Subject: Mathematical manipulations
Posted by khromova2001 on Fri, 06 Feb 2004 15:50:50 GMT
View Forum Message <> Reply to Message

Hi everybody!
Me is already couple of month in IDL and already understood that here exist huge amount of stones under the water-:(
I am currently working on the multiple scattering of X-rays in 3D.
Everythin seems was O.K. But...
The initial parameters ray direction and initial position, spheres centers and radiuses. Using Snell's law, vector product of unit vectors(incident direction and normal to the surface or refracted direction and normal) I found incident and refracted angles.
I need also to find angle between refracted ray direction and initial direction(all rays was parallel to OX [1,0,0]. Let's say t[0] x-coordinate of refraction direction is 1.0000. Acos(1.0000)=0.0000.
But me recieving 2.7905781e-006!!!!!!!
What does it mean, may be somebody may explaine me where to check?.
May be me using not correctly mathematics in IDL.
P.S.Loops I will eliminate in future
Thank you
Sincerely, Anastasiya

Part of Programme look like:

```
while p lt rays do begin
while (x_source[0,p] lt xsize) and (x_source[1,p] lt ysize) and
(x_source[2,p] lt zsize) do begin

    position[0,*]=x_source[0,p];initial position of the ray
    position[1,*]=x_source[1,p]
    position[2,*]=x_source[2,p]

    values[0:1,*]=roots60(C1,position,direction)
    j=where((values[0,*] ne 0.D) and (values[1,*] ne 0.D) and
(values[0,*] gt 0.D) and (values[1,*] gt 0.D),count)
    if count eq 0 then goto, skip_it

    min_displ=min(values[0:1,j])
    k=where((values[0,*] eq min_displ) or (values[1,*] eq
min_displ),count1)
     ent_point=position+[min_displ*direction[0],min_displ*directi on[1],min_displ*direction[2]]

   m1=k[0]+m1
   qu_norm=coords_sorted(3,m1)
   normal=ent_point-C[0:2,m1]
   ni=normal/norm(normal)
```

```
   d=direction
   n=ni


     X=[[d[1],d[2]],$
       [n[1],n[2]]]

     Y=[[d[2],d[0]],$
       [n[2],n[0]]]

     Z=[[d[0],d[1]],$
       [n[0],n[1]]]

     minor1=determ(X,/check)
     minor2=determ(Y,/check)
     minor3=determ(Z,/check)

     vector_product=sqrt(minor1^2+minor2^2+minor3^2)
     incident_angle=asin(vector_product/(norm(d)*norm(n)))

     refraction_angle1=asin(k1*sin(incident_angle)/(1.0D -9.23*10D-7))

     t[0]=k1/(1.0D -9.23*10D-7)*d[0]-(cos(refraction_angle1)-k1/(1.0D
-9.23*10D-7)*cos(incident_angle))*n[0]
     t[1]=k1/(1.0D -9.23*10D-7)*d[1]-(cos(refraction_angle1)-k1/(1.0D
-9.23*10D-7)*cos(incident_angle))*n[1]
     t[2]=k1/(1.0D -9.23*10D-7)*d[2]-(cos(refraction_angle1)-k1/(1.0D
-9.23*10D-7)*cos(incident_angle))*n[2]

print,acos(t[0])


     x_source[*,p]=ent_point

...
..
....
...
..
 ..
endwhile
p=p+1
.....
endwhile
```