Physics and Physics List in Geant4

Giuliana Milluzzo

INFN – Laboratori Nazionali del Sud

III International Geant4 and GPU programming school

Based on a presentation by G.A.P. Cirrone (INFN-LNS) and Luciano Pandola

Outline

- The philosophy of the physics definition
- How to define and activate models
- Electromagnetic physics
- Hadronic physics

Philosophy

- Provide a **general model framework** that allows the implementation of **alternative models** to **describe the same process** (e.g. Compton scattering)

- A given model could work better in a certain energy range

MULTIPLE MODELS FOR THE SAME PROCESS

- **Decouple** modeling of cross sections and of final state generation

- Provide **processes** containing
  - Many possible models and cross sections
  - Default cross sections for each model

Models under continuous development

User Classes

Initialisation classes

Invoked at the initialization

G4VUserDetectorConstruction

G4VUserPhysicsList

Global: only one instance of them exists in memory, shared by all threads (**readonly**).
Managed only by the master thread.

G4VUserPhysicsList

All physics lists **must** derive from this class

- And then be registered to the G4(MT)RunManager in the main()
- Mandatory class in Geant4

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

User must implement the following (purely virtual) methods:

- **`ConstructParticle(), ConstructProcess(), SetCuts()`**

ConstructParticle()

- Choose the particles you need in your simulation and define all of them here
  - **`G4Electron::ElectronDefinition()`**
  - **`G4Gamma::GammaDefinition()`**
  - …
- It is possible to use Geant4 classes that create categories of particles
  - **`G4BosonConstructor()`**
  - **`G4LeptonConstructor()`**
  - …

SetCuts()

Define all **production** cuts for gamma, electrons and positrons

- Recently also for protons

This is a **production cut**, not a tracking cut

- All particles, once created, are tracked down to zero kinetic energy
- The cut is used to limit the generation of secondaries (e.g. δ-rays from ionization, or gammas from bremsstrahlung)
- The cut is expressed in equivalent range
    - This is converted in energy for each material

The definition of physics

Three different ways to implement physics models:

Explicitely associating a given model to a given particle for a given energy range

- Done at code level

Use of Builders or Constructors and reference physics lists

- THE CONSTRUCTORS
  are process related (*Electromagnetic, Hadronic, Elastic, etc.*)

- THE REFERENCE PHYSICS LISTS are complete physics lists
  Can be also called by the macro file

The definition of physics - 2

Modular physics lists: the list is built from **basic "blocks"** (the constructors)

- The constructors are process-related (standard, low energy, Bertini, etc.)
- Some constructors are provided by Geant4, but users can create and register their own **customized**

Class derives from `G4VModularPhysicsList` which inherits from `G4VUserPhysicsList`

- `SetCuts()` is the only mandatory virtual method
- `ConstructParticle()` and `ConstructProcess()` are optional

How to build a modular physics list

Create a class derived by **G4VModularPhysicsList**

- `class myList : public G4VModularPhysicsList`

Implement the mandatory method **SetCuts ()**

Register the appropriate constructors (or create your own) in the constructor or in **ConstructProcess()**

- In the first case, you cannot change at run-time

```
void myList::myList ()
{
 // Hadronic physics
  RegisterPhysics(new G4HadronElasticPhysics ());
  RegisterPhysics(new G4HadronPhysicsFTFP_BERT_TRV());
  // EM physics
  RegisterPhysics(new G4EmStandardPhysics());
}
```

How to build a modular physics list

- Other option: instantiate the constructors in **`ConstructProcess()`** and invoke their own **`ConstructProcess()`**
- Constructors made out from "elementary" builders

```
void myList::ConstructProcess()
{
  //Em physics
  G4VPhysicsConstructor* emList = new G4EmStandardPhysics();
  emList->ConstructProcess();
  //Inelastic physics for protons
  G4VPhysicsConstructor* pList = new G4HadronPhysicsQGS_BIC();
  pList->ConstructProcess(); }
```

The definition of physics

Geant4 provides a few ready-for-the-use physics lists

- Complete physics lists
- Can be instantiated by UI (macro files)

Provide a complete and realistic physics with ALL models of interest

- Many options available for EM and hadronic physics

They are intended as starting point and their builders can be reused

- They are made up of constuctors, so easy to change/replace each given block

Reference physics lists

These families share components to attach certain types of processes to groups of particles. These components are:

- Electromagnetic interactions for all particles
- Inelastic interactions
- Elastic scattering
- Capture
- Decay of unstable particles
- Specialised treatment of low energy neutrons (< 20 MeV)

- They are modular physics lists by themselves, so you can register **additional** constructors (e.g. optical physics)

How to use a Geant4 physics list

In your main(), just register an instance of the physics list to the **G4(MT)RunManager**

```
#include "QGSP_BERT.hh"
int main()
{
  // Run manager
  G4RunManager * runManager = new G4RunManager();

  …
 G4VUserPhysicsList* physics = new QGSP_BERT();
  runManager-> SetUserInitialization(physics);
}
```

Where are the constructors?

```
/usr/local/geant4/geant4.10.01.p01/geant4.10.01.p01/source/physics_lists/constructors
```

```
-rw-r--r--. 1 root root  708 Jun 23 09:25 CMakeLists.txt
-rw-r--r--. 1 root root  870 Jun 23 09:25 GNUmakefile
-rw-r--r--. 1 root root 1154 Jun 23 09:25 History
drwxr-xr-x. 4 root root 4096 Jun 23 09:25 decay
drwxr-xr-x. 4 root root 4096 Jun 23 09:25 electromagnetic
drwxr-xr-x. 4 root root 4096 Jun 23 09:25 factory
drwxr-xr-x. 4 root root 4096 Jun 23 09:25 gamma_lepto_nuclear
drwxr-xr-x. 4 root root 4096 Jun 23 09:25 hadron_elastic
drwxr-xr-x. 4 root root 4096 Jun 23 09:25 hadron_inelastic
drwxr-xr-x. 4 root root 4096 Jun 23 09:25 ions
drwxr-xr-x. 4 root root 4096 Jun 23 09:25 limiters
drwxr-xr-x. 4 root root 4096 Jun 23 09:25 stopping
```

```
-rw-r--r--. 1 root root 10923 Jun 23 09:25 G4EmDNAPhysics.cc
-rw-r--r--. 1 root root 12230 Jun 23 09:25 G4EmDNAPhysics_option1.cc
-rw-r--r--. 1 root root 10274 Jun 23 09:25 G4EmLEPTSPhysics.cc
-rw-r--r--. 1 root root 15595 Jun 23 09:25 G4EmLivermorePhysics.cc
-rw-r--r--. 1 root root 16421 Jun 23 09:25 G4EmLivermorePolarizedPhysics.cc
-rw-r--r--. 1 root root 14441 Jun 23 09:25 G4EmLowEPPhysics.cc
-rw-r--r--. 1 root root 16848 Jun 23 09:25 G4EmPenelopePhysics.cc
-rw-r--r--. 1 root root 12704 Jun 23 09:25 G4EmStandardPhysics.cc
-rw-r--r--. 1 root root 10454 Jun 23 09:25 G4EmStandardPhysicsSS.cc
-rw-r--r--. 1 root root 11586 Jun 23 09:25 G4EmStandardPhysicsWVI.cc
-rw-r--r--. 1 root root 13531 Jun 23 09:25 G4EmStandardPhysics_option1.cc
-rw-r--r--. 1 root root 14287 Jun 23 09:25 G4EmStandardPhysics_option2.cc
-rw-r--r--. 1 root root 13785 Jun 23 09:25 G4EmStandardPhysics_option3.cc
-rw-r--r--. 1 root root 16587 Jun 23 09:25 G4EmStandardPhysics_option4.cc
-rw-r--r--. 1 root root 18778 Jun 23 09:25 G4OpticalPhysics.cc
-rw-r--r--. 1 root root 12985 Jun 23 09:25 G4OpticalPhysicsMessenger.cc
```

Where are the physics lists?

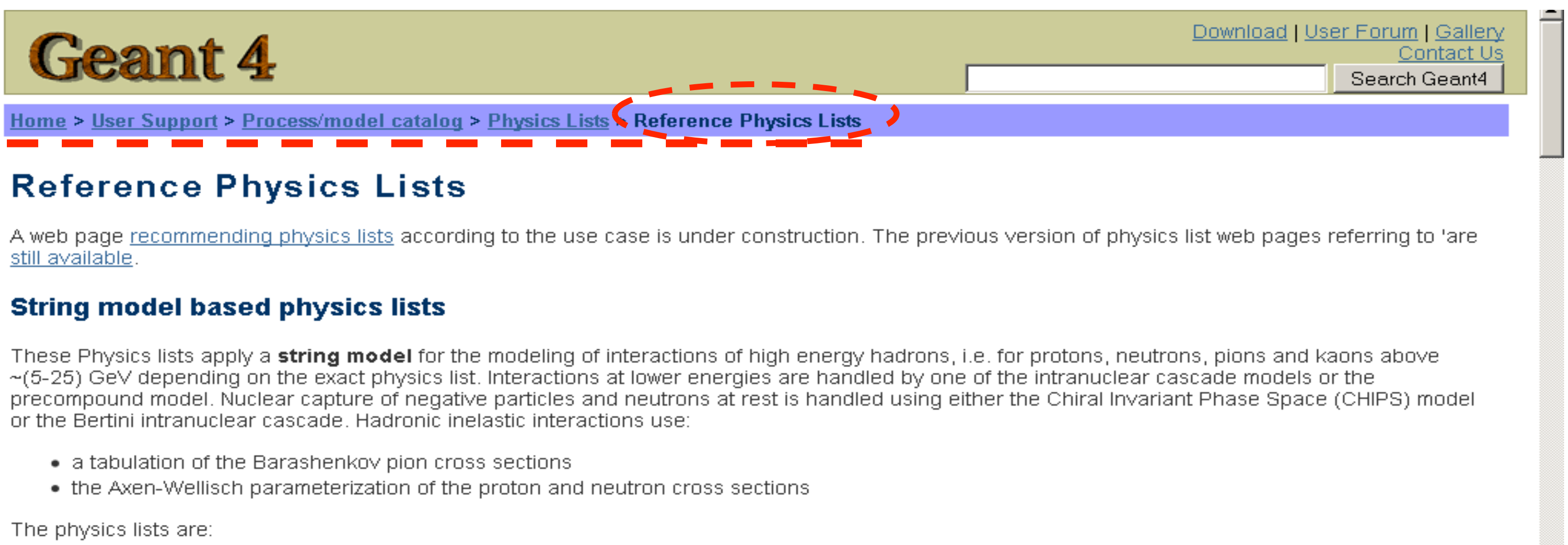
```
/usr/local/geant4/geant4.10.01.p01/geant4.10.01.p01/source/physics_lists/lists
```

```
-rw-r--r--. 1 root root  2497 Jun 23 09:25 FTFP_BERT.hh
-rw-r--r--. 1 root root  4352 Jun 23 09:25 FTFP_BERT.icc
-rw-r--r--. 1 root root  2551 Jun 23 09:25 FTFP_BERT_HP.hh
-rw-r--r--. 1 root root  3731 Jun 23 09:25 FTFP_BERT_HP.icc
-rw-r--r--. 1 root root  2595 Jun 23 09:25 FTFP_BERT_TRV.hh
-rw-r--r--. 1 root root  4438 Jun 23 09:25 FTFP_BERT_TRV.icc
-rw-r--r--. 1 root root  2185 Jun 23 09:25 FTFP_INCLXX.hh
-rw-r--r--. 1 root root  2203 Jun 23 09:25 FTFP_INCLXX_HP.hh
-rw-r--r--. 1 root root  2455 Jun 23 09:25 FTF_BIC.hh
-rw-r--r--. 1 root root  4365 Jun 23 09:25 FTF_BIC.icc
-rw-r--r--. 1 root root  2902 Jun 23 09:25 G4GenericPhysicsList.hh
-rw-r--r--. 1 root root  3760 Jun 23 09:25 G4GenericPhysicsList.icc
-rw-r--r--. 1 root root  2933 Jun 23 09:25 G4PhysListFactory.hh
-rw-r--r--. 1 root root  2995 Jun 23 09:25 G4PhysListFactoryAlt.hh
-rw-r--r--. 1 root root  3934 Jun 23 09:25 G4PhysListRegistry.hh
-rw-r--r--. 1 root root  2880 Jun 23 09:25 G4PhysListStamper.hh
-rw-r--r--. 1 root root  4184 Jun 23 09:25 INCLXXPhysicsListHelper.hh
-rw-r--r--. 1 root root  4650 Jun 23 09:25 INCLXXPhysicsListHelper.icc
```

```
// EM Physics
this->RegisterPhysics( new G4EmStandardPhysics(ver) );

// Synchroton Radiation & GN Physics
this->RegisterPhysics( new G4EmExtraPhysics(ver) );

// Decays
this->RegisterPhysics( new G4DecayPhysics(ver) );

 // Hadron Elastic scattering
this->RegisterPhysics( new G4HadronElasticPhysics(ver) );

// Hadron Physics
this->RegisterPhysics( new G4HadronPhysicsQGSP_BERT(ver));

// Stopping Physics
this->RegisterPhysics( new G4StoppingPhysics(ver) );

// Ion Physics
this->RegisterPhysics( new G4IonPhysics(ver));

// Neutron tracking cut
this->RegisterPhysics( new G4NeutronTrackingCut(ver));
```

The complete lists of Reference Physics List

FTF_BIC.hh
FTFP_BERT.hh
FTFP_BERT_HP.hh
FTFP_BERT_TRV.hh
FTFP_INCLXX.hh
FTFP_INCLXX_HP.hh
G4GenericPhysicsList.hh
G4PhysListFactory.hh
INCLXXPhysicsListHelper.hh
LBE.hh
QBBC.hh
QGS_BIC.hh
QGSP_BERT.hh
QGSP_BERT_HP.hh
QGSP_BIC.hh
QGSP_BIC_HP.hh
QGSP_FTFP_BERT.hh
QGSP_INCLXX.hh
QGSP_INCLXX_HP.hh
Shielding.hh



Geant 4

Download | User Forum | Gallery
Contact Us

Search Geant4

Home > User Support > Process/model catalog > Physics Lists > Reference Physics Lists

Reference Physics Lists

A web page recommending physics lists according to the use case is under construction. The previous version of physics list web pages referring to 'are still available.

String model based physics lists

These Physics lists apply a **string model** for the modeling of interactions of high energy hadrons, i.e. for protons, neutrons, pions and kaons above ~(5-25) GeV depending on the exact physics list. Interactions at lower energies are handled by one of the intranuclear cascade models or the precompound model. Nuclear capture of negative particles and neutrons at rest is handled using either the Chiral Invariant Phase Space (CHIPS) model or the Bertini intranuclear cascade. Hadronic inelastic interactions use:

- a tabulation of the Barashenkov pion cross sections
- the Axen-Wellisch parameterization of the proton and neutron cross sections

The physics lists are:

Where to find information?

User Support

1. Getting started
2. Training courses and materials
3. Source code
    a. Download page
    b. LXR code browser -or- draft doxygen documentation
4. Frequently Asked Questions (FAQ)
5. Bug reports and fixes
6. User requirements tracker
7. User Forum
8. Documentation
    a. Introduction to Geant4
    b. Installation Guide
    c. Application Developers Guide
    d. Toolkit Developers Guide
    e. Physics Reference Manual
    f. Software Reference Manual
9. Physics lists
    a. Electromagnetic
    b. Hadronic

Electromagnetic physics

EM concept

The same physics processes (e.g. Compton scattering) can be described by different models, that can be **alternative** or **complementary** in a given energy range

For instance: Compton scattering can be described by

- **G4KleinNishinaCompton**
- **G4LivermoreComptonModel** (specialized low-energy, based on the Livermore database)
- **G4PenelopeComptonModel** (specialized low-energy, based on the Penelope analytical model)
- **G4LivermorePolarizedComptonModel** (specialized low-energy, Livermore database with polarization)
- **G4PolarizedComptonModel** (Klein-Nishina with polarization)
- **G4LowEPComptonModel** (full relativistic 3D simulation)

Different models can be combined, so that the appropriate one is used in each given energy range (→ performance optimization)

EM concept

A physical interaction or process is described by a process class

- Naming scheme : « G4ProcessName »
- Eg. : « G4Compton » for photon Compton scattering

- A physical process can be simulated according to several models, each model being described by a model class
  - The usual naming scheme is: *« G4ModelNameProcessNameModel »*
  - Eg. : « G4LivermoreComptonModel » for the Livermore Compton model
  - Models can be alternative and/or complementary on certain energy ranges

Packages overview

Models and processes for the description of the EM interactions in Geant4 have been grouped in several packages

Package	Description
Standard	γ-rays, $e^{\pm}$ up to 100 TeV, Hadrons, ions up to 100 TeV
Muons	Muons up to 1 PeV
X-rays	X-rays and optical photon production
Optical	Optical photons interactions
High-Energy	Processes at high energy (> 10 GeV). Physics for exotic particles
Low-Energy	Specialized processes for low-energy (down to 250 eV), including atomic effects
Polarization	Simulation of polarized beams

EM processes for γ-rays, $e^{\pm}$

Particle	Process	G4Process
Photons	Gamma Conversion in $e^{\pm}$	`G4GammaConversion`
	Compton scattering	`G4ComptonScattering`
	Photoelectric effect	`G4PhotoElectricEffect`
	Rayleigh scattering	`G4RayleighScattering`
$e^{\pm}$	Ionisation	`G4eIonisation`
	Bremsstrahlung	`G4eBremsstrahlung`
	Multiple scattering	`G4eMultipleScattering`
e^{+}	Annihilation	`G4eplusAnnihilation`

EM processes muons

Particle	Process	G4Process
$\mu^{\pm}$	Ionisation	`G4MuIonisation`
	Bremsstrahlung	`G4MuBremsstrahlung`
	Multiple scattering	`G4MuMultipleScattering`
	$e^{\pm}$ pair production	`G4MuPairProduction`

Only one model available for these processes (but in principle users may write *their own* models, if needed)

Standard models

- Complete set of models for **$e^{\pm}$, γ, ions, hadrons, $\mu^{\pm}$**
- Tailored to requirements from HEP applications
  - "Cheaper" in terms of CPU
  - Include high-energy corrections (e.g. LPM), assumptions made in the low-energy regime
- Theoretical or phenomenological models
  - Bethe-Bloch, corrected Klein-Nishina, …
  - Photoabsorption Ionization (PAI)
    - ionization energy loss of a relativistic charged particle in matter
- Specific high-energy extensions available
  - Extra processes, as $\gamma \rightarrow \mu+\mu-$, $e^+e^- \rightarrow \mu+\mu-$
- Dedicated sub-library for optical photons
  - Produced by scintillation or Cherenkov effect

Livermore (& polarized) models

- Based on publicly available evaluated data tables from the Livermore data library: e^-, γ
  - EADL : Evaluated Atomic Data Library, EEDL : Evaluated Electrons Data Library, EPDL97 : Evaluated Photons Data Library, Binding energies: Scofield
  - Mixture of experiments and theories
  - In principle, tables go down to ~10 eV
- Applications: medical, underground and rare events, space
- **Polarized** models
  - Same calculation of the cross section, different way to produce the final state
  - Describe in detail the kinematics of polarized photon interactions
  - Application: space missions for the detection of polarized photons

Penelope models

- Geant4 includes the low-energy models for electrons, positrons and photons from the Monte Carlo code PENELOPE (PENetration and Energy LOss of Positrons and Electrons)
  - Nucl. Instr. Meth. B 207 (2003) 107
  - Geant4 implements v2008 of Penelope
- Physics models specifically developed by the group of F. Salvat et al.
  - Great care dedicated to the low-energy description
  - Atomic effects, fluorescence, Doppler broadening...
- Mixed approach: analytical, parameterized and database-driven
  - Applicability energy range: 100 eV – 1 GeV
- Include **positrons**
  - Not described by Livermore models

When/why to use Low Energy Models

- **Use** Low-Energy models (Livermore or Penelope), as an *alternative* to Standard models, when you:
  - need precise treatment of EM showers and interactions at low-energy (keV scale)
  - are interested in atomic effects, as fluorescence x-rays, Doppler broadening, etc.
  - can afford a more CPU-intensive simulation
  - want to cross-check an other simulation (e.g. with a different model)
- **Do not use** when you are interested in EM physics > MeV
  - same results as Standard EM models, performance penalty

Example: PhysicsList for γ-rays

```
G4ProcessManager* pmanager =
   G4Gamma::GetProcessManager();
pmanager->AddDiscreteProcess(new
  G4PhotoElectricEffect);
pmanager->AddDiscreteProcess(new G4ComptonScattering);
pmanager->AddDiscreteProcess(new G4GammaConversion);
pmanager->AddDiscreteProcess(new
  G4RayleighScattering);
```

Only PostStep

- Use **AddDiscreteProcess** because γ-rays processes have only PostStep actions

EM Physics Constructors for Geant4 10.1 - ready-for-the-use

G4EmStandardPhysics – default
G4EmStandardPhysics_option1 – HEP fast but not precise
G4EmStandardPhysics_option2 – Experimental
G4EmStandardPhysics_option3 – medical, space
G4EmStandardPhysics_option4 – optimal mixture for precision
G4EmLivermorePhysics
G4EmLivermorePolarizedPhysics
G4EmPenelopePhysics
G4EmLowEPPhysics
G4EmDNAPhysics

Combined Physics
Standard > 1 GeV
LowEnergy < 1 GeV

- **`$G4INSTALL/source/physics_list/constuctors`**
- Advantage of using of these classes – they are tested on regular basis and are used for regular validation

How to extract Physics ?

Possible to retrieve physics quantities via **`G4EmCalculator`** or directly from the physics models

- Physics List should be initialized

Example for retrieving the **total cross section** (cm^{-1}) of a process with name *procName*: for particle *partName* and material *matName*

```
G4EmCalculator emCalculator;
G4Material* material =
  G4NistManager::Instance()->FindOrBuildMaterial("matName);
G4double massSigma =
  emCalculator.ComputeCrossSectionPerVolume
  (energy,particle,procName,material);
G4cout << G4BestUnit(massSigma, "Surface/Volume") <<
  G4endl;
```

A good example: **`$G4INSTALL/examples/extended/electromagnetic/TestEm14`**

Hadronic physics

Hadronic Physics

- Data-driven models
- Parametrised models
- Theory-driven models

Hadronic physics challenge

Three energy regimes

- < 100 MeV
- 100 MeV - 10 GeV, resonance and cascade region
- > 20 GeV (QCD strings)

Within each regime there are several models

Many of these are phenomenological

Reference physics lists for Hadronic interactions

Three families of builders

- **QGS**, or list based on a model that use the Quark Gluon String model for high energy hadronic interactions of protons, neutrons, pions and kaons
- **FTF**, based on the FTF (FRITIOF like string model) for protons, neutrons, pions and kaons
- **Other** specialized physics lists

Hadronic processes

- At rest
  - Stopped muon, pion, kaon, anti-proton
  - Radioactive decay
  - Particle decay (decay-in-flight is PostStep)
- Elastic
  - **Same process** to handle all long-lived hadrons (multiple models available)
- Inelastic
  - **Different processes** for each hadron (possibly with multiple models vs. energy)
  - Photo-nuclear, electro-nuclear, mu-nuclear
- Capture
  - Pion- and kaon- in flight, neutron
- Fission

Cross sections

Default cross section sets are provided for each type of hadronic process:

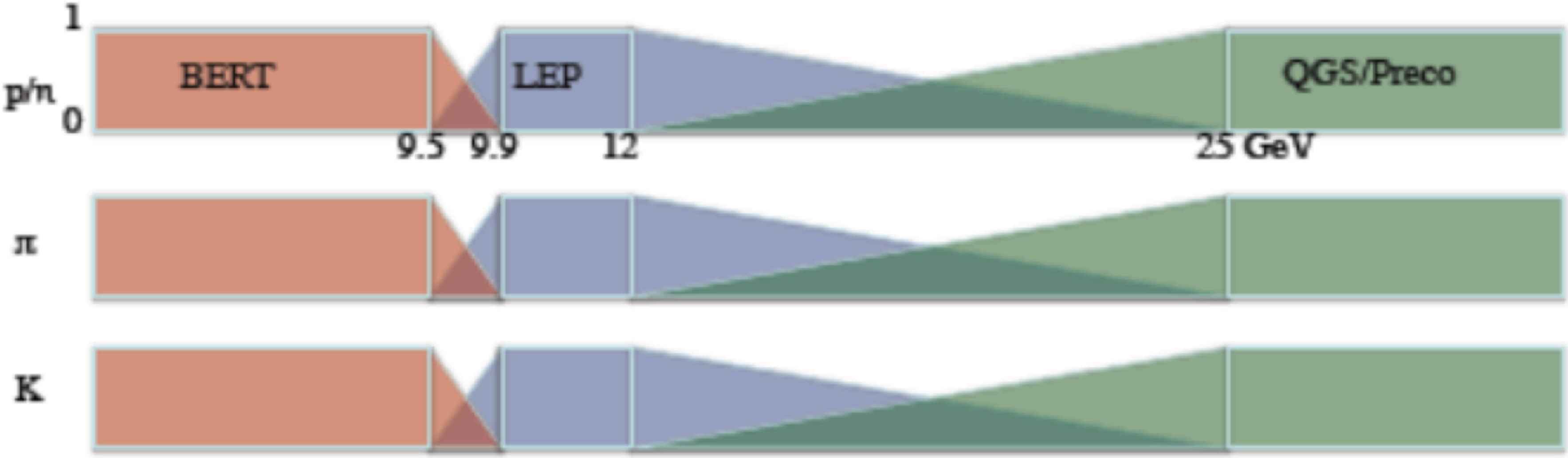
- Fission, capture, elastic, inelastic

- Low energy neutrons
  - elastic, inelastic, fission and capture (< 20 MeV)
- Neutron and proton inelastic cross sections
  - 20 MeV < E < 20 GeV
- Ion-nucleus reaction cross sections (several models)
  - Good for E/A < 1 GeV
- Isotope production data
  - E < 100 MeV
- Photo-nuclear cross sections

Neutron HP Models

- Transport of **low-energy neutrons** in matter:
  - The energy coverage of these models is from thermal energies to 20 MeV
  - The modeling is based on the data formats of ENDF/B-VI, and all distributions of this standard data format are implemented
  - Includes cross sections and final state information for *elastic* and *inelastic scattering, capture*, *fission* and *isotope production*
  - The file system is used in order to allow granular access to, and flexibility in, the use of the cross-sections for different isotopes, and channels
  - Code in sub-directory: **`/source/processes/hadronic/models/neutron_hp`**

Hadronic models match – inelastic interactions



Old picture: LEP models dismissed meanwhile: BERT interfaced directly to QGS/Preco at 9.9 GeV

Code Example

```
G4ParticleDefinition* neutron=
  G4Neutron::NeutronDefinition();
G4ProcessManager* protonProcessManager =
  proton->GetProcessManager();

// Elastic scattering
G4HadronElasticProcess* neutronElasticProcess =
 new G4HadronElasticProcess();

G4NeutronHPElastic* neutronElasticModel =
 new G4NeutronHlastic();
neutronElasticModel->SetMaxEnergy(20.*MeV);
neutronElasticProcess->
 RegisterMe(neutronElasticModel);

neutronProcessManager->
 AddDiscreteProcess(neutronElasticProcess);
```

retrieve the process manager for neutron

create the **process** for elastic scattering

get the HP **model** for elastic scattering

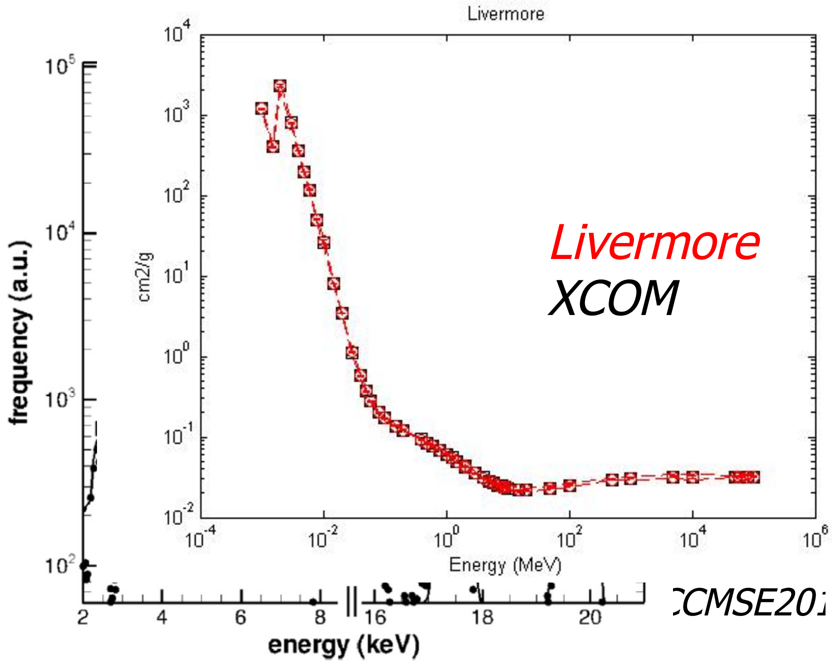
register the model to the process

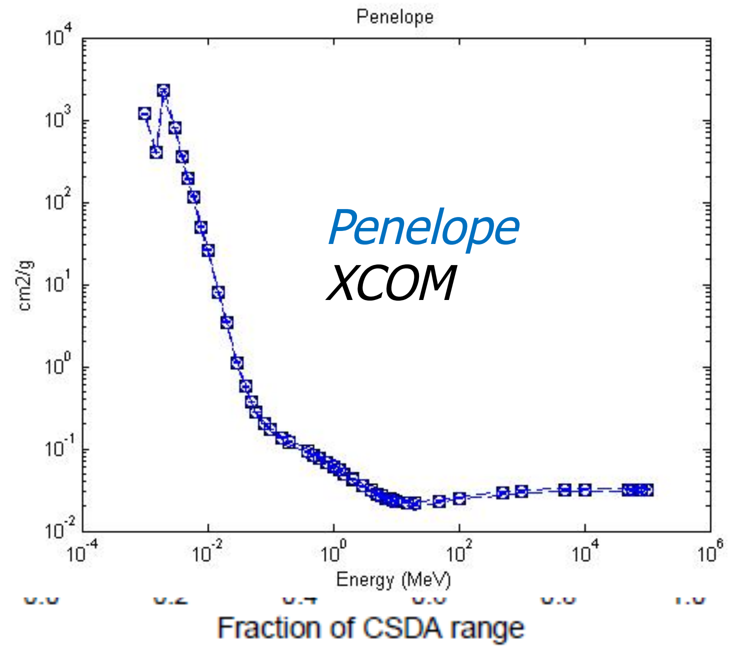
attach the process to neutron

Quick overview of validation

EM validation - 1

- Tens of papers and studies available
  - Geant4 Collaboration + User Community
- Results can depend on the specific observable/reference
  - Data selection and assessment critical

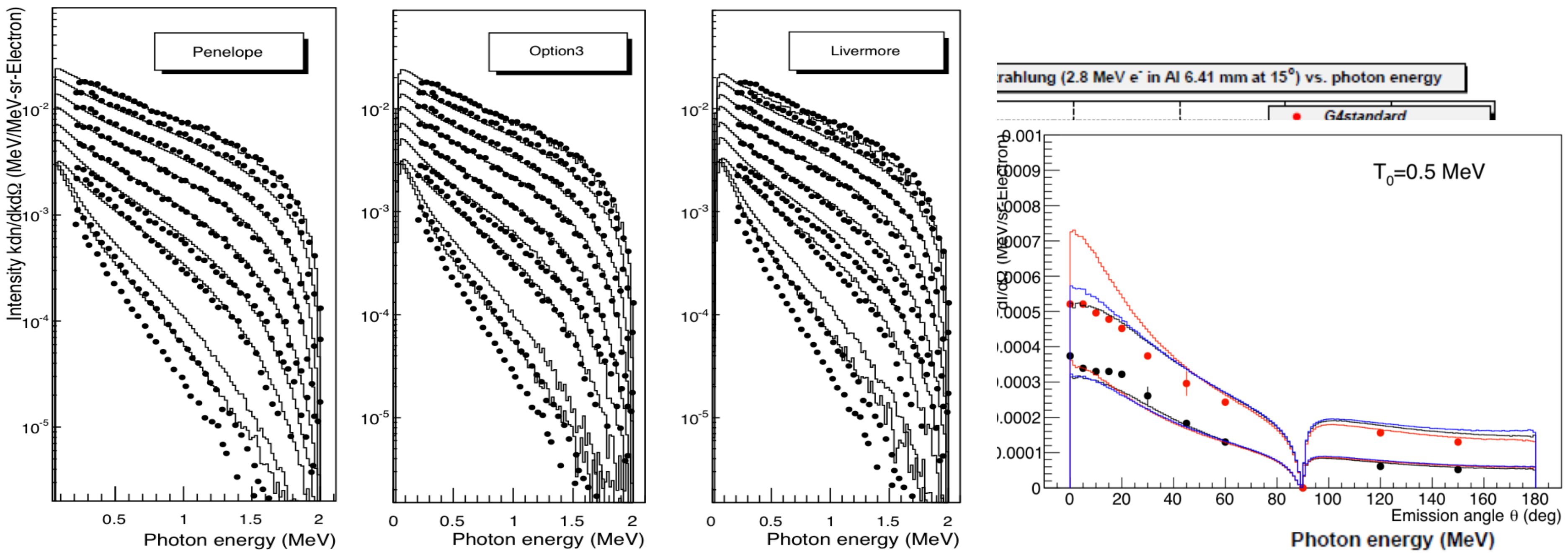




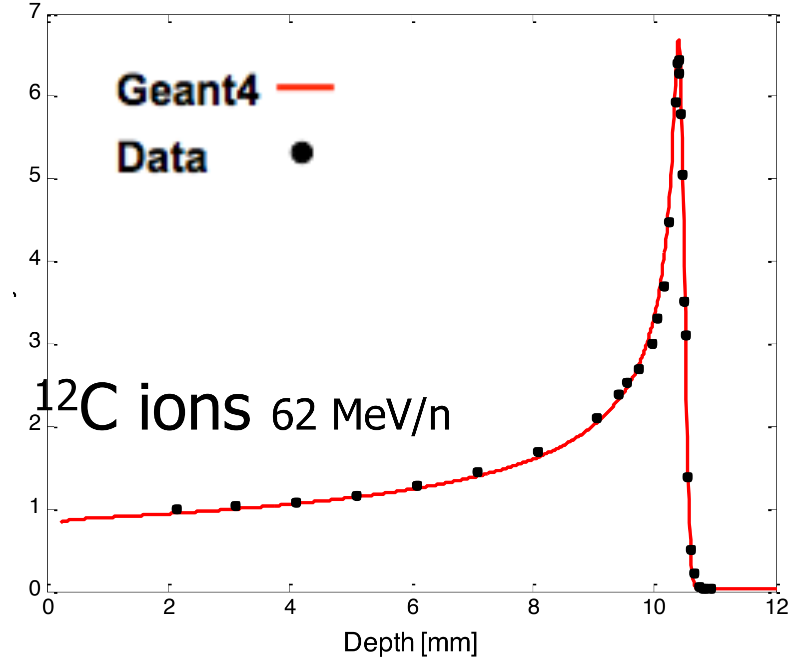
EM validation – 2

- In general satisfactory agreement
- Validation/verification repository available on web

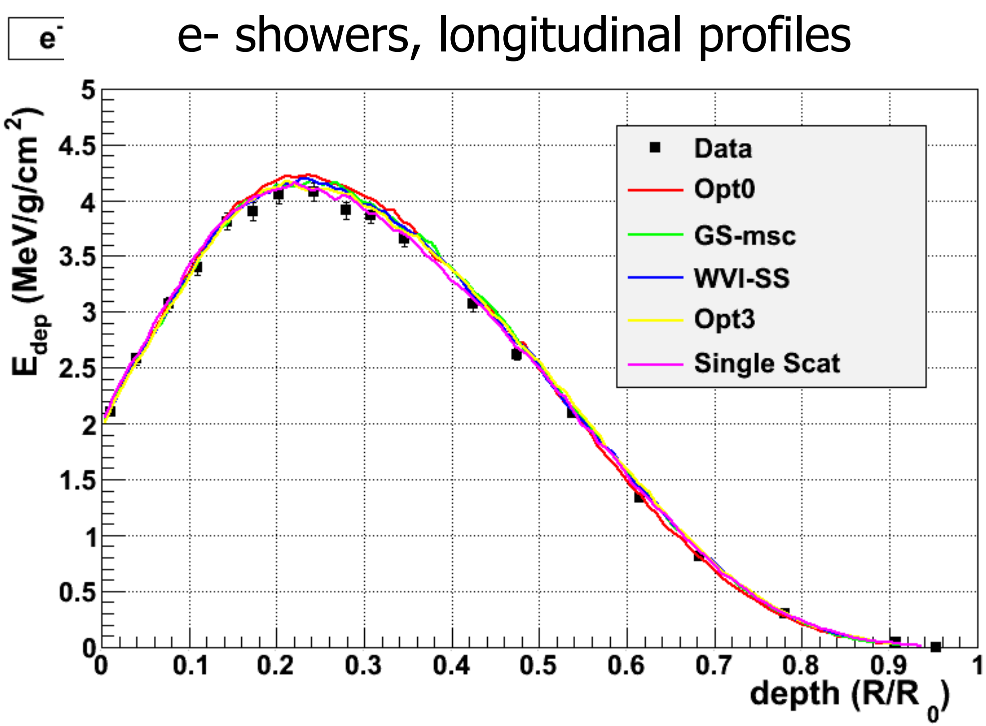
http://cern.ch/vnivanch/verification/verification/electromagnetic/

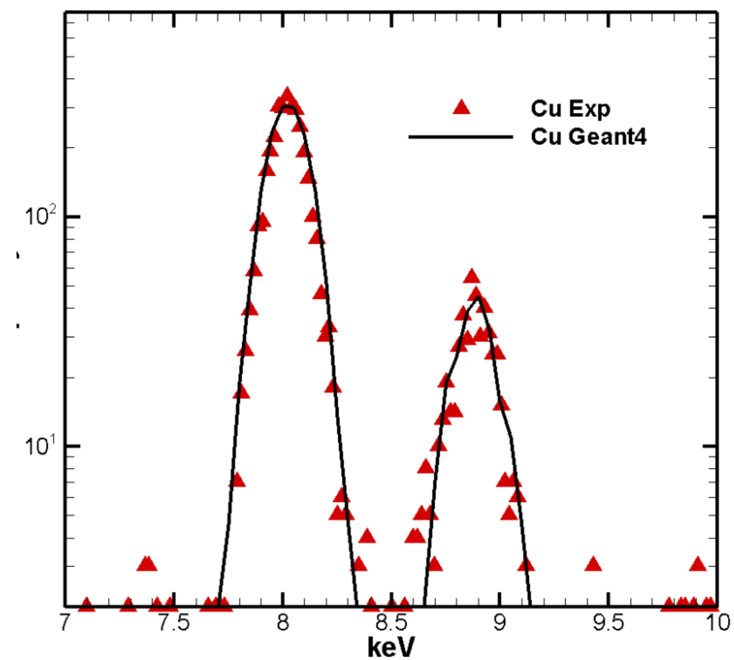


EM validation - 3



e- showers, longitudinal profiles





Hadronic validation

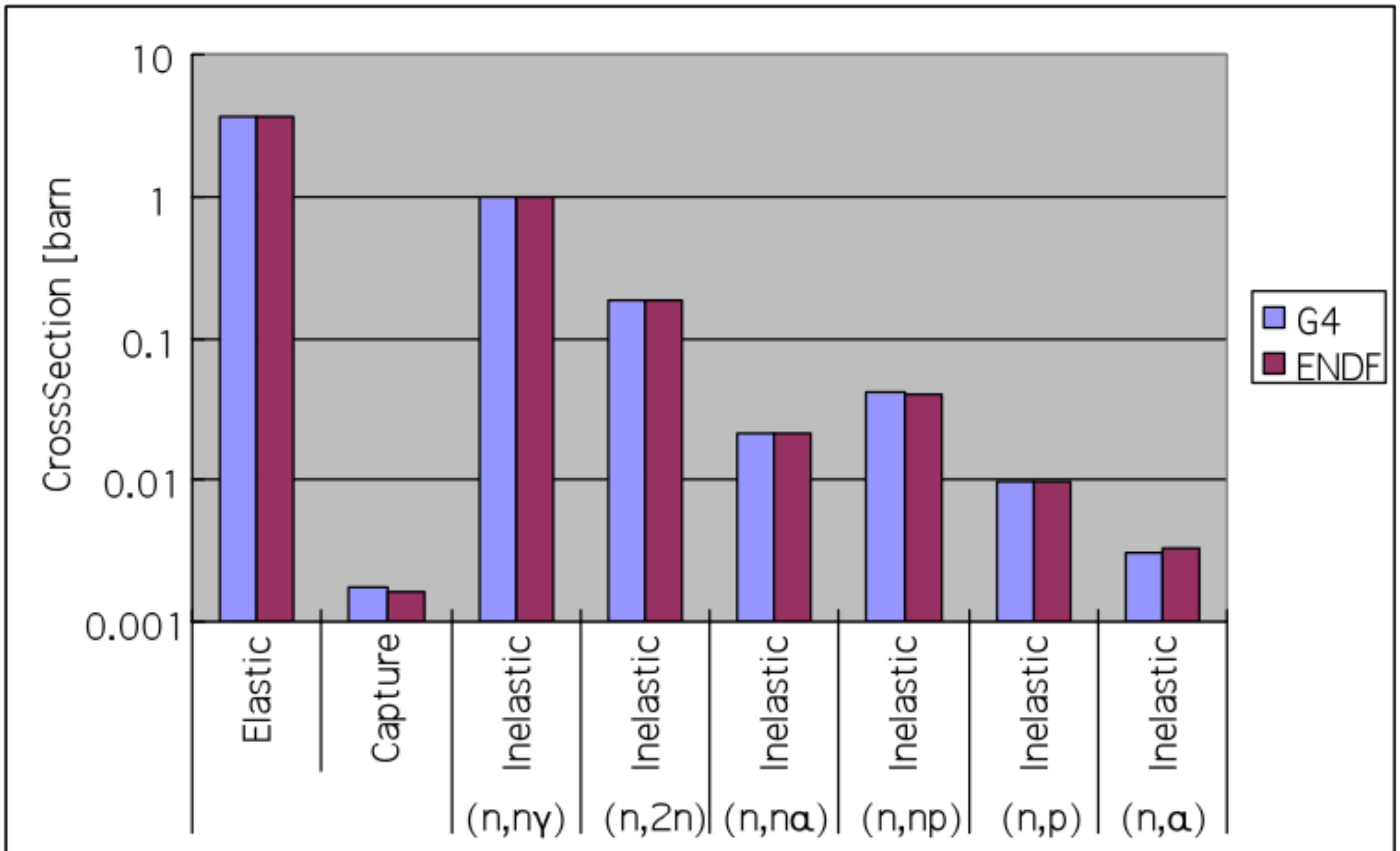
- A website is available to collect relevant information for validation of Geant4 hadronic models (plots, tables, references to data and to models, etc.)

http://geant4.cern.ch/results/validation_plots.htm

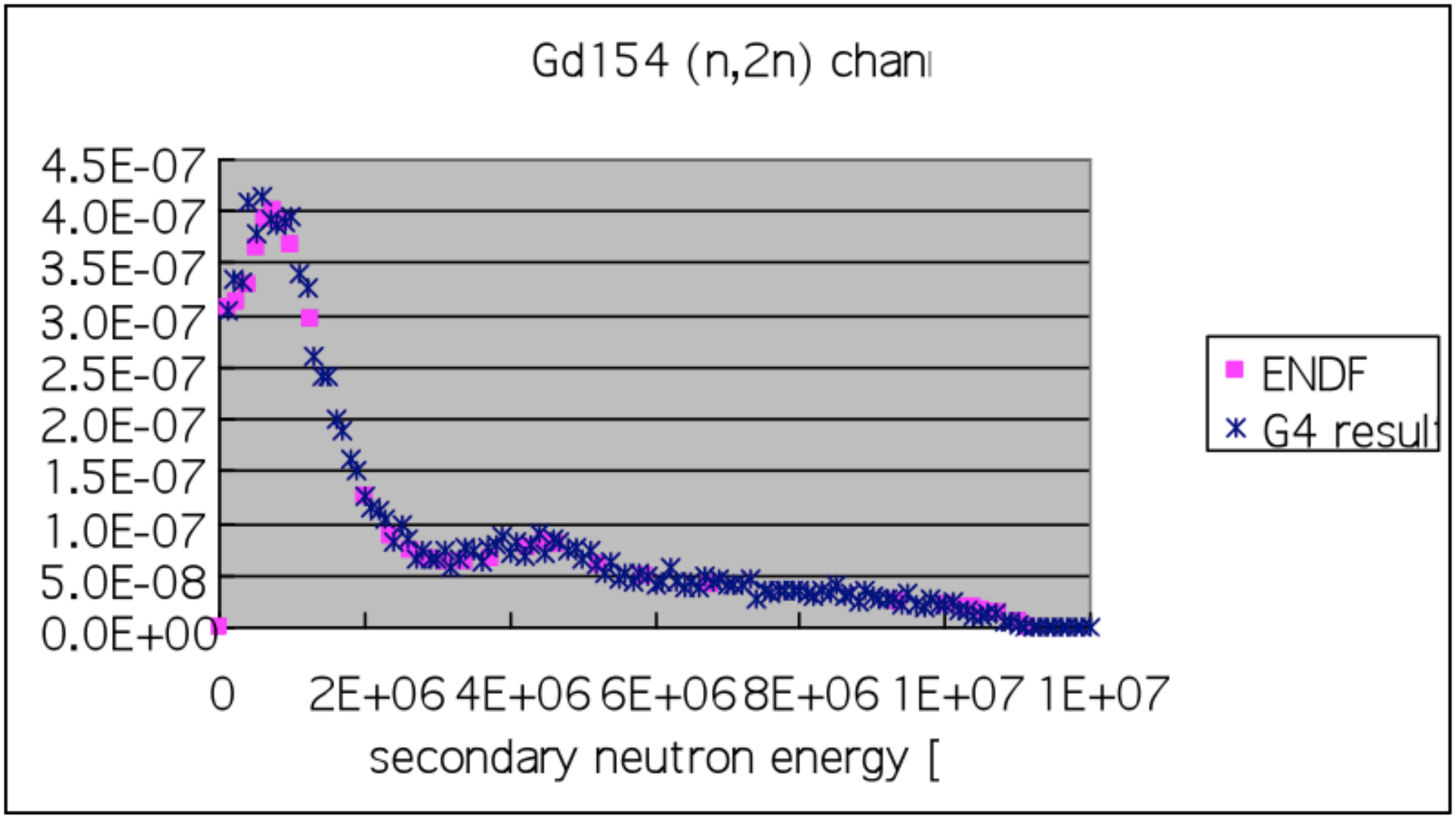
http://g4validation.fnal.gov:8080/G4ValidationWebApp/

- Several physics lists and several use-cases have been considered (e.g. thick target, stopped particles, low-energy)
- Includes final states and cross sections

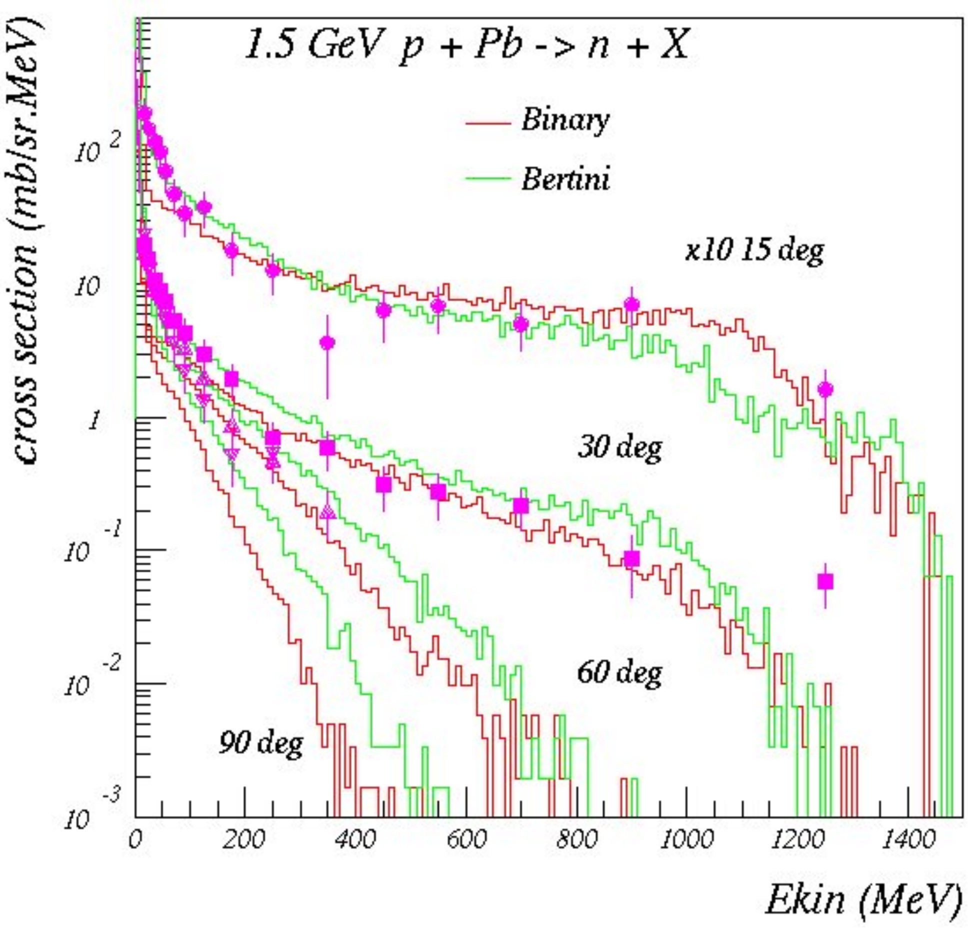
Some verification: channel cross section



Some verification: secondary energy spectrum

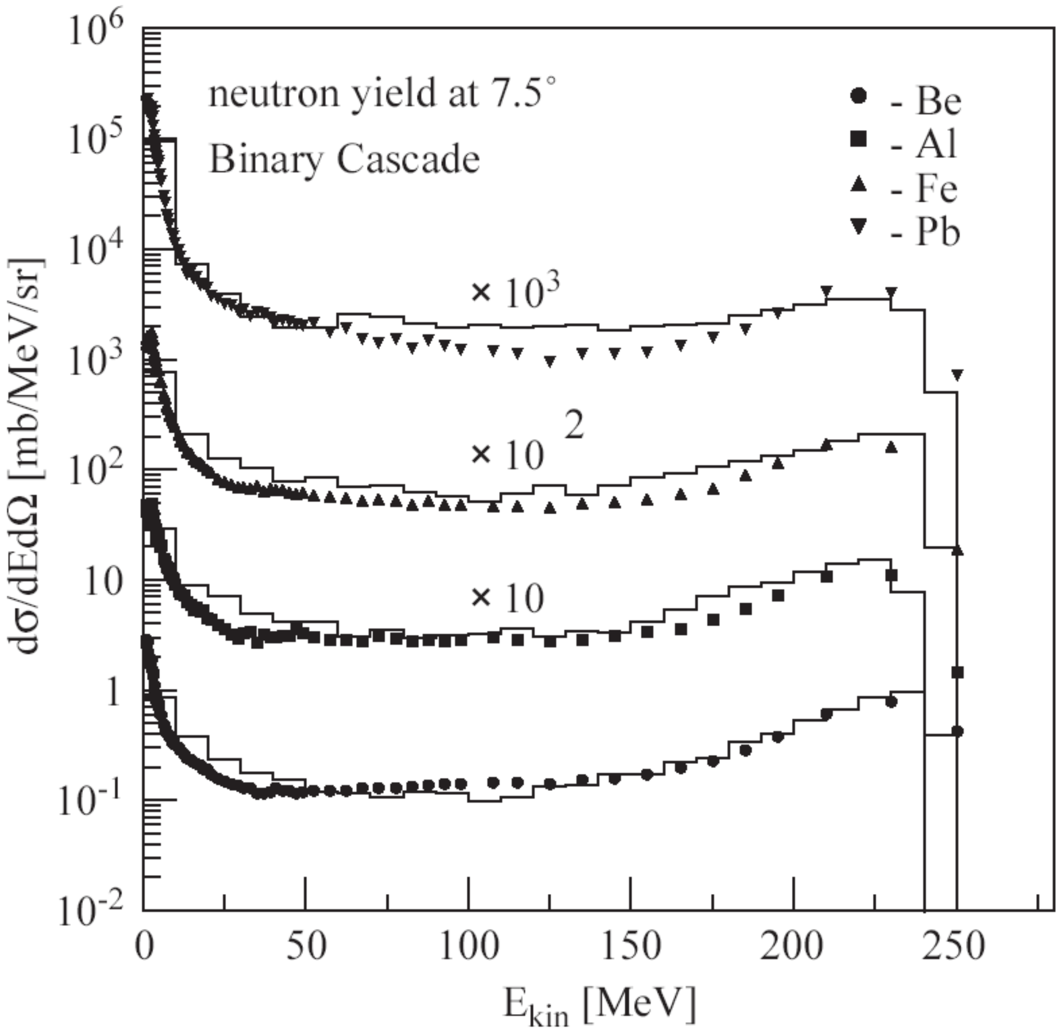


Nuclear fragmentation



Bertini and **Binary cascade** models: neutron production vs. angle from 1.5 GeV protons on Lead

Neutron production by protons



Binary cascade model: double differential cross-section for neutrons produced by 256 MeV protons impinging on different targets