```
c      program orbit1.for
C      Sep. 20, 2005
C      COMPUTES MIE SCATTERING BY a modified Luneburg lens
       IMPLICIT DOUBLE PRECISION (A-B,D-H,O-Z)
       DIMENSION AN(129),r(128),X(128),Y(128)
       DIMENSION ex0(128),ey0(128),fx0(128),fy0(128)
       DIMENSION ex(140,128),ey(140,128),fx(140,128),fy(140,128)
       DIMENSION gx(140),gg(140,128)
       DIMENSION TAM(128),TAQ(128),TAD(128),TAP(128)
       DIMENSION TBM(128),TBQ(128),TBD(128),TBP(128)
       DIMENSION TEM1(64),TEM2(64),TEM3(64),TEM4(64)
       DIMENSION TEM5(64),TEM6(64),TEM7(64),TEM8(64)
       DIMENSION CTH(361),PI(2,361)
       DIMENSION SRP(361),SIP(361),SRM(361),SIM(361),PT(3,361)
c

       PII=3.1415926536
       write(*,100)
  100  format(' particle radius in microns=')
       read(*,101) a
       a2=a*a
  101  format(f14.6)
       write(*,102)
  102  format(' magnitude of focal position/a=')   This is particular to the refractive index profile
       read(*,101) f                                of a Luneburg lens.
       f2=f*f
       f21=f2+1.
       write(*,103)
  103  FORMAT(' INTEGER MP <=7 FOR 2**MP interfaces=')
       READ(*,104) MP
  104  FORMAT(I8)
       MMAX=2**MP      # of interfaces
       MMAX1=MMAX+1    # of regions
       ammax=dfloat(mmax)
       WRITE(*,105)
  105  FORMAT(' WAVELENGTH EXAMINED IN MICRONS=')
       READ(*,101) WL
       write(*,106)
  106  format(' initial angle in degrees=')
       read(*,101) agi
       write(*,107)
  107  format(' FINAL ANGLE IN DEGREES=')
       read(*,101) AGF
       NMAX=361                 361 angles are plotted
       AGDEL=(AGF-AGI)/DFLOAT(NMAX-1)
c
         DO 200 M=1,MMAX
       r(m)=dfloat(m)*a/ammax   radius of interface m/m+1
  200  CONTINUE
       rave=.5*r(1)
       an(1)=f21/f2-rave*rave/(a2*f2)   refractive index in the middle of region m
       an(1)=dsqrt(an(1))
         do 201 m=2,mmax
       rave=.5*(r(m)+r(m-1))
       an(m)=f21/f2-rave*rave/(a2*f2)
       an(m)=dsqrt(an(m))
  201  continue
       AN(MMAX1)=1.   refractive index of the exterior region
         DO 202 M=1,MMAX
       X(M)=2.*PII*AN(M+1)*R(m)/WL   just outside interface m/m+1 in region m+1   X(MMAX)= overall size
       Y(M)=2.*PII*AN(M)*R(m)/WL     just inside interface m/m+1 in region m          parameter
  202  CONTINUE
c
         do 211 n=1,NMAX
       th=PII*(agi+agdel*Dfloat(n-1))/180.
sum over CTH(n)=DCOS(th)
angles SRP(N)=0.
       SRM(N)=0.
       SIP(N)=0.            initialize everything at every angle n
       SIM(N)=0.
       PI(1,N)=0.
       PI(2,N)=1.
  211  continue
```

overall size parameter

```
c
      QLMAX=2.+x(MMAX)+4.315*(x(MMAX)**.3333)
      LMAX=ifix(SNGL(QLMAX))
      lstart=lmax+15
```

highest partial wave based on the total size parameter

```
c
        DO 203 M=1,MMAX
```

for each interface m/m+1

```
      ex(lstart,m)=0.0d+00
      ey(lstart,m)=0.0d+00
        do 215 LL=0,lstart-2
      L=lstart-LL
      alx=dfloat(L)/x(m)
      aly=dfloat(L)/y(m)
      ex(L-1,m)=alx-1./(ex(L,m)+alx)
      ey(L-1,m)=aly-1./(ey(L,m)+aly)
  215 continue
      alx=1./x(m)
      aly=1./y(m)
      ex0(m)=alx-1./(ex(1,m)+alx)
      ey0(m)=aly-1./(ey(1,m)+aly)
```

g_ℓ'/g_ℓ by downward recursion at the m/m+1 interface $E_\ell(x)$ $E_\ell(y)$

m

```
c
      fx0(m)=-dtan(x(m))
      fy0(m)=-dtan(y(m))
      alx=1./x(m)
      aly=1./y(m)
      fx(1,m)=-alx+1./(alx-fx0(m))
      fy(1,m)=-aly+1./(aly-fy0(m))
        do 216 L=2,lmax
      alx=dfloat(L)/x(m)
      aly=dfloat(L)/y(m)
      fx(L,m)=-alx+1./(alx-fx(L-1,m))
      fy(L,m)=-aly+1./(aly-fy(L-1,m))
  216 continue
  203 CONTINUE
```

n_ℓ'/n_ℓ by upward recursion at the m/m+1 interface $F_\ell(x)$ $F_\ell(y)$

```
c
        do 219 m=2,mmax
      gg0=dtan(x(m-1))/dtan(y(m))
      alx=1./x(m-1)
      aly=1./y(m)
      gg(1,m)=gg0*((alx+fx(1,m-1))/(aly+fy(1,m)))
      gg(1,m)=gg(1,m)*((aly+ey(1,m))/(alx+ex(1,m-1)))
        do 217 L=2,lmax
      alx=dfloat(L)/x(m-1)
      aly=dfloat(L)/y(m)
      gg(L,m)=gg(L-1,m)*((alx+fx(L,m-1))/(aly+fy(L,m)))
      gg(L,m)=gg(L,m)*((aly+ey(L,m))/(alx+ex(L,m-1)))
  217 continue
  219 continue
```

m = region number

m

$\frac{g_\ell/n_\ell(x)}{g_\ell/n_\ell(y)}$ in the m region $\frac{G_\ell(x)}{G_\ell(y)}$

by upward recursion

```
c
      gx0=-dtan(x(mmax))
      alx=1./x(mmax)
      gx(1)=gx0*((alx+fx(1,mmax))/(alx+ex(1,mmax)))
        do 218 L=2,lmax
      alx=dfloat(L)/x(mmax)
      gx(L)=gx(L-1)*((alx+fx(L,mmax))/(alx+ex(L,mmax)))
  218 continue
```

$\underbrace{g_\ell/n_\ell}_{=G_\ell}$ for x just outside the outermost interface by upward recursion

```
c
        do 204 L=1,LMAX
      q=Dfloat(L)
      Q1=Q+1.
      q21=2.*q+1.
      QQ1=Q21/Q
      QQ2=Q1/Q
      QQ3=Q21/(Q*Q1)
C
        DO 205 M=1,MMAX
      TAM(M)=an(m+1)*ey(L,m)-an(m)*ex(L,m)
      TAQ(M)=an(m+1)*fy(L,m)-an(m)*fx(L,m)
      TAD(M)=an(m+1)*ey(L,m)-an(m)*fx(L,m)
      TAP(M)=an(m+1)*fy(L,m)-an(m)*ex(L,m)
      TBM(M)=an(m)*ey(L,m)-an(m+1)*ex(L,m)
      TBQ(M)=an(m)*fy(L,m)-an(m+1)*fx(L,m)
      TBD(M)=an(m)*ey(L,m)-an(m+1)*fx(L,m)
      TBP(M)=an(m)*fy(L,m)-an(m+1)*ex(L,m)
```

m = interface

single-scattering amplitudes at each interface TM (a) and TE (b)

n_ℓ

q_ℓ

d_ℓ

p_ℓ

ℓ

```
  205 CONTINUE
C
        DO 206 J=1,MP
      JP2=2**(MP-J)
      ij1=2**(j-1)
C
        DO 207 K=1,JP2
      K2=2*K
      K21=K2-1
      kk=ij1*k21+1
      grat=gg(L,kk)
      TEM1(K)=TAD(K21)*TAM(K2)-TAM(K21)*TAP(K2)*grat
      TEM2(K)=TAQ(K21)*TAD(K2)-TAP(K21)*TAQ(K2)*grat
      TEM3(K)=TAD(K21)*TAD(K2)-TAM(K21)*TAQ(K2)*grat
      TEM4(K)=TAQ(K21)*TAM(K2)-TAP(K21)*TAP(K2)*grat
      TEM5(K)=TBD(K21)*TBM(K2)-TBM(K21)*TBP(K2)*grat
      TEM6(K)=TBQ(K21)*TBD(K2)-TBP(K21)*TBQ(K2)*grat
      TEM7(K)=TBD(K21)*TBD(K2)-TBM(K21)*TBQ(K2)*grat
      TEM8(K)=TBQ(K21)*TBM(K2)-TBP(K21)*TBP(K2)*grat
  207 CONTINUE
C
        DO 208 K=1,JP2
      TAM(K)=TEM1(K)
      TAQ(K)=TEM2(K)
      TAD(K)=TEM3(K)
      TAP(K)=TEM4(K)
      TBM(K)=TEM5(K)
      TBQ(K)=TEM6(K)
      TBD(K)=TEM7(K)
      TBP(K)=TEM8(K)
  208 CONTINUE
  206 CONTINUE
C
      tam(1)=tam(1)*gx(L)
      tbm(1)=tbm(1)*gx(L)
      ADEN=TAM(1)*TAM(1)+TAD(1)*TAD(1)
      AR=TAM(1)*TAM(1)/ADEN
      AI=-TAM(1)*TAD(1)/ADEN
      BDEN=TBM(1)*TBM(1)+TBD(1)*TBD(1)
      BR=TBM(1)*TBM(1)/BDEN
      BI=-TBM(1)*TBD(1)/BDEN
      ABRP=QQ3*(AR+BR)
      ABIP=QQ3*(AI+BI)
      ABRM=QQ3*(AR-BR)
      ABIM=QQ3*(AI-BI)
C
        DO 212 N=1,NMAX
      TA=Q*CTH(N)*PI(2,N)-Q1*PI(1,N)
      PITAP=PI(2,N)+TA
      PITAM=PI(2,N)-TA
      SRP(N)=SRP(N)+ABRP*PITAP
      SRM(N)=SRM(N)+ABRM*PITAM
      SIP(N)=SIP(N)+ABIP*PITAP
      SIM(N)=SIM(N)+ABIM*PITAM
      TEMP=PI(2,N)
      PI(2,N)=QQ1*CTH(N)*PI(2,N)-QQ2*PI(1,N)
      PI(1,N)=TEMP
  212 CONTINUE
  204 CONTINUE
C
        do 213 n=1,NMAX
      S1R=.5*(SRP(N)+SRM(N))
      S2R=.5*(SRP(N)-SRM(N))
      S1I=.5*(SIP(N)+SIM(N))
      S2I=.5*(SIP(N)-SIM(N))
      pt(1,n)=S1R*S1R+S1I*S1I
      pt(2,n)=S2R*S2R+S2I*S2I
      pt(3,n)=.5*(pt(1,n)+pt(2,n))
  213 continue
C
      OPEN(UNIT=1,FILE='OUT.DAT')
        DO 210 N=1,NMAX
      TH=AGI+AGDEL*DFLOAT(N-1)
```

Handwritten annotations beside the code:

- (DO 206) j = level of combining amplitudes = iteration number
- (DO 207) K = each pair of amplitudes in the level
- (TEM1–TEM8) combine the amplitudes in the old level to get the amplitudes in the new level
- (DO 208) write the new amplitudes in the first half of the storage places for the old amplitudes. There are now half the amplitudes as in the previous iteration level. The last half of the storage places hold the old amplitudes
- (tam(1)=tam(1)*gx(L)) ← storage location 1 contains the final amplitudes
- (AR, AI) } a_ℓ
- (BR, BI) } b_ℓ
- (DO 212) calculate S_1 and S_2 on the fly for each angle n
- (OPEN / DO 210) This stores the output data

```
      WRITE(1,998) TH,PT(1,N),PT(2,N),PT(3,N)
  998 FORMAT(F14.1,3E15.5)
  210 CONTINUE
      CLOSE(UNIT=1,STATUS='KEEP')
  400 continue
      end
```