MCR-75-18
Contract NAS8-30761

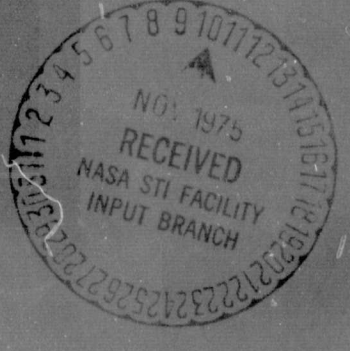
Volume III

Final Report

August 1975

Program Code

Dynamic Analysis of a Flexible Spacecraft with Rotating Components



Prepared for:
Marshall Space Flight Center
Huntsville, Alabama 35823

MARTIN MARIETTA

FOREWORD

This report, prepared by the Dynamics and Loads Section, Martin Marietta Corporation, Denver Division, under Contract NAS8-30761, presents the results of a study that developed a digital computer program for dynamic analysis of a flexible spacecraft with rotating components. The study was performed from April 1974 to August 1975 and was administered by the National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama, under the direction of Dr. John Glaese.

The report is published in three volumes:

Volume I - Analytical Developments
Volume II - Program Guide and Examples
Volume III - Program Code

ABSTRACT

This document details analytical procedures and digital computer code for the dynamic analysis of a flexible spacecraft with rotating components. Two major subject areas are considered:

(1) nonlinear response in the time domain, and
(2) linear response in the frequency domain.

The spacecraft is assumed to consist of an assembly of connected rigid or flexible subassemblies. The total system is not restricted to a topological connection arrangement and may be acting under the influence of passive or active control systems and external environments.

The analytics and associated digital code provide the user with the capability to establish spacecraft system nonlinear total response for specified initial conditions, linear perturbation response about a calculated or specified nominal motion, general frequency response and graphical display, and spacecraft system stability analysis.

The document is presented in three volumes.

```
[HDG,P    DYNAMO                                                          -000001
[FOR,IS   DYNAMO                                                          -000002
      COMPILER (XM=1),(EQUIV=CMN)                                         -000003
C   PROGRAM DYNAMO -- DYNAMIC ANALYSIS OF A FLEXIBLE SPACECRAFT WITH      -000004
C                     ROTATING COMPONENTS, CONTRACT NAS8-30761,           -000005
C                     PREPARED FOR MARSHALL SPACE FLIGHT CENTER           -000006
C                                                                         -000007
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                  -000008
C                                                                          000009
           COMMON /DRATIO/                                                000010
     *     IFL1,IFL2,DRVEC(150)                                          10000011
           COMMON /GGDATA/                                                000012
     *     GAMGI(3),GMAG,RCMAG                                            000013
           COMMON /ILINER/                                                000014
     *     IFLNER                                                         000015
           COMMON /MAXMUM/                                                000016
     *     NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU              000017
           COMMON /MISCNO/                                                000018
     *     NOPRNT,NOPLOT                                                  000019
           COMMON /NUMBRS/                                                000020
     *     ZRO,ONE,TWO,TRES                                               000021
           COMMON /PLTDTA/                                                000022
     *     NRPLOT,NCPLOT                                                  000023
           COMMON /SPECIF/                                                000024
     *     BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),     1600025
     *     DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),            1700026
     *     NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1800027
     *     LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                           1900028
           COMMON /TAPENO/                                                000029
     *     NTAPE1,NTAPE2,NTAPE3                                           000030
C                                                                          000031
      IFL1 = 0                                                            000032
C                                                                          000033
      NTAPE1 = 1                                                          000034
      NTAPE2 = 2                                                          000035
      NTAPE3 = 3                                                          000036
      NBMAX  =  5                                                        4100037
      NHMAX  =  5                                                        4200038
      NSPMAX = 10                                                        4300039
      NMWMAX =  5                                                        4400040
      NMWBOD =  3                                                        4500041
      NMDBOD =  6                                                        4600042
      KMU    = 15                                                        4700043
      KY     = 250                                                       4800044
      KU     =  65                                                       4900045
C                                                                          000046
      ZRO  = 0.D 0                                                        000047
      ONE  = 1.D 0                                                        000048
      TWO  = 2.D 0                                                        000049
      TRES = 3.D 0                                                        000050
```

```
C                                                                       000051
  999 CALL START                                                        000052
      CALL COMENT                                                       000053
C                                                                       000054
      REWIND NTAPE1                                                     000055
      REWIND NTAPE2                                                     000056
      REWIND NTAPE3                                                     000057
C                                                                       000058
      CALL DYNSAA                                                      -000059
      CALL DYNSBB                                                      -000060
      CALL DYNSEE(IFLNER,NOPLOT)                                       -000061
      GO TO 999                                                         000062
C                                                                       000063
      END                                                               000064
```

```
[HDG,P    ADDT                                                    -000065
[FOR,IS   ADDT                                                    -000066
      COMPILER (XM=1),(EQUIV=CMN)                                 -000067
      DOUBLE PRECISION FUNCTION ADDT (IC,T)                       -000068
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                         -000069
            COMMON /VECTOR/                                        000070
     *     Y(250),YDT(250)                                       2000071
      IF (IC .EQ. 19) GO TO 20                                     000072
      ADDT = YDT(65)                                               000073
      RETURN                                                       000074
   20 ADDT = YDT(66)                                               000075
      RETURN                                                       000076
C                                                                  000077
      END                                                          000078
```

```
[HDG,P    ADD3                                                          -000079
[FOR,IS   ADD3                                                          -000080
      COMPILER (XM=1),(EQUIV=CMN)                                       -000081
      SUBROUTINE ADD3(ALPHA,A,BETA,B,GAMMA,C,NR,NC,KR)                   000082
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -000083
C                                                                        000084
C     MATRIX ADDITION  A = ALPHA*A + BETA*B + GAMMA*C                    000085
C         WHERE ALPHA, BETA, GAMMA ARE INPUT SCALARS AND                 000086
C         A, B, C ARE INPUT MATRICES (NR,NC)                             000087
C                                                                        000088
      DIMENSION A(KR,1),B(KR,1),C(KR,1)                                  000089
C                                                                        000090
      DO 10 I=1,NR                                                       000091
      DO 10 J=1,NC                                                       000092
   10 A(I,J) = ALPHA*A(I,J) + BETA*B(I,J) + GAMMA*C(I,J)                 000093
C                                                                        000094
      RETURN                                                             000095
      END                                                                000096
```

```
[HDG,P     ADT                                                          -000097
[FOR,IS    ADT                                                          -000098
       COMPILER (XM=1),(EQUIV=CMN)                                      -000099
       DOUBLE PRECISION FUNCTION ADT (IC,T)                             -000100
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)                              -000101
              COMMON /VECTOR/                                            000102
      *      Y(250),YDT(250)                                           2000103
C                                                                        000104
       IF (IC .EQ. 19) GO TO 20                                          000105
       ADT  = Y(65)                                                      000106
       RETURN                                                            000107
   20  ADT  = Y(66)                                                      000108
       RETURN                                                            000109
C                                                                        000110
       END                                                               000111
```

```
[HDG,P    ALPHAA                                                    -000112
[FOR,IS   ALPHAA                                                    -000113
      COMPILER (XM=1),(EQUIV=CMN)                                   -000114
      SUBROUTINE ALPHAA (ALPHA,A,Z,NR,NC,KR)                         000115
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                            -000116
      DIMENSION A(KR,1), Z(KR,1)                                    000117
C                                                                   000118
C  SCALAR ALPHA TIMES MATRIX A. (ALPHA * A = Z).                    000119
C  MATRICES A,Z MAY SHARE SAME CORE LOCATIONS.                      000120
C  CODED BY  RL WOHLEN.  FEBRUARY 1965.                             000121
C                                                                   000122
C     SUBROUTINE ARGUMENTS                                          000123
C  ALPHA = INPUT  SCALAR.                                           000124
C  A     = INPUT  MATRIX. SIZE(NR,NC).                              000125
C  Z     = OUTPUT RESULT MATRIX. SIZE(NR,NC).                       000126
C  NR    = INPUT  NUMBER OF ROWS IN MATRICES A,Z.                   000127
C  NC    = INPUT  NUMBER OF COLS IN MATRICES A,Z.                   000128
C  KR    = INPUT  ROW DIMENSION OF A,Z IN CALLING PROGRAM.          000129
C                                                                   000130
      DO 10 I=1,NR                                                  000131
      DO 10 J=1,NC                                                  000132
   10 Z(I,J) = ALPHA * A(I,J)                                       000133
      RETURN                                                        000134
      END                                                           000135
```

```
[HDG,P    ASIMLR                                                        -000136
[FOR,IS   ASIMLR                                                        -000137
      COMPILER (XM=1),(EQUIV=CMN)                                       -000138
      SUBROUTINE ASIMLR (A,B,IV,KR)                                      000139
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -000140
C                                                                        000141
C   SUBROUTINE ESTABLISHES TRANSFORMED PARTIAL DERIVATIVE MATRIX         000142
C              BY PERFORMING A SIMILARITY TRANSFORMATION TO             000143
C              EXCHANGE PLANT STATE VARIABLES ,Y, FOR SENSOR SIGNALS,XSS 000144
C              AND CONTROL SYSTEM VARIABLES ,DELTA, FOR TORQUE          000145
C              VARIABLES ,B.                                            000146
C                                                                       000147
C   THE VAIRABLE SEQUENCE IS REORDERED FROM Y,DELTA,XSS,B               000148
C                                       TO Y,XSS,DELTA,B                000149
C                                                                       000150
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----               000151
C                                                                       000152
C   A      = INPUT  MATRIX OF PARTIAL DERIVATIVES.  COORDINATE ORDER    000153
C                   IS Y,DELTA,XSS,B.  SIZE IS NJQ,NX                   000154
C   B      = OUTPUT  TRANSFORMED AND REORDERED PARTIAL DERIVATIVE       000155
C                    MATRIX.  ORDER IS Y,XSS,DELTA,B.  SIZE IS NX,NX.   000156
C   IV     = INPUT  INTGER WORK VECTOR.  SIZE MUST BE AT LEAST NX.      000157
C   KR     = INPUT  ROW DIMENSION SIZE OF A AND B IN CALLING PROGRAM.   000158
C                                                                       000159
      DIMENSION A(KR,1), B(KR,1), IV(1)                                 000160
C                                                                       000161
      COMMON /LDSIZE/                                                   000162
     2               NX, NY, NDLTA, NXSS, NB, NJQ, NY2, ND2             000163
      COMMON /TAPENO/                                                   000164
     4               NUT1, NUT2, NUT3                                   000165
      COMMON /VECTOR/                                                   000166
     E               Y   (250), YD  (250)                              43000167
C                                                                       000168
C   SET UP C.  LOWER PART OF -A- REQUIRED TO OETAIN -T-.                000169
C                                                                       000170
      NR = NJQ - NX                                                     000171
      NC = NX + NR                                                      000172
C                                                                       000173
      CALL ZERO (B,NR,NC,KR)                                            000174
      DO 10 I=1,NP                                                      000175
      L=I + NX                                                          000176
      DO 10 J=1,NX                                                      000177
      B(I,J) = A(L,J)                                                   000178
      IF (I .EQ. J) B(I,L) = -1.DO                                      000179
   10 CONTINUE                                                          000180
C                                                                       000181
C   B = -C-                                                             000182
C   ESTABLISH SEARCH LIMIT FOR SUBROUTINE FINDT                         000183
C                                                                       000184
      NS = NX                                                           000185
```

```
      CALL FINDT (B,NR,NC,NS,A,NRET,KR,KR)                              000186
      CALL WRITE (A,NRET,NRET,4H-T- ,KR)                                000187
C                                                                       000188
C  A = -T-                                                              000189
C                                                                       000190
C  FORM -A*- = T(INV) A T                                               000191
C                                                                       000192
      READ (NUT2)  ((B(I,J),I=1,KR),J=1,KR)                             000193
      WRITE (NUT2)  ((A(I,J), I=1,KR),J=1,KR)                           000194
      REWIND NUT2                                                       000195
C  INVERT -T- USING GAUSSI                                              000196
      CALL GAUSSI (A,B,NRET,KR)                                         000197
C  B = T(INV)                                                           000198
C  TRANSFORM STATE VECTOR FOR POSSIBLE USE IN LINEARIZED RESPONSE.      000199
      CALL MULTB (B,Y,NRET,NRET,1,KR,KR)                                000200
      CALL WRITE (Y,1,NRET,4H Y* ,1)                                    000201
      READ (NUT2) ((A(I,J),I=1,KR),J=1,KR)                              000202
      CALL MULTA (B,A,NX,NX,NX,KR,KR)                                   000203
C  B= T(INV) * -A-                                                      000204
      READ (NUT2) ((A(I,J),I=1,KR),J=1,KR)                              000205
      REWIND NUT2                                                       000206
      CALL MULTB (B,A,NX,NX,NX,KR,KR)                                   000207
C B = -A*-                                                              000208
C                                                                       000209
C  REORDER    FROM Y,DELTA,XSS,B                                        000210
C               TO Y,XSS,DELTA,B                                        000211
C                                                                       000212
      DO 20 I=1,NX                                                      000213
   20 IV(I) = I                                                         000214
      DO 30 I=1,NXSS                                                    000215
      L = NY2 + I                                                       000216
      K = L + ND2                                                       000217
   30 IV(K) = L                                                         000218
      DO 40 I=1,ND2                                                     000219
      K = NY2 + I                                                       000220
      L = NY2 + NXSS + I                                                000221
   40 IV(K) = L                                                         000222
      CALL ZERO (B,NX,NX,KR)                                            000223
      CALL REVADD (1.D0,A,IV,IV,B,NX,NX,NX,NX,KR,KR)                    000224
C                                                                       000225
      RETURN                                                            000226
      END                                                               000227
```

```
[HDG,P    BABT                                                          -000228
[FOR,IS   BABT                                                          -000229
      COMPILER (XM=1),(EQUIV=CMN)                                       -000230
      SUBROUTINE BABT (A,B,Z,NRB,NCB,KA,KB)                              000231
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -000232
      DIMENSION A(KA,1), B(KB,1), Z(KB,1)                                000233
      COMMON /LWRKV1/ W( 50)                                           48000234
C                                                                        000235
C   SPECIAL TRIPLE MATRIX PRODUCT.  B*A*B(TRANSPOSE) = Z.                000236
C   A MUST BE SYMMETRIC TO GET CORRECT ANSWER.                           000237
C   Z WILL BE SYMMETRIC. UPPER HALF CALCULATED, REFLECTED TO LOWER HALF. 000238
C   THE MAXIMUM SIZE IS                                                  000239
C      NCB = 500                                                         000240
C   DEVELOPED BY CARL BODLEY.  JANUARY 1965.                             000241
C   LAST REVISION BY RL WOHLEN.  JULY 1972.                              000242
C                                                                        000243
C      SUBROUTINE ARGUMENTS                                              000244
C   A   = INPUT  INNER MATRIX. SIZE(NCB,NCB).                            000245
C   B   = INPUT  OUTER MATRIX. SIZE(NRB,NCB).                            000246
C   Z   = OUTPUT RESULT MATRIX. SIZE(NRB,NRB).                           000247
C   NRB = INPUT  NUMBER OF ROWS OF MATRIX B, SIZE OF MATRIX Z.           000248
C   NCB = INPUT  NUMBER OF COLS OF MATRIX B, SIZE OF MATRIX A. MAX=500.  000249
C   KA  = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.                  000250
C   KB  = INPUT  ROW DIMENSION OF B,Z IN CALLING PROGRAM.                000251
C                                                                        000252
C                                                                        000253
      DO 40 J=1,NRB                                                      000254
      DO 20 L=1,NCB                                                      000255
      S = 0.DO                                                           000256
      DO 10 K=1,NCB                                                      000257
   10 S = S + A(L,K)*B(J,K)                                              000258
   20 W(L) = S                                                           000259
      DO 40 I=1,J                                                        000260
      S = 0.DO                                                           000261
      DO 30 L=1,NCB                                                      000262
   30 S = S + B(I,L)*W(L)                                                000263
      Z(I,J) = S                                                         000264
   40 Z(J,I) = S                                                         000265
      RETURN                                                             000266
      END                                                                000267
```

```
[HDG,P    BAKSLV                                                    -000268
[FOR,IS   BAKSLV                                                    -000269
      COMPILER (XM=1),(EQUIV=CMN)                                   -000270
      SUBROUTINE BAKSLV (BW,NLAM,V,D,KBW)                            000271
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                            -000272
      DIMENSION BW(KBW,1),V(1),D(1)                                  000273
C                                                                    000274
      DO 25 I=2,NLAM                                                 000275
      IM1 = I - 1                                                    000276
      DO 25 J=1,IM1                                                  000277
   25 V(I) = V(I) - BW(J,I)*V(J)                                     000278
      DO 27 I=1,NLAM                                                 000279
   27 V(I) = V(I)/D(I)                                               000280
      NL1 = NLAM - 1                                                 000281
      DO 30 I=1,NL1                                                  000282
      L = NLAM - I                                                   000283
      LP1 = L + 1                                                    000284
      DO 30 J=LP1,NLAM                                               000285
   30 V(L) = V(L) - BW(L,J)*V(J)                                     000286
C                                                                    000287
      RETURN                                                         000288
      END                                                            000289
```

```
[HDG,P    BDOTQP                                                        -000290
[FOR,IS   BDOTQP                                                        -000291
      COMPILER (XM=1),(EQUIV=CMN)                                       -000292
      SUBROUTINE BDOTQP (L,BDTQ,BDTP)                                    000293
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -000294
      DIMENSION BDTQ(6,1),BDTP(6,1)                                      000295
C                                                                        000296
            COMMON /BHBSRD/                                              000297
     *     BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                200298
            COMMON /HANDS /                                              000299
     *     HATH(3, 6, 8),SIGH(3, 6, 8),HATS(3, 6,10),SIGS(3, 6,10)      400300
            COMMON /NUMBRS/                                              000301
     *     ZRO,ONE,TWO,TRES                                              000302
            COMMON /PINRP /                                              000303
     *     PIN(3,3, 5), RP2(3,3, 5), RP3(3,3, 5)                        1300304
            COMMON /SPECIF/                                              000305
     *     BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),    1600306
     *     DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),           1700307
     *     NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1800308
     *     LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                          1900309
            COMMON /VECTOR/                                              000310
     *     Y(250),YDT(250)                                              2000311
C                                                                        000312
      DIMENSION PINDT(3,3),HXDQN(3),SXDQN(3),HXDPM(3),SXDPM(3),RQP(3,3), 000313
     *     RPM(3,3),RQN(3,3),RPN(3,3),RQM(3,3),DRPQ(3,3),DRQP(3,3),      000314
     *     DRQN(3,3),DRQM(3,3),SNQ(3,3),SMP(3,3),WQN(3),WPM(3),WPQP(3),  000315
     *     WSK(3,3), VEC(3)                                             000316
C                                                                        000317
      DO 3 I=1,6                                                         000318
      DO 3 J=1,6                                                         000319
      BDTQ(I,J) = ZRO                                                    000320
    3 BDTP(I,J) = ZRO                                                    000321
      IF (L .EQ. 1) GO TO 100                                            000322
C                                                                        000323
CC  GET D/DT(PI(INVERSE))                                                000324
      B2DT = BETAHD(2,L)                                                 000325
      B3DT = BETAHD(3,L)                                                 000326
      DO 5 I=1,3                                                         000327
      DO 5 J=1,3                                                         000328
    5 PINDT(I,J) = B2DT*RP2(I,J,L) + B3DT*RP3(I,J,L)                     000329
C                                                                        000330
      NOBQ = ITOPOL(1,L)                                                 000331
      NOBP = ITOPOL(2,L)                                                 000332
      LOQ = LOCU(NOBQ) + 6                                               000333
      LOP = LOCU(NOBP) + 6                                               000334
      LEQ = IRGFLX(NOBQ)                                                 000335
      LEP = IRGFLX(NOBP)                                                 000336
      LHSQ = 2*L - 3                                                     000337
      LHSP = LHSQ + 1                                                    000338
      IF (LEQ .EQ. 0) GO TO 10                                           000339
```

```
      CALL MULT3  (HATH(1,1,LHSQ),Y(LOQ),HXDQN,3,LEQ,1,3,1,1)          000340
      CALL MULT3  (SIGH(1,1,LHSQ),Y(LOQ),SXDQN,3,LEQ,1,3,1,1)          000341
   10 IF (LEP .EQ. 0) GO TO 20                                          000342
      CALL MULT3  (HATH(1,1,LHSP),Y(LOP),HXDPM,3,LEP,1,3,1,1)          000343
      CALL MULT3  (SIGH(1,1,LHSP),Y(LOP),SXDPM,3,LEP,1,3,1,1)          000344
   20 LRNQ = 6*(L-2) + 3                                               000345
      LRMP = LRNQ + 1                                                  000346
      LRPQ = LRNQ + 2                                                  000347
      LDNQ = 7*(L-2) + 3                                               000348
      LDMP = LDNQ + 1                                                  000349
      DO 15 I=1,3                                                      000350
      DO 15 J=1,3                                                      000351
      RQP(I,J) = RH(J,I,LRPQ)                                          000352
      RQN(I,J) = RH(J,I,LRNQ)                                          000353
   15 RPM(I,J) = RH(J,I,LRMP)                                          000354
      CALL MULT3  (RQP,RPM,RQM,3,3,3,3,3,3)                            000355
      CALL MULT3  (RH(1,1,LRPQ),RQN,RPN,3,3,3,3,3,3)                   000356
      CALL SKEWV3 (DH(1,LDNQ),SNQ,3,3)                                 000357
      CALL SKEWV3 (DH(1,LDMP),SMP,3,3)                                 000358
C                                                                      000359
      IQ = LOCU(NOBQ) - 1                                              000360
      IP = LOCU(NOBP) - 1                                              000361
      DO 25 I=1,3                                                      000362
      WQN(I) = -Y(IQ+I)                                                000363
   25 WPM(I) =  Y(IP+I)                                                000364
      IF (LEQ .EQ. 0) GO TO 26                                         000365
      DO 27 I=1,3                                                      000366
   27 WQN(I) = WQN(I) - SXDQN(I)                                       000367
   26 IF (LEP .EQ. 0) GO TO 28                                         000368
      DO 29 I=1,3                                                      000369
   29 WPM(I) = WPM(I) + SXDPM(I)                                       000370
   28 CALL MULT3  (RPM,WPM,WPQP,3,3,1,3,1,1)                           000371
      CALL MULTAD (RPN,WQN,WPQP,3,3,1,3,1,1)                           000372
      CALL SKEWV3 (WPQP,WSK,1,3)                                       000373
      CALL MULT3  (WSK,RH(1,1,LRPQ),DRPQ,3,3,3,3,3,3)                  000374
      DO 30 I=1,3                                                      000375
      DO 30 J=1,3                                                      000376
   30 DRQP(I,J) = DRPQ(J,I)                                            000377
C                                                                      000378
      CALL MULT3  (DRPQ,RQN,BDTQ(4,4),3,3,3,3,3,6)                     000379
      CALL MULT3  (DRQP,RPM,DRQM,3,3,3,3,3,3)                          000380
      CALL MULT3  (PINDT,RQN,BDTQ,3,3,3,3,3,6)                         000381
      IF (LEQ .EQ. 0) GO TO 35                                         000382
      CALL MULT3  (RQN,SXDQN,VEC,3,3,1,3,1,1)                          000383
      CALL SKEWV3 (VEC,WSK,1,3)                                        000384
      CALL MULT3  (WSK,RQN,DRQN,3,3,3,3,3,3)                           000385
      CALL MULTAD (RH(1,1,LRPQ),DRQN,BDTQ(4,4),3,3,3,3,3,6)            000386
      CALL MULTAD (PIN(1,1,L),DRQN,BDTQ,3,3,3,3,3,6)                   000387
      CALL MULT3  (BDTQ(4,4),SNQ,BDTQ(4,1),3,3,3,6,3,6)                000388
      CALL SKEWV3 (HXDQN,WSK,1,3)                                      000389
```

```
      CALL MULTAD (RPN,WSK,BDTQ(4,1),3,3,3,3,3,6)                          000390
      CALL MULT3  (BDTQ,SIGH(I,1,LHSQ),BDTQ(1,7),3,3,LEQ,6,3,6)            000391
      CALL MULT3  (BDTQ(4,4),HATH(I,1,LHSQ),BDTQ(4,7),3,3,LEQ,6,3,6)       000392
      GO TO 40                                                             000393
   35 CALL MULT3  (BDTQ(4,4),SNQ,BDTQ(4,1),3,3,3,6,3,6)                    000394
   40 CONTINUE                                                             000395
C                                                                          000396
      IF (LEP .EQ. 0) GO TO 50                                             000397
      CALL MULT3  (RPM,SXDPM,VEC,3,3,1,3,1,1)                              000398
      CALL SKEWV3 (VEC,WSK,1,3)                                            000399
      CALL MULT3  (WSK,RPM,BDTP(4,4),3,3,3,3,3,6)                          000400
      CALL MULT3  (PINDT,RQM,BDTP(1,1),3,3,3,3,3,6)                        000401
      CALL MULT3  (BDTP(4,4),SMP,BDTP(4,1),3,3,3,6,3,6)                    000402
      CALL MULTAD (RQP,BDTP(4,4),DRQM,3,3,3,3,6,3)                         000403
      CALL MULTAD (PIN(1,1,L),DRQM,BDTP,3,3,3,3,3,6)                       000404
      CALL SKEWV3 (HXDPM,WSK,1,3)                                          000405
      CALL MULTAD (RPM,WSK,BDTP(4,1),3,3,3,3,3,6)                          000406
      DO 55 I=1,3                                                          000407
      IP3 = I + 3                                                          000408
      DO 55 J=1,3                                                          000409
      JP3 = J + 3                                                          000410
      BDTP(I,J) = -BDTP(I,J)                                               000411
      BDTP(IP3,J) = -BDTP(IP3,J)                                           000412
   55 BDTP(IP3,JP3) = -BDTP(IP3,JP3)                                       000413
      CALL MULT3  (BDTP(1,1),SIGH(1,1,LHSP),BDTP(1,7),3,3,LEP,6,3,6)       000414
      CALL MULT3  (BDTP(4,4),HATH(I,1,LHSP),BDTP(4,7),3,3,LEP,6,3,6)       000415
      GO TO 60                                                             000416
   50 CALL MULT3  (PINDT,RQM,BDTP,3,3,3,3,3,6)                             000417
      CALL MULTAD (PIN(1,1,L),DRQM,BDTP,3,3,3,3,3,6)                       000418
      DO 58 I=1,3                                                          000419
      DO 58 J=1,3                                                          000420
   58 BDTP(I,J) = -BDTP(I,J)                                               000421
   60 CONTINUE                                                             000422
      RETURN                                                               000423
C                                                                          000424
  100 DO 70 I=1,3                                                          000425
   70 WQN(I) = -Y(I)                                                       000426
      CALL SKEWV3 (WQN,WSK,1,3)                                            000427
      CALL MULT3  (ROL,WSK,BDTQ(4,4),3,3,3,3,3,6)                          000428
      B2DT = BETAHD(2,1)                                                   000429
      B3DT = BETAHD(3,1)                                                   000430
      DO 75 I=1,3                                                          000431
      DO 75 J=1,3                                                          000432
   75 BDTQ(I,J) = B2DT*RP2(I,J,1) + B3DT*RP3(I,J,1)                        000433
C                                                                          000434
      RETURN                                                               000435
      END                                                                  000436
```

```
[HDG,P     BHGENR                                                          -000437
[FOR,IS    BHGENR                                                          -000438
      COMPILER (XM=1),(EQUIV=CMN)                                          -000439
      SUBROUTINE BHGENR                                                     000440
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                   -000441
C                                                                           000442
            COMMON /BHBSRD/                                                 000443
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                  200444
            COMMON /HANDS /                                                 000445
     *      HATH(3, 6, 8),SIGH(3, 6, 8),HATS(3, 6,10),SIGS(3, 6,10)        400446
            COMMON /MAXMUM/                                                 000447
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU              000448
            COMMON /NUMBRS/                                                 000449
     *      ZRO,ONE,TWO,TRES                                               000450
            COMMON /PINRP /                                                 000451
     *      PIN(3,3, 5), RP2(3,3, 5), RP3(3,3, 5)                         1300452
            COMMON /SPECIF/                                                 000453
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),     1600454
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),            1700455
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1800456
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                           1900457
C                                                                           000458
      DIMENSION W1(3,3),W2(3,3)                                             000459
C                                                                           000460
      DATA I1ST /0 /                                                        000461
C                                                                           000462
      IF (I1ST .EQ. 1) GO TO 100                                            000463
      I1ST = 1                                                              000464
      LR = 2*NHMAX - 1                                                      000465
      JR = 6 + NMDBOD                                                       000466
      DO 5 L=1,LR                                                           000467
      DO 5 I=1,6                                                            000468
      DO 5 J=1,JR                                                           000469
    5 BH(I,J,L) = ZRO                                                       000470
C                                                                           000471
  100 DO 10 I=1,3                                                           000472
      IP3 = I + 3                                                           000473
      DO 10 J=1,3                                                           000474
      JP3 = J + 3                                                           000475
      BH(I,J,1) = PIN(I,J,1)                                                000476
   10 BH(IP3,JP3,1) = ROL(I,J,1)                                            000477
C                                                                           000478
      DO 20 L=2,NH                                                          000479
      LQ = 2*L - 2                                                          000480
      LP = LQ + 1                                                           000481
      LR3 = 6*(L-2) + 3                                                     000482
      LR4 = LR3 + 1                                                         000483
      LR5 = LR3 + 2                                                         000484
      LD3 = 7*(L-2) + 3                                                     000485
      LD4 = LD3 + 1                                                         000486
```

```
      DO 25 I=1,3                                                                  000487
      DO 25 J=1,3                                                                  000488
      W1(J,I) = RH(I,J,LR3)                                                        000489
      W2(J,I) = RH(I,J,LR5)                                                        000490
   25 BH(J+3,I+3,LP) = -RH(I,J,LR4)                                                000491
      CALL MULT3  (RH(1,1,LR5),W1,BH(4,4,LQ),3,3,3,3,3,6)                         000492
      CALL MULT3  (PIN(1,1,L),W1,BH(1,1,LQ),3,3,3,3,3,6)                          000493
      CALL MULT3  (W2,BH(4,4,LP),W1,3,3,3,3,6,3)                                  000494
      CALL MULT3  (PIN(1,1,L),W1,BH(1,1,LP),3,3,3,3,3,6)                          000495
      CALL SKEWV3 (DH(1,LD3),W1,3,3)                                              000496
      CALL SKEWV3 (DH(1,LD4),W2,3,3)                                              000497
      CALL MULT3  (BH(4,4,LQ),W1,BH(4,1,LQ),3,3,3,6,3,6)                          000498
      CALL MULT3  (BH(4,4,LP),W2,BH(4,1,LP),3,3,3,6,3,6)                          000499
      NOBQ = ITOPOL(1,L)                                                          000500
      NOBP = ITOPOL(2,L)                                                          000501
      NMQ = IRGFLX(NOBQ)                                                          000502
      NMP = IRGFLX(NOBP)                                                          000503
      IF (NMQ .EQ. 0) GO TO 30                                                    000504
      LHS = 2*L - 3                                                               000505
      CALL MULT3  (BH(1,1,LQ),SIGH(1,1,LHS),BH(1,7,LQ),3,3,NMQ,6,3,6)            000506
      CALL MULT3  (BH(4,4,LQ),HATH(1,1,LHS),BH(4,7,LQ),3,3,NMQ,6,3,6)            000507
   30 IF (NMP .EQ. 0) GO TO 20                                                    000508
      LHS = 2*L - 2                                                               000509
      CALL MULT3  (BH(1,1,LP),SIGH(1,1,LHS),BH(1,7,LP),3,3,NMP,6,3,6)            000510
      CALL MULT3  (BH(4,4,LP),HATH(1,1,LHS),BH(4,7,LP),3,3,NMP,6,3,6)            000511
   20 CONTINUE                                                                    000512
C                                                                                 000513
      RETURN                                                                      000514
      END                                                                         000515
```

```
[HDG,P    BSGENR                                                          -000516
[FOR,IS   BSGENR                                                          -000517
      COMPILER (XM=1),(EQUIV=CMN)                                         -000518
      SUBROUTINE BSGENR                                                    000519
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                  -000520
C                                                                          000521
            COMMON /BHBSRD/                                                000522
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                 200523
            COMMON /HANDS /                                                000524
     *      HATH(3, 6, 8),SIGH(3, 6, 8),HATS(3, 6,10),SIGS(3, 6,10)       400525
            COMMON /MAXMUM/                                                000526
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU             000527
            COMMON /NUMBRS/                                                000528
     *      ZRO,ONE,TWO,TRES                                              000529
            COMMON /SPECIF/                                                000530
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),     1600531
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),            1700532
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1800533
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                           1900534
C                                                                          000535
      DIMENSION W(3,3)                                                     000536
C                                                                          000537
      DATA I1ST / 0 /                                                      000538
C                                                                          000539
      IF (I1ST .EQ. 1) GO TO 20                                            000540
C                                                                          000541
      JR = 6 + NMDBOD                                                      000542
      DO 5 L=1,NSPT                                                        000543
      DO 5 I=1,6                                                           000544
      DO 5 J=1,JR                                                          000545
    5 BS(I,J,L) = ZRO                                                      000546
C                                                                          000547
   20 DO 10 L=1,NSPT                                                       000548
      NOB = IFTSMW(L)                                                      000549
      LF = IRGFLX(NOB)                                                     000550
      IF (LF .EQ. 0  .AND. I1ST .EQ. 1) GO TO 10                           000551
      LR2 = 2*L                                                            000552
      DO 15 I=1,3                                                          000553
      IP3 = I + 3                                                          000554
      DO 15 J=1,3                                                          000555
      JP3 = J + 3                                                          000556
      BS(J,I,L) = RS(I,J,LR2)                                              000557
   15 BS(JP3,IP3,L) = RS(I,J,LR2)                                          000558
      CALL SKEWV3 (DS(1,LR2),W,3,3)                                        000559
      CALL MULT3  (BS(1,1,L),W,BS(4,1,L),3,3,3,6,3,6)                      000560
      IF (LF .EQ. 0) GO TO 10                                              000561
      CALL MULT3  (BS(1,1,L),SIGS(1,1,L),BS(1,7,L),3,3,LF,6,3,6)           000562
      CALL MULT3  (BS(1,1,L),HATS(1,1,L),BS(4,7,L),3,3,LF,6,3,6)           000563
   10 CONTINUE                                                             000564
C                                                                          000565
```

```
      I1ST = 1                                                    000566
      RETURN                                                      000567
      END                                                         000568
```

```
[HDG,P    BTABA                                                          -000569
[FOR,IS   BTABA                                                          -000570
      COMPILER (XM=1),(EQUIV=CMN)                                        -000571
      SUBROUTINE BTABA (AZ,B,NRB,NCB,KAZ,KB)                              000572
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -000573
      DIMENSION AZ(KAZ,1), B(KB,1),W(150)                               7200574
      DATA NOT / 6/                                                       000575
C                                                                         000576
C  TRIPLE MATRIX PRODUCT.  B(TRANSPOSE) * A * B = Z.                      000577
C  A MUST BE SYMMETRIC TO GET CORRECT ANSWER.                             000578
C  Z WILL BE SYMMETRIC. UPPER HALF CALCULATED, REFLECTED TO LOWER HALF.   000579
C  USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.                       000580
C  AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE     000581
C  LARGER OF A OR Z.                                                      000582
C  CALLS FORMA SUBROUTINE ZZBOMB.                                         000583
C  THE MAXIMUM SIZES ARE                                                  000584
C     NRB = XXX                                                           000585
C     NCB = XXX                                                           000586
C  DEVELOPED BY W A BENFIELD.  MAY 1972.                                  000587
C  LAST REVISION BY R A PHILIPPUS.  JUNE 1972.                            000588
C  MODIFIED FOR USE IN GSFC PROGRAM BY CARL BODLEY, MAY 1974              000589
C                                                                         000590
C     SUBROUTINE ARGUMENTS                                                000591
C  AZ  = INPUT  INNER MATRIX. SIZE(NRB,NRB).                              000592
C      = OUTPUT RESULT MATRIX. SIZE(NCB,NCB).                             000593
C  B   = INPUT  OUTER MATRIX. SIZE(NRB,NCB).                              000594
C  NRB = INPUT  NUMBER OF ROWS OF MATRIX B, SIZE OF MATRIX A. MAX=150.    000595
C  NCB = INPUT  NUMBER OF COLS OF MATRIX B, SIZE OF MATRIX Z. MAX=150.    000596
C  KAZ = INPUT  ROW DIMENSION OF AZ IN CALLING PROGRAM.                   000597
C  KB  = INPUT  ROW DIMENSION OF B IN CALLING PROGRAM.                    000598
C                                                                         000599
      IF (NRB.GT.150 .OR. NCB.GT.150 .OR. NRB.GT.KAZ .OR. NCB.GT.KAZ)    7300600
     *   GO TO 999                                                       000601
C                                                                         000602
      DO 20 I=1,NRB                                                       000603
      DO 5 K=1,NRB                                                        000604
    5 W(K) = AZ(I,K)                                                      000605
      DO 20 J=1,NCB                                                       000606
      S = 0.D 0                                                           000607
      DO 10 K=1,NPB                                                       000608
   10 S = S + W(K)*B(K,J)                                                 000609
   20 AZ(I,J) = S                                                         000610
C                                                                         000611
      DO 30 J=1,NCB                                                       000612
      DO 25 I=1,J                                                         000613
      W(I) = 0.D 0                                                        000614
      DO 25 K=1,NRB                                                       000615
   25 W(I) = W(I)+B(K,I)*AZ(K,J)                                          000616
      DO 30 I=1,J                                                         000617
      AZ(I,J) = W(I)                                                      000618
```

```
   30 AZ(J,I) = W(I)                                                    000619
      RETURN                                                            000620
C                                                                       000621
  999 WRITE (NOT,1001)                                                  000622
 1001 FORMAT (1H1,31HERROR IN BTABA, PROGRAM STOPPED)                   000623
      STOP                                                              000624
      END                                                               000625
```

```
[HDG,P    CANCOR                                                        -000626
[FOR,IS   CANCOR                                                        -000627
      COMPILER (XM=1),(EQUIV=CMN)                                       -000628
      SUBROUTINE CANCOR (R)                                              000629
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -000630
C THIS ROUTINE CANCELS OUT THE ZERO, REAL, AND THE COMPLEX ROOTS THAT AR  000631
C COMMON TO THE NUMERATOR AND DENOMINATOR OF THE TRANSFER FUNCTION R.     000632
C                                                                         000633
C --- R(1) = NUMBER OF REAL ROOTS IN THE NUMERATOR                       000634
C --- R(2) = NUMBER OF COMPLEX PAIRS IN THE NUMERATOR                    000635
C --- R(3) = NUMBER OF ZERO ROOTS IN THE NUMERATOR                       000636
C --- R(4) = NUMBER OF REAL ROOTS IN THE DENOMINATOR                     000637
C --- R(5) = NUMBER OF COMPLEX PAIRS IN THE DENOMINATOR                  000638
C --- R(6) = NUMBER OF ZERO ROOTS IN THE DENOMINATOR                     000639
C --- R(7) = GAIN                                                        000640
C --- R(8)...R(I) = NUMERATOR REAL ROOTS ARRAY                           000641
C --- R(I+1)...R(J) = NUMERATOR COMPLEX PAIRS ARRAY                      000642
C --- R(J+1)...R(K) = DENOMINATOR REAL ROOTS ARRAY                       000643
C --- R(K+1...R(L) = DENOMINATOR COMPLEX PAIRS ARRAY                     000644
C                                                                         000645
      DIMENSION R(1)                                                    000646
      NR=R(1)+.0001D0                                                   000647
      NCP=R(2)+.0001D0                                                  000648
      MR=R(4)+.0001D0                                                   000649
      MCP=R(5)+.0001D0                                                  000650
      KK=7+NR+MR+2*(NCP+MCP)                                            000651
      N = 7 + NR                                                        000652
      IF((NR.EQ.0).OR.(MR.EQ.0)) GO TO 160                              000653
      J=8+NR+2*NCP                                                      000654
      K=J-1+MR                                                          000655
      NN=0                                                              000656
      DO 140 I=8,N                                                      000657
      II=I-NN                                                           000658
      DO 100 JJ=J,K                                                     000659
      JJJ=JJ                                                            000660
      IF(DABS(R(II)/R(JJJ)-1.D0).LT.1.0D-7) GO TO 110                   000661
  100 CONTINUE                                                          000662
      GO TO 140                                                         000663
  110 DO 130 L=II,KK                                                    000664
      IF(L.GE.(JJJ-1)) GO TO 120                                        000665
      R(L)=R(L+1)                                                       000666
      GO TO 130                                                         000667
  120 R(L)=R(L+2)                                                       000668
  130 CONTINUE                                                          000669
      KK=KK-2                                                           000670
      J=J-1                                                             000671
      K=K-2                                                             000672
      NN=NN+1                                                           000673
      IF((NN.EQ.NR).OR.(NN.EQ.MR)) GO TO 150                            000674
  140 CONTINUE                                                          000675
```

```
150 R(1)=R(1)-DFLOAT(NN)                                                    000676
    R(4)=R(4)-DFLOAT(NN)                                                    000677
    NR=NR-NN                                                                000678
    MR=MR-NN                                                                000679
160 IF((NCP.EQ.0).OR.(MCP.EQ.0)) GO TO 230                                  000680
    NNN=8+NR                                                                000681
    N=NNN-1+2*NCP                                                           000682
    J=N+1+MR                                                                000683
    K=J-1+2*MCP                                                             000684
    NN=0                                                                    000685
    DO 210 I=NNN,N,2                                                        000686
    II=I-2*NN                                                               000687
    DO 170 JJ=J,K,2                                                         000688
    JJJ=JJ                                                                  000689
    IF((R(II).EQ.R(JJJ)).AND.(R(II+1).EQ.R(JJJ+1))) GO TO 180               000690
170 CONTINUE                                                                000691
    GO TO 210                                                               000692
180 DO 200 L=II,KK                                                          000693
    IF(L.GE.(JJJ-2)) GO TO 190                                              000694
    R(L)=R(L+2)                                                             000695
    GO TO 200                                                               000696
190 R(L)=R(L+4)                                                             000697
200 CONTINUE                                                                000698
    KK=KK-4                                                                 000699
    J=J-2                                                                   000700
    K=K-4                                                                   000701
    NN=NN+1                                                                 000702
    IF((NN.EQ.NCP).OR.(NN.EQ.MCP)) GO TO 220                                000703
210 CONTINUE                                                                000704
220 R(2)=R(2)-DFLOAT(NN)                                                    000705
    R(5)=R(5)-DFLOAT(NN)                                                    000706
230 X=DMIN1(R(3),R(6))                                                      000707
    R(3)=R(3)-X                                                             000708
    R(6)=R(6)-X                                                             000709
    RETURN                                                                  000710
    END                                                                     000711
```

```
[HDG,P    COMENT                                                        -000712
[FOR,IS   COMENT                                                        -000713
      COMPILER (XM=1),(EQUIV=CMN)                                       -000714
      SUBROUTINE COMENT                                                  000715
      DIMENSION  IREMRK(13)                                              000716
      DATA NIT,NOT /5,6/                                                -000717
C                                                                        000718
C   READ COMMENT CARDS AND PRINT THEM UNDER PAGE HEADING OF FORMA        000719
C   SUBROUTINE PAGEHD. COMMENT CARDS MAY HAVE ANY KEYPUNCH SYMBOL        000720
C   IN CARD COLUMNS 1-78.                                                000721
C   IF IT IS DESIRED TO HAVE ANY GIVEN COMMENT CARD PRINT ON A NEW       000722
C   PAGE, SUPPLY THE LETTER P IN COLUMN 80 ON THAT CARD.                 000723
C   ROUTINE IS ENDED BY SUPPLYING A CARD WITH ZEROS IN COLUMNS 1 THRU 10. 000724
C   CALLS FORMA SUBROUTINE PAGEHD.                                       000725
C   CODED BY  RF HRUDA.  MARCH 1966.                                     000726
C   LAST MODIFICATION BY  J ERNST.  JUNE 1971.                           000727
C                                                                        000728
 1001 FORMAT  (13A6,1X,A1)                                               000729
 2001 FORMAT (///)                                                       000730
 2002 FORMAT  (22X,13A6)                                                 000731
C                                                                        000732
      N = 0                                                              000733
    1 READ (NIT,1001) (IREMRK(I),I=1,13),IPGHD                           000734
      IF (IREMRK(1) .EQ. 6H000000) RETURN                               -000735
      N = N+1                                                            000736
      IF (N.NE.1 .AND. IPGHD.NE.1HP) GO TO 2                            -000737
      CALL PAGEHD                                                        000738
      WRITE (NOT,2001)                                                   000739
      N = 1                                                              000740
    2 IF (N .EQ. 50)  N = 0                                              000741
      WRITE (NOT,2002) (IREMRK(I),I=1,13)                                000742
      GO TO 1                                                            000743
      END                                                                000744
```

```
[HDG,P    CONTRL                                                          -000745
[FOR,IS   CONTRL                                                          -000746
      COMPILER (XM=1),(EQUIV=CMN)                                         -000747
      SUBROUTINE CONTRL                                                    000748
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                 -000749
C                                                                          000750
            COMMON /BHBSRD/                                              000751
     *     BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                  200752
            COMMON /CONPAR/                                              000753
     *     CNTDTA(100)                                                   9500754
      COMMON /LDSIZE/ NX,NY,NDLTA,NXSS,NBTQ,NJQ,NY2,ND2                   000755
            COMMON /SPECIF/                                              000756
     *     BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),     1600757
     *     DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),            1700758
     *     NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1800759
     *     LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                           1900760
            COMMON /TIMESS/                                              000761
     *     STARTT,DELTAT,T,ENDT,TMST                                     000762
            COMMON /VECTOR/                                              000763
     *     Y(250),YDT(250)                                               2000764
CCCCCCC  THIS COMMON IS TRANSFER BETWEEN CONTRL AND SHAFTT ONLY ----      000765
           COMMON /WHEEL /                                               000766
     *    CLM(4)                                                         000767
C                                                                          000768
      DIMENSION  TQ(6),TQD(6),RHD(3),THADW(3)                            000769
      DIMENSION CPLY(10,4), KPLY(2), UI(2)                               000770
      DATA ICT4/0/, RHD / 0.D0, 0.D0, 0.D0 /                             000771
      DATA T1,T2,T3,T4,DTHE/                                             000772
     *    .2D0, 1.2D0, .7D0, 1.7D0, 1.0471975D0 /                        000773
      DATA NPLY, KRY, KCY/ 0, 10, 4/                                     000774
      DATA I1ST/ 0 /                                                     000775
      ALIM(U,V) = DMAX1(-V,DMIN1(U,V))                                   000776
C                                                                          000777
CCCCCCCCCC                                                             000778
CCCCCCCCCC                                                             000779
CCC  THE FOLLOWING STATEMENTS MUST ALWAYS BE IN CONTPL..                   000780
      IF (I1ST .NE. 0) GO TO 110                                         000781
      I1ST = 1                                                           000782
      IF (NPLY .EQ. 0) GO TO 106                                         000783
      CALL ZERO (CPLY,KRY,KCY,KRY)                                       000784
      DO 105 K=1,NPLY                                                    000785
           K2=2*K-1                                                      000786
  105 CALL READ (CPLY(1,K2),KPLY(K),N2,KRY,KCY)                          000787
      CALL WRITE (CPLY,KRY,KCY,4HCPLY,KRY)                               000788
  106 CONTINUE                                                           000789
      NDLTA = NDELTA                                                     000790
      LDEL = LOCU(2*NB+2) - 1                                            000791
  110 CONTINUE                                                           000792
      NXSS = 3                                                           000793
      NBTQ = 3                                                           000794
```

```
      IF (NDELTA .EQ. 0) RETURN                                          000795
CCCCCCCCCCC    CCC                                                       000796
CCCC----NOTE--THIS SUBROUTINE MUST ESTABLISH  NDLTA,NXSS AND NBTQ        000797
CCCCCCCCCCC                                                              000798
C                                                                        000799
CCCC  ESTABLISH THE D/DT(DELTAS)                                         000800
C                                                                        000801
CCCCCCCCCC                                                               000802
CCCC---NOTE--THIS SECTION IS TYPICAL OF USE OF TFPLY.                    000803
CCCCCCCCCC                                                               000804
C                                                                        000805
      IF (NPLY .EQ. 0) GO TO 116                                         000806
      L = LDEL+1                                                         000807
      DO 115 K=1,NPLY                                                    000808
         K2 = 2*K-1                                                      000809
      CALL TFPLY (CPLY(1,K2),CPLY(1,K2+1),UI(K),X,KPLY(K),L)             000810
      L = L+KPLY(K) - 1                                                  000811
  115 CONTINUE                                                           000812
  116 CONTINUE                                                           000813
C                                                                        000814
CCCCCCCCCC                                                               000815
      ICT4 = ICT4 + 1                                                    000816
      IA = (ICT4-1)/4                                                    000817
      IAA = (ICT4-2)/4                                                   000818
      IFLAG = IA - IAA                                                   000819
      DO 6 I=1,3                                                         000820
    6 THADW(I) = Y(6+I)                                                  000821
      DO 5 I=1,6                                                         000822
    5 TQ(I) = Y(LDEL+I)                                                  000823
C                                                                        000824
C          WHEEL 1 (ROLL INERTIA WHEEL CONTROL TORQUE)                   000825
C          DEFINE DIFFERENTIAL EQUATIONS FOR ROLL CONTROL LOOP           000826
C                                                                        000827
      U1 = 57.2958D0*ROL(3,2,1)/ROL(3,3,1)                               000828
      U5 = ALIM(TQ(5),29.D0)                                             000829
      U2 = 2.17D0*U1 - U5                                                000830
      U3 = ALIM(1.1D0*U2,1.17D0)                                         000831
      TQD(5) = (1.D0/88.D0)*(-TQ(5) + (9/1.1D0)*U3)                      000832
      U6 = ALIM(5*U3,1.68D0)                                             000833
      U8 = ALIM(TQ(6),1.9D0)                                             000834
      IF (IFLAG .EQ. 0) GO TO 32                                         000835
      UU = DABS(U8)                                                      000836
      IF (UU.GT.1.D0) GO TO 30                                           000837
      IF (UU.LT.0.5D0) GO TO 31                                          000838
      U9 = RHD(1)                                                        000839
      GO TO 10                                                           000840
   30 U9 = U8/UU                                                         000841
      GO TO 10                                                           000842
   31 U9 = 0.D0                                                          000843
      GO TO 10                                                           000844
```

```
   32 U9 = RHD(1)                                                    000845
      GO TO 33                                                       000846
   10 RHD(1) = U9                                                    000847
   33 CONTINUE                                                       000848
      TQD(6) = (-TQ(6) + 2.5D0*(U6-U9))/.5D0                         000849
C                                                                    000850
C  1500 RPM = 157.0795 RAD/SEC                                       000851
C   6 INCH*OZ = .03125 FT*LBS                                        000852
C                                                                    000853
      IF (DABS(THADW(1)).GT. 157.0795D0) U9 = 0.D0                   000854
      CLM(1) = .03125D0*U9 - 5.D-05*THADW(1)                         000855
C                                                                    000856
C          WHEEL 2 (PITCH INERTIA WHEEL CONTROL TORQUE)              000857
C          DEFINE DIFFERENTIAL EQUATIONS IN PITCH CONTROL LOOP       000858
C                                                                    000859
      U1 = -57.2958D0*ROL(3,1,1)/ROL(3,3,1)                          000860
      U5 = ALIM(TQ(1),16.4D0)                                        000861
      U2 = 2.17D0*U1 - U5                                            000862
      U3 = ALIM(.82D0*U2,1.17D0)                                     000863
      TQD(1) = (-TQ(1) + U3*(7/.82D0))/50.D0                         000864
      U6 = ALIM(5*U3,1.68D0)                                         000865
      U8 = ALIM(TQ(2),1.9D0)                                         000866
      IF (IFLAG.EQ.0) GO TO 14                                       000867
      UU = DABS(U8)                                                  000868
      IF (UU.GT. 1.D0) GO TO 15                                      000869
      IF (UU.LT.0.5D0) GO TO 16                                      000870
      U9 = RHD(2)                                                    000871
      GO TO 12                                                       000872
   15 U9 = U8/UU                                                     000873
      GO TO 12                                                       000874
   16 U9 = 0.D0                                                      000875
      GO TO 12                                                       000876
   14 U9 = RHD(2)                                                    000877
      GO TO 13                                                       000878
   12 RHD(2) = U9                                                    000879
   13 CONTINUE                                                       000880
      TQD(2) = (-TQ(2) + 2.5D0*(U6 - U9))/.5D0                       000881
      IF (DABS(THADW(2)).GE. 157.0795D0) U9 = 0                      000882
      CLM(2) = .03125D0*U9 - 5.D-05*THADW(2)                         000883
C                                                                    000884
C          WHEEL 3 (YAW INERTIA WHEEL CONTROL TORQUE)                000885
C          DEFINE DIFFERENTIAL EQUATIONS FOR YAW CONTROL LOOP        000886
C                                                                    000887
      U1 = 57.2958D0*ROL(2,1,1)/ROL(2,2,1)                           000888
      U2 = ALIM(U1,2.D0)                                             000889
      U6 = ALIM(TQ(3),29.D0)                                         000890
      U3 = 2.17D0*U2 - U6                                            000891
      U4 = ALIM(1.47D0*U3,1.17D0)                                    000892
      TQD(3) = (1.D0/88.D0)*(-TQ(3) + (9/1.47D0)*U4)                 000893
      U7 = ALIM(5*U4,1.68D0)                                         000894
```

```
      U9 = ALIM(TQ(4),1.9D0)                                          000895
      IF (IFLAG.EQ.0) GO TO 20                                        000896
      UU = DABS(U9)                                                   000897
      IF (UU.GT.1.D0) GO TO 21                                        000898
      IF (UU.LT. 0.5D0) GO TO 22                                      000899
      U10 = RHD(3)                                                    000900
      GO TO 18                                                        000901
   21 U10 = U9/UU                                                     000902
      GO TO 18                                                        000903
   22 U10 = 0.D0                                                      000904
      GO TO 18                                                        000905
   20 U10 = RHD(3)                                                    000906
      GO TO 24                                                        000907
   18 RHD(3) = U10                                                    000908
   24 CONTINUE                                                        000909
      TQD(4) = (-TQ(4) + 2.5D0*(U7 - U10))/.5D0                       000910
      IF (DABS(THADW(3)) .GT. 157.0795D0) U10 = 0.D0                  000911
      CLM(3) = .03125D0*U10 - 5.D-05*THADW(3)                         000912
C                                                                     000913
      DO 34 I=1,6                                                     000914
   34 YDT(LDEL+I) = TQD(I)                                            000915
      YDT(LDEL+7) = Y(16)                                             000916
      SK4 = CNTDTA(NDELTA+1)                                          000917
      DK4 = CNTDTA(NDELTA+2)                                          000918
      CLM(4) =-(SK4*Y(LDEL+7) + DK4*YDT(LDEL+7))                      000919
C                                                                     000920
      RETURN                                                          000921
      END                                                             000922
```

```
[HDG,P    CREA                                                          -000923
[FOR,IS   CREA                                                          -000924
      COMPILER (XM=1),(EQUIV=CMN)                                       -000925
      SUBROUTINE CREA(NREC3,NJ,NE,UVEC,A,B,KA,KB,KWS)                    000926
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -000927
C                                                                        000928
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES                                   000929
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3                               000930
C                                                                        000931
      DIMENSION A(KA,1),B(KB,1),UVEC(1)                                  000932
C                                                                        000933
      CALL ZERO(A,9,NE,KA)                                               000934
C***  FETCH M*HX                                                         000935
      CALL FETCH(NTAPE3, 1,NREC3,B,NJ,NE,KB)                             000936
      CALL SATB(-ONE,UVEC,B,A(6,1),NJ,1,NE,KWS,KB,KA)                    000937
C***  FETCH M*HY                                                         000938
      CALL FETCH(NTAPE3, 4,NREC3,B,NJ,NE,KB)                             000939
      CALL SATB( ONE,UVEC,B,A(3,1),NJ,1,NE,KWS,KB,KA)                    000940
C***  FETCH M*HZ                                                         000941
      CALL FETCH(NTAPE3, 7,NREC3,B,NJ,NE,KB)                             000942
      CALL SATB(-ONE,UVEC,B,A(2,1),NJ,1,NE,KWS,KB,KA)                    000943
      DO 49 J=1,NE                                                       000944
      A(4,J) = -A(2,J)                                                   000945
      A(7,J) = -A(3,J)                                                   000946
   49 A(8,J) = -A(6,J)                                                   000947
C                                                                        000948
      CALL WRITE(A,9,NE,4HACOF,KA)                                       000949
      WRITE(NTAPE1) ((A(I,J),J=1,NE),I=1,9)                              000950
C                                                                        000951
      RETURN                                                             000952
      END                                                                000953
```

```
[HDG,P    CREADO                                                                -000954
[FOR,IS   CREADO                                                                -000955
      COMPILER (XM=1),(EQUIV=CMN)                                               -000956
      SUBROUTINE CREADO(NREC3,NJ,NE,UVEC,A,B,C,WS,AMU,KA,KB,KC,KWS,KAMU)         000957
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                        -000958
C                                                                               000959
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES                                          000960
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3                                      000961
C                                                                               000962
      DIMENSION A(KA,1),B(KB,1),C(KC,1),WS(KWS,1),AMU(KAMU,1),UVEC(1)           000963
C                                                                               000964
C***  FETCH M*HX,M*HY,M*HZ                                                      000965
      CALL FETCH(NTAPE3, 1,NREC3,A,NJ,NE,KA)                                    000966
      CALL FETCH(NTAPE3, 4,NREC3,B,NJ,NE,KB)                                    000967
      CALL FETCH(NTAPE3, 7,NREC3,C,NJ,NE,KC)                                    000968
      CALL SATB( ONE,UVEC     ,A,AMU(4,7),NJ, 1,NE,KWS,KA,KAMU)                 000969
      CALL SATB( ONE,UVEC     ,B,AMU(5,7),NJ, 1,NE,KWS,KB,KAMU)                 000970
      CALL SATB( ONE,UVEC     ,C,AMU(6,7),NJ, 1,NE,KWS,KC,KAMU)                 000971
      CALL SATB( ONE,WS(1,13),A,AMU(2,7),NJ, 1,NE,KWS,KA,KAMU)                  000972
      CALL SATB(-ONE,WS(1,12),A,AMU(3,7),NJ, 1,NE,KWS,KA,KAMU)                  000973
      CALL SATB(-ONE,WS(1,13),B,AMU(1,7),NJ, 1,NE,KWS,KB,KAMU)                  000974
      CALL SATB( ONE,WS(1,11),B,AMU(3,7),NJ, 1,NE,KWS,KB,KAMU)                  000975
      CALL SATB( ONE,WS(1,12),C,AMU(1,7),NJ, 1,NE,KWS,KC,KAMU)                  000976
      CALL SATB(-ONE,WS(1,11),C,AMU(2,7),NJ, 1,NE,KWS,KC,KAMU)                  000977
C***  FETCH SY*SIGX,SZ*SIGX                                                     000978
      CALL FETCH(NTAPE3,13,NREC3,A,NJ,NE,KA)                                    000979
      CALL FETCH(NTAPE3,14,NREC3,B,NJ,NE,KB)                                    000980
      CALL SATB( ONE,UVEC     ,A,AMU(6,7),NJ, 1,NE,KWS,KA,KAMU)                 000981
      CALL SATB(-ONE,UVEC     ,B,AMU(5,7),NJ, 1,NE,KWS,KB,KAMU)                 000982
      CALL SATB( ONE,WS(1,12),A,AMU(1,7),NJ, 1,NE,KWS,KA,KAMU)                  000983
      CALL SATB( ONE,WS(1,13),B,AMU(1,7),NJ, 1,NE,KWS,KB,KAMU)                  000984
      CALL SATB(-ONE,WS(1,11),A,AMU(2,7),NJ, 1,NE,KWS,KA,KAMU)                  000985
      CALL SATB(-ONE,WS(1,11),B,AMU(3,7),NJ, 1,NE,KWS,KB,KAMU)                  000986
C***  FETCH SX*SIGY,SZ*SIGY                                                     000987
      CALL FETCH(NTAPE3,18,NREC3,A,NJ,NE,KA)                                    000988
      CALL FETCH(NTAPE3,19,NREC3,B,NJ,NE,KB)                                    000989
      CALL SATB(-ONE,UVEC     ,A,AMU(6,7),NJ, 1,NE,KWS,KA,KAMU)                 000990
      CALL SATB( ONE,UVEC     ,B,AMU(4,7),NJ, 1,NE,KWS,KB,KAMU)                 000991
      CALL SATB(-ONE,WS(1,12),A,AMU(1,7),NJ, 1,NE,KWS,KA,KAMU)                  000992
      CALL SATB( ONE,WS(1,13),B,AMU(2,7),NJ, 1,NE,KWS,KB,KAMU)                  000993
      CALL SATB( ONE,WS(1,11),A,AMU(2,7),NJ, 1,NE,KWS,KA,KAMU)                  000994
      CALL SATB(-ONE,WS(1,12),B,AMU(3,7),NJ, 1,NE,KWS,KB,KAMU)                  000995
C***  FETCH SX*SIGZ,SY*SIGZ                                                     000996
      CALL FETCH(NTAPE3,23,NREC3,A,NJ,NE,KA)                                    000997
      CALL FETCH(NTAPE3,24,NREC3,B,NJ,NE,KB)                                    000998
      CALL SATB( ONE,UVEC     ,A,AMU(5,7),NJ, 1,NE,KWS,KA,KAMU)                 000999
      CALL SATB(-ONE,UVEC     ,B,AMU(4,7),NJ, 1,NE,KWS,KB,KAMU)                 001000
      CALL SATB(-ONE,WS(1,13),A,AMU(1,7),NJ, 1,NE,KWS,KA,KAMU)                  001001
      CALL SATB(-ONE,WS(1,13),B,AMU(2,7),NJ, 1,NE,KWS,KB,KAMU)                  001002
      CALL SATB( ONE,WS(1,11),A,AMU(3,7),NJ, 1,NE,KWS,KA,KAMU)                  001003
```

```
      CALL SATB( ONE,WS(1,12),B,AMU(3,7),NJ, 1,NE,KWS,KB,KAMU)          001004
C***  FETCH SY*HX,SZ*HX                                                 001005
      CALL FETCH(NTAPE3, 2,NREC3,A,NJ,NE,KA)                            001006
      CALL FETCH(NTAPE3, 3,NREC3,B,NJ,NE,KB)                            001007
      CALL SATB( ONE,UVEC     ,B,AMU(2,7),NJ, 1,NE,KWS,KB,KAMU)         001008
      CALL SATB(-ONE,UVEC     ,A,AMU(3,7),NJ, 1,NE,KWS,KA,KAMU)         001009
C***  FETCH SX*HY,SZ*HY                                                 001010
      CALL FETCH(NTAPE3, 5,NREC3,A,NJ,NE,KA)                            001011
      CALL FETCH(NTAPE3, 6,NREC3,B,NJ,NE,KB)                            001012
      CALL SATB(-ONE,UVEC     ,B,AMU(1,7),NJ, 1,NE,KWS,KB,KAMU)         001013
      CALL SATB( ONE,UVEC     ,A,AMU(3,7),NJ, 1,NE,KWS,KA,KAMU)         001014
C***  FETCH SX*HZ,SY*HZ                                                 001015
      CALL FETCH(NTAPE3, 8,NREC3,A,NJ,NE,KA)                            001016
      CALL FETCH(NTAPE3, 9,NREC3,B,NJ,NE,KB)                            001017
      CALL SATB( ONE,UVEC     ,B,AMU(1,7),NJ, 1,NE,KWS,KB,KAMU)         001018
      CALL SATB(-ONE,UVEC     ,A,AMU(2,7),NJ, 1,NE,KWS,KA,KAMU)         001019
C***  FETCH JXX*SIGX,JXY*SIGX,JXZ*SIGX                                  001020
      CALL FETCH(NTAPE3,10,NREC3,A,NJ,NE,KA)                            001021
      CALL FETCH(NTAPE3,11,NREC3,B,NJ,NE,KB)                            001022
      CALL FETCH(NTAPE3,12,NREC3,C,NJ,NE,KC)                            001023
      CALL SATB( ONE,UVEC     ,A,AMU(1,7),NJ, 1,NE,KWS,KA,KAMU)         001024
      CALL SATB(-ONE,UVEC     ,B,AMU(2,7),NJ, 1,NE,KWS,KB,KAMU)         001025
      CALL SATB(-ONE,UVEC     ,C,AMU(3,7),NJ, 1,NE,KWS,KC,KAMU)         001026
C***  FETCH JXY*SIGY,JYY*SIGY,JYZ*SIGY                                  001027
      CALL FETCH(NTAPE3,15,NREC3,A,NJ,NE,KA)                            001028
      CALL FETCH(NTAPE3,16,NREC3,B,NJ,NE,KB)                            001029
      CALL FETCH(NTAPE3,17,NREC3,C,NJ,NE,KC)                            001030
      CALL SATB(-ONE,UVEC     ,A,AMU(1,7),NJ, 1,NE,KWS,KA,KAMU)         001031
      CALL SATB( ONE,UVEC     ,B,AMU(2,7),NJ, 1,NE,KWS,KB,KAMU)         001032
      CALL SATB(-ONE,UVEC     ,C,AMU(3,7),NJ, 1,NE,KWS,KC,KAMU)         001033
C***  FETCH JXZ*SIGZ,JYZ*SIGZ,JZZ*SIGZ                                  001034
      CALL FETCH(NTAPE3,20,NREC3,A,NJ,NE,KA)                            001035
      CALL FETCH(NTAPE3,21,NREC3,B,NJ,NE,KB)                            001036
      CALL FETCH(NTAPE3,22,NREC3,C,NJ,NE,KC)                            001037
      CALL SATB(-ONE,UVEC     ,A,AMU(1,7),NJ, 1,NE,KWS,KA,KAMU)         001038
      CALL SATB(-ONE,UVEC     ,B,AMU(2,7),NJ, 1,NE,KWS,KB,KAMU)         001039
      CALL SATB( ONE,UVEC     ,C,AMU(3,7),NJ, 1,NE,KWS,KC,KAMU)         001040
C                                                                       001041
      CALL WRITE(AMU(1,7),3,NE,6HDOCOEF,KAMU)                           001042
      CALL WRITE(AMU(4,7),3,NE,6HAOCOEF,KAMU)                           001043
C                                                                       001044
      RETURN                                                            001045
      END                                                               001046
```

```
[HDG,P    CREB                                                          -001047
[FOR,IS   CREB                                                          -001048
      COMPILER (XM=1),(EQUIV=CMN)                                       -001049
      SUBROUTINE CREB(NREC3,NREC2,NJ,NE,A,B,WS,KA,KB,KWS)                001050
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -001051
C                                                                        001052
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES                                   001053
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3                               001054
C                                                                        001055
      DIMENSION A(KA,1),B(KB,1),WS(KWS,1)                               001056
C                                                                        001057
      CALL ZERO(B,6,NE,KB)                                              001058
C***  FETCH M*RHOX,M*RHOY,M*RHOZ                                        001059
      CALL FETCH(NTAPE3,25,NREC3,A(1,1),NJ,1,KA)                        001060
      CALL FETCH(NTAPE3,26,NREC3,A(1,2),NJ,1,KA)                        001061
      CALL FETCH(NTAPE3,27,NREC3,A(1,3),NJ,1,KA)                        001062
      DO 52 I=1,NJ                                                      001063
      WS(I, 8) = WS(I, 8) + A(I,1)                                      001064
      WS(I, 9) = WS(I, 9) + A(I,2)                                      001065
   52 WS(I,10) = WS(I,10) + A(I,3)                                      001066
C***  FETCH HX                                                          001067
      CALL FETCH(NTAPE2, 5,NREC2,A,NJ,NE,KA)                            001068
      CALL SATB(ONE,WS(1, 8),A,B(2,1),NJ,1,NE,KWS,KA,KB)                001069
      CALL SATB(ONE,WS(1, 8),A,B(3,1),NJ,1,NE,KWS,KA,KB)                001070
      CALL SATB(ONE,WS(1, 9),A,B(4,1),NJ,1,NE,KWS,KA,KB)                001071
      CALL SATB(ONE,WS(1,10),A,B(5,1),NJ,1,NE,KWS,KA,KB)                001072
C***  FETCH HY                                                          001073
      CALL FETCH(NTAPE2, 6,NREC2,A,NJ,NE,KA)                            001074
      CALL SATB(ONE,WS(1, 9),A,B(1,1),NJ,1,NE,KWS,KA,KB)                001075
      CALL SATB(ONE,WS(1, 9),A,B(3,1),NJ,1,NE,KWS,KA,KB)                001076
      CALL SATB(ONE,WS(1, 8),A,B(4,1),NJ,1,NE,KWS,KA,KB)                001077
      CALL SATB(ONE,WS(1,10),A,B(6,1),NJ,1,NE,KWS,KA,KB)                001078
C***  FETCH HZ                                                          001079
      CALL FETCH(NTAPE2, 7,NREC2,A,NJ,NE,KA)                            001080
      CALL SATB(ONE,WS(1,10),A,B(1,1),NJ,1,NE,KWS,KA,KB)                001081
      CALL SATB(ONE,WS(1,10),A,B(2,1),NJ,1,NE,KWS,KA,KB)                001082
      CALL SATB(ONE,WS(1, 8),A,B(5,1),NJ,1,NE,KWS,KA,KB)                001083
      CALL SATB(ONE,WS(1, 9),A,B(6,1),NJ,1,NE,KWS,KA,KB)                001084
C                                                                        001085
      CALL WRITE(B,6,NE,4HBCOF,KB)                                      001086
      WRITE(NTAPE1) ((B(I,J),J=1,NE),I=1,6)                             001087
C                                                                        001088
      RETURN                                                            001089
      END                                                               001090
```

```
[HDG,P     CREC                                                          -001091
[FOR,IS    CREC                                                          -001092
      COMPILER (XM=1),(EQUIV=CMN)                                        -001093
      SUBROUTINE CREC(NREC3,NREC2,NJ,NE,A,B,C,AMU,KA,KB,KC,KAMU)          001094
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -001095
C                                                                         001096
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES                                    001097
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3                                001098
C                                                                         001099
      DIMENSION A(KA,1),B(KB,1),C(KC,1),AMU(KAMU,1)                       001100
C                                                                         001101
C***  FORM CCOF1 = CXY                                                    001102
      CALL ZERO(AMU,NE,NE,KAMU)                                           001103
C***  FETCH M*HY,SX*SIGZ,SZ*SIGX                                          001104
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)                              001105
      CALL FETCH(NTAPE3,23,NREC3,B,NJ,NE,KB)                              001106
      CALL FETCH(NTAPE3,14,NREC3,C,NJ,NE,KC)                              001107
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)                            001108
C***  FETCH HX                                                            001109
      CALL FETCH(NTAPE2, 5,NREC2,B,NJ,NE,KB)                              001110
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                         001111
C***  FETCH M*HX,SZ*SIGY,SY*SIGZ                                          001112
      CALL FETCH(NTAPE3, 1,NREC3,A,NJ,NE,KA)                              001113
      CALL FETCH(NTAPE3,19,NREC3,B,NJ,NE,KB)                              001114
      CALL FETCH(NTAPE3,24,NREC3,C,NJ,NE,KC)                              001115
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)                            001116
C***  FETCH HY                                                            001117
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)                              001118
      CALL SATB(-ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                         001119
C                                                                         001120
      CALL WRITE(AMU,NE,NE,3HCXY,KAMU)                                    001121
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                            001122
C                                                                         001123
C***  FORM CCOF2 = CXZ                                                    001124
      CALL ZERO(AMU,NE,NE,KAMU)                                           001125
C***  FETCH HZ                                                            001126
      CALL FETCH(NTAPE2, 7,NREC2,B,NJ,NE,KB)                              001127
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                         001128
C***  FETCH M*HZ,SY*SIGX,SX*SIGY                                          001129
      CALL FETCH(NTAPE3, 7,NREC3,A,NJ,NE,KA)                              001130
      CALL FETCH(NTAPE3,13,NREC3,B,NJ,NE,KB)                              001131
      CALL FETCH(NTAPE3,18,NREC3,C,NJ,NE,KC)                              001132
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)                            001133
C***  FETCH HX                                                            001134
      CALL FETCH(NTAPE2, 5,NREC2,B,NJ,NE,KB)                              001135
      CALL SATB(-ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                         001136
C                                                                         001137
      CALL WRITE(AMU,NE,NE,3HCXZ,KAMU)                                    001138
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                            001139
C                                                                         001140
```

```
C***  FORM CCOF3 = CYZ                                                  001141
      CALL ZERO(AMU,NE,NE,KAMU)                                         001142
C***  FETCH HY                                                          001143
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)                            001144
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001145
C***  FETCH M*HY,SX*SIGZ,SZ*SIGX                                        001146
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)                            001147
      CALL FETCH(NTAPE3,23,NREC3,B,NJ,NE,KB)                            001148
      CALL FETCH(NTAPE3,14,NREC3,C,NJ,NE,KC)                            001149
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)                          001150
C***  FETCH HZ                                                          001151
      CALL FETCH(NTAPE2, 7,NREC2,B,NJ,NE,KB)                            001152
      CALL SATB(-ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001153
C                                                                       001154
      CALL WRITE(AMU,NE,NE,3HCYZ,KAMU)                                  001155
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                          001156
C                                                                       001157
C***  FORM CCOF4 = C11                                                  001158
      CALL ZERO(AMU,NE,NE,KAMU)                                         001159
C***  FETCH M*HY AND HY                                                 001160
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)                            001161
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)                            001162
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001163
C***  FETCH M*HZ AND HZ                                                 001164
      CALL FETCH(NTAPE3, 7,NREC3,A,NJ,NE,KA)                            001165
      CALL FETCH(NTAPE2, 7,NREC2,B,NJ,NE,KB)                            001166
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001167
C                                                                       001168
      CALL WRITE(AMU,NE,NE,3HC11,KAMU)                                  001169
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                          001170
C                                                                       001171
C***  FORM CCOF5 = C22                                                  001172
      CALL ZERO(AMU,NE,NE,KAMU)                                         001173
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001174
C***  FETCH M*HX AND HX                                                 001175
      CALL FETCH(NTAPE3, 1,NREC3,A,NJ,NE,KA)                            001176
      CALL FETCH(NTAPE2, 5,NREC2,B,NJ,NE,KB)                            001177
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001178
C                                                                       001179
      CALL WRITE(AMU,NE,NE,3HC22,KAMU)                                  001180
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                          001181
C                                                                       001182
C***  FORM CCOF6 = C33                                                  001183
      CALL ZERO(AMU,NE,NE,KAMU)                                         001184
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001185
C***  FETCH M*HY AND HY                                                 001186
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)                            001187
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)                            001188
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001189
C                                                                       001190
```

```
      CALL WRITE(AMU,NE,NE,3HC33,KAMU)                                  001191
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                          001192
C                                                                       001193
C***  FORM CCOF7 = C12                                                  001194
      CALL ZERO(AMU,NE,NE,KAMU)                                         001195
C***  FETCH M*HX                                                        001196
      CALL FETCH(NTAPE3, 1,NREC3,A,NJ,NE,KA)                            001197
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001198
C                                                                       001199
      CALL WRITE(AMU,NE,NE,3HC12,KAMU)                                  001200
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                          001201
C                                                                       001202
C***  FORM CCOF8 = C13                                                  001203
      CALL ZERO(AMU,NE,NE,KAMU)                                         001204
C***  FETCH HZ                                                          001205
      CALL FETCH(NTAPE2, 7,NREC2,B,NJ,NE,KB)                            001206
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001207
C                                                                       001208
      CALL WRITE(AMU,NE,NE,3HC13,KAMU)                                  001209
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                          001210
C                                                                       001211
C***  FORM CCOF9 = C23                                                  001212
      CALL ZERO(AMU,NE,NE,KAMU)                                         001213
C***  FETCH M*HY                                                        001214
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)                            001215
      CALL SATB( ONE,B,A,AMU,NJ,NE,NE,KB,KA,KAMU)                       001216
C                                                                       001217
      CALL WRITE(AMU,NE,NE,3HC23,KAMU)                                  001218
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                          001219
C                                                                       001220
      RETURN                                                            001221
      END                                                               001222
```

```
[HDG,P    CREE                                                          -001223
[FOR,IS   CREE                                                          -001224
      COMPILER (XM=1),(EQUIV=CMN)                                       -001225
      SUBROUTINE CREE(NREC3,NREC2,NJ,NE,A,B,C,AMU,KA,KB,KC,KAMU)         001226
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -001227
C                                                                         001228
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES                                    001229
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3                              001230
C                                                                         001231
      DIMENSION A(KA,1),B(KB,1),C(KC,1),AMU(KAMU,1)                     001232
C                                                                         001233
C***  FETCH M*HX,SZ*SIGY,SY*SIGZ                                          001234
      CALL FETCH(NTAPE3, 1,NREC3,A,NJ,NE,KA)                            001235
      CALL FETCH(NTAPE3,19,NREC3,B,NJ,NE,KB)                            001236
      CALL FETCH(NTAPE3,24,NREC3,C,NJ,NE,KC)                            001237
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)                          001238
C***  FETCH HX                                                            001239
      CALL FETCH(NTAPE2, 5,NREC2,B,NJ,NE,KB)                            001240
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMU)                  001241
C***  FETCH M*HY,SX*SIGZ,SZ*SIGX                                          001242
      CALL FETCH(NTAPE3, 4,NREC3,A,NJ,NE,KA)                            001243
      CALL FETCH(NTAPE3,23,NREC3,B,NJ,NE,KB)                            001244
      CALL FETCH(NTAPE3,14,NREC3,C,NJ,NE,KC)                            001245
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)                          001246
C***  FETCH HY                                                            001247
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)                            001248
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMU)                  001249
C***  FETCH M*HZ,SY*SIGX,SX*SIGY                                          001250
      CALL FETCH(NTAPE3, 7,NREC3,A,NJ,NE,KA)                            001251
      CALL FETCH(NTAPE3,13,NREC3,B,NJ,NE,KB)                            001252
      CALL FETCH(NTAPE3,18,NREC3,C,NJ,NE,KC)                            001253
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)                          001254
C***  FETCH HZ                                                            001255
      CALL FETCH(NTAPE2, 7,NREC2,B,NJ,NE,KB)                            001256
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMU)                  001257
C***  FETCH JXX*SIGX,JXY*SIGY,JXZ*SIGZ                                    001258
      CALL FETCH(NTAPE3,10,NREC3,A,NJ,NE,KA)                            001259
      CALL FETCH(NTAPE3,15,NREC3,B,NJ,NE,KB)                            001260
      CALL FETCH(NTAPE3,20,NREC3,C,NJ,NE,KC)                            001261
      CALL ADD3( ONE,A,-ONE,B,-ONE,C,NJ,NE,KA)                          001262
C***  FETCH SY*HZ,SZ*HY                                                   001263
      CALL FETCH(NTAPE3, 9,NREC3,B,NJ,NE,KB)                            001264
      CALL FETCH(NTAPE3, 6,NREC3,C,NJ,NE,KC)                            001265
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)                          001266
C***  FETCH SIGX                                                          001267
      CALL FETCH(NTAPE2, 8,NREC2,B,NJ,NE,KB)                            001268
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMU)                  001269
C***  FETCH JXY*SIGX,JYY*SIGY,JYZ*SIGZ                                    001270
      CALL FETCH(NTAPE3,11,NREC3,A,NJ,NE,KA)                            001271
      CALL FETCH(NTAPE3,16,NREC3,B,NJ,NE,KB)                            001272
```

```
      CALL FETCH(NTAPE3,21,NREC3,C,NJ,NE,KC)                              001273
      CALL ADD3(-ONE,A, ONE,B,-ONE,C,NJ,NE,KA)                            001274
C***  FETCH SZ*HX,SX*HZ                                                   001275
      CALL FETCH(NTAPE3, 3,NREC3,B,NJ,NE,KB)                              001276
      CALL FETCH(NTAPE3, 8,NREC3,C,NJ,NE,KC)                              001277
      CALL ADD3( ONE,A, ONE,B,-ONE,C,NJ,NE,KA)                            001278
C***  FETCH SIGY                                                          001279
      CALL FETCH(NTAPE2, 9,NREC2,B,NJ,NE,KB)                              001280
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMU)                    001281
C***  FETCH JXZ*SIGX,JYZ*SIGY,JZZ*SIGZ                                    001282
      CALL FETCH(NTAPE3,12,NREC3,A,NJ,NE,KA)                              001283
      CALL FETCH(NTAPE3,17,NREC3,B,NJ,NE,KB)                              001284
      CALL FETCH(NTAPE3,22,NREC3,C,NJ,NE,KC)                              001285
      CALL ADD3(-ONE,A,-ONE,B, ONE,C,NJ,NE,KA)                            001286
C***  FETCH SX*HY,SY*HX                                                   001287
      CALL FETCH(NTAPE3, 5,NREC3,B,NJ,NE,KB)                              001288
      CALL FETCH(NTAPE3, 2,NREC3,C,NJ,NE,KC)                              001289
      CALL ADD3( ONE,A,ONE,B,-ONE,C,NJ,NE,KA)                             001290
C***  FETCH SIGZ                                                          001291
      CALL FETCH(NTAPE2,10,NREC2,B,NJ,NE,KB)                              001292
      CALL SATB( ONE,B,A,AMU(7,7),NJ,NE,NE,KB,KA,KAMU)                    001293
      CALL WRITE(AMU(7,7),NE,NE,5HECOEF,KAMU)                             001294
C                                                                         001295
      RETURN                                                              001296
      END                                                                 001297
```

```
[HDG,P     CREMUO                                                                -001298
[FOR,IS    CREMUO                                                                -001299
      COMPILER (XM=1),(EQUIV=CMN)                                                -001300
      SUBROUTINE CREMUO(NREC3,NJ,UVEC,A,WS,AMU,KA,KWS,KAMU)                       001301
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                         -001302
C                                                                                 001303
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES                                            001304
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3                                        001305
C                                                                                 001306
      DIMENSION A(KA,1),WS(KWS,1),AMU(KAMU,1),UVEC(1)                             001307
C                                                                                 001308
C***  FETCH M*RHOX,M*RHOY,M*RHOZ                                                  001309
      CALL FETCH(NTAPE3,25,NREC3,A(1,1),NJ,1,KA)                                  001310
      CALL FETCH(NTAPE3,26,NREC3,A(1,2),NJ,1,KA)                                  001311
      CALL FETCH(NTAPE3,27,NREC3,A(1,3),NJ,1,KA)                                  001312
C***  FORM JXXO                                                                   001313
      CALL SATB( ONE,WS(1,13), A(1, 3),AMU(1,1),NJ, 1, 1,KWS, KA,KAMU)            001314
      CALL SATB( ONE,WS(1,12), A(1, 2),AMU(1,1),NJ, 1, 1,KWS, KA,KAMU)            001315
      CALL SATB( TWO,WS(1,13),WS(1,10),AMU(1,1),NJ, 1, 1,KWS,KWS,KAMU)            001316
      CALL SATB( TWO,WS(1,12),WS(1, 9),AMU(1,1),NJ, 1, 1,KWS,KWS,KAMU)            001317
      CALL SATB( ONE,UVEC     ,WS(1, 2),AMU(1,1),NJ, 1, 1,KWS,KWS,KAMU)           001318
C***  FORM JXYO                                                                   001319
      CALL SATB(-ONE,WS(1,12), A(1, 1),AMU(1,2),NJ, 1, 1,KWS, KA,KAMU)            001320
      CALL SATB(-ONE,WS(1,12),WS(1, 8),AMU(1,2),NJ, 1, 1,KWS,KWS,KAMU)            001321
      CALL SATB(-ONE,WS(1,11),WS(1, 9),AMU(1,2),NJ, 1, 1,KWS,KWS,KAMU)            001322
      CALL SATB(-ONE,UVEC     ,WS(1, 5),AMU(1,2),NJ, 1, 1,KWS,KWS,KAMU)           001323
C***  FORM JXZO                                                                   001324
      CALL SATB(-ONE,WS(1,13), A(1, 1),AMU(1,3),NJ, 1, 1,KWS, KA,KAMU)            001325
      CALL SATB(-ONE,WS(1,13),WS(1, 8),AMU(1,3),NJ, 1, 1,KWS,KWS,KAMU)            001326
      CALL SATB(-ONE,WS(1,11),WS(1,10),AMU(1,3),NJ, 1, 1,KWS,KWS,KAMU)            001327
      CALL SATB(-ONE,UVEC     ,WS(1, 6),AMU(1,3),NJ, 1, 1,KWS,KWS,KAMU)           001328
C***  FORM JYYO                                                                   001329
      CALL SATB( ONE,WS(1,13), A(1, 3),AMU(2,2),NJ, 1, 1,KWS, KA,KAMU)            001330
      CALL SATB( ONE,WS(1,11), A(1, 1),AMU(2,2),NJ, 1, 1,KWS, KA,KAMU)            001331
      CALL SATB( TWO,WS(1,13),WS(1,10),AMU(2,2),NJ, 1, 1,KWS,KWS,KAMU)            001332
      CALL SATB( TWO,WS(1,11),WS(1, 8),AMU(2,2),NJ, 1, 1,KWS,KWS,KAMU)            001333
      CALL SATB( ONE,UVEC     ,WS(1, 3),AMU(2,2),NJ, 1, 1,KWS,KWS,KAMU)           001334
C***  FORM JYZO                                                                   001335
      CALL SATB(-ONE,WS(1,13), A(1, 2),AMU(2,3),NJ, 1, 1,KWS, KA,KAMU)            001336
      CALL SATB(-ONE,WS(1,13),WS(1, 9),AMU(2,3),NJ, 1, 1,KWS,KWS,KAMU)            001337
      CALL SATB(-ONE,WS(1,12),WS(1,10),AMU(2,3),NJ, 1, 1,KWS,KWS,KAMU)            001338
      CALL SATB(-ONE,UVEC     ,WS(1, 7),AMU(2,3),NJ, 1, 1,KWS,KWS,KAMU)           001339
C***  FORM JZZO                                                                   001340
      CALL SATB( ONE,WS(1,12), A(1, 2),AMU(3,3),NJ, 1, 1,KWS, KA,KAMU)            001341
      CALL SATB( ONE,WS(1,11), A(1, 1),AMU(3,3),NJ, 1, 1,KWS, KA,KAMU)            001342
      CALL SATB( TWO,WS(1,12),WS(1, 9),AMU(3,3),NJ, 1, 1,KWS,KWS,KAMU)            001343
      CALL SATB( TWO,WS(1,11),WS(1, 8),AMU(3,3),NJ, 1, 1,KWS,KWS,KAMU)            001344
      CALL SATB( ONE,UVEC     ,WS(1, 4),AMU(3,3),NJ, 1, 1,KWS,KWS,KAMU)           001345
C***  FORM SXO                                                                    001346
      CALL SATB( ONE,UVEC     ,WS(1, 8),AMU(3,5),NJ, 1, 1,KWS,KWS,KAMU)           001347
```

```
      CALL SATB( ONE,WS(1,11),WS(1, 1),AMU(3,5),NJ, 1, 1,KWS,KWS,KAMU)     001348
C***  FORM SYO                                                              001349
      CALL SATB( ONE,UVEC    ,WS(1, 9),AMU(1,6),NJ, 1, 1,KWS,KWS,KAMU)     001350
      CALL SATB( ONE,WS(1,12),WS(1, 1),AMU(1,6),NJ, 1, 1,KWS,KWS,KAMU)     001351
C***  FORM SZO                                                              001352
      CALL SATB( ONE,UVEC    ,WS(1,10),AMU(2,4),NJ, 1, 1,KWS,KWS,KAMU)     001353
      CALL SATB( ONE,WS(1,13),WS(1, 1),AMU(2,4),NJ, 1, 1,KWS,KWS,KAMU)     001354
      AMU(2,6) = -AMU(3,5)                                                  001355
      AMU(3,4) = -AMU(1,6)                                                  001356
      AMU(1,5) = -AMU(2,4)                                                  001357
C***  FORM MASS                                                             001358
      CALL SATB( ONE,UVEC    ,WS(1, 1),AMU(4,4),NJ, 1, 1,KWS,KWS,KAMU)     001359
      AMU(5,5) = AMU(4,4)                                                   001360
      AMU(6,6) = AMU(4,4)                                                   001361
C                                                                           001362
      CALL WRITE(AMU(1,1),3,3,5HINERO,KAMU)                                 001363
      CALL WRITE(AMU(1,4),3,3,5HSTATO,KAMU)                                 001364
      CALL WRITE(AMU(4,4),3,3,5HMASSO,KAMU)                                 001365
C                                                                           001366
      RETURN                                                                001367
      END                                                                   001368
```

```
[HDG,P    CRET3                                                          -001369
[FOR,IS   CRET3                                                          -001370
      COMPILER (XM=1),(EQUIV=CMN)                                        -001371
      SUBROUTINE CRET3(NREC2,NJ,NE,A,B,WS,KA,KB,KWS)                      001372
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -001373
C                                                                         001374
      COMMON /NUMBRS/ ZRO,ONE,TWO,TRES                                    001375
      COMMON /TAPENO/ NTAPE1,NTAPE2,NTAPE3                                001376
C                                                                         001377
      DIMENSION A(KA,1),B(KB,1),WS(KWS,1)                                 001378
C                                                                         001379
C***  FETCH M                                                             001380
      CALL FETCH(NTAPE2, 1,NREC2,WS(1, 1),NJ,1,KWS)                       001381
C***  FETCH JXX,.....,JYZ                                                 001382
      CALL FETCH(NTAPE2, 2,NREC2,WS(1, 2),NJ,6,KWS)                       001383
C***  FETCH SX,SY,SZ                                                      001384
      CALL FETCH(NTAPE2, 3,NREC2,WS(1, 8),NJ,3,KWS)                       001385
C***  FETCH GEOMETRY                                                      001386
      CALL FETCH(NTAPE2, 4,NREC2,WS(1,11),NJ,3,KWS)                       001387
C***  FETCH HX                                                            001388
      CALL FETCH(NTAPE2, 5,NREC2,A,NJ,NE,KA)                              001389
      CALL STORE(NTAPE3,WS(1, 1),A,B,NJ,NE,KWS,KA,KB)                     001390
      CALL STORE(NTAPE3,WS(1, 9),A,B,NJ,NE,KWS,KA,KB)                     001391
      CALL STORE(NTAPE3,WS(1,10),A,B,NJ,NE,KWS,KA,KB)                     001392
C***  FETCH HY                                                            001393
      CALL FETCH(NTAPE2, 6,NREC2,A,NJ,NE,KA)                              001394
      CALL STORE(NTAPE3,WS(1, 1),A,B,NJ,NE,KWS,KA,KB)                     001395
      CALL STORE(NTAPE3,WS(1, 8),A,B,NJ,NE,KWS,KA,KB)                     001396
      CALL STORE(NTAPE3,WS(1,10),A,B,NJ,NE,KWS,KA,KB)                     001397
C***  FETCH HZ                                                            001398
      CALL FETCH(NTAPE2, 7,NREC2,A,NJ,NE,KA)                              001399
      CALL STORE(NTAPE3,WS(1, 1),A,B,NJ,NE,KWS,KA,KB)                     001400
      CALL STORE(NTAPE3,WS(1, 8),A,B,NJ,NE,KWS,KA,KB)                     001401
      CALL STORE(NTAPE3,WS(1, 9),A,B,NJ,NE,KWS,KA,KB)                     001402
C***  FETCH SIGX                                                          001403
      CALL FETCH(NTAPE2, 8,NREC2,A,NJ,NE,KA)                              001404
      CALL STORE(NTAPE3,WS(1, 2),A,B,NJ,NE,KWS,KA,KB)                     001405
      CALL STORE(NTAPE3,WS(1, 5),A,B,NJ,NE,KWS,KA,KB)                     001406
      CALL STORE(NTAPE3,WS(1, 6),A,B,NJ,NE,KWS,KA,KB)                     001407
      CALL STORE(NTAPE3,WS(1, 9),A,B,NJ,NE,KWS,KA,KB)                     001408
      CALL STORE(NTAPE3,WS(1,10),A,B,NJ,NE,KWS,KA,KB)                     001409
C***  FETCH SIGY                                                          001410
      CALL FETCH(NTAPE2, 9,NREC2,A,NJ,NE,KA)                              001411
      CALL STORE(NTAPE3,WS(1, 5),A,B,NJ,NE,KWS,KA,KB)                     001412
      CALL STORE(NTAPE3,WS(1, 3),A,B,NJ,NE,KWS,KA,KB)                     001413
      CALL STORE(NTAPE3,WS(1, 7),A,B,NJ,NE,KWS,KA,KB)                     001414
      CALL STORE(NTAPE3,WS(1, 8),A,B,NJ,NE,KWS,KA,KB)                     001415
      CALL STORE(NTAPE3,WS(1,10),A,B,NJ,NE,KWS,KA,KB)                     001416
C***  FETCH SIGZ                                                          001417
      CALL FETCH(NTAPE2,10,NREC2,A,NJ,NE,KA)                              001418
```

```
      CALL STORE(NTAPE3,WS(1, 6),A,B,NJ,NE,KWS,KA,KB)                    001419
      CALL STORE(NTAPE3,WS(1, 7),A,B,NJ,NE,KWS,KA,KB)                    001420
      CALL STORE(NTAPE3,WS(1, 4),A,B,NJ,NE,KWS,KA,KB)                    001421
      CALL STORE(NTAPE3,WS(1, 8),A,B,NJ,NE,KWS,KA,KB)                    001422
      CALL STORE(NTAPE3,WS(1, 9),A,B,NJ,NE,KWS,KA,KB)                    001423
      CALL STORE(NTAPE3,WS(1, 1),WS(1,11),B,NJ,1,KWS,KWS,KB)             001424
      CALL STORE(NTAPE3,WS(1, 1),WS(1,12),B,NJ,1,KWS,KWS,KB)             001425
      CALL STORE(NTAPE3,WS(1, 1),WS(1,13),B,NJ,1,KWS,KWS,KB)             001426
C                                                                        001427
      RETURN                                                             001428
      END                                                                001429
```

```
[HDG,P    DCMPLX                                                        -001430
[FOR,IS   DCMPLX                                                        -001431
      COMPILER (XM=1),(EQUIV=CMN)                                       -001432
      COMPLEX FUNCTION DCMPLX(X,Y)                                      -001433
C                                                                       -001434
C     DOUBLE PRECISION ARGUMENTS X,Y ARE COMBINED TO FORM A             -001435
C     SINGLE PRECISION COMPLEX NUMBER RETURNED AS DCMPLX                -001436
C                                                                       -001437
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               -001438
C                                                                       -001439
      DCMPLX = CMPLX(SNGL(X),SNGL(Y))                                   -001440
      RETURN                                                            -001441
      END                                                               -001442
```

```
[HDG,P    DCOM2                                                          -001443
[FOR,IS   DCOM2                                                          -001444
      COMPILER (XM=1),(EQUIV=CMN)                                        -001445
      SUBROUTINE DCOM2 (U,D,N,KR)                                         001446
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -001447
      DIMENSION U(KR,1),D(1)                                              001448
      DATA EPS,NOT /1.D-15, 6/                                            001449
C                                                                         001450
      IF (N .EQ. 1) GO TO 20                                              001451
      NM1 = N - 1                                                         001452
      DO 15 L=1,NM1                                                       001453
      IF (DABS(U(L,L)) .LT. EPS) GO TO 998                                001454
      LP1 = L + 1                                                         001455
      D(L) = U(L,L)                                                       001456
      DO 15 I=LP1,N                                                       001457
      S = U(L,I)/U(L,L)                                                   001458
      DO 10 J=I,N                                                         001459
   10 U(I,J) = U(I,J) - S*U(L,J)                                          001460
   15 U(L,I) = S                                                          001461
   20 D(N) = U(N,N)                                                       001462
      RETURN                                                              001463
C                                                                         001464
  998 WRITE (NOT,1001)                                                    001465
 1001 FORMAT (1H1,40HMATRIX SINGULAR, DCOM2, PROGRAM STOPPED.)            001466
      STOP                                                                001467
      END                                                                 001468
```

```
[HDG,P    DCQRRT                                                        -001469
[FOR,IS   DCQRRT                                                        -001470
      COMPILER (XM=1),(EQUIV=CMN)                                       -001471
      SUBROUTINE DCQRRT (RR,RI,N,KR,KC,KZ,RLRT,CMPR)                     001472
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -001473
C                                                                       001474
C  SUBROUTINE RECEIVES QR ROOT OUTPUT OF THE FORM,                      001475
C                RR(I),RI(I),I=1,N                                      001476
C             AND PLACES REAL ROOTS (INCLUDING ZEROS) INTO              001477
C             MATRIX RLRT (SIZE KR), THEN PLACES THE                    001478
C             COMPLEX PAIRS (A(I) +J B(I)) INTO                         001479
C             MATRIX CMPR-- COMPLEX PAIR ORDER IS                       001480
C                                                 CMPR(I) = A(I)        001481
C                                                 CMPR(I+1) = B(I)      001482
C             SAVES ONLY REAL AND POSITIVE IMAG PARTS IN CMPR.          001483
C             SIZE OF CMPR IS 2 * KC.                                   001484
C                                                                       001485
C             -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                001486
C                                                                       001487
C  RR     = INPUT  ARRAY OF REAL PARTS.  SIZE IS N                      001488
C  RI     = INPUT  ARRAY OF IMAGINARY PARTS.  SIZE IS N.                001489
C  N      = INPUT  SIZE OF RR AND RI.  NUMBER OF QR ROOTS.              001490
C  KR     = OUTPUT SIZE OF RLRT.  NUMBER OF REAL ROOT2,                 001491
C                                 (INCLUDING ZEROS).                    001492
C  KC     =                       NUMBER OF COMPLEX PAIRS.              001493
C  KZ     = OUTPUT NUMBER OF ZEROS.                                     001494
C                                                                       001495
C  RLRT   = OUTPUT REAL ROOT ARRAY.  SIZE KR.                           001496
C  CMPR   = OUTPUT  COMPLEX PAIR ARRAY. SIZE 2*KC.                      001497
      DIMENSION RR(1),RI(1),RLRT(1),CMPR(1)                             001498
      KR = 0                                                            001499
      KC = 0                                                            001500
      KZ = 0                                                            001501
C                                                                       001502
      DO 10 I=1,N                                                       001503
         IF (RI(I) .EQ. 0.D0) GO TO 5                                   001504
         IF (RI(I) .LT. 0.D0) GO TO 10                                  001505
           KC = KC+1                                                    001506
           L  = 2*KC                                                    001507
         CMPR(L-1) = RR(I)                                              001508
         CMPR(L  ) = RI(I)                                              001509
      GO TO 10                                                          001510
    5 CONTINUE                                                          001511
           KR = KR+1                                                    001512
         RLRT(KR) = RR(I)                                               001513
         IF (RR(I) .EQ. 0.D0) KZ = KZ+1                                 001514
   10 CONTINUE                                                          001515
C                                                                       001516
      RETURN                                                            001517
      END                                                               001518
```

```
[HDG,P    DFORMB                                                          -001519
[FOR,IS   DFORMB                                                          -001520
      COMPILER (XM=1),(EQUIV=CMN)                                         -001521
      SUBROUTINE DFORMB(KR,KC,RLRT,CMPR,FBR,FBC,SCMPR,SF,ACCD,GB)          001522
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                  -001523
CDFORMB      FACTORED  FORM  TIME CONSTANTS, DAMPING AND FREQUENCY         001524
      DIMENSION RLRT(1),CMPR(1),FBR(1),FBC(1),SCMPR(1)                    001525
      DIMENSION GB(2)                                                     001526
      GB(1)=ACCD                                                          001527
      LZ=0                                                                001528
      SCI=1.DO/SF                                                         001529
      IF(KR)140, 140, 100                                                 001530
  100 DO 130 L=1,KR                                                       001531
      IF(RLRT(L))120, 110, 120                                            001532
  110 FBR(L)=0.DO                                                         001533
      LZ=LZ+1                                                             001534
      GO TO 130                                                           001535
  120 FBR(L)=-1.DO/(RLRT(L)*SCI)                                          001536
      GB(1)=-GB(1)*RLRT(L)                                                001537
  130 CONTINUE                                                            001538
  140 IF(KC)170, 170, 150                                                 001539
  150 KK=2*KC                                                             001540
      DO 160 L=2,KK,2                                                     001541
      GB(1)=GB(1)*(CMPR(L-1)**2+CMPR(L)**2)                               001542
      SCMPR(L-1)=CMPR(L-1)*SCI                                            001543
      FBC(L)=DSQRT(SCMPR(L-1)**2+(CMPR(L)*SCI)**2)                        001544
  160 FBC(L-1)= -SCMPR(L-1)/FBC(L)                                        001545
  170 GB(2)=GB(1)                                                         001546
      IF (LZ .EQ. 0) GO TO 180                                            001547
      GB(2)=GB(2)*SF**LZ                                                  001548
  180 RETURN                                                              001549
      END                                                                 001550
```

```
[HDG,P    DIMAG                                                    -001551
[FOR,IS   DIMAG                                                    -001552
      COMPILER (XM=1),(EQUIV=CMN)                                  -001553
      DOUBLE PRECISION FUNCTION DIMAG (Y)                          -001554
      COMPLEX    Y                                                  001555
      COMPLEX SY                                                    001556
      SY = Y                                                        001557
      DIMAG = DBLE (AIMAG(SY))                                      001558
      RETURN                                                        001559
      END                                                           001560
```

```
[HDG,P    DREAL                                                    -001561
[FOR,IS   DREAL                                                    -001562
      COMPILER (XM=1),(EQUIV=CMN)                                  -001563
      DOUBLE PRECISION FUNCTION DREAL (Y)                          -001564
      COMPLEX    Y                                                  001565
      COMPLEX SY                                                    001566
      SY = Y                                                        001567
      DREAL = DBLE (REAL(SY))                                       001568
      RETURN                                                        001569
      END                                                           001570
```

```
[HDG,P    DYNSAA                                                          -001571
[FOR,IS   DYNSAA                                                          -001572
      COMPILER (XM=1),(EQUIV=CMN)                                         -001573
      SUBROUTINE DYNSAA                                                   -001574
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                  -001575
C                                                                          001576
C                                                                          001577
            COMMON /CONPAR/                                                001578
     *      CNTDTA(100)                                                  9501579
            COMMON /GGDATA/                                                001580
     *      GAMGI(3),GMAG,RCMAG                                            001581
            COMMON /ILINER/                                                001582
     *      IFLNER                                                         001583
            COMMON /MAXMUM/                                                001584
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU              001585
            COMMON /MISCNO/                                                001586
     *      NOPRNT,NOPLOT                                                  001587
            COMMON /NHNS  /                                                001588
     *      NHPOI( 5), NSPOI( 5)                                         1201589
            COMMON /SPECIF/                                                001590
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),    1601591
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),           1701592
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1801593
     *      LOCU(12),LENU(12),MU,NBETA,NLAM,NEQ                          1901594
            COMMON /SUMMRY/                                                001595
     *      ASUMRY(10,6),ISUMRY(10,3),KSUMRY                             9801596
            COMMON /TAPENO/                                                001597
     *      NTAPE1,NTAPE2,NTAPE3                                           001598
            COMMON /TIMESS/                                                001599
     *      STARTT,DELTAT,T,ENDT,TMST                                      001600
C                                                                          001601
      DIMENSION WV(5),TMDATA(3),IPDATA(3)                                  001602
      EQUIVALENCE (GAMGI(1),WV(1))                                         001603
C                                                                          001604
C                                                                          001605
C    NB     = NO. OF BODIES                                                001606
C    NH     = NO. OF HINGES    (.GE. NB)                                   001607
C    IRGFLX(L) = (0 IF L IS RIGID), (M(L) IF L IS FLEXIBLE)                001608
C                M(L) = NO. OF MODES OF BODY (L)                           001609
C                L=1,NB                                                    001610
C    NMOW(L) = NO. MOMENTUM WHEELS/BODY(L), L=1,NB                         001611
C    ITOPOL(1,1) = 1    (BODY 1)                                           001612
C    ITOPOL(2,1) = 0    (INERTIAL REFERENCE)                               001613
C    ITOPOL(1,K) = L1,   (K .GT. 1)                                        001614
C    ITOPOL(2,K) = L2,   (K .GT. 1)                                        001615
C             BODY(L1) REL. TO BODY(L2),  K=1,NH.                          001616
C    IHDATA(1,K) = ITYPE (EULER PERMUTATION 1,2,..12), K=1,NH              001617
C          (J,K) = 0  (FORCED/FREE)                                        001618
C          (J,K) = 1  (FIXED CONSTRAINT)                                   001619
C          (J,K) = 2  (RHEONOMIC CONSTRAINT),    --  J=2,7,    K=1,NH     001620
```

```
C    BETAH(J,K) = INITIAL HINGE COORDINATES, (J=1,6,  K=1,NH)                  001621
C    BETAHD(J,K) = INITIAL HINGE RATES,      (J=1,6,  K=1,NH)                  001622
C    AM(L)  = MASS OF BODY(L),  L=1,NB                                         001623
C    SMOM(I,L) = STATIC MOMENT OF BODY(L),  L=1,NB                             001624
C    AIN(I,L)  = MOMENT OF INERTIA PROPERTIES OF BODY(L), I=1,6,  L=1,NB       001625
C                                                                              001626
C                                                                              001627
 1001 FORMAT (16I5)                                                            001628
 2001 FORMAT (//15X48HSUMMARY OF DYNAMIC-SIMULATION-PROGRAM INPUT DATA,        001629
     *  10(2H *),//3X12HACTUAL SIZES5X13HMAXIMUM SIZES4X12HINTEGRATION        001630
     *  4HDATA 12X21HGRAVITY GRADIENT DATA17X10HMISC. DATA,                   001631
     *  /3X13(1H-),4X13(1H-),4X18(1H-),5X37(1H-),6X11(1H-),                   001632
     *  /3X9HNB      = I4,4X9HNBMAX  = I4,4X9HSTARTT = 1PD10.3,4X7HG1    =   001633
     *  1PD10.3,4X8HGAMA1 = 1PD10.3,4X9HNOPRNT = I2,                          001634
     *  /3X9HNH      = I4,4X9HNHMAX  = I4,4X9HDELTAT = 1PD10.3,4X7HG2    =   001635
     *  1PD10.3,4X8HGAMA2 = 1PD10.3,4X9HNOPLOT = I2,                          001636
     *  /3X9HNSPT    = I4,4X9HNSPMAX = I4,4X9HENDT   = 1PD10.3,4X7HG3    =   001637
     *  1PD10.3,4X8HGAMA3 = 1PD10.3,4X9HIFLNER = I2,                          001638
     *  /3X9HNOFMO   = I4,4X9HNMWMAX = I4,                    27X7HGMAG =     001639
     *  1PD10.3,4X8HRCMAG = 1PD10.3,                                          001640
     *  /3X9HNDELTA = I4,4X9HNMWBOD = I4,                                     001641
     *  /3X9HNU      = I4,4X9HNMDBOD = I4,                                    001642
     *  /3X9HNBETA   = I4,4X9HKMU    = I4,                                    001643
     *  /3X9HNLAM    = I4,4X9HKY     = I4,                                    001644
     *  /3X9HNEQ     = I4,4X9HKU     = I4)                                    001645
 2002 FORMAT (//1X49HTHE TOPOLOGY ARRAY (ITOPOL) FOR THIS CASE FOLLOWS)        001646
 2003 FORMAT (//1X50HTHE CONSTRAINT SPECIFICATIONS FOR THIS CASE FOLLOW)       001647
 2004 FORMAT (//1X39HTHE SPECIFIED INITIAL HINGE ANGLES AND                    001648
     *    28HDISPLACEMENTS (BETAH) FOLLOW)                                    001649
 2005 FORMAT (//1X49HTHE SPECIFIED INITIAL HINGE RATES (BETAHD) FOLLOW)        001650
 2006 FORMAT (//1X45HTHE NO. OF ELASTIC MODES/BODY ARRAY (IRGFLX)              001651
     * 7HFOLLOWS)                                                             001652
 2007 FORMAT (//1X47HTHE NO. OF P/Q HINGE POINTS/BODY ARRAY (NHPOI)            001653
     * 7HFOLLOWS)                                                             001654
 2008 FORMAT (//1X44HTHE NO. OF SENSOR POINTS/BODY ARRAY (NSPOI)               001655
     * 7HFOLLOWS)                                                             001656
 2009 FORMAT (//1X40HTHE MOM. WHEEL/BODY TABLE (NMOW) FOLLOWS )                001657
 2010 FORMAT (//1X44HTHE STATE VECTOR LENGTH ARRAY (LENU) FOLLOWS )            001658
 2011 FORMAT (//1X46HTHE STATE VECTOR LOCATION ARRAY (LOCU) FOLLOWS)           001659
 2012 FORMAT (//1X50HTHE SPECIFIED SENSOR POINT/BODY CORRELATION ARRAY         001660
     * 16H(IFTSMW) FOLLOWS )                                                  001661
 2013 FORMAT (//1X43HTHE FOLLOWING DATA IS SPECIFIED MOM. WHEEL                001662
     * 47HINFORMATION (IF ANY) AND CONTROLLER INFORMATION /1X90(1H-))         001663
 2014 FORMAT (//1X45HTHE SPECIFIED MOM. WHEEL CONTROL ARRAY (IMO)              001664
     * 7HFOLLOWS)                                                             001665
 2015 FORMAT (//1X50HTHE SPECIFIED MOM. WHEEL RATES AND INERTIAS (AMO)         001666
     * 7HFOLLOW  )                                                            001667
 2016 FORMAT (//1X,48HTHE SPECIFIED CONTROLLER INITIAL CONDITIONS AND          001668
     * 22HCHARACTERISTICS FOLLOW  /3X30H(THE FIRST NDELTA ARE INITIAL         001669
     *  38HCONTROLLER STATE VARIABLES, THERE ARE I3,12H ADDITIONAL            001670
```

```
     *  19HCONTROL PARAMETERS))                                          001671
 2110 FORMAT (1H1, 28HINPUT DATA ERROR, NERROR =  I3)                     001672
C                                                                         001673
      DATA NIT,NOT / 5, 6/                                                001674
C                                                                         001675
CCCCC                                                                     001676
      KCONT = 100                                                        9601677
      KSUMRY = 10                                                        9901678
CCCCC                                                                     001679
      READ (NIT,1001) NB,NH,NSPT,NOFMO,NDELTA                             001680
C                                                                         001681
                                                        NERROR =  1       001682
      CALL READIM (ITOPOL,N1,N2,2,NHMAX)                                  001683
      IF (N1.NE.2 .OR. N2.NE.NH .OR. NH.LT.NB) GO TO 999                  001684
                                                        NERROR =  2       001685
      CALL READIM (IRGFLX,N1,N2,1,NBMAX)                                  001686
      IF (N2 .NE. NB) GO TO 999                                           001687
                                                        NERROR =  3       001688
      CALL READIM (IFTSMW,N1,N2,1,NSPMAX)                                 001689
      IF (N2 .NE. NSPT) GO TO 999                                         001690
                                                        NERROR =  4       001691
      CALL READIM (IHDATA,N1,N2, 7,NHMAX)                                 001692
      IF (N1.NE.7 .OR. N2.NE.NH) GO TO 999                                001693
                                                        NERROR =  5       001694
      CALL READ  (BETAH,N1,N2, 6,NHMAX)                                   001695
      IF (N1.NE.6 .OR. N2.NE.NH) GO TO 999                                001696
                                                        NERROR =  6       001697
      CALL READ  (BETAHD,N1,N2, 6,NHMAX)                                  001698
      IF (N1.NE.6 .OR. N2.NE.NH) GO TO 999                                001699
                                                        NERROR =  7       001700
      IF (ITOPOL(1,1).NE.1 .OR. ITOPOL(2,1).NE.0) GO TO 999               001701
                                                        NERROR =  8       001702
      DO 605 J=2,NH                                                       001703
      IF (ITOPOL(1,J) .EQ. ITOPOL(2,J)) GO TO 999                         001704
  605 CONTINUE                                                            001705
C                                                                         001706
CCC   PRELIMINARY TOPOLOGY CHECK, COMPLETE CHECK DONE BY ROTDH ....       001707
                                                        NERROR =  9       001708
      DO 610 N=1,NB                                                       001709
      DO 615 I=1,2                                                        001710
      DO 615 J=2,NH                                                       001711
      IF (ITOPOL(I,J) .EQ. N) GO TO 610                                   001712
  615 CONTINUE                                                            001713
      GO TO 999                                                           001714
  610 CONTINUE                                                            001715
                                                        NERROR = 10       001716
      DO 620 J=1,NB                                                       001717
      IF (IRGFLX(J).LT.0 .OR. IRGFLX(J).GT.NMDBOD) GO TO 999              001718
  620 CONTINUE                                                            001719
                                                        NERROR = 11       001720
```

```
      DO 625 J=1,NSPT                                                    001721
      IF (IFTSMW(J).LT.1 .OR. IFTSMW(J).GT.NB) GO TO 999                 001722
  625 CONTINUE                                                           001723
                                                        NERROR = 12      001724
      DO 630 J=1,NH                                                      001725
      IF (IHDATA(1,J).LT.1 .OR. IHDATA(1,J).GT.12) GO TO 999             001726
      DO 630 I=2,7                                                       001727
      IF (IHDATA(I,J).LT.0 .OR. IHDATA(I,J).GT.2) GO TO 999              001728
  630 CONTINUE                                                           001729
C                                                                        001730
      DO 5 N=1,NB                                                        001731
      NHPOI(N) = 0                                                       001732
      NSPOI(N) = 0                                                       001733
      DO 10 I=1,2                                                        001734
      DO 10 J=2,NH                                                       001735
      IF (ITOPOL(I,J) .EQ. N) NHPOI(N) = NHPOI(N) + 1                    001736
   10 CONTINUE                                                           001737
      DO 15 J=1,NSPT                                                     001738
      IF (IFTSMW(J) .EQ. N) NSPOI(N) = NSPOI(N) + 1                      001739
   15 CONTINUE                                                           001740
    5 CONTINUE                                                           001741
C                                                                        001742
      I1 = NMWBOD + 2                                                    001743
      DO 40 I=1,I1                                                       001744
      DO 40 J=1,NBMAX                                                    001745
   40 NMOW(I,J) = 0                                                      001746
      IF (NOFMO .EQ. 0) GO TO 41                                         001747
                                                        NERROR = 13      001748
      CALL READIM (IMO,N1,N2, 3,NMWMAX)                                  001749
      IF (N1.NE.3 .OR. N2.NE.NOFMO) GO TO 999                            001750
                                                        NERROR = 14      001751
      CALL READ  (AMO,N1,N2, 2,NMWMAX)                                   001752
      IF (N1.NE.2 .OR. N2.NE.NOFMO) GO TO 999                            001753
                                                        NERROR = 15      001754
      DO 635 J=1,NOFMO                                                   001755
      IF (IMO(1,J).LT.1 .OR. IMO(1,J).GT.NSPT) GO TO 999                 001756
      IF (IMO(2,J).LT.1 .OR. IMO(2,J).GT.3) GO TO 999                    001757
      IF (IMO(3,J).LT.0 .OR. IMO(3,J).GT.1) GO TO 999                    001758
      IF (AMO(2,J) .LE. 0.D 0) GO TO 999                                 001759
  635 CONTINUE                                                           001760
                                                        NERROR = 16      001761
      IF (NOFMO .EQ. 1) GO TO 641                                        001762
      N1 = NOFMO - 1                                                     001763
      DO 640 I=1,N1                                                      001764
      IP1 = I + 1                                                        001765
      DO 640 J=IP1,NOFMO                                                 001766
      IF (IMO(1,I) .EQ. IMO(1,J)) GO TO 999                              001767
  640 CONTINUE                                                           001768
                                                        NERROR = 17      001769
  641 DO 45 N=1,NOFMO                                                    001770
```

```
      NPTS = IMO(1,N)                                                    001771
      NBOD = IFTSMW(NPTS)                                                001772
      NMOW(1,NBOD) = NMOW(1,NBOD) + 1                                    001773
      IF (IMO(3,N) .NE. 0) NMOW(2,NBOD) = NMOW(2,NBOD) + 1               001774
      IC = NMOW(1,NBOD) + 2                                              001775
      IF (IC .GT. I1) GO TO 999                                          001776
   45 NMOW(IC,NBOD) = N                                                  001777
   41 CONTINUE                                                           001778
C                                                                        001779
      NEQ = 0                                                            001780
      DO 50 N=1,NB                                                       001781
      NEQ = NEQ + IRGFLX(N)                                              001782
   50 LENU(N) = 6 + IRGFLX(N) + NMOW(2,N)                                001783
      LOCU(1) = 1                                                        001784
      DO 55 N=2,NB                                                       001785
   55 LOCU(N) = LOCU(N - 1) + LENU(N - 1)                                001786
      NU = LOCU(NB) + LENU(NB) - 1                                       001787
      DO 56 N=1,NB                                                       001788
   56 LENU(N+NB) = IRGFLX(N)                                             001789
      DO 57 N=1,NB                                                       001790
      NM1 = N - 1                                                        001791
   57 LOCU(N+NB) = LOCU(NM1+NB) + LENU(NM1+NB)                           001792
      NBETA = 0                                                          001793
      NLAM = 0                                                           001794
      DO 60 I=2,7                                                        001795
      DO 60 J=1,NH                                                       001796
      IF (IHDATA(I,J) .NE. 1) NBETA = NBETA + 1                          001797
      IF (IHDATA(I,J) .NE. 0) NLAM = NLAM + 1                            001798
   60 CONTINUE                                                           001799
      NEQ = NEQ + NBETA + NDELTA + NU                                    001800
      LENU(2*NB+1) = NBETA                                               001801
      LENU(2*NB+2) = NDELTA                                              001802
      LOCU(2*NB+1) = LOCU(2*NB) + LENU(2*NB)                             001803
      LOCU(2*NB+2) = LOCU(2*NB+1) + NBETA                                001804
C                                                                        001805
      CALL READ (TMDATA,N1,N2,1,3)                                       001806
      CALL READIM (IPDATA,N1,N2,1,3)                                     001807
      CALL READ (CNTDTA,N1,NCNPAR,1,KCONT)                               001808
                                                    NERROR = 18          001809
      CALL READ  (WV,N1,N2,1,5)                                          001810
      IF (N2 .NE. 4) GO TO 999                                           001811
      RCMAG = WV(4)                                                      001812
      GMAG = DSQRT(GAMGI(1)**2 + GAMGI(2)**2 + GAMGI(3)**2)              001813
CC WV(5) IS NOW RCMAG, WV(4) IS GMAG                                     001814
      IF (GMAG .EQ. 0.D 0) GO TO 75                                      001815
      IF (RCMAG .LE. 1.D 0) GO TO 999                                    001816
      DO 70 J=1,3                                                        001817
   70 GAMGI(J) = GAMGI(J)/GMAG                                           001818
C                                                                        001819
   75 CALL PAGEHD                                                        001820
```

```
      LELO = 2*NB + 2                                                   001821
      STARTT = TMDATA(1)                                                001822
      DELTAT = TMDATA(2)                                                001823
      ENDT = TMDATA(3)                                                  001824
      NOPRNT = IPDATA(1)                                                001825
      NOPLOT = IPDATA(2)                                                001826
      IFLNER = IPDATA(3)                                                001827
      G1 = GMAG*GAMGI(1)                                                001828
      G2 = GMAG*GAMGI(2)                                                001829
      G3 = GMAG*GAMGI(3)                                                001830
      WRITE (NOT,2001) NB,NBMAX,STARTT,G1,GAMGI(1),NOPRNT,NH,NHMAX,     001831
     *  DELTAT,G2,GAMGI(2),NOPLOT,NSPT,NSPMAX,ENDT,G3,GAMGI(3),IFLNER,   001832
     *  NOFMO,NMWMAX,GMAG,RCMAG,NDELTA,NMWBOD,NU,NMDBOD,NBETA,KMU,       001833
     *  NLAM,KY,NEQ,KU                                                   001834
      WRITE (NOT,2002)                                                  001835
      CALL WRITIS (ITOPOL,2,NH,2)                                       001836
      WRITE (NOT,2003)                                                  001837
      CALL WRITIS (IHDATA,7,NH,7)                                       001838
      WRITE (NOT,2004)                                                  001839
      CALL WRITES (BETAH,6,NH,6)                                        001840
      WRITE (NOT,2005)                                                  001841
      CALL WRITES (BETAHD,6,NH,6)                                       001842
      CALL PAGEHD                                                       001843
      WRITE (NOT,2006)                                                  001844
      CALL WRITIS (IRGFLX,1,NB,1)                                       001845
      WRITE (NOT,2007)                                                  001846
      CALL WRITIS (NHPOI,1,NB,1)                                        001847
      WRITE (NOT,2008)                                                  001848
      CALL WRITIS (NSPOI,1,NB,1)                                        001849
      WRITE (NOT,2009)                                                  001850
      CALL WRITIS (NMOW,I1,NB,I1)                                       001851
      WRITE (NOT,2010)                                                  001852
      CALL WRITIS (LENU,1,LELO,1)                                       001853
      WRITE (NOT,2011)                                                  001854
      CALL WRITIS (LOCU,1,LELO,1)                                       001855
      WRITE (NOT,2012)                                                  001856
      CALL WRITIS (IFTSMW,1,NSPT,1)                                     001857
      CALL PAGEHD                                                       001858
      WRITE (NOT,2013)                                                  001859
      IF (NOFMO .NE. 0) WRITE (NOT,2014)                                001860
      IF (NOFMO .NE. 0) CALL WRITIS (IMO,3,NOFMO,3)                     001861
      IF (NOFMO .NE. 0) WRITE (NOT,2015)                                001862
      IF (NOFMO .NE. 0) CALL WRITES (AMO,2,NOFMO,2)                     001863
      NCNTRL = NCNPAR - NDELTA                                          001864
      WRITE (NOT,2016) NCNTRL                                           001865
      CALL WRITES (CNTDTA,1,NCNPAR,1)                                   001866
C                                                                       001867
      DO 20 N=1,NB                                                      001868
      IF (IRGFLX(N) .EQ. 0) GO TO 25                                    001869
      READ (NIT,1001) NTYPE                                             001870
```

```
      IF (NTYPE .EQ. 1) CALL MSMODL (N)                                 001871
      IF (NTYPE .EQ. 2) CALL MSMODC (N)                                 001872
      GO TO 20                                                          001873
   25 CALL MRIGID (N)                                                   001874
C                                                                       001875
   20 CONTINUE                                                          001876
C                                                                       001877
      RETURN                                                            001878
  999 WRITE (NOT,2110) NERROR                                           001879
      STOP                                                              001880
      END                                                               001881
```

```
[HDG,P    DYNSBB                                                          -001882
[FORL,IS  DYNSBB                                                          -001883
      COMPILER (XM=1),(EQUIV=CMN)                                         -001884
      SUBROUTINE DYNSBB                                                   -001885
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                  -001886
C                                                                          001887
            COMMON /AMUBW /                                                001888
     *      AMU(15,15, 5),BW(30, 65)                                      101889
            COMMON /CONPAR/                                                001890
     *      CNTDTA(100)                                                  9501891
            COMMON /DNAUX /                                                001892
     *      NAUX                                                          001893
            COMMON /HANDS /                                                001894
     *      HATH(3, 6, 8),SIGH(3, 6, 8),HATS(3, 6,10),SIGS(3, 6,10)       401895
            COMMON /ILINER/                                                001896
     *      IFLNER                                                        001897
            COMMON /INTGRL/                                                001898
     *      AM( 78, 5),ACOF(9, 6, 5),BCOF(6, 6, 5),                       501899
     *      COF11( 6, 6, 5),COF22( 6, 6, 5),COF33( 6, 6, 5),AK( 6, 6, 5), 601900
     *      COF12( 6, 6, 5),COF13( 6, 6, 5),COF23( 6, 6, 5),AD( 6, 6, 5), 701901
     *      COFXY( 6, 6, 5),COFXZ( 6, 6, 5),COFYZ( 6, 6, 5)               801902
            COMMON /MAXMUM/                                                001903
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU             001904
            COMMON /MISCNO/                                                001905
     *      NOPRNT,NOPLOT                                                 001906
            COMMON /NUMBRS/                                                001907
     *      ZRO,ONE,TWO,TRES                                              001908
            COMMON /PLTDTA/                                                001909
     *      NRPLOT,NCPLOT                                                 001910
            COMMON /QPRKTA/                                                001911
     *      QRK(250),PRK(4), NT                                          1501912
            COMMON /SPECIF/                                                001913
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),     1601914
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),            1701915
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1801916
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                           1901917
            COMMON /TAPENO/                                                001918
     *      NTAPE1,NTAPE2,NTAPE3                                          001919
            COMMON /TIMESS/                                                001920
     *      STARTT,DELTAT,T,ENDT,TMST                                     001921
            COMMON /VECTOR/                                                001922
     *      Y(250),YDT(250)                                              2001923
            COMMON /VINDEP/                                                001924
     *      INDEP(250)                                                   2101925
C                                                                          001926
      DATA NOT / 6/                                                        001927
C                                                                          001928
 2001 FORMAT (////1X,47HTHE FOLLOWING INTEGER ARRAY (INDEP) PRESCRIBES     001929
     *54HINDEPENDENT VARIABLES (1), AND DEPENDENT VARIABLES (0), /1X,       001930
     * 101(1H-))                                                           001931
```

```
C                                                                       001932
      PRK(1) = ONE/TWO                                                  001933
      PRK(2) = ONE - DSQRT(PRK(1))                                      001934
      PRK(3) = TWO - PRK(2)                                             001935
      PRK(4) = PRK(1)                                                   001936
C                                                                       001937
      NT = 0                                                            001938
      T    = STARTT                                                     001939
      TMST = ZRO                                                        001940
C                                                                       001941
      DO 100 I=1,NEQ                                                    001942
  100 ORK(I) = ZRO                                                      001943
C                                                                       001944
C                                                                       001945
      REWIND NTAPE1                                                     001946
C                                                                       001947
      DO 2 I=1,KY                                                       001948
      INDEP(I) = 1                                                      001949
      Y(I) = ZRO                                                        001950
    2 YDT(I) = ZRO                                                      001951
C                                                                       001952
      DO 3 N=1,NB                                                       001953
      NXE = IRGFLX(N)                                                   001954
      NP6 = 6 + NXE                                                     001955
      READ (NTAPE1) ((AMU(I,J,N),J=1,NP6),I=1,NP6)                      001956
      KNT = 0                                                           001957
      DO 6 I=1,NP6                                                      001958
      DO 6 J=I,NP6                                                      001959
      KNT = KNT + 1                                                     001960
    6 AM(KNT,N) = AMU(I,J,N)                                            001961
      IF (NXE .EQ. 0) GO TO 3                                           001962
      LXE  = LOCU(N+NB)                                                 001963
      LXED = LOCU(N   ) + 6                                             001964
      LXEN  = LXE  + NXE - 1                                            001965
      LXEDN = LXED + NXE - 1                                            001966
      READ (NTAPE1) ((ACOF(I,J,N),J=1,NXE),I=1,9)                       001967
      READ (NTAPE1) ((BCOF(I,J,N),J=1,NXE),I=1,6)                       001968
      READ (NTAPE1) ((COFXY(I,J,N),J=1,NXE),I=1,NXE)                    001969
      READ (NTAPE1) ((COFXZ(I,J,N),J=1,NXE),I=1,NXE)                    001970
      READ (NTAPE1) ((COFYZ(I,J,N),J=1,NXE),I=1,NXE)                    001971
      READ (NTAPE1) ((COF11(I,J,N),J=1,NXE),I=1,NXE)                    001972
      READ (NTAPE1) ((COF22(I,J,N),J=1,NXE),I=1,NXE)                    001973
      READ (NTAPE1) ((COF33(I,J,N),J=1,NXE),I=1,NXE)                    001974
      READ (NTAPE1) ((COF12(I,J,N),J=1,NXE),I=1,NXE)                    001975
      READ (NTAPE1) ((COF13(I,J,N),J=1,NXE),I=1,NXE)                    001976
      READ (NTAPE1) ((COF23(I,J,N),J=1,NXE),I=1,NXE)                    001977
      READ (NTAPE1) ((AK   (I,J,N),J=1,NXE),I=1,NXE)                    001978
      READ (NTAPE1) ((AD   (I,J,N),J=1,NXE),I=1,NXE)                    001979
      READ (NTAPE1) (Y(J),J=LXE ,LXEN )                                 001980
      READ (NTAPE1) (Y(J),J=LXED,LXEDN)                                 001981
```

```
      READ (NTAPE1) NHB                                                 001982
CCC NOTE ---  NHB IS NO. OF P/Q HINGES, NOT TO INCLUDE HINGE-1.         001983
CCC         OVERLAY(1,0) MUST BE SURE OF THIS.                          001984
      DO 4 L=1,NHB                                                      001985
      READ (NTAPE1) NOH                                                 001986
CCCC NOH NOT TO INCLUDE HINGE 1  ****                                   001987
      LHS = 0                                                           001988
      IF (ITOPOL(1,NOH) .EQ. N) LHS = 2*NOH - 3                         001989
      IF (ITOPOL(2,NOH) .EQ. N) LHS = 2*NOH - 2                         001990
      IF (LHS .LE. 0) GO TO 999                                         001991
      READ (NTAPE1) ((HATH(I,J,LHS),J=1,NXE),I=1,3)                     001992
    4 READ (NTAPE1) ((SIGH(I,J,LHS),J=1,NXE),I=1,3)                     001993
      READ (NTAPE1) NSB                                                 001994
      IF (NSB .EQ. 0) GO TO 3                                           001995
      DO 5 L=1,NSB                                                      001996
      READ (NTAPE1) NOS                                                 001997
      READ (NTAPE1) ((HATS(I,J,NOS),J=1,NXE),I=1,3)                     001998
    5 READ (NTAPE1) ((SIGS(I,J,NOS),J=1,NXE),I=1,3)                     001999
    3 CONTINUE                                                          002000
C                                                                       002001
      DO 10 N=1,NB                                                      002002
      LTD = LOCU(N) + IRGFLX(N) + 5                                     002003
      NMW = NMOW(1,N)                                                   002004
      IF (NMW .EQ. 0) GO TO 10                                          002005
      NMWVS = 0                                                         002006
      DO 15 I=1,NMW                                                     002007
      NOMW = NMOW(I+2,N)                                                002008
      IF (IMO(3,NOMW) .EQ. 0) GO TO 15                                  002009
      NMWVS = NMWVS + 1                                                 002010
      Y(LTD+NMWVS) = AMO(1,NOMW)                                        002011
   15 CONTINUE                                                          002012
   10 CONTINUE                                                          002013
C                                                                       002014
      LBE = LOCU(2*NB+1) - 1                                            002015
      DO 20 J=1,NH                                                      002016
      DO 20 I=1,6                                                       002017
      IP1 = I + 1                                                       002018
      IF (IHDATA(IP1,J) .EQ. 1) GO TO 20                                002019
      LBE = LBE + 1                                                     002020
      Y(LBE) = BETAH(I,J)                                               002021
   20 CONTINUE                                                          002022
C                                                                       002023
      IF (NDELTA .EQ. 0) GO TO 25                                       002024
      LDE  = LOCU(2*NB+2) - 1                                           002025
      DO 26 J=1,NDELTA                                                  002026
      L = LDE + J                                                       002027
   26 Y(L) = CNTDTA(J)                                                  002028
C                                                                       002029
   25 CALL ROTDH                                                        002030
      CALL BHGENR                                                       002031
```

```
      CALL FINDU (0)                                                    002032
C                                                                       002033
      IPRNT = 0                                                         002034
      IPLOT = 0                                                         002035
C                                                                       002036
      CALL YDOT                                                         002037
      WRITE (NOT,2001)                                                  002038
      CALL WRITIS (INDEP,1,NEQ,1)                                       002039
C                                                                       002040
      NCAM = NEQ                                                        002041
      NRAM = NCAM + NAUX                                                002042
C                                                                       002043
      CALL ENGMOM                                                       002044
      CALL PRNTOU                                                       002045
      IF (IFLNER .EQ. 1) GO TO 50                                       002046
      CALL PLOTWR                                                       002047
   11 CALL RKADAM(NEQ)                                                  002048
      CALL ENGMOM                                                       002049
      IPRNT = IPRNT + 1                                                 002050
      IF (IPRNT .NE. NOPRNT) GO TO 12                                   002051
      CALL PRNTOU                                                       002052
      IPRNT = 0                                                         002053
   12 CONTINUE                                                          002054
      IPLOT = IPLOT + 1                                                 002055
      IF (IPLOT .NE. NOPLOT) GO TO 13                                   002056
      CALL PLOTWR                                                       002057
      IPLOT = 0                                                         002058
   13 CONTINUE                                                          002059
      IF (T .LT. ENDT) GO TO 11                                         002060
      RETURN                                                            002061
C                                                                       002062
   50 CALL LINEAR (NRAM,NCAM)                                           002063
      RETURN                                                            002064
C                                                                       002065
  999 WRITE (NOT,2010)                                                  002066
 2010 FORMAT (1H1,15HERROR IN ITOPOL)                                   002067
      STOP                                                              002068
      END                                                               002069
```

```
[HDG,P     DYNSCC                                                          -002070
[FOR,IS    DYNSCC                                                          -002071
      COMPILER (XM=1),(EQUIV=CMN)                                          -002072
      SUBROUTINE DYNSCC                                                    -002073
C                                                                          -002074
C ***                                                                      -002075
C *** MSFC UNIVAC 1108 VERSION ***                                         -002076
C ***                                                                      -002077
      DOUBLE PRECISION DUM                                                 -002078
C                                                                          -002079
      COMMON / INTGR1 /  ZP(1000,16),DUM(1500),JVPL(16)                    -002080
      DIMENSION ICTITL(10),NCD( 3),PTITLE( 8)                              -002081
C                                                                          -002082
            COMMON /PLTDTA/                                                -002083
     *      NRPLOT,NCPLOT                                                 -002084
            COMMON /TAPENO/                                                -002085
     *      NTAPE1,NTAPE2,NTAPE3                                          -002086
C                                                                          -002087
      DATA NIT,NOT /5,6/                                                   -002088
      DATA KRPLOT,KCPLOT /1000,16/                                         -002089
C                                                                          -002090
      READ (NIT,1005) (ICTITL(I),I=1,10)                                   -002091
 1005 FORMAT(10A6)                                                         -002092
      CALL PAGEHD                                                          -002093
      WRITE (NOT,1001) (ICTITL(I),I=1,10)                                  -002094
 1001 FORMAT (///,30X,31HSUMMARY OF PLOTTING INFORMATION//,10X,10A6)       -002095
C                                                                          -002096
      READ (NIT,1003) NSET                                                 -002097
 1003 FORMAT(16I5)                                                         -002098
      IF (NSET .EQ. 0) RETURN                                              -002099
      WRITE (NOT,1011) NSET,NRPLOT,NCPLOT,KRPLOT,KCPLOT                    -002100
 1011 FORMAT(//,10X,10HNSET     =I5,/,                                     -002101
     *          10X,10HNRPLOT   =I5,  10X,10HNCPLOT   =I5,/,             -002102
     *          10X,10HKRPLOT   =I5,  10X,10HKCPLOT   =I5 )              -002103
C                                                                          -002104
      IF (NRPLOT .LE. KRPLOT) GO TO 1500                                   -002105
      NRPLOT = KRPLOT                                                      -002106
      WRITE (NOT,1009)                                                     -002107
 1009 FORMAT (//,10X,46HNRPLOT EXCEEDED KRPLOT AND WAS RESET TO KRPLOT)   -002108
 1500 CONTINUE                                                             -002109
C                                                                          -002110
      DO 1000 ISET=1,NSET                                                  -002111
      REWIND NTAPE3                                                        -002112
      READ (NIT,1003) JPL                                                  -002113
      IF (JPL .GT. KCPLOT) GO TO 998                                       -002114
      READ (NIT,1003) (JVPL(J),J=1,JPL)                                    -002115
      WRITE (NOT,1012) ISET,(JVPL(J),J=1,JPL)                              -002116
 1012 FORMAT(//,10X,7HISET  =  I5,/10X,7HJVPL  =16I5)                      -002117
C                                                                          -002118
      DO 2000 II=1,NRPLOT                                                  -002119
```

```
      READ (NTAPE3) (DUM(I),I=1,NCPLOT)                                 -002120
      DO 2001 J=1,JPL                                                   -002121
      JC = JVPL(J)                                                      -002122
 2001 ZP(II,J) = DUM(JC)                                                -002123
 2000 CONTINUE                                                          -002124
C                                                                       -002125
   20 READ (NIT,1003)  NCI,(NCD(I),I=1,3),NGRID                         -002126
      IF (NCI .EQ. 0) GO TO 1000                                        -002127
      IF(NCI .GT. 1) NGRID = 1                                          -002128
      IF(NGRID .EQ. 0) NGRID = 1                                        -002129
      READ (NIT,1004) TITLI,TITLD,(PTITLE(I),I=1, 8)                    -002130
 1004 FORMAT(2(A6,4X),8A6)                                              -002131
      WRITE(NOT,1006) NCI,(NCD(I),I=1,3),NGRID,                         -002132
     *              TITLI,TITLD,(PTITLE(I),I=1,8)                       -002133
 1006 FORMAT(//,15X,7HNCI    =,I5,5X,7HNCD    =,3I5,5X,7HNGRID =,I5,/,  -002134
     *              15X,A6,5X,A6,5X,8A6)                                -002135
      CALL PLTCAR(ZP,NCI,NCD,NRPLOT,NGRID,                              -002136
     *   TITLI,TITLD,PTITLE,ICTITL,KRPLOT)                              -002137
      GO TO 20                                                          -002138
C                                                                       -002139
 1000 CONTINUE                                                          -002140
C                                                                       -002141
      RETURN                                                            -002142
C                                                                       -002143
  998 WRITE (NOT,1020)                                                  -002144
 1020 FORMAT(//,10X,34HERROR IN PLOT INPUT DATA, STOP RUN)              -002145
      STOP                                                              -002146
C                                                                       -002147
      END                                                               -002148
```

```
[HDG,P    DYNSDD                                                        -002149
[FOR,IS   DYNSDD                                                        -002150
      COMPILER (XM=1),(EQUIV=CMN)                                       -002151
      SUBROUTINE DYNSDD                                                 -002152
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -002153
      REAL    FMIN,FMAX,DBMIN,DBMAX,TITLE                                002154
      REAL    AMIN,AMAX                                                  002155
C                                                                        002156
C                                                                        002157
C  THIS OVERLAY PERFORMS THE LINEARIZED SYSTEM ANALYSES.                 002158
C                                                                        002159
C                                                                        002160
C                                                                        002161
C          ----------DATA STREAM CONTROL----------                      002162
C                     FOR THIS OVERLAY                                   002163
C                                                                        002164
C-----LNAM                                          FORMAT A4            002165
C                                                                        002166
C     IF (LNAM .EQ. 4H    ) RETURN                                       002167
C     IF (LNAM .EQ. 4HTIME) GO TO 400                                    002168
C                                                                        002169
C-----CALL READIM (LRY,10,NCYC,10,KR)                                    002170
C                                                                        002171
C                  NOTE-----LRY(1,J) = ITYPE                             002172
C                           LRY(2,J) = ITFIN                             002173
C                           LRY(3,J) = JTFOUT                            002174
C                           LRY(4,J) = KPLOT                             002175
C                           LRY(5,J) = IAFLG                             002176
C                           LRY(6,J) = NO. B"S TO KEEP --ITYPE=7         002177
C                           LRY(7,J) = LOCAL ID. NO. OF B"S TO RETAIN.   002178
C                           LRY(8,J) = LOCAL ID. NO. OF B"S TO RETAIN.   002179
C                           LRY(9,J) = LOCAL ID. NO. OF B"S TO RETAIN.   002180
C                                                                        002181
C                                                                        002182
C                                                                        002183
C-----CALL READIM (IRY,3,NCYC,3,KR)                                      002184
C                                                                        002185
C                  NOTE-----IRY(1,J) = ROOT TOLERANCE EXPONENT.          002186
C                           IRY(2,J) = GAIN TOLERANCE EXPONENT           002187
C                           IRY(3,J) = ROOT TOLERANCE EXPONENT           002188
C                                      USED TO REMOVE SHIFT FREQ.        002189
C                                      IN SUBROUTINE NUMS.               002190
C                                                                        002191
C     DO 500 ICYC = 1,NCYC                                               002192
C                                                                        002193
C-----TITLE                                         FORMAT (10A6)        002194
C-----LPNAME                                        FORMAT (5A4)         002195
C                                                                        002196
C     DO 500 IOP = 1,5                                                   002197
C                                                                        002198
```

```
C                                                                       002199
C      IF  (LPNAME(IOP) .EQ. 4H    ) GO TO 500                          002200
C      IF  (LPNAME(IOP) .EQ. 4HBODE                                     002201
C     *.OR. LPNAME(IOP) .EQ. 4HNICH                                     002202
C     *.OR. LPNAME(IOP) .EQ. 4HNYQU                                     002203
C     *.OR. LPNAME(IOP) .EQ. 4HBONN                                     002204
C     *.OR. LPNAME(IOP) .EQ. 4HNINY) GO TO 200                          002205
C      IF  (LPNAME(IOP) .EQ. 4HROOT) GO TO 300                          002206
C                                                                       002207
C      GO TO 9999                                                       002208
C                                                                       002209
C 200 CONTINUE                                                          002210
C                                                                       002211
C              -----BODE, NICHOLS, NYQUIST SECTION-----                 002212
C                                                                       002213
C-----FMIN, FMAX, DBMIN, DBMAX, AMIN, AMAX         FORMAT (6F10.0)      002214
C      GO TO 500                                                        002215
C                                                                       002216
C 300  CONTINUE                                                         002217
C               -----ROOT LOCUS SECTION-----                            002218
C                                                                       002219
C-----CALL READIM (IJM, 2, NRLC, 2, KR)                                 002220
C-----CALL READ   (RLC, 7, NRLC, KR, KR)                                002221
C                                                                       002222
C      DO 350 IRC =1,NRLC                                               002223
C                                                                       002224
C 350 CONTINUE                                                          002225
C      GO TO 500                                                        002226
C                                                                       002227
C 400 CONTINUE                                                          002228
C                                                                       002229
C 500 CONTINUE                                                          002230
C      RETURN                                                           002231
C                                                                       002232
C              -----DIMENSIONED WORK SPACES-----                        002233
C                                                                       002234
C                                                                       002235
       DIMENSION VS1(214),VS2(107)                                     41202236
       DIMENSION LPNAME(5)                                              002237
       DIMENSION GNB(2),GDB(2)                                          002238
       DIMENSION RRN( 50),RIN( 50),RRD( 50),RID( 50),R2R( 50),R2I( 50)  41402239
       DIMENSION IJM(2, 50),LRY(9, 50), IRY(3, 50), KBKP(3)             41802240
C                                                                       002241
C              -----COMMON BLOCKS-----                                  002242
C                                                                       002243
C                                                                       002244
       COMMON /KDSIZE/                                                  002245
      1               KR, KRT, KRX, KV1, KV2, KVX                       002246
       COMMON /LDSIZE/                                                  002247
      2               NX, NY, NDLTA, NXSS, NB, NJQ, NY2, ND2            002248
```

```
      COMMON /LCOUNT/                                                    002249
     3               NNR, ICN, NNZ, NDR, ICD, NDZ                        002250
      COMMON /TAPENO/                                                    002251
     4               NUT1, NUT2, NUT3                                    002252
      COMMON /MISCNO/                                                    002253
     5               NOPRNT, NOPLOT                                      002254
      COMMON /LBDATA/                                                    002255
     6               FMIN, FMAX, DBMIN, DBMAX                            002256
      COMMON /LTOL  /                                                    002257
     7               TOLN,TOLD                                           002258
      COMMON /LTITLE/                                                    002259
     8               TITLE(10)                                           002260
      COMMON /LIJV  /                                                    002261
     9                IV( 50)  , JV  ( 50)                             42002262
      COMMON /LRARAY/                                                    002263
     A               FBRN( 50), FBNC( 50), FBRD( 50), FBDC( 50)        42202264
      COMMON /LROOT /                                                    002265
     B               R(107)   , RX  (214)                              42402266
      COMMON /  LV1 /                                                    002267
     C               V1  ( 50), V2  ( 50), V3  ( 50)                   42602268
      COMMON /  LV2 /                                                    002269
     D               XV1 ( 50), XV2 ( 50), XV3 ( 50), XV4 ( 50)        42802270
      COMMON /VECTOR/                                                    002271
     E               Y   (250), YD  (250)                              43002272
      COMMON /LWORK1/                                                    002273
     F               W1( 50, 50), W2( 50, 50)                          43202274
      COMMON /TIMESS/                                                    002275
     G               ST, DT, T, ET, TMST                                 002276
      COMMON /PLTDTA/                                                    002277
     I                NRPLOT,NCPLOT                                     002278
C                                                                        002279
C                                                                        002280
C        -----EQUIVALENCE MAP-----                                       002281
C                                                                        002282
C                                                                        002283
C                                                                        002284
      EQUIVALENCE (VS1(108),VS2(1))                                     43402285
C                                                                        002286
C                                                                        002287
C        -----DATA STATEMENTS-----                                       002288
C                                                                        002289
C                                                                        002290
      DATA  LBODE, LROOT, LNYQU, LNICH, LNINY, LTIME, LBLNK, LBONN/      002291
     *     4HBODE,4HROOT,4HNYQU,4HNICH,4HNINY,4HTIME,4H     ,4HBONN/     002292
C                                                                        002293
      DATA NIT/ 5/                                                       002294
      DATA NOT/ 6 /                                                      002295
C                                                                        002296
 1003 FORMAT (20A4)                                                      002297
 1004 FORMAT (6F10.0)                                                    002298
```

```
 1005 FORMAT(10A6)                                                      02299
 2001 FORMAT (//,5X,10HON ICYC = ,I2,26H,NUMERATOR GAIN LESS THAN ,     002300
     *          D10.4,16HWAS ENCOUNTERED.,//,5X,                       002301
     *          47HPROGRAM CONTINUING WITH NEXT TRANSFER FUNCTION.)    002302
C                                                                       002303
C                                                                       002304
C        -----SET UP FIXED INTEGER DATA-----                            002305
C                                                                       002306
      KR  =  50                                                        43602307
      KRT = 107                                                        43802308
      KRX = 214                                                        44002309
      KV1 = 214                                                        44202310
      KV2 = 107                                                        44402311
      KVX =  50                                                        44602312
      K3 = 3                                                            002313
      K9 = 9                                                            002314
C                                                                       002315
C        -----SET UP VARIABLE INTEGER DATA-----                         002316
C                                                                       002317
      NY2 = NX - NDLTA - NXSS                                           002318
      ND2 = NDLTA - NB                                                  002319
C                                                                       002320
      REWIND NUT2                                                       002321
      REWIND NUT3                                                       002322
C                                                                       002323
C          READ IN LINEARIZED PARTIAL DERIVATIVE MATRIX FROM UNIT NUT2  002324
C          PERFORM SIMILARITY TRANSFORMATION, AND PUT A* BACK ON NUT2.   002325
C                                                                       002326
      DO 50 L=1,NX                                                      002327
   50 READ (NUT2) (W1(I,L),I=1,NJQ)                                     002328
      REWIND NUT2                                                       002329
      WRITE (NUT2) ((W1(I,J),I=1,KR),J=1,KR)                            002330
      REWIND NUT2                                                       002331
      CALL WRITE (W1,NJQ,NX,3H-A-,KR)                                   002332
      CALL QRDRVR (W1,MX,RRD,RID,KR)                                    002333
      READ (NUT2)((W1(I,J),I=1,KR),J=1,KR)                              002334
      REWIND NUT2                                                       002335
C                                                                       002336
      CALL ASIMLR (W1,W2,IV,KR)                                         002337
      WRITE (NUT2) ((W2(I,J),I=1,KR),J=1,KR)                            002338
      REWIND NUT2                                                       002339
      CALL WRITE (W2,NX,NX,4H-A*-,KR)                                   002340
      CALL QRDRVR (W2,NX,RRN,RIN,KR)                                    002341
C  REMOVE NUMBERS SMALLER THAN 1.D-5 FROM ROOT ARRAYS.                  002342
C                                                                       002343
      CALL SIFT (RRD,NX,1.D-5)                                          002344
      CALL SIFT (RID,NX,1.D-5)                                          002345
      CALL SIFT (RRN,NX,1.D-5)                                          002346
      CALL SIFT (RIN,NX,1.D-5)                                          002347
C                                                                       002348
```

```
      CALL RWRITE (2,RRD,RID,RRN,RIN,NX,NX,4HRT A,4HRTA*)                  002349
      READ (NUT2)((W2(I,J),I=1,KR),J=1,KR)                                 002350
      REWIND NUT2                                                          002351
C                                                                          002352
C  READ IN CONTROL VARIABLE, LNAM AND BRANCH TO APPROIATE SECTION.         002353
      READ (NIT,1003) LNAM                                                 002354
C                                                                          002355
      IF (LNAM .EQ. LBLNK) RETURN                                          002356
      IF (LNAM .EQ. LTIME) GO TO 400                                       002357
C                                                                          002358
C            -----FREQUENCY DOMAIN ANALYSIS SECTION-----                   002359
C                                                                          002360
C                                                                          002361
C  READ IN FREQUENCY ANALYSIS CONTROL VARIABLES                            002362
      CALL READIM (LRY,N5,NCYC,K9,KR)                                      002363
C                                                                NERROR = 0  002364
C                                                                          002365
      CALL READIM (IRY,N3,NC2,K3,KR)                                       002366
C                                                                          002367
      IF (NC2 .NE. NCYC) GO TO 9999                                        002368
C                                                                          002369
      DO 501 ICYC = 1,NCYC                                                 002370
C                                                                          002371
      ITYPE = LRY(1,ICYC)                                                  002372
      ITFIN = LRY(2,ICYC)                                                  002373
      JTFOUT = LRY(3,ICYC)                                                 002374
      KPLOT = LRY(4,ICYC)                                                  002375
      IAFLG = LRY(5,ICYC)                                                  002376
                                                                NERROR = 1  002377
      IF (ITYPE .EQ. 0) GO TO 501                                          002378
C                                                                          002379
      IF (IABS(ITYPE) .LE. 6) GO TO 55                                     002380
      NBKP = LRY(6,ICYC)                                                   002381
      DO 54 I=1,NBKP                                                       002382
   54 KBKP(I) = LRY(I+6,ICYC)                                              002383
   55 CONTINUE                                                             002384
C                                                                          002385
C                                                                          002386
C  SET UP DEFAULT VALUES IN ARRAY IRY.                                     002387
C                                                                          002388
      IF (IRY(1,ICYC) .EQ. 0) IRY(1,ICYC) = -7                             002389
      IF (IRY(2,ICYC) .EQ. 0) IRY(2,ICYC) = -7                             002390
      IF (IRY(3,ICYC) .EQ. 0) IRY(3,ICYC) = -7                             002391
C                                                                          002392
      TOLN = 10.D0 ** IRY(1,ICYC)                                          002393
      GTOL = 10.D0 ** IRY(2,ICYC)                                          002394
      PTOL = 10.D0 ** IRY(3,ICYC)                                          002395
C                                                                          002396
      TOLD = TOLN                                                          002397
C                                                                          002398
```

```
C                                                                       002399
      IF (IABS(ITYPE) .LT. 1  .OR.  ITYPE .GT. 7) GO TO 9999            002400
C                                                                       002401
      READ (NIT,1005) TITLE                                             002402
C                                                                       002403
      READ (NUT2) ((W2(I,J),I=1,KR),J=1,KR)                             002404
      REWIND NUT2                                                       002405
C                                                                       002406
      CALL TFTYPE (W2,W1,V1,NX,NA,ITYPE,ITFIN,NBKP,KBKP,KR,KR)          002407
      CALL WRITE (W1,NA,NA,4H-AR-,KR)                                   002408
      CALL WRITE (V1,1,NA,4HBCOL,1)                                     002409
      WRITE (NUT3) ((W1(I,J),I=1,KR),J=1,KR)                            002410
      REWIND NUT3                                                       002411
C                                                                       002412
C  NOW HAVE REDUCED TF A* ON NUT3.                                      002413
C                                                                       002414
C  OBTAIN ROOTS FOR DENOMINATOR.                                        002415
C                                                                       002416
      DO 57 I=1,NA                                                      002417
      DO 57 J=1,NA                                                      002418
   57 W2(I,J) = W1(J,I)                                                 002419
C                                                                       002420
      CALL QRDRVR (W1,NA,RRD,RID,KR)                                    002421
      CALL SIFT (RRD,NA,TOLD)                                           002422
      CALL SIFT (RID,NA,TOLD)                                           002423
C                                                                       002424
      CALL QRDRVR (W2,NA,V2,V3,KR)                                      002425
      CALL SIFT (V2,NA,TOLD)                                            002426
      CALL SIFT (V3,NA,TOLD)                                            002427
      CALL RWRITE (2,RRD,RID,V2,V3,NA,NA,4HR AR,4HRART)                 002428
C                                                                       002429
      IF (IAFLG .EQ. 0) GO TO 59                                        002430
      DO 58 I=1,NA                                                      002431
      RRD(I) = V2(I)                                                    002432
   58 RID(I) = V3(I)                                                    002433
   59 CONTINUE                                                          002434
C                                                                       002435
C  OBTAIN ROOTS OF NUMERATOR.                                           002436
C                                                                       002437
      READ (NUT3) ((W1(I,J),I=1,KR),J=1,KR)                             002438
      REWIND NUT3                                                       002439
C  OBTAIN ABSOLUTE JTF LOCATION BASED UPON LOCAL JTFOUT.                002440
      JTF = NY2 + JTFOUT                                                002441
      IF (IABS(ITYPE) .EQ. 2) JTF = ND2 + JTFOUT                        002442
      IF (IABS(ITYPE) .EQ. 3) JTF = JTF + NXSS + ND2                    002443
      IF (IABS(ITYPE) .EQ. 7) JTF = JTF + NXSS + ND2                    002444
      CALL NUMS (W1,W2,V1,RRN,RIN,R2R,R2I,PTOL,                         002445
     *           GAIN,IFLG,NNUM,NZRO,JTF,NA,KR)                         002446
C                                                                       002447
      NN = NZRO                                                         002448
```

```
      ND = NA                                                          002449
C                                                                      002450
      IF (IFLG .NE. 0) GO TO 65                                        002451
      CALL PAGEHD                                                      002452
      WRITE (NOT,2001) ICYC,GTOL                                       002453
      GO TO 75                                                         002454
   65 CONTINUE                                                         002455
C                                                                      002456
      CALL SIFT (RRN,NN,TOLN)                                          002457
      CALL SIFT (RIN,NN,TOLN)                                          002458
      CALL DCQRRT (RRN,RIN,NN,NNR,ICN,NNZ,V1,V2)                       002459
      CALL DFORMB (NNR,ICN,V1,V2,FBRN,FBNC,V3,1.D0,GAIN,GNB)           002460
C                                                                      002461
   70 CONTINUE                                                         002462
C                                                                      002463
      CALL RWRITE (2,RRN,RIN,RRD,RID,NN,ND,4H NUM,4H DEN)              002464
C                                                                      002465
      CALL DCQRRT (RRD,RID,ND,NDR,ICD,NDZ,V1,V2)                       002466
      CALL DFORMB (NDR,ICD,V1,V2,FBRD,FBDC,V3,1.D0,1.D0,GDB)           002467
      GB = GNB(2)/GDB(2)                                               002468
C                                                                      002469
C-----ESTABLISH WORKING ROOT COUNTS.                                   002470
C                                                                      002471
      KNR = NNR                                                        002472
      KDR = NDR                                                        002473
      KNZ = NNZ                                                        002474
      KDZ = NDZ                                                        002475
      KCN = ICN                                                        002476
      KCD = ICD                                                        002477
C                                                                      002478
      CALL ZERO (XV1,1,KVX,1)                                          002479
      CALL ZERO (XV2,1,KVX,1)                                          002480
      CALL ZERO (XV3,1,KVX,1)                                          002481
      CALL ZERO (XV4,1,KVX,1)                                          002482
C                                                                      002483
      IF (KNR .EQ. 0) GO TO 2202                                       002484
      DO 201 I=1,KNR                                                   002485
  201 XV1(I) = FBRN(I)                                                 002486
 2202 CONTINUE                                                         002487
      IF (KDR .EQ. 0) GO TO 2203                                       002488
      DO 202 I=1,KDR                                                   002489
  202 XV3(I) = FBRD(I)                                                 002490
 2203 CONTINUE                                                         002491
      IF (KCN .EQ. 0) GO TO 2204                                       002492
      K=2*KCN                                                          002493
      DO 203 I=1,K                                                     002494
  203 XV2(I) = FBNC(I)                                                 002495
 2204 CONTINUE                                                         002496
      IF (KCD. EQ. 0) GO TO 205                                        002497
      K=2*KCD                                                          002498
```

```
      DO 204 I=1,K                                                      002499
  204 XV4(I) = FBDC(I)                                                  002500
  205 CONTINUE                                                          002501
C                                                                       002502
C-----REMOVE REAL ZEROS PRIOR TO CALL TO TTFF                           002503
      IF (KNR .NE. 0) CALL RMVZRO (XV1,KNR)                             002504
      IF (KDR .NE. 0) CALL RMVZRO (XV3,KDR)                             002505
      CALL ZERO (R,1,KRT,1)                                             002506
C                                                                       002507
      CALL TTFF (KNR,KCN,KNZ,KDR,KCD,KDZ,                               002508
     *           GB,XV1,XV2,XV3,XV4,R,KRT)                              002509
C                                                                       002510
      CALL CANCOR (R)                                                   002511
C                                                                       002512
C                                                                       002513
C  READ IN DISPLAY CONTROL VARIABLES.                                   002514
C                                                                       002515
   75 READ (NIT,1003) LPNAME                                            002516
C                                                                       002517
      DO 500 IOP = 1,5                                                  002518
C                                                                       002519
      IF  (LPNAME(IOP) .EQ. LBLNK ) GO TO 500                           002520
      IF  (LPNAME(IOP) .EQ. LBODE                                       002521
     *.OR. LPNAME(IOP) .EQ. LNICH                                       002522
     *.OR. LPNAME(IOP) .EQ. LNYQU                                       002523
     *.OR. LPNAME(IOP) .EQ. LBONN                                       002524
     *.OR. LPNAME(IOP) .EQ. LNINY ) GO TO 200                           002525
      IF  (LPNAME(IOP) .EQ. LROOT ) GO TO 300                           002526
                                                    NERROR = 2          002527
      GO TO 9999                                                        002528
C                                                                       002529
  200 CONTINUE                                                          002530
C                                                                       002531
C              -----FREQUENCY RESPONSE PROCESSING-----                  002532
C                                                                       002533
      READ (NIT,1004) FMIN, FMAX, DBMIN, DBMAX, AMIN, AMAX              002534
      IF (IFLG .EQ. 0) GO TO 500                                        002535
C                                                                       002536
      KNR = R(1) + 0.1D0                                                002537
      KCN = R(2) + 0.1D0                                                002538
      KNZ = R(3) + 0.1D0                                                002539
      KDR = R(4) + 0.1D0                                                002540
      KCD = R(5) + 0.1D0                                                002541
      KDZ = R(6) + 0.1D0                                                002542
      CALL WRITE (R,1,KRT,4HRRED,1)                                     002543
C                                                                       002544
      CALL ZERO (XV1,1,KVX,1)                                           002545
      CALL ZERO (XV2,1,KVX,1)                                           002546
      CALL ZERO (XV3,1,KVX,1)                                           002547
      CALL ZERO (XV4,1,KVX,1)                                           002548
```

```
C                                                                       002549
      IF (KNR .EQ. 0) GO TO 2207                                        002550
      DO 206 I=1,KNR                                                    002551
      L=7+I                                                             002552
  206 XV1(I) = R(L)                                                     002553
 2207 CONTINUE                                                          002554
      IF (KCN .EQ. 0) GO TO 2208                                        002555
      K=2*KCN                                                           002556
      DO 207 I=1,K                                                      002557
      L=7+KNR+I                                                         002558
  207 XV2(I) = R(L)                                                     002559
 2208 CONTINUE                                                          002560
      IF (KDR .EQ. 0) GO TO 2209                                        002561
      DO 208 I=1,KDR                                                    002562
      L = 7+KNR+2*KCN+I                                                 002563
  208 XV3(I) = R(L)                                                     002564
 2209 CONTINUE                                                          002565
      IF (KCD .EQ. 0) GO TO 2210                                        002566
      K=2*KCD                                                           002567
      DO 209 I=1,K                                                      002568
      L=7+KNR+2*KCN+KDR+I                                               002569
  209 XV4(I) = R(L)                                                     002570
 2210 CONTINUE                                                          002571
C                                                                       002572
C-----EXTEND REAL ARRAY COUNTS TO INCLUDE REAL ZEROS.                   002573
      KNR = KNR + KNZ                                                   002574
      KDR = KDR + KDZ                                                   002575
C                                                                       002576
C-----PERFORM THE FREQUENCY RESPONSE.                                   002577
C                                                                       002578
      CALL SFREQ2 (KNR,KCN,KDR,KCD,GB,                                  002579
     1             XV1,XV2,XV3,XV4,FMIN,FMAX,TITLE)                     002580
C                                                                       002581
      IF (KPLOT .EQ. 0) GO TO 220                                       002582
      IF (LPNAME(IOP) .EQ. LBODE                                        002583
     * .OR.LPNAME(IOP) .EQ. LBONN)                                     002584
     *                  CALL SPLOT (TITLE,FMAX,FMIN,DBMIN,DBMAX)       002585
C                                                                       002586
      IF  (LPNAME(IOP) .EQ. LNICH                                       002587
     * .OR.LPNAME(IOP) .EQ. LNINY                                      002588
     *.OR.LPNAME(IOP) .EQ. LBONN) CALL NIPLOT (TITLE,DBMIN,DBMAX)       002589
C                                                                       002590
      IF  (LPNAME(IOP) .EQ. LNYQU                                       002591
     * .OR.LPNAME(IOP) .EQ. LBONN                                      002592
     * .OR.LPNAME(IOP) .EQ. LNINY) CALL NYPLOT (TITLE,AMIN,AMAX)       002593
C                                                                       002594
      GO TO 500                                                         002595
  220 CALL PAGEHD                                                       002596
      WRITE (NOT,221) ICYC, LPNAME(IOP)                                 002597
  221 FORMAT (//,10X,                                                   002598
```

```
     *  48HNO FREQUENCY RESPONSE PLOTS GENERATED ON ICYC = I3,          002599
     *  //,10X,9HLPNAME = A4)                                          002600
      GO TO 500                                                        002601
  300 CONTINUE                                                         002602
C                                                                      002603
C              -----ROOT LOCUS SECTION-----                            002604
C                                                                      002605
C                                                                      002606
      CALL RTOP (R,RX,VS1,KRX)                                         002607
      NP = RX(1) + 1.D0                                                002608
      NQ = RX(2) + 1.D0                                                002609
C                                                                      002610
      DO 320 I=1,NP                                                    002611
      J=I+2                                                            002612
  320 XV1(I) = RX(J)                                                   002613
      DO 325 I=1,NQ                                                    002614
      J = I+2+NP                                                       002615
  325 XV2(I) = RX(J)                                                   002616
      CALL WRITE (XV2,1,NQ,4HPDEN,1)                                   002617
      CALL WRITE (XV1,1,NP,4HPNUM,1)                                   002618
C                                                                      002619
C                                                                      002620
C     READ IN ROOT LOCUS CONTROL VARIABLES.                            002621
C                                                                      002622
  330 CALL READIM (IJM,NR2,NRLC,2,KR)                                  002623
      IF (IFLG .EQ. 0) GO TO 340                                       002624
                                                    NERROR = 3         002625
      IF (NR2 .NE. 2  .OR.  NRLC .GT. KR) GO TO 9999                   002626
C                                                                      002627
C--------NOTE..... IJM(1,J) = ISNIM(J)                                002628
C                  IJM(2,J) = ELE. LOCATION IN ROOT ARRAY              002629
C                             FOR STARTING ROOT LOCUS.                 002630
C                                                                      002631
  340 CALL READ (W1,NR2,NC2,KR,KR)                                     002632
      IF (IFLG .EQ. 0) GO TO 500                                       002633
                                                    NERROR = 4         002634
      IF (NR2 .NE. 6  .OR.  NC2 .NE. NRLC) GO TO 9999                  002635
C                                                                      002636
C-----NOTE.....W1(1,J) = THETAO(J)                                     002637
C              W1(2,J) = SCL                                           002638
C              W1(3,J) = ALOC                                          002639
C              W1(4,J) = XMIN                                          002640
C              W1(5,J) = XMAX                                          002641
C              W1(6,J) = YMAX                                          002642
C                                                                      002643
      DO 350 IRC = 1,NRLC                                              002644
C                                                                      002645
C                                                                      002646
      ISNIM = IJM(1,IRC)                                               002647
      JJ    = IJM(2,IRC)                                               002648
```

```
C                                                                       002649
      THETAO = W1(1,IRC)                                                002650
      SCL    = W1(2,IRC)                                                002651
      ALOC   = W1(3,IRC)                                                002652
      XMIN   = W1(4,IRC)                                                002653
      XMAX   = W1(5,IRC)                                                002654
      YMAX   = W1(6,IRC)                                                002655
C                                                                       002656
C     LOCATE PROPER STARTING ROOT.                                      002657
C                                                                       002658
      IF (ISNIM .NE. 1) GO TO 341                                       002659
C                                                                       002660
C     ROOT IS AN OPEN LOOP ZERO.                                        002661
C                                                                       002662
      SR = RRN(JJ)                                                      002663
      SI = RIN(JJ)                                                      002664
      GO TO 342                                                         002665
  341 CONTINUE                                                          002666
C                                                                       002667
C     ROOT IS A POLE.                                                   002668
C                                                                       002669
      SR = RRD(JJ)                                                      002670
      SI = RID(JJ)                                                      002671
  342 CONTINUE                                                          002672
C                                                                       002673
      CALL RLOCUS (XV1,XV2,SCL,SR,SI,NP,NQ,THETAO,                      002674
     1             XMIN,XMAX,YMAX,ALOC)                                 002675
C                                                                       002676
      IF (KPLOT .EQ. 1) CALL RLPLOT (TITLE,ISNIM,ICYC,IRC)              002677
C                                                                       002678
  350 CONTINUE                                                          002679
C                                                                       002680
C                                                                       002681
      GO TO 500                                                         002682
  400 CONTINUE                                                          002683
C                                                                       002684
C             -----LINEARIZED TIME RESPONSE SECTION-----                002685
C                                                                       002686
      READ (NUT2) ((W1(I,J),I=1,KR),J=1,KR)                             002687
      REWIND NUT2                                                       002688
      CALL LTRESP                                                       002689
      RETURN                                                            002690
C                                                                       002691
  500 CONTINUE                                                          002692
  501 CONTINUE                                                          -002693
C                                                                       002694
C                                                                       002695
      RETURN                                                            002696
C                                                                       002697
 9999 WRITE (NOT,1999) NERROR                                           002698
```

```
 1999 FORMAT (1H1,//10X,44HERROR ENCOUNTERED IN DYNSDD, NERROR = ,        -002699
     *        I3,/10X,16HPROGRAM STOPPED.)                               002700
       STOP                                                             002701
C                                                                       002702
       END                                                              -002703
```

```
[HDG,P    DYNSEE                                                    -002704
[FOR,IS   DYNSEE                                                    -002705
      COMPILER (XM=1),(EQUIV=CMN)                                   -002706
      SUBROUTINE DYNSEE(IFLNER,NOPLOT)                              -002707
      DIMENSION ADARY(22)                                           -002708
      CALL IDENT(9,ADARY)                                           -002709
      IF(IFLNER .EQ. 1) CALL DYNSDD                                 -002710
      IF(NOPLOT .GT. 0) CALL DYNSCC                                 -002711
      CALL ENDJOB                                                   -002712
      RETURN                                                        -002713
      END                                                           -002714
```

```
[HDG,P     ENGMOM                                                                -002715
[FOR,IS    ENGMOM                                                                -002716
      COMPILER (XM=1),(EQUIV=CMN)                                                -002717
      SUBROUTINE ENGMOM                                                           002718
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                         -002719
C                                                                                 002720
            COMMON /BHBSRD/                                                       002721
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                       202722
            COMMON /INTGRL/                                                       002723
     *      AM( 78, 5),ACOF(9, 6, 5),BCOF(6, 6, 5),                             502724
     *      COF11( 6, 6, 5),COF22( 6, 6, 5),COF33( 6, 6, 5),AK( 6, 6, 5),       602725
     *      COF12( 6, 6, 5),COF13( 6, 6, 5),COF23( 6, 6, 5),AD( 6, 6, 5),       702726
     *      COFXY( 6, 6, 5),COFXZ( 6, 6, 5),COFYZ( 6, 6, 5)                     802727
            COMMON /MAXMUM/                                                       002728
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU                    002729
            COMMON /MOMENG/                                                       002730
     *      P( 65),PMOM(30),HTOT(3),TOTL(3),ENGKE( 5),ENGPE( 5),                1102731
     *      TOTKE, TOTPE, TOTENG, AHTOT,ATOTL                                   002732
            COMMON /NUMBRS/                                                       002733
     *      ZRO,ONE,TWO,TRES                                                    002734
            COMMON /SPECIF/                                                       002735
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),           1602736
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),                  1702737
     *      NB,NH,NSPT,NOFMC,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5),       1802738
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                                 1902739
            COMMON /VECTOR/                                                       002740
     *      Y(250),YDT(250)                                                     2002741
C                                                                                 002742
      DIMENSION WV( 6)                                                           8902743
      DIMENSION VW(3)                                                             002744
C                                                                                 002745
      KM = NMDBOD                                                                 002746
      DO 5 I=1,3                                                                  002747
      HTOT(I) = ZRO                                                               002748
    5 TOTL(I) = ZRO                                                               002749
      DO 10 N=1,NB                                                                002750
      LA = 6*N - 5                                                                002751
      LL = LA + 3                                                                 002752
      LOA = LOCU(N)                                                               002753
      LOL = LOCU(N) + 3                                                           002754
      CALL MULT3  (ROL(1,1,N),P(LOA),PMOM(LA),3,3,1,3,1,1)                       002755
      CALL MULT3  (ROL(1,1,N),P(LOL),PMOM(LL),3,3,1,3,1,1)                       002756
      PMOM(LA  ) = PMOM(LA  )+DOL(2,N)*PMOM(LL+2)-DOL(3,N)*PMOM(LL+1)             002757
      PMOM(LA+1) = PMOM(LA+1)+DOL(3,N)*PMOM(LL  )-DOL(1,N)*PMOM(LL+2)             002758
      PMOM(LA+2) = PMOM(LA+2)+DOL(1,N)*PMOM(LL+1)-DOL(2,N)*PMOM(LL  )             002759
CCC   STATEMENTS THRU 40 TO ACCOUNT FOR ANGULAR MOMENTUM DUE TO                   002760
CCC   CONSTANT SPEED MOMENTUM WHEELS.                                             002761
      NM = NMOW(1,N)                                                              002762
      IF (NM .EQ. 0) GO TO 40                                                     002763
      DO 30 I=1,NM                                                                002764
```

```
      NW = NMOW(2+I,N)                                                  002765
      IF (IMO(3,NW) .NE. 0) GO TO 30                                    002766
      NA = IMO(2,NW)                                                    002767
      NS = IMO(1,NW)                                                    002768
      PH = AMO(1,NW)*AMO(2,NW)                                          002769
      DO 35 J=1,3                                                       002770
   35 VW(J) = PH*BS(NA,J,NS)                                            002771
      CALL MULTAD (ROL(1,1,N),VW,PMOM(LA),3,3,1,3,1,1)                  002772
   30 CONTINUE                                                          002773
   40 CONTINUE                                                          002774
      DO 15 I=1,3                                                       002775
      I1 = LA - 1 + I                                                   002776
      I2 = LL - 1 + I                                                   002777
      HTOT(I) = HTOT(I) + PMOM(I1)                                      002778
   15 TOTL(I) = TOTL(I) + PMOM(I2)                                      002779
   10 CONTINUE                                                          002780
C                                                                       002781
      TOTKE = ZRO                                                       002782
      DO 20 N=1,NB                                                      002783
      LOU = LOCU(N)                                                     002784
      LO  = LOCU(N+NP)                                                  002785
      ENGPE(N) = ZRO                                                    002786
      LE  = LENU(N+NB)                                                  002787
      LEU = LENU(N)                                                     002788
      IF (LE .EQ. 0) GO TO 22                                           002789
      CALL MULT3  (AK(1,1,N),Y(LO),WV,LE,LE,1,KM,1,1)                   002790
      CALL MULT3  (Y(LO),WV,ENGPE(N),1,LE,1,1,1,1)                      002791
      ENGPE(N) = ENGPE(N)/TWO                                           002792
   22 CALL MULT3  (Y(LOU),P(LOU),ENGKE(N),1,LEU,1,1,1,1)                002793
      ENGKE(N) = ENGKE(N)/TWO                                           002794
      TOTKE = TOTKE + ENGKE(N)                                          002795
   20 TOTPE = TOTPE + ENGPE(N)                                          002796
      TOTENG = TOTKE + TOTPE                                            002797
      ATOTL = DSQRT(TOTL(1)**2 + TOTL(2)**2 + TOTL(3)**2)               002798
      AHTOT = DSQRT(HTOT(1)**2 + HTOT(2)**2 + HTOT(3)**2)               002799
C                                                                       002800
      RETURN                                                            002801
      END                                                               002802
```

```
[HDG,P     EQADD                                                          -002803
[FOR,IS    EQADD                                                          -002804
      COMPILER (XM=1),(EQUIV=CMN)                                         -002805
      SUBROUTINE EQADD                                                     002806
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                 -002807
C                                                                          002808
            COMMON /BHBSRD/                                              002809
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                202810
            COMMON /DNAUX /                                              002811
     *      NAUX                                                         002812
            COMMON /MAXMUM/                                              002813
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU            002814
            COMMON /SPECIF/                                              002815
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),   1602816
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),          1702817
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1802818
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                         1902819
            COMMON /VECTOR/                                              002820
     *      Y(250),YDT(250)                                             2002821
C                                                                          002822
      NAUX = 6                                                             002823
      LDEL = LOCU(2*NB+2) - 1                                              002824
      ACON = 57.2958D0                                                     002825
      YDT(NEQ+1) =  ACON*ROL(3,2,1)/ROL(3,3,1)                             002826
      YDT(NEQ+2) = -ACON*ROL(3,1,1)/ROL(3,3,1)                             002827
      YDT(NEQ+3) =  ACON*ROL(2,1,1)/ROL(2,2,1)                             002828
      YDT(NEQ+4) = Y(LDEL+2)                                               002829
      YDT(NEQ+5) = Y(LDEL+4)                                               002830
      YDT(NEQ+6) = Y(LDEL+6)                                               002831
      RETURN                                                               002832
      END                                                                  002833
```

```
[HDG,P     EXTOR                                                          -002834
[FOR,IS    EXTOR                                                          -002835
      COMPILER (XM=1),(EQUIV=CMN)                                         -002836
      SUBROUTINE EXTOR  (TEX,ISPN,NTEX)                                    002837
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                 -002838
      DIMENSION TEX(6,1), ISPN(1)                                          002839
C                                                                          002840
            COMMON /MAXMUM/                                                002841
     *     NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU               002842
            COMMON /SPECIF/                                                002843
     *     BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),      1602844
     *     DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),             1702845
     *     NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5),  1802846
     *     LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                            1902847
            COMMON /VECTOR/                                                002848
     *     Y(250),YDT(250)                                                2002849
C                                                                          002850
      DATA IIST / 0 /                                                      002851
C                                                                          002852
CCC ESTABLISH THE EXTERNAL FORCE/TORQUE (6-LONG VECTOR) AND NUMBER         002853
CCC  THE CORRESPONDING SENSOR POINTS. ALSO ESTABLISH THE NUMBER OF         002854
CCC  SIX-LONG VECTORS (NTEX).                                              002855
C                                                                          002856
      IF (IIST .EQ. 1) GO TO 5                                             002857
      IIST = 1                                                             002858
      DO 10 I=1,6                                                          002859
      DO 10 J=1,NSPMAX                                                     002860
   10 TEX(I,J) = 0.D 0                                                     002861
C                                                                          002862
    5 NTEX = 0                                                             002863
C                                                                          002864
      RETURN                                                               002865
      END                                                                  002866
```

```
[HDG,P     FETCH                                                          -002867
[FOR,IS    FETCH                                                          -002868
      COMPILER (XM=1),(EQUIV=CMN)                                         -002869
      SUBROUTINE FETCH(NTAPE,NRECN,NRECO,A,NRA,NCA,KRA)                    002870
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                  -002871
C                                                                          002872
C     FETCH MATRIX FROM NTAPE - ASSUMED WRITTEN BY ROWS                    002873
C         WHERE NRECN = INPUT RECORD NUMBER DESIRED                        002874
C               NRECO = OUTPUT RECORD NUMBER FETCHED                       002875
C               A     = ARRAY WHERE RECORD STORED (NRA,NCA)                002876
C                                                                          002877
      DIMENSION A(KRA,1)                                                   002878
C                                                                          002879
      IF(NRECN - NRECO) 1,2,3                                              002880
C                                                                          002881
    1 REWIND NTAPE                                                         002882
      NSKIP = NRECN - 1                                                    002883
      IF(NSKIP .EQ. 0) GO TO 20                                            002884
      DO 10 K=1,NSKIP                                                      002885
   10 READ(NTAPE) DUM                                                      002886
      GO TO 20                                                             002887
C                                                                          002888
    2 BACKSPACE NTAPE                                                      002889
      GO TO 20                                                             002890
C                                                                          002891
    3 NSKIP = NRECN - NRECO - 1                                            002892
      IF(NSKIP .EQ. 0) GO TO 20                                            002893
      DO 11 K=1,NSKIP                                                      002894
   11 READ(NTAPE) DUM                                                      002895
      GO TO 20                                                             002896
C                                                                          002897
   20 NRECO = NRECN                                                        002898
C                                                                          002899
      READ(NTAPE) ((A(I,J),J=1,NCA),I=1,NRA)                               002900
C                                                                          002901
      RETURN                                                               002902
      END                                                                  002903
```

```
[HDG,P     FINDT                                                          -002904
[FOR,IS    FINDT                                                          -002905
      COMPILER (XM=1),(EQUIV=CMN)                                         -002906
      SUBROUTINE FINDT (C,NCN,NX,NS,T,NRET,KC,KT)                          002907
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                  -002908
      DIMENSION C(KC,1), T(KT,1)                                           002909
      DIMENSION IVEC(107), JVEC(107)                                      44802910
C                                                                          002911
      DATA EPS , NOT / 1.D-15, 6 /                                         002912
 1001 FORMAT (/// 5X,36HSUBROUTINE FIND TERMINATED, SING. AT   I5)        002913
C                                                                          002914
      DO 5 I=1,NX                                                          002915
    5 JVEC(I) = 1                                                          002916
C                                                                          002917
      DO 10 L=1,NCN                                                        002918
      JBIG = 1                                                             002919
      A = DABS(C(L,1))                                                     002920
      DO 15 J=2,NS                                                         002921
      AT = DABS(C(L,J))                                                    002922
      IF (AT .LT. A) GO TO 15                                              002923
      A = AT                                                               002924
      JBIG = J                                                             002925
   15 CONTINUE                                                             002926
      IVEC(L) = -JBIG                                                      002927
      JVEC(JBIG) = 0                                                       002928
      IF (A .GT. EPS) GO TO 20                                             002929
      WRITE (NOT,1001) L                                                   002930
      STOP                                                                 002931
   20 CONTINUE                                                             002932
      CLJBIG = C(L,JBIG)                                                   002933
      DO 17 J=1,NX                                                         002934
   17 C(L,J) = C(L,J)/CLJBIG                                               002935
      DO 25 I=1,NCN                                                        002936
      F = C(I,JBIG)                                                        002937
      IF (I .EQ. L) GO TO 25                                               002938
      DO 30 J=1,NX                                                         002939
   30 C(I,J) = C(I,J) - F*C(L,J)                                           002940
   25 CONTINUE                                                             002941
   10 CONTINUE                                                             002942
C                                                                          002943
      NVAL = 0                                                             002944
      DO 40 I=1,NX                                                         002945
      IF (JVEC(I) .EQ. 0) GO TO 40                                         002946
      NVAL = NVAL + 1                                                      002947
      JVEC(I) = NVAL                                                       002948
   40 CONTINUE                                                             002949
C                                                                          002950
      NRET = NX - NCN                                                      002951
      CALL ZERO   (T,NRET,NRET,KT)                                         002952
      CALL REVADD (1.DO,C,IVEC,JVEC,T,NCN,NX,NRET,NRET,KC,KT)              002953
```

```
C                                                                       002954
      DO 50 I=1,NX                                                      002955
      IF (JVEC(I) .EQ. 0) GO TO 50                                      002956
      NR = JVEC(I)                                                      002957
      T(I,NR) = 1.DO                                                    002958
   50 CONTINUE                                                          002959
C                                                                       002960
      RETURN                                                            002961
      END                                                               002962
```

```
[HDG,P    FINDU                                                          -002963
[FOR,IS   FINDU                                                          -002964
      COMPILER (XM=1),(EQUIV=CMN)                                        -002965
      SUBROUTINE FINDU (IFLAG)                                            002966
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -002967
C                                                                         002968
      DIMENSION ICON(30), IVEC(30),JCON( 65),R(30)                       7402969
C                                                                         002970
            COMMON /AMUBW /                                               002971
     *      AMU(15,15, 5),BW(30, 65)                                      102972
            COMMON /BHESRD/                                               002973
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                202974
            COMMON /NUMBRS/                                               002975
     *      ZRO,ONE,TWO,TRES                                              002976
            COMMON /SPECIF/                                               002977
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),     1602978
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),            1702979
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1802980
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                           1902981
            COMMON /TIMESS/                                               002982
     *      STARTT,DELTAT,T,ENDT,TMST                                     002983
            COMMON /VECTOR/                                               002984
     *      Y(250),YDT(250)                                               2002985
            COMMON /VINDEP/                                               002986
     *      INDEP(250)                                                    2102987
C                                                                         002988
      DATA EPS,NOT / 1.D-06, 6/                                           002989
C                                                                         002990
      IF (IFLAG .EQ. 2) GO TO 100                                         002991
      IF (IFLAG .EQ. 1) GO TO 110                                         002992
      NCN = 6*NH                                                          002993
      DO 205 L=1,NCN                                                      002994
      IVEC(L) = 0                                                         002995
  205 ICON(L) = 1                                                         002996
      DO 207 J=1,NU                                                       002997
  207 JCON(J) = 0                                                         002998
      DO 210 N=1,NB                                                       002999
      JU = LOCU(N) - 1                                                    003000
      JRNG = 6                                                            003001
      IF (NH .GT. NB) JRNG = 6 + IRGFLX(N)                                003002
      DO 210 J=1,JRNG                                                     003003
  210 JCON(JU+J) = 1                                                      003004
      NR = 0                                                              003005
      DO 215 J=1,NH                                                       003006
      DO 215 I=1,6                                                        003007
      IP1 = I + 1                                                         003008
      NR = NR + 1                                                         003009
      R(NR) = BETAHD(I,J)                                                 003010
      IF (IHDATA(IP1,J) .EQ. 1) R(NR) = ZRO                               003011
      IF (IHDATA(IP1,J) .EQ. 2) R(NR) = ADT(NR,T)                         003012
```

```
      BETAHD(I,J) = R(NR)                                              003013
  215 CONTINUE                                                         003014
C                                                                      003015
      GO TO 150                                                        003016
  110 DO 220 L=1,NCN                                                   003017
      IVEC(L) = 0                                                      003018
  220 ICON(L) = 0                                                      003019
      DO 221 J = 1,NU                                                 -003020
  221 INDEP(J) = 1                                                    -003021
      NR = 0                                                           003022
      DO 225 J=1,NH                                                    003023
      DO 225 I=2,7                                                     003024
      NR = NR + 1                                                      003025
      ICON(NR) = IHDATA(I,J)                                           003026
  225 CONTINUE                                                         003027
  100 DO 230 L=1,NCN                                                   003028
      IF (ICON(L) .EQ. 0) GO TO 230                                    003029
      IF (ICON(L) .EQ. 1) R(L) = ZRO                                   003030
      IF (ICON(L) .EQ. 2) R(L) = ADT(L,T)                              003031
  230 CONTINUE                                                         003032
C                                                                      003033
  150 DO 310 I=1,NCN                                                   003034
      DO 310 J=1,NU                                                    003035
  310 BW(I,J) = ZRO                                                    003036
      LEQ = 6 + IRGFLX(1)                                              003037
      DO 315 I=1,6                                                     003038
      DO 315 J=1,LEQ                                                   003039
  315 BW(I,J) = BH(I,J,1)                                              003040
C                                                                      003041
      DO 320 L=2,NH                                                    003042
      LQ = 2*L - 2                                                     003043
      LP = LQ + 1                                                      003044
      NOBQ = ITOPOL(1,L)                                               003045
      NOBP = ITOPOL(2,L)                                               003046
      LH = 6*(L-1)                                                     003047
      LBQ = LOCU(NOBQ) - 1                                             003048
      LBP = LOCU(NOBP) - 1                                             003049
      LEQ = 6 + IRGFLX(NOBQ)                                           003050
      LEP = 6 + IRGFLX(NOBP)                                           003051
      DO 325 I=1,6                                                     003052
      DO 325 J=1,LEQ                                                   003053
  325 BW(I+LH,J+LBQ) = BH(I,J,LQ)                                      003054
      DO 330 I=1,6                                                     003055
      DO 330 J=1,LEP                                                   003056
  330 BW(I+LH,J+LBP) = BH(I,J,LP)                                      003057
  320 CONTINUE                                                         003058
C                                                                      003059
      DO 10 L=1,NCN                                                    003060
      IF (ICON(L) .EQ. 0) GO TO 10                                     003061
      IF (IFLAG .LT. 2) GO TO 400                                      003062
```

```
      JBIG = IVEC(L)                                                    003063
      A = DABS(BW(L,JBIG))                                              003064
      GO TO 410                                                         003065
  400 JBIG = 0                                                          003066
      A = ZRO                                                           003067
      DO 15 J=1,NU                                                      003068
      IF (JCON(J) .EQ. 0) GO TO 15                                      003069
      AT = DABS(BW(L,J))                                                003070
      IF (AT .LT. A) GO TO 15                                           003071
      A = AT                                                            003072
      JBIG = J                                                          003073
   15 CONTINUE                                                          003074
      IVEC(L) = JBIG                                                    003075
  410 IF (A .LE. EPS) GO TO 999                                         003076
      F = BW(L,JBIG)                                                    003077
      DO 17 J=1,NU                                                      003078
   17 BW(L,J) = BW(L,J)/F                                               003079
      R(L) = R(L)/F                                                     003080
      BW(L,JBIG) = ONE                                                  003081
      DO 25 I=1,NCN                                                     003082
      IF (I .EQ. L  .OR. ICON(I) .EQ. 0) GO TO 25                       003083
      F = BW(I,JBIG)                                                    003084
      DO 30 J=1,NU                                                      003085
   30 BW(I,J) = BW(I,J) - F*BW(L,J)                                     003086
      R(I) = R(I) - F*R(L)                                              003087
      BW(I,JBIG) = ZRO                                                  003088
   25 CONTINUE                                                          003089
   10 CONTINUE                                                          003090
C                                                                       003091
      DO 35 L=1,NCN                                                     003092
      LU = IVEC(L)                                                      003093
      IF (LU .EQ. 0) GO TO 35                                           003094
      Y(LU) = ZRO                                                       003095
   35 CONTINUE                                                          003096
C                                                                       003097
      DO 40 L=1,NCN                                                     003098
      IF (ICON(L) .EQ. 0) GO TO 40                                      003099
      DO 45 J=1,NU                                                      003100
   45 R(L) = R(L) - BW(L,J)*Y(J)                                        003101
   40 CONTINUE                                                          003102
      DO 50 L=1,NCN                                                     003103
      LU = IVEC(L)                                                      003104
      IF (LU .EQ. 0) GO TO 50                                           003105
      IF (IFLAG .EQ. 1) INDEP(LU) = 0                                   003106
      Y(LU) = R(L)                                                      003107
   50 CONTINUE                                                          003108
C                                                                       003109
      RETURN                                                            003110
  999 WRITE (NOT,1001)                                                  003111
 1001 FORMAT (1H1,25HSINGULAR EQUATIONS, FINDU)                         003112
```

```
      STOP                                                              003113
C                                                                       003114
      END                                                               003115
```

```
[HDG,P     FIT                                                          -003116
[FOR,IS    FIT                                                          -003117
       COMPILER (XM=1),(EQUIV=CMN)                                      -003118
       SUBROUTINE FIT(N,ARRAY,SLOPE)                                     003119
       DOUBLE PRECISION Y,SUMY,SUMIY,DLOG10,SLOPE,ARRAY(100)            -003120
C                                                                        003121
       M=0                                                               003122
       ISUM=0                                                            003123
       ISQSUM=0                                                          003124
       SUMY=0.D0                                                         003125
       SUMIY=0.D0                                                        003126
       DO 4 I=1,N                                                        003127
       IF(ARRAY(I))1,4,2                                                 003128
    1  Y=DLOG10(-ARRAY(I))                                               003129
       GO TO 3                                                           003130
    2  Y=DLOG10(ARRAY(I))                                                003131
    3  M=M+1                                                             003132
       ISUM=ISUM+I                                                       003133
       ISQSUM=ISQSUM+I*I                                                 003134
       SUMY=SUMY+Y                                                       003135
       SUMIY=SUMIY+I*Y                                                   003136
    4  CONTINUE                                                          003137
       SLOPE=(M*SUMIY-ISUM*SUMY)/(M*ISQSUM-ISUM*ISUM)                    003138
       RETURN                                                            003139
       END                                                               003140
```

```
[HDG,P     FORMB                                                          -003141
[FOR,IS    FORMB                                                          -003142
       COMPILER (XM=1),(EQUIV=CMN)                                        -003143
       SUBROUTINE FORMB (KR, KC, RLRT, CMPR, FBR, FBC)                     003144
       IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -003145
CFORMB        FACTORED  FORM  TIME CONSTANTS, DAMPING AND FREQUENCY        003146
C    CALLING SEQUENCE FOR SUBROUTINE FORMB IS AS FOLLOWS                  003147
C    CALL FORMB (KR, KC, RLRT, CMPR, FBR, FBC)                            003148
C     KR -- COUNT OF REAL ROOTS, MAY OR MAY NOT INCLUDE ZEROS.            003149
C      KC -- COUNT OF THE COMPLEX PAIRS OF ROOTS.                         003150
C       RLRT -- STORAGE BLOCK CONTAING ALL REAL ROOTS.                    003151
C       CMPR -- STORAGE BLOCK CONTAINING COMPLEX PAIRS OF ROOTS           003152
C         FBC -- FORM B COMPLEX PAIR BLOCK (OUTPUT FROM ROUTINE)          003153
C          FBR -- FORM B REAL ROOT BLOCK (OUTPUT FROM ROUTINE)            003154
C                                                                         003155
       DIMENSION RLRT(1),CMPR(1),FBR(1),FBC(1)                            003156
C                                                                         003157
       IF (KR) 140, 140, 100                                              003158
  100 DO 130 L = 1,KR                                                     003159
       IF (RLRT(L)) 120, 110, 120                                         003160
  110 FBR(L) = 0.0                                                        003161
       GO TO 130                                                          003162
  120 FBR(L) = -1.DO/ RLRT(L)                                             003163
  130 CONTINUE                                                            003164
  140 IF (KC) 170, 170, 150                                               003165
  150 KK = 2*KC                                                           003166
       DO 160 L = 2,KK,2                                                  003167
       FBC(L) = DSQRT(CMPR(L-1)**2 + CMPR(L)**2)                          003168
  160 FBC(L-1) = -CMPR(L-1)/FBC(L)                                        003169
  170 RETURN                                                              003170
       END                                                                003171
```

```
[HDG,P     GAUSSI                                                        -003172
[FOR,IS    GAUSSI                                                        -003173
       COMPILER (XM=1),(EQUIV=CMN)                                       -003174
       SUBROUTINE GAUSSI (A,R,N,KAR)                                      003175
       IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -003176
       DIMENSION A(KAR,1), R(KAR,1), IVEC(150)                           6903177
              COMMON /DRATIO/                                             003178
      *      IFL1,IFL2,DRVEC(150)                                        10003179
C                                                                         003180
       DATA EPS,NOT / 1.D-06, 6/                                          003181
 1001 FORMAT (5X,30HMATRIX SINGULAR IN GAUSSI AT  , I5)                   003182
C                                                                         003183
       DO 5 I=1,N                                                         003184
       DO 7 J=1,N                                                         003185
    7  R(I,J) = 0.D 0                                                     003186
    5  R(I,I) = 1.D 0                                                     003187
C                                                                         003188
       DO 10 L=1,N                                                        003189
       JBIG = 1                                                           003190
       A1 = DABS(A(L,1))                                                  003191
       DO 15 J=2,N                                                        003192
       A2 = DABS(A(L,J))                                                  003193
       IF (A2 .LT. A1) GO TO 15                                           003194
       A1 = A2                                                            003195
       JBIG = J                                                           003196
   15  CONTINUE                                                           003197
       IVEC(L) = JBIG                                                     003198
       IF (A1 .GT. EPS) GO TO 20                                          003199
       IF (IFL1 .EQ. 0) GO TO 75                                          003200
       IFL2 = 1                                                           003201
       GO TO 100                                                          003202
   75  WRITE (NOT,1001) L                                                 003203
       STOP                                                               003204
   20  CONTINUE                                                           003205
       ALJBIG = A(L,JBIG)                                                 003206
       DRVEC(L) = ALJBIG                                                  003207
       DO 17 J=1,N                                                        003208
       A(L,J) = A(L,J)/ALJBIG                                             003209
   17  R(L,J) = R(L,J)/ALJBIG                                             003210
       DO 25 I=1,N                                                        003211
       AIJBIG = A(I,JBIG)                                                 003212
       IF (I .EQ. L) GO TO 25                                             003213
       DO 30 J=1,N                                                        003214
       A(I,J) = A(I,J) - AIJBIG*A(L,J)                                    003215
   30  R(I,J) = R(I,J) - AIJBIG*R(L,J)                                    003216
   25  CONTINUE                                                           003217
   10  CONTINUE                                                           003218
C                                                                         003219
       DO 40 I=1,N                                                        003220
       IR = IVEC(I)                                                       003221
```

```
      DO 40 J=1,N                                                    003222
   40 A(IR, J) = R(I,J)                                              003223
      DO 50 I=1,N                                                    003224
      DO 50 J=1,N                                                    003225
   50 R(I,J) = A(I,J)                                                003226
      SN = 1.0D0                                                      03227
      DO 110 L = 1,N                                                  03228
      DO 120 J = 1,N                                                  03229
      IF(IVEC(J) .EQ. L) GO TO 110                                    03230
      IF(IVEC(J) .GT. L) SN = -SN                                     03231
  120 CONTINUE                                                        03232
  110 CONTINUE                                                        03233
      DRVEC(1) = SN*DRVEC(1)                                          03234
C                                                                     003235
  100 IFL1 = 0                                                        003236
      RETURN                                                          003237
      END                                                             003238
```

```
[HDG,P     GETBMB                                                        -003239
[FOR,IS    GETBMB                                                        -003240
      COMPILER (XM=1),(EQUIV=CMN)                                        -003241
      SUBROUTINE GETBMB                                                   003242
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -003243
C                                                                         003244
            COMMON /AMUBW /                                               003245
     *      AMU(15,15, 5),BW(30, 65)                                     103246
            COMMON /BHBSRD/                                               003247
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)               203248
            COMMON /IVCONS/                                               003249
     *      IV(6, 5)                                                     903250
            COMMON /MAXMUM/                                               003251
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU            003252
            COMMON /NUMBRS/                                               003253
     *      ZRO,ONE,TWO,TRES                                             003254
            COMMON /SPECIF/                                               003255
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),    1603256
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),           1703257
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1803258
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                          1903259
C                                                                         003260
      DIMENSION BMB(30,30),BM(6,15, 9), ITOP( 5, 5),WR(12)               8203261
      EQUIVALENCE (BW(1),BMB(1)),  (BW( 901),BM(1))                      7703262
C                                                                         003263
      DATA IIST / 0 /                                                     003264
C                                                                         003265
      IF (IIST .EQ. 1) GO TO 100                                          003266
      IIST = 1                                                            003267
      DO 5 I=1,NH                                                         003268
      DO 5 J=1,NB                                                         003269
    5 ITOP(I,J) = 0                                                       003270
      ITOP(1,1) = 1                                                       003271
      DO 3 I=1,6                                                          003272
      DO 3 J=1,NH                                                         003273
    3 IV(I,J) = 0                                                         003274
      IC = 0                                                              003275
      DO 7 J=1,NH                                                         003276
      DO 7 I=1,6                                                          003277
      IP1 = I + 1                                                         003278
      IF (IHDATA(IP1,J) .EQ. 0) GO TO 7                                   003279
      IC = IC + 1                                                         003280
      IV(I,J) = IC                                                        003281
    7 CONTINUE                                                            003282
      DO 10 L=2,NH                                                        003283
      LQ = 2*L - 2                                                        003284
      LP = LQ + 1                                                         003285
      NQ = ITOPOL(1,L)                                                    003286
      NP = ITOPOL(2,L)                                                    003287
      ITOP(L,NQ) = LQ                                                     003288
```

```
   10 ITOP(L,NP) = LP                                                    003289
C                                                                        003290
  100 LEQ = 6 + IRGFLX(1)                                                003291
      LMQ = LENU(1)                                                      003292
      CALL MLTSR  (BH,AMU,BM,LEQ,LMQ,1,IV,KMU)                           003293
C                                                                        003294
      DO 20 L=2,NH                                                       003295
      NOBQ = ITOPOL(1,L)                                                 003296
      NOBP = ITOPOL(2,L)                                                 003297
      LEQ = IRGFLX(NOBQ) + 6                                             003298
      LEP = IRGFLX(NOBP) + 6                                             003299
      LMQ = LENU(NOBQ)                                                   003300
      LMP = LENU(NOBP)                                                   003301
      LQ = 2*L - 2                                                       003302
      LP = LQ + 1                                                        003303
      CALL MLTSR (BH(1,1,LQ),AMU(1,1,NOBQ),BM(1,1,LQ),LEQ,LMQ,L,IV,KMU) 003304
      CALL MLTSR (BH(1,1,LP),AMU(1,1,NOBP),BM(1,1,LP),LEP,LMP,L,IV,KMU) 003305
   20 CONTINUE                                                           003306
C                                                                        003307
      DO 25 I=1,NLAM                                                     003308
      DO 25 J=1,NLAM                                                     003309
   25 BMB(I,J) = ZRO                                                     003310
C                                                                        003311
      DO 30 N=1,NB                                                       003312
      LE = IRGFLX(N) + 6                                                 003313
      DO 35 L=1,NH                                                       003314
      IF (ITOP(L,N) .EQ. 0) GO TO 35                                     003315
      DO 40 I=L,NH                                                       003316
      IF (ITOP(I,N) .EQ. 0) GO TO 40                                     003317
      LBM = ITOP(L,N)                                                    003318
      LBT = ITOP(I,N)                                                    003319
      DO 50 M=1,6                                                        003320
      ML = IV(M,L)                                                       003321
      IF (ML .EQ. 0) GO TO 50                                            003322
      DO 55 K=1,LE                                                       003323
   55 WR(K) = BM(M,K,LBM)                                                003324
      DO 60 J=1,6                                                        003325
      JI = IV(J,I)                                                       003326
      IF (JI .LT. ML) GO TO 60                                           003327
      S = ZRO                                                            003328
      DO 65 K=1,LE                                                       003329
   65 S = S + WR(K)*BH(J,K,LBT)                                          003330
      BMB(ML,JI) = BMB(ML,JI) + S                                        003331
   60 CONTINUE                                                           003332
   50 CONTINUE                                                           003333
   40 CONTINUE                                                           003334
   35 CONTINUE                                                           003335
   30 CONTINUE                                                           003336
C                                                                        003337
      RETURN                                                             003338
```

END

```
[HDG,P     GMISC                                                          -003340
[FOR,IS    GMISC                                                          -003341
      COMPILER (XM=1),(EQUIV=CMN)                                         -003342
      SUBROUTINE GMISC (N,LE,LO,V2)                                        003343
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                 -003344
      DIMENSION V2(1)                                                      003345
C                                                                          003346
            COMMON /MAXMUM/                                                003347
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMOBOD,KMU,KY,KU              003348
            COMMON /SPECIF/                                                003349
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),     1603350
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),            1703351
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1803352
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                           1903353
            COMMON /VECTOR/                                                003354
     *      Y(250),YDT(250)                                               2003355
C                                                                          003356
      DATA I1ST / 0 /                                                      003357
C                                                                          003358
CCC   USER SUPPLIED SUBROUTINE TO CREAT MISC. CONTRIBUTIONS TO R.H.S.      003359
CCC   INCLUDING  THE THERMAL GRADIENT ENVIRONMENT.                         003360
C                                                                          003361
      IF (I1ST .EQ. 1) GO TO 5                                             003362
      I1ST = 1                                                             003363
C                                                                          003364
    5 CONTINUE                                                             003365
      LOM1 = LO - 1                                                        003366
      DO 10 I=1,LE                                                         003367
   10 V2(I) = Y(LOM1 + I) - 0.D 0                                          003368
C                                                                          003369
      RETURN                                                               003370
      END                                                                  003371
```

```
[HDG,P     GRVGRD                                                        -003372
[FOR,IS    GRVGRD                                                        -003373
      COMPILER (XM=1),(EQUIV=CMN)                                        -003374
      SUBROUTINE GRVGRD (GGV)                                             003375
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -003376
      DIMENSION GGV(1)                                                    003377
C                                                                         003378
            COMMON /AMUBW /                                               003379
     *      AMU(15,15, 5),BW(30, 65)                                     103380
            COMMON /BHBSRD/                                               003381
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)               203382
            COMMON /GGDATA/                                               003383
     *      GAMGI(3),GMAG,RCMAG                                          003384
            COMMON /GGSAVE/                                               003385
     *      GGS( 6,9, 5)                                                 303386
            COMMON /INTGRL/                                               003387
     *      AM( 78, 5),ACOF(9, 6, 5),BCOF(6, 6, 5),                      503388
     *      COF11( 6, 6, 5),COF22( 6, 6, 5),COF33( 6, 6, 5),AK( 6, 6, 5), 603389
     *      COF12( 6, 6, 5),COF13( 6, 6, 5),COF23( 6, 6, 5),AD( 6, 6, 5), 703390
     *      COFXY( 6, 6, 5),COFXZ( 6, 6, 5),COFYZ( 6, 6, 5)              803391
            COMMON /MAXMUM/                                               003392
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU            003393
            COMMON /NUMBRS/                                               003394
     *      ZRO,ONE,TWO,TRES                                             003395
            COMMON /SPECIF/                                               003396
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),    1603397
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),           1703398
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IPGFLX( 5),IHDATA(7, 5), 1803399
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                          1903400
            COMMON /VECTOR/                                               003401
     *      Y(250),YDT(250)                                              2003402
C                                                                         003403
      DIMENSION GAMGL(3),V(3)                                             003404
      DATA NOT / 6/                                                       003405
C                                                                         003406
      IF (GMAG .EQ. ZRO) RETURN                                           003407
      IF (RCMAG .LE. ONE) GO TO 999                                       003408
      DO 10 N=1,NB                                                        003409
      LOU = LOCU(N)                                                       003410
      LO = LOCU(N+NB)                                                     003411
      LE = LENU(N+NB)                                                     003412
      CALL MULT3   (GAMGI,ROL(1,1,N),GAMGL,1,3,3,1,3,1)                  003413
      CALL MULT3   (AMU(4,1,N),GAMGL,GGV(LOU),3,3,1,KMU,1,1)             003414
      CALL MULT3   (AMU,GAMGL,V,3,3,1,KMU,1,1)                            003415
      GGV(LOU   ) = GMAG*(GGV(LOU   )                                     003416
     * + TRES*(GAMGL(2)*V(3) - GAMGL(3)*V(2))/RCMAG)                      003417
      GGV(LOU+1) = GMAG*(GGV(LOU+1)                                       003418
     * + TRES*(GAMGL(3)*V(1) - GAMGL(1)*V(3))/RCMAG)                      003419
      GGV(LOU+2) = GMAG*(GGV(LOU+2)                                       003420
     * + TRES*(GAMGL(1)*V(2) - GAMGL(2)*V(1))/RCMAG)                      003421
```

```
      V(1) = AMU(3,5,N)                                                    003422
      V(2) = AMU(1,6,N)                                                    003423
      V(3) = AMU(2,4,N)                                                    003424
      S = ZRO                                                              003425
      GCM = -GMAG*AMU(4,4,N)                                               003426
      GCR = -GMAG/RCMAG                                                    003427
      LOP2 = LOU + 2                                                       003428
      DO 20 I=1,3                                                          003429
      GGV(LOP2+I) = GCM*GAMGL(I) + GCR*V(I)                                003430
   20 S = S + GAMGL(I)*V(I)                                                003431
      S = -TRES*S*GCR                                                      003432
      DO 25 I=1,3                                                          003433
   25 GGV(LOP2+I) = GGV(LOP2+I) + S*GAMGL(I)                               003434
      IF (LE .EQ. 0) GO TO 10                                              003435
      LOP6 = LOU + 6                                                       003436
      V(1) = GAMGL(1)*RCMAG                                                003437
      V(2) = GAMGL(2)*RCMAG                                                003438
      V(3) = GAMGL(3)*RCMAG                                                003439
      CALL MULT3  (V,AMU(4,7,N),GGV(LOP6),1,3,LE,1,KMU,1)                  003440
      CALL MULTAD (AMU(7,7,N),Y(LO),GGV(LOP6),LE,LE,1,KMU,1,1)             003441
      LOP5 = LOU + 5                                                       003442
      DO 30 J=1,LE                                                         003443
      GGV(LOP5+J) =  GCR*GGV(LOP5+J)                                       003444
     *      + GCR*(BCOF(1,J,N) + BCOF(2,J,N) + BCOF(3,J,N))/TWO            003445
   30 CONTINUE                                                             003446
C                                                                          003447
      V(1) = ONE - TWO*GAMGL(1)*GAMGL(1)                                   003448
      V(2) = ONE - TWO*GAMGL(2)*GAMGL(2)                                   003449
      V(3) = ONE - TWO*GAMGL(3)*GAMGL(3)                                   003450
      S = TRES*GMAG/(TWO*RCMAG)                                            003451
C                                                                          003452
      DO 40 J=1,LE                                                         003453
      GGV(LOP5+J) = GGV(LOP5+J)                                            003454
     *  + S* (V(1)*(BCOF(1,J,N) + GGS(J,1,N))                              003455
     *  +     V(2)*(BCOF(2,J,N) + GGS(J,2,N))                              003456
     *  +     V(3)*(BCOF(3,J,N) + GGS(J,3,N))                              003457
     * + TWO*GAMGL(1)*GAMGL(2)*(BCOF(4,J,N) + GGS(J,4,N) + GGS(J,7,N))     003458
     * + TWO*GAMGL(1)*GAMGL(3)*(BCOF(5,J,N) + GGS(J,5,N) + GGS(J,8,N))     003459
     * + TWO*GAMGL(2)*GAMGL(3)*(BCOF(6,J,N) + GGS(J,6,N) + GGS(J,9,N)))    003460
   40 CONTINUE                                                             003461
C                                                                          003462
   10 CONTINUE                                                             003463
      RETURN                                                               003464
C                                                                          003465
  999 WRITE (NOT,2001)                                                     003466
 2001 FORMAT (1H1,29HRCMAG = 0., SUBROUTINE GRVGRD)                        003467
      STOP                                                                 003468
      END                                                                  003469
```

```
[HDG,P     INV1NP                                                        -003470
[FOR,IS    INV1NP                                                        -003471
      COMPILER (XM=1),(EQUIV=CMN)                                        -003472
      SUBROUTINE INV1NP (A,R,N,KA)                                        003473
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -003474
C                                                                         003475
C **********                                                            003476
C             MATRIX A MUST BE SYMMETRIC, MAX N = XX                      003477
C **********                                                            003478
C                                                                         003479
      DIMENSION A(KA,1),R(KA,1),CL( 15)                                  8003480
C                                                                         003481
      DATA EPS,NOT / 1.D-06, 6/                                           003482
C                                                                         003483
      L = 1                                                               003484
      IF (A(1,1) .LT. EPS) GO TO 999                                      003485
C                                                                         003486
      R(1,1) = 1.D+00/A(1,1)                                              003487
      DO 100  L = 2,N                                                     003488
      N1 = L - 1                                                          003489
      DO 10 I=1,N1                                                        003490
      CL(I) = 0.D+00                                                      003491
      DO 10 J=1,N1                                                        003492
   10 CL(I) = CL(I) + R(I,J)*A(J,L)                                       003493
      S = A(L,L)                                                          003494
      DO 20 I=1,N1                                                        003495
   20 S = S - A(I,L)*CL(I)                                                003496
      IF (DABS(S) .LT. EPS) GO TO 999                                     003497
      S = 1.D+00/S                                                        003498
      DO 30 I=1,N1                                                        003499
      V = -S*CL(I)                                                        003500
      R(I,L) = V                                                          003501
      R(L,I) =  R(I,L)                                                    003502
      DO 30 J=I,N1                                                        003503
      R(I,J) = R(I,J) - V*CL(J)                                           003504
   30 R(J,I) = R(I,J)                                                     003505
      R(L,L) = S                                                          003506
  100 CONTINUE                                                            003507
C                                                                         003508
      RETURN                                                              003509
C                                                                         003510
  999 WRITE(NOT,900) L                                                    003511
  900 FORMAT(/,5X,32HSINGULAR MATRIX IN INV1NP AT L =,I5,9H STOP RUN)     003512
      STOP                                                                003513
C                                                                         003514
      END                                                                 003515
```

```
[HDG,P     KHINGE                                                        -003516
[FOR,IS    KHINGE                                                        -003517
      COMPILER (XM=1),(EQUIV=CMN)                                        -003518
      SUBROUTINE KHINGE (G)                                               003519
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                -003520
      DIMENSION G(1)                                                      003521
      DIMENSION SK(3,6),DK(3,6),HNGT(3,6)                                 003522
C                                                                         003523
            COMMON /BHBSRD/                                               003524
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                203525
            COMMON /CONPAR/                                               003526
     *      CNTDTA(100)                                                 9503527
            COMMON /MAXMUM/                                               003528
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU            003529
            COMMON /MOMENG/                                               003530
     *      P( 65),PMOM(30),HTOT(3),TOTL(3),ENGKE( 5),ENGPE( 5),        1103531
     *      TOTKE, TOTPE, TOTENG, AHTOT,ATOTL                            003532
            COMMON /SPECIF/                                               003533
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),   1603534
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),          1703535
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1803536
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                         1903537
C                                                                         003538
      EQUIVALENCE (CNTDTA(12),SK(1)), (CNTDTA(30),DK(1))                  003539
C                                                                         003540
      TOTPE = 0.DO                                                        003541
C                                                                         003542
      DO 10 L=1,NH                                                        003543
      DO 10 I=1,3                                                         003544
      HNGT(I,L) = -(SK(I,L)*BETAH(I,L) + DK(I,L)*BETAHD(I,L))             003545
   10 TOTPE = TOTPE + 0.5DO*SK(I,L)*BETAH(I,L)**2                         003546
C                                                                         003547
      LEQ = IRGFLX(1) + 6                                                 003548
      DO 15 I=1,3                                                         003549
      F = HNGT(I,1)                                                       003550
      DO 16 J=1,LEQ                                                       003551
   16 G(J) = G(J) + F*BH(I,J,1)                                           003552
   15 CONTINUE                                                            003553
C                                                                         003554
      DO 20 L=2,NH                                                        003555
      NOBQ = ITOPOL(1,L)                                                  003556
      NOBP = ITOPOL(2,L)                                                  003557
      LQ = 2*L - 2                                                        003558
      LP = LQ + 1                                                         003559
      LOQ = LOCU(NOBQ) - 1                                                003560
      LOP = LOCU(NOBP) - 1                                                003561
      LEQ = IRGFLX(NOBQ) + 6                                              003562
      LEP = IRGFLX(NOBP) + 6                                              003563
      DO 20 I=1,3                                                         003564
      F = HNGT(I,L)                                                       003565
```

```
      DO 25 J=1,LEQ                                                    003566
      LOQJ = LOQ + J                                                   003567
   25 G(LOQJ) = G(LOQJ) + F*BH(I,J,LQ)                                 003568
      DO 26 J=1,LEP                                                    003569
      LOPJ = LOP + J                                                   003570
   26 G(LOPJ) = G(LOPJ) + F*BH(I,J,LP)                                 003571
   20 CONTINUE                                                         003572
C                                                                      003573
      RETURN                                                           003574
      END                                                              003575
```

C-12

```
[HDG,P    LINEAR                                                        -003576
[FOR,IS   LINEAR                                                        -003577
      COMPILER (XM=1),(EQUIV=CMN)                                       -003578
      SUBROUTINE LINEAR (NR,NC)                                          003579
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               -003580
C                                                                        003581
C  SUBROUTINE ESTABLISHES FIRST PARTIAL DERIVATIVES OF A YDOT            003582
C  DESCRIBED FUNCTIONAL AT AN INITIAL STATE, Y, USING QUADRATIC          003583
C  INTERPOLATION FUNCTIONS.                                              003584
C                                                                        003585
C  THE PARTIAL DERIVATIVE MATRIX IS WRITTEN COLUMNWISE ON UNIT NU.       003586
C  CALLS SUBROUTINE YDOT                                                 003587
            COMMON /PRWORK/                                              003588
     *      PR(250,5)                                                  1403589
            COMMON /TAPENO/                                              003590
     *      NTAPE1,NTAPE2,NTAPE3                                        003591
            COMMON /VECTOR/                                              003592
     *      Y(250),YDT(250)                                            2003593
            COMMON /VINDEP/                                              003594
     *      INDEP(250)                                                 2103595
      COMMON /LDSIZE/ NX,NY,NDLTA,NXSS,NB,NJQ,NY2,ND2                    003596
C                                                                        003597
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                003598
C  NR      = INPUT  NUMBER OF ROWS IN PARTIAL DERIVATIVE MATRIX.         003599
C  NC      = INPUT  NUMBER OF COLUMNS IN PARTIAL DERIVATIVE MATRIX.      003600
C                                                                        003601
      DIMENSION FY(250,3),Z(250),ZNEW(250),IV(250)                      7803602
      EQUIVALENCE (PR(1),FY(1)),(PR( 751),Z(1)),(PR(1001),ZNEW(1))      7903603
      EQUIVALENCE (INDEP(1),IV(1))                                       003604
      DATA NOT/ 6 /                                                      003605
      DATA PCON,PMIN/ 1.D-02,1.D-05 /                                    003606
      DATA EPS1,EPS2/ 1.D-10,1.D-04 /                                    003607
C                                                                        003608
      NU = NTAPE2                                                        003609
C                                                                        003610
C  ESTABLISH OUTPUT SIZE OF PARTIAL DERIVATIVE MATRIX                    003611
C                                                                        003612
      NJQ = 0                                                            003613
      NX  = 0                                                            003614
C                                                                        003615
      DO 5 I=1,NP                                                        003616
      IF (IV(I) .NE. 0) NJQ = NJQ+1                                      003617
      IF (I .GT. NC) GO TO 5                                             003618
      IF (IV(I) .NE. 0) NX=NX+1                                          003619
    5 CONTINUE                                                           003620
C                                                                        003621
      REWIND NU                                                          003622
C                                                                        003623
      DO 20 I=1,NR                                                       003624
   20 FY(I,1) = YDT(I)                                                   003625
```

```
C                                                                       003626
      DO 200 L=1,NC                                                     003627
      IF (IV(L) .EQ. 0) GO TO 200                                       003628
      DY = PCON * Y(L)                                                  003629
      IF (DY .LT. PMIN) DY = PMIN                                       003630
C                                                                       003631
      YY = Y(L)                                                         003632
      Y(L) = Y(L) + DY                                                  003633
      CALL YDOT                                                         003634
      Y(L) = YY                                                         003635
C                                                                       003636
      DO 30 I=1,NR                                                      003637
   30 FY(I,3) = YDT(I)                                                  003638
C                                                                       003639
      YY = Y(L)                                                         003640
      Y(L) = Y(L) + 0.5D0 * DY                                          003641
      CALL YDOT                                                         003642
      Y(L) = YY                                                         003643
C                                                                       003644
      DO 35 I=1,NR                                                      003645
   35 FY(I,2) = YDT(I)                                                  003646
C                                                                       003647
      DO 50 I=1,NR                                                      003648
      E1 = -3.D0 * FY(I,1) + 4.D0 *FY(I,2) - FY(I,3)                    003649
      Z(I) = E1 / DY                                                    003650
      IF (DABS(Z(I)) .LT. EPS1)   Z(I) = 0.D0                           003651
   50 CONTINUE                                                          003652
      ITR = 0                                                           003653
   60 DY = 0.5D0*DY                                                     003654
C                                                                       003655
      DO 70 I=1,NR                                                      003656
   70 FY(I,3) = FY(I,2)                                                 003657
C                                                                       003658
      YY = Y(L)                                                         003659
      Y(L) = Y(L) + 0.5D0 * DY                                          003660
      CALL YDOT                                                         003661
      Y(L) = YY                                                         003662
C                                                                       003663
      DO 80 I=1,NR                                                      003664
   80 FY(I,2) = YDT(I)                                                  003665
C                                                                       003666
      DO 90 I=1,NR                                                      003667
      E1 = -3.D0 * FY(I,1) + 4.D0 * FY(I,2) - FY(I,3)                   003668
      ZNEW(I) = E1/DY                                                   003669
      IF (DABS(ZNEW(I)) .LT.  EPS1)  ZNEW(I) = 0.D0                     003670
   90 CONTINUE                                                          003671
C                                                                       003672
      DO 100 I=1,NP                                                     003673
C                                                                       003674
      IF (IV(I) .EQ. 0) GO TO 100                                       003675
```

```
C                                                                       003676
      DN = DABS(Z(I))                                                   003677
      DN1 = DABS(ZNEW(I))                                               003678
C                                                                       003679
      IF (DN1 .GT. DN) DN = DN1                                         003680
      IF (DN  .LT. EPS1) GO TO 100                                      003681
      G1 = DABS(ZNEW(I) - Z(I))/ DN                                     003682
C                                                                       003683
      IF (G1 .LE. EPS2) GO TO 100                                       003684
C                                                                       003685
      ITR = ITR + 1                                                     003686
      IF (ITR .GT. 30) GO TO 999                                        003687
      DO 95 J=1,NR                                                      003688
   95 Z(J) = ZNEW(J)                                                    003689
      GO TO 60                                                          003690
  100 CONTINUE                                                          003691
C                                                                       003692
C  COMPLETION OF THE DO 100 LOOP INDICATES WE HAVE ACCEPTED             003693
C  ZNEW(I) ,I=1,NR.                                                     003694
C                                                                       003695
C  NOW PACK PARTIAL DERIVATIVES INTO A NJQ LONG VECTOR.                 003696
C                                                                       003697
      J=0                                                               003698
      DO 110 I=1,NR                                                     003699
      IF (IV(I) .EQ.0) GO TO 110                                        003700
      J=J+1                                                             003701
      ZNEW(J) = ZNEW(I)                                                 003702
  110 CONTINUE                                                          003703
C                                                                       003704
      WRITE (NU) (ZNEW(J),J=1,NJQ)                                      003705
C                                                                       003706
  200 CONTINUE                                                          003707
      RETURN                                                            003708
C                                                                       003709
  999 WRITE (NOT,2001) I,L,DY,Z(I),ZNEW(I)                              003710
 2001 FORMAT (1H1,////,20X,                                             003711
     * 36HSUBROUTINE LINEAR FAILED TO CONVERGE ,/,20X,                  003712
     * 28HIN 30  ITERATIONS ON ELEMENT ,/,10X,                         003713
     *4HI =  ,I3,/,10X,                                                003714
     *4HJ =  ,I3,/,10X,19HLAST Y INCREMENT =  ,D12.4,//,10X,            003715
     *7HZ    = D12.4,/,10X,                                            003716
     *7HZNEW = D12.4)                                                  003717
C                                                                       003718
      STOP                                                              003719
      END                                                               003720
```

```
[HDG,P    LPLTWR                                                        -003721
[FOR,IS   LPLTWR                                                        -003722
      COMPILER (XM=1),(EQUIV=CMN)                                       -003723
      SUBROUTINE LPLTWR                                                  003724
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -003725
C                                                                        003726
C  SUBROUTINE WRITES TAPE NUT3 FOR PLOTTING.                             003727
C                                                                        003728
      COMMON /LDSIZE/                                                    003729
     2               NX, NY, NDLTA, NXSS, NB, NJQ, NY2, ND2             003730
      COMMON /TAPENO/                                                    003731
     4               NUT1, NUT2, NUT3                                   003732
      COMMON /VECTOR/                                                    003733
     E               Y   (250), YD  (250)                              43003734
      COMMON /TIMESS/                                                    003735
     G               ST, DT, T, ET, TMST                                003736
      COMMON /PLTDTA/                                                    003737
     I                NRPLOT,NCPLOT                                     003738
      DATA I1ST / 0 /                                                    003739
      IF (I1ST .NE. 0) GO TO 5                                           003740
      I1ST = 1                                                           003741
      REWIND NUT3                                                        003742
      NRPLOT = 0                                                         003743
      NCPLOT = 2*NX + 1                                                  003744
    5 NRPLOT = NRPLOT + 1                                                003745
      WRITE (NUT3) T,(YD(I),I=1,NX),(Y(I),I=1,NX)                        003746
C                                                                        003747
      RETURN                                                             003748
      END                                                                003749
```

```
[HDG,P    LPRNT                                                          -003750
[FOR,IS   LPRNT                                                          -003751
      COMPILER (XM=1),(EQUIV=CMN)                                        -003752
      SUBROUTINE LPRNT                                                    003753
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -003754
C                                                                         003755
C  SUBROUTINE PRINTS OUT RESULTS OF LINEARIZED TIME RESPONSE.             003756
C                                                                         003757
      COMMON /VECTOR/                                                     003758
     E               Y   (250), YD  (250)                                43003759
      COMMON /LDSIZE/ NX,NY,NDLTA,NXSS,NB,NJQ,NY2,ND2                     003760
      COMMON /TIMESS/                                                     003761
     G               ST, DT, T, ET, TMST                                  003762
      DATA NOT/ 6 /                                                       003763
      DATA I1ST/ 0 /                                                      003764
C                                                                         003765
      IF (I1ST .NE. 0) GO TO 5                                            003766
      I1ST = 1                                                            003767
C                                                                         003768
C  PRINT OUT DATA AT START.                                               003769
C                                                                         003770
      CALL PAGEHD                                                         003771
      WRITE (NOT,11)                                                      003772
   11 FORMAT (////30X,24HLINEARIZED TIME RESPONSE /                       003773
     *          32X,24H GENERAL INFORMATION     //)                      003774
      WRITE (NOT,12) ST,DT,ET                                             003775
   12 FORMAT (//30X,26HINTEGRATION PARAMETERS ARE  //                     003776
     *          30X,13HSTART TIME = D12.5,/,                             003777
     *          30X,13HDELTA TIME = D12.5,/,                             003778
     *          30X,13HEND   TIME = D12.5)                               003779
C                                                                         003780
      LY = 1                                                              003781
      LX = LY+NY2                                                         003782
      LD=LX+NXSS                                                          003783
      LB=LD+ND2                                                           003784
C                                                                         003785
    5 CONTINUE                                                            003786
      CALL PAGEHD                                                         003787
      WRITE (NOT,101) T                                                   003788
  101 FORMAT  (/5X,18HSIMULATION TIME = D12.5)                            003789
C                                                                         003790
      WRITE (NOT,102) NY2,NXSS,ND2,NB                                     003791
  102 FORMAT  (/30X,30HNUMBER OF PLANT VARIABLES   = I5,                  003792
     *         /30X,30HNUMBER OF SENSOR SIGNALS   =  I5,                003793
     *         /30X,30HNUMBER OF CONTROL VARIABLES = I5,                003794
     *         /30X,30HNUMBER OF CONTROL TORQUES   = I5)                003795
C                                                                         003796
      WRITE (NOT,103)                                                     003797
  103 FORMAT (//20X,29HSTATE VECTOR TIME DERIVATIVES /)                   003798
      CALL WRITES (YD,1,NX,1)                                             003799
```

```
C                                                                       003800
      WRITE (NOT,104)                                                   003801
  104 FORMAT  (//20X,12HSTATE VECTOR /)                                 003802
      CALL WRITES (Y ,1,NX,1)                                           003803
C                                                                       003804
      WRITE (NOT,105)                                                   003805
  105 FORMAT (/20X,20HPLANT VARIABLE RATES )                            003806
      CALL WRITES (YD,1,NY2,1)                                          003807
C                                                                       003808
      WRITE (NOT,106)                                                   003809
  106 FORMAT (//20X,19HSENSOR SIGNAL RATES)                             003810
      CALL WRITES (YD(LX),1,NXSS,1)                                     003811
C                                                                       003812
      WRITE (NOT,107)                                                   003813
  107 FORMAT (//20X,22HCONTROL VARIABLE RATES)                          003814
      CALL WRITES (YD(LD),1,ND2,1)                                      003815
C                                                                       003816
      WRITE (NOT,108)                                                   003817
  108 FORMAT (//20X,19HTORQUE SYSTEM RATES)                             003818
      CALL WRITES (YD(LB),1,NB,1)                                       003819
C                                                                       003820
      WRITE (NOT,109)                                                   003821
  109 FORMAT (///20X,20HPLANT VARIABLE STATE)                           003822
      CALL WRITES (Y,1,NY2,1)                                           003823
C                                                                       003824
      WRITE (NOT,110)                                                   003825
  110 FORMAT (//20X,19HSENSOR SIGNAL STATE)                             003826
      CALL WRITES (Y(LX),1,NXSS,1)                                      003827
C                                                                       003828
      WRITE (NOT,111)                                                   003829
  111 FORMAT (//20X,22HCONTROL VARIABLE STATE)                          003830
      CALL WRITES (Y(LD),1,ND2,1)                                       003831
C                                                                       003832
      WRITE (NOT,112)                                                   003833
  112 FORMAT (//20X,19HTORQUE SYSTEM STATE)                             003834
      CALL WRITES (Y(LB),1,NB,1)                                        003835
C                                                                       003836
      RETURN                                                            003837
      END                                                               003838
```

```
[HDG,P    LTORQL                                                    -003839
[FOR,IS   LTORQL                                                    -003840
      COMPILER (XM=1),(EQUIV=CMN)                                   -003841
      SUBROUTINE LTORQL (VTORQ)                                      003842
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                            -003843
C                                                                    003844
      DIMENSION VTORQ(1)                                             003845
      COMMON /KDSIZE/                                                003846
     1               KR, KRT, KRX, KV1, KV2, KVX                     003847
      COMMON /VECTOR/                                                003848
     E               Y   (250), YD  (250)                          43003849
      COMMON /TIMESS/                                                003850
     G               ST, DT, T, ET, TMST                             003851
C                                                                    003852
      DATA I1ST/0/                                                   003853
C                                                                    003854
      TLMT = 10.DO*DT                                                003855
      IF (I1ST .NE. 0) GO TO 10                                      003856
      IF (T .GT. TLMT) I1ST = 1                                      003857
      CALL ZERO (VTORQ,1,KVX,1)                                      003858
      VTORQ(34) = 1.DO                                               003859
      VTORQ(35) = 1.DO                                               003860
      RETURN                                                         003861
   10 CONTINUE                                                       003862
      CALL ZERO (VTORQ,1,KVX,1)                                      003863
      RETURN                                                         003864
      END                                                            003865
```

```
[HDG,P    LTRESP                                                    -003866
[FOR,IS   LTRESP                                                    -003867
      COMPILER (XM=1),(EQUIV=CMN)                                   -003868
      SUBROUTINE LTRESP                                              003869
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                            -003870
C                                                                    003871
C  SUBROUTINE SOLVES FOR THE LINEARIZED TIME RESPONSE.               003872
C  TECHNIQUE USES A RUNGE KUTTA STARTER AND AN ADAMS CORRECTOR       003873
C  RECURRSIVE FORMULA FOR THE TIME SOLUTION.                         003874
C                                                                    003875
      COMMON /KDSIZE/                                               003876
     1               KR, KRT, KRX, KV1, KV2, KVX                    003877
      COMMON /LDSIZE/                                               003878
     2               NX, NY, NDLTA, NXSS, NB, NJQ, NY2, ND2         003879
      COMMON /TAPENO/                                               003880
     4               NUT1, NUT2, NUT3                               003881
      COMMON /MISCNO/                                               003882
     5               NOPRNT, NOPLOT                                 003883
      COMMON /  LV1 /                                               003884
     C               V1  ( 50), V2  ( 50), V3  ( 50)               42603885
      COMMON /VECTOR/                                               003886
     E               Y   (250), YD  (250)                          43003887
      COMMON /LWORK1/                                               003888
     F               W1( 50, 50), W2( 50, 50)                      43203889
      COMMON /TIMESS/                                               003890
     G               ST, DT, T, ET, TMST                            003891
C                                                                    003892
C  ASSUMES THAT W1 = A* ON ENTRY.                                    003893
C  UNIT NUT3 WILL BE WRITTEN FOR PLOTTING                            003894
C                                                                    003895
      DIMENSION PRK(4), YDS(250,3), YS(250)                         47003896
C                                                                    003897
C  STORE Y* IN YS, THEN ZERO Y.                                      003898
      DO 20 I=1,NX                                                   003899
      YS(I) = Y(I)                                                   003900
   20 Y(I) = 0.D0                                                    003901
      PRK(1) = 0.5D0                                                 003902
      PRK(2) = 1.D0 - DSQRT(0.5D0)                                   003903
      PRK(3) = 1.D0 + DSQRT(0.5D0)                                   003904
      PRK(4) = 0.5D0                                                 003905
C                                                                    003906
      NT = 0                                                         003907
      T = ST                                                         003908
      TMST = 0.D0                                                    003909
      IPRNT = 0                                                      003910
      IPLOT = 0                                                      003911
C                                                                    003912
      REWIND NUT3                                                    003913
C                                                                    003914
      WRITE (NUT3) ((W1(I,J),I=1,KR),J=1,KR)                         003915
```

```
      REWIND NUT3                                                       003916
C                                                                       003917
      CON = 3.D0 * DT/8.D0                                              003918
C                                                                       003919
      DO 30 I=1,NX                                                      003920
      DO 30 J=1,NX                                                      003921
      W1(I,J) = -CON*W1(I,J)                                            003922
      IF (J .EQ. I) W1(I,J) = 1.+W1(I,J)                                003923
   30 CONTINUE                                                          003924
C                                                                       003925
      CALL GAUSSI (W1,W2,NX,KR)                                         003926
      READ (NUT3) ((W1(I,J),I=1,KR),J=1,KR)                             003927
      REWIND NUT3                                                       003928
C                                                                       003929
      CALL ZERO (V1,1,NX,1)                                             003930
      CALL LTORQL (V2)                                                  003931
      CALL YDOTL (W1,V2,Y,YD,NX,KR)                                     003932
C                                                                       003933
      DO 10 I=1,NX                                                      003934
   10 YDS(I,1) = YD(I)                                                  003935
C  USE THE R-K STARTER.                                                 003936
      CALL LPRNT                                                        003937
      CALL LPLTWR                                                       003938
C                                                                       003939
  100 CONTINUE                                                          003940
      DO 120 J=1,4                                                      003941
      JIL = J                                                           003942
      DO 110 I=1,NX                                                     003943
      Z = YD(I) * DT                                                    003944
      GO TO (103,101,101,105),JIL                                       003945
  101 R = PRK(JIL) * (Z-V1(I))                                          003946
      GO TO 107                                                         003947
  103 R = PRK(JIL) * Z - V1(I)                                          003948
      GO TO 107                                                         003949
  105 R = (Z-2.D0 * V1(I)) / 6.D0                                       003950
  107 Y(I) = Y(I) + R                                                   003951
  110 V1(I) = V1(I) + 3.D0 * R - PRK(JIL) * Z                           003952
      IF (JIL .EQ. 1  .OR. JIL .EQ. 3) T = T+ DT/2.D0                   003953
      CALL LTORQL (V2)                                                  003954
  120 CALL YDOTL (W1,V2,Y,YD,NX,KR)                                     003955
C                                                                       003956
C                                                                       003957
      NT = NT + 1                                                       003958
      ANT = NT                                                          003959
      TMST = ANT * DT                                                   003960
      T = ST + TMST                                                     003961
C                                                                       003962
      IPRNT = IPRNT + 1                                                 003963
      IF (IPRNT .NE. NOPRNT) GO TO 130                                  003964
      CALL LPRNT                                                        003965
```

```
      IPRNT = 0                                                          003966
  130 CONTINUE                                                           003967
      IPLOT = IPLOT + 1                                                  003968
      IF (IPLOT .NE. NOPLOT) GO TO 140                                   003969
      CALL LPLTWR                                                        003970
      IPLOT = 0                                                          003971
  140 CONTINUE                                                           003972
      DO 150 I=1,NX                                                      003973
  150 YDS(I,NT+1) = YD(I)                                                003974
      IF (T .LE. ET  .AND.  NT .LT. 2) GO TO 100                         003975
C                                                                        003976
C  THE ADAMS CORRECTOR FORMULA                                           003977
C                                                                        003978
      CO = DT / 24.DO                                                    003979
      C1 = CO * 9.DO                                                     003980
      C2 = CO * 19.DO                                                    003981
      C3 = -CO * 5.DO                                                    003982
      C4 = CO                                                            003983
C                                                                        003984
C                                                                        003985
C     ESTABLISH Y AT STEP NT                                             003986
C                                                                        003987
  200 CALL LTORQL (V1)                                                   003988
C                                                                        003989
C  V1 IS EXTERNAL FORCING FUNCTION FOR THE LINEAR SYSTEM.                003990
C                                                                        003991
      DO 210 I=1,NX                                                      003992
  210 Y(I) = Y(I) +C1*V1(I) + C2*YDS(I,3) + C3*YDS(I,2) + C4*YDS(I,1)    003993
      CALL MULTB (W2,Y,NX,NX,1,KR,KR)                                    003994
C                                                                        003995
C  RESET YDS FOR NEXT STEP.                                              003996
C                                                                        003997
      DO 220 I=1,NX                                                      003998
      YDS(I,1) = YDS(I,2)                                                003999
  220 YDS(I,2) = YDS(I,3)                                                004000
C                                                                        004001
C  COMPUTE YD AT STEP NT.                                                004002
      CALL YDOTL (W1,V1,Y,YD,NX,KR)                                      004003
C                                                                        004004
      DO 225 I=1,NX                                                      004005
  225 YDS(I,3) = YD(I)                                                   004006
C                                                                        004007
      NT = NT + 1                                                        004008
      ANT = NT                                                           004009
      TMST = ANT * DT                                                    004010
      T = ST + TMST                                                      004011
C                                                                        004012
      IPRNT = IPRNT + 1                                                  004013
      IF (IPRNT .NE. NOPRNT) GO TO 230                                   004014
      CALL LPRNT                                                         004015
```

```
      IPRNT = 0                                                    004016
  230 CONTINUE                                                     004017
      IPLOT = IPLOT + 1                                            004018
      IF (IPLOT .NE. NOPLOT) GO TO 240                             004019
      CALL LPLTWR                                                  004020
      IPLOT = 0                                                    004021
  240 CONTINUE                                                     004022
C                                                                  004023
      IF (T .LT. ET) GO TO 200                                     004024
C                                                                  004025
C                                                                  004026
C                                                                  004027
      RETURN                                                       004028
      END                                                          004029
```

```
[HDG,P    MGEN                                                                  -004030
[FORL,IS  MGEN                                                                  -004031
      COMPILER (XM=1),(EQUIV=CMN)                                               -004032
      SUBROUTINE MGEN                                                            004033
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                        -004034
C                                                                                004035
            COMMON /AMUBW /                                                      004036
     *      AMU(15,15, 5),BW(30, 65)                                            104037
            COMMON /BHBSRD/                                                      004038
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                       204039
            COMMON /GGSAVE/                                                      004040
     *      GGS( 6,9, 5)                                                        304041
            COMMON /INTGRL/                                                      004042
     *      AM( 78, 5),ACOF(9, 6, 5),BCOF(6, 6, 5),                             504043
     *      COF11( 6, 6, 5),COF22( 6, 6, 5),COF33( 6, 6, 5),AK( 6, 6, 5),       604044
     *      COF12( 6, 6, 5),COF13( 6, 6, 5),COF23( 6, 6, 5),AD( 6, 6, 5),       704045
     *      COFXY( 6, 6, 5),COFXZ( 6, 6, 5),COFYZ( 6, 6, 5)                     804046
            COMMON /MAXMUM/                                                      004047
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU                   004048
            COMMON /NUMBRS/                                                      004049
     *      ZRO,ONE,TWO,TRES                                                    004050
            COMMON /SPECIF/                                                      004051
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),          1604052
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),                 1704053
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5),      1804054
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                                1904055
            COMMON /VECTOR/                                                      004056
     *      Y(250),YDT(250)                                                    2004057
C                                                                                004058
      DIMENSION RW(3, 6),CW( 6,3),VW(9),WV(6)                                   8304059
C                                                                                004060
      KM = NMDBOD                                                                004061
      DO 10 N=1,NB                                                               004062
      LO = LOCU(NB+N)                                                            004063
      LE = LENU(NB+N)                                                            004064
      KNT = 0                                                                    004065
      NP6 = 6 + LE                                                               004066
      DO 12 I=1,NP6                                                              004067
      DO 12 J=I,NP6                                                              004068
      KNT = KNT + 1                                                              004069
   12 AMU(I,J,N) = AM(KNT,N)                                                     004070
      IF (LE .EQ. 0) GO TO 50                                                    004071
      CALL MULT3  (BCOF(1,1,N),Y(LO),VW,6,LE,1,6,1,1)                            004072
      CALL MULT3  (COF11(1,1,N),Y(LO),CW(1,1),LE,LE,1,KM,1,KM)                   004073
      CALL MULT3  (COF22(1,1,N),Y(LO),CW(1,2),LE,LE,1,KM,1,KM)                   004074
      CALL MULT3  (COF33(1,1,N),Y(LO),CW(1,3),LE,LE,1,KM,1,KM)                   004075
      CALL MULT3  (Y(LO),COF12(1,1,N),RW(1,1),1,LE,LE,1,KM,3)                    004076
      CALL MULT3  (Y(LO),COF13(1,1,N),RW(2,1),1,LE,LE,1,KM,3)                    004077
      CALL MULT3  (Y(LO),COF23(1,1,N),RW(3,1),1,LE,LE,1,KM,3)                    004078
C                                                                                004079
```

```
C  PEEL OFF DATA FOR GRAVITY GRADIENT EFFECTS ON ELASTIC COORDINATES      004080
C                                                                        004081
      DO 8 J=1,LE                                                        004082
      GGS(J,1,N) = CW(J,1)                                               004083
      GGS(J,2,N) = CW(J,2)                                               004084
      GGS(J,3,N) = CW(J,3)                                               004085
      GGS(J,7,N) = RW(1,J)                                               004086
      GGS(J,8,N) = RW(2,J)                                               004087
    8 GGS(J,9,N) = RW(3,J)                                               004088
C                                                                        004089
      CALL MULT3  (Y(LO),CW,WV(1),1,LE,3,1,KM,1)                         004090
      CALL MULT3  (RW,Y(LO),WV(4),3,LE,1,3,1,1)                          004091
      DO 15 I=1,3                                                        004092
   15 AMU(I,I,N) = AMU(I,I,N) + TWO*VW(I) + WV(I)                        004093
      AMU(1,2,N) = AMU(1,2,N) - VW(4) - WV(4)                            004094
      AMU(1,3,N) = AMU(1,3,N) - VW(5) - WV(5)                            004095
      AMU(2,3,N) = AMU(2,3,N) - VW(6) - WV(6)                            004096
      CALL MULT3 (ACOF(1,1,N),Y(LO),VW,9,LE,1,9,1,1)                     004097
      DO 17 J=1,3                                                        004098
      JP3 = J + 3                                                        004099
      DO 17 I=1,3                                                        004100
      IJ = J + 3*(I-1)                                                   004101
   17 AMU(I,JP3,N) = AMU(I,JP3,N) + VW(IJ)                               004102
      CALL MULTAD (Y(LO),COFYZ(1,1,N),AMU(1,7,N),1,LE,LE,1,KM,KMU)       004103
      CALL MULTAD (Y(LO),COFXZ(1,1,N),AMU(2,7,N),1,LE,LE,1,KM,KMU)       004104
      CALL MULTAD (Y(LO),COFXY(1,1,N),AMU(3,7,N),1,LE,LE,1,KM,KMU)       004105
C                                                                        004106
      CALL MULT3  (COF12(1,1,N),Y(LO),CW(1,1),LE,LE,1,KM,1,KM)           004107
      CALL MULT3  (COF13(1,1,N),Y(LO),CW(1,2),LE,LE,1,KM,1,KM)           004108
      CALL MULT3  (COF23(1,1,N),Y(LO),CW(1,3),LE,LE,1,KM,1,KM)           004109
C                                                                        004110
C  FINISH PEELING OFF GRAVITY GRADIENT DATA                              004111
C                                                                        004112
      DO 28 J=1,LE                                                       004113
      GGS(J,4,N) = CW(J,1)                                               004114
      GGS(J,5,N) = CW(J,2)                                               004115
   28 GGS(J,6,N) = CW(J,3)                                               004116
C                                                                        004117
C                                                                        004118
   50 NMWVS = NMOW(2,N)                                                  004119
      IF (NMWVS .EQ. 0) GO TO 110                                        004120
      NMW = NMOW(1,N)                                                    004121
      LEBS = 6 + LE                                                      004122
      NV = 0                                                             004123
      J1 = LEBS + 1                                                      004124
      J2 = LEBS + NMWVS                                                  004125
      DO 70 L=1,NMW                                                      004126
      LP2 = L + 2                                                        004127
      NOMW = NMOW(LP2,N)                                                 004128
      IF (IMO(3,NOMW) .EQ. 0) GO TO 70                                   004129
```

```
      NV = NV + 1                                                    004130
      LV = 6 + LE + NV                                               004131
      NOSP = IMO(1,NOMW)                                             004132
      NA = IMO(2,NOMW)                                               004133
      AJS = AMO(2,NOMW)                                              004134
      DO 75 J=1,LEBS                                                 004135
   75 AMU(J,LV,N) = AJS*BS(NA,J,NOSP)                                004136
      DO 76 J=J1,J2                                                  004137
   76 AMU(LV,J,N) = ZRO                                              004138
      AMU(LV,LV,N) = AJS                                             004139
   70 CONTINUE                                                       004140
C                                                                    004141
  110 LEU = LENU(N)                                                  004142
      DO 77 I=1,LEU                                                  004143
      DO 77 J=I,LEU                                                  004144
   77 AMU(J,I,N) = AMU(I,J,N)                                        004145
C                                                                    004146
   10 CONTINUE                                                       004147
C                                                                    004148
      RETURN                                                         004149
      END                                                            004150
```

```
[HDG,P     MLTSR                                                    -004151
[FOR,IS    MLTSR                                                    -004152
       COMPILER (XM=1),(EQUIV=CMN)                                  -004153
       SUBROUTINE MLTSR (A,B,C,LE,LM,L,IV,KMU)                       004154
       IMPLICIT DOUBLE PRECISION(A-H,O-Z)                           -004155
       DIMENSION A(6,1),B(KMU,1),C(6,1),IV(6,1),RW(12)              8604156
C                                                                    004157
       DO 10 I=1,6                                                   004158
       IL = IV(1,L)                                                  004159
       IF (IL .EQ. 0) GO TO 10                                       004160
       DO 15 K=1,LE                                                  004161
   15 RW(K) = A(I,K)                                                 004162
       DO 20 J=1,LM                                                  004163
       S = 0.D 0                                                     004164
       DO 25 K=1,LE                                                  004165
   25 S = S + RW(K)*B(K,J)                                           004166
   20 C(I,J) = S                                                     004167
   10 CONTINUE                                                       004168
C                                                                    004169
       RETURN                                                        004170
       END                                                           004171
```

```
[HDG,P     MRIGID                                                          -004172
[FOR,IS    MRIGID                                                          -004173
      COMPILER (XM=1),(EQUIV=CMN)                                          -004174
      SUBROUTINE MRIGID (N)                                                 004175
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                   -004176
C                                                                           004177
            COMMON /NHNS  /                                                 004178
     *     NHPOI( 5), NSPOI( 5)                                            1204179
            COMMON /SPECIF/                                                 004180
     *     BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),       1604181
     *     DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),              1704182
     *     NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5),   1804183
     *     LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                             1904184
            COMMON /SUMMRY/                                                 004185
     *     ASUMRY(10,6),ISUMRY(10,3),KSUMRY                                9804186
            COMMON /TAPENO/                                                 004187
     *     NTAPE1,NTAPE2,NTAPE3                                            004188
C                                                                           004189
      DIMENSION V(6),AINER(6,6)                                             004190
      DATA NIT,NOT /5, 6/                                                   004191
 1001 FORMAT (16I5)                                                         004192
 1002 FORMAT (8D10.3)                                                       004193
 3001 FORMAT (//15X30HSUMMARY OF INPUT DATA FOR BODY,I3,                    004194
     *   16H WHICH IS RIGID. //3X29HTHE 6X6 INERTIA MATRIX IS ---)          004195
 3002 FORMAT (//5X9HFOR BODY I3,33H THE P-Q HINGE NO. AND THE EULER         004196
     * 57HROTATION TYPE APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH /       004197
     *   5X57H IS FOLLOWED BY AN ARRAY CONTAINING EULER ANGLES (1,2,3),     004198
     * 58H AND POSITION VECTOR COMPONENTS (4,5,6) THAT POSITION THE /       004199
     *   5X30HHINGE TRIAD WRT THE BODY TRIAD)                              004200
 3003 FORMAT (//5X9HFOR BODY I3,35H THE SENSOR POINT NO. AND THE EULER      004201
     * 58 H ROTATION TYPE APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH /     004202
     *   5X56HIS FOLLOWED BY AN ARRAY CONTAINING EULER ANGLES(1,2,3),      004203
     * 56 HAND POSITION VECTOR COMPONENTS (4,5,6) THAT POSITION THE/       004204
     *   5X31HSENSOR TRIAD WRT THE BODY TRIAD)                             004205
C                                                                           004206
      NHB = NHPOI(N)                                                        004207
      NSB = NSPOI(N)                                                        004208
C                                                                           004209
      CALL ZERO  (AINER,6,6,6)                                              004210
      CALL READ (V,N1,N2,1,6)                                               004211
      DO 5 J=2,4                                                            004212
    5 V(J) = -V(1)*V(J)                                                     004213
      CALL SKEWV3 (V(2),AINER(1,4),1,6)                                     004214
      DO 6 I=4,6                                                            004215
    6 AINER(I,I) = V(1)                                                     004216
      CALL READ (V,N1,N2,1,6)                                               004217
      DO 7 I=1,3                                                            004218
    7 AINER(I,I) = V(I)                                                     004219
      AINER(1,2) = -V(4)                                                    004220
      AINER(1,3) = -V(5)                                                    004221
```

```
      AINER(2,3) = -V(6)                                                004222
      DO 8 I=1,6                                                        004223
      DO 8 J=I,6                                                        004224
    8 AINER(J,I) = AINER(I,J)                                           004225
      WRITE (NTAPE1) ((AINER(I,J),J=1,6),I=1,6)                         004226
      CALL PAGEHD                                                       004227
      WRITE (NOT,3001) N                                                004228
      CALL WRITES (AINER,6,6,6)                                         004229
C                                                                       004230
      DO 10 I=1,NHB                                                     004231
      READ (NIT,1001) NOH,ITYPE                                         004232
      ISUMRY(I,1) = NOH                                                 004233
      ISUMRY(I,2) = ITYPE                                               004234
      IF (NOH .EQ. 1) GO TO 999                                         004235
      LR = 6*(NOH - 2) + 3                                              004236
      LD = 7*(NOH - 2) + 3                                              004237
      IF (ITOPOL(1,NOH) .EQ. N) GO TO 12                                004238
      IF (ITOPOL(2,NOH) .NE. N) GO TO 999                               004239
      LR = LR + 1                                                       004240
      LD = LD + 1                                                       004241
   12 READ (NIT,1002) (V(J),J=1,3)                                      004242
      READ (NIT,1002) (DH(J,LD),J=1,3)                                  004243
      DO 11 J=1,3                                                       004244
      J1 = J + 3                                                        004245
      ASUMRY(I,J ) = V(J)                                               004246
   11 ASUMRY(I,J1) = DH(J,LD)                                           004247
      CALL ROTTR  (3,ITYPE,V,RH(1,1,LR),DUM,DUM)                        004248
   10 CONTINUE                                                          004249
      WRITE (NOT,3002) N                                                004250
      CALL WRITIS (ISUMRY,NHB,2,KSUMRY)                                 004251
      CALL WRITES (ASUMRY,NHB,6,KSUMRY)                                 004252
C                                                                       004253
      IF (NSB .EQ. 0) RETURN                                            004254
      DO 20 I=1,NSB                                                     004255
      READ (NIT,1001) NOS,ITYPE                                         004256
      ISUMRY(I,1) = NOS                                                 004257
      ISUMRY(I,2) = ITYPE                                               004258
      IF (IFTSMW(NOS) .NE. N) GO TO 999                                 004259
      LR = 2*NOS                                                        004260
      READ (NIT,1002) (V(J),J=1,3)                                      004261
      READ (NIT,1002) (DS(J,LR),J=1,3)                                  004262
      DO 21 J=1,3                                                       004263
      J1 = J + 3                                                        004264
      ASUMRY(I,J ) = V(J)                                               004265
   21 ASUMRY(I,J1) = DS(J,LR)                                           004266
      CALL ROTTR  (3,ITYPE,V,RS(1,1,LR),DUM,DUM)                        004267
   20 CONTINUE                                                          004268
      WRITE (NOT,3003) N                                                004269
      CALL WRITIS (ISUMRY,NSB,2,KSUMRY)                                 004270
      CALL WRITES (ASUMRY,NSB,6,KSUMRY)                                 004271
```

```
      RETURN                                                                    004272
C                                                                               004273
  999 WRITE (NOT,2001)                                                          004274
 2001 FORMAT (1H1,49HTOPOLOGY ERROR,SUBROUTINE MRIGID, PROGRAM STOPPED)         004275
      STOP                                                                      004276
C                                                                               004277
      END                                                                       004278
```

```
[HDG,P     MSMODC                                                          -004279
[FORL,IS   MSMODC                                                          -004280
      COMPILER (XM=1),(EQUIV=CMN)                                          -004281
      SUBROUTINE MSMODC (NBOD)                                              004282
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                   -004283
C                                                                           004284
            COMMON /MAXMUM/                                                 004285
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU               004286
            COMMON /NHNS  /                                                 004287
     *      NHPOI( 5), NSPOI( 5)                                           1204288
            COMMON /NUMBRS/                                                 004289
     *      ZRO,ONE,TWO,TRES                                               004290
            COMMON /SPECIF/                                                 004291
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),      1604292
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),             1704293
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5),  1804294
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                            1904295
            COMMON /SUMMRY/                                                 004296
     *      ASUMRY(10,6),ISUMRY(10,3),KSUMRY                              9804297
            COMMON /TAPENO/                                                 004298
     *      NTAPE1,NTAPE2,NTAPE3                                           004299
C                                                                           004300
      DIMENSION A( 42, 42),B( 42, 42),IV( 42),JV( 42),C(6,6),PHR6(6,6),     6104301
     *   BC(9, 6),WS1( 6, 6),WS2( 6, 6),OM2( 42),OMGA2(12),JDOF( 7,6)      6204302
C                                                                           004303
      DATA NIT,NOT,KAB,KJDOF/ 5,6, 42, 7 /                                  6304304
 1001 FORMAT (16I5)                                                         004305
 1002 FORMAT (8D10.3)                                                       004306
 3001 FORMAT (//15X30HSUMMARY OF INPUT DATA FOR BODY,I3,                    004307
     * 44H WHICH IS FLEXIBLE W/CONSISTENT MASS MATRIX.//                     004308
     * 3X49HTHE INTEGER PARAMETERS--- IFRBM,IFDIAK,IFDIAD ARE5XI2,1H,       004309
     *   5XI2,1H,5XI2,//                                                    004310
     *  3X25HTHE JDOF TABLE FOLLOWS---)                                     004311
 3002 FORMAT (//3X36HTHE MODE SELECTION VECTOR FOLLOWS---)                  004312
 3003 FORMAT (//3X12HFOR BODY NO.,I3,25H THE POSITION VECTOR FROM           004313
     *25H THE BODY ORIGIN TO JOINT,I4,3H IS, /                              004314
     *  10X 4HX = 1PD10.3,5X5H Y = 1PD10.3,5X5H Z = 1PD10.3)                004315
 3004 FORMAT (//5X46HTHE CONSISTENT, REPARTITIONED MASS MATRIX IS--)        004316
 3005 FORMAT (//5X36HTHE REPARTITIONED MODAL MATRIX IS---)                  004317
 3006 FORMAT (//5X44HTHE  -UNDEFORMED-  INERTIA MATRIX (MU) IS---)          004318
 3007 FORMAT (//5X27HTHE  A  COEFFICIENTS ARE---)                           004319
 3008 FORMAT (//5X27HTHE  B  COEFFICIENTS ARE---)                           004320
 3009 FORMAT (//5X31HTHE  COFXY  COEFFICIENTS ARE---)                       004321
 3010 FORMAT (//5X31HTHE  COFXZ  COEFFICIENTS ARE---)                       004322
 3011 FORMAT (//5X31HTHE  COFYZ  COEFFICIENTS ARE---)                       004323
 3012 FORMAT (//5X31HTHE  C11    COEFFICIENTS ARE---)                       004324
 3013 FORMAT (//5X31HTHE  C22    COEFFICIENTS ARE---)                       004325
 3014 FORMAT (//5X31HTHE  C33    COEFFICIENTS ARE---)                       004326
 3015 FORMAT (//5X31HTHE  C12    COEFFICIENTS ARE---)                       004327
 3016 FORMAT (//5X31HTHE  C13    COEFFICIENTS ARE---)                       004328
```

```
 3017 FORMAT (//5X31HTHE  C23    COEFFICIENTS ARE---)                          004329
 3018 FORMAT (//5X33HTHE  MODAL STIFFNESS MATRIX IS---)                        004330
 3019 FORMAT (//5X31HTHE  MODAL DAMPING MATRIX IS---)                          004331
 3020 FORMAT (//5X50HTHE  INITIAL MODAL COORDINATE DISPLACEMENTS ARE---)       004332
 3021 FORMAT (//5X47HTHE  INITIAL MODAL COORDINATE VELOCITIES ARE---)          004333
 3022 FORMAT (//5X 9HFOR BODY I3,29H THE P-Q HINGE NO., THE EULER              004334
     * 57H ROTATION TYPE AND THE JOINT NO. CORRESPONDING TO THE P-Q,/          004335
     *  5X54H HINGE APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS            004336
     *    49HFOLLOWED BY AN ARRAY CONTAINING EULER ANGLES THAT,/              004337
     * 5X44H POSITION THE HINGE TRIAD WRT THE BODY TRIAD)                     004338
 3023 FORMAT (//5X 9HFOR BODY I3,32H THE SENSOR POINT NO., THE EULER           004339
     *60H ROTATION TYPE AND THE JOINT NO. CORRESPONDING TO THE SENSOR,/       004340
     *6X53HPOINT APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS               004341
     *    49HFOLLOWED BY AN ARRAY CONTAINING EULER ANGLES THAT,/              004342
     * 5X45H POSITION THE SENSOR TRIAD WRT THE BODY TRIAD)                    004343
C                                                                             004344
      REWIND NTAPE2                                                           004345
      KMO = NMDBOD                                                            004346
      CALL ZERO   (BC,9,KMO,9)                                                004347
      READ (NIT,1001) IFRBM,IDIAK,IDIAD                                       004348
      CALL READIM (JDOF,NX,N6,KJDOF,6)                                        004349
      IC = 0                                                                  004350
      DO 2 J=1,6                                                              004351
      DO 2 I=1,NX                                                             004352
      NDF = JDOF(I,J)                                                         004353
      IC = IC + 1                                                             004354
    2 IV(NDF) = IC                                                            004355
      CALL READIM (JV,N1,N2,1,KAB)                                            004356
      NY = NX                                                                 004357
      NZ = NX                                                                 004358
      NMO = 0                                                                 004359
      CALL PAGEHD                                                             004360
      WRITE (NOT,3001) NBOD,IFRBM,IDIAK,IDIAD                                 004361
      CALL WRITIS (JDOF,NX,6,KJDOF)                                           004362
      WRITE (NOT,3002)                                                        004363
      CALL WRITIS (JV,1,N2,1)                                                 004364
C                                                                             004365
      DO 3 I=1,N2                                                             004366
      IF (JV(I) .EQ. 0) GO TO 3                                               004367
      NMO = NMO + 1                                                           004368
    3 CONTINUE                                                                004369
      IF (NMO .GT. KMO + 6) GO TO 999                                         004370
      CALL READ   (A,NRA,NCA,KAB,KAB)                                         004371
      CALL REVISE (A,IV,IV,B,NRA,NCA,NRA,NRA,KAB,KAB)                         004372
      WRITE (NTAPE2) ((B(I,J),I=1,NRA),J=1,NRA)                               004373
      REWIND NTAPE2                                                           004374
C                                                                             004375
      CALL READ   (A,NRA,NMOT,KAB,KAB)                                        004376
      IF (IDIAK .EQ. 0 .AND. IDIAD .EQ. 0) GO TO 11                           004377
      CALL READ   (OM2,N1,N2,1,KAB)                                           004378
```

```
   11 NE = NMO - 6                                                      004379
      CALL REVISE (A,IV,JV,B,NRA,NMOT,NRA,NMO,KAB,KAB)                  004380
      IF (IDIAK .EQ. 0 .AND. IDIAD .EQ. 0) GO TO 12                     004381
      CALL REVISE (OM2,1,JV,OMGA2,1,N2,1,NMO,1,1)                       004382
   12 IF (IFRBM .EQ. 0) GO TO 5                                         004383
      READ (NIT,1001) JTYPCL                                            004384
      READ (NIT,1002) (OM2(J),J=1,3)                                    004385
      WRITE (NOT,3003) NBOD,JTYPCL,OM2(1),OM2(2),OM2(3)                 004386
      JRB = JTYPCL - NX                                                 004387
      DO 4 I=1,6                                                        004388
      JRB = JRB + NX                                                    004389
      DO 4 J=1,6                                                        004390
    4 PHR6(I,J) = B(JRB,J)                                              004391
      CALL GAUSSI (PHR6,C,6,6)                                          004392
      CALL ZERO  (PHR6(4,4),3,3,6)                                      004393
      CALL UNITY (PHR6(1,4),3,6)                                        004394
      CALL UNITY (PHR6(4,1),3,6)                                        004395
      CALL SKEWV3 (OM2,PHR6,1,6)                                        004396
      CALL MULTA  (C,PHR6,6,6,6,6,6)                                    004397
      CALL MULTA  (B,C,NRA,6,6,KAB,6)                                   004398
C                                                                       004399
    5 READ (NTAPE2) ((A(I,J),I=1,NRA),J=1,NRA)                          004400
      REWIND NTAPE2                                                     004401
      WRITE (NOT,3004)                                                  004402
      CALL WRITES (A  ,NRA,NRA,KAB)                                     004403
      WRITE (NOT,3005)                                                  004404
      CALL WRITES (B  ,NRA,NMO,KAB)                                     004405
C                                                                       004406
      CALL BTABA (A,B,NRA,NMO,KAB,KAB)                                  004407
      WRITE (NOT,3006)                                                  004408
      CALL WRITES (A  ,NMO,NMO,KAB)                                     004409
      WRITE (NTAPE1) ((A(I,J),J=1,NMO),I=1,NMO)                         004410
      DO 25 J=1,NE                                                      004411
      JP6 = J + 6                                                       004412
   25 OM2(J) = A(JP6,JP6)                                               004413
C                                                                       004414
      READ (NTAPE2) ((A(I,J),I=1,NRA),J=1,NRA)                          004415
      REWIND NTAPE2                                                     004416
      NRP = 3*NX                                                        004417
      CALL MULTA (A,B,NRP,NRA,NMO,KAB,KAB)                              004418
      CALL ZERO  (BC,9,NE,9)                                            004419
      DO 15 J=1,NE                                                      004420
      K = 6 + J                                                         004421
      DO 15 IX = 1,NX                                                   004422
      IY = IX + NX                                                      004423
      IZ = IY + NX                                                      004424
      BC(1,J) = BC(1,J) + A(IZ,4)*B(IY,K) - A(IY,4)*B(IZ,K)             004425
      BC(2,J) = BC(2,J) + A(IZ,5)*B(IY,K) - A(IY,5)*B(IZ,K)             004426
      BC(3,J) = BC(3,J) + A(IZ,6)*B(IY,K) - A(IY,6)*B(IZ,K)             004427
      BC(4,J) = BC(4,J) + A(IX,4)*B(IZ,K) - A(IZ,4)*B(IX,K)             004428
```

```
      BC(5,J) = BC(5,J) + A(IX,5)*B(IZ,K) - A(IZ,5)*B(IX,K)                    004429
      BC(6,J) = BC(6,J) + A(IX,6)*B(IZ,K) - A(IZ,6)*B(IX,K)                    004430
      BC(7,J) = BC(7,J) + A(IY,4)*B(IX,K) - A(IX,4)*B(IY,K)                    004431
      BC(8,J) = BC(8,J) + A(IY,5)*B(IX,K) - A(IX,5)*B(IY,K)                    004432
      BC(9,J) = BC(9,J) + A(IY,6)*B(IX,K) - A(IX,6)*B(IY,K)                    004433
   15 CONTINUE                                                                 004434
      WRITE (NTAPE1) ((BC(I,J),J=1,NF),I=1,9)                                  004435
      WRITE (NOT,3007)                                                         004436
      CALL WRITES (BC ,  9,NE ,  9)                                            004437
C                                                                              004438
      CALL ZERO  (BC,9,NE,9)                                                   004439
      DO 16 J=1,NE                                                             004440
      K = 6 + J                                                                004441
      DO 16 IX=1,NX                                                            004442
      IY = IX + NX                                                             004443
      IZ = IY + NX                                                             004444
      BC(1,J) = BC(1,J) + A(IZ,1)*B(IY,K) - A(IY,1)*B(IZ,K)                    004445
      BC(2,J) = BC(2,J) + A(IX,2)*B(IZ,K) - A(IZ,2)*B(IX,K)                    004446
      BC(3,J) = BC(3,J) + A(IY,3)*B(IX,K) - A(IX,3)*B(IY,K)                    004447
      BC(4,J) = BC(4,J) + A(IZ,1)*B(IX,K) - A(IX,1)*B(IZ,K)                    004448
     *                  + A(IY,2)*B(IZ,K) - A(IZ,2)*B(IY,K)                    004449
      BC(5,J) = BC(5,J) + A(IX,1)*B(IY,K) - A(IY,1)*B(IX,K)                    004450
     *                  + A(IY,3)*B(IZ,K) - A(IZ,3)*B(IY,K)                    004451
      BC(6,J) = BC(6,J) + A(IX,2)*B(IY,K) - A(IY,2)*B(IX,K)                    004452
     *                  + A(IZ,3)*B(IX,K) - A(IX,3)*B(IZ,K)                    004453
   16 CONTINUE                                                                 004454
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=1,6)                                  004455
      WRITE (NOT,3008)                                                         004456
      CALL WRITES (BC ,  6,NE ,  9)                                            004457
C                                                                              004458
      CALL ZERO  (WS1,NE,NE,KMO)                                               004459
      DO 17 I=1,NE                                                             004460
      KI = 6 + I                                                               004461
      DO 17 J=1,NF                                                             004462
      KJ = 6 + J                                                               004463
      DO 17 IX=1,NX                                                            004464
      IY = IX + NX                                                             004465
      IZ = IY + NX                                                             004466
      WS1(I,J) = WS1(I,J) + B(IX,KI)*A(IY,KJ) - B(IY,KI)*A(IX,KJ)             004467
   17 CONTINUE                                                                 004468
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NF)                                004469
      WRITE (NOT,3009)                                                         004470
      CALL WRITES (WS1,NE ,NE ,KMO)                                            004471
C                                                                              004472
      CALL ZERO  (WS1,NE,NE,KMO)                                               004473
      DO 18 I=1,NF                                                             004474
      KI = 6 + I                                                               004475
      DO 18 J=1,NE                                                             004476
      KJ = 6 + J                                                               004477
      DO 18 IX=1,NX                                                            004478
```

```
      IY = IX + NX                                                      004479
      IZ = IY + NX                                                      004480
      WS1(I,J) = WS1(I,J) + B(IZ,KI)*A(IX,KJ) - B(IX,KI)*A(IZ,KJ)       004481
   18 CONTINUE                                                          004482
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)                         004483
      WRITE (NOT,3010)                                                  004484
      CALL WRITES (WS1,NE ,NE ,KMO)                                     004485
C                                                                       004486
      CALL ZERO  (WS1,NE,NE,KMO)                                        004487
      DO 19 I=1,NE                                                      004488
      KI = 6 + I                                                        004489
      DO 19 J=1,NE                                                      004490
      KJ = 6 + J                                                        004491
      DO 19 IX=1,NX                                                     004492
      IY = IX + NX                                                      004493
      IZ = IY + NX                                                      004494
      WS1(I,J) = WS1(I,J) + B(IY,KI)*A(IZ,KJ) - B(IZ,KI)*A(IY,KJ)       004495
   19 CONTINUE                                                          004496
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)                         004497
      WRITE (NOT,3011)                                                  004498
      CALL WRITES (WS1,NE ,NE ,KMO)                                     004499
C                                                                       004500
      LX = 1                                                            004501
      LY = LX + NX                                                      004502
      LZ = LY + NY                                                      004503
      S1 = ONE                                                          004504
C                                                                       004505
      READ (NTAPE2) ((A(I,J),I=1,NRA),J=1,NRA)                          004506
C                                                                       004507
C                                                                       004508
      CALL ZERO   (WS1,NE,NE,KMO)                                       004509
      CALL PR3 (A(LZ,LZ),B(LY,7),B(LY,7),WS2,WS1, S1,NZ,NZ,NE,NE,       004510
     *  KAB,KAB,KAB,KJDOF,KMO)                                          004511
      CALL PR3 (A(LZ,LY),B(LZ,7),B(LY,7),WS2,WS1,-S1,NZ,NY,NE,NE,       004512
     *  KAB,KAB,KAB,KJDOF,KMO)                                          004513
      CALL PR3 (A(LY,LZ),B(LY,7),B(LZ,7),WS2,WS1,-S1,NY,NZ,NE,NE,       004514
     *  KAB,KAB,KAB,KJDOF,KMO)                                          004515
      CALL PR3 (A(LY,LY),B(LZ,7),B(LZ,7),WS2,WS1, S1,NY,NY,NE,NE,       004516
     *  KAB,KAB,KAB,KJDOF,KMO)                                          004517
      WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)                         004518
      WRITE (NOT,3012)                                                  004519
      CALL WRITES (WS1,NE ,NE ,KMO)                                     004520
      CALL ZERO   (WS1,NE,NE,KMO)                                       004521
      CALL PR3 (A(LX,LX),B(LZ,7),B(LZ,7),WS2,WS1, S1,NX,NX,NE,NE,       004522
     *  KAB,KAB,KAB,KJDOF,KMO)                                          004523
      CALL PR3 (A(LX,LZ),B(LX,7),B(LZ,7),WS2,WS1,-S1,NX,NZ,NE,NE,       004524
     *  KAB,KAB,KAB,KJDOF,KMO)                                          004525
      CALL PR3 (A(LZ,LX),B(LZ,7),B(LX,7),WS2,WS1,-S1,NZ,NX,NE,NE,       004526
     *  KAB,KAB,KAB,KJDOF,KMO)                                          004527
      CALL PR3 (A(LZ,LZ),B(LX,7),B(LX,7),WS2,WS1, S1,NZ,NZ,NE,NE,       004528
```

```
*  KAB,KAB,KAB,KJDOF,KMO)                                                004529
 WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)                               004530
 WRITE (NOT,3013)                                                        004531
 CALL WRITES (WS1,NE ,NE ,KMO)                                           004532
 CALL ZERO    (WS1,NE,NE,KMO)                                            004533
 CALL PR3 (A(LY,LY),B(LX,7),B(LX,7),WS2,WS1, S1,NY,NY,NE,NE,             004534
*  KAB,KAB,KAB,KJDOF,KMO)                                                004535
 CALL PR3 (A(LY,LX),B(LY,7),B(LX,7),WS2,WS1,-S1,NY,NX,NE,NE,             004536
*  KAB,KAB,KAB,KJDOF,KMO)                                                004537
 CALL PR3 (A(LX,LY),B(LX,7),B(LY,7),WS2,WS1,-S1,NX,NY,NE,NE,             004538
*  KAB,KAB,KAB,KJDOF,KMO)                                                004539
 CALL PR3 (A(LX,LX),B(LY,7),B(LY,7),WS2,WS1, S1,NX,NX,NE,NE,             004540
*  KAB,KAB,KAB,KJDOF,KMO)                                                004541
 WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)                               004542
 WRITE (NOT,3014)                                                        004543
 CALL WRITES (WS1,NE ,NE ,KMO)                                           004544
 CALL ZERO    (WS1,NE,NE,KMO)                                            004545
 CALL PR3 (A(LZ,LX),B(LZ,7),B(LY,7),WS2,WS1,-S1,NZ,NX,NE,NE,             004546
*  KAB,KAB,KAB,KJDOF,KMO)                                                004547
 CALL PR3 (A(LZ,LZ),B(LX,7),B(LY,7),WS2,WS1, S1,NZ,NZ,NE,NE,             004548
*  KAB,KAB,KAB,KJDOF,KMO)                                                004549
 CALL PR3 (A(LY,LX),B(LZ,7),B(LZ,7),WS2,WS1, S1,NY,NX,NE,NE,             004550
*  KAB,KAB,KAB,KJDOF,KMO)                                                004551
 CALL PR3 (A(LY,LZ),B(LX,7),B(LZ,7),WS2,WS1,-S1,NY,NZ,NE,NE,             004552
*  KAB,KAB,KAB,KJDOF,KMO)                                                004553
 WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)                               004554
 WRITE (NOT,3015)                                                        004555
 CALL WRITES (WS1,NE ,NE ,KMO)                                           004556
 CALL ZERO    (WS1,NE,NE,KMO)                                            004557
 CALL PR3 (A(LZ,LY),B(LX,7),B(LY,7),WS2,WS1,-S1,NZ,NY,NE,NE,             004558
*  KAB,KAB,KAB,KJDOF,KMO)                                                004559
 CALL PR3 (A(LZ,LX),B(LY,7),B(LY,7),WS2,WS1, S1,NZ,NX,NE,NE,             004560
*  KAB,KAB,KAB,KJDOF,KMO)                                                004561
 CALL PR3 (A(LY,LY),B(LX,7),B(LZ,7),WS2,WS1, S1,NY,NY,NE,NE,             004562
*  KAB,KAB,KAB,KJDOF,KMO)                                                004563
 CALL PR3 (A(LY,LX),B(LY,7),B(LZ,7),WS2,WS1,-S1,NY,NX,NE,NE,             004564
*  KAB,KAB,KAB,KJDOF,KMO)                                                004565
 WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)                               004566
 WRITE (NOT,3016)                                                        004567
 CALL WRITES (WS1,NE ,NE ,KMO)                                           004568
 CALL ZERO    (WS1,NE,NE,KMO)                                            004569
 CALL PR3 (A(LX,LY),B(LX,7),B(LZ,7),WS2,WS1,-S1,NX,NY,NE,NE,             004570
*  KAB,KAB,KAB,KJDOF,KMO)                                                004571
 CALL PR3 (A(LX,LX),B(LY,7),B(LZ,7),WS2,WS1, S1,NX,NX,NE,NE,             004572
*  KAB,KAB,KAB,KJDOF,KMO)                                                004573
 CALL PR3 (A(LZ,LY),B(LX,7),B(LX,7),WS2,WS1, S1,NZ,NY,NE,NE,             004574
*  KAB,KAB,KAB,KJDOF,KMO)                                                004575
 CALL PR3 (A(LZ,LX),B(LY,7),B(LX,7),WS2,WS1,-S1,NZ,NX,NE,NE,             004576
*  KAB,KAB,KAB,KJDOF,KMO)                                                004577
 WRITE (NTAPE1) ((WS1(I,J),J=1,NE),I=1,NE)                               004578
```

```
      WRITE (NOT,3017)                                                  004579
      CALL WRITES (WS1,NE ,NE ,KMO)                                     004580
C                                                                       004581
      IF (IDIAK .EQ. 1) GO TO 50                                        004582
      CALL READ (A,NRA,NCA,KAB,KAB)                                     004583
      CALL BTABA (A,B(1,7),NRA,NE,KAB,KAB)                              004584
      GO TO 51                                                          004585
   50 CALL ZERO (A,NE,NE,KAB)                                           004586
      DO 55 J=1,NE                                                      004587
      JP6 = J + 6                                                       004588
   55 A(J,J) = OM2(J)*OMGA2(JP6)                                        004589
   51 WRITE (NTAPE1) ((A(I,J),J=1,NE),I=1,NE)                           004590
      WRITE (NOT,3018)                                                  004591
      CALL WRITES (A  ,NE ,NE ,KAB)                                     004592
      IF (IDIAD .EQ. 1) GO TO 60                                        004593
      CALL READ (A,NRA,NCA,KAB,KAB)                                     004594
      CALL BTABA (A,B(1,7),NRA,NE,KAB,KAB)                              004595
      GO TO 61                                                          004596
   60 CALL ZERO (A,NE,NE,KAB)                                           004597
      DO 65 J=1,NE                                                      004598
      JP6 = J + 6                                                       004599
   65 OMGA2(JP6) = TWO*OM2(J)*DSQRT(OMGA2(JP6))                         004600
      READ (NIT,1002) (OM2(J),J=1,NE)                                   004601
      DO 66 J=1,NE                                                      004602
      JP6 = J + 6                                                       004603
   66 A(J,J) = OM2(J)*OMGA2(JP6)                                        004604
   61 WRITE (NTAPE1) ((A(I,J),J=1,NE),I=1,NE)                           004605
      WRITE (NOT,3019)                                                  004606
      CALL WRITES (A  ,NE ,NE ,KAB)                                     004607
C                                                                       004608
      READ (NIT,1002) (OM2(J),J=1,NE)                                   004609
      WRITE (NTAPE1) (OM2(J),J=1,NE)                                    004610
      WRITE (NOT,3020)                                                  004611
      CALL WRITES (OM2,  1,NE ,  1)                                     004612
      READ (NIT,1002) (OM2(J),J=1,NE)                                   004613
      WRITE (NTAPE1) (OM2(J),J=1,NE)                                    004614
      WRITE (NOT,3021)                                                  004615
      CALL WRITES (OM2,  1,NE ,  1)                                     004616
C                                                                       004617
      NHB = NHPOI(NBOD)                                                 004618
      NSB = NSPOI(NBOD)                                                 004619
CCC   NHB IS NO. OF P-Q HINGES ON THE BODY, NOT TO INCLUDE HINGE 1      004620
      WRITE (NTAPE1) NHB                                                004621
      DO 110 L=1,NHB                                                    004622
      READ (NIT,1001) NOH,ITYPE,JOINT                                   004623
      ISUMRY(L,1) = NOH                                                 004624
      ISUMRY(L,2) = ITYPE                                               004625
      ISUMRY(L,3) = JOINT                                               004626
      IF (NOH .LT. 2 .OR. NOH .GT. NH) GO TO 998                        004627
      LR = 6*(NOH - 2) + 1                                              004628
```

```
      LD = 7*(NOH - 2) + 1                                              004629
      IF (ITOPOL(1,NOH) .EQ. NBOD) GO TO 112                            004630
      IF (ITOPOL(2,NOH) .NE. NBOD) GO TO 998                            004631
      LR = LR + 1                                                       004632
      LD = LD + 1                                                       004633
  112 JHX = JOINT                                                       004634
      JHY = JHX + NX                                                    004635
      JHZ = JHY + NX                                                    004636
      JSX = JHZ + NX                                                    004637
      JSY = JSX + NX                                                    004638
      JSZ = JSY + NX                                                    004639
      DH(1,LD) = B(JHY,3)                                               004640
      DH(2,LD) = B(JHZ,1)                                               004641
      DH(3,LD) = B(JHX,2)                                               004642
      READ (NIT,1002) (OM2(J),J=1,3)                                    004643
      ASUMRY(L,1) = OM2(1)                                              004644
      ASUMRY(L,2) = OM2(2)                                              004645
      ASUMRY(L,3) = OM2(3)                                              004646
      CALL ROTTR  (3,ITYPE,OM2,RH(1,1,LR),DUM,DUM)                      004647
      DO 115 J=1,NE                                                     004648
      JP6 = J + 6                                                       004649
      BC(1,J) = B(JHX,JP6)                                              004650
      BC(2,J) = B(JHY,JP6)                                              004651
      BC(3,J) = B(JHZ,JP6)                                              004652
      BC(4,J) = B(JSX,JP6)                                              004653
      BC(5,J) = B(JSY,JP6)                                              004654
  115 BC(6,J) = B(JSZ,JP6)                                              004655
      WRITE (NTAPE1) NOH                                                004656
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=1,3)                           004657
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=4,6)                           004658
  110 CONTINUE                                                          004659
      WRITE (NOT,3022) NBOD                                             004660
      CALL WRITIS (ISUMRY,NHB,3,KSUMRY)                                 004661
      CALL WRITES (ASUMRY,NHB,3,KSUMRY)                                 004662
C                                                                       004663
      WRITE (NTAPE1) NSB                                                004664
      IF (NSB .EQ. 0) RETURN                                            004665
      DO 120 L=1,NSB                                                    004666
      READ (NIT,1001) NOS,ITYPE,JOINT                                   004667
      ISUMRY(L,1) = NOS                                                 004668
      ISUMRY(L,2) = ITYPE                                               004669
      ISUMRY(L,3) = JOINT                                               004670
      IF (IFTSMW(NOS) .NE. NBOD) GO TO 998                              004671
      LR = 2*NOS - 1                                                    004672
      JHX = JOINT                                                       004673
      JHY = JHX + NX                                                    004674
      JHZ = JHY + NX                                                    004675
      JSX = JHZ + NX                                                    004676
      JSY = JSX + NX                                                    004677
      JSZ = JSY + NX                                                    004678
```

```
      DS(1,LR) = B(JHY,3)                                                    004679
      DS(2,LR) = B(JHZ,1)                                                    004680
      DS(3,LR) = B(JHX,2)                                                    004681
      READ (NIT,1002) (OM2(J),J=1,3)                                         004682
      ASUMRY(L,1) = OM2(1)                                                   004683
      ASUMRY(L,2) = OM2(2)                                                   004684
      ASUMRY(L,3) = OM2(3)                                                   004685
      CALL ROTTR  (3,ITYPE,OM2,RS(1,1,LR),DUM,DUM)                           004686
      DO 125 J=1,NE                                                          004687
      JP6 = J + 6                                                            004688
      BC(1,J) = B(JHX,JP6)                                                   004689
      BC(2,J) = B(JHY,JP6)                                                   004690
      BC(3,J) = B(JHZ,JP6)                                                   004691
      BC(4,J) = B(JSX,JP6)                                                   004692
      BC(5,J) = B(JSY,JP6)                                                   004693
  125 BC(6,J) = B(JSZ,JP6)                                                   004694
      WRITE (NTAPE1) NOS                                                     004695
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=1,3)                                004696
      WRITE (NTAPE1) ((BC(I,J),J=1,NE),I=4,6)                                004697
  120 CONTINUE                                                               004698
      WRITE (NOT,3023) NBOD                                                  004699
      CALL WRITIS (ISUMRY,NSB,3,KSUMRY)                                      004700
      CALL WRITES (ASUMRY,NSB,3,KSUMRY)                                      004701
C                                                                            004702
      RETURN                                                                 004703
C                                                                            004704
  998 WRITE (NOT,2001)                                                       004705
 2001 FORMAT (1H1,31HTOPOLOGY ERROR, PROGRAM STOPPED)                        004706
      STOP                                                                   004707
  999 WRITE (NOT,2002)                                                       004708
 2002 FORMAT (1H1,47HMORE THAN NMDBOD MODES SELECTED,PROGRAM STOPPED)        004709
      STOP                                                                   004710
C                                                                            004711
      END                                                                    004712
```

```
[HDG,P     MSMODL                                                                -004713
[FOR,IS    MSMODL                                                                -004714
      COMPILER (XM=1),(EQUIV=CMN)                                                -004715
      SUBROUTINE MSMODL (NBOD)                                                    004716
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                         -004717
C                                                                                 004718
            COMMON /NHNS  /                                                       004719
     *      NHPOI( 5), NSPOI( 5)                                                1204720
            COMMON /NUMBRS/                                                       004721
     *      ZRO,ONE,TWO,TRES                                                     004722
            COMMON /SPECIF/                                                      004723
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),           1604724
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),                  1704725
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5),       1804726
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                                 1904727
            COMMON /SUMMRY/                                                       004728
     *      ASUMRY(10,6),ISUMRY(10,3),KSUMRY                                    9804729
            COMMON /TAPENO/                                                       004730
     *      NTAPE1,NTAPE2,NTAPE3                                                 004731
C                                                                                 004732
      DIMENSION A( 105, 6),B( 105, 6),C( 105, 6),AMU( 12, 12)                    6404733
      DIMENSION WS( 105,13),UVEC( 105),WV(3)                                     6504734
C                                                                                 004735
      DATA  KJOINT, KMODE                                                         004736
     *   /     105,     6   /                                                   6604737
      DATA NIT,NOT /5,6/                                                          004738
C                                                                                 004739
 1001 FORMAT(16I5)                                                                004740
 1002 FORMAT(8D10.3)                                                              004741
 3022 FORMAT (//5X 9HFOR BODY I3,29H THE P-Q HINGE NO., THE EULER                 004742
     * 57H ROTATION TYPE AND THE JOINT NO. CORRESPONDING TO THE P-Q,/             004743
     *  5X54H HINGE APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS                004744
     *    49HFOLLOWED BY AN ARRAY CONTAINING EULER ANGLES THAT,/                  004745
     * 5X44H POSITION THE HINGE TRIAD WRT THE BODY TRIAD)                         004746
 3023 FORMAT (//5X 9HFOR BODY I3,32H THE SENSOR POINT NO., THE EULER              004747
     *60H ROTATION TYPE AND THE JOINT NO. CORRESPONDING TO THE SENSOR,/          004748
     *6X53HPOINT APPEAR IN THE FOLLOWING INTEGER ARRAY WHICH IS                  004749
     *    49HFOLLOWED BY AN ARRAY CONTAINING EULER ANGLES THAT,/                 004750
     * 5X45H POSITION THE SENSOR TRIAD WRT THE BODY TRIAD)                       004751
C                                                                                 004752
      KA    = KJOINT                                                            004753
      KB    = KJOINT                                                            004754
      KC    = KJOINT                                                            004755
      KWS   = KJOINT                                                            004756
      KAMU = KMODE + 6                                                          004757
C                                                                                 004758
C***  INITIALIZE NTAPE2, NTAPE3                                                   004759
      REWIND NTAPE2                                                               004760
      REWIND NTAPE3                                                               004761
C                                                                                 004762
```

```
C*************************************************                004763
C                                                               004764
C***   NJ = NUMBER OF MASS POINTS ON BODY I                      004765
C***   NE = NUMBER OF ELASTIC MODES RETAINED FOR BODY I          004766
      NE = IRGFLX(NBOD)                                         004767
      NE6 = NE + 6                                              004768
C                                                               004769
C***  READ FORMA TAPE OR CARDS - CREATE NTAPE2                  004770
C                                                               004771
C     MASSES                                                    004772
      CALL READ (A,N1,N2,KJOINT,KMODE)                          004773
      NJ = N1                                                   004774
      DO 2 I=1,NJ                                               004775
      IF(A(I,1) .LT. ZRO) GO TO 995                             004776
    2 CONTINUE                                                  004777
      WRITE(NTAPE2) (A(I,1),I=1,NJ)                             004778
C                                                               004779
C     INERTIAS                                                  004780
      CALL READ (A,N1,N2,KJOINT,KMODE)                          004781
      IF(N1 .NE. NJ  .OR.  N2 .NE.  6) GO TO 999                004782
      WRITE(NTAPE2) ((A(I,J),J=1,6),I=1,NJ)                     004783
C                                                               004784
C     STATIC MOMENTS - GEOMETRIC COORDINATES                    004785
      DO 5 K=1,2                                                004786
      CALL READ (A,N1,N2,KJOINT,KMODE)                          004787
      IF(N1 .NE. NJ  .OR.  N2 .NE.  3) GO TO 999                004788
      WRITE(NTAPE2) ((A(I,J),J=1,3),I=1,NJ)                     004789
    5 CONTINUE                                                  004790
C                                                               004791
C     MODAL AMPLITUDES                                          004792
      DO 10 K=1,6                                               004793
      CALL READ (A,N1,N2,KJOINT,KMODE)                          004794
      IF(N1 .NE. NJ  .OR.  N2 .NE. NE) GO TO 999                004795
      WRITE(NTAPE2) ((A(I,J),J=1,NE),I=1,NJ)                    004796
   10 CONTINUE                                                  004797
C                                                               004798
C     STIFFNESS - DAMPING                                       004799
      DO 20 K=1,2                                               004800
      CALL READ (A,N1,N2,KJOINT,KMODE)                          004801
      IF(N1 .NE. NE  .OR.  N2 .NE. NE) GO TO 999                004802
      WRITE(NTAPE2) ((A(I,J),J=1,NE),I=1,NE)                    004803
   20 CONTINUE                                                  004804
C                                                               004805
      DO 47 I=1,NJ                                              004806
   47 UVEC(I) = ONE                                             004807
C                                                               004808
      REWIND NTAPE2                                             004809
      NREC2 = 0                                                 004810
C                                                               004811
C***  CREATE NTAPE3                                             004812
```

```
      CALL CRET3 (NREC2,NJ,NE,A,B,WS,KA,KB,KWS)                          004813
C                                                                        004814
      REWIND NTAPE3                                                      004815
      NREC3 = 0                                                          004816
C                                                                        004817
C***  FORM MUZERO MATRIX                                                 004818
      CALL ZERO(AMU,NE6,NE6,KAMU)                                        004819
C                                                                        004820
      CALL CREMUO(NREC3,NJ,UVEC,A,WS,AMU,KA,KWS,KAMU)                    004821
C                                                                        004822
C***  FORM AO AND DO COEFFICIENTS                                        004823
      CALL CREADO(NREC3,NJ,NE,UVEC,A,B,C,WS,AMU,KA,KB,KC,KWS,KAMU)       004824
C                                                                        004825
C***  FORM E COEFFICIENTS                                                004826
      CALL CREE  (NREC3,NREC2,NJ,NE,A,B,C,AMU,KA,KB,KC,KAMU)             004827
C                                                                        004828
C***  SYMMETRIZE AMU                                                     004829
      DO 48 I=1,NE6                                                      004830
      DO 48 J=I,NE6                                                      004831
   48 AMU(J,I) = AMU(I,J)                                                004832
C                                                                        004833
      CALL WRITE(AMU,NE6,NE6,3HMUO,KAMU)                                 004834
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE6),I=1,NE6)                         004835
C                                                                        004836
C***  FORM ACOF                                                          004837
      CALL CREA  (NREC3,NJ,NE,UVEC,A,B,KA,KB,KWS)                        004838
C                                                                        004839
C***  FORM BCOF                                                          004840
      CALL CREB  (NREC3,NREC2,NJ,NE,A,B,WS,KA,KB,KWS)                    004841
C                                                                        004842
C***  FORM CCOF                                                          004843
      CALL CREC  (NREC3,NREC2,NJ,NE,A,B,C,AMU,KA,KB,KC,KAMU)             004844
C                                                                        004845
C***  FETCH AND STORE STIFFNESS AND DAMPING                              004846
      CALL FETCH(NTAPE2,11,NREC2,AMU,NE,NE,KAMU)                         004847
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                           004848
      CALL FETCH(NTAPE2,12,NREC2,AMU,NE,NE,KAMU)                         004849
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,NE)                           004850
C                                                                        004851
C***  READ AND STORE INITIAL CONDITIONS                                  004852
      READ(NIT,1002) (A(1,J),J=1,NE)                                     004853
      READ(NIT,1002) (A(2,J),J=1,NE)                                     004854
      CALL WRITE(A(1,1),1,NE,4HXEO ,KA)                                  004855
      CALL WRITE(A(2,1),1,NE,4HXEOD,KA)                                  004856
      WRITE(NTAPE1) (A(1,J),J=1,NE)                                      004857
      WRITE(NTAPE1) (A(2,J),J=1,NE)                                      004858
C                                                                        004859
C***  HINGE LOOP  ******************************************             004860
C                                                                        004861
      NHB = NHPOI(NBOD)                                                  004862
```

```
      WRITE(NTAPE1) NHB                                                 004863
C                                                                       004864
      DO 150 L=1,NHB                                                    004865
      READ(NIT,1001) NOH,ITYPE,JOINT                                    004866
      ISUMRY(L,1) = NOH                                                 004867
      ISUMRY(L,2) = ITYPE                                               004868
      ISUMRY(L,3) = JOINT                                               004869
      IF(NOH .LT. 2  .OR.  NOH .GT. NH) GO TO 996                       004870
      LR = 6*(NOH-2) + 1                                                004871
      LD = 7*(NOH-2) + 1                                                004872
      IF(ITOPOL(1,NOH) .EQ. NBOD) GO TO 152                             004873
      IF(ITOPOL(2,NOH) .NE. NBOD) GO TO 996                             004874
      LR = LR+1                                                         004875
      LD = LD+1                                                         004876
  152 CONTINUE                                                          004877
C                                                                       004878
      DH(1,LD) = WS(JOINT,11)                                           004879
      DH(2,LD) = WS(JOINT,12)                                           004880
      DH(3,LD) = WS(JOINT,13)                                           004881
C                                                                       004882
C***  READ ANGLES                                                       004883
      READ(NIT,1002) (WV(J),J=1,3)                                      004884
      ASUMRY(L,1) =  WV(1)                                              004885
      ASUMRY(L,2) =  WV(2)                                              004886
      ASUMRY(L,3) =  WV(3)                                              004887
C                                                                       004888
      CALL ROTTR  (3,ITYPE,WV,RH(1,1,LR),DUM,DUM)                       004889
C                                                                       004890
C***  READ HX,HY,HZ                                                     004891
      CALL FETCH(NTAPE2, 5,NREC2,A,NJ,NE,KA)                            004892
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)                            004893
      CALL FETCH(NTAPE2, 7,NREC2,C,NJ,NE,KC)                            004894
C                                                                       004895
      DO 154 J=1,NE                                                     004896
      AMU(1,J) = A(JOINT,J)                                             004897
      AMU(2,J) = B(JOINT,J)                                             004898
  154 AMU(3,J) = C(JOINT,J)                                             004899
C                                                                       004900
C***  READ SIGX,SIGY,SIGZ                                               004901
      CALL FETCH(NTAPE2, 8,NREC2,A,NJ,NE,KA)                            004902
      CALL FETCH(NTAPE2, 9,NREC2,B,NJ,NE,KB)                            004903
      CALL FETCH(NTAPE2,10,NREC2,C,NJ,NE,KC)                            004904
C                                                                       004905
      DO 155 J=1,NE                                                     004906
      AMU(4,J) = A(JOINT,J)                                             004907
      AMU(5,J) = B(JOINT,J)                                             004908
  155 AMU(6,J) = C(JOINT,J)                                             004909
C                                                                       004910
      WRITE(NTAPE1) NOH                                                 004911
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,3)                           004912
```

```
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=4,6)                          004913
C                                                                      004914
  150 CONTINUE                                                         004915
      WRITE (NOT,3022) NBOD                                            004916
      CALL WRITIS (ISUMRY,NHB,3,KSUMRY)                                004917
      CALL WRITES (ASUMRY,NHB,3,KSUMRY)                                004918
C                                                                      004919
C***  SENSOR LOOP  ******************************************          004920
C                                                                      004921
      NSB = NSPOI(NBOD)                                                004922
      WRITE(NTAPE1) NSB                                                004923
      IF(NSB .EQ. 0) RETURN                                            004924
C                                                                      004925
      DO 160 L=1,NSB                                                   004926
      READ(NIT,1001) NOS,ITYPE,JOINT                                   004927
      ISUMRY(L,1) = NOS                                                004928
      ISUMRY(L,2) = ITYPE                                              004929
      ISUMRY(L,3) = JOINT                                              004930
      IF(IFTSMW(NOS) .NE. NBOD) GO TO 996                              004931
      LR = 2*NOS - 1                                                   004932
      DS(1,LR) = WS(JOINT,11)                                          004933
      DS(2,LR) = WS(JOINT,12)                                          004934
      DS(3,LR) = WS(JOINT,13)                                          004935
C                                                                      004936
C***  READ ANGLES                                                      004937
      READ(NIT,1002) (WV(J),J=1,3)                                     004938
      ASUMRY(L,1) =  WV(1)                                             004939
      ASUMRY(L,2) =  WV(2)                                             004940
      ASUMRY(L,3) =  WV(3)                                             004941
C                                                                      004942
      CALL ROTTR (3,ITYPE,WV,RS(1,1,LR),DUM,DUM)                       004943
C                                                                      004944
C***  READ HX,HY,HZ                                                    004945
      CALL FETCH(NTAPE2, 5,NREC2,A,NJ,NE,KA)                           004946
      CALL FETCH(NTAPE2, 6,NREC2,B,NJ,NE,KB)                           004947
      CALL FETCH(NTAPE2, 7,NREC2,C,NJ,NE,KC)                           004948
C                                                                      004949
      DO 164 J=1,NE                                                    004950
      AMU(1,J) = A(JOINT,J)                                            004951
      AMU(2,J) = B(JOINT,J)                                            004952
  164 AMU(3,J) = C(JOINT,J)                                            004953
C                                                                      004954
C***  READ SIGX,SIGY,SIGZ                                              004955
      CALL FETCH(NTAPE2, 8,NREC2,A,NJ,NE,KA)                           004956
      CALL FETCH(NTAPE2, 9,NREC2,B,NJ,NE,KB)                           004957
      CALL FETCH(NTAPE2,10,NREC2,C,NJ,NE,KC)                           004958
C                                                                      004959
      DO 165 J=1,NE                                                    004960
      AMU(4,J) = A(JOINT,J)                                            004961
      AMU(5,J) = B(JOINT,J)                                            004962
```

```
  165 AMU(6,J) = C(JOINT,J)                                              004963
C                                                                        004964
      WRITE(NTAPE1) NOS                                                  004965
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=1,3)                            004966
      WRITE(NTAPE1) ((AMU(I,J),J=1,NE),I=4,6)                            004967
C                                                                        004968
  160 CONTINUE                                                           004969
      WRITE (NOT,3023) NBOD                                              004970
      CALL WRITIS (ISUMRY,NSB,3,KSUMRY)                                  004971
      CALL WRITES (ASUMRY,NSB,3,KSUMRY)                                  004972
C                                                                        004973
      RETURN                                                             004974
C                                                                        004975
  995 WRITE(NOT,2001)                                                    004976
 2001 FORMAT(1HI,45HNEGATIVE OR ZERO LUMPED MASS, PROGRAM STOPPED)       004977
      STOP                                                               004978
  996 WRITE(NOT,2003)                                                    004979
 2003 FORMAT(1HI,31HTOPOLOGY ERROR, PROGRAM STOPPED)                     004980
      STOP                                                               004981
  999 WRITE(NOT,2004)                                                    004982
 2004 FORMAT(1HI,41HERROR IN INPUT TO MSMODL, PROGRAM STOPPED)           004983
      STOP                                                               004984
C                                                                        004985
      END                                                                004986
```

```
[HDG,P    MULTA                                                          -004987
[FOR,IS   MULTA                                                          -004988
      COMPILER (XM=1),(EQUIV=CMN)                                        -004989
      SUBROUTINE MULTA (AZ,B,NRA,NRB,NCB,KAZ,KB)                          004990
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -004991
      DIMENSION AZ(KAZ,1), B(KB,1),W(150)                                7004992
      DATA NOT / 6/                                                       004993
C                                                                         004994
C  MATRIX MULTIPLICATION. A * B = Z.                                      004995
C  USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.                       004996
C  AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE     004997
C  LARGER OF A OR Z.                                                      004998
C  CALLS FORMA SUBROUTINE ZZBOMB.                                         004999
C  THE MAXIMUM SIZE IS                                                    005000
C     NRB = XXX                                                           005001
C  DEVELOPED BY C S BODLEY.  JANUARY 1965.                                005002
C  LAST REVISION BY R F HRUDA.  JUNE 1972.                                005003
C                                                                         005004
C     SUBROUTINE ARGUMENTS                                                005005
C  AZ  = INPUT  MATRIX. SIZE(NRA,NRB).                                    005006
C      = OUTPUT RESULT MATRIX. SIZE(NRA,NCB).                             005007
C  B   = INPUT  MATRIX. SIZE(NRB,NCB)                                     005008
C  NRA = INPUT  NUMBER OF ROWS OF MATRICES A,Z.                           005009
C  NRB = INPUT  NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A. MAX=500.    005010
C  NCB = INPUT  NUMBER OF COLS OF MATRICES B,Z.                           005011
C  KAZ = INPUT  ROW DIMENSION OF AZ IN CALLING PROGRAM.                   005012
C  KB  = INPUT  ROW DIMENSION OF B IN CALLING PROGRAM.                    005013
C                                                                         005014
      IF (NRB .GT. 150) GO TO 999                                        7105015
C                                                                         005016
      DO 40 I=1,NRA                                                       005017
      DO 20 K=1,NRB                                                       005018
   20 W(K) = AZ(I,K)                                                      005019
      DO 40 J=1,NCB                                                       005020
      S = 0.D 0                                                           005021
      DO 30 K=1,NRB                                                       005022
   30 S = S + W(K)*B(K,J)                                                 005023
   40 AZ(I,J) = S                                                         005024
      RETURN                                                              005025
C                                                                         005026
  999 WRITE (NOT,1001)                                                    005027
 1001 FORMAT (1H1,31HERROR IN MULTA, PROGRAM STOPPED)                     005028
      STOP                                                                005029
      END                                                                 005030
```

```
[HDG,P    MULTB                                                          -005031
[FOR,IS   MULTB                                                          -005032
      COMPILER (XM=1),(EQUIV=CMN)                                        -005033
      SUBROUTINE MULTB (A,BZ,NRA,NRB,NCB,KA,KBZ)                          005034
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -005035
      DIMENSION A(KA,1),BZ(KBZ,1)                                         005036
      COMMON /LWRKV1/ W( 50)                                            48005037
C                                                                         005038
C MATRIX MULTIPLICATION.  A * B = Z.                                      005039
C  USES TWO WORK SPACES. RESULT (Z) IS PLACED IN B.                       005040
C  BZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE     005041
C  LARGER OF B OR Z.                                                      005042
C  THE MAXIMUM SIZE IS                                                    005043
C      NRB = 500.                                                         005044
C                                                                         005045
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                 005046
C                                                                         005047
C  A        = INPUT  MATRIX. SIZE (NRA,NRB).                            005048
C  BZ       = INPUT  MATRIX.  SIZE (NRB,NCB).                           005049
C           = OUTPUT RESULT MATRIX.  SIZE (NRA,NCB).                    005050
C  NRA      = INPUT  NUMBER OF ROWS OF MATRICES A,Z.                    005051
C  NRB      = INPUT  NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A.  MAX =  005052
C  NCB      = INPUT  NUMBER OF COLS OF MATRICES B,Z.                    005053
C  KA       = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.             005054
C  KBZ      = INPUT  ROW DIMENSION OF BZ IN CALLING PROGRAM.            005055
C                                                                         005056
      DO 40 J=1,NCB                                                       005057
      DO 20 K=1,NRB                                                       005058
   20 W(K) = BZ(K,J)                                                      005059
      DO 40 I=1,NRA                                                       005060
      S = 0.D0                                                            005061
      DO 30 K=1,NRB                                                       005062
   30 S = S + A(I,K)*W(K)                                                 005063
   40 BZ(I,J) = S                                                         005064
C                                                                         005065
      RETURN                                                              005066
      END                                                                 005067
```

```
[HDG,P     MULT3                                                          -005068
[FOR,IS    MULT3                                                          -005069
       COMPILER (XM=1),(EQUIV=CMN)                                        -005070
       SUBROUTINE MULT3(A,B,Z,NRA,NRB,NCB,KRA,KRB,KRZ)                     005071
       IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -005072
       DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1),WR(100)                       8405073
C                                                                          005074
       DO 20 I=1,NRA                                                       005075
       DO 15 J=1,NRB                                                       005076
   15  WR(J) = A(I,J)                                                      005077
       DO 20 J=1,NCB                                                       005078
       S = 0.D 0                                                           005079
       DO 30 K=1,NRB                                                       005080
   30  S = S + WR(K)*B(K,J)                                                005081
   20  Z(I,J) = S                                                          005082
C                                                                          005083
       RETURN                                                              005084
       END                                                                 005085
```

```
[HDG,P     MULTAD                                                    -005086
[FOR,IS    MULTAD                                                    -005087
      COMPILER (XM=1),(EQUIV=CMN)                                    -005088
      SUBROUTINE MULTAD (A,B,Z,NRA,NRB,NCB,KRA,KRB,KRZ)               005089
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                             -005090
      DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1),WR(100)                   8505091
C                                                                     005092
      DO 20 I=1,NRA                                                   005093
      DO 15 J=1,NRB                                                   005094
   15 WR(J) = A(I,J)                                                  005095
      DO 20 J=1,NCB                                                   005096
      S = 0.D 0                                                       005097
      DO 30 K=1,NRB                                                   005098
   30 S = S + WR(K)*B(K,J)                                            005099
   20 Z(I,J) = Z(I,J) + S                                             005100
C                                                                     005101
      RETURN                                                          005102
      END                                                             005103
```

```
[HDG,P    NIPLOT                                                        -005104
[FOR,IS   NIPLOT                                                        -005105
      COMPILER (XM=1),(EQUIV=CMN)                                       -005106
      SUBROUTINE NIPLOT (TITLE,DBMIN,DBMAX)                             -005107
C                                                                       -005108
C ***                                                                   -005109
C *** MSFC UNIVAC 1108 VERSION ***                                      -005110
C ***                                                                   -005111
C-----SUBROUTINE FORMS NICHOLS PLOT                                     -005112
C                                                                       -005113
C             -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                -005114
C                                                                       -005115
C  TITLE = INPUT ALPHA NUMERIC TITLE                                    -005116
C  DBMIN = INPUT LOWER DB LIMIT TO PLOT                                 -005117
C  DBMAX = INPUT UPPER DB LIMIT TO PLOT                                 -005118
C                                                                       -005119
      COMMON /LSTART/ IRUNNO, IDATE, NPAGE                              -005120
      COMMON /PSTUFF/                                                   -005121
     *            SAVEO(500), SAVEP(500), SAVED(500), SAVEA(500), KSAVE -005122
      COMMON /ADDPLT/ X(500),Y(500),DUMMY(500)                          -005123
C                                                                       -005124
      DIMENSION TITLE(1),TX(12),TY(12)                                  -005125
C                                                                       -005126
      EQUIVALENCE (IRUNNO,RUNNO)                                        -005127
C                                                                       -005128
      TX( 1) = 6H                                                       -005129
      DO 5 I=1,10                                                       -005130
    5 TX(I+1) = TITLE(I)                                                -005131
      TX(12) = 6H                                                       -005132
C                                                                       -005133
      TY( 1) = 6HNICHOL                                                 -005134
      TY( 2) = 6HS PLOT                                                 -005135
      TY( 3) = 6H                                                       -005136
      TY( 4) = 6HAMP RA                                                 -005137
      TY( 5) = 6HTIO IN                                                 -005138
      TY( 6) = 6H DB VS                                                 -005139
      TY( 7) = 6H PHASE                                                 -005140
      TY( 8) = 6H IN DE                                                 -005141
      TY( 9) = 6HG                                                      -005142
      TY(10) = 6H                                                       -005143
      TY(11) = 6H                                                       -005144
      TY(12) = RUNNO                                                    -005145
C                                                                       -005146
      CALL PLOTSS(DBMAX,DBMIN,YTOP,YBOT)                                -005147
C                                                                       -005148
      XLFT = 0.0                                                        -005149
      XRGT = 360.                                                       -005150
C                                                                       -005151
      IFR = 0                                                           -005152
      IFL = 0                                                           -005153
```

```
      KNT = 0                                                                -005154
      DO 80 I=1,KSAVE                                                        -005155
      DB = SAVED(I)                                                          -005156
      PH = SAVEP(I)                                                          -005157
      IF(DB .GE. DBMIN  .AND.  DB .LE. DBMAX) GO TO 81                       -005158
      IF(IFL .EQ. 0) GO TO 80                                                -005159
   75                 IF(IFR .EQ. 0) CALL QUIK3L(-1,XLFT,XRGT,              -005160
     *          YBOT,YTOP,35,TX,TY,-KNT,X,Y)                                -005161
                      IF(IFR .EQ. 1) CALL QUIK3L( 0,XLFT,XRGT,              -005162
     *          YBOT,YTOP,35,TX,TY,-KNT,X,Y)                                -005163
      IFL = 0                                                                -005164
      IFR = 1                                                                -005165
      KNT = 0                                                                -005166
      GO TO 80                                                               -005167
   81 KNT = KNT + 1                                                          -005168
      X(KNT) = PH                                                            -005169
      Y(KNT) = DB                                                            -005170
      IFL = 1                                                                -005171
      IF(I .EQ. KSAVE) GO TO 75                                              -005172
   80 CONTINUE                                                               -005173
C                                                                            -005174
      RETURN                                                                 -005175
      END                                                                    -005176
```

```
[HDG,P     NUMS                                                          -005177
[FOR,IS    NUMS                                                          -005178
      COMPILER (XM=1),(EQUIV=CMN)                                        -005179
      SUBROUTINE NUMS (A,D,B,R1R,R1I,R2R,R2I,PTOL,                        005180
     *                 GAIN,IFLG,NN,NZRO,IY,NA,KA)                      005181
C                                                                        005182
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -005183
C                                                                        005184
C   SUBROUTINE DETERMINES NUM(S)                                         005185
C   CALLS QR PACKAGE                                                     005186
C                                                                        005187
      COMMON /DRATIO/                                                    005188
     *     IFL1,IFL2,DRVEC(150)                                         10005189
C                                                                        005190
      DIMENSION A(KA,1),B(1),R1R(1),R1I(1),R2R(1),R2I(1)                 005191
      DIMENSION D(KA,1)                                                  005192
C                                                                        005193
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                005194
C                                                                        005195
C  A       = INPUT  TRANSFORMED PARTIAL DERIVATIVE MATRIX.  NA,NA        005196
C  B       = INPUT  COEFF. OF INPUT TF FOR Q(OUT)/Q(IN).  NA,1           005197
C  R1R     = OUTPUT REAL ROOTS OF FIRST TERM IN NUMERATOR.  NA-1         005198
C  R1I     = OUTPUT IMAGINARY ROOTS OF FIRST TERM IN NUMERATOR.          005199
C  R2R     = OUTPUT  REAL ROOTS OF SECOND TERM IN NUMERATOR.  NA-1       005200
C  R2I     = IMAGINARY ROOTS OF SECOND TERM IN NUMERATOR.                005201
C            FOR INVERSE SHIFT.                                          005202
C  PTOL    = INPUT TOLERANCE TO REMOME THE P(I) = 0 ROOTS                005203
C  GAIN    = OUTPUT GAIN OF NUMERATOR.                                   005204
C  NN      = OUTPUT NUMBER OF NUMERATOR TERMS.  EITHER 1 OR 2.           005205
C  IY      = INPUT  COL LOCATION OF DESIRED Q(OUT) --LOCAL--             005206
C  KA      = INPUT  ROW DIMENSION SIZE OF A IN CALLING PROGRAM.          005207
C  IFLG    = FLAG TO SET FOR ZERO GAIN.                                  005208
C          = 0 ZERO GAIN DETECTED.                                       005209
C          =1 GAIN NOT ZERO.                                             005210
C                                                                        005211
      CON = -DSQRT(3.D0)                                                 005212
C                                                                        005213
      IFL1 = 1                                                           005214
      IFL2 = 0                                                           005215
      IFLG = 1                                                           005216
C                                                                        005217
C   FORM AUGMENTED A MATRIX                                              005218
C   REPLACE COL IY OF A WITH COL NA OF A AND                             005219
C   PUT B INTO COL NA OF A                                               005220
C                                                                        005221
      DO 10 I=1,NA                                                       005222
      A(I,IY) = A(I,NA)                                                  005223
   10 A(I,NA) = -B(I)                                                    005224
C    INTERCHANGE ROW IY OF A WITH ROW NA OF A                            005225
      DO 15 J=1,NA                                                       005226
```

```
      Z = A(IY,J)                                                    005227
      A(IY,J) = A(NA,J)                                              005228
   15 A(NA,J) = Z                                                    005229
C     OBTAIN PIVOTAL ELE AND CK FOR ZERO                             005230
      Z = A(NA,NA)                                                   005231
      IF (Z .EQ. 0.D0) GO TO 50                                      005232
      GAIN = -Z                                                      005233
      NN = 1                                                         005234
      DO 20 J=1,NA                                                   005235
   20 A(NA,J) = A(NA,J) / Z                                          005236
      NA2 = NA-1                                                     005237
      DO 30 I=1,NA2                                                  005238
      DO 30 J=1,NA                                                   005239
   30 A(I,J) = A(I,J) - A(I,NA) * A(NA,J)                            005240
      CALL WRITE (A,NA,NA,4HANUM,KA)                                 005241
      CALL QRDRVR (A,NA2,R1R,R1I,KA)                                 005242
      NZRO = NA2                                                     005243
      GO TO 100                                                      005244
   50 CONTINUE                                                       005245
C                                                                    005246
      NN = 1                                                         005247
      NA2 = NA-1                                                     005248
C                                                                    005249
C     FORM (A - I*CON)                                               005250
C                                                                    005251
      DO 55 I=1,NA2                                                  005252
   55 A(I,1) = A(I,I) - CON                                          005253
C                                                                    005254
      CALL GAUSSI (A,D,NA,KA)                                        005255
      CALL WRITE(DRVEC,1,NA,6HDRATIO,1)                               05256
      IF (IFL2 .EQ. 0) GO TO 57                                      005257
      IFL2 = 0                                                       005258
      IFLG = 0                                                       005259
C                                                                    005260
C     ZERO GAIN ENCOUNTERED IN GAUSSI.                               005261
C                                                                    005262
      RETURN                                                         005263
C                                                                    005264
   57 CONTINUE                                                       005265
C                                                                    005266
C                                                                    005267
      GAIN =  DRVEC(1)                                               005268
      DO 58 I=2,NA                                                   005269
   58 GAIN = GAIN * DRVEC(I)                                         005270
C                                                                    005271
      DO 59 I=1,NA                                                   005272
   59 D(I,NA) = 0.D0                                                 005273
C                                                                    005274
      CALL QRDRVR (D,NA,R2R,R2I,KA)                                  005275
C                                                                    005276
```

```
C       REMOVE THE P(I) = 0. ROOTS.                                      005277
C                                                                        005278
        CALL SIFT (R2R,NA,PTOL)                                          005279
        CALL SIFT (R2I,NA,PTOL)                                          005280
C                                                                        005281
        K=0                                                              005282
        DO 60 I=1,NA                                                     005283
        IF (R2R(I) .EQ. 0.D0  .AND.                                      005284
     *     R2I(I) .EQ. 0.D0)        GO TO 60                             005265
        K=K+1                                                            005286
        R1R(K) = R2R(I)                                                  005287
        R1I(K) = R2I(I)                                                  005288
   60 CONTINUE                                                           005289
C                                                                        005290
        NZRO = K                                                         005291
        NEXP = NA-NZRO                                                   005292
        PIP = 1.D0                                                       005293
        I = 0                                                           -005294
        DO 70 III=1,NZRO                                                -005295
        I = I+1                                                         -005296
        IF(I .GT. NZRO) GO TO 71                                        -005297
        IF (R1I(I) .EQ. 0.D0) GO TO 65                                  -005298
        PIP = PIP * (R1R(I)**2 + R1I(I)**2)                             -005299
        I=I+1                                                           -005300
        GO TO 70                                                        -005301
   65 PIP = PIP * R1R(I)                                                -005302
   70 CONTINUE                                                          -005303
   71 CONTINUE                                                          -005304
C                                                                        005305
        GAIN = GAIN * PIP                                                005306
        IF (NEXP .EQ. 0) GO TO 75                                        005307
        DO 72 I=1,NEXP                                                   005308
   72 GAIN = -1.D0*GAIN                                                  005309
   75 CONTINUE                                                           005310
C                                                                        005311
C       REMOVE THE SHIFT VALUE TO OBTAIN TRUE ROOTS.                     005312
C                                                                        005313
        DO 80 I=1,NZRO                                                   005314
        RMOD = R1R(I)**2 + R1I(I)**2                                     005315
        R1R(I) = R1R(I)/RMOD + CON                                       005316
   80 R1I(I) = -R1I(I)/RMOD                                              005317
C                                                                        005318
  100 CONTINUE                                                           005319
        RETURN                                                           005320
        END                                                              005321
```

```
[HDG,P    NYPLOT                                                        -005322
[FOR,IS   NYPLOT                                                        -005323
      COMPILER (XM=1),(EQUIV=CMN)                                       -005324
      SUBROUTINE NYPLOT (TITLE,AMIN,AMAX)                               -005325
C                                                                       -005326
C ***                                                                   -005327
C *** MSFC UNIVAC 1108 VERSION ***                                      -005328
C ***                                                                   -005329
C-----SUBROUTINE FORMS NYQUIST PLOT                                     -005330
C                                                                       -005331
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----               -005332
C                                                                       -005333
C  TITLE = INPUT ALPHA NUMERIC TITLE                                    -005334
C  AMIN  = MINIMUM AMPLITUDE TO PLOT                                    -005335
C  AMAX  = MAXIMUM AMPLITUDE TO PLOT                                    -005336
C                                                                       -005337
      COMMON /LSTART/ IRUNNO, IDATE, NPAGE                              -005338
      COMMON /PSTUFF/                                                   -005339
     *            SAVEO(500), SAVEP(500), SAVED(500), SAVEA(500), KSAVE -005340
      COMMON /ADDPLT/ X(500),Y(500),DUMMY(500)                          -005341
C                                                                       -005342
      DIMENSION TITLE(1),TX(12),TY(12)                                  -005343
C                                                                       -005344
      EQUIVALENCE (IRUNNO,RUNNO)                                        -005345
C                                                                       -005346
      TX( 1) = 6H                                                       -005347
      DO 5 I=1,10                                                       -005348
    5 TX(I+1) = TITLE(I)                                                -005349
      TX(12) = 6H                                                       -005350
C                                                                       -005351
      TY( 1) = 6HNYQUIS                                                 -005352
      TY( 2) = 6HT PLOT                                                 -005353
      TY( 3) = 6H                                                       -005354
      TY( 4) = 6HAMPLIT                                                 -005355
      TY (5) = 6HUDE -                                                  -005356
      TY( 6) = 6HIMAG P                                                 -005357
      TY( 7) = 6HART VS                                                 -005358
      TY( 8) = 6H REAL                                                  -005359
      TY( 9) = 6HPART -                                                 -005360
      TY(10) = 6H                                                       -005361
      TY(11) = 6H                                                       -005362
      TY(12) = RUNNO                                                    -005363
C                                                                       -005364
      CALL PLOTSS(AMAX,-AMAX,YTOP,YBOT)                                 -005365
      CALL PLOTSS(AMAX,-AMAX,XRGT,XLFT)                                 -005366
C                                                                       -005367
      IFR = 0                                                           -005368
      IFL = 0                                                           -005369
      KNT = 0                                                           -005370
      DO 80 I=1,KSAVE                                                   -005371
```

```
      R = SAVEA(I)                                                      -005372
      T = SAVEP(I)/57.2958                                              -005373
      IF(R .GE. AMIN  .AND.  R .LE. AMAX) GO TO 81                      -005374
      IF(IFL .EQ. 0) GO TO 80                                           -005375
   75                    IF(IFR .EQ. 0) CALL QUIK3L(-1,XLFT,XRGT,       -005376
     *            YBOT,YTOP,35,TX,TY,-KNT,X,Y)                         -005377
                         IF(IFR .EQ. 1) CALL QUIK3L( 0,XLFT,XRGT,       -005378
     *            YBOT,YTOP,35,TX,TY,-KNT,X,Y)                         -005379
      IFL = 0                                                           -005380
      IFR = 1                                                           -005381
      KNT = 0                                                           -005382
      GO TO 80                                                          -005383
   81 KNT = KNT + 1                                                     -005384
      X(KNT) = R*COS(T)                                                 -005385
      Y(KNT) = R*SIN(T)                                                 -005386
      IFL = 1                                                           -005387
      IF(I .EQ. KSAVE) GO TO 75                                         -005388
   80 CONTINUE                                                          -005389
C                                                                       -005390
      RETURN                                                            -005391
      END                                                               -005392
```

```
[HDG,P     PAGEHD                                                          -005393
[FOR,IS    PAGEHD                                                          -005394
      COMPILER (XM=1),(EQUIV=CMN)                                          -005395
      SUBROUTINE PAGEHD                                                    -005396
C                                                                          -005397
C ***                                                                      -005398
C *** MSFC UNIVAC 1108 VERSION ***                                         -005399
C ***                                                                      -005400
      COMMON /LSTART/ IRUNNO, IDATE, NPAGE                                 -005401
      COMMON /LSTRT1/ UNAME(3), TITLE1(12), TITLE2(12)                     -005402
C                                                                          -005403
      EQUIVALENCE (IDATE,DATE)                                             -005404
C                                                                          -005405
      DATA NOT  /  6  /                                                    -005406
C                                                                          -005407
 2001 FORMAT(9HIRUN NO. A6, 42X 5HDATE A6, 42X 9HPAGE NO. I4,              -005408
     *       /55X 7HRUN BY 3A6, // 10X 12A6, 10X 15HCURRENT TIME = ,A6     -005409
     *       /10X 12A6, 10X 16HTHE CPU TIMER = ,I4,4H SEC)                 -005410
C                                                                          -005411
      CALL SCLOCK(DATE,TIME,ESEC,E60SEC)                                   -005412
      CALL CPUTIM(ISEC)                                                    -005413
      ISEC = ISEC/1000000                                                  -005414
      NPAGE = NPAGE + 1                                                    -005415
      WRITE(NOT,2001) IRUNNO,DATE,NPAGE,UNAME,TITLE1,TIME,TITLE2,ISEC      -005416
C                                                                          -005417
      RETURN                                                               -005418
      END                                                                  -005419
```

```
[HDG,P     PLOTSS                                                        -005420
[FOR,IS    PLOTSS                                                        -005421
       COMPILER (XM=1),(EQUIV=CMN)                                       -005422
       SUBROUTINE PLOTSS (YMAXIN,YMININ,YTOP,YBOT)                       -005423
C                                                                        -005424
C ***                                                                    -005425
C *** MSFC UNIVAC 1108 VERSION ***                                       -005426
C ***                                                                    -005427
C  SUBROUTINE SELECTS PLOT UPPER AND LOWER LIMITS FOR A 10 SQUARE        -005428
C  LINEAR PLOT GRID.                                                     -005429
C                                                                        -005430
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                -005431
C                                                                        -005432
C  YMAXIN = INPUT  MAXIMUM VALUE TO BE PLOTTED.                          -005433
C  YMININ = INPUT  MINIMUM VALUE TO BE PLOTTED.                          -005434
C  YTOP   = OUTPUT UPPER GRID LIMIT.                                     -005435
C  YBOT   = OUTPUT LOWER GRID LIMIT.                                     -005436
C                                                                        -005437
       DATA NOT/ 6  /                                                    -005438
       YMAX = YMAXIN                                                     -005439
       YMIN = YMININ                                                     -005440
                                                     NERROR = 1          -005441
       IF (YMAX .LT. YMIN) GO TO 999                                     -005442
       IF (YMAX .GT. YMIN) GO TO 21                                      -005443
   11  IF (YMAX .LT. 0.00) GO TO 13                                      -005444
       YMAX = 1.001*YMAX                                                 -005445
       YMIN = 0.999*YMIN                                                 -005446
       GO TO 15                                                          -005447
   13  YMAX = 0.999*YMAX                                                 -005448
       YMIN = 1.001*YMIN                                                 -005449
   15  IF (YMAX .NE. 0.0) GO TO 21                                       -005450
       YMAX = +.3                                                        -005451
       YMIN = -.3                                                        -005452
C                                                                        -005453
   21  VALUE = (YMAX-YMIN)/10.                                           -005454
       IF (VALUE .LT. ABS(YMIN/100000.)) GO TO 11                        -005455
       DO 23 I=1,66                                                      -005456
       DO 23 J=1,3                                                       -005457
       SCALE = 2.**(J-2) * 10. **(I-33)                                  -005458
       IF (SCALE .GE. VALUE) GO TO 31                                    -005459
   23  CONTINUE                                                          -005460
                                                     NERROR = 2          -005461
       GO TO 999                                                         -005462
C                                                                        -005463
   31  NSTEPS = YMIN/SCALE                                               -005464
       YBOT = FLOAT(NSTEPS)*SCALE                                        -005465
   32  IF (YMIN) 34,38,36                                                -005466
   33  YBOT = YBOT-SCALE                                                 -005467
   34  IF (YBOT .LE. YMIN) GO TO 38                                      -005468
       GO TO 33                                                          -005469
```

```
   35 YBOT = YBOT+SCALE                                                    -005470
   36 IF (YBOT-YMIN) 35,38,37                                              -005471
   37 YBOT = YBOT-SCALE                                                    -005472
   38 YTOP = YBOT+10.*SCALE                                                -005473
      IF (YTOP .GE. YMAX) RETURN                                           -005474
      IF (J .LT. 3) GO TO 39                                               -005475
      J = 0                                                                -005476
      I = I+1                                                              -005477
   39 J = J+1                                                              -005478
      SCALE = 2.**(J-2) * 10.**(I-33)                                      -005479
      GO TO 32                                                             -005480
C                                                                          -005481
  999 WRITE (NOT,2001) NERROR                                              -005482
 2001 FORMAT (5X,49HERROR ENCOUNTERED IN SUBROUTINE PLOTSS, NERROR = I3,   -005483
     *        /,5X, 16HPROGRAM STOPPED.  )                                 -005484
      STOP                                                                 -005485
      END                                                                  -005486
```

```
[HDG,P     PLOTWR                                                                -005487
[FOR,IS    PLOTWR                                                                -005488
      COMPILER (XM=1),(EQUIV=CMN)                                                -005489
      SUBROUTINE PLOTWR                                                           005490
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                         -005491
C                                                                                 005492
C     SUBROUTINE TO WRITE PLOT OUTPUT TAPE                                        005493
C                                                                                 005494
            COMMON /LAMBDA/                                                       005495
     *      ALAM(30)                                                            1005496
            COMMON /MAXMUM/                                                       005497
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU                     005498
            COMMON /MOMENG/                                                       005499
     *      P( 65),PMOM(30),HTOT(3),TOTL(3),ENGKE( 5),ENGPE( 5),                1105500
     *      TOTKE, TOTPE, TOTENG, AHTOT,ATOTL                                     005501
            COMMON /PLTDTA/                                                       005502
     *      NRPLOT,NCPLOT                                                         005503
            COMMON /SPECIF/                                                       005504
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),           1605505
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),                  1705506
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5),       1805507
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                                 1905508
            COMMON /TAPEND/                                                       005509
     *      NTAPE1,NTAPE2,NTAPE3                                                  005510
            COMMON /TIMESS/                                                       005511
     *      STARTT,DELTAT,T,ENDT,TMST                                             005512
            COMMON /VECTOR/                                                       005513
     *      Y(250),YDT(250)                                                     2005514
C                                                                                 005515
C                                                                                 005516
      DATA IIST / 0 /                                                             005517
      IF (IIST .EQ. 1) GO TO 5                                                    005518
      REWIND NTAPE3                                                               005519
      NRPLOT = 0                                                                  005520
      JR = 6*NB                                                                   005521
      IIST = 1                                                                    005522
      NLAMD = NLAM                                                                005523
      IF (NLAM .EQ. 0) NLAMD = 1                                                  005524
      NCPLOT = 1+2*NEQ+NLAMD+NU+JR+3+3+2*NB+5                                     005525
    5 NRPLOT = NRPLOT + 1                                                         005526
C                                                                                 005527
      WRITE (NTAPE3)  T                                                           005528
     *     ,   (Y(J),J=1,NEQ), (YDT(J),J=1,NEQ)                                 005529
     *     ,   (ALAM(J),J=1,NLAMD), (P(J),J=1,NU)                               005530
     *     ,   (PMOM(J),J=1,JR), (HTOT(J),J=1,3)                                005531
     *     ,   (TOTL(J),J=1,3), (ENGKE(J),J=1,NB)                               005532
     *     ,   (ENGPE(J),J=1,NB), AHTOT,ATOTL,TOTKE,TOTPE,TOTENG                005533
C                                                                                 005534
      RETURN                                                                      005535
      END                                                                         005536
```

```
[HDG,P    PLTCAR                                                        -005537
[FOR,IS   PLTCAR                                                        -005538
      COMPILER (XM=1),(EQUIV=CMN)                                       -005539
      SUBROUTINE PLTCAR (DATA,NI,ND,NR,NG,                              -005540
     *                   TITLI,TITLD,TITLP,TITLM,KR)                  -005541
C                                                                       -005542
C ***                                                                   -005543
C *** MSFC UNIVAC 1108 VERSION ***                                      -005544
C ***                                                                   -005545
C     DATA          DATA ARRAY                                        -005546
C     NI            LOCATION OF INDEPENDENT VARIABLE (X)              -005547
C     ND            LOCATION OF DEPENDENT VARIABLES  (Y(I),I=1,3)     -005548
C     NR            NO OF VALUES TO PLOT                              -005549
C     NG            NO OF GRIDS                                       -005550
C     TITLI         TITLE FOR INDEPENDENT VARIABLE AXIS               -005551
C     TITLD         TITLE FOR DEPENDENT VARIABLE AXIS                 -005552
C     TITLP         PLOT TITLE - UNIQUE TO FRAME                      -005553
C     TITLM         PLOT TITLE - ALL FRAMES                           -005554
C     KR            ROW DIMENSION OF DATA ARRAY IN CALLING PROGRAM    -005555
C                                                                       -005556
      REAL    MINI,MAXI,MIND,MAXD                                       -005557
C                                                                       -005558
      COMMON /LSTART/ IRUNNO,IDATE,NPAGE                                -005559
      COMMON /LSTRT1/ UNAME(3), TITLE1(12), TITLE2(12)                  -005560
C                                                                       -005561
      DIMENSION DATA(KR,1),TITLP(1),TITLM(1),ND(1)                      -005562
      DIMENSION TIT1(12),TITD(12),ISY(3)                                -005563
C                                                                       -005564
      EQUIVALENCE (IRUNNO,RUNNO)                                        -005565
C                                                                       -005566
      DATA ISY  /  1H1, 1H2, 1H3  /                                     -005567
C                                                                       -005568
C     FORM TITL                                                         -005569
      TITI(1) = TITLD                                                   -005570
      TITI(2) = 6H  VS                                                  -005571
      TITI(3) = TITLI                                                   -005572
      TITI(4) = 6H                                                      -005573
      DO 1 I=1,8                                                        -005574
    1 TITI(I+4) = TITLP(I)                                              -005575
      DO 2 I=1,10                                                       -005576
    2 TITD(I) = TITLM(I)                                                -005577
      TITD(11) = 6H                                                     -005578
      TITD(12) =  RUNNO                                                 -005579
C                                                                       -005580
C     NO OF PLOTS                                                       -005581
      NPLOTS = 0                                                        -005582
      DO 3 I=1,3                                                        -005583
      IF(ND(I) .NE. 0) NPLOTS = NPLOTS+1                                -005584
    3 CONTINUE                                                          -005585
C                                                                       -005586
```

```
C     FIND MAX/MIN OF DEPENDENT VARIABLES                                -005587
      J = ND(1)                                                          -005588
      MAXD = DATA(1,J)                                                   -005589
      MIND = DATA(1,J)                                                   -005590
      DO 10 L=1,NPLOTS                                                   -005591
      J = ND(L)                                                          -005592
      DO 10 I=1,NR                                                       -005593
      IF(DATA(I,J) .GT. MAXD) MAXD = DATA(I,J)                           -005594
      IF(DATA(I,J) .LT. MIND) MIND = DATA(I,J)                           -005595
   10 CONTINUE                                                           -005596
      IF(MAXD .EQ. MIND) MAXD = MIND + 10.0                              -005597
      CALL PLOTSS(MAXD,MIND,TOPD,BOTD)                                   -005598
C                                                                        -005599
C     GRID LOOP ****************************                             -005600
C                                                                        -005601
      NLFT = 1                                                           -005602
      NDIV = NR/NG                                                       -005603
C                                                                        -005604
      DO 45 II=1,NG                                                      -005605
      IF(II .GT.  1) NLFT=NRGT                                           -005606
      NRGT = II*NDIV                                                     -005607
      IF(II .EQ. NG) NRGT=NR                                             -005608
      NP   = NRGT - NLFT + 1                                             -005609
      NEWG = -1                                                          -005610
C                                                                        -005611
C     FIND MAX/MIN OF INDEPENDENT VARIABLE                               -005612
      MAXI = DATA(NLFT,NI)                                               -005613
      MINI = DATA(NLFT,NI)                                               -005614
      DO 11 I=NLFT,NRGT                                                  -005615
      IF(DATA(I,NI) .GT. MAXI) MAXI = DATA(I,NI)                         -005616
      IF(DATA(I,NI) .LT. MINI) MINI = DATA(I,NI)                         -005617
   11 CONTINUE                                                           -005618
      IF(MAXI .EQ. MINI) MAXI = MINI + 10.0                              -005619
C                                                                        -005620
C     PLOT DATA                                                          -005621
C                                                                        -005622
      DO 40 J=1,NPLOTS                                                   -005623
      IS  = ISY(J)                                                       -005624
      IIP = NI                                                           -005625
      IJP = ND(J)                                                        -005626
      IF (J .GT. 1) NEWG = 0                                             -005627
      CALL QUIK3L(NEWG,MINI,MAXI,BOTD,TOPD, 35,TITI,TITD, -NP,           -005628
     *     DATA(NLFT,IIP),DATA(NLFT,IJP))                                -005629
      CALL XSCLV1(DATA(NLFT,IIP),IXR,IXE)                                -005630
      CALL YSCLV1(DATA(NLFT,IJP),IYR,IYE)                                -005631
      CALL PRINTV(1,IS,IXR,IYR)                                          -005632
      CALL XSCLV1(DATA(NRGT,IIP),IXR,IXE)                                -005633
      CALL YSCLV1(DATA(NRGT,IJP),IYR,IYE)                                -005634
      CALL PRINTV(1,IS,IXR,IYR)                                          -005635
   40 CONTINUE                                                           -005636
```

```
C                                                                    -005637
   45 CONTINUE                                                       -005638
      RETURN                                                         -005639
      END                                                            -005640
```

```
[HDG,P     PR3                                                              -005641
[FOR,JS    PR3                                                              -005642
      COMPILER (XM=1),(EQUIV=CMN)                                           -005643
      SUBROUTINE PR3 (A,B,C,W,Z,S,NRA,NCA,NCB,NCC,KA,KB,KC,KW,KZ)            005644
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                    -005645
      DIMENSION A(KA,1),B(KB,1),C(KC,1),W(KW,1),Z(KZ,1)                      005646
CC                                                                          005647
C     W = A*B                                                               005648
CC                                                                          005649
      DO 10 I=1,NRA                                                         005650
      DO 10 J=1,NCB                                                         005651
      W(I,J) = 0.D 0                                                        005652
      DO 10 K=1,NCA                                                         005653
   10 W(I,J) = W(I,J) + A(I,K)*B(K,J)                                       005654
CC                                                                          005655
C                  T                                                        005656
C     Z = Z + S*C  *W                                                       005657
CC                                                                          005658
      DO 20 I=1,NCC                                                         005659
      DO 20 J=1,NCB                                                         005660
      DO 20 K=1,NRA                                                         005661
   20 Z(I,J) = Z(I,J) + S*C(K,I)*W(K,J)                                     005662
C                                                                           005663
      RETURN                                                                005664
      END                                                                   005665
```

```
[HDG,P    PRNTOU                                                        -005666
[FOR,IS   PRNTOU                                                        -005667
      COMPILER (XM=1),(EQUIV=CMN)                                       -005668
      SUBROUTINE PRNTOU                                                  005669
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -005670
      REAL   CTNEW,TNEW,CTOLD,TOLD,CPSTEP,CPSEC                          005671
C                                                                        005672
            COMMON /LAMBDA/                                              005673
     *     ALAM(30)                                                     1005674
            COMMON /MISCNO/                                              005675
     *     NOPRNT,NOPLOT                                                005676
            COMMON /MOMENG/                                              005677
     *     P( 65),PMOM(30),HTOT(3),TOTL(3),ENGKE( 5),ENGPE( 5),         1105678
     *     TOTKE, TOTPE, TOTENG, AHTOT,ATOTL                            005679
            COMMON /SPECIF/                                              005680
     *     BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),    1605681
     *     DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),           1705682
     *     NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1805683
     *     LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                          1905684
            COMMON /TIMESS/                                              005685
     *     STARTT,DELTAT,T,ENDT,TMST                                    005686
            COMMON /VECTOR/                                              005687
     *     Y(250),YDT(250)                                              2005688
C                                                                        005689
      DATA NOT,IIST / 6, 0 /                                             005690
C                                                                        005691
 1000 FORMAT (//10X,24HAT SIMULATION TIME, T = , 1PD10.4,32(2H* ))       005692
 1001 FORMAT ( 3X,21HTHE STATE VECTOR Y = )                              005693
 1002 FORMAT ( 3X,39HTHE STATE VECTOR TIME DERIVATIVE YDT = )            005694
 1003 FORMAT ( 3X,50HTHE BETAS (EULER ANGLES, POSITION COORDINATES) ARE) 005695
 1004 FORMAT ( 3X,29HTHE BETA TIME DERIVATIVES ARE)                      005696
 1005 FORMAT ( 3X,41HTHE DELTAS (CONTROL SYSTEM VARIABLES) ARE)          005697
 1006 FORMAT ( 3X,30HTHE DELTA TIME DERIVATIVES ARE)                     005698
 1007 FORMAT ( 3X, 9HFOR BODY ,I2,3X,18HTHE VELOCITIES ARE)              005699
 1017 FORMAT ( 3X, 9HFOR BODY ,I2,3X,29HTHE CORRESPONDING MOMENTA ARE)   005700
 1027 FORMAT ( 3X, 9HFOR BODY ,I2,3X,25HITS CONTRIBUTION TO TOTAL,       005701
     *    31H ANGULAR AND LINEAR MOMENTUM IS)                            005702
 1008 FORMAT ( 3X,48HITS CONTRIBUTION TO TOTAL KINETIC AND POTENTIAL ,   005703
     *   12HENERGIES IS ,3X,1P2D15.8)                                    005704
 1009 FORMAT ( 3X, 9HFOR BODY ,I2,3X,27HTHE ELASTIC DEFLECTIONS ARE)     005705
 1010 FORMAT ( 3X,50HTHE INTERCONNECTION CONSTRAINT FORCES(LAMBDAS) ARE) 005706
 1011 FORMAT ( 3X,36HTHE TOTAL ANGULAR MOMENTUM VECTOR IS)               005707
 1012 FORMAT ( 3X,35HTHE TOTAL LINEAR MOMENTUM VECTOR IS)                005708
 1013 FORMAT (/3X,29HTHE TOTAL ANGULAR MOMENTUM = ,1PD15.8,              005709
     *        / 3X,29HTHE TOTAL LINEAR MOMENTUM  = ,1PD15.8,             005710
     *        / 3X,29HTHE TOTAL KINETIC ENERGY   = ,1PD15.8,             005711
     *        / 3X,29HTHE TOTAL POTENTIAL ENERGY = ,1PD15.8,             005712
     *        / 3X,29HTHE TOTAL ENERGY (T + V)   = ,1PD15.8)             005713
 1014 FORMAT (//35X,33HCPU TIME/STEP  CPU TIME/REAL TIME, /38X,1PE10.4,  005714
     *           9X,1PE10.4)                                             005715
```

```
C                                                                       005716
      IF (I1ST .EQ. 1) GO TO 5                                          005717
      CTNEW = 0.                                                        005718
      TNEW = STARTT                                                     005719
    5 CALL PAGEHD                                                       005720
      WRITE (NOT,1000) T                                                005721
      WRITE (NOT,1001)                                                  005722
      CALL WRITES (Y,1,NEQ,1)                                           005723
C                                                                       005724
      WRITE (NOT,1000) T                                                005725
      WRITE (NOT,1002)                                                  005726
      CALL WRITES (YDT,1,NEQ,1)                                         005727
C                                                                       005728
      WRITE (NOT,1000) T                                                005729
      WRITE (NOT,1003)                                                  005730
      CALL WRITES (BETAH,6,NH,6)                                        005731
C                                                                       005732
      WRITE (NOT,1000) T                                                005733
      WRITE (NOT,1004)                                                  005734
      CALL WRITES (BETAHD,6,NH,6)                                       005735
C                                                                       005736
      IF (NDELTA .EQ. 0) GO TO 10                                       005737
      WRITE (NOT,1000) T                                                005738
      WRITE (NOT,1005)                                                  005739
      LO = LOCU(2*NB + 2)                                               005740
      CALL WRITES (Y(LO),1,NDELTA,1)                                    005741
      WRITE (NOT,1000) T                                                005742
      WRITE (NOT,1006)                                                  005743
      CALL WRITES (YDT(LO),1,NDELTA,1)                                  005744
C                                                                       005745
   10 DO 20 N=1,NB                                                      005746
      WRITE (NOT,1000) T                                                005747
      WRITE (NOT,1007) N                                                005748
      LO = LOCU(N)                                                      005749
      LE = LENU(N)                                                      005750
      CALL WRITES (Y(LO),1,LE,1)                                        005751
      WRITE (NOT,1017) N                                                005752
      CALL WRITES (P(LO),1,LE,1)                                        005753
      LOPM = 6*(N-1) + 1                                                005754
      WRITE (NOT,1027) N                                                005755
      CALL WRITES (PMOM(LOPM),1,6,1)                                    005756
      WRITE (NOT,1008) ENGKE(N),ENGPE(N)                                005757
      LE = LENU(N+NB)                                                   005758
      IF (LE .EQ. 0) GO TO 20                                           005759
      LO = LOCU(N+NB)                                                   005760
      WRITE (NOT,1009) N                                                005761
      CALL WRITES (Y(LO),1,LE,1)                                        005762
   20 CONTINUE                                                          005763
C                                                                       005764
      IF (NLAM .EQ. 0) GO TO 50                                         005765
```

```
      WRITE (NOT,1000) T                                                005766
      WRITE (NOT,1010)                                                  005767
      CALL WRITES (ALAM,1,NLAM,1)                                       005768
C                                                                       005769
   50 WRITE (NOT,1000) T                                                005770
      WRITE (NOT,1011)                                                  005771
      CALL WRITES (HTOT,1,3,1)                                          005772
      WRITE (NOT,1012)                                                  005773
      CALL WRITES (TOTL,1,3,1)                                          005774
C                                                                       005775
      WRITE (NOT,1013) AHTOT,ATOTL,TOTKE,TOTPE,TOTENG                   005776
C                                                                       005777
      IF (IIST .EQ. 1) GO TO 100                                        005778
      IIST = 1                                                          005779
      RETURN                                                            005780
  100 TOLD = TNEW                                                       005781
      CTOLD = CTNEW                                                     005782
      TNEW = T                                                          005783
      CALL CPUTIM(ISEC)                                                -005784
      CTNEW = FLOAT(ISEC/1000000)                                      -005785
      CPSTEP = CTNEW - CTOLD                                            005786
      CPSEC = CPSTEP/(TNEW-TOLD)                                        005787
      CPSTEP = CPSTEP/FLOAT(NOPRNT)                                     005788
      WRITE (NOT,1014) CPSTEP,CPSEC                                     005789
C                                                                       005790
      RETURN                                                            005791
      END                                                               005792
```

```
[HDG,P     QR2                                                                -005793
[FOR,IS    QR2                                                                -005794
      COMPILER (XM=1),(EQUIV=CMN)                                             -005795
      SUBROUTINE QR2 (A,N,R,SIG,D,KR)                                          005796
C             SUBROUTINE TO BE USED BY QRCON                                   005797
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                      -005798
      DIMENSION A(KR,1),G(3),PSI(2)                                            005799
      N1 = N - 1                                                               005800
      IA = N - 2                                                               005801
      IP = IA                                                                  005802
      IF(N-3) 450, 140, 100                                                    005803
  100 DO 130 J = 3,N1                                                          005804
      JI = N - J                                                               005805
      IF(DABS(A(JI+1,JI))-D)    140, 140, 110                                  005806
  110 DEN = A(JI+1,JI+1)*(A(JI+1,JI+1)-SIG)+A(JI+1,JI+2)*A(JI+2,JI+1)+R        005807
      IF(DEN) 120, 130, 120                                                    005808
  120 IF(DABS(A(JI+1,JI)*A(JI+2,JI+1)*(DABS(A(JI+1,JI+1)+A(JI+2,JI+2)          005809
     1-SIG)+DABS(A(JI+3,JI+2)))/DEN)-D)  140, 140, 130                         005810
  130 IP=JI                                                                    005811
  140 DO  150  J=1,IP                                                          005812
      JI=IP-J+1                                                                005813
      IF (DABS(A(JI+1,JI))-D)  160, 160, 150                                   005814
  150 IQ=JI                                                                    005815
  160 DO  440  I=IP,N1                                                         005816
      IF(I-IP)   180, 170, 180                                                 005817
  170 G(1)=A(IP,IP)*(A(IP,IP)-SIG)+A(IP,IP+1)*A(IP+1,IP)+R                     005818
      G(2)=A(IP+1,IP)*(A(IP,IP)+A(IP+1,IP+1)-SIG)                              005819
      G(3)=A(IP+1,IP)*A(IP+2,IP+1)                                             005820
      A(IP+2,IP)=0.DO                                                          005821
      GO TO 210                                                                005822
  180 G(1)=A(I,I-1)                                                            005823
      G(2)=A(I+1,I-1)                                                          005824
      IF(I-IA)   190, 190, 200                                                 005825
  190 G(3)=A(I+2,I-1)                                                          005826
      GO TO 210                                                                005827
  200 G(3)=0.DO                                                                005828
  210 XK=DSQRT(G(1)*G(1)+G(2)*G(2)+G(3)*G(3))                                  005829
      IF(G(1).LT.0.0DO) XK=-XK                                                 005830
  220 IF(XK)   230, 240, 230                                                   005831
  230 AL=G(1)/XK+1.DO                                                          005832
      PSI(1)=G(2)/(G(1)+XK)                                                    005833
      PSI(2)=G(3)/(G(1)+XK)                                                    005834
      GO TO 250                                                                005835
  240 AL=2.DO                                                                  005836
      PSI(1)=0.DO                                                              005837
      PSI(2)=0.DO                                                              005838
  250 IF(I-IQ)   260, 290, 260                                                 005839
  260 IF(I-IP)    280, 270, 280                                                005840
  270 A(I,I-1)=-A(I,I-1)                                                       005841
      GO TO 290                                                                005842
```

```
280 A(I,I-1)=-XK                                                    005843
290 DO 340  J=I,N                                                   005844
    IF(I-IA)  300, 300, 310                                         005845
300 C=PSI(2)*A(I+2,J)                                               005846
    GO TO 320                                                       005847
310 C=0.D0                                                          005848
320 E=AL*(A(I,J)+PSI(1)*A(I+1,J)+C)                                 005849
    A(I,J)=A(I,J)-E                                                 005850
    A(I+1,J)=A(I+1,J)-PSI(1)*E                                      005851
    IF(I-IA)   330, 330, 340                                        005852
330 A(I+2,J)=A(I+2,J)-PSI(2)*E                                      005853
340 CONTINUE                                                        005854
    IF(I-IA)   350, 350, 360                                        005855
350 L=I+2                                                           005856
    GO TO  370                                                      005857
360 L=N                                                             005858
370 DO  420  J=IQ,L                                                 005859
    IF(I-IA)  380, 380, 390                                         005860
380 C=PSI(2)*A(J,I+2)                                               005861
    GO TO  400                                                      005862
390 C=0.D0                                                          005863
400 E=AL*(A(J,I)+PSI(1)*A(J,I+1)+C)                                 005864
    A(J,I)=A(J,I)-E                                                 005865
    A(J,I+1)=A(J,I+1)-PSI(1)*E                                      005866
    IF(I-IA)  410, 410, 420                                         005867
410 A(J,I+2)=A(J,I+2)-PSI(2)*E                                      005868
420 CONTINUE                                                        005869
    IF(I-N+3)   430, 430, 440                                       005870
430 E=AL*PSI(2)*A(I+3,I+2)                                          005871
    A(I+3,I)=-E                                                     005872
    A(I+3,I+1)=-PSI(1)*E                                            005873
    A(I+3,I+2)=A(I+3,I+2)-PSI(2)*E                                  005874
440 CONTINUE                                                        005875
450 RETURN                                                          005876
    END                                                             005877
```

```
[HDG,P     QRCON                                                          -005878
[FOR,IS    QRCON                                                          -005879
      COMPILER (XM=1),(EQUIV=CMN)                                         -005880
      SUBROUTINE QRCON (A,M,ROOTR,ROOTI,KR,IPRNT,ITIMES)                   005881
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                  -005882
C       PROGRAM TO CALL QR TRANSFORMATION, MAXIMUM ITER IS 50.             005883
      DIMENSION A(KR,1),ROOTR(1),ROOTI(1)                                  005884
      TEST=10.D0**(-20)*10.D0**(1*ITIMES)                                  005885
      IF (ITIMES .EQ. 0) TEST=10.D0**(-20)                                 005886
      N = M                                                                005887
      IF(IPRNT)  100, 110, 100                                             005888
  100 WRITE (6,104)                                                        005889
  110 ZERO = 0.D0                                                          005890
      JJ=1                                                                 005891
  120 XNN=0.D0                                                             005892
      XN2=0.D0                                                             005893
      AA = 0.D0                                                            005894
      B = 0.D0                                                             005895
      C = 0.D0                                                             005896
      DD = 0.D0                                                            005897
      R=0.D0                                                               005898
      SIG=0.D0                                                             005899
      ITER = 0                                                             005900
  130 IF(N-2)    140, 180, 190                                             005901
  140 IF(IPRNT)  150, 160, 150                                             005902
  150 WRITE (6,105)A(1,1)                                                  005903
  160 ROOTR(1) = A(1,1)                                                    005904
      ROOTI(1) = 0.D0                                                      005905
  170 RETURN                                                               005906
  180 JJ=-1                                                                005907
  190 X = (A(N-1,N-1) - A(N,N))**2                                         005908
      S = 4.D0*A(N,N-1)*A(N-1,N)                                           005909
      ITER = ITER + 1                                                      005910
      IF (X .EQ. 0.D0) GO TO 240                                           005911
      IF (DABS(S/X) .GT. 1.0D-8) GO TO 240                                 005912
  200 IF(DABS(A(N-1,N-1))-DABS(A(N,N)))  220, 220, 210                     005913
  210 F = A(N-1,N-1)                                                       005914
      G = A(N,N)                                                           005915
      GO TO 230                                                            005916
  220 G = A(N-1,N-1)                                                       005917
      E = A(N,N)                                                           005918
  230 F = 0.D0                                                             005919
      H = 0.D0                                                             005920
      GO TO 290                                                            005921
  240 S = X + S                                                            005922
      X = A(N-1,N-1) + A(N,N)                                              005923
      IF(S)    280, 250, 250                                               005924
  250 SQ=DSQRT(S)                                                          005925
      F=0.D0                                                               005926
      H=0.D0                                                               005927
```

```
    IF (X) 260, 260, 270                                                    005928
260 E=(X-SQ)/2.D0                                                           005929
    G=(X+SQ)/2.D0                                                           005930
    GO TO 290                                                               005931
270 G=(X-SQ)/2.D0                                                           005932
    E=(X+SQ)/2.D0                                                           005933
    GO TO 290                                                               005934
280 F =DSQRT(-S)/2.D0                                                       005935
    E=X/2.D0                                                                005936
    G=E                                                                     005937
    H=-F                                                                    005938
290 IF(JJ)  310, 300, 300                                                   005939
300 D = TEST   *(DABS(G) + F)                                               005940
    IF(DABS(A(N-1,N-2)) .GT. D)  GO TO 340                                  005941
310 IF(IPRNT)  320, 330, 320                                                005942
320 WRITE (6,105)E,F, ITER                                                  005943
    WRITE (6,105)G,H                                                        005944
330 ROOTR(N) = E                                                            005945
    ROOTI(N) = F                                                            005946
    ROOTR(N-1) = G                                                          005947
    ROOTI(N-1) = H                                                          005948
    N=N-2                                                                   005949
    IF(JJ)  170, 120, 120                                                   005950
340 IF(DABS(A(N,N-1)) .GT. TEST   *DABS(A(N,N)))  GO TO 380                 005951
350 IF(IPRNT)  360, 370, 360                                                005952
360 WRITE (6,105)A(N,N), ZERO, ITER                                         005953
370 ROOTR(N) = A(N,N)                                                       005954
    ROOTI(N) = 0.D0                                                         005955
    N=N-1                                                                   005956
    GO TO 120                                                               005957
380 IF(DABS(DABS(XNN/A(N,N-1))-1.D0)-1.0D-8)  400, 400, 390                 005958
390 IF(DABS(DABS(XN2/A(N-1,N-2))-1.D0)-1.0D-8)  400, 400, 490               005959
400 VQ=DABS(A(N,N-1))-DABS(A(N-1,N-2))                                      005960
    IF (ITER-15) 520, 410, 440                                              005961
410 IF(VQ) 420, 420, 430                                                    005962
420 R = A(N-1,N-2)**2                                                       005963
    SIG = 2.D0*A(N-1,N-2)                                                   005964
    GO TO 570                                                               005965
430 R = A(N,N-1)**2                                                         005966
    SIG = 2.D0*A(N,N-1)                                                     005967
    GO TO 570                                                               005968
440 IF(VQ)   470, 470, 450                                                  005969
450 IF(IPRNT)  460, 330, 460                                                005970
460 WRITE (6,107)A(N-1,N-2)                                                 005971
    GO TO 320                                                               005972
470 IF(IPRNT)  480, 370, 480                                                005973
480 WRITE (6,107)A(N,N-1)                                                   005974
    GO TO 360                                                               005975
490 IF(ITER .GT. 50) GO TO 400                                              005976
    IF(ITER .GT. 5 ) GO TO 520                                              005977
```

```
500 R=E*E+F*F                                                          005978
    Z1= (E-AA) *(E-AA)+(F-B)*(F-B)                                     005979
    IF (R .NE. 0.D0) GO TO 501                                         005980
    Z1 = 0.D0                                                          005981
    GO TO 502                                                          005982
501 Z1 = Z1/R                                                          005983
502 R=G*G+H*H                                                          005984
    Z2= (G-C)*(G-C)+(H-DD)*(H-DD)                                      005985
    IF (R .NE. 0.D0) GO TO 503                                         005986
    Z2 = 0.D0                                                          005987
    GO TO 504                                                          005988
503 Z2=Z2/R                                                            005989
504 CONTINUE                                                           005990
    IF(Z1-0.25D0)  510, 510, 540                                       005991
510 IF(Z2-0.25D0)   520, 520, 530                                      005992
520 R=E*G-F*H                                                          005993
    SIG=E+G                                                            005994
    GO TO 570                                                          005995
530 R=E*E                                                              005996
    SIG=E+E                                                            005997
    GO TO 570                                                          005998
540 IF(Z2-0.25D0) 550, 550, 560                                        005999
550 R=G*G                                                              006000
    SIG=G+G                                                            006001
    GO TO 570                                                          006002
560 R = 0.D0                                                           006003
    SIG = 0.D0                                                         006004
570 XNN=A(N,N-1)                                                       006005
    XN2=A(N-1,N-2)                                                     006006
    CALL QR2 (A,N,R,SIG,D,KR)                                          006007
    AA=E                                                               006008
    B=F                                                                006009
    C=G                                                                006010
    DD=H                                                               006011
    GO TO 190                                                          006012
104 FORMAT(////1X, 9HREAL PART  6X  14HIMAGINARY PART, 26X             006013
   1  13HTAKEN AS ZERO 6X 4HITER //)                                   006014
105 FORMAT(1X,D15.8,3X,D15.8, 42X I3)                                  006015
107 FORMAT(56X  D13.8)                                                 006016
    END                                                                006017
```

```
[HDG,P     QRDRVR                                                    -006018
[FOR,IS    QRDRVR                                                    -006019
      COMPILER (XM=1),(EQUIV=CMN)                                    -006020
      SUBROUTINE QRDRVR (A,N,RR,RI,KR)                                006021
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                             -006022
      DIMENSION A(KR,1),RR(1),RI(1)                                   006023
C                                                                     006024
C     PUT MATRIX INTO UPPER HESENBERG FORM                            006025
      CALL SUBDIA (A,N,KR,RI)                                         006026
C     CALCULATE EIGENVALUES USING QR METHOD                           006027
      CALL QRCON  (A,N,RR,RI,KR,0,0)                                  006028
C     ALIGN EIGENVALUES INTO INCREASING ORDER                         006029
C                                                                     006030
      NM1 = N-1                                                       006031
      DO 35 J=1,NM1                                                   006032
      W2MIN = RR(J)                                                   006033
      WMIN = RI(J)                                                    006034
      IMIN = J                                                        006035
      JP1 = J+1                                                       006036
      DO 30 I=JP1,N                                                   006037
      IF (W2MIN .LE. RR(I)) GO TO 30                                  006038
      W2MIN = RR(I)                                                   006039
      WMIN = RI(I)                                                    006040
      IMIN = I                                                        006041
   30 CONTINUE                                                        006042
      RR(IMIN) = RR(J)                                                006043
      RI(IMIN) = RI(J)                                                006044
      RI(J) = WMIN                                                    006045
   35 RR(J) = W2MIN                                                   006046
      RETURN                                                          006047
      END                                                             006048
```

```
[HDG,P    READ                                                          -006049
[FOR,IS   READ                                                          -006050
      COMPILER (XM=1),(EQUIV=CMN)                                       -006051
      SUBROUTINE READ   (A,NR,NC,KR,KC)                                  006052
      DOUBLE PRECISION A,X                                              -006053
      DIMENSION A(KR,1),X(4),IREMRK(9)                                   006054
      DATA NIT,NOT/5,6/                                                  006055
      DATA IASTRS /1H*/                                                 -006056
C                                                                        006057
C  READ MATRIX OF REAL NUMBERS FROM CARDS  AND PRINT IT.                -006058
C  THE EXPLANATION OF FORMATS USED BELOW IS ...                          006059
C     A - DENOTES ANY KEY PUNCH SYMBOL. (EG, A1/*C).                     006060
C     I - DENOTES AN INTEGER NUMBER. (EG, 436).                          006061
C     F - DENOTES A REAL NUMBER. (EG, 24.963).                           006062
C  FIRST CARD   - MATRIX NAME, NUMBER OF ROWS, NUMBER OF COLUMNS         006063
C                 WITH A6,I4,I5 FORMAT.                                  006064
C               - REMARKS IN COLUMNS 16-69. A-TYPE FORMAT.               006065
C               - * IN COLUMN 72 TO PRINT MATRIX                        -006066
C  MIDDLE CARDS - DATA WITH FORMAT (2I5, 4D17).                          006067
C               - 1-ST I5 IS THE ROW NUMBER.                             006068
C               - 2-ND I5 IS THE COL NUMBER OF THE NEXT D17 FIELD.       006069
C               - NEXT 4D17 ARE ELEMENTS OF THE MATRIX.                  006070
C  LAST CARD    - TEN ZEROS IN COLUMNS 1-10.                             006071
C                                                                        006072
C      SUBROUTINE ARGUMENTS                                              006073
C  A    = OUTPUT  MATRIX READ FROM CARDS .                             -006074
C  NR   = OUTPUT  NUMBER OF ROWS IN MATRIX A.                            006075
C  NC   = OUTPUT  NUMBER OF COLS IN MATRIX A.                            006076
C  KR   = INPUT   ROW DIMENSION OF A IN CALLING PROGRAM.                 006077
C  KC   = INPUT   COL DIMENSION OF A IN CALLING PROGRAM.                 006078
C                                                                        006079
 1001 FORMAT(A6,I4,I5,9A6,2XA1)                                         -006080
 1002 FORMAT (2I5,4D17.0)                                                006081
 2001 FORMAT (//19H CARD INPUT MATRIX A6, 2X 1H( I4,2H X I4,2H )         006082
     *        2X 9A6,2X A1,//)                                          -006083
 2002 FORMAT (//19H CARD INPUT MATRIX A6, 2X 1H( I4,2H X I4,2H )         006084
     *        3X 9HCONTINUED //)                                        006085
 2003 FORMAT (// 1XA6,I4,I5,5X 9A6,2X A1)                                006086
 2004 FORMAT (1X 2I5,4D17.8)                                             006087
 2005 FORMAT (13HOEND OF READ.)                                          006088
 2006 FORMAT (25HOSIZE OF MATRIX READ IS (I4,2H X I4,2H ) )              006089
C                                                                        006090
C  READ IN HEADER CARD.                                                  006091
      READ (NIT,1001) ANAME,N1,N2,IREMRK,IZ1                            -006092
      IPRIN = 0                                                          006093
      IF(IZ1 .EQ. IASTRS) IPRIN = 1                                     -006094
      IF (IPRIN .EQ. 1) CALL PAGEHD                                      006095
C                                                                        006096
      NR = N1                                                            006097
      NC = N2                                                            006098
```

```
      IF(IPRIN.EQ.1) WRITE (NOT,2001) ANAME,NR,NC,IREMRK,IZ1                -006099
      NERROR = 1                                                             006100
      IF (NR.GT.KR .OR. NC.GT.KC) GO TO 999                                  006101
      NLINE = 0                                                              006102
      DO 105 I=1,NR                                                          006103
      DO 105 J=1,NC                                                          006104
  105 A(I,J) = 0.D 0                                                         006105
  110 READ (NIT,1002) I,JS,X                                                 006106
      IF (I.EQ.0 .AND. JS.EQ.0)  GO TO 400                                   006107
      NERROR = 2                                                             006108
      IF (I.LE.0 .OR. I.GT.NR .OR. JS.LE.0 .OR. JS.GT.NC) GO TO 998          006109
      JE = JS+3                                                              006110
      IF (JE.LE.NC) GO TO 115                                                006111
      JX = NC-JS+2                                                           006112
      NERROR = 3                                                             006113
      DO 112 J=JX,4                                                          006114
  112 IF (X(J) .NE. 0.D 0) GO TO 998                                         006115
      JE = NC                                                                006116
  115 N = 0                                                                  006117
      DO 120 J=JS,JE                                                         006118
      N = N+1                                                                006119
  120 A(I,J) = X(N)                                                          006120
      NLINE = NLINE+1                                                        006121
      IF (NLINE.LE.47)  GO TO 125                                            006122
      IF (IPRIN .EQ. 1) CALL PAGEHD                                          006123
      IF (IPRIN .EQ. 1) WRITE (NOT,2002) ANAME,NR,NC                         006124
      NLINE = 1                                                              006125
  125 IF (IPRIN .EQ. 1) WRITE (NOT,2004) I,JS,(A(I,J),J=JS,JE)               006126
      GO TO 110                                                              006127
C                                                                            006128
  400 IF (IPRIN .EQ. 1) WRITE (NOT,2005)                                     006129
      RETURN                                                                 006130
C                                                                            006131
  998 WRITE (NOT,2004) I,JS,X                                                006132
  999 WRITE (NOT,2010) NERROR                                                006133
 2010 FORMAT (1H1,42HPROGRAM STOPPED, ERROR IN SUBROUTINE READ,              006134
     *      10H NERROR = ,I3)                                               006135
      STOP                                                                   006136
      END                                                                    006137
```

```
[HDG,P     READIM                                                          -006138
[FOR,IS    READIM                                                          -006139
      COMPILER (XM=1),(EQUIV=CMN)                                          -006140
      SUBROUTINE READIM (IA,NR,NC,KR,KC)                                    006141
      DIMENSION IA(KR,1),IX(14),REMRK(9)                                    006142
      DATA NIT,NOT,IASTRS/ 5, 6, 1H* /                                      006143
C                                                                           006144
CC--TAKEN FROM FORMA.                                                       006145
CC--ACKNOWLEDGEMENT GIVEN TO RF HRUDA.                                      006146
C                                                                           006147
 1001 FORMAT (A6,I4,I5,9A6, 2XA1)                                           006148
 1002 FORMAT (16I5)                                                         006149
 2001 FORMAT (//27H CARD INPUT INTEGER MATRIX A6, 2X 1H( I4,2H X I4,2H )    006150
     *       2X  9A6,//)                                                    006151
 2002 FORMAT (//27H CARD INPUT INTEGER MATRIX A6, 2X 1H( I4,2H X I4,2H )    006152
     *       3X 9HCONTINUED //)                                             006153
 2004 FORMAT (1X 16I5)                                                      006154
 2005 FORMAT (15HOEND OF READIM.)                                           006155
 2006 FORMAT (1H1,37HERROR IN SUBROUTINE READIM, NERROR = ,I3)              006156
C                                                                           006157
C  READ IN HEADER CARD.                                                     006158
      READ (NIT,1001) ANAME,N1,N2,REMRK,IZ1                                 006159
      IPRIN = 0                                                             006160
      IF (IZ1 .EQ. IASTRS) IPRIN = 1                                        006161
      IF (IPRIN .EQ. 1) CALL PAGEHD                                         006162
C                                                                           006163
      NR = N1                                                               006164
      NC = N2                                                               006165
      IF (IPRIN .EQ. 1) WRITE (NOT,2001) ANAME,NR,NC,REMRK                  006166
      NERROR = 1                                                            006167
      IF (NR.GT.KR .OR. NC.GT.KC) GO TO 999                                 006168
      NLINE = 0                                                             006169
      DO 105 I=1,NR                                                         006170
      DO 105 J=1,NC                                                         006171
  105 IA(I,J) = 0                                                           006172
  110 READ (NIT,1002) I,JS,IX                                               006173
      IF (I.EQ.0 .AND. JS.EQ.0) GO TO 400                                   006174
      NERROR = 2                                                            006175
      IF (I.LE.0 .OR. I.GT.NR .OR. JS.LE.0 .OR. JS.GT.NC) GO TO 998         006176
      JE = JS + 13                                                          006177
      IF (JE .LE. NC) GO TO 115                                             006178
      JX = NC - JS + 2                                                      006179
      NERROR = 3                                                            006180
      DO 112 J=JX,14                                                        006181
      IF (IX(J) .NE. 0) GO TO 998                                           006182
  112 CONTINUE                                                              006183
      JE = NC                                                               006184
  115 N = 0                                                                 006185
      DO 120 J=JS,JE                                                        006186
      N = N + 1                                                             006187
```

```
  120 IA(I,J) = IX(N)                                                    006188
      NLINE = NLINE + 1                                                  006189
      IF (NLINE .LE. 47) GO TO 125                                       006190
      IF (IPRIN .EQ. 1) CALL PAGEHD                                      006191
      IF (IPRIN .EQ. 1) WRITE (NOT,2002) ANAME,NR,NC                     006192
      NLINE = 1                                                          006193
  125 IF (IPRIN .EQ. 1) WRITE (NOT,2004) I,JS,(IA(I,J),J=JS,JE)          006194
      GO TO 110                                                          006195
C                                                                        006196
  400 IF (IPRIN .EQ. 1) WRITE (NOT,2005)                                 006197
      RETURN                                                             006198
C                                                                        006199
  998 WRITE (NOT,2004) I,JS,IX                                           006200
  999 WRITE (NOT,2006) NERROR                                            006201
      STOP                                                               006202
      END                                                                006203
```

```
[HDG,P    REVADD                                                        -006204
[FOR,IS   REVADD                                                        -006205
      COMPILER (XM=1),(EQUIV=CMN)                                       -006206
      SUBROUTINE REVADD (ALPHA,A,IVEC,JVEC,Z,NRA,NCA,NRZ,NCZ,KRA,KRZ)    006207
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -006208
      DIMENSION A(KRA,1), IVEC(1), JVEC(1), Z(KRZ,1)                     006209
C                                                                        006210
C  REARRANGE AND ADD ROWS AND COLUMNS OF ALPHA * MATRIX A INTO           006211
C  MATRIX Z.                                                             006212
C  BE SURE MATRIX Z IS DEFINED BEFORE CALLING THIS SUBROUTINE. FOR       006213
C  EXAMPLE, CALL ZERO TO CLEAR MATRIX Z.                                 006214
C                                                                        006215
C     SUBROUTINE ARGUMENTS                                               006216
C  ALPHA  = INPUT  SCALAR THAT MULTIPLIES MATRIX A.                      006217
C  A      = INPUT  MATRIX TO BE ARRANGED AND ADDED. SIZE(NRA,NCA).       006218
C  IVEC   = INPUT  VECTOR. SIZE(NRA).                                    006219
C                  IVEC(I)=ROW POSITION OF A(ROW I) IN Z.                006220
C                  IF IVEC(I) IS PLUS ,Z=Z(ROW IVEC(I))+ALPHA*A(ROW I).  006221
C                  IF IVEC(I) IS MINUS,Z=Z(ROW IVEC(I))-ALPHA*A(ROW I).  006222
C                  IF IVEC(I) IS ZERO , A(ROW I) IS OMITTED IN Z.        006223
C  JVEC   = INPUT  VECTOR. SIZE(NCA).                                    006224
C                  JVEC(J)=COL POSITION OF A(COL J) IN Z.                006225
C                  IF JVEC(J) IS PLUS ,Z=Z(COL JVEC(J))+ALPHA*A(COL J).  006226
C                  IF JVEC(J) IS MINUS,Z=Z(COL JVEC(J))-ALPHA*A(COL J).  006227
C                  IF JVEC(J) IS ZERO , A(COL J) IS OMITTED IN Z.        006228
C  Z      = INPUT/OUTPUT MATRIX TO WHICH ALPHA*A IS ADDED. SIZE(NRZ,NCZ). 006229
C  NRA    = INPUT  NUMBER OF ROWS IN MATRIX A.                           006230
C  NCA    = INPUT  NUMBER OF COLS IN MATRIX A.                           006231
C  NRZ    = INPUT  NUMBER OF ROWS IN MATRIX Z.                           006232
C  NCZ    = INPUT  NUMBER OF COLS IN MATRIX Z.                           006233
C  KRA    = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.                006234
C  KRZ    = INPUT  ROW DIMENSION OF Z IN CALLING PROGRAM.                006235
C                                                                        006236
      DO 30 IA=1,NRA                                                     006237
      IZ = IABS(IVEC(IA))                                                006238
      IF (IZ .EQ. 0)   GO TO 30                                          006239
      DO 25 JA=1,NCA                                                     006240
      JZ = IABS(JVEC(JA))                                                006241
      IF (JZ .EQ. 0)   GO TO 25                                          006242
      SIGN = +1.DO                                                       006243
      IF (IVEC(IA).LT.0 .AND. JVEC(JA).GT.0  .OR.                        006244
     *    IVEC(IA).GT.0 .AND. JVEC(JA).LT.0) SIGN=-1.DO                  006245
      Z(IZ,JZ) = Z(IZ,JZ) + SIGN*ALPHA*A(IA,JA)                          006246
   25 CONTINUE                                                           006247
   30 CONTINUE                                                           006248
      RETURN                                                             006249
C                                                                        006250
      END                                                                006251
```

```
[HDG,P    REVISE                                                          -006252
[FOR,IS   REVISE                                                          -006253
      COMPILER (XM=1),(EQUIV=CMN)                                         -006254
      SUBROUTINE REVISE (A,IVEC,JVEC,Z,NRA,NCA,NRZ,NCZ,KRA,KRZ)           006255
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                  -006256
      DIMENSION A(KRA,1), IVEC(1), JVEC(1), Z(KRZ,1)                      006257
      DATA NOT / 6/                                                       006258
C                                                                         006259
C  REARRANGE ROWS AND COLUMNS OF MATRIX A TO FORM MATRIX Z.               006260
C  CALLS FORMA SUBROUTINE ZZBOMB.                                         006261
C  CODED BY RF HRUDA.  FEBRUARY 1965.                                     006262
C  LAST REVISION BY RL WOHLEN.  OCTOBER 1972.                             006263
C                                                                         006264
C     SUBROUTINE ARGUMENTS                                                006265
C  A      = INPUT  MATRIX TO BE REARRANGED. SIZE(NRA,NCA).                006266
C  IVEC   = INPUT  VECTOR. SIZE(NRA).                                     006267
C                  IVEC(I)=ROW POSITION OF A(ROW I) IN Z.                 006268
C                  IF IVEC(I) IS PLUS , Z(ROW IVEC(I)) = +A(ROW I).       006269
C                  IF IVEC(I) IS MINUS, Z(ROW IVEC(I)) = -A(ROW I).       006270
C                  IF IVEC(I) IS ZERO , A(ROW I) IS OMITTED IN Z.         006271
C  JVEC   = INPUT  VECTOR. SIZE(NCA).                                     006272
C                  JVEC(J)=COL POSITION OF A(COL J) IN Z.                 006273
C                  IF JVEC(J) IS PLUS , Z(COL JVEC(J)) = +A(COL J).       006274
C                  IF JVEC(J) IS MINUS, Z(COL JVEC(J)) = -A(COL J).       006275
C                  IF JVEC(J) IS ZERO , A(COL J) IS OMITTED IN Z.         006276
C  Z      = OUTPUT RESULT MATRIX. SIZE(NRZ,NCZ).                          006277
C  NRA    = INPUT  NUMBER OF ROWS IN MATRIX A.                            006278
C  NCA    = INPUT  NUMBER OF COLS IN MATRIX A.                            006279
C  NRZ    = INPUT  NUMBER OF ROWS IN MATRIX Z.                            006280
C  NCZ    = INPUT  NUMBER OF COLS IN MATRIX Z.                            006281
C  KRA    = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.                 006282
C  KRZ    = INPUT  ROW DIMENSION OF Z IN CALLING PROGRAM.                 006283
C                                                                         006284
      DO 10 I=1,NRZ                                                       006285
      DO 10 J=1,NCZ                                                       006286
   10 Z(I,J) = 0.D 0                                                      006287
C                                                                         006288
      DO 30 IA=1,NRA                                                      006289
      IZ = IABS(IVEC(IA))                                                 006290
      IF (IZ .EQ. 0)   GO TO 30                                           006291
C                                                                         006292
      IF (IZ .GT. NRZ) GO TO 999                                          006293
      DO 25 JA=1,NCA                                                      006294
      JZ = IABS(JVEC(JA))                                                 006295
      IF (JZ .EQ. 0)   GO TO 25                                           006296
C                                                                         006297
      IF (JZ .GT. NCZ) GO TO 999                                          006298
      SIGN = +1.D 0                                                       006299
      IF (IVEC(IA).LT.0 .AND. JVEC(JA).GT.0  .OR.                         006300
     *    IVEC(IA).GT.0 .AND. JVEC(JA).LT.0) SIGN=-1.D 0                  006301
```

```
      Z(IZ,JZ) = SIGN*A(IA,JA)                                          006302
   25 CONTINUE                                                          006303
   30 CONTINUE                                                          006304
      RETURN                                                            006305
C                                                                       006306
  999 WRITE (NOT,1001)                                                  006307
 1001 FORMAT (1H1,32HERROR IN REVISE, PROGRAM STOPPED)                  006308
      STOP                                                              006309
      END                                                               006310
```

```
[HDG,P    RKADAM                                                        -006311
[FOR,IS   RKADAM                                                        -006312
      COMPILER (XM=1),(EQUIV=CMN)                                       -006313
      SUBROUTINE RKADAM(NEQ)                                             006314
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -006315
C                                                                        006316
            COMMON /JILFLG/                                             -006317
     *      JIL                                                        -006318
            COMMON /PRWORK/                                             006319
     *      PR(250,5)                                                  1406320
            COMMON /QPRKTA/                                             006321
     *      QRK(250),PRK(4), NT                                        1506322
            COMMON /TIMESS/                                             006323
     *      STARTT,DELTAT,T,ENDT,TMST                                   006324
            COMMON /VECTOR/                                             006325
     *      Y(250),YDT(250)                                            2006326
            COMMON /VINDEP/                                             006327
     *      INDEP(250)                                                 2106328
C                                                                        006329
      DATA EPS1, EPS2  /  1.D-8,  1.D-2   /                              006330
      DATA NOT,MAXIT / 6, 10/                                            006331
C                                                                        006332
C  *****    ITYPE .EQ. 1       RUNGE-KUTTA INTEGRATION                  006333
C  *****    ITYPE .EQ. 2       ADAMS PREDICTOR/CORRECTOR INTEGRATION    006334
C                                                                        006335
C  SUBROUTINE TO INTEGRATE DIFFERENTIAL EQUATIONS (FIRST ORDER)          006336
C  IN THE TIME DOMAIN. USES RUNGE-KUTTA-GILL TO START THE ADAMS PREDICT  006337
C      CORRECTOR. MAY USE RUNGE-KUTTA ONLY, ON OPTION.                   006338
C    CODED BY CARL BODLEY  1971                                          006339
C   MODIFICATION TO CORRECTOR LOOP TO ACCOUNT FOR IMPULSIVE CHANGE TO    006340
C     STATE VECTOR (O)TWHICH OCCURS IN SUB. YDOT. MADE BY CARL BODLEY    006341
C     APR, 1974                                                          006342
C                                                                        006343
      DATA ITYPE /   1   /                                               006344
C                                                                        006345
      GO TO (10,20) , ITYPE                                              006346
C                                                                        006347
   20 IF (NT .GT. 0) GO TO 201                                           006348
      DNM = DELTAT/24.D 0                                                006349
      TR1 =  DNM*55.D 0                                                  006350
      TR2 = -DNM*59.D 0                                                  006351
      TR3 =  DNM*37.D 0                                                  006352
      TR4 = -DNM* 9.D 0                                                  006353
      TR5 =  DNM* 9.D 0                                                  006354
      TR6 =  DNM*19.D 0                                                  006355
      TR7 = -DNM* 5.D 0                                                  006356
      TR8 =  DNM* 1.D 0                                                  006357
C                                                                        006358
  201 IF (NT .GT. 3) GO TO 200                                           006359
      NL =   NT + 1                                                      006360
```

```
      DO 205 I=1,NEQ                                                    006361
      PR(I,NL) = YDT(I)                                                 006362
  205 PR(I, 5) = Y  (I)                                                 006363
      IF (NT .EQ. 3) GO TO 200                                          006364
C                                                                       006365
   10 DO 120 J = 1,4                                                    006366
      JIL = J                                                           006367
      DO 110 I=1,NEQ                                                    006368
      IF (INDEP(I) .EQ. 0) GO TO 110                                    006369
      Z = YDT(I)*DELTAT                                                 006370
      GO TO (103,101,101,105), JIL                                      006371
  101 R = PRK(JIL)*(Z - QRK(I))                                         006372
      GO TO 107                                                         006373
  103 R = PRK(JIL)*Z - QRK(I)                                           006374
      GO TO 107                                                         006375
  105 R = (Z - 2.D 0*QRK(I))/6.D 0                                      006376
  107 Y(I) = Y(I) + R                                                   006377
      QRK(I) = QRK(I) + 3.D 0*R - PRK(JIL)*Z                            006378
  110 CONTINUE                                                          006379
      IF (JIL .EQ. 1 .OR. JIL .EQ. 3) T = T + DELTAT/2.D 0              006380
  120 CALL YDOT                                                         006381
      GO TO 300                                                         006382
C                                                                       006383
  200 DO 204 I=1,NEQ                                                    006384
  204 Y(I) = PR(I,5)+TR1*PR(I,4)+TR2*PR(I,3)+TR3*PR(I,2)+TR4*PR(I,1)    006385
      T = T + DELTAT                                                    006386
      ITER = 0                                                          006387
  207 CALL YDOT                                                         006388
C                                                                       006389
      G = 0.D 0                                                         006390
      DO 203 I=1,NEQ                                                    006391
      IF (INDEP(I) .EQ. 0) GO TO 203                                    006392
      YC   = PR(I,5)+TR5*YDT(I) +TR6*PR(I,4)+TR7*PR(I,3)+TR8*PR(I,2)    006393
      DN = DABS(Y(I))                                                   006394
      DN1 = DABS(YC)                                                    006395
      IF (DN1 .GT. DN) DN = DN1                                         006396
      IF (DN .LT. EPS1) GO TO 203                                       006397
      G1 = DABS(YC - Y(I))/DN                                           006398
      IF (G1 .GT. G) G = G1                                             006399
      Y(I) = YC                                                         006400
  203 CONTINUE                                                          006401
      ITER = ITER + 1                                                   006402
      IF (G .LE. EPS2) GO TO 30                                         006403
      IF (ITER .EQ. MAXIT) GO TO 999                                    006404
      GO TO 207                                                         006405
C                                                                       006406
   30 DO 210 I=1,NEQ                                                    006407
      PR(I,1) = PR(I,2)                                                 006408
      PR(I,2) = PR(I,3)                                                 006409
      PR(I,3) = PR(I,4)                                                 006410
```

```
      PR(I,4) = YDT(I)                                                  006411
  210 PR(I,5) = Y(I)                                                    006412
C                                                                       006413
  300 NT = NT + 1                                                       006414
      ANT = NT                                                          006415
      TMST = ANT*DELTAT                                                 006416
      T = STARTT + TMST                                                 006417
C                                                                       006418
      RETURN                                                            006419
  999 WRITE (NOT,1001) MAXIT                                            006420
 1001 FORMAT (1H1,31HCORRECTOR FAILS TO CONVERGE IN I3,                 006421
     *    28HITERATIONS, PROGRAM STOPPED.)                             006422
      STOP                                                              006423
      END                                                               006424
```

```
[HDG,P     RLOCUS                                                          -006425
[FOR,IS    RLOCUS                                                          -006426
      COMPILER (XM=1),(EQUIV=CMN)                                          -006427
      SUBROUTINE RLOCUS (PPR,QQR,SCL,SR,SI,NP,NQ,THETAO,                    006428
     1                   XMIN,XMAX,YMAX,ALOC)                              006429
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                   -006430
      DOUBLE PRECISION K                                                   -006431
      REAL   XSAVE, YSAVE, SAVEO, SAVEP, SAVED, SAVEA                       006432
C                                                                           006433
      COMPLEX    S,P,Q,SO,SOR,ERR,DCMPLX                                    006434
      INTEGER OUT                                                           006435
C                                                                           006436
C     SUBROUTINE DETERMINES ROOT LOCI FOR A SINGLE ROOT.                    006437
C                                                                           006438
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                   006439
C                                                                           006440
C  PPR     = INPUT NUMERATOR POLYNOMINAL COEFFICIENTS.                      006441
C  QQR     = INPUT DENOMINATOR POLYNOMINAL COEFFICIENTS.                    006442
C            NOTE....  ALL POLY COEFFICIENTS ARE IN ASCENDING ORDER.        006443
C  SCL     = INPUT SCALE FACTOR.  NORMALLY = 1.                             006444
C  SR      = INPUT REAL PART OF STARTING ROOT.                              006445
C  SI      = INPUT IMAG PART OF STARTING ROOT.                              006446
C  NP      = INPUT SIZE OF PPR.                                             006447
C  NQ      = INPUT SIZE OF QQR.                                             006448
C  XMIN    = INPUT MINIMUM ADMISSIBLE REAL ROOT VALUE.                      006449
C  XMAX    = INPUT MAXIMUN ADMISSIBLE REAL ROOT VALUE.                      006450
C  YMAX    = INPUT MAXIMUM ADMISSIBLE IMAG ROOT VALUE.                      006451
C  ALOC    = INPUT PHASE CONTROL PARAMETER                                  006452
C                  ALOC = +1 ----+180 DEGREES PHASE,                        006453
C                  ALOC = -1 ----   0  DEGREES PHASE.                       006454
C                                                                           006455
      DIMENSION PPR(1),QQR(1)                                               006456
      DIMENSION PAR( 50), QAR( 50)                                         45206457
      DIMENSION XSAVE(500),YSAVE(500)                                      45406458
C                                                                           006459
      COMMON /PSTUFF/                                                       006460
     *            SAVEO(500), SAVEP(500), SAVED(500), SAVEA(500), KSAVE 44706461
      COMMON /LV1   /                                                       006462
     C              V1  ( 50), V2  ( 50), V3  ( 50)                       42606463
      EQUIVALENCE (V1(1),PAR(1)),(V2(1),QAR(1))                             006464
      EQUIVALENCE (SAVEO(1),XSAVE(1)), (SAVEP(1),YSAVE(1))                  006465
C                                                                           006466
      DATA OUT, KDSAVE /                                                    006467
     1      6 ,  500   /                                                   45606468
C                                                                           006469
      KSAVE = 1                                                             006470
C                                                                           006471
C                                                                           006472
C                                                                           006473
      PI=3.14159265358979D0                                                 006474
```

```
      ATR=180.D0/PI                                                    006475
C                                                                      006476
C     WRITE HEADINGS                                                   006477
C                                                                      006478
      NP1=NP-1                                                         006479
      NQ1=NQ-1                                                         006480
      CALL PAGEHD                                                      006481
      WRITE(OUT,38)                                                    006482
   38 FORMAT("0P(S) =")                                                006483
      WRITE(OUT,40) PPR(1),(PPR(I+1),I,I=1,NP1)                        006484
      WRITE(OUT,39)                                                    006485
   39 FORMAT("0Q(S) =")                                                006486
      WRITE(OUT,40) QQR(1),(QQR(I+1),I,I=1,NQ1)                        006487
   40 FORMAT("  + (",D22.15,")",6X,3(" + (",D22.15,")*S**",I2)/         006488
     *  " ",4(" + (",D22.15,")*S**",I2))                               006489
      WRITE(OUT,10)SR,SI                                               006490
   10 FORMAT("0STARTING POINT = (",F18.11,") + I(",F18.11,")")         006491
      WRITE(OUT,15) XMIN,XMAX,YMAX,YMAX                                006492
   15 FORMAT("0SCAN LIMITS ",D13.6," LT X LT ",D13.6/" ",12X,"-",      -006493
     *  D12.6," LT Y LT ",D13.6)                                       -006494
      NLINE = 0                                                        006495
      CALL PAGEHD                                                      006496
      WRITE(OUT,20)                                                    006497
   20 FORMAT(//13X,"GAIN",33X,"ROOTS",43X,"ERROR"//)                   006498
C                                                                      006499
C     FIND SCALE FACTOR (SCL) IF NOT SPECIFIED.                        006500
C     IF THE INPUTTED SCL IS POSITIVE, THAT VALUE WILL BE USED TO      006501
C     SCALE THE TWO POLYNOMIALS.  IF THE SCL IS NEGATIVE, A SCALE      006502
C     FACTOR WILL BE ESTIMATED FROM THE SIZE OF THE COEFFICIENTS OF    006503
C     THE POLYNOMIALS.                                                 006504
C                                                                      006505
      IF(SCL.GT.0.D0)GO TO 100                                         006506
      CALL FIT(NP,PPR,SLOPEP)                                          006507
      CALL FIT(NQ,QQR,SLOPEQ)                                          006508
      SCL=1.D1**(-(NP*SLOPEP+NQ*SLOPEQ)/(NP+NQ))                       006509
  100 WRITE(OUT,101) SCL                                               006510
  101 FORMAT("0SCALE FACTOR =",D13.6)                                  006511
C                                                                      006512
C     SCALE EQUATIONS BY LETTING  S(NEW)=SCL*S(OLD)                    006513
C                                                                      006514
      ABP=DABS(PPR(NP))                                                006515
      ABQ=DABS(QQR(NQ))                                                006516
      DO 120 I=1,NP1                                                   006517
      ABPP=PPR(I)/ABP                                                  006518
      NPI=NP-I                                                         006519
      DO 110 J=1,NPI                                                   006520
  110 ABPP=ABPP/SCL                                                    006521
  120 PAR(I)=ABPP                                                      006522
      DO 140 I=1,NQ1                                                   006523
      ABQQ=QQR(I)/ABQ                                                  006524
```

```
      NQI=NQ-I                                                          006525
      DO 130 J=1,NQI                                                    006526
  130 ABQQ=ABQQ/SCL                                                     006527
  140 QAR(I)=ABQQ                                                       006528
      PAR(NP)=DSIGN(1.DO,PPR(NP))                                       006529
      QAR(NQ)=DSIGN(1.DO,QQR(NQ))                                       006530
      VK=ABQ/ABP                                                        006531
C     VK=RATIO OF THE MAGNITUDE OF THE LEADING COEFFICIENTS             006532
      NQNP=NQ-NP                                                        006533
      DO 150 I=1,NQNP                                                   006534
  150 VK=VK*SCL                                                         006535
C                                                                       006536
C     INITIALIZE VALUES                                                 006537
C                                                                       006538
      NROOTS=0                                                          006539
      TR=SR/SCL                                                         006540
      TI=SI/SCL                                                         006541
      SO=DCMPLX(TR,TI)                                                  006542
      X=SR                                                              006543
      Y=SI                                                              006544
      IJP=0                                                             006545
      IVS=0                                                             006546
      RMINN=1.D-4                                                       006547
      RMIN=RMINN/SCL                                                    006548
      RMAX=1.D1/SCL                                                     006549
      RD=RMIN                                                           006550
      THETA1=THETAO                                                     006551
      THMAX=370.DO+THETA1                                               006552
      THETA2=0.DO                                                       006553
      DTM=1.D-8                                                         006554
      DTHETA=10.DO                                                      006555
C                                                                       006556
C     START SEARCH                                                      006557
C                                                                       006558
  190 THETA=THETA1/ATR                                                  006559
      SOR=SNGL(RD)*DCMPLX(DCOS(THETA),DSIN(THETA))                     -006560
      S=SO+SOR                                                          006561
C     S IS THE POINT TO BE EXAMINED                                     006562
C     NOW EVALUATE P(S) AND Q(S)                                        006563
      P=PAR(NP)                                                         006564
      Q=QAR(NQ)                                                         006565
      DO 200 I=1,NP1                                                    006566
  200 P=P*S+SNGL(PAR(NP-I))                                            -006567
      DO 210 I=1,NQ1                                                    006568
  210 Q=Q*S+SNGL(QAR(NQ-I))                                            -006569
C     FIND PHASE ANGLE OF P(S)/Q(S)                                     006570
      AP=DREAL(P)                                                       006571
      BP=DIMAG(P)                                                       006572
      CQ=DREAL(Q)                                                       006573
      DQ=DIMAG(Q)                                                       006574
```

```
      THN=BP*CQ-AP*DQ                                                    006575
      THD=AP*CQ+BP*DQ                                                    006576
      ANGL=DATAN2(THN,THD)                                               006577
C     ANGL=PHASE ANGLE IN RADIANS OF P(S)/Q(S)                           006578
      TH1=ATR*ANGL                                                       006579
C     TH1=PHASE ANGLE IN DEGREES                                         006580
C     NEXT CHECK TO SEE IF THIS NEW PHASE ANGLE HAS CROSSED THE 180 DEG  006581
      IF(IJP.EQ.1)GO TO 220                                              006582
      IF(THD.GE.0.D0) GO TO 219                                          006583
      IVS=-DSIGN(1.1D0,THN)                                              006584
      IJP=1                                                              006585
  219 IF(IVS.EQ.0)GO TO 280                                              006586
  220 IF(THD.LE.0.D0)GO TO 225                                           006587
      IVS=0                                                              006588
      IJP=0                                                              006589
      GO TO 260                                                          006590
C     NEXT THREE CARDS**IF THE 180 DEG LINE HAS BEEN CROSSED, TURN       006591
C     AROUND AND REDUCE THE SEARCH INCREMENT                             006592
  225 I1THN=DSIGN(1.1D0,THN)                                             006593
      IF(IVS-I1THN.NE.0.D0)GO TO 260                                     006594
      DTHETA=-DTHETA/1.D1                                                006595
C     THIS METHOD OF ADJUSTING DTHETA IS VERY INEFFICIENT.               006596
      IVS=-DSIGN(1.1D0,THN)                                              006597
C     IVS REMEMBERS ON WHICH SIDE OF THE 180 DEG LINE THE LAST PHASE     006598
C     ANGLE LAY                                                          006599
  260 IF(DABS(DTHETA).LT.DTM) GO TO 290                                  006600
  280 IF(THETA1.GT.THMAX)GO TO 300                                       006601
      THETA1=THETA1+DTHETA                                               006602
      GO TO 190                                                          006603
C                                                                        006604
C     END SEARCH                                                         006605
C     NEXT WRITE RESULTS OF SEARCH                                       006606
C                                                                        006607
  290 IF(DABS(180.D0-DABS(TH1)).LE.5.D0)GO TO 310                        006608
  300 RD=RD/1.5D0                                                        006609
      ANGINC=109.47D0                                                    006610
      THETA1=THETA3                                                      006611
      S=S0                                                               006612
      IF(RD.GT.RMIN)GO TO 350                                            006613
      IF(NROOTS.EQ.0)GO TO 304                                           006614
      WRITE(OUT,301) RMINN                                               006615
  301 FORMAT("0THE LAST POINT PRINTED IS WITHIN ",F7.5," OF A ROOT.")    006616
      RETURN                                                             006617
  304 WRITE(OUT,305) RMINN                                               006618
  305 FORMAT("0THE INITIAL POINT IS NOT WITHIN ",F7.5," OF THE LOCUS.")  006619
      RETURN                                                             006620
  310 IF(DABS(180.D0-DABS(TH1)).GE.1.D-3)WRITE(OUT,270)TH1               006621
  270 FORMAT(" MINIMUM VALUE OF DTHETA REACHED, PRESENT VALUE OF THETA I 006622
     *S",F15.9)                                                          006623
      K=VK*CABS(Q)/CABS(P)*ALOC                                         -006624
```

```
      NROOTS=NROOTS+1                                                   006625
      X=DREAL(S)*SCL                                                    006626
      Y=DIMAG(S)*SCL                                                    006627
      YY=DABS(Y)                                                        006628
      ERR=Q+P*SNGL(K/VK)                                               -006629
      THETA3=THETA1                                                     006630
      SC=S                                                              006631
      NLINE = NLINE + 1                                                 006632
      IF (NLINE .LT. 50) GO TO 314                                      006633
      NLINE = 1                                                         006634
      CALL PAGEHD                                                       006635
      WRITE (OUT,20)                                                    006636
  314 WRITE(OUT,50)K,X,Y,ERR                                            006637
      XSAVE(KSAVE) = X                                                  006638
      YSAVE(KSAVE) = Y                                                  006639
   50 FORMAT(5(6X,G18.9))                                               006640
      IF ((X .LT. XMIN .OR. X .GT. XMAX .OR. YY .GT. YMAX)              006641
     *     .OR. (KSAVE .GE. KDSAVE)) RETURN                            006642
      KSAVE = KSAVE + 1                                                 006643
C                                                                       006644
C     ADJUST SEAPCH RADIUS                                              006645
C                                                                       006646
C     FIND ACUTE ANGLE BETWEEN THETA1 AND THETA2 (THDIF)                006647
      THDIF1=DABS(THETA1-THETA2)                                        006648
      THDIF=DMIN1(THDIF1,360.D0-THDIF1)                                 006649
C     ADJUST SEARCH RADIUS IF THDIF IS LESS THAN 15 DEG OR              006650
C     GREATER THAN 30 DEG                                               006651
      IANG=THDIF/15.D0                                                  006652
      IF(IANG-1) 320,340,330                                            006653
  320 RD=1.5D0*RD                                                       006654
      ANGINC=138.59D0                                                   006655
      IF(RD.LE.RMAX)GO TO 350                                           006656
      RD=RD/1.5D0                                                       006657
      GO TO 340                                                         006658
  330 RD=RD/1.5D0                                                       006659
      ANGINC=109.47D0                                                   006660
      IF(RD.GE.RMIN)GO TO 350                                           006661
      RD=1.5D0*RD                                                       006662
  340 ANGINC=120.D0                                                     006663
  350 THETA2=THETA1                                                     006664
C     SET ANGLE SCANNING LIMITS                                         006665
      THETA1=THETA2-ANGINC                                              006666
      THMAX =THETA2+ANGINC                                              006667
      DTHETA=1.D1                                                       006668
      IJP=0                                                             006669
      IVS=0                                                             006670
      SO=S                                                              006671
      GO TO 190                                                         006672
      END                                                               006673
```

```
[HDG,P    RLPLOT                                                          -006674
[FOR,IS   RLPLOT                                                          -006675
      COMPILER (XM=1),(EQUIV=CMN)                                         -006676
      SUBROUTINE RLPLOT (TITLE,ISNIM,ICYC,IRLC)                           -006677
C                                                                         -006678
C ***                                                                     -006679
C *** MSFC UNIVAC 1108 VERSION ***                                        -006680
C ***                                                                     -006681
C-----SUBROUTINE PLOTS ROOT LOCI FOR A SINGLE ROOT.                       -006682
C                                                                         -006683
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                 -006684
C                                                                         -006685
C  TITLE   = INPUT ALPHA NUMERIC TITLE                                    -006686
C  ISNIM   = INPUT ROOT IDENTIFICATION PARAMETER.                         -006687
C          =1   STARTING POINT IS OPEN LOOP ZERO,                         -006688
C          =2   STARTING POINT IS OPEN LOOP POLE,                         -006689
C          =3   STARTING POINT IS CLOSED LOOP POLE.                       -006690
C  ICYC    = INPUT TRANSFER FUNCTION COUNT (PLACED ON PLOT).              -006691
C  IRLC    = INPUT ROOT LOCUS CYCLE NUMBER (PUT ON PLOT)                  -006692
C                                                                         -006693
      COMMON /LSTART/ IRUNNO, IDATE, NPAGE                                -006694
      COMMON /PSTUFF/                                                     -006695
     *            SAVEO(500), SAVEP(500), SAVED(500), SAVEA(500), KSAVE  -006696
C                                                                         -006697
      DIMENSION TITLE(1),TX(12),TY(12)                                    -006698
      DIMENSION AGR (99)                                                  -006699
C                                                                         -006700
      EQUIVALENCE (IRUNNO,RUNNO)                                          -006701
C                                                                         -006702
      DATA AGR  /                                                         -006703
     *  6H 1    ,6H 2    ,6H 3    ,6H 4    ,6H 5    ,6H 6    ,6H 7    , -006704
     *  6H 8    ,6H 9    ,6H 10   ,6H 11   ,6H 12   ,6H 13   ,6H 14   , -006705
     *  6H 15   ,6H 16   ,6H 17   ,6H 18   ,6H 19   ,6H 20   ,6H 21   , -006706
     *  6H 22   ,6H 23   ,6H 24   ,6H 25   ,6H 26   ,6H 27   ,6H 28   , -006707
     *  6H 29   ,6H 30   ,6H 31   ,6H 32   ,6H 33   ,6H 34   ,6H 35   , -006708
     *  6H 36   ,6H 37   ,6H 38   ,6H 39   ,6H 40   ,6H 41   ,6H 42   , -006709
     *  6H 43   ,6H 44   ,6H 45   ,6H 46   ,6H 47   ,6H 48   ,6H 49   , -006710
     *  6H 50   ,6H 51   ,6H 52   ,6H 53   ,6H 54   ,6H 55   ,6H 56   , -006711
     *  6H 57   ,6H 58   ,6H 59   ,6H 60   ,6H 61   ,6H 62   ,6H 63   , -006712
     *  6H 64   ,6H 65   ,6H 66   ,6H 67   ,6H 68   ,6H 69   ,6H 70   , -006713
     *  6H 71   ,6H 72   ,6H 73   ,6H 74   ,6H 75   ,6H 76   ,6H 77   , -006714
     *  6H 78   ,6H 79   ,6H 80   ,6H 81   ,6H 82   ,6H 83   ,6H 84   , -006715
     *  6H 85   ,6H 86   ,6H 87   ,6H 88   ,6H 89   ,6H 90   ,6H 91   , -006716
     *  6H 92   ,6H 93   ,6H 94   ,6H 95   ,6H 96   ,6H 97   ,6H 98   , -006717
     *  6H 99  /                                                         -006718
C                                                                         -006719
      TX( 1) = 6H                                                         -006720
      DO 5 I=1,10                                                         -006721
    5 TX(I+1) = TITLE(I)                                                  -006722
      TX(12) = 6H                                                         -006723
```

```
C                                                                       -006724
      TY( 1) = 6HROOT L                                                 -006725
      TY( 2) = 6HOCUS P                                                 -006726
      TY( 3) = 6HLOT                                                    -006727
      IF(ISNIM .EQ. 3) TY( 4) = 6HCLSD                                  -006728
      IF(ISNIM .NE. 3) TY( 4) = 6HOPEN                                  -006729
      TY( 5) = 6HLOOP                                                   -006730
      IF(ISNIM .EQ. 1) TY( 6) = 6HZERO                                  -006731
      IF(ISNIM .NE. 1) TY( 6) = 6HPOLE                                  -006732
      TY( 7) = 6H                                                       -006733
      TY( 8) = 6HCY EQ                                                  -006734
      TY( 9) = AGR(ICYC)                                                -006735
      TY(10) = 6HRL EQ                                                  -006736
      TY(11) = AGR(IRLC)                                                -006737
      TY(12) = RUNNO                                                    -006738
C                                                                       -006739
      XMIN = SAVEO(1)                                                   -006740
      XMAX = XMIN                                                       -006741
      YMIN = SAVEP(1)                                                   -006742
      YMAX = YMIN                                                       -006743
      DO 10 I=2,KSAVE                                                   -006744
      XMIN = AMIN1(XMIN,SAVEO(I))                                       -006745
      XMAX = AMAX1(XMAX,SAVEO(I))                                       -006746
      YMIN = AMIN1(YMIN,SAVEP(I))                                       -006747
   10 YMAX = AMAX1(YMAX,SAVEP(I))                                       -006748
C                                                                       -006749
      CALL PLOTSS(XMAX,XMIN,XRGT,XLFT)                                  -006750
      CALL PLOTSS(YMAX,YMIN,YTOP,YBOT)                                  -006751
C                                                                       -006752
      IF(ISNIM .EQ. 1) ISY = 4HZERO                                     -006753
      IF(ISNIM .NE. 1) ISY = 4HPOLE                                     -006754
      CALL QUIK3L(-1,XLFT,XRGT,YBOT,YTOP, 35,TX,TY,-KSAVE,              -006755
     *            SAVEO,SAVEP)                                          -006756
      CALL XSCLV1(SAVEO(1),IXR,IXE)                                     -006757
      CALL YSCLV1(SAVEP(1),IYR,IYE)                                     -006758
      CALL PRINTV(4,ISY,IXR,IYR)                                        -006759
C                                                                       -006760
      RETURN                                                            -006761
      END                                                               -006762
```

```
[HDG,P     RMVZRO                                                    -006763
[FOR,IS    RMVZRO                                                    -006764
      COMPILER (XM=1),(EQUIV=CMN)                                    -006765
      SUBROUTINE RMVZRO (RR,NR)                                       006766
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                             -006767
C                                                                     006768
C  SUBROUTINE REMOVES REAL ZEROS FROM REAL ROOT ARRAY.                006769
C                                                                     006770
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----             006771
C                                                                     006772
C  RR      = INPUT/OUTPUT ARRAY CONTAINING ALL REAL ROOTS.            006773
C  NR      = INPUT/OUTPUT NUMBER OF REAL ROOTS.                       006774
C                                                                     006775
      DIMENSION RR(1)                                                006776
C                                                                     006777
      K=0                                                            006778
      DO 10 I=1,NR                                                   006779
      IF (RR(I) .EQ. 0.D0) GO TO 10                                  006780
      K=K+1                                                          006781
      RR(K) = RR(I)                                                  006782
   10 CONTINUE                                                       006783
      NR = K                                                         006784
      RETURN                                                         006785
      END                                                            006786
```

```
[HDG,P    ROTDH                                                          -006787
[FORL,IS  ROTDH                                                          -006788
      COMPILER (XM=1),(EQUIV=CMN)                                        -006789
      SUBROUTINE ROTDH                                                    006790
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -006791
C                                                                         006792
            COMMON /BHBSRD/                                               006793
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                206794
            COMMON /HANDS /                                               006795
     *      HATH(3, 6, 8),SIGH(3, 6, 8),HATS(3, 6,10),SIGS(3, 6,10)      406796
            COMMON /PINRP /                                               006797
     *      PIN(3,3, 5), RP2(3,3, 5), RP3(3,3, 5)                       1306798
            COMMON /SPECIF/                                               006799
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),    1606800
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),           1706801
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IPGFLX( 5),IHDATA(7, 5), 1806802
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                          1906603
            COMMON /VECTOR/                                               006804
     *      Y(250),YDT(250)                                              2006805
C                                                                         006806
      DIMENSION DF(3),ANGF(3),ROTF(3,3),RW(3,3),                          006807
     *         ITOPW(2, 5),IOP(2, 5),NBODF( 5)                          8106808
C                                                                         006809
      DATA I1ST,NOT / 0, 6/                                               006810
C                                                                         006811
      IF (I1ST .EQ. 1) GO TO 500                                          006812
      I1ST = 1                                                            006813
      DO 105 I=1,2                                                        006814
      DO 105 J=1,NH                                                       006815
  105 ITOPW(I,J) = ITOPOL(I,J)                                            006816
      IS = 1                                                              006817
      DO 106 N=1,NB                                                       006818
  106 NBODF(N) = 0                                                        006819
      ITOPW(1,1) = 0                                                      006820
      NOP = 1                                                             006821
      IOP(1,1) = 1                                                        006822
      IOP(2,1) = 1                                                        006823
  125 DO 110 I=1,2                                                        006824
      DO 110 J=1,NH                                                       006825
      IF (ITOPW(I,J) .NE. IS) GO TO 110                                   006826
      NOP = NOP + 1                                                       006827
      IOP(1,J) = NOP                                                      006828
      IOP(2,J) = ITOPW(1,J)                                               006829
      IF (I .EQ. 2) GO TO 115                                             006830
      IOP(1,J) = -NOP                                                     006831
      IOP(2,J) = ITOPW(2,J)                                               006832
  115 IS1 = IOP(2,J)                                                      006833
      NBODF(IS1) = 1                                                      006834
      ITOPW(1,J) = 0                                                      006835
      ITOPW(2,J) = 0                                                      006836
```

```
  110 CONTINUE                                                          006837
      NBODF(IS) = 0                                                     006838
      IF (NOP .EQ. NH) GO TO 120                                        006839
      DO 116 N=1,NB                                                     006840
      IF (NBODF(N) .EQ. 0) GO TO 116                                    006841
      IS = N                                                            006842
      GO TO 125                                                         006843
  116 CONTINUE                                                          006844
      GO TO 999                                                         006845
  120 DO 130 J=1,NH                                                     006846
      JOPA = IOP(1,J)                                                   006847
      JOP = IABS(JOPA)                                                  006848
      ITOPW(1,JOP) = ISIGN(JOPA,J)                                      006849
  130 ITOPW(2,JOP) = IOP(2,J)                                           006850
C                                                                       006851
  500 CONTINUE                                                          006852
      NBET = LOCU(2*NB + 1) - 1                                         006853
      DO 5 J=1,NH                                                       006854
      DO 5 I=1,6                                                        006855
      I1 = I + 1                                                        006856
      IF (IHDATA(I1,J) .EQ. 1) GO TO 5                                  006857
      NBET = NBET + 1                                                   006858
      BETAH(I,J) = Y(NBET)                                              006859
    5 CONTINUE                                                          006860
C                                                                       006861
      DO 10 I=2,NH                                                      006862
      NOBQ = ITOPOL(1,I)                                                006863
      NOBP = ITOPOL(2,I)                                                006864
      LR1 = 1 + 6*(I-2)                                                 006865
      LD1 = 1 + 7*(I-2)                                                 006866
      LR2 = LR1 + 1                                                     006867
      LR3 = LR1 + 2                                                     006868
      LR4 = LR1 + 3                                                     006869
      LR5 = LR1 + 4                                                     006870
      LR6 = LR1 + 5                                                     006871
      LD2 = LD1 + 1                                                     006872
      LD3 = LD1 + 2                                                     006873
      LD4 = LD1 + 3                                                     006874
      LD5 = LD1 + 4                                                     006875
      LD6 = LD1 + 5                                                     006876
      LD7 = LD1 + 6                                                     006877
      NMQ = IRGFLX(NOBQ)                                                006878
      NMP = IRGFLX(NOBP)                                                006879
      IF (NMQ .EQ. 0) GO TO 15                                          006880
      LU = LOCU(NOBQ+NB)                                                006881
      LHS = 2*I - 3                                                     006882
      CALL MULT3  (HATH(1,1,LHS),Y(LU),DF,3,NMQ,1,3,1,1)                006883
      CALL MULT3  (SIGH(1,1,LHS),Y(LU),ANGF,3,NMQ,1,3,1,1)              006884
      CALL ROTTR  (3,1,ANGF,ROTF,DUM,DUM)                               006885
      CALL MULT3  (ROTF,RH(1,1,LR1),RH(1,1,LR3),3,3,3,3,3,3)            006886
```

```
      DO 12 J=1,3                                                        006887
   12 DH(J,LD3) = DH(J,LD1) + DF(J)                                      006888
C                                                                        006889
   15 IF (NMP .EQ. 0) GO TO 20                                           006890
      LU = LOCU(NOBP+NB)                                                 006891
      LHS = 2*I - 2                                                      006892
      CALL MULT3  (HATH(1,1,LHS),Y(LU),DF,3,NMP,1,3,1,1)                 006893
      CALL MULT3  (SIGH(1,1,LHS),Y(LU),ANGF,3,NMP,1,3,1,1)               006894
      CALL ROTTR  (3,1,ANGF,ROTF,DUM,DUM)                                006895
      CALL MULT3  (ROTF,RH(1,1,LR2),RH(1,1,LR4),3,3,3,3,3,3)             006896
      DO 17 J=1,3                                                        006897
   17 DH(J,LD4) = DH(J,LD2) + DF(J)                                      006898
C                                                                        006899
   20 DO 25 J=1,3                                                        006900
      JP3 = J + 3                                                        006901
      ANGF(J) = BETAH(J,I)                                               006902
   25 DH(J,LD5) = BETAH(JP3,I)                                           006903
      IT = IHDATA(1,I)                                                   006904
      CALL ROTTR  (3,IT,ANGF,RH(1,1,LR5),DUM,DUM)                        006905
      CALL ROTTR (1,IT,ANGF,PIN(1,1,I),RP2(1,1,I),RP3(1,1,I))            006906
      DO 35 L=1,3                                                        006907
      DO 35 J=1,3                                                        006908
   35 RW(L,J) = RH(J,L,LR3)                                              006909
      CALL MULT3  (RH(1,1,LR4),RH(1,1,LR5),ROTF,3,3,3,3,3,3)             006910
      CALL MULT3  (ROTF,RW,RH(1,1,LR6),3,3,3,3,3,3)                      006911
      CALL MULT3  (RH(1,1,LR4),DH(1,LD5),DF,3,3,1,3,3,1)                 006912
      CALL MULT3  (RH(1,1,LR6),DH(1,LD3),ANGF,3,3,1,3,3,1)               006913
      DO 40 J=1,3                                                        006914
   40 DH(J,LD6) = DH(J,LD4) + DF(J) - ANGF(J)                            006915
C                                                                        006916
   10 CONTINUE                                                           006917
C                                                                        006918
      DO 45 J=1,3                                                        006919
      JP3 = J + 3                                                        006920
      DOL(J,1) = BETAH(JP3,1)                                            006921
   45 ANGF(J) = BETAH(J,1)                                               006922
      IT = IHDATA(1,1)                                                   006923
      CALL ROTTR  (3,IT,ANGF,ROL(1,1,1),DUM,DUM)                         006924
      CALL ROTTR (1,IT,ANGF,PIN(1,1,1),RP2(1,1,1),RP3(1,1,1))            006925
C                                                                        006926
      DO 50 J=2,NH                                                       006927
      NOH  = ITOPW(1,J)                                                  006928
      LROJ = ITOPW(2,J)                                                  006929
      IF (NOH .LT. 0) GO TO 52                                           006930
      LR6 = 6*(NOH - 1)                                                  006931
      LRO = ITOPOL(2,NOH)                                                006932
      CALL MULT3  (ROL(1,1,LRO),RH(1,1,LR6),ROL(1,1,LROJ),3,3,3,3,3,3)   006933
      GO TO 50                                                           006934
   52 NOH = -NOH                                                         006935
      LR6 = 6*(NOH - 1)                                                  006936
```

```
      LRO = ITOPOL(1,NOH)                                                  006937
      DO 53 I=1,3                                                          006938
      DO 53 L=1,3                                                          006939
   53 RW(I,L) = RH(L,I,LR6)                                                006940
      CALL MULT3 (ROL(1,1,LRO),RW,ROL(1,1,LROJ),3,3,3,3,3,3)              006941
   50 CONTINUE                                                             006942
C                                                                          006943
      DO 60 J=2,NH                                                         006944
      LRO = ITOPOL(2,J)                                                    006945
      LD6 = 7*(J-1) - 1                                                    006946
      LD7 = LD6 + 1                                                        006947
   60 CALL MULT3  (ROL(1,1,LRO),DH(1,LD6),DH(1,LD7),3,3,1,3,3,3)          006948
C                                                                          006949
      DO 70 J=2,NH                                                         006950
      NOH  = ITOPW(1,J)                                                    006951
      LDOJ = ITOPW(2,J)                                                    006952
      IF (NOH .LT. 0) GO TO 72                                             006953
      LD7 = 7*(NOH - 1)                                                    006954
      LDO = ITOPOL(2,NOH)                                                  006955
      DO 74 I=1,3                                                          006956
   74 DOL(I,LDOJ) = DOL(I,LDO) + DH(I,LD7)                                 006957
      GO TO 70                                                             006958
   72 NOH = - NOH                                                          006959
      LD7 = 7*(NOH - 1)                                                    006960
      LDO = ITOPOL(1,NOH)                                                  006961
      DO 73 I=1,3                                                          006962
   73 DOL(I,LDOJ) = DOL(I,LDO) - DH(I,LD7)                                 006963
   70 CONTINUE                                                             006964
C                                                                          006965
      RETURN                                                               006966
C                                                                          006967
  999 WRITE (NOT,2001)                                                     006968
 2001 FORMAT (1H1,22HTOPOLOGY ERROR, ITOPOL)                               006969
      STOP                                                                 006970
C                                                                          006971
      END                                                                  006972
```

```
[HDG,P    ROTDS                                                          -006973
[FOR,IS   ROTDS                                                          -006974
      COMPILER (XM=1),(EQUIV=CMN)                                        -006975
      SUBROUTINE ROTDS                                                    006976
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -006977
C                                                                         006978
            COMMON /HANDS /                                               006979
     *     HATH(3, 6, 8),SIGH(3, 6, 8),HATS(3, 6,10),SIGS(3, 6,10)        406980
            COMMON /SPECIF/                                               006981
     *     BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),     1606982
     *     DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),            1706983
     *     NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1806984
     *     LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                           1906985
            COMMON /VECTOR/                                               006986
     *     Y(250),YDT(250)                                               2006987
C                                                                         006988
      DIMENSION DF(3),AF(3),RF(3,3)                                       006989
C                                                                         006990
      DO 10 L=1,NSPT                                                      006991
      NOB = IFTSMW(L)                                                     006992
      IF (IRGFLX(NOB) .EQ. 0) GO TO 10                                    006993
      LR1 = 2*L - 1                                                       006994
      LR2 = LR1 + 1                                                       006995
      LO = LOCU(NB+NOB)                                                   006996
      LE = LENU(NB+NOB)                                                   006997
      CALL MULT3  (HATS(1,1,L),Y(LO),DF,3,LE,1,3,1,1)                     006998
      CALL MULT3  (SIGS(1,1,L),Y(LO),AF,3,LE,1,3,1,1)                     006999
      CALL ROTTR (3,1,AF,RF,DUM,DUM)                                      007000
      CALL MULT3  (RF,RS(1,1,LR1),RS(1,1,LR2),3,3,3,3,3,3)               007001
      DO 15 I=1,3                                                         007002
   15 DS(I,LR2) = DS(I,LR1) + DF(I)                                       007003
C                                                                         007004
   10 CONTINUE                                                            007005
C                                                                         007006
      RETURN                                                              007007
      END                                                                 007008
```

```
[HDG,P    ROTTR                                                          -007009
[FOR,IS   ROTTR                                                          -007010
      COMPILER (XM=1),(EQUIV=CMN)                                        -007011
      SUBROUTINE ROTTR (I23,ITYPE,ANG,ROT,RP2,PP3)                        007012
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -007013
      DIMENSION ANG(1),ROT(3,3), RP2(3,3), RP3(3,3)                       007014
      DIMENSION IP(36),A(3,3,4),IV(3)                                     007015
              COMMON /NUMBRS/                                             007016
     *      ZRO,ONE,TWO,TRES                                             007017
C                                                                        007018
C    SUBROUTINE TO COMPUTE 12 TYPES OF EULER ANGLE TRANSFORMATIONS       007019
C    AND/OR PI INVERSE TRANSFORMATIONS.                                  007020
C    ALSO IF I23 .EQ. 1, COMPUTE RP2 AND RP3 WHICH MULTIPLY              007021
C          D/DT(THETA(2)) AND D/DT(THETA(3)) TO FORM D/DT(PI INVERSE).    007022
C    I23 = (1 IF ROT = PI INVERSE),  = (3 IF ROT = ROT. TRANS.)          007023
C    ITYPE = 1,2,..12 (TYPE OF EULER ANGLE PERMUTATION, SEE IP BELOW)    007024
C    ANG = INPUT EULER ANGLES                                            007025
C    ROT = EITHER PI INVERSE OR ROT. TRANS.                              007026
C                                                                        007027
C         CODED BY CARL BODLEY,  NOV. 20, 1973.                          007028
C    ADDITIONS MADE BY CARL BODLEY,  APR. 5, 1974                        007029
C                                                                        007030
      DATA IP / 1,2,3,    1,2,1,    1,3,1,    1,3,2,                     007031
     *          2,3,1,    2,3,2,    2,1,2,    2,1,3,                     007032
     *          3,1,2,    3,1,3,    3,2,3,    3,2,1   /,                 007033
     *      NOT, EPS / 6, 1.D-08/                                        007034
C                                                                        007035
      L1 = 3*(ITYPE - 1)                                                 007036
C                                                                        007037
      DO 15 L=1,3                                                        007038
      IV(L) = 1                                                          007039
      DO 10 I=1,3                                                        007040
      DO  5 J=1,3                                                        007041
    5 A(I,J,L) = ZRO                                                     007042
   10 A(I,I,L) = ONE                                                     007043
   15 CONTINUE                                                           007044
C                                                                        007045
      LR = 4 - I23                                                       007046
      DO 20 L=LR,3                                                       007047
      K = IP(L1 + L)                                                     007048
      S = DSIN(ANG(L))                                                   007049
      C = DCOS(ANG(L))                                                   007050
      GO TO (1,2,3), K                                                   007051
    1 A(2,2,L) =  C                                                      007052
      A(3,3,L) =  C                                                      007053
      A(2,3,L) = -S                                                      007054
      A(3,2,L) =  S                                                      007055
      GO TO 20                                                           007056
    2 A(1,1,L) =  C                                                      007057
      A(3,3,L) =  C                                                      007058
```

```
      A(1,3,L) =  S                                                    007059
      A(3,1,L) = -S                                                    007060
      GO TO 20                                                         007061
    3 A(1,1,L) =  C                                                    007062
      A(2,2,L) =  C                                                    007063
      A(1,2,L) = -S                                                    007064
      A(2,1,L) =  S                                                    007065
   20 CONTINUE                                                         007066
      IF (I23 .EQ. 1) GO TO 50                                         007067
C                                                                      007068
      CALL MULT3  (A(1,1,1),A(1,1,2),A(1,1,4),3,3,3,3,3,3)            007069
      CALL MULT3  (A(1,1,4),A(1,1,3),ROT,3,3,3,3,3,3)                  007070
      RETURN                                                           007071
C                                                                      007072
   50 DO 25 I=2,3                                                      007073
      I1 = IP(L1 + I)                                                  007074
   25 IV(I1) = I                                                       007075
      S2 = DSIN(ANG(2))                                                007076
      C2 = DCOS(ANG(2))                                                007077
      I1 = FLOAT(ITYPE)/4.2                                            007078
      K = ITYPE - 4*I1                                                 007079
      GO TO (61,62,63,64), K                                           007080
   61 IF (DABS(C2) .LT. EPS) GO TO 999                                 007081
      AL = C2                                                          007082
      AL2 = AL*AL                                                      007083
      BE = S2                                                          007084
      ALP = S2/AL2                                                     007085
      BEP = -ONE/AL2                                                   007086
      GO TO 65                                                         007087
   62 IF (DABS(C2) .LT. EPS) GO TO 999                                 007088
      AL = S2                                                          007089
      AL2 = AL*AL                                                      007090
      BE = C2                                                          007091
      ALP = -C2/AL2                                                    007092
      BEP = ONE/AL2                                                    007093
      GO TO 65                                                         007094
   63 IF (DABS(C2) .LT. EPS) GO TO 999                                 007095
      AL = -S2                                                         007096
      AL2 = AL*AL                                                      007097
      BE = C2                                                          007098
      ALP = C2/AL2                                                     007099
      BEP = -ONE/AL2                                                   007100
      GO TO 65                                                         007101
   64 IF (DABS(C2) .LT. EPS) GO TO 999                                 007102
      AL = C2                                                          007103
      AL2 = AL*AL                                                      007104
      BE = -S2                                                         007105
      ALP = S2/AL2                                                     007106
      BEP = ONE/AL2                                                    007107
C                                                                      007108
```

```
   65 K = IP(L1+3)                                                        007109
      GO TO (71,72,73), K                                                 007110
   71 A(2,2,1) =  A(2,3,3)                                                007111
      A(3,3,1) =  A(2,3,3)                                                007112
      A(3,2,1) =  A(2,2,3)                                                007113
      A(2,3,1) = -A(2,2,3)                                                007114
      A(1,1,1) = ZRO                                                      007115
      GO TO 75                                                            007116
   72 A(1,1,1) =  A(3,1,3)                                                007117
      A(3,3,1) =  A(3,1,3)                                                007118
      A(1,3,1) =  A(1,1,3)                                                007119
      A(3,1,1) = -A(1,1,3)                                                007120
      A(2,2,1) = ZRO                                                      007121
      GO TO 75                                                            007122
   73 A(1,1,1) =  A(1,2,3)                                                007123
      A(2,2,1) =  A(1,2,3)                                                007124
      A(2,1,1) =  A(1,1,3)                                                007125
      A(1,2,1) = -A(1,1,3)                                                007126
      A(3,3,1) = ZRO                                                      007127
   75 DO 80 I=1,3                                                         007128
      I1 = IV(I)                                                          007129
      DO 80 J=1,3                                                         007130
      A(I1,J,2) = A(I,J,1)                                                007131
   80 A(I1,J,4) = A(I,J,3)                                                007132
C                                                                         007133
      DO 90 J=1,3                                                         007134
      ROT(1,J) = A(1,J,4)/AL                                              007135
      RP3(1,J) = A(1,J,2)/AL                                              007136
      ROT(2,J) = A(2,J,4)                                                 007137
      RP3(2,J) = A(2,J,2)                                                 007138
      ROT(3,J) = A(3,J,4) - BE*ROT(1,J)                                   007139
      RP3(3,J) = A(3,J,2) - BE*RP3(1,J)                                   007140
      RP2(1,J) = ALP*A(1,J,4)                                             007141
      RP2(2,J) = ZRO                                                      007142
   90 RP2(3,J) = BEP*A(1,J,4)                                             007143
C                                                                         007144
      RETURN                                                              007145
  999 WRITE (NOT,1000) ITYPE,ANG(2)                                       007146
 1000 FORMAT (1H1,22HGIMBAL LOCK-- ITYPE = ,I5,8HANGLE = ,D15.8)          007147
      STOP                                                                007148
      END                                                                 007149
```

```
[HDG,P     RTOP                                                          -007150
[FOR,IS    PTOP                                                          -007151
      COMPILER (XM=1),(EQUIV=CMN)                                        -007152
      SUBROUTINE RTOP (RTS,POLY,TEMP,KPLY)                                007153
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -007154
CRTOP         TRANSFER FUNCTION ROOTS TO TRANSFER FUNCTION POLYNOMIALS     007155
C                                                                          007156
C           SUBROUTINE RTOP,  ROOTS CONVEPTED TO POLYNOMIAL                007157
C                                                                          007158
C --- RTS(1) = NUMBER OF REAL ROOTS IN THE NUMERATOR                       007159
C --- RTS(2) = NUMBER OF COMPLEX PAIRS IN THE NUMERATOR                    007160
C --- RTS(3) = NUMBER OF ZERO ROOTS IN THE NUMERATOR                       007161
C --- RTS(4) = NUMBER OF REAL ROOTS IN THE DENOMINATOR                     007162
C --- RTS(5) = NUMBER OF COMPLEX PAIRS IN THE DENOMINATOR                  007163
C --- RTS(6) = NUMBER OF ZERO ROOTS IN THE DENOMINATOR                     007164
C --- RTS(7) = GAIN FACTOR                                                 007165
C --- RTS(8)...RTS(I) = NUMERATOR REAL ROOTS ARRAY                         007166
C --- RTS(I+1)...RTS(J) = NUMERATOR COMPLEX ROOTS ARRAY                    007167
C --- RTS(J+1)...RTS(K) = DENOMINATOR REAL ROOTS ARRAY                     007168
C --- RTS(K+1)...RTS(L) = DENOMINATOR COMPLEX ROOTS ARRAY                  007169
C --- POLY(1) = DEGREE OF THE NUMERATOR                                    007170
C --- POLY(2) = DEGREE OF THE DONOMINATOR                                  007171
C --- POLY(3)...POLY(I) = ALL COEFFICIENTS OF NUMERATOR FOR ASCENDING PO   007172
C --- POLY(I+1)...POLY(J) = ALL COEFFICIENTS OF DENOMINATOR FOR ASCENDIN   007173
C --- OF S                                                                 007174
C --- TEMP = TEMPORARY STORAGE                                             007175
C --- NCD = NUMBER OF COMPLEX PAIRS IN DENOMINATOR                         007176
C --- NCN = NUMBER OF COMPLEX PAIRS IN NUMERATOR                           007177
C --- NDEN = TOTAL NUMBER OF DENOMINATOR ROOTS IN RTS ARRAY                007178
C --- NNUM = TOTAL NUMBER OF NUMERATOR ROOTS IN RTS ARRAY                  007179
C --- NRD = NUMBER OF REAL ROOTS IN THE DENOMINATOR                        007180
C --- NRN = NUMBER OF REAL ROOTS IN THE NUMERATOR                          007181
C --- NZD = NUMBER OF ZERO ROOTS IN THE DENOMINATOR                        007182
C --- NZN = NUMBER OF ZERO ROOTS IN THE NUMERATOR                          007183
C --- KPLY = DIMENSION SIZE OF POLY IN CALLING PROGRAM.                    007184
C                                                                          007185
      DIMENSION RTS(1),POLY(1),TEMP(1)                                     007186
      KTAPE = 6                                                            007187
      NRN=RTS(1) + 0.1D0                                                   007188
      NCN=RTS(2) + 0.1D0                                                   007189
      NZN=RTS(3) + 0.1D0                                                   007190
      NRD=RTS(4) + 0.1D0                                                   007191
      NCD=RTS(5) + 0.1D0                                                   007192
      NZD=RTS(6) + 0.1D0                                                   007193
      DO 100 I = 1,KPLY                                                    007194
      POLY(I)=0.D0                                                         007195
  100 TEMP(I) = 0.D0                                                       007196
C                                                                          007197
      NNUM = 2 * NCN + NRN + NZN                                           007198
      NDEN = 2 * NCD + NRD + NZD                                           007199
```

```
      POLY(1)=NNUM                                                      007200
      POLY(2)=NDEN                                                      007201
C                                                                       007202
      KP = 0                                                            007203
      POLY (3) = 1.DO                                                   007204
      IF (NRN) 510, 110, 130                                            007205
  110 IF (NCN) 510, 120, 190                                            007206
C           NUMERATOR IS GAIN TERM ONLY                                 007207
  120 KP = NZN + 3                                                      007208
      POLY(3) = 0.DO                                                    007209
      POLY(KP) = RTS(7)                                                 007210
      KP = KP+1                                                         007211
      GO TO 290                                                         007212
C           NUMERATOR REAL ROOTS                                        007213
  130 TEMP (2) = RTS(8)                                                 007214
      POLY (4) = TEMP (2)                                               007215
      IF (NRN-1) 180, 180, 140                                          007216
  140 DO 170 K = 2, NRN                                                 007217
      DO 150 K1 = 1, K                                                  007218
  150 TEMP (K1+1) = RTS(K+7) * POLY (K1+2) + POLY (K1+3)                007219
      DO 160 K2 = 1, K                                                  007220
  160 POLY (K2+3) = TEMP (K2+1)                                         007221
  170 CONTINUE                                                          007222
C                                                                       007223
  180 IF (NCN) 510, 250, 190                                            007224
C                                                                       007225
C           INCLUDE THE NUMERATOR COMPLEX ROOTS                         007226
  190 KNR = NRN                                                         007227
      KC = NRN +8                                                       007228
      KCM = 2 * NCN + KC - 1                                            007229
      DO 240 L = KC, KCM, 2                                             007230
      TEM1 = 2.DO*RTS(L)/RTS(L+1)                                       007231
      TEM2 = 1.DO/RTS(L+1)**2                                           007232
      LL = L-6                                                          007233
      DO 220 L2 = 1, LL                                                 007234
      TEM3 = 0.DO                                                       007235
      IF (L2-1) 210, 210, 200                                           007236
  200 TEM3 = TEM2 * POLY (L2+1)                                         007237
  210 TEMP(L2+1) = POLY (L2+3) + TEM1 * POLY (L2+2) + TEM3              007238
  220 CONTINUE                                                          007239
      KNR = KNR + 2                                                     007240
      DO 230 L3 = 1, KNR                                                007241
  230 POLY (L3+3) = TEMP (L3+1)                                         007242
  240 CONTINUE                                                          007243
C                                                                       007244
C           ENTER GAIN FACTOR, ZERO RLOTS, RESTORE COEFFICIENTS         007245
  250 KP = NZN + 4                                                      007246
      KS = 2 * NCN + NRN                                                007247
      DO 260 J = 1, KS, 1                                               007248
      POLY (KP) = TEMP (J+1) * RTS (7)                                  007249
```

```
  260 KP = KP+1                                                          007250
      POLY (NZN+3) = 1.D0* RTS (7)                                       007251
      IF (NZN) 290, 290, 270                                             007252
  270 DO 280 J = 1, NZN, 1                                               007253
  280 POLY (J+2) = 0.D0                                                  007254
C                                                                        007255
C          PROCESS DENOMINATOR ROOTS, KP IS LOCATION FOR STORING FIRST   007256
C          DENOMINATOR COEFFICIENT                                       007257
C                                                                        007258
  290 POLY (KP) = 1.D0                                                   007259
      IF (NRD) 510, 300, 340                                             007260
  300 IF (NCD) 510, 310, 400                                             007261
C          DENOMINATOR REAL ROOTS (KR IS LOCATION FOR FIRST ROOT)        007262
  310 IF (NZD) 510, 500, 320                                             007263
  320 KRIP = KP + NZD                                                    007264
      DO 330 I5 = KP,KRIP                                                007265
      POLY(I5) =0.D0                                                     007266
  330 CONTINUE                                                           007267
      POLY(KRIP) = 1.D0                                                  007268
      GO TO 500                                                          007269
  340 KR = 2*NCN + NRN + 8                                               007270
      POLY(KP+1) = RTS(KR)                                               007271
      TEMP(2) = RTS(KR)                                                  007272
      IF (NRD-1) 390, 390, 350                                           007273
  350 DO 380 K=2,NRD                                                     007274
      NC1 = KR+K-1                                                       007275
      DO 360 K1 = 1,K                                                    007276
      NC2 = KP+K1-1                                                      007277
  360 TEMP(K1+1) = RTS(NC1)*POLY(NC2) + POLY(NC2+1)                      007278
      DO 370 K2 = 1,K                                                    007279
      NC3 = KP+K2                                                        007280
  370 POLY(NC3) = TEMP(K2+1)                                             007281
  380 CONTINUE                                                           007282
C                                                                        007283
  390 IF (NCD) 510, 460, 400                                             007284
C                                                                        007285
C          PROCESS DENOMINATOR COMPLEX ROOTS                             007286
  400 KDR = NRD                                                          007287
      KC = 2*NCN + NRN + NRD +8                                          007288
      KCM = 2* NCD + KC-1                                                007289
      DO 450 L = KC, KCM, 2                                              007290
      TEM1 = 2.D0*RTS(L)/RTS(L+1)                                        007291
      TEM2 = 1.D0/RTS(L+1)**2                                            007292
      LL= L-(2*NCN+NRN+6)                                                007293
      DO 430 L2 = 1, LL                                                  007294
      NC5 = KP+L2-1                                                      007295
      TEM3 = 0.D0                                                        007296
      IF (L2-1) 420, 420, 410                                            007297
  410 TEM3 = TEM2 * POLY (NC5-1)                                         007298
  420 TEMP (L2+1) = POLY (NC5+1) + TEM1 * POLY (NC5) + TEM3              007299
```

```
  430 CONTINUE                                                          007300
      KDR = KDR + 2                                                     007301
      DO 440 L3 = 1, KDR                                                007302
      NC6 = KP+L3                                                       007303
  440 POLY (NC6) = TEMP (L3+1)                                          007304
  450 CONTINUE                                                          007305
C                                                                       007306
  460 KD = KP + NZD + 1                                                 007307
      KS = 2 * NCD + NRD                                                007308
      DO 470 M = 1, KS, 1                                               007309
      POLY (KD) = TEMP (M+1)                                            007310
  470 KD = KD+1                                                         007311
      KD = KP+NZD-1                                                     007312
      POLY (KD+1) = POLY (KP)                                           007313
      IF (NZD) 500, 500, 480                                            007314
  480 DO 490 J = KP, KD, 1                                              007315
  490 POLY (J) = 0.D0                                                   007316
C                                                                       007317
  500 RETURN                                                            007318
C           ERROR COMMENT AND RETURN TO MAIN PROGRAM                    007319
  510 WRITE (KTAPE,1002)                                                007320
 1002 FORMAT (86H1 A NEGATIVE COUNT OF ROOTS WAS ENCOUNTERED IN RTOP. PO 007321
     1LYNOMIAL COULD NOT BE OBTAINED.)                                  007322
      RETURN                                                            007323
      END                                                               007324
```

```
[HDG,P    RWRITE                                                        -007325
[FOR,IS   RWRITE                                                        -007326
      COMPILER (XM=1),(EQUIV=CMN)                                       -007327
      SUBROUTINE RWRITE (K,RR1,RI1,RR2,RI2,N1,N2,ANAM1,ANAM2)            007328
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -007329
C                                                                        007330
C-----SUBROUTINE PULLS UP NEW PAGE VIA PAGEHD, PRINTS OUT                007331
C      IDENTIFICATION(S), ANAM1 AND ANAM2, THEN PRINTS ROOTS.            007332
C                                                                        007333
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                007334
C                                                                        007335
C                    ALL ARGUMENTS ARE INPUT                             007336
C                                                                        007337
C  K       = NO. OF ROOT SETS TO PRINT.                                  007338
C  RR1     = REAL ROOTS (FIRST SET)                                      007339
C  RI1     = IMAG ROOTS (FIRST SET)                                      007340
C  RR2     = REAL ROOTS (SECOND SET)                                     007341
C  RI2     = IMAG ROOTS (SECOND SET)                                     007342
C  N1      = ROOT COUNT (FIRST SET)                                      007343
C  N2      = ROOT COUNT (SECOND SET)                                     007344
C                                                                        007345
C  ANAM1   = 4 CHARACTER ALPHANUMERIC TITLE (FIRST SET)                  007346
C  ANAM2   = 4 CHARACTER ALPHANUMERIC TITLE (SECOND SET)                 007347
C                                                                        007348
      DIMENSION RR1(1), RI1(1), RR2(1), RI2(1)                          007349
      DATA NOT /                                                        007350
     1     6  /                                                         007351
C                                                                        007352
  101 FORMAT (///20X,A4,34X,A4,//5X,2HNO,5X,9HREAL PART ,               007353
     1        3X,14HIMAGINARY PART,11X,9HREAL PART,3X,14HIMAGINARY PART, 007354
     2        //)                                                       007355
  102 FORMAT (5X,I2,3X,D12.5,5X,D12.5,9X,D12.5,5X,D12.5)                007356
  103 FORMAT (5X,I2,41X,D12.5,5X,D12.5)                                 007357
C                                                                        007358
      CALL PAGEHD                                                       007359
      IF (K .EQ. 2) GO TO 20                                            007360
      WRITE (NOT,101) ANAM1                                             007361
      DO 10 I=1,N1                                                      007362
      IF (I .GT. 50) CALL PAGEHD                                        007363
      IF (I .GT. 50) WRITE (NOT,101) ANAM1                              007364
   10 WRITE (NOT,102) I,RR1(I), RI1(I)                                  007365
      RETURN                                                            007366
   20 CONTINUE                                                          007367
C                                                                        007368
      L = MAXO(N1,N2)                                                   007369
      WRITE (NOT,101) ANAM1, ANAM2                                      007370
      DO 40 I=1,L                                                       007371
      IF (I .GT. 50) CALL PAGEHD                                        007372
      IF (I .GT. 50) WRITE (NOT,101) ANAM1, ANAM2                       007373
      IF (I .GT. N1  .OR.  I .GT. N2) GO TO 30                          007374
```

```
      WRITE (NOT,102) I, RR1(I),RI1(I),RR2(I),RI2(I)                    007375
      GO TO 40                                                          007376
   30 CONTINUE                                                          007377
      IF (I .GT. N2) WRITE (NOT,102) I, RR1(I), RI1(I)                  007378
      IF (I .GT. N1) WRITE (NOT,103) I, RR2(I), RI2(I)                  007379
   40 CONTINUE                                                          007380
C                                                                       007381
      RETURN                                                            007382
      END                                                               007383
```

```
[HDG,P     SATB                                                          -007384
[FOR,IS    SATB                                                          -007385
      COMPILER (XM=1),(EQUIV=CMN)                                        -007386
      SUBROUTINE SATB(S,A,B,Z,NRA,NCA,NCB,KRA,KRB,KRZ)                    007387
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -007388
C                                                    T                   007389
C     MATRIX PRODUCT   Z(NEW) = Z(OLD) + S*A *B                          007390
C         WHERE S = SCALAR                                               007391
C               A = INPUT   (NRA,NCA)                                    007392
C               B = INPUT   (NRA,NCB)                                    007393
C               Z = OUTPUT  (NCA,NCB)                                    007394
C                                                                        007395
      DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)                               007396
C                                                                        007397
      DO 20 I=1,NCA                                                      007398
      DO 20 J=1,NCB                                                      007399
      DO 20 K=1,NRA                                                      007400
   20 Z(I,J) = Z(I,J) + S*A(K,I)*B(K,J)                                  007401
C                                                                        007402
      RETURN                                                             007403
      END                                                                007404
```

```
[HDG,P     SFREQ2                                                        -007405
[FOR,IS    SFREQ2                                                        -007406
      COMPILER (XM=1),(EQUIV=CMN)                                        -007407
      SUBROUTINE SFREQ2 (NNR,ICN,NDR,ICD,GAIN,                            007408
     1                   FBRN,FBNC,FBR ,FBDC ,                           007409
     2                   FMIN,FMAX,TITLE)                                007410
C                                                                        007411
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                -007412
      REAL    SAVEO, SAVEP, SAVED, SAVEA                                  007413
      REAL    FMIN , FMAX , TITLE                                         007414
C                                                                        007415
C SUBROUTINE DETERMINES S-PLANE FREQUENCY RESPONSE                       007416
C USING VARIABLE INCREMENTING TECHNIQUES.                                007417
C                                                                        007418
C  FREQUENCY RESPONSE SAVED IN COMMON BLOCK /PSTUFF/                     007419
C                                                                        007420
C               -----SUBROUTINE ARGUMENT DESCRIPTIONS-----               007421
C                                                                        007422
C  NNR     = INPUT NUMERATOR REAL ROOT COUNT.                            007423
C  ICN     = INPUT NUMERATOR COMPLES PAIR ROOT COUNT.                    007424
C  NDR     = INPUT DENOMINATOR REAL ROOT COUNT.                          007425
C  ICD     = INPUT DENOMINATOR COMPLES PAIR ROOT COUNT.                  007426
C  GAIN    = INPUT BODE GAIN.                                            007427
C  FBRN    = INPUT NUMERATOR REAL ROOTS (INCLUDING ZEROS).               007428
C  FBNC    = INPUT NUMERATOR COMPLEX PAIRS.                              007429
C  FBR     = INPUT DENOMINATOR REAL ROOTS (INCLUDING ZEROS).             007430
C  FBDC    = INPUT DENOMINATOR COMPLEX PAIRS.                            007431
C  TITLE   = INPUT 80 CHARACTER ALPHANUMERIC TITLE.                      007432
C                                                                        007433
      COMPLEX    PROD,FDEN,FNUM,DCMPLX                                   007434
      DIMENSION FBRN (1) , FBNC (1) , FBR (1)  , FBDC (1) ,              007435
     1           TITLE(1) , WD(107) , TABG(20) , TABUP(31),             46207436
     2          TABZ(30) ,TABDN(31) , PROD(1)  ,                         007437
     3          FNUM( 50), FDEN( 50)                                    46407438
      COMMON /PSTUFF/                                                    007439
     *            SAVEO(500), SAVEP(500), SAVED(500), SAVEA(500), KSAVE 44707440
      COMMON /KDSIZE/                                                    007441
     1               KR, KRT, KRX, KV1, KV2, KVX                         007442
      COMMON /  LV1 /                                                    007443
     C               V1  ( 50), V2  ( 50), V3  ( 50)                     42607444
C                                                                        007445
C                                                                        007446
      DATA KWD/107/                                                      46607447
      DATA KDSAVE/500/                                                   46807448
      DATA NOT/ 6 /                                                      007449
C  TABG = GROSS CONSTANT TABLE                                           007450
C                                                                        007451
      DATA KTABG,KTABUP,KTABDN/18,31,29/                                 007452
C                                                                        007453
      DATA TABUP /                                                       007454
```

```
     1      0.60000D0, 0.70000D0, 0.75000D0, 0.80000D0, 0.84000D0,          007455
     2      0.88000D0, 0.90000D0, 0.92000D0, 0.94000D0, 0.96000D0,          007456
     3      0.96500D0, 0.97000D0, 0.97500D0, 0.98000D0, 0.98400D0,          007457
     4      0.98800D0, 0.99000D0, 0.99200D0, 0.99400D0, 0.99600D0,          007458
     5      0.99800D0, 0.99880D0, 0.99950D0, 0.99975D0, 0.99990D0,          007459
     6      0.99992D0, 0.99994D0, 0.99996D0, 0.99998D0, 0.99999D0,          007460
     7      1.0000D0/                                                       007461
C                                                                          007462
      DATA TABG / 1.00D0, 1.10D0, 1.25D0, 1.40D0, 1.60D0, 1.80D0,          007463
     1            2.00D0, 2.20D0, 2.50D0, 2.80D0, 3.20D0, 3.80D0,          007464
     2            4.50D0, 5.20D0, 6.20D0, 7.00D0, 7.80D0, 8.90D0,          007465
     3            0.00D0, 0.00D0/                                          007466
C                                                                          007467
C     TABUP   --- LEADING UP TO THE DAMPED NATURAL FREQUENCIES.            007468
C                                                                          007469
      VMAX  = 0.D0                                                         007470
      PDPHI = 0.D0                                                         007471
      WI    = 0.D0                                                         007472
C                                                                          007473
      KWCT = 0                                                             007474
      KOPH = 0                                                             007475
      KPRINT = 1                                                           007476
      KSAVE = 0                                                            007477
C                                                                          007478
      NRN = NNR                                                            007479
      NCN = ICN                                                            007480
      NRD = NDR                                                            007481
      NCD = ICD                                                            007482
      FK = GAIN                                                            007483
C                                                                          007484
C                                                                          007485
      DO 110 I=1,KTABDN                                                    007486
      J = KTABUP-I                                                         007487
  110 TABDN(I) = 1.D0/TABUP(J)                                             007488
C                                                                          007489
C              TABDN --- LEADING AWAY FROM DAMPED NATURAL FREQUENCIES      007490
C              LNCTR --- LINE COUNTER                                      007491
C                                                                          007492
  120 LNCTR = 40                                                           007493
      JX  = 13                                                             007494
C                                                                          007495
C             NULL WD                                                      007496
C                                                                          007497
      DO 125 I= 1,KWD                                                      007498
  125 WD(I) = 0.D0                                                         007499
  130 CONTINUE                                                             007500
      DO 140 I=1,KR                                                        007501
      FNUM(I) = (0.0,0.0)                                                  007502
      FDEN(I) = (0.0,0.0)                                                  007503
  140 CONTINUE                                                             007504
```

```
C                                                                       007505
C               FMIN = LOWER LIMIT.                                     007506
C               SAVE IT AND DESTROY SAVLO IF NEEDED                     007507
      SAVLO = DBLE(FMIN)                                                007508
C                                                                       007509
C               COMPUTE  NUMERATOR DAMPED NATURAL FREQUENCIES.          007510
C                                                                       007511
      I = 0                                                             007512
      IF (NCN) 1220, 180, 150                                           007513
  150 NTOTN = NCN*2                                                     007514
      DO 170 J=1,NTOTN,2                                                007515
      ABLE = FBNC(J) * FBNC(J+1)                                        007516
      BAKER = FBNC(J+1) * DSQRT(1.DO - FBNC(J)**2)                      007517
      TEMP  = (BAKER)**2 - (ABLE)**2                                    007518
      IF (TEMP) 170, 170, 160                                           007519
  160 I     = I+1                                                       007520
      WD(I) = DSQRT(TEMP)                                               007521
  170 CONTINUE                                                          007522
C                                                                       007523
C               COMPUTE DENOMINATOR DAMPED NATURAL FREQUENCIES.         007524
C                                                                       007525
  180 IF (NCD) 1240, 220, 190                                           007526
  190 NTOTD = NCD*2                                                     007527
C                                                                       007528
      DO 210 J=1,NTOTD,2                                                007529
      ABLE  = FBDC(J) * FBDC(J+1)                                       007530
      BAKER = FBDC(J+1) * DSQRT(1.DO - FBDC(J)**2)                      007531
      TEMP  = (BAKER)**2 - (ABLE)**2                                    007532
      IF (TEMP) 210, 210, 200                                           007533
  200 I     = I+1                                                       007534
      WD(I) = DSQRT(TEMP)                                               007535
  210 CONTINUE                                                          007536
  220 KCOUNT = I                                                        007537
C                                                                       007538
C               THERE ARE KCOUNT FREQUENCIES,                           007539
C               SORT THEM IN INCREASING MAGNITUDE.                      007540
C                                                                       007541
      IF (KCOUNT - 1) 240, 350, 250                                     007542
  240 J=1                                                               007543
      GO TO 370                                                         007544
  250 DO 270 J=1,KCOUNT                                                 007545
      DO 270 I=J,KCOUNT                                                 007546
      IF (WD(J) .LE. WD(I)) GO TO 270                                   007547
      TEMP  = WD(J)                                                     007548
      WD(J) = WD(I)                                                     007549
      WD(I) = TEMP                                                      007550
  270 CONTINUE                                                          007551
C                                                                       007552
C               SORT COMPLETE,                                          007553
C               CHECK FOR EQUAL FREQUENCIES,                            007554
```

```
C               IF SO, ELIMINATE ONE.                                   007555
C                                                                       007556
  280 I = 1                                                             007557
      J = 2                                                             007558
C                                                                       007559
  290 IF (WD(I) - WD(J)) 300, 320, 340                                  007560
  300 I = I+1                                                           007561
      J = J+1                                                           007562
  310 IF (KCOUNT - J) 350, 350, 290                                     007563
  320 DO 330 K=J,KCOUNT                                                 007564
      WD(K-1) = WD(K)                                                   007565
  330 CONTINUE                                                          007566
      WD(KCOUNT) = 0.D0                                                 007567
      KCOUNT = KCOUNT - 1                                               007568
      GO TO 310                                                         007569
  340 CALL PAGEHD                                                       007570
      WRITE (NOT,1313)                                                  007571
 1313 FORMAT (//10X,23HTHE SORT ROUTINE FAILED /                        007572
     1          10X,16HPROGRAM STOPPED. )                              007573
      STOP                                                              007574
  350 CONTINUE                                                          007575
C                                                                       007576
C                                                                       007577
C                                                                       007578
  360 I = 1                                                             007579
      J = 1                                                             007580
      IF (WD(I) .GT. 0.D0) GO TO 430                                    007581
  370 W = TABG(J) * SAVLO                                               007582
      IF (W .GT. DBLE(FMAX)) GO TO 400                                  007583
      IF (KTABG .GT. J) GO TO 410                                       007584
      SAVLO = SAVLO * 10.D0                                             007585
      J  = 1                                                            007586
      KK = 1                                                            007587
      GO TO 840                                                         007588
C                                                                       007589
C               THE SHOW IS OVER,                                       007590
C               GETOUT.                                                 007591
C                                                                       007592
  400 KK = 6                                                            007593
      GO TO 1490                                                        007594
  410 J  = J+1                                                          007595
      KK = 1                                                            007596
      GO TO 840                                                         007597
C                                                                       007598
C               ENTRY POINT FOR LOOPING ON FREQUENCY INCREMENTING.      007599
C                                                                       007600
  420 CONTINUE                                                          007601
      GO TO (370, 450, 500, 610, 660, 1490),KK                          007602
  430 IF (WD(I) .GT. SAVLO) GO TO 450                                   007603
      I=I+1                                                             007604
```

```
      GO TO 430                                                          007605
  450 IF (TABG(J)*SAVLO - TABUP(1)*WD(I)) 460, 490, 490                  007606
  460 W = TABG(J)*SAVLO                                                  007607
      IF (KTABG .GT. J) GO TO 480                                        007608
      SAVLO = SAVLO * 10.D0                                              007609
      J  = 1                                                             007610
      KK = 2                                                             007611
      GO TO 830                                                          007612
  480 J  = J+1                                                           007613
      KK = 2                                                             007614
      GO TO 830                                                          007615
  490 J  = 1                                                             007616
  500 IF (J - KTABUP) 520, 510, 530                                      007617
  510 W = TABUP(J) * WD(I)                                               007618
      KPRINT = 2                                                         007619
      J  = J+1                                                           007620
      KK = 3                                                             007621
      GO TO 830                                                          007622
  520 W  = TABUP(J) * WD(I)                                              007623
      J  = J+1                                                           007624
      KK = 3                                                             007625
      GO TO 830                                                          007626
  530 IF (WD(I+1) .GT. 0.D0) GO TO 550                                   007627
C                                                                        007628
C               THE LAST FREQUENCY IS I,                                 007629
C               MAKE I+1 A DUMMY.                                        007630
C                                                                        007631
      WD(I+1) = FMAX * TABDN(KTABDN)                                     007632
  550 IF (TABUP(JX)*WD(I+1) - WD(I)) 560, 640, 650                       007633
  560 J = JX                                                             007634
  570 IF (TABUP(J)*WD(I+1)  - WD(I)) 580, 590, 600                       007635
  580 J = J+1                                                            007636
      GO TO 570                                                          007637
  590 J = J+1                                                            007638
  600 I = I+1                                                            007639
  610 IF ((J-KTABUP) .EQ. 0) KPRINT = 2                                  007640
      IF (J .GT. KTABUP) GO TO 630                                       007641
  620 W = TABUP(J)*WD(I)                                                 007642
      J  = J+1                                                           007643
      KK = 4                                                             007644
      GO TO 830                                                          007645
  630 J = 1                                                              007646
      GO TO 530                                                          007647
  640 J = JX+1                                                           007648
      GO TO 600                                                          007649
  650 J = 1                                                              007650
  660 IF (J .GT. KTABDN) GO TO 690                                       007651
      IF (TABDN(J)*WD(I) - TABUP(JX)*WD(I+1)) 680, 740, 750              007652
  680 W = TABDN(J) * WD(I)                                               007653
      J  = J+1                                                           007654
```

```
      KK = 5                                                                007655
      GO TO 830                                                             007656
  690 IF (TABUP(1)*WD(I+1) - TABDN(KTABDN)*WD(I)) 700, 700, 760             007657
  700 J = 1                                                                 007658
  710 IF (TABUP(J)*WD(I+1) - TABDN(KTABDN)*WD(I)) 720, 730, 730             007659
  720 J = J+1                                                               007660
      GO TO 710                                                             007661
  730 I = I+1                                                               007662
      GO TO 610                                                             007663
  740 J = JX+1                                                              007664
      GO TO 600                                                             007665
  750 J = JX                                                                007666
      GO TO 600                                                             007667
  760 IF (TABDN(KTABDN)*WD(I) - TABG(KTABG)*SAVLO) 770, 810, 820            007668
  770 J = 1                                                                 007669
  780 IF (TABDN(KTABDN)*WD(I) - TABG(J)*SAVLO) 790, 790, 800                007670
  790 I = I+1                                                               007671
      GO TO 450                                                             007672
  800 J = J+1                                                               007673
      GO TO 780                                                             007674
  810 SAVLO = SAVLO * 10.D0                                                 007675
      J  = 1                                                                007676
      I  = I+1                                                              007677
      GO TO 450                                                             007678
  820 SAVLO = SAVLO * 10.D0                                                 007679
      GO TO 760                                                             007680
  830 IF (W .GT. FMAX) GO TO 1490                                           007681
  840 J1 = 1                                                                007682
  850 IF (NRN) 1210, 910, 860                                               007683
C                                                                           007684
C               EVALUATE NUMERATOR REAL ROOTS                               007685
C                                                                           007686
  860 DO 900 I1=1,NRN                                                       007687
      IF (FBRN(I1) .EQ. 0.D0) GO TO 880                                     007688
  870 FNUM(J1) = DCMPLX(1.D0,FBRN(I1)*W)                                    007689
      GO TO 890                                                             007690
  880 FNUM(J1) = DCMPLX(0.D0,W)                                             007691
  890 J1 = J1+1                                                             007692
  900 CONTINUE                                                              007693
  910 IF (NCN) 1220, 940, 920                                               007694
C                                                                           007695
C               EVALUATE NUMERATOR COMPLEX PAIRS                            007696
  920 DO 930 I1 = 1,NTOTN,2                                                 007697
C                                                                           007698
C                    REAL PART                                              007699
C                    IMAGINARY PART                                         007700
C                                                                           007701
      FNUM(J1) = DCMPLX(1.D0 - W**2 / (FBNC(I1+1))**2 ,                     007702
     1                  (2.D0* FBNC(I1)*W)/ FBNC(I1+1))                     007703
      J1 = J1+1                                                             007704
```

```
  930 CONTINUE                                                          007705
C                                                                       007706
C               REPEAT THE ABOVE PROCEDURE FOR DENOMINATOR              007707
C                                                                       007708
  940 J1 = 1                                                            007709
  950 IF (NRD) 1230, 1010, 960                                          007710
  960 DO 1000 I1=1,NRD                                                  007711
      IF (FBR(I1).EQ. 0.D0) GO TO 980                                   007712
      FDEN(J1) = DCMPLX(1.D0 , FBR (I1)*W)                              007713
      GO TO 990                                                         007714
  980 FDEN(J1) = DCMPLX(0.D0 ,W)                                        007715
  990 J1 = J1+1                                                         007716
 1000 CONTINUE                                                          007717
 1010 IF (NCD) 1240, 1040, 1020                                         007718
 1020 DO 1030 I1 = 1,NTOTD,2                                            007719
      ALPHA = 1.D0 - W**2 / (FBDC(I1+1))**2                             007720
      BETA  = 2.D0 * FBDC (I1) * W / FBDC (I1+1)                        007721
      IF (ALPHA .LT. 1.D-20 .AND. BETA .EQ. 0.D0) BETA = 1.0D-10        007722
      FDEN(J1) = DCMPLX(ALPHA,BETA)                                     007723
      J1 = J1+1                                                         007724
 1030 CONTINUE                                                          007725
C                                                                       007726
C               EVALUATE  F(S) WITH COMPLEX ARITHMETIC ROUTINE.         007727
C                                                                       007728
 1040 KN       = NRN+NCN                                                007729
      KD       = NRD+NCD                                                007730
      PROD(1) = DCMPLX(1.D0,0.D0)                                       007731
      IF (KN .LE. 0) GO TO 1090                                         007732
      IF (KD .LE. 0) GO TO 1130                                         007733
      IF (KN .GE.KD) GO TO 1110                                         007734
C                                                                       007735
C               FACTORS IN DENOMINATOR EXCEED THOSE IN NUMERATOR.       007736
C                                                                       007737
      DO 1080 I1=1,KN                                                   007738
      PROD(1) = FNUM(I1) * PROD(1)/FDEN(I1)                             007739
 1080 CONTINUE                                                          007740
 1090 K = KN+1                                                          007741
      DO 1100 I1=K,KD                                                   007742
      PROD(1) = PROD(1)/FDEN(I1)                                        007743
 1100 CONTINUE                                                          007744
      GO TO 1150                                                        007745
C                                                                       007746
C               FACTORS IN NUMERATOR EXCEED THOSE IN DENOMINATOR.       007747
C                                                                       007748
 1110 DO 1120 I1=1,KD                                                   007749
      PROD(1) = FNUM(I1)*PROD(1)/FDEN(I1)                               007750
 1120 CONTINUE                                                          007751
      IF (KN .LE. KD) GO TO 1150                                        007752
 1130 K = KD+1                                                          007753
      DO 1140 I1=K,KN                                                   007754
```

```
      PROD(1) = PROD(1)*FNUM(I1)                                        007755
 1140 CONTINUE                                                          007756
 1150 PROD(1) = PROD(1) * DCMPLX(FK,0.D0)                               007757
C                                                                       007758
C               EVALUATION OF F(S) IS NOW COMPLETE,                     007759
C               NORMAL COMPUTED FORM -- F(JW) = ALPHA + BETA.           007760
C               ALPHA = REAL PART,                                      007761
C               BETA  = IMAGINARY PART.                                 007762
C                                                                       007763
C          -----CARTESIAN FORM (X,Y)-----                               007764
      ALPHA = DREAL (PROD(1))                                           007765
      BETA  = DIMAG (PROD(1))                                           007766
C                                                                       007767
C               IN POLAR FORM -- F(JW) = (AR,PHI).                      007768
C               AR  IS AMPLITUDE                                        007769
C               PHI IS PHASE ANGLE                                      007770
C                                                                       007771
C          -----POLAR FORM-----                                         007772
      RED =  (ALPHA**2 + BETA**2)                                       007773
      AR  =  DSQRT(RED)                                                 007774
      PHI = 0.0D00                                                       07775
      IF(AR .GT. 0.0D00) PHI = DATAN2(BETA,ALPHA)                        07776
C               CONVERT PHI FROM RADIANS TO DEGREES.                    007777
      DPHI = PHI * 57.2958D0                                            007778
      IF (DPHI .LT. 0.D0) DPHI = DPHI + 360.D0                          007779
C                                                                       007780
C          -----PRINT FREQUENCY RESPONSE DATA-----                      007781
C                                                                       007782
C                                                                       007783
C               SET OUTPUT PARAMETERS                                   007784
C               CONVERT AR TO LOG BASE 10 AND DECIBELS.                 007785
C                                                                       007786
 1250 IF (AR .NE. 0.D0) GO TO 1270                                      007787
      BELL = -20.D0                                                     007788
      GO TO 1280                                                        007789
 1270 BELL = DLOG(AR)                                                   007790
 1280 DBELL = BELL * 8.6858896ID0                                       007791
      PHI = DPHI / 57.2958D0                                            007792
      W1 = W / 6.283185300                                              007793
C                                                                       007794
 1289 IF (LNCTR - 40) 1320, 1290, 1290                                  007795
 1290 CALL PAGEHD                                                       007796
      WRITE (NOT,158) (TITLE(I3),I3=1,10)                               007797
  158 FORMAT (/,10X10A6,/)                                              007798
      WRITE (NOT,159)                                                   007799
  159 FORMAT (1H0,20X,99HFREQ/RAD/SEC     FREQ/HERTZ        REAL        I 007800
     1MAG              AMP         DECIBELS    RAD        DEG /)        007801
      LNCTR = 1                                                         007802
      IF (KPRINT .EQ. 1) GO TO 1310                                     007803
      KPRINT = 1                                                        007804
```

```
      WRITE (NOT,1620) W,W1,ALPHA,BETA,AR,DBELL,PHI,DPHI                    007805
 1620 FORMAT (17H         ********D16.6,D14.6,D16.6,D14.6,D16.6,F10.3,      007806
     1    F9.4,F10.4,9H ********)                                           007807
      GO TO 1330                                                            007808
 1310 WRITE (NOT,162) W,W1,ALPHA,BETA,AR,DBELL,PHI,DPHI                     007809
  162 FORMAT (9X,D24.6,D14.6,D16.6,D14.6,D16.6,F10.3,F9.4,F10.4)            007810
      GO TO 1370                                                            007811
 1320 IF (KPRINT .EQ. 1) GO TO 1350                                         007812
      WRITE (NOT,1620) W,W1,ALPHA,BETA,AR,DBELL,PHI,DPHI                    007813
      KPRINT = 1                                                            007814
 1330 KWCT = KWCT + 1                                                       007815
      IF (WD(KWCT+1) .LE.0.D0) GO TO 1370                                   007816
      RAT = WD(KWCT) / WD(KWCT+1)                                           007817
      IF (RAT .LE. 0.42D0) GO TO 1370                                       007818
      RAT = (1.+RAT)/2.D0                                                   007819
      JX = 2                                                                007820
      X =DABS(RAT-TABUP(2))                                                 007821
      DO 1340 KKK=3,30                                                      007822
      IF (X .LE.DABS(RAT-TABUP(KKK))) GO TO 1370                            007823
      X =DABS(RAT-TABUP(KKK))                                               007824
      JX = KKK                                                              007825
 1340 CONTINUE                                                              007826
      GO TO 1360                                                            007827
 1350 WRITE (NOT,162) W,W1,ALPHA,BETA,AR,DBELL,PHI,DPHI                     007828
 1360 LNCTR = LNCTR + 1                                                     007829
 1370 CONTINUE                                                              007830
C                                                                           007831
C-----SAVE DATA TO PLOT                                                     007832
C                                                                           007833
      KSAVE = KSAVE + 1                                                     007834
      SAVEO(KSAVE) = W                                                      007835
      SAVED(KSAVE) = DBELL                                                  007836
      SAVEP(KSAVE) = DPHI                                                   007837
      SAVEA(KSAVE) = AR                                                     007838
C                                                                           007839
C                                                                           007840
 1480 CONTINUE                                                              007841
C                                                                           007842
C----------CONTINUE FREQUENCY SWEEP UNTIL LIMITS ARE EXHAUSTED.             007843
C                                                                           007844
      IF (W1 .LT. DBLE(FMAX) .AND. KSAVE .LT. KDSAVE) GO TO 420             007845
C                                                                           007846
C         NORMAL TERMINATION LOGIC.                                         007847
C                                                                           007848
 1490 CONTINUE                                                              007849
C                                                                           007850
      KSAVE = KSAVE-1                                                       007851
C                                                                           007852
      RETURN                                                                007853
C               ERROR EXITS                                                 007854
```

```
C                                                                       007855
 1210 CALL PAGEHD                                                       007856
      WRITE (NOT,135) NRN                                               007857
      GO TO 1490                                                        007858
C                                                                       007859
 1220 CALL PAGEHD                                                       007860
      WRITE (NOT,137) NCN                                               007861
      GO TO 1490                                                        007862
C                                                                       007863
 1230 CALL PAGEHD                                                       007864
      WRITE (NOT,139) NRD                                               007865
      GO TO 1490                                                        007866
C                                                                       007867
C               ERROR EXIT FORMATS                                      007868
C                                                                       007869
 1240 CALL PAGEHD                                                       007870
      WRITE (NOT,141) NCD                                               007871
  135 FORMAT (54HDATA FOR NRN IS INCORRECT IN SUBROUTINE SFREQ2, NRN = ,  007872
     1        I5)                                                       007873
  137 FORMAT (54HDATA FOR NCN IS INCORRECT IN SUBROUTINE SFREQ2, NCN = ,  007874
     1        I5)                                                       007875
  139 FORMAT (54HDATA FOR NRD IS INCORRECT IN SUBROUTINE SFREQ2, NRD = ,  007876
     1        I5)                                                       007877
  141 FORMAT (54HDATA FOR NCD IS INCORRECT IN SUBROUTINE SFREQ2, NCD = ,  007878
     1        I5)                                                       007879
C                                                                       007880
C                                                                       007881
      STOP                                                              007882
      END                                                               007883
```

```
[HDG,P    SHAFTT                                                          -007884
[FOR,IS   SHAFTT                                                          -007885
      COMPILER (XM=1),(EQUIV=CMN)                                         -007886
      SUBROUTINE SHAFTT (TSHFT)                                            007887
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                 -007888
      DIMENSION TSHFT(1)                                                   007889
C                                                                          007890
            COMMON /MAXMUM/                                                007891
     *     NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU               007892
            COMMON /SPECIF/                                                007893
     *     BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),      1607894
     *     DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),             1707895
     *     NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5),  1807896
     *     LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                            1907897
            COMMON /VECTOR/                                                007898
     *     Y(250),YDT(250)                                                2007899
CCCCCCC  THIS COMMON IS TRANSFER BETWEEN CONTRL AND SHAFTT ONLY ----       007900
           COMMON /WHEEL /                                                 007901
     *    CLM(4)                                                          007902
C                                                                          007903
      DATA IIST / 0 /                                                      007904
C                                                                          007905
      IF (IIST .EQ. 1) GO TO 10                                            007906
      IIST = 1                                                             007907
      DO 5 I=1,NMWMAX                                                      007908
    5 TSHFT(I) = 0.D 0                                                     007909
C                                                                          007910
   10 DO 15 I=1,4                                                          007911
   15 TSHFT(I) = CLM(I)                                                    007912
C                                                                          007913
      RETURN                                                               007914
      END                                                                  007915
```

```
[HDG,P    SIFT                                                          -007916
[FOR,IS   SIFT                                                          -007917
      COMPILER (XM=1),(EQUIV=CMN)                                       -007918
      SUBROUTINE SIFT (A,N,TOL)                                          007919
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -007920
      DIMENSION A(1)                                                     007921
C                                                                        007922
C  SUBROUTINE SEARCHES ARRAY, A, FOR SMALL VALUES OF A AND SETS          007923
C              THESE SMALL VALUES TO 0.0.                                007924
C                                                                        007925
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                007926
C                                                                        007927
C  A      = INPUT,OUTPUT VECTOR ARRAY TO BE SCANNED FOR SMALL VALUES.    007928
C  N      = INPUT  SIZE OF A.                                            007929
C  TOL    = INPUT  TOLERANCE.  IF (A(I) .LT. TOL  A(I) = 0.0             007930
C                                                                        007931
      DO 10 I=1,N                                                        007932
      IF (DABS(A(I)) .LT. TOL) A(I) = 0.DO                               007933
   10 CONTINUE                                                           007934
C                                                                        007935
      RETURN                                                             007936
      END                                                                007937
```

```
[HDG,P    SKEWV3                                                    -007938
[FOR,IS   SKEWV3                                                    -007939
      COMPILER (XM=1),(EQUIV=CMN)                                   -007940
      SUBROUTINE SKEWV3(V,SKV,KV,KSKV)                               007941
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                            -007942
C                                                                    007943
      DIMENSION V(KV,1),SKV(KSKV,1)                                  007944
C                                                                    007945
      SKV(2,3) = V(1,1)                                              007946
      SKV(3,1) = V(2,1)                                              007947
      SKV(1,2) = V(3,1)                                              007948
      SKV(3,2) = -SKV(2,3)                                           007949
      SKV(1,3) = -SKV(3,1)                                           007950
      SKV(2,1) = -SKV(1,2)                                           007951
      SKV(1,1) = 0.D 0                                               007952
      SKV(2,2) = 0.D 0                                               007953
      SKV(3,3) = 0.D 0                                               007954
C                                                                    007955
      RETURN                                                         007956
      END                                                            007957
```

```
[HDG,P    SPLOT                                                          -007958
[FOR,IS   SPLOT                                                          -007959
      COMPILER (XM=1),(EQUIV=CMN)                                        -007960
      SUBROUTINE SPLOT (TITLE,FMAX,FMIN,DBMIN,DBMAX)                     -007961
C                                                                        -007962
C ***                                                                    -007963
C *** MSFC UNIVAC 1108 VERSION ***                                       -007964
C ***                                                                    -007965
C-----SUBROUTINE FORMS BODE PLOTS                                        -007966
C                                                                        -007967
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                -007968
C                                                                        -007969
C  TITLE = INPUT ALPHA NUMERIC TITLE                                     -007970
C  FMAX  = INPUT UPPER LIMIT - FREQUENCY SWEEP                           -007971
C  FMIN  = INPUT LOWER LIMIT - FREQUENCY SWEEP                           -007972
C  DBMIN = INPUT MINIMUM DB TO PLOT                                      -007973
C  DBMAX = INPUT MAXIMUM DB TO PLOT                                      -007974
C                                                                        -007975
      COMMON /LSTART/ IRUNNO, IDATE, NPAGE                               -007976
      COMMON /PSTUFF/                                                    -007977
     *          SAVEO(500), SAVEP(500), SAVED(500), SAVEA(500), KSAVE    -007978
      COMMON /ADDPLT/ DB(500),PH(500),X(500)                             -007979
C                                                                        -007980
      DIMENSION TITLE(1),TX(12),TY(12)                                   -007981
C                                                                        -007982
      EQUIVALENCE (IRUNNO,RUNNO)                                         -007983
C                                                                        -007984
      TX( 1) = 6H                                                        -007985
      DO 5 I=1,10                                                        -007986
    5 TX(I+1) = TITLE(I)                                                 -007987
      TX(12) = 6H                                                        -007988
C                                                                        -007989
      CALL SMXYV(1,0)                                                    -007990
C                                                                        -007991
      KNT = 0                                                            -007992
      DO 10 I=1,KSAVE                                                    -007993
      FR = SAVEO(I)                                                      -007994
      IF(FR .LT. FMIN  .OR.  FR .GT. FMAX) GO TO 10                      -007995
      KNT = KNT + 1                                                      -007996
      DB(KNT) = SAVED(I)                                                 -007997
      IF(DB(KNT) .GT. DBMAX) DB(KNT) = DBMAX                             -007998
      IF(DB(KNT) .LT. DBMIN) DB(KNT) = DBMIN                             -007999
      PH(KNT) = SAVEP(I)                                                 -008000
      IF(PH(KNT) .GT. 180.0) PH(KNT) = PH(KNT) - 360.0                   -008001
      X(KNT) = FR                                                        -008002
   10 CONTINUE                                                           -008003
C                                                                        -008004
      TY( 1) = 6HBODE P                                                  -008005
      TY( 2) = 6HLOT                                                     -008006
      TY( 3) = 6H                                                        -008007
```

```
      TY( 4) = 6HGAIN I                                                  -008008
      TY( 5) = 6HN DB V                                                  -008009
      TY( 6) = 6HS FREQ                                                  -008010
      TY( 7) = 6H IN RA                                                  -008011
      TY( 8) = 6HD/SEC                                                   -008012
      TY( 9) = 6H                                                        -008013
      TY(10) = 6H                                                        -008014
      TY(11) = 6H                                                        -008015
      TY(12) = RUNNO                                                     -008016
      CALL PLOTSS(DBMAX,DBMIN,YTOP,YBOT)                                 -008017
      CALL QUKLOG(-1,FMIN,FMAX,YBOT,YTOP,35,TX,TY,-KNT,X,DB)             -008018
C                                                                        -008019
      TY( 4) = 6HPHASE                                                   -008020
      TY( 5) = 6HIN DEG                                                  -008021
      TY( 6) = 6H VS FR                                                  -008022
      TY( 7) = 6HEQ IN                                                   -008023
      TY( 8) = 6HRAD/SE                                                  -008024
      TY( 9) = 6HC                                                       -008025
      CALL PLOTSS(200.,-200.,YTOP,YBOT)                                  -008026
      CALL QUKLOG(-1,FMIN,FMAX,YBOT,YTOP,35,TX,TY,-KNT,X,PH)             -008027
C                                                                        -008028
      CALL SMXYV(0,0)                                                    -008029
C                                                                        -008030
      RETURN                                                             -008031
      END                                                                -008032
```

```
[HDG,P    SUBDIA                                                    -008033
[FOR,IS   SUBDIA                                                    -008034
      COMPILER (XM=1),(EQUIV=CMN)                                   -008035
      SUBROUTINE SUBDIA (A,M,KR,B)                                   006036
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                            -008037
      DIMENSION A(KR,1),P(1)                                         008038
C     SUBROUTINE TO PUT MATRIX IN UPPER HESSENBERG FORM.             006039
      IF (M - 2) 260, 260, 100                                       008040
  100 DO 250 LC = 3,M                                                008041
      N = M - LC + 3                                                 008042
      N1 = N - 1                                                     008043
      N2 = N - 2                                                     006044
      NI = N1                                                        008045
      DIV = DABS(A(N,N-1))                                           008046
      DO 120 J = 1,N2                                                006047
      IF(DABS(A(N,J))- DIV) 120, 120, 110                            008048
  110 NI = J                                                         008049
      DIV = DABS(A(N,J))                                             008050
  120 CONTINUE                                                       008051
      IF(DIV) 130, 250, 130                                          008052
  130 IF(NI - N1) 140, 170, 140                                      008053
  140 DO 150 J = 1,N                                                 008054
      DIV = A(J,NI)                                                  008055
      A(J,NI) = A(J,N1)                                              008056
  150 A(J,N1) = DIV                                                  008057
      DO 160 J = 1,M                                                 008058
      DIV = A(NI,J)                                                  008059
      A(NI,J) = A(N1,J)                                              008060
  160 A(N1,J) = DIV                                                  008061
  170 DO 180 K = 1, N1                                               008062
  180 B(K) = A(N,K)/A(N,N-1)                                         008063
      DO 240 J = 1,M                                                 008064
      SUM = 0.DO                                                     008065
      IF (J - N1) 190, 220, 220                                      008066
  190 IF(B(J))  200, 220, 200                                        008067
  200 A(N,J) = 0.DO                                                  008068
      DO 210 K = 1,N1                                                008069
      A(K,J) = A(K,J) - A(K,N1)*B(J)                                 008070
  210 SUM = SUM + A(K,J)*B(K)                                        008071
      GO TO 240                                                      008072
  220 DO 230 K = 1,N1                                                008073
  230 SUM = SUM + A(K,J)*B(K)                                        008074
  240 A(N1,J) = SUM                                                  006075
  250 CONTINUE                                                       008076
  260 RETURN                                                         008077
      END                                                            008078
```

```
[HDG,P    START                                                          -008079
[FOR,IS   START                                                          -008080
      COMPILER (XM=1),(EQUIV=CMN)                                        -008081
      SUBROUTINE START                                                   -008082
C                                                                        -008083
C ***                                                                    -008084
C *** MSFC UNIVAC 1108 VERSION ***                                       -008085
C ***                                                                    -008086
      COMMON /LSTART/ IRUNNO, IDATE, NPAGE                               -008087
      COMMON /LSTRT1/ UNAME(3), TITLE1(12), TITLE2(12)                   -008088
C                                                                        -008089
      DATA NIT, NOT,ISTOP  / 5, 6, 6HSTOP  /                            -008090
C                                                                        -008091
 1001 FORMAT(A6, 4X, 3A6)                                                -008092
 1002 FORMAT(12A6)                                                       -008093
 2003 FORMAT(36H1END OF INPUT DATA HAS BEEN REACHED.)                    -008094
C                                                                        -008095
      READ(NIT,1001) IRUNNO,UNAME                                        -008096
      IF(IRUNNO .NE.ISTOP) GO TO 10                                      -008097
      WRITE(NOT,2003)                                                    -008098
      STOP                                                               -008099
C                                                                        -008100
   10 READ(NIT,1002) TITLE1                                              -008101
      READ(NIT,1002) TITLE2                                              -008102
      NPAGE = 0                                                          -008103
C                                                                        -008104
      RETURN                                                             -008105
      END                                                                -008106
```

```
[HDG,P    STORE                                                          -008107
[FOR,IS   STORE                                                          -008108
      COMPILER (XM=1),(EQUIV=CMN)                                        -008109
      SUBROUTINE STORE(NTAPE,A,B,Z,NA,NCB,KRA,KRB,KRZ)                    008110
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -008111
C                                                                         008112
C     MATRIX PRODUCT Z = A*B WITH A = DIAGONAL AND STORED AS VECTOR       008113
C         PRODUCT WRITTEN BY ROWS ON NTAPE                                008114
C         WHERE A = INPUT  (NA,NA)                                        008115
C               B = INPUT  (NA,NCB)                                       008116
C               Z = OUTPUT (NA,NCB)                                       008117
C                                                                         008118
      DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)                               008119
C                                                                         008120
      DO 10 I=1,NA                                                       008121
      S = A(I,1)                                                         008122
      DO 10 J=1,NCB                                                      008123
   10 Z(I,J) = S*B(I,J)                                                  008124
C                                                                         008125
      WRITE(NTAPE) ((Z(I,J),J=1,NCB),I=1,NA)                             008126
C                                                                         008127
      RETURN                                                             008128
      END                                                                008129
```

```
[HDG,P     TFPLY                                                        -008130
[FOR,IS    TFPLY                                                        -008131
      COMPILER (XM=1),(EQUIV=CMN)                                       -008132
      SUBROUTINE TFPLY (A,B,U,X,NS,L)                                    008133
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               -008134
C                                                                        008135
      DIMENSION A(1), B(1)                                               008136
C                                                                        008137
C      SUBROUTINE CONVERTS TRANSFER FUNCTION POLYNOMINAL                 008138
C                 EXPRESSIONS TO FIRST ORDER CANONICAL STATE            008139
C                 SPACE FORM AND RETURNS THE TRANSFORMED OUTPUT         008140
C                  VARIABLE, X.                                         008141
C                                                                        008142
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                008143
C                                                                        008144
C  A       = INPUT  VECTOR OF DENOMINATOR POLYNOMINAL                    008145
C                   COEFFICIENTS--ASCENDING ORDER.                       008146
C  B       = INPUT  VECTOR OF NUMERATOR POLYNOMINAL                      008147
C                   COEFFICIENTS--ASCENDING ORDER.                       008148
C  U       = INPUT  STATE VARIABLE TO BE OPERATED ON BY THE              008149
C                   POLYNOMINAL TRANSFER FUNCTION.                       008150
C  X       = OUTPUT VARIABLE RESULTING FROM THE TRANSFER                 006151
C                   FUNCTION OPERATING ON U.                             008152
C  NS      = INPUT  SIZE OF A AND B.                                     008153
C  L       = INPUT  LOCATION (IN STATE VECTOR) OF THE                    008154
C                   LEADING ELEMENT OF THE NS-1 STATE VARIABLES          008155
C                   ESTABLISHED FROM THE POLYNOMINALS.                   008156
C                                                                        008157
C                                                                        008158
      COMMON /VECTOR/                                                    008159
     E                Y   (250), YD  (250)                             43008160
      DATA NIT/ 6 /                                                      008161
C                                                                        008162
C  NORMALIZE A AND B COEFFICIENTS TO COEFFICIENT OF                      008163
C  HIGHEST DERIVATIVE IN DENOMINATOR, A(NS).                             008164
C                                                                        008165
      AN = A(NS)                                                         008166
      IF (AN .EQ. 0.D0) GO TO 999                                        008167
C                                                                        008168
      DO 10 I=1,NS                                                       008169
      A(I) = A(I) / AN                                                   008170
   10 B(I) = B(I) / AN                                                   008171
C                                                                        008172
      BN = B(NS)                                                         008173
C                                                                        008174
C  FORM STATE VECTOR TIME DERIVATIVES AND PUT INTO YDOT                  008175
C  BEGINNING WITH LOCATION L IN YDOT.                                    008176
C                                                                        008177
      DO 20 I=2,NS                                                       008178
C                                                                        008179
```

```
          J = NS-I+1                                                    008180
          K = L+I-2                                                     008181
       IF (I .EQ. NS) GO TO 25                                          008182
C                                                                       008183
   20  YD(K) = -A(J)*Y(L) + Y(K+1) + (B(J)-A(J)*BN)*U                   008184
   25  YD(K) = -A(J)*Y(L) + (B(J)-A(J)*BN)*U                            008185
C                                                                       008186
       X = Y(L) + BN*U                                                  008187
C                                                                       008188
       RETURN                                                           008189
C                                                                       008190
  999  CALL PAGEHD                                                      008191
       WRITE (NIT,1001)                                                 008192
 1001  FORMAT (///,10X,33HCOEFFICIENT OF HIGHEST ,                      008193
     *          /,10X,32HDERIVATIVE OF DENOMINATOR CANNOT ,            008194
     *         /,10X,17HBE EQUAL TO ZERO. ,                            008195
     *         //,10X,16HPROGRAM STOPPED.)                             008196
C                                                                       008197
       STOP                                                             008198
       END                                                              008199
```

```
[HDG,P    TFTYPE                                                        -008200
[FOR,IS   TFTYPE                                                        -008201
      COMPILER (XM=1),(EQUIV=CMN)                                       -008202
      SUBROUTINE TFTYPE (A,Z,B,NA,NZ,ITYPE,JCOL,NBKP,KBKP,KA,KZ)         008203
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -008204
C                                                                       008205
      COMMON /LDSIZE/                                                   008206
     2               NX, NY, NDLTA, NXSS, NB, NJQ, NY2, ND2             008207
C                                                                       008208
      DIMENSION A(KA,1), Z(KZ,1), B(1), KBKP(1)                         008209
C                                                                       008210
C        ----- SUBROUTINE ARGUMENT DESCRIPTIONS -----                  008211
C                                                                       008212
C  A       = INPUT  PARTIAL DERIVATIVE MATRIX                           008213
C  Z       = OUTPUT  REDUCED PARTIAL DERIVATIVE MATRIX.  (NZ,NZ)        008214
C  B       = OUTPUT  VECTOR OF COEFF. FOR DESIRED TF INPUT. (NZ,1)      008215
C  NA      =  INPUT  SIZE OF A.                                         008216
C  NZ      = OUTPUT  SIZE OF Z                                          008217
C  ITYPE   = INPUT  =1  FORWARD PATH TF    XSS(I)/RT(J)                008218
C                    2  FEEDBACK TF       B(I)/RS(J)                  008219
C                    3  OPEN LOOP TF      B(I)/RT(J)                  008220
C                    4  OPEN LOOP TF      XSS(I)/RS(J)                008221
C                    5 CLOSED LOOP TF     XSS(I)/RT(J)                008222
C                    6 CLOSED LOOP TF     XSS(I)/RS(J)                008223
C                   7 PARTIAL OPEN LOOP  B(I)/RT(J)                  008224
C                                                                       008225
C             NOTE-- A MINUS SIGN ON ITYPE INDICATES                    008226
C                    NEGATIVE FEEDBACK FOR NUMERATOR                    008227
C                    AUGMENTATION SELECTION OF PROPER B.                008228
C                                                                       008229
C  JCOL    = INPUT  COL LOCATION IN A OF DESIRED INPUT(J). LOCAL        008230
C  NBKP    = INPUT NO. OF B"S TO RETAIN  ITYPE=7                        008231
C  KBKP    = INPUT ID VECTOR NOTING WHICH B"S TO KEEP (LOCAL)           008232
C  KA      = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM               008233
C  KZ      = INPUT  ROW DIMENSION OF Z IN CALLING PROGRAM               008234
C                                                                       008235
C        ESTABLISH LEADING ELE LOCATORS FOR EACH PARTITION OF A         008236
C        ASSUMED ORDER IS Y,XSS,DELTA,B                                 008237
C                                                                       008238
      LY = 1                                                            008239
      LX = LY + NY2                                                     008240
      LD = LX + NXSS                                                    008241
      LB = LD + ND2                                                     008242
      XSN = ISIGN(1,ITYPE)                                              008243
      ITYPE = IABS(ITYPE)                                               008244
C                                                                       008245
                  NERROR = 1                                            008246
      IF (ITYPE .LT. 1  .OR.  ITYPE .GT. 7) GO TO 999                   008247
                  NERROR = 2                                            008248
      IF (JCOL .LT. 0  .OR.  JCOL .GT. NA) GO TO 999                    008249
```

```
C                                                                       008250
      GO TO (1,2,3,4,5,6,7),ITYPE                                       008251
C                                                                       008252
    1 CONTINUE                                                          008253
C               -----ITYPE = 1 -----                                    008254
C                                                                       008255
C       FORM Z = A11,A12                                                008256
C                A21,A22                                                008257
C                                                                       008258
C            B = A14(JCOL)*XSN                                          008259
C                A24(JCOL)*XSN                                          008260
C                                                                       008261
      NZ = NY2 + NXSS                                                   008262
      KCOL = LB-1+JCOL                                                  008263
      DO 10 I=1,NZ                                                      008264
      B(I) = A(I,KCOL) * XSN                                            008265
      DO 10 J=1,NZ                                                      008266
   10 Z(I,J) = A(I,J)                                                   008267
      RETURN                                                            008268
C                                                                       008269
    2 CONTINUE                                                          008270
C               -----ITYPE = 2 -----                                    008271
C                                                                       008272
C       FORM Z = A33,A34                                                008273
C                A43,A44                                                008274
C                                                                       008275
C            B = A32(JCOL)                                              008276
C                A42(JCOL)                                              008277
C                                                                       008278
      NZ = ND2 + NB                                                     008279
      KCOL = LX-1+JCOL                                                  008280
      CALL ZERO (Z,NZ,NZ,KZ)                                            008281
      DO 20 I=1,NZ                                                      008282
      IRA = I + NY2 + NXSS                                              008283
      B(I) = A(IRA,KCOL)                                                008284
      DO 20 J=1,NZ                                                      008285
      JCA = J + NY2 + NXSS                                              008286
      Z(I,J) = A(IRA,JCA)                                               008287
   20 CONTINUE                                                          008288
      RETURN                                                            008289
C                                                                       008290
    3 CONTINUE                                                          008291
C               -----ITYPE = 3 -----                                    008292
C                                                                       008293
C       FORM Z = A11,A12,  0,  0                                        008294
C                A21,A22,  0,  0                                        008295
C                  0,A32,A33,A34                                        008296
C                  0,A42,A43,A44                                        008297
C                                                                       008298
C            B = A14(JCOL)*XSN                                          008299
```

```
C                A24(JCOL)*XSN                                          008300
C                  0                                                    008301
C                  0                                                    008302
C                                                                       008303
      NZ = NY2 + NXSS + ND2 + NB                                        008304
      KCOL = LB-1+JCOL                                                  008305
      CALL ZERO (Z,NZ,NZ,KZ)                                            008306
      M = NY2 + NXSS                                                    008307
      DO 25 I=1,M                                                       008308
      B(I) = A(I,KCOL) * XSN                                            008309
      DO 25 J=1,M                                                       008310
   25 Z(I,J) = A(I,J)                                                   008311
      DO 30 I=LD,NZ                                                     008312
      B(I) = 0.D0                                                       008313
      DO 30 J=LX,NZ                                                     008314
      Z(I,J) = A(I,J)                                                   008315
   30 CONTINUE                                                          008316
      RETURN                                                            008317
C                                                                       008318
    4 CONTINUE                                                          008319
C               -----ITYPE = 4 -----                                    008320
C       FORM Z = A11,A12,  0,A14                                        008321
C                A21,A22,  0,A24                                        008322
C                  0,  0,A33,A34                                        008323
C                  0,  0,A43,A44                                        008324
C                                                                       008325
C             B =    0                                                  008326
C                    0                                                  008327
C                  A32(JCOL)                                            008328
C                  A42(JCOL)                                            008329
C                                                                       008330
      NZ = NY2 + NXSS + ND2 + NB                                        008331
      KCOL = LX-1+JCOL                                                  008332
      CALL ZERO (Z,NZ,NZ,KZ)                                            008333
      M = NY2 + NXSS                                                    008334
      DO 35 I=1,M                                                       008335
      B(I) = 0.D0                                                       008336
      DO 35 J=1,NZ                                                      008337
      IF (J .GE. LD  .AND.  J .LT. LB) GO TO 35                         008338
      Z(I,J) = A(I,J)                                                   008339
   35 CONTINUE                                                          008340
      DO 40 I=LD,NZ                                                     008341
      B(I) = A(I,KCOL)                                                  008342
      DO 40 J=LD,NZ                                                     008343
      Z(I,J) = A(I,J)                                                   008344
   40 CONTINUE                                                          008345
      RETURN                                                            008346
C                                                                       008347
    5 CONTINUE                                                          008348
C               -----ITYPE = 5 -----                                    008349
```

```
C      FORM Z = A11,A12,  0,A14                                          008350
C               A21,A22,  0,A24                                          008351
C                 0,A32,A33,A34                                          008352
C                 0,A42,A43,A44                                          008353
C                                                                        008354
C           B = A14(JCOL)*XSN                                            008355
C               A24(JCOL)*XSN                                            008356
C                 0                                                      008357
C                 0                                                      008358
      NZ = NY2 + NXSS + ND2 + NB                                         008359
      KCOL = LB-1+JCOL                                                   008360
      M = NY2 + NXSS                                                     008361
      CALL ZERO (Z,NZ,NZ,KZ)                                             008362
      DO 45 I = 1,M                                                      008363
      B(I) = A(I,KCOL) * XSN                                             008364
      DO 45 J=1,NZ                                                       008365
      IF (J .GE. LD  .AND.  J .LT. LB) GO TO 45                          008366
      Z(I,J) = A(I,J)                                                    008367
   45 CONTINUE                                                           008368
      DO 50 I=LD,NZ                                                      008369
      B(I) = 0.DO                                                        008370
      DO 50 J=LX,NZ                                                      008371
      Z(I,J) = A(I,J)                                                    008372
   50 CONTINUE                                                           008373
      RETURN                                                             008374
C                                                                        008375
    6 CONTINUE                                                           008376
C              -----ITYPE = 6 -----                                      008377
C      FORM Z = A11,A12,  0,A14                                          008378
C               A21,A22,  0,A24                                          008379
C                 0 ,A32,A33,A34                                         008380
C                 0,A42,A43,A44                                          008381
C                                                                        008382
C           B =  0                                                       008383
C                0                                                       008384
C               A32                                                      008385
C               A42                                                      008386
C                                                                        008387
      NZ = NY2 + NXSS + ND2 + NB                                         008388
      KCOL = LX-1+JCOL                                                   008389
      CALL ZERO (Z,NZ,NZ,KZ)                                             008390
      M = NY2 + NXSS                                                     008391
      DO 55 I=1,M                                                        008392
      B(I) = 0.DO                                                        008393
      DO 55 J=1,NZ                                                       008394
      IF (J.GE. LD  .AND. J .LT.LB) GO TO 55                             008395
      Z(I,J) = A(I,J)                                                    008396
   55 CONTINUE                                                           008397
      DO 60 I=LD,NZ                                                      008398
      B(I) = A(I,KCOL)                                                   008399
```

```
      DO 60 J=LX,NZ                                                     008400
      Z(I,J) = A(I,J)                                                   008401
   60 CONTINUE                                                          008402
      RETURN                                                            008403
C                                                                       008404
C                                                                       008405
    7 CONTINUE                                                          008406
C               -----ITYPE = 7-----                                     008407
C                                                                       008408
C      FORM Z = A11,A12, 0 ,(A14)                                       008409
C               A21,A22, 0 ,(A24)                                       008410
C                0 ,A32,A33,A34                                         008411
C                0 ,A42,A43,A44                                         008412
C                                                                       008413
C            B = A14(JCOL)*XSN                                          008414
C                A24(JCOL)*XSN                                          008415
C                0                                                      008416
C                0                                                      008417
C                                                                       008418
      NZ = NY2 + NXSS + ND2 + NB                                        008419
      KCOL = LB-1+JCOL                                                  008420
      CALL ZERO (Z,NZ,NZ,KZ)                                            008421
      M = NY2 + NXSS                                                    008422
      DO 65 I=1,M                                                       008423
      B(I) = A(I,KCOL) * XSN                                            008424
      DO 62 J=1,M                                                       008425
   62 Z(I,J) = A(I,J)                                                   008426
      DO 63 J=1,NBKP                                                    008427
      LCOL = LB-1+KBKP(J)                                               008428
   63 Z(I,LCOL) = A(I,LCOL)                                             008429
   65 CONTINUE                                                          008430
      DO 70 I=LD,NZ                                                     008431
      B(I) = 0.D0                                                       008432
      DO 70 J=LX,NZ                                                     008433
      Z(I,J) = A(I,J)                                                   008434
   70 CONTINUE                                                          008435
      RETURN                                                            008436
  999 CONTINUE                                                          008437
      WRITE (6,2001) NERROR                                             008438
 2001 FORMAT (1H1,5X,48HPROGRAM STOPPED IN SUBROUTINE TFTYPE.  NERROR =  008439
     *   , I3)                                                          008440
C                                                                       008441
      STOP                                                              008442
      END                                                               008443
```

```
[HDG,P    TORQUE                                                          -008444
[FOR,IS   TORQUE                                                          -008445
      COMPILER (XM=1),(EQUIV=CMN)                                         -008446
      SUBROUTINE TORQUE (G)                                                008447
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                  -008448
      DIMENSION G(1)                                                       008449
C                                                                          008450
            COMMON /BHBSRD/                                                008451
     *     BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)                  208452
            COMMON /GGSAVE/                                                008453
     *     GGS( 6,9, 5)                                                   308454
            COMMON /INTGRL/                                                008455
     *     AM( 78, 5),ACOF(9, 6, 5),BCOF(6, 6, 5),                        508456
     *     COF11( 6, 6, 5),COF22( 6, 6, 5),COF33( 6, 6, 5),AK( 6, 6, 5),  608457
     *     COF12( 6, 6, 5),COF13( 6, 6, 5),COF23( 6, 6, 5),AD( 6, 6, 5),  708458
     *     COFXY( 6, 6, 5),COFXZ( 6, 6, 5),COFYZ( 6, 6, 5)                808459
            COMMON /MAXMUM/                                                008460
     *     NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMOBOD,KMU,KY,KU              008461
            COMMON /MOMENG/                                                008462
     *     P( 65),PMOM(30),HTOT(3),TOTL(3),ENGKE( 5),ENGPE( 5),           1108463
     *     TOTKE, TOTPE, TOTENG, AHTOT,ATOTL                               008464
            COMMON /NUMBRS/                                                008465
     *     ZRO,ONE,TWO,TRES                                                008466
            COMMON /SPECIF/                                                008467
     *     BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),      1608468
     *     DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),             1708469
     *     NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1808470
     *     LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                            1908471
            COMMON /VECTOR/                                                008472
     *     Y(250),YDT(250)                                                2008473
C                                                                          008474
      DIMENSION CW( 6,3),RW(3, 6),V1( 6),WSK(3,3),TSHFT( 5),               8708475
     *     TEX(6,10),ISPN(10),V(6),V2( 6)                                  8808476
C                                                                          008477
CCC   SUBROUTINE CONTRL ESTABLISHES THE D/DT(DELTAS)  USER SUPPLIED --     008478
      CALL CONTRL                                                          008479
C                                                                          008480
CCC   SUBROUTINE EXTOR ESTABLISHES ALL EXTERNAL TORQUES, INCLUDING RCS     008481
CCCC  CONTROL TORQUES, ETC.,  USER SUPPLIED --                             008482
      CALL EXTOR (TEX,ISPN,NTEX)                                           008483
      IF (NTEX .EQ. 0) GO TO 5                                             008484
C                                                                          008485
      DO 65 L=1,NTEX                                                       008486
      NSP = ISPN(L)                                                        008487
      NBOD = IFTSMW(NSP)                                                   008488
      LON = LOCU(NBOD) - 1                                                 008489
      LEN = IRGFLX(NBOD) + 6                                               008490
      DO 66 I=1,6                                                          008491
      TQ = TEX(I,L)                                                        008492
      IF (TQ .EQ. ZRO) GO TO 66                                            008493
```

```
      DO 67 J=1,LEN                                                    008494
      JL = J + LON                                                     008495
   67 G(JL) = G(JL) + TQ*BS(I,J,NSP)                                   008496
   66 CONTINUE                                                         008497
   65 CONTINUE                                                         008498
C                                                                      008499
CCC   SUBROUTINE SHAFTT ESTABLISHES SHAFT TORQUE FOR EACH              008500
CCCC  MOMENTUM WHEEL (ZEROS IT OUT IF CONSTANT SPEED) USER SUPPLIED --  008501
    5 CALL SHAFTT (TSHFT)                                              008502
C                                                                      008503
CCCC   SETUP HINGE SPRING AND DASHPOT RESTORING TORQUES                008504
CCCC   ALSO ACCOUNT FOR POTENTIAL ENERGY DUE TO HINGE SPRINGS          008505
      CALL KHINGE (G)                                                  008506
C                                                                      008507
C                                                                      008508
      KM = NMDBOD                                                      008509
      DO 50 N=1,NB                                                     008510
      LOU = LOCU(N)                                                    008511
      LO = LOCU(N+NB)                                                  008512
      LE = LENU(N+NB)                                                  008513
      LEU = LE * 6                                                     008514
      IF (LE .EQ. 0) LEU = 3                                           008515
      LOU1 = LOU + 1                                                   008516
      LOU2 = LOU + 2                                                   008517
      LOU3 = LOU + 3                                                   008518
      LOU4 = LOU + 4                                                   008519
      LOU5 = LOU + 5                                                   008520
      CALL SKEWV3 (Y(LOU),WSK,1,3)                                     008521
      CALL MULTAD (WSK,P(LOU),G(LOU),3,3,1,3,1,1)                      008522
      CALL MULTAD (WSK,P(LOU3),G(LOU3),3,3,1,3,1,1)                    008523
      CALL SKEWV3 (Y(LOU3),WSK,1,3)                                    008524
      CALL MULTAD (WSK,P(LOU3),G(LOU),3,3,1,3,1,1)                     008525
      IF (LE .EQ. 0) GO TO 100                                         008526
      CALL GMISC (N,LE,LO,V2)                                          008527
      CALL MULT3  (AK(1,1,N),V2,V1,LE,LE,1,KM,1,1)                     008528
      CALL MULTAD (AD(1,1,N),YDT(LO),V1,LE,LE,1,KM,1,1)                008529
      DO 10 J=1,LE                                                     008530
      I = LOU5 + J                                                     008531
   10 G(I) = G(I) - V1(J)                                              008532
C                                                                      008533
      DO 15 J=1,LE                                                     008534
      CW(J,1) = -TWO*Y(LOU )*(BCOF(1,J,N) + GGS(J,1,N))                008535
     *    + Y(LOU1)*(BCOF(4,J,N) + GGS(J,4,N) + GGS(J,7,N))            008536
     *    + Y(LOU2)*(BCOF(5,J,N) + GGS(J,5,N) + GGS(J,8,N))            008537
     *  -Y(LOU3)*ACOF(1,J,N) - Y(LOU4)*ACOF(2,J,N) - Y(LOU5)*ACOF(3,J,N) 008538
      CW(J,2) = -TWO*Y(LOU1)*(BCOF(2,J,N) + GGS(J,2,N))                008539
     *    + Y(LOU2)*(BCOF(6,J,N) + GGS(J,6,N) + GGS(J,9,N))            008540
     *    + Y(LOU )*(BCOF(4,J,N) + GGS(J,4,N) + GGS(J,7,N))            008541
     *  -Y(LOU3)*ACOF(4,J,N) - Y(LOU4)*ACOF(5,J,N) - Y(LOU5)*ACOF(6,J,N) 008542
      CW(J,3) = -TWO*Y(LOU2)*(BCOF(3,J,N) + GGS(J,3,N))                008543
```

```
     *    + Y(LOU )*(BCOF(5,J,N) + GGS(J,5,N) + GGS(J,8,N))                  008544
     *    + Y(LOU1)*(BCOF(6,J,N) + GGS(J,6,N) + GGS(J,9,N))                  008545
     *  -Y(LOU3)*ACOF(7,J,N) - Y(LOU4)*ACOF(8,J,N) - Y(LOU5)*ACOF(9,J,N)     008546
   15 CONTINUE                                                              008547
      CALL MULTAD (YDT(LO),CW,G(LOU),1,LE,3,1,KM,1)                         008548
      CALL MULT3  (COFXY(1,1,N),YDT(LO),CW(1,1),LE,LE,1,KM,1,KM)            008549
      CALL MULT3  (COFXZ(1,1,N),YDT(LO),CW(1,2),LE,LE,1,KM,1,KM)            008550
      CALL MULT3  (COFYZ(1,1,N),YDT(LO),CW(1,3),LE,LE,1,KM,1,KM)            008551
      CALL MULT3  (YDT(LO),CW,V,1,LE,3,1,KM,1)                              008552
      G(LOU ) = G(LOU ) - V(3)                                              008553
      G(LOU1) = G(LOU1) - V(2)                                              008554
      G(LOU2) = G(LOU2) - V(1)                                              008555
      DO 18 J=1,LE                                                          008556
      CW(J,1) = -Y(LOU )*ACOF(1,J,N) - Y(LOU1)*ACOF(4,J,N)                  008557
     *    -Y(LOU2)*ACOF(7,J,N)                                              008558
      CW(J,2) = -Y(LOU )*ACOF(2,J,N) - Y(LOU1)*ACOF(5,J,N)                  008559
     *    -Y(LOU2)*ACOF(8,J,N)                                              008560
      CW(J,3) = -Y(LOU )*ACOF(3,J,N) - Y(LOU1)*ACOF(6,J,N)                  008561
     *    -Y(LOU2)*ACOF(9,J,N)                                              008562
   18 CONTINUE                                                              008563
      CALL MULTAD (YDT(LO),CW,G(LOU3),1,LE,3,1,KM,1)                        008564
C                                                                           008565
      DO 20 J=1,LE                                                          008566
      I = LOU5 + J                                                          008567
      G(I) = G(I) + (Y(LOU )**2)*(BCOF(1,J,N) + GGS(J,1,N))                 008568
     *            + (Y(LOU1)**2)*(BCOF(2,J,N) + GGS(J,2,N))                 008569
     *            + (Y(LOU2)**2)*(BCOF(3,J,N) + GGS(J,3,N))                 008570
     *  - Y(LOU )*Y(LOU1)*(BCOF(4,J,N) + GGS(J,4,N) + GGS(J,7,N))           008571
     *  - Y(LOU )*Y(LOU2)*(BCOF(5,J,N) + GGS(J,5,N) + GGS(J,8,N))           008572
     *  - Y(LOU1)*Y(LOU2)*(BCOF(6,J,N) + GGS(J,6,N) + GGS(J,9,N))           008573
     *   + Y(LOU )*(Y(LOU3)*ACOF(1,J,N) + Y(LOU4)*ACOF(2,J,N)               008574
     *     + Y(LOU5)*ACOF(3,J,N))                                           008575
     *   + Y(LOU1)*(Y(LOU3)*ACOF(4,J,N) + Y(LOU4)*ACOF(5,J,N)               008576
     *     + Y(LOU5)*ACOF(6,J,N))                                           008577
     *   + Y(LOU2)*(Y(LOU3)*ACOF(7,J,N) + Y(LOU4)*ACOF(8,J,N)               008578
     *     + Y(LOU5)*ACOF(9,J,N))                                           008579
   20 CONTINUE                                                              008580
C                                                                           008581
      CALL MULT3  (COFXY(1,1,N),YDT(LO),CW(1,1),LE,LE,1,KM,1,KM)            008582
      CALL MULT3  (COFXZ(1,1,N),YDT(LO),CW(1,2),LE,LE,1,KM,1,KM)            008583
      CALL MULT3  (COFYZ(1,1,N),YDT(LO),CW(1,3),LE,LE,1,KM,1,KM)            008584
      CALL MULT3  (YDT(LO),COFXY(1,1,N),RW(1,1),1,LE,LE,1,KM,3)             008585
      CALL MULT3  (YDT(LO),COFXZ(1,1,N),RW(2,1),1,LE,LE,1,KM,3)             008586
      CALL MULT3  (YDT(LO),COFYZ(1,1,N),RW(3,1),1,LE,LE,1,KM,3)             008587
      DO 30 J=1,LE                                                          008588
      I = LOU5 + J                                                          008589
      G(I) = G(I) + Y(LOU )*(CW(J,3) - RW(3,J))                             008590
     *            + Y(LOU1)*(CW(J,2) - RW(2,J))                             008591
     *            + Y(LOU2)*(CW(J,1) - RW(1,J))                             008592
   30 CONTINUE                                                              008593
```

```
C                                                                       008594
  100 NMON = NMOW(1,N)                                                  008595
      IF (NMON .EQ. 0) GO TO 50                                         008596
      IC = 0                                                            008597
      DO 35 I=1,NMON                                                    008598
      IP2 = I + 2                                                       008599
      NW = NMOW(IP2,N)                                                  008600
      IF (IMO(3,NW) .EQ. 0) GO TO 37                                    008601
      IC = IC + 1                                                       008602
      LMO = LOU5 + LE + IC                                              008603
      TDTJ = Y(LMO)*AMO(2,NW)                                           008604
      GO TO 38                                                          008605
   37 TDTJ = AMO(1,NW)*AMO(2,NW)                                        008606
   38 NPTS = IMO(1,NW)                                                  008607
      NAX  = IMO(2,NW)                                                  008608
      CALL MULT3  (BS(1,1,NPTS),Y(LOU),V(4),3,LEU,1,6,1,1)              008609
      GO TO (41,42,43), NAX                                             008610
   41 V(1) = ZRO                                                        008611
      V(2) = -V(6)*TDTJ                                                 008612
      V(3) =  V(5)*TDTJ                                                 008613
      GO TO 40                                                          008614
   42 V(1) =  V(6)*TDTJ                                                 008615
      V(2) = ZRO                                                        008616
      V(3) = -V(4)*TDTJ                                                 008617
      GO TO 40                                                          008618
   43 V(1) = -V(5)*TDTJ                                                 008619
      V(2) =  V(4)*TDTJ                                                 008620
      V(3) = ZRO                                                        008621
   40 CALL MULTAD (V,BS(1,1,NPTS),G(LOU),1,3,LEU,1,6,1)                 008622
      IF (IMO(3,NW) .EQ. 0) GO TO 35                                    008623
      G(LMO) = TSHFT(NW)                                                008624
      IF (LE .EQ. 0) GO TO 35                                           008625
      CALL MULT3  (BS(1,7,NPTS),YDT(LO),V,3,LE,1,6,1,1)                 008626
      CALL SKEWV3 (V,WSK,1,3)                                           008627
      CALL MULT3  (WSK,V(4),V,3,3,1,3,1,1)                              008628
      G(LMO) = G(LMO) - AMO(2,NW)*V(NAX)                                008629
   35 CONTINUE                                                          008630
C                                                                       008631
   50 CONTINUE                                                          008632
C                                                                       008633
CCC   SUBROUTINE EQADD ESTABLISHES ADDITIONAL CONTROL BLOCK EQUATIONS   008634
CCCC  TO SET UP SIMILARITY TRANSFORMATION. USED ONLY FOR LINEARIZATION  008635
CCCC  AND STABILITY PACKAGE.  USER SUPPLIED --                          008636
      CALL EQADD                                                        008637
C                                                                       008638
      RETURN                                                            008639
      END                                                               008640
```

```
[HDG,P    TREB                                                          -008641
[FOR,IS   TREB                                                          -008642
      COMPILER (XM=1),(EQUIV=CMN)                                       -008643
      SUBROUTINE TREB  (ND,RX,KR,KC,KZ,FBR,FBC,GG,ZOV,KSIZE)             008644
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -008645
CTREB         TRANSFER INPUT ROOTS FORMB AND FORMC                        008646
C    ND -- IF INPUT (0) WE HAVE NUMERATOR, IF INPUT (1) A DENOMINATOR     008647
C      RX -- ENTIRE BLOCK (COUNTS,GAIN,ROOTS)                            008648
C       KR -- RUNNING COUNT OF ACCUMULATED REALS FOR ANY GIVEN CASE      008649
C        KC -- SAME AS ABOVE BUT FOR COMPLEX                             008650
C         KZ -- COUNT OF ACCUMULATED ZEROS FOR ANY GIVEN CASE            008651
C          FBR -- FORM (B) REAL STORAGE BLOCK                            008652
C           FBC -- FORM (B) COMPLEX STORAGE BLOCK                        008653
C            GG -- RUNNING GAIN TERM                                     008654
C              ZOV -- IF OUTPUT OTHER THAN ZERO, ABSF(ZETA) EXCEEDED (1) 008655
C                KSIZE =  DIMENSIONED SIZE OF FBR AND FBC .              008656
C  IT IS ASSUMED THAT THE COUNT FOR THE ACCUMULATED ROOTS WILL BE ZEROED  008657
C  OUT AT THE BEGINNING OF EACH CASE.                                     008658
C  ANOTHER TASK IN THE (MAIN) PROGRAM IS CHECKING THE (ZETA) FLAG.        008659
      DIMENSION RX(1),FBR(1),FBC(1)                                     008660
      IF (GG .EQ. 0.D0) RETURN                                          008661
      IF (ND.GT.0) GO TO 90                                             008662
      IF (RX(7) .EQ. 0.D0) GO TO 200                                    008663
      GG=GG*RX(7)                                                       008664
   90 KRX1 = RX(1)+ 0.1D0                                               008665
      KRX2 = RX(2)+ 0.1D0                                               008666
      KRX3 = RX(3)+ 0.1D0                                               008667
      KRX4 = RX(4)+ 0.1D0                                               008668
      KRX5 = RX(5)+ 0.1D0                                               008669
      KRX6 = RX(6)+ 0.1D0                                               008670
      IF (ND) 100, 100, 110                                             008671
  100 J = 7                                                             008672
      JCR = KRX1                                                        008673
      JCC = 2*KRX2                                                      008674
      JCZ = KRX3                                                        008675
      GO TO 120                                                         008676
  110 J = 7 + KRX1 + 2 * KRX2                                           008677
      JCR = KRX4                                                        008678
      JCC = 2*KRX5                                                      008679
      JCZ = KRX6                                                        008680
  120 IF (JCR) 150, 150, 130                                            008681
  130 DO 140 M = 1,JCR                                                  008682
      KR = KR+1                                                         008683
      L = J+M                                                           008684
      FBR(KR) = RX(L)                                                   008685
  140 CONTINUE                                                          008686
  150 IF (JCC) 190,190,160                                              008687
  160 DO 180 M = 2,JCC,2                                                008688
      KC = KC+1                                                         008689
      KK = 2*KC                                                         008690
```

```
      L = J+JCR+M                                                       008691
      FBC(KK-1) = RX(L-1)                                               008692
      FBC(KK) = RX(L)                                                   008693
      IF (DABS(RX(L-1))- 1.D0) 180, 180, 170                            008694
  170 ZOV = 1.D0                                                        008695
  180 CONTINUE                                                          008696
  190 KZ = KZ+JCZ                                                       008697
      RETURN                                                            008698
  200 GG=0.D0                                                           008699
      KR=0                                                              008700
      KC=0                                                              008701
      KZ=0                                                              008702
      DO 210 I=1,KSIZE                                                  008703
      FBR(I)=0.D0                                                       008704
  210 FBC(I)=0.D0                                                       008705
      RETURN                                                            008706
      END                                                               008707
```

```
[HDG,P    TTFF                                                          -008708
[FOR,IS   TTFF                                                          -008709
      COMPILER (XM=1),(EQUIV=CMN)                                       -008710
      SUBROUTINE TTFF (NR,NC,NZ,KR,KC,KZ,G,RN,CN,RD,CD,R,KSIZE)         008711
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -008712
C                                                                        008713
C --- NR = NUMBER OF REAL ROOTS IN THE NUMERATOR                        008714
C --- NC = NUMBER OF COMPLEX PAIRS IN THE NUMERATOR                     008715
C --- NZ = NUMBER OF ZERO ROOTS IN THE NUMERATOR                        008716
C --- KR = NUMBER OF REAL ROOTS IN THE DENOMINATOR                      008717
C --- KC = NUMBER OF COMPLEX PAIRS IN THE DENOMINATOR                   008718
C --- KZ = NUMBER OF ZERO ROOTS IN THE DENOMINATOR                      008719
C --- G = GAIN                                                          008720
C --- RN = NUMERATOR REAL ROOT ARRAY                                    008721
C --- CN = NUMERATOR COMPLEX PAIRS ARRAY.                               008722
C --- RD = DENOMINATOR REAL ROOT ARRAY                                  008723
C --- CD = DENOMINATOR COMPLEX PAIRS ARRAY                              008724
C --- R = ARRAY CONTAINING NUMBER OF ROOTS AND ROOT ARRAYS.             008725
C -KSIZE- = DIMENSIONED SIZE OF R IN CALLING PROGRAM.                   008726
C                                                                        008727
      DIMENSION RN(1),CN(1),RD(1),CD(1),R(1)                            008728
      IF (G.EQ.0.D0) GO TO 80                                           008729
      R(1)=NR                                                           008730
      R(2)=NC                                                           008731
      R(3)=NZ                                                           008732
      R(4)=KR                                                           008733
      R(5)=KC                                                           008734
      R(6)=KZ                                                           008735
      R(7)=G                                                            008736
      L=7+NR                                                            008737
      IF (NR.EQ.0.D0) GO TO 20                                          008738
      DO 10 I=8,L                                                       008739
   10 R(I)=RN(I-7)                                                      008740
   20 M=L+1                                                             008741
      L=L+2*NC                                                          008742
      IF(NC.LE.0) GO TO 40                                              008743
      DO 30 I=M,L                                                       008744
      J=I-M+1                                                           008745
   30 R(I)=CN(J)                                                        008746
   40 M=L+1                                                             008747
      L=L+KR                                                            008748
      IF(KR.LE.0) GO TO 60                                              008749
      DO 50 I=M,L                                                       008750
      J=I-M+1                                                           008751
   50 R(I)=RD(J)                                                        008752
   60 IF(KC.LE.0) RETURN                                                008753
      M=L+1                                                             008754
      L=L+2*KC                                                          008755
      DO 70 I=M,L                                                       008756
      J=I-M+1                                                           008757
```

```
70 R(I)=CD(J)                                                    008758
   RETURN                                                        008759
80 DO 90 I=1,KSIZE                                               008760
90 R(I)=0.0D0                                                    008761
   RETURN                                                        008762
   END                                                           008763
```

```
[HDG,P    UNITY                                                          -008764
[FOR,IS   UNITY                                                          -008765
      COMPILER (XM=1),(EQUIV=CMN)                                        -008766
      SUBROUTINE UNITY  (Z,N,KR)                                          008767
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -008768
      DIMENSION  Z(KR,1)                                                  008769
C                                                                         008770
C  GENERATE A UNITY MATRIX. (ONES ON THE DIAGONAL).                       008771
C  CODED BY  RL WOHLEN.  FEB 1965.                                        008772
C                                                                         008773
C      SUBROUTINE ARGUMENTS                                               008774
C  Z   = OUTPUT MATRIX GENERATED. SIZE(N,N).                              008775
C  N   = INPUT  SIZE OF MATRIX Z (SQUARE).                                008776
C  KR  = INPUT  ROW DIMENSION OF MATRIX Z IN CALLING PROGRAM.             008777
C                                                                         008778
      DO 20 I=1,N                                                        008779
      DO 10 J=1,N                                                        008780
   10 Z(I,J) = 0.D 0                                                     008781
   20 Z(I,I) = 1.D 0                                                     008782
      RETURN                                                             008783
      END                                                                008784
```

```
[HDG,P    WRITE                                                          -008785
[FOR,IS   WRITE                                                          -008786
      COMPILER (XM=1),(EQUIV=CMN)                                        -008787
      SUBROUTINE WRITE  (A,NR,NC,ANAME,KR)                                008788
      DOUBLE PRECISION A                                                 -008789
      DIMENSION A(KP,1)                                                   008790
      DATA NOT / 6/                                                       008791
C                                                                         008792
C  WRITE MATRIX OF REAL NUMBERS ON PAPER.                                 008793
C  REQUIRES 123 COLUMN (MINIMUM) PRINTER.                                 008794
C  UP TO 10 DATA FIELDS PER LINE. PRINTS ONLY NON-ZERO FIELD ROWS.        008795
C  CALLS FORMA SUBROUTINE PAGEHD.                                         008796
C  CODED BY  RL WOHLEN.  DECEMBER 1968.                                   008797
C                                                                         008798
C     SUBROUTINE ARGUMENTS (ALL INPUT)                                    008799
C  A      = MATRIX TO BE PRINTED. SIZE(NR,NC).                            008800
C  NR     = NUMBER OF ROWS IN MATRIX A.                                   008801
C  NC     = NUMBER OF COLS IN MATRIX A.                                   008802
C  ANAME  = MATRIX IDENTIFICATION. (A6 FORMAT).                           008803
C  KR     = ROW DIMENSION OF A IN CALLING PROGRAM.                        008804
C                                                                         008805
 2010 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H ) //             008806
     *       10X,10(7X,1H( I2,1H))/)                                      008807
 2020 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H )                008808
     *        3X, 9HCONTINUED //10X,10(7X,1H( I2,1H))/)                   008809
 2030 FORMAT (1X,2I5,2X,1P10D11.3)                                        008810
 2040 FORMAT (14HOEND OF WRITE.)                                          008811
C                                                                         008812
C  PULL UP A NEW PAGE FOR MATRIX AND PRINT MATRIX NAME.                   008813
      CALL PAGEHD                                                         008814
      WRITE (NOT,2010) ANAME,NR,NC,(L,L=1,10)                             008815
      NLINE = 0                                                           008816
C                                                                         008817
      DO 60 I=1,NP                                                        008818
      NZERO = 0                                                           008819
      JS = 1                                                              008820
   10 JE = JS+9                                                           008821
      IF (JE .GT. NC) JE=NC                                               008822
C  SEE IF ELEMENTS ARE ZERO.                                              008823
      DO 20 J=JS,JE                                                       008824
   20 IF (A(I,J) .NE. 0.D 0) GO TO 30                                     008825
      GO TO 40                                                            008826
   30 NLINE = NLINE+1                                                     008827
      IF (NLINE .LE. 44) GO TO 35                                         008828
      CALL PAGEHD                                                         008829
      WRITE (NOT,2020) ANAME,NR,NC,(L,L=1,10)                             008830
      NLINE = 1                                                           008831
   35 WRITE (NOT,2030) I,JS,(A(I,J), J=JS,JE)                             008832
      NZERO = 1                                                           008833
   40 IF (JE .EQ. NC) GO TO 50                                            008834
```

```
      JS = JS+10                                                        008835
      GO TO 10                                                          008836
C   SKIP A SPACE BETWEEN EACH ROW IF THERE ARE MORE THAN 10 COLUMNS     008837
C   AND SOMETHING HAS BEEN WRITTEN.                                     008838
   50 IF (NC.LE.10 .OR. NZERO.EQ.0 .OR. I.EQ.NR)  GO TO 60              008839
      NLINE = NLINE+1                                                   008840
      WRITE (NOT,2030)                                                  008841
   60 CONTINUE                                                          008842
C                                                                       008843
      WRITE (NOT,2040)                                                  008844
      RETURN                                                            008845
      END                                                               008846
```

```
[HDG,P    WRITES                                                        -008847
[FOR,IS   WRITES                                                        -008848
      COMPILER (XM=1),(EQUIV=CMN)                                       -008849
      SUBROUTINE WRITES (A,NR,NC,KR)                                     008850
      DOUBLE PRECISION A                                                -008851
      DIMENSION A(KR,1),ICHEAD(10)                                       008852
      DATA NOT / 6/                                                      008853
      DATA ICHEAD/4H( 1),4H( 2),4H( 3),4H( 4),4H( 5),                    008854
     *           4H( 6),4H( 7),4H( 8),4H( 9),4H(10)  /                  008855
C                                                                       008856
 2010 FORMAT ( 8X,10(7X,A4))                                            008857
 2030 FORMAT (1X,2I5,2X,1P10D11.3)                                      008858
C                                                                       008859
      LR = 10                                                           008860
      IF (NC .LT. LR) LR = NC                                           008861
      WRITE (NOT,2010)  (ICHEAD(L),L=1,LR)                              008862
      DO 60 I=1,NR                                                      008863
      JS = 1                                                            008864
   10 JE = JS + 9                                                       008865
      IF (JE .GT. NC) JE = NC                                           008866
      WRITE (NOT,2030) I,JS, (A(I,J),J=JS,JE)                           008867
      IF (JE .EQ. NC) GO TO 60                                          008868
      JS = JS + 10                                                      008869
      GO TO 10                                                          008870
   60 CONTINUE                                                          008871
C                                                                       008872
      RETURN                                                            008873
      END                                                               008874
```

```
[HDG,P    WRITIS                                                        -008875
[FOR,IS   WRITIS                                                        -008876
      COMPILER (XM=1),(EQUIV=CMN)                                       -008877
      SUBROUTINE WRITIS (IM,NR,NC,KR)                                    008878
      DIMENSION IM(KR,1), ICH(20)                                        008879
      DATA NOT / 6/                                                      008880
      DATA ICH /                                                         008881
     *  4H( 1),4H( 2),4H( 3),4H( 4),4H( 5),4H( 6),4H( 7),4H( 8),         008882
     *    4H( 9),4H(10),                                                 008883
     *  4H(11),4H(12),4H(13),4H(14),4H(15),4H(16),4H(17),4H(18),         008884
     *    4H(19),4H(20)  /                                               008885
C                                                                       008886
 2001 FORMAT (17X20(1X,A4))                                             008887
 2002 FORMAT (1X,2I5,5X,20I5)                                           008888
C                                                                       008889
      LR = 20                                                           008890
      IF (NC .LT. LR) LR = NC                                           008891
      WRITE (NOT,2001) (ICH(L),L=1,LR)                                  008892
      DO 60 I=1,NR                                                      008893
      JS = 1                                                            008894
   10 JE = JS + 19                                                      008895
      IF (JE .GT. NC) JE = NC                                           008896
      WRITE (NOT,2002) I,JS, (IM(I,J),J=JS,JE)                          008897
      IF (JE .EQ. NC) GO TO 60                                          008898
      JS = JS + 20                                                      008899
      GO TO 10                                                          008900
   60 CONTINUE                                                          008901
C                                                                       008902
      RETURN                                                            008903
      END                                                               008904
```

```
[HDG,P     YDOT                                                          -008905
[FOR,IS    YDOT                                                          -008906
      COMPILER (XM=1),(EQUIV=CMN)                                        -008907
      SUBROUTINE YDOT                                                     008908
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -008909
C                                                                         008910
            COMMON /AMUBW /                                               008911
     *      AMU(15,15, 5),BW(30, 65)                                     108912
            COMMON /BHBSRD/                                               008913
     *      BH(6,12, 9),BS(6,12,10),ROL(3,3, 5),DOL(3, 5)               208914
            COMMON /HANDS /                                               008915
     *      HATH(3, 6, 8),SIGH(3, 6, 8),HATS(3, 6,10),SIGS(3, 6,10)     408916
            COMMON /ILINER/                                               008917
     *      IFLNER                                                        008918
            COMMON /INTGRL/                                               008919
     *      AM( 78, 5),ACOF(9, 6, 5),BCOF(6, 6, 5),                     508920
     *      COF11( 6, 6, 5),COF22( 6, 6, 5),COF33( 6, 6, 5),AK( 6, 6, 5), 608921
     *      COF12( 6, 6, 5),COF13( 6, 6, 5),COF23( 6, 6, 5),AD( 6, 6, 5), 708922
     *      COFXY( 6, 6, 5),COFXZ( 6, 6, 5),COFYZ( 6, 6, 5)              808923
            COMMON /IVCONS/                                               008924
     *      IV(6, 5)                                                      908925
            COMMON /JILFLG/                                              -008926
     *      JIL                                                          -008927
            COMMON /LAMBDA/                                               008928
     *      ALAM(30)                                                     1008929
            COMMON /MAXMUM/                                               008930
     *      NBMAX,NHMAX,NSPMAX,NMWMAX,NMWBOD,NMDBOD,KMU,KY,KU            008931
            COMMON /MOMENG/                                               008932
     *      P( 65),PMOM(30),HTOT(3),TOTL(3),ENGKE( 5),ENGPE( 5),         1108933
     *      TOTKE, TOTPE, TOTENG, AHTOT,ATOTL                            008934
            COMMON /NUMBRS/                                               008935
     *      ZRO,ONE,TWO,TRES                                             008936
            COMMON /SPECIF/                                               008937
     *      BETAH(6, 5),BETAHD(6, 5),AMO(2, 5),RH(3,3,24),RS(3,3,20),    1608938
     *      DH(3,28),DS(3,20),IMO(3, 5),NMOW(5, 5),IFTSMW(10),           1708939
     *      NB,NH,NSPT,NOFMO,NDELTA,ITOPOL(2, 5),IRGFLX( 5),IHDATA(7, 5), 1808940
     *      LOCU(12),LENU(12),NU,NBETA,NLAM,NEQ                          1908941
            COMMON /TAPENO/                                               008942
     *      NTAPE1,NTAPE2,NTAPE3                                         008943
            COMMON /TIMESS/                                               008944
     *      STARTT,DELTAT,T,ENDT,TMST                                    008945
            COMMON /VECTOR/                                               008946
     *      Y(250),YDT(250)                                              2008947
C                                                                         008948
      DIMENSION GGV( 65),D(30),V(30),BDTQ(6,12),BDTP(6,12)               7508949
C                                                                         008950
      DIMENSION BMB(30,30),BM(6,15, 9)                                   7608951
      EQUIVALENCE (BW(1),BMB(1)),  (BW( 901),BM(1))                      7708952
      EQUIVALENCE (ALAM(1),V(1))                                          008953
C                                                                         008954
```

```
      DATA IFLAG / 1 /                                                  008955
C                                                                       008956
      IF(JIL .EQ. 4) IFLAG = 1                                         -008957
C                                                                       008958
      CALL ROTDH                                                        008959
      CALL BHGENR                                                       008960
      CALL ROTDS                                                        008961
      CALL BSGENR                                                       008962
      CALL MGEN                                                         008963
      IF (NLAM .GT. 0) CALL FINDU (IFLAG)                               008964
      DO 10 L=1,NP                                                      008965
      LO = LOCU(L)                                                      008966
      LE = LENU(L)                                                      008967
   10 CALL MULT3  (AMU(1,1,L),Y(LO),P(LO),LE,LE,1,KMU,1,1)              008968
      DO 11 I=1,NU                                                      008969
   11 GGV(I) = ZRO                                                      008970
      CALL GRVGRD (GGV)                                                 008971
      DO 15 L=1,NB                                                      008972
      LE = LENU(L)                                                      008973
   15 CALL INVINP (AMU(1,1,L),AMU(1,1,L),LE,KMU)                        008974
      IF (NLAM .EQ. 0) GO TO 200                                        008975
      CALL GETBMB                                                       008976
      KBMB = 6*NHMAX                                                    008977
      CALL DCOM2  (BMB,D,NLAM,KBMB)                                     008978
C                                                                       008979
C                                                                       008980
CC  CALCULATE BETADT AND PLACE INTO YDT                                 008981
  200 IC = LOCU(2*NB+1) - 1                                             008982
      DO 60 L=1,NH                                                      008983
      DO 60 I=1,6                                                       008984
      IP1 = I + 1                                                       008985
      IF (IHDATA(IP1,L) .EQ. 1) GO TO 60                                008986
      IC = IC + 1                                                       008987
      YDT(IC) = ZRO                                                     008988
      IF (L .EQ. 1) GO TO 61                                            008989
      NOBQ = ITOPOL(1,L)                                                008990
      NOBP = ITOPOL(2,L)                                                008991
      LQ = 2*L - 2                                                      008992
      LP = LQ + 1                                                       008993
      LOQ = LOCU(NOBQ) - 1                                              008994
      LOP = LOCU(NOBP) - 1                                              008995
      LEQ = IRGFLX(NOBQ) + 6                                            008996
      LEP = IRGFLX(NOBP) + 6                                            008997
      DO 62 J=1,LEQ                                                     008998
      LOQJ = LOQ + J                                                    008999
   62 YDT(IC) = YDT(IC) + BH(I,J,LQ)*Y(LOQJ)                            009000
      DO 63 J=1,LEP                                                     009001
      LOPJ = LOP + J                                                    009002
   63 YDT(IC) = YDT(IC) + BH(I,J,LP)*Y(LOPJ)                            009003
      BETAHD(I,L) = YDT(IC)                                             009004
```

```
      GO TO 60                                                          009005
   61 LEQ = IRGFLX(1) + 6                                               009006
      DO 64 J=1,LEQ                                                     009007
   64 YDT(IC) = YDT(IC) + BH(I,J,1)*Y(J)                                009008
      BETAHD(I,L) = YDT(IC)                                             009009
   60 CONTINUE                                                          009010
C                                                                       009011
C  PUT MODAL VELOCITIES (XE(DOT)) INTO YDOT                             009012
C                                                                       009013
      DO 65 N=1,NB                                                      009014
      LE = IRGFLX(N)                                                    009015
      IF (LE .EQ. 0) GO TO 65                                           009016
      LOU = LOCU(N) + 5                                                 009017
      LO = LOCU(N+NB) - 1                                               009018
      DO 66 J=1,LE                                                      009019
   66 YDT(LO+J) = Y(LOU+J)                                              009020
   65 CONTINUE                                                          009021
C                                                                       009022
      CALL TORQUE (GGV)                                                 009023
C                                                                       009024
CC  INITIALIZE UDOT, GET RHS OF LAMBDA EQUATION                         009025
C                                                                       009026
      DO 70 N=1,NB                                                      009027
      LO = LOCU(N)                                                      009028
      LE = LENU(N)                                                      009029
   70 CALL MULT3  (AMU(1,1,N),GGV(LO),YDT(LO),LE,LE,1,KMU,1,1)          009030
C                                                                       009031
      IFLAG = 2                                                         009032
      IF (NLAM .EQ. 0) RETURN                                           009033
C                                                                       009034
      DO 80 L=1,NH                                                      009035
      CALL BDOTQP (L,BDTQ,BDTP)                                         009036
      DO 80 I=1,6                                                       009037
      IP1 = I + 1                                                       009038
      IC = IV(I,L)                                                      009039
      IF (IC .EQ. 0) GO TO 80                                           009040
      V(IC) = ZRO                                                       009041
      I1C = 6*(L-1) + I                                                 009042
      IF (IHDATA(IP1,L) .EQ. 2) V(IC) = ADDT(I1C,T)                     009043
      IF (L .EQ. 1) GO TO 81                                            009044
      NOBQ = ITOPOL(1,L)                                                009045
      NOBP = ITOPOL(2,L)                                                009046
      LQ = 2*L - 2                                                      009047
      LP = LQ + 1                                                       009048
      LOQ = LOCU(NOBQ) - 1                                              009049
      LOP = LOCU(NOBP) - 1                                              009050
      LEQ = IRGFLX(NOBQ) + 6                                            009051
      LEP = IRGFLX(NOBP) + 6                                            009052
      DO 82 J=1,LEQ                                                     009053
      LOQJ = LOQ + J                                                    009054
```

```
   82 V(IC) = V(IC) - BH(I,J,LQ)*YDT(LOQJ) - BDTQ(I,J)*Y(LOQJ)          009055
      DO 83 J=1,LEP                                                     009056
      LOPJ = LOP + J                                                    009057
   83 V(IC) = V(IC) - BH(I,J,LP)*YDT(LOPJ) - BDTP(I,J)*Y(LOPJ)          009058
      GO TO 80                                                          009059
   81 LEQ = IRGFLX(I) + 6                                               009060
      DO 84 J=1,LEQ                                                     009061
   84 V(IC) = V(IC) - BH(I,J,1)*YDT(J) - BDTQ(I,J)*Y(J)                 009062
   80 CONTINUE                                                          009063
C                                                                       009064
      IF (NLAM .GT. 1) GO TO 305                                        009065
      V(1) = V(1)/D(1)                                                  009066
      GO TO 310                                                         009067
  305 CALL BAKSLV (BMB,NLAM,V,D,KBMB)                                   009068
C                                                                       009069
  310 LEQ = LENU(1)                                                     009070
      DO 85 I=1,6                                                       009071
      ILN = IV(I,1)                                                     009072
      IF (ILN .EQ. 0) GO TO 85                                          009073
      F = V(ILN)                                                        009074
      DO 86 J=1,LEQ                                                     009075
   86 YDT(J) = YDT(J) + F*BM(I,J,1)                                     009076
   85 CONTINUE                                                          009077
C                                                                       009078
      DO 90 L=2,NH                                                      009079
      NOBQ = ITOPOL(1,L)                                                009080
      NOBP = ITOPOL(2,L)                                                009081
      LQ = 2*L - 2                                                      009082
      LP = LQ + 1                                                       009083
      LOQ = LOCU(NOBQ) - 1                                              009084
      LOP = LOCU(NOBP) - 1                                              009085
      LEQ = LENU(NOBQ)                                                  009086
      LEP = LENU(NOBP)                                                  009087
      DO 90 I=1,6                                                       009088
      ILN = IV(I,L)                                                     009089
      IF (ILN .EQ. 0) GO TO 90                                          009090
      F = V(ILN)                                                        009091
      DO 95 J=1,LEQ                                                     009092
      LOQJ = LOQ + J                                                    009093
   95 YDT(LOQJ) = YDT(LOQJ) + F*BM(I,J,LQ)                              009094
      DO 96 J=1,LEP                                                     009095
      LOPJ = LOP + J                                                    009096
   96 YDT(LOPJ) = YDT(LOPJ) + F*BM(I,J,LP)                              009097
   90 CONTINUE                                                          009098
C                                                                       009099
      RETURN                                                            009100
      END                                                               009101
```

```
[HDG,P    YDOTL                                                          -009102
[FOR,IS   YDOTL                                                          -009103
      COMPILER (XM=1),(EQUIV=CMN)                                        -009104
      SUBROUTINE YDOTL (A,B,Y,YD,NY,KA)                                   009105
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                 -009106
C                                                                         009107
C  SUBROUTINE FORMS THE LINEARIZED YDOT VECTOR.                           009108
C                                                                         009109
C              -----SUBROUTINE ARGUMENT DESCRIPTIONS-----                 009110
C                                                                         009111
C  A       = INPUT  LINEARIZED COEFFICIENTS.  SIZE NA BY NY.              009112
C                   USED IN EXPRESSION YD = A * Y + B                     009113
C  B       = INPUT  EXTERNAL FORCING TORQUES.  USER SUPPLIED VIA          009114
C                   SUBROUTINE LTORQL.                                    009115
C  Y       = INPUT VECTOR OF STATE VARIABLES.                             009116
C  YD      = OUTPUT VECTOR OF STATE VECTOR TIME DERIVATIVES.              009117
C  NY      = SIZE OF STATE VECTOR TO BE INTEGRATED.                       009118
C                                                                         009119
      DIMENSION A(KA,1),B(1),Y(1),YD(1)                                  009120
C                                                                         009121
      DO 10 I=1,NY                                                       009122
      YD(I) = B(I)                                                       009123
      DO 10 J=1,NY                                                       009124
      YD(I) = YD(I) + A(I,J) * Y(J)                                      009125
   10 CONTINUE                                                           009126
      RETURN                                                             009127
      END                                                                009128
```

```
[HDG,P    ZERO                                                          -009129
[FOR,IS   ZERO                                                          -009130
      COMPILER (XM=1),(EQUIV=CMN)                                       -009131
      SUBROUTINE ZERO  (Z,NR,NC,KR)                                      009132
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)                                -009133
      DIMENSION  Z(KR,1)                                                 009134
C                                                                        009135
C  GENERATE A MATRIX OF ZEROES.                                          009136
C  CODED BY  RL WOHLEN.  FEB 1965.                                       009137
C                                                                        009138
C      SUBROUTINE ARGUMENTS                                              009139
C  Z   = OUTPUT MATRIX GENERATED. SIZE(NR,NC).                           009140
C  NR  = INPUT  NUMBER OF ROWS IN MATRIX Z.                              009141
C  NC  = INPUT  NUMBER OF COLS IN MATRIX Z.                              009142
C  KR  = INPUT  ROW DIMENSION OF MATRIX Z IN CALLING PROGRAM.            009143
C                                                                        009144
      DO 10 I=1,NR                                                       009145
      DO 10 J=1,NC                                                       009146
   10 Z(I,J) = 0.D 0                                                     009147
      RETURN                                                             009148
      END                                                                009149
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

**********      CDCDICD  //// END OF LIST ////

**********      CDCDICD  //// END OF LIST ////
```