Geant4: Electromagnetic Physics 1

V.Ivanchenko, BINP & CERN

- Process interface
- Physics categories
- Electromagnetic physics
- PhysicsList

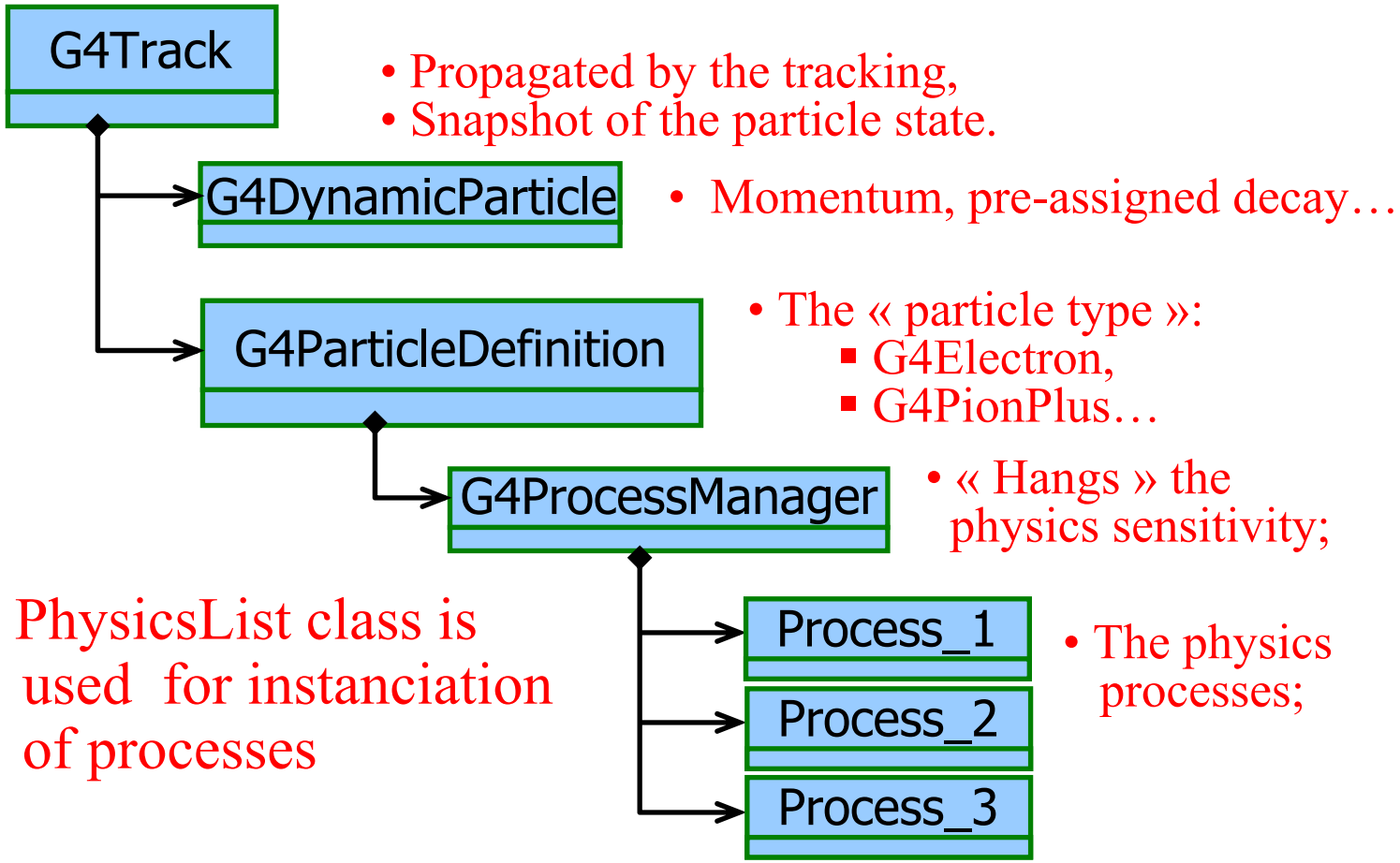
Introduction

- It a short course on Geant4 electromagnetic physics consist of 4 lectures
- It is personal view on Geant4 physics
- This course includes the material and slides from lectures and presentations of P.Gumplinger, M.Maire, P.Nieminen, M.G.Pia, M.Verderi, and L.Urban.

Geant4 physics processes

- Physics is described via abstract interface called ***process*** associated with ***particles***
- ***Process*** provides *Interaction Lenths*, *StepLimits*, and *DoIt* methods
- ***Process*** active *AlongStep, PostStep, AtRest*
- Distinction between process and model – one process may includes many models
- Generation of final state is independent from the access and use of cross sections and from tracking

What is tracked in Geant4 ?

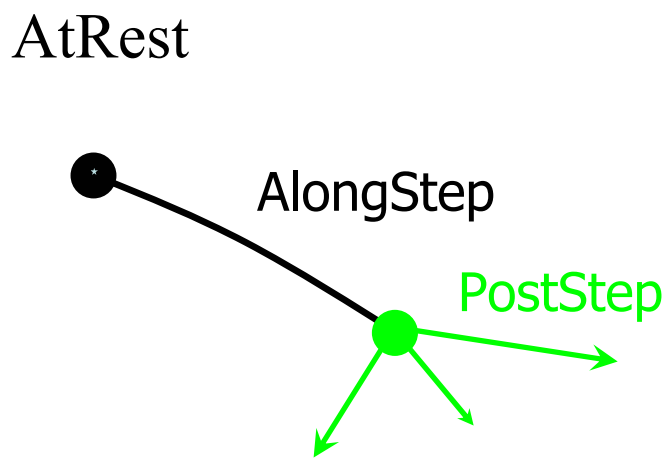


G4VProcess interface

- G4VProcess defines 6 pure virtual methods:
  - AtRestGetPhysicalInteractionLength(….)
  - AtRestDoIt(…)
  - AlongStepGetPhysicalInteractionLength(…)
  - AlongStepDoIt(…)
  - PostStepGetPhysicalInteractionLength(…)
  - PostStepDoIt(…)
- There are also other virtual methods:
  - IsApplicable(const G4ParticleDefinition&)
  - BuildPhysicsTable(const G4ParticleDefinition&)
  - ….
- G4VProcess defined in source/processes/management

G4VProcess actions

- Abstract class defining the common interface of all processes in GEANT4
- AtRest
  - decay at rest, annihilation at rest, …
- AlongStep
  - continuous energy losses, multiple scattering, …
- PostStep
  - decay in flight, hardron elastic and inelastic, …



Geant4 physics categories

- There are following categories:
  - Electromagnetic
  - Hadronic
  - Decay
  - Optical
  - Transportation
  - Parameterisation
- Subcategories of Electromagnetic domain:
  - Muons
  - Lowenergy
  - Standard
  - Xrays
  - Utils

Electromagnetic Physics

- Processes of gamma, electron, and positron interactions with media was traditionally called ***"Electromagnetic Processes" (EM)***
- Hadron interaction with atomic electrons are also EM
- Hadron photo- and electro- production are simulated in framework of G4 hadronic physics

EM packages

- ***Standard*** – basic set of processes for HEP
- ***Muons*** – basic set of muon processes for HEP
- ***Xrays*** – xray and optical proton production
- ***Lowenergy*** – alternative set of processes with low energy extension of gamma, electron, and hadron EM physics

- ***Utils – common classes and interfaces for other EM packages:***
  - ***interfaces***
  - ***energy loss tables builders***
  - ***fluctuations of energy losses***
  - ***multiple scattering***

Standard EM Physics

- The projectile is assumed to have the energy $E_{kin} > 1keV$
- The atomic electrons are quasi-free – their binding energies neglected (except some corrections at low energies)
- The atomic nucleus are fixed – no recoil
- The matter is described as homogeneous, isotropic, amorphous

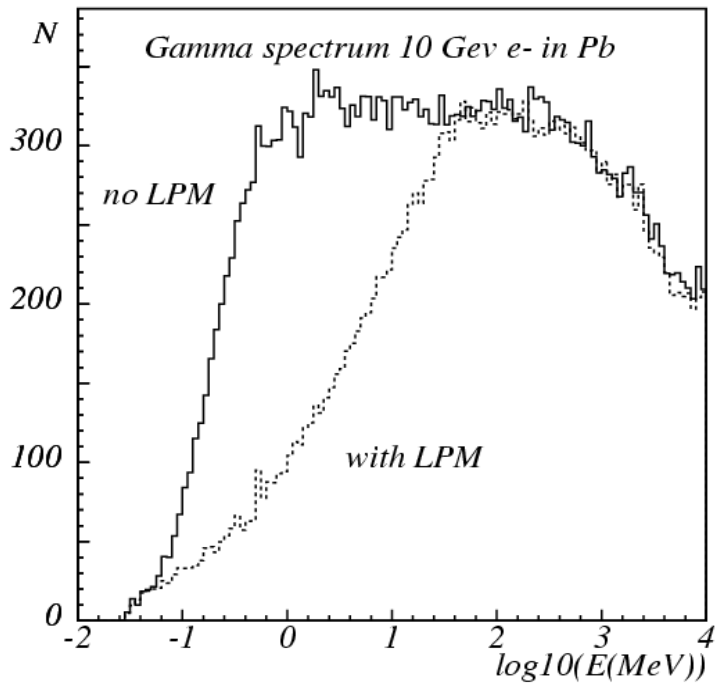
Standard EM Processes

- Gamma
  - Photo-electric effect
  - Compton scattering
  - e^+e^- pair production
  - $\mu^+\mu^-$ pair production
- Electron and positron
  - Ionization
  - Bremsstrahlung
  - Positron annihilation
- Muons
  - Ionization
  - Bremsstrahlung
  - e^+e^- pair production
- Hadrons
  - Ionization
- Ions
  - Ionization
- Multiple scattering

Standard EM Physics (Michel Maire and Laszlo Urban)

- Standard G4 physics was based on G3 knowledge/experience
- Review of G3 models have been done
- More precise theories were used if possible/ necessary
- Extension to highest energies in progress

Landau-Pomeranchuk-Migdal Effect for bremsstrahlung



PhysicsList

- It is one of the « mandatory user classes »;
  - Defined in source/run
- Defines the three pure virtual methods:
  - ConstructParticle()
  - ConstructProcesse()
  - SetCuts()
- Concrete PhysicsList needs to inherit from G4VUserPhysicsList or G4VModularPhysicsList
- For interactivity G4UserPhysicsListMessenger can be used to handle PhysicsList parameters

Example: AddTransportation

```
void G4VUserPhysicsList::AddTransportation()
{
  G4Transportation* theTransportationProcess= new G4Transportation();
    // loop over all particles in G4ParticleTable
    theParticleIterator->reset();
    while( (*theParticleIterator)() ){
        G4ParticleDefinition* particle = theParticleIterator->value();
        G4ProcessManager* pmanager = particle->GetProcessManager();
        if (!particle->IsShortLived()) {
            if ( pmanager == 0)  {
                G4Exception("G4VUserPhysicsList::AddTransportation : no process
    manager!");
            } else {
              // add transportation with ordering = ( -1, "first", "first" )
              pmanager->AddProcess(theTransportationProcess);
              pmanager->SetProcessOrderingToFirst(theTransportationProcess,
    idxAlongStep);
              pmanager->SetProcessOrderingToFirst(theTransportationProcess,
    idxPostStep);
            }
      }
}
```

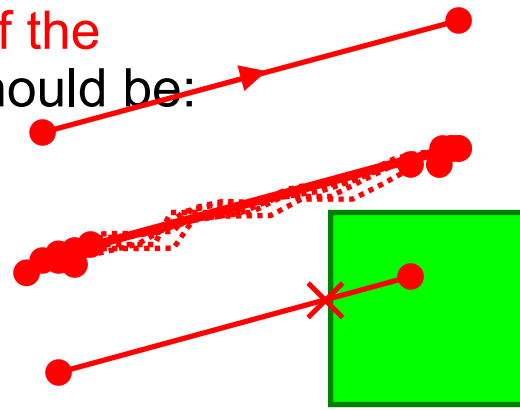
Processes ordering

- Ordering of following processes is critical:
  - Assuming n processes, the ordering of the AlongGetPhysicalInteractionLength should be:

    [n-2] …

    [n-1] multiple scattering

    [n] transportation



- Why ?
  - Processes return a « true path length »;
  - The multiple scattering convers it into into a ***shorter*** « geometrical » path length;
  - Based on this new length, the transportation can geometrically limits the step.
- Other processes ordering usually do not matter.

Example: Gamma processes

- Discrete processes - only PostStep actions;
  - Use function AddDiscreteProcess;
  - **pmanager** is the G4ProcessManager of the gamma;
  - Assume the transportation has been set by AddTransportation;
- Code sample:

```
// Construct processes for gamma:
pmanager->AddDiscreteProcess(new G4GammaConversion());
pmanager->AddDiscreteProcess(new G4ComptonScattering());
pmanager->AddDiscreteProcess(new G4PhotoElectricEffect());
```

Example: Electron processes

```
// Construct processes for positron
G4VProcess* theMultipleScattering = new G4MultipleScattering();
G4VProcess* eIonisation = new G4eIonisation();
G4VProcess* eBremsstrahlung = new G4eBremsstrahlung();
// add processes
pmanager->AddProcess(theMultipleScattering);
pmanager->AddProcess(eIonisation);
pmanager->AddProcess(eBremsstrahlung);
// set ordering for AlongStepDoIt
pmanager->SetProcessOrdering(theMultipleScattering, idxAlongStep,  1);
pmanager->SetProcessOrdering(eIonisation,           idxAlongStep,  2);
// set ordering for PostStepDoIt
pmanager->SetProcessOrdering(theMultipleScattering, idxPostStep, 1);
pmanager->SetProcessOrdering(eIonisation,           idxPostStep, 2);
pmanager->SetProcessOrdering(eBremsstrahlung,    idxPostStep, 3);
```

Example: Positrons processes

```
G4VProcess* theeplusMultipleScattering = new G4MultipleScattering();
G4VProcess* theeplusIonisation = new G4eIonisation();
G4VProcess* theeplusBremsstrahlung = new G4eBremsstrahlung();
G4VProcess* theeplusAnnihilation = new G4eplusAnnihilation();
pmanager->AddProcess(theeplusMultipleScattering);
pmanager->AddProcess(theeplusIonisation);
pmanager->AddProcess(theeplusBremsstrahlung);
pmanager->AddProcess(theeplusAnnihilation);
pmanager->SetProcessOrderingToFirst(theeplusAnnihilation, idxAtRest);
pmanager->SetProcessOrdering(theeplusMultipleScattering, idxAlongStep,  1);
pmanager->SetProcessOrdering(theeplusIonisation,          idxAlongStep,  2);
pmanager->SetProcessOrdering(theeplusMultipleScattering, idxPostStep, 1);
pmanager->SetProcessOrdering(theeplusIonisation,          idxPostStep, 2);
pmanager->SetProcessOrdering(theeplusBremsstrahlung,    idxPostStep, 3);
pmanager->SetProcessOrdering(theeplusAnnihilation,        idxPostStep, 4);
```

Hadrons EM processes

- Hadrons (pions, kaons, proton,…)
- Light ions (deuteron, triton, alpha)
- Heavy ions (GenericIon)
- Example:

```
G4VProcess* theMultipleScattering = new G4MultipleScattering();
G4VProcess* hIonisation = new G4hIonisation();
pmanager->AddProcess(theMultipleScattering);
pmanager->AddProcess(hIonisation);
pmanager->SetProcessOrdering(theMultipleScattering, idxAlongStep,  1);
pmanager->SetProcessOrdering(hIonisation,           idxAlongStep,  2);
pmanager->SetProcessOrdering(theMultipleScattering, idxPostStep, 1);
pmanager->SetProcessOrdering(hIonisation,           idxPostStep, 2);
```

How to build PhysicsList?

- PhysicsList can be build by experience user
- PhysicsList can be taken from G4 examples or from the web page:
  - http://geant4.web.cern.ch/geant4/organisation/ /working_groups.html#vg.Had
- Novice examples
  - **N02**: Simplified tracker geometry with uniform magnetic field
  - **N03**: Simplified calorimeter geometry
  - **N04**: Simplified collider detector with a readout geometry
- Extended and advanced examples

Conclusion remarks

- Using Geant4 examples novice user can design his/her PhysicsList with EM physics processes without detailed studying of interaction of particles with matter
- Default values of internal model parameters are reasonably defined
- To estimate the accuracy of simulation results indeed one have to study Geant4 in more details
- It is true for any simulation software!