Experimental Techniques in Particle Physics

Geant4: Particles and Processes

Andreas Nowack
nowack@physik.rwth-aachen.de
RWTH Aachen University

WS 2020/21

Particles and Processes

In the Physics List

How to Define Particles in G4?

#include "G4ParticleTypes.hh"

void ……..PhysicsList::ConstructParticle() { …. }

How to Define Processes in G4?

#include "G4ProcessManager.hh"

void …….PhysicsList::ConstructProcess() { …. }

Particles

- each particle is represented by its own class
  - characterized by name, mass, charge, spin, and so on
  - more than 100 particles pre-defined
    - ordinary particles (electrons, protons, gammas, …)
    - resonant particles with very short lifetimes (vector mesons, delta baryons, …)
    - nuclei (deuteron, alpha, heavy ions, …)
    - quarks, di-quarks, and gluons
  - instantiate all particles needed before initialization of physics processes

- particles are organized in six major categories:
  - boson
  - lepton
  - meson
  - baryon
  - short-lived
  - ion

Particles (Bosons & Leptons)

```
//Construct Bosons
// gamma
  G4Gamma::GammaDefinition();
// optical photon
  G4OpticalPhoton::OpticalPhotonDefinition();
```

```
  // leptons
  G4Electron::ElectronDefinition();
  G4Positron::PositronDefinition();
  G4MuonPlus::MuonPlusDefinition();
  G4MuonMinus::MuonMinusDefinition();

  G4NeutrinoE::NeutrinoEDefinition();
  G4AntiNeutrinoE::AntiNeutrinoEDefinition();
  G4NeutrinoMu::NeutrinoMuDefinition();
  G4AntiNeutrinoMu::AntiNeutrinoMuDefinition();
```

Particles (Baryons & Mesons)

```
// baryons
G4Proton::ProtonDefinition();
G4AntiProton::AntiProtonDefinition();
G4Neutron::NeutronDefinition();
G4AntiNeutron::AntiNeutronDefinition();
```

```
// mesons
G4PionPlus::PionPlusDefinition();
G4PionMinus::PionMinusDefinition();
G4PionZero::PionZeroDefinition();
G4Eta::EtaDefinition();
G4EtaPrime::EtaPrimeDefinition();
G4KaonPlus::KaonPlusDefinition();
G4KaonMinus::KaonMinusDefinition();
G4KaonZero::KaonZeroDefinition();
G4AntiKaonZero::AntiKaonZeroDefinition();
G4KaonZeroLong::KaonZeroLongDefinition();
G4KaonZeroShort::KaonZeroShortDefinition();
```

Particles

- easy way to instantiate all particles of a given category
  - G4BosonConstructor
  - G4LeptonConstructor
  - G4MesonConstructor
  - G4BaryonConstructor
  - G4ShortlivedConstructor
  - G4IonConstructor

```
// e.g. construct all leptons
 G4LeptonConstructor pConstructor;
 pConstructor.ConstructParticle();
```

Particles and Processes

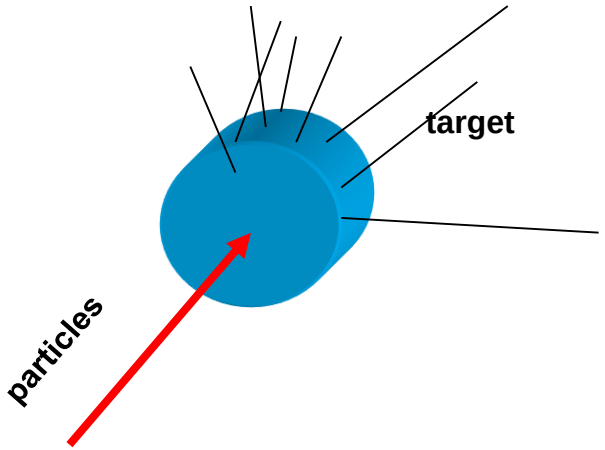
In this tutorial we will use three particles:

Electrons

Positrons

Gamma

Which processes are needed ?



Color code of tracks in Geant4:

red	negatively charged particle
green	neutral particle
blue	positively charged particle

Processes

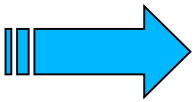
Physics processes describe how particles interact with materials.

Geant4 provides seven major categories of processes:

1. **transportation**
2. electromagnetic
3. decay
4. hadronic
5. optical
6. photolepton_hadron
7. parameterisation

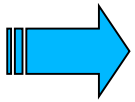
Particles and Processes

If particle is a **Gamma**



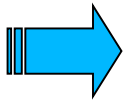
- Photo Electric Effect → G4PhotoElectricEffect.hh
- Compton Scattering → G4ComptonScattering.hh
- Gamma Conversion → G4GammaConversion.hh

If particle is an **Electrons**



- Multiple Scattering → G4eMultipleScattering.hh
- Ionisation → G4eIonisation.hh
- Bremsstrahlung → G4eBremsstrahlung.hh

If particle is a **Positrons**



- Multiple Scattering → G4eMultipleScattering.hh
- Ionisation → G4eIonisation.hh
- Bremsstrahlung → G4eBremsstrahlung.hh
- Annihilation → G4eplusAnnihilation.hh

Particles and Processes

If particle is a

Gamma

- Photo Electric Effect → G4PhotoEle…Effect.hh
- Compton Scattering → G4Com…ering.hh
- Gamma Conversion → G…sion.hh

If particle is an

Electrons

- Multiple Scat… → …ultipleScattering.hh
- Ionisati… → G4eIonisation.hh
- … → G4eBremsstrahlung.hh

If particle is a

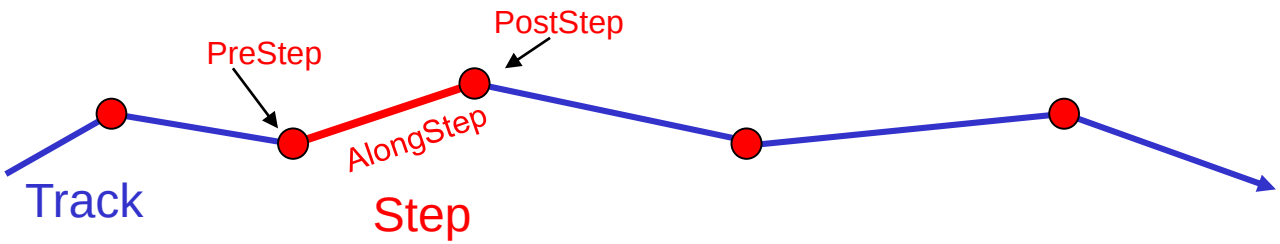
Positr

- …ple Scattering → G4eMultipleScattering.hh
- …onisation → G4eIonisation.hh
- Bremsstrahlung → G4eBremsstrahlung.hh
- Annihilation → G4eplusAnnihilation.hh

We need to order the calls for different processes for a given particle

Processes

- each particle has its own G4ProcessManager providing a list of processes that this particle can undertake



- simulation of the path of a particle step by step

- three possibilities for processes to take place
  - at rest G4VProcess::AtRestDoIt
  - along step G4VProcess::AlongStepDoIt
  - post step G4VProcess::PostStepDoIt

- …DoIt methods of the process class perform the physics processes:
  - momentum change
  - production of secondary particles

Processes

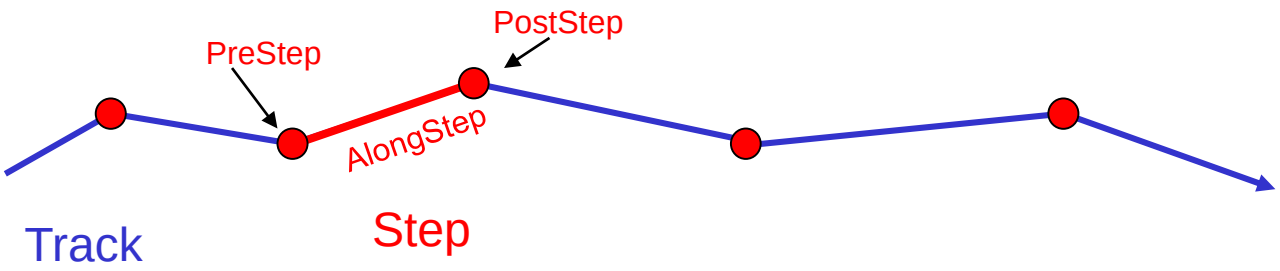
Methods

- AtRestDoIt
- AlongStepDoIt
- PostStepDoIt

```
pmanager->AddProcess(new G4eMultipleScattering, -1, 1, 1);
pmanager->AddProcess(new G4eIonisation,         -1, 2, 2);
```

-1 means inactive

```
AddProcess (G4VProcess* aProcess,
                        G4int ordAtRestDoIt=ordInActive,
                        G4int ordAlongSteptDoIt=ordInActive,
                        G4int ordPostStepDoIt=ordInActive)
```




```
AddDiscreteProcess (G4VProcess* aProcess,
                    G4int ordPostStepDoIt=ordInActive)
```

Particles and Processes

```
G4ParticleDefinition* particle = theParticleIterator->value();
G4ProcessManager* pmanager = particle->GetProcessManager();
G4String particleName = particle->GetParticleName();

AddTransportation();              // motion in space and time, mandatory for tracking particles!
if (particleName == "gamma") {                                  // gamma
  pmanager->AddDiscreteProcess(new G4PhotoElectricEffect,   1);
  pmanager->AddDiscreteProcess(new G4ComptonScattering,   2);
  pmanager->AddDiscreteProcess(new G4GammaConversion,   3);
} else if (particleName == "e-") {                              // electron
  pmanager->AddProcess(new G4eMultipleScattering,          -1, 1, 1);
  pmanager->AddProcess(new G4eIonisation,                  -1, 2, 2);
  pmanager->AddProcess(new G4eBremsstrahlung,              -1, 3, 3);
} else if (particleName == "e+") {                              // positron
  pmanager->AddProcess(new G4eMultipleScattering,          -1, 1, 1);
  pmanager->AddProcess(new G4eIonisation,                  -1, 2, 2);
  pmanager->AddProcess(new G4eBremsstrahlung,              -1, 3, 3);
  pmanager->AddProcess(new G4eplusAnnihilation,             0,-1, 4);
}
```

Processes

- registration of processes in G4ProcessManager is complex
- relations between processes are crucial in some cases
- easy way: G4PhysicsListHelper
  - users do not need to know about type of processes (at rest, discrete, continuous) and ordering

```
void MyPhysicsList::ConstructProcess() {
  AddTransportation();        // define transportation process
  ConstructEM();              // electromagnetic processes
}

void MyPhysicsList::ConstructEM() {
  G4PhysicsListHelper* ph = G4PhysicsListHelper::GetPhysicsListHelper(); // get pointer to helper

  G4ParticleDefinition* particle = G4Gamma::GammaDefinition();          // get pointer to gamma

  // construct and register processes for gamma
  ph->RegisterProcess(new G4PhotoElectricEffect(), particle);
  ph->RegisterProcess(new G4ComptonScattering(), particle);
  ph->RegisterProcess(new G4GammaConversion(), particle);
  ph->RegisterProcess(new G4RayleighScattering(), particle);
}
```

Exercise

1. Download DetectorPhys_T7.tar.gz and decompress it.
2. Follow the instructions in DetectorPhysPhysicsList.cc and define particles and processes.
3. Finally, play with the simulation and test the effects of the different processes.