R.T. Barrett

POST-WRINKLING ANALYSIS OF HIGHLY PRESSURIZED CYLINDRICAL AND CONICAL SHELLS OF REVOLUTION SUBJECTED TO BENDING LOADS

VOLUME 2
COMPUTER PROGRAM LISTING

15 October 1967

07823-6011-R000

JUL 1968
RECEIVED
NASA STI FACILITY
INPUT BRANCH

Prepared for:

NASA Lewis Research Center
Cleveland, Ohio

JAN 23 1968
NASA
LANGLEY RESEARCH CENTER
LIBRARY

N69-79945
(ACCESSION NUMBER)
354
(PAGES)
CR-106362
(NASA CR OR TMX OR AD NUMBER)
(THRU)
NONE
(CODE)
(CATEGORY)
FACILITY FORM 602

TRW
SYSTEMS GROUP

R752401C

POST-WRINKLING ANALYSIS OF HIGHLY PRESSURIZED CYLINDRICAL AND CONICAL SHELLS OF REVOLUTION SUBJECTED TO BENDING LOADS

VOLUME 2
COMPUTER PROGRAM LISTING

15 October 1967 07823-6011-R000

Prepared for:

NASA Lewis Research Center
Cleveland, Ohio

By:

Gilbert A. Greenbaum
and
Dennis C. Conroy

Approved by:

P. G. Bhuta, Manager
Advanced Technology Department

M. V. Barton, Manager
Engineering Mechanics Laboratory

D. W. Routh, Manager
Subsystems Programming Department

TRW Systems
One Space Park
Redondo Beach, California

```
C     MONITOR PWA                                                       MONI0010
      COMMON/COMAA/ICHECK                                               MONI0020
      COMMON/COMBB/CCBETA,IIMITE                                        MONI0030
      COMMON/COMCC/CONFAC                                               MONI0035
      COMMON NSGMTS, KS, ISGMT, IFTNO, LOADNO, LDNCPR, NLOAD, KKKKK     MONI0040
      COMMON PHI(7),SINP(7),COSP(7),NPSEG(7),LSP(7),MLIST(7),ECR(7),    MONI0050
     1 ECY(7),IDPT(14),IRES(10),NREC(21),INTF(10),NSGPL(10),NPTPT(10),  MONI0060
     2 NLDPL(10)                                                        MONI0070
      COMMON EI(2,7),HI(2,7),FMUI(2,7),RI(7)                            MONI0080
      COMMON FLDI(5,7,7),FLDF(5,7,7),DLDI(5,7,7),DLDF(5,7,7)            MONI0090
      COMMON FMPO(2,7,7),FNRO(2,7,7),FNYO(2,7,7),FNPTO(2,7,7),ECRS(7,7),MONI0100
     1 ECYS(7,7),RSS(7,7),SCF(16,7,7)                                   MONI0110
      COMMON IBC(4,7,2),BC(4,7,2),DLBC(4,7,2)                           MONI0120
      COMMON S(300),FNTH(300),FNPH(300),OPHI(300),OTHE(300),STP(12,7)   MONI0130
      COMMON AK(140,23),AU(140,23),AL(200,13),RHV(600),SAVE(5,10)       MONI0140
      ICHECK = 0                                                        MONI0150
      KKKKK=1                                                           MONI0160
  200 CONTINUE                                                          MONI0170
      GO TO (10,20,30,40,50,60,70,80), KKKKK                            MONI0180
C     READ INPUT GEOMETRY                                               MONI0190
   10 CALL MON1                                                         MONI0200    6
      GO TO 200                                                         MONI0210
C     CALCULATE GEOMETRY, FINITE DIFFERENCE COEFFICIENTS, AND D AND K   MONI0220
   20 CALL MON2                                                         MONI0230    9
      GO TO 200                                                         MONI0240
C     GENERATE 'K' MATRIX                                               MONI0250
   30 CALL MON3                                                         MONI0260   12
      GO TO 200                                                         MONI0270
C     SOLVE 'K' MATRIX                                                  MONI0280
   40 CALL MON4                                                         MONI0290   15
      GO TO 200                                                         MONI0300
   50 CONTINUE                                                          MONI0310
      CALL MON5                                                         MONI0320   18
      GO TO 200                                                         MONI0330
   60 CALL MON6                                                         MONI0340   21
      GO TO 200                                                         MONI0350
   70 CALL MON7                                                         MONI0360   24
```

```
      GO TO 200                                                MONI0370
   80 GO TO 300                                                MONI0380
  300 STOP                                                     MONI0390
      END                                                      MONI0400
```

```
$IBFTC FDICX   NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD FINITE DIFFERENCE COEFFICIENTS                  FDIC0010
      SUBROUTINE FDIC(FDC,S,MS,MF)                                      FDIC0020
      DIMENSION FDC(3, 1),S( 1)                                         FDIC0030
      ONE = 1.0                                                         FDIC0040
      KK  = 1                                                           FDIC0050
C     BUILD FINITE DIFFERENCE COEFFICIENTS FOR POINT 1                  FDIC0060
      MS1 = MS+1                                                        FDIC0070
      MS2 = MS+2                                                        FDIC0080
      D1  = S(MS2)-S(MS)                                                FDIC0090
      D2  = S(MS1)-S(MS)                                                FDIC0100
      D3  = S(MS2)-S(MS1)                                               FDIC0110
      FDC(1,KK) = -(D1+D2)/(D2*D1)                                      FDIC0120
      FDC(2,KK) =  D1/(D2*D3)                                           FDIC0130
      FDC(3,KK) = -D2/(D3*D1)                                           FDIC0140
C     BUILD COEFFICIENTS FOR POINTS 2 TO (N-1)                          FDIC0150
      MF1 = MF-1                                                        FDIC0160
      DO 30 I=MS1,MF1                                                   FDIC0170
      KK = KK+1                                                         FDIC0180
      D1 = S(I+1)-S(I)                                                  FDIC0190
      D2 = S(I)-S(I-1)                                                  FDIC0200
      D3 = S(I+1)-S(I-1)                                                FDIC0210
      D5 = ONE/(D2*D1)                                                  FDIC0220
      D6 = ONE/(D3*D1)                                                  FDIC0230
      FDC(1,KK) = -D1/(D2*D3)                                           FDIC0240
      FDC(2,KK) =  (D1-D2)*D5                                           FDIC0250
      FDC(3,KK) =  D2*D6                                                FDIC0260
   30 CONTINUE                                                          FDIC0270
C     BUILD COEFFICIENTS FOR LAST POINT IN SEGMENT                      FDIC0280
      MF2 = MF-2                                                        FDIC0290
      D1  = S(MF)-S(MF1)                                                FDIC0300
      D2  = S(MF)-S(MF2)                                                FDIC0310
      D3  = S(MF1)-S(MF2)                                               FDIC0320
      KK  = KK+1                                                        FDIC0330
      FDC(1,KK) =  D1/(D2*D3)                                           FDIC0340
      FDC(2,KK) = -D2/(D3*D1)                                           FDIC0350
      FDC(3,KK) =  (D1+D2)/(D2*D1)                                      FDIC0360
      RETURN                                                            FDIC0370
```

```
END                                                    FDIC0380
```

```
$IBFTC GETMX    NOLIST,DECK,M94,X27,NODD
```

```
C     FOR ALL HARMONICS OF A GIVEN LIST, MLIST                          GETM0010
C     GETM DETRMINES THE INDICES OF THOSE HARMONICS SUCH THAT           GETM0020
C          1.  IN A SUMMATION OVER HARMONICS FROM 1 TO INFINITY         GETM0030
C              THE PRODUCT  X(L) * X(M+L) IS NOT TRIVIALLY EQUAL TO     GETM0040
C              ZERO.                                                    GETM0050
C                                                                       GETM0060
C          2.  IN A SUMMATION OVER HARMONICS FROM 1 TO M-1  THE         GETM0070
C              PRODUCT X(L) * X(M-L) IS NOT TRIVIALLY EQUAL TO ZERO.    GETM0080
C                                                                       GETM0090
C          3.  IN A SUMMATION OVER HARMONICS FROM ONE TO INFINITY       GETM0100
C              THE PRODUCT X(L) * X(L) IS NOT TRIVIALLY EQUAL TO ZERO.  GETM0110
C                                                                       GETM0120
      SUBROUTINE GETM(MLIST,KST,MIND,ISFLAG,MAXH)                       GETM0130
      DIMENSION MLIST(1),MIND(MAXH,5,1),ISFLAG(3,1)                     GETM0140
      KS = KST                                                          GETM0150
      DO 250 IMM = 1,KS                                                 GETM0160
      M = MLIST(IMM)                                                    GETM0170
C     INITIALIZE FLAGS, BUFFERS, AND COUNTERS                           GETM0180
      ILGZ = 0                                                          GETM0190
      IMPL = 0                                                          GETM0200
      IMML = 0                                                          GETM0210
      MGM  = 0                                                          GETM0220
      DO 20 J=1,5                                                       GETM0230
      IF(J.GT.3) GO TO 10                                               GETM0240
      ISFLAG(J,IMM) = 0                                                 GETM0250
   10 CONTINUE                                                          GETM0260
      DO 20 K=1,KS                                                      GETM0270
      MIND(K,J,IMM) = 0                                                 GETM0280
   20 CONTINUE                                                          GETM0290
C     GET THOSE HARMONICS WITH VALUE .GT. 0                             GETM0300
      NMP1 = M                                                          GETM0310
      DO 40 I =1,KS                                                     GETM0320
      IF(MLIST(I).LT.NMP1) MGM=MGM+1                                    GETM0330
      IF(MLIST(I).LT.1) GO TO 40                                        GETM0340
      ILGZ = ILGZ+1                                                     GETM0350
      MIND(ILGZ,5,IMM) = I                                              GETM0360
   40 CONTINUE                                                          GETM0370
```

```
      ISFLAG(1,IMM) = ILGZ                                              GETM0380
      IF(ILGZ) 45,250,45                                                GETM0390
   45 CONTINUE                                                          GETM0400
C     GET THE MAXIMUM VALUE OF THE HARMONICS                            GETM0410
      MAX = -100                                                        GETM0420
      DO 60 I=1,KS                                                      GETM0430
      MAX = MAX0(MAX,MLIST(I))                                          GETM0440
   60 CONTINUE                                                          GETM0450
C     SET UPPER INDEX OF LOOP                                           GETM0460
      MAXM = MAX+IABS(M)                                                GETM0470
C     START LOOP OVER L TO DETERMINE THOSE INDICES SUCH THAT FOR THE    GETM0480
C          CURRENT HARMONIC, 'M+L' AND 'M-L' EXIST                      GETM0490
      DO 200 L=1,MAXM                                                   GETM0500
C     SET VALUES OF 'M+L' AND 'M-L'                                     GETM0510
      MPL1 = M+L                                                        GETM0520
      MML1 = M-L                                                        GETM0530
C     INITIALIZE TEMPORARY STORAGE                                      GETM0540
      MPLT = 0                                                          GETM0550
      JPLT = 0                                                          GETM0560
      MMLT = 0                                                          GETM0570
C     START LOOP TO TEST EACH OF THE HARMONICS                          GETM0580
      DO 100 J=1,ILGZ                                                   GETM0590
      I    = MIND(J,5,IMM)                                              GETM0600
C     GET VALUE OF 'I'TH HARMONIC                                       GETM0610
      MT   = MLIST(I)                                                   GETM0620
      IF(MT.NE.L) GO TO 70                                              GETM0630
C     IF 'I'TH HARMONIC EQUALS L, SAVE INDEX                            GETM0640
      JPLT = I                                                          GETM0650
   70 CONTINUE                                                          GETM0660
      IF(MT.NE.MPL1) GO TO 80                                           GETM0670
C     IF 'I'TH HARMONIC EQUALS M+L, SAVE INDEX                          GETM0680
      MPLT = I                                                          GETM0690
   80 CONTINUE                                                          GETM0700
      IF(MT.NE.MML1) GO TO 100                                          GETM0710
C     IF 'I'TH HARMONIC EQUALS M-L, SAVE INDEX                          GETM0720
      MMLT = I                                                          GETM0730
  100 CONTINUE                                                          GETM0740
  110 CONTINUE                                                          GETM0750
```

```
      IF((JPLT.EQ.0).OR.(MPLT.EQ.0)) GO TO 140                          GETM0760
C     SAVE INDICES OF HARMONICS SUCH THAT 'L' AND 'M+L' EXIST           GETM0770
C     INCREMENT COUNTER FOR EXISTENCE OF 'L' AND 'M+L'                  GETM0780
      IMPL = IMPL+1                                                     GETM0790
      MIND(IMPL,2,IMM) = JPLT                                           GETM0800
      MIND(IMPL,1,IMM) = MPLT                                           GETM0810
  140 CONTINUE                                                          GETM0820
      IF((JPLT.EQ.0).OR.(MMLT.EQ.0)) GO TO 200                          GETM0830
C     SAVE INDICES OF HARMONICS SUCH THAT 'L' AND 'M-L' EXIST           GETM0840
      IF(MGM.EQ.0) GO TO 200                                            GETM0850
C     INCREMENT COUNTER FOR EXISTENCE OF 'L' AND 'M-L'                  GETM0860
      IMML = IMML+1                                                     GETM0870
      MIND(IMML,4,IMM) = JPLT                                           GETM0880
      MIND(IMML,3,IMM) = MMLT                                           GETM0890
      MGM  = MGM-1                                                      GETM0900
  200 CONTINUE                                                          GETM0910
      ISFLAG(2,IMM) = IMPL                                              GETM0920
      ISFLAG(3,IMM) = IMML                                              GETM0930
  250 CONTINUE                                                          GETM0940
  300 CONTINUE                                                          GETM0950
      RETURN                                                            GETM0960
      END                                                               GETM0970
```

MONIX STORAGE MAP

MAIN PROGRAM

COMMON VARIABLES

COMMON BLOCK			COMAA	ORIGIN	00001	LENGTH	00001
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ICHECK	00000	I					
COMMON BLOCK			COMBB	ORIGIN	00002	LENGTH	00002
CCBETA	00000	R	IIMITE	00001	I		
COMMON BLOCK			COMCC	ORIGIN	00004	LENGTH	00001
CONFAC	00000	R					
COMMON BLOCK			//	ORIGIN	00005	LENGTH	33167
NSGMTS	00000	I	KS	00001	I	ISGMT	00002
IFTNO	00003	I	LOADNO	00004	I	LDNOPR	00005
NLOAD	00006	I	KKKKK	00007	I	PHI	00010
SINP	00017	R	COSP	00026	R	NPSEG	00035
LSP	00044	I	MLIST	00053	I	ECR	00062
ECY	00071	R	IOPT	00100	I	IRES	00116
NREC	00130	I	INTF	00155	I	NSGPL	00167
NPTPT	00201	I	NLDPL	00213	I	EI	00225
HI	00243	R	FMUI	00261	R	RI	00277
FLDI	00306	R	FLDF	00673	R	DLDI	01260
DLDF	01645	R	FMPO	02232	R	FNRO	02374
FNYO	02536	R	FNPTO	02700	R	ECRS	03042
ECYS	03123	R	RSS	03204	R	SCF	03265
IBC	04705	I	BC	04775	R	DLBC	05065
S	05155	R	FNTH	05631	R	FNPH	06305
DPHI	06761	R	DTHE	07435	R	STP	10111
AK	10235	R	AU	16461	R	AL	24705
RHV	31755	R	SAVE	33105	R		

ENTRY POINTS

..... SECTION 10

SUBROUTINES CALLED

MON1	SECTION	11	MON2	SECTION	12	MON3	SECTION
MON4	SECTION	14	MON5	SECTION	15	MON6	SECTION
MON7	SECTION	17	.EXIT.	SECTION	18	.FXEM.	SECTION
SYSLOC	SECTION	20					

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
200	3A	33205	10	5A	33222	20	8A	
30	11A	33232	40	14A	33236	50	17A	
60	20A	33246	70	23A	33252	80	26A	
300	27A	33257						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 33304.

SUBROUTINE FDIC

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ONE	00001	R	KK	00002	I	MS1	00003
MS2	00004	I	D1	00005	R	D2	00006
D3	00007	R	MF1	00010	I	D5	00011
D6	00012	R	MF2	00013	I		

ENTRY POINTS

FDIC SECTION 3

SUBROUTINES CALLED

SYSLOC SECTION 4

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
30	28A	00177					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00355.

SUBROUTINE GETM

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
KS	00001	I	IMM	00002	I	M	00003
ILG7	00004	I	IMPL	00005	I	IMML	00006
MGM	00007	I	J	00010	I	NMP1	00011
T	00012	I	MAX	00013	I	MAXM	00014
L	00015	I	MPL1	00016	I	MML1	00017
MPLT	00020	I	JPLT	00021	I	MMLT	00022
MT	00023	I					

ENTRY POINTS

GETM SECTION 3

SUBROUTINES CALLED

SYSLOC SECTION 4

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
250	96A	00406	20	22A	00136	10	18A
40	37A	00177	45	41A	00205	60	46A
200	92A	00400	100	73A	00307	70	65A
80	69A	00301	110	75A	00311	140	82A
300	99A	00411					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00613.

```
$IBMAP PADR     LIST,REF,DECK,NOSYM,M94,RELMOD,NO( ),NOMFTC,NODD          PADR0000
```

7094 RELMCD ASSEMBLY.

$IBLDR PADR PADR0000

```
ASSEMBLED TEXT.

          $TEXT  PADR                                                    PADR0001

                                    ENTRY    .PADR

BINARY CARD ID. PADR0002
      00000  0 00000 0 00000  10000  .PADR  PZE
                       00000  01111         END
```

```
CONTROL DICTIONARY

          $CDICT PADR                                                    PADR0003

BINARY CARD ID. PADR0004
          000CC1000000          PREFACE    START=0,LENGTH=1,TYPE=7094,CMPLX=5
          000C04000005
          472124516060    PADR  DECK       LOC=0,LENGTH=1
          000001000000
          334721245160    .PADR REAL       LOC=0,LENGTH=0
          000C00000000

          $DKEND PADR                                                    PADR0005

NO MESSAGES FOR THIS ASSEMBLY
```

SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

CLASS	SYMBOL	VALUE	REFERENCES
	.PADR	00000	
LCTR	BLCTR		
DUAL	UNQS		
LCTR	//		

```
$IBMAP UN50     LIST,REF,DECK,NOSYM,M94,RELMOD,NO(),NOMFTC,NODD          UN500000
```

7094 RELMOD ASSEMBLY.

```
$IBLDR JN50                                                                    UN500000
$FILE  UN50    '.UNIT50',PCH,READY,INOUT,BLK=257,BIN,LOW,HOLD,NOLIST           UN500001
```

FILE DICTIONARY.

```
$FDICT UN50                                                    UN500002

BINARY CARD ID. UN500003
        206002000401  UNIT50 FILE   'UNIT50          '        BIN,INOUT,NOHCVN,BLK=257
        000000000000
        644531630500
        606060606060
        606060606060
```

```
ASSEMBLED TEXT.

          $TEXT  UN50                                                    UN500004

                                        ENTRY   .UN50.

BINARY CARD ID. UN500005
      00000  0 00000 0 04001  10010  .UN50. PZE     UNIT50
                                     UNIT50 FILE    ,PCH,READY,INOUT,BLK=257,BIN,LOW,HOLD,NOLIST
                       00000  01111         END
```

```
CONTROL DICTIONARY

          $CDICT UN50                                                    UN500006

BINARY CARD ID. UN500C07
          000001000000          PREFACE    START=0,LENGTH=1,TYPE=7094,CMPLX=5
          000004000005
          644505006060    UN50  DECK       LOC=0,LENGTH=1
          000C01000000
          336445050033    .UN50. REAL      LOC=0,LENGTH=0
          000000000000

          $DKEND UN50                                                    UN500008

NO MESSAGES FOR THIS ASSEMBLY
```

SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

CLASS	SYMBOL	VALUE	REFERENCES
	.UN50.	00000	
LCTR	BLCTR		
QUAL	UNQS		
LCTR	//		
FILE	UNIT50	1	0

```
$IBMAP GEOD     LIST,REF,DECK,NOSYM,M94,RELMOD,NO(),NOMFTC,NODD          GE000000
```

7094 RELMOD ASSEMBLY.

$IBLDR GEOO GEOOOOOO

ASSEMBLED TEXT.

```
                $TEXT  GEOO                                                    GEOOO001

                                                ENTRY   GEOBOT
                                                ENTRY   GEOEOT
                        00000          GEOBOT   NULL

BINARY CARD ID. GEOOO002
        00000  0 00000 0 00000  10000  GEOEOT   PZE
                         00000  01111          END
```

```
CONTROL DICTIONARY

          $CDICT GEOO                                                    GEOO0003

BINARY CARD ID. GEOO0004
          000001000000          PREFACE    START=0,LENGTH=1,TYPE=7094,CMPLX=5
          000004000005
          272546466060    GEOO  DECK       LOC=0,LENGTH=1
          000001000000
          272546224663    GEOBOT REAL      LOC=0,LENGTH=0
          000000000000
          272546254663    GEOEOT REAL      LOC=0,LENGTH=0
          000000000000

          $DKEND GEOO                                                    GEOO0005

NO MESSAGES FOR THIS ASSEMBLY
```

SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

CLASS	SYMBOL	VALUE	REFERENCES
	GEOBOT	00000	
	GEOEOT	00000	
LCTR	BLCTR		
QUAL	UNOS		
LCTR	//		

```
$IBFTC MON1X   NOLIST,DECK,M94,XR7,NODD
```

```
C                 MONITER FOR REGION 1--READ INPUT DATA                 MON10010
      SUBROUTINE MON1                                                   MON10020
      DIMENSION AA(12),SG( 8,7),MB(10,2)                                MON10030
      EQUIVALENCE(AK(1,1),SG(1,1)),(AK(60,1),MB(1,1))                   MON10040
      COMMON/COMAA/ICHECK                                               MON10050
      COMMON/COMBB/CCBETA,IIMITE                                        MON10060
      COMMON/COMCC/CONFAC                                               MON10065
      COMMON NSGMTS, KS, ISGMT, IFTNO, LOADNO, LDNOPR, NLOAD, KKKKK     MON10070
      COMMON PHI(7),SINP(7),COSP(7),NPSEG(7),LSP(7),MLIST(7),ECR(7),    MON10080
     1 ECY(7),IOPT(14),IRES(10),NREC(21),INTF(10),NSGPL(10),NPTPT(10),  MON10090
     2 NLDPL(10)                                                        MON10100
      COMMON EI(2,7),HI(2,7),FMJI(2,7),RI(7)                            MON10110
      COMMON FLDI(5,7,7),FLDF(5,7,7),DLDI(5,7,7),DLDF(5,7,7)            MON10120
      COMMON FMPO(2,7,7),FNRO(2,7,7),FNYO(2,7,7),FNPTO(2,7,7),ECRS(7,7),MON10130
     1 ECYS(7,7),RSS(7,7),SCF(16,7,7)                                   MON10140
      COMMON IBC(4,7,2),BC(4,7,2),DLBC(4,7,2)                           MON10150
      COMMON S(300),FNTH(300),FNPH(300),OPHI(300),OTHE(300),STP(12,7)   MON10160
      COMMON AK(140,23),AU(140,23),AL(200,13),RHV(600),SAVE(5,10)       MON10170
 1000 FORMAT(12A6)                                                      MON10180
 1001 FORMAT(7I10)                                                      MON10190
 1002 FORMAT(14I5)                                                      MON10200
 1003 FORMAT(1H1,12A6)                                                  MON10210
 1011 FORMAT(2I10,5F10.0)                                               MON10220
 1012 FORMAT(7F10.0)                                                    MON10230
 1013 FORMAT(I10,6F10.0)                                                MON10240
 1014 FORMAT(I10,4E15.8/ 10X,4E15.8/10X,4E15.8)                         MON10250
 1040 FORMAT(1H0,10X,39HNUMBER OF SEGMENTS EXCEEDS SIZE MAXIMUM)        MON10260
 1041 FORMAT(1H0,10X,43HNUMBER OF FOURIER HARMONICS EXCEEDS MAXIMUM)    MON10270
 1050 FORMAT(1H0,10X, 7HSEGMENT,I2,25H REQUESTS TOO MANY POINTS)        MON10280
 1051 FORMAT(1H0,10X,7HSEGMENT,I2,32H --TOO MANY LEADING AND TRAILING,  MON10290
     1 10H INTERVALS)                                                   MON10300
 1052 FORMAT(1H0,10X,45HNUMBER OF POINTS REQUESTED EXCEEDS CASE LIMIT)  MON10310
 1059 FORMAT(1H0,10X,35HGEOMETRY DATA NOT IN ORDER, SEGMENT,I3)         MON10320
 1060 FORMAT(1H0,10X,30HLOAD DATA NOT IN ORDER-SEGMENTI2,9H HARMONIC,I2)MON10330
 1061 FORMAT(1H0,10X,41HELASTIC SUPPORT DATA NOT IN ORDER-SEGMENT,I2,   MON10340
     1 9H HARMONIC,I2)                                                  MON10350
 1100 FORMAT(1H0, 9HCASE DATA/)                                         MON10360
```

```
1101 FORMAT(17H NO. OF SEGMENTS=,I2,5X,17HNO. OF HARMONICS=,I2,5X,      MON10370
    1 18HNO. OF LOAD STEPS=,I3,5X,19HNO. M=0 ITERATIONS=,I2)            MON10380
1102 FORMAT(16H HARMONIC VALUES/)                                       MON10390
1103 FORMAT(4H   M(,I2,4H) = ,I4)                                       MON10400
1104 FORMAT(1H0,19HBOUNDARY CONDITIONS/)                                MON10410
1105 FORMAT(1H0,3X,24HMAX. NO. OF ITERATIONS =,I3,5X,11HBETA(MAX) =,    MON10420
    1 F8.4,5X,20HCONVERGENCE FACTOR =,F8.4)                             MON10430
1110 FORMAT(1H0,14HOUTPUT OPTIONS/)                                     MON10440
1111 FORMAT(2X,4HOPT(,I2,4H) = ,I2)                                     MON10450
1112 FORMAT(1H0,15HRESTART OPTIONS/)                                    MON10460
1113 FORMAT(2X,5HROPT(I2,4H) = ,I2)                                     MON10470
1120 FORMAT(1H0,12HSEGMENT DATA/)                                       MON10480
1121 FORMAT(1H0,7HSEGMENT,I2//5X,16HNO. INTERVALS = ,I3,  3X,           MON10490
    118HNO. LEADING INT. =,F5.0,3X,19HNO. TRAILING INT. =,F5.0)         MON10500
1123 FORMAT(6X,20HTRAILING ( X, Y) = (,1PE13.6,1H, E13.6,1H)/)          MON10510
1122 FORMAT(6X,20HLEADING  ( X, Y) = (,1PE13.6,1H, E13.6,1H)/)          MON10520
1124 FORMAT(1H0,24X,1HE,17X,1HH,16X,2HMU/)                              MON10530
1125 FORMAT(5X,7HINITIAL,5(1PE19.7))                                    MON10540
1126 FORMAT(5X,5HFINAL,2X,5(1PE19.7))                                   MON10550
1127 FORMAT(1H0,3X,10HHARMONIC =,I3//5X,9HLOAD DATA)                    MON10560
1128 FORMAT(23X,4HXBAR,15X,4HYBAR,15X,4HZBAR,14X,5HAXIAL,14X,6HRADIAL/)MON10570
1129 FORMAT(10X,12HINCREMENT TO)                                        MON10580
1130 FORMAT(1H0,10X,20HELASTIC SUPPORT DATA/)                           MON10590
1131 FORMAT(1H ,10X,5(1PE19.6))                                         MON10600
1132 FORMAT(1H0,19X,5HMP(J),14X,5HNR(J),14X,5HNY(J),13X,6HNPT(J)/)      MON10610
1133 FORMAT(1H0,10X,16HSUPPORT RADIUS =,1PE11.4/11X,25HECCENTRICITY(C.GMON10620
    1. OFFSET)/15X,6HE(Y) =,E11.4/15X,6HE(R) =,E11.4)                   MON10630
1134 FORMAT(1H0,10X,27HSTIFFNESS COEFFICIENTS-C(I),/14X,1HI,11X,4HC(I))MON10640
1135 FORMAT(12X,I3,1PE19.6)                                             MON10650
1136 FORMAT(1H ,5X,20HSIZE LEADING INT.  =,1PE11.4 /6X,20HSIZE TRAILINGMON10660
    1 INT. =,1PE11.4)                                                   MON10670
1137 FORMAT(6X,6HE(Y) =,1PE13.6,5X,6HE(R) =,E13.6)                      MON10680
1140 FORMAT(1H0,5X,13HSCALE FACTORS//)                                  MON10690
1141 FORMAT(I10/10X,1P4E15.7/10X,4E15.7/10X,4E15.7)                     MON10700
1142 FORMAT(1H0,10X,50HINPUT HARMONIC VALUE NOT EQUAL HARMONIC LIST VALMON10710
    1UE)                                                                MON10720
1144 FORMAT(1H0,10X,43HCONVERGENCE CRITERION NOT GREATER THAN ZERO )    MON10730
1150 FORMAT(1H0,10X,21HEXECUTION TERMINATED,I3,13H INPUT ERRORS)        MON10740
```

```
C     ZERO LOAD NUMBER AND NEXT LOADING TO BE PRINTED FLAG          MON10750
      LOADNO = 0                                                    MON10760
      LDNOPR = 0                                                    MON10770
C     ZERO INTERNAL FLAGS                                           MON10780
      DO 1 I=1,10                                                   MON10790
    1 INTF(I) = 0                                                   MON10800
      KILL=0                                                        MON10810
      NOPT=14                                                       MON10820
      NRES= 7                                                       MON10830
C     SET CASE MAXIMUMS                                             MON10835
      MAXSEG=7                                                      MON10840
      MAXKS =7                                                      MON10850
      MAXINT=73                                                     MON10860
      MAXPTS=300                                                    MON10870
      NPTS=0                                                        MON10880
C     ZERO SPECIAL ITERATION FORCES                                 MON10882
      DO 12 I = 1,MAXPTS                                            MON10883
      FNTH(I) = 0.0                                                 MON10884
      FNPH(I) = 0.0                                                 MON10885
      OPHI(I) = 0.0                                                 MON10886
   12 OTHE(I) = 0.0                                                 MON10887
C     READ CASE COMMENTS                                            MON10890
      READ(5,1000)(AA(I),I=1,12)                                    MON10900   28
      WRITE(6,1003)(AA(I),I=1,12)                                   MON10910
C     READ FLAGS                                                    MON10920   35
      READ(5,1001) NSGMTS,KS,NLOAD,NZERO                            MON10930   42
      WRITE(6,1100)                                                 MON10940   47
      WRITE(6,1101) NSGMTS,KS,NLOAD,NZERO                           MON10950   48
      INTF(01) = NZERO                                              MON10960
      INTF(02) = NZERO                                              MON10970
      IF(NSGMTS.LE.MAXSEG) GO TO 3                                  MON10980
      WRITE(6,1040)                                                 MON10990   51
      KILL=KILL+1                                                   MON11000
    3 IF(KS.LE.MAXKS) GO TO 6                                       MON11010
      WRITE(6,1041)                                                 MON11020   54
      KILL=KILL+1                                                   MON11030
    6 IF(KILL.NE.0) GO TO 3030                                      MON11040
C     READ LIST OF HARMONICS                                        MON11050
```

```
      READ(5,1001)(MLIST(I),I=1,KS)                                     MON11060   57
      WRITE(6,1102)                                                     MON11070   64
      DO 2 IC=1,KS                                                      MON11080
    2 WRITE(6,1103) IC,MLIST(IC)                                        MON11090
C     READ INPUT OPTIONS                                                MON11100   68
      READ(5,1002) (IOPT(I),I=1,NOPT)                                   MON11110   71
      WRITE(6,1110)                                                     MON11120   78
      DO 16 IC=1,NOPT                                                   MON11130
   16 WRITE(6,1111) IC,IOPT(IC)                                         MON11140
C     IF ICHECK = 0, SKIP CHECKOUT PRINTING OPTIONS                     MON11150   82
      IF(ICHECK.GT.0) GO TO 18                                          MON11160
      CALL PDIN(IOPT,MB,NSGPL,NPTPT,NLDPL,KILL)                         MON11170
C     SET LOAD PRINTING FLAG FOR FIRST LOADING                          MON11180   89
      LDNOPR = 1 - IOPT(10)                                             MON11190
   18 CONTINUE                                                          MON11200
C     READ RESTART OPTIONS                                              MON11210
      READ(5,1001) (IRES(I),I=1,NRES)                                   MON11220   91
      KILL=KILL+IRES(1)                                                 MON11230
      WRITE(6,1112)                                                     MON11240   98
      DO 8 IC=1,NRES                                                    MON11250
    8 WRITE(6,1113) IC,IRES(IC)                                         MON11260  102
      READ(5,1013)  IIMITE,CCBETA,CONFAC                                MON11270  105
      WRITE(6,1105) IIMITE,CCBETA,CONFAC                                MON11280  107
      IF((IIMITE.GT.0) .AND. (CCBETA.GT.0.)) GO TO 11                   MON11290
      WRITE(6,1144)                                                     MON11300  110
      KILL = KILL + 1                                                   MON11310
   11 CONTINUE                                                          MON11320
C                                                                       MON11330
C     READ BOUNDARY CONDITIONS FOR LEADING AND TRAILING EDGE            MON11340
      WRITE(6,1104)                                                     MON11350  111
      CALL BCON(IBC,BC,DLBC,KS,MLIST,KILL)                              MON11360
C                                                                       MON11370
C     READ SEGMENT DATA                                                 MON11380  112
      WRITE(6,1120)                                                     MON11390  113
      DO 300 IJK=1,NSGMTS                                               MON11400
      ISGMT=IJK                                                         MON11410
      READ(5,1011) ISG1,MINT,(SG(I,IJK),I=1,4)                          MON11420  117
      WRITE(6,1121) IJK,MINT,SG(1,IJK),SG(2,IJK)                        MON11430  125
```

```
      WRITE(6,1136) SG(3,IJK),SG(4,IJK)                                 MON11440   128
      NPSEG(IJK)=MINT+1                                                 MON11450
C     INSERT GEOM CALC. HERE                                            MON11460
      NPTS=NPTS+MINT+1                                                  MON11470
      IF(NPTS.LE.MAXPTS) GO TO 10                                       MON11480
      WRITE(6,1052)                                                     MON11490   135
      KILL=KILL+1                                                       MON11500
   10 IF(MINT.LE.MAXINT) GO TO 20                                       MON11510
      WRITE(6,1050) ISGMT                                               MON11520   138
      KILL=KILL+1                                                       MON11530
   20 FMINT=MINT                                                        MON11540
      IF((SG(1,IJK)+SG(2,IJK)).LE.FMINT) GO TO 30                       MON11550
      WRITE(6,1051) ISGMT                                               MON11560   144
      KILL=KILL+1                                                       MON11570
   30 CONTINUE                                                          MON11580
C                                                                       MON11590
C     CALCULATE STARTING INDEX FOR THIS SEGMENT                         MON11600
      IF(ISGMT.NE.1) GO TO 40                                           MON11610
      LSP(ISGMT)=1                                                      MON11620
      GO TO 50                                                          MON11630
   40 IK=ISGMT-1                                                        MON11640
      LSP(ISGMT)=LSP(IK)+NPSEG(IK)                                      MON11650
   50 CONTINUE                                                          MON11660
      READ(5,1013) ISG2,(SG(I,IJK),I=5,8),ECY(IJK),ECR(IJK)             MON11670   156
      WRITE(6,1122) SG(5,IJK),SG(6,IJK)                                 MON11680   164
      WRITE(6,1123) SG(7,IJK),SG(8,IJK)                                 MON11690   167
      WRITE(6,1137) ECY(IJK),ECR(IJK)                                   MON11700   170
      READ(5,1013) ISG3,HI(1,IJK),HI(2,IJK),EI(1,IJK),EI(2,IJK),        MON11710
     1 FMUI(1,IJK),FMUI(2,IJK)                                          MON11720   173
      WRITE(6,1124)                                                     MON11730   181
      WRITE(6,1125)  EI(1,IJK),HI(1,IJK),FMUI(1,IJK)                    MON11740   182
      WRITE(6,1126)  EI(2,IJK),HI(2,IJK),FMUI(2,IJK)                    MON11750   186
      RI(IJK) = SG(5,IJK)                                               MON11760
      IF(ISG1+ISG2+ISG3.EQ.3*IJK) GO TO 60                              MON11770
      WRITE(6,1059) IJK                                                 MON11780   194
      KILL=KILL+1                                                       MON11790
   60 CONTINUE                                                          MON11800
C                                                                       MON11810
```

```
C     READ LOAD AND ELASTIC SUPPORT DATA                                MON11820
      DO 250 IJH=1,KS                                                   MON11830
      IFTNO = MLIST(IJH)                                                MON11840
      READ(5,1011) ISEG,IFOU,(FLDI(I,IJK,IJH),I=1,5)                    MON11850   199
      WRITE(6,1127) IFTNO                                               MON11860   206
      WRITE(6,1128)                                                     MON11870   207
      WRITE(6,1125) (FLDI(I,IJK,IJH),I=1,5)                             MON11880   208
      READ(5,1011) ISEG1,IFOU1,(FLDF(I,IJK,IJH),I=1,5)                  MON11890   213
      WRITE(6,1126) (FLDF(I,IJK,IJH),I=1,5)                             MON11900   220
      WRITE(6,1129)                                                     MON11910   225
      WRITE(6,1128)                                                     MON11920   226
      READ(5,1011) ISEG2,IFOU2,(DLDI(I,IJK,IJH),I=1,5)                  MON11930   227
      WRITE(6,1125) (DLDI(I,IJK,IJH),I=1,5)                             MON11940   234
      READ(5,1011) ISEG3,IFOU3,(DLDF(I,IJK,IJH),I=1,5)                  MON11950   239
      WRITE(6,1126) (DLDF(I,IJK,IJH),I=1,5)                             MON11960   246
      IF(ISEG+ISEG1+ISEG2+ISEG3.EQ.4*IJK) GO TO 100                     MON11970
      WRITE(6,1060) IJK,IJH                                             MON11980   253
      KILL=KILL+1                                                       MON11990
  100 IF(IFOU+IFOU1+IFOU2+IFOU3.EQ.4*MLIST(IJH)) GO TO 110              MON12000
      WRITE(6,1060) IJK,IJH                                             MON12010   257
      KILL=KILL+1                                                       MON12020
  110 CONTINUE                                                          MON12030
C     READ ELASTIC SUPPORT DATA                                         MON12040
      ISEG1=0                                                           MON12050
      IHARM=0                                                           MON12060
      WRITE(6,1130)                                                     MON12070   260
      DO 130 I=1,2                                                      MON12080
      READ(5,1011) ISEG,IFOU ,FMPO(I,IJK,IJH),FNRO(I,IJK,IJH),          MON12090
     1 FNYO(I,IJK,IJH),FNPTO(I,IJK,IJH)                                 MON12100   262
      IF(I.EQ.1) GO TO 125                                              MON12110
      WRITE(6,1129)                                                     MON12120   271
  125 WRITE(6,1132)                                                     MON12130   272
      WRITE(6,1131) FMPO(I,IJK,IJH),FNRO(I,IJK,IJH),FNYO(I,IJK,IJH),    MON12140
     1 FNPTO(I,IJK,IJH)                                                 MON12150   273
      ISEG1=ISEG1+ISEG                                                  MON12160
      IHARM =IHARM+IFOU                                                 MON12170
  130 CONTINUE                                                          MON12180
      READ(5,1011) ISEG,IFOU,RSS(IJK,IJH),ECYS(IJK,IJH),ECRS(IJK,IJH)   MON12190   280
```

```
      WRITE(6,1133) RSS(IJK,IJH),ECYS(IJK,IJH),ECRS(IJK,IJH)          MON12200   286
      READ(5,1011) ISEG2,IFDU1,(  SCF(I,IJK,IJH),I=1,5)               MON12210   290
      READ(5,1012) (SCF(I,IJK,IJH),I=6,16)                            MON12220   297
      WRITE(6,1134)                                                   MON12230   302
      DO 145 IC=1,16                                                  MON12240
  145 WRITE(6,1135) IC,SCF(IC,IJK,IJH)                                MON12250   305
      IF(ISEG+ISEG1+ISEG2.EQ.4*IJK) GO TO 140                         MON12260
      WRITE(6,1061) IJK,IJH                                           MON12270   310
      KILL=KILL+1                                                     MON12280
  140 IF((IHARM+IFDU+IFDU1).EQ.4*MLIST(IJH)) GO TO 150                MON12290
      WRITE(6,1061) IJK,IJH                                           MON12300   314
      KILL=KILL+1                                                     MON12310
  150 CONTINUE                                                        MON12320
  250 CONTINUE                                                        MON12330
  300 CONTINUE                                                        MON12340
      IF(IRES(3).EQ.0) GO TO 350                                      MON12350
      IF(IRES(4) .EQ. 1) GO TO 350                                    MON12355
      WRITE(6,1140)                                                   MON12360
C     READ SCALE FACTORS FOR RESTART OPTION, ALL HARMONICS            MON12370   326
      DO 330 IJH=1,KS                                                 MON12380
      READ(5,1014) IHARM,(STP(I,IJH),I=1,12)                          MON12390   329
      WRITE(6,1141) IHARM,(STP(I,IJH),I=1,12)                         MON12400   335
      IF(IHARM-MLIST(IJH)) 310,320,310                                MON12410
  310 WRITE(6,1142)                                                   MON12420   342
      KILL = KILL+1                                                   MON12430
  320 CONTINUE                                                        MON12440
  330 CONTINUE                                                        MON12450
  350 CONTINUE                                                        MON12460
      IF(KILL.EQ.0) GO TO 450                                         MON12470
      WRITE(6,1150) KILL                                              MON12480   350
      GO TO 3030                                                      MON12490
  450 CONTINUE                                                        MON12500
      KKKKK=2                                                         MON12510
      RETURN                                                          MON12520
 3030 STOP                                                            MON12530
      END                                                             MON12540
```

```
$IBFTC PDINX   NOLIST,DECK,M94,XR7,NODD
```

```
C     READ PLOTTING AND PRINTING DATA                                  PDIN0010
      SUBROUTINE PDIN(IOPT,MB,NSGPL,NPTPT,NLDPL,KILL)                  PDIN0020
      DIMENSION IOPT( 1),MB(10,1),NSGPL( 1),NPTPT( 1),NLDPL( 1)        PDIN0030
 1000 FORMAT(7I10)                                                     PDIN0040
 1003 FORMAT(10I5)                                                     PDIN0050
 1005 FORMAT(1H0,10X,45HTOO MANY POINTS FOR INDIVIDUAL POINT PLOTTING) PDIN0060
 1006 FORMAT(1H0,10X,36HTOO MANY LOADINGS INPUT FOR PLOTTING)          PDIN0070
 1010 FORMAT(1H0,18HSEGMENT PLOT FLAGS/5X,7HSEGMENT,5X,4HFLAG/         PDIN0080
     1 7(7X,I2,10X,I2/) )                                              PDIN0090
 1011 FORMAT(1H0,25HINDIVIDUAL POINT PLOTTING/5X,7HSEGMENT,5X,5HPOINT/ PDIN0100
     1 10(7X,I2,10X,I2/) )                                             PDIN0110
 1012 FORMAT(1H0,5X,17HTHETA INCREMENT =,I3,8H DEGREES)                PDIN0120
 1020 FORMAT(1H0,25HLOAD LEVEL PLOTTING FLAGS/3X,10HLOAD LEVEL,5X,     PDIN0130
     1 4HFLAG/10(7X,I2,10X,I2/) )                                      PDIN0140
      MAXT = 180                                                       PDIN0150
C     SET INPUT FLAGS                                                  PDIN0160
      IF(IOPT( 8).EQ.0) IOPT( 8) =  1                                  PDIN0170
      IF(IOPT(10).EQ.0) IOPT(10) =  1                                  PDIN0180
C     SET THETA INCREMENT                                              PDIN0190
      IF(IOPT( 9))  40, 40, 60                                         PDIN0200
   40 CONTINUE                                                         PDIN0210
      IOPT( 9) =  20                                                   PDIN0220
      GO TO 80                                                         PDIN0230
   60 CONTINUE                                                         PDIN0240
      IOPT( 9) = MAXT/IOPT( 9)                                         PDIN0250
   80 CONTINUE                                                         PDIN0260
      WRITE(6,1012) IOPT( 9)                                           PDIN0270
C     ARE SEGMENTS TO BE PLOTTED                                       PDIN0280   11
      NT = IOPT(11)                                                    PDIN0290
      IF(NT) 120,120,100                                               PDIN0300
  100 CONTINUE                                                         PDIN0310
C     READ SEGMENT PLOT FLAGS                                          PDIN0320
      READ(5,1000) (NSGPL(I),I=1,NT)                                   PDIN0330   14
      WRITE(6,1010) (I,NSGPL(I),I=1,NT)                                PDIN0340   21
  120 CONTINUE                                                         PDIN0350
C     READ INDIVIDUAL POINT PLOTTING FLAGS                             PDIN0360
      NT = IOPT(12)                                                    PDIN0370
```

```
      IF(NT) 200,200,140                                                PDIN0380
  140 CONTINUE                                                          PDIN0390
      READ(5,1003) (MB(I,1),MB(I,2), I=1,NT)                            PDIN0400   31
      WRITE(6,1011) (MB(I,1),MB(I,2) ,I=1,NT)                           PDIN0410   39
      IF(NT-10) 160,160,150                                             PDIN0420
  150 CONTINUE                                                          PDIN0430
C     PRINT MESSAGE,TOO MANY POINTS INPUT FOR INDIVIDUAL POINT PLOTTING PDIN0440
      WRITE(6,1005)                                                     PDIN0450   48
      KILL = KILL + 1                                                   PDIN0460
      NT = 10                                                           PDIN0470
  160 CONTINUE                                                          PDIN0480
      IF(IOPT(13)) 170,170,180                                          PDIN0490
  170 CONTINUE                                                          PDIN0500
      IOPT(13) =  5                                                     PDIN0510
      GO TO 190                                                         PDIN0520
  180 CONTINUE                                                          PDIN0530
      IOPT(13) = MAXT/IOPT(13)                                          PDIN0540
      IF(IOPT(13).LT.2) IOPT(13) = 2                                    PDIN0550
  190 CONTINUE                                                          PDIN0560
      WRITE(6,1012) IOPT(13)                                            PDIN0570   57
  200 CONTINUE                                                          PDIN0580
      IF(IOPT(14)) 300,300,220                                          PDIN0590
C     READ LOADING PLOT  FLAGS                                          PDIN0600
  220 CONTINUE                                                          PDIN0610
      NT = IOPT(14)                                                     PDIN0620
      READ(5,1003) (NLDPL(I),I=1,NT)                                    PDIN0630   61
      WRITE(6,1020) (I,NLDPL(I),I=1,NT)                                 PDIN0640   68
      IF(NT-10) 300,300,230                                             PDIN0650
  230 CONTINUE                                                          PDIN0660
      WRITE(6,1006)                                                     PDIN0670   76
      KILL = KILL + 1                                                   PDIN0680
  300 CONTINUE                                                          PDIN0690
      RETURN                                                            PDIN0700
      END                                                               PDIN0710
```

```
$IBFTC BCONX   NOLIST,DECK,M94,XR7,NODD
```

```
C                    READ BOUNDARY CONDITIONS                          BCON0010
      SUBROUTINE BCON(IBC,BC,DLBC,KS,MLIST,KILL)                       BCON0020
      DIMENSION IBC(4,7,2),BC(4,7,2),DLBC(4,7,2),MLIST(7)              BCON0030
 1000 FORMAT(2I10,5F10.0)                                              BCON0040
 1050 FORMAT(1H0,10X,32HBOUNDARY CONDITIONS NOT IN ORDER)              BCON0050
 1051 FORMAT(1H0,10X,41HBOUNDARY CONDITIONS DUPLICATED, HARMONIC=,I2)  BCON0060
 1105 FORMAT(1H ,12HLEADING EDGE/)                                     BCON0070
 1106 FORMAT(19H HARMONIC  B.C. NO.,5X,13HINITIAL VALUE,7X,9HINCREMENT/)BCON0080
 1107 FORMAT(4X,I2,8X,I2,8X,1PE13.6,5X,E13.6)                          BCON0090
 1108 FORMAT(1H0,13HTRAILING EDGE/)                                    BCON0100
C     READ LEADING AND TRAILING BOUNDARY CONDITIONS                    BCON0110
      DO 50 IL=1,2                                                     BCON0120
      IF(IL.EQ.2) GO TO 10                                             BCON0130
      WRITE(6,1105)                                                    BCON0140   7
      GO TO 20                                                         BCON0150
   10 WRITE(6,1108)                                                    BCON0160   9
   20 WRITE(6,1106)                                                    BCON0170   10
      DO 50 IJ=1,KS                                                    BCON0180
      DO 40 IK=1,4                                                     BCON0190
      READ(5,1000) IHARM,IBC(IK,IJ,IL),BC(IK,IJ,IL),DLBC(IK,IJ,IL)     BCON0200   14
      WRITE(6,1107) IHARM,IBC(IK,IJ,IL),BC(IK,IJ,IL),DLBC(IK,IJ,IL)    BCON0210   19
      IF(IHARM.EQ.MLIST(IJ)) GO TO 40                                  BCON0220
      WRITE(6,1050)                                                    BCON0230   26
      KILL=KILL+1                                                      BCON0240
   40 CONTINUE                                                         BCON0250
      DO 45 IK=1,3                                                     BCON0260
      IN=IK+1                                                          BCON0270
      DO 45 IM=IN,4                                                    BCON0280
      IF(IBC(IK,IJ,IL).NE.IBC(IM,IJ,IL)) GO TO 45                      BCON0290
      WRITE(6,1051) MLIST(IJ)                                          BCON0300   38
      KILL=KILL+1                                                      BCON0310
   45 CONTINUE                                                         BCON0320
   50 CONTINUE                                                         BCON0330
      RETURN                                                           BCON0340
      END                                                              BCON0350
```

MON1X STORAGE MAP

SUBROUTINE MON1

COMMON VARIABLES

	COMMON BLOCK		COMAA		ORIGIN	00001	LENGTH	00001
SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ICHECK	00000	I						
	COMMON BLOCK		COMBB		ORIGIN	00002	LENGTH	00002
CCBETA	00000	R		IIMITE	00001	I		
	COMMON BLOCK		COMCC		ORIGIN	00004	LENGTH	00001
CONFAC	00000	R						
	COMMON BLOCK		//		ORIGIN	00005	LENGTH	33167
NSGMTS	00000	I		KS	00001	I	ISGMT	00002
IFTNO	00003	I		LOADNO	00004	I	LDNOPR	00005
NLOAD	00006	I		KKKKK	00007	I	PHI	00010
SINP	00017	R		COSP	00026	R	NPSEG	00035
LSP	00044	I		MLIST	00053	I	ECR	00062
ECY	00071	R		IOPT	00100	I	IRES	00116
NREC	00130	I		INTF	00155	I	NSGPL	00167
NPTPT	00201	I		NLDPL	00213	I	EI	00225
HI	00243	R		FMUI	00261	R	RI	00277
FLDI	00306	R		FLDF	00673	R	DLDI	01260
DLDF	01645	R		FMPO	02232	R	FNRO	02374
FNYO	02536	R		FNPTO	02700	R	ECRS	03042
ECYS	03123	R		RSS	03204	R	SCF	03265
IBC	04705	I		BC	04775	R	DLBC	05065
S	05155	R		FNTH	05631	R	FNPH	06305
DPHI	06761	R		OTHE	07435	R	STP	10111
AK	10235	R		AU	16461	R	AL	24705
RHV	31755	R		SAVE	33105	R	AK	10235
SG	10235	R		MB	10330	I		

STORAGE MAP

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
AA	33174	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
I	33210	I	KILL	33211	I	NOPT	33212
NRES	33213	I	MAXSEG	33214	I	MAXKS	33215
MAXINT	33216	I	MAXPTS	33217	I	NPTS	33220
NZERO	33221	I	IC	33222	I	IJK	33223
ISG1	33224	I	MINT	33225	I	FMINT	33226
IK	33227	I	ISG2	33230	I	ISG3	33231
IJH	33232	I	ISEG	33233	I	IFOU	33234
ISEG1	33235	I	IFOU1	33236	I	ISEG2	33237
IFOU2	33240	I	ISEG3	33241	I	IFOU3	33242
IHARM	33243	I					

ENTRY POINTS

MON1	SECTION	10

SUBROUTINES CALLED

.FRDD.	SECTION	11	.FWRD.	SECTION	12	PDIN	SECTION
BCDN	SECTION	14	.EXIT.	SECTION	15	.UN05.	SECTION
.FRTN.	SECTION	17	.FCNV.	SECTION	18	.UN06.	SECTION
.FFIL.	SECTION	20	E.1	SECTION	21	E.2	SECTION
E.3	SECTION	23	E.4	SECTION	24	CC.1	SECTION
CC.2	SECTION	26	CC.3	SECTION	27	CC.4	SECTION
SYSLOC	SECTION	29					

EFN IFN CORRESPONDENCE

MONIX STORAGE MAP

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
1000	FORMAT	33273	1001	FORMAT	33274	1002	FORMAT
1003	FORMAT	33276	1011	FORMAT	33300	1012	FORMAT
1013	FORMAT	33305	1014	FORMAT	33307	1040	FORMAT
1041	FORMAT	33326	1050	FORMAT	33340	1051	FORMAT
1052	FORMAT	33366	1059	FORMAT	33400	1060	FORMAT
1061	FORMAT	33423	1100	FORMAT	33440	1101	FORMAT
1102	FORMAT	33470	1103	FORMAT	33474	1104	FORMAT
1105	FORMAT	33505	1110	FORMAT	33526	1111	FORMAT
1112	FORMAT	33536	1113	FORMAT	33543	1120	FORMAT
1121	FORMAT	33553	1123	FORMAT	33577	1122	FORMAT
1124	FORMAT	33621	1125	FORMAT	33627	1126	FORMAT
1127	FORMAT	33641	1128	FORMAT	33650	1129	FORMAT
1130	FORMAT	33666	1131	FORMAT	33674	1132	FORMAT
1133	FORMAT	33712	1134	FORMAT	33734	1135	FORMAT
1136	FORMAT	33751	1137	FORMAT	33766	1140	FORMAT
1141	FORMAT	34002	1142	FORMAT	34011	1144	FORMAT
1150	FORMAT	34036	1	7A	34052	12.	25A
3	52A	34217	6	55A	34235	3030	354A
2	68A	34273	16	82A	34341	18	91A
8	102A	34433	11	111A	34521	300	318A
10	136A	34664	20	139A	34704	30	145A
40	151A	34747	50	156A	34766	60	195A
250	316A	36150	100	254A	35545	110	258A
130	279A	35723	125	272A	35671	145	305A
140	311A	36121	150	315A	36150	350	347A
330	344A	36263	310	342A	36252	320	343A
450	352A	36302					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 36342.

SUBROUTINE PDIN

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
MAXT	00001	I	NT	00002	I	I	00003

ENTRY POINTS

PDIN	SECTION	3

SUBROUTINES CALLED

.FWRD.	SECTION	4	.FRDD.	SECTION	5	.UN06.	SECTION
.FFIL.	SECTION	7	.FCNV.	SECTION	8	.UN05.	SECTION
.FRTN.	SECTION	10	SYSLOC	SECTION	11		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
1000	FORMAT	00021	1003	FORMAT	00022	1005	FORMAT
1006	FORMAT	00035	1010	FORMAT	00046	1011	FORMAT
1012	FORMAT	00102	1020	FORMAT	00112	40	8A
60	10A	00153	80	11A	00160	120	28A
100	14A	00175	200	58A	00352	140	31A
160	50A	00321	150	48A	00306	170	52A
180	54A	00327	190	57A	00342	300	77A
220	60A	00355	230	75A	00423		

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00537.

BCONX STORAGE MAP

SUBROUTINE BCON

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IL	00001	I	IJ	00002	I	IK	00003
IHARM	00004	I	IN	00005	I		

ENTRY POINTS

BCON	SECTION	3

SUBROUTINES CALLED

.FWRD.	SECTION	4	.FRDD.	SECTION	5	.UN06.	SECTION
.FFIL.	SECTION	7	.FCNV.	SECTION	8	.UN05.	SECTION
.FRTN.	SECTION	10	SYSLOC	SECTION	11		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
1000	FORMAT	00017	1050	FORMAT	00022	1051	FORMAT	
1105	FORMAT	00044	1106	FORMAT	00050	1107	FORMAT	
1108	FORMAT	00070	50	43A	00315	10	9A	
20	10A	00123	40	27A	00233	45	40A	

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00401.

```
$IBFTC MON2X   NOLIST,DECK,M94,XR7,NODD
```

```
C                 MONITER FOR REGION 2--BUILD GEOMETRY                  MON20010
      SUBROUTINE MON2                                                   MON20020
C     CALCULATE GEOMETRY,FINITE DIFFERENCE COEFFICIENTS, STIFFNESSES-D,KMON20030
      DIMENSION Y(300) ,R(300),D(300),K(300),FMU(300),E(300),H(300),    MON20040
     1 FDC(300),SG( 8,7),MB(10,2),PS(600,7),RHM(2400)                   MON20050
      EQUIVALENCE (AK(60,1),MB(1,1))                                    MON20060
      EQUIVALENCE (AK(1,1),SG(1,1)),(AK(1,2),Y(1)),(AK(1,5),R(1)),      MON20070
     1 (AK(1, 8),D(1)),(AK(1,11),K(1)),(AK(1,14),FMU(1)),               MON20080
     2 (AK(20,18),PS(1,1)),(AL(1,2),E(1)),(AL(101,3),H(1)),             MON20090
     3 (AL(1,5),FDC(1)), (AL(1,1),RHM(1))                               MON20100
      REAL K                                                            MON20110
      COMMON/COMAA/ICHECK                                               MON20120
      COMMON NSGMTS, KS, ISGMT, IFTNO, LOADNO, LDNOPR, NLOAD, KKKKK     MON20130
      COMMON PHI(7),SINP(7),COSP(7),NPSEG(7),LSP(7),MLIST(7),ECR(7),    MON20140
     1 ECY(7),IOPT(14),IRES(10),NREC(21),INTF(10),NSGPL(10),NPTPT(10),  MON20150
     2 NLDPL(10)                                                        MON20160
      COMMON EI(2,7),HI(2,7),FMJI(2,7),RI(7)                            MON20170
      COMMON FLDI(5,7,7),FLDF(5,7,7),DLDI(5,7,7),DLDF(5,7,7)            MON20180
      COMMON FMPO(2,7,7),FNRO(2,7,7),FNYO(2,7,7),FNPTO(2,7,7),ECRS(7,7),MON20190
     1 ECYS(7,7),RSS(7,7),SCF(16,7,7)                                   MON20200
      COMMON IBC(4,7,2),BC(4,7,2),DLBC(4,7,2)                           MON20210
      COMMON S(300),FNTH(300),FNPH(300),OPHI(300),OTHE(300),STP(12,7)   MON20220
      COMMON AK(140,23),AU(140,23),AL(200,13),RHV(600),SAVE(5,10)       MON20230
 1000 FORMAT(1H0,30HFINITE DIFFERENCE COEFFICIENTS/)                    MON20240
 1001 FORMAT(5X,I4,6(1PE16.7))                                          MON20250
 1002 FORMAT(5X,14HSEGMENT NUMBER,I2/5X,5HPOINT,5X,3HFDC)               MON20260
 1003 FORMAT(1H0,1X,7HSEGMENT,I3//3X,14HPHI(DEGREES) =,F11.6/3X,        MON20270
     1 8HSIN(PHI),5X,1H=,F11.6/3X,8HCOS(PHI),5X,1H=,F11.6)              MON20280
 1004 FORMAT(2X,I3,2X,6(1PE16.6))                                       MON20290
 1005 FORMAT(1H1,8HGEOMETRY/)                                           MON20300
 1006 FORMAT(7H  POINT,10X,1HS,15X,1HR,15X,1HY/)                        MON20310
 1007 FORMAT(1H0,11HSTIFFNESSES/)                                       MON20320
 1008 FORMAT(1H0,7HSEGMENT,I2,/2X,5HPOINT,10X,1HE,15X,1HH,14X,2HMU,15X, MON20330
     1 1HD,15X,1HK/)                                                    MON20340
 1010 FORMAT(1H0,10X,64HRESTART OPTION 4 SPECIFIED WHEN FIRST HARMONIC NMON20344
     1OT EQUAL TO ZERO//10X,14HTERMINATE CASE)                          MON20345
 1250 FORMAT(1H0,10X,36HEXECUTION TERMINATED BY INPUT OPTION)           MON20350
```

```
      MAXPTS=300                                                        MON20360
      MAXMPT = 17                                                       MON20370
      MAXSEG=7                                                          MON20380
      DO 100 IJK=1,NSGMTS                                               MON20390
      ISGMT=IJK                                                         MON20400
      MS=LSP(IJK)                                                       MON20410
      MF=MS+NPSEG(ISGMT)-1                                              MON20420
C                                                                       MON20430
C     CALCULATE GEOMETRY                                                MON20440
      CALL GEOM(PHI,SINP,COSP,SG,S,MAXPTS,MAXSEG,MS,MF,ISGMT,Y,NPSEG,R) MON20450
C     SET XBAR AND ZBAR FROM RADIAL AND AXIAL LOADINGS                  MON20460   13
      DO 40 IJH = 1,KS                                                  MON20470
      FLDI(1,IJK,IJH) = FLDI(1,IJK,IJH)                                 MON20480
     1  + FLDI(4,IJK,IJH) * COSP(IJK) + FLDI(5,IJK,IJH) * SINP(IJK)     MON20490
      FLDF(1,IJK,IJH) = FLDF(1,IJK,IJH)                                 MON20500
     1  + FLDF(4,IJK,IJH) * COSP(IJK) + FLDF(5,IJK,IJH) * SINP(IJK)     MON20510
      DLDI(1,IJK,IJH) = DLDI(1,IJK,IJH)                                 MON20520
     1  + DLDI(4,IJK,IJH) * COSP(IJK) + DLDI(5,IJK,IJH) * SINP(IJK)     MON20530
      DLDF(1,IJK,IJH) = DLDF(1,IJK,IJH)                                 MON20540
     1  + DLDF(4,IJK,IJH) * COSP(IJK) + DLDF(5,IJK,IJH) * SINP(IJK)     MON20550
      FLDI(3,IJK,IJH) = FLDI(3,IJK,IJH)                                 MON20560
     1  + FLDI(4,IJK,IJH) * SINP(IJK) - FLDI(5,IJK,IJH) * COSP(IJK)     MON20570
      FLDF(3,IJK,IJH) = FLDF(3,IJK,IJH)                                 MON20580
     1  + FLDF(4,IJK,IJH) * SINP(IJK) - FLDF(5,IJK,IJH) * COSP(IJK)     MON20590
      DLDI(3,IJK,IJH) = DLDI(3,IJK,IJH)                                 MON20600
     1  + DLDI(4,IJK,IJH) * SINP(IJK) - DLDI(5,IJK,IJH) * COSP(IJK)     MON20610
      DLDF(3,IJK,IJH) = DLDF(3,IJK,IJH)                                 MON20620
     1  + DLDF(4,IJK,IJH) * SINP(IJK) - DLDF(5,IJK,IJH) * COSP(IJK)     MON20630
   40 CONTINUE                                                          MON20640
C                                                                       MON20650
C     CALCULATE FINITE DIFFERENCE COEFFICIENTS                          MON20660
      MF1=MF-1                                                          MON20670
      DO 50 IJ=MS,MF1                                                   MON20680
      IJ1=IJ+1                                                          MON20690
      FDC(IJ)=1./(S(IJ1)-S(IJ))                                         MON20700
   50 CONTINUE                                                          MON20710
C                                                                       MON20720
C     CALCULATE STIFFNESS TABLE--D AND K                                MON20730
```

```
      DEL=ABS(S(MF)-S(MS))                                      MON20740
      H1 = HI(1,ISGMT)                                          MON20750
      H2 = HI(2,ISGMT)                                          MON20760
      CALL EHMU(H1,H2,H,MS,MF,DEL,S,MAXPTS,ISGMT)               MON20770   80
      E1 = EI(1,ISGMT)                                          MON20780
      E2 = EI(2,ISGMT)                                          MON20790
      CALL EHMU(E1,E2,E,MS,MF,DEL,S,MAXPTS,ISGMT)               MON20800   84
      FMU1 = FMUI(1,ISGMT)                                      MON20810
      FMU2 = FMUI(2,ISGMT)                                      MON20820
      CALL EHMU(FMU1,FMU2,FMU,MS,MF,DEL,S,MAXPTS,ISGMT)         MON20830   88
      CALL DANK(D,K,MS,MF,E,H,FMU,MAXPTS)                       MON20840   90
  100 CONTINUE                                                  MON20850
      IF(IOPT(1).NE.1) GO TO 150                                MON20860
C                                                               MON20870
C     PRINT GEOMETRY                                            MON20880
      WRITE(6,1005)                                             MON20890   96
      DO 140 IJK=1,NSGMTS                                       MON20900
      PHITEM=PHI(IJK)*57.29578                                  MON20910
      WRITE(6,1003) IJK,PHITEM,SINP(IJK),COSP(IJK)              MON20920   101
      WRITE(6,1006)                                             MON20930   104
      MS=LSP(IJK)                                               MON20940
      MF=MS+NPSEG(IJK)-1                                        MON20950
      KK=0                                                      MON20960
      DO 130 IJ=MS,MF                                           MON20970
      KK=KK+1                                                   MON20980
  130 WRITE(6,1004) KK,S(IJ),R(IJ),Y(IJ)                        MON20990   112
  140 CONTINUE                                                  MON21000
  150 IF(IOPT(2).NE.1) GO TO 200                                MON21010
C                                                               MON21020
C     PRINT FINITE DIFFERENCE COEFFICIENTS                      MON21030
      WRITE(6,1000)                                             MON21040   122
      DO 180 IJK=1,NSGMTS                                       MON21050
      WRITE(6,1002) IJK                                         MON21060   125
      MS=LSP(IJK)                                               MON21070
      MF=MS+NPSEG(IJK)-1                                        MON21080
      MF1=MF-1                                                  MON21090
      KK=0                                                      MON21100
      DO 180 IJ=MS,MF1                                          MON21110
```

```
      KK=KK+1                                                           MON21120
      WRITE(6,1001) KK,FDC(IJ)                                          MON21130   134
  180 CONTINUE                                                          MON21140
  200 IF(IOPT(3).NE.1) GO TO 250                                        MON21150
C                                                                       MON21160
C     PRINT STIFFNESSES--D AND K                                        MON21170
      WRITE(6,1007)                                                     MON21180   142
      DO 240 IJK=1,NSGMTS                                               MON21190
      MS=LSP(IJK)                                                       MON21200
      MF=MS+NPSEG(IJK)-1                                                MON21210
      WRITE(6,1008) IJK                                                 MON21220   149
      KK=0                                                              MON21230
      DO 230 IJ=MS,MF                                                   MON21240
      KK=KK+1                                                           MON21250
  230 WRITE(6,1004) KK,E(IJ),H(IJ),FMU(IJ),D(IJ),K(IJ)                  MON21260   153
  240 CONTINUE                                                          MON21270
  250 CONTINUE                                                          MON21280
      KKKKK=3                                                           MON21290
      IF(IRES(2).NE.1) GO TO 260                                        MON21300
      WRITE(6,1250)                                                     MON21310   167
      CALL EXIT                                                         MON21320   168
  260 IF(IRES(3).EQ.0) GO TO 300                                        MON21330
C     CALCULATE NUMBER OF POINTS PER RECORD FOR RESTART OPTION          MON21340
      CALL NRPR(LSP,NPSEG,NREC,NSGMTS,MAXMPT)                           MON21350
C     TRANSFER' OLD' PRIOR SOLUTION TAPE TO 'NEW' PRIOR SOLUTION TAPE   MON21360   172
      REWIND 3                                                          MON21370   173
      REWIND 4                                                          MON21380   174
      KK = 0                                                            MON21385
      DO 280 IJK = 1,NSGMTS                                             MON21390
      NROS = 8 * NPSEG(IJK)                                             MON21393
      READ(03)  ((PS(I,J),I=1,NROS),J=1,KS)                             MON21396   180
      IF(IRES(4).EQ.0) GO TO 280                                        MON21398
      IF(MLIST(1).EQ.0) GO TO 265                                       MON21400
      WRITE(6,1010)                                                     MON21402   193
      REWIND 3                                                          MON21403   194
      KKKKK = 8                                                         MON21404
      RETURN                                                            MON21406
  265 CONTINUE                                                          MON21408
```

```
C     MOVE SOLUTION VECTOR FOR M=0 TO IN-CORE STORAGE ARRAY          MON21410
      DO 275 I = 1,NRCS                                              MON21412
      KK = KK + 1                                                    MON21414
  275 RHM(KK) = PS(I,1)                                              MON21416
  280 WRITE(04) ((PS(I,J),I=1,NRCS),J=1,KS)                          MON21420   203
      REWIND 3                                                       MON21430   212
      REWIND 4                                                       MON21440
C     SET SPECIAL ITERATION FLAG = 0                                 MON21450   213
      INTF(1) = 1                                                    MON21455
      INTF(2) = 0                                                    MON21460
C     SET OVERLAY FLAG FOR REGION 5                                  MON21470
      KKKKK = 5                                                      MON21480
C     SET HARMONIC INDEX FOR FIRST HARMONIC                          MON21490
      INTF(5) = 1                                                    MON21500
C     SET ORDINARY ITERATION FLAG FOR ZERO'TH ITERATION              MON21510
      INTF(6) = 0                                                    MON21520
      IF(IRES(4).LE.0) GO TO 300                                     MON21522
C     SET FLAGS TO DO ''LAST SPECIAL'' ITERATION                     MON21524
      INTF( 1) = 2                                                   MON21526
      INTF( 2) = 1                                                   MON21527
      IRES( 4) = 1                                                   MON21528
  300 CONTINUE                                                       MON21530
C     IF(ICHECK .EQ. 0) GO TO 350                                    MON21540
      IF(IOPT(8) .EQ. 0)  IOPT(8) = 1                                MON21545
C     CALCULATE INDIVIDUAL POINT PLOT INDICES                        MON21550
      NT = IOPT(12)                                                  MON21560
      IF(NT.EQ.0) GO TO 350                                          MON21570
      DO 320 I=1,NT                                                  MON21580
      IJK = MB(I,1)                                                  MON21590
      NPTPT(I) = MB(I,2) + LSP(IJK) - 1                              MON21600
  320 CONTINUE                                                       MON21610
  350 CONTINUE                                                       MON21620
      RETURN                                                         MON21630
      END                                                            MON21640
```

MON2X STORAGE MAP

SUBROUTINE MON2

COMMON VARIABLES

COMMON BLOCK COMAA ORIGIN 00001 LENGTH 00001

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ICHECK	00000	I					

COMMON BLOCK // ORIGIN 00002 LENGTH 33167

NSGMTS	00000	I	KS	00001	I	ISGMT	00002
IFTNO	00003	I	LOADNO	00004	I	LUNOPR	00005
NLOAD	00006	I	KKKKK	00007	I	PHI	00010
SINP	00017	R	COSP	00026	R	NPSEG	00035
LSP	00044	I	MLIST	00053	I	ECR	00062
ECY	00071	R	IOPT	00100	I	IRES	00116
NREC	00130	I	INTF	00155	I	NSGPL	00167
NPTPT	00201	I	NLDPL	00213	I	EI	00225
HI	00243	R	FMUI	00261	R	RI	00277
FLDI	00306	R	FLDF	00673	R	DLDI	01260
DLDF	01645	R	FMPO	02232	R	FNRO	02374
FNYO	02536	R	FNPTO	02700	R	ECRS	03042
ECYS	03123	R	RSS	03204	R	SCF	03265
IBC	04705	I	BC	04775	R	DLBC	05065
S	05155	R	FNTH	05631	R	FNPH	06305
DPHI	06761	R	DTHE	07435	R	STP	10111
AK	10235	R	AU	16461	R	AL	24705
RHV	31755	R	SAVE	33105	R	AK	10235
MB	10330	I	SG	10235	R	Y	10451
R	11315	R	D	12161	R	K	13025
FMU	13671	R	PS	14774	R	AL	24705
E	25215	R	H	25671	R	FDC	26345
RHM	24705	R					

UNDIMENSIONED PROGRAM VARIABLES

MON2X

STORAGE MAP

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
MAXPTS	33171	I	MAXMPT	33172	I	MAXSEG	33173
IJK	33174	I	MS	33175	I	MF	33176
MF1	33177	I	IJ	33200	I	IJ1	33201
DEL	33202	R	H1	33203	R	H2	33204
E1	33205	R	E2	33206	R	FMU1	33207
FMU2	33210	R	PHITEM	33211	R	KK	33212
NROS	33213	I	J	33214	I	NT	33215

ENTRY POINTS

MON2 SECTION 6

SUBROUTINES CALLED

GEOM	SECTION	7	EHMU	SECTION	8	DANK	SECTION
.FWRD.	SECTION	10	EXIT	SECTION	11	NRPR	SECTION
.FRWT.	SECTION	13	.FRDB.	SECTION	14	.FWRB.	SECTION
.UN06.	SECTION	16	.FFIL.	SECTION	17	.FCNV.	SECTION
.UN03.	SECTION	19	.UN04.	SECTION	20	.FRLR.	SECTION
.FBLT.	SECTION	22	.FBDT.	SECTION	23	.FWLR.	SECTION
SYSLOC	SECTION	25					

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
1000	FORMAT	33242	1001	FORMAT	33251	1002	FORMAT
1003	FORMAT	33264	1004	FORMAT	33307	1005	FORMAT
1006	FORMAT	33316	1007	FORMAT	33324	1008	FORMAT
1010	FORMAT	33345	1250	FORMAT	33366	100	91A
40	65A	33665	50	74A	33715	150	120A
140	117A	34162	130	112A	34142	200	140A
180	136A	34247	250	163A	34347	240	160A
230	153A	34321	260	169A	34366	300	218A
280	203A	34524	265	196A	34506	275	199A

350 2364 34642 320 233A 34640

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 34666.

```
$IBFTC GEOMX   NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD GEOMETRY FOR ONE SEGMENT                        GEOM0010
      SUBROUTINE GEOM(PHI,SINP,COSP,SG,S,MAXPTS,MAXSEG,MS,MF,ISGMT,Y,   GEOM0020
     1 NPSEG,R)                                                         GEOM0030
      DIMENSION PHI(MAXSEG),SINP(MAXSEG),COSP(MAXSEG),SG( 8,MAXSEG),    GEOM0040
     1 S(MAXPTS),Y(MAXPTS),R(MAXPTS),NPSEG(MAXSEG)                      GEOM0050
      MINT=NPSEG(ISGMT)-1                                               GEOM0060
      MONE= SG( 1,ISGMT)                                                GEOM0070
      MTWO= SG( 2,ISGMT)                                                GEOM0080
      DELS1= SG( 3,ISGMT)                                               GEOM0090
      DELSM= SG( 4,ISGMT)                                               GEOM0100
      Z1= SG( 7,ISGMT)-SG( 5,ISGMT)                                     GEOM0110
      Z2=-SG( 8,ISGMT)+SG( 6,ISGMT)                                     GEOM0120
      STOT= SQRT(Z1*Z1+Z2*Z2)                                           GEOM0130
C     CALCULATE ARCTAN(PHI),SIN(PHI),COS(PHI)                           GEOM0140   14
      AFLG=1.                                                           GEOM0150
      BFLG=1.                                                           GEOM0160
      PI2=1.57079633                                                    GEOM0170
      PIX2 = 6.28318531                                                 GEOM0175
      IF(Z1.LT.0.) AFLG=-1.                                             GEOM0180
      IF(Z2.LT.0.) BFLG=-1.                                             GEOM0190
      IF(Z2.NE.0.) GO TO 32                                             GEOM0200
      PHI(ISGMT)=0.+PI2*(1.-AFLG)                                       GEOM0210
      SINP(ISGMT)=0.                                                    GEOM0220
      COSP(ISGMT)=AFLG*1.                                               GEOM0230
      GO TO 36                                                          GEOM0240
   32 IF(Z1.NE.0.) GO TO 34                                             GEOM0250
      PHI(ISGMT)=PI2+PI2*(1.-BFLG)                                      GEOM0260
      COSP(ISGMT)=0.                                                    GEOM0270
      SINP(ISGMT)=BFLG*1.                                               GEOM0280
      GO TO 36                                                          GEOM0290
   34 CONTINUE                                                          GEOM0300
      PHITEM= ATAN2(Z2,Z1)                                              GEOM0310   34
      IF(Z2.LT.0.) PHITEM = PHITEM+PIX2                                 GEOM0320
      SINP(ISGMT)= SIN(PHITEM)                                          GEOM0340   37
      COSP(ISGMT)= COS(PHITEM)                                          GEOM0350   39
      PHI(ISGMT) = PHITEM                                               GEOM0360
   36 CONTINUE                                                          GEOM0370
```

```
C                                                                       GEOM0380
C     CALCULATE S-MESH, R, AND Y VECTORS                                GEOM0390
      CALL MESH(MINT,MONE,MTWO,DELS1,DELSM,STOT,S,MS,MF,MAXPTS)         GEOM0400    43
      Z1= COSP(ISGMT)                                                   GEOM0410
      Z2= SINP(ISGMT)                                                   GEOM0420
      R1=SG( 5,ISGMT)                                                   GEOM0430
      S1=SG( 6,ISGMT)                                                   GEOM0440
      DO 60 IJ=MS,MF                                                    GEOM0450
      R(IJ)=R1+S(IJ)*Z1                                                 GEOM0460
   60 Y(IJ)=S1-S(IJ)*Z2                                                 GEOM0470
      RETURN                                                            GEOM0480
      END                                                               GEOM0490
```

```
$IBFTC MESHX   NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD 'S' MESH FOR ONE SEGMENT                        MESH0010
      SUBROUTINE MESH(MINT,MONE,MTWO,DELS1,DELSM,STOT,S,MS,MF,MAXPTS)   MESH0020
      DIMENSION S(MAXPTS)                                               MESH0030
      ISTRT=MS                                                          MESH0040
      IEND =MF                                                          MESH0050
      FMINT=MINT                                                        MESH0060
      FMONE=MONE                                                        MESH0070
      FMTWO=MTWO                                                        MESH0080
      MINT1=MINT+1                                                      MESH0090
C                                                                       MESH0100
C     CALCULATE DELTA S(C)                                              MESH0110
      FM1=FMONE+1.                                                      MESH0120
      FM2=FMTWO+1.                                                      MESH0130
      IF(MONE+MTWO.NE. 0) GO TO 10                                      MESH0140
      DELSC=STOT/FMINT                                                  MESH0150
      GO TO 40                                                          MESH0160
   10 IF(MONE.NE. 0) GO TO 20                                           MESH0170
      DELSC= (STOT-0.5*FM2*DELSM)/(FMINT-0.5*FM2)                       MESH0180
      GO TO 40                                                          MESH0190
   20 IF(MTWO.NE. 0) GO TO 30                                           MESH0200
      DELSC=(STOT-0.5*FM1*DELS1)/(FMINT-0.5*FM1)                        MESH0210
      GO TO 40                                                          MESH0220
   30 DELSC=(STOT-0.5*(FM1*DELS1+FM2*DELSM))/(FMINT-0.5*(FM1+FM2))      MESH0230
   40 CONTINUE                                                          MESH0240
   70 CONTINUE                                                          MESH0250
      S(IEND)=STOT                                                      MESH0260
      J=1                                                               MESH0270
      IF(MTWO.EQ.0) GO TO 90                                            MESH0280
C                                                                       MESH0290
C     CALCULATE TRAILING INTERVALS                                      MESH0300
      COM=(DELSC-DELSM)/FMTWO                                           MESH0310
      DO 80 I=1,MTWO                                                    MESH0320
      FI=I                                                              MESH0330
      II=IEND-J                                                         MESH0340
      S(II)  =  STOT-FI*(DELSM+0.5*COM*(FI-1.))                         MESH0350
   80 J=J+1                                                             MESH0360
   90 CONTINUE                                                          MESH0370
```

```
      KI=MINT-(MONE+MTWO)                                               MESH0380
      IF(KI.EQ.0) GO TO 110                                             MESH0390
C                                                                       MESH0400
C     CALCULATE MIDDLE INTERVALS                                        MESH0410
      DO 100 I=1,KI                                                     MESH0420
      II=IEND-J                                                         MESH0430
      JJ=II+1                                                           MESH0440
      S(II)=S(JJ)-DELSC                                                 MESH0450
  100 J=J+1                                                             MESH0460
      IF(MONE.EQ.0) GO TO 130                                           MESH0470
  110 CONTINUE                                                          MESH0480
C                                                                       MESH0490
C     CALCULATE LEADING INTERVALS                                       MESH0500
      COM=(DELSC-DELS1)/FMONE                                           MESH0510
      MONE1=MONE-1                                                      MESH0520
      DO 120 I=1,MONE1                                                  MESH0530
      FI=I                                                              MESH0540
      II=IEND-J                                                         MESH0550
      JJ=II+1                                                           MESH0560
      S(II)  =  S(JJ)-(DELS1+COM*(FMONE-FI))                            MESH0570
  120 J=J+1                                                             MESH0580
  130 CONTINUE                                                          MESH0590
      S(MS)=0.                                                          MESH0600
      RETURN                                                            MESH0610
      END                                                               MESH0620
```

```
$IBFTC EHMUX    NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD EACH SHELL PROPERTY FOR ONE SEGMENT                EHMU0010
      SUBROUTINE EHMU(A,B,C,I,J,DEL,S,MAXPTS,ISGMT)                       EHMU0020
      DIMENSION C(MAXPTS),S(MAXPTS)                                       EHMU0030
      IF(A.NE.0..AND.B.NE.0.) GO TO 15                                    EHMU0040
      WRITE(6,1000) ISGMT                                                 EHMU0050    4
 1000 FORMAT(1H0,5X, 7HSEGMENT,I2,34H--ELASTICITY, THICKNESS, OR MU =0.)EHMU0060
      CALL EXIT                                                           EHMU0070    5
   15 IF(A.NE.B) GO TO 20                                                 EHMU0080
      DO 10 IJ=I,J                                                        EHMU0090
   10 C(IJ)=A                                                             EHMU0100
      GO TO 40                                                            EHMU0110
   20 SLOPE=(B-A)/DEL                                                     EHMU0120
      DO 30 IJ=I,J                                                        EHMU0130
   30 C(IJ)=SLOPE*S(IJ)+A                                                 EHMU0140
   40 RETURN                                                              EHMU0150
      END.                                                                EHMU0160
```

```
$IBFTC DANKX  NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD 'D' AND 'K' VECTORS FOR ONE SEGMENT                DANK0010
      SUBROUTINE DANK(D,K,MS,MF,E,H,FMU,MAXPTS)                            DANK0020
      REAL K                                                               DANK0030
      DIMENSION D(MAXPTS),K(MAXPTS),E(MAXPTS),H(MAXPTS),FMU(MAXPTS)        DANK0040
      O12 = 8.3333333E-2                                                   DANK0045
      MU=0                                                                 DANK0050
      IE=0                                                                 DANK0060
      IH=0                                                                 DANK0070
      IF(FMU(MS).EQ.FMU(MF)) MU=1                                          DANK0080
      IF(E(MS).EQ.E(MF)) IE=1                                              DANK0090
      IF(H(MS).EQ.H(MF)) IH=1                                              DANK0100
      COM=1.0/(1.0-FMU(MS)**2)                                             DANK0110
      COM1=H(MS)**2                                                        DANK0120
      IF(MU+IE+IH.LT.3) GO TO 20                                           DANK0130
      AN=(E(MS)*H(MS))*COM                                                 DANK0140
      AM=AN*O12*COM1                                                       DANK0150
      DO 10 IJ=MS,MF                                                       DANK0160
      D(IJ)=AN                                                             DANK0170
   10 K(IJ)=AM                                                             DANK0180
      GO TO 40                                                             DANK0190
   20 CONTINUE                                                             DANK0200
      DO 30 IJ=MS,MF                                                       DANK0210
      IF(MU.NE.1) COM=1.0/(1.0-FMU(IJ)**2)                                 DANK0220
      D(IJ)= E(IJ)*H(IJ)*COM                                               DANK0230
      IF(IH.NE.1) COM1=H(IJ)**2                                            DANK0240
   30 K(IJ)=O12*D(IJ)*COM1                                                 DANK0250
   40 CONTINUE                                                             DANK0260
      RETURN                                                               DANK0270
      END                                                                  DANK0280
```

```
$IBFTC NRPRX    NOLIST,DECK,M94,XR7,NODD
```

```
C                 CALCULATE THE NUMBER OF ROWS PER RECORD          NRPR0010
      SUBROUTINE NRPR(LSP,NPSEG,NREC,NSGMTS,MAXMPT)                NRPR0020
      DIMENSION LSP( 1),NPSEG( 1),NREC( 1)                         NRPR0030
C     DETERMINE TOTAL NUMBER OF POINTS IN SHELL                    NRPR0040
      NPTS=0                                                       NRPR0050
      DO 50 I=1,NSGMTS                                             NRPR0060
   50 NPTS=NPTS+NPSEG(I)                                           NRPR0070
C     INITIALIZE FLAGS AND COUNTERS                                NRPR0080
      J   = 1                                                      NRPR0090
      MPT = 0                                                      NRPR0100
      IND = 0                                                      NRPR0110
      IJK=1                                                        NRPR0120
      MS =LSP(IJK)                                                 NRPR0130
      DO 200 I=1,NPTS                                              NRPR0140
C     TYPICAL POINT IN SEGMENT HAS 8 ROWS                          NRPR0150
      IND = IND+8                                                  NRPR0160
      MPT = MPT+1                                                  NRPR0170
      IF(I.NE.MS) GO TO 100                                        NRPR0180
C     FIRST POINT IN SEGMENT HAS 12 ROWS                           NRPR0190
      IND = IND+4                                                  NRPR0200
      MF  = MS+NPSEG(IJK)-1                                        NRPR0210
      GO TO 150                                                    NRPR0220
  100 CONTINUE                                                     NRPR0230
      IF(I.NE.MF) GO TO 150                                        NRPR0240
C     LAST  POINT IN SEGMENT HAS  4 ROWS                           NRPR0250
      IND = IND-4                                                  NRPR0260
      IJK = IJK+1                                                  NRPR0270
      IF(IJK.GT.NSGMTS) GO TO 170                                  NRPR0280
      MS  = LSP(IJK)                                               NRPR0290
  150 CONTINUE                                                     NRPR0300
      IF(MPT.LT.MAXMPT) GO TO 200                                  NRPR0310
  170 CONTINUE                                                     NRPR0320
C     STORE NUMBER OF ROWS FOR RECORD 'J'                          NRPR0330
      NREC(J) = IND                                                NRPR0340
      IND = 0                                                      NRPR0350
      J   = J+1                                                    NRPR0360
      MPT = 0                                                      NRPR0370
```

```
200 CONTINUE                                    NRPR0380
    RETURN                                      NRPR0390
    END                                         NRPR0400
```

STORAGE MAP

SUBROUTINE GEOM

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
MINT	00001	I	MONE	00002	I	MTWO	00003
DELS1	00004	R	DELSM	00005	R	Z1	00006
Z2	00007	R	STOT	00010	R	AFLG	00011
BFLG	00012	R	PI2	00013	R	PIX2	00014
PHITEM	00015	R	R1	00016	R	S1	00017

ENTRY POINTS

GEOM	SECTION	3

SUBROUTINES CALLED

SQRT	SECTION	4	ATAN2	SECTION	5	SIN	SECTION	
COS	SECTION	7	MESH	SECTION	8	CC.1	SECTION	
CC.2	SECTION	10	CC.3	SECTION	11	CC.4	SECTION	
SYSLOC	SECTION	13						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
32	26A	00144	36	42A	00216	34	33A	
60	54A	00257						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00421.

MESHX STORAGE MAP

SUBROUTINE MESH

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ISTRT	00001	I	IEND	00002	I	FMINT	00003
FMONE	00004	R	FMTWO	00005	R	MINT1	00006
FM1	00007	R	FM2	00010	R	DELSC	00011
J	00012	I	COM	00013	R	I	00014
FI	00015	R	II	00016	I	KI	00017
JJ	00020	I	MONE1	00021	I		

ENTRY POINTS

MESH SECTION 3

SUBROUTINES CALLED

E.1	SECTION	4	E.2	SECTION	5	E.3	SECTION
E.4	SECTION	7	CC.1	SECTION	8	CC.2	SECTION
CC.3	SECTION	10	CC.4	SECTION	11	SYSLOC	SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
10	9A	00111	40	18A	00204	20	13A
30	17A	00157	70	19A	00205	90	34A
80	30A	00262	110	53A	00334	100	46A
130	66A	00412	120	62A	00405		

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00515.

EHMUX STORAGE MAP

SUBROUTINE EHMU

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
SLOPE	00001	R					

ENTRY POINTS

EHMU	SECTION	3

SUBROUTINES CALLED

.FWRD.	SECTION	4	EXIT	SECTION	5	.UN06.	SECTION
.FFIL.	SECTION	7	.FCNV.	SECTION	8	SYSLOC	SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
15	6A	00045	1000	FORMAT	00011	20	17A	
10	12A	00055	40	26A	00100	30	21A	

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00154.

SUBROUTINE DANK

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
012	00001	R	MU	00002	I	IE	00003
TH	00004	I	COM	00005	R	COM1	00006
AN	00007	R	AM	00010	R		

ENTRY POINTS

DANK SECTION 3

SUBROUTINES CALLED

SYSLOC SECTION 4

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
20	35A	00126	10	30A	00121	40	55A
30	50A	00160					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00274.

STORAGE MAP

SUBROUTINE NRPR

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
NPTS	00001	I	I	00002	I	J	00003
MPT	00004	I	IND	00005	I	IJK	00006
MS	00007	I	MF	00010	I		

ENTRY POINTS

NRPR SECTION 3

SUBROUTINES CALLED

SYSLOC SECTION 4

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
50	6A	00025	200	46A	00141	100	28A	
150	38A	00122	170	42A	00127			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00205.

```
$IBFTC MON3X   NOLIST,DECK,M94,X37,NODD
```

```
C                 MONITOR FOR REGION 3--SPECIFY HARMONICS                MON30010
      SUBROUTINE MON3                                                    MON30020
C     INTF(01)--FLAG INDICATING THE NO. OF M=0 SPECIAL ITERATIONS        MON30030
C     INTF(02)--FLAG INDICATING THE REMAINING NO. OF M=0 ITERATIONS      MON30040
C     INTF(03)--FLAG INDICATING WHETHER OR NOT M=0 FORCES  MAY BE USED   MON30050
C     INTF(04)--OVERLAY CONTROL FLAG FOR REGIONS 4 AND 5                 MON30060
      DIMENSION PS(600,7)                                                MON30070
      EQUIVALENCE(AU(1,1),PS(1,1))                                       MON30080
      EQUIVALENCE (INTF(1),INTF1),(INTF(2),INTF2),(INTF(3),INTF3)        MON30090
      COMMON/COM4A/ICHECK                                                MON30100
      COMMON/COMCC/CONFAC                                                MON30110
      COMMON/COM31/AC(8,8),AD(8,8)                                       MON30120
      COMMON/COM32/SINE,COSI,M                                           MON30130
      COMMON/COM33/IJ,MS,MF,IND,IJK,IJH,IDIAG                            MON30140
      COMMON/COM34/LOCP                                                  MON30150
      COMMON/COM36/GOTH(300),GOPH(300),GNPT(300),GNTH(300),SIFOR         MON30160
      COMMON NSGMTS, KS, ISGMT, IFTNO, LOADNO, LDNOPR, NLOAD, KKKKK      MON30170
      COMMON PHI(7),SINP(7),COSP(7),NPSEG(7),LSP(7),MLIST(7),ECR(7),     MON30180
     1 ECY(7),IOPT(14),IRES(10),NREC(21),INTF(10),NSGPL(10),NPTPT(10),   MON30190
     2 NLDPL(10)                                                         MON30200
      COMMON EI(2,7),HI(2,7),FMUI(2,7),RI(7)                             MON30210
      COMMON FLDI(5,7,7),FLDF(5,7,7),DLDI(5,7,7),DLDF(5,7,7)             MON30220
      COMMON FMPO(2,7,7),FNRO(2,7,7),FNYO(2,7,7),FNPTO(2,7,7),ECRS(7,7),MON30230
     1 ECYS(7,7),RSS(7,7),SCF(16,7,7)                                    MON30240
      COMMON IBC(4,7,2),BC(4,7,2),DLBC(4,7,2)                            MON30250
      COMMON S(300),FNTH(300),FNPH(300),OPHI(300),OTHE(300),STP(12,7)    MON30260
      COMMON AK(140,23),AU(140,23),AL(200,13),RHV(600),SAVE(5,10)        MON30270
      REWIND 14                                                          MON30280
C     SET FLAG FOR SPECIAL ITERATION FORCES                              MON30290    1
      LOOP = INTF3                                                       MON30300
C     ZERO FLAG FOR USE OF SPECIAL ITERATION FORCES - UPDATE             MON30310
      SIFOR = 0.0                                                        MON30320
      IJH=0                                                              MON30330
C     ARE THERE ANY SPECIAL ITERATIONS  FOR M=0                          MON30340
      IF(INTF1.EQ.0) GO TO 100                                           MON30350
C     ARE ALL OF THE SPECIAL ITERATIONS COMPLETED                        MON30360
      IF(INTF2.EQ.0) GO TO 100                                           MON30370
```

```
C     BUILD MATRIX FOR ZERO HARMONIC                                  MON30380
      IJH = 1                                                         MON30390
      M   = MLIST(1)                                                  MON30400
      IFTNO = M                                                       MON30410
      CALL HARM                                                       MON30420   13
      GO TO 500                                                       MON30430
  100 CONTINUE                                                        MON30440
C     BUILD MATRIX FOR ALL HARMONICS                                  MON30450
      IJH=IJH+1                                                       MON30460
      IF(IJH.GT.KS) GO TO 480                                         MON30470
      M = MLIST(IJH)                                                  MON30480
      IFTNO = M                                                       MON30490
C     ZERO FLAG FOR USE OF SPECIAL ITERATION FORCES - UPDATE          MON30500
      SIFOR = 0.0                                                     MON30510
      IF(IRES(4) .EQ. 0) GO TO 450                                    MON30520
C     TEST FOR FIRST HARMONIC...IF SO, SKIP S.I. FORCE GENERATION     MON30530
      IF(IJH .EQ. 1) GO TO 450                                        MON30540
C     SET SPECIAL ITERATION FORCES FOR THIS HARMONIC                  MON30550
C     DETERMINE INDEX OF HARMONIC = 2*M                               MON30560
      II = 0                                                          MON30570
      DO 150 I = IJH,KS                                               MON30580
      IF(MLIST(I).NE.2*MLIST(IJH)) GO TO 150                          MON30590
      II = I                                                          MON30600
C     SET FLAG FOR USE OF SPECIAL ITERATION FORCES - UPDATE           MON30610
      SIFOR = 1.0                                                     MON30620
  150 CONTINUE                                                        MON30630
      GM = FLOAT(MLIST(II))                                           MON30640
      REWIND 4                                                        MON30650   43
      DO 400 JJK = 1,NSGMTS                                           MON30660
      NPTS = NPSEG(JJK)                                               MON30670
      NROS = 8 * NPTS                                                 MON30680
C     SET SIN(PHI) AND COS(PHI) FOR THIS SEGMENT                      MON30690
      SINN = SINP(JJK)                                                MON30700
      COSS = COSP(JJK)                                                MON30710
      MS   = LSP(JJK)                                                 MON30720
      R1   = RI(JJK)                                                  MON30730
      S1   = S(MS)                                                    MON30740
C     SET SLOPES OF 'E', 'H', 'MU',                                   MON30750
```

```
      ESL  = (EI(2,JJK) - EI(1,JJK)) / S1                               MON30760
      HSL = (HI(2,JJK) - HI(1,JJK)) / S1                                MON30770
      FMU = (FMUI(2,JJK) - FMUI(1,JJK)) / S1                            MON30780
  200 CONTINUE                                                          MON30790
      READ(04) ((PS(I,J),I=1,NRBS),J=1,KS)                              MON30800
C     SET INDEX OF UNKNOWN - 1 FOR FIRST POINT IN SEGMENT OF SHELL      MON30810    62
      JJ =  MS - 1                                                      MON30820
      KK = - 7                                                          MON30830
      DO 300 I = 1,NPTS                                                 MON30840
      JJ  =  JJ + 1                                                     MON30850
      KK  =  KK + 8                                                     MON30860
C     SET N(PHI) FOR ZERO HARMONIC                                      MON30870
      FNPH(JJ) = PS(KK+5, 1) * CONFAC                                   MON30880
      IF(II) 300,300,220                                                MON30890
  220 CONTINUE                                                          MON30900
C     ADD N(PHI) FOR ''2M''                                             MON30910
      FNPH(JJ) = FNPH(JJ) + .5 * PS(KK+5,II)                            MON30920
C     GET OMEGA(THETA) AND NBAR(PHI-THETA) FOR ''2M''                   MON30930
      GOTH(JJ) = PS(KK+3,II)                                            MON30940
      GNPT(JJ) = PS(KK+4,II)                                            MON30950
C     CALCULATE PHYSICAL CHARACTERISTICS FOR POINT                      MON30960
      EE  = ESL * S(JJ) + EI(1,JJK)                                     MON30970
      HH  = HSL * S(JJ) + HI(1,JJK)                                     MON30980
      FF  = FMU * S(JJ) + FMUI(1,JJK)                                   MON30990
      RRI = 1. / (S(JJ)*COSS + R1)                                      MON31000
      OSS = 1.0 - FF * FF                                               MON31010
      OSI = 1.0 / OSS                                                   MON31020
      DD  = (EE * HH) * OSI                                             MON31030
C     CALCULATE OMEGA(PHI)                                              MON31040
      GOPH(JJ) = RRI * (GM *PS(KK+2,II) + SINN * PS(KK,II) )            MON31050
C     CALCULATE N(THETA) ASSUMING LAMBDA(3) IS SMALL                    MON31060
      GNTH(JJ) = FF*PS(KK+5,II) + DD*OSS *                              MON31070
     1 ( RRI*(SINN*PS(KK+2,II) + GM*PS(KK,II) + COSS*PS(KK+1,II) ) )    MON31080
  300 CONTINUE                                                          MON31090
  400 CONTINUE                                                          MON31100
      IF(ICHECK .EQ. 0) GO TO 450                                       MON31110
      WRITE(6,2000) MLIST(II)                                           MON31120    109
 2000 FORMAT(1H1,11HUPDATE 2M =,I3//5X,5HPOINT,6X,4HGOTH,11X,4HGOPH,    MON31130
```

```
    1 11X,4HGNPT,11X,4HGNTH,11X,4HFNPH,11X,4HOPHI//)                MON31140
      DO 425 I=1,JJ                                                 MON31150
  425 WRITE(6,2050) I,GOTH(I),GOPH(I),GNPT(I),GNTH(I),FNPH(I),OPHI(I)  MON31160   114
 2050 FORMAT(7X,I3,1P6E15.7)                                        MON31170
  450 CONTINUE                                                      MON31180
      CALL HARM                                                     MON31190   124
      GO TO 100                                                     MON31200
  480 CONTINUE                                                      MON31210
C     SET O. I. FLAG FOR ZEROTH ORDINARY ITERATION                  MON31220
      INTF(6) = 0                                                   MON31230
C     ZERO OUT 'UPDATE' FLAG                                        MON31240
      IRES(4) = 0                                                   MON31250
      REWIND 4                                                      MON31260   127
  500 CONTINUE                                                      MON31270
      REWIND 14                                                     MON31280   128
      KKKKK=4                                                       MON31290
      INTF(04) = 1                                                  MON31300
      RETURN                                                        MON31310
      END                                                           MON31320
```

```
$IBFTC HARMX   NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD K MATRIX FOR ONE HARMONIC                       HARM0000
      SUBROUTINE HARM                                                   HARM0010
      REAL K1,K2                                                        HARM0020
      EQUIVALENCE (RHM(1),AL(1,1)),(RHK(1),AK(1,1))                     HARM0025
      DIMENSION FMAT(2,6),RHM(2400),RHK(2400)                           HARM0030
      COMMON/COM31/AC(8,8),AD(8,8)                                      HARM0040
      COMMON/COM32/SINE,COSI,M                                          HARM0050
      COMMON/COM33/IJ,MS,MF,IND,IJK,IJH,IDIAG                           HARM0060
      COMMON/COM35/ R1,D1,K1,FDC1,FM1,R2,D2,K2,FDC2,FM2                 HARM0070
      COMMON NSGMTS, KS, ISGMT, IFTNO, LOADNO, LDNOPR, NLOAD, KKKKK     HARM0080
      COMMON PHI(7),SINP(7),COSP(7),NPSEG(7),LSP(7),MLIST(7),ECR(7),    HARM0090
     1 ECY(7),IOPT(14),IRES(10),NREC(21),INTF(10),NSGPL(10),NPTPT(10),  HARM0100
     2 NLDPL(10)                                                        HARM0105
      COMMON EI(2,7),HI(2,7),FMUI(2,7),RI(7)                            HARM0110
      COMMON FLDI(5,7,7),FLDF(5,7,7),DLDI(5,7,7),DLDF(5,7,7)            HARM0120
      COMMON FMPO(2,7,7),FNRO(2,7,7),FNYO(2,7,7),FNPTO(2,7,7),ECRS(7,7),HARM0130
     1 ECYS(7,7),RSS(7,7),SCF(16,7,7)                                   HARM0140
      COMMON IBC(4,7,2),BC(4,7,2),DLBC(4,7,2)                           HARM0150
      COMMON S(300),FNTH(300),FNPH(300),DPHI(300),DTHE(300),STP(12,7)   HARM0160
      COMMON AK(140,23),AU(140,23),AL(200,13),RHV(600),SAVE(5,10)       HARM0180
      DATA FMAT(1,1),FMAT(1,2),FMAT(1,3),FMAT(1,4),FMAT(1,5),FMAT(1,6)/ HARM0190
     1 11HLEADING B C,12HTRAILING B C,11HLEADING D E,12HTRAILING D E,   HARM0196
     2 11HEQUIL. EQN.,10HR.H.VECTOR/                                    HARM0204
 2000 FORMAT(84HSEGMENT,I3/1X,2A6/)                                     HARM0210
 2001 FORMAT(2H0 , 1P6E16.7/2X,6E16.7/2X,6E16.7/2X,5E16.7)              HARM0220
 2002 FORMAT(1H0 ,2A6,5HPOINT,I3/)                                      HARM0230
 2003 FORMAT(1H0,14X,5HPOINT,I3)                                        HARM0240
 2004 FORMAT(13H1 HARMONIC = ,I3)                                       HARM0250
 2005 FORMAT(5X,1PE13.5)                                                HARM0260
      IDIAG  =  12                                                      HARM0270
      MAXSEG =   7                                                      HARM0280
      MAXHRM =   7                                                      HARM0290
      IBNWT  =  23                                                      HARM0300
      NROWS  = 140                                                      HARM0310
      MAXPTS = 300                                                      HARM0320
      MAXMPT =  17                                                      HARM0330
      IJK    =   1                                                      HARM0340
```

```
      MPT     =    1                                                  HARM0350
      NRHV    =    0                                                  HARM0355
      IND     =    0                                                  HARM0360
      IRE     =    0                                                  HARM0365
      IRHVFL  =    0                                                  HARM0370
      IJL     =    1                                                  HARM0380
      MS=LSP(IJK)                                                     HARM0390
C     SET FLAG DETERMINING WHETHER OR NOT A RHV IS GENERATED          HARM0392
      NRFLAG=0                                                        HARM0394
      IF((INTF(1).GT.0).AND.(INTF(1).EQ.INTF(2))) NRFLAG=1            HARM0396
      IF(IOPT(4).NE.1) GO TO 20                                       HARM0400
      WRITE(6,2004) M                                                 HARM0410    23
   20 CONTINUE                                                        HARM0420
      KK=KK+1                                                         HARM0430
      IJ=IJL                                                          HARM0440
      IF(IJL.NE.MS) GO TO 30                                          HARM0450
C     INITIALIZE FLAGS FOR START OF NEW SEGMENT                       HARM0460
      ISGMT=IJK                                                       HARM0470
      KK   = 1                                                        HARM0480
      MF=MS+NPSEG(IJK)-1                                              HARM0490
      MF1=MF-1                                                        HARM0500
      SINE=SINP(IJK)                                                  HARM0510
      COSI=COSP(IJK)                                                  HARM0520
   30 CONTINUE                                                        HARM0530
C     COMPUTE COM35 VALUES                                            HARM0540
      CALL MISC(EI,HI,FMUI,RI,MAXPTS,S,FLDI,FLDF)                     HARM0550    38
      IF(IJL.NE.MS)  GO TO 60                                         HARM0560
      IF(IJK.NE.1) GO TO 50                                           HARM0570
C     COMPUTE LEADING  EDGE BOUNDARY CONDITIONS                       HARM0580
      IBCF=1                                                          HARM0590
      NPFLG=1                                                         HARM0600
      MPV=1                                                           HARM0610
      GO TO 140                                                       HARM0620
   50 CONTINUE                                                        HARM0630
C     COMPUTE STARTING DISCONTINUITY EQUATIONS                        HARM0640
      MPQ=1                                                           HARM0650
      NPFLG=3                                                         HARM0660
      MPV=2                                                           HARM0670
```

```
      GO TO 145                                                         HARM0680
   60 CONTINUE                                                          HARM0690
      IF(IJL.NE.MF)        GO TO 90                                     HARM0700
      IF(ISGMT.NE.NSGMTS) GO TO  80                                     HARM0710
C     COMPUTE TRAILING EDGE BOUNDARY CONDITIONS                         HARM0720
      IBCF=2                                                            HARM0730
      NPFLG=2                                                           HARM0740
      IRHVFL=2                                                          HARM0750
      MPV=1                                                             HARM0760
      GO TO 140                                                         HARM0770
   80 CONTINUE                                                          HARM0780
C     COMPUTE ENDING DISCONTINUITY EQUATIONS                            HARM0790
      MPQ=2                                                             HARM0800
      IRHVFL=1                                                          HARM0810
      NPFLG=4                                                           HARM0820
      MPV=2                                                             HARM0830
      GO TO 145                                                         HARM0840
  140 CONTINUE                                                          HARM0850
      CALL GAMM(IBCF,IBC,AK,IBNWT,NROWS)                                HARM0860   70
      GO TO 150                                                         HARM0870
  145 CONTINUE                                                          HARM0880
      CALL BETA(MPQ,AK,ECRS,ECYS,ECR,ECY,SCF,RI,MAXSEG,MAXHRM,          HARM0890
     1 SINP,COSP,IBNWT,NROWS,RSS)                                       HARM0900   73
  150 CONTINUE                                                          HARM0910
      IF(NRFLAG.EQ.0) GO TO 155                                         HARM0920
C     CALCULATE RIGHT HAND VECTOR                                       HARM0930
      CALL BRHV(RHV,BC,FMPO,FNRO,FNYO,FNPTO,FLDI,FLDF,MAXSEG,MAXHRM,MPV,HARM0940
     1 S,MAXPTS)                                                        HARM0950   78
  155 CONTINUE                                                          HARM0960
      IF(IOPT(4).EQ.0) GO TO 190                                        HARM0970
C     PRINT BOUNDARY CONDITIONS OR DISCONTINUITY EQUATIONS              HARM0980
      WRITE(6,2000) IJK,FMAT(1,NPFLG),FMAT(2,NPFLG)                     HARM0990   82
      DO 160 I=1,4                                                      HARM1000
      IZ=IND-4+I                                                        HARM1010
      WRITE(6,2001) (AK(IZ,J),J=1,IBNWT)                                HARM1020   88
  160 CONTINUE                                                          HARM1030
      IF(NRFLAG.EQ.0) GO TO 190                                         HARM1040
      NPS1=6                                                            HARM1050
```

```
      WRITE(6,2002) FMAT(1,NPGL),FMAT(2,NPGL),KK                        HARM1060   99
      IN=4                                                              HARM1070
      DO 185 I=1,IN                                                     HARM1080
      IZ=IND-IN+I                                                       HARM1090
  185 WRITE(6,2005) RHV(IZ)                                             HARM1100   106
  190 GO TO (100,200,100,200), NPFLG                                    HARM1110
   90 CONTINUE                                                          HARM1120
  100 CONTINUE                                                          HARM1130
      MPV=3                                                             HARM1140
      NPFLG=5                                                           HARM1150
C     COMPUTE 'A' COEFFICIENTS                                          HARM1160
      CALL ACAD(S,FNTH,FNPH,DPHI,DTHE,MAXPTS)                           HARM1170
C     COMPUTE EQUILIBRIUM EQUATIONS                                     HARM1180   115
      CALL EQUM(AK,NROWS,IBNWT)                                         HARM1190   117
      IF(NRFLAG.EQ.0) GO TO 162                                         HARM1200
C     CALCULATE RIGHT HAND VECTOR                                       HARM1210
      CALL BRHV(RHV,BC,FMPD,FNRD,FNYD,FNPTD,FLDI,FLDF,MAXSEG,MAXHRM,MPV,HARM1220
     1 S,MAXPTS)                                                        HARM1230   122
  162 IF(IOPT(4).EQ.0) GO TO 200                                        HARM1240
C     PRINT EQUILIBRIUM EQUATIONS                                       HARM1250
      IF(IJL.NE.MS) GO TO 165                                           HARM1260
      WRITE(6,2002) FMAT(1,NPFLG),FMAT(2,NPFLG),KK                      HARM1270   128
      GO TO 166                                                         HARM1280
  165 WRITE(6,2003) KK                                                  HARM1290   132
  166 CONTINUE                                                          HARM1300
      DO 170 I=1,8                                                      HARM1310
      IZ=IND-8+I                                                        HARM1320
      WRITE(6,2001) (AK(IZ,J),J=1,IBNWT)                                HARM1330   137
  170 CONTINUE                                                          HARM1340
      IF(NRFLAG.EQ.0) GO TO 200                                         HARM1350
      NPFLG=6                                                           HARM1360
      WRITE(6,2002) FMAT(1,NPFLG),FMAT(2,NPFLG),KK                      HARM1370   148
      IN=8                                                              HARM1380
      DO 180 I=1,IN                                                     HARM1390
      IZ=IND-IN+I                                                       HARM1400
  180 WRITE(6,2005) RHV(IZ)                                             HARM1410   155
  200 CONTINUE                                                          HARM1420
      IF(IRHVFL-1) 300,290,310                                          HARM1430
```

```
  290 CONTINUE                                                      HARM1440
C     INITIALIZE FLAGS FOR START OF NEW SEGMENT                     HARM1450
      IJK=IJK+1                                                     HARM1460
      IRHVFL=0                                                      HARM1470
      MS=LSP(IJK)                                                   HARM1480
  300 IF(MPT.LT.MAXMPT) GO TO 500                                   HARM1490
  310 CONTINUE                                                      HARM1500
C     WRITE 'K' MATRIX ON TAPE                                      HARM1510
      WRITE(14) ((AK(I,J),I=1,IND),J=1,IBNWT)                       HARM1520    168
      IRE=IRE+1                                                     HARM1524
      NREC(IRE)=IND                                                 HARM1526
      IF(NRFLAG.EQ.0) GO TO 320                                     HARM1530
C     SAVE RHV IN CORE                                              HARM1540
      DO 315 I=1,IND                                                HARM1544
      NRHV = NRHV+1                                                 HARM1548
      RHM(NRHV) = RHV(I)                                            HARM1552
  315 CONTINUE                                                      HARM1556
  320 CONTINUE                                                      HARM1560
C     INITIALIZE COUNTERS FOR NEW RECORD                            HARM1570
      MPT=0                                                         HARM1575
      IND=0                                                         HARM1580
      IF(IRHVFL.EQ.2) GO TO 1000                                    HARM1590
C     INCREMENT POINT NUMBER                                        HARM1600
  500 MPT=MPT+1                                                     HARM1610
      IJL=IJL+1                                                     HARM1620
      GO TO 20                                                      HARM1630
 1000 CONTINUE                                                      HARM1640
      RETURN                                                        HARM1650
      END                                                           HARM1660
```

```
$IBFTC MISCX    NOLIST,DECK,M94,XR7,NODD
```

```
C                 CALCULATE COM35 VALUES FOR POINT 'I' AND 'I+1'          MISC0000
      SUBROUTINE MISC( EI,HI,FMUI,RI,MAXPTS,S,FLDI,FLDF)                  MISC0010
      DIMENSION S(MAXPTS),EI(2,7),HI(2,7),FMUI(2,7),RI(7)                 MISC0020
      DIMENSION FLDI(5,7,7),FLDF(5,7,7)                                   MISC0025
      REAL K1,K2                                                          MISC0030
      COMMON/COM32/SINE,COSI,M                                            MISC0040
      COMMON/COM33/IJ,MS,MF,IND,IJK,IJH,IDIAG                             MISC0050
      COMMON/COM35/ R1,D1,K1,FDC1,FM1,R2,D2,K2,FDC2,FM2                   MISC0060
      COMMON/COM37/ZBR(2)                                                 MISC0065
      O12=8.3333333E-2                                                    MISC0070
      IZ = IJ                                                             MISC0080
      IF(IZ.NE.MS) GO TO 100                                              MISC0090
      DEL = ABS(S(MF)-S(MS))                                              MISC0100
      SLOPEE =(EI(2,IJK)-EI(1,IJK))/DEL                                   MISC0110
      SLOPEH = (HI(2,IJK)-HI(1,IJK))/DEL                                  MISC0120
      SLOPEU = (FMUI(2,IJK)-FMUI(1,IJK))/DEL                              MISC0130
      SLOPZB = (FLDF(3,IJK,1) - FLDI(3,IJK,1))/DEL                        MISC0135
      R1 = S(IZ)*COSI+RI(IJK)                                             MISC0140
      FM1 = SLOPEU*S(IZ)+FMUI(1,IJK)                                      MISC0150
      H1  = SLOPEH*S(IZ)+HI(1,IJK)                                        MISC0160
      E1  = SLOPEE*S(IZ)+EI(1,IJK)                                        MISC0170
      ZBR(1) = SLOPZB*S(IZ) + FLDI(3,IJK,1)                               MISC0175
      D1  = E1*H1/(1.-FM1*FM1)                                            MISC0180
      K1  = .012*D1*H1*H1                                                 MISC0190
      IK = IZ+1                                                           MISC0200
      FDC1 = 1./(S(IK)-S(IZ))                                             MISC0210
      GO TO 150                                                           MISC0220
  100 CONTINUE                                                            MISC0230
      E1 = E2                                                             MISC0240
      H1 = H2                                                             MISC0250
      FM1= FM2                                                            MISC0260
      ZBR(1) = ZBR(2)                                                     MISC0265
      R1 = R2                                                             MISC0270
      D1 = D2                                                             MISC0280
      K1 = K2                                                             MISC0290
      FDC1=FDC2                                                           MISC0300
      IF(IZ.EQ.MF) GO TO 200                                              MISC0310
```

```
150 CONTINUE                                                    MISC0320
    IZ = IZ+1                                                   MISC0330
    E2  = SLOPEE*S(IZ)+EI(1,IJK)                                MISC0340
    H2  = SLOPEH*S(IZ)+HI(1,IJK)                                MISC0350
    FM2 = SLOPEU*S(IZ)+FMUI(1,IJK)                              MISC0360
    ZBR(2) = SLOPZB*S(IZ) + FLDI(3,IJK,1)                       MISC0365
    R2 = S(IZ)*COSI+RI(IJK)                                     MISC0370
    D2  = E2*H2/(1.-FM2*FM2)                                    MISC0380
    K2  =  D12*D2*H2*H2                                         MISC0390
    IK = IZ+1                                                   MISC0400
    FDC2 = 1./(S(IK)-S(IZ))                                     MISC0410
200 CONTINUE                                                    MISC0420
    RETURN                                                      MISC0430
    END                                                         MISC0440
```

```
$IBFTC ACADX   NOLIST,DECK,M94,XR7,NODD
```

```
C                 'A' COEFFICIENTS FOR 'I' AND 'I+1'                    ACAD0000
      SUBROUTINE ACAD(S,FNTH,FNPH,OPHI,OTHE,MAXPTS)                     ACAD0010
C     AC --'A' COEFFICIENTS FOR POINT 'I'                               ACAD0020
C     AD --'A' COEFFICIENTS FOR POINT 'I+1'                             ACAD0030
      DIMENSION S(01)                                                   ACAD0040
      COMMON/COM31/AC(8,8),AD(8,8)                                      ACAD0050
      COMMON/COM32/SINE,COSI,M                                          ACAD0060
      COMMON/COM33/IJ,MS,MF,IND,IJK,IJH,IDIAG                           ACAD0070
      COMMON/COM34/LOOP                                                 ACAD0080
      COMMON/COM35/ R1,D1,K1,FDC1,FM1,R2,D2,K2,FDC2,FM2                 ACAD0090
      REAL K1,K2                                                        ACAD0100
      IF(IJ.NE.MS) GO TO 120                                            ACAD0110
C     'A' COEFFICIENTS FOR POINT 'I'                                    ACAD0120
      CALL ACOF(AC,R1,D1,K1,FM1,MAXPTS)                                 ACAD0130
      IF(LOOP.EQ.0) GO TO 140                                           ACAD0140
C     MODIFIED 'A' COEFFICIENTS                                         ACAD0160
      CALL MACD(AC,R1,D1,K1,FM1,FNTH,FNPH,OPHI,OTHE,IJ)                 ACAD0170
      GO TO 140                                                         ACAD0180
  120 CONTINUE                                                          ACAD0190
      DO 130 JJ=1,8                                                     ACAD0200
      DO 130 KK=1,8                                                     ACAD0210
  130 AC(JJ,KK)=AD(JJ,KK)                                               ACAD0220
  140 CONTINUE                                                          ACAD0230
C     'A' COEFFICIENTS FOR POINT 'I+1'                                  ACAD0240
      CALL ACOF(AD,R2,D2,K2,FM2,MAXPTS)                                 ACAD0250
      IF(LOOP.EQ.0) GO TO 150                                           ACAD0260
C     MODIFIED 'A' COEFFICIENTS                                         ACAD0280
      IZ = IJ + 1                                                       ACAD0285
      CALL MACD(AD,R2,D2,K2,FM2,FNTH,FNPH,OPHI,OTHE,IZ)                 ACAD0290
  150 CONTINUE                                                          ACAD0300
      RETURN                                                            ACAD0310
      END                                                               ACAD0320
```

```
$IBFTC ACJFX   NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD 'A' COEFFICIENTS FOR GIVEN POINT                ACOF0000
      SUBROUTINE ACOF(ZZ,R1,D1,K1,FM1,MAXPTS)                           ACOF0010
      REAL K1,LNI,M1,MSQ                                                ACOF0020
      DIMENSION ZZ(8,8)                                                 ACOF0030
      COMMON/COM32/SINE,COSI,M                                          ACOF0040
      R1I=1./R1                                                         ACOF0050
      R2 =R1*R1                                                         ACOF0060
      R2I=1./R2                                                         ACOF0070
      R3 =R1**3                                                         ACOF0080
      R3I=1./R1**3                                                      ACOF0090
      OMU =1.-FM1                                                       ACOF0100
      OMUS=1.-FM1*FM1                                                   ACOF0110
      SISQ=SINE*SINE                                                    ACOF0120
      COSQ=COSI*COSI                                                    ACOF0130
      SICO=COSI*SINE                                                    ACOF0140
      M1    = M                                                         ACOF0150
      MSQ   = M1*M1                                                     ACOF0160
      LNI=1./(1.+2.25*K1*SISQ*R2I/D1)                                   ACOF0170
      T1 = K1*OMU*R2I*LNI                                               ACOF0180
      ZZ(1,1) =-MSQ*OMUS*R2I*(D1+K1*SISQ*R2I)                           ACOF0190
      ZZ(1,2)=-M1*OMUS*R2I*D1*COSI                                      ACOF0200
      ZZ(1,3)=-M1*OMUS*SINE*R2I*(D1+MSQ*K1*R2I)                         ACOF0210
      ZZ(1,4) =-M1*OMUS*R3I*K1*SICO                                     ACOF0220
      ZZ(1,5)   = 2.*COSI*R1I                                           ACOF0230
      ZZ(1,6)   =-M1*FM1*R1I                                            ACOF0240
      ZZ(1,7)   = 0.                                                    ACOF0250
      ZZ(1,8)   =-M1*FM1*SINE*R2I                                       ACOF0260
      ZZ(2,1)   =-M1*D1*OMUS *COSI*R2I                                  ACOF0270
      ZZ(2,2)   =-R2I*(D1*OMJS*COSQ+2.*MSQ*OMU*SISQ*LNI*R2I*K1)         ACOF0280
      ZZ(2,3)   =-SICO*R2I*(D1*OMJS-2.*MSQ*K1*OMU*LNI*R2I)              ACOF0290
      ZZ(2,4)   = 2.*MSQ*OMJ*SINE*R3I*LNI*K1                            ACOF0300
      ZZ(2,5)   = M1*R1I*(1.-3.*K1*SISQ*LNI*R2I/D1)                     ACOF0310
      ZZ(2,6)   = OMU*COSI*R1I                                          ACOF0320
      ZZ(2,7)   = 0.                                                    ACOF0330
      ZZ(2,8)   = 0.                                                    ACOF0340
      ZZ(3,1)   =-M1*SINE*R2I*(D1*OMUS+MSQ*K1*OMUS*R2I)                 ACOF0350
      ZZ(3,2)   =-SICO*R2I*(D1*OMJS-2.*MSQ*K1*OMU*LNI*R2I)              ACOF0360
```

```
      ZZ(3,3)  =-R2I*(D1*OMJS*SISQ+2.*MSQ*T1  *COSQ+K1*OMUS*R2I*M1**4)   ACOF0370
      ZZ(3,4)  =-MSQ*K1*COSI*R3I*(2.*OMU*LNI+OMUS)                       ACOF0380
      ZZ(3,5)  = 3.*M1*K1*SICO*LNI*R3I/D1                                ACOF0390
      ZZ(3,6)  =-FM1*SINE*R1I                                            ACOF0400
      ZZ(3,7)  = COSI*R1I                                                ACOF0410
      ZZ(3,8)  =-MSQ*FM1*R2I                                             ACOF0420
      ZZ(4,1) =-M1*K1*OMUS*SICO*R3I                                      ACOF0430
      ZZ(4,2) = MSQ*K1*OMU*SINE*R3I*LNI*2.0                              ACOF0440
      ZZ(4,3) =-MSQ*K1*COSI*R3I*(OMUS+2.*OMU*LNI)                        ACOF0450
      ZZ(4,4) =-K1*R2I*(OMJS*COSQ+2.*MSQ*OMU*LNI)                        ACOF0460
      ZZ(4,5) = 3.*M1*K1*SINE*LNI*R2I/D1                                 ACOF0470
      ZZ(4,6) = 0.                                                       ACOF0480
      ZZ(4,7) =-1.                                                       ACOF0490
      ZZ(4,8) = COSI*OMU*R1I                                             ACOF0500
      ZZ(5,1) =-COSI*R1I                                                 ACOF0510
      ZZ(5,2) =-M1*LNI*R1I*(1.-.75*K1*SISQ*R2I/D1)                       ACOF0520
      ZZ(5,3) =-M1*LNI* 3.*K1*SICO*R3I/D1                                ACOF0530
      ZZ(5,4) =-M1*LNI* 3.*K1*SINE*R2I/D1                                ACOF0540
      ZZ(5,5) =-2.*LNI/(D1*OMU)                                          ACOF0550
      DO 50 II=6,8                                                       ACOF0560
   50 ZZ(5,II)= 0.                                                       ACOF0570
      ZZ(6,1) = FM1*M1*R1I                                               ACOF0580
      ZZ(6,2) = FM1*COSI*R1I                                             ACOF0590
      ZZ(6,3) = FM1*SINE*R1I                                             ACOF0600
      DO 60 II= 4,8                                                      ACOF0610
   60 ZZ(6,II)= 0.                                                       ACOF0620
      ZZ(6,6) =-1./D1                                                    ACOF0630
      DO 70 II= 1,8                                                      ACOF0640
   70 ZZ(7,II)= 0.                                                       ACOF0650
      ZZ(7,4) = 1.                                                       ACOF0660
      ZZ(8,1) = M1*FM1*SINE*R2I                                          ACOF0670
      ZZ(8,2) = 0.                                                       ACOF0680
      ZZ(8,3) = MSQ*FM1*R2I                                              ACOF0690
      ZZ(8,4) = FM1*COSI*R1I                                             ACOF0700
      DO 80 II= 5,7                                                      ACOF0710
   80 ZZ(8,II)= 0.                                                       ACOF0720
      ZZ(8,8) =-1./K1                                                    ACOF0730
 1000 RETURN                                                             ACOF0740
```

```
END                                                    ACOF0750
```

```
C                BUILD MODIFIED 'A' COEFFICIENTS FOR GIVEN POINT        MAC00010
      SUBROUTINE MACO(ZZ,R1,D1,K1,FM1,FNTH,FNPH,OPHI,OTHE,IK)           MAC00020
      DIMENSION ZZ(8,8),FNTH(1),FNPH( 1),OPHI( 1),OTHE( 1)              MAC00030
      COMMON/COM32/SINE,COSI,M                                          MAC00040
      COMMON/COM33/IJ,MS,MF,IND,IJK,IJH,IDIAG                           MAC00050
      COMMON/COM36/GOTH(300),GOPH(300),GNPT(300),GNTH(300),SIFOR        MAC00060
      COMMON/COM37/ZBR(2)                                               MAC00070
      REAL K1,M1,MSQ,LNI                                                MAC00080
      MZB = 2                                                           MAC00090
      IF(IK.EQ.MS) MZB = 1                                              MAC00100
      FNT= FNTH(IK)                                                     MAC00110
      FNP= FNPH(IK)                                                     MAC00120
      OT = OTHE(IK)                                                     MAC00130
      OP = OPHI(IK)                                                     MAC00140
      FM = FM1                                                          MAC00150
      M1 = M                                                            MAC00160
      MSQ= M1*M1                                                        MAC00170
      OMU =1.-FM                                                        MAC00180
      OMUS=1.-FM*FM                                                     MAC00190
      R1I = 1./R1                                                       MAC00200
      R2  = R1*R1                                                       MAC00210
      R2I = 1./R2                                                       MAC00220
      R3  = R1**3                                                       MAC00230
      R3I = 1./R3                                                       MAC00240
      D1I = 1./D1                                                       MAC00250
      SISQ= SINE*SINE                                                   MAC00260
      COSQ= COSI*COSI                                                   MAC00270
      SICO= SINE*COSI                                                   MAC00280
      LNI = 1.0/(1.+2.25*K1*SISQ*R2I*D1I)                               MAC00290
      ZZ(1,1) = ZZ(1,1)-SISQ*R2I*(FNT+D1*OMUS*OP)+ZBR(MZB)*SINE*R1I     MAC00300
      ZZ(1,3) = ZZ(1,3)-M1*SINE*R2I*(FNT+D1*OMUS*OP)+ZBR(MZB)*M1*R1I    MAC00310
      ZZ(1,5) = ZZ(1,5)-SINE*R1I*OT                                     MAC00320
      T1=1.5*M1*K1*OMU*SINE*R3I*LNI*OT                                  MAC00330
      ZZ(2,1) = ZZ(2,1)+R1I*SISQ*T1                                     MAC00340
      ZZ(2,3) = ZZ(2,3)+R1I*M1*SINE*T1                                  MAC00350
      ZZ(2,4) = ZZ(2,4) + ZBR(MZB)                                      MAC00360
      T2= D1*OMUS*OP+1.5*K1*OMU*SICO*OT*LNI*R2I+FNT                     MAC00370
```

```
      ZZ(3,1) = ZZ(3,1)-M1*SINE*R2I*T2                                  MAC00380
      ZZ(3,3) = ZZ(3,3)-MSQ*R2I*T2                                      MAC00390
      ZZ(3,5) = ZZ(3,5)-M1*R1I*OT                                       MAC00400
      ZZ(4,1) = ZZ(4,1)-T1*SINE                                         MAC00410
      ZZ(4,3) = ZZ(4,3)-T1*M1                                           MAC00420
      ZZ(4,4) = ZZ(4,4)-FNP                                             MAC00430
      ZZ(4,6) = ZZ(4,6)-OT                                              MAC00440
      ZZ(5,1) = ZZ(5,1)+SINE*R1I*LNI*OT                                 MAC00450
      ZZ(5,3) = ZZ(5,3)+M1*R1I*LNI*OT                                   MAC00460
      ZZ(6,4) = ZZ(6,4)+OT                                              MAC00470
C     SKIP 'UPDATE' SPECIAL ITERATION FORCES ADDITION                   MAC00480
      IF(SIFOR.EQ.0.0) GO TO 500                                        MAC00490
      HNT  = GNTH(IK)                                                   MAC00500
      HNPT = GNPT(IK)                                                   MAC00510
      HOT  = GOTH(IK)                                                   MAC00520
      HOP  = GOPH(IK)                                                   MAC00530
C     ADD ''UPDATE'' MODIFIED 'A' COEFFICIENT TERMS                     MAC00540
      ZZ(1,1) = ZZ(1,1) + .5*SISQ*R2I*HNT - SINE*R2I*D1*OMUS*M1*HOP     MAC00550
      ZZ(1,2) = ZZ(1,2) - .5*SICO*R2I*D1*OMUS*HOP                       MAC00560
      ZZ(1,3) = ZZ(1,3) + .5*M1*SINE*R2I*HNT                            MAC00570
     1                  - .5*D1*OMUS*R2I*(SISQ + MSQ)*HOP               MAC00580
      ZZ(1,4) = ZZ(1,4) - .5*SINE*R1I*HNPT                              MAC00590
      ZZ(1,5) = ZZ(1,5) + .5*SINE*R1I*HOT                               MAC00600
      ZZ(1,6) = ZZ(1,6) - .5*SINE*R1I*FM*HOP                            MAC00610
      TT1     = 1.5*M1*K1*OMU*SINE*R3I*LNI                              MAC00620
      ZZ(2,1) = ZZ(2,1) - .5*SISQ*R1I*TT1*HOT - .5*D1*OMUS*R2I*SICO*HOP MAC00630
      ZZ(2,3) = ZZ(2,3) - .5*M1*SINE*R1I*TT1*HOT                        MAC00640
     1                  - .5*D1*OMUS*R2I*COSI*M1*HOP                    MAC00650
      ZZ(2,4) = ZZ(2,4) + .5*SINE*TT1*HOP                               MAC00660
      TT2     = .75*K1*OMU*R2I*LNI*SICO*HOT                             MAC00670
      TT3     =  .5*D1*OMUS*R2I*(SISQ + MSQ)*HOP                        MAC00680
      ZZ(3,1) = ZZ(3,1) + M1*SINE*R2I*(TT2     + .5*HNT) - TT3          MAC00690
      ZZ(3,2) = ZZ(3,2) - .5*D1*OMUS*R2I*M1*COSI*HOP                    MAC00700
      ZZ(3,3) = ZZ(3,3) + MSQ*R2I*(.5*HNT + TT2) - TT3                  MAC00710
      ZZ(3,4) = ZZ(3,4) - .5*M1*R1I*HNPT-.75*M1*K1*OMU*LNI*R3I*SICO*HOP MAC00720
      ZZ(3,5) = ZZ(3,5) + .5*M1*R1I*HOT                                 MAC00730
      ZZ(3,6) = ZZ(3,6) - .5*M1*R1I*FM*HOP                              MAC00740
      TT4     = .75*M1*K1*OMU*R2I*LNI*SINE                              MAC00750
```

```
      ZZ(4,1) = ZZ(4,1) + SINE*R1I*TT4*HOT - .5*SINE*R1I*HNPT          MAC00760
      ZZ(4,3) = ZZ(4,3) + M1*R1I*TT4*HOT - .5*M1*R1I*HNPT              MAC00770
      ZZ(4,4) = ZZ(4,4) - TT4*HOP                                      MAC00780
      ZZ(4,5) = ZZ(4,5) - .5*HOP                                       MAC00790
      ZZ(4,6) = ZZ(4,6) - .5*HOT                                       MAC00800
      ZZ(5,1) = ZZ(5,1) - .5*SINE*R1I*LNI*HOT                          MAC00810
      ZZ(5,3) = ZZ(5,3) - .5*M1*R1I*LNI*HOT                            MAC00820
      ZZ(5,4) = ZZ(5,4) + .5*LNI*HOP                                   MAC00830
      ZZ(6,4) = ZZ(6,4) + .5*HOT                                       MAC00840
  500 CONTINUE                                                         MAC00850
      RETURN                                                           MAC00860
      END                                                              MAC00870
```

```
$IBFTC GAMMX    VOLIST,DECK,M94,X37,NODD
```

```
C                 BUILD BOUNDARY CONDITION EQUATIONS                    GAMM0010
      SUBROUTINE GAMM(IBCF,IBC,AK,IBNWT,NROWS)                          GAMM0020
C     IBCF = 1---LEADING  BOUNDARY CONDITIONS                           GAMM0030
C     IBCF = 2---TRAILING BOUNDARY CONDITIONS                           GAMM0040
      DIMENSION IBC(4,7,2),AK(NROWS,IBNWT)                              GAMM0050
      REAL K1,K2                                                        GAMM0060
      COMMON/COM32/SINE,COSI,M                                          GAMM0070
      COMMON/COM33/IJ,MS,MF,IND,IJK,IJH,IDIAG                           GAMM0080
      COMMON/COM35/ R1,D1,K1,FDC1,FM1,R2,D2,K2,FDC2,FM2                 GAMM0090
      PI = 3.14159265                                                   GAMM0100
      ICON=IDIAG                                                        GAMM0110
      IF(IBCF.EQ.2) ICON=ICON-4                                         GAMM0120
      IN=IND+1                                                          GAMM0130
      IM=IN+3                                                           GAMM0140
C     ZERO OUT 4 ROWS OF AK                                             GAMM0150
      DO 20 IL=IN,IM                                                    GAMM0160
      DO 20 IK=1,IBNWT                                                  GAMM0170
   20 AK(IL,IK)=0.                                                      GAMM0180
      DO 100 IK=1,4                                                     GAMM0190
      IND=IND+1                                                         GAMM0200
      ICON1=2                                                           GAMM0210
      MFG = IBC(IK,IJH,IBCF)                                            GAMM0220
      FLAG = 1.0                                                        GAMM0230
      GO TO (30,30,30,30,46,56,30,30,30,30,42,54,52,40,46,50), MFG      GAMM0240
   30 NCOL=ICON-IK+MFG                                                  GAMM0250
      IF(MFG.GT.6) NCOL=NCOL-2                                          GAMM0260
      AK(IND,NCOL) = 1.                                                 GAMM0270
      GO TO 100                                                         GAMM0280
   40 CONTINUE                                                          GAMM0290
      FLAG = PI * R1                                                    GAMM0300
   42 ICON1 = 6                                                         GAMM0310
   46 CONTINUE                                                          GAMM0320
      NCOL = ICON + ICON1 - IK                                          GAMM0330
      AK(IND,NCOL)   =  COSI * FLAG                                     GAMM0340
      AK(IND,NCOL+1) =  SINE * FLAG                                     GAMM0350
      IF(MFG.EQ.14) AK(IND,NCOL-1) = -PI * R1                           GAMM0360
      IF(MFG.EQ.15) AK(IND,NCOL-1) = 1.                                 GAMM0370
```

```
      GO TO 100                                                 GAMM0380
   50 CONTINUE                                                  GAMM0390
      FLAG = 1. / R1                                            GAMM0400
      GO TO 56                                                  GAMM0410
   52 FLAG = PI * R1 * R1                                       GAMM0420
   54 ICON1 = 6                                                 GAMM0430
   56 CONTINUE                                                  GAMM0440
      NCOL = ICON + ICON1 - IK                                  GAMM0450
      AK(IND,NCOL)   =  SINE * FLAG                             GAMM0460
      AK(IND,NCOL+1) = -COSI * FLAG                             GAMM0470
      IF(MFG.EQ.13) AK(IND,NCOL+2) = PI * R1                    GAMM0480
      IF(MFG.EQ.16) AK(IND,NCOL+2) = -1.                        GAMM0490
  100 CONTINUE                                                  GAMM0500
      RETURN                                                    GAMM0510
      END                                                       GAMM0520
```

```
$IBFTC BETAX    NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD DISCONTINUITY EQUATIONS                         BETA0000
      SUBROUTINE BETA(MPQ,AK,ECRS,ECYS,ECR,ECY,SCF,RI,MAXSEG,MAXHRM,    BETA0010
     1 SINP,COSP,IBNWT,NROWS,RSS)                                       BETA0020
C     MPQ=1--DISCONTINUITY EQUATIONS FOR BEGINNING OF SEGMENT           BETA0030
C     MPQ=2--DISCONTINUITY EQUATIONS FOR END OF SEGMENT                 BETA0040
      DIMENSION AK(NROWS,IBNWT),ECRS(MAXSEG,MAXHRM),ECYS(MAXSEG,MAXHRM),BETA0050
     1 ECR(MAXSEG),ECY(MAXSEG),SCF(16,MAXSEG,MAXHRM),RI(MAXSEG),        BETA0060
     2 SINP(MAXSEG),COSP(MAXSEG),RSS(MAXSEG,MAXHRM)                     BETA0070
      DIMENSION AF(4,4)                                                 BETA0080
      REAL K1,K2                                                        BETA0085
      COMMON/COM32/SIND,COSD,M                                          BETA0090
      COMMON/COM33/IJ,MS,MF,IND,IJK,IJH,IDIAG                           BETA0100
      COMMON/COM35/ R1,D1,K1,FDC1,FM1,R2,D2,K2,FDC2,FM2                 BETA0110
C     ZERO OUT 4 ROWS OF AK MATRIX                                      BETA0120
      IK=IND+1                                                          BETA0130
      IM=IK+3                                                           BETA0140
      DO 10 I=IK,IM                                                     BETA0150
      DO 10 J=1,IBNWT                                                   BETA0160
   10 AK(I,J)=0.                                                        BETA0170
      IF(MPQ-1) 105,105,15                                              BETA0180
   15 CONTINUE                                                          BETA0190
C     BUILD 'K' COEFFICIENTS                                            BETA0200
      ERS=ECRS(IJK,IJH)                                                 BETA0210
      EYS=ECYS(IJK,IJH)                                                 BETA0220
      RA=R1                                                             BETA0230
      RB=RI(IJK+1)                                                      BETA0240
      I=0                                                               BETA0250
      DO 20 IO=1,4                                                      BETA0260
      DO 20 IP=1,4                                                      BETA0270
      I=I+1                                                             BETA0280
   20 AF(IO,IP)=SCF(I,IJK,IJH)                                          BETA0290
C     SET VALUES FOR SINE(PHI) AND COS(PHI) FOR THIS SEGMENT            BETA0303
      SINE = SIND                                                       BETA0306
      COSI = COSD                                                       BETA0308
      RAA = RSS(IJK,IJH)                                                BETA0295
      CALL KCOF(AF,ERS,EYS,R1,RAA)                                      BETA0300
C     BUILD BOUNDARY CONDITIONS                                         BETA0310
```

```
      S2=SINP(IJK+1)                                                    BETA0320
      C2=COSP(IJK+1)                                                    BETA0330
      SPSI = S2*COSI-C2*SINE                                            BETA0340
      CPSI = C2*COSI+S2*SINE                                            BETA0350
      ER=ECR(IJK)                                                       BETA0360
      EY=ECY(IJK)                                                       BETA0370
      A1=RB/RA                                                          BETA0380
      A2 = ER*S2-EY*C2                                                  BETA0390
      A3 = ER*C2+EY*S2                                                  BETA0400
      DO 100 IA=1,4                                                     BETA0410
      NCOL =IDIAG-4-IA                                                  BETA0420
      IND=IND+1                                                         BETA0430
      DO 40  IB=1,4                                                     BETA0440
      NCOL1=NCOL+IB                                                     BETA0450
      AK(IND,NCOL1) = AF(IA,IB)                                         BETA0460
   40 AK(IND,IDIAG) = 1.                                                BETA0470
      NCOL=IDIAG+10-IA                                                  BETA0480
      GO TO (50,60,70,80), IA                                           BETA0490
   50 AK(IND,IDIAG+8) = -A1                                             BETA0500
      GO TO 100                                                         BETA0510
   60 AK(IND,NCOL)  =-A1*CPSI                                           BETA0530
      AK(IND,NCOL+1)=-A1*SPSI                                           BETA0540
      GO TO 100                                                         BETA0550
   70 AK(IND,NCOL)    = A1*SPSI                                         BETA0560
      AK(IND,NCOL+1) =-A1*CPSI                                          BETA0570
      GO TO 100                                                         BETA0580
   80 AK(IND,NCOL)  =-A1*A2                                             BETA0590
      AK(IND,NCOL+1)= A1*A3                                             BETA0600
      AK(IND,NCOL+2)=-A1                                                BETA0610
  100 CONTINUE                                                          BETA0620
      GO TO 210                                                         BETA0630
  105 CONTINUE                                                          BETA0640
      R1I=1./RA                                                         BETA0650
      FM=M                                                              BETA0660
      NCOL=IDIAG-8                                                      BETA0670
      DO 200 IA=1,4                                                     BETA0680
      IND=IND+1                                                         BETA0690
      AK(IND,NCOL) = 1.                                                 BETA0700
```

```
      GO TO (110,130,150,170), IA                                BETA0710
  110 AK(IND,NCOL)   =1.+ER*R1I                                  BETA0720
      AK(IND,NCOL+1)=FM*R1I*(COSI*ER+SINE*EY)                    BETA0730
      AK(IND,NCOL+2)=FM*R1I*(SINE*ER-COSI*EY)                    BETA0740
      AK(IND,IDIAG )=-1.                                         BETA0750
      GO TO 200                                                  BETA0760
  130 AK(IND,NCOL+2) = ER*SINE-EY*COSI                           BETA0770
      AK(IND,IDIAG)   = -CPSI                                    BETA0780
      AK(IND,IDIAG+1)= -SPSI                                     BETA0790
      GO TO 200                                                  BETA0800
  150 AK(IND,NCOL+1) = -ER*COSI-EY*SINE                          BETA0810
      AK(IND,IDIAG-1)=  SPSI                                     BETA0820
      AK(IND,IDIAG)   = -CPSI                                    BETA0830
      GO TO 200                                                  BETA0840
  170 AK(IND,IDIAG)   = -1.                                      BETA0850
  200 CONTINUE                                                   BETA0860
  210 CONTINUE                                                   BETA0870
      RETURN                                                     BETA0880
      END                                                        BETA0890
```

```
$IBFTC KCOFX   NOLIST,DECK,194,XR7,NODD
```

```
C                 BUILD 'K' COEFFICIENTS                          KCOF0000
      SUBROUTINE KCOF(AF,ERS,EYS,R1,RAA)                          KCOF0010
      DIMENSION AF(4,4), A(4,4), C(4,4)                           KCOF0020
      COMMON/COM31/AC(8,8),AD(8,8)                                KCOF0030
      COMMON/COM32/SINE,COSI,M                                    KCOF0040
      EQUIVALENCE( A(1,1),AC(1,1)),(C(1,1),AC(1,3))               KCOF0050
      FM=M                                                        KCOF0060
      DO 10 I=1,4                                                 KCOF0070
      DO 10 J=1,4                                                 KCOF0080
   10 A(I,J) = 0.                                                 KCOF0090
      R1I=1./R1                                                   KCOF0100
      RBB=RAA/R1                                                  KCOF0105
C     BUILD 'A' MATRIX                                            KCOF0110
      A(1,1) =  1.+ERS*R1I                                        KCOF0120
      A(1,2) =  FM*R1I*(ERS*COSI+EYS*SINE)                        KCOF0130
      A(1,3) =  FM*R1I*(ERS*SINE-EYS*COSI)                        KCOF0140
      A(2,2) =  SINE                                              KCOF0150
      A(2,3) = -COSI                                              KCOF0160
      A(2,4) =  ERS                                               KCOF0170
      A(3,2) =  COSI                                              KCOF0180
      A(3,3) =  SINE                                              KCOF0190
      A(3,4) = -EYS                                               KCOF0200
      A(4,4) =  1.                                                KCOF0210
C     CALCULATE 'K' COEFFICIENTS                                  KCOF0220
      DO 20 I=1,4                                                 KCOF0230
      DO 20 J=1,4                                                 KCOF0240
      C(I,J) =0.                                                  KCOF0250
      DO 20 K=1,4                                                 KCOF0260
   20 C(I,J) =C(I,J)+AF(I,K)*A(K,J)                               KCOF0270
      DO 40 I=1,4                                                 KCOF0280
      DO 40 J=1,4                                                 KCOF0290
      AF(I,J)=0.                                                  KCOF0300
      DO 30 K=1,4                                                 KCOF0310
   30 AF(I,J)=AF(I,J)+ A(K,I) * C(K,J)                            KCOF0320
   40 AF(I,J)=RBB*AF(I,J)                                         KCOF0330
      RETURN                                                      KCOF0340
      END                                                         KCOF0350
```

```
$IBFTC EQUMX    NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD EQUILIBRIJM EQUATIONS                          EQUM0000
      SUBROUTINE EQUM(AK,NROWS,IBNWT)                                  EQUM0010
      DIMENSION AK(NROWS,IBNWT)                                        EQUM0020
      REAL K1,K2                                                       EQUM0025
      COMMON/COM31/AC(8,8),AD(8,8)                                     EQUM0030
      COMMON/COM33/IJ,MS,MF,IND,IJK,IJH,IDIAG                          EQUM0040
      COMMON/COM35/ R1,D1,K1,FDC1,FM1,R2,D2,K2,FDC2,FM2                EQUM0050
      DO 190 IK=1,8                                                    EQUM0060
      IND=IND+1                                                        EQUM0070
C     ZERO OUT ONE ROW OF K MATRIX                                     EQUM0080
      DO 165 NN=1,IBNWT                                                EQUM0090
  165 AK(IND,NN)=C.                                                    EQUM0100
C     BUILD 8 EQUATIONS FOR POINT 'N'                                  EQUM0110
      DO 170 IL=1,8                                                    EQUM0120
      IN=8-IK+IL                                                       EQUM0130
  170 AK(IND,IN)= .5*AC(IK,IL)                                         EQUM0140
C     BUILD 8 EQUATIONS FOR POINT 'N+1'                                EQUM0150
      DO 180 IL=1,8                                                    EQUM0160
      IN=16-IK+IL                                                      EQUM0170
  180 AK(IND,IN)= .5*AD(IK,IL)                                         EQUM0180
C     INSERT MAJOR DIAGONAL ADDITION                                   EQUM0190
      DDAG1= FDC1                                                      EQUM0200
      DDAG = DDAG1                                                     EQUM0210
      IF(IK.GT.4) DDAG=-DDAG                                           EQUM0220
      AK(IND,IDIAG)=AK(IND,IDIAG)-DDAG                                 EQUM0230
C     INSERT MINOR DIAGONAL ADDITION                                   EQUM0240
      DDAG=DDAG1                                                       EQUM0250
      IDAG1=IDIAG+8                                                    EQUM0260
      IF(IK.LT.5) GO TO 186                                            EQUM0270
      IDAG1=IDIAG-8                                                    EQUM0280
      DDAG =-DDAG1                                                     EQUM0290
  186 AK(IND,IDAG1)=AK(IND,IDAG1)+DDAG                                 EQUM0300
  190 CONTINUE                                                         EQUM0310
      RETURN                                                           EQUM0320
      END                                                              EQUM0330
```

```
$IBFTC BRHVX    NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD INITIAL RIGHT HAND VECTOR                       BRHV0000
      SUBROUTINE BRHV(RHV,BC,FMPO,FNRO,FNYO,FNPTO,FLDI,FLDF,MSG,MHM,MPV,BRHV0010
     1 S,MAXP)                                                          BRHV0020
C     MPV=1--BOUNDARY CONDITIONS                                        BRHV0030
C     MPV=2--DISCONTINUITY EQUATIONS                                    BRHV0040
C     MPV=3--EQUILIBRIUM EQUATIONS                                      BRHV0050
      DIMENSION X(3,2)                                                  BRHV0060
      DIMENSION RHV(600),BC(4,MHM,2),FMPO(2,MSG,MHM),FNRO(2,MSG,MHM),   BRHV0070
     1 FNYO(2,MSG,MHM),FNPTO(2,MSG,MHM),FLDI(5,MSG,MHM),FLDF(5,MSG,MHM),BRHV0080
     2 S(MAXP)                                                          BRHV0090
      COMMON/COM32/SINE,COSI,M                                          BRHV0100
      COMMON/COM33/IJ,MS,MF,IND,IJK,IJH,IDIAG                           BRHV0110
      GO TO(200,400,600), MPV                                           BRHV0120
  200 CONTINUE                                                          BRHV0130
C     LEADING BOUNDARY CONDITIONS--RHV                                  BRHV0140
      IFG=1                                                             BRHV0150
      IRHV=IND-4                                                        BRHV0160
      DEL=ABS(S(MF)-S(MS))                                              BRHV0170
      IF(IJ.EQ.MS) GO TO 210                                            BRHV0180
C     TRAILING BOUNDARY CONDITIONS--RHV                                 BRHV0190
      IFG=2                                                             BRHV0200
  210 CONTINUE                                                          BRHV0210
      DO 250 I=1,4                                                      BRHV0220
      IRHV=IRHV+1                                                       BRHV0230
      RHV(IRHV)=BC(I,IJH,IFG)                                           BRHV0240
  250 CONTINUE                                                          BRHV0250
      GO TO 800                                                         BRHV0260
  400 CONTINUE                                                          BRHV0270
      IRHV=IND-4                                                        BRHV0280
      IF(IJ.EQ.MF) GO TO 500                                            BRHV0290
C     STARTING DISCONTINUITY EQUATIONS--RHV                             BRHV0300
      DEL=ABS(S(MF)-S(MS))                                              BRHV0310
      DO 450 I=1,4                                                      BRHV0320
      IRHV=IRHV+1                                                       BRHV0330
  450 RHV(IRHV)=0.                                                      BRHV0340
      GO TO 800                                                         BRHV0350
  500 CONTINUE                                                          BRHV0360
```

```
C     ENDING DISCONTINUITY EQUATIONS--RHV                          BRHV0370
      IRHV=IRHV+1                                                  BRHV0380
      RHV(IRHV) = FNPTO(1,IJK,IJH)                                 BRHV0390
      IRHV=IRHV+1                                                  BRHV0400
      RHV(IRHV) = FNRO(1,IJK,IJH)*COSI+FNYO(1,IJK,IJH)*SINE        BRHV0410
      IRHV=IRHV+1                                                  BRHV0420
      RHV(IRHV) = FNRO(1,IJK,IJH)*SINE-FNYO(1,IJK,IJH)*COSI        BRHV0430
      IRHV=IRHV+1                                                  BRHV0440
      RHV(IRHV) = FMPO(1,IJK,IJH)                                  BRHV0450
      GO TO 800                                                    BRHV0460
  600 CONTINUE                                                     BRHV0470
      IRHV=IND-8                                                   BRHV0480
C     CALCULATE VALUES OF XBAR,YBAR,AND ZBAR                       BRHV0490
      DO 650 I=1,3                                                 BRHV0500
      SLOPE = (FLDF(I,IJK,IJH)-FLDI(I,IJK,IJH))/DEL                BRHV0510
      DO 650 J=1,2                                                 BRHV0520
      IZ=IJ-1+J                                                    BRHV0530
      X(I,J)= FLDI(I,IJK,IJH)+ SLOPE*S(IZ)                         BRHV0540
  650 CONTINUE                                                     BRHV0550
C     EQUILIBRIUM EQUATIONS--RHV                                   BRHV0560
      DO 680 I=1,3                                                 BRHV0570
      IRHV=IRHV+1                                                  BRHV0580
      RHV(IRHV) = -.5*(X(I,1)+X(I,2))                              BRHV0590
  680 CONTINUE                                                     BRHV0600
      DO 700 I=1,5                                                 BRHV0610
      IRHV=IRHV+1                                                  BRHV0620
      RHV(IRHV)=0.                                                 BRHV0630
  700 CONTINUE                                                     BRHV0640
  800 RETURN                                                       BRHV0650
      END                                                          BRHV0660
```

SUBROUTINE MON3

COMMON VARIABLES

	COMMON BLOCK		COMAA		ORIGIN	00001	LENGTH	00001
SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ICHECK	00000	I						
	COMMON BLOCK		COMCC		ORIGIN	00002	LENGTH	00001
CONFAC	00000	R						
	COMMON BLOCK		COM31		ORIGIN	00003	LENGTH	00200
AC	00000	R		AD	00100	R		
	COMMON BLOCK		COM32		ORIGIN	00203	LENGTH	00003
SINE	00000	R		COSI	00001	R	M	00002
	COMMON BLOCK		COM33		ORIGIN	00206	LENGTH	00007
IJ	00000	I		MS	00001	I	MF	00002
IND	00003	I		IJK	00004	I	IJH	00005
IDIAG	00006	I						
	COMMON BLOCK		COM34		ORIGIN	00215	LENGTH	00001
LOOP	00000	I						
	COMMON BLOCK		COM36		ORIGIN	00216	LENGTH	02261
GOTH	00000	R		GOPH	00454	R	GNPT	01130
GNTH	01604	R		SIFOR	02260	R		
	COMMON BLOCK		//		ORIGIN	02477	LENGTH	33167
NSGMTS	00000	I		KS	00001	I	ISGMT	00002
IETNO	00003	I		LOADNO	00004	I	LDNOPR	00005
NLOAD	00006	I		KKKKK	00007	I	PHI	00010
SINP	00017	R		COSP	00026	R	NPSEG	00035
LSP	00044	I		MLIST	00053	I	ECR	00062
ECY	00071	R		IOPT	00100	I	IRES	00116

NREC	00130	I	INTF	00155	I	NSGPL	00167
NPTPT	00201	I	NLDPL	00213	I	EI	00225
HI	00243	R	FMUI	00261	R	RI	00277
FLDI	00306	R	FLDF	00673	R	DLDI	01260
DLDF	01645	R	FMPO	02232	R	FNRO	02374
FNYO	02536	R	FNPTO	02700	R	ECRS	03042
ECYS	03123	R	RSS	03204	R	SCF	03265
IBC	04705	I	BC	04775	R	DLBC	05065
S	05155	R	FNTH	05631	R	FNPH	06305
OPHI	06761	R	OTHE	07435	R	STP	10111
AK	10235	R	AU	16461	R	AL	24705
RHV	31755	R	SAVE	33105	R	PS	16461
INTF	00155	I	INTF1	00155	I	INTF2	00156
INTF3	00157	I					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
II	35666	I	I	35667	I	GM	35670
JJK	35671	I	NPTS	35672	I	NROS	35673
SINN	35674	R	COSS	35675	R	R1	35676
S1	35677	R	ESL	35700	R	HSL	35701
FMU	35702	R	J	35703	I	JJ	35704
KK	35705	I	EE	35706	R	HH	35707
FF	35710	R	RRI	35711	R	OSS	35712
OSI	35713	R	DD	35714	R		

ENTRY POINTS

MON3	SECTION	18

SUBROUTINES CALLED

.FRWT.	SECTION	19	HARM	SECTION	20	.FRDB.	SECTIO	
.FWRD.	SECTION	22	.UN14.	SECTION	23	.UN04.	SECTIO	

.FRLR.	SECTION	25	.FBLT.	SECTION	26	.FBDT.	SECTION
.UN06.	SECTION	28	.FFIL.	SECTION	29	.FCNV.	SECTION
E.1	SECTION	31	E.2	SECTION	32	E.3	SECTION
E.4	SECTION	34	CC.1	SECTION	35	CC.2	SECTION
CC.3	SECTION	37	CC.4	SECTION	38	SYSLOC	SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
100	15A	36016	500	128A	36526	480	126A
450	123A	36514	150	39A	36070	400	104A
200	62A	36161	300	102A	36436	220	79A
2000	FORMAT	35737	425	114A	36466	2050	FORMAT

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 36564.

SUBROUTINE HARM

COMMON VARIABLES

COMMON BLOCK	COM31		ORIGIN	00001		LENGTH	00200
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
AC	00000	R	AD	00100	R		
COMMON BLOCK	COM32		ORIGIN	00201		LENGTH	00003
SINE	00000	R	COSI	00001	R	M	00002
COMMON BLOCK	COM33		ORIGIN	00204		LENGTH	00007
IJ	00000	I	MS	00001	I	MF	00002
IND	00003	I	IJK	00004	I	IJH	00005
IDIAG	00006	I					
COMMON BLOCK	COM35		ORIGIN	00213		LENGTH	00012
R1	00000	R	D1	00001	R	K1	00002
FDC1	00003	R	FM1	00004	R	R2	00005
D2	00006	R	K2	00007	R	FDC2	00010
FM2	00011	R					
COMMON BLOCK	//		ORIGIN	00225		LENGTH	33167
NSGMTS	00000	I	KS	00001	I	ISGMT	00002
IFTNO	00003	I	LOADNO	00004	I	LDNOPR	00005
NLOAD	00006	I	KKKKK	00007	I	PHI	00010
SINP	00017	R	COSP	00026	R	NPSEG	00035
LSP	00044	I	MLIST	00053	I	ECR	00062
ECY	00071	R	IOPT	00100	I	IRES	00116
NREC	00130	I	INTF	00155	I	NSGPL	00167
NPTPT	00201	I	NLDPL	00213	I	EI	00225
HI	00243	R	FMUI	00261	R	RI	00277
FLDI	00306	R	FLDF	00673	R	DLDI	01260
DLDF	01645	R	FMPO	02232	R	FNRO	02374
FNYO	02536	R	FNPTO	02700	R	ECRS	03042
ECYS	03123	R	RSS	03204	R	SCF	03265

HARMX STORAGE MAP

IBC	04705	I	BC	04775	R	DLBC	05065
S	05155	R	FNTH	05631	R	FNPH	06305
OPHI	06761	R	OTHE	07435	R	STP	10111
AK	10235	R	AU	16461	R	AL	24705
RHV	31755	R	SAVE	33105	R	RHM	24705
RHK	10235	R	AK	10235	R		

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
FMAT	33414	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
MAXSEG	33430	I	MAXHRM	33431	I	IBNWT	33432
NROWS	33433	I	MAXPTS	33434	I	MAXMPT	33435
MPT	33436	I	NRHV	33437	I	IRE	33440
IRHVF_	33441	I	IJL	33442	I	NRFLAG	33443
KK	33444	I	MF1	33445	I	IBCF	33446
NPFLG	33447	I	MPV	33450	I	MPQ	33451
I	33452	I	IZ	33453	I	J	33454
NPG1	33455	I	IN	33456	I		

ENTRY POINTS

HARM	SECTION	12

SUBROUTINES CALLED

.FWRD.	SECTION	13	MISC	SECTION	14	GAMM	SECTION
BETA	SECTION	16	BRHV	SECTION	17	ACAD	SECTION
EQUM	SECTION	19	.FWRB.	SECTION	20	.FXEM.	SECTION
.UNO6.	SECTION	22	.FFIL.	SECTION	23	.FCNV.	SECTION

.UN14.	SECTION	25	.FWLR.	SECTION	26	.FBLT.	SECTIO	
.FBDT.	SECTION	28	SYSLOC	SECTION	29			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
2000	FORMAT	33506	2001	FORMAT	33512	2002	FORMAT
2003	FORMAT	33526	2004	FORMAT	33532	2005	FORMAT
20	24A	33621	30	37A	33652	60	53A
50	49A	33704	140	69A	33745	145	72A
90	111A	34172	80	64A	33734	150	74A
155	79A	34024	190	110A	34161	160	93A
185	106A	34144	100	112A	34172	200	159A
162	123A	34241	165	132A	34271	166	133A
170	142A	34333	180	155A	34376	300	165A
290	161A	34420	310	168A	34434	500	198A
320	192A	34516	315	189A	34514	1000	201A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 34564.

SUBROUTINE MISC

COMMON VARIABLES

COMMON BLOCK	COM32		ORIGIN	00001		LENGTH	00003
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
SINE	00000	R	COSI	00001	R	M	00002
COMMON BLOCK	COM33		ORIGIN	00004		LENGTH	00007
IJ	00000	I	MS	00001	I	MF	00002
IND	00003	I	IJK	00004	I	IJH	00005
IDIAG	00006	I					
COMMON BLOCK	COM35		ORIGIN	00013		LENGTH	00012
R1	00000	R	D1	00001	R	K1	00002
FDC1	00003	R	FM1	00004	R	R2	00005
D2	00006	R	K2	00007	R	FDC2	00010
FM2	00011	R					
COMMON BLOCK	COM37		ORIGIN	00025		LENGTH	00002
ZBR	00000	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
D12	00027	R	IZ	00030	I	DEL	00031
SLOPEE	00032	R	SLOPEH	00033	R	SLOPEU	00034
SLOPZB	00035	R	H1	00036	R	E1	00037
IK •	00040	I	E2	00041	R	H2	00042

ENTRY POINTS

MISC SECTION 11

SUBROUTINES CALLED

SYSLOC SECTION 12

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
100	30A	00203	150	34A	00227	200	49A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 0C444.

STORAGE MAP

SUBROUTINE ACAD

COMMON VARIABLES

COMMON BLOCK	COM31		ORIGIN	00001		LENGTH	00200
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
AC	00000	R	AD	00100	R		
COMMON BLOCK	COM32		ORIGIN	00201		LENGTH	00003
SINE	00000	R	COSI	00001	R	M	00002
COMMON BLOCK	COM33		ORIGIN	00204		LENGTH	00007
IJ	00000	I	MS	00001	I	MF	00002
IND	00003	I	IJK	00004	I	IJH	00005
IDIAG	00006	I					
COMMON BLOCK	COM34		ORIGIN	00213		LENGTH	00001
LOOP	00000	I					
COMMON BLOCK	COM35		ORIGIN	00214		LENGTH	00012
R1	00000	R	D1	00001	R	K1	00002
FDC1	00003	R	FM1	00004	R	R2	00005
D2	00006	R	K2	00007	R	FDC2	00010
FM2	00011	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
JJ	00226	I	IZ	00227	I		

ENTRY POINTS

ACAD SECTION 13

ACADX STORAGE MAP

SUBROUTINES CALLED

ACOF	SECTION 14	MACO	SECTION 15	SYSLOC	SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
120	12A	00274	140	24A	00316	130	18A	
150	32A	00353						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00416.

SUBROUTINE ACOF

COMMON VARIABLES

COMMON BLOCK COM32 ORIGIN 00001 LENGTH 00003

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
SINE	00000	R	COSI	00001	R	M	00002

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
LNI	00004	R	M1	00005	R	MSQ	00006
R1I	00007	R	R2	00010	R	R2I	00011
R3	00012	R	R3I	00013	R	OMU	00014
OMUS	00015	R	SISQ	00016	R	COSQ	00017
SICO	00020	R	T1	00021	R		

ENTRY POINTS

ACOF SECTION 5

SUBROUTINES CALLED

E.1	SECTION	6	E.2	SECTION	7	E.3	SECTION
E.4	SECTION	9	CC.1	SECTION	10	CC.2	SECTION
CC.3	SECTION	12	CC.4	SECTION	13	SYSLOC	SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
50	5A	00766	60	12A	01013	70	19A	
80	26A	01060	1000	30A	01071			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01364.

SUBROUTINE MACO

COMMON VARIABLES

COMMON BLOCK	COM32		ORIGIN	00001		LENGTH	00003
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
SINE	00000	R	COSI	00001	R	M	00002
COMMON BLOCK	COM33		ORIGIN	00004		LENGTH	00007
IJ	00000	I	MS	00001	I	MF	00002
IND	00003	I	IJK	00004	I	IJH	00005
IDIAG	00006	I					
COMMON BLOCK	COM36		ORIGIN	00013		LENGTH	02261
GOTH	00000	R	GOPH	00454	R	GNPT	01130
GNTH	01604	R	SIFOR	02260	R		
COMMON BLOCK	COM37		ORIGIN	02274		LENGTH	00002
ZBR	00000	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
M1	02276	R	MSQ	02277	R	LNI	02300
MZB	02301	I	FNT	02302	R	FNP	02303
OT	02304	R	OP	02305	R	FM	02306
OMU	02307	R	OMUS	02310	R	R1I	02311
R2	02312	R	R2I	02313	R	R3	02314
R3I	02315	R	D1I	02316	R	SISQ	02317
COSQ	02320	R	SICO	02321	R	T1	02322
T2	02323	R	HNT	02324	R	HNPT	02325
HOT	02326	R	HOP	02327	R	TT1	02330
TT2	02331	R	TT3	02332	R	TT4	02333

MACOX STORAGE MAP

ENTRY POINTS

MACO	SECTION	11

SUBROUTINES CALLED

E.1	SECTION	12	E.2	SECTION	13	E.3	SECTION
E.4	SECTION	15	CC.1	SECTION	16	CC.2	SECTION
CC.3	SECTION	18	CC.4	SECTION	19	SYSLOC	SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
500	19A	03512						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 03765.

STORAGE MAP

SUBROUTINE GAMM

COMMON VARIABLES

COMMON BLOCK	COM32		ORIGIN	00001		LENGTH	00003
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
SINE	00000	R	COSI	00001	R	M	00002
COMMON BLOCK	COM33		ORIGIN	00004		LENGTH	00007
IJ	00000	I	MS	00001	I	MF	00002
IND	00003	I	IJK	00004	I	IJH	00005
IDIAG	00006	I					
COMMON BLOCK	COM35		ORIGIN	00013		LENGTH	00012
R1	00000	R	D1	00001	R	K1	00002
FDC1	00003	R	FM1	00004	R	R2	00005
D2	00006	R	K2	00007	R	FDC2	00010
FM2	00011	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
PI	00025	R	ICON	00026	I	IN	00027
IM	00030	I	IL	00031	I	IK	00032
ICON1	00033	I	MFG	00034	I	FLAG	00035
NCOL	00036	I					

ENTRY POINTS

GAMM SECTION 9

SUBROUTINES CALLED

GAMMX STORAGE MAP

.FXEM. SECTION 10 SYSLOC SECTION 11

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
20	12A	00131	100	58A	00403	30	25A
46	33A	00243	56	48A	00331	42	32A
54	47A	00327	52	46A	00322	40	31A
50	44A	00316					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00541.

SUBROUTINE BETA

COMMON VARIABLES

COMMON BLOCK		COM32		ORIGIN	00001	LENGTH	00003
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
SIND	00000	R	COSD	00001	R	M	00002
COMMON BLOCK		COM33		ORIGIN	00004	LENGTH	00007
IJ	00000	I	MS	00001	I	MF	00002
IND	00003	I	IJK	00004	I	IJH	00005
IDIAG	00006	I					
COMMON BLOCK		COM35		ORIGIN	00013	LENGTH	00012
R1	00000	R	D1	00001	R	K1	00002
FDC1	00003	R	FM1	00004	R	R2	00005
D2	00006	R	K2	00007	R	FDC2	00010
FM2	00011	R					

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
AF	00025	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IK	00045	I	IM	00046	I	I	00047
ERS	00050	R	EYS	00051	R	RA	00052
RB	00053	R	ID	00054	I	SINE	00055
COSI	00056	R	RAA	00057	R	S2	00060
C2	00061	R	SPSI	00062	R	CPSI	00063
ER	00064	R	EY	00065	R	A1	00066
A2	00067	R	A3	00070	R	IA	00071

BETAX STORAGE MAP

NCOL	00072	I	IB	00073	I	NCOL1	00074
R1I	00075	R	FM	00076	R		

ENTRY POINTS

BETA SECTION 9

SUBROUTINES CALLED

KCOF	SECTION	10	.FXEM.	SECTION	11	E.1	SECTION
E.2	SECTION	13	E.3	SECTION	14	E.4	SECTION
CC.1	SECTION	16	CC.2	SECTION	17	CC.3	SECTION
CC.4	SECTION	19	SYSLOC	SECTION	20		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
10	9A	00157	105	71A	00530	15	15A	
20	25A	00220	100	67A	00525	40	48A	
50	52A	00420	60	55A	00432	70	59A	
80	63A	00477	210	100A	00767	200	97A	
110	79A	00576	130	85A	00661	150	90A	
170	95A	00754						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01253.

SUBROUTINE KCOF

COMMON VARIABLES

COMMON BLOCK COM31 ORIGIN 00001 LENGTH 00200

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
AC	00000	R	AD	00100	R	A	00000
AC	00000	R	C	00020	R	AC	00000

COMMON BLOCK COM32 ORIGIN 00201 LENGTH 00003

SINE	00000	R	COSI	00001	R	M	00002

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
FM	00204	R	I	00205	I	J	00206
R1I	00207	R	RBB	00210	R		

ENTRY POINTS

KCOF	SECTION	7

SUBROUTINES CALLED

E.1	SECTION	8	E.2	SECTION	9	E.3	SECTION	
E.4	SECTION	11	CC.1	SECTION	12	CC.2	SECTION	
CC.3	SECTION	14	CC.4	SECTION	15	SYSLOC	SECTION	

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
10	7A	00240	20	20A	00362	40	43A	

30 37A 00435
THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00527.

SUBROUTINE EQUM

COMMON VARIABLES

COMMON BLOCK		COM31		ORIGIN	00001	LENGTH	00200
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
AC	00000	R	AD	00100	R		
COMMON BLOCK		COM33		ORIGIN	00201	LENGTH	00007
IJ	00000	I	MS	00001	I	MF	00002
IND	00003	I	IJK	00004	I	IJH	00005
IDIAG	00006	I					
COMMON BLOCK		COM35		ORIGIN	00210	LENGTH	00012
R1	00000	R	D1	00001	R	K1	00002
FDC1	00003	R	FM1	00004	R	R2	00005
D2	00006	R	K2	00007	R	FDC2	00010
FM2	00011	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IK	00222	I	IL	00223	I	IN	00224
DDAG1	00225	R	DDAG	00226	R	IDAG1	00227

ENTRY POINTS

EQUM SECTION 9

SUBROUTINES CALLED

SYSLOC SECTION 10

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
190	35A	00426	165	8A	00263	170	13A	
180	19A	00332	186	32A	00413			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00512.

BRHVX STORAGE MAP

SUBROUTINE BRHV

COMMON VARIABLES

COMMON BLOCK COM32 ORIGIN 00001 LENGTH 00003

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
SINE	0000C	R	COSI	00001	R	M	00002

COMMON BLOCK COM33 ORIGIN 00004 LENGTH 00007

IJ	00000	I	MS	00001	I	MF	00002
IND	00003	I	IJK	00004	I	IJH	00005
IDIAG	00006	I					

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
X	00013	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IFG	00021	I	IRHV	00022	I	DEL	00023
I	00024	I	SLOPE	00025	R	J	00026
IZ	00027	I					

ENTRY POINTS

BRHV SECTION 7

SUBROUTINES CALLED

.FXEM. SECTION 8 SYSLOC SECTION 9

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
200	3A	00070	400	22A	00141	600	53A
210	11A	00113	250	18A	00136	800	89A
500	37A	00173	450	32A	00164	650	67A
680	78A	00346	700	86A	00360		

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00570.

```
$IBFTC MDN4X   NOLIST,DECK,M94,XR7,NODD
```

```
C                 MONITOR REGION 4, SOLVE EQUATIONS                     MON40010
      SUBROUTINE MON4                                                   MON40020
      DIMENSION TC(12,23),RHK(2400),RHM(2400)                           MON40030
      DIMENSION AX(140),P(140)                                          MON40040
      DOUBLE PRECISION TC,CT,ALT,DTM                                    MON40050
      EQUIVALENCE(INTF(1),INTF1),(INTF(4),INTF4),(INTF(5),INTF5)        MON40060
      EQUIVALENCE(RHK(1),AK(1,1)),(RHM(1),AL(1,1))                      MON40070
      EQUIVALENCE(RHV(201),P(1)),(AX(1),RHV(1))                         MON40080
      COMMON/COMAA/ICHECK                                               MON40090
      COMMON/COM41/KRMP1,KRMP2,MRT,KBL,IPTBL(7)                         MON40100
      COMMON/COM42/AB(23),U(23),ST(12),AL1(13),R(1),X(1),C(1)           MON40110
      COMMON/COM43/ DTM,DTM1                                            MON40120
      COMMON/COM44/CRT(12,1),ALT(13),CT(12)                             MON40130
      COMMON/COM45/INTF2                                                MON40140
      COMMON/COM49/IOPT5                                                MON40150
      COMMON NSGMTS, KS, ISGMT, IFTNO, LOADNO, LDNCPR, NLOAD, KKKKK     MON40160
      COMMON PHI(7),SINP(7),COSP(7),NPSEG(7),LSP(7),MLIST(7),ECR(7),    MON40170
     1 ECY(7),IOPT(14),IRES(10),NREC(21),INTF(10),NSGPL(10),NPTPT(10),  MON40180
     2 NLDPL(10)                                                        MON40190
      COMMON EI(2,7),HI(2,7),FMUI(2,7),RI(7)                            MON40200
      COMMON FLDI(5,7,7),FLDF(5,7,7),DLDI(5,7,7),DLDF(5,7,7)            MON40210
      COMMON FMPO(2,7,7),FNRO(2,7,7),FNYO(2,7,7),FNPTO(2,7,7),ECRS(7,7),MON40220
     1 ECYS(7,7),RSS(7,7),SCF(16,7,7)                                   MON40230
      COMMON IBC(4,7,2),BC(4,7,2),DLBC(4,7,2)                           MON40240
      COMMON S(300),FNTH(300),FNPH(300),OPHI(300),CTHE(300),STP(12,7)   MON40250
      COMMON AK(140,23),AU(140,23),AL(200,13),RHV(600),SAVE(5,10)       MON40260
 5003 FORMAT(14HOSCALE FACTORS)                                         MON40270
 5004 FORMAT(1H0,2X,10HHARMONIC =,I3/3X,1P4E15.7/3X,4E15.7/3X,4E15.7)   MON40280
 5010 FORMAT(1H1,A6,6HVECTOR)                                           MON40290
 5050 FORMAT(10X,50HSOLUTION HAS DIVERGED OR MACHINE ERROR HAS OCCURED, MON40300
     1 //10X,30HPRIOR SOLUTION TAPE STILL GOOD//10X,14HTERMINATE CASE)  MON40310
 5060 FORMAT(30X,29HINCREMENTAL SOLUTION FOR M = ,I4//)                 MON40320
C     SET CASE FLAGS AND COMMON BLOCKS                                  MON40330
      IOPT5  = IOPT(5)                                                  MON40340
      KRMP1  =  12                                                      MON40350
      KRMP2  =  13                                                      MON40360
      MRT    =   1                                                      MON40370
```

```
      KBL     =  23                                                     MON40380
      N12     =  12                                                     MON40390
      NROWS   = 140                                                     MON40400
      NROWL   = 200                                                     MON40410
      NCOLL   =  13                                                     MON40420
      MAXMPT  =  17                                                     MON40430
C     SET IPTBL FLAGS                                                   MON40440
      NPTS=0                                                            MON40450
      DO 50 IJK=1,NSGMTS                                                MON40460
   50 NPTS = NPTS+NPSEG(IJK)                                            MON40470
      IPTBL(2) = 8*NPTS                                                 MON40480
      IPTBL(4) = 11                                                     MON40490
      IPTBL(7) =  1                                                     MON40500
      INTF2  = INTF(02)                                                 MON40510
C     CALCULATE NUMBER OF RECORDS USED FOR 'K' MATRIX                   MON40520
      IREC   = (NPTS-1)/MAXMPT+1                                        MON40530
      MAXREC = IREC                                                     MON40540
      IF(INTF4-1) 100,100,400                                           MON40550
  100 CONTINUE                                                          MON40560
C     IF ICHECK = 0, SKIP CHECKOUT PRINTING OPTIONS                     MON40570
      IF(ICHECK.EQ.0) GO TO 110                                         MON40580
      IF(IOPT(14).EQ.0) GO TO 110                                       MON40590
      IF(IOPT(14)-2) 105,110,105                                        MON40600
  105 CONTINUE                                                          MON40610
      IJH = 1                                                           MON40620
      IF(INTF(2)) 107,107,108                                           MON40630
  107 CALL PRMY(RHK,LSP,NPSEG,NSGMTS,IJH)                               MON40640   37
      GO TO 110                                                         MON40650
  108 CALL PRMY(RHM,LSP,NPSEG,NSGMTS,IJH)                               MON40660   40
  110 CONTINUE                                                          MON40670
      KS1=KS                                                            MON40680
  120 CONTINUE                                                          MON40690
      DO 350 IJH=1,KS1                                                  MON40700
      INTF5 = IJH                                                       MON40710
C     CALCULATE 'U' AND 'K' MATRICES                                    MON40720
      CALL LANU(AK,AU,AL,RHV,P,TC,NREC,LSP,NPSEG,RHK,RHM,NROWS,NROWL,   MON40730
     1  N12)                                                            MON40740   47
      IF(IJH.EQ.KS) REWIND 14                                           MON40750   49
```

```
      DO 130 I=1,KRMP1                                          MON40760
  130 STP(I,IJH)=ST(I)                                          MON40770
  150 CONTINUE                                                  MON40780
C     IF THIS IS A M=0 ITERATION, GO TO 450                     MON40790
      IF(INTF2.GT.0) GO TO 450                                  MON40800
      IUREC= IREC                                               MON40810
  200 CONTINUE                                                  MON40820
C     BACKSPACE U-TAPE PROPER NUMBER OF RECORDS                 MON40830
      BACKSPACE 15                                              MON40840   61
      IF(IUREC.NE.IREC) BACKSPACE 15                            MON40850   62
  250 CONTINUE                                                  MON40860
C     READ A RECORD FROM 'U' TAPE                               MON40870
      NREC1=NREC(IUREC)                                         MON40880
      READ(15) ((AU(I,J),I=1,NREC1),J=1,KBL)                    MON40890
C     WRITE RECORD OF 'U' ONTO 'L' TAPE                         MON40900   66
      WRITE(09) ((AU(I,J),I=1,NREC1),J=1,KBL)                   MON40910   74
  300 CONTINUE                                                  MON40920
C     DECREMENT RECORD INDEX                                    MON40930
      IUREC = IUREC-1                                           MON40940
C     ARE ALL RECORDS DONE                                      MON40950
      IF(IUREC.GT.0) GO TO 200                                  MON40960
      REWIND 15                                                 MON40970   86
  350 CONTINUE                                                  MON40980
C     SET HARMONIC INDEX FOR FIRST ORDINARY ITERATION           MON40990
      INTF5 = 1                                                 MON41000
      WRITE(6,5003)                                             MON41010   90
      DO 355 I=1,KS                                             MON41020
      WRITE(6,5004) MLIST(I),(STP(J,I),J=1,KRMP1)               MON41030   93
  355 CONTINUE                                                  MON41040
C     ZERO OUT RIGHT HAND VECTOR ARRAY                          MON41050
      DO 370 I=1,2400                                           MON41060
      RHK(I) = 0.0                                              MON41070
  370 CONTINUE                                                  MON41080
C     REWIND 'U' AND 'L' TAPE                                   MON41090
      REWIND 9                                                  MON41100  108
      GO TO 1000                                                MON41110
  400 CONTINUE                                                  MON41120
C     SET FLAGS FOR MATRIX MULTIPLICATION                       MON41130
```

```
C     GET ST VECTOR FOR THIS HARMONIC                                  MON41140
      IJH = INTF5                                                      MON41150
      DO 410 I=1,KRMP1                                                 MON41160
  410 ST(I) = STP(I,IJH)                                               MON41170
C     CALCULATE 'P' VECTOR                                             MON41180
      CALL PELR(AL,NREC,RHV,RHK,P,NROWL,MAXREC)                        MON41190   120
  450 CONTINUE                                                         MON41200
C     IF ICHECK = 0, SKIP CHECKOUT PRINTING OPTIONS                    MON41210
      IF(ICHECK.EQ.0) GO TO 500                                        MON41220
      IF(IOPT(6).EQ.0) GO TO 500                                       MON41230
      IF(INTF2) 480,480,460                                            MON41240
  460 CONTINUE                                                         MON41250
      CALL PRNX(RHM,NPSEG,LSP,NSGMTS)                                  MON41260   130
      GO TO 500                                                        MON41270
  480 CONTINUE                                                         MON41280
      CALL PRNX(RHK,NPSEG,LSP,NSGMTS)                                  MON41290   133
  500 CONTINUE                                                         MON41300
C     PERFORM BACK SUBSTITUTION                                        MON41310
      CALL BSUB(AU,AX,P,RHK,RHM,NREC,NROWS,MAXREC)                     MON41320   135
      IF(IJH.EQ.KS) REWIND 9                                           MON41330   137
      REWIND 15                                                        MON41340   138
      IF(INTF2.GT.0) GO TO 700                                         MON41350
      TOOBIG = 1.0E 2                                                  MON41360
C     CHECK FOR DIVERGENCE ON  U, V, W, AND OMEGA(THETA)               MON41370
      DO 550 I=1,NPTS                                                  MON41380
      KK = 8 * (I-1)                                                   MON41390
      DO 550 IZ = 1,4                                                  MON41400
      KK = KK + 1                                                      MON41410
      IF(ABS(RHK(KK)) - TOOBIG) 550,550,520                            MON41420
C     DISPLACEMENTS ARE ''TOOBIG'' ... TERMINATE CASE                  MON41430
  520 WRITE(6,5050)                                                    MON41440   151
      WRITE(6,5060) MLIST(IJH)                                         MON41450   152
      CALL PRNX(RHK,NPSEG,LSP,NSGMTS)                                  MON41460   154
      KKKKK = 8                                                        MON41470
      RETURN                                                           MON41480
  550 CONTINUE                                                         MON41490
  700 CONTINUE                                                         MON41500
C     IF ICHECK = 0, SKIP CHECKOUT PRINTING OPTIONS                    MON41510
```

```
      IF(ICHECK.EQ.0) GO TO 1000                                MON41520
      IF(IOPT(14).EQ.0) GO TO 1000                              MON41530
      IF(IOPT(14)-2) 1000,750,750                               MON41540
  750 CONTINUE                                                  MON41550
      IF(INTF2) 900,900,800                                     MON41560
  800 CONTINUE                                                  MON41570
      CALL PRNX(RHM,NPSEG,LSP,NSGMTS)                           MON41580    172
      GO TO 1000                                                MON41590
  900 CONTINUE                                                  MON41600
      CALL PRNX(RHK,NPSEG,LSP,NSGMTS)                           MON41610    175
 1000 CONTINUE                                                  MON41620
C     RESTORE INTERNAL FLAGS                                    MON41630
      INTF(02) = INTF2                                          MON41640
C     SET OVERLAY FLAG FOR REGION 5                             MON41650
      KKKKK = 5                                                 MON41660
      RETURN                                                    MON41670
      END                                                       MON41680
```

SUBROUTINE MON4

COMMON VARIABLES

	COMMON BLOCK		COMAA	ORIGIN	00001		LENGTH	00001
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION
ICHECK	00000	I						
	COMMON BLOCK		COM41	ORIGIN	00002		LENGTH	00013
KRMP1	00000	I	KRMP2	00001	I		MRT	00002
KBL	00003	I	IPTBL	00004	I			
	COMMON BLOCK		COM42	ORIGIN	00015		LENGTH	00112
AB	00000	R	U	00027	R		ST	00056
AL1	00072	R	R	00107	R		X	00110
C	00111	R						
	COMMON BLOCK		COM43	ORIGIN	00127		LENGTH	00003
DTM	00000	D	DTM1	00002	R			
	COMMON BLOCK		COM44	ORIGIN	00132		LENGTH	00076
CRT	00000	R	ALT	00014	D		CT	00046
	COMMON BLOCK		COM45	ORIGIN	00230		LENGTH	00001
INTF2	00000	I						
	COMMON BLOCK		COM49	ORIGIN	00231		LENGTH	00001
IOPT5	00000	I						
	COMMON BLOCK		//	ORIGIN	00232		LENGTH	33167
NSGMTS	00000	I	KS	00001	I		ISGMT	00002
IFTNO	00003	I	LOADNO	00004	I		LDNOPR	00005
NLOAD	00006	I	KKKKK	00007	I		PHI	00010
SINP	00017	R	COSP	00026	R		NPSEG	00035
LSP	00044	I	MLIST	00053	I		ECR	00062
ECY	00071	R	IOPT	00100	I		IRES	00116

NREC	00130	I	INTF	00155	I	NSGPL	00167
NPTPT	00201	I	NLDPL	00213	I	EI	00225
HI	00243	R	FMUI	00261	R	RI	00277
FLDI	00306	R	FLDF	00673	R	DLDI	01260
DLDF	01645	R	FMPO	02232	R	FNRO	02374
FNYO	02536	R	FNPTO	02700	R	ECRS	03042
ECYS	03123	R	RSS	03204	R	SCF	03265
IBC	04705	I	BC	04775	R	DLBC	05065
S	05155	R	FNTH	05631	R	FNPH	06305
DPHI	06761	R	DTHE	07435	R	STP	10111
AK	10235	R	AU	16461	R	AL	24705
RHV	31755	R	SAVE	33105	R	INTF	00155
INTF1	00155	I	INTF4	00160	I	INTF5	00161
RHK	10235	R	RHM	24705	R	RHV	31755
P	32265	R	AX	31755	R	RHV	31755

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
TC	33421	D					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
N12	34471	I	NROWS	34472	I	NROWL	34473
NCOLL	34474	I	MAXMPT	34475	I	NPTS	34476
IREC	34477	I	MAXREC	34500	I	IJH	34501
KS1	34502	I	I	34503	I	IUREC	34504
NREC1	34505	I	J	34506	I	TOOBIG	34507
KK	34510	I					

ENTRY POINTS

MON4 SECTION 18

SUBROUTINES CALLED

PRMY	SECTION	19	LANU	SECTION	20	.FRWT.	SECTION	
.FRST.	SECTION	22	.FRDB.	SECTION	23	.FWKB.	SECTION	
.FWRD.	SECTION	25	PELR	SECTION	26	PRNX	SECTION	
BSUB	SECTION	28	.UN14.	SECTION	29	.UN15.	SECTION	
.FRLR.	SECTION	31	.FRLT.	SECTION	32	.FBDT.	SECTION	
.UN09.	SECTION	34	.FWLR.	SECTION	35	.UN06.	SECTION	
.FFIL.	SECTION	37	.FCNV.	SECTION	38	SYSLOC	SECTION	

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
5003	FORMAT	34536	5004	FORMAT	34542	5010	FORMAT	
5050	FORMAT	34557	5060	FORMAT	34604	50	16A	
100	25A	34702	400	110A	35231	110	41A	
105	33A	34715	107	36A	34722	108	39A	
120	42A	34745	350	87A	35151	130	53A	
150	57A	35022	450	121A	35262	200	61A	
250	64A	35044	300	82A	35135	355	99A	
370	106A	35222	1000	176A	35477	410	115A	
500	134A	35314	480	132A	35305	460	129A	
700	161A	35441	550	157A	35435	520	151A	
750	169A	35455	900	174A	35470	800	171A	

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 35527.

```
$IBFTC PRNXX   NOLIST,DECK,M94,XR7,NODD
```

```
C                 PRINT 'X' FOR ALL SEGMENTS                          PRNX0010
      SUBROUTINE PRNX(XVAL,NPSEG,LSP,NSGMTS)                          PRNX0020
      DIMENSION XVAL(    1),NPSEG( 1),LSP( 1)                         PRNX0030
 1001 FORMAT(1H0,7HSEGMENT,I3/7X,1HU,12X,1HV,12X,1HW,9X,5HOMEGA,9X,   PRNX0040
     1 3HNPT,9X,4HNPHI,9X,4HQPHI,9X,4HMPHI)                           PRNX0050
 1002 FORMAT(6H POINT,I4/8(1PE13.5))                                  PRNX0060
C     CALCULATE NUMBER OF POINTS IN SHELL                             PRNX0070
      NPTS = 0                                                        PRNX0080
      DO 50 I=1,NSGMTS                                                PRNX0090
   50 NPTS = NPTS+NPSEG(I)                                            PRNX0100
C     INITIALIZE COUNTERS                                             PRNX0110
      IJK = 1                                                         PRNX0120
      II  = 0                                                         PRNX0130
      IF  = 0                                                         PRNX0140
      MS  = LSP(IJK)                                                  PRNX0150
      DO 200 I=1,NPTS                                                 PRNX0160
      IF(I.NE.MS) GO TO 100                                           PRNX0170
      IJ = 0                                                          PRNX0180
      MF = MS+NPSEG(IJK)-1                                            PRNX0190
      WRITE(6,1001) IJK                                               PRNX0200
      GO TO 150                                                       PRNX0210
  100 CONTINUE                                                        PRNX0220
      IF(I.NE.MF) GO TO 150                                           PRNX0230
      IJK = IJK+1                                                     PRNX0240
      MS  = LSP(IJK)                                                  PRNX0250
  150 CONTINUE                                                        PRNX0260
      IJ = IJ+1                                                       PRNX0270
      II = IF+1                                                       PRNX0280
      IF = IF+8                                                       PRNX0290
      WRITE(6,1002) IJ,(XVAL(J),J=II,IF)                              PRNX0300
  200 CONTINUE                                                        PRNX0310
      RETURN                                                          PRNX0320
      END                                                             PRNX0330
```

```
$IBFTC PRMYX   NOLIST,DECK,M94,XR7,NODD
```

```
C     PRINT NEW RIGHT HAND VECTOR                                    PRMY0010
      SUBROUTINE PRMY(XVAL,LSP,NPSEG,NSGMTS,IJH)                     PRMY0020
      DIMENSION XVAL( 1),LSP( 1),NPSEG( 1)                           PRMY0030
 1000 FORMAT(1H ,6HPOINT ,I2/8(1PE13.5) )                            PRMY0040
 1001 FORMAT(1H1,5X,7HSEGMENT,I2,5X,16HHARMONIC INDEX =,I2)          PRMY0050
C     INITIALIZE FLAGS                                               PRMY0060
      II = 0                                                         PRMY0070
      IF = 0                                                         PRMY0080
C     START LOOP ON SEGMENTS                                         PRMY0090
      DO 400 IJK = 1,NSGMTS                                          PRMY0100
      IJ = 1                                                         PRMY0110
      II = IF+1                                                      PRMY0120
      IF = IF+4                                                      PRMY0130
      NPT1 = NPSEG(IJK) - 1                                          PRMY0140
      WRITE(6,1001) IJK,IJH                                          PRMY0150
C     PRINT LEADING DISCONTINUITY EQN. VALUES OR BOUNDARY COND. VALUES PRMY0160   11
      WRITE(6,1000) IJ,(XVAL(J),J=II,IF)                             PRMY0170   12
      DO 300 IJ = 1,NPT1                                             PRMY0180
      II = IF+1                                                      PRMY0190
      IF = IF+8                                                      PRMY0200
C     PRINT INTERIOR POINT VALUES                                    PRMY0210
      WRITE(6,1000) IJ,(XVAL(J),J=II,IF)                             PRMY0220   20
  300 CONTINUE                                                       PRMY0230
      II = IF+1                                                      PRMY0240
      IF = IF+4                                                      PRMY0250
      IJ = IJ+1                                                      PRMY0260
C     PRINT ENDING DISCONTINUITY EQN. VALUES OR BOUNDARY COND. VALUES PRMY0270
      WRITE(6,1000) IJ,(XVAL(J),J=II,IF)                             PRMY0280   29
  400 CONTINUE                                                       PRMY0290
      RETURN                                                         PRMY0300
      END                                                            PRMY0310
```

```
$IBFTC LAVUX    NOLIST,DECK,M94,XR7,NODD
```

```
C                 CALCULATE 'L' AND 'U' MATRICES FOR ONE HARMONIC          LANU0010
      SUBROUTINE LANU(AK,AU,AL,RHV,P,TC,NREC,LSP,NPSEG,RHK,RHM,NROWS,      LANU0020
     1 NROWL,N12)                                                          LANU0030
C     INTF2 = 0--NOT A M=0 ITERATION.  RIGHT-HAND VECTOR STORED IN RHK     LANU0040
C              --ALSO, 'P' VECTOR NOT DESIRED                              LANU0050
C     INTF2 = 1--IS A  M=0 ITERATION,  RIGHT-HAND VECTOR STORED IN RHM     LANU0060
C              --'L' MATRIX NOT DESIRED, 'P' VECTOR IS DESIRED             LANU0070
C     INTF2 = 2 OR GREATER, SAME AS INTF2 = 1.                             LANU0080
      DIMENSION AK(NROWS, 1),AU(NROWS, 1),AL(NROWL, 1),RHV( 1),P( 1),      LANU0090
     1 TC(N12, 1),NREC( 1),LSP(1),NPSEG(1),RHK( 1),RHM( 1),PDAT(3)         LANU0100
      DOUBLE PRECISION TC,CT,ALT,DTM                                       LANU0110
      EQUIVALENCE(IPTBL(1),IPT1)                                           LANU0120
      COMMON/COMAA/ICHECK                                                  LANU0130
      COMMON/COM41/KRMP1,KRMP2,MRT,KBL,IPTBL(7)                            LANU0140
      COMMON/COM42/AB(23),U(23),ST(12),AL1(13),R(1),X(1),C(1)              LANU0150
      COMMON/COM43/ DTM,DTM1                                               LANU0160
      COMMON/COM44/CRT(12,1),ALT(13),CT(12)                                LANU0170
      COMMON/COM45/INTF2                                                   LANU0180
      COMMON/COM49/IOPTS                                                   LANU0190
      DATA PDAT(1),PDAT(2),PDAT(3)/6H'L' MA,5HTRIX ,6H'U' MA/              LANU0200
 4001 FORMAT(1H0,1X,1PE13.5,7E13.5,/2X,8E13.5/2X,7E13.5)                   LANU0210
 4002 FORMAT(8HOSEGMENT,I2,1X,2A6,5HPOINT,I3)                              LANU0220
      MONE=1                                                               LANU0230
      IPTBL(7) = 1                                                         LANU0240
      IF(INTF2.GT.0) IPTBL(7)=0                                            LANU0250
      INDR = 0                                                             LANU0260
      INDP = 0                                                             LANU0270
      MXNRKR = (IPTBL(2)-8)/(NROWS-4)+1                                    LANU0280
C     INITIALIZE MATRIX TRIANGULARIZATION SUBROUTINE                       LANU0290
      CALL MABT(AB,U,KBL,C,CC,MONE,TC,CT,ST,KRMP1,AL1,ALT,KRMP2,DTM,       LANU0300
     1 DTM1,IPTBL)                                                         LANU0310
C     INITIALIZE RIGHT HAND VECTOR                                         LANU0320   8
      DO 30 I=1,NROWS                                                      LANU0330
   30 RHV(I) = 0.                                                          LANU0340
C     SET RECORD AND ROW COUNTERS                                          LANU0350
      NRKR = 0                                                             LANU0360
      NRUL = 1                                                             LANU0370
```

```
      NCUL = 0                                                          LANU0380
      IF(IOPT5.EQ.0) GO TO 60                                           LANU0390
C     INITIALIZE FLAGS FOR PRINTING 'U' AND 'L' MATRICES                LANU0400
      IJK = 1                                                           LANU0410
      MS=LSP(IJK)                                                       LANU0420
      MF=MS+NPSEG(IJK)-1                                                LANU0430
      KK  = 0                                                           LANU0440
      MXKK=12                                                           LANU0450
      LPDI= 0                                                           LANU0460
      LPDF= 0                                                           LANU0470
      IJ  = 1                                                           LANU0480
      IJJ = 1                                                           LANU0490
      GO TO 60                                                          LANU0500
   50 CONTINUE                                                          LANU0510
      IF(NCK.LT.NREC(NRKR)) GO TO 100                                   LANU0520
   60 CONTINUE                                                          LANU0530
C     SET COUNTERS FOR NEW RECORD                                       LANU0540
      NRKR = NRKR+1                                                     LANU0550
      NCK  = 0                                                          LANU0560
      NREC1 =NREC(NRKR)                                                 LANU0570
C     IF NRKR.GT.MXNRKR, ALL OF RECORDS HAVE BEEN READ                  LANU0580
      IF(NRKR.GT.MXNRKR) GO TO 100                                      LANU0590
C     READ 'K' MATRIX                                                   LANU0600
      READ(14) ((AK(I,J),I=1,NREC1),J=1,KBL)                            LANU0610
C     READ RIGHT HAND VECTOR                                            LANU0620   46
      IF(INTF2.GT.0) GO TO  90                                          LANU0630
C     GET RHV FROM RHK                                                  LANU0640
      DO 80 I=1,NREC1                                                   LANU0650
      INDR = INDR+1                                                     LANU0660
      RHV(I) = RHK(INDR)                                                LANU0670
   80 CONTINUE                                                          LANU0680
      GO TO 100                                                         LANU0690
   90 CONTINUE                                                          LANU0700
C     THIS IS A SPECIAL M=0 ITERATION                                   LANU0710
      DO 95 I=1,NREC1                                                   LANU0720
      INDR = INDR+1                                                     LANU0730
      RHV(I) = RHM(INDR)                                                LANU0740
   95 CONTINUE                                                          LANU0750
```

```
  100 CONTINUE                                                          LANU0760
      IF(IPT1.GT.1) GO TO 150                                           LANU0770
C     INCREMENT 'K' AND 'RHV' COUNTER                                   LANU0780
      NCK = NCK+1                                                       LANU0790
C     INPUT NEW ROW OF 'K' AND 'RHV'                                    LANU0800
      DO 110 I=1,KBL                                                    LANU0810
  110 AB(I) = AK(NCK,I)                                                 LANU0820
      C(1) = RHV(NCK)                                                   LANU0830
  150 CONTINUE                                                          LANU0840
      IF(IPT1.EQ.0) GO TO 200                                           LANU0850
C     INCREMENT COUNTER FOR 'U' AND 'L'                                 LANU0860
      NCUL = NCUL+1                                                     LANU0870
      KK=KK+1                                                           LANU0880
      IF(INTF2.GT.C) GO TO 175                                          LANU0890
C     GET ROW OF 'L' MATRIX                                             LANU0900
      DO 170 I=1,KRMP2                                                  LANU0910
  170 AL(NCUL,I) = AL1(I)                                               LANU0920
      GO TO 178                                                         LANU0930
  175 CONTINUE                                                          LANU0940
C     GET VALUE OF 'P' VECTOR                                           LANU0950
      P(NCUL) = CC                                                      LANU0960
  178 CONTINUE                                                          LANU0970
C     GET ROW OF 'U' MATRIX                                             LANU0980
      DO 180 I=1,KBL                                                    LANU0990
  180 AU(NCUL,I) = U(I)                                                 LANU1000
  200 CONTINUE                                                          LANU1010
C     IF ICHECK = 0, SKIP CHECKOUT PRINTING OPTIONS                     LANU1020
      IF(ICHECK.EQ.C) GO TO 300                                         LANU1030
      IF(IOPT5.EQ.0) GO TO 300                                          LANU1040
C     PRINT 'U' AND 'L' MATRICES                                        LANU1050
      IF(KK.LT.MXKK) GO TO 300                                          LANU1060
      LROI=LROF+1                                                       LANU1070
      LROF=LROF+MXKK                                                    LANU1080
      IF(IOPT5.EQ.1) GO TO 220                                          LANU1090
C     PRINT 'U' MATRIX                                                  LANU1100
      WRITE(6,4002) IJK,PDAT(3),PDAT(2),IJJ                             LANU1110    137
      DO 210 I=LROI,LROF                                                LANU1120
  210 WRITE(6,4001) (AU(I,J),J=1,KBL)                                   LANU1130    141
```

```
  220 CONTINUE                                                    LANU1140
      IF(IOPT5.EQ.2) GO TO 240                                    LANU1150
C     PRINT 'L' MATRIX                                            LANU1160
      WRITE(6,4002) IJK,PDAT(1),PDAT(2),IJJ                       LANU1170    151
      DO 230 I=LRCI,LROF                                          LANU1180
  230 WRITE(6,4001) (AL(I,J),J=1,KRMP2)                           LANU1190    155
  240 CONTINUE                                                    LANU1200
      IF(IPT1.EQ.3) GO TO 300                                     LANU1210
C     SET PRINT FLAGS FOR NEXT POINT                              LANU1220
      IJ=IJ+1                                                     LANU1230
      IJJ = IJJ+1                                                 LANU1240
      MXKK=8                                                      LANU1250
      KK  =0                                                      LANU1260
      IF(IJ.NE.MS) GO TO 250                                      LANU1270
      MXKK=MXKK+4                                                 LANU1280
      MF  = MS+NPSEG(IJK)-1                                       LANU1290
      IJJ = 1                                                     LANU1300
  250 CONTINUE                                                    LANU1310
      IF(IJ.NE.MF) GO TO 260                                      LANU1320
      MXKK=MXKK-4                                                 LANU1330
      IJK = IJK+1                                                 LANU1340
      MS  = LSP(IJK)                                              LANU1350
  260 CONTINUE                                                    LANU1360
  300 CONTINUE                                                    LANU1370
      IF(NCUL.LT.NREC(NRUL)) GO TO 350                            LANU1380
      NREC1=NREC(NRUL)                                            LANU1390
C     WRITE RECORD OF 'U'                                         LANU1400
      WRITE(15) ((AU(I,J),I=1,NREC1),J=1,KBL)                     LANU1410    190
      IF(INTF2.GT.0) GO TO 320                                    LANU1420
C     WRITE RECORD OF 'L'                                         LANU1430
      WRITE( 9) ((AL(I,J),I=1,NREC1),J=1,KRMP2)                   LANU1440    202
      GO TO 340                                                   LANU1450
  320 CONTINUE                                                    LANU1460
C     SAVE 'P' VECTOR                                             LANU1470
      DO 330 I=1,NREC1                                            LANU1480
      INDP = INDP+1                                               LANU1490
      RHM(INDP) = P(I)                                            LANU1500
  330 CONTINUE                                                    LANU1510
```

```
  340 CONTINUE                                                          LANU1520
C     INCREMENT RECORD COUNTER AND INITIALIZE 'U' AND 'L' ROW COUNTER   LANU1530
      NRUL = NRUL+1                                                     LANU1540
      NCUL = 0                                                          LANU1550
      LROI=0                                                            LANU1560
      LROF=0                                                            LANU1570
  350 CONTINUE                                                          LANU1580
      IF(IPT1.EQ.3) GO TO 400                                           LANU1590
C     INTERACT WITH MATRIX TRIANGULARIZER                               LANU1600
      CALL M&BTU                                                        LANU1610    231
      GO TO 50                                                          LANU1620
  400 CONTINUE                                                          LANU1630
      RETURN                                                            LANU1640
      END                                                               LANU1650
```

```
$IBFTC PELRX   NOLIST,DECK,M94,XR7,NODD
```

```
C                 'P' VECTOR = 'L' MATRIX TIMES 'RHV'                   PELR0010
      SUBROUTINE PELR(AL,NREC,RHV,RHK,P,NROWL,MAXREC)                   PELR0020
      DOUBLE PRECISION CT,ALT                                           PELR0030
      DIMENSION AL(NROWL, 1),NREC( 1),RHV( 1),RHK( 1),P( 1)             PELR0040
      DIMENSION ALRT(1)                                                 PELR0050
      EQUIVALENCE(ALRT(1),C(1)),(IPTBL(1),IPT1)                         PELR0060
      COMMON/COM41/KRMP1,KRMP2,MRT,KBL,IPTBL(7)                         PELR0070
      COMMON/COM42/AB(23),J(23),ST(12),AL1(13),R(1),X(1),C(1)           PELR0080
      COMMON/COM44/CRT(12,1),ALT(13),CT(12)                             PELR0090
C     INITIALIZE FLAGS                                                  PELR0100
C     INDP   - POINT INDEX FOR 'P', ENTIRE SHELL                        PELR0110
      INDP = 0                                                          PELR0120
C     IRHV   - POINT INDEX FOR 'R', EACH RECORD                         PELR0130
      MRNR  = 0                                                         PELR0140
      NREC2 = NREC(1)                                                   PELR0150
      IRHV  = NREC2                                                     PELR0160
C     INP    - POINT INDEX FOR 'P', EACH RECORD                         PELR0170
      INP   = 0                                                         PELR0180
C     IRHK   - POINT INDEX FOR 'R', ENTIRE SHELL                        PELR0190
      IRHK = 0                                                          PELR0200
C     MRNL   - RECORD INDEX FOR 'L'                                     PELR0210
      MRNL = 0                                                          PELR0220
C     MRNP   - RECORD INDEX FOR 'P'                                     PELR0230
      MRNP = 1                                                          PELR0240
C     INITIALIZE MATRIX MULTIPLIER                                      PELR0250
      CALL MABL( R,ALRT,MRT,CRT,ST,KRMP1,AL1,KRMP2,IPTBL)               PELR0260    10
      GO TO 60                                                          PELR0270
   50 CONTINUE                                                          PELR0280
      IF(INL.LT.NREC(MRNL)) GO TO 70                                    PELR0290
C     READ A RECORD OF 'L'                                              PELR0300
   60 CONTINUE                                                          PELR0310
      MRNL  = MRNL+1                                                    PELR0320
      IF(MRNL.GT.MAXREC) GO TO 70                                       PELR0330
      NREC1 = NREC(MRNL)                                                PELR0340
      READ(09) ((AL(I,J),I=1,NREC1),J=1,KRMP2)                          PELR0350
C     INITIALIZE COUNTER FOR 'L'                                        PELR0360
C     INL    - ROW INDEX FOR 'L', EACH RECORD                           PELR0370    23
```

```
      INL  = 0                                                          PELR0380
   70 CONTINUE                                                          PELR0390
      IF(IRHV.LT.NREC2) GO TO 90                                        PELR0400
      MRNR  = MRNR + 1                                                  PELR0410
      IF(MRNR.GT.MAXREC) GO TO 90                                       PELR0420
      NREC2 = NREC(MRNR)                                                PELR0430
C     READ NEW 'RECORD' OF RHV                                          PELR0440
      DO 80 I=1,NREC2                                                   PELR0450
      IRHK=IRHK+1                                                       PELR0460
   80 RHV(I) = RHK(IRHK)                                                PELR0470
C     INITIALIZE POINT INDEX FOR NEW RECORD OF 'R' OR 'RHV'             PELR0480
      IRHV = 0                                                          PELR0490
   90 CONTINUE                                                          PELR0500
      IF(IPT1.GT.2) GO TO 100                                           PELR0510
      IRHV = IRHV+1                                                     PELR0520
C     INPUT NEW VALUE OF RIGHT HAND VECTOR                              PELR0530
      R(01) = RHV(IRHV)                                                 PELR0540
  100 CONTINUE                                                          PELR0550
      IF((IPT1.EQ.0).OR.(IPT1.EQ.4)) GO TO 150                          PELR0560
C     INCREMENT COUNTER FOR 'L' MATRIX                                  PELR0570
      INL =INL+1                                                        PELR0580
C     INPUT NEW ROW OF 'L' MATRIX                                       PELR0590
      DO 110 I=1,KRMP2                                                  PELR0600
  110 AL1(I) = AL(INL,I)                                                PELR0610
  150 CONTINUE                                                          PELR0620
      IF(IPT1.LT.2) GO TO 200                                           PELR0630
C     INCREMENT COUNTER FOR 'P'                                         PELR0640
      INP = INP+1                                                       PELR0650
C     GET NEW VALUE OF 'P'                                              PELR0660
      P(INP) = C(1)                                                     PELR0670
  200 CONTINUE                                                          PELR0680
C     SAVE RECORD OF 'P'                                                PELR0690
      IF(INP.LT.NREC(MRNP)) GO TO 350                                   PELR0700
      NREC1=NREC(MRNP)                                                  PELR0710
      DO 320 I=1,NREC1                                                  PELR0720
      INDP = INDP+1                                                     PELR0730
  320 PHK(INDP) = P(I)                                                  PELR0740
C     INITIALIZE COUNTER FOR 'P'                                        PELR0750
```

```
      INP = 0                                                   PELR0760
      MRNP = MRNP+1                                             PELR0770
  350 CONTINUE                                                  PELR0780
      IF(IPT1.EQ.4) GO TO 500                                   PELR0790
C     INTERACT WITH MATRIX MULTIPLIER                           PELR0800
      CALL MABLU                                                PELR0810    96
      GO TO 50                                                  PELR0820
  500 CONTINUE                                                  PELR0830
      RETURN                                                    PELR0840
      END                                                       PELR0850
```

```
$IBFTC BSJBX   NOLIST,DECK,M94,XR7,NODD
```

```
C                    PERFORM BACK-SUBSTITION FOR ONE HARMONIC          BSUB0010
      SUBROUTINE BSUB(AU,AX,P,RHK,RHM,NREC,NROWS,MAXR)                BSUB0020
      DOUBLE PRECISION CT,ALT                                         BSUB0030
      DIMENSION AU(NROWS, 1),AX( 1),P( 1),RHK( 1),RHM( 1),NREC( 1)    BSUB0040
      DIMENSION XIN(23,1)                                             BSUB0050
      EQUIVALENCE(IPTBL(1),IPT1),(IPTBL(2),IPT2),(IPTBL(3),IPT3)      BSUB0060
      EQUIVALENCE(XIN(1,1),AB(1)),(MRT,MST)                           BSUB0070
      COMMON/COM41/KRMP1,KRMP2,MRT,KBL,IPTBL(7)                       BSUB0080
      COMMON/COM42/AB(23),J(23),ST(12),AL1(13),R(1),X(1),C(1)         BSUB0090
      COMMON/COM44/CRT(12,1),ALT(13),CT(12)                           BSUB0100
      COMMON/COM45/INTF2                                              BSUB0110
C     INITIALIZE FLAGS AND ARGUMENTS FOR MABB                         BSUB0120
      ICX  = IPT2+1                                                   BSUB0130
      IRHK = IPT2+1                                                   BSUB0140
      IRECU= MAXR+1                                                   BSUB0150
      IAX  = NREC(MAXR)                                               BSUB0160
      NRECX= IAX                                                      BSUB0170
      IRX  = MAXR                                                     BSUB0180
C     INITIALIZE BACK-SUBSTITUTION ROUTINE                            BSUB0190
      CALL MABB(U,KBL,R,X,XIN,MST,IPTBL)                              BSUB0200   9
      GO TO 60                                                        BSUB0210
   50 CONTINUE                                                        BSUB0220
      IF(ICU.GT.0) GO TO 250                                          BSUB0230
   60 CONTINUE                                                        BSUB0240
      IRECU = IRECU-1                                                 BSUB0250
      IF(IRECU.LE.0) GO TO 250                                        BSUB0260
      NRECU = NREC(IRECU)                                             BSUB0270
      IF(INTF2.GT.0) GO TO 100                                        BSUB0280
C     IF NOT A SPECIAL ITERATION, READ 'U' FROM TAPE 9                BSUB0290
      READ(9) ((AU(I,J),I=1,NRECU),J=1,KBL)                           BSUB0300   24
      GO TO 150                                                       BSUB0310
  100 CONTINUE                                                        BSUB0320
C     IF IT IS A SPECIAL ITERATION, READ 'U' FROM TAPE 15             BSUB0330
      BACKSPACE 15                                                    BSUB0340   35
      IF(IRECU.NE.MAXR) BACKSPACE 15                                  BSUB0350   36
      READ(15) ((AU(I,J),I=1,NRECU),J=1,KBL)                          BSUB0360   37
  150 CONTINUE                                                        BSUB0370
```

```
C     INITIALIZE COUNTER FOR 'U'                                BSUB0380
      ICU = NRECU                                               BSUB0390
C     IS THIS A SPECIAL ITERATION                               BSUB0400
      IF(INTF2.GT.0) GO TO 200                                  BSUB0410
C     NO--GET 'P' FROM 'RHK'                                    BSUB0420
      DO 175 I=1,NRECU                                          BSUB0430
      IRHK = IRHK-1                                             BSUB0440
      J=NRECU-I+1                                               BSUB0450
      P(J) = RHK(IRHK)                                          BSUB0460
  175 CONTINUE                                                  BSUB0470
      GO TO 230                                                 BSUB0480
  200 CONTINUE                                                  BSUB0490
C     YES--GET 'P' FROM 'RHM'                                   BSUB0500
      DO 225 I=1,NRECU                                          BSUB0510
      IRHK = IRHK-1                                             BSUB0520
      J=NRECU-I+1                                               BSUB0530
      P(J) = RHM(IRHK)                                          BSUB0540
  225 CONTINUE                                                  BSUB0550
  230 CONTINUE                                                  BSUB0560
  250 CONTINUE                                                  BSUB0570
      IF(IPT1.NE.0) GO TO 300                                   BSUB0580
C     PUT NEW ROW OF 'U' INTO BUFFER                            BSUB0590
      DO 260 I=1,KBL                                            BSUB0600
      U(I) = AU(ICU,I)                                          BSUB0610
  260 CONTINUE                                                  BSUB0620
C     PUT NEW VALUE OF 'P' INTO BUFFER                          BSUB0630
      R(1) = P(ICU)                                             BSUB0640
C     DECREMENT COUNTER FOR 'P' AND 'U'                         BSUB0650
      ICU = ICU-1                                               BSUB0660
  300 CONTINUE                                                  BSUB0670
C     INTERACT WITH BACK-SUBSTITUTION SUBROUTINE                BSUB0680
      CALL MABBU                                                BSUB0690
C     GET VALUE OF 'X' FROM BUFFER                              BSUB0700    89
      AX(IAX) = X(1)                                            BSUB0710
C     DECREMENT 'X' COUNTER                                     BSUB0720
      IAX = IAX-1                                               BSUB0730
C     SAVE A 'RECORD' OF 'X'                                    BSUB0740
      IF(IAX.GT.0) GO TO 400                                    BSUB0750
```

```
C     IS THIS A SPECIAL ITERATION                                BSUB0760
      IF(INTF2.GT.0) GO TO 340                                   BSUB0770
C     NO--SAVE 'X' IN 'RHK'                                      BSUB0780
      DO 320 I=1,NRECX                                           BSUB0790
      ICX=ICX-1                                                  BSUB0800
      J  = NRECX-I+1                                             BSUB0810
      RHK(ICX) = AX(J)                                           BSUB0820
  320 CONTINUE                                                   BSUB0830
      GO TO 370                                                  BSUB0840
  340 CONTINUE                                                   BSUB0850
C     YES--SAVE 'X' IN 'RHM'                                     BSUB0860
      DO 360 I=1,NRECX                                           BSUB0870
      ICX = ICX-1                                                BSUB0880
      J   = NRECX-I+1                                            BSUB0890
      RHM(ICX) = AX(J)                                           BSUB0900
  360 CONTINUE                                                   BSUB0910
  370 CONTINUE                                                   BSUB0920
C     DECREMENT RECORD COUNTER                                   BSUB0930
      IRX = IRX-1                                                BSUB0940
      NRECX=NREC(IRX)                                            BSUB0950
      IAX = NRECX                                                BSUB0960
  400 CONTINUE                                                   BSUB0970
C     THIS HARMONIC DONE                                         BSUB0980
      IF(IPT1.EQ.1) GO TO 450                                    BSUB0990
      GO TO 50                                                   BSUB1000
  450 CONTINUE                                                   BSUB1010
      RETURN                                                     BSUB1020
      END                                                        BSUB1030
```

```
$IBFTC MABTX    NOLIST,DECK,M94,XR7,NODD
```

```
      SUBROUTINE MABT(AB,U,KBL,C,CC,M,TC,CT,ST,KRMP1,AL,ALT,KRMP2,DTM,   MABT0000
     *DTM1,IPTBL)                                                        MABT0001
C     MABT IS A TRIANGULARIZATION SUBROUTINE FOR ALMOST BAND MATRICES.   MABT0002
C     LAST MODIFICATION 7/31/67-K.E.M.                                   MABT0003
C     IPTBL(1)=0 FOR INPUT NEW ROWS ONLY (SET INTERNIALLY).              MABT0004
C             =1 FOR OUTPUT COMPUTED ROWS AND INPUT NEW ROWS (S.I.).     MABT0005
C             =2 FOR OUTPUT COMPUTED ROWS ONLY (S.I.).                   MABT0006
C             =3 FOR OUTPUT COMPUTED ROWS AND TERMINATE (S.I.).          MABT0007
C         (2)=N.                                                         MABT0008
C         (3)=THE ACTUAL ROW OF U,CC AND AL BEING OUTPUT (S.I.)          MABT0009
C         (4)=KRM.                                                       MABT0010
C         (5)=IX, THE POWER OF 2 BY WHICH DTM MUST BE MULTIPLIED (S.I.)MABT0011
C         (6)=IX1, THE POWER OF 2 BY WHICH DTM1 MUST BE MULTIPLIED       MABT0012
C             (S.I.).                                                    MABT0013
C         (7)=0 IF AL IS NOT TO BE OUTPUT.                               MABT0014
C             =1 IF AL IS TO BE OUTPUT.                                  MABT0015
      DOUBLE PRECISION CT,TC,ALT,DTM,DTM2,EPS,TI,TS                      MABT0016
      DOUBLE PRECISION TDTM                                              MABT0017
      DIMENSION AB(KBL),C(M),TC(KRMP1,KBL),CT(KRMP1,M),ST(KRMP1),        MABT0018
     *ALT(KRMP2),AL(KRMP2),J(KBL),CC(M),EPS(5),IPS(5)                    MABT0019
      DIMENSION IPTBL(7)                                                 MABT0020
C     INITIALIZE                                                         MABT0021
 10   IB=0                                                               MABT0022
      IT=0                                                               MABT0023
      DTM=0.0D0                                                          MABT0024
      TDTM=1.0D0                                                         MABT0025
      IX=0                                                               MABT0026
      KBL=KBL                                                            MABT0027
      MM=M                                                               MABT0028
      N=IPTBL(2)                                                         MABT0029
      KRM=IPTBL(4)                                                       MABT0030
      KRC=KRMP1                                                          MABT0031
      KM=KRC                                                             MABT0032
      KMP=KRMP2                                                          MABT0033
      ISAVL=IPTBL(7)                                                     MABT0034
      DO 12 I=1,5                                                        MABT0035
      EPS(I)=134217728.0D0                                               MABT0036
```

```
 12   IPS(I)=0                                                  MABT0037
      IPTBL=0                                                   MABT0038
      JPTBL=0                                                   MABT0039
      RETURN                                                    MABT0040
      ENTRY MABTU                                               MABT0041
C     USE                                                       MABT0042
C     GET AND SCALE ROW                                         MABT0043
      IF(JPTBL-3) 100,480,480                                   MABT0044
 100  IB=IB+1                                                   MABT0045
      IF(IB-N) 102,102,122                                      MABT0046
 102  IF(IB-KRC) 104,105,106                                    MABT0047
C     INITIAL STATE                                             MABT0048
 104  I=IB                                                      MABT0049
      JC=KMP-I                                                  MABT0050
      GO TO 108                                                 MABT0051
C     STEADY STATE                                              MABT0052
C     FIND LARGEST COLUMN ELEMENT                               MABT0053
 105  JPTBL=1                                                   MABT0054
 106  I=KRC                                                     MABT0055
      JC=1                                                      MABT0056
      IT=IT+1                                                   MABT0057
 108  STA=0.0                                                   MABT0058
      J=1                                                       MABT0059
      DO 110 JJ=JC,KRL                                          MABT0060
      TC(I,J)=AB(JJ)                                            MABT0061
      STA=AMAX1(ABS(AB(JJ)),STA)                                MABT0062
 110  J=J+1                                                     MABT0063
      JK=J-1                                                    MABT0064
      TDTM=TDTM*STA                                             MABT0065
      CALL RDCX(TDTM,IXP)                                       MABT0066
      IX=IX+IXP                                                 MABT0067
C     CALCULATE SCALE FACTOR                                    MABT0068
      ST(I)=1.0/STA                                             MABT0069
C     SCALE ROWS                                                MABT0070
      DO 112 J1=1,JK                                            MABT0071
 112  TC(I,J1)=TC(I,J1)/STA                                     MABT0072
C     FILL ROW WITH TRAILING ZEROS IF NECESSARY                 MABT0073
      IF(JK-KRL) 114,118,118                                    MABT0074
```

```
 114  DO 116 JJ=J,KRL                                          MABT0075
 116  TC(I,JJ)=0.0D0                                           MABT0076
 118  DO 120 J=1,MM                                            MABT0077
 120  CT(I,J)=C(J)/STA                                         MABT0078
      IF(IB-KRM) 454,454,200                                   MABT0079
C     TERMINAL STATE                                           MABT0080
 122  I=KRC                                                    MABT0081
      IT=IT+1                                                  MABT0082
      KRC=KRC-1                                                MABT0083
C     PIVOT                                                    MABT0084
C     SEARCH FOR ROW WITH LARGEST 1ST ELEMENT                  MABT0085
 200  TI=DABS(TC(1,1))                                         MABT0086
      K=1                                                      MABT0087
      IF(KRC-1) 201,201,202                                    MABT0088
 201  ALT(1)=K                                                 MABT0089
      GO TO 210                                                MABT0090
 202  DO 204 KK=2,KRC                                          MABT0091
      IF(TI-DABS(TC(KK,1))) 203,204,204                        MABT0092
 203  K=KK                                                     MABT0093
      TI=DABS(TC(KK,1))                                        MABT0094
 204  CONTINUE                                                 MABT0095
C     ROW K HAS LARGEST 1ST ELEMENT                            MABT0096
      ALT(1)=K                                                 MABT0097
      IF(K-1) 210,210,205                                      MABT0098
 205  TDTM=-TDTM                                               MABT0099
      DO 206 J=1,KRL                                           MABT0100
C     EXCHANGE ROWS                                            MABT0101
      TS=TC(K,J)                                               MABT0102
      TC(K,J)=TC(1,J)                                          MABT0103
 206  TC(1,J)=TS                                               MABT0104
      DO 208 J=1,MM                                            MABT0105
      TS=CT(K,J)                                               MABT0106
      CT(K,J)=CT(1,J)                                          MABT0107
 208  CT(1,J)=TS                                               MABT0108
 210  ALT(2)=ST(KM)                                            MABT0109
C     CHECK NEW ROW FOR SMALLEST ELEMENT FOUND                 MABT0110
      IF(TI-EPS(5)) 212,222,222                                MABT0111
 212  DO 214 J=1,4                                             MABT0112
```

```
      IF(TI-EPS(J)) 216,214,214                                         MABT0113
  214 CONTINUE                                                          MABT0114
      J=5                                                               MABT0115
      GO TO 220                                                         MABT0116
C     NEW ROW HAS A SMALLEST ELEMENT, PLACE IN ASCENDING ORDER AMONG THEMABT0117
C       FIVE SMALLEST ELEMENTS FOUND.                                   MABT0118
  216 K=5                                                               MABT0119
  218 EPS(K)=EPS(K-1)                                                   MABT0120
      IPS(K)=IPS(K-1)                                                   MABT0121
      K=K-1                                                             MABT0122
      IF(K-J) 220,220,218                                               MABT0123
  220 EPS(J)=TI                                                         MABT0124
      IPS(J)=IT                                                         MABT0125
  222 IF(TC(1,1)) 224,300,224                                           MABT0126
  224 TDTM=TDTM*TC(1,1)                                                 MABT0127
      IF(KRC-1) 300,300,226                                             MABT0128
C     MORE THAN ONE ROW                                                 MABT0129
  226 DO 234 I=2,KRC                                                    MABT0130
C     FORM LT AND ELIMINATE                                             MABT0131
  228 J=I+1                                                             MABT0132
      ALT(J)=-TC(I,1)/TC(1,1)                                           MABT0133
      IF(ALT(J)) 230,234,230                                            MABT0134
  230 IF(KRL-1) 2305,2315,2305                                          MABT0135
 2305 DO 231 K=2,KRL                                                    MABT0136
  231 TC(I,K)=TC(I,K)+ALT(J)*TC(1,K)                                    MABT0137
 2315 DO 232 K=1,MM                                                     MABT0138
  232 CT(I,K)=CT(I,K)+ALT(J)*CT(1,K)                                    MABT0139
  234 CONTINUE                                                          MABT0140
C     SAVE L IF REQUESTED, OUTPUT A ROW OF L.                           MABT0141
  300 IF(ISAVL) 304,310,304                                             MABT0142
  304 IF(IT-N) 306,306,310                                              MABT0143
  306 DO 308 J=1,KMP                                                    MABT0144
  308 AL(J)=ALT(J)                                                      MABT0145
C     OUTPUT A ROW OF U                                                 MABT0146
  310 DO 312 J=1,KRL                                                    MABT0147
  312 U(J)=TC(1,J)                                                      MABT0148
C     OUTPUT A ROW OF CC                                                MABT0149
      DO 314 J=1,MM                                                     MABT0150
```

```
 314  CC(J)=CT(1,J)                                          MABT0151
 350  IF(KRC-1) 400,400,352                                  MABT0152
 352  IF(KRL-1) 400,400,354                                  MABT0153
C     SHIFT TC AND CT                                        MABT0154
 354  DO 356 I=2,KRC                                         MABT0155
      DO 355 J=2,KRL                                         MABT0156
 355  TC(I-1,J-1)=TC(I,J)                                    MABT0157
      TC(I-1,KRL)=0.0D0                                      MABT0158
      DO 356 J=1,MM                                          MABT0159
 356  CT(I-1,J)=CT(I,J)                                      MABT0160
C     TEST FOR COMPLETION                                    MABT0161
 400  IF(IT-N) 402,460,460                                   MABT0162
 402  IF(IB-N) 454,450,450                                   MABT0163
 450  JPTBL=2                                                MABT0164
 454  IPTBL=JPTBL                                            MABT0165
      IPTBL(3)=IT                                            MABT0166
 455  RETURN                                                 MABT0167
C     TERMINATE                                              MABT0168
 460  IF(EPS) 462,470,462                                    MABT0169
 462  DTM2=TDTM/EPS                                          MABT0170
      CALL RDCX(DTM2,IEXP)                                   MABT0171   248
      DTM1=DTM2                                              MABT0172
      IX1=IEXP+IX                                            MABT0173
      DTM=TDTM                                               MABT0174
      GO TO 475                                              MABT0175
 470  DTM1=TDTM                                              MABT0176
      IX1=IX                                                 MABT0177
      DTM=0.0D0                                              MABT0178
      IX=0                                                   MABT0179
 475  JPTBL=3                                                MABT0180
 480  IPTBL(3)=IT                                            MABT0181
      IPTBL(5)=IX                                            MABT0182
      IPTBL(6)=IX1                                           MABT0183
      IPTBL=3                                                MABT0184
      RETURN                                                 MABT0185
      END                                                    MABT0186
```

```
$IBFTC RDCXX   NOLIST,DECK,M94,XR7,NODD
```

```
      SUBROUTINE RDCX (X,IEXP)                                          RDCX0000
C     LAST MODIFICATION 4/18/67-K.E.M.                                  RDCX0001
C     RDCX REDUCES A DOUBLE PRECISION DECIMAL X TO A DECIMAL WITH       RDCX0002
C          ABSOLUTE VALUE BETWEEN 0.0D0 AND 1.0D0, AND CONVERTS THE     RDCX0003
C          EXTRACTED POWER OF 2 INTO AN INTEGER IEXP.                   RDCX0004
C     IF THE MOST SIGNIFICANT PART OF X IS 0.0, IEXP IS SET EQUAL TO 0.0RDCX0005
C          AND THE REDUCTION IS IGNORED.                                RDCX0006
C     IF THE DECIMAL X IS SINGLE PRECISION AND THE MACHINE IS THE       RDCX0007
C          IBM 7094, THE NEXT CONTIGUOUS WORD WILL BE ALTERED.          RDCX0008
      EQUIVALENCE (A,IA)                                                RDCX0009
      DIMENSION X(2),IA(2),A(2)                                         RDCX0010
      DATA IA/0,0/,MCHECK/6H00000 /                                     RDCX0011
      DATA MASK1/077600000000/                                          RDCX0012
      A=X                                                               RDCX0013
      IEXP=0                                                            RDCX0014
      IF(A) 5,40,5                                                      RDCX0015
    5 IA(2)=ISIGN(1,IA)                                                 RDCX0016
      IF(MCHECK-16) 30,10,30                                            RDCX0017
C     GE 635 MACHINE.                                                   RDCX0018
   10 IEXP=AND(IA,MASK1)                                                RDCX0019
      IA=IA-IEXP                                                        RDCX0020
      X=A                                                               RDCX0021
      IEXP=((IEXP*IA(2))/268435456)*IA(2)                               RDCX0022
      RETURN                                                            RDCX0023
C     IBM 7094 MACHINE.                                                 RDCX0024
   30 IEXP=((IA(2)*IA)/134217728)-128                                   RDCX0025
      IA(2)=IA(2)*IEXP*134217728                                        RDCX0026
      IA=IA-IA(2)                                                       RDCX0027
      X=A                                                               RDCX0028
      A=X(2)                                                            RDCX0029
      IA=IA-IA(2)                                                       RDCX0030
      X(2)=A                                                            RDCX0031
   40 RETURN                                                            RDCX0032
      END                                                               RDCX0033
```

```
$IBFTC MABLX    NOLIST,DECK,M94,XR7,NODD
```

```
      SUBROUTINE MABL(RT,ALRT,MRT,CRT,ST,KRMP1,RL,KRMP2,IPTBL)          MABL0000
C     MABL IS A MULTIPLICATION ON THE LEFT SUBROUTINE FOR ALMOST BAND   MABL0001
C       MATRICES.                                                       MABL0002
C     LAST MODIFICATION 1/18/67-K.E.M.                                  MABL0003
C     IPTBL(1)=0 FOR INPUT NEW ROW OF RT AND LAST ROW OF ST (SET INTERN-MABL0004
C               IALLY).                                                 MABL0005
C              =1 FOR INPUT NEW ROWS OF RT AND RL (S.I.).               MABL0006
C              =2 FOR OUTPUT COMPUTED ROW OF ALRT, AND INPUT NEW ROWS OF MABL0007
C               RT AND RL (S.I.).                                       MABL0008
C              =3 FOR OUTPUT COMPUTED ROW OF ALRT AND INPUT NEW ROW OF RLMABL0009
C               (S.I.).                                                 MABL0010
C              =4 FOR OUTPUT COMPUTED ROW OF ALRT AND TERMINATE (S.I.). MABL0011
C           (2)=N.                                                      MABL0012
C           (3)=THE ACTUAL ROW OF ALRT BEING OUTPUT (S.I.).             MABL0013
      DIMENSION RT(MRT),ALRT(MRT),CRT(KRMP1,MRT),ST(KRMP1),RL(KRMP2),   MABL0014
     *IPTBL(3)                                                          MABL0015
C     INITIALIZE.                                                       MABL0016
      IS=0                                                              MABL0017
      IT=0                                                              MABL0018
      N=IPTBL(2)                                                        MABL0019
      M=MRT                                                             MABL0020
      KM=KRMP1                                                          MABL0021
      KRM=KM-1                                                          MABL0022
      IF(KM-1) 10,10,12                                                 MABL0023
   10 JPTBL=1                                                           MABL0024
      GO TO 14                                                          MABL0025
   12 JPTBL=0                                                           MABL0026
   14 IPTBL=JPTBL                                                       MABL0027
      RETURN                                                            MABL0028
      ENTRY MABLU                                                       MABL0029
C     USE.                                                              MABL0030
C     GET AND SCALE ROW.                                                MABL0031
C     TEST FOR TERMINATE.                                               MABL0032
      IF(JPTBL-4) 100,160,160                                           MABL0033
  100 IS=IS+1                                                           MABL0034
      IF(IS-N) 104,102,116                                              MABL0035
  102 JPTBL=3                                                           MABL0036
```

```
 104  IF(IS-KM) 106,110,112                                  MABL0037
C     INITIAL STATE.                                         MABL0038
 106  DO 107 J=1,M                                           MABL0039
 107  CRT(IS,J)=RT(J)*ST(IS)                                 MABL0040
      IPTBL(3)=IT                                            MABL0041
      IF(IS-KRM) 108,109,109                                 MABL0042
 108  IPTBL=0                                                MABL0043
      RETURN                                                 MABL0044
 109  IPTBL=1                                                MABL0045
      RETURN                                                 MABL0046
C     STEADY STATE.                                          MABL0047
 110  IT=1                                                   MABL0048
      JPTBL=2                                                MABL0049
 112  DO 114 J=1,M                                           MABL0050
 114  CRT(KM,J)=RT(J)*RL(2)                                  MABL0051
      GO TO 120                                              MABL0052
C     TERMINAL STATE.                                        MABL0053
 116  DO 118 J=1,M                                           MABL0054
 118  CRT(KM,J)=0.0                                          MABL0055
      KM=KM-1                                                MABL0056
C     PIVIT                                                  MABL0057
 120  IF(RL(1)-1.0) 122,126,122                              MABL0058
 122  K=RL(1)                                                MABL0059
      DO 124 J=1,M                                           MABL0060
      TS=CRT(K,J)                                            MABL0061
      CRT(K,J)=CRT(1,J)                                      MABL0062
 124  CRT(1,J)=TS                                            MABL0063
C     MULTIPLY.                                              MABL0064
 126  IF(KM-1) 129,129,127                                   MABL0065
 127  DO 128 K=2,KM                                          MABL0066
      DO 128 J=1,M                                           MABL0067
 128  CRT(K,J)=CRT(K,J)+RL(K+1)*CRT(1,J)                     MABL0068
C     OUTPUT A ROW OF ALRT.                                  MABL0069
 129  DO 130 J=1,M                                           MABL0070
 130  ALRT(J)=CRT(1,J)                                       MABL0071
      IPTBL(3)=IT                                            MABL0072
      IT=IT+1                                                MABL0073
C     TEST FOR COMPLETION.                                   MABL0074
```

```
      IF(IT-N) 140,140,150                                      MABL0075
  140 IPTBL=JPTBL                                               MABL0076
      IF(KM-1) 145,145,142                                      MABL0077
  142 DO 144 K=2,KM                                             MABL0078
      DO 144 J=1,M                                              MABL0079
  144 CRT(K-1,J)=CRT(K,J)                                       MABL0080
  145 RETURN                                                    MABL0081
C     TERMINATE.                                                MABL0082
  150 JPTBL=4                                                   MABL0083
  155 IPTBL=4                                                   MABL0084
      RETURN                                                    MABL0085
C     ERROR, IPTBL IS GREATER THAN OR EQUAL TO 4 ON ENTRY.      MABL0086
  160 IPTBL(3)=IT-1                                             MABL0087
      GO TO 155                                                 MABL0088
      END                                                       MABL0089
```

```
$IBFTC MABBX   NOLIST,DECK,M94,XR7,NODD
```

```
      SUBROUTINE MABB(U,KBL,R,X,XIN,MST,IPTBL)                          MABB0000
C     MABB IS A BACK SUBSTITUTION SUBROUTINE FOR ALMOST BAND MATRICES.  MABB0001
C     LAST MODIFICATION 2/24/67-K.E.M.                                  MABB0002
C     IPTBL(1)=0 FOR INPUT NEW ROWS OF U AND R, AND OUTPUT A ROW OF X   MABB0003
C              (SET INTERNALLY).                                        MABB0004
C             =1 FOR OUTPUT A ROW OF X AND TERMINATE (S.I.).            MABB0005
C     IPTBL(2)=N.                                                       MABB0006
C     IPTBL(3)=IR, THE ACTUAL ROW OF U,R AND X BEING OPERATED ON (S.I.).MABB0007
      DIMENSION U(KBL),R(MST),X(MST),XIN(KBL,MST)                       MABB0008
      DIMENSION IPTBL(3)                                                MABB0009
C     INITIALIZE                                                        MABB0010
      M=MST                                                             MABB0011
      KRL=KBL                                                           MABB0012
      KBLM1=KRL-1                                                       MABB0013
      IF(KBLM1) 504,504,501                                             MABB0014
  501 DO 502 J=1,M                                                      MABB0015
  502 XIN(2,J)=0.0                                                      MABB0016
  504 N=IPTBL(2)                                                        MABB0017
      IT=0                                                              MABB0018
      IPTBL=0                                                           MABB0019
      JPTBL=0                                                           MABB0020
      RETURN                                                            MABB0021
      ENTRY MABBU                                                       MABB0022
C     USE                                                               MABB0023
C     TEST FOR TERMINATE.                                               MABB0024
      IF(JPTBL) 540,510,534                                             MABB0025
  510 IT=IT+1                                                           MABB0026
      IR=N+1-IT                                                         MABB0027
      IF(IT-KRL) 512,514,514                                            MABB0028
  512 JU=IT+1                                                           MABB0029
      GO TO 521                                                         MABB0030
  514 JU=KRL                                                            MABB0031
  521 DO 524 K=1,M                                                      MABB0032
      TS=0.0                                                            MABB0033
      IF(KBLM1) 524,524,5215                                            MABB0034
 5215 DO 522 J=2,JU                                                     MABB0035
  522 TS=TS+U(J)*XIN(J,K)                                               MABB0036
```

```
  524   XIN(1,K)=(R(K)-TS)/U(1)                                  MABB0037
        DO 528 K=1,M                                             MABB0038
  528   X(K)=XIN(1,K)                                            MABB0039
  530   IF(KBLM1) 534,534,531                                    MABB0040
  531   DO 532 K=1,M                                             MABB0041
  532   XIN(JU,K)=XIN(JU-1,K)                                    MABB0042
        JU=JU-1                                                  MABB0043
        IF(JU-1) 534,534,530                                     MABB0044
  534   IPTBL(3)=IR                                              MABB0045
        IF(IT-N) 536,538,538                                     MABB0046
  536   IPTBL=0                                                  MABB0047
        RETURN                                                   MABB0048
C       TERMINATE                                                MABB0049
  538   JPTBL=1                                                  MABB0050
  540   IPTBL=1                                                  MABB0051
        RETURN                                                   MABB0052
        END                                                      MABB0053
```

STORAGE MAP

SUBROUTINE PRNX

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
NPTS	00001	I	I	00002	I	IJK	00003
II	00004	I	IF	00005	I	MS	00006
IJ	00007	I	MF	00010	I		

ENTRY POINTS

PRNX	SECTION	3

SUBROUTINES CALLED

.FWRD.	SECTION	4	.UNOS.	SECTION	5	.FFIL.	SECTION	
.FCNV.	SECTION	7	SYSLOC	SECTION	8			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
1001	FORMAT	00021	1002	FORMAT	00042	50	6A	
200	40A	00167	100	26A	00122	150	32A	

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00232.

PRMYX STORAGE MAP

SUBROUTINE PRMY

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IT	00001	I	IF	00002	I	IJK	00003
IJ	00004	I	NPT1	00005	I		

ENTRY POINTS

PRMY	SECTION	3

SUBROUTINES CALLED

.FWRD.	SECTION	4	.UNO6.	SECTION	5	.FFIL.	SECTION	
.FCNV.	SECTION	7	SYSLOC	SECTION	8			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
1000	FORMAT	00017	1001	FORMAT	00024	400	344	
300	25A	00145						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00244.

LANUX STORAGE MAP

SUBROUTINE LANU

COMMON VARIABLES

COMMON BLOCK	COMAA		ORIGIN	00001		LENGTH	00001
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ICHECK	00000	I					
COMMON BLOCK	COM41		ORIGIN	00002		LENGTH	00013
KRMP1	00000	I	KRMP2	00001	I	MRT	00002
KBL	00003	I	IPTBL	00004	I	IPT1	00004
COMMON BLOCK	COM42		ORIGIN	00015		LENGTH	00112
AB	00000	R	U	00027	R	ST	00056
AL1	00072	R	R	00107	R	X	00110
C	00111	R					
COMMON BLOCK	COM43		ORIGIN	00127		LENGTH	00003
DTM	00000	D	DTM1	00002	R		
COMMON BLOCK	COM44		ORIGIN	00132		LENGTH	00076
CRT	00000	R	ALT	00014	D	CT	00046
COMMON BLOCK	COM45		ORIGIN	00230		LENGTH	00001
INTF2	00000	I					
COMMON BLOCK	COM49		ORIGIN	00231		LENGTH	00001
IOPT5	00000	I					

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
PDAT	00232	R					

LANUX STORAGE MAP

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
MONE	00235	I	INDR	00236	I	INDP	00237
MXNRKR	00240	I	CC	00241	R	I	00242
NRKR	00243	I	NRUL	00244	I	NCUL	00245
IJK	00246	I	MS	00247	I	MF	00250
KK	00251	I	MXKK	00252	I	LROI	00253
LROF	00254	I	IJ	00255	I	IJJ	00256
NCK	00257	I	NREC1	00260	I	J	00261

ENTRY POINTS

LANU	SECTION	17

SUBROUTINES CALLED

MABT	SECTION	18	.FRDB.	SECTION	19	.FWRD.	SECTION
.FWRB.	SECTION	21	MABTU	SECTION	22	.UN14.	SECTION
.FRLR.	SECTION	24	.FBLT.	SECTION	25	.FBDT.	SECTION
.UN06.	SECTION	27	.FFIL.	SECTION	28	.FCNV.	SECTION
.UN15.	SECTION	30	.FWLR.	SECTION	31	.UN09.	SECTION
SYSLOC	SECTION	33					

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
4001	FORMAT	00304	4002	FORMAT	00314	30	13A
60	39A	00445	50	34A	00437	100	80A
90	70A	00540	80	66A	00535	95	77A
150	94A	00610	110	88A	00600	200	123A
175	112A	00647	170	106A	00641	178	114A
180	118A	00664	300	184A	01122	220	148A
210	141A	00751	240	162A	01052	230	155A
250	176A	01105	260	183A	01122	350	227A

320	213A	01227	340	223A	01245	330	220A
400	233A	01263					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01451.

SUBROUTINE PELR

COMMON VARIABLES

COMMON BLOCK COM41 ORIGIN 00001 LENGTH 00013

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
KRMP1	00000	I	KRMP2	00001	I	MRT	00002
KBL	00003	I	IPTBL	00004	I	IPT1	00004

COMMON BLOCK COM42 ORIGIN 00014 LENGTH 00112

AB	00000	R	U	00027	R	ST	00056
AL1	00072	R	R	00107	R	X	00110
C	00111	R	ALRT	00111	R		

COMMON BLOCK COM44 ORIGIN 00126 LENGTH 00076

CRT	00000	R	ALT	00014	D	CT	00046

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
INDP	00224	I	MRNR	00225	I	NREC2	00226
IRHV	00227	I	INP	00230	I	IRHK	00231
MRNL	00232	I	MRNP	00233	I	INL	00234
NREC1	00235	I	J	00236	I		

ENTRY POINTS

PELR SECTION 9

SUBROUTINES CALLED

MABL	SECTION	10	.FRDB.	SECTION	11	MABLU	SECTION
.UNO9.	SECTION	13	.FRLR.	SECTION	14	.FBLT.	SECTION

PELRX STORAGE MAP

.FBDT. SECTION 16 SYSLOC SECTION 17

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
40	17A	00310	50	12A	00302	70	33A
90	51A	00416	80	46A	00406	100	57A
150	70A	00462	110	65A	00455	200	76A
350	92A	00530	320	86A	00515	500	98A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00642.

SUBROUTINE BSUB

COMMON VARIABLES

COMMON BLOCK COM41 ORIGIN 00001 LENGTH 00013

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
KRMP1	00000	I	KRMP2	00001	I	MRT	00002
KBL	00003	I	IPTBL	00004	I	IPTBL	00004
IPT1	00004	I	IPT2	00005	I	IPT3	00006
MST	00002	I					

COMMON BLOCK COM42 ORIGIN 00014 LENGTH 00112

AB	00000	R	U	00027	R	ST	00056
AL1	00072	R	R	00107	R	X	00110
C	00111	R	XIN	00000	R		

COMMON BLOCK COM44 ORIGIN 00126 LENGTH 00076

CRT	00000	R	ALT	00014	D	CT	00046

COMMON BLOCK COM45 ORIGIN 00224 LENGTH 00001

INTF2	00000	I

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ICX	00225	I	IRHK	00226	I	IRECU	00227
IAX	00230	I	NRECX	00231	I	IRX	00232
ICU	00233	I	NRECU	00234	I	I	00235
J	00236	I					

ENTRY POINTS

BSUB SECTION 11

BSUBX STORAGE MAP

SUBROUTINES CALLED

MABB	SECTION	12	.FRDB.	SECTION	13	.FBST.	SECTION
MABBU	SECTION	15	.UN09.	SECTION	16	.FRLR.	SECTION
.FBLT.	SECTION	18	.FBDT.	SECTION	19	.UN15.	SECTION
SYSLOC	SECTION	21					

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
60	15A	00310	50	11A	00303	250	75A	
100	35A	00364	150	47A	00432	200	63A	
175	59A	00466	230	74A	00520	225	71A	
300	88A	00551	260	84A	00540	400	125A	
340	110A	00624	320	106A	00621	370	121A	
360	118A	00651	450	130A	00670			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01002.

SUBROUTINE MABT

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
EPS	00001	D	IPS	00013	I		

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
DTM2	00021	D	TI	00023	D	TS	00025
TDTM	00027	D	IB	00031	I	IT	00032
IX	00033	I	KRL	00034	I	MM	00035
N	00036	I	KRM	00037	I	KRC	00040
KM	00041	I	KMP	00042	I	ISAVL	00043
I	00044	I	JPTBL	00045	I	JC	00046
STA	00047	R	J	00050	I	JK	00051
IXP	00052	I	K	00053	I	KK	00054
IEXP	00055	I	IX1	00056	I		

ENTRY POINTS

MABT	SECTION	3	MABTU	SECTION	4

SUBROUTINES CALLED

RDCX	SECTION	5	E.1	SECTION	6	E.2	SECTION	
E.3	SECTION	8	E.4	SECTION	9	CC.1	SECTION	
CC.2	SECTION	11	CC.3	SECTION	12	CC.4	SECTION	
SYSLOC	SECTION	14						

EFN IFN CORRESPONDENCE

MABTX STORAGE MAP

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
10	1A	00113	12	17A	00146	100	25A
480	255A	01241	102	28A	00170	122	81A
104	30A	00174	105	33A	00202	106	34A
108	37A	00213	110	45A	00251	112	57A
114	63A	00332	118	71A	00356	116	67A
120	75A	00372	454	242A	01201	200	84A
201	87A	00435	202	89A	00446	210	123A
204	98A	00501	203	95A	00470	205	102A
206	109A	00545	208	119A	00573	212	126A
222	147A	00657	214	132A	00620	216	136A
220	144A	00647	218	137A	00630	224	149A
300	184A	01012	226	151A	00670	234	181A
228	155A	00714	230	161A	00734	2305	163A
2315	172A	00763	231	166A	00754	232	175A
304	196A	01014	310	197A	01032	306	188A
308	192A	01025	312	201A	01041	314	209A
350	214A	01062	400	237A	01167	352	216A
354	218A	01072	356	231A	01157	355	224A
402	239A	01173	460	245A	01206	450	241A
455	244A	01205	462	247A	01210	470	252A
475	254A	01237					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01537.

STORAGE MAP

SUBROUTINE RDCX

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
A	00001	R	IA	00001	I		

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
MCHECK	00003	I	MASK1	00004	I		

ENTRY POINTS

RDCX SECTION 3

SUBROUTINES CALLED

CC.1	SECTION	4	CC.2	SECTION	5	CC.3	SECTION
CC.4	SECTION	7	SYSLOC	SECTION	8		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
5	4A	00023	40	14A	00111	30	10A
10	6A	00032					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00150.

SUBROUTINE MABL

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IS	00001	I	IT	00002	I	N	00003
M	00004	I	KM	00005	I	KRM	00006
JPTBL	00007	I	K	00010	I	TS	00011

ENTRY POINTS

MABL	SECTION	3	MABLU	SECTION	4

SUBROUTINES CALLED

CC.1	SECTION	5	CC.2	SECTION	6	CC.3	SECTION
CC.4	SECTION	8	SYSLOC	SECTION	9		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
10	9A	00047	12	11A	00052	14	12A
100	16A	00062	160	117A	00374	104	20A
102	19A	00072	116	49A	00163	106	22A
110	37A	00136	112	39A	00142	107	26A
108	33A	00131	109	35A	00133	114	43A
120	57A	00204	118	53A	00175	122	59A
126	71A	00240	124	67A	00233	129	87A
127	73A	00244	128	79A	00270	130	91A
140	98A	00326	150	114A	00367	145	113A
142	101A	00334	144	107A	00357	155	115A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00570.

STORAGE MAP

SUBROUTINE MABB

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
M	00001	I	KRL	00002	I	KBLM1	00003
N	00004	I	IT	00005	I	JPTBL	00006
IR	00007	I	JU	00010	I	K	00011
TS	00012	R					

ENTRY POINTS

MABB SECTION 3 MABBU SECTION 4

SUBROUTINES CALLED

SYSLOC SECTION 5

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
504	14A	00053	501	6A	00040	502	10A
540	71A	00234	510	21A	00065	534	66A
512	25A	00100	514	27A	00104	521	28A
524	40A	00143	5215	33A	00126	522	36A
528	48A	00161	530	53A	00166	531	55A
532	59A	00206	536	68A	00230	538	70A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00414.

```
$IBFTC MON5X   NOLIST,DECK,M94,XR7,NODD
```

```
      1 I3)                                                             MON50370
        MITE = IIMITE                                                   MON50380
        TNO = CCBETA                                                    MON50390
        REWIND 4                                                        MON50400    3
        REWIND 14                                                       MON50410    4
        MAXHRM =   7                                                    MON50420
        NROSEG = 600                                                    MON50430
C       SET HARMONIC VALUE FOR LINK                                     MON50440
        IJH = INTF5                                                     MON50450
        M   = MLIST(IJH)                                                MON50460
        HM = DBLE(M)                                                    MON50470
C       INITIALIZE CONVERGENCE FLAG                                     MON50480
C       NCCS--CONVERGENCE FLAG SET FOR BUILDING UP NEW RIGHT HAND VECTOR MON50490
        NCCS = 0                                                        MON50500
C       INITIALIZE FLAG FOR ZERO'TH ORDINARY ITERATION                  MON50510
        NZOI = 0                                                        MON50520
C       INITIALIZE INDEX FOR SOLUTION VECTOR                            MON50530
        INDK = 0                                                        MON50540
C       CALCULATE SUMMATION INDICES FOR ALL HARMONICS                   MON50550
        NPSH = LSP(NSGMTS)+NPSEG(NSGMTS)-1                              MON50560
        NPTS = 8*NPSH                                                   MON50570
        NPT = NPTS                                                      MON50580
C       IF SPECIAL ITERATION, MOVE RHM TO RHK                           MON50590
        IF(INTF2) 80,80,50                                              MON50600
   50   CONTINUE                                                        MON50610
        DO 60 I=1,NPTS                                                  MON50620
        RHK(I) = RHM(I)                                                 MON50630
   60   CONTINUE                                                        MON50640
C       GET FORCES FOR SPECIAL ITERATION, OMEGA(THETA) AND N(PHI)       MON50650
        IKK = 4                                                         MON50660
        DO 70 I=1,NPSH                                                  MON50670
        OTHE(I) = OTHE(I) + RHK(IKK)                                    MON50680
        FNPH(I) = FNPH(I) + CONFAC * RHK(IKK+2)                         MON50690
        IKK = IKK+8                                                     MON50700
   70   CONTINUE                                                        MON50710
C       SET FLAG FOR USE OF SPECIAL ITERATION FORCES                    MON50720
        INTF2 = 1                                                       MON50730
   80   CONTINUE                                                        MON50740
```

```
   80 CONTINUE                                                          MON50750
      IF(IRES( 4).GT.0) GO TO 160                                       MON50755
C     IF RESTART OPTION (3), DON'T INITIALIZE PRIOR SOLUTION TAPE       MON50760
      IF(IRES( 3).EQ.1) GO TO 160                                       MON50770
C     TEST FOR FIRST TIME THRU SOLUTION                                 MON50780
C     FIRST...FOR ORDINARY ITERATION CASE                               MON50790
      IF(INTF1) 100,100,110                                             MON50800
  100 CONTINUE                                                          MON50810
      IF(INTF6) 120,120,160                                             MON50820
  110 CONTINUE                                                          MON50830
C     SECOND..FOR SPECIAL ITERATION CASE                                MON50840
      IF(INTF1-INTF2) 160,120,160                                       MON50850
  120 CONTINUE                                                          MON50860
C     INITIALIZE PRIOR SOLUTION TAPE                                    MON50870
      DO 130 J=1,KS                                                     MON50880
      DO 130 I=1,NROSEG                                                 MON50890
  130 PS(I,J) = 0.                                                      MON50900
C     INITIALIZE PRIOR SOLUTION TAPE                                    MON50910
      DO 150 IK=1,NSGMTS                                                MON50920
      NPS = 8*NPSEG(IK)                                                 MON50930
      WRITE(04) ((PS(I,J),I=1,NPS),J=1,KS)                              MON50940    69
  150 CONTINUE                                                          MON50950
      REWIND 4                                                          MON50960    79
  160 CONTINUE                                                          MON50970
      CALL GETM(MLIST,KS,MIND,ISFLAG,MAXHRM)                            MON50980    81
      IJK = 0                                                           MON50990
      JRH = 0                                                           MON51000
C     START LOOP ON SEGMENTS                                            MON51010
  200 CONTINUE                                                          MON51020
C     INCREMENT SEGMENT COUNTER                                         MON51030
      IJK = IJK+1                                                       MON51040
C     SET SEGMENT FLAGS AND VALUES                                      MON51050
      MS = LSP(IJK)                                                     MON51060
      ME = NPSEG(IJK)+MS-1                                              MON51070
      SINE = DBLE(SINP(IJK))                                            MON51080
      COSI = DBLE(COSP(IJK))                                            MON51090
C     CALCULATE NUMBER OF ROWS FOR THIS SEGMENT                         MON51100
      NROS = 8*NPSEG(IJK)                                               MON51110
```

```
  250 CONTINUE                                                          MON51120
C     READ PRIOR SOLUTION FOR THIS SEGMENT                              MON51130
      READ(4) ((PS(I,J),I=1,NROS),J=1,KS)                               MON51140    93
      IF(IJK.EQ.NSGMTS) REWIND 4                                        MON51150
C     IF RESTART OPTION 4, JUST BUILD R.H.V. I.E. SPEC. ITERATION FORCESMON51154    103
      IF(IRES( 4).GT.0) GO TO 255                                       MON51156
C     IF RESTART OPTION (3) ON, JUST BUILD RIGHT HAND VECTOR            MON51160
      IF(IRES(3)) 252,252,253                                           MON51170
  252 CONTINUE                                                          MON51180
C     IF SPECIAL ITERATION, DON'T SKIP SCRATCH TAPE USE                 MON51190
      IF(INTF2.GT.0) GO TO 254                                          MON51200
C     IF FIRST ORDINARY ITERATION, DON'T INCREMENT HARMONIC INDEX       MON51210
      IF(INTF6) 253,253,254                                             MON51220
  253 CONTINUE                                                          MON51230
C     SAVE HARMONIC INDEX                                               MON51240
      IJHTT = IJH                                                       MON51250
C     SET FLAG FOR ZERO'TH ORDINARY ITERATION                           MON51260
      NZOI = 1                                                          MON51270
C     INITIALIZE BETA PRINTING FLAG                                     MON51280
      INTF(8) = KS                                                      MON51290
      GO TO 303                                                         MON51300
  254 CONTINUE                                                          MON51310
      IF(NCCS) 258,258,255                                              MON51320
  255 CONTINUE                                                          MON51330
      IJH = 1                                                           MON51340
      IJHTT = IJH                                                       MON51345
      GO TO 303                                                         MON51350
  258 CONTINUE                                                          MON51360
C     SHOULD WE TEST CONVERGENCE CRITERION                              MON51370
C     IF INTF2 .GT. 0, SPECIAL ITERATION, NO.                           MON51380
      IF(INTF2) 270,260,270                                             MON51390
  260 CONTINUE                                                          MON51400
C     TEST CONVERGENCE CRITERION                                        MON51410
      CALL TCCR(PS,RHK,BETA,IBETA,KS)                                   MON51420
C     TEST FOR BETA PRINTING                                            MON51430    126
      IF(IJK.LT.NSGMTS) GO TO 270                                       MON51440
      IF(INTF6.LT.INTF(8)) GO TO 270                                    MON51450
C     PRINT BETA VALUES                                                 MON51460
```

```
      IF6 = INTF6/KS                                                    MON51470
      WRITE(6,2020) IF6                                                 MON51480   134
  263 WRITE(6,2000)                                                     MON51490   135
      DO 265 JJ=1,KS                                                    MON51500
      WRITE(6,2005) MLIST(JJ),(IBETA(II,JJ),BETA(II,JJ),II=1,5)         MON51510   138
  265 CONTINUE                                                          MON51520
      IF(NCCS.EQ.1) GO TO 365                                           MON51530
C     INCREMENT BETA PRINTING FLAG                                      MON51540
      INTF(8) = INTF(8) + KS                                            MON51550
  270 CONTINUE                                                          MON51560
C     ADD INCREMENTAL SOLUTION TO PRIOR SOLUTION                        MON51570
      DO 275 I=1,NROS                                                   MON51580
      JRH = JRH+1                                                       MON51590
      PS(I,IJH) = PS(I,IJH) + RHK(JRH)                                  MON51600
  275 CONTINUE                                                          MON51610
  280 CONTINUE                                                          MON51620
C     WRITE PRIOR SOLUTION ON TAPE                                      MON51630
      WRITE(14) ((PS(II,JJ),II=1,NROS),JJ=1,KS)                         MON51640   162
      IF(IJK.EQ.NSGMTS) REWIND 14                                       MON51650   172
  300 CONTINUE                                                          MON51660
C     SAVE HARMONIC INDEX                                               MON51670
      IJHTT = IJH                                                       MON51680
C     INTF( 2) .GT. 0, DON'T CHANGE HARMONIC INDEX                      MON51690
C     INTF( 2) .EQ. 0, INCREMENT HARMONIC INDEX                         MON51700
      IF(INTF2) 301,301,303                                             MON51710
  301 CONTINUE                                                          MON51720
C     SET HARMONIC INDEX FOR NEXT HARMONIC                              MON51730
      IF(IJH.EQ.KS) IJH=0                                               MON51740
      IJH = IJH+1                                                       MON51750
  303 CONTINUE                                                          MON51760
      M  = MLIST(IJH)                                                   MON51770
      HM = DBLE(M)                                                      MON51780
C     CALCULATE NEW RIGHT HAND VECTOR FOR THIS SEGMENT, NEXT HARMONIC   MON51790
      CALL CNRH                                                         MON51800
C     RESTORE HARMONIC INDEX FOR PREVIOUS SOLUTION                      MON51810   183
      IJH = IJHTT                                                       MON51820
      M  = MLIST(IJH)                                                   MON51830
      HM = DBLE(M)                                                      MON51840
```

```
  304 CONTINUE                                                          MON51850
C     IF ALL SEGMENTS AREN'T DONE, START LOOP                           MON51860
      IF(IJK-NSGMTS) 200,305,305                                        MON51870
  305 CONTINUE                                                          MON51880
      IF(IRES( 4).GT.0) GO TO 400                                       MON51885
      IF(NCCS.EQ.1) GO TO 395                                           MON51890
C     IF ZERO'TH ITERATION, DON'T USE SCRATCH TAPE                      MON51900
      IF(NZOT) 306,306,450                                              MON51910
  306 CONTINUE                                                          MON51920
C     TRANSFER PRIOR SOLUTION FROM TAPE 14 TO TAPE 4                    MON51930
      DO 307 I=1,NSGMTS                                                 MON51940
      NPTS = 8*NPSEG(I)                                                 MON51950
      READ(14)  ((PS(II,JJ),II=1,NPTS),JJ=1,KS)                         MON51960   203
      WRITE(04) ((PS(II,JJ),II=1,NPTS),JJ=1,KS)                         MON51970   211
  307 CONTINUE                                                          MON51980
      REWIND 14                                                         MON51990   221
      REWIND 4                                                          MON52000   222
      IF(INTF2) 320,320,310                                             MON52010
  310 CONTINUE                                                          MON52020
C     MOVE RHK TO RHM                                                   MON52030
      DO 315 I=1,INDK                                                   MON52040
      RHM(I) = RHK(I)                                                   MON52050
  315 CONTINUE                                                          MON52060
      GO TO 400                                                         MON52070
  320 CONTINUE                                                          MON52080
C     CHECK FOR C.C. SATISFACTION                                       MON52090
      ITI = 0                                                           MON52100
      DO 340 I=1,KS                                                     MON52110
      IF(ABS(BETA(1,I)) - TNO) 330,340,340                              MON52120
  330 CONTINUE                                                          MON52130
      ITI = ITI+1                                                       MON52140
  340 CONTINUE                                                          MON52150
      IF(ITI-KS) 400,350,400                                            MON52160
  350 CONTINUE                                                          MON52170
C     TEST FOR CONVERGENCE WITHOUT USING ALL HARMONICS                  MON52180
      IF(INTF6 - KS) 400,360,360                                        MON52190
  360 CONTINUE                                                          MON52200
      WRITE(6,2040) INTF7                                               MON52210
```

```
C     SET CONVERGENCE FLAG                                              MON52214   248
      NCCS = 1                                                          MON52215
      GO TO 263                                                         MON52220
  365 CONTINUE                                                          MON52230
      IF(INTF7 .GE. NLOAD-1) GO TO 395                                  MON52240
C     INCREMENT LOADING                                                 MON52250
      DO 380 IJH = 1,KS                                                 MON52260
      DO 372 I=1,4                                                      MON52270
      BC(I,IJH,1) = BC(I,IJH,1) + DLBC(I,IJH,1)                         MON52280
      BC(I,IJH,2) = BC(I,IJH,2) + DLBC(I,IJH,2)                         MON52290
  372 CONTINUE                                                          MON52300
      DO 380 IJK = 1,NSGMTS                                             MON52310
      DO 375   I = 1,5                                                  MON52320
      FLDI(I,IJK,IJH) = FLDI(I,IJK,IJH)+DLDI(I,IJK,IJH)                 MON52330
      FLDF(I,IJK,IJH) = FLDF(I,IJK,IJH)+DLDF(I,IJK,IJH)                 MON52340
  375 CONTINUE                                                          MON52350
      FMPD(1,IJK,IJH) = FMPD(1,IJK,IJH)+FMPD(2,IJK,IJH)                 MON52360
      FNRD(1,IJK,IJH) = FNRD(1,IJK,IJH)+FNRD(2,IJK,IJH)                 MON52370
      FNYD(1,IJK,IJH) = FNYD(1,IJK,IJH)+FNYD(2,IJK,IJH)                 MON52380
      FNPTD(1,IJK,IJH)= FNPTD(1,IJK,IJH)+FNPTD(2,IJK,IJH)               MON52390
  380 CONTINUE                                                          MON52400
C     INCREMENT LOADING NUMBER                                          MON52410
      INTF( 7) = INTF( 7) + 1                                           MON52420
C     INITIALIZE BETA PRINTING FLAG FOR NEW LOAD LEVEL                  MON52430
      INTF(8) = KS                                                      MON52440
  385 CONTINUE                                                          MON52450
C     INITIALIZE SEGMENT COUNTER                                        MON52480
      IJK  = 0                                                          MON52490
C     INITIALIZE INDEX FOR SOLUTION VECTOR                              MON52500
      INDK = 0                                                          MON52510
      GO TO 200                                                         MON52520
  395 CONTINUE                                                          MON52530
C     INITIALIZE FLAGS AND 'L+U' TAPE FOR NEW LOAD LEVEL                MON52540
C     FLAGS .. INTF4 -- OVERLAY FLAG FOR REGION 4                       MON52542
C              INTF5 -- HARMONIC INDEX                                  MON52544
C              INTF6 -- NUMBER OF ORDINARY ITERATIONS                   MON52546
      INTF4  = 2                                                        MON52550
      INTF5  = 1                                                        MON52552
```

```
      INTF6  = 1                                                MON52554
      REWIND   9                                                MON52556
C     SOLUTION HAS CONVERGED....GO TO REGION 6                  MON52560    301
      KKKKK = 6                                                 MON52570
      RETURN                                                    MON52580
  400 CONTINUE                                                  MON52590
      IF(INTF2) 450,450,410                                     MON52600
  410 CONTINUE                                                  MON52610
C     SET FLAGS TO RETURN TO REGION 3                           MON52620
      INTF2 = INTF2-1                                           MON52630
      INTF4 = 1                                                 MON52640
C     SET HARMONIC FOR SPECIAL ITERATION                        MON52650
      INTF5 = 1                                                 MON52660
      KKKKK = 3                                                 MON52670
      IF(ICHECK.GT.0) KKKKK = 7                                 MON52680
      RETURN                                                    MON52690
  450 CONTINUE                                                  MON52700
C     SET OVERLAY FLAG POINTER FOR REGION 4                     MON52710
      INTF4 = 2                                                 MON52720
      IF(INTF6.EQ.0) GO TO 460                                  MON52730
C     SET HARMONIC FOR ORDINARY ITERATION--NEXT HARMONIC        MON52740
      IF(IJH.EQ.KS) IJH = 0                                     MON52750
      INTF5 = IJH + 1                                           MON52760
  460 CONTINUE                                                  MON52770
C     TEST FOR MAXIMUM NUMBER OF ITERATIONS                     MON52780
      IF(INTF6 - MITE*KS) 500,480,480                           MON52790
  480 CONTINUE                                                  MON52800
C     SET FLAGS TO PRINT VALUES, THEN TERMINATE                 MON52810
      LOAD = LOADNO + 1                                         MON52815
      WRITE(5,2050) MITE,LOAD                                   MON52820    322
      IOPT( 6) =  3                                             MON52830
      IOPT( 7) =  3                                             MON52840
      IOPT( 8) =  1                                             MON52850
      IOPT( 9) =180                                             MON52855
      IOPT(10) =  1                                             MON52860
      IOPT(11) =  0                                             MON52870
      IOPT(12) =  0                                             MON52880
      LDNOPR = NLOAD                                            MON52890
```

```
      LOADNO = NLOAD - 1                                    MON52900
      KKKKK = 6                                             MON52910
      RETURN                                                MON52920
  500 CONTINUE                                              MON52930
C     INCREMENT ORDINARY ITERATION COUNTER                  MON52940
      INTF6 = INTF6 + 1                                     MON52950
C     CLEAR RESTART OPTION (3) FLAG                         MON52960
      INES(3) = 0                                           MON52970
      KKKKK = 4                                             MON52980
      IF(ICHECK.GT.0) KKKKK = 7                             MON52990
      RETURN                                                MON53000
      END                                                   MON53010
```

```
C     TEST CONVERGENCE CRITERION                                        TCCR0010
      SUBROUTINE TCCR(PS,RHK,BETA,IBETA,KS)                             TCCR0020
      DIMENSION PS(600,1),RHK(    1),BETA(5,1),IBETA(5,1)               TCCR0030
      DIMENSION UNKMAX(8)                                               TCCR0040
      COMMON/COM51/IJK,IJH,IJ,MS,ME                                     TCCR0050
C     INITIALIZE FLAGS AND COUNTERS                                     TCCR0060
      IPS = 0                                                           TCCR0070
      IRH = 8*(MS-1)                                                    TCCR0080
      JRH = IRH                                                         TCCR0090
      IBB = 0                                                           TCCR0100
      FACTOR = 5.0E-1                                                   TCCR0110
C     INITIALIZE BETA FOR THIS HARMONIC                                 TCCR0120
      IF(IJK.GT.1) GO TO 51                                             TCCR0130
      DO 50 I=1,5                                                       TCCR0140
      BETA(I,IJH)  = 0.0                                                TCCR0150
      IBETA(I,IJH) = 0                                                  TCCR0160
   50 CONTINUE                                                          TCCR0170
   51 CONTINUE                                                          TCCR0180
C     INITIALIZE MAXIMUM UNKNOWN VALUE ARRAY                            TCCR0190
      DO 52 I=1,8                                                       TCCR0200
   52 UNKMAX(I) = 0.0                                                   TCCR0210
C     CALCULATE THE NUMBER OF VALUES IN THIS SEGMENT                    TCCR0220
      NPTS = ME - MS + 1                                                TCCR0230
      NVLS = 8 * NPTS                                                   TCCR0240
C     GET MAXIMUM VALUES FOR EACH OF THE UNKNOWNS                       TCCR0250
      DO 56 I=1,NVLS                                                    TCCR0260
      IF(IBB.EQ.8) IBB = 0                                              TCCR0270
      PSS = 0.0                                                         TCCR0280
      DO 54 J=1,KS                                                      TCCR0290
   54 PSS = PSS + PS(I,J)                                               TCCR0300
      IBB = IBB + 1                                                     TCCR0310
      JRH = JRH + 1                                                     TCCR0320
      UNKMAX(IBB) = AMAX1( UNKMAX(IBB), ABS(PSS) )                      TCCR0330
   56 CONTINUE                                                          TCCR0340
      DO 58 I=1,8                                                       TCCR0350
   58 UNKMAX(I) = UNKMAX(I) * FACTOR                                    TCCR0360
C     START LOOP                                                        TCCR0370
```

```
      IBB = 0                                                           TCCR0380
      DO 200 IK=1,NVLS                                                  TCCR0390
C     GET VALUES FOR CURRENT POINT                                      TCCR0400
      IRH = IRH+1                                                       TCCR0410
      IPS = IPS+1                                                       TCCR0420
      IF(IBB.EQ.8) IBB = 0                                              TCCR0430
      IBB = IBB + 1                                                     TCCR0440
C     SKIP SUM ON 'U' AND 'NBAR(P-T)'                                   TCCR0450
      IF(IBB.EQ.1) GO TO 150                                            TCCR0460
      IF(IBB.EQ.5) GO TO 150                                            TCCR0470
      PSS = 0.0                                                         TCCR0480
      DO 59 I=1,KS                                                      TCCR0490
   59 PSS = PSS + PS(IPS,I)                                             TCCR0500
      PST   =  PS(IPS,IJH)                                              TCCR0510
      RHH   =  RHK(IRH)                                                 TCCR0520
C     TEST FOR INSIGNIFICANT SOLUTION VALUE--P.S. .LT. FACTOR*P.S.(MAX) TCCR0530
      IF(ABS(PSS) .LT.UNKMAX(IBB) ) GO TO 150                           TCCR0540
C     FORM BETA VALUE FOR THIS POINT                                    TCCR0550
   60 BT = RHH/(PSS + RHH)                                              TCCR0560
C     INDEX OF THIS POINT = INDEX OF FIRST POINT IN SEGMENT +           TCCR0570
C              (UNKNOWN INDEX - 1) / NUMBER OF UNKNOWNS AT EACH POINT   TCCR0580
      IJT = MS + (IK-1)/8                                               TCCR0590
C     TEST FOR LARGEST VALUE OF BETA                                    TCCR0600
      BR = BT                                                           TCCR0610
      DO 100 I=1,5                                                      TCCR0620
      IF(ABS(BR) - ABS(BETA(I,IJH)) ) 100,100,70                        TCCR0630
   70 CONTINUE                                                          TCCR0640
      BETAU = BETA(I,IJH)                                               TCCR0650
      BETA(I,IJH) = BR                                                  TCCR0660
      BR = BETAU                                                        TCCR0670
      IJU = IBETA(I,IJH)                                                TCCR0680
      IBETA(I,IJH) = IJT                                                TCCR0690
      IJT = IJU                                                         TCCR0700
  100 CONTINUE                                                          TCCR0710
  150 CONTINUE                                                          TCCR0720
  200 CONTINUE                                                          TCCR0730
      RETURN                                                            TCCR0740
      END                                                               TCCR0750
```

```
$IBFTC CNRHX    VOLIST,DECK,M94,XR7,NODD
```

```
C     BUILD NEW RIGHT HAND VECTOR FOR THIS SEGMENT                      CNRH0010
      SUBROUTINE CNRH                                                   CNRH0020
      DOUBLE PRECISION SINE,COSI,HM,D,BK,R,FM,DS,XBR,YBR,RSTR,ZBR,GOP,  CNRH0030
     1 GOM,GNT,GNPT,FLAM,EPS,GMT,GMPT,X,A,B,C,DZ,GLAM(3,2,7)            CNRH0040
      DIMENSION PS(600,7),RHK(2400),RHM(2400),SGOP(75,7),SGOM(75,7),    CNRH0050
     1 SEPS(75,7),FDC(3,75),MIND(7,5,7),ISFLAG(3,7),BETA(5,7),IBETA(5,7)CNRH0060
      EQUIVALENCE (AK(1,1),RHK(1)),(PS(1,1),AK(21,18)),(AL(1,1),RHM(1)),CNRH0070
     1 (AL(1,2),SGOP(1,1)),(AL(126,4),SGOM(1,1)),(AL(51,7),SEPS(1,1)),  CNRH0080
     2 (AL(176,9),FDC(1,1)),(AL(1,11),MIND(1,1,1)),                     CNRH0090
     3 (AL(144,12),ISFLAG(1,1)),(AL(166,12),IBETA(1,1)),                CNRH0100
     4 (AL(2,13),GLAM(1,1,1)),(AL(166,13),BETA(1,1))                    CNRH0110
      COMMON/COMAA/ICHECK                                               CNRH0120
      COMMON/COM51/IJK,IJH,IJ,MS,MF                                     CNRH0130
      COMMON/COM52/SINE,COSI,HM,M,NONE                                  CNRH0140
      COMMON/COM53/D(2),BK(2),R(2),FM(2),DS(2),XBR(2),YBR(2)            CNRH0150
      COMMON/COM54/RSTR(8),ZBR(7,2),GOP(7,2),GOM(7,2),GNT(7,2),         CNRH0160
     1 GNPT(7,2),FLAM(9,2),EPS(7,2)                                     CNRH0170
      COMMON/COM55/GMT(2),GMPT(2)                                       CNRH0180
      COMMON/COM56/X(8,2,7)                                             CNRH0190
      COMMON/COM58/A(7),B(7),C(7),DZ(7)                                 CNRH0200
      COMMON/COM59/IND,INDK                                             CNRH0210
      COMMON NSGMTS, KS, ISGMT, IFTNO, LOADNO, LDNOPR, NLOAD, KKKKK     CNRH0220
      COMMON PHI(7),SINP(7),COSP(7),NPSEG(7),LSP(7),MLIST(7),ECR(7),    CNRH0230
     1 ECY(7),IOPT(14),IRES(10),NREC(21),INTF(10),NSGPL(10),NPTPT(10),  CNRH0240
     2 NLOPL(10)                                                        CNRH0250
      COMMON EI(2,7),HI(2,7),FMJI(2,7),RI(7)                            CNRH0260
      COMMON FLDI(5,7,7),FLDF(5,7,7),DLDI(5,7,7),DLDF(5,7,7)            CNRH0270
      COMMON FMPO(2,7,7),FNRO(2,7,7),FNYO(2,7,7),FNPTO(2,7,7),ECRS(7,7),CNRH0280
     1 ECYS(7,7),RSS(7,7),SCF(16,7,7)                                   CNRH0290
      COMMON IBC(4,7,2),BC(4,7,2),DLBC(4,7,2)                           CNRH0300
      COMMON S(300),FNTH(300),FNPH(300),DPHI(300),DTHE(300),STP(12,7)   CNRH0310
      COMMON AK(140,23),AU(140,23),AL(200,13),RHV(600),SAVE(5,10)       CNRH0320
C     SET INITIAL PARAMETERS                                            CNRH0330
      MFMS2M = (MF-MS)/2 -3                                             CNRH0340
      MFMS2P = (MF-MS)/2+3                                              CNRH0350
      MAXSEG =   7                                                      CNRH0360
      MAXHFM =   7                                                      CNRH0370
```

```
      NPTSEG =  75                                                      CNRH0380
      MAXPTS = 300                                                      CNRH0390
      NROSEG = 600                                                      CNRH0400
      INTF2 = INTF(C2)                                                  CNRH0410
      LOPE = 2                                                          CNRH0420
      NONE = 1                                                          CNRH0430
      IJ   = MS                                                         CNRH0440
      III = IJ                                                          CNRH0450
C     SET INDEX FOR SPECIAL ITERATION FORCE N(THETA)                    CNRH0460
      INTH = MS - 1                                                     CNRH0470
   30 CONTINUE                                                          CNRH0480
C     CALCULATE OMEGA(PHI), OMEGA, AND EPSILON-BAR FOR THIS SEGMENT     CNRH0490
      CALL OMEG(SGOP,SGOM,SEPS,PS,S,FDC,RI,NPTSEG,NROSEG,KS,MLIST)      CNRH0500   15
      IF(INTF2) 50,50,40                                                CNRH0510
   40 CONTINUE                                                          CNRH0520
C     TRANSFER VALUES OF OMEGA(PHI) FOR SPECIAL ITERATIONS              CNRH0530
      INOP = MS-1                                                       CNRH0540
      NPTS = NPSEG(IJK)                                                 CNRH0550
      DO 45 I=1,NPTS                                                    CNRH0560
      INOP = INOP+1                                                     CNRH0570
      TT = 0.0                                                          CNRH0580
      DO 43 JJ=1,KS                                                     CNRH0590
   43 TT = TT + SGOP(I,JJ)**2                                           CNRH0600
      OPHI(INOP) = 0.5 * TT                                             CNRH0610
   45 CONTINUE                                                          CNRH0620
C     START LOOP ON POINTS                                              CNRH0630
   50 CONTINUE                                                          CNRH0640
C     BUILD MISCELLANEOUS VALUES FOR THIS POINT                         CNRH0650
      CALL SDAT(EI,HI,FMUI,RI,PS,FLDI,FLDF,S,SGOP,SGOM,SEPS,NROSEG,     CNRH0660
     1 MAXSEG,NPTSEG,III,KS)                                            CNRH0670
C     BUILD FORCES, LAMBDA'S, AND ROTATIONS                             CNRH0680   37
      CALL PVAL(GLAM,MLIST,MIND,ISFLAG,KS,MAXHRM)                       CNRH0690   39
      IF(INTF2)   90,90,85                                              CNRH0700
   85 CONTINUE                                                          CNRH0710
      IF(IJ.EQ.MF) GO TO 90                                             CNRH0720
C     GET VALUE OF N(THETA) FOR SPECIAL ITERATION                       CNRH0730
      INTH = INTH+1                                                     CNRH0740
      FNTH(INTH) = SNGL(GNT(IJH,NONE))                                  CNRH0750
```

```
   90 CONTINUE                                                          CNRH0760
      IF(LOPE-1) 80,80,70                                               CNRH0770
   70 CONTINUE                                                          CNRH0780
      LOPE = 1                                                          CNRH0790
      NONE = 2                                                          CNRH0800
      III = III+1                                                       CNRH0810
      GO TO 50                                                          CNRH0820
   80 CONTINUE                                                          CNRH0830
C     IF NOT THE FIRST POINT OF SEGMENT, TRANSFER                       CNRH0840
      IF(IJ.NE.MS) GO TO 140                                            CNRH0850
C     DETERMINE IF THIS SEGMENT IS THE FIRST SEGMENT                    CNRH0860
      IF(IJK.NE.1) GO TO 120                                            CNRH0870
C     INITIALIZE COUNTER FOR RHV                                        CNRH0880
      IND = 0                                                           CNRH0890
      IBCF = 1                                                          CNRH0900
      GO TO  180                                                        CNRH0910
  120 CONTINUE                                                          CNRH0920
C     COMPUTE LEADING  DISCONTINUITY EQUATION VALUES                    CNRH0930
      NFLAG = 2                                                         CNRH0940
      GO TO  170                                                        CNRH0950
  140 CONTINUE                                                          CNRH0960
C     IS THIS POINT THE LAST POINT OF THE SEGMENT-- 'MF'                CNRH0970
      IF(IJ.NE.MF) GO TO 200                                            CNRH0980
      IF(IJK.NE.NSGMTS) GO TO 160                                       CNRH0990
      IBCF = 2                                                          CNRH1000
      GO TO 180                                                         CNRH1010
  160 CONTINUE                                                          CNRH1020
C     COMPUTE TRAILING DISCONTINUITY EQUATION VALUES                    CNRH1030
      NFLAG = 1                                                         CNRH1040
  170 CONTINUE                                                          CNRH1050
C     BUILD DISCONTINUITY EQUATION VALUES                               CNRH1060
      CALL DVEC(RHK,RHV,ECR,ECY,ECRS,ECYS,SINP,COSP,RI,SCF,FMPO,FNRO,   CNRH1070
     1 FNYO,FNPTO,NFLAG,MAXSEG,MIND,ISFLAG,MLIST,KS,MAXHRM,RSS)         CNRH1080   76
      IF(NFLAG-1) 300,300,200                                           CNRH1090
  180 CONTINUE                                                          CNRH1100
C     BUILD BOUNDARY CONDITION VALUES                                   CNRH1110
      CALL BVEC(IBCF,IBC,BC,RHV,RHK,MAXHRM)                             CNRH1120   80
      IF(IBCF-1) 200,200,300                                            CNRH1130
```

```
  200 CONTINUE                                                          CNRH1140
C     BUILD NEW VALUES OF P                                             CNRH1150
      CALL RANP(RHV)                                                    CNRH1160
C     IF ICHECK = 0, SKIP CHECKOUT PRINTING OPTIONS                     CNRH1170    84
      IF(ICHECK.EQ.0) GO TO 8000                                        CNRH1180
      IF(IOPT(13).EQ.0) GO TO 8000                                      CNRH1190
      IF(IJ.LT.MS+5) GO TO 7000                                         CNRH1200
      IF(IJ.GT.MF-5) GO TO 7000                                         CNRH1210
      IF( (IJ.GT.MFMS2M).AND.(IJ.LT.MFMS2P) ) GO TO 7000                CNRH1220
      GO TO 8000                                                        CNRH1230
 8001 FORMAT(1H1,9HHARMONIC=,I2,2X,7HSEGMENT,I2,2X,6HPOINT ,I2,2X,      CNRH1240
     1 6HNONE =,I2)                                                     CNRH1250
 8002 FORMAT(//8X,1HD,11X,1HK,11X,1HR,9X,2HMU,9X,4HD(S),8X,4HXBAR,8X,   CNRH1260
     1 4HYBAR,8X,4HM(T),7X,5HM(PT))                                     CNRH1270
 8003 FORMAT(1H ,9(1PD12.5))                                            CNRH1280
 8004 FORMAT(4HO  M, 7X,4HZBAR,5X,10HOMEGA(PHI),6X, 5HOMEGA,11X,        CNRH1290
     1 6HN(THE),8X,6HN(P-T),9X,3HEPS)                                   CNRH1300
 8005 FORMAT(1H ,I3,6(1PD14.6))                                         CNRH1310
 8006 FORMAT(1HO,8X,1HM,5X,7HLAMB(2),8X,7HLAMB(3),8X,7HLAMB(4))         CNRH1320
 8007 FORMAT(7X,I3,3(1PD15.7))                                          CNRH1330
 8009 FORMAT(1H ,10X,7HLAMBDA(,I2,3H) =,1PD15.7)                        CNRH1340
 8010 FORMAT(1H ,10X,7HR-STAR(,I2,3H) =,1PD15.7)                        CNRH1350
 8011 FORMAT(1HO,2X,1HM,9X,2HX1,11X,2HX2,11X,2HX3,11X,2HX4,11X,2HX5,    CNRH1360
     1 11X,2HX6,11X,2HX7,11X,2HX8)                                      CNRH1370
 8012 FORMAT(1X,I3,8(1PD13.5) )                                         CNRH1380
 7000 CONTINUE                                                          CNRH1390
C     PRINT VALUES FOR THIS POINT                                       CNRH1400
      IJJ = IJ - MS + 1                                                 CNRH1410
      WRITE(6,8001) MLIST(IJH),IJK,IJJ,NONE                             CNRH1420    103
      IKZ = 1                                                           CNRH1430
      IF(NONE.EQ.1) IKZ = 2                                             CNRH1440
C     CHECK SDATX VALUES.                                               CNRH1450
      WRITE(6,8002)                                                     CNRH1460    107
      WRITE(6,8003) D(IKZ),BK(IKZ),R(IKZ),FM(IKZ),DS(IKZ),XBR(IKZ),     CNRH1470
     1 YBR(IKZ),GMT(IKZ),GMPT(IKZ)                                      CNRH1480
C     CHECK UNKNOWN VALUE TRANSFERS                                     CNRH1490    108
      WRITE(6,8011)                                                     CNRH1500    118
      DO 7825 JJ = 1,KS                                                 CNRH1510
```

```
 7825 WRITE(6,8012)  MLIST(JJ),(X(II,IKZ,JJ),II=1,8)                CNRH1520
C     CHECK FORCES, OMEGA'S, AND ZBAR                               CNRH1530   123
      WRITE(6,8004)                                                 CNRH1540   130
      DO 7850 JJ=1,KS                                               CNRH1550
 7850 WRITE(6,8005) MLIST(JJ),ZBR(JJ,IKZ),GOP(JJ,IKZ),GOM(JJ,IKZ),  CNRH1560
     1 GNT(JJ,IKZ),GNPT(JJ,IKZ),EPS(JJ,IKZ)                         CNRH1570
C     CHECK LAMBDA'S                                                CNRH1580   134
      WRITE(6,8006)                                                 CNRH1590   143
      DO 7900 JJ=1,KS                                               CNRH1600
 7900 WRITE(6,8007) MLIST(JJ),(GLAM(IZ,IKZ,JJ),IZ=1,3)              CNRH1610   147
      WRITE(6,8009) (I,FLAM(I,IKZ),I=1,9)                           CNRH1620
C     CHECK VALUES OF R-STAR                                        CNRH1630   154
      WRITE(6,8010) (I,RSTR(I),I=1,8)                               CNRH1640   161
 7999 CONTINUE                                                      CNRH1650
 8000 CONTINUE                                                      CNRH1660
      IF(NONE-1) 220,220,230                                        CNRH1670
  220 CONTINUE                                                      CNRH1680
      NONE = 2                                                      CNRH1690
      GO TO 240                                                     CNRH1700
  230 NONE = 1                                                      CNRH1710
  240 CONTINUE                                                      CNRH1720
C     INCREMENT POINT COUNTER                                       CNRH1730
      IJ = IJ+1                                                     CNRH1740
      III = III+1                                                   CNRH1750
      GO TO 50                                                      CNRH1760
  300 CONTINUE                                                      CNRH1770
      RETURN                                                        CNRH1780
      END                                                           CNRH1790
```

```
$IBFTC OMEGX    NOLIST,DECK,M94,X27,NODD
```

```
C                 CALCULATE OMEGA(PHI), OMEGA, AND EPSILON-BAR          OMEG0010
      SUBROUTINE OMEG(SGOP,SGOM,SEPS,PS,S,FDC,RI,NPSG,NRSG,KS,MLIST)    OMEG0020
      DIMENSION SGOP(NPSG, 1),SGOM(NPSG, 1),SEPS(NPSG, 1),PS(NRSG, 1),  OMEG0030
     1 S( 1),FDC(3, 1),RI( 1),MLIST( 1)                                 OMEG0040
      DOUBLE PRECISION SINE,COSI,HM,ONE,HALF,R1,Y(3),FD1,FD2,FD3,XDOT,  OMEG0050
     1 YDOT,R1I,HN                                                      OMEG0060
      COMMON/COM51/IJK,IJH,IJ,MS,MF                                     OMEG0070
      COMMON/COM52/SINE,COSI,HM,M,NONE                                  OMEG0080
C     OBTAIN THE FINITE DIFFERENCE COEFFICIENTS                         OMEG0090
      CALL FDIC(FDC,S,MS,MF)                                            OMEG0100
C     SET THE CONSTANT VALUES                                           OMEG0110
      HALF = .5D0                                                       OMEG0120
      ONE = 1.0D0                                                       OMEG0130
C     CALCULATE THE SEGMENT LENGTH                                      OMEG0140
      DEL = ABS(S(MF)-S(MS))                                            OMEG0150
      R1  = DBLE(RI(IJK))                                               OMEG0160
C     CALCULATE THE NUMBER OF POINTS IN THIS SEGMENT                    OMEG0170
      NPTS = MF-MS+1                                                    OMEG0180
      DO 300 IH1=1,KS                                                   OMEG0190
      HN = DBLE(FLOAT(MLIST(IH1)))                                      OMEG0200
C     INITIALIZE S-VECTOR INDEX                                         OMEG0210
      J   = MS-1                                                        OMEG0220
C     SET INITIAL INDICES OF PS TO GET 'U' AND 'V'                      OMEG0230
      NM8 = 1                                                           OMEG0240
      NZ  =  9                                                          OMEG0250
      NP8 = 17                                                          OMEG0260
      MM8 =  2                                                          OMEG0270
      MZ  = 10                                                          OMEG0280
      MP8 = 18                                                          OMEG0290
      DO 200 I=1,NPTS                                                   OMEG0300
C     INCREMENT  S-VECTOR INDEX                                         OMEG0310
      J   = J+1                                                         OMEG0320
C     SET INDICES FOR U(N-1), U(N), AND U(N+1).  LIKEWISE FOR V         OMEG0330
      IF(I.LT.3) GO TO 30                                               OMEG0340
      IF(I.EQ.NPTS) GO TO 30                                            OMEG0350
      NMS = NZ                                                          OMEG0360
      NZ  = NP8                                                         OMEG0370
```

```
      NP8 = NP8+8                                                       OMEG0380
      MM8 = NM8+1                                                       OMEG0390
      MZ  = NZ +1                                                       OMEG0400
      MP8 = NP8+1                                                       OMEG0410
   30 CONTINUE                                                          OMEG0420
C     GET THE ROW NUMBER FOR 'U' AT THIS POINT                          OMEG0430
      KK  = 8*(I-1)                                                     OMEG0440
C     GET U, V, AND W                                                   OMEG0450
      DO 50 IX=1,3                                                      OMEG0460
      KK     = KK+1                                                     OMEG0470
   50 Y(IX) = DBLE(PS(KK,IH1))                                          OMEG0480
C     SET THE FINITE DIFFERENCE COEFFICIENTS FOR THIS POINT             OMEG0490
      FD1 = DBLE(FDC(1,I))                                              OMEG0500
      FD2 = DBLE(FDC(2,I))                                              OMEG0510
      FD3 = DBLE(FDC(3,I))                                              OMEG0520
C     CALCULATE U-DOT AND V-DOT                                         OMEG0530
      XDOT= DBLE(PS(NM8,IH1))*FD1+DBLE(PS(NZ,IH1))*FD2+DBLE(PS(NP8,IH1))OMEG0540
     1 *FD3                                                             OMEG0550
      YDOT= DBLE(PS(MM8,IH1))*FD1+DBLE(PS(MZ,IH1))*FD2+DBLE(PS(MP8,IH1))OMEG0560
     1 *FD3                                                             OMEG0570
C     CALCULATE THE INVERSE OF THE RADIUS AT THIS POINT                 OMEG0580
      R1I = ONE/(DBLE(S(J))*COSI+R1)                                    OMEG0590
C     CALCULATE OMEGA(PHI) FOR THIS POINT AND THIS HARMONIC             OMEG0600
      SGOP(I,IH1) = SNGL(R1I*(HN*Y(3)+Y(1)*SINE))                       OMEG0610
C     CALCULATE OMEGA      FOR THIS POINT AND THIS HARMONIC             OMEG0620
      SGOM(I,IH1) = SNGL(HALF*(XDOT+R1I*(Y(1)*COSI+HN*Y(2))))           OMEG0630
C     CALCULATE EPSILON BAR FOR THIS POINT AND THIS HARMONIC            OMEG0640
      SEPS(I,IH1) = SNGL(YDOT+R1I*(Y(2)*COSI + HN*Y(1) + Y(3)*SINE) )   OMEG0650
  200 CONTINUE                                                          OMEG0660
  300 CONTINUE                                                          OMEG0670
      RETURN                                                            OMEG0680
      END                                                               OMEG0690
```

```
$IBFTC SDATX   NOLIST,DECK,M94,XR7,NODD
```

```
C                    BUILD MISCELLANEOUS VALUES                          SDAT0010
      SUBROUTINE SDAT(EI,HI,FMUI,RI,PS,FLDI,FLDF,S,SGOP,SGOM,SEPS,NRSG, SDAT0020
     1 MAXS,NPSG,III,KS)                                                 SDAT0030
      DIMENSION EI(2, 1),HI(2, 1),FMUI(2, 1),RI( 1),PS(NRSG, 1),         SDAT0040
     1 FLDI(5,MAXS, 1),FLDF(5,MAXS, 1),S( 1), SGOP(NPSG, 1),             SDAT0050
     2 SGOM(NPSG, 1),SEPS(NPSG, 1)                                       SDAT0060
      DOUBLE PRECISION SINE,COSI,HM,D,BK,R,FM,DS,XBR,YBR,RSTR,ZBR,GOP,   SDAT0070
     1 GOM,GNT,GNPT,FLAM,EPS,X                                           SDAT0080
      DOUBLE PRECISION D12,SI,S2,SLPE,SLPH,SLPU,SLPX(2),SLPZ(7),E1,H1,   SDAT0090
     1 U1,ET,HT,UT,X1,Y1,R1                                              SDAT0100
      COMMON/COM51/IJK,IJH,IJ,MS,MF                                      SDAT0110
      COMMON/COM52/SINE,COSI,HM,M,NONE                                   SDAT0120
      COMMON/COM53/D(2),BK(2),R(2),FM(2),DS(2),XBR(2),YBR(2)             SDAT0130
      COMMON/COM54/RSTR(8),ZBR(7,2),GOP(7,2),GOM(7,2),GNT(7,2),          SDAT0140
     1 GNPT(7,2),FLAM(9,2),EPS(7,2)                                      SDAT0150
      COMMON/COM56/X(8,2,7)                                              SDAT0160
      D12 = 8.333333333333333D-2                                         SDAT0170
      IZ = III                                                           SDAT0180
      JZ = III - MS + 1                                                  SDAT0190
      SI = DBLE(S(IZ))                                                   SDAT0200
      R1 = DBLE(RI(IJK))                                                 SDAT0210
      IF(IZ.NE.MS) GO TO 100                                             SDAT0220
C     CALCULATE SEGMENT LENGTH                                           SDAT0230
      DEL = ABS(S(MF)-S(MS))                                             SDAT0240
C     CALCULATE SLOPE OF 'E', 'H', AND 'MU'                              SDAT0250
      SLPE = DBLE( (EI(2,IJK)-EI(1,IJK))/DEL)                            SDAT0260
      SLPH = DBLE( (HI(2,IJK)-HI(1,IJK))/DEL)                            SDAT0270
      SLPU = DBLE( (FMUI(2,IJK)-FMUI(1,IJK))/DEL)                        SDAT0280
C     CALCULATE XBAR AND YBAR SLOPES                                     SDAT0290
      DO 50 I=1,2                                                        SDAT0300
   50 SLPX(I) = DBLE( (FLDF(I,IJK,IJH)-FLDI(I,IJK,IJH))/DEL)             SDAT0310
C     CALCULATE ZBAR SLOPES FOR ALL HARMONICS                            SDAT0320
      DO 70 I=1,KS                                                       SDAT0330
   70 SLPZ(I) = DBLE( (FLDF(3,IJK,I)-FLDI(3,IJK,I))/DEL)                 SDAT0340
C     CALCULATE INITIAL VALUES OF 'E', 'H', 'MU', 'XBAR', AND 'YBAR'     SDAT0350
      ET = DBLE(EI(1,IJK))                                               SDAT0360
      HT = DBLE(HI(1,IJK))                                               SDAT0370
```

```
      UT = DBLE(FMUI(1,IJK))                                            SDAT0380
      X1 = DBLE(FLDI(1,IJK,IJH))                                        SDAT0390
      Y1 = DBLE(FLDI(2,IJK,IJH))                                        SDAT0400
  100 CONTINUE                                                          SDAT0410
C     CALCULATE 'R' FOR THIS POINT                                      SDAT0420
      R(NONE) = SI*COSI+R1                                              SDAT0430
C     CALCULATE 'E', 'H', AND 'MU' FOR THIS POINT                       SDAT0440
      E1 = SLPF*SI+ET                                                   SDAT0450
      H1 = SLPH*SI+HT                                                   SDAT0460
      U1 = SLPU*SI+UT                                                   SDAT0470
C     CALCULATE 'D', 'K', 'MU', 'XBAR', AND 'YBAR' FOR THIS POINT       SDAT0480
      D(NONE) = E1*H1/(1.0D0-U1*U1)                                     SDAT0490
      BK(NONE) = C12*D(NONE)*H1*H1                                      SDAT0500
      FM(NONE) = U1                                                     SDAT0510
      XBR(NONE) = SLPX(1)*SI+X1                                         SDAT0520
      YBR(NONE) = SLPX(2)*SI+Y1                                         SDAT0530
C     CALCULATE 'ZBAR' FOR THIS POINT, ALL HARMONICS                    SDAT0540
      DO 125 I=1,KS                                                     SDAT0550
  125 ZBR(I,NONE) = SLPZ(I)*SI+DBLE(FLDI(3,IJK,I))                      SDAT0560
      IF(IZ.EQ.MF) GO TO 150                                            SDAT0570
      S2 = DBLE(S(IZ+1))                                                SDAT0580
C     CALCULATE CENTRAL DIFFERENCE FOR THIS POINT                       SDAT0590
      DS(NONE) = 1.0D0/(S2-SI)                                          SDAT0600
  150 CONTINUE                                                          SDAT0610
C     TRANSFER 8 UNKNOWNS FOR 7 HARMONICS                               SDAT0620
C     CALCULATE STARTING AND ENDING POINTS                              SDAT0630
      IP = 8*(JZ-1)+1                                                   SDAT0640
      JP= IP+7                                                          SDAT0650
      DO 200 L=1,KS                                                     SDAT0660
      IX =0                                                             SDAT0670
C     TRANSFER EIGHT UNKNOWNS FOR THIS POINT, THIS HARMONIC             SDAT0680
      DO 160 IN=IP,JP                                                   SDAT0690
      IX =IX+1                                                          SDAT0700
      X(IX,NONE,L) = DBLE(PS(IN,L))                                     SDAT0710
  160 CONTINUE                                                          SDAT0720
C     TRANSFER VALUES OF OMEGA, OMEGA(PHI), AND EPSILON BAR             SDAT0730
      GOP(L,NONE) = DBLE(SGOP(JZ,L))                                    SDAT0740
      GOM(L,NONE) = DBLE(SGOM(JZ,L))                                    SDAT0750
```

```
      EPS(L,NONE) = DBLE(SEPS(JZ,L))                                    SDAT0760
  200 CONTINUE                                                          SDAT0770
      RETURN                                                            SDAT0780
      END                                                               SDAT0790
```

```
$IBFTC PVALX   NOLIST,DECK,M94,XR7,NODD
```

```
C     BUILD PRELIMINARY VALUES                                          PVAL0010
C     FOR ALL HARMONICS,GET M(THETA), N(PHI-THETA), AND LAMBDA(I),I=2-4 PVAL0020
C     FOR THIS HARMONIC, GET M(THETA), M(PHI-THETA) AND LAMBDA(I)       PVAL0030
      SUBROUTINE PVAL(GLAM,MLIST, MIND,ISFLAG,KST,MAXH)                 PVAL0040
      DOUBLE PRECISION SINE,COSI,HM,D,BK,R,FM,DS,XBR,YBR,RSTR,ZBR,GOP,  PVAL0050
     1 GOM,GNT,GNPT,FLAM,EPS,GMT,GMPT,X,GLAM                            PVAL0060
      DOUBLE PRECISION HALF,TONE,ONE, OHLF,TWO,TP25,THREE,R1I,R2I,OM,   PVAL0070
     1 OMS,ZLI,CO                                                       PVAL0080
      DIMENSION GLAM(3,2,MAXH),MLIST(MAXH), MIND(7,5,MAXH),             PVAL0090
     1 ISFLAG(3,MAXH)                                                   PVAL0100
      COMMON/COM51/IJK,IJH,IJ,MS,MF                                     PVAL0110
      COMMON/COM52/SINE,COSI,HM,M,NONE                                  PVAL0120
      COMMON/COM53/D(2),BK(2),R(2),FM(2),DS(2),XBR(2),YBR(2)            PVAL0130
      COMMON/COM54/RSTR(8),ZBR(7,2),GOP(7,2),GOM(7,2),GNT(7,2),         PVAL0140
     1 GNPT(7,2),FLAM(9,2),EPS(7,2)                                     PVAL0150
      COMMON/COM55/GMT(2),GMPT(2)                                       PVAL0160
      COMMON/COM56/X(8,2,7)                                             PVAL0170
C     SET VALUES FROM INPUT ARGUMENTS                                   PVAL0180
      KS = KST                                                          PVAL0190
      MAXHPM = MAXH                                                     PVAL0200
C     SAVE CURRENT HARMONIC VALUES                                      PVAL0210
      MTEM   = M                                                        PVAL0220
      IJHTEM = IJH                                                      PVAL0230
C     SET CONSTANT VALUES                                               PVAL0240
      HALF = 0.50D0                                                     PVAL0250
      TONE = 0.75D0                                                     PVAL0260
      ONE  = 1.00D0                                                     PVAL0270
      OHLF = 1.50D0                                                     PVAL0280
      TWO  = 2.00D0                                                     PVAL0290
      TP25 = 2.25D0                                                     PVAL0300
      THREE= 3.0D0                                                      PVAL0310
C     GET SEGMENT VALUES FOR POINT                                      PVAL0320
      R1I  = ONE/R(NONE)                                                PVAL0330
      R2I  = ONE/(R(NONE)*R(NONE))                                      PVAL0340
      OM   = ONE-FM(NONE)                                               PVAL0350
      OMS  = ONE-FM(NONE)*FM(NONE)                                      PVAL0360
      ZLI  = ONE/(ONE+TP25*BK(NONE)*SINE*SINE*R2I/D(NONE))              PVAL0370
```

```
      CO    = OHLF*BK(NONE)*OM*SINE*ZLI*R2I                          PVAL0380
C     GET LAMBDA(I), I= 3 AND 4 FOR ALL HARMONICS                    PVAL0390
      LFLAG = 2                                                      PVAL0400
      CALL LAMB(LFLAG,MLIST,MAXHRM,KS,GLAM,MIND,ISFLAG,IJH)          PVAL0410
C     GET N(THETA) FOR ALL HARMONICS                                 PVAL0420   16
      DO 40 IJH=1,KS                                                 PVAL0430
      HM = DBLE(FLOAT(MLIST(IJH)))                                   PVAL0440
      GNT(IJH,NONE) = FM(NONE)*X(6,NONE,IJH) + D(NONE)*OMS *( R1I*(  PVAL0450
     1 SINE*X(3,NONE,IJH) + HM*X(1,NONE,IJH) + COSI*X(2,NONE,IJH) ) + PVAL0460
     2 HALF*GLAM(2,NONE,IJH))                                        PVAL0470
   40 CONTINUE                                                       PVAL0480
C     GET LAMBDA(2) FOR ALL HARMONICS                                PVAL0490
      LFLAG = 2                                                      PVAL0500
      CALL LAMB(LFLAG,MLIST,MAXHRM,KS,GLAM,MIND,ISFLAG,IJH)          PVAL0510
C     COMPUTE N(PHI-THETA) FOR ALL HARMONICS                         PVAL0520   34
      DO 60 IJH=1,KS                                                 PVAL0530
      HM = DBLE(FLOAT(MLIST(IJH)))                                   PVAL0540
      GNPT(IJH,NONE) = X(5,NONE,IJH)*ZLI - CO*HM*SINE*X(2,NONE,IJH)*R1I PVAL0550
     1 + CO*HM*COSI*X(3,NONE,IJH)*R1I + CO*HM*X(4,NONE,IJH)          PVAL0560
     2 - HALF*GLAM(1,NONE,IJH)*ZLI + TONE*SINE*GLAM(3,NONE,IJH)*CO   PVAL0570
   60 CONTINUE                                                       PVAL0580
  100 CONTINUE                                                       PVAL0590
C     RESTORE VALUES OF M, HM, AND IJH                               PVAL0600
      M    = MTEM                                                    PVAL0610
      HM   = DBLE(FLOAT(M))                                          PVAL0620
      IJH = IJHTEM                                                   PVAL0630
C     GET LAMBDA(I), I=2, 3, AND 4 FOR THIS HARMONIC                 PVAL0640
      DO 200 I=1,3                                                   PVAL0650
      J = I+1                                                        PVAL0660
      FLAM(J,NONE) = GLAM(I,NONE,IJH)                                PVAL0670
  200 CONTINUE                                                       PVAL0680
C     BUILD LAMBDA(I), I=1,5,6,7,8,9                                 PVAL0690
      LFLAG = 3                                                      PVAL0700
      CALL LAMB(LFLAG,MLIST,MAXHRM,KS,GLAM,MIND,ISFLAG,IJH)          PVAL0710
C     CALCULATE M(THETA) AND M(PHI-THETA)                            PVAL0720   63
      GMT(NONE) = FM(NONE)*X(8,NONE,IJH) + BK(NONE)*OMS*R1I*         PVAL0730
     1 (HM*HM*X(3,NONE,IJH)*R1I + HM*SINE*X(1,NONE,IJH)*R1I          PVAL0740
     2 + COSI*X(4,NONE,IJH) )                                        PVAL0750
```

```
      GMPT(NONE) = HALF*BK(NONE) *OM*ZLI*RII* (-(TWO *HM)*X(4,NONE,IJH) PVAL0760
     1 - (TWO*HM)*COSI*X(3,NONE,IJH)*RII+(TWO*HM)*SINE*X(2,NONE,IJH)*RIIPVAL0770
     2 + THREE*SINE*X(5,NONE,IJH)/(D(NONE)*OM)                          PVAL0780
     3 - OHLF*SINE*    (FLAM(2,NONE)/(D(NONE)*OM) + FLAM(4,NONE)) )     PVAL0790
      RETURN                                                            PVAL0800
      END                                                               PVAL0810
```

```
$IBFTC LAMBX   NOLIST,DECK,M94,X37,NODD
```

```
C                 COMPUTE LAMBDA VALUES                                LAMB0010
      SUBROUTINE LAMB(LFLAG,MLIST,MAXHMR,KST,GLAM,MIND,ISFLAG,IJHT)   LAMB0020
      DOUBLE PRECISION HM,RSTR,ZBR,GOP,GOM,GNT,GNPT,FLAM,GMT,GMPT,X,  LAMB0030
     1 SINE,COSI,A,B,C,D,EPS,VALU,GLAM                                LAMB0040
      DIMENSION MLIST(MAXHMR),GLAM(3,2,MAXHMR),MIND(7,5,MAXHMR),      LAMB0050
     1 ISFLAG(3,MAXHMR)                                               LAMB0060
      DIMENSION IABCD(1),IJH(1),IFLAG(1),VALU(1)                      LAMB0070
      COMMON/COM52/SINE,COSI,HM,M,NONE                                LAMB0080
      COMMON/COM54/RSTR(8),ZBR(7,2),GOP(7,2),GOM(7,2),GNT(7,2),       LAMB0090
     1 GNPT(7,2),FLAM(9,2),EPS(7,2)                                   LAMB0100
      COMMON/COM55/GMT(2),GMPT(2)                                     LAMB0110
      COMMON/COM56/X(8,2,7)                                           LAMB0120
      COMMON/COM58/A(7),B(7),C(7),D(7)                                LAMB0130
      IJH = IJHT                                                      LAMB0140
      MAXHRM = MAXHMR                                                 LAMB0150
      KS = KST                                                        LAMB0160
C     INITIALIZE SUMMATION SUBROUTINE                                 LAMB0170
      CALL SUMM(MLIST,MIND,ISFLAG,IFLAG,VALU,KS,IJH,IABCD)            LAMB0180   5
      IF(LFLAG-2) 100,50,175                                          LAMB0190
   50 CONTINUE                                                        LAMB0200
C     CALCULATE LAMBDA(2)                                             LAMB0210
      DO 70 I=1,KS                                                    LAMB0220
      A(I) = X(6,NONE,I) + GNT(I,NONE)                                LAMB0230
      B(I) = GOM(I,NONE)                                              LAMB0240
   70 CONTINUE                                                        LAMB0250
      IFLAG = 0                                                       LAMB0260
      IABCD = 1                                                       LAMB0270
      DO 95 JJ = 1,KS                                                 LAMB0280
      IJH = JJ                                                        LAMB0290
      M = MLIST(IJH)                                                  LAMB0300
      IF(M.EQ.0) GO TO 80                                             LAMB0310
      CALL SUMMU                                                      LAMB0320   31
      GO TO 90                                                        LAMB0330
   80 VALU = 0.0D0                                                    LAMB0340
   90 CONTINUE                                                        LAMB0350
      GLAM(1,NONE,IJH) = VALU                                         LAMB0360
   95 CONTINUE                                                        LAMB0370
```

```
      GO TO 600                                                 LAMB0380
  100 CONTINUE                                                  LAMB0390
C     COMPUTE LAMBDA(I), I=3, 4                                 LAMB0400
      DO 174 JJ = 1,KS                                          LAMB0410
      IJH = JJ                                                  LAMB0420
      M = MLIST(IJH)                                            LAMB0430
C     COMPUTE LAMBDA(3)                                         LAMB0440
      IF(M.EQ.0) GO TO 130                                      LAMB0450
      DO 110 I=1,KS                                             LAMB0460
      A(I) = GOP(I,NONE)                                        LAMB0470
  110 B(I) = GOM(I,NONE)                                        LAMB0480
      IFLAG=0                                                   LAMB0490
      IABCD = 3                                                 LAMB0500
      CALL SUMMU                                                LAMB0510   60
      GO TO 150                                                 LAMB0520
  130 CONTINUE                                                  LAMB0530
      DO 140 I=1,KS                                             LAMB0540
      A(I) = GOP(I,NONE)                                        LAMB0550
  140 C(I) = GOM(I,NONE)                                        LAMB0560
      IFLAG=1                                                   LAMB0570
      IABCD = 4                                                 LAMB0580
      CALL SUMMU                                                LAMB0590   73
  150 CONTINUE                                                  LAMB0600
      GLAM(2,NONE,IJH) = VALU                                   LAMB0610
C     COMPUTE LAMBDA(4)                                         LAMB0620
      IF(M.EQ.0) GO TO 170                                      LAMB0630
      DO 160 I=1,KS                                             LAMB0640
      A(I) = X(4,NONE,I)                                        LAMB0650
  160 B(I) = GOP(I,NONE)                                        LAMB0660
C     FOR M.NE.0, IFLAG IS ALREADY SET                          LAMB0670
      IABCD = 1                                                 LAMB0680
      CALL SUMMU                                                LAMB0690   88
      GO TO 173                                                 LAMB0700
  170 CONTINUE                                                  LAMB0710
      VALU = 0.0D0                                              LAMB0720
  173 GLAM(3,NONE,IJH) = VALU                                   LAMB0730
  174 CONTINUE                                                  LAMB0740
      GO TO 600                                                 LAMB0750
```

```
  175 CONTINUE                                                    LAMB0760
C     COMPUTE LAMBDA(I), I = 6, 7, 8, 9, 1, 5                     LAMB0770
C     COMPUTE LAMBDA(6)                                           LAMB0780
      IFLAG = 1                                                   LAMB0790
      IF(M.EQ.0) GO TO 190                                        LAMB0800
      DO 185 I=1,KS                                               LAMB0810
      A(I) = X(4,NONE,I)                                          LAMB0820
  185 B(I) = GOM(I,NONE)                                          LAMB0830
      IABCD = 3                                                   LAMB0840
      CALL SUMMU                                                  LAMB0850    113
      IF(MLIST(1).EQ.0) VALU = VALU + 2.0D0*A(1)*A(IJH)           LAMB0855
      GO TO 200                                                   LAMB0860
  190 CONTINUE                                                    LAMB0870
      DO 195 I=1,KS                                               LAMB0880
      A(I) = X(4,NONE,I)                                          LAMB0890
  195 C(I) = GOM(I,NONE)                                          LAMB0900
      IABCD = 4                                                   LAMB0910
      CALL SUMMU                                                  LAMB0920    131
      IF(MLIST(1).EQ.0) VALU=VALU+X(4,NONE,1)*X(4,NONE,1)         LAMB0930
  200 CONTINUE                                                    LAMB0940
      FLAM(6,NONE) = VALU                                         LAMB0950
  210 CONTINUE                                                    LAMB0960
C     COMPUTE LAMBDA(7)                                           LAMB0970
      DO 220 I=1,KS                                               LAMB0980
      A(I) = ZBR(I,NONE)                                          LAMB0990
  220 B(I) = X(4,NONE,I)                                          LAMB1000
      IF(M.EQ.0) GO TO 230                                        LAMB1010
C     IFLAG SET IN LAMBDA(6) COMPUTATIONS FOR M.NE.0              LAMB1020
      IABCD = 1                                                   LAMB1030
      CALL SUMMU                                                  LAMB1040    154
      GO TO 250                                                   LAMB1050
  230 CONTINUE                                                    LAMB1060
      VALU = 0.0D0                                                LAMB1070
      ILGZ = ISFLAG(1,IJH)                                        LAMB1080
C     IF ILGZ=0, NO NON-ZERO TERMS EXIST FOR THIS SUMMATION       LAMB1090
      IF(ILGZ.EQ.0) GO TO 245                                     LAMB1100
      DO 240 I=1,ILGZ                                             LAMB1110
      JI = MIND(I,5,IJH)                                          LAMB1120
```

```
  240 VALU = VALU + A(JI)*B(JI)                                    LAMB1130
      VALU = .5D0*VALU                                             LAMB1140
  245 CONTINUE                                                     LAMB1150
      IF(MLIST(1).EQ.0) VALU = VALU+A(1)*B(1)                      LAMB1160
  250 CONTINUE                                                     LAMB1170
      FLAM(7,NONE) = VALU                                          LAMB1180
  260 CONTINUE                                                     LAMB1190
C     COMPUTE LAMBDA(8)                                            LAMB1200
      IF(M.EQ.0) GO TO 270                                         LAMB1210
C     ZBAR WAS PLACED INTO THE 'A' BUFFER IN LAMBDA(7) COMPUTATION LAMB1220
      DO 265 I=1,KS                                                LAMB1230
  265 B(I) = GOP(I,NONE)                                           LAMB1240
      IFLAG=0                                                      LAMB1250
      IABCD = 1                                                    LAMB1260
      CALL SUMMU                                                   LAMB1270   190
      GO TO 275                                                    LAMB1280
  270 VALU = 0.0D0                                                 LAMB1290
  275 CONTINUE                                                     LAMB1300
      FLAM(8,NONE) = VALU                                          LAMB1310
  280 CONTINUE                                                     LAMB1320
C     CALCULATE LAMBDA(9)                                          LAMB1330
      IF(M.EQ.0) GO TO 285                                         LAMB1340
C     ZBAR WAS PLACED INTO 'A' BUFFER IN LAMBDA(7) CALCULATION     LAMB1350
      DO 282 I=1,KS                                                LAMB1354
  282 B(I) = EPS(I,NONE)                                           LAMB1356
      IFLAG = 1                                                    LAMB1360
      IABCD = 1                                                    LAMB1370
      CALL SUMMU                                                   LAMB1380   209
      GO TO 295                                                    LAMB1390
  285 CONTINUE                                                     LAMB1400
      VALU = 0.0D0                                                 LAMB1410
      ILGZ = ISFLAG(1,IJH)                                         LAMB1420
C     IF ILGZ=0, NO NON-ZERO TERMS EXIST FOR THIS SUMMATION        LAMB1430
      IF(ILGZ.EQ.0) GO TO 293                                      LAMB1440
      DO 290 I=1,ILGZ                                              LAMB1450
      JI = MIND(I,5,IJH)                                           LAMB1460
  290 VALU = VALU + 0.5D0 * A(JI) * EPS(JI,NONE)                   LAMB1470
  293 CONTINUE                                                     LAMB1480
```

```
      IF(MLIST(1).EQ.0) VALU = VALU + A(1) * EPS(1,NONE)          LAMB1490
  295 CONTINUE                                                    LAMB1500
      FLAM(9,NONE) = VALU                                         LAMB1510
  300 CONTINUE                                                    LAMB1520
C     CALCULATE LAMBDA(1)                                         LAMB1530
      IFLAG = 0                                                   LAMB1540
      IF(M.EQ.0) GO TO 320                                        LAMB1550
      DO 310 I=1,KS                                               LAMB1560
      A(I) = GNPT(I,NONE)                                         LAMB1570
      B(I) = X(4,NONE,I)                                          LAMB1580
      C(I) = GNT(I,NONE)                                          LAMB1590
  310 D(I) = GOP(I,NONE)                                          LAMB1600
      IABCD = 2                                                   LAMB1610
      CALL SUMMU                                                  LAMB1620   252
      GO TO 325                                                   LAMB1630
  320 VALU = 0.0D0                                                LAMB1640
  325 CONTINUE                                                    LAMB1650
      FLAM(1,NONE) = VALU                                         LAMB1660
  350 CONTINUE                                                    LAMB1670
  400 CONTINUE                                                    LAMB1680
C     CALCULATE LAMBDA(5)                                         LAMB1690
      DO 410 I=1,KS                                               LAMB1700
      A(I) = X(4,NONE,I)                                          LAMB1710
      B(I) = X(6,NONE,I)                                          LAMB1720
      C(I) = GNPT(I,NONE)                                         LAMB1730
  410 D(I) = GOP(I,NONE)                                          LAMB1740
      IF(M.EQ.0) GO TO 430                                        LAMB1750
      IFLAG = 1                                                   LAMB1760
      IABCD = 2                                                   LAMB1770
      CALL SUMMU                                                  LAMB1780   278
      GO TO 450                                                   LAMB1790
  430 CONTINUE                                                    LAMB1800
C     IFLAG WAS SET IN LAMBDA(1) CALCULATION                      LAMB1810
      IABCD = 4                                                   LAMB1820
      CALL SUMMU                                                  LAMB1830   282
      VALU = VALU +B(1)*A(1)                                      LAMB1840
  450 CONTINUE                                                    LAMB1850
      FLAM(5,NONE) = VALU                                         LAMB1860
```

```
600 CONTINUE                                    LAMB1870
    RETURN                                      LAMB1880
    END                                         LAMB1890
```

```
$IBFTC SUMMX   NOLIST,DECK,M94,XR7,NODD
```

```
C     SUBROUTINE TO PERFORM THE SUMMATIONS                          SUMM0010
      SUBROUTINE SUMM(MLIST,MIND,ISFLAG,IFLG,VAUL,KST,IJHT,IABCD)   SUMM0020
      DIMENSION MLIST( 1),MIND(7,5, 1),ISFLAG(3, 1)                 SUMM0030
      COMMON/COM58/A(7),B(7),C(7),D(7)                              SUMM0040
      DOUBLE PRECISION A,B,C,D,FLAG,VALUT,VAUL,VALU                 SUMM0050
      RETURN                                                        SUMM0060
      ENTRY SUMMU                                                   SUMM0070
C     INITIALIZE FLAGS AND VALUES                                   SUMM0080
      IFLAG = IFLG                                                  SUMM0090
      IJH  = IJHT                                                   SUMM0100
      VALU = 0.0D0                                                  SUMM0110
      VALUT= 0.0D0                                                  SUMM0120
      FLAG = 1.0D0                                                  SUMM0130
      IF(IFLAG.EQ.1) FLAG = -1.0D0                                  SUMM0140
      IMPL = ISFLAG(2,IJH)                                          SUMM0150
      IMML = ISFLAG(3,IJH)                                          SUMM0160
      GO TO (200,400,600,800), IABCD                                SUMM0170
  200 CONTINUE                                                      SUMM0180
C                 PERFORM 'A' SUMMATION                             SUMM0190
C     IF FIRST HARMONIC.NE.0, CONSTANT TERMS DON'T EXIST            SUMM0200
      IF(MLIST(1).NE.0) GO TO 250                                   SUMM0210
C     ADD CONSTANT TERMS                                            SUMM0220
      VALU = VALU+A(1)*B(IJH)                                       SUMM0230
      IF(IFLAG.EQ.0) GO TO 250                                      SUMM0240
      VALU = VALU+A(IJH)*B(1)                                       SUMM0250
  250 CONTINUE                                                      SUMM0260
C     CALCULATE SUM OF FORM A(L)*B(M+L)+ OR - A(M+L)*B(L)           SUMM0270
C     IF IMPL=0, NO NON-ZERO TERMS EXIST IN SUMMATION               SUMM0280
      IF(IMPL.EQ.0) GO TO 300                                       SUMM0290
      DO 290 I=1,IMPL                                               SUMM0300
      JI =  MIND(I,2,IJH)                                           SUMM0310
      JM =  MIND(I,1,IJH)                                           SUMM0320
      VALUT = VALUT+A(JI)*B(JM)-FLAG*A(JM)*B(JI)                    SUMM0330
  290 CONTINUE                                                      SUMM0340
      VALU = VALU+.5D0*VALUT                                        SUMM0350
  300 CONTINUE                                                      SUMM0360
C     CALCULATE SUM OF FORM A(L)*B(M-L)+A(M-L)*B(L)                 SUMM0370
```

```
C     IF IMML=0, NO NON-ZERO TERMS EXIST IN SUMMATION                  SUMM0380
      IF(IMML.EQ.0) GO TO 350                                          SUMM0390
      VALUT = 0.0D0                                                    SUMM0400
      DO 340 I=1,IMML                                                  SUMM0410
      JI =  MIND(I,4,IJH)                                              SUMM0420
      JM =  MIND(I,3,IJH)                                              SUMM0430
      VALUT = VALUT+A(JI)*B(JM)+A(JM)*B(JI)                            SUMM0440
  340 CONTINUE                                                         SUMM0450
      VALU = VALU+.25D0*VALUT                                          SUMM0460
  350 CONTINUE                                                         SUMM0470
      GO TO 1000                                                       SUMM0480
  400 CONTINUE                                                         SUMM0490
C                 PERFORM 'B' SUMMATION                                SUMM0500
C     IF FIRST HARMONIC.NE.0, CONSTANT TERMS DON'T EXIST               SUMM0510
      IF(MLIST(1).NE.0) GO TO 450                                      SUMM0520
      VALU = B(1) * A(IJH)                                             SUMM0530
      IF(IFLAG) 420,420,430                                            SUMM0532
  420 VALU = VALU + C(1) * D(IJH)                                      SUMM0534
      GO TO 450                                                        SUMM0536
  430 VALU = VALU + B(IJH) * A(1)                                      SUMM0538
  450 CONTINUE                                                         SUMM0540
C     CALCULATE SUMMATION OF THE FORM                                  SUMM0550
C     B(L)*A(M+L)+C(L)*D(M+L)-FLAG*(B(M+L)*A(L)+C(M+L)*D(L))           SUMM0560
C     IF IMPL=0, NO NON-ZERO TERMS EXIST IN THIS SUMMATION             SUMM0570
      IF(IMPL.EQ.0) GO TO 500                                          SUMM0580
      DO 490 I=1,IMPL                                                  SUMM0590
      JI =  MIND(I,2,IJH)                                              SUMM0600
      JM =  MIND(I,1,IJH)                                              SUMM0610
      VALUT = VALUT+B(JI)*A(JM)+C(JI)*D(JM)-FLAG*(B(JM)*A(JI)+         SUMM0620
     1 C(JM)*D(JI))                                                    SUMM0630
  490 CONTINUE                                                         SUMM0640
      VALU = VALU+0.5D0*VALUT                                          SUMM0650
  500 CONTINUE                                                         SUMM0660
C     CALCULATE SUMMATION OF THE FORM                                  SUMM0670
C     B(L)*A(M-L)+C(L)*D(M-L)+B(M-L)*A(L)+C(M-L)*D(L)                  SUMM0680
C     IF IMML=0, NO NON-ZERO TERMS EXIST IN THIS SUMMATION             SUMM0690
      IF(IMML.EQ.0) GO TO 550                                          SUMM0700
      VALUT = 0.0D0                                                    SUMM0710
```

```
      DO 540 I=1,IMML                                                   SUMM0720
      JI =  MIND(I,4,IJH)                                               SUMM0730
      JM =  MIND(I,3,IJH)                                               SUMM0740
      VALUT = VALUT+B(JI)*A(JM)+FLAG*C(JI)*D(JM)+B(JM)*A(JI)+           SUMM0750
     1 FLAG*C(JM)*D(JI)                                                 SUMM0760
  540 CONTINUE                                                          SUMM0770
      VALU = VALU + .25D0*VALUT                                         SUMM0780
  550 CONTINUE                                                          SUMM0790
      GO TO 1000                                                        SUMM0800
  600 CONTINUE                                                          SUMM0810
C                 PERFORM 'C' SUMMATION                                 SUMM0820
C     CALCULATE SUMMATION OF THE FORM A(L)*A(M+L) + B(L)*B(M+L)         SUMM0830
C     IF IMPL=0, NO NON-ZERO TERMS EXIST IN THIS SUMMATION              SUMM0840
      IF(IMPL.EQ.0) GO TO 650                                           SUMM0850
      DO 640 I=1,IMPL                                                   SUMM0860
      JI =  MIND(I,2,IJH)                                               SUMM0870
      JM =  MIND(I,1,IJH)                                               SUMM0880
      VALU  = VALU +A(JI)*A(JM)+B(JI)*B(JM)                             SUMM0890
  640 CONTINUE                                                          SUMM0900
  650 CONTINUE                                                          SUMM0910
C     CALCULATE SUMMATION OF THE FORM A(L)*A(M-L)+B(L)*B(M-L)           SUMM0920
C     IF IMML=0, NO NON-ZERO TERMS EXIST IN THIS SUMMATION              SUMM0930
      IF(IMML.EQ.0) GO TO 700                                           SUMM0940
      DO 690 I=1,IMML                                                   SUMM0950
      JI =  MIND(I,4,IJH)                                               SUMM0960
      JM =  MIND(I,3,IJH)                                               SUMM0970
      VALUT = VALUT+A(JI)*A(JM)+FLAG*B(JI)*B(JM)                        SUMM0980
  690 CONTINUE                                                          SUMM0990
      VALU = VALU - .5D0*FLAG*VALUT                                     SUMM1000
  700 CONTINUE                                                          SUMM1010
      GO TO 1000                                                        SUMM1020
  800 CONTINUE                                                          SUMM1030
C                 PERFORM 'D' SUMMATION                                 SUMM1040
C     IF IFLAG=0, A, B, C, AND D ARE DISTINCT                           SUMM1050
C     IF IFLAG=1, A=B AND C=D, HOWEVER ONLY A AND C ARE INPUT           SUMM1060
      ILGZ  = ISFLAG(1,IJH)                                             SUMM1070
C     IF ILGZ=0, NO NON-ZERO TERMS EXIST IN THIS SUMMATION              SUMM1080
      IF(ILGZ.EQ.0) GO TO 950                                           SUMM1090
```

```
      IF(IFLAG.EQ.0) GO TO 900                                  SUMM1100
      KS = KST                                                  SUMM1110
      DO 850 I=1,KS                                             SUMM1120
      B(I) = A(I)                                               SUMM1130
      D(I) = C(I)                                               SUMM1140
  850 CONTINUE                                                  SUMM1150
  900 CONTINUE                                                  SUMM1160
C     CALCULATE SUMMATION OF THE FORM A(L)*B(L)+C(L)*D(L)       SUMM1170
      DO 940 I=1,ILGZ                                           SUMM1180
      JI =  MIND(I,5,IJH)                                       SUMM1190
      VALU = VALU+A(JI)*B(JI)+C(JI)*D(JI)                       SUMM1200
  940 CONTINUE                                                  SUMM1210
      VALU = 0.5D0*VALU                                         SUMM1220
  950 CONTINUE                                                  SUMM1230
 1000 CONTINUE                                                  SUMM1240
      VAUL = VALU                                               SUMM1250
      RETURN                                                    SUMM1260
      END                                                       SUMM1270
```

```
$IBFTC BVECX   NOLIST,DECK,M94,XR7,NODD
```

```
C     BUILD BOUNDARY CONDITION VALUES                                   BVEC0010
      SUBROUTINE BVEC(IBCG,IBC,BC,RHV,RHK,MAXH)                         BVEC0020
      DIMENSION IBC(4,MAXH,2),BC(4,MAXH,2),RHV(1),RHK(1)                BVEC0030
      DOUBLE PRECISION SINE,COSI,HM,D,BK,R,FM,DS,XBR,YBR,X              BVEC0040
      DOUBLE PRECISION PI,BC1,FLAG,TEMP                                 BVEC0050
      COMMON/COM51/IJK,IJH,IJ,MS,MF                                     BVEC0060
      COMMON/COM52/SINE,COSI,HM,M,NONE                                  BVEC0070
      COMMON/COM53/D(2),BK(2),R(2),FM(2),DS(2),XBR(2),YBR(2)            BVEC0080
      COMMON/COM56/X(8,2,7)                                             BVEC0090
      COMMON/COM59/IND,INDK                                             BVEC0100
      PI = 3.141592653589793 D0                                         BVEC0110
      NONZ = 1                                                          BVEC0120
      IF(NONE.EQ.1) NONZ = 2                                            BVEC0130
      IBCF = IBCG                                                       BVEC0140
      DO 100 IK=1,4                                                     BVEC0150
      IND = IND+1                                                       BVEC0160
      MFG = IBC(IK,IJH,IBCF)                                            BVEC0170
      BC1 = DBLE( BC(IK,IJH,IBCF))                                      BVEC0180
      FLAG = 1.0D0                                                      BVEC0190
      TEMP = 0.0D0                                                      BVEC0200
      GO TO (30,30,30,30,55,65,40,40,40,40,75,85,70,80,50,60), MFG      BVEC0210
   40 MFG = MFG-2                                                       BVEC0220
   30 CONTINUE                                                          BVEC0230
      TEMP = BC1 - X(MFG,NONZ,IJH)                                      BVEC0240
      GO TO 90                                                          BVEC0250
   50 CONTINUE                                                          BVEC0260
      TEMP = - X(1,NONZ,IJH)                                            BVEC0270
   55 CONTINUE                                                          BVEC0280
      TEMP = TEMP + BC1 - X(2,NONZ,IJH)*COSI - X(3,NONZ,IJH)*SINE       BVEC0290
      GO TO 90                                                          BVEC0300
   60 CONTINUE                                                          BVEC0310
      FLAG = 1.0D0/R(NONZ)                                              BVEC0320
      TEMP = X(4,NONZ,IJH)                                              BVEC0330
   65 CONTINUE                                                          BVEC0340
      TEMP = TEMP + BC1 -                                               BVEC0350
     1 FLAG*( X(2,NONZ,IJH)*SINE - X(3,NONZ,IJH)*COSI )                 BVEC0360
      GO TO 90                                                          BVEC0370
```

```
   70 CONTINUE                                                          BVEC0380
      FLAG = PI*R(NONZ)**2                                              BVEC0390
      TEMP =-PI * R(NONZ) * X(8,NONZ,IJH)                               BVEC0400
   75 CONTINUE                                                          BVEC0410
      TEMP = TEMP + BC1 -                                               BVEC0420
     1 FLAG*( X(6,NONZ,IJH)*SINE - X(7,NONZ,IJH)*COSI )                 BVEC0430
      GO TO 90                                                          BVEC0440
   80 CONTINUE                                                          BVEC0450
      FLAG = PI * R(NONZ)                                               BVEC0460
      TEMP = PI * R(NONZ) * X(5,NONZ,IJH)                               BVEC0470
   85 CONTINUE                                                          BVEC0480
      TEMP = TEMP + BC1 -                                               BVEC0490
     1 FLAG*( X(6,NONZ,IJH)*COSI + X(7,NONZ,IJH)*SINE )                 BVEC0500
   90 CONTINUE                                                          BVEC0510
      RHV(IND) = SNGL(TEMP)                                             BVEC0520
  100 CONTINUE                                                          BVEC0530
      IF(IBCF-1) 200,200,110                                            BVEC0540
  110 CONTINUE                                                          BVEC0550
C     TRANSFER RHV TO RHK                                               BVEC0560
      DO 150 I=1,INC                                                    BVEC0570
      INDK = INDK+1                                                     BVEC0580
      RHK(INDK) = RHV(I)                                                BVEC0590
  150 CONTINUE                                                          BVEC0600
  200 CONTINUE                                                          BVEC0610
      RETURN                                                            BVEC0620
      END                                                               BVEC0630
```

```
$IBFTC DVECX   NOLIST,DECK,M94,XR7,NODD
```

```
C                 BUILD DISCONTINUITY EQUATION VALUES                   DVEC0010
C     NFLAG = 1---BUILD TRAILING DISCONTINUITY EQUATION VALUES          DVEC0020
C     NFLAG = 2---BUILD LEADING  DISCONTINUITY EQUATION VALUES          DVEC0030
      SUBROUTINE DVEC(RHK,RHV,ECR,ECY,ECRS,ECYS,SINP,COSP,RI,SCF,       DVEC0040
     1 FMPO,FNRO,FNYO,FNPTO,NFLAG,MAXS,MIND,ISFLAG,MLIST,KS,MAXHRM,RSS) DVEC0050
      DIMENSION RHK( 1),RHV( 1),ECR( 1),ECY( 1),ECRS(MAXS, 1),          DVEC0060
     1 ECYS(MAXS, 1),SINP(1),COSP(1),RI(1),SCF(16,MAXS,1),              DVEC0070
     2 FMPO(2,MAXS,1),FNRO(2,MAXS,1),FNYO(2,MAXS,1),FNPTO(2,MAXS,1),    DVEC0080
     3 X1(8),AF(4,4),TEMP(8),MIND(7,5,1),ISFLAG(3,1),MLIST(1),          DVEC0090
     4 RSS(MAXS, 1)                                                     DVEC0095
      DOUBLE PRECISION SIND,COSD,HM,D,BK,R,FM,DS,XBR,YBR,X,A,B,C,DZ     DVEC0100
      DOUBLE PRECISION X1,TEMP,RA,RB,RIN,S2,C2,SPSI,CPSI,ER,EY,HNPTO,   DVEC0110
     1 HNRO,HNYO,HMPO,R1I,VALU,SINE,COSI                                DVEC0120
      EQUIVALENCE(C(1),X1(1))                                           DVEC0130
      COMMON/COM51/IJK,IJH,IJ,MS,MF                                     DVEC0140
      COMMON/COM52/SIND,COSD,HM,M,NONE                                  DVEC0150
      COMMON/COM53/D(2),BK(2),R(2),FM(2),DS(2),XBR(2),YBR(2)            DVEC0160
      COMMON/COM56/X(8,2,7)                                             DVEC0170
      COMMON/COM57/ERS,EYS,R1,RBB                                       DVEC0175
      COMMON/COM58/A(7),B(7),C(7),DZ(7)                                 DVEC0180
      COMMON/COM59/IND,INDK                                             DVEC0190
      IJKP = IJK+1                                                      DVEC0200
      NONZ = 1                                                          DVEC0210
      IF(NONE.EQ.1) NONZ = 2                                            DVEC0220
C     GET VALUES OF THE 8 UNKNOWNS FOR THIS POINT AND HARMONIC          DVEC0230
      DO 50 I=1,8                                                       DVEC0240
      X1(I) = X(I,NONZ,IJH)                                             DVEC0250
   50 CONTINUE                                                          DVEC0260
      IF(NFLAG-1) 100,100,400                                           DVEC0270
  100 CONTINUE                                                          DVEC0280
C     SET VALUES OF SIN(PHI) AND COS(PHI)                               DVEC0290
      SINE = SIND                                                       DVEC0300
      COSI = COSD                                                       DVEC0310
C     GET VALUES OF R(LAST) AND R(FIRST)                                DVEC0320
      RA = R(NONZ)                                                      DVEC0330
      RB = DBLE(RI(IJKP))                                               DVEC0340
      RIN = RB/RA                                                       DVEC0350
```

```
C     CALCULATE K-COEFFICIENTS                                  DVEC0360
      ERS = ECRS(IJK,IJH)                                       DVEC0370
      EYS = ECYS(IJK,IJH)                                       DVEC0380
      I = 0                                                     DVEC0390
      DO 120 IO=1,4                                             DVEC0400
      DO 120 IP=1,4                                             DVEC0410
      I = I+1                                                   DVEC0420
  120 AF(IO,IP) = SCF(I,IJK,IJH)                                DVEC0430
      R1 = SNGL(R(NCNE))                                        DVEC0440
      RBB = RSS(IJK,IJH) / R1                                   DVEC0445
      CALL KCCG(AF)                                             DVEC0450
C     CALCULATE SINE(PSI) AND COSINE(PSI)                       DVEC0460   34
      S2 = DBLE(SINP(IJKP))                                     DVEC0470
      C2 = DBLE(COSP(IJKP))                                     DVEC0480
      SPSI = S2*COSI-C2*SINE                                    DVEC0490
      CPSI = C2*COSI+S2*SINE                                    DVEC0500
C     GET E(R) AND E(Y)                                         DVEC0510
      ER = DBLE(ECR(IJK))                                       DVEC0520
      EY = DBLE(ECY(IJK))                                       DVEC0530
C     BUILD VALUES                                              DVEC0540
      DO 160 I=1,4                                              DVEC0550
  160 TEMP(I) = 0.0D0                                           DVEC0560
C     SUBTRACT TERMS INVOLVING 'K' COEFFICIENTS                 DVEC0570
      DO 200 I=1,4                                              DVEC0580
      DO 200 J=1,4                                              DVEC0590
      TEMP(I) = TEMP(I)- DBLE(AF(I,J))*X1(J)                    DVEC0600
  200 CONTINUE                                                  DVEC0610
C     ADD THE REMAINING TERMS TO THE VALUES                     DVEC0620
      HNPTO= DBLE(FNPTO(1,IJK,IJH))                             DVEC0630
      HNRO = DBLE(FNRO(1,IJK,IJH))                              DVEC0640
      HNYO = DBLE(FNYO(1,IJK,IJH))                              DVEC0650
      HMPO = DBLE(FMPO(1,IJK,IJH))                              DVEC0660
      R1I  = 1.0D0/RA                                           DVEC0670
      TEMP(1) = TEMP(1)+ HNPTO - X1(5)                          DVEC0680
      TEMP(2) = TEMP(2)+ HNRO*COSI+HNYO*SINE - X1(6)            DVEC0690
      TEMP(3) = TEMP(3)+ HNRO*SINE - HNYO*COSI - X1(7)          DVEC0700
      TEMP(4) = TEMP(4)+ HMPO- X1(8)                            DVEC0710
      TEMP(5) = -X1(1)*(1.0D0 + ER*R1I)                         DVEC0720
```

```
     1 - X1(2)*( HM*R1I*(ER*COSI+EY*SINE) )                          DVEC0730
     2 - X1(3)*( HM*R1I*(ER*SINE-EY*COSI) )                          DVEC0740
      TEMP(6) = - X1(2) - X1(4)*(ER*SINE-EY*COSI)                    DVEC0750
      TEMP(7) = - X1(3) + X1(4)*(ER*COSI+EY*SINE)                    DVEC0760
      TEMP(8) = - X1(4)                                              DVEC0770
      GO TO 600                                                      DVEC0780
  400 CONTINUE                                                       DVEC0790
C     COMPLETE THE CALCULATION OF THE DISCONTINUITY EQUATION VALUES  DVEC0800
      TEMP(1) = TEMP(1) + RIN*X1(5)                                  DVEC0810
      TEMP(2) = TEMP(2) + RIN*( X1(6)*CPSI + X1(7)*SPSI)             DVEC0820
      TEMP(3) = TEMP(3) + RIN*( -X1(6)*SPSI + X1(7)*CPSI)            DVEC0830
C     CALCULATE  N(PHI) * OMEGA(THETA)                               DVEC0840
C     LET A = N(PHI) AND B = OMEGA(THETA)                            DVEC0850
      DO 450 I=1,KS                                                  DVEC0860
      A(I) = X(6,NONZ,I)                                             DVEC0870
  450 B(I) = X(4,NONZ,I)                                             DVEC0880
      VALU = 0.0D0                                                   DVEC0890
      IF(M.NE.0) GO TO 475                                           DVEC0900
      ILGZ = ISFLAG(1,IJH)                                           DVEC0910
      DO 460 I=1,ILGZ                                                DVEC0920
      JI = MIND(I,5,IJH)                                             DVEC0930
  460 VALU = VALU + A(JI)*B(JI)                                      DVEC0940
      VALU = 0.5D0*VALU + A(1)*B(1)                                  DVEC0950
      GO TO 500                                                      DVEC0960
  475 CONTINUE                                                       DVEC0970
C     INITIALIZE SUMMATION SUBROUTINE                                DVEC0980
      CALL SUMM(MLIST,MIND,ISFLAG,IFLAG,VALU,KS,IJH,IABCD)           DVEC0990    92
      IFLAG = 1                                                      DVEC1000
      IABCD = 1                                                      DVEC1010
      CALL SUMMU                                                     DVEC1020    96
  500 CONTINUE                                                       DVEC1030
      TEMP(4) = TEMP(4) -RIN*(ER*COSI+EY*SINE)*VALU                  DVEC1040
     1 +RIN*((ER*SINE-EY*COSI)*X1(6) - (ER*COSI+EY*SINE)*X1(7) +X1(8)) DVEC1050
      TEMP(5) = TEMP(5) + X1(1)                                      DVEC1060
      TEMP(6) = TEMP(6) + X1(2)*CPSI + X1(3)*SPSI                    DVEC1070
      TEMP(7) = TEMP(7) - X1(2)*SPSI + X1(3)*CPSI                    DVEC1080
      TEMP(8) = TEMP(8) + X1(4)                                      DVEC1090
C     TRANSFER 4 TRAILING D.E. VALUES TO RHV                         DVEC1100
```

```
      DO 525 I=1,4                                              DVEC1110
      IND = IND+1                                               DVEC1120
      RHV(IND) = SNGL(TEMP(I))                                  DVEC1130
  525 CONTINUE                                                  DVEC1140
C     SAVE RHV IN RHK VECTOR                                    DVEC1150
      DO 550 I=1,IND                                            DVEC1160
      INDK = INDK+1                                             DVEC1170
  550 RHK(INDK) = RHV(I)                                        DVEC1180
C     INITIALIZE FLAG FOR RHV                                   DVEC1190
      IND = 0                                                   DVEC1200
C     TRANSFER LEADING D.E. VALUES TO RHV                       DVEC1210
      DO 575 I=5,8                                              DVEC1220
      IND = IND+1                                               DVEC1230
  575 RHV(IND) = SNGL(TEMP(I))                                  DVEC1240
  600 CONTINUE                                                  DVEC1250
      RETURN                                                    DVEC1260
      END                                                       DVEC1270
```

```
$IBFTC KCOGX   NOLIST,DECK,M94,XR7,NODD
```

```
C     BUILD 'K' COEFFICIENTS IN REGION 5                                KCOG0010
      SUBROUTINE KCOG(AF)                                               KCOG0020
      DOUBLE PRECISION SINT,COST,HM,AZ,BZ,CZ,DZ                         KCOG0030
      DIMENSION AF(4,4), A(4,4), C(4,4)                                 KCOG0040
      COMMON/COM52/SINT,COST,HM,M,NONE                                  KCOG0050
      COMMON/COM57/ERS,EYS,R1,RBB                                       KCOG0052
      COMMON/COM58/AZ(7),BZ(7),CZ(7),DZ(7)                              KCOG0054
      EQUIVALENCE (A(1,1),AZ(1)),(FM,BZ(2)),(SINE,BZ(3)),(COSI,BZ(4)),  KCOG0056
     1 (R1I,BZ(5))                                                      KCOG0058
      FM=M                                                              KCOG0060
      SINE = SNGL(SINT)                                                 KCOG0070
      COSI = SNGL(COST)                                                 KCOG0080
      DO 10 I=1,4                                                       KCOG0090
      DO 10 J=1,4                                                       KCOG0100
   10 A(I,J) = 0.                                                       KCOG0110
      R1I=1./R1                                                         KCOG0120
C     BUILD 'A' MATRIX                                                  KCOG0130
      A(1,1) = 1. + ERS*R1I                                             KCOG0140
      A(1,2) =  FM*R1I*(ERS*COSI+EYS*SINE)                              KCOG0150
      A(1,3) =  FM*R1I*(ERS*SINE-EYS*COSI)                              KCOG0160
      A(2,2) =  SINE                                                    KCOG0170
      A(2,3) = -COSI                                                    KCOG0180
      A(2,4) =  ERS                                                     KCOG0190
      A(3,2) =  COSI                                                    KCOG0200
      A(3,3) =  SINE                                                    KCOG0210
      A(3,4) = -EYS                                                     KCOG0220
      A(4,4) =  1.                                                      KCOG0230
C     CALCULATE 'K' COEFFICIENTS                                        KCOG0240
      DO 20 I=1,4                                                       KCOG0250
      DO 20 J=1,4                                                       KCOG0260
      C(I,J) =0.                                                        KCOG0270
      DO 20 K=1,4                                                       KCOG0280
   20 C(I,J) =C(I,J)+AF(I,K)*A(K,J)                                     KCOG0290
      DO 40 I=1,4                                                       KCOG0300
      DO 40 J=1,4                                                       KCOG0310
      AF(I,J)=0.                                                        KCOG0320
      DO 30 K=1,4                                                       KCOG0330
```

```
30 AF(I,J)=AF(I,J)+ A(K,I) * C(K,J)                    KCOG0340
40 AF(I,J)=R38*AF(I,J)                                 KCOG0350
   RETURN                                              KCOG0360
   END                                                 KCOG0370
```

```
$IBFTC RANPX   NOLIST,DECK,M94,XR7,NODD
```

```
C     BUILD R-STAR AND NEW RIGHT HAND VECTOR FOR ONE POINT                RANP0010
      SUBROUTINE RANP(RHV)                                                RANP0020
      DOUBLE PRECISION SINE,COSI,HM,D,BK,R,FM,DS,XBR,YBR,RSTR,ZBR,GOP,    RANP0030
     1 GCM,GNT,GNPT,FLAM,EPS,GMT,GMPT,X                                   RANP0040
      DOUBLE PRECISION HALF,ONE,TWO,THREE,ONP(2),ONM(2),R1I,R2I,          RANP0050
     1 CM,OMS,TPS(8),R1,FORT                                              RANP0060
      DIMENSION RHV( 1),NNN(2)                                            RANP0070
      COMMON/COM51/IJK,IJH,IJ,MS,MF                                       RANP0080
      COMMON/COM52/SINE,COSI,HM,M,NONE                                    RANP0090
      COMMON/COM53/D(2),BK(2),R(2),FM(2),DS(2),XBR(2),YBR(2)              RANP0100
      COMMON/COM54/RSTR(8),ZBR(7,2),GOP(7,2),GCM(7,2),GNT(7,2),           RANP0110
     1 GNPT(7,2),FLAM(9,2),EPS(7,2)                                       RANP0120
      COMMON/COM55/GMT(2),GMPT(2)                                         RANP0130
      COMMON/COM56/X(8,2,7)                                               RANP0140
      COMMON/COM59/IND,INDK                                               RANP0150
C     SET INITIAL FLAGS AND VALUES                                        RANP0160
      FORT = 0.25D0                                                       RANP0170
      HALF = 0.50D0                                                       RANP0180
      ONE  = 1.00D0                                                       RANP0190
      TWO  = 2.00D0                                                       RANP0200
      THREE= 3.00D0                                                       RANP0210
C     ZERO OUT R-STAR AND TEMPORARY STORAGE FOR NEW RIGHT HAND VECTOR     RANP0220
      DO 50 I = 1,8                                                       RANP0230
      TPS(I)  = 0.0D0                                                     RANP0240
   50 RSTR(I) = 0.0D0                                                     RANP0250
C     SET INDICES                                                         RANP0260
      IF(NONE-1) 75,75,100                                                RANP0270
   75 CONTINUE                                                            RANP0280
C     IF NONE=1, VALUES FOR 'IJ' POINT ARE IN (NONE=2) POSITION           RANP0290
      NONZ   = 2                                                          RANP0300
      NNN(1) = 2                                                          RANP0310
      NNN(2) = 1                                                          RANP0320
      GO TO 150                                                           RANP0330
  100 CONTINUE                                                            RANP0340
C     IF NONE=2, VALUES FOR IJ POINT ARE IN (NONE=1) POSITION             RANP0350
      NONZ   = 1                                                          RANP0360
      NNN(1) = 1                                                          RANP0370
```

```
      NNN(2) = 2                                                        RANP0380
      CNP(1) = ONE                                                      RANP0390
      CNP(2) =-ONE                                                      RANP0400
      CNM(1) =-ONE                                                      RANP0410
      CNM(2) = ONE                                                      RANP0420
  150 CONTINUE                                                          RANP0430
C     BUILD R(STAR)                                                     RANP0440
      DO 300 I=1,2                                                      RANP0450
C     SET THE INDEX POSITION FOR THE 'I'TH POINT                        RANP0460
      NN  = NNN(I)                                                      RANP0470
C     SET THE COMMON FACTORS AND RADIUS VALUES FOR THE 'I'TH POINT      RANP0480
      R1I = ONE/R(NONE)                                                 RANP0490
      R2I = R1I*R1I                                                     RANP0500
      CM  = ONE-FM(NN)                                                  RANP0510
      CMS = ONE-FM(NN)*FM(NN)                                           RANP0520
C     CALCULATE R-STAR FOR ALL EIGHT UNKNOWNS                           RANP0530
      RSTR(1) = RSTR(1) - HALF*(((TWO*COSI*R1I)+ONM(I)*(TWO*DS(NONZ)) ) RANP0540
     1 * X(5,NN,IJH) - HM * (R1I*GNT(IJH,NN))                           RANP0550
     2 - (HM*SINE)*(R2I*GMT(NN)) + XBR(NN) )                            RANP0560
      RSTR(2) = RSTR(2) - HALF*(((COSI*R1I)+ONM(I)*(TWO*DS(NONZ)))      RANP0570
     1 * X(6,NN,IJH) - COSI*GNT(IJH,NN)*R1I + HM*X(5,NN,IJH)*R1I        RANP0580
     2 - TWO*(HM*SINE)*GMPT(NN)*R2I + YBR(NN) )                         RANP0590
      RSTR(3) = RSTR(3) - HALF*(((COSI*R1I)+ONM(I)*(TWO*DS(NONZ)))      RANP0600
     1 * X(7,NN,IJH) - SINE*GNT(IJH,NN)*R1I + TWO*HM*COSI*GMPT(NN)*R2I  RANP0610
     2 - HM*HM* GMT(NN)*R2I +  ZBR(IJH,NN) )                            RANP0620
      RSTR(4) = RSTR(4) - HALF*(((COSI*R1I)+ONM(I)*(TWO*DS(NONZ)))      RANP0630
     1 * X(8,NN,IJH) - COSI*GMT(NN)*R1I + TWO*HM*GMPT(NN)*R1I           RANP0640
     2 - X(7,NN,IJH) )                                                  RANP0650
      RSTR(5) = RSTR(5) - HALF*(-(((COSI*R1I)+ ONP(I)*(TWO*DS(NONZ)))   RANP0660
     1 * X(1,NN,IJH) - HM*X(2,NN,IJH)*R1I - TWO*X(5,NN,IJH) /(D(NN)*CM) RANP0670
     2 +(THREE*SINE*GMPT(NN)*R1I  / (D(NN)*CM ) ) )                     RANP0680
      RSTR(6) = RSTR(6) - HALF*((TWO*DS(NONZ))*ONM(I)*X(2,NN,IJH)       RANP0690
     1 - (X(6,NN,IJH)-FM(NN)*GNT(IJH,NN))/(D(NN)*CMS) )                 RANP0700
      RSTR(7) = RSTR(7) - HALF*((TWO*DS(NONZ))*ONM(I)*X(3,NN,IJH)       RANP0710
     1 + X(4,NN,IJH) )                                                  RANP0720
      RSTR(8) = RSTR(8) -HALF*((TWO*DS(NONZ))*ONM(I)*X(4,NN,IJH)        RANP0730
     1 -(X(8,NN,IJH)-FM(NN)*GMT(NN))/(BK(NN)*CMS) )                     RANP0740
C     CALCULATE THE REMAINING TERMS IN THE NEW RIGHT HAND VECTOR        RANP0750
```

```
      TPS(1) = TPS(1) + HALF*((SINE*R1I)*FLAM(1,NN) +                   RANP0760
     1 (COSI*R1I)*FLAM(2,NN) - FLAM(8,NN) )                             RANP0770
      TPS(2) = TPS(2) + HALF*((HM*R1I)*FLAM(2,NN) - FLAM(7,NN) )        RANP0780
      TPS(3) = TPS(3) + HALF*((HM*R1I)*FLAM(1,NN) - FLAM(9,NN) )        RANP0790
      TPS(4) = TPS(4) + HALF*FLAM(5,NN)                                 RANP0800
      TPS(5) = TPS(5) - HALF*(FLAM(2,NN)/(D(NN)*OM) + FLAM(4,NN) )      RANP0810
      TPS(6) = TPS(6) - FORT*FLAM(6,NN)                                 RANP0820
  300 CONTINUE                                                          RANP0830
C     CALCULATE NEW VALUES OF RHV                                       RANP0840
      DO 350 I=1,8                                                      RANP0850
      IND = IND+1                                                       RANP0860
      RHV(IND) = SNGL( RSTR(I)+TPS(I) )                                 RANP0870
  350 CONTINUE                                                          RANP0880
      RETURN                                                            RANP0890
      END                                                               RANP0900
```

SUBROUTINE MON5

COMMON VARIABLES

	COMMON BLOCK		COMAA		ORIGIN	00001	LENGTH	00001
SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ICHECK	00000	I						
	COMMON BLOCK		COMBB		ORIGIN	00002	LENGTH	00002
CCBETA	00000	R		IIMITE	00001	I		
	COMMON BLOCK		COMCC		ORIGIN	00004	LENGTH	00001
CONFAC	00000	R						
	COMMON BLOCK		COM51		ORIGIN	00005	LENGTH	00005
IJK	00000	I		IJH	00001	I	IJ	00002
MS	00003	I		MF	00004	I		
	COMMON BLOCK		COM52		ORIGIN	00012	LENGTH	00010
SINE	00000	D		COSI	00002	D	HM	00004
M	00006	I		NONE	00007	I		
	COMMON BLOCK		COM59		ORIGIN	00022	LENGTH	00002
IND	00000	I		INDK	00001	I		
	COMMON BLOCK		//		ORIGIN	00024	LENGTH	33167
NSGMTS	00000	I		KS	00001	I	ISGMT	00002
IFTNO	00003	I		LOADNO	00004	I	LDNOPR	00005
NLOAD	00006	I		KKKKK	00007	I	PHI	00010
STNP	00017	R		COSP	00026	R	NPSEG	00035
LSP	00044	I		MLIST	00053	I	ECR	00062
ECY	00071	R		IOPT	00100	I	IRES	00116
NREC	00130	I		INTF	00155	I	NSGPL	00167
NPTRT	00201	I		NLDPL	00213	I	EI	00225
HI	00243	R		FMUI	00261	R	RI	00277
FLDI	00306	R		FLDF	00673	R	DLDI	01260

DLDF	01645	R	FMPO	02232	R	FNRO	02374
FNYO	02536	R	FNPTO	02700	R	ECRS	03042
ECYS	03123	R	RSS	03204	R	SCF	03265
IBC	04705	I	BC	04775	R	DLBC	05065
S	05155	R	FNTH	05631	R	FNPH	06305
DPHI	06761	R	OTHE	07435	R	STP	10111
AK	10235	R	AU	16461	R	AL	24705
RHV	31755	R	SAVE	33105	R	INTF1	00155
INTF	00155	I	INTF2	00156	I	INTF3	00157
INTF4	00160	I	INTF5	00161	I	INTF6	00162
INTF7	00163	I	AK	10235	R	RHK	10235
PS	14775	R	AL	24705	R	RHM	24705
SGOP	25215	R	SGOM	26232	R	SEPS	27247
FDC	30264	R	MIND	30625	I	ISFLAG	31354
IBETA	31402	I	GLAM	31446	D	BETA	31712

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
WORD	33213	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
MITE	33225	I	TNO	33226	R	MAXHRM	33227
NROSEG	33230	I	NCCS	33231	I	NZOI	33232
NPSH	33233	I	NPTS	33234	I	NPT	33235
I	33236	I	IKK	33237	I	J	33240
IK	33241	I	NPS	33242	I	JRH	33243
NROS	33244	I	IJHTT	33245	I	IF6	33246
JJ	33247	I	ITI	33250	I	LOAD	33251

ENTRY POINTS

MDN5 SECTION 16

SUBROUTINES CALLED

.FRWT.	SECTION	17	.FWRB.	SECTION	18	GETM	SECTION
.FRDB.	SECTION	20	TCCR	SECTION	21	.FWRD.	SECTION
CNRH	SECTION	23	.UN04.	SECTION	24	.UN14.	SECTION
.FWLR.	SECTION	26	.FBLT.	SECTION	27	.FBDT.	SECTION
.FRLR.	SECTION	29	.UN06.	SECTION	30	.FFIL.	SECTION
.FCNV.	SECTION	32	.UN09.	SECTION	33	CC.1	SECTION
CC.2	SECTION	35	CC.3	SECTION	36	CC.4	SECTION
SYSLOC	SECTION	38					

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
2000	FORMAT	33277	2005	FORMAT	33324	2020	FORMAT
2040	FORMAT	33340	2050	FORMAT	33350	80	41A
50	19A	33433	60	25A	33441	70	38A
90	42A	33474	160	80A	33614	100	50A
110	52A	33514	120	54A	33517	130	60A
150	77A	33606	200	84A	33626	250	93A
255	120A	33750	252	109A	33726	253	114A
254	118A	33745	303	180A	34174	258	123A
270	151A	34072	260	125A	33757	263	135A
265	145A	34061	365	250A	34425	275	159A
280	162A	34117	300	174A	34160	301	177A
304	188A	34215	305	190A	34221	400	303A
395	298A	34617	306	198A	34235	450	312A
307	219A	34332	320	234A	34360	310	224A
315	230A	34355	340	242A	34400	330	241A
350	246A	34406	360	248A	34412	380	291A
372	265A	34471	375	277A	34564	385	295A
410	305A	34637	460	319A	34674	500	326A
480	321A	34703					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 35006.

TCCRX STORAGE MAP

SUBROUTINE TCCR

COMMON VARIABLES

COMMON BLOCK COM51 ORIGIN 00001 LENGTH 00005

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
UNKMAX	00006	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IPS	00016	I	IRH	00017	I	JRH	00020
IBB	00021	I	FACTOR	00022	R	I	00023
NPTS	00024	I	NVLS	00025	I	PSS	00026
IK	00027	I	PST	00030	R	RHH	00031
BT	00032	R	IJT	00033	I	BR	00034
BETAU	00035	R	IJU	00036	I		

ENTRY POINTS

TCCR SECTION 5

SUBROUTINES CALLED

SYSLOC SECTION 6

TCCRX STORAGE MAP

EEN IEN CORRESPONDENCE

EEN	IEN	LOCATION	EEN	IEN	LOCATION	EEN	IEN
51	17A	00110	50	14A	00106	52	21A
56	41A	00200	54	34A	00146	58	47A
200	93A	00377	150	92A	00377	59	69A
60	78A	00324	100	90A	00375	70	84A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00454.

SUBROUTINE CNRH

COMMON VARIABLES

COMMON BLOCK	COMAA		ORIGIN	00001		LENGTH	00001
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ICHECK	00000	I					
COMMON BLOCK	COM51		ORIGIN	00002		LENGTH	00005
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		
COMMON BLOCK	COM52		ORIGIN	00007		LENGTH	00010
SINE	00000	D	COSI	00002	D	HM	00004
M	00006	I	NONE	00007	I		
COMMON BLOCK	COM53		ORIGIN	00017		LENGTH	00034
D	00000	D	BK	00004	D	R	00010
FM	00014	D	DS	00020	D	XBR	00024
YBR	00030	D					
COMMON BLOCK	COM54		ORIGIN	00053		LENGTH	00334
RSTR	00000	D	ZBR	00020	D	GOP	00054
GOM	00110	D	GNT	00144	D	GNPT	00200
FLAM	00234	D	EPS	00300	D		
COMMON BLOCK	COM55		ORIGIN	00407		LENGTH	00010
GMT	00000	D	GMPT	00004	D		
COMMON BLOCK	COM56		ORIGIN	00417		LENGTH	00340
X	00000	D					
COMMON BLOCK	COM58		ORIGIN	00757		LENGTH	00070
A	00000	D	B	00016	D	C	00034
DZ	00052	D					

CNRHX STORAGE MAP

COMMON BLOCK			COM59	ORIGIN	01047	LENGTH	00002
IND	00000	I	INDK	00001	I		
COMMON BLOCK			//	ORIGIN	01051	LENGTH	33167
NSGMTS	00000	I	KS	00001	I	ISGMT	00002
IFTNO	00003	I	LOADNO	00004	I	LDNOPR	00005
NLOAD	00006	I	KKKKK	00007	I	PHI	00010
SINP	00017	R	COSP	00026	R	NPSEG	00035
LSP	00044	I	MLIST	00053	I	ECR	00062
ECY	00071	R	IOPT	00100	I	IRES	00116
NREC	00130	I	INTF	00155	I	NSGPL	00167
NPTPT	00201	I	NLDPL	00213	I	EI	00225
HI	00243	R	FMUI	00261	R	RI	00277
FLDI	00306	R	FLDF	00673	R	DLDI	01260
DLDF	01645	R	FMPO	02232	R	FNRO	02374
FNYO	02536	R	FNPTO	02700	R	ECRS	03042
ECYS	03123	R	RSS	03204	R	SCF	03265
IBC	04705	I	BC	04775	R	DLBC	05065
S	05155	R	FNTH	05631	R	FNPH	06305
DPHI	06761	R	CTHE	07435	R	STP	10111
AK	10235	R	AU	16461	R	AL	24705
RHV	31755	R	SAVE	33105	R	AK	10235
RHK	10235	R	PS	14775	R	AL	24705
RHM	24705	R	SGOP	25215	R	SGOM	26232
SEPS	27247	R	FDC	30264	R	MIND	30625
ISFLAG	31354	I	IBETA	31402	I	GLAM	31446
BETA	31712	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
MFMS2M	34240	I	MFMS2P	34241	I	MAXSEG	34242
MAXHRM	34243	I	NPTSEG	34244	I	MAXPTS	34245
NROSEG	34246	I	INTF2	34247	I	LOPE	34250
III	34251	I	INTH	34252	I	INOP	34253
NPTS	34254	I	I	34255	I	TT	34256
JJ	34257	I	IBCF	34260	I	NFLAG	34261

CNRHX STORAGE MAP

IJJ	34262	I	IKZ	34263	I

ENTRY POINTS

CNRH	SECTION	22

SUBROUTINES CALLED

OMEG	SECTION	23	SDAT	SECTION	24	PVAL	SECTION
DVEC	SECTION	26	BVEC	SECTION	27	RANP	SECTION
.FWRD.	SECTION	29	.UN06.	SECTION	30	.FFIL.	SECTION
.FCNV.	SECTION	32	SYSLOC	SECTION	33		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
30	14A	34523	50	36A	34614	40	18A	
45	33A	34512	43	28A	34575	90	49A	
85	42A	34653	80	55A	34712	70	51A	
140	66A	34731	120	64A	34726	180	79A	
170	75A	34746	200	83A	35021	160	74A	
300	177A	35447	8000	169A	35427	7000	102A	
8001	FORMAT	34314	8002	FORMAT	34331	8003	FORMAT	
8004	FORMAT	34353	8005	FORMAT	34373	8006	FORMAT	
8007	FORMAT	34410	8009	FORMAT	34413	8010	FORMAT	
8011	FORMAT	34427	8012	FORMAT	34446	7825	123A	
7850	134A	35263	7900	147A	35326	7999	168A	
220	171A	35433	230	173A	35436	240	174A	

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 35473.

SUBROUTINE OMEG

COMMON VARIABLES

COMMON BLOCK COM51 ORIGIN 00001 LENGTH 00005

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		

COMMON BLOCK COM52 ORIGIN 00006 LENGTH 00010

SINE	00000	D	COSI	00002	D	HM	00004
M	00006	I	NONE	00007	I		

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
Y	00016	D					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ONE	00024	D	HALF	00026	D	R1	00030
FD1	00032	D	FD2	00034	D	FD3	00036
XDOT	00040	D	YDOT	00042	D	R11	00044
HN	00046	D	DEL	00050	R	NPTS	00051
IHI	00052	I	J	00053	I	NM8	00054
NZ	00055	I	NP8	00056	I	MM8	00057
MZ	00060	I	MP8	00061	I	I	00062
KK	00063	I					

ENTRY POINTS

OMEG SECTION 7

SUBROUTINES CALLED

FDIC	SECTION	8	E.1	SECTION	9	E.2	SECTION	
E.3	SECTION	11	E.4	SECTION	12	CC.1	SECTION	
CC.2	SECTION	14	CC.3	SECTION	15	CC.4	SECTION	
SYSLOC	SECTION	17						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
300	56A	00563	200	54A	00560	30	33A	
50	37A	00306						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00741.

SUBROUTINE SDAT

COMMON VARIABLES

COMMON BLOCK	COM51		ORIGIN	00001		LENGTH	00005
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		
COMMON BLOCK	COM52		ORIGIN	00006		LENGTH	00010
SINE	00000	D	COSI	00002	D	HM	00004
M	00006	I	NONE	00007	I		
COMMON BLOCK	COM53		ORIGIN	00016		LENGTH	00034
D	00000	D	BK	00004	D	R	00010
FM	00014	D	DS	00020	D	XBR	00024
YBR	00030	D					
COMMON BLOCK	COM54		ORIGIN	00052		LENGTH	00334
RSTR	00000	D	ZBR	00020	D	GOP	00054
GDM	00110	D	GNT	00144	D	GNPT	00200
FLAM	00234	D	EPS	00300	D		
COMMON BLOCK	COM56		ORIGIN	00406		LENGTH	00340
X	00000	D					

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
SLPX	00746	D	SLPZ	00752	D		

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION

SDATX

STORAGE MAP

012	00770	D	SI	00772	D	S2	00774
SLPE	00776	D	SLPH	01000	D	SLPU	01002
E1	01004	D	H1	01006	D	U1	01010
ET	01012	D	HT	01014	D	UT	01016
X1	01020	D	Y1	01022	D	R1	01024
IZ	01026	I	JZ	01027	I	DEL	01030
IP	01031	I	JP	01032	I	L	01033
TX	01034	I					

ENTRY POINTS

SDAT SECTION 13

SUBROUTINES CALLED

CC.1	SECTION	14	CC.2	SECTION	15	CC.3	SECTION	
CC.4	SECTION	17	SYSLOC	SECTION	18			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
100	40A	01311	50	20A	01206	70	29A	
125	51A	01407	150	62A	01447	200	82A	
160	74A	01537						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 02061.

SUBROUTINE PVAL

COMMON VARIABLES

COMMON BLOCK COM51 ORIGIN 00001 LENGTH 00005

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		

COMMON BLOCK COM52 ORIGIN 00006 LENGTH 00010

SINE	00000	D	COSI	00002	D	HM	00004
M	00006	I	NONE	00007	I		

COMMON BLOCK COM53 ORIGIN 00016 LENGTH 00034

D	00000	D	BK	00004	D	R	00010
FM	00014	D	DS	00020	D	XBR	00024
YBR	00030	D					

COMMON BLOCK COM54 ORIGIN 00052 LENGTH 00334

RSTR	00000	D	ZBR	00020	D	GOP	00054
GOM	00110	D	GNT	00144	D	GNPT	00200
FLAM	00234	D	EPS	00300	D		

COMMON BLOCK COM55 ORIGIN 00406 LENGTH 00010

GMT	00000	D	GMPT	00004	D		

COMMON BLOCK COM56 ORIGIN 00416 LENGTH 00340

X	00000	D					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
HALF	00756	D	TONE	00760	D	ONE	00762
OHLF	00764	D	TWO	00766	D	TP25	00770
THREE	00772	D	R1I	00774	D	R2I	00776

PVALX

STORAGE MAP

DM	01000	D	DMS	01002	D	ZLI	01004
CD	01006	D	KS	01010	I	MAXHRM	01011
MTEM	01012	I	IJHTEM	01013	I	LFLAG	01014
T	01015	I	J	01016	I		

ENTRY POINTS

PVAL	SECTION	15

SUBROUTINES CALLED

LAMB	SECTION	16	E.1	SECTION	17	E.2	SECTION	
E.3	SECTION	19	E.4	SECTION	20	CC.1	SECTION	
CC.2	SECTION	22	CC.3	SECTION	23	CC.4	SECTION	
SYSLOC	SECTION	25						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
40	30A	01315	60	47A	01452	100	50A	
200	59A	01531						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01762.

SUBROUTINE LAMB

COMMON VARIABLES

COMMON BLOCK			COM52	ORIGIN	00001	LENGTH	00010	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	
SINE	00000	D	COSI	00002	D	HM	00004	
M	00006	I	NONE	00007	I			
COMMON BLOCK			COM54	ORIGIN	00011	LENGTH	00334	
RSTR	00000	D	ZBR	00020	D	GOP	00054	
GOM	00110	D	GNT	00144	D	GNPT	00200	
FLAM	00234	D	EPS	00300	D			
COMMON BLOCK			COM55	ORIGIN	00345	LENGTH	00010	
GMT	00000	D	GMPT	00004	D			
COMMON BLOCK			COM56	ORIGIN	00355	LENGTH	00340	
X	00000	D						
COMMON BLOCK			COM58	ORIGIN	00715	LENGTH	00070	
A	00000	D	B	00016	D	C	00034	
D	00052	D						

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
VALU	01005	D	IABCD	01007	I	IJH	01010
IFLAG	01011	I					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
MAXHRM	01012	I	KS	01013	I	JJ	01014

LAMBX STORAGE MAP

ILGZ	01015	I	JI	01016	I

ENTRY POINTS

LAMB SECTION 13

SUBROUTINES CALLED

SUMM SECTION 14 SUMMU SECTION 15 SYSLOC SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
100	40A	01201	50	8A	01075	175	97A
70	17A	01125	95	36A	01176	80	33A
90	34A	01163	600	286A	02232	174	93A
130	62A	01263	110	54A	01247	150	74A
140	67A	01277	170	90A	01365	160	83A
173	91A	01367	190	119A	01464	185	107A
200	137A	01533	195	125A	01506	210	139A
220	145A	01562	230	156A	01602	250	174A
245	171A	01651	240	166A	01630	260	176A
270	192A	01717	265	183A	01703	275	193A
280	195A	01726	285	211A	01761	282	202A
295	230A	02046	293	226A	02034	290	221A
300	232A	02053	320	254A	02130	310	246A
325	255A	02132	350	257A	02137	400	258A
410	268A	02172	430	280A	02214	450	284A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 02326.

SUBROUTINE SUMM

COMMON VARIABLES

COMMON BLOCK CGM58 ORIGIN 00001 LENGTH 00070

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
A	00000	D	B	00016	D	C	00034
D	00052	D					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
FLAG	00071	D	VALUT	00073	D	VALU	00075
IFLAG	00077	I	IJH	00100	I	IMPL	00101
IMML	00102	I	JI	00103	I	JM	00104
ILGZ	00105	I	KS	00106	I		

ENTRY POINTS

SUMM SECTION 5 SUMMU SECTION 6

SUBROUTINES CALLED

.FXEM. SECTION 7 SYSLOC SECTION 8

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
200	11A	00177	400	58A	00373	600	115A
800	154A	01012	250	20A	00225	300	38A
290	35A	00301	350	56A	00372	340	53A
1000	188A	01113	450	69A	00427	420	64A

430	67A	00420	500	91A	00535	490	88A
550	113A	00646	540	110A	00640	650	134A
640	131A	00722	700	152A	01011	690	149A
950	187A	01113	900	174A	01045	850	171A
940	184A	01106					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01211.

SUBROUTINE BVEC

COMMON VARIABLES

	COMMON BLOCK		COM51		ORIGIN	00001	LENGTH	00005
SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I		IJH	00001	I	IJ	00002
MS	00003	I		MF	00004	I		
	COMMON BLOCK		COM52		ORIGIN	00006	LENGTH	00010
SINE	00000	D		COSI	00002	D	HM	00004
M	00006	I		NONE	00007	I		
	COMMON BLOCK		COM53		ORIGIN	00016	LENGTH	00034
D	00000	D		BK	00004	D	R	00010
FM	00014	D		DS	00020	D	XBR	00024
YBR	00030	D						
	COMMON BLOCK		COM56		ORIGIN	00052	LENGTH	00340
X	00000	D						
	COMMON BLOCK		COM59		ORIGIN	00412	LENGTH	00002
INC	00000	I		INDK	00001	I		

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
PI	00414	D	BC1	00416	D	FLAG	00420
TEMP	00422	D	NONZ	00424	I	IBCF	00425
MFG	00426	I					

ENTRY POINTS

BVEC SECTION 13

STORAGE MAP

SUBROUTINES CALLED

.FXEM.	SECTION	14	CC.1	SECTION	15	CC.2	SECTION
CC.3	SECTION	17	CC.4	SECTION	18	SYSLOC	SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
100	49A	00765	30	15A	00623	55	20A
65	27A	00671	40	14A	00620	75	36A
85	44A	00744	70	31A	00706	80	40A
50	18A	00644	60	24A	00664	90	47A
200	63A	01011	110	53A	00773	150	60A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01103.

SUBROUTINE DVEC

COMMON VARIABLES

COMMON BLOCK COM51 ORIGIN 00001 LENGTH 00005

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		

COMMON BLOCK COM52 ORIGIN 00006 LENGTH 00010

SIND	00000	D	COSD	00002	D	HM	00004
M	00006	I	NONE	00007	I		

COMMON BLOCK COM53 ORIGIN 00016 LENGTH 00034

D	00000	D	BK	00004	D	R	00010
FM	00014	D	DS	00020	D	XBR	00024
YBR	00030	D					

COMMON BLOCK COM56 ORIGIN 00052 LENGTH 00340

X	00000	D					

COMMON BLOCK COM57 ORIGIN 00412 LENGTH 00004

ERS	00000	R	EYS	00001	R	R1	00002
RBB	00003	R					

COMMON BLOCK COM58 ORIGIN 00416 LENGTH 00070

A	00000	D	B	00016	D	C	00034
DZ	00052	D	X1	00034	D		

COMMON BLOCK COM59 ORIGIN 00506 LENGTH 00002

IND	00000	I	INDK	00001	I		

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION

DVECX STORAGE MAP

AF	00510	R	TEMP	00530	D

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
RA	00550	D	RB	00552	D	RIN	00554
S2	00556	D	C2	00560	D	SPSI	00562
CPSI	00564	D	ER	00566	D	EY	00570
HNPTO	00572	D	HNRO	00574	D	HNYO	00576
HMPO	00600	D	R1I	00602	D	VALU	00604
SINE	00606	D	COSI	00610	D	IJKP	00612
NONZ	00613	I	I	00614	I	IO	00615
ILOZ	00616	I	JI	00617	I	IFLAG	00620
IABCD	00621	I					

ENTRY POINTS

DVEC	SECTION	17

SUBROUTINES CALLED

KCOG	SECTION	18	SUMM	SECTION	19	SUMMU	SECTION	
CC.1	SECTION	21	CC.2	SECTION	22	CC.3	SECTION	
CC.4	SECTION	24	SYSLOC	SECTION	25			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
50	11A	00717	100	15A	00726	400	65A
120	26A	00772	160	43A	01102	200	56A
600	124A	01645	450	71A	01373	475	91A
460	85A	01427	500	97A	01477	525	104A
550	110A	01616	575	119A	01633		

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 02110.

SUBROUTINE KCOG

COMMON VARIABLES

COMMON BLOCK COM52 ORIGIN 00001 LENGTH 00010

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
SINT	00000	D	COST	00002	D	HM	00004
M	00006	I	NONE	00007	I		

COMMON BLOCK COM57 ORIGIN 00011 LENGTH 00004

ERS	00000	R	EYS	00001	R	R1	00002
RBB	00003	R					

COMMON BLOCK COM58 ORIGIN 00015 LENGTH 00070

AZ	00000	D	BZ	00016	D	CZ	00034
DZ	00052	D	A	00000	R	FM	00020
BZ	00016	D	SINE	00022	R	COSI	00024
R11	00026	R	BZ	00016	D		

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
C	00105	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
I	00125	I	J	00126	I		

ENTRY POINTS

KCOG SECTION 9

SUBROUTINES CALLED

E.1	SECTION	10	E.2	SECTION	11	E.3	SECTION
E.4	SECTION	13	CC.1	SECTION	14	CC.2	SECTION
CC.3	SECTION	16	CC.4	SECTION	17	SYSLOC	SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
10	7A	00162	20	20A	00301	40	43A
30	37A	00354					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00430.

SUBROUTINE RANP

COMMON VARIABLES

	COMMON BLOCK		COM51	ORIGIN	00001		LENGTH	00005
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION
IJK	00000	I	IJH	00001	I		IJ	00002
MS	00003	I	MF	00004	I			
	COMMON BLOCK		COM52	ORIGIN	00006		LENGTH	00010
SINE	00000	D	COSI	00002	D		HM	00004
M	00006	I	NONE	00007	I			
	COMMON BLOCK		COM53	ORIGIN	00016		LENGTH	00034
D	00000	D	BK	00004	D		R	00010
FM	00014	D	DS	00020	D		XBR	00024
YBR	00030	D						
	COMMON BLOCK		COM54	ORIGIN	00052		LENGTH	00334
RSTR	00000	D	ZBR	00020	D		GOP	00054
GOM	00110	D	GNT	00144	D		GNPT	00200
FLAM	00234	D	EPS	00300	D			
	COMMON BLOCK		COM55	ORIGIN	00406		LENGTH	00010
GMT	00000	D	GMPT	00004	D			
	COMMON BLOCK		COM56	ORIGIN	00416		LENGTH	00340
X	00000	D						
	COMMON BLOCK		COM59	ORIGIN	00756		LENGTH	00002
INC	00000	I	INDK	00001	I			

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION

RANPX STORAGE MAP

ONP	00760	D	CNM	00764	D	TPS	00770
NNN	01010	I					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
HALF	01012	D	ONE	01014	D	TWO	01016
THREE	01020	D	R1I	01022	D	R2I	01024
OM	01026	D	CMS	01030	D	R1	01032
FORT	01034	D	NONZ	01036	I	NN	01037

ENTRY POINTS

RANP SECTION 17

SUBROUTINES CALLED

SYSLOC SECTION 18

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
50	6A	01116	75	11A	01127	100	14A
150	16A	01160	300	89A	02010	350	99A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 02060.

```
$IBFTC MON6X   NOLIST,DECK,M94,XR7,NODD
```

```
C     MONITOR REGION 6 --- PRINT AND PLOT RESULTS                       MON60010
      SUBROUTINE MON6                                                   MON60020
      DIMENSION RHK(2400),PS(600,7),ENT(75,7),EMPT(75,7),EOM(75,7),     MON60030
     1 EMT(75,7),FDC(3,75),MIND(7,5,7),ISFLAG(3,7),PB(75,5),SP(91),     MON60040
     2 TP(91),JOPT(9)                                                   MON60050
      EQUIVALENCE (RHK(1),AK(1,1)), (PS(1,1),AK(21,18)),                MON60060
     1  (AL(  1, 2), EOM( 1,1)), (AL(126,4),ENT(01,1)),                 MON60070
     2  (AL( 51, 7),EMPT( 1,1)), (AL(176,9),EMT( 1,1)),                 MON60080
     3  (AL(  1,13),SP( 1)),     (AL(92,13),ISFLAG(1,1))                MON60090
      EQUIVALENCE  (JOPT(1), IOPT6),                                    MON60100
     1  (RHV(  1), PB( 1, 1)),            (RHV(  1), MIND( 1, 1, 1)),   MON60110
     2  (RHV(246), FDC( 1, 1)),           (RHV(471), TP( 1))            MON60120
      COMMON/COMAA/ICHECK                                               MON60125
      COMMON/COM61/IJK,IJH,IJ,MS,MF                                     MON60130
      COMMON/COM62/R,D,CK,FM,SD,ARC,HT                                  MON60140
      COMMON/COM64/X(8,7)                                               MON60150
      COMMON/COM65/A(7),B(7)                                            MON60160
      COMMON/COM66/GLAM(3,7)                                            MON60170
      COMMON/COM67/SINE,COSI,HM,M                                       MON60180
      COMMON/COM69/KST,NSOC,PTEM(20)                                    MON60190
      COMMON/COM6A/STHE(7),CTHE(7)                                      MON60200
      COMMON/COM6B/IOPT6,IOPT7,IOPT8,IOPT9,IOPT10,IOPT11,IOPT12,        MON60210
     1 IOPT13,IOPT14                                                    MON60220
      COMMON/COM6C/NFLAG,MFLAG,MMFLAG,TPLTIM                            MON60230
      COMMON/COM6D/THET                                                 MON60240
      COMMON/COM6E/NPLOT                                                MON60250
      COMMON/COM6F/SIGMAX,SIGTHE,ISIGPT                                 MON60260
      COMMON NSGMTS, KS, ISGMT, IFTNO, LOADNO, LDNOPR, NLOAD, KKKKK     MON60270
      COMMON PHI(7),SINP(7),COSP(7),NPSEG(7),LSP(7),MLIST(7),ECR(7),    MON60280
     1 ECY(7),IOPT(14),IRES(10),NREC(21),INTF(10),NSGPL(10),NPTPT(10),  MON60290
     2 NLDPL(10)                                                        MON60300
      COMMON EI(2,7),HI(2,7),FMUI(2,7),RI(7)                            MON60310
      COMMON FLDI(5,7,7),FLDF(5,7,7),DLDI(5,7,7),DLDF(5,7,7)            MON60320
      COMMON FMPO(2,7,7),FNRO(2,7,7),FNYO(2,7,7),FNPTO(2,7,7),ECRS(7,7),MON60330
     1 ECYS(7,7),RSS(7,7),SCF(16,7,7)                                   MON60340
      COMMON IBC(4,7,2),BC(4,7,2),DLBC(4,7,2)                           MON60350
      COMMON S(300),FNTH(300),FNPH(300),DPHI(300),DTHE(300),STP(12,7)   MON60360
```

```
      COMMON AK(140,23),AU(140,23),AL(200,13),RHV(600),SAVE(5,10)       MON60370
 2020 FORMAT(1H0,16HMAXIMUM STRESS =, 1PE13.5, 5X,8HTHETA = ,E13.5,     MON60380
     1 5X,6HPOINT ,I3)                                                  MON60390
 2030 FORMAT(13H0 LOAD NUMBER,I3/5X,17HNUMBER OF PLOTS =,I4,5X,         MON60400
     1 15HPLOTTING TIME =,F6.2,8H MINUTES)                              MON60410
 2040 FORMAT(1H1,5X,8HLOAD NO.,11X,8HDELTA(Y) //)                       MON60420
 2050 FORMAT(8X,I3,9X,1PE15.6)                                          MON60430
      RADI = 1. / 57.29578                                              MON60440
C     INITIALIZE PLOTTING TIME FOR THIS LOADING                         MON60450
      TPLTIM = 0.0                                                      MON60460
C     INITIALIZE VALUES                                                 MON60470
      NPLOT = 0                                                         MON60480
      MAXHRM =   7                                                      MON60490
      NPTSEG =  75                                                      MON60500
      NROSEG = 600                                                      MON60510
      TMAX   = 180.0                                                    MON60520
C     MOVE FLAGS TO NAMED COMMON BLOCK                                  MON60530
      II = 0                                                            MON60540
      DO 50 I = 6,14                                                    MON60550
      II = II + 1                                                       MON60560
      JOPT(II) = IOPT(I)                                                MON60570
   50 CONTINUE                                                          MON60580
C     INCREMENT LOAD NUMBER                                             MON60590
      LOADNO = LOADNO + 1                                               MON60600
      IF(ICHECK . EQ. 0) GO TO 100                                      MON60602
      IOPT6 =  3                                                        MON60603
      IOPT9 =180                                                        MON60605
      IOPT10=  1                                                        MON60606
      IOPT11=  0                                                        MON60607
      IOPT12=  0                                                        MON60608
  100 CONTINUE                                                          MON60610
C     SET INDEX OF NEXT LOADING TO BE PRINTED                           MON60620
      LDNOPR = LDNOPR + IOPT10                                          MON60630
      IF(LDNOPR.GT.NLOAD) LDNOPR = NLOAD                                MON60640
C     INITIALIZE SEGMENT LOOP FLAGS                                     MON60650
      IJK = 0                                                           MON60660
C     START SEGMENT LOOP                                                MON60670
  300 CONTINUE                                                          MON60680
```

```
C     OBTAIN SUMMATION INDICES                                          MON60690
      CALL GETM(MLIST,KS,MIND,ISFLAG,MAXHRM)                            MON60700
C     INCREMENT SEGMENT INDICES                                         MON60710    29
      IJK = IJK + 1                                                     MON60720
      MS = LSP(IJK)                                                     MON60730
      ME = MS + NPSEG(IJK) - 1                                          MON60740
      SINE = SINP(IJK)                                                  MON60750
      COSI = COSP(IJK)                                                  MON60760
C     INITIALIZE THETA VALUE                                            MON60770
      THET = 0.0                                                        MON60780
C     CALCULATE NUMBER OF ROWS IN THIS SEGMENT                          MON60790
      NROS = 8*NPSEG(IJK)                                               MON60800
C     INITIALIZE MAXIMUM STRESS FOR SEGMENT                             MON60810
      SIGMAX = 0.0                                                      MON60820
C     READ PRIOR SOLUTION FOR THIS SEGMENT                              MON60830
      READ(C4) ((PS(I,J),I=1,NROS),J=1,KS)                              MON60840    40
      IF(IJK.EQ.NSGMTS) REWIND 4                                        MON60850
C     CALCULATE N(THETA), OMEGA, M(PHI-THETA), M(THETA) FOR ALL M VALUESMON60860    50
      CALL FMRO(ENT,EMPT,EOM,EMT,FDC,S,MLIST,MIND,ISFLAG,EI,HI,FMUI,    MON60870
     1 RI,PS,MAXHRM,KS,NPTSEG,NROSEG)                                   MON60880
C     PRINT THIS LOAD LEVEL--- NO, GO TO 650, YES-CONTINUE              MON60890    52
      IF(LOADNO.NE.LDNOPR) GO TO 650                                    MON60900
C     PRINT VALUES                                                      MON60910
      IF(IOPT6) 650,650 ,330                                            MON60920
  330 CONTINUE                                                          MON60930
C     YES                                                               MON60940
C     CHECK FOR DISPLACEMENT PRINTING                                   MON60950
      IF(IOPT6-2) 350,370,350                                           MON60960
  350 CONTINUE                                                          MON60970
      NSOC = 1                                                          MON60980
      IJH  = 1                                                          MON60990
      KST = 1                                                           MON61000
      M  = MLIST(IJH)                                                   MON61010
      GO TO 390                                                         MON61020
  370 CONTINUE                                                          MON61030
      NSOC = 2                                                          MON61040
      IJH  = KS                                                         MON61050
      KST = KS                                                          MON61060
```

```
      M    = 0                                                    MON61070
      THET = - FLOAT(IOPT9)                                       MON61080
  390 CONTINUE                                                    MON61090
      FM = FLOAT(M)                                               MON61100
      MFLAG = 0                                                   MON61110
C     INITIALIZE SUBROUTINES                                      MON61120
C           POINT VALUES                                          MON61130
      CALL SDAV(ENT,EMPT,EMT,EOM,PS,S,EI,HI,FMUI,RI,MLIST,NPTSEG, MON61140
     1 NROSEG,MFLAG,KS)                                           MON61150
C           HEADING PRINTER                                       MON61160   73
      CALL STRT(MLIST)                                            MON61170
C           DISPLACEMENT PRINTER                                  MON61180   75
      CALL DISP(SAVE,LOADNO,NSGMTS)                               MON61190
C     INITIALIZE POINT COUNTER                                    MON61200   77
      IJ   = MS - 1                                               MON61210
      IPT  = MS                                                   MON61220
C     START POINT LOOP                                            MON61230
  410 CONTINUE                                                    MON61240
C     INCREMENT POINT COUNTER                                     MON61250
      IJ = IJ + 1                                                 MON61260
      IF(IJ - MF)  415,430,430                                    MON61270
  415 IF(IJ - IPT) 410,420,420                                    MON61280
  420 CONTINUE                                                    MON61290
  430 CONTINUE                                                    MON61300
      IPT  = IPT + IOPT8                                          MON61310
C     OBTAIN POINT VALUES                                         MON61320
      CALL SDAVU                                                  MON61330   88
  450 CONTINUE                                                    MON61340
      IF(NSOC-1) 500,500,460                                      MON61350
  460 CONTINUE                                                    MON61360
      IF(THET-TMAX) 480,470,470                                   MON61370
  470 CONTINUE                                                    MON61380
      THET = -FLOAT(IOPT9)                                        MON61390
      GO TO 550                                                   MON61400
  480 CONTINUE                                                    MON61410
      THET = THET + FLOAT(IOPT9)                                  MON61420
      IF(THET.GT.TMAX) THET = TMAX                                MON61430
C     CALCULATE SINE(M*THETA) AND COS(M*THETA)                    MON61440
```

```
      DO 490 I=1,KS                                                     MON61450
      HNN = FLOAT(MLIST(I)) * THET * RADI                               MON61460
      STHE(I)  = SIN(HNN)                                               MON61470    102
      CTHE(I)  = COS(HNN)                                               MON61480    104
  490 CONTINUE                                                          MON61490
  500 CONTINUE                                                          MON61500
C     PRINT HEADING                                                     MON61510
      CALL STRTU                                                        MON61520
C     ARE DISPLACEMENTS TO BE PRINTED                                   MON61530    110
      IF(ICPT7-2) 510,520,510                                           MON61540
  510 CONTINUE                                                          MON61550
C     YES                                                               MON61560
      CALL DISPU                                                        MON61570    114
  520 CONTINUE                                                          MON61580
C     PRINT FORCES                                                      MON61590
      CALL FORC(MLIST)                                                  MON61600    116
  540 CONTINUE                                                          MON61610
C     IF THIS IS A COMBINED HARMONIC LOOP, GO BACK TO USE ANOTHER THETA MON61620
      IF(NSOC-1) 550,550,460                                            MON61630
  550 CONTINUE                                                          MON61640
C     IF NOT LAST POINT GO TO BEGINNING OF POINT LOOP                   MON61650
      IF(IJ - MF) 410,570,570                                           MON61660
  570 CONTINUE                                                          MON61670
      IF(NSOC-1) 580,580,625                                            MON61680
  580 CONTINUE                                                          MON61690
C     ALL HARMONICS DONE                                                MON61700
      IF(IJH - KS) 590,600,600                                          MON61710
  590 CONTINUE                                                          MON61720
C     INCREMENT HARMONIC INDEX                                          MON61730
      IJH = IJH + 1                                                     MON61740
      M = MLIST(IJH)                                                    MON61750
      GO TO 390                                                         MON61760
  600 CONTINUE                                                          MON61770
C     PRINT COMBINED HARMONIC RESULTS                                   MON61780
      IF(ICPT6 - 2) 650,370,370                                         MON61790
  625 CONTINUE                                                          MON61800
      WRITE(6,2020) SIGMAX, SIGTHE, ISIGPT                              MON61810    131
  650 CONTINUE                                                          MON61820
```

```
C     IF THIS LOADING IS NOT TO BE PLOTTED, GO TO 900                MON61830
      IF( NLDPL(LOADNO).EQ.0) GO TO 900                              MON61840
C     SET FLAG FOR NUMBER OF HARMONICS                               MON61850
      KST = KS                                                       MON61860
C     DO PLOTTING                                                    MON61870
C     SET NFLAG                                                      MON61880
      NFLAG = 0                                                      MON61890
      NFLG  = 0                                                      MON61900
      IF(IOPT11) 680,680,670                                         MON61910
  670 CONTINUE                                                       MON61920
      NFLG  = 1                                                      MON61930
  680 CONTINUE                                                       MON61940
      IF(IOPT12) 700,700,690                                         MON61950
  690 CONTINUE                                                       MON61960
      NFLAG = 2                                                      MON61970
  700 CONTINUE                                                       MON61980
      NFLAG = NFLAG + NFLG                                           MON61990
      IF(NFLAG) 900,900,720                                          MON62000
  720 CONTINUE                                                       MON62010
C     IS THIS SEGMENT TO BE PLOTTED                                  MON62020
      IF(NSGPL(IJK)-1) 900,740,730                                   MON62030
  730 CONTINUE                                                       MON62040
C     SET NFLAG FOR INDIVIDUAL POINT PLOTTING ONLY                   MON62050
      NFLAG = 2                                                      MON62060
  740 CONTINUE                                                       MON62070
C     PLOT N(THETA)                                                  MON62080
      MMFLAG = 1                                                     MON62090
      MFLAG = 2                                                      MON62100
      CALL SUFP(ENT,PB,S,LSP,NPTPT,SP,TP,MLIST,NPTSEG)               MON62110
C     PLOT M(THETA)                                                  MON62120   155
      MMFLAG = 2                                                     MON62130
      MFLAG = 2                                                      MON62140
      CALL SUFP(EMT,PB,S,LSP,NPTPT,SP,TP,MLIST,NPTSEG)               MON62150
C     MOVE NBAR(PHI-THETA),NN(PHI), AND M(PHI) TO PLOT BUFFERS       MON62160   159
      NPTS = NPSEG(IJK)                                              MON62170
      DO 800 IK = 1,KS                                               MON62180
      II =-3                                                         MON62190
      DO 800 I=1,NPTS                                                MON62200
```

```
      II = II + 8
      ENT(I,IK) = PS(II  ,IK)                                   MON62210
      EOM(I,IK) = PS(II+1,IK)                                   MON62220
      EMT(I,IK) = PS(II+3,IK)                                   MON62230
  800 CONTINUE                                                  MON62240
C     PLOT NBAR(PHI-THETA)                                      MON62250
      MMFLAG = 3                                                MON62260
      MFLAG = 1                                                 MON62270
      CALL SUFP(ENT,PB,S,LSP,NPTPT,SP,TP,MLIST,NPTSEG)          MON62280
C     PLOT N(PHI)                                               MON62290
      MFLAG = 2                                                 MON62300   182
      MMFLAG = 4                                                MON62310
      CALL SUFP(EOM,PB,S,LSP,NPTPT,SP,TP,MLIST,NPTSEG)          MON62320
C     PLOT M(PHI)                                               MON62330
      MMFLAG = 5                                                MON62340   186
      MFLAG = 2                                                 MON62350
      CALL SUFP(EMT,PB,S,LSP,NPTPT,SP,TP,MLIST,NPTSEG)          MON62360
  900 CONTINUE                                                  MON62370   190
C     ARE ALL SEGMENTS                                          MON62380
      IF(NSGMTS - IJK) 930,930,300                              MON62390
  930 CONTINUE                                                  MON62400
      WRITE(6,2030) LOADNO,NPLOT,TPLTIM                         MON62410
C     SET OVERLAY FLAGS FOR REGION 4                            MON62420
      KKKKK = 4                                                 MON62430   193
C     IF ALL LOADINGS DONE, SET OVERLAY FLAG TO TERMINATE CASE  MON62440
      IF(LOADNO - NLOAD) 970,950,950                            MON62450
  950 CONTINUE                                                  MON62460
C     SET OVERLAY FLAG TO TERMINATE CASE                        MON62470
      KKKKK = 8                                                 MON62480
C     PRINT SAVED VALUES                                        MON62490
      WRITE(6,2040)                                             MON62500
      DO 960 I=1,LOADNO                                         MON62510   197
      WRITE(6,2050) I,(SAVE(J,I),J=1, 1)                        MON62520
  960 CONTINUE                                                  MON62530   200
C     TERMINATE PLOT TAPE                                       MON62540
      CALL SUFPU                                                MON62550
  970 CONTINUE                                                  MON62560   208
      RETURN                                                    MON62570
                                                                MON62580
```

```
END                                                    MON62590
```

```
$IBFTC FMROX   NOLIST,DECK,M94,XR7,NODO
```

```
C     CALCULATE OMEGA,   N(THETA), M(THETA), M(PHI-THETA)                FMR00010
      SUBROUTINE FMRO(ENT,EMPT,EOM,EMT,FDC,S,MLIST,MIND,ISFLAG,EI,       FMR00020
     1 HI,FMUI,RI,PS,MAXH,KS,NPSG,NRSG)                                  FMR00030
      DIMENSION ENT(NPSG,1),EMPT(NPSG,1),ECM(NPSG,1),EMT(NPSG,1),        FMR00040
     1 FDC(3,1),S(1),MLIST(1),MIND(MAXH,5,1),ISFLAG(3,1),EI(2,1),        FMR00050
     2 HI(2,1),FMUI(2,1),RI(1),PS(NRSG,1)                                FMR00060
      COMMON/COM61/IJK,IJH,IJ,MS,MF                                      FMR00070
      COMMON/COM62/R,D,CK,FM,SD,ARC,HT                                   FMR00080
      COMMON/COM63/HNT(7),HMPT(7),HOP(7),HOM(7),HMT(7)                   FMR00090
      COMMON/COM64/X(8,7)                                                FMR00100
      COMMON/COM66/GLAM(3,7)                                             FMR00110
      COMMON/COM67/SINE,COSI,HM,M                                        FMR00120
C     SET CONSTANT VALUES                                                FMR00130
      HALF  = 0.5                                                        FMR00140
C     CALCULATE FINITE DIFFERENCE COEFFICIENTS FOR THIS SEGMENT          FMR00150
      CALL FDIC(FDC,S,MS,MF)                                             FMR00160
C     INITIALIZE SUBROUTINE                                              FMR00170    2
      MFLAG = 1                                                          FMR00180
      CALL SDAV(ENT,EMPT,EMT,EOM,PS,S,EI,HI,FMUI,RI,MLIST,NPSG,NRSG,     FMR00190
     1 MFLAG,KS)                                                         FMR00200
C     CALCULATE THE NUMBER OF POINTS IN THIS SEGMENT                     FMR00210    5
      NPTS = MF - MS + 1                                                 FMR00220
C     START LOOP OVER POINTS                                             FMR00230
      IJ = MS - 1                                                        FMR00240
C     INITIALIZE INDICIES TO GET U(I)                                    FMR00250
      NM8 =  1                                                           FMR00260
      NZ  =  9                                                           FMR00270
      NP8 = 17                                                           FMR00280
      DO 400 I = 1,NPTS                                                  FMR00290
      IF(I.LT.3) GO TO 50                                                FMR00300
      IF(I.EQ.NPTS) GO TO 50                                             FMR00310
C     SET INDICES FOR U(N-1), U(N), AND U(N+1)                           FMR00320
      NM8 = NZ                                                           FMR00330
      NZ  = NP8                                                          FMR00340
      NP8 = NP8 + 8                                                      FMR00350
   50 CONTINUE                                                           FMR00360
      IJ = IJ + 1                                                        FMR00370
```

```
C     CALCULATE SEGMENT PROPERTIES FOR THIS POINT                       FMR00380
      CALL SDAVU                                                        FMR00390    25
      R1I =  1. / R                                                     FMR00400
      R2I =  1. / (R*R)                                                 FMR00410
      CM  =  1. - FM                                                    FMR00420
      CMS =  1. - FM * FM                                               FMR00430
      ZLI = 1./(1. + 2.25 * CK * SINE * SINE * R2I /D )                 FMR00440
C     START LOOP OVER HARMONICS                                         FMR00450
      DO 200 IHI = 1,KS                                                 FMR00460
C     GET HARMONIC VALUE                                                FMR00470
      HN = FLOAT(MLIST(IHI))                                            FMR00480
C     CALCULATE UDOT                                                    FMR00490
      UDOT  =  PS(NM8,IHI)*FDC(1,I) + PS(NZ,IHI)*FDC(2,I)               FMR00500
     1      +  PS(NP8,IHI)*FDC(3,I)                                     FMR00510
C     CALCULATE INDEX OF PRIOR SOLUTION                                 FMR00520
      KK  = 8 * (I-1)                                                   FMR00530
C     GET UNKNOWNS FOR THIS POINT, THIS HARMONIC                        FMR00540
      DO 75 IX = 1,8                                                    FMR00550
      KK = KK+1                                                         FMR00560
   75 X(IX,IHI) = PS(KK,IHI)                                            FMR00570
C     CALCULATE OMEGA(PHI) FOR THIS POINT AND HARMONIC                  FMR00580
      HOP(IHI) = R1I*(HN*X(3,IHI) + SINE*X(1,IHI))                      FMR00590
C     CALCULATE OMEGA      FOR THIS POINT AND HARMONIC                  FMR00600
      ECM(I,IHI)  =  HALF * (UDOT + R1I * (COSI*X(1,IHI)+HN*X(2,IHI)))  FMR00610
C     SET NAMED COMMON BLOCKS                                           FMR00620
      HCM(IHI) = ECM(I,IHI)                                             FMR00630
  200 CONTINUE                                                          FMR00640
C     CALCULATE LAMBDA(3) AND LAMBDA(4) FOR THIS POINT, ALL HARMONICS   FMR00650
      LFLAG = 1                                                         FMR00660
      CALL LANB(MLIST,MIND,ISFLAG,MAXH,LFLAG,KS)                        FMR00670
C     CALCULATE N(THETA)                                                FMR00680    54
      DO 250 IHI = 1,KS                                                 FMR00690
C     GET HARMONIC VALUE                                                FMR00700
      HN = FLOAT(MLIST(IHI))                                            FMR00710
      ENT(I,IHI) = FM * X(6,IHI) + D*CMS * ( R1I *                      FMR00720
     1 ( SINE*X(3,IHI) + HN*X(1,IHI) + COSI*X(2,IHI) ) + .5*GLAM(2,IHI))FMR00730
      HNT(IHI) = ENT(I,IHI)                                             FMR00740
  250 CONTINUE                                                          FMR00750
```

```
C     CALCULATE LAMBDA(2) FOR ALL HARMONICS                          FMR00760
      LFLAG = 2                                                      FMR00770
      CALL LANB(MLIST,MIND,ISFLAG,MAXH,LFLAG,KS)                     FMR00780    70
      DO 300 IHI = 1,KS                                              FMR00790
C     GET HARMONIC VALUE                                             FMR00800
      HN = FLOAT(MLIST(IHI))                                         FMR00810
C     CALCULATE M(PHI-THETA) FOR ALL HARMONICS                       FMR00820
      EMPT(I,IHI) = 0.5 * CK * OM * ZLI * R1I * (-(2.*HN) * X(4,IHI) FMR00830
     1 - (2.*HN) * COSI * X(3,IHI) * R1I                             FMR00840
     2 + (2.*HN) * SINE * X(2,IHI) * R1I                             FMR00850
     3 + 3. * SINE * X(5,IHI)/(D * OM)                               FMR00860
     4 - 1.5 * SINE       * ( GLAM(1,IHI)/(D * OM) + GLAM(3,IHI) )  ) FMR00870
C     CALCULATE M(THETA) FOR ALL HARMONICS                           FMR00880
      EMT(I,IHI) = FM*X(6,IHI) +                                     FMR00890
     1 CK*CMS*R1I*(HN*R1I*(HN*X(3,IHI) + SINE*X(1,IHI)) + COSI*X(4,IHI))FMR00900
  300 CONTINUE                                                       FMR00910
  400 CONTINUE                                                       FMR00920
      RETURN                                                         FMR00930
      END                                                            FMR00940
```

```
$IBFTC SCAVX    NOLIST,DECK,M94,XR7,NODD
```

```
C     CALCULATE MISCELLANEOUS VALUES FOR EACH POINT                     SDAV0010
      SUBROUTINE SDAV(ENT,EMPT,EMT,EOM,PS,S,EI,HI,FMUI,RI,MLIST,NPSG,   SDAV0020
     1 NRSG,MFLAG,KS)                                                   SDAV0030
      DIMENSION ENT(NPSG, 1),EMPT(NPSG,1),EMT(NPSG,1),EOM(NPSG,1),      SDAV0040
     1 PS(NRSG,1),S(1),EI(2,1),HI(2,1),FMUI(2,1),RI(1),MLIST(01)        SDAV0050
      COMMON/COM61/IJK,IJH,IJ,MS,MF                                     SDAV0060
      COMMON/COM62/R,D,CK,FM,SD,ARC,HT                                  SDAV0070
      COMMON/COM63/HNT(7),HMPT(7),HOP(7),HOM(7),HMT(7)                  SDAV0080
      COMMON/COM64/X(8,7)                                               SDAV0090
      COMMON/COM67/SINE,COSI,HM,M                                       SDAV0100
      C12 = 8.3333333E-2                                                SDAV0110
      ONE = 1.0                                                         SDAV0120
      RETURN                                                            SDAV0130
      ENTRY SDAVU                                                       SDAV0140
      IZ  = IJ                                                          SDAV0150
      JZ = IJ - MS + 1                                                  SDAV0160
      IF(IZ.NE.MS) GO TO 100                                            SDAV0170
C     CALCULATE SEGMENT LENGTH                                          SDAV0180
      DEL = ABS(S(MF)-S(MS))                                            SDAV0190
C     CALCULATE SLOPES OF 'E', 'H', AND 'MU'                            SDAV0200
      SLPE = (EI(2,IJK)-EI(1,IJK))/DEL                                  SDAV0210
      SLPH = (HI(2,IJK)-HI(1,IJK))/DEL                                  SDAV0220
      SLPU = (FMUI(2,IJK)-FMUI(1,IJK))/DEL                              SDAV0230
C     CALCULATE INITIAL VALUES FOR SEGMENT                              SDAV0240
      R1    = RI(IJK)                                                   SDAV0250
      S1    = S(MS)                                                     SDAV0260
      E1    = EI(1,IJK)                                                 SDAV0270
      H1    = HI(1,IJK)                                                 SDAV0280
      U1    = FMUI(1,IJK)                                               SDAV0290
  100 CONTINUE                                                          SDAV0300
C     CALCULATE VALUES FOR THIS POINT                                   SDAV0310
      SD  = S(IZ)                                                       SDAV0320
      R   = S(IZ)*COSI + R1                                             SDAV0330
      ARC = S(IZ+1) - S(IZ)                                             SDAV0340
      IF(IZ.EQ.MF) ARC = 0.0                                            SDAV0350
C     CALCULATE STIFFNESSES FOR POINT 'IZ'                              SDAV0360
      HT    = SLPH*SD + H1                                              SDAV0370
```

```
      FM    = SLPU*SD + U1                                          SDAV0380
      D     = (SLPE*SD+E1) * HT/(1.0 - FM*FM)                       SDAV0390
      CK    = O12 * D * HT * HT                                     SDAV0400
  150 CONTINUE                                                      SDAV0410
      IF(MFLAG.EQ.1) GO TO 250                                      SDAV0420
      R1I = ONE / R                                                 SDAV0430
C     SET INDICES OF PRIOR SOLUTION                                 SDAV0440
      IP = 8*(JZ-1) + 1                                             SDAV0450
      JP = IP + 7                                                   SDAV0460
      DO 200 L = 1,KS                                               SDAV0470
      HN = MLIST(L)                                                 SDAV0480
      IX = 0                                                        SDAV0490
C     TRANSFER VALUES OF PRIOR SOLUTION FOR THIS POINT              SDAV0500
      DO 160 IN = IP,JP                                             SDAV0510
      IX = IX +1                                                    SDAV0520
  160 X(IX,L) = PS(IN,L)                                            SDAV0530
C     CALCULATE OMEGA(PHI)                                          SDAV0540
      HOP(L) = R1I * (HN*X(3,L) + SINE*X(1,L))                      SDAV0550
C     TRANSFER VALUES OF OMEGA,     M(PHI), N(THETA), AND M(PHI-THETA) SDAV0560
C         FOR THIS HARMONIC                                         SDAV0570
      HNT(L) = ENT(JZ,L)                                            SDAV0580
      HMPT(L)= EMPT(JZ,L)                                           SDAV0590
      HOM(L) = EOM(JZ,L)                                            SDAV0600
      HMT(L) = EMT(JZ,L)                                            SDAV0610
  200 CONTINUE                                                      SDAV0620
  250 CONTINUE                                                      SDAV0630
      RETURN                                                        SDAV0640
      END                                                           SDAV0650
```

```
$IBFTC LANBX   NOLIST,DECK,M94,XR7,NODD
```

```
C     BUILD LAMBDA(I),I= 2, 3, 4 FOR ALL HARMONICS                LANB0010
      SUBROUTINE LANB(MLIST,MIND,ISFLAG,MAXH,LFLAG,KS)            LANB0020
      DIMENSION MLIST( 1),MIND(MAXH,5,MAXH),ISFLAG(3, 1)          LANB0030
      DIMENSION VALU(1),IJHT(1)                                   LANB0040
      COMMON/COM63/HNT(7),HMPT(7),HOP(7),HOM(7),HMT(7)            LANB0050
      COMMON/COM64/X(8,7)                                         LANB0060
      COMMON/COM65/A(7),B(7)                                      LANB0070
      COMMON/COM66/GLAM(3,7)                                      LANB0080
      IF(LFLAG-1) 50,50,200                                       LANB0090
   50 CONTINUE                                                    LANB0100
C     INITIALIZE SUMMATION SUBROUTINE                             LANB0110
      CALL SUNN(MLIST,MIND,ISFLAG,VALU,IJHT,MAXH)                 LANB0120
C     SET INPUT BUFFERS FOR LAMBDA(4)                             LANB0130   4
      DO  30 I=1,KS                                               LANB0140
      A(I) =  X(4,I)                                              LANB0150
      B(I) = HOP(I)                                               LANB0160
   30 CONTINUE                                                    LANB0170
C     CALCULATE LAMBDA(3) AND LAMBDA(4) FOR ALL HARMONICS         LANB0180
      DO 190 JJ =1,KS                                             LANB0190
      IJHT = JJ                                                   LANB0200
      M = MLIST(IJHT)                                             LANB0210
      VALUT = 0.0                                                 LANB0220
      VALU  = 0.0                                                 LANB0230
C     CALCULATE LAMBDA(3)                                         LANB0240
      IF(M.NE.0) GO TO 80                                         LANB0250
C     GET NUMBER OF HARMONICS .GT. 0                              LANB0260
      ILGZ = ISFLAG(1,IJHT)                                       LANB0270
C     IF ILGZ=0, NO TERMS EXIST FOR THIS HARMONIC                 LANB0280
      IF(ILGZ.EQ.0) GO TO 80                                      LANB0290
      DO 75 I=1,ILGZ                                              LANB0300
      JI = MIND(I,5,IJHT)                                         LANB0310
      VALU = VALU + HOP(JI)*HOP(JI) + HOM(JI)*HOM(JI)             LANB0320
   75 CONTINUE                                                    LANB0330
      VALU = 0.5*VALU                                             LANB0340
      GO TO 130                                                   LANB0350
   80 CONTINUE                                                    LANB0360
      IMPL = ISFLAG(2,IJHT)                                       LANB0370
```

```
      M = MLIST(IJHT)                                   LANB0760
      VALU = 0.0                                        LANB0770
      IF(M.EQ.0) GO TO 250                              LANB0780
      CALL SUNNU                                        LANB0790    110
  250 CONTINUE                                          LANB0800
      GLAM(1,IJHT) = VALU                               LANB0810
  300 CONTINUE                                          LANB0820
  400 CONTINUE                                          LANB0830
      RETURN                                            LANB0840
      END                                               LANB0850
```

```
$IBFTC SUNNX   NOLIST,DECK,M94,XR7,NODD
```

```
C     PERFORM 'A' SUMMATION                                          SUNN0010
      SUBROUTINE SUNN(MLIST,MIND,ISFLAG,VALU,IJHT,MAXH)              SUNN0020
      DIMENSION MLIST( 1), MIND(MAXH,5, 1), ISFLAG(3, 1)             SUNN0030
      COMMON/COM65/A(7),B(7)                                         SUNN0040
      RETURN                                                         SUNN0050
      ENTRY SUNNU                                                    SUNN0060
      VALU  = 0.0                                                    SUNN0070
      VALUT = 0.0                                                    SUNN0080
C     SET SUMMATION INDICES                                          SUNN0090
      IMPL = ISFLAG(2,IJHT)                                          SUNN0100
      IMML = ISFLAG(3,IJHT)                                          SUNN0110
C     IF FIRST HARMONIC .NE. 0, CONSTANT TERMS DON'T EXIST           SUNN0120
      IF(MLIST(1).NE.0) GO TO 50                                     SUNN0130
C     ADD CONSTANT TERMS                                             SUNN0140
      VALU = A(1)*B(IJHT)                                            SUNN0150
   50 CONTINUE                                                       SUNN0160
C     IF IMPL=0, NO NON-ZERO TERMS EXIST FOR THIS SUMMATION          SUNN0170
      IF(IMPL.EQ.0) GO TO 100                                        SUNN0180
      DO 80 I=1,IMPL                                                 SUNN0190
      JI = MIND(I,2,IJHT)                                            SUNN0200
      JM = MIND(I,1,IJHT)                                            SUNN0210
      VALUT= VALUT+ A(JI)*B(JM) - A(JM)*B(JI)                        SUNN0220
   80 CONTINUE                                                       SUNN0230
      VALU = VALU + 0.5*VALUT                                        SUNN0240
  100 CONTINUE                                                       SUNN0250
C     IF IMML=0, NO NON-ZERO TERMS EXIST IN THIS SUMMATION           SUNN0260
      IF(IMML.EQ.0) GO TO 150                                        SUNN0270
      VALUT = 0.0                                                    SUNN0280
      DO 130 I=1,IMML                                                SUNN0290
      JI = MIND(I,4,IJHT)                                            SUNN0300
      JM = MIND(I,3,IJHT)                                            SUNN0310
      VALUT = VALUT + A(JI)*B(JM) + A(JM)*B(JI)                      SUNN0320
  130 CONTINUE                                                       SUNN0330
      VALU = VALU + 0.25*VALUT                                       SUNN0340
  150 CONTINUE                                                       SUNN0350
      RETURN                                                         SUNN0360
      END                                                            SUNN0370
```

```
$IBFTC STRTX   NOLIST,DECK,M94,XR7,NODD
```

```
C     PRINT HEADING FOR EACH PAGE AND POINT VALUES                      STRT0010
      SUBROUTINE STRT(MLIST)                                            STRT0020
      DIMENSION MLIST(01)                                               STRT0030
      COMMON/COM61/IJK,IJH,IJ,MS,MF                                     STRT0040
      COMMON/COM62/R,D,CK,FM,SD,ARC,HT                                  STRT0050
      COMMON/COM69/KST,NSOC,PTEM(20)                                    STRT0060
      COMMON/COM68/IOPT6,IOPT7,IOPT8,IOPT9,IOPT10,IOPT11,IOPT12,        STRT0070
     1 IOPT13,IOPT14                                                    STRT0080
      COMMON/COM6D/THET                                                 STRT0090
 1000 FORMAT(1X,5HPOINT,1X,7HSEGMENT,6X,8HHARMONIC,5X,9HTHICKNESS,17X,  STRT0100
     1 6HRADIUS, 7X,1HS,12X,3HARC,10X,5HTHETA)                          STRT0110
 1001 FORMAT( 7X,1HU,12X,1HV,12X,1HW,12X,8HOMEGA(T),5X,8HOMEGA(P), 5X,  STRT0120
     1  5HOMEGA,8X,8HDELTA(R),5X,8HDELTA(Y) )                           STRT0130
 1002 FORMAT( 7X,9HNBAR(P-T),4X,4HN(P), 9X,4HN(T), 9X,4HN(R), 9X,4HN(Y),STRT0140
     1  9X,7HQBAR(P),6X,4HM(P), 9X,4HM(T)/7X,6HM(P-T) )                 STRT0150
 1010 FORMAT(1H0,I3,6X,I2,10X,I3,6X,1PE13.5,13X,4E13.5)                 STRT0160
 2000 FORMAT(1H1)                                                       STRT0170
C2000 FORMAT(1H1///)                                                    STRT0180
 2010 FORMAT(26H SEPERATE HARMONIC RESULTS)                             STRT0190
 2020 FORMAT(26H COMBINED HARMONIC RESULTS)                             STRT0200
 2030 FORMAT(7X,10HSIGMA(P) +,3X,10HSIGMA(P) -,3X,10HSIGMA(T) +,3X,     STRT0210
     1 10HSIGMA(T) -,2X,12HSIGMA(P-T) +,1X,12HSIGMA(P-T) -,1X,          STRT0220
     2 12HSIGMA(EFF) +,1X,12HSIGMA(EFF) -)                              STRT0230
C     SET NUMBER OF POINTS PER PAGE                                     STRT0240
C     IN CDRC  40 LINES PER PAGE                                        STRT0250
C                                                                       STRT0260
C     PRINT DISPLACEMENTS AND ROTATIONS-- 5 POINTS.... 6 LINES PER POINTSTRT0270
C     PRINT FORCES ONLY                -- 7 POINTS.....5 LINES PER POINTSTRT0280
C     PRINT DISPLACEMENTS ONLY         --12 POINTS.....3 LINES PER POINTSTRT0290
C                                                                       STRT0300
C     AT LEWIS RESEARCH CENTER, 60 LINES PER PAGE, USE 56               STRT0310
C     PRINT DISPLACEMENTS AND ROTATIONS-- 8 POINTS.... 6 LINES PER POINTSTRT0320
C     PRINT FORCES ONLY                --10 POINTS.... 5 LINES PER POINTSTRT0330
C     PRINT DISPLACEMENTS ONLY         --17 POINTS.... 3 LINES PER POINTSTRT0340
      NPDAR  =  5                                                       STRT0350
      NPFO   =  7                                                       STRT0360
      NPDO   = 12                                                       STRT0370
```

```
C       NPDAR  =  8                                          STRT0380
C       NPFO   = 10                                          STRT0390
C       NPDO   = 17                                          STRT0400
C       OBTAIN NUMBER OF POINTS PER PAGE                     STRT0410
        IF(IOPT7-2) 40,60,80                                 STRT0420
   40   CONTINUE                                             STRT0430
C       PRINT DISPLACEMENTS ONLY                             STRT0440
        NPPAGE = NPDO                                        STRT0450
        GO TO 90                                             STRT0460
   60   CONTINUE                                             STRT0470
C       PRINT FORCES ONLY                                    STRT0480
        NPPAGE = NPFO                                        STRT0490
        GO TO 90                                             STRT0500
   80   CONTINUE                                             STRT0510
C       PRINT DISPLACEMENTS AND ROTATIONS                    STRT0520
        NPPAGE = NPDAR                                       STRT0530
   90   CONTINUE                                             STRT0540
C       SET POINT COUNTER                                    STRT0550
        NPT    = NPPAGE                                      STRT0560
        RETURN                                               STRT0570
        ENTRY STRTU                                          STRT0580
C       INCREMENT POINT COUNTER                              STRT0590
        NPT = NPT + 1                                        STRT0600
        IF(NPT-NPPAGE) 200,200,140                           STRT0610
  140   CONTINUE                                             STRT0620
C       EJECT PAGE                                           STRT0630
        WRITE(6,2000)                                        STRT0640   16
        IF(NSOC - 1) 145,145,150                             STRT0650
  145   CONTINUE                                             STRT0660
        WRITE(6,2010)                                        STRT0670   18
        GO TO 155                                            STRT0680
  150   CONTINUE                                             STRT0690
        WRITE(6,2020)                                        STRT0700   20
  155   CONTINUE                                             STRT0710
C       INITIALIZE POINT COUNTER                             STRT0720
        NPT = 1                                              STRT0730
C       PRINT POINT VALUE HEADING                            STRT0740
        WRITE(6,1000)                                        STRT0750   22
```

```
C     CALCULATE AND PRINT THE DISPLACEMENTS FOR THIS POINT          DISP0010
      SUBROUTINE DISP(SAVE,LOADNO,NSGMTS)                           DISP0020
      DIMENSION SAVE(5,10)                                          DISP0030
      COMMON/COM61/IJK,IJH,IJ,MS,MF                                 DISP0040
      COMMON/COM63/HNT(7),HMPT(7),HOP(7),HOM(7),HMT(7)              DISP0050
      COMMON/COM64/X(8,7)                                           DISP0060
      COMMON/COM65/A(7),B(7)                                        DISP0070
      COMMON/COM66/GLAM(3,7)                                        DISP0080
      COMMON/COM67/SINE,COSI,HM,M                                   DISP0090
      COMMON/COM68/DELR(7),DELY(7),ANR(7),ANY(7),AMP(7)             DISP0100
      COMMON/COM69/KST,NSOC,PTEM(20)                                DISP0110
      COMMON/COM6A/STHE(7),CTHE(7)                                  DISP0120
      COMMON/COM6D/THET                                             DISP0125
      RETURN                                                        DISP0130
      ENTRY DISPU                                                   DISP0140
C     CALCULATE DELTA(R) AND DELTA(Y) FOR THIS HARMONIC             DISP0150
C     SET HARMONIC INDICES                                          DISP0160
      IIJ  = IJH                                                    DISP0170
      IJJ  = KST                                                    DISP0180
   50 CONTINUE                                                      DISP0190
      DELR(IIJ) = X(2,IIJ) * COSI + X(3,IIJ) * SINE                 DISP0200
      DELY(IIJ) = X(2,IIJ) * SINE - X(3,IIJ) * COSI                 DISP0210
C     DECREMENT HARMONIC INDEX                                      DISP0220
      IJJ  =  IJJ - 1                                               DISP0230
      IIJ = IIJ - 1                                                 DISP0240
      IF(IJJ) 100,100,50                                            DISP0250
  100 CONTINUE                                                      DISP0260
      IF(NSOC-1) 200,200,120                                        DISP0270
  120 CONTINUE                                                      DISP0280
C     PERFORM SUMMATIONS FOR CURRENT VALUE OF THETA                 DISP0290
C     ZERO OUT PRINTING ARRAY                                       DISP0300
      DO 130 I=1,10                                                 DISP0310
  130 PTEM(I) = 0.0                                                 DISP0320
      DO 150 IJJ = 1,KST                                            DISP0330
      PTEM(1) = PTEM(1) + X(1,IJJ) * STHE(IJJ)                      DISP0340
      PTEM(2) = PTEM(2) + X(2,IJJ) * CTHE(IJJ)                      DISP0350
      PTEM(3) = PTEM(3) + X(3,IJJ) * CTHE(IJJ)                      DISP0360
```

```
      PTEM(4) = PTEM(4) + X(4,IJJ) * CTHE(IJJ)                  DISP0370
      PTEM(5) = PTEM(5) + HOP(IJJ) * STHE(IJJ)                  DISP0380
      PTEM(6) = PTEM(6) + HOM(IJJ) * STHE(IJJ)                  DISP0390
      PTEM(7) = PTEM(7) + DELR(IJJ)* CTHE(IJJ)                  DISP0400
      PTEM(8) = PTEM(8) + DELY(IJJ)* CTHE(IJJ)                  DISP0410
  150 CONTINUE                                                  DISP0420
C     IF LAST  SEGMENT, SAVE DELTA(Y) FOR THIS LOAD LEVEL       DISP0430
      IF(NSGMTS - IJK) 170,170,250                              DISP0440
  170 CONTINUE                                                  DISP0450
      IF(IJ.NE.MF) GO TO 250                                    DISP0453
      IF(THET.NE.0.0) GO TO 250                                 DISP0456
      SAVE(1,LOADNO) = PTEM(8)                                  DISP0460
      GO TO 250                                                 DISP0470
  200 CONTINUE                                                  DISP0480
C     SET PRINTING ARRAY EQUAL TO PROPER VALUES                 DISP0490
      DO 225 I=1,4                                              DISP0500
  225 PTEM(I) = X(I,IJH)                                        DISP0510
      PTEM(5) = HOP(IJH)                                        DISP0520
      PTEM(6) = HOM(IJH)                                        DISP0530
      PTEM(7) = DELR(IJH)                                       DISP0540
      PTEM(8) = DELY(IJH)                                       DISP0550
  250 CONTINUE                                                  DISP0560
C     PRINT DISPLACEMENT AND ROTATION VALUES                    DISP0570
      WRITE(6,1000) (PTEM(I),I=1,8)                             DISP0580    69
 1000 FORMAT(5X,1P8E13.5)                                       DISP0590
      RETURN                                                    DISP0600
      END                                                       DISP0610
```

```
$IBFTC FORCX   NOLIST,DECK,M94,XR7,NODD
```

```
C     CALCULATE AND PRINT FORCES FOR THIS POINT                          FORC0010
      SUBROUTINE FORC(MLIST)                                             FORC0020
      DIMENSION MLIST(1)                                                 FORC0030
      COMMON/COM61/IJK,IJH,IJ,MS,MF                                      FORC0040
      COMMON/COM62/R,D,CK,FM,SD,ARC,HT                                   FORC0050
      COMMON/COM63/HNT(7),HMPT(7),HOP(7),HOM(7),HMT(7)                   FORC0060
      COMMON/COM64/X(8,7)                                                FORC0070
      COMMON/COM67/SINE,COSI,HM,M                                        FORC0080
      COMMON/COM68/DELR(7),DELY(7),ANR(7),ANY(7),AMP(7)                  FORC0090
      COMMON/COM69/KST,NSOC,PTEM(20)                                     FORC0100
      COMMON/COM6A/STHE(7),CTHE(7)                                       FORC0110
      COMMON/COM6B/IOPT6,IOPT7,IOPT8,IOPT9,IOPT10,IOPT11,IOPT12,         FORC0120
     1 IOPT13,IOPT14                                                     FORC0130
      COMMON/COM6D/THET                                                  FORC0140
      COMMON/COM6F/SIGMAX,SIGTHE,ISIGPT                                  FORC0150
C     CALCULATE N(R) AND N(Y)                                            FORC0160
C     SET HARMONIC INDEX                                                 FORC0170
 2000 FORMAT(5X,1P8E13.5)                                                FORC0180
      IJJ = IJH                                                          FORC0190
      IIJ = KST                                                          FORC0200
   50 CONTINUE                                                           FORC0210
      HN  = FLOAT(MLIST(IJJ))                                            FORC0220
      ANR(IJJ) = X(6,IJJ)*COSI + X(7,IJJ)*SINE                           FORC0230
      ANY(IJJ) = X(6,IJJ)*SINE - X(7,IJJ)*COSI                           FORC0240
C     DECREMENT HARMONIC COUNTER AND INDEX                               FORC0250
      IIJ = IIJ - 1                                                      FORC0260
      IJJ = IJJ - 1                                                      FORC0270
      IF(IIJ) 100,100,50                                                 FORC0280
  100 CONTINUE                                                           FORC0290
  120 CONTINUE                                                           FORC0300
      IF(NSOC-1) 250,250,150                                             FORC0310
  150 CONTINUE                                                           FORC0320
C     ZERO OUT PRINTING ARRAY                                            FORC0330
      DO 130 I=1,20                                                      FORC0340
  130 PTEM(I) = 0.0                                                      FORC0350
C     CALCULATE COMBINED FORCES FOR THIS POINT AND THIS THETA            FORC0360
      DO 200 IJJ = 1,KST                                                 FORC0370
```

```
      PTEM(1) = PTEM(1) + X(5,IJJ) * STHE(IJJ)                          FORC0380
      PTEM(2) = PTEM(2) + X(6,IJJ) * CTHE(IJJ)                          FORC0390
      PTEM(3) = PTEM(3) + HNT(IJJ) * CTHE(IJJ)                          FORC0400
      PTEM(4) = PTEM(4) + ANR(IJJ) * CTHE(IJJ)                          FORC0410
      PTEM(5) = PTEM(5) + ANY(IJJ) * CTHE(IJJ)                          FORC0420
      PTEM(6) = PTEM(6) + X(7,IJJ) * CTHE(IJJ)                          FORC0430
      PTEM(7) = PTEM(7) + X(8,IJJ) * CTHE(IJJ)                          FORC0440
      PTEM(8) = PTEM(8) + HMT(IJJ) * CTHE(IJJ)                          FORC0450
      PTEM(9) = PTEM(9) + HMPT(IJJ)* STHE(IJJ)                          FORC0460
  200 CONTINUE                                                          FORC0470
C     CALCULATE STRESSES                                                FORC0480
C     SET THICKNESS CONSTANTS                                           FORC0490
      HT1  = 1. /  HT                                                   FORC0500
      HT2  = 6. / (HT*HT)                                               FORC0510
C     CALCULATE SIGMA(PHI)                                              FORC0515
      PTEM(10) = PTEM(2)*HT1 + PTEM(7)*HT2                              FORC0520
      PTEM(11) = PTEM(2)*HT1 - PTEM(7)*HT2                              FORC0530
C     CALCULATE SIGMA(THETA)                                            FORC0535
      PTEM(12) = PTEM(3)*HT1 + PTEM(8)*HT2                              FORC0540
      PTEM(13) = PTEM(3)*HT1 - PTEM(8)*HT2                              FORC0550
C     CALCULATE SIGMA(PHI-THETA)                                        FORC0555
      PTEM(14) = PTEM(1)*HT1 + PTEM(9)*HT2                              FORC0560
      PTEM(15) = PTEM(1)*HT1 - PTEM(9)*HT2                              FORC0570
C     CALCULATE SIGMA(EFFECTIVE)                                        FORC0575
      PTEM(16) = SQRT(PTEM(10)**2 + PTEM(12)**2 - PTEM(10)*PTEM(12)     FORC0580
     1 + 3.*PTEM(14)**2)                                                FORC0590   49
      PTEM(17) = SQRT(PTEM(11)**2 + PTEM(13)**2 - PTEM(11)*PTEM(13)     FORC0600
     1 + 3.*PTEM(15)**2)                                                FORC0610
C     TEST FOR MAXIMUM STRESS                                           FORC0620   50
      PT = AMAX1(PTEM(16),PTEM(17))                                     FORC0630
      IF(SIGMAX - PT) 230,230,300                                       FORC0640
  230 CONTINUE                                                          FORC0650
      SIGMAX = PT                                                       FORC0660
      SIGTHE = THET                                                     FORC0670
      ISIGPT = IJ                                                       FORC0680
      GO TO 300                                                         FORC0690
  250 CONTINUE                                                          FORC0700
C     SET PRINTING ARRAY EQUAL TO FORCES FOR THIS POINT, THIS HARMONIC  FORC0710
```

```
      PTEM(1) = X(5,IJH)                                        FORC0720
      PTEM(2) = X(6,IJH)                                        FORC0730
      PTEM(3) = HNT(IJH)                                        FORC0740
      PTEM(4) = ANR(IJH)                                        FORC0750
      PTEM(5) = ANY(IJH)                                        FORC0760
      PTEM(6) = X(7,IJH)                                        FORC0770
      PTEM(7) = X(8,IJH)                                        FORC0780
      PTEM(8) = HMT(IJH)                                        FORC0790
      PTEM(9) = HMPT(IJH)                                       FORC0800
  300 CONTINUE                                                  FORC0810
C     TEST TO SEE WHETHER OR NOT FORCES ARE TO BE PRINTED       FORC0820
      IF(IOPT7-1) 400,400,320                                   FORC0830
  320 CONTINUE                                                  FORC0840
      WRITE(6,2000) (PTEM(I),I=1,9)                             FORC0850   66
      IF(NSOC-1) 400,400,350                                    FORC0860
  350 CONTINUE                                                  FORC0870
C     PRINT STRESSES                                            FORC0880
      WRITE(6,2000) (PTEM(I),I=10,17)                           FORC0890   74
  400 CONTINUE                                                  FORC0900
      RETURN                                                    FORC0910
      END                                                       FORC0920
```

```
$IBFTC SUFPX   NOLIST,DECK,M94,XR7,NODD
```

```
C     CALCULATE SUMMATIONS FOR FORCE  AND PLOT THEM                     SUFP0010
      SUBROUTINE SUFP(XVL,PB,S,LSP,NPTPT,SP,TP,MLIST,NPSG)              SUFP0020
      DIMENSION XVL(NPSG,1),PB(NPSG,1),S(01),LSP(01),NPTPT(01),SP(01),  SUFP0030
     1 TP(01),MLIST(01)                                                 SUFP0040
      DIMENSION INPPS(10),SCT(7,2)                                      SUFP0050
      DIMENSION BTITLE(12),YTITLE(12),TTITLE(12),ATITLE(5,5),TITLE(5,20)SUFP0060
      COMMON/COM61/IJK,IJH,IJ,MS,MF                                     SUFP0070
      COMMON/COM69/KST,NSOC,PTEM(20)                                    SUFP0080
      COMMON/COM6A/STHE(7),CTHE(7)                                      SUFP0090
      COMMON/COM6B/IOPT6,IOPT7,IOPT8,IOPT9,IOPT10,IOPT11,IOPT12,        SUFP0100
     1 IOPT13,IOPT14                                                    SUFP0110
      COMMON/COM6C/NFLAG,MFLAG,MMFLAG,TPLTIM                            SUFP0120
      COMMON/COM6D/THET                                                 SUFP0130
      COMMON/COM6E/NPLOT                                                SUFP0140
      REAL NUMB                                                         SUFP0150
      EQUIVALENCE(STHE(1),SCT(1,1))                                     SUFP0160
      DATA TITLE(1, 1)/30HCIRCUMFERENTIAL FORCE          /,             SUFP0170
     1     TITLE(1, 2)/30HCIRCUMFERENTIAL BENDING MOMENT/,              SUFP0180
     2     TITLE(1, 3)/30HIN PLANE SHEAR FORCE          /,              SUFP0190
     3     TITLE(1, 4)/30HMERIDIONAL FORCE              /,              SUFP0200
     4     TITLE(1, 5)/30HMERIDIONAL BENDING MOMENT     /,              SUFP0210
     5     TITLE(1, 6)/30H    SEGMENT        POINT     /,              SUFP0220
     6     TITLE(1, 7)/30H      THETA - DEGREES         /,              SUFP0230
     7     TITLE(1, 8)/30H      FORCE                   /,              SUFP0240
     8     TITLE(1, 9)/30H          S                   /,              SUFP0250
     9     TITLE(1,10)/30H                              /,              SUFP0260
     A     TITLE(1,11)/30H                              /,              SUFP0270
     B     TITLE(1,12)/30H                              /,              SUFP0280
     C     TITLE(1,13)/30H                              /,              SUFP0290
     D     TITLE(1,14)/30H                              /,              SUFP0300
     E     TITLE(1,15)/30H                              /,              SUFP0310
     F     TITLE(1,16)/30H     THETA =                  /               SUFP0320
      DATA  ATITLE(1,1)/30H    THETA =                   /,             SUFP0330
     1       ATITLE(1,2)/30H   THETA =                   /,             SUFP0340
     2       ATITLE(1,3)/30H   THETA =                   /,             SUFP0350
     3       ATITLE(1,4)/30H   THETA =                   /,             SUFP0360
     4       ATITLE(1,5)/30H   THETA =                   /              SUFP0370
```

```
      DATA BTITLE(1)/72H                                                SUFP0380
     1                   /                                              SUFP0390
      DATA TTITLE(1)/72H                                                SUFP0400
     1                   /                                              SUFP0410
      DATA YTITLE(1)/72H                                                SUFP0420
     1                   /                                              SUFP0430
C     NTE     - PLOT BUFFER INDEX FOR THETA                             SUFP0440
C     NCX     - SYMBOL FREQUENCY ON CURVES                              SUFP0450
C     IP      - PLOT BUFFER INDEX FOR POINT                             SUFP0460
C     PB(75,5) - PLOT BUFFER                                            SUFP0470
C     SP(91)    - ABSCISSA BUFFER FOR PLOTING                           SUFP0480
C     TP(91)    - ORDINATE BUFFER FOR FORCE VS THETA PLOTTING           SUFP0490
C     IPT       - FLAG CONTAINING INDEX OF NEXT POINT TO BE PLOTTED     SUFP0500
C     NFLAG = 1...PLOT FORCE VS. S FOR VALUES OF THETA ONLY.            SUFP0510
C     NFLAG = 2...PLOT FORCE VS. THETA AT SEVERAL POINTS ONLY.          SUFP0520
C     NFLAG = 3...PLOT BOTH 1. AND 2.                                   SUFP0530
      MARG = 3                                                          SUFP0540
      NCX = 3                                                           SUFP0550
      XSCAL =  9.0                                                      SUFP0560
      IONE =  1                                                         SUFP0570
      MTHE = 91                                                         SUFP0580
      TMAX = 180.                                                       SUFP0590
      RADI = 1. / 57.29578                                              SUFP0600
      IF(NFLAG-2)  50,500,50                                            SUFP0610
   50 CONTINUE                                                          SUFP0620
C     SET PLOT TITLE ARRAYS FOR THIS FORCE                              SUFP0630
      DO 70 I=1,5                                                       SUFP0640
      J = I + MARG                                                      SUFP0650
      TTITLE(J) = TITLE(I,MMFLAG)                                       SUFP0660
      YTITLE(J) = TITLE(I,MMFLAG+10)                                    SUFP0670
      BTITLE(J) = TITLE(I,9)                                            SUFP0680
   70 CONTINUE                                                          SUFP0690
C     CALCULATE THE NUMBER OF POINTS IN THIS SEGMENT                    SUFP0700
      NPTS = ME - MS + 1                                                SUFP0710
C     INITIALIZE THETA INDEX                                            SUFP0720
      NTE = 0                                                           SUFP0730
      THET = -FLOAT(IOPT9)                                              SUFP0740
C     START THETA LOOP                                                  SUFP0750
```

```
  100 CONTINUE                                                      SUFP0760
      IF(THET-TMAX) 120,320,320                                     SUFP0770
  120 CONTINUE                                                      SUFP0780
C     INCREMENT THETA                                               SUFP0790
      THET = THET + FLOAT(IOPT9)                                    SUFP0800
      NTE = NTE + 1                                                 SUFP0810
      IF(THET.GT.TMAX) THET = TMAX                                  SUFP0820
C     GET HOLLERITH REPRESENTATION OF THETA                         SUFP0822
      CALL IBCDFR(THET,NUMB)                                        SUFP0824   29
      ATITLE(3,NTE) = NUMB                                          SUFP0826
C     CALCULATE SIN(M*THETA) AND COS(M*THETA) FOR ALL HARMONICS     SUFP0830
      DO 130 I=1,KST                                                SUFP0840
      HN = FLOAT(MLIST(I)) * RADI                                   SUFP0850
      SCT(I,1) = SIN(HN*THET)                                       SUFP0860   36
      SCT(I,2) = COS(HN*THET)                                       SUFP0870   38
  130 CONTINUE                                                      SUFP0880
C     INITIALIZE POINT INDEX                                        SUFP0890
      IPT = MS                                                      SUFP0900
      IJ  = MS-1                                                    SUFP0910
C     INITIALIZE NUMBER OF POINTS PLOTTED                           SUFP0920
      IP = 0                                                        SUFP0930
      DO 300 IZ= 1,NPTS                                             SUFP0940
      IJ  = IJ+1                                                    SUFP0950
      IF(IJ.EQ.ME) GO TO 150                                        SUFP0960
      IF(IPT-IJ) 300,150,300                                        SUFP0970
  150 CONTINUE                                                      SUFP0980
      IPT = IPT + IOPT8                                             SUFP0990
      IP = IP+1                                                     SUFP1000
      SP(IP) = S(IJ)                                                SUFP1010
C     CALCULATE SUMMATION FOR THIS POINT                            SUFP1020
      VAL = 0.0                                                     SUFP1030
      DO 170 I=1,KST                                                SUFP1040
      VAL = VAL + YVL(IJ,I)*SCT(I,MFLAG)                            SUFP1050
  170 CONTINUE                                                      SUFP1060
      PB(IP,NTE) = VAL                                              SUFP1070
  300 CONTINUE                                                      SUFP1080
C     TEST TO SEE IF PLOT ARRAY FULL                                SUFP1090
      IF(NTE-5) 100,320,320                                         SUFP1100
```

```
  320 CONTINUE                                                          SUFP1110
C     YES.  PLOT FORCE VS. S FOR THETA VALUES                           SUFP1120
      LAST = 1                                                          SUFP1150
      CALL PLOT(SP,PB,XSCAL,NTE,IP,BTITLE,YTITLE,TTITLE,ATITLE,NCX,     SUFP1160
     1 LAST,NPSG,PLTIM)                                                 SUFP1170   72
      TPLTIM = TPLTIM + PLTIM                                           SUFP1180
C                                                                       SUFP1190
C     INCREMENT NUMBER OF PLOTS                                         SUFP1200
      NPLOT = NPLOT + 1                                                 SUFP1210
C     INITIALIZE THETA INDEX FOR PLOT BUFFER                            SUFP1220
      NTE = 0                                                           SUFP1225
      IF(THET-TMAX) 100,480,480                                         SUFP1230
  480 CONTINUE                                                          SUFP1240
      IF(NFLAG-2) 900,500,500                                           SUFP1250
  500 CONTINUE                                                          SUFP1260
C     PLOT FORCE VS THETA AT SOME POINT                                 SUFP1270
C     DETERMINE IF ANY OF POINTS TO BE PLOTTED ARE IN THIS SEGMENT      SUFP1280
C     INITIALIZE THE NUMBER OF POINTS IN THIS SEGMENT                   SUFP1290
      NPPS = 0                                                          SUFP1300
      DO 540 I=1,10                                                     SUFP1310
      IF(NPTPT(I)-MS) 540,530,520                                       SUFP1320
  520 CONTINUE                                                          SUFP1330
      IF(NPTPT(I)-ME) 530,530,540                                       SUFP1340
  530 CONTINUE                                                          SUFP1350
C     INCREMENT NUMBER OF POINTS                                        SUFP1360
      NPPS = NPPS + 1                                                   SUFP1370
C     SAVE INDEX OF POINT                                               SUFP1380
      INPPS(I) = NPTPT(I)                                               SUFP1390
  540 CONTINUE                                                          SUFP1400
C     IF NO POINTS ARE IN THIS SEGMENT, RETURN                          SUFP1410
      IF(NPPS) 900,900,550                                              SUFP1420
  550 CONTINUE                                                          SUFP1430
C     SET PLOT TITLE ARRAYS FOR THIS FORCE                              SUFP1440
      DO 570 I=1,5                                                      SUFP1450
      J = I + MARG                                                      SUFP1460
      TTITLE(J) = TITLE(I,6)                                            SUFP1470
      YTITLE(J) = TITLE(I,MMFLAG)                                       SUFP1480
      BTITLE(J) = TITLE(I,7)                                            SUFP1490
```

```
  570 CONTINUE                                                          SUFP1500
      FIJK = FLOAT(IJK)                                                 SUFP1510
      CALL IBCDER(FIJK,NUMB)                                            SUFP1520    111
      TTITLE(MARG + 3) = NUMB                                           SUFP1530
      DO 800 IZ = 1,NPPS                                                SUFP1540
C     SET POINT INDEX OF THIS POINT                                     SUFP1550
      IJ = INPPS(IZ) - LSP(IJK) + 1                                     SUFP1560
C     INITIALIZE THETA VALUE                                            SUFP1570
      THET = -FLOAT(IOPT13)                                             SUFP1580
C     INITIALIZE THETA INDEX                                            SUFP1590
      NTE = 0                                                           SUFP1600
C     START LOOP ON THETA'S                                             SUFP1610
  600 CONTINUE                                                          SUFP1620
      IF(THET-TMAX) 620,650,650                                         SUFP1630
  620 CONTINUE                                                          SUFP1640
C     INCREMENT PLOT BUFFER FOR THETA                                   SUFP1650
      NTE = NTE + 1                                                     SUFP1660
C     INCREMENT THETA VALUE                                             SUFP1670
      THET  = THET + FLOAT(IOPT13)                                      SUFP1680
      IF(THET.GT.TMAX) THET = TMAX                                      SUFP1690
C     CALCULATE SIN(M*THETA), COS(M*THETA) AND SUMMATION                SUFP1700
C     INITIALIZE VALUE                                                  SUFP1710
      VAL = 0.0                                                         SUFP1720
      DO 640 I=1,KST                                                    SUFP1730
      HN = FLOAT(MLIST(I)) * RADI                                       SUFP1740
      SCT(I,1) = SIN(HN*THET)                                           SUFP1750    129
      SCT(I,2) = COS(HN*THET)                                           SUFP1760    131
      VAL = VAL + SCT(I,MFLAG)*XVL(IJ,I)                                SUFP1770
  640 CONTINUE                                                          SUFP1780
C     SAVE VALUES IN PLOT BUFFERS                                       SUFP1790
      SP(NTE) = THET                                                    SUFP1800
      TP(NTE) = VAL                                                     SUFP1810
      GO TO 600                                                         SUFP1820
  650 CONTINUE                                                          SUFP1830
C     PLOT FORCE VS THETA FOR THIS POINT                                SUFP1840
      FIJ = FLOAT(IJ)                                                   SUFP1850
      CALL IBCDER(FIJ,NUMB)                                             SUFP1860    141
      TTITLE(MARG + 5) = NUMB                                           SUFP1870
```

```
      LAST = 1                                                          SUFP1880
      CALL PLOT(SP,TP,XSCAL,IONE,NTE,BTITLE,YTITLE,TTITLE,ATITLE,NCX,   SUFP1890
     1 LAST,MTHE,PLTIM)                                                 SUFP1900    145
      TPLTIM = TPLTIM + PLTIM                                           SUFP1910
C     INCREMENT NUMBER OF PLOTS                                         SUFP1920
      NPLOT = NPLOT + 1                                                 SUFP1930
  800 CONTINUE                                                          SUFP1940
  900 CONTINUE                                                          SUFP1950
      RETURN                                                            SUFP1960
      ENTRY SUFPU                                                       SUFP1970
C     TERMINATE PLOT TAPE                                               SUFP1980
      CALL CCP(6)                                                       SUFP1990    152
      RETURN                                                            SUFP2000
      END                                                               SUFP2010
```

```
$IBFTC IBCDFR  NOLIST,DECK,M94,XR7,NODD
```

```
C     SET A FLOATING POINT INTEGER TO HOLLERITH FORM                    IBC00100
      SUBROUTINE IBCDFR (X,KWD)                                         IBC00200
      DIMENSION IRD (3)                                                 IBC00300
      DATA KX/6H0     /                                                 IBC00330
      KWD = 0                                                           IBC00360
      IRD(1) = AMOD(X,10.0)                                             IBC00500
      NCU = 1                                                           IBC00600
      IRD(2) = AMOD(X,100.0)/10.0                                       IBC00700
      IF (IRD(2) .NE. 0) NCU = 2                                        IBC00800
      IRD(3) = X/100.0                                                  IBC00900
      IF (IRD(3) .NE. 0) NCU = 3                                        IBC01000
      DO 20 I = 1,NCU                                                   IBC01100
      K = NCU + 1 - I                                                   IBC01200
   20 KWD = IRD(K) + KWD * 64                                           IBC01300
      NCU1 = 6 - NCU                                                    IBC01400
      DO 30 I = 1,NCU1                                                  IBC01500
      KWD = KWD * 64 + 48                                               IBC01600
   30 CONTINUE                                                          IBC01700
      IF(ABS(X) .LT. 1.0E-5) KWD = KX                                   IBC01750
      RETURN                                                            IBC01800
      END                                                               IBC01900
```

```
$IBFTC PLOTX   NOLIST,DECK,MS4,XR7,NODD
```

```
      SUBROUTINE PLOT (X,Y,XSCAL,NC,NP,BTITLE,YTITLE,TTITLE,ATITLE,NCX,
     X  LAST, NYOUT, PLTIM)
C     X - INDEPENDENT VARIABLE
C     Y - DEPENDENT VARIABLE
C     XSCAL - LENGTH OF PAPER (INCHES)
C     NC - NO. OF CURVES  (IF NC = 1, NO ANNOTATION AND ATITLE MAY BE 0)
C     NP - NO. OF POINTS
C     TTITLE - TOP TITLE - 12 WORDS LONG
C     BTITLE - BOTTOM TITLE - 12 WORDS LONG
C     YTITLE - LEFT HAND TITLE - 12 WORDS LONG
C     ATITLE - ANOTATION TITLE (TO EXPLAIN SYMBOLS) - MAX. OF 7 TITLES,
C     5 WORDS LONG
C     NCX - SYMBOL FREQUENCY OF CURVES (ISF1)
C     NYOUT - FIRST DIMENSION OF THE Y VARIABLE.
      DIMENSION X(NP), Y(NYOUT,NC), ATITLE(5,NC), YMIN(7), YMAX(7),
     X  DXO(7,2), SYMX(6,7)
      DATA (SYMX (1,I) , I = 1,7) /0606015601360, 0606016601360,06060176
     101360,0606035601360,0606036601360,0606020601360,0606067601360 /
      YSCAL = 10.0
      IF(XSCAL .EQ. 0.0) XSCAL = 14.0
      IF(XSCAL .LT.14.0) YSCAL =  7.0
      NXP = 15
      IF (XSCAL .EQ. 9.0) NXP = 20
      IF (XSCAL .EQ. 14.0) NXP = 25
      XMAX = X(1)
      XMIN = X(1)
      DO 20 I = 1,NC
      YMAX(I) = Y(1,I)
   20 YMIN(I) = Y(1,I)
      DO 30 I = 1,NP
      XMAX = AMAX1(XMAX,X(I))
   30 XMIN = AMIN1(XMIN,X(I))
      DO 50   I = 1,NC
      DO 50   J = 1,NP
       YMAX(I) = AMAX1(YMAX(I),Y (J,I))
       YMIN(I) = AMIN1(YMIN(I),Y (J,I))
   50 CONTINUE
```

```
      YX = YMAX(1)                                                      PLOT0373
      YN = YMIN(1)                                                      PLOT0376
      DO 60 I = 1,NC                                                    PLOT0380
      YX = AMAX1(YMAX(I),YX)                                            PLOT0390
      YN = AMIN1(YMIN(I),YN)                                            PLOT0400
   60 CONTINUE                                                          PLOT0410
      CALL SCAL (YX,YN,YSCAL,DY,YO,XE)                                  PLOT0420    52
      CALL SCAL (XMAX,XMIN,XSCAL,DX,XO,XE )                             PLOT0430
C                                                                       PLOT0440    54
      CALL CCP (1,NXP,XO,DX,X,0,1,YO       ,DY       ,0,0,Y(1,1),72,TTITLEPLOT0450
     1,72,BTITLE,72,YTITLE)                                             PLOT0460    56
      DO 70 I = 1,NC                                                    PLOT0470
      CALL CCP (-1,NXP,XO,DX,X,0,1,YO      ,DY       ,I,NCX,Y(1,I),0,0,72PLOT0480
     1,BTITLE,0,0)                                                      PLOT0490    62
      CALL CCP (2,NP,1)                                                 PLOT0500    64
   70 CONTINUE                                                          PLOT0510
      IF (NC .EQ. 1) GO TO 100                                          PLOT0520
      DO 75 I = 1,NC                                                    PLOT0530
      DO 75 J = 1,5                                                     PLOT0540
      SYMX(J + 1,I) = ATITLE(J,I)                                       PLOT0550
   75 CONTINUE                                                          PLOT0560
      IX = 600                                                          PLOT0570
      IY = 1000                                                         PLOT0580
      IF(NXP .NE. 20) GO TO 78                                          PLOT0590
      IX = 400                                                          PLOT0592
      IY = 700                                                          PLOT0594
   78 CONTINUE                                                          PLOT0596
      DO 80 I = 1,NC                                                    PLOT0600
      CALL CCP(3,IX,IY,0.15,36,SYMX(1,I))                               PLOT0610    93
      IY = IY - 30                                                      PLOT0620
   80 CONTINUE                                                          PLOT0630
  100 CALL CCP (5,3000,PLTIM)                                           PLOT0640    99
      IF (LAST .NE. 0) GO TO 110                                        PLOT0650
      CALL CCP(6)                                                       PLOT0660   104
  110 LAST = 0                                                          PLOT0670
      RETURN                                                            PLOT0680
      END                                                               PLOT0690
```

CCPX - EFN SOURCE STATEMENT - IFN(S) -

```
C     DUMMY PLOT ROUTINE INCLUDED TO PREVENT PROGRAM HANG - UPS        CCP00010
C     MESSAGE PRINTED EACH TIME CCP WOULD BE ENTERED                   CCP00020
      SUBROUTINE CCP                                                   CCP00030
      WRITE(6,1000)                                                    CCP00040   1
 1000 FORMAT(5X,51HINSTALLATION CALCOMP PLOTTING ROUTINES NOT INSERTED) CCP00050
      RETURN                                                           CCP00060
      END                                                              CCP00070
```

```
$IBFTC SCALX   NOLIST,DECK,M94,XR7,NODD
```

SCALX - EFN SOURCE STATEMENT - IFN(S) -

```
C     DUMMY SUBROUTINE FOR 'SCAL' TO SCALE NUMBERS FOR PLOTTING      SCAL0010
      SUBROUTINE SCAL                                                SCAL0020
      RETURN                                                         SCAL0030
      END                                                            SCAL0040
```

SUBROUTINE MON6

COMMON VARIABLES

COMMON BLOCK		COMAA		ORIGIN	00001	LENGTH	00001
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
ICHECK	00000	I					
COMMON BLOCK		COM61		ORIGIN	00002	LENGTH	00005
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		
COMMON BLOCK		COM62		ORIGIN	00007	LENGTH	00007
R	00000	R	D	00001	R	CK	00002
FM	00003	R	SD	00004	R	ARC	00005
HT	00006	R					
COMMON BLOCK		COM64		ORIGIN	00016	LENGTH	00070
X	00000	R					
COMMON BLOCK		COM65		ORIGIN	00106	LENGTH	00016
A	00000	R	B	00007	R		
COMMON BLOCK		COM66		ORIGIN	00124	LENGTH	00025
GLAM	00000	R					
COMMON BLOCK		COM67		ORIGIN	00151	LENGTH	00004
SINE	00000	R	COSI	00001	R	HM	00002
M	00003	I					
COMMON BLOCK		COM69		ORIGIN	00155	LENGTH	00026
KST	00000	I	NSOC	00001	I	PTEM	00002
COMMON BLOCK		COM6A		ORIGIN	00203	LENGTH	00016
STHE	00000	R	CTHE	00007	R		

	COMMON BLOCK		COM6B	ORIGIN	00221	LENGTH	00011
IOPT6	00000	I	IOPT7	00001	I	IOPT8	00002
IOPT9	00003	I	IOPT10	00004	I	IOPT11	00005
IOPT12	00006	I	IOPT13	00007	I	IOPT14	00010
JOPT	00000	I					
	COMMON BLOCK		COM6C	ORIGIN	00232	LENGTH	00004
NFLAG	00000	I	MFLAG	00001	I	MMFLAG	00002
TPLTIM	00003	R					
	COMMON BLOCK		COM6D	ORIGIN	00236	LENGTH	00001
THET	00000	R					
	COMMON BLOCK		COM6E	ORIGIN	00237	LENGTH	00001
NPLOT	00000	I					
	COMMON BLOCK		COM6F	ORIGIN	00240	LENGTH	00003
SIGMAX	00000	R	SIGTHE	00001	R	ISIGPT	00002
	COMMON BLOCK		//	ORIGIN	00243	LENGTH	33167
NSEMTS	00000	I	KS	00001	I	ISGMT	00002
IFTNO	00003	I	LOADNO	00004	I	LDNOPR	00005
NLOAD	00006	I	KKKKK	00007	I	PHI	00010
SINP	00017	R	COSP	00026	R	NPSEG	00035
LSP	00044	I	MLIST	00053	I	ECR	00062
ECY	00071	R	IOPT	00100	I	IRES	00116
NREC	00130	I	INTF	00155	I	NSGPL	00167
NPTPT	00201	I	NLDPL	00213	I	EI	00225
HI	00243	R	FMUI	00261	R	RI	00277
FLDI	00306	R	FLDF	00673	R	DLDI	01260
DLDF	01645	R	FMPO	02232	R	FNRO	02374
FNYO	02536	R	FNPTO	02700	R	ECRS	03042
ECYS	03123	R	RSS	03204	R	SCF	03265
IBC	04705	I	BC	04775	R	DLBC	05065
S	05155	R	FNTH	05631	R	FNPH	06305
OPHI	06761	R	OTHE	07435	R	STP	10111
AK	10235	R	AU	16461	R	AL	24705
RHV	31755	R	SAVE	33105	R	RHK	10235

MON6X STORAGE MAP

AK	10235	R	PS	14775	R	AL	24705
ECM	25215	R	ENT	26232	R	EMPT	27247
EMT	30264	R	SP	31445	R	ISFLAG	31600
RHV	31755	R	PB	31755	R	MIND	31755
FDC	32342	R	TP	32703	R		

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
RACI	33432	R	MAXHRM	33433	I	NPTSEG	33434
NRCSEG	33435	I	TMAX	33436	R	II	33437
I	33440	I	NROS	33441	I	J	33442
IPT	33443	I	HNN	33444	R	NFLG	33445
NPTS	33446	I	IK	33447	I		

ENTRY POINTS

MON6	SECTION	32

SUBROUTINES CALLED

GETM	SECTION	33	.FRDB.	SECTION	34	.FRWT.	SECTIO
FMRO	SECTION	36	SDAV	SECTION	37	STRT	SECTIO
DISP	SECTION	39	SDAVU	SECTION	40	SIN	SECTIO
COS	SECTION	42	STRTU	SECTION	43	DISPU	SECTIO
FORC	SECTION	45	.FWRD.	SECTION	46	SUFP	SECTIO
SUFPU	SECTION	48	.UN04.	SECTION	49	.FRLR.	SECTIO
.FBLT.	SECTION	51	.FBDT.	SECTION	52	.UN06.	SECTIO
.FFIL.	SECTION	54	.FCNV.	SECTION	55	E.1	SECTIO
E.2	SECTION	57	E.3	SECTION	58	E.4	SECTIO
CC.1	SECTION	60	CC.2	SECTION	61	CC.3	SECTIO
CC.4	SECTION	63	SYSLOC	SECTION	64		

EFN IFN CORRESPONDENCE

MON6X STORAGE MAP

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
2020	FORMAT	33477	2030	FORMAT	33514	2040	FORMAT
2050	FORMAT	33543	50	13A	33575	100	24A
300	28A	33630	650	132A	34274	330	58A
350	60A	33763	370	66A	33775	390	71A
410	80A	34067	415	83A	34076	430	86A
420	85A	34102	450	89A	34110	500	109A
460	91A	34114	480	95A	34132	470	93A
550	119A	34230	490	106A	34205	510	113A
520	115A	34220	540	117A	34224	570	121A
580	123A	34240	625	131A	34260	590	125A
600	129A	34253	900	191A	34535	680	142A
670	141A	34310	700	145A	34317	690	144A
720	148A	34325	740	152A	34334	730	151A
800	176A	34451	930	193A	34541	970	209A
950	196A	34563	960	205A	34623		

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 34651.

SUBROUTINE FMRO

COMMON VARIABLES

COMMON BLOCK COM61 ORIGIN 00001 LENGTH 00005

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		

COMMON BLOCK COM62 ORIGIN 00006 LENGTH 00007

R	00000	R	D	00001	R	CK	00002
FM	00003	R	SD	00004	R	ARC	00005
HT	00006	R					

COMMON BLOCK COM63 ORIGIN 00015 LENGTH 00043

HNT	00000	R	HMPT	00007	R	HOP	00016
HOM	00025	R	HMT	00034	R		

COMMON BLOCK COM64 ORIGIN 00060 LENGTH 00070

X	00000	R

COMMON BLOCK COM66 ORIGIN 00150 LENGTH 00025

GLAM	00000	R

COMMON BLOCK COM67 ORIGIN 00175 LENGTH 00004

SINE	00000	R	COSI	00001	R	HM	00002
M	00003	I					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
HALF	00201	R	MFLAG	00202	I	NPTS	00203
NM8	00204	I	NZ	00205	I	NP8	00206
I	00207	I	R1I	00210	R	R2I	00211
OM	00212	R	CMS	00213	R	ZLI	00214

FMROX STORAGE MAP

IHI	00215	I	HN	00216	R	UDOT	00217
KK	00220	I	LFLAG	00221	I		

ENTRY POINTS

FMRO SECTION 15

SUBROUTINES CALLED

FDIC	SECTION	16	SDAV	SECTION	17	SDAVU	SECTIC
LANB	SECTION	19	E.1	SECTION	20	E.2	SECTIC
E.3	SECTION	22	E.4	SECTION	23	CC.1	SECTIC
CC.2	SECTION	25	CC.3	SECTION	26	CC.4	SECTIC
SYSLOC	SECTION	28					

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
400	89A	01073	50	23A	00364	200	51A
75	39A	00532	250	67A	00676	300	87A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01305.

SUBROUTINE SDAV

COMMON VARIABLES

COMMON BLOCK		COM61		ORIGIN	00001	LENGTH	00005
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		
COMMON BLOCK		COM62		ORIGIN	00006	LENGTH	00007
R	00000	R	D	00001	R	CK	00002
FM	00003	R	SD	00004	R	ARC	00005
HT	00006	R					
COMMON BLOCK		COM63		ORIGIN	00015	LENGTH	00043
HNT	00000	R	HMPT	00007	R	HOP	00016
HOM	00025	R	HMT	00034	R		
COMMON BLOCK		COM64		ORIGIN	00060	LENGTH	00070
X	00000	R					
COMMON BLOCK		COM67		ORIGIN	00150	LENGTH	00004
SINE	00000	R	COSI	00001	R	HM	00002
M	00003	I					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
O12	00154	R	ONE	00155	R	IZ	00156
JZ	00157	I	DEL	00160	R	SLPE	00161
SLPH	00162	R	SLPU	00163	R	R1	00164
S1	00165	R	E1	00166	R	H1	00167
U1	00170	R	R1I	00171	R	IP	00172
JP	00173	I	L	00174	I	HN	00175
IX	00176	I					

ENTRY POINTS

SDAV	SECTION	13	SDAVU	SECTION	14

SUBROUTINES CALLED

E.1	SECTION	15	E.2	SECTION	16	E.3	SECTIO
E.4	SECTION	18	CC.1	SECTION	19	CC.2	SECTIO
CC.3	SECTION	21	CC.4	SECTION	22	SYSLOC	SECTIO

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
100	21A	00300	150	28A	00352	250	59A
200	56A	00502	160	41A	00447		

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00700.

SUBROUTINE LANB

COMMON VARIABLES

COMMON BLOCK	COM63		ORIGIN	00001		LENGTH	00043
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
HNT	00000	R	HMPT	00007	R	HOP	00016
HOM	00025	R	HMT	00034	R		
COMMON BLOCK	COM64		ORIGIN	00044		LENGTH	00070
X	00000	R					
COMMON BLOCK	COM65		ORIGIN	00134		LENGTH	00016
A	00000	R	B	00007	R		
COMMON BLOCK	COM66		ORIGIN	00152		LENGTH	00025
GLAM	00000	R					

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
VALU	00177	R	IJHT	00200	I		

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
JJ	00201	I	M	00202	I	VALUT	00203
ILGZ	00204	I	JI	00205	I	IMPL	00206
IMML	00207	I	JM	00210	I		

ENTRY POINTS

LANB SECTION 11

SUBROUTINES CALLED

SUNN	SECTION	12	SUNNU	SECTION	13	SYSLOC	SECTIO	

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
50	3A	00237	200	89A	00564	30	13A
190	85A	00561	80	41A	00364	75	38A
130	77A	00532	100	61A	00452	90	59A
125	75A	00523	170	83A	00552	400	116A
210	98A	00576	300	113A	00635	250	111A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01016.

SUNNX STORAGE MAP

SUBROUTINE SUNN

COMMON VARIABLES

COMMON BLOCK COM65 ORIGIN 00001 LENGTH 00016

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
A	00000	R	B	00007	R		

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
VALUT	00017	R	IMPL	00020	I	IMML	00021
JI	00022	I	JM	00023	I		

ENTRY POINTS

SUNN SECTION 5 SUNNU SECTION 6

SUBROUTINES CALLED

SYSLOC SECTION 7

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
50	11A	00067	100	29A	00146	80	26A
150	47A	00226	130	44A	00220		

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00370.

SUBROUTINE STRT

COMMON VARIABLES

COMMON BLOCK	COM61		ORIGIN	00001		LENGTH	00005
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		
COMMON BLOCK	COM62		ORIGIN	00006		LENGTH	00007
R	00000	R	D	00001	R	CK	00002
FM	00003	R	SD	00004	R	ARC	00005
HT	00006	R					
COMMON BLOCK	COM69		ORIGIN	00015		LENGTH	00026
KST	00000	I	NSOC	00001	I	PTEM	00002
COMMON BLOCK	COM6B		ORIGIN	00043		LENGTH	00011
IOPT6	00000	I	IOPT7	00001	I	IOPT8	00002
IOPT9	00003	I	IOPT10	00004	I	IOPT11	00005
IOPT12	00006	I	IOPT13	00007	I	IOPT14	00010
COMMON BLOCK	COM6D		ORIGIN	00054		LENGTH	00001
THET	00000	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
NPCAR	00055	I	NPFO	00056	I	NPDO	00057
NPPAGE	00060	I	NPT	00061	I	IZ	00062

ENTRY POINTS

STRT	SECTION	13	STRTU	SECTION	14

STORAGE MAP

SUBROUTINES CALLED

.FWRD.	SECTION	15	.UNO6.	SECTION	16	.FFIL.	SECT
.FCNV.	SECTION	18	SYSLOC	SECTION	19		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
1000	FORMAT	00075	1001	FORMAT	00116	1002	FORMAT
1010	FORMAT	00162	2000	FORMAT	00172	2010	FORMAT
2020	FORMAT	00201	2030	FORMAT	00207	40	6A
60	8A	00257	80	10A	00262	90	11A
200	30A	00372	140	16A	00276	145	18A
150	20A	00317	155	21A	00325	160	24A
170	25A	00346	180	27A	00352	190	29A
250	31A	00373					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00455.

SUBROUTINE DISP

COMMON VARIABLES

	COMMON BLOCK		COM61		ORIGIN	00001	LENGTH	00005
SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I		IJH	00001	I	IJ	00002
MS	00003	I		MF	00004	I		
	COMMON BLOCK		COM63		ORIGIN	00006	LENGTH	00043
HNT	00000	R		HMPT	00007	R	HOP	00016
HOM	00025	R		HMT	00034	R		
	COMMON BLOCK		COM64		ORIGIN	00051	LENGTH	00070
X	00000	R						
	COMMON BLOCK		COM65		ORIGIN	00141	LENGTH	00016
A	00000	R		B	00007	R		
	COMMON BLOCK		COM66		ORIGIN	00157	LENGTH	00025
GLAM	00000	R						
	COMMON BLOCK		COM67		ORIGIN	00204	LENGTH	00004
SINE	00000	R		COSI	00001	R	HM	00002
M	00003	I						
	COMMON BLOCK		COM68		ORIGIN	00210	LENGTH	00043
DELR	00000	R		DELY	00007	R	ANR	00016
ANY	00025	R		AMP	00034	R		
	COMMON BLOCK		COM69		ORIGIN	00253	LENGTH	00026
KST	00000	I		NSOC	00001	I	PTEM	00002
	COMMON BLOCK		COM6A		ORIGIN	00301	LENGTH	00016
STHE	00000	R		CTHE	00007	R		

DISPX STORAGE MAP

COMMON BLOCK COM6D ORIGIN 00317 LENGTH 00001

THET	00000	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IIJ	00320	I	IJJ	00321	I		

ENTRY POINTS

DISP	SECTION	23	DISPU	SECTION	24

SUBROUTINES CALLED

.FWRD.	SECTION	25	.UN06.	SECTION	26	.FFIL.	SECTIC
.FCNV.	SECTION	28	SYSLOC	SECTION	29		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
50	4A	00343	100	14A	00376	200	56A
120	16A	00402	130	20A	00403	150	43A
170	47A	00462	250	69A	00523	225	60A
1000	FORMAT	00333					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00574.

SUBROUTINE FORC

COMMON VARIABLES

COMMON BLOCK			COM61		ORIGIN	00001	LENGTH	00005
SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I		IJH	00001	I	IJ	00002
MS	00003	I		MF	00004	I		
COMMON BLOCK			COM62		ORIGIN	00006	LENGTH	00007
R	00000	R		D	00001	R	CK	00002
FM	00003	R		SD	00004	R	ARC	00005
HT	00006	R						
COMMON BLOCK			COM63		ORIGIN	00015	LENGTH	00043
HNT	00000	R		HMPT	00007	R	HOP	00016
HOM	00025	R		HMT	00034	R		
COMMON BLOCK			COM64		ORIGIN	00060	LENGTH	00070
X	00000	R						
COMMON BLOCK			COM67		ORIGIN	00150	LENGTH	00004
SINE	00000	R		COSI	00001	R	HM	00002
M	00003	I						
COMMON BLOCK			COM68		ORIGIN	00154	LENGTH	00043
DELR	00000	R		DELY	00007	R	ANR	00016
ANY	00025	R		AMP	00034	R		
COMMON BLOCK			COM69		ORIGIN	00217	LENGTH	00026
KST	00000	I		NSOC	00001	I	PTEM	00002
COMMON BLOCK			COM6A		ORIGIN	00245	LENGTH	00016
STHE	00000	R		CTHE	00007	R		
COMMON BLOCK			COM6B		ORIGIN	00263	LENGTH	00011

FORCX STORAGE MAP

IOPT6	00000	I	IOPT7	00001	I	IOPT8	00002
IOPT9	00003	I	IOPT10	00004	I	IOPT11	00005
IOPT12	00006	I	IOPT13	00007	I	IOPT14	00010

	COMMON BLOCK		COM6D	ORIGIN	00274	LENGTH	00001
THET	00000	R					

	COMMON BLOCK		COM6F	ORIGIN	00275	LENGTH	00003
SIGMAX	00000	R	SIGTHE	00001	R	ISIGPT	00002

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJJ	00300	I	IIJ	00301	I	HN	00302
HT1	00303	R	HT2	00304	R	PT	00305

ENTRY POINTS

FORC	SECTION	25

SUBROUTINES CALLED

SQRT	SECTION	26	.FWRD.	SECTION	27	.UNO6.	SECTI
.FFIL.	SECTION	29	.FCNV.	SECTION	30	E.1	SECTI
E.2	SECTION	32	E.3	SECTION	33	E.4	SECTI
CC.1	SECTION	35	CC.2	SECTION	36	CC.3	SECTI
CC.4	SECTION	38	SYSLOC	SECTION	39		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
2000	FORMAT	00326	50	3A	00335	100	14A
12C	15A	00402	250	54A	00645	150	17A
130	21A	00407	200	46A	00463	230	52A

300	64A	00673	400	81A	00731	320	66A
350	74A	00716					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00756.

SUBROUTINE SUFP

COMMON VARIABLES

COMMON BLOCK COM61 ORIGIN 00001 LENGTH 00005

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	00000	I	IJH	00001	I	IJ	00002
MS	00003	I	MF	00004	I		

COMMON BLOCK COM69 ORIGIN 00006 LENGTH 00026

KST	00000	I	NSOC	00001	I	PTEM	00002

COMMON BLOCK COM6A ORIGIN 00034 LENGTH 00016

STHF	00000	R	CTHF	00007	R	SCT	00000

COMMON BLOCK COM6B ORIGIN 00052 LENGTH 00011

IOPT6	00000	I	IOPT7	00001	I	IOPT8	00002
IOPT9	00003	I	IOPT10	00004	I	IOPT11	00005
IOPT12	00006	I	IOPT13	00007	I	IOPT14	00010

COMMON BLOCK COM6C ORIGIN 00063 LENGTH 00004

NFLAG	00000	I	MFLAG	00001	I	MMFLAG	00002
TPLTIM	00003	R					

COMMON BLOCK COM6D ORIGIN 00067 LENGTH 00001

THET	00000	R					

COMMON BLOCK COM6E ORIGIN 00070 LENGTH 00001

NPLOT	00000	I					

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
INPPS	00071	I	BTITLE	00103	R	YTITLE	00117
TTITLE	00133	R	ATITLE	00147	R	TITLE	00200

SUEPX STORAGE MAP

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
NUMB	00344	R	MARG	00345	I	NCX	00346
XSCAL	00347	R	IONE	00350	I	MTHE	00351
TMAX	00352	R	RADI	00353	R	I	00354
J	00355	I	NPTS	00356	I	NTE	00357
HN	00360	R	IPT	00361	I	IP	00362
IZ	00363	I	VAL	00364	R	LAST	00365
PLTIM	00366	R	NPPS	00367	I	EIJK	00370
EIJ	00371	R					

ENTRY POINTS

SUEP	SECTION	17	SUEPU	SECTION	18

SUBROUTINES CALLED

IBCDER	SECTION	19	SIN	SECTION	20	COS	SECTIO
PLOT	SECTION	22	CCP	SECTION	23	E.1	SECTIO
E.2	SECTION	25	E.3	SECTION	26	E.4	SECTIO
CC.1	SECTION	28	CC.2	SECTION	29	CC.3	SECTIO
CC.4	SECTION	31	SYSLOC	SECTION	32		

EEN IEN CORRESPONDENCE

EEN	IEN	LOCATION	EEN	IEN	LOCATION	EEN	IEN
50	7A	00440	500	79A	00771	70	18A
100	23A	00514	120	25A	00520	320	70A
130	40A	00614	300	66A	00722	150	54A
170	63A	00706	480	77A	00765	900	150A
540	93A	01010	530	89A	01003	520	86A
550	97A	01015	570	108A	01047	800	147A
600	120A	01125	620	122A	01131	650	140A
640	135A	01231					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01454.

STORAGE MAP

SUBROUTINE IBCDFR

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IRD	00001	I					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
NCU	00004	I	I	00005	I	K	00006
NCU1	00007	I	KX	00010	I		

ENTRY POINTS

IBCDFR SECTION 3

SUBROUTINES CALLED

CC.1	SECTION	4	CC.2	SECTION	5	CC.3	SECTION
CC.4	SECTION	7	SYSLOC	SECTION	8		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
20	11A	00124	30	20A	00152		

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00222.

SUBROUTINE PLOT

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
YMIN	00001	R	YMAX	00010	R	DXO	00017
SYMX	00035	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
YSCAL	00107	R	NXP	00110	I	XMAX	00111
XMIN	00112	R	I	00113	I	YX	00114
YN	00115	R	DY	00116	R	YO	00117
XE	00120	R	DX	00121	R	XO	00122
IX	00123	I	IY	00124	I		

ENTRY POINTS

PLOT SECTION 3

SUBROUTINES CALLED

SCAL SECTION 4 CCP SECTION 5 SYSLOC SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
20	16A	00233	30	25A	00253	50	40A
60	49A	00374	70	65A	00516	100	98A
75	78A	00550	78	88A	00570	80	95A
110	105A	00635					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 01001.

CCPX STORAGE MAP

SUBROUTINE CCP

ENTRY POINTS

CCP	SECTION	3

SUBROUTINES CALLED

.FWRD.	SECTION	4	.UNO6.	SECTION	5	.FFIL.	SECTION	
.FCNV.	SECTION	7	SYSLOC	SECTION	8			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	L
1000	FORMAT	00007						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00044.

SCALX STORAGE MAP

SUBROUTINE SCAL

ENTRY POINTS

SCAL SECTION 3

SUBROUTINES CALLED

SYSLOC SECTION 4

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00022.

```
$IBFTC MCN7X   NOLIST,DECK,M94,XR7,NODD
```

```
C     MONITOR SPECIAL PRINTING IN PROGRAM CHECKOUT                      MON70010
      SUBROUTINE MON7                                                   MON70020
      DIMENSION RHK(2400),BETA(5,7),IBETA(5,7),RHM(2400)                MON70030
      EQUIVALENCE (AK(1,1),RHK(1)),(AL(166,13),BETA(1,1)),              MON70040
     1 (AL(166,12),IBETA(1,1)),(AL(1,1),RHM( 1))                        MON70050
      COMMON NSGMTS, KS, ISGMT, IFTNO, LOADNO, LDNOPR, NLOAD, KKKKK     MON70060
      COMMON PHI(7),SINP(7),COSP(7),NPSEG(7),LSP(7),MLIST(7),ECR(7),    MON70070
     1 ECY(7),IOPT(14),IRES(10),NREC(21),INTF(10),NSGPL(10),NPTPT(10),  MON70080
     2 NLDPL(10)                                                        MON70090
      COMMON EI(2,7),HI(2,7),FMUI(2,7),RI(7)                            MON70100
      COMMON FLDI(5,7,7),FLDF(5,7,7),DLDI(5,7,7),DLDF(5,7,7)            MON70110
      COMMON FMPO(2,7,7),FNRO(2,7,7),FNYO(2,7,7),FNPTO(2,7,7),ECRS(7,7),MON70120
     1 ECYS(7,7),RSS(7,7),SCF(16,7,7)                                   MON70130
      COMMON IBC(4,7,2),BC(4,7,2),DLBC(4,7,2)                           MON70140
      COMMON S(300),FNTH(300),FNPH(300),OPHI(300),OTHE(300),STP(12,7)   MON70150
      COMMON AK(140,23),AU(140,23),AL(200,13),RHV(600),SAVE(5,10)       MON70160
 1000 FORMAT(3X,1HM,5X,2HI1,6X,7HBETA(1),5X,2HI2,6X,7HBETA(2),5X,       MON70170
     1 2HI3,6X,7HBETA(3),5X,2HI4,6X,7HBETA(4),5X,2HI5,6X,7HBETA(5) )    MON70180
 1001 FORMAT(1H ,   I3,3X, 5(I4,1PE16.6) )                              MON70190
 4000 FORMAT(1H0,4X,5HPOINT,3X,10HOMEGA(PHI),5X,10HOMEGA(THE),6X,       MON70200
     1 7HEPSILON)                                                       MON70210
 4002 FORMAT(1H0,4X,5HPOINT,5X,6HFDC(1),9X,6HFDC(2),9X,6HFDC(3))        MON70220
 4003 FORMAT(1H ,I3,6X,4(1PE15.7))                                      MON70230
 4004 FORMAT(1H1,24HSPECIAL ITERATION FORCES//6HOPOINT,10X,6HN(THE),    MON70240
     1 9X,6HN(PHI),7X,8HOMEGA(P),7X,8HOMEGA(T) )                        MON70250
      IJH = INTF(5)                                                     MON70260
      IF(INTF(2).LE.0) GO TO 400                                        MON70270
      IF(IOPT(9).EQ.0) GO TO 350                                        MON70275
      NPS = LSP(NSGMTS)+NPSEG(NSGMTS)-1                                 MON70280
      WRITE(6,4004)                                                     MON70290   11
      DO 300 I=1,NPS                                                    MON70300
      WRITE(6,4003) I,FNTH(I),FNPH(I),OPHI(I),OTHE(I)                   MON70310   14
  300 CONTINUE                                                          MON70320
      CALL PRNY(RHM,LSP,NPSEG,NSGMTS,IJH)                               MON70330   22
  350 IOPT(12) = IOPT(12)-1                                             MON70340
      KKKKK=3                                                           MON70350
      IF( (IOPT(12).LE.0) .AND. (IOPT(11).LE.0) ) KKKKK = 8             MON70360
```

```
      IOPT(10) = 1                                                      MON70370
      RETURN                                                            MON70380
  400 CONTINUE                                                          MON70390
      IF(IOPT(9).EQ.0) GO TO 650                                        MON70395
      CALL PRNY(RHK,LSP,NPSEG,NSGMTS,IJH)                               MON70400    31
      WRITE(6,1000)                                                     MON70410    32
      DO 600 IJH = 1,KS                                                 MON70420
      WRITE(6,1001) MLIST(IJH),(IBETA(I,IJH),BETA(I,IJH),I=1,5)         MON70430    35
  600 CONTINUE                                                          MON70440
  650 IOPT(11) = IOPT(11)-1                                             MON70450
      KKKKK = 4                                                         MON70460
      IF(IOPT(10).GT.0) KKKKK = 3                                       MON70470
      IOPT(10) = 0                                                      MON70480
      IF(IOPT(11).LE.0) KKKKK = 8                                       MON70490
 2000 CONTINUE                                                          MON70500
      RETURN                                                            MON70510
      END                                                               MON70520
```

```
$IBFTC PRNYX    NOLIST,DECK,M94,XR7,NODD
```

```
C     PRINT NEW RIGHT HAND VECTOR                                      PRNY0010
      SUBROUTINE PRNY(XVAL,LSP,NPSEG,NSGMTS,IJH)                       PRNY0020
      DIMENSION XVAL( 1),LSP( 1),NPSEG( 1)                             PRNY0030
 1000 FORMAT(1H ,6HPOINT ,I2/8(1PE13.5) )                              PRNY0040
 1001 FORMAT(1H1,5X,7HSEGMENT,I2,5X,16HHARMONIC INDEX =,I2)            PRNY0050
C     INITIALIZE FLAGS                                                 PRNY0060
      II = 0                                                           PRNY0070
      IF = 0                                                           PRNY0080
C     START LOOP ON SEGMENTS                                           PRNY0090
      DO 400 IJK = 1,NSGMTS                                            PRNY0100
      IJ = 1                                                           PRNY0110
      II = IF+1                                                        PRNY0120
      IF = IF+4                                                        PRNY0130
      NPT1 = NPSEG(IJK) - 1                                            PRNY0140
      WRITE(6,1001) IJK,IJH                                            PRNY0150
C     PRINT LEADING DISCONTINUITY EQN. VALUES OR BOUNDARY COND. VALUES PRNY0160   11
      WRITE(6,1000) IJ,(XVAL(J),J=II,IF)                               PRNY0170   12
      DO 300 IJ = 1,NPT1                                               PRNY0180
      II = IF+1                                                        PRNY0190
      IF = IF+8                                                        PRNY0200
C     PRINT INTERIOR POINT VALUES                                      PRNY0210
      WRITE(6,1000) IJ,(XVAL(J),J=II,IF)                               PRNY0220   20
  300 CONTINUE                                                         PRNY0230
      II = IF+1                                                        PRNY0240
      IF = IF+4                                                        PRNY0250
      IJ = IJ+1                                                        PRNY0260
C     PRINT ENDING  DISCONTINUITY EQN. VALUES OR BOUNDARY COND. VALUES PRNY0270
      WRITE(6,1000) IJ,(XVAL(J),J=II,IF)                               PRNY0280   29
  400 CONTINUE                                                         PRNY0290
      RETURN                                                           PRNY0300
      END                                                              PRNY0310
```

SUBROUTINE MCN7

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 33167

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
NSGMTS	00000	I	KS	00001	I	ISGMT	00002
IFTNO	00003	I	LOADNO	00004	I	LDNOPR	00005
NLCAD	00006	I	KKKKK	00007	I	PHI	00010
SINP	00017	R	COSP	00026	R	NPSEG	00035
LSF	00044	I	MLIST	00053	I	ECR	00062
ECY	00071	R	IOPT	00100	I	IRES	00116
NREC	00130	I	INTF	00155	I	NSGPL	00167
NPTPT	00201	I	NLDPL	00213	I	EI	00225
HI	00243	R	FMUI	00261	R	RI	00277
FLDI	00306	R	FLDF	00673	R	DLDI	01260
DLDF	01645	R	FMPO	02232	R	FNRO	02374
FNYO	02536	R	FNPTO	02700	R	ECRS	03042
ECYS	03123	R	RSS	03204	R	SCF	03265
IBC	04705	I	BC	04775	R	DLBC	05065
S	05155	R	FNTH	05631	R	FNPH	06305
OPHI	06761	R	OTHE	07435	R	STP	10111
AK	10235	R	AU	16461	R	AL	24705
RHV	31755	R	SAVE	33105	R	RHK	10235
AL	24705	R	BETA	31712	R	IBETA	31402
RHM	24705	R					

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
IJK	33170	I	NPS	33171	I	I	33172

ENTRY POINTS

MON7 SECTION 4

SUBROUTINES CALLED

.FWRD.	SECTION	5	PRNY	SECTION	6	.UN06.	SECTION
.FFIL.	SECTION	8	.FCNV.	SECTION	9	SYSLOC	SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
1000	FORMAT	33206	1001	FORMAT	33233	4000	FORMAT
4002	FORMAT	33254	4003	FORMAT	33265	4004	FORMAT
400	27A	33422	350	23A	33400	300	19A
650	45A	33503	600	42A	33501	2000	51A

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 33546.

SUBROUTINE PRNY

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
II	00001	I	IF	00002	I	IJK	00003
IJ	00004	I	NPT1	00005	I		

ENTRY POINTS

PRNY SECTION 3

SUBROUTINES CALLED

.FWRD.	SECTION	4	.UN06.	SECTION	5	.FFIL.	SECTION
.FCNV.	SECTION	7	SYSLOC	SECTION	8		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN
1000	FORMAT	00017	1001	FORMAT	00024	400	34A
300	25A	00145					

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00244.