Version 11.0

Overview

John Apostolakis (CERN)

Geant4 Advanced Course

Recall: What is Geant4?

- Geant4 is a **toolkit**: it provides all capabilities required for particle transport simulation
  - Defines beams (or sources)
  - Enables user to defined the setup/model of material, geometry, field
  - Includes a complete set of physics models for all interactions (many times with options for speed or precision)
  - Tracks particles through the geometry setup/model
  - Records information in general ways (scoring) and in fully customisable ways (hits)
  - Provides tools to configure, visualize and store outputs scoring and hits
- Geant4 provides template for simple applications, ranging from
  - Full ('advanced') applications for key, targeted topics (e.g. hadrontherapy, ..)
  - Extended examples which demonstrate important capabilities
  - Basic examples that are a starting place, and potential starting points for others;
- Geant4 can be used to build applications (tools) for a set of problems and a community, e.g.
  - Accelerator modelling (interactions with collimators, beampipes, ..)
  - Medical imaging and/or radiotherapy

Starting point and what to expect

- This course expects that you already
  - know the overall structure of Geant4,
  - have written (at least parts) of a Geant4 application,
  - have a minimum working knowledge of C++, including knowledge of key modern C++11/14 features (auto, ..)
- The course will go in depth each area of Geant4, and seeks to cover
  - the overall picture of capabilities or physics models,
  - the key aspects which a user should know or control
  - inform and guide you to avoid some pitfalls
  - but not a 'total' coverage of all capabilities, corner cases etc.
- We hope and expect that you will ask questions frequently, e.g.
  - about anything that is unclear
  - to understand how to choose capabilities, and whether they are applicable,
  - about the differences between different options offered.

The topics of each day

- Day 1 – Physics (processes & models, EM physics, hadronic physics)
- Day 2 – Geometry and EM Field
- Day 3 – Multithreading, recording information (scoring) and other ‘user’ classes
- Day 4 – More about physics (optical, physics lists, hadronic physics)
- Day 5 – How to get faster results (event biasing, fast simulation)

Day 1: Physics Part 1

- Particles and Processes
  - What is a Geant4 ‘physics process’ ?
  - What **classes** describe a particle, a process, a track ? What does each class **represent** and hold ?
  - What happens in a simulation **step** ? How does tracking ‘call’ the chosen processes & compose their results?
  - How is a custom process created ?
- EM physics – for electrons, gammas, charged hadrons
  - How are **energy loss** processes modelled for charged particles?
  - Why production thresholds (‘**cuts**’), why are they needed, and how do we configure them?
  - Are physics model configurations with different precision/computing cost available – how to choose them?
    - Can one select an alternative model for part (region) of the geometry / setup ?
  - How to know if our application needs atomic **deexcitation**, and how to enable it ?
- Hadronic physics – first what is it and who needs it?
  - What are the challenges in modelling it?
  - How is it organised (framework) – and how is the process (interaction) that will occur chosen ?
  - What parts model the results of interactions ? Are there lookup tables for ‘difficult’ interactions ?

Day 2: Describing the setup

- Geometry / Materials
  - Do you tream ‘individual’ **unique** volumes differently from ones that are **repeated** thousands of times?
  - How can you describe a complicated **hierarchy** of volumes ?
  - How to select **part of your geometry** for special treatement ?
  - Could I measure the neutron flux in a volume grid that **overlaps** much of my detector ?
- External Electromagnetic (EM) Fields
  - How to **define a magnetic field** for the whole detector/setup or part of it
  - What methods are available to **integrate** the path of charged particles in an (electro)magnetic field ?
  - Does Geant4 offer ‘perfect’ **accuracy** for integration of tracks and the intersection with surface boundaries?
  - What you need to know about the **approximations** that must be made – and how to **control** them.

Day 4: Physics Part 2

- Optical photons
  - What processes **generate** them and how are they **propagated** ?
  - What optical physics processes are included ?
  - Idiosyncrasies and challenges.
- Physics lists
  - How to obtain a **complete, consistent** configuration of physics models / cross sections ?
  - How to **compare results** between experiments ? – and other benefits of using a common configuration.
  - What are the **pre-packaged physics lists**, and how to choose one ?
    - Or .... where you will try to decypher what FTFP_BERT and QGS_BIC_HP mean. Hopefully.
- More hadronic physics
  - Are hadronic models as reliable as models of electromagnetic interactions?
  - What are the hadronic models used in Geant4 – are some applied only in a specific energy range?
  - Neutron physics – can G4 use the extensive data libraries from years of measurements and detailed modeling ?

Day 3: Recording information – and using multiple threads

- Multithreading (MT)
  - How does Geant4 Multi-threading work ?
  - What do I need to do to adapt my application to work with MT ?
  - How to write out
- Recording information (scoring)
  - Does geant4 have built-in ways to estimate energy deposition, flux and dose ?
  - How can I define what I want to measure ?
  - If I need something more complicated for my detector / experiment, what do I do?
- Other 'user' classes allow you to view what is going on inside a Geant4 simulation. You can
  - get **your code** called **at each step**, to investigate, verify or just record something.
  - create something at the start or end of a 'run' or 'event'
  - have Geant4 propagate your information that you attached to an Event, Track, Primary or Region
  - stop Geant4 simulating particles that go outside a timing window, or go too far in length.
- User commands for your own configuration parameters – your custom UI commands
  - How to set run parameters or control a position
  - How to change size of a piece of your geometry

Day 5: Speeding up my simulation: can it be done ?

- Are there methods to get faster results, and what are their strengths / drawbacks ?
  - Is there a free lunch here?
  - What effort is required ?
  - Is it a ‘science’ or an art ?
- Event biasing are a set of techniques for specific applications/observables
  - Chooses to focus the simulation time/computation on ‘important’ particles/tracks
  - Kills some tracks, make others more important (carry more ‘weight’)
  - Force a particular interaction to occur (especially if its probability would be very small)
  - What types of applications are suitable – and how to ensure correctness ?
- Fast simulation
  - When doing detailed simulation is just too much work, can you leverage Geant4 to get faster results?
  - How to decide if your simulation is a candidate for this ?
  - What ‘tools’ have been created – both traditional and recent (Machine Learning based) ?
  - How can you verify that your results are correct ?

Areas that we do not cover in this course

- Analysis
  - Geant4 offers an embedded analysis module (portable between OSes) – adapted to MT
  - You can use the analysis package that you are familiar with already (e.g. Root) instead
- Visualization
  - There are multiple visualisation options (Qt, OpenGL, OpenInventor, dawn);
  - Capabilities
    - Basic capabilities are demonstrated in the beginner course(s);
    - Some capabilities are specific to certain drivers;
    - The common capabilities of drivers have greatly increased over the last years;
  - Uses
    - It can be used to verify a setup, understand behaviour of tracks or to produce ‘pretty pictures’
    - It is less used in large HEP experiments, which have other ‘event display’ visualisation;
  - Due to time/effort constraints, currently it is beyond the scope of this course.