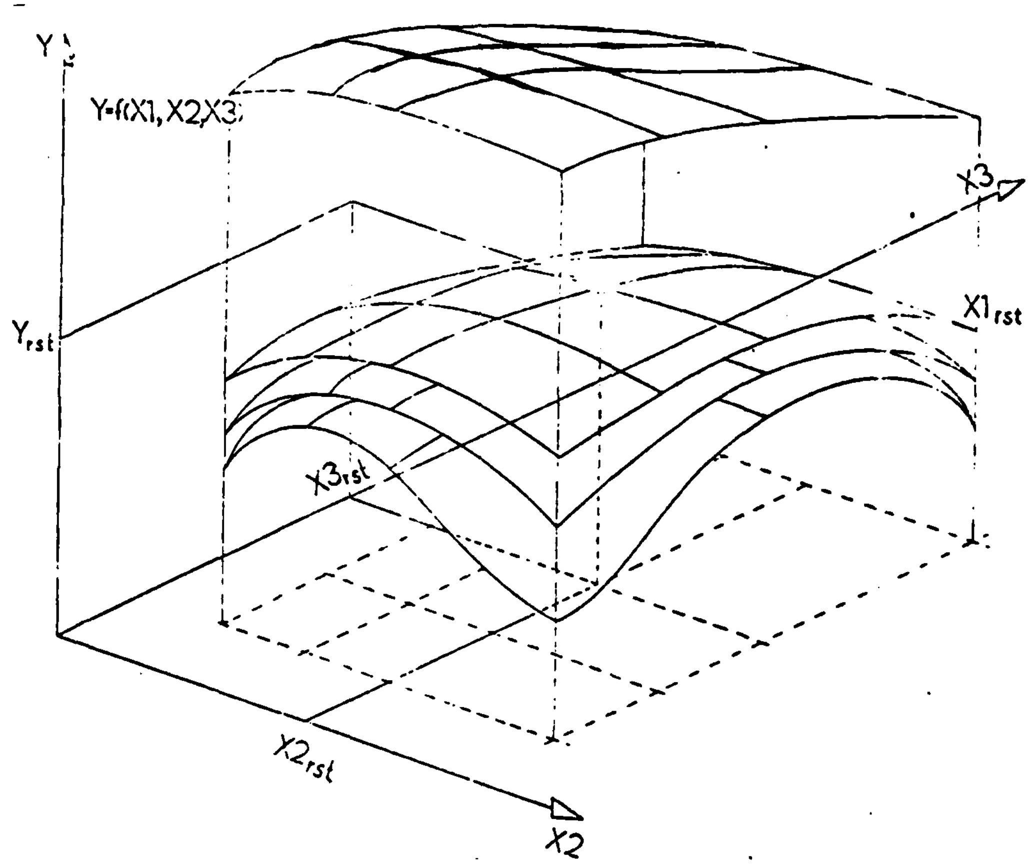
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	.531	-1.563	-.003	.440	.395	.560	.503	.370	.001	-.024	-1.066	-1.439	.704
2	.221	-1.388	-.244	.492	.656	.942	1.232	.900	.343	-.294	-1.443	-1.900	.850
3	.961 -.776	-1.755	-.657	.320	.663	.397	.985	.970	.367	-.431	-1.540	-2.057	.975
4	1.035	-2.435	-.032	.135	2.509	.744	.795	.026	.124	-.473	-1.497	-2.182	.947
5	.781	-1.846	-.552	.783	1.335	1.346	.604	.855	.410	-.400	-1.420	-2.154	1.328
6	.761	-1.008	-.302	.350	.098	1.009	.595	-.016	-.149	-.202	-.361	-.320	.357
	1	2	3	4	5	6	7	8	9	10	11	12	13

ii) graphic output



4. FOUR DIMENSIONAL APPROXIMATION

4.1. Problem:



A data set of NPNTS1 * NPNTS2 * NPNTS3 data points is given, where Y depends on the parameters X1, X2 and X3. The data pattern is to be approximated by a function Y = f(X1,X2,X3), so that the error between data points and analytically determined points is lower.

4.2. Approach

Again, as in the two and three dimensional case, a polynominal method is used. For

$$Y = f(X1, X2, X3)$$

we assume

$$\begin{aligned} Y = \{ & [C_{111} + C_{112} X3 + \ldots + C_{11N} X3^{N-1}] + \\ & + [C_{121} + C_{122} X3 + \ldots + C_{12N} X3^{N-1}] \cdot X2 + \\ & \cdots \\ & + [C_{1M1} + C_{1M2} X3 + \ldots + C_{1MN} X3^{N-1}] \cdot X2^{M-1} \} + \\ & + \{ [C_{211} + C_{212} X3 + \ldots + C_{21N} X3^{N-1}] + \\ & [C_{221} + C_{222} X3 + \ldots + C_{22N} X3^{N-1}] \cdot X2 + \\ & \cdots \\ & + [C_{2M1} + C_{2M2} X3 + \ldots + C_{2MN} X3^{N-1}] \cdot X2^{M-1} \} X1 + \\ & \cdots \\ & + \{ [C_{L11} + C_{L12} X3 + \ldots + C_{L1N} X3^{N-1}] + \\ & + [C_{L21} + C_{L22} X3 + \ldots + C_{L2N} X3^{N-1}] \cdot X2 + \\ & \cdots \\ & + [C_{LM1} + C_{LM2} X3 + \ldots + C_{LMN} X3^{N-1}] \cdot X2^{M-1} \} X1^{L-1} \end{aligned}$$

or

$$Y = \sum^{L} \{ \sum^{M} \{ \sum^{N} C_{ijk} X3^{k-1} \} X2^{j-1} \} X1^{i-1} \qquad (4.1.)$$

This data arrangement which discriminates between data points taken at constant X1, X2 and X3, respectively is needed in the application of this method to the description of pneumatic-velocity probe characteristics. The various surfaces in the above sketch for example can be seen as data taken at different Mach number (X1). At each constant Mach number (X1 = constant) yaw angle (X2) and pitch angle (X3) are varied, and the probe is measured to be Y. Even though probes are usually balanced to the average flow yaw angle in some cases this is not possible and the four dimensional approximation is then needed. A similar method for describing probe calibration data is used at the institute for Jet Engines and Turbomachines at the Technical University Aachen (Ref. 2).

The least squares criterion is then

$$R = \sum_{r=1}^{NPNTS1} \{ \sum_{s=1}^{NPNTS2} \{ \sum_{t=1}^{NPNTS3} [f(X1_{rst}, X2_{rst}, X3_{rst}) - Y_{rst}]^2 \}\} \tag{4.2.}$$

where the indices rst denote data points. Using equation (4.1.), R becomes

$$\begin{aligned} R = \sum_{r=1}^{NPNTS1} \{ \sum_{s=1}^{NPNTS2} \{ \sum_{t=1}^{NPNTS3} [\, & c_{111} + c_{112} X3_{rst} + \ldots + c_{11N} X3_{rst}^{N-1} + \\ & + (c_{121} + c_{122} X3_{rst} + \ldots + c_{12N} X3_{rst}^{N-1}) \cdot X2_{rst} + \\ & \ldots \\ & + (c_{1M1} + c_{1M2} X3_{rst} + \ldots + c_{1MN} X3_{rst}^{N-1}) \cdot X2_{rst}^{M-1} + \\ & + \langle c_{211} + c_{212} X3_{rst} + \ldots + c_{21N} X3_{rst}^{N-1} + \\ & + (c_{221} + c_{222} X3_{rst} + \ldots + c_{22N} X3^{N-1}) \cdot X2_{rst} + \\ & \ldots \\ & + (c_{2M1} + c_{2M2} X3_{rst} + \ldots + c_{2MN} X3^{N-1}) \cdot X2_{rst}^{M-1} \rangle \cdot X1_{rst} \\ & \ldots \end{aligned}$$

$$+\langle C_{L11} + C_{L12} X3_{rst} + \ldots + C_{L1N} X3^{N-1} +$$

$$(C_{L21} + C_{L22} X3_{rst} + \ldots + C_{L2N} X3^{N-1}) \cdot X2_{rst} +$$

$$\ldots$$

$$(C_{LM1} + C_{LM2} X3_{rst} + \ldots + C_{LMN} X3^{N-1}) \cdot X2_{rst}^{M-1} \rangle \cdot X1_{rst}^{L-1}] \} \}$$

The term in the [ ] - bracket shall be called B

$$R = \sum_{r=1}^{NPNTS1} \{ \sum_{s=1}^{NPNTS2} \{ \sum_{t=1}^{NPNTS3} B^2 \} \}$$

4.3. Solution:

To minimize the error, R is partially differentiated with respect to the coefficients C_{ijk} and the partial derivatives are set to zero, thus:

$$\frac{\partial R}{\partial C_{ijk}} \stackrel{!}{=} 0 \qquad i = 1 \ldots L; \; j = 1 \ldots M; \; k = 1 \ldots N$$

$$\frac{\partial R}{\partial C_{ijk}} = \sum_{r=1}^{NPNTS1} \{ \sum_{s=1}^{NPNTS2} \{ \sum_{t=1}^{NPNTS3} 2B \cdot \frac{\partial B}{\partial C_{ijk}} \} \} = 0$$

Assuming, that the summations extend over all data points, $\Sigma\Sigma\Sigma$ should be understood to be $\sum_{r=1}^{NPNTS1} \sum_{s=1}^{NPNTS2} \sum_{t=1}^{NPNTS3}$. Now even a very large sheet of paper can't take the system matrix, unless we skip to submatrix notation.

$$\begin{vmatrix} \mathbb{A}_{11}^{*} & \mathbb{A}_{12}^{*} & \ldots & \mathbb{A}_{1L}^{*} \\ \mathbb{A}_{21}^{*} & \mathbb{A}_{22}^{*} & \ldots & \mathbb{A}_{2L}^{*} \\ \ldots & & & \\ \mathbb{A}_{L1}^{*} & \mathbb{A}_{L2}^{*} & \ldots & \mathbb{A}_{LL}^{*} \end{vmatrix} \begin{vmatrix} \mathbb{C}_{1}^{*} \\ \mathbb{C}_{2}^{*} \\ \ldots \\ \mathbb{C}_{L}^{*} \end{vmatrix} = \begin{vmatrix} \mathbb{B}_{1}^{*} \\ \mathbb{B}_{2}^{*} \\ \ldots \\ \mathbb{B}_{L}^{*} \end{vmatrix} \qquad (4.3.)$$

The asterisk indicates a submatrix 1. class. On the following pages the submatrices and subvectors 1. class are listed.

$$
A_N^{s} = \left[\begin{array}{cccc|cccc}
NPNTS1+NPNTS2+NPNTS3 & \sum\sum\sum x3_{rst} & \cdots & \sum\sum\sum x3_{rst}^{N-1} & \sum\sum\sum x2_{rst} & \sum\sum\sum x2_{rst}\cdot x3_{rst} & \cdots & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{N-1} \\
\sum\sum\sum x3_{rst} & \sum\sum\sum x3_{rst}^{2} & \cdots & \sum\sum\sum x3_{rst}^{N} & \sum\sum\sum x2_{rst}\cdot x3_{rst} & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{2} & \cdots & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{N} \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x3_{rst}^{N-1} & \sum\sum\sum x3_{rst}^{N} & \cdots & \sum\sum\sum x3_{rst}^{2N-2} & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{N-1} & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{N} & \cdots & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{2N-2} \\
\hline
\sum\sum\sum x2_{rst} & \sum\sum\sum x2_{rst}\cdot x3_{rst} & \cdots & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{N-1} & \sum\sum\sum x2_{rst}^{2} & \sum\sum\sum x2_{rst}^{2}\cdot x3_{rst} & \cdots & \sum\sum\sum x2_{rst}^{2}\cdot x3_{rst}^{N-1} \\
\sum\sum\sum x2_{rst}\cdot x3_{rst} & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{2} & \cdots & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{N} & \sum\sum\sum x2_{rst}^{2}\cdot x3_{rst} & \sum\sum\sum x2_{rst}^{2}\cdot x3_{rst}^{2} & \cdots & \sum\sum\sum x2_{rst}^{2}\cdot x3_{rst}^{N} \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x2_{rst}\cdot x3_{rst}^{N-1} & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{N} & \cdots & \sum\sum\sum x2_{rst}\cdot x3_{rst}^{2N-2} & \sum\sum\sum x2_{rst}^{2}\cdot & \sum\sum\sum x2_{rst}^{2}\cdot & \cdots & \sum\sum\sum x2_{rst}^{2}\cdot x3_{rst}^{2N-2} \\
\hline
 & & & & & & & \\
\cdots & & & & & & & \\
 & & & & & & & \\
\hline
\sum\sum\sum x2_{rst}^{N-1} & \sum\sum\sum x2_{rst}^{N-1}\cdot x3_{rst} & \cdots & \sum\sum\sum x2_{rst}^{N-1}\cdot x3_{rst}^{N-1} & \sum\sum\sum x2_{rst}^{N} & \sum\sum\sum x2_{rst}^{N}\cdot x3_{rst} & & \sum\sum\sum x2_{rst}^{N}\cdot x3_{rst}^{N-1} \\
\sum\sum\sum x2_{rst}^{N-1}\cdot x3_{rst} & \sum\sum\sum x2_{rst}^{N-1}\cdot x3_{rst}^{2} & \cdots & \sum\sum\sum x2_{rst}^{N-1}\cdot x3_{rst}^{N} & \sum\sum\sum x2_{rst}^{N}\cdot x3_{rst} & \sum\sum\sum x2_{rst}^{N}\cdot x3_{rst}^{2} & & \sum\sum\sum x2_{rst}^{N}\cdot x3_{rst}^{N} \\
\cdots & & & & & & & \\
\sum\sum\sum x2_{rst}^{N-1}\cdot x3_{rst}^{N-1} & \sum\sum\sum x2_{rst}^{N-1}\cdot x3_{rst}^{N} & \cdots & \sum\sum\sum x2_{rst}^{N-1}\cdot x3_{rst}^{2N-2} & \sum\sum\sum x2_{rst}^{N}\cdot x3_{rst}^{N-1} & \sum\sum\sum x2_{rst}^{N}\cdot x3_{rst}^{N} & & \sum\sum\sum x2_{rst}^{N}\cdot x3_{rst}^{2N-2}
\end{array}\right.
$$

1

$$
\begin{array}{lll}
\Sigma\Sigma x2_{rst}\cdot x3_{rst} & \cdots & \Sigma\Sigma\Sigma x2_{rst}\cdot x3_{rst}^{M-1} \\
\Sigma\Sigma x2_{rst}\cdot x3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma x2_{rst}\cdot x3_{rst}^{M} \\
 & & \\
\Sigma\Sigma x2_{rst}\cdot x3_{rst}^{M} & \cdots & \Sigma\Sigma\Sigma x2_{rst}\cdot x3_{rst}^{2M-2} \\
\hline
\Sigma\Sigma x2_{rst}^{2}\cdot x3_{rst} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{2}\cdot x3_{rst}^{M-1} \\
\Sigma\Sigma x2_{rst}^{2}\cdot x3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{2}\cdot x3_{rst}^{M} \\
 & & \\
\Sigma\Sigma x2_{rst}^{2}\cdot & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{2}\cdot x3_{rst}^{2M-2} \\
\hline
 & & \\
\hline
\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst} & & \Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{M-1} \\
\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{2} & & \Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{M} \\
 & & \\
\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{M} & & \Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{2M-2}
\end{array}
\quad \cdots \quad
\begin{array}{llll}
\Sigma\Sigma\Sigma x2_{rst}^{M-1} & \Sigma\Sigma\Sigma x2_{rst}^{M-1}\cdot x3_{rst} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{M-1}\cdot x3_{rst}^{M-1} \\
\Sigma\Sigma\Sigma x2_{rst}^{M-1}\cdot x3_{rst} & \Sigma\Sigma\Sigma x2_{rst}^{M-1}\cdot x3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{M-1}\cdot x3_{rst}^{M} \\
\cdots & & & \\
\Sigma\Sigma\Sigma x2_{rst}^{M-1}\cdot x3_{rst}^{M-1} & \Sigma\Sigma\Sigma x2_{rst}^{M-1}\cdot x3_{rst}^{M} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{M-1}\cdot x3_{rst}^{2M-2} \\
\hline
\Sigma\Sigma\Sigma x2_{rst}^{M} & \Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{M-1} \\
\Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst} & \Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{M} \\
\cdots & & & \\
\Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{M-1} & \Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{M} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{M}\cdot x3_{rst}^{2M-2} \\
\hline
\cdots & & & \\
\hline
\Sigma\Sigma\Sigma x2_{rst}^{2M-2} & \Sigma\Sigma\Sigma x2_{rst}^{2M-2}\cdot x3_{rst} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{2M-2}\cdot x3_{rst}^{M-1} \\
\Sigma\Sigma\Sigma x2_{rst}^{2M-2}\cdot x3_{rst} & \Sigma\Sigma\Sigma x2_{rst}^{2M-2}\cdot x3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{2M-2}\cdot x3_{rst}^{M} \\
\cdots & & & \\
\Sigma\Sigma\Sigma x2_{rst}^{2M-2}\cdot x3_{rst}^{M-1} & \Sigma\Sigma\Sigma x2_{rst}^{2M-2}\cdot x3_{rst}^{M} & \cdots & \Sigma\Sigma\Sigma x2_{rst}^{2M-2}\cdot x3_{rst}^{2M-2}
\end{array}
$$

56

$$
A^{k}_{rs} = \left[\begin{array}{cccc|cccc}
\sum\sum\sum X1_{rst} & \sum\sum\sum X1_{rst}\cdot X3_{rst} & \cdots & \sum\sum\sum X1_{rst}\cdot X3_{rst}^{M-1} & \sum\sum\sum X1_{rst}\cdot X2_{rst} & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{M-1} \\
\sum\sum\sum X1_{rst}\cdot X3_{rst} & \sum\sum\sum X1_{rst}\cdot X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}\cdot X3_{rst}^{M} & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst} & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{M} \\
\cdots & & & \cdots & & & & \\
\sum\sum\sum X1_{rst}\cdot X3_{rst}^{M-1} & \sum\sum\sum X1_{rst}\cdot X3_{rst}^{M} & & \sum\sum\sum X1_{rst}\cdot X3_{rst}^{2M-2} & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{M-1} & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{M} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{2M-2} \\
\hline
\sum\sum\sum X1_{rst}\cdot X2_{rst} & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst} & & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{M-1} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{2} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{2}\cdot X3_{rst} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{2}\cdot X3_{rst}^{M-1} \\
\sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst} & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{2} & & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{M} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{2}\cdot X3_{rst} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{2}\cdot X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{2}\cdot X3_{rst}^{M} \\
\cdots & & & \vdots & \cdots & & & \\
\sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{M-1} & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{M} & & \sum\sum\sum X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{2M-2} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{2}\cdot X3_{rst}^{M-1} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{2}\cdot X3_{rst}^{M} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{2}\cdot X3_{rst}^{2M-2} \\
\hline
 & & & & & & & \\
\cdots & & & & & & & \\
 & & & & & & & \\
\hline
\sum\sum\sum X1_{rst}\cdot X2_{rst}^{M-1} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M-1}\cdot X3_{rst} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{M-1} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M}\cdot X3_{rst} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M}\cdot X3_{rst}^{M-1} \\
\sum\sum\sum X1_{rst}\cdot X2_{rst}^{M-1}\cdot X3_{rst} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{M} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M}\cdot X3_{rst} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M}\cdot X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M}\cdot X3_{rst}^{M} \\
\cdots & & & \cdots & \cdots & & & \\
\sum\sum\sum X1_{rst}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{M-1} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{M} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{2M-2} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M}\cdot X3_{rst}^{M-1} & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M}\cdot X3_{rst}^{M} & \cdots & \sum\sum\sum X1_{rst}\cdot X2_{rst}^{M}\cdot X3_{rst}^{2M-2}
\end{array}\right]
$$

$\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}\cdot x3_{rst}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}\cdot x3_{rst}^{M-1}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M-1}$	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M-1}\cdot x3_{rst}$	..	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{M-1}$
$\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}\cdot x3_{rst}^{2}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}\cdot x3_{rst}^{M}$		$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M-1}\cdot x3_{rst}$	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{2}$	..	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{M}$
				...			
$\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}\cdot x3_{rst}^{M}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}\cdot x3_{rst}^{2M-1}$		$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{M-1}$	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{M}$	..	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{2M-2}$
$\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2}\cdot x3_{rst}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2}\cdot x3_{rst}^{M-1}$		$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}$	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}$	..	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{M-1}$
$\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2}\cdot x3_{rst}^{2}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2}\cdot x3_{rst}^{M}$		$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}$	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{2}$	..	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{M}$
				...			
$\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2}\cdot x3_{rst}^{M}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2}\cdot x3_{rst}^{2M-2}$		$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{M-1}$	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{M}$	..	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{2M-2}$
$\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{M-1}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2M-2}$	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2M-2}\cdot x3_{rst}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2M-2}\cdot x3_{rst}^{M-1}$
$\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{2}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{M}$		$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2M-2}\cdot x3_{rst}$	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2M-2}\cdot x3_{rst}^{2}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2M-2}\cdot x3_{rst}^{M}$
				...			
$\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{M}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{M}\cdot x3_{rst}^{2M-2}$		$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2M-2}\cdot x3_{rst}^{M-1}$	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2M-2}\cdot x3_{rst}^{M}$	...	$\Sigma\Sigma\Sigma\, x1_{rst}\cdot x2_{rst}^{2M-2}\cdot x3_{rst}^{2M-2}$

58

2

$$
A_{1L}^{4} = \left[\begin{array}{cccc|cccc}
\sum\sum\sum x1_{rst}^{L-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x3_{rst}^{N-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{N-1} \\
\sum\sum\sum x1_{rst}^{L-1}\cdot x3_{rst} & \sum\sum\sum x1_{rst}^{L-1}\cdot x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x3_{rst}^{N} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{N} \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x1_{rst}^{L-1}\cdot x3_{rst}^{N-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x3_{rst}^{N} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x3_{rst}^{2N-2} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{N-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{N} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{2N-2} \\
\hline
\sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{N-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{2} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{2}\cdot x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{2}\cdot x3_{rst}^{N-1} \\
\sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{N} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{2}\cdot x3_{rst} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{2}\cdot x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{2}\cdot x3_{rst}^{N} \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{N-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{N} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}\cdot x3_{rst}^{2N-2} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{2}\cdot x3_{rst}^{N-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{2}\cdot x3_{rst}^{N} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{2}\cdot x3_{rst}^{2N-2} \\
\hline
 & & & & & & & \\
\cdots & & & & & & & \\
 & & & & & & & \\
\hline
\sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M-1}\cdot x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{N-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M}\cdot x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M}\cdot x3_{rst}^{N-1} \\
\sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M-1}\cdot x3_{rst} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{N} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M}\cdot x3_{rst} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M}\cdot x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M}\cdot x3_{rst}^{N} \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{N-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{N} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M-1}\cdot x3_{rst}^{2N-2} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M}\cdot x3_{rst}^{N-1} & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M}\cdot x3_{rst}^{N} & \cdots & \sum\sum\sum x1_{rst}^{L-1}\cdot x2_{rst}^{M}\cdot x3_{rst}^{2N-2}
\end{array}\right]
$$

|

$$
\begin{array}{ccc|c|cccc}
X1_{rst}^{L-1}\cdot X2_{rst}\cdot X3_{rst} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}\, X3_{rst}^{N-1} & & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M-1} & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M-1}\cdot X3_{rst} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{N-1} \\
X1_{rst}^{L-1}\cdot X2_{rst}\cdot X3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}\cdot X3_{rst}^{N} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M-1}\cdot X3_{rst} & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{N} \\
 & & & & \cdots & & & \\
X1_{rst}^{L-1}\cdot X2_{rst}\cdot X3_{rst}^{N} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}\cdot X3_{rst}^{2N-2} & & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{N-1} & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M-1}\cdot X2_{rst}^{N} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{2N-2} \\
\hline
X1_{rst}^{L-1}\cdot X2_{rst}^{2}\cdot X3_{rst} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{2}\, X3_{rst}^{N-1} & & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M} & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N-1} \\
X1_{rst}^{L-1}\cdot X2_{rst}^{2}\cdot X3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\, X2_{rst}^{2}\cdot X3_{rst}^{N} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst} & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N} \\
 & & & & \cdots & & & \\
X1_{rst}^{L-1}\cdot X2_{rst}^{2}\cdot X3_{rst}^{N} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\, X2_{rst}^{2}\cdot X3_{rst}^{2N-2} & & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N-1} & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst}^{2N-2} \\
\hline
 & & & & & & & \\
\hline
X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\, X2_{rst}^{M}\cdot X3_{rst}^{N-1} & & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{2M-2} & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{2M-2}\cdot X3_{rst} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{2M-2}\cdot X3_{rst}^{N-1} \\
X1_{rst}^{L-1}\, X2_{rst}^{M}\cdot X3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{2M-2}\cdot X3_{rst} & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{2M-2}\cdot X3_{rst}^{2} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{2M-2}\cdot X3_{rst}^{N} \\
 & & & & \cdots & & & \\
X1_{rst}^{L-1}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\, X2_{rst}^{M}\, X3_{rst}^{2N-2} & & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{2M-2}\cdot X3_{rst}^{N-1} & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{2M-2}\cdot X3_{rst}^{N} & \cdots & \Sigma\Sigma\Sigma X1_{rst}^{L-1}\cdot X2_{rst}^{2M-2}\cdot X3_{rst}^{2N-2}
\end{array}
$$

$$
A_{2L}^{*} = \left[\begin{array}{cccc|cccc}
\sum\sum\sum x1^L_{rst} & \sum\sum\sum x1^L_{rst}\cdot x3_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x3^{M-1}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2_{rst} \\
\sum\sum\sum x1^L_{rst}\cdot x3_{rst} & \sum\sum\sum x1^L_{rst}\cdot x3^2_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x3^M_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3^2_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2 \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x1^L_{rst}\cdot x3^{M-1}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x3^M_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x3^{2M-2}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3^{M-1}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3^M_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2 \\
\hline
\sum\sum\sum x1^L_{rst}\cdot x2_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3^{M-1}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^2_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^2_{rst}\cdot x3_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2^2 \\
\sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3^2_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3^M_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^2_{rst}\cdot x3_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^2_{rst}\cdot x3^2_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2^2 \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3^{M-1}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3^M_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2_{rst}\cdot x3^{2M-2}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^2_{rst}\cdot x3^{M-1}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^2_{rst}\cdot x3^M_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2^2 \\
\hline
\cdots & & & & & & & \\
\hline
\sum\sum\sum x1^L_{rst}\cdot x2^{M-1}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^{M-1}_{rst}\cdot x3_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2^{M-1}_{rst}\cdot x3^{M-1}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^M_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^M_{rst}\cdot x3_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2 \\
\sum\sum\sum x1^L_{rst}\cdot x2^{M-1}_{rst}\cdot x3_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^{M-1}_{rst}\cdot x3^2_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2^{M-1}_{rst}\cdot x3^M_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^M_{rst}\cdot x3_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^M_{rst}\cdot x3^2_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2 \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x1^L_{rst}\cdot x2^{M-1}_{rst}\cdot x3^{M-1}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^{M-1}_{rst}\cdot x3^M_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2^{M-1}_{rst}\cdot x3^{2M-2}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^M_{rst}\cdot x3^{M-1}_{rst} & \sum\sum\sum x1^L_{rst}\cdot x2^M_{rst}\cdot x3^M_{rst} & \cdots & \sum\sum\sum x1^L_{rst}\cdot x2
\end{array}\right.
$$

$X1^L_{rst}\cdot X2_{rst}\cdot X3_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2_{rst}\cdot X3^{M-1}_{rst}$		$\sum\sum\sum X1^L_{rst}\cdot X2^{M-1}_{rst}$	$\sum\sum\sum X1^L_{rst}\cdot X2^{M-1}_{rst}\cdot X3_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^{M-1}_{rst}\cdot X3^{M-1}_{rst}$
$X1^L_{rst}\cdot X2_{rst}\cdot X3^2_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2_{rst}\cdot X3^M_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^{M-1}_{rst}\cdot X3_{rst}$	$\sum\sum\sum X1^L_{rst}\cdot X2^{M-1}_{rst}\cdot X3^2_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^{M-1}_{rst}\cdot X3^M_{rst}$
				...			
$X1^L_{rst}\cdot X2_{rst}\cdot X3^M_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2_{rst}\cdot X3^{2M-2}_{rst}$		$\sum\sum\sum X1^L_{rst}\cdot X2^{M-1}_{rst}\cdot X3^{M-1}_{rst}$	$\sum\sum\sum X1^L_{rst}\cdot X2^{M-1}_{rst}\cdot X3^M_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^{M-1}_{rst}\cdot X3^{2M-2}_{rst}$
$X1^L_{rst}\cdot X2^2_{rst}\cdot X3_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^2_{rst}\cdot X3^{M-1}_{rst}$		$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}$	$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3^{M-1}_{rst}$
$X1^L_{rst}\cdot X2^2_{rst}\cdot X3^2_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^2_{rst}\cdot X3^M_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3_{rst}$	$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3^2_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3^M_{rst}$
				...			
$X1^L_{rst}\cdot X2^2_{rst}\cdot X3^M_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^2_{rst}\cdot X3^{2M-2}_{rst}$		$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3^{M-1}_{rst}$	$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3^M_{rst}$		$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3^{2M-2}_{rst}$
$X1^L_{rst}\cdot X2^M_{rst}\cdot X3_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3^{M-1}_{rst}$		$\sum\sum\sum X1^L_{rst}\cdot X2^{2M-2}_{rst}$	$\sum\sum\sum X1^L_{rst}\cdot X2^{2M-2}_{rst}\cdot X3_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^{2M-2}_{rst}\cdot X3^{M-1}_{rst}$
$X1^L_{rst}\cdot X2^M_{rst}\cdot X3^2_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3^M_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^{2M-2}_{rst}\cdot X3_{rst}$	$\sum\sum\sum X1^L_{rst}\cdot X2^{2M-2}_{rst}\cdot X3^2_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^{2M-2}_{rst}\cdot X3^M_{rst}$
				...			
$X1^L_{rst}\cdot X2^M_{rst}\cdot X3^M_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^M_{rst}\cdot X3^{2M-2}_{rst}$		$\sum\sum\sum X1^L_{rst}\cdot X2^{2M-2}_{rst}\cdot X3^{M-1}_{rst}$	$\sum\sum\sum X1^L_{rst}\cdot X2^{2M-2}_{rst}\cdot X3^M_{rst}$	...	$\sum\sum\sum X1^L_{rst}\cdot X2^{2M-2}_{rst}\cdot X3^{2M-2}_{rst}$

62

2

$$
A_{ii}^{*} = \left[\begin{array}{cccc|cccc}
\sum\sum\sum x1_{rst}^{2i-2} & \sum\sum\sum x1_{rst}^{2i-2} x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x3_{rst}^{m-1} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} \\
\sum\sum\sum x1_{rst}^{2i-2} x3_{rst} & \sum\sum\sum x1_{rst}^{2i-2} x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x3_{rst}^{m} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x1_{rst}^{2i-2} x3_{rst}^{m-1} & \sum\sum\sum x1_{rst}^{2i-2} x3_{rst}^{m} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x3_{rst}^{2m-2} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst}^{m-1} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst}^{m} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} \\
\hline
\sum\sum\sum x1_{rst}^{2i-2} x2_{rst} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst}^{m-1} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{2} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{2} x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{2} \\
\sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst}^{m} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{2} x3_{rst} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{2} x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{2} \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst}^{m-1} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst}^{m} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst} x3_{rst}^{2m-2} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{2} x3_{rst}^{m-1} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{2} x3_{rst}^{m} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{2} \\
\hline
\cdots & & & & & & & \\
\hline
\sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m-1} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m-1} x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m-1} x3_{rst}^{m-1} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m} x3_{rst} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m} \\
\sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m-1} x3_{rst} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m-1} x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m-1} x3_{rst}^{m} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m} x3_{rst} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m} x3_{rst}^{2} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m} \\
\cdots & & & & \cdots & & & \\
\sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m-1} x3_{rst}^{m-1} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m-1} x3_{rst}^{m} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m-1} x3_{rst}^{2m-2} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m} x3_{rst}^{m-1} & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m} x3_{rst}^{m} & \cdots & \sum\sum\sum x1_{rst}^{2i-2} x2_{rst}^{m}
\end{array}\right.
$$

$$
\begin{array}{lcl}
{}^{2L-2}_{rst}\, X2_{rst}\, X3_{rst} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\, X2_{rst}\, X3_{rst}^{N-1} \\
{}^{2L-2}_{rst}\, X2_{rst}\, X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\, X2_{rst}\cdot X3_{rst}^{N} \\
 & & \\
{}^{2L-2}_{rst}\, X2_{rst}\, X3_{rst}^{N} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\, X2_{rst}\cdot X3_{rst}^{2N-2} \\
\hline
{}^{2L-2}_{rst}\cdot X2_{rst}^{2}\, X3_{rst} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\, X2_{rst}^{2}\, X3_{rst}^{N-1} \\
{}^{2L-2}_{rst}\cdot X2_{rst}^{2}\, X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{2}\, X3_{rst}^{N} \\
 & & \\
{}^{2L-2}_{rst}\cdot X2_{rst}^{2}\, X3_{rst}^{N} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{2}\cdot X3_{rst}^{2N-2} \\
\hline
\end{array}
\qquad \cdots
$$

$$
\begin{array}{lcl}
{}^{2L-2}_{rst}\cdot X2_{rst}^{M}\, X3_{rst} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N-1} \\
{}^{2L-2}_{rst}\cdot X2_{rst}^{M}\cdot X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N} \\
 & & \\
{}^{2L-2}\, X2_{rst}^{M}\cdot X3_{rst}^{N} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\, X2_{rst}^{M}\, X3_{rst}^{2N-2}
\end{array}
\qquad \cdots
$$

$$
\left|\begin{array}{llcl}
\sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M-1} & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M-1}\cdot X3_{rst} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{N-1} \\
\sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M-1}\cdot X3_{rst} & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{N} \\
\cdots & & & \\
\sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{N-1} & \sum\sum\sum X1_{rst}^{2L-2}\, X2_{rst}^{M-1}\cdot X3_{rst}^{N} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{2N-3} \\
\hline
\sum\sum\sum X1_{rst}^{2L-2}\, X2_{rst}^{M} & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M}\cdot X3_{rst} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N-1} \\
\sum\sum\sum X1_{rst}^{2L-2}\, X2_{rst}^{M}\cdot X3_{rst} & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M}\cdot X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N} \\
\cdots & & & \\
\sum\sum\sum X1_{rst}^{2L-2}\, X2_{rst}^{M}\cdot X3_{rst}^{N-1} & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M}\cdot X3_{rst}^{N} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{M}\cdot X3_{rst}^{2N-2} \\
\hline
 & & & \\
\hline
\sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{2M-2} & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{2M-2}\cdot X3_{rst} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{2M-2}\cdot X3_{rst}^{N-1} \\
\sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{2M-2}\cdot X3_{rst} & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{2M-2}\cdot X3_{rst}^{2} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{2M-2}\cdot X3_{rst}^{N} \\
\cdots & & & \\
\sum\sum\sum X1_{rst}^{2L-2}\, X2_{rst}^{2M-2}\, X3_{rst}^{N-1} & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{2M-2}\, X3_{rst}^{N} & \cdots & \sum\sum\sum X1_{rst}^{2L-2}\cdot X2_{rst}^{2M-2}\cdot X3_{rst}^{2N-2}
\end{array}\right|
$$

2

$$
\mathbf{C}_1^{*} = \left|\begin{array}{c}
c_{111} \\ c_{112} \\ \cdots \\ c_{11N} \\ \hline
c_{121} \\ c_{122} \\ \cdots \\ c_{12N} \\ \hline
\cdots \\ \hline
c_{1M1} \\ c_{1M2} \\ \cdots \\ c_{1MN}
\end{array}\right|
\qquad
B_1^{*} = \left|\begin{array}{l}
\sum\sum\sum Y_{rst} \\
\sum\sum\sum Y_{rst}\cdot X3_{rst} \\
\quad\cdots \\
\sum\sum\sum Y_{rst}\cdot X3_{rst}^{N-1} \\ \hline
\sum\sum\sum Y_{rst}\cdot X2_{rst} \\
\sum\sum\sum Y_{rst}\cdot X2_{rst}\cdot X3_{rst} \\
\quad\cdots \\
\sum\sum\sum Y_{rst}\cdot X2_{rst}\cdot X3_{rst}^{N-1} \\ \hline
\quad\cdots \\ \hline
\sum\sum\sum Y_{rst}\cdot X2_{rst}^{M-1} \\
\sum\sum\sum Y_{rst}\cdot X2_{rst}^{M-1}\cdot X3_{rst} \\
\quad\cdots \\
\sum\sum\sum Y_{rst}\cdot X2_{rst}^{M-1}\cdot X3_{rst}^{N-1}
\end{array}\right|
$$

$$
\mathbf{C}_2^{*} = \left|\begin{array}{c}
c_{211} \\ c_{212} \\ \cdots \\ c_{21N} \\ \hline
c_{221} \\ c_{222} \\ \cdots \\ c_{22N} \\ \hline
\cdots \\ \hline
c_{2M1}
\end{array}\right.
\qquad
B_2^{*} = \left|\begin{array}{l}
\sum\sum\sum Y_{rst}\cdot X1_{rst} \\
\sum\sum\sum Y_{rst}\cdot X1_{rst}\cdot X3_{rst} \\
\quad\cdots \\
\sum\sum\sum Y_{rst}\cdot X1_{rst}\cdot X3_{rst}^{N-1} \\ \hline
\sum\sum\sum Y_{rst}\cdot X1_{rst}\cdot X2_{rst} \\
\sum\sum\sum Y_{rst}\cdot X1_{rst}\cdot X2_{rst}\cdot X3_{rst} \\
\quad\cdots \\
\sum\sum\sum Y_{rst}\cdot X1_{rst}\cdot X2_{rst}\cdot X3_{rst}^{N-1} \\ \hline
\quad\cdots \\ \hline
\sum\sum\sum Y_{rst}\cdot X1_{rst}\cdot X2_{rst}^{M-1}
\end{array}\right.
$$

$$
C_2^* = \begin{bmatrix} \vdots \\ C_{212} \\ \vdots \\ C_{21N} \\ \hline \vdots \\ \hline C_{2M1} \\ C_{2M2} \\ \vdots \\ C_{2MN} \end{bmatrix}
\qquad
B_2^* = \begin{bmatrix} \vdots \\ \vdots \\ \sum\sum\sum Y_{rst} \cdot X1_{rst} \cdot X2_{rst} \cdot X3_{rst}^{N-1} \\ \hline \vdots \\ \hline \sum\sum\sum Y_{rst} \cdot X1_{rst} \cdot X2_{rst}^{M-1} \\ \sum\sum\sum Y_{rst} \cdot X1_{rst} \cdot X2_{rst}^{M-1} \cdot X3_{rst} \\ \vdots \\ \sum\sum\sum Y_{rst} \cdot X1_{rst} \cdot X2_{rst}^{M-1} \cdot X3_{rst}^{N-1} \end{bmatrix}
$$

$$\vdots \qquad\qquad \vdots$$

$$
C_L^* = \begin{bmatrix} C_{L11} \\ C_{L12} \\ \vdots \\ C_{L1N} \\ \hline C_{L21} \\ C_{L22} \\ \vdots \\ C_{L2N} \\ \hline \vdots \\ \hline C_{LM1} \\ C_{LM2} \\ \vdots \\ C_{LMN} \end{bmatrix}
\qquad
B_L^* = \begin{bmatrix} \sum\sum\sum Y_{rst} \cdot X1_{rst}^{L-1} \\ \sum\sum\sum Y_{rst} \cdot X1_{rst}^{L-1} \cdot X3_{rst} \\ \vdots \\ \sum\sum\sum Y_{rst} \cdot X1_{rst}^{L-1} \cdot X3_{rst}^{N-1} \\ \hline \sum\sum\sum Y_{rst} \cdot X1_{rst}^{L-1} \cdot X2_{rst} \\ \sum\sum\sum Y_{rst} \cdot X1_{rst}^{L-1} \cdot X2_{rst} \cdot X3_{rst} \\ \vdots \\ \sum\sum\sum Y_{rst} \cdot X1_{rst}^{L-1} \cdot X2_{rst} \cdot X3_{rst}^{N-1} \\ \hline \vdots \\ \hline \sum\sum\sum Y_{rst} \cdot X1_{rst}^{L-1} \cdot X2_{rst}^{M-1} \\ \sum\sum\sum Y_{rst} \cdot X1_{rst}^{L-1} \cdot X2_{rst}^{M-1} \cdot X3_{rst} \\ \vdots \\ \sum\sum\sum Y_{rst} \cdot X1_{rst}^{L-1} \cdot X2_{rst}^{M-1} \cdot X3_{rst}^{N-1} \end{bmatrix}
$$

2

4.4. Structure of Equation System:

4.4.1. System Matrix A

It can be seen, that all submatrices 1. class on the same diagonal bands 2. order of the system matrix are identical and therefore will be renamed from $A^*_{i,j}$ to A^*_k

$$
\begin{aligned}
A_1^* &= A_{11}^* \\
A_2^* &= A_{21}^* = A_{12}^* \\
A_3^* &= A_{31}^* = A_{22}^* = A_{13}^* \\
&\dots \\
A_L^* &= A_{L,1}^* = A_{L-1,2}^* = A_{L-2,3}^* = \dots = A_{2,L-1}^* = A_{1,L}^* \\
A_{L+1}^* &= A_{L,2}^* = A_{L-1,3}^* = \dots = A_{3,L-1}^* = A_{2,L}^* \\
A_{L+2}^* &= A_{L,3}^* = \dots = A_{4,L-1}^* = A_{3,L}^* \\
&\dots \\
A_{2L-1}^* &= A_{L,L}^*
\end{aligned}
$$

The submatrices 1. class A^*_k (k=1,...,2L-1) themselves are divided up into submatrices 2. class.

$$
A_k^* = \begin{vmatrix}
A_{k;11}^{**} & A_{k;12}^{**} & \dots & A_{k;1,M}^{**} \\
A_{k;21}^{**} & A_{k;22}^{**} & \dots & A_{k;2,M}^{**} \\
\dots & \dots & \dots & \dots \\
A_{k;M1}^{**} & A_{k;M,2}^{**} & \dots & A_{k;M,M}^{**}
\end{vmatrix}
$$

The double asterisk indicates a submatrix 2. class. All submatrices 2. class on the same diagonal bands 2. order of the submatrices 1. class A_k^* (k = 1,...,2L) are identical and therefore will be renamed from $A_{k;i,j}^{**}$ to $A_{k;l}^{**}$

$$A_{k;1}^{**} = A_{k;1,1}^{**}$$

$$A_{k;2}^{**} = A_{k;2,1}^{**} = A_{k;1,2}^{**}$$

$$A_{k;3}^{**} = A_{k;3,1}^{**} = A_{k;2,2}^{**} = A_{k;1,3}^{**}$$

...

$$A_{k;M}^{**} = A_{k;M,1}^{**} = A_{k;M-1,2}^{**} = A_{k;M-2,3}^{**} = \ldots = A_{k;2,M-1}^{**} = A_{k;1,M}^{**}$$

$$A_{k;M+1}^{**} = A_{k;M,2}^{**} = A_{k;M-1,3}^{**} = \ldots = A_{k;3,M-1}^{**} = A_{k;2,M}^{**}$$

$$A_{k;M+2}^{**} = A_{k,M,3}^{**} = \ldots = A_{k;4,M-1}^{**} = A_{k;3,M}^{**}$$

...

$$A_{k;2M-1}^{**} = A_{k;1,M}^{**}$$

where $A_{k;1}^{**}$ can be written as

$$A_{k;l}^{**} \left| \begin{array}{llll} \sum\sum\sum X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} & \sum\sum\sum X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3_{rst} & \ldots & \sum\sum\sum X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3_{rst}^{M-1} \\ \sum\sum\sum X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3_{rst} & \sum\sum\sum X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3_{rst}^{2} & \ldots & \sum\sum\sum X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3_{rst}^{M} \\ \ldots & \ldots & \ldots & \ldots \\ \sum\sum\sum X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3_{rst}^{M-1} & \sum\sum\sum X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3_{rst}^{M} & \ldots & \sum\sum\sum X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3_{rst}^{2M-2} \end{array} \right.$$

$$k = 1,\ldots,2L-1;\ l = 1,\ldots,2M-1$$

Note, that this submatrix 2. class is not only symmetrical, but also the elements on each diagonal line 2. order are identical. The elements $a_{k;l;i,j}$ of the submatrix 2. class $A^{**}_{k;l}$ therefore are renamed to $a_{k;L;m}$; L specifies the place of the submatrix 2. class $A^{**}_{k;l}$ in the submatrix 1. class A^{*}_{k}, k specifies the place of the submatrix 1. class A^{*}_{k} in the system matrix A and i,j specifies the place of the element $a_{k;l;i,j}$ in the submatrix 2. class $A^{**}_{k;l}$

$$a_{k;l;1} = a_{k,l;1,1}$$

$$a_{k;l;2} = a_{k;l;2,1} = a_{k;l;1,2}$$

$$a_{k;l;3} = a_{k;l;3,1} = a_{k;l;2,2} = a_{k;l;1,3}$$

...

$$a_{k;l;N} = a_{k;l;N,1} = a_{k;l;N-1,2} = a_{k;l;N-2,3} = \dots = a_{k;l;2,N-1} = a_{k;l;1,N}$$

$$a_{k;l;N+1} = a_{k;l;N,2} = a_{k;l;N-1,3} = \dots = a_{k;l;3,N-1} = a_{k;l;2,N}$$

$$a_{k;l;N+2} = a_{k;l;N,3} = \dots = a_{k;l;4,N-1} = a_{k;l;3,N}$$

...

$$a_{k;l;2N-1} = a_{k;l;N,N}$$

where $a_{k;l;m}$ can be written as

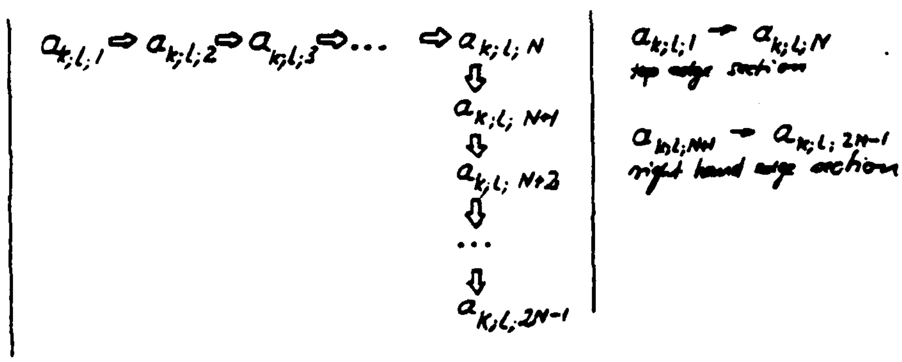
$$a_{k;l;m} = \sum_{r=1}^{\mathrm{NPWTS1}} \{ \sum_{s=1}^{\mathrm{NPWTS2}} \{ \sum_{t=1}^{\mathrm{NPWTS3}} X1^{k-1}_{rst} \cdot X2^{l-1}_{rst} \cdot X3^{m-1}_{rst} \}\} \qquad (4.4.)$$

$$k = 1, \dots, 2L-1 \; ; \; l = 1, \dots, 2M-1 \; ; \; m = 1, \dots, 2N-1$$

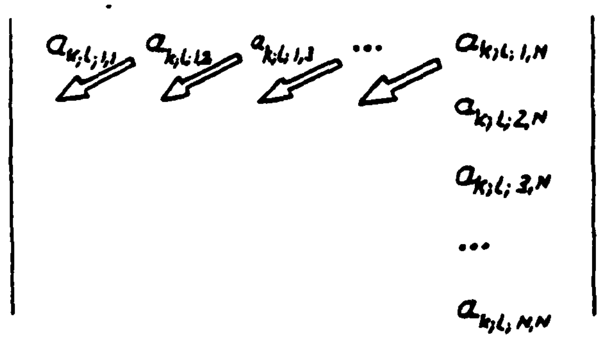
This operation is programmed in REAL FUNCTION S4 (NPNTS1,NPNTS2,NPNTS3, IPOWR1,IPOWR2,IPOWR3,IY). The data X1, X2, X3 and Y are known to this function through a common block named DATA4. So only NPNTS1, NPNTS2, NPNTS3, IPOWR1 (= k-1), IPOWR2 (= L-1), IPOWR3 (= m-1) and IY (=0) have to be passed to the function and the value of $a_{k;l;m}$ is returned through the function name S4.

SUBROUTINE MAT42 presets the submatrices 2. class $A^{**}_{k,l}$ in the following way

i) <u>Preset edge section elements</u> (using REAL FUNCTION S4)



ii) <u>Copy defined elements diagonally</u>



$$\left|\begin{array}{ccccc} a_{k;l;1,1} & a_{k;l;1,2} & a_{k;l;1,3} & \cdots & a_{k;l;1,N} \\ a_{k;l;2,1} & a_{k;l;2,2} & a_{k;l;3,3} & \cdots & a_{k;l;2,N} \\ & & & & a_{k;l;3,N} \\ & & & & \cdots \\ & & & & a_{k;l;N,N} \end{array}\right|$$

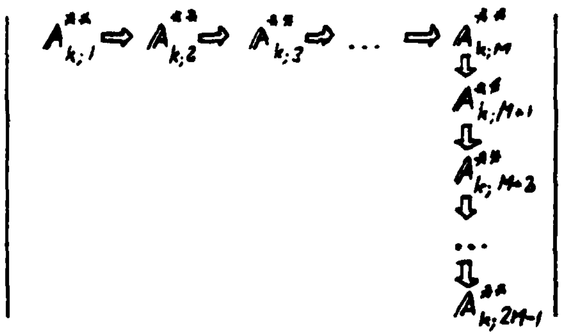
$$\left|\begin{array}{ccccc} a_{k;l;1,1} & a_{k;l;1,2} & a_{k;l;1,3} & \cdots & a_{k;l;1,N} \\ a_{k;l;2,1} & a_{k;l;2,2} & a_{k;l;2,3} & \cdots & a_{k;l;2,N} \\ a_{k;l;2,1} & a_{k;l;3,2} & a_{k;l;3,3} & \cdots & a_{k;l;2,N} \\ & & & & \cdots \\ & & & & a_{k;l;N,N} \end{array}\right|$$

$$\left|\begin{array}{ccccc} a_{k;l;1,1} & a_{k;l;1,2} & a_{k;l;1,3} & \cdots & a_{k;l;1,N} \\ a_{k;l;2,1} & a_{k;l;2,2} & a_{k;l;2,3} & \cdots & a_{k;l;2,N} \\ a_{k;l;2,1} & a_{k;l;3,2} & a_{k;l;2,3} & \cdots & a_{k;l;3,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & a_{k;l;N,N} \end{array}\right|$$

$$A^{**}_{k;l} = \left|\begin{array}{ccccc} a_{k;l;1,1} & a_{k;l;1,2} & a_{k;l;1,3} & \cdots & a_{k;l;1,N} \\ a_{k;l;2,1} & a_{k;l;2,2} & a_{k;l;2,3} & \cdots & a_{k;l;2,N} \\ a_{k;l;3,1} & a_{k;l;3,2} & a_{k;l;2,3} & \cdots & a_{k;l;3,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{k;l;N,1} & a_{k;l;N,2} & a_{k;l;N,3} & \cdots & a_{k;l;N,N} \end{array}\right|$$

This subroutine that needs NPNTS1,NPNTS2,NPNTS3,IPOWR1(=k-1) and IPOWR2(=L-1) as input parameters, returns the submatrix 2. class $A^{**}_{k;L}$ (SUBM2(4,4)) to SUBROUTINE MAT41, which foresets the submatrices 1. class A^{*}_{k} in the following way

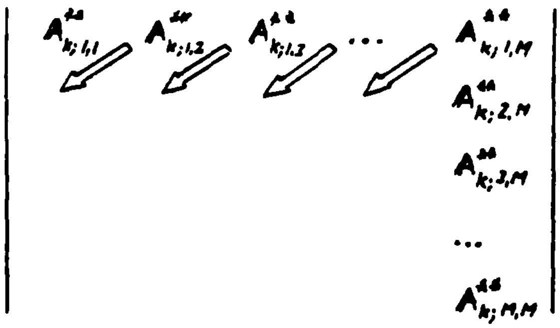
i) <u>Preset edge section submatrices</u> (using SUBROUTINE MAT42)

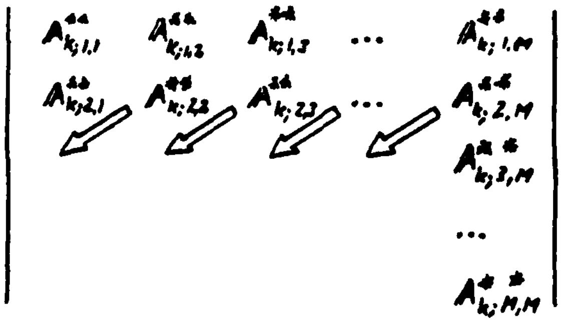


$A_{k;1} \rightarrow A_{k;M}$
top edge section

$A_{k;M+1} \rightarrow A_{k;2M-1}$

ii) <u>Copy defined submatrices diagonally</u>





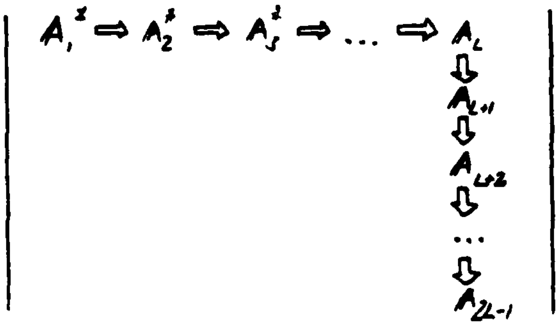
$$\begin{vmatrix} A^{**}_{k;1,1} & A^{**}_{k;1,2} & A^{**}_{k;1,3} & \cdots & A^{**}_{k;1,M} \\ A^{**}_{k;2,1} & A^{**}_{k;2,2} & A^{**}_{k;2,3} & \cdots & A^{**}_{k;2,M} \\ A^{**}_{k;3,1} & A^{**}_{k;3,2} & A^{**}_{k;3,3} & \cdots & A^{**}_{k;3,M} \\ & & & & \cdots \\ & & & & A^{**}_{k;M,M} \end{vmatrix}$$

$$\begin{vmatrix} A^{**}_{k;1,1} & A^{**}_{k;1,2} & A^{**}_{k;1,3} & \cdots & A^{**}_{k;1,M} \\ A^{**}_{k;2,1} & A^{**}_{k;2,2} & A^{**}_{k;2,3} & \cdots & A^{**}_{k;2,M} \\ A^{**}_{k;3,1} & A^{**}_{k;3,2} & A^{**}_{k;3,3} & \cdots & A^{**}_{k;3,M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & A^{**}_{k;M,M} \end{vmatrix}$$

$$A^{*}_{k} = \begin{vmatrix} A^{**}_{k;1,1} & A^{**}_{k;1,2} & A^{**}_{k;1,3} & \cdots & A^{**}_{k;1,M} \\ A^{**}_{k;2,1} & A^{**}_{k;2,2} & A^{**}_{k;2,3} & \cdots & A^{**}_{k;2,M} \\ A^{**}_{k;3,1} & A^{**}_{k;3,2} & A^{**}_{k;3,3} & \cdots & A^{**}_{k;3,M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ A^{**}_{k;M,1} & A^{**}_{k;M,2} & A^{**}_{k;M,3} & \cdots & A_{k;M,M} \end{vmatrix}$$

This subroutine, that needs NPNTS1,NPNTS2,NPNTS3,M,N,IPOWR1(=L-1) as input parameters, returns the subroutine 1st class A^{*}_{k} (SUBM1(16,16)) to SUBROUTINE MAT4, which presets the entire system matrix A in the following way.

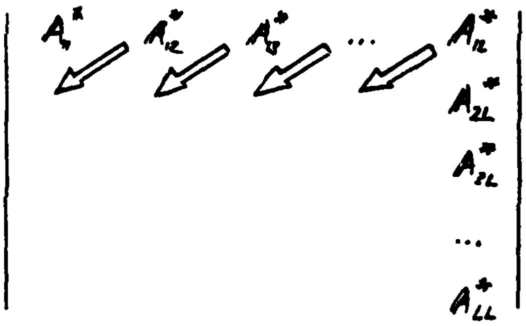
i) <u>Preset edge section submatrices</u> (using SUBROUTINE MAT41)

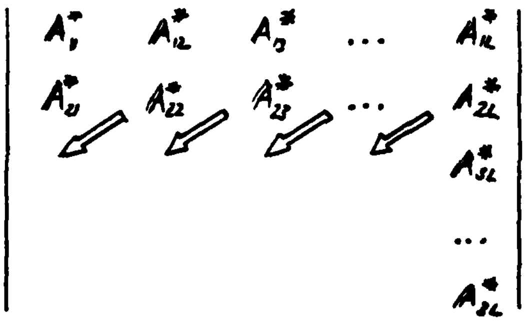


$A_1 \rightarrow A_L$ top edge section

$A_{L+1} \rightarrow A_{2L-1}$ right hand edge sections

ii) <u>Copy defined submatrices diagonally</u>





$$\begin{vmatrix} A^*_{11} & A^*_{12} & A^*_{13} & \cdots & A^*_{1L} \\ A^*_{21} & A^*_{22} & A^*_{23} & \cdots & A^*_{2L} \\ A^*_{31} & A^*_{32} & A^*_{33} & \cdots & A^*_{3L} \\ & & & & \cdots \\ & & & & A_{LL} \end{vmatrix}$$

$$\begin{vmatrix} A^*_{11} & A^*_{12} & A^*_{13} & \cdots & A_{1L} \\ A^*_{21} & A^*_{22} & A^*_{23} & \cdots & A_{2,L} \\ A^*_{3,1} & A^*_{3,2} & A^*_{3,3} & \cdots & A_{3,L} \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{vmatrix}$$

$$A = \begin{vmatrix} A^*_{11} & A^*_{12} & A^*_{13} & \cdots & A^*_{1L} \\ A^*_{21} & A^*_{22} & A^*_{23} & \cdots & A_{2L} \\ A^*_{31} & A^*_{32} & A^*_{33} & \cdots & A_{3L} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ A_{L1} & A_{L2} & A_{L3} & \cdots & A_{LL} \end{vmatrix}$$

4.4.2. <u>Right Hand Side Vector B</u>:

The right hand side subvectors 1. class B_k^* generally can be written as

$$
B_k^* = \left|
\begin{array}{l}
\sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \\
\sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X3_{rst} \\
\quad \cdots \\
\sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X3_{rst}^{N-1} \\
\\
\sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst} \\
\sum\sum\sum Y_{rst} X1_{rst}^{k-1} \cdot X2_{rst} \cdot X3_{rst} \\
\quad \cdots \\
\sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} X2_{rst} \cdot X3_{rst}^{N-1} \\
\\
\\
\quad \cdots \\
\\
\\
\sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{M-1} \\
\sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{M-1} \cdot X3_{rst} \\
\quad \cdots \\
\sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{M-1} \cdot X3_{rst}^{N-1}
\end{array}
\right|
$$

$$k = 1, \ldots, L$$

The subvectors 1. class B_k^* (k = 1,...,L) themselves will be divided up into subvectors 2. class:

$$B_K^* = \begin{vmatrix} B_{k,1}^{**} \\ B_{k,2}^{**} \\ \cdots \\ B_{k,L}^{**} \end{vmatrix}$$

where subvectors 2. class $B_{k;L}^{**}$ can be written as

$$B_{k;l}^{**} = \begin{vmatrix} \sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \\ \sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3 \\ \\ \sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3^{N-1} \end{vmatrix}$$

$$k = 1, \ldots, L; \; l = 1, \ldots, M; \; m = 1, \ldots, N$$

Finally, the elements of the subvectors 2. class $B_{k;1}$ can be written as

$$b_{k;l;m} = \sum\sum\sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{l-1} \cdot X3_{rst}^{m-1}$$

$$k = 1, \ldots, L; \; l = 1, \ldots, M; \; m = 1, \ldots, N$$

To calculate $b_{k;1;m}$, REAL FUNCTION S4 is used, again. Data X1, X2, X3 and Y are available through COMMON block DATA4. NPNTS1, NPNTS2, NPNTS3, IPOWR1 (= k-1), IPOWR2 (= L-1) IPOWR3 (= m-1) and IY (=1) have to be passed to the function and the value of $b_{k;1;m}$ is returned through the function name S4. Using FORTRAN programming language, the allocation of $b_{k;1;m}$ can be performed by three stacked DO-loops.

```
      I=0
      DO 07 L1=1,L,1
      IPOWR1=L1-1
      DO 08 M1=1,M,1
      IPOWR2=M1-1
      DO 08 N1=1,N,1
      IPOWR3=N1-1
      I=I+1
   08 B(I)=S4(NPNTS1,NPNTS2,NPNTS3,IPOWR1,IPOWR2,IPOWR3,1)
```

4.5. Software:

The software to compute the coefficients for a two dimensional approximation is described in APPENDIX A and is implemented in the TPL HP-21MX computer system. To work correctly with these program modules, the user has to conform to the following conventions.

i) Provide the data in four arrays (Type: REAL) of 5*5*5* elements through a COMMON block, named DATA4.

```
COMMON / DATA4 / X1,X2,X3,Y
REAL X1(5,5,5),X2(5,5,5),X3(5,5,5),Y(5,5,5)
```

ii) Dimension an array (Type: REAL) of 5*5*5 elements to contain the coefficients,

```
REAL COEF (5,5,5)
```

iii) Define the parameters NPNTS1, NPNTS2,

NPNTS3,L,M,N and IPRINT

NPNTS1... # of X1 variations
$L \leq NPNTS1 \leq 5$

NPNTS2... # of X2 variations
$M \leq NPNTS2 \leq 5$

NPNTS3... # of X3 variations
$N \leq NPNTS3 \leq 5$

L ... (desired order of approximation polynominal w.r.t. X1) + 1
$1 \leq L \leq 4$

M ... (desired order of approximation polynominal w.r.t. X2) + 1
$1 \leq M \leq 4$

N ... (desired order of approximation polynominal w.r.t. X3) + 1
$1 \leq N \leq 4$

IPRINT ... Controls quantity of print out

- 2 ... Print system matrix and right hand side vector before and after Gauss Jordan Elimination.
- 1 ... Print equation system after Gauss Jordan Elimination
- <0... No print out
- >0... Print equation (4.1.) with the actual parameters.

iv) If a user program uses subroutine MAT4, the software modules have to be loaded using the procedure in section 1.4.iv.

If all these requirements are met, the correct call for the subroutine is:

```
CALL MAT4 (NPNTS1,NPNTS2,NPNTS3,L,M,N,COEF,IPRINT)
```

Upon completed execution of this approximation routine array COEF contains the coefficients. Externals used by MAT4: AB4, DATA4, MAT41, MAT42, S4. Under no circumstances may the user use any of these names for a modules of his own user program.

4.6. Sample User Program

4.6.1 FTN4 Compiler listing of sample user program

```
FTN4,L
      PROGRAM FITR (3,99)
C
C  APPROXIMATION PROGRAM TO FIT A SURFACE THROUGH DATA POINTS Y,
C  WHICH DEPEND ON THREE INDEPENDENT PARAMETERS X1, X2 AND X3.
C
C  AUTHOR: HANS ZEBNER
C  DATE  : AUGUST 26, 1980
C
     *, Hans Zebner:     4D-Approximation;  Y=f(X1,X2,X3).

      COMMON / DATA4 / X1,X2,X3,Y
      COMMON / AB4   / A,B

      REAL X1(5,5,5),X2(5,5,5),X3(5,5,5),Y(5,5,5)
      REAL A(64,64),B(64)

      REAL COEF(5,5,5)
      INTEGER IDCB(144),IFILE(3),NOCR(2),ICLR(3)

      DATA NOCR /000033B,040433B/, ICLR /015524B,015515B,006537B/

  101 FORMAT (" ENTER NPNTS1 NPNTS2 NPNTS3 "2A2
     *)
  102 FORMAT (" ENTER L M N "2A2)
  103 FORMAT (" ENTER DATA FILE NAME "2A2)
  104 FORMAT (3A2)
  131 FORMAT (" ENTER IPRINT "2A2)
  132 FORMAT (" OUTPUT RAW DATA YES OR NO "2A2)
  105 FORMAT (/" X1")
  106 FORMAT (/1X,I2". DATA SET"/)
  107 FORMAT (3X,12I10)
  108 FORMAT (1X,I2,14F10.3)
  109 FORMAT (/" X2")
  110 FORMAT (/" X3")
  111 FORMAT (/" Y")
  149 FORMAT (""((3A2)))
 1101 FORMAT ("HANNES")
C
C  READ RAW DATA FROM FILE
C
   01 WRITE (1,101) NOCR
      READ  (1,  *) NPNTS1,NPNTS2,NPNTS3
      WRITE (1,149) ICLR
      NMAX=5
      IF (NPNTS1.GT.NMAX) GO TO 01
      IF (NPNTS2.GT.NMAX) GO TO 01
      IF (NPNTS3.GT.NMAX) GO TO 01
   02 WRITE (1,102) NOCR
      READ  (1,  *) L,M,N
      WRITE (1,149) ICLR
      IF (L*M*N.GT.64) GO TO 02
   03 WRITE (1,103) NOCR
      READ  (1,104) IFILE
      WRITE (1,149) ICLR
   31 WRITE (1,131) NOCR
      READ  (1,  *) IPRINT
      WRITE (1,149) ICLR
      WRITE (1,132) NOCR
      READ  (1,104) IDUM
      WRITE (1,149) ICLR
      CALL OPEN (IDCB,IERR,IFILE)
      IF (IERR.LT.0) GO TO 03
      IL=2*NPNTS1*NPNTS2*NPNTS3
      CALL READF (IDCB,IERR,X1,IL,LEN,1)
      IF (IERR.LT.0) STOP 0001
      CALL READF (IDCB,IERR,X2,IL,LEN,54)
      IF (IERR.LT.0) STOP 0002
      CALL READF (IDCB,IERR,X3,IL,LEN,107)
      IF (IERR.LT.0) STOP 0003
      CALL READF (IDCB,IERR,Y,IL,LEN,160)
      IF (IERR.LT.0) STOP 0004
      CALL CLOSE (IDCB,IERR)
      IF (IERR.LT.0) STOP 0005
      IF (IDUM.NE.2HYE) GO TO 17
      WRITE (6,105)
```

```
      DO 04 K1=1,NPNTS1,1
      WRITE (6,106) K1
      WRITE (6,107) (K3,K3=1,NPNTS3,1)
      DO 04 K2=1,NPNTS2,1
04    WRITE (6,108) K2,(X1(K1,K2,K3),K3=1,NPNTS3,1)
      WRITE (6,109)
      DO 05 K1=1,NPNTS1,1
      WRITE (6,106) K1
      WRITE (6,107) (K3,K3=1,NPNTS3,1)
      DO 05 K2=1,NPNTS2,1
05    WRITE (6,108) K2,(X2(K1,K2,K3),K3=1,NPNTS3,1)
      WRITE (6,110)
      DO 06 K1=1,NPNTS1,1
      WRITE (6,106) K1
      WRITE (6,107) (K3,K3=1,NPNTS3,1)
      DO 06 K2=1,NPNTS2,1
06    WRITE (6,108) K2,(X3(K1,K2,K3),K3=1,NPNTS3,1)
      WRITE (6,111)
      DO 07 K1=1,NPNTS1,1
      WRITE (6,106) K1
      WRITE (6,107) (K3,K3=1,NPNTS3,1)
      DO 07 K2=1,NPNTS2,1
07    WRITE (6,108) K2,(Y(K1,K2,K3),K3=1,NPNTS3,1)
17    CALL MAT4 (NPNTS1,NPNTS2,NPNTS3,L,M,N,COEF,IPRINT)
      CALL CODE
      WRITE (IFILE,1101)
      CALL OPEN (IDCB,IERR,IFILE,IOPTN,0,26,144)
      IF (IERR.LT.0) STOP 0006
      CALL WRITF (IDCB,IERR,COEF,250)
      IF (IERR.LT.0) STOP 0007
      CALL CLOSE (IDCB,IERR)
      IF (IERR.LT.0) STOP 0010
      STOP 0077
      END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 01259 COMMON = 00000

4.6.2. Load map of sample user program FITR

```
FITR    10042 12414   Hans Zebner:      4D-Approximation:  Y=f(X1,X2,X3).
```

```
to load these program modules, enter (from LOADR):  MS,$TPLBL

DATA4   12415 14364   4D-Approximation  / DATA4 /
MAT4    14365 20251   4D-Approximation arrange system matrix and vector.
MAT41   20252 20666   4D-Approximation      arrange submatrix 1. class.
MAT42   20667 21066   4D-Approximation      arrange submatrix 2. class.
S4      21067 21370   4D-Approximation              compute summations.
AB4     21371 41570   4D-Approximation  / AB4   /
```

```
LOGLU   41571 41646   92067-16268 REV.1903 790228
READF   41647 42610   92067-16125 REV.1940 790719
OPEN    42611 43106   92067-16125 REV.1903 790215
CLOSE   43107 43316   92067-16125 REV.1903 781229
CLRIO   43317 43325   750701  24998-16001
OVRD.   43326 43326   92067-16125 REV.1903 780526
$SMVE   43327 43415   92067-16268 REV.1903 790202
LURQ    43416 44000   92067-16268 REV.1903 790223
.DADS   44001 44110    780818  24998-16001
.DMP    44111 44256    780818  24998-16001
.DDI    44257 44557    781021  24998-16001
SESSN   44560 44575   92067-16125 REV.1903 780413
R/W$    44576 44734   92067-16125 REV.1903 781214
P.PAS   44735 44763   92067-16125 REV.1903 740801
.DNG    44764 44773    780818  24998-16001
PAUSE   44774 45074   771122  24998-16001
$ALRN   45075 45212   92067-16268 REV.1903 770715
FMTIO   45213 46511    24998-16002 REV.1926 790417
ERR0    46512 46601    771122  24998-16001
ABS     46602 46610     750701  24998-16001
.DDE    46611 46622    780818  24998-16001
.DIN    46623 46630    780818  24998-16001
.RTOI   46631 46724    780921  24998-16001
.FPWR   46725 46766    781106  24998-16001
.SBT    46767 47027    770518  24998-16001
..FCM   47030 47044   750701  24998-16001
PAU.E   47045 47045   750701  24998-16001
ER0.E   47046 47046   750701  24998-16001
$OPEN   47047 47223   92067-16125 REV.1903 790103
RW$UB   47224 47571   92067-16125 REV.1903 781003
RWND$   47572 47714   92067-16125 REV.1903 780801
FRMTR   47715 53352    24998-16002 REV.1926 790503
FMT.E   53353 53353    24998-16002 REV.1901 781107
REIO    53354 53500   92067-16268 REV.1903 790316
RMPAR   53501 53543   781106  24998-16001
PNAME   53544 53611    771121  24998-16001
LUTRU   53612 53720   92067-16268 REV.1903 790223
$SETP   53721 53745    781106  24998-16001
.CFER   53746 54023   750701  24998-16001
.LBT    54024 54054    770518  24998-16001

20 PAGES RELOCATED    20 PAGES REQ'D    NO PAGES EMA      NO PAGES MSEG
LINKS:BP     PROGRAM:LB   LOAD:TE   COMMON:NC
 /LOADR:FITR    READY AT  3:53 PM  FRI., 26  SEPT, 1980

 /LOADR:$END
```

4.6.3. Results

printed output

X1

1. DATA SET

	1	2	3	4	5
1	.100	.100	.100	.100	.100
2	.100	.100	.100	.100	.100
3	.100	.100	.100	.100	.100
4	.100	.100	.100	.100	.100
5	.100	.100	.100	.100	.100

2. DATA SET

	1	2	3	4	5
1	.200	.200	.200	.200	.200
2	.200	.200	.200	.200	.200
3	.200	.200	.200	.200	.200
4	.200	.200	.200	.200	.200
5	.200	.200	.200	.200	.200

3. DATA SET

	1	2	3	4	5
1	.300	.300	.300	.300	.300
2	.300	.300	.300	.300	.300
3	.300	.300	.300	.300	.300
4	.300	.300	.300	.300	.300
5	.300	.300	.300	.300	.300

4. DATA SET

	1	2	3	4	5
1	.400	.400	.400	.400	.400
2	.400	.400	.400	.400	.400
3	.400	.400	.400	.400	.400
4	.400	.400	.400	.400	.400
5	.400	.400	.400	.400	.400

5. DATA SET

	1	2	3	4	5
1	.500	.500	.500	.500	.500
2	.500	.500	.500	.500	.500
3	.500	.500	.500	.500	.500
4	.500	.500	.500	.500	.500
5	.500	.500	.500	.500	.500

X2

1. DATA SET

	1	2	3	4	5
1	-20.000	-20.000	-20.000	-20.000	-20.000
2	-10.000	-10.000	-10.000	-10.000	-10.000
3	0.000	0.000	0.000	0.000	0.000
4	10.000	10.000	10.000	10.000	10.000
5	20.000	20.000	20.000	20.000	20.000

2. DATA SET

	1	2	3	4	5
1	-20.000	-20.000	-20.000	-20.000	-20.000
2	-10.000	-10.000	-10.000	-10.000	-10.000
3	0.000	0.000	0.000	0.000	0.000
4	10.000	10.000	10.000	10.000	10.000
5	20.000	20.000	20.000	20.000	20.000

3. DATA SET

	1	2	3	4	5
1	-20.000	-20.000	-20.000	-20.000	-20.000
2	-10.000	-10.000	-10.000	-10.000	-10.000
3	0.000	0.000	0.000	0.000	0.000
4	10.000	10.000	10.000	10.000	10.000
5	20.000	20.000	20.000	20.000	20.000

4. DATA SET

1	2	3	4	5

1	-20.000	-20.000	-20.000	-20.000	-20.000
2	-10.000	-10.000	-10.000	-10.000	-10.000
3	0.000	0.000	0.000	0.000	0.000
4	10.000	10.000	10.000	10.000	10.000
5	20.000	20.000	20.000	20.000	20.000

5. DATA SET

	1	2	3	4	5
1	-20.000	-20.000	-20.000	-20.000	-20.000
2	-10.000	-10.000	-10.000	-10.000	-10.000
3	0.000	0.000	0.000	0.000	0.000
4	10.000	10.000	10.000	10.000	10.000
5	20.000	20.000	20.000	20.000	20.000

X3

1. DATA SET

	1	2	3	4	5
1	-20.000	-10.000	0.000	10.000	20.000
2	-20.000	-10.000	0.000	10.000	20.000
3	-20.000	-10.000	0.000	10.000	20.000
4	-20.000	-10.000	0.000	10.000	20.000
5	-20.000	-10.000	0.000	10.000	20.000

2. DATA SET

	1	2	3	4	5
1	-20.000	-10.000	0.000	10.000	20.000
2	-20.000	-10.000	0.000	10.000	20.000
3	-20.000	-10.000	0.000	10.000	20.000
4	-20.000	-10.000	0.000	10.000	20.000
5	-20.000	-10.000	0.000	10.000	20.000

3. DATA SET

	1	2	3	4	5
1	-20.000	-10.000	0.000	10.000	20.000
2	-20.000	-10.000	0.000	10.000	20.000
3	-20.000	-10.000	0.000	10.000	20.000
4	-20.000	-10.000	0.000	10.000	20.000
5	-20.000	-10.000	0.000	10.000	20.000

4. DATA SET

	1	2	3	4	5
1	-20.000	-10.000	0.000	10.000	20.000
2	-20.000	-10.000	0.000	10.000	20.000
3	-20.000	-10.000	0.000	10.000	20.000
4	-20.000	-10.000	0.000	10.000	20.000
5	-20.000	-10.000	0.000	10.000	20.000

5. DATA SET

	1	2	3	4	5
1	-20.000	-10.000	0.000	10.000	20.000
2	-20.000	-10.000	0.000	10.000	20.000
3	-20.000	-10.000	0.000	10.000	20.000
4	-20.000	-10.000	0.000	10.000	20.000
5	-20.000	-10.000	0.000	10.000	20.000

Y

1. DATA SET

	1	2	3	4	5
1	.400	.700	.800	.700	.400
2	.450	.750	.850	.750	.450
3	.500	.800	.900	.800	.500
4	.450	.750	.850	.750	.450
5	.400	.700	.800	.700	.400

2. DATA SET

	1	2	3	4	5
1	.463	.763	.863	.763	.463
2	.513	.813	.913	.813	.513
3	.563	.863	.963	.863	.563
4	.513	.813	.913	.813	.513

5	.463	.763	.863	.763	.463

3. DATA SET

	1	2	3	4	5
1	.650	.950	1.050	.950	.650
2	.700	1.000	1.100	1.000	.700
3	.750	1.050	1.150	1.050	.750
4	.700	1.000	1.100	1.000	.700
5	.650	.950	1.050	.950	.600

4. DATA SET

	1	2	3	4	5
1	.963	1.263	1.363	1.263	.963
2	1.013	1.313	1.413	1.313	1.013
3	1.063	1.363	1.463	1.363	1.063
4	1.013	1.313	1.413	1.313	1.013
5	.963	1.263	1.363	1.263	.963

5. DATA SET

	1	2	3	4	5
1	1.400	1.700	1.800	1.700	1.400
2	1.450	1.750	1.850	1.750	1.450
3	1.500	1.800	1.900	1.800	1.500
4	1.450	1.750	1.850	1.750	1.450
5	1.400	1.700	1.800	1.700	1.400

Coefficients COEF(I,J,K)

I = 1

	K = 1	2	3
J = 1	-.418E+04	-.965E+03	.691E+02
J = 2	-.134E+01	-.390E-01	.282E-02
J = 3	.121E+02	-.284E+01	-.203E+00
	K = 1	2	3

I = 2

	K = 1	2	3
J = 1	.458E+05	-.106E+05	-.757E+03
J = 2	.143E+02	.417E+00	-.300E-01
J = 3	-.133E+03	.312E+02	.222E+01
	K = 1	2	3

I = 3

	K = 1	2	3
J = 1	-.994E+05	.230E+05	.165E+04
J = 2	-.238E+02	-.696E+00	.499E-01
J = 3	.289E+03	-.678E+02	-.484E+01
	K = 1	2	3

5. CONCLUSIONS AND RECOMMENDATIONS

Compared to the probe calibration approach described in Ref 3. the method presented here provides a much simpler way to both calculate the probe calibration surfaces from measured data and to apply the calibration to on-line data reduction. The iteration required in the method of Ref 3 is completely omitted. It is noted however, that the Gauss Jordan Elimination lacks some sophistication. The system matrix for example undergoes the elimination without prior conditioning. Particularly in the case of the 3-D and the 4-D approximations, numerical round off errors have an influence on the accurracy of the coefficients. Consequently it is recommended, that the Gauss Jordan Elimination routine be revised to use double precision constants and a routine that conditions the system matrix be added. Time constraints made these steps impossible for the author.

Following these refinements, a calibration procedure, similar to McGuire's (ref 3) should be formalised. Since the 4-D- approximation requires large arrays and therefore extensive CP - memory, extended memory access (EMA) is necessary if the HP 21MX computer is used to perform the calculations.

LIST OF REFERENCES

1. McCracken, D. D. and Dorn, W. S.: Numerical Methods and FORTRAN Programming, Third Printing. New York: Wiley & Sons Inc., 1965. Page 262ff.

2. Simon, H.: Anwendung verschiedener Berechnungsverfahren zur Auslegung eines Überschallveridchler-Laufrades, und dessen experimentielle Untersuchung, Dissertation, RWTH Aachen, 1973, Page 134f.

   [Application of Various Calculation Methods for the Design of a Supersonic Rotor and Its Experimental Investigation, Dissertation, Technical University Aachen, 1973.]

3. McGuire, A. G.: Pneumatic Velocity Probe Calibration-Users Manual for Data Acquisition and Reduction. Monterey: Naval Postgraduate School, Turbopropulsion Laboratory, Technical Note 80-01, 1980.

APPENDIX A: Some Useful Matrix Conventions and Operations

A1. Submatrix Notation

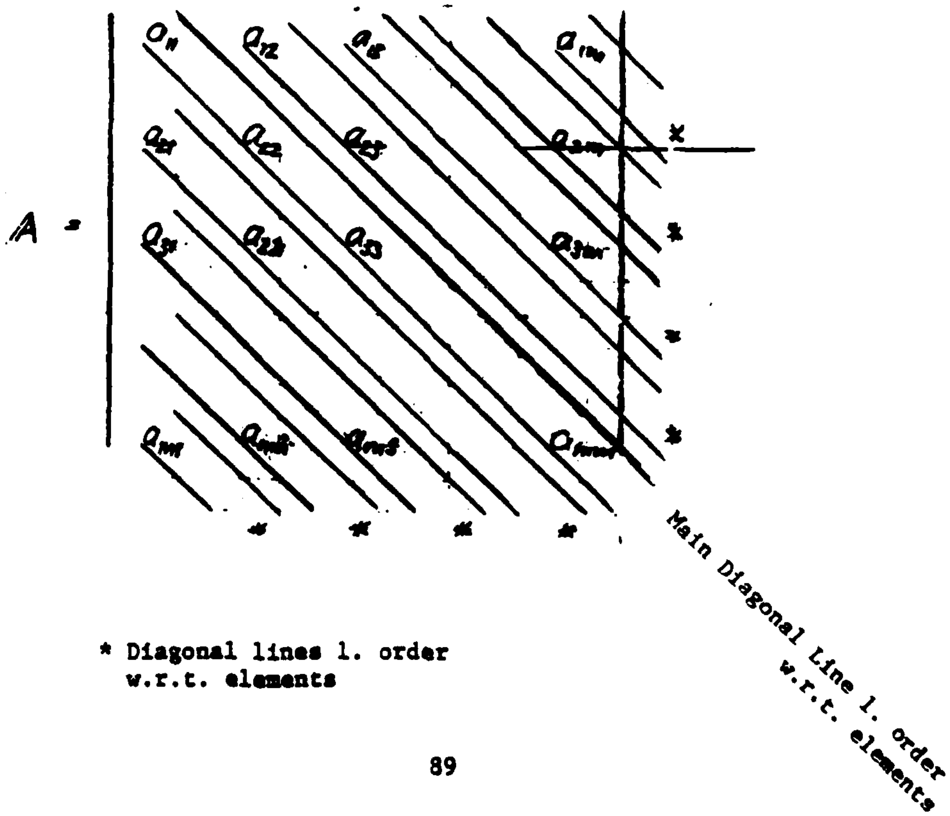
$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{1m} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{2m} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{3m} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{4m} \\ & & & & \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & a_{mm} \end{vmatrix} \quad \text{or} \quad \begin{vmatrix} A^*_{11} & A^*_{12} & \cdots & A^*_{1m} \\ A^*_{21} & A^*_{22} & \cdots & A^*_{2m} \\ A^*_{m1} & A^*_{m2} & \cdots & A^*_{mm} \end{vmatrix}$$

defined through elements a_{ij} or submatrices A^*_{ij}

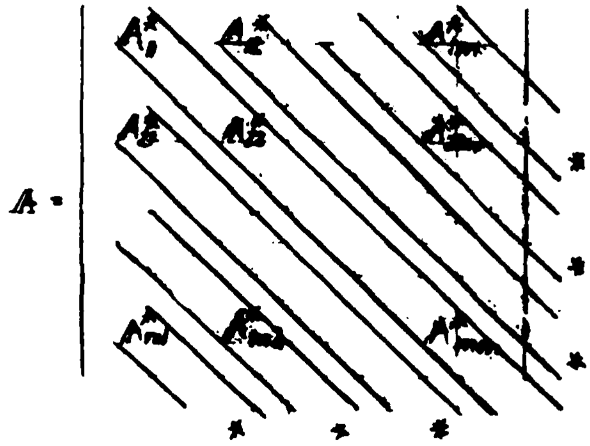
A2. Diagonal Lines and Diagonal Bands 1. order

Diagonal lines 1. order and Main diagonal line 1. order

Diagonal bands 1. order and Main diagonal band 1. order



* Diagonal lines 1. order
w.r.t. elements

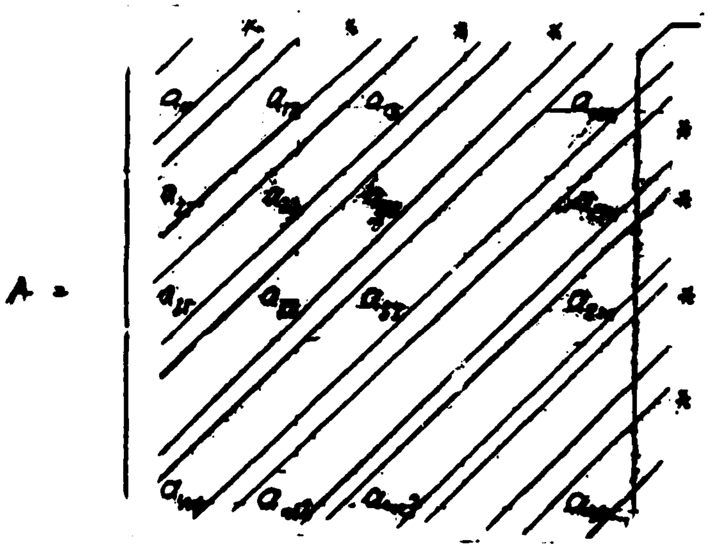


* Diagonal bands 1. order
w.r.t. submatrices

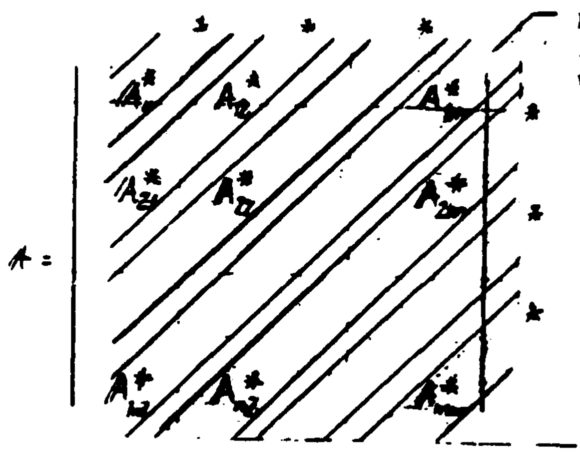
A3. <u>Diagonal Lines and Diagonal Bands 2. order</u>

Diagonal lines 2. order and Main diagonal line 2. order

Diagonal bands 2. order and Main diagonal band 2. order



* Diagonal lines 2. order w.r.t. elements

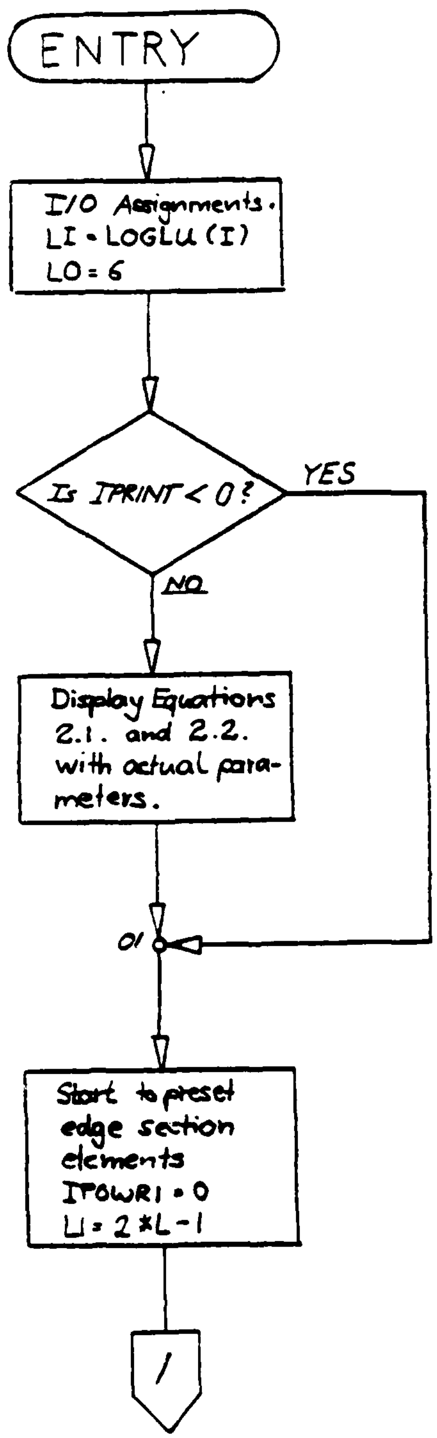


* Dia~~nal bands 2. order
w.r.t. submatrices

APPENDIX B: Software Description: Flow Charts

The following are given:

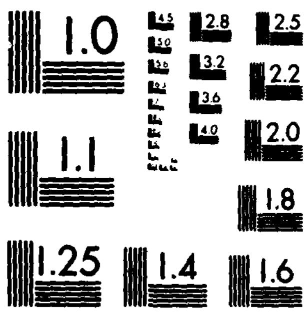
Flow chart MAT2

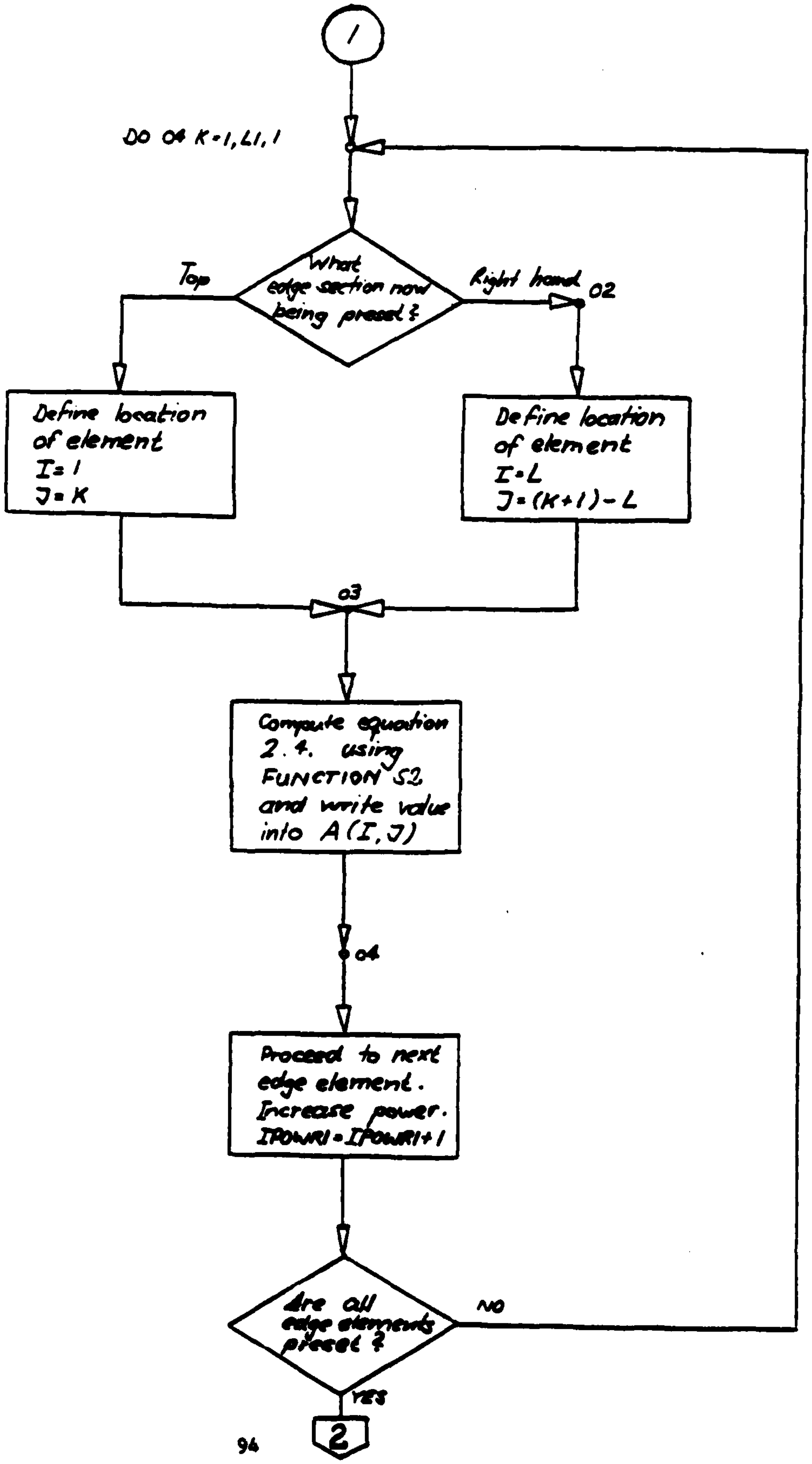


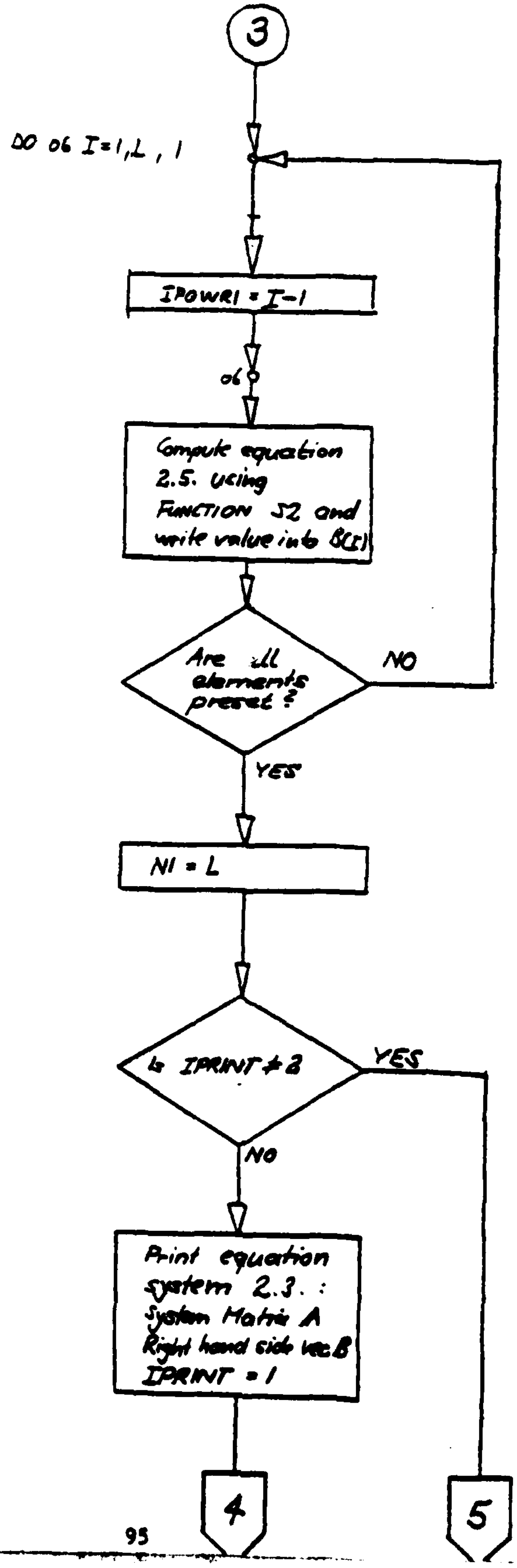
AD-A110 673   BDM CORP MONTEREY CA   F/G 20/4
PROCEDURE AND COMPUTER PROGRAM FOR THE APPROXIMATION OF DATA (W--ETC(U)
AUG 80 H ZEBNER   N00014-78-C-0204
UNCLASSIFIED   NPS-67-80-001CR   NL

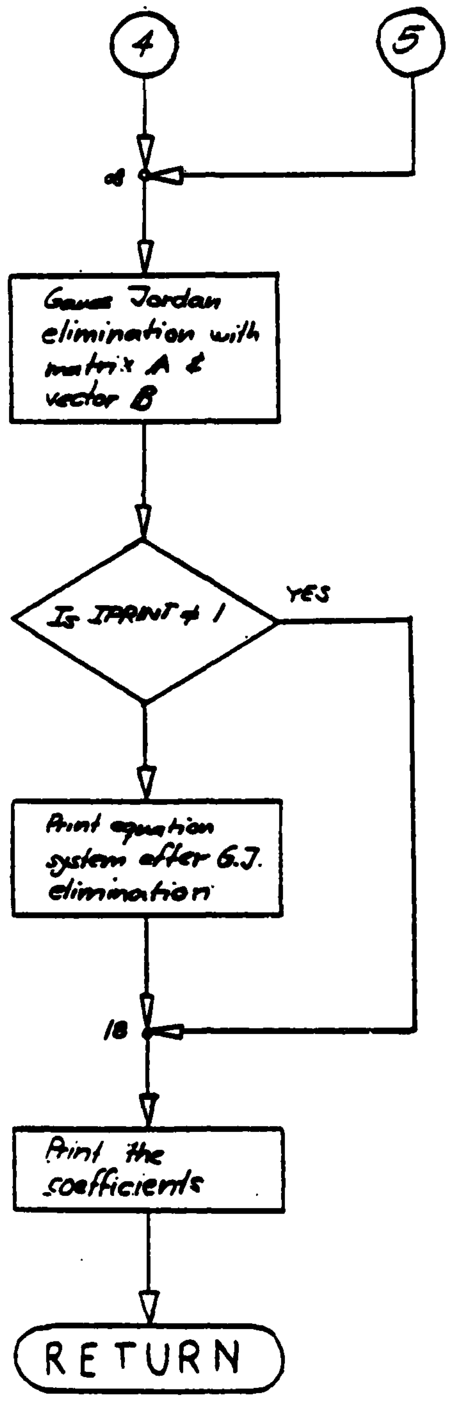
2 of 2

END
DATE FILMED
3 82
DTIC

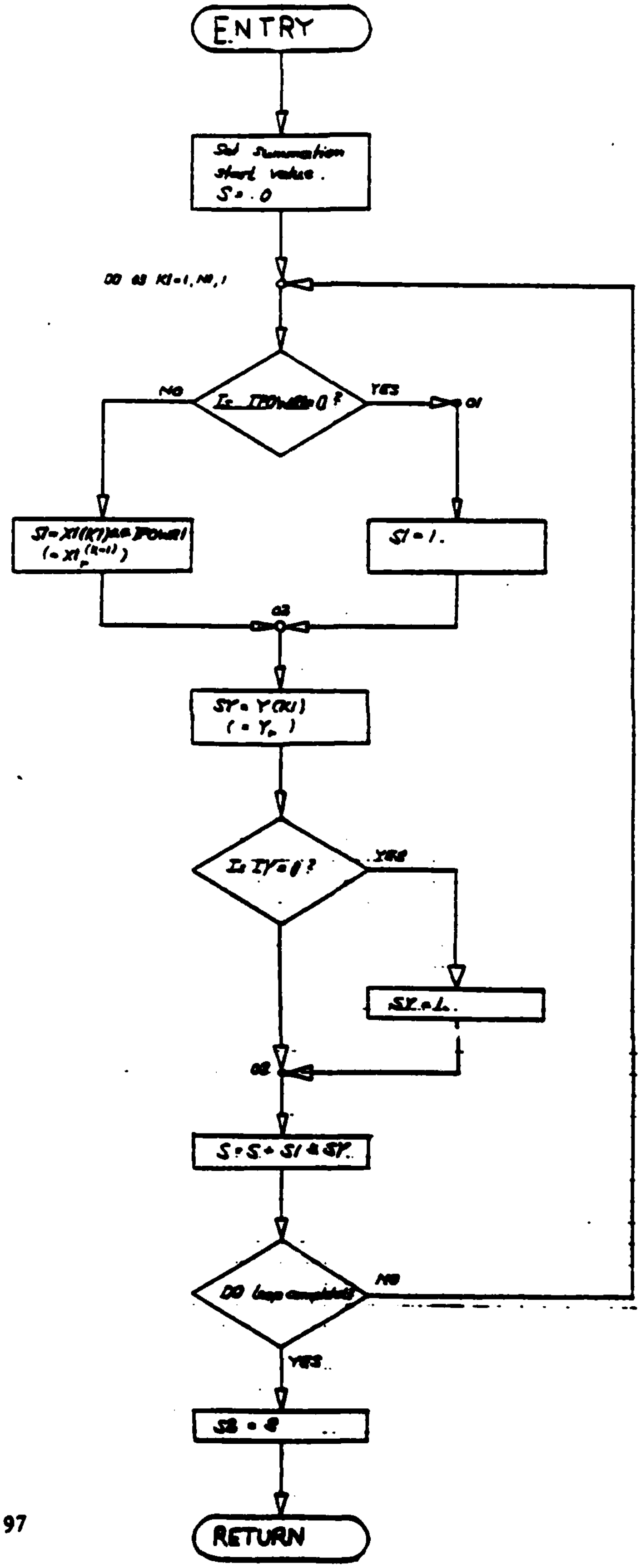

1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0


1
DO 04 K=1,L1,1
What edge section now being preset?
Top
Right hand
02
Define location of element
I=1
J=K
Define location of element
I=L
J=(K+1)-L
03
Compute equation 2.4. using FUNCTION S2 and write value into A(I,J)
04
Proceed to next edge element. Increase power. IPOWR1=IPOWR1+1
Are all edge elements preset?
NO
YES
2

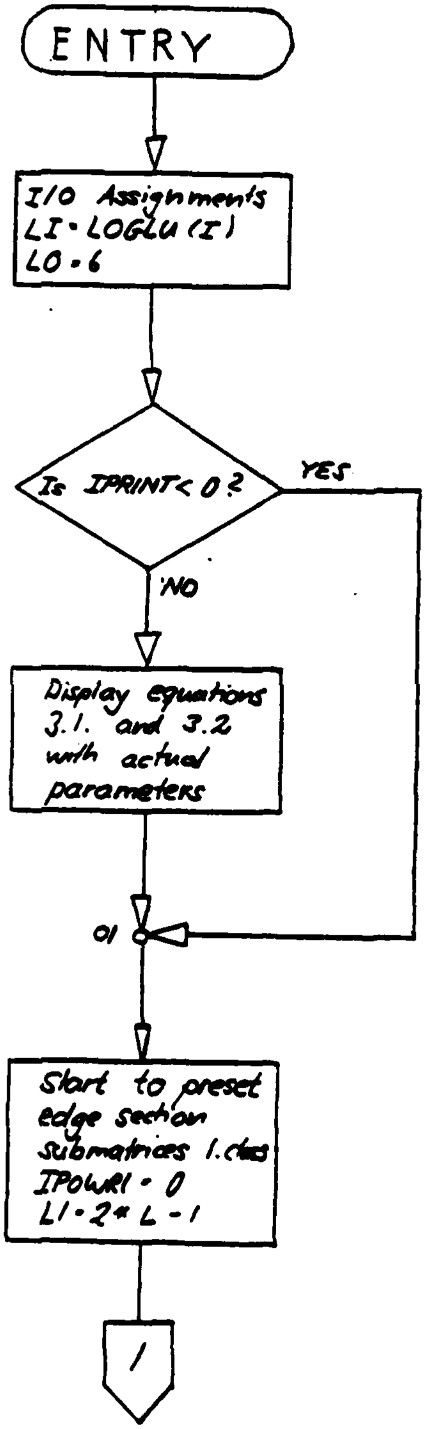

3
DO 06 I=1,L,1
IPOWR1 = I-1
06
Compute equation 2.5. using FUNCTION S2 and write value into B(I)
Are all elements preset?
NO
YES
N1 = L
Is IPRINT ≠ 2
YES
NO
Print equation system 2.3.: System Matrix A Right hand side vec B IPRINT = 1
4
5

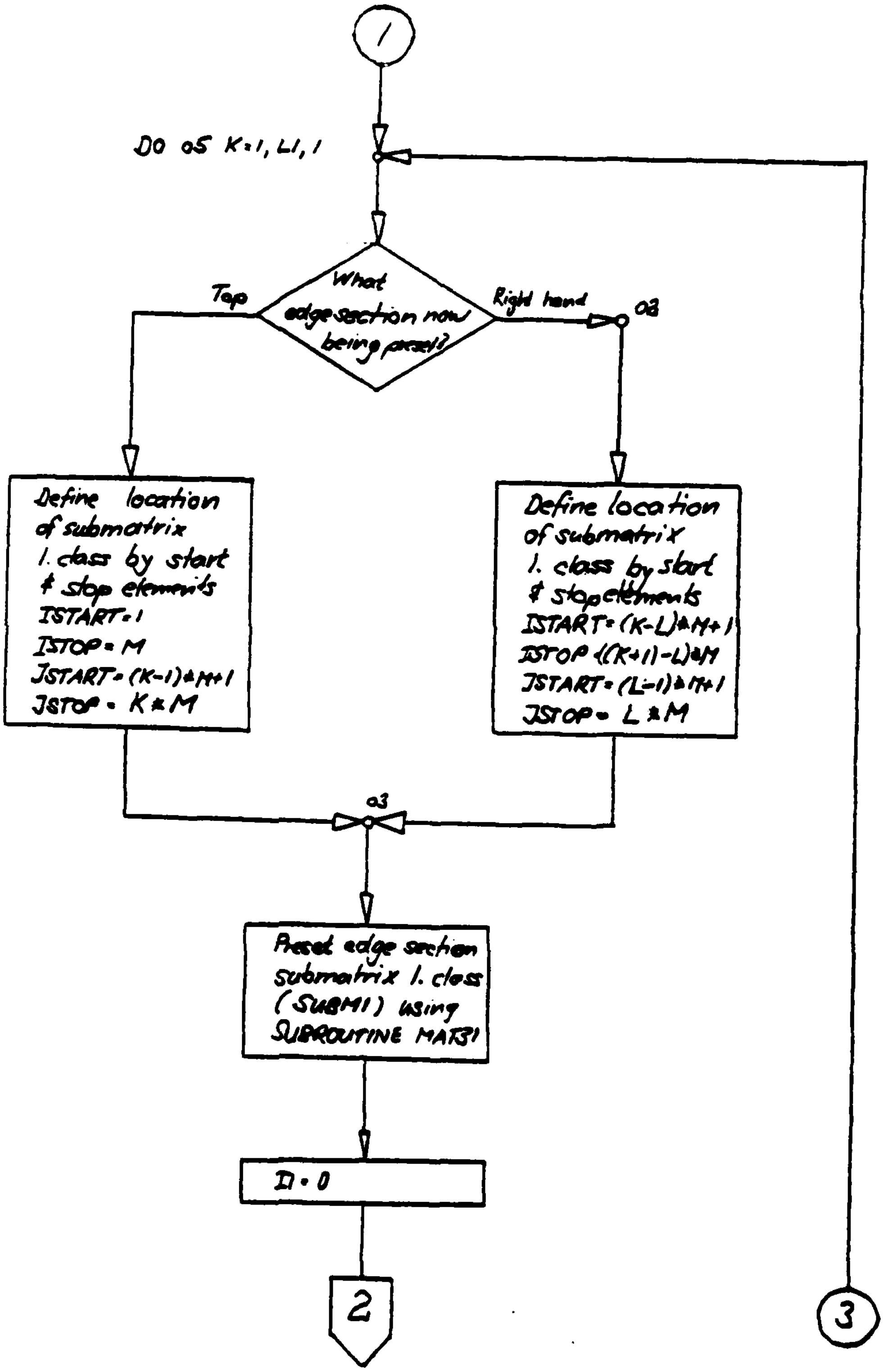

4
5
08
Gauss Jordan elimination with matrix A & vector B
Is IPRINT ≠ 1
YES
Print equation system after G.J. elimination
18
Print the coefficients
RETURN

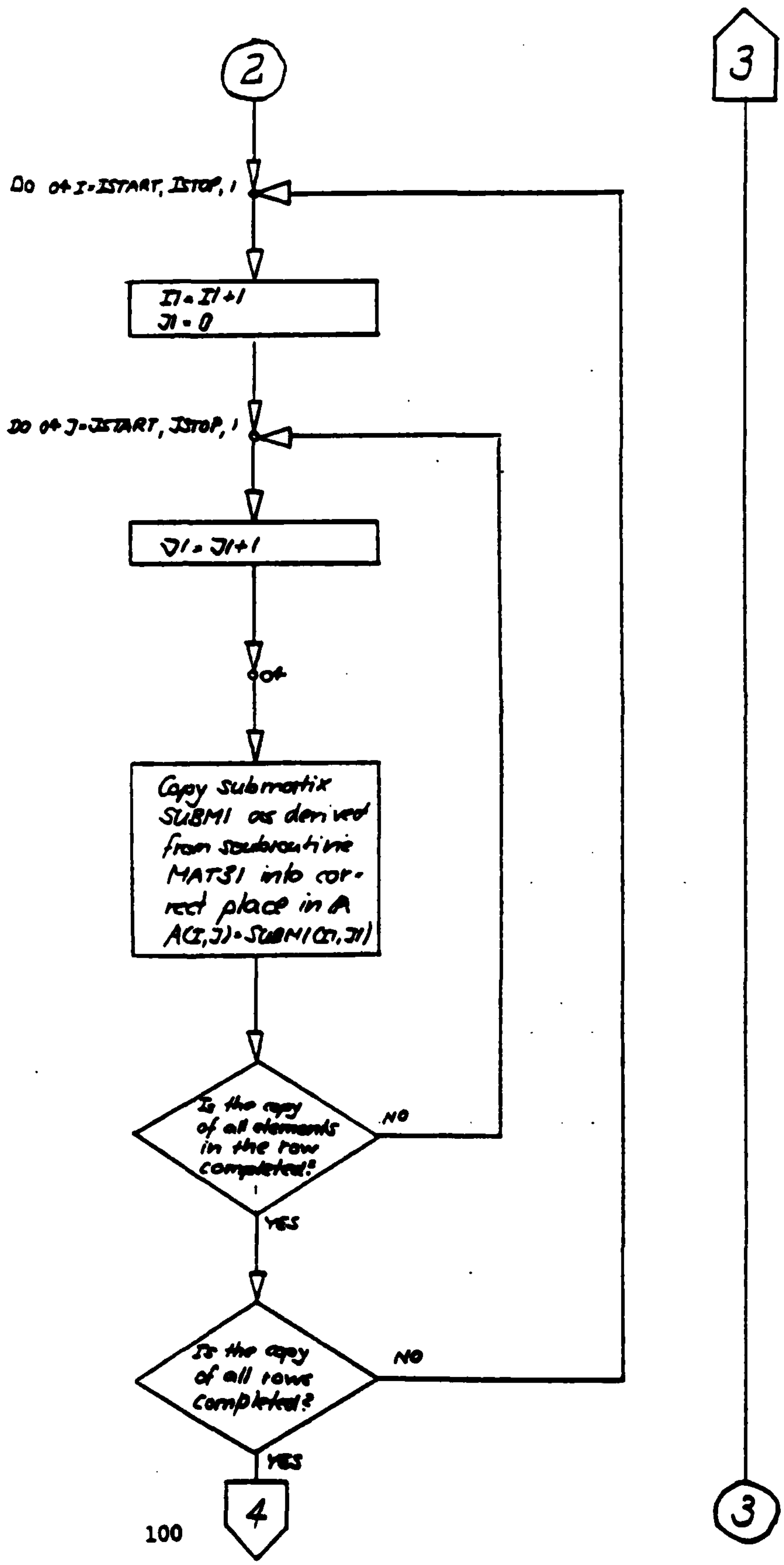
Flow chart S2

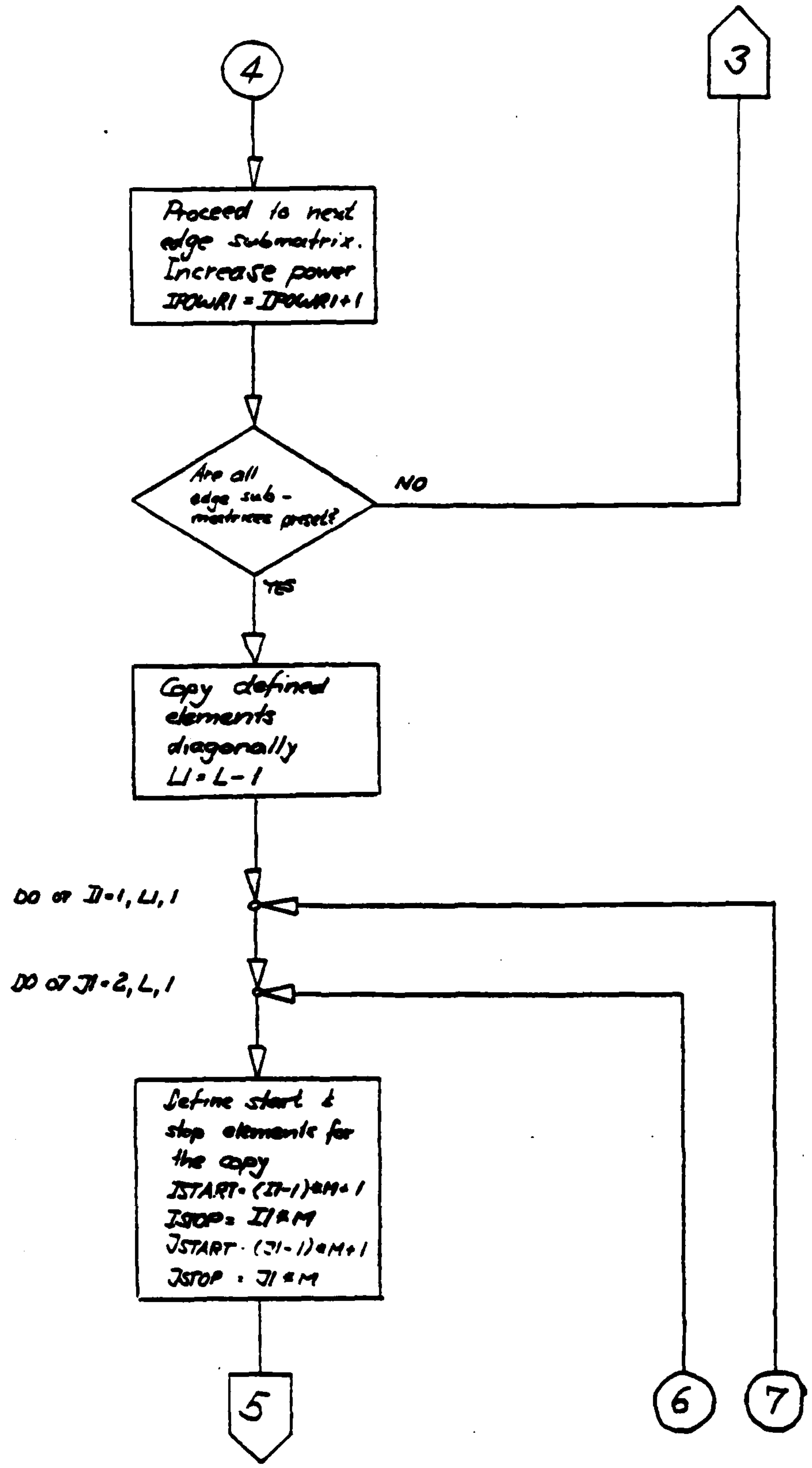


Flow chart MAT3

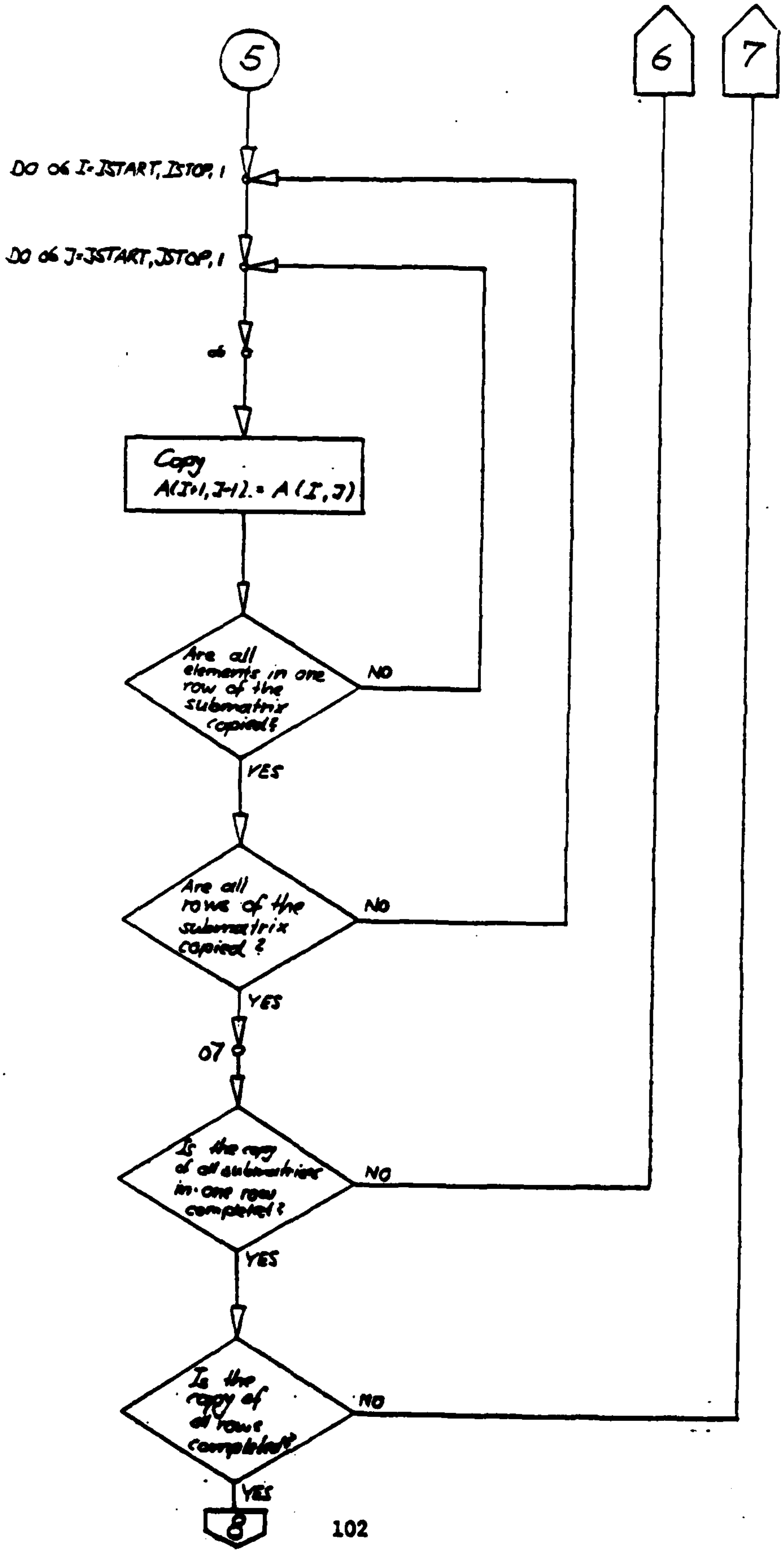


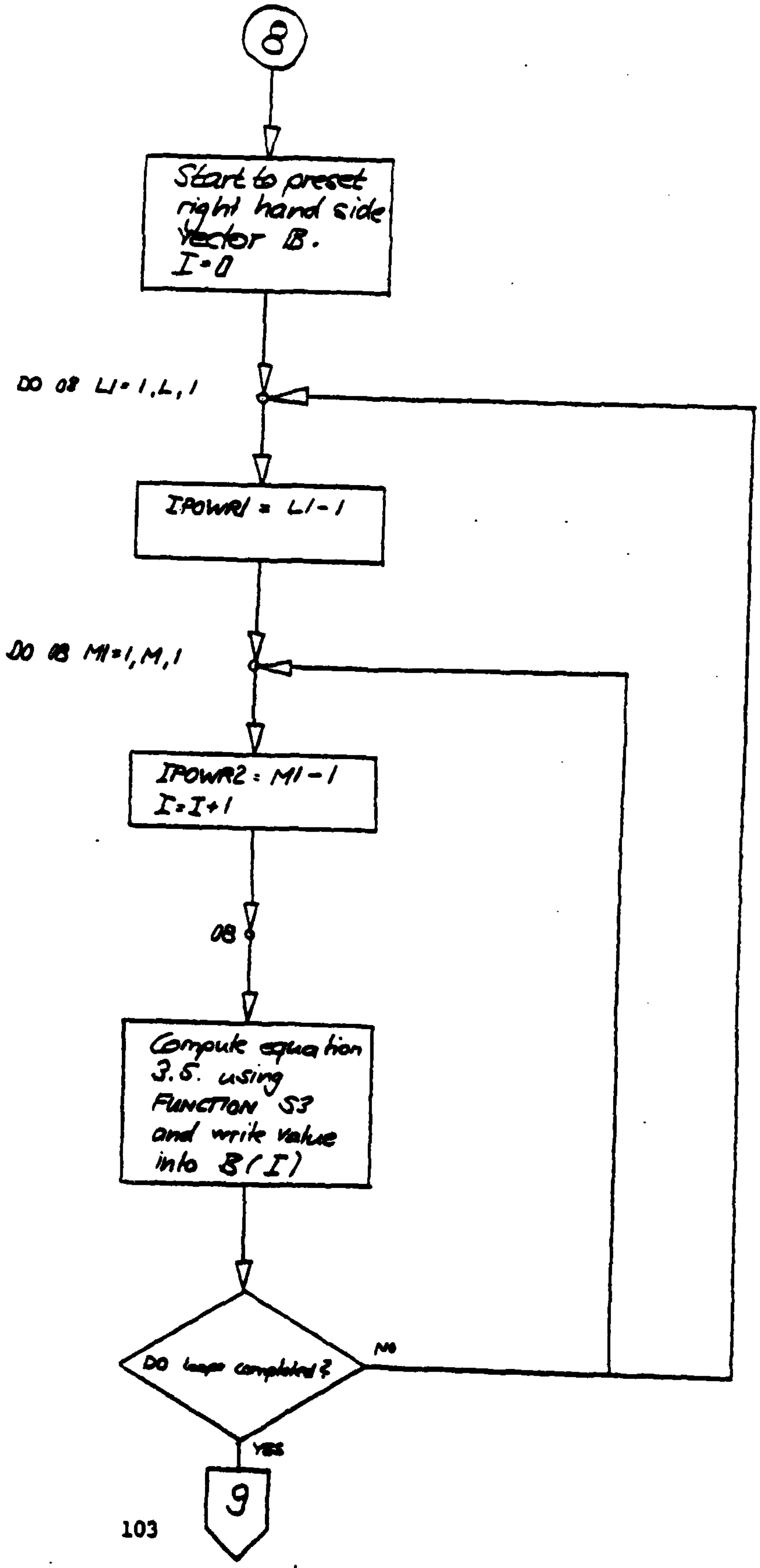

1
DO 05 K=1, L1, 1
What edge section now being preset?
Top
Right hand
02
Define location of submatrix 1. class by start & stop elements
ISTART=1
ISTOP= M
JSTART=(K-1)*M+1
JSTOP= K*M
Define location of submatrix 1. class by start & stop elements
ISTART=(K-L)*M+1
ISTOP=((K+1)-L)*M
JSTART=(L-1)*M+1
JSTOP= L*M
03
Preset edge section submatrix 1. class (SUBM1) using SUBROUTINE MAT31
II=0
2
3

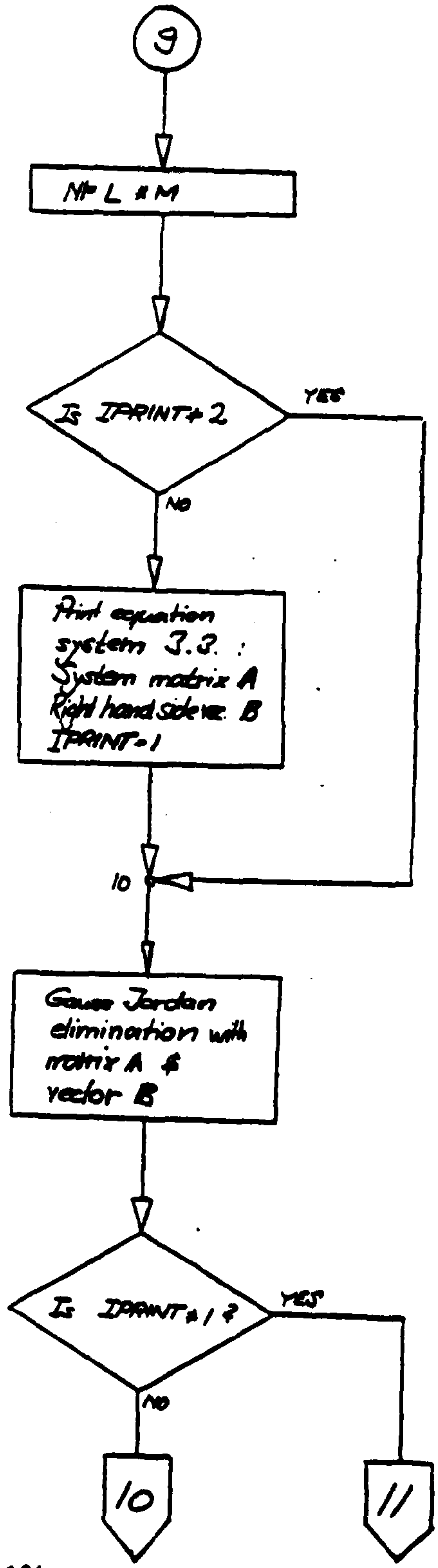

2
3
DO 04 I=ISTART, ISTOP, 1
I1 = I1+1
J1 = 0
DO 04 J=JSTART, JSTOP, 1
J1 = J1+1
04
Copy submatrix SUBM1 as derived from subroutine MAT31 into correct place in A
A(I,J)=SUBM1(I1,J1)
Is the copy of all elements in the row completed?
NO
YES
Is the copy of all rows completed?
NO
YES
4
3



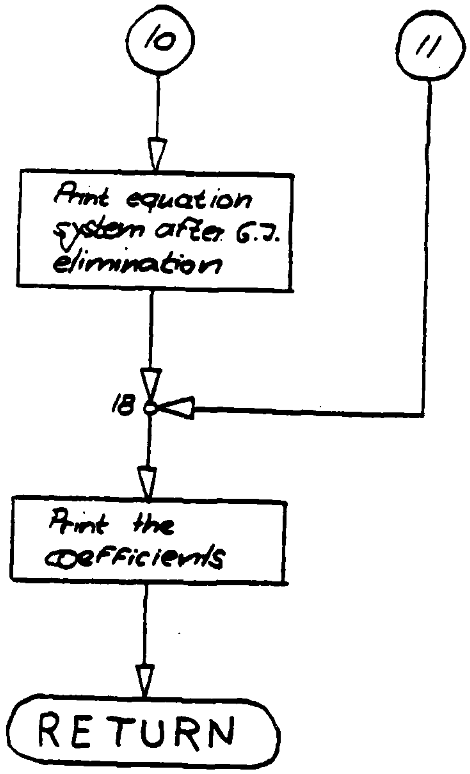
4
3
Proceed to next edge submatrix. Increase power IPOWR1 = IPOWR1+1
Are all edge sub-matrices preset?
NO
YES
Copy defined elements diagonally L1 = L-1
DO 07 I1=1, L1, 1
DO 07 J1=2, L, 1
Define start & stop elements for the copy ISTART = (I1-1)*M+1 ISTOP = I1*M JSTART = (J1-1)*M+1 JSTOP = J1*M
5
6
7


5
6
7
DO 06 I=ISTART,ISTOP,1
DO 06 J=JSTART,JSTOP,1
06
Copy
A(I+1,J+1) = A(I,J)
Are all elements in one row of the submatrix copied?
NO
YES
Are all rows of the submatrix copied?
NO
YES
07
Is the copy of all submatrices in one row completed?
NO
YES
Is the copy of all rows completed?
NO
YES
8

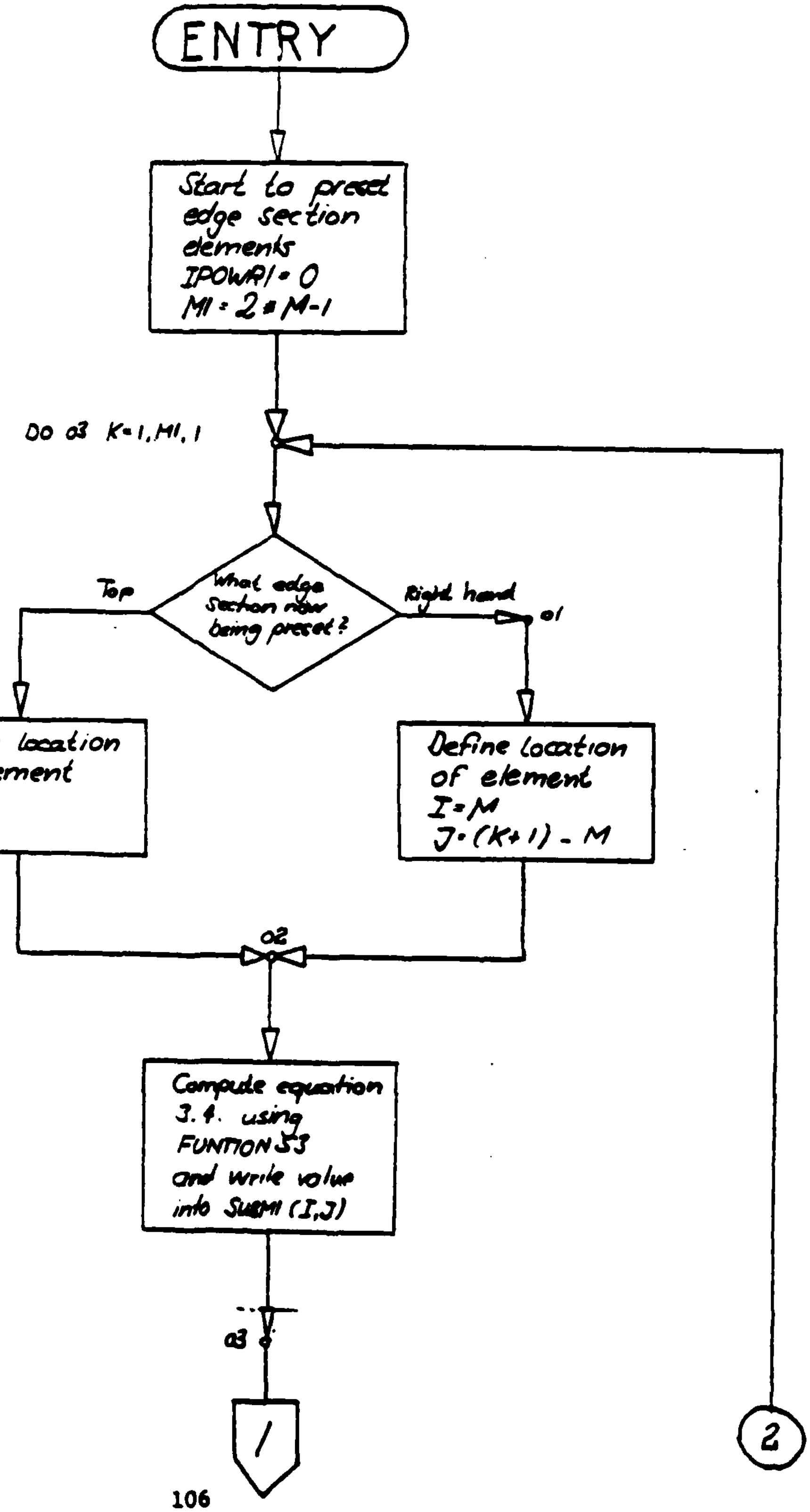

8
Start to preset right hand side Vector B.
I=0
DO 08 L1=1,L,1
IPOWR1 = L1-1
DO 08 M1=1,M,1
IPOWR2 = M1-1
I=I+1
08
Compute equation 3.5. using FUNCTION S3 and write value into B(I)
DO loops completed?
NO
YES
9

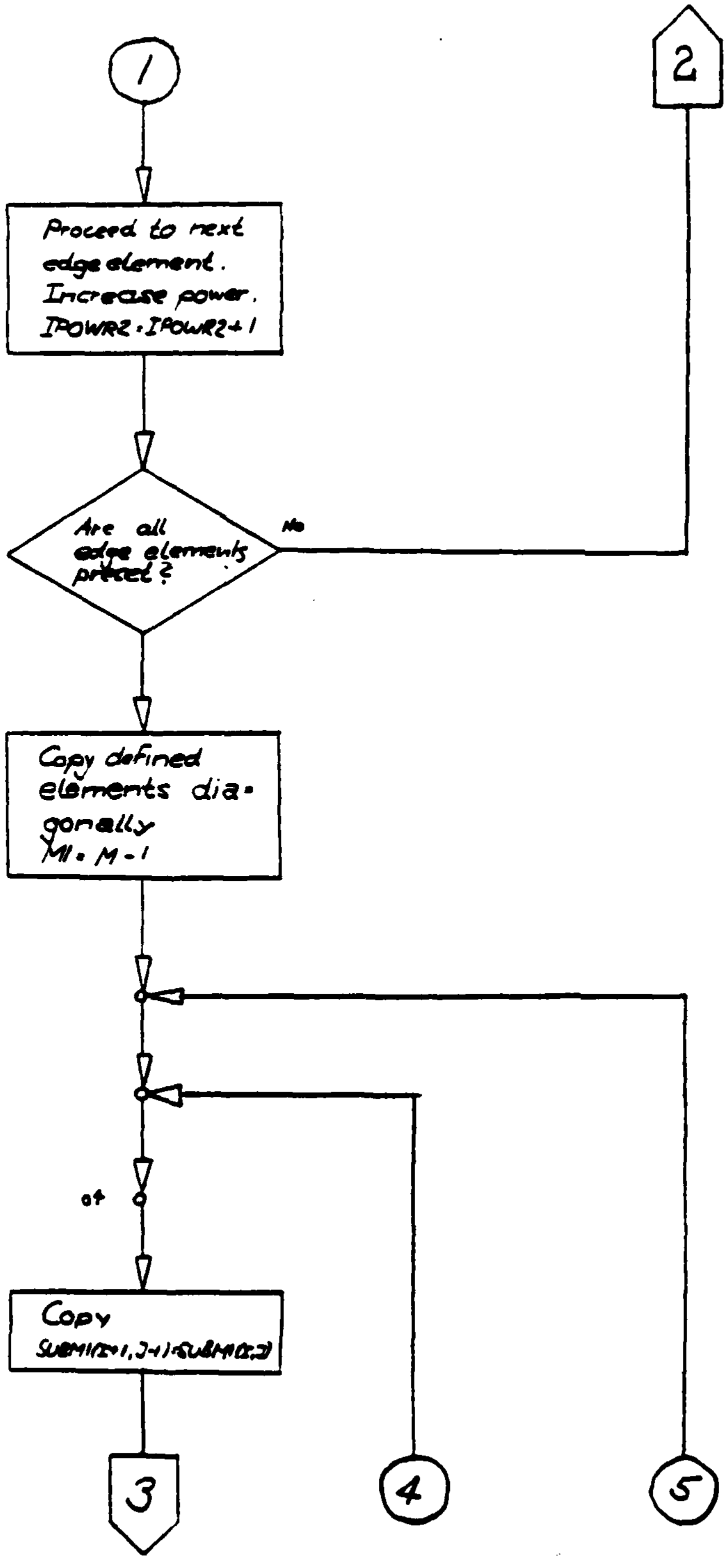


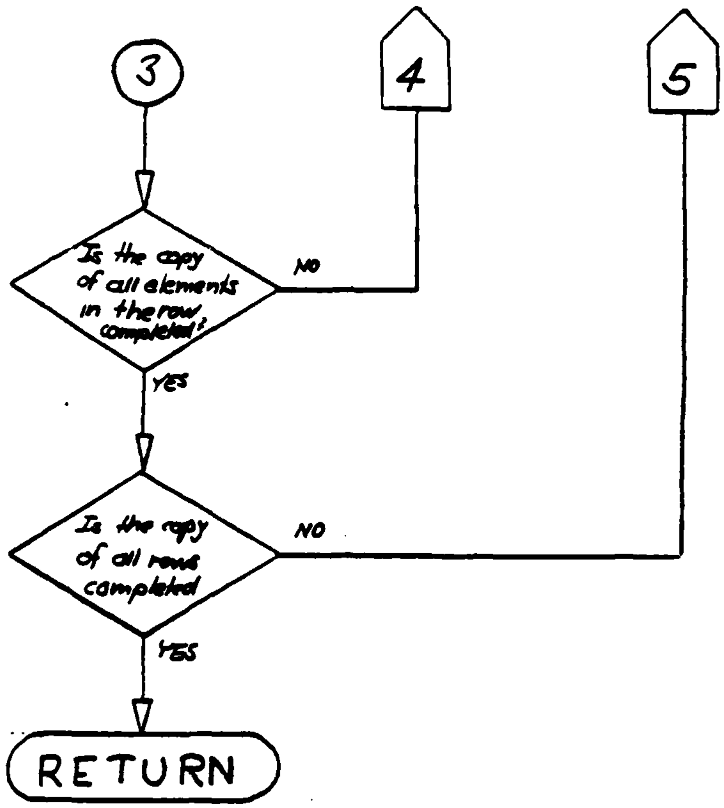
9
N≠L≠M
Is IPRINT≠2
YES
NO
Print equation system 3.3 :
System matrix A
Right hand side vec B
IPRINT=1
10
Gauss Jordan elimination with matrix A &
vector B
Is IPRINT≠1 ?
YES
NO
10
11


10
11
Print equation system after G.J. elimination
18
Print the coefficients
RETURN

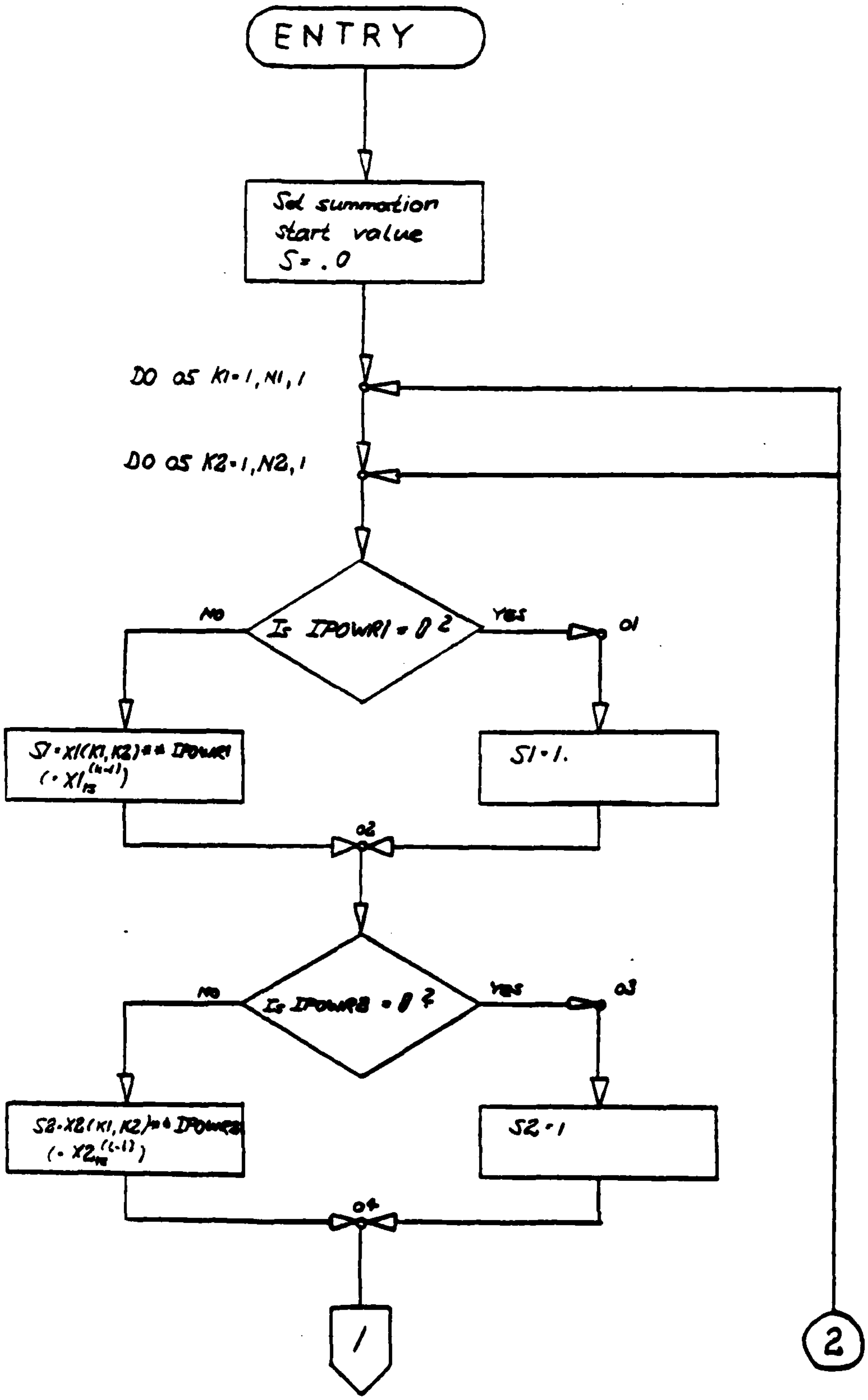
Flow chart MAT31

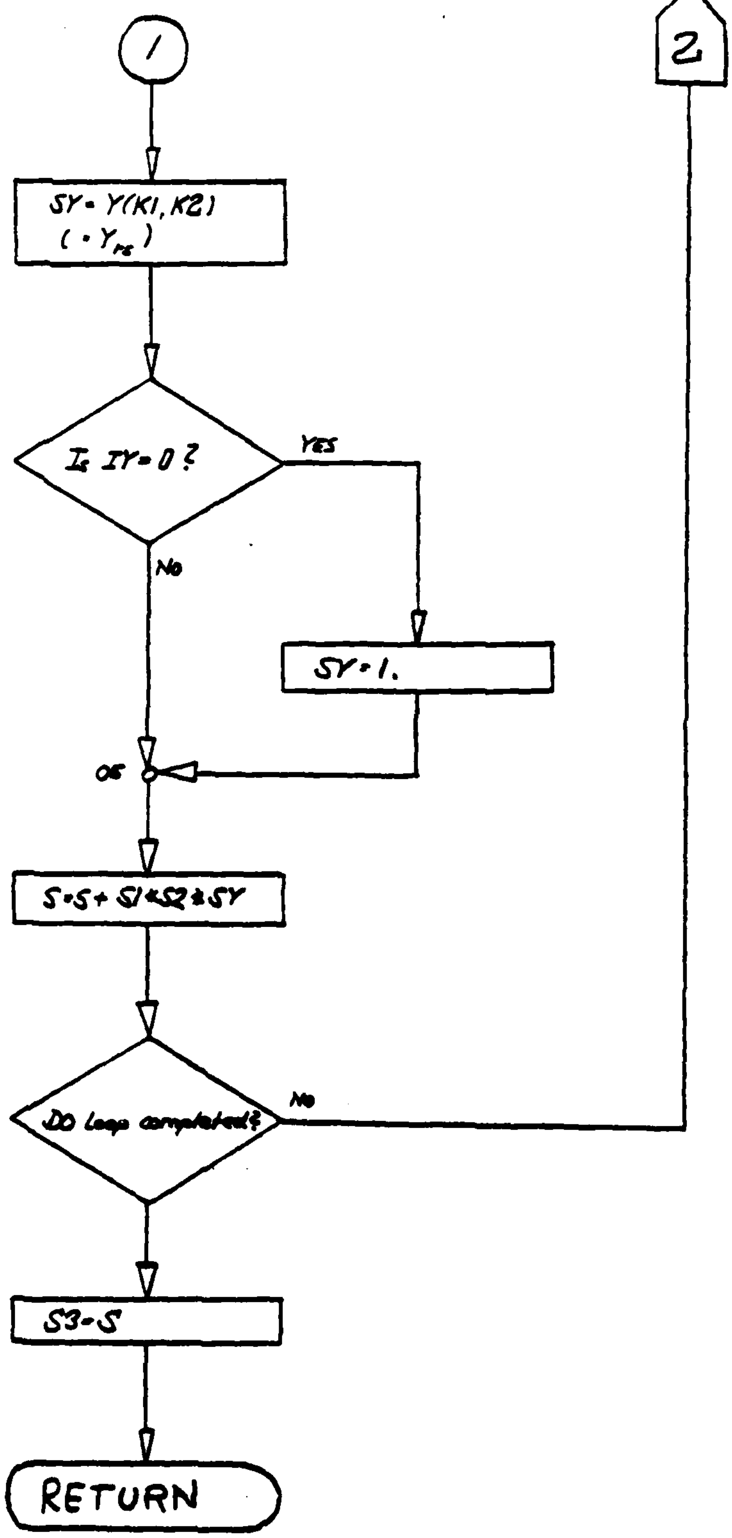



1
2
Proceed to next edge element. Increase power. IPOWR2 = IPOWR2 + 1
Are all edge elements proced?
No
Copy defined elements diagonally M1 = M - 1
Copy SUBM1(I+1, J+1) = SUBM1(I,J)
3
4
5

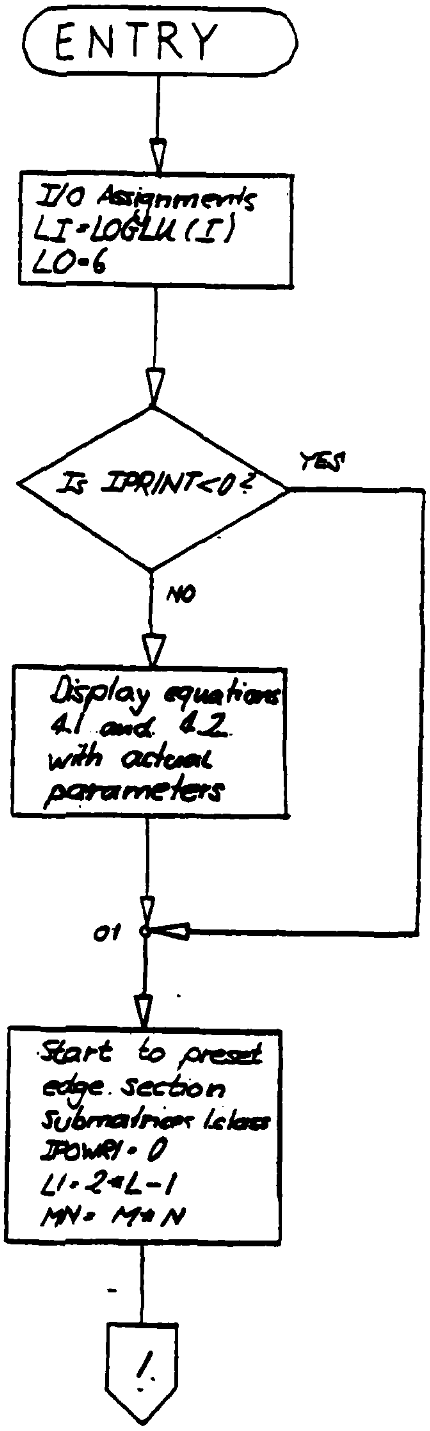

3
4
5
Is the copy of all elements in the row completed?
NO
YES
Is the copy of all rows completed
NO
YES
RETURN

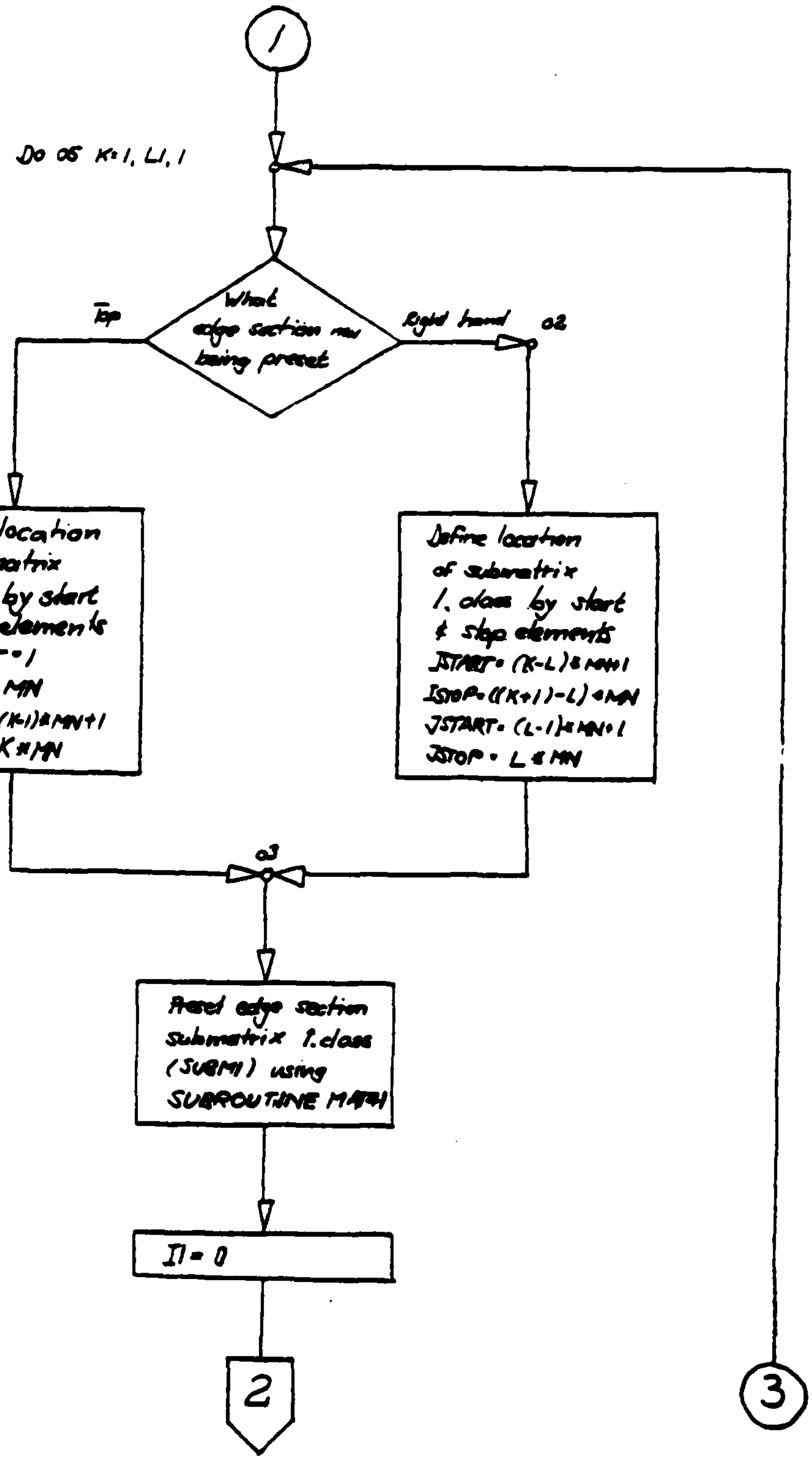
Flow chart S3

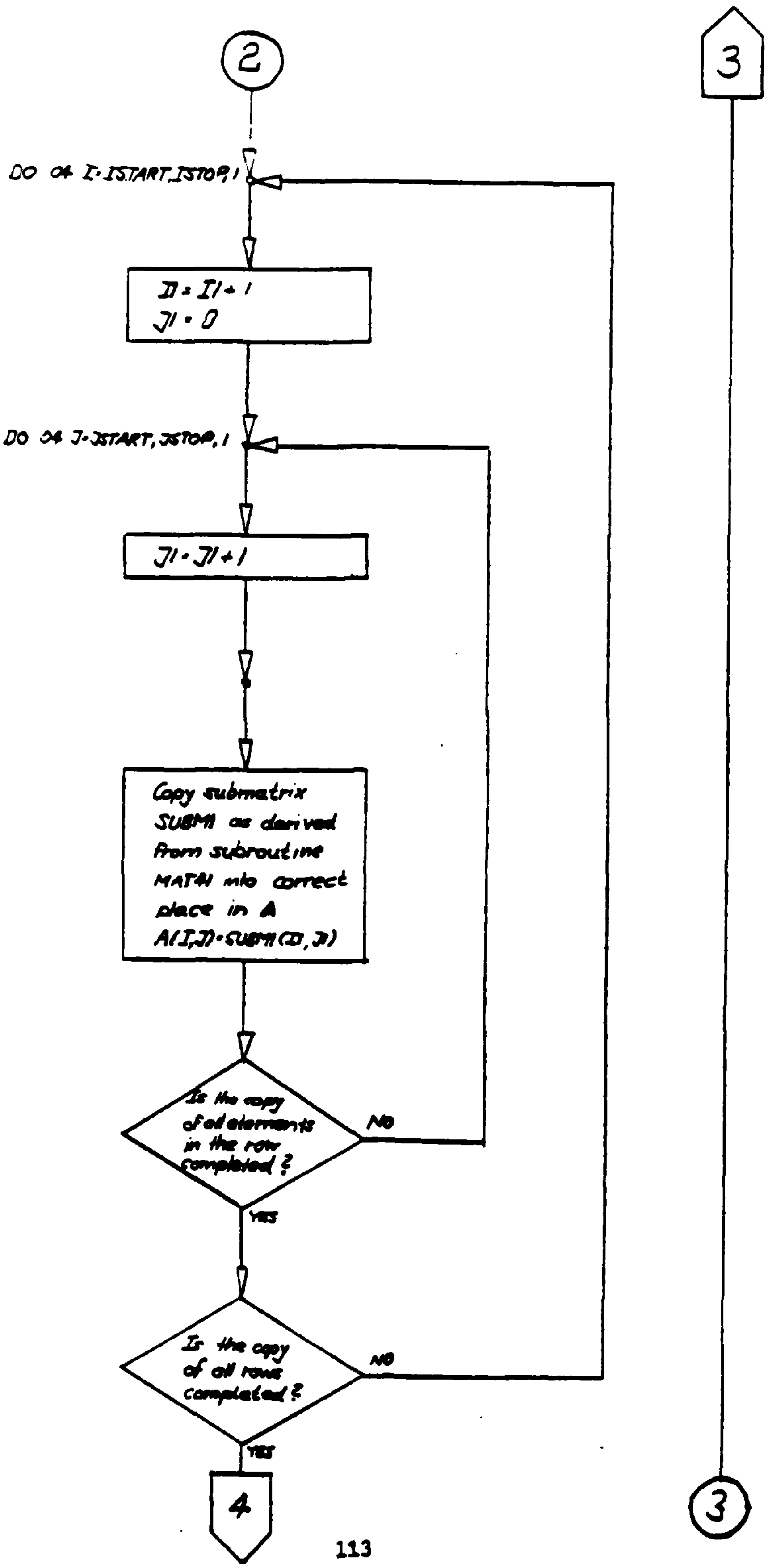



1
2
SY = Y(K1, K2)
(= Y_{rs})
Is IY = 0 ?
YES
No
SY = 1.
OR
S = S + S1*S2*SY
DO Loop completed?
No
S3 = S
RETURN

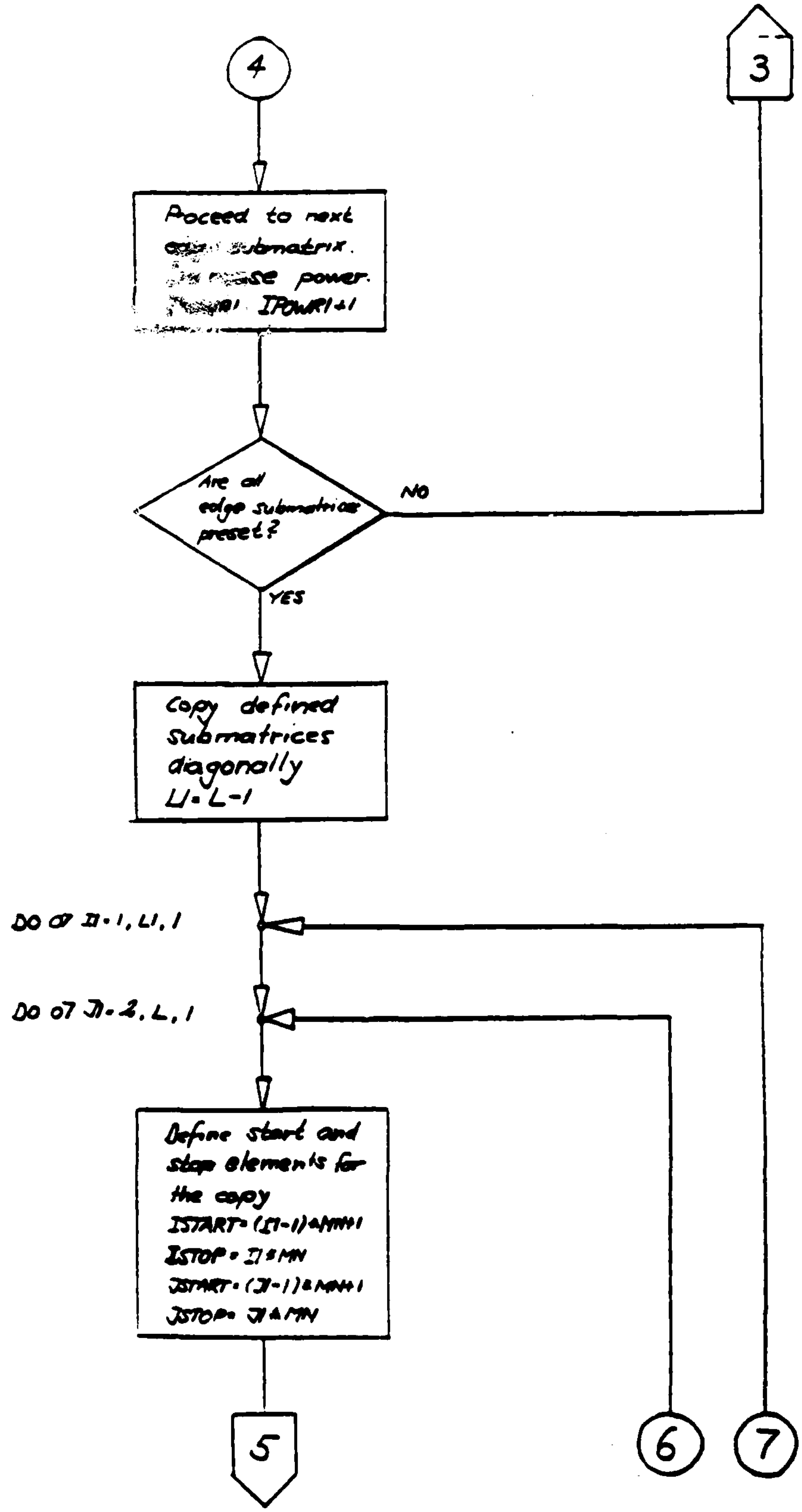
Flow chart MAT4

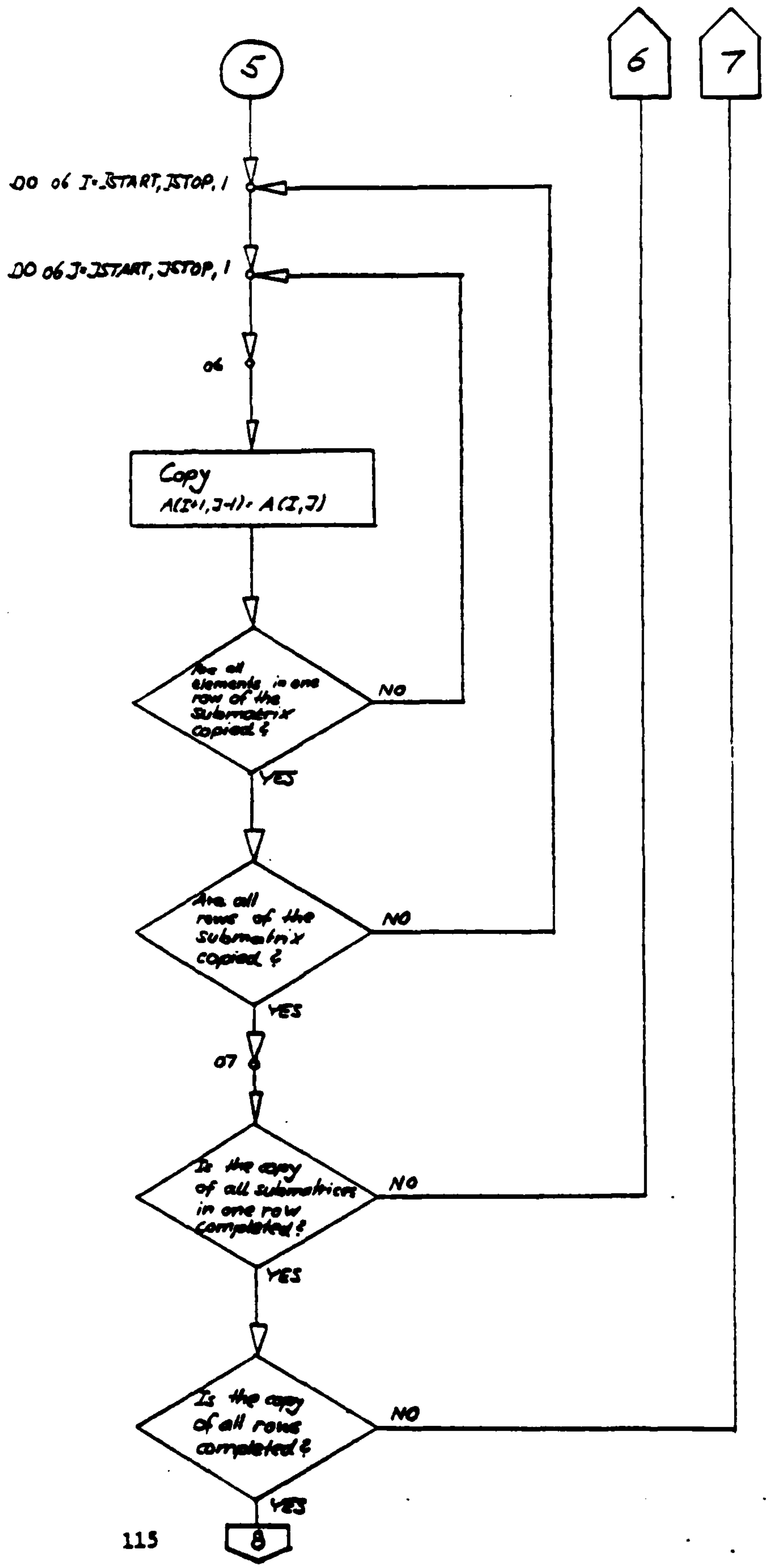


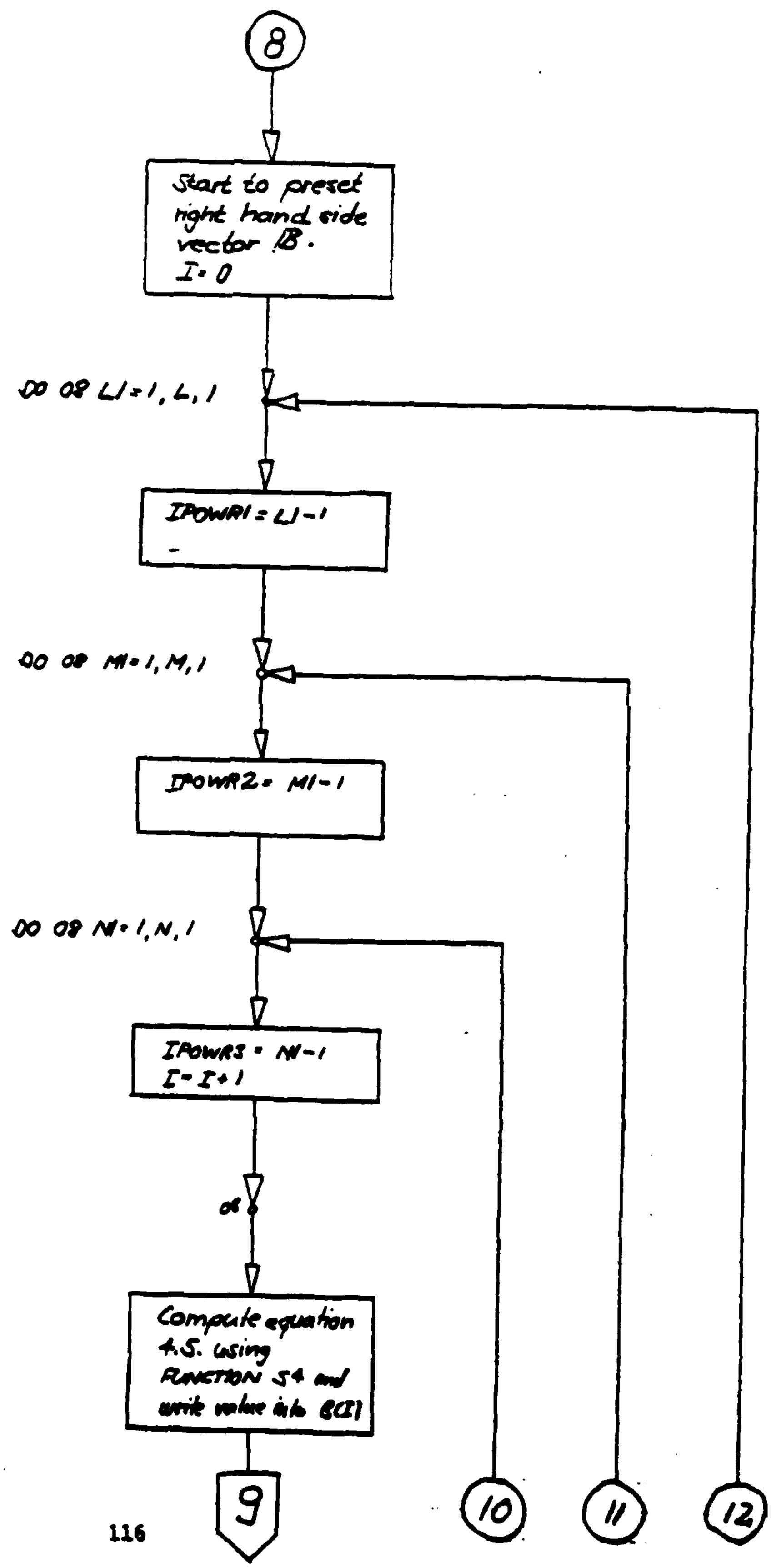

1
Do 05 K=1, L1, 1
What edge section now being preset
Top
Right hand
02
Define location of submatrix 1 class by start & stop elements
ISTART = 1
ISTOP = MN
JSTART = (K-1)*MN+1
JSTOP = K*MN
Define location of submatrix 1. class by start & stop elements
ISTART = (K-L)*MN+1
ISTOP = ((K+1)-L)*MN
JSTART = (L-1)*MN+1
JSTOP = L*MN
03
Preset edge section submatrix 1. class (SUBM1) using SUBROUTINE MATH
I1 = 0
2
3

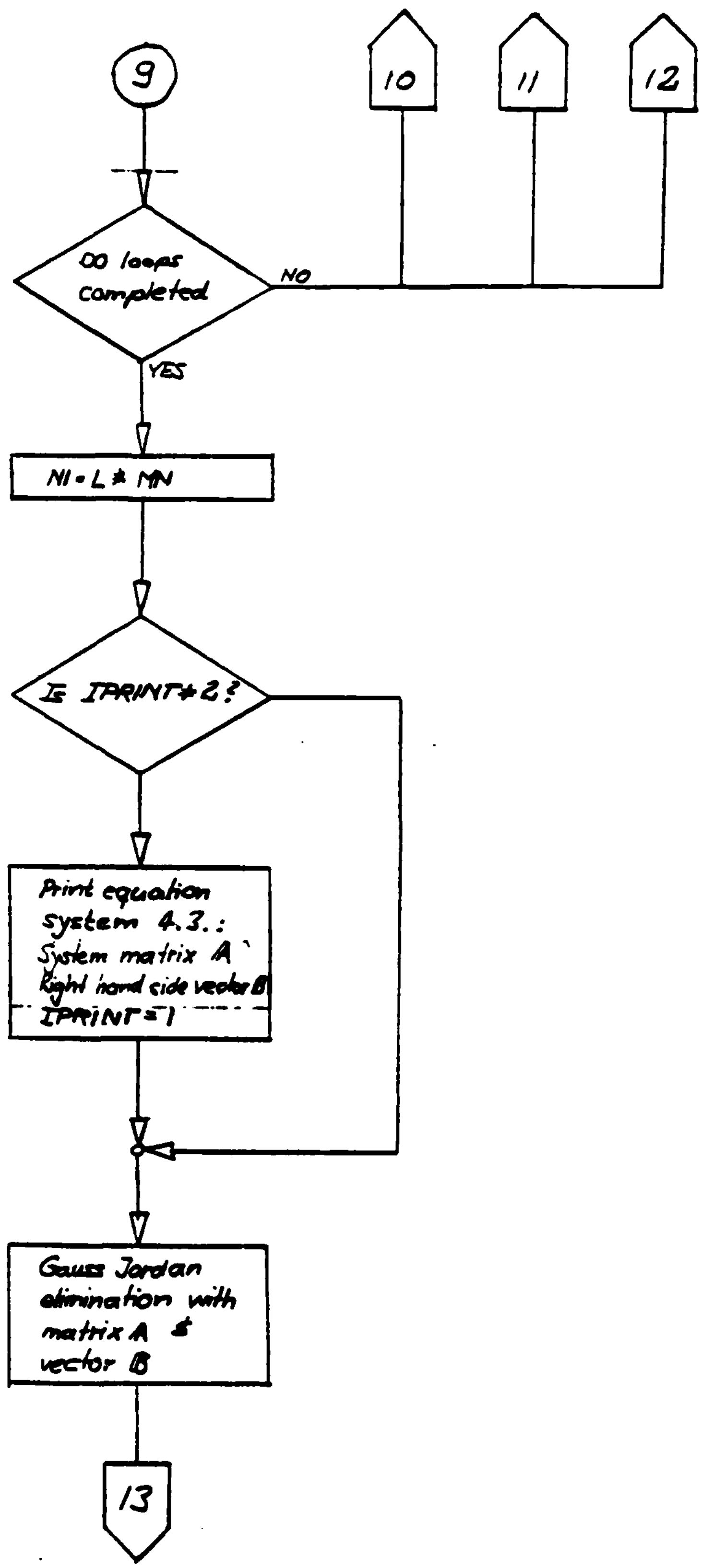


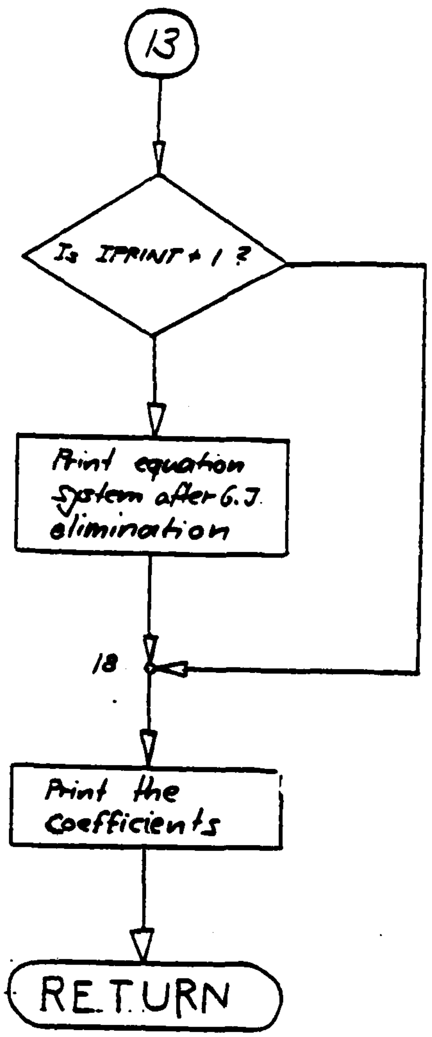
2
3
DO 04 I=ISTART,ISTOP,1
II = I1+1
J1 = 0
DO 04 J=JSTART,JSTOP,1
J1 = J1+1
Copy submatrix SUBM1 as derived from subroutine MATH into correct place in A
A(I,J)=SUBM1(II,J1)
Is the copy of all elements in the row completed?
NO
YES
Is the copy of all rows completed?
NO
YES
4
3


4
3
Poceed to next edge submatrix. Increase power. IPOWR1= IPOWR1+1
Are all edge submatrices preset?
NO
YES
Copy defined submatrices diagonally LI=L-1
DO 07 II=1,LI,1
DO 07 JI=2,L,1
Define start and stop elements for the copy
ISTART=(II-1)*MN+1
ISTOP=II*MN
JSTART=(JI-1)*MN+1
JSTOP=JI*MN
5
6
7

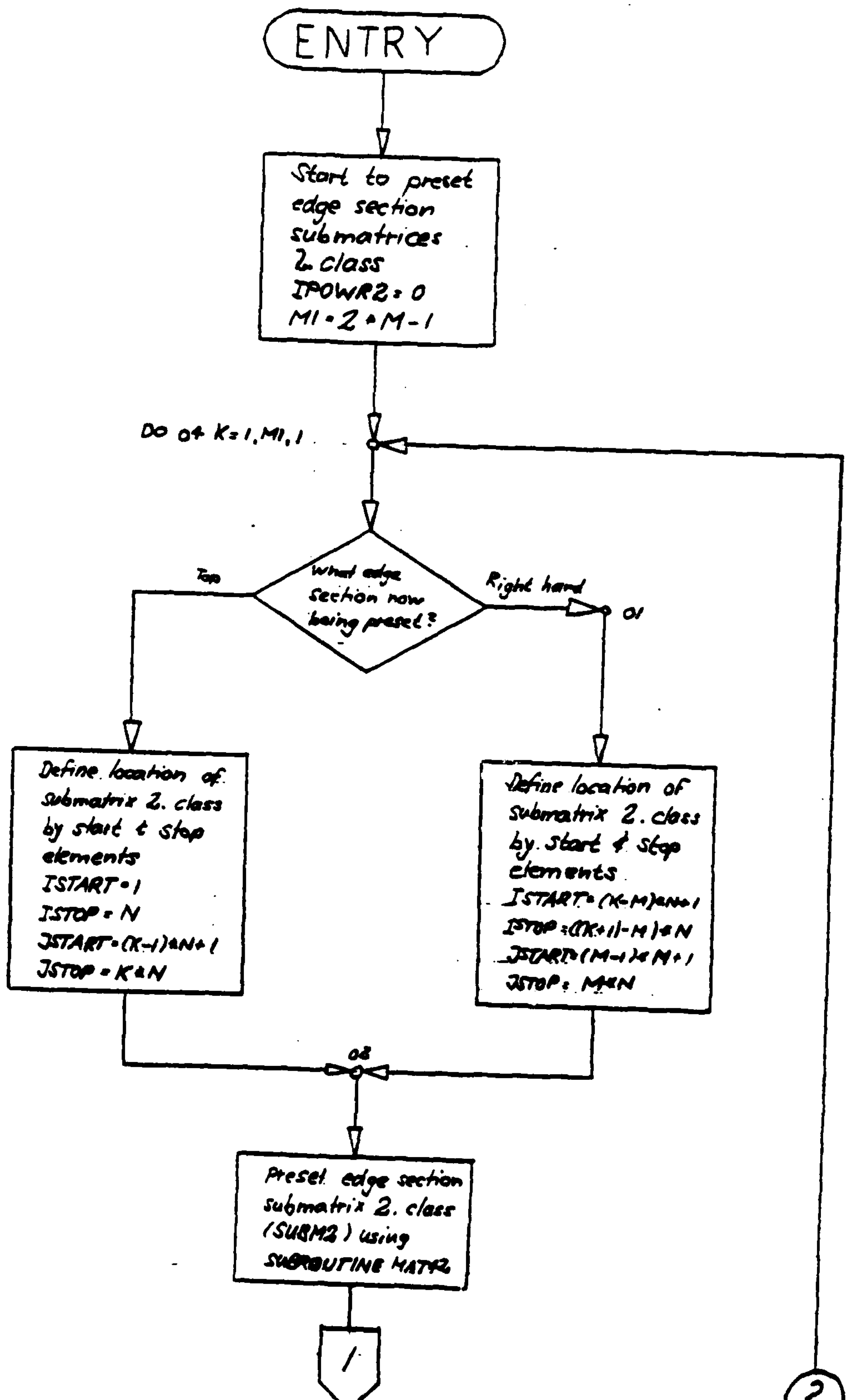

5
6
7
DO 06 I=ISTART,ISTOP,1
DO 06 J=JSTART,JSTOP,1
06
Copy
A(I+1,J+1)=A(I,J)
Are all elements in one row of the submatrix copied?
NO
YES
Are all rows of the submatrix copied?
NO
YES
07
Is the copy of all submatrices in one row completed?
NO
YES
Is the copy of all rows completed?
NO
YES
8

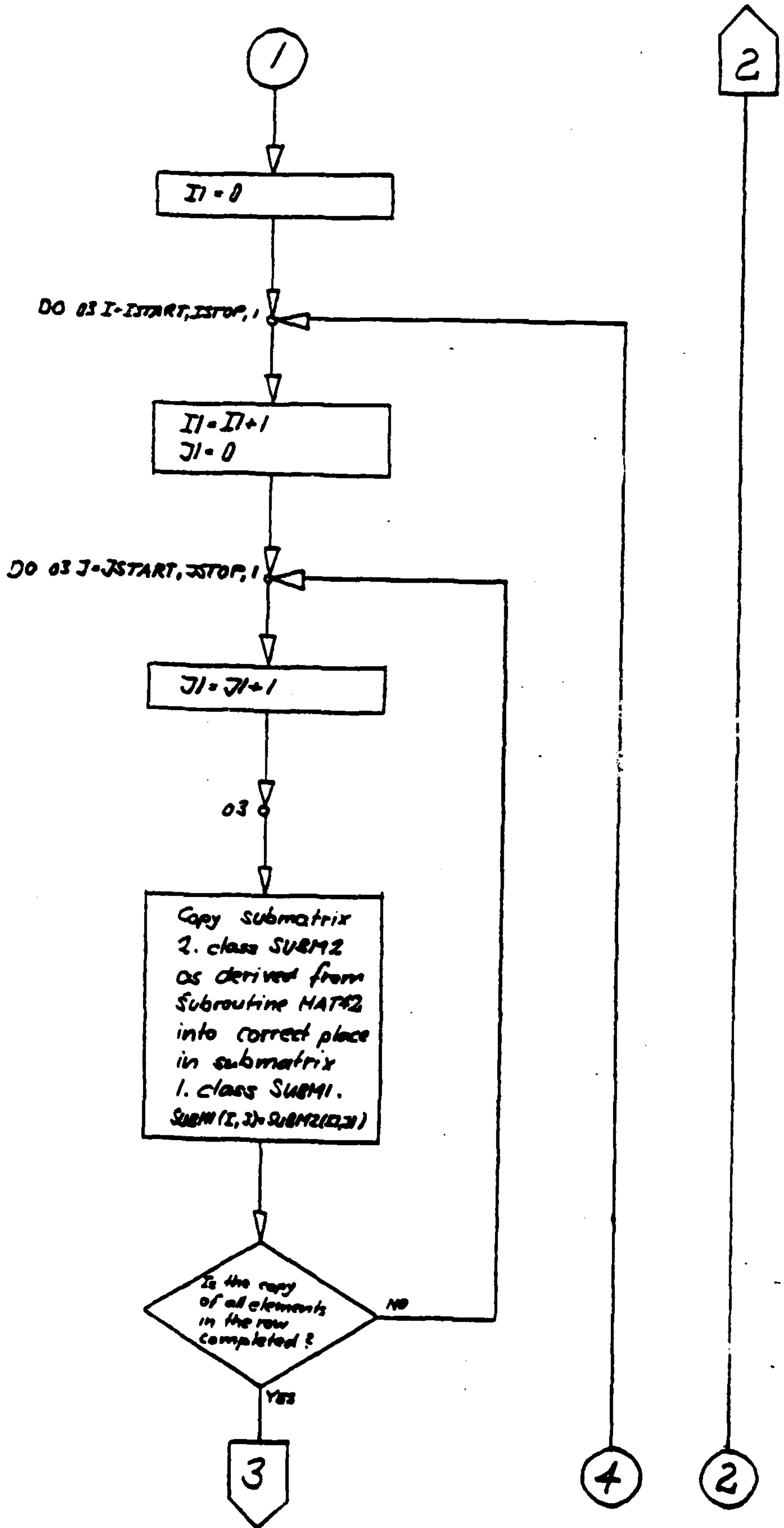

8
Start to preset right hand side vector B.
I = 0
DO 08 L1 = 1, L, 1
IPOWR1 = L1 - 1
DO 08 M1 = 1, M, 1
IPOWR2 = M1 - 1
DO 08 N1 = 1, N, 1
IPOWR3 = N1 - 1
I = I + 1
08
Compute equation 4.5. using FUNCTION S4 and write value into B(I)
9
10
11
12

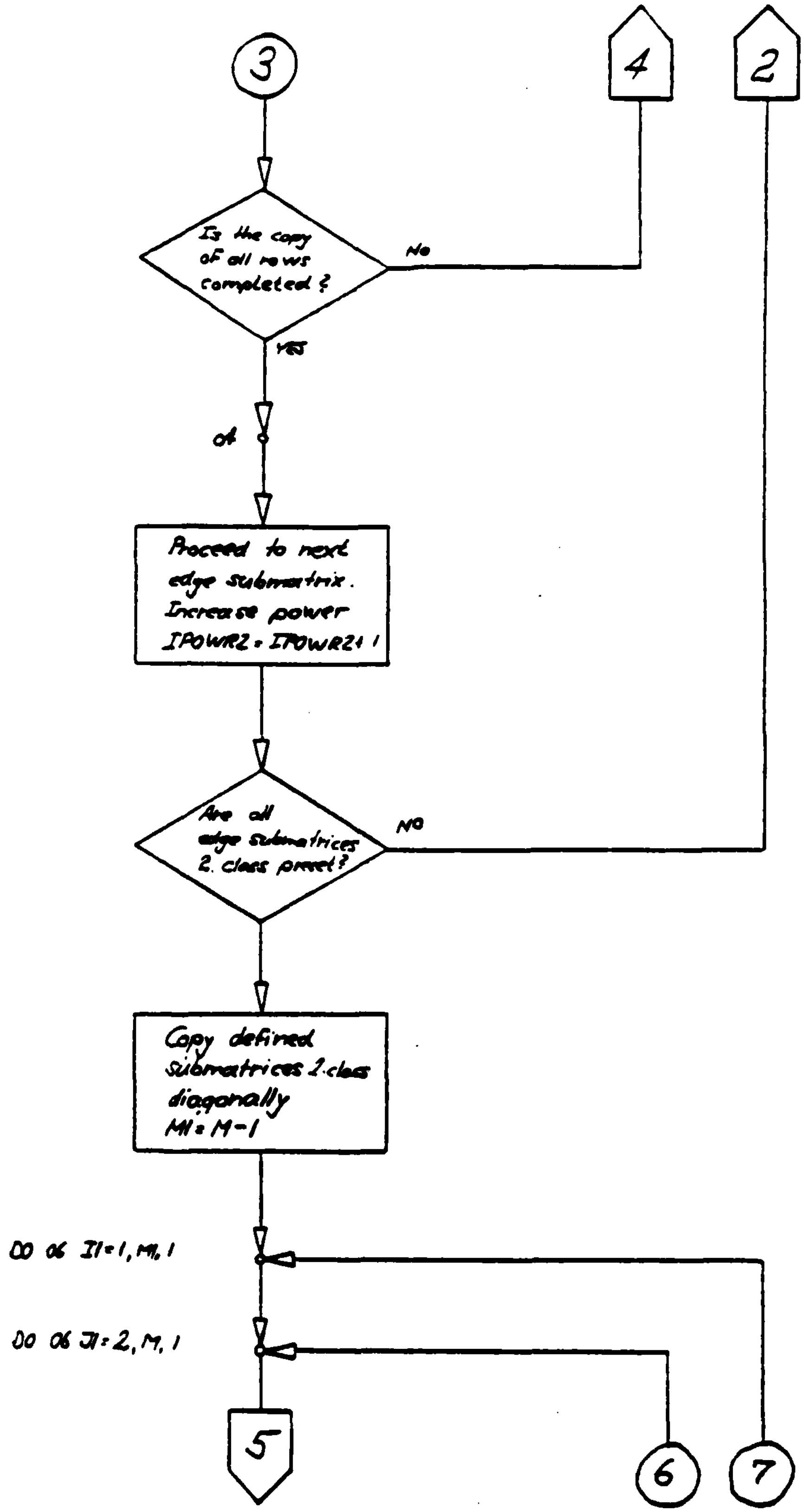

9
10
11
12
DO loops completed
NO
YES
NI = L * MN
Is IPRINT ≠ 2?
Print equation system 4.3.:
System matrix A
Right hand side vector B
IPRINT = 1
Gauss Jordan elimination with matrix A & vector B
13

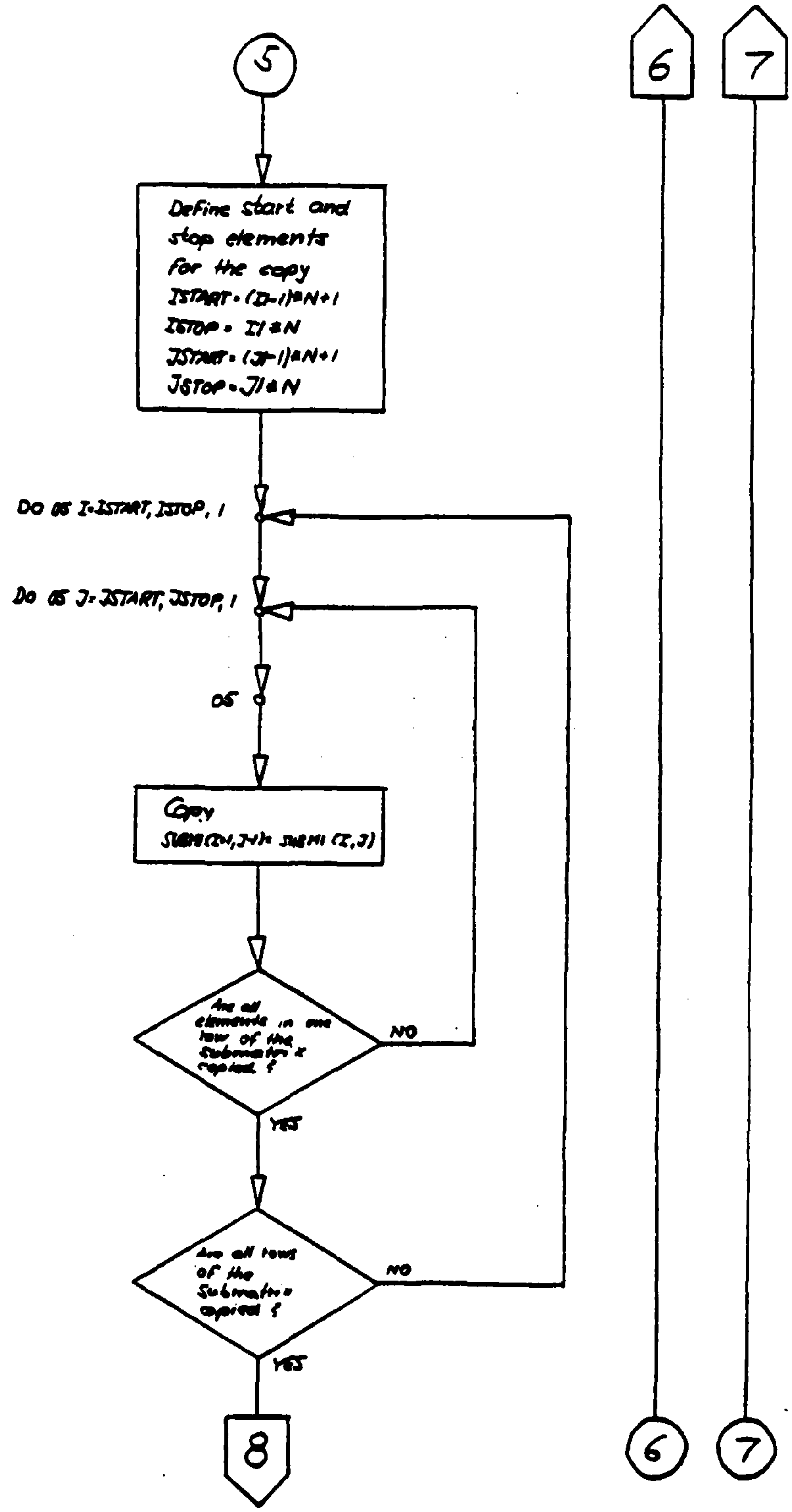

13
Is IPRINT ≠ 1 ?
Print equation system after G.J. elimination
18
Print the coefficients
RETURN

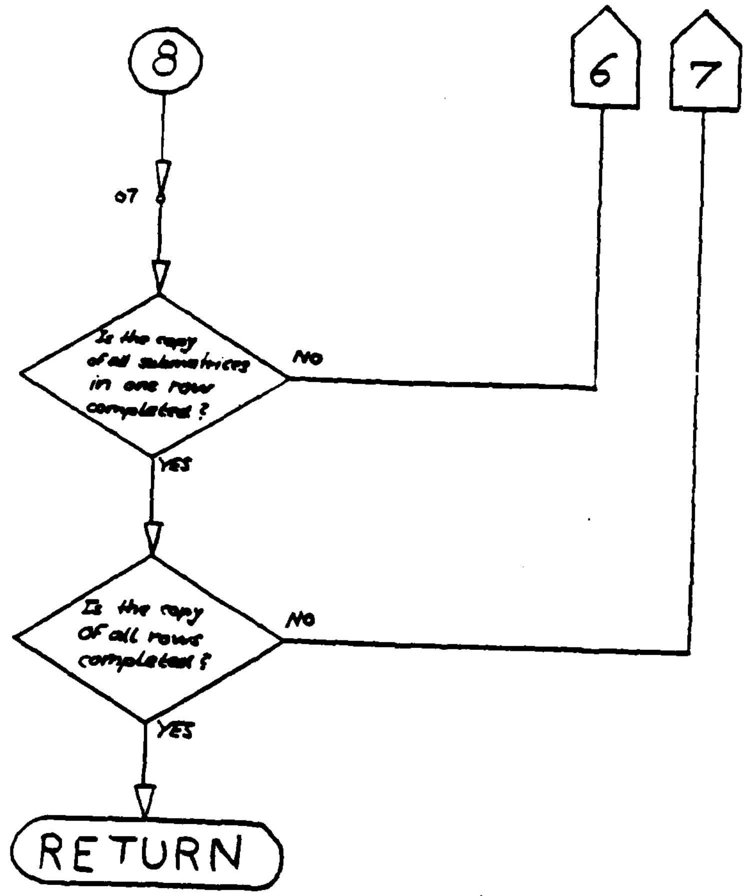
Flow chart MAT41



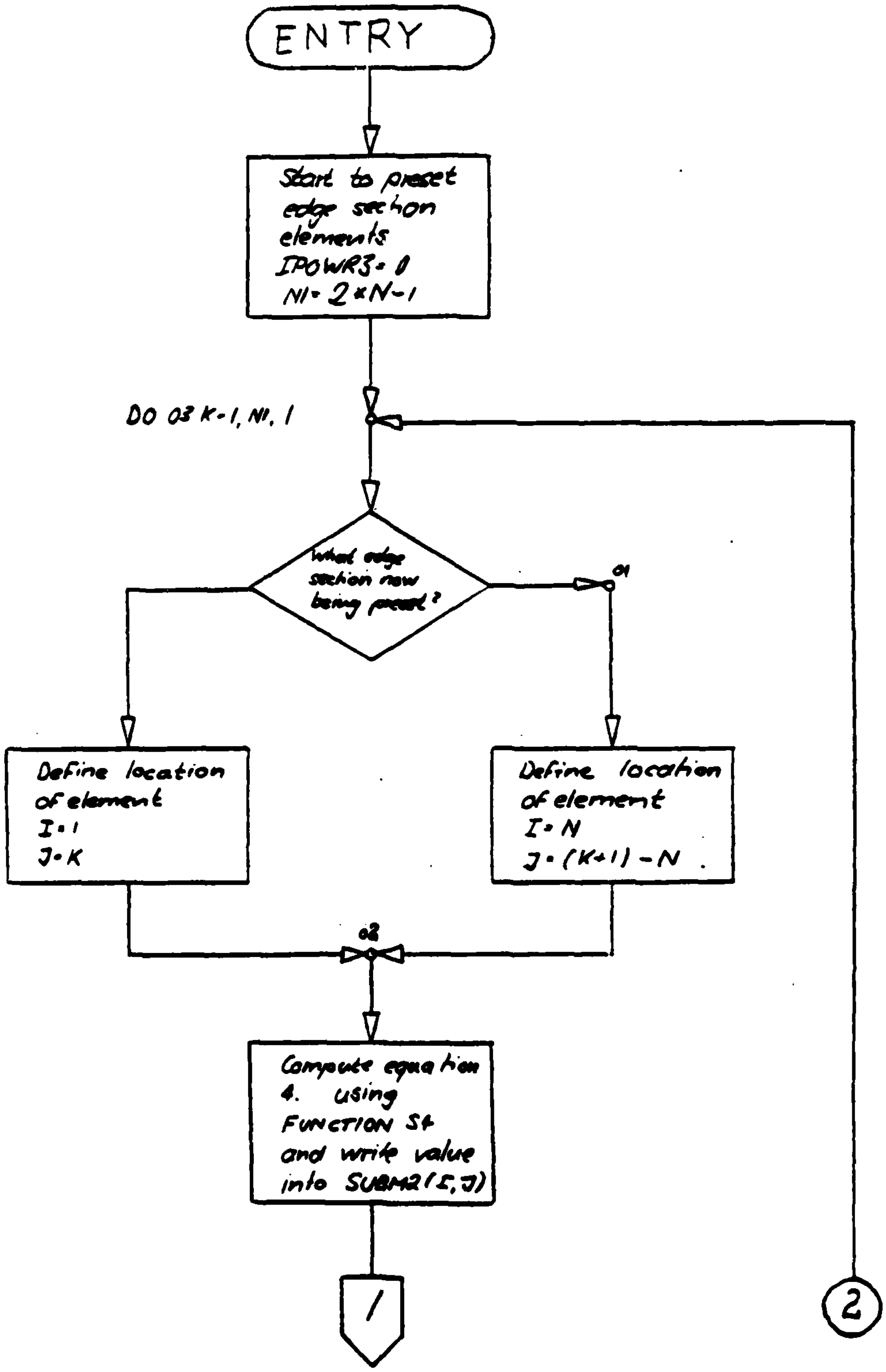

1
2
I1 = 0
DO 03 I=ISTART,ISTOP,1
I1 = I1+1
J1 = 0
DO 03 J=JSTART,JSTOP,1
J1 = J1+1
03
Copy submatrix 2. class SUBM2 as derived from Subroutine HAT42 into correct place in submatrix 1. class SUBM1.
SUBM1(I,J)=SUBM2(I1,J1)
Is the copy of all elements in the row completed?
NO
YES
3
4
2

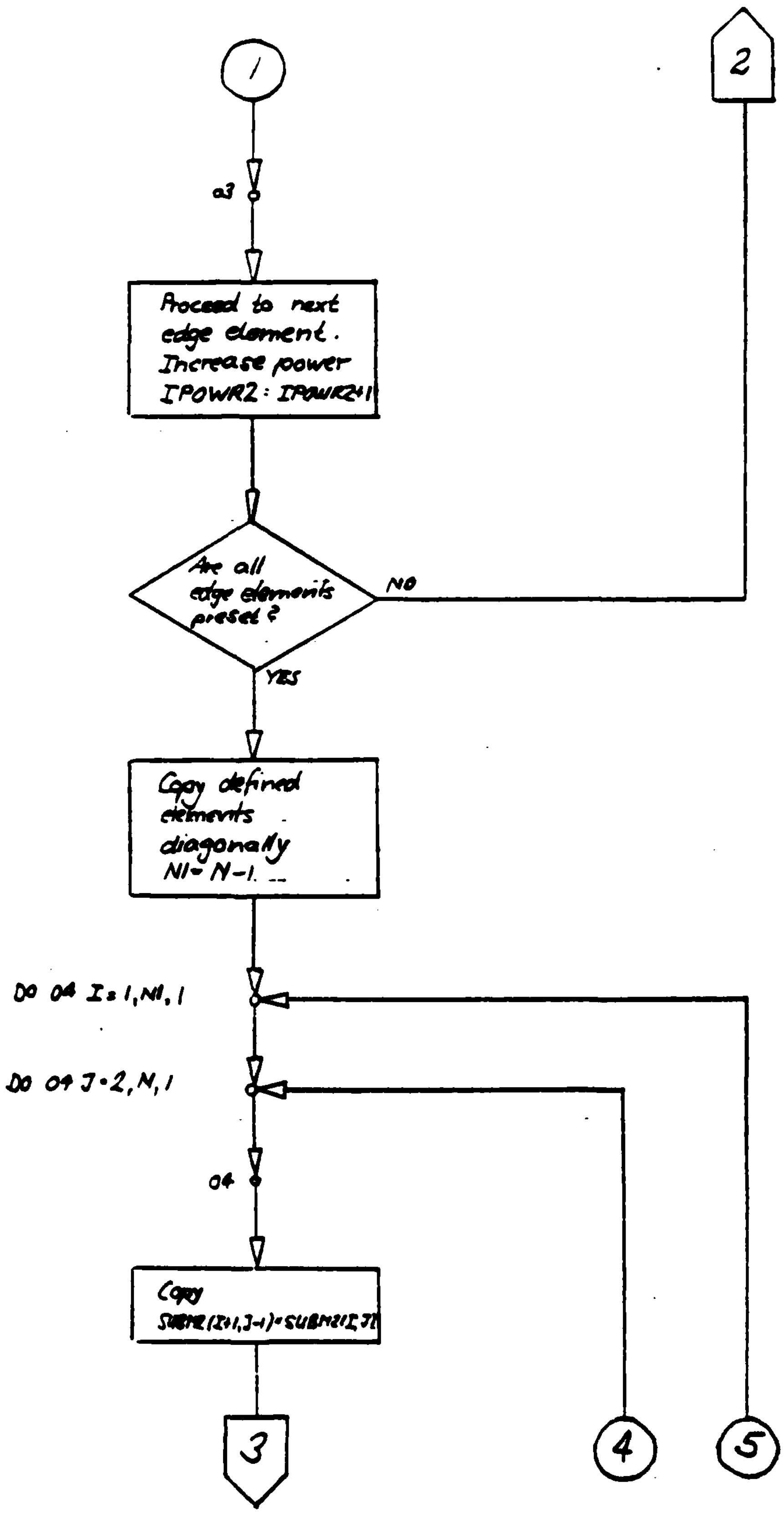

3
4
2
Is the copy of all rows completed?
No
YES
Proceed to next edge submatrix. Increase power IPOWR2 = IPOWR2+1
Are all edge submatrices 2. class preset?
NO
Copy defined submatrices 2.class diagonally M1 = M-1
DO 06 I1=1,M1,1
DO 06 J1=2,M,1
5
6
7

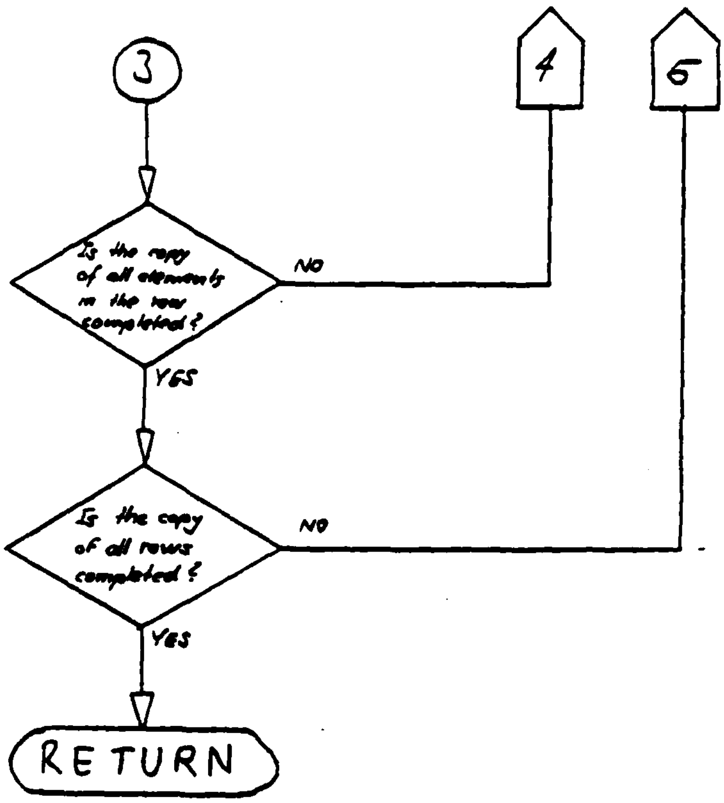

5
6
7
Define start and stop elements for the copy
ISTART = (II-1)*N+1
ISTOP = II*N
JSTART = (JI-1)*N+1
JSTOP = JI*N
DO 05 I=ISTART, ISTOP, 1
Do 05 J=JSTART, JSTOP, 1
05
Copy
SUBM(I-I,J-I) = SUBMI (I,J)
Are all elements in one row of the Submatrix copied ?
NO
YES
Are all rows of the Submatrix copied ?
NO
YES
8
6
7


8
6
7
o7
Is the copy of all submatrices in one row completed ?
NO
YES
Is the copy of all rows completed ?
NO
YES
RETURN

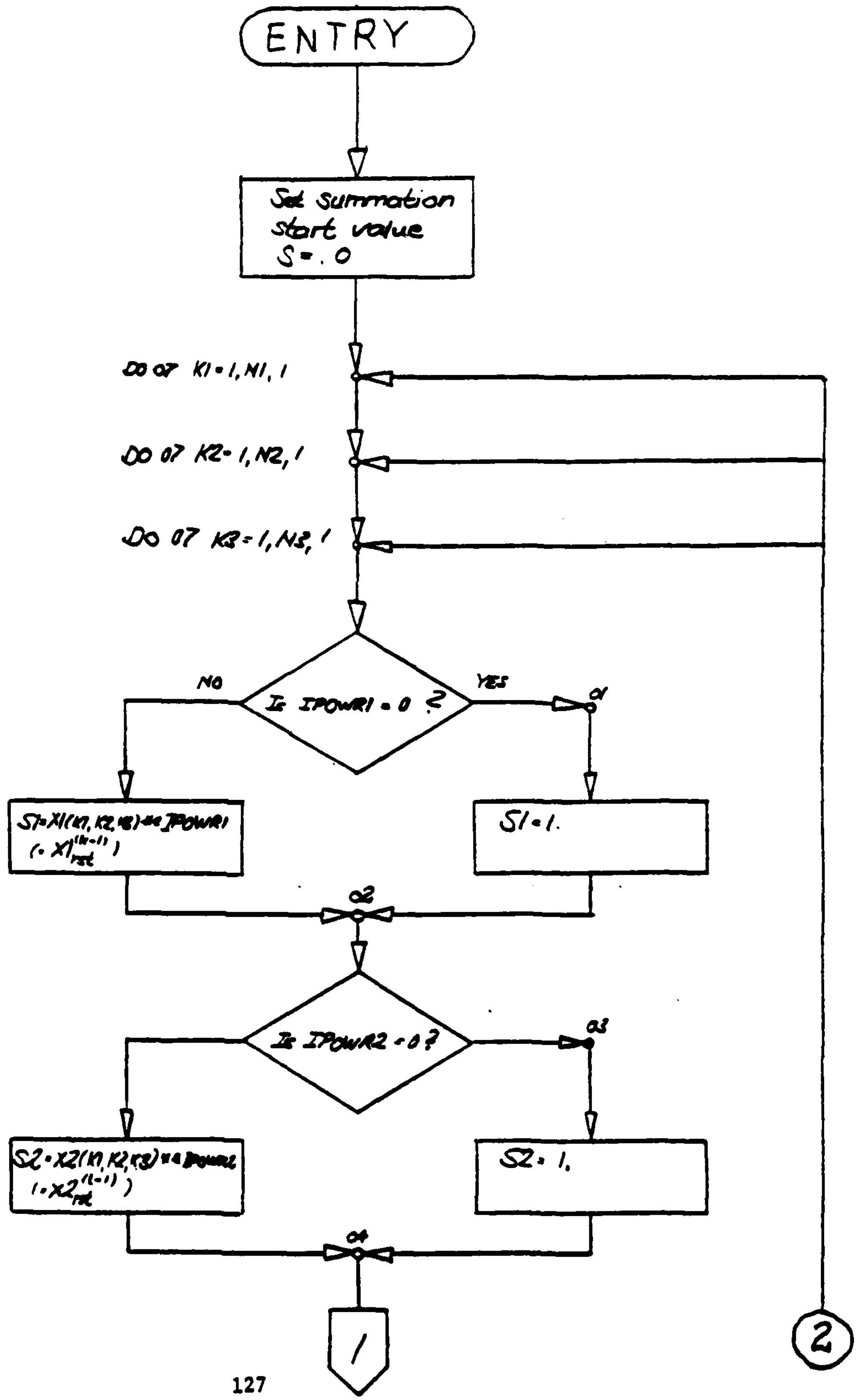
Flow chart MAT42

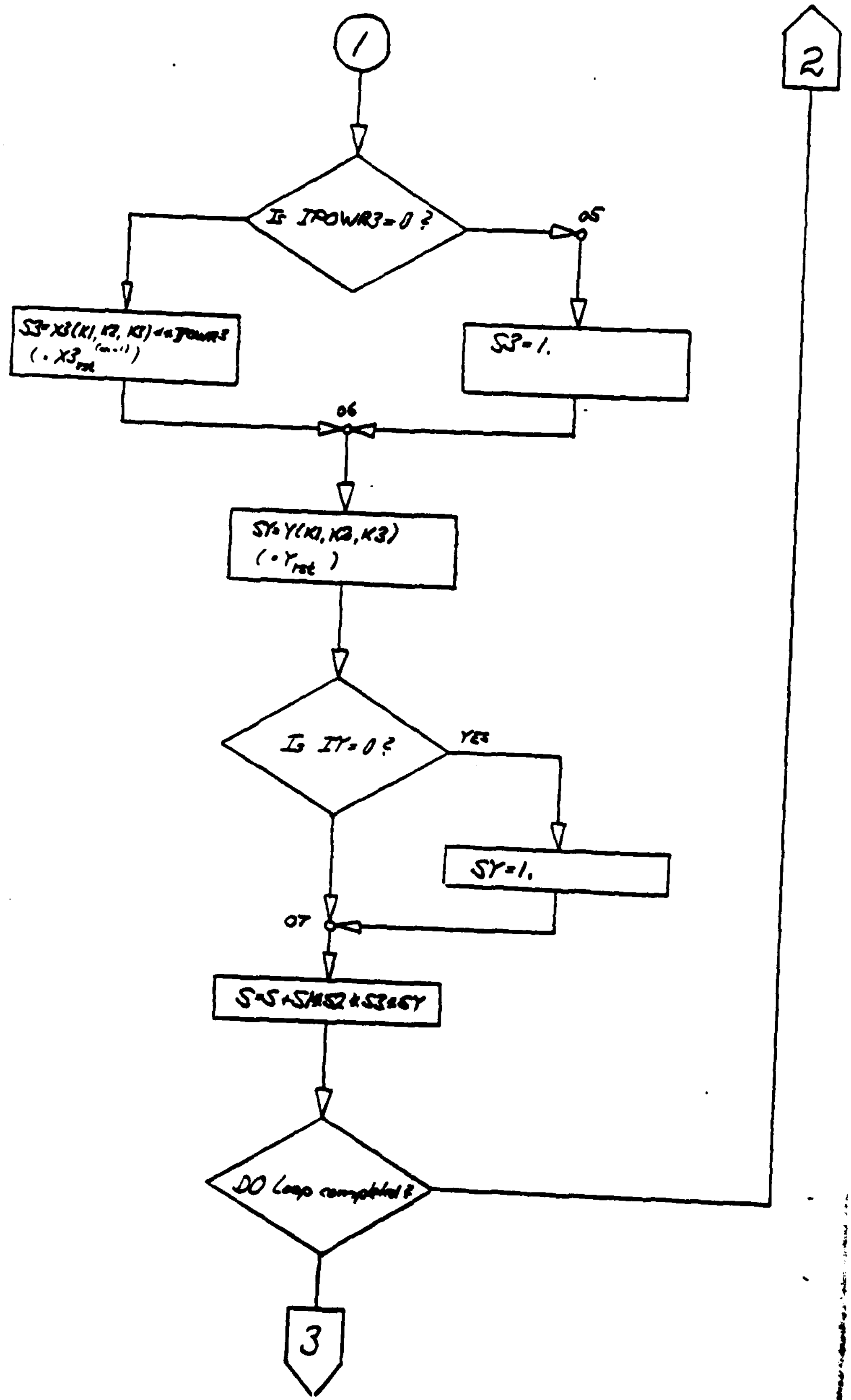


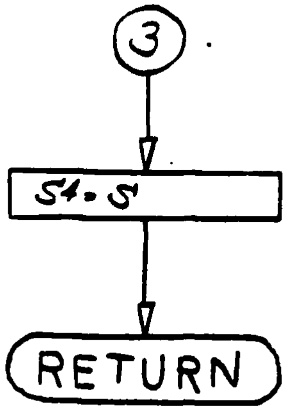

1
2
a3
Proceed to next edge element. Increase power IPOWR2 = IPOWR2+1
Are all edge elements preset?
NO
YES
Copy defined elements diagonally N1 = N-1.
DO 04 I = 1,N1,1
DO 04 J = 2,N,1
04
Copy
3
4
5


3
4
6
Is the copy of all elements in the row completed?
NO
YES
Is the copy of all rows completed?
NO
YES
RETURN

Flow chart S4




1
2
Is IPOWR3 = 0 ?
05
S3 = X3(K1, K2, K3) ** IPOWR3
S3 = 1.
06
SY = Y(K1, K2, K3)
Is IY = 0 ?
YES
SY = 1.
07
DO Loop completed ?
3


3
S4=S
RETURN

APPENDIX C: Software Description: Listings

The following are given:

BLOCK DATA AB2   74/175 OPT=2 PMDMP   FTN 4.8+[illegible]   [illegible]   PAGE

```
1         BLOCK DATA AB2
          COMMON / AB2 / A,B
          REAL A(7,7),B(7)
          END
```

```
1          BLOCK DATA DTA2
           COMMON / DTA2 / X1,Y
           REAL X1(256),Y(256)
           END
```

```
      PROGRAM DEMO2 (INPUT,OUTPUT,TAPE1,TAPE6,TAPE9=INPUT)
C     ************************************************************
C     *                                                          *
C     *  THIS IS A DEMONSTRATION PROGRAM AND IT SHOWS THE CORRECT USE
C     *  OF THE TPL BINARY LIBRARY USING EXAMPLES.               *
C     *                                                          *
C     *  AUTHOR:    HANS ZEHNER                                  *
C     *  DATE:      DECEMBER 24, 1980                            *
C     *                                                          *
C     ************************************************************
      COMMON / DTA2 /X1,Y
      REAL X1(256),Y(256)
      REAL COEF(7)
      LOGICAL IPLOT
      DATA PI     /3.141592654/
      DATA IPLOT /.TRUE./
C     FORMATS DEMO2 START
  101 FORMAT (////72H  HELLO,  THIS IS PROGRAM DEMO2 AND YOU'LL SEE HOW T
     *O USE THE MARVELLOUS/70H  TPL BINARY LIBRARY.  YOUR INTEREST WILL
     *BE GREATLY AWARDED BY EASIER/23H  AND FASTER PROGRAMMING=///)
  102 FORMAT (22H  PLOTTER INITIALIZED.)
  103 FORMAT (25H  DATA POINTS GENERATED.)
  104 FORMAT (19X,5HXMIN,6X,5HXMAX,7X,4HYMIN,7X,4HYMAX/9X,7HPLOTTER,4(
     *1X,F10.3)/9X,4HUSER,4(1X,F10.3))
  105 FORMAT (47H  PLOTTER AND USER AXES DEFINED AND AXES DRAWN.)
  106 FORMAT (20H  DATA POINTS DRAWN.)
  107 FORMAT (54H  ENTER ORDER OF POLYNOMIAL (NORDR1), IPRINT AND ISTAT)
  108 FORMAT (22H  CURVE FIT COMPLETED.)
  109 FORMAT (18H  CURVE FIT DRAWN.)
  601 FORMAT (1H1/1H1)
C     FORMATS DEMO2 STOP
C
C     ************************************************************
C     *                                                          *
C     *  I/O ASSIGNMENTS.                                        *
C     *                                                          *
C     ************************************************************
      LI=LOGLU(ISESSN)
      LO=6
      LP=13
      WRITE (LI, 101)
      WRITE (LO, 601)
C
C     ************************************************************
C     *                                                          *
C     *  INITIALIZE THE PLOTTER.                                 *
C     *                                                          *
C     ************************************************************
      IF (IPLOT) CALL INITG (LP)
      WRITE (LI, 102)
C
C     ************************************************************
C     *                                                          *
C     *  VARIATION OF POLYNOMIAL DEGREE.  LOOP STARTS HERE.      *
C     *                                                          *
C     ************************************************************
      N1STRT=1
      N1STOP=6
      DO 66 NORDR1=N1STRT,N1STOP,1
C
C     ************************************************************
C     *                                                          *
C     *  GENERATE DATA SCATTERED AROUND A POLYNOMIAL.            *
C     *                                                          *
C     ************************************************************
      XSTART = -6.
      XSTOP  = +2.
      NPNTS1 = 51
      DX     = (XSTOP-XSTART)/(NPNTS1-1)
      NORDR1 = 4
      COEF(1)= -1.
      COEF(2)=  1.3
      COEF(3)=  -.0001
      COEF(4)=   .3
```

```
      COEF(5)=    .039
      ARG     = (2.*PI)*.03125
      DO 01 I=1,NPNTS1
      X       = XSTART+(I-1)*DX
      X1(I)   = X
      Y(I)    = FZ(COEF,X,NORDR1) + .25*SIN(X/ARG)
   01 CONTINUE
      ISTOP1  = NPNTS1-1
      DO 02 I=1,ISTOP1,2
      DUMMY   = Y(I+1)
      Y(I+1)  = Y(I)
   02 Y(I)    = DUMMY
      WRITE (LI, 103)
C
C     ************************************************************
C     *  DEFINE PLOTTER AND USER AREAS;  DRAW AXES.              *
C     *                                                          *
C     ************************************************************
      XPMIN   = 2.
      XPMAX   = 20.
      YPMIN   = 2.
      YPMAX   = 20.
      XUMIN   = -9.
      XUMAX   = 3.
      YUMIN   = -3.
      YUMAX   = 16.
      WRITE (LI, 104) XPMIN,XPMAX,YPMIN,YPMAX,XUMIN,XUMAX,YUMIN,YUMAX
      ALPHAX  = 0.
      ALPHAY  = 90.
      XA      = (XPMAX       -XPMIN       )/(XUMAX-XUMIN)
      XB      = (XPMIN*XUMAX-XPMAX*XUMIN)/(XUMAX-XUMIN)
      XL      = (XPMAX       -XPMIN       )
      YA      = (YPMAX       -YPMIN       )/(YUMAX-YUMIN)
      YB      = (YPMIN*YUMAX-YPMAX*YUMIN)/(YUMAX-YUMIN)
      YL      = (YPMAX       -YPMIN       )
      IF (IPLOT) CALL AXES (XPMIN,YPMIN,XL,ALPHAX,2HX1,+2,XUMIN,XUMAX,4H
     *F6.1,+2,+4)
      IF (IPLOT) CALL AXES (XPMIN,YPMIN,YL,ALPHAY,2H Y,-2,YUMIN,YUMAX,2H
     *I2,2,2)
      WRITE (LI, 105)
C
C     ************************************************************
C     *  PLOT DATA POINTS.                                       *
C     *                                                          *
C     ************************************************************
      DO 03 I=1,NPNTS1,1
      XPLOT   = X1(I)
      YPLOT   = Y(I)
      XPLOT   = XPLOT*XA+XB
      YPLOT   = YPLOT*YA+YB
      IF (IPLOT) CALL PLOT (XPLOT,YPLOT,3)
   03 IF (IPLOT) CALL SYMBL (2,XPLOT,YPLOT)
      WRITE (LI, 106)
C
C     ************************************************************
C     *  CALCULATE CURVE FIT THROUGH DATA POINTS.                *
C     *                                                          *
C     ************************************************************
   04 WRITE (LI, 107)
C     READ  ( 5,    *) NORDR1,IPRINT,ISTAT
      NORDR1  = NORDR1
      IPRINT  = 2
      ISTAT   = 2
      L       = NORDR1+1
      IF (L.LT.1 .OR. L.GT.7) GO TO 76
      CALL AXIS (NPNTS1,L,COEF,IPRINT,ISTAT,IERROR)
      IF (IERROR.LT.0) GO TO 08
      WRITE (LI, 108)
C
C     ************************************************************
C     *  PLOT POLYNOMIAL FIT.                                    *
C     *                                                          *
C     ************************************************************
```

```
         DO 08 II=1,NPNTS1,1
         XPLOT  = X1(II)
         YAPP   = F1(C3GP,X1(II),N3RD81)
         XPLOT  = XPLOT*XA+XO
         YPLOT  = YAPP *YA+YO
         IF (II.EQ.1 .AND. IPLOT) CALL PLOT (XPLOT,YPLOT,3)
   08    IF (II.GT.1 .AND. IPLOT) CALL PLOT (XPLOT,YPLOT,2)
         WRITE (LL, 109)
   06    IF (IPLOT) CALL NUPAGE

C        ************************************************************
C        *                                                          *
C        *  TERMINATE GRAFICS.                                      *
C        *                                                          *
C        ************************************************************
         IF (IPLOT) CALL STOPG

         STOP 0077
         END
```

```
      SUBROUTINE MAT2 (NPNTSI,L,COEF,IPRINT,ISTAT,IERROR)
C     ***************************************************************
C     *                                                             *
C     *  PRESET SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B FOR A  *
C     *  2D-POLYNOMIAL APPROXIMATION.  THE LINEAR EQUATION SYSTEM THEN *
C     *  IS SOLVED BY USING A GAUSS JORDAN ELIMINATION.             *
C     *                                                             *
C     *  AUTHOR:     HANS ZEHNER                                    *
C     *  DATE:       DECEMBER 24, 1980                              *
C     *                                                             *
C     *  VARIABLE    EXPLANATION                           TYPE     *
C     *  [illegible] [illegible]                           INTEGER  *
C     *  NPNTSI      NUMBER OF DATA POINTS                 INTEGER  *
C     *  L           NUMBER OF COEFFICIENTS FOR THE        INTEGER  *
C     *              APPROXIMATION POLYNOMIAL =(ORDER+1)            *
C     *  COEF(7)     POLYNOMIAL COEFFICIENTS               REAL     *
C     *  IPRINT      CONTROL QUANTITY OF PRINT OUT         INTEGER  *
C     *              =2 PRINT EQUATION SYSTEM BEFORE                *
C     *                 AND AFTER THE GAUSS JORDAN                  *
C     *                 ELIMINATION.                                *
C     *              =1 PRINT EQUATION SYSTEM AFTER                 *
C     *                 THE GAUSS JORDAN ELIMINATION.               *
C     *              =0 PRINT BASIC EQUATIONS WITH                  *
C     *                 ACTUAL PARAMETERS.                          *
C     *              <0 NO OUTPUT AT ALL.                           *
C     *  ISTAT       CONTROL STATISTICS TO BE PRINTED      INTEGER  *
C     *              /ISTAT/=2 PRINT ORDINATES OF DATA              *
C     *                        AND APPROXIMATED POINTS              *
C     *                        AND DEVIATIONS (ERROR).              *
C     *              /ISTAT/=1 ONLY PRINT ERRORS.                   *
C     *              ISTAT>0   GET ABSOLUTE ERROR.                  *
C     *              ISTAT<0   GET RELATIVE ERROR.                  *
C     *  IERROR      ERROR FLAG RETURNED TO THE CALLING    INTEGER  *
C     *              PROGRAM.                                       *
C     *              =+1 NO ERROR DETECTED. GOOD JOB.               *
C     *              =-1 AN ERROR OCCURED IN THE GAUSSIAN           *
C     *                  ELIMINATION; COEFFICIENTS ARE NOT          *
C     *                  ACCURATE.  REVIEW PARAMETERS.              *
C     *                                                             *
C     *  X1(256)     X1-COORDINATES OF THE DATA POINTS     REAL     *
C     *  Y(256)      Y-COORDINATES OF THE DATA POINTS      REAL     *
C     *                                                             *
C     *  IMPORTANT:  THE DATA POINTS X1 AND Y HAVE TO BE PROVIDED THRU *
C     *              THE LABELED COMMON BLOCK DTA2 AND THE SYSTEM   *
C     *              MATRIX A AND THE RIGHT HAND SIDE VECTOR B ARE  *
C     *              AVAILABLE THROUGH THE LABELED COMMON BLOCK AB2. *
C     *              THIS SAVES A LOT OF THAT PRECIOUS THING CALLED *
C     *              MEMORY.                                        *
C     *                                                             *
C     ***************************************************************
      COMMON / DTA2  / X1,Y
      COMMON / AB2   / A,B
      REAL X1(256),Y(256)
      REAL A(7,7),B(7)
      REAL COEF(7),STAT(256)
C     FORMATS MAT2 START
  101 FORMAT (////7H  AFTER,I3,41H STEPS NO SOLUTION WITHIN THE ACCURACY
     *OF,E10.3,11H   I QUIT./)
  601 FORMAT (////51H  ORDER OF APPROXIMATION POLYNOMIAL IN X1 DIRECTION,
     *2X,I3/40H  NUMBER OF DATA POINTS WITH CONSTANT X1,[illegible])
  602 FORMAT (////62H  SYSTEM MATRIX A AND VECTOR B BEFORE GAUSS JORDAN E
     *LIMINATION/)
  603 FORMAT ([illegible])
  604 FORMAT ([illegible])
  605 FORMAT (////46H  EQUATION SYSTEM AFTER GAUSS JORDAN ELIMINATION/)
  606 FORMAT (////22H  COEFFICIENTS COEF(I)/)
  607 FORMAT (2X,5HC [illegible])
  608 FORMAT (////39H  ORDINATES OF DATA POINTS  -  AS GIVEN/)
  609 FORMAT ((5(2X,I3,1X,1PE9.2)))
  610 FORMAT (////46H  ORDINATES OF DATA POINTS  -  AS APPROXIMATED/)
  611 FORMAT (////32H  ABSOLUTE ERROR AT EACH POINT/)
  612 FORMAT (////35H  RELATIVE ERROR AT EACH POINT IN %/)
  613 FORMAT ((5(2X,I3,1X,2PF9.2)))
  614 FORMAT (////21H  STATISTICS SUMMARY//7X,13HAVERAGE ERROR,10X,1PE9.
     *2/7X,19HMAXIMUM ERROR AT I =,I3,1X,1PE9.2)
  615 FORMAT (////21H  STATISTICS SUMMARY//7X,13HAVERAGE ERROR,10X,2PF9.
     *2,2H %/7X,19HMAXIMUM ERROR AT I =,I3,1X,2PF9.2,2H %)
  616 FORMAT (////7H  AFTER,I3,41H STEPS NO SOLUTION WITHIN THE ACCURACY
     *OF,E10.3,11H   I QUIT./)
C     FORMATS MAT2 STOP
```

```
      LI=LOGLU(ISESSN)
      LU=0
      IERROR=1
      ITMAX=50
      EPSLON=.10CE-007
C
C     PRINT APPROXIMATION PARAMETERS. IF IPRINT IS GREATER THAN 0.
C
      NORDR1=L-1
      IF (IPRINT.LT.0) GO TO 01
      WRITE (LO, 001) NORDR1,NPNTS1
C
C     START TO PRESET EDGE SECTION ELEMENTS.
C
   01 IPOWR1=0
      L1=2*L-1
      DO 04 K=1,L1,1
      IF (K.GT.L) GO TO 02
C
C     TOP EDGE SECTION ELEMENTS.
C
      I=1
      J=K
      GO TO 03
C
C     RIGHT HAND EDGE SECTION ELEMENTS.
C
   02 I=(K+1)-L
      J=L
C
C     PRESET EDGE SECTION ELEMENTS IN SYSTEM MATRIX USING S2.
C
   03 A(I,J)=S2(NPNTS1,IPOWR1,0)
   04 IPOWR1=IPOWR1+1
C
C     COPY DEFINED ELEMENTS DIAGONALLY.
C
      L1=L-1
      DO 05 I=1,L1,1
      DO 05 J=2,L,1
   05 A(I+1,J-1)=A(I,J)
C
C     PRESET RIGHT HAND SIDE VECTOR B.
C
      DO 06 I=1,L,1
      IPOWR1=I-1
   06 B(I)=S2(NPNTS1,IPOWR1,1)
C
C     PRINT SYSTEM MATRIX AND RIGHT HAND SIDE VECTOR B. IF IPRINT
C     IS SET TO 2.
C
      IF (IPRINT.NE.2) GO TO 08
      WRITE (LO, 002)
      WRITE (LO, 003) (J,J=1,L,1)
      DO 07 I=1,L,1
   07 WRITE (LO, 004) I,(A(I,J),J=1,L,1),B(I)
      WRITE (LO, 003) (J,J=1,L,1)
      IPRINT=1
C
C   . GAUSS JORDAN ELIMINATION.
C
   08 M1=L
      EPSLON=EPSLON*M1
      CALL IELZ (M1,ITER,ITMAX,EPSLON)
      IF (ITER.LT.0) GO TO 21
C
C     PRINT EQUATION SYSTEM, IF IPRINT IS SET TO 1.
C
      IF (IPRINT.NE.1) GO TO 10
      WRITE (LO, 005)
      WRITE (LO, 003) (J,J=1,M1,1)
      DO 09 I=1,M1,1
   09 WRITE (LO, 004) I,(A(I,J),J=1,M1,1),B(I)
      WRITE (LO, 003) (J,J=1,M1,1)
C
C     DEFINE COEFFICIENTS AND PRINT THEM, IF IPRINT IS GREATER THAN 0.
C
   10 CONTINUE
      DO 11 L1=1,L,1
   11 COEF(L1)=B(L1)
      IF (IPRINT.LT.0) GO TO 13
      WRITE (LO, 006)
      DO 12 L1=1,L,1
   12 WRITE (LO, 007) L1,COEF(L1)
C
```

```
C
C     STATISTICS, IF ISTAT ANYTHING ELSE BUT 0.
C
   13 IF (ISTAT.EQ.0) GO TO 23
      IF (ISTAT.LT.0) IERROR=1
      IF (ISTAT.GT.0) IERROR=2
      ISTAT=IABS(ISTAT)
C
C     COMPUTE ABSOLUTE ERROR.
C     ( =(APPROXIMATED VALUE  -  GIVEN VALUE) )
C
      DO 14 I=1,NPNTS1,1
   14 STAT(I)=F2(COEF,X1(I),NORDR1)-Y(I)
C
C     COMPUTE RELATIVE ERROR, IF ISTAT WAS LESS THAN ZERO, I.E. IERROR=1.
C     ( =(APPROXIMATED VALUE  -  GIVEN VALUE)/(GIVEN VALUE) )
C
      IF (IERROR.NE.1) GO TO 16
      DO 15 I=1,NPNTS1,1
   15 STAT(I)=STAT(I)/Y(I)
C
C     ERROR STATISTICS:  COMPUTE AVERAGE ERROR AND FIND LOCATION
C     OF MAXIMUM ERROR.
C
   16 ERRAVE=.0
      ERRMAX=ABS(STAT(1))
      IMAX=1
      DO 17 I=1,NPNTS1,1
      IF (ABS(STAT(I)).LE.ERRMAX) GO TO 17
      ERRMAX=ABS(STAT(I))
      IMAX=I
   17 ERRAVE=ERRAVE+ABS(STAT(I))
      ERRAVE=ERRAVE/NPNTS1
      IF (ISTAT.NE.2) GO TO 18
C
C     PRINT ORDINATES OF DATA POINTS AND APPROXIMATED VALUES,
C     IF /ISTAT/ IS EQUAL 2.
C
      WRITE (LO, 608)
      WRITE (LO, 609) (I,Y(I),I=1,NPNTS1,1)
      WRITE (LO, 610)
      WRITE (LO, 609) (I,F2(COEF,X1(I),NORDR1),I=1,NPNTS1,1)
      ISTAT=1
C
C     PRINT ERRORS, IF /ISTAT/ WAS 1 OR 2.
C
   18 IF (ISTAT.NE.1) GO TO 20
      IF (IERROR.EQ.1) GO TO 19
C
C     PRINT ABSOLUTE ERRORS.
C
      WRITE (LO, 611)
      WRITE (LO, 609) (I,STAT(I),I=1,NPNTS1,1)
      GO TO 20
C
C     PRINT RELATIVE ERRORS.
C
   19 WRITE (LO, 612)
      WRITE (LO, 613) (I,STAT(I),I=1,NPNTS1,1)
C
C     PRINT ERROR STATISTICS.
C
   20 IF (IERROR.EQ.2) WRITE (LO, 614) ERRAVE,IMAX,ERRMAX
      IF (IERROR.EQ.1) WRITE (LO, 615) ERRAVE,IMAX,ERRMAX
      GO TO 23
C
C     FATAL ERROR DETECTED.  OUTPUT ERROR MESSAGE AND PRESET COEFFICIENTS.
C
   21 ITER=-ITER
      IERROR=-IERROR
      B(1)=Y(1)
      DO 22 IC=2,N1,1
   22 B(IC)=.0
      WRITE (L1, 101) ITER,EPSLON
      WRITE (LO, 616) ITER,EPSLON
      GO TO 10
C
C     RETURN.
C
   23 RETURN
      END
```

```
      REAL FUNCTION S2 (NPNTS1,IPOWR1,IY)
C***********************************************************************
C*                                                                     *
C*  PERFORM SUMMATIONS TO PRESET THE ELEMENTS A(I,J) IN THE            *
C*  SYSTEM MATRIX A AND THE ELEMENTS B(K) IN THE RIGHT HAND            *
C*  SIDE VECTOR B FOR THE LINEAR EQUATION SYSTEM, THAT MUST            *
C*  BE SOLVED FOR A POLYNOMIAL 2D-APPROXIMATION.                       *
C*                                                                     *
C*  AUTHOR:     HANS ZEHNER                                            *
C*  DATE:       DECEMBER 24, 1980                                      *
C*                                                                     *
C*  VARIABLE    EXPLANATION                                 TYPE       *
C*  ********    ***********************************         *******    *
C*  NPNTS1      NUMBER OF DATA POINTS AT CONSTANT XI        INTEGER    *
C*  IPOWR1      POWER FOR THE XI VALUES IN THE              INTEGER    *
C*              SUMMATION.                                             *
C*  IY          CONTROL PARAMETER                           INTEGER    *
C*              =0  Y VALUES ARE NOT INCLUDED IN THE                   *
C*                  SUMMATION.                                         *
C*                  (-> SYSTEM MATRIX ELEMENTS)                        *
C*              =1  Y VALUES ARE INCLUDED IN THE            INTEGER    *
C*                  SUMMATION.                                         *
C*                  (-> RIGHT HAND SIDE VECTOR ELEM.)                  *
C*  S2          RESULT OF SUMMATION.                        REAL       *
C*                                                                     *
C*  XI(250)     XI-COORDINATES OF THE DATA POINTS           REAL       *
C*  Y(250)      Y-COORDINATES OF THE DATA POINTS            REAL       *
C*                                                                     *
C*  IMPORTANT:  THE DATA POINTS XI AND Y HAVE TO BE PROVIDED THRU      *
C*              THE LABELED COMMON BLOCK DTA2.                         *
C*                                                                     *
C***********************************************************************
      COMMON / DTA2 / XI,Y
      REAL XI(250),Y(250)
      S=0.
      DO 93 K1=1,NPNTS1
      IF (IPOWR1.EQ.0) GO TO 91
      SX=XI(K1)**IPOWR1
      GO TO 92
   91 SX=1.
   92 SY=Y(K1)
      IF (IY.EQ.0) SY=1.
   93 S=S+SX*SY
      S2=S
      RETURN
      END
```

```
      INTEGER FUNCTION IELZ2 (N)
C     ***************************************************************
C     *                                                             *
C     *  GAUSS JORDAN ELIMINATION WITH MATRIX A AND VECTOR B.  THE  *
C     *  VARIABLE IELZ IS SET TO 1, IF THE ELIMINATION IS COMPLETED.*
C     *  IELZ LESS THAN 0 INDICATES AN ERROR.                       *
C     *                                                             *
C     *  AUTHOR:     HANS ZEAMER                                    *
C     *  DATE:       DECEMBER 24, 1980                              *
C     *  VARIABLE    EXPLANATION                          TYPE      *
C     *  ********    ******************************       ********  *
C     *  N           NUMBER OF EQUATIONS                  INTEGER   *
C     *  IELZ        CONTROL PARAMETER, RETURNED TO THE   INTEGER   *
C     *              CALLING ROUTINE.                               *
C     *              =1 GAUSS JORDAN ELIMINATION WAS                *
C     *                 PERFORMED CORRECTLY.                        *
C     *              <0 ERROR ENCOUNTERED DURING GAUSS              *
C     *                 JORDAN ELIMINATION.                         *
C     *                                                             *
C     *  A(7,7)      SYSTEM MATRIX A                      REAL      *
C     *  B(7)        RIGHT HAND SIDE VECTOR B             REAL      *
C     *                                                             *
C     *  IMPORTANT:  MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH
C     *              THE LABELED COMMON BLOCK ABZ.  BOTH A AND B WILL
C     *              WILL UNDERGO CHANGES DURING EXECUTION, I.E. THE
C     *              ELIMINATION.                                   *
C     *                                                             *
C     ***************************************************************
      COMMON / ABZ  / A,B
      REAL A(7,7),B(7)
      IELZ=1
      IELZ2=IELZ
      EPS=1.0E-030
      DO 06 I=1,N
      DO 02 K=1,N
      IF (K.EQ.I) GO TO 02
      CONST=-A(K,I)/A(I,I)
      DO 01 J=1,N
      A(K,J)=A(K,J)+CONST*A(I,J)
      IF (J.EQ.I) A(K,J)=0.0
   01 CONTINUE
      B(K)=B(K)+CONST*B(I)
   02 CONTINUE
      CONST=A(I,I)
      DO 03 J=1,N
   03 A(I,J)=A(I,J)/CONST
      A(I,I)=1.
      B(I)=B(I)/CONST
      DO 06 J=1,N
      IERROR=0
      DO 04 K=1,N
      IF (ABS(A(J,K)).GT.EPS) GO TO 04
      IERROR=IERROR+1
   04 CONTINUE
      IF (ABS(B(J)).GT.EPS) GO TO 05
      IERROR=IERROR+1
   05 IF (IERROR.GT.N+1) GO TO 07
   06 CONTINUE
      GO TO 08
   07 IELZ=-1
      IELZ2=IELZ
   08 RETURN
      END
```

```
      SUBROUTINE IELZ (N,ITER,ITRMAX,EPSILON)
C     *****************************************************************
C     *                                                               *
C     *  GAUSSIAN ELIMINATION WITH LINEAR EQUATION SYSTEM V (THE RIGHT *
C     *  HAND SIDE VECTOR B IS INTEGRATED TOGETHER WITH THE SYSTEM MA- *
C     *  TRIX A IN THE MATRIX V). ITER IS THE NUMBER OF ITERATIONS RE- *
C     *  TURNED TO THE CALLING PROGRAM. THE ITERATION TO IMPROVES THE  *
C     *  SOLUTION SO MANY TIMES, UNTIL THE REMAINING ERROR IS WITHIN   *
C     *  THE MAGNITUDE OF EPSILON. SHOULD THE NUMBER OF ITERATIONS EX- *
C     *  CEED ITRMAX, THE SUBROUTINE TERMINATES THE CALCULATION AND    *
C     *  SETS ITER TO A NEGATIVE VALUE.                                *
C     *                                                               *
C     *  AUTHORS:   KONRAD VOGELER, HANS ZEBNER                        *
C     *  DATE:      DECEMBER 24, 1980                                  *
C     *                                                               *
C     *  VARIABLE   EXPLANATION                          TYPE          *
C     *  ********   *******************************      *******       *
C     *  N          NUMBER OF EQUATIONS                  INTEGER       *
C     *  ITER       NUMBER OF ITERATIONS TO BE RETURNED  INTEGER       *
C     *             >0 GAUSSIAN ELIMINATION PERFORMED                 *
C     *                CORRECTLY.                                     *
C     *             <0 NO CONVERGENCE OF THE SOLUTION FOR             *
C     *                THE DESIRED RESOLUTION EPSILON.                *
C     *                ITERATION DID TERMINATE AFTER                  *
C     *                ITRMAX ITERATION STEPS.                        *
C     *  ITRMAX     MAXIMUM NUMBER OF ITERATION STEPS    INTEGER       *
C     *  EPSILON    DESIRED RESOLUTION OF THE SOLUTION   REAL          *
C     *                                                               *
C     *  A(7,7)     SYSTEM MATRIX A                      REAL          *
C     *  B(7)       RIGHT HAND SIDE VECTOR B             REAL          *
C     *                                                               *
C     *  IMPORTANT: MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH *
C     *             THE LABELED COMMON BLOCK AUS. BOTH A AND B WILL   *
C     *             WILL UNDERGO CHANGES DURING EXECUTION, I.E. THE   *
C     *             ELIMINATION.                                      *
C     *                                                               *
C     *****************************************************************
      COMMON / AUS / V
      REAL V(7,8),[illegible],A(7,8),B(8)
      LOGICAL IPRINT
      DATA IPRINT /.FALSE./
      DATA PI /3.14159265/
C     FORMATS IELZ START
  001 FORMAT (///30H  MATRIX A UND VEKTOR B VOR ELIMINATION/)
  002 FORMAT ([illegible])
  003 FORMAT ([illegible])
  004 FORMAT (///25H  [illegible] SYSTEM [illegible] ITERATIONSSCHRITT/
     *)
  005 FORMAT (////2X,I2,24H KOEFFIZIENTEN AUS GAUSS/)
  006 FORMAT ([illegible])
  007 FORMAT ([illegible] ZEILENVEKTORSCHMITTWINKEL [illegible])
  008 FORMAT ([illegible])
  009 FORMAT ([illegible])
  010 FORMAT (///25H  SYSTEM NACH ELIMINATION/)
C     FORMATS IELZ STOP
      L3=6
      M=N+1
      K=N-1
C
C     REARRANGE VECTOR B IN ARRAY V, IF N IS LESS THAN 7.
C
      IF (N.EQ.7) GO TO 02
      DO 01 I=1,N
   01 V(I,M)=V(I,8)
C
C     PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B BEFORE THE
C     ELIMINATION, IF IPRINT IS SET .TRUE. .
C
   02 IF (IPRINT) WRITE (L3, 001)
      IF (IPRINT) WRITE (L3, 002) (J,J=1,M)
      DO 03 I=1,N
   03 IF (IPRINT) WRITE (L3, 003) I,(V(I,J),J=1,M)
      IF (IPRINT) WRITE (L3, 002) (J,J=1,M)
C
C     LINEAR EQUATION SYSTEM  V(N,X)=B(N)=V(N,N+1)
C
      DO 04 I=1,N
      DO 04 J=1,N
   04 A(I,J)=V(I,J)
      DO 05 I=1,N
   05 B(I)=.0
C
C     CHECK FOR STABILITY.
```

```
C
      FI=.0
      DO 06 I=1,N-1
   06 G(I)=.0
      DO 08 I=1,N-1
      DO 07 J=1,N-1
   07 G(I)=G(I)+V(I,J)**2
   08 G(I)=SQRT(G(I))
      DO 10 I=1,N-2
      JJ=I+1
      DO 10 J=JJ,N-1
      SK=.0
      DO 09 I3=1,N-1
   09 SK=SK+V(I,I3)*V(J,I3)
      F=SK/(G(I)*G(J))
      IF (ABS(F).GT.ABS(FI)) FI=F
   10 CONTINUE
      FI=ABS(FI)
      FI=ACOS(FI)
      FI=(FI*180.)/PI
C
C     START ITERATION TO FIND AND IMPROVE SOLUTION.
C
      DO 27 LI=1,999,1
      ITER=LI
      DO 15 J=1,N-1
C
C     PIVOTISATION AND GAUSSIAN STEP FOR COLUMN J.
C
      IX=J
      E=V(J,J)
      DO 12 I=J,N-1
      IF (ABS(V(I,J)).GT.ABS(E)) GO TO 11
      GO TO 12
   11 IX=I
      E=V(I,J)
   12 CONTINUE
      IF (IX.EQ.J) GO TO 14
      DO 13 I=1,N+1
      G(I)=V(IX,I)
      V(IX,I)=V(J,I)
   13 V(J,I)=G(I)
      G(N)=.0
   14 MM=J+1
      DO 15 I=MM,N-1
      W=V(I,J)/V(J,J)
      DO 15 KK=J,M
      V(I,KK)=V(I,KK)-V(J,KK)*W
   15 CONTINUE
      I=N
      IF (V(I,I).EQ.0.) GO TO 32
C
C     SOLUTION.
C
      DO 16 I=1,N-1
   16 G(I)=.0
      G(N)=V(N,M)/V(N,N)
      DO 19 II=1,N-1
      I=N-II
      IF (V(I,I).EQ.0.) GO TO 32
      G(I)=V(I,M)
      DO 17 JJ=1,N-1
      J=N-JJ
      IF (J.EQ.I) GO TO 18
   17 G(I)=G(I)-G(J)*V(I,J)
   18 CONTINUE
   19 G(I)=G(I)/V(I,I)
C
C     CORRECT SOLUTION BY AMOUNT OF ERROR, WHICH WAS INVOLVED IN THIS
C     ITERATION STEP.  TAKE THIS VECTOR AS START VECTOR FOR THE NEXT
C     ITERATION.
C
      DO 20 I=1,N-1
   20 G(I)=B(I)+G(I)
C
C     COMPUTE THE REMAINING ERROR.
C
      DO 22 I=1,N-1
      V(I,M)=.0
      DO 21 J=1,N-1
   21 V(I,M)=V(I,M)+G(J)*A(I,J)
   22 V(I,M)=A(I,M)-V(I,M)
      R=.0
      DO 23 I=1,N-1
   23 R=R+V(I,M)*V(I,M)
```

```
      R=SQRT(R)
      IF (R.LT.EPSLUM) GO TO 26
      IF (L1.GT.ITERMAX) GO TO 33
      DO 24 I=1,M+1
   24 B(I)=G(I)
C
C     PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B AFTER THE
C     PIVOTISATION.
C
      IF (IPRINT) WRITE (LO, 604) LX
      IF (IPRINT) WRITE (LO, 602) (J,J=1,M+1)
      DO 25 I=1,M+1
   25 IF (IPRINT) WRITE (LO, 603) I,(V(I,J),J=1,M+1),G(I)
      IF (IPRINT) WRITE (LO, 502) (J,J=1,M+1)
      DO 26 I=1,M+1
      DO 26 J=1,M+1
   26 V(I,J)=A(I,J)
C
C     STOP LOOP TO FIND SOLUTION.
C
   27 CONTINUE
C
C     REARRANGE SOLUTION IN THE ARRAY V.
C
   28 CONTINUE
      DO 29 I=1,M+1
   29 V(I,6)=G(I)
C
C     PRINT SOLUTION, IF IPRINT IS SET .TRUE. .
C
   30 IF (IPRINT) WRITE (LO, 605) 4
      IF (IPRINT) WRITE (LO, 606) (I,G(I),I=1,M+1)
      IF (IPRINT) WRITE (LO, 607) FI
      IF (IPRINT) WRITE (LO, 608) 4,LX
C
C     DRUCKEN DER MATRIX A UND DES VECTORS B NACH ELIMINATION.
C
      IF (IPRINT) WRITE (LO, 511)
      IF (IPRINT) WRITE (LO, 602) (J,J=1,M+1)
      DO 31 I=1,M+1
   31 IF (IPRINT) WRITE (LO, 503) I,(V(I,J),J=1,M+1)
      IF (IPRINT) WRITE (LO, 502) (J,J=1,M+1)
      RETURN
C
C     ERROR ENCOUNTERED.  STOP.
C
   32 ITER=-1
      WRITE (LO, 609) I,I,V(I,I)
      G(I)=G(I)/V(I,I)
      RETURN
C
C     NO CONVERGENCE OF THE SOLUTION.
C
   33 ITER=-ITER
      RETURN
      END
```

```
FUNCTION F2        74/175  OPT=2  PROMP                  FTN ...

      REAL FUNCTION F2 (COEF,X1,NORDR1)
C     ***************************************************************
C     *                                                             *
C     *  COMPUTE FUNCTION VALUE Y=F(X1) OF A 2D-POLYNOMIAL USING    *
C     *  THE HORNER SCHEME IN ORDER TO INCREASE SPEED AND ACCURACY. *
C     *                                                             *
C     *  AUTHOR:    HANS ZEHNER                                     *
C     *  DATE:      DECEMBER 24, 1980                               *
C     *                                                             *
C     *  VARIABLE   EXPLANATION                           TYPE      *
C     *  ========   ===========                           ====      *
C     *  COEF(*)    POLYNOMIAL COEFFICIENTS               REAL      *
C     *  X1         X1-VALUE, WHERE Y WILL BE COMPUTED.   REAL      *
C     *  NORDR1     ORDER OF THE POLYNOMIAL IN THE        INTEGER   *
C     *             X1 DIRECTION.                                   *
C     *  F2         ORDINATE OF THE 2D-POLYNOMIAL.        REAL      *
C     *                                                             *
C     ***************************************************************
      REAL COEF(1)
      NCOEF1=NORDR1+1
      Y=COEF(NCOEF1)
      DO 31 I1=1,NORDR1,1
      I=NCOEF1-I1
   31 Y=Y*X1+COEF(I)
      F2=Y
      RETURN
      END
```

```
BLOCK DATA AB3          74/175  OPT=2  FTNMAP                    FTN [illegible]          [illegible]          PAGE  1

1               BLOCK DATA AB3
                COMMON / AB3  / A,B
                REAL A(40,40),B(40)
                END
```

```
1                BLOCK DATA DATA3
                 COMMON / DATA3 / Z1,Z2,Y
                 REAL Z1(10,10),Z2(10,10),Y(10,10)
                 END
```

```
      PROGRAM DEMO3 (INPUT,OUTPUT,TAPE1,TAPE6,TAPE2,TAPE3=INPUT)
C     ************************************************************
C     *                                                          *
C     *  THIS IS A DEMONSTRATION PROGRAM AND IT SHOWS THE CORRECT USE
C     *  OF THE TPL BINARY LIBRARY USING EXAMPLES.               *
C     *                                                          *
C     *  AUTHOR:     HANS ZEMMER                                 *
C     *  DATE:       DECEMBER 31, 1980                           *
C     *                                                          *
C     ************************************************************
      COMMON / AFLO  / PLOTR
      COMMON / DATA1 / X1,X2,Y
      REAL PLOTR(256)
      REAL X1(10,10),X2(10,10),Y(10,10)
      REAL COEF(49),SUBM1(7,7)
      INTEGER IPAM(5)
      LOGICAL IPLOT
      DATA ICNT  /10*0        /
      DATA IPLOT /.TRUE./
      DATA PI    /3.141592654/
C     FORMATS DEMO3 START
  101 FORMAT (////72H   HELLO. THIS IS PROGRAM DEMO3 AND WE'LL SEE, HOW T
     *O USE THE MARVELLOUS/70H  TPL BINARY LIBRARY.  YOUR INTEREST WILL
     *BE GREATLY AWARDED BY EASIER/23H  AND FASTER PROGRAMMING///)
  102 FORMAT (32H  PLOTTER INITIALIZED.)
  103 FORMAT (31H  DATA POINTS READ FROM TAPE 0.)
  104 FORMAT ([illegible]
     *[illegible])
  105 FORMAT (47H  PLOTTER AND USER AXES DEFINED AND AXES DRAWN.)
  106 FORMAT (30H  MEASURED DATA SURFACE DRAWN.)
  107 FORMAT (69H  ENTER ORDER OF POLYNOMIAL IN X1- AND X2-DIRECTION, IP
     *RINT AND ISTAT)
  108 FORMAT (25H  SURFACE FIT COMPLETED.)
  109 FORMAT (29H  APPROXIMATED SURFACE DRAWN.)
  601 FORMAT (1H1/1HT)
  801 FORMAT (A1,A1,A1,A9)
  802 FORMAT (2I15)
  803 FORMAT (3F15.6)
C     FORMATS DEMO3 STOP
C
C     ************************************************************
C     *                                                          *
C     *  I/O ASSIGNMENTS.                                        *
C     *                                                          *
C     ************************************************************
      LI=LOGLU(ISESS)
      LO=6
      LD=8
      LP=11
      WRITE (LI, 101)
      WRITE (LO, 601)
C
C     ************************************************************
C     *                                                          *
C     *  INITIALIZE THE PLOTTER.                                 *
C     *                                                          *
C     ************************************************************
      IF (IPLOT) CALL INITG (LP)
      WRITE (LI, 102)
C
C     ************************************************************
C     *                                                          *
C     *  VARIATION OF POLYNOMIAL DEGREE.  LOOP STARTS HERE.      *
C     *                                                          *
C     ************************************************************
      M1STRT=1
      M1STOP=6
      M2STRT=1
      M2STOP=6
      DO 60 MORDR1=M1STRT,M1STOP,1
      DO 60 MORDR2=M2STRT,M2STOP,1
C
C     ************************************************************
C     *
```

```
C     :  READ MEASURED DATA FROM TAPE 9 AND SIMULTANEOUSLY SEARCH THE   :
C     :  MINIMUM AND MAXIMUM VALUE OF EACH X1, X2 AND Y (REQUIRED TO    :
C     :  GET THE PLOT WITHIN THE BOUNDARIES OF A SHEET OF PAPER).       :
C     :                                                                 :
C     :.................................................................:
      REWIND LO
      X1MIN = +99999.
      X1MAX = -99999.
      X2MIN = +99999.
      X2MAX = -99999.
      YMIN  = +99999.
      YMAX  = -99999.
      IVAL=10H
   01 READ  (LO, 801) IVAL,IRAW
      IF (IVAL.EQ.ICNT) GO TO 01
      DECODE (30,802,IRAW) NPNTS1,NPNTS2
      DO 03 I1=1,NPNTS1,1
      DO 03 I2=1,NPNTS2,1
      IVAL=10H
   02 READ  (LO, 801) IVAL,IRAW
      IF (IVAL.EQ.ICNT) GO TO 02
      DECODE (45,803,IRAW) X1(I1,I2),X2(I1,I2),Y(I1,I2)
      IF (X1(I1,I2).LT.X1MIN) X1MIN=X1(I1,I2)
      IF (X2(I1,I2).LT.X2MIN) X2MIN=X2(I1,I2)
      IF (Y(I1,I2).LT.YMIN )  YMIN =Y(I1,I2)
      IF (X1(I1,I2).GT.X1MAX) X1MAX=X1(I1,I2)
      IF (X2(I1,I2).GT.X2MAX) X2MAX=X2(I1,I2)
      IF (Y(I1,I2) .GT.YMAX )  YMAX =Y(I1,I2)
   03 CONTINUE
      WRITE (LI, 103)

C     :.................................................................
C     :  DEFINE PLOTTER AND USER AREAS;  DRAW AXES.                     :
C     :                                                                 :
C     :.................................................................:
      X0          = 14.0
      Y0          =  3.0
      X1PMIN      =   .0
      X1PMAX      = 10.0
      X2PMIN      =   .0
      X2PMAX      = 10.0
      YPMIN       =   .0
      YPMAX       = 10.0
      X1UMIN      = -1.0
      X1UMAX      =  1.0
      X2UMIN      =  -.1
      X2UMAX      =   .0
      YUMIN       =-13.0
      YUMAX       = 13.0
      ALFAX1      = 30.0
      ALFAX2      =150.0
      ALFAY       = 90.0
      PLOTR(61)   = (ALFAX1*PI)/180.
      PLOTR(62)   = (ALFAX2*PI)/180.
      PLOTR(63)   = (ALFAY *PI)/180.
      PLOTR(64)   = X0
      PLOTR(65)   = Y0
      PLOTR(66)   = (X1PMAX       -X1PMIN        )/(X1UMAX-X1UMIN)
      PLOTR(67)   = (X1PMIN*X1UMAX-X1PMAX*X1UMIN)/(X1UMAX-X1UMIN)
      PLOTR(68)   = (X2PMAX       -X2PMIN        )/(X2UMAX-X2UMIN)
      PLOTR(69)   = (X2PMIN*X2UMAX-X2PMAX*X2UMIN)/(X2UMAX-X2UMIN)
      PLOTR(70)   = ( YPMAX       - YPMIN        )/( YUMAX-YUMIN )
      PLOTR(71)   = ( YPMIN*YUMAX - YPMAX*YUMIN  )/( YUMAX-YUMIN )
      X1L         = (X1PMAX       -X1PMIN        )
      X2L         = (X2PMAX       -X2PMIN        )
      YL          = ( YPMAX       - YPMIN        )
      WRITE (LI, 104) X1PMIN,X1PMAX,X2PMIN,X2PMAX,YPMIN,YPMAX,X1UMIN,X1U
     *MAX,X2UMIN,X2UMAX,YUMIN,YUMAX
      IF (IPLOT) CALL AXES (X0,Y0,X1L,ALFAX1,2HX1,+2,X1UMIN,X1UMAX,4HF7.
     *3,7,6)
      IF (IPLOT) CALL AXES (X0,Y0,X2L,ALFAX2,2HX2,+2,X2UMIN,X2UMAX,4HF7.
     *3,7,6)
      IF (IPLOT) CALL AXES (X0,Y0,YL ,ALFAY ,2HY ,-2,YUMIN ,YUMAX ,4HF7.
     *3,7,6)
      WRITE (LI, 105)

C     :.................................................................
C     :  DRAW MEASURSMENT DATA SURFACE.                                 :
C     :                                                                 :
```

```
C
      DO 05 I1=1,NPNTS1,1
      DO 04 I2=1,NPNTS2,1
      IF (IPLOT) CALL THRTY (XPLOT,YPLOT,X1(I1,I2),X2(I1,I2),Y(I1,I2))
      IF (IPLOT .AND. I2.EQ.1) CALL PLOT (XPLOT,YPLOT,3)
   04 IF (IPLOT .AND. I2.GT.1) CALL PLOT (XPLOT,YPLOT,2)
      DO 05 I2=1,NPNTS2,1
      DO 05 I1=1,NPNTS1,1
      IF (IPLOT) CALL THRTY (XPLOT,YPLOT,X1(I1,I2),X2(I1,I2),Y(I1,I2))
      IF (IPLOT .AND. I1.EQ.1) CALL PLOT (XPLOT,YPLOT,3)
   05 IF (IPLOT .AND. I1.GT.1) CALL PLOT (XPLOT,YPLOT,2)
      WRITE (LI, 106)

C     ****************************************************************
C     *                                                              *
C     *  CALCULATE SURFACE FIT THROUGH DATA POINTS.                  *
C     *                                                              *
C     ****************************************************************
   06 WRITE (LI, 107)
C     READ (  9,    *) NORDR1,NORDR2,IPRINT,ISTAT
      NORDR1=NORDR1
      NORDR2=NORDR2
      IPRINT=0
      ISTAT=2
      L=NORDR1+1
      M=NORDR2+1
      IF (L.LT.1 .OR. L.GT.7) GO TO 06
      IF (M.LT.1 .OR. M.GT.7) GO TO 06
      IF (L*M.LT.1 .OR. L*M.GT.40) GO TO 06
      CALL MATS (NPNTS1,NPNTS2,L,M,COEF,IPRINT,ISTAT,SUM1,IERROR)
      IF (IERROR.LT.0) GO TO 06
      WRITE (LI, 108)

C     ****************************************************************
C     *                                                              *
C     *  DRAW APPROXIMATED SURFACE.                                  *
C     *                                                              *
C     ****************************************************************
      DO 08 I1=1,NPNTS1,1
      DO 07 I2=1,NPNTS2,1
      YAPP=F3(COEF,X1(I1,I2),X2(I1,I2),NORDR1,NORDR2)
      IF (IPLOT) CALL THRTY (XPLOT,YPLOT,X1(I1,I2),X2(I1,I2),YAPP)
      IF (IPLOT .AND. I2.EQ.1) CALL PLOT (XPLOT,YPLOT,3)
   07 IF (IPLOT .AND. I2.GT.1) CALL PLOT (XPLOT,YPLOT,2)
      DO 08 I2=1,NPNTS2,1
      DO 08 I1=1,NPNTS1,1
      YAPP=F3(COEF,X1(I1,I2),X2(I1,I2),NORDR1,NORDR2)
      IF (IPLOT) CALL THRTY (XPLOT,YPLOT,X1(I1,I2),X2(I1,I2),YAPP)
      IF (IPLOT .AND. I1.EQ.1) CALL PLOT (XPLOT,YPLOT,3)
   08 IF (IPLOT .AND. I1.GT.1) CALL PLOT (XPLOT,YPLOT,2)
      WRITE (LI, 109)
   09 IF (IPLOT) CALL NUPAGE

C     ****************************************************************
C     *                                                              *
C     *  TERMINATE GRAFICS.                                          *
C     *                                                              *
C     ****************************************************************
      IF (IPLOT) CALL STOPG

      STOP 0077
      END
```

```
      SUBROUTINE MAT3 (NPNTS1,NPNTS2,L,M,COEF,IPRINT,ISTAT,SUBM1,IERROR)
*************************************************************************
*                                                                       *
*   PRESET SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B FOR A           *
*   3D-POLYNOMIAL APPROXIMATION.  THE LINEAR EQUATION SYSTEM THEN       *
*   IS SOLVED BY USING A GAUSS JORDAN ELIMINATION.                      *
*                                                                       *
*   AUTHOR:     HANS ZEANER                                             *
*   DATE:       DECEMBER 31, 1980                                       *
*                                                                       *
*   VARIABLE    EXPLANATION                                  TYPE       *
*   ********    ***************************************              *
*   NPNTS1      NUMBER OF DATA POINTS AT CONSTANT X1         INTEGER    *
*   NPNTS2      NUMBER OF DATA POINTS AT CONSTANT X2         INTEGER    *
*   COEF(49)    POLYNOMIAL COEFFICIENTS                      REAL       *
*   L           NUMBER OF COEFFICIENTS FOR THE POLY-         INTEGER    *
*               NOMIAL IN X1 DIRECTION.                                 *
*   M           NUMBER OF COEFFICIENTS FOR THE POLY-         INTEGER    *
*               NOMIAL IN X2 DIRECTION.                                 *
*   IPRINT      CONTROL QUANTITY OF PRINT OUT                INTEGER    *
*               =2 PRINT EQUATION SYSTEM BEFORE                         *
*                  AND AFTER THE GAUSS JORDAN                           *
*                  ELIMINATION.                                         *
*               =1 PRINT EQUATION SYSTEM AFTER                          *
*                  THE GAUSS JORDAN ELIMINATION.                        *
*               >0 PRINT BASIC EQUATIONS WITH                           *
*                  ACTUAL PARAMETERS.                                   *
*               <0 NO OUTPUT AT ALL.                                    *
*   ISTAT       CONTROL STATISTSICS TO BE PRINTED            INTEGER    *
*               /ISTAT/=2 PRINT ORDINATES OF DATA                       *
*                         AND APPROXIMATED POINTS                       *
*                         AND DEVIATIONS (ERRORS).                      *
*               /ISTAT/=1 ONLY PRINT ERRORS.                            *
*               ISTAT>0   GET ABSOLUTE ERROR.                           *
*               ISTAT<0   GET RELATIVE ERROR.                           *
*   SUBM1(M,M)  SUBMATRIX 1. CLASS IN THE SYSTEM             REAL       *
*               MATRIX.                                                 *
*   IERROR      ERROR FLAG RETURNED TO THE CALLING           INTEGER    *
*               PROGRAM.                                                *
*               =0 NO ERROR DETECTED, GOOD JOB.                         *
*               =1 AN ERROR OCCURED IN THE GAUSSIAN                     *
*                  ELIMINATION.  COEFFICIENTS ARE NOT                   *
*                  ACCURATE.  REVIEW PARAMETERS.                        *
*                                                                       *
*   X1(10,10)   X1-COORDINATES OF THE DATA POINTS            REAL       *
*   X2(10,10)   X2-COORDINATES OF THE DATA POINTS            REAL       *
*   Y(10,10)    Y-COORDINATES OF THE DATA POINTS             REAL       *
*   A(49,49)    SYSTEM MATRIX.                               REAL       *
*   B(49)       RIGHT HAND SIDE VECTOR.                      REAL       *
*                                                                       *
*   IMPORTANT:  THE DATA POINTS X1, X2 AND Y HAVE TO BE PROVIDED        *
*               THROUGH THE LABELED COMMON BLOCK DATA3 AND THE          *
*               SYSTEM MATRIX A AND THE RIGHT HAND SIDE VECTOR B        *
*               ARE AVAILABLE THROUGH THE LABELED COMMON BLOCK          *
*               AB3.  THIS SAVES A LOT OF THAT PRECIOUS THING           *
*               CALLED MEMORY.                                          *
*                                                                       *
*************************************************************************
      COMMON / DATA3 / X1,X2,Y
      COMMON / AB3   / A,B

      REAL X1(10,10),X2(10,10),Y(10,10)
      REAL A(49,49),B(49)

      REAL COEF(49),SUBM1(M,M),STAT(10,10)

C     FORMATS MAT3 START
  101 FORMAT (///7H  AFTER,I3,41H STEPS NO SOLUTION WITHIN THE ACCURACY
     *OF,E10.3,11H )  ( QUIT.//)
  601 FORMAT (///51H  ORDER OF APPROXIMATION POLYNOMIAL IN X1 DIRECTION,
     *9X,I3/51H  ORDER OF APPROXIMATION POLYNOMIAL IN X2 DIRECTION,9X,I3
     *//40H  NUMBER OF DATA POINTS WITH CONSTANT X1,10X,I3/40H  NUMBER O
     *F DATA POINTS WITH CONSTANT X2,10X,I3/)
  602 FORMAT (///62H  SYSTEM MATRIX A AND VECTOR B BEFORE GAUSS JORDAN E
     *LIMINATION/)
  603 FORMAT (19X,13(8X,I2))
  604 FORMAT (2X,I2,13(1X,1PE9.2),1X/((4X,13(1X,1PE9.2))))
  605 FORMAT (///50H  EQUATION SYSTEM AFTER GAUSS JORDAN ELIMINATION//)
  606 FORMAT (///24H  COEFFICIENTS COEF(I,J)//)
  607 FORMAT (12X,3HJ =,I2,11(8X,I2),I7/ 3(7X,[illegible](8X,I2)/))
  608 FORMAT (2X,3HI =,I2,L2(1X,E9.3),1X/(1[illegible]L2(1X,E9.3))/)
  609 FORMAT (///30H  ORDINATES OF DATA POINTS  -  AS GIVEN//)
  610 FORMAT (///40H  ORDINATES OF DATA POINTS  -  AS APPROXIMATED//)
  611 FORMAT (///30H  ABSOLUTE ERROR AT EACH POINT//)
```

```
  612 FORMAT (///35H  RELATIVE ERROR AT EACH POINT IN S/)
  613 FORMAT (///21H  STATISTICS SUMMARY///7X,13HAVERAGE ERROR,13X,1PE9.
     *2/7X,19HMAXIMUM ERROR AT I=,I3,3X,2HJ=,I3,1X,1PE9.2)
  614 FORMAT (///21H  STATISTICS SUMMARY//7X,13HAVERAGE ERROR,13X,1PE9.
     *2,2H S/7X,19HMAXIMUM ERROR AT I=,I3,3X,2HJ=,I3,1X,1PE9.2,2H S)
  619 FORMAT (///7H  AFTER,I3,41H STEPS NO SOLUTION WITHIN THE ACCURACY
     *OF,E10.3,11H  I QUIT./)
C     FORMATS MAT3 STOP
      LI=LOGLU(ISESSN)
      LO=0
      IERROR=1
      ITMAX=50
      EPSLON=.100E-07
C
C     PRINT APPROXIMATION PARAMETERS, IF IPRINT IS GREATER THAN 0.
C
      NORDR1=L-1
      NORDR2=M-1
      IF (IPRINT.LT.0) GO TO 01
      WRITE (LO, 601) NORDR1,NORDR2,NPNTS1,NPNTS2
C
C     START TO PRESET EDGE SECTION SUBMATRICES 1. CLASS.
C
   01 IPOWR1=0
      L1=2*L-1
      DO 05 K=1,L1,1
      IF (K.GT.L) GO TO 02
C
C     TOP EDGE SECTION SUBMATRICES 1. CLASS.
C
      ISTART=1
      ISTOP=M
      JSTART=(K-1)*M+1
      JSTOP=K*M
      GO TO 03
C
C     RIGHT HAND EDGE SECTION SUBMATRICES 1. CLASS.
C
   02 ISTART=(K-L)*M+1
      ISTOP=(K-L+1)*M
      JSTART=(L-1)*M+1
      JSTOP=L*M
C
C     PRESET EDGE SECTION SUBMATRICES 1. CLASS IN SYSTEM MATRIX USING MAT31.
C
   03 CALL MAT31 (NPNTS1,NPNTS2,M,IPOWR1,SUBM1)
      I1=0
      DO 04 I=ISTART,ISTOP,1
      I1=I1+1
      J1=0
      DO 04 J=JSTART,JSTOP,1
      J1=J1+1
   04 A(I,J)=SUBM1(I1,J1)
   05 IPOWR1=IPOWR1+1
C
C     COPY DEFINED SUBMATRICES 1. CLASS DIOGONALLY.
C
      L1=L-1
      DO 07 I1=1,L1,1
      DO 07 J1=2,L,1
      ISTART=(I1-1)*M+1
      ISTOP=I1*M
      JSTART=(J1-1)*M+1
      JSTOP=J1*M
      DO 06 I=ISTART,ISTOP,1
      DO 06 J=JSTART,JSTOP,1
   06 A(I+M,J-M)=A(I,J)
   07 CONTINUE
C
C     PRESET RIGHT HAND SIDE VECTOR B.
C
      I=0
      DO 08 L1=1,L,1
      IPOWR1=L1-1
      DO 08 M1=1,M,1
      IPOWR2=M1-1
      I=I+1
   08 B(I)=S3(NPNTS1,NPNTS2,IPOWR1,IPOWR2,1)
      M1=L*M
C
C     PRINT SYSTEM MATRIX AND RIGHT HAND SIDE VECTOR B, IF IPRINT
C     IS SET TO 2.
C
      IF (IPRINT.NE.2) GO TO 10
```

```
      WRITE (LO, 602)
      WRITE (LO, 603) (J,J=1,N1,1)
      DO 09 I=1,N1,1
   09 WRITE (LO, 604) I,(A(I,J),J=1,N1,1),B(I)
      WRITE (LO, 603) (J,J=1,N1,1)
      IPRINT=1
C
C     GAUSS JORDAN ELIMINATION.
C
   10 EPSLON=EPSLON*1
      CALL ISL3 (N1,ITER,ITRMAX,EPSLON)
      IF (ITER.LT.0) GO TO 27
C
C     PRINT EQUATION SYSTEM, IF IPRINT IS SET TO 1.
C
      IF (IPRINT.NE.1) GO TO 12
      WRITE (LO, 605)
      WRITE (LO, 603) (J,J=1,N1,1)
      DO 11 I=1,N1,1
   11 WRITE (LO, 604) I,(A(I,J),J=1,N1,1),B(I)
      WRITE (LO, 603) (J,J=1,N1,1)
C
C     DEFINE COEFFICIENTS AND PRINT THEM, IF IPRINT IS GREATER THAN 0.
C
   12 CONTINUE
      DO 13 L1=1,N1,1
   13 COEF(L1)=B(L1)
      IF (IPRINT.LT.0) GO TO 15
      WRITE (LO, 606)
      WRITE (LO, 607) (I2,I2=1,N,1)
      DO 14 I1=1,L,1
      ICSTRT=(I1-1)*N+1
      ICSTOP=I1*N
   14 WRITE (LO, 608) I1,(COEF(IC),IC=ICSTRT,ICSTOP,1)
      WRITE (LO, 607) (I2,I2=1,N,1)
C
C     STATISTICS, IF ISTAT ANYTHING ELSE BUT 0.
C
   15 IF (ISTAT.EQ.0) GO TO 20
      IF (ISTAT.LT.0) IERROR=1
      IF (ISTAT.GT.0) IERROR=2
      ISTAT=IABS(ISTAT)
C
C     COMPUTE ABSOLUTE ERROR.
C     ( =(APPROXIMATED VALUE  -  GIVEN VALUE) )
C
      DO 16 I1=1,NPNTS1,1
      DO 16 I2=1,NPNTS2,1
   16 STAT(I1,I2)=P3(COEF,X1(I1,I2),X2(I1,I2),NORDR1,NORDR2)-Y(I1,I2)
C
C     COMPUTE RELATIVE ERROR, IF ISTAT WAS LESS THAN ZERO, I.E. IERROR=1.
C     ( =(APPROXIMATED VALUE  -  GIVEN VALUE)/(GIVEN VALUE) )
C
      IF (IERROR.NE.1) GO TO 18
      DO 17 I1=1,NPNTS1,1
      DO 17 I2=1,NPNTS2,1
   17 STAT(I1,I2)=STAT(I1,I2)/Y(I1,I2)
C
C     ERROR STATISTICS:  COMPUTE AVERAGE ERROR AND FIND LOCATION
C     OF MAXIMUM ERROR.
C
   18 ERRAVE=0.0
      ERRMAX=ABS(STAT(1,1))
      IMAX=1
      JMAX=1
      DO 19 I1=1,NPNTS1,1
      DO 19 I2=1,NPNTS2,1
      IF (ABS(STAT(I1,I2)).LE.ERRMAX) GO TO 19
      ERRMAX=STAT(I1,I2)
      IMAX=I1
      JMAX=I2
   19 ERRAVE=ERRAVE+ABS(STAT(I1,I2))
      ERRAVE=ERRAVE/(NPNTS1*NPNTS2)
      IF (ISTAT.NE.2) GO TO 22
C
C     PRINT ORDINATES OF DATA POINTS AND APPROXIMATED VALUES,
C     IF /ISTAT/ IS EQUAL 2.
C
      WRITE (LO, 609)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      DO 20 I1=1,NPNTS1,1
   20 WRITE (LO, 604) I1,(Y(I1,I2),I2=1,NPNTS2,1)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      WRITE (LO, 610)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
```

```
      DO 21 I1=1,NPNTS1,1
   21 WRITE (LO, 604) I1,(F3(COEF,X1(I1,I2),X2(I1,I2),NORDR1,NORDR2),I2=
     *1,NPNTS2,1)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      ISTAT=1
C
C     PRINT ERRORS, IF /ISTAT/ WAS 1 OR 2.
C
   22 IF (ISTAT.NE.1) GO TO 26
      IF (IERROR.EQ.1) GO TO 24
C
C     PRINT ABSOLUTE ERRORS.
C
      WRITE (LO, 611)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      DO 23 I1=1,NPNTS1,1
   23 WRITE (LO, 604) I1,(STAT(I1,I2),I2=1,NPNTS2,1)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      GO TO 26
C
C     PRINT RELATIVE ERRORS.
C
   24 WRITE (LO, 612)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      DO 25 I1=1,NPNTS1,1
   25 WRITE (LO, 604) I1,(STAT(I1,I2),I2=1,NPNTS2,1)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
C
C     PRINT ERROR STATISTICS.
C
   26 IF (IERROR.EQ.2) WRITE (LO, 613) ERRAVE,IMAX,JMAX,ERRMAX
      IF (IERROR.EQ.1) WRITE (LO, 614) ERRAVE,IMAX,JMAX,ERRMAX
      GO TO 29
C
C     FATAL ERROR DETECTED.  OUTPUT ERROR MESSAGE AN PRESET COEFFICIENTS.
C
   27 ITER=-ITER
      IERROR=-IERROR
      B(1)=Y(1,1)
      DO 28 IC=2,NL,1
   28 B(IC)=0.0
      WRITE (LO, 101) ITER,EPSLON
      WRITE (LO, 615) ITER,EPSLON
      GO TO 12
C
C     RETURN.
C
   29 RETURN
      END
```

```
      SUBROUTINE MAT31 (NPNTS1,NPNTS2,M,IPOWR1,SUBM1)
C     ****************************************************************
C     *                                                              *
C     *  PRESET SUBMATRIX 1. CLASS SUBM1 IN THE SYSTEM MATRIX A FOR A*
C     *  3D-POLYNOMIAL APPROXIMATION.                                *
C     *                                                              *
C     *  AUTHOR:    HANS ZEANER                                       *
C     *  DATE:      DECEMBER 31, 1980                                 *
C     *                                                              *
C     *  VARIABLE   EXPLANATION                             TYPE     *
C     *  ********   ************************************    *******  *
C     *  NPNTS1     NUMBER OF DATA POINTS AT CONSTANT X1    INTEGER  *
C     *  NPNTS2     NUMBER OF DATA POINTS AT CONSTANT X2    INTEGER  *
C     *  M          NUMBER OF COEFFICIENTS FOR THE POLY-    INTEGER  *
C     *             NOMIAL IN X2 DIRECTION.                          *
C     *  IPOWR1     POWER FOR THE X1 VALUES IN THE          INTEGER  *
C     *             SUMMATION.                                       *
C     *  SUBM1(M,M) SUBMATRIX 1. CLASS IN THE SYSTEM        REAL     *
C     *                                                              *
C     ****************************************************************
      REAL SUBM1(M,M)
      IPOWR2=0
      M1=2*M-1
      DO 03 K=1,M1,1
      IF (K.GT.M) GO TO 01
C
C     TOP EDGE SECTION ELEMENTS.
C
      I=1
      J=K
      GO TO 02
C
C     RIGHT HAND EDGE SECTION ELEMENTS.
C
   01 I=(K-M)+1
      J=M
C
C     PRESET EDGE SECTION ELEMENTS IN SUBMATRIX 1. CLASS USING S3.
C
   02 SUBM1(I,J)=S3(NPNTS1,NPNTS2,IPOWR1,IPOWR2,0)
   03 IPOWR2=IPOWR2+1
C
C     COPY DEFINED ELEMENTS DIAGONALLY.
C
      M1=M-1
      DO 04 I=1,M1,1
      DO 04 J=2,M,1
   04 SUBM1(I+1,J-1)=SUBM1(I,J)
      RETURN
      END
```

```
76/176   OPT=2   PMDMP                         FTN [illegible]          [illegible]          PAGE    1

      REAL FUNCTION S3 (NPNTS1,NPNTS2,IPOWR1,IPOWR2,IY)
C     ***************************************************************
C     *                                                             *
C     *  PERFORM SUMMATIONS TO PRESET THE ELEMENTS A(I,J) IN THE    *
C     *  SYSTEM MATRIX A AND THE ELEMENTS B(I) IN THE RIGHT HAND    *
C     *  SIDE VECTOR B FOR THE LINEAR EQUATION SYSTEM, THAT MUST    *
C     *  BE SOLVED FOR A POLYNOMIAL 3D-APPROXIMATION.               *
C     *                                                             *
C     *  AUTHOR:      HANS ZEGNER                                   *
C     *  DATE:        DECEMBER 31, 1980                             *
C     *                                                             *
C     *  VARIABLE     EXPLANATION                          TYPE     *
C     *  ********     **********************************   *******  *
C     *  NPNTS1       NUMBER OF DATA POINTS AT CONSTANT X1  INTEGER  *
C     *  NPNTS2       NUMBER OF DATA POINTS AT CONSTANT X2  INTEGER  *
C     *  IPOWR1       POWER FOR THE X1 VALUES IN THE       INTEGER  *
C     *               SUMMATION.                                    *
C     *  IPOWR2       POWER FOR THE X2 VALUES IN THE       INTEGER  *
C     *               SUMMATION.                                    *
C     *  IY           CONTROL PARAMETER                    INTEGER  *
C     *               =0  Y VALUES ARE NOT INCLUDED IN THE          *
C     *                   SUMMATION.                                *
C     *                   (-> SYSTEM MATRIX ELEMENTS)               *
C     *               =1  Y VALUES ARE INCLUDED IN THE     INTEGER  *
C     *                   SUMMATION.                                *
C     *                   (-> RIGHT HAND SIDE VECTOR ELEM.)         *
C     *  S3           RESULT OF SUMMATION.                 REAL     *
C     *                                                             *
C     *  X1(16,16)    X1-COORDINATES OF THE DATA POINTS    REAL     *
C     *  X2(16,16)    X2-COORDINATES OF THE DATA POINTS    REAL     *
C     *  Y(16,16)     Y-COORDINATES OF THE DATA POINTS     REAL     *
C     *                                                             *
C     *  IMPORTANT:   THE DATA POINTS X1, X2 AND Y HAVE TO BE PROVIDED
C     *               THROUGH THE LABELED COMMON BLOCK DATA3.       *
C     *                                                             *
C     ***************************************************************
      COMMON / DATA3 / X1,X2,Y
      REAL X1(16,16),X2(16,16),Y(16,16)
      S=0.
      DO 05 K1=1,NPNTS1,1
      DO 05 K2=1,NPNTS2,1
      IF (IPOWR1.EQ.0) GO TO 01
      S1=X1(K1,K2)**IPOWR1
      GO TO 02
   01 S1=1.
   02 IF (IPOWR2.EQ.0) GO TO 03
      S2=X2(K1,K2)**IPOWR2
      GO TO 04
   03 S2=1.
   04 SY=Y(K1,K2)
      IF (IY.EQ.0) SY=1.
   05 S=S+S1*S2*SY
      S3=S
      RETURN
      END
```

```
      INTEGER FUNCTION IEL33 (N)
C     ****************************************************************
C     *                                                              *
C     *  GAUSS JORDAN ELIMINATION WITH MATRIX A AND VECTOR B.  THE   *
C     *  VARIABLE IEL3 IS SET TO 1, IF THE ELIMINATION IS COMPLETED. *
C     *  IEL3 LESS THAN 0 INDICATES AN ERROR.                        *
C     *                                                              *
C     *  AUTHOR:     HANS ZEHNER                                     *
C     *  DATE:       DECEMBER 31, 1980                               *
C     *                                                              *
C     *  VARIABLE    EXPLANATION                        TYPE         *
C     *  ********    ******************************     *******      *
C     *  N           NUMBER OF EQUATIONS                INTEGER      *
C     *  IEL3        CONTROL PARAMETER, RETURNED TO THE INTEGER      *
C     *              CALLING ROUTINE.                                *
C     *              =1 GAUSS JORDAN ELIMINATION WAS                 *
C     *                 PERFORMED CORRECTLY.                         *
C     *              <0 ERROR ENCOUNTERED DURING GAUSS               *
C     *                 JORDAN ELIMINATION.                          *
C     *                                                              *
C     *  A(49,49)    SYSTEM MATRIX A                    REAL         *
C     *  B(49)       RIGHT HAND SIDE VECTOR B           REAL         *
C     *                                                              *
C     *  IMPORTANT:  MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH
C     *              THE LABELED COMMON BLOCK ABS.  BOTH A AND B WILL
C     *              WILL UNDERGO CHANGES DURING EXECUTION, I.E. THE *
C     *              ELIMINATION.                                    *
C     *                                                              *
C     ****************************************************************
      COMMON / ABS  / A,B
      REAL A(49,49),B(49)
      IEL3=1
      IEL33=IEL3
      EPS=1.00E-030
      DO 06 I=1,N+1
      DO 02 K=1,N+1
      IF (K.EQ.I) GO TO 02
      CONST=-A(K,I)/A(I,I)
      DO 01 J=1,N+1
      A(K,J)=A(K,J)+CONST*A(I,J)
      IF (J.EQ.I) A(K,J)=0.0
   01 CONTINUE
      B(K)=B(K)+CONST*B(I)
   02 CONTINUE
      CONST=A(I,I)
      DO 03 J=1,N+1
   03 A(I,J)=A(I,J)/CONST
      A(I,I)=1.
      B(I)=B(I)/CONST
      DO 06 J=1,N+1
      IERROR=0
      DO 04 K=1,N+1
      IF (ABS(A(J,K)).GT.EPS) GO TO 04
      IERROR=IERROR+1
   04 CONTINUE
      IF (ABS(B(J)).GT.EPS) GO TO 05
      IERROR=IERROR+1
   05 IF (IERROR.GT.N+1) GO TO 07
   06 CONTINUE
      GO TO 08
   07 IEL3=-1
      IEL33=IEL3
   08 RETURN
      END
```

```
      SUBROUTINE IEL3 (N,ITER,ITMAX,EPSLON)
C     ***********************************************************
C     *                                                         *
C     *  GAUSSIAN ELIMINATION WITH LINEAR EQUATION SYSTEM V (THE RIGHT
C     *  HAND SIDE VECTOR B IS INTEGRATED TOGETHER WITH THE SYSTEM MA-
C     *  TRIX A IN THE MATRIX V). ITER IS THE NUMBER OF ITERATIONS RE-
C     *  TURNED TO THE CALLING PROGRAM. THE ITERATION TO IMPROVES THE
C     *  SOLUTION SO MANY TIMES, UNTIL THE REMAINING ERROR IS WITHIN
C     *  THE MAGNITUDE OF EPSLON. SHOULD THE NUMBER OF ITERATIONS EX-
C     *  CEED ITMAX, THE SUBROUTINE TERMINATES THE CALCULATION AND
C     *  SETS ITER TO A NEGATIVE VALUE.
C     *
C     *  AUTHORS:   KONRAD VOGELER, HANS ZESHER
C     *  DATE:      DECEMBER 11, 1980
C     *
C     *  VARIABLE   EXPLANATION                            TYPE
C     *  ********   ***********************************    *******
C     *  N          NUMBER OF EQUATIONS                    INTEGER
C     *  ITER       NUMBER OF ITERATIONS TO BE RETURNED    INTEGER
C     *             >0 GAUSSIAN ELIMINATION PERFORMED
C     *                CORRECTLY.
C     *             <0 NO CONVERGENCE OF THE SOLUTION FOR
C     *                THE DESIRED RESOLUTION EPSLON.
C     *                ITERATION DID TERMINATE AFTER
C     *                ITMAX ITERATION STEP.
C     *  ITMAX      MAXIMUM NUMBER OF ITERATION STEPS      INTEGER
C     *  EPSLON     DESIRED RESOLUTION OF THE SOLUTION     REAL
C     *
C     *  A(49,49)   SYSTEM MATRIX A                        REAL
C     *  B(49)      RIGHT HAND SIDE VECTOR B               REAL
C     *
C     *  IMPORTANT: MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH
C     *             THE LABELED COMMON BLOCK AB3. BOTH A AND B WILL
C     *             WILL UNDERGO CHANGES DURING EXECUTION, I.E. THE
C     *             ELIMINATION.
C     *
C     ***********************************************************
      COMMON / AB3 / V
      REAL V(49,50),G(50),A(49,50),B(50)
      LOGICAL IPRINT
      DATA IPRINT /.FALSE./
      DATA PI /3.14159265/
C     FORMATS IEL3 START
  601 FORMAT (///39H  MATRIX A UND VEKTOR B VOR ELIMINATION//)
  602 FORMAT ((4X,13(8X,I2)))
  603 FORMAT [illegible]
  604 FORMAT (///23H  PIVOTISIERTES SYSTEM (N=,I3,19H, ITERATIONSSCHRITT/
     *)
  605 FORMAT (///2X,I2,20H KOEFFIZIENTEN AUS GAUSS//)
  606 FORMAT [illegible]
  607 FORMAT (///39H  KLEINSTER ZEILENVEKTOR(MITTWINKEL =,E12.3)
  608 FORMAT (27X,12H  RESIDIUM =,E12.3,5H  NACH,I3,12H ITERATIONEN)
  609 FORMAT [illegible]
  610 FORMAT (///23H  SYSTEM NACH ELIMINATION//)
C     FORMATS IEL3 STOP
      LO=6
      M=N+1
      K=N-1
C
C     REARRANGE VECTOR B IN ARRAY V, IF N IS LESS THAN 49.
C
      IF (N.EQ.49) GO TO 02
      DO 01 I=1,N
   01 V(I,M)=V(I,50)
C
C     PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B BEFORE THE
C     ELIMINATION, IF IPRINT IS SET .TRUE. .
C
   02 IF (IPRINT) WRITE (LO, 601)
      IF (IPRINT) WRITE (LO, 602) (J,J=1,M)
      DO 03 I=1,N
   03 IF (IPRINT) WRITE (LO, 603) I,(V(I,J),J=1,M)
      IF (IPRINT) WRITE (LO, 602) (J,J=1,M)
C
C     LINEAR EQUATION SYSTEM  V(N,N)*G(N)=V(N,N+1)
C
      DO 04 I=1,N
      DO 04 J=1,M
   04 A(I,J)=V(I,J)
      DO 05 I=1,N
   05 B(I)=0.0
C
C     CHECK FOR STABILITY.
```

```
C
      FI=.0
      DO 06 I=1,N,1
   06 G(I)=.0
      DO 08 I=1,N,1
      DO 07 J=1,N,1
   07 G(I)=G(I)+V(I,J)**2
   08 G(I)=SQRT(G(I))
      DO 10 I=1,K,1
      JJ=I+1
      DO 10 J=JJ,N,1
      SR=.0
      DO 09 I3=1,N,1
   09 SR=SR+V(I,I3)*V(J,I3)
      F=SR/(G(I)*G(J))
      IF (ABS(F).GT.ABS(FI)) FI=F
   10 CONTINUE
      FI=ABS(FI)
      FI=ACOS(FI)
      FI=(FI*180.)/PI
C
C     START ITERATION TO FIND AND IMPROVE SOLUTION.
C
      DO 27 LX=1,999,1
      ITER=LX
      DO 15 J=1,K,1
C
C     PIVOTISATION AND GAUSSIAN STEP FOR COLUMN J.
C
      IX=J
      E=V(J,J)
      DO 12 I=J,N,1
      IF (ABS(V(I,J)).GT.ABS(E)) GO TO 11
      GO TO 12
   11 IX=I
      E=V(I,J)
   12 CONTINUE
      IF (IX.EQ.J) GO TO 14
      DO 13 I=1,M,1
      G(I)=V(IX,I)
      V(IX,I)=V(J,I)
   13 V(J,I)=G(I)
      G(M)=.0
   14 MM=J+1
      DO 15 I=MM,N,1
      W=V(I,J)/V(J,J)
      DO 15 KK=J,M
      V(I,KK)=V(I,KK)-V(J,KK)*W
   15 CONTINUE
      I=N
      IF (V(I,I).EQ..0) GO TO 32
C
C     SOLUTION.
C
      DO 16 I=1,N,1
   16 G(I)=.0
      G(N)=V(N,M)/V(N,N)
      DO 19 II=1,K,1
      I=N-II
      IF (V(I,I).EQ..0) GO TO 32
      G(I)=V(I,M)
      DO 17 JJ=1,K,1
      J=N-JJ
      IF (J.EQ.I) GO TO 18
   17 G(I)=G(I)-G(J)*V(I,J)
   18 CONTINUE
   19 G(I)=G(I)/V(I,I)
C
C     CORRECT SOLUTION BY AMOUNT OF ERROR, WHICH WAS INVOLVED IN THIS
C     ITERATION STEP.  TAKE THIS VECTOR AS START VECTOR FOR THE NEXT
C     ITERATION.
C
      DO 20 I=1,N,1
   20 G(I)=G(I)+G(I)
C
C     COMPUTE THE REMAINING ERROR.
C
      DO 22 I=1,N,1
      V(I,M)=.0
      DO 21 J=1,N,1
   21 V(I,M)=V(I,M)+G(J)*A(I,J)
   22 V(I,M)=A(I,M)-V(I,M)
      R=.0
      DO 23 I=1,N,1
   23 R=R+V(I,M)*V(I,M)
```

```
      R=SQRT(R)
      IF (R.LT.EPSLON) GO TO 28
      IF (LX.GT.ITRMAX) GO TO 33
      DO 24 I=1,N+1
   24 B(I)=G(I)
C
C     PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B AFTER THE
C     PIVOTISATION.
C
      IF (IPR[illegible]) WRITE (L3, 604) LX
      IF (IPR[illegible]) WRITE (L3, 602) (J,J=1,N+1)
      DO 25 I=[illegible],N+1
   25 IF (IPRINT) WRITE (L3, 603) I,(V(I,J),J=1,N+1),G(I)
      IF (IPRINT) WRITE (L3, 602) (J,J=1,N+1)
      DO 26 I=1,N+1
      DO 26 J=1,N+1
   26 V(I,J)=A(I,J)
C
C     STOP LOOP TO FIND SOLUTION.
C
   27 CONTINUE
C
C     REARRANGE SOLUTION IN THE ARRAY V.
C
   28 CONTINUE
      DO 29 I=1,N+1
   29 V(I,30)=G(I)
C
C     PRINT SOLUTION, IF IPRINT IS SET .TRUE. .
C
   30 IF (IPRINT) WRITE (L3, 605) V
      IF (IPRINT) WRITE (L3, 505) (I,G(I),I=1,N+1)
      IF (IPRINT) WRITE (L3, 607) R
      IF (IPRINT) WRITE (L3, 508) R,LX
C
C     DRUCKEN DER MATRIX A UND DES VECTORS B NACH ELIMINATION.
C
      IF (IPRINT) WRITE (L3, 610)
      IF (IPRINT) WRITE (L3, 602) (J,J=1,N+1)
      DO 31 I=1,N+1
   31 IF (IPRINT) WRITE (L3, 603) I,(V(I,J),J=1,N+1)
      IF (IPRINT) WRITE (L3, 602) (J,J=1,N+1)
      RETURN
C
C     ERROR ENCOUNTERED.  STOP.
C
   32 ITER=-1
      WRITE (L3, 609) I,I,V(I,I)
      G(I)=G(I)/V(I,I)
      RETURN
C
C     NO CONVERGENCE OF THE SOLUTION.
C
   33 ITER=-ITER
      RETURN
      END
```

```
      REAL FUNCTION F3 (COEF,X1,X2,NORDR1,NORDR2)
C     ****************************************************************
C     *                                                              *
C     * COMPUTE FUNCTION VALUE Y=F(X1,X2) OF A 3D-POLYNOMIAL USING   *
C     * THE HORNER SCHEME IN ORDER TO INCREASE SPEED AND ACCURACY.   *
C     *                                                              *
C     * AUTHOR:    HANS ZEHNER                                       *
C     * DATE:      DECEMBER 31, 1983                                 *
C     *                                                              *
C     * VARIABLE   EXPLANATION                              TYPE     *
C     * ********   **************************************   ****     *
C     * COEF(49)   POLYNOMIAL COEFFICIENTS                  REAL     *
C     * X1         X1-VALUE, WHERE Y WILL BE COMPUTED.      REAL     *
C     * X2         X2-VALUE, WHERE Y WILL BE COMPUTED.      REAL     *
C     * NORDR1     ORDER OF THE POLYNOMIAL IN THE           INTEGER  *
C     *            X1 DIRECTION.                                     *
C     * NORDR2     ORDER OF THE POLYNOMIAL IN THE           INTEGER  *
C     *            X2 DIRECTION.                                     *
C     * F3         ORDINATE OF THE 3D-POLYNOMIAL.           REAL     *
C     *                                                              *
C     ****************************************************************
      REAL COEF(49),D(49)
      NCOEF1=NORDR1+1
      NCOEF2=NORDR2+1
C
C     COMPUTE COEFFICIENTS D BASED ON COEFFICIENTS COEF AND X2.
C
      DO 02 I=1,NCOEF1,1
      IC=I*NCOEF2
      Y=COEF(IC)
      DO 01 J1=1,NORDR2,1
      J=NCOEF2-J1
      IC=(I-1)*NCOEF2+J
   01 Y=COEF(IC)+X2*Y
   02 D(I)=Y
C
C     COMPUTE Y BASED ON COEFFICIENTS D AND X1
C
      Y=D(NCOEF1)
      DO 03 J1=1,NORDR1,1
      J=NCOEF1-J1
   03 Y=D(J)+X1*Y
      F3=Y
      RETURN
      END
```

```
BLOCK DATA AB4        74/175  OPT=2  FTN 4.8+498        06/22/79. 14.25.24       PAGE    1

1                 BLOCK DATA AB4
                  COMMON / AB4   / A,B
                  REAL A(64,64),B(64)
                  END
```

```
      BLOCK DATA DATA4
      COMMON / DATA4 / Z1,Z2,Z3,T
      REAL Z1(9,7,5),Z2(9,7,5),Z3(9,7,5),T(9,7,5)
      END
```

```
      PROGRAM DEMO4 (INPUT,OUTPUT,TAPE1,TAPE6,TAPE7,TAPE9=INPUT)
C     ****************************************************************
C     *                                                              *
C     *  THIS IS A DEMONSTRATION PROGRAM AND IT SHOWS THE CORRECT USE *
C     *  OF THE TPL BINARY LIBRARY USING EXAMPLES.                   *
C     *                                                              *
C     *  AUTHOR:     HANS ZEGNER                                     *
C     *  DATE:       DECEMBER 31, 1980                               *
C     *                                                              *
C     ****************************************************************
      COMMON / AFLD  / PLOTR
      COMMON / DATA4 / X1,X2,X3,Y
      REAL PLOTR(250)
      REAL X1(9,7,5),X2(9,7,5),X3(9,7,5),Y(9,7,5)
      REAL COEF(64),SUBM1(64,64),SUBM2(7,7)
      INTEGER IRAN(6)
      LOGICAL IPLOT
      DATA ICNT  /1UH0                         /
      DATA IPLOT /.TRUE./
      DATA PI    /3.141592654/
C     FORMATS DEMO4 START
  101 FORMAT (////72H   HELLO,  THIS IS PROGRAM DEMO4 AND HERE, SEE, HOW T
     *O USE THE MARVELLOUS/70H  TPL BINARY LIBRARY.  YOUR INTEREST WILL
     *BE GREATLY AWARDED BY EASIER/29H   AND FASTER PROGRAMMING.////)
  102 FORMAT (22H  PLOTTER INITIALIZED.)
  103 FORMAT (31H  DATA POINTS READ FROM TAPE 7.)
  104 FORMAT ([illegible])
  105 FORMAT (47H  PLOTTER AND USER AREA DEFINED AND AXES DRAWN.)
  106 FORMAT (31H  MEASURED DATA SURFACES DRAWN.)
  107 FORMAT (74H  ENTIRE ORDER OF POLYNOMIAL IN X1-, X2- AND X3-DIRECTIO
     *N, IPRINT AND ISTAT)
  108 FORMAT (23H  SURFACE FITS COMPLETED.)
  109 FORMAT (30H  APPROXIMATED SURFACES DRAWN.)
  601 FORMAT (1H1/1H7)
  701 FORMAT (A1,6A13)
  702 FORMAT (3I5[illegible])
  703 FORMAT ([illegible])
C     FORMATS DEMO4 STOP
C     ****************************************************************
C     *                                                              *
C     *  I/O ASSIGNMENTS.                                            *
C     *                                                              *
C     ****************************************************************
      LI=LOGCU(ISESSN)
      LO=6
      LD=7
      LP=13
      WRITE (LI, 101)
      WRITE (LO, 601)
C     ****************************************************************
C     *                                                              *
C     *  INITIALIZE THE PLOTTER.                                     *
C     *                                                              *
C     ****************************************************************
      IF (IPLOT) CALL INITG (LP)
      WRITE (LI, 102)
C     ****************************************************************
C     *                                                              *
C     *  VARIATION OF POLYNOMIAL DEGREE.  LOOP STARTS HERE.          *
C     *                                                              *
C     ****************************************************************
      N1STRT=1
      N1STOP=6
      N2STRT=1
      N2STOP=6
      N3STRT=1
      N3STOP=6
      DO 10 NORDR1=N1STRT,N1STOP,1
      DO 10 NORDR2=N2STRT,N2STOP,1
      DO 10 NORDR3=N3STRT,N3STOP,1
      ITM=(NORDR1+1)*(NORDR2+1)*(NORDR3+1)
      IF (ITM.GT.64) GO TO 10
```

```
C     ****************************************************************
C     *                                                              *
C     *  READ MEASURED DATA FROM TAPE7 AND SIMULTANEOUSLY SEARCH THE *
C     *  MINIMUM AND MAXIMUM VALUE OF EACH X1, X2 AND Y (REQUIRED TO *
C     *  GET THE PLOT WITHIN THE BOUNDARIES OF A SHEET OF PAPER).    *
C     *                                                              *
C     ****************************************************************
      REWIND LO
      IVAL   = 10H
      X1MIN  = +99999.
      X1MAX  = -99999.
      X2MIN  = +99999.
      X2MAX  = -99999.
      YMIN   = +99999.
      YMAX   = -99999.
   01 READ   (LO, 701) IVAL, IRAW
      IF (IVAL.EQ.ICNT) GO TO 01
      DECODE (45,702,IRAW) NPNTS1,NPNTS2,NPNTS3
      DO 03 I1=1,NPNTS1,1
      DO 03 I2=1,NPNTS2,1
      DO 03 I3=1,NPNTS3,1
      IVAL=10H
   02 READ   (LO, 701) IVAL, IRAW
      IF (IVAL.EQ.ICNT) GO TO 02
      DECODE (60,703,IRAW) X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3),Y(I1,I
     *2,I3)
      IF (X1(I1,I2,I3).LT.X1MIN) X1MIN=X1(I1,I2,I3)
      IF (X2(I1,I2,I3).LT.X2MIN) X2MIN=X2(I1,I2,I3)
      IF (Y(I1,I2,I3) .LT.YMIN ) YMIN =Y(I1,I2,I3)
      IF (X1(I1,I2,I3).GT.X1MAX) X1MAX=X1(I1,I2,I3)
      IF (X2(I1,I2,I3).GT.X2MAX) X2MAX=X2(I1,I2,I3)
      IF (Y(I1,I2,I3) .GT.YMAX ) YMAX =Y(I1,I2,I3)
   03 CONTINUE
      WRITE (LI, 103)
C     ****************************************************************
C     *                                                              *
C     *  DEFINE PLOTTER AND USER AREAS;  DRAW AXES.                  *
C     *                                                              *
C     ****************************************************************
      X0          =  15.
      Y0          =   3.
      X1PMIN      =    .0
      X1PMAX      =  10.0
      X2PMIN      =    .0
      X2PMAX      =  10.0
      YPMIN       =    .0
      YPMAX       =  10.0
      X1UMIN      =  X1MIN
      X1UMAX      =  X1MAX
      X2UMIN      =  X2MIN
      X2UMAX      =  X2MAX
      YUMIN       =  YMIN
      YUMAX       =  YMAX
      ALFAX1      =  60.0
      ALFAX2      = 120.0
      ALFAY       = 180.0
      PLOTR(61)   =  (ALFAX1*PI)/180.
      PLOTR(62)   =  (ALFAX2*PI)/180.
      PLOTR(63)   =  (ALFAY *PI)/180.
      PLOTR(64)   =  X0
      PLOTR(65)   =  Y0
      PLOTR(66)   =  (X1PMAX        -X1PMIN        )/(X1UMAX-X1UMIN)
      PLOTR(67)   =  (X1PMIN*X1UMAX-X1PMAX*X1UMIN)/(X1UMAX-X1UMIN)
      PLOTR(68)   =  (X2PMAX        -X2PMIN        )/(X2UMAX-X2UMIN)
      PLOTR(69)   =  (X2PMIN*X2UMAX-X2PMAX*X2UMIN)/(X2UMAX-X2UMIN)
      PLOTR(70)   =  ( YPMAX        - YPMIN        )/( YUMAX-YUMIN )
      PLOTR(71)   =  ( YPMIN*YUMAX  - YPMAX*YUMIN  )/( YUMAX-YUMIN )
      X1L         =  (X1PMAX        -X1PMIN        )
      X2L         =  (X2PMAX        -X2PMIN        )
      YL          =  ( YPMAX        - YPMIN        )
      WRITE (LI, 160) X1PMIN,X1PMAX,X2PMIN,X2PMAX,YPMIN,YPMAX,X1UMIN,X1U
     *MAX,X2UMIN,X2UMAX,YUMIN,YUMAX
      IF (IPLOT) CALL AXES (X0,Y0,X1L,ALFAX1,2HX1,+2,X1UMIN,X1UMAX,4HF7.
     *3,7,4)
      IF (IPLOT) CALL AXES (X0,Y0,X2L,ALFAX2,2HX2,+2,X2UMIN,X2UMAX,4HF7.
     *3,7,4)
      IF (IPLOT) CALL AXES (X0,Y0,YL ,ALFAY ,2HY ,-2,YUMIN ,YUMAX ,4HF7.
     *3,7,4)
      WRITE (LI, 105)
```

```
C     ***************************************************************
C     *                                                             *
C     *  DRAW MEASUREMENT DATA SURFACES.                            *
C     *                                                             *
C     ***************************************************************
      DO 04 I3=1,NPNTS3,1
      DO 04 I2=1,NPNTS2,1
      DO 04 I1=1,NPNTS1,1
      IF (IPLOT) CALL THRTW (XPLOT,YPLOT,X1(I1,I2,I3),X2(I1,I2,I3),Y(I1,
     *I2,I3))
      IF (IPLOT .AND. I2.EQ.1) CALL PLOT (XPLOT,YPLOT,3)
   04 IF (IPLOT .AND. I2.GT.1) CALL PLOT (XPLOT,YPLOT,2)
      DO 05 I3=1,NPNTS3,1
      DO 05 I2=1,NPNTS2,1
      DO 05 I1=1,NPNTS1,1
      IF (IPLOT) CALL THRTW (XPLOT,YPLOT,X1(I1,I2,I3),X2(I1,I2,I3),Y(I1,
     *I2,I3))
      IF (IPLOT .AND. I1.EQ.1) CALL PLOT (XPLOT,YPLOT,3)
   05 IF (IPLOT .AND. I1.GT.1) CALL PLOT (XPLOT,YPLOT,2)
      WRITE (LI, 106)
C
C     ***************************************************************
C     *                                                             *
C     *  CALCULATE SURFACE FITS THROUGH DATA POINTS.                *
C     *                                                             *
C     ***************************************************************
C
   06 WRITE (LI, 107)
      READ  ( 5,    *) NORDR1,NORDR2,NORDR3,IPRINT,ISTAT
      NORDR1=NORDR1
      NORDR2=NORDR2
      NORDR3=NORDR3
      IPRINT=0
      ISTAT=1
      L=NORDR1+1
      M=NORDR2+1
      N=NORDR3+1
      LMN=L*M*N
      IF (L.LT.1 .OR. L.GT.7) GO TO 06
      IF (M.LT.1 .OR. M.GT.7) GO TO 06
      IF (N.LT.1 .OR. N.GT.7) GO TO 06
      IF (LMN.LT.1 .OR. LMN.GT.64) GO TO 06
      CALL DATA (NPNTS1,NPNTS2,NPNTS3,L,M,N,COEF,IPRINT,ISTAT,SUBM1,MN,S
     *UBM2,IERROR)
      IF (IERROR.LT.0) GO TO 09
      WRITE (LI, 108)
C
C     ***************************************************************
C     *                                                             *
C     *  DRAW APPROXIMATED SURFACES.                                *
C     *                                                             *
C     ***************************************************************
      DO 07 I3=1,NPNTS3,1
      DO 07 I1=1,NPNTS1,1
      DO 07 I2=1,NPNTS2,1
      YAPP=F4(COEF,X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3),NORDR1,NORDR2,
     *NORDR3)
      IF (IPLOT) CALL THRTW (XPLOT,YPLOT,X1(I1,I2,I3),X2(I1,I2,I3),YAPP)
      IF (IPLOT .AND. I2.EQ.1) CALL PLOT (XPLOT,YPLOT,3)
   07 IF (IPLOT .AND. I2.GT.1) CALL PLOT (XPLOT,YPLOT,2)
      DO 08 I3=1,NPNTS3,1
      DO 08 I2=1,NPNTS2,1
      DO 08 I1=1,NPNTS1,1
      YAPP=F4(COEF,X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3),NORDR1,NORDR2,
     *NORDR3)
      IF (IPLOT) CALL THRTW (XPLOT,YPLOT,X1(I1,I2,I3),X2(I1,I2,I3),YAPP)
      IF (IPLOT .AND. I1.EQ.1) CALL PLOT (XPLOT,YPLOT,3)
   08 IF (IPLOT .AND. I1.GT.1) CALL PLOT (XPLOT,YPLOT,2)
      WRITE (LI, 109)
   09 IF (IPLOT) CALL NUPAGE
   10 CONTINUE
C
C     ***************************************************************
C     *                                                             *
C     *  TERMINATE GRAFICS.                                         *
C     *                                                             *
C     ***************************************************************
```

```
      IF (IPLOT) CALL STOPG
200   STOP 0077
      END
```

```
      SUBROUTINE MAT4 (NPNTS1,NPNTS2,NPNTS3,L,M,N,COEF,IPRINT,ISTAT,SUBM
     *1,MM,SUBM2,IERROR)
C     ****************************************************************
C     *                                                              *
C     *  PRESET SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B FOR A   *
C     *  4D-POLYNOMIAL APPROXIMATION. THE LINEAR EQUATION SYSTEM THEN*
C     *  IS SOLVED BY USING A GAUSS JORDAN ELIMINATION.              *
C     *                                                              *
C     *  AUTHOR:    HANS ZEAMER                                      *
C     *  DATE:      DECEMBER 31, 1980                                *
C     *                                                              *
C     *  VARIABLE   EXPLANATION                            TYPE      *
C     *  ********   *******************************       *******   *
C     *  NPNTS1     NUMBER OF DATA POINTS AT CONSTANT X1   INTEGER   *
C     *  NPNTS2     NUMBER OF DATA POINTS AT CONSTANT X2   INTEGER   *
C     *  NPNTS3     NUMBER OF DATA POINTS AT CONSTANT X3   INTEGER   *
C     *  COEF(64)   POLYNOMIAL COEFFICIENTS                REAL      *
C     *  L          NUMBER OF COEFFICIENTS FOR THE POLY-   INTEGER   *
C     *             NOMIAL IN X1 DIRECTION.                          *
C     *  M          NUMBER OF COEFFICIENTS FOR THE POLY-   INTEGER   *
C     *             NOMIAL IN X2 DIRECTION.                          *
C     *  N          NUMBER OF COEFFICIENTS FOR THE POLY-   INTEGER   *
C     *             NOMIAL IN X3 DIRECTION.                          *
C     *  IPRINT     CONTROL QUANTITY OF PRINT OUT          INTEGER   *
C     *             =2 PRINT EQUATION SYSTEM BEFORE                  *
C     *                AND AFTER THE GAUSS JORDAN                    *
C     *                ELIMINATION.                                  *
C     *             =1 PRINT EQUATION SYSTEM AFTER                   *
C     *                THE GAUSS JORDAN ELIMINATION.                 *
C     *             =0 PRINT BASIC EQUATIONS WITH                    *
C     *                ACTUAL PARAMETERS.                            *
C     *             <0 NO OUTPUT AT ALL.                             *
C     *  ISTAT      CONTROL STATISTICS TO BE PRINTED       INTEGER   *
C     *             /ISTAT/=2 PRINT ORDINATES OF DATA                *
C     *                       AND APPROXIMATED POINTS                *
C     *                       AND DEVIATIONS (ERROR).                *
C     *             /ISTAT/=1 ONLY PRINT [illegible]                 *
C     *             ISTAT>0   ABSOLUTE ERROR.                        *
C     *             ISTAT<0   RELATIVE ERROR.                        *
C     *  SUBM1(MM,MM) SUBMATRIX 1. CLASS IN THE SYSTEM     REAL      *
C     *             MATRIX.                                          *
C     *  MM         DIMENSIONING VARIABLE FOR SUBM1 [illegible] INTEGER *
C     *  SUBM2(N,N) SUBMATRIX 2. CLASS IN THE SUBMATRIX    REAL      *
C     *             1. CLASS.                                        *
C     *  IERROR     ERROR FLAG RETURNED TO THE CALLING     INTEGER   *
C     *             PROGRAM.                                         *
C     *             =0 NO ERROR DETECTED, GOOD JOB.                  *
C     *             =1 [illegible]                                   *
C     *                [illegible] NOT                               *
C     *                ACCURATE. REVISE PARAMETERS.                  *
C     *                                                              *
C     *  X1(9,7,5)  X1-COORDINATES OF THE DATA POINTS      REAL      *
C     *  X2(9,7,5)  X2-COORDINATES OF THE DATA POINTS      REAL      *
C     *  X3(9,7,5)  X3-COORDINATES OF THE DATA POINTS      REAL      *
C     *  Y(9,7,5)   Y-COORDINATES OF THE DATA POINTS       REAL      *
C     *  A(64,64)   SYSTEM MATRIX                          REAL      *
C     *  B(64)      RIGHT HAND SIDE VECTOR.                REAL      *
C     *                                                              *
C     *  IMPORTANT: THE DATA POINTS X1, X2, X3 AND Y HAVE TO BE PRO- *
C     *             VIDED THROUGH THE LABELED COMMON BLOCK DATA4 AND *
C     *             THE SYSTEM MATRIX A AND THE RIGHT HAND SIDE VECTOR*
C     *             B ARE AVAILABLE THROUGH THE LABELED COMMON BLOCK *
C     *             AB4. THIS SAVES A LOT OF THAT PRECIOUS THING     *
C     *             CALLED MEMORY.                                   *
C     *                                                              *
C     ****************************************************************
      COMMON / DATA4 / X1,X2,X3,Y
      COMMON / AB4   / A,B
      REAL X1(9,7,5),X2(9,7,5),X3(9,7,5),Y(9,7,5)
      REAL A(64,64),B(64)
      REAL COEF(64),SUBM1(MM,MM),SUBM2(N,N),STAT(9,7,5)
C     FORMATS THAT START
  101 FORMAT (///7H  AFTER,I3,44H STEPS NO SOLUTION WITHIN THE ACCURACY
     *OF,E10.3,11H [illegible]/)
  601 FORMAT (///51H  ORDER OF APPROXIMATION POLYNOMIAL IN X1 DIRECTION,
     *5X,I3/51H  ORDER OF APPROXIMATION POLYNOMIAL IN X2 DIRECTION,5X,I3
     */51H  ORDER OF APPROXIMATION POLYNOMIAL IN X3 DIRECTION,5X,I3
     */40H  NUMBER OF DATA POINTS WITH CONSTANT X1,13X,I3/40H  NUMBER OF
     * DATA POINTS WITH CONSTANT X2,13X,I3/40H  NUMBER OF DATA POINTS WI
     *TH CONSTANT X3,13X,I3//)
  602 FORMAT (///46H  SYSTEM MATRIX A AND VECTOR B BEFORE GAUSS JORDAN E
```

```
     *LIMINATION/)
 603 FORMAT ((4X,13(8X,I2)))
 604 FORMAT (2X,I2,13(1X,1PE9.2),1X/(16X,13(1X,1PE9.2)))
 605 FORMAT (///40H  EQUATION SYSTEM AFTER GAUSS JORDAN ELIMINATION/)
 606 FORMAT (///20H  COEFFICIENTS COEF(I,J,K)/)
 607 FORMAT (2X,3HI =,I2)
 608 FORMAT (12X,3HK =,I2,11(8X,I2),1X/3(7X,12(8X,I2)/))
 609 FORMAT (2X,3HJ =,I2,12(1X,E9.3),1X/3(6X,12(1X,E9.3)/))
 610 FORMAT (///39H  ORDINATES OF DATA POINTS  -  AS GIVEN)
 611 FORMAT (/8H  Y(I,J),(I2,1H))
 612 FORMAT (///46H  ORDINATES OF DATA POINTS  -  AS APPROXIMATED)
 613 FORMAT (///30H  ABSOLUTE ERROR AT EACH POINT)
 614 FORMAT (/13H  ERROR (I,J),(I2,1H))
 615 FORMAT (///31H  RELATIVE ERROR AT EACH POINT IN %)
 616 FORMAT (///21H  STATISTICS SUMMARY//7X,13HAVERAGE ERROR,29X,1PE9.
    *2/7X,19HMAXIMUM ERROR AT I=,I3,3X,2HJ=,I3,3X,2HK=,I3,4X,1PE9.2)
 617 FORMAT (///21H  STATISTICS SUMMARY//7X,13HAVERAGE ERROR,29X,0PF9.
    *2,2H %/7X,19HMAXIMUM ERROR AT I=,I3,2X,2HJ=,I3,3X,2HK=,I3,1X,0PF9.
    *2,2H %)
 618 FORMAT (///7H  AFTER,I3,41H STEPS NO SOLUTION WITHIN THE ACCURACY
    *OF,E10.3,11H  I QUIT./)
C     FORMATS MAT2 STOP
      LI=LOGLU(ISESSN)
      LO=6
      IERROR=1
      ITMAX=50
      EPSLON=.100E-007
C
C     PRINT APPROXIMATION PARAMETERS, IF IPRINT IS GREATER THAN 0.
C
      NORDR1=L-1
      NORDR2=M-1
      NORDR3=N-1
      IF (IPRINT.LT.0) GO TO 01
      WRITE (LO, 601) NORDR1,NORDR2,NORDR3,NPNTS1,NPNTS2,NPNTS3
C
C     START TO PRESET EDGE SECTION SUBMATRICES 1. CLASS.
C
   01 IPOWR1=0
      L1=2*L-1
      MN=M*N
      DO 05 K=1,L1,1
      IF (K.GT.L) GO TO 02
C
C     TOP EDGE SECTION SUBMATRICES 1. CLASS.
C
      ISTART=1
      ISTOP=MN
      JSTART=(K-1)*MN+1
      JSTOP=K*MN
      GO TO 03
C
C     RIGHT HAND EDGE SECTION SUBMATRICES 1. CLASS.
C
   02 ISTART=(K-L)*MN+1
      ISTOP=((K-L)+1)*MN
      JSTART=(L-1)*MN+1
      JSTOP=L*MN
C
C     PRESET EDGE SECTION SUBMATRICES 1. CLASS IN SYSTEM MATRIX USING MAT41.
C
   03 CALL MAT41 (NPNTS1,NPNTS2,NPNTS3,M,N,IPOWR1,SUBM1,MN,SUBM2)
      I1=0
      DO 04 I=ISTART,ISTOP,1
      I1=I1+1
      J1=0
      DO 04 J=JSTART,JSTOP,1
      J1=J1+1
   04 A(I,J)=SUBM1(I1,J1)
   05 IPOWR1=IPOWR1+1
C
C     COPY DEFINED SUBMATRICES 1. CLASS DIGONALLY.
C
      L1=L-1
      DO 07 I1=1,L1,1
      DO 07 J1=2,L,1
      ISTART=(I1-1)*MN+1
      ISTOP=I1*MN
      JSTART=(J1-1)*MN+1
      JSTOP=J1*MN
      DO 06 I=ISTART,ISTOP,1
      DO 06 J=JSTART,JSTOP,1
   06 A(I+MN,J-MN)=A(I,J)
   07 CONTINUE
```

```
C
C     PRESET RIGHT HAND SIDE VECTOR B.
C
      I=0
      DO 08 L1=1,L,1
      IPOWR1=L1-1
      DO 08 M1=1,M,1
      IPOWR2=M1-1
      DO 08 N1=1,N,1
      IPOWR3=N1-1
      I=I+1
   08 B(I)=S4(NPNTS1,NPNTS2,NPNTS3,IPOWR1,IPOWR2,IPOWR3,1)
      N1=L*MN
C
C     PRINT SYSTEM MATRIX AND RIGHT HAND SIDE VECTOR B, IF IPRINT
C     IS SET TO 2.
C
      IF (IPRINT.NE.2) GO TO 10
      WRITE (LO, 802)
      WRITE (LO, 803) (J,J=1,N1,1)
      DO 09 I=1,N1,1
   09 WRITE (LO, 804) I,(A(I,J),J=1,N1,1),B(I)
      WRITE (LO, 803) (J,J=1,N1,1)
      IPRINT=1
C
C     GAUSS JORDAN ELIMINATION.
C
   10 EPSLON=EPSLON*N1
      CALL IEL4 (N1,ITER,ITRMAX,EPSLON)
      IF (ITER.LT.0) GO TO 32
C
C     PRINT EQUATION SYSTEM, IF IPRINT IS SET TO 1.
C
      IF (IPRINT.NE.1) GO TO 12
      WRITE (LO, 805)
      WRITE (LO, 803) (J,J=1,N1,1)
      DO 11 I=1,N1,1
   11 WRITE (LO, 804) I,(A(I,J),J=1,N1,1),B(I)
      WRITE (LO, 803) (J,J=1,N1,1)
C
C     DEFINE COEFFICIENTS AND PRINT THEM, IF IPRINT IS GREATER THAN 0.
C
   12 CONTINUE
      DO 13 L1=1,N1,1
   13 COEF(L1)=B(L1)
      IF (IPRINT.LT.0) GO TO 16
      WRITE (LO, 806)
      DO 15 I1=1,L,1
      WRITE (LO, 807) I1
      WRITE (LO, 808) (I3,I3=1,N,1)
      DO 14 I2=1,M,1
      ICSTRT=(I1-1)*MN+(I2-1)*N+1
      ICSTOP=(I1-1)*MN+I2*N
   14 WRITE (LO, 804) I2,(COEF(IC),IC=ICSTRT,ICSTOP,1)
   15 WRITE (LO, 808) (I3,I3=1,N,1)
C
C     STATISTICS, IF ISTAT ANYTHING ELSE BUT 0.
C
   16 IF (ISTAT.EQ.0) GO TO 36
      IF (ISTAT.LT.0) IERROR=1
      IF (ISTAT.GT.0) IERROR=2
      ISTAT=IABS(ISTAT)
C
C     COMPUTE ABSOLUTE ERROR.
C     ( =(APPROXIMATED VALUE - GIVEN VALUE) )
C
      DO 17 I1=1,NPNTS1,1
      DO 17 I2=1,NPNTS2,1
      DO 17 I3=1,NPNTS3,1
   17 STAT(I1,I2,I3)=P4(COEF,X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3),NORD
     *1,NORD2,NORD3)-Y(I1,I2,I3)
C
C     COMPUTE RELATIVE ERROR, IF ISTAT WAS LESS THAN ZERO, I.E. IERROR=1.
C     ( =(APPROXIMATED VALUE - GIVEN VALUE)/(GIVEN VALUE) )
C
      IF (IERROR.NE.1) GO TO 19
      DO 18 I1=1,NPNTS1,1
      DO 18 I2=1,NPNTS2,1
      DO 18 I3=1,NPNTS3,1
   18 STAT(I1,I2,I3)=STAT(I1,I2,I3)/Y(I1,I2,I3)
C
C     ERROR STATISTICS: COMPUTE AVERAGE ERROR AND FIND LOCATION
C     OF MAXIMUM ERROR.
C
```

```
   19 ERRAVE=.0
      ERRMAX=ABS(STAT(1,1,1))
      IMAX=1
      JMAX=1
      KMAX=1
      DO 20 I1=1,NPNTS1,1
      DO 20 I2=1,NPNTS2,1
      DO 20 I3=1,NPNTS3,1
      IF (ABS(STAT(I1,I2,I3)).LE.ERRMAX) GO TO 20
      ERRMAX=STAT(I1,I2,I3)
      IMAX=I1
      JMAX=I2
      KMAX=I3
   20 ERRAVE=ERRAVE+ABS(STAT(I1,I2,I3))
      ERRAVE=ERRAVE/(NPNTS1*NPNTS2*NPNTS3)
      IF (ISTAT.NE.2) GO TO 25
C
C     PRINT ORDINATES OF DATA POINTS AND APPROXIMATED VALUES,
C     IF /ISTAT/ IS EQUAL 2.
C
      WRITE (LO, 610)
      DO 22 I3=1,NPNTS3,1
      WRITE (LO, 611) I3
      WRITE (LO, 663) (I2,I2=1,NPNTS2,1)
      DO 21 I1=1,NPNTS1,1
   21 WRITE (LO, 664) I1,(Y(I1,I2,I3),I2=1,NPNTS2,1)
   22 WRITE (LO, 663) (I2,I2=1,NPNTS2,1)
      WRITE (LO, 612)
      DO 24 I3=1,NPNTS3,1
      WRITE (LO, 611) I3
      WRITE (LO, 663) (I2,I2=1,NPNTS2,1)
      DO 23 I1=1,NPNTS1,1
   23 WRITE (LO, 664) I1,(F4(COEF,X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3)
     *,NORDR1,NORDR2,NORDR3),I2=1,NPNTS2,1)
   24 WRITE (LO, 663) (I2,I2=1,NPNTS2,1)
      ISTAT=1
C
C     PRINT ERRORS, IF /ISTAT/ WAS 1 OR 2.
C
   25 IF (ISTAT.NE.1) GO TO 31
      IF (IERROR.EQ.1) GO TO 28
C
C     PRINT ABSOLUTE ERRORS.
C
      WRITE (LO, 613)
      DO 27 I3=1,NPNTS3,1
      WRITE (LO, 614) I3
      WRITE (LO, 663) (I2,I2=1,NPNTS2,1)
      DO 26 I1=1,NPNTS1,1
   26 WRITE(LO, 606) I1,(STAT(I1,I2,I3),I2=1,NPNTS2,1)
   27 WRITE (LO, 663) (I2,I2=1,NPNTS2,1)
      GO TO 31
C
C     PRINT RELATIVE ERRORS.
C
   28 WRITE (LO, 615)
      DO 30 I3=1,NPNTS3,1
      WRITE (LO, 614) I3
      WRITE (LO, 663) (I2,I2=1,NPNTS2,1)
      DO 29 I1=1,NPNTS1,1
   29 WRITE (LO, 606) I1,(STAT(I1,I2,I3),I2=1,NPNTS2,1)
   30 WRITE (LO, 663) (I2,I2=1,NPNTS2,1)
C
C     PRINT ERROR STATISTICS.
C
   31 IF (IERROR.EQ.2) WRITE (LO, 616) ERRAVE,IMAX,JMAX,KMAX,ERRMAX
      IF (IERROR.EQ.1) WRITE (LO, 617) ERRAVE,IMAX,JMAX,KMAX,ERRMAX
      GO TO 34
C
C     FATAL ERROR DETECTED.  OUTPUT ERROR MESSAGE AND PRESET COEFFICIENTS.
C
   32 ITER=-ITER
      IERROR=-IERROR
      B(1)=Y(1,1,1)
      DO 33 IC=2,N1,1
   33 B(IC)=.0
      WRITE (LI, 101) ITER,EPSLON
      WRITE (LO, 618) ITER,EPSLON
      GO TO 12
C
C     RETURN.
C
   34 RETURN
      END
```

```
      SUBROUTINE MAT41 (NPNTS1,NPNTS2,NPNTS3,M,N,IPOWR1,SUBM1,MM,SUBM2)
C     ********************************************************************
C     *                                                                  *
C     *  PRESET SUBMATRIX 1. CLASS SUBM1 IN THE SYSTEM MATRIX A FOR A    *
C     *  4D-POLYNOMIAL APPROXIMATION.                                    *
C     *                                                                  *
C     *  AUTHOR:     HANS ZEBNER                                         *
C     *  DATE:       DECEMBER 31, 1980                                   *
C     *                                                                  *
C     *  VARIABLE    EXPLANATION                               TYPE      *
C     *  [illegible] [illegible]                               [illegible] *
C     *  NPNTS1      NUMBER OF DATA POINTS AT CONSTANT X1      INTEGER   *
C     *  NPNTS2      NUMBER OF DATA POINTS AT CONSTANT X2      INTEGER   *
C     *  NPNTS3      NUMBER OF DATA POINTS AT CONSTANT X3      INTEGER   *
C     *  M           NUMBER OF COEFFICIENTS FOR THE POLY-      INTEGER   *
C     *              NOMIAL IN X2 DIRECTION.                             *
C     *  N           NUMBER OF COEFFICIENTS FOR THE POLY-      INTEGER   *
C     *              NOMIAL IN X3 DIRECTION.                             *
C     *  IPOWR1      POWER FOR THE X1 VALUES IN THE            INTEGER   *
C     *              SUMMATION.                                          *
C     *  SUBM1(MM,MM) SUBMATRIX 1. CLASS IN THE SYSTEM         REAL      *
C     *              MATRIX.                                             *
C     *  MM          DIMENSIONING VARIABLE FOR SUBM1 =M*N      INTEGER   *
C     *  SUBM2(N,N)  SUBMATRIX 2. CLASS IN THE SUBMATRIX       REAL      *
C     *              1. CLASS.                                           *
C     *                                                                  *
C     ********************************************************************
      REAL SUBM1(MM,MM),SUBM2(N,N)
      IPOWR2=0
      M1=2*M-1
      DO 04 K=1,M1
      IF (K.GT.M) GO TO 01
C
C     TOP EDGE SECTION SUBMATRICES 2. CLASS.
C
      ISTART=1
      ISTOP=N
      JSTART=(K-1)*N+1
      JSTOP=K*N
      GO TO 02
C
C     RIGHT HAND EDGE SECTION SUBMATRICES 2. CLASS.
C
   01 ISTART=(K-M)*N+1
      ISTOP=((K+1)-M)*N
      JSTART=(M-1)*N+1
      JSTOP=M*N
C
C     PRESET EDGE SECTION SUBMATRICES 2. CLASS IN SUBMATRIX 1. CLASS
C     USING MAT42.
C
   02 CALL MAT42 (NPNTS1,NPNTS2,NPNTS3,N,IPOWR1,IPOWR2,SUBM2)
      I1=0
      DO 03 I=ISTART,ISTOP,1
      I1=I1+1
      J1=0
      DO 03 J=JSTART,JSTOP,1
      J1=J1+1
   03 SUBM1(I,J)=SUBM2(I1,J1)
   04 IPOWR2=IPOWR2+1
C
C     COPY DEFINED SUBMATRICES 2. CLASS DIAGONALLY.
C
      M1=M-1
      DO 06 I1=1,M1,1
      DO 06 J1=2,M,1
      ISTART=(I1-1)*N+1
      ISTOP=I1*N
      JSTART=(J1-1)*N+1
      JSTOP=J1*N
      DO 05 I=ISTART,ISTOP,1
      DO 05 J=JSTART,JSTOP,1
   05 SUBM1(I+N,J-N)=SUBM1(I,J)
   06 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MAT62 (NPNTS1,NPNTS2,NPNTS3,N,IPOWR1,IPOWR2,SUBM2)
C     ****************************************************************
C     *                                                              *
C     *  PRESET SUBMATRIX 2. CLASS SUMS IN THE SUBMATRIX 1. CLASS    *
C     *  SUBM1 FOR A 3D-POLYNOMIAL APPROXIMATION.                    *
C     *                                                              *
C     *  AUTHOR:    HANS ZEGNER                                      *
C     *  DATE:      DECEMBER 31, 1980                                *
C     *                                                              *
C     *  VARIABLE   EXPLANATION                            TYPE      *
C     *  NPNTS1     NUMBER OF DATA POINTS AT CONSTANT X1   INTEGER   *
C     *  NPNTS2     NUMBER OF DATA POINTS AT CONSTANT X2   INTEGER   *
C     *  NPNTS3     NUMBER OF DATA POINTS AT CONSTANT X3   INTEGER   *
C     *  N          NUMBER OF COEFFICIENTS FOR THE POLY-   INTEGER   *
C     *             NOMIAL IN X3 DIRECTION.                          *
C     *  IPOWR1     POWER FOR THE X1 VALUES IN THE         INTEGER   *
C     *             SUMMATION.                                       *
C     *  IPOWR2     POWER FOR THE X2 VALUES IN THE         INTEGER   *
C     *             SUMMATION.                                       *
C     *  SUBM2(N,N) SUBMATRIX 2. CLASS IN THE SUBMATRIX    REAL      *
C     *             1. CLASS.                                        *
C     *                                                              *
C     ****************************************************************
      REAL SUBM2(N,N)
      IPOWR3=0
      N1=2*N-1
      DO 03 K=1,N1,1
      IF (K.GT.N) GO TO 01
C
C     TOP EDGE SECTION ELEMENTS
C
      I=1
      J=K
      GO TO 02
C
C     RIGHT HAND EDGE SECTION ELEMENTS.
C
   01 I=(K-N)+1
      J=N
C
C     PRESET SECTION ELEMENTS IN SUBMATRIX 2. CLASS USING S4.
C
   02 SUBM2(I,J)=S4(NPNTS1,NPNTS2,NPNTS3,IPOWR1,IPOWR2,IPOWR3,J)
   03 IPOWR3=IPOWR3+1
      N1=N-1
      DO 04 I=1,N1,1
      DO 04 J=2,N,1
   04 SUBM2(I+1,J-1)=SUBM2(I,J)
      RETURN
      END
```

```
      REAL FUNCTION S4 (NPNTS1,NPNTS2,NPNTS3,IPOWR1,IPOWR2,IPOWR3,IY)
C**********************************************************************
C*                                                                    *
C*    PERFORM SUMMATIONS TO PRESET THE ELEMENTS A(I,J) IN THE        *
C*    SYSTEM MATRIX A AND THE ELEMENTS B(I) IN THE RIGHT HAND        *
C*    SIDE VECTOR B FOR THE LINEAR EQUATION SYSTEM, THAT MUST        *
C*    BE SOLVED FOR A POLYNOMIAL X3-APPROXIMATION.                   *
C*                                                                    *
C*    AUTHOR:     HANS ZESNER                                         *
C*    DATE:       DECEMBER 31, 1980                                   *
C*                                                                    *
C*    VARIABLE    EXPLANATION                               TYPE      *
C*    ********    *******************************          *******   *
C*    NPNTS1      NUMBER OF DATA POINTS AT CONSTANT X1     INTEGER   *
C*    NPNTS2      NUMBER OF DATA POINTS AT CONSTANT X2     INTEGER   *
C*    NPNTS3      NUMBER OF DATA POINTS AT CONSTANT X3     INTEGER   *
C*    IPOWR1      POWER FOR THE X1 VALUES IN THE           INTEGER   *
C*                SUMMATION.                                          *
C*    IPOWR2      POWER FOR THE X2 VALUES IN THE           INTEGER   *
C*                SUMMATION.                                          *
C*    IPOWR3      POWER FOR THE X3 VALUES IN THE           INTEGER   *
C*                SUMMATION.                                          *
C*    IY          CONTROL PARAMETER                        INTEGER   *
C*                =0  Y VALUES ARE NOT INCLUDED IN THE               *
C*                    SUMMATION.                                      *
C*                    (-> SYSTEM MATRIX ELEMENTS)                     *
C*                =1  Y VALUES ARE INCLUDED IN THE         INTEGER   *
C*                    SUMMATION.                                      *
C*                    (-> RIGHT HAND SIDE VECTOR ELEM.)               *
C*    S4          RESULT OF SUMMATION.                     REAL      *
C*                                                                    *
C*    X1(9,7,5)   X1-COORDINATES OF THE DATA POINTS        REAL      *
C*    X2(9,7,5)   X2-COORDINATES OF THE DATA POINTS        REAL      *
C*    X3(9,7,5)   X3-COORDINATES OF THE DATA POINTS        REAL      *
C*    Y(9,7,5)    Y-COORDINATES OF THE DATA POINTS         REAL      *
C*                                                                    *
C*    IMPORTANT:  THE DATA POINTS X1, X2, X3 AND Y HAVE TO BE PRO-   *
C*                VIDED THROUGH THE LABELED COMMON BLOCK DATA4.       *
C*                                                                    *
C**********************************************************************
      COMMON / DATA4 / X1,X2,X3,Y
      REAL X1(9,7,5),X2(9,7,5),X3(9,7,5),Y(9,7,5)
      S=0.0
      DO 07 K1=1,NPNTS1,1
      DO 07 K2=1,NPNTS2,1
      DO 07 K3=1,NPNTS3,1
      IF (IPOWR1.EQ.0) GO TO 01
      S1=X1(K1,K2,K3)**IPOWR1
      GO TO 02
   01 S1=1.
   02 IF (IPOWR2.EQ.0) GO TO 03
      S2=X2(K1,K2,K3)**IPOWR2
      GO TO 04
   03 S2=1.
   04 IF (IPOWR3.EQ.0) GO TO 05
      S3=X3(K1,K2,K3)**IPOWR3
      GO TO 06
   05 S3=1.
   06 SY=Y(K1,K2,K3)
      IF (IY.EQ.0) SY=1.
   07 S=S+((S1*S2)*S3)*SY
      S4=S
      RETURN
      END
```

```
      INTEGER FUNCTION IEL44 (N)
C     ***************************************************************
C     *                                                             *
C     *  GAUSS JORDAN ELIMINATION WITH MATRIX A AND VECTOR B.  THE  *
C     *  VARIABLE IEL4 IS SET TO 1, IF THE ELIMINATION IS COMPLETED.*
C     *  IEL4 LESS THAN 0 INDICATES AN ERROR.                       *
C     *                                                             *
C     *  AUTHOR:     HANS ZEBMER                                    *
C     *  DATE:       DECEMBER 31, 1980                              *
C     *                                                             *
C     *  VARIABLE    EXPLANATION                         TYPE       *
C     *  ********    ******************************      *******    *
C     *  N           NUMBER OF EQUATIONS                 INTEGER    *
C     *  IEL4        CONTROL PARAMETER, RETURNED TO THE  INTEGER    *
C     *              CALLING ROUTINE.                               *
C     *              =1 GAUSS JORDAN ELIMINATION WAS                *
C     *                 PERFORMED CORRECTLY.                        *
C     *              <0 ERROR ENCOUNTERED DURING GAUSS              *
C     *                 JORDAN ELIMINATION.                         *
C     *                                                             *
C     *  A(64,64)    SYSTEM MATRIX A                     REAL       *
C     *  B(64)       RIGHT HAND SIDE VECTOR B            REAL       *
C     *                                                             *
C     *  IMPORTANT:  MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH
C     *              THE LABELED COMMON BLOCK A64.  BOTH A AND B WILL
C     *              WILL UNDERGO CHANGES DURING EXECUTION, I.E. THE *
C     *              ELIMINATION.                                   *
C     *                                                             *
C     ***************************************************************
      COMMON / A64  / A,B
      REAL A(64,64),B(64)
      IEL4=1
      IEL44=1
      EPS=1.00E-030
      DO 06 I=1,N+1
      DO 02 K=1,N+1
      IF (K.EQ.I) GO TO 02
      CONST=-A(K,I)/A(I,I)
      DO 01 J=1,N+1
      A(K,J)=A(K,J)+CONST*A(I,J)
      IF (J.EQ.I) A(K,J)=0.0
   01 CONTINUE
      B(K)=B(K)+CONST*B(I)
   02 CONTINUE
      CONST=A(I,I)
      DO 03 J=1,N+1
   03 A(I,J)=A(I,J)/CONST
      A(I,I)=1.
      B(I)=B(I)/CONST
      DO 06 J=1,N+1
      IERROR=0
      DO 04 K=1,N+1
      IF (ABS(A(J,K)).GT.EPS) GO TO 04
      IERROR=IERROR+1
   04 CONTINUE
      IF (ABS(B(J)).GT.EPS) GO TO 05
      IERROR=IERROR+1
   05 IF (IERROR.GT.N+1) GO TO 07
   06 CONTINUE
      GO TO 08
   07 IEL4=-1
      IEL44=-1
   08 RETURN
      END
```

```
      SUBROUTINE IEL4 (N,ITER,ITRMAX,EPSLON)
C     ************************************************************
C     *                                                          *
C     *  GAUSSIAN ELIMINATION WITH LINEAR EQUATION SYSTEM V (THE RIGHT *
C     *  HAND SIDE VECTOR B IS INTEGRATED TOGETHER WITH THE SYSTEM MA- *
C     *  TRIX A IN THE MATRIX V). ITER IS THE NUMBER OF ITERATIONS RE- *
C     *  TURNED TO THE CALLING PROGRAM. THE ITERATION TO IMPROVES THE *
C     *  SOLUTION SO MANY TIMES, UNTIL THE REMAINING ERROR IS WITHIN *
C     *  THE MAGNITUDE OF EPSLON. SHOULD THE NUMBER OF ITERATIONS EX- *
C     *  CEED ITRMAX, THE SUBROUTINE TERMINATES THE CALCULATION AND *
C     *  SETS ITER TO A NEGATIVE VALUE.                          *
C     *                                                          *
C     *  AUTHORS:   KONRAD VOGELER, HANS ZEBNER                  *
C     *  DATE:      DECEMBER 31, 1980                            *
C     *                                                          *
C     *  VARIABLE   EXPLANATION                          TYPE    *
C     *  ********   ********************************     ******* *
C     *  N          NUMBER OF EQUATIONS                  INTEGER *
C     *  ITER       NUMBER OF ITERATIONS TO BE RETURNED  INTEGER *
C     *             >0 GAUSSIAN ELIMINATION PERFORMED            *
C     *                CORRECTLY.                                *
C     *             <0 NO CONVERGENCE OF THE SOLUTION FOR        *
C     *                THE DESIRED RESOLUTION EPSLON.            *
C     *                ITERATION DID TERMINATE AFTER             *
C     *                ITRMAX ITERATION STEP.                    *
C     *  ITRMAX     MAXIMUM NUMBER OF ITERATION STEPS    INTEGER *
C     *  EPSLON     DESIRED RESOLUTION OF THE SOLUTION   REAL    *
C     *                                                          *
C     *  A(64,64)   SYSTEM MATRIX A                      REAL    *
C     *  B(64)      RIGHT HAND SIDE VECTOR B             REAL    *
C     *                                                          *
C     *  IMPORTANT: MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH *
C     *             THE LABELED COMMON BLOCK AB3. BOTH A AND B WILL *
C     *             WILL UNDERGO CHANGES DURING EXECUTION, I.E. THE *
C     *             ELIMINATION.                                 *
C     *                                                          *
C     ************************************************************
      COMMON / AB4 / V
      REAL V(64,65),G(55),A(64,65),B(65)
      LOGICAL IPRINT
      DATA IPRINT /.FALSE./
      DATA PI /3.14159265/
C     FORMATS IEL4 START
  601 FORMAT (///30H  MATRIX A UND VEKTOR B VOR ELIMINATION/)
  602 FORMAT ((4X,13(8X,I2)))
  603 FORMAT (2X,I2,13(1X,1PE9.2)/(4X,13(1X,1PE9.2)))
  604 FORMAT (///25H  LOESUNGSVEKTOR DES SYSTEMS (N,I3,10H) ITERATIONSSCHRITT/
     *)
  605 FORMAT (///2X,I2,26H KOEFFIZIENTEN AUS GAUSS/)
  606 FORMAT ((2X,I2,1X,1PE9.2))
  607 FORMAT (///39H  KLEINSTER ZEILENVEKTORSCHNITTWINKEL =,E12.3)
  608 FORMAT (27X,12H RESIDIUM =,E12.3,5H NACH,I3,12H ITERATIONEN)
  609 FORMAT (///20X,11H FEHLER V(,I2,1H,,I2,3H) =,E12.3/)
  610 FORMAT (///25H  SYSTEM NACH ELIMINATION/)
C     FORMATS IEL4 STOP
      LO=6
      N1=N+1
      K=N-1
C
C     REARRANGE VECTOR B IN ARRAY V, IF N IS LESS THAN 64.
C
      IF (N.EQ.64) GO TO 02
      DO 01 I=1,N1
   01 V(I,N1)=V(I,65)
C
C     PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B BEFORE THE
C     ELIMINATION, IF IPRINT IS SET .TRUE. .
C
   02 IF (IPRINT) WRITE (LO, 601)
      IF (IPRINT) WRITE (LO, 602) (J,J=1,N1)
      DO 03 I=1,N1
   03 IF (IPRINT) WRITE (LO, 603) I,(V(I,J),J=1,N1)
      IF (IPRINT) WRITE (LO, 602) (J,J=1,N1)
C
C     LINEAR EQUATION SYSTEM  V(N,N)*G(N)=V(N,N+1)
C
      DO 04 I=1,N1
      DO 04 J=1,N1
   04 A(I,J)=V(I,J)
      DO 05 I=1,N1
   05 B(I)=.0
C
C     CHECK FOR STABILITY.
```

```
C
      FI=.0
      DO 06 I=1,N,1
   06 G(I)=.0
      DO 08 I=1,N,1
      DO 07 J=1,N,1
   07 G(I)=G(I)+V(I,J)**2
   08 G(I)=SQRT(G(I))
      DO 10 I=1,N,1
      JJ=I+1
      DO 10 J=JJ,N,1
      SK=.0
      DO 09 I3=1,M,1
   09 SK=SK+V(I,I3)*V(J,I3)
      F=SK/(G(I)*G(J))
      IF (ABS(F).GT.ABS(FI)) FI=F
   10 CONTINUE
      FI=ABS(FI)
      FI=ACOS(FI)
      FI=(FI*180.)/PI
C
C     START ITERATION TO FIND AND IMPROVE SOLUTION.
C
      DO 27 LX=1,999,1
      ITER=LX
      DO 15 J=1,N,1
C
C     PIVOTISATION AND GAUSSIAN STEP FOR COLUMN J.
C
      IX=J
      E=V(J,J)
      DO 12 I=J,N,1
      IF (ABS(V(I,J)).GT.ABS(E)) GO TO 11
      GO TO 12
   11 IX=I
      E=V(I,J)
   12 CONTINUE
      IF (IX.EQ.J) GO TO 14
      DO 13 I=1,M,1
      G(I)=V(IX,I)
      V(IX,I)=V(J,I)
   13 V(J,I)=G(I)
      G(N)=.0
   14 MM=J+1
      DO 15 I=MM,N,1
      W=V(I,J)/V(J,J)
      DO 15 KK=J,M
      V(I,KK)=V(I,KK)-V(J,KK)*W
   15 CONTINUE
      I=N
      IF (V(I,I).EQ..0) GO TO 32
C
C     SOLUTION.
C
      DO 16 I=1,N,1
   16 G(I)=.0
      G(N)=V(N,M)/V(N,N)
      DO 19 II=1,N,1
      I=N-II
      IF (V(I,I).EQ..0) GO TO 32
      G(I)=V(I,M)
      DO 17 JJ=1,N,1
      J=M-JJ
      IF (J.EQ.I) GO TO 18
   17 G(I)=G(I)-G(J)*V(I,J)
   18 CONTINUE
   19 G(I)=G(I)/V(I,I)
C
C     CORRECT SOLUTION BY AMOUNT OF ERROR, WHICH WAS INVOLVED IN THIS
C     ITERATION STEP. TAKE THIS VECTOR AS START VECTOR FOR THE NEXT
C     ITERATION.
C
      DO 20 I=1,N,1
   20 G(I)=B(I)+G(I)
C
C     COMPUTE THE REMAINING ERROR.
C
      DO 22 I=1,N,1
      V(I,M)=.0
      DO 21 J=1,N,1
   21 V(I,M)=V(I,M)+G(J)*A(I,J)
   22 V(I,M)=A(I,M)-V(I,M)
      R=.0
      DO 23 I=1,N,1
   23 R=R+V(I,M)*V(I,M)
```

```
      R=SQRT(R)
      IF (R.LT.EPSLON) GO TO 28
      IF (LI.GT.ITMAX) GO TO 33
      DO 24 I=1,N+1
   24 B(I)=G(I)
C
C     PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B AFTER THE
C     PIVOTISATION.
C
      IF (IPRINT) WRITE (L3, 504) LI
      IF (IPRINT) WRITE (L3, 602) (J,J=1,N+1)
      DO 25 I=1,N+1
   25 IF (IPRINT) WRITE (L3, 603) I,(V(I,J),J=1,N+1),G(I)
      IF (IPRINT) WRITE (L3, 602) (J,J=1,N+1)
      DO 26 I=1,N+1
      DO 26 J=1,N+1
   26 V(I,J)=A(I,J)
C
C     STOP LOOP TO FIND SOLUTION.
C
   27 CONTINUE
C
C     REARRANGE SOLUTION IN THE ARRAY V.
C
   28 CONTINUE
      DO 29 I=1,N+1
   29 V(I,N3)=G(I)
C
C     PRINT SOLUTION, IF IPRINT IS SET .TRUE. .
C
   30 IF (IPRINT) WRITE (L3, 503) R
      IF (IPRINT) WRITE (L3, 505) (I,G(I),I=1,N+1)
      IF (IPRINT) WRITE (L3, 507) F1
      IF (IPRINT) WRITE (L3, 506) R,LI
C
C     DRUCKEN DER MATRIX A UND DES VECTORS B NACH ELIMINATION.
C
      IF (IPRINT) WRITE (L3, 610)
      IF (IPRINT) WRITE (L3, 602) (J,J=1,N+1)
      DO 31 I=1,N+1
   31 IF (IPRINT) WRITE (L3, 603) I,(V(I,J),J=1,N+1)
      IF (IPRINT) WRITE (L3, 602) (J,J=1,N+1)
      RETURN
C
C     ERROR ENCOUNTERED.  STOP.
C
   32 ITER=-1
      WRITE (L3, 609) I,I,V(I,I)
      G(I)=G(I)/V(I,I)
      RETURN
C
C     NO CONVERGENCE OF THE SOLUTION.
C
   33 ITER=-ITER
      RETURN
      END
```

```
      REAL FUNCTION F4 (COEF,X1,X2,X3,NORDR1,NORDR2,NORDR3)
C     ***********************************************************
C     *                                                         *
C     *  COMPUTE FUNCTION VALUE Y=F(X1,X2,X3) OF A 4D-POLYNOMIAL USING *
C     *  THE HORNER SCHEME IN ORDER TO INCREASE SPEED AND ACCURACY. *
C     *                                                         *
C     *  AUTHOR:    HANS ZEBNER                                 *
C     *  DATE:      DECEMBER 31, 1980                           *
C     *                                                         *
C     *  VARIABLE   EXPLANATION                        TYPE     *
C     *  ********   ********************************   ******** *
C     *  COEF(64)   POLYNOMIAL COEFFICIENTS            REAL     *
C     *  X1         X1-VALUE, WHERE Y WILL BE COMPUTED. REAL    *
C     *  X2         X2-VALUE, WHERE Y WILL BE COMPUTED. REAL    *
C     *  X3         X3-VALUE, WHERE Y WILL BE COMPUTED. REAL    *
C     *  NORDR1     ORDER OF THE POLYNOMIAL IN THE      INTEGER *
C     *             X1 DIRECTION.                               *
C     *  NORDR2     ORDER OF THE POLYNOMIAL IN THE      INTEGER *
C     *             X2 DIRECTION.                               *
C     *  NORDR3     ORDER OF THE POLYNOMIAL IN THE      INTEGER *
C     *             X3 DIRECTION.                               *
C     *  F4         ORDINATE OF THE 4D-POLYNOMIAL.      REAL    *
C     *                                                         *
C     ***********************************************************
      REAL COEF(64),D(64),E(64)
      NCOEF1=NORDR1+1
      NCOEF2=NORDR2+1
      NCOEF3=NORDR3+1
C
C     COMPUTE COEFFICIENTS D BASED ON COEFFICIENTS COEF AND X3
C
      DO 02 I1=1,NCOEF1,1
      DO 02 I2=1,NCOEF2,1
      IC=(I1-1)*NCOEF2*NCOEF3+I2*NCOEF3
      ID=(I1-1)*NCOEF2+I2
      Y=COEF(IC)
      DO 01 I31=1,NORDR3,1
      I3=NCOEF3-I31
      IC=(I1-1)*NCOEF2*NCOEF3+(I2-1)*NCOEF3+I3
   01 Y=COEF(IC)+X3*Y
   02 D(ID)=Y
C
C     COMPUTE COEFFICIENTS E BASED ON COEFFICIENTS D AND X2.
C
      DO 04 I1=1,NCOEF1,1
      ID=I1*NCOEF2
      Y=D(ID)
      DO 03 I21=1,NORDR2,1
      I2=NCOEF2-I21
      ID=(I1-1)*NCOEF2+I2
   03 Y=D(ID)+X2*Y
   04 E(I1)=Y
C
C     COMPUTE Y BASED ON COEFFICIENTS E AND X1.
C
      Y=E(NCOEF1)
      DO 05 I11=1,NORDR1,1
      I1=NCOEF1-I11
   05 Y=E(I1)+X1*Y
      F4=Y
      RETURN
      END
```

DISTRIBUTION LIST

		No. of Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	1
2.	Library Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Office of Research Administration Code 012A Naval Postgraduate School Monterey, California 93940	1
4.	Chairman Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5.	Director, Turbopropulsion Laboratory Department of Aeronautics Naval Postgraduate School Monterey, California 93940	30
6.	Dr. Gerhard Heiche Naval Air Systems Command Code AIR-310 Navy Department Washington, D.C. 20360	1
7.	Mr. Karl H. Guttmann Naval Air Systems Command Code AIR-330 Navy Department Washington, D.C. 20360	1
8.	Mr. James R. Patton, Jr. Power Program, Code 473 Office of Naval Research Arlington, Virginia 22218	1
9.	Dr. A. D. Wood Office of Naval Research Eastern/ Central Regional Office 666 Summer Street Boston, Massachusetts 02210	1

10. Commanding Officer 1
Naval Air Propulsion Test Center
Attn: Mr. Vernon Lubosky
Trenton, New Jersey 08628

11. National Aeronautics & Space Administration 1
Lewis Research Center (Library)
2100 Brookpark Road
Cleveland, Ohio 44135

12. CAG Library 1
The Boeing Company
Seattle, Washington 98124

13. Library 1
General Electric Company
Aircraft Engine Technology Division
DTO Mail Drop H43
Cincinnati, Ohio 45215

14. Library 1
Pratt and Whitney Aircraft
Post Office Box 2691
West Palm Beach, Florida 33402

15. Library 1
Pratt and Whitney Aircraft
East Hartford, Connecticut 06108

16. Chief, Fan and Compressor Branch 1
Mail Stop 5-9
NASA Lewis Research Center
2100 Brookpark Road
Cleveland, Ohio 44135

17. Prof. D. Adler 1
Technion Israel Institute of Technology
Department of Mechanical Engineering
Haifa 32000
Israel

18. Director, Whittle Laboratory 1
Department of Engineering
Cambridge University
England

19. Prof. F. A. E. Breugelmans 1
Institut von Karman de la Dynamique
des Fluides
72 Chausee de Waterloo
1640 Rhode-St. Genese
Belgium

20. Library 1
Air Research Mfg. Corporation
Division of Garrett Corporation
402 South 36th Street
Phoenix, Arizona 85034

21. Dr. Robert P. Dring 1
United Technologies Research Labs
400 Main Street
Hartford, Connecticut 06108

22. Mr. Jean Fabri 1
ONERA
29, Ave. de la Division Leclerc
92 Chatillon
France

23. Prof. Dr. Ing Heinz E. Gallus 1
Lehrstuhl und Institut fur Strahlantiebe
und Turboarbeitsmashinen
Rhein.-Westf. Techn. Hochschule Aachen
Templergraben 55
5100 Aachen, Germany

24. Mr. R. A. Langworthy 1
Army Aviation Material Laboratories
Department of the Army
Fort Eustis, Virginia 23604

25. Dr. W. Schlachter 1
Brown, Boveri & Company Ltd.
Department T-T
Haselstrasse
CH-5401 Baden, Switzerland

26. Prof. T. H. Okiishi 1
Professor of Mechanical Engineering
208 Mechanical Engineering Building
Iowa State University
Ames, Iowa 50011

27. Dr. Leroy H. Smith, Jr. 1
Manager, Compressor & Fan Technology Operation
General Electric Company
Aircraft Engine Technology Division
DTO Mail Drop H43
Cincinnati, Ohio 45215

28. Dr. Arthur J. Wennerstrom 1
ARL/LF
Wright-Patterson AFB
Dayton, Ohio 45433

29. Dr. H. Weyer 1
DFVLR
Linder Hohe
505 Porz-Wahn
Germany

30. Mr. P. F. Yaggy 1
Director
U.S. Army Aeronautical Research Laboratory
AMES Research Center
Moffett Field, California 94035

31. Prof. C. H. Wu 1
P.O. Box 2706
Beijing 100080
China

32. Director 1
Gas Turbine Establishment
P.O. Box 305
Jiangyou County
Sichuan Province
China

END

DATE
FILMED
3-82

DTIC