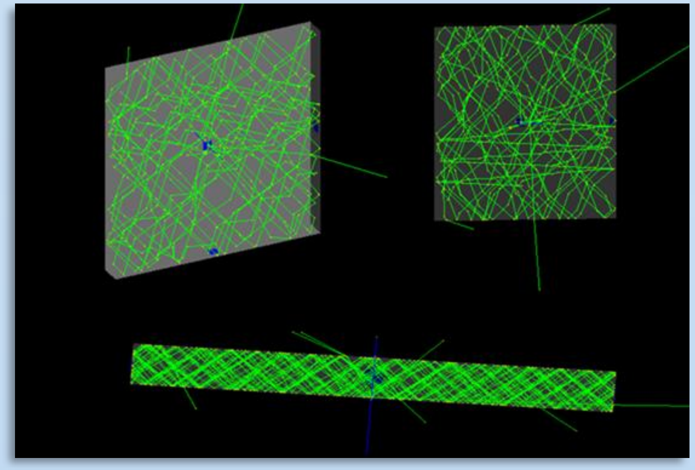
Simulation of optical photon propagation for generic scintillator-based detectors

Lecture 5

Optical physics



Davide Serini
davide.serini@ba.infn.it

Introduction

- This part of the course provides an overview of Geant4:
  - its capabilities
  - how they can be used in an experimental simulation application (we will focus on the optical simulations)
- Geant4 is a complex and powerful toolkit
  - Impossible to teach (and learn) everything in few days and also our own expertise is not infinite

- This course would provide you with a global vision of Geant4 and a method for "how to use it"
  - Finding **your way** in the complexity of Geant4 is not easy
  - Use the Geant4 User Documentation and the "**BookForApplicationDevelopers**"

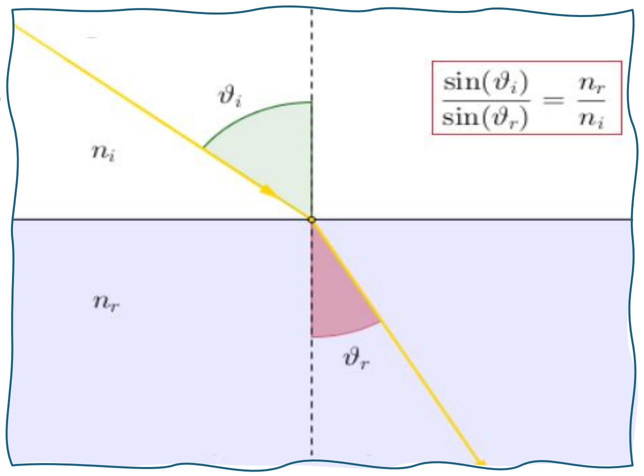
Geant4 web site: http://cern.ch/geant4

Optical processes: generical properties

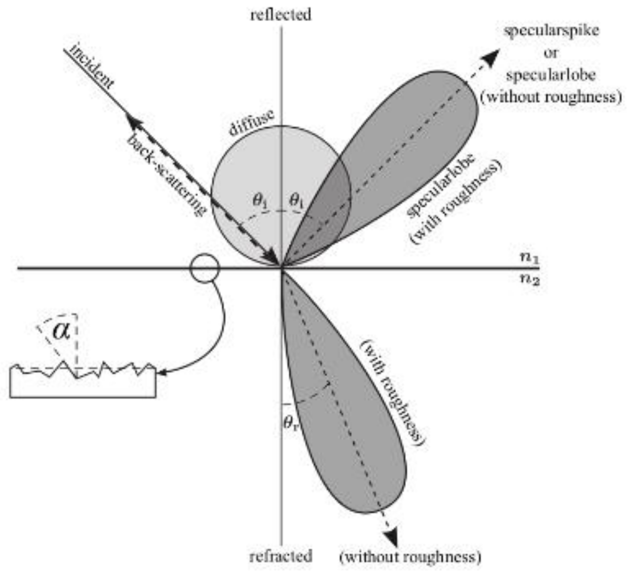
- Bulk
  - Optical properties related to the single material

- Scintillation emission
- Absorption length
- Refractive index
- Scintillation Yield
- Birks' constant
- …

- Boundary
  - Optical properties related to the photon behaviour at the surface between two materials
  - **If not defined the default behaviour is the Snell's Law**

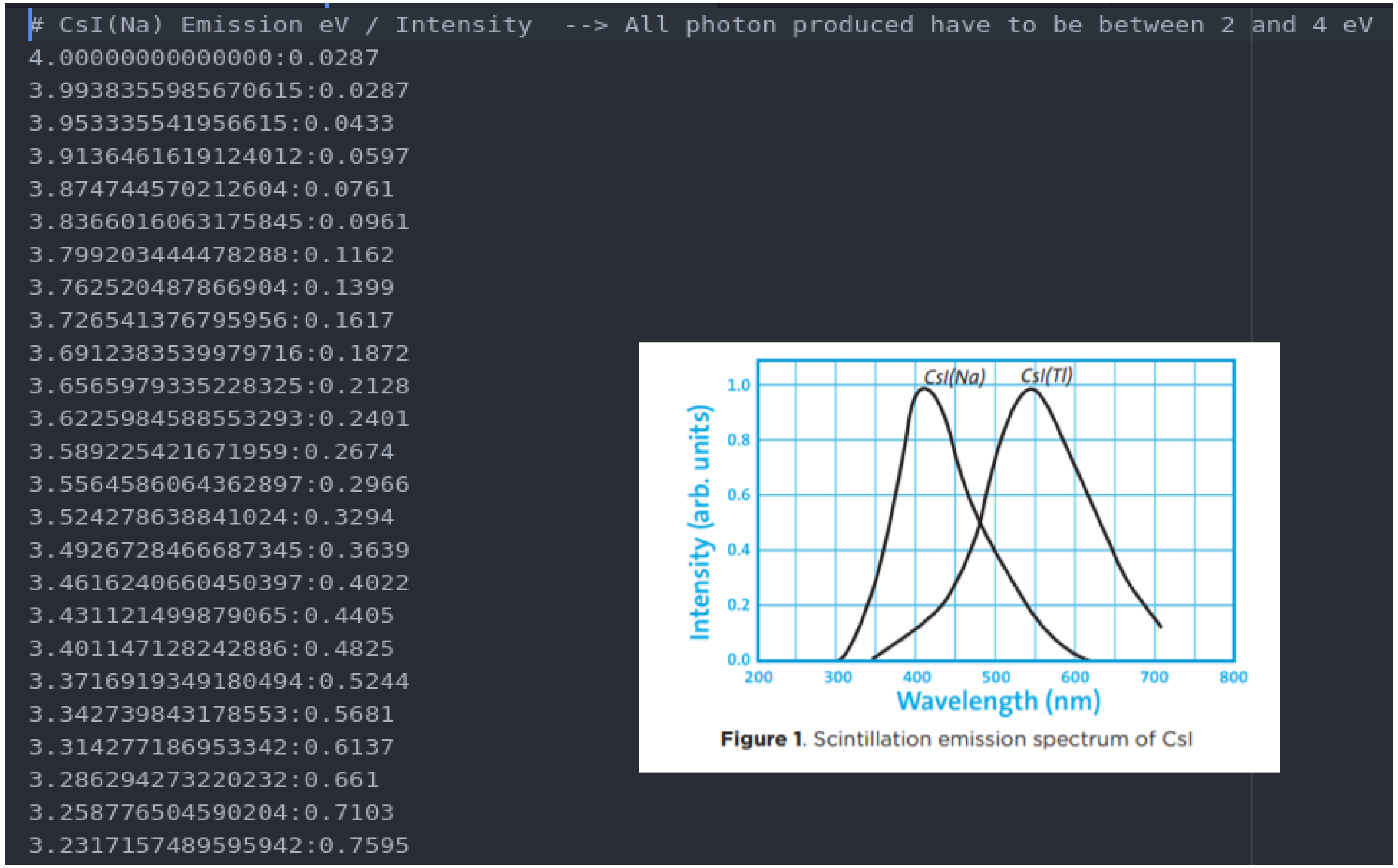


ϑ_i
n_i
$\frac{\sin(\vartheta_i)}{\sin(\vartheta_r)} = \frac{n_r}{n_i}$
n_r
ϑ_r



A complete Reference of the optical properties is available in the Application Developers Guide, Chapter 5.2 (Physics Process)

Scintillator optical bulk properties



```
# CsI(Na) Emission eV / Intensity  --> All photon produced have to be between 2 and 4 eV
4.000000000000000:0.0287
3.9938355985670615:0.0287
3.953335541956615:0.0433
3.9136461619124012:0.0597
3.874744570212604:0.0761
3.8366016063175845:0.0961
3.799203444478288:0.1162
3.762520487866904:0.1399
3.726541376795956:0.1617
3.6912383539979716:0.1872
3.6565979335228325:0.2128
3.6225984588553293:0.2401
3.589225421671959:0.2674
3.5564586064362897:0.2966
3.524278638841024:0.3294
3.4926728466687345:0.3639
3.4616240660450397:0.4022
3.431121499879065:0.4405
3.401147128242886:0.4825
3.3716919349180494:0.5244
3.342739843178553:0.5681
3.314277186953342:0.6137
3.286294273220232:0.661
3.258776504590204:0.7103
3.2317157489595942:0.7595
```

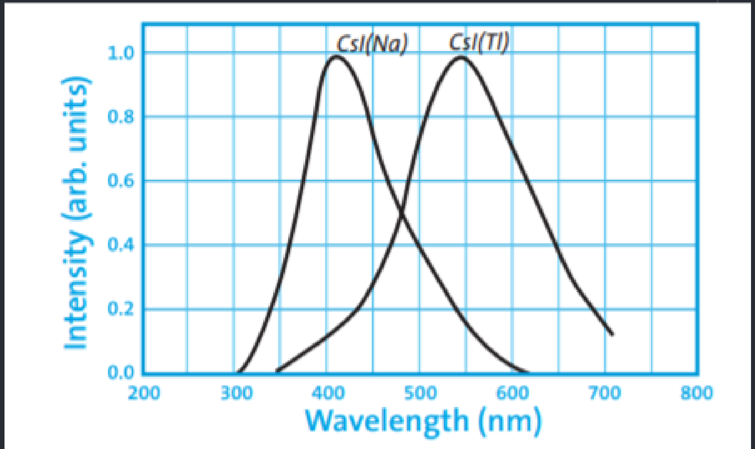
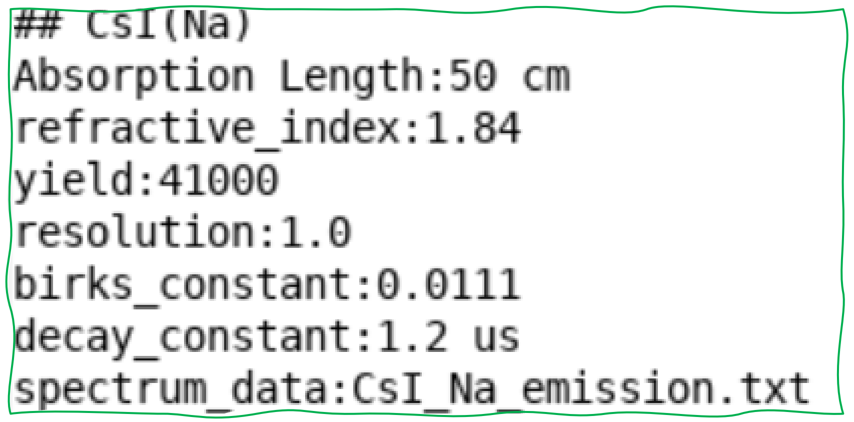


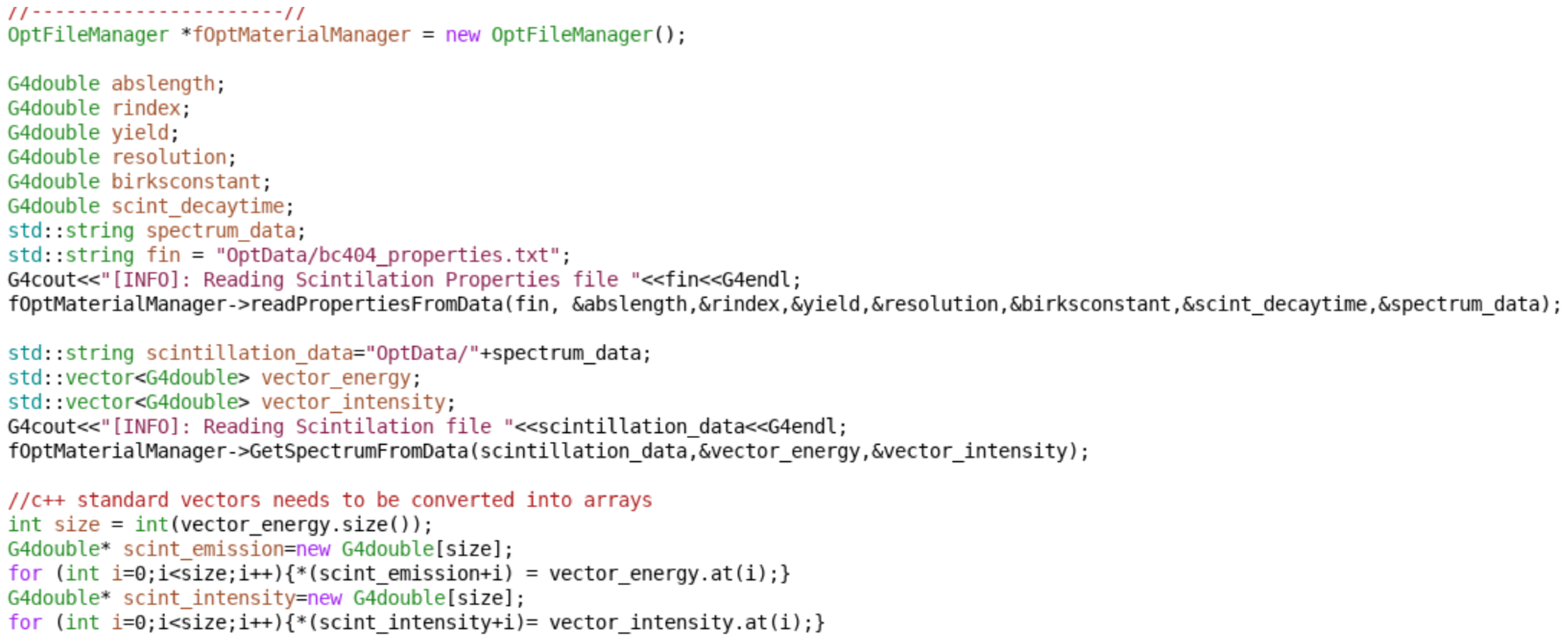
Figure 1. Scintillation emission spectrum of CsI



```
## CsI(Na)
Absorption Length:50 cm
refractive_index:1.84
yield:41000
resolution:1.0
birks_constant:0.0111
decay_constant:1.2 us
spectrum_data:CsI_Na_emission.txt
```

A custom class to read out the main properties from file: OptFileManager

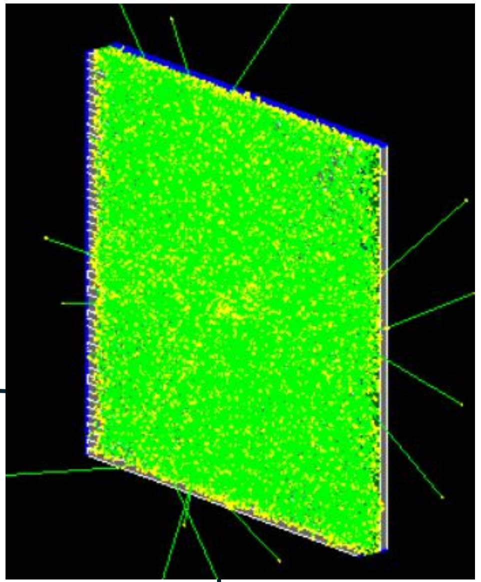
Read data from files using your OptFileManager class



```
//---------------------//
OptFileManager *fOptMaterialManager = new OptFileManager();

G4double abslength;
G4double rindex;
G4double yield;
G4double resolution;
G4double birksconstant;
G4double scint_decaytime;
std::string spectrum_data;
std::string fin = "OptData/bc404_properties.txt";
G4cout<<"[INFO]: Reading Scintilation Properties file "<<fin<<G4endl;
fOptMaterialManager->readPropertiesFromData(fin, &abslength,&rindex,&yield,&resolution,&birksconstant,&scint_decaytime,&spectrum_data);

std::string scintillation_data="OptData/"+spectrum_data;
std::vector<G4double> vector_energy;
std::vector<G4double> vector_intensity;
G4cout<<"[INFO]: Reading Scintilation file "<<scintillation_data<<G4endl;
fOptMaterialManager->GetSpectrumFromData(scintillation_data,&vector_energy,&vector_intensity);

//c++ standard vectors needs to be converted into arrays
int size = int(vector_energy.size());
G4double* scint_emission=new G4double[size];
for (int i=0;i<size;i++){*(scint_emission+i) = vector_energy.at(i);}
G4double* scint_intensity=new G4double[size];
for (int i=0;i<size;i++){*(scint_intensity+i)= vector_intensity.at(i);}
```

Scintillator optical bulk properties

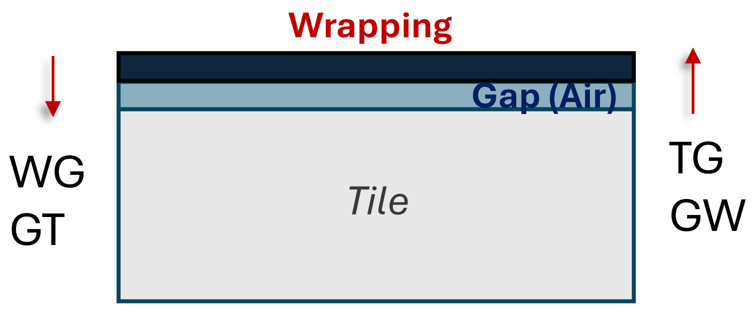
- Two kind of properties:
  - Photon wavelength dependent
    - *RINDEX,ABSLENGTH,SCINTCOMPONENT1 (2,3)*
  - Constants
    - *SCINTILLATIONYIELD«, RESOLUTIONSCALE,*
    - *SCINTILLATIONTIMECONSTANT1 (2,3)*



```
G4MaterialPropertiesTable* mpt_Scintillator =
                              new G4MaterialPropertiesTable();
 mpt_Scintillator->AddProperty("RINDEX",scint_emission,RIndex_Scintillator,size);
mpt_Scintillator->AddProperty("ABSLENGTH",scint_emission,AbsorptionLength_Scintillator,size);
mpt_Scintillator->AddProperty("SCINTILLATIONCOMPONENT1",scint_emission,scint_intensity,size);
mpt_Scintillator->AddConstProperty("SCINTILLATIONYIELD",yield/MeV);
mpt_Scintillator->AddConstProperty("RESOLUTIONSCALE",resolution);
mpt_Scintillator->AddConstProperty("SCINTILLATIONTIMECONSTANT1",scint_decaytime);
scint_mat->SetMaterialPropertiesTable(mpt_Scintillator);
```

Boundary surfaces

```
G4OpticalSurface* WrappingSurface = new G4OpticalSurface("Wrapping Surface");
WrappingSurface→ SetModel(unified);
WrappingSurface→ SetType(dielectric_metal);
WrappingSurface→ SetFinish(ground);
G4double sigma_alpha = 0.1;
WrappingSurface→ SetSigmaAlpha(sigma_alpha);
new G4LogicalBorderSurface("Wrapping
Surface",pysicalScint,pysicalWrapping,WrappingSurface);
```

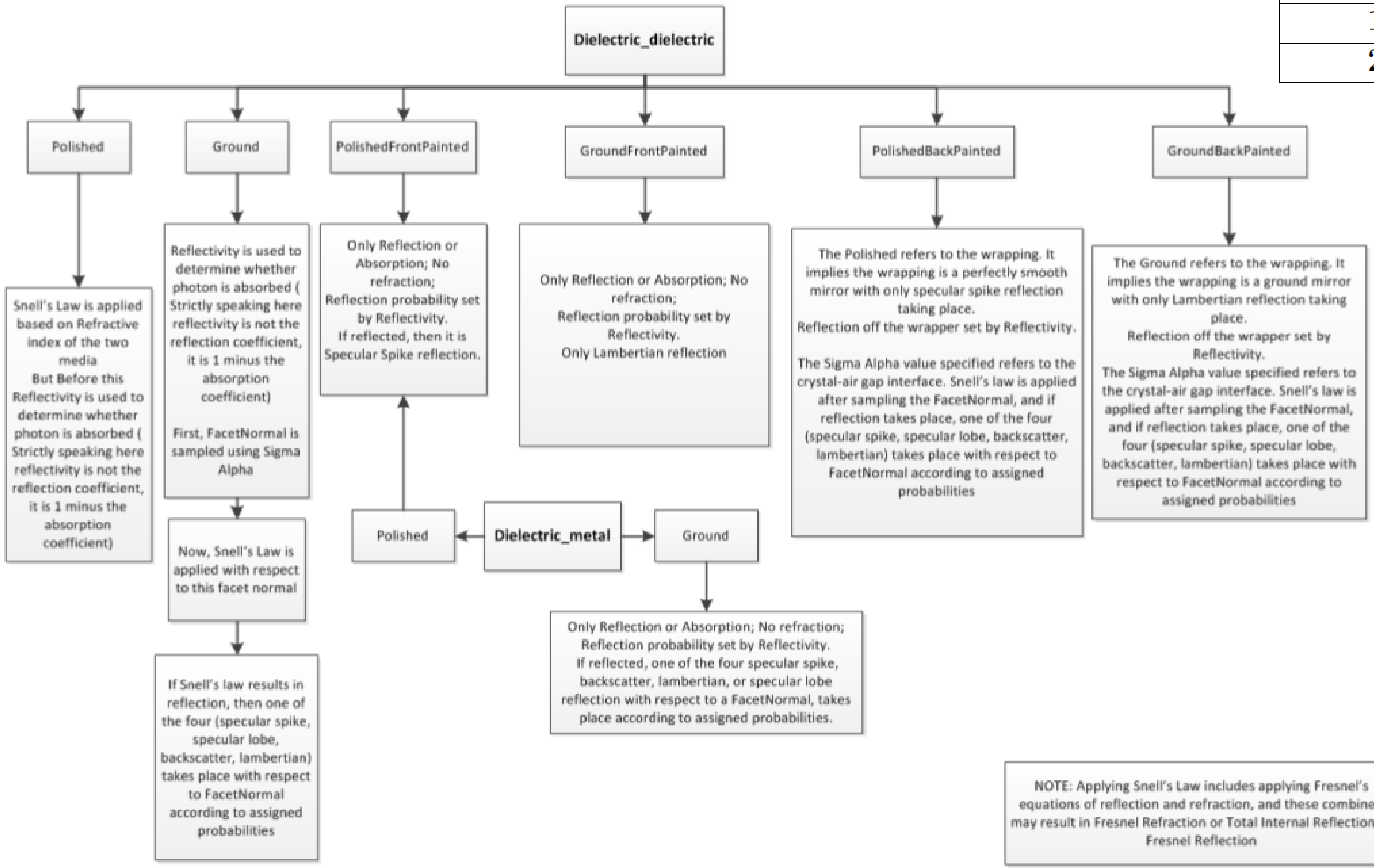


The boundary properties are unidirectional!

- The **Finish** add information about the roughness of the surface and add other properties like coating influence.
- The **sigma alpha** represents the roughness of the surface in terms of the standard deviation of the Gaussian distribution (around zero) of the angle between the local, microscopic surface and the overall mean surface
- **Reflectivity** is the (1-Absorption at the surface) where the Absorption at the surface is the probability that an optical photons is absorbed at surface
- By default, REFLECTIVITY equals 1 and TRANSMITTANCE equals 0. At a surface interaction, a random number is chosen r
  - *If the r > REFLECTIVITY + TRANSMITTANCE* ➔ *the photon is absorbed.*
  - *elif REFLECTIVITY + TRANSMITTANCE < r < REFLECTIVITY* ➔ *the photon is transmitted.*
  - *Otherwise* ➔ *usual calculation of scattering takes place.*

UNIFIED MODEL FOR OPTICAL SURFACES

Command	Effect
0	dielectric-dielectric
1	dielectric-metal
2	Type LUT



Dielectric_dielectric

Polished

Snell's Law is applied based on Refractive index of the two media
But Before this Reflectivity is used to determine whether photon is absorbed (Strictly speaking here reflectivity is not the reflection coefficient, it is 1 minus the absorption coefficient)

Ground

Reflectivity is used to determine whether photon is absorbed (Strictly speaking here reflectivity is not the reflection coefficient, it is 1 minus the absorption coefficient)

First, FacetNormal is sampled using Sigma Alpha

Now, Snell's Law is applied with respect to this facet normal

If Snell's law results in reflection, then one of the four (specular spike, specular lobe, backscatter, lambertian) takes place with respect to FacetNormal according to assigned probabilities

PolishedFrontPainted

Only Reflection or Absorption; No refraction;
Reflection probability set by Reflectivity.
If reflected, then it is Specular Spike reflection.

GroundFrontPainted

Only Reflection or Absorption; No refraction;
Reflection probability set by Reflectivity.
Only Lambertian reflection

PolishedBackPainted

The Polished refers to the wrapping. It implies the wrapping is a perfectly smooth mirror with only specular spike reflection taking place.
Reflection off the wrapper set by Reflectivity.

The Sigma Alpha value specified refers to the crystal-air gap interface. Snell's law is applied after sampling the FacetNormal, and if reflection takes place, one of the four (specular spike, specular lobe, backscatter, lambertian) takes place with respect to FacetNormal according to assigned probabilities

GroundBackPainted

The Ground refers to the wrapping. It implies the wrapping is a ground mirror with only Lambertian reflection taking place.
Reflection off the wrapper set by Reflectivity.
The Sigma Alpha value specified refers to the crystal-air gap interface. Snell's law is applied after sampling the FacetNormal, and if reflection takes place, one of the four (specular spike, specular lobe, backscatter, lambertian) takes place with respect to FacetNormal according to assigned probabilities

Dielectric_metal

Polished

Ground

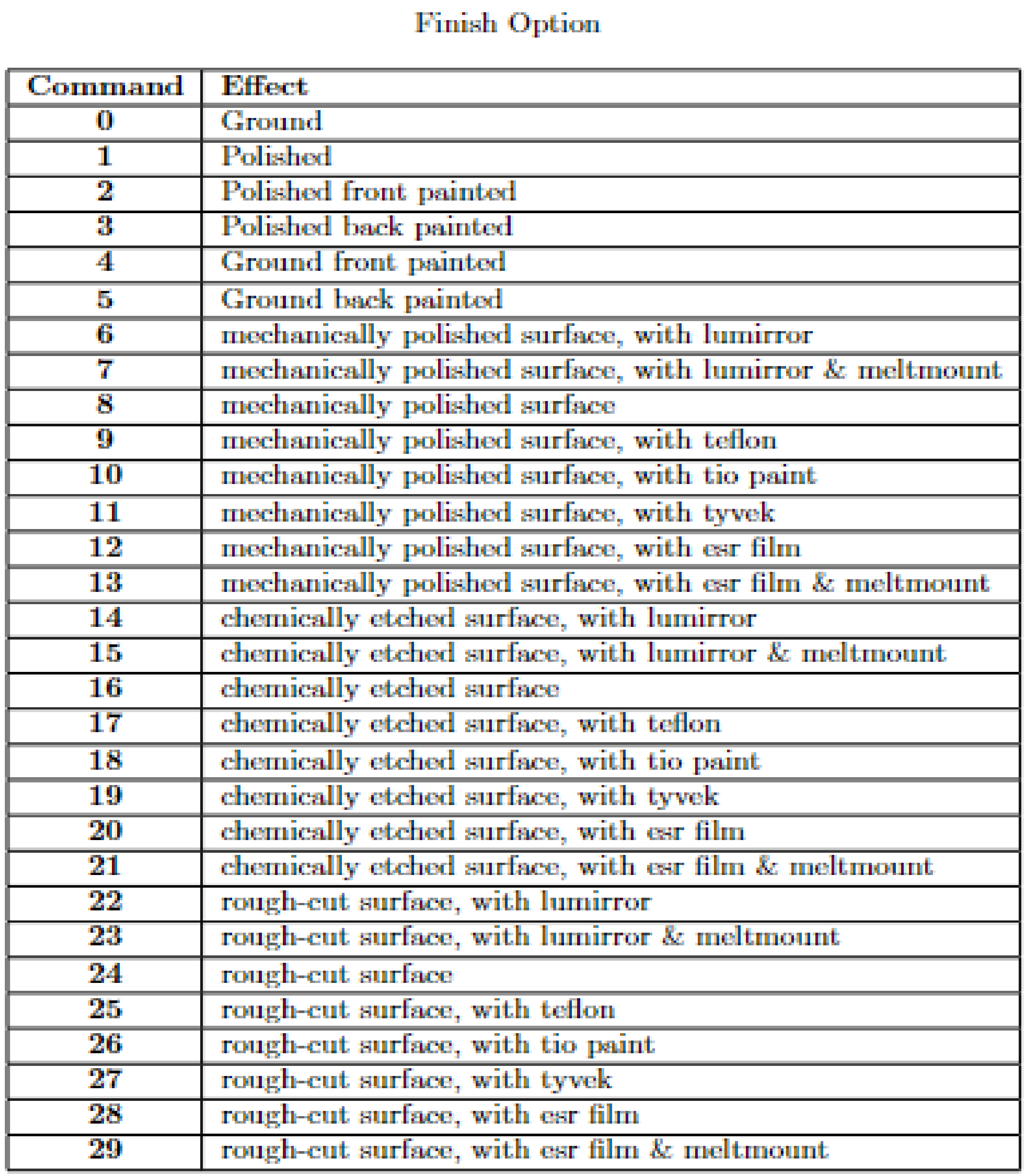
Only Reflection or Absorption; No refraction;
Reflection probability set by Reflectivity.
If reflected, one of the four specular spike, backscatter, lambertian, or specular lobe reflection with respect to a FacetNormal, takes place according to assigned probabilities.

NOTE: Applying Snell's Law includes applying Fresnel's equations of reflection and refraction, and these combined may result in Fresnel Refraction or Total Internal Reflection or Fresnel Reflection

Dielectric-dielectric
Photons can undergo total internal reflection,refraction or reflection, depending on the photons wavelength, angle of incidence, and the refractive indices on both sides of the boundary.

Dielectric-metal
Photons can be absorbed by the metal or reflected back into the dielectric

Finish options for unified model



Finish Option

Command	Effect
0	Ground
1	Polished
2	Polished front painted
3	Polished back painted
4	Ground front painted
5	Ground back painted
6	mechanically polished surface, with lumirror
7	mechanically polished surface, with lumirror & meltmount
8	mechanically polished surface
9	mechanically polished surface, with teflon
10	mechanically polished surface, with tio paint
11	mechanically polished surface, with tyvek
12	mechanically polished surface, with esr film
13	mechanically polished surface, with esr film & meltmount
14	chemically etched surface, with lumirror
15	chemically etched surface, with lumirror & meltmount
16	chemically etched surface
17	chemically etched surface, with teflon
18	chemically etched surface, with tio paint
19	chemically etched surface, with tyvek
20	chemically etched surface, with esr film
21	chemically etched surface, with esr film & meltmount
22	rough-cut surface, with lumirror
23	rough-cut surface, with lumirror & meltmount
24	rough-cut surface
25	rough-cut surface, with teflon
26	rough-cut surface, with tio paint
27	rough-cut surface, with tyvek
28	rough-cut surface, with esr film
29	rough-cut surface, with esr film & meltmount

Exercise – Example1B

- Adding all the optical properties to your detector materials
- Try to implement different Scintillator models using the custom class OptFileManager and export different emission spectra associated to all the scintillator used
  - Change some properties and try to run different simulations
- Implement a «Perfect Absorber» and a «Mirror» surfaces between the Tile and the Wrapping
  - Perform your simulations changing the surfaces
    - *What happens to the number of photon absorbed?*
    - *What happens to the total track length?*

NB to store some «basic» information I have implemented the ProcessHits function in the SensitiveDetector class (we will talk about sensitive detector in the next lectures)

Check that inside the SensitiveDetector::ProcessHits() there is a boundary check for the photons otherwise if a photon going outside the world it can not be processed (the PostStepPoint doesn't exist)

```
if(fWorldBoundary!=aStep->GetPostStepPoint()->GetStepStatus() && IsAnOpticalPhoton(aStep))
    processPhotonHit(aStep);
```

Links

Blank

```
wget 'https://istnazfisnucl-my.sharepoint.com/:u:/g/personal/serini_infn_it/EX-
1XCKrJdlImqv1wKKZY20BiiMb9b5eAdP1ozxKD2wLyQ?e=Y7DJeZ&download=1' -O Blank.tar.gz
```

Ex0

```
wget 'https://istnazfisnucl-my.sharepoint.com/:u:/g/personal/serini_infn_it/EfRyYDwP7gZJruci7aTx1V4Bhpm-
EaYmroRFsEnDIGLRTQ?e=PICleo&download=1' -O Ex0.tar.gz
```

Ex1

```
wget 'https://istnazfisnucl-
my.sharepoint.com/:u:/g/personal/serini_infn_it/ERnXARL7WtJMhMlfPvs8AW8BukNkJhT4Sytw0Q7yY3r3zA?e=51
DStU&download=1' -O Ex1.tar.gz
```