U.S. DEPARTMENT OF THE INTERIOR

U.S. GEOLOGICAL SURVEY

Estimating Trajectories of Supersonic Objects using arrival Times of Sonic Booms

by
Jim Mori[1]
and
Hiroo Kanamori[2]

Open-File Report 91-48

[1] U. S. Geological Survey
525 So. Wilson Avenue
Pasadena, Ca., 91106

[2] Seismological Lab
California Institute of Technology
Pasadena, Ca., 91125

1991

ABSTRACT

A computer program was written to determine the trajectory of supersonic aircraft from the arrival times of the sonic boom. The program was used to estimate the flight path, velocity and height of eight space shuttle landings and one flight of the SR-71 Blackbird plane, which were recorded on the southern California seismic network during 1989 and 1990. It was found that the direction of the flight path can be resolved fairly accurately, while there is much more uncertainty in the velocity and height estimates.

INTRODUCTION

Sonic booms from aircraft and possibly meteors (Anglin and Haddon, 1988) traveling at supersonic velocities are sometimes recorded on short-period seismic instruments. In southern California, the space shuttles on their way to landing at Edwards Air Force Base cause particularly strong shock waves (Kanamori et al., 1991) that are recorded on the southern California seismic network. Differences between the signals recorded for earthquakes and sonic booms can be seen in both waveforms and the pattern of arrival times. Supersonic aircraft often produce the characteristic "N" shaped pressure waves (Fig. 1), where the width of the "N" depends on such factors as the plane length, velocity and height (Carlson and Maglieri, 1972). The waveforms in Fig. 1 are "backward N's" because positive pressure from the shock wave causes downward ground motion. The arrival times of these waves can be picked to an accuracy of a few tenths of seconds. The pattern of arrival times form hypobolic shaped isochrons, as opposed to the pattern of concentric circles produced by earthquakes. In order to confidently identify sonic booms, it is useful to have a computer program to quickly determine if the pattern of short-period arrivals is consistent with the characteristic hyperbolas.

SONIC BOOM PROPAGATION

Objects moving at supersonic speed (U) through the air produce cone-shaped shock wave fronts (Fig. 2). The slope of the cone (β)is given by

$$sin\beta = \frac{1}{M} \tag{1}$$

where M is the Mach number,

$$M = \frac{U}{c} \tag{2}$$

and c is the speed of sound in the air, in this case assumed to be 300 m/sec. For an object flying above and parallel to the x axis as in Fig. 2, the intersection of the Mach cone with

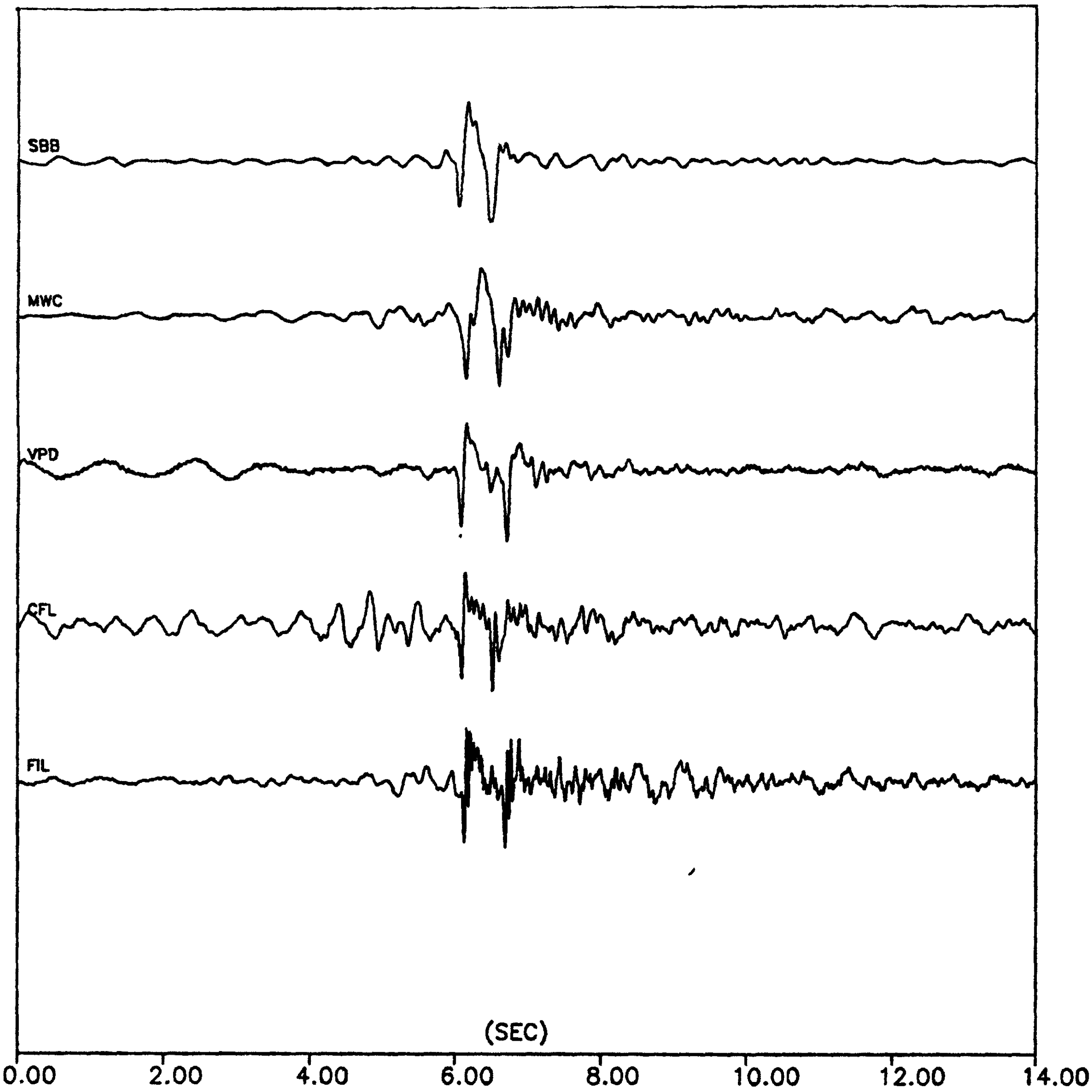

Fig. 1. Shock wave from the 8/13/89 space shuttle flight recorded on short-period instruments of the southern California seismic network. The double peak waveform is the characteristic N wave of the sonic boom. The traces have been normalized to the peak amplitudes. The actual ground displacements are a few tenths of a micron.

the ground surface forms a hyperbola,

$$\frac{x^2}{D^2} - \frac{y^2}{(\beta D)^2} = 1 \tag{3}$$

with assymptotes,

$$y = \pm \beta x. \tag{4}$$

The distance from the apex of the hyperbola to the interesection of the assymptotes is given by,

$$D = \frac{H}{tan\beta} \tag{5}$$

where H is the height of the flying object. Using equation 3, the arrival time (t) of the

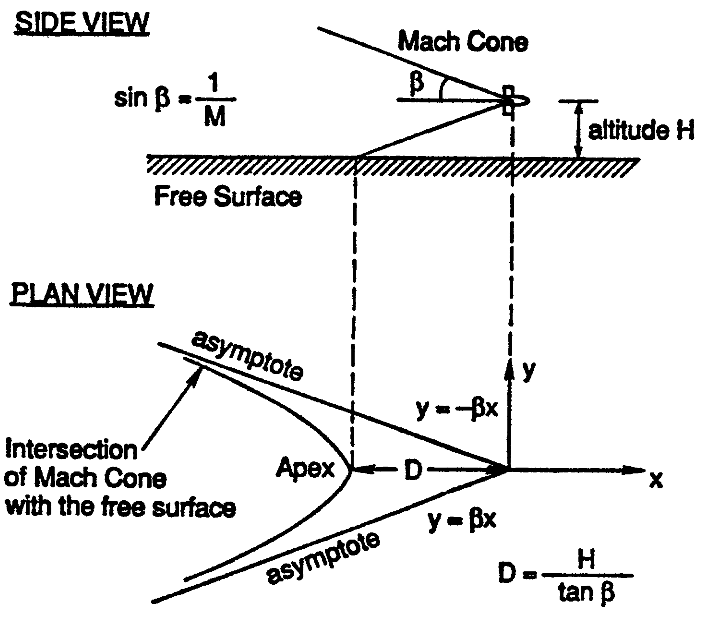

Fig. 2. Diagrams of the geometry of a Mach cone formed by an object traveling at Mach number M and height H.

shock wave at some point (x,y) on the ground is given by,

$$t(x,y) = \frac{(x + \sqrt{1 + \frac{y^2}{(\beta D)^2}} * D + D)}{U} + t_0 \tag{6}$$

where t_0 is the arrival time of the shock wave at the origin.

For a constant velocity and constant height Mach cone, the hyperbolic isochrons of the shock wave moving across the ground surface can be described by 5 parameters: height, speed, trajectory (specified by a back azimuth and a lateral offset, as shown in Fig. 3) and an origin time. A program was written to systematically search through combinations of the first four parameters and solve for the origin time, to find the set that produces arrival times that best match the data.

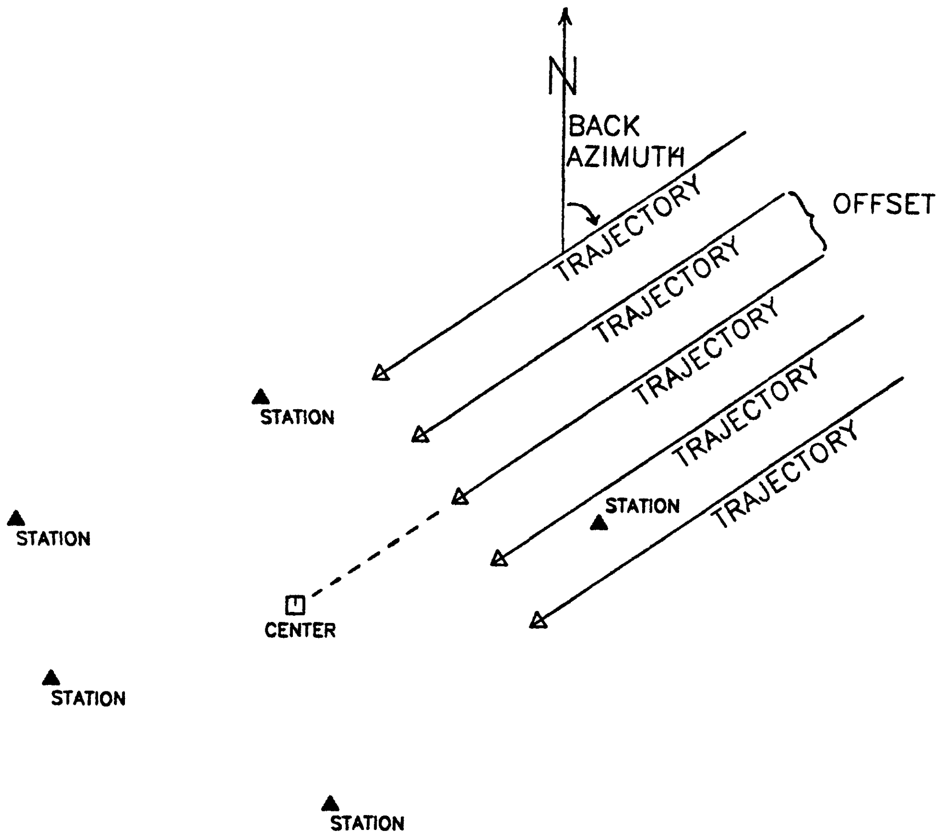

Fig. 3. Diagram of the parameters (back azimuth and offset) used to specify the flight trajectory.

EXAMPLES

Figs. 4 shows examples of arrival times from four strong sonic booms recorded by the Southern California Seismic Network. They include three space shuttle landings and a flight of the SR-71 Blackbird plane. The arrival times are plotted on a map along with the best fitting trajectory determined by our program. The trajectory is plotted by the hyperbolic isochrons at 20 sec intervals. The parameters for these and four other sonic booms recorded during 1989 and 1990 are given in Table 1.

Date	Time	No. Sta.	Offset Point	Back Azimuth	Height (km)	Mach No.	RMS (sec)
SR-71 Blackbird							
3/06/90	1406	15	34.42°N 118.55°W	249°	21	2.6	3.5
Space Shuttles							
8/13/89	1331	26	34.05°N 118.42°W,	204°	28	2.7	7.7
10/23/89	1628	20	34.17°N 118.69°W	213°	24	3.5	6.5
11/28/89	0025	22	34.39°N 119.67°W	241°	23	3.5	10.9
1/20/90	0931	22	34.15°N 119.63°W	236°	34	4.1	7.9
3/04/90	1805	19	35.84°N 118.34°W	329°	28	2.6	5.0
4/29/90	1343	23	34.50°N 119.37°W	246°	27	3.6	8.1
10/10/90	1345	27	34.35°N 120.10°W	249°	31	3.8	6.0
12/11/90	0547	28	34.53°N 119.59°W	252°	31	4.3	11.0

Table 1. Parameters estimated for sonic booms recorded on the southern California Seismic Network. The offset is an arbitrary point used along with the back azimuth to specify the flight path.

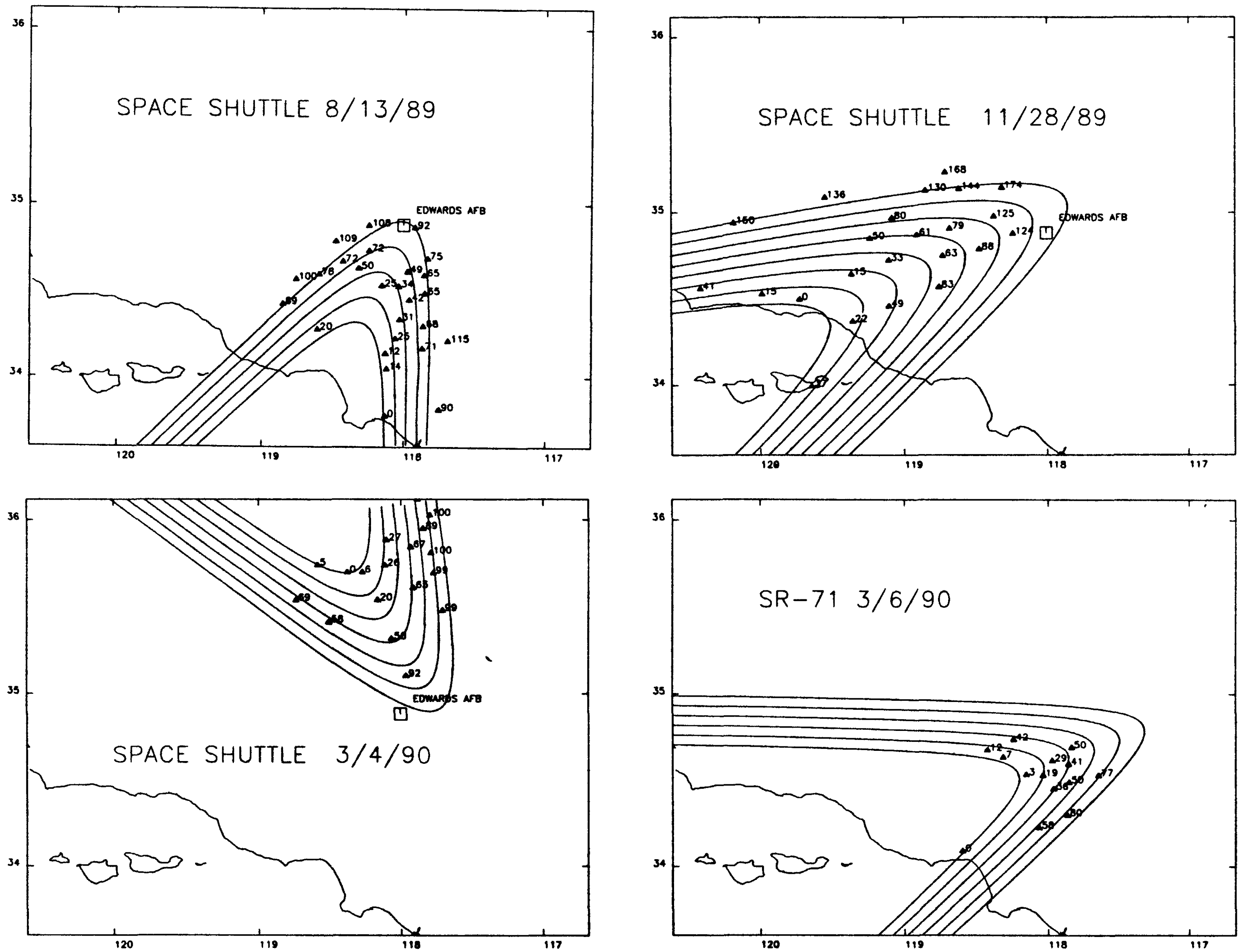

Fig. 4. Arrival time data (sec) and estimated flight paths for aircraft that caused strong sonic booms. The flight path is shown by hyperbolas which represent 20 sec isochrons. For the space shuttle cases, the square marks the landing site at Edwards Air Force Base.

The root mean square (RMS) errors between the model and observed arrival times were 3 to 11 sec for the various sonic booms. The smallest error is for the Blackbird flight which was probably a relatively level and constant velocity flight over the network. The largest error was for the 11/28/89 space shuttle landing which was recorded across about 150 km of the network during which the aircraft's height and velocity probably changed substantially. These errors are relatively large, compared to residuals in earthquake locations because of the slow speed of sound in the air (300 m/sec). One large source of error is caused by the difference in heights of the stations, for which corrections were not made. An elevation difference of 1 to 2 kilometers will cause arrival time differences of several seconds. A second source of error for the cases of the space shuttle landings, is that the aircriaft is slowing down and dropping in height as it comes over the network. In these cases the estimated velocities and heights would represent average values of the flight path over the network. A third source of error is the variation of the sound velocity at different heights in the atmosphere. We are working on a modified version of the program which will take into account these additional factors.

In general, the direction of the flight path can be well resolved if the spread of stations is large enough to resolve the curvature in the hyperbolic isochrons. Using the data from the sonic boom caused by the SR-71 Blackbird flight of 3/6/90, Fig. 5 shows how the RMS varies as a function of each of the four tested parameters, while keeping all the remaining parameters fixed. The sharp minima in the back azimuth and offset indicate that the trajectory can be resolved within a few degrees azimuth and a few kilometers offset.

If the object is moving at a fairly constant velocity, as was probably the case of the SR-71 flight, the velocity can also be estimated fairly accurately. The width of the minimum in the plot of RMS as a function of Mach number indicates that the velocity can be estimated within a few tenths of a Mach number.

The height is the most uncertain parameter, because stations have to be located close to the apex of the hyperbolic isochrons in order to resolve this parameter. Fig. 5 shows a much broader minimum in the variation of RMS as a function of height, indicating that the average height is resolved to only 5 to 10 km.

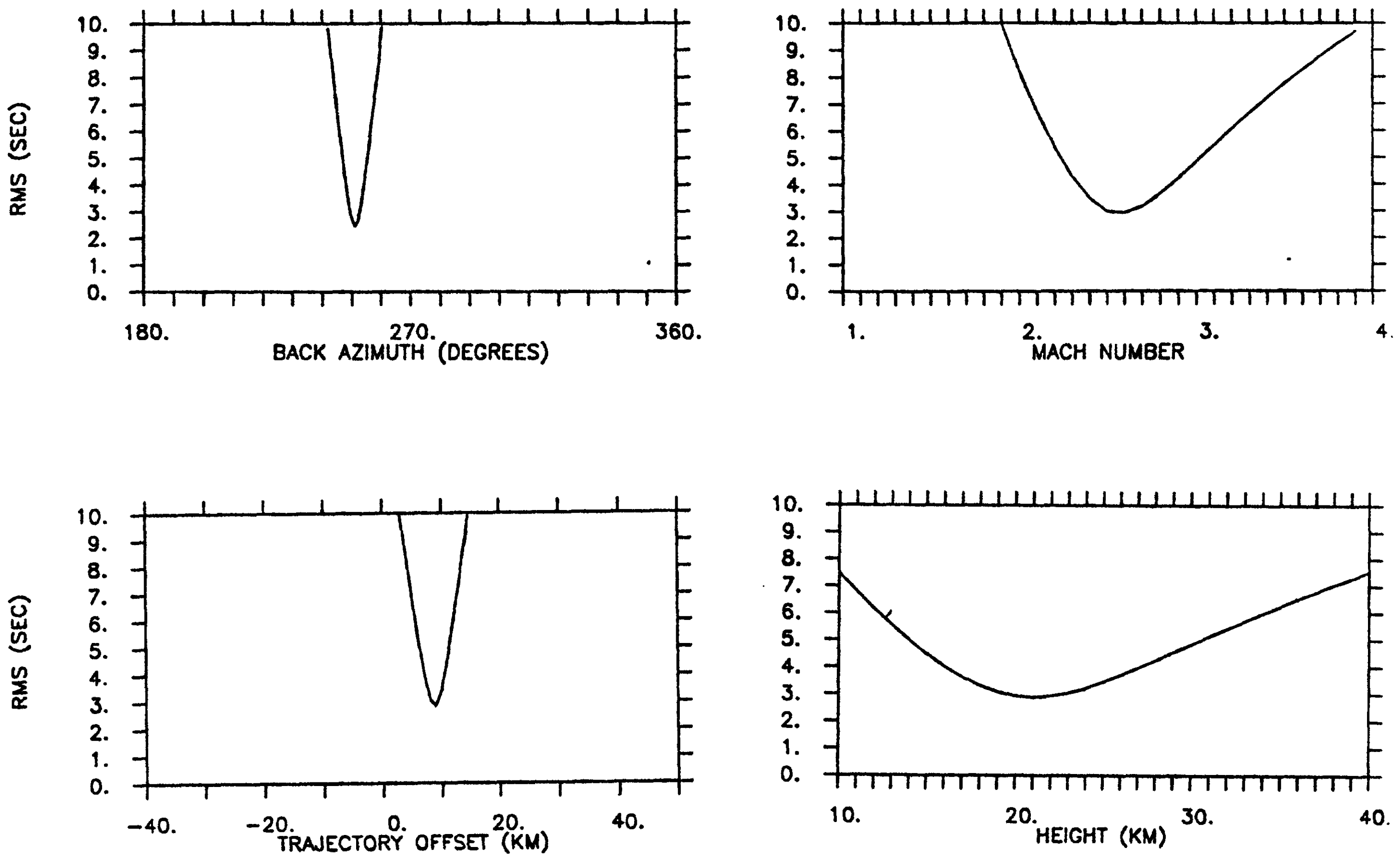

Fig. 5. Variation of the RMS error as a function of each of the four parameters used to estimate the flight paths. In each case, the plot shows the values of the RMS errors when one parameter is tested, while holding the other three fixed. This example uses data from the 3/6/90 SR-71 sonic boom.

ACKNOWLEDGEMENTS

We thank T. Heaton for helpful discussion and comments on this report.

REFERENCES

Anglin, F.M. and R.A.W. Haddon, 1988, What was that?, Earthquakes and Volcanoes, 20, p.161.

Carlson, H.W. and D. J. Maglieri, 1972, Review of sonic-boom generation theory and prediction methods, J. Acoiustic Soc. Amer., 51, p. 675.

Kanamori, H., J. Mori, D. Anderson, and T. Heaton, 1991, Seismic excitation by the space shuttle Columbia, submitted to Nature.

Klein, F.W., 1985, User's guide to HYPOINVERSE, a program for VAX and PC350 computers to solve for earthquake locations, U.S. Geol. Survey Open-File Report 85-515, 24 pp.

APPENDIX

FORTRAN source for BOOMFIT, a program to estimate supersonic trajectories from sonic boom arrival times.

Input:

Coordinates of approximate center of recording stations.

Approximate radius (km) of recording stations.

File with station coordinates in HYPOINVERSE format (Klein, 1985) named ALLSTN.DCK

File with arrival time data in HYPOINVERSE format.

Input ranges of the Mach number, height, and the back azimuth and offset of the trajectory are specified by three values: starting value, number of values to test and the increment of the tested values. All of these parameters are fairly straight forward with the possible exception of the lateral offset in the trajectory. The offset is specified by points along a line perpendicular to the azimuthal line to the approximate center of the stations (Fig. 3). The program asks for the number (n) and spacing of points to test on either side of the azimuthal line. 2n + 1 pathes are tested for each back azimuth.

Output:

Results are printed in the file BOOMFIT.OUT.

```
C  PROGRAM TO FIT SUPER SONIC TRAJECTORIES TO SONIC BOOM ARRIVAL TIMES
C

       PARAMETER(NSTAP=500, NARRP=200, NAZP = 100, NPTP = 50)


C      NSTAP = NO. OF STATIONS IN STATION LIST
C      NARRP = NO. OF ARRIVALS
C      NAZP  = NO. OF AZIMUTHS TO TEST
C      NPTP  = NO. OF STARTING POINTS TO TEST


       DIMENSION TOR(NARRP), TIME(NARRP), XTREJ(NARRP)
       DIMENSION ILAT(NSTAP), FLATM(NSTAP), ILONG(NSTAP), FLONGM(NSTAP)
       DIMENSION STLAT(NARRP), STLON(NARRP)
       DIMENSION CALTIM1(NARRP), CALTIM2(NARRP)
       CHARACTER*4 STAN(NSTAP),STA(NARRP), CTIME
       CHARACTER*7 CLAT, CLON
       CHARACTER*72 TITLE,FILE1
C
C OPEN FILE FOR OUTPUT
C

        OPEN(7,FILE='BOOMFIT.OUT',STATUS='NEW',CARRIAGECONTROL='LIST')
C
C OPEN FILE AND READ STATION COORDINATES
C
       OPEN(4,FILE='ALLSTN.DCK', STATUS='OLD',SHARED)

       JJ=1
       DO 8 J=1,500
        READ(4,907,END=9)STAN(J),ILAT(J),FLATM(J),ILONG(J),FLONGM(J)
        JJ=JJ + 1
   8   CONTINUE

   9   NSTATION= JJ -1
C
C OPEN AND READ FILE WITH DATA
C
       WRITE(*,*)' ENTER NAME OF FILE WITH DATA'
       READ(*,900)FILE1
       OPEN(3,FILE=FILE1,STATUS='OLD',SHARED)

       TIMMIN = 1.0E10
       DO 10 I=1,200
         READ(3,911,END=11)STA(I),IMIN,ISEC
         TIME(I) = FLOAT(IMIN*60) + FLOAT(ISEC/100)
         IF(TIME(I).LT.TIMMIN) TIMMIN = TIME(I)
  10   CONTINUE
  11   NSTA = I - 1
       CLOSE(3)

       WRITE(*,*) ' '
       WRITE(*,*)'  ARRIVAL TIME DATA'
       WRITE(*,*) ' '

       DO 12 I=1,NSTA
         TIME(I) = TIME(I) - TIMMIN
         WRITE(*,*)STA(I), TIME(I)
  12   CONTINUE
C
C INPUT SEARCH PARAMETERS
C
       WRITE(*,*)' ENTER LAT AND LONG OF APPROXIMATE CENTER'
       WRITE(*,*)'  OF RECORDING STATIONS'
       WRITE(*,*)'   (DECIMAL DEGREES, WEST IS POSITIVE)'
       READ(*,*) CELAT, CELONG
       CELONG = -1.0*CELONG
       WRITE(*,*)' ENTER APPROXIMATE RADIUS OF NETWORK'
       READ(*,*)RADNET

       WRITE(*,*)' ENTER STARTING MACH NUMBER, NO. AND INTERVAL'
       READ(*,*)FMACH1, NMACH, DMACH
       WRITE(*,*)' ENTER STARTING HEIGHT, NO. AND INTERVAL'
       READ(*,*)HGT1, NHGT, DHGT
       WRITE(*,*)' ENTER STARTING AZIMUTH, NO. AND INTERVAL'
       READ(*,*)SAZ,NAZ,DAZ
       WRITE(*,*)' ENTER NO. OF PATHS TO TEST ON EACH SIDE OF AZIMUTH'
       READ(*,*)NPT
       WRITE(*,*)' ENTER SPACING OF PATHS (KM)'
```

```
            READ(*,*)DPT
C
C PUT STATION INFORMATION INTO ARRAY
C
 70         DO 75 ISTA=1,NSTA
               DO 72 II=1,NSTATION
                  IF(STAN(II).NE.STA(ISTA)) GOTO 72
                  STLAT(ISTA) = ILAT(II) + FLATM(II)/60.0
                  STLON(ISTA) = ILONG(II) + FLONGM(II)/60.0
                  STLON(ISTA)=-1.0 * STLON(ISTA)
                  GOTO 75
 72            CONTINUE

               WRITE(*,909)STA(ISTA)

 75         CONTINUE


C
C  LOOP OVER MACH NUMBERS
C
            RMSMIN = 1.0E10
            DO 530 IMACH = 1, NMACH
              FMACH = (IMACH-1)*DMACH + FMACH1
              WRITE(*,705)FMACH
C
C LOOP OVER HEIGHTS
C
              DO 520 IHGT=1, NHGT
                HGT= (IHGT-1)*DHGT + HGT1
                WRITE(*,704)HGT
C
                BETA=ASIN(1.0/FMACH)
                D=HGT/TAN(BETA)
C  HYPERBOLA CONSTANTS
                A=D
                B=BETA*A

C  LOOP OVER AZIMUTHS
                DO 500 IAZ=1,NAZ
                  AZ = SAZ + (IAZ-1)* DAZ
                  ANG = 270.- SAZ  - ((IAZ-1)*DAZ)
                  SAZ1 = SAZ + (IAZ-1)*DAZ
                  ANG = 3.1416*ANG/180.

C  LOOP OVER OFFSET POINTS
                  NPT1 = -1*NPT
                  DO 400 IPT= NPT1,NPT

                    XORKM= -1.0*COS(ANG)*RADNET - SIN(ANG)*(IPT*DPT)
                    YORKM= -1.0*SIN(ANG)*RADNET + COS(ANG)*(IPT*DPT)

                    CALL KM_DEG(CELAT,CELONG, XORKM,YORKM,YOR,XOR)

C  GET STATION COORDINATES

                    DO 300 ISTA=1,NSTA

                      CALL DEG_KM(STLAT(ISTA), STLON(ISTA),YOR,XOR,XSTA,YSTA)

C  ROTATE COORDINATES

                      XNEW = XSTA*COS(ANG) + YSTA*SIN(ANG)
                      YNEW = YSTA*COS(ANG) - XSTA*SIN(ANG)


C  CALCULATE EQUIVALENT X VALUE ALONG TRAJECTORY

                      XTREJ(ISTA) = XNEW + SQRT((1.0 + (YNEW/B)**2) * A**2) - D
                      TOR(ISTA) = TIME(ISTA) - XTREJ(ISTA)/(FMACH*0.30)

 300                CONTINUE

C  CALCULATE ORIGIN TIME

                    THIS = 0.
                    DO 310 ISTA=1,NSTA
                      THIS=THIS + TOR(ISTA)
 310                CONTINUE
                    TORG = THIS/NSTA

```

```
C  CALCULATE RMS

                   RMS = 0.
                   DO 320 ISTA=1,NSTA
                     CALTIME=TORG + XTREJ(ISTA)/(FMACH*0.33)
                     CALTIM1(ISTA) = CALTIME
                     RMS = RMS + (CALTIME - TIME(ISTA))**2
 320               CONTINUE

                   RMS = (RMS/NSTA)**0.5


                   IF(RMS.LT.RMSMIN) THEN
                     RMSMIN = RMS
                     AZMIN = SAZ + (IAZ-1)* DAZ
                     YORMIN = YOR
                     XORMIN = XOR
                     TORGMIN = TORG
                     FMACHM = FMACH
                     HGTMIN = HGT
                     AMIN = A
                     BMIN = B
                     DO 322 ISTA = 1,NSTA
                       CALTIM2(ISTA) = CALTIM1(ISTA)
 322                 CONTINUE
                   ENDIF


 400             CONTINUE

 500           CONTINUE

 520         CONTINUE

 530       CONTINUE

C
C    END OF LOOPS
C
           WRITE(*,706)YORMIN,XORMIN, AZMIN
           WRITE(*,707)RMSMIN
           WRITE(*,708)TORGMIN
           WRITE(*,709)FMACHM, HGTMIN

           WRITE(7,*)' '
           WRITE(7,*)' '
           WRITE(7,706)YORMIN,XORMIN, AZMIN
           WRITE(7,707)RMSMIN
           WRITE(7,708)TORGMIN
           WRITE(7,709)FMACHM, HGTMIN

           WRITE(7,*)' '
           WRITE(7,*)' '
           WRITE(7,*)'  STATION   ARR. TIME    PRED. TIME   RESIDUAL'
           WRITE(7,*)' '
           DO 532 ISTA=1,NSTA
             DIFF = TIME(ISTA) - CALTIM2(ISTA)
             WRITE(7,710)STA(ISTA), TIME(ISTA), CALTIM2(ISTA),DIFF
 532       CONTINUE

           STOP

 700       FORMAT(' BETA,D,A,B', 4E12.3)
 701       FORMAT(' XSTA, XNEW, YNEW,XTREJ  ', 4E12.4)
 702       FORMAT(' TOR, TIME  ', 2E12.4)
 704       FORMAT('  HEIGHT ',F6.2)
 705       FORMAT('  MACH NO. ',F6.2)
 706       FORMAT('  STARTING PT:', 2F8.2, '  AZIMUTH: ', F4.0)
 707       FORMAT('     RMS: ', F10.2, ' SEC')
 708       FORMAT('  ORIGIN TIME AT STARTING POINT ', F7.2)
 709       FORMAT(' MACH NO.: ', F5.2, '   HEIGHT: ', F5.2)
 710       FORMAT(1X,A3,7X,3F12.2)

 900       FORMAT(A72)
 904       FORMAT(A1)
 907       FORMAT(A4,1X,I2,1X,F5.2,1X,I3,1X,F5.2)
 909       FORMAT(1X, A4,'  STATION NOT FOUND')
 911       FORMAT(A4,13X,I2,I5)
```

```
            END


            SUBROUTINE DEG_KM
            1                (PLAT, PLON,     !Coords of the point
            1                 CLAT, CLON,     !Coords of the origin
            1                 X, Y)           !x-y of the point in km from origin

      c     Given a latitude and a longitude, and the lat and long of the origin
      c     return the x and y offset of the point from the origin on the surface
      c     in kilometers.

            PARAMETER (R = 6371.0,            !radius of the earth
            1         FAC = 0.01745329)       !degrees to radians

      c     convert to radians (don't convert (contaminate) original values)

              DLAT = (PLAT - CLAT) * FAC
      c       DLON = (PLON - CLON) * FAC

              olat = clat * FAC
              olon = clon * FAC

      c       rlat = plat * FAC
              rlon = plon * FAC

              Y = R * DLAT
              X = (RLON - OLON) * R * COS((DLAT/2.0)+OLAT)

             RETURN
             END

            SUBROUTINE KM_DEG (OLAT, OLON, X, Y, PLAT, PLON)

      c     Given the lat and lon of an origin point (OLAT, OLON) and the coords
      c     in km of a point away from that point (X, Y), compute the lat and lon
      c     (PLAT, PLON) of the point

            parameter (r = 6371,              !radius of the earth in km
            1         fac = 0.01745329)       !deg => radians

            rolat = olat * fac
            rolon = olon * fac

            plat = (y / r + rolat) / fac

            plon = (X/(COS(ROLAT)*R) + ROLON) /FAC

            return
            end
```