Version 10.5

Physics-II: Particles and Processes in Geant4

Vladimir Ivantchenko

CERN, Geneva, Switzerland & Tomsk State University, Russia

Geant4 Course

CERN, 26-27 March 2019

Outline

- Geant4 interface to physics
  - G4VUserPhysicsList
- Geant4 particles
- Geant4 processes
- Ions and exotic particles
- Geant4 Cuts
- Special EM topics
- How to use particles, processes, cuts

Geant4 interface to physics

- The interface of Geant4 kernel to physics is abstract
- Base physics abstract classes are following:
  - The `G4ParticleDefinition` objects shared between threads
  - The `G4VProcess` thread local objects
  - The `G4ProcessManager` thread local interface class
- These interfaces are stable for ~20 years allowing users to work with different Geant4 versions and providing a basis for new developments
  - Internal modification happens when multi-treaded mode was introduced
- Configuration of physics is prepared in the G4VUserPhysicsList mandatory user class
- A user may be also a developer of a custom particle or process

- Physics List is an object that is responsible to:
  - specify all the particles that will be used in the simulation application
  - together with the list of physics processes assigned to each individual particles
- One out of the 3 mandatory objects that the user needs to provide to the G4RunManager in case of all Geant4 applications:
  - it provides the information when, how and what set of physics needs to be invoked
- Provides a very flexible way to set up the physics environment:
  - the user can chose and specify the particles that they want to be used
  - the user can chose the physics (processes) to assign to each particle

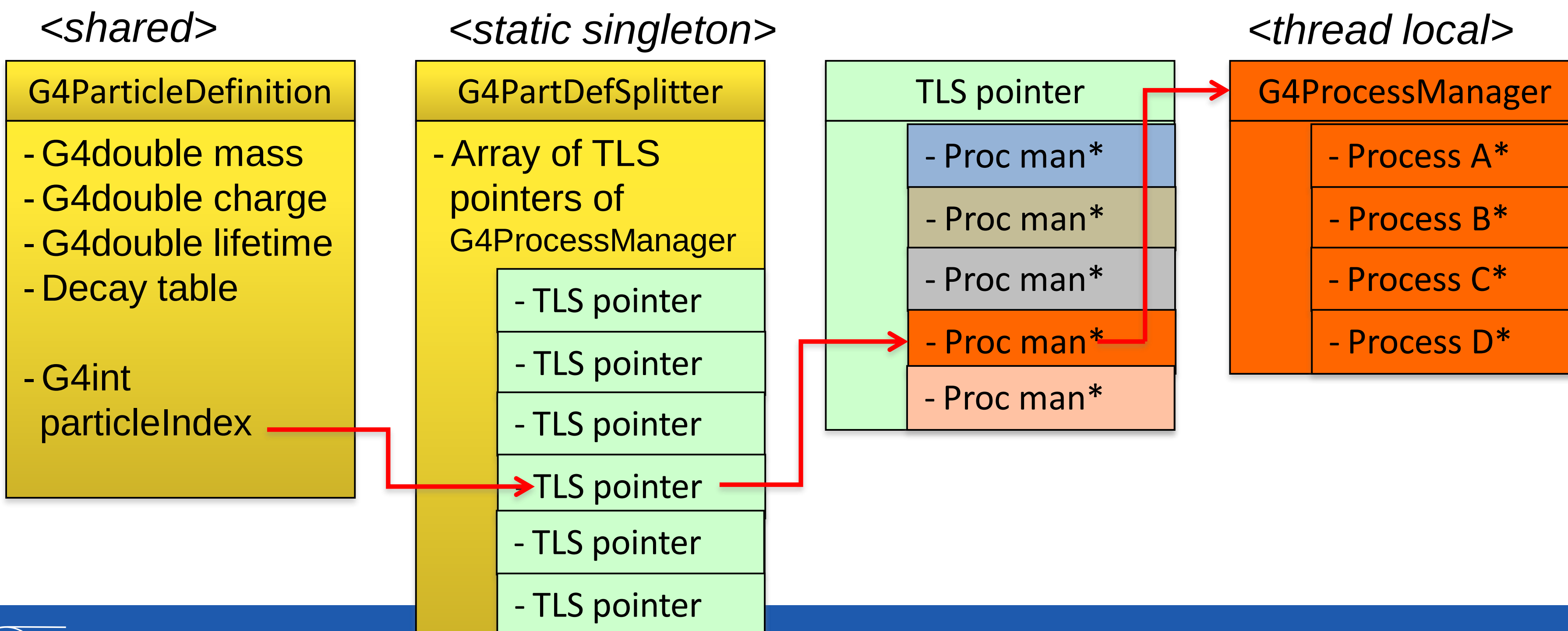
GEANT4 PARTICLES

Geant4 particles

- G4ParticleDefinition is the main object keeping static information about particles
  - Name, mass, charge, quantum numbers, decay table….
- "Stable" particles
  - Leptons: $e^{\pm}$, $\mu^{\pm}$, ….
  - Bosons: G4Gamma, G4OpticalPhoton, ….
  - Geantino is a particle without any interaction
  - "Stable" hadrons: $\pi^{\pm}$, $K^{\pm}$, ….
  - Light ions: d, t, ^{3}He, ^{4}He
  - G4GenericIon is used to define physics for all other ions
- "Unstable" hadrons normally do not tracked by Geant4 but used internally by hadronic models

Split class – case of particle definition

- In Geant4, each particle type has its own dedicated object of G4ParticleDefinition class.
  - Static quantities : mass, charge, life time, decay channels, etc.,
    - To be shared by all threads.
  - Dedicated object of G4ProcessManager : list of physics processes this particular kind of particle undertakes.
    - Physics process object must be thread-local.



GEANT4 PROCESSES

From G4Track to processes

G4Track
- Propagated by the tracking,
- Snapshot of the particle state.

G4DynamicParticle
- Momentum, pre-assigned decay…

G4ParticleDefinition
- The « particle type »:
  - **G4Electron**,
  - **G4PionPlus**…

G4ProcessManager
- Holds pointers to processes

Process_1, **Process_2**, **Process_3**
- Implementing physics processes

Handled by kernel

Configured by you, in your "*physics list*"

Geant4 process

- Processes are classified as:
  - Electromagnetic
  - Hadronic
  - Decay
  - Parameterized
  - Transportation
- Any process has process has type and sub-type
  - const G4String& G4VProcess::GetProcessType();
  - G4int G4VProcess::GetSubType();
    - This method is recommended to be used for MC truth
- Any process may be initialized using virtual methods:
  - **G4bool IsApplicable(const G4ParticleDefinition &);**
    - Used to check if a process can handle the given particle type
  - void PreparePhysicsTable(const G4ParticleDefinition&);
  - void BuildPhysicsTable(const G4ParticleDefinition&);
    - Used for initialization of internal data of the process

Geant4 Physics: Electromagnetic

- the standard EM part: provides a complete set of EM interactions (processes) of charged particles and gammas from 1 keV to ~PeV
  - Used practically in all kind of Geant4 applications
- the low energy EM part: includes special treatments for low energy e-/+, gammas and charged hadrons:
  - more sophisticated approximations valid down to lower energies e.g. more atomic shell structure details
  - some of these processes will be valid down to below keV but some can be used only up to few GeV
- optical photons: interactions special only for long wavelength photons
  - processes for reflection/refraction, absorption, wavelength shifting, (special) Rayleigh scattering
  - G4OpticalPhoton is the particle type
- Phonon physics is also implemented within Geant4

Geant4 Physics: Hadronic

- Pure hadronic interactions for 0 to 100 TeV
  - elastic, inelastic, capture, fission
- Radioactive decay:
  - both at-rest and in-flight
- Photo-nuclear interaction from ~1 MeV up to 100 TeV
- Lepto-nuclear interaction from ~100 MeV up to 100 TeV
  - e+ and e- induced nuclear reactions
  - muon induced nuclear reactions
- Very new processes of neutrino-nuclear interactions
- Processes of phonon interactions in crystals

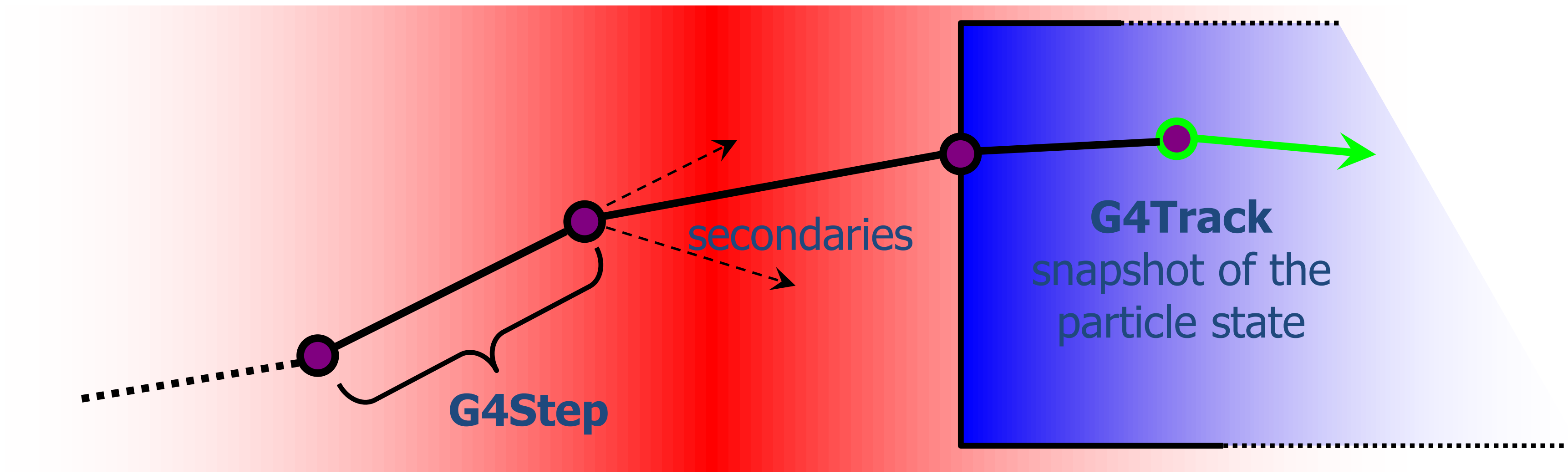
Geant4 Physics: Decay, Parameterized and Transportation

- decay processes includes:
  - weak decay (leptonic, semi-leptonic decay, radioactive decay of nuclei)
  - electromagnetic decay (π^0, Σ^0, etc.)
  - strong decay not included by default
    - they are part of hadronic models
    - may be assigned by a user to a particle
- parameterized process:
  - assigned to G4LogicalVolume
  - instead of step-by-step simulation provides hits in the logical volume and list of particles living the volume
  - for example, EM shower generation in a calorimeter based on parameters obtained from averaged events
- transportation process:
  - responsible for propagating a particle through the geometry in electromagnetic or gravitational field
  - needs to be assigned to each “stable” particle

WHAT HAPPENS AT A STEP

Geant4 tracking

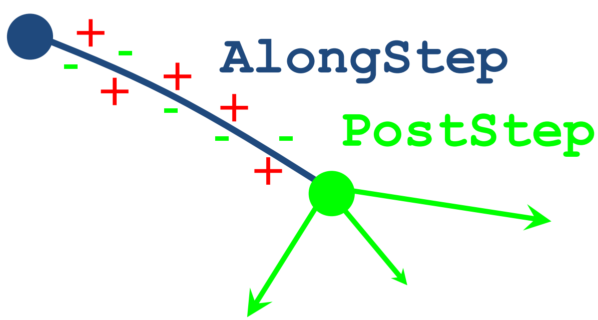
- **`G4Track`** is the object "pushed" step by step by the tracking :



- Moving by one step is the responsibility of the "stepping"
  - Which is the core engine of the "tracking" machinery
- These moves/steps are defined by physics or by geometry
  - Step length limit is a result of competition of processes
  - Any process may change the **`G4Track`**, let's see how.

G4VProcess: 3 kind of actions

- Abstract class defining the common interface of all processes in Geant4:
  - Used by all processes
    - including transportation, etc…
  - Defined in `source/processes/management`
- 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



G4VProcess : actions summary

- The virtual «action» methods are following:
  - `AtRestGetPhysicalInteractionLength(), AtRestDoIt();`
  - `AlongStepGetPhysicalInteractionLength(), AlongStepDoIt();`
  - `PostStepGetPhysicalInteractionLength(), PostStepDoIt();`
- Optional run time virtual methods:
  - **StartTracking(G4Track*);**
    - Allowing the process preparation for a new G4Track
  - **EndTracking();**
    - End of given G4Track

G4VProcess: extensions

- A process can implement any combination of the three **`AtRest`**, **`AlongStep`** and **`PostStep`** actions:
  - decay = **`AtRest`** + **`PostStep`**
- If you plan to implement your own process:
  - A set on intermediate classes exist implementing various combinations of actions:
    - For example:
      - **`G4VDiscreteProcess`**: only **`PostStep`** actions
      - **`G4VContinuousDiscreteProcess`**: **`AlongStep +`** **`PostStep`** actions

G4ProcessManager

- It is a Geant4 kernel class
  - A user should not change it
- **`G4ProcessManager`** maintains three vectors of actions :
  - One for the **`AtRest`** methods of the particle;
  - One for the **`AlongStep`** ones;
  - And one for the **`PostStep`** actions.
- Note, that the ordering of processes provided by/to the **`G4ProcessManager`** vectors is relevant and used by the stepping
  - There are few critical points you should be aware of
    - Multiple scattering can shift end point of a step
    - Scintillation, Cerenkov and some other processes assuming that step and energy deposition at the step are defined

IONS AND EXOTIC PARTICLES

Geant4 Approach for Ions

- Light ions are individual Geant4 particles:
  - G4Deuteron
  - G4Triton
  - G4He3
  - G4Alpha
- Generic ion serves all other ions:
  - G4GenericIon - only one particle
  - Not a real particle (charge +1, mass = Mp)
  - Serving for any kind of ion with Z>2
  - All concrete ions peak up processes and cross sections of the G4GenericIon
- Ion names
  - "C12" means that the carbon ion is in the ground state
  - "Co60[58.590]" is the first excitation state of Co60
  - Extra information about atomic shell may be filled

Ion energy loss model

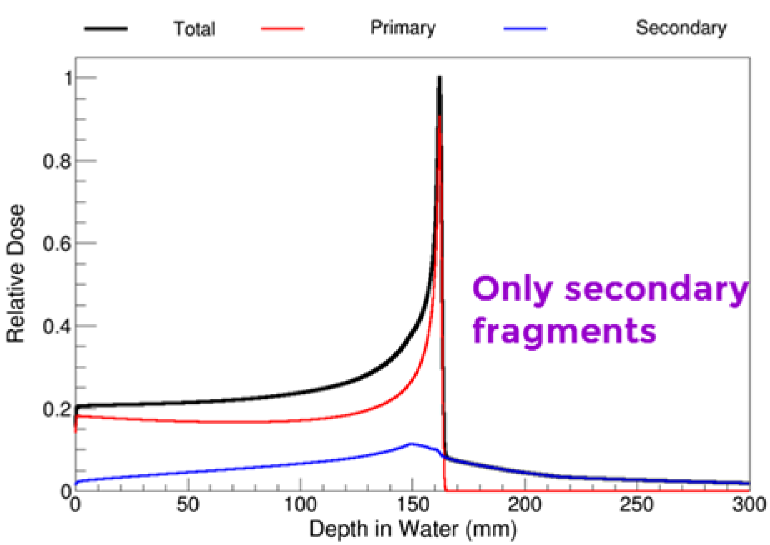
- Restricted stopping powers are calculated using 3 approaches
  - $T < T_{Low}$: Free electron gas model
  - $T_{Low} \leq T \leq T_{High}$: parameterization (ICRU'73) approach
  - $T > T_{High}$: Bethe-Bloch formula (using an effective charge and higher order corrections
- ICRU'73 parameterization

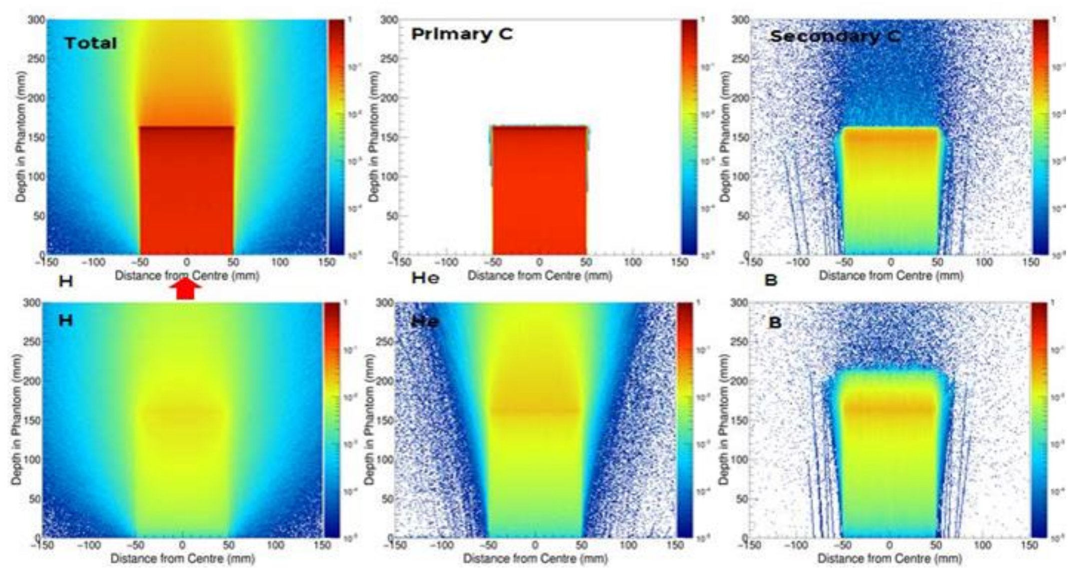
Nucl. Instrum. Meth. B 268 (2010) 2343-2354

  - Large range of ion-materials combination
    - Incident ions : projectile Z>2, targets all Z>0
    - Targets: all elemental materials, 31 compounds
  - Stopping powers based on the binary theory, effective charge approach
  - Special case: water
    - Revised ICRU'73 tables by P. Sigmund
    - Mean ionization potential is 78 eV
  - Energy limits
    - T_{High} = 1 GeV/nucleon
    - T_{Low} = 0.025 MeV/nucleon (lower boundary of ICRU'73 tables)

Carbon Ion Therapy

- Methods of usage of proton and carbon beam for cancer treatment have been established in recent ~20 years in Japan
- Geant4 was validated using HIBMC facility, which provides high quality beam allowing to measure of Bragg peak position in water with accuracy 0.1 mm.
  - Mixed radiation field produced by a Carbon Ion Beam with clinical energy (290 MeV/u)
  - This and other studies confirms, that nuclear fragmentation make a significant contribution to the radiation field





Exotic particles

- Not discovered particles are not part of Geant4 particle library
- To search exotics users should introduce non-existing particles in the user code
  - Such particles should be instantiated in ConstructParticle() method of one of user G4VPhysicsConstructor
  - User should take care attaching processes to exotic particles
- Geant4 offers two extended examples
  - $G4INSTALL/examples/extended/exoticphysics/monopole
  - $G4INSTALL/examples/extended/exoticphysics/dmparticle
  - These examples demonstrate different variants of addition of extra particles and
- In the monopole example additional classes are available for tracking of the magnetic monopole in magnetic field
  - G4MonopoleTransportation, G4MonopoleEquation

GEANT4 CUTS

How tracks are created and killed

- G4Track can be created
  - By G4VUserPrimaryGeneratorAction
  - By any G4VProcess
- All particles are tracked until it is killed by one of Geant4 processes, for example:
  - Transport out of the world volume
  - Inelastic interaction
  - Decay
  - G4NeutronKiller
  - If during tracking kinetic energy become zero and there is no processes AtRest the particle is killed by the stepping manager
- Geant4 introduced conception of “**cut in range**”
  - Physically this means required spatial accuracy of simulation
  - At initialization for each material a production threshold for kinetic energy of secondary particles is computed
    - This means different production thresholds for different materials
  - This is the main difference between Geant4 and other simulation tools, which implement only tracking cuts

Cut and production thresholds for energy loss processes

- User defines cut in range expressed in units of length

- Using this range Geant4 kernel compute production threshold T_{cut} for each material during initialization
- For a typical process (G4hIonisation, G4eIonisation, …), the production threshold T_{cut} subdivides the continuous and discrete parts of energy loss:
  - Mean rate of energy lost due to soft energy transfers
  - Total XS for discrete δ-electron production above T_{cut}

$$\frac{dE(E,T_{cut})}{dx} = n_{at} \int_0^{T_{cut}} T \frac{d\sigma(Z,E,T)}{dT} dT$$

$$\sigma(Z,E,T_{cut}) = \int_{T_{cut}}^{T_{\max}} \frac{d\sigma(Z,E,T)}{dT} dT$$

- At each step energy deposition is sampled by a fluctuation model using the computed mean energy loss

- Optionally, energy loss may be modified :
  - for the generation of extra δ-electrons under the threshold when the track is in the vicinity of a geometrical boundary (sub-cutoff)
  - for the sampling of fluorescence and Auger–electrons emission
- 4-momentum balance is provided in all cases

What particles have cut in range?

- Cuts in range are defined for
  - Gamma
  - Electron
  - Positron
  - Proton
- Cut for proton is used for all hadrons and ions by elastic scattering processes
- By default cut in range is defined globally
  - It is possible to have different cut in range for particle type
  - It is possible to define specific cut in range per G4Region

Which processes use cut in range ?

- It is not mandatory to use cuts
  - They are needed to achieve CPU performance of simulation
- Energy thresholds (derived from cut in range) are used
  - for gamma are used in Bremsstrahlung
  - for electrons are used in ionisation and e+e- pair production processes
  - for positrons is used in the e+e- pair production process
  - for gamma and electrons are used optionally (“ApplyCuts” options) in some discrete processes
    - Photoelectric effect, Compton, gamma conversion
- Production threshold for gamma and e^{+-} obtained from range cut cannot be whatever
  - The default low energy limit is 0.999 keV
  - May be changed via UI command:
    - /cuts/setLowEdge 100 keV
- Energy threshold for protons are used to define the threshold for kinetic energy of a nuclear recoil
  - EM single scattering process
  - Hadron elastic scattering

Tracking cuts

- Additionally to cut in range it is possible to use various tracking cuts
  - Unwanted particles may be killed after the step if corresponded flag is proposed
- In the default physics configurations two types of tracking cuts are applied:
  - Low-energy thresholds for charged particles by ionization 1 keV
  - Time cut for neutron transport 10000 ns
- Tracking cuts values are customizable and can be changed via UI commands
  - Energy cut may be set to zero
- User may easily setup extra tracking cut or step limiter
  - The best is to add extra G4VProcess

SPECIAL EM TOPICS

Steering of EM physics

- EM parameters of any EM physics list may be modified at initialization of Geant4 using C++ interface to the G4EmParameter class or via UI commands
- Example of interfaces of G4EmParameters:
  - SetMuHadLateralDisplacement()
  - SetMscMuHadRangeFactor()
  - SetMscMuHadStepLimitType()
- Corresponding UI commands:
  - /process/msc/MuHadLateralDisplacement
  - /process/msc/RangeFactorMuHad
  - /process/msc/StepLimitMuHad
- Some other UI commands:
  - /process/em/deexcitationIgnoreCut true
  - /process/eLoss/UseAngularGenerator true
  - /process/em/lowestElectronEnergy 50 eV
  - /process/em/lowestMuHadEnergy 100 keV
  - ....

User Interfaces and Helper Classes

- Geant4 UI commands to define cuts and other EM parameters
- G4EmCalculator
  - easy access to cross sections and stopping powers (TestEm0)
- G4EmParameters
  - C++ interface to EM options alternative to UI commands
- G4EmSaturation
  - Birks effect (recombination effects)
- G4ElectronIonPair
  - sampling of ionisation clusters in gaseous or silicon detectors
- G4EmConfigurator
  - add models per energy range and geometry region

How to extract Physics ?

- Possible to retrieve Physics quantities using a G4EmCalculator object
- Physics List should be initialized
- Example for retrieving the total cross section of a process with name procName, for particle and material matName

```
#include "G4EmCalculator.hh"
...
G4EmCalculator emCalculator;

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

- A good example: $G4INSTALL/examples/extended/electromagnetic/TestEm0
  Look in particular at the RunAction.cc class

HOW TO USE PARTICLES, PROCESSES, CUTS

Start examples/extended/electromagnetic/TestEm7

- Start VM and open a terminal window
- Copy, compile, and build the example into working area
  - cd $G4WORKDIR
  - mkdir TaskM
  - cd TaskM
  - cp $G4INSTALL//share/Geant4-10.5.0/examples/extended/electromagnetic/TestEm7 ./
  - cd TestEm7
  - mkdir build
  - cd build
  - cmake -DGeant4_DIR=${G4COMP} ../
  - make
- Open interactive session
  - ./TestEm7
  - /control/execute/vis.mac
  - /run/beamOn 1

Displaying processes and particles

- When you application has started and when the run manager has been initialized, you can:

- Check the physics processes attached and their ordering:
  - **`/particle/select e-`**
  - **`/particle/processes/dump`**

- Check what particles exist:
  - **`/particle/list`**
- Check a particle property:
  - **`/particle/select e-`**
  - **`/particle/property/dump`**
- Please type “help” to get the full set of commands for particle category

Cuts definition

- Using UI interface Geant4 kernel change cuts and try to count number of steps in the same run
  - /run/setCut 0.01 mm
  - /run/beamOn 100
- Define cuts only for electrons
  - /run/setCutForAGivenParticle e- 10 um
  - /run/dumpCouples
- How simulation results for proton Bragg peak depend on cuts?
  - Change cut in range value and run 100 events