ESD-TR-68-213
ESTI FILE COPY

ESD RECORD COPY
RETURN TO
SCIENTIFIC & TECHNICAL INFORMATION DIVISION
(ESTI), BUILDING 1211

ESD-TR-68-213

A COMPUTER PROGRAM FOR BACKSCATTER BY SMOOTHLY JOINED, SECOND DEGREE SURFACES OF REVOLUTION - 2430-6

C.E. Ryan, Jr.

ESD ACCESSION LIST
ESTI Call No. AL 60773
Copy No. 1 of 2 cys.

April 1968

DEPUTY FOR SURVEILLANCE AND CONTROL SYSTEMS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

lof 2
ESSXS

This document has been approved for public release and sale; its distribution is unlimited

(Prepared for Contract No. AF 19(628)-67-C-0308 by The Ohio State University, ElectroScience Laboratory, Department of Electrical Engineering, 1320 Kinnear Road, Columbus, Ohio.)

AD669807

LEGAL NOTICE

When U. S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

ABSTRACT

A computer program for calculation of the echo area of smoothly joined, N section convex conducting surfaces of revolution, described by a second degree equation is presented. For the case of E_θ (parallel) polarization of the incident and scattered fields the solution is obtained by a combination of geometrical optics and creeping wave theory. For the case of E_ϕ (perpendicular) polarization the solution is obtained using geometrical optics, and the creeping wave is neglected. The computed results for E_θ polarization are in good agreement with measurements for prolate spheroids, prolate spheroid-sphere, and prolate spheroid-oblate spheroid combinations.

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	TARGET DESCRIPTION	1
III.	THE CREEPING WAVE COMPUTER PROGRAM	2
IV.	FUNCTIONS AND SUBROUTINES	6
V.	DATA	8
VI.	CONCLUSION	8
APPENDIX I -	THE COMPUTER PROGRAM LISTING	9
APPENDIX II -	THE COMPUTER FLOW DIAGRAM	17
REFERENCES		25

A COMPUTER PROGRAM FOR BACKSCATTER BY SMOOTHLY JOINED, SECOND DEGREE SURFACES OF REVOLUTION

I. INTRODUCTION

The computer program listed in Appendix I, applies the geometrical optics and creeping wave solutions described in Ref. 1, 2 to obtain the backscattered fields of a surface of revolution. The target is composed of N sections, each section described by a second degree equation. In addition the target must be convex, and be smoothly joined at the boundaries between sections. The program as listed computes the backscattered field and echo area for E_θ (parallel) polarization as a function of the incidence angle, for a given wavelength. The geometrical optics backscattered field is independent of polarization, thus the geometrical optics scattered field for the case of E_ϕ (perpendicular) polarization is also obtained. The creeping wave scattered field has not been included for E_ϕ polarization due to the difficulty in obtaining the ray path of the creeping wave in this case.[1] This program has been tested for E_θ polarization for prolate spheroid, prolate spheroid-sphere, and prolate spheroid-oblate spheroid targets. The results of these tests are presented in Ref. 1.

II. TARGET DESCRIPTION

The surface is described in each section by the second degree equation

$$F(r,\theta) = A_1 r^2 \sin^2\theta + A_2 r^2 \cos^2\theta + A_3 r^2 \sin\theta\cos\theta + A_4 r\cos\theta + A_5 r \sin\theta + A_6 = 0 \quad . \tag{1}$$

The constants $A_1 \ldots A_6$ specify the surface in the section (i) bounded by the angles θ_i and θ_{i+1}. The surface which may be represented by Eq. (1) includes the sphere, prolate and oblate spheroids, and the ogive. In general any surface derived from a conic section can be described by this equation. More complex surfaces may be represented by using a large number of sections and approximating the desired surface by a second degree surface within each section. This program provides for 20 sections but this number can be readily increased.

A restriction on the target specification, which is a result of the method for finding the specular point, is that the coordinate origin be located such that the normal to the surface at z = 0 be r directed. That is, $dr/d\theta|_{z=0} = 0$. Discontinuities in the derivatives $(d^n r/d\theta^n)$ at the junctions between sections may exist. However, the effects of such discontinuities as well as the effects of tips are not included in this program. A discontinuity in the first derivative $(dr/d\theta)$ at the junction ("wedge" discontinuities) causes diffraction, and can be evaluated using wedge diffraction techniques.[1, 2] A discontinuity in the second derivative $(d^2 r/d\theta^2)$ at the junction results in reflection and transmission of an incident creeping wave at such a discontinuity. An example of this effect, the spherically capped ogive, has been discussed in Refs. 1 and 2. As the effects of a discontinuity in the first derivative are significent this program may not give good results for such a target. The creeping wave reflection effect due to a moderate second derivative discontinuity may be neglected with good results.[1] Thus good results for the scattered field of perfectly conducting convex, smoothly joined surfaces of revolution may be obtained using this program.

III. THE CREEPING WAVE COMPUTER PROGRAM

The creeping wave computer program given in Appendix I uses geometrical optics and creeping wave theory to calculate the backscattered field of the target. The geometrical optics field is obtained by identifying the specular point and calculating the Gaussian curvature at the specular point. The backscattered field is then calculated as given in Ref. 1. The creeping wave backscattered field is obtained by identifying the points of attachment and reradiation of the creeping wave, calculating the diffraction coefficient at these points, and by computing the attenuation of the creeping wave on a path defined by the E-plane of the target. The backscattered field is then calculated as given in Ref. 1. These contributions are added to obtain the total backscattered field.

Referring to the computer program listing shown in Appendix I, the function of the significant sections of the program will be discussed. The card numbers associated with each section will be specified. This discussion, together with the comment cards included in the program listing, is intended to give sufficient information about the program to enable a qualified programmer to both use and modify the program. Statements which are in common use in Fortran IV, such as DIMENSION, COMPLEX, and FORMAT statements will not be discussed as it is assumed that the reader has a knowledge of Fortran IV.

The COMMON declaration (0011) is used to store the constants required in Eq. (1) in the common block labelled /DATA/. This common block is used in conjunction with the unlabelled common block to transfer a particular set of constants A1 (I) to A6 (I) into the unlabelled common regions shared by the subroutines. This provision reduces the number of calling variables required by each subroutine.

The COMMON declaration (0013) is used to store and manipulate the angles corresponding to the section boundaries.

The statements 0018-0022 initialize constants which are required in the calculations.

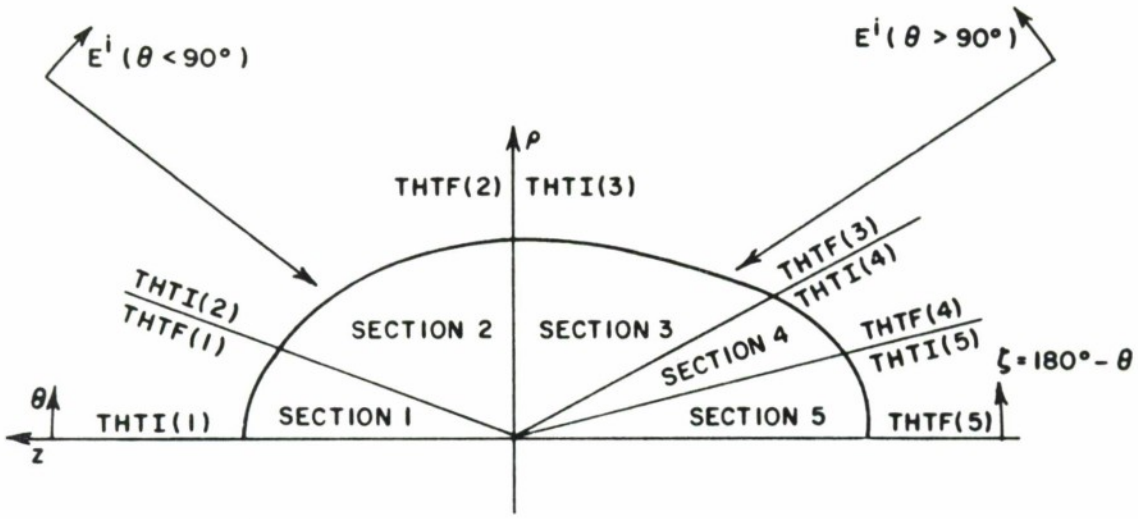
The READ statements 0023-0027 read the required data, and provision is also made to write out this data for the purpose of identification.

The statements 0028-0030 set up the incrementation of the incidence angle THT and the statement 0031 calculates the propagation factor FK.

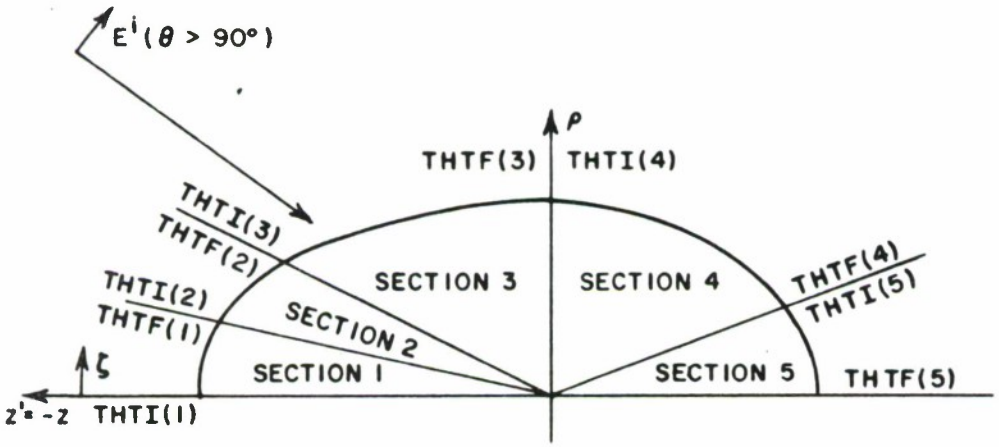
Next the loop for incrementing the incidence angle is entered, and the angle converted to radians. The logic statements (0036, 0037) serve two purposes if the incidence angle exceeds 90°. First the incidence angle is constrained to the range $0. < THT < \pi/2.$ Then the subroutine FNVRT (N) is called. The subroutine FNVRT (N) inverts the target, i.e., the portion of the target defined as section #1 becomes section #N, section #2 becomes section #(N-1) and so forth. This provision is necessary so that the propagation direction of the creeping waves which start at the point of attachment is always the positive angular direction. This provision simplifies the logic required to compute the creeping wave contributions. Figure 1 illustrates this provision.

The propagation vector and polarization vector of the incident wave are computed next (0038-0046). These vectors are used in the determination of the specular point and the points of attachment and reradiation of the creeping wave.

Next, the location of the specular point is determined (0047-0060). This is done by incrementing by DTHB along the surface of the target in the $\phi = 0$ plane, performing the scalar product of the propagation vector and the surface normal vector at each point, and by finding the point at which this product is a maximum. This point, in section NSP, is the specular point (THSPP).



(a) A 5 SECTION TARGET



(b) THE INVERTED TARGET

Fig. 1. A target (a) and the inversion (b).

The location of the points of attachment and reradiation of the creeping wave for $\phi = 0, \pi$ is determined next (0062-0090). This is done in the same manner as the determination of the specular point location except that the scalar product of the polarization and normal vectors is used. After these points have been found they are written out together with the specular point (0091, 0092).

Having determined the location of the specular point (NSP, THSPP, RSP) the Gaussian curvature at the specular point is computed and used to calculate the geometrical optics scattered field (0093-0103).

The creeping wave path length is computed in 0104-0129. This is done by starting at the attachment point (THCWL, $\phi = 0$) and performing a numerical integration of the differential arc length along the surface to the point $\theta = \pi$. The length thus determined is CWL1. This process is then repeated for the attachment point (THCWL, $\phi = \pi$) with a result CWL2. The product of the propagation factor and the sum of the path lengths then gives the free space phase of the creeping wave which propagates around the target.

Next the complex attenuation of the creeping wave is computed (0130-0150). The same process used to compute the path length is applied, with the exception that the complex attenuation coefficient must be integrated. This is accomplished using the EXTERNAL ALPHDS as the integration function.

Having obtained the creeping wave path length, attenuation, and points of attachment and reradiation, it is easy to calculate the creeping wave scattered field. This is done in 0151-0161. The phase is first calculated and the square of the diffraction coefficient determined. Then the creeping wave scattered field ECW is computed.

The total scattered field and echo area in square wavelengths are computed in 0162-0165. Next the common regions are reset if THTD $> 90^{\circ}$ and the incidence angle in degrees, the geometrical optics, creeping wave, total scattered field and echo area are written. At the completion of the loop (0169) the program is terminated.

IV. FUNCTIONS AND SUBROUTINES

In addition to the arguments required to call each subroutine or function as given in the description below, it is necessary to transfer the constants in Eq. (1) for the section (I) from the /DATA/ common block to the unlabelled common block. All function and subroutines in which these COMMON statements appear require that such a transfer be made. This provision reduces the number of arguments required in the functions and subroutines.

Complex Function DELSP(THT) 0172-0179

This function calculates the incremental arc length in the θ direction at THT. Although the arc length is a real number this function is declared complex so that it may be used as an external function in the numerical integration.

Subroutine FNVRT(N) 0180-0195

This subroutine interchanges the data which specifies the target, thus inverting the target by 180° in θ.

Complex Function SPHASE(THTI, PHII, THTB, PHIB, RB, FK) 0196-0204

This function determines the phase of the incident field (THTI, PHII) at a point on the target (RB, THTB, PHIB) for a propagation factor FK.

Complex Function PHASE(THTI, PHII, THTB, PHIB, RB, FK) 0205-0214

This function determines the backscattered phase of the field at (RB, THTB, PHIB) for an incident field (THTI, PHII) and propagation factor FK.

Complex Function ALPH (RH01, RH02, WAVE) 0215-0230

This function computes the complex creeping wave attenuation coefficient for the orthogonal radii of curvature RH01 and RH02 and a wavelength WAVE. RH01 is the radius of curvature in the propagation direction.

Complex Function DSQ (RI1, RSI, WAVE) 0231-0243

This function computes the square of the creeping wave diffraction coefficient as a function of the radii of curvature RI1, RS1 in the propagation direction and the wavelength WAVE.

Complex Function ALPHDS(THT) 0244-0260

This function determines the product of the creeping wave attenuation coefficient and the metric of the surface which is needed in the integration for the attenuation along the path.

Subroutine DIFFGO(R, THT, RTHT, RTTH, ECAP, FCAP, GCAP, ELC, FLC, GLC) 0261-0301

This subroutine calculates the first (RTHT) and second (RTTH) derivatives of the distance (R) from the origin to the surface at the point (R, THT). In addition the coefficients of the first and second fundamental forms of differential geometry of the surface ECAP, FCAP, GCAP, ELC, FLC, GLC are computed.[1]

Subroutine FSPDT(RSP, THSP, FNT, THTS, THTF, DTHT, PHI, VX, VY, VZ) 0302-0326

This subroutine finds the point (RSP, THSP) in the section FNT bounded by THTS and THTF where the scalar product of the surface normal vector and the vector (VX, VY, VZ) is a maximum. This is done by incrementing in angle by DTHT and selecting the largest scalar product.

Subroutine FINT (SSS, FCTI, FLL, FUL, ERRR, NX) 0327-0377

This is a subroutine for numerical integration of the complex external function FCTI between the limits FLL and FUL. The complex result is returned in SSS. ERRR specifies the percent error desired and the integer NX determines whether equal integration increments (NX = 1) or adjusted increments (NX = 2) are used. A description of this integration technique is given in Ref. 3.

Subroutine FNORM (FNVX, FNVY, FNVZ, R, THT, PHI) 0378-0400

The subroutine FNORM calculates the surface normal vector (FNVX, FNVY, FNVZ) at the point on the surface described by the spherical coordinates R, THT, PHI.

Function RAD(THT) 0401-0419

This function computes the distance from the origin to the point on the surface at the angle THT.

Subroutine FCOMM(I) 0420-0431

This subroutine transfers the constants required by Eq. (1) for the I^{th} section from the /DATA/ common storage block to the unlabelled common block.

V. DATA

A set of input data is shown in Fig. 2. The order of cards is as follows:

Card 1 - specifies the number of sections (N) in the target.
Card 2 - specifies the wavelength and the increment in incidence angle.
Next N cards - specify the initial and final angular boundaries of the section in radians and the constants required in Eq. (1).

The particular target specified by this data is the prolate spheroid-oblate spheroid combination described in Ref. 1.

```
00002
  1.0          2.0
  0.        1.570796    2.4799     .5102        0.          0.          0.
1.570796    3.1415927   2.4799      5.0955      0.          0.          0.
```

Fig. 2. The input data.

VI. CONCLUSION

A computer program for backscatter by smoothly joined, second degree surfaces of revolution has been developed using the theory presented in Refs. 1 and 2. This program has been tested for prolate spheroids, prolate spheroid-sphere, and prolate spheroid-oblate spheroid combinations with good results.[1] The description of the computer program given in this report should enable a capable programmer to use and modify this program.

The creeping wave computer program was originally intended to be integrated with the computer program based upon wedge diffraction techniques.[1,4] This goal can be obtained by modification of the logic sections of the two programs, and by converting the computational sections of each of the programs to a subroutine form. Thus a program utilizing both wedge diffraction and creeping wave theory can be assembled.

APPENDIX I
THE COMPUTER PROGRAM LISTING

```
$EXECUTE        IBJOB                                                          0000
$IBJOB          GO,MAP                                                         0001
$IBFTC CREEP    LIST,NODECK                                                    0002
  1    FORMAT(1I5)                                                            0003
 2     FORMAT(2F10.5)                                                         0004
  3    FORMAT(8F10.5)                                                         0005
 4     FORMAT(3F15.8)                                                         0006
 5     FORMAT(7F15.8)                                                         0007
   6   FORMAT(7H THSPP=F15.8,6H THCWU=F15.8,7H THCWL=F15.8)                   0008
   7   FORMAT(5H NSP=I15,6H NSWU=I15,6H NCWL=I15)                             0009
  8    FORMAT(6H CWL1=F15.8,6H CWL2=F15.8)                                    0010
       COMMONRA1,RA3,RB1,RA9,RA10,RA11,WAVE/DATA/AR1(20),AR3(20),BR1(20),     0011
      CAR9(20),AR10(20),AR11(20)                                              0012
       COMMON/ANGL/THT1(20),THTF(20),THTP(20)                                 0013
       COMPLEX PALP1,PALP2                                                    0014
       COMPLEX PHASP,EGO,ALPHDS,ATTEN,ALP1,ALP2,PHCW1,PHCW2,DSQC,ECW,ETOT     0015
       COMPLEX CPTHL,PHASE,ALPH,DSQ,DELSP,CML1,CML2,CEXP                      0016
       COMPLEX SPHASE                                                         0017
       DEGRAD=0.01745329                                                      0018
       RADEG=57.29578                                                         0019
       PI=3.1415927                                                           0020
        PI2=PI/2.                                                             0021
       TP=2.*PI                                                               0022
       READ(5,1) N                                                            0023
       READ(5,2)WAVE,DTHT                                                     0024
       WRITE(6,2)WAVE,DTHT                                                    0025
       READ(5,3)(THTI(IR),THTF(IR),AR1(IR),AR3(IR),BR1(IR),AR9(IR),AR10(I     0026
      CR),AR11(IR),IR=1,N)                                                    0027
       DTHB=DEGRAD                                                            0028
       FNT=180./DTHT                                                          0029
       NT=FNT-1.                                                              0030
       FK=TP/WAVE                                                             0031
       DO 100 NTH=1,NT,1                                                      0032
       FNTH=NTH                                                               0033
       THTD=FNTH*DTHT                                                         0034
       THT=DEGRAD*THTD                                                        0035
       IF(THTD.GT.90.) THT=PI-THT                                             0036
       IF(THTD.GT.90.) CALL FNVRT(N)                                          0037
C      CALCULATE INCIDENCE VECTOR                                             0038
       PHI=0.                                                                 0039
       VIX=-SIN(THT)                                                          0040
       VIZ=-COS(THT)                                                          0041
       VIY=0.                                                                 0042
C      CALCULATE INCIDENT E-VECTOR-PARALLEL POL.                              0043
       EIX=COS(THT)                                                           0044
       EIY=0.                                                                 0045
       EIZ=-SIN(THT)                                                          0046
C      DETERMINE SPECULAR POINT                                               0047
       FNSPP=0.                                                               0048
       NSP=0                                                                  0049
       RSPP=0.                                                                0050
       THSPP=0.                                                               0051
       DO410 ISP=1,N,1                                                        0052
       CALL FCOMM(ISP)                                                        0053
       CALL FSPDT(RSP,THSP,FNSP,THT1(ISP),THTF(ISP),DTHB,0.,VIX,VIY,VIZ)      0054
       IF(FNSPP.GT.FNSP.OR.THSP.GT.PI2)GOTO 411                               0055
       FNSPP=ABS(FNSP)                                                        0056
       RSPP=RSP                                                               0057
       THSPP=THSP                                                             0058
       NSP=ISP                                                                0059
 411   CONTINUE                                                               0060
 410   CONTINUE                                                               0061
C      DETERMINE CREEPING WAVE POINTS                                         0062
       FCWU=0.                                                                0063
```

```
      NCWU=0                                                                0064
      RSCWU=0.                                                              0065
      THCWU=0.                                                              0066
      DO 510 ICW=I,N,I                                                      0067
      CALL FCOMM(ICW)                                                       0068
      CALL FSPDT(RCWU,THWU,FWU,THTI(ICW),THTF(ICW),DTHB,0.,EIX,EIY,EIZ)     0069
      IF(FCWU.GT.FWU.OR.THWU.LT.PI2)GO TO 511                               0070
      FCWU=ABS(FWU)                                                         0071
      RSCWU=RCWU                                                            0072
      THCWU=THWU                                                            0073
      NCWU=ICW                                                              0074
  511 CONTINUE                                                              0075
  510 CONTINUE                                                              0076
      FCWL=0.                                                               0077
      NCWL=0                                                                0078
      RSCWL=0.                                                              0079
      THCWL=0.                                                              0080
      DO 520 ICW=I,N,1                                                      0081
      CALL FCOMM(ICW)                                                       0082
      CALL FSPDT(RCWL,THWL,FWL,THTI(ICW),THTF(ICW),DTHB,PI,EIX,EIY,EIZ)     0083
      IF(FCWL.GT.FWL.OR.THWL.GT.PI2) GO TO 521                              0084
      FCWL=FWL                                                              0085
      RSCWL=RCWL                                                            0086
      THCWL=THWL                                                            0087
      NCWL=ICW                                                              0088
  521 CONTINUE                                                              0089
  520 CONTINUE                                                              0090
      WRITE(6,7) NSP,NCWU,NCWL                                              0091
      WRITE(6,6)THSPP,THCWU,THCWL                                           0092
C     CALCULATE GEOMETRICAL OPTICS TERM                                     0093
      CALL FCOMM(NSP)                                                       0094
      CALL DIFFGO(RSPP,THSPP,RDUM,RDDM,ECAP,FCAP,GCAP,ELC,FLC,GLC)          0095
      IF(GCAP) 20,21,20                                                     0096
   21 GAUSS=ELC/ECAP                                                        0097
      GAUSS=GAUSS*GAUSS                                                     0098
      GO TO 22                                                              0099
   20 GAUSS=(ELC*GLC-FLC*FLC)/(ECAP*GCAP-FCAP*FCAP)                         0100
   22 CONTINUE                                                              0101
      PHASP=PHASE(THT,0.,THSPP,0.,RSPP,FK)                                  0102
      EGO=-SQRT(1./GAUSS)*PHASP/2.                                          0103
C     CALCULATE CREEPING WAVE PATH LENGTH                                   0104
      EXTERNAL DELSP                                                        0105
      CWL1=0.                                                               0106
      DO 522 NCPU=NCWU,N,I                                                  0107
      CALL FCOMM(NCPU)                                                      0108
      TCW1=THCWU                                                            0109
      TCW2=THTF(NCPU)                                                       0110
      IF(NCPU.GT.NCWU) TCW1=THTI(NCPU)                                      0111
      CALL FINT(CML1,DELSP,TCW1,TCW2,5.0,2)                                 0112
      RCML1=REAL(CML1)                                                      0113
      CWL1=CWLI+RCML1                                                       0114
  522 CONTINUE                                                              0115
      CWL2=0.                                                               0116
      DO 524 NCPL=NCWL,N,1                                                  0117
      CALL FCOMM(NCPL)                                                      0118
      TCW1=THCWL                                                            0119
      TCW2=THTF(NCPU)                                                       0120
      IF(NCPL.GT.NCWL) TCW1=THTI(NCPL)                                      0121
      CALL FINT(CML2,DELSP,TCW1,TCW2,5.0,2)                                 0122
      RCML2=REAL(CML2)                                                      0123
      CWL2=CLW2+RCML2                                                       0124
  524 CONTINUE                                                              0125
      WRITE(6,8) CWL1,CWL2                                                  0126
      FKL1=FK*CWL1                                                          0127
```

```
      FKL2=FK*CWL2                                                        0128
      FKLCW=FKLI+FKL2                                                     0129
C     CALCULATE CREEPING WAVE ATTENUATION                                 0130
      EXTERNAL ALPHDS                                                     0131
      ALPI=(0.,0.)                                                        0132
      DO 530 NCPU=NCWU,N,I                                                0133
      CALL FCOMM(NCPU)                                                    0134
      TCWI=THCWU                                                          0135
      TCW2=THTF(NCPU)                                                     0136
      IF(NCPU.GT.NCWU) TCW1=THTI(NCPU)                                    0137
      CALL FINT(PALP1,ALPHDS,TCW1,TCW2,5.0,2)                             0138
      ALPI=ALPI+PALP1                                                     0139
 530  CONTINUE                                                            0140
      ALP2=(0.,0.)                                                        0141
      DO 532 NCPL=NCWL,N,I                                                0142
      CALL FCOMM(NCPL)                                                    0143
      TCWI=THCWL                                                          0144
      TCW2=THTF(NCPL)                                                     0145
      IF(NCPL.GT.NCWL) TCWI=THTI(NCPL)                                    0146
      CALL FINT(PALP2,ALPHDS,TCW1,TCW2,5.0,2)                             0147
      ALP2=ALP2+PALP2                                                     0148
 532  CONTINUE                                                            0149
      ATTEN=ALP1+ALP2                                                     0150
C     CALCULATE CREEPING WAVE                                             0151
      PHCWI=SPHASE(THT,0.,THCWU,0.,RSCWU,FK)                              0152
      PHCW2=SPHASE(THT,0.,THCWL,0.,RSCWL,FK)                              0153
      CALLDIFFGO(RSCWU,THCWU,RDUM,RDDM,ECAP1,FCAP1,GCAP1,ELCI,FLC1,GLCI)  0154
      CALLDIFFGO(RSCWL,THCWL,RDUM,RDDM,ECAP2,FCAP2,GCAP2,ELC2,FLC2,GLC2)  0155
      FKAP1=ELCI/ECAPI                                                    0156
      FKAP2=ELC2/ECAP2                                                    0157
      RHCWI=I./FKAPI                                                      0158
      RHCW2=1./FKAP2                                                      0159
      DSQC=DSQ(RHCW1,RHCW2,WAVE)                                          0160
      ECW=-2.*DSQC*PHCWI*PHCW2*CEXP(-ATTEN+(0.,-I.)*FKLCW)                0161
      ETOT=EGO+ECW                                                        0162
      EMAG=CABS(ETOT)                                                     0163
      SIGMA=2.*TP*EMAG*EMAG                                               0164
      SIGMAL=10.*ALOG10(SIGMA)                                            0165
      IF(THTD.GT.90.) CALL FNVRT(N)                                       0166
      WRITE(6,5)THTD,EGO,ECW,ETOT                                         0167
      WRITE(6,4)THTD,SIGMA,SIGMAL                                         0168
 100  CONTINUE                                                            0169
      STOP                                                                0170
      END                                                                 0171
$IBFTC DELSP.  LIST                                                       0172
      COMPLEX FUNCTION DELSP(THT)                                         0173
      R=RAD(THT)                                                          0174
      CALL DIFFGO(R,THT,RDUM,RDDM,ECAP,FCAP,GCAP,ELC,FLC,GLC)             0175
      ECAP=ABS(ECAP)                                                      0176
      DELSP=CMPLX(SQRT(ECAP),0.)                                          0177
      RETURN                                                              0178
      END                                                                 0179
$IBFTC FNVRT.  LIST                                                       0180
      SUBROUTINE FNVRT(N)                                                 0181
      COMMONRA1,RA3,RBI,RA9,RA10,RA11,WAVE/DATA/AR1(20),AR3(20),BR1(20),  0182
     CAR9(20),ARI0(20),AR11(20)                                           0183
       COMMON/ANGL/THTI(20),THTF(20),THTP(20)                             0184
      PI=3.1415927                                                        0185
       DO I0 I=1,N,I                                                      0186
      BRI(I)=-BR1(I)                                                      0187
      AR9(I)=-AR9(I)                                                      0188
      THTP(I)=PI-THTF(I)                                                  0189
      THTF(I)=PI-THTI(I)                                                  0190
      THTI(I)=THTP(I)                                                     0191
```

```
 10   CONTINUE                                                          0192
      RETURN                                                            0193
      END                                                               0194
$IBFTC SPHAS    LIST                                                    0195
      COMPLEX FUNCTION SPHASE(THTI,PHII,THTB,PHIB,RB,FK)                0196
      FS=1.                                                             0197
      TEST=ABS(PHII-PHIB)                                               0198
      IF(TEST.EQ.0.) FS=-1.                                             0199
      FL=RB*COS(THTI+FS*THTB)                                           0200
      FKL=FK*FL                                                         0201
      SPHASE=CMPLX(COS(FKL),SIN(FKL))                                   0202
       RETURN                                                           0203
      END                                                               0204
$IBFTC PHAS.    LIST                                                    0205
       COMPLEX FUNCTION PHASE(THTI,PHII,THTB,PHIB,RB,FK)                0206
      FS=1.                                                             0207
      TEST=ABS(PHII-PHIB)                                               0208
      IF(TEST.EQ.0.) FS=-1.                                             0209
      FL=2.*RB*COS(THT1+FS*THTB)                                        0210
      FKL=FK*FL                                                         0211
      PHASE=CMPLX(COS(FKL),SIN(FKL))                                    0212
       RETURN                                                           0213
      END                                                               0214
$IBFTC ALPH.    LIST                                                    0215
      COMPLEX FUNCTION ALPH(RHO1,RHO2,WAVE)                             0216
      COMPLEX ALPH,EX                                                   0217
      EX=CMPLX(0.86603,0.5)                                             0218
      IF(RHO2.EQ.0.) GO TO 10                                           0219
      RA=RHO1/RHO2                                                      0220
      U2=(1.48/EXP(0.84*RA))+0.20                                       0221
      GO TO 11                                                          0222
 10   U2=0.20                                                           0223
 11   AR1=2.*ALOG(RHO1)/3.                                              0224
      FR1=1./EXP(AR1)                                                   0225
      WV1=ALOG(WAVE)/3.                                                 0226
      FWV=1./EXP(WV1)                                                   0227
      ALPH=U2*FR1*FWV*EX                                                0228
      RETURN                                                            0229
      END                                                               0230
$IBFTC DSQ.     LIST                                                    0231
      COMPLEX FUNCTION DSQ(RI1,RS1,WAVE)                                0232
      COMPLEX DSQ,EX                                                    0233
      EX=CMPLX(.96593,-.25882)                                          0234
      U1=0.270                                                          0235
      A=SQRT(RI1*RS1)                                                   0236
      AL=ALOG(A)/3.                                                     0237
      WV=2.*ALOG(WAVE)/3.                                               0238
      FA=EXP(AL)                                                        0239
      FW=EXP(WV)                                                        0240
      DSQ=U1*FA*FW*EX                                                   0241
      RETURN                                                            0242
      END                                                               0243
$IBFTC ALPDS.   LIST                                                    0244
      COMPLEX FUNCTION ALPHDS(THT)                                      0245
      COMMON AR1,AR3,BR1,AR9,AR10,AR11,WAVE                             0246
      COMPLEX ALPHDS,ALP,ALPH                                           0247
      PI=3.1415927                                                      0248
      R=RAD(THT)                                                        0249
      CALL DIFFGO(R,THT,RTHT,RDDM,ECAP,FCAP,GCAP,ELC,FLC,GLC)           0250
      FMET=SQRT(ECAP)                                                   0251
      RHO1=ECAP/ELC                                                     0252
      RHO2=GCAP/GLC                                                     0253
      IF(THT.EQ.0..OR.THT.EQ.PI) RHO2=RHO1                              0254
      RHO1=ABS(RHO1)                                                    0255
```

```
      RHO2=ABS(RHO2)                                                    0256
      ALP=ALPH(RHO1,RHO2,WAVE)                                          0257
      ALPHDS=ALP*FMET                                                   0258
      RETURN                                                            0259
      END                                                               0260
$IBFTC DIFGO   LIST                                                     0261
      SUBROUTINE DIFFGO(R,THT,RTHT,RTTH,ECAP,FCAP,GCAP,ELC,FLC,GLC)     0262
       COMMON A1,A3,B1,A9,A10,A11,WAVE                                  0263
C     DIFFERENTIAL GEOMETRY PROPERTIES OF QUADRIC SURFACE OF REVOLUTION 0264
C     F(R,THT)=A1*R*R*SIN(THT)*SIN(THT)+A3*R*R*COS(THT)*COS(THT)+B1*R*R* 0265
C     SIN(THT)*COS(THT)+A9*R*COS(THT)+A10*R*SIN(THT)+A11=0              0266
C     THETA-PHI COORDINATES                                             0267
C     UPPER AND LOWER CASE F = 0                                        0268
C     INPUT A1,A3,B1,A9,A10,A11,R,THT                                   0269
      PHI=0.                                                            0270
      ST=SIN(THT)                                                       0271
      CT=COS(THT)                                                       0272
      S2T=SIN(2.*THT)                                                   0273
      C2T=COS(2.*THT)                                                   0274
      ST2=ST*ST                                                         0275
      CT2=CT*CT                                                         0276
      SP = SIN(PHI)                                                     0277
      CP = COS(PHI)                                                     0278
      U=A1*ST2+A3*CT2+B1*ST*CT                                          0279
      V=A9*CT+A10*ST                                                    0280
      UTHT=A1*S2T-A3*S2T+B1*(CT2-ST2)                                   0281
      VTHT=-A9*ST+A10*CT                                                0282
      UTTH=2.*(A1-A3)*C2T-2.*B1*S2T                                     0283
      VTTH=-A9*CT-A10*ST                                                0284
      S1=2.*R*U+V                                                       0285
      S2=R*UTHT+VTHT                                                    0286
      FMAG=SQRT(S1*S1+S2*S2)                                            0287
      FMAG2=FMAG*FMAG                                                   0288
      FMAG3=FMAG2*FMAG                                                  0289
C     CALCULATE UPPER CASE E,F,G                                        0290
      RTHT=-R*S2/S1                                                     0291
      ECAP = R*R+RTHT*RTHT                                              0292
      FCAP = 0.                                                         0293
      GCAP = R*R*ST2                                                    0294
C     CALCULATE LOWER CASE E,F,G                                        0295
      RTTH=-(2.*RTHT*(2.*R*UTHT+VTHT)+2.*RTHT*RTHT*U+R*(R*UTTH+VTTH))/S1 0296
      ELC=(RTTH*S1+2.*RTHT*S2-R*S1)/FMAG                                0297
      FLC = 0.                                                          0298
      GLC=-R*(S1*ST2+S2*ST*CT)/FMAG                                     0299
      RETURN                                                            0300
      END                                                               0301
$IBFTC FSPDT.  LIST                                                     0302
      SUBROUTINE FSPDT(RSP,THSP,FNT,THTS,THTF,DTHT,PHI,VX,VY,VZ)        0303
C     INPUT THTS,THTF,DTHT,PHI,VX,VY,VZ                                 0304
      COMMON AR1,AR3,BR1,AR9,AR10,AR11,WAVE                             0305
C     SEARCH FOR LARGEST SCALAR PRODUCT IN SECTION                      0306
      FND=ABS((THTF-THTS)/DTHT)                                         0307
      ND=FND+1.                                                         0308
      FNT=0.                                                            0309
      RSP=0.                                                            0310
      THSP=0.                                                           0311
       DO 100 I=1,ND,1                                                  0312
      FI=I-1                                                            0313
      THT=FI*DTHT+THTS                                                  0314
      R=RAD(THT)                                                        0315
      CALL FNORM(FNX,FNY,FNZ,R,THT,PHI)                                 0316
      FNE=VX*FNX+VY*FNY+VZ*FNZ                                          0317
      FNE=ABS(FNE)                                                      0318
      IF(FNT.GT.FNE) GO TO 11                                           0319
```

```
      FNT=ABS(FNE)                                                      0320
      RSP=R                                                             0321
      THSP=THT                                                          0322
 11   CONTINUE                                                          0323
 100  CONTINUE                                                          0324
      RETURN                                                            0325
      END                                                               0326
$IBFTC FINT.   LIST                                                     0327
      SUBROUTINE FINT(SSS,FCTI,FLL,FUL,ERRR,NX)                         0328
C     INPUT FCTI,FLL,FUL,ERRR,NX                                        0329
C     EXTERNAL DECLARATION FOR FUNCTION FCTI REQUIRED                   0330
      COMPLEX SSS,SS,FSS,FCTI,TRAP,TRAZ,SIMP,SIMZ,FNCP,FNCZ             0331
      FN=NX                                                             0332
      DEL=(FUL-FLL)/FN                                                  0333
      SSS=(0.,0.)                                                       0334
      ERR=0.01*ERRR/FN                                                  0335
      A=FLL                                                             0336
      DO 40 NNX=1,NX,1                                                  0337
      MXX=0                                                             0338
      B=A+DEL                                                           0339
      SS=(0.,0.)                                                        0340
      MX=2                                                              0341
      DX=DEL/2.                                                         0342
      LX=1                                                              0343
      X=A                                                               0344
      GO TO 15                                                          0345
 5    TRAZ=DX*SS                                                        0346
      MX=1                                                              0347
      LX=1                                                              0348
      DX=DEL                                                            0349
 10   SS=0.                                                             0350
      LX=LX+1                                                           0351
      DX=0.5*DX                                                         0352
      X=A+DX                                                            0353
 15   DO 20 IX=1,MX,1                                                   0354
      FSS=FCTI(X)                                                       0355
      SS=SS+FSS                                                         0356
 20   X=X+2.*DX                                                         0357
      IF(LX.EQ.1) GO TO 5                                               0358
      MX=2*MX                                                           0359
      TRAP=0.5*TRAZ+DX*SS                                               0360
      DIF=CABS(TRAP-TRAZ)                                               0361
      IF(DIF.GE.DIP) MXX=MXX+1                                          0362
      DIP=DIF                                                           0363
      SIMP=(4.*TRAP-TRAZ)/3.                                            0364
      FNCP=(16.*SIMP-SIMZ)/15.                                          0365
      ER=CABS(1.-FNCZ/FNCP)                                             0366
      TRAZ=TRAP                                                         0367
      SIMZ=SIMP                                                         0368
      FNCZ=FNCP                                                         0369
      IF(LX.LT.4) GO TO 10                                              0370
      IF(MXX.GT.4) GO TO 30                                             0371
      IF(ER.GT.ERR) GO TO 10                                            0372
 30   SSS=SSS+FNCP                                                      0373
 40   A=A+DEL                                                           0374
 50   CONTINUE                                                          0375
      RETURN                                                            0376
      END                                                               0377
$IBFTC FFNRM.  LIST                                                     0378
      SUBROUTINE FNORM(FNVX,FNVY,FNVZ,R,THT,PHI)                        0379
C     INPUT R,THT,PHI                                                   0380
      COMMON AR1,AR3,BR1,AR9,AR10,AR11,WAVE                             0381
      ST=SIN(THT)                                                       0382
      CT=COS(THT)                                                       0383
```

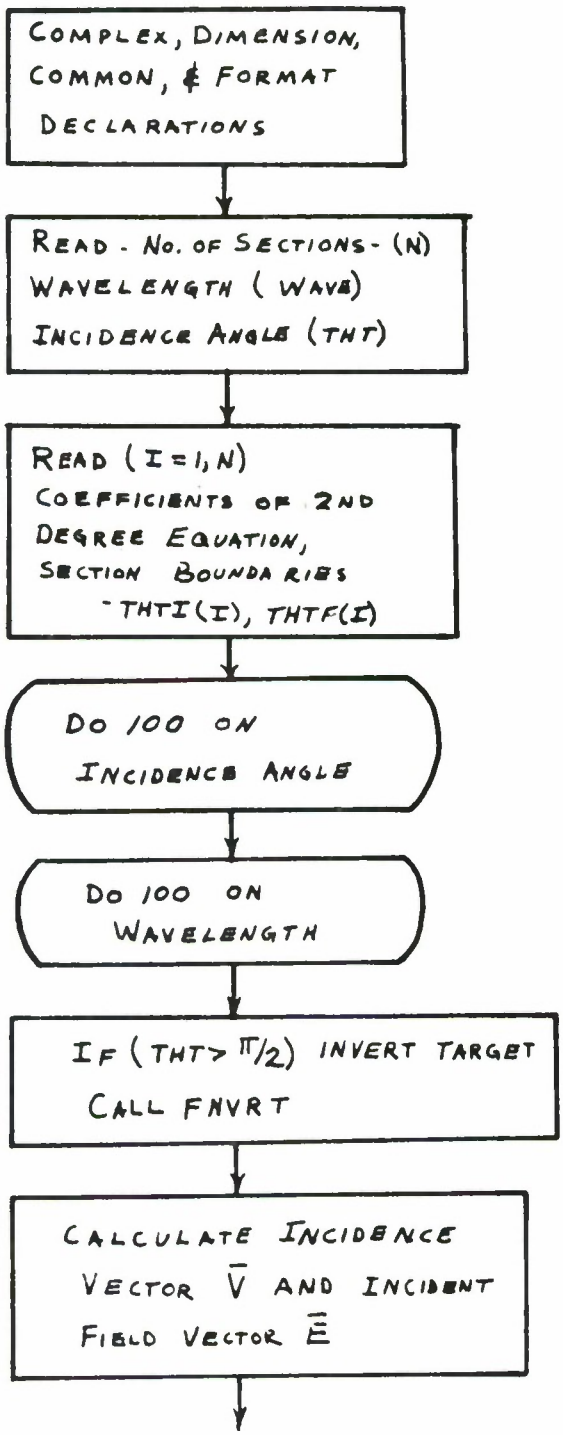
```
      SP=SIN(PHI)                                                        0384
      CP=COS(PHI)                                                        0385
      U=AR1*ST*ST+AR3*CT*CT+BR1*ST*CT                                    0386
      V=AR9*CT+AR10*ST                                                   0387
      UTH=2.*(AR1-AR3)*ST*CT+BR1*(CT*CT-ST*ST)                           0388
       VTH=-AR9*ST+AR10*CT                                               0389
      F1=2.*R*U+V                                                        0390
      F2=R*UTH+VTH                                                       0391
      FX=ST*CP*F1+CT*CP*F2                                               0392
      FY=ST*SP*F1+CT*SP*F2                                               0393
      FZ=CT*F1-ST*F2                                                     0394
      FN=SQRT(FX*FX+FY*FY+FZ*FZ)                                         0395
      FNVX=FX/FN                                                         0396
      FNVY=FY/FN                                                         0397
      FNVZ=FZ/FN                                                         0398
      RETURN                                                             0399
      END                                                                0400
$IBFTC RADD    LIST                                                      0401
      FUNCTION RAD(THT)                                                  0402
C     CALCULATE R                                                        0403
      COMMON AR1,AR3,BR1,AR9,AR10,AR11,WAVE                              0404
      ST=SIN(THT)                                                        0405
      CT=COS(THT)                                                        0406
      C1=AR1*ST*ST+AR3*CT*CT+BR1*ST*CT                                   0407
      C2=AR9*CT+AR10*ST                                                  0408
      C3=AR11                                                            0409
      IF(C1.EQ.0.)GO TO 11                                               0410
      ARG=SQRT(C2*C2-4.*C1*C3)                                           0411
      R=(-C2+ARG)/(2.*C1)                                                0412
      R2=(-C2-ARG)/(2.*C1)                                               0413
      IF(R2.LT.R.AND.R2.GT.0.) R=R2                                      0414
      GO TO 12                                                           0415
 11   R=-C3/C2                                                           0416
 12   RAD=R                                                              0417
      RETURN                                                             0418
      END                                                                0419
$IBFTC FCOM.   LIST                                                      0420
      SUBROUTINE FCOMM(I)                                                0421
      COMMONRA1,RA3,RB1,RA9,RA10,RA11,WAVE/DATA/AR1(20),AR3(20),BR1(20),  0422
     CAR9(20),AR10(20),AR11(20)                                          0423
      RA1=AR1(I)                                                         0424
      RA3=AR3(I)                                                         0425
      RB1=BR1(I)                                                         0426
      RA9=AR9(I)                                                         0427
      RA10=AR10(I)                                                       0428
      RA11=AR11(I)                                                       0429
      RETURN                                                             0430
      END                                                                0431
$DATA                                                                    0432
```

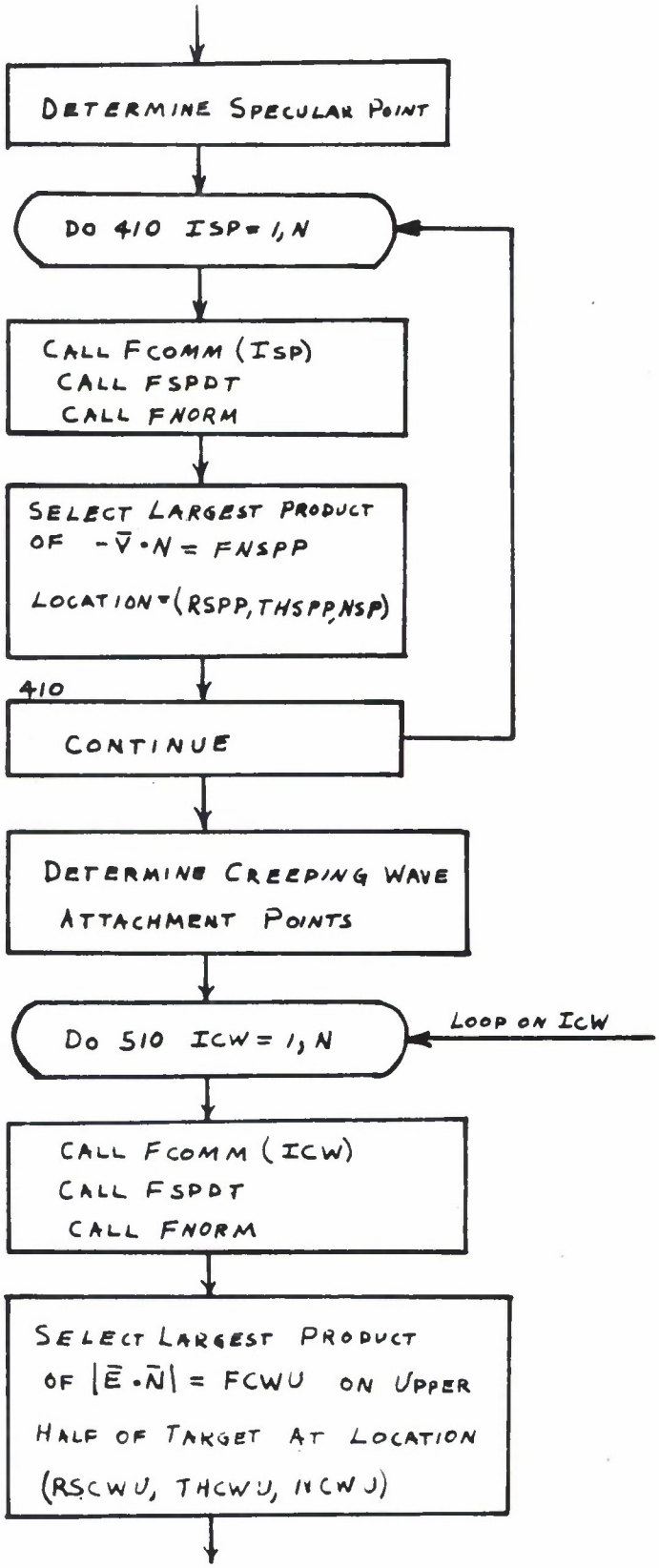
APPENDIX II
THE COMPUTER FLOW DIAGRAM

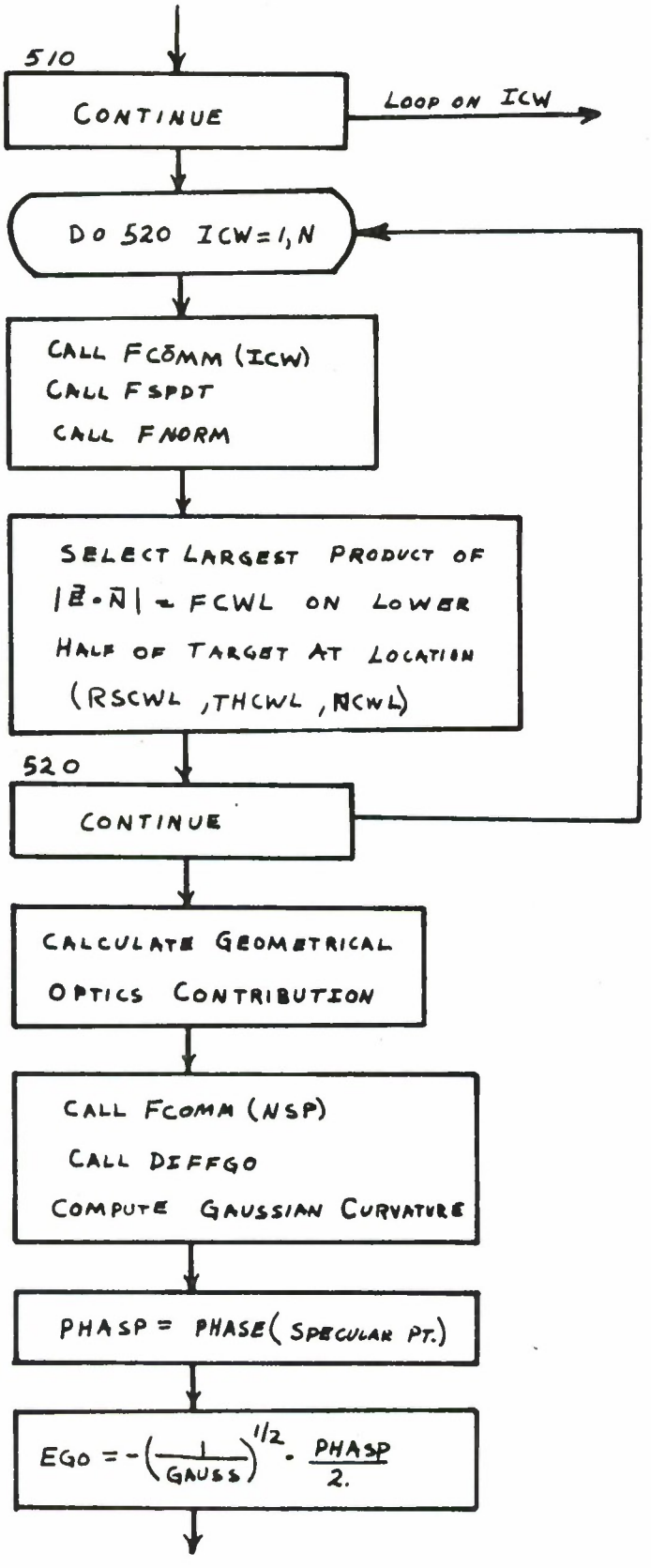
The flow diagram presented in Ref. 1 is presented here. This diagram shows the sequence in which the operations are performed.

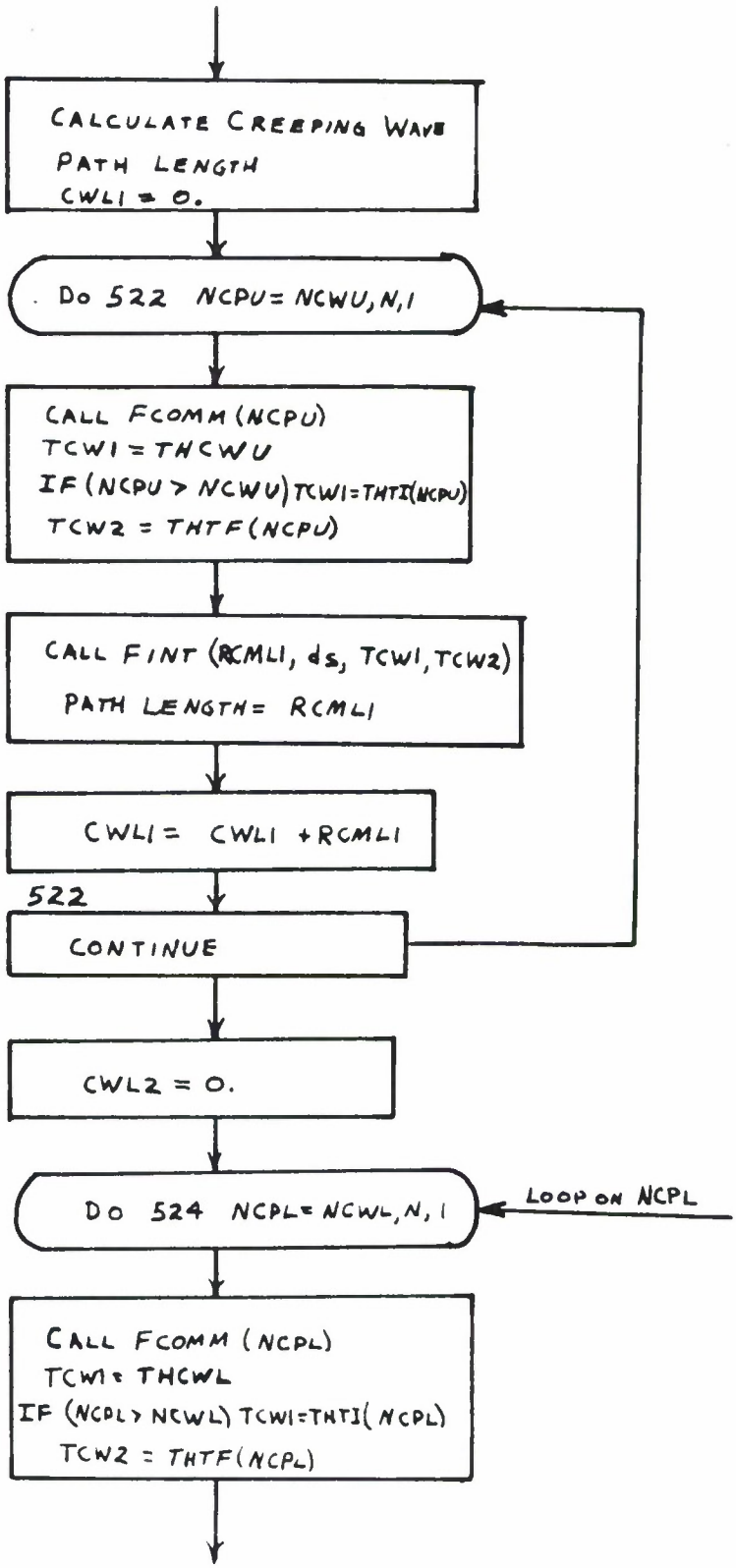
CREEPING WAVE COMPUTER PROGRAM

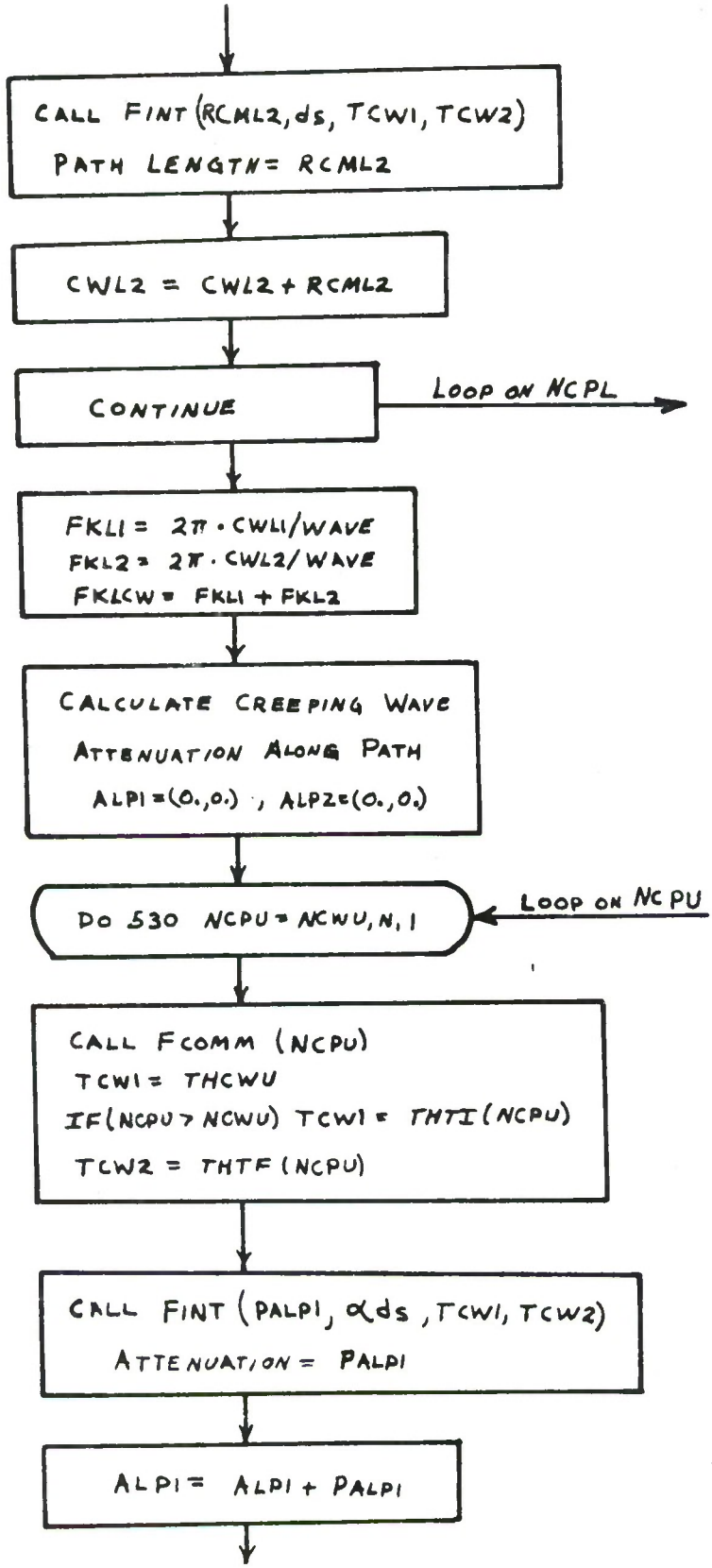
HORIZONTAL POLARIZATION

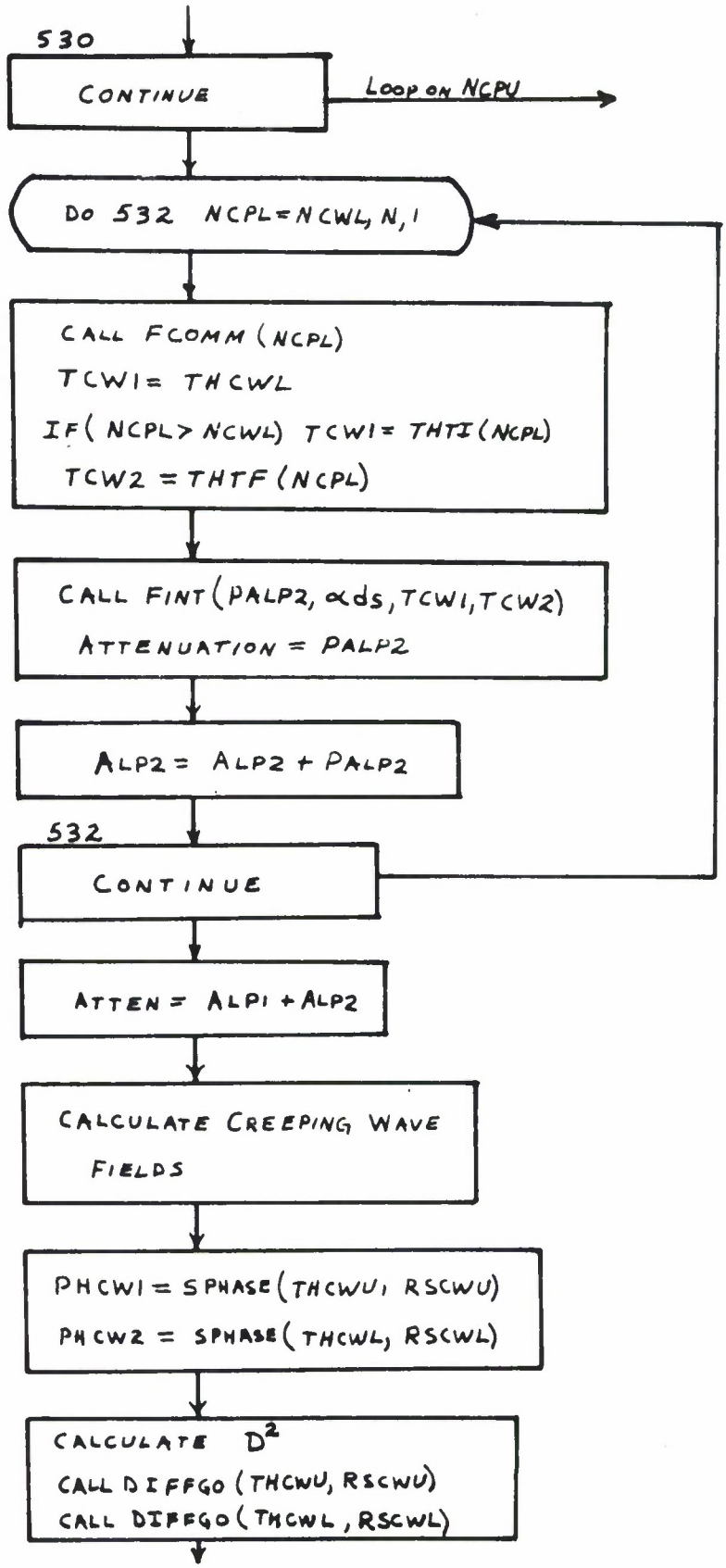


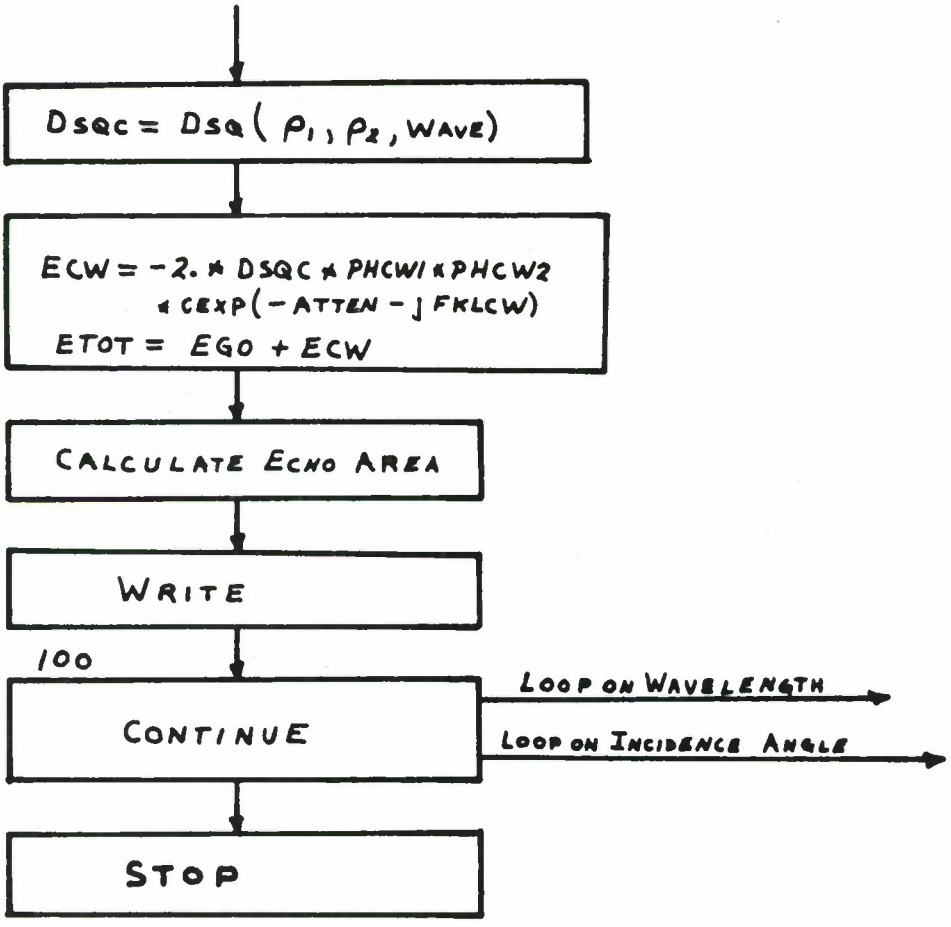

DETERMINE SPECULAR POINT
DO 410 ISP = 1, N
CALL FCOMM (ISP)
CALL FSPDT
CALL FNORM
SELECT LARGEST PRODUCT OF $-\bar{V} \cdot N = FNSPP$
LOCATION = (RSPP, THSPP, NSP)
410
CONTINUE
DETERMINE CREEPING WAVE ATTACHMENT POINTS
DO 510 ICW = 1, N
LOOP ON ICW
CALL FCOMM (ICW)
CALL FSPDT
CALL FNORM
SELECT LARGEST PRODUCT OF $|\bar{E} \cdot \bar{N}| = FCWU$ ON UPPER HALF OF TARGET AT LOCATION (RSCWU, THCWU, NCWJ)


510
CONTINUE
LOOP ON ICW
DO 520 ICW=1,N
CALL FCOMM (ICW)
CALL FSPDT
CALL FNORM
SELECT LARGEST PRODUCT OF
|E·N| ~ FCWL ON LOWER
HALF OF TARGET AT LOCATION
(RSCWL, THCWL, NCWL)
520
CONTINUE
CALCULATE GEOMETRICAL
OPTICS CONTRIBUTION
CALL FCOMM (NSP)
CALL DIFFGO
COMPUTE GAUSSIAN CURVATURE
PHASP = PHASE(SPECULAR PT.)
EGO = -(1/GAUSS)^1/2 · PHASP/2.


CALCULATE CREEPING WAVE
PATH LENGTH
CWL1 = 0.
DO 522 NCPU = NCWU, N, 1
CALL FCOMM (NCPU)
TCW1 = THCWU
IF (NCPU > NCWU) TCW1 = THTI(NCPU)
TCW2 = THTF(NCPU)
CALL FINT (RCML1, ds, TCW1, TCW2)
PATH LENGTH = RCML1
CWL1 = CWL1 + RCML1
522
CONTINUE
CWL2 = 0.
DO 524 NCPL = NCWL, N, 1
LOOP ON NCPL
CALL FCOMM (NCPL)
TCW1 = THCWL
IF (NCPL > NCWL) TCW1 = THTI(NCPL)
TCW2 = THTF(NCPL)


CALL FINT (RCML2, ds, TCW1, TCW2)
PATH LENGTH = RCML2
CWL2 = CWL2 + RCML2
CONTINUE
Loop on NCPL
FKL1 = 2π · CWL1/WAVE
FKL2 = 2π · CWL2/WAVE
FKLCW = FKL1 + FKL2
CALCULATE CREEPING WAVE
ATTENUATION ALONG PATH
ALP1 = (0.,0.) , ALP2 = (0.,0.)
DO 530 NCPU = NCWU, N, 1
Loop on NCPU
CALL FCOMM (NCPU)
TCW1 = THCWU
IF (NCPU > NCWU) TCW1 = THTI (NCPU)
TCW2 = THTF (NCPU)
CALL FINT (PALP1, αds, TCW1, TCW2)
ATTENUATION = PALP1
ALP1 = ALP1 + PALP1


530
CONTINUE
LOOP ON NCPU
DO 532 NCPL=NCWL, N, 1
CALL FCOMM (NCPL)
TCW1= THCWL
IF(NCPL> NCWL) TCW1= THTI(NCPL)
TCW2 =THTF (NCPL)
CALL FINT(PALP2, αds, TCW1, TCW2)
ATTENUATION = PALP2
ALP2 = ALP2 + PALP2
532
CONTINUE
ATTEN = ALP1 + ALP2
CALCULATE CREEPING WAVE FIELDS
PHCW1 = SPHASE(THCWU, RSCWU)
PHCW2 = SPHASE(THCWL, RSCWL)
CALCULATE D²
CALL DIFFGO (THCWU, RSCWU)
CALL DIFFGO(THCWL, RSCWL)


DSQC = DSQ(ρ1 , ρ2 , WAVE)
ECW = -2. * DSQC * PHCW1 * PHCW2
* CEXP(- ATTEN - J FKLCW)
ETOT = EGO + ECW
CALCULATE ECHO AREA
WRITE
100
CONTINUE
STOP
LOOP ON WAVELENGTH
LOOP ON INCIDENCE ANGLE

REFERENCES

1. Ryan, C.E., Jr., "A Geometrical Theory of Diffraction Analysis of the Radar Cross Section of a Sectionally Continuous Second Degree Surface of Revolution," Report 2430-4, April 1968, ElectroScience Laboratory, The Ohio State University Research Foundation; prepared under Contract F-19628-67-C-0318 for Department of the Air Force, Electronic Systems Division, L.G. Hanscom Field, Bedford, Massachusetts.

2. Ryan, C.E., Jr., "Memorandum on Analysis of Echo Area of Targets Using Geometrical Theory of Diffraction and Creeping Wave Theory," Report 2430-1, 22 May 1967, ElectroScience Laboratory, The Ohio State University Research Foundation; prepared under Contract F-19628-67-C-0318 for Department of the Air Force, Electronic Systems Division, L.C. Hanscom Field, Bedford, Massachusetts. (AD 658 469).

3. Richmond, J.H., "Scattering by Wire Loops and Square Plates in the Resonance Region," Report 2097-1, 28 January 1966, Antenna Laboratory, The Ohio State University Research Foundation; prepared under Contract No. AF 19(628)-4883 for Air Force Systems Command, United States Air Force, Bedford, Massachusetts.

4. Ryan, C.E., Jr., "A Computer Program for Backscatter by Targets Composed of Cones, Cylinders, and Disks," (in process), ElectroScience Laboratory, The Ohio State University Research Foundation; prepared under Contract F-19628-67-C-0318 for Department of the Air Force, ElectronicsSystems Division, L.C. Hanscom Field, Bedford, Massachusetts.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*: ElectroScience Laboratory, The Ohio State University, Department of Electrical Engineering, 1320 Kinnear Rd., Columbus, Ohio

2a. REPORT SECURITY CLASSIFICATION: Unclassified

2b. GROUP: N/A

3. REPORT TITLE: A Computer Program for Backscatter by Smoothly Joined, Second Degree Surfaces of Revolution

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*: Technical Report

5. AUTHOR(S) *(Last name, first name, initial)*: Ryan, C.E., Jr.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1968	25	4

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
F-19628-67-C-0308	ESD-TR-68-213
b. PROJECT NO.	
c. TASK	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. AVAILABILITY/LIMITATION NOTICES: This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY: Deputy for Surveillance and Control Systems, Electronic Systems Division, L. G. Hanscom Field, Bedford, Mass.

13. ABSTRACT

A computer program for calculation of the echo area of smoothly joined, N section convex conducting surfaces, described by a second degree equation is described. For the case of Eθ (parallel) polarization of the incident and scattered fields the solution is obtained by a combination of geometrical optics and creeping wave theory. For the case of Eϕ (perpendicular) polarization the solution is obtained using geometrical optics, and the creeping wave is neglected. The computed results for Eθ polarization are in good agreement with measurements for prolate spheroids, prolate spheroid-sphere, and prolate spheroid-oblate spheroid combinations.

DD FORM 1 JAN 64 1473

UNCLASSIFIED

Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Radar Cross Section Backscatter Surface of Revolution Computer Program Geometrical Optics Creeping Waves						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through

_____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through

_____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through

_____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

UNCLASSIFIED

Security Classification