RECEIVED BY DTIC DEC 24 1969

MASTER

GEAP-10007
AEC RESEARCH AND
DEVELOPMENT REPORT
MARCH 1969

RESTART FOR THE PAPA PROGRAM

G. O. RITZE
AIRCRAFT ENGINE GROUP
GENERAL ELECTRIC CO.
CINCINNATI, OHIO

U. S. ATOMIC ENERGY COMMISSION
CONTRACT AT(04-3)-189
PROJECT AGREEMENT 37

ATOMIC POWER EQUIPMENT DEPARTMENT
GENERAL ELECTRIC
SAN JOSE, CALIFORNIA

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

GEAP-10007
AEC Research and
Development Report
March 1969

RESTART FOR THE PAPA PROGRAM

G. O. Ritze
Aircraft Engine Group
General Electric Company
Cincinnati, Ohio

APPROVED:

S. R. Vandenberg
S. R. Vandenberg,
Project Engineer

J. L. Murray for
D. H. Imhoff, Manager
Development Engineering

Prepared for the
U.S. Atomic Energy Commission
Contract AT(04-3)-189
Project Agreement 37

Printed in U.S.A. Available from the
Clearing House for Federal Scientific and Technical Information
National Bureau of Standards, U.S. Department of Commerce
Springfield, Virginia
Price: \$3.00 per copy

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission: A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe upon privately owned rights; or B. Assumes any liability with respect to the use of, or for damages resulting from the use of, any information, apparatus, method, or process disclosed in this report.

As used in the above paragraph, the person acting on behalf of the Commission is any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor, prepares, with the Commission, or his employment with such contractor.

5896-Dev. Eng.-60
200-AG-11/69

ATOMIC POWER EQUIPMENT DEPARTMENT
GENERAL ELECTRIC
SAN JOSE, CALIFORNIA

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

T 2e

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or*
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.*

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	1
SECTION I INTRODUCTION	1
SECTION II USING RESTART	1
2.1 Initial Run (First Run)	1
2.2 Restarting a Previous Run (Second Run)	5
SECTION III PAPA RESTART PROGRAM	5
3.1 Link 23	5
3.2 Checkpoints	5
3.3 Program Flow	8
3.3.1 MAIN 33	8
3.3.2 XLINK	8
3.3.3 MASRST	8
3.3.4 MAIN23	8
3.4 Link 50	8
APPENDIX A NOTES ON TAPE FORMATS AT RESTART CHECKPOINTS	9
APPENDIX B FLOW CHARTS OF RESTART ROUTINES	21
APPENDIX C LISTINGS OF RESTART SUBROUTINES	26
APPENDIX D NOTES ON CHANGES MADE TO PAPA TO IMPLEMENT RESTART	46
DISTRIBUTION	50

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1	PAPA Plate and Panel Analysis	2
2	Deck Setup to Create Restart Tapes	3
3	Sample Comments Appearing in PAPA Restart Run	4
4	Deck Setup to Restart a Previous Run	6
5	Two Tapes for Needed Information	7
B-1	Flow Chart of MASRST	22
B-2	Flow Chart of MAIN23	23
B-3	Flow Chart of WRAP23	24
B-4	Flow Chart of WRAP50	25

ABSTRACT

The restart features of the PAPA computer program, as written in FORTRAN IV for use on a GE-635 computer, are described. This report should be considered as a users manual and a programmers reference manual. Examples of all restart messages and re-initiation features of the program are discussed.

SECTION I

INTRODUCTION

This report is intended to serve two purposes: (a) a users manual providing the information necessary to use the restart feature of the PAPA Program (Plate and Panel Analysis), and (b) a programmers reference manual describing the changes and additions made to the PAPA Program in order to implement the restart feature. It is a supplement to "PAPA - The Digital Computer Program for the Static Analysis of Structures Made Up of Plate and Panel Elements."*

*GEAP-5553, by L. Beitch and E. L. North, October 1967.

The restart procedure is designed so that a PAPA run that fails to reach a successful completion need not be rerun in its entirety. Calculations performed by the program are accumulated on tape so that a rerun can make use of calculations already performed on a previous run. This will result in a substantial reduction in machine time on subsequent runs.

It should be noted that this restart procedure does not allow the user to make changes to his model by restarting a previous run.

SECTION II

USING RESTART

Two runs are discussed here: the first run is the initial run that creates the restart tapes yet fails to go to completion, and the second run is a restart of the first run.

2.1 INITIAL RUN (FIRST RUN)

To initiate a run that saves the necessary information for restarting, the user enters `RESTR = 2`, as shown in Figure 1, on a sample input sheet. This is the only card that needs to be added to the NAME-LIST input.

Figure 2 shows the deck setup for running from the PAPA R* program tape. Note that two tapes are being created and dismounted—Tape 11 and Tape 04. These are the restart tapes.

The output from the restart run will be the same as for an ordinary PAPA run except for the appearance of comments, samples of which are shown in Figure 3. These comments verify that information needed for restarting is being saved on tape at the various checkpoints in the program.

MARCH, 1969

Page 1 of __

PAPA PLATE AND PANEL ANALYSIS

Identification Sheet

Plus signs may be omitted. Use exponents for long numbers. Enter 30×10^6 as 30E6. Enter $.15 \times 10^{-5}$ as .15E-5. Do not use slashes. Do not use commas. Enter 3,800 as 3800. Percents are entered as decimals. Enter 2.5% as .025.

col. 2

\$CQ(1)NBR=5HU3493\$

\$W(1)PRØGM=1009\$

\$W(1)PRØGM=1009\$

\$W

NAME=60H(1) E. NORTH (3) DATE (1) JUNE-67 (3) CHARGE(1) 000

ADDRES=60H(1) GE-EVENDALE (3) MAIL DRØP (1) 15 (3) EXT(1) 4049

IDENT=60H(1) CYL - CØNE - CYL(1) TEST(1) ANALYSTS

PRØGM=1031\$

\$W

TØL= 70 , reference temperature

RESTRT = , enter 2 if restart tapes are to be created and saved.

If this is a restart of a previous run, enter as
instructed by output of previous run.

ISUBRS = , if this is a restart of a previous run, enter as
instructed by output of previous run.

FIGURE 1. PAPA PLATE AND PANEL ANALYSIS

Col.	①	⑧	① 6
\$	IDENT		
\$	OPTION	FØRTTRAN	
\$	EXECUTE	DUMP	
\$	DISC	H*,B1R,100R	
\$	TAPE	R*,X9D,,NNNNN,,PAPAR*	
\$	TAPE	02,X1R	
\$	TAPE	08,X2R	
\$	TAPE	09,X3R	
\$	TAPE	11,X4D,,,,NTAP5/RESTART	
\$	TAPE	03,X5R	
\$	TAPE	07,X6R	
\$	TAPE	04,X7D,,,,NTAP12/RESTART	
\$	PERM	14,MDATA	
\$	DISC	D1,D1R,100R	
\$	LIMITS	_,42000,,_____	
\$	INCØDE	IBMF	

NAMELIST INPUT
(model)

\$ ENDJØB

FIGURE 2. DECK SETUP TO CREATE RESTART TAPES

```

*RECOVERY FILES IN LK23 HAVE BEEN CREATED
*PROGRAM IS AT RECOVERY POINT 1 FOR SUBSTRUCTURE 1
*TO RECOVER, INPUT ISUBRS=1, RESTRT=1023.0

*RECOVERY FILES IN LK23 HAVE BEEN CREATED
*PROGRAM IS AT RECOVERY POINT 2 FOR SUBSTRUCTURE 1
*TO RECOVER, INPUT ISUBRS=1, RESTRT=1023.0

*RECOVERY FILES IN LK23 HAVE BEEN CREATED
*PROGRAM IS AT RECOVERY POINT 3 FOR SUBSTRUCTURE 1
*TO RECOVER, INPUT ISUBRS=1, RESTRT=1023.0

*RECOVERY FILES IN LK23 HAVE BEEN CREATED
*PROGRAM IS AT RECOVERY POINT 4 FOR SUBSTRUCTURE 1
*TO RECOVER, INPUT ISUBRS=1, RESTRT=1023.0

*RECOVERY FILES IN LK23 HAVE BEEN CREATED
*PROGRAM IS AT RECOVERY POINT 2 FOR SUBSTRUCTURE 2
*TO RECOVER, INPUT ISUBRS=2, RESTRT=1023.0

*RECOVERY FILES IN LK23 HAVE BEEN CREATED
*PROGRAM IS AT RECOVERY POINT 3 FOR SUBSTRUCTURE 2
*TO RECOVER, INPUT ISUBRS=2, RESTRT=1023.0

*RECOVERY FILES IN LK23 HAVE BEEN CREATED
*PROGRAM IS AT RECOVERY POINT 4 FOR SUBSTRUCTURE 2
*TO RECOVER, INPUT ISUBRS=2, RESTRT=1023.0

*RECOVERY FILES IN LK23 HAVE BEEN CREATED
*PROGRAM IS AT RECOVERY POINT 5 FOR SUBSTRUCTURE 2
*TO RECOVER, INPUT ISUBRS=2, RESTRT=1023.0

*RECOVERY FILES IN LK50 HAVE BEEN CREATED
*TO RECOVER, INPUT ISUBRS=2, RESTRT=1050.0

```

FIGURE 3. SAMPLE COMMENTS APPEARING IN PAPA RESTART RUN

2.2 RESTARTING A PREVIOUS RUN (SECOND RUN)

If the initial run just discussed should fail to run to completion, the run can be restarted from the two restart tapes which were dismounted and saved. The last comment (type shown in Figure 3) appearing in the printout of the first run contains instructions for restarting. For example, if the last comment appearing in the PAPA output is

```
*RECOVERY FILES IN LK23 HAVE BEEN
  CREATED
*PROGRAM IS AT RECOVERY POINT 3
  FOR SUBSTRUCTURE 4
*TO RECOVER, INPUT ISUBRS=4,
  RESTRT=1023.0,
```

then the second run would be set up as shown in Figure 4. Three items are noted in Figure 4:

- a. The two tapes created in the previous run are mounted. The user must convey to the operator that both of these tapes are to be mounted with the "ring in" so that additional records can be written on these tapes.
- b. The values for RESTRT and ISUBRS which appeared in the last comment to be printed out in the first run are entered here for the second run. These numbers tell the program the link and the substructure in which to restart.
- c. Only the portion of the model shown here needs to be included with the second run. The rest of the input for the model is already on the restart tapes.

If the second run should also fail to go to completion, it is possible to restart again from the same two tapes.

SECTION III

PAPA RESTART PROGRAM

The PAPA Program uses seven scratch tapes for storing data. The file codes for these tape files are stored in the program in labeled common C00 under the variable names NTAP5, NTAP6, NTAP7, NTAP8, NTAP10, NTAP11, and NTAP12; the tapes will be referred to by these names.

For a substructure run, the program can be restarted in either of two links: Link 23 and Link 50. For a single structure run, the program can be restarted only in Link 50. These two links will be discussed separately.

3.1 LINK 23

The ability to restart requires that all data accumulated by the program, along with

the counters and indicators that direct the logic, be on tape files so that they will not be lost if the run should stop before completion. In Link 23, most of the needed data are stored on the seven scratch tapes, but it would be impractical to save all of the tapes for restarting. To avoid this, everything needed is channeled onto two tapes, as illustrated in Figure 5. This illustration shows that information from NTAP8, as well as the contents of two labeled common regions, is written on NTAP5. Records from NTAP6, NTAP10, and NTAP11 are written on NTAP12.

3.2 CHECKPOINTS

There are five checkpoints in subroutine MAIN23 at which restart information is

FIGURE 4. DECK SETUP TO RESTART A PREVIOUS RUN

FIGURE 5. TWO TAPES FOR NEEDED INFORMATION

gathered. See Appendix B for a flow chart of the section of MAIN23 in which the five checkpoints appear. At each checkpoint there is a call to subroutine WRAP23. This routine does the transferring of records to the restart tapes from other tapes. Before returning from WRAP23 at each checkpoint, a comment is printed out indicating that restart information is being saved.

3.3 PROGRAM FLOW

Two separate runs are discussed:

- a. The initial run which creates the restart tapes, and
- b. The second run which uses the restart tapes from the previous run in order to restart where the first run left off.

On the initial run, the program will proceed just as on a non-restart run (see verbal flow chart in GEAP-5553) until it reaches routine MAIN23 in the solution link. In MAIN23, the checkpoints are encountered and the appropriate information is written on the two restart tapes by calling subroutine WRAP23. Other than these occasional calls to WRAP23, the restart run proceeds identically to the non-restart run.

Assuming the first run aborts somewhere in the solution link, it is desirable to make a second run, using the information saved from the first run. The second run proceeds normally until it reaches subroutine MAIN33.

3.3.1 MAIN33

Here the value entered for the variable RESTRT is read in. RESTRT = 1023.0 will be read in if the run is attempting to restart in Link 23. This value of RESTRT is tested in MAIN33 and results in a call to subroutine XLINK.

3.3.2 XLINK

XLINK calls in the restart link and then calls subroutine MASRST.

3.3.3 MASRST

MASRST takes the two restart tapes and transfers records onto the other tapes, recreating the other tapes exactly as they existed on the first run at the last checkpoint before it aborted. Also, labeled common regions C00 and C23 from the previous run are read from the restart tapes. MASRST then calls XLINK, which loads in Link 23 and then calls subroutine MAIN23.

3.3.4 MAIN23

The variable IND in common C23 tells from which checkpoint the program is restarting. MAIN23 looks at this integer and transfers directly to that checkpoint in the subroutine, and from there proceeds as on an initial restart run.

3.4 LINK 50

Link 50 contains one checkpoint, at the beginning of subroutine MAIN50. Here MAIN50 calls routine WRAP50 to save the restart information on tape. At this stage of the program, there is relatively little information that needs to be saved for restarting. The records from NTAP6 and NTAP10 are copied onto NTAP5. This is the only tape that needs to be saved for restarting.

The program flow for the initial run to be restarted in Link 50 will be the same as described for Link 23 in Subsection 3.1, except it is assumed that the run has successfully completed Link 23. It then enters Link 50, encounters the checkpoint in MAIN50, and for some reason aborts in Link 50.

The second run will then also be similar to the second run described in subsection 3.1, except that subroutine MASRST will recreate the tape files as they existed at the checkpoint in Link 50 and then transfer to Link 50 to continue the first run.

APPENDIX A

NOTES ON TAPE FORMATS AT RESTART CHECKPOINTS

1. The following diagrams show the contents of each tape file before returning from the WRAP subroutine at each checkpoint.
2. The following nomenclature applies to the diagrams:
N = Number of substructures in model,
J = The particular substructure being processed in Link 23.
3. The arrows at the left of each column indicate where the file is positioned upon returning from the WRAP subroutine.
4. Brackets to the right of columns for NTAP5 and NTAP12 indicate which of the other files the records were copied from for restarting.

CHECKPOINT 1

CHECKPOINT 1 (Continued)

CHECKPOINT 2 FOR SUBSTRUCTURE J

CHECKPOINT 2 FOR SUBSTRUCTURE J. (Continued)

CHECKPOINT 3 FOR SUBSTRUCTURE J

CHECKPOINT 3 FOR SUBSTRUCTURE J (Continued)

NTAP11

NEJS, JNES
SS #1
EXT.JT.CONST.
VEC.SS#1
EXT. JT.
STIFF. MAT.
COLS. SS#1
EØF

NEJS, JNES
SS #J
EXT.JT.CONST.
VEC.SS#J
EXT. JT.
STIFF. MAT.
COLS.
SS #J

NTAP12

EØF
ELEMENT
STIFF.MAT.
& CON.VEC.
FOR SS#1
EØF

ELEMENT
STIFF. MAT.
& CON.VEC.
FOR SS#N
EØF
EØF
JN, NJ
DEF
NDM, KBC,
KAPC, BES.

JN, NJ
DEF
NDM, KBC,
KAPC, BES.
EØF
NEJ, JNES
DEF
FØRCE
SPRING
NDM, KBC,
KAPC, BES.
EØF
EØF
EØF
EØF

DUMMY
PLOT
EØF
PLOT
INDICATORS
EØF
EØF
CON.VEC. SS#1
EXT.JT.COLS.
SS #1
EØF

CON.VEC. SS#J
EXT.JT.COLS.
SS #J
EØF
REVERSE
DECOM. MAT.
SS #J
EØF
EØF

NEJS, JNES
SS#1
EXT.JT.CONST.
VEC. SS#1
EXT. JT.
STIFF. MAT.
COLS. SS#1
EØF

NEJS, JNES
SS #J
EXT.JT.CONST
VEC.SS#J
EXT. JT.
STIFF. MAT.
COLS. SS#J
EØF
EØF

SS#1

SS#N

CHECKPOINT 4 FOR SUBSTRUCTURE J

CHECKPOINT 4 FOR SUBSTRUCTURE J (Continued)

NTAP11

NEJS, JNES SS #1	
EXT.JT.CONST. VEC. SS#1	
EXT.JT. STIFF.MAT. COLS. SS#1	
EØF

NEJS, JNES SS #J	
EXT.JT.CONST. VEC. SS#J	
EXT. JT. STIFF.MAT. COLS. SS#J	
EØF	

NTAP12

EØF
ELEMENT
STIFF.MAT.
& CON.VEC.
FOR SS#1
EØF
ELEMENT
STIFF.MAT.
& CON.VEC.
FOR SS#N
EØF
EØF
JN, NJ
DEF
NDM, KBC,
KAPC, BES.
JN, NJ
DEF
NDM, KBC,
KAPC, BES.
EØF
NEJ, JNES
DEF
FØRCE
SPRING
NDM, KBC,
KAPC, BES.
EØF
EØF
EØF
EØF

SS#1

SS#N

DUMMY	
PLØT	
EØF	
PLØT	
INDICATORS	
EØF	
EØF	
CON.VEC.SS#1	
EXT.JT.COLS.	
SS #1	
EØF	
CON.VEC.SS#J	
EXT.JT.COLS.	
SS #J	
EØF	
REVERSE	
DECOM. MAT.	
SS #J	
EØF	
EØF	

NEJS, JNES	
SS #1	
EXT.JT.CONST.	
VEC. SS#1	
EXT. JT.	
STIFF.MAT.	
CØLS. SS#1	
EØF

NEJS, JNES	
SS #J	
EXT.JT.CONST.	
VEC.SS#J	
EXT. JT.	
STIFF.MAT.	
CØLS. SS#J	
EØF	
EØF	

CHECKPOINT 5

CHECKPOINT 5 (Continued)

NTAP11

NEJS, JNES
SS #1
EXT.JT.CONST.
VEC. SS#1
EXT.JT.
STIFF.MAT.
COLS. SS#1
EØF

NEJS, JNES
SS #N
EXT.JT.CONST.
VEC. SS#N

NTAP12

EØF
ELEMENT
STIFF.MAT.
& CON.VEC.
FOR SS#1
EØF

ELEMENT
STIFF.MAT.
& CØN.VEC.
FOR SS#N

EØF
EØF

JN, NJ
DEF
NDM, KBC,
KAPC, BES.

JN, NJ
DEF
NDM, KBC,
KAPC, BES.

EØF

NEJ, JNES

DEF

FØRCE

SPRING

NDM, KBC,
KAPC, BES.

EØF

EØF

EØF

EØF

DUMMY
PLØT
EØF
PLØT
INDICATORS
EØF
EØF
CØN.VEC.SS#1
EXT.JT.COLS.
SS #1
EØF

CON.VEC.SS#N
EXT.JT.COLS.
SS #N

EØF

REVERSE

DECOM.MAT.

SS #N

EØF

EØF

NEJS, JNES
SS #1
EXT.JT.CONST.
SS #1
EXT.JT.
STIFF.MAT.
COLS. SS#1
EØF

NEJS, JNES
SS #N

EXT.JT.CONST.
VEC. SS#N

EXT. JT.
STIFF. MAT.
CØLS. SS#N

EØF

EØF

CHECKPOINT IN MAIN50

APPENDIX B

FLOW CHARTS OF RESTART ROUTINES

B-1. MASRST

B-2. MAIN23

B-3. WRAP23

B-4. WRAP50

FIGURE B-1. FLOW CHART OF MASRST

FIGURE B-2. FLOW CHART OF MAIN 23

FIGURE B-3. FLOW CHART OF WRAP 23

FIGURE D-4. FLOW CHART OF WRAP 50

APPENDIX C

LISTINGS OF RESTART SUBROUTINES

1. MASRST
2. CØPTAP
3. WRAP50
4. WRAP23
5. MAIN23

```

$      INCODE  IBMF
CMASRST      SUBR. MASRST
      SUBROUTINE MASRST
C$          C00
C$          C00
CC00        C00
      COMMON/C00/
1  PROGM,  AID,      IOF,      NTAP2,  NTAP3,  NTAP4,  NTAP5,  NTAP6,
2  NTAP7,  NTAP8,  NTAP10, NTAP11, NTAP12, NTAP13, BITS,  IBITS,
3  LK00,   LK0999, LK31,   LK32,   LK30,   LK20,   LK21,   LK22,
4  LK23,   LK50,   TOL,    MS,     OPERA,  NOITER, STATI,  SUBCAS,
5  NAID,   CPS,    NU3340, INER,   LK09,   LK99,   LK27,
6  LK24,   LK25,   LK26,   LK3132, LK2021, DISC1,  LK2223, LK28,
7  LK33,   LK2425, LK29,   LK2930, LK18,   LK19,   LK2728,
8  LK34,   LK1818,  LKRS,
9  DUMY(4), RESTRT, LKRS23, ISUBRS, LK59,
A  IVIBR,YKK, LK6162, IBCKLD, LK70,  PLTB(30), INDI,  NTAP15
      DIMENSION DUMC00(105)
      EQUIVALENCE (DUMC00(1),PROGM)
C  PLTB(1) THRU PLTB(30) AND DUMC00(103) THRU DUMC00(105) UNUSED.
CC00        C00
C$          END
C$          C23
C$          C23
CC23
      COMMON/C23/
1  IND,      MNSUB,    L2,      IRHS,
2  NS,      LNSUB,    NRHS,
3  NEX,      NJ,      NEJS,    MSE,      NOPCHG,  NORM,
4  NORMSB,  NA,      NEJS6,    MRHS,    NOP,      JN(200),
5  JNES(50), JNEX(200)
      DIMENSION DUMC23(473)
      EQUIVALENCE (IND,DUMC23(1))
      COMMON/C23/
1  F(1200),EXJC(1200),LFSI,LASI,LF4,LF6,LFSI6
C$          END
      DIMENSION A(7600)
      NREC = 0
      IF(RESTRT.EQ.1050.) GO TO 5
      IF(RESTRT.EQ.1023.) GO TO 50
      CALL ERROR
C
C...      SET UP FILES FOR RESTARTING IN LINK50
C
C...      POSITION NTAP5 IN FRONT OF FILES BELONGING ON NTAP6
5  CALL FNREW(NTAP5)
      CALL FILPAS(4,NTAP5)
      CALL FNREW(NTAP6)
C...      COPY FILES SAVED FROM NTAP6 BACK ONTO NTAP6
      IF ( ISUBRS .EQ. 1 ) CALL COPTAP(NTAP5,NTAP6,1,A)
      IF ( ISUBRS .GT. 1 ) CALL COPTAP(NTAP5,NTAP6,4,A)
C...      POSITION NTAP5 IN FRONT OF PLOT FILES
      CALL FNPT(1,NTAP5)
C...      COPY 2 PLOT FILES SAVED ON NTAP5 BACK ONTO NTAP10
      CALL FNREW(NTAP10)
      CALL COPTAP(NTAP5,NTAP10,2,A)
C...      POSITION NTAP5 IN FRONT OF COMMON
      CALL FNPT(1,NTAP5)

```

```

C...   READ COMMON C00
      CALL FNRTB(NTAP5,1,DUMC00,105,IER)
C...   REPOSITION FILES FOR RESTARTING IN LINK50
      CALL FNREW(NTAP6)
      CALL FILPAS(1,NTAP6)
      CALL FNREW(NTAP5)
      CALL FILPAS(2,NTAP5)
      WRITE(6,31)
31  FORMAT(1H1/16HRECOVERY IN LK50/1H1)
      RESTRT = 0.0
      CALL XLINK(1050,0)
      CALL ERROR
C
C...   SET UP FILES FOR RESTARTING IN LINK23
C
C...   NSUB TELLS WHICH SUBSTRUCTURE WE ARE RESTARTING IN
50  NSUB = ISUBRS
      CALL FNREW(NTAP7)
      CALL FNWEF(NTAP7)
      CALL FNREW(NTAP7)
      CALL FNREW(NTAP8)
      CALL FNWEF(NTAP8)
C...   POSITION NTAP5 IN FRONT OF COMMONS
      CALL FNREW(NTAP5)
      NPAS=2*NSUB+5
      IF(NSUB.EQ.0) NPAS=NPAS 1
      CALL FILPAS(NPAS,NTAP5)
C...   READ COMMONS C00 AND C23
      CALL FNRTB(NTAP5,2,DUMC00,105,DUMC23,473,IER)
C...   POSITION NTAP5 IN FRONT OF FIRST STIFFNESS MATRIX
      CALL FNREW(NTAP5)
      CALL FILPAS(4,NTAP5)
C...   CALC. NO. OF STIFF. MATRICES AND DECOM. MATRICES
      NPAS=2*NSUB
C
C...   WRITE STIFFNESS MATRICES AND DECOM. MATRICES ONTO NTAP8
C
      IF ( NPAS .EQ. 0 )   GO TO 60
      CALL COPTAP(NTAP5,NTAP8,NPAS,A)
C
C
60  CALL FNREW(NTAP6)
C...   POSITION NTAP12 IN FRONT OF NTAP6 RECORDS
      CALL FNREW(NTAP12)
      NPAS = SUBCAS  2.0
      CALL FILPAS(NPAS,NTAP12)
C
C...   WRITE 4 NTAP6 FILES SAVED ON NTAP12 BACK ONTO NTAP6
C
      CALL COPTAP(NTAP12,NTAP6,4,A)
C
C
      CALL FNREW(NTAP10)
C...   POSITION NTAP12 IN FRONT OF PLOT RECORDS SAVED FROM NTAP10
      CALL FNPT(1,NTAP12)
C
C...   WRITE 2 PLOT FILES SAVED ON NTAP12 BACK ONTO NTAP10
C

```

```

      CALL COPTAP(NTAP12,NTAP10,2,A)
C
      IF ( IND .GT. 1 )   GO TO 200
C
C...      REPOSITION FILES TO RESTART AT CHECK POINT ONE
C
      CALL FNPT(1,NTAP5)
      GO TO 900
C
      200 CALL FNWEF(NTAP10)
C...      POSITION NTAP12 IN FRONT OF CONST.VEC.AND EXT.JT.MAT.SAVED FROM NTAP10
      CALL FNPT(1,NTAP12)
C
C...      WRITE CONST.VEC.AND EXT.JT.MATRICES ONTO NTAP10
C
      GO TO 230
C...      WRITE A CONST.VEC.OR AN EXT.JT.MAT.COLUMN ONTO NTAP10
      220 CALL FNWTB(NTAP10,1,A,IER-10000,IER)
C...      READ A CONST. VEC. OR AN EXT. JT. MATRIX COLUMN FROM NTAP12
      230 CALL FNRTB(NTAP12,1,A,-7600,IER)
      IF ( IER .NE. 8 )   GO TO 220
      CALL FNWEF(NTAP10)
      CALL FNRTB(NTAP12,1,A,-7600,IER)
      IF ( IER .NE. 8 )   GO TO 220
C
C
C
C...      COPY EXT. JT. STIFFNESS MATRICES BACK ONTO NTAP11
C
      CALL FNREW(NTAP11)
      GO TO 330
      320 CALL FNWTB(NTAP11,1,A,IER-10000,IER)
      330 CALL FNRTB(NTAP12,1,A,-7600,IER)
      IF ( IER .NE. 8 )   GO TO 320
      CALL FNWEF(NTAP11)
      CALL FNRTB(NTAP12,1,A,-7600,IER)
      IF ( IER .NE. 8 )   GO TO 320
C
C
      GO TO ( 900 , 2000 , 3000 , 4000 , 5000 ) , IND
C
C...      REPOSITION FILES TO RESTART AT CHECK POINT TWO

      2000 CALL FNPT(-2,NTAP12)
      CALL FNREW(NTAP8)
      NPAS=2*NSUB
      CALL FILPAS(NPAS,NTAP8)
      CALL FILPAS(-3,NTAP10)
      CALL FNPT(2,NTAP10)
      IF ( NSUB .GT. 1 )   GO TO 2300
      CALL FNREW(NTAP11)
      GO TO 2400
      2300 CALL FILPAS(-2,NTAP11)
      CALL FNPT(1,NTAP11)
      2400 CALL FNPT(2,NTAP11)
      CALL FNREW(NTAP6)
      NPAS=3*NSUB
      CALL FNPT(NPAS,NTAP6)

```

```
CALL FNPT(1,NTAP5)
GO TO 900
```

```
C
C...      REPOSITION FILES TO RESTART AT CHECK POINT THREE
C
```

```
3000 CALL FNPT(-2,NTAP12)
      CALL FNREW(NTAP8)
      NPAS=2*NSUB
      CALL FILPAS(NPAS,NTAP8)
      CALL FILPAS(-3,NTAP10)
      NPAS=2+L2
      CALL FNPT(NPAS,NTAP10)
      IF ( NSUB .GT. 1 ) GO TO 3700
      CALL FNREW(NTAP11)
      NPAS = NPAS - 1
      GO TO 3800
3700 CALL FILPAS(-2,NTAP11)
3800 NPAS = NPAS + 1
      CALL FNPT(NPAS,NTAP11)
      CALL FNREW(NTAP6)
      NPAS=3*NSUB
      CALL FNPT(NPAS,NTAP6)
      CALL FNPT(1,NTAP5)
      GO TO 900
```

```
C
C...      REPOSITION FILES TO RESTART AT CHECK POINT FOUR
C
```

```
4000 NREC = -3
```

```
C
C...      REPOSITION FILES TO RESTART AT CHECK POINT FIVE
C
```

```
5000 NREC = NREC + 5
      CALL FNREW(NTAP12)
      NPAS=1+NSUB
      CALL FILPAS(NPAS,NTAP12)
      CALL FNPT(NREC,NTAP5)
      CALL FNREW(NTAP6)
      NPAS=3*NSUB
      CALL FNPT(NPAS,NTAP6)
      CALL FNPT(-1,NTAP10)
      CALL FILPAS(-1,NTAP10)
      CALL FNPT(1,NTAP10)
900  NSUB=SUBCAS
      RESTRT = 2323.0
      WRITE(6,2002)
2002 FORMAT(1H1/16HRECOVERY IN LK23/1H1)
      CALL XLINK(1023,2)
      CALL ERROR
1000 RETURN
      END
```

```
$      INCODE  IBMF
CCOPTAP      SUBR. COPTAP
      SUBROUTINE COPTAP(NT1,NT2,NFILES,A)
C...      THIS ROUTINE COPIES NFILES FILES FROM NT1 ONTO NT2
C...      NT1 = TAPE FROM WHICH FILES ARE BEING COPIED
C...      NT2 = TAPE ONTO WHICH FILES ARE BEING WRITTEN
C...      NFILES = NO. OF FILES TO COPY
      DIMENSION A(1)
      DO 300 I = 1,NFILES
      GO TO 200
100 CALL FNWTB(NT2,1,A,IER-10000,IER)
200 CALL FNRTB(NT1,1,A,-7600,IER)
      IF ( IER .NE. 8 ) GO TO 100
      CALL FNWEF(NT2)
300 CONTINUE
      RETURN
      END
```

```

$      INCODE  IBMF
CWRAP50      SUBR. WRAP50
      SUBROUTINE WRAP50
C$          C00
C$          C00
CC00       C00
      COMMON/C00/
1  PROGM,  AID,    IOF,    NTAP2,  NTAP3,  NTAP4,  NTAP5,  NTAP6,
2  NTAP7,  NTAP8,  NTAP10, NTAP11, NTAP12, NTAP13, BITS,  IBITS,
3  LK00,   LK0999, LK31,   LK32,   LK30,   LK20,   LK21,   LK22,
4  LK23,   LK50,   TOL,    MS,     OPERA,  NOITER, STATI,  SUBCAS,
5  NAID,   CPS,    NU3340, INER,   LK09,   LK99,   LK27,
6  LK24,   LK25,   LK26,   LK3132, LK2021, DISC1,  LK2223, LK28,
7  LK33,   LK2425, LK29,   LK2930, LK18,   LK19,   LK2728,
8  LK34,   LK1818, LKRS,
9  DUMY(4), RESTRT, LKRS23, ISUBRS, LK59,
A  IVIBR,YKK, LK6162, IBCKLD, LK70,  PLTB(30), INDI,  NTAP15
      DIMENSION DUMC00(105)
      EQUIVALENCE (DUMC00(1),PROGM)
C  PLTB(1) THRU PLTB(30) AND DUMC00(103) THRU DUMC00(105) UNUSED.
CC00       C00
C$          END
C$          C50
C$          C50
CC50       C50
      COMMON /C50/
1  TOTFOR, BES,    XM5,    XM4,    XM3,    XM2,    XM1,    XL,
2  TWGT,   TOTWGT, TOTPE,  TOTKE,  SPROGM, SP,    RPM,    NS,
3  NLINE,  NJ,     NDM,    LL,     LINE,   L,      KK,
4  KBC,    KAPC,   K,      JJ,     J,      ISKIP,  IPROGM, IDLM2,
5  IDUM1,  II,     IFREQ,  IERCOD, IE,     IDET,   ID,     I,
6  FREQ,   EN,     EK,     E,      DET,    DEF,    DAPREV, BUFFER,
7  XY,     XZ,     YZ,     DIMEN,  DRAW,   XMAX,   XMIN,   X,
8  Y,      Z,      INTR1,  INTR2,  ISUB,   BIG,    TOTE,   JNE,
9  JN
      COMMON /C50/
1ZCG,XPMAX,XPMIN,YPMAX,YPMIN,SF,DE(24,24),NJE,BUFF(400),MM,NN,M,N
C
      DIMENSION TMM(6,6), TMN(6,6), GG3D(18,18)
      DIMENSION X(30),Y(30),Z(30),INTR1(24),INTR2(24),JE(200)
      DIMENSION
1  XP(6,6),XM(6,6),XI(6,6),TOTE(6,200),TOTFOR(6,200),TML(6,6),
2  TMK(6,6),TMJ(6,6),TMI(6,6),SOLUTN(1200),S(24),JNE(200),JN(200),
3  GGJJ(6,6),GGJI(6,6),GGII(6,6),GGG(12,12),GG(24,24),
4  FRRJ(6),FRRJ(6),FPP(24),FPN(24),FN(66),FLO(120),FFJ(6),FFI(6),
5  FFF(12),F(66),EFF(120),DUMMY(6,6),DN(66),DEFX(12),DEFT(24),
6  DEFN(24),DEFJ(6),DEFI(6),DEFE(6,200),DEF(6,200),
7  DAPREV(400),D(66),CONINV(12,12),BUFFER(6),BES(3400),BCDFL(1200)
      DIMENSION TM(6,6),P(6,6),RHS(6),RHS1(6)
C
      EQUIVALENCE
1(BES( 577),D,JE),(BES( 643),DEFI),(BES( 649),DEFJ),
2(GG(501),GGJJ,TMM), (GG(537),TMN), (GG,GG3D)
      EQUIVALENCE
1 (BES( 577),D,JE),(BES( 643),DEFI),(BES( 649),DEFJ),
2 (BES( 655),DEFN),(BES( 679),DEFT),(BES( 703),DEFX),
3 (BES( 715),DN ),(BES( 781),EFF ),(BES( 901),F ),
4 (BES( 967),FFF ),(BES( 979),FFI ),(BES( 985),FFJ ),

```

```

5 (BES( 991),FLO ),(BES(1111),FN ),(BES(1177),FPN ),
6 (BES(1201),FPP ),(BES(1225),FRRJ),(BES(1231),FRRJ),
7 (BES(1237),GGG ),(BES(1381),S ),(BES(1405),TMI ),
8 (BES(1441),TMJ ),(BES(1477),TMK ),(BES(1513),TML ),
9 (BES(1549),XI ),(BES(1585),XM ),(BES(1621),XP )

```

EQUIVALENCE

```

1 (BES(1657),DEFE),(TOTFOR,BCDFL ),(TOTE,SOLUTN )

```

EQUIVALENCE(TOTFOR(1,1),TM),(TOTFOR(1,7),P),

```

1 (TOTFOR(1,13),RHS),(TOTFOR(1,14),RHS1)

```

CC50 C50

C\$ END

C... THIS SUBROUTINE SAVES ON NTAP5 ALL INFORMATION NEEDED FOR
C RESTARTING AT THIS POINT

C... POSITION NTAP10 AFTER PLOT FILES

CALL FNREW(NTAP10)

CALL FILPAS(2,NTAP10)

C... TEMPORARILY STORE BLOCK OF 5 RECORDS AT END OF NTAP5 ONTO NTAP10

CALL COPTAP(NTAP5,NTAP10,1,BES)

C... POSITION NTAP10 IN FRONT OF 5 RECORDS JUST WRITTEN ON NTAP10

CALL FILPAS(-2,NTAP10)

CALL FNPT(1,NTAP10)

C... POSITION NTAP5

CALL FNREW(NTAP5)

CALL FILPAS(2,NTAP5)

C... WRITE BLOCK OF 5 RECORDS BACK ONTO NTAP5

CALL COPTAP(NTAP10,NTAP5,1,BES)

CALL FNWEF(NTAP5)

C... COPY FILES FROM NTAP6 ONTO NTAP5

CALL FNREW(NTAP6)

IF (SUBCAS .EQ. 1.0) CALL COPTAP(NTAP6,NTAP5,1,BES)

IF (SUBCAS .GT. 1.0) CALL COPTAP(NTAP6,NTAP5,4,BES)

CALL FNWEF(NTAP5)

C... COPY TWO PLOT FILES FROM NTAP10 ONTO NTAP5

CALL FNREW(NTAP10)

CALL COPTAP(NTAP10,NTAP5,2,BES)

CALL FNWEF(NTAP5)

C... WRITE COMMON C00 ONTO NTAP5

CALL FNWTB(NTAP5,1,DUMC00,105,IER)

CALL FNWEF(NTAP5)

CALL FNWEF(NTAP5)

C... REPOSITION FILES FOR RETURNING TO MAIN50

CALL FNREW(NTAP5)

CALL FILPAS(2,NTAP5)

C

CALL FNREW(NTAP6)

CALL FILPAS(1,NTAP6)

C

ISUBCS = SUBCAS

WRITE(6,9001) ISUBCS

9001 FORMAT(1H1/42H *RECOVERY FILES IN LK50 HAVE BEEN CREATED/

\$ 27H *TO RECOVER, INPUT ISUBRS=,13,14H RESTR=1050.0/1H1)

RETURN

END


```

$      INCODE  IBMF
CWRAP23      SUBR.WRAP23
      SUBROUTINE WRAP23(A)
C$          C00
C$          C00
CC00        C00
      COMMON/C00/
1  PROG,  AID,      IOF,      NTAP2,  NTAP3,  NTAP4,  NTAP5,  NTAP6,
2  NTAP7,  NTAP8,  NTAP10, NTAP11, NTAP12, NTAP13, BITS,  IBITS,
3  LK00,   LK0999, LK31,    LK32,    LK30,    LK20,    LK21,    LK22,
4  LK23,   LK50,   TOL,     MS,      OPERA,  NOITER, STATI,  SUBCAS,
5  NAID,   CPS,    NU3340, INER,    LK09,    LK99,    LK27,
6  LK24,   LK25,   LK26,    LK3132, LK2021, DISC1,  LK2223, LK28,
7  LK33,   LK2425, LK29,    LK2930, LK18,    LK19,    LK2728,
8  LK34,   LK1818, LKRS,
9  DUMY(4), RESTR,  LKRS23, ISUBRS,  LK59,
A  IVIBR,YKK, LK6162, IBCKLD, LK70,  PLTB(30), INDI,  NTAP15
      DIMENSION DUMC00(105)
      EQUIVALENCE (DUMC00(1),PROG)
C  PLTB(1) THRU PLTB(30) AND DUMC00(103) THRU DUMC00(105) UNUSED.
CC00        C00
C$          END
C$          C23
C$          C23
CC23
      COMMON/C23/
1  IND,      MNSUB,    L2,      IRHS,
2  NS,       LNSUB,    NRHS,
3  NEX,      NJ,       NEJS,    MSE,      NOPCHG,  NORM,
4  NORMSB,   NA,       NEJS6,   MRHS,    NOP,      JN(200),
5  JNES(50), JNEX(200)
      DIMENSION DUMC23(473)
      EQUIVALENCE (IND,DUMC23(1))
      COMMON/C23/
1  F(1200),EXJC(1200),LFSI,LASI,LF4,LF6,LFSI6
C$          END
C...      THIS ROUTINE SAVES INFORMATION ON TAPE FOR RESTARTING IN LINK 23
C...      INFO. ON NTAP6 IS STORED ON NTAP12
C...      INFO. ON NTAP8 IS STORED ON NTAP5
C...      INFO. ON NTAP10 IS STORED ON NTAP12
C...      INFO. ON NTAP11 IS STORED ON NTAP12
      DIMENSION A(7600)
G...      MNSUB IS THE SUBSTRUCTURE WE ARE IN
      MNSUB = LNSUB
C...      NSUB IS THE TOTAL NO. OF SUBSTRUCTURES IN THIS CASE
      NSUB = SUBCAS
C...      BRANCH ON IND, WHICH TELLS WHICH CHECK POINT WE ARE AT
      GO TO ( 1000 , 2000 , 3000 , 4000 , 5000 ) , IND
C
C...      WE ARE AT CHECK POINT ONE
C
C...      WE HAVENT-BEGUN PROCESSING SUBSTRUCTURES IN PHASE 1 OF MAIN23
1000  LNSUB = 0
      MNSUB = 0
      CALL FNWEF(NTAP5)
      CALL FNWEF(NTAP5)
C...      WRITE COMMONS C00 AND C23 ONTO NTAP5
      CALL FNWTB(NTAP5,2,DUMC00,105,DUMC23,473,IER)

```

```

C...      POSITION NTAP12 AT END OF STIFFNESS MATRICES
CALL FILPAS(NSUB+1,NTAP12)
C
C...      COPY 4 FILES FROM NTAP6 ONTO NTAP12
C
CALL FNREW(NTAP6)
CALL COPTAP(NTAP6,NTAP12,4,A)
C
CALL FNWEF(NTAP12)
C
C...      COPY TWO PLOT FILES FROM NTAP10 ONTO NTAP12
C
CALL FNREW(NTAP10)
CALL COPTAP(NTAP10,NTAP12,2,A)
C
CALL FNWEF(NTAP12)
C...      REPOSITION FILES FOR RETURNING TO MAIN23
CALL FNREW(NTAP6)
CALL FNPT(1,NTAP10)
CALL FNREW(NTAP12)
CALL FNPT(1,NTAP12)
GO TO 9000
C
C...      WE ARE AT CHECK POINT TWO
C
C...      POSITION NTAP5 IN FRONT OF COMMONS
2000 CALL FNPT(-1,NTAP5)
C...      IF QTHR THAT FIRST SUBSTRUCTURE, POSITION NTAP5 IN FRONT
C...      OF EOF CLOSING OFF NTAP8 INFORMATION
IF ( MNSUB .GT. 1 ) CALL FILPAS(-1,NTAP5)
C...      POSITION NTAP8 IN FRONT OF STIFFNESS MATRIX FOR THIS SUBSTRUCTURE
CALL FILPAS(-2,NTAP8)
CALL FNPT(1,NTAP8)
C
C...      COPY STIFFNESS MATRIX AND DECOM. MATRIX FROM NTAP8 ONTO NTAP5
C
CALL COPTAP(NTAP8,NTAP5,2,A)
CALL FNWEF(NTAP5)
C...      WRITE COMMONS C00 AND C23 ONTO NTAP5
CALL FNWTB(NTAP5,2,DUMC00,105,DUMC23,473,IER)
CALL FNWEF(NTAP5)
CALL FNWEF(NTAP5)
C...      POSITION NTAP10 IN FRONT OF CONST. VEC. FOR THIS SUBSTRUCTURE
CALL FILPAS(-1,NTAP10)
CALL FNPT(1,NTAP10)
C...      POSITION NTAP12
CALL FILPAS(NSUB+8,NTAP12)
C
C...      COPY CONST. VEC. AND EXT. JT. COLS. FROM NTAP10 ONTO NTAP12
C
CALL COPTAP(NTAP10,NTAP12,2,A)
C
CALL FNWEF(NTAP12)
C
C...      COPY ONE FILE FOR EACH PREVIOUS SUBSTRUCTURE FROM NTAP11 ONTO NTAP12
C
NFILES = LNSUB - 1
CALL FNREW(NTAP11)

```

```

      IF ( NFILES .NE. 0 ) CALL COPTAP(NTAP11,NTAP12,NFILES,A)
C
C...   COPY 2 RECORDS FOR THIS SUBSTRUCTURE FROM NTAP11 ONTO NTAP12
      DO 2500 I = 1,2
      CALL FNRTB(NTAP11,1,A,-7600,IER)
      CALL FNWTB(NTAP12,1,A,IER-10000,IER)
2500  CONTINUE
      CALL FNWEF(NTAP12)
      CALL FNWEF(NTAP12)
C...   REPOSITION FILES FOR RETURNING TO MAIN23
      CALL FILPAS(-2,NTAP8)
      CALL FNPT(1,NTAP8)
      CALL FNPT(-3,NTAP5)
      CALL FILPAS(-3,NTAP10)
      CALL FNPT(2,NTAP10)
      CALL FNPT(-2,NTAP12)
      GO TO 9000
C
C...   WE ARE AT CHECK POINT THREE
C
C...   WRITE COMMONS C00 AND C23 ONTO NTAP5
3000  CALL FNWTB(NTAP5,2,DUMC00,105,DUMC23,473,IER)
      CALL FNWEF(NTAP5)
      CALL FNWEF(NTAP5)
C...   NS TELLS THE NO. OF EXT. JT. STIFF. MAT. COLS. ON NTAP11
C...   FOR THIS SUBSTRUCTURE
      CALL FNPT(-NS,NTAP11)
C
C...   COPY EXT. JT. STIFF. MAT. FROM NTAP11 ONTO NTAP12
C
      DO 3500 I = 1,NS
      CALL FNRTB(NTAP11,1,A,-7600,IER)
      CALL FNWTB(NTAP12,1,A,IER-10000,IER)
3500  CONTINUE
      CALL FNWEF(NTAP12)
      CALL FNWEF(NTAP12)
C...   REPOSITION FILES FOR RETURNING TO MAIN23
      CALL FNPT(-3,NTAP5)
      CALL FNPT(-2,NTAP12)
      GO TO 9000
C
C...   WE ARE AT CHECK POINT FOUR
C
C...   WRITE COMMONS C00 AND C23 ONTO NTAP5
4000  CALL FNWTB(NTAP5,2,DUMC00,105,DUMC23,473,IER)
      CALL FNWEF(NTAP5)
      CALL FNWEF(NTAP5)
C...   REPOSITION FILES FOR RETURNING TO MAIN23
      CALL FNPT(-2,NTAP5)
      CALL FNREW(NTAP12)

```

```

C...      POSITION NTAP12 IN FRONT OF STIFFNESS MATRICES FOR NEXT SUBSTRUCTURE
          CALL FILPAS(LNSUB+1,NTAP12)
          GO TO 9000
C
C...      WE ARE AT CHECK POINT FIVE
C
5000 CALL FNPT(-1,NTAP5)
C...      WRITE COMMONS C00 AND C23 ONTO NTAP5
          CALL FNWTB(NTAP5,2,DUMC00,105,DUMC23,473,IER)
          CALL FNWEF(NTAP5)
          CALL FNWEF(NTAP5)
C...      WRITE INDICATORS ON NTAP5
          CALL FNWTB(NTAP5,7,MS,1,OPERA,1,NOITER,1,STATI,1,SUBCAS,1,NAID,1,
$ CPS,1,IER)
          CALL FNWEF(NTAP5)
          CALL FNWEF(NTAP5)
          CALL FNPT(-2,NTAP5)
9000 WRITE(6,9001) IND,LNSUB,LNSUB
9001 FORMAT(1H0,41H*RECOVERY FILES IN LK23 HAVE BEEN CREATED/
130H *PROGRAM IS AT RECOVERY POINT,12,17H FOR SUBSTRUCTURE,13/
227H *TO RECOVER, INPUT ISUBRS=,13,15H ,RESTR1=1023.0//)
          RETURN
          END

```

```

$      INCODE  IBMF
CMAIN23      MAIN23
      SUBROUTINE MAIN23
C$          C00
CC00        C00
      COMMON/C00/
1  PROGM,  AID,    IOF,    NTAP2,  NTAP3,  NTAP4,  NTAP5,  NTAP6,
2  NTAP7,  NTAP8,  NTAP10, NTAP11, NTAP12, NTAP13, BITS,  IBITS,
3  LK00,   LK0999, LK31,   LK32,   LK30,   LK20,   LK21,   LK22,
4  LK23,   LK50,   TOL,    MS,     OPERA,  NOITER, STAT1,  SUBCAS,
5  NAID,   CPS,    NU3340, INER,   LK09,   LK99,   LK27,
6  LK24,   LK25,   LK26,   LK3132, LK2021, DISC1,  LK2223, LK28,
7  LK33,   LK2425, LK29,   LK2930, LK18,   LK19,   LK2728,
8  LK34,   LK1818, LKRS,
9  DUMY(4), RESTRT, LKRS23, ISUBRS, LK59,
A  IVIBR,YKK, LK6162, IBCKLD, LK70,  PLTB(30), INDI,  NTAP15
      DIMENSION DUMC00(105)
      EQUIVALENCE (DUMC00(1),PROGM)
C  PLTB(1) THRU PLTB(30) AND DUMC00(103) THRU DUMC00(105) UNUSED.
CC00        C00
C$          END
C$          C23
C$          C23
CC23
      COMMON/C23/
1  IND,     MNSUB,   L2,     IRHS,
2  NS,      LNSUB,  NRHS,
3  NEX,     NJ,     NEJS,   MSE,   NOPCHG,  NORM,
4  NORMSB,  NA,     NEJS6,  MRHS,  NOP,     JN(200),
5  JNES(50), JNEX(200)
      DIMENSION DUMC23(473)
      EQUIVALENCE (IND,DUMC23(1))
      COMMON/C23/
1  F(1200),EXJC(1200),LFSI,LASI,LF4,LF6,LFSI6
C$          END
      DIMENSION A(15200),IA(15200),DEF(1200),
1  DETVC(100),IDETVC(100)
C
      EQUIVALENCE(A,IA)
C
C
C
C
C
C
C
C*****INITIALIZE
C
      IF ( RESTRT .NE. 2323.0 )   GO TO 10
      NSUB = SUBCAS
      RESTRT = 2.0
      IF ( IND .NE. 1 )   GO TO 15
C
C
10  CALL FNREW(NTAP6)
      CALL FNREW(NTAP11)
      CALL FNREW(NTAP12)
      CALL FNREW(NTAP8)

```

RESTART
RESTART
RESTART
RESTART

```

CALL FNWEF(NTAP8)
CALL FNWEF(NTAP10)
CALL FNPT(1,NTAP12)
15 LFSI=1200
   LFSI6=LFSI*6
   LF6=LFSI/6
   LF4=LF6/4
   LASI=15200
   NORM=0
   NORMSB=0
   IDC=0
   IDCR=0
   NOPCHG=0
   IF ( IND .GE. 5 )   GO TO 500
   IF ( IND .GT. 1 )   GO TO 90
C*****END INITIALIZATION
C
C      READ IN EXTERNAL JOINT NUMBERS
CALL FILPAS(1,NTAP6)
CALL FNRTB(NTAP6,2,NEX,1,JNEX,LF6,IERCOD)
CALL TPTEST(NTAP6 ,IERCOD)
CALL FNREW(NTAP6)
MSE=6*NEX
IF ( IND .EQ. 1 )   GO TO 90
C
C
C      READ IN PERTINANT INDICATOR
CALL FNRTB(NTAP5,7,MS,1,OPERA,1,NOITER,1,STATI,1, SUBCAS,1,NAID,1,
1  CPS,1,IERCOD)
CALL TPTES2(NTAP5,IERCOD)
NSUB=SUBCAS
NOP=OPERA
C
C      NOP=1 RESPONSE
C      NOP=2 DETERMINANT VALUE
C      NOP=3 STATIC CASE
C      NOP=4 DETERMINANT VALUE
C      NOP=5 GET MODE SHAPE
C
GO TO (40,40,100,40,40),NOP
40 CALL FNPT(1,NTAP7)
GO TO 905
C
C
C*****PHAS1   SOLVE SUBSTRUCTURES TO GENERATE
C      COEFFICIENTS OF EXTERNAL JOINT SYSTEM.
C
100 IF ( RESTRT .NE. 2.0 )   GO TO 90
   IND = 1
   CALL WRAP23(A)
C
90 DO 400 LNSUB=1,NSUB
   IF ( MNSUB .GT. LNSUB )   GO TO 400
   IF ( IND .GE. 2 )   GO TO 181
   IF(JNEX(1).EQ.0) CALL ERROR
   CALL FNWEF(NTAP5)
   CALL FNWEF(NTAP5)

```

RESTART
RESTART

RESTART

RESTART
RESTART
RESTART

RESTART
RESTART

CALL ZEROT(A,LASI,1,1)

READ IN JOINT NUMBERS

65 CALL FNRTB(NTAP6,2,NJ,1,JN,LF6,IERCOD)
CALL TPTES2(NTAP6,IERCOD)
70 CALL FNPT(2,NTAP6)
MS=6*NJ

GET EXT. JTS. IN SUBSTRC INTO A VECTR

CALL IZEROT(JNES,LF4,1,1)
NEJS=0
DO 110 L=1,NJ
IF(JN(L))117,115,110
115 CALL ERROR
GO TO 2000
117 NEJS=NEJS+1
IF(NEJS.GT.LF4)CALL ERROR
JNES(NEJS)=-JN(L)
110 CONTINUE

130 CALL FNWTB(NTAP11,2,NEJS,1,JNES,LF4,IERCOD)
CALL TPTES2(NTAP11,IERCOD)

READ IN MATRIX

CALL ASEM23(A,IA,LASI,MS,NTAP12,NTAP7,NTAP8,NTAP10,
1 NOP,NOPCHG,NORM,NORMSB,LNSUB,NJ,JN)
CALL FILPAS(-2,NTAP8)

SAVE EXTERNAL JOINT COLUMNS

CALL SORTEX(A,IA,LASI,EXJC,MS,DEF,DEF,NJ,JN,NOP,
1 NEJS,JNES,NTAP12,NTAP7,NTAP10)
CALL FNWEF(NTAP10)
CALL FNPT(-1,NTAP10)
CALL FILPAS(-1,NTAP10)
CALL FNPT(1,NTAP10)

DECOMPOSE THE MATRIX

170 CALL HYSIMQ(A,LASI,MS,DET,IDET,NTAP8,NTAP5,NOITER,0,IER)
IF(DET)180,160,180
160 CALL ERROR

NOW SOLVE FOR ALL RIGHT HAND SIDES

DO FOR CONSTANT VECTER FIRST

BACK SUBSTITUTE FOR SOLUTION

180 IER=1
CALL HYSIMQ(A,LASI,MS,DET,IDET,NTAP8,NTAP10,NOITER,-1,IER)

SAVE ONLY--SOLUTION AT EXT. JTS.

NS=1
CALL SAEXC(A,A(7201),A(8401),NEJS,NJ,JN,NTAP11,NTAP10,NS)
IF (RESTR .NE. 2.0) GO TO 182
IND = 2
CALL WRAP23(A)
181 IF (IND .GE. 3) GO TO 240
182 CALL FILPAS(-1,NTAP10)

RESTART
RESTART
RESTART
RESTART

```

CALL FNPT(2,NTAP10)
L2=0
C      USE HYSIMQ TRUNCATED VALUE OF STORAGE SIZE
NA=(LASI/76)*76
C      CALCULATE NO OF RHS TO BE DONE
NEJS6=NEJS*6
C      CALCULATE NO OF RHS WE CAN GET
C      CALCULATE NO OF RHS SAEXC WILL ALLOW
NRHS=(NA-1200)/(MS+NEJS6)
C      CHECK IF DECOMPOSED MACRO COLUMN IS MORE
IF(1200+NRHS*NEJS6.LT.38*NJ) NRHS=(NA-38*NJ)/MS
C      CALCULATE NO OF RHS LEFT
NNN=NEJS6
C      HOW MANY MORE CAN WE DO
200 IRHS=-NRHS
IF(NNN.LT.NRHS) IRHS=-NNN
IER=2
CALL HYSIMQ(A,LASI,MS,DET,IDET,NTAP8,NTAP10,NOITER,IRHS,IER)
L2=L2+IABS(IRHS)
NS=IABS(IRHS)
MSNS1=MS*NS+1
CALL SAEXC(A,A(MSNS1),A(MSNS1+1200),NEJS,NJ,JN,NTAP11,NTAP10,NS)
IF ( RESTRT .NE. 2.0 ) GO TO 230
IND = 3
CALL WRAP23(A)
C      TEST IF ANY ARE LEFT
230 NNN=NEJS6-L2
IF(NNN.EQ.0) GO TO 250
MNNN=-NNN
CALL FNPT(MNNN,NTAP10)
GO TO 200
C
240 IF ( IND .EQ. 4 ) GO TO 350
NNN = NEJS6 - L2
IF ( NNN .NE. 0 ) GO TO 200
250 CONTINUE
300 CALL FNWEF(NTAP11)
CALL FILPAS(1,NTAP8)
CALL FNPT(1,NTAP10)
IF ( RESTRT .NE. 2.0 ) GO TO 400
IND = 4
CALL WRAP23(A)
350 IND = 1
400 CONTINUE
IF ( RESTRT .NE. 2.0 ) GO TO 500
IND = 5
CALL WRAP23(A)
C
500 CALL FNWEF(NTAP11)
CALL FNREW(NTAP11)
C*****END PHAS1
C
C
C*****PHAS2 ASSEMBLE AND MODIFY DUE
C      TO BOUNDARY CONDITIONS, THE EXTERNAL JOINT
C      SYSTEM MATRIX.
C
C

```

RESTART
RESTART
RESTART

RESTART
RESTART
RESTART

RESTART
RESTART
RESTART
RESTART

RESTART
RESTART
RESTART


```

C      READ IN JOINT NUMBERS OF EXTERNAL JOINTS
CALL FNPT(2,NTAP6)
C
C      ASSEMBLE MATRIX OF EXT. JT. FROM TAPE
LFSI61=LFSI6+1
CALL ASMX23(A,A(LFSI61),A(LFSI61))
GO TO 514
ENTRY MN23RT
NSUB=SUBCAS
CALL FILPAS(-1,NTAP5)
CALL FNPT(1,NTAP5)
CALL FNRTB(NTAP5,3,DUMC23,473,F,1200,EXJC,1205,IER)
CALL TPTEST(NTAP5,IER)
CALL FNPT(1,NTAP5)
CALL FNREW(NTAP6)
CALL FILPAS(1,NTAP6)
CALL FNPT(2,NTAP6)
LFSI6=LFSI*6
LFSI61=LFSI6+1
CALL ASMXRT(A,A(LFSI61),A(LFSI61))
514 CONTINUE
C*****END PHAS2
C
C
C
C      THE FOLLOWING IS AN OPTION FOR STOPPING THE RUN AFTER
C      ASSEMBLING THE STIFFNESS MATRIX FOR EXTERNAL JOINTS
C      USE      1      8      1
C              6
C              $      EXECUTE DUMP,ON6
C
C      CALL SSWTCH(6,JOBSTP)
C      IF(JOBSTP.EQ.1) CALL STIFIN(NSUB,NEX,A,A(7601),IA(7601))
C      GO TO (525,515,525,515,525),NOP
C
C      GET DETERMINANT VALUE ONLY.
515 CALL HYSIMQ(A,LASI,MSE,DET,IDET,NTAP8,NTAP5,NOITER,2,IER)
IF(DET)519,516,519
516 CALL ERROR
519 IDC=IDC+1
DETV C(IDC)=DET
IDETV C(IDC)=IDET
IF(NOPCHG)520,905,520
520 CALL FILPAS(-2,NTAP7)
CALL FNPT(1,NTAP7)
GO TO 905
C
C      SOLVE SYSTEM OF EXT. JT.
525 CALL FNREW(NTAP6)
CALL FILPAS(-2,NTAP8)
CALL HYSIMQ(A,LASI,MSE,DET,IDET,NTAP8,NTAP5,NOITER,1,IER)
IF(DET)555,550,555
550 CALL ERROR
C

```

```

*      R
C
C*****PHAS3      SAVE EXT. JT. SYSTEM AND APPLY
C      THESE DEFLECTIONS TO THE SUBSTRUCTURES.
C
C
C      SAVE SOLUTION FOR EXTERNAL JOINTS
555 IF(NOP-5)558,556,558
556 IDCR=IDCR+1
    DET=DETVC(IDCR)
    IDET=IDETVC(IDCR)
558 CALL FNWTB(NTAP5,3,RPM,1,DET,1,IDET,1,IERCOD)
    CALL TPTES2(NTAP5,IERCOD)
    CALL FNWTB(NTAP5,1,A,MSE,IERCOD)
    CALL TPTES2(NTAP5,IERCOD)
C
C      PUT EXT. JT. SOLUTION IN DEF ARRAY TEMPORARILLY
C      SUPERPOSE KNOWN DEFLECTIONS
    DO 560 L1=1,MSE
    IF(FIXDI(EXJC(L1),BITS))561,562,561
C
C      THE KNOWN EXT. JT. DEFLECTIONS ARE IN
C      EXJC, WHERE THEY WERE PUT BY ASMX23.
561 DEF(L1)=EXJC(L1)
    GO TO 560
562 DEF(L1)=A(L1)
560 CONTINUE
C
    MNSUB=-NSUB-1
    CALL FILPAS(MNSUB,NTAP10)
    CALL FNPT(1,NTAP10)
    NSUB2=-2*NSUB 1
    CALL FILPAS(NSUB2,NTAP8)
    CALL FNPT(+1,NTAP8)
C
    DO 900 I=1,NSUB
570 CALL FNRTB(NTAP6,2,NJ,1,JN,LF6,IERCOD)
580 CALL TPTES2(NTAP6,IERCOD)
C
590 CALL FNPT(2,NTAP6)
595 MS=NJ*6
    CALL FNRTB(NTAP10,1,F,MS,IERCOD)
600 CALL TPTES2(NTAP10,IERCOD)
C
C      READ IN EXT. JT. COLUMNS,MULTIPLY BY
C      DEFLECTIONS AT EXT. JT. AND SUBTRACT
C      FROM RIGHT HAND SIDE
C      LDEF=LOCATION IN DEF DIVIDED BY 6.
C      LJN=LOCATION IN JN
C      LDON=PORTION OF JN ALREADY SEARCHED
    LDON=1
690 DO 700 L=LDON,NJ
    IF(JN(L))720,710,700
710 CALL ERROR
700 CONTINUE
715 CALL FNWTB(NTAP5,1,F,MS,IERCOD)
    CALL TPTES2(NTAP5,IERCOD)
    CALL FNPT(-1,NTAP5)

```

GO TO 800

LJN=L

LDON=L+1

CALL SETJT2(JNEX,LF6,JN(LJN),LDEF)

LDEF=(LDEF-1)*6

DO 795 L2=1,6

CALL FNRTB(NTAP10,1,EXJC,MS,IERCOD)

CALL TPTES2(NTAP10,IERCOD)

LDEF=LDEF+1

DO 790 L=1,MS

790 F(L)=F(L)-EXJC(L)*DEF(LDEF)

795 CONTINUE

PUT ON TAPE AT 715

IF(LDON-NJ)690,690,715

800 CALL FNPT(1,NTAP10)

BACK SUBSTITUTE

CALL HYSIMW(A,LASI,MS,DET,IDET,NTAP8,NTAP5,NOITER,-1,IER)

820 CALL FILPAS(2,NTAP8)

CALL FNPT(-1,NTAP5)

SAVE SOLUTION FOR SUBSTRUCTURE

CALL FNWTB(NTAP5,1,A,MS,IERCOD)

CALL TPTES2(NTAP5,IERCOD)

900 CONTINUE

MNSUB2=-2*NSUB

CALL FILPAS(MNSUB2,NTAP8)

CALL FNPT(-1,NTAP8)

C*****END PHAS3

GO TO (902,905,1000,905,902),NOP

902 CALL FNWEF(NTAP5)

TEST FOR LAST RPM

905 CALL FNRTB(NTAP7,3,RPM,1,RPM,1,RPM,1,IERCOD)

IF(IERCOD.EQ.8)GO TO 1000

CALL TPTEST(NTAP7,IERCOD)

920 CALL FNREW(NTAP11)

CALL FNREW(NTAP6)

IF(NOP-5)960,950,960

950 NOP=4

NOPCHG=1

GO TO 100

960 IF(NOPCHG)970,100,970

970 NOP=5

NOPCHG=0

GO TO 100

```

*      R
C
C
*****PHASOT - EXIT PROCEDURES
C
1000 GO TO (1001,1002,1003,1004,1005),NOP
C
1001 CALL FNWEF(NTAP5)
      CALL FNREW(NTAP5)
      CALL FILPAS(2,NTAP5)
      CALL FNPT(1,NTAP6)
      CALL FNREW(NTAP8)
      CALL XLINK(1050,0)
C
1003 CALL FNWEF(NTAP5)
      CALL FNWEF(NTAP5)
      CALL FILPAS(-3,NTAP5)
      CALL FNPT(1,NTAP5)
      CALL FNPT(1,NTAP6)
      CALL FNPT(-1,NTAP8)
      IF (IBCKLD.GE.1) CALL XLINK(1061,0)
      CALL XLINK(1050,0)
C
1002 CONTINUE
1004 CALL FNWTB(NTAP5,3,DETV, IDC, IDETV, IDC, IDC, 1, IERCOD)
      CALL TPTES2(NTAP5,IERCOD)
      CALL FNWEF(NTAP5)
      CALL FNPT(-3,NTAP5)
      CALL FNPT(-1,NTAP8)
      MNSUB1=-NSUB-1
      CALL FILPAS(MNSUB1,NTAP10)
      CALL XLINK(1021,0)
C
C
1005 CALL FNWEF(NTAP5)
      CALL FNREW(NTAP5)
      CALL FILPAS(2,NTAP5)
      CALL FNPT(1,NTAP6)
      CALL FNPT(-1,NTAP8)
      CALL XLINK(1021,0)
C
2000 CALL XLINK(1099,0)
C
      RETURN
      END

```

APPENDIX D

NOTES ON CHANGES MADE TO PAPA TO IMPLEMENT RESTART

The following partial listings of routines have statements
blocked in which were changed or are new:

1. INIT31
2. BKDC00
3. MAIN33

	Y(I)=0.0	INIT081	
	Z(I)=0.0		
	140 NJJ(I)=0		
C			INIT084C
	DO 150 I=1,6		INIT085C
	RESI(I)=0.0		INIT086C
	RESJ(I)=0.0		INIT087C
	RESK(I)=0.0		
	RESL(I)=0.0		
	ITT(I)=0		
	150 IT(I)=0		INIT088C
C	READ 1ST RECORD		INIT089C
	160 CALL FNRTB(NTAP6,10,TOL,1,RESJ,1,DRAW,1,XY,1,XZ,1,YZ,1,		
	*DIMEN,1,AID,1,RESTR,1,PROGM,1,KCODE)		
	CALL TPTEST(NTAP6 ,KCODE)		
	RSJSAV=RESJ		
	165 IF(10F)190,170,190		INIT0910
C	TEST FOR AID RESTART		INIT0920
	170 DO 172 I=1,301		INIT0930
	172 DA(I)=BITS		INIT0940
	<u>IF(RESTR.NE.1.0) GO TO 180</u>		
	175 CALL FNWEF(NTAP10)		
	176 CALL XLINK(1061,4)		
	179 CALL ERRORA		INIT0990
	CALL EXIT		INIT1000
C	TEST FOR AID		INIT1010
	180 IF(AID)185,190,185		INIT1020
	185 CALL FNREW(NTAP8)		
	NTAP=NTAP11		
	CALL FNWTB(NTAP8,3,1039.0,1,1039.0,1,1039.0,1,KCODE)		
	CALL TPTEST(NTAP8,KCODE)		
	GO TO 200		
C	NO AID		INIT1090
	190 CALL BSRW		INIT1100
	NTAP=NTAP8		
C			INIT1150
	200 IF(DRAW.EQ.0.)GO TO 205		
	CALL FNREW(NTAP12)		
	CALL SBZ(XY,1)		
	CALL SBZ(XZ,1)		
	CALL SBZ(YZ,1)		
	205 CONTINUE		
C			INIT1180
	300 IF(RESJ)310,1000,310		
	310 CALL USCRBL(RESJ,RESI)		INIT1200
	1000 RETURN		
	END		INIT1220

5 INCODE IBMF
 CRKDC00 BKDC00
 BLOCK DATA
 C5 C00
 CC00 C00

COMMON/C00/

1 PROG, AID, IOF, NTAP2, NTAP3, NTAP4, NTAP5, NTAP6,
 2 NTAP7, NTAP8, NTAP10, NTAP11, NTAP12, NTAP13, BITS, IBITS,
 3 LK00, LK0999, LK31, LK32, LK30, LK20, LK21, LK22,
 4 LK23, LK50, TOL, MS, OPERA, NOITER, STATI, SUBCAS,
 5 NAID, CPS, NU3340, INER, LK09, LK99, LK27,
 6 LK24, LK25, LK26, LK3132, LK2021, DISC1, LK2223, LK28,
 7 LK33, LK2425, LK29, LK2930, LK18, LK19, LK2728,
 8 LK34, LK1818, LKRS,
 9 DUMY(4), RESTRT, LKRS23, ISUBRS, LK59,
 A IVIBR,YKK, LK6162, IBCKLD, LK70, PLTB(30), INDI, NTAP15
 DIMENSION DUMC00(105)
 EQUIVALENCE (DUMC00(1),PROG)

C PLTB(1) THRU PLTB(30) AND DUMC00(103) THRU DUMC00(105) UNUSED.

CC00 C00

C\$ END

C FOR LINK ZERO

DATA NTAP2/5/, NTAP3/6/, NTAP4/1/, NTAP5/4/,
 1 NTAP6/2/, NTAP7/8/, NTAP8/3/, NTAP10/7/,
 2 NTAP11/9/, NTAP12/11/, NTAP13/10/

C

DATA LK00/4HLK00/, LK09/4HLK09/, LK31/4HLK31/,
 1 LK32/4HLK32/, LK30/4HLK30/, LK20/4HLK20/,
 2 LK21/4HLK21/, LK22/4HLK22/, LK23/4HLK23/,
 3 LK50/4HLK50/, LK99/4HLK99/, LK0999/6HLK0999/,
 4 LK27/4HLK27/, LK24/4HLK24/, LK25/4HLK25/,
 5 LK26/4HLK26/, LK2021/6HLK2021/, LK3132/6HLK3132/,
 6 DISC1/2HD1/, LK2223/6HLK2223/, LK28/4HLK28/,
 7 LK29/4HLK29/, LK2930/6HLK2930/, LK18/4HLK18/, LK19/4HLK19/,
 8 LK2728/6HLK2728/, LK33/4HLK33/, LK2425/6HLK2425/,
 9 LK34/4HLK34/, LK1818/6HLK1818/, LK59/4HLK59/

DATA LKRS / 4HLKRS / , LKRS23 / 6HLKRS23 /

C

DATA BITS/0377777777777777/, IBITS/0377777777777777/

C

DATA NU3340/5HU3340/
 DATA LK6162/6HLK6162/
 DATA LK70/4HLK70/

C

END

INCOD E IBMF
 MAIN33 MAIN33
 SUBROUTINE MAIN33

```

C
C$      C00
CC00      C00
COMMON/C00/
1  PROGM, AID, IOF, NTAP2, NTAP3, NTAP4, NTAP5, NTAP6,
2  NTAP7, NTAP8, NTAP10, NTAP11, NTAP12, NTAP13, BITS, IBITS,
3  LK00, LK0999, LK31, LK32, LK30, LK20, LK21, LK22,
4  LK23, LK50, TOL, MS, OPERA, NOITER, STAT1, SUBCAS,
5  NAID, CPS, NU3340, INER, LK09, LK99, LK27,
6  LK24, LK25, LK26, LK3132, LK2021, DISC1, LK2223, LK28,
7  LK33, LK2425, LK29, LK2930, LK18, LK19, LK2728,
8  LK34, LK1818, LKRS,
9  DUMY(4), RESTRT, LKRS23, ISUBRS, LK59,
A  IVIBR,YKK, LK6162, IBCKLD, LK70, PLTB(30), INDI, NTAP15
  DIMENSION DUMC00(105)
  EQUIVALENCE (DUMC00(1),PROGM)
C  PLTB(1) THRU PLTB(30) AND DUMC00(103) THRU DUMC00(105) UNUSED.
CC00      C00
C$      END
C$      C33
C$      C33
CC33      C33
COMMON/C33/
1  IVIB, NOSTRT,NJ, DRAW, DIMEN, XY,
2  XZ, YZ, DA, PVEC, RVEC, JN,
3  NSUB, JT, JEL, M, MEL, NM
COMMON/C33/
1  JEMAX, NRJMAX,NJMAX, NMMAX, NSUBMX,IRROR,
2  I3D, JD1, JD2, NM3D
  DIMENSION
1  JT(2000), JEL(2000), M(4,3000),
2  MEL(3000), DA(301), PVEC(9,10),
3  RVEC(4,200)
  DIMENSION M3D(6,2000)
C
  EQUIVALENCE (M3D(1,1),M(1,1))
CC33      C33
C$      END
  DIMENSION ESPEC(2)
C
  NAMELIST/W/TOL,RESJ,PROGM,DA,ESPEC,DRAW,
1  DIMEN,XY,XZ,YZ,AID,RESTRT,I3D,YKK,IBCKLD,INDI,ISUBRS
C
  NAMELIST/INFO/JT,JEL,M,MEL
C
  DATA BLANK/6H /
C
  INITIALIZATION OF VARIABLES
  CALL SBITS(RVEC,800)
  CALL ZEROT(PVEC,9,10,1)
  CALL IZEROT(JEL,2000,1,1)
  CALL IZEROT(MEL,3000,1,1)
  CALL IZEROT(M,4,3000,1)
  CALL IZEROT(JT,2000,1,1)
  IBCKLD=0
  IOUT=0

```