Reunion IMNC/SHFJ
24 Fevrier 2012

Dielectric-dielectric surface

```
<surface name="MySurface" type="dielectric_dielectric" sigmaalpha="20.0" finish="ground">
<propertiestable>
    <propertyvector name="SPECULARLOBECONSTANT" energyunit="eV">
      <ve energy="2.34" value="0.0"/>
      <ve energy="1.97" value="0.0"/>
    </propertyvector>
      <propertyvector name="SPECULARSPIKECONSTANT" energyunit="eV">
      <ve energy="2.34" value="0."/>
      <ve energy="1.97" value="0."/>
    </propertyvector>
    <propertyvector name="BACKSCATTERCONSTANT" energyunit="eV">
      <ve energy="2.34" value="0."/>
      <ve energy="1.97" value="0."/>
    </propertyvector>
<!--      <propertyvector name="REFLECTIVITY" energyunit="eV">
      <ve energy="2.34" value="1.0"/>
      <ve energy="1.97" value="1.0"/>
    </propertyvector>-->
  </propertiestable>
</surface>
```

Reflection about the normal of a microfacet

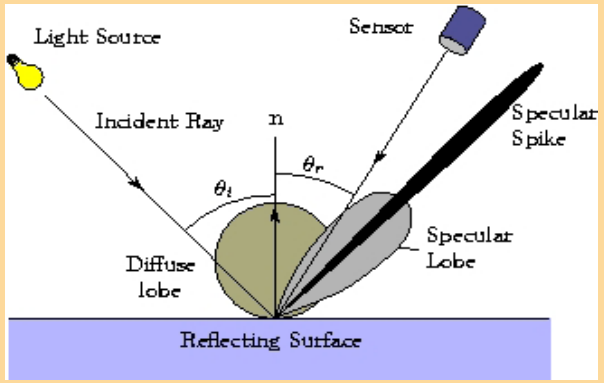
Reflection about the AVERAGE surface normal

Several reflections with the ultimate result of exact back-scatter

Example of 100% Lambertian Reflection

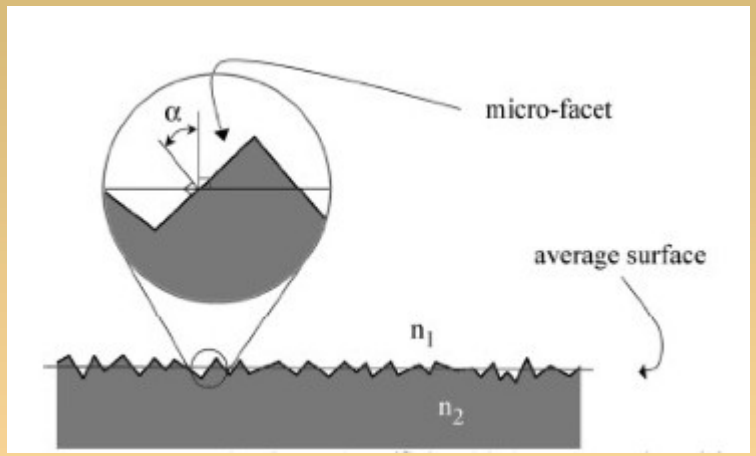
- 4 parameters: Reflectivity, Specular Lobe, Specular Spike, Back-scatter (Lambertian is implicit)
- Default Reflectivity is 1.
- Surface finish: polished and ground
- Polished interface: no surface needs to be specified

  (only 2 media Rindex)



- Ground interface: If not 100% Lambertian (diffuse), you can simulate specular and ultimate back-scatter.
- Not all rough surfaces are perfect lambertian reflectors, but it is a good approximation if surface characteristics are unknown (Geant4 Developper guide).

SigmaAlpha dependence

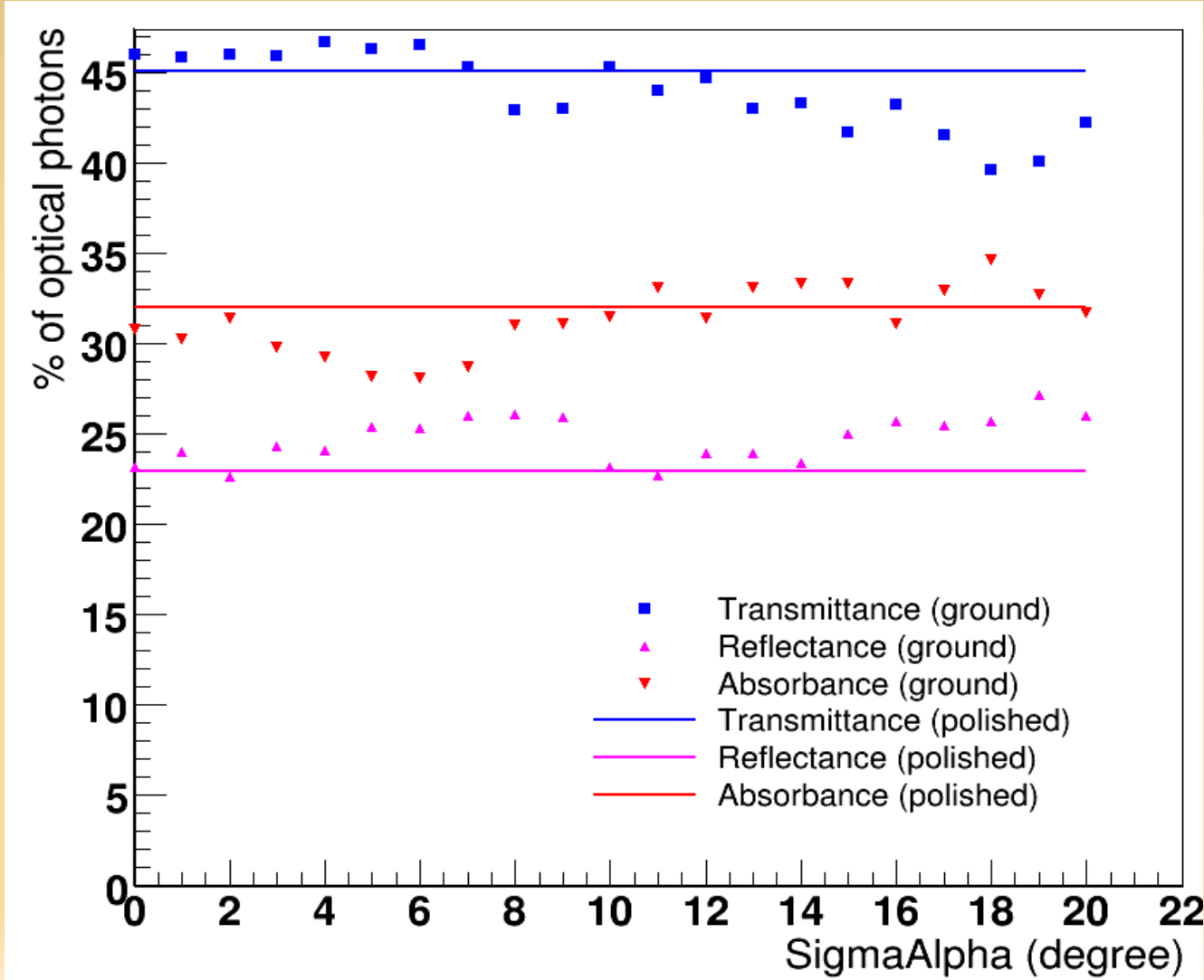


The surface normals micro-facets are distributed around the average surface normal, following a normal distribution with standard deviation:

σ_α

Difference is ~ 5%

5.5% T
4.3% R
3.9% A



hgate Physics Models

- Version 3 du code hGate (J. Bert)
- Physics Models for EM: Standard and Livermore (low energy electron and gamma)
  - Livermore models for γ, e- below 1 GeV
  - Standard models above 1 GeV
- Standard Compton scattering: interaction of a high energy photon with an electron ==> Klein-Nishina formula: differential cross section.
- Standard Photo Electric Effect (ejection of an electron from a material after a photon has been absorbed by that material): Parameterized formula (Biggs et al – Sandia Tables)

Livermore Low Energy Model

- Based on the Livermore database (experimental data parameterisations) developed by the Lawrence Livermore National Laboratory.
- Rayleigh Scattering: Angular distribution $\Phi(E,\theta)=[1+\cos 2(\theta)]\cdot F2(q)$, where F(q) is the energy-dependent form factor (LLNL database).
- Compton Scattering: Energy distribution of the scattered photon according to Klein-Nishina formula multiplied by scattering functions (LLNL database).
- PhotoElectric Effect: Subshell from which the electron is emitted is selected according to the cross sections of the sub-shells, transition probabilities from LLNL database.

v03_hgate_livermore_standard: Main Code

- mc_main.cu
  - Number of required simulated particles
  - Physics list choice = Rayleigh, Compton, PhotoElectric
  - Water box is built
  - Physics model choice = Standard or Livermore
    - NB: there is no Standard Model Rayleigh.
  - Woodcock tracking (also called delta tracking):
    - Material cross-sections vary within a geometric region
    - One common approach: Breaking the flight path into short segments within which the cross-section is assumed constant. ==> expensive calculations.
    - Delta-tracking approach: Uses a fictitious cross-section Σmax (the largest value expected during the particle flight). A trial flight distance is sampled using that cross section and the particle is moved that distance.
    - P= Σ(s) / Σmax is determined. With probability P, the flight distance is accepted; with probability 1-P, it is rejected, and the entire procedure is repeated.

v03_hgate_livermore_standard: Functions

- mc_fun.cu
  - Functions from CLHEP: loglog interpolation, linear interpolation, deflect particle (due to Compton of Rayleigh scatter)
  - Random Number generator BRENT xor256
  - Livermore: Rayleigh, Compton and PhotoElectric cross sections per Atom
  - Standard: Compton and PhotoElectric cross sections per Atom
  - Scatter Factors (Livermore): Rayleigh and Compton
  - Functions: Compton scatter (STD and Livermore), Rayleigh Scatter (Livermore only), Total Compton and Photoelectric cross sections for a given material (STD and Livermore), Rayleigh Total cross section for a given material (Livermore only)
  - Compton and Rayleigh Element Selector: Z material mixture
  - Sources: Ficticious tracking (STD and Livermore), Interaction (physics and new particle direction), Phase space (which particle will be reloaded in stack), Open cross sections Tables.

v03_hgate_livermore_standard: Materials, Cross-Sections, Data

List of index materials

0 – Air
1 – Water
2 – Body
3 – Lung
4 – Breast
5 – Heart
6 – SpineBone
7 - RibBone
8 – Intestine
9 – Spleen
10 – Blood
11 – Liver
12 – Kidney
13 – Brain
14 - Pancreas

Data files (binary)

Compton (x-sec and SF) Livermore

PhotoElectric (x-sec)

Rayleigh (x-sec and SF) Livermore

mc_cst.cu

Constants, Materials

Some cross-sections

Scatter Factors (Compton, Rayleigh)

PhotoElectric:

Atomic number (Z)/ Atomic Mass (A)

Ionization Potentials

Sandia Tables (Z=1 to 30)

Atomic Number Density for each element of materials

Atom Bind Energy

Doppler Profiles for Compton ...

Notre status @ CCRT

- Le groupe DSV n'a pas d'acces au GPU sur la machine Titane (48 serveurs GPU Tesla Nvidia destines aux utilisateurs GENCI)
- Contact: Christine Menache
  - Titane-GPU: demande Dari pour le 2ème semestre
  - Curie-GPU: demande Preparatory Access de type B qui dure 6 mois et donne acces a ~60 000h GPU (deadline 01.03.12).
- BACKUP plan: Utilisation de la machine Cesium (chaque noeud est equipe d'une carte GPU Nvidia) [tests uniquement]

Environnement et Compilation

- Environnement de travail CUDA sur Cesium

```
#!/bin/bash

#Chargement des modules GPU
module load cuda/4.0

#Variable MPI_HOME
export MPI_HOME=`which mpicc | sed 's/\/bin\/mpicc//g'
```

- Compilation code

```
nvcc -o mc_main mc_main.cu -arch compute_20 -use_fast_math
```

Aucun Probleme!

Execution du code sur Cesium

- Les calculs sur les cartes graphiques se font en batch uniquement

```
#!/bin/bash
#MSUB -r slinky
#MSUB -N 1    # nombre de noeuds a allouer
#MSUB -n 1    # nombre de processeurs a allouer
##MSUB -T 1800
#MSUB -q test
#MSUB -o slinky.o
#MSUB -e slinky.e
set -x
cd $BRIDGE_MSUB_PWD
#mpirun -np 1 /applications/cuda-4.0/sdk/C/bin/linux/release/deviceQuery
mpirun -np 1 ./mc_main 10
```

argument: Random Seed

```
340460.cesium22     slinky    vcuplov         test@cesium22          0 PEN          0     1800        0     1800        0    7168     0     1
```

```
Tot particles simulated: 1000 - 1000
Running time is 0.042941 s  - tot export 0
```

- Differente architecture: Fermi (2.0) (julien) versus Quadro (1.3)
- memoire insuffisante? Non :(
- en contact (permanent) avec le prof de la formation GPU au CCC :)

Resultats

- Cesium ne fonctionne pas bien en batch
- Le code tourne (6s) sur Titane
- Pour continuer à travailler sur Cesium, il faut utiliser visuportal (bureau virtuel) et travailler en interactif.