```
PRINT "model fish(f) ray trace and simplified calculation for m = 0   10/3/92 csc "

DIM fr(4200),fi(4200),p(4200)
DIM Sr(2000),Si(2000)
DIM xfb(50),zufb(50),zlfb(50),wfb(50), xsb(50),zusb(50),zlsb(50),wsb(50)
DIM u(50),vm(50),vu(50),eas(50),ma(50),du(50),bu(50),maf(50),uf(50)
DIM vmf(50),vuf(50),maf(50),duf(50),buf(50),dxf(50),vlf(50),blf(50)

 pi = 4*ATN(1)
 ipi = 1/pi
 lgcv =20/LOG(10)
 esp = .000005 :'set level for the dB calc and glitches

'coef for J0(x)
jc2=-2.25: jc4=1.2656

'coef for Y0(x)
  yb0=2/pi: yb1=.367467: yb2= .60559: yb4= -.7435

'coef for J1(x)
 ja0=.5: ja2=-.5625 : ja4 = .211

 ' coef for Y1
 ya0=2/pi: ya1=-.63662: ya2=.2212: ya4 = 2.17

'Kirchhoff coefficients to match curves at ka = .15
    psh = -1.7: ksh = .25: pb = 50: kb = .2
    gfac = .98 :   qp1 = .79 : qp2 = .2

'data for air filled swimbladder
 csb = 345
 PA = 101000!
 z = 0
 rhow = 1035
 cw = 1500
 rhosb =1.24 : ' kg/m^3

 'data for fluid filled fish body
 rhof = 1070: pw = 1000
 cfb = 1575

 df = 2000
 n2 = 120
 n1=3
 thetad = 90 :theta = pi/2

 PRINT" fish in water"

11    PRINT "menu"
      PRINT"      'rdf' to read data file"
     PRINT"input 'f' for f increments and n2"
     PRINT"      'z' for depth "
     PRINT"      'pp' print parameters"
     PRINT"      'cp' to compute fish parameters - auto for read data file"
     PRINT"input 'c' to compute"
     PRINT"      'g' to graph"
     PRINT"      'mdf' to make data file"
     PRINT"      'q' to quit"
     PRINT"      'm' for menu"
12     PRINT"go: 'm' for menu";:INPUT q$

     IF q$ = "mdf" GOTO 4000
     IF q$ = "cp" GOTO 4500
     IF q$ = "rdf" GOTO 5000
```

```
     IF q$ = "f" GOTO 60
     IF q$ = "z" GOTO 70
     IF q$ = "pp" GOTO 100
     IF q$ = "c" GOTO 200
     IF q$ = "g" GOTO 500
     IF q$ = "q" GOTO 6000
     IF q$ = "m" GOTO 11
     GOTO 12

60 PRINT "delta f = ";df;"  nmin ="; n1;"  nmax ="; n2
  PRINT "nmax = 4200: Input delta f =";:INPUT df
  PRINT "input nmin, nmax =";:INPUT n1,n2

 GOTO 12

70 PRINT"old depth =";zd;"  input depth z=";:INPUT zd

 GOTO 12


 100 PRINT
 PRINT"              init n1 =";n1
 PRINT"              final n2 =";n2
 PRINT"              delta f =";df
 PRINT"                 depth =";zd
 PRINT"
 PRINT
GOTO 12

200 'compute
'Computations are reduced, S(ka)/L
' b0 = -1/(1+ic0)
'S(ka)/L = -i(1/pi) b0 = (1/pi)[c0/(1+c0^2) +i/(1+c0^2)]
'Use Clay J. Acoust .Soc.89, 2168-2179 (1991)
'Use polynomial approximations for the Bessel functions.
'subroutines for J0(x), J1(x),Y0(x) and Y1(x) are short for modes 0 and 1
'when the range of ka is less than .5.

'For gas bladders, only the m = 0 terms at very small ka.
'the calculations for c(0) use (11) of csc. Sign adjusted for exp(2pi ft -kr)
'The subroutines are put in the calculations.
'dJ0 = -j1 and dy0 = - y1

' The ray- Kirchhoff approximation uses empirical amplitude
'     qk = ksh*(1+x/(kb+x)),      x+qk -> x for large x
'and phase shifts  adjustments.
'     qp = psh*(1+x/(pb+x)),      qp -> psh for small x

'finite cylinder model
'    mas(j) = effective radius of j' cylinder
'    mzs(j) = mean depth for j'th culinder
'    dq = 2*pi*df/cw
'    dka = dq*mas(j)

' Kirchhoff coefficients to fit the Kirchhoff curves for ka > 0.15
'to the mode curves for ka <  0.15

'   real S(n) = Gk1*SQR(x+qk)*SIN(2*x+qp) with Gk1 = refl/[2*sqr(pi)]
'note---here x= +ka. in fish model, x = -2*pi*f*[vu(j)+b01*du(j)]
'   imag S(n) = -Gk1*SQR(x+qk)*COS(2*x+qp).
'   mkz = n*dq*mzs(j)

 FOR n = 0 TO 2000
    p(n) = 0
```

```
    fr(n) = 0
    fi(n) = 0
    Sr(n) = 0
    Si(n) = 0
NEXT n

PRINT"do swimbladder 'y' or 'n'";:INPUT q$
IF q$ = "n" GOTO 260

  PA = 101000!+9.8*rhow*zd
  g = PA*rhosb/(rhow*101000&)
  h = csb/cw

PRINT"g = ";g;" h = ";h
   rfl = (g*h-1)/(g*h+1)
   Gk1 = rfl/(2*SQR(pi))
    dq = 2*pi*df/cw
 FOR n = n1 TO n2
   FOR j = 0 TO Jsb-1  : 'loop on the finite cylinder model
     dka = dq*eas(j)
     mkz = dq*vm(j)
      x = n*dka
      IF x >.15 GOTO 220  : ' kirchhoff aprox for each j
 'water, calc  Bessel functions
    q=x/3:q1=q
    q2=q^2:q3=q^3:q4=q^4:
    yLx= ya0*LOG(x/2)

    J0 = 1+jc2*q2+jc4*q4
    J1 = x*(.5+ja2*q2+ja4*q4)
    y0 = yLx*J0+yb1+yb2*q2+yb4*q4
    y1 = yLx*J1+(ya1+ya2*q2+ya4*q4)/x

    jw0 = J0  :  jw1 = J1:  djw0 = -J1
    yw0 = y0  :dyw0 = -y1

'cylinder, calc Bessel functions
    x = n*dka/h
    q=x/3:q1=q
    q2=q^2:q3=q^3:q4=q^4:
    yLx= ya0*LOG(x/2)

    J0 = 1+jc2*q2+jc4*q4
    J1 = x*(.5+ja2*q2+ja4*q4)
    y0 = yLx*J0+yb1+yb2*q2+yb4*q4
    y1 = yLx*J1+(ya1+ya2*q2+ya4*q4)/x

    jc0 = J0 : jc1 = J1:  djc0 = -J1
    yc0 = y0 :dyc0 = -y1

'compute c(n) for b(n) and the scattering amplitude

    cn0 = djc0*yw0 - g*h*dyw0*jc0
    cd0 = djc0*jw0 - g*h*jc0*djw0
    c0 = cn0/cd0
    u0 = 1+c0^2
    rSn = ipi*c0/(u0*sfLm)
    iSn = -ipi/(u0*sfLm)
     Sr(n) = Sr(n) + (rSn*COS(2*mkz) + iSn*SIN(2*mkz))*dxs(j)
     Si(n) = Si(n) - (iSn*COS(2*mkz) - rSn*SIN(2*mkz))*dxs(j)
     xt = n*dka :' x is value in water
   IF xt =< .15 GOTO 240

220  'Kirchhoff approximation for x = ka > 0.15
```

```
        sncu = 1
       x = n*dq*ma(j)
       kbu= n*dq*bu(j)*du(j)
       IF ABS(kbu)>.1 THEN sncu = SIN(kbu)/kbu
      qk = .5*(1+x/(.5+x))
      qp = x/(40+x)-.2/(1+x) -1
      junk = n*dq*(2*vu(j) + bu(j)*du(j)) + qp +pi/2
      rSn = sncu*Gk1*SQR(x+qk)*COS(junk)/sfLm
      iSn = -sncu*Gk1*SQR(x+qk)*SIN(junk)/sfLm
      Sr(n) = Sr(n) + rSn*dxs(j)
      Si(n) = Si(n) + iSn*dxs(j)
240    NEXT j
 NEXT n

 260 'fluid cylinder
PRINT"do fluid filled fish body 'y' or 'n'";:INPUT q$
IF q$ = "n" GOTO 280
      gfb = rhof/rhow
      hfb = cfb/cw
      refl = (gfb*hfb-1)/(gfb*hfb+1)
      Tc = 1 - refl^2
      dpsi = 2 *(1-hfb)
      Gk1 = gfac*refl/(2*SQR(pi))
      dq = 2*pi*df/cw
      PRINT"gfb=";gfb;" hfb=";hfb

FOR n = n1 TO n2
    FOR j = 0 TO Jfb-1
      sncu = 1
      sncL = 1
      k1 = n*dq
      k2 = k1/hfb
      zu = vuf(j) + buf(j)*duf(j)/2
      zL = vlf(j) + blf(j)*duf(j)/2
         x = ABS(k1*zu)
      ka =ABS( k1*maf(j))
      xh = x/hfb
      ampx = Gk1*SQR(ABS(ka))
      psi = dpsi*xh
      qp = -.5*pi*x/(x+.4)
      ang1 = 2*k1*zu
      ang2 = - 2*k1*zu + 2*k2*(zu-zL) + qp
      kbu= k1*buf(j)*duf(j)
            IF ABS(kbu)>.1 THEN sncu = SIN(kbu)/kbu
      kbL= k2*blf(j)*duf(j)
            IF ABS(kbL)>.1 THEN sncL = SIN(kbL)/kbL

      rSn = -ampx*(SIN(ang1)*sncu + Tc*SIN(ang2)*sncL)*duf(j)
      iSn = -ampx*(COS(ang1)*sncu - Tc*COS(ang2)*sncL)*duf(j)
      Sr(n) = Sr(n) + rSn/sfLm
      Si(n) = Si(n) + iSn/sfLm
   NEXT j
NEXT n

280 GOTO 12

500 REM         MAKE GRAPH

501 PRINT " graph p or log p, input 'p' or 'y' ";:INPUT gc$
gp$ = "S(f)/sFL"
rdf = 1
PRINT" plot reduced S(f) 'y' or 'n'";:INPUT r$
IF r$ = "n" THEN rdf = sfLm
IF r$ = "n" THEN gp$ = "S(f)"
```

```
 IF gc$ = "p" GOTO 505
 IF gc$ = "y" GOTO 510
 GOTO 501

505 PRINT" choose 'r' , 'i' , 'a' of S(f)/L, 'k' "; :INPUT b$
g$ = b$
asmax = 0
FOR n = n1 TO n2
IF b$  = "r" THEN p(n) = rdf*Sr(n)
IF b$  = "i" THEN p(n) = rdf*Si(n)
IF b$  = "a" THEN p(n) = rdf*SQR(Sr(n)^2+Si(n)^2)
IF asmax < ABS(p(n)) THEN asmax = ABS(p(n))
NEXT n
 PRINT " |S max | = " ;asmax*rdf
 PRINT "input amp factor =";:INPUT af
   afp = af
   GOTO 515

510 ' plot p(n) in dB
   pmax = -100000!
 FOR n = n1 TO n2
    p(n) = lgcv*LOG(rdf^2*(Sr(n)^2 + Si(n)^2)+esp)/2
    IF pmax < p(n) THEN pmax = p(n)
 NEXT n

 PRINT "pmax = ";pmax;" dB"

PRINT"input reference level dB =";:INPUT dbr
  af = 1
  s$ = "log"

515 PRINT " max f , kHz= ";n2*df/1000;
 PRINT" input ticks at delta f, kHz =";:INPUT sikkhz
 sika = sikkhz*1000
   f2 = n2*df
   lamda = cw*1000/sika : 'in mm

REM     SCREEN DIMENSIONS
      XL = 480
      YL = 260

REM     SET SCALES

 X0 = 20
      XS = (XL - X0)/f2:    '  X(NM) IS MAXIMUM VALUE OF X
          y0 = YL/2:            '  TO PUT Y=0 near MIDDLE
          YS = YL/3:            '  THIS SETS THE AMPLITUDE FACTOR.

 IF gc$ = "y" THEN
      YU = 20
      YS = (YL-YU)/80
END IF

REM     TOOL BOX CALLS REQUIRE INTEGERS. % INDICATES INTEGER
REM     CALCULATE X% AND Y% AND THEN PLOT TO X1% AND Y1%.

 CLS          : REM CLS clears the screen
 PICTURE ON   : REM PICTURE ON puts screen graphics in storage.
 SHOWPEN      : REM SHOWPEN also puts graphics on the screen

   FOR n = n1 TO n2-1
     x = n*df
     x1 = (n+1)*df
```

```
        x% =  INT (XS * x + X0)
        x1% =  INT (XS * x1 + X0)
 IF gc$ = "p"  THEN
         py =  afp*YS*p(n)
        py1 = afp*YS*p(n+1)
        y% = INT (y0 - py)
        y1% = INT (y0 -py1)
     LINE (x%,y%)-(x1%,y1%)
  END IF

   IF gc$ = "y"   THEN
       y% = INT (YU - YS *(p(n)-dbr))
       y1% =INT (YU- YS *(p(n + 1)-dbr))
       LINE (x%,y%)-(x1%,y1%)
  END IF

  NEXT n

REM     PUT TICS ON THE X-AXIS

 x% = INT (X0) : x1% = INT (XS*f2 + X0)
 y% = y0
 YU% = YU
 YL% = YL
 np = sika/(df)

 ya = y0
 IF gc$ = "y" THEN ya = YU +70*YS
 ya% = ya

 LINE (x%,ya%) - (x1%,ya%)            :REM draw axis
 LINE (x%,YL%) - (x%,YU%)

 FOR n = 0 TO n2 STEP np
      x = n*df
     x% =  INT (XS * x + X0)             :' locate tics
     y% = INT (ya )                         :' make tics
     y1% = INT (ya +5)
     LINE (x%,y%) - (x%,y1%)          :' draw tics
     num = INT(100*n*df+.1)/100000&
     CALL MOVETO (x%-9,260) : PRINT num : ' moveto and print N
 NEXT n

 x% = INT (X0)

 IF gc$ = "p" THEN
 FOR m = -5 TO 5
      y% = INT(y0 - m*YS/5)
      LINE (x%,y%) - (x%+5,y%)
 NEXT m

 ELSE
 FOR m = 0 TO 8
      y% = INT(YU + m*YS*10)
      LINE (x%,y%) - (x%+5,y%)
 NEXT m

 END IF

 CALL MOVETO (20,16): PRINT g$;gp$;" theta =";thetad;"z=";zd;" step sFL/lamda=";sFL/lamd
 CALL MOVETO (20,280)
  IF gc$="p" THEN PRINT name2$;" ";s$;" y-tics=";.2/af;"sfL=";sFL;"den f,w=";rhof;rhow;"
  IF gc$="y" THEN PRINT name2$;" ";s$;" dB ref=";dbr;"sfL=";sFL;"den f,w=";rhof;rhow;"c
```

```
 PICTURE OFF : REM PICTURE OFF ends graphics operations.
 INPUT q$

 CALL MOVETO (20,280)
   PRINT " input 'mf' to make a file "
    INPUT q$
 IF q$ <> "mf" GOTO 520

 pic$ = PICTURE$          :REM PICTURE$ is name of stored picture.

 CALL MOVETO (50, 25)     :REM name the file
 PRINT  "I've got the picture in pic$ ("; LEN (pic$); ")"
 pictFile$ = FILES$ (0, "Enter name for PICT file:")
 PRINT "PICT file name is:"; pictFile$

REM     SAVE FILE IN 'PICT' FORMATE.

 OPEN pictFile$ FOR OUTPUT AS #1

REM     FOR-NEXT LOOP MAKES A HEADER FOR PICT FILE FORMATE.

 FOR i = 1 TO 512 : PRINT  #1, CHR$ (0); : NEXT

 PRINT  #1, pic$

 CLOSE  :REM the picture 'pic$' is stored as a text file.

REM     CHANGE THE FILE TYPE FROM TEXT TO PICT

 NAME pictFile$ AS pictFile$, "PICT"

REM     USE MacDraw TO READ THE FILE. THEN,
REM     IT CAN BE SAVED AS A MacDraw DRAWING.

520  CLS    :REM clear screen and clean memory
PICTURE ON
PICTURE OFF

GOTO 12

4000  PRINT "make a spectrum file for IFFT"
 PRINT "   complex data is in fr(n) and fi(n)."
 PRINT "   n2, number of data in calc = ";n2
 PRINT "choose the number of frequency coefficients, nt = 2^n"
 PRINT "max nt = 4200. input nt =";:INPUT nt
 PRINT "file maker constructs the the coefficients from nt/2 to nt."

 FOR n = 0 TO nt/2-1
    fr(nt-n)= fr(n)
    fi(nt-n) = -fi(n)
 NEXT n
 fr(nt/2) = 0
 fi(nt/2) = 0

 PRINT"give file name":INPUT n3$
 OPEN n3$ FOR OUTPUT AS #3

 WRITE #3, nt

     FOR n = 0 TO nt
         WRITE #3,fr(n),fi(n)
     NEXT n

 WRITE #3, dka     :'delta ka
```

```
 WRITE #3, a        :' nominal radius
 WRITE #3, pw       :'water density
 WRITE #3, cw       :'sound vel water
 WRITE #3, pcyli    :'density in cylinder
 WRITE #3, ccyl     :'sound speed in cylinder
 WRITE #3, zd        :'depth
 WRITE #3, delta     :'deflection of cyl in a

 CLOSE #3

 GOTO 12

5000 ' read data file in 'fish data file maker' format
PRINT"read file ";:INPUT name2$
  OPEN name2$ FOR INPUT AS #2
 INPUT #2, ftype$
 INPUT #2, words1$,fL
 INPUT #2, words2$,mfb
 INPUT #2,Jfb
 INPUT #2,words7$
 INPUT #2,words5$

 FOR j = 0 TO Jfb
    INPUT #2,xfb(j),zufb(j),zlfb(j),wfb(j)
 NEXT j

 INPUT #2, words6$

  INPUT #2, Jsb
  FOR j = 0 TO Jsb
    INPUT #2,xsb(j),zusb(j),zlsb(j),wsb(j)
 NEXT j

 INPUT #2,words8$
 CLOSE #2

4500 ' compute equivalent cylinders
'convert initial fish dimensions in mm to m
' fish body ---   dxf(50),mxf(50),mzf(50),eaf(50)
' swimbladder --- dxs(50),mxs(50),mzs(50),eas(50)

'u and v are rotated axis rotation is theta in std cyl scat convention.
'u is along the incident wave front
'v is along the ray path back to the receiver
'theta = pi/2 is normal incidence on the cylinder.

' u(j) is the rotated displacement of the center of jth
' element of cylinder along the axis of the cylinder
' vm(j) is the v  of the mean of the  jth element of cylinder.
' vu(j) is the displacement of the top (upper face)
'  of jth element of cylinder
' ma(j) is the mean half width of the upper face
' bu(j) is the slope of the upper face
' du(j) is the length
PRINT" fish length = ";fL;" mm"
sbL =xsb(Jsb)-xsb(0)
PRINT" swimbladder length = ";sbL;" mm"
PRINT" scale length = 150 mm lets L/lamda = 1 correspond to 10 kHz."
PRINT "  input scale fish length =";: INPUT sFL
PRINT "  old theta =";theta*180/pi;" new=";: INPUT thetad
theta = thetad*pi/180
sF = sFL/fL
sfLm = sFL/1000
PRINT "scale fish length =";sF*fL
```

```
PRINT "scale swimbladder =";sF*sbL


'geometry for breathing mode volume dv(j) and scatter from upper face
snth=SIN(theta)
csth = COS(theta)
'PRINT "ma(j)", "vu(j)", "bu(j)", "u(j)"
FOR j = 0 TO Jsb-1
    z0 = sF*(zusb(j)-zlsb(j))/2000
    z1 = sF*(zusb(j+1)-zlsb(j+1))/2000
    y0 = sF*wsb(j)/2000
    y1 = sF*wsb(j+1)/2000
    dx = sF*(xsb(j+1)-xsb(j))/1000
    dxs(j) = dx
    xm = sF*(xsb(j)/1000 + dx/2 )
    duz = sF*(zusb(j+1)-zusb(j))/1000
    zm = sF*(zusb(j)+zlsb(j) + zusb(j+1)+zlsb(j+1) ) /4000
    zus = sF*(zusb(j) + zusb(j+1)) /2000
    yb = (y1-y0)/dx
    dv(j) = pi*ABS( (z0*y0*dx + (zb*y0+yb*z0)*dx^2/2 + zb*yb*dx^3/3 ) )
    eas(j) = SQR(dv(j)/(pi*dx))
    u(j) = xm*snth - zm*csth
    vm(j) = xm*csth+zm*snth
    vu(j) = xm*csth + zus*snth
    ma(j) =  (y0 + y1)/2
    du(j) = dx*snth
    IF du(j)<>0 THEN bu(j) = (dx*csth+duz*snth)/du(j)

    'PRINT ma(j),vu(j),bu(j),u(j)
NEXT j
'INPUT q$

'PRINT "ma(j)", "vu(j)", "bu(j)", "u(j)"
FOR j = 0 TO Jfb-1
    z0 = sF*(zufb(j)-zlfb(j))/2000
    z1 = sF*(zufb(j+1)-zlfb(j+1))/2000
    y0 = sF*wfb(j)/2000
    y1 = sF*wfb(j+1)/2000
    dx = sF*(xfb(j+1)-xfb(j))/1000
    xm = sF*(xfb(j)/1000 + dx/2 )
    duz = sF*(zufb(j+1)-zufb(j))/1000
    zm = sF*(zufb(j)+zlfb(j) + zufb(j+1)+zlfb(j+1) ) /4000
    zus = sF*(zufb(j) + zufb(j+1)) /2000
    yb = (y1-y0)/dx
    uf(j) = xm*snth - zm*csth
    vmf(j) = xm*csth+zm*snth
    vuf(j) = xm*csth + zus*snth
    maf(j) =  (y0 + y1)/2
    duf(j) = dx*snth
    IF duf(j)<>0 THEN buf(j) = (dx*csth+duz*snth)/duf(j)
    'PRINT maf(j),vuf(j),buf(j),uf(j)
NEXT j
'INPUT q$

'PRINT "ma(j)", "vu(j)", "bu(j)", "u(j)"
FOR j = 0 TO Jfb-1
    z0 = sF*(zlfb(j)-zlfb(j))/2000
    z1 = sF*(zlfb(j+1)-zlfb(j+1))/2000
    y0 = sF*wfb(j)/2000
    y1 = sF*wfb(j+1)/2000
    dx = sF*(xfb(j+1)-xfb(j))/1000
    xm = sF*(xfb(j)/1000 + dx/2 )
    dlz = sF*(zlfb(j+1)-zlfb(j))/1000
    zm = sF*(zufb(j)+zlfb(j) + zufb(j+1)+zlfb(j+1) ) /4000
```

```
    zls = sF*(zlfb(j) + zlfb(j+1)) /2000
    yb = (y1-y0)/dx
    vlf(j) = xm*csth + zls*snth
    IF duf(j)<>0 THEN blf(j) = (dx*csth+dlz*snth)/duf(j)
    'PRINT maf(j),vlf(j),blf(j),uf(j)
NEXT j
'INPUT q$
 GOTO 12

6000 END
```