GEANT4
A SIMULATION TOOLKIT
Version 10.5

Particles and Processes

Vladimir Ivantchenko (CERN & Tomsk State University, Russia)

Geant4 Beginners Course

Outline

- Particles
- Processes
- What happens at a step
- Cuts

Geant4 interface to physics

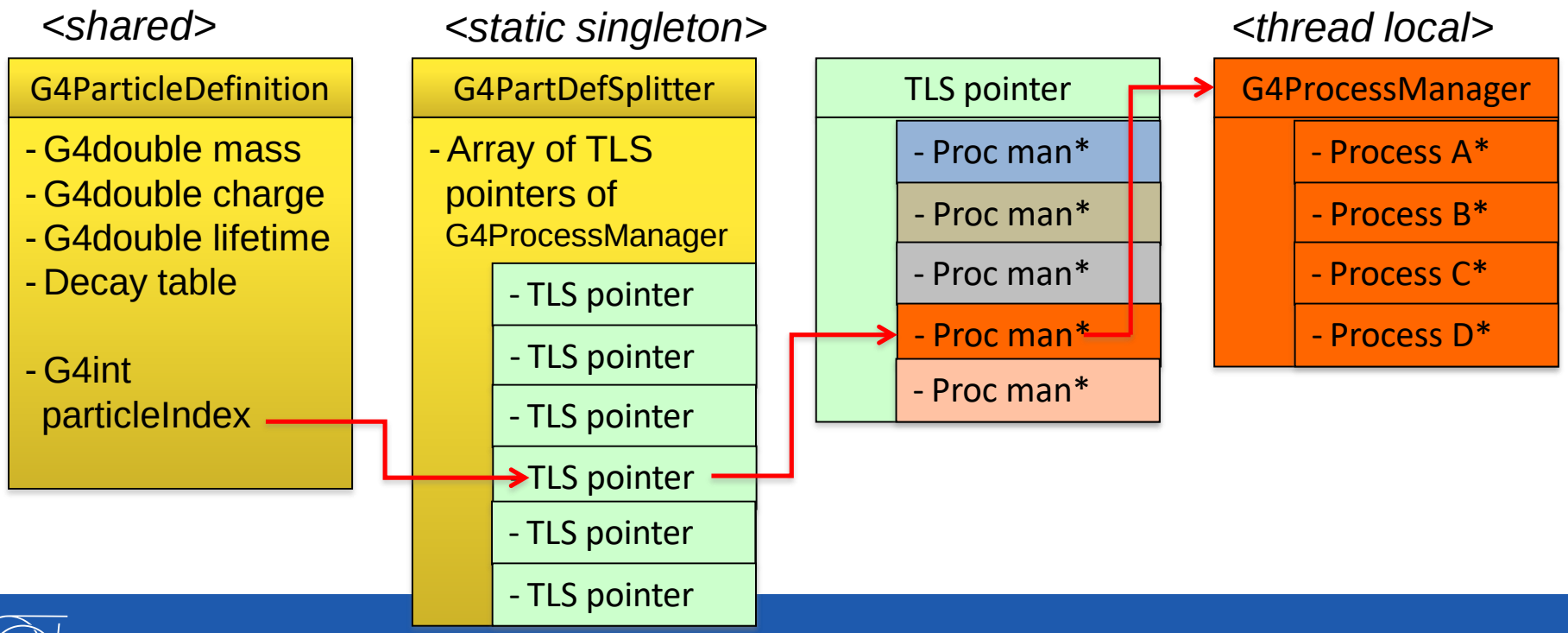
- The interface of Geant4 kernel to physics is abstract
- Base physics abstract classes are following:
  - The `G4ParticleDefinition` objects shared between threads
  - The `G4VProcess` thread local objects
  - The `G4ProcessManager` thread local interface class
- These interfaces are stable for ~20 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
  - Details in the next lecture
- A user may be also a developer of a custom particle or process

Geant4 particles

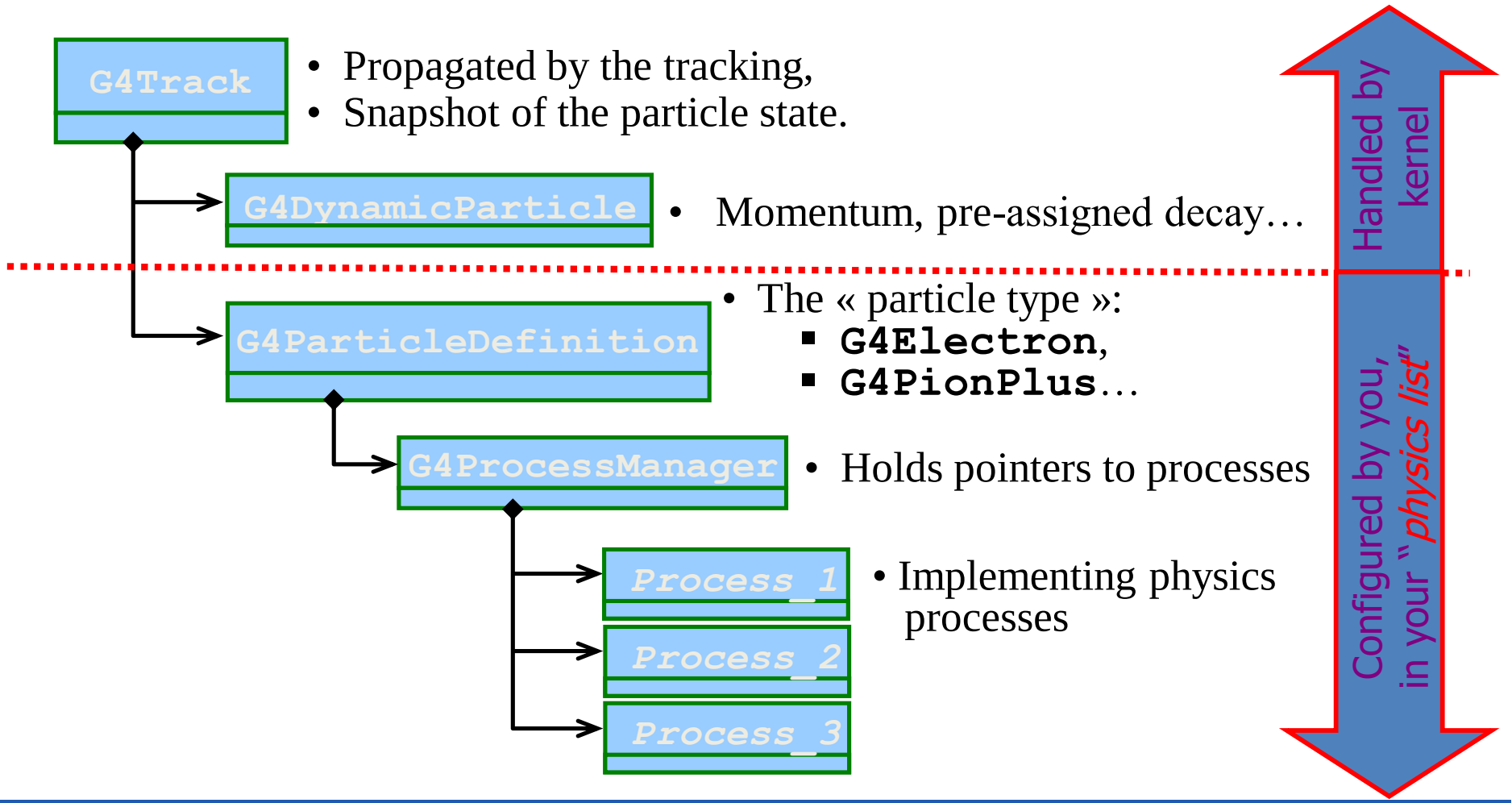
- G4ParticleDefinition is the main object keeping static information about particles
  - Name, mass, charge, quantum numbers, decay table….
- “Stable” particles
  - Leptons: $e^{\pm}$, $\mu^{\pm}$, ….
  - Bosons: G4Gamma, G4OpticalPhoton, ….
  - Geantino is a particle without any interaction
  - “Stable” 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.



From G4Track to processes



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

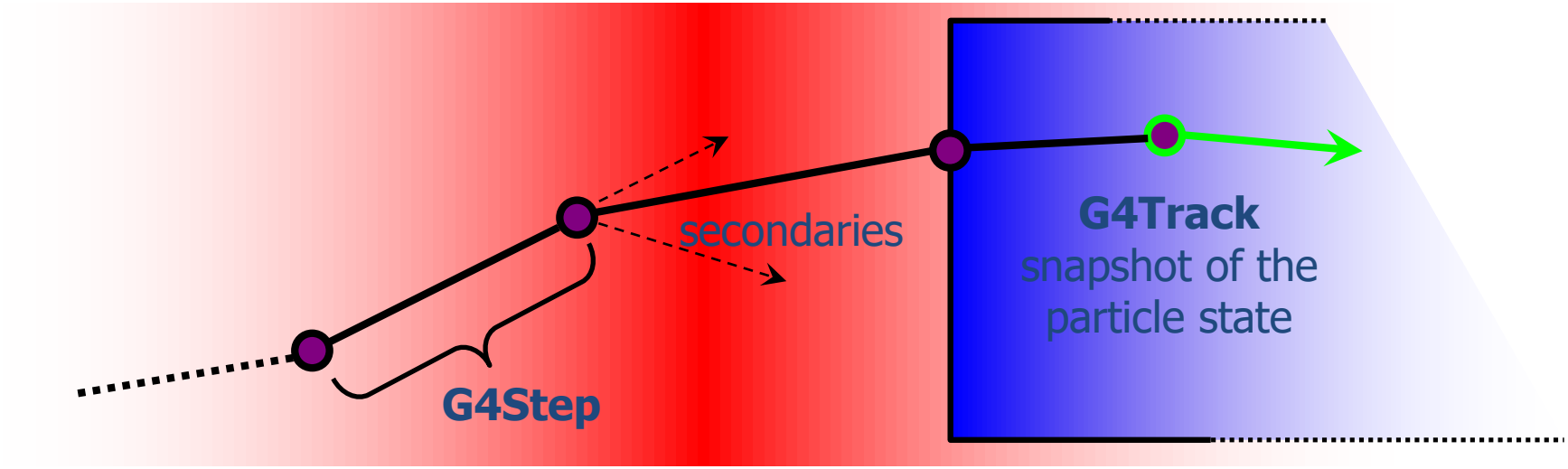
- Pure hadronic interactions for 0 to 100 TeV
  - elastic, inelastic, capture, fission
- Radioactive decay:
  - both at-rest and in-flight
- Photo-nuclear interaction from ~1 MeV up to 100 TeV
- Lepto-nuclear interaction from ~100 MeV up to 100 TeV
  - e+ and e- induced nuclear reactions
  - muon induced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

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

Geant4 tracking

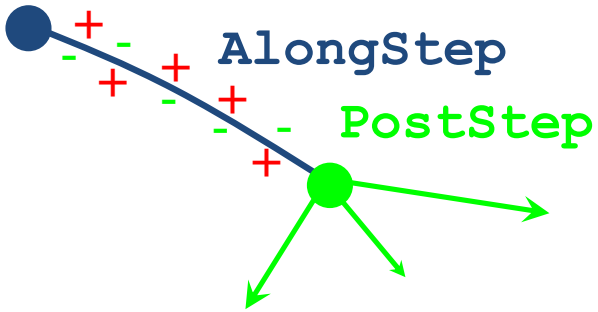
- **`G4Track`** is the object "pushed" step by step by the tracking :



- 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

Geant4 approach for cuts

- All particles are tracked until it is killed by one of Geant4 processes, for example:
  - Transport out of the world volume
  - Inelastic interaction
  - Decay
- 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 tracking cuts

What particles have cut in range?

- Cuts 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 for gamma are used in Bremsstrahlung
- Energy thresholds for electrons are used in ionisation and e+e- pair production processes
- Energy threshold for positrons is used in the e+e- pair production process
- Energy thresholds for gamma and electrons are used optionally ("ApplyCuts" options) in some discrete processes
  - Photoelectric effect, Compton, gamma conversion
- Energy threshold for protons are used in processes of elastic scattering for hadrons and ions defining the threshold for kinetic energy of nuclear recoil

Tracking cuts

- Additionally to cut in range it is possible to use various tracking cuts
- In the default physics configurations two types of tracking cuts are applied:
  - Low-energy thresholds for charged particles by ionization process 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
- Production threshold for gamma and e^{+-} obtained from range cut cannot be whatever
  - There are low energy limits for these values
  - The default value is 0.999 keV

HANDS-ON TASK M1

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