ACKNOWLEDGMENT

REFERENCES

N00014-76-C-0067

UNCLASSIFIED REF-79-7
The programs presented compute the acoustical reflectivity of a sphere in a fluid medium and differ in the allowed physical properties of the spheres. The principal outputs of the programs are plots of reflectivity, \( R^2 \), as a function of size-frequency, \( k a \). The reflectivity is defined as \( R^2 = \frac{8b}{a^2/4} \), where \( b \) is the backscattering cross-section of the sphere, and \( a^2/4 \) is the backscattering cross-section of a completely reflecting sphere of radius \( a \). The variable \( k \) is the wavenumber in the medium. These plots may be considered...
as dimensionless representations of target strength vs. frequency. The conversions are $TS = 10 \log \left(\frac{R^2a^2}{4}\right)$ and $f = \frac{cka}{2\pi a}$ with (a) in meters, (c) in meters per second and (f) in kHz. This relation for frequency is such that the product of frequency in kHz and radius in mm is 240 when $ka = 1$. These programs have been used primarily for low contrast cases at relatively low values of $ka$ (less than 10). Some adjustments to the tolerance parameters may be necessary for other cases.
BACKSCATTERING PROGRAMS FOR
SPHERICAL TARGETS*

Richard K. Johnson¹
Lawrence Flax²
David Standley¹

Reference 79-7
April 1979

¹Oregon State University, School of Oceanography
²Naval Research Laboratory

*Supported by the Office of Naval Research
Introduction

These programs compute the acoustical reflectivity of a sphere in a fluid medium. The programs differ in the allowed physical properties of the spheres. The principal outputs of the programs are plots of reflectivity, $R^2$, as a function of size-frequency, $ka$. The reflectivity is defined as

$$R^2 = \frac{\sigma_b}{a^2/4} = \frac{1}{(a_{\text{back}})} (a_{\text{back}})$$

where $\sigma_b$ is the backscattering cross-section of the sphere, and $a^2/4$ is the backscattering cross-section of a completely reflecting sphere of radius $a$. The variable $k$ is the wavenumber in the medium.

These plots may be considered as dimensionless representations of target strength vs. frequency. The conversions are

$$TS = 10 \log (R^2 a^2/4)$$

$$f = cka/2 \omega$$

with $a$ in meters, $c$ in meters per second and $f$ in kHz. This relation for frequency is such that the product of frequency in kHz and radius in mm is 240 when $ka = 1$.

These programs have been used primarily for low contrast cases (with $g$ and $h$ near one) at relatively low values of $ka$ (less than 10). Some adjustments to the tolerance parameters may be necessary for other cases.

Program Variables

**Input Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB1</td>
<td>$\beta_c$</td>
<td>compressional attenuation in sphere (dB/wavelength)</td>
</tr>
<tr>
<td>AB2</td>
<td>$\beta_s$</td>
<td>shear attenuation in sphere (dB/wavelength)</td>
</tr>
<tr>
<td>DRATIO</td>
<td>$g$</td>
<td>density of sphere/density of medium</td>
</tr>
<tr>
<td>PRATIO</td>
<td>$h$</td>
<td>compressional speed in sphere/speed in medium</td>
</tr>
<tr>
<td>SRATIO</td>
<td>$s$</td>
<td>shear speed in sphere/speed in medium</td>
</tr>
<tr>
<td>Z</td>
<td>$ka$</td>
<td>size-frequency parameter</td>
</tr>
</tbody>
</table>

**Output Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>$R$</td>
<td>reflectivity or form function</td>
</tr>
<tr>
<td>G2</td>
<td>$R^2$</td>
<td>reflectivity squared</td>
</tr>
</tbody>
</table>
Fluid Sphere

The program SPHERF is a simplified version of SPHERE. It calculates $R^2$ for a sphere which differs from the fluid medium only in density and compressional sound speed.

Bibliography


PROGRAM SPHERF
C
    Computes back scattering form function for a fluid sphere
C
    Written by Larry Flax
C
    Modified by R K Johnson and D Standley
C
DIMENSION ZZ(1000), IE(10), EG(1000)

COMPLEX CITEMP, SUM, F
DATA NO/2HN /PD/0.1/, EPS/0.0005/
DATA PLT/P(14)PIROTPISIZEvNXCHPNYCHXFMT,YFMTXLARYLAB

DIMENSION XFMT(2), YFMT(2), XLAB(IO), YLAB(IO), PLID(3)
DIMENSION DATLBL(10), PRL(3), PDRL(3)
EQUIVALENCE (DATLBL, PLID(3))
DATA PLID/'SPHERF', ' ', ' '
DATA P/7., 8., 2., 2., 0., 0., 0., 0., 0., 0., 1., 10., 1., 10., 1., 10., 1., 10. /
DATA XFMT/('F4.' , 'I') ' ', '/', YFMT/('F5.' , 'O') '/'
DATA PENUP/1/, PENDWN/0/
DATA PRL/'H ', ' ', ' '
DATA ORL/'G ', ' ', ' ' /
DATA XLAB/ 'KA ', ' '
DATA YLAB/ 'R (' , 'DB) ', ' ' /
INTEGER PENUP, PENDWN

C GET ACOUSTIC PROPERTIES AND KA RANGE

CALL DATMSG
CALL DTMSG(DATLBL) !FOR PLOT
ACCEPT 101, DRATIO, PRATIO !G, H
ACCEPT 101, ZFROM, ZTO, ZSTEP
IEND=(ZTO-ZFROM)/ZSTEP+1.5
IF(IEND.LE.1000)GO TO 30
TYPE 110
GO TO 20
30 TYPE 103
ACCEPT ITYP
IF(ITYP.EQ.NO)GO TO 50
50 CONTINUE
CI=(O.91.)
G2LM=1000. !SET G2L MINIMUM ARBITRARILY HIGH

DO 800 Z=1.IEND
Z=FLOAT(Z-1)*ZSTEP+ZFROM
ZL=Z/PRATIO
TE1=0.
TE2=0.
TEMP=(O., O.)
CALL SBSAJ(ZL,0,BJ,0,IE(1))
CALL SBSAJ(ZL,0,BJ,1,IE(2))
CALL SBSAJ(ZL,0,BY,0,IE(3))
DO 6 K=1,50
L=K-1
X=(2*L+1)*(-1)**L
CALL SBSAJ(ZL,K,BJ,0,IE(4)) !Spherical Bessel Functions
DO 50 I=1,6
IF(IE(I).LT.0)GO TO 80
50 IE(IK)=I
80 IE(IK)=0
C. INCLUDE MORE MODES UNTIL CHANGE IS LESS THAN EPS

0069 IF (GEl .LE. EPS. AND. 0E2 .LE. EPS) GO TO 13

0070 IF (GE1 .LE. EPS. AND. GE2 .LE. EPS) GO TO 13

0072 TF1 = T

0074 BJ = BJI

0075 BJL = BJL

0076 BY = BY1

0077 6 CONTINUE

0078 13 F = 2. * TEMP / Z

0079 F1 = REAL (F)

0080 F2 = AIMAG (F)

C. MODULUS=SORT(BACKSCATTERING/GEOMETRIC CROSS-SECTION)

0081 G2 = F1 * F1 + F2 * F2

0082 G = SORT (G2)

0083 Z(IZ) = ALOG10 (Z)

0084 G2L = 10. * ALOG10 (G2)

0085 IF (ITYP .NE. 0) TYPE 107, Z, G, G2L

0087 G2(IZ) = G2L

0088 IF (G2L .LT. G2LM) G2LM = G2L

0090 800 CONTINUE

C.------------------------
C PLOTTING ROUTINE:
C

0091 TYPE 108, G2LM

0092 ACCEPT 101, YS

0093 IF (YS .GE. 0) GO TO 1

0095 ZLOGS = ALOG10 (ZFROM)

0096 ZLOGE = ALOG10 (ZTO)

0097 MIN = INT ((SIGN (ABS (ZLOGS) + 0.96, ZLOGS))

0098 IF (MIN .GT. 0) MIN = MIN - 1

0100 XMIN = 10. ** MIN

0101 MAX = INT ((SIGN (ABS (ZLOGE) + 0.96, ZLOGF))

0102 IF (MAX .LT. 0) MAX = MAX + 1

0104 XMAX = 10. ** MAX

0105 P(1) = 7. X LENGTH

0106 P(2) = 8. Y LENGTH

0107 P(3) = 2.

0108 P(4) = 2.

0109 P(5) = XMIN

0110 P(6) = XMAX

0111 P(7) = YS

0112 P(8) = YS + 80.

0113 P(9) = P(5)

0114 P(10) = YS

0115 CALL AXIS (3)

0116 ENCODE (4 + 109, PRL (2) ) DRATIO

0117 ENCODE (4 + 109, PRL (2) ) PRATIO

0118 CALL PLOTXY (P(5), P(8), PENUP, 0)

0119 CALL PLOT (5, IX, IT)

0120 IX = IX + 200
CALL STRING(I,X,Y,28,FLID,0.2)  !LABEL THE PLOT
CALL STRING(I,X,Y-40,10,DRL,0.2)
CALL STRING(I,X,Y-120,10,PRL,0.2)
CALL PLOTXY(P(5),P(7),PENUP,0)
DO 600 I=1, IEND
YY = G0(I)
XXX = ZZ(I)
IF(I .EQ. 1) CALL PLOTXY(XXX, YY, PENUP, 0) !GO TO FIRST POINT
CALL PLOTXY(XXX, YY, PENUP, 0)
600 CALL PLOTXY(XXX, YY, PENUP, 0)
CALL PLWAIT
CALL PLOT(2,1885)
CALL PLOT(3)
CALL PLWAIT
CALL PLOTXY(XXX, YY, PENUP, 0)
GO TO 1
100 FORMAT(' ENTER DRATI0, PRATI0 ... [G,H]: ','$
101 FORMAT(3F10.4)
102 FORMAT(' ENTER ZFROM, ZTO, ZSTEP : ','$
103 FORMAT(' TYPE RESULTS?(Y/N): ','$
104 FORMAT(A2)
105 FORMAT(' KA MODE MODULUS 20LOG '/)
106 FORMAT(' REQ PREC NOT ACHIEVED. ROUTINE #',I2, ' ER=',I2)
107 FORMAT(F10.3,I8,F14.4,F11.1)
108 FORMAT('0PLOT:GIVE START OF Y SCALE (MAX VALUE=',F5.1,') : ',')
109 FORMAT(F6.3)
110 FORMAT(' TOO MANY INCREMENTS ... TRY AGAIN'//)
END
Fig. 1  Output plot from SPHERF.
Elastic Sphere

The program SPHERE calculates $R^2$ for an elastic sphere in a fluid medium. The original version of this program was written by Larry Flax.

Bibliography


PROGRAM SPHERE
C
C COMPUTES BACK SCATTERING FORM FUNCTION FOR AN ELASTIC SPHERE
C WRITTEN BY LARRY FLAX, NRL
C MODIFIED BY R K JOHNSON, T KEFFER, AND D STANDLEY
C
DIMENSION ZZ(500), IE(10), GG(500), A(3,3)
COMPLEX CIPTEMPPSUMF
REAL LMlp LP1
DATA NO/2HN /PD/0.1/PEPS/0.001(/
C ---
COMMON /PLT/P(14) ,IROTPISIZEPNXCH.NYCHXFMTYFMT.XLABvYLAB
DIMENSION XFMT(2) ,YFMT(2) ,XLAB(10),YLAB(10) ,PLID(3)
DIMENSION DATLBL(10),PRL(3),DRL(3),SRL(3)
EQUIVALENCE (DATLBL,PLID(3))
DATA PLID/'SPHE'F'RE'/'
DATA P/.8.2,0.O,Op . .. ,0. .,10Pl./
DATA XFMT/'CF4.'fi'1) 'P/PYFMT/'(F5.'0')'/
DATA PENUP/1/p PENDWN/0/
DATA PRL/'H-92*'/
DATA DRL/'G-82*'/
DATA XLAB/' KA ',9*'/
DATA SRL/IS
DATA YLAB/' R ('P'DB)'98*'
INTEGER PENUP, PENDWN
C
GET ACOUSTIC PROPERTIES AND KA RANGE
CALL DATMSG
CALL DTMSG(DATLBL,IFOR PLOT
ACCEPT 101PDRATIO,PRATIORSRATIO,SRATIOT
GO TO 103
ACCEPT 101, ZFROM,ZTO,ZSTEP
IEND=(ZTO-ZFROM)/ZSTEP+i.5
IF(IEND.LE. 500)GO TO 30
GO TO 20
CONTINUE
40 IF(ITYP.EQ.NO)GO TO 40
C=SRAST#2/(PRATI0+2.*SRAST0#2) !POISSON'S RATIO
I=C=(0.+1.)
GZL=1000. !SET G2L MINIMUM ARBITRARILY HIGH
C
START LOOP
DO BOO ZZ=19IEND
Z=FLOAT(Z-1)*ZSTEP+ZFROM
ZL=Z/PRATIO
Z5=Z/SRATIO
TE1=0.
TE2=0.
TEMP=(0.+0.,)
CALL SBESJ(Z,0,BJ1,D+IE(1))
CALL SBESJ(ZL,0,BJ1,D+IE(2))
CALL SBESJ(Z,0,BYIE(3))
CALL SBESJZS,0,BJS,D+IE(4))
DO 6 K=1,50
L=L+1
X=(ZL+L)*X-1*2MOD(L,2))
CALL SBESJZK,BJ1,D+IE(5)) !SPHERICAL BESSEL FUNCTIONS
CALL SBESJZL,K,BJ1,D+IE(6))
CALL SBESJZS,K,BJS,D+IE(7))
CALL SBESYCZPKPBYIPIE(S))

DO 80 IECICK = 1,8

IF(IECICK).EQ.0)GO TO 80

TYPE 106, IECICK=IECICK

80 CONTINUE

FIRST DERIVATIVES

SECOND DERIVATIVES

END

C..INCLUDE MORE
MODES
UNTIL
CHANGE
IS
LESS THAN EPS

IF(OL1.LE.EPS.AND.OE2.LE.EPS)GO TO 13

TE1=T

13 F = 2. * TEMP / Z 'FORM FUNCTION

F1=REAL(F)

F2=AIMAG(F)

SEARCH FOR MINIMUM VALUE OF G2L

IF(G2L.LT.G2LM)G2LM=G2L

CONTINUE

C.....MODULUS=SQRT(BACKSCATTERING/GEOMETRIC CROSS-SECTION)

C PLOTTING ROUTINE:

C------------

C--------------
FORTRAN IV

0120 XMIN = 10. ** MIN
0121 MAX = INT((SIGN(ABS(ZLOGE)+0.96PZLOGE)))
0122 IF(MAX .LT. 0)MAX = MAX + 1
0124 XMAX = 10. ** MAX
0125 P(1) = 7. ** X LENGTH
0126 P(2) = 8. ** Y LENGTH
0127 P(3) = 2. **
0128 P(4) = 2. **
0129 P(5) = XMIN
0130 P(6) = XMAX
0131 P(7) = YS ** YMIN
0132 P(8) = YS + B0. ** YMAX
0133 P(9) = P(5) ** XO
0134 P(10) = YS ** YO
0135 CALL AXIS(3) ** DRAW AXES
0136 ENCODE(6,109), DRL(2) ** DRATIO
0137 ENCODE(6,109), PRL(2) ** PRATIO
0138 ENCODE(6,109), SRL(2) ** SRATIO
0139 CALL PLOTXY(P(5), P(8), PENUP, 0) ** GO TO (XMIN,YMAX)
0140 CALL PLOT(5, IX, IY)
0141 IX = IX + 200
0142 CALL STRING(IX, IY, 28, PLID, 0, 2) ** LABEL THE PLOT
0143 CALL STRING(IX, IY-60, 10, DRL, 0, 2)
0144 CALL STRING(IX, IY-120, 10, PRL, 0, 2)
0145 CALL STRING(IX, IY-180, 10, SRL, 0, 2)
0146 CALL PLOTXY(P(5), P(7), PENUP, 0)
0147 599 DO 600 I = 1, IEND
0148 YY = GG(I)
0149 XXX = ZZ(I)
0150 IF(I .EQ. 1) CALL PLOTXY(XXX, YY, PENUP, 0) ** GO TO FIRST POINT
0152 600 CALL PLOTXY(XXX, YY, PENDW, 0)
0153 CALL PLWAIT
0154 CALL PLOT(2, 1885)
0155 CALL PLOT(3)
0156 CALL PLWAIT
0157 GO TO 1
0158 100 FORMAT(' ENTER DRATIO, PRATIO, SRATIO ... [G,H,S]: ',$)
0159 101 FORMAT(3F10.4)
0160 102 FORMAT(' ENTER ZFROM, ZTO, ZSTEP : ',$)
0161 103 FORMAT(' TYPE RESULTS? [Y/N] : ',$)
0162 104 FORMAT(A2)
0163 105 FORMAT(/ ' KS MODE MODULUS 20 LOG '/)
0164 106 FORMAT(' REQ PREC NOT ACHIEVED. ROUTINE END',$)
0165 107 FORMAT(F10.3, I6, F14.4, F11.1)
0166 108 FORMAT(' OPLT: GIVE START OF Y SCALE (MAX VALUE =', F5.1, '): ',$)
0167 109 FORMAT(F6.3)
0168 110 FORMAT(' TOO MANY INCREMENTS ... TRY AGAIN'/)
0169 END
Fig. 2  Output plot from SPHERE for a case with very low shear speed.
Fig. 3  Output plot from SPHERE for a case with moderate shear speed.
Viscoelastic Sphere

The program ABSPHR calculates $R^2$ for a viscoelastic (absorbing) sphere in a fluid medium. The original version of this program was supplied by Tokahi Hasegawa.

Bibliography


PROGRAM AsePHR

C========================================================================================================
C AsePHR
C
C ACOUSTIC BACK SCATTERING FROM AN ABSORBING SPHERE:
C
C THIS PROGRAM CALCULATES THE REFLECTION FORM FUNCTION FOR AN
C ABSORBING ELASTIC SPHERE BASED ON TAKAHI HASEGAWA, YOSHIKO
C KITAGAWA AND YUMIKO WATANABE. SOUND REFLECTION FROM AN
C
C MODIFIED BY R K JOHNSON AND D STANDLEY
C========================================================================================================

0002 COMPLEX XI,X2,JN,JB,AB,XY,EH,EH,PH,CH

0004 REAL KIA,K2A

0005 DIMENSION XX(500),YP(500),JN(50),JB(50),BJ(50)

0006 DIMENSION BM(50),ALP(50),BJD(50),DJ(50)

0007 DATA YES/YES'/, EPS/0.001/, PX/54.5757/

C---PLOT COMMON, ETC.

0008 COMMON /PLT/P(14),IROT,ISIZE,NXCH,NYCH,XFMT,YFMT,XLAB,YLAB

0009 DIMENSION XFMT(2),YFMT(2),XLAB(10),YLAB(10),PLID(3)

0010 DIMENSION DLBL(10),ABLL(3),AB2L(3),PRL(3),DRL(3),SRL(3)

0012 DATA PLID/'AP','HR'/

0013 DATA P/7.,S..2.,T2.,0.,O.,O.,O.,O.,O.,O.,O.,O.,O.,O./

0014 DATA XFMT/"(F4.1)",/,

0015 CALL DATMSG

C GET OPTION
C 1) CALCULATE AND SAVE RESULTS ON FILE FOR LATER PLOTTING
C 2) CALCULATE AND PLOT RESULTS
C 3) READ RESULTS FILE AND PLOT

0020 TYPE 104

0021 ACCEPT 100, IQPT

0022 IF(IQPT.EQ.3) GOTO TO 410

0023 TYPE 900

0024 ACCEPT 100, AB1, AB2

0025 TYPE 906

0026 ACCEPT 100, DRAT1, PRAT1, SRAT1

0027 CC1 = 1. / PRAT1

0028 CC2 = 1. / SRAT1

0029 TYPE 907

0030 ACCEPT 100, ZFROM, ZTO, ZSTEP

0031 IF(ZSTEP.EQ.0.0) ZSTEP = 0.25

0032 IF(MAX = IFIX (ZTO - ZFROM) / ZSTEP + 1.0001)

0033 IF(INAX .LE. 500) GOTO TO 10

0034 TYPE 114

0035 GO TO 9

0036 TM = 1000. ! LARGE MINIMUM

0037 IMR = 0

0038 ET = SECnds(0.) ! ELAPSED TIME

C========================================================================================================
C---START LOOP

0040 DO 45 IM = 1, INAX

0041 XX(I) = ZFROM + ZSTEP * FLOAT(I-1)
DO 221 IC = 1, NNM
DO 222 IC = 1, NNM
CALL CSBBSJ(XI, JN, NNM, IER)
IF(IER .EQ. 0) GO TO 4
CALL BSBSBS(NNM, X, SN)
PJJ = 0.
PNM = 0.
DO 20 NI = 1, NNM
T = FLOAT(NI - 1)
N1 = N
N2 = N + 1
TC = CMPLX(T,O.)
A = TC * JN(N1) - X1 * JN(N2)
B = A - JN(N1)
D1C = CMPLX(1.,O.)
D2C = CMPLX(2.,O.)
G = (X2**2 / D2C - TC - (TC - D1C) * JN(N1) - D2C * X1 * JN(N2))
A = A / B
D = 0 / B
C = D2C + TC - (TC + D1C) * JB(N1)
E = D2C + TC - (TC - D1C) * JB(N1) + D2C * X2 * JB(N2)
H = D2C + (T C + D1C) * ((D1C - TC) * JB(N1) + X2 * JB(N2))
I = C / E
E = H / E
PSC = CMPLX(S,N,O.)
DNNM = CMPLX(DNATD,O.)
F = PSCS (A - B) * X2**2 / ((D - E) * DNNM)
PJ = T * BJ(N1) - X * BJ(N2)
PH = T * BNH(N1) - X * BNH(N2)
CM = CMPLX(BJ(N1), -BNH(N1))
PH = CMPLX(PJ, -PH)
CN = CMPLX(PJ,0.) - F3CMPLX(BJ(N1),0.) / (F3HM-PH)
ALP31 = REAL(CN)
BET(N1) = IMAH(CN)
NS = N - 1
YN = (2, FLOAT(N5) + 1.) * (-1.)**NS
PJJ = PJJ + YN * BET(N)
C......TEST FOR CONVERGENCE
0120 PNN = PNN + YN * ALP(N)
0121 T1 = ABS(YN * BET(N)) / PJJ
0122 T2 = ABS(YN * ALP(N)) / PNN
0123 IF(ABS(T1), LT. EPS, AND. ABS(T2), LT. EPS) GO TO 201
0125 20 CONTINUE
0126 N = N + 1
0127 IF(NN+2, GT. NNN) GO TO 200
0129 NN = NN + 1
0130 GO TO 25
0131 200 TYPE 113, XPNNMPNMN
0132 201 YP(I) = 2.0 * SORT(PJJ**2 + PNNS*2) / X
0133 IF(YP(I) < YM) YN = YP(I)

C......TYPEOUT OPTION
0134 ISW = IPEEK('177570')
0135 IF(ISW, LT. 0) GO TO 30
0139 IF(ISW, ED. 0) GO TO 291
0141 IF(IHDR, EQ. 0) GO TO 110
0143 IHDR = 1
0144 DT = SECNDS(ET)
0145 TYPE 111, X, YP(I) * 20. * ALOG10(YP(I))
0146 ET = SECNDS(0.)
0147 GO TO 30
0148 291 IMAX = I
0149 GO TO 31
0150 30 CONTINUE
0151 31 TYPE 102, (XX(I), YP(I), 20. * ALOG10(YP(I)), IMAX)

C----------------------
C CHECK OPTIONS
C
0152 GO TO (400, 450, 410), IOPT
0153 400 TYPE 106
0154 CALL ASSIGN(1, -1, 'NEW')
0155 CALL ASSIGN(1, -1, 'RDO')
0156 WRITE(1, 108) (XX(I), YP(I), IMAX)
0157 CALL CLOSE(1)
0158 CALL EXIT
0159 410 TYPE 112
0160 CALL ASSIGN(1, -1, 'NEW')
0161 CALL ASSIGN(1, -1, 'RDO')
0162 READ(1, 107) AB1, AB2, DRATIO, PRATIO, SRATIO, YM, IMAX
0163 CALL CLOSE(1)
0164 XX = XX(I)
0165 ZTO = XX(IMAX)

C----------------------
C PLOTTING ROUTINE:
C
0166 450 TYPE 109
0167 ACCEPT 904, YS
0168 ZLOGS = ALOG10(ZLOGS)
0169 ZLOGE = ALOG10(ZTO)
0170 MIN = INT((SIGN(ABS(ZLOGS), 0.96) * ZLOGS)))
0171 IF(MIN, LT. 0) MIN = MIN - 1
0173 XMIN = 10. ** MIN
0174 MAX = INT((SIGN(ABS(ZLOGE), 0.96) * ZLOGE)))
0175 IF(MAX, LT. 0) MAX = MAX + 1
0177 XMAX = 10. ** MAX
0178 P(1) = 7.
0179 P(2) = 8.
0180 P(3) = 2.
0181 P(4) = 2.
0182 P(5) = XMIN
0183 P(6) = XMEX

16
0184  P(7) = YS  !YMIN
0185  P(8) = YS + 80.  !YMAX
0186  P(9) = P(5)  !XO
0187  P(10) = YS  !YO
0188  CALL AXIS(3)  !DRAW AXES
0189  ENCODE(6,103,ABIL(2))AB1
0190  ENCODE(6,103,AB2(2))AB2
0191  ENCODE(6,103,BRL(2))DRATIO
0192  ENCODE(6,103,PRL(2))PRATIO
0193  ENCODE(6,103,SRL(2))SRATIO
0194  CALL PLOTXY(P(5),P(8),PENUP,0)  !GO TO (XMIN,YMAX)
0195  CALL PLOT(P(5),IX,IV)
0196  IX = IX + 200
0197  CALL STRING(IX,IV,28,PLID,0)  !LABEL THE PLOT
0198  CALL STRING(IX,IV-Y,60,ABIL,0)  !LABEL THE PLOT
0199  CALL STRING(IX,IV-Y-120,AB2,0)
0200  CALL STRING(IX,IV-Y-240,BRL,0)
0201  CALL STRING(IX,IV-Y-300,PRL,0)
0202  CALL PLOTXY(P(5),P(7),PENUP,0)
0203  CALL PLOTXY(XXX,YYY)  !PLOT THE POINTS
0204  DO 600  I=1,IMAX
0205  YY = LOG10(YP(I))  !ALOG10(YP(I))
0206  XXX = LOG10(XX(I))  !ALOG10(XX(I))
0207  IF(I.EQ.1)CALL PLOTXY(XXX,YYY,PENUP,0)  !GO TO FIRST POINT
0208  600  CALL PLOTXY(XXX,YYY,PENDWN,0)
0209  CALL PLWAIT
0210  CALL PLOT(3)
0211  CALL PLOT(2,1885)
0212  CALL PLOT(3)
0213  CALL PLWAIT
0214  999  GO TO 1
0215  100  FORMAT(8F10.5)
0216  102  FORMAT(3(5X,'KA',6X,'YP',6X,'X1R',6X,'X2R',/3(2X,F7.3,F8.4,2X,F7.1))
0217  103  FORMAT(F7.3)
0218  104  FORMAT(0 OPTION: 1)CALC&SAVE RESULTS 2)CALC&PLOT 3)READ ', 
0219  105  FORMAT(I2)
0220  106  FORMAT(OUTPUT FILE ,'@')
0221  107  FORMAT(6E15.5,16)
0222  108  FORMAT(6E15.5)
0223  109  FORMAT('PLOT:GIVE START OF Y SCALE (MAX VALUE=',F6.1,'!':9)
0224  110  FORMAT(3X,'KA',6X,'YP',4X,'X1R',20LOG3X,'X2R',X1R,'X2R',/3X,'MODE',5X,'X1R',7X,'X2R',7X,'X2R','SECONDS'/)
0225  111  FORMAT(F7.3,F8.4,F7.1,F14,13X,2X,F10.3,2X,FOPF.4,0)
0226  112  FORMAT( INPUT FILE ,'@')
0227  113  FORMAT('YP DID NOT CONVERGE AT X = ','F5.2,
0228  114  1 'NNM = ',13,' NM = ',13)
0229  115  FORMAT(' TOO MANY INCREMENTS ... TRY AGAIN')
0230  116  902  FORMAT(ENTER AB1, AB2 : ',@)
0231  903  FORMAT(2F10.5)
0232  905  FORMAT('OCSSBJ ERROR *',I4,2X,'MODE=',I3,2X,'2E16.7)
0233  906  FORMAT(' ENTER DRATIO, PRATIO, SRATIO ... [G,H,S] : ',@)
0234  907  FORMAT(' ENTER ZFROM, ZTO, ZSTEP : ',@)
0235  END
Fig. 4  Output from ABSPHR for a case with shear attenuation.
Real Spherical Bessel Functions

SBESJ

This procedure evaluates the spherical Bessel function of the first kind, \( j_n(x) \), for real arguments. The method used for computation depends on the argument and the order.

For orders 0 and 1, the exact relations are
\[
\begin{align*}
J_0(x) &= \sin(x)/x \\
J_1(x) &= \sin(x)/x^2 - \cos(x)/x.
\end{align*}
\] (1)

For higher orders and small arguments, the procedure uses the series approximations
\[
J_n(x) = \frac{x^n}{1 \cdot 3 \cdot 5 \cdots (2n + 1)}.
\] (2)

This approximation is acceptable only when
\[ x^2/D < 2n, \]
where \( D \) is the required accuracy of the procedure.

For all other cases, the procedure uses the recurrence relation
\[
J_n + 1(x) = \frac{2n + 1}{x} J_n(x) - J_{n - 1}(x).
\] (3)

This relation can be used either to ascend to higher orders or to descend to lower orders.

Ascending recurrence is used for \( x > 2n \) since, in this case, the error in \( J_n(x) \) will not be increased in \( J_n + 1(x) \).

For the remaining case ( \( x < 2n \)), the procedure uses Miller's device, which is a decreasing recurrence technique. This method uses the approximation \( \hat{J}_m(x) = 0 \) and \( \hat{J}_{m - 1}(x) = 1 \) for some \( m > n \), then uses decreasing recurrence to calculate \( \hat{J}_i(x) \) for \( 0 \leq i < m - 1 \). A scale factor \( P \) is calculated from
\( j_0(x)/\tilde{j}_0(x) \), where \( j_0(x) \) is determined from equation (1). The correct value \( j_n(x) \) is the \( P^*j_n(x) \). This sequence is repeated for higher values of \( m \) until successive values for \( j_n(x) \) differ by less than \( D \).

**SBESY**

This procedure evaluates the spherical Bessel function of the second kind (also called the spherical Neumann function), \( y_n(x) \) or \( n_n(x) \), for real arguments. The method is ascending recurrence and presents no computational problems.

These programs were written by Richard Johnson and Thomas Keffer.

**Bibliography**

SUBROUTINE SBESJ

PURPOSE
  COMPUTE THE J SPHERICAL BESSEL FUNCTION FOR A GIVEN ARGUMENT
  AND ORDER

USAGE
  CALL SBESJ(X,N,BJ,D;IER)

DESCRIPTION OF PARAMETERS
  X -THE ARGUMENT OF THE J BESSEL FUNCTION DESIRED
  N -THE ORDER OF THE J BESSEL FUNCTION DESIRED
  BJ -THE RESULTANT J BESSEL FUNCTION
  D -REQUIRED ACCURACY
  IER-RESULTANT ERROR CODE WHERE
    IER=0 NO ERROR
    IER=1 N IS NEGATIVE
    IER=2 X IS NEGATIVE OR ZERO
    IER=3 REQUIRED ACCURACY NOT OBTAINED
    IER=4 RANGE OF N COMPARED TO X NOT CORRECT (SEE REMARKS)

REMARKS
  N MUST BE GREATER THAN OR EQUAL TO ZERO, BUT IT MUST BE
  LESS THAN
  20+10*X-X**2/3 FOR X LESS THAN OR EQUAL TO 15
  90+X/2 FOR X GREATER THAN 15

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
  NONE

METHOD
  RECURRENCES RELATION TECHNIQUE DESCRIBED BY H. A. ANTONIEWICZ
  IN HANDBOOK OF MATHEMATICAL FUNCTIONS PP. 438-439
  AND 452-453. THIS CODE PATTERNED AFTER THAT FOR
  BESJ.

0001 SUBROUTINE SBESJ(X,N,BJ,D;IER)
0002          BJ=.0
0003          IF(N)10,20,20
0004          10 IER=1
0005          RETURN
0006          20 IF(X)30,31,31
0007          30 IER=2
0008          RETURN
0009          31 IF(X-15.)32,32,34
0010          32 NTEST=20.+10.*X-X**2/3.
0011          GO TO 36
0012          34 NTEST=90.+X/2.
0013          36 IF(N-NTEST)40,38,38
0014          38 IER=4
0015          RETURN
0016          40 IER=0
0017          N1=N+1
0018          BPREV=0
0019          IF (N IS 0 OR 1, COMPUTE DIRECTLY.
0020          IF (N-1) 42,43,49
C
C IF X IS VERY SMALL USE ASCENDING SERIES
0024 49 IF(X*X/D-FLOAT(2*N))300,50,50
C
C IF X IS LARGE, USE INCREASING RECURRENCE
0025 50 IF(X-FLOAT(2*N))60,210,210
C
C IF X IS MIDDLE, USE DECREASING RECURRENCE
0026 60 MA=X+0.
0027 MB=H+IFIX(X)/4+2
0028 MZERO=MAXO(MA,MB)
C
C SET UPPER LIMIT OF M
0029 30hua MMAX=MTEST
0030 DO 190 M="ZERDMMAXP3
0031 FH=1.0
0032 FM=0.0
0033 DO 160 K=1,M-1
0034 MK=M-K
0035 BMK=FLOAT(2*MK+1##FM1/X-FM
0036 FM=FM1
0037 FH=BMK
0038 IF(MK-N-1)160,210,160
0039 140 BJ=BMK
0040 160 CONTINUE
C
C SCALE FACTOR
0041 P=SIN(X)/(X*DMK)
0042 BJ=P-SBJ
0043 IF(ABS(BJ-BPREV)-ABS((B&BJ))##200,200,190
0044 190 BPREV=BJ
0045 IER=3
0046 200 RETURN
C
C INCREASING RECURRENCE
0047 210 BJA=SIN(X)/X
0048 BJ=BJA(SIN(X)/(X*BJ)-COS(X)/X
0049 K=1
0050 220 T=FLOAT(2*K+1)/X
0051 BJC=T*BJB-BJA
0052 K=K+1
0053 IF (K-N) 230,240,230
0054 230 BJA=BJB
0055 BJB=BJC
0056 GO TO 220
0057 240 BJB=BJC
0058 RETURN
C
C ASCENDING SERIES:
0059 300 BJ=1.0
0060 DO 350 IFAC=1,N
0061 BJ=BJX/(FLOAT(2#IFAC+1))
0062 350 CONTINUE
0063 RETURN
0064 END
SUBROUTINE SBESY

PURPOSE
COMPUTE THE Y SPHERICAL BESSEL FUNCTION FOR A GIVEN
ARGUMENT AND ORDER

USAGE
CALL BESY(X,N,B,Y,IER)

DESCRIPTION OF PARAMETERS
X -THE ARGUMENT OF THE Y BESSEL FUNCTION DESIRED
N -THE ORDER OF THE Y BESSEL FUNCTION DESIRED
B -THE RESULTANT Y BESSEL FUNCTION
Y -RESULTANT ERROR CODE WHERE
IER=0 NO ERROR
IER=1 N IS NEGATIVE
IER=2 X IS NEGATIVE OR ZERO
IER=3 BY HAS EXCEEDED MAGNITUDE OF 10**36

REMARKS
X MUST BE GREATER THAN ZERO
N MUST BE GREATER THAN OR EQUAL TO ZERO

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE

METHOD
RECURRENC RELATION AND TRIG FORMULAE AS DESCRIBED
BY H.A. ANTOSIEWICZ IN HANDBOOK OF MATHEMATICAL
FUNCTIONS, PP. 438-439. THIS CODE PATTERNED AFTER
THAT FOR BESY.

SUBROUTINE SBESY(X,N,B,Y,IER)

CHECK FOR ERRORS IN N AND X

IF(N)180,10*10
10 IER=0
IF(X)190,190,20
20 EVALUATE YO AND Y1.

YO=-COS(X)/X
Y1=-COS(X)/(X*X)-SIN(X)/X

CHECK IF ONLY YO OR Y1 IS DESIRED

IF(N-1)100,100,130
RETURN EITHER YO OR Y1 AS REQUIRED

PERFORM RECURRENCE OPERATIONS TO FIND YN(X)
0013  130  YA=YO
0014    YB=Y1
0015     K=1
0016   140  T=FLOAT(7X+1)/X
0017     YC=T*YB-YA
0018 IF(ABS(YC)\textless{}1.0E36)145,145,141
0019  141  IER=3
0020    RETURN
0021  145   K=K+1
0022 IF(K-N)150,160,150
0023  150  YA=YB
0024     YB=YC
0025  GO TO 140
0026  160  BY=YC
0027  170  RETURN
0028  180  IER=1
0029    RETURN
0030  190  IER=2
0031    RETURN
0032    END

#
Complex Spherical Bessel Functions

CSBSJ

This procedure evaluates the spherical Bessel of the first kind, $j_n(Z)$, for complex arguments by means of a modified continued fraction technique. This technique is far more effective than upward or downward recursion methods.

The spherical Bessel function of order $n$ can be expressed in terms of smaller orders as

$$j_n(Z) = \frac{j_n(Z)}{j_{n-1}(Z)} \cdot \frac{j_{n-1}(Z)}{j_{n-2}(Z)} \cdot \ldots \cdot \frac{j_1(Z)}{j_0(Z)} \cdot j_0(Z),$$

(1)

and $j_0(Z) = \sin(Z)/Z$.

Each of the ratios in (1) is found by a modified continued fraction method. Now

$$\frac{j_1(Z)}{j_{1-1}(Z)} = \frac{j_{1+1/2}(Z)}{j_{1-1/2}(Z)}$$

(2)

by definition. From Abramowitz and Stegun²,

$$\frac{j_1(Z)}{j_{1-1}(Z)} = \frac{1}{a_0 - \frac{1}{a_1 - \frac{1}{a_2 - \ldots}}}$$

(3)

where $a_k = 2(i + k)/Z$.

The inverse of this ratio can be rewritten as

$$\frac{j_{i-1}(Z)}{j_i(Z)} = \frac{N_0 * N_1 * N_2 * \ldots}{D_0 * D_1 * \ldots}$$

(4)

where $N_k = A_k - 1/N_{k-1}$, $N_0 = A_0$,

and $D_k = A_{k+1} - 1/D_{k-1}$, $D_0 = A_1$.
In the algorithm, the computation of (4) is continued until \( N_k/D_{k-1} - 1 < \epsilon \), where \( \epsilon \) is the allowable error.

This program was written by David Standley.

Bibliography


SUBROUTINE CSBSJ(X, JJ, M, IER)

C DESCRIPTION OF PARAMETERS
C X - THE ARGUMENT OF THE J SPHERICAL BESSEL FUNCTION DESIRED
C JJ - THE RESULTANT J SPHERICAL BESSEL FUNCTION
C M - THE ORDER OF THE BESSEL FUNCTION DESIRED
C IER - ERROR CODE WHERE:
C IER = 0 NO ERROR
C IER .LT. 0 UNDERFLOW OCCURRED AT ORDER (-IER)
C IER = 2 IMAGINARY PART OF X IS >80. OR <-80.
C IER > 49 ORDER TOO BIG
C
C THE ACCURACY IS 5 TO 6 SIGNIFICANT FIGURES

COMPLEX X, A, SBES, QUO(SO), NUMN, NUMNM, DENN, DENNM, JJ(SO)
COMPLEX ONE, TWO, TWOX, CSX, RATIO
A(N) = TWOX * CMPLX(V+FLOAT(N-1), ZERO)
CI = 1.
EPSR = 1.E-5
EPSI = 1.E-3
ZERO = 0.
ONE = (1., O.)
TWO = (2., O.)
IF(M.GT.49) GO TO 514 ! ORDER TOO BIG
XR = REAL(X)
XI = AIMAG(X)
CSX = CSIN(X)
VV = FLOAT(M)
IXR = 0
IXI = 0
ICSR = 0
IF(ABS(XI).GT.80.) GO TO 513 ! OVERFLOW CONDITION!
IF(XR.EQ.0.) GO TO 4
ICSR = IFIX(ALOG10(ABS(REAL(CSX))))
IXR = IFIX(ALOG10(ABS(XR)))
4 ICSI = 0
ICSI = 0
IF(XI.EQ.0.) GO TO 5
ICSI = IFIX(ALOG10(ABS(AIMAG(CSX))))
IXI = IFIX(ALOG10(ABS(XI)))
5 V = VV + 1.5
IF(VV.EQ.0.) GO TO 501 ! CALCULATE DIRECTLY
TWOX = TWO / X

C CALCULATE THE RATIO OF J(V-1)/J(V)
C
DO 500 IV = 1, IFIX(VV) ! GET NECESSARY RATIOS
500 V = V - 1.
L = IFIX(V - 0.5)
FORTRAN IV

0038 NUMNM1 = A(I)
0039 DENNM1 = A(I+2)
0040 QUO(L) = NUMNM1
0041 N = 1
0042 10 N = N + 1
0043 NUMN = A(N) - ONE / NUMNM1
0044 DENN = A(N+1) - ONE / DENNM1
0045 RATIO = NUMN / DENNM1
0046 IF(ABS(REAL(RATIO) - 0.1) .GT. EPSR) GO TO 11 ! CHECK FOR CONVERGENCE
0048 IF(ABS(IMAG(RATIO)) .LT. EPSI) GO TO 500 ! STOP THE FRACTION
0050 11 QUO(L) = QUO(L) * RATIO ! CONTINUE THE FRACTION
0051 NUMNM1 = NUMN
0052 DENNM1 = DENN
0053 GO TO 10
0054 500 CONTINUE

C-----------------------------
C CALCULATE EACH ORDER
C
0055 501 DO 600 I = 1, M+1
0056 SBES = ONE
0057 N = I - 1
0058 IF(N.EQ.0) GO TO 510
0060 DO 502 J = 1, N
0061 502 SBES = SBES * ONE / QUO(J)
C-----------------------------
C FROM HERE TO 510 CHECK FOR UNDERFLOW
C
0062 IF(REAL(SBES) .EQ. ZERO) GO TO 512
0064 ISR = IFIX(ALOG10(ABS(REAL(SBES))))
0065 ISI = 0
0066 IF(XI .EQ. ZERO) GO TO 504
0068 503 ISI = IFIX(ALOG10(ABS(IMAG(SBES))))
0069 504 IF((ISR+ICSR).LT.-36.AND.(ISR+ICSR-IXR).LT.-36) GO TO 505
0071 GO TO 512 ! REAL UNDERFLOW
0072 505 IF((ISI+ICSI).LT.-36.AND.(ISI+ICSI-IXI).LT.-36) GO TO 510
0074 GO TO 512 ! IMAG. UNDERFLOW
C-----------------------------
C HERE COMES THE ANSWER
C
0075 510 SBES = SBES * CSX / X
0076 600 JJ(I) = SBES
0077 IER = 0 ! NORMAL RETURN
0078 RETURN
0079 512 IER = - N ! UNDERFLOW AT ORDER N
0080 RETURN
0081 513 IER = 2 ! OVERFLOW
0082 RETURN
0083 514 IER = 3 ! ORDER TOO BIG
0084 END
Physical Notes

The reflectivity of a fluid sphere generally increases as its density and sound speed contrasts increase. The reflectivity increases most dramatically when the values for the sphere are less than those for the medium.

For the elastic sphere, a shear speed which is very small with respect to the compressional speed of the sphere and the medium will lead to a reflectivity which is indistinguishable from the corresponding fluid case. Apparently the impedance contrast in this case is such that little or no energy is converted to shear.

The elastic sphere program produces strange results for targets with high shear speed. Many of these cases can be ruled out physically since the shear speed of a substance must be less than 0.7 x its compressional speed. For cases with moderate shear speeds, the results are similar to those for a fluid sphere but spikey. It is probably necessary to include the effects of attenuation in order to get valid results.

For cases with very small attenuation, the viscoelastic sphere program produces results which are indistinguishable from the corresponding elastic cases. A moderate amount of shear attenuation will smooth out the spikiness caused by a moderate shear speed. Moderate values of compressional attenuation will raise the level of the curve; large values will also decrease the depth of the dips.
Computational Notes

These programs and subroutines have explicit parameters that are used to test for convergence. In order to reduce execution times, these parameters have been set relatively high. The values were selected on the basis of the output plots, so the relative accuracy may be only about one percent.

The output plots are generated by means of an in-house developed plotting package. The output sections of the programs can be easily modified to work with other locally available plotting software.