Physics I: Physics Lists

Geant4 Tutorial at Stanford
4 March 2014
Dennis Wright (SLAC)

SLAC NATIONAL ACCELERATOR LABORATORY

Outline

- Introduction
- The G4VUserPhysicsList class
- Modular physics lists
- Packaged physics lists
- Choosing the appropriate physics list
- Validating your physics list

What is a Physics List?

- A class which collects all the particles, physics processes and production thresholds needed for your application
- It tells the run manager how and when to invoke physics
- It is a very flexible way to build a physics environment
  - user can pick the particles he wants
  - user can pick the physics to assign to each particle
- But, user must have a good understanding of the physics required
  - omission of particles or physics could cause errors or poor simulation

Why Do We Need a Physics List?

- Physics is physics – shouldn't Geant4 provide, as a default, a complete set of physics processes that everyone can use?
- No:
  - there are many different physics models and approximations
    - very much the case for hadronic physics
    - but also true for electromagnetic physics
  - computation speed is an issue
    - a user may want a less-detailed, but faster approximation
  - no application requires all the physics and particles that Geant4 has to offer
    - e.g., most medical applications do not want multi-GeV physics

Why Do We Need a Physics List?

- For this reason Geant4 takes an atomistic, rather than an integral approach to physics
  - provide many physics components (processes) which are decoupled from one another (for the most part)
  - user selects these components in custom-designed physics lists in much the same way as a detector geometry is built
- Exceptions
  - a few electromagnetic processes must be used together
  - future processes involving interference of electromagnetic and strong interactions may require coupling as well

Physics Processes Provided by Geant4

- EM physics
  - "standard" processes valid from ~ 1 keV to ~PeV
  - "low energy" valid from 250 eV to ~PeV
  - optical photons
- Weak interaction physics
  - decay of subatomic particles
  - radioactive decay of nuclei
- Hadronic physics
  - pure strong interaction physics valid from 0 to ~TeV
  - electro- and gamma-nuclear valid from 10 MeV to ~TeV
- Parameterized or "fast simulation" physics

G4VUserPhysicsList

- All physics lists must derive from this class
  - and then be registered with the run manager

- Example:

```
class MyPhysicsList: public G4VUserPhysicsList {
   public:
      MyPhysicsList();
      ~MyPhysicsList();
      void ConstructParticle();
      void ConstructProcess();
      void SetCuts();
}
```

- User must implement the methods ConstructParticle, ConstructProcess and SetCuts

G4VUserPhysicsList: Required Methods

- ConstructParticle() – choose the particles you need in your simulation and define them all here
- ConstructProcess() – for each particle, assign all the physics processes important in your simulation
  - What's a process?
  - → a class that defines how a particle should interact with matter (it's where the physics is!)
  - more on this later
- SetCuts() – set the range cuts for secondary production
  - What's a range cut?
  - → essentially a low energy limit on particle production
  - more on this later

ConstructParticle()

```
void MyPhysicsList::ConstructParticle()  {
    G4BaryonConstructor* baryonConstructor =
                                    new G4BaryonConstructor();
    baryonConstructor->ConstructParticle();
    delete baryonConstructor;
    G4BosonConstructor* bosonConstructor =
                                    new G4BosonConstructor();
    bosonConstructor->ConstructParticle();
    delete bosonConstructor;
    …
    …
}
```

ConstructParticle() (alternate)

```
void MyPhysicsList::ConstructParticle()
{
    G4Electron::ElectronDefinition();
    G4Proton::ProtonDefinition();
    G4Neutron::NeutronDefinition();
    G4Gamma::GammaDefinition();
    …
    …
}
```

ConstructProcess()

```
void MyPhysicsList::ConstructProcess()  {
    AddTransportation();
    // method provided by G4VUserPhysicsList assigns transportation
    // process to all particles defined in ConstructParticle()

    ConstructEM();
    // method may be defined by user (for convenience)
    // put electromagnetic physics here

    ConstructGeneral();
    // method may be defined by user to hold all other processes
}
```

ConstructEM()

```
void MyPhysicsList::ConstructEM()  {
   G4PhysicsListHelper* ph = G4PhysicsListHelper::GetPhysicsListHelper();

   theParticleIterator->reset();
   while( (*theParticleIterator)() ) {
      G4ParticleDefinition* particle = theParticleIterator->value();
      if  (particle == G4Gamma::Gamma() ) {
         ph->RegisterProcess(new G4GammaConversion(),  particle);
          ….  // add more processes
      }
       …  // do electrons, positrons, etc.
   }
```

ConstructGeneral()

```
void MyPhysicsList::ConstructGeneral()  {
   G4PhysicsListHelper* ph = G4PhysicsListHelper::GetPhysicsListHelper();
   // Add decay process
   G4Decay* theDecayProcess = new G4Decay();
   theParticleIterator->reset();
   while( (*theParticleIterator)() ) {
      G4ParticleDefinition* particle = theParticleIterator->value();
      if  (theDecayProcess->IsApplicable(*particle) ) {
         ph->RegisterProcess(theDecayProcess, particle);
      }
   }
   // Add other physics
```

SetCuts()

```
void MyPhysicsList::SetCuts()
{
   defaultCutValue = 0.7*mm;
   SetCutValue(defaultCutValue, “gamma”);
   SetCutValue(defaultCutValue, “e-”);
   SetCutValue(defaultCutValue, “e+”);
   SetCutValue(defaultCutValue, “proton”);
   //
   // These are all the production cuts you need to set
   // - not required for any other particle
}
```

G4VModularPhysicsList

- The physics list in our example is quite simple
- A realistic physics list is likely to have many more physics processes
  - such a list can become quite long, complicated and hard to maintain
  - try a modular physics list instead
- Features of G4VModularPhysicsList
  - derived from G4VUserPhysicsList
  - AddTransportation() automatically called for all registered particles
  - allows you to define “physics modules”: EM physics, hadronic physics, optical physics, etc.

A Simple G4VModularPhysicsList

- Constructor:

```
MyModPhysList::MyModPhysList(): G4VModularPhysicsList() {
    defaultCutValue = 0.7*mm;
    RegisterPhysics(new ProtonPhysics() );
    // all physics processes having to do with protons
    RegisterPhysics(new ElectronPhysics() );
    // all physics processes having to do with electrons
    RegisterPhysics(new DecayPhysics() );
    // physics of unstable particles
}
```

- SetCuts:

```
void MyModPhysList::SetCuts() { SetCutsWithDefault(); }
```

Physics Constructors

- Allows you to group particle and process construction according to physics domains
- class ProtonPhysics : public G4VPhysicsConstructor

```
{
    public:
        ProtonPhysics(const G4String& name = "proton");
        virtual ~ProtonPhysics();
        virtual void ConstructParticle()
        // easy – only one particle to build in this case
        virtual void ConstructProcess();
        // put here all the processes a proton can have
}
```

Packaged Physics Lists

- Our example dealt mainly with electromagnetic physics
- A realistic physics list can be found in basic example B3
  - uses “standard” EM physics and decay physics
  - a good starting point
  - add to it according to your needs
- Adding hadronic physics is more involved
  - for any one hadronic process, user may choose from several hadronic models
  - choosing the right models for your application requires care
  - to make things easier, pre-packaged physics lists are provided according to some reference use cases

Packaged Physics Lists

- Each pre-packaged physics list includes different choices of EM and hadronic physics
- A list of these can be found in your copy of the toolkit at

  geant4/source/physics_lists/lists/include
- Caveats
  - these lists are provided as a “best guess” of the physics needed in a given use case
  - the user is responsible for validating the physics for his own application and adding (or subtracting) the appropriate physics
  - they are intended as starting points or templates

Reference Physics Lists

- Among the pre-packaged physics lists are the “Reference” physics lists
  - a small number of well-maintained and tested physics lists
  - also the most used (ATLAS, CMS, etc.) and most recommended
- These are updated less frequently
  - more stable
- More on these, and which ones we recommend, later

A Short Guide to Choosing a Physics List

Choosing a Physics List

- Which physics list you use is highly dependent on your use case
- Before choosing, or building your own, familiarize yourself with the major physics processes available
  - the process-model catalog is useful for this
  - see Geant4 web page under User Support, item 11b

- Geant4 provides several "reference physics lists" which are routinely validated and updated with each release
  - these should be considered only as starting points which you may need to validate or modify for your application
- There are also many physics lists in the examples which you can copy
  - these are often very specific to a given use case

Choosing a Physics List

- There are currently 19 packaged physics lists available
  - but you will likely be interested in only a few, namely the "reference" physics lists
  - many physics lists are either developmental or customized in some way, and so not very useful to new users

- All but one of the packaged physics lists use templates
  - the LBE physics list is the old-style "flat" list without templates or physics builders

- 6 reference physics lists:
  - FTFP_BERT, FTFP_BERT_HP
  - QGSP_BERT, QGSP_BERT_HP, QGSP_BIC
  - QGSP_FTFP_BERT

Physics List Naming Convention

- The following acronyms refer to various hadronic options
  - QGS -> Quark Gluon String model (>~20 GeV)
  - FTF -> Fritiof string model (>~5 GeV)
  - BIC -> Binary Cascade (<~ 10 GeV)
  - BERT -> Bertini-style cascade (<~ 10 GeV)
  - HP -> High Precision neutron model (< 20 MeV)
  - P -> G4Precompund model used for de-excitation

- EM options designated by
  - no suffix : standard EM physics
  - EMV suffix : older but faster EM processes
  - other suffixes for other EM options

Reference Physics Lists

- FTFP_BERT
  - recommended by Geant4 for HEP
  - contains all standard EM processes
  - uses Bertini-style cascade for hadrons < 5 GeV
  - uses FTF (Fritiof) model for high energies (> 4 GeV)
- QGSP_BERT
  - all standard EM processes
  - Bertini-style cascade up to 9.9 GeV
  - QGS model for high energies (> ~18 GeV)
  - FTF in between

Reference Physics Lists

- QGSP_BIC
  - same as QGSP_BERT, but replaces Bertini cascade with Binary cascade and G4Precompound model
  - recommended for use at energies below 200 MeV (many medical applications)

- FTFP_BERT_HP
  - same as FTFP_BERT, but with high precision neutron model used for neutrons below 20 MeV
  - significantly slower than FTFP_BERT when full thermal cross sections used
    - there's an option to turn this off
  - for radiation protection and shielding applications

Other Physics Lists

- Shielding
  - based on FTFP_BERT_HP with improved neutron cross sections from JENDL
  - better ion interactions using QMD model
  - currently used by SuperCDMS dark matter search
  - recommended for:
    - shielding applications
    - space physics
    - HEP

Other Physics Lists

- FTFP_INCLXX, FTFP_INCLXX_HP
  - like FTFP_BERT, but with BERT replaced by INCL++ cascade model
- QBBC
  - uses both BERT and BIC cascade models
  - latest coherent elastic scattering
  - neutronXS models (faster CPU-wise)
- QGSP_BIC_HP
  - same as QGSP_BIC, but with high precision neutron model used for neutrons below 20 MeV
  - recommended for radiation protection, shielding and medical applications

Other Physics Lists (based on use case)

- If primary particle energy in your application is < 5 GeV (for example, clinical proton beam of 150 MeV)
  - start with a physics list which includes BIC or BERT
  - e.g. QGSP_BIC, QGSP_BERT, FTFP_BERT, etc.
- If neutron transport is important
  - start with physics list containing “HP”
  - e.g. QGSP_BIC_HP, FTFP_BERT_HP, etc.
- If you’re interested in Bragg curve physics
  - use a physics list ending in “EMV” or “EMX”
  - e.g. QGSP_BERT_EMV

Other Physics Lists (based on use case)

- For optical photon transport
  - start with the LBE physics list
  - list is a bit old, but optical code can be extracted for other applications
- For radioactive decay
  - use LBE list as an example, or the physics list in example B3
- For detailed line emissions from EM processes
  - LBE or see following slide

Alternate EM Physics Lists

- Up to now, most physics lists mentioned have used the "standard" EM processes, but "low energy" EM physics is also available
  - G4EmLivermorePhysics (physics list suffix = LIV)
  - G4EmLivermorePolarizedPhysics
  - G4EmPenelopePhysics (suffix = PEN)
  - G4EmDNAPhysics

- Physics lists containing these are recommended for micro-dosimetry applications

- For examples using a DNA physics list, go to
  - geant4/source/examples/advanced

Using Alternate EM Physics Lists

- These physics list classes derive from the G4VPhysicsConstructor abstract base class
- A good implementation example that uses these already available physics lists can be found in
  - examples/extended/electromagnetic/TestEm2

- Once you know the desired hadronic part of the physics list name (e.g. FTFP_BERT) an easy way to keep straight the various EM options is to use the G4PhysListFactory class:
  - G4PhysListFactory factory;

```
G4VModularPhysicsList* physList =
        factory.GetReferencePhysList("FTFP_BERT_ XXX");
// where XXX = EMV or EMX or LIV or PEN
```

Using Geant4 Validation to Choose Physics Lists

- Ultimately you must choose a physics list based on how well its component processes and models perform
  - physics performance
  - CPU performance
- Geant4 provides validation (comparison to data) for most of its physics codes
  - validation is a continuing task, performed at least as often as each release
  - more validation tests added as time goes on
- To access these comparisons, go to Geant4 website
  – follow the chain: click on “Results and Publications” -> “Validation and testing” -> Validation Database: “FNAL_DB”

Geant 4

Welcome to the Geant4 Validation Repository

Please make your selection from the menu on the top

Database statistics	
Number of test setups	21
Number of test results (public and internal)	18136

List of Tests		
Name	**Description**	**Working Group**
ATLAS	shower characteristics of ATLAS Calorimeters	LHC-feedback
CMS	shower characteristics of CMS Calorimeters	LHC-feedback
HadrIon	Test of Physics Lists (thick targets, ion beams)	hadronic
HadrXS	Test of Physics Lists (cross sections)	hadronic
Hadrcap	is an analogous to Hadr00, with advanced features.	hadronic
IAEA	IAEA Benchmark of Nuclear Spallation Models	hadronic
Ndata	Test concerning developments of new nXS, it is calling HP XS as well as HPW XS.	hadronic

Welcome to the Geant4 Validation Repository.
Within each test, results may be grouped by certain criteria. Please make your selection, if applicable, from the menu on the right.

List of hadronic Tests

- HadrIon
- HadrXS
- IAEA
- Testfragm
- simplifiedCalo
- test19
- test22
- test30
  - geant4-09-06-ref-00
  - p + Fe -> n + X, 113 MeV/c
- test35
- test45
- test47
- test48
- test75

Geant 4

Login | Download | User Forum | Gallery | Site Index
Contact Us

Search Geant4

Home > Results & Publications > **Physics Validation and Verification**

Home | Validation Overview | Release Highlights | Electromagnetic | Hadronic | LHC-feedback | Expert

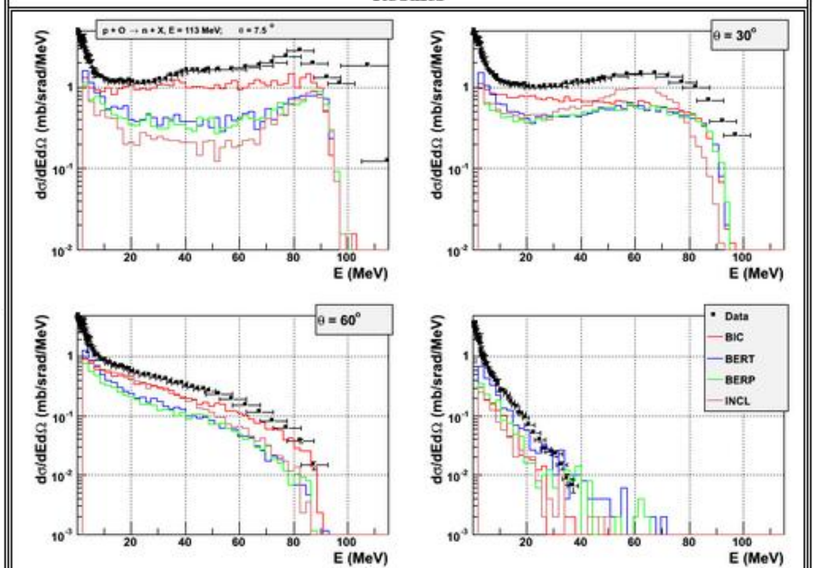
Name of the Test:	test30
Responsible:	V. Ivanchenko
Description:	Test of hadronic generators of inelastic processes

Geant4 Version:	geant4-09-06-ref-00a
Observable:	pn_o_113
Reaction:	p + O -> n + X, 113 MeV/c
Status:	public

Test Conditions

Name	**Description**
Target	Oxygen
Particle	proton
Observable	dSigma/dEdOmega
Energy	113 MeV/c
Upload date	Thu Dec 20 17:44:00 CET 2012
Description	Neutron spectra
Data Source	Meier et al., Nucl. Sci. Eng. 104, 1990
last-modified	2012-12-27 13:41:33 CST
Score:	passed
Type:	expert

Results



List of hadronic Tests

- HadrIon
- HadrXS
- IAEA
- Testfragm
- simplifiedCalo
- test19
- test22
- test30
  - geant4-09-06-ref-00
  - p + O -> n + X, 113 MeV/c
- test35
- test45
- test47
- test48
- test75

Summary

- All the particles, physics processes and production cuts needed for an application must go into a physics list
- Two kinds of physics list classes are available for users to derive from
  - G4VUserPhysicsList – for relatively simple physics lists
  - G4VModularPhysicsList – for detailed physics lists
- Some physics lists are provided by Geant4 as a starting point for users
- Care is required by user in choosing the right physics
  - use the validation, Luke