



GEANT4
A SIMULATION TOOLKIT

Physics II: Overview, Processes, Production Threshold, Regions & Cuts per region

Geant4 PHENIICS & ANF IN2P3 Tutorial,
22 – 26 May 2023,
Orsay

Marc Verderi
LLR, Ecole polytechnique





Credits...

- › Daniel Brandt, Makoto Asai, Dennis Wright (SLAC),
- › Gunter Folger (CERN), etc.
- › Ivana,
- › ...

Introduction

- › In this presentation, we will give an overview of the physics catalog of Geant4
- › We will explain also how physics processes
 - are modeled, in term of C++ class
 - and how they are used during the tracking
- › This is for “cultural” aspects, as in general, you will not have to worry about the structure of the process class
 - Because you will be using existing physics lists
 - Which setup properly processes and take care of technical details
- › We will speak also about “cuts”, which, at the opposite are something you must care about.
 - They must be defined for some physics processes
 - › to run the simulation under time $< \infty$
 - “Cuts” is actually a bad usage name...
 - ... and the issue is « production threshold »
 - But some care has to be taken in defining them



Outline

I. Physics Overview

- Overview of Geant4 physics capability

II. Processes

- How physics processes are modeled in Geant4

III. Production Thresholds (aka cuts)

IV. Regions

V. Cuts per region

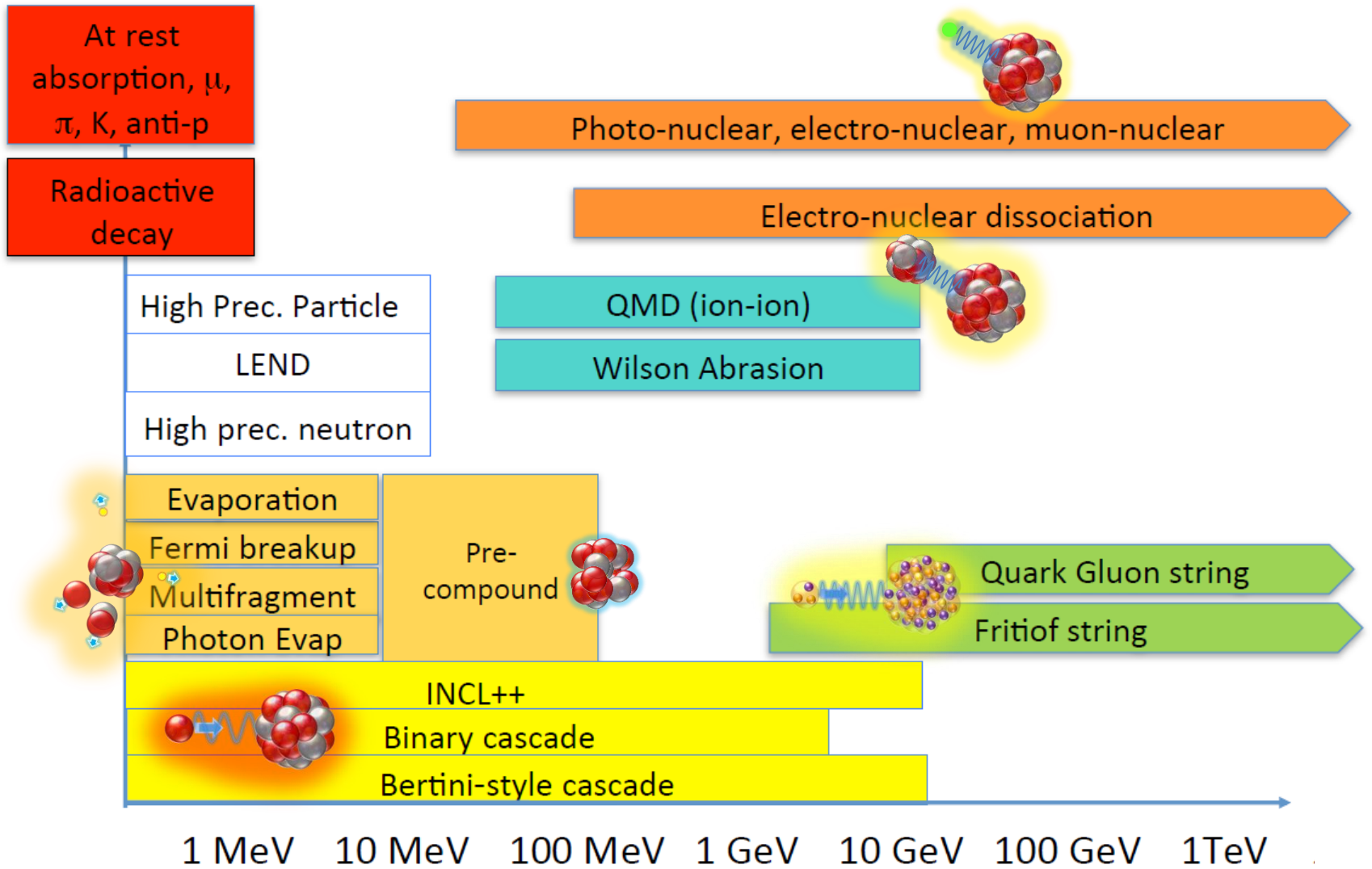


I. Physics Overview

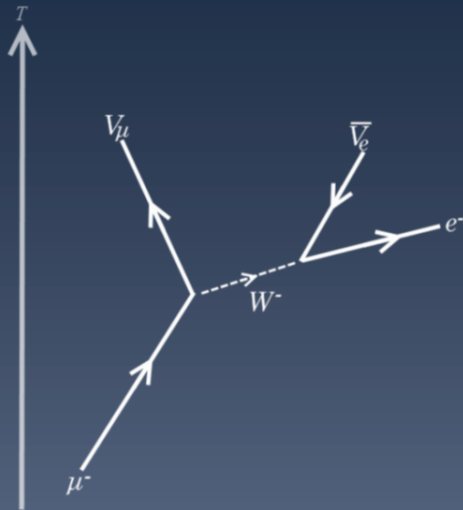
Electromagnetic Processes

- › **Standard** : Complete set of processes covering charged particles and gammas.
 - Energy range 1 keV - ~PeV
- › **Low energy** : More precise description at low energy for e^+ , e^- , γ , charged hadrons incident particle.
 - More atomic shell structure detail
 - Some processes valid down to hundreds of eV
 - Some processes not valid above 1 GeV
- › **DNA & MuElec** : for microdosimetry studies
 - Processes down to a few eV (!)
 - Plus chemistry stage for DNA
- › **Optical photon** : Long wavelength γ (X-ray, UV, visible)
 - Reflection, refraction, absorption, wavelength shifts, Rayleigh scattering
- › **Phonons** : under development. Acoustic phonons, for now. Suited for low-temperature (tens of mK) detectors.

Hadronic Processes



Decay & « Technical »

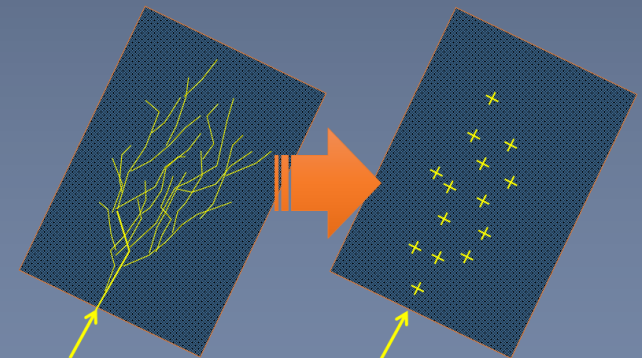


> Decay processes

- Decay of particles of width narrow enough
 - > i.e. : exclude hadronic resonances
- weak decay (leptonic decays, semi-leptonic decays, radioactive decay of nuclei)
- electromagnetic decay (e.g. π^0 , Σ^0)

> « Technical » processes:

- Processes without physics content but which act as interfaces for:
- Parameterization
 - > Fast Simulation fonctionnalité
 - > Hook to shortcut the detailed tracking
- Parallel geometries
 - > Limit the step on parallel geometry boundaries / switch tracking geometries
- Scoring
 - > Collect user requested information
- Biasing
 - > Modify physics behavior wrt to the reference standard one



Fast simulation : a full shower is replaced by a parametrized version of it



II. Processes

How Geant4 models processes

G4VProcess: 3 kind of actions (1/2)

- › Abstract class defining the common interface of **all processes** in Geant4:
 - Used by all « physics » processes
 - but is also used by the transportation, etc...
 - Defined in **source/processes/management**

- › Define **three kinds of actions**:

- **AtRest** actions:

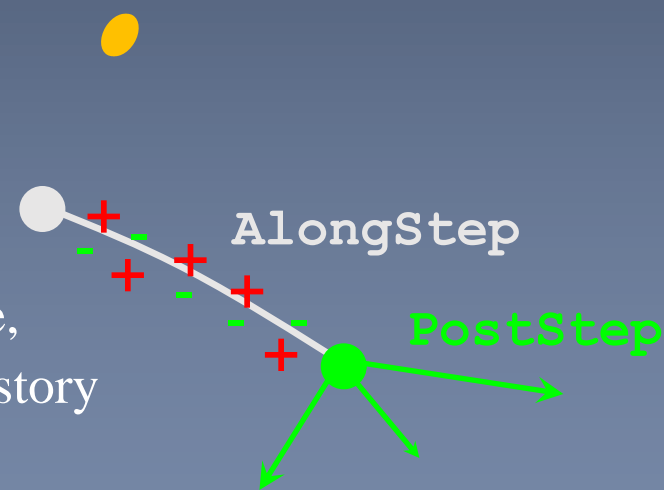
- Decay, e^+ annih., absorption ...

- **AlongStep** actions:

- To describe continuous (inter)actions, occurring along the path of the particle, like ionisation; used for condensed history

- **PostStep** actions:

- For describing point-like (inter)actions, like decay in flight, hard radiation...



G4VProcess: 3 kind of actions (2/2)

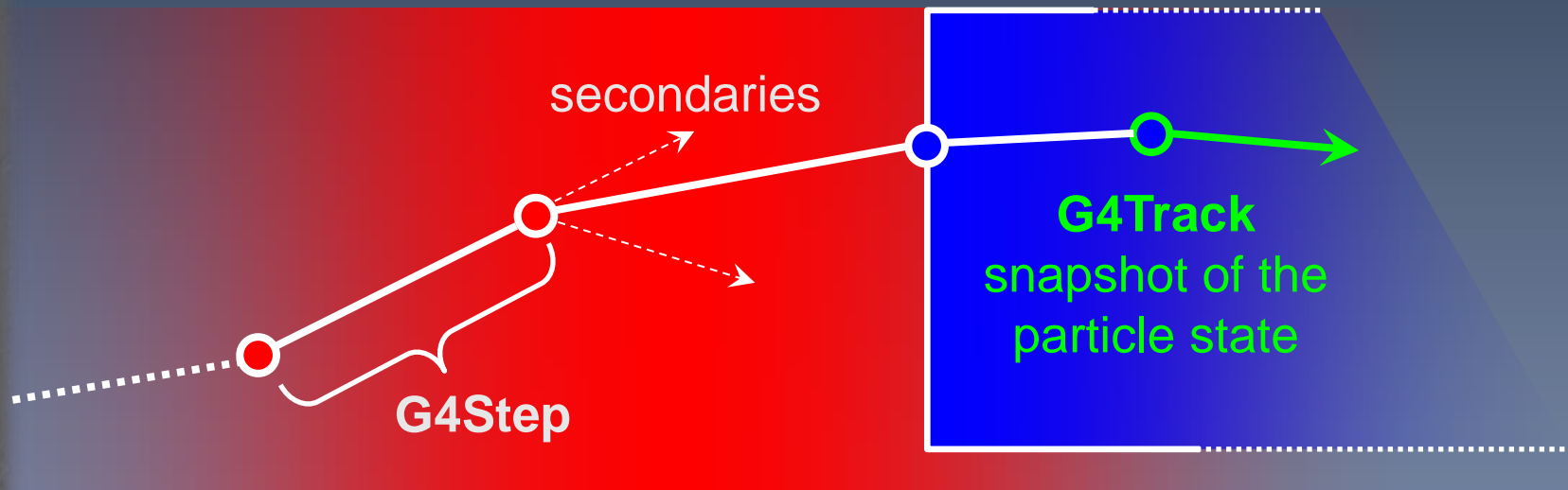
- › A process can implement **any combination** of the three **AtRest**, **AlongStep** and **PostStep** actions:
 - eg: decay = **AtRest** + **PostStep**

- › Each action defines **two methods**:
 - **GetPhysicalInteractionLength ()**:
 - › Used to **limit the step**:
 - either because the process « triggers » an interaction, a decay
 - or any other reasons, like fraction of energy loss, geometry boundary, user's limit ...
 - **DoIt ()**:
 - › Implements the **actual action** to be applied on the track;
 - › And the related production of secondaries.

How Geant4 uses the processes during tracking ?

› Remember:

– Tracks are moved step by step :

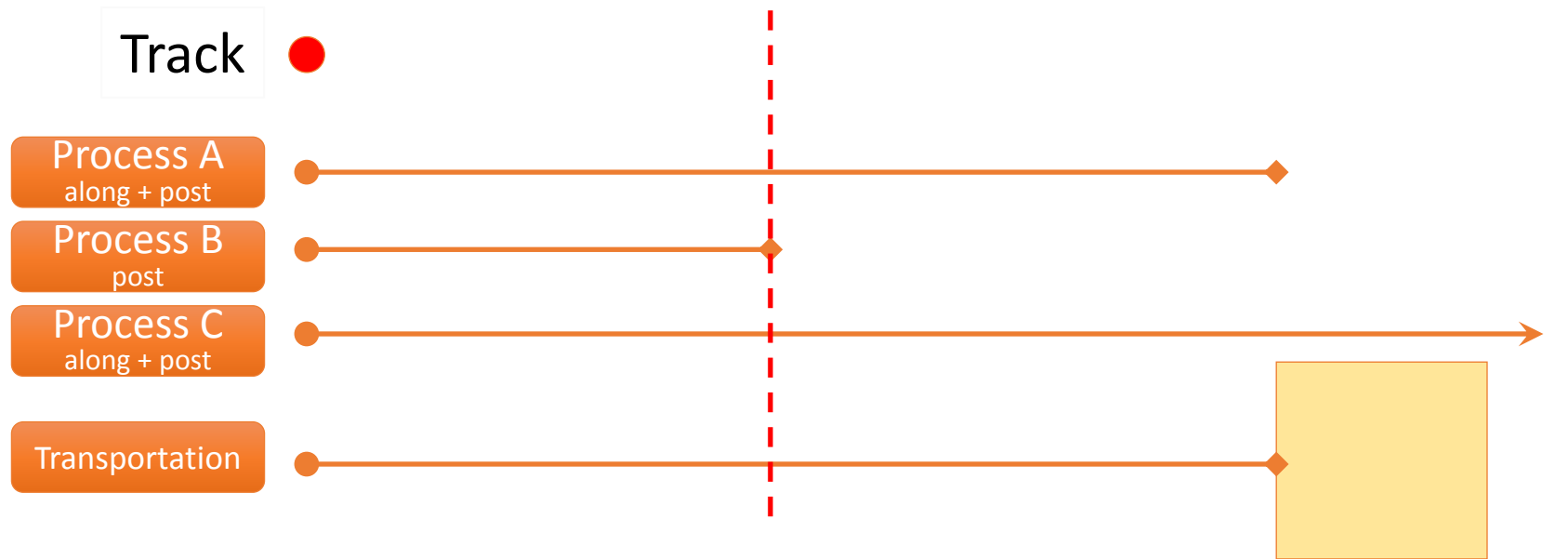


– And several processes are attached to the track

› So what happens during a step ?

Process Handling by the Stepping

1. At the beginning of the step, the step length is determined:
 - Consider all processes attached to the current `G4Track`;
 - Define the step length as the smallest of the lengths among:
 - > All `AlongStepGetPhysicalInteractionLength()`
 - > All `PostStepGetPhysicalInteractionLength()`

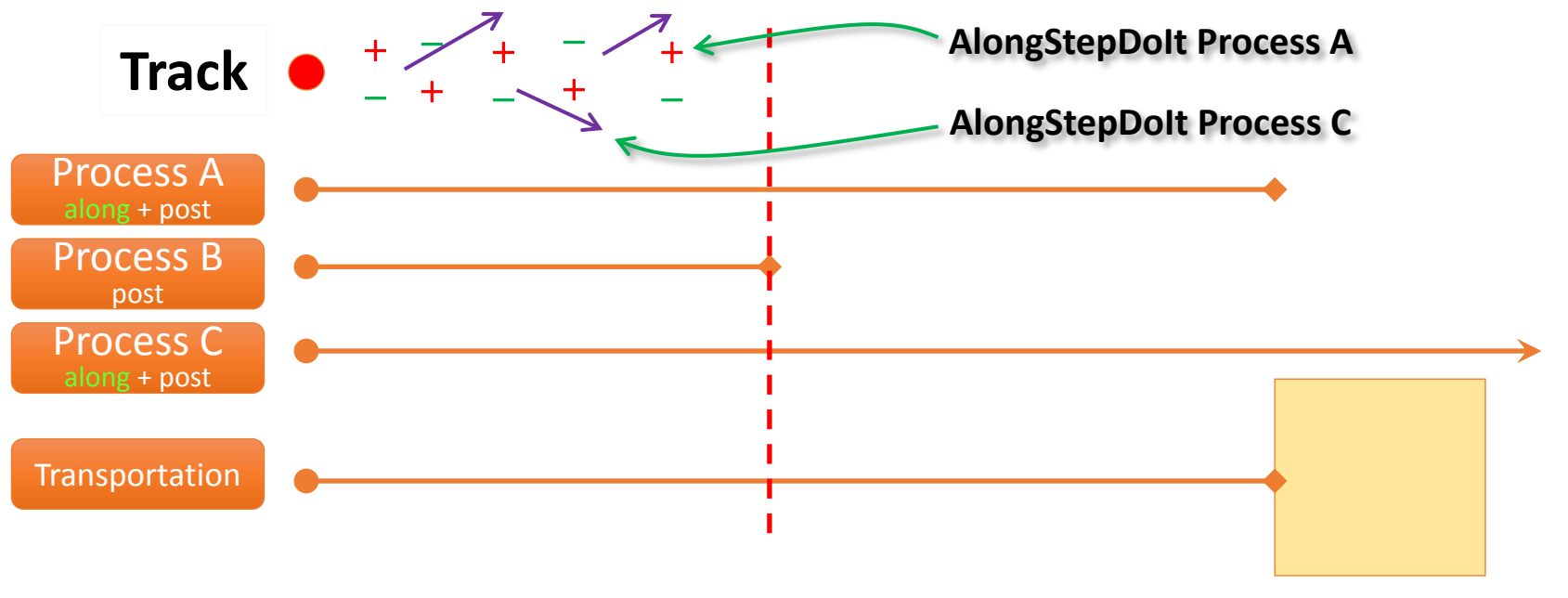


Process Handling by the Stepping

1. At the beginning of the step, the step length is determined:
 - Consider all processes attached to the current `G4Track`;
 - Define the step length as the smallest of the lengths among:
 - > All `AlongStepGetPhysicalInteractionLength()`
 - > All `PostStepGetPhysicalInteractionLength()`
2. Apply all `AlongStepDoIt()` actions, « at once »:
 - Changes computed from particle state at the beginning of the step;
 - Accumulated in the `G4Step`;
 - Then applied to the `G4Track`, from the `G4Step`.

Process Handling by the Stepping

1. At the beginning of the step, the step length is determined:
 - Consider all processes attached to the current `G4Track`;
 - Define the step length as the smallest of the lengths among:
 - > All `AlongStepGetPhysicalInteractionLength()`
 - > All `PostStepGetPhysicalInteractionLength()`

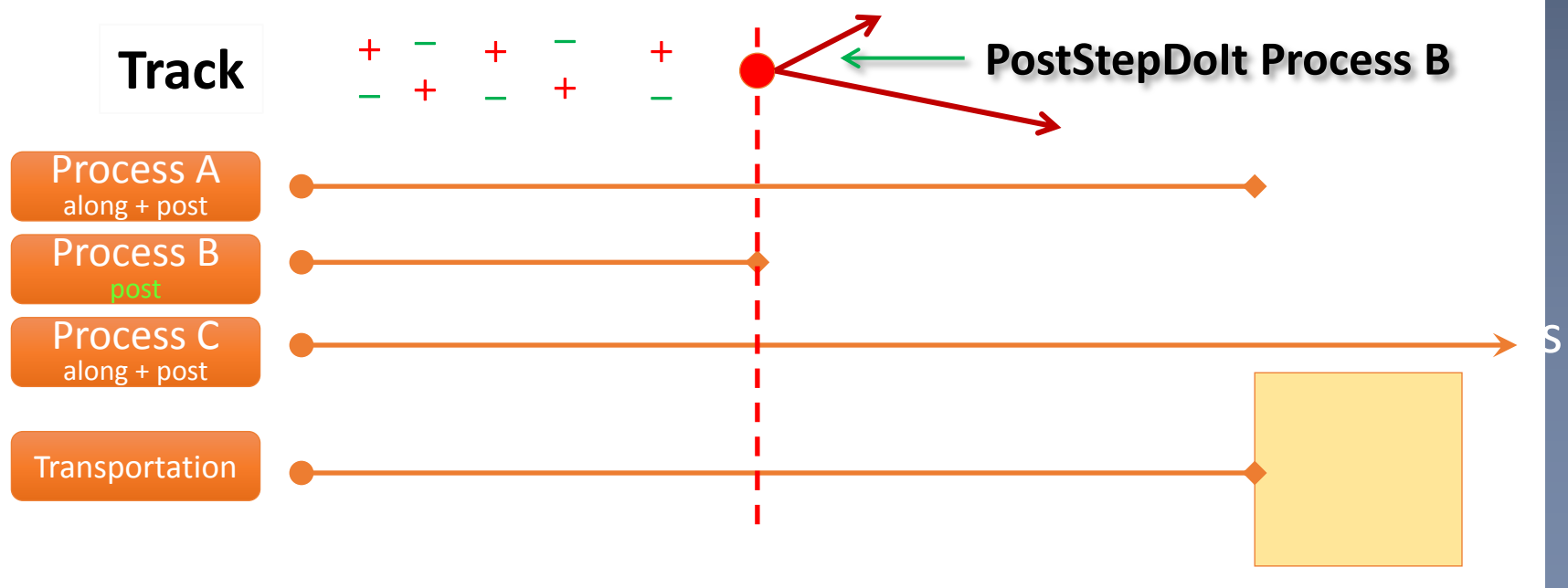


Process Handling by the Stepping

1. At the beginning of the step, the step length is determined:
 - Consider all processes attached to the current `G4Track`;
 - Define the step length as the smallest of the lengths among:
 - › All `AlongStepGetPhysicalInteractionLength()`
 - › All `PostStepGetPhysicalInteractionLength()`
2. Apply all `AlongStepDoIt()` actions, « at once »:
 - Changes computed from particle state at the beginning of the step;
 - Accumulated in the `G4Step`;
 - Then applied to the `G4Track`, from the `G4Step`.
3. Apply `PostStepDoIt()` action(s) « sequentially », as long as the particle is alive:
 - Apply `PostStepDoIt()` of process which limited the step (if any);
 - And apply any other « forced » processes (not discussed here)

Process Handling by the Stepping

1. At the beginning of the step, the step length is determined:
 - Consider all processes attached to the current `G4Track`;
 - Define the step length as the smallest of the lengths among:
 - > All `AlongStepGetPhysicalInteractionLength()`
 - > All `PostStepGetPhysicalInteractionLength()`



Process Handling by the Stepping

1. At the beginning of the step, the step length is determined:
 - Consider all processes attached to the current `G4Track`;
 - Define the step length as the smallest of the lengths among:
 - › All `AlongStepGetPhysicalInteractionLength()`
 - › All `PostStepGetPhysicalInteractionLength()`
2. Apply all `AlongStepDoIt()` actions, « at once »:
 - Changes computed from particle state at the beginning of the step;
 - Accumulated in the `G4Step`;
 - Then applied to the `G4Track`, from the `G4Step`.
3. Apply `PostStepDoIt()` action(s) « sequentially », as long as the particle is alive:
 - Apply `PostStepDoIt()` of process which limited the step (if any);
 - And apply any other « forced » processes (not discussed here)



III. Production Thresholds (aka « cuts »)

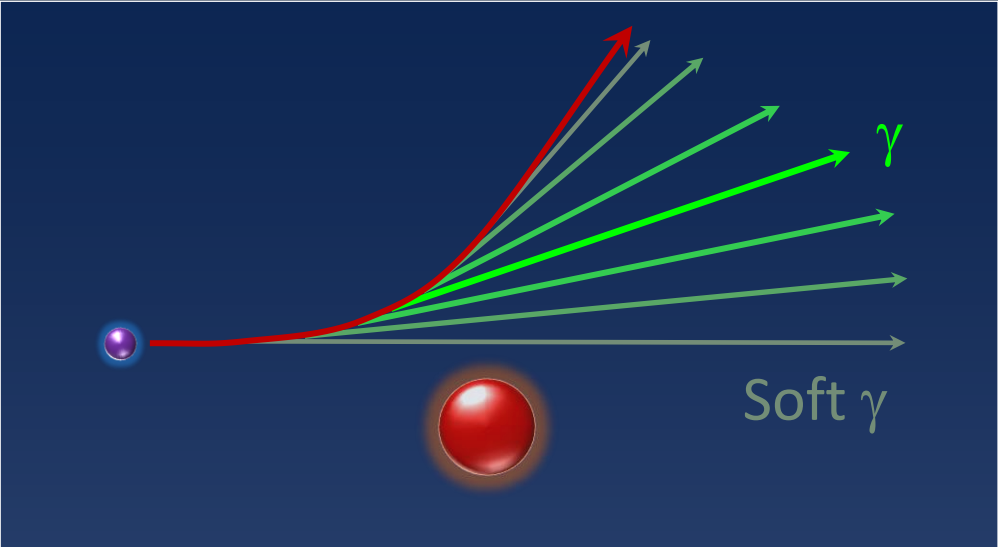
Threshold for Secondary Production

- › Simulation accuracy limited by:
 - Intrinsic limitation of physics process modeling
 - › For example details of atomic or molecular structure, etc.
 - Finite computing power
 - › Which forces to limit the production to some # of events
 - › Which restricts the usage to models “fast enough”
 - I.e. : no way to use lattice QCD in detector simulation
 - › **Which forces to suppress the production of very low energy particles for processes having infrared divergences**
 - Infinite or very large number of produced secondaries

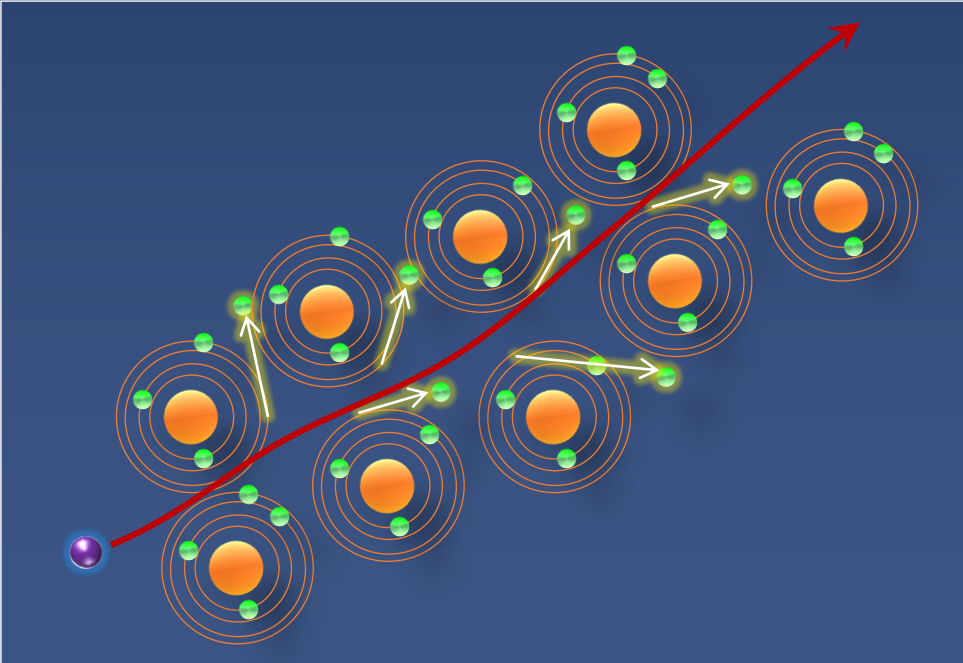
›

›

›



Bremstrahlung: actual divergence in forward production of ultra-soft gammas.



Ionisation : large production of ionisation electrons, from loosely bound atomic ones.



Threshold for Secondary Production

- › Simulation accuracy limited by:
 - Intrinsic limitation of physics process modeling
 - › For example details of atomic or molecular structure, etc.
 - Finite computing power
 - › Which forces to limit the production to some # of events
 - › Which restricts the usage to models “fast enough”
 - I.e. : no way to use lattice QCD in detector simulation
 - › **Which forces to suppress the production of very low energy particles for processes having infrared divergences**
 - Infinite or very large number of produced secondaries
- › Every simulation developer must answer the question:
 - How low can I go?
- › This is a balancing act:
 - Need to go low enough to get the physics you’re interested in
 - Can’t go too low because of CPU time consumption
- ›

Threshold for Secondary Production

› Geant4 solution: impose a production threshold

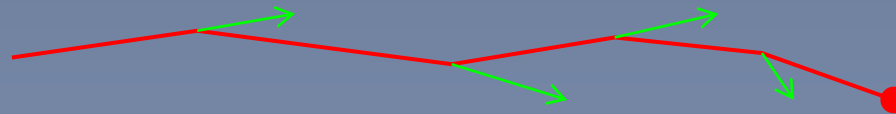
- this threshold is a distance, not an energy
 - › This a “range threshold”
- default = 0.7 mm
- What makes this “range threshold” ?
 - › In Nature, the primary particle loses energy by producing secondary electrons or gammas
 - (or by exciting atoms, molecules, etc. : not of interest here)
 - › In Geant4, the threshold makes only secondary particles able to travel > 0.7 mm to be created
 - The rest is accounted for a “continuous energy loss” (condensed history)

› Only one value (per region) of range threshold is needed for all materials

- And this distance is internally converted into the related energy thresholds by Geant4
- Conversion “range \rightarrow energy” is made to according to material

› Near the primary particle end-point:

- When the primary becomes of too low energy to produce secondaries above threshold:
 - › discrete energy loss ceases (no more secondaries produced)
 - › the primary is tracked down to zero kinetic energy using continuous energy loss



- Note that this makes Geant4 having “no tracking cuts”

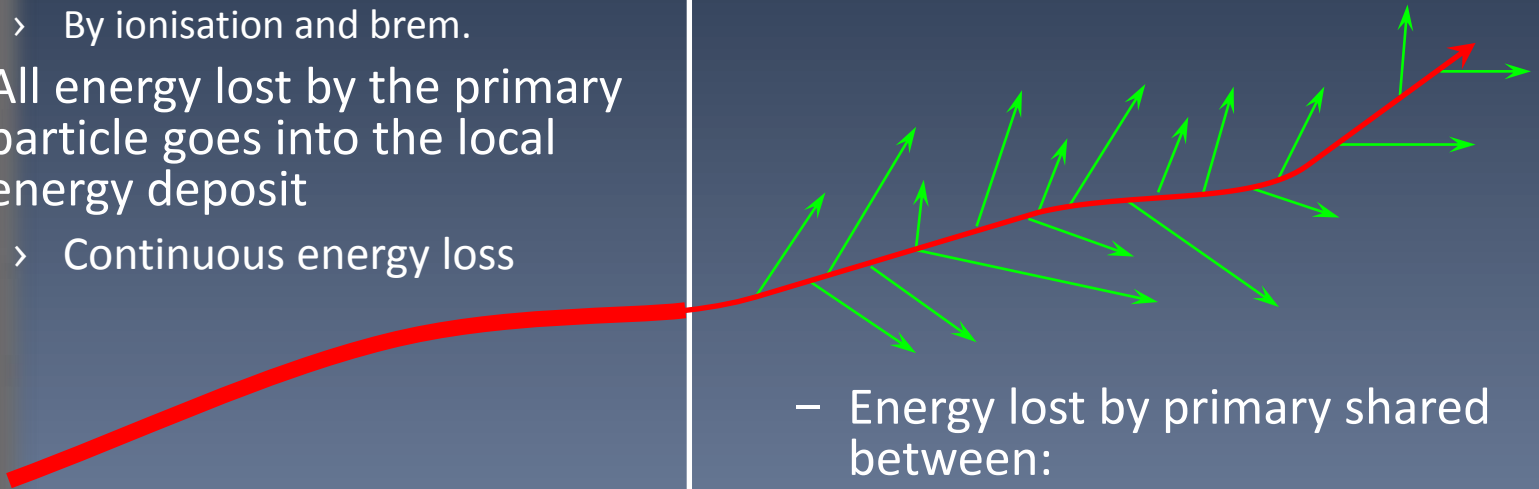
High and low thresholds : what differences ?

> High Threshold

- No secondary production
 - > By ionisation and brem.
- All energy lost by the primary particle goes into the local energy deposit
 - > Continuous energy loss
- You'll see as:
 - > step-> `GetTotalEnergyDeposit()` is high
 - > You don't have energy deposit elsewhere than on primary path

> Low Threshold

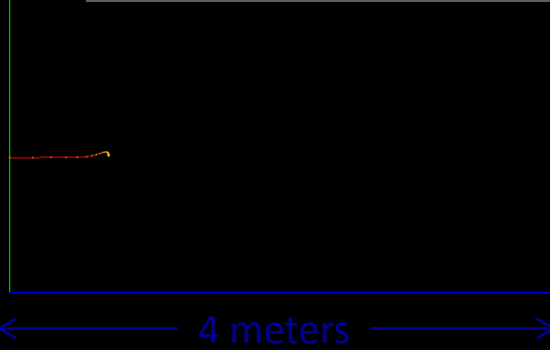
- Many secondaries produced
- Energy lost by primary shared between:
 - > Local energy deposit
 - > discrete secondary production
- You'll see as:
 - > step-> `GetTotalEnergyDeposit()` is lower than before
 - > Energy deposit more scattered due to subsequent deposit of secondary particles



10 GeV e^- in liquid Argon, with cuts of 1 km, 1 m, 1 mm and 1 μm

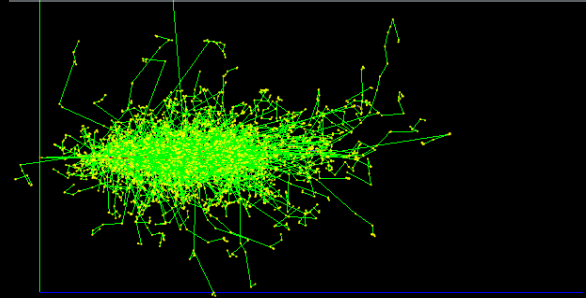
1 km

#track=1; #step=22; time~0s



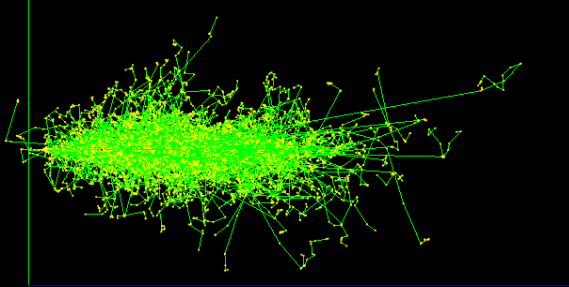
1 m

#track=8k; #step=20k; time=60ms



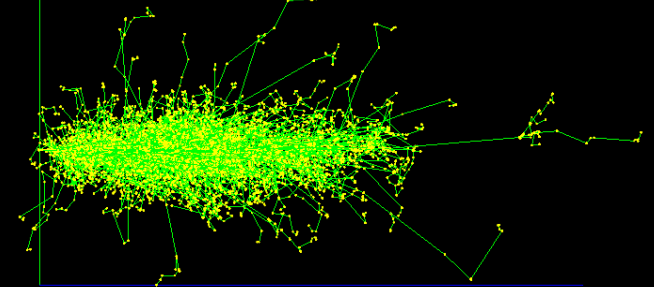
1 mm

#track=18k; #step=39k; time=90ms



1 μm

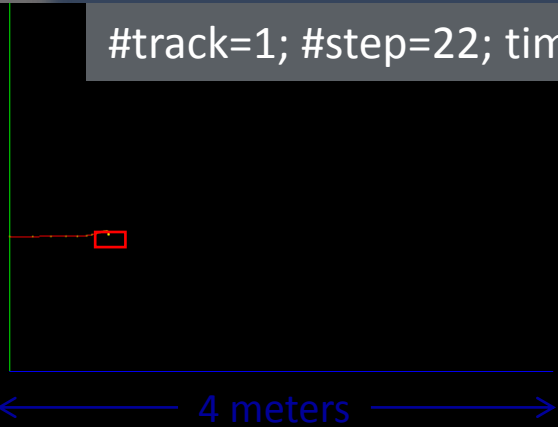
#track=724k; #step=1.5M; time=4.6s



10 GeV e^- in liquid Argon, with cuts of 1 km, 1 m, 1 mm and 1 μm

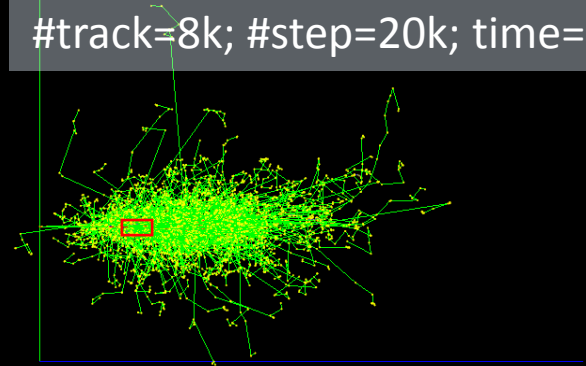
1 km

#track=1; #step=22; time~0s



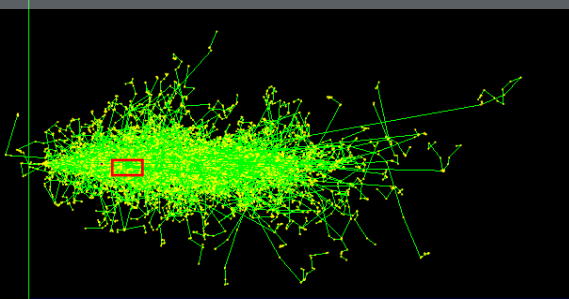
1 m

#track=8k; #step=20k; time=60ms



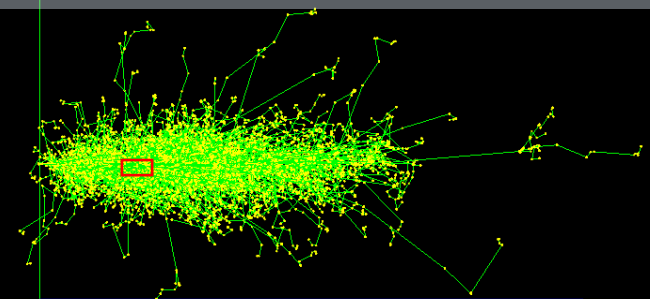
1 mm

#track=18k; #step=39k; time=90ms



1 μm

#track=724k; #step=1.5M; time=4.6s



10 GeV e^- in liquid Argon, with cuts of 1 km, 1 m, 1 mm and 1 μm

1 km

#track=1; #step=22; time~0s

1 m

#track=8k; #step=20k; time=60ms

1 mm

#track=18k; #step=39k; time=90ms

1 μm

#track=724k; #step=1.5M; time=4.6s



Assigning cuts to your simulation

- › You must assign cuts to γ , e^- and e^+ .
 - For γ 's : needed to limit production from infrared divergence of brem. process
 - For e^- 's : needed to limit high production from ionization
 - For e^+ 's : bit of historical reasons (no infrared divergence process)
 - › (Plans in Geant4 to review this)
- › You may assign cuts to protons
 - To define the threshold for producing proton by **recoil in elastic collisions**
 - Threshold used for recoil ions too.
- › The easiest way to define cuts is at run time
 - On command line or with a macro
 - For γ , e^- and e^+ and p in one go, eg:

```
/run/setCut 2 mm
```

- Per particle threshold, eg:

```
/run/setCutForAGivenParticle e- 0.1 mm
```

- (later we'll add the case of "region")

Getting information on range to energy conversion

```
/run/setCut 1 mm  
/run/beamOn 1 (to force calculations of thresholds)  
/run/dumpCouples
```

```
===== Table of registered couples =====
```

```
Index : 0      used in the geometry : Yes
```

```
Material : G4_Galactic
```

```
Range cuts      : gamma 1 mm      e- 1 mm      e+ 1 mm      proton 1 mm
```

```
Energy thresholds : gamma 990 eV      e- 990 eV      e+ 990 eV      proton 100 keV
```

```
Region(s) which use this couple :
```

```
DefaultRegionForTheWorld
```

```
Index : 1      used in the geometry : Yes
```

```
Material : G4_Pb
```

```
Range cuts      : gamma 1 mm      e- 1 mm      e+ 1 mm      proton 1 mm
```

```
Energy thresholds : gamma 101.843 keV      e- 1.36749 MeV      e+ 1.27862 MeV      proton 100 keV
```

```
Region(s) which use this couple :
```

```
DefaultRegionForTheWorld
```

```
Index : 2      used in the geometry : Yes
```

```
Material : G4_PLASTIC_SC_VINYLTOLUENE
```

```
Range cuts      : gamma 1 mm      e- 1 mm      e+ 1 mm      proton 1 mm
```

```
Energy thresholds : gamma 2.40367 keV      e- 356.639 keV      e+ 344.855 keV      proton 100 keV
```

```
Region(s) which use this couple :
```

```
DefaultRegionForTheWorld
```

```
=====
```

Getting information on range to energy conversion

```
/run/setCut 0.01 mm  
/run/beamOn 1 (to force calculations of thresholds)  
/run/dumpCouples
```

```
===== Table of registered couples =====
```

```
Index : 0      used in the geometry : Yes
```

```
Material : G4_Galactic
```

```
Range cuts      : gamma 10 um      e- 10 um      e+ 10 um      proton 10 um  
Energy thresholds : gamma 990 ev    e- 990 ev     e+ 990 ev     proton 1 keV
```

```
Region(s) which use this couple :
```

```
DefaultRegionForTheWorld
```

```
Index : 1      used in the geometry : Yes
```

```
Material : G4_Pb
```

```
Range cuts      : gamma 10 um      e- 10 um      e+ 10 um      proton 10 um  
Energy thresholds : gamma 5.995 keV e- 58.1082 keV e+ 56.9484 keV proton 1 keV
```

```
Region(s) which use this couple :
```

```
DefaultRegionForTheWorld
```

```
Index : 2      used in the geometry : Yes
```

```
Material : G4_PLASTIC_SC_VINYLTOLUENE
```

```
Range cuts      : gamma 10 um      e- 10 um      e+ 10 um      proton 10 um  
Energy thresholds : gamma 990 ev    e- 15.1173 keV e+ 14.6763 keV proton 1 keV
```

```
Region(s) which use this couple :
```

```
DefaultRegionForTheWorld
```

```
=====
```

Getting information on range to energy conversion

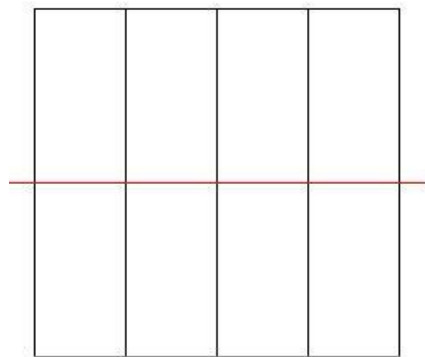
```
/run/setCut 1 nm  
/run/beamOn 1 (to force calculations of thresholds)  
/run/dumpCouples
```

```
===== Table of registered couples =====  
Index : 0      used in the geometry : Yes  
Material : G4_Galactic  
Range cuts      : gamma 10 Ang    e- 10 Ang    e+ 10 Ang proton 10 Ang  
Energy thresholds : gamma 990 eV    e- 990 eV    e+ 990 eV    proton 0.1 eV  
Region(s) which use this couple :  
DefaultRegionForTheWorld  
Index : 1      used in the geometry : Yes  
Material : G4_Pb  
Range cuts      : gamma 10 Ang    e- 10 Ang    e+ 10 Ang proton 10 Ang  
Energy thresholds : gamma 990 eV    e- 990 eV    e+ 990 eV    proton 0.1 eV  
Region(s) which use this couple :  
DefaultRegionForTheWorld  
Index : 2      used in the geometry : Yes  
Material : G4_PLASTIC_SC_VINYLTOLUENE  
Range cuts      : gamma 10 Ang    e- 10 Ang    e+ 10 Ang proton 10 Ang  
Energy thresholds : gamma 990 eV    e- 990 eV    e+ 990 eV    proton 0.1 eV  
Region(s) which use this couple :  
DefaultRegionForTheWorld  
=====
```

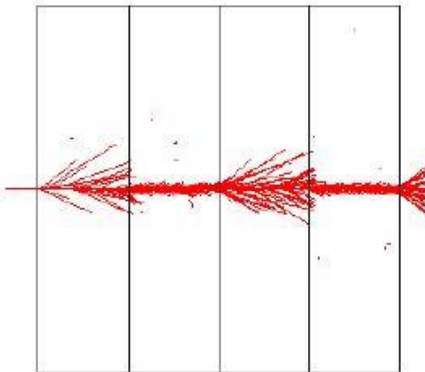
Production Threshold vs. Energy Cut

Example: 500 MeV p in LAr-Pb Sampling Calorimeter

Energy Threshold

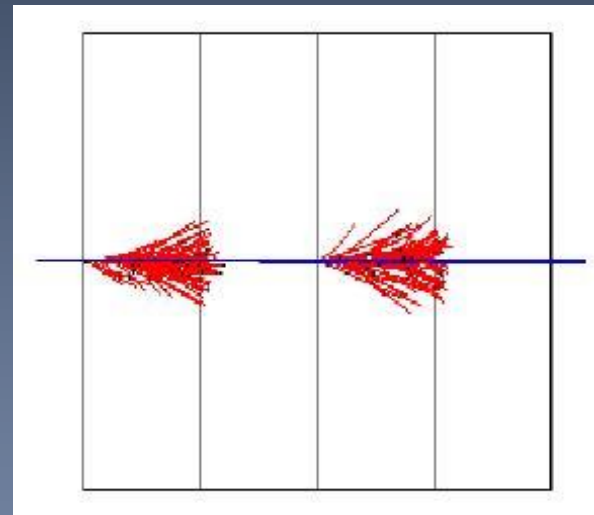


Cut = 2 MeV

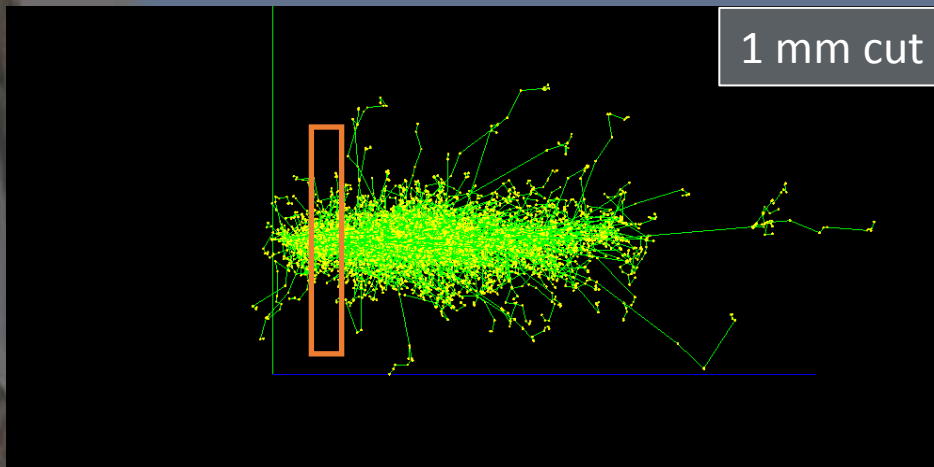
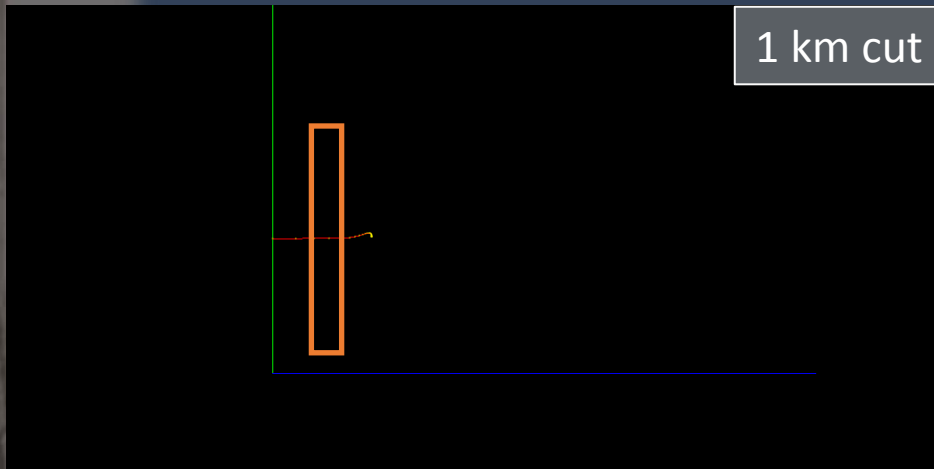


Cut = 450 keV

Geant4 Production
Range Threshold

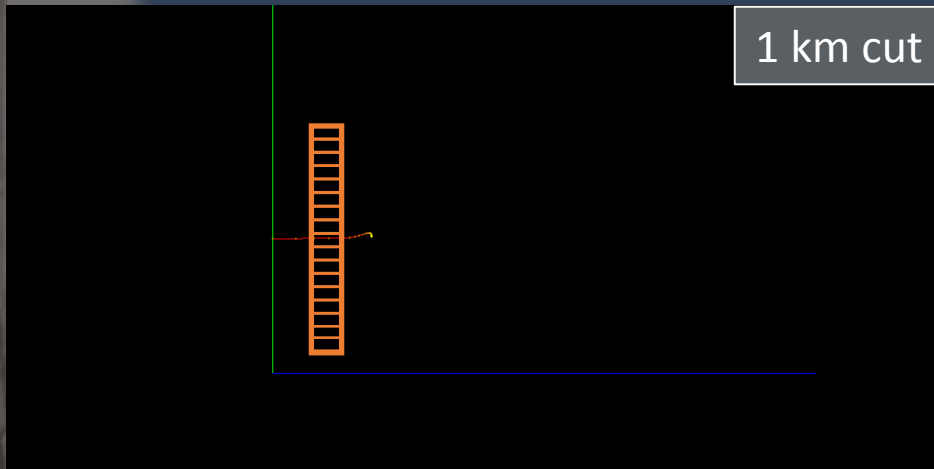


Energy recorded

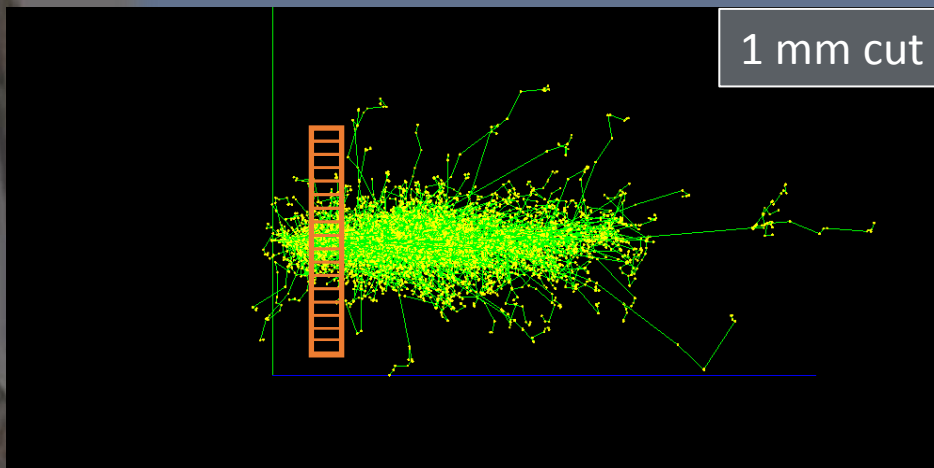


- > If recording energy deposit in a big volume
 - No difference between high and low energy thresholds

Energy recorded

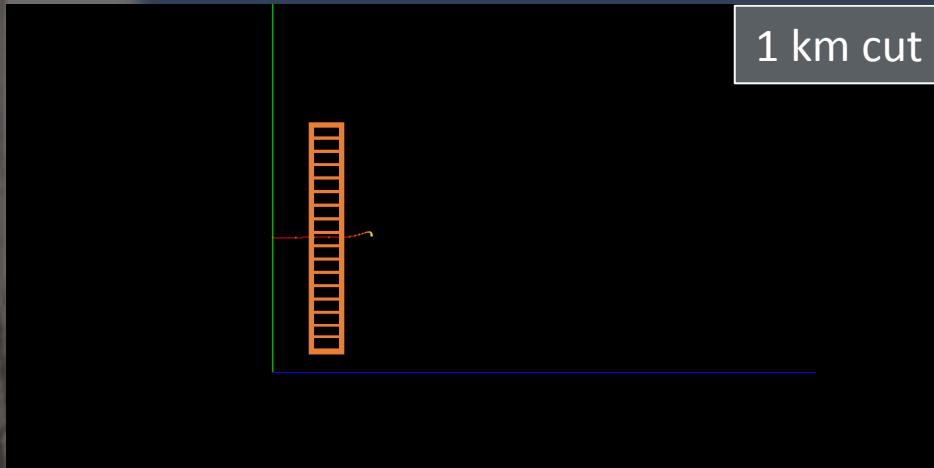


- > If recording energy deposit in a big volume
 - No difference between high and low energy thresholds

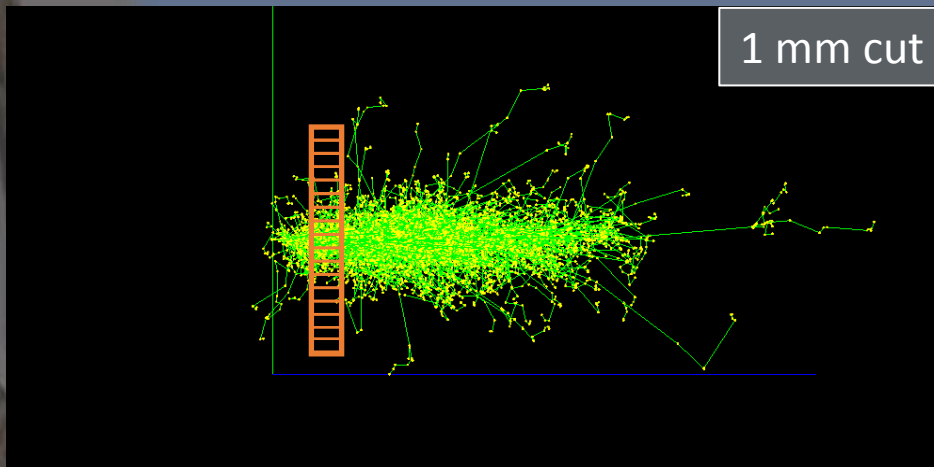


- > But if recording in small volumes
 - Big differences !

Energy recorded



- > If recording energy deposit in a big volume
 - No difference between high and low energy thresholds

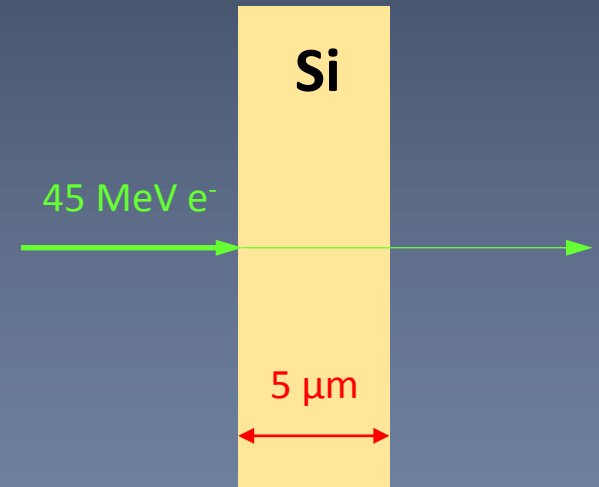
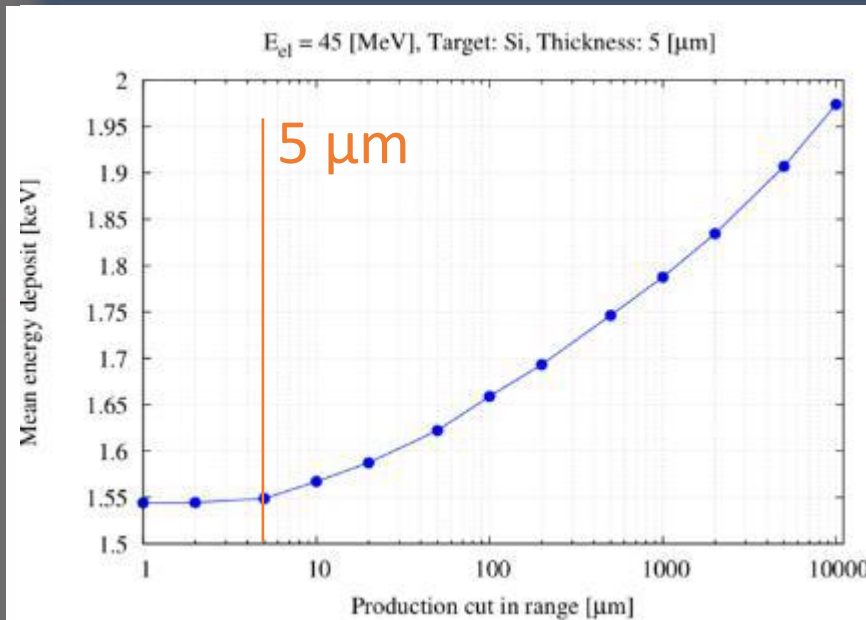


- > But if recording in small volumes
 - Big differences !

> Typically : range cut
~ volume dimension

Energy recorded : concrete case

- › 45 MeV electrons crossing a 5 μm thick Si volume
 - Mean energy deposit as function of range cut



- › Flat part if cut < 5 μm , but increasing if cut > 5 μm
 - Why ?



IV. Regions

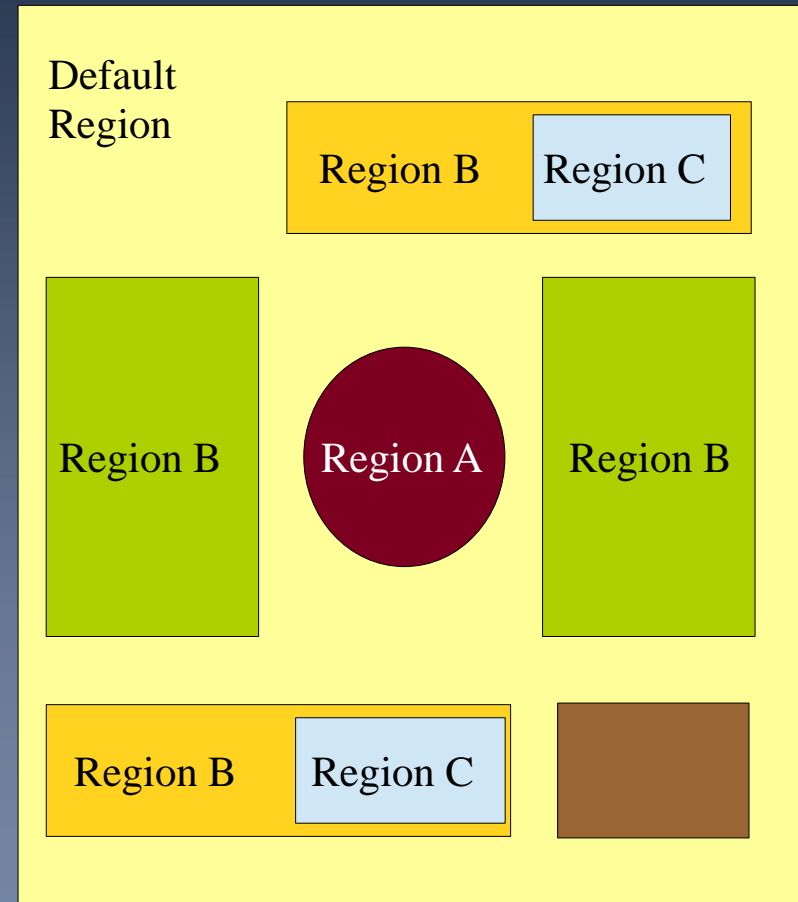
A quick geometry detour

Concept of Regions (1/2)

- On the top of the volume hierarchy users can define regions which are selected sets of volumes, typically of sub-systems
 - E.g. barrel + end-caps of the calorimeter, “deep” areas of support structures can be a region.
- A region can be any group of volumes
- A region can hold a set of various properties:
 - Production thresholds (cuts)
 - User limits
 - User region information
 - Fast simulation manager
 - Regional user stepping action
 - Field manager

Concept of Regions (2/2)

- A region is always associated with one or more logical volumes
- A **root logical volume** = volume associated to a region
 - All daughter volumes share the same region, unless a daughter volume itself becomes an other root.
 - A logical volume can not be shared among regions.
- World logical volume is always associated with the default region
 - Users do not need to define it.



Creating a region, accessing it, creating a user region information object

MyDetectorConstruction.cc

```
#include "G4Region.hh"

// Create a region
G4Region* myRegion = new G4Region("MyRegion");
// Attach a logical volume to the region
myRegion->AddRootLogicalVolume(myLV);
```

MyOtherClass.cc

```
#include "G4RegionStore.hh"
#include "MyRegionInformation.hh"

// Retrieve the region by its name
G4Region* region
    = G4RegionStore::GetInstance()->GetRegion("MyRegion");

// Create some property to be assigned to a region
MyRegionInformation* myInfo = new MyRegionInformation();

// Set myInfo to the region
region->SetUserInformation(myInfo);
```

› We will see just after how to assign « cuts » to a region.



V. Cuts per region

Why cuts per region ?

- > Running with “as low as possible” cuts is:

- Good for physics quality
- Bad for CPU consumption

- > In large applications (ie : HEP) not all parts of detector simulation require the same level of accuracy:

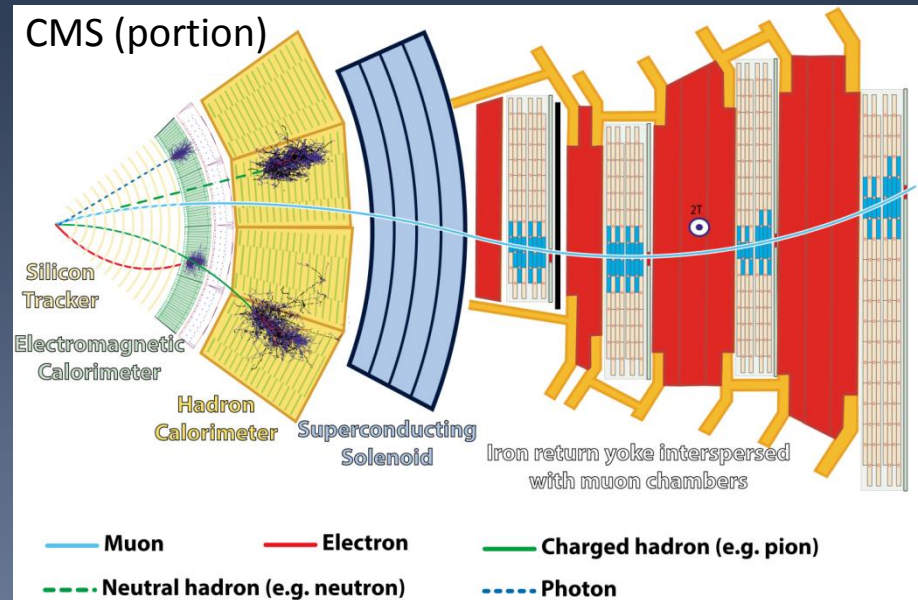
- Tracking systems:
 - > Good accuracy needed
 - > So, low cuts needed
- Hadron calorimeter:
 - > Low accuracy is enough
 - > So high cuts ok

- > Issue:

- Low cuts : Accuracy in tracking systems ✓ Processing time for hadron calorimeter ✗
- High cuts : Accuracy in tracking systems ✗ Processing time for hadron calorimeter ✓
- Medium cuts: Make everybody unhappy ;)

- > **Solution:**

- Allow “cuts per region”
- Tracking system = a region with low cuts
- Hadron calorimeter = a region with high cuts



Assigning cuts to a region

- › Assume you define a region with name “MyRegion” in your detector construction
- › To assign cuts to it, you do:

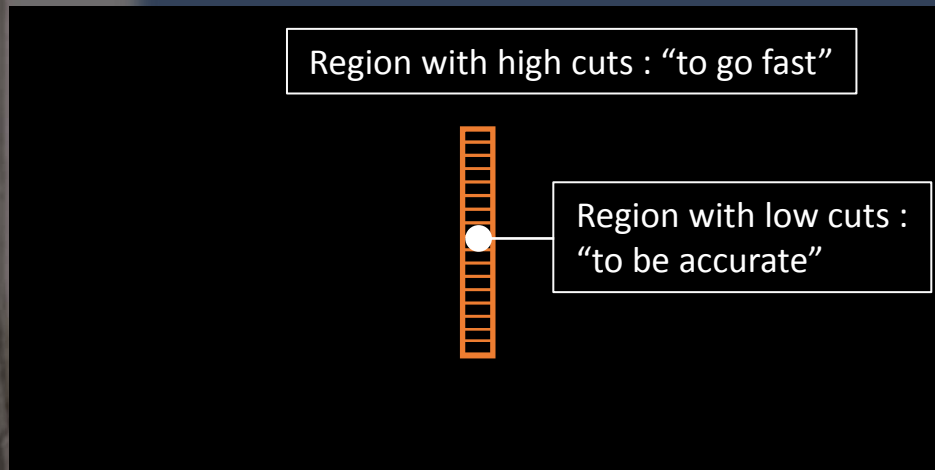
```
...  
// Create the region  
G4Region* myRegion = new G4Region("MyRegion");  
...  
...  
// Define cuts object for the new region and set values  
G4ProductionCuts* cuts = new G4ProductionCuts;  
cuts->SetProductionCut(0.01*mm); // for gamma, e+, e-, p  
// Assign cuts to region  
myRegion->SetProductionCuts(cuts);  
...
```

- › And you can change cut values with command line (or macro) as:

```
/run/setCutForRegion MyRegion 1 mm
```

- › Note that the world volume is in fact a region : it is the “default” one.

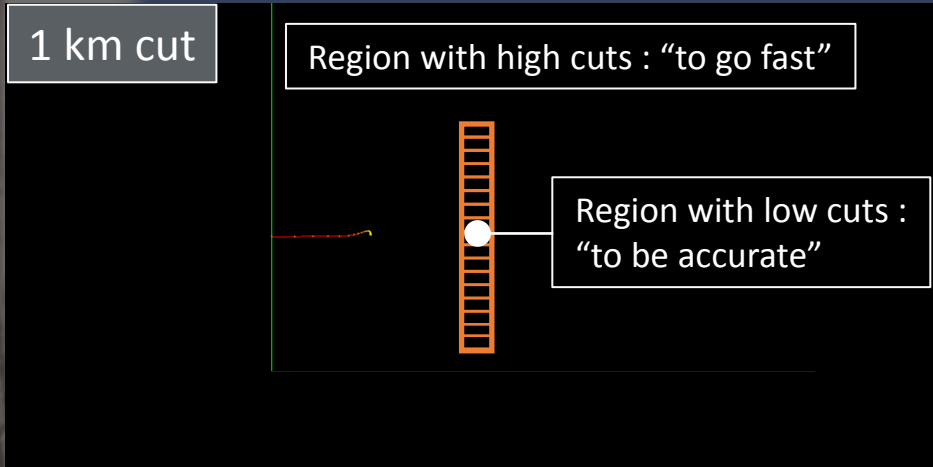
Be critical : temptation for mistake



> Temptation:

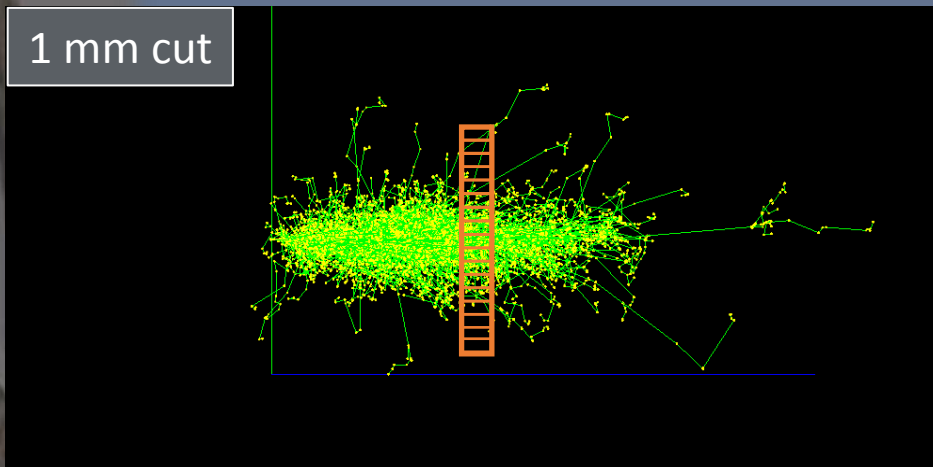
- *“Let me gain time putting a high energy threshold in the parts where I am not interested in details and putting low thresholds where I am interested in details !”*

Be critical : temptation for mistake



> Temptation:

- “Let me gain time putting a high energy threshold in the parts where I am not interested in details and putting low thresholds where I am interested in details !”



> Bad idea !

- What happens in one volume is not only determined by this volume, but also by what happens ***before*** this volume.
- Our example with two extreme threshold cases makes it clear...



Threshold for Secondary Production

- › Instead of “secondary production threshold distance” it is more convenient to simply say “cuts”
- › The cuts values are set in the SetCuts() method of your physics list
 - Either for the entire simulation (ie for the entire world volume)
 - Or per region
 - Geant4 proposes the default value of 0.7 mm
- › They can be defined with command line, eg:
 - Idle> /run/setCuts 1 mm
- › User needs to decide the best value:
 - The lower the better
 - › To be balanced with your available computing power
 - Typically range cut \sim volume dimension is fine
 - › Being careful of not having cut too severely before this volume

Summary

- › Geant4 supplies many physics processes which cover electromagnetic, hadronic, decay physics and “technical”.
- › A unique interface, G4VProcess, allows processes to specify their nature: AtRest, Along (continuous), PostStep (discrete)
 - A process may mix several of these
- › Geant4 does not have “tracking cut”
 - Produced particles are tracked down to zero energy.
- › Geant4 makes use of a “range cut” for controlling the production of secondary particles
 - For some particles and some processes only
- › It is recommended to use a range cut \sim smallest dimension you’re interested in.