IS-3179

INTERACTIVE COMPUTER PROGRAMS FOR LINEAR AND NON-LINEAR PROJECTION METHODS

J. A. Schmitz, D. D. Georg and R. F. Keller

AMES LABORATORY, USAEC

IOWA STATE UNIVERSITY

AMES, IOWA

Date Transmitted: July 1973

PREPARED FOR THE U. S. ATOMIC ENERGY COMMISSION DIVISION OF RESEARCH UNDER CONTRACT W-7405-eng-82

IS-3179

INTERACTIVE COMPUTER PROGRAMS FOR LINEAR AND NON-LINEAR PROJECTION METHODS

J. A. Schmitz, D. D. Georg and R. F. Keller

Ames Laboratory, USAEC
Iowa State University
Ames, Iowa 50010

Date Transmitted: July 1973

PREPARED FOR THE U. S. ATOMIC ENERGY COMMISSION
DIVISION OF RESEARCH UNDER CONTRACT NO. W-7405-eng-82

Available from: National Technical Information Service
Department A
Springfield, VA 22151

Price: Microfiche $0.95

TABLE OF CONTENTS

ABSTRACT

User instructions and documentation are given for two separate interactive programs. These interactive programs were designed to aid in the investigation of projection methods as a basis for solving linear and non-linear systems of equations. At each iteration step, the user must specify the order k of the subspace being projected on where k = 1, 2, or 3, and the columns of the coefficient matrix or Jacobian matrix which form this subspace.

I. INTRODUCTION

These sets of user instructions and documentation illustrate the use of and describe two CPS BASIC programs which were designed to aid in the investigation of linear and non-linear projection methods. In both programs, the user has the option of specifying a 1, 2 or 3 dimensional projection at each iteration step. The algorithms used in these programs are described by the references listed in the Bibliography.

II. PROGRAM FOR LINEAR SYSTEM

A. USE INSTRUCTION

After loading the program, and before execution, the user must type in:

20 mat read A(N,N),B(N),X(N)

where N is the integer constant specifying the dimensions of the system.

Lines 30-39 can be used to enter the elements of matrices A, B, and x. The format to be used here is:

Ln# DATA A(1,1), A(1,2), . . . , A(1,N), A(2,1), . . . ,

A(N,N), B(1), . . . , B(N), x(1), . . . , x(N)

Where A(I,J) denotes the constant in the I-th row and J-th column of the coefficient matrix A, B(I) denotes the I-th element of the constant vector B, and x(I) denotes the I-th element of the initial approximation to the solution vector.

During execution, the user is asked to input several constants. Following are the letters the program will type, and the response the user should type in:

N - integer specifying dimension of system

U - number of columns to be projected on; to end execution, the user should enter a negative value for U

I - the number of the columns you want to project on. For a 1-dimensional projection method, you will type in 1 number, 2 for the 2-dimensional method and 3 for the 3-dimensional method

V - if the user wants to keep the last change in the x-vector, he should type a 0; to restore the old approximation to x, type a 1.

Anytime the user makes an error when typing, a correction can be made by backspacing over the mistake and typing in the correct characters.

B. OUTPUT

Initially the C matrix, its maximum and minimum elements and their difference are displayed. At each iteration, the following is printed:

R - residue vector

x - approximation to solution vector

T - matrix

norm = $(r^k,r^k)^{\frac{1}{2}}$

C. ERROR MESSAGES

The only program-produced error message results when a non-negative number not equal to 1, 2, or 3 is input for the dimension of the system to be projected on. In this case, the user is asked to re-enter the dimension.

D. SAMPLE RUN

```
LOGIN(CPS  ,          ,BASIC)
GOOD MORNING,  USER 02; TIME  9:56:13  7/06/73.
load(projn)
20 mat read a(4,4),b(4),x(4)
30 data 1.00, 0.96, 0.84, 0.64              *
31 data 0.96, 0.92, 0.44, 0.22          (A)
32 data 0.84, 0.44, 1.00, 0.34
33 data 0.64, 0.22, 0.34, 1.00
34 data 3.44, 2.54, 2.63, 2.21
35 data 0,0,0,0
execute 1 thru ...                          *
N                                       (B)
4

MAT C    4 BY    4
0                10.812381105697 8.3116065886182 2.9467455387781
10.812381105697 0                3.7824357039317 1.8018412581165
8.3116065886182 3.7824357039317 0                2.1973919914941
2.9467455387781 1.8018412581165 2.1973919914941 0
maximum= 10.812, minimum=  1.802, delta c=  9.011
norm=  5.485
R              T              X
.34400E01      .98820E00      .0
.25400E01      .87702E00      .0
.26300E01      .90059E00      .0
.22100E01      .72647E00      .0

dimension of projection=?
U                                           *
2                                       (C)

I
1,2
norm=  0.565
R              T              X
.39234E00      .4137E-29      .34696E01
-.3865E00      .3929E-31      -.4395E00
-.911E-01      .1487E-01      .0
.8616E-01      .9755E-01      .0

erase this change?  1=yes,0=no              *
V                                       (D)
0
```

*The letters refer to a written explanation in the sample program description.

```
dimension of projection=?
U
1

I
4
norm=  0.537
R              T              X
.30234E00      .7141E-01      .34696E01
-.4174E00      .4810E-01      -.4395E00
-.1389E00      .1307E-01      .0
-.545E-01      .3255E-33      .14062E00

erase this change?  1=yes,0=no
V
0

dimension of projection=?
U
3


I
1,2,3
norm=  0.450
R              T              X
.27927E00      .1770E-27      .27087E01
-.2439E00      .1085E-27      -.889E-01
-.1865E00      .1803E-28      .53248E00
.17429E00      .17458E00      .14062E00

erase this change?  1=yes,0=no
V
0

dimension of projection=?
U
2

I
2,4
norm=  0.373
R              T              X
.25945E00      .2600E-01      .27087E01
-.1566E00      .1901E-31      -.2483E00
-.2082E00      .2279E-01      .53248E00
-.608E-01      .8811E-33      .41079E00

erase this change?  1=yes,0=no
V                                            (E)*
1

dimension of projection=?
U
3
```

```
I
2,4,1
norm=  0.195
R            T            X
.10874E00    .6279E-30    .17467E01
-.1315E00    .6277E-31    .62616E00
.6994E-01    .8640E-01    .53248E00
-.644E-01    .1195E-27    .83775E00

erase this change?  1=yes,0=no
V
0

dimension of projection=?
U
-1
```

Ⓕ*

```
**    640 XEQ "END".
logout
ACCOUNT NUMBER        ;  DATE  7/06/73;  TOTAL RUN COST    $1.35
ITEMIZED COSTS:            CPU      $.67; TERM      $.30; PAGE      $.38;
TIME 10:06:14; TIME USED: CPU 00:00:10; TERM 00:10:00; PAGE 00:38:40;
```

E. SAMPLE PROGRAM DESCRIPTION

A. The user loads the program and types in the values of the A matrix and the B and x vectors. Lines 30 through 33 each contain one row of the A-matrix, line 34 contains the B-vector and line 35 contains the first approximation to the x-vector. The numbers must be entered in this order, although it is not necessary to enter four per line.

B. User types in dimension of the system.

C. User has the option of typing in 1, 2 or 3 depending on the projection method he wants to use. In this case, he decides to use a two-dimensional projection method. Had he typed in a negative number, execution would have ended. Any other response would have caused an error message. The user types in 1, 2 for the value of I, causing the projection method to be based on columns 1 and 2 of A.

D. User decides to keep the change, and types in 0.

E. At this point, the user has decided to erase the last change. He therefore types a 1. The previous values of x and the residue vector are restored to the system.

F. The user types a -1 to end execution.

F. TECHNICAL BASIS

The residual vector r^k is defined to be $r^k = B-Ax^k$ for a system of linear equations, where x^k is the k-th approximation to the solution vector x.

In the one dimensional projection method, the i-th component of the vector x is changed according to:

$$x_i^{k+1} = x_i^k + (a_i, r^k)/(a_i, a_i)$$

The residual vector then becomes,

$$r^{k+1} = r^k - \frac{(a_i, r^k)}{(a_i, a_i)} a_i$$

For the two dimensional method we have:

$$x_i^{k+1} = x_i^k + \Delta x_i^k$$

$$x_j^{k+1} = x_j^k + \Delta x_j^k$$

where

$$\Delta x_i^k = [(r^k, a_i) - (r^k, a_j)(a_i, a_j)/(a_j, a_j)]C(i,j)/A(i,i)$$

$$\Delta x_j^k = [(r^k, a_j) - \Delta x_i^k(a_i, a_j)]/(a_j, a_j)$$

$$r^{k+1} = r^k - \Delta x_i^k a_i - \Delta x_j^k a_j$$

The three dimensional algorithm is based on the equations:

$$(r^{k+1}, a_1) = 0$$

$$(r^{k+1}, a_2) = 0$$

$$(r^{k+1}, a_3) = 0$$

Where 1, 2, and 3 are the first, second and third columns being projected on

Here,

$$r^{k+1} = r^k - \Delta x_1^k a_1 - \Delta x_2^k a_2 - \Delta x_3^k a_3$$

Therefore,

$$\Delta x_1^k = 1/D\ [(r^k, a_1)S_1 + (r^k, a_2)t_1 + (r^k, a_3)t_2]$$

$$\Delta x_2^k = 1/D\ [(r^k, a_2)S_2 + (r^k, a_1)t_1 + (r^k, a_3)t_3]$$

$$\Delta x_3^k = 1/D\ [(r^k, a_3)S_3 + (r^k, a_2)t_3 + (r^k, a_1)t_2]$$

The following definitions are used:

$$D = 1 + 2\cos\theta_{12}\cos\theta_{13}\cos\theta_{23} - \cos^2\theta_{12} - \cos^2\theta_{13} - \cos^2\theta_{23}$$

$$S_1 = (1 - \cos^2\theta_{23})/(a_1, a_1)$$

$$S_2 = (1 - \cos^2\theta_{13})/(a_2, a_2)$$

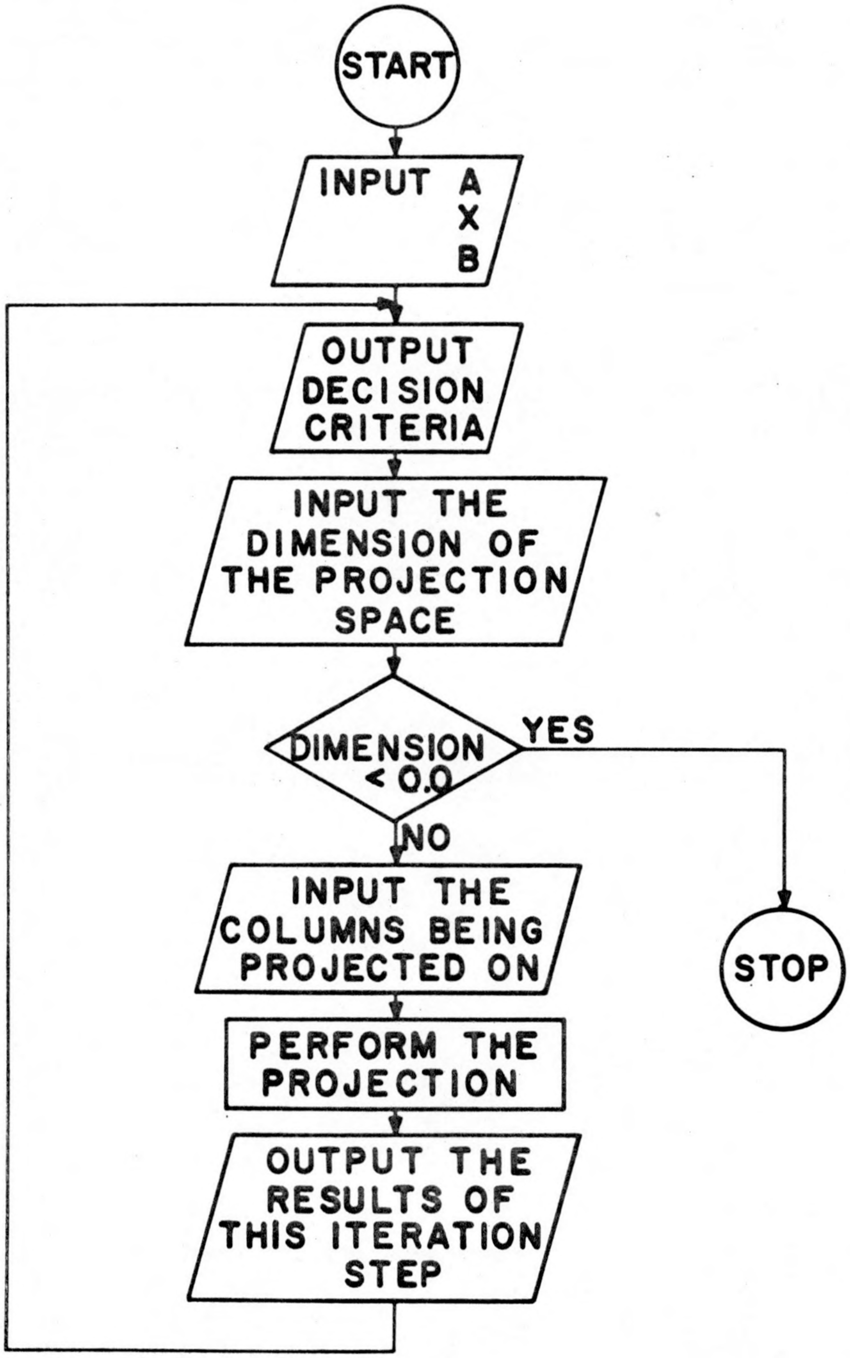
$$S_3 = (1-\theta\cos^2\theta_{12})/(a_3,a_3)$$

$$t_1 = (\cos\theta_{13}\cos\theta_{23} - \cos\theta_{12})/((a_1,a_1)(a_2,a_2))^{\frac{1}{2}}$$

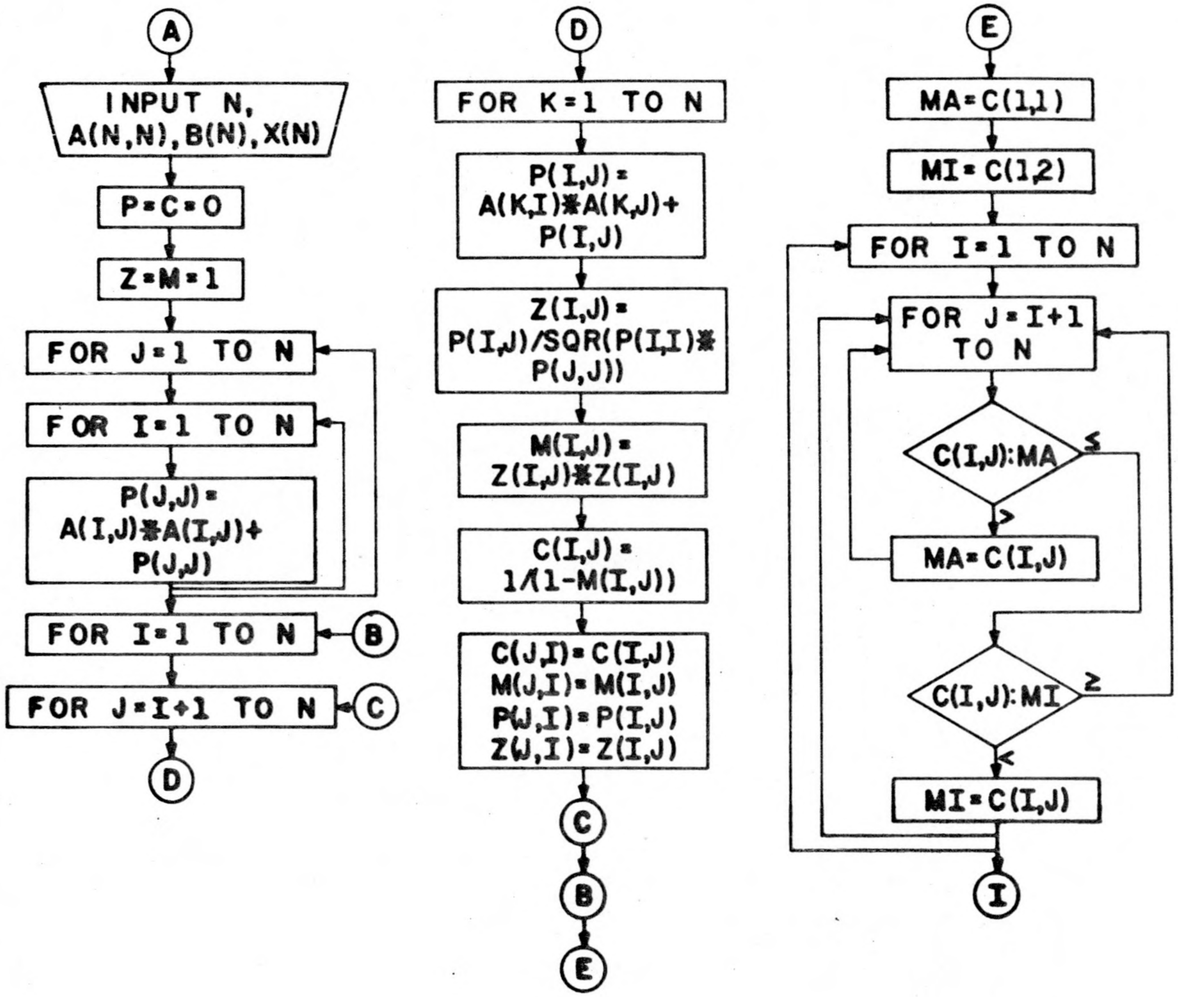
$$t_2 = (\cos\theta_{12}\cos\theta_{23} - \cos\theta_{13})/((a_1,a_1)(a_3,a_3))^{\frac{1}{2}}$$

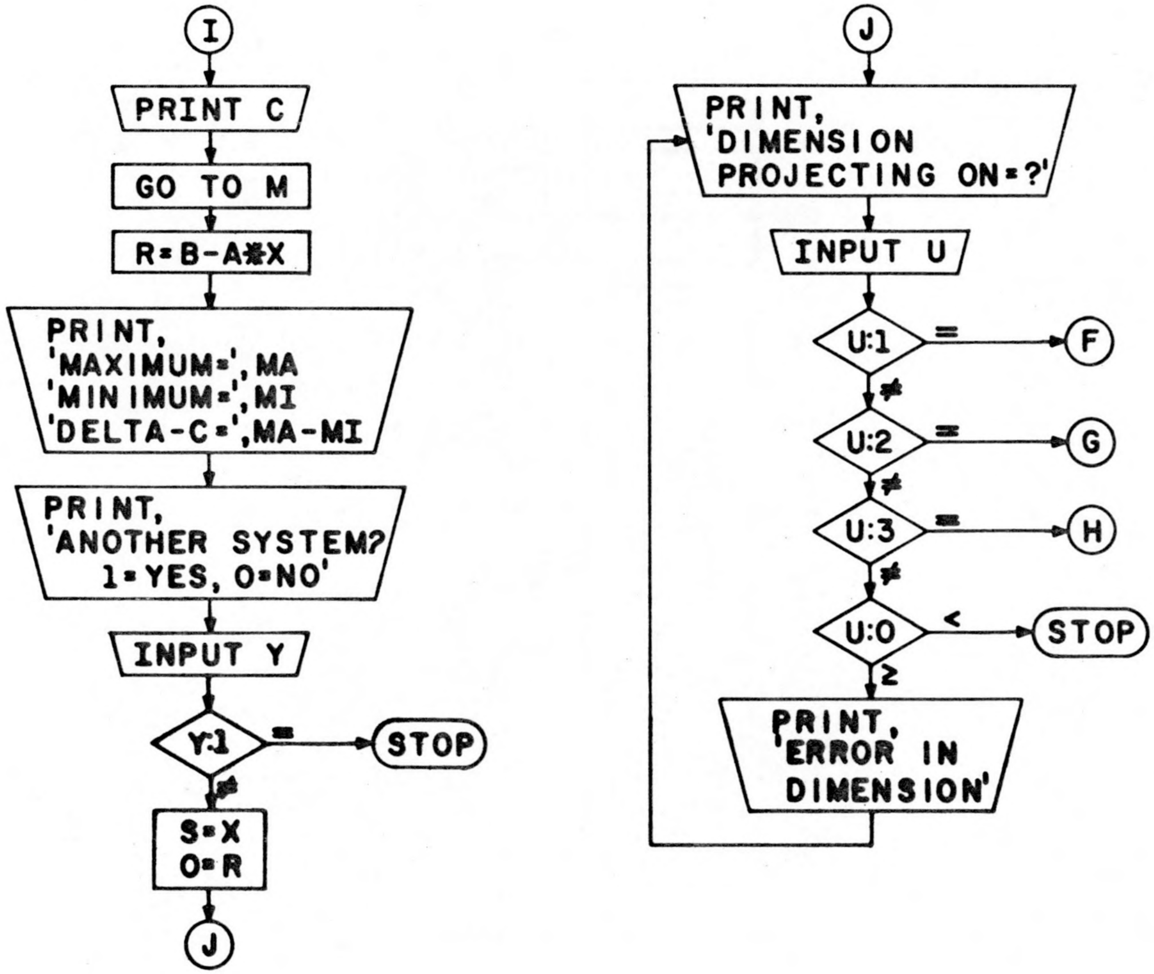
$$t_3 = (\cos\theta_{12}\cos\theta_{13} - \cos\theta_{23})/((a_2,a_2)(a_3,a_3))^{\frac{1}{2}}$$

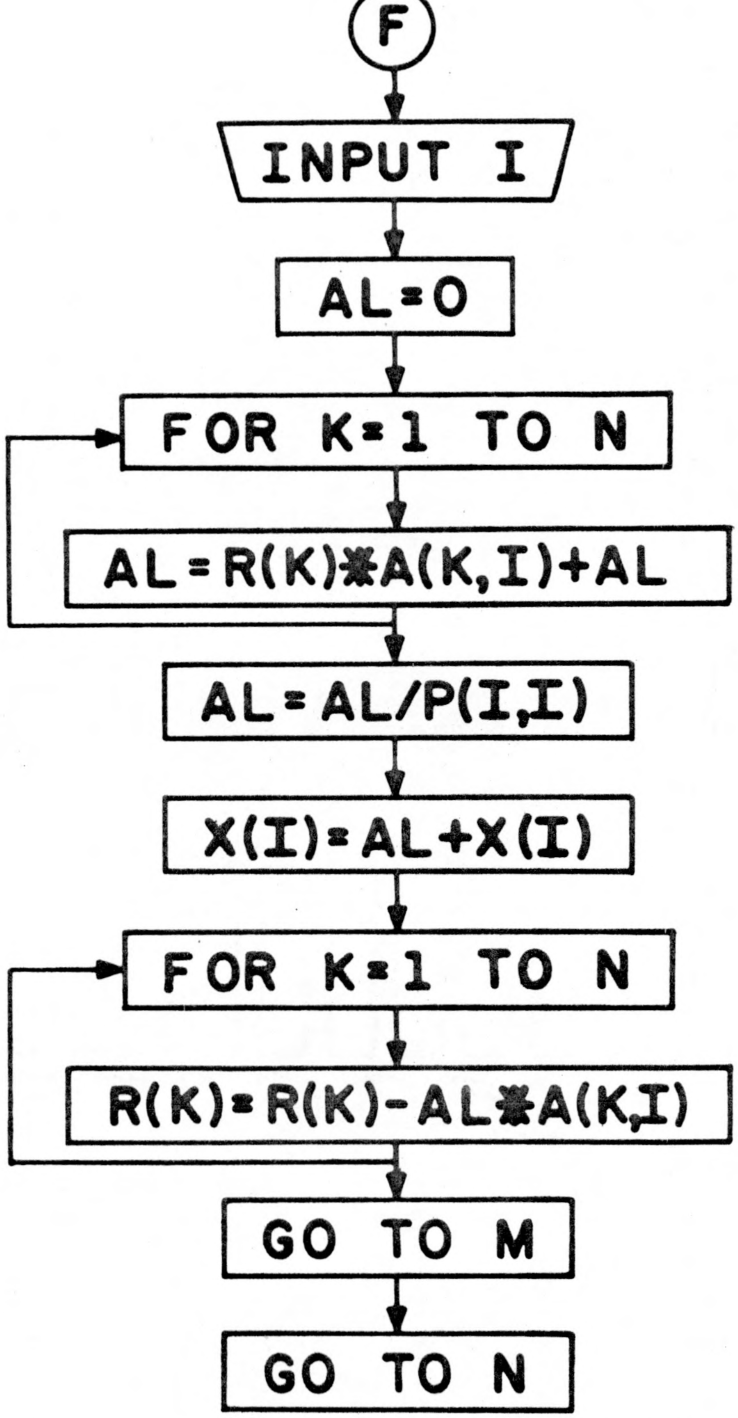
G. LINEAR SYSTEM FLOW CHART

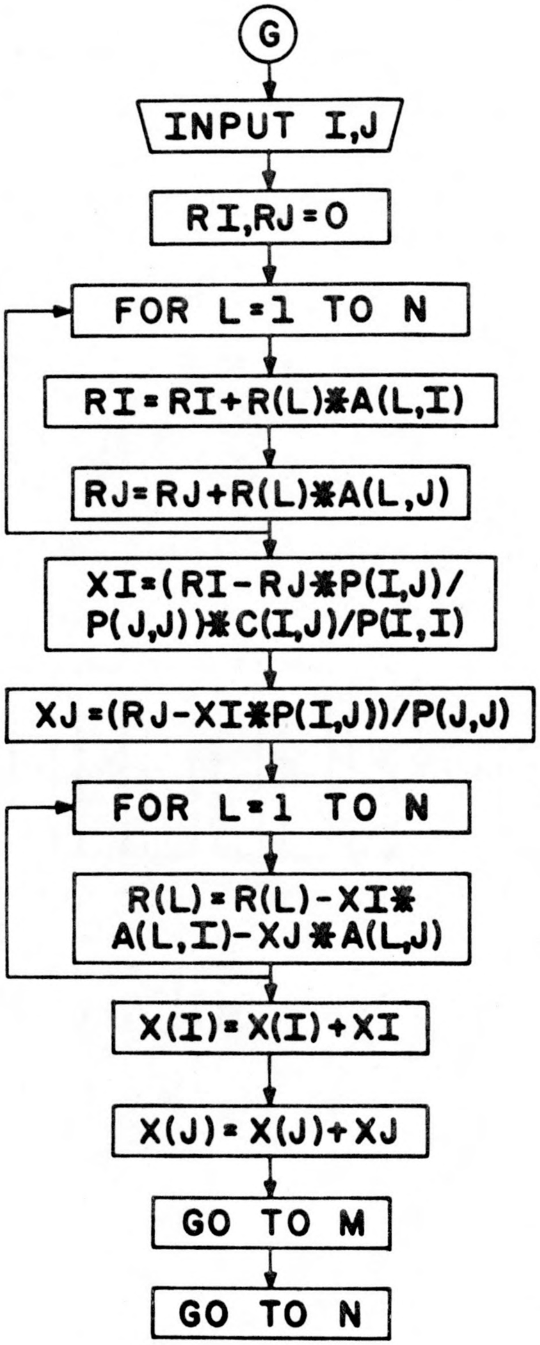


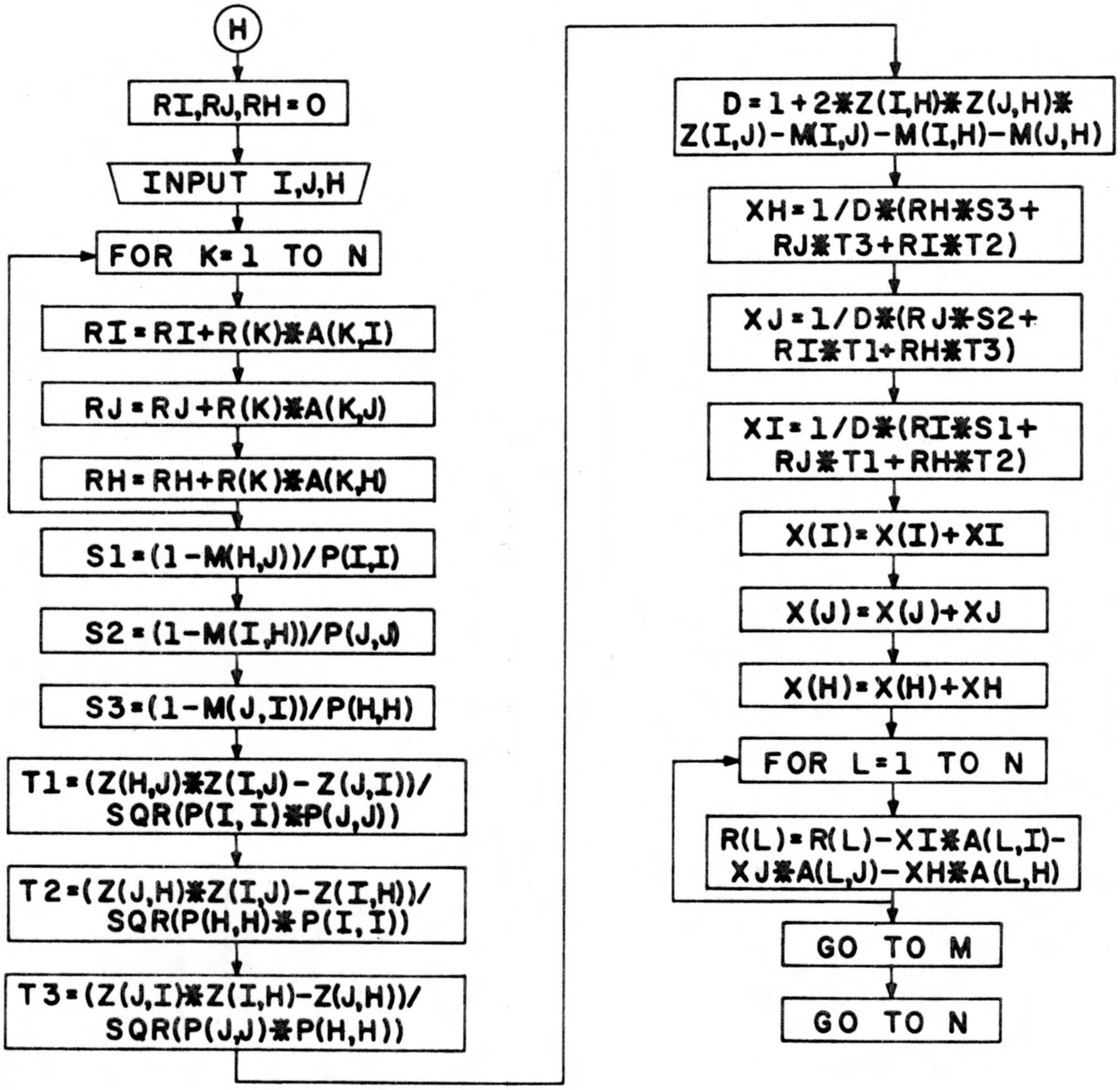
H. LINEAR PROGRAM FLOWCHART

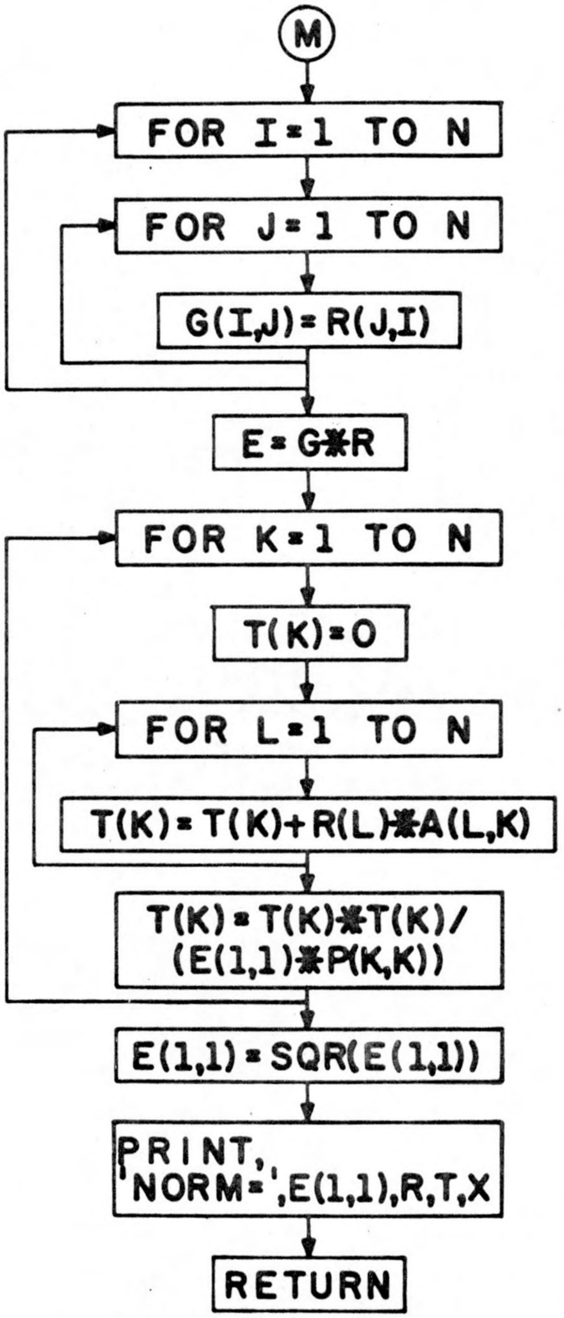


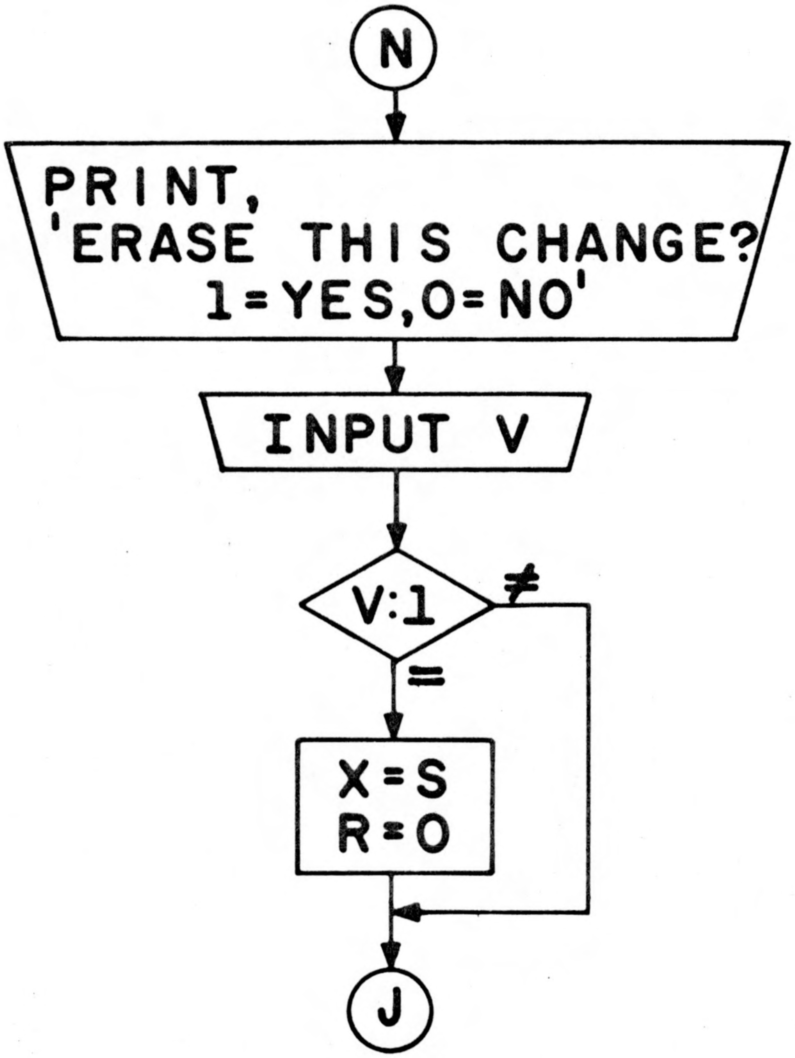

I
PRINT C
GO TO M
R=B-A*X
PRINT, 'MAXIMUM=',MA 'MINIMUM=',MI 'DELTA-C=',MA-MI
PRINT, 'ANOTHER SYSTEM? 1=YES, O=NO'
INPUT Y
Y:1
=
STOP
≠
S=X O=R
J
J
PRINT, 'DIMENSION PROJECTING ON=?'
INPUT U
U:1
=
F
≠
U:2
=
G
≠
U:3
=
H
≠
U:0
<
STOP
≥
PRINT, 'ERROR IN DIMENSION'


F
INPUT I
AL=0
FOR K=1 TO N
AL=R(K)*A(K,I)+AL
AL=AL/P(I,I)
X(I)=AL+X(I)
FOR K=1 TO N
R(K)=R(K)-AL*A(K,I)
GO TO M
GO TO N


G
INPUT I,J
RI,RJ=0
FOR L=1 TO N
RI=RI+R(L)*A(L,I)
RJ=RJ+R(L)*A(L,J)
XI=(RI-RJ*P(I,J)/
P(J,J))*C(I,J)/P(I,I)
XJ=(RJ-XI*P(I,J))/P(J,J)
FOR L=1 TO N
R(L)=R(L)-XI*
A(L,I)-XJ*A(L,J)
X(I)=X(I)+XI
X(J)=X(J)+XJ
GO TO M
GO TO N


H
RI,RJ,RH = 0
INPUT I,J,H
FOR K=1 TO N
RI = RI+R(K)*A(K,I)
RJ = RJ+R(K)*A(K,J)
RH = RH+R(K)*A(K,H)
S1=(1-M(H,J))/P(I,I)
S2 = (1-M(I,H))/P(J,J)
S3=(1-M(J,I))/P(H,H)
T1=(Z(H,J)*Z(I,J) - Z(J,I))/ SQR(P(I,I)*P(J,J))
T2=(Z(J,H)*Z(I,J)-Z(I,H))/ SQR(P(H,H)*P(I,I))
T3=(Z(J,I)*Z(I,H)-Z(J,H))/ SQR(P(J,J)*P(H,H))
D = 1+2*Z(I,H)*Z(J,H)* Z(I,J)-M(I,J)-M(I,H)-M(J,H)
XH=1/D*(RH*S3+ RJ*T3+RI*T2)
XJ=1/D*(RJ*S2+ RI*T1+RH*T3)
XI=1/D*(RI*S1+ RJ*T1+RH*T2)
X(I)=X(I)+XI
X(J)=X(J)+XJ
X(H)=X(H)+XH
FOR L=1 TO N
R(L)=R(L)-XI*A(L,I)- XJ*A(L,J)-XH*A(L,H)
GO TO M
GO TO N


M
FOR I=1 TO N
FOR J=1 TO N
G(I,J)=R(J,I)
E=G*R
FOR K=1 TO N
T(K)=0
FOR L=1 TO N
T(K)=T(K)+R(L)*A(L,K)
T(K)=T(K)*T(K)/
(E(1,1)*P(K,K))
E(1,1)=SQR(E(1,1))
PRINT,
'NORM=',E(1,1),R,T,X
RETURN


N
PRINT,
'ERASE THIS CHANGE?
1=YES,0=NO'
INPUT V
V:1
≠
=
X=S
R=0
J

I. PROGRAM LISTING

```
 10 INPUT  N
 15 PRINT USING 9035, ' '
 20 MAT READ A(6,6),B(6,1),X(6,1)
 30 DATA .3,-.5,.5,-.5,.4,-.5
 31 DATA .2,-.4,.4,-.5,.3,-.4
 32 DATA .1,-.3,.1,-.2,.2,-.4
 33 DATA .1,-.2,.2,-.1,.2,-.3
 34 DATA -.2,.3,-.2,.2,-.4,.2
 35 DATA -.3,.1,-.1,.1,-.2,.2
 36 DATA -.3,-.4,-.5,-.1,-.1,-.2
 37 DATA 0,0,0,0,0,0
 40 MAT P,C=ZER(N,N)
 45 MAT Z,M=CON(N,N)
 50 FOR J=1 TO N
 60 FOR I=1 TO N
 70 LET P(J,J)=A(I,J)*A(I,J)+P(J,J)
 80 NEXT I
 90 NEXT J
110 FOR I=1 TO N
120 FOR J=I+1 TO N
130 FOR K=1 TO N
140 LET P(I,J)=A(K,I)*A(K,J)+P(I,J)
150 NEXT K
160 LET Z(I,J)=P(I,J)/SQR(P(I,I)*P(J,J))
165 LET M(I,J)=Z(I,J)*Z(I,J)
170 LET C(I,J)=1/(1-M(I,J))
200 LET C(J,I)=C(I,J)
205 LET M(J,I)=M(I,J)
210 LET P(J,I)=P(I,J)
215 LET Z(J,I)=Z(I,J)
220 NEXT J
230 NEXT I
240 LET MA=C(1,1)
250 LET MI=C(1,2)
260 FOR I=1 TO N
270 FOR J=I+1 TO N
280 IF C(I,J)>MA THEN 400
290 IF C(I,J)<MI THEN 450
300 GO TO 460
400 LET MA=C(I,J)
410 GO TO 460
450 LET MI=C(I,J)
460 NEXT J
470 NEXT I
475 MAT PRINT C
```

```
  476 MAT R=A*X
  477 MAT R=B-R
  478 PRINT USING 9030, MA,MI,MA-MI
  480 GOSUB 2310
  540 PRINT USING 9000, ' '
  550 INPUT  U
  560 PRINT USING 9035, ' '
  570 MAT S=X
  580 MAT O=R
  590 IF U=1 THEN 760
  600 IF U=2 THEN 1030
  610 IF U=3 THEN 2030
  615 IF U<0 THEN 640
  620 PRINT USING 625, ' '
  625 IMAGE
error in dimension-
  630 GO TO 540
  640 STOP
  760 INPUT  I
  770 LET AL=0
  780 FOR K=1 TO N
  785 LET AL=R(K)*A(K,I)+AL
  800 NEXT K
  805 LET AL=AL/P(I,I)
  810 LET X(I)=X(I)+AL
  820 FOR K=1 TO N
  830 LET R(K)=R(K)-AL*A(K,I)
  840 NEXT K
  850 GOSUB 2310
  855 GOSUB 2450
  860 GO TO 540
 1030 INPUT  I,J
 1050 LET RI,RJ=0
 1060 FOR L=1 TO N
 1070 LET RI=RI+R(L)*A(L,I)
 1080 LET RJ=RJ+R(L)*A(L,J)
 1090 NEXT L
 1100 LET XI=(RI-RJ*P(I,J)/P(J,J))*C(I,J)/P(I,I)
 1110 LET XJ=(RJ-XI*P(I,J))/P(J,J)
 1130 FOR L=1 TO N
 1140 LET R(L)=R(L)-XI*A(L,I)-XJ*A(L,J)
 1150 NEXT L
 1160 LET X(I)=X(I)+XI
 1170 LET X(J)=X(J)+XJ
 1180 GOSUB 2310
 1185 GOSUB 2450
 1190 GO TO 540
 2030 LET RI,RJ,RH=0
 2035 PRINT USING 9035, ' '
 2040 INPUT  I,J,H
```

```
2050 FOR K=1 TO N
2060 LET RI=RI+R(K)*A(K,I)
2070 LET RJ=RJ+R(K)*A(K,J)
2080 LET RH=RH+R(K)*A(K,H)
2090 NEXT K
2120 LET S1=(1-M(H,J))/P(I,I)
2130 LET S2=(1-M(I,H))/P(J,J)
2140 LET S3=(1-M(J,I))/P(H,H)
2180 LET T1=(Z(H,J)*Z(I,H)-Z(J,I))/SQR(P(I,I)*P(J,J))
2190 LET T2=(Z(J,H)*Z(I,J)-Z(I,H))/SQR(P(H,H)*P(I,I))
2200 LET T3=(Z(J,I)*Z(I,H)-Z(H,J))/SQR(P(J,J)*P(H,H))
2210 LET D=1+2*Z(I,H)*Z(J,H)*Z(I,J)-M(I,H)-M(J,H)-M(I,J)
2220 LET XH=1/D*(RH*S3+RJ*T3+RI*T2)
2230 LET XJ=1/D*(RJ*S2+RI*T1+RH*T3)
2240 LET XI=1/D*(RI*S1+RJ*T1+RH*T2)
2250 LET X(I)=XI+X(I)
2260 LET X(J)=XJ+X(J)
2270 LET X(H)=XH+X(H)
2280 FOR L=1 TO N
2290 LET R(L)=R(L)-XI*A(L,I)-XJ*A(L,J)-XH*A(L,H)
2300 NEXT L
2305 GOSUB 2310
2306 GOSUB 2450
2307 GO TO 540
2310 MAT G=TRN(R)
2320 MAT E=G*R
2330 MAT T=ZER(N)
2340 FOR K=1 TO N
2350 FOR L=1 TO N
2360 LET T(K)=T(K)+R(L)*A(L,K)
2370 NEXT L
2380 LET T(K)=T(K)*T(K)/(E(1,1)*P(K,K))
2390 NEXT K
2400 LET E(1,1)=SQR(E(1,1))
2405 PRINT USING 9040, E(1,1)
2410 PRINT USING 9050, ' '
2420 FOR I=1 TO N
2430 PRINT USING 9060, R(I),T(I),X(I)
2435 NEXT I
2440 PRINT USING 9035, ' '
2445 RETURN
2450 PRINT USING 2460, ' '
2460 IMAGE
erase this change?  1=yes,0=no-
2470 INPUT  V
2480 IF V<>1 THEN 2555
2490 MAT X=S
2500 MAT R=0
2555 PRINT USING 9035, ' '
2560 RETURN
```

```
 9000 IMAGE
dimension of projection=?-
 9010 IMAGE
another system?  1=yes,0=no-
 9030 IMAGE
maximum=---.---, minimum=---.---, delta c=---.---
 9035 IMAGE
-
 9040 IMAGE
norm=---.---
 9050 IMAGE
R            T            X
 9060 IMAGE
.........   .........   .........
_
```

J. VARIABLE DICTIONARY

A = coefficient matrix

AI = $(r^k, a_i)/(a_i, a_i)$

B = constant vector of the linear system

C(I,J) = $1/(1-\cos^2\theta_{ij})$

E = $(r^k, r^k)^{\frac{1}{2}}$

G = transpose of residue vector

H = third column being projected on (3 dimensional projection)

I = first column being projected on

J = second column being projected on (2 or 3 dimensional projection)

M(I,J) = $\cos^2\theta_{ij} = (a_i, a_j)^2/(a_i, a_i)/(a_j, a_j)$

MA = maximum element of C-matrix

MI = minimum element of C-matrix

N = dimension of system

O = R^{k-1}

P(I,J) = (a_i, a_j)

R = residue vector

RI = (r^k, a_i)

RJ = (r^k, a_j)

S = x^{k-1}

S1 = $(1-\cos^2\theta_{jh})/(a_i, a_i)$

S2 = $(1-\cos^2\theta_{ih})/(a_i, a_j)$

S3 = $(1-\cos^2\theta_{ij})/(a_h, a_h)$

T1 $= (\cos\theta_{ih}\cos\theta_{jh} - \cos\theta_{ij})/((a_i,a_i)(a_j,a_j))^{\frac{1}{2}}$

T2 $= (\cos\theta_{ij}\cos\theta_{jh} - \cos\theta_{ih})/((a_i,a_i)(a_h,a_h))^{\frac{1}{2}}$

T3 $= (\cos\theta_{ij}\cos\theta_{ih} - \cos\theta j_h)/((a_j,a_j)(a_h,a_h))^{\frac{1}{2}}$

T(I) $= \dfrac{(r^k,a_i)(r^k,a_i)}{(r^k,r^k)(a_i,a_i)}$

U = number of dimensions projecting on

V = indicates if last change in the approximation to the solution vector should be kept

W = indicates whether another iteration is to be performed

X = solution vector

XH $= X^k(H) - X^{k-1}(H)$

XI $= X^k(I) - X^{k-1}(I)$

XJ $= X^k(J) - X^{k-1}(J)$

Z(I,J) $= \cos\theta_{ij} = (a_i,a_j)/((a_i,a_i)(a_j,a_j))^{\frac{1}{2}}$

III. PROGRAM FOR NON-LINEAR SYSTEMS

A. USE INSTRUCTIONS

After loading the program, the user is required to supply the following lines:

```
5000 DIM P(N), F(N,N), T(N)
5030 MAT  READ x(N)
5196 PRINT USING 5197, F(I,1), . . . , F(I,N)
5203 PRINT USING 5197, C(I,1), . . . , C(I,N)
```

Where N is the integer specifying the size of the system. The user must also type:

```
5010 DATA N, NS, AC
```

where N, NS, and AC are the values of the dimension of the system, the maximum step size, and the accuracy goal respectively.

```
5035 DATA X(1), . . . , x(N)
```

where x(I) is the value of the i-th component of the approximation to the solution vector.

Lines 5207-5239 can be used to supply the equations for P(x). The first statement must begin with line #5207, and the last statement must be a RETURN.

Lines 5240-5300 can be used to supply the equations necessary to evaluate the Jacobian. The first statement must begin with line #5240 and the last statement must be a RETURN.

During execution, the user will be asked to input a number of constants. Following are the letters the computer will print and the response the user should type in:

ND - the number of columns the user wants to project on at this step. A negative answer will end execution.

II - columns the user wants to project on. The user will type in 1 number for a 1-dimensional projection, 2 for the 2-dimensional case, and for a 3-dimensional method, 3 numbers.

Y - user types in a 0 to keep the change. A 1 will restore the old system.

B. OUTPUT

After each iteration, the following information is printed:

Step number

Jacobian

C-matrix corresponding to the Jacobian

Approximate solution vector x

P(x)

Theta(airk) = $(P(x),F)/((P(x), P(x))/F_i,F_i))^{\frac{1}{2}}$ =

cosine of the angle between column i of the Jacobian and the residue vector.

Norm = $(P(x), P(x))^{\frac{1}{2}}$

C. ERROR MESSAGES

The only program-produced error message results when a non-negative number not equal to 1, 2, or 3 is input for the dimension of the system to be projected on. In this case, the user is asked to re-enter the dimension.

D. SAMPLE RUN

```
LOGIN(CPS    ,          ,BASIC)
GOOD MORNING,  USER 02; TIME 10:07:50  7/06/73.        (A)*
load(NPROD)                                    *(B)
5000 dim p(3),f(3,3),t(3)
5020 data 3,7,.001                                     (C)*
5030 mat read x(3)
5035 data 10,10,10
5196 print using 5197,f(i,1),f(i,2),f(i,3)
5203 print using 5197,c(i,1),c(i,2),c(i,3)
5207 p(1)=x(2)*x(3)-1                                  (D)*
5208 p(2)=x(1)*x(3)-1
5209 p(3)=x(1)*x(2)-1
5210 return
5240 f(1,1)=0                                           (E)*
5241 f(1,2)=x(3)
5242 f(1,3)=x(2)
5243 f(2,1)=x(3)
5244 f(2,2)=0
5245 f(2,3)=x(1)
5246 f(3,1)=x(2)
5247 f(3,2)=x(1)
5248 f(3,3)=0
5249 return
run


step number=    1
ND
2                                                      (F)*

II
1,2
```

*The letters refer to a written explanation in the sample program description.

```
JACOBIAN:
.0              .1000000E02     .1000000E02
.1000000E02     .0              .1000000E02
.1000000E02     .1000000E02     .0

C:
.0              .1333333E01     .1333333E01
.1333333E01     .0              .1333333E01
.1333333E01     .1333333E01     .0


X               P(X)              Theta(airk)
.34000E01      .33000E02      .81650E00
.34000E01      .33000E02      .81650E00
.10000E02      .10560E02      .81650E00
norm=.4785E02
erase this change?  1=yes,0=no
Y                                                    (G)*
0


step number=    2
ND
3

II
1,2,3

JACOBIAN:
.0              .1000000E02     .3400000E01
.1000000E02     .0              .3400000E01
.3400000E01     .3400000E01     .0

C:
.0              .1010854E01     .1812216E01
.1010854E01     .0              .1812216E01
.1812216E01     .1812216E01     .0


X               P(X)              Theta(airk)
.18471E01      .79796E01      .72400E00
.18471E01      .79796E01      .72400E00
.48616E01      .24116E01      .97534E00
norm=.1154E02
erase this change?  1=yes,0=no
Y                                                    (H)*
1


step number=    3
ND
1
```

```
II
3

JACOBIAN:
.0                .1000000E02    .3400000E01
.1000000E02       .0             .3400000E01
.3400000E01       .3400000E01    .0

C:
.0                .1010854E01    .1812216E01
.1010854E01       .0             .1812216E01
.1812216E01       .1812216E01    .0

X              P(X)              Theta(airk)
.34000E01      .8882E-15      .72400E00
.34000E01      .8882E-15      .72400E00
.29412E00      .10560E02      .97534E00
norm=.1056E02
erase this change?  1=yes,0=no
Y
0

step number=     4
ND
2

II
1,3

JACOBIAN:
.0                .2941176E00    .3400000E01
.2941176E00       .0             .3400000E01
.3400000E01       .3400000E01    .0

C:
.0                .6756773E02    .1003728E01
.6756773E02       .0             .1003728E01
.1003728E01       .1003728E01    .0

X              P(X)              Theta(airk)
.30570E00      .45504E00      .99628E00
.34000E01      -.8692E00      .99628E00
.42795E00      .3936E-01      .1189E-15
norm=.9819E00
erase this change?  1=yes,0=no
Y
0
```

```
step number=    5
ND
1
II
2

JACOBIAN:
.0              .4279544E00     .3400000E01
.4279544E00     .0              .3056952E00
.3400000E01     .3056952E00     .0

C:
.0              .1498326E01     .1000125E01
.1498326E01     .0              .2914030E01
.1000125E01     .2914030E01     .0


X              P(X)               Theta(airk)
.30570E00     .13512E00     -.708E-01
.26524E01     -.8692E00     .40042E00
.42795E00     -.1892E00     .38231E00
norm=.8997E00
erase this change?  1=yes,0=no
Y
0


step number=    6
ND
-1
**    5083 XEQ "END".
logout
ACCOUNT NUMBER        ;  DATE  7/06/73;  TOTAL RUN COST    $1.50
ITEMIZED COSTS:             CPU     $.79; TERM      $.36; PAGE      $.35;
TIME 10:19:53; TIME USED: CPU 00:00:12; TERM 00:12:02; PAGE 00:35:33;
```

(I)*

E. SAMPLE PROGRAM DESCRIPTION

A - User dimensions p, f and t for his system of dimension 3

B - User enters dimension of the system. He wants to have a maximum of 7 steps and to reach an accuracy of .001.

C - Initial approximation to the vector x.

D - Values for P-matrix.

E - Values of Jacobian.

F - User has the option of typing a 1, 2, or 3 depending on the projection method he wants to use. In this case, he decides to use a two-dimensional method. Had he typed in a negative number, execution would have ended. Any other response would have caused an error message. The user types in 1, 2 for the value of II, causing the projection method to be based on columns 1 and 2 of the Jacobian.

G - The user wants to keep the last change. The next approximation will be based on the values of x just calculated.

H - The user decides to erase this change. The previous values of x and P(x) are restored and used in calculating the next approximation.

I - User types in a negative number to end execution.

F. TECHNICAL BASIS

The projection algorithm described by White(9) and MacEachern(4) for solving the non-linear system of equations Px = 0 is used in this program.

At each iteration step k, a change vector dx^k is calculated, and a new approximate solution vector is formed:

$$x^{k+1} = x^k + dx^k$$

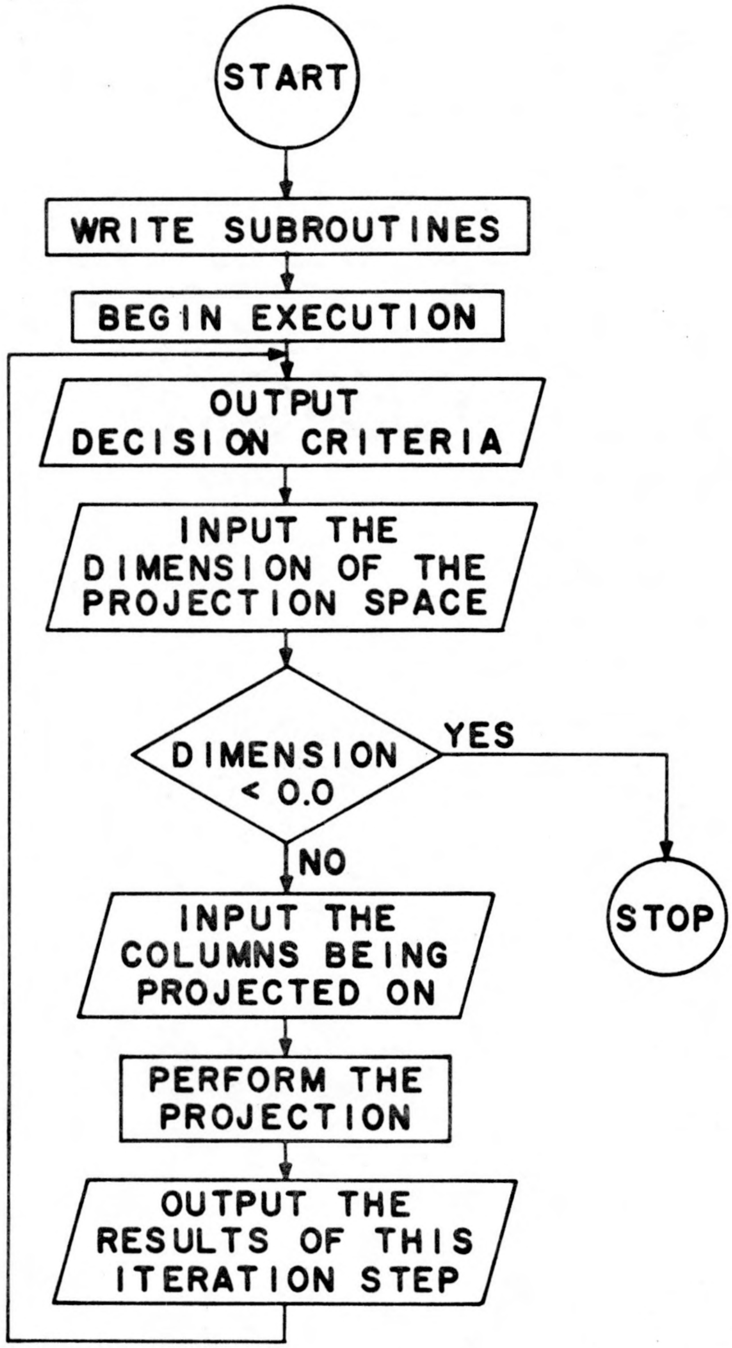
dx^k will have all components zero except for those components which correspond to the components of the approximate solution vector which are being modified at step k.

The nonzero components of dx^k can be solved for by approximating the functions that compose P with Taylor series and truncating the second order terms and above. If n components of the approximate solution vector are modified at step k, then the linear system that must be solved in order to obtain the nonzero components of dx^k is of dimension n. The coefficient matrix of the system being solved at step k is a submatrix of the Jacobian of P evaluated at x^k.

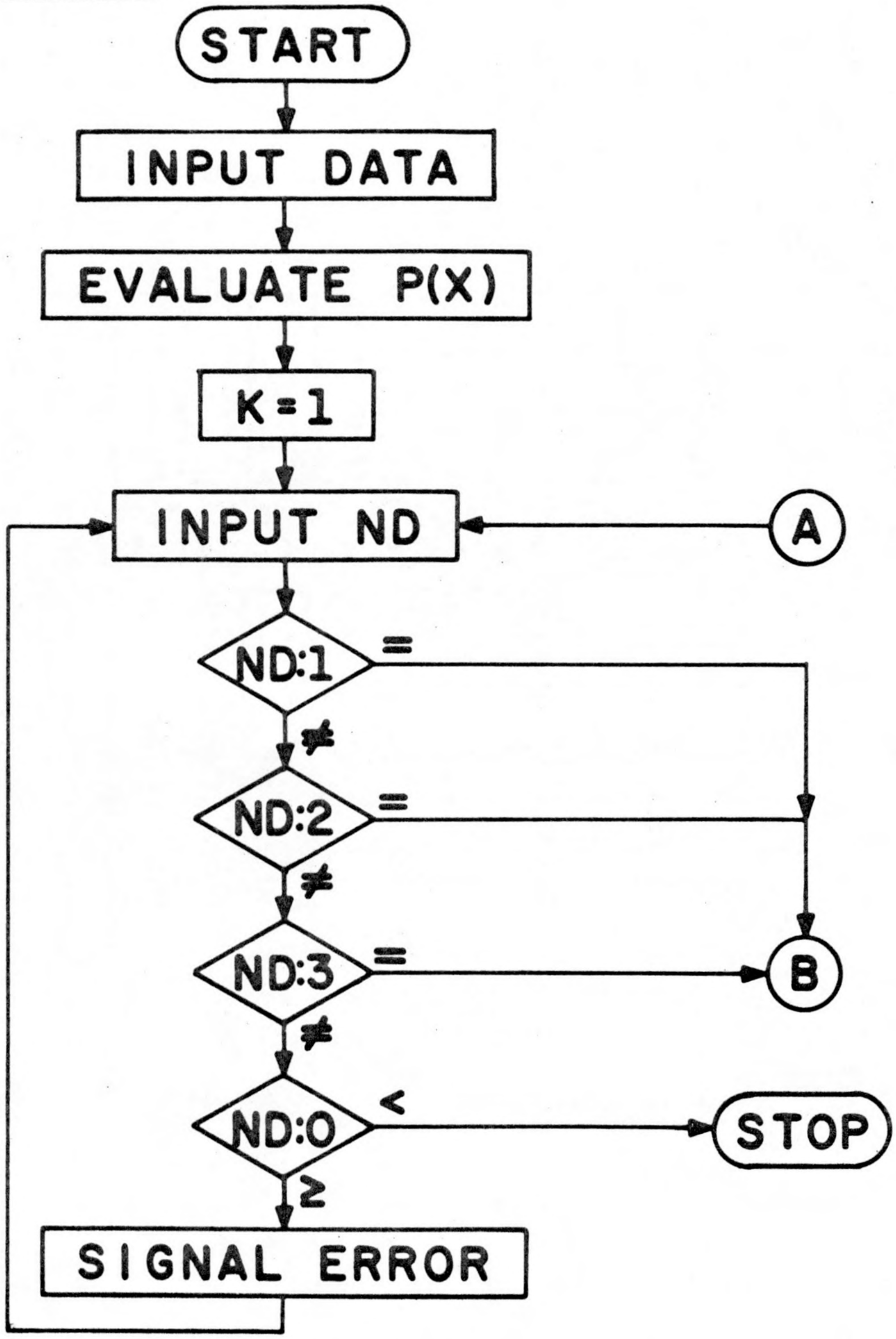
The user has the option with this program of projecting on 1, 2, or 3 dimensional subspaces. It should be noted that the sequence of projection spaces and the columns used to form the projection spaces does effect the rate of convergence.

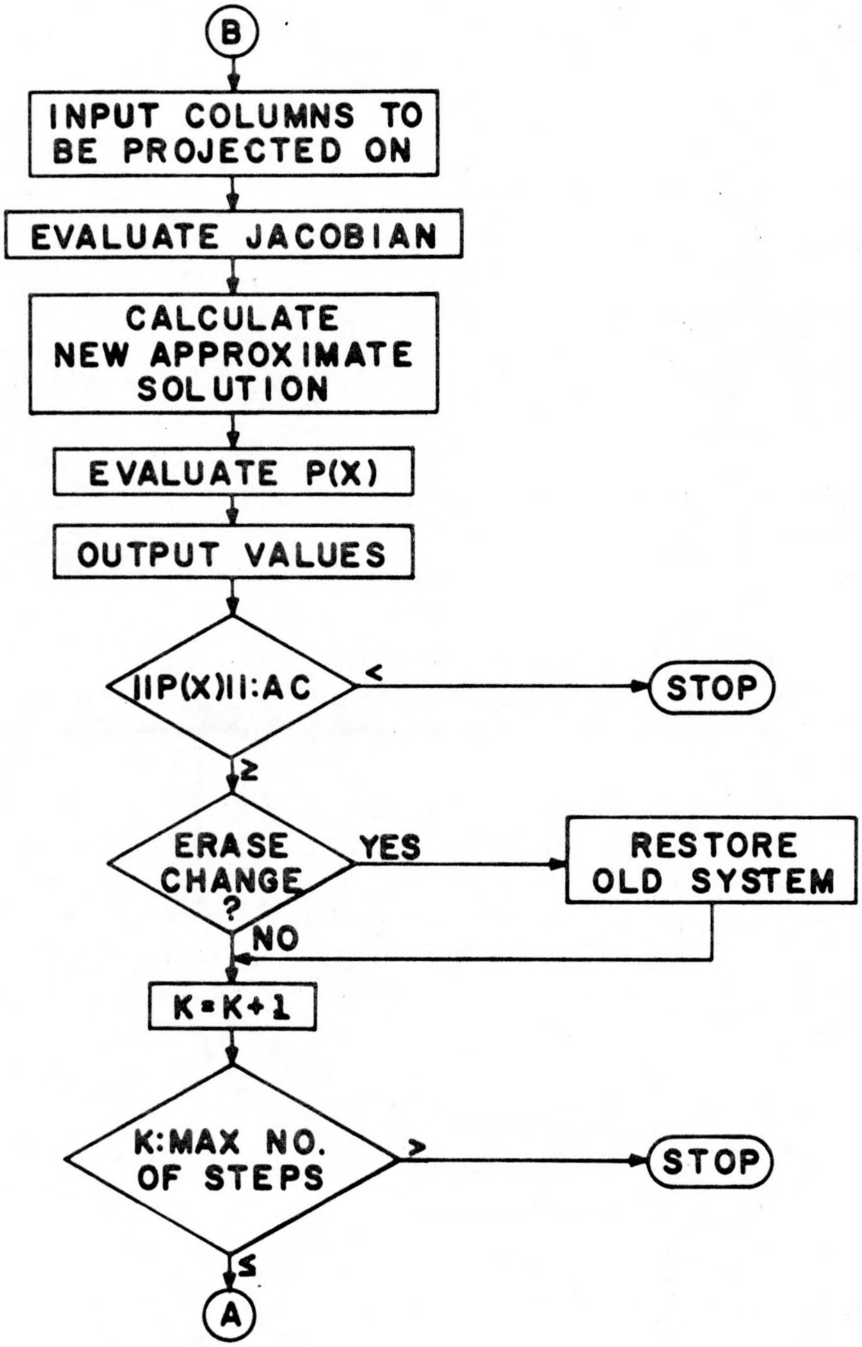
The nonzero components of the change vector are solved with Cramer's rule. The use of Cramer's rule is illustrated in the Technical Basis section for the linear projection method.

G. SYSTEM FLOWCHART



H. PROGRAM FLOWCHART




B
INPUT COLUMNS TO BE PROJECTED ON
EVALUATE JACOBIAN
CALCULATE NEW APPROXIMATE SOLUTION
EVALUATE P(X)
OUTPUT VALUES
||P(X)||:AC
<
STOP
≥
ERASE CHANGE ?
YES
RESTORE OLD SYSTEM
NO
K=K+1
K:MAX NO. OF STEPS
>
STOP
≤
A

I. PROGRAM LISTING

```
5000 DIM P(5),F(5,5),T(5)
5010 READ N,NS,AC
5020 DATA 5,5,.001
5030 MAT READ X(5,1)
5035 DATA 10,10,10,10,10
5040 LET M=N-1
5045 GOSUB 5207
5050 LET SF=0
5052 FOR I=1 TO N
5055 LET SF=SF+P(I)*P(I)
5060 NEXT I
5065 FOR IS=1 TO NS
5066 PRINT USING 5301, ' '
5067 PRINT USING 5301, ' '
5068 PRINT USING 5311, IS
5069 MAT R=X
5070 MAT O=P
5071 INPUT  ND
5072 LET SV=SF
5073 IF ND=1 THEN 5084
5074 IF ND=2 THEN 5097
5075 IF ND=3 THEN 5117
5076 IF ND<0 THEN 5083
5080 PRINT USING 5321, ' '
5082 GO TO 5071
5083 STOP
5084 INPUT  II
5085 GOSUB 5240
5086 GOSUB 5166
5087 LET SA,SB=0
5088 FOR I=1 TO N
5089 LET SA=SA+F(I,II)*F(I,II)
5090 LET SB=SB-P(I)*F(I,II)
5091 NEXT I
5092 LET XI=SB/SA
5093 LET X(II)=X(II)+XI
5094 GOSUB 5207
5095 GOSUB 5152
5096 GO TO 5143
5097 PRINT USING 5301, ' '
5098 INPUT  II,JJ
5099 GOSUB 5240
5100 GOSUB 5166
5101 LET SA,SB,SC,SD,SE=0
5102 FOR I=1 TO N
```

```
5102 FOR I=1 TO N
5103 LET SA=SA+F(I,II)*F(I,II)
5104 LET SB=SB+F(I,II)*F(I,JJ)
5105 LET SC=SC+F(I,JJ)*F(I,JJ)
5106 LET SD=SD-P(I)*F(I,II)
5107 LET SE=SE-P(I)*F(I,JJ)
5108 NEXT I
5109 LET TM=SA*SC-SB*SB
5110 LET XI=(SD*SC-SB*SE)/TM
5111 LET XJ=(SA*SE-SB*SD)/TM
5112 LET X(II)=X(II)+XI
5113 LET X(JJ)=X(JJ)+XJ
5114 GOSUB 5207
5115 GOSUB 5152
5116 GO TO 5143
5117 PRINT USING 5301, ' '
5118 INPUT  II,JJ,KK
5119 GOSUB 5240
5120 GOSUB 5166
5121 LET A1,A2,A3,A4,A5,A6=0
5122 LET SA,SB,SC=0
5123 FOR I=1 TO N
5124 LET A1=A1+F(I,II)*F(I,II)
5125 LET A2=A2+F(I,II)*F(I,JJ)
5126 LET A3=A3+F(I,II)*F(I,KK)
5127 LET A4=A4+F(I,JJ)*F(I,JJ)
5128 LET A5=A5+F(I,JJ)*F(I,KK)
5129 LET A6=A6+F(I,KK)*F(I,KK)
5130 LET SA=SA-P(I)*F(I,II)
5131 LET SB=SB-P(I)*F(I,JJ)
5132 LET SC=SC-P(I)*F(I,KK)
5133 NEXT I
5134 LET TM=A1*A4*A6+2*A5*A3*A2-A3*A3*A4-A5*A5*A1-A2*A2*A6
5135 LET XI=(SA*(A6*A4-A5*A5)+SB*(A3*A5-A2*A6)+SC*(A5*A2-A3*A4))/TM
5136 LET XJ=(SB*(A1*A6-A3*A3)+SA*(A5*A3-A2*A6)+SC*(A2*A3-A5*A1))/TM
5137 LET XK=(SC*(A1*A4-A2*A2)+SA*(A2*A5-A4*A3)+SB*(A2*A3-A5*A1))/TM
5138 LET X(II)=XI+X(II)
5139 LET X(JJ)=XJ+X(JJ)
5140 LET X(KK)=XK+X(KK)
5141 GOSUB 5207
5142 GOSUB 5152
5143 PRINT USING 5320, ' '
5144 INPUT  Y
5145 IF Y=1 THEN 5147
5146 GO TO 5150
5147 LET SF=SV
5148 MAT X=R
5149 MAT P=0
5150 NEXT IS
5151 STOP
```

```
 5152 PRINT USING 5301, ' '
 5153 PRINT USING 5154, ' '
 5154 IMAGE
X            P(X)              Theta(airk)
 5155 LET SF=0
 5156 FOR I=1 TO N
 5157 LET SF=SF+P(I)*P(I)
 5158 PRINT USING 5159, X(I),P(I),T(I)
 5159 IMAGE
.........    .........    .........
 5160 NEXT I
 5161 LET SG=SQR(SF)
 5162 PRINT USING 5163, SG
 5163 IMAGE
norm=........
 5164 IF SF<AC THEN 5155
 5165 RETURN
 5166 MAT A=ZER(N)
 5167 MAT C=ZER(N,N)
 5168 FOR I=1 TO N
 5169 LET SM,SJ=0
 5170 FOR J=1 TO N
 5171 LET SM=SM+F(J,I)*F(J,I)
 5172 LET SJ=SJ+F(J,I)*P(J)
 5173 NEXT J
 5174 LET A(I)=SM
 5175 LET T(I)=SJ/SQR(SF*A(I))
 5176 NEXT I
 5177 FOR I=1 TO M
 5178 FOR J=I+1 TO N
 5179 LET SM=0
 5180 FOR K=1 TO N
 5181 LET SM=SM+F(K,I)*F(K,J)
 5182 NEXT K
 5183 LET C(I,J)=1/(1-SM*SM/A(I)/A(J))
 5184 LET C(J,I)=C(I,J)
 5185 NEXT J
 5186 NEXT I
 5192 PRINT USING 5301, ' '
 5193 PRINT USING 5194, ' '
 5194 IMAGE
JACOBIAN:
 5195 FOR I=1 TO N
 5196 PRINT USING 5197, F(I,1),F(I,2),F(I,3),F(I,4),F(I,5)
 5197 IMAGE
...........    ...........    ...........    ...........    ...........
 5198 NEXT I
 5199 PRINT USING 5301, ' '
 5200 PRINT USING 5201, ' '
```

```
 5201 IMAGE
C:
 5202 FOR I=1 TO N
 5203 PRINT USING 5197, C(I,1),C(I,2),C(I,3),C(I,4),C(I,5)
 5204 NEXT I
 5205 PRINT USING 5301, ' '
 5206 RETURN
 5207 LET P(1)=X(1)*X(2)/X(3)+X(2)*X(3)/X(4)-1
 5208 LET P(2)=X(2)*X(3)/X(4)+X(3)*X(4)/X(5)-1
 5209 LET P(3)=X(3)*X(4)/X(5)+X(4)*X(5)/X(2)-1
 5210 LET P(4)=X(4)*X(5)/X(2)+X(5)*X(1)/X(3)-1
 5211 LET P(5)=X(5)*X(1)/X(3)+X(1)*X(2)/X(4)-1
 5212 RETURN
 5240 LET F(1,1)=X(2)/X(3)
 5241 LET F(1,2)=X(1)/X(3)+X(3)/X(4)
 5242 LET F(1,3)=-(X(1)*X(2))/(X(3)*X(3))+X(2)/X(4)
 5243 LET F(1,4)=-(X(2)*X(3))/(X(4)*X(4))
 5244 LET F(1,5)=0
 5245 LET F(2,1)=0
 5246 LET F(2,2)=X(3)/X(4)
 5247 LET F(2,3)=X(2)/X(4)+X(4)/X(5)
 5248 LET F(2,4)=-(X(2)*X(3))/(X(4)*X(4))+X(3)/X(5)
 5249 LET F(2,5)=-(X(3)*X(4))/(X(5)*X(5))
 5250 LET F(3,1)=0
 5251 LET F(3,2)=-(X(4)*X(5))/(X(2)*X(2))
 5252 LET F(3,3)=X(4)/X(5)
 5253 LET F(3,4)=X(3)/X(5)+X(5)/X(2)
 5254 LET F(3,5)=-(X(3)*X(4))/(X(5)*X(5))+X(4)/X(2)
 5255 LET F(4,1)=X(5)/X(3)
 5256 LET F(4,2)=-(X(4)*X(5))/(X(2)*X(2))
 5257 LET F(4,3)=-(X(5)*X(1))/(X(3)*X(3))
 5258 LET F(4,4)=X(5)/X(2)
 5259 LET F(4,5)=X(4)/X(2)+X(1)/X(3)
 5260 LET F(5,1)=X(5)/X(3)+X(2)/X(4)
 5261 LET F(5,2)=X(1)/X(4)
 5262 LET F(5,3)=-(X(5)*X(1))/(X(3)*X(3))
 5263 LET F(5,4)=-(X(1)*X(2))/(X(4)*X(4))
 5264 LET F(5,5)=X(1)/X(3)
 5265 RETURN
 5301 IMAGE
-
 5311 IMAGE
step number=----
 5321 IMAGE
error in dimension-

_
```

J. VARIABLE DICTIONARY

AC - accuracy goal

$$C(I,J) = 1/\left(1 - \frac{(F_i,F_j)^2}{(F_j,F_j)(F_i,F_i)}\right)$$

F - Jacobian matrix

II - column pointer

IS - step number

JJ - column pointer

KK - column pointer

N - dimension of the system

ND - number of dimensions projecting on

NS - maximum number of steps

P - P(x)

X - solution vector

XI - change in ii-th component of approximate solution vector

XJ - change in jj-th component of approximate solution vector

XK - change in kk-th component of approximate solution vector

$$T(I) = \frac{(P(x),F_i)}{((P(x),P(x))(F_i,F_i))^{\frac{1}{2}}}$$

IV. BIBLIOGRAPHY

1. D. Georg. Criteria for ordering columns in two-dimensional projection methods. Unpublished M.S. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1973.

2. R. F. Keller. A single-step gradient method for solving systems of linear equations. University of Missouri at Columbia, Math. Sci. Tech. Report No. 8, 1964.

3. R. F. Keller and I-Ming Shen. Acceleration of projection method for solving systems of linear algebraic equations. Unpublished paper presented at the Siam National Meeting, Denver, June 1970, Mimeo. Ames, Iowa, Iowa State University of Science and Technology, Department of Computer Science, 1970.

4. A. MacEachern. A generalized projection method for systems of non-linear equations. Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1971.

5. A. MacEachern and R. F. Keller. A projection method for solving non-linear systems of equations. Document Library, Ames Laboratory, USAEC, Ames, Iowa, Report No. IS-3040, 1972.

6. H. D. Pyron. A non-stationary two-dimensional acceleration for the one-dimensional projection method. Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1970.

7. I-Ming Shen. Acceleration of the projection method for solving systems of linear equations. Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1970.

8. M. Tokko. Optimal three-dimensional projection method for solving linear algebraic equations. Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1972.

9. M. White. A class of projection methods for solving systems of n non-linear equations in n unknowns allowing projections of order one through n. Unpublished Ph.D. thesis, Ames, Iowa, Library, Iowa State University of Science and Technology, 1971.

V. ACKNOWLEDGEMENTS

We wish to thank Mrs. Betty Dahlgren for the expert typing of this report.