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

C. E. Ryan, Jr.

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

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.)
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 $E_\theta$ (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 $E_\phi$ (perpendicular) polarization the solution is obtained using geometrical optics, and the creeping wave is neglected. The computed results for $E_\theta$ polarization are in good agreement with measurements for prolate spheroids, prolate spheroid-sphere, and prolate spheroid-oblative spheroid combinations.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. TARGET DESCRIPTION</td>
<td>1</td>
</tr>
<tr>
<td>III. THE CREEPING WAVE</td>
<td>2</td>
</tr>
<tr>
<td>COMPUTER PROGRAM</td>
<td></td>
</tr>
<tr>
<td>IV. FUNCTIONS AND SUBROUTINES</td>
<td>6</td>
</tr>
<tr>
<td>V. DATA</td>
<td>8</td>
</tr>
<tr>
<td>VI. CONCLUSION</td>
<td>8</td>
</tr>
<tr>
<td>APPENDIX I - THE COMPUTER PROGRAM LISTING</td>
<td>9</td>
</tr>
<tr>
<td>APPENDIX II - THE COMPUTER FLOW DIAGRAM</td>
<td>17</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>25</td>
</tr>
</tbody>
</table>
A COMPUTER PROGRAM FOR BACKSCATTER
BY SMOOTHLY JOINED, SECOND DEGREE
SURFACES OF REVOLUTION

I. INTRODUCTION

The computer program listed in Appendix 1 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 \( E_\theta \) (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 \( E_\phi \) (perpendicular) polarization is also obtained. The creeping wave scattered field has not been included for \( E_\phi \) polarization due to the difficulty in obtaining the ray path of the creeping wave in this case. This program has been tested for \( E_\theta \) 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

\[
F(r, \theta) = A_1 r^2 \sin^2 \theta + A_2 r^2 \cos^2 \theta + A_3 r^2 \sin \theta \cos \theta + A_4 r \cos \theta + A_5 r \sin \theta + A_6 = 0
\]

The constants \( A_1 \ldots A_6 \) specify the surface in the section (i) bounded by the angles \( \theta_i \) and \( \theta_{i+1} \). 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 \( \mathbf{r} \) directed. That is, \( \left. \frac{d\mathbf{r}}{d\theta} \right|_z = 0 = 0 \). Discontinuities in the derivatives \( \left( \frac{d^n\mathbf{r}}{d\theta^n} \right) \) 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 \( \left( \frac{d\mathbf{r}}{d\theta} \right) \) at the junction ("wedge" discontinuities) causes diffraction, and can be evaluated using wedge diffraction techniques.\(^1\),\(^2\) A discontinuity in the second derivative \( \left( \frac{d^2\mathbf{r}}{d\theta^2} \right) \) 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 significant 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 re-radiation 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).
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 \( \theta \) 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 \( 180^\circ \) in \( \theta \).

**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, RSI 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.

<table>
<thead>
<tr>
<th>Card</th>
<th>0</th>
<th>1.0</th>
<th>2.0</th>
<th>3.1415927</th>
<th>2.4799</th>
<th>5.0955</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1.570796</td>
<td>2.4799</td>
<td>5.102</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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. 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. 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  IJOB  0000
$IBJOB   GO,MAP  0001
$IBFTC CREEP   LIST,NODECK  0002

1  FORMAT(115)  0003
2  FORMAT(2F10.5)  0004
3  FORMAT(8F10.5)  0005
4  FORMAT(3F15.8)  0006
5  FORMAT(7F15.8)  0007
6  FORMAT(7H THSSPP=F15.8, GH THCWU=F15.8, 7H THCWL=F15.8)  0008
7  FORMAT(5H NSP=15.6H NSW=F15.6H NCW=115)  0009
8  FORMAT(6H CWL1=F15.8, 6H CWL2=F15.8)  0010
COMMON RA1, RA3, RB1, RA9, RA10, RA11, WAVE/DATA/AR1(20), AR3(20), BR1(20),  0011
CAR9(20), AR10(20), AR11(20)
COMMON ANGL, TH120, THF20, THTP20
COMPLEX PALP1, PALP2
COMPLEX PHASP, EGO, ALPHDS, ATTN, ALP1, ALP2, PHCW1, PHCW2, DSGC, ECW, ETOT  0015
COMPLEX CPHL, PHASE, ALPH, DSG, DELSP, CMPL1, CMPL2, CEXP
COMMON SPHASE
DEGRAD=0.01745329
RADEG=57.29578
PI=3.1415927
PI2=PI/2.
TP=2.*PI
READ(5.1) N  0020
READ(5.2) WAVE, DTHT
WRITE(6.2) WAVE, DTHT
READ(5.3) (THT(1R), THTF(1R), AR1(1R), AR3(1R), BR1(1R), AR9(1R), AR10(1R),  0026
AR11(1R), IR=1, N)
DTHB=DEGRAD
FNT=180./DTHT
NT=FNT-1.
FK=TP/WAVE
DO 100 N=1, NT  0030
FNTH=NTH
THTD=FNTH*DTHT
THT=DEGRAD*THTD
IF(THTD.GT.90.) THT=PI-THT
CALL FNVRT(N)  0037
C CALCULATE INCIDENCE VECTOR

PHI=0.
VIX=-SIN(THT)
VIZ=-COS(THT)
VIY=0.

C CALCULATE INCIDENT E-VECTOR-PARALLEL POL.

EIX=COS(THT)
EIY=0.
EIZ=-SIN(THT)

C DETERMINE SPECULAR POINT

FNPP=0.
NSP=0.
RSPP=0.
THPP=0.
DO 410 ISP=1, N, 1
CALL FCMPM(15P)
CALL FSPT(RSPP, THPP, FNPP, THT1(1SP), THTF1(1SP), DTHB00, VIX, VIY, VIZ)
IF(FNPP.GT.FNPP.OR.THPP.GT.PI2)GOTO 411
FNPP=ABS(FNPP)
RSPP=RSPP
THPP=THPP
NSP=ISP
411 CONTINUE
410 CONTINUE
C DETERMINE CREEPING WAVE POINTS
FCWU=0.
NCWU=0
RSCWU=0
THCWU=0
DO 510 ICW=1,N+1
CALL FCOMM(ICW)
CALL FSPDT(RCWU,THWU,FWU,THTF(ICW),THTF(ICW),DTHB,0,EIX,EIY,EIZ)
IF(FCWU>GT,FWU,OR,THWU>LT,P12)GO TO 511
FCWU=ABS(FWU)
RSCWU=RCWU
THCWU=THWU
NCWU=ICW
511 CONTINUE
510 CONTINUE
FCWL=0
RSCWL=0
THCWL=0
DO 520 ICW=1,N+1
CALL FCOMM(ICW)
CALL FSPDT(RCWL,THWL,FWL,THTF(ICW),THTF(ICW),DTHB,ICW,EIX,EIY,EIZ)
IF(FCWL>FWL,OR,THWL>P12)GO TO 521
FCWL=FWL
RSCWL=RCWL
THCWL=THWL
NCWL=ICW
521 CONTINUE
520 CONTINUE
WRITE(6,7) NSP,NCWU,NCWL
WRITE(6,6)THSPP,THCWU,THCWU
CALCULATE GEOMETRICAL OPTICS TERM
CALL FCOMM(NSP)
CALL D1FFG0(RSPP,THSPP,RDUM,RRDM,ECAP,FCAP,GCAP,ELC,FLC,GLC)
IF(GCAP) 20,21.20
GAUSS=ELC/ECAP
GAUSS=GAUSS*GAUSS
GO TO 22
GAUSS=(ELC*GLC-FLC*FLC)/(ECAP*GCAP-FCAP*FCAP)
20 CONTINUE
PHASP=PHASE(THT,0,THSPP,0,RSPP,FK)
EGO=SQR(1/GAUSS)*PHASP/2
CALCULATE CREEPING WAVE PATH LENGTH
EXTERNAL DELSP
CWL1=0
DO 522 NCPU=NCWU,N+1
CALL FCOMM(NCPU)
TCWI=THCWU
TCW2=THTF(NCPU)
IF(NCPU>GT,NCWU) TCWI=THTI(NCPU)
CALL FINT(CML1,DELSP,TCW1,TCW2,5,0,2)
RCLM1=REAL(CML1)
CWL1=CWL1+RCLM1
522 CONTINUE
CWL2=0
DO 524 NCPL=NCWL,N+1
CALL FCOMM(NCPL)
TCW1=THCWL
TCW2=THTF(NCPU)
IF(NCPU>GT,NCWL) TCW1=THTI(NCPU)
CALL FINT(CML2,DELSP,TCW1,TCW2,5,0,2)
RCLM2=REAL(CML2)
CWL2=CWL2+RCLM2
524 CONTINUE
WRITE(6,8) CWL1,CWL2
FKL1=FK*CWL1
FKL2 = FK * CWL2
FKL2 = FKLI + FKL2

C CALCULATE CREEPING WAVE ATTENUATION
EXTERNAL ALPHDS
ALPI = (0., 0.)
DO 530 NCPU = NCWU * N + 1
CALL FCOMM (NCPU)
TCW1 = THCWU
TCW2 = THTF (NCPU)
IF (NCPU > GT * NCWU) TCW1 = THTI (NCPU)
CALL FINT (PALP1, ALPHDS, TCW1, TCW2, 5.0, 2)
ALP1 = ALP1 + PALP1
530 CONTINUE
ALP2 = (0., 0.)
DO 532 NCPL = NCWL * N + 1
CALL FCOMM (NCPL)
TCW1 = THCWL
TCW2 = THTF (NCPL)
IF (NCPL > GT * NCWL) TCW1 = THTI (NCPL)
CALL FINT (PALP2, ALPHDS, TCW1, TCW2, 5.0, 2)
ALP2 = ALP2 + PALP2
532 CONTINUE
ATTEN = ALP1 + ALP2

C CALCULATE CREEPING WAVE
PHCW1 = SPhASE (THCWU, 0., RSCWU, FK)
PHCW2 = SPhASE (THCWL, 0., RSCWL, FK)
CALL DIFFGO (RSCWU, THCWU, RDUM, RDDM, ECAP1, FCAP1, GCAP1, ELC1, FLC1, GLC1)
CALL DIFFGO (RSCWL, THCWL, RDUM, RDDM, ECAP2, FCAP2, GCAP2, ELC2, FLC2, GLC2)
FKAP1 = ELC1 / ECAP1
FKAP2 = ELC2 / ECAP2
RHCW1 = 1. / FKAP1
RHCW2 = 1. / FKAP2
DSQC = DSQ (RHCW1, RHCW2, WAVE)
ECW = -2. * DSQC * PHCW1 * PHCW2 * CEXP (-ATTEN + (0., -1.) * FKLCW)
ETOT = EGO + ECW
EMAG = CABS (ETOT)
SIGMA = 2. * TP * EMAG * EMAG
SIGMAL = 10. * ALOG10 (SIGMA)
IF (THTD > 90.) CALL FNVRT (N)
WRITE (6, 5) THTD, EGO, ECW, ETOT
WRITE (6, 4) THTD, SIGMA, SIGMAL
100 CONTINUE
STOP
END

$IBFTC DELSP. LIST
COMPLEX FUNCTION DELSP (THT)
R = RAD (THT)
CALL DIFFGO (R, THT, RDUM, RDDM, ECAP, FCAP, GCAP, ELC, FLC, GLC)
ECAP = ABS (ECAP)
DELSP = CMPLX (SORT (ECAP), 0.)
RETURN
END

$IBFTC FNVRT. LIST
SUBROUTINE FNVRT (N)
COMMON RA1, RA3, RB1, RA9, RA10, RA11, WAVE / DATA / AR1 (20), AR3 (20), BR1 (20),
CR9 (20), AR10 (20), AR11 (20)
COMMON / ANGL / THTI (20), THTF (20), THTP (20)
PI = 3.1415927
DO 10 I = 1 * N + 1
BRI (I) = -BR1 (I)
AR9 (I) = -AR9 (I)
THTF (I) = PI - THTF (I)
THTP (I) = PI - THTP (I)
THTI (I) = THTI (I)
10 CONTINUE
CONTINUE
RETURN
END

SIBFTC SPHAS LIST
COMPLEX FUNCTION SPHASE(THT1,PHI1,THTB,PHIB,RB,FK)
FL=RB*COS(THTI+FS*THTB)
FKL=FK*FL
SPHASE=CMPLX(COS(FKL),SIN(FKL))
RETURN
END

SIBFTC PHASE LIST
COMPLEX FUNCTION PHASE(THTI,PHI1,THTB,PHIB,RB,FK)
FL=2.*RB*COS(THTI+FS*THTB)
FKL=FK*FL
PHASE=CMPLX(COS(FKL),SIN(FKL))
RETURN
END

SIBFTC ALPH LIST
COMPLEX FUNCTION ALPH(RH01,RH02,WAVE)
EX=CMPLX(0.86603,0.5)
IF(RH02.EQ.0.) GO TO 10
RA=RH01/RH02
U2=(1.48/EXP(0.84*RA))+0.20
10 U2=0.*20
11 AR1=2.*ALOG(RH01)/3.
FR1=1.*EXP(AR1)
WV1=ALOG(WAVE)/3.
FWV=1.*EXP(WV1)
ALPH=U2*FR1*FWV*EX
RETURN
END

SIBFTC DSQ LIST
COMPLEX FUNCTION DSQ(RI1,RS1,WAVE)
EX=CMPLX(0.9659325882)
U1=0.*270
A=SQRTR(RI1*RS1)
AL=ALOG(A)/3.
WV=2.*ALOG(WAVE)/3.
FA=EXP(AL)
FW=EXP(WV)
DSQ=U1*FA*FW*EX
RETURN
END

SIBFTC ALPDS LIST
COMPLEX FUNCTION ALPDS(THT)
COMMON AR1,AR3,BR1,AR9,AR10,AR11,WAVE
COMPLEX ALPHDS,ALP,ALPH
PI=3.1415927
R=RAD(THT)
CALL DIFFGO(R,THT,RTHT,RDDM,ECAP,FCAP,GCAP,ELC,FLC,GLC)
FMET=SQRT(ECAP)
RH01=ECAP/ELC
RH02=GCAP/GLC
IF(THT.EQ.0.*OR.THT.EQ.PI) RH02=RH01
RH01=ABS(RH01)
RH02=ABS(RHO2)
ALP=ALPH(RHO1,RH02,WAVE)
ALPHDS=ALP*FMET
RETURN
END

$IBTC DIFGO LIST
SUBROUTINE DIFGO(R,THT,RTHT,RTTH,ECAP,FCAP,GCAP,ELC,FLC,GLC)
COMMON A1,A3,B1,A9,A10,A11,WAVE
C DIFFERENTIAL GEOMETRY PROPERTIES OF QUADRIC SURFACE OF REVOLUTION
C SIN(THT)*COS(THT)+A9*R*COS(THT)+A10*R*Sin(THT)+A11=0
C THETA-PHI COORDINATES
C UPPER AND LOWER CASE F = 0
C INPUT A1,A3,B1,A9,A10,A11,R,THT
PHI=0.
ST=SIN(THT)
CT=COS(THT)
S2T=SIN(2.*THT)
C2T=COS(2.*THT)
ST2=ST*ST
CT2=CT*CT
SP=SIN(PHI)
CP=COS(PHI)
U=A1*ST2+A3*CT2+B1*ST*CT
V=A9*CT+A10*ST
UTHT=A1*ST2-A3*S2T+B1*(CT2-ST2)
VTHT=-A9*ST+A10*CT
S1=2.*R*U+V
S2=R*UTHT+VTHT
FMAG=SQRT(S1*S1+S2*S2)
FMAG2=FMAG*FMAG
FMAG3=FMAG2*FMAG
C CALCULATE UPPER CASE E,F,G
RTHT=-R*S2/S1
ECAP = R*R+RTHT*RTHT
FCAP = 0.
GCAP = R*R*ST2
C CALCULATE LOWER CASE E,F,G
RTTH=-(2.*RTHT*(2.*R*UTHT+VTHT)+2.*RTHT*RTHT*U+R*(R*UTTH+VTTH))/S1
ELC=(RTTH*S1+2.*RTHT*S2-R*S1)/FMAG
FLC = 0.
GLC=-R*(S1*ST2+S2*ST*CT)/FMAG
RETURN
END

$IBTC FSPDT LIST
SUBROUTINE FSPDT(RSP,THSP,FNT,THTS,THTF,DTHT,PHI,VX,VY,VZ)
COMMON A1,A3,B1,A9,A10,A11,WAVE
C SEARCH FOR LARGEST SCALAR PRODUCT IN SECTION
FND=ABS((THTF-THTS)/DTHT)
ND=FND+1
FNT=0.
RSP=0.
THSP=0.
DO 100 I=1,ND+1
F1=1-I
THT=F1*DTHT+THTS
R=RAD(THT)
CALL FNORM(FNX,FNY,FNZ,R,THT,PHI)
FNE=VX*FNX+VY*FNY+VZ*FNZ
FNE=ABS(FNE)
1F(FNE.GT.FNT) GO TO 11
FNT = ABS(FNE)
RSP = R
THSP = THT
11 CONTINUE
100 CONTINUE
RETURN
END

SIBFTC FINT. LIST
SUBROUTINE FINT(SSS,FCTI,FLL,FUL,ERRR,NX)
C     INPUT FCTI,FLL,FUL,ERRR,NX
C     EXTERNAL DECLARATION FOR FUNCTION FCTI REQUIRED
COMPLEX SSS,SS,FSS,FCTI,TRAP,TRAZ,SIMP,SIMZ,FNCP,FNCP
FN=NX
DEL=(FUL-FLL)/FN
SSS=(0.,0.)
ERR=0.01*ERRR/FN
A=FLL
DO 40 NNX=1,NX,1
MXX=0
B=A+DEL
SS=(0.,0.)
MX = 2
DX=DEL/2.
LX=1
X=A
GO TO 15
5 TRAZ=DX*SS
MX = 1
LX = 1
DX = DEL
10 SS = 0.
LX = LX+1
DX = 0.5*DX
X = A + DX
15 DO 20 IX=1,MX,1
FSS=FCTI(X)
SS=SS+FSS
20 X=X+2.*DX
IF(LX.EQ.1) GO TO 5
MX = 2*MX
TRAP=0.5*TRAZ+DX*SS
DIF=ABS(TRAP-TRAZ)
IF(DIF.GE.DIP) MXX=MXX+1
DIP=DIF
SIMP=(4.*TRAP-TRAZ)/3.
FNCP=(16.*SIMP-SIMZ)/15.
ER=ABS(1.-FNCP/FNCP)
TRAZ=TRAP
SIMZ=SIMP
FNCP=FNCP
IF(LX.LT.4) GO TO 10
IF(MXX.GT.4) GO TO 30
IF(ER.GT.ERR) GO TO 10
30 SSS=SSS+FNCP
A=A+DEL
50 CONTINUE
RETURN
END

SIBFTC FNORM. LIST
SUBROUTINE FNORM(FNVX,FNVY,FNVZ,R,THT PHI)
C     INPUT R,THT PHI
COMMON AR1,AR3,BRI,AR9,AR10,AR11,WAVE
ST=SIN(THT)
CT=COS(THT)
SP = \sin(\phi)
CP = \cos(\phi)
U = AR1*ST*ST + AR3*CT*CT + BR1*ST*CT
V = AR9*CT + AR10*ST
UTH = 2*(AR1 - AR3)*ST*CT + BR1*(CT*CT - ST*ST)
VTH = -AR9*ST + AR10*CT
F1 = 2*R*U + V
F2 = R*UTH + VTH
FX = ST*CP*F1 + CT*CP*F2
FY = ST*SP*F1 + CT*SP*F2
FZ = CT*F1 - ST*F2
FN = SQRT(FX*FX + FY*FY + FZ*FZ)
FNVX = FX/FN
FNVY = FY/FN
FNVZ = FZ/FN
RETURN

FUNCTION R(A(THT))
C
COMMON AR1, AR3, BR1, AR9, AR10, AR11, WAVE
ST = SIN(THT)
CT = COS(THT)
C1 = AR1*ST*ST + AR3*CT*CT + BR1*ST*CT
C2 = AR9*CT + AR10*ST
C3 = AR11
IF(C1.EQ.0.) GOTO 11
ARG = SQRT(C2*C2 - 4*C1*C3)
R = (-C2 + ARG)/(2*C1)
R2 = (-C2 - ARG)/(2*C1)
IF(R2.LT.R.AND.R2.GT.0.) R = R2
GO TO 12
11
R = -C3/C2
12
RAD = R
RETURN
END

SUBROUTINE FCOMM(I)
COMMON RA1, RA3, RB1, RA9, RA10, RA11, WAVE, DATA/AR1(20), AR3(20), BR1(20),
CAR9(20), CAR10(20), RA11(20)
RA1 = AR1(I)
RA3 = AR3(I)
RB1 = BR1(I)
RA9 = AR9(I)
RA10 = AR10(I)
RA11 = AR11(I)
RETURN
END

$DATA
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

1. **COMPLEX, DIMENSION, COMMON, & FORMAT DECLARATIONS**

2. **READ - NO. OF SECTIONS - (N)**
   - WAVELENGTH (WAVE)
   - INCIDENCE ANGLE (THI)

3. **READ (I = 1, N)**
   - COEFFICIENTS OF 2nd DEGREE EQUATION,
   - SECTION BOUNDARIES
     - THT(I), THTF(I)

4. **DO 100 ON INCIDENCE ANGLE**

5. **DO 100 ON WAVELENGTH**

6. **IF (THT > π/2) INVERT TARGET CALL FMHVT**

7. **CALCULATE INCIDENCE VECTOR T AND INCIDENT FIELD VECTOR E**
Determine Specular Point

DO 410 ISP = 1, N

CALL FCOMM (ISP)
CALL FSPDT
CALL FNORM

SELECT LARGEST PRODUCT OF $-\n \cdot N = FNSPP$
LOCATION = (RSPP, THSPP, NSP)

410
CONTINUE

Determine Creeping Wave Attachment Points

Do 510 ICW = 1, N

CALL FCOMM (ICW)
CALL FSPDT
CALL FNORM

SELECT LARGEST PRODUCT OF $|E \cdot N| = FCWU$ ON UPPER HALF OF TARGET AT LOCATION (RSCWU, THCWU, ICWJ)

Loop on ICW
CONTINUE

DO 520 ICW = I, N

CALL FC0MM (ICW)
CALL FSPDT
CALL FNNORM

SELECT LARGEST PRODUCT OF 12 - N - FCWL ON LOWER
HALF OF TARGET AT LOCATION (RSCWL, TWCWL, NCWL)

CONTINUE

CALCULATE GEOMETRICAL OPTICS CONTRIBUTION

CALL FC0MM (NSP)
CALL DgFFGO
COMPUTE GAUSSIAN CURVATURE

PHASP = PHASE (SPECULAR PT.)

EGO = \(-\left(\frac{1}{\text{GAUSS}}\right)^{\frac{1}{2}}\) \cdot \frac{PHASP}{2}.\)
CALCULATE CREEPING WAVE PATH LENGTH CWLI = 0.

1. Do 522 NCPUL = NCWU, N, 1

CALL FCOMM(NCPU)
TCW1 = THCWU
IF (NCPU > NCWU) TCWI = THTI(NCPU)
TCW2 = THTF(NCPU)

CALL FINT (RCMLI, ds, TCW1, TCW2)
PATH LENGTH = RCMLI

CWLI = CWLI + RCMLI

522 CONTINUE

CWL2 = 0.

2. Do 524 NCPL = NCWL, N, 1

CALL FCOMM(NCPL)
TCW1 = THCWL
IF (NCPL > NCWL) TCWI = THTI(NCPL)
TCW2 = THTF(NCPL)

LOOP ON NCPL
CALL FINT(RCML2, ds, TCW1, TCW2)
PATH LENGTH = RCML2

CWL2 = CWL2 + RCML2

CONTINUE

FKL1 = 2π * CWL1/WAVE
FKL2 = 2π * CWL2/WAVE
FKLCW = FKL1 + FKL2

CALCULATE CREEPING WAVE
ATTENUATION ALONG PATH
ALPI = (0, 0), ALPZ = (0, 0)

DO 530 NCPU = NCWU, N, 1

CALL FCOMM(NCPU)
TCW1 = THCWU
IF(NCPU > NCWU) TCW1 = THTI(NCPU)
TCW2 = THTF(NCPU)

CALL FINT(PALPI, α(ds), TCW1, TCW2)
ATTENUATION = PALPI

ALPI = ALPI + PALPI
CONTINUE

DO 532 NCPL=NCWL,N,1

CALL FCOMM(NCPL)
TCWI = THCWL
IF (NCPL > NCWL) TCWI = THF(NCPL)

CALL FINT(PALP2, wcI, TCWI, TCW2)
ATTENUATION = PALP2

ALP2 = ALP1 + PALP2

CONTINUE

ATTEN = ALP1 + ALP2

CALCULATE CREEPING WAVE FIELDS

PHCW1 = SPHASE(THCW1, RSCWU)
PHCW2 = SPHASE(THCW2, RSCW2)

CALCULATE $D^2$
CALL DIFFGO(THCWU, RSCWU)
CALL DIFFGO(THCWL, RSCWL)
\[ D_{SQC} = D_{sq}(\rho_1, \rho_2, \text{WAVE}) \]

\[ ECW = -2 \times D_{SQC} \times \text{PHCW1} \times \text{PHCW2} \times c \exp(-\text{ATTEN} - jF_{KLCW}) \]

\[ ETOT = EGO + ECW \]

**CALCULATE ECHO AREA**

**WRITE**

100

**CONTINUE**

**STOP**
REFERENCES


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 $E\theta$ (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 $E\phi$ (perpendicular) polarization the solution is obtained using geometrical optics, and the creeping wave is neglected. The computed results for $E\theta$ polarization are in good agreement with measurements for prolate spheroids, prolate spheroid-sphere, and prolate spheroid-oblate spheroid combinations.
Radar Cross Section  
Backscatter  
Surface of Revolution  
Computer Program  
Geometrical Optics  
Creeping Waves