

AD/A-006 431

**FAILURE ANALYSIS BY STATISTICAL  
TECHNIQUES (FAST). VOLUME II.  
USER'S MANUAL (APPENDICES)**

**William H. Rowan, et al**

**TRW Systems Group**

**Prepared for:**

**Defense Nuclear Agency**

**31 October 1974**

**DISTRIBUTED BY:**

**NTIS**

**National Technical Information Service  
U. S. DEPARTMENT OF COMMERCE**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER DNA 3336F-2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER AD/A-OC 6431	
4. TITLE (and Subtitle) FAILURE ANALYSIS BY STATISTICAL TECHNIQUES (FAST) Volume II, User's Manual (Appendices)		5. TYPE OF REPORT & PERIOD COVERED Final Report for Period 1 April 1973-30 April 1974	
		6. PERFORMING ORG. REPORT NUMBER 24006-001-RU-00	
7. AUTHOR(S) William H. Rowan                      Norman Lipner Jack J. Farrell                         John D. Cliver Aileen S. Nii		8. CONTRACT OR GRANT NUMBER(S) DNA 001-73-C-0222	
9. PERFORMING ORGANIZATION NAME AND ADDRESS TRW Systems Group One Space Park Redondo Beach, California 90278		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NWED Subtask Y99QAXSC061-21	
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D.C. 20305		12. REPORT DATE 31 October 1974	
		13. NUMBER OF PAGES 210	
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 This work sponsored by the Defense Nuclear Agency under Subtask Y99QAXSC061-21.			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  FAST  Reproduced by <b>NATIONAL TECHNICAL INFORMATION SERVICE</b> US Department of Commerce Springfield, VA. 22151  <b>PRICES SUBJECT TO CHANGE</b>			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This document is a user's manual for the FAST (Failure Analysis by Statistical Techniques) computer code. FAST is a technique for performing a statistical evaluation of the survivability of military or civilian systems subjected to a hostile environment such as would be caused by a nuclear burst or earthquake. The document contains a description of the analysis formulation, the inputs needed to perform system assessment studies, and the code and its operation, and a demonstration of the code operation through documentation of a sample			

DD FORM 1473  
1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

**UNCLASSIFIED**

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

**20. ABSTRACT (Continued)**

problem which has been designed to address the key aspects of the methodology and its practical application. The two volumes of the document are those of Volume I, User's Manual, and Volume II, User's Manual (Appendices). The code itself was developed by TRW Systems Group over the past decade, mainly for MINUTEMAN under sponsorship of the Air Force Space and Missile Systems Organization (SAMSO).

;

))

**UNCLASSIFIED**

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

## TABLE OF CONTENTS

APPENDIX A. Sample Problem Computer Printouts	1
APPENDIX B. Listing of FAST Computer Code	71
APPENDIX C. Flow Charts of FAST Program	151

### APPENDIX A

#### SAMPLE PROBLEM COMPUTER PRINTOUTS

This Appendix contains printouts from four computer runs:

- Baseline System
- Case of Experimental Design
- Case of Steepest Ascent
- Yield and Height of Burst Perturbations

The input cards for each run are identified by \$ comment cards.

The first run shows a complete computer printout, including the following:

- Subroutine USER
- Load Map
- Input Cards
- FAST Results and Plots
- Computer Control Cards (for TRW's CDC 6600)

The other three runs show only the input cards.

```

000002      SUBROUTINE USFR(LOP)
000002      COMMON/AZMUTH/AZPN
000002      EQUIVALENCE (AZRN,RN)
000002      COMMON/CT04 / R1,R2,R3,R4,R5,R6,R7,R8,R9,R10
000002      DIMENSION X1(25),Y1(25),X2(10),Y2(10),X3(9),Y3(9),X4(9),Y4(9)
000002      DATA X1/0.,30.,45.,56.,60.,70.,90.,110.,120.,135.,
*           150.,180.,210.,225.,236.,240.,250.,270.,290.,300.,304.,
*           315.,330.,360./
*           Y1/1.,.866,.707,.552,.578,.627,.667,.627,.578,.552,
*           .707,.866,1.,.866,.707,.552,.578,.627,.667,.627,
*           .578,.552,.707,.866,1./
*           DATA X2/0.,0.,45.,45.,1.,135.,135.,1.,225.,225.,1.,315.,1.,360./
*           Y2/1.,0.,1.,0.,0.50,0.50,1.,0.,1.,0.,0.50,0.50,1.,0.,1.,0./
*           DATA X3/0.,45.,50.,135.,180.,225.,270.,315.,360./
*           Y3/1.,1.,25.,1.,33,1.,25,1.,1.,25,1.,33,1.,25,1./
*           DATA X4/0.,45.,90.,135.,180.,225.,270.,315.,360./
*           Y4/1.,.63,.5.,.63,1.,.63,.5.,.63,1./
*           GO TO (10,20,30,40)ICP
10  AZ=RN*360.0
    CALL TABLE(X1,Y1,25,AZ)
    R1=R1*AZ
    RETURN
20  AZ=RN*360.0
    CALL TABLE(X2,Y2,10,AZ)
    R1=R1*AZ
    RETURN
30  AZ = RN*360.0
    CALL TABLE(X3,Y3,9,AZ)
    R1 = R1*AZ
    RETURN
40  AZ = RN*360.0
    CALL TABLE(X4,Y4,9,AZ)
    R1 = R1*AZ
    RETURN
    END
*FOLLOWING VARIABLES EQUIVALENCED BUT NOT REFFERENCED
AZRN

```

SUBPROGRAM LENGTH  
000233

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS  
10 - 000013 20 - 000022 30 - 000031 40 - 000040

BLOCK NAMES AND LENGTHS  
AZMUTH - C00001 CT04 - C00012

VARIABLE ASSIGNMENTS  
AZ - 000232 RP - 00000001 F1 - 00000002 X1 - 000060  
X2 - 000142 X3 - 000166 X4 - 000210 Y1 - 000111  
Y2 - 000154 Y3 - 000177 Y4 - 000221

START OF CONSTANTS  
000051

START OF TEMPORARIES  
000056

START OF INDIRECTS  
000060

UNUSED COMPILER SPACE  
026300

LCAD OPTIONS - Y

FL REQUIRED TO LOAD 36000  
 FL REQUIRED TO RUN 65600  
 INITIAL TRANSFER TO MAIN - 101

BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE
MAIN	100	1032	INPUT	01/26/74.
NLGET	1132	264	INPUT	01/04/74.
NLNAME	1416	263	INPUT	01/04/74.
NLVECI	1701	33	INPUT	01/04/74.
NLVECH	1734	33	INPUT	01/04/74.
NLINTG	1767	54	INPUT	01/04/74.
NLREAL	2043	41	INPUT	01/04/74.
NLNUMB	2104	252	INPUT	01/04/74.
RESET	2356	70	INPUT	01/04/74.
NLREAD	2446	427	INPUT	01/04/74.
CCMPNT	3075	1310	INPUT	01/04/74.
BCOLE	4405	633	INPUT	01/04/74.
LOGIC	5240	325	INPUT	01/04/74.
NETCK	5565	526	INPUT	01/04/74.
SELECT	6313	356	INPUT	01/04/74.
RTOP	6671	142	INPUT	01/04/74.
PTGR	7034	156	INPUT	01/04/74.
PRCG	7212	1736	INPUT	01/16/74.
RNVD	11150	526	INPUT	01/04/74.
PRCMED	11706	433	INPUT	01/16/74.
TABLE	12341	46	INPUT	01/15/74.
PROBL	12407	113	INPUT	01/15/74.
DCPLOT	12522	126	INPUT	01/04/74.
SPLUT	12660	111	INPUT	01/04/74.
PHPLOT	12771	152	INPUT	01/04/74.
RCHD	13144	120	INPUT	01/04/74.
CAN	13264	100	INPUT	01/04/74.

LCAD MAP. PROGRAM MAIN  
BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE
USER	13364	253	IGO	03/07/74.
ANPN	13617	233	RUNLIB	*NO-DATE*
FNKN	14052	747	RUNLIB	*NO-DATE*
UPRI	15021	10	RUNLIB	*NO-DATE*
ACGOFR	15031	12	RUNLIB	*NO-DATE*
ALNLOG	15042	67	RUNLIB	*NO-DATE*
EXP	15132	57	RUNLIB	*NO-DATE*
FILMGR	15211	167	RUNLIB	11/11/73.
IPAIEX	15400	31	RUNLIB	*NO-DATE*
INPUTC	15431	225	RUNLIB	08/01/73.
CLTPTC	16656	1356	RUNLIB	07/24/73.
RBAREX	20234	41	RUNLIB	*NO-DATE*
RBAREX	20275	57	RUNLIB	*NO-DATE*
SQRT	20354	43	RUNLIB	*NO-DATE*
SYSTEM	20417	1221	RUNLIB	02/03/74.
SYSTEMF	21640	17	RUNLIB	07/25/73.
SYSTEMQ	21657	43	RUNLIB	11/26/72.
CPUCDD	21722	15	RUNLIB	05/28/71.
CFUCIUX	21735	16	RUNLIB	07/23/73.
CFURDW	21753	254	RUNLIB	04/05/73.
CFUWTW	22227	163	RUNLIB	06/10/73.
CPUSYS	22412	24	RUNLIB	10/10/71.
/FIJRUF1/	22430	4044		
/WORKEY/	26502	213		
/ICUNIT/	26715	4		
/CARDAT/	26721	136		
/CHRLST/	27057	57		
/NLIST/	27136	333		
/CE01/	27471	177		
/CF02/	27663	74		
/CE03/	27757	74		
/CE04/	30053	36		
/CE05/	30111	74		
/CE06/	30205	36		
/CE09A/	30245	76		

LCAD MAP. PROGRAM  
BLOCK ASSIGNMENTS.

MAIN

03/07/74. 10.27.40. PAGE

3

BLOCK ADDRESS LENGTH FILE DATE

/CE09B/	30301	36		
/CE06A/	30337	721		
/CE06B/	31260	721		
/CE06C/	32201	721		
/CE06D/	33122	36		
/CE07A/	33160	721		
/CE07B/	34101	721		
/CE07C/	35022	721		
/CE07D/	25743	721		
/CE07E/	36664	36		
/CE08/	36722	1604		
/CR01/	40526	22		
/CR02/	40550	32		
/CR03/	40602	26		
/CNC1/	40630	212		
/CNC2/	41042	416		
/CNC3/	41460	230		
/CC01/	41710	1442		
/CC02/	43352	1130		
/CC03/	44502	1130		
/CC04/	45632	1130		
/CC05/	46762	1130		
/CC06/	50112	1130		
/CC07/	51242	1440		
/CN01/	52702	1752		
/CT01/	54654	36		
/CT02/	54712	36		
/CT03/	54750	36		
/CT04/	55006	36		
/CT05/	55044	207		
/CT06/	55252	5632		
/CT07/	63105	1277		
/CBETA/	64404	211		
/AZMUTH/	64715	1		
/CT07R/	64716	572		

LOAD MAP. PROGRAM MAIN  
BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE
/FI08UFR/	65310	230		

COMMON BLOCK REFERENCES.

BLOCK	ADDRESS	REFERENCES	USER	LOGIC	MAIN	NLGET	NLINTG	NLNAME	NLNUMB
/AZMUTH/	64715	PROG	USER						
/CARDAT/	26721	ROOLE	NLREAL						
		NLRFAD	PROG						
/CBETA/	64404	MAIN	PROG						
/CC01/	41710	COMPNT	MAIN						
/CC02/	43352	COMPNT	MAIN						
/CC03/	44502	COMPNT	MAIN						
/CC04/	45632	COMPNT	MAIN						
/CC05/	46762	COMPNT	MAIN						
/CC06/	50112	COMPNT	MAIN						
/CC07/	51242	COMPNT	MAIN						
/CE01/	27471	COMPNT	MAIN						
/CE02/	27663	MAIN	NLREAD						
/CE03/	27757	MAIN	NLREAD						
/CE04/	30053	MAIN	NLREAD						
/CE05/	30111	MAIN	NLREAD						
/CE06A/	30337	MAIN	NLREAD						
/CE06B/	31260	MAIN	NLREAD						
/CE06C/	32201	MAIN	PROG						
/CE06D/	33122	MAIN	PROG						
/CE07A/	33160	MAIN	PROG						
/CE07B/	34101	MAIN	PROG						
/CE07C/	35022	MAIN	NLREAD						
/CE07D/	35743	MAIN	PROG						
/CE07E/	36664	MAIN	PROG						
/CE08/	36722	MAIN	PROG						
/CE09/	30205	MAIN	NLREAD						
/CE09A/	30243	MAIN	NLREAD						

LLAD MAP. PROGRAM MAIN  
COMMON HLK REFERENCES.

BLOCK	ADDRESS	REFERENCES	ALREAD	COMPNT	SPLOT	CCMPNT	SELECT	LOGIC	MAIN	NETCK	NLGET	ALNAME
/CT094/ /CHRST/	30301 27057	MAIN MODULE	ALREAD COMPNT	COMPNT	SPLOT	CCMPNT	SELECT DUPLOT	LOGIC	MAIN	NETCK	NLGET	ALNAME PROMED
/CNC1/	40530	MODULE RESET	COMPNT	LOGIC	ALREAD	MAIN	LUPLOT	LOGIC	MAIN	NETCK	PRG	
/CNC2/	41042	MODULE	COMPNT	LOGIC	ALREAD	MAIN	MAIN	NETCK	PRG	NETCK	PRG	
/CNC3/	41460	DUPLOT	LOGIC	COMPNT	ALREAD	MAIN	MAIN	NETCK	PRG	PRG	RESET	
/CNO1/	52702	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CNO1/	40526	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CRD2/	40550	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CRD2/	40550	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CP03/	40602	DUPLOT	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CT01/	54654	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CT02/	54712	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CT03/	54750	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CT04/	55006	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CT05/	55044	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CT06/	55253	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CT07/	63105	DUPLOT	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/CT07A/	64716	DUPLOT	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/FI00FF/	25310	INPUTC	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/FI00FF/	22436	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/ICUNIT/	26715	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/NL1ST/	27136	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		
/MURKEY/	26502	MODULE	COMPNT	LOGIC	ALREAD	MAIN	PRG	MAIN	NETCK	PRG		

ENTRY POINTS.

HLK/ENTRY ADDRESS	REFERENCES	HLK/ENTRY ADDRESS	REFERENCES
ACGOER 15071	MODULE	ACGOER 15032	MODULE
ACGOER 15032	RNVD	ALNLOG 15043	RNVD

LCAD MAP. PROGRAM  
ENTRY POINTS.

MAIN

REFERENCES

BLK/ENTRY ADDRESS	REFERENCES	PRPG ALPFAI	RBAREX PRJG	SELECT SELECT
ALUG 15047	ANRN			
ALOG10 15044	CAN			
ANRN 13617				
ANRN 13620				
BCULE 4405				
BJOLE 4406	ALREAD			
CAN 13264				
CAN 13265	PRJG			
CCMPNT 3075				
COMPNT 3076	ALREAD			
CFUGDD 21722				
CDD 21722	INPUTC	CLPTC		
CFUCIOX 21735				
CIO= 21740	CPURDW	CPURDW	FIMGR	INPUTC
CPURDW 21753				OUTPTC
LCB\$ 21754				
RDC= 22J05				
KDH= 22034				
RDO= 22133	INPUTC			
RDS= 22064				
RDS\$ 22147				
KDW= 22147				
RDX\$ 22167				
SKW\$ 22214				
CFUSYS 22412				
MSG= 22433				
RCL= 22420	CPUCIOX	CPURDW	CPURDW	FILMGR
SYS= 22413	CPUCIOX	FIMGR	CPURDW	SYSTEM
WNR= 22424				
CFLATW 22227				
LCG\$ 22230				
WTC= 22242				
WTH= 22261				
WTH= 22347				
WTS= 22313	OUTPTC			



ENTRY POINTS.

RUK/ENTRY ADDRESS	REFERENCES	ALNAME
NLREAL 2043	COMPNT NLFFAD NLVECR	
NLREAL 2044		
NLVECT 1701	NLHEAD	
NLVECT 1702		
NLVECP 1734		
NLVECP 1735		
CUTPTC 16056		
KOPER 17026		
CUTPTC 16060		
PFCHL 12407	COMPNT PRJG PRCMED	
PROBL 12410		
PRG 7212		
PRG 7213	NLHEAD	
PRMED 11706		
PRMED 11707		
PPPLJT 12771		
PPPLJT 12772		
PTOR 7034		
PTOR 7035		
RFALX 20234	SFLECT	
RBATEX 20235	NLJMP NLPEAL	
RRAREX 20275		
RRAREX 20276		
RESET 2356		
RESET 2357		
RNVD 11150	NLHEAD	
RNVD 11151		
RCMD 12144		
ROKU 12145		
RTCP 6671		
RTUP 6672		
SFLECT 6313		
SFLECT 6314		
SPLJT 12660		

LEAD MAP: PROGRAM  
ENTRY POINTS.

MAIN CE/C7/74. 10.37.46. PAGE 9

PLK/ENTRY ADDRESS REFERENCES

PLK/ENTRY ADDRESS	REFERENCES
SPLT 12601	COPLT
SCRT 20354	ANPN
SQRT 20355	ANPN
SYSTEM 20417	ACQDER
ANORML 20716	INPUTC
AERR 21354	ANRN
ASA. 20604	NLGET
END 20642	PROBL
	RTOP
EXIT 20662	INPUTC
FL. 20605	
LDRUSX= 20563	
LINESIZ 20635	
LINESZ. 20640	
NTEMP 21351	
PAUSE 20670	
CHTRY 20420	
STOP 20703	
SYSTEM 20767	MAIN MAIN ACQDER SQRT
SYSTEMC 20733	
SYSTEMP 20762	
TERM 21346	
SYSTEMF 21640	
SYSTEMM 21640	
SYSTEMQ 21657	
DEBUG= 21657	
DEBUG= 21357	
SMPDVL= 21657	
SYSABT= 21661	
SYSTEM= 21706	SYSTEM SYSTEM
TABLE 12341	USER
TABLE 12342	

Reproduced from  
best available copy. 6

ENTRY POINTS.

BLK/ENTRY ADDRESS REFERENCES

UPK1	15021		
UPF1	15022	ANRN	PROG
USER	13364		
USER	13365	PROG	

\$ DNA SAMPLE PULPER - BASELINE SYSTEM

PLOTON=400\$

\$ ELECTRIC RACKS

C101=/0,30,35,40/1,40,45,50/R1=12,2,1.2\$  
 C102=/0,15,17,19/1,19,21,23/R1=17/USER(1)/R1=R1,1.0,1.2\$  
 C103=/0,50/1,50/R1=18,1.0,1.2\$

\$ BATTERY MOUNTS

C601=/0,50C,550,600/1,600,650,700/R1=4,1.0,1.2\$  
 C602=/0,175,200,225/1,225,250,275/R1=8/USER(2)/R1=R1,1.0,1.2\$  
 C603=/0,20/1,20/R1=1,1,1/USFF(4)\$  
 C604=/0,10/1,10/R1=6,1,1\$

\$ BATTERY ELEMENTS

C705=/0,45C,500,550/1,600,650,700/R1=2,1,1.2\$  
 C706=/0,15C,175,200/1,225,250,275/R1=8,1,1.2\$

\$ SIS

C131=/0,30,35,40/1,35,40,45/R1=1,1,1.2\$  
 C132=/0,30,35,40/1,35,40,45/R1=5,2,1.3\$  
 C133=/0,30,35,40/1,35,40,45/R1=4,.08,1.2\$

\$ COMMUNICATIONS

C2 = /0,.7237,.9408,1.223/.0668,.7416,.9641,1.253/.3065,.7599,.9379,1.284/  
 .6915,.7787,1.012,1.316/.9332,.7979,1.037,1.348/1,.8176,1.063,1.352/  
 R1=11,.000400,1.3/USER(3)\$  
 C3 = /0,.4704,.9408,1.882/.0668,.4820,.5641,1.928/.3085,.4939,.5879,1.976/  
 .6915,.5061,1.012,2.024/.9332,.5286,1.037,2.074/1,.5314,1.063,2.126/  
 R1=19,.04,1.4\$

\$ BLAST VALVE

C5 = /0,.6125,.7962,1.035/.0668,.6709,.8722,1.134/.5085,.7350,.9554,1.242/  
 .6915,.8051,1.047,1.361/.9332,.8819,1.146,1.490/1,.9661,1.256,1.633/  
 R1=10,.0002,1.3\$

\$ FACILITY

C9 = /0,.6635,.7962,.9554/.0668,.7268,.8722,1.047/.3085,.7962,.9554,1.146/  
 .6915,.8722,1.047,1.256/.9332,.9554,1.146,1.376/1,1.047,1.256,1.577/  
 R1=14,.0015,1.1\$  
 C11 = /0,.6003,.7204,.8645/.0668,.6845,.8214,.9956/.3085,.7804,.9365,1.124/  
 .6915,.8858,1.068,1.281/.9332,1.014,1.217,1.461/1,1.157,1.388,1.666/  
 R1=15,.00135,1.2\$  
 C12 = /0,.5472,.6566,.7380/.0668,.6475,.7770,.9324/.3085,.7661,.9153,1.103/  
 .6915,.9065,1.089,1.305/.9332,1.012,1.287,1.544/1,1.269,1.523,1.827/

R1=16, .0C12 +1.3\$  
 \$ AIR CONDITIONER  
 C8 = /0, .6125, .7962, 1.0315, .0668, .6709, .8722, 1.1347, .3085, .7350, .9554, 1.2427  
 .6915, .8051, 1.047, 1.3617, .5332, .8815, 1.146, 1.4907, .9661, 1.256, 1.6337  
 R1=12, .01, 1.4\$  
 SS1=101+102+103\$  
 SS2 = (131+132)\*133\$  
 SS3=601\*602+602\*604\$  
 SS4=705+706\$  
 SS5=(2+3)\*5\$  
 SS6=9\$  
 SS7=10+11+12\$  
 SS8=8\$  
 SYS100=SS1+SS2+SS3+SS4+SS5+SS6+SS7+SS8\$  
 SYS1=SS1+SS2+SS3+SS4+SS5+SS6+SS7\$  
 MODE=16\*4, 3\*1

ULN=2.785, 5.94, 4.094, 6.346, 2.639, 1.435, 3.807, 4.205, 0.647, 7.783, 7.244,  
 4.605, 5.258, 6.397, 6.357, 6.397, 3.296, 3.367, 2.485  
 ALPHA=.17, .083, .083, .083, .25, .17, .063, .083, 0, .08, .04, .04, .08,  
 .17, 1.33, .67, 1.58

15

BETA =.67, 1.1, 1.1, .23, .67, .5, .5, .33, 1.8, 1.37, 5.1, .2, .2, -3.86			
S 1 =	.1927,	.1767,	.0894,
	.3448,	.0085,	.2031,
	.0000,	0.0000,	.0015,
S 2 =	.1765,	.1713,	.1767,
	.3445,	.0083,	.0013,
	.0000,	.0000,	.0013,
S 3 =	.0806,	.0840,	.0854,
	.0114,	.0083,	.0013,
	.0000,	.0000,	.0013,
S 4 =	.1765,	.1713,	.1767,
	.0083,	.0013,	.0013,
	.2228,	.0039,	.0084,
S 5 =	.0031,	.0039,	.0084,
	.2099,	.0111,	.0170,
	.0015,	.0015,	.0116,
S 6 =	.0043,	.0042,	.0042,
	.0042,	.0042,	.0042,
	.0043,	.0042,	.0042,
S 7 =	.0042,	.0042,	.0042,
	.0042,	.0042,	.0042,
	.0042,	.0042,	.0042,
S 8 =	.0042,	.0042,	.0042,
	.0042,	.0042,	.0042,
	.0042,	.0042,	.0042,
S 9 =	.0042,	.0042,	.0042,
	.0042,	.0042,	.0042,
	.0042,	.0042,	.0042,

S 10 =	.6745,	.0205,	.0150,	.0010,	.0010,	.0423,	0.0000,
S 11 =	-.1708,	C.0000,	.0114,	.0114,	.0114,	0.0000,	0.0000,
S 12 =	C.0000,	.0082,	.0082,	.0000,	.0000,	0.0000,	0.0000,
S 13 =	.0083,	.0083,	.0083,	0.0000,	0.0000,	0.0000,	0.0000,
S 14 =	.0113,	.0075,	C.0000,	0.0000,	0.0000,	0.0000,	0.0000,
S 15 =	.0118,	.0077,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
S 16 =	.0154,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
S 17 =	.0230,	.0107,	.0160,				
S 18 =	.0824,	.0091,					
S 19 =	.0393,						
K-FAC=	2.52,	1.86,	1.96,	4.95,	2.52,	2.03,	2.12,
A 1 =	1.56,	1.56,	1.62,	1.65,	3.02,	2.86,	2.31,
A 2 =	1.0000,	.6123,	.2679,	.5640,	.9087,	.0567,	.0532,
A 3 =	1.0000,	.6125,	.1050,	.0621,	.0821,	.0330,	.1549,
A 4 =	1.0000,	.0000,	C.0000,	.3005,	.5317,	.0788,	-.2576,
A 5 =	1.0000,	.5144,	.9213,	.1203,	.0611,	.2295,	0.0000,
A 6 =	1.0000,	-.2819,	.1053,	.2327,	.0526,	-.1000,	.3563,
A 7 =	1.0000,	1.0000,	.1097,	.0557,	.0429,	0.0000,	-.0274,
A 8 =	1.0000,	.1404,	.4898,	.0775,	.0726,	.0424,	.1418,
A 9 =	1.0000,	.1108,	.0563,	.0434,	.2114,	-.2597,	0.0000,
A 10 =	1.0000,	.7691,	.0734,	.0219,	-.4642,	.0221,	.0173,
A 11 =	1.0000,	-.0113,	-.0087,	.0630,	0.0000,	0.0000,	0.0000,
A 12 =	1.0000,	.0493,	.0462,	-.5724,	.5319,	.0911,	.0713,
A 13 =	1.0000,	.0286,	.1345,	.7770,	-.1637,	0.0000,	.0629,
A 14 =	1.0000,	.5373,	.0471,	.0000,	.0727,	.0678,	.0530,
A 15 =	1.0000,	.0485,	.7283,	.0682,	-.0156,	0.0000,	.0455,
A 16 =	1.0000,	.0441,	C.0000,	-.0147,	C.0000,	.0497,	.0590,
A 17 =	1.0000,	-.2669,	-.0011,	-.0000,	-.0009,	.0281,	-.0443,
A 18 =	1.0000,	.0676,	C.0000,	.1126,	.0496,	.2275,	0.0000,
A 19 =	1.0000,	.1440,	.2151,	.2555,	.1570,	C.0000,	0.0000,
A 20 =	1.0000,	.2151,	.1957,	.1540,	.1495,	C.0000,	0.0000,
A 21 =	1.0000,	.3348,	.1540,	.1455,	0.0000,	C.0000,	0.0000,
A 22 =	1.0000,	.1997,	.4210,	C.0000,	0.0000,	C.0000,	0.0000,
A 23 =	1.0000,	.3557,					

A 15 = 1.C0J0, .2069, 0.0000, 0.0000, 0.0000,  
 A 16 = 1.C0J0, 0.0000, 0.0000, 0.0000, 0.0000,  
 A 17 = 1.C0J0, .4572, .4179,  
 A 18 = 1.CC00, .2225,  
 A 19 = 1.0000,

NOMP#H=6CC,58

W=5

OP=200,400,500,600,700,800,900,1000,1200

RUN\$







101	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
INDIVIDUAL SUPSYSTEMS												
NAME	1	10	30	50	70	90	99					
8	1.000	1.000	1.000	1.000	1.000	1.000	1.000					
7	1.000	1.000	1.000	1.000	1.000	1.000	1.000					
6	1.000	1.000	1.000	1.000	1.000	1.000	1.000					
5	1.000	1.000	1.000	1.000	1.000	1.000	1.000					
4	1.000	1.000	1.000	1.000	1.000	1.000	1.000					
3	1.000	1.000	1.000	1.000	1.000	1.000	1.000					
2	1.000	1.000	1.000	1.000	1.000	1.000	1.000					
1	1.000	1.000	1.000	1.000	1.000	1.000	1.000					

INDIVIDUAL SYSTEMS	1	10	30	50	70	90	94
NAME	1	10	30	50	70	90	94
-100	1.000	1.000	1.000	1.000	1.000	1.000	1.000
-1	1.000	1.000	1.000	1.000	1.000	1.000	1.000
ULN( 0)=	4.00000E+02						
ULN( 1)=	1.23461E+01						
ULN( 2)=	2.53290E+02						
ULN( 4)=	3.80135E+02						
ULN( 5)=	1.22460E+01						
ULN( 6)=	3.20060E+00						
ULN( 8)=	5.47221E+01						
ULN(10)=	1.15451E+03						
ULN(11)=	8.03127E+02						
ULN(12)=	6.66553E+01						
ULN(13)=	1.33251E+02						
ULN(14)=	4.00028E+02						
ULN(15)=	4.00028E+02						
ULN(16)=	4.00028E+02						
ULN(17)=	1.73100E+01						
ULN(18)=	2.15532E+01						
ULN(19)=	6.77202E+00						

W= 5.0000E+00 H= C.  
 NO CONVERGENCE - NI= 0 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 1 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 2 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 3 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 4 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 5 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 6 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 7 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 8 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 9 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 10 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 11 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 12 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 13 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 14 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 15 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 16 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 17 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 18 EPS= 5.0E-02  
 NO CONVERGENCE - NI= 19 EPS= 5.0E-02

NO CONVERGENCE	- NI=	7 NC=	4 FPS=	5.0E-02
NO CONVERGENCE	- NI=	7 NC=	5 FPS=	5.0E-02
NO CONVERGENCE	- NI=	6 ND=	6 FPS=	5.0E-02
NO CONVERGENCE	- NI=	5 NI=	7 FPS=	5.0E-02
NO CONVERGENCE	- NI=	6 NC=	7 FPS=	5.0E-02
NO CONVERGENCE	- NI=	6 ND=	8 FPS=	5.0E-02
NO CONVERGENCE	- NI=	7 ND=	9 FPS=	5.0E-02
NO CONVERGENCE	- NI=	12 NG=	10 FPS=	5.0E-02
NO CONVERGENCE	- NI=	16 ND=	10 FPS=	5.0E-02
NO CONVERGENCE	- NI=	18 NU=	10 FPS=	5.0E-02
NO CONVERGENCE	- NI=	19 NU=	10 FPS=	5.0E-02
NO CONVERGENCE	- NI=	20 ND=	10 FPS=	5.0E-02
NO CONVERGENCE	- NI=	20 ND=	10 FPS=	5.0E-02
NO CONVERGENCE	- NI=	16 ND=	11 FPS=	5.0E-02
NO CONVERGENCE	- NI=	19 NC=	12 FPS=	5.0E-02
NO CONVERGENCE	- NI=	19 NC=	12 FPS=	5.0E-02
NO CONVERGENCE	- NI=	17 ND=	13 FPS=	5.0E-02
NO CONVERGENCE	- NI=	15 NC=	14 FPS=	5.0E-02
NO CONVERGENCE	- NI=	15 ND=	15 FPS=	5.0E-02
NO CONVERGENCE	- NI=	16 ND=	16 FPS=	5.0E-02
NO CONVERGENCE	- NI=	16 ND=	16 FPS=	5.0E-02
NO CONVERGENCE	- NI=	16 ND=	17 FPS=	5.0E-02
NO CONVERGENCE	- NI=	15 ND=	18 FPS=	5.0E-02
NO CONVERGENCE	- NI=	14 ND=	19 FPS=	5.0E-02
NO CONVERGENCE	- NI=	13 ND=	20 FPS=	5.0E-02
NO CONVERGENCE	- NI=	12 ND=	21 FPS=	5.0E-02
NO CONVERGENCE	- NI=	12 ND=	22 FPS=	5.0E-02
NO CONVERGENCE	- NI=	12 ND=	23 FPS=	5.0E-02
NO CONVERGENCE	- NI=	13 ND=	23 FPS=	5.0E-02
NO CONVERGENCE	- NI=	13 ND=	24 FPS=	5.0E-02
NO CONVERGENCE	- NI=	15 ND=	25 FPS=	5.0E-02
NO CONVERGENCE	- NI=	15 ND=	25 FPS=	5.0E-02
NO CONVERGENCE	- NI=	14 ND=	26 FPS=	5.0E-02
NO CONVERGENCE	- NI=	17 ND=	27 FPS=	5.0E-02
NO CONVERGENCE	- NI=	15 ND=	27 FPS=	5.0E-02
NO CONVERGENCE	- NI=	18 NC=	27 FPS=	5.0E-02
NO CONVERGENCE	- NI=	17 ND=	28 FPS=	5.0E-02
NO CONVERGENCE	- NI=	19 ND=	29 FPS=	5.0E-02
NO CONVERGENCE	- NI=	18 ND=	24 FPS=	5.0E-02
NO CONVERGENCE	- NI=	18 ND=	30 FPS=	5.0E-02
NO CONVERGENCE	- NI=	17 ND=	31 FPS=	5.0E-02
NO CONVERGENCE	- NI=	20 ND=	32 FPS=	5.0E-02

NO CONVERGENCE	-	NI=	20	NO=	32	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	19	NO=	33	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	34	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	35	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	17	NO=	36	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	19	NO=	37	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	19	NO=	37	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	38	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	39	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	17	NO=	40	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	17	NO=	41	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	42	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	42	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	43	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	44	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	45	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	20	NO=	46	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	19	NO=	46	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	19	NO=	47	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	19	NO=	48	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	49	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	18	NO=	50	EPS=	9.0E-04
NO CONVERGENCE	-	NI=	19	NO=	50	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	19	NO=	50	EPS=	5.0E-02
NO CONVERGENCE	-	NI=	19	NO=	100	EPS=	9.0E-04
NO CONVERGENCE	-	NI=	19	NO=	117	EPS=	9.0E-04
NO CONVERGENCE	-	NI=	19	NO=	124	EPS=	9.0E-04
NO CONVERGENCE	-	NI=	19	NO=	132	EPS=	4.0E-02
NO CONVERGENCE	-	NI=	19	NO=	182	EPS=	4.0E-02
NO CONVERGENCE	-	NI=	19	NO=	232	EPS=	4.0E-02
NO CONVERGENCE	-	NI=	19	NO=	282	EPS=	4.0E-02
NO CONVERGENCE	-	NI=	19	NO=	332	EPS=	4.0E-02
NO CONVERGENCE	-	NI=	19	NO=	382	EPS=	4.0E-02
NO CONVERGENCE	-	NI=	19	NO=	432	EPS=	4.0E-02
NO CONVERGENCE	-	NI=	19	NO=	482	EPS=	4.0E-02
NO CONVERGENCE	-	NI=	19	NO=	532	EPS=	4.0E-02
NO CONVERGENCE	-	NI=	19	NO=	582	EPS=	4.0E-02
NO CONVERGENCE	-	NI=	19	NO=	625	EPS=	4.0E-02

PS=0 (----- HISTOGRAM (IN STEPS OF .05) FOR PS GREATER THAN 0 AND LESS THAN -----) PS=1

13	3	2	1	1	1	4	4	2	1	5	4	8	6	7	7	7	8	6	534	101		
35	20	17	11	13	5	12	11	4	8	11	21	14	13	14	9	17	26	20	37	45	262	102



2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3	.604	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
7C5	.803	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
601	0.000	.412	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10	.587	.945	.986	1.000	1.000	1.000	1.000	1.000	1.000
11	.481	.914	.976	1.000	1.000	1.000	1.000	1.000	1.000
12	.612	.954	.988	1.000	1.000	1.000	1.000	1.000	1.000
8	.020	.535	.553	1.000	1.000	1.000	1.000	1.000	1.000
101	0.000	.796	1.000	1.000	1.000	1.000	1.000	1.000	1.000

INDIVIDUAL SUBSYSTEMS

NAME	1	10	30	50	70	90	59
7	.291	.786	.953	.977	1.000	1.000	1.000
8	.020	.535	.958	1.000	1.000	1.000	1.000
5	.958	1.000	1.000	1.000	1.000	1.000	1.000
4	.803	1.000	1.000	1.000	1.000	1.000	1.000
6	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	0.000	.434	.990	1.000	1.000	1.000	1.000
1	0.000	.561	.966	1.000	1.000	1.000	1.000
3	.456	1.000	1.000	1.000	1.000	1.000	1.000

INDIVIDUAL SYSTEMS

NAME	1	10	30	50	70	90	59
-100	.002	.061	.453	.787	.943	.999	1.000
-1	0.000	.133	.623	.872	.968	1.000	1.000
ULN( 0) =	5.00000E+02						
ULN( 1) =	1.43370E+01						
ULN( 2) =	3.16612E+02						
ULN( 4) =	4.75173E+02						
ULN( 5) =	1.31818E+01						
ULN( 6) =	3.71673E+00						
ULN( 8) =	6.11812E+01						
ULN(10) =	1.72817E+03						
ULN(11) =	1.09021E+03						
ULN(12) =	8.33152E+01						
ULN(13) =	1.66614E+02						
ULN(14) =	5.00035E+02						
ULN(15) =	5.00035E+02						
ULN(16) =	5.00035E+02						
ULN(17) =	2.21315E+01						
ULN(18) =	2.55655E+01						
ULN(19) =	7.25015E+00						



NO CONVERGENCE	- NI=	14 NU=	25 EPS= 5.0E-02
NO CONVERGENCE	- NI=	14 NO=	30 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	31 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	31 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	32 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	32 EPS= 5.0E-02
NO CONVERGENCE	- NI=	14 NO=	34 EPS= 5.0E-02
NO CONVERGENCE	- NI=	14 NO=	35 EPS= 5.0E-02
NO CONVERGENCE	- NI=	14 NO=	36 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	37 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	37 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	38 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	39 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	40 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	41 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	42 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	43 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	44 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	45 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NO=	46 EPS= 5.0E-02
NO CONVERGENCE	- NI=	16 NO=	47 EPS= 5.0E-02
NO CONVERGENCE	- NI=	16 NO=	48 EPS= 5.0E-02
NO CONVERGENCE	- NI=	16 NO=	49 EPS= 5.0E-02
NO CONVERGENCE	- NI=	16 NO=	50 EPS= 9.0E-04
NO CONVERGENCE	- NI=	15 NC=	50 EPS= 5.0E-02
NO CONVERGENCE	- NI=	15 NG=	100 EPS= 9.0E-04
NO CONVERGENCE	- NI=	15 NO=	122 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NO=	172 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NO=	222 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NO=	272 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NC=	322 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NC=	372 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NO=	422 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NO=	472 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NO=	522 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NO=	572 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NO=	622 EPS= 4.0E-02
NO CONVERGENCE	- NI=	15 NC=	625 EPS= 4.0E-02

PS=0 (-----) HISTOGRAM (IN STEPS OF .05) FOR PS GREATER THAN 0 AND LESS THAN -----) PS=1  
 03 3 11 9 2 4 5 8 8 10 12 13 7 13 16 12 15 14 16 17 10 331 101

63	30	25	17	12	11	10	12	13	13	13	20	22	16	15	15	20	19	10	30	30	30	49	152
11	0	13	7	6	2	0	0	0	0	0	0	4	0	0	0	0	0	0	1	37	58	0	380
119	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	396
17	0	0	0	0	1	3	2	3	1	3	10	13	1	13	16	15	0	0	20	22	22	1	484
0	0	0	2	4	0	6	3	0	2	7	8	10	0	9	22	27	3	40	54	1	387	0	604
6	1	2	0	0	0	1	0	1	0	1	2	0	0	2	4	2	1	4	0	0	0	7	589
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	619
14	1	1	0	0	0	1	0	1	0	1	0	0	1	1	0	1	0	1	1	1	1	1	600
57	11	24	18	17	11	16	19	14	7	20	15	16	19	15	20	17	17	34	58	58	24	166	
305	6	6	5	5	6	5	5	2	4	4	3	3	4	3	7	5	3	1	5	5	6	227	2
0	0	0	0	0	0	1	0	0	1	0	1	1	1	2	2	2	6	8	22	28	548	0	0
4	3	6	5	7	1	1	4	1	4	4	2	3	3	3	5	2	7	12	13	16	526	3	3
0	0	0	0	0	1	2	0	3	11	9	15	14	13	24	33	44	33	77	105	87	149	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
0	0	2	2	5	5	11	6	3	13	5	11	11	8	11	19	32	26	38	61	217	138	10	10
0	7	4	1	2	3	6	4	5	4	14	5	5	13	13	15	10	24	42	37	222	178	11	11
0	1	0	1	1	2	0	3	0	4	1	8	3	7	10	8	10	22	24	56	250	204	12	12
7	39	14	16	15	15	10	9	11	14	13	7	14	13	15	13	14	24	24	41	147	142	8	8
4	183	31	22	26	23	20	22	24	24	16	22	18	16	19	14	21	25	20	26	43	8	0	0
5	222	43	32	29	25	18	20	20	24	15	16	11	16	13	16	17	17	21	14	23	4	0	0
7	39	14	16	15	15	10	5	11	14	13	7	18	13	15	13	14	24	28	41	147	142	8	8
0	8	4	5	11	15	14	13	5	17	22	15	26	19	24	22	39	41	57	69	157	33	7	7
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6
0	0	0	0	0	0	0	0	0	0	0	1	0	1	2	3	2	5	9	23	36	543	5	5
6	1	2	0	0	0	1	0	1	2	1	2	0	0	2	4	3	1	4	0	12	583	4	4
14	0	0	0	0	0	1	1	1	1	3	6	6	2	5	7	11	2	12	19	9	525	3	3
33	12	18	10	8	6	5	15	11	5	14	9	0	11	13	13	16	9	24	36	31	313	2	2
89	8	16	11	6	9	9	10	11	15	15	15	12	17	20	15	19	14	30	37	16	233	1	1

INDIVIDUAL COMPONENTS

NAME	1	10	30	50	70	90	59
134	0.000	0.000	0.001	.063	1.000	1.000	1.000
102	0.000	0.000	.277	.687	.964	1.000	1.000
132	0.000	.025	.404	.750	.963	1.000	1.000
5	.401	.621	.620	.914	.978	1.000	1.000
8	J.000	.108	.544	.921	.965	1.000	1.000
10	.172	.544	.872	.950	.984	1.000	1.000
11	.045	.565	.899	.970	.968	1.000	1.000
12	.321	.805	.955	.975	1.000	1.000	1.000
604	.252	.726	.924	1.000	1.000	1.000	1.000
601	J.000	.683	1.000	1.000	1.000	1.000	1.000

103	0.000	.478	.900	1.000	1.000	1.000	1.000	1.000	1.000
2	.631	.574	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3	.037	.269	1.000	1.000	1.000	1.000	1.000	1.000	1.000
131	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
602	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
706	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
9	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
705	.013	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
601	0.000	0.000	.628	1.000	1.000	1.000	1.000	1.000	1.000
101	0.000	0.000	.625	1.000	1.000	1.000	1.000	1.000	1.000

INDIVIDUAL SUBSYSTEMS

NAME	1	10	30	50	70	90	99
7	.039	.202	.659	.353	.951	.991	1.000
1	0.000	0.000	.462	.656	1.000	1.000	1.000
8	0.000	.108	.557	.921	.985	1.000	1.000
4	.013	1.000	1.000	1.000	1.000	1.000	1.000
5	.737	.973	1.000	1.000	1.000	1.000	1.000
6	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	0.000	.099	.717	1.000	1.000	1.000	1.000
3	0.000	.860	1.000	1.000	1.000	1.000	1.000

INDIVIDUAL SYSTEMS

NAME	1	10	30	50	70	90	99
-100	.000	.017	.041	.168	.449	.849	.995
-1	.001	.016	.051	.308	.614	.923	1.000
ULN( 0)=	6.00000E+02						
ULN( 1)=	1.61959E+01						
ULN( 2)=	3.79935E+02						
ULN( 4)=	5.70207E+02						
ULN( 5)=	1.39952E+01						
ULN( 6)=	4.19565E+00						
ULN( 8)=	6.70206E+01						
ULN(10)=	2.39946E+02						
ULN(11)=	1.39968E+03						
ULN(12)=	9.69330E+01						
ULN(13)=	1.95737E+02						
ULN(14)=	2.00042E+02						
ULN(15)=	6.00042E+02						
ULN(16)=	6.00042E+02						
ULN(17)=	2.70044E+01						
ULN(18)=	2.69514E+01						



NO CONVERGENCE - NI=	13	NO=	23	EPS=	5.0E-02
NO CONVERGENCE - NI=	13	NO=	29	EPS=	5.0E-02
NO CONVERGENCE - NI=	12	NO=	30	EPS=	5.0E-02
NO CONVERGENCE - NI=	12	NO=	31	EPS=	5.0E-02
NO CONVERGENCE - NI=	12	NO=	32	EPS=	5.0E-02
NO CONVERGENCE - NI=	11	NO=	33	EPS=	5.0E-02
NO CONVERGENCE - NI=	11	NO=	34	EPS=	5.0E-02
NO CONVERGENCE - NI=	11	NO=	35	EPS=	5.0E-02
NO CONVERGENCE - NI=	11	NO=	36	EPS=	5.0E-02
NO CONVERGENCE - NI=	11	NO=	37	EPS=	5.0E-02
NO CONVERGENCE - NI=	11	NO=	38	EPS=	5.0E-02
NO CONVERGENCE - NI=	11	NO=	39	EPS=	5.0E-02
NO CONVERGENCE - NI=	11	NO=	40	EPS=	5.0E-02
NO CONVERGENCE - NI=	10	NO=	41	EPS=	5.0E-02
NO CONVERGENCE - NI=	10	NO=	42	EPS=	5.0E-02
NO CONVERGENCE - NI=	10	NO=	43	EPS=	5.0E-02
NO CONVERGENCE - NI=	10	NO=	44	EPS=	5.0E-02
NO CONVERGENCE - NI=	10	NO=	45	EPS=	5.0E-02
NO CONVERGENCE - NI=	10	NO=	46	EPS=	5.0E-02
NO CONVERGENCE - NI=	10	NO=	47	EPS=	5.0E-02
NO CONVERGENCE - NI=	9	NO=	48	EPS=	5.0E-02
NO CONVERGENCE - NI=	9	NO=	49	EPS=	5.0E-02
NO CONVERGENCE - NI=	9	NO=	50	EPS=	5.0E-02
NO CONVERGENCE - NI=	9	NO=	53		

PS=0	----	HISTOGRAM	(IN STEPS IF	PS) FOR PS	GREATER THAN	AND LESS	THAN	-----	PS=1		
14	2	C	0	1	1	1	1	1	2	1	21
9	3	0	1	2	1	0	2	1	1	1	9
1	0	0	0	2	0	0	0	0	1	3	0
19	0	1	0	0	1	1	2	1	0	0	29
0	C	C	0	0	0	0	0	0	0	0	22
0	0	1	0	0	1	0	0	1	3	2	53
0	0	1	0	0	1	0	0	1	1	1	39
4	0	0	1	0	1	0	0	0	0	1	27
0	0	0	0	0	0	0	0	0	0	0	45
0	C	C	0	0	0	0	0	0	0	0	53
2	2	4	0	0	2	0	7	1	0	1	52
35	C	C	0	0	0	1	C	1	C	0	7
1	0	0	2	0	0	1	0	0	0	0	14
0	0	1	1	0	0	0	0	0	1	3	32
0	0	0	0	0	2	1	2	0	0	1	47
0	0	0	0	0	2	1	2	0	1	6	5

9  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60  
 61  
 62  
 63  
 64  
 65  
 66  
 67  
 68  
 69  
 70  
 71  
 72  
 73  
 74  
 75  
 76  
 77  
 78  
 79  
 80  
 81  
 82  
 83  
 84  
 85  
 86  
 87  
 88  
 89  
 90  
 91  
 92  
 93  
 94  
 95  
 96  
 97  
 98  
 99  
 100

INDIVIDUAL COMPONENTS

NAME	1	10	30	50	70	90	99
133	0.000	0.000	0.000	0.000	.377	1.000	1.000
8	0.000	.002	.001	.325	.802	.984	1.000
102	0.000	0.000	.245	.625	.876	1.000	1.000
132	0.000	.066	.378	.705	.851	1.000	1.000
10	0.000	.162	.445	.712	.921	.982	1.000
11	0.000	.255	.564	.758	.954	.992	1.000
101	0.000	0.000	.048	.613	1.000	1.000	1.000
5	.263	.407	.697	.834	.925	.997	1.000
601	0.000	0.000	0.000	.875	1.000	1.000	1.000
12	.113	.365	.699	.820	.975	1.000	1.000
604	.076	.688	.994	1.000	1.000	1.000	1.000
2	0.000	.755	.957	1.000	1.000	1.000	1.000
3	.076	.555	1.000	1.000	1.000	1.000	1.000
603	.076	.722	1.000	1.000	1.000	1.000	1.000
103	0.000	.615	.886	1.000	1.000	1.000	1.000
131	.376	1.000	1.000	1.000	1.000	1.000	1.000
602	1.000	1.000	1.000	1.000	1.000	1.000	1.000
706	1.000	1.000	1.000	1.000	1.000	1.000	1.000
9	1.000	1.000	1.000	1.000	1.000	1.000	1.000
705	0.000	.215	1.000	1.000	1.000	1.000	1.000

INDIVIDUAL SUBSYSTEMS

NAME	1	10	30	50	70	90	99
E	0.000	.002	.004	.325	.802	.984	1.000

7	.003	.629	.151	.375	.651	.892	.691
1	0.000	0.000	.023	.744	.741	1.000	1.000
2	0.000	.672	.597	.755	1.000	1.000	1.000
4	0.000	.215	1.000	1.000	1.000	1.000	1.000
5	.676	.883	.979	1.000	1.000	1.000	1.000
6	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3	.076	.755	1.000	1.000	1.000	1.000	1.000

INDIVIDUAL SYSTEMS

NAME	1	10	30	50	70	90	99
-100	0.000	.002	.019	.034	.051	.185	.674
-1	0.000	.007	.025	.042	.102	.425	.973
ULN( 0) =	7.00000E+02						
ULN( 1) =	1.79624E+01						
ULN( 2) =	4.43257E+02						
ULN( 4) =	6.65242E+02						
ULN( 5) =	1.47258E+01						
ULN( 6) =	4.65558E+00						
ULN( 8) =	7.23905E+01						
ULN(10) =	3.16678E+03						
ULN(11) =	1.72381E+03						
ULN(12) =	1.16647E+02						
ULN(13) =	2.33259E+02						
ULN(14) =	7.00049E+02						
ULN(15) =	7.00049E+02						
ULN(16) =	7.00049E+02						
ULN(17) =	3.19164E+01						
ULN(18) =	3.24083E+01						
ULN(19) =	1.48801E+01						

F = 4.9056E-01 P = 7.0000E+02

M = 5.0000E+00	H = 0.	10 NO =	C LPS = 5.0E-02
NO CONVERGENCE	- NI =	20 NO =	0 FPS = 5.0E-02
NO CONVERGENCE	- NI =	40 NO =	0 FPS = 5.0E-02
NO CONVERGENCE	- NI =	53 NO =	0 FPS = 5.0E-02
NO CONVERGENCE	- NI =	49 AU =	0 FPS = 5.0E-02
NO CONVERGENCE	- NI =	26 AU =	1 FPS = 5.0E-02
NO CONVERGENCE	- NI =	20 AU =	2 FPS = 5.0E-02
NO CONVERGENCE	- NI =	18 AU =	3 FPS = 5.0E-02
NO CONVERGENCE	- NI =	16 AU =	4 FPS = 5.0E-02
NO CONVERGENCE	- NI =	15 AU =	5 FPS = 5.0E-02
NO CONVERGENCE	- NI =	14 AU =	6 FPS = 5.0E-02

NO CONVERGENCE	-	NI=	13	NO=	7	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	12	NO=	8	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	12	NO=	9	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	12	NO=	10	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	12	NO=	11	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	11	NO=	12	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	11	NO=	13	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	10	NO=	14	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	10	NO=	15	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	10	NO=	16	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	10	NO=	17	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	9	NO=	18	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	9	NO=	19	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	9	NO=	20	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	9	NO=	21	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	8	NO=	22	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	8	NO=	23	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	8	NO=	24	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	8	NO=	25	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	8	NO=	26	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	8	NO=	27	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	8	NO=	28	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	7	NO=	29	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	7	NO=	30	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	7	NO=	31	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	7	NO=	32	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	7	NO=	33	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	7	NO=	34	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	7	NO=	35	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	7	NO=	36	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	7	NO=	37	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	7	NO=	38	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	6	NO=	39	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	6	NO=	40	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	6	NO=	41	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	6	NO=	42	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	6	NO=	43	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	6	NO=	44	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	6	NO=	45	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	6	NO=	46	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	6	NO=	47	FPS=	5.0E-02
NO CONVERGENCE	-	NI=	6	NO=	48	FPS=	5.0E-02

NO CONVERGENCE - NI= 6 NJ= 49 EPS= 5.0E-02  
 CONVERGENCE - NI= 6 NO= 50

PS=0 (----- HISTOGRAM (IN STEPS OF .CS) FOR PS GREATER THAN 0 AND LESS THAN -----) PS=1

31	1	0	1	0	0	1	4	C	0	2	1	0	1	1	0	0	0	2	0	0	3	2	0	3	2	101
20	4	2	0	1	4	0	1	1	1	1	1	0	1	2	1	0	2	0	0	2	0	2	0	0	6	102
8	0	0	1	0	0	1	0	0	1	0	0	2	0	3	4	3	0	0	0	0	0	0	0	0	24	103
29	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	601
6	0	0	0	0	0	0	0	1	1	0	1	3	0	1	1	0	1	3	2	0	0	0	0	0	50	602
6	0	0	0	0	0	0	0	1	2	0	0	4	1	2	4	4	4	4	0	0	2	0	0	0	30	603
8	1	1	0	0	1	0	0	2	0	0	0	1	1	0	1	0	0	1	0	0	2	31	0	28	604	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	49	0	1	705	
2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	706
6	1	1	4	0	1	1	0	2	2	1	1	0	0	1	0	2	3	4	1	1	16	0	0	0	131	
4	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	132	
4	3	2	1	0	1	0	1	0	0	1	2	0	0	3	2	0	1	1	1	1	3	5	19	0	133	
2	1	0	2	1	1	1	1	1	1	1	0	2	1	0	0	2	1	1	1	1	0	32	0	0	3	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	
2	3	7	2	3	5	1	1	3	3	0	1	4	2	0	3	2	0	3	1	2	4	1	4	0	10	
1	0	1	1	2	1	1	1	0	0	2	1	0	3	3	3	3	3	3	3	3	3	3	3	0	11	
9	10	3	3	1	1	1	1	3	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	12	
17	28	1	1	1	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	
11	36	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	
9	10	3	3	1	1	1	1	3	0	1	1	4	1	1	1	1	1	1	1	1	1	1	1	1	-100	
1	12	6	6	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	
0	0	0	0	0	0	0	0	0	0	1	2	1	1	2	0	1	2	4	4	4	4	4	4	4	6	
8	1	1	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	5	
3	0	0	2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	4	
8	2	0	0	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	
33	3	2	0	0	1	2	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	

INDIVIDUAL COMPONENTS

NAME	1	10	30	50	70	90	5'
601	0.000	0.000	0.000	0.000	.750	1.000	1.000
133	0.000	0.000	0.000	0.000	0.000	1.000	1.000
101	0.000	0.000	0.000	0.000	.212	.900	1.000
102	0.000	0.000	0.000	.075	.500	1.000	1.000
8	0.000	0.000	.030	.150	.500	.850	1.000

10	0.000	.050	.162	.250	.550	.867	.992
11	.013	.100	.217	.600	.850	.973	1.000
5	.008	.175	.500	.700	.840	1.000	1.000
132	0.000	0.000	.350	.800	1.000	1.000	1.000
103	0.000	0.000	.683	.850	1.000	1.000	1.000
12	0.000	.200	.600	.850	.957	.993	1.000
2	0.000	.017	.550	.933	1.000	1.000	1.000
3	0.000	.150	.825	1.000	1.000	1.000	1.000
604	.375	.600	.663	1.000	1.000	1.000	1.000
603	0.000	.125	.850	1.000	1.000	1.000	1.000
131	0.000	1.000	1.000	1.000	1.000	1.000	1.000
602	1.000	1.000	1.000	1.000	1.000	1.000	1.000
706	.975	1.000	1.000	1.000	1.000	1.000	1.000
9	.912	.588	1.000	1.000	1.000	1.000	1.000
705	0.000	0.000	.700	1.000	1.000	1.000	1.000

INDIVIDUAL SUBSYSTEMS

NAME	1	10	30	50	70	90	99
1	0.000	0.000	0.000	0.000	.033	.550	.992
8	0.000	0.000	.030	.150	.500	.850	1.000
7	0.000	.017	.067	.150	.275	.475	.537
2	0.000	0.000	.400	.850	1.000	1.000	1.000
5	.162	.350	.950	.942	1.000	1.000	1.000
4	0.000	0.000	.700	1.000	1.000	1.000	1.000
6	.912	.588	1.000	1.000	1.000	1.000	1.000
3	0.000	.150	.883	1.000	1.000	1.000	1.000

INDIVIDUAL SYSTEMS

NAME	1	10	30	50	70	90	99
-1	0.000	0.000	0.000	.014	.032	.050	.275
-100	0.000	0.000	.006	.019	.033	.047	.125
ULN( 0) =	8.00000E+02						
ULN( 1) =	1.96435E+01						
ULN( 2) =	5.06580E+02						
ULN( 4) =	7.60276E+02						
ULN( 5) =	1.53933E+01						
ULN( 6) =	5.09229E+00						
ULN( 8) =	7.73867E+01						
ULN(10) =	4.02720E+03						
ULN(11) =	2.07585E+C3						
ULN(12) =	1.33311E+02						
ULN(13) =	2.66582E+02						







INDIVIDUAL SYSTEMS

NAME	1	10	30	50	70	90	99
-100	0.000	0.000	0.000	0.000	0.015	0.040	.175
-1	0.000	0.000	0.000	0.000	0.000	0.038	.175
ULN( 0)=	1.00000E+02						
ULN( 1)=	2.28112E+01						
ULN( 2)=	6.33225E+02						
ULN( 4)=	9.50346E+02						
ULN( 5)=	1.65696E+01						
ULN( 6)=	5.91359E+00						
ULN( 8)=	8.65232E+01						
ULN(10)=	6.01785E+03						
ULN(11)=	2.81814E+03						
ULN(12)=	1.66638E+02						
ULN(13)=	3.33228E+02						
ULN(14)=	1.00007E+03						
ULN(15)=	1.00007E+03						
ULN(16)=	4.68174E+01						
ULN(17)=	4.18393E+01						
ULN(18)=	2.43612E+01						

W=	H=	R=	P=
5.0000E+00	0.	4.3210E-01	1.0000E+03
NO CONVERGENCE	- NI=	5 NO=	0 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	1 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	2 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	3 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	4 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	5 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	6 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	7 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	8 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	9 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	10 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	11 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	12 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	13 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	14 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	15 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	16 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	17 EPS= 5.0E-02
NO CONVERGENCE	- NI=	5 NO=	18 EPS= 5.0E-02





131	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
602	.675	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
706	.575	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

INDIVIDUAL SUBSYSTEMS

NAME	1	10	30	50	70	90	99
8	0.000	0.000	0.000	0.000	0.000	.050	.475
1	0.000	0.000	0.000	0.000	0.000	0.000	.175
4	0.000	0.000	0.000	.013	.875	1.000	1.000
7	0.000	0.000	.003	.021	.033	.046	.175
5	0.000	0.000	.300	.475	.756	.850	1.000
6	0.000	.075	.240	.600	.750	.842	.575
2	0.000	0.000	.150	.775	1.000	1.000	1.000
3	0.000	0.000	.575	1.000	1.000	1.000	1.000

INDIVIDUAL SYSTEMS

NAME	1	10	30	50	70	90	99
-1	0.000	0.000	0.000	0.000	0.000	0.000	.037
-100	0.000	0.000	0.000	0.000	0.000	0.025	.047
ULN( 0)=	1.20000E+03						
ULN( 1)=	2.57751E+01						
ULN( 2)=	7.59870E+02						
ULN( 4)=	1.14041E+03						
ULN( 5)=	1.75972E+01						
ULN( 6)=	6.68194E+00						
ULN( 8)=	9.47814E+01						
ULN(10)=	8.35541E+03						
ULN(11)=	3.61776E+03						
ULN(12)=	1.99966E+02						
ULN(13)=	3.99873E+02						
ULN(14)=	1.20008E+02						
ULN(15)=	1.20008E+03						
ULN(16)=	1.20008E+03						
ULN(17)=	5.68486E+01						
ULN(18)=	4.76204E+01						
ULN(19)=	3.12739E+01						

W= 5.0000E+00 H= 0.  
 NO CONVERGENCE - NI=  
 NO CONVERGENCE - NI=  
 NO CONVERGENCE - NI=  
 NO CONVERGENCE - NI=  
 U= 4.0502E-01 P= 1.2000E+03  
 5 NI= 0 EPS= 5.0E-02  
 5 NI= 1 EPS= 5.0E-02  
 5 NI= 2 EPS= 5.0E-02  
 5 NI= 3 EPS= 5.0E-02







-100)



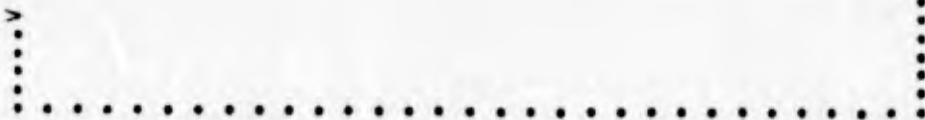
A

V

V

.....A.....

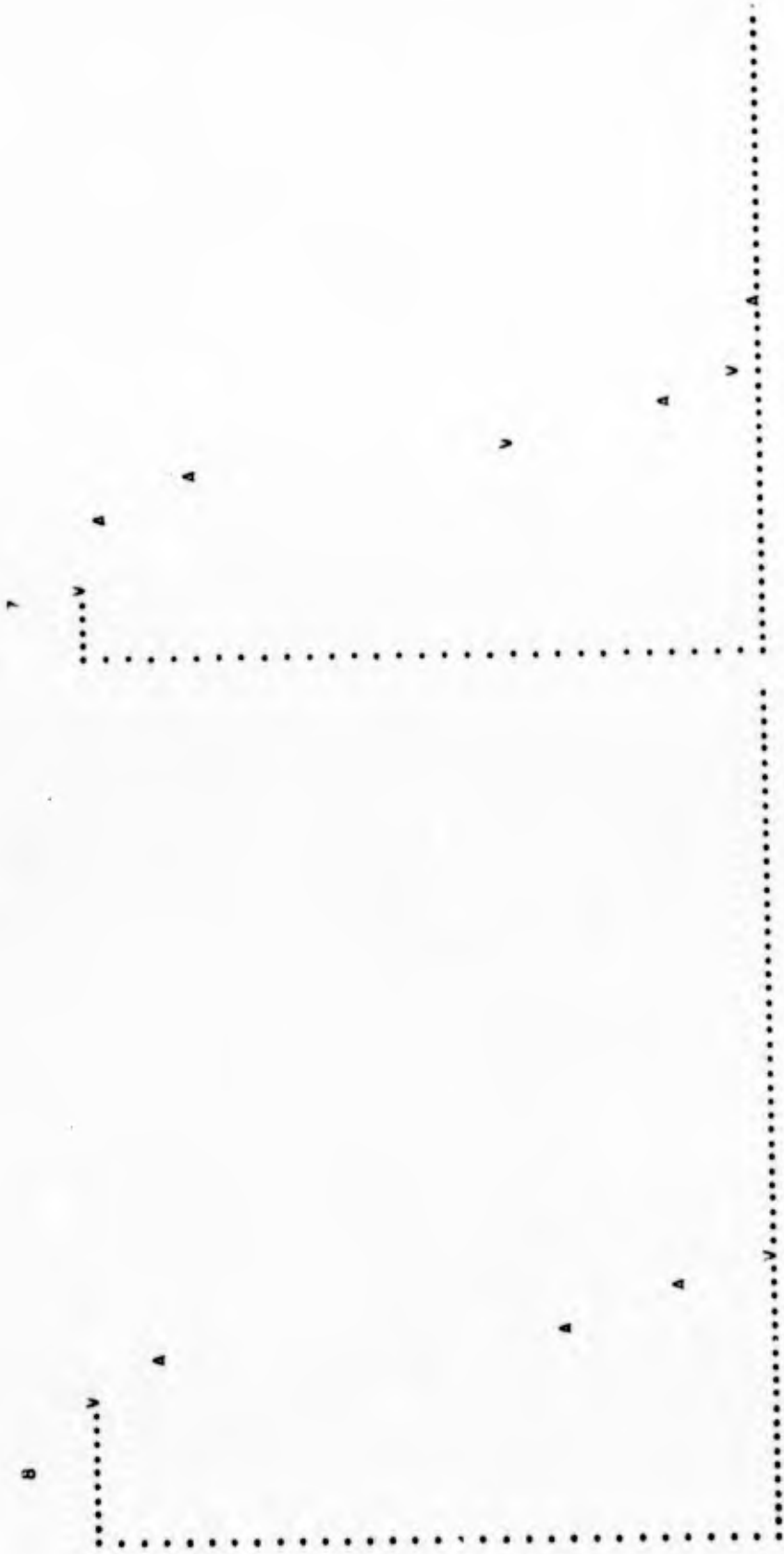
-1

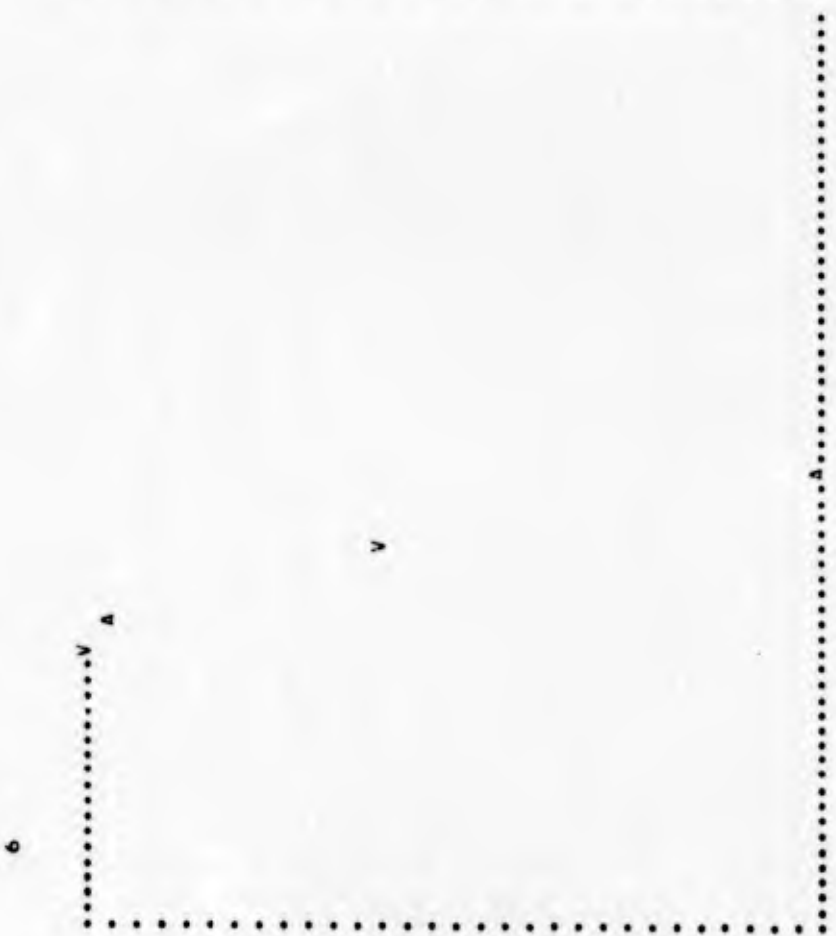


V

V

A





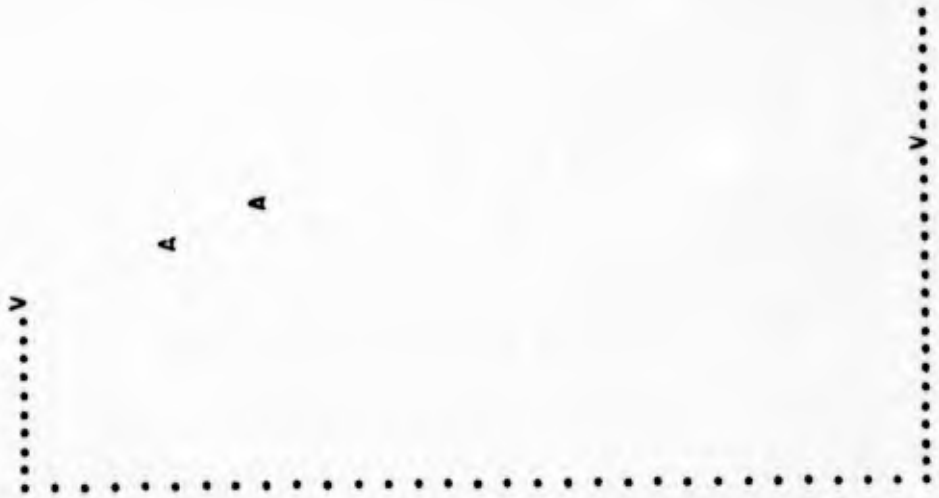
3



4



1



2



10.37.43.JCB,K7042CB.  
10.37.43.ACCOUNT,JC54066.  
10.37.43.NAME,641100,54680,CULIVER,J.  
10.37.43.\$PROBLEM,235217.  
10.37.43.\$PAXTIM,2000.  
10.37.43.\$FLENGTH,70000.  
10.37.43.FUNCS,,,,110000)  
10.37.45.LINK,I=INPUT,I=LGO,LJ=JN,X.  
10.37.47. MIN FL - 36000 LOAD. 65600 EXECUTE.  
10.43.27.STOP  
10.43.27.ATIM 122.052 SEC.  
10.43.27.CPU 119.761 SEC.  
10.43.27.PFU 3.115 SEC.  
10.43.27.MUCM 3.190 MWD-SFC.  
10.43.27.DISK 2.608 KPHUS.  
10.43.27.PRNT 2100 EST LINES.

\$ DNA SAMPLE PROBLEM - CASE OF EXPERIMENTAL DESIGN

PLOTON=40CS  
 \$ ELECTRONIC RACKS  
 C101=/0,30,35,40/1,40,45,50/R1=13,.2,1.2\$  
 C102=/0,15,17,19/1,19,21,23/R1=17/USER(1)/R1=K1,1.0,1.2\$  
 C103=/0,50/1,50/R1=18,1.0,1.2\$  
 C1011 = /0,32.4,37.4,43.2/1,43.2,48.6,54/R1=13,.2,1.2\$  
 \$ BATTERY MCUNTS  
 C601=/0,50C,550,600/1,600,650,700/R1=4,1.0,1.2\$  
 C602=/0,175,200,225/1,225,250,275/R1=8/USER(2)/R1=K1,1.0,1.2\$  
 C603=/0,20/1,20/R1=1,1.1/USER(4)\$  
 C604=/0,10/1,10/R1=6,1,1\$  
 \$ BATTERY ELEMENTS  
 C705=/0,45C,500,550/1,600,650,700/R1=2,1,1.2\$  
 C706=/0,15C,175,200/1,225,250,275/R1=8,1,1.2\$  
 \$ SIS  
 C131=/0,30,35,40/1,35,40,45/R1=1,1,1.2\$  
 C132=/0,30,35,40/1,35,40,45/R1=5,2,1.3\$  
 C133=/0,30,35,40/1,35,40,45/R1=4,.C8,1.2\$  
 C1311 = /0,34.5,40,25,46/1,40,25,46,51.75/R1=1,1,1.2\$  
 C1321 = /0,34.5,40,25,46/1,40,25,46,51.75/R1=5,2,1.3\$  
 C1331 = /0,34.5,40,25,46/1,40,25,46,51.75/R1=4,.C8,1.2\$  
 \$ COMMUNICATIONS  
 C2 = /0,.7237,.9408,1.223/.0668,.7416,.5641,1.253/.3085,.7559,.9879,1.284/  
 .6915,.7787,1.012,1.316/.9332,.7979,1.037,1.348/1,.6176,1.003,1.282/  
 R1=11,.000409,1.3/USER(3)\$  
 C3 = /0,.4704,.9408,1.882/.0666,.4820,.5641,1.928/.3085,.4939,.9979,1.976/  
 .6915,.5061,1.012,2.024/.9332,.5286,1.037,2.074/1,.5314,1.003,2.126/  
 R1=19,.04,1.4\$  
 C5 = /0,.6125,.7962,1.075/.0668,.6709,.8722,1.134/.3085,.7350,.9554,1.242/  
 .6915,.3051,1.047,1.361/.9332,.8815,1.146,1.490/1,.9661,1.256,1.633/  
 R1=10,.0002,1.3\$  
 \$ BLAST VALVE  
 C9 = /0,.6635,.7962,.9554/.0668,.7268,.8722,1.047/.3085,.7462,.9554,1.146/  
 .6915,.8722,1.047,1.256/.9332,.9554,1.146,1.376/1,1.047,1.256,1.577/  
 R1=0,.001,1.2\$  
 \$ FACILITY  
 C10 = /0,.6635,.7962,.9554/.0668,.7268,.8722,1.047/.3085,.7462,.9554,1.146/  
 .6915,.8722,1.047,1.256/.9332,.9554,1.146,1.376/1,1.047,1.256,1.577/  
 R1=14,.0015,1.1\$  
 C11 = /0,.6033,.7204,.8645/.0668,.6845,.8214,.9855/.3085,.6804,.9300,1.124/

.6915,.8855,1.066,1.281/.9122,1.014,1.217,1.441/1,1.157,1.383,1.600/  
 R1=15,.00135,1.25  
 C12 = /0,.5472,.6566,.7830/.0668,.6475,.7770,.9324/.9055,.7061,.5153,1.103/  
 .6915,.9065,1.088,1.305/.0232,1.072,1.287,1.544/1,1.289,1.523,1.827/  
 R1=16,.0012,1.38  
 C1000 = /0,.6635,.7462,.8554/.0668,.7206,.8722,1.047/.3085,.7962,.9554,1.140/  
 .6915,.8722,1.047,1.256/.9332,.5554,1.146,1.376/1,1.047,1.256,1.577/  
 R1=14,.0014,1.15  
 C1100 = /0,.6003,.7204,.8645/.0668,.6845,.8214,.9856/.3085,.7604,.9365,1.124/  
 .6915,.8856,1.068,1.281/.9332,1.014,1.217,1.461/1,1.157,1.388,1.660/  
 R1=15,.00125,1.25  
 C1200 = /0,.5472,.6566,.7880/.0668,.6475,.7770,.9324/.3085,.7661,.9193,1.103/  
 .6915,.9065,1.088,1.305/.9332,1.072,1.287,1.544/1,1.269,1.523,1.827/  
 R1=16,.0011,1.35  
 SS1=101+102+103\$  
 SS2 = (131+132)\*133\$  
 SS3=601+603+602+604\$  
 SS4=705+706\$  
 SS5=(2+3)\*5\$  
 SS6=9\$  
 SS7=10+11+12\$  
 SS11 = 1011+102\*103\$  
 SS21 = (1311+1321)\*1331\$  
 SS71 = 100C+1109+1200\$  
 SYS1=SS1+SS2+SS3+SS4+SS5+SS6+SS7\$  
 SYS2 = SS11+SS2+SS3+SS4+SS5+SS6+SS7\$  
 SYS3 = SS1+SS21+SS3+SS4+SS5+SS6+SS7\$  
 SYS4 = SS11+SS21+SS3+SS4+SS5+SS6+SS7\$  
 SYS5 = SS1+SS2+SS3+SS4+SS5+SS6+SS71\$  
 SYS6 = SS11+SS2+SS3+SS4+SS5+SS6+SS71\$  
 SYS7 = SS1+SS21+SS3+SS4+SS5+SS6+SS71\$  
 SYS8=SS11+SS21+SS2+SS4+SS5+SS6+SS71\$  
 MOJE=16\*4,3\*1  
 ULN=2.785,5.94,4.094,6.340,2.029,1.435,3.807,4.205,0.642,7.782,7.244,  
 4.605,5.258,6.347,6.397,3.296,3.367,2.485  
 ALPHA=.17,.083,.083,.063,.25,.17,.083,.083,0,.063,0,.06,.04,.04,.03,  
 .17,1.23,.67,1.53  
 BETA =.67,1.01,1.01,.35,.67,.5,.5,.33,1.6,1.37,5+1.0,-3,-2,-3.56  
 S 1 = .1927, .1767, .0894, .1767, .2031, .2927, .0111, .0111,  
 -.1705, .3448, .0116, .0085, .0045, .0015, .0015, .0224,  
 0.0000, -.0873, 0.0000,  
 S 2 = .1765, .0892, .1765, .1713, .1767, .0110, .0110, .0110, -.1546,

S 3 =	.3445,	.0114,	.1033,	.0082,	.0013,	.0013,	.0223,	0.0000,
	-.0872,	0.0000,	.0840,	.0554,	.0072,	.0072,	-.0749,	.1737,
	.0806,	.0842,	.0083,	.0013,	.0013,	.0013,	0.0000,	-.0169,
	.0114,	.0082,						
S 4 =	0.0000,							
	.1765,	.1713,	.1767,	.0110,	.0110,	-.1546,	.3445,	.0114,
	.0083,	.0083,	.0013,	.0013,	.0223,	0.0000,	-.0873,	0.0000,
	.2328,	.2031,	.0084,	.0084,	-.1862,	.3351,	.0042,	.0031,
	.0031,	-.0039,	-.0039,	.0170,	0.0000,	-.0873,	0.0000,	
S 6 =	.2099,	.0111,	.0111,	-.1477,	.3448,	.0116,	.0065,	.0085,
	.0015,	.0015,	.0224,	0.0000,	-.0973,	0.0000,		
S 7 =	.0043,	.0043,	-.0024,	.0190,	.0057,	.00+2,	.0042,	.0042,
	.0042,	.0042,	0.0000,	-.0038,	0.0000,			
S 8 =	.0043,	-.0024,	.0190,	.0057,	.0042,	.0042,	.0042,	.0042,
	.0042,	0.0000,	-.0039,	0.0000,				
S 9 =	.1828,	-.3068,	-.0002,	-.0002,	-.0002,	.0068,	.0068,	-.0141,
	.0000,	.0797,	0.0000,					
S 10 =	.6745,	.0205,	.0150,	.0150,	.0010,	.0010,	.0423,	0.0000,
	-.1708,	0.0000,						
S 11 =	.0162,	.0114,	.0114,	.0114,	.0114,	.0114,	0.0000,	0.0000,
	0.0000,							
S 12 =	.0083,	.0083,	.0083,	.0083,	.0083,	0.0000,	0.0000,	0.0000,
S 13 =	.0083,	.0083,	.0083,	.0083,	.0083,	0.0000,	0.0000,	0.0000,
S 14 =	.0118,	.0113,	.0075,	0.0000,	0.0000,	0.0000,		
S 15 =	.0118,	.0077,	0.0000,	0.0000,	0.0000,			
S 16 =	.0154,	0.0000,	0.0000,	0.0000,				
S 17 =	.0230,	.0107,	.0180,					
S 18 =	.0824,	.0091,						
S 19 =	.0393,							
K-FAC =	2.52, 1.86,	1.98,	1.96, 4.95,	2.52,	2.03, 2.12,	5.75,	3.30, 1.61,	
	1.56, 1.56,	1.45,	1.62, 1.45,	3.02,	2.86, 2.81,			
A 1 =	1.0000,	.6123,	.2679,	.5640,	.9037,	.8684,	.0567,	.0532,
	-.4858,	.6125,	.1050,	.0921,	.0821,	.0427,	.0330,	.1549,
	0.0000,	-.1885,	0.0000,					
A 2 =	1.0000,	.4776,	.9213,	.3005,	.5317,	.0341,	.0786,	-.2570,
	.5144,	.1539,	.1293,	.1203,	.0611,	.0471,	.2295,	0.0000,
	-.2919,	0.0000,						
A 3 =	1.0000,	.4400,	.1053,	.2427,	.0526,	.0493,	-.1000,	.3962,
	.1404,	.1097,	.1057,	.0557,	.0420,	.2093,	0.0000,	-.0274,
	0.0000,							
A 4 =	1.0000,	.2709,	.4308,	.0775,	.0726,	-.2373,	.8424,	.1410,
	.1108,	.1108,	.0563,	.0434,	.2114,	0.0000,	-.2537,	0.0000,

A 5 =	1.0000,	.7691,	.0234,	.0219,	-.4642,	.3355,	.0221,	.0173,
A 6 =	1.0000,	-.0113,	-.0687,	.0630,	0.0000,	-.1055,	0.0000,	
A 7 =	1.0000,	.0453,	.0462,	-.5724,	.5215,	.0311,	.0713,	.0713,
A 8 =	1.0000,	.0286,	.1245,	0.0000,	-.1627,	0.0000,		
A 9 =	1.0000,	.9373,	.7770,	.0727,	.0678,	.0530,	.0530,	.0629,
A 10 =	1.0000,	.0471,	0.0000,	-.0156,	0.0000,	.0457,	.0590,	.0455,
A 11 =	1.0000,	.7283,	.0682,	.0635,	.0457,			
A 12 =	1.0000,	.0000,	-.0147,	0.0000,	-.0009,	.0281,	.0216,	-.0443,
A 13 =	1.0000,	-.2669,	-.0011,	-.0009,	-.0009,			
A 14 =	1.0000,	.0376,	0.0000,	.1126,	.0486,	.0375,	.2276,	0.0000,
A 15 =	1.0000,	.1440,	.1126,	.2555,	.1970,	.1912,	0.0000,	0.0000,
A 16 =	1.0000,	0.0000,	.2151,	.1540,	.1455,	0.0000,	0.0000,	0.0000,
A 17 =	1.0000,	.2151,	.1957,	.1455,	0.0000,	0.0000,	0.0000,	0.0000,
A 18 =	1.0000,	.3346,	.1540,	.1455,	0.0000,	0.0000,	0.0000,	0.0000,
A 19 =	1.0000,	.1957,	.1540,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
A 20 =	1.0000,	.3557,	.4210,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
A 21 =	1.0000,	.2069,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
A 22 =	1.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
A 23 =	1.0000,	.4572,	.4179,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
A 24 =	1.0000,	.2223,						

NOMPWH=60C,5\$  
M=5

OP=200,400,500,600,700,800,1000,1200  
\$ THE NEXT CARD REPLACES THE PRECEDING CARD TO REMOVE 200 PSI  
OP= 40C,500,600,700,800,1000,1200  
RUNS

\$DNA SAMPLE PROBLEM - CASE OF STEEPEST ASCENT

PLOTGN=400\$

\$ ELECTRONIC KACKS

C101=/0,30,35,40/1,40,45,50/K1=13,,2,1.2\$  
 C102=/0,15,17,19,21,23/K1=17/USER(1)/R1=K1,1.0,1.2\$  
 C103=/0,50/1,50/K1=18,1.0,1.2\$  
 C1012 = /0,31,20,36,40,41,60/1,41,60,46,80,52,00/P1,1,3,,2,1.2\$  
 C1013 = /0,37,50,43,75,50,00/1,50,00,56,25,62,50/P1,1,3,,2,1.2\$  
 C1014 = /C,45,00,52,50,60,00/1,60,00,67,50,75,00/P1,1,3,,2,1.2\$  
 C1015 = /0,52,50,61,25,70,00/1,70,00,78,75,87,50/P1,1,3,,2,1.2\$  
 C1016 = /C,60,00,70,00,80,00/1,80,00,90,00,100,00/P1,1,3,,2,1.2\$

\$ BATTERY PLUNTS

C601=/0,50C,550,600/1,600,650,700/P1=9,1.0,1.2\$  
 C602=/0,175,200,225/1,225,250,275/K1=3/USER(2)/K1=K1,1.0,1.2\$

C603=/0,20/1,20/R1=1,1,1,1/USER(4)\$

C604=/0,1C/1,10/K1=6,1,1\$

\$ BATTERY ELEMENTS

C705=/0,45C,500,550/1,600,650,700/R1=2,1,1.2\$  
 C706=/0,15C,175,200/1,225,250,275/R1=8,1,1.2\$

\$ SIS

C131=/0,30,35,40/1,35,40,45/P1=1,1,1.2\$  
 C132=/0,30,35,40/1,35,40,45/R1=5,2,1.3\$  
 C133=/0,30,35,40/1,35,40,45/K1=4,,C8,1.2\$  
 C1312 = /0,32,40,37,80,43,20/1,37,30,42,20,48,60/P1=1,1,1.2\$  
 C1322 = /C,32,40,37,80,43,20/1,37,30,43,20,48,60/P1=5,2,1.3\$  
 C1332 = /C,32,40,37,80,43,20/1,37,30,43,20,48,60/P1=4,,08,1.2\$  
 C1313 = /0,46,50,54,25,62,00/1,54,25,62,00,69,75/P1=1,1,1.2\$  
 C1323 = /C,46,50,54,25,62,00/1,54,25,62,00,69,75/P1=5,2,1.3\$  
 C1333 = /C,46,50,54,25,62,00/1,54,25,62,00,69,75/P1=4,,08,1.2\$  
 C1314 = /C,63,00,73,50,84,00/1,73,50,84,00,94,50/P1=1,1,1.2\$  
 C1324 = /C,63,00,73,50,84,00/1,73,50,84,00,94,50/P1=5,2,1.3\$  
 C1334 = /C,63,00,73,50,84,00/1,73,50,84,00,94,50/P1=4,,08,1.2\$  
 C1315 = /C,79,50,92,75,106,00/1,92,75,106,00,119,25/P1=1,1,1.2\$  
 C1325 = /0,79,50,92,75,106,00/1,92,75,106,00,119,25/P1=5,2,1.3\$  
 C1335 = /0,79,50,92,75,106,00/1,92,75,106,00,119,25/P1=4,,08,1.2\$  
 C1316 = /C,50,00,112,00,128,00/1,112,00,128,00,144,00/P1=1,1,1.2\$  
 C1326 = /0,50,00,112,00,128,00/1,112,00,128,00,144,00/P1=5,2,1.3\$  
 C1336 = /C,50,00,112,00,128,00/1,112,00,128,00,144,00/P1=4,,08,1.2\$

\$ COMMOUNICATIONS

C2 = /0,,7237,,5400,1.223/0,608,,7414,,5641,1.253/,,3089,,7539,,0879,1.284/  
 .6915,,7767,1.012,1.316/,,33,,7776,1.037,1.333/1,,4176,1.065,1.362/

RI=11, .00040, 1.3/USE (4) 1  
 C3 = /0, .4704, .5463, 1.832/.0000, .820, .944, 1.1528/.3085, .4084, .5071, 1.516/  
 .6915, .5081, 1.012, 2.024/.0232, .6236, 1.037, 2.074/1.0114, 1.014, 1.014, 2.122/  
 RI=13, .04, 1.44  
 C5 = /0, .6125, .7962, 1.035/.0000, .9704, .8722, 1.134/.3085, .735, .9554, 1.242/  
 .6915, .8051, 1.047, 1.361/.9332, .3315, 1.146, 1.490/1.0114, .5081, 1.256, 1.033/  
 RI=10, .0002, 1.34  
 \$ BLAST VALVE  
 C9 = /0, .6635, .7962, .9554/.0668, .7268, .8722, 1.047/.3085, .7962, .9554, 1.146/  
 .6915, .8722, 1.047, 1.256/.9332, .9554, 1.146, 1.376/1.1047, 1.256, 1.577/  
 RI=14, .0015, 1.14  
 \$ FACILITY  
 C10 = /0, .6635, .7962, .9554/.0668, .7268, .8722, 1.047/.3085, .7962, .9554, 1.146/  
 .6915, .8722, 1.047, 1.256/.9332, .9554, 1.146, 1.376/1.1047, 1.256, 1.577/  
 RI=15, .0015, 1.14  
 C11 = /0, .6635, .7962, .9554/.0668, .7268, .8722, 1.047/.3085, .7962, .9554, 1.146/  
 .6915, .8722, 1.047, 1.256/.9332, .9554, 1.146, 1.376/1.1047, 1.256, 1.577/  
 RI=15, .0015, 1.14  
 C12 = /0, .5472, .6566, .7890/.0468, .6475, .7770, .9324/.3085, .7661, .9143, 1.103/  
 .6915, .9065, 1.088, 1.305/.9332, 1.072, 1.287, 1.544/1.1047, 1.269, 1.523, 1.827/  
 RI=16, .0012, 1.34  
 C1002 = /0, .6635, .7962, .9554/.0668, .7268, .8722, 1.047/.3085, .7962, .9554, 1.146/  
 .6915, .8722, 1.047, 1.256/.9332, .9554, 1.146, 1.376/1.1047, 1.256, 1.577/  
 RI=14, .0014, 1.14  
 C1102 = /0, .6003, .7204, .8645/.0668, .6845, .8214, .9856/.3085, .7804, .9365, 1.124/  
 .6915, .8854, 1.068, 1.281/.9332, 1.014, 1.217, 1.461/1.1047, 1.269, 1.523, 1.827/  
 RI=15, .0013, 1.24  
 C1202 = /0, .5472, .6566, .7890/.0668, .6475, .7770, .9324/.3085, .7661, .9143, 1.103/  
 .6915, .9065, 1.088, 1.305/.9332, 1.072, 1.287, 1.544/1.1047, 1.269, 1.523, 1.827/  
 RI=16, .0015, 1.34  
 C1003 = /0, .6635, .7962, .9554/.0668, .7268, .8722, 1.047/.3085, .7962, .9554, 1.146/  
 .6915, .8722, 1.047, 1.256/.9332, .9554, 1.146, 1.376/1.1047, 1.256, 1.577/  
 RI=14, .0015, 1.14  
 C1103 = /0, .6003, .7204, .8645/.0668, .6845, .8214, .9856/.3085, .7804, .9365, 1.124/  
 .6915, .8854, 1.068, 1.281/.9332, 1.014, 1.217, 1.461/1.1047, 1.269, 1.523, 1.827/  
 RI=15, .0014, 1.24  
 C1203 = /0, .5472, .6566, .7890/.0668, .6475, .7770, .9324/.3085, .7661, .9143, 1.103/  
 .6915, .9065, 1.088, 1.305/.9332, 1.072, 1.287, 1.544/1.1047, 1.269, 1.523, 1.827/  
 RI=16, .0002, 1.34  
 C1004 = /0, .6635, .7962, .9554/.0668, .7268, .8722, 1.047/.3085, .7962, .9554, 1.146/  
 .6915, .8722, 1.047, 1.256/.9332, .9554, 1.146, 1.376/1.1047, 1.256, 1.577/  
 RI=14, .0005, 1.14

C1104 = /0,.6003,.7204,.8645/.0668,.6845,.8214,.9856/.3085,.7804,.9365,1.124/  
.6915,.8858,1.068,1.281/.9332,1.014,1.217,1.461/1,1.157,1.368,1.666/  
R1=15,.00064,1.2\$  
C1204 = /0,.5472,.6566,.7880/.0668,.6475,.7770,.9324/.3085,.7561,.9193,1.103/  
.6915,.9065,1.088,1.305/.9332,1.072,1.287,1.544/1,1.269,1.523,1.827/  
R1=16,.00075,1.3\$  
C1005 = /0,.6635,.7962,.9554/.0668,.7268,.8722,1.047/.3085,.7962,.9554,1.146/  
.6915,.8722,1.047,1.256/.9332,.9554,1.146,1.376/1,1.047,1.256,1.577/  
R1=14,.00079,1.1\$  
C1105 = /0,.6003,.7204,.8645/.0668,.6845,.8214,.9856/.3085,.7804,.9365,1.124/  
.6915,.8858,1.068,1.281/.9332,1.014,1.217,1.461/1,1.157,1.368,1.666/  
R1=15,.00071,1.2\$  
C1205 = /0,.5472,.6566,.7880/.0668,.6475,.7770,.9324/.3085,.7661,.9193,1.103/  
.6915,.9065,1.088,1.305/.9332,1.072,1.287,1.544/1,1.269,1.523,1.827/  
R1=16,.00083,1.3\$  
C1006 = /0,.6635,.7962,.9554/.0668,.7268,.8722,1.047/.3085,.7962,.9554,1.146/  
.6915,.8722,1.047,1.256/.9332,.9554,1.146,1.376/1,1.047,1.256,1.577/  
R1=14,.00088,1.1\$  
C1106 = /0,.6003,.7204,.8645/.0668,.6845,.8214,.9856/.3085,.7804,.9365,1.124/  
.6915,.8858,1.068,1.281/.9332,1.014,1.217,1.461/1,1.157,1.368,1.666/  
R1=15,.00061,1.2\$  
C1206 = /0,.5472,.6566,.7880/.0668,.6475,.7770,.9324/.3085,.7661,.9193,1.103/  
.6915,.9065,1.088,1.305/.9332,1.072,1.287,1.544/1,1.269,1.523,1.827/  
R1=16,.00055,1.3\$  
SS1=101+102\*103\$  
SS2 = (121+132)\*133\$  
SS3=601+602+602\*604\$  
SS4=705+706\$  
SS5=(2+3)\*5\$  
SS6=9\$  
SS7=10+11+12\$  
SS12 = 1012+102\*103\$  
SS13 = 1013+102\*103\$  
SS14 = 1014+102\*103\$  
SS15 = 1015+102\*103\$  
SS16 = 1016+102\*103\$  
SS22 = (1312+1322)\*13324  
SS23 = (1313+1323)\*13324  
SS24 = (1314+1324)\*13324  
SS25 = (1315+1325)\*13354  
SS26 = (1316+1326)\*13364  
SS72 = 1002+1102+1202\$



S 14 =	.C118,	.C113,	.0075,	.C.0000,	0.0000,	0.0000,	0.0000,
S 15 =	.C119,	.0C77,	C.0000,	C.0000,	0.0000,	0.0000,	
S 16 =	.C154,	C.0000,	C.0000,	C.0000,			
S 17 =	.C230,	.01C7,	.0180,				
S 18 =	.C824,	.0091,					
S 19 =	.C393,						
K-FAC=	2.53, 1.86,	1.98,	1.96, 4.95,	2.92, 2.03,	2.12, 5.75,	2.30, 1.61,	
A 1 =	1.56, 1.56,	1.45,	1.62, 1.65,	3.02, 2.86,	2.81,		
A 2 =	1.0000,	.6123,	.2679,	.5640,	.9087,	.8684,	.0532,
A 3 =	-.4658,	.6125,	.1050,	.0821,	.0621,	.0427,	.1549,
A 4 =	0.0000,	-.1885,	0.0000,				
A 5 =	1.0000,	.4776,	.9213,	.3005,	.5317,	.0841,	.0788,
A 6 =	.9144,	.1539,	.1203,	.1203,	.0611,	.0471,	.2295,
A 7 =	-.2819,	C.0000,					0.0000,
A 8 =	1.0000,	.4400,	.1055,	.2327,	.0526,	.0453,	-.1000,
A 9 =	.1404,	.1057,	.1097,	.0557,	.0429,	.2053,	0.0000,
A 10 =	C.0000,						-.0274,
A 11 =	1.0000,	.2768,	.4898,	.0775,	.0726,	-.2373,	.8424,
A 12 =	.1108,	.1108,	.0563,	.0434,	.2114,	0.0000,	.1418,
A 13 =	1.0000,	.7891,	.0234,	.0219,	-.4642,	.3355,	0.0000,
A 14 =	.C173,	-.0113,	-.0067,	.0630,	0.0000,	-.1055,	.0173,
A 15 =	1.0000,	.0493,	.0462,	-.5724,	.5319,	.0911,	0.0000,
A 16 =	.0371,	.0286,	.1345,	C.0000,	-.1637,	0.0000,	.0713,
A 17 =	1.0000,	.9373,	.7770,	.0727,	.0678,	.0530,	.0629,
A 18 =	.0435,	.0471,	0.0000,	-.0156,	0.0000,		
A 19 =	1.0000,	.7283,	.0682,	.0635,	.0497,	.0457,	.0455,
	.C441,	C.0000,	-.0147,	C.0000,			
	1.0000,	-.2665,	-.0011,	-.0005,	-.0005,	.0281,	-.0443,
	0.0000,	.0876,	C.0000,				
	1.0000,	.1440,	.1126,	.1126,	.0486,	.0375,	0.0000,
	-.2839,	C.0000,					
	1.0000,	.2151,	.2151,	.2555,	.1570,	.1912,	0.0000,
	C.0000,						
	1.0000,	.5546,	.1547,	.1540,	.1455,	0.0000,	0.0000,
	1.0000,	.1997,	.1540,	.1495,	0.0000,	0.0000,	0.0000,
	1.0000,	.3557,	.4210,	C.0000,	0.0000,	0.0000,	0.0000,
	1.0000,	.2065,	C.0000,	C.0000,	C.0000,		
	1.0000,	C.0000,	C.0000,	C.0000,			
	1.0000,	.4572,	.4174,				
	1.0000,	.4223,					
A 19 =	1.0000,						

NDMPWH=6CC,53

W=5  
OP=2CC,4CC,500,600,700,300,1000,1200  
S THE NEXT CARD REPLACES THE PRECEDING CARD TO REMOVE 200 PSI  
OP= 4CC,500,600,700,900,1000,1200  
RUN\$

\$ UNA SAMPLE PRJHLEM - YIELD AND FUM PERTURBATIONS  
 PLOTON=400\$  
 \$ ELECTRONIC RACKS  
 C101=/0.30.35.40/1.40.45.50/R1=13.2.1.2\$  
 C102=/0.15.17.19/1.19.21.23/R1=17/USER(1)/R1=R1.1.0.1.2\$  
 C103=/0.50/1.50/P1=18.1.0.1.2\$  
 C1016 = /0.60.00.70.00.30.00/1.80.00.90.00.100.0/R1.13.2.1.2\$  
 \$ BATTERY MOUNTS  
 C601=/C.500.550.600/1.600.650.700/R1=4.1.0.1.2\$  
 C602=/0.175.200.225/1.225.250.275/R1=8/USER(2)/R1=R1.1.0.1.2\$  
 C603=/0.20/1.20/R1=1.1.1/USER(4)\$  
 C604=/C.10/1.10/R1=6.1.1\$  
 \$ BATTERY ELEMENTS  
 C705=/C.450.500.550/1.600.650.700/R1=2.1.1.2\$  
 C706=/C.150.175.200/1.225.250.275/R1=8.1.1.2\$  
 \$ SIS  
 C131=/0.30.35.40/1.35.40.45/R1=1.1.1.2\$  
 C132=/0.30.35.40/1.35.40.45/R1=5.2.1.3\$  
 C133=/0.30.35.40/1.35.40.45/R1=4.08.1.2\$  
 C1316 = /0.96.00.112.00.128.00/1.112.00.128.00.144.00/R1=1.1.1.2\$  
 C1326 = /0.95.00.112.00.128.00/1.112.00.128.00.144.00/R1=5.2.1.3\$  
 C1336 = /0.96.00.112.00.128.00/1.112.00.128.00.144.00/R1=4.08.1.2\$  
 \$ COMMUNICATIONS  
 C2 = /0.7237.9408.1.223/.0668.7416.9641.1.253/.3085.7599.9879.1.284/  
 R1=11.000400.1.3/USER(3)\$  
 C3 = /0.4704.9408.1.892/.0668.4820.9641.1.928/.3085.4939.9879.1.970/  
 R1=19.04.1.4\$  
 C5 = /0.6125.7962.1.035/.0668.6709.8722.1.134/.3035.7350.9554.1.242/  
 .6915.8051.1.047.1.361/.9332.8819.1.146.1.490/1.9661.1.256.1.633/  
 R1=10.0002.1.3\$  
 \$ BLAST VALVE  
 C9 = /0.6635.7962.9554/.0668.7263.8722.1.047/.3085.7962.9554.1.146/  
 .6915.8722.1.047.1.256/.9332.9554.1.146.1.376/1.1.047.1.256.1.577/  
 R1=0.001.1.2\$  
 \$ FACILITY  
 C10 = /0.6035.7902.9554/.0668.7263.8722.1.047/.3085.7962.9554.1.146/  
 .6915.8722.1.047.1.256/.9332.9554.1.146.1.376/1.1.047.1.256.1.577/  
 R1=14.0015.1.1\$  
 C11 = /0.6033.7209.945/.0668.8214.9850/.065.7804.9365.1.124/

```

.6915..8358.1.034.1.281/.9552.1.021.1.217.1.461/1.1.157.1.083.1.666/
R1=15..00135.1.23
C12 = /0..5472..6566..7880/.0658..6475..7770..9324/.3085..7661..9193.1.103/
.6915..9065.1.088.1.305/.9332.1.072.1.267.1.544/1.1.239.1.023.1.827/
R1=16..0012.1.33
C1006 = /0..6535..7962..9554/.0668..7208..8722.1.047/.3085..7962..9554.1.146/
.6915..8722.1.047.1.256/.9332..9554.1.146.1.376/1.1.047.1.256.1.577/
R1=14..00068.1.13
C1106 = /0..6003..7204..8645/.0668..6445..8214..9856/.3085..7804..9365.1.124/
.6915..8898.1.068.1.281/.9332.1.014.1.217.1.461/1.1.157.1.388.1.666/
R1=15..00061.1.23
C1206 = /0..5472..6566..7880/.0668..6475..7770..9324/.3085..7661..9193.1.103/
.6915..9065.1.088.1.305/.9332.1.072.1.267.1.544/1.1.239.1.023.1.827/
R1=16..00055.1.33
SS1=101+102*103$
SS2 = (131+132)*133$
SS3=601+603+602*604$
SS4=705+706$
SS5=(2+3)*5$
SS6=9$
SS7=10+11+12$
SS16 = 1016+102*103$
SS26 = (1316+1326)*1336$
SS76 = 1006+1106+1206$
SYS16 = SS16+SS26+SS3+SS4+SS5+SS6+SS75$
$ THE FOLLOWING IS A FEATURE OF THE PROGRAM WHICH SUPPRESSES COMP AND
$ SUBSYSTEM EQUATION CARDS.
C101=$
C131=$
C132=$
C133=$
C10=$
C11=$
C12=$
SS1=$
SS2=$
SS7=$
MODE=16*4.3*1
ULN=2.785.5.94.4.094.6.340.2.639.1.435.3.807.4.205.0.642*7.785.7.244.
4.605.5.298.6.397.6.397.6.397.3.296.3.367.2.485
ALPHA=.17..083..083..25..17..083..083.0..08.04..04..04..04..04..08.
.17.1.33..07.1.53

```

BETA = .67, 1.1, 1.1, .33, .67, .5, .5, .33, 1.3, 1.3, 1.3, 5, 5, 1, .3, .2, .2, .3, .86

S 1 =	.1927,	.1767,	.3448,	.0894,	.1767,	.2031,	.1927,	.0111,	.0111,
	-.1705,	.3448,	.0116,	.0085,	.0085,	.0015,	.0015,	.0015,	.0224,
S 2 =	0.0000,	-.0873,	0.0000,	.1713,	.1767,	.0110,	.0110,	-.1546,	0.0000,
	-.1765,	.0892,	.1765,	.0083,	.0013,	.0223,	.0223,	0.0000,	0.0000,
	.3445,	.0114,	.0083,	.0894,	.0013,	.0072,	.0072,	.1737,	.1737,
S 3 =	-.0873,	0.0000,	.0892,	.0083,	.0013,	.0223,	0.0000,	-.0169,	-.0169,
	.0114,	.0083,	.0892,	.0083,	.0013,	.0223,	0.0000,	-.0169,	-.0169,
S 4 =	0.0000,	-.1765,	.0083,	.1767,	.0110,	-.1546,	.3445,	.0114,	.0114,
	.0083,	.2031,	.0084,	.0013,	.0223,	0.0000,	-.0873,	0.0000,	0.0000,
S 5 =	-.2328,	.0031,	-.0039,	.0084,	-.1862,	.3351,	.0042,	.0031,	.0031,
	.0031,	-.0039,	.0173,	.0084,	-.1862,	.3351,	.0042,	.0031,	.0031,
S 6 =	-.2099,	.0111,	-.0111,	.0111,	.3448,	.0116,	.0085,	.0085,	.0085,
	.0015,	.0015,	.0043,	.0111,	.3448,	.0116,	.0085,	.0085,	.0085,
S 7 =	.0042,	.0042,	.0042,	.0190,	.0057,	.0042,	.0042,	.0042,	.0042,
	.0042,	.0042,	.0042,	.0190,	.0057,	.0042,	.0042,	.0042,	.0042,
S 8 =	.0042,	.0042,	.0042,	.0190,	.0057,	.0042,	.0042,	.0042,	.0042,
	.0042,	.0042,	.0042,	.0190,	.0057,	.0042,	.0042,	.0042,	.0042,
S 9 =	.1828,	-.3068,	-.0002,	.0000,	.0000,	.0000,	.0000,	.0000,	.0000,
	.0000,	-.0002,	.0000,	.0000,	.0000,	.0000,	.0000,	.0000,	.0000,
S 10 =	-.6745,	.0205,	.0150,	.0150,	.0010,	.0010,	.0428,	0.0000,	0.0000,
	.0150,	.0150,	.0150,	.0150,	.0010,	.0010,	.0428,	0.0000,	0.0000,
S 11 =	-.0162,	.0114,	.0114,	.0114,	.0114,	.0114,	.0114,	0.0000,	0.0000,
	.0114,	.0114,	.0114,	.0114,	.0114,	.0114,	.0114,	0.0000,	0.0000,
S 12 =	.0083,	.0083,	.0083,	.0083,	.0033,	0.0000,	0.0000,	0.0000,	0.0000,
	.0083,	.0083,	.0083,	.0083,	.0033,	0.0000,	0.0000,	0.0000,	0.0000,
S 13 =	.0083,	.0083,	.0083,	.0083,	.0033,	0.0000,	0.0000,	0.0000,	0.0000,
	.0083,	.0083,	.0083,	.0083,	.0033,	0.0000,	0.0000,	0.0000,	0.0000,
S 14 =	.0118,	.0113,	.0075,	.0075,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
	.0118,	.0113,	.0075,	.0075,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
S 15 =	.0118,	.0077,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
	.0118,	.0077,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
S 16 =	.0154,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
	.0154,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,	0.0000,
S 17 =	.0230,	.0107,	.0180,	.0180,	.0180,	.0180,	.0180,	.0180,	.0180,
	.0230,	.0107,	.0180,	.0180,	.0180,	.0180,	.0180,	.0180,	.0180,
S 18 =	.0824,	.0091,	.0091,	.0091,	.0091,	.0091,	.0091,	.0091,	.0091,
	.0824,	.0091,	.0091,	.0091,	.0091,	.0091,	.0091,	.0091,	.0091,
S 19 =	.0393,	.0393,	.0393,	.0393,	.0393,	.0393,	.0393,	.0393,	.0393,
	.0393,	.0393,	.0393,	.0393,	.0393,	.0393,	.0393,	.0393,	.0393,
K-FAC =	2.53, 1.86,	1.98,	1.96, 4.55,	2.92,	2.03, 2.12,	5.75,	3.30, 1.61,		
	1.56, 1.56,	1.45,	1.62, 1.55,	3.03,	2.80, 2.81,				
A 1 =	1.0000,	.6123,	.2679,	.5640,	.9087,	.0684,	.0567,	.0532,	.0532,
	.4958,	.6125,	.1050,	.0821,	.0821,	.0427,	.0330,	.1547,	.1547,
A 2 =	0.0000,	-.1885,	0.0000,	.3005,	.5317,	.0841,	.0780,	-.2576,	-.2576,
	1.0000,	.4776,	.9213,	.1203,	.0611,	.0471,	.2295,	0.0000,	0.0000,
	.9144,	.1539,	.1203,	.1203,	.0611,	.0471,	.2295,	0.0000,	0.0000,
	-.2819,	0.0000,							

A 3 =	1.0000.	.4400.	.1053.	.2327.	.0526.	.0493.	-.1000.	.3963.
	.1404.	.1097.	.1097.	.0557.	.0429.	.2093.	0.0000.	-.0274.
	0.0000.							
A 4 =	1.0000.	.2768.	.4898.	.0775.	.0726.	-.2373.	.8424.	.1418.
	.1108.	.1108.	.0563.	.0434.	.2114.	0.0000.	-.2597.	0.0000.
A 5 =	1.0000.	.7891.	.0234.	.0219.	-.4642.	.3355.	.0221.	-.0173.
	.0173.	-.0113.	-.0067.	.0630.	0.0000.	-.1095.	0.0000.	
A 6 =	1.0000.	.0493.	.0462.	-.5724.	.5319.	.0911.	-.0713.	.0713.
	.0371.	.0286.	.1345.	0.0000.	-.1637.	0.0000.	.0530.	.0629.
A 7 =	1.0000.	.9373.	.7770.	.0727.	.0678.	.0530.	.0530.	.0629.
	.0485.	.0471.	0.0000.	-.0156.	0.0000.	.0497.	.0590.	.0455.
A 8 =	1.0000.	.7283.	.0682.	.0635.	.0497.	.0497.	.0590.	.0455.
	.0441.	0.0000.	-.0147.	0.0000.	.0497.	.0497.	.0590.	.0455.
A 9 =	1.0000.	-.2669.	-.0011.	-.0009.	-.0009.	.0281.	.0216.	-.0443.
	0.0000.	.0876.	0.0000.			.0375.	.2276.	0.0000.
A 10 =	1.0000.	.1440.	.1126.	.1126.	.0486.	.0375.	.2276.	0.0000.
	-.2839.	0.0000.				.1912.	0.0000.	0.0000.
A 11 =	1.0000.	.2151.	.2151.	.2555.	.1970.	.1912.	0.0000.	0.0000.
	0.0000.							
A 12 =	1.0000.	.3346.	.1997.	.1540.	.1495.	0.0000.	0.0000.	0.0000.
A 13 =	1.0000.	.1997.	.1540.	.1495.	0.0000.	0.0000.	0.0000.	0.0000.
A 14 =	1.0000.	.3557.	.4210.	0.0000.	0.0000.	0.0000.	0.0000.	0.0000.
A 15 =	1.0000.	.2069.	0.0000.	0.0000.	0.0000.	0.0000.	0.0000.	0.0000.
A 16 =	1.0000.	0.0000.	0.0000.	0.0000.	0.0000.	0.0000.	0.0000.	0.0000.
A 17 =	1.0000.	.4572.	.4179.					
A 18 =	1.0000.	.2223.						
A 19 =	1.0000.							

NOMPW=600.55

W=5

OP=200.400.500.600.700.800.1000.1200

\$ THE NEXT CARD REPLACES THE PRECEDING CARD TO REMOVE 200 PSI

OP= 400.500.600.700.800.1000.1200

RUN\$

\$HIGH YIELD HOB

C603 = /0.20/1.20/R1=1.2.1/USER(4) \$  
C1316 = /0.96.00.112.00.128.00/1.112.00.128.00.144.00/R1=1.2.1.2 \$  
C1326 = /0.96.00.112.00.128.00/1.112.00.128.00.144.00/R1=5.8.1.3 \$  
C9 = /0..6635..7962..9554/.0668..7268..8722.1.047/.3085..7962,.5554,1.146/  
.6915,.8722,1.047,1.256/.9332..9554,1.146,1.376/1.1.047,1.256,1.577/  
R1=0..0011,1.2 \$  
C1006 = /0..6635..7962..9554/.0668..7268..8722,1.047/.3085,.7962,.9554,1.146/  
.6915,.8722,1.047,1.256/.9332..9554,1.146,1.376/1.1.047,1.256,1.577/  
R1=14..000748,1.1 \$  
C1106 = /0..6003..7204..8645/.0668..6845..8214..9856/.3085,.7804,.9365,1.124/  
.6915,.8898,1.068,1.281/.9332,1.014,1.217,1.461/1.1.157,1.388,1.666/  
R1=15..000671,1.2 \$  
C1206 = /0..5472..6566..7880/.0668..6475..7770..9324/.3085,.7661,.9193,1.103/  
.6915..9065,1.088,1.305/.9332,1.072,1.287,1.544/1.1.269,1.523,1.827/  
R1=16..000605,1.3 \$  
M=5.H=1.5  
GAMMA=16\*0..34.6,17.3.42.1  
RUNS

\$ LOW YIELD HOB

M=.5.H=1.5  
RUNS

APPENDIX B

LISTING OF FAST COMPUTER CODE

```

PROGRAM MAIN (INPUT,OUTPUT,TAPES=INPUT,TAPES=OUTPUT)
COMMON/WORKEY / IKEY, IKEY, MKFY, IPAD1, WORK (135)
COMMON/INPUT / IREAD, IPRINT, IPRINT, IPRINT, IPRINT
COMMON/CARDAT / NCARD, IFCRD, IPRINT, IPRINT, NCODE,
* NIJLNT, NEXT, IREP, IPRINT, FRCPRF, IING, IEXPT,
* ICOL, ICOL, MCOL, ICARD(80)
COMMON /CHRIST/ KSET(47)
COMMON/NLIST/ NAMES(219)
COMMON/CF01 / NRYN, VHYM, MODE(30,4)
COMMON/CF02 / IILN (30,2)
COMMON/CF03 / FACTK (30,2)
COMMON/CF04 / ALPHA (30)
COMMON/CF05 / BETA (30,2)
COMMON/CF09 / GAMMA (30)
COMMON/CF09A/ HR (30)
COMMON/CF09R/ GAMMA(30)
COMMON/CF06A/ SIGSS (445)
COMMON/CF06R/ SIGSSS(445)
COMMON/CF06C/ CSS(445)
COMMON/CF06D/ IXSS(30)
COMMON/CF07A/ SIGAU (445)
COMMON/CF07R/ RHO(445)
COMMON/CF07C/ RHOCS(445)
COMMON/CF07D/ CAU(445)
COMMON/CF07E/ IXAU(30)
COMMON/CF08 / RMDY(30,30)
COMMON/CR01 / IONSS, IIMIN, NIMAX, MINIP, NCDEFI, NIIMR, NIIMNR,
* EPSI, EPSO, EPSOC, EPSMAT, CONE1(7)
COMMON/CR02 / W, W, H0, 4, NPO, R0, R(20)
COMMON/CR03 / NB, NBP, NP(20)
COMMON/CNC1 / NCR, NCYS, MNAME, NAME(135)
COMMON/CNC2 / INSLIC(2,135)
COMMON/CNC3 / MDS, IAD2, P(150)
COMMON/CC01 / NCINS, MCINS, INSC(4,200)
COMMON/CC02 / CIEF(3,200)
COMMON/CC03 / EGY(4,100)
COMMON/CC04 / EGY(4,100)
COMMON/CC05 / EC (4,100)
COMMON/CC06 / EY (4,100)
COMMON/CC07 / TD(P,100)
COMMON/CC01 / MINS, MININS, INSNET(4,240)

```

```

COMMON/CT01 / X(130)
COMMON/CT02 / XBAR(130)
COMMON/CT03 / XDEL(130)
COMMON/CT04 / DEG(130)
COMMON/CT05 / P7(135)
COMMON/CT06 / PSAVE(22,135)
COMMON/CT07 / OPTMCH,MPROR,MPLOT,P304(20,45)
COMMON/CHETA / IPAD3,MAXHD,DMAX,DM(9),RD(21,4)
DIMENSION DR(21,9)
DATA DR/0.030,051,071,090,110,129,148,168,189,211
*
* 234,258,286,313,345,382,426,482,566,1..
* 0.061,091,117,140,162,184,205,227,248,271
* 294,319,346,371,402,437,477,528,602,1..
* 0.106,145,176,203,228,251,274,296,319,341
* 364,388,413,440,463,501,533,586,652,1..
* 0.163,209,244,274,301,326,350,373,396,419
* 442,465,490,516,543,574,603,652,711,1..
* 0.225,279,318,350,379,405,430,454,477,500
* 523,546,570,595,621,650,682,721,775,1..
* 0.289,348,391,426,457,484,510,535,558,581
* 606,627,650,674,693,726,756,791,837,1..
* 0.348,414,461,499,531,560,587,612,636,659
* 681,704,726,749,772,797,826,855,894,1..
* 0.398,472,523,563,598,629,656,682,706,729
* 752,773,795,816,839,860,883,909,933,1..
* 0.436,518,574,618,655,687,716,742,766,789
* 811,832,852,871,890,910,923,949,970,1..

```

DIMENSION ISET(47)

```

DATA ISET/
* 4HA 4HB 4HC 4HD 4HE 4HF 4HG 4HH 4HI 4HJ 4HK 4HL 4HM 4HN 4HO 4HP 4HQ 4HR 4HS 4HT 4HU 4HV 4HW 4HX 4HY 4HZ 4HA=
* 4HB= 4HC= 4HD= 4HE= 4HF= 4HG= 4HH= 4HI= 4HJ= 4HK= 4HL= 4HM= 4HN= 4HO= 4HP= 4HQ= 4HR= 4HS= 4HT= 4HU= 4HV= 4HW= 4HX= 4HY= 4HZ=
* 4HA= 4HB= 4HC= 4HD= 4HE= 4HF= 4HG= 4HH= 4HI= 4HJ= 4HK= 4HL= 4HM= 4HN= 4HO= 4HP= 4HQ= 4HR= 4HS= 4HT= 4HU= 4HV= 4HW= 4HX= 4HY= 4HZ=
DIMENSION NAMES1(96)
DIMENSION NAMES2(103)
DATA NAMES1/
* 1. 4HA 4HB 4HC 4HD 4HE 4HF 4HG 4HH 4HI 4HJ 4HK 4HL 4HM 4HN 4HO 4HP 4HQ 4HR 4HS 4HT 4HU 4HV 4HW 4HX 4HY 4HZ
* 2. 4HA= 4HB= 4HC= 4HD= 4HE= 4HF= 4HG= 4HH= 4HI= 4HJ= 4HK= 4HL= 4HM= 4HN= 4HO= 4HP= 4HQ= 4HR= 4HS= 4HT= 4HU= 4HV= 4HW= 4HX= 4HY= 4HZ
* 3. 1,4HC

```



```

I PRINT=6
I PUNCH=0
I TAPE=0
C * * * * *
C * STORE META DISTRIBUTIONS IN COMMON BLOCK /CR-TA/
C * * * * *
RM=0.0
DO 5 I=1.9
RM=RM+0.1
DM(I)=RM
DO 5 J=1.21
RD(I,J)=DR(I.1)
5 CONTINUE
MAYRD=21
RM MAX=20.0
C * * * * *
C * STORE CHARACTER SET OF THIS COMPUTER IN COMMON BLOCK /CHRLST/
C * * * * *
DO 1 I=1.67
1 KSET(I)=1SET(I)
C * * * * *
C * STORE INPUT NAMES IN COMMON BLOCK /MLIST/
C * * * * *
DO 2 I=1.94
2 NAMES(I)=NAMES1(I)
J=94
DO 3 I=1.103
J=J+1
3 NAMES(I)=NAMES2(I)
C * * * * *
C * SET MAXIMUM PROBLEM SIZE DEPENDING ON COMMON BLOCK DIMENSIONS
C * * * * *
MKEY=135
MPYM=30
MMABE=135
MPS=150
MCINSE=200
MNTINS=250
MPOINT=20
C * * * * *
C * SET DEFAULT VALUES - MUST HAVE BE RESET BY INPUT CARDS
C * * * * *

```

Reproduced from  
best available copy.

```

* * * * *
C * * * * *
  IONES=0
  NIM4=5
  NIMAX=50
  NIMIN=50
  EPS1=.05
  EPS0=.03
  EPSIC=.02
  NCOWF=1
  COWF(1)=0.5
  NIMIR=1
  NIMNP=1
  EPSMAT=0.0001
  NPINCH=0.0
C * * * * *
C * * * * *
C * * * * *
  ***** IFCRD=0 WILL SUPPRESS PRINTING OF CARD INPUT
  IFCRD=1PRINT
  NCIRD=0
  CALL MLREAD
  STOP
  END

```

```

SUBROUTINE M0250
COMMON/IDENT1/ IREAD,IPRINT,IPUNCH,ITAPE
COMMON/CARDAT/ MCARD,TECHO,INHIT,MCODE,
* PHICNT,NEXT,IREP,INTPRT,FRCPRT,LOG,TEXPRT,
* ICOL,ICOL,MCOL,ICARD(40)
COMMON/CE01 / MRYN,MRYM,MODE(1)
COMMON/CE02 / ULM (1)
COMMON/CE03 / FACTK (1)
COMMON/CE04 / ALPHA (1)
COMMON/CE05 / BETA (1)
COMMON/CE06R/ SIGSSS(1)
COMMON/CE07C/ REFCOS(1)
COMMON/CE09 / GAMMA (1)
COMMON/CE09A/ HR (1)
COMMON/CE09R/ GAMMAR(1)
COMMON/CR01 / TROSS,NIMIN,NIMAX,NOMIN,NCONEI,NIMUR,NIMUR,
* EPSI,EPSC,EPSCC,EPSCAT,CONEI(7)
COMMON/CR02 / W0,W,H0,H,OP0,RO,R(1)
COMMON/CR03 / NR,NTR,OP(1)
COMMON/CT07 / OPINCH,NPROR,MPL0T,PROR(1)
C* * * * *
C RESET$ ASSUMED REFORM FIRST JOB IS READ
C* * * * *
CALL RESET
C* * * * *
C READ THE NEXT DATA CARD
C* * * * *
1 CONTINUE
WRITE(IPRINT,999)
999 FORMAT(1H)
CALL M0250
C* * * * *
C IDENTIFY TYPE OF CARD BY INPUT NAME
C* * * * *
2 CONTINUE
CALL NAMEFIX,TYPE,ISUB1,ISUB2)
GO TO(10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,
* 180,190,200,210,220,230,240,250,260,270)
* NAMEFIX
C* * * * *
C RIMS CARD

```

DO NOT EXECUTE WITH THIS DATA IF AN ERROR HAS BEEN FOUND

```
C
C IF NO INPUT ERROR -
C SELECT THE REQUIRED ENVIRONMENTS
C CHECK FOR NETWORK ERRORS
C PERFORM MONTE CARLO PROCESS
C* * * * *
10 IF(INHIRT.NE.0) GO TO 1
   CALL SELECT
   CALL NETCK(INHIRT)
   IF(INHIRT.NE.0) GO TO 1
   WRITE(TPRINT,999)
   CALL PRNG
   GO TO 1
C***** STOP$ CARD
20 RETURN
C***** COMPONENT CARD
30 CALL COMPNT(IDES)
   GO TO 2
C***** SUBSYSTEM CARD
40 CALL RNDLE( IDES)
   GO TO 2
C***** SYSTEM CARD
50 CALL RNDLE( -IDES)
   GO TO 2
C***** ENV CARD
60 ISUBJ=IDES+MRYM
   CALL NIINTG(MODE (ISUBJ))
   CALL NLREAL(IUN (ISUBJ))
   CALL NLREAL(FACTK(ISUBJ))
   CALL NLREAL(ALPHA (IDES))
   CALL NLREAL(BETA (ISUBJ))
   IF(NEXT.GT.1) GO TO 2
   CALL NLREAL(GAMMA(IDES))
   IF(NEXT.GT.1) GO TO 2
   CALL NLREAL(HH(IDES))
   CALL NLREAL(GAMMAH(IDES))
   GO TO 2
C***** S MATRIX CARD
70 ISUBJ=(IDES-1)*MRYM-(IDES-1)*(IDES-2))/2+ISUBJ
   CALL NLVECP(SIGSSS(ISUBJ),ISUBJ)
   GO TO 2
```

```

C***** A MATRIX CARD
R0 TSUR1=(IDES-1)*RVM-(IDES-1)*(IDES-2)/2+TSUR1
CALL N1VECR(RHCONS(TSUR1),TSUR2)
GO TO 2
C***** CONF1 CARD
90 CALL N1INTG(N1MIN)
IF(NEXT.EQ.1) CALL N1INTG(N1MAX)
IF(NEXT.EQ.1) CALL N1INTG(N1MIN)
IF(NEXT.EQ.1) CALL N1REAL(EPS1)
IF(NEXT.EQ.1) CALL N1REAL(EPS0)
IF(NEXT.EQ.1) CALL N1REAL(EPSDC)
IF(NEXT.GT.1) GO TO 2
NCONF1=0
91 NCONF1=NCONF1+1
CALL N1REAL(CONF1(NCONF1))
IF(NEXT.EQ.1) GO TO 91
GO TO 2
C***** NOMPWH CARD
100 CALL N1REAL(R0)
CALL N1REAL(W0)
IF(NEXT.EQ.1) CALL N1REAL(H0)
CALL RTOP(R0#6.0761,OP0,W0#1000.,H0)
IF(OP0.LT.100.) WRITE(IPRINT,2000)
2000 FORMAT(37HMAPING - PRESSURE LESS THAN 100 PSI )
GO TO 2
C***** RANGE CARD
110 CALL N1VECR(R,TSUR1)
NR=TSUR1
NDP=0
GO TO 2
C***** PRESSURE CARD
120 CALL N1VECR(OP,TSUR1)
NRP=TSUR1
NRP=NDP
GO TO 2
C***** YIELD CARD
130 CALL N1REAL(W)
GO TO 2
C***** RESETS CARD
140 CALL RESET
GO TO 2

```

```

C***** RANIN CARD
150 CALL MLINTG(NUMMIR)
    CALL MLINTG(NUMMR)
    GO TO 2
C***** MODE CARD
160 CALL NLVECT(MODE(MRYM+1),[SIIR])
    GO TO 2
C***** IIN CARD
170 CALL NLVECR(IIN(MRYM+1),[SIIR])
    GO TO 2
C***** K-FAC CARD
180 CALL NLVECR(FACTK(MRYM+1),[SIIR])
    GO TO 2
C***** ALPHA CARD
190 CALL NLVECR(ALPHA,[SIIR])
    GO TO 2
C***** BETA CARD
200 CALL NLVECR(BETA(MRYM+1),[SIIR])
    GO TO 2
C***** PLINOFF$ CARD
210 OPINCH=0.0
    GO TO 2
C***** PLINON CARD
220 CALL NIREAL(OPINCH)
    NPROR=0
    GO TO 2
C***** PLINT$ CARD
230 IF(INHIRT.NE.0) GO TO 2
    CALL DUPLNT
    NPROR=0
    GO TO 1
C***** GAMMA CARD
240 CALL NLVECR(GAMMA,[SIIR])
    GO TO 2
C***** 44-GAMMA CARD
250 CALL NIREAL(HR([SIIR]))
    IF(MEXI.GT.1) GO TO 2
    [SIIR]=[SIIR]+1
    GO TO 250
C***** MUDUWH CARD

```

```
250 CALL MLREAL (RPO)
CALL MLREAL (RPO)
IF (NEXT.EQ.1) CALL MLREAL (RPO)
CALL PTHR (RPO,RPO,WC*1000.,HC)
IF (RPO.EQ.0.) INHIRT=INHIRT+1
R0=R0/6.,0761
GO TO 2
C***** HOR CARD
270 CALL MLREAL (H)
GO TO 2
END
```

```

SUBROUTINE PSEF1
COMMON/CARDAT/ NCARD, IECNO, INHIRT, NCODE,
*           MILENT, NEXT, IREP, INTPT, ERCPRT, ILOG, IEXPT,
*           ICOL, LCOL, MCOL, ICARD(40)
COMMON/CE01 / NRYN, MRYM, MODE(1)
COMMON/CE02 / GAMMA (1)
COMMON/CE03 / HA (1)
COMMON/CE04 / SIGSS(1)
COMMON/CE05 / RHOCDS(1)
COMMON/CR02 / W0, W, HD, H, UP0, RO, R(1)
COMMON/CR03 / NR, NRP, NP(1)
COMMON/CNC1 / NCR, NSYS, MNAME, NAME(1)
COMMON/CC01 / NCINS, MCINS, INSC(4,1)
COMMON/CNO1 / NNINS, MNINS, INSNF(4,1)
COMMON/CT07 / OPINCH, NPROR, MPLNT, PROR(1)
C * * * * *
C * TURN OFF THE INPUT ERROR FLAG AND NUMERIC FIELD RPFAT
C * * * * *
INHIRT=0
IREP=0
C * * * * *
C * INITIALIZE ENVIRONMENTS
C * SELECT INTERPRETS MODE(K)=-1 AS A MISSING ENVIRONMENT
C * MODE(KK)=0 UNLESS THIS ENVIRONMENT IS REQUIRED BY A COMPONENT
C * HP(1)=-1 ASSUMES NO HOR BREAK POINT
C * GAMMA(1)=0 ASSUMES NO HOR SCALING
C * SFT S AND A MATRICES TO ZERO
C * * * * *
K=MRYM
KK=K+MRYM
DO 1 I=1,MRYM
K=K+1
KK=KK+1
MODE(K)=-1
MODE(KK)=0
GAMMA(1)=0.
HP(1)=-1.
1 CONTINUE
KK=(MRYM*(MRYM+1))/2
DO 2 I=1, KK
SIGSS(I)=0.

```

```

RHO(05(I))=0.
2 CONTINUE
C * * * * *
C * THE FOLLOWING VARIABLES WILL CONTAIN THE NUMBER OF RANGES.
C * PRESSURES, COMPONENTS, SURSYS AND SYS, TRANSFER FUNCTION
C * INSTRUCTIONS AND NETWORK INSTRUCTIONS.
C * NPROB IS USED TO STORE PLOT POINTS (NPROB= TOTAL NUMBER
C * OF PRESSURES FOR ONE OR MORE STACKED CASES).
C * ASSUME ZERO HOR
C *
NR=0
NPP=0
NCR=0
NSYS=0
NCINS=0
NMINS=0
NPROB=0
H=0.
HC=0.
RETURN
END

```

Reproduced from  
best available copy.

```

SURQUITINE MUGET
COMMON/INUIT/ READ,IPRINT,IPUNCH,ITAPE
COMMON/CARDAT/ NCARD,IECHO,INHIT,NCODE,
NILCNT,NEXT,IREP,INTPRT,FRCPRT,ILNG,IFXPRT,
*
* COMMON /CHRIST/
ICOL,LCOL,MCOL,ICARD(RO)
KA,KB,KC,KD,KE,KF,KG,KH,KI,KJ,
KK,KL,KM,KN,KO, KP, KQ, KR, KS, KT,
KU, KV, KW, KX, KY, KZ, KO, KI, K2, K3,
K4, K5, K6, K7, K8, K9, KCM, KPL, KMI, KAS,
KSL, KPR, KLP, KRP, KFO, KDL, KRL
*
DIMENSION IT(47),KCODE(2,47)
EQUIVALENCE (IT,KA)
* * * * *
C * * * * * READ AN RO-COLUMN CARD - ONE CHARACTER/WORD
C * * * * * NCARD CONTAINS NUMBER OF CARDS READ (USED ONLY FOR ERROR MESSAGES)
C * * * * *
C * * * * * 1 READ(IREAD,1000) (ICARD(I),I=1,RO)
1000 FORMAT(ROA1)
C * * * * * NCARD=NCARD+1
C * * * * * UNLESS THIS IS A RECODE CARD GO TO 5
C * * * * *
IF(ICARD(1).NE.KR) GO TO 5
IF(ICARD(2).NE.KF) GO TO 5
IF(ICARD(3).NE.KC) GO TO 5
IF(ICARD(4).NE.KO) GO TO 5
IF(ICARD(5).NE.KD) GO TO 5
IF(ICARD(6).NE.KF) GO TO 5
C * * * * * IDENTIFY WHICH CHARACTERS TO RECODE - THEN READ NEXT CARD
C * * * * *
C * * * * * NCODE=0
DO 2 I=1,47
IF(ICARD(I+7).EQ.IT(I)) GO TO 2
NCODE=NCODE+1
KCODE(1,NCODE)=ICARD(I+7)
KCODE(2,NCODE)=IT(I)
2 CONTINUE
GO TO 1
C * * * * * NCODE=0 (SET IN MAIN) UNLESS A RECODE CARD DID NOT MATCH /CHRIST/
C

```



```

SUBROUTINE NL_NAME(NAMFIX,IOFS,ISUR1,ISUR2)
COMMON/IUNIT/ IREAD,IPRINT,IPUNCH,ITAPE
COMMON/CARDAT/ NCARD,ICARD,INHIT,NCODE,
* NHLCNT,NEXT,IREP,INTPR,FRCPRF,IUNG,I=XPRT,
* ICOL,LCOL,MCOL,ICARD(IR)
COMMON /CHRIST/ KA,KR,KC,KD,KE,KF,KG,KH,KI,KJ,
* KK,KL,KM,KN,KO,KP,KQ,KR,KS,KT,
* KU,KV,KW,KX,KY,KZ,KO,KI,K2,K3,
* K4,K5,K6,K7,K8,K9,KCM,KPL,KMI,KAS,
* KSI,KPR,KLP,KQP,KFO,KOI,KRI
DIMENSION MATCH(10)
EQUIVALENCE (MATCH,KO)
COMMON /NLIST/ NAMES(1)
C * * * * *
C SFT SUBSCRIPTS FOR VECTOR (ISUR1) AND MATRIX (ISUR1,ISUR2)
C VECTOR=... IS UNDERSTOOD TO MEAN VECTOR(1)=...
C MATRIX=... IS UNDERSTOOD TO MEAN MATRIX(1,1)=...
C * * * * *
50 ISUR1=1
ISUR2=1
C * * * * *
C THERE ARE TWO CLASSES OF NAMES -
C IPASS=1 LOOKS FOR NAMES WITHOUT DIGITS
C IPASS=2 LOOKS FOR NAMES WITH DIGITS (E.G., C.S.S AND ENV)
C * * * * *
IPASS=1
C * * * * *
C FORM THE NAME BY REMOVING BLANKS AND LOOKING FOR A
C ( * = * OR END-OF-CARD.
C * * * * *
NCAR=0
1 IF(ICARD(ICOL),FO,K=0) GO TO 4
IF(ICARD(ICOL),FO,KO1) GO TO 5
IF(ICARD(ICOL),FO,KRI) GO TO 2
IF(ICARD(ICOL),FO,KLP) GO TO 3
C * * * * * STORE CHARACTER
NCAR=NCAR+1
ICARD(NCAR)=ICARD(ICOL)
2 ICOL=ICOL+1
IF(ICOL,GT,LCOL) GO TO 4
GO TO 1

```

```

C***** FOUND ( LOOK FOR SUBSCRIPTS (TSUB1) OF (TSUB1,TSUB2)
3  ICOL=ICOL+1
   CALL MULTIG(TSUB1)
   IF(NEXT.FO.1) CALL MULTIG(TSUB2)
   ICOL=ICOL+1
   GO TO 1
C***** FOUND = INCREMENT PAST = AND SEARCH FOR NAME
4  ICOL=ICOL+1
   GO TO 3
C***** FOUND $ IF NCR=0 THIS IS A COMMENT - READ NEXT CARD
C***** OTHERWISE IT IS A NAME$
5  IF(NCAR.NE.0) GO TO 3
   CALL NUGET
   GO TO 1
C* * * * *
C* SEARCH TABLE OF NAMES. IF NAME NOT FOUND GO TO 15
C* * * * *
9  IT=2
   GO TO 11
C***** INCREMENT TO NEXT NAME IN /NLIST/
10 IT=IT+NAMES(IT)+2
C***** LOOK FOR END-OF-/NLIST/
11 IF(NAMES(IT).EQ.0) GO TO 15
C***** SEE IF NAME HAS SAME NUMBER OF CHARACTERS
   IF(NAMES(IT).NE.NCAR) GO TO 10
   ITC=IT
   DO 13 I=1,NCAR
     ITC=ITC+1
C***** SEE IF EACH CHARACTER MATCHES
   IF(NAMES(ITC).NE.ICAR(I)) GO TO 10
13 CONTINUE
C* * * * *
C* FOUND THE NAME - SET NAMEIX, CHECK FOR END-OF-CARD AND RETURN
C* * * * *
   NAMEIX=NAMES(IT-1)
   IF(ICOL.GT.400) CALL NUGET
   RETURN
C* * * * *
C* NAME NOT FOUND - ERROR, IF BOTH CLASSES OF NAMES HAVE BEEN CHECKED
C* * * * *
15 IF(IPASS.EQ.2) GO TO 100

```

```

C* * * * *
C PASS=2 REGIMS BY PERMIGC EMPHEDD OIGITS
C C101= REGIMS C= WITH IDES=101.
C* * * * *
C IPASS=2
C IDFS=0
C ITC=VNCAR
C NCAR=0
C IT=0
20 IT=IT+1
IF(IT.GT.ITC ) GO TO 25
DO 21 I=1,10
IF(ICARD(IT).NE.MATCH(I)) GO TO 21
IDES=IDES+10+I-1
GO TO 20
21 CONTINUE
NCAR=NCAR+1
ICARD(NCAR)=ICARD(IT)
GO TO 20
C* * * * *
C MAKE SECOND PASS IF DIGITS WERE RECEIVED ( NAME MUST BE 4 CHARACTER )
C* * * * *
25 IF(NCAR.LT.ITC) GO TO 4
C* * * * *
C INITIAT EXECUTION - COULD NOT IDENTIFY NAME
C* * * * *
100 INITIAT=INITIAT+1
WRITE(IPRINT,1001)
1001 FORMAT(42H PROGRAM COULD NOT IDENTIFY VARIABLE NAME. )
1000 FORMAT(1PRINT,1000) ICOL,NCARD
* 47H ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED)
C* * * * *
C SKIP NUMERIC FIELDS UNTIL *COMMENT OR NEXT NAME FOUND
C* * * * *
101 IF(IPRD.GT.1) IPRD=1
CALL NUMBER
GO TO (101,101, 50,50),TEXT
END

```

```

SUBROUTINE AL_VECT(NUM,IX)
COMMON/CARDAT / NCARD,ICARD,ICARD,INHIRT,NCODE,
*           NHUCM,NEXT,IREP,INTPRT,FRCPRT,LDG,LEXPRT,
*           ICOL,ICOL,VCOL,ICARD(RD)
*
DIMENSION NUM(1)
C* * * * *
C* CONTINUE READING AND COUNTING ELEMENTS AS LONG AS NEXT=1
C* THE VECTOR WILL END WITH A / (NEXT=2), $ (NEXT=3) OR NAME (NEXT=4)
C* * * * *
1 CALL NLINTG(NUM(IX))
  IX=IX+1
  GO TO (1,2,2,2),NEXT
C***** IX IS THE NUMBER OF ELEMENTS IN THE VECTOR
2 IX=IX-1
  RETURN
  END

```

```

SUBROUTINE NILVECR(FNUM,IX)
COMMON/CARDAT/ NCARD,LECHO,INHIT,NCODE,
*          NILCNT,NEXT,IRP,INTPT,FRPRT,ILUG,LEXPRT,
*          ICOL,LCOL,MCOL,ICARD(80)
C          DIMENSION FNUM(1)
C          * * * * *
C          CONTINUE READING AND COUNTING ELEMENTS AS LONG AS NEXT=1
C          THE VECTOR WILL END WITH A / (NEXT=2), $ (NEXT=3) OR NAME (NEXT=4)
C          * * * * *
C          1 CALL NILREAL(FNUM(IX))
            IX=IX+1
            GO TO (1,2,2,2),NEXT
C          ***** IX IS THE NUMBER OF ELEMENTS IN THE VECTOR
            2 IX=IX-1
              RETURN
            END

```



```

SUBROUTINE NI_REAL(FNUM)
COMMON/CARDAT/ NCARD,IECHI,INHIT,MCODE,
*           MLCNT,NEXT,IRP,INTPT,FRCPT,ILOG,IFXPRT,
*           ICOL,ICOL,MCOL,ICARD(RN)
C* * * * *
C* GET THE NEXT NUMBER. IF THE FIELD IS NULL
C (NO DIGITS) DO NOT CHANGE THE CURRENT VALUE OF FNUM
C* * * * *
CALL MINIMA
IF(NMLCNT.EQ.0) RETURN
C* * * * *
C ADD INTEGER PART TO FRACTION (BOTH HAVE SAME SIGN - SEE NMINIM)
C TAKE LOG BASE 10, FXPONENT TO BASE 10, OR JUST RETURN FNUM.
C* * * * *
FNUM=INTPT
FNUM=FNUM+FRCPT
IF(ILOG.EQ.1) GO TO 1
IF(IFXPRT.EQ.0) RETURN
FNUM=FNUM*10.0#IFXPRT
RETURN
1 R=IFXPRT
FNUM=R+ALOG10(FNUM)
RETURN
END

```



```

NH(IPART)=NH(IPART)+10+I-1
NC(IPART)=NC(IPART)+1
6 ICOL=ICOL+1
  IF(ICOL.GT.LCOL) GO TO 20
  ISYM=ICARD(ICOL)
  IF(ISYM.EQ.KHL) GO TO 6
  GO TO 4
5 CONTINUE
C***** LINK FOR *. . . I. OR -
  IF(ISYM.EQ.KAS) GO TO 10
  IF(ISYM.EQ.KPR) GO TO 11
  IF(ISYM.EQ.KL) GO TO 12
  IF(ISYM.EQ.KF) GO TO 13
  IF(ISYM.EQ.KMI) GO TO 14
C***** LINK FOR *. . . / . < OR NEXT INPUT NAME
  NEXTI=4
  IF(ISYM.EQ.KCM) NEXTI=1
  IF(ISYM.EQ.KSL) NEXTI=2
  IF(ISYM.EQ.KDI) NEXTI=3
  GO TO (22,22,30,25).NEXTI
C***** FMINI *. . . SET REPEAT OPTION
  10 IF(NH(1).NE.0) IREP=NH(1)
  NH(1)=0
  NC(1)=0
  GO TO 2
C***** FMINI *. . . BEGIN FRACTION PART
  11 IPART=2
  GO TO 2
C***** FMINI I. SET LOG10 OPTION
  12 ILOG=1
C***** FMINI F OR I. BEGIN EXPONENT PART
  13 IPART=3
C***** SAVE CURRENT COLUMN INDEX - AN INPUT NAME MAY BEGIN WITH F OR I.
  KCOL=ICOL
  GO TO 2
C***** FMINI - SIGN
  14 NS(IPART)=1
  GO TO 2
C * * * * *
C T-15 SECTION DE CHER DETERMINEZ HOW SUB-EPIC FIELD TERMINATES
C TO DEFINE VARIABLE NEXT

```

```

C NEXT=1 TWO NUMBERS SEPARATED BY COMMA(,) OR END-OF-CARD
C NEXT=2 TWO NUMBERS SEPARATED BY SLASH (/)
C NEXT=3 NUMBER FOLLOWED BY $COMMENT
C NEXT=4 NUMBER FOLLOWED BY INPUT NAME OR SPECIAL SYMBOL
C* * * * *
18 IF(NEXTT.NE.0) GO TO 30
NEXTT=1
GO TO 22
19 IF(NEXTT.NE.0) GO TO 30
NEXTT=2
GO TO 22
20 NEXTT=0
21 IF(ICOL.GT.MCOL) CALL MUGET
ISYM=ICARD(ICOL)
IF(ISYM.NE.KRL) GO TO 23
22 ICOL=ICOL+1
GO TO 21
23 IF(ISYM.EQ.KMI) GO TO 30
IF(ISYM.EQ.KPL) GO TO 30
IF(ISYM.EQ.KPR) GO TO 30
IF(ISYM.EQ.KCM) GO TO 18
IF(ISYM.EQ.KSL) GO TO 19
DO 24 I=1,10
IF(ISYM.EQ.MATCH(I)) GO TO 30
24 CONTINUE
NEXTT=4
IF(ISYM.EQ.KOL) NEXTT=3
GO TO 30
C* * * * *
FOUND NEXT INPUT NAME. IF IT BEGINS WITH E OR I, IT MAY
HAVE ERRONEOUSLY BEEN INTERPRETED AS AN EXPONENT.
C* * * * *
25 IF(IPART.NE.3) GO TO 30
IF(NC(3).NE.0) GO TO 30
ICOL=KCOL
ILOG=0
C* * * * *
INTEGER PART AND FRACTION PART MUST HAVE SAME SIGN.
C FLOAT FRACTION PART AND SET EXPONENT SIGN. THEN ADD NUMBER
C OF DIGITS IN INTEGER AND FRACTION (MULTIPLIED IF FIELD IS BLANK).
C* * * * *

```

```

30  INT PRT=NS(1)*MII(1)
    FRC PRT=NS(1)*MII(2)
    FRC PRT=FRC PRT/(10.0**NC(2))
    IFXPRT=NS(3)*MII(3)
    NIILCNT=NC(1)+NC(2)
C* * * * *
C. REPEAT OPTION - SFF IF THIS IS LAST VALUE TO BE RETURNED
C* * * * *
    100  IREP=IREP-1
        NEXT=NCXT
        IF(IREP.NE.0) NEXT=1
        RETURN
    END

```



```

C   ARE STORED IN INSC(1,IR) THROUGH INSC(1,IFE).
C   * * * * *
3  IR=INSLC(1,M)
   IS=INSLC(2,N)+1
   IF=IR+IS
   IFF=IF-1
C   * * * * *
C   REMOVE PREVIOUS REFERENCES TO THOSE ENVIRONMENTS WHICH MAY NO
C   LONGER BE REQUIRED. (MODE IS USED BY SUBROUTINE SELECT)
C   * * * * *
   DO 104 I=IR,IFE
   INP=INSC(1,I)
   GO TO (104,104,103,104,104,104,103,104,104,103,104,104,103,104,
      * 104),INP
103 INP=2*BYM+INSC(3,I)
   MODE(INP)=MODE(INP)-1
104 CONTINUE
C   * * * * *
C   NOW REMOVE THE OLD TRANSFER FUNCTION AND RELOCATE OTHER FUNCTIONS.
C   THIS MAKES STORAGE AVAILABLE AT THE END OF INSC AND COEF FOR
C   NEW FUNCTIONS.
C   * * * * *
   DO 4 I=IF,NCINS
   INSC(1,IR)=INSC(1,I)
   INSC(2,IR)=INSC(2,I)
   INSC(3,IR)=INSC(3,I)
   INSC(4,IR)=INSC(4,I)
   COEF(1,IR)=COEF(1,I)
   COEF(2,IR)=COEF(2,I)
   COEF(3,IR)=COEF(3,I)
   IR=IR+1
4  CONTINUE
   NCINS=NCINS-IS
   IR=INSLC(1,M)
   DO 5 I=1,NCR
   IF(INSLC(1,I).GT,IR) INSLC(1,I)=INSLC(1,I)-IS
5  CONTINUE
C   * * * * *
C   BEGIN STORING NEW COMPONENT.
C   INSLC IS STARTING ADDRESS FOR NEW TRANSFER FUNCTION
C   IADR=0 INDICATES THAT NO FRAGILITY POINTS HAVE UNCERTAINTY

```

```

C * * * * * IF $ IS EQUAL, GO TO 10) WHICH WILL DELETE THE COMPONENT * * * * *
C * * * * * 6 INSLDG(I,N)=MCMST+1 * * * * *
C * * * * * IRD(R,N)=0 * * * * *
C * * * * * CALL NUNFG(IS) * * * * *
C * * * * * GO TO (7,7,10,7).NEXT * * * * *
C * * * * * NOTE THAT THE LOGIO OPTION IS TURNED OFF BUT IS TURNED ON IF * * * * *
C * * * * * FIRST FRAGILITY PARAMETER CONTAINS A 1. (E.G., 5L0) * * * * *
C * * * * * ENTER FRAGILITY CURVE - * * * * *
C * * * * * FCY PROBABILITY FOR ORIGINATE * * * * *
C * * * * * FCX BEST ESTIMATE FOR NO UNCERTAINTY * * * * *
C * * * * * FC BEST ESTIMATE WITH UNCERTAINTY * * * * *
C * * * * * FU WIDTH OF UNCERTAINTY * * * * *
C * * * * * IRD DISTRIBUTION SHAPE FOR UNCERTAINTY * * * * *
C * * * * * IRD(7) NUMBER OF FRAGILITY POINTS (1 THROUGH 6) * * * * *
C * * * * * 7 CONTINUE * * * * *
C * * * * * LLOG=0 * * * * *
C * * * * * I=0 * * * * *
C * * * * * 10 I=I+1 * * * * *
C * * * * * CALL NLREAL(FCY(I,N)) * * * * *
C * * * * * CALL NLREAL(FI) * * * * *
C * * * * * IF(ILOG.EQ.1) LLOG=1 * * * * *
C * * * * * GO TO (11,12,12,12).NEXT * * * * *
C * * * * * 11 CALL NLREAL(FR) * * * * *
C * * * * * CALL NLREAL(FHP) * * * * *
C * * * * * FU(I,N)=(FR-FI)/FU(I,N)*10.0+0.5 * * * * *
C * * * * * IRD(I,N)=FR * * * * *
C * * * * * FC(I,N)=FR * * * * *
C * * * * * IF(IRD(I,N).EQ.0) IRD(I,N)=1 * * * * *
C * * * * * IF(IRD(I,N).EQ.10) IRD(I,N)=9 * * * * *
C * * * * * IRD(R,N)=1 * * * * *
C * * * * * GO TO (10,10,13,13).NEXT * * * * *
C * * * * * 12 IRD(I,N)=0 * * * * *
C * * * * * FC(I,N)=FI * * * * *
C * * * * * FCX(I,N)=FI * * * * *
C * * * * * GO TO (10,10,13,13).NEXT * * * * *
C * * * * * 13 IRD(7,N)=1 * * * * *
C * * * * * ENTER A TRANSFER FUNCTION INSTRUCTION (LOOK FOR R, CAN. * * * * *

```

```

C * * * * * OR USER - EACH INSTRUCTION IS ENCLOSED WITH SLASHES (/) * * * * *
C * * * * * 20 NCINS=NCINS+1
    IF(NCINS.GT.MCINS) GO TO 99R
    IF(ICARD(ICOL).EQ.KR) GO TO 23
    IF(ICARD(ICOL).EQ.KC) GO TO 40
    IF(ICARD(ICOL).EQ.KH) GO TO 50
    WRITE(IPRINT,1001)
1001 FORMAT(39H COULD NOT RECOGNIZE LOCAL ENV RELATION)
    GO TO 999
998 WRITE(IPRINT,1002)
1002 FORMAT(42H TOO MANY ENV RELATIONSHIPS FOR INSC TABLE)
    NCINS=NCINS-1
999 WRITE(IPRINT,1000) ICOL,NCARD
1000 FORMAT(28H FATAL ERROR PRECEDES COLUMN.13.5H CARD.13/
    * 47H ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED)
    INHIRT=INHIRT+1
21 IF(ICARD(ICOL).EQ.KDL) GO TO 100
    ICOL=ICOL+1
    IF(ICOL.GT.MCOL) CALL MLGET
    GO TO 21
C * * * * * FOUND R - TRANSFER FUNCTION INSTRUCTION * * * * *
C * * * * * 23 TF=1.0 * * * * *
    IF(LLOG.EQ.1) TF=0.0
    C=TF
    TFU=TF
    ICOL=ICOL+1
C***** GET SUBSCRIPT FOR FIRST R
    CALL MLINTG(INSC(2,MCINS))
    ICOL=ICOL+1
C***** LOOK FOR SECOND R (E.G., R2=R1) OR ENVIRONMENT REFERENCE (R1=5)
    CALL MLINTG(15)
    IF(MLCONT.NE.0) GO TO 24
    ICOL=ICOL+1
C***** FOUND R - GET SUBSCRIPT FOR SECOND R
    CALL MLINTG(15)
    INP=1
    GO TO 25
C***** FOUND ENV. - REQUEST SELECT TO STORE ENV. FOR MINUTE CARD

```

```

C***** PROCESS, UNLESS IS=0 (OVERPRESSURE ENVIRONMENT)
24 IOP=2
   IF(IS.=0.0) GO TO 25
   IOP=4
   IF=2*MYM+IS
   MODE(TF)=MODE(TF)+1
C***** SET OPERATION FLAG (IOP) DEPENDING ON WHICH PARAMETERS (TF,TFII,
C***** C) ARE FOUND. IOP IS MODIFIED FOR LOGIO OPERATIONS.
25 IF(FLAG.EQ.1) IOP=IOP+3
   INSC(3,NCINS)=IS
   GO TO (26,27,27,27).NEXT
26 CALL MLREAL(TF)
   GO TO (28,27,27,27).NEXT
27 INSC(1,NCINS)=IOP
   COEF(3,NCINS)=TF
   GO TO (20,20,100,20).NEXT
28 CALL MLREAL(TFII)
   GO TO (29,30,30,30).NEXT
29 CALL MLREAL(C)
30 IF(FLAG.EQ.1) GO TO 31
   IF(ABS(TFII-1.0).GT.0.00005) IOP=IOP+4
   GO TO 32
31 IF(TFII.EQ.0.0) IOP=IOP+6
32 INSC(1,NCINS)=IOP
C***** SET UP TRANSFER FUNCTION COEFFICIENTS DEPENDING ON IOP.
   GO TO (33,33,33,34,34,34,35,35,36,36,36).IOP
33 COEF(3,NCINS)=TF#C
   GO TO (20,20,100,20).NEXT
34 COEF(3,NCINS)=TF+C
   GO TO (20,20,100,20).NEXT
35 COEF(1,NCINS)=C*TF/TFII
   COEF(2,NCINS)=C*TF*TFII-COEF(1,NCINS)
   INSC(4,NCINS)=10.0/(1.0+TFII)+0.5
   GO TO (20,20,100,20).NEXT
36 COEF(2,NCINS)=SORT((TF+TFII)/(TF-TFII))
   COEF(1,NCINS)=C+TF-COEF(2,NCINS)
   COEF(2,NCINS)=2.0*COEF(2,NCINS)
   INSC(4,NCINS)=5
   GO TO (20,20,100,20).NEXT
C* * * * *
C   FOUND CAN - SET FOR LOGIO OR NON-LOG ADDITION. ENTER NUMBER * * * * *

```



```

NAME(I)=NAME(IR)
INSLC(1,I)=INSLC(1,IR)
INSLC(2,I)=INSLC(2,IR)
IFE=TRD(7,IR)
TRD(7,I)=IFE
TRD(R,I)=TRD(R,IR)
DO 109 IF=1,IFE
FCY(IE,I)=FCY(IE,IR)
FCX(IE,I)=FCX(IE,IR)
FC(IE,I)=FC(IE,IR)
FU(IE,I)=FU(IE,IR)
TRD(IE,I)=TRD(IE,IR)
109 CONTINUE
110 CONTINUE
C* * * * *
C IF THERE ARE ANY NETWORK FOUNDATIONS, SCAN FOR REFERENCES TO THE
C DELETED COMPONENT. REPLACE THE REFERENCE WITH
C -(IDES#10+1) WHERE IDES IS THE DELETED COMPONENT NAME.
C SUBROUTINE NETCK USES THIS INFORMATION. IF NETWORK IS NOT
C REPLACED OR DELETED, NETCK WILL IDENTIFY THIS VALUE AS AN ERROR.
C* * * * *
IF(NSYS,EO,0) RETURN
NCRP1=NCR+1
IDES=-(IDES#10+1)
IXSYS=NAME-NSYS+1
DO 130 I=IXSYS,MMAME
IL=INSLC(1,I)
LI=IL+INSLC(2,I)
DO 120 L=IL,LI
IR1=INSNET(1,L)
IF(IR1.GT.NCRP1) GO TO 115
IF(IR1-N) 115,113,114
113 INSNET(1,L)=IDES
GO TO 115
114 INSNET(1,L)=IR1-1
115 IR1=INSNET(3,L)
IF(IR1.GT.NCRP1) GO TO 120
IF(IR1-N) 120,118,119
118 INSNET(3,L)=IDES
GO TO 120
119 INSNET(3,L)=IR1-1

```

120 CONTINUE  
130 CONTINUE  
RETURN  
END

```

SURROUTINE RNDL=(ID=5)
COMMON/WORKEY/ IKEY,LKEY,MKEY,IPAD,KEY(1)
COMMON/IDUNIT/ IREAD,IPRINT,IPUNCH,ITAPE
COMMON/CARDAT/ NCARD,IFCHD,IMHRT,NCODE,
* MULT,NT,NEXT,IREP,INTRT,EXCPRF,LOG,TEXPRF,
* ICOL,ICOL,MCOL,ICARD(RO)
COMMON /CHRLIST/ KA,KR,KC,KD,KE,KF,KG,KH,KI,KJ,
* KK,KL,KM,KN,KO,KP,KQ,KR,KS,KT,
* KV,KW,KX,KY,KZ,K0,K1,K2,K3,
* K4,K5,K6,K7,K8,K9,KCM,KPL,KMI,KAS,
* KSL,KPR,KLP,KRP,KEO,KDL,KRL
DIMENSION MATCH(10)
EQUIVALENCE (MATCH,K0)
COMMON/CNC1 / NCR,NSYS,MNAME,NAME(1)
COMMON/CNC2 / INSLC(2,1)
COMMON/CNO1 / NNINS,MNINS,INSNET(4,1)
C* * * * *
C COMPONENT NAMES ARE STORED BEGINNING IN NAME(1) TO NAME(NCR), BUT
C SS AND SYS NAMES ARE STORED BEGINNING IN NAME(1) TO NAME(NCR), BUT
C WORD NAME(MNAME)
C IF THIS IS THE FIRST NETWORK GO TO 2
C IF NOT, SEE IF THIS NETWORK WAS READ PREVIOUSLY
C* * * * *
IE=MNAME-NSYS+1
IF(NSYS.EQ.0) GO TO 2
DO 1 I=1,MNAME
IF(IDES.NE.NAME(I)) GO TO 1
N=I
GO TO 3
1 CONTINUE
C* * * * *
C PREPARE TO ENTER NEW NETWORK
C* * * * *
? IE=IE-1
NSYS=NSYS+1
N=IF
NAME(N)=IDES
GO TO 6
C* * * * *
C NETWORK WAS READ BEFORE SO THIS IS A REPLACEMENT OR DELETION.
C REMOVE OLD NETWORK TO MAKE SPACE AVAILABLE FOR NEW EQUATIONS.

```

```

C* * * * *
3 IR=INSLC(1,N)
  IS=INSLC(2,N)+1
  IF=IR+IS
  DO 4 I=IF,NNINS
    INSNFT(1,IR)=INSNFT(1,I)
    INSNFT(2,IR)=INSNFT(2,I)
    INSNFT(3,IR)=INSNFT(3,I)
    INSNFT(4,IR)=INSNFT(4,I)
  IR=IR+1
4 CONTINUE
  NNINS=NNINS-IS
  IR=INSLC(1,N)
  IF=MNAME-MSYS+1
  DO 5 I=IF,MNAME
    IF(INSLC(1,I).GT.1R) INSLC(1,I)=INSLC(1,I)-IS
5 CONTINUE
C* * * * *
C INITIALIZE KEY VECTOR AND BEGIN MOVING SYMBOLS LIKE (.).+. *
C AND $ FROM THE CARD TO KEY. EACH TIME C, SS OR SYS NAME IS FOUND
C ON THE CARD, LOCATE THE NAME IN THE NAME VECTOR OF COMMON /CNCL/.
C STORE THE LOCATION IN KEY INSTEAD OF THE NAME.
C* * * * *
6 INSLC(1,N)=NNINS+1
  LKEY=1
  KEY(LKEY)=KLP
  ISYS=1
  IRAL=0
  GO TO 11
10 ICOL=ICOL+1
11 IF(ICOL.GT.MCOL) CALL NUGET
  ISYM=ICARD(ICOL)
12 IF(ISYM.EQ.KRL) GO TO 10
  IF(ISYM.EQ.KC) GO TO 10
  IF(ISYM.EQ.KPL) GO TO 40
  IF(ISYM.EQ.KAS) GO TO 40
  IF(ISYM.EQ.KOL) GO TO 40
  IF(ISYM.EQ.KLP) GO TO 30
  IF(ISYM.EQ.KRP) GO TO 31
  IF(ISYM.EQ.KS ) GO TO 25
C***** CONVERT DIGITS OF NAME TO INTEGER

```

```

NAD=0
13 DO 14 I=1,10
  IF (ISYM.NE.MATCH(I)) GO TO 14
  NAD=NAD+10+I-1
  ICOL=ICOL+1
  IF (ICOL.GT.MCOL) CALL MUGET
  ISYM=ICARD(ICOL)
  GO TO 13
14 CONTINUE
  GO TO (15,17,18),ISYS
C***** LOCATE COMPONENT NAME
15 DO 16 I=1,NCR
  IF (NAD.NE.NAME(I)) GO TO 16
  NAD=I
  GO TO 20
16 CONTINUE
  GO TO 99R
C***** SYS NAMES STORED AS NEGATIVE NUMBERS
17 NAD=-NAD
C***** LOCATE SS OR SYS NAME
18 DO 19 I=1,MSNAME
  IF (NAD.NE.NAME(I)) GO TO 19
  NAD=I
  ISYS=I
  GO TO 20
19 CONTINUE
  GO TO 99R
C***** STORE LOCATION OF NAME IN KEY
20 LKEY=LKEY+1
  IF (LKEY.GT.MKEY) GO TO 997
  KEY(LKEY)=NAD
  GO TO 12
C***** FOUND S - SET ISYS=3 FOR SS AND ISYS=2 FOR SYS
25 ISYS=3
26 ICOL=ICOL+1
  IF (ICOL.GT.MCOL) CALL MUGET
  ISYM=ICARD(ICOL)
  IF (ISYM.EQ.KRI) GO TO 26
  IF (ISYM.EQ.KSI) GO TO 26
  IF (ISYM.NE.KY) GO TO 27
  ISYS=2

```

```

GO TO 26
27 NAD=0
GO TO 13
30 IHAL=IHAI+1
GO TO 40
31 IHAL=IHAI-1
C***** STORE SYMBOL (.).+. * OR $ IN KEY
40 LKEY=LKEY+1
IF(LKEY.GT.MKEY) GO TO 997
KEY(LKEY)=ISYM
C***** DONE IF $ FOUND AND STORED
IF(ISYM.EQ.KOL) GO TO 100
GO TO 10
C* * * * *
C CHECK TO SEE IF PARENTHESES ARE BALANCED
C IF LKEY=?. THIS IS SS OR SYS DELETION. IF NOT, CALL LOGIC THEN
C STORE NUMBER OF INSTRUCTIONS GENERATED BY LOGIC AND RESET LAST
C INSTRUCTION TO STORE PROBABILITY (DURING MONTE CARLO PROCESS) IN
C PROPER LOCATION OF P VECTOR. NOTE THAT NAME(N) CONTAINS THE C.
C SS OR SYS NAME AND P(N) WILL CONTAIN THE PROBABILITY.
C* * * * *
100 IF(IHAL.NE.0) GO TO 996
IF(LKEY.EQ.?) GO TO 200
CALL LOGIC
INSLOC(2,N)=NNINS-INSLOC(1,N)
INSNET(4,NNINS)=N
RETURN
C* * * * *
C DELETE SS OR SYS EQUATION
C* * * * *
200 I=MNAME-MSYS+1
IF=N-1
DO 201 I=IR,IE
NAME(N)=NAME(N-1)
INSLOC(1,N)=INSLOC(1,N-1)
INSLOC(2,N)=INSLOC(2,N-1)
N=N-1
201 CONTINUE
C* * * * *
C SCAN OTHER EQUATIONS FOR REFERENCES TO DELETED SS OR SYS. NOTE
C REPLACE REFERENCE (FOR NETCK) WITH MODIFIED NAME. IDES.

```

```

C      THAT I=1 WAS USED FOR COMPONENTS, I=2 FOR SS AND I=3 FOR SYS
C      IN EQUATION      IDES=-(IDES*(I+1))
C* * * * *
      I=2
      IF (IDES.LT.0) I=3
      IDES=-(IARS(IDES)*(I+1))
      IXSYS=IR+1
      IE=IF+1
      DO 300 I=IXSYS,IF
      IL=INSLC(1,I)
      LL=IL+INSLC(2,I)
      DO 250 L=IL,LL
      IRI=INSNET(1,L)
      IF (IRI.LT.IR) GO TO 210
      IF (IRI-IF) 209,20R,210
      20R INSNET(1,L)=IDES
      GO TO 210
      209 INSNET(1,L)=IRI+1
      210 IRI=INSNET(3,L)
      IF (IRI.LT.IR) GO TO 220
      IF (IRI-IF) 219,21R,220
      21R INSNET(3,L)=IDES
      GO TO 220
      219 INSNET(3,L)=IRI+1
      220 IF (L.NE.LL) GO TO 250
      IRI=INSNET(4,L)
      IF (IRI.LT.IR) GO TO 250
      IF (IRI-IF) 229,22R,250
      22R INSNET(4,L)=IDES
      GO TO 250
      229 INSNET(4,L)=IRI+1
      250 CONTINUE
      300 CONTINUE
      NSYS=NSYS-1
      RETURN
      996 WRITE (IPRINT,1001)
      1001 FORMAT(27H PARENTHESIS DO NOT BALANCE)
      WRITE (IPRINT,1000) ICDL,NCARD
      INHRT=INHRT+1
      RETURN
      997 WRITE (IPRINT,1002)

```

```

1002 FORMAT(46H EQUATION TOO LONG FOR KEY - DIVIDE EQUATION)
      GO TO 999
958  WRITE(1PRINT,1003)
1003 FORMAT(31H NAME NOT FOUND OR SYMBOL ERROR)
999  WRITE(1PRINT,1000) (COL,NCARD)
1000 FORMAT(28H FATAL ERROR PREFIXES COLUMN,13.5H CARD,13/
      * 47H ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED)
      INHTRT=INHTRT+1
1010 IF(CICARD(COL).EQ.KOL) GO TO 1020
      ICOL=ICOL+1
      IF(ICOL.GT.MCOL) CALL MUGET
      GO TO 1010
1020 CALL MUGET
      RET:PN
      END

```

```

SUBROUTINE LOGIC
COMMON/MRKEY/ TRKY,LKEY,MKEY,TPAD1,KEY(1)
COMMON/TPRINT/ TREAD,TPRINT,TPUNCH,TTAPE
COMMON/CARDAT/ NCARD,ICARD,INHIST,NCODE,
MULTCAT,NEXT,TRFP,INTPRT,FRCPRT,ILNG,LEXPRT,
*
* ICOL,LCOL,MCOL,ICARD(R0)
COMMON /CHRIST/ KA,KR,KC,KD,KF,KG,KH,KI,KJ,
* KK,KL,KN,KO,KP,KQ,KR,KS,KT,
* KU,KV,KW,KX,KY,KZ,K0,K1,K2,K3,
* K4,K5,K6,K7,K8,K9,KCM,KDI,KMT,KAS,
* KSI,KPR,KLP,KRP,KFO,KDL,KRL
COMMON/CNO1 / NNINS,MNINS,INSNET(4,1)
COMMON/CNC1 / NCR,NCYS,MNAME,NAME(1)
COMMON/CNC3 / MPS,TPAD2,P(1)
C * * * * *
C LOGIC SCANS THE EQUATION STORED IN KEY TO RESOLVE PARENTHESES
C AND FORMS A LOGIC SEQUENCE FROM THREE TYPES OF INSTRUCTIONS.
C
C -RULE-- -PRORABILITY- -INSTRUCTION (ADDR IS ADDRESS)-
C C = A + R C=(1-A)(1-R) ADDR A, IOP=1, ADDR R, STORF IN ADDR C
C C = A * R C=(A)(R) ADDR A, IOP=2, ADDR R, STORF IN ADDR C
C C = A C=(A) ADDR A, IOP=3,-----, STORF IN ADDR C
C
C THE ADDRESSES ARE TO THE P VECTOR. P IS DIMENSIONED LONGER
C THAN NAME SO THAT THE EXCESS CELLS CAN BE USED AS SCRATCH PAD.
C * * * * *
L=3
LL=?
MR=MNAME+1
LOC=MR
LOCN=LOC+1
IF(KEY(L).NE.KDI) GO TO 10
NNINS=NNINS+1
IF(NNINS.GT.MNINS) GO TO 999
INSNET(1,NNINS)=KEY(L)
INSNET(2,NNINS)=3
INSNET(3,NNINS)=0
INSNET(4,NNINS)=LOC
RETURN
C * * * * * CHECK FOR *
10 IF(KEY(L).NE.KAS) GO TO 20
C * * * * * FOUND * - MAKE SURE ADDRESSES COMBINED NOT A*(. ) * A OR ) * (

```

```

IF(KEY(LL),FO,KRP,OR,KEY(L+1),FO,KLP) GO TO 50
INP=2
GO TO 30
C***** CHECK FOR +
20 IF(KEY(L),NE,KPL) GO TO 55
C***** FOUND + - MAKE SURE ADDRESSES COMBINED NOT A+(. )+A OR )+(
IF(KEY(LL),FO,KRP,OR,KEY(L+1),FO,KLP) GO TO 50
C***** ALSO CHECK FOR *A+R, A+R*, *A+R* (* HAS PRIORITY OVER +)
IF(KEY(LL-1),FO,KAS,OR,KEY(L+2),FO,KAS) GO TO 50
INP=1
C***** PREPARE TO BUILD AND STORE INSTRUCTION
30 NNINS=NNINS+1
IF(NNINS.GT,MMINS) GO TO 99R
LOCA=KEY(LL)
LOCR=KEY(L+1)
INSNET(1,NNINS)=LOCA
INSNET(2,NNINS)=LOCR
INSNET(3,NNINS)=LOCR
INSNET(4,NNINS)=LOCR
C***** CHECK FOR AVAILABLE CELLS - DONT USE P(1) THROUGH P(MVAME)
IF(LOCA,LT,MR) LOCA=LOCN
IF(LOCR,LT,MR) LOCR=LOCN
LOC=LOCA
IF(LOCA,LE,LOCR) GO TO 35
LOCR=LOCR
LOCA=LOCA
35 IF(LOC,EO,LOCN) GO TO 36
IF(LOCR,EO,LOCN-1) LOCN=LOCR
GO TO 40
36 LOCN=LOCN+1
IF(LOCN,LE,MPS) GO TO 40
WRITE(I,PRINT,1001)
1001 FORMAT(47H DIMENSION PS ARRAY LARGER FOR 3001EAF SOLUTION )
GO TO 999
99R WRITE(I,PRINT,1002)
1002 FORMAT(33H TOO MANY INSTRUCTIONS FOR IMSN=T)
999 WRITE(I,PRINT,1000) LOCALCARD
1000 FORMAT(28H FATAL ERROR PRECEDES COLUMN.13.5H CARD.13/
* 47H ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED)
INHIRT=INHIRT+1
RETURN

```

```

C***** CHECK TO SEE IF ( ) CAN BE REMOVED
40 CONTINUE
  L=L+2
  IF(KEY(LL-1).EQ.KLP.AND.KEY(L).EQ.KRP) GO TO 41
  GO TO 42
41 LL=LL-1
  L=L+1
C***** CHECK TO SEE IF KEY IS EMPTY
42 IF(LL.LE.2.AND.KEY(L).FO.KDL ) RETURN
  KEY(LL)=INSNET(4,NMINS)
  GO TO 10
C* * * * *
C  MOVE INFORMATION TOWARD BEGINNING OF EQUATION WHEN THE INFORMATION
C  CANT BE USED TO BUILD AN INSTRUCTION
C* * * * *
50 LL=LL+2
  L=L+2
  KEY(LL-1)=KEY(L-2)
  KEY(LL)=KEY(L-1)
  IF(KEY(L).NE.KLP) GO TO 10
55 IF(KEY(L).FO.KDL ) GO TO 101
  LL=LL+1
  KEY(LL)=KEY(L)
  L=L+1
  GO TO 10
C* * * * *
C  REACHED END-OF-EQUATION. BEGIN NEXT PASS - SOME TERMS NOT RESOLVED
C* * * * *
101 KEY(LL+1)=KDL
  LL=2
  L=3
  GO TO 10
END

```



```

G1=GAMMA(I)
G2=G1
HHO=HO/WO*(1./3.)
HHR=HHO
HH=H/W*(1./3.)
IF(HR(I).EQ.-1.) GO TO 14
HHR=HR(I)
IF(HHO.GT.HHR) G1=GAMMA(I)
IF(HH.GT.HHR) G2=GAMMA(I)
14 G=EXP(-G1*(HHR-HHO)-G2*(HH-HHR))
SCALE=(W/WO)*ALPHA(I)*G
GO TO (1,2,3,1,2,3),IXMODE
1 ILL(NRYN)=ILLN(IN)+ALOG(SCALE)
GO TO 10
2 ILL(NRYN)=ILLN(IN)*SCALE
GO TO 10
3 ILL(NRYN)=ILLN(IN)+ALOG10(SCALE)
10 CONTINUE
C* * * * *
C* SELECT MATRIX ELEMENTS FOR REQUIRED ENVIRONMENTS
C* * * * *
L=0
NRYM1=NRYM-1
DO 20 I=1,NRYM
J=KEY(I)-1
J=J*NRYM-(J*(I-1))/2
DO 20 II=1,NRYM
K=J+KEY(II)
L=L+1
RHOCOL(L)=RHOCOS(K)
SIGSS(L)=SIGSS(K)
20 CONTINUE
WWR=W*1000.
IF(NRP.NF.0) GO TO 25
C* * * * *
C* CONVERT RANGES TO CORRESPONDING PRESSURES
C* * * * *
DO 30 I=1,NR
CALL RTOP(I)*5.0751*OP(I)*WWR.H
IF(OP(I).LT.100.) WRITE(IPRINT,2000)
2000 FORMAT(37H0-DARNING - PRESSURE LESS THAN 100 PSI )

```

Reproduced from  
best available copy.

```

30 CONTINUE
RETURN
C * * * * *
C * CONVERT PRESSURES TO CORRESPONDING RANGES
C * * * * *
35 DO 36 I=1,NR
CALL PTRR(R(I),OP(I),JWR,H)
IF(P(I).EQ.0.0) INHIRT=INHIRT+1
36 R(I)=R(I)/6.0761
RETURN
900 IN=IN-MRYM
WRITE(IPRINT,1000) IN
1000 FORMAT(12H ENVIRONMENT,13.34H IS MISSING. EXECUTION SUPPRESSED)
INHIRT=INHIRT+1
RETURN
END

```

```

C * * * * * SHORTEST TIME RTIME(P,P,0.01,H) * * * * *
C * * * * * COMPUTES OVERPRESSURE AS A FUNCTION OF RANGE(K-FT). VIEW(KT) * * * * *
C * * * * * AND HORIZ(K-FT) (RANGE CURVES) * * * * *
C * * * * * WPH=W/SORT(R**2+H**2)**.3 * * * * *
C * * * * * RTWRH=SQRT(WRH) * * * * *
C * * * * * DPW=1.5R*WRH+5.4*RTWRH+.0215 * * * * *
C * * * * * P=(.002655*DPW)/(1.+(.000172R+1.921E-9*DPW)*DPW) * * * * *
C * * * * * *(1.00471R+(.0483+6.856E-6*DPW)*DPW)/(1.+(.00799+ * * * * *
C * * * * * *3.844E-6*DPW)*DPW)+2.)*DPW * * * * *
C * * * * * IF(P.LT.H) RETURN * * * * *
C * * * * * HNR2=(H/R)**2 * * * * *
C * * * * * P=P*HNR2+(3.16*WRH+7.6367556*RTWRH+.0215)*(1.-HNR2) * * * * *
C * * * * * RETURN * * * * *
C * * * * * END * * * * *

```

```

SUBROUTINE PTOR(R,P,W,H)
COMMON/INUNIT/ IREAD,IPRINT,IPUNCH,ITAPE
C * * * * *
C COMPUTES RANGE(K-FT) AS A FUNCTION OF OVERPRESSURE (PSI) AND
C YIELD(KT) AT ZERO HRR(K-FT). THEN ITERATES TO FIND RANGE AT
C A NON-ZERO HRR(K-FT) (BRIDGE CURVE)
C * * * * *
IF(P.LT.100.) WRITE (IPRINT,2000)
2000 FORMAT(37HWARNING - PRESSURE LESS THAN 100 PSI )
R0=(2.*W*(3.14/(-5.4+SQRT(29.02412+6.32*P)))*2)**(1./3.)
CALL RTOP(R0,P0,W,H)
R1=.9*R0
1 IF(ABS(P-P0).LT.1.E-4) GO TO 10
CALL RTOP(R1,P1,W,H)
R2=(P-P0)*(R1-R0)/(P1-P0)+R0
IF(R2.LT.0.) GO TO 100
R0=R1
P0=P1
R1=R2
GO TO 1
10 R=R0
RETURN
100 R=0.
WRITE (IPRINT,1000)
1000 FORMAT(28H FATAL ERROR - HRR TOO LARGE )
RETURN
END

```

```

SUBROUTINE NETCK(TITLE)
COMMON/TPRINT/ TPRINT,TPRINT,TPRINTCH,ITAPE
COMMON/CDLIST/ KA,KB,KC,KD,KE,KF,KG,KH,KI,KJ,KL,
* KM,KN,KO,KP,KQ,KR,KS,KT,
* KU,KV,KW,KX,KY,KZ,KA,K1,K2,K3,
* K4,K5,K6,K7,K8,K9,KCM,KPI,KMI,KAS,
* KSI,KPR,KLP,KRP,KEO,KOL,KRI
COMMON/CNC1 / NCR,MSYS,MNAME,NAME(1)
COMMON/CNC2 / INSLNC(2,1)
COMMON/CNC3 / MPS,TPAD2,ND(1)
COMMON/CNO1 / MNINS,MNINS,INSNFT(4,1)
DIMENSION ITYPE(3,3)
C * * * * *
C ITYPE IS USED FOR EQUIP MESSAGES
C * * * * *
ITYPE(1,1)=KRI
ITYPE(2,1)=KRI
ITYPE(3,1)=KC
ITYPE(1,2)=KRI
ITYPE(2,2)=KS
ITYPE(3,2)=KS
ITYPE(1,3)=KS
ITYPE(2,3)=KV
ITYPE(3,3)=KS
C * * * * *
SEARCH THROUGH NETWORK EQUATIONS TO DETERMINE
C 1) IF THERE ARE ANY COMPONENTS THAT ARE NOT USED IN THE NETWORK
C 2) IF ANY EQUATION REFERENCES A DELETED C. SS. OR SYS
C * * * * *
INHIRE=0
DO 1 I=1,NCR
1 N0(I)=0
IXSYS=I-NAME-MSYS+1
DO 10 I=IXSYS,MNAME
INAM=NAME(I)
IT=2
IF(INAM.GT.0) GO TO 2
INAM=-INAM
IT=3
2 IL=INSLNC(I,1)
LI=I+INSLNC(2,1)

```

Reproduced from  
best available copy.

```

DO 10 I=1,LL
J=1
3 IR=INSMET(J,L)
IF(IR.GT.NCR) GO TO 9
IF(IR.GT.0) GO TO 8
C***** IR IS NEGATIVE IF C. SS OR SYS WAS DELETED
C***** IS=1 FOR C. IS=2 FOR SS. IS=3 FOR SYS
C***** IR IS THE INPUT NAME
IS=-IR+(IR/10)*10
IR=-IR/10
INHIRT=INHIRT+1
INHIRE=1
WRITE(IPRINT,1000) ITYPE(1,IS),ITYPE(2,IS),ITYPE(3,IS),IR
*,
1000 FORMAT(17H FATAL ERROR - 3A1,16,25H DELETED. CHECK NETWORK
*3A1,16)
GO TO 9
A NO(IR)=NO(IR)+1
9 IF(INSMET(2,L).EQ.3) GO TO 10
J=J+2
IF(J.EQ.3) GO TO 3
DO 11 I=1,NCR
IF(NO(I).EQ.0) WRITE(6,1011) NAME(I)
1011 FORMAT(22H WARNING - COMPONENT,16,16H IS NOT REFERENCED)
11 CONTINUE
C RETURN ONLY IF NETWORK REFERENCES DELETED C. SS. OR SYS
IF(INHIRE.NE.0) RETURN
C * * * * *
C THIS SECTION OF CODE COMPLETELY TRACES THROUGH EACH SYSTEM NETWORK
C TO MAKE SURE THAT IN COMBINING COMPONENTS THROUGH USE OF
C SUBSYSTEMS THAT NO COMPONENT IS USED MORE THAN ONCE.
C IN OTHER WORDS, A SYSTEM IS NOT PERMITTED TO HAVE A REDUNDANCY.
C * * * * *
DO 100 IS=1,IXSYS,MNAME
IF(NAME(IS).GT.0) GO TO 100
DO 20 I=1,NCR
20 NO(I)=0
DO 21 I=1,IXSYS,MNAME
21 NO(I)=0
I=INSLC(I,IS)

```

```

LL=1+INSLNC(2,IS)
NO(IS)=-1
DO 31 I=1,LL
IR=INSNET(1,I)
IF(IR.GT.MNAME) GO TO 30
NO(IR)=NO(IR)+1
30 IF(INSNET(2,I).EQ.3) GO TO 31
IR=INSNET(3,I)
IF(IR.GT.MNAME) GO TO 31
NO(IR)=NO(IR)+1
31 CONTINUE
32 DO 50 I=IXSYS.MNAME
IF(NO(I).LT.1) GO TO 50
NO(I)=-NO(I)
IL=INSLNC(1,I)
LL=IL+INSLNC(2,I)
DO 40 L=1,LL
IR=INSNET(1,L)
IF(IR.GT.MNAME) GO TO 35
IT=1
IF(NO(IR).LT.0) IT=-1
NO(IR)=NO(IR)+IT
35 IF(INSNET(2,L).EQ.3) GO TO 40
IR=INSNET(3,L)
IF(IR.GT.MNAME) GO TO 40
IT=1
IF(NO(IR).LT.0) IT=-1
NO(IR)=NO(IR)+IT
40 CONTINUE
50 CONTINUE
INAM=-NAME(IS)
DO 55 I=1,NCR
IF(NO(I).LT.2) GO TO 55
WRITE(IPRINT,1050) INAM,NAME(I),NO(I)
1050 FORMAT(4H SYS.I6.7H USES C.216,
# 41H TIMES. NETWORK PROBABILITY IS INCORRECT)
INHRT=INHRT+1
55 CONTINUE
DO 60 I=IXSYS.MNAME
IF(NO(I).GT.-2) GO TO 60

```

```

NN(I)=-NN(I)
IR=NAME(I)
IT=2
IF(IR.GT.0) GO TO 54
IR=-IR
IT=3
50 WRITE(1PRINT,1056) INAM,ITYPE(1,IT),ITYPE(2,IT),ITYPE(3,IT)
   * IR,NN(I)
1056 FORMAT(4H SYS,16.6H USES ,3A1.216,
   * 41H TIMES. NETWORK PRIORITY IS INCORRECT)
   INHRT=INHRT+1
60 CONTINUE
100 RETURN
END

```

STARBUCK TIME 2005

```

COMMON/MDRKEY/  KEY,KEY,MKEY,IPAD1,P72(1)
COMMON/IDUNIT/  IPAD,IPRINT,IPUNCH,ITAPE
COMMON/CARPAT/  NCARD,IFCH0,INHIRT,NCODE,
*             *   MILENT,NEXT,IREP,INTPT,FRCPRT,PLNG,LEPRT,
*             *   ICOL,ICOL,MCOL,ICARD(RO)
COMMON/CE01 /  MRYN,MRYM,MNDE(1)
COMMON/CE02 /  IIN  (1)
COMMON/CE05 /  FACTK (1)
COMMON/CE05 /  RETA  (1)
COMMON/CE06A/  SIGSS (1)
COMMON/CE06C/  CSS(1)
COMMON/CE06D/  TXSS(1)
COMMON/CE07A/  SIGAU (1)
COMMON/CE07B/  RHOCOL(1)
COMMON/CE07D/  CAU(1)
COMMON/CE07E/  TYAU(1)
COMMON/CE08 /  RNVDX(1)
COMMON/CR01 /  IODSS,NIMIN,NIMAX,NOMIN,NCOMFI,NUMMR,
*             *   FPSI,FPSO,FPSOC,FPSMAT,CONE1(7)
COMMON/CR02 /  W0,W0,H0,P0,R0,R(1)
COMMON/CR03 /  NR,NRP,OP(1)
COMMON/CR01 /  NCR,NSYS,MNAME,NAME(1)
COMMON/CR02 /  INSLC(2,1)
COMMON/CR03 /  MPS,TPAD2,P(1)
COMMON/CE01 /  NCINS,MCINS,INSC(4,1)
COMMON/CC02 /  COFF(3,1)
COMMON/CC03 /  FCY(4,1)
COMMON/CC04 /  FCX(4,1)
COMMON/CC05 /  FC  (4,1)
COMMON/CC06 /  EU  (4,1)
COMMON/CC07 /  THD(8,1)
COMMON/CM01 /  NMINS,MNINS,INSNET(4,1)
COMMON/CT01 /  X(1)
COMMON/CT02 /  XBAR(1)
COMMON/CT03 /  XDEL(1)
COMMON/CT04 /  REG(1)
*             *   EQUVALENCE (REG(1),71)
COMMON/CT05 /  D7(1)
COMMON/CT06 /  PSAVE(22,1)
COMMON/CRETA/  IPAD3,MAXRH,ADMAX,DM(3),HD(21,3)

```

```

DIMENSION PERCENT(14),PCT(14)
COMMON/AZMUTH/ A7RN
C * * * * *
C INITIALIZE SUBROUTINE PROG
C MODE(MRYMT3) - ORIGINAL NAME AND POSITION OF ENVIRONMENTS
C NRYNPI - USED IN INDEXING SYSTEMATIC COVARIANCE MATRIX
C IXSYS THROUGH MNAME - INDEX TO SS AND SYS PROBABILITIES
C * * * * *
MRYMT3=3*MRYM
NRYNPI=NRYM+1
C***** RANDOM VARIATION CONVERGENCE TEST NEEDS SUM AND SUM-SQ OF
C***** PROBABILITIES FOR EACH SS AND SYS
IXSYS=MNAME-NSYS+1
IF(IXSYS.GT.NSYS) GO TO 2
WRITE(I,PRINT,ERR)
RRR FORMAT(44H PROGRAM ERROR - CONVERGENCE STORAGE PROBLEM )
RETURN
C***** SET CONVERGENCE TEST VALUES
? EPSI2=EPSI*EPSI
TNEPSO=2.*EPSOC
EPSOSO=EPSO*EPSO
C***** DECOMPOSE SIGSS COVARIANCE MATRIX - RESULT IS CSS
NRSS=1
CALL RNVD(NRSS,NRYM,SIGSS,CSS,RNVDX,IXSS)
C***** IF DECOMPOSITION FAILED, RETURN
IF(NRSS.EQ.0) RETURN
C***** ASSUME NO LOGIO ENVIRONMENT - SET TO DECOMPOSE SYSTEMATIC
C***** COVARIANCE MATRIX ONLY ONCE FOR ALL R/OP
MODEGN=1
C*****
C BEGIN RANGE/PRESSURE LOOP - THIS LOOP INCLUDE ALL OF THE CODE
C DOWN TO THE END CAPD
C*****
IR=0
C * * * * *
C SELECT RANGE/PRESSURE
C * * * * *
10 IR=IR+1
IF(IR.GT.10) RETURN
C***** PRINT MIN(O)=PRESSURE
IN=0

```

```

WRITE(IPRINT,1000) I0,OP(IR)
1000 FORMAT(5H IULN(.12,2H)=,F15.5)
C* * * * *
C SCALF AND PRINT ENVIRONMENTS
C* * * * *
I0=MAXVAL3
DO 25 I=1,NRYN
C***** GET THE ENVIRONMENT NAME FROM MODE(I0) FOR PRINTING
I0=I0+1
C***** SCALE EACH ENVIRONMENT USING RANGE OR PRESSURE
IOP=MODE(I)
IF(IOP.LT.4) SCALF=(R(IR)/R0)**HETA(I)
IF(IOP.GT.3) SCALF=(OP(IR)/OP0)**HETA(I)
GO TO (1,12,13,11,12,13),IOP
C***** MODE IS LN NORMAL
11 XRAR(I)=IULN(I)+ALOG(SCALF)
XDEL(I)=EXP(XRAR(I))
WRITE(IPRINT,1000) MODE(I0),XDEL(I)
GO TO 25
C***** MODE IS NORMAL
12 XRAR(I)=IULN(I)*SCALF
WRITE(IPRINT,1000) MODE(I0),XRAR(I)
GO TO 25
C***** MODE IS NORMAL LOG10 - SFT TO CONSTRUCT AND DECOMPSE
C***** SYSTEMATIC COVARIANCE MATRIX FOR EACH R/OP
13 XRAR(I)=IULN(I)+ALOG10(SCALF)
MODEG0=3
WRITE(IPRINT,1000) MODE(I0),XRAR(I)
25 CONTINUE
C***** PRINT CURRENT THREAT CONDITIONS
WRITE(IPRINT,3000) W,H,R(IR),OP(IR)
3000 FORMAT(3H W=,F11.4,3H H=,F11.4,3H R=,F11.4,3H P=,F11.4)
C* * * * *
C CONSTRUCT SYSTEMATIC COVARIANCE MATRIX
C CONSTRUCT ONLY ONCE
C SKIP IF MATRIX IS CONSTRUCTED
C LOG10 REQUIRES CONSTRUCTION EACH TIME
C* * * * *
GO TO (30,40,31),MODEG0
30 MODEG0=2
C***** K-FACTOR DETERMINES STANDARD DEVIATION

```

Reproduced from  
best available copy.

```

31  II=1
   DO 32  I=1,NRYN
   DI=FACTK(I)
   SIGAU(II)=0.0
   IF(ABS(DI)-1.0).LT.0.0001) GO TO 32
   IF(MODE(I).EQ.3)  DI=SQRT((XBAR(I)+DI)/(XBAR(I)-DI))
   IF(MODE(I).EQ.6)  DI=SQRT((XBAR(I)+DI)/(XBAR(I)-DI))
   SIGAU(II)=ALOG(DI)/1.96
32  II=II+NRYNPI-1
C***** RHOCOL  CONTAINS CORRELATION COEFFICIENT
L=0
II=1
KI=N1,NM+2
DO 34  I=2,NRYN
L=L+1
KK=II
DO 33  K=I,NRYN
L=L+1
KK=KK+KI--K
SIGAU(L)=RHOCOL(L)*SIGAU(II)*SIGAU(KK)
33  CONTINUE
34  II=II+KI-1
SIGAU(II)=SIGAU(II)**2
SIGAU(II)=SIGAU(II)**2
C***** DECOMPOSE  SIGAU  COVARIANCE MATRIX - RESULT IS CAU
NRAU=2
CALL  RNVD(NRAU,NRYN,SIGAU,CAU,RNVDX,IXAU)
C***** IF DECOMPOSITION FAILED, RETURN
IF(NRAU.EQ.0) RETURN
C***** SET SUMMATION CELLS AND COUNTERS
40  DO 50  I=1,NNAME
P7(I)=0.0
P72(I)=0.
DO 50  II=1,22
PSAVE(II,II)=0.0
50  CONTINUE
N0=0
N1=0
NP72=0
NTEST=NIMAX
NISTOP=NIMIN

```



```

C***** SKIP IF NO POINTS VARY
IF (TRD(R,I).EQ.0) GO TO 130
C----- UNIFORM RANDOM NUMBER GENERATOR
CALL UPR1(NIMNR,RN)
FL1=RN*RDMAX+1.0
L=FL1
LP1=L+1
FL2=L
FL1=FL1-FL2
FL2=1.0-FL1
KK=TRD(7,I)
DO 129 K=1,KK
LL=TRD(K,I)
IF (LL.EQ.0) GO TO 129
RDV=FL1*RD(LP1,LL)+FL2*RD(L,LL)-DM(LL)
FCX(K,I)=FC(K,I)+FII(K,I)*RDV
129 CONTINUE
130 CONTINUE
C*****
C BEGIN RANDOM VARIATION LOOP (INNER LOOP)
C SELECT ENVIRONMENT VARIATION
C*****
200 DO 201 I=1,NRYN
201 RNVDX(I)=0.
L=0
DO 202 I=1,NRSS
C----- NORMAL RANDOM NUMBER GENERATOR
CALL ENRN(NIMNR,RN)
DO 202 J=1,NRSS
L=L+1
202 RNVDX(I)=RNVDX(I)+RN*CSS(L)
L=IXSS(I)
FL1=X(I)+RNVDX(I)
C***** TAKE ANTILING OF LN NORMAL ENV.
IF (MODE(I).EQ.1) FL1=EXP(FL1)
IF (MODE(I).EQ.4) FL1=EXP(FL1)
C***** USE INPUT SUBSCRIPT TO STORE ENV.
L=MBYMI3+1
LL=MODE(L)
XDFL(LL)=FL1

```



```

LL=IHD(7,N)
IF(71.GT.FCX(LL,N)) GO TO 229
DO 227 I=2,LL
IF(71.GT.FCX(I,N)) GO TO 227
P(N)=(FCY(I,N)-FCY(I-1,N))*71-FCX(I-1,N)
* / (FCX(I,N)-FCX(I-1,N))+FCY(I-1,N)
P7(N)=P7(N)+P(N)
GO TO 219
227 CONTINUE
228 P(N)=0.0
GO TO 219
229 P(N)=1.0
P7(N)=P7(N)+1.0
219 CONTINUE
C* * * * *
C USE NETWORK TO FIND SS AND SYS PROBABILITY OF FAILURE
C* * * * *
II=MNAME+1
231 II=II-1
IF(II.LT.IXSYS) GO TO 251
L=INSLUC(1,II)
LL=L+INSLUC(2,II)
DO 250 I=L,LL
IR1=INSNET(1,I)
IRP=INSNET(2,I)
IR2=INSNET(3,I)
IR3=INSNET(4,I)
GO TO (241,242,243),IRP
241 P(IR3)=P(IR1)+P(IR2)-P(IR1)*P(IR2)
GO TO 250
242 P(IR3)=P(IR1)*P(IR2)
GO TO 250
243 P(IR3)=P(IR1)
250 CONTINUE
P7(II)=P7(II)+P(IR3)
GO TO 231
251 CONTINUE
C* * * * *
C END OF INNER LOOP - INCREMENT NUMBER OF ITERATIONS
* * * * *

```



```

256 P72(I)=0.0
   GO TO 300
C***** EACH O.L. NOW REQUIRES NISTOP I.L. ITERATIONS
263 IF(NI.LT.NISTOP) GO TO 200
C* * * * *
C   PLACE A TIC MARK FOR EACH C. SS AND SYS IN HISTOGRAM.
C   THE POSITION OF THE MARK DEPENDS ON AVERAGE PS FROM INNER LOOP.
C* * * * *
300 FNI=NI
   DO 301 I=1,NCR
     L=((FNI-P7(I))/FNI)*20.
     L=L+2
     IF(PZ(I).EQ.FNI) L=L-1
     PZ(I)=0.0
     PSAVE(L,I)=PSAVE(L,I)+1.0
301 CONTINUE
   DO 302 I=1XSYS,MNAME
     L=((FNI-P7(I))/FNI)*20.
     L=L+2
     IF(P7(I).EQ.FNI) L=L-1
     PZ(I)=0.0
     PSAVE(L,I)=PSAVE(L,I)+1.0
302 CONTINUE
C* * * * *
C   TEST CONVERGENCE OF OUTER LOOP. DO REQUIRED OUTER LOOP ITERATIONS
C   BEFORE TESTING.
C* * * * *
   NI=0
   NN=NN+1
   IF(NN.LT.NNSTOP) GO TO 100
   FND=NO
   IF(NC.NI.EQ.1) GO TO 306
C***** TEST FOR CONVERGENCE ON THE MEAN
   DO 305 I=1XSYS,MNAME
     IF(NAME(I).LT.0) GO TO 303
     IF(I*OSS.EQ.0) GO TO 305
303 FI=-.025
     S=PSAVE(22,I)
     S2=S
   DO 304 L=2.21
     FI=FI+.05

```

```

WT=PSAVE(L,I)*F1
S=S+WT
304 S2=S2+WJ*F1
NTMP=(S2-S*S/EN)/EPSN*(EN-1.0)+.5
IF(NTMP.LE.N) GO TO 305
C***** NO CONVERGENCE - ESTIMATE ITERATIONS REQUIRED
IF(NTMP.GT.(N+N)) NTMP=N+N
WRITE(IPRINT,1002) N1STOP,N0,EPSN0
N1STOP=NTMP
GO TO 100
305 CONTINUE
C***** MEAN CONVERGED
NMI=1
IF(CONFI(I).EQ.0.0) GO TO 404
C***** TEST FOR CONVERGENCE ON THE MEDIAN
306 L=0
DO 401 I=1,NCONFI
CN=CONFI(I)*FN
FNI= SORT(CN*(1.0-CONFI(I)))
L=L+1
PERCENT(L)=CN-FNI
L=L+1
401 PERCENT(L)=CN+FNI
DO 405 I=1,NSYS,MNAME
IF(NAME(I).LT.0) GO TO 402
IF(JNDS.EQ.0) GO TO 405
402 CALL PRRI(L,PSAVE(I,I),PERCENT,PCT)
II=0
403 II=II+2
IF(II.GT.L) GO TO 405
IF((PCT(II)-PCT(II-1)).LE.TNPSN) GO TO 403
C***** NO CONVERGENCE - SET TO DO MORE ITERATIONS (UP TO 625)
WRITE(IPRINT,1002) N1STOP,N0,TNPSN
IF(N0.GE.625) GO TO 407
N1STOP=N1STOP+NMIN
IF(N1STOP.GT.625) N1STOP=625
GO TO 100
405 CONTINUE
C* * * * *
C ALL CONVERGENCE TESTS HAVE BEEN SATISFIED
C* * * * *

```

```

406 WRITE(IPRINT,1001) N1STOP,NO
1001 FORMAT(21H CONVERGENCE - N1=,I6.4H NO=,I6)
C * * * * *
C PRINT RESULTS FOR THIS RANGE/PRESSURE
C * * * * *
407 CALL PROMEN(FNO)
GO TO 10
END

```

```

SUBROUTINE RIND(MAT,N,S,CS,C,IV)
COMMON/WRKEY/ IKEY,LKEY,WKEY,IPADL,KEY(1)
COMMON/IDUNIT/ IPAD,IPRINT,IPINCH,ITAPE
COMMON/CRO1 / IDOSS,NIMIN,NIMAX,NOMIN,NCOMF,NUMMR,NUMNR,
* EPSI,EPSC,EPSCC,EPSCMAT,CONF(7)
DIMENSION S(1),CS(1),C(N,1),IV(1),JV(1)
EQUIVALENCE (JV(1),KEY(1))
C* * * * *
C IV=0 INDICATES ROW AND COLUMN NOT YET PROCESSED
C EXPAND THE TRIANGULAR MATRIX TO A FULL MATRIX
C* * * * *
L=0
DO 5 J=1,N
IV(J)=0
DO 5 I=J,N
L=L+1
C(I,J)=S(L)
5 C(J,I)=S(L)
C* * * * *
C BEGIN DECOMPOSITION OF COVARIANCE MATRIX
C* * * * *
DO 40 K=1,N
C* * * * *
C LOOK FOR LARGEST DIAGONAL ELEMENT WHICH IS PIVOTAL ELEMENT OF
C ROW AND COLUMN NOT YET PROCESSED
C* * * * *
I1=0
DMAX=-1.E10
DO 10 I=1,N
IF (IV(I).NE.0)GO TO 10
IF (DMAX.GE. C(I,I))GO TO 10
DMAX=C(I,I)
I1=I
10 CONTINUE
C* * * * * SEE IF ALL ROWS AND COLUMNS HAVE BEEN PROCESSED
IF (I1.EQ.0) GO TO 40
C* * * * * IV LISTS THE ORDER IN WHICH ROWS AND COLUMNS ARE PROCESSED
C* * * * * JV LISTS THE ORIGINAL ORDER OF ROWS AND COLUMNS
IV(I1)=K
JV(K)=I1
C(11)=C(I1,I1)

```



```

C * * * * * ELIMINATES ZEROS WHEN RANK IS LESS THAN DIMENSION * * * * *
C * * * * *
NR=N
L=0
DO DO J=1,N
JJ=JV(J)
IF(JJ.NE.0) GO TO 60
NR=NR-1
GO TO 80
60 DO I=J,N
II=JV(I)
IF(II.EQ.0) GO TO 70
L=L+1
CS(L)=C(JJ,II)
70 CONTINUE
80 CONTINUE
IF(NR.EQ.N) GO TO R3
GO TO(R1,R2),MAT
R1 WRITE(IPRINT,10R1) N,NR
10R1 FORMAT(21H S MATRIX - DIMENSION,14,2H HAS RANK,14)
GO TO R3
R2 WRITE(IPRINT,10R2) N,NR
10R2 FORMAT(21H A MATRIX - DIMENSION,14,9H HAS RANK,14)
R3 MAT=NR
IF(MAT.NE.0) RETURN
CS(1)=0.0
MAT=1
RETURN
90 DO I=1,N
IF(IV(I).NE.0) GO TO 91
II=I
C111=C(I,I)
GO TO 100
91 CONTINUE
100 GO TO (101,102),MAT
101 WRITE(IPRINT,1101) II,II,C111,-EPSMAT
1101 FORMAT(47HCS MATRIX NOT POSITIVE SEMI DEFINITE. FLEMPNT(
*12,14,,12,2H)=FR,1.35H IS LESS THAN -EPSMAT WHERE EPSMAT= FR,1)
MAT=0
RETURN
102 WRITE(IPRINT,1102) II,II,C111,EPSMAT

```

```
1102 FORMAT(47H0A MATRIX NOT POSITIVE SEMI DEFINITE. ELEMENT(  
#I2.1H.,I2.2H)=E8.1,34H IS LESS THAN -EPSMAT WHERE EPSMAT= FR.1)  
MAT=0  
RETURN  
END
```



```

SUBROUTINE TABLE(X,V,NX,V)
  DIMENSION X(1),Y(1)
  C * * * * *
  C * THIS ROUTINE MAY BE CALLED BY USER (FOR AZIMUTH SENSITIVITY) FOR
  C * TABLE LOOK-UP WITH LINEAR INTERPOLATION.
  C * * * * *
  DO 10 I=2,NX
    IF(V.GT.X(I)) GO TO 10
    V=(Y(I)-Y(I-1))*(V-X(I-1))/(X(I)-X(I-1))+Y(I-1)
  GO TO 11
  10 CONTINUE
  11 RETURN
  END

```

```

SUBROUTINE PRINTED(FEND)
COMMON/CT06 IT/ TREAD, IPRINT, IPINCH, ITAPE
COMMON/CT07 IKEY, IKEY, MKEY, IPAD, KEY(1)
COMMON/CT01 / MCR, NSYS, MNAME, NAME(1)
COMMON/CT03 / MDS, IPAD2, P(1)
COMMON/CT06 / PSAVE(22,1)
COMMON/CT07 / OPINCH, PROR, MPI, OI, P2, OR(1)
DIMENSION PERCENT(7), PCT(7)
C * * * * *
C * * * * * PRINT THE HISTOGRAM WHICH REPRESENTS THE DENSITY DISTRIBUTION * * * * *
C * * * * *
C * * * * * IXSYS=MNAME-NSYS+1
WRITE(IPRINT,1005)
1005 FORMAT(90HOPS=0 (----- HISTOGRAM (IN STEPS OF .05) FOR PS GREATER
*THAN 0 AND LESS THAN -----) P5=1 )
DO 365 I=1,MCR
WRITE(IPRINT,1006) (PSAVE(I,I), II=1,22), NAME(I)
1006 FORMAT(1H ,22F4.0,19)
365 CONTINUE
366 CONTINUE
C * * * * * PREPARE PERCENTILE LEVELS
PERCENT(1)=.01*FND
PERCENT(2)=.1*FND
PERCENT(3)=.3*FND
PERCENT(4)=.5*FND
PERCENT(5)=.7*FND
PERCENT(6)=.9*FND
PERCENT(7)=.99*FND
NPER=7
MIDPER=4
IR=1
IF=MCR
C * * * * *
C * * * * * FOR EACH C, SS AND SYS, CALL FOR THE CUMULATIVE PROBABILITY * * * * *
C * * * * * AT THE DIFFERENT PERCENTILES.
C * * * * *
1 DO 50 I=IR,IE
CALL PUNR(MPER,PSAVE(1,I),PERCENT,PCT)
40 DO 41 IP=1,MPER

```

```

41 PSAVE(IP,I)=PCT(IP)
P(I)=PCT(MIDPER)
KEY(I)=I
50 CONTINUE
IF(IR.GT.1) GO TO 60
IR=MNAME-NSYS+1
IF=MNAME
GO TO 1
60 CONTINUE
C * * * * *
C * RANK COMPONENTS BY HARDNESS AT THE 50 PERCENT LEVEL.
C * * * * *
C * * * * *
CALL RORD(P,KEY,NCR)
IF(OPINCH.FO.O.O) GO TO 62
C * * * * *
C * IF PLOTS REQUESTED SAVE 50 PERCENT VALUES FOR SS AND SYS
C * * * * *
NPROR=NPROR+1
IP=NPROR
DO 61 I=14,16
PRR(IP)=P(I)
IP=IP+NPLT
61 CONTINUE
C * * * * *
C * RANK SS AND SYS HARDNESS AT THE 50 PERCENT LEVEL.
C * * * * *
62 CALL RORD(P(IH),KEY(IR),NSYS)
C * * * * *
C * PRINT CUMULATIVE PROBABILITY RESULTS FOR C. SS AND SYS
C * * * * *
WRITE(IPRINT,1000)
1000 FORMAT(22HINDIVIDUAL COMPONENTS )
WRITE(IPRINT,1001)
1001 FORMAT(IH ,6X,4HNAME,7X,1H).6X,2H10.6X,2H30.6X,2H50.6X,2H70.
      *6X,2H90.6X,2H99)
DO 70 I=1,NCR
N=KEY(I)
WRITE(IPRINT,1002) MAVE(N),(PSAVE(IP,N),IP=1,NPER)
1002 FORMAT(IH ,19.7F8.3)
70 CONTINUE
WRITE(IPRINT,1003)

```

Reproduced from  
best available copy.

```

1003 FORMAT(22H0INDIVIDUAL SUBSYSTEMS )
WRITE(IPRINT,1003)
DO 90 I=1R,IE
N=KEY(I)
IF(NAME(N).LT.0) GO TO 90
WRITE(IPRINT,1002) NAME(N),(PSAVE(IP,N),IP=1,NPER)
90 CONTINUE
WRITE(IPRINT,1004)
1004 FORMAT(19H0INDIVIDUAL SYSTEMS )
WRITE(IPRINT,1001)
DO 90 I=1R,IE
N=KEY(I)
IF(NAME(N).GT.0) GO TO 90
WRITE(IPRINT,1002) NAME(N),(PSAVE(IP,N),IP=1,NPER)
90 CONTINUE
RETURN
END

```

Reproduced from  
best available copy. 6

```

SURROUTINE PRORL(LI,PS,PFCNT,PCT)
DIMENSION PS(1),PFCNT(1),PCT(1)
C* * * * *
C   GIVEN THE HISTOGRAM DENSITY, PRORL FORMS THE CUMULATIVE AND
C   DETERMINES, BY LINEAR INTERPOLATION, THE VARIOUS PERCENTILES
C* * * * *
CUMN=PS(1)
I=1
L=0
C***** PROCESS DISCRETE EVENT - PROBABILITY=0
1 L=L+1
IF(L.GT.LI) RETURN
IF(PFCNT(L).GT.CUMN) GO TO 2
PCT(L)=0.
GO TO 1
C***** PROCESS CONTINUOUS DISTRIBUTION
2 I=I+1
IF(I.EQ.22) GO TO 4
CUM=CUMN
CUMN=CUMN+PS(I)
3 IF(PFCNT(L).GT.CUMN) GO TO 2
VAL=I-2
PCT(L)=.05*(VAL+(PFCNT(L)-CUM)/PS(I))
L=L+1
IF(L.GT.LI) RETURN
GO TO 3
C***** PROCESS DISCRETE EVENT - PROBABILITY=1
4 DO 5 I=L+1,L
5 PCT(IL)=1.0
RETURN
END

```

```

SUBROUTINE RORD(L,LL,N)
REAL L,LT
DIMENSION L(N),LL(N)
C*****
C THIS ROUTINE RANKS THE HARDNESS OF COMPONENTS, SUBSYSTEMS AND
C SYSTEMS BY THEIR MEDIAN (50 PERCENTILE) VALUES.
C*****
IF(N.LE.1) RETURN
I = N/2
M = N
NFLAG = 0
GO TO 40
I = I-1
IF(I.GT. 1) GO TO 40
I = 1
M = N
NFLAG = 1
K = I
LT = L(K)
LL=LL(K)
GO TO 60
L(K) = L(J)
LL(K)=LL(J)
K = J
J = 2*K
IF(J-M) 70,80,90
IF(L(J+1) .LE. L(J)) GO TO 80
J = J+1
IF(L(J) .GT. LT) GO TO 50
L(K) = LT
LL(K)=LL
IF(NFLAG .EQ. 0) GO TO 20
LT = L(I)
LL=LL(I)
L(I) = L(M)
LL(I)=LL(M)
L(M) = LT
LL(M)=LL
M = M-1
IF(M .GT. 1) GO TO 40
RETURN
END

```

```

SUBROUTINE DPLOT
COMMON/IOUNIT/ IREAD, IPRINT, IPINCH, ITAPE
COMMON/CRO3 / NR, NNP, NP(1)
COMMON/CNC1 / NCR, NSYS, MNAME, NAME(1)
COMMON/CT07 / OPINCH, NPROR, MPLOR, PROR(1)
COMMON/CT07R/ IR(63), IS(63), IPLI(31,4)
COMMON /CHRLST/ KA, KR, KC, KD, KE, KF, KG, KH, KI, KJ,
* KK, KL, KM, KN, KO, KP, KQ, KR, KS, KT,
* KU, KV, KW, KX, KY, KZ, KO, KI, K2, K3,
* K4, K5, K6, K7, K8, K9, KCM, KPL, KMI, KAS,
* KSL, KPR, KLP, KRP, KFD, KDL, KRL
COMMON/CNC3 / MPS, IPAD2, P(1)
C* * * * *
C SCALFS PRESSURES BY THE VALUE ENTERED ON PLOTION CARD AND
C INITIALIZES PRINT LINES.
C* * * * *
DO 11 I=1, NR
11 P(I)=OP(I)/OPINCH
DO 1 I=1, 62
IR(I)=KPR
IS(I)=KRL
1 CONTINUE
IR(1)=KRL
IS(2)=KPR
IR(63)=KRL
IS(63)=KRL
C* * * * *
C LOCATE PLOT DATA FOR TWO PLOTS AND PRINT IDENTIFICATIONS
C* * * * *
IP=1-MPLOR
NR=MNAME-NSYS-1
? NR=NR+?
IF(NR.GT.MNAME) RETURN
NF=NR+1
IF(NF.GT.MNAME) NF=NR
WRITE(IPRINT,1000) (NAME(I), I=NR, NF)
1000 FORMAT(1H,16,5X,16)
C* * * * *
C INITIALIZE VECTORS FOR POSITIONING PLOT POINTS BY SPLIT AND
C CALL SPLIT ONCE FOR EACH PLOT
C* * * * *

```

```

00 3 I=1,31
IPLT(I,1)=2
IPLT(I,3)=2
IPLT(I,2)=KRI
IPLT(I,4)=KRI
3 CONTINUE
IPLT(31,2)=KPP
IPLT(31,4)=KPR
IP=IP+PLNT
CALL SPLNT( P,PRNR(IP),NR,IPLT(I,1))
IF(NR,FO,NE) GO TO 4
IP=IP+PLNT
CALL SPLNT( P,PRNR(IP),NR,IPLT(I,3))
4 CALL PZPLNT
GO TO 2
END

```



```

SUBROUTINE PRPLOT
COMMON/IUNIT/ IREAD, IPRINT, IPINCH, ITAPE
COMMON/CT07A/ IR(63), IS(63), IPLT(31,4)
C * * * * * THIS ROUTINE PRINTS TWO PLOTS PER PAGE WITH AXES AND A OR V
C * * * * * SYMBOLS TO REPRESENT POINTS
C * * * * *
C * * * * * WRITE(IPRINT,1000)
1000 FORMAT(IH0)
C***** PRINT LINE OF ..... FOR PROBABILITY EQUAL TO 1
ISKP=IPLT(1,1)
ISKP2=ISKP+2
ISKP3=IPLT(1,3)
WRITE(IPRINT,1001) (IR(I), I=1, ISKP), IPLT(1,2)
*, (IS(I), I=ISKP2,63), (IR(I), I=1, ISKP3), IPLT(1,4)
C***** PRINT LINE OF BLANKS FOR OTHER PROBABILITIES
DO 1 L=2,30
ISKP=IPLT(L,1)
ISKP2=ISKP+2
ISKP3=IPLT(L,3)
WRITE(IPRINT,1001) (IS(I), I=1, ISKP), IPLT(L,2)
*, (IS(I), I=ISKP2,63), (IS(I), I=1, ISKP3), IPLT(L,4)
1 CONTINUE
C***** PRINT LINE OF ..... FOR PROBABILITY EQUAL TO 0
ISKP=IPLT(31,1)
ISKP2=ISKP+2
ISKP3=IPLT(31,3)
ISKP4=ISKP3+2
WRITE(IPRINT,1001) (IR(I), I=1, ISKP), IPLT(31,2), (IR(I), I=ISKP2,63)
*, (IR(I), I=1, ISKP3), IPLT(31,4), (IR(I), I=ISKP4,63)
1001 FORMAT(126A1)
RETURN
END

```

APPENDIX C  
FLOW CHARTS OF FAST PROGRAM

PROGRAM MAIN (INPUT,OUTPUT,TAPES=INPUT,TAPES=OUTPUT)

```

COMMON/MORKEY/ IKEY,LKEY,RKEY,IPAD1,WORK(135)
COMMON/ICUNIT/ IREAD,IPRINT,IPUNCH,ITAPE
COMMON/CARDAT/ NCARD,IECMD,INMIOT,NCODE,
MULCNT,NEXT,IREF,INTPRT,FRCPRT,ILOG,LEXPT,
ICOL,LCOL,RCOL,ICARD(80)
COMMON/CMPLT/ KSEI(47)
COMMON/NL167/ NAMES(219)
COMMON/CE01/ NBYN,NBYH,MODE(30,4)
COMMON/CE02/ ULN (30,2)
COMMON/CE03/ FACTR (30)
COMMON/CE04/ ALPHA (30)
COMMON/CE05/ BETA (30,2)
COMMON/CE06/ ORAMA (30)
COMMON/CE06A/ HB (30)
COMMON/CE06B/ ORAMA(30)
COMMON/CE06C/ SIGS6 (465)
COMMON/CE06D/ SIGS6(465)
COMMON/CE06E/ C66(465)
COMMON/CE06F/ XS6(30)
COMMON/CE07A/ SIGAU (465)
COMMON/CE07B/ RHOCOL(465)
COMMON/CE07C/ RHOCOL(465)
COMMON/CE07D/ CRAU(465)
COMMON/CE07E/ XRAU(30)
COMMON/CE08/ RNVX(30,30)
COMMON/CRO1/ I066,NMIN,NMAX,NMIN,NCONF1,NUMUR,NUMMR,
EP61,EP60,EP60C,EP6AT,CONF(7)
COMMON/CRO2/ MO,W,MO,H,OPD,RO,R(20)
COMMON/CRO3/ NR,NOP,OP(20)
COMMON/CNC1/ NCR,NGY6,NAME,NAME(135)
COMMON/CNC2/ INGLOC(2,135)
COMMON/CNC3/ MFS,IPAD2,P(150)
COMMON/CC01/ NCINS,MCINS,INSC(4,200)
COMMON/CC02/ COEF(3,200)
COMMON/CC03/ FCY(6,100)
COMMON/CC04/ FCX(6,100)
COMMON/CC05/ FC (6,100)
COMMON/CC06/ FU (6,100)
COMMON/CC07/ IBO(8,100)
COMMON/CN01/ NNINS,NNINS,INSNET(4,250)
COMMON/CT01/ X(30)
COMMON/CT02/ XBAR(30)
COMMON/CT03/ XDEL(30)
COMMON/CT04/ RE(30)
COMMON/CT05/ PZ(135)
COMMON/CT06/ PSAVE(22,135)
COMMON/CT07/ OP,INCH,NPROB,MPLOT,PROB(20,35)
COMMON/COETA/ IPAD3,MAXBO,BOMAX,OM(9),BO(21,9)
    
```

```

DIMENSION OB(21,9)
DATA OB/0 ..030,051,071,090,110,129,148,168,189,211
..234,258,284,313,345,382,426,482,564,1
0 ..061,081,117,140,162,184,205,227,248,271
..294,318,344,371,402,437,477,528,602,1..
0 ..106,145,176,203,228,251,274,296,319,341
..364,388,413,440,469,501,539,586,652,1..
0 ..163,209,244,274,301,326,350,373,396,419
..442,465,490,516,543,574,609,652,711,1..
0 ..223,279,318,350,379,405,430,454,477,500
..523,546,570,595,621,650,682,721,775,1..
0 ..289,348,391,426,457,484,510,535,558,581
..604,627,650,674,699,726,756,791,837,1..
0 ..348,414,461,499,531,560,587,612,636,659
..681,704,726,749,772,797,824,855,894,1..
0 ..398,472,523,563,598,629,656,682,706,729
..752,773,795,816,838,860,883,909,939,1..
0 ..436,518,574,618,655,687,716,742,766,789
..811,832,852,871,890,910,929,949,970,1..
    
```

```

DIMENSION I6ET(47)
DATA I6ET/
.4MA .4MB .4MC .4MD .4ME .4MF .4MG .4MH .4MI .
.4MJ .4MK .4ML .4MM .4MN .4MO .4MP .4MQ .4MR .
.4MS .4MT .4MU .4MV .4MW .4MX .4MY .4MZ .4MO .
.4MI .4MZ .4M3 .4M4 .4M5 .4M6 .4M7 .4M8 .4M9 .
.4M. .4M+ .4M- .4M# .4M/ .4M. .4M( .4M) .4M= .
.4M8 .4M /
    
```

DIRENÇION NAMES1(94)		DIRENÇION NAMES2(103)	
DATA NAMES1/			
1.	3.4MR	.4MU	.4MN
2.	4.4MS	.4MT	.4MO
3.	1.4MC		
4.	2.4MS	.4MS	
5.	3.4MS	.4MY	.4MS
6.	3.4ME	.4MN	.4MV
7.	1.4MS		
8.	1.4MR		
9.	5.4MC	.4MO	.4MN
10.	6.4MN	.4MO	.4MR
11.	1.4MR		
12.	2.4MO	.4MP	
13.	1.4MV		
14.	5.4MR	.4ME	.4MS
15.	5.4MR	.4MR	.4MN
2.	4.4MS	.4MT	.4MO
8.	5.4MC	.4MO	.4MN
10.	6.4MN	.4MO	.4MR
DATA NAMES2/			
12.	2.4MO	.4MP	
16.	4.4MR	.4MO	.4ME
16.	4.4MR	.4MO	.4ME
17.	3.4MU	.4ML	.4MN
18.	5.4MR	.4M	.4MF
19.	5.4MR	.4ML	.4MP
20.	4.4MB	.4ME	.4MT
21.	7.4MP	.4ML	.4MO
22.	6.4MP	.4ML	.4MO
23.	4.4MP	.4ML	.4MO
24.	5.4MC	.4MR	.4MN
25.	9.4MR	.4MB	.4MC
26.	6.4MR	.4MO	.4MN
26.	6.4MR	.4MO	.4MN
27.	1.1MR		
0.	0/		

```

C . . . . . INSTALLATION DEPENDENT FEATURES
C . . . . . CHANGE NORMAL AND UNIFORM RANDOM NUMBER GENERATORS
C . . . . . CHANGE CALLS TO GENERATORS WHICH ARE IN SUBROUTINE PROO
C . . . . . CHANGE DATA STATEMENT FORMATS IF NECESSARY
C . . . . . CHANGE FORTRAN INPUT/OUTPUT UNITS FOR CARD READER AND PRINTER
C . . . . .
C . . . . . IREAD=5
C . . . . . IPRINT=6
C . . . . . IPUNCH=0
C . . . . . ITAPE=0
    
```

```

C . . . . . STORE BETA DISTRIBUTIONS IN COMMON BLOCK /CBETA/
C . . . . .
C . . . . . BM=0.0
C . . . . . DO 5 I=1,9
C . . . . . BM=BM+0.1
C . . . . . BM(I)=BM
C . . . . . DO 5 J=1,21
C . . . . . BM(J,I)=BM(J,I)
C . . . . . 5 CONTINUE
C . . . . . MAXBO=21
C . . . . . BOMAX=20.0
    
```

```

C . . . . . STORE CHARACTER SET OF THIS COMPUTER IN COMMON BLOCK /CHARST/
C . . . . .
C . . . . . DO 1 I=1,47
C . . . . . 1 KSET(I)=1627(I)
    
```

```

C . . . . . STORE INPUT NAMES IN COMMON BLOCK /NLIST/
C . . . . .
C . . . . . DO 2 I=1,94
C . . . . . 2 NAMES1(I)=NAMES1(I)
C . . . . . J=94
C . . . . . DO 3 I=1,103
C . . . . . 3 NAMES2(I)=NAMES2(I)
    
```

```

C . . . . . SET MAXIMUM PROBLEM SIZE DEPENDING ON COMMON BLOCK DIMENSIONS
C . . . . .
C . . . . . MKEY=135
C . . . . . MBYM=30
C . . . . . MNAME=135
C . . . . . MP6=150
C . . . . . MCING=200
C . . . . . MNING=250
C . . . . . MPL0T=20
    
```

PO 3

```
C * * * * *  
C * GET DEFAULT VALUES - MOST MAY BE RESET BY INPUT CARDS * * * * *  
C * * * * *  
10066=0  
NIMIN=5  
NIMAX=50  
NOMIN=50  
EP61=.03  
EP62=.03  
EP63=.02  
NCONF1=1  
CONF111=0.5  
NUMUR=1  
NUMNR=1  
EP6MAT=0.0001  
OPINCH=0.0
```

```
C * * * * *  
C * INITILIZE READING OF CARDS BY NLOET * * * * *  
C * * * * *  
NCRD=0  
C * * * * * IECMB=0 WILL SUPRESS PRINTING OF CARD INPUT * * * * *  
IECMB=IPRINT  
NCRD=0
```

CALL NLREAD

STOP  
END

SUBROUTINE M\_READ

```
COMMON/OUNTY/ IREAD,IPRINT,IPUNCH,IYAPE
COMMON/CARDAT/ NCARD,IECHO,INHIBT,NCODE,
               NULCNT,NEXT,IREF,INTPRT,PRCPRT,ILOG,IEXPT,
               ICOL,LCOL,NCOL,ICARD(80)
COMMON/CE01 / NBYN,NBYN,MODE(1)
COMMON/CE02 / ULN (1)
COMMON/CE03 / PACTR (1)
COMMON/CE04 / ALPHA (1)
COMMON/CE05 / BETA (1)
COMMON/CE06 / SIGEGG(1)
COMMON/CE07 / RHCCGG(1)
COMMON/CE08 / OMMR (1)
COMMON/CE09 / NS (1)
COMMON/CE09B / DAMMR(1)
COMMON/CR01 / IOGG,NMIN,NIMAX,NOMIN,NCONF1,NUNUR,NUNNR,
             EPF,EP60,EP60C,EP6AT,CONF(7)
COMMON/CR02 / MO,N,HO,N,OPD,RO,R(1)
COMMON/CR03 / NR,NOP,OP(1)
COMMON/CT07 / OPINCH,NPROG,NPLOT,PROB(1)
```

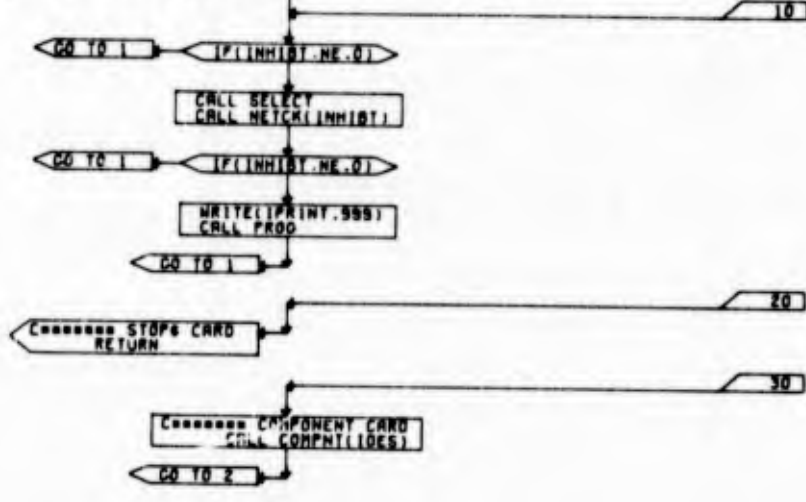
```
*****
C     RESETS ASSUMED BEFORE FIRST JOB IS READ
C     CALL RESET
```

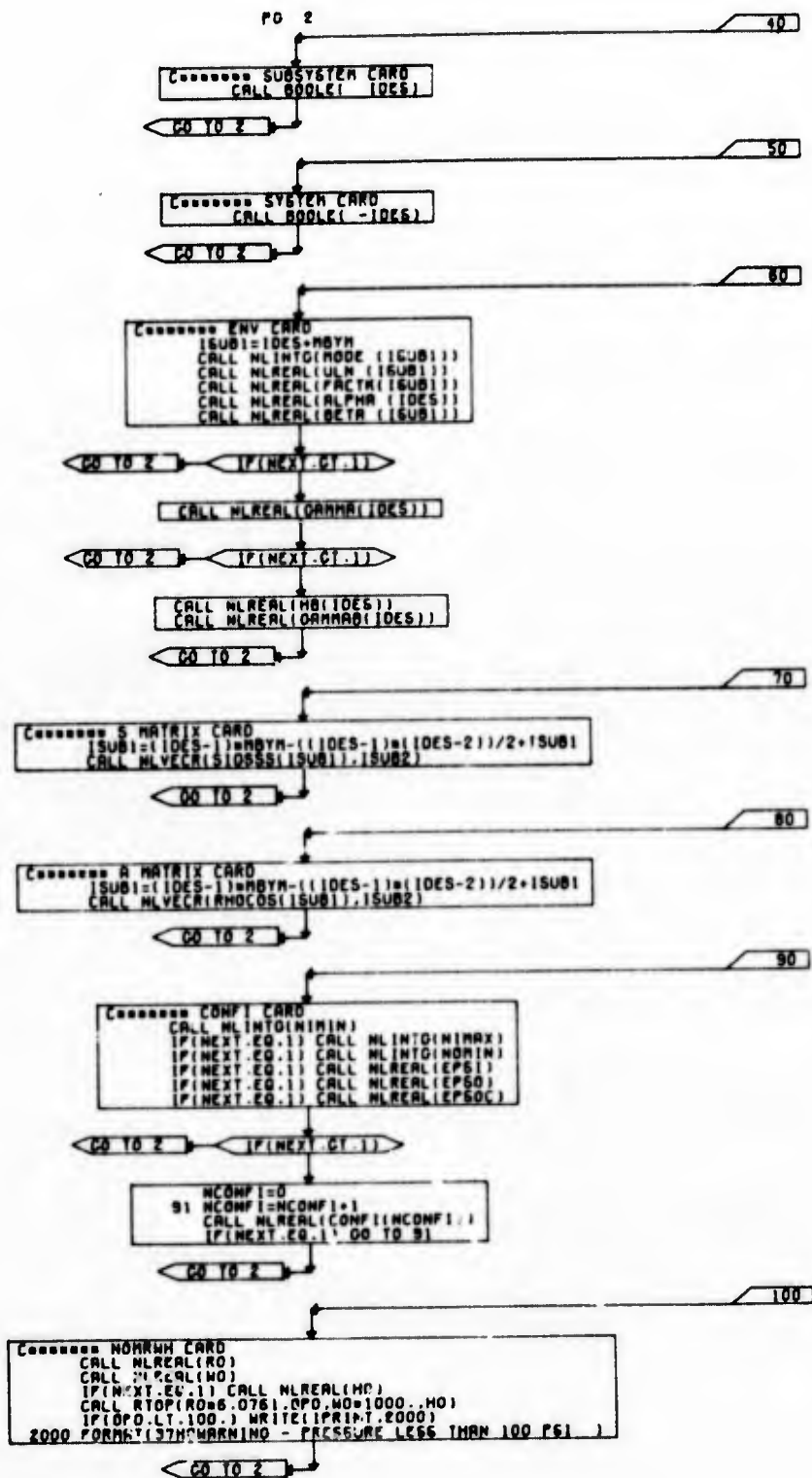
```
*****
C     READ THE NEXT DATA CARD
C     CONTINUE
C     WRITE(IPRINT,999)
C     999 FORMAT(1H1)
C     CALL NLOET
```

```
*****
C     IDENTIFY TYPE OF CARD BY INPUT NAME
C     CONTINUE
C     CALL N1NAME(NAME(X,IOCS,ISUB1,ISUB2))
```

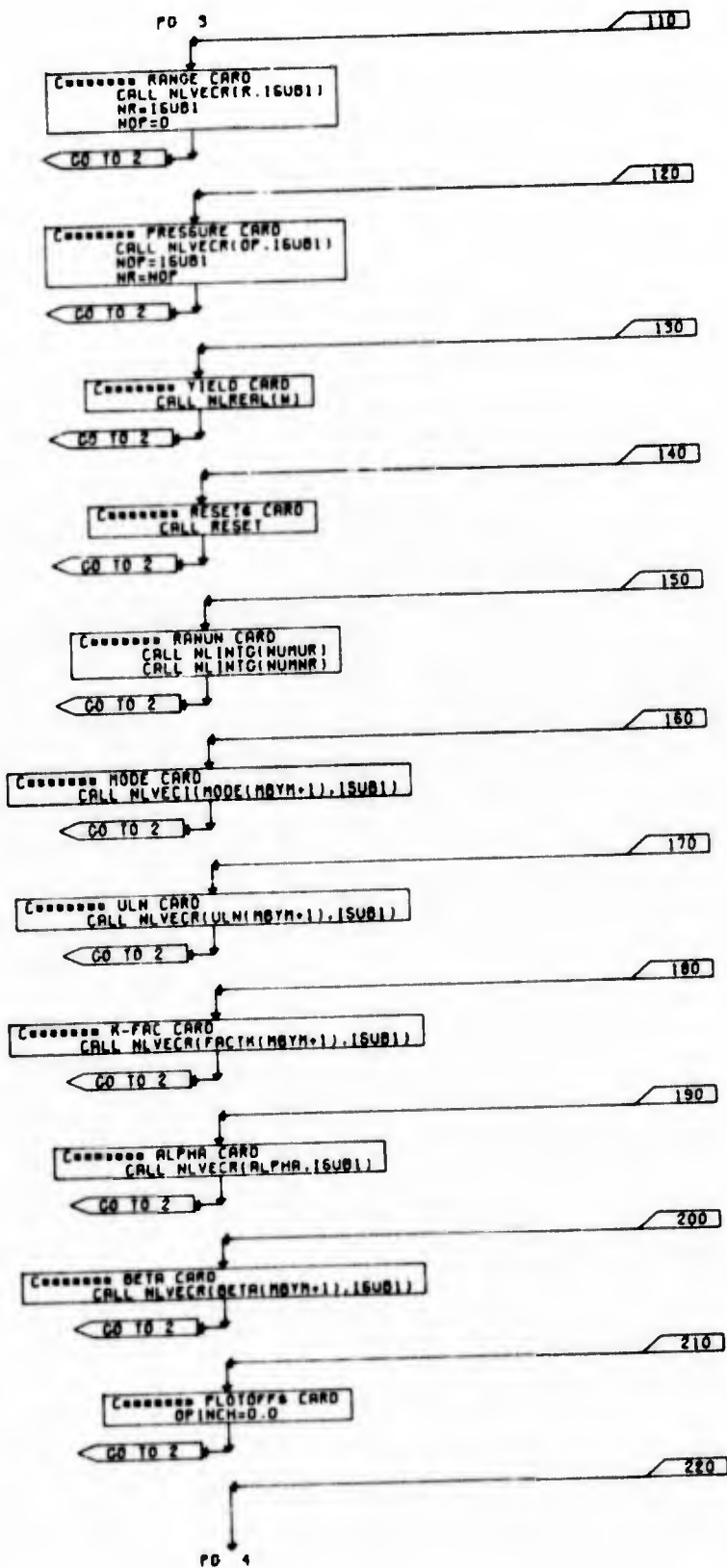
```
GO TO(10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,
      180,190,200,210,220,230,240,250,260,270)
      NAME(X)
```

```
*****
C     RUNS CARD
C     DO NOT EXECUTE WITH THIS DATA IF AN ERROR HAS BEEN FOUND
C     IF NO INPUT ERROR -
C     SELECT THE REQUIRED ENVIRONMENTS
C     CHECK FOR NETWORK ERRORS
C     PERFORM MONTE CARLO PROCESS
```

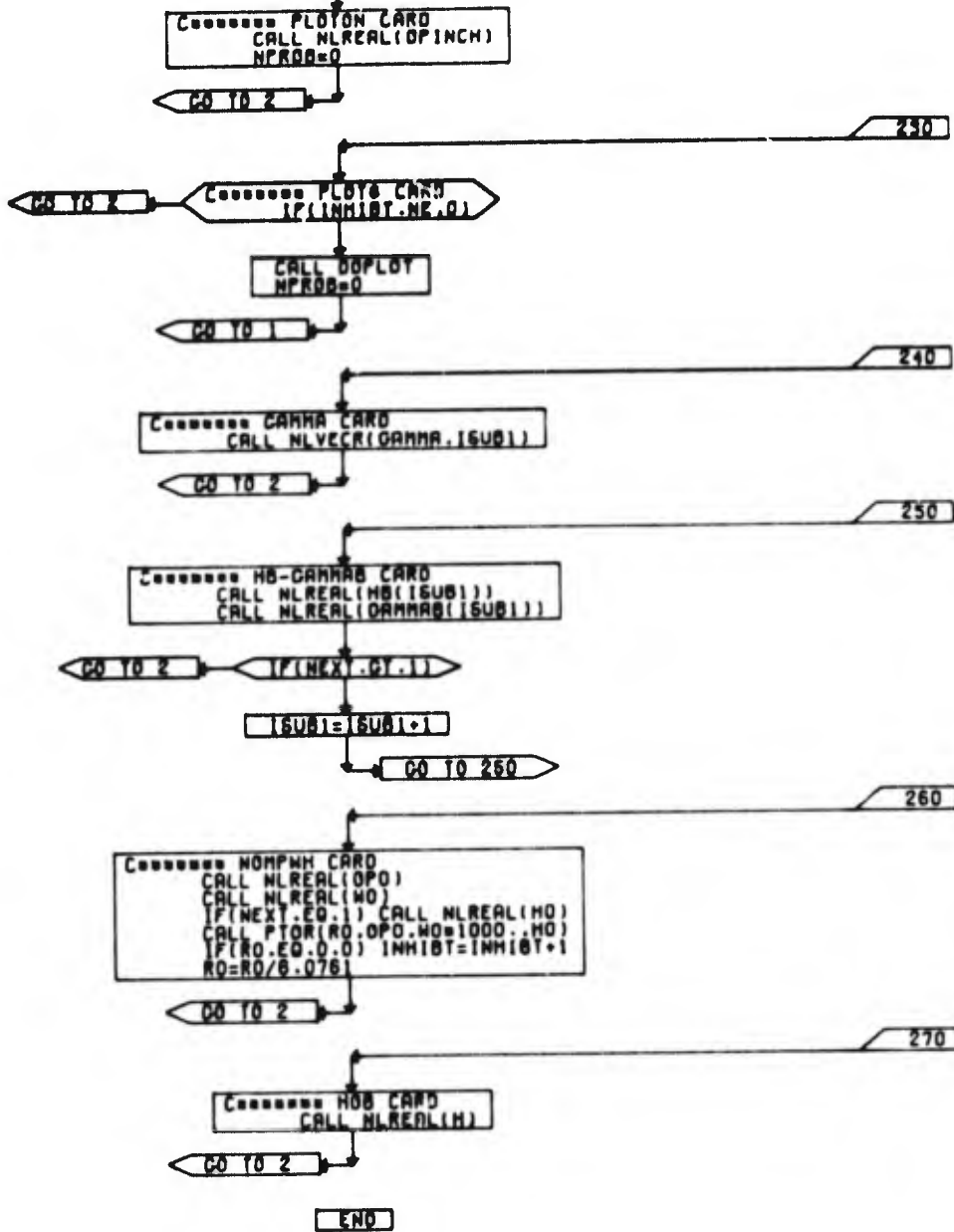




PG 3



PD 4



SUBROUTINE RESET

```
COMMON/CARDAT/ NCARD, I ECHO, INHIB, MCODE,  
              NOLCNT, NEXT, REP, INTPRT, FRCPRT, ILOC, IEXPT,  
              ICOL, LCOL, MCOL, I(CARD180)  
COMMON/CE01 / NBYN, NBYM, MODE(1)  
COMMON/CE09 / OAMMA (1)  
COMMON/CE09A/ MB (1)  
COMMON/CE09B/ SIGSSS(1)  
COMMON/CE07C/ RHOCOS(1)  
COMMON/CR02 / W, W, MO, H, OPO, RO, R(1)  
COMMON/CR03 / MR, MOP, OP(1)  
COMMON/CNC1 / MCR, NSYS, MNAME, NAME(1)  
COMMON/CC01 / MCINS, NCINS, INSC(4,1)  
COMMON/CNO1 / MNINS, MNINS, INSMN(4,1)  
COMMON/CT07 / OPINCH, NPROB, NPLOT, PROBI(1)
```

```
C * * * * *  
C * * * * * TURN OFF THE INPUT ERROR FLAG AND NUMERIC FIELD REPEAT * * * * *  
C * * * * *  
INHIB=0  
REP=0
```

```
C * * * * *  
C * * * * * INITIALIZE ENVIRONMENTS * * * * *  
C * * * * * SELECT INTERPRETS MODE(K)=-1 AS A MISSING ENVIRONMENT * * * * *  
C * * * * * MODE(KK)=0 UNLESS THIS ENVIRONMENT IS REQUIRED BY A COMPONENT * * * * *  
C * * * * * MB(1)=-1 ASSUMES NO MOB BREAK POINT * * * * *  
C * * * * * OAMMA(1)=0 ASSUMES NO MOB SCALING * * * * *  
C * * * * * SET B AND A MATRICIES TO ZERO * * * * *  
C * * * * *  
R=NBYN  
KK=R+NBYM  
DO 1 I=1,NBYM  
R=R+1  
KK=KK+1  
MODE(K)=-1  
MODE(KK)=0  
OAMMA(1)=0.  
MB(1)=-1.  
1 CONTINUE  
KK=(NBYN+NBYM+1)/2  
DO 2 I=1, KK  
SIGSSS(I)=0.  
RHOCOS(I)=0.  
2 CONTINUE
```

```
C * * * * *  
C * * * * * THE FOLLOWING VARIABLES WILL CONTAIN THE NUMBER OF RANGES, * * * * *  
C * * * * * PRESSURES, COMPONENTS, SUBSYS AND SYS, TRANSFER FUNCTION * * * * *  
C * * * * * INSTRUCTIONS AND NETWORK INSTRUCTIONS. * * * * *  
C * * * * * NPROB IS USED TO STORE PLOT POINTS (NPROB= TOTAL NUMBER * * * * *  
C * * * * * OF PRESSURES FOR ONE OR MORE STACKED CASES). * * * * *  
C * * * * * ASSUME ZERO MOB * * * * *  
C * * * * *  
MR=0  
MOP=0  
MCR=0  
NSYS=0  
MCINS=0  
MNINS=0  
NPROB=0  
M=0  
MO=0.
```

RETURN

END

SUBROUTINE NLOET

```

COMMON /OUNI/ IREAD, IPRINT, IPUNCH, ITAPE
COMMON /CARDAT/ NCARD, IECHO, INH(5), NCODE
NOL, CH1, NEXT, INEP, INIPRT, FRCPRT, ILOG, IEXPT.
COMMON /CHRLST/ KA, KB, KC, KD, KE, KF, KG, KH, KI, KJ,
KK, KL, KM, KN, KO, KP, KQ, KR, KS, KT,
KU, KV, KW, KX, KY, KZ, KA, KB, KC, KD, KE, KF,
K7, K8, K9, K0, KA, KB, KC, KD, KE, KF,
KBL, KPL, KRP, KEQ, KDL, KBL
DIMENSION IT(17), NCODE(2,47)
EQUIVALENCE (IT, KA)
    
```

```

C . . . . . READ AN 80-COLUMN CARD - ONE CHARACTER/WORD
C . . . . . NCARD CONTAINS NUMBER OF CARDS READ (USED ONLY FOR ERROR MESSAGES)
C . . . . .
1000 READ(IREAD,1000) (ICARD(I),I=1,80)
      FORMAT(80A1)
      NCARD=NCARD+1
    
```

```

C . . . . . UNLESS THIS IS A RECODE CARD GO TO 5
    
```

```

IF (ICARD(1).NE.KR)
IF (ICARD(2).NE.KE)
IF (ICARD(3).NE.KC)
IF (ICARD(4).NE.KD)
IF (ICARD(5).NE.KO)
IF (ICARD(6).NE.KI)
      GO TO 5
    
```

```

C . . . . . IDENTIFY WHICH CHARACTERS TO RECODE - THEN RECD NEXT CARD
C . . . . .
NCODE=0
DO 2 I=1,17
  IF (ICARD(I).EQ.IT(I)) GO TO 2
  NCODE=NCODE+1
  NCODE(1,NCODE)=(ICARD(I),7)
  NCODE(2,NCODE)=IT(I)
2 CONTINUE
      GO TO 1
    
```

```

C . . . . . NCODE=0 (SET IN MAIN) UNLESS A RECODE CARD DID NOT MATCH /CHRLST/
C . . . . .
      GO TO 5
    
```

```

IF (NCODE.EQ.0) GO TO 30
    
```

```

C . . . . . RECODE CARD CHARACTERS THAT DO NOT MATCH /CHRLST/
C . . . . .
DO 20 I=1,80
  DO 10 N=1,NCODE
    IF (ICARD(I).NE.NCODE(1,N)) GO TO 10
    ICARD(I)=NCODE(2,N)
  GO TO 20
10 CONTINUE
20 CONTINUE
      GO TO 30
    
```

```

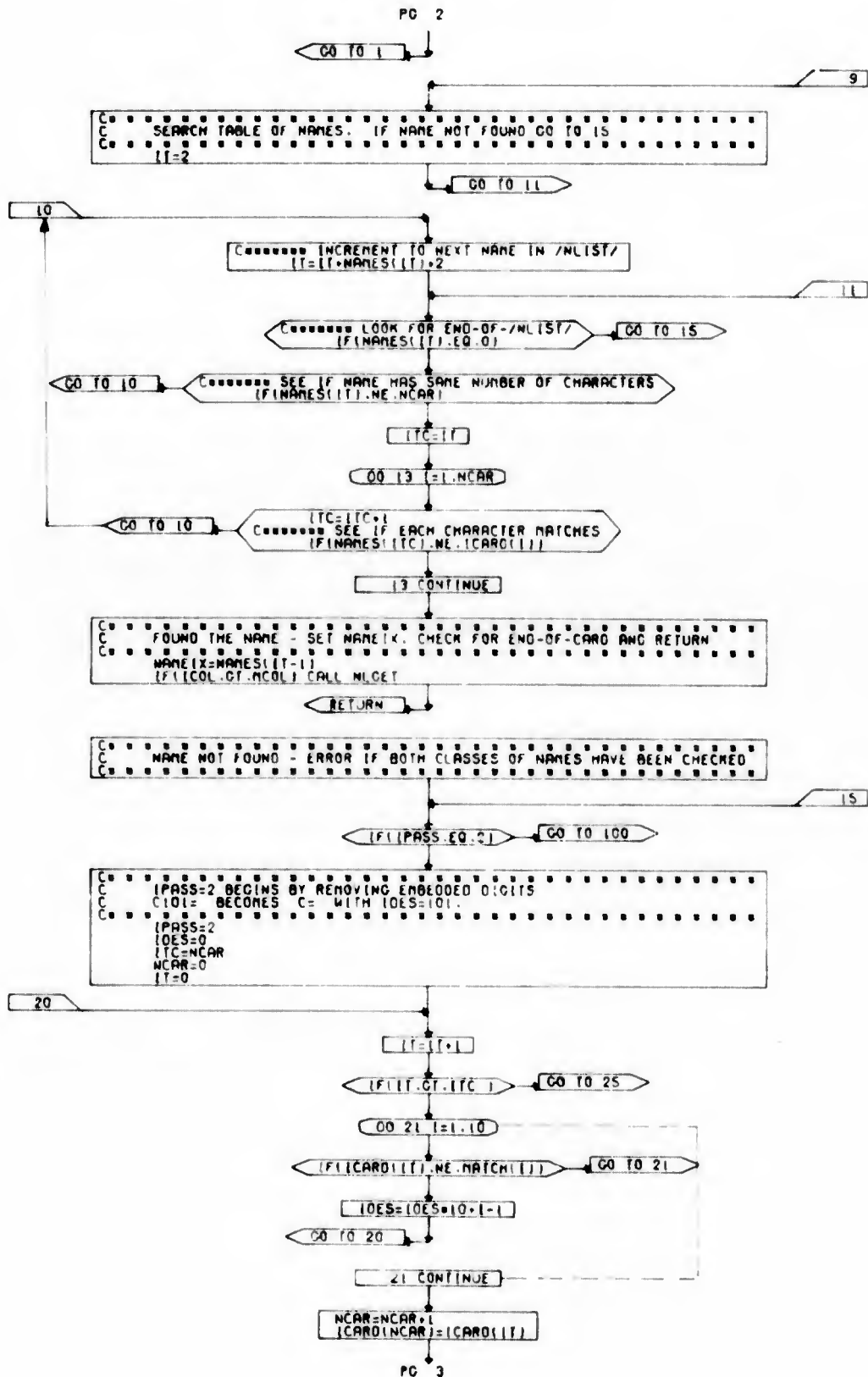
C . . . . . CURRENTLY IECHO=IPRINT (SET IN MAIN) TO LIST INPUT
C . . . . .
      IF (IECHO.NE.0) WRITE(IECHO,1001) (ICARD(I),I=1,80)
1001 FORMAT(1H,80A1)
    
```

```

C . . . . . SET LCOL AND RCOL TO LAST COLUMN OF INFORMATION.
C . . . . .
ICOL=1
LCOL=80
40 IF (ICARD(ICOL).NE.KBL) GO TO 50
LCOL=LCOL-1
IF (LCOL.GT.0) GO TO 40
GO TO 1
50 RCOL=LCOL
      RETURN
    
```

END





PG 3

GO TO 20

```
C * * * * * MAKE SECOND PASS IF DIGITS WERE REMOVED (NAME MUST BE SHORTER) * * * * *  
C * * * * *  
C * * * * *
```

25

IF(NCAR,LT,ITC) GO TO 9

100

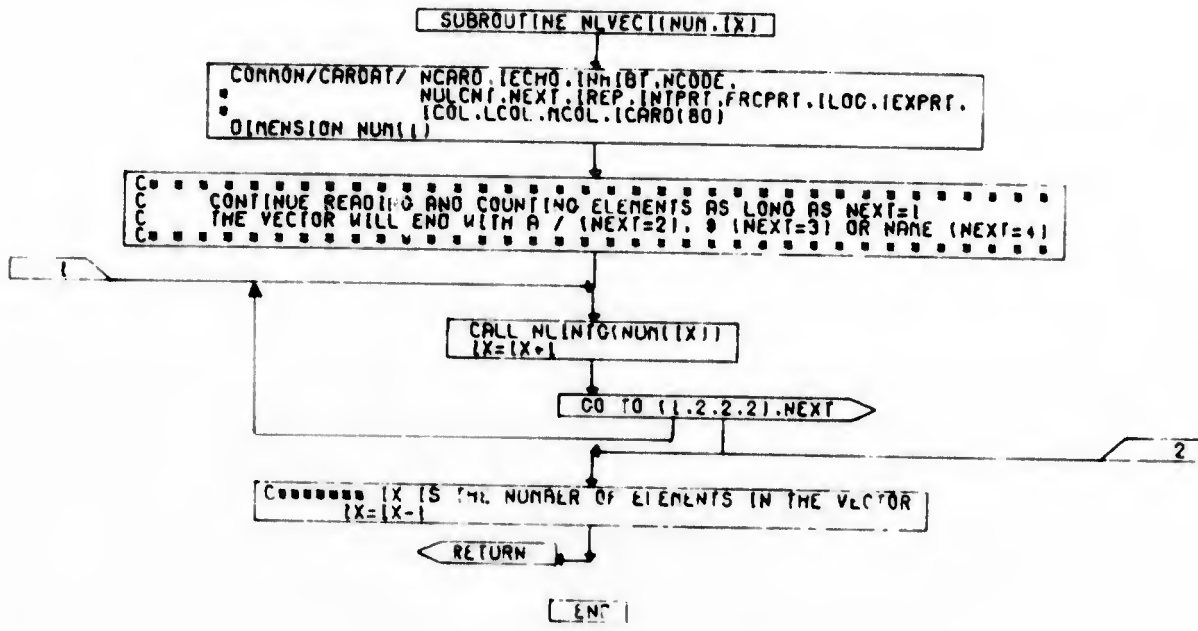
```
C * * * * * INHIBIT EXECUTION - COULD NOT IDENTIFY NAME * * * * *  
C * * * * *  
C * * * * * INHIBT=INHIBT+1  
C * * * * * WRITE(IPRINT,1001)  
C * * * * * 1001 FORMAT(42H PROGRAM COULD NOT IDENTIFY VARIABLE NAME. )  
C * * * * * WRITE(IPRINT,1000) ICDL,NCARD  
C * * * * * 1000 FORMAT(20H FATAL ERROR PRECEDES COLUMN.13,5H CARD.13/  
C * * * * * * 47H ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED)
```

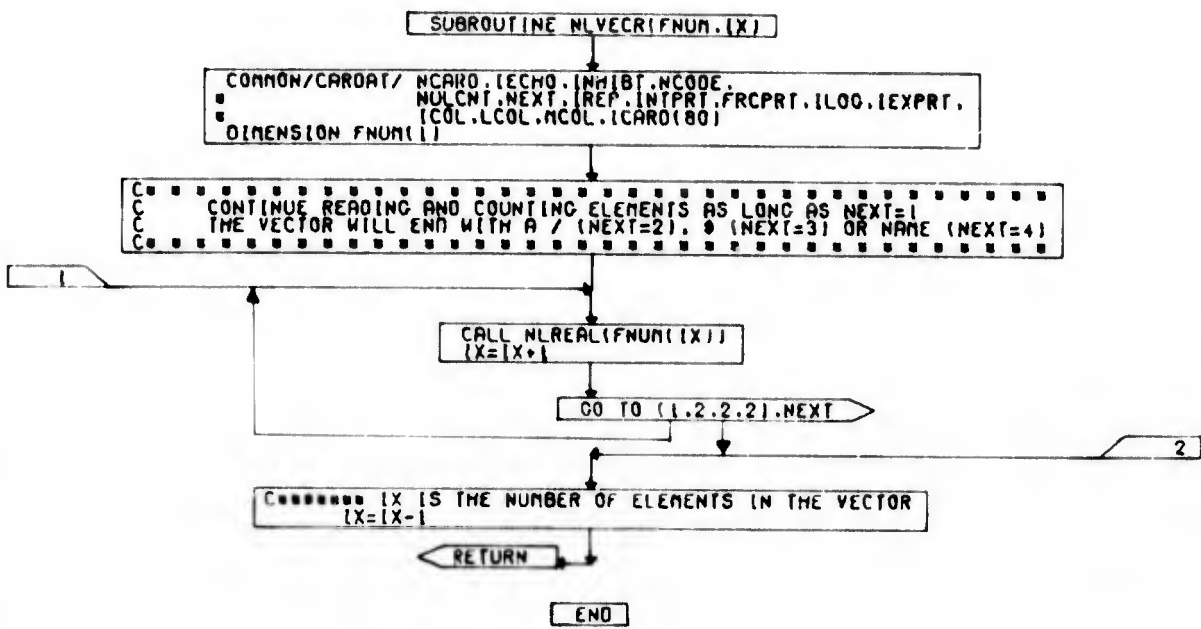
101

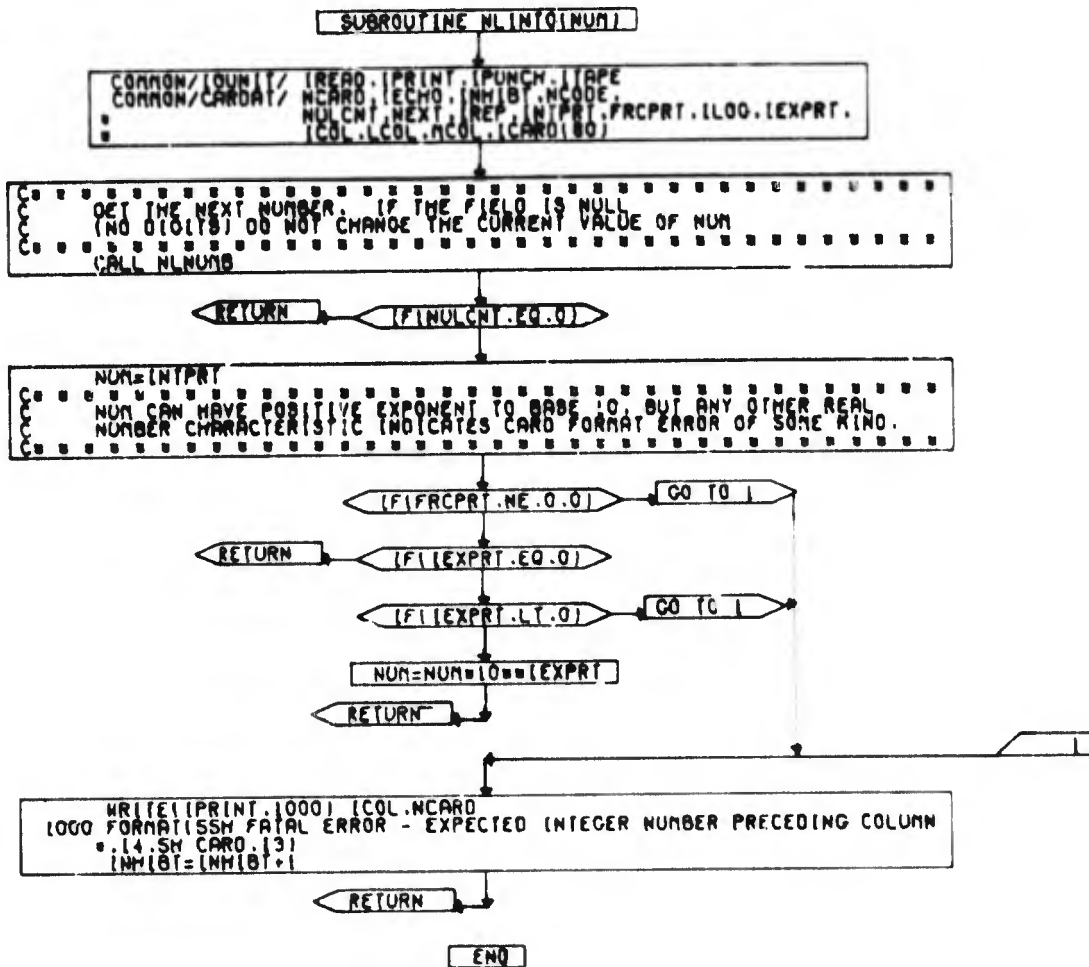
```
C * * * * * SKIP NUMERIC FIELDS UNTIL *COMMENT OR NEXT NAME FOUND * * * * *  
C * * * * *  
C * * * * * IF(IREP,GT,1) IREP=1  
C * * * * * CALL NLNUMB
```

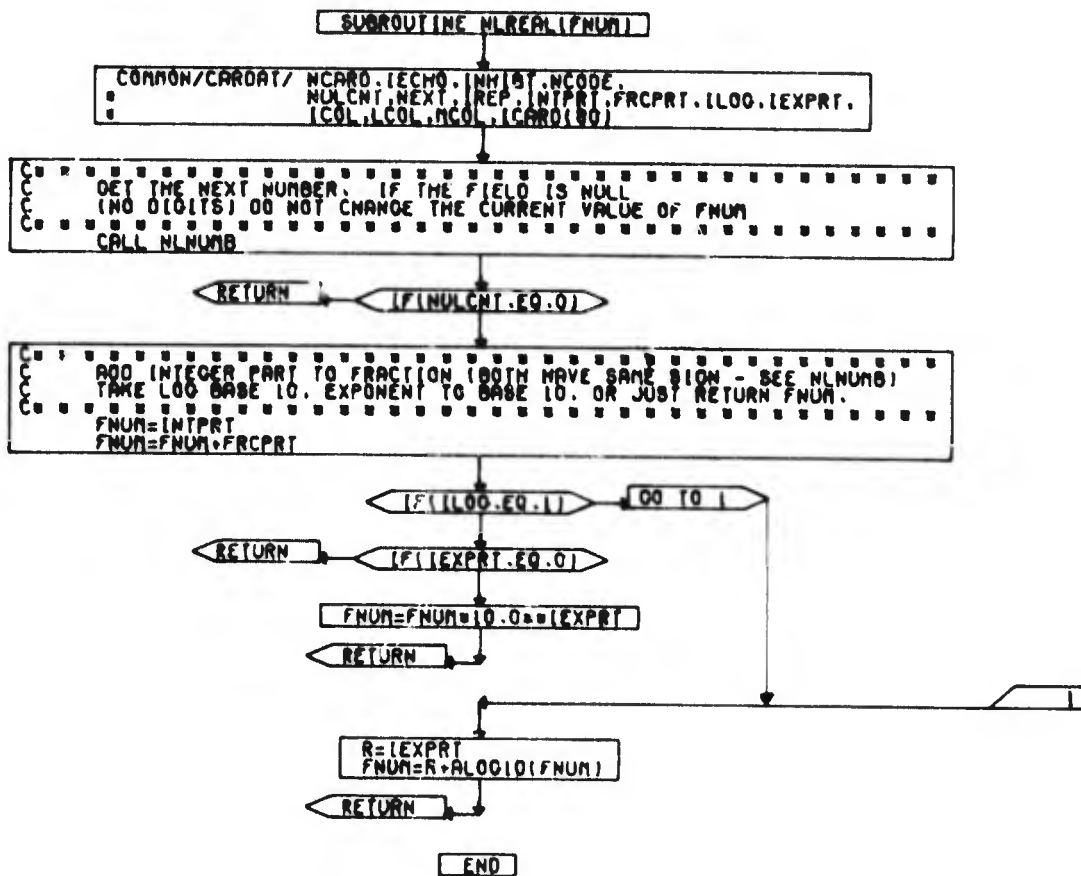
GO TO (101,10), 50,50),NXT

END









SUBROUTINE NLNUMB

```

COMMON/CARDAT/ NCARD,TECHO,INHIBY,NCODE,
NULCNT,NEXT,IREP,INTPRF,FRCPRT,ILOO,IEXPRT,
ICOL,LCOL,MCOL,ICARD(80)
COMMON /CHRLST/ KA,KB,MC,MD,ME,MF,KG,KH,KI,KJ,
KK,KL,KM,KN,KO,KP,KQ,KR,KS,KT,
KU,KV,KW,KX,KY,KZ,K1,K2,K3,
K4,K5,K6,K7,K8,K9,KCM,KPL,KRI,KR6,
KSL,KPR,KLP,KRP,KEG,KOL,KBL
DIMENSION MATCH(10)
EQUIVALENCE (MATCH,KO)
DIMENSION NU(3),NC(3),NB(3)
    
```

```

C . . . . . IF THE NUMERIC FIELD REPEAT OPTION IS IN EFFECT GO TO 100
C . . . . .
    
```

IF (IREP.NE.0) GO TO 100

```

C . . . . . TURN REPEAT AND LOOIO OPTIONS OFF
IREP=1
ILOO=0
    
```

```

C . . . . . ASSUMES EVERY NUMERIC FIELD CONTAINS 1) INTEGER PART,
2) FRACTION PART AND 3) EXPONENT PART
C . . . . .
DO 1 I=1,3
NU(I)=0
NC(I)=0
NB(I)=1
1 CONTINUE
C . . . . . BEGIN INTEGER PART
IPART=1
    
```

GO TO 3

```

2
C . . . . . INCREMENT TO NEXT COLUMN
ICOL=ICOL+1
    
```

IF (ICOL.GT.LCOL) GO TO 20

```

ISYM=ICARD(ICOL)
C . . . . . IGNORE BLANKS AND + SIGN
    
```

IF (ISYM.EQ.KBL)
IF (ISYM.EQ.KPL)
GO TO 7

```

4
C . . . . . LOOK FOR 010176. IF FOUND CONVERT TO INTEGER NUMBER
DO 5 I=1,10
    
```

IF (ISYM.NE.MATCH(I)) GO TO 6

```

NU(IPART)=NU(IPART)+10*(I-1)
NC(IPART)=NC(IPART)+1
    
```

6 ICOL=ICOL+1

IF (ICOL.GT.LCOL) GO TO 20

ISYM=ICARD(ICOL)

IF (ISYM.EQ.KBL)
GO TO 6

GO TO 4

5 CONTINUE

PG 2

\*\*\*\*\* LOOK FOR . , / L OR -

IF (ISYM.EQ.KAS) GO TO 10

IF (ISYM.EQ.KPR) GO TO 11

IF (ISYM.EQ.KL) GO TO 12

IF (ISYM.EQ.KE) GO TO 13

IF (ISYM.EQ.KM) GO TO 14

\*\*\*\*\* LOOK FOR . / . & OR NEXT INPUT NAME  
NEXTT=4  
IF (ISYM.EQ.KCM) NEXTT=1  
IF (ISYM.EQ.KSL) NEXTT=2  
IF (ISYM.EQ.KOL) NEXTT=3

GO TO (22,22,30,25),NEXTT

10

\*\*\*\*\* FOUND . SET REPEAT OPTION  
IF (NUI) (NE.O) (REP=NUI)  
NUI=0  
NCI=0

GO TO 2

11

\*\*\*\*\* FOUND . BEGIN FRACTION PART  
IPART=2

GO TO 2

12

\*\*\*\*\* FOUND L SET LOOP OPTION  
ILOO=1

13

\*\*\*\*\* FOUND E OR L BEGIN EXPONENT PART  
IPART=3  
\*\*\*\*\* SAVE CURRENT COLUMN INDEX - AN INPUT NAME MAY BEGIN WITH E OR L  
ICOL=ICOL

GO TO 2

14

\*\*\*\*\* FOUND - SIGN  
NSI(IPART)=-1

GO TO 2

\*\*\*\*\* THIS SECTION OF CODE DETERMINES HOW NUMERIC FIELD TERMINATES  
TO DEFINE VARIABLE NEXT  
NEXT=1 TWO NUMBERS SEPARATED BY COMMA (,) OR END-OF-CARD  
NEXT=2 TWO NUMBERS SEPARATED BY GLASH (/)  
NEXT=3 NUMBER FOLLOWED BY COMMENT  
NEXT=4 NUMBER FOLLOWED BY INPUT NAME OR SPECIAL SYMBOL

18

IF (NEXTT.NE.O) GO TO 30

NEXTT=1

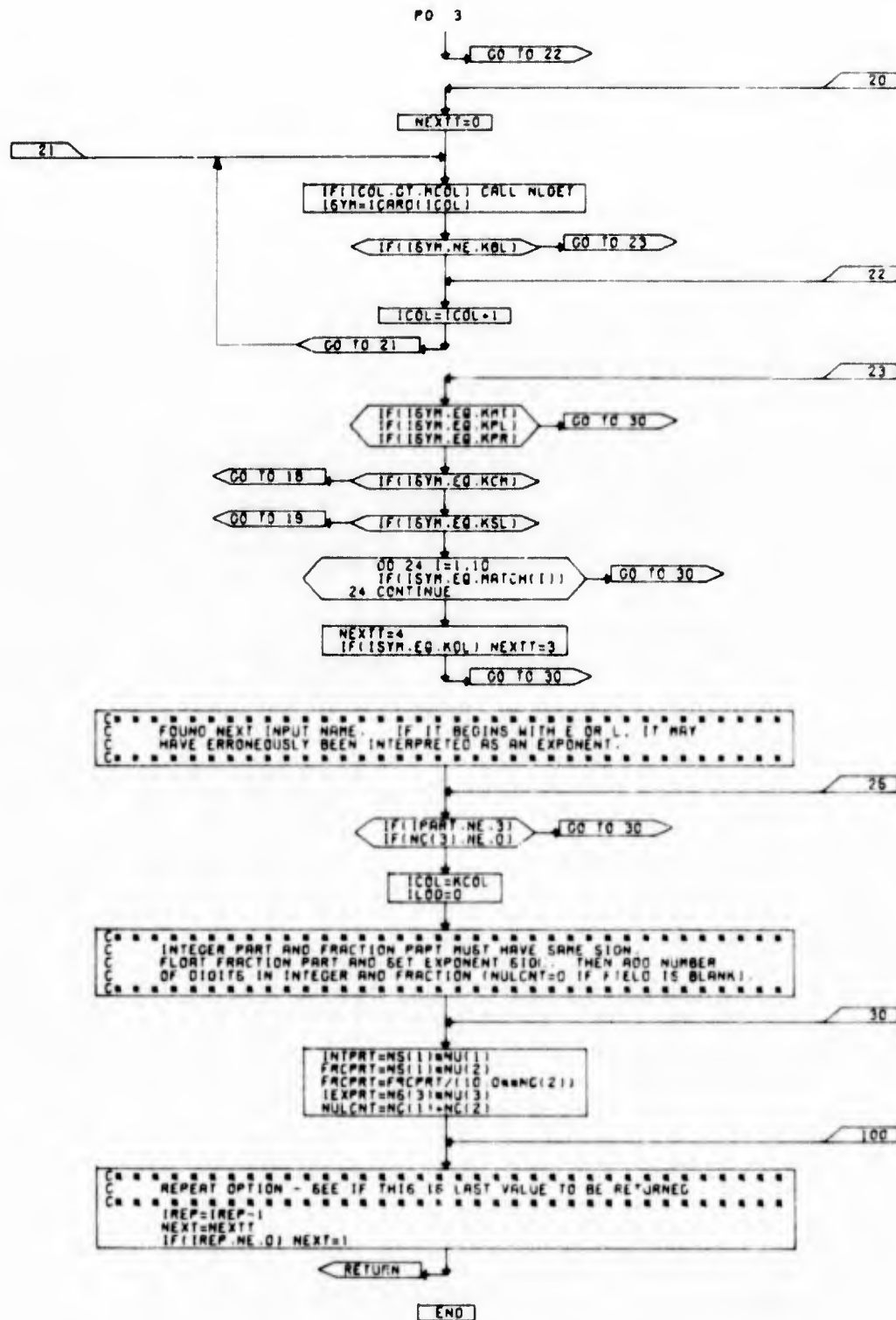
GO TO 22

19

IF (NEXTT.NE.O) GO TO 30

NEXTT=2

PG 3

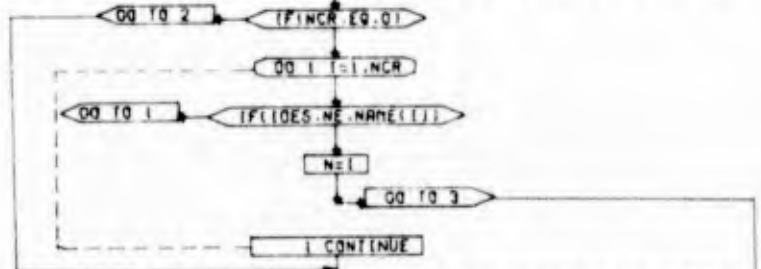


SUBROUTINE COMPNT(10ES)

```

COMMON/IOINIT/ IREAD,IPRINT,IPUNCH,ITAPE
COMMON/CARDAT/ NCRAD,IECH3,INHIBT,NCODE,
  MULCNT,NEXT,IREP,INTPRY,FRCPRT,ILOG,IEXPRT,
  ICOL,LCOL,NCOL,CARD(180)
COMMON /CHRLST/ KA,KB,MC,MD,ME,MF,MO,MH,MI,MJ,
  KK,ML,MN,MN,KO,MP,MQ,MR,MS,MT,
  KU,MV,MW,MX,MY,MZ,NO,NL,M2,M3,
  N4,NS,NG,N7,NB,N8,N9,NEN,NPL,NM1,NAS,
  NSL,NPR,NLP,NAP,NEQ,NOL,NBL
COMMON/CE01 / NBYN,NBYN,NODE(1)
COMMON/CNC1 / NCR,NSYS,NNARE,NAME(1)
COMMON/CNC2 / INSLC(2,1)
COMMON/CC01 / NCINS,NCINS,INSCI(4,1)
COMMON/CC02 / CDEF(3,1)
COMMON/CC03 / FCY(16,1)
COMMON/CC04 / FCX(16,1)
COMMON/CC05 / FC(16,1)
COMMON/CC06 / FU(16,1)
COMMON/CC07 / IBO(18,1)
COMMON/CN01 / NNINS,NNINS,INSNET(4,1)
  
```

1 IF THIS IS THE FIRST COMPONENT READ, GO TO 2  
 IF NOT, SEE IF THIS COMPONENT WAS READ PREVIOUSLY



2 PREPARE FOR NEW COMPONENT  
 NCR=NCR+1  
 N=NCR  
 NAME(N)=10ES

GO TO 6

3 COMPONENT WAS READ BEFORE SO THIS IS A REPLACEMENT OR DELETION.  
 THE FRAGILITY CURVE IS LOCATED BY NCR, BUT TRANSFER FUNCTIONS  
 ARE STORED IN INSC(1,18) THROUGH INSC(1,1EE).  
 IB=INSLC(1,N)  
 IS=INSLC(2,N)+1  
 IE=IB+IS  
 IEE=IE-1

4 REMOVE PREVIOUS REFERENCES TO THOSE ENVIRONMENTS WHICH MAY NO  
 LONGER BE REQUIRED. (MODE IS USED BY SUBROUTINE SELECT)  
 DO 104 I=IB,IEE  
 IOP=INSC(1,I)  
 GO TO (104,104,103,104,104,103,104,104,103,104,  
 104,104,103,104,  
 103 IOP=2\*NBYN+INSC(3,I)  
 NODE(IOP)=NODE(IOP)-1  
 104 CONTINUE

```

*****
NOW REMOVE THE OLD TRANSFER FUNCTION AND RELOCATE OTHER FUNCTIONS.
THIS MAKES STORAGE AVAILABLE AT THE END OF INSC AND COEF FOR
NEW FUNCTIONS.
*****
DO 4 I=1E,NCINS
  INSC(I,IB)=INSC(I,1)
  INSC(I,2)=INSC(I,2)
  INSC(I,3)=INSC(I,3)
  INSC(I,4)=INSC(I,4)
  COEF(I,1)=COEF(I,1)
  COEF(I,2)=COEF(I,2)
  COEF(I,3)=COEF(I,3)
  IB=IB+1
4 CONTINUE
NCINS=NCINS-15
IB=INSLC(1,N)
DO 5 I=1,NCR
  IF(INSLC(1,I).GT.IB) INSLC(1,I)=INSLC(1,I)-15
5 CONTINUE

```

6

```

*****
BEGIN STORING NEW COMPONENT
INSLC IS STARTING ADDRESS FOR NEW TRANSFER FUNCTION
IB(1)=0 INDICATES THAT NO FRAILTY POINTS HAVE UNCERTAINTY
IF B IS FOUND, GO TO 101 WHICH WILL DELETE THE COMPONENT
*****
INSLC(1,N)=NCINS+1
IB(1,N)=0
CALL NLINT(15)

```

GO TO (7,7,101,7).NEXT

```

*****
NOTE THAT THE LLOO OPTION IS TURNED OFF BUT IS TURNED ON IF
FIRST FRAILTY PARAMETER CONTAINS A L (E.G., 6L0)
ENTER FRAILTY CURVE -
FCY PROBABILITY FOR ORDINATE
FCX BEST ESTIMATE FOR NO UNCERTAINTY
FC BEST ESTIMATE WITH UNCERTAINTY
FU WIDTH OF UNCERTAINTY
IBO DISTRIBUTION SHAPE FOR UNCERTAINTY
IBO(7) NUMBER OF FRAILTY POINTS (1 THROUGH 6)
*****
7 CONTINUE
LLOO=0
I=0

```

10

```

I=I+1
CALL NLREAL(FCY(I,N))
CALL NLREAL(FL)
IF(LLOO.EQ.1) LLOO=1

```

GO TO (11,12,12,12).NEXT

```

11 CALL NLREAL(FB)
CALL NLREAL(FUP)
FUL(N)=FUP-FL
IBO(N)=(FB-FL)/FU(N)*10.0+0.5
FC(I,N)=FB
IF(IBO(N).EQ.0) IBO(N)=1
IF(IBO(N).EQ.10) IBO(N)=9
IBO(N)=1

```

GO TO (10,10,13,13).NEXT

12

```

IBO(I,N)=0
FC(I,N)=FL
FCX(I,N)=FL

```

GO TO (10,10,13,13).NEXT

13

```

IBO(7,N)=1

```

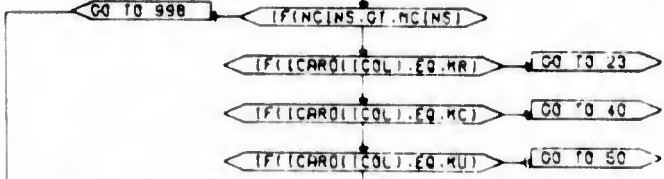
PG 3

20

```

C ..... ENTER A TRANSFER FUNCTION INSTRUCTION (LOOK FOR R, CAN,
C ..... OR USER - EACH INSTRUCTION IS ENCLOSED WITH SLASHES (/))
C .....
C ..... NCINS=NCINS+1

```



```

WRITE(PRINT,1001)
1001 FORMAT(39H COULD NOT RECOGNIZE LOCAL ENV RELATION)
GO TO 998

```

```

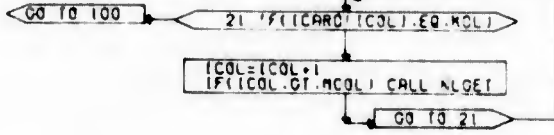
998 WRITE(PRINT,1002)
1002 FORMAT(42H TOO MANY ENV RELATIONSHIPS FOR INSC TABLE)
NCINS=NCINS-1

```

```

999 WRITE(PRINT,1000) ICOL,NCARD
1000 FORMAT(20H FATAL ERROR PRECEDES COLUMN,13.5H CARD,(3/
      47H ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED)
INMIBT=INMIBT-1

```

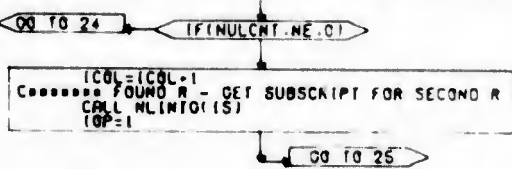


23

```

C ..... FOUND R - TRANSFER FUNCTION INSTRUCTION
C .....
C ..... IF=1.0
C ..... IF((OO.EQ.1) IF=0.0
C ..... C=IF
C ..... IFU=IF
C ..... ICOL=ICOL+1
C ..... GET SUBSCRIPT FOR FIRST R
C ..... CALL NLINTO(INSC(2,NCINS))
C ..... ICOL=ICOL+1
C ..... LOOK FOR SECOND R (E.O., R2=R1) OR ENVIRONMENT REFERENCE (R1=S)
C ..... CALL NLINTO(15)

```

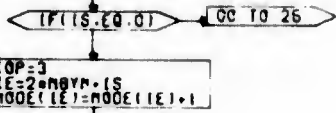


24

```

C ..... FOUND ENV. - REQUEST SELECT TO STORE ENV. FOR MONTE CARLO
C ..... PROCESS, UNLESS IS=0 (OVERPRESSURE ENVIRONMENT)
TOP=2

```



PG 4

C\*\*\*\*\* SET OPERATION FLAG (IOP) DEPENDING ON WHICH PARAMETERS (TF,TFU,  
C\*\*\*\*\* C) ARE FOUND. IOP IS MODIFIED FOR LOGIO OPERATIONS.  
IF(LL00.EQ.1) IOP=IOP\*3  
INSC(3,NCINS)=15

GO TO (26,27,27,27).NEXT

26 CALL NLREAL(TF)

GO TO (28,27,27,27).NEXT

27 INSC(1,NCINS)=IOP  
COEF(3,NCINS)=TF

GO TO (20,20,100,20).NEXT

28 CALL NLREAL(TFU)

GO TO (29,30,30,30).NEXT

29 CALL NLREAL(C)

30 IF(LL00.EQ.1) GO TO 31

IF(ABS(TFU-1.0).GT.0.00005) IOP=IOP\*6

GO TO 32

31 IF(TFU.EQ.0.0) IOP=IOP\*6

32 INSC(1,NCINS)=IOP

C\*\*\*\*\* SET UP TRANSFER FUNCTION COEFFICIENTS DEPENDING ON IOP.  
GO TO (33,33,33,34,34,34,35,35,35,36,36,36).IOP

33 COEF(3,NCINS)=TF\*C

GO TO (20,20,100,20).NEXT

34 COEF(3,NCINS)=TF\*C

GO TO (20,20,100,20).NEXT

35 COEF(1,NCINS)=C\*TF/TFU  
COEF(2,NCINS)=C\*TF-iTFU-COEF(1,NCINS)  
INSC(4,NCINS)=10.0/(1.0\*TFU)\*0.5

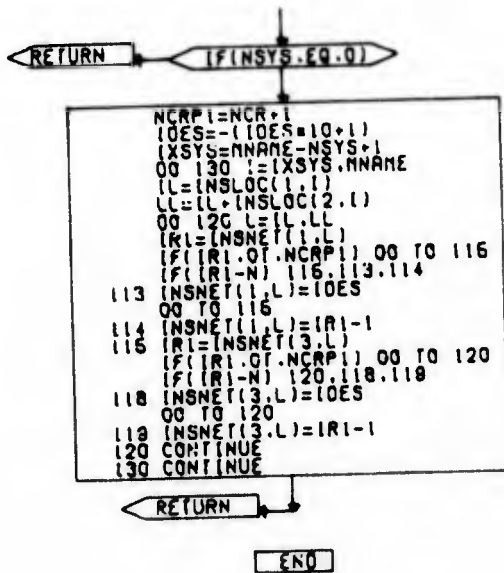
GO TO (20,20,100,20).NEXT

36 COEF(2,NCINS)=SQRT((TF+TFU)/(TF-TFU))  
COEF(1,NCINS)=C\*TF-COEF(2,NCINS)  
COEF(2,NCINS)=2.0\*COEF(2,NCINS)  
INSC(4,NCINS)=5

GO TO (20,20,100,20).NEXT



PO 6



SUBROUTINE BOOLE(IDES)

```

COMMON /WORKEY/ (KEY,LKEY,NKEY,IPAD,KEY(1))
COMMON /OUNIT/ (READ,PRINT,IPUNCH,ITAPE)
COMMON /CARDAT/ (NCARD,TECHO,INHIBT,NCODE,
                NUCNT,NEXT,REP,INTPRT,FRCPRT,ILOO,EXPRT,
                ICOL,LCOL,NCOL,ICARD(80))
COMMON /CHRLST/ (KA,KB, KC, KD, KE, KF, KG, KH, KI, KJ,
                KK, KL, KM, KN, KO, KP, KQ, KR, KS, KT,
                KU, KV, KW, KX, KY, KZ, KA, KB, KC, KD, KE,
                KF, KG, KH, KI, KJ, KK, KL, KM, KN, KO,
                KP, KQ, KR, KS, KT, KU, KV, KW, KX, KY, KZ,
                KSL, KPR, KLP, KRP, KEQ, KOL, KBL)
DIMENSION MATCH(10)
EQUIVALENCE (MATCH,KO)
COMMON /CNC1/ (NCR,NSYS,NNAME,NAME(1))
COMMON /CNC2/ (INSLOC(2,1))
COMMON /CNC3/ (NN(NS,NN(NS,INSNET(4,1))
    
```

```

C *****
C ***** COMPONENT NAMES ARE STORED BEGINNING IN NAME(1) TO NAME(NCR), BUT
C ***** SS AND SYS NAMES ARE STORED IN REVERSE ORDER BEGINNING IN THE LAST
C ***** WORD NAME(NNAME)
C ***** IF THIS IS THE FIRST NETWORK GO TO 2
C ***** IF NOT, SEE IF THIS NETWORK WAS READ PREVIOUSLY
C ***** (E=NNAME-NSYS+1)
    
```

IF(NSYS.EQ.0) GO TO 2

DO 1 I=1,E,NNAME

IF(IDES.NE.NAME(I)) GO TO 1

N=I

GO TO 3

1 CONTINUE

```

C *****
C ***** PREPARE TO ENTER NEW NETWORK
C *****
C ***** IF=I-1
C ***** NSYS=NSYS+1
C ***** N=I
C ***** NAME(N)=IDES
    
```

GO TO 6

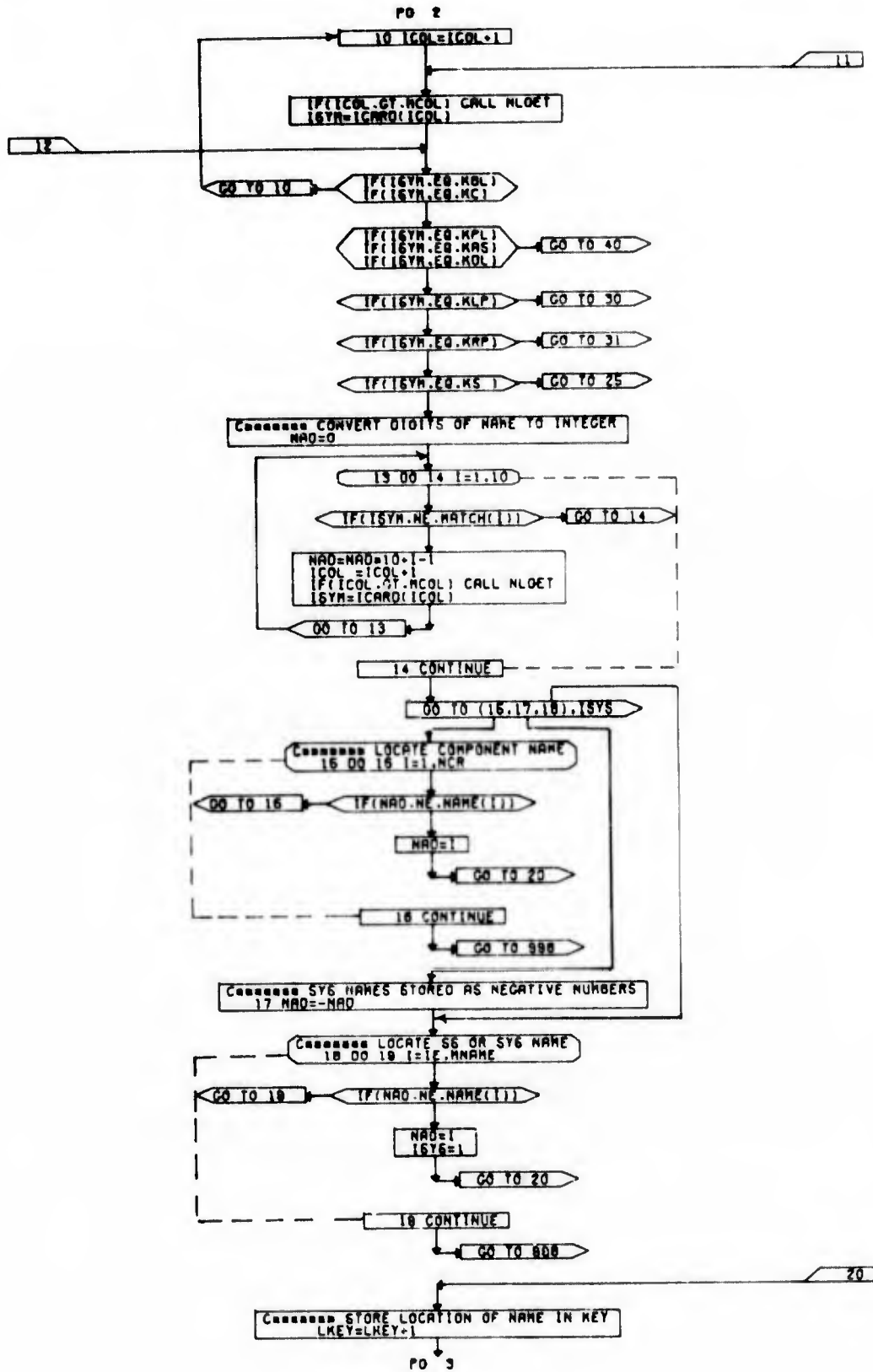
```

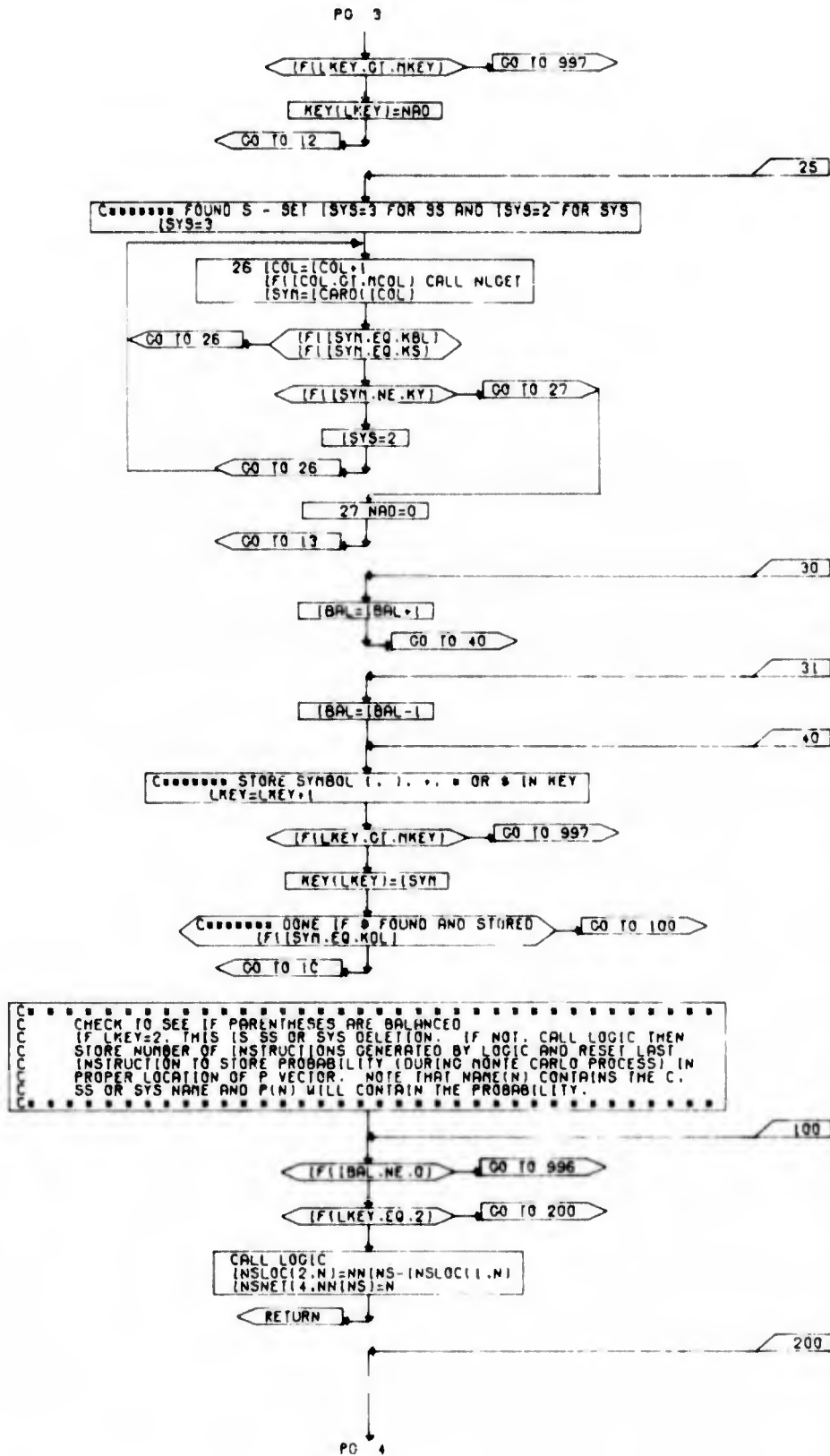
C *****
C ***** NETWORK WAS READ BEFORE SO THIS IS A REPLACEMENT OR DELETION.
C ***** REMOVE OLD NETWORK TO MAKE SPACE AVAILABLE FOR NEW EQUATIONS.
C *****
C ***** (B=INSLOC(I,N))
C ***** (S=INSLOC(I,N)+1)
C ***** (E=B+S)
C ***** DO 4 I=1,NN(NS)
C ***** (INSNET(I,B)=INSNET(I,1))
C ***** (INSNET(I,2,B)=INSNET(I,2,1))
C ***** (INSNET(I,3,B)=INSNET(I,3,1))
C ***** (INSNET(I,4,B)=INSNET(I,4,1))
C ***** (B=B+1)
C ***** 4 CONTINUE
C ***** (NN(NS)=NN(NS)-1S)
C ***** (B=INSLOC(I,N))
C ***** (E=NNAME-NSYS+1)
C ***** DO 5 I=1,E,NNAME
C ***** IF(INSLOC(I,1).GT.(B) INSLOC(I,1)=INSLOC(I,1)-1S)
C ***** 5 CONTINUE
    
```

```

C *****
C ***** INITIALIZE KEY VECTOR AND BEGIN MOVING SYMBOLS LINE 1, 2, 3, 4
C ***** AND 5 FROM THE CARD TO KEY. EACH TIME C, SS OR SYS NAME IS FOUND
C ***** ON THE CARD, LOCATE THE NAME IN THE NAME VECTOR OF COMMON /CNC1/.
C ***** STORE THE LOCATION IN KEY INSTEAD OF THE NAME.
C *****
C ***** (INSLOC(I,N)=NN(NS)+1)
C ***** (LKEY=I)
C ***** (KEY(LKEY)=KLP)
C ***** (SYS=1)
C ***** (BAL=0)
    
```

GO TO 11





```

C C C C C
DELETE SS OR SYS EQUATION
I=N-1
IB=NNAME-NSYS-1
DO 201 I=IB,IE
NAME(I)=NAME(N-1)
INSLOC(1,N)=INSLOC(1,N-1)
INSLOC(2,N)=INSLOC(2,N-1)
N=N-1
201 CONTINUE
    
```

```

C C C C C
SCAN OTHER EQUATIONS FOR REFERENCES TO DELETED SS OR SYS.
REPLACE REFERENCE (FOR NETCK) WITH MODIFIED NAME, IOES. NOTE
THAT I=1 WAS USED FOR COMPONENTS, I=2 FOR SS AND I=3 FOR SYS
IN EQUATION IOES=(IOES*10+I)
I=2
IF (IOES.LT.0) I=3
IOES=(ABS(IOES)*10+I)
IXSYS=IB+I
IE=IE+I
DO 300 L=(XSYS,IE)
LL=INSLOC(1,L)
LL=LL+INSLOC(2,L)
DO 250 L=LL,LL
IRI=INSNET(1,LL)
IF (IRI.LT.10) GO TO 210
IF (IRI-IE) 209,208,210
208 INSNET(1,LL)=IOES
GO TO 210
209 INSNET(1,LL)=IRI+1
210 IRI=INSNET(3,LL)
IF (IRI.LT.10) GO TO 220
IF (IRI-IE) 219,218,220
218 INSNET(3,LL)=IOES
GO TO 220
219 INSNET(3,LL)=IRI+1
220 FIL.NE.LL) GO TO 250
IRI=INSNET(4,LL)
IF (IRI.LT.10) GO TO 250
IF (IRI-IE) 229,228,250
228 INSNET(4,LL)=IOES
GO TO 250
229 INSNET(4,LL)=IRI+1
250 CONTINUE
300 CONTINUE
NSYS=NSYS-1
    
```

RETURN

996  
 WRITE(PRINT,1001)  
 1001 FORMAT(7H PARENTHESIS DO NOT BALANCE)  
 WRITE(PRINT,1000) (COL,NCARD  
 (NMIBT=NMIBT+1)  
 RETURN

997  
 WRITE(PRINT,1002)  
 1002 FORMAT(44H EQUATION TOO LONG FOR KEY - DIVIDE EQUATION)  
 GO TO 999

998  
 WRITE(PRINT,1003)  
 1003 FORMAT(31H NAME NOT FOUND OR SYMBOL ERROR)

999 WRITE(PRINT,1000) (COL,NCARD  
 1000 FORMAT(20H FATAL ERROR PRECEDES COLUMN,13,5H CARD,13/  
 \* 47\* ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED)  
 (NMIBT=NMIBT+1)

1010 IF (CARD(COL).EQ.NO.) GO TO 1020  
 COL=COL+1  
 IF (COL.GT.NCOL) CALL NLGET  
 GO TO 1010

1020 CALL NLGET

RETURN

END

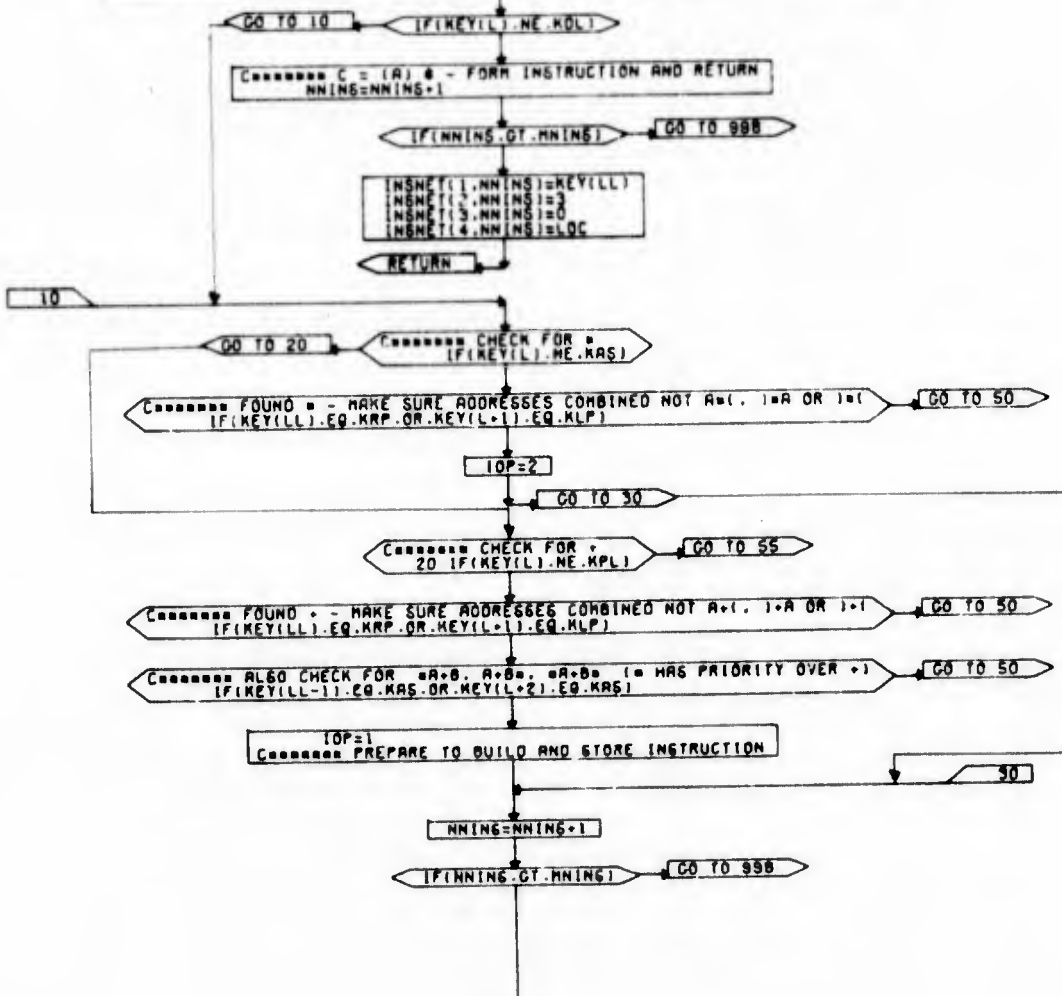
SUBROUTINE LOGIC

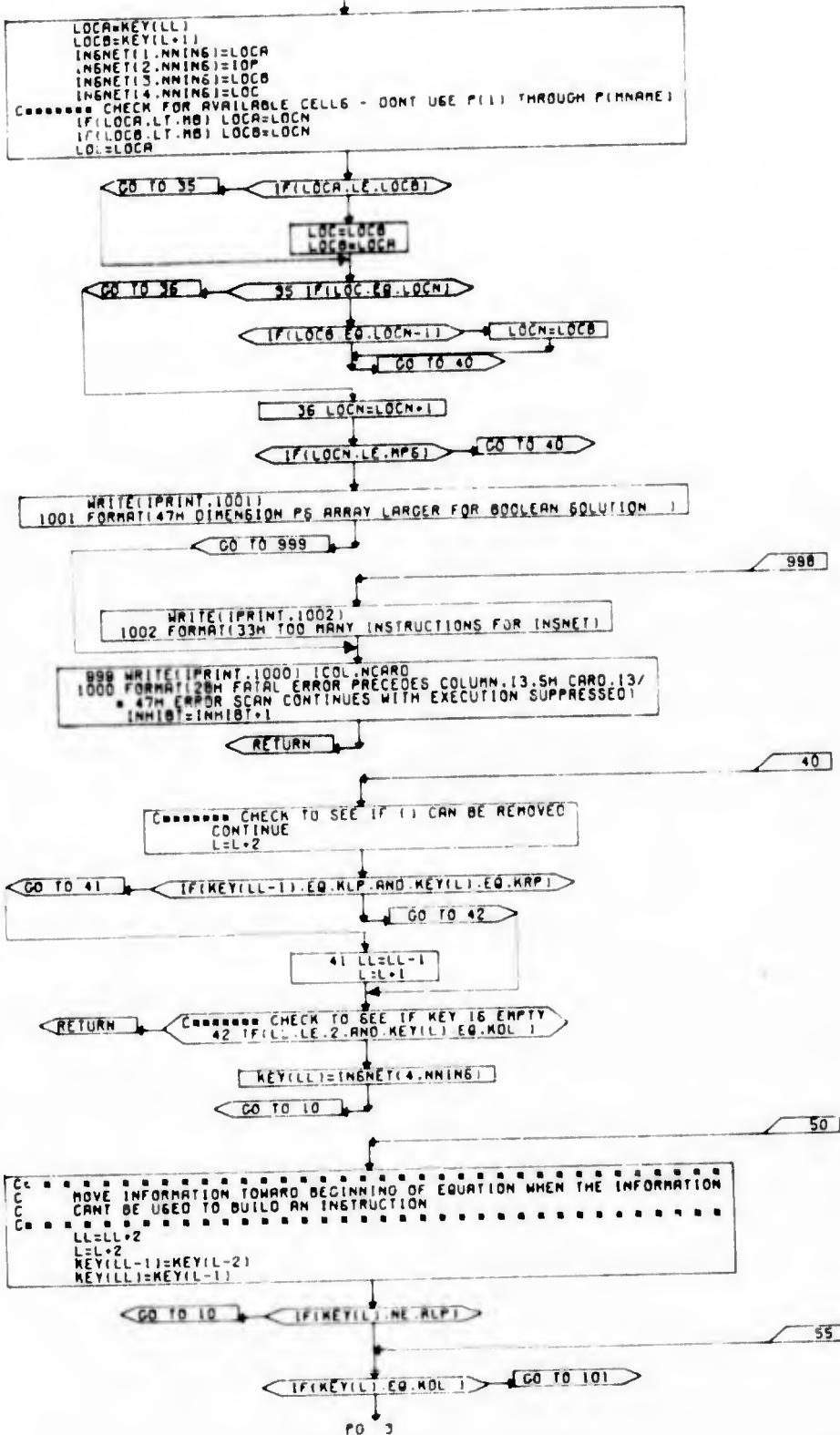
```

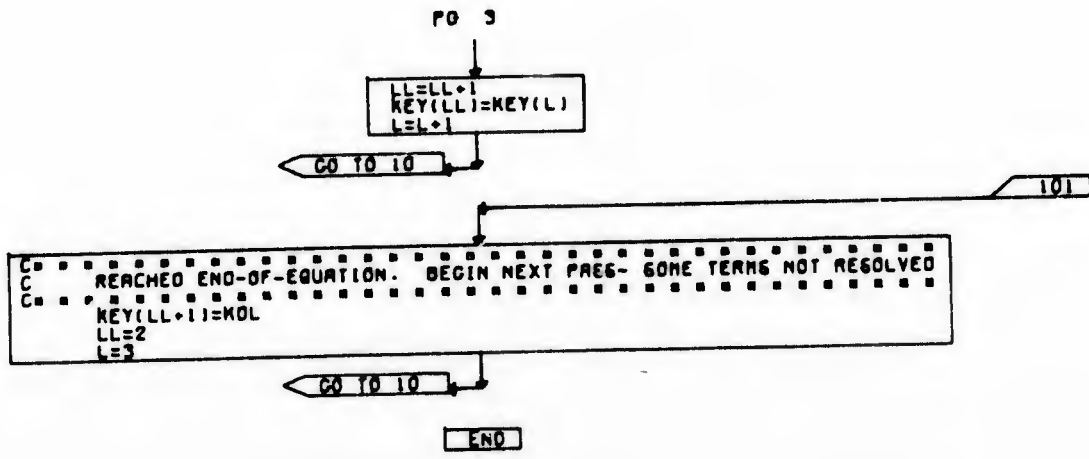
COMMON/WORKKEY/ IKEY,LKEY,MKEY,IPAO1,KEY(1)
COMMON/IDUNIT/ IREAD,IPRINT,IPUNCH,ITAPE
COMMON/CARDAT/ NCARD,IECMD,INHIBT,NCODE,
               NULCNT,NEXT,IREF,INTPT,FRCPRT,ILOO,IEXPRT,
               ICOL,LCOL,NCOL,ICARD(80)
COMMON /CHARLET/ KA,KB,KC,KD,KE,KF,KG,KH,KI,KJ,
                 KK,KL,KM,KN,KO,KP,KQ,KR,KS,KT,
                 KU,KV,KW,KX,KY,KZ,K1,K2,K3,K4,
                 K5,K6,K7,K8,K9,K0,KCH,KPL,KRI,KAS,
                 KSL,KPR,KLP,KRP,KEQ,KDL,KBL
COMMON/END1 / NNING,NNING,INSNET(4,1)
COMMON/END1 / NCR,NBYE,NAME,NAME(1)
COMMON/END3 / NPG,IPAO2,P(1)
    
```

```

*****
***** LOGIC SCANS THE EQUATION STORED IN KEY TO RESOLVE PARENTHESES
***** AND FORMS A LOGIC SEQUENCE FROM THREE TYPES OF INSTRUCTIONS.
***** -BOOLE- -PROBABILITY- -INSTRUCTION (ADDR IS ADDRESS)-
***** C = A + B C=(1-A)(1-B) ADDR A.TOP=1,ADDR B.STORE IN ADDR
***** C = A * B C=(A)(B) ADDR A.TOP=2,ADDR B.STORE IN ADDR
***** C = A C=(A) ADDR A.TOP=3,-----STORE IN ADDR
***** THE ADDRESSES ARE TO THE P VECTOR. P IS DIMENSIONED LONGER
***** THAN NAME SO THAT THE EXCESS CELLS CAN BE USED AS SCRATCH PAD.
*****
L=3
LL=2
AB=NAME+1
LOC=AB
LOCN=LOC+1
    
```







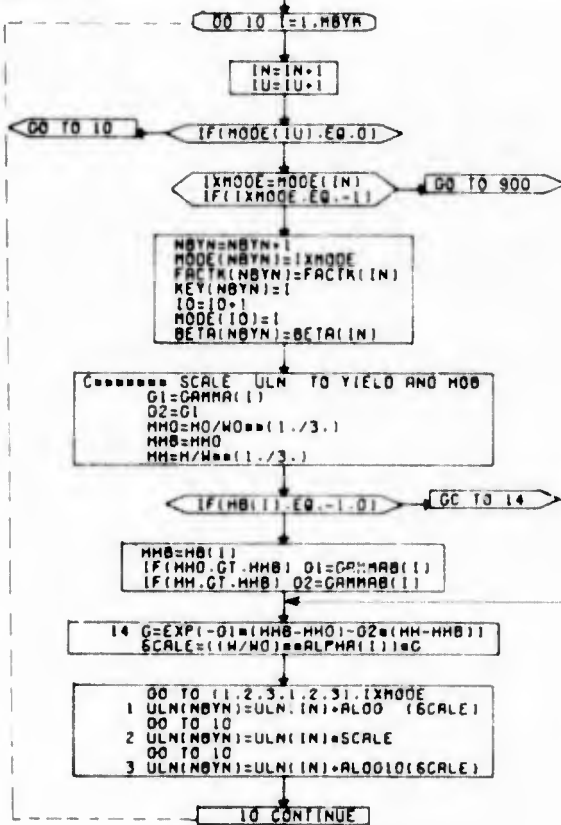
SUBROUTINE SELECT

```

COMMON/MORKEY/ IKEY,LKEY,MKEY (PA0),KEY(1)
COMMON/IOUNIT/ IREAD,IPRINT,IPUNCH,ITAPE
COMMON/CARDAT/ NCARD,ICMD,IMHBT,NCODE,
* NULCNT,NEXT,IREP,INTPRT,FRCPRT,ILOD,EXPT,
* ICOL,LCOL,MCOL,ICARD(B0)
COMMON/CE01 / NBYN,NBYN,MODE(1)
COMMON/CE02 / ULN (1)
COMMON/CE03 / FACTK (1)
COMMON/CE04 / ALPHA (1)
COMMON/CE05 / BETA (1)
COMMON/CE06A/ S1066 (1)
COMMON/CE06B/ S1066(1)
COMMON/CE07C/ RHOCD(1)
COMMON/CE08 / GAMMA (1)
COMMON/CE08A/ MS (1)
COMMON/CE08B/ GAMMAB(1)
COMMON/CR02 / MO,W,MO,H,OPD,RO,R(1)
COMMON/CR03 / NR,NOP,OP(1)
    
```

```

C .....
C SELECT ONLY THOSE ENVIRONMENT PARAMETERS REQUIRED FOR PROBLEM
C .....
IN=NBYN
IU=2*NBYN
IO=3*NBYN
NBYN=0
    
```



```

C . . . . .
C . . . . . SELECT MATRIX ELEMENTS FOR REQUIRED ENVIRONMENTS . . . . .
C . . . . .
L=0
MBYN1=MBYM-1
DO 20 I=1,MBYM
J=KEY(I)-1
J=J+MBYM-(J=(J-1))/2
DO 20 I=1,MBYM
K=J+KEY(I)
L=L+1
RHOCOL(L)=RHOCOL(K)
S1065(L)=S1065(K)
20 CONTINUE
    
```

WMB=1000.

GO TO 35 IF(NOP.NE.0)

```

C . . . . .
C . . . . . CONVERT RANGES TO CORRESPONDING PRESSURES . . . . .
C . . . . .
DO 30 I=1,NR
CALL RTOP(R(I)/6.0761,OP(I),WMB,H)
IF(OP(I).LT.100.) WRITE(IPRINT,2000)
2000 FORMAT(37HWARNING - PRESSURE LESS THAN 100 PSI )
30 CONTINUE
    
```

RETURN

```

C . . . . .
C . . . . . CONVERT PRESSURES TO CORRESPONDING RANGES . . . . .
C . . . . .
35 DO 36 I=1,NR
CALL PTOR(R(I),OP(I),WMB,H)
IF(R(I).EQ.0.0) INHIBT=INHIBT+1
36 R(I)=R(I)/6.0761
    
```

RETURN

900

```

IN=IN-MBYM
WRITE(IPRINT,1000) IN
1000 FORMAT(12H ENVIRONMENT.13.34H IS MISSING. EXECUTION SUPPRESSED)
INHIBT=INHIBT+1
    
```

RETURN

END

SUBROUTINE RTOP(R,P,W,H)

```
C . . . . .  
C COMPUTES OVERPRESSURE AS A FUNCTION OF RANGE(K-FT), YIELD(KT)  
C AND MOB(K-FT) (BRODE CURVE)  
C . . . . .  
WRM=W/SQRT(R**2+H**2)**3  
RTWRM=SQRT(WRM)  
OPW=1.58*WRM+5.4*RTWRM+.0215  
P=(.002655*OPW/(1.+(.0001728+1.921E-9*OPW)*OPW)  
+.004218*(.0483+6.856E-6*OPW)*OPW)/(1.+(.00799*  
3.844E-6*OPW)*OPW)*2.)*OPW  
IF(R.LT.H) RETURN  
MOR2=(H/R)**2  
P=P*MOR2*(3.16*WRM+7.6367556*RTWRM+.0215)/(1.-MOR2)
```

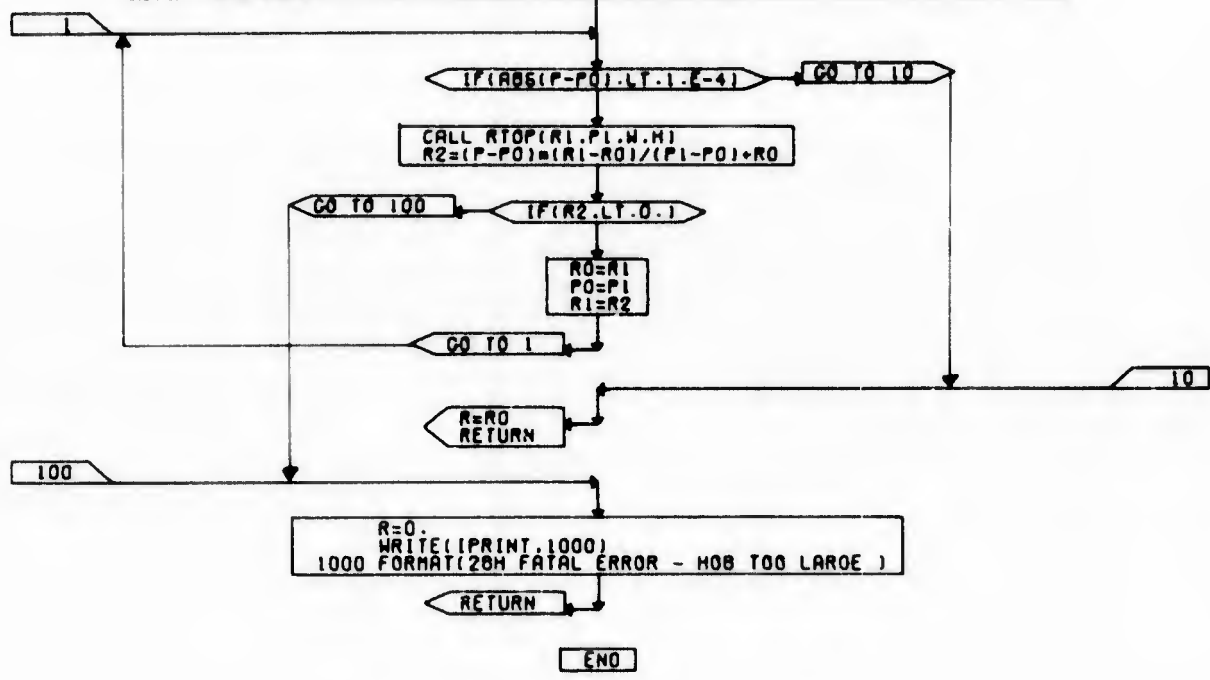
RETURN

END

SUBROUTINE PTOR(R,P,W,H)

COMMON/OUNIT/ IREAD,IPRINT,IPUNCH,ITAPE

```
C * * * * *  
C * COMPUTES RANGE(K-FT) AS A FUNCTION OF OVERPRESSURE(PG1) AND  
C * YIELD(KT) AT ZERO MOB(K-FT). THEN ITERATES TO FIND RANGE AT  
C * A NON-ZERO MOB(K-FT) (BRODE CURVE)  
C * * * * *  
C * IF(P.LT.100.) WRITE(IPRINT,2000)  
2000 FORMAT(37HWARNING - PRESSURE LESS THAN 100 PSI )  
R0=(2.*W*(3.16/(-5.4*SQRT(29.02+12*6.92*P)))**2)**(1./3.)  
CALL RTOP(R0,P0,W,H)  
R1=.9*R0
```



SUBROUTINE NETCK(INHIBT)

```

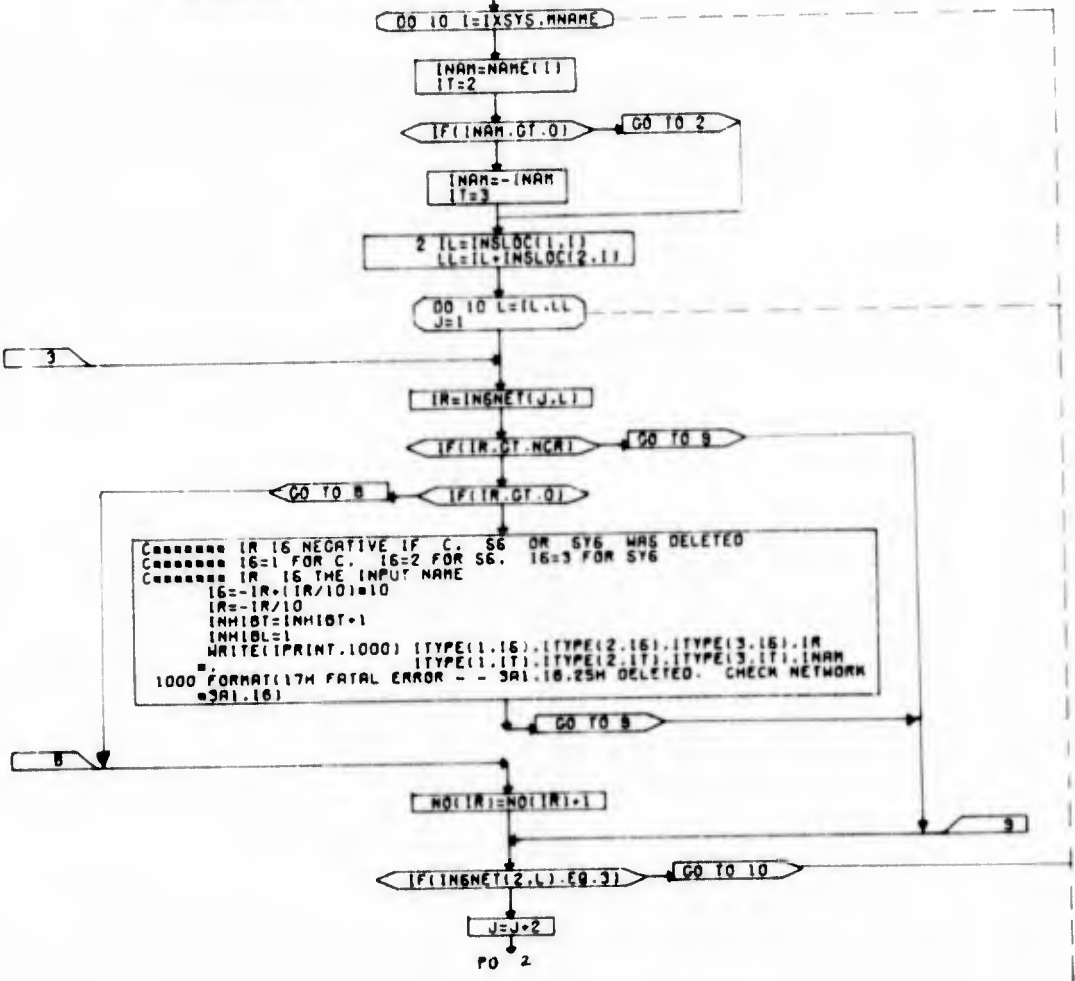
COMMON/IOUNIT/ IREAD, IPRINT, IPUNCH, ITAPE
COMMON /CHRL6T/  KA, KB, KC, KD, KE, AF, KG, KH, KI, KJ,
                KK, KL, KM, KN, KO, KP, KQ, KR, KS, KT,
                KU, KV, KW, KX, KY, KZ, KO, KI, KZ, KS,
                KA, KB, KC, KD, KE, KF, KG, KH, KI, KJ,
                KKL, KPR, KLP, KRP, KEQ, KOL, KBL
COMMON/CNC1 / NCR, NSYS, MNAME, NAME(1)
COMMON/CNC2 / INSLOC(2,1)
COMMON/CNC3 / NPS, IPR2, NOL1
COMMON/CNO1 / NNING, NNING, INNET(4,1)
DIMENSION ITYPE(3,3)
    
```

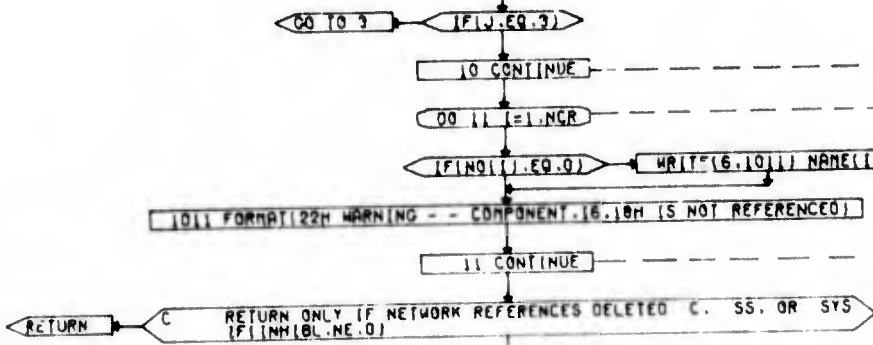
```

C .....
C ITYPE 16 USED FOR ERROR MESSAGES
C .....
ITYPE(1,1)=KBL
ITYPE(2,1)=KBL
ITYPE(3,1)=KC
ITYPE(1,2)=KBL
ITYPE(2,2)=KS
ITYPE(3,2)=KS
ITYPE(1,3)=KS
ITYPE(2,3)=KY
ITYPE(3,3)=KS
    
```

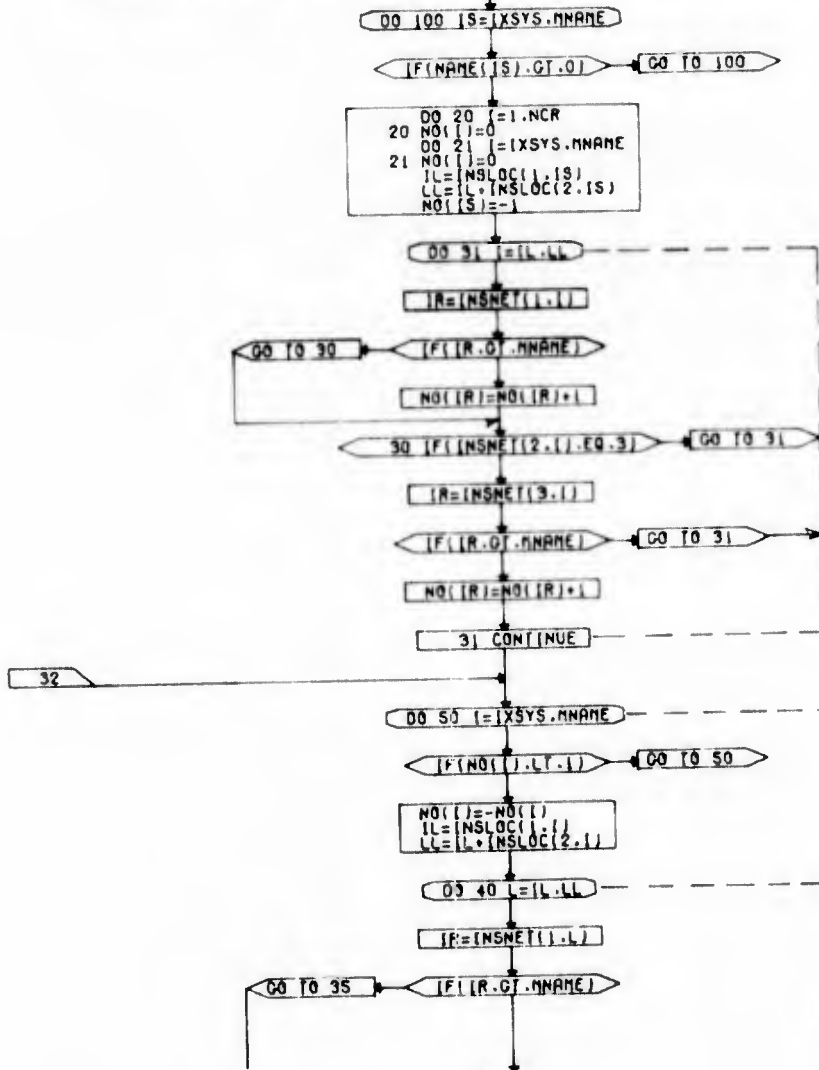
```

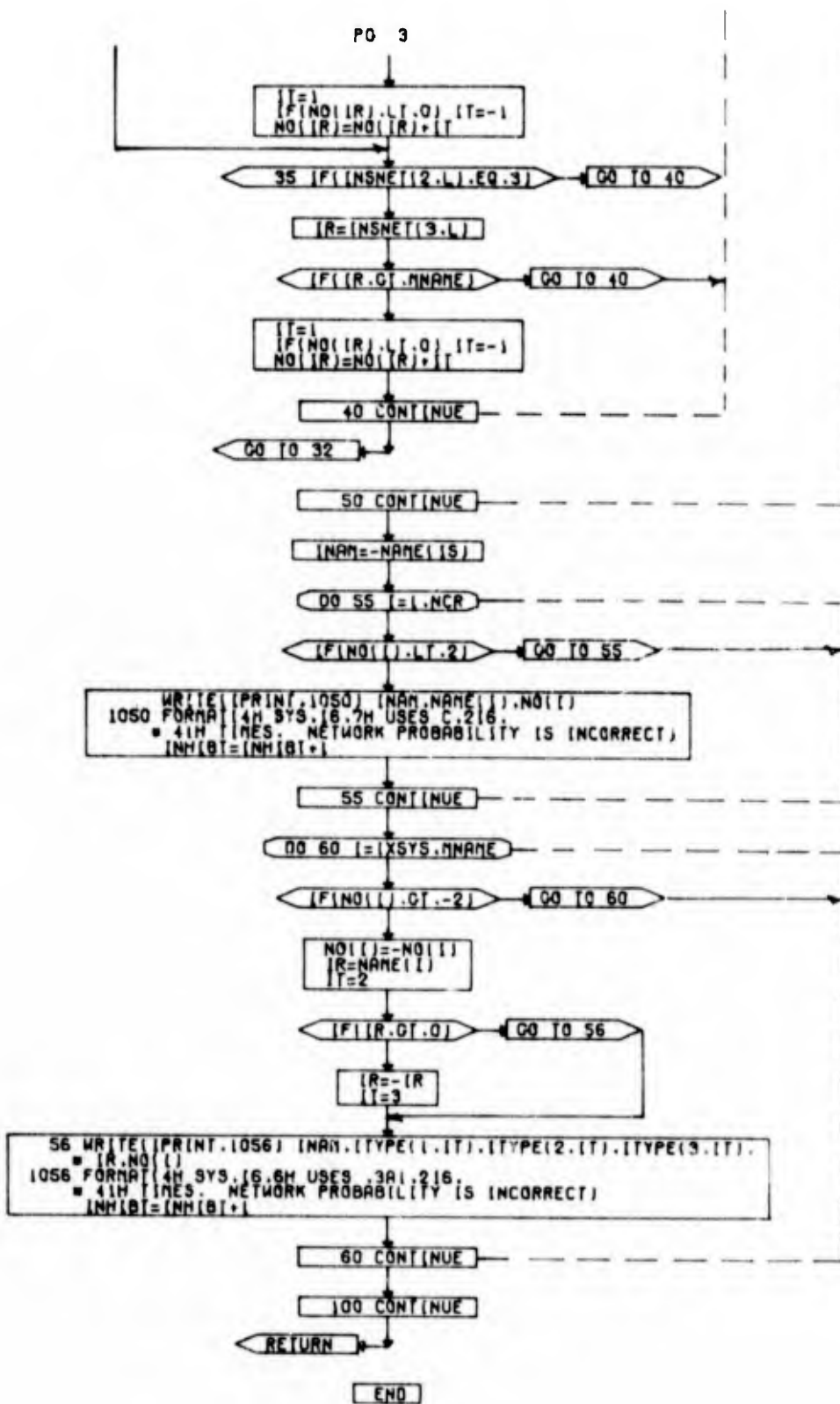
C .....
C SEARCH THROUGH NETWORK EQUATIONS TO DETERMINE
C 1) IF THERE ARE ANY COMPONENTS THAT ARE NOT USED IN THE NETWORK
C 2) IF ANY EQUATION REFERENCES A DELETED C, S6, OR SYS
C .....
INHIBL=0
DO 1 I=1,NCR
1 NOL1)=0
IXSYS=MNAME-NSYS+1
    
```





THIS SECTION OF CODE COMPLETELY TRACES THROUGH EACH SYSTEM NETWORK TO MAKE SURE THAT IN COMBINING COMPONENTS THROUGH USE OF SUBSYSTEMS THAT NO COMPONENT IS USED MORE THAN ONCE. IN OTHER WORDS, A SYSTEM IS NOT PERMITTED TO HAVE A REDUNDANCY.





SUBROUTINE PROO

```

COMMON/WORKKEY/ IKEY,LKEY,MKEY,IPAD1,PZ2(1)
COMMON/IDUNIT/ IREAD,IPRINT,IPUNCH,ITAPE
COMMON/CARROT/ NCRD,TECHO,INHBT,NCODE,
NULCNT,NEXT,IREP,INTPT,FRCPRT,ILOO,TEXPRT,
ICOL,LCOL,MCOL,ICARD(80)
COMMON/CL01 / NBYN,NBYM,MODE(1)
COMMON/CE02 / LLN (1)
COMMON/CE03 / FACK (1)
COMMON/CE05 / BETA (1)
COMMON/CE06A/ SIGSS (1)
COMMON/CE06C/ C66(1)
COMMON/CE06D/ IXS(1)
COMMON/CE07A/ STORU (1)
COMMON/CE07B/ RMCOL(1)
COMMON/CE07D/ CAU(1)
COMMON/CE07E/ IXAU(1)
COMMON/CE08 / ANVOX(1)
COMMON/CRO1 / IO66,NIMIN,NIMAX,NOMIN,NCONF1,NUMUR,NUMMR,
EP61,EP60,EP60C,EP6MAT,CONF(1)
COMMON/CRO2 / WD,W,MO,H,DP,RO,R(1)
COMMON/CRO3 / NR,NOP,OP(1)
COMMON/CNC1 / NCR,NSYS,MNAME,NAME(1)
COMMON/CNC2 / INGLC(2,1)
COMMON/CNC3 / NPS,IPAD2,P(1)
COMMON/CC01 / NCING,NCINS,INGC(4,1)
COMMON/CC02 / COE(3,1)
COMMON/CC03 / FCY(6,1)
COMMON/CC04 / FCX(6,1)
COMMON/CC05 / FC (6,1)
COMMON/CC06 / FU (8,1)
COMMON/CC07 / I8D(8,1)
COMMON/CN01 / NNINS,NNINS,INGNET(4,1)
COMMON/CT01 / X(1)
COMMON/CT02 / XBAR(1)
COMMON/CT03 / XOEL(1)
COMMON/CT04 / REG(1)
EQUIVALENCE (REG(1),Z1)
COMMON/CT05 / PZ(1)
COMMON/CT06 / PSAVE(22,1)
COMMON/CBETA / IPAD3,MAX80,BOMAX,OM(9),BO(21,9)
DIMENSION PERCNT(14),PCT(14)
COMMON/AZAUTH/ AZRN
    
```

```

C ..... INITIALIZE SUBROUTINE PROO
C ..... MODE(MBYN1) - ORIGINAL NAME AND POSITION OF ENVIRONMENTS
C ..... NBYNPI - USED IN INDEXING SYSTEMATIC COVARIANCE MATRIX
C ..... IXSYS THROUGH MNAME - INDEX TO SS AND SYS PROBABILITIES
C .....
C ..... MBYN1=3*MBYM
C ..... MBYNPI=MBYN*1
C ..... RANDOM VARIATION CONVERGENCE TEST NEEDS SUM AND SUM-SQ OF
C ..... PROBABILITIES FOR EACH SS AND SYS
C ..... IXSYS=MNAME-NSYS+1
    
```

IF (IXSYS.GT.NSYS) GO TO 2

WRITE (IPRINT,888)
888 FORMAT(44H PROGRAM ERROR - CONVERGENCE STORAGE PROBLEM )

RETURN

SET CONVERGENCE TEST VALUES
EP61Z=EP61\*EP61
IOEP60=2.\*EP60C
EP60G=EP60\*EP60

PD 2

DECOMPOSE SIOGG COVARIANCE MATRIX - RESULT IS C66  
NRSG=1  
CALL RNV01NRSG,NBYN,SIOGG,C66,RNV0X,1X66

RETURN IF DECOMPOSITION FAILED. RETURN  
IF(NRSG.EQ.0)

ASSUME NO L0010 ENVIRONMENT - GET TO DECOMPOSE SYSTEMATIC  
COVARIANCE MATRIX ONLY ONCE FOR ALL R/OP  
MODEGO=1

BEGIN RANGE/PRESSURE LOOP - THIS LOOP INCLUDE ALL OF THE CODE  
DOWN TO THE END CARD

IR=0

10

SELECT RANGE/PRESSURE  
IR=IR+1

RETURN IF (IR GT. NR)

PRINT ULN(10)=PRESSURE  
IO=0  
WRITE(IPRINT,1000) IO,OP1(IR)  
1000 FORMAT(5H ULN(12.2H)=.E15.5)

SCALE AND PRINT ENVIRONMENTS

IO=RDYNT3

00 26 I=1,NBYN

GET THE ENVIRONMENT NAME FROM MODE(10) FOR PRINTING  
IO=IO+1  
SCALE EACH ENVIRONMENT USING RANGE OR PRESSURE  
TOP=MODE(1)

IF (TOP.LT.4) SCALE=(R(IR)/RO)=BETA(1)

IF (TOP.GT.5) SCALE=(OP(IR)/OPO)=BETA(1)

GO TO (11,12,13,11,12,13),TOP

MODE 16 LN NORMAL  
11 XBAR(1)=ULN(1)+ALOG(SCALE)  
XDEL(1)=EXP(XBAR(1))  
WRITE(IPRINT,1000) MODE(10),XDEL(1)

GO TO 25

MODE 16 NORMAL  
12 XBAR(1)=ULN(1)+SCALE  
WRITE(IPRINT,1000) MODE(10),XBAR(1)

GO TO 25

MODE 16 NORMAL L0010 - GET TO CONSTRUCT AND DECOMPOSE  
SYSTEMATIC COVARIANCE MATRIX FOR EACH R/OP  
13 XBAR(1)=ULN(1)+ALOG(SCALE)  
MODEGO=3  
WRITE(IPRINT,1000) MODE(10),XBAR(1)

25 CONTINUE

PD 3

PD 3

```
C***** PRINT CURRENT THREAT CONDITIONS  
WRITE(IPRINT,3000) W.H.R(I,R).OP(I,R)  
3000 FORMAT(3H0W=E11.4,3H M=E11.4,3H R=E11.4,3H P=E11.4)
```

```
C*****  
CONSTRUCT SYSTEMATIC COVARIANCE MATRIX  
CONSTRUCT ONLY ONCE  
SKIP IF MATRIX IS CONSTRUCTED  
LOGIO REQUIRES CONSTRUCTION EACH TIME  
C*****
```

GO TO (30,40,51),NODEGO

30 NODEGO=2

```
C***** K-FACTOR DETERMINES STANDARD DEVIATION  
31 II=1  
DO 32 I=1,NBYN  
O1=FACTK(I)  
SIOAU(II)=0.0  
IF(ABS(O1)-1.0).LT.0.0001; GO TO 32  
IF(NODE(II).EQ.3) O1=SQRT((XBAR(II)-O1)/(XBAR(II)-O1))  
IF(NODE(II).EQ.6) O1=SQRT((XBAR(II)-O1)/(XBAR(II)-O1))  
SIOAU(II)=ALOG(O1)/1.96  
32 II=II+NBYN-1
```

```
C***** RHOCOL CONTAINS CORRELATION COEFFICIENT  
L=0  
II=1  
K1=NBYN-2  
DO 34 I=2,NBYN  
L=L+1  
KK=II  
DO 33 K=1,NBYN  
L=L+1  
KK=KK+K1-K  
SIOAU(L)=RHOCOL(L)=SIOAU(II)*SIOAU(KK)  
33 CONTINUE  
SIOAU(II)=SIOAU(II)**2  
34 II=II+K1-1  
SIOAU(II)=SIOAU(II)**2
```

```
C***** DECOMPOSE SIOAU COVARIANCE MATRIX - RESULT IS CAU  
NRAU=2  
CALL ANVO(NRAU,NBYN,SIOAU,CAU,ANVOX,IXAU)
```

RETURN

C\*\*\*\*\* IF DECOMPOSITION FAILED, RETURN  
(F(NRAU.EQ.0))

40

```
C***** SET SUMMATION CELLS AND COUNTERS  
DO 50 I=1,NNAME  
PZ(I)=0.0  
PZ2(I)=0.  
DO 50 II=1,22  
PBAVE(II,1)=0.0  
50 CONTINUE  
NO=0  
NI=0  
NPZ2=0  
ILTEST=NIMAX  
NISTOP=NIMIN  
NDSTOP=NOMIN  
NOOL=0
```

PD 4

PO 4

100

```

C .....
C BEGIN SYSTEMATIC VARIATION LOOP (OUTER LOOP)
C SELECT ENVIRONMENT (ENV) VARIATION
C .....
  GO 101 I=1,NBYN
101 ANVOX(I)=0.
  L=0
  GO 102 I=1,NRAU
C----- NORMAL RANDOM NUMBER GENERATOR
  CALL FNRN(NUMRA,AN)
  GO 102 J=1,NRAU
  L=L+1
102 ANVOX(J)=ANVOX(J)+AN*CRU(L)
  GO 103 I=1,NBYN
  L=IXAU(I)
103 X(I)=XBAR(I)+ANVOX(L)

```

```

C .....
C SELECT TRANSFER VARIATION
C .....
  GO 110 I=1,NCINS
  ICP=INSC(I,I)
  IF(ICP.LT.7) GO TO 110
  IF(ICP.GT.12) GO TO 110
C----- UNIFORM RANDOM NUMBER GENERATOR
  CALL UPR1(NUMUR,AN)
  LL=INSC(I,I)
  FL1=AN*BDMAX+1.0
  L=FL1
  LP1=L+1
  FL2=L
  FL1=FL1-FL2
  FL2=1.0-FL1
  BDV=FL1*BD(LP1,LL)+FL2*BD(L,LL)
  COEF(3,I)=COEF(1,I)+BDV*COEF(2,I)
110 CONTINUE

```

```

C .....
C SELECT FRAILITY VARIATION
C .....

```

GO 130 I=1,NCR

\*\*\*\*\* SKIP IF NO POINTS VARY IF(100(8,I).EQ.0) GO TO 130

```

C----- UNIFORM RANDOM NUMBER GENERATOR
  CALL UPR1(NUMUR,AN)
  FL1=AN*BDMAX+1.0
  L=FL1
  LP1=L+1
  FL2=L
  FL1=FL1-FL2
  FL2=1.0-FL1
  KK=100(7,I)

```

GO 129 K=1, KK LL=100(K,I)

GO TO 129 IF(LL.EQ.0)

```

  BDV=FL1*BD(LP1,LL)+FL2*BD(L,LL)-DH(LL)
  FCX(K,I)=FCX(K,I)+FU(K,I)*BDV

```

129 CONTINUE

130 CONTINUE

PO 5

200

```

C .....
C C BEGIN RANDOM VARIATION LOOP (INNER LOOP)
C C SELECT ENVIRONMENT VARIATION
C .....
C C DO 201 I=1,NBYN
201 RNVGX(I)=0.
L=0
DO 202 I=1,NRGG
C----- NORMAL RANDOM NUMBER GENERATOR
CALL FNRN(NUNNR,RN)
DO 202 J=1,NRGG
L=L+1
202 RNVGX(J)=RNVGX(J)+RN*CGG(L)
DO 205 I=1,NBYN
L=LXGG(I)
FLI=X(I)+RNVGX(L)
C----- TAKE ANTILOG OF LN NORMAL ENV.
IF(NODE(I).EQ.1) FLI=EXP(FLI)
IF(NODE(I).EQ.4) FLI=EXP(FLI)
C----- USE INPUT SUBSCRIPT TO STORE ENV.
L=NBYN(I)+1
LL=NODE(L)
XDEL(LL)=FLI
205 CONTINUE
    
```

```

C .....
C C SELECT AZIMUTH VARIATION
C .....
C C----- UNIFORM RANDOM NUMBER GENERATOR
CALL UPRN(NUNUR,AZRN)
    
```

```

C .....
C C COMPUTE PROBABILITY OF FAILURE (PF) FOR EACH COMPONENT
C .....
    
```

00 219 N=1,NCR

```

C .....
C C TRANSFER ENVIRONMENT(S) TO FIND COMPONENT RESPONSE
C .....
L=INSLC(1,N)
LL=INSLC(2,N)
DO 230 I=1,LL
OP=INSC(1,I)
IR1=INSC(2,I)
IR2=INSC(3,I)
IF(OP.GT.6) OP=10P-6
DO TO (211,212,213,214,215,216,217,218),OP
211 REG(IR1)=REG(IR2)*COEF(3,I)
DO TO 230
212 REG(IR1)=OP(IR1)*COEF(3,I)
DO TO 230
213 REG(IR1)=XDEL(IR2)*COEF(3,I)
DO TO 230
214 REG(IR1)=REG(IR2)*COEF(3,I)
DO TO 230
215 REG(IR1)=OP(IR1)*COEF(3,I)
DO TO 230
216 REG(IR1)=XDEL(IR2)*COEF(3,I)
DO TO 230
217 CALL CAN(IR1,IR2)
DO TO 230
218 CALL USER(IR1)
230 CONTINUE
    
```

```

C .....
C C FIND PF FROM COMPONENT RESPONSE AND FRAGILITY CURVE
C .....
    
```

IF(ZI.LE.FCX(1,N)) GO TO 228

LL=100(7,N)

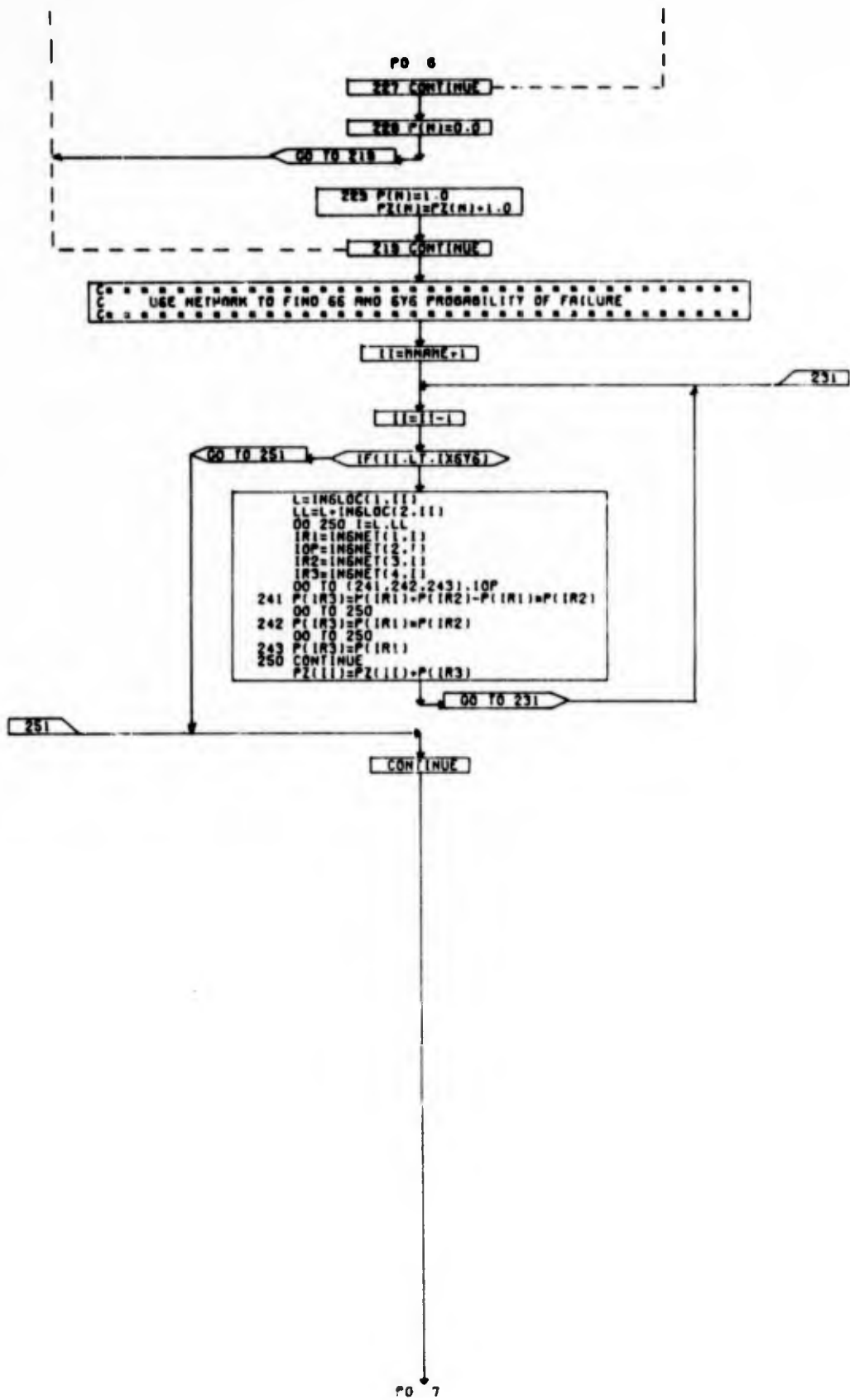
IF(ZI.GT.FCX(LL,N)) GO TO 225

00 227 L=2,LL

IF(ZI.GT.FCX(L,N)) GO TO 227

P(N)=(FCY(L,N)-FCY(L-1,N))/(ZI-FCX(L-1,N))  
 PZ(N)=PZ(N)+P(N)

GO TO 219



PO 7

C . . . . . END OF INNER LOOP - INCREMENT NUMBER OF ITERATIONS  
C . . . . .

N1=N1+1

GO TO 263 C . . . . . SEE IF INNER LOOP TESTING HAS BEEN COMPLETED  
IF(N1.GT.1LTEST)

C . . . . . TEST CONVERGENCE OF INNER LOOP  
C . . . . . ACCUMULATE INFORMATION FOR TEST  
C . . . . .  
NPZ2=NPZ2+1  
DO 252 I=1XSY6.MNAME  
252 PZ2(I)=PZ2(I)+P(I)\*P(I)

GO TO 200 IF(N1.LT.N1STOP)

C . . . . . TEST CONVERGENCE  
FNI=N1  
ENPZ2=NPZ2-(N1+1)  
ENPZ2=ENPZ2+EPSI2  
NSTOP=5  
I1=0  
DO 254 I=1XSY6.MNAME  
I1=I1+1  
IF(MNAME(I).LT.0) GO TO 253  
IF(10066.EQ.0) GO TO 254  
253 VTEMP=PZ2(I1)+PZ2(I1)-PZ(I1)\*P(I1)/FNI  
NTEMP=VTEMP/ENPZ2+.5  
IF(NTEMP.GT.NSTOP) NSTOP=NTEMP  
254 CONTINUE

IF(NSTOP.LT.N1STOP) GO TO 255

C . . . . . NO CONVERGENCE - ESTIMATE ITERATIONS REQUIRED  
IF(NSTOP.GT.2\*N1STOP) NSTOP=2\*N1STOP  
N1STOP=NSTOP  
WRITE(IPRINT,1002) N1STOP,NO.EPSI  
1002 FORMAT(21H NO CONVERGENCE - N1=,16.4H N0=,16.5H EPS=,E8.1)

GO TO 200

255

C . . . . . INNER LOOP CONVERGED  
N1STOP=NSTOP  
WRITE(IPRINT,1002) N1STOP,NO.EPSI

C . . . . . ACCUMULATE I.L. INFORMATION OVER SEVERAL O.L.  
I1=0  
DO 256 I=1XSY6.MNAME  
I1=I1+1  
PZ2(I1)=PZ2(I1)+PZ2(I1)-PZ(I1)\*P(I1)/FNI  
256 PZ2(I1)=0.0

GO TO 300

263

GO TO 200 C . . . . . EACH O.L. NOW REQUIRES N1STOP I.L. ITERATIONS  
IF(N1.LT.N1STOP)

300

C . . . . . PLACE A TIC MARK FOR EACH C, S5 AND S16 IN HISTOGRAM.  
C . . . . . THE POSITION OF THE MARK DEPENDS ON AVERAGE % FROM INNER LOOP.  
C . . . . .  
FNI=N1  
DO 301 I=1,NCR  
L=((FNI-PZ(I1))/FNI)\*20.  
L=L+2  
IF(PZ(I1).EQ.FNI) L=L-1  
PZ(I1)=0.0  
PSAVE(L,1)=PSAVE(L,1)+1.0  
301 CONTINUE  
DO 302 I=1XSY6.MNAME  
L=((FNI-PZ(I1))/FNI)\*20.  
L=L+2  
IF(PZ(I1).EQ.FNI) L=L-1  
PZ(I1)=0.0  
PSAVE(L,1)=PSAVE(L,1)+1.0  
302 CONTINUE

PO 8

PD 8

TEST CONVERGENCE OF OUTER LOOP. DO REQUIRED OUTER LOOP ITERATIONS BEFORE TESTING.  
NI=0  
NO=NO-1

GO TO 100 IF (NO.LT.NOSTOP)

FNO=NO

IF (NO0.EQ.1) GO TO 306

TEST FOR CONVERGENCE ON THE MEAN  
DO 305 I=1XSY6.MNAME

IF (NAME(I).LT.0) GO TO 303

GO TO 305 IF (10065.EQ.0)

305 F1=-.025  
S=PSAVE(22.1)  
S2=S  
DO 304 L=2.21  
F1=F1+.05  
M1=PSAVE(L.1)=F1  
S=S+M1  
304 S2=S2+M1\*F1  
NTEMP=(S2-S/FNO)/(EP6060\*(FNO-1.0))+.5

GO TO 305 IF (NTEMP.LE.NO)

NO CONVERGENCE - ESTIMATE ITERATIONS REQUIRED  
IF (NTEMP.GT.(NO-NO)) NTEMP=NO+NO  
WRITE (PRINT.1002) N1STOP.NO.EPS0SQ  
NOSTOP=NTEMP

GO TO 100

305 CONTINUE

MEAN CONVERGED  
NO0=

GO TO 406 IF (CONF(1).EQ.0.0)

306

TEST FOR CONVERGENCE ON THE MEDIAN  
L=0  
DO 401 I=1.NCONF  
CN=CONF(I)\*FNO  
FNI= SORT(CN\*(1.0-CONF(I)))  
L=L+1  
PERCNT(L)=CN-FNI  
L=L+1  
401 PERCNT(L)=CN\*FNI

DO 405 I=1XSY6.MNAME  
IF (NAME(I).LT.0) GO TO 402

GO TO 405 IF (10065.EQ.0)

402 CALL PROBL1.PSAVE(1.1).PERCNT.PCT)  
II=0

403 II=II+2

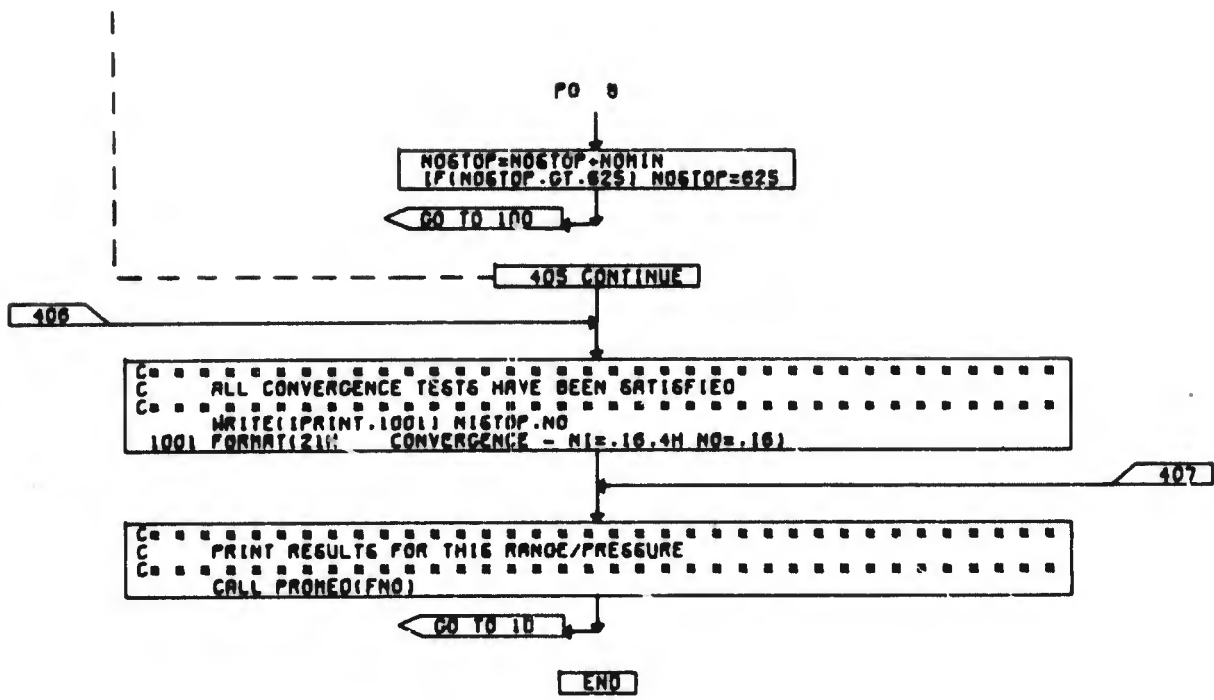
GO TO 405 IF (II.GY.L)

IF ((PCT(II)-PCT(II-1)).LE.TOEP60) GO TO 403

NO CONVERGENCE - GET TO DO MORE ITERATIONS (UP TO 625)  
WRITE (PRINT.1002) N1STOP.NO.TOEP60

IF (NO.GE.625) GO TO 407

PD 9



SUBROUTINE RNVD(MAT,N,S,C6,C,IV)

```

COMMON/WRKEY/ IKEY,LKEY,AREY,(PAOI,KEY(1))
COMMON/IOUNIT/ IREAD,IWRITE,IPUNCH,ITAPE
COMMON/CRO1 / IOO6,NIMIN,NIMAX,NOMIN,NONFI,NUMR,NUMNR,
      EPS1,EP6C,EP6OC,EP6MAT,CONF(7)
DIMENSION S(1),C6(1),C(N,1),IV(1),JV(1)
EQUIVALENCE (JV(1),KEY(1))
    
```

```

C ..... IV=0 INDICATES ROW AND COLUMN NOT YET PROCESSED
C ..... EXPAND THE TRIANGULAR MATRIX TO A FULL MATRIX
C .....
      L=0
      DO 5 J=1,N
        IV(J)=0
      DO 5 I=J,N
        L=L+1
        C(I,J)=S(L)
      S C(I,J)=S(L)
    
```

```

C ..... BEGIN DECOMPOSITION OF COVARIANCE MATRIX
C .....
    
```

DO 40 K=1,N

```

C ..... LOOK FOR LARGEST DIAGONAL ELEMENT WHICH IS PIVOTAL ELEMENT OF
C ..... ROW AND COLUMN NOT YET PROCESSED
C .....
      II=0
      OMAX=-1.E10
      DO 10 I=1,N
        IF (IV(I).NE.0) GO TO 10
        IF (OMAX.GE. C(I,I)) GO TO 10
        OMAX=C(I,I)
      II=I
    10 CONTINUE
    
```

..... SEE IF ALL ROWS AND COLUMNS HAVE BEEN PROCESSED IF(II.EQ.0) GO TO 80

```

C ..... IV LISTS THE ORDER IN WHICH ROWS AND COLUMNS ARE PROCESSED
C ..... JV LISTS THE ORIGINAL ORDER OF ROWS AND COLUMNS
      IV(II)=K
      JV(K)=II
      C(II)=C(I,II)
    
```

```

C ..... MATRIX NOT SEMIDEFINITE IF DIAGONAL TERM IS LESS THAN ZERO
C ..... OTHERWISE, 100=1 IF TERM EQUALS ZERO, AND 100=2 IF TERM IS
C ..... GREATER THAN ZERO
    
```

IF(C(II).LT.-EPSMAT) GO TO 100

100=1 IF(C(II).GT.EPSMAT) 100=2

..... SEE IF MATRIX RANK IS LESS THAN DIMENSIONED IF(100.EQ.1) JV(K)=0

```

C ..... PROCESS ELEMENTS IN THE ROW AND COLUMN OF THE PIVOTAL ELEMENT
C .....
    
```

DO 20 I=1,N

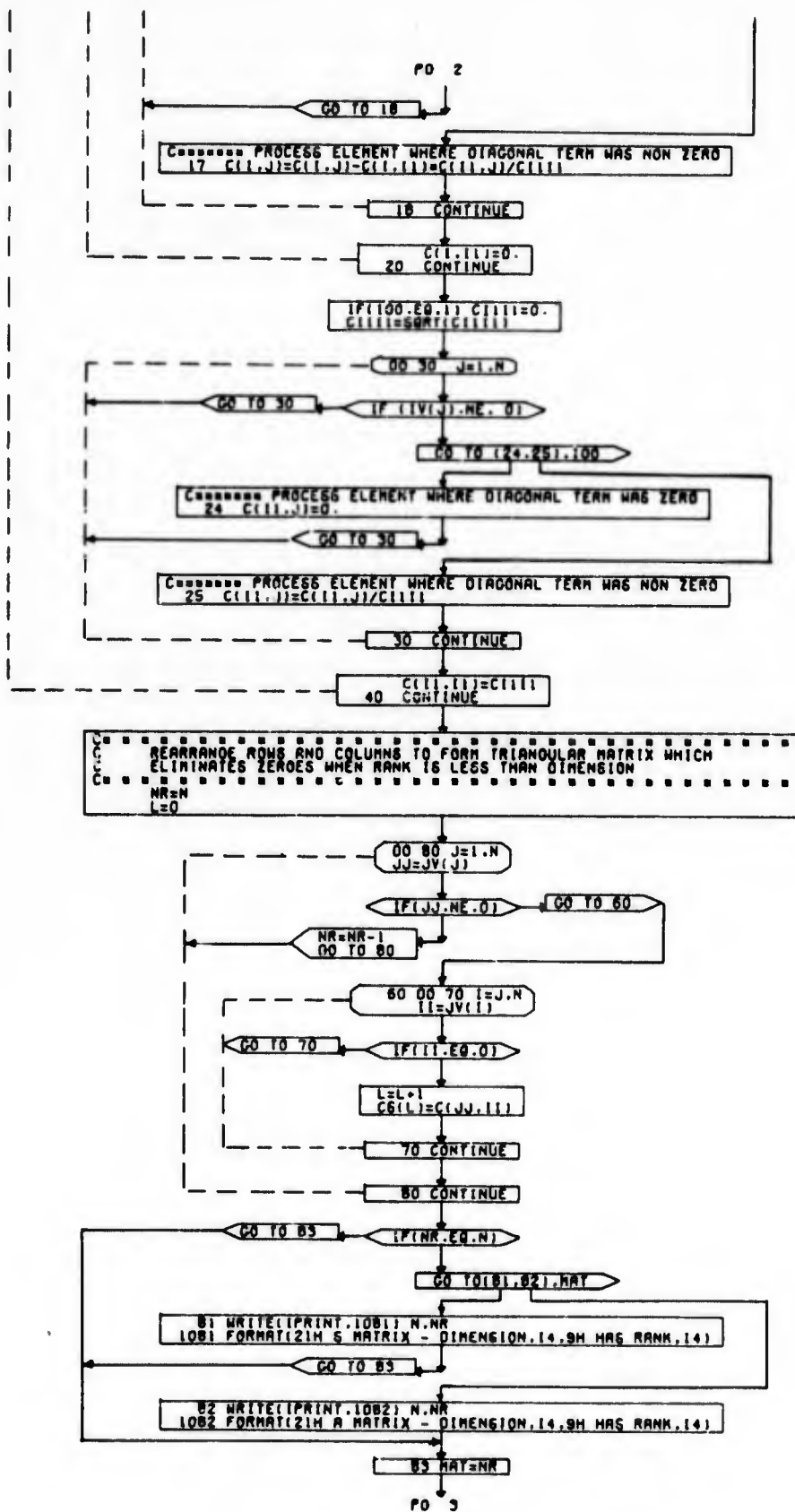
GO TO 20 IF (IV(I).NE.0)

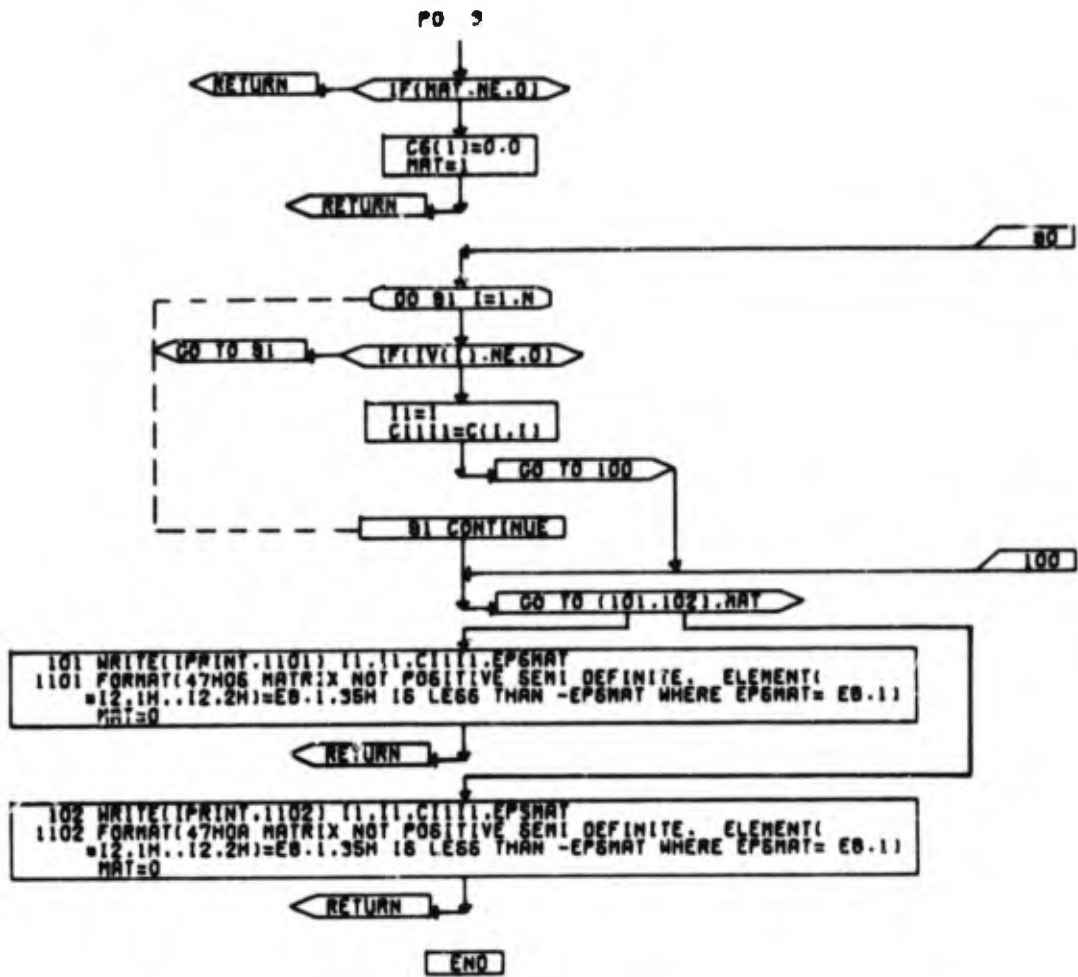
DO 18 J=1,N

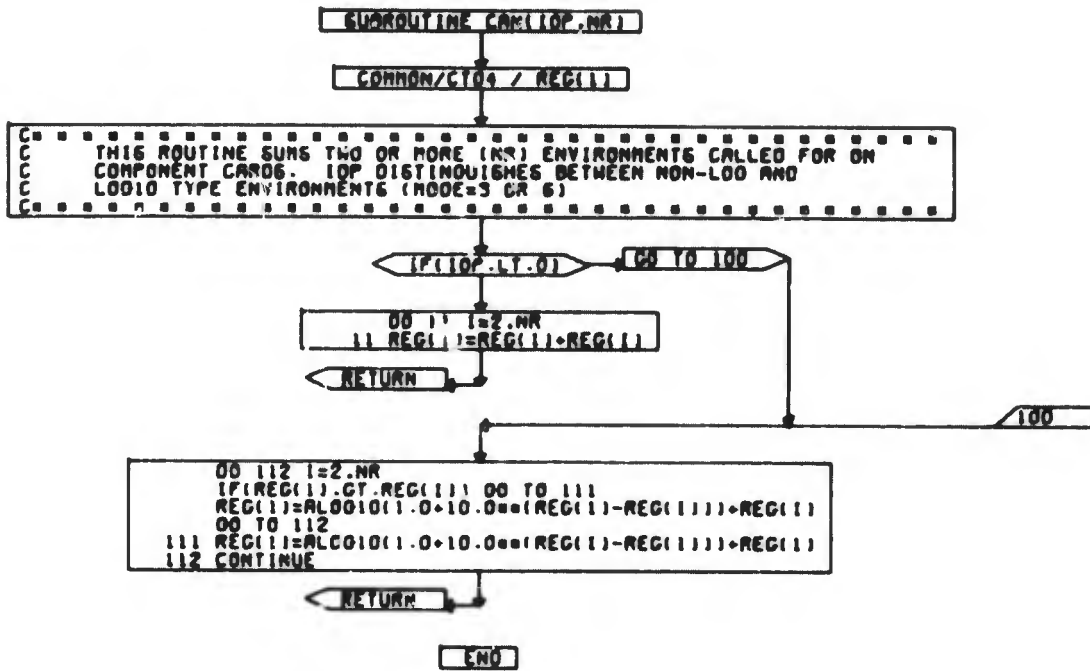
GO TO 18 IF (IV(J).NE.0)

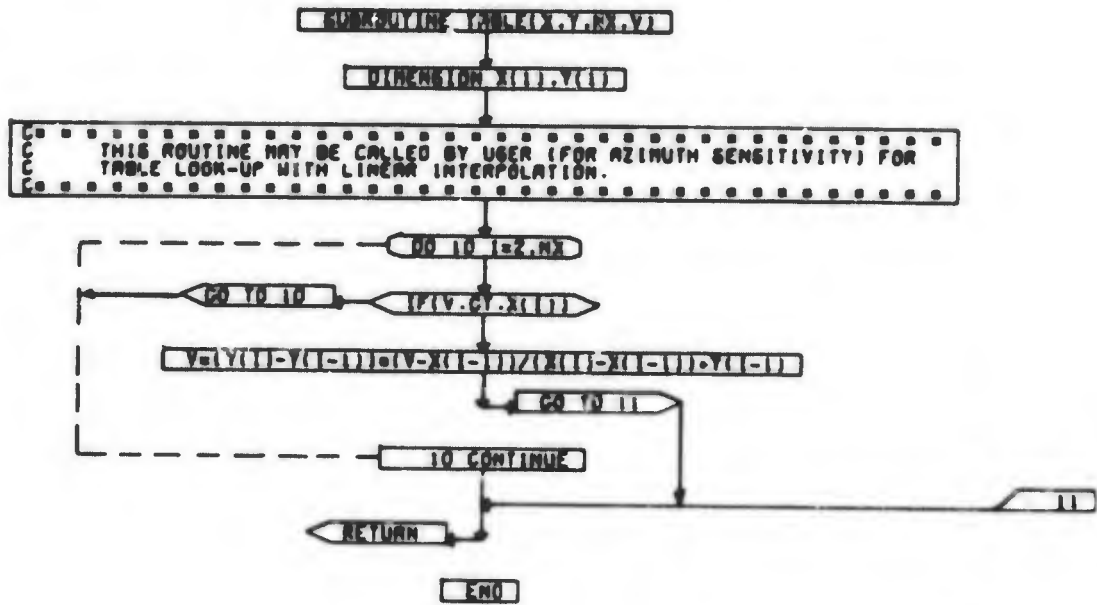
GO TO (18,17),100

..... PROCESS ELEMENT WHERE DIAGONAL TERM WAS ZERO C(I,J)=0









PO 1

SUBROUTINE PROB0(FNO)

```

COMMON/OUNTY/ IREAD,IPRINT,IPUNCH,ITAPE
COMMON/MDRNET/ IKEY,LKEY,MKEY,IPAD1,KEY(11)
COMMON/CNC1 / NCR,N5YS,MNAME,MARE(11)
COMMON/CNC3 / NPS,IPAD2,P(11)
COMMON/CT08 / PSAVE(22,1)
COMMON/CT07 / OPINCH,NPROB,NPLOT,PROB(11)
DIMENSION PERCNT(7),PCT(7)

```

```

C * * * * * PRINT THE HISTOGRAM WHICH REPRESENTS THE DENSITY DISTRIBUTION * * * * *
C * * * * *
IXYS=MNAME-N5YS-1
WRITE(IPRINT,1005)
1005 FORMAT(90HP#0 (----- HISTOGRAM (IN STEPS OF .05) FOR P6 GREATER
      *THAN 0 AND LESS THAN -----) P6#1 1)
DO 365 I=1,NCR
WRITE(IPRINT,1006) (PSAVE(11,1),I=1,22),MARE(11)
1006 FORMAT(1M .22F4.0.19)
365 CONTINUE
DO 366 I=1,IXYS,MNAME
WRITE(IPRINT,1008) (PSAVE(11,1),I=1,22),MARE(11)
366 CONTINUE

```

```

C * * * * * PREPARE PERCENTILE LEVELS * * * * *
PERCNT(1)=.01*FNO
PERCNT(2)=.1*FNO
PERCNT(3)=.5*FNO
PERCNT(4)=.5*FNO
PERCNT(5)=.7*FNO
PERCNT(6)=.9*FNO
PERCNT(7)=.99*FNO
NPCR=7
NIOPER=4
I=1
IE=NCR

```

```

C * * * * * FOR EACH C, SS AND SYS, CALL FOR THE CUMULATIVE PROBABILITY * * * * *
C * * * * * AT THE DIFFERENT PERCENTILES. * * * * *
DO 50 I=1,IE
CALL PROB1(NPCR,PSAVE(1,1),PERCNT,PCT)
40 DO 41 IP=1,NPCR
41 PSAVE(IP,1)=PCT(IP)
P(1)=PCT(NIOPER)
KEY(1)=I
50 CONTINUE

```

(IF I0.GT.1) GO TO 60

I0=MNAME-N5YS-1  
IE=MNAME

GO TO 1

```

80 CONTINUE
C * * * * * RANK COMPONENTS BY HARDNESS AT THE 50 PERCENT LEVEL * * * * *
CALL RORD(IP,KEY,NCR)

```

(IF (OPINCH.EQ.0.01) GO TO 62

```

C * * * * * IF PLOTS REQUESTED GAVE 50 PERCENT VALUES FOR S6 AND S7S * * * * *
C * * * * *
NPROB=NPROB-1
IP=NPROB
DO 61 I=1,IE
PROB(IP)=P(1)
IP=IP+NPLOT
61 CONTINUE

```

62

```

C * * * * * RANK S6 AND S7S HARDNESS AT THE 50 PERCENT LEVEL * * * * *
C * * * * *
CALL RORD(IP(1),KEY(1),N5YS)

```

PO 2

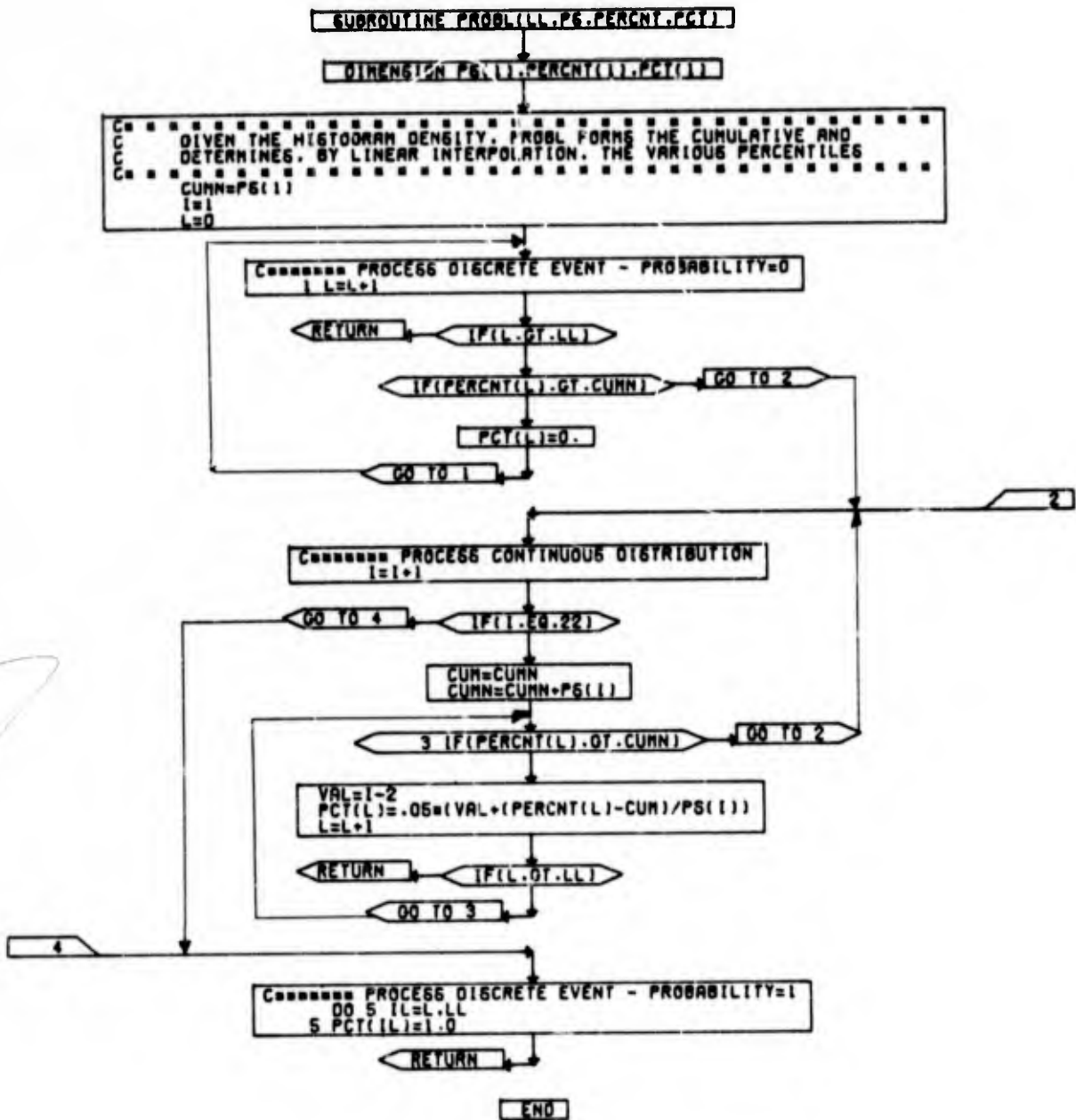
```

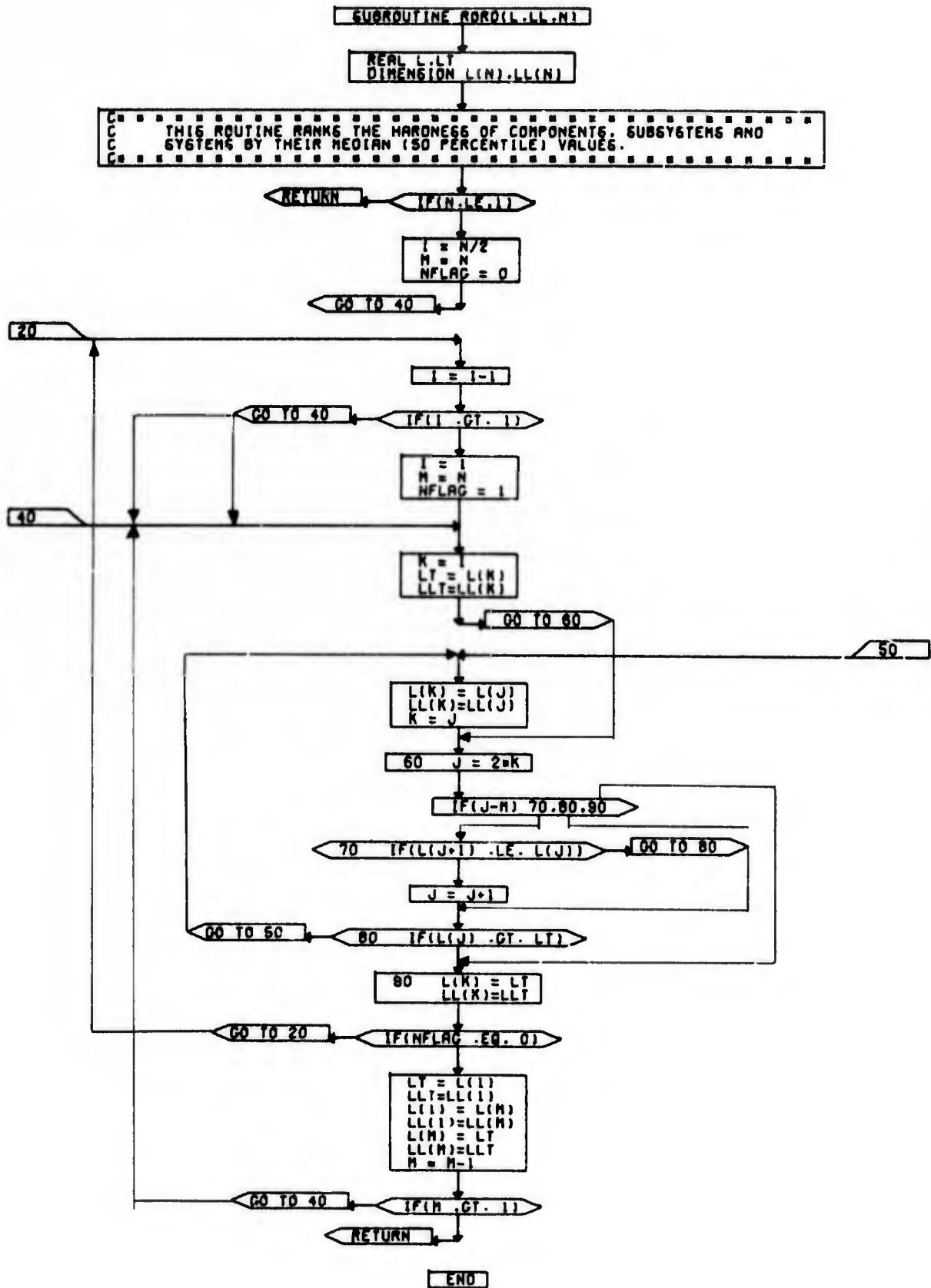
C * * * * *
C * PRINT CUMULATIVE PROBABILITY RESULTS FOR C. SS AND SYS
C * * * * *
WRITE( IPRINT, 1000)
1000 FORMAT( 22H0INDIVIDUAL COMPONENTS )
WRITE( IPRINT, 1001)
1001 FORMAT( 1H .4X, 4HNAME, 7X, 1H1, 6X, 2H10, 6X, 2H30, 6X, 2H50, 6X, 2H70,
* 6X, 2H80, 6X, 2H88)
DO 70 I=1, NCR
N=KEY( I )
WRITE( IPRINT, 1002) NAME( N ), ( PSAVE( I, N ), I, P=1, NPER)
1002 FORMAT( 1H .18, 7F8.3)
70 CONTINUE
WRITE( IPRINT, 1003)
1003 FORMAT( 22H0INDIVIDUAL SUBSYSTEMS )
WRITE( IPRINT, 1001)
DO 80 I=18, IE
N=KEY( I )
IF( NAME( N ), LT, 0) GO TO 80
WRITE( IPRINT, 1002) NAME( N ), ( PSAVE( I, N ), I, P=1, NPER)
80 CONTINUE
WRITE( IPRINT, 1004)
1004 FORMAT( 18H0INDIVIDUAL SYSTEMS )
WRITE( IPRINT, 1001)
DO 90 I=18, IE
N=KEY( I )
IF( NAME( N ), GT, 0) GO TO 90
WRITE( IPRINT, 1002) NAME( N ), ( PSAVE( I, N ), I, P=1, NPER)
90 CONTINUE

```

← RETURN →

END





SUBROUTINE OOPLOT

```
COMMON/IOUNIT/ IREAD,IPRINT,IPUNCH,ITAPE
COMMON/CRO3 / NR,NOP,OP(1)
COMMON/CNC1 / NCR,N6Y6,MNAME,NAME(1)
COMMON/CTO7 / OPINCH,NPROB,MPLT,PROB(1)
COMMON/CTO7B/ IB(63),IB(63),IPLT(31,4)
COMMON /CHRL6T/ KA .KB .KC .KD .KE .KF .KG .KH .KI .KJ .
. KK .KL .KM .KN .KO .KP .KQ .KR .KS .KT .
. KU .KV .KW .KX .KY .KZ .KO .KI .K2 .K3 .
. K4 .K5 .K6 .K7 .K8 .K9 .KCM .KPL .KMI .KAS .
. K6L .KPR .KLP .KRP .KEG .KOL .KBL
COMMON/CNC3 / MP6,IPAG2,P(1)
```

```
C . . . . . SCALES PRESSURES BY THE VALUE ENTERED ON PLOTON CARD AND
C . . . . . INITIALIZES PRINT LINES.
C . . . . .
DO 11 I=1,NR
11 P(I)=OP(I)/OPINCH
23 1 I=1,62
IB(I)=KPR
16(I)=KBL
1 CONTINUE
IB(1)=KBL
16(2)=KPR
IB(63)=KBL
16(63)=KBL
```

```
C . . . . . LOCATE PLOT DATA FOR TWO PLOTS AND PRINT PLOT IDENTIFICATIONS
C . . . . .
IP=1-MPLT
NB=MNAME-N6Y6-1
```

NB=NB-2

RETURN IF(NB.GT.MNAME)

```
NE=NB-1
IF(NE.GT.MNAME) NE=NB
WRITE(IPRINT,1000) (NAME(I),I=NB,NE)
1000 FORMAT(1H1,16.56X,16)
```

```
C . . . . . INITIALIZE VECTORS FOR POSITIONING PLOT POINTS BY SPLOT AND
C . . . . . CALL SPLOT ONCE FOR EACH PLOT
C . . . . .
DO 3 I=1,31
IPLT(I,1)=2
IPLT(I,3)=2
IPLT(I,2)=KBL
IPLT(I,4)=KBL
3 CONTINUE
IPLT(31,2)=KPR
IPLT(31,4)=KPR
IP=IP+MPLT
CALL SPLOT(P,PROB(IP),NR,IPLT(I,1))
```

IF(NB.EQ.NE) GO TO 4

```
IP=IP+MPLT
CALL SPLOT(P,PROB(IP),NR,IPLT(I,3))
```

4 CALL PRPLOT

GO TO 2

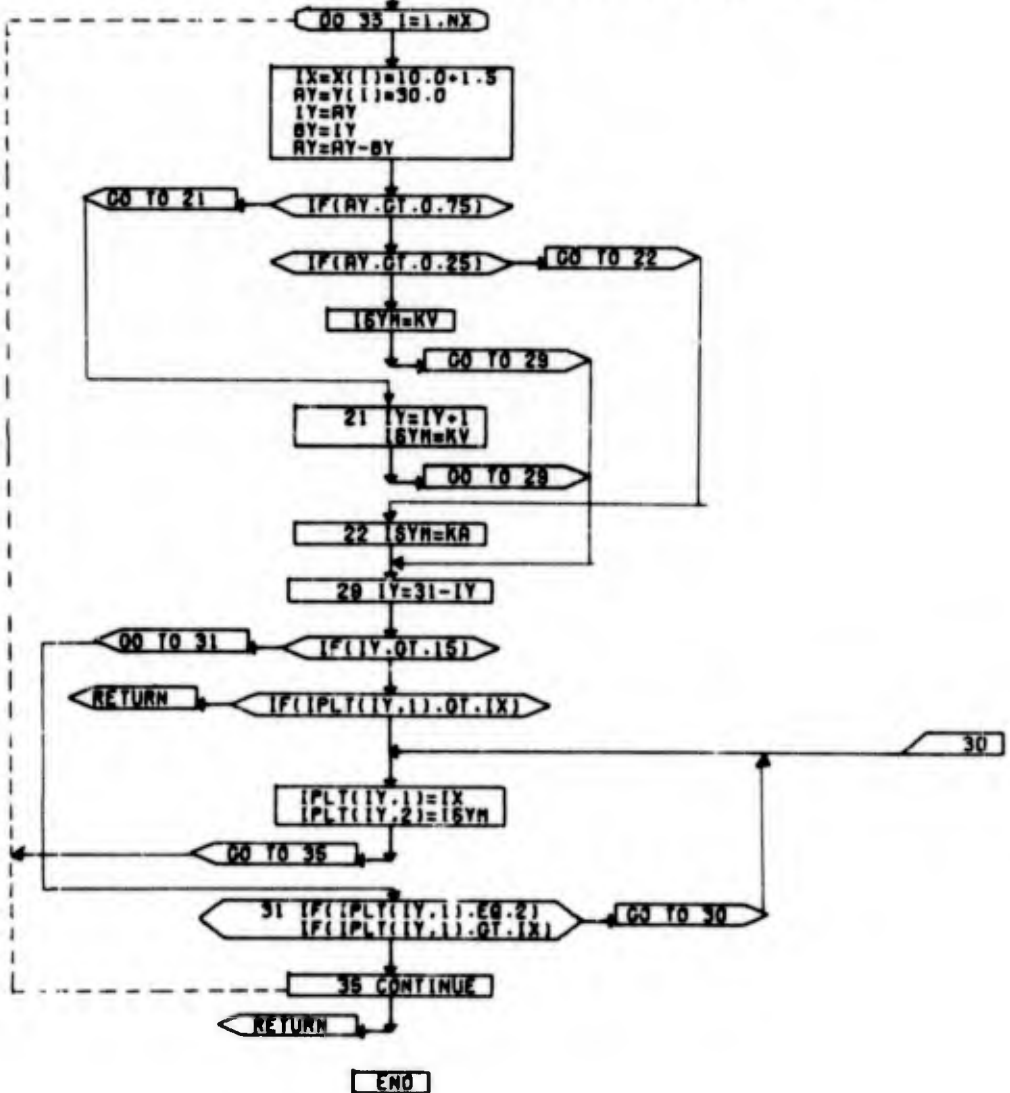
END

SUBROUTINE SPLAY(X,Y,NX,IPLY)

```

DIMENSION X(1),Y(1),IPLY(31,2)
COMMON /CHRG1/ KA ,KB ,KC ,KD ,KE ,KF ,KG ,KH ,KI ,KJ ,
               KK ,KL ,KM ,KN ,KO ,KP ,KQ ,KR ,KS ,KT ,
               KU ,KV ,KW ,KX ,KY ,KZ ,KO ,KI ,K2 ,K3 ,
               K4 ,K5 ,K6 ,K7 ,K8 ,K9 ,KCH,KPL,KMI,KAS ,
               KGL,KPR,KLP,KRP,KEG,KOL,KBL
    
```

THIS ROUTINE INTEGRIZES PROBABILITIES AND PRESSURES TO PRINT LINE AND PRINT WHEEL. ALSO DETERMINES WHETHER TO USE A OR V



SUBROUTINE PFL01

COMMON/IOUNIT/ IREAD, IPRINT, IPUNCH, ITYPE  
COMMON/CT078/ I0(53), I6(63), IPLT(31,4)

C \* \* \* \* \*  
C THIS ROUTINE PRINTS TWO PLOTS PER PAGE WITH AXES AND A OR V  
C SYMBOLS TO REPRESENT POINTS  
C \* \* \* \* \*  
WRITE(IPRINT,1000)  
1000 FORMAT(1H0)

C \* \* \* \* \* PRINT LINE OF \* \* \* \* \* FOR PROBABILITY EQUAL TO 1  
ISKP=IPLT(1,1)  
ISKP2=ISKP-2  
ISKP3=IPLT(1,3)  
WRITE(IPRINT,1001) ((I0(I),I=1,ISKP),IPLT(1,2))  
\* ((I6(I),I=ISKP2,63),(I0(I),I=1,ISKP3),IPLT(1,4))

C \* \* \* \* \* PRINT LINE OF BLANKS FOR OTHER PROBABILITIES  
DO 1 L=2,30  
ISKP=IPLT(L,1)  
ISKP2=ISKP-2  
ISKP3=IPLT(L,3)  
WRITE(IPRINT,1001) ((I0(I),I=1,ISKP),IPLT(L,2))  
\* ((I6(I),I=ISKP2,63),(I0(I),I=1,ISKP3),IPLT(L,4))  
1 CONTINUE

C \* \* \* \* \* PRINT LINE OF \* \* \* \* \* FOR PROBABILITY EQUAL TO 0  
ISKP=IPLT(31,1)  
ISKP2=ISKP-2  
ISKP3=IPLT(31,3)  
ISKP4=ISKP3-2  
WRITE(IPRINT,1001) ((I0(I),I=1,ISKP),IPLT(31,2)),(I0(I),I=ISKP2,63)  
\* ((I0(I),I=1,ISKP3),IPLT(31,4)),(I0(I),I=ISKP4,63)  
1001 FORMAT(126A1)

RETURN

END