Geant4 optical physics

Hans Wenzel

LarSoft coordination meeting

31st January 2017

Outline

- Optical Photon Processes in Geant4.
- (LArG4) Requirements.
- Adding and configuring optical physics.
- DetectorConstruction: Adding optical properties.
- Using gdml to describe detector/with optical properties
- Accessing the Nr. of optical (Cerenkov) photons produced in sensitive detector (TrackerSD).
- Some timing results and Summary.

Optical Photon Processes in GEANT4

Optical photon production in Geant4:
- Cerenkov Process.
- Scintillation Process.
- Transition Radiation.

Processes:
- Refraction and Reflection at medium boundaries.
- Bulk Absorption.
- Rayleigh/Mie scattering.
- Wavelength shifting

User has to provide optical properties of material as function of photon momentum (e.g. refraction index, absorption length, scattering length, surface properties, scintillation yield and spectrum…)

Problem: until recently optical photons were always put on the stack → very expensive operation. If you didn't want to track them you had to kill them via G4UserStackingAction. But often you just want to count the optical photons!

(LArG4) Requirements

- Convenient way to add optical physics.
- Convenient way to switch selected optical processes on/off.
- disable stacking.
- Need access to the number of produced photons
- Use splines/functions to provide smooth optical property input (refraction index as function of photon momentum, scintillation spectrum…)
- LArG4 should use proper Geant4 interfaces→ work with Geant4 if something is missing

Will show how this now can be done with G4OpticalPhysics physics list. Thanks to Peter Gumplinger for providing the code.

Caveat: disable stacking is available in geant4.10.3. But Geant4 needs to be patched to make setting available via G4OpticalPhysics (messenger).

Adding and configuring optical physics (main)

```
G4PhysListFactory factory;
 G4VModularPhysicsList* phys = NULL;
 G4String physName = "";
 char* path = getenv("PHYSLIST");
 if (path) {
    physName = G4String(path);
 } else {
    physName = "FTFP_BERT"; // default
 }
 // reference PhysicsList via its name
 if (factory.IsReferencePhysList(physName)) {
    phys = factory.GetReferencePhysList(physName);
 }
 //
 // Now add and configure optical physics
 //
 G4OpticalPhysics* opticalPhysics = new G4OpticalPhysics();
 phys->RegisterPhysics(opticalPhysics);
 opticalPhysics->Configure(kCerenkov, true);
 opticalPhysics->SetCerenkovStackPhotons(false);
 opticalPhysics->Configu
 phys->DumpList();
```

Can also be done via messenger at run time:
/process/optical/defaults/cerenkov/setStackPhotons false

Available Reference Physics lists (at the moment)

```
// Physics List name defined via
// environmental variable
// The following physics lists are available:
//"FTFP_BERT"
//"FTFP_BERT_TRV"
//"FTFP_BERT_ATL"
//"FTFP_BERT_HP"
//"FTFP_INCLXX"
//"FTFP_INCLXX_HP"
//"FTF_BIC"
//"LBE"
//"QBBC"
//"QGSP_BERT"
//"QGSP_BERT_HP"
//"QGSP_BIC"
//"QGSP_BIC_HP"
//"QGSP_BIC_AllHP"
//"QGSP_FTFP_BERT"
//"QGSP_INCLXX"
//"QGSP_INCLXX_HP"
//"QGS_BIC"
//"Shielding"
//"ShieldingLEND"
//"ShieldingM"
//"NuBeam"
```

```
// The following em options are available:
//"",
//"_EMV"
//"_EMX"
//"_EMY"
//"_EMZ"
//"_LIV"
//"_PEN"
//"__GS"
```

Step limiter

```
//set mandatory initialization classes
runManager->SetUserInitialization(new DetectorConstruction());
//StepLimiter?
if (ConfigurationManager::getInstance()->GetstepLimit()) {
    phys->RegisterPhysics(new G4StepLimiterPhysics());
}
runManager->SetUserInitialization(phys);
//set user action classes
runManager->SetUserAction(new PrimaryGeneratorAction());
runManager->SetUserAction(new RunAction());
runManager->SetUserAction(new EventAction());
//runManager->SetUserAction(new HadStackingAction());
```

DetectorConstruction: Adding optical properties

```
targetMaterial = G4NistManager::Instance()->FindOrBuildMaterial("G4_C");
worldMaterial = G4NistManager::Instance()->FindOrBuildMaterial("G4_AIR");
  //
// ------------ Generate & Add Material Properties Table ------------
//
G4double photonEnergy[] ={2.034 * eV, 2.068 * eV, 2.103 * eV, 2.139 * eV,
   2.177 * eV, 2.216 * eV, 2.256 * eV, 2.298 * eV,
   2.341 * eV, 2.386 * eV, 2.433 * eV, 2.481 * eV,
   2.532 * eV, 2.585 * eV, 2.640 * eV, 2.697 * eV,
   2.757 * eV, 2.820 * eV, 2.885 * eV, 2.954 * eV,
   3.026 * eV, 3.102 * eV, 3.181 * eV, 3.265 * eV,
   3.353 * eV, 3.446 * eV, 3.545 * eV, 3.649 * eV,
   3.760 * eV, 3.877 * eV, 4.002 * eV, 4.136 * eV};
 const G4int nEntries = sizeof (photonEnergy) / sizeof (G4double);
//
G4double refractiveIndex[] ={1.3435, 1.344, 1.3445, 1.345, 1.3455,
   1.346, 1.3465, 1.347, 1.3475, 1.348,
   1.3485, 1.3492, 1.35, 1.3505, 1.351,
   1.3518, 1.3522, 1.3530, 1.3535, 1.354,
   1.3545, 1.355, 1.3555, 1.356, 1.3568,
   1.3572, 1.358, 1.3585, 1.359, 1.3595,
   1.36, 1.3608};
G4MaterialPropertiesTable* myMPT1 = new G4MaterialPropertiesTable();
myMPT1->AddProperty("RINDEX", photonEnergy, refractiveIndex, nEntries)
  ->SetSpline(true);
targetMaterial->SetMaterialPropertiesTable(myMPT1);
```

Don't care if physical or not!

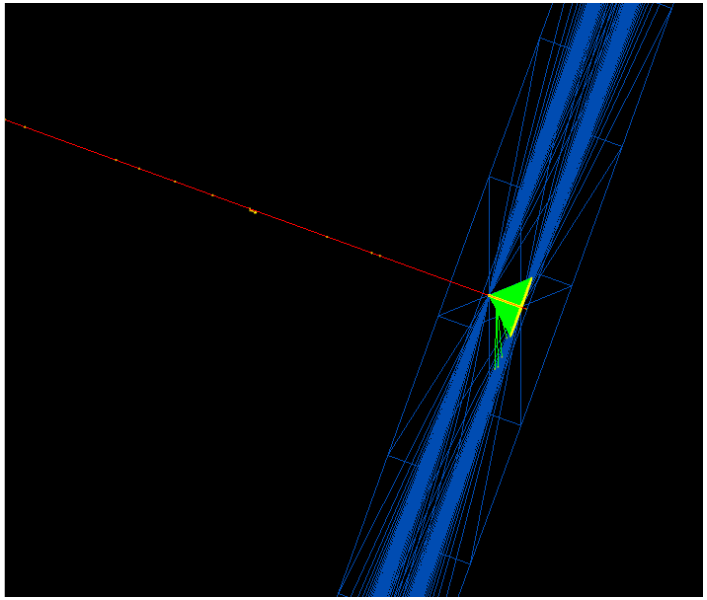
Fermilab

Using gdml to describe detector/with optical properties

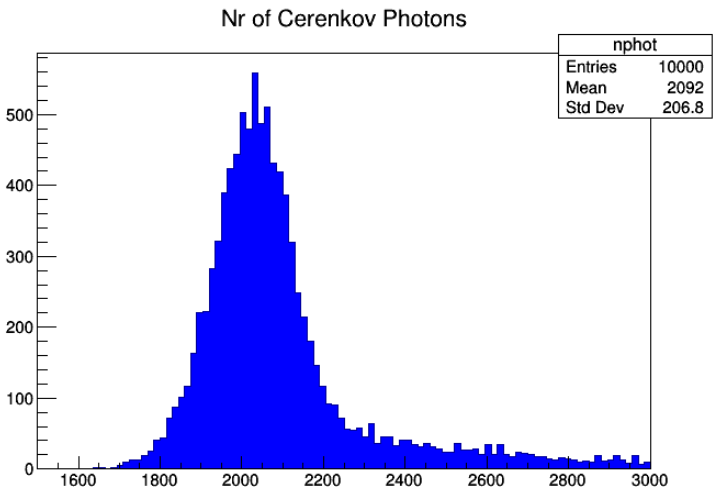
- Of course the detector description, including all optical properties can be expressed in (readable) gdml. For an example look at:
  https://cdcvs.fnal.gov/redmine/projects/artg4tk/repository/changes/gdml/singlecrystal_PbF2.gdml
- → geometry defined at one place!!

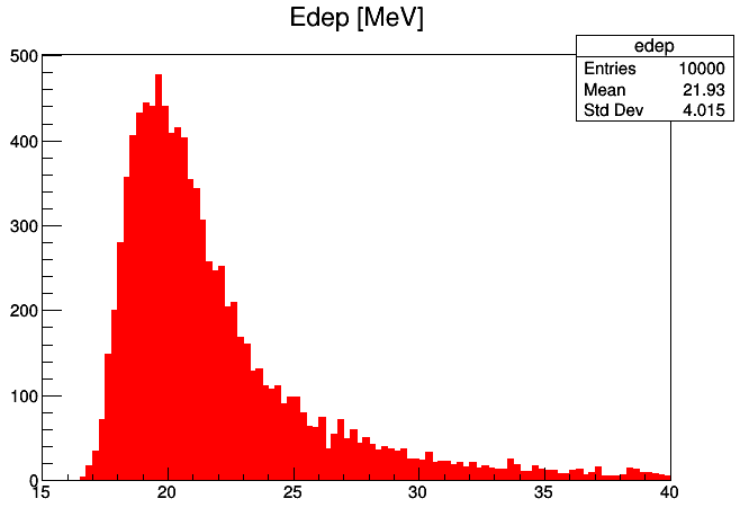
Accessing the Nr. Of optical (Cerenkov) photons produced in sensitive detector (TrackerSD)

```
G4bool TrackerSD::ProcessHits(G4Step* aStep,
        G4TouchableHistory*) {
    G4double edep = aStep->GetTotalEnergyDeposit();
    if (edep == 0.) return false;
    G4int photons = 0;
    if (aStep->GetPostStepPoint()->GetProcessDefinedStep()->GetProcessName() == "Cerenkov") {
        G4Cerenkov* proc = (G4Cerenkov*) aStep->GetPostStepPoint()->GetProcessDefinedStep();
        photons = proc->GetNumPhotons();
    }
    Analysis* analysis = Analysis::getInstance();
    analysis->FillEvent(edep,
            aStep->GetTrack()->GetPosition().x(),
            aStep->GetTrack()->GetPosition().y(),
            aStep->GetTrack()->GetPosition().z(),
            aStep->GetStepLength(),
            photons,
            G4RunManager::GetRunManager()->GetCurrentEvent()->GetEventID());
    return true;
}
```



6 cm thick Carbon disk, with optical properties.





Fermilab

Some timing results and summary

Configuration	Time for 10000 5GeV muons
Cerenkov on, Scintillation off, Stacking off (new)	5.14 [sec]
Cerenkov on, Scintillation on, Stacking off (switch off what you don't need)	11.83 [sec]
Cerenkov on, Scintillation off, Stacking on, photons killed in UserStackingAction (for long time the recommended way)	12.1 [sec]
Cerenkov on, Scintillation off, Stacking on, tracking	1 min 43.2 sec.

Patched version of Geant4 10.3. provides easy, flexible and efficient way to configure the Geant4 optical physics. Just what we asked for. No more ripping out of Geant4 functions necessary!