Geant4: an overview of physics

Luciano Pandola

INFN

Part I: build a physics list

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.

Action classes

Invoked during the execution loop

- G4VUserActionInitialization
  - G4VUserPrimaryGeneratorAction
  - G4UserRunAction (*)
  - G4UserEventAction
  - G4UserTrackingAction
  - G4UserStackingAction
  - G4UserSteppingAction

Local: an instance of each action class exists **for each thread**.

(*) Two RunAction's allowed: one for master and one for threads

Why a physics list?

- *"Physics is physics – shouldn't Geant4 provide, as a default, a complete set of physics that everybody can use?"*
- **NO:**
  - Software can only capture Physics through a modelling
    - **No unique** Physics modelling
    - Very much the case for hadronic physics
    - But also the electromagnetic physics
    - Existing models still evolve and new models are created
  - Some modellings are more suited to some energy ranges
    - Medical applications not interested in multi-GeV physics in general
    - HEP experiments not interested in effects due to atomic shell structure
- Computation speed is an issue
  - a user may want a less-detailed, but faster approximation

Philosophy

- Provide a **general model framework** that allows the implementation of **complementary/alternative models** to **describe the same process** (e.g. Compton scattering)
  - A certain model could work better in a certain energy range

- **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

G4VUserPhysicsList

- All physics lists **must** derive from this class
  - And then be registered to the G4(MT)RunManager
  - 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 use Geant4 classes that create categories of particles
  - **`G4BosonConstructor()`**
  - **`G4LeptonConstructor()`**
  - ...

SetCuts()

- Define all **production** cuts for gamma, electrons and positrons
  - Recently also for protons
- Notice: 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 - 1

- At the beginning of Geant4 the philosophy was: "the user is in charge for deciding and implemented the most suitable models for his/her own application"
  - Completely transparent physics (no black box!)
  - Complicated to known and assess the validity of many models
- Long **"flat"** physics lists:
  - Explicitely associating a given model to a given particle for a given energy range
    - Done at code level (requires C++ coding)
- Still a possibility
  - Provided you know what you are doing

The definition of physics - 2

- Modular physics lists: the list is built from **basic "blocks"** (constructors)
  - The constructors are process-related (standard, lowenergy, Bertini, etc.)
    - Allows **mix-and-match** done by the user
  - Some constructors 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

Builder with the G4VModularPhysicsList

- **`AddTransportation()`** automatically called
- Allows the definition of "physics modules" for a given process
  - Electromagnetic
  - Hadronic
  - Decay, Optical physics, Ion physics
- User customized constructors can be created, derived class from **`G4VPhysicsConstructor`**
- Modules can be registered using the method **`RegisterPhysics()`**
  - Can be done at *run-time* (i.e. select physics via macro)

How to build a modular physics list - 1

- 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 - 2

- 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();
```

- **`$G4INSTALL/source/physics_lists/constructors`**

The definition of physics - 3

- 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
- Provided according to some use-cases
  - 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 hadronic interactions
  - Elastic scattering (hadronic)
  - 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);
}
```

The complete lists of Reference Physics List

$G4INSTALL/source/physics_lists/lists

```
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:

Part II: particles, processes and cuts

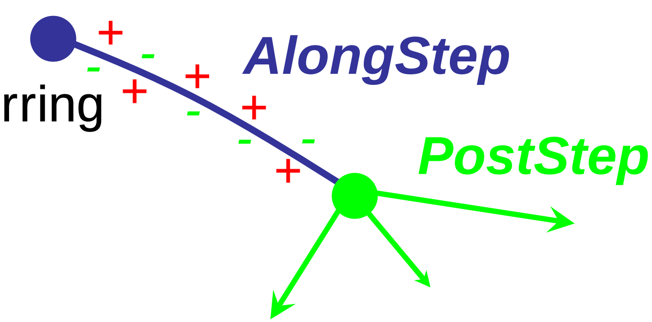
Particles: basic concepts

- There are three levels of class to describe particles in Geant4:
  - **G4ParticleDefinition**
    - define a particle
    - aggregates information to characterize a particle's static properties (name, mass, spin, etc...)
  - **G4DynamicParticle**
    - describe a particle interacting with materials
    - aggregates information to describe the dynamic of particles (energy, momentum, polarization, etc...)
  - **G4Track**
    - describe a particle travelling in space and time
    - includes all the information for tracking in a detector simulation (position, step, current volume, track ID, parent ID, etc...)

G4VProcess

- Physics processes are derived from the **G4VProcess** base class
- Abstract class defining the common interface of all processes in Geant4:
  - Used by all physics processes (also by the transportation ...)
- Defines three kinds of actions:
  - **AtRest** actions:
    - Decay, e+ annihilation ...

  - **AlongStep** actions:
    - To describe continuous (inter)actions, occurring along the path of the particle, like ionisation

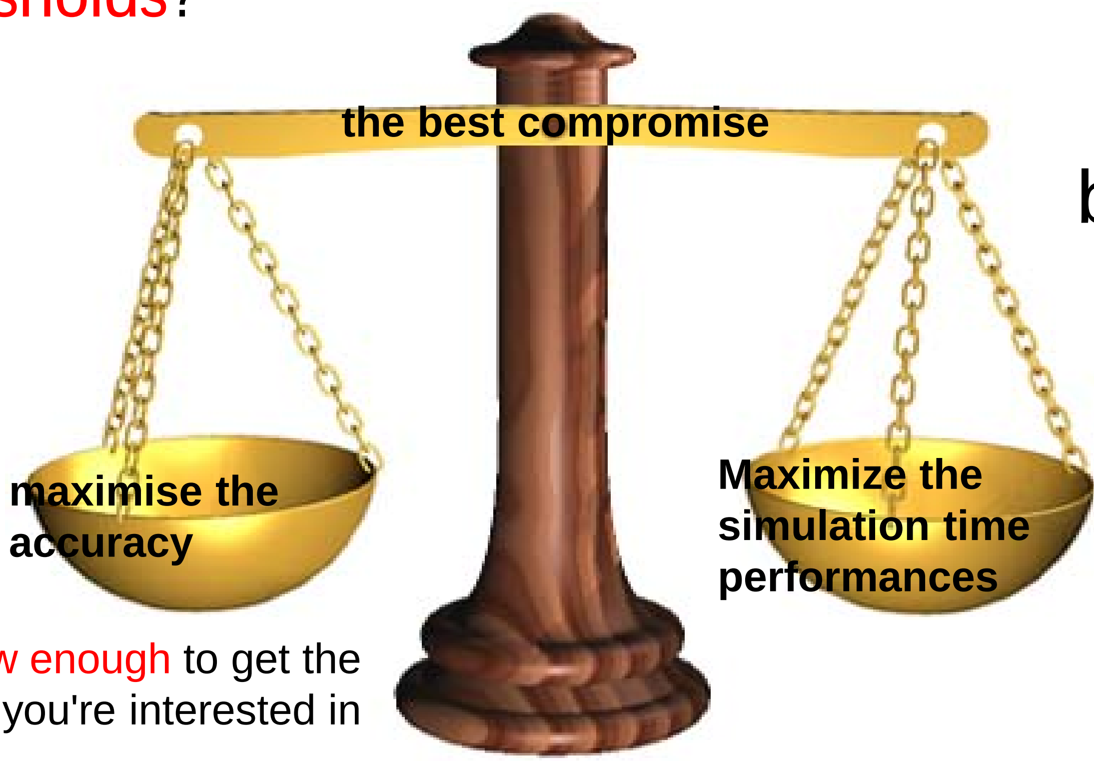


  - **PostStep** actions:
    - For describing point-like (inter)actions, like decay in flight, hadronic interactions ...

Production thresholds

- Each simulation developer must answer the question: how low in energy can you go?
  - should I produce (and track) everything or consider thresholds?

This is a balancing act:



need to go low enough to get the physics you're interested in

can't go too low because some processes have infrared divergence causing huge CPU time

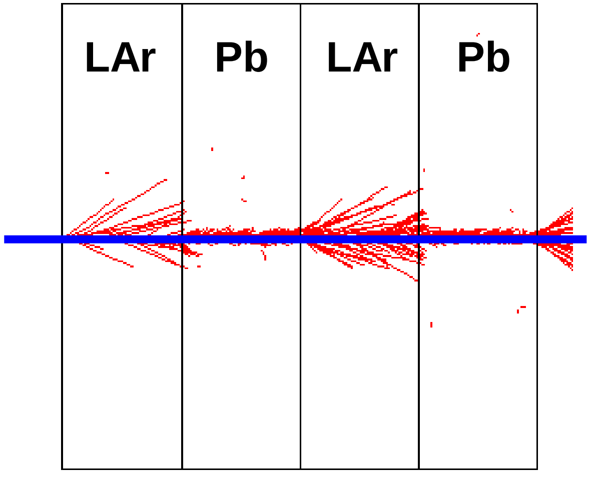
Production thresholds: mixed simulation

- In Geant4 there are **no tracking cuts**
  - particles are tracked down to a zero range/kinetic energy
- Only production cuts exist
  - i.e. cuts allowing a secondary particle to be born or not
  - Applied to: gamma, electron, positron, proton
- Why are production cuts needed ?
  - Some electromagnetic processes involve infrared divergences
    - this leads to a huge number of smaller and smaller energy photons/electrons (such as in Bremsstrahlung, δ-ray production)
    - production cuts limit this production to particles above the threshold
  - the remaining part is treated as a **continuous effect** (i.e. AlongStep action)

Geant4 way for production thresholds

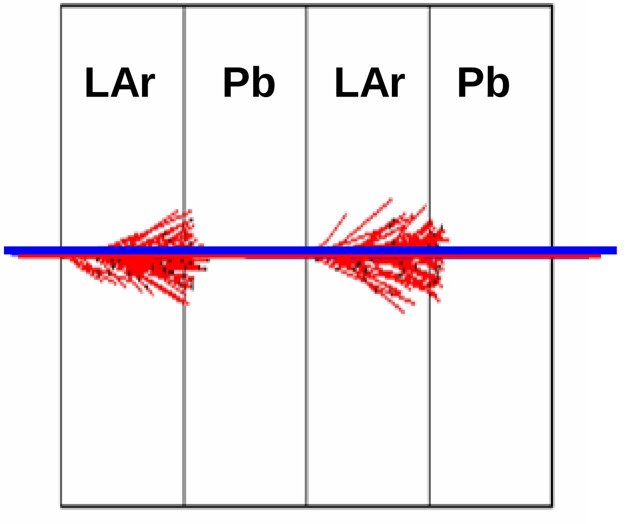
- Geant4 solution: impose a “range” production threshold
  - this threshold is a distance, not an energy
  - default = 1 mm
  - the primary particle loses energy by producing secondary particles which can travel at least the given distance
  - if primary no longer has enough energy to produce secondaries which travel at least 1mm, two things happen:
    - discrete energy loss ceases (no more secondaries produced)
    - the primary is tracked down to zero energy using continuous energy loss
- Stopping location is therefore correct
- Only one value of production threshold distance is needed for all materials because it corresponds to different energies depending on material

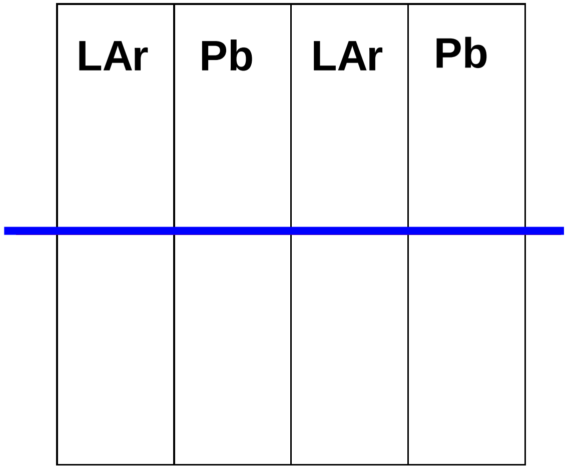
Production threshold: cut in range



Cut = 455 keV
(range in LAr = 1.5 mm)

500 MeV p in LAr-Pb sampling calorimeter





Cut = 2 MeV

Threshold in range: 1.5 mm

455 keV electron energy in liquid Ar

2 MeV electron energy in Pb

Cuts per region

- In a complex detector there may be **many different types** of sub-detectors involving
  - very small or segmented sensitive materials (e.g. a Si tracker)
  - large, undivided volumes (e.g. a calorimeter)
  - inert materials
- The same value of the secondary production threshold may **not** be appropriate for all of these
  - user can define regions of similar properties and assign a different set of production thresholds (cuts) to each
  - Equivalent to require a different tracking (spatial) **precision** in the different regions
- This feature is very useful (and CPU-saving!) when simulating complex detectors

Part III: electromagnetic and hadronic physics

EM concept - 1

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

EM concept - 2

- 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
  - Refer to the Geant4 manual for the full list of available models

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`

Inventory (and specs) of the models for γ-rays

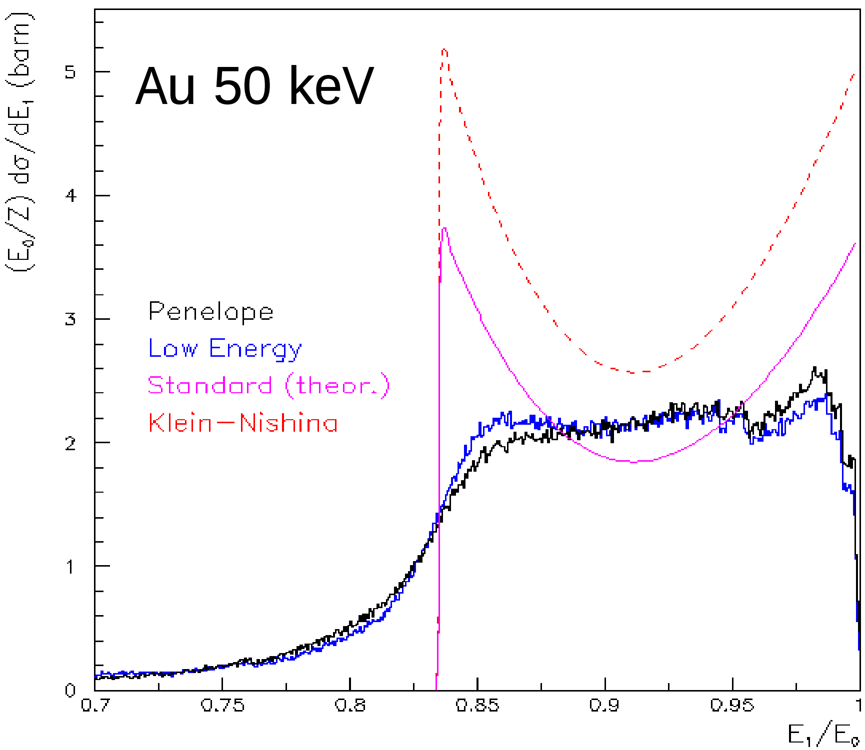
1 MeV γ in Al

- Many models available for each process
  - Plus one full set of polarized models
- Differ for energy range, precision and CPU speed
  - Final state generators
- Different mixtures available the Geant4 EM constructors

Model	E_{min}	E_{max}	CPU
G4LivermoreRayleighModel	100 eV	10 PeV	1.2
G4PenelopeRayleighModel	100 eV	10 GeV	0.9
G4KleinNishinaCompton	100 eV	10 TeV	1.4
G4KleinNishinaModel	100 eV	10 TeV	1.9
G4LivermoreComptonModel	100 eV	10 TeV	2.8
G4PenelopeComptonModel	10 keV	10 GeV	3.6
G4LowEPComptonModel	100 eV	20 MeV	3.9
G4BetheHeitlerModel	1.02 MeV	100 GeV	2.0
G4PairProductionRelModel	10 MeV	10 PeV	1.9
G4LivermoreGammaConversionModel	1.02 MeV	100 GeV	2.1
G4PenelopeGammaConversionModel	1.02 MeV	10 GeV	2.2
G4PEEFluoModel	1 keV	10 PeV	1
G4LivermorePhotoElectricModel	10 eV	10 PeV	1.1
G4PenelopePhotoElectricModel	10 eV	10 GeV	2.9

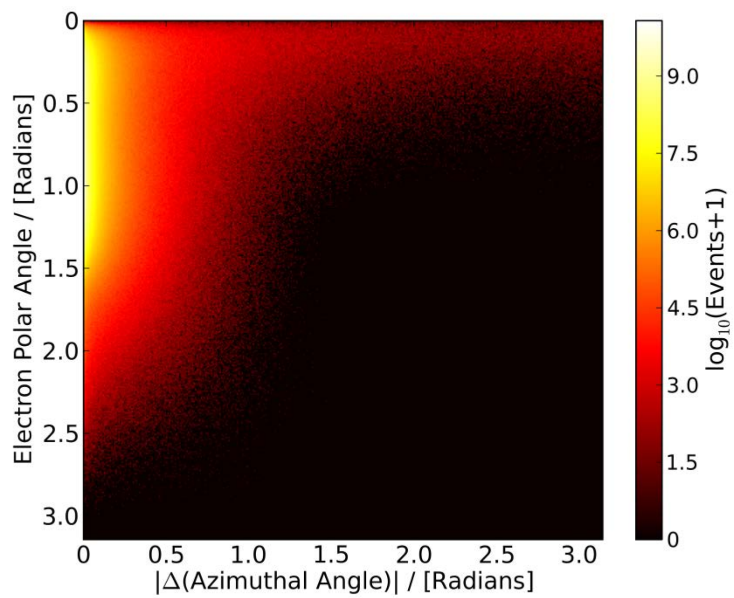
Similar situation for **$e^{\pm}$**

For example: Compton scattering



- New model: **G4LowEPComptonModel** (Monash U.)
  - Two-body relativistic 3-dim framework
  - Relativistic impulse approximation
  - Bound atomic electrons
  - Electron distribution not uniform in φ wrt photon scattering plane

250 keV γ Pb



CPU time is the **price to pay** for better precision

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

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

EM Physics Constructors for Geant4 10.0 - 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

Hadronic Physics

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

Hadronic physics challenge

- Three energy regimes
  - < 100 MeV
  - resonance and cascade region (100 MeV - 10 GeV)
  - > 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
- Up to Geant4 9.6**: LEP** and **HEP**
  - parameterised modelling of hadronic interactions
    - Based on the old GEISHA package of Geant3
  - Deprecated as obsolete, dismissed from version 10.0

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

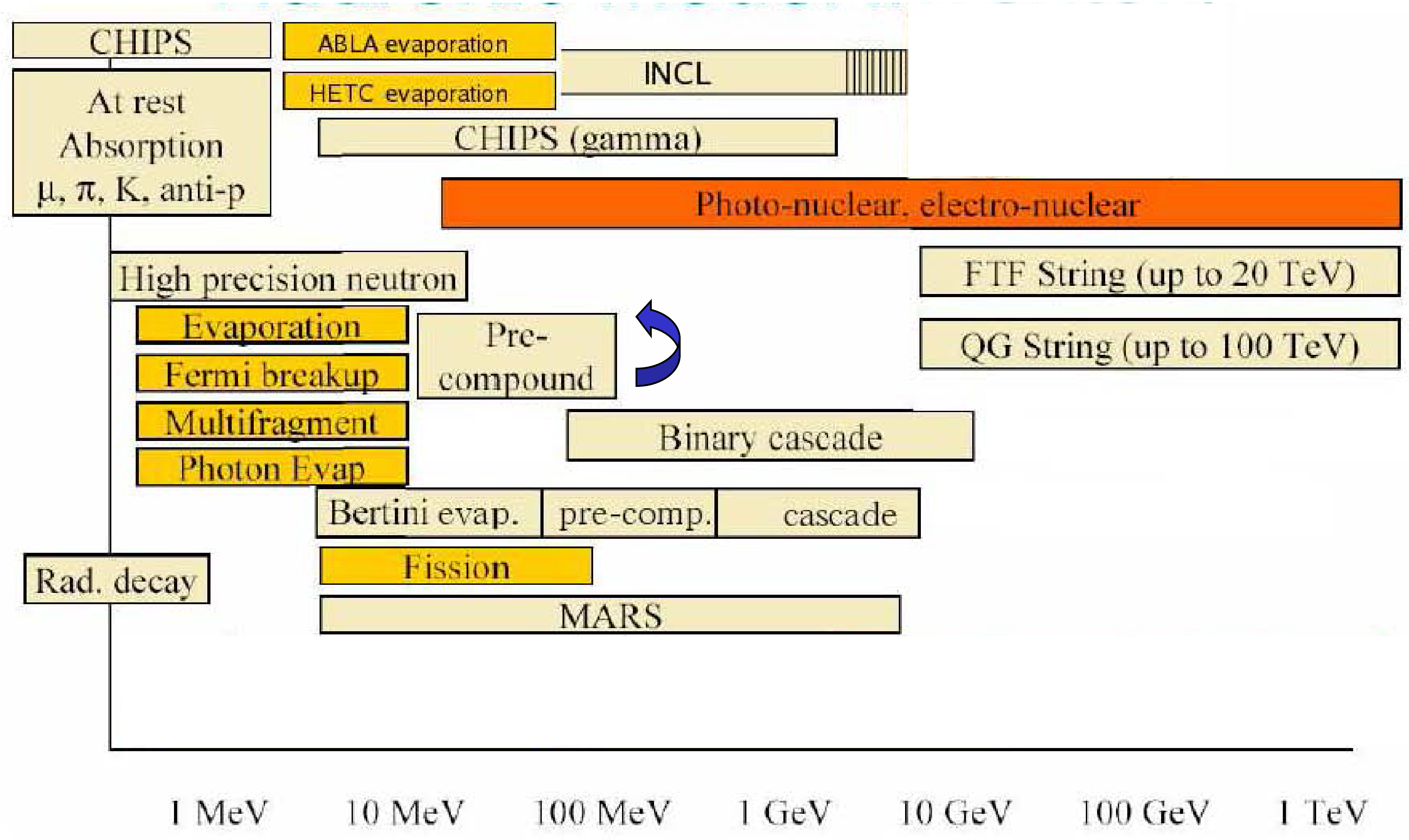
- Default cross section sets are provided for each type of hadronic process:
  - Fission, capture, elastic, inelastic
- Can be **overridden** or **completely replaced**
- Different types of cross section sets:
  - Some contain only a few numbers to parameterize cross section
  - Some represent large databases (data driven models)
- Cross section management
  - **`GetCrossSection`**`()` → sees last set loaded for energy range

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 model inventory

http://geant4.cern.ch/support/proc_mod_catalog/models



Hands-on session (task3)

http://geant4.lngs.infn.it/TRISEP2014/introduction/index.html