Final Report

for

SPACE SHUTTLE PROPULSION ESTIMATION DEVELOPMENT VERIFICATION

Volume II

by

Robert M. Rogers

Prepared under:

Contract NAS8-36152

for

GEORGE C. MARSHALL SPACE FLIGHT CENTER

Marshall Space Flight Center, Alabama 35812

28 February 1989

Abstract

The work described in this report details the application of extended Kalman filtering to estimating the Space Shuttle Propulsion performance, i.e. specific impulse, from flight data in a post-flight processing computer program. The flight data used includes inertial platform acceleration, SRB head pressure, SSME chamber pressure and flow rates, and ground based radar tracking data. The key feature in this application is the model used for the SRB's, which is a nominal or reference quasi-static internal ballistics model normalized to the propellant burn depth. Dynamic states of mass overboard and propellant burn depth are included in the filter model to account for real-time deviations from the reference model used. Aerodynamic, plume, wind and main engine uncertainties are also included for an integrated system model. Assuming uncertainty within the propulsion system model and attempts to estimate its deviations represent a new application of parameter estimation for rocket powered vehicles. Illustrations from the results of applying this estimation approach to several missions show good quality propulsion estimates.

Table of Contents

```
      PROGRAM MERGE
C
C   MERGES DATA FILES GENERATED FROM IMUTST,PREPRCS,AND REDRDR
C
       DIMENSION DIMU(9),DPROL(3,6),DPROS(2),DRDR(15)
C
CSTG2 DATA IMAX, IBEGIN, IEND / 500, 130, 500 /
      DATA IMAX, IBEGIN, IEND / 130, 1, 130 /
C
      OPEN( UNIT=1,FILE='VEHROT',STATUS='OLD',ACCESS='SEQUENTIAL')
      OPEN( UNIT=2,FILE='PREPOT',STATUS='OLD',ACCESS='SEQUENTIAL')
      OPEN( UNIT=3,FILE='TRACK',STATUS='OLD',ACCESS='SEQUENTIAL')
      OPEN( UNIT=4,FILE='REALOUT',STATUS='NEW',ACCESS='SEQUENTIAL')
C
      DO 100 ITIME = 1, IMAX
C
      READ(1,901,END=100) TIME
      READ(1,906) (DIMU(I), I = 1, 3)
      READ(1,906) (DIMU(I), I = 4, 6)
      READ(1,906) (DIMU(I), I = 7, 9)
      READ(2,902) PTIME,((DPROL(I,J),I=1,3),J=1,6),(DPROS(K),K=1,2)
      READ(3,906) RTIME
      READ(3,906) (DRDR(I), I = 1, 15)
C
      IF ( (ITIME.LT.IBEGIN).OR.(ITIME.GT.IEND) ) GO TO 100
C
      WRITE(4,904) (DIMU(I), I = 1, 3)
      WRITE(4,905) ((DPROL(I,J), J = 1, 6), I = 1, 3)
      WRITE(4,906) (DRDR(I), I = 1, 9)
      WRITE(4,907) (DPROS(I), I = 1, 2)
C
100   CONTINUE
C
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
      CLOSE(UNIT=4)
C
      STOP
C
901   FORMAT(7E15.8)
902   FORMAT( 3(7E15.8,/) )
903   FORMAT(F5.1,9E14.7)
904   FORMAT(5X, 7E15.8)
905   FORMAT(5X, 6E15.8)
906   FORMAT(5X, 3E15.8)
907   FORMAT(5X, E15.8)
C
      END
```

```
C     PROGRAM VEHREF
      PROGRAM MAIN
C
      DIMENSION STALE(39),VAL(39)
     1 ,SPTIME(600),TSHFT(600),GRAPH(600,39),PVAR(600),VEC(600)
      DIMENSION TMP1(3,3),TMP2(3,3),TMP3(3,3),TMP4(3,3),TMP5(3,3)
     1 ,REFMT1(3,3),REFMT2(3,3),REFMT3(3,3),DV1(3),DV2(3),DV3(3)
     2 ,VOLD1(3),VOLD2(3),VOLD3(3),CIBRI(3,3),VECX(3)
     3, CBI1(3,3),CBI2(3,3),CBI3(3,3),C50BR1(3,3),C50BR2(3,3)
     4, TMP6(3,3),C50BR3(3,3),TBBRI0(3,3),UNIT(3,3)
     5, THT1(3),THT2(3),THT3(3),THTM(3),DVM(3),OMEG(3),C50BR(3,3)
     6, Q(4),TMP7(4,4),DERQ(4),PREQ(4),CORQ(4)
C
      INTEGER*4 IPACK(30),IW(1440),ISTAT(39),ERROR,IDBN(4)
      REAL*8 TIME, TSTART, TSTOP
      CHARACTER MSID*12(39),CPACK*8,DBNAME*17,CHARP*4(30),YLAB*12
      EQUIVALENCE (IPACK(2),CPACK),(DBNAME,IDBN),(IPACK,CHARP)
C
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2, CRAD
      DATA IPACK(8),IPACK(9),IPACK(10),IPACK(11),IPACK(12),IPACK(13)
     1 / 88,273,15,37,00,000 /
CVAX  DATA IPACK(2),IPACK(3),IPACK(4),IPACK(5),IPACK(6),IPACK(7)
C61C 1 / 'STS6','1CDB','.DAT','    ','    ','    ' /
C61A 1 / 'STS6','1ADB','.DAT','    ','    ','    ' /
C61B 1 / 'STS6','1BDB','.DAT','    ','    ','    ' /
C51B 1 / 'STS5','1BDB','.DAT','    ','    ','    ' /
CVAX  DATA IPACK(8),IPACK(9),IPACK(10),IPACK(11),IPACK(12),IPACK(13)
C61C 1 / 86,012,11,54,59,997 /
C61A 1 / 85,303,17,00,00,010 /
C61B 1 / 85,331,00,29,00,006 /
C51B 1 / 85,119,16,02,17,989 /
      DATA IPACK(14)/25/,IPACK(15)/2/,IPACK(16)/1440/
C
      DATA IST / 13 /
      DATA VOLD1,VOLD2,VOLD3 / 3*0.0,3*0.0,3*0.0 /
      DATA TBBRI0 / 2*0.0,-1.0,0.0,1.0,0.0,1.0,2*0.0 /
      DATA Q / 0.707106, 0.0, 0.707106, 0.0 /
C
C51B  DATA REFSMT1/-.694542,.496024,-.521125,-.301635,.45684,.836847
C51B 1 ,.653167,.738415,-.167677/
C51B  DATA REFSMT2/-.120625,-.078044,.989626,.779794,-.624357,.045811
C51B 1 ,.614305,.77723,.136171/
C51B  DATA REFSMT3/.517692,-.716205,-.468022,-.244835,.400140,-.883144
C51B 1 ,.817987,.571785,.0317968/
C
C61A  DATA REFSMT1/.584975,-.342753,.735068,.487126,-.576157,-.656315
C61A 1 ,.648469,.741999,-.170073/
C61A  DATA REFSMT2/.339075,-.0968083,-.935765,-.712631,.622931,-.322666
C61A 1 ,.614154,.776263,.142232/
C61A  DATA REFSMT3/-.561507,.802985,.199810,.0904577,-.180458,.979414
C61A 1 ,.822512,.568022,.0286922/
C
C61B  DATA REFSMT1/-.756027,.641557,-.129719,.0119258,.211652,.977272
C61B 1 ,.654431,.737297,-.167666/
C61B  DATA REFSMT2/.212771,-.327267,.920665,.760344,-.536331,-.366369
```

```
C61B 1 ,.613682,.777975,.134720/
C61B  DATA REFSMT3/.371235,-.485959,-.791219,-.436979,.660420,-.610651
C61B 1 ,.819288,.572441,.0328175/
C
C61CS DATA REFSMT1/.6134,-.3773,.6938,.4504,-.5546,-.6997
C61CS1 ,.6488,.7417,-.1702/
C61CS DATA REFSMT2/.2954,-.05881,-.9535,-.7317,.6278,-.2654
C61CS1 ,.6142,.7762,.1424/
C61CS DATA REFSMT3/-.5544,.7906,.2599,.1251,-.2296,.9652
C61CS1 ,.8228,.5676,.02839/
C61C  DATA REFSMT1/.6133972,-.3773153,.6938130,.4503511,-.5545661
C61C 1              ,-.6997428,.6487887,.741680,-.1702449/
C61C  DATA REFSMT2/.2954440,-.05880850,-.9535483,-.7317411,.6277722
C61C 1              ,-.2654373,.6142209,.7761723,.1424394/
C61C  DATA REFSMT3/-.5544043,.7906296,.2598850,.1250687,-.2295741
C61C 1            ,.9652219,.8227961,.5676264,.02839340/
C
      DATA REFMT1/.7124555,-.6688998,-.2120820,-.2697270,.0179653
     1 ,-.9627689,.6478066,.7431343,-.1676210/
      DATA REFMT2/-.4540111,.4971972,-.7393705,-.6433136,.3912134
     1 ,.6581028,.6164586,.7744331,.1422388/
      DATA REFMT3/-.1936510,.2368643,.9520475,.5359492,-.7872760
     1 ,.3048846,.8217406,.5692902,.0255092/
C
      DATA TSTART,TSTOP,SRATE / -10.0D0, 525.0D0, 1.0 /
      DATA JUMP / 0 /
      DATA DVMAX,THTMAX,TMAX / 15.0, 180., 525.0 /
C
      OPEN( UNIT=1,FILE='VEHRIN',STATUS='OLD',ACCESS='SEQUENTIAL' )
      OPEN( UNIT=3,FILE='VEHRPL',STATUS='NEW',ACCESS='SEQUENTIAL' )
      OPEN( UNIT=9,FILE='VEHROT',STATUS='NEW',ACCESS='SEQUENTIAL' )
C
      IPACK(1) = -1
      CPACK = 'ASTS317'
      DBNAME ='STS26DB'
      CRAD = 57.295779
C
      DO 10 I = 1, 4
   10 IPACK(I+3) = IDBN(I)
C
      WRITE(6, 12) IPACK(1),CPACK,DBNAME, (IPACK(J), J = 8, 14)
   12 FORMAT( 2X, I2, 2X, A8,A17, 2X, 7I4 )
C
      READ(1, 20) (MSID(J), J=1, (IPACK(14)+10))
   20 FORMAT( A12 )
      CLOSE ( UNIT=1 )
C
      WRITE(6, 25) (MSID(J), J=1, (IPACK(14)+10))
   25 FORMAT( 2X, A12 )
C
      I=0
      CALL ACCREQ(2,IPACK(2),DBNAME,IPACK(14),MSID,NUMBR,ISTAT,ERROR)
      IF ( ERROR.NE.0 ) THEN
      WRITE(6,501) ERROR
  501 FORMAT('ERROR IN ACCREQ - ERROR=', I5)
```

```
      STOP
      END IF
C
   50 CONTINUE
C
      I = I + 1
      TIME = TSTART
C
      CALL ACCESS( IPACK,IW,MSID,SRATE,TIME,TSTOP,STALE,VAL,ISTAT )
C
      IF( IPACK(1).EQ.6 ) GO TO 200
C
      SPTIME(I)=SNGL( TIME ) - 2.0
      SPTMAX = SNGL( TSTOP )
      IF ( SPTIME(I).GE.SPTMAX ) GO TO 200
C
      IF ( SPTIME(I).GT.125. ) DVMAX = 0.15
C
      IF ( VAL(1).NE.0.0 ) THEN
      PITCH=CRAD*VAL(1)
      YAW=CRAD*VAL(2)
      ROLI=CRAD*VAL(3)
      ROLO=CRAD*VAL(4)
C
      CALL TMATY ( -YAW, TMP1 )
      CALL TMATR ( -ROLI, TMP2 )
      CALL MULT ( TMP2, TMP1, TMP3, 3, 3, 3 )
      CALL TMATP ( -PITCH, TMP1 )
      CALL MULT ( TMP1, TMP3, TMP2, 3, 3, 3 )
      CALL TMATR ( -ROLO, TMP1 )
      CALL MULT ( TMP1, TMP2, TMP3, 3, 3, 3 )
      CALL TMATP ( -10.6, TMP2 )
      CALL MULT ( TMP2, TMP3, TMP4, 3, 3, 3 )
      ELSE
      END IF
C
      IF ( VAL(8).NE.0.0 ) THEN
      PITCH = CRAD*VAL(8)
      YAW = CRAD*VAL(9)
      ROLI = CRAD*VAL(10)
      ROLO = CRAD*VAL(11)
C
      CALL TMATY ( -YAW, TMP1 )
      CALL TMATR ( -ROLI, TMP2 )
      CALL MULT ( TMP2, TMP1, TMP3, 3, 3, 3 )
      CALL TMATP ( -PITCH, TMP1 )
      CALL MULT ( TMP1, TMP3, TMP2, 3, 3, 3 )
      CALL TMATR ( -ROLO, TMP1 )
      CALL MULT ( TMP1, TMP2, TMP3, 3, 3, 3 )
      CALL TMATP ( -10.6, TMP1 )
      CALL MULT ( TMP1, TMP3, TMP5, 3, 3, 3 )
      ELSE
      END IF
C
      IF ( VAL(15).NE.0.0 ) THEN
```

```
      PITCH = CRAD*VAL(15)
      YAW = CRAD*VAL(16)
      ROLI = CRAD*VAL(17)
      ROLO = CRAD*VAL(18)
C
      CALL TMATY ( - YAW, TMP1 )
      CALL TMATR ( - ROLI, TMP2 )
      CALL MULT ( TMP2, TMP1, TMP3, 3, 3, 3 )
      CALL TMATP ( - PITCH, TMP1 )
      CALL MULT ( TMP1, TMP3, TMP2, 3, 3, 3 )
      CALL TMATR ( - ROLO, TMP1 )
      CALL MULT ( TMP1, TMP2, TMP3, 3, 3, 3 )
      CALL TMATP ( -10.6, TMP1 )
      CALL MULT ( TMP1, TMP3, TMP6, 3, 3, 3 )
C
      ELSE
      END IF
C
      IF ( JUMP.LT.1 ) THEN
C
      CALL MULT ( TBBRI0, TMP4, C50BR1, 3, 3, 3 )
      CALL MULT ( TBBRI0, TMP5, C50BR2, 3, 3, 3 )
      CALL MULT ( TBBRI0, TMP6, C50BR3, 3, 3, 3 )
      JUMP=1
C
      WRITE(6, 905) ((C50BR1(J,K), J = 1, 3), K = 1, 3)
      WRITE(6, 905) ((C50BR2(J,K), J = 1, 3), K = 1, 3)
      WRITE(6, 905) ((C50BR3(J,K), J = 1, 3), K = 1, 3)
C
      CALL MULT ( C50BR1, REFMT1, C50BR, 3, 3, 3 )
      CALL MULT ( C50BR2, REFMT2, TMP2, 3, 3, 3 )
      CALL MULT ( C50BR3, REFMT3, TMP3, 3, 3, 3 )
C
      WRITE(6, 999)
      WRITE(6, 905) ((C50BR(II,JJ), JJ = 1, 3), II = 1, 3)
      WRITE(6, 999)
      WRITE(6, 905) ((TMP2(II,JJ), JJ = 1, 3), II = 1, 3)
      WRITE(6, 999)
      WRITE(6, 905) ((TMP3(II,JJ), JJ = 1, 3), II = 1, 3)
      WRITE(6, 999)
C
      ELSE
      END IF
C
      IF ( VAL(1).NE.0.0 ) THEN
      CALL TRANS ( TMP4, TMP3, 3, 3 )
      CALL MULT ( C50BR1, TMP3, CBI1, 3, 3, 3 )
      IF( ABS(CBI1(3,1)).GT.1.0 ) CBI1(3,1)=SIGN(1.0, CBI1(3,1))
      THT1(2) = CRAD*ASIN( -CBI1(3,1) )
      THT1(1) = CRAD*ATAN2( CBI1(3,2), CBI1(3,3) )
      THT1(3) = CRAD*ATAN2( CBI1(2,1), CBI1(1,1) )
C
      DV1(1) = VAL(5) - VOLD1(1)
      DV1(2) = VAL(6) - VOLD1(2)
      DV1(3) = VAL(7) - VOLD1(3)
```

```
C
      CALL MULT ( REFMT1, DV1, VECX, 3, 3, 1 )
      CALL MULT ( C50BR1, VECX, DV1, 3, 3, 1 )
      ELSE
      END IF
C
      IF ( VAL(8).NE.0.0 ) THEN
      CALL TRANS ( TMP5, TMP3, 3, 3 )
      CALL MULT ( C50BR2, TMP3, CBI2, 3, 3, 3 )
      IF ( ABS(CBI2(3,1)).GT.1.0 ) CBI2(3,1) = SIGN(1.0, CBI2(3,1))
      THT2(2) = CRAD*ASIN( -CBI2(3,1) )
      THT2(1) = CRAD*ATAN2( CBI2(3,2), CBI2(3,3) )
      THT2(3) = CRAD*ATAN2( CBI2(2,1), CBI2(1,1) )
C
      DV2(1) = VAL(12) - VOLD2(1)
      DV2(2) = VAL(13) - VOLD2(2)
      DV2(3) = VAL(14) - VOLD2(3)
C
      CALL MULT ( REFMT2, DV2, VECX, 3, 3, 1 )
      CALL MULT ( C50BR2, VECX, DV2, 3, 3, 1 )
      ELSE
      END IF
C
      IF ( VAL(15).NE.0.0 ) THEN
      CALL TRANS ( TMP6, TMP3, 3, 3 )
      CALL MULT ( C50BR3, TMP3, CBI3, 3, 3, 3 )
      IF ( ABS(CBI3(3,1)).GT.1.0 ) CBI3(3,1) = SIGN(1.0,CBI3(3,1))
      THT3(2) = CRAD*ASIN( -CBI3(3,1) )
      IF ( CBI3(3,3).NE.0.0 ) THEN
      THT3(1) = CRAD*ATAN2( CBI3(3,2), CBI3(3,3) )
      ELSE
      THT3(1) = 0.0
      END IF
      IF ( CBI3(1,1).NE.0.0 ) THEN
      THT3(3) = CRAD*ATAN2( CBI3(2,1), CBI3(1,1) )
      ELSE
      THT3(3) = 0.0
      END IF
C
      DV3(1) = VAL(19) - VOLD3(1)
      DV3(2) = VAL(20) - VOLD3(2)
      DV3(3) = VAL(21) - VOLD3(3)
C
      CALL MULT ( REFMT3, DV3, VOLD3, 3, 3, 1 )
      CALL MULT ( C50BR3, VOLD3, DV3, 3, 3, 1 )
      ELSE
      END IF
C
      IF ( VAL(8).EQ.0.0 ) THEN
C
      DO 75 II = 1, 3
      THTM(II) = THT1(II)
      DVM(II) = DV1(II)
75    CONTINUE
C
```

```
      ELSE
      END IF
C
      DO 100 II = 1, 3
C
      IF ( ABS(THT1(II) - THT2(II)).GT.THTMAX ) THEN
C
         THTM(II) = THTM(II)
C
      WRITE(*, 940) II, SPTIME(I)
C
      ELSE
C
         THTM(II) = 0.5*( THT1(II) + THT2(II) )
C
      END IF
C
      IF ( ABS(DV1(II) - DV2(II)).GT.DVMAX ) THEN
C
         DVM(II) = DVM(II)
C
         WRITE(*, 950) II, SPTIME(I)
C
      ELSE
C
         DVM(II) = ( DV1(II) + DV2(II) + DV3(II) )/3.0
C        DVM(II) = 0.5*( DV1(II) + DV2(II) )
C
      END IF
C
      IF ( SPTIME(I).GT.TEND ) THEN
C
         DVM(II) = ( DV1(II) + DV2(II) + DV3(II) )/3.0
C        DVM(II) = 0.5*( DV1(II) + DV2(II) )
C
      ELSE
      END IF
C
100   CONTINUE
C
      OMEG(1) = VAL(24)/CRAD
      OMEG(2) = VAL(22)/CRAD
      OMEG(3) = VAL(23)/CRAD
C
      CALL QMTRX ( OMEG, TMP7 )
C
      QS = 0.0
      DO 130 J = 1, 4
         DERQ(J) = 0.0
         DO 120 K = 1, 4
            DERQ(J) = DERQ(J) + TMP7(J,K)*Q(K)
120      CONTINUE
         PREQ(J) = Q(J) + DERQ(J)/SRATE
         QS = QS + PREQ(J)*PREQ(J)
130   CONTINUE
```

```
C
      QSM1 = QS
      QS = 0.0
      DO 170 J = 1, 4
         TEMP = 0.0
         DO 160 K = 1, 4
            TEMP = TEMP + TMP7(J,K)*PREQ(K)
160      CONTINUE
         CORQ(J) = Q(J) + 0.5*(DERQ(J) + TEMP)/SRATE
         QS = QS + CORQ(J)*CORQ(J)
170   CONTINUE
C
      DO 175 J = 1, 4
      Q(J) = CORQ(J)
175   CONTINUE
C
      CALL CBIMXQ ( Q, TMP1 )
      CALL QUAT2E ( CORQ, THTM )
C
      ALOAD = VAL(25)
C
      IF( (I-IST).LE.0 ) GO TO 180
      GRAPH((I-IST),1) = DVM(1)
      GRAPH((I-IST),2) = DVM(2)
      GRAPH((I-IST),3) = DVM(3)
      GRAPH((I-IST),4) = THTM(1)
      GRAPH((I-IST),5) = THTM(2)
      GRAPH((I-IST),6) = THTM(3)
      GRAPH((I-IST),7) = OMEG(1)*CRAD
      GRAPH((I-IST),8) = OMEG(2)*CRAD
      GRAPH((I-IST),9) = OMEG(3)*CRAD
      GRAPH((I-IST),10) = ALOAD
      TSHFT(I-IST) = SPTIME(I)
180   CONTINUE
      WRITE(6, 905) SPTIME(I)
      WRITE(6, 902) (DVM(II), II = 1, 3), ALOAD
      WRITE(6, 905) (THTM(II), II = 1, 3)
      WRITE(6, 905) (OMEG(II), II = 1, 3)
C
      IMAX = I - IST
C
      VOLD1(1) = VAL(5)
      VOLD1(2) = VAL(6)
      VOLD1(3) = VAL(7)
      VOLD2(1) = VAL(12)
      VOLD2(2) = VAL(13)
      VOLD2(3) = VAL(14)
      VOLD3(1) = VAL(19)
      VOLD3(2) = VAL(20)
      VOLD3(3) = VAL(21)
C
      THTM(1) = THTM(1)
      THTM(2) = THTM(2)
      THTM(3) = THTM(3)
      DVM(1) = DVM(1)
```

```
      DVM(2) = DVM(2)
      DVM(3) = DVM(3)
      DVMAG = SQRT(DVM(1)**2+DVM(2)**2+DVM(3)**2)
C
      GO TO 50
C
200   CONTINUE
C
      DO 220 I = 1, IMAX
      WRITE(9,905) TSHFT(I)
      WRITE(9,902) (GRAPH(I,J), J = 1, 3), GRAPH(I,10)
      WRITE(9,905) (GRAPH(I,J), J = 4, 6)
      WRITE(9,905) (GRAPH(I,J), J = 7, 9)
220   CONTINUE
C
      CLOSE ( UNIT=9 )
C
      XMIN = TSHFT(1)
      XMAX = TSHFT(IMAX)
C
      DO 300 NM = 1, 10
C
      YMIN = GRAPH(1,NM)
      YMAX = YMIN
C
      YLAB = MSID( IPACK(14)+NM )
C
      DO 250 I = 1, IMAX
      YMIN = AMIN1( GRAPH(I,NM), YMIN )
      YMAX = AMAX1( GRAPH(I,NM), YMAX )
      PVAR(I) = GRAPH(I,NM)
250   CONTINUE
C
C
      WRITE(6,25) YLAB
C
      CALL PLOT2 ( XMIN, XMAX, YMIN, YMAX )
      CALL PLOT3 ( '.', TSHFT, PVAR(1), IMAX )
      CALL PLOT4 ( 9, 'TIME  (SEC)',6,YLAB,13,'BRI IMU DATA' )
C
300   CONTINUE
C
C
      CLOSE ( UNIT = 3 )
C
      STOP
C
3     FORMAT(T2, 7I5)
901   FORMAT(3X, 4E15.8)
902   FORMAT( 5X, 4E15.8 )
903   FORMAT( 5X, I5 )
905   FORMAT(5X, 3E15.8)
940   FORMAT( 5X,I5,2X,28HATTITUDE DATA EDITED AT TIME,3E15.8 )
950   FORMAT( 5X,I5,2X,32HACCELERATION DATA EDITED AT TIME,3E15.8 )
999   FORMAT ( / )
```

END

```
C       PROGRAM PREPRCS
        PROGRAM MAIN
C
        DIMENSION STALE(32), VAL(32)
       1 , SPTIME(600), GRAPH(600,20), PVAR(600)
        DIMENSION FCV(3,2),PXM(3,2),TXM(3,2),PX(3,2),TX(3,2),WEHE(3,2)
       1 , PCM(3), Y8(3), PH(3), TH(3)
C
        INTEGER*4 IPACK(30),IW(1920),ISTAT(32),ERROR,IDBN(4)
        REAL*8 TIME, TSTART, TSTOP
        CHARACTER MSID*12(52),CPACK*8,DBNAME*17,CHARP*4(30),YLAB*12
        EQUIVALENCE (IPACK(2),CPACK),(DBNAME,IDBN),(IPACK,CHARP)
C
        COMMON / DATAG / DTP(2),C(32,2),GAMMA(2),R(2),VP(8,2),IE
        COMMON / CDATA / EM(2),PATM,RK,FR,GAM
        COMMON/NDATA/AK,ALWS(3,2),AREAHX(2),BK,CD(3,2),CK,
       1                 DAR(3,2),DCD(3,2),FK(3,2),TOL(2)
C
CVAX    DATA IPACK(2),IPACK(3),IPACK(4),IPACK(5),IPACK(6),IPACK(7)
C61C 1 / 'STS6','1CDB','.DAT','    ','    ','    ' /
C61A 1 / 'STS6','1ADB','.DAT','    ','    ','    ' /
C61B 1 / 'STS6','1BDB','.DAT','    ','    ','    ' /
C51B 1 / 'STS5','1BDB','.DAT','    ','    ','    ' /
CVAX    DATA IPACK(8),IPACK(9),IPACK(10),IPACK(11),IPACK(12),IPACK(13)
C61C 1 / 86,012,11,54,59,997 /
C61A 1 / 85,303,17,00,00,010 /
C61B 1 / 85,331,00,29,00,006 /
C51B 1 / 85,119,16,02,17,989 /
C
        DATA IPACK(8),IPACK(9),IPACK(10),IPACK(11),IPACK(12),IPACK(13)
       1 / 88,273,15,37,00,016 /
        DATA IPACK(14)/32/,IPACK(15)/2/,IPACK(16)/1920/
C
        DATA TSTART,TSTOP,SRATE, TSTAGE / 0.0D0, 524.D0, 1.0, 128. /
C
        OPEN( UNIT=1,FILE='PREPIN',STATUS='OLD',ACCESS='SEQUENTIAL' )
        OPEN( UNIT=3,FILE='PREPPL',STATUS='NEW',ACCESS='SEQUENTIAL' )
        OPEN( UNIT=9,FILE='PREPOT',STATUS='NEW',ACCESS='SEQUENTIAL' )
C
        IPACK(1) = -1
        CPACK = 'ASTS317'
        DBNAME = 'STS26DB'
C61C    DBNAME = 'STS61CDB'
        COND = 2.831685E-1/4.535924
C
        DO 10 I = 1, 4
   10   IPACK(I+3) = IDBN(I)
C
        WRITE(6,12) IPACK(1),CPACK,DBNAME,(IPACK(J), J = 8, 14)
   12   FORMAT( 2X, I2, 2X, A8, A17, 2X, 7I4 )
C
        READ(1, 20) (MSID(J), J = 1, (IPACK(14)+20))
   20   FORMAT( A12 )
C
        WRITE(6, 25) (MSID(J), J = 1, (IPACK(14)+20))
```

```
   25 FORMAT( 2X, A12 )
C
      I = 0
      CALL ACCREQ(2,IPACK(2),DBNAME,IPACK(14),MSID,NUMBR,ISTAT,ERROR)
      IF ( ERROR.NE.0 ) THEN
      WRITE(6,501) ERROR
501   FORMAT(' ERROR IN ACCREQ - ERROR=', I5 )
      STOP
      END IF
C
   50 CONTINUE
C
      I = I + 1
      TIME = TSTART
C
      CALL ACCESS( IPACK,IW,MSID,SRATE,TIME,TSTOP,STALE,VAL,ISTAT )
C
      IF( IPACK(1).EQ.6 ) GO TO 200
C
      SPTIME(I) = SNGL( TIME )
C
      FCV(1,1) = FLOAT( IFIX ( VAL(1) ) )
      FCV(2,1) = FLOAT( IFIX ( VAL(2) ) )
      FCV(3,1) = FLOAT( IFIX ( VAL(3) ) )
      FCV(1,2) = FLOAT( IFIX ( VAL(4) ) )
      FCV(2,2) = FLOAT( IFIX ( VAL(5) ) )
      FCV(3,2) = FLOAT( IFIX ( VAL(6) ) )
      PXM(1,1) = VAL(7)
      PXM(2,1) = VAL(8)
      PXM(3,1) = VAL(9)
      PXM(1,2) = VAL(10)
      PXM(2,2) = VAL(11)
      PXM(3,2) = VAL(12)
      TXM(1,1) = VAL(13) + FR
      TXM(2,1) = VAL(14) + FR
      TXM(3,1) = VAL(15) + FR
      TXM(1,2) = VAL(16) + FR
      TXM(2,2) = VAL(17) + FR
      TXM(3,2) = VAL(18) + FR
C
C  CALCULATE PRESSURANT MASS FLOW RATES FROM PRESSURE AND TEMPERATURES
C
      DO 100 L = 1, 2
      DO 100 M = 1, 3
C
      WZ = 0.0
      IF( PXM(M,L).LE.0.0 ) GO TO 95
      PX(M,L) = PXM(M,L)
      TX(M,L) = TXM(M,L)
      PP = PXM(M,L)/PATM
      TT = TXM(M,L)
C
   80 CONTINUE
C
      CALL FINDD( PP, TT, L, DD )
```

```
      RHO = DD*COND*EM(L)
C
   90 CONTINUE
C
      WEHE(M,L) = (ALWS(M,L) + FCV(M,L)*DAR(M,L))*PX(M,L)*SQRT(EM(L)/
     1         (TX(M,L)*1545.0))*3.8839427*(CD(M,L) + FCV(M,L)*DCD(M,L))
C     WRITE(*, 901) AREAHX(L), RHO
      VEL = 144.0*WEHE(M,L)/(AREAHX(L)*RHO)
C
      PX(M,L) = PXM(M,L)+(1.0-FK(M,L))*RHO*VEL**2/(2.0*32.174*144.0)
      PSTAT = PX(M,L) - RHO*VEL**2/(2.0*32.174*144.0)
      TX(M,L) = TXM(M,L)*(PX(M,L)/PSTAT)**GAM
C
      IF( ABS(WEHE(M,L) - WZ).LE.TOL(L) ) GO TO 100
      WZ = WEHE(M,L)
      PP = PSTAT/PATM
      GO TO 80
C
95    CONTINUE
      WEHE(M,L) = 0.0
C
  100 CONTINUE
C
      PCM(1) = VAL(19)
      PCM(2) = VAL(20)
      PCM(3) = VAL(21)
      Y8(1) = VAL(22)
      Y8(2) = VAL(23)
      Y8(3) = VAL(24)
      PH(1) = VAL(25)
      PH(2) = VAL(26)
      PH(3) = VAL(27)
      TH(1) = VAL(28)
      TH(2) = VAL(29)
      TH(3) = VAL(30)
C
      DO 150 J = 1, 3
      GRAPH(I,J) = PCM(J)
      GRAPH(I,J+3) = WEHE(J,1)
      GRAPH(I,J+6) = WEHE(J,2)
      GRAPH(I,J+9) = Y8(J)
      GRAPH(I,J+12) = PH(J)
      GRAPH(I,J+15) = TH(J)
150   CONTINUE
C
      IF ( SPTIME(I).LT.TSTAGE ) THEN
C
      GRAPH(I,19) = VAL(31)
      GRAPH(I,20) = VAL(32)
C
      ELSE
C
      GRAPH(I,19) = 0.0
      GRAPH(I,20) = 0.0
C
```

```
      END IF
C
      IMAX = I
C
      GO TO 50
C
200   CONTINUE
C
      DO 225 I = 1, IMAX
      WRITE(9, 995) SPTIME(I), (GRAPH(I,J), J = 1, 20)
995   FORMAT( 3(7E15.8,/) )
225   CONTINUE
C
      XMIN = SPTIME(1)
      XMAX = SPTIME(IMAX)
C
      DO 300 NM = 1, 20
C
      YMIN = GRAPH(1,NM)
      YMAX = YMIN
      YLAB = MSID( IPACK(14)+NM )
      DO 250 I = 1, IMAX
      YMIN = AMIN1( GRAPH(I,NM), YMIN )
      YMAX = AMAX1( GRAPH(I,NM), YMAX )
      PVAR(I) = GRAPH(I,NM)
250   CONTINUE
      WRITE(6,25) YLAB
C
      CALL PLOT2( XMIN, XMAX, YMIN, YMAX )
      CALL PLOT3( '.', SPTIME, PVAR(1), IMAX )
      CALL PLOT4( 9, 'TIME  (SEC)',6, YLAB, 13, 'PROP MEASURES' )
C
300   CONTINUE
C
      WRITE(6,903) IPACK(1)
903   FORMAT( 5X, I5 )
C
      CLOSE ( UNIT=3 )
      CLOSE ( UNIT=9 )
C
      STOP
      END
```

```
C
      DIMENSION AZM(9),ELM(9),RNGM(9),AZM2(5),ELM2(5),RNGM2(5)
     1          ,KOUT(5)
CSTGX1CSTGX     ,KOUT(3)
C
      DATA NOUT / 5 /
CSTG1 DATA NOUT / 3 /
      DATA KOUT / 1, 3, 5, 4, 6 /
CSTG1 DATA KOUT / 1, 3, 5 /
CSTG2 DATA KOUT / 3, 4, 6 /
      DATA ITIME / 0 /
C
      OPEN ( UNIT=1, NAME='RADAR.DAT', STATUS='OLD' )
      OPEN ( UNIT=3, NAME='TRACK.DAT', STATUS='NEW' )
C
10    CONTINUE
C
      READ(1,901,END=100) TIME,(AZM(I),ELM(I),RNGM(I),I = 1,6)
C
      TSAVE = TIME
      IF( ITIME.GT.0 ) GO TO 16
      ITIME = 1
      TINC = 1.0
      TIME = 1.0
12    CONTINUE
      DO 14 I = 1, NOUT
      AZM2(I) = 0.0
      ELM2(I) = 0.0
      RNGM2(I) = 0.0
14    CONTINUE
C     WRITE(3, 902) TIME, (AZM2(I), ELM2(I), RNGM2(I), I = 1, NOUT)
      WRITE(3, 902) TIME
      WRITE(3, 902) (AZM2(I), ELM2(I), RNGM2(I), I = 1, NOUT)
      TIME = TIME + TINC
      IF( TIME.GE.TSAVE ) GO TO 16
      GO TO 12
16    CONTINUE
C
      ITIME = IFIX( TIME )
      XTIME = FLOAT( ITIME )
      IF( XTIME.EQ.TLAST ) GOTO 10
C
      TLAST = TIME
      DO 20 IPICK= 1, NOUT
      IOUT = KOUT(IPICK)
      AZM2(IPICK) = AZM(IOUT)
      IF( AZM2(IPICK).GT.180.0 ) AZM2(IPICK) = AZM(IOUT) - 360.0
      ELM2(IPICK) = ELM(IOUT)
      RNGM2(IPICK) = RNGM(IOUT)
20    CONTINUE
C
C     WRITE(3, 902) TIME, (AZM2(I), ELM2(I), RNGM2(I), I = 1, NOUT)
      WRITE(3, 902) TIME
      WRITE(3, 902) (AZM2(I), ELM2(I), RNGM2(I), I = 1, NOUT)
C
      GO TO 10
C
100   CONTINUE
C
      CLOSE ( UNIT=3 )
      CLOSE ( UNIT=1 )
C
901   FORMAT( 1X, F5.1,7(1X,F8.4,1X,F7.4,1X,F8.0) )
902   FORMAT( 5X, 3E15.8 )
C
```

```
STOP
END
```

```
C     PROGRAM CONTRL
      PROGRAM MAIN
C
      DIMENSION STALE(26),VAL(26)
     1 ,SPTIME(600),TSHFT(600),GRAPH(600,17),PVAR(600),VEC(600)
      DIMENSION TMP1(3,3),TMP2(3,3),TMP3(3,3),TMP4(3,3),TMP5(3,3)
     1 ,REFMT1(3,3),REFMT2(3,3),REFMT3(3,3)
     3, CBI1(3,3),CBI2(3,3),CBI3(3,3),C50BR1(3,3),C50BR2(3,3)
     4, TBBRI0(3,3)
     5, THT1(3),THT2(3),THTM(3)
C
      INTEGER*4 IPACK(30),IW(1440),ISTAT(26),ERROR,IDBN(4)
      REAL*8 TIME1, TIME2, TSTART, TSTOP
      CHARACTER MSID*12(43),CPACK*8,DBNAME*17,CHARP*4(30),YLAB*12
      EQUIVALENCE (IPACK(2),CPACK),(DBNAME,IDBN),(IPACK,CHARP)
C
      DATA IPACK(8),IPACK(9),IPACK(10),IPACK(11),IPACK(12),IPACK(13)
     1 / 88,273,15,37,00,000 /
      DATA IPACK(14)/26/,IPACK(15)/2/,IPACK(16)/1560/
C
      DATA IST / 11 /
      DATA TBBRI0 / 2*0.0,-1.0,0.0,1.0,0.0,1.0,2*0.0 /
C
C51B  DATA REFMT1/-.694542,.496024,-.521125,-.301635,.45684,.836847
C51B 1 ,.653167,.738415,-.167677/
C51B  DATA REFMT2/-.120625,-.078044,.989626,.779794,-.624357,.045811
C51B 1 ,.614305,.77723,.136171/
C51B  DATA REFMT3/.517692,-.716205,-.468022,-.244835,.400140,-.883144
C51B 1 ,.817987,.571785,.0317968/
C
C61A  DATA REFMT1/.584975,-.342975,.735068,.487126,-.576157,-.656315
C61A 1 ,.648469,.741999,-.170073 /
C61A  DATA REFMT2/.339075,-.0968083,-.935765,-.712631,.622931,-.322666
C61A 1 ,.614154,.776263,.142232 /
C61A  DATA REFMT3/-.561507,.802985,.199810,.0904577,-.180458,.979414
C61A 1 ,.822512,.568022,.0286922/
C
C61C  DATA REFMT1/.6133972,-.3773153,.6938130,.4503511,-.5545661
C61C 1             ,-.6997428,.6487887,.741680,-.1702449/
C61C  DATA REFMT2/.2954440,-.05880850,-.9535483,-.7317411,.6277722
C61C 1             ,-.2654373,.6142209,.7761723,.1424394/
C61C  DATA REFMT3/-.5544043,.7906296,.2598850,.1250687,-.2295741
C61C 1             ,.9652219,.8227961,.5676264,.02839340/
C
      DATA REFMT1/.7124555,-.6688998,-.2120820,-.2697270,.0179653
     1             ,-.9627689,.6478066,.7431343,-.1676210/
      DATA REFMT2/-.4540111,.4971972,-.7393705,-.6433136,.3912134
     1             ,.6581028,.6164586,.7744331,.1422388/
      DATA REFMT3/-.1936510,.2368643,.9520475,.5359492,-.7872760
     1             ,.3048846,.8217406,.5692902,.0255092/
C
      DATA TSTART,TSTOP,SRATE / -10.0D0, 523.0D0, 1.0 /
      DATA JUMP / 0 /
C
      OPEN( UNIT=1,FILE='CTRLIN',STATUS='OLD',ACCESS='SEQUENTIAL' )
```

```
      OPEN( UNIT=3,FILE='CTRLPL',STATUS='NEW',ACCESS='SEQUENTIAL' )
      OPEN( UNIT=9,FILE='CTRLOT',STATUS='NEW',ACCESS='SEQUENTIAL' )
C
      CRAD = 57.295779
      IPACK(1) = -1
      CPACK = 'ASTS317'
      DBNAME = 'STS26DB'
C61C  DBNAME = 'STS61CDB'
C
      DO 10 I = 1, 4
   10 IPACK(I+3) = IDBN(I)
C
      WRITE(6, 12) IPACK(1),CPACK,DBNAME,(IPACK(J), J= 8, 14)
   12 FORMAT( 2X, I2, 2X, A4,A17, 2X, 7I4 )
C
      READ(1, 20) (MSID(J), J=1, (IPACK(14)+17))
   20 FORMAT( A12 )
      CLOSE ( UNIT=1 )
C
      WRITE(6, 25) (MSID(J), J=1, (IPACK(14)+17))
   25 FORMAT( 2X, A12 )
C
      I=0
      CALL ACCREQ(2,IPACK(2),DBNAME,IPACK(14),MSID,NUMBR,ISTAT,ERROR)
      IF ( ERROR.NE.0 ) THEN
      WRITE(6,501) ERROR
501   FORMAT(' ERROR IN ACCREQ - ERROR=', I5)
      STOP
      END IF
C
   50 CONTINUE
C
      I = I + 1
      TIME1 = TSTART
C
      CALL ACCESS( IPACK,IW,MSID,SRATE,TIME1,TSTOP,STALE,VAL,ISTAT )
      IF( IPACK(1).EQ.6 ) GO TO 200
      SPTIME(I) = SNGL( TIME1 ) - 1.0
      SPTMAX = SNGL( TSTOP )
C
      PITCH=CRAD*VAL(12)
      YAW=CRAD*VAL(13)
      ROLI=CRAD*VAL(14)
      ROLO=CRAD*VAL(15)
C
      CALL TMATY ( -YAW, TMP1 )
      CALL TMATR ( -ROLI, TMP2 )
      CALL MULT ( TMP2, TMP1, TMP3, 3, 3, 3 )
      CALL TMATP ( -PITCH, TMP1 )
      CALL MULT ( TMP1, TMP3, TMP2, 3, 3, 3 )
      CALL TMATR ( -ROLO, TMP1 )
      CALL MULT ( TMP1, TMP2, TMP3, 3, 3, 3 )
      CALL TMATP ( -10.6, TMP2 )
      CALL MULT ( TMP2, TMP3, TMP4, 3, 3, 3 )
C
```

```
      PITCH = CRAD*VAL(16)
      YAW = CRAD*VAL(17)
      ROLI = CRAD*VAL(18)
      ROLO = CRAD*VAL(19)
C
      CALL TMATY ( -YAW, TMP1 )
      CALL TMATR ( -ROLI, TMP2 )
      CALL MULT ( TMP2, TMP1, TMP3, 3, 3, 3 )
      CALL TMATP ( -PITCH, TMP1 )
      CALL MULT ( TMP1, TMP3, TMP2, 3, 3, 3 )
      CALL TMATR ( -ROLO, TMP1 )
      CALL MULT ( TMP1, TMP2, TMP3, 3, 3, 3 )
      CALL TMATP ( -10.6, TMP1 )
      CALL MULT ( TMP1, TMP3, TMP5, 3, 3, 3 )
C
      IF ( JUMP.LT.1 ) THEN
C
      CALL MULT ( TBBRI0, TMP4, C50BR1, 3, 3, 3 )
      CALL MULT ( TBBRI0, TMP5, C50BR2, 3, 3, 3 )
      JUMP=1
C
      WRITE(6, 905) ((C50BR1(J,K), J = 1, 3), K = 1, 3)
      WRITE(6, 905) ((C50BR2(J,K), J = 1, 3), K = 1, 3)
C
      ELSE
      END IF
C
      CALL TRANS ( TMP4, TMP3, 3, 3 )
      CALL MULT ( C50BR1, TMP3, CBI1, 3, 3, 3 )
      IF( ABS(CBI1(3,1)).GT.1.0 ) CBI1(3,1)=SIGN(1.0, CBI1(3,1))
      THT1(2) = CRAD*ASIN( -CBI1(3,1) )
      THT1(1) = CRAD*ATAN2( CBI1(3,2), CBI1(3,3) )
      THT1(3) = CRAD*ATAN2( CBI1(2,1), CBI1(1,1) )
C
      CALL TRANS ( TMP5, TMP3, 3, 3 )
      CALL MULT ( C50BR2, TMP3, CBI2, 3, 3, 3 )
      IF ( ABS(CBI2(3,1)).GT.1.0 ) CBI2(3,1) = SIGN( 1.0, CBI2(3,1) )
      THT2(2) = CRAD*ASIN( -CBI2(3,1) )
      THT2(1) = CRAD*ATAN2( CBI2(3,2), CBI2(3,3) )
      THT2(3) = CRAD*ATAN2( CBI2(2,1), CBI2(1,1) )
C
      DO 100 II = 1, 3
C
      IF ( ABS(THT1(II) - THT2(II)).GT.1.0 ) THEN
C
         THTM(II) = THTM(II)
C
      ELSE
C
         THTM(II) = 0.5*( THT1(II) + THT2(II) )
C
      END IF
C
100   CONTINUE
C
```

```
      QI = VAL(24)
      RI = VAL(25)
      PI = VAL(26)
C
      IF( (I-IST).LE.0 ) GO TO 180
      GRAPH((I-IST),1) = VAL(1)
      GRAPH((I-IST),2) = VAL(2)
      GRAPH((I-IST),3) = VAL(3)
      GRAPH((I-IST),4) = VAL(4)
      GRAPH((I-IST),5) = VAL(5)/100.
      GRAPH((I-IST),6) = VAL(6)
      GRAPH((I-IST),7) = VAL(7)
      GRAPH((I-IST),8) = VAL(8)
      GRAPH((I-IST),9) = VAL(9)
      GRAPH((I-IST),10) = VAL(10)
      GRAPH((I-IST),11) = VAL(11)
      GRAPH((I-IST),15)=THTM(1)
      GRAPH((I-IST),16)=THTM(2)
      GRAPH((I-IST),17)=THTM(3)
      GRAPH((I-IST),12)=PI/CRAD
      GRAPH((I-IST),13)=QI/CRAD
      GRAPH((I-IST),14)=RI/CRAD
      TSHFT(I-IST) = SPTIME(I)
180   CONTINUE
C
      IMAX = I - IST
C
      THTM(1) = THTM(1)
      THTM(2) = THTM(2)
      THTM(3) = THTM(3)
C
      GO TO 50
C
200   CONTINUE
C
220   CONTINUE
C
      XMIN = TSHFT(1)
      XMAX = TSHFT(IMAX)
C
      DO 300 NM = 1, 17
C
      YMIN = GRAPH(1,NM)
      YMAX = YMIN
C
      YLAB = MSID( NM+IPACK(14) )
C
      DO 250 I = 1, IMAX
      YMIN = AMIN1( GRAPH(I,NM), YMIN )
      YMAX = AMAX1( GRAPH(I,NM), YMAX )
      PVAR(I) = GRAPH(I,NM)
250   CONTINUE
C
      DO 270 II = 1, IMAX
      VEC(II) = PVAR(II)
```

```
270   CONTINUE
C
C     CALL DATSMTH ( IMAX, VEC, PVAR )
C
      DO 280 II = 1, IMAX
      GRAPH(II, NM) = PVAR(II)
280   CONTINUE
C
      CALL PLOT2 ( XMIN, XMAX, YMIN, YMAX )
      CALL PLOT3 ( '.', TSHFT, PVAR(1), IMAX )
      CALL PLOT4 ( 9, 'TIME  (SEC)',6,YLAB,13,'CONTRL INPUT' )
C
300   CONTINUE
C
      DO 350 I = 1, IMAX
      WRITE(9, 905) TSHFT(I),(GRAPH(I,J), J = 1, 4)
      WRITE(9, 905) (GRAPH(I,J), J = 5, 11)
      WRITE(9, 905) (GRAPH(I,J), J = 12, 14)
      WRITE(9, 905) (GRAPH(I,J), J = 15, 17)
350   CONTINUE
C
      CLOSE ( UNIT = 3 )
      CLOSE ( UNIT = 9 )
C
      STOP
C
3     FORMAT(T2, 7I5)
901   FORMAT(3X, 4E15.8)
902   FORMAT( 5X, 4E15.8 )
903   FORMAT( 5X, I5 )
905   FORMAT(5X, 7E15.8)
C
      END
```

```
C     PROGRAM METTST
      PROGRAM MAIN
C
      DIMENSION STALE(11), VAL(11)
     1          ,ALT(405),GRAPH(405,13),PVAR(405)
C
      INTEGER*4 IPACK(30),IW(660),ISTAT(11),ERROR,IDBN(4)
      REAL*8 TIME, TSTART, TSTOP
      CHARACTER MSID*12(14),CPACK*8,DBNAME*17,CHARP*4(30),YLAB*12
      EQUIVALENCE (IPACK(2),CPACK),(DBNAME,IDBN),(IPACK,CHARP)
C
      DATA IPACK(8),IPACK(9),IPACK(10),IPACK(11),IPACK(12),IPACK(13)
     1 / 88,273,15,37,00,000 /
C61C 1 / 86,012,11,54,59,997 /
      DATA IPACK(14)/11/,IPACK(15)/2/,IPACK(16)/660/
C
CVAX  DATA TSTART,TSTOP,SRATE / 0.0D0, 160000.D0, .00328084 /
      DATA TSTART,TSTOP,SRATE / 0.0D0, .804D0,  500. /
C
      OPEN( UNIT=1,FILE='METDIN',STATUS='OLD',ACCESS='SEQUENTIAL' )
      OPEN( UNIT=3,FILE='METDOT',STATUS='NEW',ACCESS='SEQUENTIAL' )
C
      IPACK(1) = -1
      CPACK = 'ASTS317'
      DBNAME = 'MET026  FIN1E'
C61C  DBNAME = 'MET61C  FIN1M'
      CRAD = 57.295779
C
      DO 10 I = 1, 4
   10 IPACK(I+3) = IDBN(I)
C
      WRITE(6, 12) IPACK(1),CPACK,DBNAME,(IPACK(J), J = 8, 14)
   12 FORMAT( 2X, I2, 2X, A8,A17, 2X, 7I4 )
C
      READ(1, 20) (MSID(J), J = 1, (IPACK(14)+3))
   20 FORMAT( A12 )
C
      WRITE(6, 25) (MSID(J), J = 1, (IPACK(14)+3))
   25 FORMAT( 2X, A12 )
C
      I = 0
      CALL ACCREQ(2,IPACK(2),DBNAME,IPACK(14),MSID,NUMBR,ISTAT,ERROR)
      IF ( ERROR.NE.0 ) THEN
      WRITE(6,501) ERROR
501   FORMAT(' ERROR IN ACCREQ - ERROR=', I5 )
      STOP
      END IF
C
   50 CONTINUE
C
      I = I + 1
      TIME = TSTART
C
      CALL ACCESS( IPACK,IW,MSID,SRATE,TIME,TSTOP,STALE,VAL,ISTAT )
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF( IPACK(1).EQ.6 ) GO TO 200
C
CVAX  SPALT = SNGL( TIME )/.3048
      SPALT = VAL(11)
C
      WS = VAL(1)
      WD = VAL(2)
C
      IF ( I.EQ.1 ) THEN
      DENS = .002378
      PRES = 2116.
      ELSE
      DENS = VAL(4)*.6243E-4/32.174
      PRES = VAL(5)*144.*.014505
      END IF
      TEMP = ( VAL(3) + 273.16 )*9./5.
C
      VSOUND = 49.02*SQRT(TEMP)
      VWX = -WS*COS( WD/CRAD )
      VWY = -WS*SIN( WD/CRAD )
C
      ALT(I) = SPALT
      GRAPH(I,1) = WS
      GRAPH(I,2) = WD
      GRAPH(I,3) = TEMP
      GRAPH(I,4) = DENS
      GRAPH(I,5) = PRES
      GRAPH(I,6) = VSOUND
      GRAPH(I,7) = VWX
      GRAPH(I,8) = VWY
C
      UWS = VAL(6)
      UWD = VAL(7)
      UDENS = VAL(9)*.6243E-4/32.174
      UPRES = VAL(10)*144.*.014505
      UTEMP = VAL(8)*9./5.
C
      UVS = 49.02*UTEMP/(2.*SQRT(TEMP))
      UVWX1 = UWS*COS( WD/CRAD )
      UVWX2 = -WS*SIN( WD/CRAD )*UWD/CRAD
      UVWX = SQRT ( UVWX1**2 + UVWX2**2 )
      UVWY1 = UWS*SIN( WD/CRAD )
      UVWY2 = WS*COS( WD/CRAD )*UWD/CRAD
      UVWY = SQRT ( UVWY1**2 + UVWY2**2 )
C
      GRAPH(I,9) = UDENS
      GRAPH(I,10) = UPRES
      GRAPH(I,11) = UTEMP
      GRAPH(I,12) = UVWX
      GRAPH(I,13) = UVWY
C
      IMAX = I
C
      GO TO 50
C
```

```
20     CONTINUE
C
       DO 220 I = 1, IMAX
       WRITE(3, 998) ALT(I),(GRAPH(I,J), J = 4, 8)
       WRITE(3, 999) (GRAPH(I,J), J = 9, 13)
220    CONTINUE
998    FORMAT( 5X, 6E15.8 )
999    FORMAT( 20X, 5E15.8 )
C
       XMIN = ALT(1)
       XMAX = ALT(IMAX)
C
       DO 300 NM = 1, (IPACK(14)+3)
C
       YMIN = GRAPH(1, NM)
       YMAX = YMIN
       YLAB = MSID( NM )
       DO 250 I = 1, IMAX
       YMIN = AMIN1( GRAPH(I,NM), YMIN )
       YMAX = AMAX1( GRAPH(I,NM), YMAX )
       PVAR(I) = GRAPH(I,NM)
250    CONTINUE
C
       CALL PLOT2 ( XMIN, XMAX, YMIN, YMAX )
       CALL PLOT3 ( '.', ALT, PVAR(1), IMAX )
       CALL PLOT4 ( 9, 'FT ALTITUDE',6,YLAB,13,'METEOROLOGY' )
C
300    CONTINUE
       WRITE(*, 903) IPACK(1)
903    FORMAT( 5X, I5 )
C
       CLOSE ( UNIT=3 )
C
       STOP
C
       END
```

```
C     ROUTINE TO PLOT OUTPUT OF THE LFILTER.EXE PROGRAM
C
      DIMENSION YLBFIL(3,45),T(500),PVAR(500,3),XKM(500,30),PKM(500,30)
     1          ,RG(30),COVZ(500,15), RESID(500,15)
      DIMENSION YLAB(3)
C
      OPEN(UNIT=2,FILE='YLABLFIL.DAT',STATUS='OLD',ACCESS='SEQUENTIAL',
     1 RECL=9,CARRIAGECONTROL='LIST' )
C
      READ(2,5)((YLBFIL(I,J),I=1,3),J=1,45)
5     FORMAT( 2A4, A1 )
C
      CLOSE ( UNIT=2 )
C
      WRITE(*,6)((YLBFIL(I,J),I=1,3),J=1,45)
6     FORMAT( T2, 2A4, A1 )
C
      OPEN ( UNIT=1, FILE='LFILOUT.DAT', STATUS='OLD',
     1 ACCESS='SEQUENTIAL', FORM='UNFORMATTED' )
      OPEN ( UNIT=3, FILE='PLTFIL.DAT', STATUS='NEW' )
C
      I = 1
10    CONTINUE
      J = 0
20    CONTINUE
C
      J = J + 1
C
      READ(1, END = 100) T(I), TL, IMEAS, NMEAS
      WRITE(*, 901) T(I), TL, IMEAS, NMEAS
      READ(1) (XKM(I,K), K = 1, 30)
      READ(1) (PKM(I,K), K = 1, 30)
C     READ(1) (RG(K), K = 1, 30)
      READ(1) COVZ(I,J)
      READ(1) RESID(I,J)
C
      IF ( IMEAS - NMEAS ) 20, 30, 30
30    CONTINUE
C
      IMAX = I
      I = I + 1
      IF ( I.GE.500 ) GO TO 100
      GO TO 10
C
100   CONTINUE
C
      XMIN = T(1)
      XMAX = T(IMAX)
C
      DO 200 KK = 1, 30
C
      DO 110 II = 1, IMAX
C
      SIGV = SQRT ( PKM(II,KK) )
      BUPPER = XKM(II,KK) + SIGV
      BLOWER = XKM(II,KK) - SIGV
C
      PVAR(II,1) = BUPPER
      PVAR(II,2) = XKM(II,KK)
      PVAR(II,3) = BLOWER
```

```
C
110   CONTINUE
C
C     WRITE(3, 903) ((PVAR(II, JJ), JJ = 1, 3), II = 1, IMAX)
C
      WRITE(*, 904) KK
      WRITE(*,6) YLBFIL(1,KK)
      YLAB(1) = YLBFIL(1,KK)
      YLAB(2) = YLBFIL(2,KK)
      YLAB(3) = YLBFIL(3,KK)
C     WRITE(*,6)(YLAB(L),L=1,3)
C
      YMIN = PVAR(1,3)
      YMAX = PVAR(1,1)
C
      DO 120 II = 1, IMAX
      YMIN = AMIN1( PVAR(II,3), YMIN )
      YMAX = AMAX1( PVAR(II,1), YMAX )
120   CONTINUE
C
      CALL PLOT2 ( XMIN, XMAX, YMIN, YMAX )
      CALL PLOT3 ( '+', T(1), PVAR(1,1), IMAX )
      CALL PLOT3 ( '.', T(1), PVAR(1,2), IMAX )
      CALL PLOT3 ( '-', T(1), PVAR(1,3), IMAX )
      CALL PLOT4 ( 9,'TIME  (SEC)',6,YLAB(1),13,'PLTFIL ESTIMATE ' )
C
200   CONTINUE
C
      DO 300 JJ = 1, 15
C
      DO 210 II = 1, IMAX
C
      SIGR = SQRT ( COVZ(II, JJ) )
C
      PVAR(II,1) = SIGR
      PVAR(II,2) = RESID(II,JJ)
      PVAR(II,3) = -SIGR
C
210   CONTINUE
C
C      WRITE(3, 903) ((PVAR(II,KK), KK = 1, 3), II = 1, IMAX)
C
      YLAB(1) = YLBFIL(1,JJ+30)
      YLAB(2) = YLBFIL(2,JJ+30)
      YLAB(3) = YLBFIL(3,JJ+30)
C
      YMIN = PVAR(1,3)
      YMAX = PVAR(1,1)
C
      DO 220 II = 1, IMAX
      YMIN = AMIN1 ( PVAR(II,3), YMIN )
      YMIN = AMIN1 ( PVAR(II,2), YMIN )
      YMAX = AMAX1 ( PVAR(II,1), YMAX )
      YMAX = AMAX1 ( PVAR(II,2), YMAX )
220   CONTINUE
C
      WRITE(*, 904) JJ
      WRITE(*, 6) YLBFIL(1,JJ+30)
      CALL PLOT2 ( XMIN, XMAX, YMIN, YMAX )
      CALL PLOT3 ( '-', T(1), PVAR(1,1), IMAX )
```

```
      CALL PLOT3 ( '+', T(1), PVAR(1,2), IMAX )
      CALL PLOT3 ( '-', T(1), PVAR(1,3), IMAX )
      CALL PLOT4 ( 9,'TIME  (SEC)',6,YLAB(1),13,'PLTFIL RESIDUAL ' )
C
300   CONTINUE
C
C
      CLOSE ( UNIT=1 )
      CLOSE ( UNIT=3 )
C
      STOP
C
901   FORMAT( 2E15.8, 2I5 )
902   FORMAT( 5E15.8 )
903   FORMAT( 5X, 3E15.8 )
904   FORMAT( 2X, I5 )
C
      END
```

```
C
      COMMON / EARTH / GRAV(3), AGMX(3,3)
      COMMON / RDMEAS / AZM(5), ELM(5), RNGM(5)
      COMMON / RDREST / AZHAT(5), ELHAT(5), RNGHAT(5)
      COMMON / PROPAG / XKP(30), PKP(30,30)
      COMMON / PREUP / XKM(30), PKM(30,30)
      COMMON / UDWORK / RH(30),U(465),PO(465),SF(465),SG(30),RESID,COVZ
      COMMON / UDWRK2 / PHI(30,30),PHIU(30,30),W(30,54)
     1                 ,D(30),E(30),DW(54),V(54)
      COMMON / SMOWRK / TMP(30,24), GT(24,30), GQGT(30,30)
      COMMON / RSTATE / RBRI(3),VBRI(3),ABRI(3),THTI(3)
     1                 ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                 ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
      COMMON / METEOR / TALT(400),TSRHO(400),TSP(400),TSSUND(400)
     1                                        ,TVWX(400),TVWY(400),NALT
     2                                        ,USRHO(400),USP(400),USSUND(400)
     3                                        ,UVWX(400),UVWY(400)
C
C     BLOCK DATA
C
      COMMON / TIMDAT / TIME, TMAX, HSTEP, TSAMP, TSTOP
      COMMON / ITYPE / ISMOOTH
      COMMON / IREFOUT / IASSESS
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
      COMMON / LAUCOR / OLATD, OLONG, OHT
     1                 ,CM50EF(3,3)
      COMMON / RDRDAT / XNO(5),RLAT(5),RLONG(5),RHT(5),NRDR
      COMMON / APRIOR / ER(3), EV(3)
     1                 ,ETLT(3),EAB(3),EASF(3),EGRV(3),EX(3)
     2                 ,ERDR(3)
      COMMON / NOISES / R(3), G(30,24), Q(24,24), NP
      COMMON / FPARAM / TAUT(3),TAUA(3),TAUS(3),TAUG(3),TAUX(3),TAUR(3)
     1                 ,UT(3),UA(3),US(3),UG(3),UX(3),UR(3)
      COMMON / LINFMT / F(30,30), N
      COMMON / LINHMT / H(15,30), NRMEAS
C
      OPEN( UNIT=2, FILE='TRACK.DAT', STATUS='OLD' )
      OPEN( UNIT=3, FILE='REFIPT.DAT', STATUS='OLD' )
      OPEN( UNIT=6, FILE='LFILOUT.DAT', STATUS='NEW'
     1      ,ACCESS='SEQUENTIAL', FORM='UNFORMATTED' )
C
      IF ( ISMOOTH.EQ.1 ) THEN
      OPEN( UNIT=7, FILE='SREFIPT.DAT', STATUS='NEW' )
C    1C     ,ACCESS='SEQUENTIAL', FORM='UNFORMATTED' )
      OPEN( UNIT=8, FILE='SMOIPT.DAT', STATUS='NEW'
     1      ,ACCESS='SEQUENTIAL', FORM='UNFORMATTED' )
      ELSE
      END IF
C
      IF ( IASSESS.EQ.1 ) THEN
C
      OPEN( UNIT=4, FILE='METDAT.DAT', STATUS='OLD' )
C
      DO 20 I = 1, NALT
      READ(4,998) TALT(I),TSRHO(I),TSP(I),TSSUND(I),TVWX(I),TVWY(I)
      READ(4,999) USRHO(I),USP(I),USSUND(I),UVWX(I),UVWY(I)
C     WRITE(*,997) TALT(I),TSRHO(I),TSP(I),TSSUND(I)
20    CONTINUE
C
      CLOSE( UNIT=4 )
C
      OPEN( UNIT=9, FILE='LASSOUT.DAT',STATUS='NEW' )
C    1      , ACCESS='DIRECT', RECL=90, IOSTAT=IOS )
C
```

```
      ELSE
      END IF
C
      KTEST = TSAMP/HSTEP
C
      CALL INITIL
C
50    CONTINUE
C
      TIME = TIME + HSTEP
C
      CALL UDTIME
C
      KS = KS + 1
      IF( KS - KTEST ) 200, 100, 100
100   CONTINUE
      KS = 0
C
      CALL GETDAT( TIME )
C
      CALL RADAR
C
      IF ( IASSESS.EQ.1 ) THEN
C
         CALL ASSESS
C
      ELSE
      END IF
C
      GO TO 800
C
200   CONTINUE
C
800   CONTINUE
C
      IF( TIME - TMAX ) 50, 900, 900
C
900   CONTINUE
C
      CLOSE( UNIT=2 )
      CLOSE( UNIT=3 )
      CLOSE( UNIT=6 )
      CLOSE( UNIT=7 )
      CLOSE( UNIT=8 )
      CLOSE( UNIT=9 )
C
901   FORMAT ( I5 )
902   FORMAT( 5X, 3E15.8 )
997   FORMAT( 3X, 5E15.8 )
998   FORMAT( 5X, 6E15.8 )
999   FORMAT( 20X, 5E15.8 )
C
      STOP
      END
      SUBROUTINE GETDAT( TIME )
C
      DIMENSION ACC(3), VOLD(3)
     1         , VEC1(3), VEC2(3)
C
      COMMON / RDRDAT / XNO(5),RLAT(5),RLONG(5),RHT(5),NRDR
      COMMON / EARTH / GRAV(3), AGMX(3,3)
      COMMON / RSTATE / RBRI(3),VBRI(3),ABRI(3),THTI(3)
     1                 ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                 ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
      COMMON / RDMEAS / AZM(5), ELM(5), RNGM(5)
C
```

```
      DATA [redacted]D /[redacted]/
      DATA VOLD / 3*0.0 /
C
      CALL ZEROM ( RNGM, 5, 1 )
      READ( 2, 901, END=100 ) RTIME
      READ( 2, 901 )(AZM(I), ELM(I), RNGM(I), I = 1, NRDR)
100   CONTINUE
      READ( 3, 901 ) XMTIME
      READ( 3, 901 ) (ABRI(I), I = 1, 3)
      READ( 3, 901 ) (THTI(I), I = 1, 3)
      READ( 3, 901 ) (OMEG(I), I = 1, 3)
      IF ( TIME.GT.125. ) RNGM(3) = 0.0
C
      IF ( THTI(1).LT.0.0 ) THTI(1) = 360. + THTI(1)
      OOSAMP = 1.0/0.96
      CALL SMLT ( OOSAMP, ABRI, ABRI, 3, 1 )
C
      RETURN
C
901   FORMAT( 5X, 3E15.8 )
C
      END
```

```
      BLOCK DATA
C
      COMMON / TIMDAT / TIME, TMAX, HSTEP, TSAMP, TSTOP
      COMMON / ITYPE / ISMOOTH
      COMMON / IREFOUT / IASSESS
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
      COMMON / LAUCOR / OLATD, OLONG, OHT
     1                 ,CM50EF(3,3)
      COMMON / RDRDAT / XNO(5),RLAT(5),RLONG(5),RHT(5),NRDR
      COMMON / APRIOR / ER(3), EV(3)
     1                 ,ETLT(3),EAB(3),EASF(3),EGRV(3),EX(3)
     2                 ,ERDR(3)
      COMMON / NOISES / R(3), G(30,24), Q(24,24), NP
      COMMON / FPARAM / TAUT(3),TAUA(3),TAUS(3),TAUG(3),TAUX(3),TAUR(3)
     1                 ,UT(3),UA(3),US(3),UG(3),UX(3),UR(3)
      COMMON / LINFMT / F(30,30), N
      COMMON / LINHMT / H(15,30), NRMEAS
      COMMON / METEOR / TALT(400),TSRHO(400),TSP(400),TSSUND(400)
     1                                               ,TVWX(400),TVWY(400),NALT
     2                                               ,USRHO(400),USP(400),USSUND(400)
     3                                               ,UVWX(400),UVWY(400)
C
      DATA TIME, TMAX, HSTEP, TSAMP / 0.0, 523., 1.0, 1.0 /
C
      DATA ISMOOTH / 1 /
      DATA IASSESS / 0 /
C
      DATA NALT / 400 /
C
      DATA N / 30 /
      DATA NP / 24 /
C     DATA NRMEAS / 5 /
      DATA NRMEAS / 15 /
      DATA F / 900*0.0 /
      DATA H / 450*0.0 /
      DATA G / 720*0.0 /
      DATA Q / 576*0.0 /
C
      DATA CRAD / 57.295779 /
      DATA AGRAV, HRSEC, PERCNT, XMRAD / 32.174, 3600., 100., 1000. /
      DATA RE, FLAT, OMEGE, XMU, XJ2 / 20925606.,298.257224,.7292155E-4
CFISCHDATA RE, FLAT, OMEGE, XMU, XJ2 / 20925741.,298.3,0.7292115E-4
     1, 1.4076468E+16, .10827E-2 /
C
      DATA ER / 2*10.0,10.0 /
      DATA EV / 2*1.0,1.0 /
      DATA ETLT / 3*2.E-2 /
      DATA EAB / 3*0.1 /
      DATA EASF / 3*0.1 /
      DATA EGRV / 3*0.0 /
      DATA EX / 3*1.0E+0 /
      DATA ERDR / 1.E-2,1.E-1,1.E+2 /
C
      DATA TAUT / 3*100. /
      DATA TAUA / 3*100. /
      DATA TAUS / 3*100. /
      DATA TAUG / 3*200. /
      DATA TAUX / 3*200. /
      DATA TAUR / 3*400. /
C
      DATA UT / 3*2.E-2 /
      DATA UA / 3*0.1 /
      DATA US / 3*0.1 /
      DATA UG / 3*0.0 /
      DAT[redacted] / 3[redacted]E+0 /
```

```
 [illegible]ATA [illegible] 1.[illegible].E-[illegible]E+2[illegible]
C
      DATA NRDR / 5 /
      DATA XNO / 4*0.0, 0.0000 /
      DATA RLAT / 28.226391,28.463168,28.625932,26.983000,28.424700 /
      DATA RLONG/-80.599287,-80.583111,-80.682805,-80.108204,-80.664399/
      DATA RHT / -44., -49., -58., -21., -56. /
C61CFEDATA RLAT / 28.226553,28.528885,28.626090,26.615781,32.348051 /
C61CFEDATA RLONG/-80.599293,-80.590225,-80.682808,-78.347837,-64.653613/
C61CFEDATA RHT / 65.,37.,52.,63.,75. /
C
      DATA R / 1.E-4,1.E-4,1.E+4 /
C
      DATA OLATD, OLONG, OHT / 28.62721, -80.62079, 0.0 /
C39AFEDATA OLATD, OLONG, OHT / 28.608420, -80.604089, 0.0 /
      DATA CM50EF/.422004,.490734,.762294,.233136,-.871298
     1            ,.431842,.876105,-.004520,-.482099/
C
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE INITIL
C
      DIMENSION CUENED(3,3),VEC(3),TMP1(3,3),TMP2(3,3),TMP3(3,3)
C
      COMMON / LAUCOR / OLATD, OLONG, OHT
     1                 ,CM50EF(3,3)
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
      COMMON / APRIOR / ER(3), EV(3)
     1                  ,ETLT(3), EAB(3), EASF(3), EGRV(3), EX(3)
     2                  ,ERDR(3)
      COMMON / LINFMT / F(30,30), N
      COMMON / PROPAG / X(30), PKP(30,30)
      COMMON / PREUP / XKM(30), PKM(30,30)
      COMMON / UDWORK / RH(30),U(465),PO(465),SF(465),SG(30),RESID,COVZ
      COMMON / RSTATE / RBRI(3),VBRI(3),ABRI(3),THTI(3)
     1                  ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                  ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
C
      DATA CUENED / 2*0.0,-1.0,0.0,1.0,0.0,1.0,2*0.0 /
C
C*** COMPUTE TRANSFORMATION FROM EARTH CENTERED INERTIAL TO BRI
C
      CALL ECPOS ( OLATD, OLONG, OHT, VEC(1), VEC(2), VEC(3) )
      CALL ECPOS ( OLATD, OLONG, 0.0, X0, Y0, Z0 )
      CALL CIEFMX ( 0.0, TMP1 )
      CALL TMATY ( -OLONG, TMP2 )
      CALL MULT ( TMP1, TMP2, TMP3, 3, 3, 3 )
C
      E = 1.0/FLAT
      RDMAG = SQRT( X0**2 + Y0**2 )
      RMAG = SQRT( VEC(1)**2 + VEC(2)**2 + VEC(3)**2 )
      DEV = CRAD*2.*E*(1.-E/2.)*(RDMAG/RMAG)*SIN(3.1415926-OLATD/CRAD)
      OLATC = OLATD - DEV
C
      CALL TMATP ( OLATD, TMP1 )
      CALL MULT ( TMP1, TMP3, TMP2, 3, 3, 3 )
      CALL MULT ( CUENED, TMP2, CIBRI, 3, 3, 3 )
C
C*** TIME ZERO VEHICLE POSITION IN BOOST REFERENCE FRAME
C
      WRITE(*, 901) ((CIBRI(I,J), J = 1, 3), I = 1, 3)
      CALL MULT ( CIBRI, VEC, RBRI, 3, 3, 1 )
      WRITE(*, 901) (RBRI(I), I = 1, 3)
      CALL SWITCH ( RBRI, RIT0, 3, 1 )
C
      VIT0(1) = 0.0
      VIT0(2) = 0.0
      VIT0(3) = OMEGE
C
      CALL MULT ( CIBRI, VIT0, VEC, 3, 3, 1 )
      CALL SKEW ( RBRI, TMP1 )
      CALL SMLT ( -1.0, TMP1, TMP1, 3, 3 )
      CALL MULT ( TMP1, VEC, VBRI, 3, 3, 1 )
      WRITE(*, 901) (VBRI(I), I = 1, 3)
      CALL SWITCH ( VBRI, VIT0, 3, 1 )
C
      CALL CIEFMX ( TIME, TMP1 )
      CALL TRANS ( TMP1, TMP2, 3, 3 )
      CALL MULT ( CIBRI, TMP2, CEFBRI, 3, 3, 3 )
      CALL TRANS ( CEFBRI, CBRIEF, 3, 3 )
C
C*** TIME ZERO SENSED ACCELERATION IN BOOST REFERENCE FRAME
C
      ABRI(1) = 0.0
      ABRI(2) = 0.0
```

```
C
C*** INITIALIZE ERROR COVARIANCES AND U-D FACTOR
C
      DO 100 I = 1, 3
C
C  POSITION AND VELOCITY ERROR IN BRI
C
      XKM(I) = 0.0
      XKM(I+3) = 0.0
      X(I) = XKM(I)
      X(I+3) = XKM(I+3)
      PKM(I,I) = ER(I)**2
      PKM(I+3,I+3) = EV(I)**2
C
C  PLATFORM TILT, ACCELEROMETER BIAS AND SCALE FACTOR
C
      PKM(I+6,I+6) = (ETLT(I)/CRAD)**2
      PKM(I+9,I+9) = (EAB(I))**2
C     PKM(I+12,I+12) = (EX(I)/(CRAD*HRSEC))**2
      PKM(I+12,I+12) = (EASF(I)/PERCNT)**2
C
C  RADAR TRACK AZIMUTH, ELEVATION AND RANGE BIASES
C
      PKM(I+15,I+15) = (ERDR(I))**2
      PKM(I+18,I+18) = (ERDR(I))**2
      PKM(I+21,I+21) = (ERDR(I))**2
      PKM(I+24,I+24) = (ERDR(I))**2
      PKM(I+27,I+27) = (ERDR(I))**2
C
100   CONTINUE
C
      WRITE(*, 901) (PKM(I,I), I = 1, N)
C
      KJ = 0
      DO 200 J = 1, N
      DO 200 K = 1, J
      KJ = KJ + 1
      U(KJ) = PKM(K,J)
200   CONTINUE
C
      CALL COV2UD ( U, N )
C
      RETURN
C
901   FORMAT( 5X, 3E15.8 )
C
      END
```

```
      SUBROUTINE UDTIME
C
      COMMON / TIMDAT / TIME, TMAX, HSTEP, TSAMP, TSTOP
      COMMON / ITYPE / ISMOOTH
      COMMON / LINFMT / F(30,30), N
      COMMON / LINHMT / H(15,30), NMEAS
      COMMON / NOISES / R(3), G(30,24), Q(24,24), NP
      COMMON / PROPAG / X(30), PKP(30,30)
      COMMON / PREUP / XKM(30), PKM(30,30)
      COMMON / RSTATE / RBRI(3),VBRI(3),ABRI(3),THTI(3)
     1                 ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                 ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
      COMMON / UDWORK / RH(30),U(465),PO(465),SF(465),SG(30),RESID,COVZ
      COMMON / UDWRK2 / PHI(30,30),PHIU(30,30),W(30,54)
     1                 ,D(30),B(24),DW(54),V(54)
C
      NPNP = N + NP
      NP1 = N + 1
C
C**   FORM SYSTEM LINEARIZED DYNAMICS MATRICIES  F, G AND Q
C
      CALL SYSTEM
C
C**   COMPUTE STATE TRANSITION MATRIX  PHI
C
      DO 20 I = 1, N
      DO 10 J = 1, N
      PHI(I,J) = F(I,J)*HSTEP
10    CONTINUE
      PHI(I,I) = 1.0 + PHI(I,I)
20    CONTINUE
C
C**   PROPAGATE STATE ERROR ESTIMATES FORWARD IN TIME
C
      DO 22 I = 1, N
      SUM = 0.0
      DO 21 J = 1, N
      SUM = SUM + PHI(I,J)*XKM(J)
21    CONTINUE
      X(I) = SUM
22    CONTINUE
C
      DO 25 I = 1, 3
      RBRI(I) = RBRI(I) + VBRI(I)*HSTEP + 0.5*ANET(I)*HSTEP*HSTEP
      VBRI(I) = VBRI(I) + ANET(I)*HSTEP
25    CONTINUE
C
C**   COMPUTE PRODUCT OF PHI AND U OF THE U-D FACTOR
C
      CALL XPHIU ( PHI, N, N, N, U, N, PHIU, N )
C
      CALL IMBED ( PHIU, N, N, W, N, NPNP, 1, 1 )
      CALL IMBED ( G, N, NP, W, N, NPNP, 1, NP1 )
C
      DO 30 I = 1, NP
      B(I) = Q(I,I)*HSTEP
30    CONTINUE
C
      DO 40 L = 1, N
      J = NP1 - L
      KJ = J*(J+1)/2
      D(J) = U(KJ)
40    CONTINUE
C
      CALL IMBED ( D, N, 1, DW, NPNP, 1, 1, 1 )
      CALL IMBED ( B, NP, 1, DW, NPNP, 1, NP1, 1 )
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C**  WEIGHTED GRAMM-SCHMIDT FACTORIZATION FOR TIME UPDATE
C
      CALL WGS ( W, N, N, NPNP, DW, U, V )
C
      CALL UD2COV ( U, PO, N )
C
      IJ = 0
      DO 100 J = 1, N
      DO 100 I = 1, J
      IJ = IJ + 1
      PKP(I,J) = PO(IJ)
      PKP(J,I) = PKP(I,J)
100   CONTINUE
C
      CALL SWITCH ( X, XKM, N, 1 )
C
      RETURN
C
991   FORMAT( 3X, 9E8.2 )
992   FORMAT ( / )
C
      END
```

```
      SUBROUTINE RADAR
C
      DIMENSION RRDR(3),CEFLLR(3,3),DR(3),UTEMP(465)
     1          ,C1(3,3),VC(3),TMP(3,3),TMP1(3,3)
C
      COMMON / TIMDAT / TIME, TMAX, HSTEP, TSAMP, TSTOP
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
      COMMON / RSTATE / RBRI(3),VBRI(3),ABRI(3),THTI(3)
     1                 ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                 ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
      COMMON / NOISES / R(3), G(30,24), Q(24,24), NP
      COMMON / LINFMT / F(30,30), N
      COMMON / LINHMT / H(15,30), NMEAS
      COMMON / RDRDAT / XNO(5),RLAT(5), RLONG(5), RHT(5), NR
      COMMON / RDMEAS / AZM(5), ELM(5), RNGM(5)
      COMMON / RDREST / AZHAT(5), ELHAT(5), RNGHAT(5)
      COMMON / PREUP / XKM(30), PKM(30,30)
      COMMON / PROPAG / X(30), PKP(30,30)
      COMMON / UDWORK / RH(30),U(465),PO(465),SF(465),SG(30),RESID,COVZ
C
      DATA IAZ, IEL, IRG / 16, 17, 18 /
      DATA C1 / 2*0.0,1.0,1.0,3*0.0,1.0,0.0 /
      DATA CS / 9.8356796E+8 /
C
      KMEAS = 0
C
C  COMPUTE VEHICLE POSITION IN EARTH CENTERED EARTH FIXED COORDINATES
C
      CALL CIEFMX ( TIME, TMP )
      CALL TRANS ( TMP, TMP1, 3, 3 )
      CALL MULT ( CIBRI, TMP1, CEFBRI, 3, 3, 3 )
      CALL TRANS ( CEFBRI, CBRIEF, 3, 3 )
C
C  PROCESS MEASUREMENTS FOR EACH RADAR
C
      DO 300 IR = 1, NR
C
      DO 100 I = 1, 3
      DR(I) = XKM(I)
100   CONTINUE
C
      CALL ADD ( RBRI, DR, DR, 3, 1 )
      CALL MULT ( CBRIEF, DR, REF, 3, 3, 1 )
      CALL ECPOS(RLAT(IR),RLONG(IR),RHT(IR),RRDR(1),RRDR(2),RRDR(3))
      CALL SUBT ( REF, RRDR, VC, 3, 1)
C
      CALL SUBT ( VBRI, VIT0, RRDR, 3, 1 )
      CALL MULT ( CBRIEF, RRDR, DR, 3, 3, 1 )
      CALL INNER ( VC, DR, RRATE, 3 )
      RVCORR = -RRATE/CS
C
      CALL TMATY ( -RLONG(IR), TMP1 )
      CALL TMATP ( RLAT(IR), CEFLLR )
      CALL MULT ( CEFLLR, TMP1, TMP, 3, 3, 3 )
      CALL MULT ( C1, TMP, CEFLLR, 3, 3, 3 )
      CALL MULT ( CEFLLR, VC, DR, 3, 3, 1 )
      CALL TRANS ( CEFBRI, TMP1, 3, 3 )
      CALL MULT ( CEFLLR, TMP1, TMP, 3, 3, 3 )
C
      RNGES = DR(1)*DR(1) + DR(2)*DR(2) + DR(3)*DR(3)
      RNGH = SQRT(RNGES)
      XYS = RNGES - DR(3)*DR(3)
      XY = SQRT(XYS)
      AZHAT(IR) = CRAD*ATAN2(DR(1),DR(2)) + XKM(3*(IR-1) + IAZ)
      ELH = CRAD*ATAN(DR(3)/XY)
      CALL DEFRAC ( IR, ELH, RNGH, DEL, DRANGE )
```

```
      ELHA(IR) = ELEV + XKM(3*(IR-1) + IEL) + DELE
      RNGHAT(IR) = RNGH + RVCORR + XKM(3*(IR-1) + IRG) + DRANGE
C
C   MEASUREMENT LINEARIZATION - AZIMUTH
C
      VC(1) = CRAD*DR(2)/XYS
      VC(2) = -CRAD*DR(1)/XYS
      VC(3) = 0.0
C
      H(3*(IR-1)+1,1)=VC(1)*TMP(1,1)+VC(2)*TMP(2,1)+VC(3)*TMP(3,1)
      H(3*(IR-1)+1,2)=VC(1)*TMP(1,2)+VC(2)*TMP(2,2)+VC(3)*TMP(3,2)
      H(3*(IR-1)+1,3)=VC(1)*TMP(1,3)+VC(2)*TMP(2,3)+VC(3)*TMP(3,3)
C
      H(3*(IR-1)+1,3*(IR-1)+IAZ) = 1.0
C
C   MEASUREMENT LINEARIZATION - ELEVATION
C
      VC(1) = -CRAD*DR(1)*DR(3)/(RNGES*XY)
      VC(2) = -CRAD*DR(2)*DR(3)/(RNGES*XY)
      VC(3) = CRAD*XY/RNGES
C
      H(3*(IR-1)+2,1)=VC(1)*TMP(1,1)+VC(2)*TMP(2,1)+VC(3)*TMP(3,1)
      H(3*(IR-1)+2,2)=VC(1)*TMP(1,2)+VC(2)*TMP(2,2)+VC(3)*TMP(3,2)
      H(3*(IR-1)+2,3)=VC(1)*TMP(1,3)+VC(2)*TMP(2,3)+VC(3)*TMP(3,3)
C
      H(3*(IR-1)+2,3*(IR-1)+IEL) = 1.0
C
C   MEASUREMENT LINEARIZATION - RANGE
C
      VC(1) = DR(1)/RNGHAT(IR)
      VC(2) = DR(2)/RNGHAT(IR)
      VC(3) = DR(3)/RNGHAT(IR)
C
      H(3*(IR-1)+3,1)=VC(1)*TMP(1,1)+VC(2)*TMP(2,1)+VC(3)*TMP(3,1)
      H(3*(IR-1)+3,2)=VC(1)*TMP(1,2)+VC(2)*TMP(2,2)+VC(3)*TMP(3,2)
      H(3*(IR-1)+3,3)=VC(1)*TMP(1,3)+VC(2)*TMP(2,3)+VC(3)*TMP(3,3)
C
      H(3*(IR-1)+3,3*(IR-1)+IRG) = 1.0
C
C  PROCESS AZIMUTH, ELEVATION AND RANGE MEASUREMENTS
C
      DO 200 IM = 1, 3
C
      KMEAS = KMEAS + 1
C
      IF( RNGM(IR).EQ.0.0 ) THEN
      RESID = 0.0
      COVZ = 0.0
      CALL ZEROM ( RH, N, 1 )
      ELSE
C
C  UDU**T FACTORED COVARIANCE UPDATE
C
      DO 140 J = 1, N
      RH(J) = H(KMEAS,J)
140   CONTINUE
C
      CALL SWITCH ( U, UTEMP, 465, 1 )
      RR = R(IM)
      ALPHA = -1.0
      CALL UDMEAS( U, N, RR, RH, SF, SG, ALPHA )
C
      IF ( IM.EQ.1 ) THEN
C
      RESID = AZM(IR) - AZHAT(IR)
C
```

```
      ELSE
C
         IF ( IM.EQ.2 ) THEN
C
         RESID = ELM(IR) - ELHAT(IR)
C
         ELSE
C
         RESID = RNGM(IR) - RNGHAT(IR)
C
         END IF
C
      END IF
C
      COVZ = ALPHA
C
      IF ( ABS(RESID).GT.(6.*SQRT(COVZ)) ) THEN
C
      COVZ = 0.0
      CALL SWITCH ( UTEMP, U, 465, 1 )
      GO TO 190
C
      ELSE
      END IF
C
      DO 150 J = 1, N
      XKM(J) = XKM(J) + (SG(J)/ALPHA)*RESID
150   CONTINUE
C
      END IF
C
C  CONVERT U-D FACTORS TO COVARIANCE MATRICES FOR OUTPUT
C
      CALL UD2COV( U, PO, N )
C
      IJ = 0
      DO 160 J = 1, N
      DO 160 I = 1, J
      IJ = IJ + 1
      PKM(I,J) = PO(IJ)
      PKM(J,I) = PKM(I,J)
160   CONTINUE
C
190   CONTINUE
C
      CALL OUTPUT
C
200   CONTINUE
C
300   CONTINUE
C
      RETURN
C
995   FORMAT( 3X, 5E15.8 )
998   FORMAT( 7X, 4E15.8 )
997   FORMAT( 5X, 3E15.8 )
C
      END
```

```
C
      COMMON / TIMDAT / TIME, TMAX, HSTEP, TSAMP, TSTOP
      COMMON / ITYPE / ISMOOTH
      COMMON / PREUP / XKM(30), PKM(30,30)
      COMMON / LINFMT / F(30,30), N
      COMMON / UDWORK / RH(30),U(465),PO(465),SF(465),SG(30),RESID,COVZ
      COMMON / SMOWRK / TMP(30,24), GT(24,30), GQGT(30,30)
      COMMON / LINHMT / H(15,30), NMEAS
      COMMON / NOISES / R(3), G(30,24), Q(24,24), NP
      COMMON / RSTATE / RBRI(3),VBRI(3),ABRI(3),THTI(3)
     1                  ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                  ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
C
      DATA KO / 0 /
      DATA TLAST / 0.0 /
C
      KO = KO + 1
C
      WRITE(6) TIME, TLAST, KO, NMEAS
      WRITE(6) (XKM(I), I = 1, N)
      WRITE(6) (PKM(I,I), I = 1, N)
C     WRITE(6, 902) (RH(I), I = 1, N)
      WRITE(6) COVZ
      WRITE(6) RESID
C
      IF ( KO - NMEAS ) 20, 10, 10
   10 CONTINUE
C
      WRITE(*, 901) TIME, TLAST, KO, NMEAS
C
      IF ( ISMOOTH.EQ.1 ) THEN
C
      CALL TRANS ( G, GT, N, NP )
      CALL MULT ( Q, GT, TMP, NP, NP, N )
      CALL MULT ( G, TMP, GQGT, N, NP, N )
C
      WRITE(7, 903) TIME
      WRITE(7, 903) (ABRI(I), I = 1, 3)
      WRITE(7, 903) (THTI(I), I = 1, 3)
      WRITE(7, 903) (OMEG(I), I = 1, 3)
      WRITE(7, 903) (RBRI(I), I = 1, 3)
      WRITE(7, 903) (VBRI(I), I = 1, 3)
      WRITE(7, 903) ((CIBRI(I,J), J = 1, 3), I = 1, 3)
      WRITE(7, 903) ((CBRIEF(I,J), J = 1, 3), I = 1, 3)
C
      WRITE(8) TIME, TLAST, KO, NMEAS
      WRITE(8) (XKM(I), I = 1, N)
      WRITE(8) ((PKM(I,J), J = 1, N), I = 1, N)
      WRITE(8) ((F(I,J), J = 1, N), I = 1, N)
      WRITE(8) ((GQGT(I,J), J = 1, N), I = 1, N)
C
      ELSE
      END IF
C
      TLAST = TIME
C
      KO = 0
C
   20 CONTINUE
C
      RETURN
C
  901 FORMAT( 2E15.8, 2I5 )
  902 FORMAT( 5E15.8 )
  903 FORMAT( 4X, 3E15.8 )
```

C END

```
C     [redacted]UBRO[redacted]E A[redacted]S
      DIMENSION UNIT(3,3),CBI(3,3),CIB(3,3),VB(3),AB(3)
     1          ,VEC1(3),VEC2(3),VEC3(3),VEC4(3),VEC5(3)
     2          ,TMP1(3,3),TMP2(3,3),TMP3(3,3),TMP4(3,3),TMP5(3,3)
     3          ,VECX(3),VECY(3),VECZ(3),TMPX(3,3),TMPY(3,3),TMPZ(3,3)
     4          ,TMPA(3,3),CCIB(3,3),VBR(3),RBR(3),ACC(3)
     5          ,VW(3),VR(3),ARRAY(90),BRRAY(90)
     6          ,PVW(3,3),C1(3,3)
     7          ,VEC6(6),VEC7(6)
C
      COMMON / TIMDAT / TIME, TMAX, HSTEP, TSAMP, TSTOP
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
      COMMON / LAUCOR / OLATD, OLONG, OHT
     1                 ,CM50EF(3,3)
      COMMON / PREUP / XKM(30), PKM(30,30)
      COMMON / EARTH / GRAV(3), AGMX(3,3)
      COMMON / RSTATE / RBRI(3),VBRI(3),ABRI(3),THTI(3)
     1                 ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                 ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
      COMMON / METEOR / TALT(400),TSRHO(400),TSP(400),TSSUND(400)
     1                                        ,TVWX(400),TVWY(400),NALT
     2                                        ,USRHO(400),USP(400),USSUND(400)
     3                                        ,UVWX(400),UVWY(400)
C
      DATA UNIT / 1.,3*0.,1.,3*0.,1. /
      DATA NR / 16 /
      DATA RHO, PS, VSOUND / .002378, 2116., 1117. /
      DATA VWX, VWY / 2*0.0 /
      DATA PRHOPH, PSPH, PVSPH / 3*0.0 /
      DATA PVWXPH, PVWYPH / 2*0.0 /
      DATA PVW / 9*0.0 /
      DATA C1 / 2*0.0,1.0,1.0,3*0.0,1.0,0.0 /
C
      CALL ZEROM ( ARRAY, 90, 1 )
      CALL ZEROM ( BRRAY, 90, 1 )
C
      CALL CBIMX ( THTI, CBI )
      CALL TRANS ( CBI, CIB, 3, 3 )
C
      DO 20 I = 1, 3
      DO 10 J = 1, 3
      TMP1(I,J) = PKM(I,J)
      TMP2(I,J) = PKM(I+3,J+3)
      TMP3(I,J) = PKM(I+6,J+6)
      TMP4(I,J) = PKM(I+9,J+9)
      TMP5(I,J) = PKM(I+12,J+12)
10    CONTINUE
      RBR(I) = RBRI(I) + XKM(I)
      VBR(I) = VBRI(I) + XKM(I+3)
      VEC2(I) = VBR(I)
      VEC3(I) = XKM(I+6)
      VEC4(I) = XKM(I+9)
      VEC5(I) = XKM(I+12)
20    CONTINUE
C
      CALL SKEW ( VEC3, TMPX )
      CALL ADD ( UNIT, TMPX, TMPX, 3, 3 )
      CALL MULT ( CIB, TMPX, CCIB, 3, 3, 3 )
      ARRAY(1) = TIME
      BRRAY(1) = 0.0
C
C**   VEHICLE POSITION
C
      ARRAY(2) = RBR(1)
```

```
      ARRAY(3) = RBR(2)
      ARRAY(4) = RBR(3)
      BRRAY(2) = SQRT( PKM(1,1) )
      BRRAY(3) = SQRT( PKM(2,2) )
      BRRAY(4) = SQRT( PKM(3,3) )
C
      CALL COOR ( REF(1), REF(2), REF(3), XLAT, XLONG, ALT )
      RM = SQRT ( REF(1)**2 + REF(2)**2 + REF(3)**2 )
      XYSQR = REF(1)**2 + REF(2)**2
      OMECC = 1.0 - 1.0/FLAT
      ALT = ( 1.0 - RE*OMECC/SQRT( OMECC**2*XYSQR+REF(3)**2 ) )*RM
      ARRAY(49) = XLONG
      ARRAY(50) = XLAT
      ARRAY(51) = ALT
      ARRAY(54) = RM
      BRRAY(49) = CRAD*SQRT( PKM(2,2) )/(RM*COS( XLAT/CRAD ))
      BRRAY(50) = CRAD*SQRT( PKM(1,1) )/RM
      RMS = RM*RM
      CALL MULT ( TMP1, RBR, VECX, 3, 3, 1 )
      CALL INNER ( RBR, VECX, SIGRMS, 3 )
      SIGRM = SQRT ( SIGRMS/RMS )
      BRRAY(51) = SIGRM
      BRRAY(54) = SIGRM
C
      CALL ECPOS ( OLATD, OLONG, OHT, VECZ(1), VECZ(2), VECZ(3) )
      DO 30 I = 1, 3
      VECX(I) = REF(I) - VECZ(I)
30    CONTINUE
      RS = SQRT( VECX(1)**2 + VECX(2)**2 + VECX(3)**2 )
      ARRAY(58) = RS
      CALL MULT ( TMP1, VECX, VECY, 3, 3, 1 )
      CALL INNER ( VECX, VECY, SIGRSS, 3 )
      IF ( RS.NE.0.0 ) THEN
      SIGRSS = SIGRSS/(RS*RS)
      ELSE
      SIGRSS = 0.0
      END IF
      BRRAY(58) = SQRT ( SIGRSS )
C
C**   BOOST REFERENCE VELOCITY AND UNCERTAINTY BOUNDS
C
      VMI = SQRT ( VEC2(1)**2 + VEC2(2)**2 + VEC2(3)**2 )
      ARRAY(5) = VBR(1)
      ARRAY(6) = VBR(2)
      ARRAY(7) = VBR(3)
      ARRAY(55) = VMI
      BRRAY(5) = SQRT( PKM(4,4) )
      BRRAY(6) = SQRT( PKM(5,5) )
      BRRAY(7) = SQRT( PKM(6,6) )
      VMIS = VMI*VMI
      CALL MULT ( TMP2, VBR, VECX, 3, 3, 1 )
      CALL INNER ( VBR, VECX, SIGVMS, 3 )
      IF ( VMI.NE.0.0 ) THEN
      SIGVM = SQRT ( SIGVMS/VMIS )
      ELSE
      SIGVM = 0.0
      END IF
      BRRAY(55) = SIGVM
C
C**   BODY REFERENCED VELOCITY AND UNCERTAINTY BOUNDS
C
      CALL ZEROM ( VEC1, 3, 1 )
      VEC1(3) = OMEGE
      CALL SKEW ( VEC1, TMPX )
      CALL MULT ( TMPX, REF, VEC1, 3, 3, 1 )
      CALL MULT ( CIBRI, VEC1, VECX, 3, 3, 1 )
```

```
      CALL [redacted] ( [redacted], VB[redacted]VEC[redacted] 1 [redacted]
      CALL MULT ( CCIB, VECY, VB, 3, 3, 1 )
C
      VT = SQRT ( VB(1)**2 + VB(2)**2 + VB(3)**2 )
      ARRAY(20) = VB(1)
      ARRAY(21) = VB(2)
      ARRAY(22) = VB(3)
C
C*    INERTIAL VELOCITY
C
      CALL MULT ( TMP2, CBI, TMPY, 3, 3, 3 )
      CALL MULT ( CIB, TMPY, TMPX, 3, 3, 3 )
C
C*    PLATFORM TILT
C
      CALL SKEW ( VEC2, TMPY )
      CALL MULT ( CIB, TMPY, TMPZ, 3, 3, 3 )
      CALL TRANS ( TMPZ, TMPA, 3, 3 )
      CALL MULT ( TMPZ, TMP3, TMPY, 3, 3, 3 )
      CALL MULT ( TMPY, TMPA, TMPZ, 3, 3, 3 )
C
      CALL ADD ( TMPX, TMPZ, TMPX, 3, 3 )
      BRRAY(20) = SQRT( TMPX(1,1) )
      BRRAY(21) = SQRT( TMPX(2,2) )
      BRRAY(22) = SQRT( TMPX(3,3) )
C
C** ANGLE OF ATTACK, SIDESLIP AND VELOCITY WRT AIR MASS W/UNCERTAIN
C
      IF ( ALT.LT.400000. ) THEN
C
      CALL INTRP1 ( ALT, TALT, TSRHO, NALT, RHO, PRHOPH )
      CALL INTRP1 ( ALT, TALT, TSP, NALT, PS, PPSPH )
      CALL INTRP1 ( ALT, TALT, TSSUND, NALT, VSOUND, PVSPH )
      CALL INTRP1 ( ALT, TALT, TVWX, NALT, VWX, PVWXPH )
      CALL INTRP1 ( ALT, TALT, TVWY, NALT, VWY, PVWYPH )
      CALL INTRP1 ( ALT, TALT, USRHO, NALT, UNCRHO, SLOPE )
      CALL INTRP1 ( ALT, TALT, USP, NALT, UNCSP, SLOPE )
      CALL INTRP1 ( ALT, TALT, USSUND, NALT, UNCVS, SLOPE )
      CALL INTRP1 ( ALT, TALT, UVWX, NALT, UNCVWX, SLOPE )
      CALL INTRP1 ( ALT, TALT, UVWY, NALT, UNCVWY, SLOPE )
C
      ELSE
      END IF
C
      VW(1) = VWX
      VW(2) = VWY
      VW(3) = 0.0
      PVW(1,1) = UNCVWX
      PVW(2,2) = UNCVWY
C
      ATEMP = (VSOUND/49.02)**2
      ARRAY(79) = ATEMP
      ARRAY(80) = PS
      ARRAY(81) = RHO
      BRRAY(79) = UNCVS*SQRT( ATEMP )*(2./49.02)
      BRRAY(80) = UNCSP
      BRRAY(81) = UNCRHO
C
      CALL TMATY ( -XLONG, TMPX )
      CALL TMATP ( XLAT, TMPY )
      CALL MULT ( TMPY, TMPX, TMPZ, 3, 3, 3 )
      CALL TRANS ( TMPZ, TMPX, 3, 3 )
      CALL MULT ( CEFBRI, TMPX, TMPY, 3, 3, 3 )
      CALL MULT ( TMPY, VW, VECX, 3, 3, 1 )
      CALL SUBT ( VB, VECX, VR, 3, 1 )
C
```

```
      VMA = SQRT ( VR(1)**2 + VR(2)**2 + VR(3)**2 )
      VMAS = VMA*VMA
      ALPHA = CRAD*ATAN( VR(3)/VR(1) )
      BETA = CRAD*ASIN( VR(2)/VMA )
      ARRAY(68) = VMA
      ARRAY(72) = ALPHA
      ARRAY(84) = BETA
C
      CALL TRANS ( TMPY, TMPZ, 3, 3 )
      CALL MULT ( PVW, TMPZ, TMPA, 3, 3, 3 )
      CALL MULT ( TMPY, TMPA, TMPZ, 3, 3, 3 )
      CALL ADD ( TMPX, TMPZ, TMPX, 3, 3 )
      BRRAY(68) = SQRT( TMPX(1,1) + TMPX(2,2) + TMPX(3,3) )
      V13S = VR(1)**2 + VR(3)**2
      V13 = SQRT ( V13S )
      VECX(1) = -VR(3)/V13S
      VECX(2) = 0.0
      VECX(3) = VR(1)/V13S
      CALL MULT ( TMPX, VECX, VECY, 3, 3, 1)
      CALL INNER ( VECX, VECY, SIGAS, 3 )
      BRRAY(72) = CRAD*SQRT ( SIGAS )
C
      IF ( V13.NE.0.0 ) THEN
C
      VECX(1) = VR(2)*VR(1)/(VMAS*V13)
      VECX(2) = VR(2)*VR(2)/(VMAS*V13)
      VECX(3) = VR(2)*VR(3)/(VMAS*V13)
      CALL MULT ( TMPX, VECX, VECY, 3, 3, 1 )
      CALL INNER ( VECX, VECY, SIGBS, 3 )
      VECX(1) = 0.0
      VECX(2) = 1.0/V13
      VECX(3) = 0.0
      CALL MULT ( TMPX, VECX, VECY, 3, 3, 1 )
      CALL INNER ( VECX, VECY, SIGS, 3 )
      SIGB = SQRT ( SIGBS + SIGS )
      BRRAY(84) = CRAD*SIGB
C
      ELSE
      END IF
C
C**  MACH NUMBER AND DYNAMIC PRESSURE W/UNCERTAINTY BOUNDS
C
      PDYNMC = 0.5*RHO*VMAS
      XMACH = VMA/VSOUND
      ARRAY(74) = PDYNMC
      ARRAY(77) = XMACH
      CALL MULT ( TMPX, VR, VECX, 3, 3, 1 )
      CALL INNER ( VR, VECX, SIGVS, 3 )
      SIGQS = RHO*RHO*SIGVS
      SIGS = (0.5*VMAS*PRHOPH*SIGRM)**2
      SIGQS = SIGQS + SIGS
      BRRAY(74) = SQRT ( SIGQS )
      VSS = VSOUND*VSOUND
      SIGMS = SIGVS/VSS
      SIGS = (VMA*PVSPH*SIGRM/VSS)**2
      SIGMS = SIGMS + SIGS
      BRRAY(84) = SQRT ( SIGMS )
C
C**  Q-ALPHA AND Q-BETA WITH UNCERTAINTY BOUNDS
C
      QALPHA = PDYNMC*ALPHA
      QBETA = PDYNMC*BETA
      ARRAY(75) = QALPHA
      ARRAY(76) = QBETA
      ALPHAS = ALPHA*ALPHA
      BETAS = BETA*BETA
```

```
[redacted]DYN[redacted]PDYN[redacted]DYN[redacted]
      BRRAY(75) = SQRT(ALPHAS*BRRAY(74)**2 + PDYNS*BRRAY(72)**2)
      BRRAY(76) = SQRT(BETAS*BRRAY(74)**2 + PDYNS*BRRAY(84)**2)
C
C**  VEHICLE ATTITUDE AND UNCERTAINTY BOUNDS
C
      CALL SMLT ( CRAD, VEC3, VEC3, 3, 1 )
      CALL ADD ( THTI, VEC3, VEC3, 3, 1 )
      DO 40 I = 1, 3
      VECZ(I) = CRAD*SQRT(TMP3(I,I))
40    CONTINUE
      ARRAY(85) = VEC3(3)
      ARRAY(86) = VEC3(2)
      ARRAY(87) = VEC3(1)
      BRRAY(85) = VECZ(3)
      BRRAY(86) = VECZ(2)
      BRRAY(87) = VECZ(1)
C
C**  (B) AND (I) REFERENCED ACCELERATION AND UNCERTAINTY BOUNDS
C
      CALL ADD ( ABRI, VEC4, VEC1, 3, 1 )
      CALL DIAG ( ABRI, 3, TMPX )
      CALL MULT ( TMPX, VEC5, AB, 3, 3, 1 )
      CALL ADD ( AB, VEC1, VEC1, 3, 1 )
      ARRAY(8) = VEC1(1)
      ARRAY(9) = VEC1(2)
      ARRAY(10) = VEC1(3)
C
      CALL MULT ( CCIB, VEC1, AB, 3, 3, 1 )
C
CRFTSTAT = SQRT( ABRI(1)**2 + ABRI(2)**2 + ABRI(3)**2 )
      AT = SQRT ( AB(1)**2 + AB(2)**2 + AB(3)**2 )
      ARRAY(23) = AB(1)
      ARRAY(24) = AB(2)
      ARRAY(25) = AB(3)
      ARRAY(83) = AT/AGRAV
C
C*   ACCELEROMETER BIAS
C
C     TMP4
C
C*   ACCELEROMETER SCALE FACTOR
C
      CALL DIAG ( ABRI, 3, TMPZ )
      CALL MULT ( TMPZ, TMP5, TMPX, 3, 3, 3 )
C
C*   PLATFORM TILT
C
      CALL SKEW ( ABRI, TMPZ )
      CALL MULT ( TMPZ, TMP3, TMPY, 3, 3, 3 )
C
      CALL ADD ( TMP4, TMPX, TMPZ, 3, 3 )
      CALL ADD ( TMPY, TMPZ, TMPX, 3, 3 )
      BRRAY(8) = SQRT( TMPX(1,1) )
      BRRAY(9) = SQRT( TMPX(2,2) )
      BRRAY(10) = SQRT( TMPX(3,3) )
C
      CALL MULT ( CIB, TMPX, TMPY, 3, 3, 3 )
      CALL MULT ( TMPY, CBI, TMPX, 3, 3, 3 )
      BRRAY(23) = SQRT( TMPX(1,1) )
      BRRAY(24) = SQRT( TMPX(2,2) )
      BRRAY(25) = SQRT( TMPX(3,3) )
      BRRAY(83) = SQRT(BRRAY(23)**2+BRRAY(24)**2+BRRAY(25)**2)/AGRAV
C
C**  FLIGHT PATH ANGLE
C
```

```
      CALL INNER ( RBR, VBR, RDOTV, 3 )
      IF ( VMI.NE.0.0 ) THEN
      GAMMA = CRAD*ASIN( RDOTV/(RM*VMI) )
      ELSE
      GAMMA = 0.0
      END IF
      ARRAY(56) = GAMMA
C
      B = RM*VMI
      RFAC = RDOTV/B
      RFAC1 = RFAC*VMI/RM
      RFAC2 = RFAC*RM/VMI
      DENO = SQRT( B*B - RDOTV*RDOTV )
C
      DO 50 I = 1, 3
      VEC6(I) = ( VBR(I) + RFAC1*RBR(I) )/DENO
      VEC6(I+3) = ( RBR(I) + RFAC2*VBR(I) )/DENO
50    CONTINUE
C
      DO 70 I = 1, 6
      TEMP = 0.0
         DO 60 J = 1, 6
         TEMP = TEMP + PKM(I,J)*VEC6(J)
60       CONTINUE
      VEC7(I) = TEMP
70    CONTINUE
C
      SIGGMS = 0.0
      DO 80 I = 1, 6
      SIGGMS = SIGGMS + VEC6(I)*VEC7(I)
80    CONTINUE
      SIGGM = CRAD*SQRT ( SIGGMS )
      BRRAY(56) = SIGGM
C
      CALL TMATY ( -XLONG, TMPX )
      CALL TMATP ( XLAT, TMPY )
      CALL MULT ( TMPY, TMPX, TMPZ, 3, 3, 3 )
      CALL MULT ( C1, TMPZ, TMPX, 3, 3, 3 )
      CALL MULT ( CBRIEF, VBR, VECX, 3, 3, 1 )
      CALL MULT ( TMPX, VECX, VECY, 3, 3, 1 )
      IF ( VECY(2).NE.0.0 ) THEN
      SIGMA = CRAD*ATAN2( VECY(1), VECY(2) )
      ELSE
      SIGMA = 90.0
      END IF
C
      ARRAY(57) = SIGMA
C
      DENO = VECY(1)*VECY(1) + VECY(2)*VECY(2)
      VECX(1) = VECY(2)/DENO
      VECX(2) = VECY(1)/DENO
      VECX(3) = 0.0
C
      CALL TRANS ( TMPX, TMPY, 3, 3 )
      CALL MULT ( TMPY, VECX, VECZ, 3, 3, 1 )
      CALL MULT ( TMP2, VECZ, VECY, 3, 3, 1 )
      CALL INNER ( VECX, VECY, SIGSMS, 3 )
      IF ( SIGSMS.GE.0.0 ) THEN
      SIGSM = CRAD*SQRT ( SIGSMS )
      END IF
      BRRAY(57) = SIGSM
C
C** VEHICLE ROTATION RATE
C
      ARRAY(45) = OMEG(3)
      ARRAY(46) = OMEG(2)
```

```
      ARRA[illegible]) = [illegible](1)
C
C** OUTPUT COMPUTED QUANTITIES
C
C     WRITE(9'NR) (ARRAY(I), I = 1, 90)
      WRITE(9, 902) (ARRAY(I), I = 1, 90)
C     WRITE(9, 902) (BRRAY(I), I = 1, 90)
      NR = NR + 1
C
      RETURN
C
901   FORMAT ( 7X, 4E15.8 )
902   FORMAT ( 2X, 6E12.5 )
C
      END
```

```
      SUBROUTINE SYSTEM
C
      DIMENSION UNIT(3,3),TEMP(3,3),VEC(3)
C
      COMMON / TIMDAT / TIME, TMAX, HSTEP, TSAMP, TSTOP
      COMMON / RSTATE / RBRI(3),VBRI(3),ABRI(3),THTI(3)
     1                 ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                 ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
      COMMON / CONST / CRAD, AAGRAV, HRSEC, PERCNT, XMRAD
      COMMON / EARTH / GRAV(3),AGMX(3,3)
      COMMON / FPARAM / TAUT(3),TAUA(3),TAUS(3),TAUG(3),TAUX(3),TAUR(3)
     1                 ,UT(3),UA(3),US(3),UG(3),UX(3),UR(3)
      COMMON / LINFMT / F(30,30), N
      COMMON / NOISES / R(3), G(30,24), Q(24,24), NP
C
      DATA JUMP / 0 /
      DATA UNIT / 1.0, 3*0.0, 1.0, 3*0.0, 1.0 /
C
      IF ( JUMP ) 10, 10, 20
10    CONTINUE
C
C*****  FIXED OR CONSTANT MATRIX ELEMENTS
C
C**  INTEGRATE VELOCITY ERROR INTO POSITION ERROR
C
      CALL IMBED ( UNIT, 3, 3, F, N, N, 1, 4 )
C
C**  ACCELEROMETER BIAS
C
      CALL IMBED ( UNIT, 3, 3, F, N, N, 4, 10 )
C
C**  PROCESS NOISE
C
C*  PLATFORM TILT ERROR
C
      CALL OODIAG ( TAUT, 3, TEMP )
      CALL IMBED ( TEMP, 3, 3, F, N, N, 7, 7 )
C
C*  ACCELEROMETER BIAS
C
      CALL OODIAG ( TAUA, 3, TEMP )
      CALL IMBED ( TEMP, 3, 3, F, N, N, 10, 10 )
C
C*  ACCELEROMETER SCALE FACTOR ERROR
C
      CALL OODIAG ( TAUS, 3, TEMP )
      CALL IMBED ( TEMP, 3, 3, F, N, N, 13, 13 )
C
C*  RADAR AZIMUTH, ELEVATION AND RANGE BIAS
C
      CALL OODIAG ( TAUR, 3, TEMP )
      CALL IMBED ( TEMP, 3, 3, F, N, N, 16, 16 )
      CALL IMBED ( TEMP, 3, 3, F, N, N, 19, 19 )
      CALL IMBED ( TEMP, 3, 3, F, N, N, 22, 22 )
      CALL IMBED ( TEMP, 3, 3, F, N, N, 25, 25 )
      CALL IMBED ( TEMP, 3, 3, F, N, N, 28, 28 )
C
      DO 15 I = 1, 3
C
      Q(I,I) = 2.0*TAUT(I)*(UT(I)/CRAD)**2
      G(I+6,I) = 1.0/TAUT(I)
      Q(I+3,I+3) = 2.0*TAUA(I)*(UA(I))**2
      G(I+9,I+3) = 1.0/TAUA(I)
      Q(I+6,I+6) = 2.0*TAUS(I)*(US(I)/PERCNT)**2
      G(I+12,I+6) = 1.0/TAUS(I)
      Q(I+9,I+9) = 2.0*TAUR(I)*(UR(I))**2
```

```
      [redacted]
      Q(I+12,I+12) = 2.0*TAUR(I)*(UR(I))**2
      G(I+18,I+12) = 1.0/TAUR(I)
      Q(I+15,I+15) = 2.0*TAUR(I)*(UR(I))**2
      G(I+21,I+15) = 1.0/TAUR(I)
      Q(I+18,I+18) = 2.0*TAUR(I)*(UR(I))**2
      G(I+24,I+18) = 1.0/TAUR(I)
      Q(I+21,I+21) = 2.0*TAUR(I)*(UR(I))**2
      G(I+27,I+21) = 1.0/TAUR(I)
C
15    CONTINUE
C
      JUMP = 1
C
20    CONTINUE
C
C*****  TIME VARYING MATRIX ELEMENTS
C
C**  TILT ERROR
C
      CALL SKEW ( ABRI, TEMP )
      CALL SMLT ( -1.0, TEMP, TEMP, 3, 3 )
      CALL IMBED ( TEMP, 3, 3, F, N, N, 4, 7 )
C
C**  GRAVITY ERROR
C
      CALL MULT ( CBRIEF, RBRI, VEC, 3, 3, 1 )
      CALL AGRAV ( VEC )
C     CALL MULT ( CEFBRI, GRAV, GRAVI, 3, 3, 1 )
      DO 40 I = 1, 3
      SUM = 0.0
      DO 30 J = 1, 3
      SUM = SUM + CEFBRI(I,J)*GRAV(J)
30    CONTINUE
      GRAVI(I) = SUM
40    CONTINUE
      CALL ADD ( GRAVI, ABRI, ANET, 3, 1 )
      CALL MULT ( CEFBRI, AGMX, TEMP, 3, 3, 3 )
      CALL MULT ( TEMP, CBRIEF, AGMX, 3, 3, 3 )
      CALL IMBED ( AGMX, 3, 3, F, N, N, 4, 1 )
C
      TFAC = HSTEP/2.0
      CALL SMLT ( TFAC, AGMX, AGMX, 3, 3 )
      CALL IMBED ( AGMX, 3, 3, F, N, N, 1, 1 )
      CALL IMBED ( AGMX, 3, 3, F, N, N, 4, 4 )
C
C**  ACCELEROMETER SCALE FACTOR
C
      CALL DIAG ( ABRI, 3, TEMP )
      CALL IMBED ( TEMP, 3, 3, F, N, N, 4, 13 )
C
      RETURN
C
991   FORMAT ( 3X, 9E8.2 )
C
      END
$
```

```
      SUBROUTINE COV2UD ( U, N )
C
C     IMPLICIT DOUBLE PRECISION ( A-H, O-Z )
C
      DIMENSION U(1)
C
      Z = 0.E0
      ONE = 1.E0
      NONE = 1
C
      JJ = N*(N+1)/2
      NP2 = N + 2
      DO 50 L = 2, N
        J = NP2 - L
        ALPHA = Z
        IF ( U(JJ).GE.Z ) GO TO 10
C       WRITE(*, 100) J, U(JJ)
        U(JJ) = Z
   10   IF ( U(JJ).GT.Z ) ALPHA = ONE/U(JJ)
        JJ = JJ - J
        KK = 0
        KJ = JJ
        JM1 = J - 1
        DO 40 K = 1, JM1
          KJ = KJ + 1
          BETA = U(KJ)
          U(KJ) = ALPHA*U(KJ)
          IJ = JJ
          IK = KK
          DO 30 I = 1, K
            IK = IK + 1
            IJ = IJ + 1
   30       U(IK) = U(IK) - BETA*U(IJ)
   40     KK = KK + K
   50   CONTINUE
      IF ( U(1).GE.Z ) GO TO 60
C     WRITE(*, 100) NONE, U(1)
      U(1) = Z
   60 RETURN
C
  100 FORMAT ( 1H0, 20X,8H AT STEP,I4,16HDIAGONAL ENTRY =,F12.4 )
      END
      SUBROUTINE UDMEAS ( U, N, R, A, F, G, ALPHA )
C
C     IMPLICIT DOUBLE PRECISION ( A-H, O-Z )
C
      DIMENSION U(1), A(1), F(1), G(1)
C     DOUBLE PRECISION SUM, BETA, GAMMA
      LOGICAL IEST
C
      ZERO = 0.E0
      IEST = .FALSE.
      ONE = 1.E0
      NP1 = N + 1
      NP2 = N + 2
      NTOT = N*NP1/2
      IF ( ALPHA.LT.ZERO ) GO TO 3
      SUM = A(NP1)
      DO 1 J = 1, N
    1 SUM = SUM - A(J)*U(NTOT+J)
      U(NTOT+NP1) = SUM
      IEST = .TRUE.
C
    3 JJN = NTOT
      DO 10 L = 2, N
        J = NP2 - L
```

```
      JJ = JJN - J
      SUM = A(J)
      JM1 = J - 1
      DO 5 K = 1, JM1
    5    SUM = SUM + U(JJ+K)*A(K)
      F(J) = SUM
    G(J) = SUM*U(JJN)
   10 JJN = JJ
    F(1) = A(1)
    G(1) = U(1)*F(1)
C
    SUM = R + G(1)*F(1)
    GAMMA = 0
    IF ( SUM.GT.ZERO ) GAMMA = ONE/SUM
    IF ( F(1).NE.ZERO ) U(1) = U(1)*R*GAMMA
C
    KJ = 2
    DO 20 J = 2, N
      BETA = SUM
      TEMP = G(J)
      SUM = SUM + TEMP*F(J)
      P = -F(J)*GAMMA
      JM1 = J - 1
      DO 15 K = 1, JM1
        S = U(KJ)
        U(KJ) = S + P*G(K)
        G(K) = G(K) + TEMP*S
   15     KJ = KJ + 1
    IF ( TEMP.EQ.ZERO ) GO TO 20
      GAMMA = ONE/SUM
      U(KJ) = U(KJ)*BETA*GAMMA
   20 KJ = KJ + 1
    ALPHA = SUM
C
    IF ( .NOT.IEST ) RETURN
    F(NP1) = U(NTOT+NP1)*GAMMA
    DO 30 J = 1, N
   30   U(NTOT+J) = U(NTOT+J) + G(J)*F(NP1)
C
    RETURN
    END
    SUBROUTINE UD2COV ( UIN, POUT, N )
C
C   IMPLICIT DOUBLE PRECISION  ( A-H, O-Z )
C
    DIMENSION UIN(1), POUT(1)
C
    POUT(1) = UIN(1)
    JJ = 1
    DO 20 J = 2, N
      JJL = JJ
      JJ = JJ + J
      POUT(JJ) = UIN(JJ)
      S = POUT(JJ)
      II = 0
      JM1 = J - 1
      DO 20 I = 1, JM1
        II = II + I
        ALPHA = S*UIN(JJL+I)
        IK = II
        DO 10 K = I, JM1
          POUT(IK) = POUT(IK) + ALPHA*UIN(JJL+K)
   10     IK = IK + K
   20   POUT(JJL+I) = ALPHA
C
    RETURN
```

```
      END
      SUBROUTINE XPHIU ( PHI, MAXPHI, IRPHI, ICPHI, U, N, PHIU, MPHIU )
C
C     IMPLICIT DOUBLE PRECISION ( A-H, O-Z )
C
      DIMENSION PHI(MAXPHI, 1), U(1), PHIU(MPHIU, 1)
C     DOUBLE PRECISION SUM
C
      DO 10 I = 1, IRPHI
   10 PHIU(I,1) = PHI(I,1)
C
      NP2 = N + 2
      KJS = N*(N+1)/2
      DO 40 L = 2, N
         J = NP2 - L
         KJS = KJS - J
      JM1 = J - 1
      DO 30 I = 1, IRPHI
      SUM = PHI(I,J)
      IF ( J.LE.ICPHI ) GO TO 15
      SUM = 0.0
      JM1 = ICPHI
   15 DO 20 K = 1, JM1
   20 SUM = SUM + PHI(I,K)*U(KJS+K)
   30 PHIU(I,J) = SUM
   40   CONTINUE
C
      RETURN
      END
      SUBROUTINE WGS ( W, IMAXW, IW, JW, DW, U, V )
C
C     IMPLICIT DOUBLE PRECISION ( A-H, O-Z )
C
      DIMENSION W(IMAXW, 1), DW(1), U(1), V(1)
C
      Z = 0.0
      ONE = 1.0
      IWP2 = IW + 2
      DO 100 L = 2, IW
         J = IWP2 - L
         SUM = Z
         DO 40 K = 1, JW
            V(K) = W(J,K)
            U(K) = DW(K)*V(K)
   40       SUM = V(K)*U(K) + SUM
         W(J,J) = SUM
         DINV = SUM
         JM1 = J - 1
         IF ( SUM.GT.Z ) GO TO 45
C     W(J,.) = 0.0 WHEN DINV = 0.0  (DINV = NORM(W(J,.)**2))
         DO 44 K = 1, JM1
   44       W(J,K) = Z
         GO TO 100
   45    DO 70 K = 1, JM1
            SUM = Z
            DO 50 I = 1, JW
   50          SUM = W(K,I)*U(I) + SUM
            SUM = SUM/DINV
C     DIVIDE HERE (IN PLACE OF RECIPROCAL MULTIPLY) TO AVOID
C     POSSIBLE OVERFLOW
C
            DO 60 I = 1, JW
   60          W(K,I) = W(K,I) - SUM*V(I)
   70       W(J,K) = SUM
  100    CONTINUE
C
```

```
C [redacted] LO[redacted]ART[redacted]W IS[redacted]RAN[redacted]E
C
      SUM = Z
      DO 105 K = 1, JW
  105    SUM = DW(K)*W(1,K)**2 + SUM
      U(1) = SUM
      IJ = 1
      DO 110 J = 2, IW
         DO 110 I = 1, J
            IJ = IJ + 1
  110       U(IJ) = W(J,I)
C
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE REFRAC ( IRDR, EL, RNGH, DELEL, DELRNG )
C
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / RDRDAT / XXNO(5), RLAT(5), RLONG(5), SALT(5), NRDR
C
      DATA A, B, C / 17590., 30.55, 0.0 /
      DATA RO / 6378165. /
      DATA CRAD / 57.295779 /
C
      RANGE = .3048*RNGH
      HSTA = .3048*SALT( IRDR )
      XNO = XXNO( IRDR )
C
      ALT = SQRT( RO**2 + RANGE**2 + 2.*RO*RANGE*SIN( EL/CRAD ) ) - RO
C
      HS0 = 7000.
C
      DO 10 I = 1, 50
C
      HS = A - B*(XNO*1.E+6)*EXP( ( HSTA - C )/HS0 )
C     WRITE(*, 901) HS, HS0
901   FORMAT ( 10X, 2E12.5 )
C
      IF ( ABS( HS - HS0 ).LE. 1.0 ) GO TO 20
C
      HS0 = HS
C
C     WRITE(*, 902) I
902   FORMAT ( 22H FAILED TO CONVERGE IN, I, 18H STEPS IN "REFRAC")
C
10    CONTINUE
C
20    CONTINUE
C
      IF ( ALT.LT.6000. ) THEN
C
      RRNG = RANGE/RO
      RR2 = RRNG*RRNG
C
      RALT = ALT/HS
      RA2 = RALT*RALT
      RA3 = RA2*RALT
C
      DELRNG = XNO*RANGE*( 1. + 67.*RR2 )*
     1          ( 1. - .49939*RALT + .17472*RA2 - .04344*RA3 )
      DELEL = XNO*RANGE*COS(EL/CRAD)/(2.*HS)*( 1. + 46.*RR2 )*
     1         ( 1.-.33324*RALT + .08558*RA2 - .01681*RA3 )
C
      ELSE
C
         IF ( ALT.LT.1.E+5 ) THEN
C
         HSTR = ( 1. - EXP( -ALT/HS ) )*HS
C
         ELSE
C
         HSTR = HS
C
         END IF
C
      XNO2 = XNO*XNO
      RALT = ALT/RO
      RA2 = RALT*RALT
      RHS = HSTR/RO
C
      A1 = .394 + 1.16E+5*XNO2
```

```
      A3 = .004 + 1.E+5*XNO2
      B1 = 59.9 + 1.14E+4*XNO - 1.9E+7*XNO2
      B2 = -3.379 - 1.E+3*XNO + 8.E+6*XNO2
      B3 = .007
      C1 = .7181 - 246.*XNO + 2.1E+4*XNO2
      C2 = 27.5 - 7.8E+3*XNO - 9.96E+7*XNO2
      C3 = -4.2 - 2.3E+3*XNO - 4.32E+7*XNO2
      C4 = 141.1 - 1.1E+4*XNO + 4.03E+8*XNO2
      C5 = -20.4 - 9.E+4*XNO - 3.1E+7*XNO2
C
      AA = ( A1 + A2*RALT )/( 1. + A3*RALT )
      BB = ( B1 + B2*RALT )/( 1. + B3*RALT )
      CC = ( C1 + C2*RALT + C3 * RA2 )/( 1. + C4*RALT + C5*RA2 )
C
      SEL = SIN( EL/CRAD )
      CEL = COS( EL/CRAD )
C
      ROOT = SQRT( SEL**2 + 2.*RHS )
C
      DELEL = XNO*CEL*( 1.+AA*EXP( -BB*SEL ) )/(( 1.-CC )*ROOT+C*SEL)
      DELEL = DELEL*( 1.- 2.*RHS/( ROOT + SEL ) )
      DELRNG = 2.*XNO*HSTR/( ROOT + SEL )
      DELRNG = DELRNG*( 1.-2.7E+7*(XNO**1.5)*RHS*(CEL**(1.4E+6*XNO)) )
C
      END IF
C
      DELRNG = DELRNG/.3048
      DELEL = DELEL*CRAD
C
      RETURN
      END
```

```
      SUBROUTINE COOR ( RXE, RYE, RZE, PHI, LAMDA, HTI )
      REAL LAT, LONG, LAMDA
C
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
C
      DATA EPS / 1.0E-5 /
C
      LAMDA = CRAD*ATAN2( RYE, RXE )
      CLAM = COS( LAMDA/CRAD )
      E = 1.0/FLAT
      OMES = ( 1.0 - E )**2
C
      DO 100 I = 1, 50
C
      PHII = PHI
C
      CPHI = COS( PHII/CRAD )
      SPHI = SIN( PHII/CRAD )
C
      ASQ = CPHI**2 + OMES*SPHI**2
      A = SQRT( ASQ )
      ASQSQ = ASQ*ASQ
C
      HTI = RXE/(CPHI*CLAM) - RE/A
C
      FPHI = RZE - SPHI*(RE*OMES/A + HTI)
      DFDP = -CPHI*((RE*OMES/A)*(1.0 + (E*(E-2.0)*SPHI**2)/ASQSQ)+HTI)
      DFDP=CPHI*((RE*OMES/A)*( E*(E-2.)*SPHI**2/ASQ -1. )-HTI)
      DPHI = (FPHI/DFDP)*CRAD
C
      PHI = PHII - DPHI
C
      IF ( ABS( DPHI ) - EPS ) 200, 200, 100
C
  100 CONTINUE
C
      WRITE(*, 901) I
901   FORMAT( 23H FAILED TO CONVERGE IN , I2, 16H STEPS IN "COOR")
      WRITE(*, 902) DPHI
902   FORMAT( E15.8 )
C
  200 CONTINUE
C
      RETURN
      END
      SUBROUTINE ECPOS ( LAT, LONG, HT, RXE, RYE, RZE )
      REAL LAT, LONG
C
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
C
      CLAT = COS( LAT/CRAD )
      SLAT = SIN( LAT/CRAD )
      CLONG = COS( LONG/CRAD )
      SLONG = SIN( LONG/CRAD )
C
      E = 1.0/FLAT
      OMES = ( 1.0 - E )**2
      ASQ = CLAT**2 + OMES*SLAT**2
      A = SQRT( ASQ )
C
      R = RE/A + HT
      S = RE*OMES/A + HT
C
      RXE = R*CLAT*CLONG
```

```
      RZE = S*SLAT
C
      RETURN
      END
      SUBROUTINE AGRAV ( R )
      DIMENSION R(3)
C
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / CONST / CRAD, AAGRAV, HRSEC, PERCNT, XMRAD
      COMMON / EARTH / GRAVI(3), AGMX(3,3)
C
      RMAGS = R(1)*R(1) + R(2)*R(2) + R(3)*R(3)
      RMAG = SQRT( RMAGS )
      RMAGQ = RMAGS*RMAG
C
      RRS = (RE/RMAG)**2
      RZS = (R(3)/RMAG)**2
      XMUX = XMU*(1.0+1.5*XJ2*RRS*(1.0-5.0*RZS))
      XMUY = XMUX
      XMUZ = XMUX + 3.0*XJ2*RRS*XMU
C
      GRAVI(1) = -XMUX*R(1)/RMAGQ
      GRAVI(2) = -XMUY*R(2)/RMAGQ
      GRAVI(3) = -XMUZ*R(3)/RMAGQ
C
      FAC1 = XMU*XJ2*RRS*( 2.*(1.-5.*RZS) - 1. )/RMAGQ
      FAC4 = XMU*XJ2*RRS*( 2.*(1.-5.*RZS) + 4. )/RMAGQ
C
      AGMX(1,1) = -XMUX/RMAGQ + 3.0*(R(1)/RMAG)**2*(XMUX/RMAGQ + FAC1)
      AGMX(2,1) = 3.0*(R(1)*R(2)/RMAGS)*(XMUX/RMAGQ + FAC1)
      AGMX(3,1) = 3.0*(R(1)*R(3)/RMAGS)*(XMUX/RMAGQ + FAC4)
C
      AGMX(1,2) = AGMX(2,1)
      AGMX(2,2) = -XMUY/RMAGQ + 3.0*(R(2)/RMAG)**2*(XMUY/RMAGQ + FAC1)
      AGMX(3,2) = 3.0*(R(2)*R(3)/RMAGS)*(XMUY/RMAGQ + FAC4)
C
      AGMX(1,3) = AGMX(3,1)
      AGMX(2,3) = AGMX(3,2)
      AGMX(3,3) = -AGMX(1,1) - AGMX(2,2)
C
      RETURN
      END
      SUBROUTINE CIEFMX ( TIME, CIEF )
C
      DIMENSION CIEF(3,3)
C
      COMMON / EDATA / RE, FLAT, OMEGAE, XMU, XJ2
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
C
      THTE = OMEGAE*CRAD*TIME
      STHT = SIN(THTE/CRAD)
      CTHT = COS(THTE/CRAD)
C
      CIEF(1,1) = CTHT
      CIEF(2,1) = -STHT
      CIEF(3,1) = 0.0
      CIEF(1,2) = STHT
      CIEF(2,2) = CTHT
      CIEF(3,2) = 0.0
      CIEF(1,3) = 0.0
      CIEF(2,3) = 0.0
      CIEF(3,3) = 1.0
C
      RETURN
      END
```

```
      SUBROUTINE AXCBIQ ( Q, V, P )
C
      DIMENSION Q(4), V(3), P(3,4)
C
      P(1,1) = Q(1)*V(1) - Q(4)*V(2) + Q(3)*V(3)
      P(2,1) = Q(4)*V(1) + Q(1)*V(2) - Q(2)*V(3)
      P(3,1) = -Q(3)*V(1) + Q(2)*V(2) + Q(1)*V(3)
C
      P(1,2) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
      P(2,2) = Q(3)*V(1) - Q(2)*V(2) - Q(1)*V(3)
      P(3,2) = Q(4)*V(1) + Q(1)*V(2) - Q(2)*V(3)
C
      P(1,3) = -Q(3)*V(1) + Q(2)*V(2) + Q(1)*V(3)
      P(2,3) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
      P(3,3) = -Q(1)*V(1) + Q(4)*V(2) - Q(3)*V(3)
C
      P(1,4) = -Q(4)*V(1) - Q(1)*V(2) + Q(2)*V(3)
      P(2,4) = Q(1)*V(1) - Q(4)*V(2) + Q(3)*V(3)
      P(3,4) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
C
      CALL SMLT( 2.0, P, P, 3, 4 )
C
      RETURN
      END
      SUBROUTINE AUXVAB
C
      DIMENSION VEC(3),VEC2(3),TMP(3,3)
C
      COMMON / VARIAB / VR(3), VW(3), VWGRAD(3)
      COMMON / TMAT / CBI(3,3), CIB(3,3), CLLB(3,3), CEFBRI(3,3)
      COMMON / PARTLA / PVMVB(3), PAVB(3), PBVB(3), PVMH, PAH, PBH
     1                , PVMVW(3), PAVW(3), PBVW(3)
C
      DATA CRAD / 57.295779 /
C
      VMAGS = VR(1)*VR(1) + VR(2)*VR(2) + VR(3)*VR(3)
      IF( VMAGS.LT.1.0 ) VMAGS = 1.0
      VMAG = SQRT(VMAGS)
C
      VMAGQ = VMAGS*VMAG
C
      DO 10 I = 1, 3
      PVMVB(I) = VR(I)/VMAG
10    CONTINUE
C
      VR13S = VMAGS - VR(2)*VR(2)
      IF( VR13S.LT.1.0 ) VR13S = 1.0
      VR13 = SQRT(VR13S)
C
      PAVB(1) = -VR(3)/VR13S
      PAVB(2) = 0.0
      PAVB(3) = VR(1)/VR13S
      CALL SMLT ( CRAD, PAVB, PAVB, 3, 1 )
C
      PBVB(1) = -VR(1)*VR(2)/(VMAGQ*VR13)
      PBVB(2) = VR13/VMAGQ
      PBVB(3) = -VR(2)*VR(3)/(VMAGQ*VR13)
      CALL SMLT ( CRAD, PBVB, PBVB, 3, 1 )
C
      CALL MULT ( CLLB, VWGRAD, VEC, 3, 3, 1 )
      CALL INNER ( PVMVB, VEC, PVMH, 3 )
      CALL INNER ( PAVB, VEC, PAH, 3 )
      CALL INNER ( PBVB, VEC, PBH, 3 )
C
      CALL TRANS ( CLLB, TMP, 3, 3 )
      CALL MULT ( TMP, VR, VEC, 3, 3, 1 )
```

```
C
      DO 20 I = 1, 3
      VEC2(I) = VR(1)*CLLB(3,I) - VR(3)*CLLB(1,I)
20    CONTINUE
C
      A = -1.0/VR13S
      CALL SMLT ( A, VEC2, PAVW, 3, 1 )
      CALL SMLT ( CRAD, PAVW, PAVW, 3, 1 )
C
      DO 30 I = 1, 3
      VEC2(I) = VMAG*CLLB(2,I) - VR(2)*PVMVW(I)/(2.*VMAG)
30    CONTINUE
C
      A = -1.0/(VMAG*VR13)
      CALL SMLT ( A, VEC2, PBVW, 3, 1 )
      CALL SMLT ( CRAD, PBVW, PBVW, 3, 1 )
C
100   CONTINUE
C
      RETURN
      END
      SUBROUTINE AXCIBQ( Q, V, P )
C
      DIMENSION Q(4), V(3), P(3,4)
C
      P(1,1) = Q(1)*V(1) + Q(4)*V(2) - Q(3)*V(3)
      P(2,1) = -Q(4)*V(1) + Q(1)*V(2) + Q(2)*V(3)
      P(3,1) = Q(3)*V(1) - Q(2)*V(2) + Q(1)*V(3)
C
      P(1,2) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
      P(2,2) = Q(3)*V(1) - Q(2)*V(2) + Q(1)*V(3)
      P(3,2) = Q(4)*V(1) - Q(1)*V(2) - Q(2)*V(3)
C
      P(1,3) = -Q(3)*V(1) + Q(2)*V(2) - Q(1)*V(3)
      P(2,3) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
      P(3,3) = Q(1)*V(1) + Q(4)*V(2) - Q(3)*V(3)
C
      P(1,4) = -Q(4)*V(1) + Q(1)*V(2) + Q(2)*V(3)
      P(2,4) = -Q(1)*V(1) - Q(4)*V(2) + Q(3)*V(3)
      P(3,4) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
C
      CALL SMLT( 2.0, P, P, 3, 4 )
C
      RETURN
      END
      SUBROUTINE QMTRX( OMEGA, QDMTRX )
C
      DIMENSION OMEGA(3), QDMTRX(4,4)
C
      QDMTRX(1,1) = 0.0
      QDMTRX(1,2) = -0.5*OMEGA(1)
      QDMTRX(1,3) = -0.5*OMEGA(2)
      QDMTRX(1,4) = -0.5*OMEGA(3)
      QDMTRX(2,1) = 0.5*OMEGA(1)
      QDMTRX(2,2) = 0.0
      QDMTRX(2,3) = 0.5*OMEGA(3)
      QDMTRX(2,4) = -0.5*OMEGA(2)
      QDMTRX(3,1) = 0.5*OMEGA(2)
      QDMTRX(3,2) = -0.5*OMEGA(3)
      QDMTRX(3,3) = 0.0
      QDMTRX(3,4) = 0.5*OMEGA(1)
      QDMTRX(4,1) = 0.5*OMEGA(3)
      QDMTRX(4,2) = 0.5*OMEGA(2)
      QDMTRX(4,3) = -0.5*OMEGA(1)
      QDMTRX(4,4) = 0.0
```

```
C
      RETURN
      END
      SUBROUTINE PQOMEG( Q, P )
C
      DIMENSION Q(4), P(4,3)
C
      P(1,1) = -0.5*Q(2)
      P(1,2) = -0.5*Q(3)
      P(1,3) = -0.5*Q(4)
      P(2,1) = 0.5*Q(1)
      P(2,2) = -0.5*Q(4)
      P(2,3) = 0.5*Q(3)
      P(3,1) = 0.5*Q(4)
      P(3,2) = 0.5*Q(1)
      P(3,3) = -0.5*Q(2)
      P(4,1) = -0.5*Q(3)
      P(4,2) = 0.5*Q(2)
      P(4,3) = 0.5*Q(1)
C
      RETURN
      END
      SUBROUTINE AXVABQ
C
      DIMENSION VEC1(4), VEC2(4), VEC3(4)
C
      COMMON / STATES / RI(3), VB(3), Q(4), OMEGA(3)
      COMMON / VARIAB / VR(3), VW(3), VWGRAD(3)
      COMMON / PARTLB / PVRQ(3,4), PVMQ(4), PAQ(4), PBQ(4)
C
      DATA CRAD / 57.295779 /
C
      VMS = VR(1)**2 + VR(2)**2 + VR(3)**2
      VM = SQRT(VMS)
      VR2VMS = VR(2)**2 + VMS
      VR1R3S = VR(1)**2 + VR(3)**2
      IF( VR1R3S.LT.1.0 ) VR1R3S = 1.0
      IF ( VR2VMS.LT.1.0 ) VR2VMS = 1.0
      IF ( VM.LT.1.0 ) VM = 1.0
C
      CALL AXCIBQ( Q, VW, PVRQ )
      CALL SMLT( -1.0, PVRQ, PVRQ, 3, 4 )
C
      DO 10 I = 1, 4
      VEC1(I) = PVRQ(1,I)
      VEC2(I) = PVRQ(2,I)
      VEC3(I) = PVRQ(3,I)
10    CONTINUE
C
      DO 20 I = 1, 4
      PVMQ(I) = VEC1(I)*VR(1) + VEC2(I)*VR(2) + VEC3(I)*VR(3)
      PVMQ(I) = 0.5*PVMQ(I)/VM
20    CONTINUE
C
      DO 30 I = 1, 4
      PAQ(I) = VEC3(I)*VR(1) - VEC1(I)*VR(3)
      PAQ(I) = CRAD*PAQ(I)/VR1R3S
30    CONTINUE
C
      DO 40 I = 1, 4
      PBQ(I) = VEC2(I)*VM - PVMQ(I)*VR(2)
      PBQ(I) = CRAD*PBQ(I)/VR2VMS
40    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE CBI( Q, CBI )
C
      DIMENSION Q(4), CBI(3,3)
C
      CBI(1,1) = Q(1)*Q(1) + Q(2)*Q(2) - Q(3)*Q(3) - Q(4)*Q(4)
      CBI(2,1) = 2.*( Q(2)*Q(3) + Q(1)*Q(4) )
      CBI(3,1) = 2.*( Q(2)*Q(4) - Q(1)*Q(3) )
C
      CBI(1,2) = 2.*( Q(2)*Q(3) - Q(1)*Q(4) )
      CBI(2,2) = Q(1)*Q(1) - Q(2)*Q(2) + Q(3)*Q(3) - Q(4)*Q(4)
      CBI(3,2) = 2.*( Q(1)*Q(2) + Q(3)*Q(4) )
C
      CBI(1,3) = 2.*( Q(2)*Q(4) + Q(1)*Q(3) )
      CBI(2,3) = 2.*( Q(3)*Q(4) - Q(1)*Q(2) )
      CBI(3,3) = Q(1)*Q(1) - Q(2)*Q(2) - Q(3)*Q(3) + Q(4)*Q(4)
C
      RETURN
      END
      SUBROUTINE CBIMX( THT, CBI )
C
      DIMENSION THT(3), CBI(3,3)
C
      DATA CRAD / 57.295779 /
C
      SPHI = SIN(THT(1)/CRAD)
      CPHI = COS(THT(1)/CRAD)
      STHT = SIN(THT(2)/CRAD)
      CTHT = COS(THT(2)/CRAD)
      SPSI = SIN(THT(3)/CRAD)
      CPSI = COS(THT(3)/CRAD)
C
      CBI(1,1) = CTHT*CPSI
      CBI(2,1) = CTHT*SPSI
      CBI(3,1) = -STHT
      CBI(1,2) = SPHI*STHT*CPSI - CPHI*SPSI
      CBI(2,2) = SPHI*STHT*SPSI + CPHI*CPSI
      CBI(3,2) = SPHI*CTHT
      CBI(1,3) = CPHI*STHT*CPSI + SPHI*SPSI
      CBI(2,3) = CPHI*STHT*SPSI - SPHI*CPSI
      CBI(3,3) = CPHI*CTHT
C
      RETURN
      END
      SUBROUTINE E2QUAT( THT, Q )
C
C   SUBROUTINE TO CONVERT EULER ANGLES INTO QUATERNIONS REF: PERRY
C
      DIMENSION THT(3), AM(3,3), V(3), Q(4)
C
      DATA CRAD / 57.295779 /
C
      PHI = THT(1)
      THE = THT(2)
      PSI = THT(3)
C
      CPHI = COS( PHI/CRAD )
      SPHI = SIN( PHI/CRAD )
      CTHT = COS( THE/CRAD )
      STHT = SIN( THE/CRAD )
      CPSI = COS( PSI/CRAD )
      SPSI = SIN( PSI/CRAD )
C
      AM(1,1) = CPSI*CTHT
      AM(1,2) = -SPSI*CPHI + CPSI*STHT*SPHI
      AM(1,3) = SPSI*SPHI + CPSI*STHT*CPHI
C
```

```
      AM(2,1) = SPSI*CTHT
      AM(2,2) = CPSI*CPHI + SPSI*STHT*SPHI
      AM(2,3) = -CPSI*SPHI + SPSI*STHT*CPHI
C
      AM(3,1) = -STHT
      AM(3,2) = CTHT*SPHI
      AM(3,3) = CTHT*CPHI
C
      IF( (ABS(THE-90.).LT.1.).AND.(ABS(PHI-PSI).LT.1.) ) THEN
C
C     LOOP TO CONVERSION ( SEE PERRY )
C
      I = 0
C
      T = AM(1,1) + AM(2,2) + AM(3,3)
      S = T
C
      DO 10 II = 1, 3
      IF ( AM(II,II).GT.S ) THEN
      S = AM(II,II)
      I = II
      ELSE
      I = 0
      END IF
   10 CONTINUE
C
      T = SQRT( 1.0 + 2.0*S - T )
C
      K = 2
      J = 3
      N = 1
C
   20 CONTINUE
C
      IF ( ( I.EQ.0 ).OR.( N.EQ.I ) ) THEN
      S = (AM(J,K) - AM(K,J))/T
      V(N) = S
      ELSE
      V(J+K-I) = (AM(J,K) + AM(K,J))/T
      END IF
C
      IF ( N - 2 ) 30, 40, 50
   30 CONTINUE
      K = 3
      J = 1
      N = 2
      GO TO 20
   40 CONTINUE
      K = 1
      J = 2
      N = 3
      GO TO 20
   50 CONTINUE
C
      IF ( I.EQ.0 ) THEN
      S = T
      ELSE
      V(I) = T
      END IF
C
      IF ( S.GE.0.0 ) T = 0.5
      IF ( S.LT.0.0 ) T = -0.5
C
      S = T*S
      DO 60 II = 1, 3
      V(II) = T*V(II)
```

```
6     CONTINUE
      Q(1) = S
      Q(2) = V(1)
      Q(3) = V(2)
      Q(4) = V(3)
C
      ELSE
C
      Q(1) = SQRT( (AM(1,1)+AM(2,2)+AM(3,3)+1.0)/4.0 )
      Q(4) = (AM(2,1)-AM(1,2))/(4.*Q(1))
      Q(3) = (AM(2,3)+AM(3,2))/(4.*Q(4))
      Q(2) = (AM(1,2)+AM(2,1))/(4.*Q(3))
C
      END IF
C
      RETURN
      END
      SUBROUTINE QUAT2E( Q, THT )
C
      DIMENSION Q(4), THT(3)
C
      DATA CRAD / 57.295779 /
C
      TPSI = 2.*(Q(2)*Q(3)+Q(1)*Q(4))
      DPSI = Q(1)**2+Q(2)**2-Q(3)**2-Q(4)**2
      STHT = -2.*(Q(2)*Q(4)-Q(3)*Q(1))
      TPHI = 2.*(Q(3)*Q(4)+Q(1)*Q(2))
      DPHI = Q(1)**2+Q(4)**2-Q(2)**2-Q(3)**2
C
      IF ( DPHI.NE.0.0 ) THEN
      THT(1) = CRAD*ATAN2( TPHI, DPHI )
      END IF
      IF ( ABS(STHT).LE.1.0 ) THEN
      THT(2) = CRAD*ASIN( STHT )
      END IF
      IF ( DPSI.NE.0.0 ) THEN
      THT(3) = CRAD*ATAN2( TPSI, DPSI )
      END IF
C
      RETURN
      END
      SUBROUTINE AUXRAT( D, O, A )
C
      DIMENSION D(3), O(3), A(3,3)
C
      A(1,1) = O(2)*D(2) + O(3)*D(3)
      A(2,1) = O(2)*D(1) - 2.0*O(1)*D(2)
      A(3,1) = O(3)*D(1) - 2.0*O(1)*D(3)
C
      A(1,2) = O(1)*D(2) - 2.0*O(2)*D(1)
      A(2,2) = O(1)*D(1) + O(3)*D(3)
      A(3,2) = O(3)*D(2) - 2.0*O(2)*D(3)
C
      A(1,3) = O(1)*D(3) - 2.0*O(3)*D(1)
      A(2,3) = O(2)*D(3) - 2.0*O(3)*D(2)
      A(3,3) = O(1)*D(1) + O(2)*D(2)
C
      RETURN
      END
      SUBROUTINE TMATY( YAW, TYAW )
C
      DIMENSION TYAW(3,3)
C
      DATA CRAD / 57.295779 /
C
      SY = SIN( YAW/CRAD )
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CY = COS( YAW/CRAD )
C
      TYAW(1,1) = CY
      TYAW(1,2) = -SY
      TYAW(1,3) = 0.0
      TYAW(2,1) = SY
      TYAW(2,2) = CY
      TYAW(2,3) = 0.0
      TYAW(3,1) = 0.0
      TYAW(3,2) = 0.0
      TYAW(3,3) = 1.0
C
      RETURN
      END
      SUBROUTINE TMATP( PITCH, TPITCH )
C
      DIMENSION TPITCH(3,3)
C
      DATA CRAD / 57.295779 /
C
      SP = SIN( PITCH/CRAD )
      CP = COS( PITCH/CRAD )
C
      TPITCH(1,1) = CP
      TPITCH(1,2) = 0.0
      TPITCH(1,3) = SP
      TPITCH(2,1) = 0.0
      TPITCH(2,2) = 1.0
      TPITCH(2,3) = 0.0
      TPITCH(3,1) = -SP
      TPITCH(3,2) = 0.0
      TPITCH(3,3) = CP
C
      RETURN
      END
```

```
      SUBROUTINE ADD ( A, B, C, NRA, NCA )
      DIMENSION A(NRA,NCA), B(NRA,NCA), C(NRA,NCA)
C
      DO 10 J = 1, NCA
      DO 10 I = 1, NRA
      C(I,J) = A(I,J) + B(I,J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE SUBT ( A, B, C, NRA, NCA )
      DIMENSION A(NRA,NCA), B(NRA,NCA), C(NRA,NCA)
C
      DO 10 J = 1, NCA
      DO 10 I = 1, NRA
      C(I,J) = A(I,J) - B(I,J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE MULT ( A, B, C, NRA, NCA, NCB )
      DIMENSION A(NRA,NCA), B(NCA, NCB), C(NRA,NCB)
C
      DO 20 J = 1, NCB
            DO 20 I = 1, NRA
                  K = 0
                  TEMP = 0.0
            DO 10 L = 1, NCA
            K = K + 1
            TEMP = TEMP + A(I,K)*B(K,J)
   10       CONTINUE
      C(I,J) = TEMP
   20 CONTINUE
C
      RETURN
      END
      SUBROUTINE TRANS ( A, B, NRA, NCA )
      DIMENSION A(NRA,NCA), B(NCA,NRA)
C
      DO 10 I = 1, NRA
      DO 10 J = 1, NCA
      B(J,I) = A(I,J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE SMLT ( C, A, B, NRA, NCA )
      DIMENSION A(NRA,NCA), B(NRA,NCA)
C
      DO 10 I = 1, NRA
      DO 10 J = 1, NCA
      B(I,J) = C*A(I,J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE SWITCH ( A, B, NRA, NCA )
      DIMENSION A(NRA,NCA), B(NRA,NCA)
C
      DO 10 I = 1, NRA
      DO 10 J = 1, NCA
      B(I,J) = A(I,J)
   10 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SKEW ( A, B )
      DIMENSION A(3), B(3,3)
C
      B(1,1) = 0.0
      B(1,2) = -A(3)
      B(1,3) = A(2)
      B(2,1) = -B(1,2)
      B(2,2) = 0.0
      B(2,3) = -A(1)
      B(3,1) = -B(1,3)
      B(3,2) = -B(2,3)
      B(3,3) = 0.0
C
      RETURN
      END
      SUBROUTINE IMBED ( SUB, M, N, BIG, IM, IN, IROW, ICOL )
      DIMENSION SUB(1), BIG(1)
C
      IF ( (M+IROW-1 .LE. IM) .AND. (N+ICOL-1 .LE. IN) ) GOTO 5
            WRITE(*,6000) IM, IN, M, N, IROW, ICOL
            STOP
C
    5 IB = ( ICOL - 1 )*IM + ( IROW - 1 )
      IS = 0
      IOFFST = IM - M
C
      DO 20 I = 1, N
      DO 10 J = 1, M
      IB = IB + 1
      IS = IS + 1
      BIG(IB) = SUB(IS)
   10 CONTINUE
      IB = IB + IOFFST
   20 CONTINUE
C
      RETURN
C
 6000 FORMAT ( 5X, 6I10 )
C
      END
      SUBROUTINE INV2X2 ( A, DET, AI )
      DIMENSION A(2,2), AI(2,2)
C
      DET = A(1,1)*A(2,2) - A(1,2)*A(2,1)
C
      AI(1,1) = A(2,2)/DET
      AI(1,2) = -A(1,2)/DET
      AI(2,1) = -A(2,1)/DET
      AI(2,2) = A(1,1)/DET
C
      RETURN
      END
      SUBROUTINE ZEROM( A, NROW, NCOL )
C
      DIMENSION A(NROW,NCOL)
C
      DO 10 I = 1, NROW
      DO 10 J = 1, NCOL
C
      A(I,J) = 0.0
C
10    CONTINUE
C
      RETURN
      END
      [illegible]BRO[illegible]E O[illegible] ( [illegible] N, [illegible] )
```

```
      DIMENSION TAU(N), TMP(N,N)
C
      DO 20 I = 1, N
      DO 10 J = 1, N
      TMP(I,J) = 0.0
10    CONTINUE
      TMP(I,I) = -1./TAU(I)
20    CONTINUE
C
      RETURN
      END
      SUBROUTINE DIAG ( V, N, A )
C
      DIMENSION V(N), A(N,N)
C
      DO 20 I = 1, N
      DO 10 J = 1, N
C
      A(I,J) = 0.0
C
10    CONTINUE
C
      A(I,I) = V(I)
C
20    CONTINUE
C
      RETURN
      END
      SUBROUTINE SYMTRK ( A, NS, AS )
      DIMENSION A(NS,NS), AS(NS,NS)
C
      DO 10 I = 1, NS
      DO 10 J = I, NS
      AS(I,J) = 0.5*(A(I,J) + A(J,I))
      AS(J,I) = AS(I,J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE INNER ( A, B, C, NELEM )
      DIMENSION A(NELEM), B(NELEM)
C
      TEMP = 0.0
C
      DO 10 I = 1, NELEM
      TEMP = TEMP + A(I)*B(I)
   10 CONTINUE
C
      C = TEMP
C
      RETURN
      END
      SUBROUTINE INTRP1 ( VM, TM, TC, NM, C, SLOPE )
      DIMENSION TM(1), TC(1)
C
      DO 10 I = 1, NM
      IF ( VM - TM(I) ) 20, 20, 10
   10 CONTINUE
      K = NM
   20 CONTINUE
      K = I - 1
      IF ( K.EQ.0 ) K = 1
      SLOPE = ( TC(K+1) - TC(K) )/(TM(K+1) - TM(K))
      C = TC(K) + SLOPE*(VM - TM(K))
      RETURN
```

END

```
C     PROGRAM SSM
C
      DIMENSION TME(530)
C
      COMMON / TDATA / TIME, TNEXT, TMAX, TSAMP, TSTOP
      COMMON / FILDAT / XKM(30),PKM(30,30),F(30,30),Q(30,30),N,NMEAS
      COMMON/SMTHER/PHI(30,30),PHIT(30,30),PKP1M(30,30),PKPI(30,30)
     1       ,AK(30,30),XKGN(30),PKGN(30,30),XKP1GN(30),PKP1GN(30,30)
      COMMON / EARTH / GRAV(3), AGMX(3,3)
      COMMON / RSTATE / RBRI(3), VBRI(3), ABRI(3), THTI(3)
     1                 ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                 ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
      COMMON / METEOR / TALT(400),TSRHO(400),TSP(400),TSSUND(400)
     1                           ,TVWX(400),TVWY(400),NALT
     2                           ,USRHO(400),USP(400),USSUND(400)
     3                           ,UVWX(400),UVWY(400)
C
C     BLOCK DATA
C
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
      COMMON / LAUCOR / OLATD, OLONG, OHT
C
      OPEN ( UNIT=2, FILE='SMOIPT.DAT', STATUS='OLD',
     1 ACCESS='SEQUENTIAL', FORM='UNFORMATTED' )
      OPEN ( UNIT=3, FILE='SREFIPT.DAT', STATUS='OLD' )
C     1 ,ACCESS='SEQUENTIAL', FORM='UNFORMATTED' )
      OPEN ( UNIT=4, FILE='METDAT.DAT', STATUS='OLD' )
      OPEN ( UNIT=6, FILE='SMOOUT.DAT', STATUS='NEW' )
      OPEN ( UNIT=7, FILE='SASSOUT.DAT', STATUS='NEW'
     1       , ACCESS='DIRECT', RECL=90, IOSTAT=IOS )
      OPEN ( UNIT=8, FILE='SCRATCH.DAT', STATUS='UNKNOWN',
     1 ACCESS='SEQUENTIAL', FORM='UNFORMATTED' )
      OPEN ( UNIT=9, FILE='NREFIPT.DAT', STATUS='NEW' )
C
      DO 20 I = 1, NALT
      READ(4,998) TALT(I),TSRHO(I),TSP(I),TSSUND(I),TVWX(I),TVWY(I)
      READ(4,999) USRHO(I),USP(I),USSUND(I),UVWX(I),UVWY(I)
C     WRITE(*,997) TALT(I),TSRHO(I),TSP(I),TSSUND(I),TVWX(I),TVWY(I)
20    CONTINUE
C
      CLOSE( UNIT=4 )
C
      NTMAX = IFIX ( TMAX/TSAMP )
C
      DO 50 I = 1, NTMAX
      IT = NTMAX - I + 1
      TME(I) = IT*TSAMP
50    CONTINUE
C
      TSRCH = TME(1)
      CALL RTSIPT ( TSRCH )
C
      CALL RTSINT
C
      DO 1000 K = 2, NTMAX
C
      TSRCH = TME(K)
      CALL RTSIPT ( TSRCH )
C
      IF ( K.EQ.NTMAX ) THEN
      ELSE
      DT = TIME - TNEXT
      END IF
      CALL RTSPRP ( DT )
C
```

```
      CALL RTSUPD
C
      WRITE(6,901) TSRCH
      WRITE(6,901) (XKM(I),PKM(I,I),XKGN(I),PKGN(I,I), I = 1, N)
C
      CALL ASSESS
C
 1000 CONTINUE
C
      CLOSE ( UNIT=2 )
      CLOSE ( UNIT=3 )
      CLOSE ( UNIT=6 )
      REWIND 8
C
      DO 2000 J = 1, (NTMAX-1)
C
      IT = (NTMAX-1) - J + 2
      TSRCH = TME(IT)
C
      CALL REVERSE ( TSRCH )
C
2000  CONTINUE
C
      CLOSE ( UNIT=7 )
      CLOSE ( UNIT=8 )
      CLOSE ( UNIT=9 )
C
  901 FORMAT( 5X, 4E15.7 )
  905 FORMAT( 5X, 2F10.2, I5 )
  997 FORMAT( 3X, 5E15.8 )
  998 FORMAT( 5X, 6E15.8 )
  999 FORMAT( 20X, 5E15.8 )
C
      STOP
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      BLOCK DATA
C
      COMMON / TDATA / TIME, TNEXT, TMAX, TSAMP, TSTOP
      COMMON / FILDAT / XKM(30),PKM(30,30),F(30,30),Q(30,30),N,NMEAS
      COMMON/SMTHER/PHI(30,30),PHIT(30,30),PKP1M(30,30),PKPI(30,30)
     1       ,AK(30,30),XKGN(30),PKGN(30,30),XKP1GN(30),PKP1GN(30,30)
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
      COMMON / LAUCOR / OLATD, OLONG, OHT
      COMMON / METEOR / TALT(400),TSRHO(400),TSP(400),TSSUND(400)
     1                           ,TVWX(400),TVWY(400),NALT
     2                           ,USRHO(400),USP(400),USSUND(400)
     3                           ,UVWX(400),UVWY(400)
C
      DATA TIME, TMAX, TSAMP / 0.0, 520., 1.0 /
C
      DATA NALT / 400 /
C
      DATA N / 30 /
C
      DATA PHI / 900*0.0 /
      DATA PHIT / 900*0.0 /
      DATA PKP1M / 900*0.0 /
      DATA PKPI / 900*0.0 /
C
      DATA AK / 900*0.0 /
      DATA XKGN / 30*0.0 /
      DATA PKGN / 900*0.0 /
      DATA XKP1GN / 30*0.0 /
      DATA PKP1GN / 900*0.0 /
C
      DATA CRAD / 57.295779 /
      DATA AGRAV, HRSEC, PERCNT, XMRAD / 32.174, 3600., 100., 1000. /
      DATA RE, FLAT, OMEGE, XMU, XJ2 / 20925604.,298.257224,.7292115E-4
CFISCHDATA RE, FLAT, OMEGE, XMU, XJ2 / 20925741.,298.3,0.7292115E-4
     1, 1.4076468E+16, .10827E-2 /
C
      DATA OLATD, Olong, OHT / 28.62721, -80.62079, 0.0 /
C39AFEDATA OLATD, OLONG, OHT / 28.608420, -80.604089, 60.0 /
C
      END
```

```
      SUBROUTINE RTSIPT ( TSRCH )
C
      DIMENSION RHO(30,30)
C
      COMMON / TDATA / TIME, TNEXT, TMAX, TSAMP, TSTOP
      COMMON / FILDAT / XKM(30),PKM(30,30),F(30,30),Q(30,30),N,NMEAS
      COMMON / RSTATE / RBRI(3), VBRI(3), ABRI(3), THTI(3)
     1                 ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                 ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
C
      N = 30
C
10    CONTINUE
C
      READ(2) TIME, TNEXT, KO, NMEAS
C     WRITE(*, 901) TIME, TNEXT, KO, NMEAS
      READ(2) (XKM(I), I = 1, N)
      READ(2) ((PKM(I,J), J = 1, N), I = 1, N)
      READ(2) ((F(I,J), J = 1, N), I = 1, N)
      READ(2) ((Q(I,J), J = 1, N), I = 1, N)
C
      READ(3, 904) XMTIME
      READ(3, 904) (ABRI(I), I = 1, 3)
      READ(3, 904) (THTI(I), I = 1, 3)
      READ(3, 904) (OMEG(I), I = 1, 3)
      READ(3, 904) (RBRI(I), I = 1, 3)
      READ(3, 904) (VBRI(I), I = 1, 3)
      READ(3, 904) ((CIBRI(I,J), J = 1, 3), I = 1, 3)
      READ(3, 904) ((CBRIEF(I,J), J = 1, 3), I = 1, 3)
C
      IF ( ABS(TIME-TSRCH).LT.1.0E-4 ) THEN
C
      WRITE(*, 902) TIME, TNEXT, KO, NMEAS
C
      DO 50 I = 1, N
      DO 50 J = 1, N
C
      RHO(I,J) = PKM(I,J)/(SQRT(PKM(I,I))*SQRT(PKM(J,J)))
C
50    CONTINUE
C
      REWIND 2
      REWIND 3
      RETURN
C
      ELSE
C
      GO TO 10
C
      END IF
C
  901 FORMAT( 2E15.8, 2I5 )
  902 FORMAT( 5X, 2E15.8, 2I5 )
  903 FORMAT( 5E15.8 )
  904 FORMAT( 4X, 3E15.8 )
C
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE RTSINT
C
      COMMON / FILDAT / XKM(30),PKM(30,30),F(30,30),Q(30,30),N,NMEAS
      COMMON/SMTHER/PHI(30,30),PHIT(30,30),PKP1M(30,30),PKPI(30,30)
     1       ,AK(30,30),XKGN(30),PKGN(30,30),XKP1GN(30),PKP1GN(30,30)
C
      CALL SWITCH ( XKM, XKGN, N, 1 )
      CALL SWITCH ( XKM, XKP1GN, N, 1 )
C
      CALL SWITCH ( PKM, PKGN, N, N )
      CALL SWITCH ( PKM, PKP1GN, N, N )
C
      WRITE(*, 901) (XKGN(I), I = 1, N)
      WRITE(*, 901) (PKGN(I,I), I = 1, N)
C
      RETURN
C
901   FORMAT( 5X, 4E15.8 )
C
      END
```

```
      SUBROUTINE RTSUPD
C
      DIMENSION VEC1(30), VEC2(30), DUMB(30,30)
C
      COMMON / FILDAT / XKM(30),PKM(30,30),F(30,30),Q(30,30),N,NMEAS
      COMMON/SMTHER/PHI(30,30),PHIT(30,30),PKP1M(30,30),PKPI(30,30)
     1       ,AK(30,30),XKGN(30),PKGN(30,30),XKP1GN(30),PKP1GN(30,30)
C
C     CALL MULT ( PHI, XKM, VEC1, N, N, 1 )
      DO 20 I = 1, N
      TEMP = 0.0
      DO 10 J = 1, N
      TEMP = TEMP + PHI(I,J)*XKM(J)
10    CONTINUE
      VEC1(I) = TEMP
20    CONTINUE
C     CALL SUBT ( XKP1GN, VEC1, VEC2, N, 1 )
      DO 30 I = 1, N
      VEC2(I) = XKP1GN(I) - VEC1(I)
30    CONTINUE
C     CALL MULT ( AK, VEC2, VEC1, N, N, 1 )
      DO 50 I = 1, N
      TEMP = 0.0
      DO 40 J = 1, N
      TEMP = TEMP + AK(I,J)*VEC2(J)
40    CONTINUE
      VEC1(I) = TEMP
50    CONTINUE
C     CALL ADD ( XKM, VEC1, XKGN, N, 1 )
      DO 60 I = 1, N
      XKGN(I) = XKM(I) + VEC1(I)
60    CONTINUE
C
      CALL SWITCH ( XKGN, XKP1GN, N, 1 )
C
      CALL SUBT ( PKP1GN, PKP1M, DUMB, N, N )
      CALL MULT ( AK, DUMB, PKGN, N, N, N )
      CALL TRANS ( AK, DUMB, N, N )
      CALL MULT ( PKGN, DUMB, AK, N, N, N )
      CALL ADD ( PKM, AK, PKGN, N, N )
C
      CALL SWITCH ( PKGN, PKP1GN, N, N )
C
      RETURN
  901 FORMAT( 5X, 4E15.8 )
  909 FORMAT( 5X, I5 )
      END
```

```
      SUBROUTINE RTS ( D
C
      DIMENSION DUMB(30,30), L(30), M(30)
C
      COMMON / FILDAT / XKM(30),PKM(30,30),F(30,30),Q(30,30),N,NMEAS
      COMMON/SMTHER/PHI(30,30),PHIT(30,30),PKP1M(30,30),PKPI(30,30)
     1       ,AK(30,30),XKGN(30),PKGN(30,30),XKP1GN(30),PKP1GN(30,30)
C
      N = 30
C
      DO 20 I = 1, N
      DO 10 J = 1, N
      PHI(I,J) = F(I,J)*DT
10    CONTINUE
      PHI(I,I) = 1.0 + PHI(I,I)
20    CONTINUE
C
      CALL TRANS ( PHI, PHIT, N, N )
      CALL MULT ( PHI, PKM, DUMB, N, N, N )
      CALL MULT ( DUMB, PHIT, PKP1M, N, N, N )
      CALL ADD ( PKP1M, Q, PKP1M, N, N )
C
      CALL SWITCH ( PKP1M, PKPI, N, N )
      CALL INVNXN ( PKPI, N, D, L, M )
      CALL MULT ( PKP1M, PKPI, DUMB, N, N, N )
      CALL MULT ( PHIT, PKPI, DUMB, N, N, N )
      CALL MULT ( PKM, DUMB, AK, N, N, N )
C
      RETURN
901   FORMAT( 3X, 4E15.8 )
      END
```

```
      SUBROUTINE ASSESS
C
      DIMENSION UNIT(3,3),CBI(3,3),CIB(3,3),VB(3),AB(3)
     1          ,VEC1(3),VEC2(3),VEC3(3),VEC4(3),VEC5(3)
     2          ,TMP1(3,3),TMP2(3,3),TMP3(3,3),TMP4(3,3),TMP5(3,3)
     3          ,VECX(3),VECY(3),VECZ(3),TMPX(3,3),TMPY(3,3),TMPZ(3,3)
     4          ,TMPA(3,3),CCIB(3,3),VBR(3),RBR(3),ACC(3)
     5          ,VW(3),VR(3),ARRAY(90),BRRAY(90)
     6          ,PVW(3,3),C1(3,3)
     7          ,VEC6(6),VEC7(6)
C
      COMMON / TDATA / TIME, TNEXT, TMAX, TSAMP, TSTOP
      COMMON / FILDAT / XKM(30),PKM(30,30),F(30,30),Q(30,30),N,NMEAS
      COMMON/SMTHER/PHI(30,30),PHIT(30,30),PKP1M(30,30),PKPI(30,30)
     1       ,AK(30,30),XKGN(30),PKGN(30,30),XKP1GN(30),PKP1GN(30,30)
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2
      COMMON / CONST / CRAD, AGRAV, HRSEC, PERCNT, XMRAD
      COMMON / LAUCOR / OLATD, OLONG, OHT
      COMMON / EARTH / GRAV(3), AGMX(3,3)
      COMMON / RSTATE / RBRI(3),VBRI(3),ABRI(3),THTI(3)
     1                  ,GRAVI(3),CIBRI(3,3),CBRIEF(3,3),CEFBRI(3,3)
     2                  ,RIT0(3),VIT0(3),REF(3),ANET(3),OMEG(3)
      COMMON / METEOR / TALT(400),TSRHO(400),TSP(400),TSSUND(400)
     1                           ,TVWX(400),TVWY(400),NALT
     2                           ,USRHO(400),USP(400),USSUND(400)
     3                           ,UVWX(400),UVWY(400)
C
      DATA UNIT / 1.,3*0.,1.,3*0.,1. /
      DATA NR / 16 /
      DATA RHO, PS, VSOUND / .002378, 2116., 1117. /
      DATA VWX, VWY / 2*0.0 /
      DATA PRHOPH, PSPH, PVSPH / 3*0.0 /
      DATA PVWXPH, PVWYPH / 2*0.0 /
      DATA PVW / 9*0.0 /
      DATA C1 / 2*0.0,1.0,1.0,3*0.0,1.0,0.0 /
C
      CALL ZEROM ( ARRAY, 90, 1 )
      CALL ZEROM ( BRRAY, 90, 1 )
C
      CALL CBIMX ( THTI, CBI )
      CALL TRANS ( CBI, CIB, 3, 3 )
C
      DO 20 I = 1, 3
      DO 10 J = 1, 3
      TMP1(I,J) = PKGN(I,J)
      TMP2(I,J) = PKGN(I+3,J+3)
      TMP3(I,J) = PKGN(I+6,J+6)
      TMP4(I,J) = PKGN(I+9,J+9)
      TMP5(I,J) = PKGN(I+12,J+12)
10    CONTINUE
      RBR(I) = RBRI(I) + XKGN(I)
      VBR(I) = VBRI(I) + XKGN(I+3)
      VEC2(I) = VBR(I)
      VEC3(I) = XKGN(I+6)
      VEC4(I) = XKGN(I+9)
      VEC5(I) = XKGN(I+12)
20    CONTINUE
      CALL MULT ( CBRIEF, RBR, REF, 3, 3, 1 )
C
      CALL SKEW ( VEC3, TMPX )
      CALL ADD ( UNIT, TMPX, TMPX, 3, 3 )
      CALL MULT ( CIB, TMPX, CCIB, 3, 3, 3 )
      ARRAY(1) = TIME
      BRRAY(1) = 0.0
C
C**   VEHICLE POSITION
```

```
      ARRAY(2) = RBR(1)
      ARRAY(3) = RBR(2)
      ARRAY(4) = RBR(3)
      BRRAY(2) = SQRT( PKGN(1,1) )
      BRRAY(3) = SQRT( PKGN(2,2) )
      BRRAY(4) = SQRT( PKGN(3,3) )
C
      CALL COOR ( REF(1), REF(2), REF(3), XLAT, XLONG, ALT )
      RM = SQRT ( REF(1)**2 + REF(2)**2 + REF(3)**2 )
      ARRAY(49) = XLONG
      ARRAY(50) = XLAT
      ARRAY(51) = ALT
      ARRAY(54) = RM
      BRRAY(49) = CRAD*SQRT( PKGN(2,2) )/(RM*COS( XLAT/CRAD ))
      BRRAY(50) = CRAD*SQRT( PKGN(1,1) )/RM
      RMS = RM*RM
      CALL MULT ( TMP1, RBR, VECX, 3, 3, 1 )
      CALL INNER ( RBR, VECX, SIGRMS, 3 )
      SIGRM = SQRT ( SIGRMS/RMS )
      BRRAY(51) = SIGRM
      BRRAY(54) = SIGRM
C
      CALL ECPOS ( OLATD, OLONG, OHT, VECZ(1), VECZ(2), VECZ(3) )
      DO 30 I = 1, 3
      VECX(I) = REF(I) - VECZ(I)
30    CONTINUE
      RS = SQRT( VECX(1)**2 + VECX(2)**2 + VECX(3)**2 )
      ARRAY(58) = RS
      CALL MULT ( TMP1, VECX, VECY, 3, 3, 1 )
      CALL INNER ( VECX, VECY, SIGRSS, 3 )
      IF ( RS.NE.0.0 ) THEN
      SIGRSS = SIGRSS/(RS*RS)
      ELSE
      SIGRSS = 0.0
      END IF
      BRRAY(58) = SQRT ( SIGRSS )
C
C**   BOOST REFERENCE VELOCITY AND UNCERTAINTY BOUNDS
C
      VMI = SQRT ( VEC2(1)**2 + VEC2(2)**2 + VEC2(3)**2 )
      ARRAY(5) = VBR(1)
      ARRAY(6) = VBR(2)
      ARRAY(7) = VBR(3)
      ARRAY(55) = VMI
      BRRAY(5) = SQRT( PKGN(4,4) )
      BRRAY(6) = SQRT( PKGN(5,5) )
      BRRAY(7) = SQRT( PKGN(6,6) )
      VMIS = VMI*VMI
      CALL MULT ( TMP2, VBR, VECX, 3, 3, 1 )
      CALL INNER ( VBR, VECX, SIGVMS, 3 )
      IF ( VMI.NE.0.0 ) THEN
      SIGVM = SQRT ( SIGVMS/VMIS )
      ELSE
      SIGVM = 0.0
      END IF
      BRRAY(55) = SIGVM
C
C**   BODY REFERENCED VELOCITY AND UNCERTAINTY BOUNDS
C
      CALL ZEROM ( VEC1, 3, 1 )
      VEC1(3) = OMEGE
      CALL SKEW ( VEC1, TMPX )
      CALL MULT ( TMPX, REF, VEC1, 3, 3, 1 )
      CALL MULT ( CIBRI, VEC1, VECX, 3, 3, 1 )
      CALL SUBT ( VEC2, VECX, VECY, 3, 1 )
```

```
      CALL MULT ( CCIB, VECY, VB, 3, 3, 1 )
C
      VT = SQRT ( VB(1)**2 + VB(2)**2 + VB(3)**2 )
      ARRAY(20) = VB(1)
      ARRAY(21) = VB(2)
      ARRAY(22) = VB(3)
C
C*    INERTIAL VELOCITY
C
      CALL MULT ( TMP2, CBI, TMPY, 3, 3, 3 )
      CALL MULT ( CIB, TMPY, TMPX, 3, 3, 3 )
C
C*    PLATFORM TILT
C
      CALL SKEW ( VEC2, TMPY )
      CALL MULT ( CIB, TMPY, TMPZ, 3, 3, 3 )
      CALL TRANS ( TMPZ, TMPA, 3, 3 )
      CALL MULT ( TMPZ, TMP3, TMPY, 3, 3, 3 )
      CALL MULT ( TMPY, TMPA, TMPZ, 3, 3, 3 )
C
      CALL ADD ( TMPX, TMPZ, TMPX, 3, 3 )
      BRRAY(20) = SQRT( TMPX(1,1) )
      BRRAY(21) = SQRT( TMPX(2,2) )
      BRRAY(22) = SQRT( TMPX(3,3) )
C
C** ANGLE OF ATTACK, SIDESLIP AND VELOCITY WRT AIR MASS W/UNCERTAIN
C
      IF ( ALT.LT.400000. ) THEN
C
      CALL INTRP1 ( ALT, TALT, TSRHO, NALT, RHO, PRHOPH )
      CALL INTRP1 ( ALT, TALT, TSP, NALT, PS, PPSPH )
      CALL INTRP1 ( ALT, TALT, TSSUND, NALT, VSOUND, PVSPH )
      CALL INTRP1 ( ALT, TALT, TVWX, NALT, VWX, PVWXPH )
      CALL INTRP1 ( ALT, TALT, TVWY, NALT, VWY, PVWYPH )
      CALL INTRP1 ( ALT, TALT, USRHO, NALT, UNCRHO, SLOPE )
      CALL INTRP1 ( ALT, TALT, USP, NALT, UNCSP, SLOPE )
      CALL INTRP1 ( ALT, TALT, USSUND, NALT, UNCVS, SLOPE )
      CALL INTRP1 ( ALT, TALT, UVWX, NALT, UNCVWX, SLOPE )
      CALL INTRP1 ( ALT, TALT, UVWY, NALT, UNCVWY, SLOPE )
C
      ELSE
      END IF
C
      VW(1) = VWX
      VW(2) = VWY
      VW(3) = 0.0
      PVW(1,1) = UNCVWX
      PVW(2,2) = UNCVWY
C
      ATEMP = (VSOUND/49.02)**2
      ARRAY(79) = ATEMP
      ARRAY(80) = PS
      ARRAY(81) = RHO
      BRRAY(79) = UNCVW*SQRT( ATEMP )*(2./49.02)
      BRRAY(80) = UNCPS
      BRRAY(81) = UNCRHO
C
      CALL TMATY ( -XLONG, TMPX )
      CALL TMATP ( XLAT, TMPY )
      CALL MULT ( TMPY, TMPX, TMPZ, 3, 3, 3 )
      CALL TRANS ( TMPZ, TMPX, 3, 3 )
      CALL MULT ( CEFBRI, TMPX, TMPY, 3, 3, 3 )
      CALL MULT ( TMPY, VW, VECX, 3, 3, 1 )
      CALL SUBT ( VB, VECX, VR, 3, 1 )
C
      VMA = SQRT ( VR(1)**2 + VR(2)**2 + VR(3)**2 )
```

```
      VMAS = VMA*VMA
      ALPHA = CRAD*ATAN( VR(3)/VR(1) )
      BETA = CRAD*ASIN( VR(2)/VMA )
      ARRAY(68) = VMA
      ARRAY(72) = ALPHA
      ARRAY(84) = BETA
C
      CALL TRANS ( TMPY, TMPZ, 3, 3 )
      CALL MULT ( PVW, TMPZ, TMPA, 3, 3, 3 )
      CALL MULT ( TMPY, TMPA, TMPZ, 3, 3, 3 )
      CALL ADD ( TMPX, TMPZ, TMPX, 3, 3 )
      BRRAY(68) = SQRT( TMPX(1,1) + TMPX(2,2) + TMPX(3,3) )
      V13S = VR(1)**2 + VR(3)**2
      V13 = SQRT ( V13S )
      VECX(1) = -VR(3)/V13S
      VECX(2) = 0.0
      VECX(3) = VR(1)/V13S
      CALL MULT ( TMPX, VECX, VECY, 3, 3, 1)
      CALL INNER ( VECX, VECY, SIGAS, 3 )
      BRRAY(72) = CRAD*SQRT ( SIGAS )
C
      IF ( V13.NE.0.0 ) THEN
C
      VECX(1) = VR(2)*VR(1)/(VMAS*V13)
      VECX(2) = VR(2)*VR(2)/(VMAS*V13)
      VECX(3) = VR(2)*VR(3)/(VMAS*V13)
      CALL MULT ( TMPX, VECX, VECY, 3, 3, 1 )
      CALL INNER ( VECX, VECY, SIGBS, 3 )
      VECX(1) = 0.0
      VECX(2) = 1.0/V13
      VECX(3) = 0.0
      CALL MULT ( TMPX, VECX, VECY, 3, 3, 1 )
      CALL INNER ( VECX, VECY, SIGS, 3 )
      SIGB = SQRT ( SIGBS + SIGS )
      BRRAY(84) = CRAD*SIGB
C
      ELSE
      END IF
C
C**  MACH NUMBER AND DYNAMIC PRESSURE W/UNCERTAINTY BOUNDS
C
      PDYNMC = 0.5*RHO*VMAS
      XMACH = VMA/VSOUND
      ARRAY(74) = PDYNMC
      ARRAY(77) = XMACH
      CALL MULT ( TMPX, VR, VECX, 3, 3, 1 )
      CALL INNER ( VR, VECX, SIGVS, 3 )
      SIGQS = RHO*RHO*SIGVS
      SIGS = (0.5*VMAS*PRHOPH*SIGRM)**2
      SIGQS = SIGQS + SIGS
      BRRAY(74) = SQRT ( SIGQS )
      VSS = VSOUND*VSOUND
      SIGMS = SIGVS/VSS
      SIGS = (VMA*PVSPH*SIGRM/VSS)**2
      SIGMS = SIGMS + SIGS
      BRRAY(84) = SQRT ( SIGMS )
C
C**  Q-ALPHA AND Q-BETA WITH UNCERTAINTY BOUNDS
C
      QALPHA = PDYNMC*ALPHA
      QBETA = PDYNMC*BETA
      ARRAY(75) = QALPHA
      ARRAY(76) = QBETA
      ALPHAS = ALPHA*ALPHA
      BETAS = BETA*BETA
      PDYNS = PDYNMC*PDYNMC
```

```
      BRRAY(75) = SQRT(ALPHAS*BRRAY(74)**2 + PDYNS*BRRAY(72)**2)
      BRRAY(76) = SQRT(BETAS*BRRAY(74)**2 + PDYNS*BRRAY(84)**2)
C
C**   VEHICLE ATTITUDE AND UNCERTAINTY BOUNDS
C
      CALL SMLT ( CRAD, VEC3, VEC3, 3, 1 )
      CALL ADD ( THTI, VEC3, VEC3, 3, 1 )
      DO 40 I = 1, 3
      VECZ(I) = CRAD*SQRT(TMP3(I,I))
40    CONTINUE
      ARRAY(85) = VEC3(3)
      ARRAY(86) = VEC3(2)
      ARRAY(87) = VEC3(1)
      BRRAY(85) = VECZ(3)
      BRRAY(86) = VECZ(2)
      BRRAY(87) = VECZ(1)
C
C**   (B) AND (I) REFERENCED ACCELERATION AND UNCERTAINTY BOUNDS
C
      CALL ADD ( ABRI, VEC4, VEC1, 3, 1 )
      CALL DIAG ( ABRI, 3, TMPX )
      CALL MULT ( TMPX, VEC5, AB, 3, 3, 1 )
      CALL ADD ( AB, VEC1, VEC1, 3, 1 )
      ARRAY(8) = VEC1(1)
      ARRAY(9) = VEC1(2)
      ARRAY(10) = VEC1(3)
C
      CALL MULT ( CCIB, VEC1, AB, 3, 3, 1 )
C
CRFTSTAT = SQRT( ABRI(1)**2 + ABRI(2)**2 + ABRI(3)**2 )
      AT = SQRT ( AB(1)**2 + AB(2)**2 + AB(3)**2 )
      ARRAY(23) = AB(1)
      ARRAY(24) = AB(2)
      ARRAY(25) = AB(3)
      ARRAY(83) = AT/AGRAV
C
C*    ACCELEROMETER BIAS
C
C     TMP4
C
C*    ACCELEROMETER SCALE FACTOR
C
      CALL DIAG ( ABRI, 3, TMPZ )
      CALL MULT ( TMPZ, TMP5, TMPX, 3, 3, 3 )
C
C*    PLATFORM TILT
C
      CALL SKEW ( ABRI, TMPZ )
      CALL MULT ( TMPZ, TMP3, TMPY, 3, 3, 3 )
C
      CALL ADD ( TMP4, TMPX, TMPZ, 3, 3 )
      CALL ADD ( TMPY, TMPZ, TMPX, 3, 3 )
      BRRAY(8) = SQRT( TMPX(1,1) )
      BRRAY(9) = SQRT( TMPX(2,2) )
      BRRAY(10) = SQRT( TMPX(3,3) )
C
      CALL MULT ( CIB, TMPX, TMPY, 3, 3, 3 )
      CALL MULT ( TMPY, CBI, TMPX, 3, 3, 3 )
      BRRAY(23) = SQRT( TMPX(1,1) )
      BRRAY(24) = SQRT( TMPX(2,2) )
      BRRAY(25) = SQRT( TMPX(3,3) )
      BRRAY(83) = SQRT(BRRAY(23)**2+BRRAY(24)**2+BRRAY(25)**2)/AGRAV
C
C**   FLIGHT PATH ANGLE
C
      CALL INNER ( VBR, VBR, RDOTV, 3 )
```

```
      IF ( VMI.NE.0.0 ) THEN
      GAMMA = CRAD*ASIN( RDOTV/(RM*VMI) )
      ELSE
      GAMMA = 0.0
      END IF
      ARRAY(56) = GAMMA
C
      B = RM*VMI
      RFAC = RDOTV/B
      RFAC1 = RFAC*VMI/RM
      RFAC2 = RFAC*RM/VMI
      DENO = SQRT( B*B - RDOTV*RDOTV )
C
      DO 50 I = 1, 3
      VEC6(I) = ( VBR(I) + RFAC1*RBR(I) )/DENO
      VEC6(I+3) = ( RBR(I) + RFAC2*VBR(I) )/DENO
50    CONTINUE
C
      DO 70 I = 1, 6
      TEMP = 0.0
         DO 60 J = 1, 6
         TEMP = TEMP + PKGN(I,J)*VEC6(J)
60       CONTINUE
      VEC7(I) = TEMP
70    CONTINUE
C
      SIGGMS = 0.0
      DO 80 I = 1, 6
      SIGGMS = SIGGMS + VEC6(I)*VEC7(I)
80    CONTINUE
      SIGGM = CRAD*SQRT ( SIGGMS )
      BRRAY(56) = SIGGM
C
      CALL TMATY ( -XLONG, TMPX )
      CALL TMATP ( XLAT, TMPY )
      CALL MULT ( TMPY, TMPX, TMPZ, 3, 3, 3 )
      CALL MULT ( C1, TMPZ, TMPX, 3, 3, 3 )
      CALL MULT ( CBRIEF, VBR, VECX, 3, 3, 1 )
      CALL MULT ( TMPX, VECX, VECY, 3, 3, 1 )
      IF ( VECY(2).NE.0.0 ) THEN
      SIGMA = CRAD*ATAN2( VECY(1), VECY(2) )
      ELSE
      SIGMA = 90.0
      END IF
      ARRAY(57) = SIGMA
C
      DENO = VECY(1)*VECY(1) + VECY(2)*VECY(2)
      VECX(1) = VECY(2)/DENO
      VECX(2) = VECY(1)/DENO
      VECX(3) = 0.0
C
      CALL TRANS ( TMPX, TMPY, 3, 3 )
      CALL MULT ( TMPY, VECX, VECZ, 3, 3, 1 )
      CALL MULT ( TMP2, VECZ, VECY, 3, 3, 1 )
      CALL INNER ( VECX, VECY, SIGSMS, 3 )
      SIGSM = CRAD*SQRT ( SIGSMS )
      BRRAY(57) = SIGSM
C
C** VEHICLE ROTATION RATE
C
      ARRAY(45) = OMEG(3)
      ARRAY(46) = OMEG(2)
      ARRAY(47) = OMEG(1)
C
C** OUTPUT COMPUTED QUANTITIES
C
```

```
      WRITE(8) (ARRAY(I), I = 1, 90)
C     WRITE(8) (BRRAY(I), I = 1, 90)
      NR = NR + 1
C
      RETURN
C
901   FORMAT ( 7X, 4E15.8 )
902   FORMAT ( 2X, 6E12.4 )
C
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      [illegible]UBRO[illegible]E R[illegible]SE ( [illegible]CH )
C
      DIMENSION ARRAY(90), BRRAY(90)
C
      DATA NARRAY / 90 /
      DATA NR / 16 /
C
10    CONTINUE
C
      READ(8) (ARRAY(I), I = 1, NARRAY)
C
      TIME = ARRAY(1)
C
      IF ( ABS(TIME-TSRCH).LT.1.0E-4 ) THEN
C
C     WRITE(*, 901) TIME
      WRITE(7'NR) (ARRAY(I), I = 1, NARRAY)
      NR = NR + 1
      WRITE(*,901) (ARRAY(I), I = 1, NARRAY)
      WRITE(9,902) ARRAY(1)
      WRITE(9,902) (0.96*ARRAY(8)),(0.96*ARRAY(9)),(0.96*ARRAY(10))
      WRITE(9,902) ARRAY(87),ARRAY(86),ARRAY(85)
      WRITE(9,902) ARRAY(47),ARRAY(46),ARRAY(45)
C
      REWIND 8
      RETURN
      ELSE
C
      GO TO 10
C
      END IF
C
  901 FORMAT( 2X, 6E12.5 )
902   FORMAT( 5X, 3E15.8 )
C
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C     PROGRAM FILTER
      PROGRAM MAIN
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / PREUP / XKM(71), PKM(71,71)
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / DRVITV / XDOT(71), PD(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / LINHMT / H(35,71), NRMEAS
C
      COMMON / NOZDAT / TTIME(500),TRLD(500),TTLD(500)
     1                          ,TRRD(500),TTRD(500)
     2                 ,TPL(500),TPL1(500),TYL1(500)
     3                          ,TPL2(500),TYL2(500)
     4                          ,TPL3(500),TYL3(500)
     5                 ,TRATER(500),TRATEP(500),TRATEY(500)
     6                 ,TROLL(500), TPITCH(500), TYAW(500)
      COMMON / SWIND / TALT(400),TVWX(400),TVWY(400),NALT
      COMMON / SATMOS / TSRHO(400), TSP(400), TSSUND(400), IATMOS
      COMMON / ATMUNC / UWX(400),UWY(400),URHO(400),USP(400),USS(400)
      COMMON / GIMBAL / CCLB1(3,3),CCLB2(3,3),CCLB3(3,3),CCLBA(3,3)
     1                 ,CCLBB(3,3),PLN,RATEC(3),THTC(3)
C
C     BLOCK DATA
C
      COMMON / CONST / A(71), S(71), R(12)
      COMMON / TIMDAT / TMAX, HSTEP, TS, TSTART, TSTOP, TRINT
      COMMON / LAYOUT / RS(3),RA(3),RT1(3),RT2(3),RT3(3),RT4(3),RT5(3)
      COMMON / GEOMET / AREA, DIA, PSMEPS, PSRBPS
      COMMON / SMETBL / PR(4,3,5), YN(4,3), XN(4,2)
     1                , PRZ(4,3,3), PRX(4,2,2), ZN(4,3)
     2                , RPLTAG(3),WDO2TU(3),WDH2TU(3),TVACTU(3)
      COMMON/SRBTBL/TFSA(15),TTAU(15),TPVOL(15),TAT(15),TCSTR(15)
     1              ,TCT(15),NTAU
     2,CM,G,GAM,PE,ABAR,PBAR,EPBAR,RHOP,XLNGTH,RBAR,XMBAR,TMP,PI
     3,DTSEP
      COMMON / STRUCT / PTC1,YWC1,PTC2,YWC2,PTC3,YWC3
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2, CRAD
      COMMON / LAUCOR / OLATD, OLONG, OHT
      COMMON/ASTRON/CUENED(3,3),RNP(3,3),CIBRI(3,3),CBRIB(3,3),TGMTT0
      COMMON / RDRDAT / XNO(3),RLAT(3),RLONG(3),RHT(3),NRDR
      COMMON / ATMOSP / THALT(15), THRHO(15), THP(15), THSUND(15)
      COMMON / IPROPR / XMASSI, XMASS2
      COMMON / APRIOR / ER(3),EVB(3),EQ(4),EO(3)
     1                 ,ESME(9),ESRB(5)
     2                 ,EARO(3),EPLM(3)
     3                 ,EWND(3)
     4                 ,EACB(3),ERDR(3)
      COMMON / NMEASR / NMEAS, IMEAS(5)
      COMMON / IPARAM / ISSME,ISRB,IAERO,IPLUME,IWIND,JACB,JRDR
      COMMON / TYPE / ITYPE
      COMMON / STAGE / ISTAGE
C
      OPEN( UNIT=1, FILE='CONTRL', STATUS='OLD', ACCESS='SEQUENTIAL' )
      IF ( ITYPE.LT.1 ) THEN
```

PRECEDING PAGE BLANK NOT FILMED

PAGE 90 INTENTIONALLY BLANK

```
      OPEN( UNIT=2, FILE='SYNTHT', STATUS='OLD', ACCESS='SEQUENTIAL' )
      ELSE
      OPEN( UNIT=2, FILE='REALME', STATUS='OLD', ACCESS='SEQUENTIAL' )
      END IF
      OPEN( UNIT=4, FILE='METDAT', STATUS='OLD', ACCESS='SEQUENTIAL' )
      OPEN( UNIT=7, FILE='FILOUT', STATUS='NEW', ACCESS='SEQUENTIAL' )
C    1     , FORM='UNFORMATTED' )
      OPEN( UNIT=8, FILE='SSMEOT', STATUS='NEW', ACCESS='SEQUENTIAL' )
      OPEN( UNIT=9, FILE='SRBOUT', STATUS='NEW', ACCESS='SEQUENTIAL' )
C
      DO 10 I = 1, 500
      READ(1,998) TTIME(I),TRLD(I),TTLD(I),TRRD(I),TTRD(I)
      READ(1,999) TPL(I),TPL1(I),TYL1(I),TPL2(I),TYL2(I),TPL3(I),TYL3(I)
      READ(1,998) TRATER(I), TRATEP(I), TRATEY(I)
      READ(1,998) TROLL(I), TPITCH(I), TYAW(I)
10    CONTINUE
C
      CLOSE ( UNIT = 1 )
C
      DO 20 I = 1, NALT
      READ(4,999)TALT(I),TSRHO(I),TSP(I),TSSUND(I),TVWX(I),TVWY(I)
      READ(4,996)URHO(I),USP(I),USS(I),UWX(I),UWY(I)
20    CONTINUE
C
      CLOSE ( UNIT = 4 )
C
      CALL ZERO
      KTEST = TS/HSTEP
      DER(1) = HSTEP
C
      CALL INITIL
C
      CALL PROPAF
C
50    CONTINUE
C
      CALL RK4FIL
C
      TIME = VAR(1)
C
      KS = KS + 1
      IF( KS - KTEST ) 200, 100, 100
100   CONTINUE
      KS = 0
C
      CALL GETDAT( TIME )
C
      CALL UPDATE
C
      GO TO 700
C
200   CONTINUE
C
700   CONTINUE
C
```

```
      IF ( TIME - TMAX ) 50, 800, 800
C
800   CONTINUE
C
      IF ( TIME - TSTOP ) 50, 900, 900
C
900   CONTINUE
C
      WRITE(6,991) TIME
      WRITE(6,991) (XKM(II), II = 1, 71)
      WRITE(6,991) (PKM(II,II), II = 1, 71)
C
      CLOSE( UNIT=2 )
      CLOSE( UNIT=7 )
      CLOSE( UNIT=8 )
      CLOSE( UNIT=9 )
C
991   FORMAT ( 5X, 3E15.8 )
995   FORMAT(5X, 2E15.8)
996   FORMAT( 20X, 5E15.8 )
997   FORMAT(3X, 5E15.8 )
998   FORMAT( 5X, 5E15.8 )
999   FORMAT( 5X, 7E15.8 )
C
      STOP
      END
      SUBROUTINE GETDAT( TIME )
C
      COMMON / ACMEAS / ACM(3)
C
      COMMON / MEMEAS / YMSME(6,3)
      COMMON / SRMEAS / YMSRB(2)
      COMMON / RDMEAS / AZM(3), ELM(3), RNGM(3)
      COMMON / TYPE / ITYPE
C
      IF( ITYPE.LT.1 ) THEN
C
C
      ELSE
C
      READ( 2, 901 ) (ACM(I), I = 1, 3)
      READ( 2, 902 ) ((YMSME(I,J), I = 1, 6), J = 1, 3)
      READ( 2, 901 )(AZM(I), ELM(I), RNGM(I), I = 1, 3)
      READ( 2, 903 ) YMSRB(1)
      READ( 2, 903 ) YMSRB(2)
      DO 10 I = 1, 3
      ACM(I) = ACM(I)/0.96
10    CONTINUE
      IF ( TIME.LT.20.                    ) THEN
      CALL ZEROM ( ACM, 3, 1 )
      ELSE
      END IF
      IF ( TIME.GT.70. ) THEN
C     RNGM(3) = 0.0
      ELSE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      END IF
C
      END IF
C
      RETURN
901   FORMAT( 5X,3E15.8 )
902   FORMAT( 5X,6E15.8 )
903   FORMAT( 5X, E15.8 )
      END
      SUBROUTINE ZERO
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
C
      DER(1) = 0.0
      VAR(1) = 0.0
      DO 10 I = 1, 2699
      DER(I+1) = 0.0
      VAR(I+1) = 0.0
      TEMP(I) = 0.0
      K = I + 2699
      TEMP(K) = 0.0
   10 CONTINUE
C
      RETURN
      END
```

```
      BLOCK DATA
C
      COMMON / CONST / A(71), S(71), R(12)
      COMMON / TIMDAT / TMAX, HSTEP, TSAMP, TSTART, TSTOP, TRINT
      COMMON / LAYOUT / RS(3),RA(3),RT1(3),RT2(3),RT3(3),RT4(3),RT5(3)
      COMMON / GEOMET / AREA, DIA, PSMEPS, PSRBPS
      COMMON / SMETBL / PR(4,3,5), YN(4,3), XN(4,2)
     1                , PRZ(4,3,3),PRX(4,2,2),ZN(4,3)
     2                , RPLTAG(3), WDO2TU(3), WDH2TU(3), TVACTU(3)
      COMMON/SRBTBL/TFSA(15),TTAU(15),TPVOL(15),TAT(15),TCSTR(15)
     1            ,TCT(15),NTAU
     2,CM,G,GAM,PE,ABAR,PBAR,EPBAR,RHOP,XLNGTH,RBAR,XMBAR,TEMP,PI
     3,DTSEP
      COMMON / STRUCT / PTC1,YWC1,PTC2,YWC2,PTC3,YWC3
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2, CRAD
      COMMON / LAUCOR / OLATD, OLONG, OHT
      COMMON/ASTRON/CUENED(3,3),RNP(3,3),CIBRI(3,3),CBRIB(3,3),TGMTT0
      COMMON / RDRDAT / XNO(3),RLAT(3),RLONG(3),RHT(3),NRDR
      COMMON / ATMOSP / THALT(15), THRHO(15), THP(15), THSUND(15)
      COMMON / IPROPR / XMASSI, XMASS2
      COMMON / APRIOR / ER(3),EVB(3),EQ(4),EO(3)
     1                 ,ESME(9),ESRB(5)
     2                 ,EARO(3),EPLM(3)
     3                 ,EWND(3)
     4                 ,EACB(3),ERDR(3)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / LINHMT / H(35,71), NRMEAS
      COMMON / TYPE / ITYPE
      COMMON / NMEASR / NMEAS, IMEAS(5)
      COMMON / IPARAM / ISSME,ISRB,IAERO,IPLUME,IWIND,JACB,JRDR
      COMMON / STAGE / ISTAGE
      COMMON / STGDAT / R2(3),VB2(3),Q2(4),OMEGA2(3)
      COMMON / SATMOS / TSRHO(400), TSP(400), TSSUND(400), IATMOS
      COMMON / SWIND / TALT(400), TVWX(400), TVWY(400), NALT
C
      DATA ITYPE / 1 /
      DATA ISTAGE / 1 /
CSTG2 DATA TMAX,HSTEP,TSAMP,TSTART,TRINT / 515.,0.5,1.0,129.,170. /
      DATA TMAX,HSTEP,TSAMP,TSTART,TRINT / 0.5,0.2,1.0,0.0,15. /
      DATA TSTOP / 999. /
      DATA NMEAS / 5 /
CSTG2 DATA IMEAS / 1, 0, 3, 4, 0 /
      DATA IMEAS / 1, 0, 3, 4, 5 /
      DATA IATMOS / 1 /
      DATA NALT / 400 /
      DATA NS, NPAR / 13, 58 /
      DATA F / 5041*0.0 /
CSTG2 DATA NRMEAS / 30 /
      DATA NRMEAS / 32 /
      DATA H / 2485*0.0 /
C
C*** GEOMETRICAL DATA
C
      DATA AREA, DIA, PSMEPS, PSRBPS / 2690.0,107.525,44.3778,122.598 /
      DATA RT1 / -100.8333, 0.0, -31.625 /
```

```
      DATA RT2 / -102.7642, -4.4167, -23.2617 /
      DATA RT3 / -102.7642, 4.4167, -23.2617 /
      DATA RT4 / -119.5417, -20.875, 0.0 /
      DATA RT5 / -119.5417, 20.875, 0.0 /
C
C*** SRB TABULAR DATA
C
      DATA NTAU / 15 /
      DATA TFSA/.4628E+6,.4675E+6,.4756E+6,.4809E+6,.4457E+6,.4165E+6,
     1         .3978E+6,.4013E+6,.4115E+6,.4112E+6,.3795E+6,
     2         .3534E+6,.3152E+6,.816E+5,0.0/
      DATA TPVOL/.15E+7,.3E+7,.45E+7,.52E+7,.116E+8,.181E+8,
     1          .266E+8,.335E+8,.435E+8,.523E+8,.751E+8,
     2          1.E+8,1.46E+8,3.E+9,3.E+9/
      DATA TAT/.2278E+4,.2287E+4,.2296E+4,.2316E+4,.2334E+4,.2351E+4,
     1        .2366E+4,.2381E+4,.2396E+4,.2412E+4,.2426E+4,
     2        .2439E+4,.2451E+4,.2458E+4,.2458E+4/
      DATA TCSTR/.6044E+5,.6045E+5,.6046E+5,.6046E+5,.6039E+5,.6031E+5,
     1          .6024E+5,.6024E+5,.6025E+5,.6025E+5,.6017E+5,
     2          .601E+5,.6005E+5,.5903E+5,.5813E+5/
      DATA TTAU/0.,1.94,3.96,8.02,12.04,15.94,
     1         19.77,23.51,27.27,31.00,34.38,
     2         37.94,41.05,42.93,43.2/
      DATA TCT / 1.7347,1.7352,1.7345,1.7325,1.7306,1.7283,
     1         1.7270,1.7263,1.7252,1.7234,1.7221,
     2         1.7210,1.7185,1.6890,1.722 /
      DATA CM,G,GAM,PE,ABAR,PBAR,EPBAR,RHOP /
     1    .987,386.09,1.1382,5.,.3680,625.,.35,.063456/
      DATA XLNGTH,RBAR,XMBAR,TEMP,PI/1350.,1545.4,28.319,6092.6,3.1416/
      DATA DTSEP / 4.9 /
C
C*** SSME TABULAR DATA
C
      DATA RPLTAG / 3*3006. /
      DATA WDO2TU / 889.59, 892.04, 891.62 /
      DATA WDH2TU / 147.68, 148.08, 148.02 /
      DATA TVACTU / 468028.,468968.,469079./
C61C  DATA WDO2TU / 892.41,892.55,892.29 /
C61C  DATA WDH2TU / 148.43,148.45,148.40 /
C61C  DATA TVACTU / 469250.,470091.,469741./
C
      DATA PR/.450195,-.0717593,.0250795,0.0,-.849847,3*0.0,.206968,
     1 -.0596156,.0223651,0.0,1.448318,3*0.0,4*0.0,
     2 1.196201,3*0.0,-1.49969,.14682,
     3 6*0.0,-.224458,.05352,-.00757568,0.0,1.0,7*0.0,-.0621559,3*0.0,
     4 3.11118,-.175361,6*0.0,724.708,-138.313,40.2932,0.0/
      DATA YN/-5.29485,918.6935,-45.4427,24.8829,-.98861,153.563,
     1 -8.18426,4.4164,-2668.66,479997.,-17680.5,11576.7/
      DATA XN/-.27717,2.97447,-1.06196,0.0,.00685,.693611,2*0.0/
      DATA PRZ/-.142009,.420972,-.515623,.208368,1.09595E-2,
     1 -3.09852E-3,2.46712E-3,-2.80349E-4,-8.86243,5.79738,
     2 -4.53574,.993182,.210435,-1.10734,.459839,0.0,-8.09872E-2,
     3 .394479,-.160137,0.0,43.7468,-245.92,101.362,0.0,-1.38568,
     4 8.41244,-3.11896,0.0,.549916,-3.00494,1.08354,0.0,-277.342,
     5 1856.63,-677.963,0.0/
```

```
      DATA PRX/-4.2873,5.52804,-2.64673,0.0,8.35593,-18.3258,
     1 9.85097,0.0,-6.3079E-3,3*0.0,2.26311E-4,-5.49228E-4,
     2 3.19372E-4,0.0/
      DATA ZN / 400., 15000., 0.0, 0.0, -100.0, 350.,
     1 0.0, 0.0, 37.3, 5.0, 0.0, 0.0/
C
C*** INITIAL VEHICLE MASS, POSITION, VELOCITY AND ATTITUDES
C
      DATA XMASSI, XMASS2 / 4.520E+6, 1.540E+6 /
C61C  DATA XMASSI, XMASS2 / 4.512E+6, 1.538E+6 /
C61A  DATA XMASSI, XMASS2 / 4.491E+6, 1.538E+6 /
C51B  DATA XMASSI, XMASS2 / 4.502E+6, 1.539E+6 /
C
      DATA R2,VB2,Q2,OMEGA2 /3*0.0,3*0.0
     1                      ,4*0.0,3*0.0 /
C61C  DATA R2,VB2,Q2,OMEGA2/-.53164E+5,.36720E+6,-.21062E+8
C61C 1                     ,4450., 70.,-0.
C61C 2                     ,4*0.0,3*0.0 /
C61A  DATA R2,VB2,Q2,OMEGA2 / .135991E+6,.279625E+6,-.21073438E+8
C61A 1                      ,4210.,-100.,20.
C61A 2                      ,.05367,.9174,.2748,-.2559,3*0.0 /
C51B  DATA R2, VB2, Q2, OMEGA2 / .141696E+6,.274866E+6,-.210687E+8
C51B 1                    ,4230.,2*0.0
C51B 2                    ,.5367E-1,.9174E+0,.2748E+0,-.2559E+0.3*0.0/
C
      DATA ISSME,ISRB,IAERO,IPLUME,IWIND,JACB,JRDR/
     1 1,1,3*1,2*1 /
CSTG21 1,0,3*1,2*1 /
C
      DATA NRDR / 3 /
CSTG2 DATA XNO / 0.0, 0.0, 0.0003 /
C
C*** RADAR SITE COORDINATES
C
C61C  DATA RLAT / 28.226553, 28.528886, 28.626091 /
C61C  DATA RLONG / -80.599293, -80.590226, -80.682809 /
C61C  DATA RHT / 65.0, 37.0, 52.0 /
      DATA RLAT / 28.226391, 28.463168, 28.625932 /
      DATA RLONG / -80.599287, -80.583111, -80.682805 /
      DATA RHT / -43.8, -48.6, -58.1 /
C
C*** PROCESS NOISE VARIANCES
C
      DATA A / 3*0.0, 2*0.05, 0.1, 4*1.E-4, 3*0.0
     *, 0.,1.E-1,1.E+0,2*1.E-1,1.E+2,1.E+1,1.E+0,1.E+1
     +, 0.,1.E-1,1.E+0,2*1.E-1,1.E+2,1.E+1,1.E+0,1.E+1
     1, 0.,1.E-1,1.E+0,2*1.E-1,1.E+2,1.E+1,1.E+0,1.E+1
     2, 1.E-2,2*1.E-2,5.E+4,2*1.5E+5,2*1.E+0,1.E+0,3*1.E-2
     3, 1.E-2,1.E-2,1.E+2,1.E-2,1.E-2,1.E+2,1.E-2,1.E-2,1.E+2
     4, 2*0.0,3.E-4,4.E-3,2.E+2,2*0.0,3.E-4,4.E-3,2.E+2/
C
C*** MEASUREMENT NOISE VARIANCES
C     DATA R/ 12*1.E+32 /
      DATA R / .25
     1       ,1.E+16
```

```
     2         ,1.E+4,1.E+1,1.E+0,1.E+5,1.E+4,1.E+1
     3         ,1.E-3,1.E-3,1.E+4
     4         ,1.E+3 /
C
C*** INITIAL STATE & PARAMETER UNCERTAINTIES, ONE-SIGMA VALUES
C
      DATA ER / 3*100. /
      DATA EVB / 3*1. /
      DATA EQ / 4*1.E-3 /
      DATA EO / 3*0.0 /
C
      DATA ESME /1.E+2,1.E-1,1.E+0,2*1.E-1,1.E+2,1.E+1,1.E+0,1.E+1 /
      DATA ESRB / 1.E+3,1.E-2,3.E-4,4.E-3,1.E+2 /
      DATA EARO / 1.E-2,2*1.E-2 /
      DATA EPLM / 5.E+4,2*1.5E+5 /
      DATA EWND / 2*1.E+0,1.E+0 /
      DATA EACB / 3*1.E-2 /
      DATA ERDR / 1.E-2,1.E-2,1.E+2 /
C
C*** BODY REFERENCED COORDINATES OF IMU LOCATION
C
      DATA RS / -15.387, 0.0, -29.875 /
C
C*** BODY REFERENCED AERODYNAMIC REFERENCE CENTER
C
      DATA RA / 3*0.0 /
C
C*** TABULAR ATMOSPHERIC MODEL DATA FOR ALTITUDED ABOVE 60k FEET
C
      DATA THALT / 60000.0,70000.0,80000.0,90000.0,100000.0
     1,110000.0,120000.0,130000.0,140000.0,150000.0
     2,160000.0,170000.0,180000.0,190000.0,200000.0 /
C
      DATA THRHO / .2256E-3,.1392E-3,.8571E-4,.5315E-4,.3318E-4
     1,.2066E-4,.1290E-4,.8191E-5,.5281E-5,.3456E-5
     2,.2322E-5,.1590E-5,.1105E-5,.7654E-6,.5277E-6 /
C
      DATA THP / 151.,93.73,58.51,36.78,23.27
     1,14.84,9.602,6.31,4.206,2.842
     2,1.942,1.33,.9084,.6155,.4135 /
C
      DATA THSUND / 967.6,970.4,977.1,983.4,990.4
     1,1002.,1020.,1038.,1055.,1072.
     2,1081.,1081.,1072.,1060.,1047. /
C
C*** EARTH MODEL AND CONSTANT DATA
C
      DATA RE,FLAT,OMEGE,XMU,XJ2/20925606.,298.257224,.7292115E-4
     1, 1.4076468E+16, .10827E-2 /
      DATA CRAD / 57.295779 /
      DATA RNP / -1.0,3*0.0,-1.0,3*0.0,1.0 /
      DATA TGMTT0 / 0.0 /
      DATA CUENED / 2*0.0,-1.0,0.0,1.0,0.0,0.0,1.0,2*0.0 /
      DATA CBRIB / 2*0.0,1.0,0.0,0.0,1.0,0.0,-1.0,2*0.0 /
C
```

```
C*** LAUNCH SITE COORDINATES
C
      DATA OLATD, OLONG, OHT / 28.62666, -80.62162 , 50. /
C
C*** MAIN ENGINE THRUST STRUCTURAL DEFLECTION COEFFICIENTS
C
      DATA PTC1,YWC1,PTC2,YWC2,PTC3,YWC3/-0.4,0.1,-0.3,-0.4,-1.4,0.8/
C
      END
```

```
      SUBROUTINE INITIL
C
      DIMENSION OMEG(3),THT(3),VEC(3),TMP1(3,3),TMP2(3,3),TMP3(3,3)
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / TIMDAT / TMAX, HSTEP, TS, TSTART, TSTOP, TRINT
      COMMON / CONST / A(71), S(71), RR(12)
      COMMON / LAUCOR / OLATD, OLONG, OHT
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2, CRAD
      COMMON/ASTRON/CUENED(3,3),RNP(3,3),CIBRI(3,3),CBRIB(3,3),TGMTT0
      COMMON / APRIOR / ER(3),EVB(3),EQ(4),EO(3)
     1                  ,ESME(9),ESRB(5)
     2                  ,EARO(3),EPLM(3)
     3                  ,EWND(3)
     4                  ,EACB(3),ERDR(3)
      COMMON / STATES / R(3), VB(3), Q(4), OMEGA(3)
      COMMON / RDOTI / RDI(3)
      COMMON / STAGE / ISTAGE
      COMMON / STGDAT / R2(3),VB2(3),Q2(4),OMEGA2(3)
      COMMON / PREUP / XKM(71), PKM(71,71)
C
      CALL ECPOS ( OLATD, OLONG, OHT, VEC(1), VEC(2), VEC(3) )
      CALL ECPOS ( OLATD, OLONG, 0.0, X0, Y0, Z0 )
C
      E = 1.0/FLAT
      RDMAG = SQRT( X0**2 + Y0**2 )
      RMAG = SQRT( VEC(1)**2 + VEC(2)**2 + VEC(3)**2 )
      DEV = CRAD*2.*E*(1.-E/2.)*(RDMAG/RMAG)*SIN((180.0-OLATD)/CRAD)
      OLATC = OLATD - DEV
C
      CALL CIEFMX ( 0.0, TMP1 )
      CALL TMATY ( -OLONG, TMP2 )
      CALL MULT ( TMP2, TMP1, TMP3, 3, 3, 3 )
      CALL TMATP ( OLATD, TMP1 )
      CALL MULT ( TMP1, TMP3, TMP2, 3, 3, 3 )
      CALL MULT ( CUENED, TMP2, CIBRI, 3, 3, 3 )
C
      RDI(1) = 0.0
      RDI(3) = 0.0
      RAD = SQRT ( VEC(1)**2 + VEC(2)**2 )
      RDI(2) = OMEGE*RAD
C
      IF ( ISTAGE.EQ.1 ) THEN
C
      CALL MULT ( CIBRI, VEC, R, 3, 3, 1 )
      WRITE(*, 901) (R(II), II = 1, 3)
C
CRATE CALL SMLT ( OMEGE, OMEG, VEC, 3, 1 )
CRATE CALL MULT ( TMP1, VEC, OMEGA, 3, 3, 1 )
C
      IF( CBRIB(3,3) ) 10, 20, 10
10    CONTINUE
      THT(1) = CRAD*ATAN2( CBRIB(2,3), CBRIB(3,3) )
20    CONTINUE
      IF( CBRIB(3,3).EQ.0.0 ) THT(1) = 0.0
```

```
      THT(2) = CRAD*ASIN( -CBRIB(1,3) )
      IF( CBRIB(1,1) ) 30, 40, 30
30    CONTINUE
      THT(3) = CRAD*ATAN2( CBRIB(1,2), CBRIB(1,1) )
40    CONTINUE
      IF( CBRIB(1,1).EQ.0.0 ) THT(3) = 0.0
C
      CALL E2QUAT ( THT, Q )
C
      ELSE
C
      DO 60 I = 1, 3
      R(I) = R2(I)
      VB(I) = VB2(I)
      Q(I) = Q2(I)
CRATE OMEGA(I) = OMEGA2(I)
60    CONTINUE
      Q(4) = Q2(4)
      WRITE(*, 901) (R(I), I = 1, 3)
      WRITE(*, 901) (VB(I), I = 1, 3)
      WRITE(*, 901) (Q(I), I = 1, 4)
C
      END IF
C
      CALL ZEROM( PKM, 71, 71 )
      CALL ZEROM( XKM, 71, 1 )
      CALL ZEROM( S, 71, 1 )
C
      VAR(1) = TSTART
C
      DO 100 I = 1, 3
      XKM(I) = R(I)
      VAR(I+1) = R(I)
      PKM(I,I) = ER(I)**2
      XKM(I+3) = VB(I)
      VAR(I+4) = VB(I)
      PKM(I+3,I+3) = EVB(I)**2
CTHT  XKM(I+6) = Q(I)
CTHT  VAR(I+7) = Q(I)
      PKM(I+6,I+6) = EQ(I)**2
CRATE XKM(I+10) = OMEGA(I)
CRATE VAR(I+11) = OMEGA(I)
CRATE PKM(I+10,I+10) = EO(I)**2
      S(I+6) = A(I+6)**2
100   CONTINUE
C     XKM(10) = Q(4)
C     VAR(11) = Q(4)
      PKM(10,10) = EQ(4)**2
      S(10) = A(10)**2
C
      DO 120 I = 1, 9
      IP13 = I + 13
      IP13P9 = IP13 + 9
      IP22P9 = IP13P9 + 9
      PKM(IP13,IP13) = ESME(I)**2
```

```
      PKM(IP13P9,IP13P9) = ESME(I)**2
      PKM(IP22P9,IP22P9) = ESME(I)**2
120   CONTINUE
C
      DO 130 I = 1, 5
      IP61 = I + 61
      IP61P5 = IP61 + 5
      PKM(I+61,I+61) = ESRB(I)**2
      PKM(IP61P5,IP61P5) = ESRB(I)**2
130   CONTINUE
C
      DO 140 I = 1, 3
      PKM(I+40,I+40) = EARO(I)**2
140   CONTINUE
C
      DO 150 I = 1, 3
      PKM(I+43,I+43) = EPLM(I)**2
150   CONTINUE
C
      DO 160 I = 1, 3
      PKM(I+46,I+46) = EWND(I)**2
160   CONTINUE
C
      DO 170 I = 1, 3
      PKM(I+49,I+49) = EACB(I)**2
170   CONTINUE
C
      DO 180 I = 1, 3
      PKM(I+52,I+52) = ERDR(I)**2
      PKM(I+55,I+55) = ERDR(I)**2
      PKM(I+58,I+58) = ERDR(I)**2
180   CONTINUE
C
      L = 0
      DO 200 I = 1, 71
      DO 200 J = I, 71
      L = L + 1
      VAR(71+L+1) = PKM(I,J)
200   CONTINUE
C
      RETURN
C
901   FORMAT( 5X, 3E15.8 )
C
      END
```

```
      SUBROUTINE PROPAF
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / DRVITV / XDOT(71), PD(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / TYPE / ITYPE
C
      NDERV = NS + NPAR
      NMTRX = (NS+NPAR)*( (NS+NPAR+1) )/2
      NDER = NDERV
      IF ( ITYPE.GT.0 ) NDER = NDERV + NMTRX
C
      DO 10 I = 1, (NS+NPAR)
      X(I) = VAR(I+1)
10    CONTINUE
C
      CALL XDVEC
      CALL ZDVEC
C
      DO 20 I = 1, (NS+NPAR)
      DER(I+1) = XDOT(I)
20    CONTINUE
C
      IF ( ITYPE.LT.1 ) GO TO 50
C
      L = 0
      DO 30 I = 1, (NS+NPAR)
      DO 30 J = I, (NS+NPAR)
      L = L + 1
      P(I,J) = VAR(NS+NPAR+L+1)
      P(J,I) = P(I,J)
30    CONTINUE
C
      CALL PDMTRX
C
      L = 0
      DO 40 I = 1, (NS+NPAR)
      DO 40 J = I, (NS+NPAR)
      L = L + 1
      DER(NS+NPAR+L+1) = PD(I,J)
40    CONTINUE
C
50    CONTINUE
C
      RETURN
C
901   FORMAT( 5X, 3I5 )
902   FORMAT( 5X, 5E15.8 )
C
      END
      SUBROUTINE RK4FIL
      REAL K1(2700), K2(2700), K3(2700), K4(2700)
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
C
```

```
      DO 100 I = 1, NDER
      TEMP(I) = VAR(I+1)
  100 CONTINUE
C
      TME = VAR(1)
      DO 10 I = 1, NDER
      K1(I) = DER(1)*DER(I+1)
   10 CONTINUE
C
      DELT = 0.5*DER(1)
      TME = TME + DELT
      VAR(1) = TME
C
      DO 20 I = 1, NDER
      K = I + NDER
      TEMP(K) = TEMP(I) + 0.5*K1(I)
      VAR(I+1) = TEMP(K)
   20 CONTINUE
C
      CALL PROPAF
C
      DO 30 I = 1, NDER
      K2(I) = DER(1)*DER(I+1)
      K = I + NDER
      TEMP(K) = TEMP(I) + 0.5*K2(I)
      VAR(I+1) = TEMP(K)
   30 CONTINUE
C
      CALL PROPAF
C
      TME = TME + DELT
      VAR(1) = TME
C
      DO 40 I = 1, NDER
      K3(I) = DER(1)*DER(I+1)
      K = I + NDER
      TEMP(K) = TEMP(I) + K3(I)
      VAR(I+1) = TEMP(K)
   40 CONTINUE
C
      CALL PROPAF
C
      DO 50 I = 1, NDER
      K4(I) = DER(1)*DER(I+1)
      VAR(I+1) = TEMP(I)+0.16666667*(K1(I)+2.0*K2(I)+2.0*K3(I)+K4(I))
   50 CONTINUE
C
      CALL PROPAF
C
      RETURN
      END
```

```
      SUBROUTINE COOR ( RXE, RYE, RZE, PHI, LAMDA, HTI )
      REAL LAT, LONG, LAMDA
C
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2, CRAD
C
      DATA EPS / 1.E-9 /
C
      LAMDA = CRAD*ATAN2( RYE, RXE )
      CLAM = COS( LAMDA/CRAD )
      E = 1.0/FLAT
      OMES = ( 1.0 - E )**2
C
      DO 100 I = 1, 50
C
      PHII = PHI
C
      CPHI = COS( PHII/CRAD )
      SPHI = SIN( PHII/CRAD )
C
      ASQ = CPHI**2 + OMES*SPHI**2
      A = SQRT( ASQ )
      ASQSQ = ASQ*ASQ
C
      HTI = RXE/(CPHI*CLAM) - RE/A
C
      FPHI = RZE - SPHI*(RE*OMES/A + HTI)
      DFDP = -CPHI*((RE*OMES/A)*(1.0 + (E*(E-2.0)*SPHI**2)/ASQSQ)+HTI)
      DPHI = (FPHI/DFDP)*CRAD
C
      PHI = PHII - DPHI
C
      IF ( ABS( DPHI ) - EPS ) 200, 200, 100
C
  100 CONTINUE
C
  200 CONTINUE
C
      IF ( HTI.LT.0.0 ) HTI = 1.0
C
      RETURN
      END
      SUBROUTINE ECPOS ( LAT, LONG, HT, RXE, RYE, RZE )
      REAL LAT, LONG
C
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2, CRAD
C
      CLAT = COS( LAT/CRAD )
      SLAT = SIN( LAT/CRAD )
      CLONG = COS( LONG/CRAD )
      SLONG = SIN( LONG/CRAD )
C
      E = 1.0/FLAT
      OMES = ( 1.0 - E )**2
      ASQ = CLAT**2 + OMES*SLAT**2
      A = SQRT( ASQ )
```

```
C
      R = RE/A + HT
      S = RE*OMES/A + HT
C
      RXE = R*CLAT*CLONG
      RYE = R*CLAT*SLONG
      RZE = S*SLAT
C
      RETURN
      END
      SUBROUTINE AGRAV ( R )
      DIMENSION R(3)
C
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2, CRAD
      COMMON / EARTH / GRAVI(3), AGMX(3,3)
C
      RMAGS = R(1)*R(1) + R(2)*R(2) + R(3)*R(3)
      RMAG = SQRT( RMAGS )
      RMAGQ = RMAGS*RMAG
C
      RRS = (RE/RMAG)**2
      RZS = (R(3)/RMAG)**2
      XMUX = XMU*(1.0+1.5*XJ2*RRS*(1.0-5.0*RZS))
      XMUY = XMUX
      XMUZ = XMU*(1.0+1.5*XJ2*RRS*(3.0-5.0*RZS))
C
      GRAVI(1) = -XMUX*R(1)/RMAGQ
      GRAVI(2) = -XMUY*R(2)/RMAGQ
      GRAVI(3) = -XMUZ*R(3)/RMAGQ
C
      FAC1 = XMU*XJ2*RRS*( 2.*(1.-5.*RZS) - 1. )/RMAGQ
      FAC4 = XMU*XJ2*RRS*( 2.*(1.-5.*RZS) + 4. )/RMAGQ
C
        AGMX(1,1) = -XMUX/RMAGQ + 3.0*(R(1)/RMAG)**2*(XMUX/RMAGQ + FAC1)
        AGMX(2,1) = 3.0*(R(1)*R(2)/RMAGS)*(XMUX/RMAGQ + FAC1)
        AGMX(3,1) = 3.0*(R(1)*R(3)/RMAGS)*(XMUX/RMAGQ + FAC4)
C
        AGMX(1,2) = AGMX(2,1)
        AGMX(2,2) = -XMUY/RMAGQ + 3.0*(R(2)/RMAG)**2*(XMUY/RMAGQ + FAC1)
        AGMX(3,2) = 3.0*(R(2)*R(3)/RMAGS)*(XMUY/RMAGQ + FAC4)
C
        AGMX(1,3) = AGMX(3,1)
        AGMX(2,3) = AGMX(3,2)
        AGMX(3,3) = -AGMX(1,1) - AGMX(2,2)
C
        RETURN
        END
      SUBROUTINE CIEFMX ( TIME, CIEF )
C
      DIMENSION CIEF(3,3)
C
      COMMON / EDATA / RE, FLAT, OMEGAE, XMU, XJ2, CRAD
C
      THTE = OMEGAE*CRAD*TIME
      STHT = SIN( THTE/CRAD )
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CTHT = COS( THTE/CRAD )
C
      CIEF(1,1) =  CTHT
      CIEF(2,1) = -STHT
      CIEF(3,1) =  0.0
      CIEF(1,2) =  STHT
      CIEF(2,2) =  CTHT
      CIEF(3,2) =  0.0
      CIEF(1,3) =  0.0
      CIEF(2,3) =  0.0
      CIEF(3,3) =  1.0
C
      RETURN
      END
```

```
      SUBROUTINE XDVEC
C
      DIMENSION RD(3),VBD(3),QD(4),OCV(3),QDMTRX(4,4),TSSME(3),TSRB(3)
     1         ,CEFLL(3,3),CBRIEF(3,3),VA(3),AERO(3),THRSTL(3),THRSTS(3)
     2         ,OUT(3),VEC(3),TMP1(3,3),TMP2(3,3),TMP3(3,3)
C    3         ,OMEGAD(3),DRT(3),DRA(3),DRACFP(3),DRACCF(3),TPLUM(3)
C    4         ,OCIO(3),TTHRST(3),TTSME(3),TTSRB(3)
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / TIMDAT / TMAX, HSTEP, TSAMP, TSTART, TSTOP, TRINT
      COMMON / CONST / A(71), S(71), RR(12)
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / DRVITV / XDOT(71), PD(71,71)
      COMMON / STATES / R(3), VB(3), Q(4), OMEGA(3)
      COMMON / GIMBAL / CCLB1(3,3),CCLB2(3,3),CCLB3(3,3),CCLBA(3,3)
     1                 ,CCLBB(3,3),PL,RATEC(3),THTC(3)
      COMMON / PROPER / XMASS, OMASST, XIMTRX(3,3), XIMATI(3,3), RCG(3)
      COMMON / ENDOAT / RHO, PS, VSOUND, XMACH, PRHOH, PPSH
      COMMON / LAYOUT / RS(3),RA(3),RT1(3),RT2(3),RT3(3),RTA(3),RTB(3)
      COMMON / GEOMET / AREA, DIA, PSMEPS, PSRBPS
      COMMON / AERO / CF0(3),CFALP(3),CFBET(3),CFQ(3,3),CM0(3),CMALP(3)
     1               ,CMBET(3),CMQ(3,3),CF(3),CM(3)
      COMMON / PLUME / FPLUME(3),FPALP(3),FPBET(3),TPLUME(3),TPALP(3)
     1                ,TPBET(3),FPALT(3),TPALT(3),FPDPL(3),TPDPL(3)
      COMMON / TMAT / CBI(3,3), CIB(3,3), CLLB(3,3), CEFBRI(3,3)
      COMMON / VARIAB / VR(3), VW(3), VWGRAD(3)
      COMMON / COMBO / AVSO2M,AVSDO2,RAV,RAVD,RAVOM,QA,QAD,QAOM,OOMASS
     1                ,RAOM,QOM,QDYN,RHOA,RAD,OODIA
      COMMON / APARTL / PAMBRI(3,3),PAMBVB(3,3),ACC(3),THRUST(3)
      COMMON /TPERDV/TSSME1(3),TSSME2(3),TSSME3(3),TSRBA(3),TSRBB(3)
      COMMON / TTPRDV/TTSME1(3),TTSME2(3),TTSME3(3),TTSRBA(3),TTSRBB(3)
      COMMON / AVECTR / AVEC(3)
      COMMON / VEHPOS / REF(3)
      COMMON / SMEDAT / TVACL(3),XISPL(3),WDO2H(3),WDH2H(3),WD(3)
     1                 ,CVTVCL(3),CVISPL(3),CVWDO2(3),CVWDH2(3)
     2                 ,CVWD(3),CVWDGO(3),CVWDGH(3),PCTAG(3),OMASL(3)
      COMMON / SRBDAT / TVACS(2),XISPS(2),XMD(2),AEXIT(2)
     1                 ,CVPOH(2),CVTVCS(2),CVISPS(2),CVMD(2),CVAE(2)
     2                 ,OMASS(2)
      COMMON / EARTH / GRAV(3), AGMX(3,3)
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2, CRAD
      COMMON / RDOTI / RDI(3)
      COMMON / ASTRON/CUENED(3,3),RNP(3,3),CIBRI(3,3),CBRIB(3,3),TGMTT0
      COMMON / TYPE / ITYPE
      COMMON / STAGE / ISTAGE
C
      DATA JUMP / 0 /
C
      TIME = VAR(1)
C
      CALL CONTRL ( TIME, PL )
C
      DO 100 I = 1, 3
      R(I) = X(I)
      VB(I) = X(I+3)
```

```
      OMEGA(I) = RATEC(I)
100   CONTINUE
C
C   R-DOT EQUATION
C
      CALL CBIMX ( THTC, CBI )
C
      IF ( JUMP ) 510, 510, 530
510   CONTINUE
C
      CALL E2QUAT ( THTC, QD )
      DO 520 I = 1, 4
      VAR(I+6+1) = QD(I)
      X(I+6) = QD(I)
520   CONTINUE
C
530   CONTINUE
C
      IF ( TIME - TRINT ) 580, 540, 540
540   CONTINUE
C
      IF ( JUMP ) 550, 550, 560
550   CONTINUE
C
      WRITE(*, 998) TIME, TRINT
      WRITE(*, 998) (QD(I), I = 1, 4)
      JUMP = 1
C
560   CONTINUE
C
      DO 570 I = 1, 4
      Q(I) = X(I+6)
570   CONTINUE
C
      CALL CBIMXQ ( Q, CBI )
      CALL QUAT2E ( Q, THTC )
C
580   CONTINUE
C
      CALL MULT ( CBI, VB, RD, 3, 3, 1 )
C
      CALL CIEFMX ( TIME, TMP1 )
      CALL TRANS ( TMP1, TMP2, 3, 3 )
      CALL MULT ( CIBRI, TMP2, CEFBRI, 3, 3, 3 )
      CALL TRANS ( CEFBRI, CBRIEF, 3, 3 )
      CALL MULT ( CBRIEF, R, REF, 3, 3, 1 )
      CALL COOR ( REF(1), REF(2), REF(3), XLAT, XLONG, ALT )
C
      CALL AGRAV ( REF )
C
      CALL ATMOS ( ALT, RHO, PS, VSOUND, PRHOH, PPSH )
C
      CALL TMATY ( -XLONG, TMP1 )
      CALL TMATP ( XLAT, TMP2 )
      CALL MULT ( TMP2, TMP1, TMP3, 3, 3, 3 )
```

```
      CALL MULT ( CUENED, TMP3, CEFLL, 3, 3, 3 )
      CALL MULT ( CEFLL, CBRIEF, TMP1, 3, 3, 3 )
      CALL MULT ( TMP1, CBI, TMP2, 3, 3, 3 )
      CALL TRANS ( TMP2, CLLB, 3, 3 )
      CALL MULT ( CLLB, VW, VA, 3, 3, 1 )
      CALL SUBT ( VB, VA, VR, 3, 1 )
C
      VM = SQRT( VR(1)*VR(1) + VR(2)*VR(2) + VR(3)*VR(3) )
      IF(VR(1))6,7,6
6     ALPHA = CRAD*ATAN2( VR(3), VR(1) )
7     IF(VR(1).EQ.0.0)ALPHA = 0.0
      IF(VM)8,9,8
8     BETA = CRAD*ASIN ( VR(2)/VM )
9     IF(VM.EQ.0.0) BETA = 0.0
C
      XMACH = VM/VSOUND
      VMS = VM*VM
      QDYN = 0.5*RHO*VMS
C
      CALL NPLUME ( ALPHA, BETA, ALT, QDYN, PL )
C
C      SSME
C
      CALL NASSME
C
      THRSTL(1) = TVACL(1) - PSMEPS*PS
      THRSTL(2) = 0.0
      THRSTL(3) = 0.0
C
      CALL MULT ( CCLB1, THRSTL, TSSME1, 3, 3, 1 )
C
      THRSTL(1) = TVACL(2) - PSMEPS*PS
C
      CALL MULT ( CCLB2, THRSTL, TSSME2, 3, 3, 1 )
      CALL ADD ( TSSME1, TSSME2, TSSME, 3, 1 )
C
      THRSTL(1) = TVACL(3) - PSMEPS*PS
C
      CALL MULT ( CCLB3, THRSTL, TSSME3, 3, 3, 1 )
      CALL ADD ( TSSME, TSSME3, TSSME, 3, 1 )
C
C      SRB
C
      IF ( ISTAGE.EQ.1 ) THEN
C
      CALL NASSRB
C
      THRSTS(1) = TVACS(1) - PSRBPS*PS
      THRSTS(2) = 0.0
      THRSTS(3) = 0.0
C
      CALL MULT ( CCLBA, THRSTS, TSRBA, 3, 3, 1 )
C
      THRSTS(1) = TVACS(2) - PSRBPS*PS
C
```

```
      CALL MULT ( CCLBB, THRSTS, TSRBB, 3, 3, 1 )
      CALL ADD ( TSRBA, TSRBB, TSRB, 3, 1 )
C
      ELSE
C
      CALL ZEROM ( TSRB, 3, 1 )
      CALL ZEROM ( TSRBA, 3, 1 )
      CALL ZEROM ( TSRBB, 3, 1 )
      OMASS(1) = 0.0
      OMASS(2) = 0.0
C
      END IF
C
      CALL ADD ( TSSME, TSRB, THRUST, 3, 1 )
C
      OMASST = OMASL(1)+OMASL(2)+OMASL(3)+OMASS(1)+OMASS(2)
C
      CALL NMASS
C
      AVSO2M = AREA*VMS/(2.0*XMASS)
      AVSDO2 = AREA*VMS*DIA/2.0
      RAV = RHO*AREA*VM
      RAVD = RAV*DIA
      RHOA = RHO*AREA
      RAD = RHOA*DIA
      QA = QDYN*AREA
      QAOM = QA/XMASS
      QAD = QA*DIA
      OOMASS = 1.0/XMASS
      OODIA = 1.0/DIA
      RAOM = RHO*AREA/XMASS
      RAVOM = RAOM*VM
      QOM = QDYN/XMASS
      IF(VM)11,12,11
11    DO2V = DIA/(2.0*VM)
12    IF ( VM.EQ.0.0 ) DO2V = 0.0
C
      CALL NAERO ( XMACH, ALPHA, BETA, DO2V )
C
      CALL TRANS ( CBI, CIB, 3, 3 )
      CALL MULT ( CIB, CEFBRI, TMP1, 3, 3, 3 )
      CALL MULT ( TMP1, GRAV, OUT, 3, 3, 1 )
C
      CALL CROSS ( OMEGA, VB, OCV )
      CALL MULT ( CFQ, OMEGA, VEC, 3, 3, 1 )
      CALL SMLT ( DO2V, VEC, VEC, 3, 1 )
C
C   VB-DOT EQUATION
C
      DO 200 I = 1, 3
      CF(I) = CF0(I)
     1      + ALPHA*CFALP(I) + BETA*CFBET(I)
     2      + VEC(I)
      AERO(I) = QAOM*CF(I)
      ACC(I) = AERO(I) + (FPLUME(I) + THRUST(I))/XMASS
```

```
      VBD(I) = ACC(I) + OUT(I)
     1       - OCV(I)
200   CONTINUE
C
      CALL ZEROM ( TMP3, 3, 3 )
      TMP3(1,1) = A(4)**2
      TMP3(2,2) = A(5)**2
      TMP3(3,3) = A(6)**2
C
      CALL MULT ( CLLB, TMP3, TMP2, 3, 3, 3 )
      CALL TRANS ( CLLB, TMP3, 3, 3 )
      CALL MULT ( TMP2, TMP3, TMP1, 3, 3, 3 )
C
      CALL QMTRX ( OMEGA, QDMTRX )
C
C   Q-DOT EQUATION
C
       CALL MULT ( QDMTRX, Q, QD, 4, 4, 1 )
C
C     CALL SUBT ( RA, RCG, DRA, 3, 1 )
C     CALL CROSS ( DRA, CF, DRACCF )
C     CALL SMLT ( OODIA, DRACCF, DRACCF, 3, 1 )
C     CALL MULT ( XIMTRX, OMEGA, VEC, 3, 3, 1 )
C     CALL CROSS ( OMEGA, VEC, OCIO )
C
C     CALL CROSS ( DRA, FPLUME, DRACFP )
C
C     SSME
C
C     CALL SUBT ( RT1, RCG, DRT, 3, 1 )
C     CALL CROSS ( DRT, TSSME1, TTSME1 )
C
C     CALL SUBT ( RT2, RCG, DRT, 3, 1 )
C     CALL CROSS ( DRT, TSSME2, TTSME2 )
C     CALL ADD ( TTSME1, TTSME2, TTSME, 3, 1 )
C
C     CALL SUBT ( RT3, RCG, DRT, 3, 1 )
C     CALL CROSS ( DRT, TSSME3, TTSME3 )
C     CALL ADD ( TTSME, TTSME3, TTSME, 3, 1 )
C
C      SRB
C
C     IF ( ISTAGE.EQ.1 ) THEN
C
C     CALL SUBT ( RTA, RCG, DRT, 3, 1 )
C     CALL CROSS ( DRT, TSRBA, TTSRBA )
C
C     CALL SUBT ( RTB, RCG, DRT, 3, 1 )
C     CALL CROSS ( DRT, TSRBB, TTSRBB )
C     CALL ADD ( TTSRBA, TTSRBB, TTSRB, 3, 1 )
C
C     ELSE
C
C     CALL ZEROM ( TTSRB, 3, 1 )
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C       END IF
C
C       CALL ADD ( TTSME, TTSRB, TTHRST, 3, 1 )
C
C       CALL MULT ( CMQ, OMEGA, VEC, 3, 3, 1 )
C       CALL SMLT ( DO2V, VEC, VEC, 3, 1 )
C
C     OMEGA-DOT EQUATION
C
C       DO 400 I = 1, 3
C       CM(I) = CM0(I) + ALPHA*CMALP(I) + BETA*CMBET(I)
C      1     + VEC(I)
C       TPLUM(I) = TPLUME(I) + DRACFP(I)
C       AVEC(I) = QAD*CM(I) + QA*DRACCF(I) + TPLUM(I)
C       VEC(I) = AVEC(I) + TTHRST(I)
C       VEC(I) = VEC(I) - OCIO(I)
400     CONTINUE
C       CALL MULT ( XIMATI, VEC, OMEGAD, 3, 3, 1 )
C
        DO 500 I = 1, 3
        XDOT(I) = RD(I) + RDI(I)
        XDOT(I+3) = VBD(I)
        S(I+3) = TMP1(I,I)
500     CONTINUE
C
        IF ( TIME - TRINT ) 610, 590, 590
C
590     CONTINUE
C
        DO 600 I = 1, 4
        XDOT(I+6) = QD(I)
600     CONTINUE
C
610     CONTINUE
C
        RETURN
C
997     FORMAT(10X, 3E15.8)
998     FORMAT(7X, 4E15.8 )
999     FORMAT(3X, 5E15.8)
C
        END
```

```
      SUBROUTINE PDMTRX
C
      DIMENSION DUMB(71,71)
      COMMON DER(2700), VAR(2700), TMP(5400), NDER
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / CONST / A(71), S(71), R(12)
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / DRVITV / XDOT(71), PD(71,71)
C
      NTS = NS + NPAR
C
      CALL LINFS
      CALL LINFA
      CALL LINFB
C
      DO 20 I = 1, NTS
      DO 20 J = 1, NTS
      TEMP = 0.0
      DO 10 K = 1, NTS
      IF( F(I,K).EQ.0.0 ) GO TO 10
      TEMP = TEMP + F(I,K)*P(K,J)
10    CONTINUE
      PD(I,J) = TEMP
      DUMB(J,I) = TEMP
20    CONTINUE
C
      DO 30 I = 1, NTS
      DO 30 J = 1, NTS
      PD(I,J) = PD(I,J) + DUMB(I,J)
30    CONTINUE
C
      DO 40 I = 1, NTS
      PD( I, I ) = PD( I, I ) + S(I)
40    CONTINUE
C
C
C
      RETURN
902   FORMAT( 5X, 5E15.8 )
      END
      SUBROUTINE LINFS
C
      DIMENSION PCBIQ(3,4),PCIBQ(3,4),RIUNIT(3),QDMTRX(4,4)
     1          ,VEC1(3),VEC2(3),VEC3(3),TMP1(3,3),TMP2(3,3),TMP3(3,3)
     2          ,TMP2Q(3,4),TMP3Q(3,4)
C    3          ,DRA(3),DRT(3),PQO(4,3)
C    4          ,VEC4(3),VEC5(3),VEC6(3),VEC7(3)
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / TIMDAT / TMAX, HSTEP, TSAMP, TSTART, TSTOP, TRINT
      COMMON / STATES / R(3), VB(3), Q(4), OMEGA(3)
      COMMON / VARIAB / VR(3), VW(3), VWGRAD(3)
      COMMON / ENDOAT / RHO, PS, VSOUND, XMACH, PRHOH, PPSH
      COMMON / LAYOUT / RS(3),RA(3),RT1(3),RT2(3),RT3(3),RTA(3),RTB(3)
      COMMON / GEOMET / AREA, DIA, PSMEPS, PSRBPS
```

```
      COMMON / PROPER / XMASS, OMASS, XIMTRX(3,3), XIMATI(3,3), RCG(3)
      COMMON / GIMBAL / CCLB1(3,3),CCLB2(3,3),CCLB3(3,3),CCLBA(3,3)
     1                 ,CCLBB(3,3),PLN,RATEC(3),THTC(3)
      COMMON / AERO / CF0(3),CFALP(3),CFBET(3),CFQ(3,3),CM0(3),CMALP(3)
     1               ,CMBET(3),CMQ(3,3),CF(3),CM(3)
      COMMON /PLUME/ FPLUME(3),FPALP(3),FPBET(3),TPLUME(3),TPALP(3)
     1              ,TPBET(3),FPALT(3),TPALT(3),FPDPL(3),TPDPL(3)
      COMMON / TMAT / CBI(3,3), CIB(3,3), CLLB(3,3), CEFBRI(3,3)
      COMMON / ASTRON/CUENED(3,3),RNP(3,3),CIBRI(3,3),CIB0(3,3),TGNTT0
      COMMON / COMBO / AVSO2M,AVSDO2,RAV,RAVD,RAVOM,QA,QAD,QAOM,OOMASS
     1                ,RAOM,QOM,QDYN,RHOA,RAD,OODIA
      COMMON / PARTLA / PVMVB(3),PAVB(3),PBVB(3),PVMH,PAH,PBH
     1                 ,PVMVW(3),PAVW(3),PBVW(3)
      COMMON / PARTLB / PVRQ(3,4), PVMQ(4), PAQ(4), PBQ(4)
      COMMON / APARTL / PAMBRI(3,3),PAMBVB(3,3),ACC(3),THRUST(3)
      COMMON / VEHPOS / REF(3)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / EARTH / GRAV(3), AGMX(3,3)
C
      TIME = VAR(1)
C
      VMAG = SQRT ( VR(1)*VR(1) + VR(2)*VR(2) + VR(3)*VR(3) )
      RHOVM = RHO*VMAG
C
C*********POSITION
C
C       RID WRT VB
C
      CALL IMBED ( F, 71, 71, CBI, 3, 3, 1, 4 )
C
C       RID WRT Q
C
      IF ( TIME - TRINT ) 220, 210, 210
C
210   CONTINUE
C
      CALL AXCBIQ ( Q, VB, PCBIQ )
      CALL IMBED ( F, 71, 71, PCBIQ, 3, 4, 1, 7 )
C
220   CONTINUE
C
C*********VELOCITY
C
C       VBDOT WRT RI
C
C             GRAVITY
C
      CALL TRANS ( CBI, CIB, 3, 3 )
      CALL MULT ( CIB, CEFBRI, TMP2, 3, 3, 3 )
      CALL MULT ( TMP2, AGMX, TMP1, 3, 3, 3 )
      CALL TRANS ( CEFBRI, TMP3, 3, 3 )
      CALL MULT ( TMP1, TMP3, TMP2, 3, 3, 3 )
      CALL SWITCH ( TMP2, TMP1, 3, 3 )
C
C             ALTITUDE
```

```
C
      CALL AUXVAB
C
      CALL SMLT ( PRHOH, CF, VEC1, 3, 1 )
      CALL SMLT ( AVSO2M, VEC1, VEC1, 3, 1 )
      CALL SMLT ( PVMH, CF, VEC2, 3, 1 )
      CALL SMLT ( RAVOM, VEC2, VEC2, 3, 1 )
      CALL ADD ( VEC1, VEC2, VEC1, 3, 1 )
      CALL SMLT ( PAH, CFALP, VEC2, 3, 1 )
      CALL SMLT ( QAOM, VEC2, VEC2, 3, 1 )
      CALL ADD ( VEC1, VEC2, VEC1, 3, 1 )
      CALL SMLT ( PBH, CFBET, VEC2, 3, 1 )
      CALL SMLT ( QAOM, VEC2, VEC2, 3, 1 )
      CALL ADD ( VEC1, VEC2, VEC1, 3, 1 )
      CALL SMLT ( -1.0, VEC1, VEC1, 3, 1 )
C
C      FOR ALL FIVE DEVICES
C
      DO 10 I = 1, 3
      VEC3(I) = CCLB1(I,1)*PSMEPS
10    CONTINUE
      DO 20 I = 1, 3
      VEC2(I) = CCLB2(I,1)*PSMEPS
20    CONTINUE
      CALL ADD ( VEC2, VEC3, VEC2, 3, 1 )
      DO 30 I = 1, 3
      VEC3(I) = CCLB3(I,1)*PSMEPS
30    CONTINUE
      CALL ADD ( VEC2, VEC3, VEC2, 3, 1 )
      DO 40 I = 1, 3
      VEC3(I) = CCLBA(I,1)*PSRBPS
40    CONTINUE
      CALL ADD ( VEC2, VEC3, VEC2, 3, 1 )
      DO 50 I = 1, 3
      VEC3(I) = CCLBB(I,1)*PSRBPS
50    CONTINUE
      CALL ADD ( VEC2, VEC3, VEC2, 3, 1 )
      CALL SMLT ( PPSH, VEC2, VEC2, 3, 1 )
C
      CALL SMLT ( -1.0, FPALT, FPALT, 3, 1 )
      CALL ADD ( VEC2, FPALT, VEC2, 3, 1 )
C
C     CALL SMLT ( PAH, FPALP, VEC3, 3, 1 )
C     CALL ADD ( VEC2, VEC3, VEC2, 3, 1 )
C     CALL SMLT ( PBH, FPBET, VEC3, 3, 1 )
C     CALL ADD ( VEC2, VEC3, VEC2, 3, 1 )
C     CALL SMLT ( -1.0, VEC2, VEC2, 3, 1 )
C
      CALL SMLT ( OOMASS, VEC2, VEC2, 3, 1 )
      CALL ADD ( VEC1, VEC2, VEC1, 3, 1 )
C
      RMAGS = R(1)*R(1) + R(2)*R(2) + R(3)*R(3)
      RMAG = SQRT(RMAGS)
C
      RIUNIT(1) = -R(1)/RMAG
```

```
      RIUNIT(2) = -R(2)/RMAG
      RIUNIT(3) = -R(3)/RMAG
C
      CALL OUTER ( VEC1, RIUNIT, TMP2, 3, 3 )
      CALL SWITCH ( TMP2, PAMBRI, 3, 3 )
C
      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
      CALL IMBED ( F, 71, 71, TMP1, 3, 3, 4, 1 )
C
C      VBDOT WRT V
C
      CALL OUTER ( CF, PVMVB, TMP1, 3, 3 )
      CALL SMLT ( RAOM, TMP1, TMP1, 3, 3 )
      CALL SMLT ( VMAG, TMP1, TMP1, 3, 3 )
      CALL OUTER ( CFALP, PAVB, TMP2, 3, 3 )
      CALL SMLT ( QAOM, TMP2, TMP2, 3, 3 )
      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
      CALL OUTER ( CFBET, PBVB, TMP2, 3, 3 )
      CALL SMLT ( QAOM, TMP2, TMP2, 3, 3 )
      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
C
      CALL OUTER ( FPALP, PAVB, TMP2, 3, 3 )
      CALL SMLT ( OOMASS, TMP2, TMP2, 3, 3 )
      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
      CALL OUTER ( FPBET, PBVB, TMP2, 3, 3 )
      CALL SMLT ( OOMASS, TMP2, TMP2, 3, 3 )
      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
C
      CALL SWITCH ( TMP1, PAMBVB, 3, 3 )
      CALL SKEW ( OMEGA, TMP2 )
      CALL SUBT ( TMP1, TMP2, TMP1, 3, 3 )
      CALL IMBED ( F, 71, 71, TMP1, 3, 3, 4, 4 )
C
C   VBDOT WRT Q
C
      IF ( TIME - TRINT ) 240, 230, 230
C
230   CONTINUE
C
      CALL MULT ( CEFBRI, GRAV, VEC1, 3, 3, 1 )
      CALL AXCIBQ ( Q, VEC1, PCIBQ )
C
      CALL MULT ( CLLB, VW, VEC1, 3, 3, 1 )
      CALL AXCBIQ ( Q, VEC1, PVRQ )
      CALL SMLT ( -1.0, PVRQ, PVRQ, 3, 4 )
C
      CALL OUTER ( CF, PVMVB, TMP1, 3, 3 )
      CALL SMLT ( RAOM, TMP1, TMP1, 3, 3 )
      CALL SMLT ( VMAG, TMP1, TMP1, 3, 3 )
      CALL OUTER ( CFALP, PAVB, TMP2, 3, 3 )
      CALL SMLT ( QAOM, TMP2, TMP2, 3, 3 )
      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
      CALL OUTER ( CFBET, PBVB, TMP2, 3, 3 )
      CALL SMLT ( QAOM, TMP2, TMP2, 3, 3 )
      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
```

```
      CALL MULT ( TMP1, PVRQ, TMP2Q, 3, 3, 4 )
C
      CALL OUTER ( FPALP, PAVB, TMP2, 3, 3 )
      CALL OUTER ( FPBET, PBVB, TMP3, 3, 3 )
      CALL ADD ( TMP2, TMP3, TMP2, 3, 3 )
      CALL MULT ( TMP2, PVRQ, TMP3Q, 3, 3, 4 )
      CALL SMLT ( OOMASS, TMP3Q, TMP3Q, 3, 4 )
C
      CALL ADD ( TMP2Q, TMP3Q, TMP2Q, 3, 4 )
      CALL ADD ( TMP2Q, PCIBQ, TMP2Q, 3, 4 )
      CALL IMBED ( F, 71, 71, TMP2Q, 3, 4, 4, 7 )
C
240   CONTINUE
C
C      VBDOT WRT OMEGA
C
C     CALL SMLT ( QAOM, CFQ, TMP1, 3, 3 )
C     CALL SKEW ( VB, TMP2 )
C     CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
C     CALL IMBED ( F, 71, 71, TMP1, 3, 3, 4, 11 )
C
C********ATTITUDE
C
C      QD WRT Q
C
      IF ( TIME - TRINT ) 260, 250, 250
C
250   CONTINUE
C
      CALL QMTRX( OMEGA, QDMTRX )
      CALL IMBED ( F, 71, 71, QDMTRX, 4, 4, 7, 7 )
C
260   CONTINUE
C
C      QD WRT OMEGA
C
C     CALL PQOMEG ( Q, PQO )
C     CALL IMBED ( F, 71, 71, PQO, 4, 3, 7, 11 )
C
C********ANGULAR RATE
C
C     CALL SUBT ( RA, RCG, DRA, 3, 1 )
C     CALL SMLT ( OODIA, DRA, DRA, 3, 1 )
C
C      OMEGAD WRT RI
C
C     CALL SMLT ( RAVD, CM, VEC1, 3, 1 )
C     CALL CROSS ( DRA, CF, VEC2 )
C     CALL SMLT ( RAV, VEC2, VEC2, 3, 1 )
C     CALL ADD ( VEC1, VEC2, VEC1, 3, 1 )
C     CALL SMLT ( AVSDO2, CM, VEC4, 3, 1 )
C     CALL SMLT ( PRHOH, VEC4, VEC4, 3, 1 )
C     CALL ADD ( VEC1, VEC4, VEC4, 3, 1 )
C     CALL SMLT ( PVMH, VEC4, VEC4, 3, 1 )
C     CALL SMLT ( QAD, CMALP, VEC2, 3, 1 )
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C     CALL CROSS ( DRA, CFALP, VEC3 )
C     CALL SMLT ( QA, VEC3, VEC3, 3, 1 )
C     CALL ADD ( VEC2, VEC3, VEC2, 3, 1 )
C     CALL SMLT ( PAH, VEC2, VEC5, 3, 1 )
C     CALL ADD ( VEC4, VEC5, VEC4, 3, 1 )
C     CALL SMLT ( QAD, CMBET, VEC3, 3, 1 )
C     CALL CROSS ( DRA, CFBET, VEC5 )
C     CALL SMLT ( QA, VEC5, VEC5, 3, 1 )
C     CALL ADD (VEC3, VEC5, VEC3, 3, 1 )
C     CALL SMLT ( PBH, VEC3, VEC5, 3, 1 )
C     CALL ADD ( VEC4, VEC5, VEC4, 3, 1 )
C
C      FOR ALL FIVE DEVICES
C
C     DO 110 I = 1, 3
C     VEC5(I) = CCLB1(I,1)*PSMEPS
110   CONTINUE
C     CALL SUBT ( RT1, RCG, DRT, 3, 1 )
C     CALL CROSS ( DRT, VEC5, VEC6 )
C     DO 120 I = 1, 3
C     VEC5(I) = CCLB2(I,1)*PSMEPS
120   CONTINUE
C     CALL SUBT ( RT2, RCG, DRT, 3, 1 )
C     CALL CROSS ( DRT, VEC5, VEC7 )
C     CALL ADD ( VEC6, VEC7, VEC6, 3, 1 )
C     DO 130 I = 1, 3
C     VEC5(I) = CCLB3(I,1)*PSMEPS
130   CONTINUE
C     CALL SUBT ( RT3, RCG, DRT, 3, 1 )
C     CALL CROSS ( DRT, VEC5, VEC7 )
C     CALL ADD ( VEC6, VEC7, VEC6, 3, 1 )
C     DO 140 I = 1, 3
C     VEC5(I) = CCLBA(I,1)*PSRBPS
140   CONTINUE
C     CALL SUBT ( RTA, RCG, DRT, 3, 1 )
C     CALL CROSS ( DRT, VEC5, VEC7 )
C     CALL ADD ( VEC6, VEC7, VEC6, 3, 1 )
C     DO 150 I = 1, 3
C     VEC5(I) = CCLBB(I,1)*PSRBPS
150   CONTINUE
C     CALL SUBT ( RTB, RCG, DRT, 3, 1 )
C     CALL CROSS ( DRT, VEC5, VEC7 )
C     CALL ADD ( VEC6, VEC7, VEC6, 3, 1 )
C
C     CALL SMLT ( PPSH, VEC6, VEC5, 3, 1 )
C     CALL ADD ( VEC5, TPALT, VEC5, 3, 1 )
C     CALL SMLT ( PAH, TPALP, VEC6, 3, 1 )
C     CALL ADD ( VEC5, VEC6, VEC5, 3, 1 )
C     CALL SMLT ( PBH, TPBET, VEC6, 3, 1 )
C     CALL ADD ( VEC5, VEC6, VEC5, 3, 1 )
C     CALL ADD ( VEC4, VEC5, VEC4, 3, 1 )
C     CALL OUTER ( VEC4, RIUNIT, TMP2, 3, 3 )
C     CALL MULT ( XIMATI, TMP2, TMP1, 3, 3, 3 )
C     CALL IMBED ( F, 71, 71, TMP1, 3, 3, 11, 1 )
C
```

```
C       OMEGAD WRT VB
C
C      CALL OUTER ( VEC1, PVMVB, TMP1, 3, 3 )
C      CALL OUTER ( VEC2, PAVB, TMP2, 3, 3 )
C      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
C      CALL OUTER ( VEC3, PBVB, TMP2, 3, 3 )
C      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
C      CALL OUTER ( TPALP, PAVB, TMP2, 3, 3 )
C      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
C      CALL OUTER ( TPBET, PBVB, TMP2, 3, 3 )
C      CALL ADD ( TMP1, TMP2, TMP2, 3, 3 )
C      CALL MULT ( XIMATI, TMP2, TMP1, 3, 3, 3 )
C      CALL IMBED ( F, 71, 71, TMP1, 3, 3, 11, 4 )
C
C       OMEGAD WRT Q
C
C        CALL AXVABQ
C
C      CALL OUTER ( VEC1, PVMQ, TMP2Q, 3, 4 )
C      CALL OUTER ( VEC2, PAQ, TMP3Q, 3, 4 )
C      CALL ADD ( TMP2Q, TMP3Q, TMP2Q, 3, 4 )
C      CALL OUTER ( VEC3, PBQ, TMP3Q, 3, 4 )
C      CALL ADD ( TMP2Q, TMP3Q, TMP2Q, 3, 4 )
C      CALL OUTER ( TPALP, PAQ, TMP3Q, 3, 4 )
C      CALL ADD ( TMP2Q, TMP3Q, TMP2Q, 3, 4 )
C      CALL OUTER ( TPBET, PBQ, TMP3Q, 3, 4 )
C      CALL ADD ( TMP2Q, TMP3Q, TMP2Q, 3, 4 )
C      CALL MULT ( XIMATI, TMP2Q, TMP3Q, 3, 3, 4 )
C      CALL IMBED ( F, 71, 71, TMP3Q, 3, 4, 11, 7 )
C
C       OMEGAD WRT OMEGA
C
C      CALL SMLT ( QAD, CMQ, TMP1, 3, 3 )
C      CALL SKEW ( DRA, TMP2 )
C      CALL MULT ( TMP2, CFQ, TMP3, 3, 3, 3 )
C      CALL SMLT ( QA, TMP3, TMP3, 3, 3 )
C      CALL ADD ( TMP1, TMP3, TMP1, 3, 3 )
C      CALL MULT ( XIMTRX, OMEGA, VEC1, 3, 3, 1 )
C      CALL SKEW ( VEC1, TMP2 )
C      CALL ADD ( TMP1, TMP2, TMP1, 3, 3 )
C      CALL SKEW ( OMEGA, TMP2 )
C      CALL MULT ( TMP2, XIMTRX, TMP3, 3, 3, 3 )
C      CALL SUBT ( TMP1, TMP3, TMP2, 3, 3 )
C      CALL MULT ( XIMATI, TMP2, TMP1, 3, 3, 3 )
C      CALL IMBED ( F, 71, 71, TMP1, 3, 3, 11, 11 )
C
       RETURN
C
901    FORMAT( 11E7.1 )
902    FORMAT( 5X, 3E15.8 )
C
       END
       SUBROUTINE LINFA
C
       DIMENSION UNIT(3,3)
```

```
     1          ,VEC(3),VEC2(3),TMP(3,3)
C    2          ,DRT(3)
C    3          ,TMP2(3,3),TMP3(3,3)
C
      COMMON / MEPRTL / PFTMR(3),PFTPC(3),PFTGO(3),PFTGH(3)
      COMMON / RBPRTL / PFT(2),PFA(2),PFCS(2)
      COMMON / AERO / CF0(3),CFALP(3),CFBET(3),CFQ(3,3),CM0(3),CMALP(3)
     1               ,CMBET(3),CMQ(3,3),CF(3),CM(3)
      COMMON / COMBO / AVSO2M,AVSDO2,RAV,RAVD,RAVOM,QA,QAD,QAOM,OOMASS
     1                ,RAOM,QOM,QDYN,RHOA,RAD,OODIA
      COMMON / PLUME / FPLUME(3),FPALP(3),FPBET(3),TPLUME(3),TPALP(3)
     1                ,TPBET(3),FPALT(3),TPALT(3),FPDPL(3),TPDPL(3)
      COMMON / PARTLA / PVMVB(3),PAVB(3),PBVB(3),PVMH,PAH,PBH
     1                 ,PVMVW(3),PAVW(3),PBVW(3)
      COMMON / LAYOUT / RS(3),RA(3),RT1(3),RT2(3),RT3(3),RTA(3),RTB(3)
      COMMON / GIMBAL / CCLB1(3,3),CCLB2(3,3),CCLB3(3,3),CCLBA(3,3)
     1                 ,CCLBB(3,3),PLN,RATEC(3),THTC(3)
      COMMON / PROPER / XMASS, OMASS, XIMTRX(3,3), XIMATI(3,3), RCG(3)
      COMMON / APARTL / PAMBRI(3,3),PAMBVB(3,3),ACC(3),THRUST(3)
      COMMON / PARARO / PVBDCF(3,3),PWDCF(3,3),PWDCM(3,3)
      COMMON / PARPLM / PVBDFP(3,3),PWDFP(3,3),PWDTP(3,3)
      COMMON / PARRCG / PWDRCG(3,3)
      COMMON / PARVWX / PVBDVW(3,3),PWDVW(3,3)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / IPARAM / ISSME,ISRB,IAERO,IPLUME,IWIND,JACB,JRDR
C
      DATA UNIT / 1.0,3*0.0,1.0,3*0.0,1.0 /
      RHOOM = RHO*OOMASS
C
C      VBDOT WRT SSME1 PARAMETERS
C
      DO 10  I = 1, 3
      VEC(I) = CCLB1(I,1)/XMASS
      VEC2(I) = CCLB1(I,1)
C
        F( (3+I), 14 ) = THRUST(I)/(32.174*XMASS**2)
        F( (3+I), 15 ) = VEC(I)*PFTMR(1)
        F( (3+I), 16 ) = VEC(I)*PFTPC(1)
        F( (3+I), 17 ) = VEC(I)*PFTGO(1)
        F( (3+I), 18 ) = VEC(I)*PFTGH(1)
C
10    CONTINUE
C
C      OMEGAD WRT SSME1 PARAMETERS
C
C     CALL SUBT ( RT1, RCG, DRT, 3, 1)
C     CALL MULT ( XIMATI, DRT, VEC, 3, 3, 1 )
C     CALL CROSS ( VEC, VEC2, DRT )
C
C     DO 20 I = 1, 3
C
C     F( (10+I), 15 ) = DRT(I)*PFTMR(1)
C     F( (10+I), 16 ) = DRT(I)*PFTPC(1)
C     F( (10+I), 17 ) = DRT(I)*PFTGO(1)
C     F( (10+I), 18 ) = DRT(I)*PFTGH(1)
```

```
C
20    CONTINUE
C
C      VBDOT WRT SSME2 PARAMETERS
C
      DO 30  I = 1, 3
      VEC(I) = CCLB2(I,1)/XMASS
      VEC2(I) = CCLB2(I,1)
C
        F( (3+I), 23 ) = THRUST(I)/(32.174*XMASS**2)
        F( (3+I), 24 ) = VEC(I)*PFTMR(2)
        F( (3+I), 25 ) = VEC(I)*PFTPC(2)
        F( (3+I), 26 ) = VEC(I)*PFTGO(2)
        F( (3+I), 27 ) = VEC(I)*PFTGH(2)
C
30    CONTINUE
C
C      OMEGAD WRT SSME2 PARAMETERS
C
C     CALL SUBT ( RT2, RCG, DRT, 3, 1 )
C     CALL MULT ( XIMATI, DRT, VEC, 3, 3, 1 )
C     CALL CROSS ( VEC, VEC2, DRT )
C
C     DO 40  I = 1, 3
C
C     F( (10+I), 24 ) = DRT(I)*PFTMR(2)
C     F( (10+I), 25 ) = DRT(I)*PFTPC(2)
C     F( (10+I), 26 ) = DRT(I)*PFTGO(2)
C     F( (10+I), 27 ) = DRT(I)*PFTGH(2)
C
40    CONTINUE
C
C      VBDOT WRT SSME3 PARAMETERS
C
      DO 50  I = 1, 3
      VEC(I) = CCLB3(I,1)/XMASS
      VEC2(I) = CCLB3(I,1)
C
        F( (3+I), 32 ) = THRUST(I)/(32.174*XMASS**2)
        F( (3+I), 33 ) = VEC(I)*PFTMR(3)
        F( (3+I), 34 ) = VEC(I)*PFTPC(3)
        F( (3+I), 35 ) = VEC(I)*PFTGO(3)
        F( (3+I), 36 ) = VEC(I)*PFTGH(3)
C
50    CONTINUE
C
C      OMEGAD WRT SSME3 PARAMETERS
C
C     CALL SUBT ( RT3, RCG, DRT, 3, 1 )
C     CALL MULT ( XIMATI, DRT, VEC, 3, 3, 1 )
C     CALL CROSS ( VEC, VEC2, DRT )
C
C     DO 60  I = 1, 3
C
C     F( (10+I), 33 ) = DRT(I)*PFTMR(3)
```

```
C      F( (10+I), 34 ) = DRT(I)*PFTPC(3)
C      F( (10+I), 35 ) = DRT(I)*PFTGO(3)
C      F( (10+I), 36 ) = DRT(I)*PFTGH(3)
C
60     CONTINUE
C
       KSTATE = 61
       IF(ISRB) 200, 200, 70
70     CONTINUE
C
C       VBDOT WRT SRBA PARAMETERS
C
       DO 80  I = 1, 3
       VEC(I) = CCLBA(I,1)/XMASS
       VEC2(I) = CCLBA(I,1)
C
         F( (3+I), KSTATE+1 ) = THRUST(I)/(32.174*XMASS**2)
         F( (3+I), KSTATE+2 ) = VEC(I)*PFT(1)
         F( (3+I), KSTATE+3 ) = VEC(I)*PFA(1)
         F( (3+I), KSTATE+4 ) = VEC(I)*PFCS(1)
C
80     CONTINUE
C
C       OMEGAD WRT SRBA PARAMETERS
C
C      CALL SUBT ( RTA, RCG, DRT, 3, 1 )
C      CALL MULT ( XIMATI, DRT, VEC, 3, 3, 1 )
C      CALL CROSS ( VEC, VEC2, DRT )
C
C      DO 90  I = 1, 3
C
C      F( (10+I), KSTATE+2 ) = DRT(I)*PFT(1)
C      F( (10+I), KSTATE+3 ) = DRT(I)*PFA(1)
C      F( (10+I), KSTATE+4 ) = DRT(I)*PFCS(1)
C
90     CONTINUE
C
C       VBDOT WRT SRBB PARAMETERS
C
       DO 100 I = 1, 3
       VEC(I) = CCLBB(I,1)/XMASS
       VEC2(I) = CCLBB(I,1)
C
         F( (3+I), KSTATE+6 ) = THRUST(I)/(32.174*XMASS**2)
         F( (3+I), KSTATE+7 ) = VEC(I)*PFT(2)
         F( (3+I), KSTATE+8 ) = VEC(I)*PFA(2)
         F( (3+I), KSTATE+9 ) = VEC(I)*PFCS(2)
C
100    CONTINUE
C
C       OMEGAD WRT SRBB PARAMETERS
C
C      CALL SUBT ( RTB, RCG, DRT, 3, 1 )
C      CALL MULT ( XIMATI, DRT, VEC, 3, 3, 1 )
C      CALL CROSS ( VEC, VEC2, DRT )
```

```
C
C     DO 110 I = 1, 3
C
C     F( (10+I), KSTATE+7 ) = DRT(I)*PFT(2)
C     F( (10+I), KSTATE+8 ) = DRT(I)*PFA(2)
C     F( (10+I), KSTATE+9 ) = DRT(I)*PFCS(2)
C
110   CONTINUE
C
200   CONTINUE
C
      KSTATE = 40
C
      IF( IAERO ) 300, 300, 210
210   CONTINUE
C
      CALL SMLT ( QAOM, UNIT, PVBDCF, 3, 3 )
      CALL IMBED ( F, 71, 71, PVBDCF, 3, 3, 4, (KSTATE+1) )
C
C     CALL SUBT ( RA, RCG, VEC, 3, 1 )
C     CALL SMLT ( OODIA, VEC, VEC, 3, 1 )
C     CALL SKEW ( VEC, TMP )
C     CALL SMLT ( QA, TMP, TMP, 3, 3 )
C     CALL MULT ( XIMATI, TMP, PWDCF, 3, 3, 3 )
C     CALL IMBED ( F, 71, 71, PWDCF, 3, 3, 11, (KSTATE+1) )
C
C     CALL SMLT ( QAD, XIMATI, PWDCM, 3, 3 )
C     CALL IMBED ( F, 71, 71, PWDCM, 3, 3, 11, (KSTATE+4) )
C
      KSTATE = KSTATE + 3
C
300   CONTINUE
C
      IF( IPLUME ) 400, 400, 310
310   CONTINUE
C
      CALL SMLT ( OOMASS, UNIT, PVBDFP, 3, 3 )
      CALL IMBED ( F, 71, 71, PVBDFP, 3, 3, 4, (KSTATE+1) )
C
C     CALL SUBT ( RA, RCG, VEC, 3, 1 )
C     CALL SKEW ( VEC, TMP )
C     CALL MULT ( XIMATI, TMP, PWDFP, 3, 3, 3 )
C     CALL IMBED ( F, 71, 71, PWDFP, 3, 3, 11, (KSTATE+1) )
C
C     CALL SWITCH ( XIMATI, PWDTP, 3, 3 )
C     CALL IMBED ( F, 71, 71, PWDTP, 3, 3, 11, (KSTATE+4) )
C
      KSTATE = KSTATE + 3
C
400   CONTINUE
C
      IF( IWIND ) 600, 600, 510
510   CONTINUE
C
      CALL OUTER ( CF, PVMVW, TMP, 3, 3 )
```

```
      CALL SMLT ( RAOM, TMP, PVBDVW, 3, 3 )
      CALL SMLT ( 0.5, PVBDVW, PVBDVW, 3, 3 )
      CALL OUTER ( CFALP, PAVW, TMP, 3, 3 )
      CALL SMLT ( QAOM, TMP, TMP, 3, 3 )
      CALL ADD ( PVBDVW, TMP, PVBDVW, 3, 3 )
      CALL OUTER ( CFBET, PBVW, TMP, 3, 3 )
      CALL SMLT ( QAOM, TMP, TMP, 3, 3 )
      CALL ADD ( PVBDVW, TMP, PVBDVW, 3, 3 )
      CALL OUTER ( FPALP, PAVW, TMP, 3, 3 )
      CALL SMLT ( OOMASS, TMP, TMP, 3, 3 )
      CALL ADD ( PVBDVW, TMP, PVBDVW, 3, 3 )
      CALL OUTER ( FPBET, PBVW, TMP, 3, 3 )
      CALL SMLT ( OOMASS, TMP, TMP, 3, 3 )
      CALL ADD ( PVBDVW, TMP, PVBDVW, 3, 3 )
      CALL IMBED ( F, 71, 71, PVBDVW, 3, 3, 4, (KSTATE+1) )
C
C     CALL OUTER ( CM, PVMVW, TMP, 3, 3 )
C     CALL SMLT ( RAD, TMP, PWDVW, 3, 3 )
C     CALL SMLT ( 0.5, PWDVW, PWDVW, 3, 3 )
C     CALL OUTER ( CMALP, PAVW, TMP, 3, 3 )
C     CALL SMLT ( QAD, TMP, TMP, 3, 3 )
C     CALL ADD ( PWDVW, TMP, PWDVW, 3, 3 )
C     CALL OUTER ( CMBET, PBVW, TMP, 3, 3 )
C     CALL SMLT ( QAD, TMP, TMP, 3, 3 )
C     CALL ADD ( PWDVW, TMP, PWDVW, 3, 3 )
C     CALL SUBT ( RA, RCG, VEC, 3, 1 )
C     CALL SMLT ( OODIA, VEC, VEC, 3, 1 )
C     CALL CROSS ( VEC, CF, VEC2 )
C     CALL OUTER ( PVMVW, VEC2, TMP, 3, 3 )
C     CALL SMLT ( RHOA, TMP, TMP, 3, 3 )
C     CALL ADD ( PWDVW, TMP, PWDVW, 3, 3 )
C     CALL SMLT ( 0.5, PWDVW, PWDVW, 3, 3 )
C     CALL OUTER ( CFALP, PAVW, TMP, 3, 3 )
C     CALL SKEW ( VEC, TMP2 )
C     CALL MULT ( TMP2, TMP, TMP3, 3, 3, 3 )
C     CALL SMLT ( QA, TMP3, TMP3, 3, 3 )
C     CALL ADD ( PWDVW, TMP3, PWDVW, 3, 3 )
C     CALL OUTER ( CFBET, PBVW, TMP, 3, 3 )
C     CALL MULT ( TMP2, TMP, TMP3, 3, 3, 3 )
C     CALL SMLT ( QA, TMP3, TMP3, 3, 3 )
C     CALL ADD ( PWDVW, TMP3, PWDVW, 3, 3 )
C     CALL OUTER ( TPALP, PAVW, TMP, 3, 3 )
C     CALL OUTER ( TPBET, PBVW, TMP3, 3, 3 )
C     CALL ADD ( TMP, TMP3, TMP, 3, 3 )
C     CALL ADD ( PWDVW, TMP, PWDVW, 3, 3 )
C     CALL CROSS ( FPALP, PAVW, TMP, 3, 3 )
C     CALL CROSS ( FPBET, PBVW, TMP3, 3, 3 )
C     CALL ADD ( TMP, TMP3, TMP, 3, 3 )
C     CALL MULT ( TMP2, TMP, TMP3, 3, 3, 3 )
C     CALL ADD ( PWDVW, TMP3, TMP, 3, 3 )
C     CALL MULT ( XIMATI, TMP, PWDVW, 3, 3, 3 )
C
C     CALL IMBED ( F, 71, 71, PWDVW, 3, 3, 11, (KSTATE+1) )
C
      KSTATE = KSTATE + 3
```

```
C
600   CONTINUE
C
      RETURN
C
902   FORMAT( 5X, 9E8.1 )
903   FORMAT( 7X, 3E12.5 )
      END
```

```
      SUBROUTINE ATMOS( ALT, RHO, PS, SOUND, PRHOH, PPH )
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / CONST / A(71), S(71), RR(12)
      COMMON / ATMOSP / THALT(15), THRHO(15), THP(15), THSUND(15)
      COMMON / SWIND / TALT(400), TVWX(400), TVWY(400), NALT
      COMMON / SATMOS / TSRHO(400), TSP(400), TSSUND(400), IATMOS
      COMMON / PREUP / XKM(71), PKM(71,71)
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / DRVITV / XDOT(71), PD(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / IPARAM / ISSME,ISRB,IAERO,IPLUME,IWIND,JACB,JRDR
      COMMON / VARIAB / VR(3), VW(3), VWGRAD(3)
C
      DATA TAUW / 2.E+1 /
      DATA RHO0,T0,XLAPSE,P0,ZTROP/.0023769,518.7,.00357,2116.2,36089./
      DATA G,R/32.174,1715.0/
C
      IF ( IATMOS ) 1, 1, 100
C
1     CONTINUE
C
C      1962 STANDARD ATMOSPHERE FOR ALTITUDES UP TO 100000 FT
C
      IF( ALT ) 2, 4, 4
2     CONTINUE
      T = T0
      RHO = RHO0
      PS = P0
      PRHOH = 0.0
      PPH = 0.0
      GO TO 30
4     CONTINUE
      IF ( ALT - 60000.0 ) 5, 40, 40
5     CONTINUE
C
      POWER=(G/(XLAPSE*R))-1.0
      IF(ALT-ZTROP)20,10,10
10    CONTINUE
      T=T0-XLAPSE*ZTROP
      TROP=EXP(-G*(ALT-ZTROP)/(R*T))
      TMP=(T/T0)**POWER*TROP
      RHO = RHO0*TMP
      RHOT = RHO/TROP
      RRHOT = RHOT/RHO0
      RTMPT = T/T0
      PZTROP = P0*RRHOT*RTMPT
      RRHO = RHO/RHOT
      RTMP = 1.0
      PS = PZTROP*RRHO*RTMP
      PRHOH = -RHO0*TMP*G/(R*T)
      PPH = -P0*TMP*G/(R*T)
      GO TO 30
C
20    CONTINUE
```

```
      T=T0-XLAPSE*ALT
      RHO=RHO0*(T/T0)**POWER
      RRHO = RHO/RHO0
      RTMP = T/T0
      PS = P0*RRHO*RTMP
      PRHOH = -RHO0*POWER*XLAPSE*(T/T0)**(POWER-1.)
      PPH = -P0*(POWER+1.)*XLAPSE*(T/T0)**POWER
C
30    CONTINUE
C
      SOUND=49.02*SQRT(T)
C
      GO TO 50
40    CONTINUE
C
      CALL INTRP1 ( ALT, THALT, THRHO, 15, RHO, PRHOH )
      CALL INTRP1 ( ALT, THALT, THP, 15, PS, PPH )
      CALL INTRP1 ( ALT, THALT, THSUND, 15, SOUND, PSH )
C
50    CONTINUE
C
      GO TO 200
C
100   CONTINUE
C
      VWZ = 0.0
      VWGZ = 0.0
C
      IF ( ALT.LT.390000. ) THEN
      CALL INTRP1 ( ALT, TALT, TSRHO, NALT, RHO, PRHOH )
      CALL INTRP1 ( ALT, TALT, TSP, NALT, PS, PPH )
      CALL INTRP1 ( ALT, TALT, TSSUND, NALT, SOUND, PSH )
      CALL INTRP1 ( ALT, TALT, TVWX, NALT, VWX, VWGX )
      CALL INTRP1 ( ALT, TALT, TVWY, NALT, VWY, VWGY )
      ELSE
         IF ( ALT.LT.100. ) THEN
         RHO = .002377
         PS = 2116.
         SOUND = 1117.
         VWX = 0.0
         VWY = 0.0
         END IF
      END IF
C
      VW(1) = VWX
      VW(2) = VWY
      VW(3) = VWZ
      VWGRAD(1) = VWGX
      VWGRAD(2) = VWGY
      VWGRAD(3) = VWGZ
C
      IF( IWIND.GT.0 ) THEN
C
      OOTAU = 1./TAUW
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      F(47,47) = -OOTAU
      F(48,48) = -OOTAU
      F(49,49) = -OOTAU
C
      S(47) = 2.*A(47)**2/TAUW
      S(48) = 2.*A(48)**2/TAUW
      S(49) = 2.*A(49)**2/TAUW
C
      XDOT(47) = -OOTAU*X(47)
      XDOT(48) = -OOTAU*X(48)
      XDOT(49) = -OOTAU*X(49)
C
      VW(1) = VW(1) + X(47)
      VW(2) = VW(2) + X(48)
      VW(3) = VW(3) + X(49)
C
      ELSE
      END IF
C
200   CONTINUE
C
      RETURN
901   FORMAT(10X, 3E15.8)
      END
```

```
      SUBROUTINE NASSME
C
      DIMENSION XNOM(5),YRPL(3),VEC1(71),VEC2(71)
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / CONST / A(71), S(71), R(12)
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / DRVITV / XDOT(71), PD(71,71)
      COMMON / SMETBL / PR(4,3,5),YN(4,3),XN(4,2)
     1                 ,PRZ(4,3,3),PRX(4,2,2),ZN(4,3)
     2                 ,RPLTAG(3),WDO2TU(3),WDH2TU(3),TVACTU(3)
      COMMON / SMEDAT / TVACL(3),XISPL(3),WDO2H(3),WDH2H(3),WD(3)
     1                 ,CVTVAC(3),CVISPL(3),CVWDO2(3),CVWDH2(3)
     2                 ,CVWD(3),CVWDGO(3),CVWDGH(3),PCTAG(3),OMASE(3)
      COMMON / MEPRTL / PFTMR(3),PFTPC(3),PFTGO(3),PFTGH(3)
      COMMON / GIMBAL / CCLB1(3,3),CCLB2(3,3),CCLB3(3,3),CCLBA(3,3)
     1                 ,CCLBB(3,3),PLN,RATEC(3),THTC(3)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / SMEMEA / XLH(6,3)
      COMMON / LINHMT / H(35,71), NMEAS
C
      DATA XNOM,YRPL / 6.,1.,1.0114,1.55,.7,895.74,149.168,471437./
C
      DATA TMR,TPC,TWDGO2,TWDGH2 / 4*10.0 /
      DATA TY8,TPH,TTH,TPOB / 4*100. /
C
      ISSME = 9
      MSSME = 6
C
C   FOR EACH SSME
C
      DO 1000 I = 1, 3
C
      CALL ZEROM ( VEC1, 71, 1 )
      CALL ZEROM ( VEC2, 71, 1 )
C
      PCTAG(I) = RPLTAG(I)*PLN
C
      PCHAT = PCTAG(I) + X( 16+ISSME*(I-1) ) + X( 22+ISSME*(I-1) )
      PL = PCHAT/RPLTAG(I)
      XLH( 1, I ) = PCHAT
C
      WDO2U = POW( YN(1, 1), 1.0, 4 )
      WDO2P = POW( YN(1, 1), PLN, 4 )
      WDO2 = WDO2TU(I) + ( WDO2P - WDO2U )
      WDH2U = POW( YN(1, 2), 1.0, 4 )
      WDH2P = POW( YN(1, 2), PLN, 4 )
      WDH2 = WDH2TU(I) + ( WDH2P - WDH2U )
      TVACU = POW( YN(1, 3), 1.0, 4 )
      TVACP = POW( YN(1, 3), PLN, 4 )
      TVAC = TVACTU(I) + ( TVACP - TVACU )
      WDGO = POW( XN( 1, 1), PLN, 4 )
      WDGH = POW( XN( 1, 2), PLN, 4 )
C
      Y8 = POW( ZN( 1, 1), PLN, 4 )
```

```
      PH = POW( ZN( 1, 2), PLN, 4 )
      TH = POW( ZN( 1, 3), PLN, 4 )
C
      PGHPLN = XN( 2, 2)
      PGOPLN = XN( 2, 1) + XN(3, 1)*PLN*2.
C
      PWOMR = POW( PR(1, 1, 1), PLN, 4 )
      PWOMR = PWOMR*WDO2/XNOM(1)
      PWFMR = POW( PR(1, 2, 1), PLN, 4 )
      PWFMR = PWFMR*WDH2/XNOM(1)
      PFTMRI = POW( PR(1, 3, 1), PLN, 4 )
      PFTMR(I) = PFTMRI*TVACL(I)/XNOM(1)
      PWOPLN = POW( PR(1, 1, 2), PLN, 4 )
      PWFPLN = POW( PR(1, 2, 2), PLN, 4 )
      PFTPLN = POW( PR(1, 3, 2), PLN, 4 )
      PFTPC(I) = PFTPL/PCTAG(I)
      PWOGO = POW( PR(1, 1, 4), PLN, 4 )
      PWFGO = POW( PR(1, 2, 4), PLN, 4 )
      PFTGO(I) = POW( PR(1, 3, 4), PLN, 4 )
      PWOGH = POW( PR(1, 1, 5), PLN, 4 )
      PWFGH = POW( PR(1, 2, 5), PLN, 4 )
      PFTGH(I) = POW( PR(1, 3, 5), PLN, 4 )
      PGHMR = POW( PRX( 1, 2, 1), PLN, 4 )
      PGOMR = POW( PRX( 1, 1, 1), PLN, 4 )
      PWGHGO = POW( PRX( 1, 2, 2), PLN, 4 )
      PWGOGH = POW( PRX( 1, 1, 2), PLN, 4 )
C
      PWOY8 = POW( PRZ( 1, 1, 1), PLN, 4 )
      PWFY8 = POW( PRZ( 1, 2, 1), PLN, 4 )
      PFTY8 = POW( PRZ( 1, 3, 1), PLN, 4 )
      PWOPH = POW( PRZ( 1, 1, 2), PLN, 4 )
      PWFPH = POW( PRZ( 1, 2, 2), PLN, 4 )
      PFTPH = POW( PRZ( 1, 3, 2), PLN, 4 )
      PWOTH = POW( PRZ( 1, 1, 3), PLN, 4 )
      PWFTH = POW( PRZ( 1, 2, 3), PLN, 4 )
      PFTTH = POW( PRZ( 1, 3, 3), PLN, 4 )
C
      PFTWF = PFTMRI/PWFMR
      PWOPC = PWOPL/PCTAG(I)
      PWFPC = PWFPL/PCTAG(I)
      PWOWF = XNOM(1)
C
      PWDMR = PWOMR + PWFMR - PGOMR - PGHMR
      PWDPC = ( PWOPL + PWFPL - PGOPL - PGHPL )/PCTAG(I)
      PWDGO = PWOGO + PWFGO - 1.0 + PWGHGO
      PWDGH = PWOGH + PWFGH - 1.0 + PWGOGH
      PWDWF = PWOMR/PWFMR + 1.0
C
      OMASE(I) = X( 14+ISSME*(I-1) )
C
      F( (14+ISSME*(I-1)), (15+ISSME*(I-1)) ) = PWDMR
      F( (14+ISSME*(I-1)), (16+ISSME*(I-1)) ) = PWDPC
      F( (14+ISSME*(I-1)), (17+ISSME*(I-1)) ) = PWDGO
      F( (14+ISSME*(I-1)), (18+ISSME*(I-1)) ) = PWDGH
C
```

```
      F( (15+ISSME*(I-1)), (15+ISSME*(I-1)) ) = -1./TMR
      F( (16+ISSME*(I-1)), (16+ISSME*(I-1)) ) = -1./TPC
      F( (17+ISSME*(I-1)), (17+ISSME*(I-1)) ) = -1./TWDGO2
      F( (18+ISSME*(I-1)), (18+ISSME*(I-1)) ) = -1./TWDGH2
      F( (19+ISSME*(I-1)), (19+ISSME*(I-1)) ) = -1./TY8
      F( (20+ISSME*(I-1)), (20+ISSME*(I-1)) ) = -1./TPH
      F( (21+ISSME*(I-1)), (21+ISSME*(I-1)) ) = -1./TTH
      F( (22+ISSME*(I-1)), (22+ISSME*(I-1)) ) = -1./TPOB
C
      S( 15+ISSME*(I-1) ) = 2.*(A( 15+ISSME*(I-1) ))**2/TMR
      S( 16+ISSME*(I-1) ) = 2.*(A( 16+ISSME*(I-1) ))**2/TPC
      S( 17+ISSME*(I-1) ) = 2.*(A( 17+ISSME*(I-1) ))**2/TWDGO2
      S( 18+ISSME*(I-1) ) = 2.*(A( 18+ISSME*(I-1) ))**2/TWDGH2
      S( 19+ISSME*(I-1) ) = 2.*(A( 19+ISSME*(I-1) ))**2/TY8
      S( 20+ISSME*(I-1) ) = 2.*(A( 20+ISSME*(I-1) ))**2/TPH
      S( 21+ISSME*(I-1) ) = 2.*(A( 21+ISSME*(I-1) ))**2/TTH
      S( 22+ISSME*(I-1) ) = 2.*(A( 22+ISSME*(I-1) ))**2/TPOB
C
      H( (7+MSSME*(I-1)), (22+ISSME*(I-1)) ) = 1.0
      H( (7+MSSME*(I-1)), (16+ISSME*(I-1)) ) = 1.0
      H( (8+MSSME*(I-1)), (17+ISSME*(I-1)) ) = 1.0
      H( (9+MSSME*(I-1)), (18+ISSME*(I-1)) ) = 1.0
      H( (10+MSSME*(I-1)), (19+ISSME*(I-1)) ) = 1.0
      H( (11+MSSME*(I-1)), (20+ISSME*(I-1)) ) = 1.0
      H( (12+MSSME*(I-1)), (21+ISSME*(I-1)) ) = 1.0
C
      XLH( 2, I ) = WDGO + X( 17+ISSME*(I-1) )
      XLH( 3, I ) = WDGH + X( 18+ISSME*(I-1) )
      XLH( 4, I ) = Y8 + X( 19+ISSME*(I-1) )
      XLH( 5, I ) = PH + X( 20+ISSME*(I-1) )
      XLH( 6, I ) = TH + X( 21+ISSME*(I-1) )
C
      WDH2H(I) = WDH2 + PWFMR*X( 15+ISSME*(I-1) )
     1                + PWFPC*X( 16+ISSME*(I-1) )
     2                + PWFGO*X( 17+ISSME*(I-1) )
     3                + PWFGH*X( 18+ISSME*(I-1) )
C
      WDO2H(I) = WDO2 + PWOMR*X( 15+ISSME*(I-1) )
     1                + PWOPC*X( 16+ISSME*(I-1) )
     2                + PWOGO*X( 17+ISSME*(I-1) )
     3                + PWOGH*X( 18+ISSME*(I-1) )
C
      WD(I) = WDH2H(I) + WDO2H(I) - (XLH( 2, I) +XLH( 3, I))
C
      TVACL(I) = TVAC + PFTMR(I)*X( 15+ISSME*(I-1) )
     1                + PFTPC(I)*X( 16+ISSME*(I-1) )
     2                + PFTGO(I)*X( 17+ISSME*(I-1) )
     3                + PFTGH(I)*X( 18+ISSME*(I-1) )
C
      XISPL(I) = TVACL(I)/WD(I)
C
      XDOT( 14+ISSME*(I-1) ) = WD(I)
      XDOT( 15+ISSME*(I-1) ) = -(1./TMR)*X( 15+ISSME*(I-1) )
      XDOT( 16+ISSME*(I-1) ) = -(1./TPC)*X( 16+ISSME*(I-1) )
      XDOT( 17+ISSME*(I-1) ) = -(1./TWDGO2)*X( 17+ISSME*(I-1) )
```

```
      XDOT( 18+ISSME*(I-1) ) = -(1./TWDGH2)*X( 18+ISSME*(I-1) )
      XDOT( 19+ISSME*(I-1) ) = -(1./TY8)*X( 19+ISSME*(I-1) )
      XDOT( 20+ISSME*(I-1) ) = -(1./TPH)*X( 20+ISSME*(I-1) )
      XDOT( 21+ISSME*(I-1) ) = -(1./TTH)*X( 21+ISSME*(I-1) )
      XDOT( 22+ISSME*(I-1) ) = -(1./TPOB)*X( 22+ISSME*(I-1) )
C
C   VACUUM THRUST COVARIANCE
C
      VEC1( 15+ISSME*(I-1) ) = PFTMR(I)
      VEC1( 16+ISSME*(I-1) ) = PFTPC(I)
      VEC1( 17+ISSME*(I-1) ) = PFTGO(I)
      VEC1( 18+ISSME*(I-1) ) = PFTGH(I)
C
      DO 20 II = 1, (NS+NPAR)
      SUM = 0.0
      DO 10 JJ = 1, (NS+NPAR)
      SUM = SUM + P(II,JJ)*VEC1(JJ)
   10 CONTINUE
      VEC2(II) = SUM
   20 CONTINUE
C
      CALL INNER ( VEC1, VEC2, COVTVC, (NS+NPAR) )
      CVTVAC(I) = COVTVC
C
C   SPECIFIC IMPULSE COVARIANCE
C
      VEC1( 15+ISSME*(I-1) ) = (PFTMR(I) - XISPL(I)*PWDMR)/WD(I)
      VEC1( 16+ISSME*(I-1) ) = (PFTPC(I) - XISPL(I)*PWDPC)/WD(I)
      VEC1( 17+ISSME*(I-1) ) = (PFTGO(I) - XISPL(I)*PWDGO)/WD(I)
      VEC1( 18+ISSME*(I-1) ) = (PFTGH(I) - XISPL(I)*PWDGH)/WD(I)
C
      DO 40 II = 1, (NS+NPAR)
      SUM = 0.0
      DO 30 JJ = 1, (NS+NPAR)
      SUM = SUM + P(II,JJ)*VEC1(JJ)
   30 CONTINUE
      VEC2(II) = SUM
   40 CONTINUE
C
      CALL INNER ( VEC1, VEC2, COVISP, (NS+NPAR) )
      CVISPL(I) = COVISP
C
C   LIQUID OXYGEN FLOW RATE COVARIANCE
C
      VEC1( 15+ISSME*(I-1) ) = PWOMR
      VEC1( 16+ISSME*(I-1) ) = PWOPC
      VEC1( 17+ISSME*(I-1) ) = PWOGO
      VEC1( 18+ISSME*(I-1) ) = PWOGH
C
      DO 60 II = 1, (NS+NPAR)
      SUM = 0.0
      DO 50 JJ = 1, (NS+NPAR)
      SUM = SUM + P(II,JJ)*VEC1(JJ)
   50 CONTINUE
      VEC2(II) = SUM
```

```
   60 CONTINUE
C
      CALL INNER ( VEC1, VEC2, COVWDO, (NS+NPAR) )
      CVWDO2(I) = COVWLO
C
C    FUEL FLOW RATE COVARIANCE
C
      VEC1( 15+ISSME*(I-1) ) = PWFMR
      VEC1( 16+ISSME*(I-1) ) = PWFPC
      VEC1( 17+ISSME*(I-1) ) = PWFGO
      VEC1( 18+ISSME*(I-1) ) = PWFGH
C
      DO 80 II = 1, (NS+NPAR)
      SUM = 0.0
      DO 70 JJ = 1, (NS+NPAR)
      SUM = SUM + P(II,JJ)*VEC1(JJ)
   70 CONTINUE
      VEC2(II) = SUM
   80 CONTINUE
C
      CALL INNER ( VEC1, VEC2, COVWDH, (NS+NPAR) )
      CVWDH2(I) = COVWDH
C
C    OVERBOARD AND PRESSURANT FLOW RATE COVARIANCES
C
      CVWD(I) = P( 14+ISSME*(I-1), 14+ISSME*(I-1) )
      CVWDGO(I) = P( 17+ISSME*(I-1), 17+ISSME*(I-1) )
      CVWDGH(I) = P( 18+ISSME*(I-1), 18+ISSME*(I-1) )
C
1000  CONTINUE
C
901   FORMAT( 5X, 3E15.8 )
C
      RETURN
      END
      FUNCTION POW( D, PL, M )
C
      DIMENSION D(M)
C
      SUM = 0.0
      DO 10 I = 2, M
      J = M - I + 2
   10 SUM = PL*( D(J) + SUM )
      POW = D(1) + SUM
      RETURN
      END
```

```
      SUBROUTINE RADAR
C
      DIMENSION C1(3,3),RRDR(3),CEFLLR(3,3),DR(3),DRNG(3),DEL(3)
     1         ,VC(3),TMP(3,3),TMP1(3,3)
C
      COMMON DER(2700), VAR(2700), PMET(5400), NDER
      COMMON / TIMDAT / TMAX, HSTEP, TSAMP, TSTART, TSTOP, TRINT
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / PREUP / XKM(71), PKM(71,71)
      COMMON / LINHMT / H(35,71), NMEAS
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / VEHPOS / REF(3)
      COMMON / RDRDAT / XNO(3),RLAT(3),RLONG(3),RHT(3),NR
      COMMON / ASTRON/CUENED(3,3),RNP(3,3),CIBRI(3,3),CBRIB(3,3),TGMTT0
      COMMON / TMAT / CBI(3,3), CIB(3,3), CLLB(3,3), CEFBRI(3,3)
      COMMON / TYPE / ITYPE
      COMMON / CONST / A(71), S(71), R(12)
      COMMON / RDMEAS / AZM(3), ELM(3), RNGM(3)
      COMMON / RDREST / AZHAT(3), ELHAT(3), RNGHAT(3)
      COMMON/UDWORK/RG(71),U(2556),PO(2556),SF(2556),SG(71),RESID,COVZ
      COMMON / QUPDWK / TVEC(71), VEC(71)
C
      DATA C1 / 2*0.0,1.0,1.0,3*0.0,1.0,0.0 /
      DATA CRAD / 57.295779 /
C
      NTS = NS + NPAR
      TIME = VAR(1)
C
      KMEAS = 24
C
      DO 300 IR = 1, NR
C
      DO 10 I = 1, (NS+NPAR)
      X(I) = XKM(I)
10    CONTINUE
C
      CALL XDVEC
C
      CALL ECPOS(RLAT(IR),RLONG(IR),RHT(IR),RRDR(1),RRDR(2),RRDR(3))
      CALL SUBT ( REF, RRDR, VC, 3, 1)
      CALL TMATY ( -RLONG(IR), TMP1 )
      CALL TMATP ( RLAT(IR), CEFLLR )
      CALL MULT ( CEFLLR, TMP1, TMP, 3, 3, 3 )
      CALL MULT ( C1, TMP, CEFLLR, 3, 3, 3 )
      CALL MULT ( CEFLLR, VC, DR, 3, 3, 1 )
      CALL TRANS ( CEFBRI, TMP1, 3, 3 )
      CALL MULT ( CEFLLR, TMP1, TMP, 3, 3, 3 )
C
      RNGES = DR(1)*DR(1) + DR(2)*DR(2) + DR(3)*DR(3)
C
      RNGHAT(IR) = SQRT(RNGES) + XKM( 55 + NR*(IR - 1) )
C
      XYS = RNGES - DR(3)*DR(3)
      XY = SQRT(XYS)
C
```

```
      AZHAT(IR) = CRAD*ATAN2(DR(1),DR(2)) + XKM( 53 + NR*(IR - 1) )
      ELHAT(IR) = CRAD*ATAN(DR(3)/XY) + XKM( 54 + NR*(IR-1) )
      CALL REFRAC ( IR, ELHAT(IR), RNGHAT(IR), DEL(IR), DRNG(IR) )
      RNGHAT(IR) = RNGHAT(IR) + DRNG(IR)
      ELHAT(IR) = ELHAT(IR) + DEL(IR)
      WRITE(*, 997) AZHAT(IR), ELHAT(IR), RNGHAT(IR)
      WRITE(*, 997) AZM(IR), ELM(IR), RNGM(IR)
C
      IF ( ITYPE ) 13, 13, 20
13    CONTINUE
C
      SRNG = SQRT(R(11))
      SELE = SQRT(R(10))
      SAZI = SQRT(R(9))
      CALL NOISE ( SRNG, VRNG )
      CALL NOISE ( SELE, VELE )
      CALL NOISE ( SAZI, VAZI )
      RNGM(IR) = RNGHAT(IR) + VRNG
      AZM(IR) = AZHAT(IR) + VAZI
      ELM(IR) = ELHAT(IR) + VELE
C
      WRITE(2,997) AZM(IR),ELM(IR),RNGM(IR)
      GO TO 300
C
20    CONTINUE
C
      VC(1) = CRAD*DR(2)/XYS
      VC(2) = -CRAD*DR(1)/XYS
      VC(3) = 0.0
C
      H(25+NR*(IR-1),1)=VC(1)*TMP(1,1)+VC(2)*TMP(2,1)+VC(3)*TMP(3,1)
      H(25+NR*(IR-1),2)=VC(1)*TMP(1,2)+VC(2)*TMP(2,2)+VC(3)*TMP(3,2)
      H(25+NR*(IR-1),3)=VC(1)*TMP(1,3)+VC(2)*TMP(2,3)+VC(3)*TMP(3,3)
C
      H(25+NR*(IR-1),53+NR*(IR-1)) = 1.0
C
      VC(1) = -CRAD*DR(1)*DR(3)/(RNGES*XY)
      VC(2) = -CRAD*DR(2)*DR(3)/(RNGES*XY)
      VC(3) = CRAD*XY/RNGES
C
      H(26+NR*(IR-1),1)=VC(1)*TMP(1,1)+VC(2)*TMP(2,1)+VC(3)*TMP(3,1)
      H(26+NR*(IR-1),2)=VC(1)*TMP(1,2)+VC(2)*TMP(2,2)+VC(3)*TMP(3,2)
      H(26+NR*(IR-1),3)=VC(1)*TMP(1,3)+VC(2)*TMP(2,3)+VC(3)*TMP(3,3)
C
      H(26+NR*(IR-1),54+NR*(IR-1)) = 1.0
C
      VC(1) = DR(1)/RNGHAT(IR)
      VC(2) = DR(2)/RNGHAT(IR)
      VC(3) = DR(3)/RNGHAT(IR)
C
      H(27+NR*(IR-1),1)=VC(1)*TMP(1,1)+VC(2)*TMP(2,1)+VC(3)*TMP(3,1)
      H(27+NR*(IR-1),2)=VC(1)*TMP(1,2)+VC(2)*TMP(2,2)+VC(3)*TMP(3,2)
      H(27+NR*(IR-1),3)=VC(1)*TMP(1,3)+VC(2)*TMP(2,3)+VC(3)*TMP(3,3)
C
      H(27+NR*(IR-1),55+NR*(IR-1)) = 1.0
```

```
C
      DO 200 IM = 1, 3
C
      KMEAS = KMEAS + 1
C
      IF ( RNGM(IR).EQ.0.0 ) THEN
      RESID = 0.0
      COVZ = 0.0
      CALL ZEROM( RG, NTS, 1 )
      ELSE
C
C     UDU**T FACTORED COVARIANCE UPDATE
C
      KJ = 0
      DO 140 J = 1, (NS+NPAR)
      RG(J) = H(KMEAS,J)
      DO 140 K = 1, J
      KJ = KJ + 1
      U(KJ) = PKM(J,K)
140   CONTINUE
C
      CALL COV2UD( U, NTS )
C
      RR = R(8+IM)
C
      ALPHA = -1.0
      CALL UDMEAS( U, NTS, RR, RG, SF, SG, ALPHA )
C
      CALL UD2COV( U, PO, NTS )
C
      RESID = AZM(IR) - AZHAT(IR)
      IF( IM.EQ.2 ) RESID = ELM(IR) - ELHAT(IR)
      IF( IM.EQ.3 ) RESID = RNGM(IR) - RNGHAT(IR)
      COVZ = ALPHA
C
C     IF ( ABS(RESID).GT.(6.*SQRT(COVZ)) ) GO TO 180
C
      DO 150 J = 1, (NS+NPAR)
      X(J) = XKM(J) + (SG(J)/ALPHA)*RESID
150   CONTINUE
C
      IJ = 0
      DO 160 J = 1, (NS+NPAR)
      DO 160 I = 1, J
      IJ = IJ + 1
      PKM(I,J) = PO(IJ)
      PKM(J,I) = PKM(I,J)
160   CONTINUE
C
      IF ( TIME - TRINT ) 320, 310, 310
C
310   CONTINUE
C
      DQ1 = XKM(7) - X(7)
      DQ2 = XKM(8) - X(8)
```

```
      DQ3 = XKM(9) - X(9)
      DQS = XKM(7)**2 + XKM(8)**2 + XKM(9)**2
      IF ( DQS.LT.1.0 ) THEN
      DQ4 = -(XKM(7)*DQ1+XKM(8)*DQ2+XKM(9)*DQ3)/SQRT( 1.0 - DQS )
      XKM(10) = X(10) + DQ4
      END IF
      IF( ABS((DQS+XKM(10)**2)-1.0).GT.1.E-4 ) THEN
      XKM(7) = X(7)
      XKM(8) = X(8)
      XKM(9) = X(9)
      XKM(10) = X(10)
      ELSE
      END IF
      TVEC(7) = -XKM(7)/X(10)
      TVEC(8) = -XKM(8)/X(10)
      TVEC(9) = -XKM(9)/X(10)
      CALL MULT ( TVEC, PKM, VEC, 1, NTS, NTS )
      DO 170 I = 1, (NS+NPAR)
      PKM(10,I) = VEC(I)
      PKM(I,10) = VEC(I)
170   CONTINUE
      CALL INNER ( VEC, TVEC, TEMP, NTS )
      PKM(10,10) = TEMP
C
320   CONTINUE
C
180   CONTINUE
C
      END IF
C
      CALL OUTPUT
C
200   CONTINUE
C
300   CONTINUE
C
      RETURN
C
995   FORMAT( 3X, 5E15.8 )
998   FORMAT( 7X, 4E15.8 )
997   FORMAT( 5X, 3E15.8 )
C
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE NASSRB
C
      DIMENSION POHB(2), VEC1(71), VEC2(71)
     1         ,K(2), TTTAU(15), TDTSRB(15)
C
      COMMON DER(2700), VAR(2700), TMP(5400), NDER
      COMMON / TIMDAT / TMAX, HSTEP, TSAMP, TSTART, TSTOP, TRINT
      COMMON / CONST / AA(71), S(71), R(12)
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / DRVITV / XDOT(71), PD(71,71)
      COMMON/SRBTBL/TFSA(15),TTAU(15),TPVOL(15),TAT(15),TCSTR(15)
     1             ,TCT(15),NTAU
     2 ,CMN,G,GAMMA,PE,ABAR,PBAR,EP,RHOP,XLNGTH,RBAR,XMBAR,TEMP,PI
     3 ,DTSEP
      COMMON / SRBDAT / TVACS(2),XISPS(2),XMD(2),AEXIT(2)
     1                 ,CVPOH(2),CVTVAC(2),CVISPS(2),CVMD(2),CVAE(2)
     2                 ,OMASS(2)
      COMMON / RBPRTL / PFT(2), PFA(2), PFCM(2)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / SRBMEA / POHHAT(2)
      COMMON / PONOZZ / PON(2)
      COMMON / LINHMT / H(35,71), NMEAS
C
      DATA TTTAU/0.0,.73,1.73,2.16,3.46,4.75,6.48,8.86
     1          ,12.53,17.28,25.92,34.34,35.42,40.61,43.2/
      DATA TDTSRB/0.,23500.,34000.,20500.,22500.,18000.,17000.,9000.
     1           ,500.,-5500.,-5500.,-6500.,500.,-18500.,0./
      DATA NTTAU, TSF / 15, .9925 /
      DATA XMDOFS / 12. /
      DATA GF / 32.174 /
      DATA TAUA, TAUCM, TAUPB / 1.E+1, 1.E+2, 1.E+3 /
C
      IF ( IJUMP ) 5, 5, 10
    5 CONTINUE
C
      RT = RBAR*TEMP/(XMBAR*GF)
      GM1OG = (GAMMA-1.0)/GAMMA
      GP1GM1 = (GAMMA+1.0)/(GAMMA-1.0)
      F1GAM = ((2.*GAMMA**2)/(GAMMA-1.0))*((2./(GAMMA+1.))**GP1GM1)
      F2GAM = SQRT(F1GAM)
      AN = ABAR*PBAR**(-EP)
      OOMN = 1./(1.-EP)
      XNOOMN = EP*OOMN
      AFAC = ( 150.208 - 149.644 )/43.2
C
      IJUMP = 1
C
   10 CONTINUE
C
      ISRB = 5
C
C   FOR EACH SRB
C
      DO 1000 I = 1, 2
C
```

```
      CALL ZEROM ( VEC1, 71, 1 )
      CALL ZEROM ( VEC2, 71, 1 )
C
      TAU = X(63 + ISRB*(I-1))
C
      AEXIT(I) = (PI/4.)*( 149.644 + AFAC*TAU )**2
C
      CALL INTRP1 ( TAU, TTAU, TFSA, NTAU, AB, PABT )
      CALL INTRP1 ( TAU, TTAU, TPVOL, NTAU, PVOL, PPVOLT )
      CALL INTRP1 ( TAU, TTAU, TAT, NTAU, AT, PATT )
      CALL INTRP1 ( TAU, TTAU, TCSTR, NTAU, CSTAR, PCST )
      CALL INTRP1 ( TAU, TTAU, TCT, NTAU, CT, PCTT )
      CALL INTRP1 ( TAU, TTTAU, TDTSRB, NTTAU, DTSRB, SLOPE )
C
      IF ( AB.LE.0.0 ) AB = 1.0
C
      OMASS(I) = X(62 + ISRB*(I-1))
      A = AN + X(64 + ISRB*(I-1))
      CM = CMN + X(65 + ISRB*(I-1))
C
      PO = (CSTAR*RHOP*A*AB/(G*AT))**OOMN
C
      PON(I) = PO
C
      GEO = 16.*PI*RT*(AT*G/(CSTAR*AB))**2
      GEO = GEO*XLNGTH**3/PVOL
      DP = ( 1.0 + SQRT( 1.0 + GEO ) )/2.0
      POH = PO*DP
      TAUD = A*PO**EP
      XMD(I) = RHOP*TAUD*AB + XMDOFS
C
      TVACS(I) = CM*CT*AT*PO*TSF + DTSRB
C
      PPOT = OOMN*TAUD*(RHOP/G)*((CSTAR/AT)*PABT+(AB/AT)*PCST
     1      -(CSTAR*AB/(AT**2))*PATT)
      PTDT = A*EP*(PO**(EP-1.))*PPOT
      PTDA = OOMN*PO**EP
      PFT(I) = CM*(CT*PPOT*AT+CT*PO*PATT+PO*AT*PCTT)
      PMDA = +RHOP*AB*PTDA
      PPOA = OOMN*((CSTAR*RHOP*AB/(G*AT))**OOMN)*(A**XNOOMN)
      PPOHA = PPOA*DP
      PMDT = +RHOP*(PABT*TAUD + AB*PTDT)
      FPEPO = (PE/PO)**GM1OG
C     PCTPO = 0.5*(F2GAM/SQRT(1.0-FPEPO))*(GM1OG*FPEPO)/PO
C    1      - (PE - PA)*AE/(AT*PO**2)
      PCTPO = 0.0
      PFA(I) = CM*CSTAR*(PCTPO*PPOA*XMD(I) + PMDA*CT)/G
      PFCM(I) = CT*PO*AT
      PPOCS = OOMN*((RHOP*A*AB/(G*AT))**OOMN)*CSTAR**XNOOMN
      PFCS = CM*CT*AT*GF*PPOCS
      PTDCS = A*XNOOMN*((RHOP*A*AB/(G*AT))**XNOOMN)*CSTAR**(XNOOMN-1.)
      PMDCS = +RHOP*AB*PTDCS
      PMDAT = +XNOOMN*XMD(I)/AT
      PTDAT = -XNOOMN*TAUD/AT
      PPOAT = -OOMN*PO/AT
```

```
C
      PPOHT = PPOT*DP
     1       +GEO*(-PPVOLT/PVOL-2.*PABT/AB+2.*PATT/AT-2.*PCST/CSTAR)/DP
      PPOHA = PPOA*DP
      PPOHCS = PPOCS*DP
     1        -GEO/(2.*CSTAR*DP)
      PPOHAT = PPOAT*DP
     1        +GEO/(2.*AT*DP)
C
      F( (62+ISRB*(I-1)), (63+ISRB*(I-1)) ) = PMDT
      F( (62+ISRB*(I-1)), (64+ISRB*(I-1)) ) = PMDA
      F( (63+ISRB*(I-1)), (63+ISRB*(I-1)) ) = PTDT
      F( (63+ISRB*(I-1)), (64+ISRB*(I-1)) ) = PTDA
C
      F( (64+ISRB*(I-1)), (64+ISRB*(I-1)) ) = -1./TAUA
      F( (65+ISRB*(I-1)), (65+ISRB*(I-1)) ) = -1./TAUCM
      F( (66*ISRB*(I-1)), (66+ISRB*(I-1)) ) = -1./TAUPB
C
      S( 64+ISRB*(I-1) ) = 2.*(AA( 64+ISRB*(I-1) ))**2/TAUA
      S( 65+ISRB*(I-1) ) = 2.*(AA( 65+ISRB*(I-1) ))**2/TAUCM
      S( 66+ISRB*(I-1) ) = 2.*(AA( 66+ISRB*(I-1) ))**2/TAUPB
C
      H( (33+I), (62+ISRB*(I-1)) ) = 0.0
      H( (33+I), (63+ISRB*(I-1)) ) = PPOHT
      H( (33+I), (64+ISRB*(I-1)) ) = PPOHA
      H( (33+I), (65+ISRB*(I-1)) ) = 0.0
      H( (33+I), (66+ISRB*(I-1)) ) = 1.0
C
      POHHAT(I) = POH + X( 66+ISRB*(I-1) )
C
      XISPS(I) = TVACS(I)/XMD(I)
C
      XDOT( 62+ISRB*(I-1) ) = XMD(I)
      XDOT( 63+ISRB*(I-1) ) = TAUD
      XDOT( 64+ISRB*(I-1) ) = -(1./TAUA)*X( 64+ISRB*(I-1) )
      XDOT( 65+ISRB*(I-1) ) = -(1./TAUCM)*X( 65+ISRB*(I-1) )
      XDOT( 66+ISRB*(I-1) ) = -(1./TAUPB)*X( 66+ISRB*(I-1) )
C
      IF ( POHHAT(I).LT.50. ) THEN
      K(I) = 1
      ELSE
      K(I) = 0
      END IF
C
      IF ( KJUMP ) 15, 15, 20
15    CONTINUE
      KK = K(1)*K(2)
      IF ( KK.GT.0 ) THEN
      TSTOP = VAR(1) + DTSEP
      WRITE(*, 901) TSTOP, (POHHAT(II), II = 1, 2)
      KJUMP = 1
      ELSE
      END IF
20    CONTINUE
C
```

```
C    HEAD PRESSURE COVARIANCE
C
       VEC1( 63+ISRB*(I-1) ) = PPOHT
       VEC1( 64+ISRB*(I-1) ) = PPOHA
       VEC1( 65+ISRB*(I-1) ) = 0.0
       VEC1( 66+ISRB*(I-1) ) = 1.0
C
       DO 40 II = 1, (NS+NPAR)
       SUM = 0.0
       DO 30 JJ = 1, (NS+NPAR)
       SUM = SUM + P(II,JJ)*VEC1(JJ)
    30 CONTINUE
       VEC2(II) = SUM
    40 CONTINUE
C
       CALL INNER ( VEC1, VEC2, COVPOH, (NS+NPAR) )
       CVPOH(I) = COVPOH
C
C    VACUUM THRUST COVARIANCE
C
       VEC1( 63+ISRB*(I-1) ) = 0.0
       VEC1( 64+ISRB*(I-1) ) = PFA(I)
       VEC1( 65+ISRB*(I-1) ) = PFCM(I)
       VEC1( 66+ISRB*(I-1) ) = 0.0
C
       DO 60 II = 1, (NS+NPAR)
       SUM = 0.0
       DO 50 JJ = 1, (NS+NPAR)
       SUM = SUM + P(II,JJ)*VEC1(JJ)
    50 CONTINUE
       VEC2(II) = SUM
    60 CONTINUE
C
       CALL INNER ( VEC1, VEC2, COVTVC, (NS+NPAR) )
       CVTVAC(I) = COVTVC
C
C    SPECIFIC IMPULSE COVARIANCE
C
       VEC1( 63+ISRB*(I-1) ) = -( RHOP*XISPS(I)*PABT*TAUL
      1                         + EP*G*AT*XISPS(I)*PPOT/CSTAR
      2                         + CM*CT*AT*PPOT )/XMD(I)
       VEC1( 64+ISRB*(I-1) ) = -( XMD(I)*XISPS(I)/A
      1                         + EP*G*AT*XISPS(I)*PPOA/CSTAR
      2                         + CM*CT*AT*PPOA )/XMD(I)
       VEC1( 65+ISRB*(I-1) ) = CT*AT*PO/XMD(I)
CSTAR  VEC1( 65+ISRB*(I-1) ) = CM*CT/G
       VEC1( 66+ISRB*(I-1) ) = 0.0
C
       DO 80 II = 1, (NS+NPAR)
       SUM = 0.0
       DO 70 JJ = 1, (NS+NPAR)
       SUM = SUM + P(II,JJ)*VEC1(JJ)
    70 CONTINUE
       VEC2(II) = SUM
    80 CONTINUE
```

```
C
      CALL INNER ( VEC1, VEC2, COVISP, (NS+NPAR) )
      CVISPS(I) = COVISP
C
C   MASS OVERBOARD RATE COVARIANCE
C
      VEC1( 63+ISRB*(I-1) ) = 0.0
      VEC1( 64+ISRB*(I-1) ) = PMDA
      VEC1( 65+ISRB*(I-1) ) = 0.0
      VEC1( 66+ISRB*(I-1) ) = 0.0
C
      DO 100 II = 1, (NS+NPAR)
      SUM = 0.0
      DO 90 JJ = 1, (NS+NPAR)
      SUM = SUM + P(II,JJ)*VEC1(JJ)
   90 CONTINUE
      VEC2(II) = SUM
  100 CONTINUE
C
      CALL INNER ( VEC1, VEC2, COVMD, (NS+NPAR) )
      CVMD(I) = COVMD
C
C   EXIT AREA COVARIANCE
C
      VEC1( 63+ISRB*(I-1) ) = (PI/2.)*AFAC*TAU
      VEC1( 64+ISRB*(I-1) ) = 0.0
      VEC1( 65+ISRB*(I-1) ) = 0.0
      VEC1( 66+ISRB*(I-1) ) = 0.0
C
      DO 120 II = 1, (NS+NPAR)
      SUM = 0.0
      DO 110 JJ = 1, (NS+NPAR)
      SUM = SUM + P(II,JJ)*VEC1(JJ)
  110 CONTINUE
      VEC2(II) = SUM
  120 CONTINUE
C
      CALL INNER ( VEC1, VEC2, COVAE, (NS+NPAR) )
      CVAE(I) = COVAE
C
1000  CONTINUE
C
901   FORMAT( 5X, 3E15.8 )
902   FORMAT( 5X, 5E15.8 )
991   FORMAT( 2X,7E10.4 )
992   FORMAT( 12X,2E10.4 )
C
      RETURN
      END
```

```
      SUBROUTINE NAERO ( XMACH, ALPHA, BETA, DO2V )
C
C
      DIMENSION TMACH(15),TCA0(15),TCAA(15),TCAAB2(15)
     1        ,TCN0(15),TCNA(15),TCNAB2(15),TCAA2(15),TCDAB2(15)
     2        ,TCLM0(15),TCLMA(15),TCMAB2(15)
     3         ,TCY0(15),TCYB(15),TCYAB(15),TCYA2B(15)
     4         ,TCLN0(15),TCLNB(15),TCLNAB(15),TCNA2B(15)
     5        ,TCLL0(15),TCLLB(15),TCLLAB(15),TCLA2B(15)
      DIMENSION TCYP0(15),TCYPA(15),TCYR0(15),TCYRA(15)
     1          ,TCLMQ0(15),TCLMQA(15)
     2          ,TCLNP0(15),TCLNPA(15),TCLNR0(15),TCLNRA(15)
     3          ,TCLLP0(15),TCLLPA(15),TCLLR0(15),TCLLRA(15)
C
C
       DIMENSION TMACH2(5),TCA20(5),TCA2A(5),TCA2A2(5)
     1       ,TCN20(5),TCN2A(5),TCN2A2(5)
     2  ,TCLM20(5),TCLM2A(5),TCM2A2(5)
     3   ,TCY20(5),TCY2B(5),TCY2AB(5),TY2A2B(5)
     4   ,TCLN20(5),TCLN2B(5),TLN2AB(5),TN2A2B(5)
     5   ,TCLL20(5),TCLL2B(5),TCL2AB(5),TL2A2B(5)
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / CONST / A(71), S(71), RR(12)
      COMMON / STATES / RI(3), VB(3), Q(4), OMEGA(3)
      COMMON / PROPAG / X(71), PMAT(71,71)
      COMMON / DRVITV / XDOT(71), PD(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / AERO / CF0(3),CFALP(3),CFBET(3),CFQ(3,3),CM0(3),CMALP(3)
     1      , CMBET(3),CMQ(3,3),CF(3),CM(3)
      COMMON / STAGE / ISTAGE
      COMMON / IPARAM / ISSME,ISRB,IAERO,IPLUME,IWIND,JACB,JRDR
C
      DATA CAAB2,CAB2,CAA2,CNAB2,CYAB,CYA2B,CYPA,CYRA / 8*0.0 /
C
      DATA TAUF / 2.E+1 /
C
      DATA TMACH / 0.6,0.8,0.9,0.95,1.0,1.1,1.15,1.25,1.4,1.55
     1    ,1.8,2.2,2.5,3.5,4.5/
      DATA TMACH2 /3.5,4.5,6.0,8.0,10.0/
C
C
      DATA TCLL0/0.00009800,-0.00100400,0.00119100,0.00113500,
     1 0.00102300,0.00255850,0.00142150,0.00088950,0.00162650,
     2 0.00162500,0.00160650,0.00164350,0.00175700,0.00078500,
     3 -0.00035500/
      DATA TCLLB/-0.55153607E-02,-0.51660468E-02,-0.53059668E-02,
     1 -0.57055471E-02,-0.65046088E-02,-0.73706545E-02,-0.64209271E-02,
     2 -0.61848979E-02,-0.60260380E-02,-0.57253810E-02,-0.55433516E-02,
     3 -0.53707338E-02,-0.46027126E-02,-0.38413065E-02,-0.32225023E-02/
      DATA TCLLAB/-0.18314904E-03,-0.24067984E-03,-0.17415131E-03,
     1 -0.18708523E-03,-0.21297160E-03,0.23723571E-04,-0.10909401E-03,
     2 -0.15336619E-03,-0.40835867E-03,-0.20703886E-03,-0.17273379E-03,
     3 -0.55630488E-04,-0.66665525E-04,-0.45428569E-04,-0.65075328E-04/
      DATA TCLA2B/0.25525271E-05,-0.47074923E-05,0.40663767E-6,
```

```
     1 0.26400435E-05,0.71316317E-05,0.41424133E-04,0.19684549E-04,
     2 0.76067136E-05,-0.20967154E-04,0.41421335E-05,0.71404006E-05,
     3 0.12803971E-04,0.22202585E-05,-0.94020070E-05,-0.85059000E-05/
C
C
      DATA TCLN0/0.00101900,-0.00196700,0.00001700,-0.00033150,
     1 -0.00102900,-0.00056750,-0.00357700,-0.00415650,-0.00269050,
     2 -0.00395800,-0.00297900,-0.00073950,-0.00348650,-0.00259800,
     3 0.00097000/
      DATA TCLNB/0.14767525E-01,0.15323062E-01,0.15911508E-01,
     1 0.16263550E-01,0.16967123E-01,0.15625395E-01,0.15465586E-01,
     2 0.15171567E-01,0.14852828E-01,0.15761562E-01,0.17182207E-01,
     3 0.18076595E-01,0.16406817E-01,0.13792288E-01,0.11427909E-01/
      DATA TCLNAB/0.56945679E-04,0.11934886E-03,-0.12992905E-04,
     1 -0.21390285E-05,0.19518508E-04,0.11725974E-04,0.40781717E-04,
     2 0.17625467E-03,0.32387782E-03,0.23744810E-03,0.21290279E-03,
     3 0.11312054E-03,-0.58733131E-04,-0.43780621E-03,-0.46182974E-03/
      DATA TCNA2B/0.81995468E-05,0.30770094E-04,0.26990148E-04,
     1 0.21246726E-04,0.97513848E-05,0.28817041E-04,0.19518948E-04,
     2 0.30797459E-04,0.55297049E-04,0.29008272E-04,0.23987936E-05,
     3 0.10975672E-04,0.34038654E-04,0.47138979E-05,-0.30127571E-05/
C
C
      DATA TCY0/0.00040550,0.00611750,0.00208050,0.00340600,
     1 0.00659450,0.00389250,0.00896900,0.01099600,0.00895500,
     2 0.00684850,0.00122250,-0.00312350,0.00190050,0.00643500,
     3 0.00000500/
        DATA TCYB/-0.34736611E-01,-0.35624493E-01,-0.37405483E-01,
     1 -0.37516881E-01,-0.37924547E-01,-0.35857856E-01,
     2 -0.34745876E-01,-0.33643633E-01,-0.32880556E-01,
     3 -0.36689218E-01,-0.37921809E-01,-0.39925124E-01,
     4 -0.38449090E-01,-0.34494311E-01,-0.30585317E-01/
        DATA TCYAB/-0.24396958E-03,-0.19144324E-03,-0.96200623E-04,
     1 -0.16656239E-03,-0.28471148E-03,-0.67672569E-04,
     2 -0.10685262E-03,-0.21200742E-03,-0.30565454E-03,
     3 -0.21161194E-03,0.23066868E-03,0.12333597E-03,
     4  0.28201332E-03,0.69754681E-03,0.73324196E-03/
        DATA TCYA2B/-0.34743120E-04,-0.68390669E-04,-0.65476503E-04,
     1 -0.65729291E-04,-0.51625058E-04,-0.67499590E-04,
     2 -0.56302903E-04,-0.70274968E-04,-0.94529889E-04,
     3 -0.58715108E-04,0.11196062E-04,-0.23788327E-04,
     4 -0.50999923E-04,-0.20773427E-04,-0.23277323E-05/
C
C
        DATA TCA0/0.14888462,0.14096716,0.15895303,0.18570921,
     1 0.24431951,0.25983587,0.26992825,0.28716668,0.30411604,
     2 0.30857432,0.30735463,0.28761023,0.28410554,0.28069207,
     3 0.28001353/
      DATA TCAA/-0.14939267E-02,-0.16073198E-02,-0.87178504E-03,
     1 -0.80946332E-03,-0.68517815E-03,-0.53749926E-03,
     2 -0.20928589E-03,-0.47267828E-03,-0.53303648E-03,
     3 -0.51839335E-03,-0.11330346E-02,-0.97124977E-03,
     4 -0.12258916E-02,-0.27267837E-02,-0.40642847E-02/
      DATA TCAAB2/0.52950736E-05,0.80431100E-05,-0.79241381E-05,
     1 -0.97817874E-05,-0.85392085E-05,0.31472900E-05,
```

```
     2 -0.46378850E-05,-0.17535077E-05,-0.35093940E-05,
     3 -0.30256535E-06,0.91515847E-06,-0.10002519E-04,
     4 -0.14255994E-04,-0.18130060E-04,-0.13194465E-04/
      DATA TCDAB2/-0.19513888E-03,-0.15666662E-03,-0.10849207E-03,
     1 -0.14827383E-03,-0.20781747E-03,-0.17412701E-03,
     2 -0.13521819E-03,-0.12434516E-03,-0.19692448E-03,
     3 -0.21642850E-03,-0.18192461E-03,-0.89146888E-04,
     4 -0.38432758E-04,0.14365064E-03,0.15932534E-03/
      DATA TCAA2/-0.36778764E-03,-0.29035713E-03,-0.17375973E-03,
     1 -0.22280747E-03,-0.32113091E-03,-0.28126000E-03,
     2 -0.23748010E-03,-0.26166640E-03,-0.16787715E-03,
     3 -0.12008918E-03,-0.79305413E-04,-0.40277255E-05,
     4 0.16438302E-04,0.11210326E-03,0.14424580E-03/
C
C
      DATA TCLM0/-0.08278762,-0.10786715,-0.09791238,-0.08543953,
     1 -0.07620762,-0.06862953,-0.07373904,-0.07957237,-0.07712714,
     2 -0.06862857,-0.05165952,-0.02029048,0.00235619,0.01028095,
     3 0.01356667/
      DATA TCLMA/-0.15543575E-01,-0.18000530E-01,-0.19910170E-01,
     1 -0.20583043E-01,-0.21927318E-01,-0.20931607E-01,
     2 -0.20659992E-01,-0.20311428E-01,-0.20116061E-01,
     3 -0.20584987E-01,-0.22142328E-01,-0.19702138E-01,
     4 -0.18991778E-01,-0.13785718E-01,-0.11521422E-01/
      DATA TCMAB2/0.16693601E-05,0.21009337E-04,0.46228233E-05,
     1 0.85657601E-04,0.21230151E-05,-0.26527516E-04,
     2 0.65993113E-05,-0.95234293E-06,-0.17661379E-04,
     3 0.12942927E-05,0.42312208E-05,-0.14325634E-04,
     4 -0.52894680E-04,-0.24528768E-04,-0.27753043E-04/
C
C
      DATA TCN0/0.06175191,0.08705191,0.06880905,0.06304239,
     1 0.05150905,0.04629619,0.05327953,0.06288191,0.05508667,
     2 0.04784143,0.02033000,-0.00378286,-0.03606287,
     3 -0.04565714,-0.05337614/
      DATA TCNA/0.47317315E-01,0.51266242E-01,0.55013947E-01,
     1 0.55984642E-01,0.57926252E-01,0.58463752E-01,
     2 0.56734987E-01,0.56824278E-01,0.56955535E-01,
     3 0.56824278E-01,0.57594139E-01,0.52612297E-01,
     4 0.51134817E-01,0.39903577E-01,0.36437500E-01/
      DATA TCNAB2/-0.34952373E-06,-0.21971118E-04,0.37438092E-05,
     1 0.12846999E-04,0.31073960E-04,0.18769655E-04,0.49979085E-05,
     2 0.19573376E-04,0.44349854E-04,0.24865894E-04,-0.13175871E-04,
     3 0.82223523E-05,0.49027673E-04,0.62995699E-04,0.43675380E-04/
C
C
        DATA TCY20/-0.00283400,-0.00476067,-0.00862133,-0.01420600,
     1 -0.01559400/
        DATA TCY2B/-0.35310987E-01,-0.32411996E-01,-0.28534999E-01,
     1  -0.25360515E-01,-0.23243498E-01/
        DATA TCY2AB/0.67670009E-03,0.65709988E-03,0.60950016E-03,
     1 0.55275002E-03,0.48189994E-03/
        DATA TY2A2B/0.41799358E-04,0.32219676E-04,0.41402013E-05,
     1 -0.10169592E-04,0.23179970E-04/
C
```

```
C
      DATA TCLN20/0.00310133,0.00433133,0.00328200,0.00237733,
     1 0.00358800/
      DATA TCLN2B/0.14233000E-01,0.11893496E-01,0.90009952E-02,
     1 0.77250013E-02,0.71255034E-02/
      DATA TLN2AB/-0.41310006E-03,-0.37830003E-03,-0.31595002E-03,
     1 -0.28525002E-03,-0.27645013E-03/
      DATA TN2A2B/-0.87519926E-04,-0.43759825E-04,0.36501278E-05,
     1 -0.22901133E-05,-0.23530063E-04/
C
C
      DATA TCLL20/-0.00099533,-0.00212667,-0.00141267,-0.00262400,
     1 -0.00266133/
      DATA TCLL2B/-0.55765007E-02,-0.46779984E-02,-0.38770009E-02,
     1 -0.33380005E-02,-0.32379995E-02/
      DATA TCL2AB/0.17604997E-03,0.11775004E-03,-0.11815000E-03,
     1 0.46599998E-04,0.14849989E-04/
      DATA TL2A2B/-0.12449930E-04,0.12530111E-04,0.31110012E-04,
     1 -0.91998515E-06,0.11949987E-04/
C
C
      DATA TCA20/0.17786667,0.16868572,0.16880476,
     1 0.17327619,0.16560951/
      DATA TCA2A/-0.38053570E-02,-0.42267856E-02,-0.43714279E-02,
     1 -0.46357140E-02,-0.47821430E-02/
      DATA TCA2A2/0.11815455E-03,0.16160712E-03,0.12023804E-03,
     1 0.10416679E-03,0.14226201E-03/
C
C
      DATA TCN20/-0.08282144,-0.07680953,-0.06490477,
     1 -0.06376192,-0.06909525/
      DATA TCN2A/0.27379462E-01,0.24214283E-01,0.21875000E-01,
     1 0.20946428E-01,0.20232143E-01/
      DATA TCN2A2/0.24330383E-03,0.18452371E-03,0.38690865E-04,
     1 -0.32738029E-04,-0.29761522E-05/
C
C
      DATA TCLM20/0.05514286,0.04723810,0.04276191,0.03900000,
     1 0.03719047/
      DATA TCLM2A/-0.10821427E-01,-0.86785713E-02,-0.68392851E-02,
     1 -0.56071421E-02,-0.50357138E-02/
      DATA TCM2A2/-0.17857179E-04,0.11904769E-03,0.86309483E-04,
     1 0.89285677E-04,0.11309532E-03/
C
C
C   DATA FOR MACH 4.5 ONLY
C
      DATA CLMQ20,CLNP20,CLNR20,CLLP20,CLLR20/
     1 -.936,.225,-.799,-.145,.195/
C
C*****************************************************************************
C
      DATA TCLMQ0/-.2520E1,-.2731E1,-.2880E1,-.2972E1,-.3164E1,-.3248E1,
     1 -.3112E1,-.3354E1,-.3189E1,-.2835E1,-.2082E1,-.1375E1,
     2 -.1300E1,-.1031E1,-.1069E1/
```

```
C
C
      DATA TCLMQA/.375E-2,-.2382E-1,-.2000E-1,-.8209E-2,.2947E-1,
     1 .5103E-1,.71509E-1,.10104E0,.11064E0,.90109E-1,.2674E-1,
     2 -.1699E-1,.1250E-1,.3607E-1,.3945E-1/
C
C
      DATA TCLNP0/.240E0,.271E0,.285E0,.291E0,.302E0,.307E0,.312E0,
     1 .320E0,.329E0,.333E0,.325E0,.263E0,.227E0,.234E0,.152E0/
C
C
      DATA TCLNPA/-.123E-1,-.626E-1,-.387E-2,-.292E-2,-.143E-2,
     1 -.850E-3,-.340E-3,.550E-3,.168E-2,.259E-2,.350E-2,.337E-2,
     2 .560E-2,-.512E-2,-.155E-2/
C
C
      DATA TCLNR0/.5000E0,-.9310E0,-.1570E1,-.1746E1,-.1839E1,-.1800E1,
     1 -.1733E1,-.1598E1,-.1493E1,-.1481E1,-.1572E1,-.1662E1,
     2 -.1580E1,-.1106E1,-.9080E1/
C
C
      DATA TCLNRA/-.3750E-1,-.5136E-1,-.4000E-1,-.3331E-1,-.2062E-1,
     1 -.1510E-1,-.1045E-1,-.4760E-2,-.4910E-2,-.1168E-1,-.2651E-1,
     2 -.2070E-1,-.6900E-2,-.2168E-1,-.7200E-3/
C
C
      DATA TCLLP0/-.295E0,-.259E0,-.251E0,-.256E0,-.279E0,-.294E0,
     1 -.309E0,-.331E0,-.343E0,-.333E0,-.290E0,-.228E0,-.218E0,
     2 -.207E0,-.168E0/
C
C
      DATA TCLLPA/.100E-2,-.137E-2,-.162E-2,-.127E-2,.900E-4,
     1 .940E-3,.181E-2,.324E-2,.438E-2,.465E-2,.399E-2,.278E-2,
     2 .356E-2,.386E-2,.344E-2/
C
C
      DATA TCLLR0/.282E0,.438E0,.481E0,.482E0,.449E0,.423E0,
     1 .394E0,.342E0,.294E0,.274E0,.275E0,.274E0,.250E0,.167E0,
     2 .122E0/
C
C
      DATA TCLLRA/.1470E-1,.1874E-1,.1710E-1,.1572E-1,.1254E-1,
     1 .1097E-1,.9590E-2,.7890E-2,.8119E-2,.1044E-1,.1561E-1,
     2 .1561E-1,.1240E-1,.7230E-1,-.9360E-2/
C
C
      DATA NMACH, NMACH2 / 15, 5 /
C
        P = OMEGA(1)
        R = OMEGA(3)
C
      CALL ZEROM( CFQ, 3, 3 )
      CALL ZEROM( CFALP, 3, 1 )
      CALL ZEROM( CFBET, 3, 1 )
C
```

```
      ALPHAS = ALPHA*ALPHA
      BETAS = BETA*BETA
      PDO2V = P*DO2V
      RDO2V = R*DO2V
C
      IF ( ISTAGE.EQ.1 ) THEN
C
      IF ( XMACH.LT.0.6 ) XMACH = 0.6
C
C      AXIAL COEFFICIENT
C
      CALL INTRP1 ( XMACH, TMACH, TCA0, NMACH, CA0, SLOPE )
      CALL INTRP1 ( XMACH, TMACH, TCAA, NMACH, CAA, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCAAB2, NMACH, CAAB2, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCAA2, NMACH, CAA2, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCDAB2, NMACH, CAB2, SLOPE )
C
      CF0(1) = -CA0
      CAALP = CAA + CAA2*ALPHA + CAAB2*BETAS
      CABET = (CAAB2*ALPHA + CAB2)*BETA
      CFALP(1) = -CAALP
      CFBET(1) = -CABET
C
C      NORMAL FORCE COEFFICIENT
C
      CALL INTRP1 ( XMACH, TMACH, TCN0, NMACH, CN0, SLOPE )
      CALL INTRP1 ( XMACH, TMACH, TCNA, NMACH, CNA, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCNAB2, NMACH, CNAB2, SLOPE )
C
      CF0(3) = -CN0
      CNALP = CNA + CNAB2*BETAS
      CNBET = CNAB2*ALPHA*BETA
      CFALP(3) = -CNALP
      CFBET(3) = -CNBET
C
C      PITCH MOMENT COEFFICIENT
C
C     CALL INTRP1 ( XMACH, TMACH, TCLM0, NMACH, CLM0, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLMA, NMACH, CLMA, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCMAB2, NMACH, CLMAB2, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLMQ0, NMACH, CLMQ0, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLMQA, NMACH, CLMQA, SLOPE )
C
C     CM0(2) = CLM0
C     CLMALP = CLMA + CLMAB2*BETAS
C     CLMBET = (CLMAB2*ALPHA + CLMB2)*BETA
C     CLMQ = CLMQ0 + CLMQA*ALPHA
C     CMALP(2) = CLMALP
C     CMBET(2) = CLMBET
C     CMQ(2,2) = CLMQ
C
C      SIDE FORCE COEFFICIENT
C
      CALL INTRP1 ( XMACH, TMACH, TCY0, NMACH, CY0, SLOPE )
      CALL INTRP1 ( XMACH, TMACH, TCYB, NMACH, CYB, SLOPE )
```

```
C     CALL INTRP1 ( XMACH, TMACH, TCYAB, NMACH, CYAB, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCYA2B, NMACH, CYA2B, SLOPE )
      CALL INTRP1( XMACH, TMACH, TCYP0, NMACH, CYP0, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCYPA, NMACH, CYPA, SLOPE )
      CALL INTRP1 ( XMACH, TMACH, TCYR0, NMACH, CYR0, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCYRA, NMACH, CYRA, SLOPE )
C
      CF0(2) = CY0
      CYALP = (CYAB + CYA2B*ALPHA)*BETA + CYPA*PDO2V + CYRA*RDO2V
      CYBET = CYB + CYAB*ALPHA + CYA2B*ALPHAS
      CYP = CYP0 + CYPA*ALPHA
      CYR = CYR0 + CYRA*ALPHA
      CFALP(2) = CYALP
      CFBET(2) = CYBET
      CFQ(2,1) = CYP
      CFQ(2,3) = CYR
C
C      YAW MOMENT COEFFICIENT
C
C     CALL INTRP1 ( XMACH, TMACH, TCLN0, NMACH, CLN0, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLNB, NMACH, CLNB, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLNAB, NMACH, CLNAB, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCNA2B, NMACH, CLNA2B, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLNP0, NMACH, CLNP0, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLNPA, NMACH, CLNPA, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLNR0, NMACH, CLNR0, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLNRA, NMACH, CLNRA, SLOPE )
C
C     CM0(3) = CLN0
C     CLNBET = CLNB + CLNAB*ALPHA + CLNA2B*ALPHAS
C     CLNALP = (CLNAB+CLNA2B*ALPHA)*BETA+CLNPA*PDO2V+CLNRA*RDO2V
C     CLNP = CLNP0 + CLNPA*ALPHA
C     CLNR = CLNR0 + CLNRA*ALPHA
C     CMALP(3) = CLNALP
C     CMBET(3) = CLNBET
C     CMQ(3,1) = CLNP
C     CMQ(3,3) = CLNR
C
C      ROLL MOMENT COEFFICIENT
C
C     CALL INTRP1 ( XMACH, TMACH, TCLL0, NMACH, CLL0, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLLB, NMACH, CLLB, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLLAB, NMACH, CLLAB, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLA2B, NMACH, CLLA2B, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLLP0, NMACH, CLLP0, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLLPA, NMACH, CLLPA, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLLR0, NMACH, CLLR0, SLOPE )
C     CALL INTRP1 ( XMACH, TMACH, TCLLRA, NMACH, CLLRA, SLOPE )
C
C     CM0(1) = CLL0
C     CLLBET = CLLB + CLLAB*ALPHA + CLLA2B*ALPHAS
C     CLLALP = (CLLAB+CLLA2B*ALPHA)*BETA+CLLPA*PDO2V+CLLRA*RDO2V
C     CLLP = CLLP0 + CLLPA*ALPHA
C     CLLR = CLLR0 + CLLRA*ALPHA
C     CMALP(1) = CLLALP
```

```
C      CMBET(1) = CLLBET
C      CMQ(1,1) = CLLP
C      CMQ(1,3) = CLLR
C
       ELSE
C
       IF ( XMACH.LT.4.5 ) XMACH=4.5
C
C    AXIAL FORCE COEFFICIENT
C
       CALL INTRP1 ( XMACH, TMACH2, TCA20, NMACH2, CA20, SLOPE )
       CALL INTRP1 ( XMACH, TMACH2, TCA2A, NMACH2, CA2A, SLOPE )
       CALL INTRP1 ( XMACH, TMACH2, TCA2A2, NMACH2, CA2A2, SLOPE )
C
       CF0(1) = -CA20
       CAALP = CA2A + CA2A2*ALPHA
       CFALP(1) = -CAALP
C
C    NORMAL FORCE COEFFICIENT
C
       CALL INTRP1 ( XMACH, TMACH2, TCN20, NMACH2, CN20, SLOPE )
       CALL INTRP1 ( XMACH, TMACH2, TCN2A, NMACH2, CN2A, SLOPE )
       CALL INTRP1 ( XMACH, TMACH2, TCN2A2, NMACH2, CN2A2, SLOPE )
C
       CF0(3) = -CN20
       CNALP = CN2A + CN2A2*ALPHA
       CFALP(3) = -CNALP
C
C    PITCH MOMENT COEFFICIENT
C
C      CALL INTRP1 ( XMACH, TMACH2, TCLM20, NMACH2, CLM20, SLOPE )
C      CALL INTRP1 ( XMACH, TMACH2, TCLM2A, NMACH2, CLM2A, SLOPE )
C      CALL INTRP1 ( XMACH, TMACH2, TCM2A2, NMACH2, CLM2A2, SLOPE )
C
C      CM0(2) = CLM20
C      CLMALP = CLM2A + CLM2A2*ALPHA
C      CMALP(2) = CLMALP
C      CMQ(2,2) = CLMQ20
C
C    SIDE FORCE COEFFICIENT
C
       CALL INTRP1 ( XMACH, TMACH2, TCY20, NMACH2, CY20, SLOPE )
       CALL INTRP1 ( XMACH, TMACH2, TCY2B, NMACH2, CY2B, SLOPE )
       CALL INTRP1 ( XMACH, TMACH2, TCY2AB, NMACH2, CY2AB, SLOPE )
       CALL INTRP1 ( XMACH, TMACH2, TY2A2B, NMACH2, CY2A2B, SLOPE )
C
       CF0(2) = CY20
       CYALP = (CY2AB + CY2A2B*ALPHA)*BETA
       CYBET = CY2B + CY2AB*ALPHA + CY2A2B*ALPHAS
       CFALP(2) = CYALP
       CFBET(2) = CYBET
C
C    YAW MOMENT COEFFICIENT
C
C      CALL INTRP1 ( XMACH, TMACH2, TCLN20, NMACH2, CLN20, SLOPE )
```

```
C      CALL INTRP1 ( XMACH, TMACH2, TCLN2B, NMACH2, CLN2B, SLOPE )
C      CALL INTRP1 ( XMACH, TMACH2, TLN2AB, NMACH2, CLN2AB, SLOPE )
C      CALL INTRP1 ( XMACH, TMACH2, TN2A2B, NMACH2, CN2A2B, SLOPE )
C
C      CM0(3) = CLN20
C      CLNBET = CLN2B + CLN2AB*ALPHA + CN2A2B*ALPHAS
C      CLNALP = (CLN2AB + CN2A2B*ALPHA)*BETA
C      CMALP(3) = CLNALP
C      CMBET(3) = CLNBET
C      CMQ(3,1) = CLNP20
C      CMQ(3,3) = CLNR20
C
C    ROLL MOMENT COEFFICIENT
C
C      CALL INTRP1 ( XMACH, TMACH2, TCLL20, NMACH2, CLL20, SLOPE )
C      CALL INTRP1 ( XMACH, TMACH2, TCLL2B, NMACH2, CLL2B, SLOPE )
C      CALL INTRP1 ( XMACH, TMACH2, TCL2AB, NMACH2, CLL2AB, SLOPE )
C      CALL INTRP1 ( XMACH, TMACH2, TL2A2B, NMACH2, CL2A2B, SLOPE )
C
C      CM0(1) = CLL20
C      CLLBET = CLL2B + CLL2AB*ALPHA + CL2A2B*ALPHAS
C      CLLALP = (ALL2AB + CL2A2B*ALPHA)*BETA
C      CMALP(1) = CLLALP
C      CMBET(2) = CLLBET
C      CMQ(1,1) = CLLP20
C      CMQ(1,3) = CLLR20
C
       END IF
C
       IF( IAERO.GT.0 ) THEN
C
       OOTAUF = 1./TAUF
C
       F(41,41) = -OOTAUF
       F(42,42) = -OOTAUF
       F(43,43) = -OOTAUF
C
       S(41) = 2.*A(41)**2/TAUF
       S(42) = 2.*A(42)**2/TAUF
       S(43) = 2.*A(43)**2/TAUF
C
       XDOT(41) = -OOTAUF*X(41)
       XDOT(42) = -OOTAUF*X(42)
       XDOT(43) = -OOTAUF*X(43)
C
       CF0(1) = CF0(1) + X(41)
       CF0(2) = CF0(2) + X(42)
       CF0(3) = CF0(3) + X(43)
C
       ELSE
       END IF
C
       RETURN
C
901    FORMAT( 10X, 3E15.8 )
```

END

```
      SUBROUTINE NPLUME ( ALPHA, BETA, ALT, Q, PL )
C
C
      DIMENSION TALT(50),TK1(50),TK2(50),TKA(50)
     1         ,TKB(50),TDFNB(50),TDFAB(50),TDMMB(50),TQREF(50)
     2         ,TFAORB(50)
      DIMENSION TALT2(30),TQREF2(30),TDFNB2(30),TDFAB2(30),TDMMB2(30)
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / CONST / A(71), S(71), R(12)
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / DRVITV / XDOT(71), PD(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / PLUME / FPLUME(3),FPALP(3),FPBET(3),TPLUME(3),TPALP(3)
     1 ,TPBET(3),FPALT(3),TPALT(3),FPDPL(3),TPDPL(3)
      COMMON / IPARAM / ISSME,ISRB,IAERO,IPLUME,IWIND,JACB,JRDR
      COMMON / STAGE / ISTAGE
C
      DATA TAUF / 2.E+1 /
      DATA XKB / 0.0 /
C
      CALL ZEROM( FPLUME, 3, 1 )
      CALL ZEROM( TPLUME, 3, 1 )
C
      DATA TALT/.519E3,.250E4,.500E4,.750E4,.100E5,.125E5,.150E5,
     1    .175E5,.190E5,.200E5,.210E5,.220E5,.225E5,.230E5,.235E5,
     2    .240E5,.250E5,.260E5,.270E5,.280E5,.290E5,.300E5,.320E5,
     3    .340E5,.360E5,.380E5,.400E5,.420E5,.440E5,.460E5,.480E5,
     4    .500E5,.525E5,.550E5,.575E5,.600E5,.650E5,.700E5,.750E5,
     5    .800E5,.850E5,.900E5,.950E5,.100E6,.110E6,.120E6,.130E6,
     6    .140E6,.145E6,.160E6/
C
C
      DATA TDFNB/.00000E4,.35620E4,.68230E4,.88080E4,.10264E5,
     1 .11058E5,.11343E5,.11536E5,.11219E5,.10883E5,.10960E5,
     2 .11440E5,.11978E5,.12575E5,.13158E5,.13634E5,.14185E5,
     3 .15697E5,.15783E5,.15289E5,.14357E5,.13242E5,.10722E5,
     4 .82690E4,.64340E4,.49170E4,.37280E4,.28310E4,.19060E4,
     5 .10100E4,.20000E3,-.56699E3,-.14070E4,-.20430E4,-.25840E4,
     6 -.30710E4,-.37330E4,-.40780E4,-.39440E4,-.33340E4,-.26700E4,
     7 -.22940E4,-.19850E4,-.17050E4,-.14860E4,-.13190E4,-.11490E4,
     8 -.10240E4,-.95899E3,-.76399E3/
C
C
      DATA TDMMB/.000000E6,-.126045E6,-.342060E6,-.442584E6,
     1 -.512378E6,-.533795E6,-.537157E6,-.545920E6,-.525332E6,
     2 -.502047E6,-.496484E6,-.510283E6,-.536406E6,-.565396E6,
     3 -.594163E6,-.617065E6,-.645573E6,-.720716E6,-.732389E6,
     4 -.718480E6,-.677641E6,-.617735E6,-.474167E6,-.323480E6,
     5 -.190900E6,-.786510E6,.241000E3,.415890E5,.764280E5,
     6 .115649E6,.156796E6,.201963E6,.250782E6,.284125E6,.307795E6,
     7 .327006E6,.347164E6,.350765E6,.324147E6,.263832E6,.203965E6,
     8 .170470E6,.146288E6,.127146E6,.114200E6,.10350736,.901060E5,
     9 .810620E5,.762830E5,.619770E5/
C
```

```
C
      DATA TDFAB/.000000E5,.303730E5,.518570E5,.730880E5,.882930E5,
     1 .985240E5,.102516E6,.101977E6,.100423E6,.987290E5,.986660E5,
     2 .999500E5,.101827E6,.104014E6,.106268E6,.108802E6,.113076E6,
     3 .118262E6,.118770E6,.115389E6,.108843E6,.100893E6,.827850E5,
     4 .653280E5,.509650E5,.383290E5,.270710E5,.173420E5,.799200E4,
     5 -.249000E3,-.671899E4,-.124840E5,-.189830E5,-.246200E5,
     6 -.299630E5,-.349550E5,-.406320E5,-.426450E5,-.419750E5,
     7 -.412530E5,-.399940E5,-.380480E5,-.361740E5,-.347590E5,
     8 -.314130E5,-.272020E5,-.249430E5,-.229200E5,-.218610E5,
     9 -.186830E5/
C
C
      DATA TKA/-.100E-4,-.500E-4,-.100E-3,-.300E-3,-.500E-3,
     1 -.770E-3,-.121E-2,-.217E-2,-.246E-2,-.251E-2,-.111E-2,
     2 .290E-2,.178E-2,.328E-2,.477E-2,.627E-2,.814E-2,.815E-2,
     3 .729E-2,.743E-2,.763E-2,.664E-2,.319E-2,.130E-2,.147E-2,
     4 .231E-2,.284E-2,.312E-2,.299E-2,.286E-2,.273E-2,.260E-2,
     5 .246E-2,.233E-2,.219E-2,.205E-2,.178E-2,.150E-2,.137E-2,
     6 .125E-2,.112E-2,.100E-2,.970E-3,.950E-3,6*.900E-3/
      DATA TKB/.100E-4,.500E-4,.100E-3,.500E-3,.900E-3,.920E-3,
     1 .160E-2,.280E-2,.308E-2,.307E-2,.242E-2,.177E-2,.183E-2,
     2 .189E-2,.195E-2,.201E-2,.267E-2,.365E-2,.418E-2,.331E-2,
     3 .206E-2,.144E-2,.187E-2,.250E-2,.285E-2,.306E-2,.314E-2,
     4 .312E-2,.299E-2,.286E-2,.273E-2,.260E-2,.246E-2,.233E-2,
     5 .219E-2,.205E-2,.178E-2,.150E-2,.137E-2,.125E-2,.112E-2,
     6 .100E-2,.970E-3,.950E-3,6*.900E-3/
C
C
      DATA TK1/4*0.0,-.14004E0,-.22038E0,-.18579E0,-.19561E0,
     1 -.11937E0,-.39090E0,-.39980E-1,-.42710E-1,-.24110E-1,
     2 -.55200E-2,.42030E-1,.10260E0,.24528E0,-.21800E-1,.15058E0,
     3 .32297E0,.49535E0,.66774E0,.83920E0,.10106E1,.84838E0,
     4 .68609E0,.62891E0,.57173E0,.70384E0,.69398E0,.71203E0,
     5 .77934E0,.77717E0,.77500E0,.76750E0,.76000E0,.74299E0,
     6 .73860E0,12*.73860E0/
C
C
      DATA TK2/4*0.0,.58160E-1,.88780E-1,.73170E-1,.37260E-1,
     1 -.17880E-1,-.62020E-1,-.11063E0,-.24138E0,-.37350E0,-.50562E0,
     2 -.67265E0,-.82207E0,-.99600E0,-.70858E0,-.96743E0,-.12263E1,
     3 -.148512E1,-.174396E1,-.180346E1,-.186296E1,-.144259E1,
     4 -.102222E1,-.636110E0,-.250000E0,-.14000E0,21*0.0/
C
C
      DATA TFAORB/.00000E0,.92279E4,.14086E5,.18148E5,.21268E5,
     1 .24541E5,.24972E5,.24485E5,.24021E5,.24516E5,.25873E5,
     2 .27014E5,.28280E5,.29500E5,.30517E5,.31623E5,.34768E5,
     3 .34683E5,.33276E5,.31142E5,.28983E5,.24395E5,.20319E5,
     4 .20319E5,.17978E5,.16137E5,.14370E5,.12389E5,.10068E5,
     5 .80150E5,.63620E5,.50180E5,.35230E5,.22620E5,.10220E4,
     6 -.1690E3,-.2001E4,-.3201E4,-.3634E4,-.3435E4,-.3012E4,
     7 -.2758E4,-.2430E4,-.2034E4,-.1652E4,-.1390E4,-.1213E4,
     8 -.1054E4,-.9740E3,-.7330E3/
C
```

```
C
      DATA TQREF/.000E0,.890E2,.208E3,.308E3,.390E3,.457E3,.511E3,
     1 .556E3,.580E3,.592E3,.603E3,.614E3,.619E3,.622E3,.625E3,
     2 .629E3,.635E3,.642E3,.643E3,.645E3,.646E3,.648E3,.648E3,
     3 .649E3,.649E3,.649E3,.649E3,.644E3,.639E3,.633E3,.623E3,
     4 .623E3,.596E3,.577E3,.555E3,.530E3,.480E3,.430E3,.382E3,
     5 .339E3,.301E3,.264E3,.231E3,.201E3,.152E3,.114E3,.840E2,
     6 .580E2,.490E2,.330E2/
C
C
      DATA TALT2 / 96.E3,100.E3,105.E3,110.E3,115.E3,120.E3,125.E3,
     1 130.E3,135.E3,140.E3,145.E3,150.E3,155.E3,160.E3,165.E3,
     2 170.E3,175.E3,180.E3,185.E3,190.E3,195.E3,200.E3,205.E3,
     3 210.E3,215.E3,220.E3,225.E3,230.E3,235.E3,240.E3/
C
C
      DATA TQREF2 / .231E3,.201E3,.152E3,.114E3,.840E2,.580E2,
     1 .490E2,.440E2,.390E2,.330E2,.310E2,.290E2,.270E2,.250E2,
     2 .23E2,.20E2,.18E2,.15E2,.12E2,.11E2,.10E2,.9E1,.8E1,.7E1,
     3 .5E1,.4E1,.3E1,.2E1,.15E1,.1E1/
C
C
      DATA TDFNB2 / 2392.,2061.,1690.,1348.,992.,674.,419.,168.,
     1 -69.,-297.,-449.,-579.,-636.,-673.,-652.,-650.,-649.,-626.,
     2 -579.,-534.,-523.,-489.,-463.,-446.,-460.,5*-492./
C
C
      DATA TDFAB2 / 12029.,10623.,8703.,7135.,5589.,4186.,2961.,
     1 1879.,866.,-49.,-813.,-1505.,-2015.,-2436.,-2625.,-2731.,
     2 -2685.,-2441.,-2126.,-1783.,-1590.,-1350.,-1150.,-1000.,
     3 -940.,5*-900./
C
C
      DATA TDMMB2 / -259998.,-224221.,-182865.,-146624.,-106960.,
     1 -73303.,-44649.,-18269.,8290.,32311.,48200.,62987.,71038.,
     2 73111.,70875.,70679.,70222.,67988.,63584.,58081.,56525.,
     3 9*53081./
C
C
      DATA NALT, NALT2 / 50, 30 /
C
      DPL = 1.09 - PL
      DPLS = DPL*DPL
      BETAS = BETA*BETA
C
      ALTT = ALT
      IF( (ALT.LT.520.).OR.(ALT.GT.240000.) ) THEN
      CALL ZEROM( FPLUME, 3, 1 )
      CALL ZEROM( TPLUME, 3, 1 )
      CALL ZEROM( FPALP, 3, 1 )
      CALL ZEROM( FPBET, 3, 1 )
      CALL ZEROM( FPALT, 3, 1)
      CALL ZEROM( FPDPL, 3, 1 )
      CALL ZEROM( TPALP, 3, 1 )
      CALL ZEROM( TPBET, 3, 1 )
```

```
      CALL ZEROM( TPALT, 3, 1 )
      CALL ZEROM( TPDPL, 3, 1 )
      ELSE
C
      IF ( ISTAGE.EQ.1 ) THEN
C
      CALL INTRP1 ( ALTT, TALT, TQREF, NALT, QREF, PQRFH )
      CALL INTRP1 ( ALTT, TALT, TK1, NALT, XK1, PXK1H )
      CALL INTRP1 ( ALTT, TALT, TK2, NALT, XK2, PXK2H )
      CALL INTRP1 ( ALTT, TALT, TKA, NALT, XKA, PXKAH )
C     CALL INTRP1 ( ALTT, TALT, TKB, NALT, XKB, PXKBH )
      CALL INTRP1 ( ALTT, TALT, TDFNB, NALT, DFNB, PFNBH )
      CALL INTRP1 ( ALTT, TALT, TDFAB, NALT, DFAB, PDFABH )
C     CALL INTRP1 ( ALTT, TALT, TDMMB, NALT, DMMB, PMMBH )
      CALL INTRP1 ( ALTT, TALT, TFAORB, NALT, FAORB, PFAOH )
C
      RQ = Q/QREF
C
      ELSE
C
      CALL INTRP1 ( ALTT, TALT2, TQREF2, NALT2, QREF, PQRFH )
      CALL INTRP1 ( ALTT, TALT2, TDFNB2, NALT2, DFNB, PFNBH )
      CALL INTRP1 ( ALTT, TALT2, TDFAB2, NALT2, DFAB, PDFABH )
C     CALL INTRP1 ( ALTT, TALT2, TDMMB2, NALT2, DMMB, PMMBH )
C
      FAORB = 0.0
      XK1 = 0.0
      XK2 = 0.0
      XKA = 0.0
      XKB = 0.0
      PFAOH = 0.0
      PXK1H = 0.0
      PXK2H = 0.0
      PXKAH = 0.0
      PXKBH = 0.0
C
      RQ = 1.0
C
      END IF
C
      FDP = XK2*DPLS + XK1*DPL + 1.0
      FAB = XKB*BETAS + XKA*ALPHA + 1.0
      FABHDP = FDP*FAB*RQ
      PPALP = XKA*FDP*RQ
      PPBET = 2.0*XKB*BETA*FDP*RQ
      PFDPH = PXK2H*DPLS + PXK1H*DPL
      PFABH = PXKBH*BETAS + PXKAH*ALPHA
      PRQH = -Q*PQRFH/(QREF**2)
      PABHPH = PFDPH*FAB*RQ+FDP*PFABH*RQ+FDP*FAB*PRQH
      PPDPL = (2.0*XK2*DPL + XK1)*FAB*RQ
C
      FPLUME(1) = -( DFAB - FAORB + FAORB*FDP )*FAB*RQ
      FPLUME(2) = 0.0
      FPLUME(3) = -DFNB*FABHDP
      TPLUME(1) = 0.0
```

```
      TPLUME(2) = DMMB*FABHDP
      TPLUME(3) = 0.0
C
      FPALP(1) = -DFAB*PPALP
      FPALP(2) = 0.0
      FPALP(3) = -DFNB*PPALP
C
      FPBET(1) = -DFAB*PPBET
      FPBET(2) = 0.0
      FPBET(3) = -DFNB*PPBET
C
      TPALP(1) = 0.0
      TPALP(2) = DMMB*PPALP
      TPALP(3) = 0.0
C
      TPBET(1) = 0.0
      TPBET(2) = DMMB*PPBET
      TPBET(3) = 0.0
C
      FPALT(1) = -(PDFABH-PFAOH+PFAOH*FDP+FAORB*PFDPH)*FAB*RQ
     1           -(DFAB-FAORB+FAORB*FDP)*PFABH*RQ
     2           -(DFAB-FAORB+FAORB*FDP)*FAB*PRQH
      FPALT(2) = 0.0
      FPALT(3) = -PFNBH*FABHDP-DFNB*PABHPH
C
      TPALT(1) = 0.0
      TPALT(2) = PMMBH*FABHDP+DMMB*PABHPH
      TPALT(3) = 0.0
C
      FPDPL(1) = -DFAB*PPDPL
      FPDPL(2) = 0.0
      FPDPL(3) = -DFNB*PPDPL
C
      TPDPL(1) = 0.0
      TPDPL(2) = DMMB*PPDPL
      TPDPL(3) = 0.0
      END IF
C
        IF( IPLUME.GT.0 ) THEN
C
      OOTAUF = 1./TAUF
C
      F(44,44) = -OOTAUF
      F(45,45) = -OOTAUF
      F(46,46) = -OOTAUF
C
      S(44) = 2.*A(44)**2/TAUF
      S(45) = 2.*A(45)**2/TAUF
      S(46) = 2.*A(46)**2/TAUF
C
      XDOT(44) = -OOTAUF*X(44)
      XDOT(45) = -OOTAUF*X(45)
      XDOT(46) = -OOTAUF*X(46)
C
      FPLUME(1) = FPLUME(1) + X(44)
```

```
      FPLUME(2) = FPLUME(2) + X(45)
      FPLUME(3) = FPLUME(3) + X(46)
C
      ELSE
      END IF
C
      RETURN
C
901   FORMAT( 5X, 3E15.8 )
C
      END
```

```
      SUBROUTINE NMASS
C
C  ORBITER MASS PROPERTIES FOR MATED VEHICLE
C
      DIMENSION DELWT(27),TIX(27),TIY(27),TIZ(27),TIXY(27),
     1 TIXZ(27),TIYZ(27),TXCG(27),TYCG(27),TZCG(27)
     2 , RCGOFS(3)
      DIMENSION WEIGHT(21),DELTWT(21),AIXX(21),AIYY(21),
     1 AIZZ(21),AIXY(21),AIXZ(21),AIYZ(21)
     2 ,DELTW2(21),AXCG(21),AYCG(21),AZCG(21)
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / CONST / A(71), S(71), R(12)
      COMMON / IPROPR / XMASSI, XMASS2
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / DRVITV / XDOT(71), PD(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / PROPER / XMASS, OMASS, XIMTRX(3,3), XIMATI(3,3), RCG(3)
      COMMON / IPARAM / ISSME,ISRB,IAERO,IPLUME,IWIND,JACB,JRDR
      COMMON / STAGE / ISTAGE
C
      DATA DELWT/.0000000E0,.1240190E6,.2562150E6,.3902000E6,
     1 .5254230E6,.6595580E6,.7858090E6,.9053790E6,.1017844E7,
     2 .1126465E7,.1232233E7,.1336714E7,.1444331E7,.1554804E7,
     3 .1667689E7,.1781976E7,.1896941E7,.2008500E7,.2114813E7,
     4 .2116022E7,.2312411E7,.2401764E7,.2486219E7,.2554876E7,
     5 .2590745E7,.2612162E7,.3000000E7/
      DATA TIZ/.34571177E9,.33717742E9,.32783875E9,.31822454E9,
     1 .30834637E9,.29833384E9,.28865425E9,.27925903E9,.27026107E9,
     2 .26134089E9,.25247448E9,.24366268E9,.23451537E9,.22514319E9,
     3 .21554077E9,.20585032E9,.19625134E9,.18703819E9,.17834035E9,
     4 .17039781E9,.16305341E9,.15644651E9,.15081833E9,.14654725E9,
     5 .14444865E9,.14309413E9,.14252396E9/
      DATA TIY/.31533244E9,.30825736E9,.30048762E9,.29246577E9,
     1 .28419591E9,.27577624E9,.26757729E9,.25957609E9,.25190968E9,
     2 .24426837E9,.23664011E9,.22903716E9,.22111858E9,.21301242E9,
     3 .20470884E9,.19633535E9,.18805854E9,.18011987E9,.17262489E9,
     4 .16581405E9,.15953335E9,.15391125E9,.14917163E9,.14559313E9,
     5 .14375563E9,.14247118E9,.14190717E9/
      DATA TIX/.43565352E8,.42035072E8,.40384680E8,.38708230E8,
     1 .37013454E8,.35333184E8,.33766354E8,.32283937E8,.30866762E8,
     2 .29504164E8,.28182811E8,.26890217E8,.25573799E8,.24214993E8,
     3 .22813373E8,.21385839E8,.19943183E8,.18545048E8,.17219038E8,
     4 .15960817E8,.14768943E8,.13656280E8,.12645017E8,.11850270E8,
     5 .11541629E8,.11448761E8,.11434923E8/
      DATA TIXY/-.1109000E4,.20231000E5,.331730005,.44269000E5,
     1 .53301000E5,.60838000E5,.81050000E4,-.3177600E5,-.5693800E5,
     2 -.6097900E5,-.8345100E5,-.7652300E5,-.7268600E5,-.6996800E5,
     3 -.7753200E5,-.8290800E5,-.5791700E5,-.4125200E5,-.5736800E5,
     4 -.7099200E5,-.6496600E5,-.5387100E5,-.4373700E5,.86980000E4,
     5 .25832000E5,.34847000E5,.36875000E5/
      DATA TIXZ/-.6716835E7,-.6723358E7,-.6736725E7,-.6760808E7,
     1 -.6796678E7,-.6843763E7,-.6901428E7,-.6984498E7,-.7094909E7,
     2 -.7224324E7,-.7379288E7,-.7537877E7,-.7702677E7,-.7884695E7,
     3 -.8087421E7,-.8309431E7,-.8549404E7,-.8792555E7,-.9031727E7,
```

```
     4 -.9261782E7,-.9481698E7,-.9685493E7,-.9859969E7,-.9972303E7,
     5 -.9963524E7,-.9910083E7,-.9875095E7/
      DATA TIYZ/.3079300E5,.3209200E5,.3348800E5,.3479300E5,
     1 .3600700E5,.3702100E5,.3759700E5,.3643700E5,.3350300E5,
     2 .3193100E5,.3219600E5,.3257800E5,.3310900E5,.3368200E5,
     3 .3400900E5,.3449200E5,.3428300E5,.3443900E5,.3523600E5,
     4 .3595200E5,.3601600E5,.3560900E5,.3562300E5,.3178700E5,
     5 .2910300E5,.2790400E5,.2772200E5/
C
C
      DATA TXCG/-.36748E2,-.36721E2,-.3666E2,-.3655E2,
     1 -.36384999E2,-.36167000E2,-.35901999E2,-.35515999E2,
     2 -.34989999E2,-.34384999E2,-.33672999E2,-.32944999E2,
     3 -.321890E2,-.313530E2,-.303990E2,-.293420E2,-.28215000E2,
     4 -.270740E2,-.259490E2,-.248780E2,-.23844000E2,-.22889000E2,
     5 -.220640E2,-.215280E2,-.215690E2,-.2182700E2,-.21996000E2/
      DATA TYCG/.2167E-1,.2084E-1,.199E-1,.1887E-1,
     1 .177300E-1,.168900E-1,.173100E-1,.180600E-1,.172800E-1,
     2 .16400E-1,.151600E-1,.133700E-1,.108800E-1,.819999E-2,
     3 .578999E-2,.362000E-2,.201000E-2,.220999E-2,.256000E-2,
     4 .173000E-2,.192000E-2,.404999E-2,.546999E-2,.256300E-1,
     5 .350199E-1,.406599E-1,.415699E-1/
      DATA TZCG/-.15193E1,-.15623E1,-.16109E1,-.16634E1,
     1 -.1720000E1,-.1780200E1,-.1840900E1,-.1902100E1,-.1962900E1,
     2 -.2025899E1,-.2092199E1,-.2162099E1,-.2238999E1,-.2323899E1,
     3 -.2416299E1,-.2517400E1,-.2629299E1,-.2747899E1,-.2871000E1,
     4 -.2999799E1,-.3132899E1,-.3269099E1,-.3405500E1,-.3526500E1,
     5 -.3593100E1,-.3634099E1,-.3651100E1/
C
C     SECOND STAGE MASS PROPERTIES
C
      DATA WEIGHT/.1502336E7,.1467900E7,.1402928E7,.1337955E7,
     1 .1272982E7,.1208009E7,.1143036E7,.1078064E7,.1013091E7,
     2 .9481180E6,.8831450E6,.8181720E6,.7531990E6,.6882770E6,
     3 .6232540E6,.5582810E6,.4933080E6,.4325670E6,.3792600E6,
     4 .3380360E6,.3378410E6/
      DATA DELTWT/0.0,.344360E5,.994080E5,.164381E6,.229354E6,
     1 .294327E6,.359300E6,.424272E6,.489245E6,.554218E6,.619191E6,
     2 .684164E6,.749137E6,.814109E6,.879082E6,.944055E6,.1009028E7,
     3 .1069769E7,.1123076E7,.1164300E7,.1164495E7/
      DATA AIXX/.6385328E7,.6361136E7,.6312001E7,.6257807E7,
     1 .6198557E7,.6133434E7,.6060237E7,.5977493E7,.5885223E7,
     2 .5781466E7,.5659620E7,.5516285E7,.5352096E7,.5516285E7,
     3 .4916627E7,.4612541E7,.4255102E7,.3831761E7,.3322381E7,
     4 .2907861E7,.2906098E7/
      DATA AIYY/.112565130E9,.111091340E9,.108383980E9,.105701360E9,
     1 .102924850E9,.100047310E9,.971172750E8,.940918410E8,
     2 .909411020E8,.876503060E8,.841533930E8,.804180780E8,
     3 .765008070E8,.723613170E8,.675720440E8,.622070900E8,
     4 .566112810E8,.507625200E8,.444636860E8,.399201570E8,
     5 .399081290E8/
      DATA AIZZ/.108275840E9,.106826230E9,.104167990E9,
     1 .101539550E9,.988222770E8,.960098470E8,.931529870E8,
     2 .902102710E8,.871517780E8,.839647140E8,.805796140E8,
     3 .769975940E8,.732444780E8,.692959140E8,.647510930E8,
```

```
     4 .596902230E8,.544517600E8,.490261270E8,.432367370E8,
     5 .390926310E8,.390822210E8/
      DATA AIXY/.71683E5,.70499E5,.67073E5,.62141E5,.57624E5,
     1 .53543E5,.48724E5,.43058E5,.38199E5,.34207E5,.26927E5,
     2 .16781E5,.93040E4,.48090E4,-.11717E5,-.32714E5,-.36809E5,
     3 -.37000E5,-.56732E5,-.26755E5,-.26414E5/
      DATA AIXZ/-.12700109E8,-.12546131E8,-.12248261E8,-.11939382E8,
     1 -.11618950E8,-.11285010E8,-.10930293E8,-.10550928E8,
     2 -.10149881E8,-.97224580E7,-.92464110E7,-.87138290E7,
     3 -.81352500E7,-.74966560E7,-.67126220E7,-.57799210E7,
     4 -.47462130E7,-.35800620E7,-.22568430E7,-.14736020E7,
     5 -.14732390E7/
      DATA AIYZ/.20592E5,.20427E5,.19886E5,.19038E5,.18223E5,
     1 .17447E5,.16478E5,.15280E5,.14175E5,.13181E5,.11403E5,
     2 .88610E4,.67580E4,.52080E4,.52800E3,-.5696E4,-.7898E4,
     3 -.9114E4,-.16078E5,-.10170E5,-.10074E5/
C
      DATA DELTW2/0.0,.38834E5,.101257E6,.163669E6,.226072E6,
     1 .288467E6,.350857E6,.413243E6,.475626E6,.538008E6,
     2 .600388E6,.662769E6,.725150E6,.787532E6,.849916E6,
     3 .912299E6,.974685E6,.1036935E7,.1093997E6,.1144084E7,
     4 .1165805E7/
      DATA AXCG/-.10733E2,-.11539E2,-.12862E2,-.14228E2,-.15646E2,
     1 -.17124E2,-.18679E2,-.2033E2,-.22078E2,-.23941E2,
     2 -.25956E2,-.28201E2,-.3065E2,-.33356E2,-.36403E2,
     3 -.4017E2,-.4444E2,-.49406E2,-.55234E2,-.61756E2,
     4 -.63833E2/
      DATA AYCG/.563E-2,.572E-2,.598E-2,.637E-2,.675E-2,
     1 .712E-2,.754E-2,.810E-2,.862E-2,.910E-2,
     2 .969E-2,.1091E-1,.11964E0,.12804E0,.13523E0,
     3 .17088E0,.19165E0,.19156E0,.21512E0,.26286E0,
     4 .11459E0/
      DATA AZCG/-.45599E1,-.46809E1,-.489E1,-.51192E1,-.53698E1,
     1 -.56451E1,-.5951E1,-.62946E1,-.66777E1,-.71079E1,
     2 -.75988E1,-.81747E1,-.88353E1,-.96026E1,-.10511E2,
     3 -.11677E2,-.130649E2,-.147601E2,-.168204E2,-.192242E2,
     4 -.200905E2/
C
      DATA NNMSS, NMASS2 / 27, 21 /
C
      IF ( ISTAGE.EQ.1 ) THEN
C
      XMASS = (XMASSI - OMASS )/32.174
C
C     CALL INTRP1 ( OMASS, DELWT, TIX, NNMSS, XIX, SLOPE )
C     CALL INTRP1 ( OMASS, DELWT, TIY, NNMSS, YIY, SLOPE )
C     CALL INTRP1 ( OMASS, DELWT, TIZ, NNMSS, ZIZ, SLOPE )
C     CALL INTRP1 ( OMASS, DELWT, TIXY, NNMSS, PIXY, SLOPE )
C     CALL INTRP1 ( OMASS, DELWT, TIXZ, NNMSS, PIXZ, SLOPE )
C     CALL INTRP1 ( OMASS, DELWT, TIYZ, NNMSS, PIYZ, SLOPE )
      CALL INTRP1 ( OMASS, DELWT, TXCG, NNMSS, XCG, SLOPE )
      CALL INTRP1 ( OMASS, DELWT, TYCG, NNMSS, YCG, SLOPE )
      CALL INTRP1 ( OMASS, DELWT, TZCG, NNMSS, ZCG, SLOPE )
C
      ELSE
```

```
C
      XMASS = (XMASS2 - OMASS)/32.174
C
C     CALL INTRP1 ( OMASS, DELTWT, AIXX, NMASS2, XIX, SLOPE )
C     CALL INTRP1 ( OMASS, DELTWT, AIYY, NMASS2, YIY, SLOPE )
C     CALL INTRP1 ( OMASS, DELTWT, AIZZ, NMASS2, ZIZ, SLOPE )
C     CALL INTRP1 ( OMASS, DELTWT, AIXY, NMASS2, PIXY, SLOPE )
C     CALL INTRP1 ( OMASS, DELTWT, AIXZ, NMASS2, PIXZ, SLOPE )
C     CALL INTRP1 ( OMASS, DELTWT, AIYZ, NMASS2, PIYZ, SLOPE )
      CALL INTRP1 ( OMASS, DELTW2, AXCG, NMASS2, XCG, SLOPE )
      CALL INTRP1 ( OMASS, DELTW2, AYCG, NMASS2, YCG, SLOPE )
      CALL INTRP1 ( OMASS, DELTW2, AZCG, NMASS2, ZCG, SLOPE )
C
      END IF
C
C     XIMTRX(1,1) = XIX
C     XIMTRX(2,2) = YIY
C     XIMTRX(3,3) = ZIZ
C     XIMTRX(1,2) = -PIXY
C     XIMTRX(2,1) = XIMTRX(1,2)
C     XIMTRX(1,3) = -PIXZ
C     XIMTRX(3,1) = XIMTRX(1,3)
C     XIMTRX(2,3) = -PIYX
C     XIMTRX(3,2) = XIMTRX(2,3)
C
C     CALL INV3X3 ( XIMTRX, XIMATI )
C
      RCG(1) = XCG
      RCG(2) = YCG
      RCG(3) = ZCG
C
      RETURN
C
901   FORMAT( 5X, 3E15.8 )
C
        END
```

```
      SUBROUTINE AXCBIQ ( Q, V, P )
      DIMENSION Q(4), V(3), P(3,4)
C
      P(1,1) = Q(1)*V(1) - Q(4)*V(2) + Q(3)*V(3)
      P(2,1) = Q(4)*V(1) + Q(1)*V(2) - Q(2)*V(3)
      P(3,1) = -Q(3)*V(1) + Q(2)*V(2) + Q(1)*V(3)
C
      P(1,2) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
      P(2,2) = Q(3)*V(1) - Q(2)*V(2) - Q(1)*V(3)
      P(3,2) = Q(4)*V(1) + Q(1)*V(2) - Q(2)*V(3)
C
      P(1,3) = -Q(3)*V(1) + Q(2)*V(2) + Q(1)*V(3)
      P(2,3) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
      P(3,3) = -Q(1)*V(1) + Q(4)*V(2) - Q(3)*V(3)
C
      P(1,4) = -Q(4)*V(1) - Q(1)*V(2) + Q(2)*V(3)
      P(2,4) = Q(1)*V(1) - Q(4)*V(2) + Q(3)*V(3)
      P(3,4) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
C
      CALL SMLT( 2.0, P, P, 3, 4 )
C
      RETURN
      END
      SUBROUTINE AUXVAB
C
      DIMENSION VEC(3),VEC2(3),TMP(3,3)
C
      COMMON / VARIAB / VR(3), VW(3), VWGRAD(3)
      COMMON / TMAT / CBI(3,3), CIB(3,3), CLLB(3,3), CEFBRI(3,3)
      COMMON / PARTLA / PVMVB(3), PAVB(3), PBVB(3), PVMH, PAH, PBH
     1                , PVMVW(3), PAVW(3), PBVW(3)
C
      DATA CRAD / 57.295779 /
C
      VMAGS = VR(1)*VR(1) + VR(2)*VR(2) + VR(3)*VR(3)
      IF( VMAGS.LT.1.0 ) VMAGS = 1.0
      VMAG = SQRT( VMAGS )
C
      VMAGQ = VMAGS*VMAG
C
      DO 10 I = 1, 3
      PVMVB(I) = VR(I)/VMAG
10    CONTINUE
C
      VR13S = VMAGS - VR(2)*VR(2)
      IF( VR13S.LT.1.0 ) VR13S = 1.0
      VR13 = SQRT( VR13S )
C
      PAVB(1) = -VR(3)/VR13S
      PAVB(2) = 0.0
      PAVB(3) = VR(1)/VR13S
      CALL SMLT ( CRAD, PAVB, PAVB, 3, 1 )
C
      PBVB(1) = -VR(1)*VR(2)/(VMAGS*VR13)
      PBVB(2) = VR13/VMAGS
```

```
      PBVB(3) = -VR(2)*VR(3)/(VMAGS*VR13)
      CALL SMLT ( CRAD, PBVB, PBVB, 3, 1 )
C
      CALL MULT ( CLLB, VWGRAD, VEC, 3, 3, 1 )
      CALL SMLT ( -1.0, VEC, VEC, 3, 1 )
      CALL INNER ( PVMVB, VEC, PVMH, 3 )
      CALL INNER ( PAVB, VEC, PAH, 3 )
      CALL INNER ( PBVB, VEC, PBH, 3 )
C
      CALL TRANS ( CLLB, TMP, 3, 3 )
      CALL MULT ( TMP, VR, VEC, 3, 3, 1 )
      CALL SMLT ( -2.0, VEC, PVMVW, 3, 1 )
C
      DO 20 I = 1, 3
      VEC2(I) = VR(1)*CLLB(3,I) - VR(3)*CLLB(1,I)
20    CONTINUE
C
      A = -1.0/VR13S
      CALL SMLT ( A, VEC2, PAVW, 3, 1 )
      CALL SMLT ( CRAD, PAVW, PAVW, 3, 1 )
C
      DO 30 I = 1, 3
      VEC2(I) = VMAG*CLLB(2,I) - VR(2)*PVMVW(I)/(2.*VMAG)
30    CONTINUE
C
      A = -1.0/(VMAG*VR13)
      CALL SMLT ( A, VEC2, PBVW, 3, 1 )
      CALL SMLT ( CRAD, PBVW, PBVW, 3, 1 )
C
100   CONTINUE
C
      RETURN
      END
      SUBROUTINE AXCIBQ( Q, V, P )
C
      DIMENSION Q(4), V(3), P(3,4)
C
      P(1,1) = Q(1)*V(1) + Q(4)*V(2) - Q(3)*V(3)
      P(2,1) = -Q(4)*V(1) + Q(1)*V(2) + Q(2)*V(3)
      P(3,1) = Q(3)*V(1) - Q(2)*V(2) + Q(1)*V(3)
C
      P(1,2) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
      P(2,2) = Q(3)*V(1) - Q(2)*V(2) + Q(1)*V(3)
      P(3,2) = Q(4)*V(1) - Q(1)*V(2) - Q(2)*V(3)
C
      P(1,3) = -Q(3)*V(1) + Q(2)*V(2) - Q(1)*V(3)
      P(2,3) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
      P(3,3) = Q(1)*V(1) + Q(4)*V(2) - Q(3)*V(3)
C
      P(1,4) = -Q(4)*V(1) + Q(1)*V(2) + Q(2)*V(3)
      P(2,4) = -Q(1)*V(1) - Q(4)*V(2) + Q(3)*V(3)
      P(3,4) = Q(2)*V(1) + Q(3)*V(2) + Q(4)*V(3)
C
      CALL SMLT( 2.0, P, P, 3, 4 )
C
```

```
      RETURN
      END
      SUBROUTINE QMTRX( OMEGA, QDMTRX )
C
      DIMENSION OMEGA(3), QDMTRX(4,4)
C
      QDMTRX(1,1) = 0.0
      QDMTRX(1,2) = -0.5*OMEGA(1)
      QDMTRX(1,3) = -0.5*OMEGA(2)
      QDMTRX(1,4) = -0.5*OMEGA(3)
      QDMTRX(2,1) = 0.5*OMEGA(1)
      QDMTRX(2,2) = 0.0
      QDMTRX(2,3) = 0.5*OMEGA(3)
      QDMTRX(2,4) = -0.5*OMEGA(2)
      QDMTRX(3,1) = 0.5*OMEGA(2)
      QDMTRX(3,2) = -0.5*OMEGA(3)
      QDMTRX(3,3) = 0.0
      QDMTRX(3,4) = 0.5*OMEGA(1)
      QDMTRX(4,1) = 0.5*OMEGA(3)
      QDMTRX(4,2) = 0.5*OMEGA(2)
      QDMTRX(4,3) = -0.5*OMEGA(1)
      QDMTRX(4,4) = 0.0
C
      RETURN
      END
      SUBROUTINE PQOMEG( Q, P )
C
      DIMENSION Q(4), P(4,3)
C
      P(1,1) = -0.5*Q(2)
      P(1,2) = -0.5*Q(3)
      P(1,3) = -0.5*Q(4)
      P(2,1) = 0.5*Q(1)
      P(2,2) = -0.5*Q(4)
      P(2,3) = 0.5*Q(3)
      P(3,1) = 0.5*Q(4)
      P(3,2) = 0.5*Q(1)
      P(3,3) = -0.5*Q(2)
      P(4,1) = -0.5*Q(3)
      P(4,2) = 0.5*Q(2)
      P(4,3) = 0.5*Q(1)
C
      RETURN
      END
      SUBROUTINE AXVABQ
C
      DIMENSION VEC1(4), VEC2(4), VEC3(4)
C
      COMMON / STATES / RI(3), VB(3), Q(4), OMEGA(3)
      COMMON / VARIAB / VR(3), VW(3), VWGRAD(3)
      COMMON / PARTLB / PVRQ(3,4), PVMQ(4), PAQ(4), PBQ(4)
C
      DATA CRAD / 57.295779 /
C
      VMS = VR(1)**2 + VR(2)**2 + VR(3)**2
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      VM = SQRT(VMS)
      VR2VMS = VR(2)**2 + VMS
      VR1R3S = VR(1)**2 + VR(3)**2
      IF( VR1R3S.LT.1.0 ) VR1R3S = 1.0
      IF ( VR2VMS.LT.1.0 ) VR2VMS = 1.0
      IF ( VM.LT.1.0 ) VM = 1.0
C
      CALL AXCIBQ( Q, VW, PVRQ )
      CALL SMLT( -1.0, PVRQ, PVRQ, 3, 4 )
C
      DO 10 I = 1, 4
      VEC1(I) = PVRQ(1,I)
      VEC2(I) = PVRQ(2,I)
      VEC3(I) = PVRQ(3,I)
10    CONTINUE
C
      DO 20 I = 1, 4
      PVMQ(I) = VEC1(I)*VR(1) + VEC2(I)*VR(2) + VEC3(I)*VR(3)
      PVMQ(I) = 0.5*PVMQ(I)/VM
20    CONTINUE
C
      DO 30 I = 1, 4
      PAQ(I) = VEC3(I)*VR(1) - VEC1(I)*VR(3)
      PAQ(I) = CRAD*PAQ(I)/VR1R3S
30    CONTINUE
C
      DO 40 I = 1, 4
      PBQ(I) = VEC2(I)*VM - PVMQ(I)*VR(2)
      PBQ(I) = CRAD*PBQ(I)/VR2VMS
40    CONTINUE
C
      RETURN
      END
      SUBROUTINE CBIMXQ( Q, CBI )
C
      DIMENSION Q(4), CBI(3,3)
C
      CBI(1,1) = Q(1)*Q(1) + Q(2)*Q(2) - Q(3)*Q(3) - Q(4)*Q(4)
      CBI(2,1) = 2.*( Q(2)*Q(3) + Q(1)*Q(4) )
      CBI(3,1) = 2.*( Q(2)*Q(4) - Q(1)*Q(3) )
C
      CBI(1,2) = 2.*( Q(2)*Q(3) - Q(1)*Q(4) )
      CBI(2,2) = Q(1)*Q(1) - Q(2)*Q(2) + Q(3)*Q(3) - Q(4)*Q(4)
      CBI(3,2) = 2.*( Q(1)*Q(2) + Q(3)*Q(4) )
C
      CBI(1,3) = 2.*( Q(2)*Q(4) + Q(1)*Q(3) )
      CBI(2,3) = 2.*( Q(3)*Q(4) - Q(1)*Q(2) )
      CBI(3,3) = Q(1)*Q(1) - Q(2)*Q(2) - Q(3)*Q(3) + Q(4)*Q(4)
C
      RETURN
      END
      SUBROUTINE CBIMX( THT, CBI )
C
      DIMENSION THT(3), CBI(3,3)
C
```

```
      DATA CRAD / 57.295779 /
C
      SPHI = SIN( THT(1)/CRAD )
      CPHI = COS( THT(1)/CRAD )
      STHT = SIN( THT(2)/CRAD )
      CTHT = COS( THT(2)/CRAD )
      SPSI = SIN( THT(3)/CRAD )
      CPSI = COS( THT(3)/CRAD )
C
      CBI(1,1) = CTHT*CPSI
      CBI(2,1) = CTHT*SPSI
      CBI(3,1) = -STHT
      CBI(1,2) = SPHI*STHT*CPSI - CPHI*SPSI
      CBI(2,2) = SPHI*STHT*SPSI + CPHI*CPSI
      CBI(3,2) = SPHI*CTHT
      CBI(1,3) = CPHI*STHT*CPSI + SPHI*SPSI
      CBI(2,3) = CPHI*STHT*SPSI - SPHI*CPSI
      CBI(3,3) = CPHI*CTHT
C
      RETURN
      END
      SUBROUTINE E2QUAT( THT, Q )
C
C   SUBROUTINE TO CONVERT EULER ANGLES INTO QUATERNIONS REF: PERRY
C
      DIMENSION THT(3), AM(3,3), V(3), Q(4)
C
      DATA CRAD / 57.295779 /
C
      PHI = THT(1)
      THE = THT(2)
      PSI = THT(3)
C
      CPHI = COS( PHI/CRAD )
      SPHI = SIN( PHI/CRAD )
      CTHT = COS( THE/CRAD )
      STHT = SIN( THE/CRAD )
      CPSI = COS( PSI/CRAD )
      SPSI = SIN( PSI/CRAD )
C
      AM(1,1) = CPSI*CTHT
      AM(1,2) = -SPSI*CPHI + CPSI*STHT*SPHI
      AM(1,3) = SPSI*SPHI + CPSI*STHT*CPHI
C
      AM(2,1) = SPSI*CTHT
      AM(2,2) = CPSI*CPHI + SPSI*STHT*SPHI
      AM(2,3) = -CPSI*SPHI + SPSI*STHT*CPHI
C
      AM(3,1) = -STHT
      AM(3,2) = CTHT*SPHI
      AM(3,3) = CTHT*CPHI
C
      IF( (ABS(THE-90.).LT.1.).AND.(ABS(PHI-PSI).LT.1.) ) THEN
C
C     LOOP TO CONVERSION ( SEE PERRY )
```

```
C
      I = 0
C
      T = AM(1,1) + AM(2,2) + AM(3,3)
      S = T
C
      DO 10 II = 1, 3
      IF ( AM(II,II).GT.S ) THEN
      S = AM(II,II)
      I = II
      ELSE
      I = 0
      END IF
   10 CONTINUE
C
      T = SQRT( 1.0 + 2.0*S - T )
C
      K = 2
      J = 3
      N = 1
C
   20 CONTINUE
C
      IF ( ( I.EQ.0 ).OR.( N.EQ.I ) ) THEN
      S = (AM(J,K) - AM(K,J))/T
      V(N) = S
      ELSE
      V(J+K-I) = (AM(J,K) + AM(K,J))/T
      END IF
C
      IF ( N - 2 ) 30, 40, 50
   30 CONTINUE
      K = 3
      J = 1
      N = 2
      GO TO 20
   40 CONTINUE
      K = 1
      J = 2
      N = 3
      GO TO 20
   50 CONTINUE
C
      IF ( I.EQ.0 ) THEN
      S = T
      ELSE
      V(I) = T
      END IF
C
      IF ( S.GE.0.0 ) T = 0.5
      IF ( S.LT.0.0 ) T = -0.5
C
      S = T*S
      DO 60 II = 1, 3
      V(II) = T*V(II)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
   60 CONTINUE
      Q(1) = S
      Q(2) = V(1)
      Q(3) = V(2)
      Q(4) = V(3)
C
      ELSE
C
      Q(1) = SQRT( (AM(1,1)+AM(2,2)+AM(3,3)+1.0)/4.0 )
      Q(4) = (AM(2,1)-AM(1,2))/(4.*Q(1))
      Q(3) = (AM(1,3)-AM(3,1))/(4.*Q(1))
      Q(2) = (AM(3,2)-AM(2,3))/(4.*Q(1))
C
      END IF
C
      RETURN
      END
      SUBROUTINE QUAT2E( Q, THT )
C
      DIMENSION Q(4), THT(3)
C
      DATA CRAD / 57.295779 /
C
      TPSI = 2.*(Q(2)*Q(3)+Q(1)*Q(4))
      DPSI = Q(1)**2+Q(2)**2-Q(3)**2-Q(4)**2
      STHT = -2.*(Q(2)*Q(4)-Q(3)*Q(1))
      TPHI = 2.*(Q(3)*Q(4)+Q(1)*Q(2))
      DPHI = Q(1)**2+Q(4)**2-Q(2)**2-Q(3)**2
C
      THT(1) = CRAD*ATAN2( TPHI, DPHI )
      THT(2) = CRAD*ASIN( STHT )
      THT(3) = CRAD*ATAN2( TPSI, DPSI )
C
      RETURN
      END
      SUBROUTINE AUXRAT( D, O, A )
      DIMENSION D(3), O(3), A(3,3)
C
      A(1,1) = O(2)*D(2) + O(3)*D(3)
      A(2,1) = O(2)*D(1) - 2.0*O(1)*D(2)
      A(3,1) = O(3)*D(1) - 2.0*O(1)*D(3)
C
      A(1,2) = O(1)*D(2) - 2.0*O(2)*D(1)
      A(2,2) = O(1)*D(1) + O(3)*D(3)
      A(3,2) = O(3)*D(2) - 2.0*O(2)*D(3)
C
      A(1,3) = O(1)*D(3) - 2.0*O(3)*D(1)
      A(2,3) = O(2)*D(3) - 2.0*O(3)*D(2)
      A(3,3) = O(1)*D(1) + O(2)*D(2)
C
      RETURN
      END
      SUBROUTINE TMATY( YAW, TYAW )
C
      DIMENSION TYAW(3,3)
```

```
C
      DATA CRAD / 57.295779 /
C
      SY = SIN( YAW/CRAD )
      CY = COS( YAW/CRAD )
C
      TYAW(1,1) = CY
      TYAW(1,2) = -SY
      TYAW(1,3) = 0.0
      TYAW(2,1) = SY
      TYAW(2,2) = CY
      TYAW(2,3) = 0.0
      TYAW(3,1) = 0.0
      TYAW(3,2) = 0.0
      TYAW(3,3) = 1.0
C
      RETURN
      END
      SUBROUTINE TMATP( PITCH, TPITCH )
C
      DIMENSION TPITCH(3,3)
C
      DATA CRAD / 57.295779 /
C
      SP = SIN( PITCH/CRAD )
      CP = COS( PITCH/CRAD )
C
      TPITCH(1,1) = CP
      TPITCH(1,2) = 0.0
      TPITCH(1,3) = SP
      TPITCH(2,1) = 0.0
      TPITCH(2,2) = 1.0
      TPITCH(2,3) = 0.0
      TPITCH(3,1) = -SP
      TPITCH(3,2) = 0.0
      TPITCH(3,3) = CP
C
      RETURN
      END
```

```
      SUBROUTINE INTRP1 ( VM, TM, TC, NM, C, SLOPE )
      DIMENSION TM(1), TC(1)
C
      DO 10 I = 1, NM
      IF ( VM - TM(I) ) 20, 20, 10
   10 CONTINUE
      K = NM
   20 CONTINUE
      K = I - 1
      IF ( K.EQ.0 ) K = 1
      SLOPE = ( TC(K+1) - TC(K) )/(TM(K+1) - TM(K))
      C = TC(K) + SLOPE*(VM - TM(K))
      RETURN
      END
```

```
      SUBROUTINE CONTRL( TIME, PL )
C
      DIMENSION GIMYWL(3),GIMPTL(3),GIMYWS(2),GIMPTS(2)
C
      COMMON / NOZDAT / TTIME(500),TRLD(500),TTLD(500)
     1                          ,TRRD(500),TTRD(500)
     2                   ,TPL(500),TPL1(500),TYL1(500)
     3                          ,TPL2(500),TYL2(500)
     4                          ,TPL3(500),TYL3(500)
     5                  ,TRATER(500),TRATEP(500),TRATEY(500)
     6                  , TROLL(500), TPITCH(500), TYAW(500)
      COMMON / STRUCT / PTC1,YWC1,PTC2,YWC2,PTC3,YWC3
      COMMON / PONOZZ / PON(2)
      COMMON / STATES / R(3), VB(3), Q(4), OMEGA(3)
      COMMON / TMAT / CBI(3,3), CIB(3,3), CLLB(3,3), CEFBRI(3,3)
      COMMON / TYPE / ITYPE
      COMMON / GIMBAL / CCLB1(3,3),CCLB2(3,3),CCLB3(3,3),CCLBA(3,3)
     1                  ,CCLBB(3,3),PLN,RATEC(3),THTC(3)
C
      DATA GIMYWL,GIMPTL / 3*0.0, 3*0.0 /
      DATA GKPY, GKR / 0.1, 0.1 /
C
C
      CALL INTRP1 ( TIME, TTIME, TPL, 500, PL, SLOPE )
      PLN = PL
C
      CALL INTRP1 ( TIME, TTIME, TRATER, 500, RATEC(1), SLOPE )
      CALL INTRP1 ( TIME, TTIME, TRATEP, 500, RATEC(2), SLOPE )
      CALL INTRP1 ( TIME, TTIME, TRATEY, 500, RATEC(3), SLOPE )
C
      CALL INTRP1 ( TIME, TTIME, TROLL, 500, THTC(1), SLOPE )
      CALL INTRP1 ( TIME, TTIME, TPITCH, 500, THTC(2), SLOPE )
      CALL INTRP1 ( TIME, TTIME, TYAW, 500, THTC(3), SLOPE )
C
C
      CALL INTRP1 ( TIME, TTIME, TPL1, 500, GIMPTL(1), SLOPE )
      CALL INTRP1 ( TIME, TTIME, TYL1, 500, GIMYWL(1), SLOPE )
      CALL INTRP1 ( TIME, TTIME, TPL2, 500, GIMPTL(2), SLOPE )
      CALL INTRP1 ( TIME, TTIME, TYL2, 500, GIMYWL(2), SLOPE )
      CALL INTRP1 ( TIME, TTIME, TPL3, 500, GIMPTL(3), SLOPE )
      CALL INTRP1 ( TIME, TTIME, TYL3, 500, GIMYWL(3), SLOPE )
C
      CALL CCLBMX( 1,PTC1,YWC1,GIMYWL(1),GIMPTL(1),PL,CCLB1 )
      CALL CCLBMX( 2,PTC2,YWC2,GIMYWL(2),GIMPTL(2),PL,CCLB2 )
      CALL CCLBMX( 3,PTC3,YWC3,GIMYWL(3),GIMPTL(3),PL,CCLB3 )
C
      CALL INTRP1 ( TIME, TTIME, TRLD, 500, RLD, SLOPE )
      CALL INTRP1 ( TIME, TTIME, TTLD, 500, TLD, SLOPE )
      CALL INTRP1 ( TIME, TTIME, TRRD, 500, RRD, SLOPE )
      CALL INTRP1 ( TIME, TTIME, TTRD, 500, TRD, SLOPE )
C
      RL = .792*RLD + 0.5*( 1.0 - PON(1)/595. )
      TL = .792*TLD + 0.5*( 1.0 - PON(1)/595. )
      RR = .792*RRD + 0.5*( 1.0 - PON(2)/595. )
      TR = .792*TRD + 0.5*( 1.0 - PON(2)/595. )
```

```
C
      GIMYWS(1) = -.707*RL - .707*TL
      GIMPTS(1) = .707*RL - .707*TL
      GIMYWS(2) = .707*RR + .707*TR
      GIMPTS(2) = -.707*RR + .707*TR
C
      IF ( ITYPE ) 10, 10, 20
10    CONTINUE
C
      DO 15 I = 1, 2
      GIMYWS(I) = GIMYWS(I) + 0.5*GKPY*OMEGA(3)
      GIMPTS(I) = GIMPTS(I) + 0.5*GKPY*OMEGA(2)
15    CONTINUE
C
       GIMPTS(1) = GIMPTS(1) - 0.5*GKR*OMEGA(1)
       GIMPTS(2) = GIMPTS(2) + 0.5*GKR*OMEGA(1)
C
20    CONTINUE
C
C
      CALL CCLBMX( 4,0.0,0.0,GIMYWS(1),GIMPTS(1),0.0,CCLBA )
      CALL CCLBMX( 5,0.0,0.0,GIMYWS(2),GIMPTS(2),0.0,CCLBB )
C
      RETURN
      END
      SUBROUTINE CCLBMX( IENG,PTC,YWC,GIMYW,GIMPT,PL,TEMP )
C
      DIMENSION TY(3,3), TP(3,3), TEMP(3,3)
C
      DEF = -10.0
      IF(IENG.LT.2) DEF = -16.0
        IF(IENG.GT.3) DEF = 0.0
C
      YAW = YWC*PL + GIMYW
      PITCH = PTC*PL + DEF + GIMPT
      CALL TMATP ( PITCH, TP )
      CALL TMATY ( YAW, TY )
      CALL MULT ( TP, TY, TEMP, 3, 3, 3 )
C
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE UPDATE
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / NMEASR / NMEAS, IMEAS(5)
      COMMON / PREUP / XKM(71), PKM(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
C
      DO 1500 II = 1, NMEAS
C
      K = IMEAS(II)
      IF( K.EQ.0 ) GO TO 1000
C
      DO 10 I = 1, (NS+NPAR)
      XKM(I) = VAR(I+1)
10      CONTINUE
C
      L = 0
      DO 20 I = 1, (NS+NPAR)
      DO 20 J = I, (NS+NPAR)
      L = L + 1
      PKM(I,J) = VAR(NS+NPAR+L+1)
      PKM(J,I) = PKM(I,J)
20      CONTINUE
C
      GO TO ( 100, 200, 300, 400, 500 ) K
C
100      CONTINUE
C
      CALL ACCEL
      GO TO 1000
C
200      CONTINUE
C
C     CALL ATTIT
      GO TO 1000
C
300      CONTINUE
C
      CALL SSME
      GO TO 1000
C
400      CONTINUE
C
      CALL RADAR
      GO TO 1000
C
500      CONTINUE
C
      CALL SRB
C
1000     CONTINUE
C
      DO 1010 I = 1, (NS+NPAR)
      VAR(I+1) = XKM(I)
1010    CONTINUE
```

```
C
      L = 0
      DO 1020 I = 1, (NS+NPAR)
      DO 1020 J = I, (NS+NPAR)
      L = L + 1
      VAR(NS+NPAR+L+1) = PKM(I,J)
1020    CONTINUE
C
1500    CONTINUE
C
      RETURN
      END

      SUBROUTINE COV2UD ( U, N )
C
C     IMPLICIT DOUBLE PRECISION ( A-H, O-Z )
C
      DIMENSION U(1)
C
      Z = 0.E0
      ONE = 1.E0
      NONE = 1
C
      JJ = N*(N+1)/2
      NP2 = N + 2
      DO 50 L = 2, N
      J = NP2 - L
      ALPHA = Z
      IF ( U(JJ).GE.Z ) GO TO 10
C       WRITE(*, 100) J, U(JJ)
      U(JJ) = Z
   10   IF ( U(JJ).GT.Z ) ALPHA = ONE/U(JJ)
      JJ = JJ - J
      KK = 0
      KJ = JJ
      JM1 = J - 1
      DO 40 K = 1, JM1
        KJ = KJ + 1
        BETA = U(KJ)
        U(KJ) = ALPHA*U(KJ)
        IJ = JJ
        IK = KK
        DO 30 I = 1, K
          IK = IK + 1
          IJ = IJ + 1
   30       U(IK) = U(IK) - BETA*U(IJ)
   40     KK = KK + K
   50   CONTINUE
      IF ( U(1).GE.Z ) GO TO 60
C     WRITE(*, 100) NONE, U(1)
      U(1) = Z
   60 RETURN
C
  100 FORMAT ( 1H0, 20X,8H AT STEP,I4,16HDIAGONAL ENTRY =,F12.4 )
      END
```

```
      SUBROUTINE UDMEAS ( U, N, R, A, F, G, ALPHA )
C
C     IMPLICIT DOUBLE PRECISION ( A-H, O-Z )
C
      DIMENSION U(1), A(1), F(1), G(1)
C     DOUBLE PRECISION SUM, BETA, GAMMA
      LOGICAL IEST
C
      ZERO = 0.E0
      IEST = .FALSE.
      ONE = 1.E0
      NP1 = N + 1
      NP2 = N + 2
      NTOT = N*NP1/2
      IF ( ALPHA.LT.ZERO ) GO TO 3
      SUM = A(NP1)
      DO 1 J = 1, N
    1 SUM = SUM - A(J)*U(NTOT+J)
      U(NTOT+NP1) = SUM
      IEST = .TRUE.
C
    3 JJN = NTOT
      DO 10 L = 2, N
        J = NP2 - L
        JJ = JJN - J
        SUM = A(J)
        JM1 = J - 1
        DO 5 K = 1, JM1
    5      SUM = SUM + U(JJ+K)*A(K)
        F(J) = SUM
      G(J) = SUM*U(JJN)
   10 JJN = JJ
      F(1) = A(1)
      G(1) = U(1)*F(1)
C
      SUM = R + G(1)*F(1)
      GAMMA = 0
      IF ( SUM.GT.ZERO ) GAMMA = ONE/SUM
      IF ( F(1).NE.ZERO ) U(1) = U(1)*R*GAMMA
C
      KJ = 2
      DO 20 J = 2, N
        BETA = SUM
        TEMP = G(J)
        SUM = SUM + TEMP*F(J)
        P = -F(J)*GAMMA
        JM1 = J - 1
        DO 15 K = 1, JM1
           S = U(KJ)
           U(KJ) = S + P*G(K)
           G(K) = G(K) + TEMP*S
   15         KJ = KJ + 1
      IF ( TEMP.EQ.ZERO ) GO TO 20
        GAMMA = ONE/SUM
        U(KJ) = U(KJ)*BETA*GAMMA
```

```
   20 KJ = KJ + 1
      ALPHA = SUM
C
      IF ( .NOT.IEST ) RETURN
      F(NP1) = U(NTOT+NP1)*GAMMA
      DO 30 J = 1, N
   30    U(NTOT+J) = U(NTOT+J) + G(J)*F(NP1)
C
      RETURN
      END
      SUBROUTINE UD2COV ( UIN, POUT, N )
C
C     IMPLICIT DOUBLE PRECISION  ( A-H, O-Z )
C
      DIMENSION UIN(1), POUT(1)
C
      POUT(1) = UIN(1)
      JJ = 1
      DO 20 J = 2, N
      JJL = JJ
      JJ = JJ + J
      POUT(JJ) = UIN(JJ)
      S = POUT(JJ)
      II = 0
      JM1 = J - 1
      DO 20 I = 1, JM1
        II = II + I
        ALPHA = S*UIN(JJL+I)
        IK = II
        DO 10 K = I, JM1
          POUT(IK) = POUT(IK) + ALPHA*UIN(JJL+K)
   10       IK = IK + K
   20   POUT(JJL+I) = ALPHA
C
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE NOISE ( SIG, V )
C
      DATA II / 0 /
C
      SUM = 0.0
      DO 10 I = 1, 12
      RV = RANDOM(II)
      SUM = SUM + RV
   10 CONTINUE
      V = SIG*(SUM - 6.0)
C
      RETURN
      END
      REAL FUNCTION RANDOM(I)
      INTEGER I, J, K, M
      DATA J, K, M / 5243, 55397, 262139 /
C
      I = MOD( I*J + K, M )
C
      RANDOM = (REAL(I) + 0.5)/REAL(M)
C
      RETURN
      END
```

```
      SUBROUTINE ACCEL
C
      DIMENSION UNIT(3,3),CBS(3,3),DRS(3),ACCM(3),PAMQ(3,4)
     2          ,VEC1(3),VEC2(3),VEC3(3),TMP1(3,3),TMP2(3,3)
C     3          ,PAMO(3,3)
C
      COMMON DER(2700), VAR(2700), TMP(5400), NDER
      COMMON / TIMDAT / TMAX, HSTEP, TSAMP, TSTART, TSTOP, TRINT
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / PREUP / XKM(71), PKM(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / LINHMT / H(35,71), NMEAS
      COMMON / ACMEAS / ACM(3)
      COMMON / STATES / RI(3), VB(3), Q(4), OMEGA(3)
      COMMON / PROPER / XMASS, OMASS, XIMTRX(3,3), XIMATI(3,3), RCG(3)
      COMMON / MEPRTL / PFTMR(3),PFTPC(3),PFTGO(3),PFTGH(3)
      COMMON / RBPRTL / PFT(2),PFA(2),PFCS(2)
      COMMON / LAYOUT / RS(3),RA(3),RT1(3),RT2(3),RT3(3),RT4(3),RT5(3)
      COMMON / GIMBAL / CCLB1(3,3),CCLB2(3,3),CCLB3(3,3),CCLBA(3,3)
     1                 ,CCLBB(3,3),PLN,RATEC(3),THTC(3)
      COMMON / APARTL / PAMBRI(3,3),PAMBVB(3,3),ACC(3),THRUST(3)
      COMMON / PARARO / PVBDCF(3,3),PWDCF(3,3),PWDCM(3,3)
      COMMON / PARPLM / PVBDFP(3,3),PWDFP(3,3),PWDTP(3,3)
      COMMON / PARVWX / PVBDVW(3,3),PWDVW(3,3)
      COMMON / CONST / A(71), S(71), R(12)
      COMMON / TYPE / ITYPE
      COMMON / IPARAM / ISSME,ISRB,IAERO,IPLUME,IWIND,JACB,JRDR
      COMMON/UDWORK/RG(71),U(2556),PO(2556),SF(2556),SG(71),RESID,COVZ
      COMMON / QUPDWK / TVEC(71), VEC(71)
      COMMON / ACMEST / ACMHAT(3)
C
      DATA UNIT / 1.0, 3*0.0, 1.0, 3*0.0, 1.0 /
      DATA GF / 32.174 /
      DATA CRAD / 57.295779 /
C
      TIME = VAR(1)
      CALL ZEROM( H, 35, 71 )
      CALL ZEROM( TVEC, 71, 1 )
C
      NTS = NS + NPAR
C
      DO 990 IMEAS = 1, 3
C
      DO 10 I = 1, (NS+NPAR)
      X(I) = XKM(I)
10    CONTINUE
C
      DO 11 I = 1, 3
      VEC3(I) = XKM(I+49)
11    CONTINUE
C
      CALL XDVEC
      CALL ZDVEC
C
      CALL CBIMX ( THTC, CBS )
```

```
C
      IF ( TIME - TRINT ) 1020, 1010, 1010
C
1010  CONTINUE
C
      CALL CBIMXQ ( Q, CBS )
C
1020  CONTINUE
C
      CALL SUBT ( RS, RCG, DRS, 3, 1 )
C
      CALL CROSS ( OMEGA, DRS, VEC1 )
      CALL CROSS ( OMEGA, VEC1, VEC2 )
      CALL ADD ( ACC, VEC2, VEC1, 3, 1 )
      CALL MULT ( CBS, VEC1, ACCM, 3, 3, 1 )
C
      CALL ADD ( ACCM, VEC3, ACMHAT, 3, 1 )
      WRITE(*, 997) (ACMHAT(II), II = 1, 3)
      WRITE(*, 997) (ACM(II), II = 1, 3)
C
      IF ( ITYPE ) 13, 13, 20
13    CONTINUE
C
      DO 15 II = 1, 3
      SA = SQRT(R(1))
      CALL NOISE ( SA, VA )
      ACM(II) = ACMHAT(II) + VA
15    CONTINUE
C
      WRITE(2,997) (ACM(I), I=1,3)
      GO TO 1000
C
20    CONTINUE
C
      CALL LINFS
C
C       ACCEL WRT RI
C
      CALL MULT ( CBS, PAMBRI, TMP2, 3, 3, 3 )
      CALL IMBED ( H, 35, 71, TMP2, 3, 3, 1, 1 )
C
C       ACCEL WRT VB
C
      CALL MULT ( CBS, PAMBVB, TMP2, 3, 3, 3 )
      CALL IMBED ( H, 35, 71, TMP2, 3, 3, 1, 4 )
C
C       ACCEL WRT QUATERNIONS
C
      CALL AXCBIQ ( Q, VEC1, PAMQ )
C
      DO 25 I = 1, 3
      DO 25 J = 1, 3
      TMP1(I,J) = PAMQ(I,J)
25    CONTINUE
C
```

```
      IF ( TIME - TRINT ) 1040, 1030, 1030
C
1030  CONTINUE
C
      CALL IMBED ( H, 35, 71, TMP1, 3, 3, 1, 7 )
C
1040  CONTINUE
C
C       ACCEL WRT OMEGA
C
C     CALL SUBT ( RS, RCG, DRS, 3, 1 )
C     CALL AUXRAT ( DRS, OMEGA, PAMO )
C     CALL MULT ( CBS, PAMO, TMP2, 3, 3, 3 )
CRATE CALL IMBED ( H, 35, 71, TMP2, 3, 3, 1, 11 )
C
C       ACCEL WRT SSME1 PARAMETERS
C
      CALL MULT ( CBS, THRUST, VEC3, 3, 3, 1 )
C
      DO 30 I = 1, 3
      VEC1(I) = CCLB1(I,1)/XMASS
30    CONTINUE
      CALL MULT ( CBS, VEC1, VEC2, 3, 3, 1 )
      H( IMEAS, 14 ) = VEC3(IMEAS)/(GF*XMASS**2)
      H( IMEAS, 15 ) = VEC2(IMEAS)*PFTMR(1)
      H( IMEAS, 16 ) = VEC2(IMEAS)*PFTPC(1)
      H( IMEAS, 17 ) = VEC2(IMEAS)*PFTGO(1)
      H( IMEAS, 18 ) = VEC2(IMEAS)*PFTGH(1)
      H( IMEAS, 19 ) = 0.0
      H( IMEAS, 20 ) = 0.0
      H( IMEAS, 21 ) = 0.0
      H( IMEAS, 22 ) = 0.0
C
C       ACCEL WRT SSME2 PARAMETERS
C
      DO 50 I = 1, 3
      VEC1(I) = CCLB2(I,1)/XMASS
50    CONTINUE
      CALL MULT ( CBS, VEC1, VEC2, 3, 3, 1 )
      H( IMEAS, 23 ) = VEC3(IMEAS)/(GF*XMASS**2)
      H( IMEAS, 24 ) = VEC2(IMEAS)*PFTMR(2)
      H( IMEAS, 25 ) = VEC2(IMEAS)*PFTPC(2)
      H( IMEAS, 26 ) = VEC2(IMEAS)*PFTGO(2)
      H( IMEAS, 27 ) = VEC2(IMEAS)*PFTGH(2)
      H( IMEAS, 28 ) = 0.0
      H( IMEAS, 29 ) = 0.0
      H( IMEAS, 30 ) = 0.0
      H( IMEAS, 31 ) = 0.0
C
C       ACCEL WRT SSME3 PARAMETERS
C
      DO 70 I = 1, 3
      VEC1(I) = CCLB3(I,1)/XMASS
70    CONTINUE
      CALL MULT ( CBS, VEC1, VEC2, 3, 3, 1 )
```

```
      H( IMEAS, 32 ) = VEC3(IMEAS)/(GF*XMASS**2)
      H( IMEAS, 33 ) = VEC2(IMEAS)*PFTMR(3)
      H( IMEAS, 34 ) = VEC2(IMEAS)*PFTPC(3)
      H( IMEAS, 35 ) = VEC2(IMEAS)*PFTGO(3)
      H( IMEAS, 36 ) = VEC2(IMEAS)*PFTGH(3)
      H( IMEAS, 37 ) = 0.0
      H( IMEAS, 38 ) = 0.0
      H( IMEAS, 39 ) = 0.0
      H( IMEAS, 40 ) = 0.0
C
      KSTATE = 61
      IF ( ISRB.LT.1 ) GO TO 200
C
C       ACCEL WRT SRBA PARAMETERS
C
      DO 90 I = 1, 3
      VEC1(I) = CCLBA(I,1)/XMASS
90    CONTINUE
      CALL MULT ( CBS, VEC1, VEC2, 3, 3, 1 )
      H( IMEAS, KSTATE+1 ) = VEC3(IMEAS)/(GF*XMASS**2)
      H( IMEAS, KSTATE+2 ) = VEC2(IMEAS)*PFT(1)
      H( IMEAS, KSTATE+3 ) = VEC2(IMEAS)*PFA(1)
      H( IMEAS, KSTATE+4 ) = VEC2(IMEAS)*PFCS(1)
      H( IMEAS, KSTATE+5 ) = 0.0
C
C       ACCEL WRT SRBB PARAMETERS
C
      DO 110 I = 1, 3
      VEC1(I) = CCLBB(I,1)/XMASS
110   CONTINUE
      CALL MULT ( CBS, VEC1, VEC2, 3, 3, 1 )
      H( IMEAS, KSTATE+6 ) = VEC3(IMEAS)/(GF*XMASS**2)
      H( IMEAS, KSTATE+7 ) = VEC2(IMEAS)*PFT(2)
      H( IMEAS, KSTATE+8 ) = VEC2(IMEAS)*PFA(2)
      H( IMEAS, KSTATE+9 ) = VEC2(IMEAS)*PFCS(2)
      H( IMEAS, KSTATE+10 ) = 0.0
C
200     CONTINUE
C
      KSTATE = 40
      CALL LINFA
C
      IF( IAERO ) 300, 300, 210
210   CONTINUE
C
      CALL MULT ( CBS, PVBDCF, TMP1, 3, 3, 3)
      CALL IMBED ( H, 35, 71, TMP1, 3, 3, 1, (KSTATE+1) )
C
      KSTATE = KSTATE + 3
C
300   CONTINUE
C
      IF( IPLUME ) 400, 400, 310
310   CONTINUE
C
```

```
      CALL MULT ( CBS, PVBDFP, TMP1, 3, 3, 3 )
      CALL IMBED ( H, 35, 71, TMP1, 3, 3, 1, (KSTATE+1) )
C
      KSTATE = KSTATE + 3
C
400   CONTINUE
C
      IF( IWIND ) 600, 600, 510
510   CONTINUE
C
      CALL MULT ( CBS, PVBDVW, TMP1, 3, 3, 3 )
      CALL IMBED ( H, 35, 71, TMP1, 3, 3, 1, (KSTATE+1) )
C
      KSTATE = KSTATE + 3
C
600   CONTINUE
C
      IF ( JACB ) 700, 700, 610
610   CONTINUE
C
      CALL IMBED ( H, 35, 71, UNIT, 3, 3, 1, (KSTATE+1) )
C
700   CONTINUE
C
      IF ( ACM(IMEAS).EQ.0.0 ) THEN
      RESID = 0.0
      COVZ = 0.0
      CALL ZEROM ( RG, (NS+NPAR), 1 )
      ELSE
C
C     UDU**T FACTORED COVARIANCE UPDATE
C
        KJ = 0
        DO 940 J = 1, (NS+NPAR)
        RG(J) = H(IMEAS,J)
        DO 940 K = 1, J
        KJ = KJ + 1
        U(KJ) = PKM(K,J)
940     CONTINUE
C
        CALL COV2UD( U, NTS )
C
        RACC = R(1)
        ALPHA = -1.0
        CALL UDMEAS( U, NTS, RACC, RG, SF, SG, ALPHA )
C
        CALL UD2COV( U, PO, NTS )
C
        RESID = ACM(IMEAS) - ACMHAT(IMEAS)
        COVZ = ALPHA
C       IF ( ABS(RESID).GT.(6.*SQRT(COVZ)) ) GO TO 980
C
        DO 950 J = 1, (NS+NPAR)
        XKM(J) = X(J) + (SG(J)/ALPHA)*RESID
950     CONTINUE
```

```
C
        IJ = 0
        DO 960 J = 1, (NS+NPAR)
        DO 960 I = 1, J
        IJ = IJ + 1
        PKM(I,J) = PO(IJ)
        PKM(J,I) = PKM(I,J)
960     CONTINUE
C
      IF ( TIME - TRINT ) 1060, 1050, 1050
C
1050  CONTINUE
C
        DQ1 = XKM(7) - X(7)
        DQ2 = XKM(8) - X(8)
        DQ3 = XKM(9) - X(9)
        DQS = XKM(7)**2 + XKM(8)**2 + XKM(9)**2
        IF ( DQS.LT.1.0 ) THEN
        DQ4 = -(XKM(7)*DQ1+XKM(8)*DQ2+XKM(9)*DQ3)/SQRT( 1.0 - DQS )
        XKM(10) = X(10) + DQ4
        END IF
        IF( ABS((DQS+XKM(10)**2)-1.0).GT.1.E-4 ) THEN
        XKM(7) = X(7)
        XKM(8) = X(8)
        XKM(9) = X(9)
        XKM(10) = X(10)
        ELSE
        END IF
        TVEC(7) = -XKM(7)/X(10)
        TVEC(8) = -XKM(8)/X(10)
        TVEC(9) = -XKM(9)/X(10)
        CALL MULT ( TVEC, PKM, VEC, 1, NTS, NTS )
        DO 970 I = 1, (NS+NPAR)
        PKM(10,I) = VEC(I)
        PKM(I,10) = VEC(I)
970     CONTINUE
        CALL INNER ( VEC, TVEC, TEMP, NTS )
        PKM(10,10) = TEMP
C
1060    CONTINUE
C
980     CONTINUE
C
      END IF
C
        CALL OUTPUT
C
990     CONTINUE
C
1000    CONTINUE
C
        RETURN
C
995     FORMAT( 3X, 5E15.8 )
998     FORMAT( 7X, 4E15.8 )
```

```
997     FORMAT( 5X, 3E15.8 )
C
      END
      SUBROUTINE SSME
C
      COMMON DER(2700), VAR(2700), PMET(5400), NDER
      COMMON / TIMDAT / TMAX, HSTEP, TSAMP, TSTART, TSTOP, TRINT
      COMMON / MEMEAS / YMSME(6,3)
      COMMON / CONST / A(71),S(71),R(12)
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / PREUP / XKM(71),PKM(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / LINHMT / H(35,71), NMEAS
      COMMON / SMEMEA / XLH(6,3)
      COMMON / TYPE / ITYPE
      COMMON/UDWORK/RG(71),U(2556),PO(2556),SF(2556),SG(71),RESID,COVZ
      COMMON / QUPDWK / TVEC(71), VEC(71)
C
      TIME = VAR(1)
C
      NTS = NS + NPAR
C
      KMEAS = 6
C
      DO 200 IENG = 1, 3
      DO 200 JMEAS = 1, 6
C
      DO 10 I = 1, (NS+NPAR)
      X(I) = XKM(I)
10    CONTINUE
C
      CALL NASSME
C
      IF ( ITYPE ) 13, 13, 20
13    CONTINUE
C
      DO 15 II = 1, 3
      DO 15 JJ = 1, 6
      SME = SQRT(R(2+JJ))
      CALL NOISE ( SME, VSME )
      YMSME(JJ,II) = XLH(JJ,II) + VSME
15    CONTINUE
C
      DO 17 II = 1 ,3
      WRITE(2,997) (YMSME(JJ,II), JJ=1,6)
17    CONTINUE
      GO TO 300
C
20    CONTINUE
C
      KMEAS = KMEAS + 1
C
C     UDU*^T FACTORED COVARIANCE UPDATE
C
        KJ = 0
```

```
      DO 140 J = 1, (NS+NPAR)
      RG(J) = H(KMEAS,J)
      DO 140 K = 1, J
      KJ = KJ + 1
      U(KJ) = PKM(K,J)
140   CONTINUE
C
      CALL COV2UD( U, NTS )
C
      RSSME = R(2+JMEAS)
      ALPHA = -1.0
      CALL UDMEAS( U, NTS, RSSME, RG, SF, SG, ALPHA )
C
      CALL UD2COV( U, PO, NTS )
C
      RESID = YMSME( JMEAS, IENG ) - XLH( JMEAS, IENG )
      COVZ = ALPHA
C     IF ( ABS(RESID).GT.(6.*SQRT(COVZ)) ) GO TO 180
C
      DO 150 J = 1, (NS+NPAR)
      XKM(J) = X(J) + (SG(J)/ALPHA)*RESID
150   CONTINUE
C
      IJ = 0
      DO 160 J = 1, (NS+NPAR)
      DO 160 I = 1, J
      IJ = IJ + 1
      PKM(I,J) = PO(IJ)
      PKM(J,I) = PKM(I,J)
160   CONTINUE
C
      IF ( TIME - TRINT ) 320, 310, 310
C
310   CONTINUE
C
      DQ1 = XKM(7) - X(7)
      DQ2 = XKM(8) - X(8)
      DQ3 = XKM(9) - X(9)
      DQS = XKM(7)**2 + XKM(8)**2 + XKM(9)**2
      IF ( DQS.LT.1.0 ) THEN
      DQ4 = -(XKM(7)*DQ1+XKM(8)*DQ2+XKM(9)*DQ3)/SQRT( 1.0 - DQS )
      XKM(10) = X(10) + DQ4
      END IF
      IF( ABS((DQS+XKM(10)**2)-1.0).GT.1.E-4 ) THEN
      XKM(7) = X(7)
      XKM(8) = X(8)
      XKM(9) = X(9)
      XKM(10) = X(10)
      ELSE
      END IF
      TVEC(7) = -XKM(7)/X(10)
      TVEC(8) = -XKM(8)/X(10)
      TVEC(9) = -XKM(9)/X(10)
      CALL MULT ( TVEC, PKM, VEC, 1, NTS, NTS )
      DO 170 I = 1, (NS+NPAR)
```

```
        PKM(10,I) = VEC(I)
        PKM(I,10) = VEC(I)
170     CONTINUE
        CALL INNER ( VEC, TVEC, TEMP, NTS )
        PKM(10,10) = TEMP
C
320     CONTINUE
C
180     CONTINUE
C
        CALL OUTPUT
C
200     CONTINUE
C
300     CONTINUE
C
        RETURN
C
991     FORMAT( 2X, 2I5 )
995     FORMAT( 3X, 5E15.8 )
997     FORMAT( 5X, 6E15.8 )
998     FORMAT( 7X, 4E15.8 )
C
      END
      SUBROUTINE SRB
C
      COMMON DER(2700), VAR(2700), PMET(5400), NDER
      COMMON / TIMDAT / TMAX, HSTEP, TSAMP, TSTART, TSTOP, TRINT
      COMMON / GMDATS / GIMYW(2), GIMPT(2)
      COMMON / SRMEAS / YMSRB(2)
      COMMON / SRBMEA / POHHAT(2)
      COMMON / PROPAG / X(71), P(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / LINHMT / H(35,71), NMEAS
      COMMON / CONST / A(71), S(71), R(12)
      COMMON / PREUP / XKM(71), PKM(71,71)
      COMMON / TYPE / ITYPE
      COMMON/UDWORK/RG(71),U(2556),PO(2556),SF(2556),SG(71),RESID,COVZ
      COMMON / QUPDWK / TVEC(71), VEC(71)
C
      TIME = VAR(1)
C
      NTS = NS + NPAR
C
      KMEAS = 33
C
      DO 200 IMTR = 1, 2
C
      DO 10 I = 1, (NS+NPAR)
      X(I) = XKM(I)
10    CONTINUE
C
      CALL NASSRB
      WRITE(*, 997) POHHAT(IMTR)
      WRITE(*, 997) YMSRB(IMTR)
```

```
C
      IF ( ITYPE ) 13, 13, 20
13    CONTINUE
C
      DO 15 II = 1, 2
      SSRB = SQRT(R(12))
      CALL NOISE ( SSRB, VSRB )
      YMSRB(II) = POHHAT(II) + VSRB
C
      WRITE(2,997) YMSRB(II)
15    CONTINUE
      GO TO 300
C
20      CONTINUE
C
      KMEAS = KMEAS + 1
C
C     UDU**T FACTORED COVARIANCE UPDATE
C
        KJ = 0
        DO 140 J = 1, (NS+NPAR)
        RG(J) = H(KMEAS, J)
        DO 140 K = 1, J
        KJ = KJ + 1
        U(KJ) = PKM(K,J)
140     CONTINUE
C
        CALL COV2UD( U, NTS )
C
        RPOH = R(12)
        ALPHA = -1.0
        CALL UDMEAS( U, NTS, RPOH, RG, SF, SG, ALPHA )
C
        CALL UD2COV( U, PO, NTS )
C
        RESID = YMSRB(IMTR) - POHHAT(IMTR)
        COVZ = ALPHA
C
        DO 150 J = 1, (NS+NPAR)
        XKM(J) = X(J) + (SG(J)/ALPHA)*RESID
150     CONTINUE
C
        IJ = 0
        DO 160 J = 1, (NS+NPAR)
        DO 160 I = 1, J
        IJ = IJ + 1
        PKM(I,J) = PO(IJ)
        PKM(J,I) = PKM(I,J)
160     CONTINUE
C
        IF ( TIME - TRINT ) 320, 310, 310
C
310     CONTINUE
C
        DQ1 = XKM(7) - X(7)
```

```
      DQ2 = XKM(8) - X(8)
      DQ3 = XKM(9) - X(9)
      DQS = XKM(7)**2 + XKM(8)**2 + XKM(9)**2
      IF ( DQS.LT.1.0 ) THEN
      DQ4 = -(XKM(7)*DQ1+XKM(8)*DQ2+XKM(9)*DQ3)/SQRT( 1.0 - DQS )
      XKM(10) = X(10) + DQ4
      END IF
      IF( ABS((DQS+XKM(10)**2)-1.0).GT.1.E-4 ) THEN
      XKM(7) = X(7)
      XKM(8) = X(8)
      XKM(9) = X(9)
      XKM(10) = X(10)
      ELSE
      END IF
      TVEC(7) = -XKM(7)/X(10)
      TVEC(8) = -XKM(8)/X(10)
      TVEC(9) = -XKM(9)/X(10)
      CALL MULT ( TVEC, PKM, VEC, 1, NTS, NTS )
      DO 170 I = 1, (NS+NPAR)
      PKM(10,I) = VEC(I)
      PKM(I,10) = VEC(I)
170   CONTINUE
      CALL INNER ( VEC, TVEC, TEMP, NTS )
      PKM(10,10) = TEMP
C
320   CONTINUE
C
180   CONTINUE
C
      CALL OUTPUT
C
200   CONTINUE
C
300   CONTINUE
C
      RETURN
C
995   FORMAT( 3X, 5E15.8 )
997   FORMAT( 5X, 3E15.8 )
998   FORMAT( 7X, 4E15.8 )
C
     END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE ADD ( A, B, C, NRA, NCA )
      DIMENSION A(NRA,NCA), B(NRA,NCA), C(NRA,NCA)
C
      DO 10 J = 1, NCA
      DO 10 I = 1, NRA
      C(I,J) = A(I,J) + B(I,J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE SUBT ( A, B, C, NRA, NCA )
      DIMENSION A(NRA,NCA), B(NRA,NCA), C(NRA,NCA)
C
      DO 10 J = 1, NCA
      DO 10 I = 1, NRA
      C(I,J) = A(I,J) - B(I,J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE MULT ( A, B, C, NRA, NCA, NCB )
      DIMENSION A(NRA,NCA), B(NCA, NCB), C(NRA,NCB)
C
      DO 20 J = 1, NCB
      DO 20 I = 1, NRA
      K = 0
      TEMP = 0.0
      DO 10 L = 1, NCA
      K = K + 1
      TEMP = TEMP + A(I,K)*B(K,J)
   10 CONTINUE
      C(I,J) = TEMP
   20 CONTINUE
C
      RETURN
      END
      SUBROUTINE TRANS ( A, B, NRA, NCA )
      DIMENSION A(NRA,NCA), B(NCA,NRA)
C
      DO 10 I = 1, NRA
      DO 10 J = 1, NCA
      B(J,I) = A(I,J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE SMLT ( C, A, B, NRA, NCA )
      DIMENSION A(NRA,NCA), B(NRA,NCA)
C
      DO 10 I = 1, NRA
      DO 10 J = 1, NCA
      B(I,J) = C*A(I,J)
   10 CONTINUE
C
      RETURN
```

```
      END
      SUBROUTINE SWITCH ( A, B, NRA, NCA )
      DIMENSION A(NRA,NCA), B(NRA,NCA)
C
      DO 10 I = 1, NRA
      DO 10 J = 1, NCA
      B(I,J) = A(I,J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE SYMTRK ( A, NS, AS )
      DIMENSION A(NS,NS), AS(NS,NS)
C
      DO 10 I = 1, NS
      DO 10 J = I, NS
      AS(I,J) = 0.5*(A(I,J) + A(J,I))
      AS(J,I) = AS(I,J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE OUTER ( A, B, C, NRA, NCB )
      DIMENSION A(NRA), B(NCB), C(NRA,NCB)
C
      DO 10 I = 1, NRA
      DO 10 J = 1, NCB
      C(I,J) = A(I)*B(J)
   10 CONTINUE
C
      RETURN
      END
      SUBROUTINE INNER ( A, B, C, NELEM )
      DIMENSION A(NELEM), B(NELEM)
C
      TEMP = 0.0
C
      DO 10 I = 1, NELEM
      TEMP = TEMP + A(I)*B(I)
   10 CONTINUE
C
      C = TEMP
C
      RETURN
      END
      SUBROUTINE SKEW ( A, B )
      DIMENSION A(3), B(3,3)
C
      B(1,1) = 0.0
      B(1,2) = -A(3)
      B(1,3) = A(2)
      B(2,1) = -B(1,2)
      B(2,2) = 0.0
      B(2,3) = -A(1)
      B(3,1) = -B(1,3)
```

```
      B(3,2) = -B(2,3)
      B(3,3) = 0.0
C
      RETURN
      END
      SUBROUTINE IMBED ( BIG, IM, IN, SUB, M, N, IROW, ICOL )
      DIMENSION SUB(1), BIG(1)
C
      IF ( (M+IROW-1 .LE. IM) .AND. (N+ICOL-1 .LE. IN) ) GOTO 5
      WRITE(6,6000) IM, IN, M, N, IROW, ICOL
      STOP
C
    5 IB = ( ICOL - 1 )*IM + ( IROW - 1 )
      IS = 0
      IOFFST = IM - M
C
      DO 20 I = 1, N
      DO 10 J = 1, M
      IB = IB + 1
      IS = IS + 1
      BIG(IB) = SUB(IS)
   10 CONTINUE
      IB = IB + IOFFST
   20 CONTINUE
C
      RETURN
C
 6000 FORMAT ( 5X, 6I10 )
C
      END
      SUBROUTINE INV2X2 ( A, AI )
      DIMENSION A(2,2), AI(2,2)
C
      DET = A(1,1)*A(2,2) - A(1,2)*A(2,1)
C
      AI(1,1) = A(2,2)/DET
      AI(1,2) = -A(1,2)/DET
      AI(2,1) = -A(2,1)/DET
      AI(2,2) = A(1,1)/DET
C
      RETURN
      END
      SUBROUTINE INVNXN ( A, N, D, L, M )
      DIMENSION A(1), L(1), M(1)
C
      D = 1.0
      NK = -N
      DO 80 K = 1, N
      NK = NK + N
      L(K) = K
      M(K) = K
      KK = NK + K
      BIGA = A(KK)
      DO 20 J = K, N
      IZ = N*(J-1)
```

```
      DO 20 I = K, N
      IJ = IZ + I
   10 IF ( ABS(BIGA) - ABS(A(IJ)) ) 15, 20, 20
   15 BIGA = A(IJ)
      L(K) = I
      M(K) = J
   20 CONTINUE
C
      J = L(K)
      IF ( J - K ) 35, 35, 25
   25 KI = K - N
      DO 30 I = 1, N
      KI = KI + N
      HOLD = -A(KI)
      JI = KI - K + J
      A(KI) = A(JI)
   30 A(JI) = HOLD
C
   35 I = M(K)
      IF ( I - K ) 45, 45, 38
   38 JP = N*(I - 1)
      DO 40 J = 1, N
      JK = NK + J
      JI = JP + J
      HOLD = -A(JK)
      A(JK) = A(JI)
   40 A(JI) = HOLD
C
   45 IF ( BIGA ) 48, 46, 48
   46 D = 0.0
      RETURN
   48 DO 55 I = 1, N
      IF ( I - K ) 50, 55, 50
   50 IK = NK + I
      A(IK) = A(IK)/(-BIGA)
   55 CONTINUE
C
      DO 65 I = 1, N
      IK = NK + I
      HOLD = A(IK)
      IJ = I - N
      DO 65 J = 1, N
      IJ = IJ + N
      IF ( I - K ) 60, 65, 60
   60 IF ( J - K ) 62, 65, 62
   62 KJ = IJ - I + K
      A(IJ) = HOLD*A(KJ) + A(IJ)
   65 CONTINUE
C
      KJ = K - N
      DO 75 J = 1, N
      KJ = KJ + N
      IF ( J - K ) 70, 75, 70
   70 A(KJ) = A(KJ)/BIGA
   75 CONTINUE
```

```
C
      D = D*BIGA
C
      A(KK) = 1.0/BIGA
   80 CONTINUE
C
      K = N
  100 K = K - 1
      IF ( K ) 150, 150, 105
  105 I = L(K)
      IF ( I - K ) 120, 120, 108
  108 JQ = N*(K - 1)
      JR = N*(I - 1)
      DO 110 J = 1, N
      JK = JQ + J
      HOLD = A(JK)
      JI = JR + J
      A(JK) = -A(JI)
  110 A(JI) = HOLD
  120 J = M(K)
      IF ( J - K ) 100, 100, 125
  125 KI = K - N
      DO 130 I = 1, N
      KI = KI + N
      HOLD = A(KI)
      JI = KI - K + J
      A(KI) = -A(JI)
  130 A(JI) = HOLD
      GO TO 100
  150 RETURN
      END
      SUBROUTINE INV3X3( A, AINV )
      DIMENSION A(3,3), AINV(3,3), B(3,3)
C
      DET = A(1,1)*A(2,2)*A(3,3) + A(1,2)*A(2,3)*A(3,1)
     1    + A(1,3)*A(2,1)*A(3,2) - A(1,3)*A(2,2)*A(3,1)
     2    - A(1,1)*A(2,3)*A(3,2) - A(1,2)*A(2,1)*A(3,3)
C
      B(1,1) = A(2,2)*A(3,3) - A(2,3)*A(3,2)
      B(1,2) = A(1,3)*A(3,2) - A(1,2)*A(3,3)
      B(1,3) = A(1,2)*A(2,3) - A(2,2)*A(1,3)
      B(2,1) = A(3,1)*A(2,3) - A(2,1)*A(3,3)
      B(2,2) = A(1,1)*A(3,3) - A(3,1)*A(1,3)
      B(2,3) = A(2,1)*A(1,3) - A(1,1)*A(2,3)
      B(3,1) = A(2,1)*A(3,2) - A(3,1)*A(2,2)
      B(3,2) = A(3,1)*A(1,2) - A(1,1)*A(3,2)
      B(3,3) = A(1,1)*A(2,2) - A(1,2)*A(2,1)
C
      DET = 1.0/DET
C
      DO 100 I = 1, 3
      DO 100 J = 1, 3
      AINV(I,J) = DET*B(I,J)
100   CONTINUE
C
```

```
      RETURN
      END
      SUBROUTINE ZEROM( A, NROW, NCOL )
      DIMENSION A(NROW,NCOL)
C
      DO 10 I = 1, NROW
      DO 10 J = 1, NCOL
      A(I,J) = 0.0
10    CONTINUE
C
      RETURN
      END
      SUBROUTINE CROSS( A, B, C )
      DIMENSION A(3), B(3), C(3)
C
      C(1) = A(2)*B(3) - A(3)*B(2)
      C(2) = A(3)*B(1) - A(1)*B(3)
      C(3) = A(1)*B(2) - A(2)*B(1)
C
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE OUTPUT
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / PREUP / XKM(71), PKM(71,71)
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON/UDWORK/RG(71),U(2556),PO(2556),SF(2556),SG(71),RESID,COVZ
      COMMON / LINHMT / H(35,71), NMEAS
      COMMON / SRBDAT / TVACS(2),XISPS(2),XMD(2),AEXIT(2)
     1                  ,CVPOH(2),CVTVCS(2),CVISPS(2),CVDM(2),CVAE(2)
     2                  ,OMASS(2)
      COMMON / SRBMEA / POHHAT(2)
      COMMON / SMEDAT / TVACL(3),XISPL(3),WDO2H(3),WDH2H(3),WD(3)
     1                  ,CVTVCL(3),CVISPL(3),CVWDO2(3),CVWDH2(3)
     2                  ,CVWD(3),CVWDGO(3),CVWDGH(3),PCTAG(3),OMASE(3)
      COMMON / SMEMEA / XLH(6,3)
      COMMON / GIMBAL / CCLB1(3,3),CCLB2(3,3),CCLB3(3,3),CCLBA(3,3)
     1                  ,CCLBB(3,3),PLN,RATEC(3),THTC(3)
      COMMON / STAGE / ISTAGE
C
      DATA KO / 0 /
      DATA TLAST / 0.0 /
C
      DATA WTLOU,WTLHU,WTLOON,WTLHON/4*0.0/
C
      NTS = NS + NPAR
      KO = KO + 1
C
C   FILTER STATE VARIABLES
C
      WRITE(*, 901) VAR(1), TLAST, KO, NMEAS
      WRITE(7,901) VAR(1), TLAST, KO, NMEAS
      WRITE(7,902) (XKM(I), I = 1, NTS)
      WRITE(7,902) (PKM(I,I), I = 1, NTS)
C     WRITE(7,902) (RG(I), I = 1, NTS)
      WRITE(7,902) COVZ
      WRITE(7,902) RESID
C
      IF ( KO - NMEAS ) 20, 10, 10
   10 CONTINUE
C
C   MAIN ENGINE PERFORMANCE ESTIMATES
C
      WRITE(8,902) VAR(1),PLN,(TVACL(I),I = 1, 3),(XISPL(I),I = 1, 3)
      WRITE(8,903) (WDO2H(I),I = 1, 3),(WDH2H(I),I = 1, 3)
      WRITE(8,903) (XLH(2,I), I = 1, 3),(XLH(3,I), I = 1, 3)
      WRITE(8,903) (WD(I), I = 1, 3),WTLOU,WTLHU,WTLOON,WTLHON
C
      IF ( ISTAGE.EQ.1 ) THEN
C
C   SOLID ROCKET BOOSTER PERFORMANCE ESTIMATES
C
      WRITE(9,910)VAR(1),POHHAT(1),XISPS(1),TVACS(1),XMD(1),AEXIT(1)
      WRITE(9,910)VAR(1),POHHAT(2),XISPS(2),TVACS(2),XMD(2),AEXIT(2)
C
      ELSE
```

```
      END IF
C
      TLAST = VAR(1)
C
      KO = 0
C
   20 CONTINUE
C
      RETURN
C
  901 FORMAT( 2E15.8, 2I5 )
  902 FORMAT( 8E10.4 )
  903 FORMAT( 10X, 7E10.4 )
  910 FORMAT(3X,F7.3,2F10.3,F13.1,2F10.3)
C
      END
```

```
      SUBROUTINE ZDVEC
C
      COMMON / DRVITV / XDOT(71), PD(71,71)
      COMMON / PROGAG / X(71), P(71,71)
C
      DATA TAUACB,TAURAZ,TAUREL,TAURRG / 1.E+2,3*4.E+2 /
C
      OOTAUA = 1./TAUACB
      OOTAUZ = 1./TAURAZ
      OOTAUE = 1./TAUREL
      OOTAUR = 1./TAURRG
C
      XDOT(50) = -OOTAUA*X(50)
      XDOT(51) = -OOTAUA*X(51)
      XDOT(52) = -OOTAUA*X(52)
C
      XDOT(53) = -OOTAUZ*X(53)
      XDOT(54) = -OOTAUE*X(54)
      XDOT(55) = -OOTAUR*X(55)
      XDOT(56) = -OOTAUZ*X(56)
      XDOT(57) = -OOTAUE*X(57)
      XDOT(58) = -OOTAUR*X(58)
      XDOT(59) = -OOTAUZ*X(59)
      XDOT(60) = -OOTAUE*X(60)
      XDOT(61) = -OOTAUR*X(61)
C
      RETURN
      END
```

```
      SUBROUTINE LINFB
C
      COMMON DER(2700), VAR(2700), TEMP(5400), NDER
      COMMON / LINFMT / F(71,71), NS, NPAR
      COMMON / CONST / A(71), S(71), R(12)
C
      DATA TAUACB,TAURAZ,TAUREL,TAURRG / 1.E+2,3*4.E+2 /
C
      OOTAUA = 1./TAUACB
      OOTAUZ = 1./TAURAZ
      OOTAUE = 1./TAUREL
      OOTAUR = 1./TAURRG
C
      F(50,50) = -OOTAUA
      F(51,51) = -OOTAUA
      F(52,52) = -OOTAUA
C
      F(53,53) = -OOTAUZ
      F(54,54) = -OOTAUE
      F(55,55) = -OOTAUR
      F(56,56) = -OOTAUZ
      F(57,57) = -OOTAUE
      F(58,58) = -OOTAUR
      F(59,59) = -OOTAUZ
      F(60,60) = -OOTAUE
      F(61,61) = -OOTAUR
C
      S(50) = 2.*A(50)**2/TAUACB
      S(51) = 2.*A(51)**2/TAUACB
      S(52) = 2.*A(52)**2/TAUACB
C
      S(53) = 2.*A(53)**2/TAURAZ
      S(54) = 2.*A(54)**2/TAUREL
      S(55) = 2.*A(55)**2/TAURRG
      S(56) = 2.*A(56)**2/TAURAZ
      S(57) = 2.*A(57)**2/TAUREL
      S(58) = 2.*A(58)**2/TAURRG
      S(59) = 2.*A(59)**2/TAURAZ
      S(60) = 2.*A(60)**2/TAUREL
      S(61) = 2.*A(61)**2/TAURRG
C
      RETURN
      END
```

```
      SUBROUTINE REFRAC ( IRDR, EL, RNGH, DELEL, DELRNG )
C
      COMMON / EDATA / RE, FLAT, OMEGE, XMU, XJ2, CRAD
      COMMON / RDRDAT / XXNO(3), RLAT(3), RLONG(3), SALT(3), NRDR
C
      DATA A, B, C / 17590., 30.55, 0.0 /
      DATA RO / 6378165. /
C     DATA CRAD / 57.295779 /
C
      RANGE = .3048*RNGH
      HSTA = .3048*SALT( IRDR )
      XNO = XXNO( IRDR )
C
      ALT = SQRT( RO**2 + RANGE**2 + 2.*RO*RANGE*SIN( EL/CRAD ) ) - RO
C
      HS0 = 7000.
C
      DO 10 I = 1, 50
C
      HS = A - B*(XNO*1.E+6)*EXP( ( HSTA - C )/HS0 )
C     WRITE(*, 901) HS, HS0
901   FORMAT ( 10X, 2E12.5 )
C
      IF ( ABS( HS - HS0 ).LE. 1.0 ) GO TO 20
C
      HS0 = HS
C
C     WRITE(*, 902) I
902   FORMAT ( 22H FAILED TO CONVERGE IN,I5, 18H STEPS IN "REFRAC")
C
10    CONTINUE
C
20    CONTINUE
C
      IF ( ALT.LT.6000. ) THEN
C
      RRNG = RANGE/RO
      RR2 = RRNG*RRNG
C
      RALT = ALT/HS
      RA2 = RALT*RALT
      RA3 = RA2*RALT
C
      DELRNG = XNO*RANGE*( 1. + 67.*RR2 )*
     1          ( 1. - .49939*RALT + .17472*RA2 - .04344*RA3 )
      DELEL = XNO*RANGE*COS(EL/CRAD)/(2.*HS)*( 1. + 46.*RR2 )*
     1         ( 1.-.33324*RALT + .08558*RA2 - .01681*RA3 )
C
      ELSE
C
         IF ( ALT.LT.1.E+5 ) THEN
C
         HSTR = ( 1. - EXP( -ALT/HS ) )*HS
C
         ELSE
```

```
C
      HSTR = HS
C
      END IF
C
     XNO2 = XNO*XNO
     RALT = ALT/RO
     RA2 = RALT*RALT
     RHS = HSTR/RO
C
     A1 = .394 + 1.16E+5*XNO2
     A2 = .009 + 7.1E+5*XNO2
     A3 = .004 + 1.E+5*XNO2
     B1 = 59.9 + 1.14E+4*XNO - 1.9E+7*XNO2
     B2 = -3.379 - 1.E+3*XNO + 8.E+6*XNO2
     B3 = .007
     C1 = .7181 - 246.*XNO + 2.1E+4*XNO2
     C2 = 27.5 - 7.8E+3*XNO - 9.96E+7*XNO2
     C3 = -4.2 - 2.3E+3*XNO - 4.32E+7*XNO2
     C4 = 141.1 - 1.1E+4*XNO + 4.03E+8*XNO2
     C5 = -20.4 - 9.E+4*XNO - 3.1E+7*XNO2
C
     AA = ( A1 + A2*RALT )/( 1. + A3*RALT )
     BB = ( B1 + B2*RALT )/( 1. + B3*RALT )
     CC = ( C1 + C2*RALT + C3 * RA2 )/( 1. + C4*RALT + C5*RA2 )
C
     SEL = SIN( EL/CRAD )
     CEL = COS( EL/CRAD )
C
     ROOT = SQRT( SEL**2 + 2.*RHS )
C
     DELEL = XNO*CEL*( 1.+AA*EXP( -BB*SEL ) )/(( 1.-CC )*ROOT+C*SEL)
     DELEL = DELEL*( 1.- 2.*RHS/( ROOT + SEL ) )
     DELRNG = 2.*XNO*HSTR/( ROOT + SEL )
     DELRNG = DELRNG*( 1.-2.7E+7*(XNO**1.5)*RHS*(CEL**(1.4E+6*XNO)) )
C
     END IF
C
     DELRNG = DELRNG/.3048
     DELEL = DELEL*CRAD
C
     RETURN
     END
```

ORIGINAL PAGE IS
OF POOR QUALITY

C-3

Distribution

	No. Copies
MSFC/AS24D	5
MSFC/AT01	1
MSFC/CC01/Wofford	1
MSFC/EM12A	1
MSFC/EP55	5+repro
NASA/Scientific and Technical Information Facility	1+repro