**Backscattering modeling part**

Produce imaginary index of calcite as a continuous function of wavelenght

(\* input imaginary coccolithophore core index (aimee,stramski) \*)

filename="coccolithophore-imaginary-index-smoothed-v1.txt";

datatable=Import[filename,"Table"];

colx=1;

coly=2;

wavemicron=datatable[[All,colx]];

kcc=datatable[[All,coly]];

kccwave=Transpose[{wavemicron,kcc}];

(\* coccolithophore core absorption index wavelenght in micron \*)

kabscc[wave\_]:=Interpolation[kccwave,InterpolationOrder->1][wave]

Produce the real index of refraction of pure water

(\* water real index from water steam standard board. Wavelength in microns, temperature 20 C \*)

fwater[wave\_]:=0.244257733

+9.7463447610^(-3)

-3.7323499610^(-3)(293.15/273.15)

+2.6867847210^(-4)(293.15/273.15)(wave/0.589)^2

+1.58920570 10^(-3)/(wave/0.589)^2

+2.45934259 10^(-3)/((wave/0.589)^2-0.229202^2)

+0.900704920/((wave/0.589)^2-5.432937^2)

-1.66626219 10^(-2)

nwater[wave\_]:=Sqrt[(1+2 fwater[wave])/(1-fwater[wave])]

Produce the orientation averaged real index of calcite relative to water

(\* calcite index from Gosh for both ordinary and extraordinary. Wavelenght in microns \*)

fcalo[wave\_]:=0.73358749+0.96464345 wave^2/(wave^2-1.94325203 10^(-2))+1.82831454 wave^2/(wave^2-120.)

ncalo[wave\_]:=Sqrt[1+fcalo[wave]]

fcale[wave\_]:=0.35859695+0.82427830 wave^2/(wave^2-1.066889543 10^(-2))+1.4429128 wave^2/(wave^2-120.)

ncale[wave\_]:=Sqrt[1+fcale[wave]]

(\* compute indices relative to water for liths \*)

nlitho[wave\_]:=ncalo[wave]/nwater[wave]

nlithe[wave\_]:=ncale[wave]/nwater[wave]

(\* produce a mean orientation averaged extraordinary index assuming random orientation of the optical birefringent axis \*)

nemeansimp[no\_,ne\_]:=no/2(1+(ne^2/no^2)/Sqrt[1-ne^2/no^2]

Log[no/ne(1+Sqrt[1-ne^2/no^2])])

(\* mean exteordinary index \*)

nlithebar[wave\_]:=nemeansimp[nlitho[wave],nlithe[wave]]

The final result for the mean index for random calcite orientation is:

(\* mean sum of ordinary and extraordinary index \*)

nemeantot[no\_,ne\_]:=no/2 +nemeansimp[no,ne]/2

nlithbar[wave\_]:=nemeantot[nlitho[wave],nlithe[wave]]

Produce the basic reflection coefficients

(\* basic reflective total scattering efficiency (front surface formulas only \*)

omperp[n\_]:=(3n+1)(n-1)/(3 (n+1)^2) = omega\_perp

ompar[n\_]:=(1/((n^2+1)^3(n^2-1)^2))((n^4-1)(n^6-4n^5-7n^4+4n^3-n^2-1)+2n^2((n^2-1)^4Log[(n-1)/(n+1)]+8 n^2(n^4+1)Log[n])) = omega\_par

om[n\_]:=(omperp[n]+ompar[n])/2 = omega , replaced with omega(n)

(\* basic reflective backscatter efficiency formulas for smooth surfaces \*)

ombperp[n\_]:=(3n^4-16n^3+12n^2-1+2(2n^2-1)^(3/2))/(6 (n^2-1)^2)

ombpar[n\_]:=((3-Log[16])+(37/40)((n-1)/(n+1)))ombperp[n]

omb[n\_]:=(ombperp[n]+ombpar[n])/2, replaced with omegab(n)

(\* use simple formulas with no absorption since applicable to calcite \*)

rb[n\_]:=(1/2)Abs[(n-1)/(n+1)]^2

omlith[nl\_]:=4 om[nl]

omblith[nl\_]:=4 omb[nl]

omblithrough[nl\_]:=(5/6) omlith[nl]

Produce list of geometry parameters for Young coccolith.

(\* coccolith measurement from Jeremy Young \*)

(\* there are 34 wedges/rays \*)

(\* dimensions in microns \*)

{"semi-major axis of coccolith center in microns"}

majoraxiscenter=1.7541/2

{"semi-minor axis of coccolith center in microns"}

minoraxiscenter=1.1475/2

{"semi-major axis of coccolith inner wedge edge in microns"}

majoraxisinner=2.3934/2

{"semi-minor axis of coccolith inner wedge edge in microns"}

minoraxisinner=1.7705/2

{"semi-major axis of coccolith outer1 wedge edge in microns"}

majoraxisouter=3.5738/2

{"semi-minor axis of coccolith outer wedge edge in microns"}

minoraxisouter=2.9508/2

{"semi-major axis of complete coccolith in microns"}

majoraxisouter=3.7377/2

{"semi-minor axis of complete coccolith in microns"}

minoraxisouter=3.1147/2

(\* width of inner ring \*)

{"width of inner ring/tube in microns"}

innerringwidth=0.6311/2

{"width of wedge ring in microns"}

wedgeringwidth=1.1803/2

{"width of outer ring in microns"}

outerringwidth=0.1639/2

{" bo/ao complete semi-minor/complete semi-major"}

boao= minoraxisouter/ majoraxisouter

{" ao/bo complete semi-major/complete semi-minor"}

aobo=majoraxisouter/minoraxisouter

{"wr/ao width of outer ring/complete semi-major"}

wrao=outerringwidth/majoraxisouter

{"ww/ao width of wedge ring/complete semi-major"}

wwao=wedgeringwidth/majoraxisouter

{"wt/ao width of tube ring/complete semi-major"}

wtao=innerringwidth/majoraxisouter

{"tl/ao thickness of coccolith/complete semi-major"}

tlao=.2

(\* formulas for the various a/ao and b/bo \*)

arao[wrn\_]:=1-wrn

brao[aob\_,wrn\_]:=1/aob-(wrn)

brbo[aob\_,wrn\_]:=1-(aob wrn)

awao[wrn\_,wwn\_]:=1-(wrn+wwn)

bwao[aob\_,wrn\_,wwn\_]:=1/aob-(wrn+wwn)

bwbo[aob\_,wrn\_,wwn\_]:=1-aob(wrn+wwn)

atao[wrn\_,wwn\_,wtn\_]:=1-(wrn+wwn+wtn)

btao[aob\_,wrn\_,wwn\_,wtn\_]:=1/aob-(wrn+wwn+wtn)

btbo[aob\_,wrn\_,wwn\_,wtn\_]:=1-aob(wrn+wwn+wtn)

(\* obtain the maximum fraction of the surface area of the lith that can become rough \*)

frgmax[aob\_,wrn\_,wwn\_]:=(arao[wrn]brbo[aob,wrn]-awao[wrn,wwn]bwbo[aob,wrn,wwn])

Produce formulas for the perimeter of an ellipse

(\* formula for perimeter of ellipse. note majoraxisouter=ao \*)

h[an\_,bn\_]:=(an-bn)^2/(an+bn)^2

pn[an\_,bn\_]:=(an+bn)(1+3 h[an,bn]/(10+Sqrt[4-3h[an,bn]]))

pe[a\_,b\_,ao\_]:=Pi ao pn[a/ao,b/ao]

Produce general gap shape formula for coccoliths

(\* compute shape dependant backscatter effficiency \*)

(\* formula for generalized gap shape \*)

wcngen[gapshape\_,ngap\_,gapratio\_,wwn\_,wrn\_,aob\_,ao\_,wave\_]:=

(1/(1-gapshape))wwn(1-(wave/(4 Pi ao)) (ngap (1+gapratio)/pn[1-wrn,1/aob-wrn]))

(\* set negative values to zero \*)

(Note Mathematica If function syntax is: If[condition statement, true outcome, false outcome])

wcnabsgenlow[gapshape\_,ngap\_,gapratio\_,wwn\_,wrn\_,aob\_,ao\_,wave\_]:=If[wcngen[gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave]<0,0,wcngen[gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave]]

(\* set values of wc greater than wr to wr \*)

wcnabsgen[gapshape\_,ngap\_,gapratio\_,wwn\_,wrn\_,aob\_,ao\_,wave\_]:=If[wcnabsgenlow

[gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave]>wwn,wwn,wcnabsgenlow[gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave]]

(\* produce major a minor axes normalized to outer axes of elliptical lith \*)

acngen[gapshape\_,ngap\_,gapratio\_,wwn\_,wrn\_,aob\_,ao\_,wave\_]:=1-(wrn+wcnabsgen

[gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave])

bcngen[gapshape\_,ngap\_,gapratio\_,wwn\_,wrn\_,aob\_,ao\_,wave\_]:=1-aob(wrn+wcnabsgen

[gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave])

(\* produce rough scattering surface fraction \*)

frggen[gapshape\_,ngap\_,gapratio\_,wwn\_,wrn\_,aob\_,ao\_,wave\_]:=(arao[wrn]brbo[aob,wrn]-acngen[gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave] bcngen[gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave])

(\* produce lith backscattering efficiency \*)

qbblithgen[index\_,gapshape\_,ngap\_,gapratio\_,wwn\_,wrn\_,aob\_,ao\_,wave\_]:=omblith[index](1-frggen[gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave])

+omblithrough[index]frggen[gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave]

Produce Qbb test cases to match with Mathematica code

Plot[{qbblithgen[1.2,0.001,34,0.666,wwao,wrao,aobo,1.0,waveao/1.333],

qbblithgen[1.2,0.25,34,0.666,wwao,wrao,1.2,1.0,waveao/1.333],

qbblithgen[1.2,0.5,34,0.666,wwao,wrao,1.2,1.0,waveao/1.333],

qbblithgen[1.2,0.75,34,0.666,wwao,wrao,1.2,1.0,waveao/1.333],

qbblithgen[1.2,0.90,34,0.666,wwao,wrao,1.2,1.0,waveao/1.333]},

{waveao,0,1.}

(\* produce cross-section for bb of individual lith, tln is the lith thickness normalised to the semi-major axis \*)

crossgeo[tln\_,a\_,b\_]:=(Pi a b/2)(1+(tln/2)(a/b) pn[1,b/a])

crossbbgen[index\_,tln\_,gapshape\_,ngap\_,gapratio\_,wwn\_,wrn\_,aob\_,ao\_,wave\_]:=qbblithgen[index,gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave]crossgeo[tln,ao,ao/aob]

Produce the mean backscattering cross-section for second order gamma size distribution

(\* produce mean cross-section for mu=2,nu=2m gamma distribution \*)

(\* do integral over m=2,nu=2 gamma distribution \*)

psize22norm[beta\_,rmin\_,r\_]:=UnitStep[r-rmin](4beta^(3/2)/Sqrt[Pi]) (r-rmin)^2

Exp[-beta(r-rmin)^2]

{"standard deviation of the semi-major axis size distribution in microns=sig"}

{"mean of the semi-major axis size distribution in microns=mean"}

Produce parameters of the gamma distribution as a function of Standard deviation and mean

betamusig22[sig\_]:=(3 Pi-8)/(2 Pi sig^2)

rminmusig22[mu\_,sig\_]:=mu-sig/Sqrt[(3Pi-8)/8]

Produce mean backscattering cross-section by integrating over size second order distribution

crossbb22meanlith[sig\_,mu\_,index\_,tln\_,gapshape\_,ngap\_,gapratio\_,wwn\_,wrn\_,aob\_,wave\_]:=NIntegrate[crossbbgen[index,tln,gapshape,ngap,gapratio,wwn,wrn,aob,ao,wave]

psize22norm[betamusig22[sig],rminmusig22[mu,sig],ao],{ao,0,Infinity}]

Produce test data to check against Mathematica code

(\* plot mean backscatter cross-section for second order gamma distribution m=2, nu=2 of semi-major axis,std=0.15,mean=1.25,1.5,1.75 microns, tln=0.20,gapshape 0.50 \*)

Plot[{crossbb22meanlith[0.15,1.25,1.2,.20,0.50,34,0.666,wwao,wrao,aobo,waveao/1.333],

crossbb22meanlith[0.15,1.5,1.2,.20,0.50.34,0.666,wwao,wrao,aobo,waveao/1.333],

crossbb22meanlith[0.15,1.75,1.2,.20,0.50,34,0.666,wwao,wrao,aobo,waveao/1.333]},

{waveao,.3,1.}]