Geant4 Physics List

V.Ivanchenko
Thanks to P.Gumplinger, M.Maire, H.P.Wellisch

- General Physics
- Electromagnetic Physics
- Optical Photons
- Hadronic Physics
PhysicsList

- It is one of the « mandatory user classes »;
  - Defined in source/run
- Defines the three pure virtual methods:
  - ConstructParticle()
  - ConstructProcess()
  - SetCuts()
- Concrete PhysicsList needs to inherit from G4VUserPhysicsList or G4VModularPhysicsList
- For interactivity G4UserPhysicsListMessenger can be used to handle PhysicsList parameters
The list of particles used in simulation needs to be registered
- G4Gamma::Gamma()
- G4Proton::Proton()
- ........

The list of physics processes per particle need to be registered before initialization of G4RunManager
- /run/initialize
How to build PhysicsList?

- PhysicsList can be build by experience user
- Components are distributed inside subdirectory
  - $G4INSTALL/physics_list
- PhysicsList can be studied using G4 novice examples
  - N02: Simplified tracker geometry with uniform magnetic field
  - N03: Simplified calorimeter geometry
  - N04: Simplified collider detector with a readout geometry
- Copy PhysicsList from extended and advanced examples
  - electromagnetic: 13 examples for different aspects of EM physics
  - N06 and extended/optical: specifics of optical photons
  - advanced: different mini-applications of Geant4 based on real experimental setups
- Use predefined PhysicsList from
  - $G4INSTALL/physics_list/hadronic
Example: AddTransportation

```cpp
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);
            }
        }
    }
}
```
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();

✿ The most simple code:

```c++
// Construct processes for gamma:
pmanager->AddDiscreteProcess(new G4GammaConversion());
pmanager->AddDiscreteProcess(new G4ComptonScattering());
pmanager->AddDiscreteProcess(new G4PhotoElectricEffect());
```
Example: electron and positron

Main interface with definition of the process order:
G4ProcessManager::AddProcess(G4VProcess*, int orderAtRest,
    int orderAlongStep, int orderPostStep);

NOTE: if (order < 0) – process inactive; else – the order of DoIt method;
    inverse order of GetInteractionLength

// add processes for e−
G4ProcessManager* pmanager = G4Electron::Electron()->GetProcessManager();
pmanager->AddProcess (new G4MultipleScattering, -1, 1, 1 );
pmanager->AddProcess (new G4eIonisation, -1, 2, 2 );
pmanager->AddProcess (new G4eBremsstrahlung, -1, 3, 3 );

// add processes for e+
pmanager = G4Positron::Positron()->GetProcessManager();
pmanager->AddProcess (new G4MultipleScattering, -1, 1, 1 );
pmanager->AddProcess (new G4eIonisation, -1, 2, 2 );
pmanager->AddProcess (new G4eBremsstrahlung, -1, 3, 3 );
pmanager->AddProcess (new G4eplusAnnihilation, 1, -1, 4 );
Example: hadrons and ions

- Hadrons (pions, kaons, proton, …)
- Light ions (deuteron, triton, alpha)
- Heavy ions (GenericIon)
- Example loop over list of particles:

```cpp
G4ProcessManager* pmanager = particle->GetProcessManager();
G4String pName = particle->GetParticleName();

// Ions
If ( pName == "GenericIon" || pName == "alpha" || pName == "He3") {
  pmanager->AddProcess (new G4MultipleScattering, -1, 1, 1);
  pmanager->AddProcess (new G4ionIonisation, -1, 2, 2);
}

// Hadrons
} else if (particle->GetPDGCharge() != 0 && particle->GetPDGMass() > 130.*MeV) {
  pmanager->AddProcess (new G4MultipleScattering, -1, 1, 1);
  pmanager->AddProcess (new G4hIonisation, -1, 2, 2);
}
```
Processes ordering

✦ Ordering of following processes is critical:
  − Assuming $n$ processes, the ordering of the `AlongGetPhysicalInteractionLength` should be:
    
    $[n-2] \ldots$
    $[n-1]$ multiple scattering
    $[n]$ transportation

✦ Why?
  − Processes return a «true path length»;
  − The multiple scattering converts it 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
Standard EM Physics List

- Standard EM PhysicsList:
  - $G4INSTALL/physics_lists/electromagnetic/standard

- Different aspects of EM physics are demonstrated in examples:
  - $G4INSTALL/examples/electromagnetic/extended

- There are UI commands for defining cuts and to choose options for the EM physics:
  - /testem/phys/setCuts 0.01 mm
  - /testem/phys/addPhysics standard
  - ....

- Steering is also provided via:
  - G4EmProcessOptions::SetMaxEnergy(10.0*GeV);
  - G4EmProcessOptions::SetVerbose(2);

- Components of this PhysicsList are provided as physics_list subdirectory and in different extended examples
**Geant4 low energy EM physics**

- When energy transfer becomes close to energy of atomic electrons atomic shell structure should be taken into account.
- Problems with theory, so phenomenology and experimental data are used.
Geant4 low energy EM physics

✦ Validity down to 250 eV
  – 250 eV is a “suggested” lower limit
  – data libraries down to 10 eV
  – 1 < Z < 100

✦ Exploit evaluated data libraries (from LLNL):
  • EADL (Evaluated Atomic Data Library)
  • EEDL (Evaluated Electron Data Library)
  • EPDL97 (Evaluated Photon Data Library)

Photon transmission, 1μm Pb

shell effects
Geant4 low energy EM physics

- Compton scattering
- Polarised Compton
- Rayleigh scattering
- Photoelectric effect
- Pair production
- Bremsstrahlung
- Electron ionisation
- Hadron ionisation
- Atomic relaxation
- Set of Penelope models (new)

- It is relatively new package
- Development is driven by requirements which come from medicine and space research
- There are also users in HEP instrumentation
To use G4 low energy package user has to substitute a standard process in the PhysicsList by the corresponding low energy process:

- G4hIonisation $\rightarrow$ G4hLowEnergyIonisation
- G4eIonisation $\rightarrow$ G4LowEnergyIonisation
- ....

The environment variable G4LEDATA should be defined

- setenv G4LEDATA $G4INSTALL/data/G4EMLOW3.0
Optical photons generated by following processes (processes/electromagnetic/xrays):
- Scintillation
- Cherenkov
- Transition radiation

Optical photons have following physics processes (processes/optical/):
- Refraction and Reflection at medium boundaries
- Bulk Absorption
- Rayleigh scattering

ExampleN06 at /examples/novice/N06
Material properties should be defined for \texttt{G4Scintillation} process, so only inside the scintillator the process is active.

\texttt{G4Scintillation} should be ordered after all energy loss processes.

\texttt{G4Cerenkov} is active only if for the given material an index of refraction is provided.

For simulation of optical photons propagation \texttt{G4OpticalSurface} should be defined for a given optical system.
Geant4 19.05.05 PhysicsList 17

G4Cerenkov: User Options

- Suspend primary particle and track Cherenkov photons first
- Set the max number of Cherenkov photons per step

in ExptPhysicsList:

#include "G4Cerenkov.hh"

G4Cerenkov* theCerenkovProcess = new G4Cerenkov("Cerenkov");
theCerenkovProcess -> SetTrackSecondariesFirst(true);
G4int MaxNumPhotons = 300;
theCerenkovProcess->SetMaxNumPhotonsPerStep(MaxNumPhotons);
Boundary Processes

- **G4OpticalSurface** needs to be defined
- **Dielectric - Dielectric**
  Depending on the photon’s wave length, angle of incidence, (linear) polarization, and refractive index on both sides of the boundary:
  (a) total internal reflected
  (b) Fresnel refracted
  (c) Fresnel reflected

- **Dielectric - Metal**
  (a) absorbed (detected)
  (b) reflected
Some remarks

- Geant4 **Standard EM** package the optimal for most part of HEP applications
- Geant4 **Lowenergy** package provide a possibility to apply toolkit to variety of applications for which atomic shell structure is essential
- Optical photons generation and tracking can be simulated inside the same geometry
Hadronic Physics

✦ Interaction of particles with atomic nuclei
✦ Interactions on-fly are simulated by discrete processes - only PostStep actions
  • Cross section calculation and secondary generators are separated
  • Different secondary generators should be applied for different energy ranges and particle type
✦ Capture of stopping particles: only AtRest actions
A not totally correct hadronic model summary

Absorption at rest
μ, π, K, p-bar, n-bar

CHIPS 1
CHIPS (gamma)
LEP
HEP
Evap
Binary cascade
FTF string
QGS string
multifrag
Fermi
Bertini cascade

Rad. Dec.
Exciton prec.
Fission
Inclusive mars
LEpp, np

GEM
EM dissociation of ions
Neutron_hpt
HE Elastic

Ablation/abrasion

1 MeV 10 MeV 100 MeV 1 GeV 10 GeV 100 GeV 1 TeV 10 TeV 100 TeV

CERN/EP/SFT
Hadronic Physics: proton

- Cross section data set and list of models need to be defined

- Example:

  ```c++
  // proton inelastic by Binary Cascade and LHEP
  particle = G4Proton::Proton();
pmanager = particle->GetProcessManager();
G4ProtonInelasticProcess* p = G4ProtonInelasticProcess);
p->RegisterMe(new G4LEProtonInelastic );
p->RegisterMe(new G4BinaryCascade );
p->AddDataSet(new G4ProtonInelasticCrossSection );
pmanager->AddDiscreteProcess(p);
  ```
Hadronic Physics: neutron

// neutron inelastic by Binary Cascade and LHEP, fission, and capture
particle = G4Neutron::Neutron();
pmanager = particle->GetProcessManager();
G4NeutronInelasticProcess* p = G4NeutronInelasticProcess);
// Default energy ranges for models
p->RegisterMe(new G4LENeutronInelastic );
p->RegisterMe(new G4BinaryCascade );
p->AddDataSet(new G4NeutronInelasticCrossSection );
pmanager->AddDiscreteProcess(p);
// fission
theFissionProcess->RegisterMe(new G4LFission );
pmanager->AddDiscreteProcess(theFissionProcess);
// capture
G4HadronCaptureProcess* theCaptureProcess = new G4HadronCaptureProcess;
theCaptureProcess->RegisterMe(new G4LCapture );
pmanager->AddDiscreteProcess(theCaptureProcess );
Predefined Physics Lists

- Hadronic physics is complex
- Different experiments/groups need to coordinate simulation efforts
- Predefined Physics Lists were designed
  - $G4INSTALL/physics_lists/hadronic
- Both hadronic and EM physics are included, method of compilation and linking is provided
- Different use-cases
  - Geant4 web
From TeV down to MeV energy scale for precise simulation of detector response

Following lists are recommended for Geant4 version 7.0p01:

- QGSP\_GN (Quark Gluon String + PreCompound + Gamma-Nuclear)
- QGSP\_BERT (Quark Gluon String + PreCompound + Bertini Cascade)
- QGSC (Quark Gluon String + CHIPS)
- LHEP\_BERT (Low-energy Parameterized + Bertini cascade)
Extensions of Hadronic Physics

- For neutron penetration
  High Precision model of neutron transport down to very low energies
  - NeutronHPCrossSections
- For radioactive decays of nuclei:
  - G4RADIOACTIVEDATA
- Gamma and electro–nuclear interaction by the CHIPS package
Conclusion remarks

✦ Using Geant4 examples and physics_lists package novice user can take existing PhysicsList 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 and physics in more details
✦ It is true for any simulation software!