Geant4: particles, processes and cuts

Luciano Pandola

INFN

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()`**
  - ...

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...)

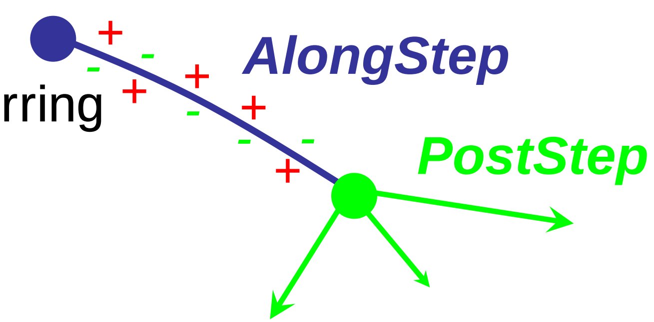
SetCuts()

- 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

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

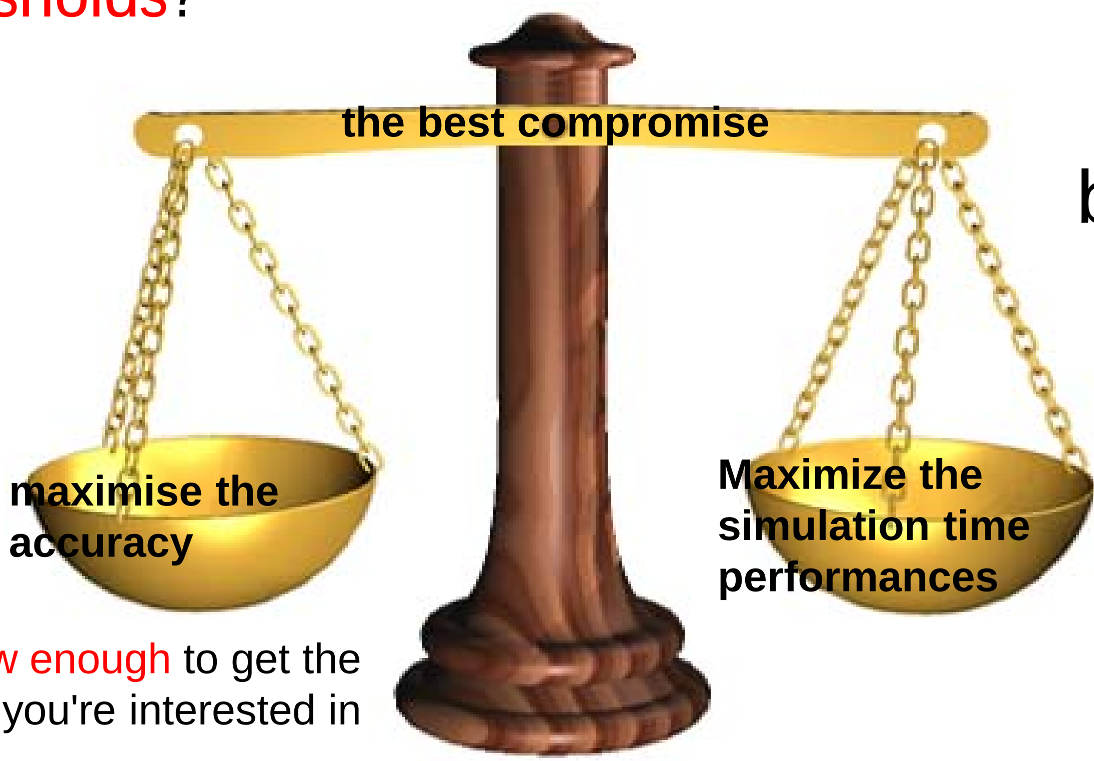


  - **PostStep** actions:
    - For describing point-like (inter)actions, like decay in flight, hadronic interactions ...

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

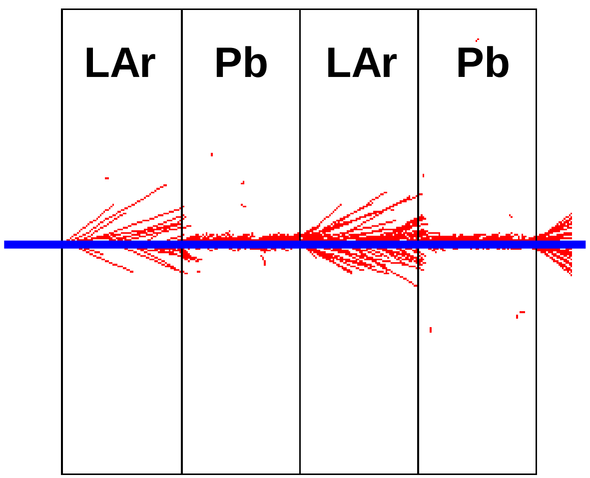
Production thresholds: mixed simulation

- 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

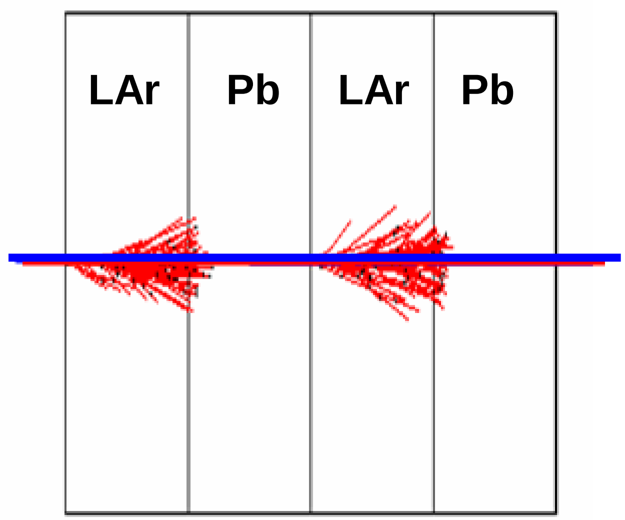
- Geant4 solution: impose a “range” production threshold
  - this threshold is a distance, not an energy
  - default = 1 mm
  - the primary particle loses energy by producing secondary particles which can travel at least the given distance
  - if primary no longer has enough energy to produce secondaries 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

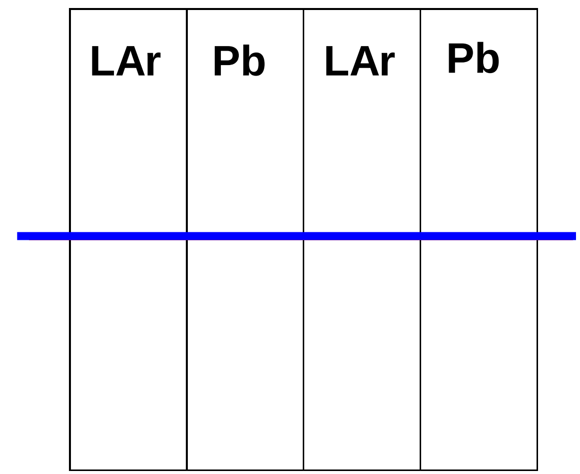
Production threshold: cut in range



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