NASA Technical Memorandum 80047

(NASA-TM-80047) A PRELIMINARY STUDY OF A VERY LARGE SPACE RADIOMETRIC ANTENNA (NASA) 45 p HC A03/MF A01 CSCL 20N

N81-19365

Unclas
G3/32 19155

A PRELIMINARY STUDY OF A VERY LARGE SPACE RADIOMETRIC ANTENNA

PRADEEP K. AGRAWAL

JANUARY 1979

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

MAR 1979
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-80047		

4. Title and Subtitle	5. Report Date
A preliminary Study of a Very Large Space Radiometric Antenna	January 1979
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Pradeep K. Agrawal	
	10. Work Unit No.
9. Performing Organization Name and Address	
NASA Langley Research Center, Hampton, Virginia 23665	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and space Administration, Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

The author is affiliated with the Joint Institute for Advancement of Flight Sciences, NASA Langley Research Center, Hampton, VA 23665

16. Abstract

A preliminary study to compute the size of a special radiometric reflector antenna is presented. Operating at 1 GHz, this reflector is required to produce 200 simultaneous contiguous beams, each with a 3 dB footprint of 1 km from an assumed satellite height of 650 km. The overall beam efficiency for each beam is required to be more than 90%.

17. Key Words (Suggested by Author(s))

Reflector Antennas
Scanning Spherical Reflector
Radiometer Antennas
Distorted Reflectors

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	42	

CONTENTS

1. Summary

A preliminary study to compute the size of a special radiometric reflector antenna is presented. Operating at 1 GHz, this reflector is required to produce 200 simultaneous contiguous beams, each with a 3 dB footprint of 1 km from an assumed satellite height of 650 km. The overall beam efficiency for each beam is required to be more than 90%.

2. Introduction

The purpose of this report is to present a preliminary study for the design of a large radiometric reflector antenna system. When orbiting at a height of 650 km, this antenna system is required to produce simultaneously 200 contiguous 3 dB circular footprints on the ground, each having a diameter of 1 km. The lowest frequency of operation is 1 GHz. The footprints are required to be as identical to each other as possible. The single most important requirement on the system is that the overall beam efficiency for the copolarized component in each of the 200 beams be better than 90% within the two and a half 3 dB beamwidths. This means that among other things, the cross polarization be minimum too (<25 dB).

When in orbit, the reflector may undergo considerable thermal distortions and its performance may change. A method is therefore needed to predict the performance of even a distorted reflector. Such a technique is discussed in Section 6 of this report.

3. Spherical Reflector Approach

A solution that meets the requirements set in Section 2 is schematically shown in Figure 1. The 200 beams are simultaneously obtained by stacking 200 identical feed antennas along a concentric circular arc in front of a spherical reflector such that each feed is pointing radially towards the spherical reflector surface. Each feed thus creates its own independent footprint. And since each feed antenna essentially sees an identical segment of the spherical reflector, the resulting 200 footprints are also practically identical. Observe that the angular separation between any two consecutive feed antennas (called θ) is the same as the angular separation between the two adjacent footprints. This, for an altitude of 650 km and a footprint size of 1 km, turns out to be 0.088° and the 200 feeds stacked along the feed arc thus subtend a total of 16.6° angle at the center of the spherical reflector (Figure 2).

Notice that the angular separation between any two consecutive feeds depends only upon the altitude and the footprint size; the physical separtion, however, is the product of the angular separation (θ = 0.088°) and the feed arc radius, and therefore, depends upon the radius of the feed arc also. The radius of the feed arc, therefore, should be large enough to provide enough physical room for each of the 200 feed antennas. It is assumed

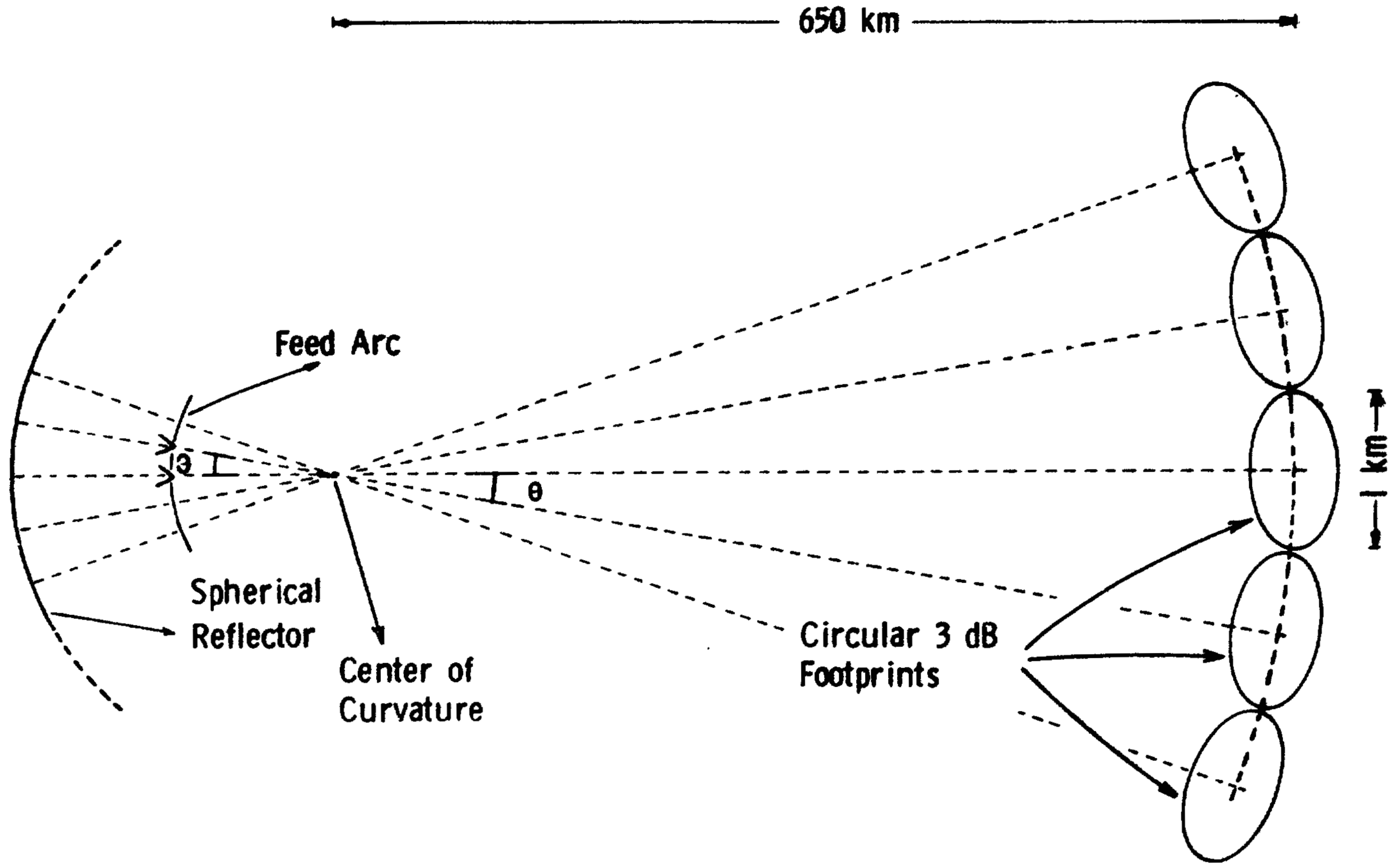

Figure 1 -- Multibeam Spherical Reflector Concept

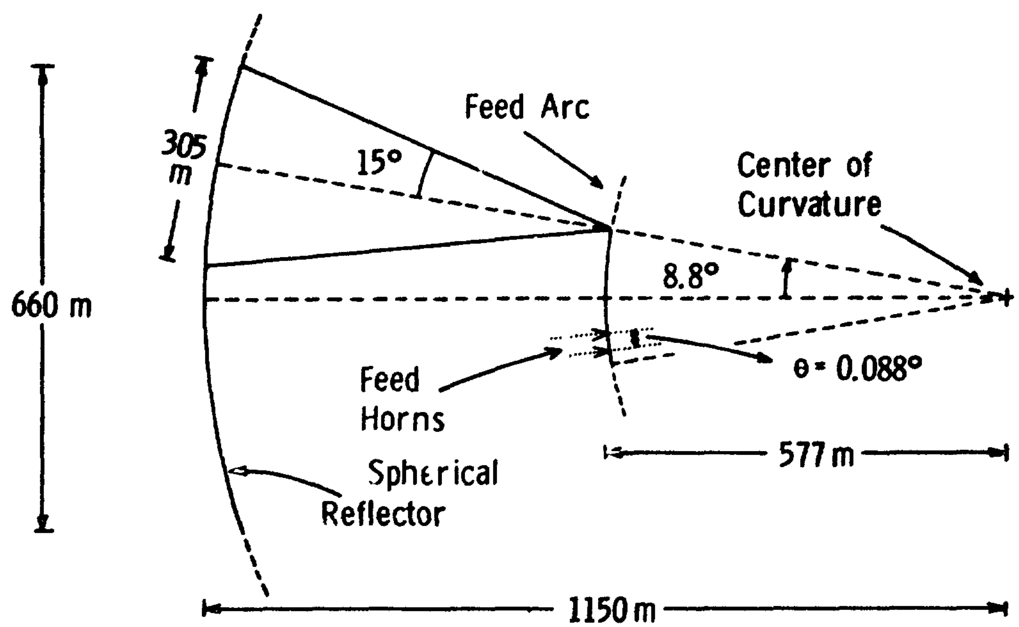

Figure 2 -- Geometry of Sphe:ical Reflector Antenna

at this stage that each feed antenna should have, on an average, a room of at least 88 cm, which leads to a feed arc radius of 577 m. And since for spherical reflectors, the feed arc is generally located about halfway between the reflector and its center of curvature (nearer to the reflector), the radius of curvature of the spherical reflector is chosen to be 1175 m. Note that there is no specific reason to pick 88 cm for feed spacing except that the estimations of practical feed horn sizes suggest that a room of about a meter be available for each feed antenna. And of course, whether a feed horn limited in size to 88 cm at 1 GHz feeding a reflector with dimensions chosen above can give a satisfactory secondary far field pattern or not, remains to be checked.

Let us now consider an individual footprint which is caused by an individual feed antenna located at the feed arc. Each feed antenna illuminates a portion of the spherical reflector and it is the far field of this illuminated reflector aperture which must have (a) a 3 dB beamwidth of 0.088°, and (b) a beam efficiency of better than 90% within the two and a half 3 dB beamwidths. The later requires that the highest side lobe of the reflected pattern be less than -32 dB with wide angle side lobes below -80 dB. A study of various aperture distributions [1] indicates that for an operating frequency of 1 GHz, an aperture diameter of about 300 m (say, 305 m) with a rotationally symmetric cosine

ORIGINAL PAGE IS
OF POOR QUALITY

squared field distribution produces both a 3 dB beamwidth of 0.088° and a side lobe at -32 dB, the side lobe fall off being -18 dB/Octave.

Returning to Figure 2, an illuminated aperture with a diameter of 305 m on the reflector corresponds to a cone of 15° half angle emanating from each feed antenna. Therefore, each feed antenna whose nominal diameter has been fixed to 88 cm at 1 GHz has to be able to produce a rotationally symmetric cosine squared far field pattern over $\pm$ 15°. The overall diameter of the spherical reflector dish to produce 200 beams then turns out to be 660 m as shown in Figure 2.

4. Feed Considerations

In the previous section, it was assumed that after reflection, each feed pattern gave rise to a rotationally symmetric aperture field which varied as cosine squared in the radial direction. For the reflector dimensions under consideration, the portion of the reflector illuminated by a feed is such a small fraction of the full sphere, that it is practically flat and therefore a rotationally symmetric cosine squared aperture distribution is easily achieved by a feed which too has a rotationally symmetric cosine squared pattern.

A rotationally symmetric cosine squared feed pattern can be generated by any one of the several types of horns. In the present study, however, a circular corrugated horn [2] is considered. The feed pattern of the corrugated horn used for the computations presented in the following sections is shown in Figure 3. Observe that the horn diameter at the mouth is 2 m which is larger than 88 cm, the space designated for each feed at the feed arc. Therefore, the feed horns will have to be staggered around the feed arc so that they are still on an average 88 cm or 0.088° apart. The beam efficiency of the feed pattern within 15°, which corresponds to the edge of the reflector is 98.3%. This number is important because the overall beam efficiency of the antenna system is the product of this efficiency and the beam efficiency of the secondary pattern.

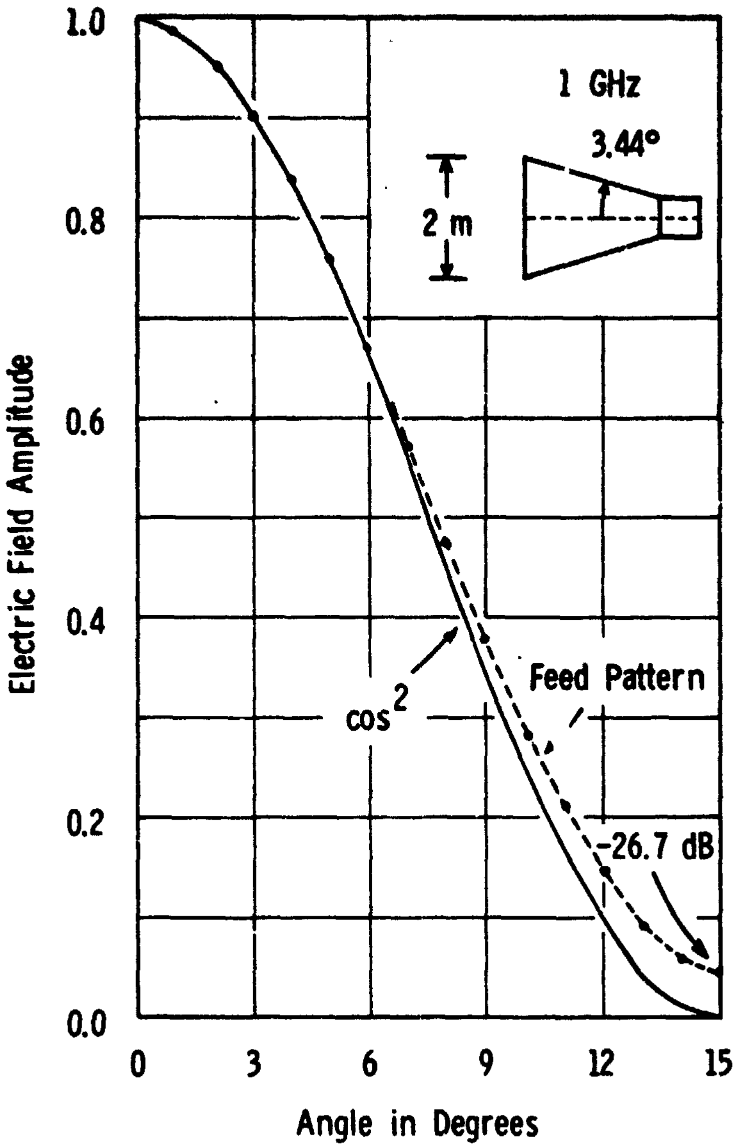

Figure 3 -- Feed Pattern of a Circular Corrugated Horn

5. Secondary Pattern

For the refle[illegible] geometry shown in Figure 2 and using the feed pattern presented in Section 4, the computed [3] secondary radiation pattern is shown in Figure 4. It has a 3 dB beamwidth of 0.080° and a maximum cross polarization level of less than -200 dB. The beam efficiency of the secondary pattern at two and a half 3 dB beamwidths is 93.4%, the overall beam efficiency, therefore, being better than 91%.

One of the concerns in spherical reflector applications is the resulting spherical aberration. It is of interest to note that in the present case, such a small segment of the sphere is being used as reflector that the maximum spherical aberration near the edge of the illuminated aperture (where the field strength is -26.7 dB, Figure 3) is equivalent to a phase error of only about 18°. Such a small phase error causes a negligible degradation in the antenna gain.

ORIGINAL PAGE IS
OF POOR QUALITY

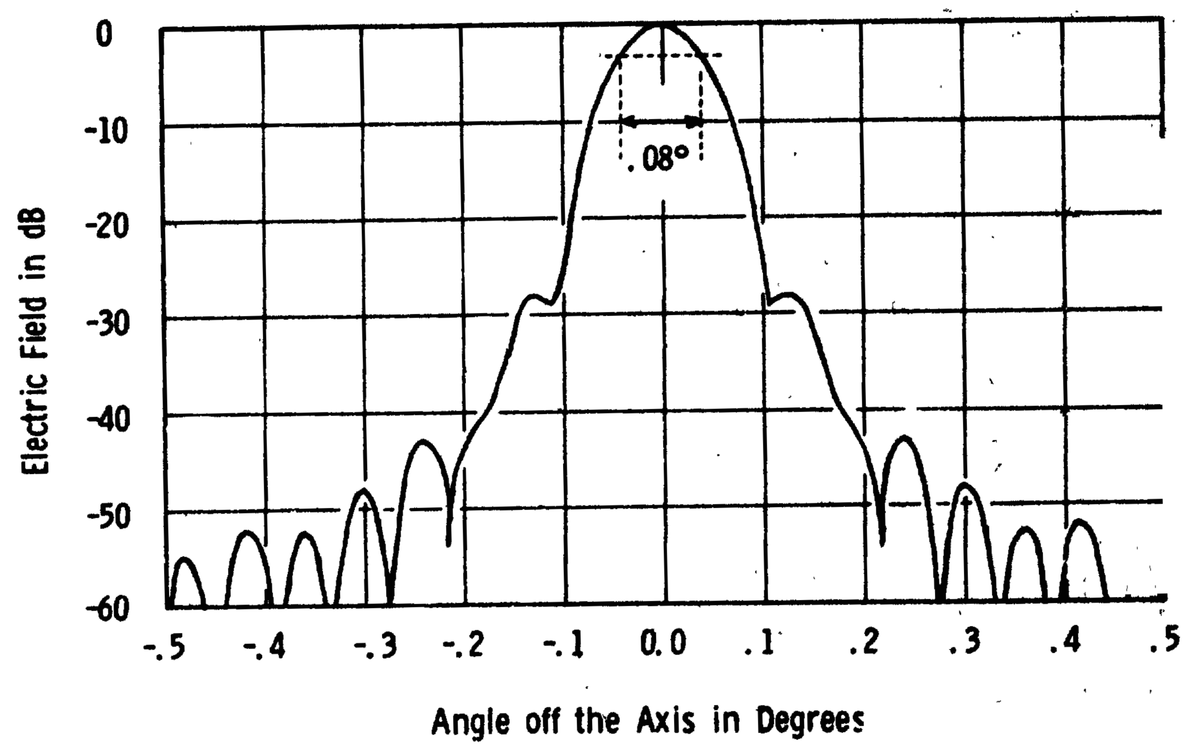

Figure 4 -- Computed Secondary Pattern. Feed Pattern was Used Over ±16° with 1° Increment.

6. Thermal Distortion Considerations

The performance of a reflector antenna in space is sometimes not the same as predicted by the initial design because the reflector undergoes severe distortions due to thermal variations. If the distorted shape of the reflector is quite a bit different from the original spherical shape, the reflector performance may change significantly and may even become unsatisfactory. Therefore, it will be desirable to be able to predict the performance of even the distorted reflector. If the distorted reflector surface could be known analytically, then the reflector performance ofcourse could be accurately predicted. It is not generally possible to know an analytic expression for the entire distorted reflector surface at all times. Alternatively, a sampling scheme can be implemented such that the coordinates of many discrete target points located along a rectangular grid on the reflector surface are known. Then, a smooth tight cubic surface can be fitted through the four corners of each of the rectangular grid patches such that the whole composite reflector surface is continuous and has continuous partial derivatives. Using this piecewise analytic expression for the reflector surface, the reflector properties can be computed. Needless to say, the target points on the reflector surface must be dense enough to sample the distortions and such that the surface between the measured points could be assumed tightly stretched.

To demonstrate that the far field radiation pattern can indeed be accurately computed even when the reflector surface is known only at certain discrete points, the following example is presented.

Computations presented in Figure 4 are made again, this time, instead of using a single analytic expression for the entire spherical reflector surface, though, the x-, y-, z- coordinates of 45 equispaced points on the reflector surface are used. These surface points are located on the reflector surface along a rectangular grid as shown in Figure 5, the points being 40 m or 133.3 wavelengths apart. For computational purposes, the reflector surface over any rectangular patch is expressed as a bi-spline under tension [4]. In Figure 6, the far field radiation pattern computed by using a single analytic expression for the entire spherical reflector surface (as in Figure 4) is shown by solid lines. On the same figure, the far field radiation pattern computed by using the piecewise analytic composite surface through 45 target points on the reflector is plotted with solid dots. The field values not shown by solid dots are too close to the solid line curve values to distinguish. The modifications needed in the computer program in Reference 3 to make the present surface fitting computations are shown in the Appendix.

In conclusion, for actual distorted reflector conditions, where the whole distorted reflector surface is not known

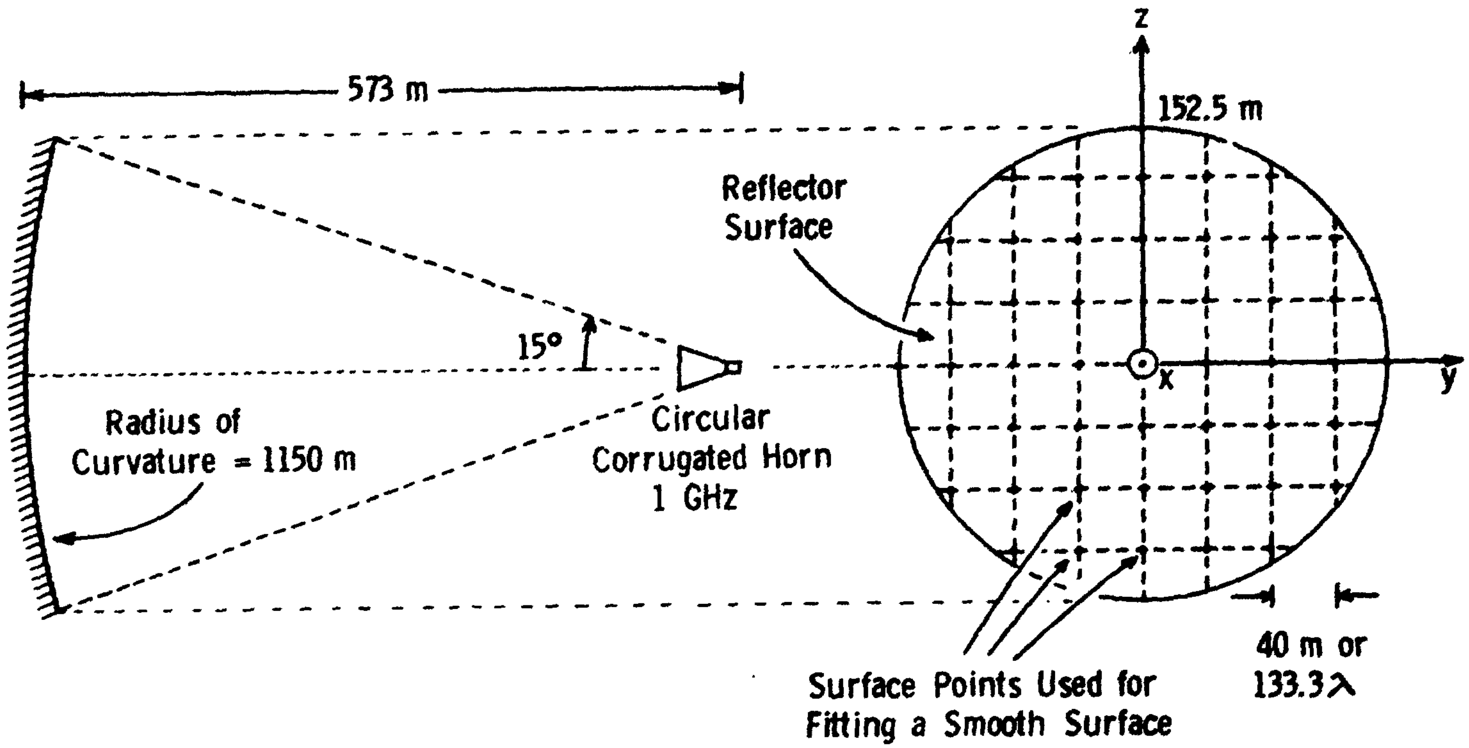

Figure 5 -- Location of Surface Target ı oints used for Surface Fitting Calculations

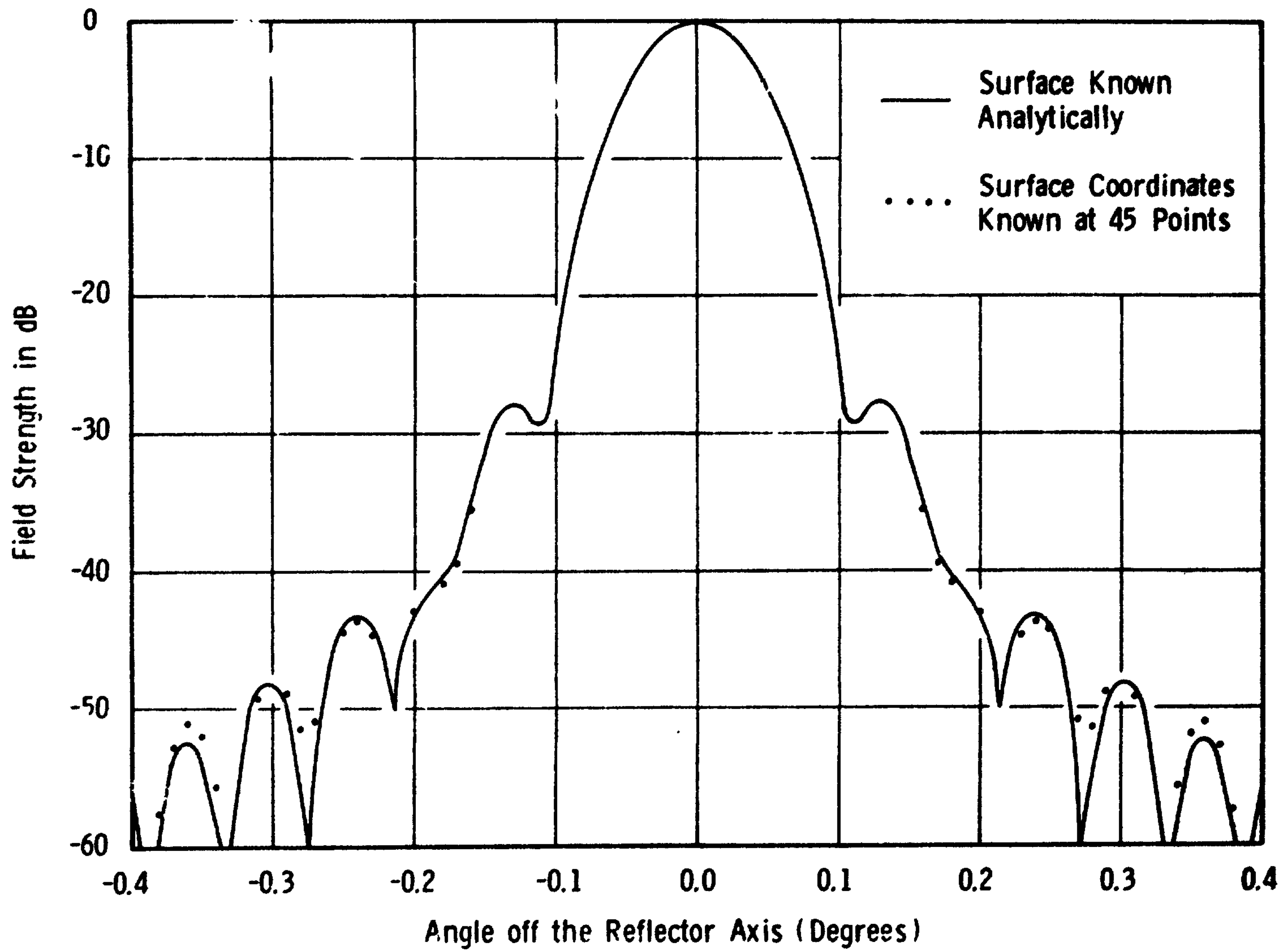

Figure 6 -- Computed Secondary Pattern

analytically, accurate far field computations can be made by using our computer program which will accept for the reflector surface geometry, a set of discrete reflector surface points. The basic underlying assumption is that the surface is smooth between the target sample points.

7. References

1. Harris, F. J., "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," Proceedings of the IEEE, Vol. 66, No. 1, January 1978.

2. Caldecott, R., Mentzer, C. A., Peters, L., and Toth, J., "High Performance S-Band Horn Antennas for Radiometric Use," The Ohio State University ElectroScience Laboratory Report 3033-1, May 1972.

3. Agrawal, P.K., "A Computer Program to Calculate Radiation Properties of Reflector Antennas," NASA Technical Memorandum No. 78721, May 1978.

4. Cline, A. K., "Six Subprograms for Curve Fitting Using Splines Under Tension," Comm. A.C.M. 17, 4, April 1974.

Appendix

The computer program documented in Reference 3 has been modified such that now it can be used for making also the surface fitting reflector calculations similar to the ones presented in Section 6. The purpose of this appendix is to document the corresponding modifications, changes, and additions.

Two new features have been added to the computer program REFLCTR (documented in Reference 3). The first feature, which is not of direct concern to the subject matter of this report is that in addition to parabolic, spherical, and ellipsoidal reflectors, the program REFLCTR can now also handle planar reflectors. A planar reflector is specified by (a) three cartesian coordinates of a point on the surface of the planar reflector - PLNPNT(1), PLNPNT(2), and PLNPNT(3), and (b) the three cartesian components of a unit vector normal to the reflector surface - PLNORM(1), PLNORM(2), and PLNORM(3). The value of the integer variable SURFACE must be set to 4 for a planar reflector.

The second new feature is that in addition to specifying a reflector surface as being parabolic, spherical, ellipsoidal, or planar by setting SURFACE = 1,2,3 or 4 respectively, one can now also choose to specify any of the above reflector surfaces in terms of only a finite number of discrete target points located along a rectangular grid (assumed square grid here for simplicity) on the reflector surface. This is done by setting the integer variable NEEDFIT to a nonzero positive value. SIGMA is the tension factor (defined later) used for fitting the surface through the grid points, the grid spacing being DISFAC for both the y- and the z- directions. All these newly defined variables along with the ones already defined in Reference 3 are read, as before, in the subroutine NPUT. A listing of modified NPUT is given in Figure A-1.

As a result of implementing the above two features, the subroutine APERTUR also changes. A listing of the new APERTUR is given in Figure A-2. Notice that in this subroutine, before statement number 100, the coordinates of surface target points are first stored in dimensioned arrays called EXTRAX, EXTRAY, and EXTRAZ and then the subroutine SURF1 is called to compute the parameters necessary to compute an interpolatory surface passing through the surface grid points. Later on in the subroutine APERTUR (before statement number 130), subroutine SURFD2 is called to interpolate the reflector surface at the given coordinate pair and to compute the components of a normal vector at the interpolated point.

The subroutines SURF1 and SURFD2 are from "A Spline Under Tension Package for Curve and Surface Fitting" by A. K. Cline, Department of Computer Science, University of Texas, Austin. This package of subroutines is an extension of Cline's work reported in Reference 4. A listing of subroutines SURF1 and SURFD2 which also includes definition of parameters used in the subroutines is presented in Figure A-3.

```
      SUBROUTINE NPUT(P)
      COMMON/PARAMS/TITLE(16),ADRORF,XLAM,GRID,SURFACE,APROTA,FEED(3),
     .               ALPHA,BETA,GAMMA,XC,YC,ZC,HEMAFX,HEMIFX,RMTP,RMPP,
     .               NT,NP,NPOINT,MAXPTS,RELLP,PLNPNT(3),PLNORM(3)
      COMMON/BLOCKG/YCBL,ZCBL,HEMABL,HEMIBL
      COMMON/PATTERN/PHI(3),THETA(3)
      COMMON/MATH/PI,PI2,PID2,DTOR,RTOD
      COMMON/SURFIT/NEEDFIT,SIGMA,DISFAC
      INTEGER TITLE,SURFACE
      READ 200, TITLE
      READ *, ADRORF,XLAM,GRID,SURFACE,NEEDFIT,APROTA
      READ *, BELLP,PLNPNT,PLNORM,SIGMA,DISFAC
      READ *, FEED,ALPHA,BETA,GAMMA
      READ *, XC,YC,ZC,HEMAFX,HEMIFX,YCBL,ZCBL,HEMABL,HEMIBL
      READ *, PHI,THETA
      IF(APRDTA.GT.0.0) WRITE(20,555)TITLE
      IF(APRDTA.GT.0.0) WRITE(20,556)FEED,ADRORF,XLAM,GRID,ALPHA,BETA,   TAPE 20
     .                        GAMMA,XC,YC,ZC,HEMAFX,HEMIFX               TAPE 20
      GO TO (130,140,150,160),SURFACE                                    ********
130   PRINT 578, TITLE,XLAM,FEED,ALPHA,BETA,GAMMA,PLNPNT,PLNORM
      GO TO 170
140   PRINT 579, TITLE,XLAM,FEED,ALPHA,BETA,GAMMA,ADRORF,BELLP
      GO TO 170
150   PRINT 580, TITLE,XLAM,FEED,ALPHA,BETA,GAMMA,ADRORF
      GO TO 170
160   PRINT 581, TITLE,XLAM,FEED,ALPHA,BETA,GAMMA,ADRORF
170   PRINT 582, XC,YC,ZC,HEMAFX,HEMIFX,GRID,YCBL,ZCBL,HEMABL,HEMIBL,
     .           THETA,PHI
      IF(NEEDFIT.GT.0) PRINT 583, SIGMA,DISFAC
200   FORMAT(8A10)
555   FORMAT(1X,8A10)
556   FORMAT(1X,F15.4)
578   FORMAT(1H1,////,13X,*PLANAR REFLECTOR FAR FIELD RADIATION        */
     .      *                          PATTERN CALCULATION*///* *8A10/* *8A1
     .     0/* WAVELENGTH OF ELECTRIC FIELD.......................*F9.4/
     .       * LOCATION OF COORDINATE ORIGIN WRT FEED (X,Y,Z).....*3F7.2
     .      /* FEED ROTATION ANGLES (ALPHA,BETA,GAMMA)............*3F7.2
     .      /* A POINT ON THE REFLECTOR SURFACE (X,Y,Z)...........*3F7.2
     .      /* COMPONENTS OF UNIT NORMAL TO SURFACE (X,Y,Z).......*3F7.2
     .      )
579   FORMAT(1H1,////,11X,*ELLIPTICAL REFLECTOR FAR FIELD RADIATION    */
     .      *                          PATTERN CALCULATION*///* *8A10/* *8A1
     .     0/* WAVELENGTH OF ELECTRIC FIELD.......................*F9.4/
     .       * LOCATION OF COORDINATE ORIGIN WRT FEED (X,Y,Z).....*3F7.2
     .      /* FEED ROTATION ANGLES (ALPHA,BETA,GAMMA)............*3F7.2
     .      /* MAJOR AXIS OF THE ELLIPTICAL REFLECTOR.............*F7.2/
```

Figure A-1

ORIGINAL PAGE
OF POOR QUALITY

```
     .       * MINOR AXIS OF THE ELLIPTICAL REFLECTOR..............*F7.2)
580   FORMAT(1H1,////,11X,*SPHERICAL REFLECTOR FAR FIELD RADIATION    */
     .       *                        PATTERN CALCULATION*///* *8A10/* *8A1
     .     0/* WAVELENGTH OF ELECTRIC FIELD.........................*F9.4/
     .       * LOCATION OF COORDINATE ORIGIN WRT FEED (X,Y,Z).....*3F7.2
     .      /* FEED ROTATION ANGLES (ALPHA,BETA,GAMMA).............*3F7.2
     .      /* RADIUS OF THE REFLECTOR SPHERE......................*F7.2)
581   FORMAT(1H1,////,11X,*PARABOLIC REFLECTOR FAR FIELD RADIATION    */
     .       *                        PATTERN CALCULATION*///* *8A10/* *8A1
     .     0/* WAVELENGTH OF ELECTRIC FIELD.........................*F9.4/
     .       * LOCATION OF COORDINATE ORIGIN WRT FEED (X,Y,Z).....*3F7.2
     .      /* FEED ROTATION ANGLES (ALPHA,BETA,GAMMA).............*3F7.2
     .      /* FOCAL LENGTH OF THE REFLECTOR.......................*F7.2)
582   FORMAT(* APERTURE PLANE LOCATION (XC).......................* F8.2/
     .       * COORDINATES OF THE APERTURE PLANE CENTER............*2F7.2
     .      /* HALF MAJOR AXIS OF APERTURE PLANE (ALONG Y)........*F7.2/
     .       * HALF MINOR AXIS OF APERTURE PLANE (ALONG Z)........*F7.2/
     .       * GRID SIZE USED FOR NUMERICAL INTEGRATION............*F9.4/
     .       * FEED SHADOW CENTER COORDINATES IN APERTURE PL......*2F7.2
     .      /* HALF MAJOR AXIS OF FEED SHADOW......................*F7.2/
     .       * HALF MINOR AXIS OF FEED SHADOW......................*F7.2/
     .       * THETA RANGE FOR FEED PATTERN (L,H,I - DEGREES).....*3F7.2
     .      /* PHI RANGE FOR FEED PATTERN (L,H,I - DEGREES).......*3F7.2
     .       )
583   FORMAT(* SIGMA USED FOR SURFACE FITTING......................*F7.2/
     .       * SPACING BETWEEN SURFACE POINTS......................*F7.2)
590   FORMAT(* ----- INSUFFICIENT WORK STORAGE, NEEDED *I5* AVAILABLE IS
     . ONLY *I5* ----- *)
      PI=ACOS(-1.0)
      PI2=PI+PI
      PID2=0.5*PI
      DTOR=PI/180.
      RTOD=180./PI
      NP=(PHI(2)-PHI(1))/PHI(3)+1.5
      NT=(THETA(2)-THETA(1))/THETA(3)+1.5
      NPOINT=NT*NP
      IF(NPOINT .GE. MAXPTS) GO TO 600
      CALL FILL(P,NT,NP)
      RETURN
600   PRINT 590, NPOINT,MAXPTS
      STOP
      END
```

Figure A-1 (Continued)

```
      SUBROUTINE APERTUR(P,NTX,NPX)
      COMMON/PARAMS/TITLE(16),AORORF,XLAM,GRID,SURFACE,APRDTA,FEED(3),
     .              ALPHA,BETA,GAMMA,XC,YC,ZC,HEMAEX,HEMIEX,BMTP,BMPP,
     .              NT,NP,NPOINT,MAXPTS,BELLP,PLNPNT(3),PLNORM(3)
      COMMON/BLOCKG/YCBL,ZCBL,HEMABL,HEMIBL
      COMMON/MATH/PI,PI2,PID2,DTOR,RTOD
      COMMON/POINTS/NEDGE,NINTR
      COMMON/SURFIT/NEEDFIT,SIGMA,DISFAC
      REAL NHAT
      INTEGER SEDGE,SURFACE
      DIMENSION P(5,NTX,NPX),POLD(5),PNEW(5),PINT(5),PBLK(5),A(3,3),
     .          B(3,2),BB(3,2),NHAT(3),C(3),SR(3),FI(3),FR(3)
      DIMENSION EXTRAY(19),EXTRAZ(19),EXTRAX(19,19),EXTRA(1083),TEMP(57)
      DIMENSION ZX1(19),ZXM(19),ZY1(19),ZYN(19)
      IF(NEEDFIT.LE.0) GO TO 100
      IFIT=19
      DO 50 I=1,IFIT
      EXTRAZ(I)=EXTRAY(I)=-10.0*DISFAC+I*DISFAC
50    CONTINUE
      DO 90 I=1,IFIT
      DO 90 J=1,IFIT
      GO TO (81,82,83,84) SURFACE                                       ********
81    EXTRAX(I,J)=            PLNPNT(1) *PLNORM(1)
     .           -(EXTRAY(I)-PLNPNT(2))*PLNORM(2)
     .           -(EXTRAZ(J)-PLNPNT(3))*PLNORM(3)
      EXTRAX(I,J)=EXTRAX(I,J)/PLNORM(1)
      GO TO 90
82    EXTRAX(I,J)=-AORORF*SQRT(1.-(EXTRAY(I)**2-EXTRAZ(J)**2)/BELLP**2)
      GO TO 90
83    EXTRAX(I,J)=-SQRT(AORORF**2-EXTRAY(I)**2-EXTRAZ(J)**2)
      GO TO 90
84    EXTRAX(I,J)=(-4.0*AORORF**2+EXTRAY(I)**2+EXTRAZ(J)**2)/(4.*AORORF)
90    CONTINUE
      CALL SURF1(IFIT,IFIT,EXTRAY,EXTRAZ,EXTRAX,IFIT,ZX1,ZXM,ZY1,ZYN,
     .          ZXY11,ZXYM1,ZXY1N,ZXYMN,255,EXTRA,TEMP,SIGMA,IERR)
100   ALPHAR=ALPHA*DTOR
      BETAR=BETA*DTOR
      GAMMAR=GAMMA*DTOR
      A(1,1)=COS(ALPHAR)*COS(GAMMAR)-SIN(ALPHAR)*SIN(BETAR)*SIN(GAMMAR)
      A(1,2)=SIN(ALPHAR)*COS(GAMMAR)+COS(ALPHAR)*SIN(BETAR)*SIN(GAMMAR)
      A(1,3)=-COS(BETAR)*SIN(GAMMAR)
      A(2,1)=-SIN(ALPHAR)*COS(BETAR)
      A(2,2)= COS(ALPHAR)*COS(BETAR)
      A(2,3)= SIN(BETAR)
      A(3,1)=COS(ALPHAR)*SIN(GAMMAR)+SIN(ALPHAR)*SIN(BETAR)*COS(GAMMAR)
      A(3,2)=SIN(ALPHAR)*SIN(GAMMAR)-COS(ALPHAR)*SIN(BETAR)*COS(GAMMAR)
      A(3,3)= COS(BETAR)*COS(GAMMAR)
      IF(APRDTA.GT.0.0) WRITE(20,110)                                   TAPE 20
110   FORMAT(1H1/,10H**********,7X,*THETA*,9X,*Y*,9X,*Z*,7X,*ERY*,7X,
     .       *ERZ*,5X,*PHASE*,9X,*R*)
```

Figure A-2

ORIGINAL PAGE IS OF POOR QUALITY

```
      BMTEST=1.0E+40
      NINTR=NEDGE=0
      PBLK(3)=PBLK(4)=PBLK(5)=0.0
      DO 5000 IP=1,NP
      DEGPHI=P(5,1,IP)*RTOD
      IF(APROTA.GT.0.0) WRITE(20,120) DEGPHI                          TAPE 20
120   FORMAT(1X,*PHI=*,F10.4)
      DO 4000 IT=1,NT
      DEGTHET=P(4,IT,IP)*RTOD
      SINP=SIN(P(5,IT,IP))
      COSP=COS(P(5,IT,IP))
      SINT=SIN(P(4,IT,IP))
      COST=COS(P(4,IT,IP))
      BB(1,1)=SINT*COSP
      BB(2,1)=SINT*SINP
      BB(3,1)=COST
      BB(1,2)=+FEED(1)
      BB(2,2)=+FEED(2)
      BB(3,2)=+FEED(3)
      CALL MULT32(B,A,BB)
      GO TO (121,122,124,126),SURFACE                                 ********
121   AR=0.0
      BR=B(1,1)*PLNORM(1)+B(2,1)*PLNORM(2)+B(3,1)*PLNORM(3)
      CR=-(B(1,2)+PLNPNT(1))*PLNORM(1)
     .   -(B(2,2)+PLNPNT(2))*PLNORM(2)
     .   -(B(3,2)+PLNPNT(3))*PLNORM(3)
      GO TO 128
122   AR=B(1,1)**2/ADRORF**2+(B(2,1)**2+B(3,1)**2)/BELLP**2
      BR=-2.0*(B(1,1)*B(1,2)/ADRORF**2+(B(2,1)*B(2,2)+B(3,1)*B(3,2))/
     .                                BELLP**2)
      CR=B(1,2)**2/ADRORF**2+(B(2,2)**2+B(3,2)**2)/BELLP**2-1.0
      GO TO 128
124   AR=B(1,1)*B(1,1)+B(2,1)*B(2,1)+B(3,1)*B(3,1)
      BR=-2.*(B(1,1)*B(1,2)+B(2,1)*B(2,2)+B(3,1)*B(3,2))
      CR=B(1,2)*B(1,2)+B(2,2)*B(2,2)+B(3,2)*B(3,2)-ADRORF*ADRORF
      GO TO 128
126   AR=B(2,1)*B(2,1)+B(3,1)*B(3,1)
      BR=-2.0*(B(2,1)*B(2,2)+B(3,1)*B(3,2)+2.0*ADRORF*B(1,1))
      CR=B(2,2)*B(2,2)+B(3,2)*B(3,2)+4.0*ADRORF*B(1,2)-4.0*ADRORF**2
128   IF (AR.LT.1.0E-10) R=-CR/BR
      IF (AR.LT.1.0E-10) GO TO 129
      R=(-BR+SQRT(BR*BR-4.*AR*CR))/(AR+AR)
129   CONTINUE
      XO=B(1,1)*R-B(1,2)
      YO=B(2,1)*R-B(2,2)
      ZO=B(3,1)*R-B(3,2)
```

Figure A-2 (Continued)

```
      IF(NEEDFIT.LE.0) GO TO 130
      X0=SURFD2(Y0,Z0,YNORM,ZNORM,IFIT,IFIT,EXTRAY,EXTRAZ,EXTRAX,IFIT
     .        ,EXTRA,SIGMA)
      R=SQRT((X0+B(1,2))**2+(Y0+B(2,2))**2+(Z0+B(3,2))**2)
      XMAG=SQRT(1.0+YNORM**2+ZNORM**2)
      NHAT(1)=1.0/XMAG
      NHAT(2)=-SIGN(YNORM,Y0)/XMAG
      NHAT(3)=-SIGN(ZNORM,Z0)/XMAG
      GO TO 138
130   GO TO (131,132,134,136),SURFACE                          ********
131   NHAT(1)=PLNORM(1)
      NHAT(2)=PLNORM(2)
      NHAT(3)=PLNORM(3)
      GO TO 138
132   NHAT(1)=-X0*BELLP **2/SQRT(X0**2*BELLP**4+(Y0**2+Z0**2)*AORORE**4)
      NHAT(2)=-Y0*AORORE**2/SQRT(X0**2*BELLP**4+(Y0**2+Z0**2)*AORORE**4)
      NHAT(3)=-Z0*AORORE**2/SQRT(X0**2*BELLP**4+(Y0**2+Z0**2)*AORORE**4)
      GO TO 138
134   NHAT(1)=-X0/AORORE
      NHAT(2)=-Y0/AORORE
      NHAT(3)=-Z0/AORORE
      GO TO 138
136   NHAT(1)=2.0*AORORE/SQRT(4.0*AORORE**2+Y0**2+Z0**2)
      NHAT(2)=         -Y0/SQRT(4.0*AORORE**2+Y0**2+Z0**2)
      NHAT(3)=         -Z0/SQRT(4.0*AORORE**2+Y0**2+Z0**2)
138   SCALAR=2.0*(B(1,1)*NHAT(1)+B(2,1)*NHAT(2)+B(3,1)*NHAT(3))
      DO 1500 I=1,3
1500  SR(I)=B(I,1)-SCALAR*NHAT(I)
      ETI=P(1,IT,IP)/R
      EPI=P(2,IT,IP)/R
      C(1)=COST*COSP*ETI-SINP*EPI
      C(2)=COST*SINP*ETI+COSP*EPI
      C(3)=-SINT*ETI
      DO 2000 I=1,3
      EI(I)=0.0
      DO 2000 J=1,3
2000  EI(I)=EI(I)+A(I,J)*C(J)
      SCALAR=2.0*(EI(1)*NHAT(1)+EI(2)*NHAT(2)+EI(3)*NHAT(3))
      DO 2500 I=1,3
2500  ER(I)=SCALAR*NHAT(I)-EI(I)
      Y=Y0+(XC-X0)*SR(2)/SR(1)
      Z=Z0+(XC-X0)*SR(3)/SR(1)
      D=SQRT((XC-X0)*(XC-X0)+(Y-Y0)*(Y-Y0)+(Z-Z0)*(Z-Z0))
      PHASE=PI2*(R+D)/XLAM+P(3,IT,IP)
      PNEW(1)=PBLK(1)=Y
      PNEW(2)=PBLK(2)=Z
```

Figure A-2 (Continued)

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      PNEW(3)=ER(2)
      PNEW(4)=ER(3)
      PNEW(5)=PHASE
      TEST=  HEMAEX*HEMAEX*HEMIEX*HEMIEX-HEMAEX*HEMAEX*(Z-ZC)*(Z-ZC)
     .                                  -HEMIEX*HEMIEX*(Y-YC)*(Y-YC)
      TESTBL=HEMABL*HEMABL*HEMIBL*HEMIBL-HEMABL*HEMABL*(Z-ZCBL)*(Z-ZCBL)
     .                                  -HEMIBL*HEMIBL*(Y-YCBL)*(Y-YCBL)
      IF (TEST) 2701,2501,2601
 2501 NEDGE=NEDGE+1                                                   EDGE
      SEDGE=MAXPTS-NEDGE
      IF (TESTBL.LT.0.0) GO TO 2510
      CALL MOVEM(PBLK,P(1,SEDGE),5)
      IF(APRDTA.GT.0.0) WRITE(20,2505) PBLK                           TAPE 20
 2505 FORMAT(1X,*$*,23X,2F10.4,2F10.7,F10.4,12X,*EDGE POINT, BLOCKED*)
      GO TO 2515
 2510 CALL MOVEM(PNEW,P(1,SEDGE),5)
      IF(APRDTA.GT.0.0) WRITE(20,2512) PNEW                           TAPE 20
 2512 FORMAT(1X,*$*,23X,2F10.4,2F10.7,F10.4,12X,*EDGE POINT*)
 2515 CONTINUE
      GO TO 2800
 2601 NINTR=NINTR+1                                                   INSIDE
      IF (TESTBL.LT.0.0) GO TO 2610
      CALL MOVEM(PBLK,P(1,NINTR),5)
      IF(APRDTA.GT.0.0) WRITE(20,2605) DEGTHET,PBLK,R                 TAPE 20
 2605 FORMAT(1X,*$*,13X,3F10.4,2F10.7,2F10.4,14X,*BLOCKED*)
      GO TO 2615
 2610 CALL MOVEM(PNEW,P(1,NINTR),5)
      IF(APRDTA.GT.0.0) WRITE(20,2612) DEGTHET,PNEW,R                 TAPE 20
 2612 FORMAT(1X,*$*,13X,3F10.4,2F10.7,2F10.4)
 2615 CONTINUE
 2701 IF (IT.EQ.1) GO TO 2800                                         OUTSIDE
      IF (TEST*TESTO) 2704,2800,2900
 2704 ZTEST1=ZC-POLD(2)
      ZTEST2=ZC-PNEW(2)
      IF(ZTEST1*ZTEST2.LT.0.0) GO TO 2800
      CALL INTERP(POLD,PNEW,PINT)
      NEDGE=NEDGE+1
      SEDGE=MAXPTS-NEDGE
      CALL MOVEM(PINT,PBLK,2)
      Y=PINT(1)
      Z=PINT(2)
      TESTBL=HEMABL*HEMABL*HEMIBL*HEMIBL-HEMABL*HEMABL*(Z-ZCBL)*(Z-ZCBL)
     .                                  -HEMIBL*HEMIBL*(Y-YCBL)*(Y-YCBL)
      IF (TESTBL.LT.0.0) GO TO 2710
      CALL MOVEM(PBLK,P(1,SEDGE),5)
      IF(APRDTA.GT.0.0) WRITE(20,2705) PBLK                           TAPE 20
```

Figure A-2 (Continued)

```
2705  FORMAT(1X,*S*,23X,2F10.4,2F10.7,F10.4,10X,*INTERPOLATED, BLOCKED*)
      GO TO 2715
2710  CALL MOVEM(PINT,P(1,SEDGE),5)
      IF(APRDTA.GT.0.0) WRITE(20,2712) PINT                           TAPE 20
2712  FORMAT(1X,*S*,23X,2F10.4,2F10.7,F10.4,10X,*INTERPOLATED*)
2715  CONTINUE
2800  CALL MOVEM(PNEW,POLD,5)
      TESTO=TEST
      TEST=(DEGTHET-90.0)**2+(DEGPHI-180.0)**2
      IF (TEST-BMTEST) 2980,3000,3000
2980  BMTEST=TEST
      BMT=DEGTHET
      BMP=DEGPHI
      BMSRX=SR(1)
      BMSRY=SR(2)
      BMSRZ=SR(3)
3000  CONTINUE
4000  CONTINUE
5000  CONTINUE
      PRINT 5025, NINTR,NEDGE
5025  FORMAT(* NUMBER OF INTERNAL POINTS...............................*I5/
     .        * NUMBER OF EDGE POINTS .................................*I5)
      COSBMTP=BMSRZ/SQRT(BMSRX**2+BMSRY**2+BMSRZ**2)
      SINBMPP=BMSRY/SQRT(BMSRX**2+BMSRY**2)
      BMTP=RTOD*ACOS(COSBMTP)
      BMPP=RTOD*ASIN(SINBMPP)
      PRINT 5050, BMT,BMP,BMSRX,BMSRY,BMSRZ,BMTP,BMPP
5050  FORMAT(* THE FEED MAX IS...............................THETA=*F7.2/
     .        *                  ...............................PHI=*F7.2/
     .        * THE REFLECTED RAY VECTOR (X,Y,Z)...................*3F7.2
     .       /* THE REFLECTED BEAM MAX IS....................THETA=*F9.4/
     .        *                            .....................PHI=*F9.4)
      RETURN
      END
```

Figure A-2 (Continued)

```
      SUBROUTINE SURF1 (M,N,X,Y,Z,IZ,ZX1,ZXM,ZY1,ZYN,ZXY11,
     *                  ZXYM1,ZXY1N,ZXYMN,ISLPSW,ZP,TEMP,
     *                  SIGMA,IERR)
C
C THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO
C COMPUTE AN INTERPOLATORY SURFACE PASSING THROUGH A RECT-
C ANGULAR GRID OF FUNCTIONAL VALUES. THE SURFACE DETERMINED
C CAN BE REPRESENTED AS THE TENSOR PRODUCT OF SPLINES UNDER
C TENSION. THE X- AND Y-PARTIAL DERIVATIVES AROUND THE
C BOUNDARY AND THE X-Y-PARTIAL DERIVATIVES AT THE FOUR
C CORNERS MAY BE SPECIFIED OR OMITTED. FOR ACTUAL MAPPING
C OF POINTS ONTO THE SURFACE IT IS NECESSARY TO CALL THE
C FUNCTION SURF2.
C
C ON INPUT--
C
C   M IS THE NUMBER OF GRID LINES IN THE X-DIRECTION, I. E.
C   LINES PARALLEL TO THE Y-AXIS (M .GE. 2).
C
C   N IS THE NUMBER OF GRID LINES IN THE Y-DIRECTION, I. E.
C   LINES PARALLEL TO THE X-AXIS (N .GE. 2).
C
C   X IS AN ARRAY OF THE M X-COORDINATES OF THE GRID LINES
C   IN THE X-DIRECTION. THESE SHOULD BE STRICTLY INCREASING.
C
C   Y IS AN ARRAY OF THE N Y-COORDINATES OF THE GRID LINES
C   IN THE Y-DIRECTION. THESE SHOULD BE STRICTLY INCREASING.
C
C   Z IS AN ARRAY OF THE M * N FUNCTIONAL VALUES AT THE GRID
C   POINTS, I. E. Z(I,J) CONTAINS THE FUNCTIONAL VALUE AT
C   (X(I),Y(J)) FOR I = 1,...,M AND J = 1,...,N.
C
C   IZ IS THE ROW DIMENSION OF THE MATRIX Z USED IN THE
C   CALLING PROGRAM (IZ .GE. M).
C
C   ZX1 AND ZXM ARE ARRAYS OF THE N X-PARTIAL DERIVATIVES
C   OF THE FUNCTION ALONG THE X(1) AND X(M) GRID LINES,
C   RESPECTIVELY. THUS ZX1(J) AND ZXM(J) CONTAIN THE X-PART-
C   IAL DERIVATIVES AT THE POINTS (X(1),Y(J)) AND
C   (X(M),Y(J)), RESPECTIVELY, FOR J = 1,...,N. EITHER OF
C   THESE PARAMETERS WILL BE IGNORED (AND APPROXIMATIONS
C   SUPPLIED INTERNALLY) IF ISLPSW SO INDICATES.
C
C   ZY1 AND ZYN ARE ARRAYS OF THE M Y-PARTIAL DERIVATIVES
C   OF THE FUNCTION ALONG THE Y(1) AND Y(N) GRID LINES,
C   RESPECTIVELY. THUS ZY1(I) AND ZYN(I) CONTAIN THE Y-PART-
```

Figure A - 3

```
C     IAL DERIVATIVES AT THE POINTS (X(I),Y(1)) AND
C     (X(I),Y(N)), RESPECTIVELY, FOR I = 1,...,M. EITHER OF
C     THESE PARAMETERS WILL BE IGNORED (AND ESTIMATIONS
C     SUPPLIED INTERNALLY) IF ISLPSW SO INDICATES.
C
C     ZXY11, ZXYM1, ZXY1N, AND ZXYMN ARE THE X-Y-PARTIAL
C     DERIVATIVES OF THE FUNCTION AT THE FOUR CORNERS,
C     (X(1),Y(1)), (X(M),Y(1)), (X(1),Y(N)), AND (X(M),Y(N)),
C     RESPECTIVELY. ANY OF THE PARAMETERS WILL BE IGNORED (AND
C     ESTIMATIONS SUPPLIED INTERNALLY) IF ISLPSW SO INDICATES.
C
C     ISLPSW CONTAINS A SWITCH INDICATING WHICH BOUNDARY
C     DERIVATIVE INFORMATION IS USER-SUPPLIED AND WHICH
C     SHOULD BE ESTIMATED BY THIS SUBROUTINE. TO DETERMINE
C     ISLPSW, LET
C          I1 = 0 IF ZX1 IS USER-SUPPLIED (AND = 1 OTHERWISE),
C          I2 = 0 IF ZXM IS USER-SUPPLIED (AND = 1 OTHERWISE),
C          I3 = 0 IF ZY1 IS USER-SUPPLIED (AND = 1 OTHERWISE),
C          I4 = 0 IF ZYN IS USER-SUPPLIED (AND = 1 OTHERWISE),
C          I5 = 0 IF ZXY11 IS USER-SUPPLIED
C                                        (AND = 1 OTHERWISE),
C          I6 = 0 IF ZXYM1 IS USER-SUPPLIED
C                                        (AND = 1 OTHERWISE),
C          I7 = 0 IF ZXY1N IS USER-SUPPLIED
C                                        (AND = 1 OTHERWISE),
C          I8 = 0 IF ZXYMN IS USER-SUPPLIED
C                                        (AND = 1 OTHERWISE),
C     THEN ISLPSW = I1 + 2*I2 + 4*I3 + 8*I4 + 16*I5 + 32*I6
C                   + 64*I7 + 128*I8
C     THUS ISLPSW = 0 INDICATES ALL DERIVATIVE INFORMATION IS
C     USER-SUPPLIED AND ISLPSW = 255 INDICATES NO DERIVATIVE
C     INFORMATION IS USER-SUPPLIED. ANY VALUE BETWEEN THESE
C     LIMITS IS VALID.
C
C     ZP IS AN ARRAY OF AT LEAST 3*M*N LOCATIONS.
C
C     TEMP IS AN ARRAY OF AT LEAST N+N+M LOCATIONS WHICH IS
C     USED FOR SCRATCH STORAGE.
C
C AND
C
C     SIGMA CONTAINS THE TENSION FACTOR. THIS VALUE INDICATES
C     THE CURVINESS DESIRED. IF ABS(SIGMA) IS NEARLY ZERO
C     (E. G. .001) THE RESULTING SURFACE IS APPROXIMATELY THE
C     TENSOR PRODUCT OF CUBIC SPLINES. IF ABS(SIGMA) IS LARGE
C     (E. G. 50.) THE RESULTING SURFACE IS APPROXIMATELY
```

Figure A - 3 (Continued)

```
C     BI-LINEAR. IF SIGMA EQUALS ZERO TENSOR PRODUCTS OF
C     CUBIC SPLINES RESULT. A STANDARD VALUE FOR SIGMA IS
C     APPROXIMATELY 1. IN ABSOLUTE VALUE.
C
C ON OUTPUT--
C
C     ZP CONTAINS THE VALUES OF THE XX-, YY-, AND XXYY-PARTIAL
C     DERIVATIVES OF THE SURFACE AT THE GIVEN NODES.
C
C     IERR CONTAINS AN ERROR FLAG,
C          = 0 FOR NORMAL RETURN,
C          = 1 IF N IS LESS THAN 2 OR M IS LESS THAN 2,
C          = 2 IF THE X-VALUES OR Y-VALUES ARE NOT STRICTLY
C              INCREASING.
C
C AND
C
C     M, N, X, Y, Z, IZ, ZX1, ZXM, ZY1, ZYN, ZXY11, ZXYM1,
C     ZXY1N, ZXYMN, ISLPSW, AND SIGMA ARE UNALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULES CEEZ, TERMS,
C AND SNHCSH.
C
C-----------------------------------------------------------------
C
      INTEGER M,N,IZ,ISLPSW
      REAL X(M),Y(N),Z(IZ,N),ZX1(N),ZXM(N),ZY1(M),ZYN(M),
     *        ZXY11,ZXYM1,ZXY1N,ZXYMN,ZP(M,N,3),TEMP(1)
      MM1 = M-1
      MP1 = M+1
      NM1 = N-1
      NP1 = N+1
      NPM = N+M
      IERR = 0
      IF (N .LE. 1 .OR. M .LE. 1) GO TO 46
      IF (Y(N) .LE. Y(1)) GO TO 47
      SIGMAY = ABS(SIGMA)*FLOAT(N-1)/(Y(N)-Y(1))
      IF ((ISLPSW/8)*2 .NE. (ISLPSW/4)) GO TO 2
      DO 1 I = 1,M
    1   ZP(I,1,1) = ZY1(I)
      GO TO 5
    2 DELY1 = Y(2)-Y(1)
      DELY2 = DELY1+DELY1
      IF (N .GT. 2) DELY2 = Y(3)-Y(1)
      IF (DELY1 .LE. 0. .OR. DELY2 .LE. DELY1) GO TO 47
      CALL CEEZ (DELY1,DELY2,SIGMAY,C1,C2,C3,N)
```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure A-3 (Continued)

```
      DO 3 I = 1,M
    3   ZP(I,1,1) = C1*Z(I,1)+C2*Z(I,2)
      IF (N .EQ. 2) GO TO 5
      DO 4 I = 1,M
    4   ZP(I,1,1) = ZP(I,1,1)+C3*Z(I,3)
    5 IF ((ISLPSW/16)*2 .NE. (ISLPSW/8)) GO TO 7
      DO 6 I = 1,M
        NPI = N+I
    6   TEMP(NPI) = ZYM(I)
      GO TO 10
    7 DELYN = Y(N)-Y(NM1)
      DELYNM = DELYN+DELYN
      IF (N .GT. 2) DELYNM = Y(N)-Y(N-2)
      IF (DELYN .LE. 0. .OR. DELYNM .LE. DELYN) GO TO 47
      CALL CEEZ (-DELYN,-DELYNM,SIGMAY,C1,C2,C3,N)
      DO 8 I = 1,M
        NPI = N+I
    8   TEMP(NPI) = C1*Z(I,N)+C2*Z(I,NM1)
      IF (N .EQ. 2) GO TO 10
      DO 9 I = 1,M
        NPI = N+I
    9   TEMP(NPI) = TEMP(NPI)+C3*Z(I,N-2)
   10 IF (X(M) .LE. X(1)) GO TO 47
      SIGMAX = ABS(SIGMA)*FLOAT(M-1)/(X(M)-X(1))
      IF ((ISLPSW/2)*2 .NE. ISLPSW) GO TO 12
      DO 11 J = 1,N
   11   ZP(1,J,2) = ZX1(J)
   12 IF ((ISLPSW/32)*2 .EQ. (ISLPSW/16) .AND.
     *    (ISLPSW/128)*2  .EQ. (ISLPSW/64)) GO TO 15
      DELX1 = X(2)-X(1)
      DELX2 = DELX1+DELX1
      IF (M .GT. 2) DELX2 = X(3)-X(1)
      IF (DELX1 .LE. 0. .OR. DELX2 .LE. DELX1) GO TO 47
      CALL CEEZ (DELX1,DELX2,SIGMAX,C1,C2,C3,M)
      IF ((ISLPSW/2)*2 .EQ. ISLPSW) GO TO 15
      DO 13 J = 1,N
   13   ZP(1,J,2) = C1*Z(1,J)+C2*Z(2,J)
      IF (M .EQ. 2) GO TO 15
      DO 14 J = 1,N
   14   ZP(1,J,2) = ZP(1,J,2)+C3*Z(3,J)
   15 IF ((ISLPSW/32)*2 .NE. (ISLPSW/16)) GO TO 16
      ZP(1,1,3) = ZXY11
      GO TO 17
   16 ZP(1,1,3) = C1*ZP(1,1,1)+C2*ZP(2,1,1)
      IF (M .GT. 2) ZP(1,1,3) = ZP(1,1,3)+C3*ZP(3,1,1)
   17 IF ((ISLPSW/128)*2 .NE. (ISLPSW/64)) GO TO 18
```

Figure A - 3 (Continued)

```
      ZXY1NS = ZXY1N
      GO TO 19
   18 ZXY1NS = C1*TEMP(N+1)+C2*TEMP(N+2)
      IF (M .GT. 2) ZXY1NS = ZXY1NS+C3*TEMP(N+3)
   19 IF ((ISLPSW/4)*2 .NE. (ISLPSW/2)) GO TO 21
      DO 20 J = 1,N
        NPMPJ = NPM+J
   20   TEMP(NPMPJ) = ZXM(J)
   21 IF ((ISLPSW/64)*2 .EQ. (ISLPSW/32) .AND.
     *     (ISLPSW/256)*2 .EQ. (ISLPSW/128)) GO TO 24
      DELXM = X(M)-X(MM1)
      DELXMM = DELXM+DELXM
      IF (M .GT. 2) DELXMM = X(M)-X(M-2)
      IF (DELXM .LE. 0. .OR. DELXMM .LE. DELXM) GO TO 47
      CALL CEEZ (-DELXM,-DELXMM,SIGMAX,C1,C2,C3,M)
      IF ((ISLPSW/4)*2 .EQ. (ISLPSW/2)) GO TO 24
      DO 22 J = 1,N
        NPMPJ = NPM+J
   22   TEMP(NPMPJ) = C1*Z(M,J)+C2*Z(MM1,J)
      IF (M .EQ. 2) GO TO 24
      DO 23 J = 1,N
        NPMPJ = NPM+J
   23   TEMP(NPMPJ) = TEMP(NPMPJ)+C3*Z(M-2,J)
   24 IF ((ISLPSW/64)*2 .NE. (ISLPSW/32)) GO TO 25
      ZP(M,1,3) = ZXYM1
      GO TO 26
   25 ZP(M,1,3) = C1*ZP(M,1,1)+C2*ZP(MM1,1,1)
      IF (M .GT. 2) ZP(M,1,3) = ZP(M,1,3)+C3*ZP(M-2,1,1)
   26 IF ((ISLPSW/256)*2 .NE. (ISLPSW/128)) GO TO 27
      ZXYMNS = ZXYMN
      GO TO 28
   27 ZXYMNS = C1*TEMP(NPM)+C2*TEMP(NPM-1)
      IF (M .GT. 2) ZXYMNS = ZXYMNS+C3*TEMP(NPM-2)
   28 DEL1 = Y(2)-Y(1)
      IF (DEL1 .LE. 0.) GO TO 47
      DELI = 1./DEL1
      DO 29 I = 1,M
   29   ZP(I,2,1) = DELI*(Z(I,2)-Z(I,1))
      ZP(1,2,3) = DELI*(ZP(1,2,2)-ZP(1,1,2))
      ZP(M,2,3) = DELI*(TEMP(NPM+2)-TEMP(NPM+1))
      CALL TERMS (DIAG1,SDIAG1,SIGMAY,DEL1)
      DIAGI = 1./DIAG1
      DO 30 I = 1,M
   30   ZP(I,1,1) = DIAGI*(ZP(I,2,1)-ZP(I,1,1))
      ZP(1,1,3) = DIAGI*(ZP(1,2,3)-ZP(1,1,3))
      ZP(M,1,3) = DIAGI*(ZP(M,2,3)-ZP(M,1,3))
```

Figure A - 3 (Continued)

ORIGINAL PAGE IS
OF POOR QUALITY

```
      TEMP(1) = DIAG1*SDIAG1
      IF (N .EQ. 2) GO TO 34
      DO 33 J = 2,NM1
        JM1 = J-1
        JP1 = J+1
        NPMPJ = NPM+J
        DEL2 = Y(JP1)-Y(J)
        IF (DEL2 .LE. 0.) GO TO 47
        DELI = 1./DEL2
        DO 31 I = 1,M
31        ZP(I,JP1,1) = DELI*(Z(I,JP1)-Z(I,J))
        ZP(1,JP1,3) = DELI*(ZP(1,JP1,2)-ZP(1,J,2))
        ZP(M,JP1,3) = DELI*(TEMP(NPMPJ+1)-TEMP(NPMPJ))
        CALL TERMS (DIAG2,SDIAG2,SIGMAY,DEL2)
        DIAGIN = 1./(DIAG1+DIAG2-SDIAG1*TEMP(JM1))
        DO 32 I = 1,M
32        ZP(I,J,1) = DIAGIN*(ZP(I,JP1,1)-ZP(I,J,1)-
     *                        SDIAG1*ZP(I,JM1,1))
        ZP(1,J,3) = DIAGIN*(ZP(1,JP1,3)-ZP(1,J,3)-
     *                      SDIAG1*ZP(1,JM1,3))
        ZP(M,J,3) = DIAGIN*(ZP(M,JP1,3)-ZP(M,J,3)-
     *                      SDIAG1*ZP(M,JM1,3))
        TEMP(J) = DIAGIN*SDIAG2
        DIAG1 = DIAG2
33      SDIAG1 = SDIAG2
34    DIAGIN = 1./(DIAG1-SDIAG1*TEMP(NM1))
      DO 35 I = 1,M
        NPI = N+I
35      ZP(I,N,1) = DIAGIN*(TEMP(NPI)-ZP(I,N,1)-
     *                      SDIAG1*ZP(I,NM1,1))
      ZP(1,N,3) = DIAGIN*(ZXY1NS-ZP(1,N,3)-
     *                    SDIAG1*ZP(1,NM1,3))
      TEMP(N) = DIAGIN*(ZXYMNS-ZP(M,N,3)-
     *                  SDIAG1*ZP(M,NM1,3))
      DO 37 J = 2,N
        JBAK = NP1-J
        JBAKP1 = JBAK+1
        T = TEMP(JBAK)
        DO 36 I = 1,M
36        ZP(I,JBAK,1) = ZP(I,JBAK,1)-T*ZP(I,JBAKP1,1)
        ZP(1,JBAK,3) = ZP(1,JBAK,3)-T*ZP(1,JBAKP1,3)
37      TEMP(JBAK) = ZP(M,JBAK,3)-T*TEMP(JBAKP1)
      DEL1 = X(2)-X(1)
      IF (DEL1 .LE. 0.) GO TO 47
      DELI = 1./DEL1
      DO 38 J = 1,N
```

Figure A - 3 (Continued)

```
      ZP(2,J,2) = DELI*(Z(2,J)-Z(1,J))
38    ZP(2,J,3) = DELI*(ZP(2,J,1)-ZP(1,J,1))
      CALL TERMS (DIAG1,SDIAG1,SIGMAX,DEL1)
      DIAG1 = 1./DIAG1
      DO 39 J = 1,N
        ZP(1,J,2) = DIAG1*(ZP(2,J,2)-ZP(1,J,2))
39      ZP(1,J,3) = DIAG1*(ZP(2,J,3)-ZP(1,J,3))
      TEMP(N+1) = DIAG1*SDIAG1
      IF (M  .EQ. 2) GO TO 43
      DO 42 I = 2,MM1
        IM1 = I-1
        IP1 = I+1
        NPI = N+I
        DEL2 = X(IP1)-X(I)
        IF (DEL2 .LE. 0.) GO TO 47
        DELI = 1./DEL2
        DO 40 J = 1,N
          ZP(IP1,J,2) = DELI*(Z(IP1,J)-Z(I,J))
40        ZP(IP1,J,3) = DELI*(ZP(IP1,J,1)-ZP(I,J,1))
        CALL TERMS (DIAG2,SDIAG2,SIGMAX,DEL2)
        DIAGIN = 1./(DIAG1+DIAG2-SDIAG1*TEMP(NPI-1))
        DO 41 J = 1,N
          ZP(I,J,2) = DIAGIN*(ZP(IP1,J,2)-ZP(I,J,2)-
     *                        SDIAG1*ZP(IM1,J,2))
41        ZP(I,J,3) = DIAGIN*(ZP(IP1,J,3)-ZP(I,J,3)-
     *                        SDIAG1*ZP(IM1,J,3))
        TEMP(NPI) = DIAGIN*SDIAG2
        DIAG1 = DIAG2
42      SDIAG1 = SDIAG2
43    DIAGIN = 1./(DIAG1-SDIAG1*TEMP(NPM-1))
      DO 44 J = 1,N
        NPMPJ = NPM+J
        ZP(M,J,2) = DIAGIN*(TEMP(NPMPJ)-ZP(M,J,2)-
     *                      SDIAG1*ZP(MM1,J,2))
44      ZP(M,J,3) = DIAGIN*(TEMP(J)-ZP(M,J,3)-
     *                      SDIAG1*ZP(MM1,J,3))
      DO 45 I = 2,M
        IBAK = MP1-I
        IBAKP1 = IBAK+1
        NPIBAK = N+IBAK
        T = TEMP(NPIBAK)
        DO 45 J = 1,N
          ZP(IBAK,J,2) = ZP(IBAK,J,2)-T*ZP(IBAKP1,J,2)
45        ZP(IBAK,J,3) = ZP(IBAK,J,3)-T*ZP(IBAKP1,J,3)
      RETURN
46    IERR = 1
```

Figure ^ - 3 (Continued)

```
      RETURN
   47 IERR = 2
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure A - 3 (Continued)

```
      FUNCTION SURFD2 (XX,YY,ZX,ZY,M,N,X,Y,Z,IZ,ZP,SIGMA)
C
C THIS FUNCTION INTERPOLATES A SURFACE AT A GIVEN COORDINATE
C PAIR USING A BI-SPLINE UNDER TENSION. THE SUBROUTINE SURF1
C SHOULD BE CALLED EARLIER TO DETERMINE CERTAIN NECESSARY
C PARAMETERS.
C
C ON INPUT--
C
C   XX AND YY CONTAIN THE X- AND Y-COORDINATES OF THE POINT
C   TO BE MAPPED ONTO THE INTERPOLATING SURFACE.
C
C   M AND N CONTAIN THE NUMBER OF GRID LINES IN THE X- AND
C   Y-DIRECTIONS, RESPECTIVELY, OF THE RECTANGULAR GRID
C   WHICH SPECIFIED THE SURFACE.
C
C   X AND Y ARE ARRAYS CONTAINING THE X- AND Y-GRID VALUES,
C   RESPECTIVELY, EACH IN INCREASING ORDER.
C
C   Z IS A MATRIX CONTAINING THE M * N FUNCTIONAL VALUES
C   CORRESPONDING TO THE GRID VALUES (I. E. Z(I,J) IS THE
C   SURFACE VALUE AT THE POINT (X(I),Y(J)) FOR I = 1,...,M
C   AND J = 1,...,N).
C
C   IZ CONTAINS THE ROW DIMENSION OF THE ARRAY Z AS DECLARED
C   IN THE CALLING PROGRAM.
C
C   ZP IS AN ARRAY OF 3*M*N LOCATIONS STORED WITH THE
C   VARIOUS SURFACE DERIVATIVE INFORMATION DETERMINED BY
C   SURF1.
C
C AND
C
C   SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED).
C
C THE PARAMETERS M, N, X, Y, Z, IZ, ZP, AND SIGMA SHOULD BE
C INPUT UNALTERED FROM THE OUTPUT OF SURF1.
C
C ON OUTPUT--
C
C   SURFD2 CONTAINS THE INTERPOLATED SURFACE VALUE.
C   ZX IS PARTIAL DERIVATIVE WITH RESPECT TO X.
C   ZY IS PARTIAL DERIVATIVE WITH RESPECT TO Y.
C
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C THIS FUNCTION REFERENCES PACKAGE MODULES INTRVL AND
```

Figure A - 3 (Continued)

ORIGINAL PAGE IS
OF POOR QUALITY

```
C SNHCSH.
C
C-----------------------------------------------------------------
      INTEGER M,N,IZ
      REAL XX,YY,ZX,ZY,X(M),Y(N),Z(IZ,N),ZP(M,N,3),SIGMA
      HERMZ (F1,F2,FP1,FP2) = (F2*DEL1+F1*DEL2)/DELS-DEL1*
     *                             DEL2*(FP2*(DEL1+DELS)+
     *                                   FP1*(DEL2+DELS))/
     *                             (6.*DELS)
      HERMD (F1,F2,FP1,FP2) = (F2-F1)/DELS
     *        +((2.*DEL1*DEL1-DEL2*(DEL1+DELS))*FP2
     *        -(2.*DEL2*DEL2-DEL1*(DEL2+DELS))*FP1)
     *        /(6.*DELS)
      HERMNZ (F1,F2,FP1,FP2,SIGMAP) = (F2*DEL1+F1*DEL2)/DELS
     *             +(FP2*(SINHM1*DEL2-DEL1*(2.*(COSHP1+1.)*
     *                               SINHP2+SIGMAP*COSHP1*DEL2))
     *              +FP1*(SINHM2*DEL1-DEL2*(2.*(COSHP2+1.)*
     *                               SINHP1+SIGMAP*COSHP2*DEL1))
     *             )/(SIGMAP*SIGMAP*DELS*(SINHMS+SIGMAP*DELS))
      HERMND (F1,F2,FP1,FP2,SIGMAP) = (F2-F1)/DELS
     *        +((DELS*SIGMAP*COSHM1-SINHMS)*FP2
     *        -(DELS*SIGMAP*COSHM2-SINHMS)*FP1)
     *        /(SIGMAP*SIGMAP*DELS*(SINHMS+SIGMAP*DELS))
      SIGMAX = ABS(SIGMA)*FLOAT(M-1)/(X(M)-X(1))
      SIGMAY = ABS(SIGMA)*FLOAT(N-1)/(Y(N)-Y(1))
      JM1 = INTRVL (YY,Y,N)
      J = JM1+1
      IM1 = INTRVL (XX,X,M)
      I = IM1+1
      DEL1 = YY-Y(JM1)
      DEL2 = Y(J)-YY
      DELS = Y(J)-Y(JM1)
      IF (SIGMAY .NE. 0.) GO TO 1
      ZIM1 = HERMZ(Z(I-1,J-1),Z(I-1,J),ZP(I-1,J-1,1),
     *                                      ZP(I-1,J,1))
      ZI = HERMZ(Z(I,J-1),Z(I,J),ZP(I,J-1,1),ZP(I,J,1))
      ZXXIM1 = HERMZ(ZP(I-1,J-1,2),ZP(I-1,J,2),
     *                  ZP(I-1,J-1,3),ZP(I-1,J,3))
      ZXXI = HERMZ(ZP(I,J-1,2),ZP(I,J,2),
     *                  ZP(I,J-1,3),ZP(I,J,3))
      ZYIM1 = HERMD (Z(I-1,J-1),Z(I-1,J),ZP(I-1,J-1,1),ZP(I-1,J,1))
      ZYI = HERMD (Z(I,J-1),Z(I,J),ZP(I,J-1,1),ZP(I,J,1))
      ZYXXIM = HERMD (ZP(I-1,J-1,2),ZP(I-1,J,2),ZP(I-1,J-1,3),
     *          ZP(I-1,J,3))
      ZYXXI = HERMD (ZP(I,J-1,2),ZP(I,J,2),ZP(I,J-1,3),ZP(I,J,3))
      GO TO 2
```

Figure A - 3 (Continued)

```
1 DELP1 = (DEL1+DELS)/2.
  DELP2 = (DEL2+DELS)/2.
  CALL SNHCSH (SINHM1,COSHM1,SIGMAY*DEL1,0)
  CALL SNHCSH (SINHM2,COSHM2,SIGMAY*DEL2,0)
  CALL SNHCSH (SINHMS,DUMMY,SIGMAY*DELS,-1)
  CALL SNHCSH (SINHP1,DUMMY,SIGMAY*DEL1/2.,-1)
  CALL SNHCSH (SINHP2,DUMMY,SIGMAY*DEL2/2.,-1)
  CALL SNHCSH (DUMMY,COSHP1,SIGMAY*DELP1,1)
  CALL SNHCSH (DUMMY,COSHP2,SIGMAY*DELP2,1)
  ZIM1 = HERMNZ(Z(I-1,J-1),Z(I-1,J),ZP(I-1,J-1,1),
 *              ZP(I-1,J,1),SIGMAY)
  ZI = HERMNZ(Z(I,J-1),Z(I,J),ZP(I,J-1,1),ZP(I,J,1),
 *            SIGMAY)
  ZXXIM1 = HERMNZ(ZP(I-1,J-1,2),ZP(I-1,J,2),
 *                ZP(I-1,J-1,3),ZP(I-1,J,3),SIGMAY)
  ZXXI = HERMNZ(ZP(I,J-1,2),ZP(I,J,2),
 *              ZP(I,J-1,3),ZP(I,J,3),SIGMAY)
  ZYIM1 = HERMND(Z(I-1,J-1),Z(I-1,J),ZP(I-1,J-1,1),ZP(I-1,J,1),
 *        SIGMAY)
  ZYI = HERMND(Z(I,J-1),Z(I,J),ZP(I,J-1,1),ZP(I,J,1),SIGMAY)
  ZYXXIM = HERMND(ZP(I-1,J-1,2),ZP(I-1,J,2),ZP(I-1,J-1,3),
 *         ZP(I-1,J,3),SIGMAY)
  ZYXXI = HERMND(ZP(I,J-1,2),ZP(I,J,2),ZP(I,J-1,3),ZP(I,J,3),
 *        SIGMAY)
2 DEL1 = XX-X(IM1)
  DEL2 = X(I)-XX
  DELS = X(I)-X(IM1)
  IF (SIGMAX .NE. 0.) GO TO 3
  SURFD2 = HERMZ(ZIM1,ZI,ZXXIM1,ZXXI)
  ZX = HERMD (ZIM1,ZI,ZXXIM1,ZXXI)
  ZY = HERMZ (ZYIM1,ZYI,ZYXXIM,ZYXXI)
  RETURN
3 DELP1 = (DEL1+DELS)/2.
  DELP2 = (DEL2+DELS)/2.
  CALL SNHCSH (SINHM1,COSHM1,SIGMAX*DEL1,0)
  CALL SNHCSH (SINHM2,COSHM2,SIGMAX*DEL2,0)
  CALL SNHCSH (SINHMS,DUMMY,SIGMAX*DELS,-1)
  CALL SNHCSH (SINHP1,DUMMY,SIGMAX*DEL1/2.,-1)
  CALL SNHCSH (SINHP2,DUMMY,SIGMAX*DEL2/2.,-1)
  CALL SNHCSH (DUMMY,COSHP1,SIGMAX*DELP1,1)
  CALL SNHCSH (DUMMY,COSHP2,SIGMAX*DELP2,1)
  SURFD2 = HERMNZ(ZIM1,ZI,ZXXIM1,ZXXI,SIGMAX)
  ZX = HERMND (ZIM1,ZI,ZXXIM1,ZXXI,SIGMAX)
  ZY = HERMNZ (ZYIM1,ZYI,ZYXXIM,ZYXXI,SIGMAX)
  RETURN
  END
```

Figure A - 3 (Continued)

```
      SUBROUTINE SNHCSH (SINHM,COSHM,X,ISW)
C
C THIS SUBROUTINE RETURNS APPROXIMATIONS TO
C        SINHM(X) = SINH(X)-X
C        COSHM(X) = COSH(X)-1
C AND
C        COSHMM(X) = COSH(X)-1-X*X/2
C WITH RELATIVE ERROR LESS THAN 3.42E-14
C
C ON INPUT--
C
C   X CONTAINS THE VALUE OF THE INDEPENDENT VARIABLE.
C
C   ISW INDICATES THE FUNCTION DESIRED
C          = -1 IF ONLY SINHM IS DESIRED.
C          =  0 IF BOTH SINHM AND COSHM ARE DESIRED.
C          =  1 IF ONLY COSHM IS DESIRED.
C          =  2 IF ONLY COSHMM IS DESIRED.
C          =  3 IF BOTH SINHM AND COSHMM ARE DESIRED.
C
C ON OUTPUT--
C
C   SINHM CONTAINS THE VALUE OF SINHM(X) IF ISW .LE. 0 OR
C   ISW .EQ. 3 (SINHM IS UNALTERED IF ISW .EQ.1 OR ISW .EQ.
C   2).
C
C   COSHM CONTAINS THE VALUE OF COSHM(X) IF ISW .EQ. 0 OR
C   ISW .EQ. 1 AND CONTAINS THE VALUE OF COSHMM(X) IF ISW
C   .GE. 2 (COSHM IS UNALTERED IF ISW .EQ. -1).
C
C AND
C
C   X AND ISW ARE UNALTERED.
C
C------------------------------------------------------------------
      INTEGER ISW
      REAL SINHM,COSHM,X
      DATA SP4/4.50227693331333E-07/,
     *     SP3/8.95278544215320E-06/,
     *     SP2/8.72064975791602E-04/,
     *     SP1/4.36314556981690E-02/,
     *     SQ1/-6.36254430175110E-03/
      DATA CP4/1.78419556749014E-07/,
     *     CP3/2.87277223799044E-05/,
     *     CP2/2.15151519902028E-03/,
     *     CP1/7.58181822562556E-02/,
```

Figure A - 3 (Continued)

```
     *      CQ1/-7.51515105679867E-03/
      DATA ZP3/5.592971162647208E-07/,
     *      ZP2/1.779434880308946E-04/,
     *      ZP1/1.698004618947926E-02/,
     *      ZQ4/1.334125354923376E-09/,
     *      ZQ3/-5.808589441386638E-07/,
     *      ZQ2/1.278149644038636E-04/,
     *      ZQ1/-1.635323714331815E-02/
      XX = X
      AX = ABS(XX)
      XS = XX*XX
      IF ((AX .GE. 2.70) .OR. (AX .GE. 1.15 .AND.
     *    ISW .NE. 2)) EXPX = EXP(AX)
      IF (ISW .EQ. 1 .OR. ISW .EQ. 2) GO TO 2
      IF (AX .GE. 1.15) GO TO 1
      SINHM = (((((SP4*XS+SP3)*XS+SP2)*XS+SP1)*XS+1.)*XS*XX)
     *        /((SQ1*XS+1.)*6.)
      GO TO 2
    1 SINHM = -(((1./EXPX+AX)+AX)-EXPX)/2.
      IF (XX .LT. 0.) SINHM = -SINHM
    2 IF (ISW .NE. 0 .AND. ISW .NE. 1) GO TO 4
      IF (AX .GE. 1.15) GO TO 3
      COSHM = (((((CP4*XS+CP3)*XS+CP2)*XS+CP1)*XS+1.)*XS)
     *        /((CQ1*XS+1.)*2.)
      GO TO 4
    3 COSHM = ((1./EXPX-2.)+EXPX)/2.
    4 IF (ISW .LE. 1) RETURN
      IF (AX .GE. 2.70) GO TO 5
      COSHM = ((((ZP3*XS+ZP2)*XS+ZP1)*XS+1.)*XS*XS)/(((((ZQ4
     *        *XS+ZQ3)*XS+ZQ2)*XS+ZQ1)*XS+1.)*24.)
      RETURN
    5 COSHM = (((1./EXPX-2.)-XS)+EXPX)/2.
      RETURN
      END
      SUBROUTINE TERMS (DIAG,SDIAG,SIGMA,DEL)
C
C THIS SUBROUTINE COMPUTES THE DIAGONAL AND SUPERDIAGONAL
C TERMS OF THE TRIDIAGONAL LINEAR SYSTEM ASSOCIATED WITH
C SPLINE UNDER TENSION INTERPOLATION.
C
C ON INPUT--
C
C   SIGMA CONTAINS THE TENSION FACTOR.
C
C AND
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure A - 3 (Continued)

```
C    DEL CONTAINS THE STEP SIZE.
C
C ON OUTPUT--
C
C                 (SIGMA*DEL*COSH(SIGMA*DEL) - SINH(SIGMA*DEL)
C    DIAG = DEL*--------------------------------------------.
C                       (SIGMA*DEL)**2 * SINH(SIGMA*DEL)
C
C                    SINH(SIGMA*DEL) - SIGMA*DEL
C    SDIAG = DEL*----------------------------------.
C                  (SIGMA*DEL)**2 * SINH(SIGMA*DEL)
C
C AND
C
C    SIGMA AND DEL ARE UNALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULE SNHCSH.
C
C-----------------------------------------------------------------
      REAL DIAG,SDIAG,SIGMA,DEL
      IF (SIGMA .NE. 0.) GO TO 1
      DIAG = DEL/3.
      SDIAG = DEL/6.
      RETURN
    1 SIGDEL = SIGMA*DEL
      CALL SNHCSH (SINHM,COSHM,SIGDEL,0)
      DENOM = DEL/((SINHM+SIGDEL)*SIGDEL*SIGDEL)
      DIAG = DENOM*(SIGDEL*COSHM-SINHM)
      SDIAG = DENOM*SINHM
      RETURN
      END
      SUBROUTINE CEEZ (DEL1,DEL2,SIGMA,C1,C2,C3,N)
C
C THIS SUBROUTINE DETERMINES THE COEFFICIENTS C1, C2, AND C3
C USED TO DETERMINE ENDPOINT SLOPES. SPECIFICALLY, IF
C FUNCTION VALUES Y1, Y2, AND Y3 ARE GIVEN AT POINTS X1, X2,
C AND X3, RESPECTIVELY, THE QUANTITY C1*Y1 + C2*Y2 + C3*Y3
C IS THE VALUE OF THE DERIVATIVE AT X1 OF A SPLINE UNDER
C TENSION (WITH TENSION FACTOR SIGMA) PASSING THROUGH THE
C THREE POINTS AND HAVING THIRD DERIVATIVE EQUAL TO ZERO AT
C X1. OPTIONALLY, ONLY TWO VALUES, C1 AND C2 ARE DETERMINED.
C
C ON INPUT--
C
C    DEL1 IS X2-X1 (.GT. 0.).
C
```

Figure A - 3 (Continued)

```
C    DEL2 IS X3-X1 (.GT. 0.). IF N .EQ. 2, THIS PARAMETER IS
C    IGNORED.
C
C    SIGMA IS THE TENSION FACTOR.
C
C AND
C
C    N IS A SWITCH INDICATING THE NUMBER OF COEFFICIENTS TO
C    BE RETURNED. IF N .EQ. 2 ONLY TWO COEFFICIENTS ARE
C    RETURNED. OTHERWISE ALL THREE ARE RETURNED.
C
C ON OUTPUT--
C
C    C1, C2, AND C3 CONTAIN THE COEFFICIENTS.
C
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C THIS SUBROUTINE REFERENCES PACKAGE MODULE SNHCSH.
C
C-----------------------------------------------------------------
      REAL DEL1,DEL2,SIGMA,C1,C2,C3
      IF (N .EQ. 2) GO TO 2
      IF (SIGMA .NE. 0.) GO TO 1
      DEL = DEL2-DEL1
      C1 = -(DEL1+DEL2)/(DEL1*DEL2)
      C2 = DEL2/(DEL1*DEL)
      C3 = -DEL1/(DEL2*DEL)
      RETURN
    1 CALL SNHCSH (DUMMY,COSHM1,SIGMA*DEL1,1)
      CALL SNHCSH (DUMMY,COSHM2,SIGMA*DEL2,1)
      DELP = SIGMA*(DEL2+DEL1)/2.
      DELM = SIGMA*(DEL2-DEL1)/2.
      CALL SNHCSH (SINHMP,DUMMY,DELP,-1)
      CALL SNHCSH (SINHMM,DUMMY,DELM,-1)
      DENOM = COSHM1*(DEL2-DEL1)-2.*DEL1*(SINHMP+DELP)*
     *        (SINHMM+DELM)
      C1 = 2.*(SINHMP+DELP)*(SINHMM+DELM)/DENOM
      C2 = -COSHM2/DENOM
      C3 = COSHM1/DENOM
      RETURN
    2 C1 = -1./DEL1
      C2 = -C1
      RETURN
      END
      FUNCTION INTRVL (T,X,N)
C
```

ORIGINAL PAGE IS OF POOR QUALITY

Figure A - 3 (Continued)

```
C THIS FUNCTION DETERMINES THE INDEX OF THE INTERVAL
C (DETERMINED BY A GIVEN INCREASING SEQUENCE) IN WHICH
C A GIVEN VALUE LIES.
C
C ON INPUT--
C
C   T IS THE GIVEN VALUE.
C
C   X IS A VECTOR OF STRICTLY INCREASING VALUES.
C
C AND
C
C   N IS THE LENGTH OF X (N .GE. 2).
C
C ON OUTPUT--
C
C   INTRVL RETURNS AN INTEGER I SUCH THAT
C
C            I = 1         IF            T .LE. X(2)   .
C            I = N-1       IF X(N-1) .LE. T            .
C            OTHERWISE        X(I)   .LE. T .LE. X(I+1).
C
C NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C-----------------------------------------------------------------
      INTEGER N
      REAL T,X(N)
      TT = T
      IF (TT .LE. X(2)) GO TO 4
      IF (TT .GE. X(N-1)) GO TO 5
      IL = 2
      IH = N-1
    1 I = IL+IFIX(FLOAT(IH-IL)*(TT-X(IL))/(X(IH)-X(IL)))
      IF (TT .LT. X(I)) GO TO 2
      IF (TT .LE. X(I+1)) GO TO 3
      IL = I+1
      GO TO 1
    2 IH = I
      GO TO 1
    3 INTRVL = I
      RETURN
    4 INTRVL = 1
      RETURN
    5 INTRVL = N-1
      RETURN
      END
```

ORIGINAL PAGE IS OF POOR QUALITY

Figure A - 3 (Continued)