S of

OCEANOGRAPHY

Backscattering Programs for Spherical Targets

by

Richard K. Johnson
Lawrence Flax
David Standley

Office of Naval Research
N00014-76-C-0067
NR 038-102

Reference 79-7 April 1979

OREGON STATE UNIVERSITY

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 79-7
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): BACKSCATTERING PROGRAMS FOR SPHERICAL TARGETS
5. TYPE OF REPORT & PERIOD COVERED:
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Richard K. Johnson, Lawrence Flax, David Standley
8. CONTRACT OR GRANT NUMBER(s): N00014-76-C-0067
9. PERFORMING ORGANIZATION NAME AND ADDRESS: School of Oceanography, Oregon State University, Corvallis, OR 97330
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 038-102
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, Ocean Science & Technology Division, Arlington, VA 22217
12. REPORT DATE: April 1979
13. NUMBER OF PAGES: 30
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY NOTES: Report dated April 1979
19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

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, ka. The reflectivity is defined as $R^2 = \frac{\sigma_b}{a^2/4}$, where σ_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

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

as dimensionless representations of target strength vs. frequency. The conversions are TS = 10 log ($R^2a^2/4$) and f = cka/2πa 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. Johnson[1]
Lawrence Flax[2]
David Standley[1]

Reference 79-7
April 1979

G. Ross Heath
Dean

[1]Oregon State University, School of Oceanography
[2]Naval 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 prinicipal outputs of the programs are plots of reflectivity, R^2, as a function of size-frequency, ka. The reflectivity is defined as

$$R^2 = \frac{\sigma_b}{a^2/4},$$

where σ_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

$$TS = 10 \log (R^2 a^2/4) \text{ and}$$

$$f = cka/2\pi a$$

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 constrast 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

Name	Symbol	Meaning
AB1	β_c	compressional attenuation in sphere (dB/wavelength)
AB2	β_s	shear attenuation in sphere (dB/wavelength)
DRATIO	g	density of sphere/density of medium
PRATIO	h	compressional speed in sphere/speed in medium
SRATIO	s	shear speed in sphere/speed in medium
Z	ka	size-frequency parameter

Output Parameters

G	R	reflectivity or form function
G2	R^2	reflectivity squared

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

Anderson, V.C., 1950. Sound scattering from a fluid sphere. J. Acoust. Soc. Am. 22, 426-431.

Johnson, R.K., 1977. Sound scattering from a fluid sphere revisted. J. Acoust. Soc. Am. 61, 375-377.

```
0001          PROGRAM SPHERF
      C*************************************************************************
      C SPHERF
      C     COMPUTES BACK SCATTERING FORM FUNCTION FOR A FLUID SPHERE
      C     WRITTEN BY LARRY FLAX,NRL
      C     MODIFIED BY R K JOHNSON AND D STANDLEY
      C
      C*************************************************************************
0002          DIMENSION ZZ(1000),IE(10),GG(1000)
0003          COMPLEX CI,TEMP,SUM,F
0004          DATA NO/2HN /,D/0.1/,EPS/0.0005/
      C---PLOT COMMON, ETC.
0005          COMMON  /PLT/P(14),IROT,ISIZE,NXCH,NYCH,XFMT,YFMT,XLAB,YLAB
0006          DIMENSION XFMT(2),YFMT(2),XLAB(10),YLAB(10),PLID(3)
0007          DIMENSION DATLBL(10),PRL(3),DRL(3)
0008          EQUIVALENCE (DATLBL,PLID(3))
0009          DATA PLID/'SPHE','RF  ','    '/
0010          DATA P/7.,8.,2.,2.,0.,0.,0.,0.,0.,1.,10.,1.,10./
0011          DATA XFMT/'(F4.','1)  ',/,YFMT/'(F5.','0)  '/,NXCH/4/,NYCH/5/
0012          DATA  PENUP/1/, PENDWN/0/
0013          DATA PRL/'H = ',2*'    '/, DRL/'G = ',2*'    '/
0014          DATA XLAB/' KA ',9*'    '/
0015          DATA YLAB/' R (','DB) ',8*'    '/
0016          INTEGER PENUP, PENDWN
      C-----------------------------------------
      C     GET ACOUSTIC PROPERTIES AND KA RANGE
      C
0017    1     CALL DATMSG
0018          CALL DTMSG(DATLBL)          !FOR PLOT
0019          TYPE 100
0020          ACCEPT 101, DRATIO, PRATIO  !G,H
0021    20    TYPE 102
0022          ACCEPT 101, ZFROM,ZTO,ZSTEP
0023          IEND=(ZTO-ZFROM)/ZSTEP+1.5
0024          IF(IEND .LE. 1000)GO TO 30
0026          TYPE 110
0027          GO TO 20
0028    30    TYPE 103
0029          ACCEPT 104, ITYP
0030          IF(ITYP.EQ.NO)GO TO 50
0032          TYPE 105
0033    50    CONTINUE
0034          CI=(0.,1.)
0035          G2LM=1000.                  !SET G2L MINIMUM ARBITRARILY HIGH
      C******  START LOOP
0036          DO 800 IZ=1,IEND
0037          Z=FLOAT(IZ-1)*ZSTEP+ZFROM
0038          ZL=Z/PRATIO
0039          TE1=0.
0040          TE2=0.
0041          TEMP=(0.,0.)
0042          CALL SBESJ(Z,0,BJ,D,IE(1))
0043          CALL SBESJ(ZL,0,BJL,D,IE(2))
0044          CALL SBESY(Z,0,BY,IE(3))
0045          DO 6 K=1,50
0046          L=K-1
0047          X=(2*L+1)*(-1)**L
0048          CALL SBESJ(Z,K,BJ1,D,IE(4)) !SPHERICAL BESSEL FUNCTIONS
0049          CALL SBESJ(ZL,K,BJL1,D,IE(5))
0050          CALL SBESY(Z,K,BY1,IE(6))
0051          DO 80 ICK=1,6
0052          IF(IE(ICK).EQ.0)GO TO 80
0054          TYPE 106,ICK,IE(ICK)
0055       80 IE(ICK) = 0
```

```
0056          BJP=BJ*L/Z-BJ1                !FIRST DERIVATIVES
0057          BJPL=BJL*L/ZL-BJL1
0058          BYP=BY*L/Z-BY1
0059          E2=ZL*BJPL/BJL
0060          AN=E2/DRATIO
0061          R=BJ*AN-Z*BJP
0062          S=BY*AN-Z*BYP
0063          U=R*R+S*S
0064          SUM=(X/U)*(CI*R*R-R*S)
0065          TEMP=SUM+TEMP
0066          T=REAL(TEMP)
0067          T1=AIMAG(TEMP)
0068          QE1=ABS((T-TE1)/T)
0069          QE2=ABS((T1-TE2)/T1)
        C.....INCLUDE MORE MODES UNTIL CHANGE IS LESS THAN EPS
0070          IF(QE1.LE.EPS.AND.QE2.LE.EPS)GO TO 13
0072          TE1=T
0073          TE2=T1
0074          BJ = BJ1                      !BESSEL OUTPUT FOR NEXT ITERATION
0075          BJL = BJL1
0076          BY = BY1
0077    6     CONTINUE
0078    13    F = 2. * TEMP / Z             !FORM FUNCTION
0079          F1=REAL(F)
0080          F2=AIMAG(F)
        C.....MODULUS=SQRT(BACKSCATTERING/GEOMETRIC CROSS-SECTION)
0081          G2=F1*F1+F2*F2
0082          G=SQRT(G2)
0083          ZZ(IZ)=ALOG10(Z)
0084          G2L=10.*ALOG10(G2)
0085          IF(ITYP.NE.NO)TYPE 107,Z,L,G,G2L
0087          GG(IZ)=G2L
0088          IF(G2L.LT.G2LM)G2LM=G2L       !SEARCH FOR MINIMUM VALUE OF 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,ZLOGE)))
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                     !YMIN
0112          P(8) = YS + 80.               !YMAX
0113          P(9) = P(5)                   !XO
0114          P(10) = YS                    !YO
0115          CALL AXIS(3)                  !DRAW AXES
0116          ENCODE(6,109,DRL(2))DRATIO
0117          ENCODE(6,109,PRL(2))PRATIO
0118          CALL PLOTXY(P(5),P(8),PENUP,0)              !GO TO (XMIN,YMAX)
0119          CALL PLOT(5,IX,IY)
0120          IX = IX + 200
```

```
0121          CALL STRING(IX,IY,28,PLID,0,2)                !LABEL THE PLOT
0122          CALL STRING(IX,IY-60,10,DRL,0,2)
0123          CALL STRING(IX,IY-120,10,PRL,0,2)
0124          CALL PLOTXY(P(5),P(7),PENUP,0)
0125          DO 600 I=1, IEND
0126           YY = GG(I)
0127           XXX = ZZ(I)
0128           IF(I .EQ. 1)CALL PLOTXY(XXX, YY, PENUP, 0) !GO TO FIRST POINT
0130  600     CALL PLOTXY(XXX, YY, PENDWN, 0)
0131          CALL PLWAIT
0132          CALL PLOT(2,1885)
0133          CALL PLOT(3)
0134          CALL PLWAIT
0135          GO TO 1
0136  100     FORMAT(' ENTER DRATIO, PRATIO ... [G,H] : ',$)
0137   101    FORMAT (3F10.4)
0138   102    FORMAT(' ENTER ZFROM, ZTO, ZSTEP : ',$)
0139   103    FORMAT(' TYPE RESULTS?(Y/N) ',$)
0140   104    FORMAT(A2)
0141   105    FORMAT (//'       KA       MODE       MODULUS       20LOG '/)
0142   106    FORMAT(' REQ PREC NOT ACHIEVED. ROUTINE #',I2,' ER=',I2)
0143  107     FORMAT(F10.3,I8,F14.4,F11.1)
0144   108    FORMAT('0PLOT:GIVE START OF Y SCALE   (MAX VALUE=',F5.1,') : ',$)
0145  109     FORMAT(F6.3)
0146  110     FORMAT(' TOO MANY INCREMENTS ... TRY AGAIN'//)
0147          END
*
```

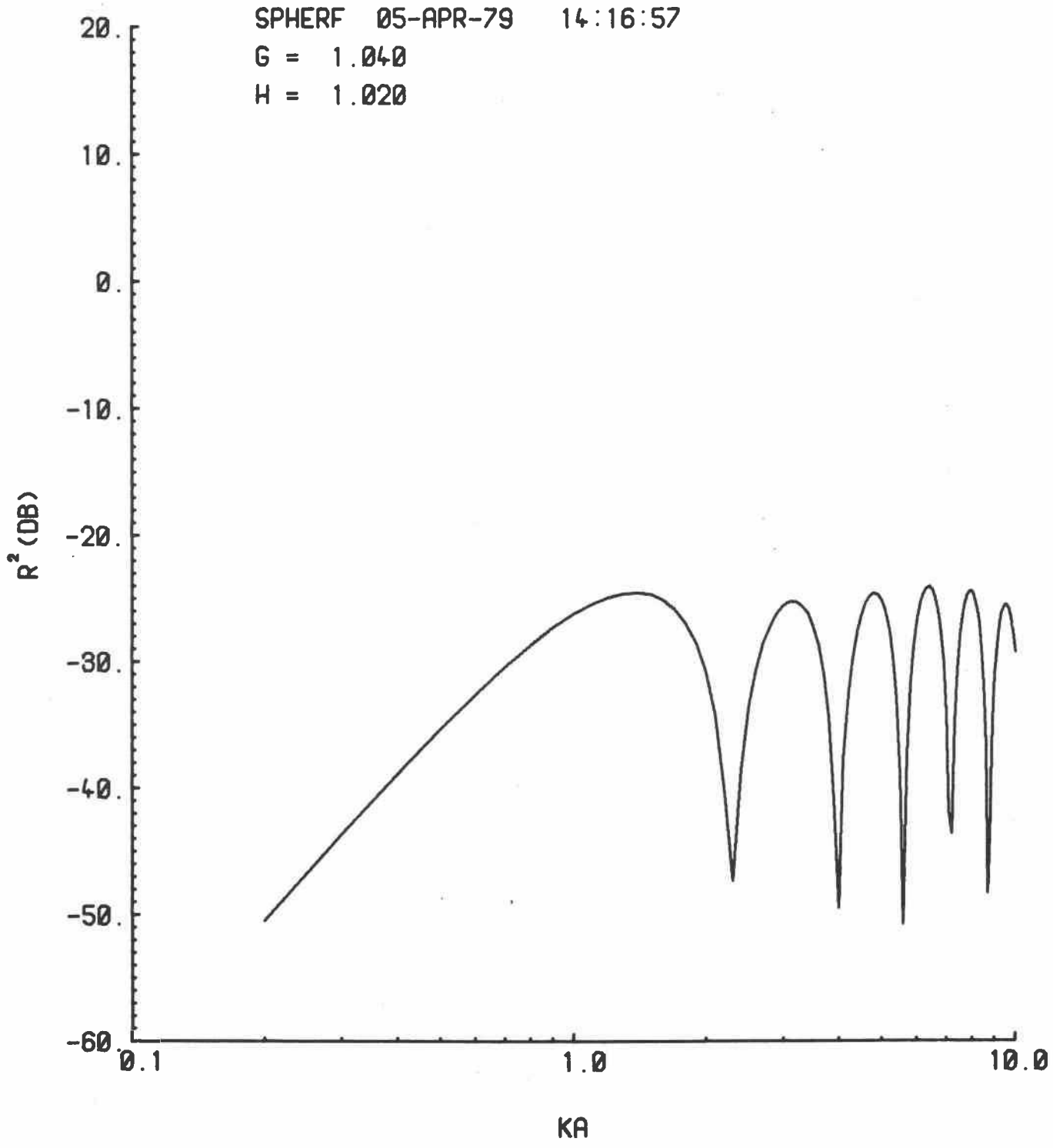

Fig. 1 Output plot from SPHERF.

Elastic Sphere

The program SPHERE calculates R^2 for a elastic sphere in a fluid medium.

The original version of this program was written by Larry Flax.

Bibliography

Neubauer, W.G., Vogt, R.H., and L.R. Dragonette, 1974. Acoustic reflection from elastic spheres. I. Steady-state signals. J. Acoust. Soc. Am. 55, 1123-1129.

Faran Jr., J.J., 1951. Sound Scattering by solid cylinders and spheres. J. Acoust. Soc. Am. 23, 405.

Hampton, L.D. and C.M. McKinney, 1961. Experimental study of the scattering of acoustic energy from solid metal spheres in water. J. Acoust. Soc. Am. 33, 694.

Hickling, R., 1962. Analysis of echoes from solid elastic sphere in water. J. Acoust. Soc. Am. 34, 1582.

Rudgers, A.J., 1967. Techniques for numerically evaluating the formulas describing monostatic reflections of acoustic waves by elastic spheres. NRL Rep. 6551

```
0001          PROGRAM SPHERE
      C*************************************************************************
      C SPHERE
      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
      C*************************************************************************
0002          DIMENSION ZZ(500),IE(10),GG(500),A(3,3)
0003          COMPLEX CI,TEMP,SUM,F
0004          REAL  LM1, LP1
0005          DATA NO/2HN /,D/0.1/,EPS/0.001/
      C---PLOT COMMON, ETC.
0006          COMMON  /PLT/P(14),IROT,ISIZE,NXCH,NYCH,XFMT,YFMT,XLAB,YLAB
0007          DIMENSION XFMT(2),YFMT(2),XLAB(10),YLAB(10),PLID(3)
0008          DIMENSION DATLBL(10),PRL(3),DRL(3),SRL(3)
0009          EQUIVALENCE (DATLBL,PLID(3))
0010          DATA PLID/'SPHE','RE  ','    '/
0011          DATA P/7.,8.,2.,2.,0.,0.,0.,0.,0.,0.,1.,10.,1.,10./
0012          DATA XFMT/'(F4.','1)  ',/,YFMT/'(F5.','0)  '/,NXCH/4/,NYCH/5/
0013          DATA  PENUP/1/, PENDWN/0/
0014          DATA PRL/'H = ',2*'    '/, DRL/'G = ',2*'    '/
0015          DATA XLAB/' KA ',9*'    '/,SRL/'S = ',2*'    '/
0016          DATA YLAB/' R ('DB) ',8*'    '/
0017          INTEGER PENUP, PENDWN
      C-----------------------------------------
      C     GET ACOUSTIC PROPERTIES AND KA RANGE
      C
0018    1     CALL DATMSG
0019          CALL DTMSG(DATLBL)                    !FOR PLOT
0020          TYPE 100
0021          ACCEPT 101,DRATIO,PRATIO,SRATIO       !G,H,S
0022    20    TYPE 102
0023          ACCEPT 101, ZFROM,ZTO,ZSTEP
0024          IEND=(ZTO-ZFROM)/ZSTEP+1.5
0025          IF(IEND .LE. 500)GO TO 30
0027          TYPE 110
0028          GO TO 20
0029    30    TYPE 103
0030          ACCEPT 104, ITYP
0031          IF(ITYP.EQ.NO)GO TO 40
0033          TYPE 105
0034    40    C=SRATIO**2/(PRATIO**2-2.*SRATIO**2)  !POISSON'S RATIO
0035    50    CONTINUE
0036          CI=(0.,1.)
0037          G2LM=1000.                    !SET G2L MINIMUM ARBITRARILY HIGH
      C******  START LOOP
0038          DO 800 IZ=1,IEND
0039          Z=FLOAT(IZ-1)*ZSTEP+ZFROM
0040          ZL=Z/PRATIO
0041          ZS=Z/SRATIO
0042          TE1=0.
0043          TE2=0.
0044          TEMP=(0.,0.)
0045          CALL SBESJ(Z,0,BJ,D,IE(1))
0046          CALL SBESJ(ZL,0,BJL,D,IE(2))
0047          CALL SBESY(Z,0,BY,IE(3))
0048          CALL SBESJ(ZS,0,BJS,D,IE(4))
0049          DO 6 K=1,50
0050          L=K-1
0051          X=(2*L+1)*(1-2*MOD(L,2))
0052          CALL SBESJ(Z,K,BJ1,D,IE(5)) !SPHERICAL BESSEL FUNCTIONS
0053          CALL SBESJ(ZL,K,BJL1,D,IE(6))
0054          CALL SBESJ(ZS,K,BJS1,D,IE(7))
```

```
0055          CALL SBESY(Z,K,BY1,IE(8))
0056          DO 80 ICK=1,8
0057          IF(IE(ICK).EQ.0)GO TO 80
0059          TYPE 106,ICK,IE(ICK)
0060       80 IE(ICK) = 0
0061          BJP=BJ*L/Z-BJ1                    !FIRST DERIVATIVES
0062          BJPL=BJL*L/ZL-BJL1
0063          BJPS=BJS*L/ZS-BJS1
0064          BYP=BY*L/Z-BY1
0065          ZS2=ZS*ZS                         !SECOND DERIVATIVES
0066          LM1=FLOAT(L*(L-1))
0067          LP1=FLOAT(L*(L+1))
0068          BJPPL=(LM1/(ZL*ZL)-1.)*BJL+2.*BJL1/ZL
0069          BJPPS=(LM1/ZS2-1.)*BJS+2.*BJS1/ZS
0070          A(1,1)=(BJL-2.*C*BJPPL)/(1.+2.*C)
0071          A(1,3)=-2.*LP1*(ZS*BJPS-BJS)/ZS2
0072          A(2,1)=ZL*BJPL
0073          A(2,3)=LP1*BJS
0074          A(3,1)=2.*(ZL*BJPL-BJL)
0075          A(3,3)=ZS2*BJPPS+FLOAT((L+2)*(L-1))*BJS
0076          E1=A(2,1)*A(3,3)-A(3,1)*A(2,3)
0077          E=A(1,1)*A(3,3)-A(1,3)*A(3,1)
0078          E2=E1/E
0079          AN=E2/DRATIO
0080          R=BJ*AN-Z*BJP
0081          S=BY*AN-Z*BYP
0082          U=R*R+S*S
0083          SUM=(X/U)*(CI*R*R-R*S)
0084          TEMP=SUM+TEMP
0085          T=REAL(TEMP)
0086          T1=AIMAG(TEMP)
0087          QE1=ABS((T-TE1)/T)
0088          QE2=ABS((T1-TE2)/T1)
      C.....INCLUDE MORE MODES UNTIL CHANGE IS LESS THAN EPS
0089          IF(QE1.LE.EPS.AND.QE2.LE.EPS)GO TO 13
0091          TE1=T
0092          TE2=T1
0093          BJ = BJ1                          !BESSEL OUTPUT FOR NEXT ITERATION
0094          BJL = BJL1
0095          BJS = BJS1
0096          BY = BY1
0097    6     CONTINUE
0098    13    F = 2. * TEMP / Z                 !FORM FUNCTION
0099          F1=REAL(F)
0100          F2=AIMAG(F)
      C.....MODULUS=SQRT(BACKSCATTERING/GEOMETRIC CROSS-SECTION)
0101          G2=F1*F1+F2*F2
0102          G=SQRT(G2)
0103          ZZ(IZ)=ALOG10(Z)
0104          G2L=10.*ALOG10(G2)
0105          IF(ITYP.NE.NO)TYPE 107,Z,L,G,G2L
0107          GG(IZ)=G2L
0108          IF(G2L.LT.G2LM)G2LM=G2L            !SEARCH FOR MINIMUM VALUE OF G2L
0110      800 CONTINUE
      C-------------------------
      C       PLOTTING ROUTINE:
      C
0111          TYPE 108,G2LM
0112          ACCEPT 101, YS
0113          IF(YS.GE.0)GO TO 1
0115          ZLOGS=ALOG10(ZFROM)
0116          ZLOGE=ALOG10(ZTO)
0117          MIN =  INT((SIGN(ABS(ZLOGS)+0.96,ZLOGS)))
0118          IF(MIN .GT. 0)MIN = MIN - 1
```

```
0120          XMIN = 10. ** MIN
0121          MAX =  INT((SIGN(ABS(ZLOGE)+0.96,ZLOGE)))
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 + 80.                 !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, PENDWN, 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) ',$)
0162    104   FORMAT(A2)
0163    105   FORMAT (/'       KA       MODE       MODULUS       20LOG '/)
0164    106   FORMAT(' REQ PREC NOT ACHIEVED. ROUTINE #',I2,' ER=',I2)
0165   107    FORMAT(F10.3,I8,F14.4,F11.1)
0166    108   FORMAT('0PLOT:GIVE START OF Y SCALE   (MAX VALUE=',F5.1,') : ',$)
0167   109    FORMAT(F6.3)
0168   110    FORMAT(' TOO MANY INCREMENTS ... TRY AGAIN'//)
0169          END
*
```

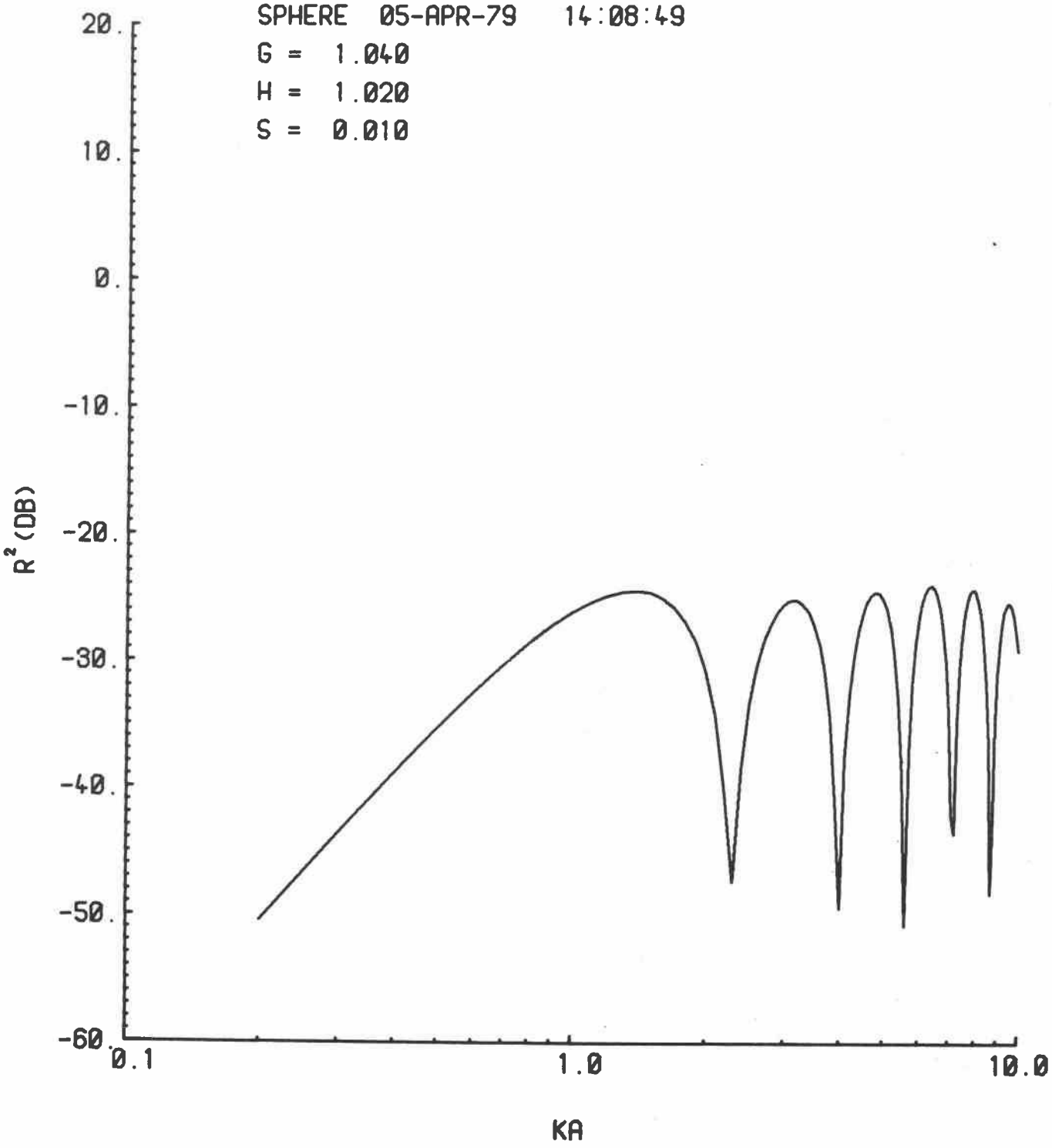

Fig. 2 Output plot from SPHERE for a case with very low shear speed.

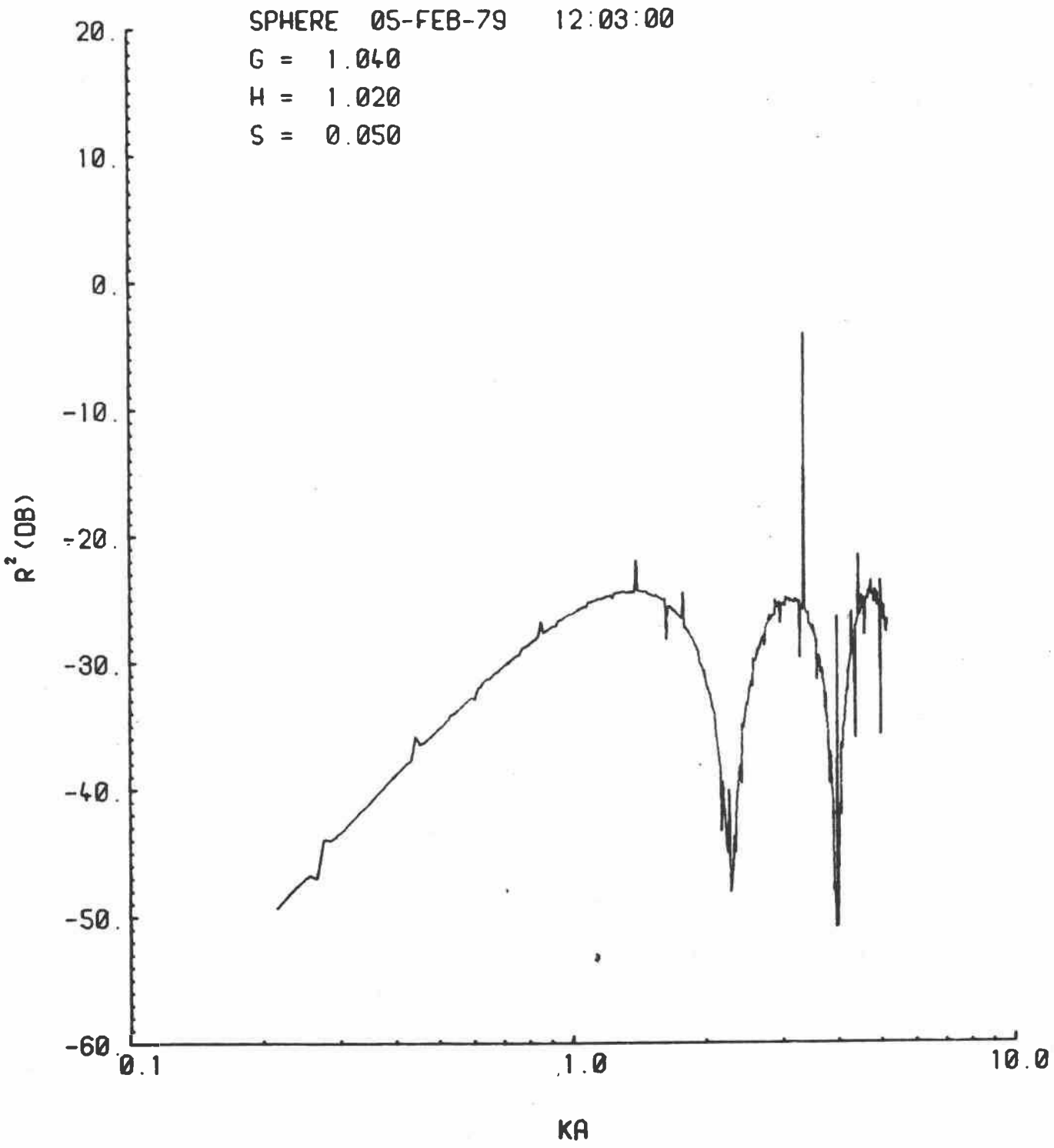

Fig. 3 Output plot from SPHERE for a case with moderate shear speed.

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

Davis, C.M., Dragonette, L.R., and L. Flax, 1978. Acoustic scattering from silicone rubber cylinders and spheres. J. Acoust. Soc. Am. 63, 1694-1698.

Hasegawa, T., Kitagawa, Y., and Y. Watanabe, 1977. Sound reflection from an absorbing sphere. J. Acoust. Soc. Am., 62, 1298-1300.

Vogt, R.H., Flax, L., Dragonette, L.R., and W.G. Neubauer,1975. Monostatic reflection of a plane wave from an absorbing sphere. J. Acoust. Soc. Am. 57, 558-561.

```
0001          PROGRAM ABSPHR
      C*****************************************************************
      C ABSPHR
      C
      C  ACOUSTIC BACK SCATTERING FROM AN ABSORBING SPHERE:
      C    REFLECTION FORM FUNCTION VS KA
      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  ABSORBING SPHERE. J.ACOUST. SOC. AM.62, 1298-1300. 1977.
      C
      C  MODIFIED BY R K JOHNSON  AND  D STANDLEY
      C*****************************************************************
0002          COMPLEX X1,X2,JN,JB,A,B,G,D,C,E,H,F,HN,PH,CN
0003          COMPLEX D1C,D2C,TC,P5C,DENSC,X3
0004          REAL K1A, K2A
0005          DIMENSION XX(500),YP(500),JN(50),JB(50),SJ(50)
0006          DIMENSION SN(50),ALP(50),BET(50),DJN(50),DJB(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 DATLBL(10),AB1L(3),AB2L(3),PRL(3),DRL(3),SRL(3)
0011          EQUIVALENCE (DATLBL,PLID(3))
0012          DATA PLID/'ABSP','HR  ','    '/
0013          DATA P/7.,8.,2.,2.,0.,0.,0.,0.,0.,1.,10.,1.,10./
0014          DATA XFMT/'(F4.','1)  ',/,YFMT/'(F5.','0)  '/,NXCH/4/,NYCH/5/
0015          DATA  PENUP/1/, PENDWN/0/
0016          DATA AB1L(1)/'AB1='/,AB2L(1)/'AB2='/,DRL(1)/'G = '/
0017          DATA PRL(1)/'H = '/,SRL(1)/'S = '/
0018          DATA XLAB/' KA ',9*'    '/
0019          DATA YLAB/' R (','DB) ',8*'    '/
0020          INTEGER PENUP, PENDWN
0021    1     CALL DATMSG
0022          CALL DTMSG(DATLBL)  !FOR PLOT
      C-------------------------------
      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
      C
0023          TYPE 104
0024          ACCEPT 105, IOPT
0025          IF(IOPT .EQ. 3)GO TO 410
0027    2     TYPE 902
0028          ACCEPT 100,AB1,AB2
0029          TYPE 906
0030          ACCEPT 100,DRATIO,PRATIO,SRATIO   !G,H,S
0031          CC1 = 1. / PRATIO
0032          CC2 = 1. / SRATIO
0033    9     TYPE 907
0034          ACCEPT 100, ZFROM, ZTO, ZSTEP
0035          IF(ZSTEP .EQ. 0.0)ZSTEP = 0.25
0037          IMAX = IFIX( (ZTO - ZFROM) / ZSTEP + 1.0001)
0038          IF(IMAX .LE. 500)GO TO 10
0040          TYPE 114
0041          GO TO 9
0042    10    YM = 1000.                        !LARGE MINIMUM
0043          IHDR = 0
0044          ET = SECNDS(0.)                   !ELAPSED TIME
      C****** START LOOP
0045          DO 30 I=1,IMAX
0046          XX(I) = ZFROM + ZSTEP * FLOAT(I-1)
```

```
0047          X = XX(I)                              !X IS KA
0048          K1A = CC1 * X                          !COMPRESSIONAL
0049          K2A = CC2 * X                          !SHEAR
0050          AL1 = -AB1 * K1A / PX
0051          AL2 = -AB2 * K2A / PX
0052          X1 = CMPLX(K1A,AL1)                    !COMPLEX COM. WAVE NUMBER
0053          X2 = CMPLX(K2A,AL2)                    !COMPLEX SHEAR WAVE NUMBER
0054          NM = IFIX(X + 5.0)
0055          IF(X .LE. 3.41)NM = 7
0057          IF(X .LE. 2.20)NM = 6
0059          IF(X .LE. 1.83)NM = 5
0061          IF(X .LE. 1.12)NM = 4
0063          IF(X .LE. 0.5)NM = 3
0065          NNM = NM+ 3                             !NUMBER OF MODES TO CALCULATE
0066          IF(AIMAG(X1) .EQ. 0.)GO TO 21          !FASTER IF NOT CMPLX
0068          CALL CSBSJ(X1, JN, NNM, IER)
0069          IF(IER .EQ. 0)GO TO 3
0071          TYPE 905, IER, NNM, X1
0072          GO TO 3
0073    21    CALL SJBES(NNM, REAL(X1), DJN)
0074          DO 211 IC = 1, NNM
0075    211   JN(IC) = CMPLX(DJN(IC), 0.)
0076    3     IF(AIMAG(X2) .EQ. 0.)GO TO 22
0078          CALL CSBSJ(X2, JB, NNM, IER)
0079          IF(IER .EQ. 0)GO TO 4
0081          TYPE 905, IER, NNM, X2
0082          GO TO 4
0083    22    CALL SJBES(NNM, REAL(X2), DJB)
0084          DO 221 IC= 1, NNM
0085    221   JB(IC) = CMPLX(DJB(IC), 0.)
0086    4     CALL SJBES(NNM, X, SJ)
0087          CALL SNBES(NNM, X, SN)
0088    25    PJJ = 0.
0089          PNN = 0.
0090          DO 20 N=1,NM
0091            T = FLOAT(N - 1)
0092            N1 = N
0093            N2 = N + 1
0094            TC = CMPLX(T,0.)
0095            A = TC * JN(N1) - X1 * JN(N2)
0096            B = A - JN(N1)
0097            D1C = CMPLX(1.,0.)
0098            D2C = CMPLX(2.,0.)
0099            G = (X2**2 / D2C - TC * (TC - D1C)) * JN(N1) - D2C * X1 * JN(N2)
0100            A = A / B
0101            D = G / B
0102            C = D2C * TC * (TC + D1C) * JB(N1)
0103            E = (D2C * TC * TC - X2**2 - D2C) * JB(N1) + D2C * X2 * JB(N2)
0104            H =D2C* TC* (TC+D1C) * ((D1C-TC) * JB(N1) + X2 * JB(N2))
0105            B = C / E
0106            E = H / E
0107            P5C = CMPLX(.5,0.)
0108            DENSC = CMPLX(DRATIO,0.)
0109            F = P5C* (A - B) * X2**2 /((D - E) *DENSC)
0110            PJ = T * SJ(N1) - X * SJ(N2)
0111            PN = T * SN(N1) - X * SN(N2)
0112            HN = CMPLX(SJ(N1), -SN(N1))
0113            PH = CMPLX(PJ, -PN)
0114            CN = (CMPLX(PJ,0.)-F*CMPLX(SJ(N1),0.))/(F*HN-PH)
0115            ALP(N1) = REAL(CN)
0116            BET(N1) = AIMAG(CN)
0117            N5 = N - 1
0118            YN = (2. * FLOAT(N5) + 1.) * (-1.)**N5
0119            PJJ = PJJ + YN * BET(N)             !ADD UP REAL AND IMAGINARY MODES
```

```
0120          PNN = PNN + YN * ALP(N)
      C.....TEST FOR CONVERGENCE
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(NM+2 .GT. NNM)GO TO 200
0129          NM = NM + 1
0130          GO TO 25
0131   200    TYPE 113,X,NNM,NM
0132   201    YP(I) = 2.0 * SQRT(PJJ**2 + PNN**2) / X !CONVERT TO FORM FUNCTION
0133          YPL = 20. * ALOG10(YP(I))
0134          IF(YPL .LT. YM)YM = YPL
      C.....TYPEOUT OPTION
0136          ISW = IPEEK('177570)                    !READ CONSOLE SWITCHES
0137          IF(ISW .LT. 0)GO TO 30                  !NORMAL
0139          IF(ISW .EQ. 0)GO TO 291                 !EXIT LOOP IF SWITCHES=0
0141          IF(IHDR .EQ. 0)TYPE 110
0143          IHDR = 1                                !TYPEOUT IF SWITCHES +
0144          DT = SECNDS(ET)
0145          TYPE 111, X, YP(I), 20.*ALOG10(YP(I)), NNM,NM, X1, X2,  DT
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)),I=1,IMAX)
      C-------------------------
      C  CHECK OPTIONS
      C
0152          GO TO(400,450,410),IOPT
0153   400    TYPE 106                    !OUTPUT FILE
0154          CALL ASSIGN(1,,-1,'NEW')
0155          WRITE(1,107)AB1,AB2,DRATIO,PRATIO,SRATIO,YM,IMAX
0156          WRITE(1,108)(XX(I),YP(I),I=1,IMAX)
0157          CALL CLOSE(1)
0158          CALL EXIT
0159   410    TYPE 112                    !INPUT FILE
0160          CALL ASSIGN(1,,-1,'RDO')
0161          READ(1,107)AB1,AB2,DRATIO,PRATIO,SRATIO,YM,IMAX
0162          READ(1,108)(XX(I),YP(I),I=1,IMAX)
0163          CALL CLOSE(1)
0164          ZFROM = XX(1)
0165          ZTO = XX(IMAX)
      C-------------------------
      C     PLOTTING ROUTINE:
      C
0166   450    TYPE 109,YM
0167          ACCEPT 904, YS
0168          ZLOGS=ALOG10(ZFROM)
0169          ZLOGE=ALOG10(ZTO)
0170          MIN =  INT((SIGN(ABS(ZLOGS)+0.96,ZLOGS)))
0171          IF(MIN .GT. 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.                   !X LENGTH
0179          P(2) = 8.                   !Y LENGTH
0180          P(3) = 2.
0181          P(4) = 2.
0182          P(5) = XMIN
0183          P(6) = XMAX
```

```
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,AB1L(2))AB1
0190          ENCODE(6,103,AB2L(2))AB2
0191          ENCODE(6,103,DRL(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(5,IX,IY)
0196          IX = IX + 200
0197          CALL STRING(IX,IY,28,PLID,0,2)                !LABEL THE PLOT
0198          CALL STRING(IX,IY-60,10,AB1L,0,2)
0199          CALL STRING(IX,IY-120,10,AB2L,0,2)
0200          CALL STRING(IX,IY-180,10,DRL,0,2)
0201          CALL STRING(IX,IY-240,10,PRL,0,2)
0202          CALL STRING(IX,IY-300,10,SRL,0,2)
0203          CALL PLOTXY(P(5),P(7),PENUP,0)
0204          DO 600 I=1, IMAX                              !PLOT THE POINTS
0205           YY = 20. * ALOG10(YP(I))
0206           XXX = ALOG10(XX(I))
0207           IF(I .EQ. 1)CALL PLOTXY(XXX, YY, PENUP, 0) !GO TO FIRST POINT
0209  600     CALL PLOTXY(XXX, YY, PENDWN, 0)
0210          CALL PLWAIT
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,'20LOG'),/,3(2X,F7.3,F8.4,2X,F7.1))
0217  103     FORMAT(F6.3)
0218  104     FORMAT(' OPTION: 1)CALC&SAVE RESULTS, 2)CALC&PLOT, 3)READ ',
             1     'RESULTS&PLOT : ',$)
0219  105     FORMAT(I2)
0220  106     FORMAT('0OUTPUT FILE ',$)
0221  107     FORMAT(6E15.5,I6)
0222  108     FORMAT(6E15.5)
0223   109    FORMAT('0PLOT:GIVE START OF Y SCALE   (MAX VALUE=',F6.1,') :',$)
0224  110     FORMAT(/3X,'KA',6X,'YP',4X,'20LOG',3X,'MODE',5X,'X1R',7X,'X1I',
             1   7X,'X2R',7X,'X2I',5X,'SECONDS'/)
0225  111     FORMAT(F7.3,F8.4,F7.1,I4,I3,2X,4(1PE10.3),2X,0PF4.0)
0226  112     FORMAT(' INPUT FILE ',$)
0227  113     FORMAT(' YP DID NOT CONVERGE AT X = ',F5.2,
             1    '  NNM = ',I3,'  NM = ',I3)
0228  114     FORMAT(' TOO MANY INCREMENTS ... TRY AGAIN')
0229  902     FORMAT(' ENTER AB1, AB2 : ',$)
0230  904     FORMAT(2F10.5)
0231  905     FORMAT('0CSBSJ ERROR #',I4,2X,'MODE=',I3,2X,2E16.7)
0232  906     FORMAT(' ENTER DRATIO, PRATIO, SRATIO ... [G,H,S] : ',$)
0233  907     FORMAT(' ENTER ZFROM, ZTO, ZSTEP : ',$)
0234          END
*
```

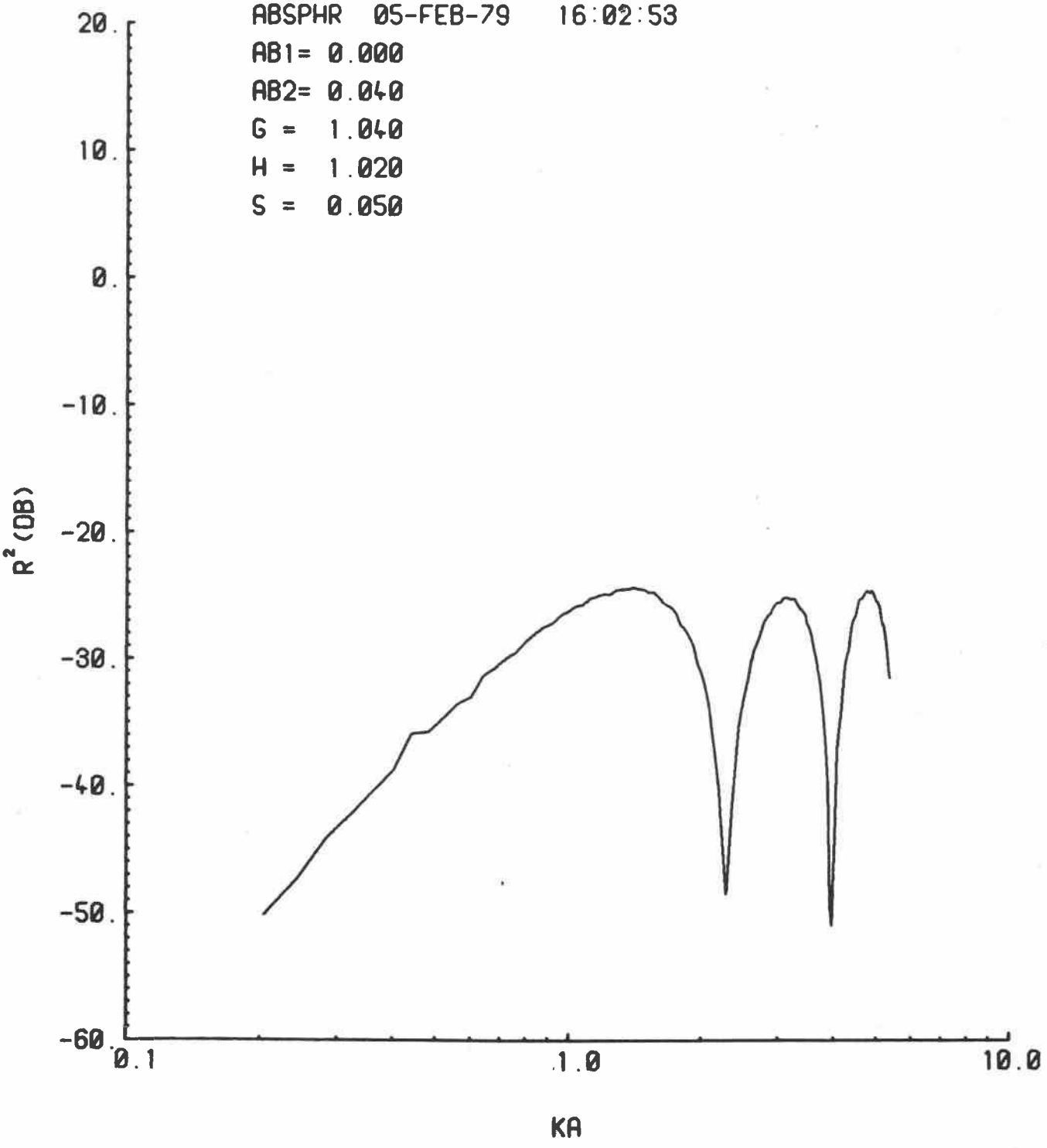

Fig. 4 Output from ABSPHR for a case with shear attenuation.

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

$$j_0(x) = \sin(x)/x$$
$$j_1(x) = \sin(x)/x^2 - \cos(x)/x. \qquad (1)$$

For higher orders and small arguments, the procedure uses the series approximations

$$j_n(x) = \frac{x^n}{1 \cdot 3 \cdot 5 \cdots (2n + 1)} \cdot \qquad (2)$$

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

$$j_{n+1}(x) = \frac{2n + 1}{x} j_n(x) - j_{n-1}(x). \qquad (3)$$

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

$j_o(x)/\hat{j}_o(x)$, where $j_o(x)$ is determined from equation (1). The correct value $j_n(x)$ is the $P*\hat{j}_n(x)$. 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

Abramowitz, M. and I.A. Stegun, 1970. Handbook of Mathematical Functions (U.S. Government Printing Office) 435-478.

```
      C
      C     ******************************************************************
      C
      C        SUBROUTINE SBESJ
      C
      C        PURPOSE
      C           COMPUTE THE J SPHERICAL BESSEL FUNCTION FOR A GIVEN ARGUMENT
      C        AND ORDER
      C
      C        USAGE
      C           CALL SBESJ(X,N,BJ,D,IER)
      C
      C        DESCRIPTION OF PARAMETERS
      C           X  -THE ARGUMENT OF THE J BESSEL FUNCTION DESIRED
      C           N  -THE ORDER OF THE J BESSEL FUNCTION DESIRED
      C           BJ -THE RESULTANT J BESSEL FUNCTION
      C           D  -REQUIRED ACCURACY
      C           IER-RESULTANT ERROR CODE WHERE
      C              IER=0  NO ERROR
      C              IER=1  N IS NEGATIVE
      C              IER=2  X IS NEGATIVE OR ZERO
      C              IER=3  REQUIRED ACCURACY NOT OBTAINED
      C              IER=4  RANGE OF N COMPARED TO X NOT CORRECT (SEE REMARKS)
      C
      C        REMARKS
      C           N MUST BE GREATER THAN OR EQUAL TO ZERO, BUT IT MUST BE
      C           LESS THAN
      C              20+10*X-X** 2/3   FOR X LESS THAN OR EQUAL TO 15
      C              90+X/2            FOR X GREATER THAN 15
      C
      C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
      C           NONE
      C
      C        METHOD
      C           RECURRENCE RELATION TECHNIQUE DESCRIBED BY H. A. ANTOSIEWICZ
      C       IN HANDBOOK OF MATHEMATICAL FUNCTIONS,PP.438-439
      C       AND 452-453.THIS CODE PATTERNED AFTER THAT FOR
      C       BESJ.
      C
      C     ******************************************************************
      C
0001        SUBROUTINE SBESJ(X,N,BJ,D,IER)
      C
0002        BJ=.0
0003        IF(N)10,20,20
0004     10 IER=1
0005        RETURN
0006     20 IF(X)30,30,31
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
      C
      C     IF N IS 0 OR 1, COMPUTE DIRECTLY.
      C
0019        IF (N-1) 42,43,49
```

```
0020      42 BJ=SIN(X)/X
0021         RETURN
0022      43 BJ=SIN(X)/(X*X)-COS(X)/X
0023         RETURN
      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
      C
0025      50       IF(X-FLOAT(2*N))60,210,210
      C
      C      IF X IS MIDDLIN, USE DECREASING RECURRENCE
      C
0026      60 MA=X+8.
0027         MB=N+IFIX(X)/4+2
0028         MZERO=MAXO(MA,MB)
      C
      C      SET UPPER LIMIT OF M
      C
0029         MMAX=NTEST
0030          DO 190 M=MZERO,MMAX,3
0031         FM1=1.
0032         FM=.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         FM1=BMK
0038         IF(MK-N-1)160,140,160
0039     140 BJ=BMK
0040     160      CONTINUE
      C
      C      SCALE FACTOR
      C
0041         P=SIN(X)/(X*BMK)
0042         BJ=P*BJ
0043         IF(ABS(BJ-BPREV)-ABS(D*BJ))200,200,190
0044     190 BPREV=BJ
0045         IER=3
0046     200 RETURN
      C
      C      INCREASING RECURRENCE
      C
0047     210 BJA=SIN(X)/X
0048         BJB=SIN(X)/(X*X)-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 BJ=BJC
0058         RETURN
      C
      C      ASCENDING SERIES:
0059     300 BJ=1.0
0060         DO 350 IFAC=1,N
0061         BJ=BJ*X/(FLOAT(2*IFAC+1))
0062     350 CONTINUE
0063         RETURN
0064         END
```

```
      C
      C     ******************************************************************
      C
      C        SUBROUTINE SBESY
      C
      C        PURPOSE
      C           COMPUTE THE Y SPHERICAL BESSEL FUNCTION FOR A GIVEN
      C        ARGUMENT AND ORDER
      C
      C        USAGE
      C           CALL BESY(X,N,BY,IER)
      C
      C        DESCRIPTION OF PARAMETERS
      C           X  -THE ARGUMENT OF THE Y BESSEL FUNCTION DESIRED
      C           N  -THE ORDER OF THE Y BESSEL FUNCTION DESIRED
      C           BY -THE RESULTANT Y BESSEL FUNCTION
      C           IER-RESULTANT ERROR CODE WHERE
      C              IER=0  NO ERROR
      C              IER=1  N IS NEGATIVE
      C              IER=2  X IS NEGATIVE OR ZERO
      C              IER=3  BY HAS EXCEEDED MAGNITUDE OF 10**36
      C
      C        REMARKS
      C           X MUST BE GREATER THAN ZERO
      C           N MUST BE GREATER THAN OR EQUAL TO ZERO
      C
      C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
      C           NONE
      C
      C        METHOD
      C           RECURRENCE RELATION AND TRIG FORMULAE AS DESCRIBED
      C          BY H.A. ANTOSIEWICZ IN HANDBOOK OF MATHEMATICAL
      C          FUNCTIONS, PP. 438-439. THIS CODE PATTERNED AFTER
      C             THAT FOR BESY.
      C
      C     ******************************************************************
      C
0001        SUBROUTINE SBESY(X,N,BY,IER)
      C
      C     CHECK FOR ERRORS IN N AND X
      C
0002        IF(N)180,10,10
0003     10 IER=0
0004        IF(X)190,190,20
      C
      C     EVALUATE Y0 AND Y1.
      C
0005     20 Y0=-COS(X)/X
0006        Y1=-COS(X)/(X*X)-SIN(X)/X
      C
      C     CHECK IF ONLY Y0 OR Y1 IS DESIRED
      C
0007     90 IF(N-1)100,100,130
      C
      C     RETURN EITHER Y0 OR Y1 AS REQUIRED
      C
0008    100 IF(N)110,120,110
0009    110 BY=Y1
0010        GO TO 170
0011    120 BY=Y0
0012        GO TO 170
      C
      C    PERFORM RECURRENCE OPERATIONS TO FIND YN(X)
      C
```

```
0013     130 YA=Y0
0014         YB=Y1
0015         K=1
0016     140 T=FLOAT(2*K+1)/X
0017         YC=T*YB-YA
0018         IF(ABS(YC)-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

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

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

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

$$\frac{j_i(Z)}{j_{i-1}(Z)} = \frac{j_{i+1/2}(Z)}{j_{i-1/2}(Z)} \qquad (2)$$

by definition. From Abramowitz and Stegun[2],

$$\frac{j_i(Z)}{j_{i-1}(Z)} = \frac{1}{a_0 -} \frac{1}{a_1 -} \frac{1}{a_2 -} \ldots \qquad (3)$$

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

The inverse of this ratio can be rewritten as

$$\frac{j_{i-1}(Z)}{j_i(Z)} = \frac{N_0 * N_1 * N_2 * \ldots}{D_0 * D_1 * \ldots} \qquad (4)$$

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 $N_k/D_{k-1} - 1 < \varepsilon$, where ε is the allowable error.

This program was written by David Standley.

Bibliography

Abramowitz, M. and I.A. Stegun, 1970. Handbook of Mathematical Functions (U.S. Government Printing Office) 435-478.

Lentz, W.J., 1973. A new method of computing spherical Bessel functions of complex argument with tables. U.S. Army Electronics Command Technical Report ECOM-5509.

```
0001          SUBROUTINE CSBSJ(X, JJ, M, IER)
      C*************************************************************************
      C  12-FEB-79  CHECK FOR XR=0. BEFORE ALOG IS TAKEN
      C   8-DEC-78  STOP ITERATIONS WHEN (RATIO-1)<EPS
      C  26-OCT-78   OPTIMIZE FOR SPEED
      C  4-AUG-78
      C  D. STANDLEY
      C
      C   COMPUTE COMPLEX SPHERICAL BESSEL FUNCTIONS BY USE OF CONTINUED FRACTIONS.
      C  SEE: 'A METHOD OF COMPUTING SPHERICAL BESSEL FUNCTIONS OF COMPLEX
      C        ARGUMENT WITH TABLES'  BY W. J. LENTZ
      C-------------------------------------------------------------
      C   SUBROUTINE CSBSJ(X, JJ, M, IER)
      C
      C   DESCRIPTION OF PARAMETERS
      C      X  -THE ARGUMENT OF THE J SPHERICAL BESSEL FUNCTION DESIRED
      C          X IS COMPLEX
      C     JJ  -THE RESULTANT J SPHERICAL BESSEL FUNCTION
      C          JJ IS THE ARRAY OF M ORDERS
      C          JJ IS COMPLEX
      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=3 ORDER > 49
      C
      C  THE ACCURACY IS 5 TO 6 SIGNIFICANT FIGURES
      C
      C*************************************************************************
0002          COMPLEX X, A, SBES, QUO(50), NUMN, NUMNM1, DENN, DENNM1, JJ(50)
0003          COMPLEX ONE,TWO,TWODX,CSX,RATIO
0004          A(N) = TWODX * CMPLX(V+FLOAT(N-1), ZERO)
0005          C1 = 1.
0006          EPSR = 1.E-5
0007          EPSI = 1.E-3
0008          ZERO = 0.
0009          ONE = (1.,0.)
0010          TWO = (2.,0.)
0011          IF(M .GT. 49)GO TO 514            !ORDER TOO BIG
0013          XR = REAL(X)
0014          XI = AIMAG(X)
0015          CSX = CSIN(X)
0016          VV = FLOAT(M)
0017          IXR = 0
0018          ICSR = 0
0019          IF(ABS(XI) .GT. 80.)GO TO 513     !OVERFLOW CONDITION!
0021          IF(XR .EQ. ZERO)GO TO 4
0023          ICSR = IFIX(ALOG10(ABS(REAL(CSX))))
0024          IXR = IFIX(ALOG10(ABS(XR)))
0025    4     ICSI = 0
0026          IXI = 0
0027          IF(XI .EQ. ZERO)GO TO 5
0029          ICSI = IFIX(ALOG10(ABS(AIMAG(CSX))))
0030          IXI = IFIX(ALOG10(ABS(XI)))
0031    5     V = VV + 1.5
0032          IF(VV .EQ. ZERO)GO TO 501         !CALCULATE DIRECTLY
0034          TWODX = TWO / X
      C-----------------------
      C  CALCULATE THE RATIO OF J(V-1)/J(V)
      C
0035          DO 500 IV = 1, IFIX(VV)           !GET NECESSARY RATIOS
0036          V = V - 1.
0037          L = IFIX(V - 0.5)
```

```
0038          NUMNM1 = A(1)
0039          DENNM1 =A(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)-C1) .GT. EPSR)GO TO 11 !CHECK FOR CONVERGENCE
0048          IF(ABS(AIMAG(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(AIMAG(SBES))))
0069    504   IF((ISR+ICSR).GT.-36.AND.(ISR+ICSR-IXR).GT.-36)GO TO 505
0071          GO TO 512                          !REAL UNDERFLOW
0072    505   IF((ISI+ICSI).GT.-36.AND.(ISI+ICSI-IXI).GT.-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.