3.3.05

U.S. Department
of Transportation

Federal Aviation
Administration

800 Independence Avenue, SW
Washington, DC 20591

March 3, 2005

Captain H. Ray Lahr (Ret)
18254 Coastline Drive
Malibu, CA 90265-5604

Freedom of Information Act (FOIA) Request 2005-3308

Dear Captain Lahr:

This is in reference to the February 22, 2005, referral from the Central Intelligence Agency of Federal Aviation Administration documents concerning TWA flight 800 and your October 8, 2003, FOIA request.

The Air Traffic Organization has reviewed the documents and determined that the records are releasable under the FOIA (22 pages)

There are no fees for processing your request, as the cost was under $10.

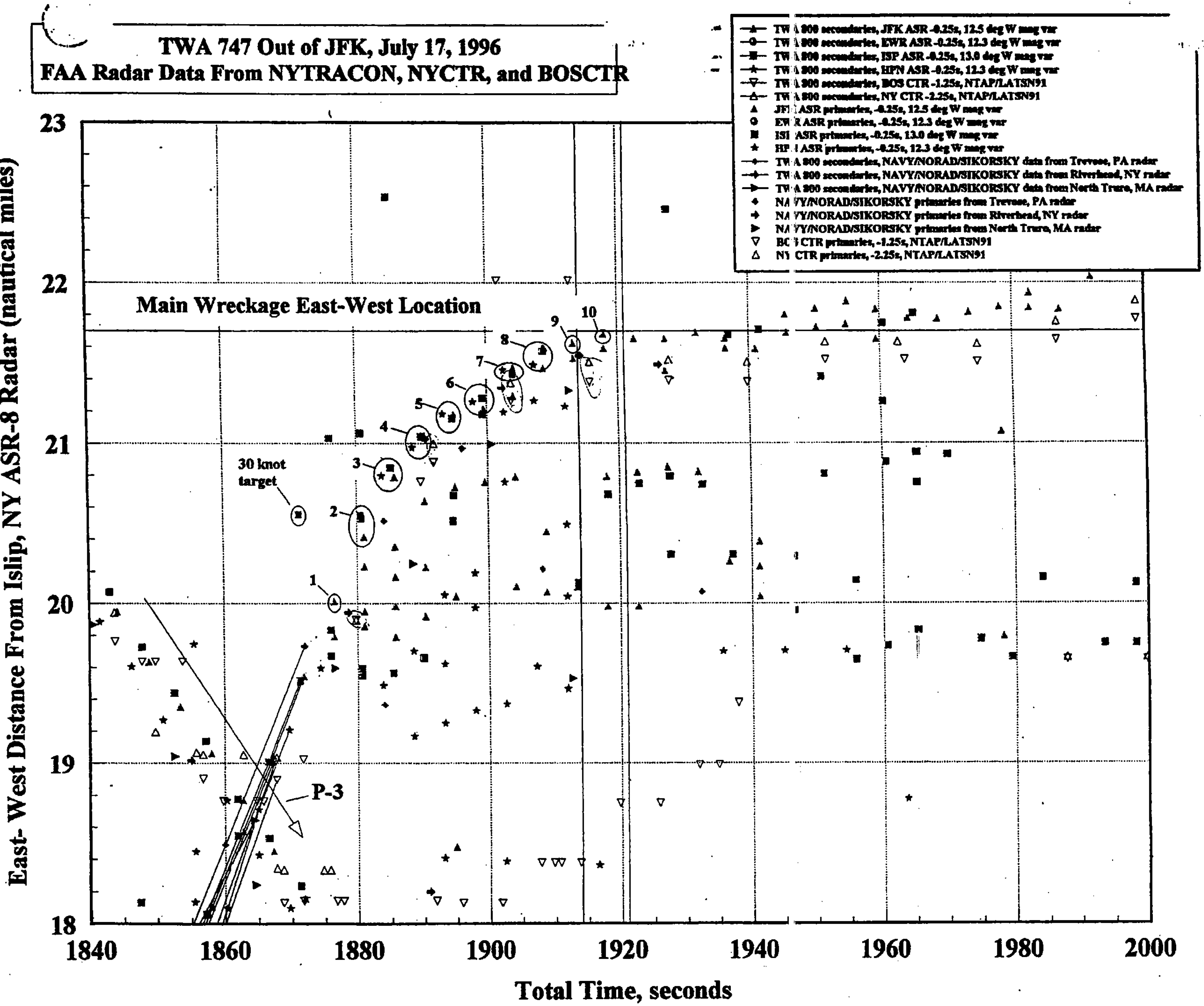
Sincerely,

Tracy Paquin

Tracy Paquin
Executive Manager for
Regions and Center Operations

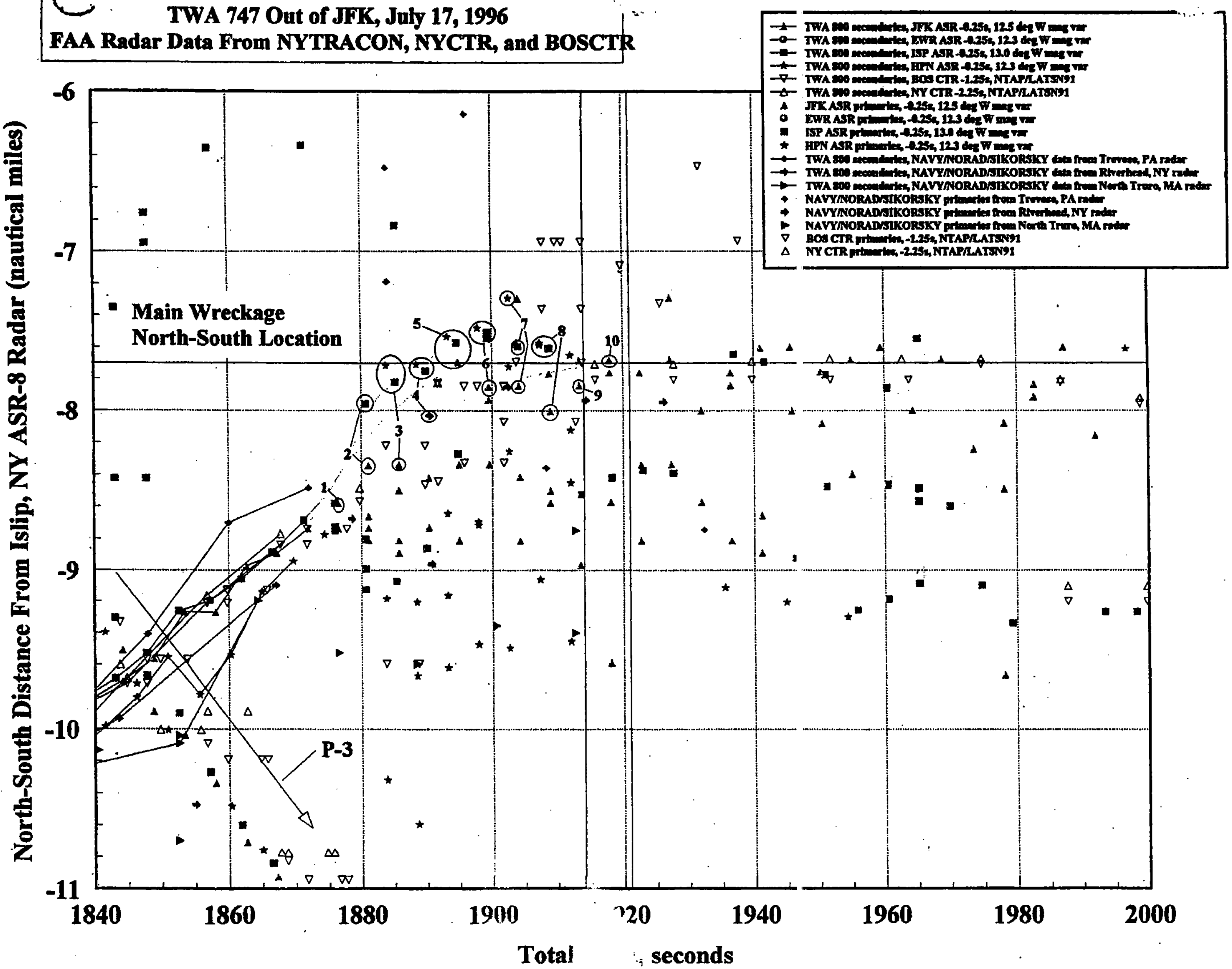
Enclosures (2)

TWA 747 Out of JFK, July 17, 1996
FAA Radar Data From NYTRACON, NYCTR, and BOSCTR



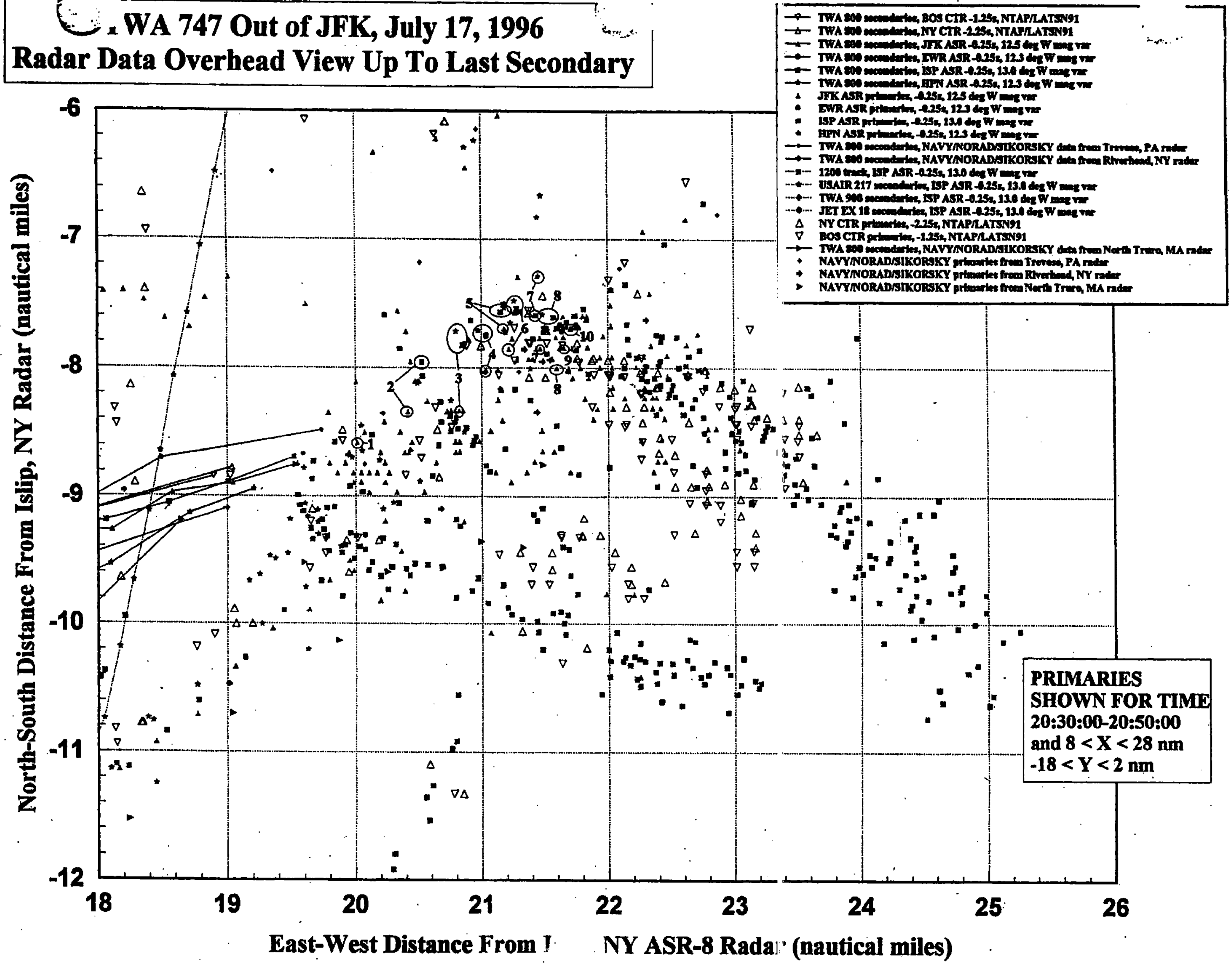
TWAXT_97. SGX
SLIDE 1

TWA 747 Out of JFK, July 17, 1996
FAA Radar Data From NYTRACON, NYCTR, and BOSCTR



TWAXT_97.SGX
SLIDE 2

TWA 747 Out of JFK, July 17, 1996
Radar Data Overhead View Up To Last Secondary



TWAXY_97.SGX
SLIDE 8

8 Jun 98

~~05 Dec 97~~

```
C
C
C  Program to analyze FAA radar data [xy2rng.f]
C
C      The first part of the problem is to correlate
C      data points from the JFK radar with data
C      points in the other radars.  Only range measurements are used.
C      Plots of the JFK range vs. time show what is probably the main
C      object.
C
C      A preliminary position history is calculated by finding the
C      intersection of main object JFK range data and a line radiating
C      from the initial position (last JFK transponded point) at an
C      assumed flight direction.  The distance from these points to the
C      other radars is calculated and compared with the range data
C      measured from that radar.  To match the apparent main object
C      data in both data sets requires a rotation of the xy coordinate
C      system used for JFK (probably due to an incorrect magnetic
C      offset.
C
C      Once the main object points are chosen for all data sets,
C      another set of positions is calculated using the range
C      measurements of the same object from the JFK and ISP radars
C      (triangulation).  (Uncertainty calculations show that these
C      positions have a large cross range uncertainty.)
C
C      Another technique solves for points along a line to
C      best fit (least squares) the radar range and azimuth data.
C
C  Definitions of variables used in main routine
C
C    NOBS     Number of data pairs from each radar
C    MAIN     Index of data selected as main object
C    TRDR     Time of xy radar data (sec from even min)
C    ROBS     Range measurements
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/ RDRDTA / TRDR(99,9), XRDR(99,9), YRDR(99,9),
     <                 ROBS(99,9), AOBS(99,9), TOBSN(199,9),
     <                 ROBSN(199,9), AOBSN(199,9), HOBSN(199,9),
     <                 NOBS(9), NOBSN(9)
C
      COMMON/ OBSRDR / RMAIN(20,9), AMAIN(20,9), TMAIN(20,9),
     <                 RORDR(9), AORDR(9), TORDR(9),
     <                 MAIN(20,9), NMAIN(9)
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
        REAL*8  LATRDR, LONRDR, MAGOFF
C
      CHARACTER*1 STAR, BLANK, MNOBJ
      DATA  STAR / '*' /, BLANK / ' ' /
      CHARACTER*4 CRDR(9)
      DATA  CRDR / ' JFK', ' ISP', ' HPN', ' RHD', 5*'    ' /
C
C
C  Set constant parameters
C
      CALL  PARAMS
C
C  Call subroutine to read XY data
C
      CALL  RDDATA
C
C  Store data for points selected as main object
C
      CALL  STMAIN
C
C  Write headers for output
C
      WRITE(8,400)
C
C  Print JFK range data and calculate position assuming fixed flt azimuth
C
      CALL  JFKPLN
C
C  Interpolate main obj JFK range data to ISP data times and triangulate
C
      CALL  ISPTRI
C
```

```
C
      J = MOD( N,3 ) + 1
      K = MOD( N+1,3 ) + 1
C
      Y(N) = X(N)
      Y(J) = (X(K) * ST) + (X(J) * CT)
      Y(K) = (X(K) * CT) - (X(J) * ST)
C
      RETURN
C
      END
C
C ******************************************************************
C
      SUBROUTINE  RAH2LL( LAT, LON, RNM, ARAD, HNM, KR )
C
C Convert range, az, height to latitude and longitude
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
        REAL*8  LATRDR, LONRDR, MAGOFF
      COMMON/ XYREF / LATREF, LONREF
        REAL*8 LATREF, LONREF
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
      DIMENSION X(3), XT(3), XE(3)
      REAL*8  LAT, LON
C
C
C Spherical earth equations
C
      IF( FEORSE .EQ. 1 )  THEN
        R = RNM * CNMM
        H = HNM * CNMM
        HRDR = ALTRDR(KR) * CFM
        EL = DASIN( ( ( RE + HRDR )**2 + R**2 - ( RE + H )**2 )
     <          / ( 2. * R * ( RE + HRDR ) ) )
C
        X(1) = R * DCOS(EL) * DCOS(ARAD)
        X(2) = R * DCOS(EL) * DSIN(ARAD)
        X(3) = R * DSIN(EL)
C
        CALL  ROT( X, XT, (LATRDR(KR)-90.)*CDR, 1 )
        CALL  ROT( XT, X, (270.+LONRDR(KR))*CDR, 3 )
C
        REV = 0.
        DO  1  I = 1, 3
          XE(I) = X(I) + XOFRDR(I,KR)
          REV = REV + XE(I)**2
 1      CONTINUE
        REV = DSQRT(REV)
C
        LAT = DASIN( XE(3) / REV ) / CDR
        LON = DATAN2( XE(2), XE(1) ) / CDR
C
      END IF
C
C Flat earth equations
C
      IF( FEORSE .EQ. 0 )  THEN
        RGTRUE = DSQRT( RNM**2 - HNM**2 )
        X(1) = RGTRUE * DCOS(ARAD) + XOFRDR(1,KR)
        X(2) = RGTRUE * DSIN(ARAD) + XOFRDR(2,KR)
        LAT = LATREF + X(2) / 60.
        LON = LONREF - X(1) / 60. / DCOS(LAT*CDR)
      END IF
C
      RETURN
      END
```

```
C
      DIMENSION  XORDR(9), YORDR(9)
C
C
C  Constants
C
      CNMM = 1852.
      CFM = 0.3048D0
      CFTNM = CFM / CNMM
      CDR = 3.14159265D0 / 180.
      RE = 6370000.
      FEORSE = 0.
C
      NRDRS = 4
C
      LATRDR(1)  = 40.6362
      LONRDR(1)  = 73.7669
      ALTRDR(1)  = 17.
      MAGOFF(1)  = -12.5
      TBIAS(1)  = -0.25
C
      LATRDR(2)  = 40.7940
      LONRDR(2)  = 73.1001
      ALTRDR(2)  = 27.
      MAGOFF(2)  = -13.
      TBIAS(2)  = -0.25
C
      LATRDR(3)  = 41.0612
      LONRDR(3)  = 73.7142
      ALTRDR(3)  = 77.
      MAGOFF(3)  = -12.3
      TBIAS(3)  = -0.25
C
      LATRDR(4)  = 40.8785
      LONRDR(4)  = 72.6878
      ALTRDR(4)  = 260.
      MAGOFF(4)  = 0.
      TBIAS(4)  = -2.8
C
      LATRDR(5)  = 40.8008
      LONRDR(5)  = 72.6276
      ALTRDR(5)  = 0.
C
C  Impact at 40 39 48, 72 37 26
      LATIMP = 40.6632
      LONIMP = 72.6238
      T0 = 12.26
      TFLT = 49.
      TIMP = T0 + TFLT
C
      LATREF = LATRDR(1)
      LONREF = LONRDR(1) + 0.00001
C
C  Calculated fixed parameters
C
      DO  1  I = 1, NRDRS
C
        ALTNM = ALTRDR(I) * CFTNM
        CALL  LLA2X( LATRDR(I), LONRDR(I), ALTNM, XOFRDR(1,I),
     <               XOFRDR(2,I), XOFRDR(3,I) )
C
 1    CONTINUE
C
C
      RETURN
      END
C
C  ****************************************************************
C
      SUBROUTINE  RDDATA
      IMPLICIT REAL*8(A-H,O-Z)
C
C  Read in radar data
C
C  Parameters used in RDDATA
C
C    TRDR      Times of xy data (sec from even minute)
C    X         X position relative to radar (nm)
C    Y         Y position relative to radar (nm)
C    XODRDR    X positions of radar in refernce coordinate system
```

```
 21      CONTINUE
 22     CONTINUE
C
        RETURN
C
 500    FORMAT( 4F10.4 )
C
        END
C
C
C  ********************************************************************
C
        SUBROUTINE DMS(LAT,LON,LATDEG,LATMIN,LATSEC,LONDEG,LONMIN,LONSEC)
C
C  Convert decimal angles to deg, min, sec
C
        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 LAT, LON
C
        LATDEG = LAT
        LATMIN = ( LAT - LATDEG ) * 60.
        LATSEC = ( ( LAT - LATDEG ) * 60. - LATMIN ) * 60.
        LONDEG = LON
        LONMIN = ( LON - LONDEG ) * 60.
        LONSEC = ( ( LON - LONDEG ) * 60. - LONMIN ) * 60.
C
        RETURN
        END
C
C *********************************************************************
C
        FUNCTION  RNTRP( TI, R, T, N )
C
C  Interpolation function
C
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION R(1), T(1)
C
        DO  1  I = 2, N
          IF( TI .LE. T(I) )  GO TO  2
 1      CONTINUE
        I = N
C
 2      CONTINUE
C
        RNTRP = R(I-1) + ( R(I) - R(I-1) ) * ( TI - T(I-1) ) /
       <         ( T(I) - T(I-1) )
C
        RETURN
        END
C
C  ********************************************************************
C
        SUBROUTINE  LLA2X( LAT, LON, ALT, X )
C
C  Convert latitude, longitude, altitude to cartesian position
C
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON/ XYREF / LATREF, LONREF
           REAL*8 LATREF, LONREF
        COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
        DIMENSION X(3)
        REAL*8  LAT, LON
C
C
C  Flat earth equations
C
        IF( FEORSE .EQ. 0 )  THEN
          X(1) = ( LONREF - LON ) * 60. * DCOS( LAT * CDR )
          X(2) = ( LAT - LATREF ) * 60.
          X(3) = ALT
        END IF
C
C  Spherical earth equations
C
        IF( FEORSE .EQ. 1 )  THEN
          R = RE + ALT * CNMM
          X(1) = R * DCOS(LAT*CDR) * DCOS(-LON*CDR)
          X(2) = R * DCOS(LAT*CDR) * DSIN(-LON*CDR)
          X(3) = R * DSIN(LAT*CDR)
```

```
C
       IMPLICIT REAL*8(A-H,O-Z)
C
       COMMON/ RDRDTA / TRDR(99,9), XRDR(99,9), YRDR(99,9),
     <                  ROBS(99,9), AOBS(99,9), TOBSN(199,9),
     <                  ROBSN(199,9), AOBSN(199,9), HOBSN(199,9),
     <                  NOBS(9), NOBSN(9)
       COMMON/ OBSRDR / RMAIN(20,9), AMAIN(20,9), TMAIN(20,9),
     <                  R0RDR(9), A0RDR(9), T0RDR(9),
     <                  MAIN(20,9), NMAIN(9)
       COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                  MAGOFF(9), NRDRS
          REAL*8  LATRDR, LONRDR, MAGOFF
C
C                    Main object, JFK, upper
       DATA  MAIN / 2,3,9,12,16,19,23,27,31,33,37, 9*0,
C                    Main object, JFK, lower
C      DATA  MAIN / 2,3,9,12,16,20,22,26,31,33,37, 9*0,
C                    Main object, ISP
     <              2,5,7,9,11,12,14,15,12*0,
C                      Main object, HPN from MO
C    <              1,7,18*0,
C                      Main object, HPN from CP
     <              1,2,4,7,11,17,19,13*0,
C                      Main object, RHD
     <              1,3,6,9,16*0,  100*0 /
C
C
       DO  2  J = 1, NRDRS
         K = 1
         DO  1  I = 1, NOBS(J)
           IF( I .EQ. MAIN(K,J) )  THEN
             RMAIN(K,J) = ROBS(I,J)
             AMAIN(K,J) = AOBS(I,J)
             TMAIN(K,J) = TRDR(I,J)
             K = K + 1
           END IF
 1       CONTINUE
         NMAIN(J) = K - 1
       CONTINUE
C
       RETURN
       END
C
C  *******************************************************************
C
       SUBROUTINE  JFKPLN
C
C  Print range observations from JFK.  Also, calculate first estimate
C  of flight path.  Calculate position of intersection of assumed
C  flight path direction and range from JFK.
C
C  Definition of variables used:
C
C    DM       Offset of xy axes - to west (deg)
C    DMR      Offset of xy axes - to west (rad)
C    AZFP     Flight azimuth - up from east (deg)
C    X0N      X position of X0JFK in N/E system, origin at JFK (nm)
C    Y0N      Y position of Y0JFK in N/E system, origin at JFK (nm)
C    LAT0     Latitude of last transponded
C    LON0     Longitude of last transponded
C    R0P      Distance from JFK to where flt path would cross x axis
C    A0       Az from JFK to last transponded in true N/E system
C    R        Range of xy data to JFK
C    BE       Angle: JFK - data - last transponded
C    RR       Distance along flight path to where rng data would intersect
C    XCALC    Calculated x pos given range to JFK and path along assumed
C               flight azimuth
C    YCALC    Calculated y pos given range to JFK and path along assumed
C               flight azimuth
C    LATCAL   Latitude using JFK data and assumed plane constraint
C    LONCAL   Longitude using JFK data and assumed plane constraint
C    D2RDR    Distance from JFK points to other radars
(    TRDR     Times of radar data rel to even minute (sec)
C
       IMPLICIT REAL*8 (A-H,O-Z)
C
       COMMON/ OBSRDR / RMAIN(20,9), AMAIN(20,9), TMAIN(20,9),
     <                  R0RDR(9), A0RDR(9), T0RDR(9),
     <                  MAIN(20,9), NMAIN(9)
```

```
      COMMON/ XYREF / LATREF, LONREF
         REAL*8 LATREF, LONREF
      COMMON/ CPATH / TPATH(20), NPATH, NPATHP, NPATHN
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
C
      EQUIVALENCE ( APARAM(1),  LAT0 ),       ( APARAM(2),  LON0 ),
     <            ( APARAM(3),  AZFP0 ),      ( APARAM(4),  DT   ),
     <            ( APARAM(5),  TSTRT ),      ( APARAM(11), BARDR(1) ),
     <            ( APARAM(21), BRRDR(1) ), ( APARAM(31), RPATH(1) ),
     <            ( APARAM(51), TBIASS(1) ),( APARAM(61), ALT(1) ),
     <            ( APARAM(6),  AZFPD(1) ), ( APARAM(8),  AZFPN )
      DIMENSION RPATH(20), BARDR(9), BRRDR(9), TBIASS(9), ALT(20),
     <          X(3), AZFPD(2)
      DIMENSION TMAINJ(21), RMAINJ(21), AMAINJ(21)
      REAL*8 LAT, LON, LAT0, LON0, LATC, LONC
      CHARACTER*4 CRDR(9)
      DATA  CRDR / ' JFK', ' ISP', ' HPN', ' RHD', 5*'    ' /
C
      IWOUT = 9
      IWPLT0 = 50
C
      CALL  LLA2X( LATRDR(5), LONRDR(5), 0.0D0, XOFRDR(1,5),
     <             XOFRDR(2,5), XOFRDR(3,5) )
C
C  Print range and position along flight path vs node times
C
      WRITE( IWOUT, 200 )
      DO  1  I = 1, NPATH
        T = TPATH(I)
C
        CALL  DXDY( DX, DY, T, RPATH, TPATH, AZFP0, AZFPD, AZFPN,
     <              NPATH )
C
        LAT = LAT0 + DY / 60.
        LON = LON0 - DX / 60. / DCOS(LAT*CDR)
C       RR = DSQRT( DX**2 + DY**2 )
C       IF( T .LT. 0.0 )  RR = -RR
        RR = RPATH(I)
C
        CALL  DMS(LAT,LON,LATDEG,LATMIN,LATSEC,LONDEG,LONMIN,LONSEC)
        WRITE( IWOUT, 100 )  T, RR, LATDEG, LATMIN, LATSEC,
     <    LONDEG, LONMIN, LONSEC
        LON = -LON
        WRITE( IWPLT0, 500 )  LON, LAT
C
C  Interpolate radar observations to node times
C
        HNM = ALT(I) * CFTNM
        DO  3  J = 1, NRDRS
C
C  Fill in interpolation array with observations for this radar
C
          NM1 = 0
C
C  Fill in interpolation array for negative node times
C
          DO  4  K = 1, NOBSN(J)
            NM1 = NM1 + 1
            TMAINJ(NM1) = TOBSN(K,J)  + TBIASS(J)  - TSTRT
            RMAINJ(NM1) = ROBSN(K,J)  - BRRDR(J)
            AMAINJ(NM1) = AOBSN(K,J)  - BARDR(J)  * CDR
 4        CONTINUE
C
C  Fill in interpolation array for positive node times
C
          DO  2  K = 1, NMAIN(J)
            NM1 = NM1 + 1
            TMAINJ(NM1) = TMAIN(K,J)  + TBIASS(J)  - TSTRT
            RMAINJ(NM1) = RMAIN(K,J)  - BRRDR(J)
            AMAINJ(NM1) = AMAIN(K,J)  - BARDR(J)  * CDR
 2        CONTINUE
C
          IF( T .GE. TMAINJ(1) .AND. T .LE. TMAINJ(NM1) )  THEN
            RNG = RNTRP( T, RMAINJ, TMAINJ, NM1 )
            AZ = RNTRP( T, AMAINJ, TMAINJ, NM1 )
            CALL  RAH2LL( LATC, LONC, RNG, AZ, HNM, J )
            WRITE(2,500) T, LONC, LATC
          END IF
C
 3      CONTINUE
```

Handwritten annotations: "HFT = HNM * CFTNM" (beside RR = RPATH(I)); ", HFT" (after LONSEC); "HNM = ALTTNM(T, ALT, ALTB)" (beside Interpolate line); "C" before HNM line; illegible note at upper right.

```
      END
C
C ********************************************************************
C
      SUBROUTINE  ISPTRI
C
C Interpolate main obj JFK range data to ISP data time and triangulate
C
C Definition of variables used:
C
C   NOBS      Number of data pairs in each radar's input file
C   LATTRI    Latitude using JFK/ISP triangulation
C   LONTRI    Longitude using JFK/ISP triangulation
C   DJF2IS    Distance from JFK to ISP
C   AJF2IS    Azimuth from JFK to ISP (up from east)
C   RMAIN     Observed range data from radars
C   TMAIN     Times of observed radar data--main object
C   RJ        Range to JFK interpolated from main object JFK range data
C               to ISP data times
C   AID       Angle from ISP to data relative to JFK
C   AD        Azimuth of JFK/ISP intersection, up from east
C   XOFRDR    Postion of radars in E/N cartesian
C   XCALC2    X position of JFK/ISP intersectin in JFK-E/N cartesian
C   YCALC2    Y position of JFK/ISP intersectin in JFK-E/N cartesian
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/ OBSRDR / RMAIN(20,9), AMAIN(20,9), TMAIN(20,9),
     <                 RORDR(9), AORDR(9), TORDR(9),
     <                 MAIN(20,9), NMAIN(9)
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
         REAL*8  LATRDR, LONRDR, MAGOFF
      COMMON/ RDRDTA / TRDR(99,9), XRDR(99,9), YRDR(99,9),
     <                 ROBS(99,9), AOBS(99,9), TOBSN(199,9),
     <                 ROBSN(199,9), AOBSN(199,9), HOBSN(199,9),
     <                 NOBS(9), NOBSN(9)
      COMMON/ CTBIAS / TBIAS(9)
      COMMON/ XYREF / LATREF, LONREF
         REAL*8 LATREF, LONREF
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
C
C
      REAL*8  LATTRI, LONTRI
      CHARACTER*1 STAR, BLANK, MNOBJ
      DATA  STAR / '*' /, BLANK / ' ' /
C
      WRITE(8,500)
C
      DJF2IS = DSQRT( ( XOFRDR(1,2) - XOFRDR(1,1) )**2
     <              + ( XOFRDR(2,2) - XOFRDR(2,1) )**2 )
      AJF2IS = DATAN2( ( XOFRDR(2,2) - XOFRDR(2,1) ),
     <                 ( XOFRDR(1,2) - XOFRDR(1,1) ) )
      K = 1
      DO  1   I = 1, NOBS(2)
        T = TRDR(I,2) + TBIAS(2) - TBIAS(1)
        RJ = RNTRP( T, RMAIN, TMAIN, NMAIN(1) )
        RI = ROBS(I,2)
        DI = DJF2IS
        COSAID = ( DI**2 + RJ**2 - RI**2 ) / ( 2.0 * DI * RJ )
        AID = DACOS( COSAID )
        AD = AJF2IS - AID
        XCALC2 = RJ * DCOS( AD ) + XOFRDR(1,1)
        YCALC2 = RJ * DSIN( AD ) + XOFRDR(2,1)
        LATTRI = LATREF + YCALC2 / 60.
        LONTRI = LONREF - XCALC2 / 60. / DCOS( LATTRI*CDR )
C
        MNOBJ = BLANK
        IF( I .EQ. MAIN(K,2) )  THEN
          MNOBJ = STAR
          K = K + 1
        END IF
C
        WRITE(8,300)  I, MNOBJ, TRDR(I,2), ROBS(I,2), LATTRI, LONTRI
C
 1    CONTINUE
C
      RETURN
C
 300  FORMAT( I5, A1, F10.2, 6F10.4 )
```

```
C
      RETURN
      END
C
C ********************************************************************
C
      SUBROUTINE  OUTPUT
C
C Print final results
C
      IMPLICIT REAL*8(A-H,O-Z)

      COMMON/ OBSEND / LATIMP, LONIMP, TIMP, T0, TFLT
         REAL*8 LATIMP, LONIMP
C
      COMMON/ OBSRDR / RMAIN(20,9), AMAIN(20,9), TMAIN(20,9),
     <                 R0RDR(9), A0RDR(9), T0RDR(9),
     <                 MAIN(20,9), NMAIN(9)
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
         REAL*8  LATRDR, LONRDR, MAGOFF
      COMMON/ RDRDTA / TRDR(99,9), XRDR(99,9), YRDR(99,9),
     <                 ROBS(99,9), AOBS(99,9), TOBSN(199,9),
     <                 ROBSN(199,9), AOBSN(199,9), HOBSN(199,9),
     <                 NOBS(9), NOBSN(9)
C
      COMMON/ TOSET / DATA(3,3000), W(25), APARAM(100), ND, IR, IW, NDQ
      COMMON/ XYREF / LATREF, LONREF
         REAL*8 LATREF, LONREF
      COMMON/ CPATH / TPATH(20), NPATH, NPATHP, NPATHN
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
C
      EQUIVALENCE ( APARAM(1),  LAT0 ),      ( APARAM(2),  LON0 ),
     <            ( APARAM(3),  AZFP0 ),     ( APARAM(4),  DT   ),
     <            ( APARAM(5),  TSTRT ),     ( APARAM(11), BARDR(1) ),
     <            ( APARAM(21), BRRDR(1) ),  ( APARAM(31), RPATH(1) ),
     <            ( APARAM(51), TBIASS(1) ),( APARAM(61), ALT(1) ),
     <            ( APARAM(6),  AZFPD(1) ),  ( APARAM(8),  AZFPN )
      DIMENSION RPATH(20), BARDR(9), BRRDR(9), TBIASS(9), ALT(20),
     <          X(3), AZFPD(2)
      DIMENSION TMAINJ(21), RMAINJ(21), AMAINJ(21)
      REAL*8 LAT, LON, LAT0, LON0, LATC, LONC
      CHARACTER*4 CRDR(9)
      DATA  CRDR / ' JFK', ' ISP', ' HPN', ' RHD', 5*'    ' /
C
C Print range and position along flight path vs node times
C
      CALL  PRNODE
C
C ********************************************************************  7— ✓
C
C Print lat, lon and obs vs computed range at radar times
C
      IWOUT = 9
      IWPLT0 = 50
      IWPLR0 = 60
C
      DO  11  J = 1, NRDRS
C
        WRITE( IWOUT, 400 )  CRDR(J)
        IWPLT = IWPLT0 + J
        IWPLTR = IWPLR0 + J
C
C Compute latitude, longitude of adjusted range/azimuth location
C
        NOBSPN = NOBSN(J) + NMAIN(J)
        DO  12  I = 1, NOBSPN
          IF( I .LE. NOBSN(J) )  THEN
            T = TOBSN(I,J) + TBIASS(J) - TSTRT
            ATRUE = AOBSN(I,J) - BARDR(J)*CDR
            RTRUE = ROBSN(I,J) - BRRDR(J)
          ELSE
            II = I - NOBSN(J)
            T = TMAIN(II,J) + TBIASS(J) - TSTRT
            ATRUE = AMAIN(II,J) - BARDR(J)*CDR
            RTRUE = RMAIN(II,J) - BRRDR(J)
          END IF
c         HNM = RNTRP( T, ALT, TPATH, NPATH ) * CFTNM   = alt HNM=ALTTNM(T,ALT,ALTB)✓
C
          CALL  RAH2LL( LAT, LON, RTRUE, ATRUE, HNM, J )
```

```
      FEORSE = 1.
C
C  Fill in data array for latitude and longitude at impact
C
      ND = ND + 1
      DATA(1,ND) = LATIMP
      DATA(2,ND) = TIMP
      DATA(3,ND) = 1.
      ND = ND + 1
      DATA(1,ND) = LONIMP
      DATA(2,ND) = TIMP
      DATA(3,ND) = 2.
C
      NOBSNT(1) = NOBSN(1)
      NOBSNT(2) = NOBSN(2)
      NOBSNT(3) = NOBSN(3)
      NOBSNT(4) = NOBSN(4)
C
C  Override number of observations read in
C
      NOBSNT(1) = 6
      NOBSNT(2) = 6
      NOBSNT(3) = 5
      NOBSNT(4) = 2
C
      DO  3  J = 1, NRDRS
C
C  Range and azimuth of transponded points
C
        DO  11  I = NOBSNT(J), 1, -1
          ND = ND + 1
          II = NOBSN(J) - (I-1)
          DATA(1,ND) = ROBSN(II,J)
          DATA(2,ND) = TOBSN(II,J)
          DATA(3,ND) = 3. + 4. * (J-1)
 11     CONTINUE
        DO  12  I = NOBSNT(J), 1, -1
          ND = ND + 1
          II = NOBSN(J) - (I-1)
          DATA(1,ND) = AOBSN(II,J) / CDR
          DATA(2,ND) = TOBSN(II,J)
          DATA(3,ND) = 4. + 4. * (J-1)
 12     CONTINUE
C
C  Fill in data array radar range and azimuth (primaries)
C
        DO  1  I = 1, NMAIN(J)
          ND = ND + 1
          DATA(1,ND) = RMAIN(I,J)
          DATA(2,ND) = TMAIN(I,J)
          DATA(3,ND) = 5. + 4*(J-1)
 1      CONTINUE
        DO  2  I = 1, NMAIN(J)
          ND = ND + 1
          DATA(1,ND) = AMAIN(I,J) / CDR
          DATA(2,ND) = TMAIN(I,J)
          DATA(3,ND) = 6. + 4*(J-1)
 2      CONTINUE
C
 3    CONTINUE
C
C  Set data weights
C
      W(1) = 0.001
      W(2) = 0.001
      W(3) = 0.0625    .03
      W(4) = 0.2       .25
      W(5) = 0.0625
      W(6) = 0.2
      W(7) = 0.0625
      W(8) = 0.2
      W(9) = 0.0625
      W(10) = 0.2
      W(11) = 0.0625
      W(12) = 0.2
      W(13) = 0.0625
      W(14) = 0.2
      W(15) = 0.0625
      W(16) = 0.2
      W(17) = 0.0625
```

correct + blue?

```
      W(18) = 0.2
C
C  Set initial values of parameters
C
      DO  4  I = 1, 100
        APARAM(I) = 0.
 4    CONTINUE
      LAT0 = 40.6456
      LON0 = 72.6718
      AZFP0 = 42.55
      azfp0 = 24.
      AZFPD(1) = 1.45
      AZFPD(2) = 0.
      AZFPN = 24.
      DT = 4.7
      TSTRT = T0
      RPATHP(1) = 0.0
      RPATHP(2) = 0.275
      RPATHP(3) = 1.144
      RPATHP(4) = 1.556
      RPATHP(5) = 1.848
      RPATHP(6) = 1.957
      RPATHP(7) = 2.094
      RPATHP(8) = 2.283
      RPATHP(9) = 2.421
      RPATHP(10)= 2.455
      RPATHP(11) = 2.548
      RPATHP(12) = 2.497
      RPATHP(13) = 2.5
C
      ALTP(1)   = 13700.
      ALTP(2)   = 14000.  13700
      ALTP(3)   = 15000.
      ALTP(4)   = 16000.
      ALTP(5)   = 16000.
      ALTP(6)   = 14000.
      ALTP(7)   = 12000.
      ALTP(8)   = 10000.
      ALTP(9)   =  8000.
      ALTP(10)  =  5000.
      ALTP(11)  =  2000.
      ALTP(12)  =   000.
C
      NPATHP = 12
C
      RPATHN(1) = -0.5
      RPATHN(2) = -1.0
      RPATHN(3) = -1.5
      RPATHN(4) = -2.0
      RPATHN(5) = -2.5
C
      ALTN(1) = 13700.
      ALTN(2) = 13600.
      ALTN(3) = 13500.
      ALTN(4) = 13400.
      ALTN(5) = 13300.
C
      NPATHN = 5
C
      NPATH = 0
C
      if(npathn.ne.0) then
      TN0 = -NPATHN * DT - 0.7
      DO  7  I = NPATHN, 1, -1
        NPATH = NPATH + 1
        RPATH(NPATH) = RPATHN(I)
        TPATH(NPATH) = TN0 + (NPATH-1) * DT
        ALT(NPATH) = ALTN(I)
 7    CONTINUE
      end if
C
      DO  8  I = 1, NPATHP
        NPATH = NPATH + 1
        RPATH(NPATH) = RPATHP(I)
        ALT(NPATH) = ALTP(I)
        IF( I .EQ. 1 )  TPATH(NPATH) = 0.
        IF( I .NE. 1 )  TPATH(NPATH) = (I-1) * DT - 0.7
 8    CONTINUE
C
      DO  5  I = 1, 9
```

```
        TBIASS(I) = TBIAS(I)
 5    CONTINUE
C
C  Calculate radar positions in reference coordinate system
C
      DO  6  I = 1, NRDRS
        ALTNM = ALTRDR(I) * CFTNM
        CALL LLA2X( LATRDR(I), LONRDR(I), ALTNM, XOFRDR(1,I),
     <              XOFRDR(2,I), XOFRDR(3,I) )
 6    CONTINUE
C
C
C  Call least squares fit package
C
      CALL LSQSET
C
      RETURN
      END
C
C ***********************************************************************
C
      SUBROUTINE  PXRES( KTHDV )
C
C  Calculate observations from current values of parameters
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/ TOSET / DATA(3,3000), W(25), APARAM(100), ND, IR, IW, NDQ
      COMMON/ TOPXR / COMPUT, PP(100)
      COMMON/ XYREF / LATREF, LONREF
         REAL*8 LATREF, LONREF
C
      COMMON/ OBSRDR / RMAIN(20,9), AMAIN(20,9), TMAIN(20,9),
     <                 RORDR(9), AORDR(9), TORDR(9),
     <                 MAIN(20,9), NMAIN(9)
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
         REAL*8  LATRDR, LONRDR, MAGOFF
      COMMON/ RDRDTA / TRDR(99,9), XRDR(99,9), YRDR(99,9),
     <                 ROBS(99,9), AOBS(99,9), TOBSN(199,9),
     <                 ROBSN(199,9), AOBSN(199,9), HOBSN(199,9),
     <                 NOBS(9), NOBSN(9)
      COMMON/ CPATH / TPATH(20), NPATH, NPATHP, NPATHN
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
C
      REAL*8 LAT, LON, LAT0, LON0
      EQUIVALENCE ( APARAM(1),  LAT0 ),      ( APARAM(2),  LON0 ),
     <            ( APARAM(3),  AZFP0 ),     ( APARAM(4),  DT   ),
     <            ( APARAM(5),  TSTRT ),     ( APARAM(11), BARDR(1) ),
     <            ( APARAM(21), BRRDR(1) ),  ( APARAM(31), RPATH(1) ),
     <            ( APARAM(51), TBIASS(1) ),( APARAM(61), ALT(1) ),
     <            ( APARAM(6),  AZFPD(1) ),  ( APARAM(8),  AZFPN )
C
      DIMENSION RPATH(20), BARDR(9), BRRDR(9), TBIASS(9), ALT(20),
     <          X(3), AZFPD(2)
C
C
      KD = DATA(3,KTHDV)
      KRDR = ( KD + 1 ) / 4
      T = DATA(2,KTHDV) + TBIASS(KRDR) - TSTRT
C
C  Current distance along flight path and current position
C
      CALL  DXDY( DX, DY, T, RPATH, TPATH, AZFP0, AZFPD, AZFPN, NPATH )
C
      LAT = LAT0 + DY / 60.
      LON = LON0 - DX / 60. / DCOS( LAT * CDR )
      IF( KD .EQ. 1 )  COMPUT = LAT
      IF( KD .EQ. 2 )  COMPUT = LON
C
      IF( KRDR .NE. 0 )  THEN
C
C       ALTNM = RNTRP( T, ALT, TPATH, NPATH ) * CFTNM      <-- alt
                                                — ALTNM = ALTTNM(T, ALT, ALTB) ✓
        CALL  LLA2X( LAT, LON, ALTNM, X )
C
        CALL  X2RAZ( RNG, AZ, EL, X, KRDR )
C
        KRORAZ = MOD( KD + 1 - 4*KRDR, 2 )
        IF( KRORAZ .EQ. 0 )  COMPUT = RNG + BRRDR(KRDR)
        IF( KRORAZ .EQ. 1 )  COMPUT = AZ  + BARDR(KRDR)
      END IF
```

```
 500  FORMAT( /// T25, 'ISP Data and JFK/ISP triangulated position ' //
     <  '             Time  Rng fm ISP   Lat-tri    Lon-tri ' )
C
      END
C
C ********************************************************************
C
      SUBROUTINE  FIT
C
C  Initialize parameters for LSQSET for least squares fit
C
C    Observations are range and azimuth from JFK and ISP and impact location
C
C  Definitions of variables in FIT
C
C    DATA(1,i)  Observation used in fit
C    DATA(2,i)  Time of observation
C    DATA(3,i)  Type of observation
C                 1 = latitude
C                 2 = longitude
C                 3,7,... = range from radar (secondaries)
C                 4,8,... = azimuth from radar ( " - up from east)
C                 5,9,... = range from radar (primaries)
C                 6,10,...= azimuth from radar (primaries)
C    ND         Number of observations used in fit
C    APARAM     Array of potential solve parameters in fit
C                 1 = latitude at TSTRT (LAT0)
C                 2 = longitude at TSTRT (LON0)
C                 3 = flight path azimuth at TSTRT(AZFP0)
C                 6 = flight path azimuth rate (AZFPD)
C                 4 = time interval between range points along flight path
C                 5 =  start time (TSTRT)
C                 11-19 = radar azimuth bias for each radar (BARDR)
C                 21-29 = radar azimuth bias for each radar (BARDR)
C                 31-45 = range along path from TSTRT (RPATH)
C                 51-59 = time biases (TBIAS)
C                 61-75 = altitude at time intervals (ft) (ALT)
C    W(j)       Weighting put on type j (=DATA(3,i) observation
C    NDQ        Flag set to 1 to indicate diff/quot method used for partials
C    IR         Read file for LSQSET control parameters
C    IW         Write file for LSQSET output
C    TBIAS      Input value of time biases--added to input time
C    TBIASS     Time biases (adjusted from input value if solved for)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/ TOSET / DATA(3,3000), W(25), APARAM(100), ND, IR, IW, NDQ
         REAL*8 LATJFK, LONJFK, LATISP, LONISP, LATHPN, LONHPN
      COMMON/ OBSEND / LATIMP, LONIMP, TIMP, T0, TFLT
         REAL*8 LATIMP, LONIMP
C
      COMMON/ OBSRDR / RMAIN(20,9), AMAIN(20,9), TMAIN(20,9),
     <                 R0RDR(9), A0RDR(9), T0RDR(9),
     <                 MAIN(20,9), NMAIN(9)
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
         REAL*8  LATRDR, LONRDR, MAGOFF
      COMMON/ RDRDTA / TRDR(99,9), XRDR(99,9), YRDR(99,9),
     <                 ROBS(99,9), AOBS(99,9), TOBSN(199,9),
     <                 ROBSN(199,9), AOBSN(199,9), HOBSN(199,9),
     <                 NOBS(9), NOBSN(9)
      COMMON/ CTBIAS / TBIAS(9)
      COMMON/ CPATH / TPATH(20), NPATH, NPATHP, NPATHN
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
C
      REAL*8 LAT0, LON0
      DIMENSION RPATH(20), BARDR(9), BRRDR(9), TBIASS(9), ALT(20),
     <     RPATHP(15), RPATHN(5), ALTP(15), ALTN(5), AZFPD(2),
     <     AZFPD(2), NOBSNT(9)
C
      EQUIVALENCE ( APARAM(1),  LAT0 ),      ( APARAM(2),  LON0 ),
     <            ( APARAM(3),  AZFP0 ),     ( APARAM(4),  DT   ),
     <            ( APARAM(5),  TSTRT ),     ( APARAM(11), BARDR(1) ),
     <            ( APARAM(21), BRRDR(1) ), ( APARAM(31), RPATH(1) ),
     <            ( APARAM(51), TBIASS(1) ),( APARAM(61), ALT(1) ),
     <            ( APARAM(6),  AZFPD(1) ), ( APARAM(8),  AZFPN )
C                                                  (APARAM(9), ALTB) ✓
      IR = 20
      IW = 8
      NDQ = 1
      ND = 0
```

```
C
          CALL DMS(LAT,-LON,LATDEG,LATMIN,LATSEC,LONDEG,LONMIN,LONSEC)
          WRITE( IWOUT, 300 )  T, LATDEG, LATMIN, LATSEC,
     <      LONDEG, LONMIN, LONSEC
C         LON = -LON
C
C  Calculated position at radar observation times
C
          CALL  DXDY( DX, DY, T, RPATH, TPATH, AZFP0, AZFPD, AZFPN,
     <                NPATH )
          LATC = LAT0 + DY / 60.
          LONC = LON0 - DX / 60. / DCOS(LAT*CDR)
          LONC = -LONC
          WRITE( IWPLT, 500 )  LON, LAT, LONC, LATC
C
C    Print observed range and computed range interpolated along flt path
C
          CALL  DXDY( DX, DY, T, RPATH, TPATH, AZFP0, AZFPD, AZFPN,
     <                NPATH )
C
          LAT = LAT0 + DY / 60.
          LON = LON0 - DX / 60. / DCOS(LAT*CDR)
          ALTNM = RNTRP( T, ALT, TPATH, NPATH ) * CFTNM
C
          CALL  LLA2X( LAT, LON, ALTNM, X )
C
          CALL  X2RAZ( RNGC, AZC, EL, X, J )
C
          IF( J .EQ. 2 )  RNGC = RNGC + 25.
          IF( J .EQ. 4 )  RNGC = RNGC + 42.
          RMOBS = RTRUE
          IF( J .EQ. 2 )  RMOBS = RMOBS + 25.
          IF( J .EQ. 4 )  RMOBS = RMOBS + 42.
          WRITE( IWPLTR, 500 )  T, RMOBS, RNGC
C
 12     CONTINUE
C
 11   CONTINUE
C
C   Print observed and computed range data for all data
C
      CALL  PRRALL
C
      RETURN
C
 100  FORMAT( 2F10.2, I4, I3, I3, I4, I3, I3 )
 200  FORMAT( '      Range from initial point along flight path' /
     <        '   Time(s)  Dist(nm) Lat(d,m,s) Lon(d,m,s) ' )
 300  FORMAT( F10.2, I4, I3, I3, I4, I3, I3 )
 400  FORMAT( // '      Adjusted radar hit locations for ', A4 /
     <        '   Time(s) Lat(d,m,s) Lon(d,m,s) ' )
 500  FORMAT( 4F10.4 )
C
      END
C
C ************************************************************************
C
      SUBROUTINE  PRNODE
C
C  Print range along path at node time. Print lat and lon of best traj
C  at each node. Interpolate radar data to node times and print
C  position corresponding to interpolated R,A,H.
C
      IMPLICIT REAL*8(A-H,O-Z)

      COMMON/ OBSEND / LATIMP, LONIMP, TIMP, T0, TFLT
         REAL*8 LATIMP, LONIMP
C
      COMMON/ OBSRDR / RMAIN(20,9), AMAIN(20,9), TMAIN(20,9),
     <                 R0RDR(9), A0RDR(9), T0RDR(9),
     <                 MAIN(20,9), NMAIN(9)
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
         REAL*8  LATRDR, LONRDR, MAGOFF
      COMMON/ RDRDTA / TRDR(99,9), XRDR(99,9), YRDR(99,9),
     <                 ROBS(99,9), AOBS(99,9), TOBSN(199,9),
     <                 ROBSN(199,9), AOBSN(199,9), HOBSN(199,9),
     <                 NOBS(9), NOBSN(9)
C
      COMMON/ TOSET / DATA(3,3000), W(25), APARAM(100), ND, IR, IW, NDQ
```

```
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
         REAL*8  LATRDR, LONRDR, MAGOFF
      COMMON/ CTBIAS / TBIAS(9)
      COMMON/ RDRDTA / TRDR(99,9), XRDR(99,9), YRDR(99,9),
     <                 ROBS(99,9), AOBS(99,9), TOBSN(199,9),
     <                 ROBSN(199,9), AOBSN(199,9), HOBSN(199,9),
     <                 NOBS(9), NOBSN(9)
      COMMON/ XYREF / LATREF, LONREF
         REAL*8 LATREF, LONREF
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
C
      REAL*8 LAT0, LON0, LATCAL, LONCAL
      DIMENSION  D2RDR(9)
      CHARACTER*1 STAR, BLANK, MNOBJ
      DATA  STAR / '*' /, BLANK / ' ' /
C
      AZFP = 26.5
      DM = 1.5
C
C  Calculate latitude and longitude of initial point
C
      DMR = DM * CDR
      XON = RORDR(1) * DCOS( AORDR(1) + DMR )
      YON = RORDR(1) * DSIN( AORDR(1) + DMR )
      ROP = XON - YON / DTAN(AZFP*CDR)
      A0 = DATAN2(YON,XON)
      LAT0 = LATRDR(1) + YON / 60.
      LON0 = LONRDR(1) - XON / 60. / DCOS(LAT0*CDR)
C
      WRITE(8,401) AZFP, DM, LAT0, LON0
C
      K = 1
      DO  1  I = 1, NOBS(1)
C
C  Calculate position assuming fixed flight azimuth from initial position
C
        R = ROBS(I,1)
        BE = DASIN( DSIN(AZFP*CDR) * ROP / R )
        RR = R**2 + RORDR(1)**2 - 2.0*R*RORDR(1) * DCOS(AZFP*CDR-BE-A0)
        RR = DSQRT( RR )
        DX = RR * DCOS(AZFP*CDR)
        DY = RR * DSIN(AZFP*CDR)
        XCALC = XON + DX + XOFRDR(1,1)
        YCALC = YON + DY + XOFRDR(2,1)
        LATCAL = LATREF + YCALC / 60.
        LONCAL = LONREF - XCALC / 60. / DCOS(LATCAL*CDR)
C
        DO  2  J = 2, NRDRS
          D2RDR(J) = ( XOFRDR(1,J) - XCALC )**2
     <               + ( XOFRDR(2,J) - YCALC )**2
          D2RDR(J) = DSQRT(D2RDR(J))
 2      CONTINUE
C
        MNOBJ = BLANK
        IF( I .EQ. MAIN(K,1) )  THEN
          MNOBJ = STAR
          K = K + 1
        END IF
C
        T = TRDR(I,1) + TBIAS(1)
        WRITE(8,300) I, MNOBJ, T, R, RR, LATCAL, LONCAL,
     <               (D2RDR(J),J=2,NRDRS)
C
 1    CONTINUE
C
      RETURN
C
 300  FORMAT( I5, A1, F10.2, 7F10.4 )
 401  FORMAT(//T25, 'JFK Data' //
     <         T50, 'PARAMETERS USED' /
     <         T50, ' Flight azimuth =', F10.2 /
     <         T50, ' Magnetic offset =', F10.2 /
     <         T50, ' Lat-0 =', F10.4 /
     <         T50, ' Lon-0 =', F10.4 //
     <         '          Time     Range data  Range     --Calculated-- ',
     <         '   Distance  Distance  Distance' /
     <         '          (sec)      (nm)        fm pos-0   lat          lon',
     <         '     to ISP    to HPN    to RHD ' // )
C
```

```
C
C  Calculate sighting data for observer
C
        CALL  LLA2X( LAT, -LON, HNM, X )
        CALL  X2RAZ( RNGOBS, AZOBS, EL, X, 5 )
        AZOBS = 90. - AZOBS
        WRITE( 3, 500 )  T, RNGOBS, AZOBS, EL
C
 1    CONTINUE
C
      RETURN
C
 100  FORMAT( 2F10.2, I4, I3, I3, I4, I3, I3 )
 200  FORMAT( '      Range from initial point along flight path' /
     <        '   Time(s)  Dist(nm) Lat(d,m,s) Lon(d,m,s) ' )
 500  FORMAT( 4F10.4 )
C
      END
C
C ******************************************************************
C
      SUBROUTINE  PRRALL
C
C  Print computed range vs time and observed range vs time for all data
C
      IMPLICIT REAL*8(A-H,O-Z)

      COMMON/ OBSEND / LATIMP, LONIMP, TIMP, T0, TFLT
         REAL*8 LATIMP, LONIMP
C
      COMMON/ OBSRDR / RMAIN(20,9), AMAIN(20,9), TMAIN(20,9),
     <                 RORDR(9), AORDR(9), TORDR(9),
     <                 MAIN(20,9), NMAIN(9)
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
         REAL*8  LATRDR, LONRDR, MAGOFF
      COMMON/ RDRDTA / TRDR(99,9), XRDR(99,9), YRDR(99,9),
     <                 ROBS(99,9), AOBS(99,9), TOBSN(199,9),
     <                 ROBSN(199,9), AOBSN(199,9), HOBSN(199,9),
     <                 NOBS(9), NOBSN(9)
C
      COMMON/ TOSET / DATA(3,3000), W(25), APARAM(100), ND, IR, IW, NDQ
      COMMON/ XYREF / LATREF, LONREF
         REAL*8 LATREF, LONREF
      COMMON/ CPATH / TPATH(20), NPATH, NPATHP, NPATHN
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
C
      EQUIVALENCE ( APARAM(1),  LAT0 ),      ( APARAM(2),  LON0 ),
     <            ( APARAM(3),  AZFP0 ),     ( APARAM(4),  DT    ),
     <            ( APARAM(5),  TSTRT ),     ( APARAM(11), BARDR(1) ),
     <            ( APARAM(21), BRRDR(1) ),  ( APARAM(31), RPATH(1) ),
     <            ( APARAM(51), TBIASS(1) ),( APARAM(61), ALT(1) ),
     <            ( APARAM(6),  AZFPD(1) ),  ( APARAM(8),  AZFPN )
      DIMENSION RPATH(20), BARDR(9), BRRDR(9), TBIASS(9), ALT(20),
     <          X(3), AZFPD(2)
      DIMENSION TMAINJ(21), RMAINJ(21), AMAINJ(21)
      REAL*8 LAT, LON, LAT0, LON0, LATC, LONC
      CHARACTER*4 CRDR(9)
      DATA  CRDR / ' JFK', ' ISP', ' HPN', ' RHD', 5*'    ' /
C
      IWPLA0 = 70
C
      DO  22  J = 1, NRDRS
        IWPLRA = IWPLA0 + J
        DO  21  I = 1, NOBS(J)
          T = TRDR(I,J) + TBIASS(J) - TSTRT
          CALL  DXDY( DX, DY, T, RPATH, TPATH, AZFP0, AZFPD, AZFPN,
     <                NPATH )
C
          LAT = LAT0 + DY / 60.
          LON = LON0 - DX / 60. / DCOS(LAT*CDR)
          ALTNM = RNTRP( T, ALT, TPATH, NPATH ) * CFTNM
C
          CALL  LLA2X( LAT, LON, ALTNM, X )
C
          CALL  X2RAZ( RNGC, AZC, EL, X, J )
C
          RNGO = ROBS(I,J) - BRRDR(J)
          WRITE( IWPLRA, 500 )  T, RNGO, RNGC
C
```

/CFTNM

Print all tie ...
SIGB = 162.
HP, SIG = ALTTNM(T, ALT, ALTB + SIGB,
HM, SIG = ALTTNM(T, ALT, ALTB - SIGB
WRITE(4, 500) T, HFT, HP, SIG, HM, SIG

alt
ALTNM = ALTTNM(T, ALT, ALTB)

```
C     YODRDR   Y positions of radar in refernce coordinate system
C     ROBS     Range measurements (nm)
C     AOBS     Azimuth measurements (nm)
C     NOBS     Number of xy pairs for each radar
C     IRRDR    Read files for xy data
C
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
         REAL*8  LATRDR, LONRDR, MAGOFF
      COMMON/ RDRDTA / TRDR(99,9), XRDR(99,9), YRDR(99,9),
     <                 ROBS(99,9), AOBS(99,9), TOBSN(199,9),
     <                 ROBSN(199,9), AOBSN(199,9), HOBSN(199,9),
     <                 NOBS(9), NOBSN(9)
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
      COMMON/ CTBIAS / TBIAS(9)
C
      CHARACTER*4 ALP(2)
      DIMENSION  IRRDR(9), IRRDRN(9)
C
C
C  Set read file units
C
      IRRDR(1) = 21
      IRRDR(2) = 22
      IRRDR(3) = 33
      IRRDR(4) = 14
      IRRDRN(1) = 16
      IRRDRN(2) = 17
      IRRDRN(3) = 18
      IRRDRN(4) = 19
C
C  Read rng. az data
C
      DO  3  J = 1, NRDRS
        READ(IRRDR(J),200) ALP(2)
        DO  1  I = 1, 99
          READ( IRRDR(J), 100, ERR=2 )  TRDR(I,J), X, Y
          ROBS(I,J) = X
          AOBS(I,J) = ( 90. - ( Y + MAGOFF(J) ) ) * CDR
 1      CONTINUE
 2      NOBS(J) = I - 1
 3    CONTINUE
C
C  Read files with pre-event data
C
      DO  6  J = 1, NRDRS
        READ( IRRDRN(J), 200 )  ALP(2)
        DO  4  I = 1, 199
          READ( IRRDRN(J), 100, ERR=5 )  T, ROBSN(I,J), Y, HOBSN(I,J)
          IF( J .NE. 4 )  T = T - TBIAS(J)
          IF( T .GT. 100 )  TOBSN(I,J) = T - 160.
          IF( T .LE. 100 )  TOBSN(I,J) = T
          IF( J .NE. 4 )  Y = Y * 360. / 4096.
          AOBSN(I,J) = ( 90. - ( Y + MAGOFF(J) ) ) * CDR
 4      CONTINUE
 5      NOBSN(J) = I - 1
 6    CONTINUE
C
      RETURN
C
 100  FORMAT( F10.2, 3F10.0 )
 200  FORMAT( A4/A4 )
C
      END
C
C  ***************************************************************
C
      SUBROUTINE  STMAIN
C
C  Store data for points previously selected as main object
C
C  Parameters used in STMAIN
C
C    NOBS     Number of data pairs in each radar's read file
C    MAIN     Indices of data selected as main object
C    RMAIN    Array containing range from radar to main object
C    AMAIN    Array containing azimuth from radar to main object
C               data (measured up from horizontal)
C    TMAIN    Array containing times of main object data
C    NMAIN    Number of main object data points in JFK data
```

```
      END IF
C
      RETURN
      END
C
  ***********************************************************************

      SUBROUTINE  DXDY( DX, DY, T, RPATH, TPATH, AZFP0, AZFPD, AZFPN,
     <                  NPATH )
C
C  Calculate east/north displacement from initial position
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
      DIMENSION  RPATH(15), TPATH(15), AZFPD(2)
C
C
      AZFP = AZFPN * CDR
      DX = RPATH(1) * DCOS( AZFP )
      DY = RPATH(1) * DSIN( AZFP )
C
      DO  1  I = 1, NPATH - 1
        TP = TPATH(I)
        IF( TPATH(I) .GE. 0.  .and. tpath(i).lt.35.  )
     <      AZFP = ( AZFP0 + AZFPD(1)*TP + AZFPD(2)*TP**2 ) * CDR
        IF( T .LE. TPATH(I+1) )  GO TO  2
        DX = DX + ( RPATH(I+1) - RPATH(I) ) * DCOS( AZFP )
        DY = DY + ( RPATH(I+1) - RPATH(I) ) * DSIN( AZFP )
 1    CONTINUE
 2    CONTINUE
C
      DR = ( RPATH(I+1) - RPATH(I) ) * ( T - TPATH(I) )
     <     / ( TPATH(I+1) - TPATH(I) )
      DX = DX + DR * DCOS(AZFP)
      DY = DY + DR * DSIN(AZFP)
C
C
      RETURN
      END
C
C ***********************************************************************
C
      SUBROUTINE  X2RAZ( RNG, AZ, EL, X, KRDR )
C
C  Convert cartesian position to range and azimuth from a radar
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
         REAL*8  LATRDR, LONRDR, MAGOFF
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
      DIMENSION  X(3), XT(3), XTT(3)
C
      CUN2NM = 1.
      DO  1  I = 1, 3
        XT(I) = X(I) - XOFRDR(I,KRDR)
 1    CONTINUE
C
      IF( FEORSE .EQ.1 )  THEN
        CALL  ROT( XT, XTT, (-LONRDR(KRDR)+90.)*CDR, 3 )
        CALL  ROT( XTT, XT, (90.-LATRDR(KRDR))*CDR, 1 )
        CUN2NM = 1. / CNMM
      END IF
C
      RNG = DSQRT( XT(1)**2 + XT(2)**2 + XT(3)**2 ) * CUN2NM
      AZ = DATAN2( XT(2), XT(1) ) / CDR
      EL = DASIN( XT(3) / ( RNG / CUN2NM ) ) / CDR
C
      RETURN
      END
C
C ***********************************************************************
C
      SUBROUTINE  ROT  (X,Y,TH,N)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(3), X(3)
C
      CT = DCOS( TH )
      ST = DSIN( TH )
```

```
C  Print range data from other radars
C
      DO  11  J = 3, NRDRS

        WRITE(8,600)  CRDR(J), CRDR(J)
C
        K = 1
        DO  12  I = 1, NOBS(J)
          MNOBJ = BLANK
          IF( I .EQ. MAIN(K,J) )  THEN
            MNOBJ = STAR
            K = K + 1
          END IF
          WRITE(8,300)  I, MNOBJ, TRDR(I,J), ROBS(I,J)
 12     CONTINUE
C
 11   CONTINUE
C
C  Setup for least squares fit
C
      CALL  FIT
C
C  Print final results from fit
C
      CALL  OUTPUT
C
      STOP
C
 300  FORMAT( I5, A1, F10.2, 6F10.4 )
 400  FORMAT( /T20, 'Program to Analyze x,y Data From Radars' )
 600  FORMAT( /// T25, A4, ' Data ' //
     <   '            Time  Rng fm', A4 )
C
      END
C
C  *******************************************************************
C
      SUBROUTINE  PARAMS
C
C  Set constant parameters
C
C  Definition of variables used
C
C    NRDRS    Number of radars
C    LATREF   Latitude of reference coordinate system
C    LONREF   Longitude of reference coordinate system
C    LATRDR   Latitude of radars
C    LONRDR   Longitude of radars
C    XOFRDR   Positions of radars in REF coordinates
C    R0RDR    Distance to last transponded (JFK)
C    A0RDR    Az from JFK to last transponded as measured
C    X0RDR    X position at last transponded in REF cs (nm)
C    Y0RDR    Y position at last JFK transponded in REF cs (nm)
C    T0RDR    Times of last transponded point (sec from even minute)
C    LATIMP   Impact latitude
C    LONIMP   Impact longitude
C    T0       Time of explosion (sec from even minute)
C    TFLT     Flight duration after explosion
C    TIMP     Time of impact (sec from even minute)
C    TBIAS    Time biases (sec added to input data times)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON/ OBSEND / LATIMP, LONIMP, TIMP, T0, TFLT
         REAL*8 LATIMP, LONIMP
C
      COMMON/ OBSRDR / RMAIN(20,9), AMAIN(20,9), TMAIN(20,9),
     <                 R0RDR(9), A0RDR(9), T0RDR(9),
     <                 MAIN(20,9), NMAIN(9)
C
      COMMON/ LOCRDR / LATRDR(9), LONRDR(9), ALTRDR(9), XOFRDR(3,9),
     <                 MAGOFF(9), NRDRS
         REAL*8  LATRDR, LONRDR, MAGOFF
C
      COMMON/ XYREF / LATREF, LONREF
         REAL*8 LATREF, LONREF
C
      COMMON/ CTBIAS / TBIAS(9)
C
      COMMON/ CONST / CDR, CFTNM, CNMM, CFM, RE, FEORSE
```

6 Jan 98

xy2rng.run.s

```
 #! /bin/csh -x
# C-shell script to run xy2rng.f
# echo "$1"
#if ($#argv == 0) then
#    echo 'Enter input filename:'
     set input_fn = $<
#    if ($input_fn == '') then
#       exit
#    endif
#else
#    set input_fn = $argv[1]
#endif
#echo "xy2rng is running $input_fn"
set input1_fn = faa_jfk1.dat
set input2_fn = faa_isp1.dat
set input3_fn = faa_hpn1.dat
set input4_fn = xy2rng.lsq.inp
set input5_fn = sen3.xyz
set input6_fn = nvy44.dat
set input7_fn = ntsb.jfk
set input8_fn = ntsb.isp
set input9_fn = ntsb.hpn
mkdir output
cp $input4_fn ./output/fort.20
cp $input1_fn ./output/fort.11
cp $input7_fn ./output/fort.21
cp $input2_fn ./output/fort.12
cp $input8_fn ./output/fort.22
cp $input3_fn ./output/fort.13
cp $input5_fn ./output/fort.23
cp $input9_fn ./output/fort.33
cp $input6_fn ./output/fort.14
cd output
  ../xy2rng
cd ..
exit
```