Version 11.2

Course 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 in 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 will not (cannot) be a ‘total’ coverage of all capabilities, corner cases etc.
- But we hope and expect that you will **ask questions frequently**, e.g.
  - about anything that is unclear
  - to understand whether capabilities they are applicable,
  - how to make choices between different methods or physics models,
  - about the differences or strengths of the options/models offered.
- Please use MatterMost to ask your questions

The main topics of each day

- Day 1 – Geometry (part 1); framework for physics processes & models; hadronic physics; multi-threading (intro)
- Day 2 – Geometry (part 2); EM Physics; Visualisation; and Multithreading (more)
- Day 3 – How to get faster results (Fast simulation); Recording Information (scoring) and 'user' classes
- Day 4 – More about physics (optical, physics lists); external EM fields; defining your own User Commands
- Day 5 – How to get faster results (event biasing, physics biasing), more on Hadronic Physics

- We intentionally split several of the 'big' topics into lectures over two days
  - to give you a chance to absorb key concepts,
  - to prepare questions about seems unclear or needs clarification, and
  - to have a better chance to build on what you have absorbed.

- Discussion sessions: a chance to come back to a topic, ask more questions

- New: on Tuesday-Friday there is a short session for questions about the previous day, including any open questions posed, exercises suggested, …

Describing the setup / detector – days 1, 2 and 4

- Geometry (days 1 & 2)
  - What shapes (solids) are available to describe individual volumes?
  - How can you describe a complicated hierarchy of volumes ?
  - How to check for errors (overlaps) in your geometry model
- External ElectroMagnetic Fields (day 4)
  - 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.
  - If the default method is not giving the speed or accuracy, are there alternatives?

Physics framework, models and processes

- How are physics processes modelled in Geant4 ?
- What are the key objects involved ?
  - G4Track: a snapshot of the state of the current of a track
  - G4Step: the current status, with a start and end state
- Geant4's process interface: G4Process
  - What is a process' role in G4 ?
  - What does a process need to implement ?
  - How can it affect the state of a track ?
- Overview of key 'families' processes
  - EM processes
  - Hadronic processes
  - Decay
- These are spread out over several days

Day 1

- Processes & models
  - How Geant4 organises physics
- Hadronic physics
  - Types of models
  - Key models

Day 2

- EM Physics

- Day 4
  - Optical photons
  - Physics lists

- Day 5
  - More hadronic physics

Recording information – and using multiple threads

- Recording information (scoring) – day 3
  - What are the different ways that you can record information?
  - How to create ‘tallies’ to score flux, dose without C++ code
  - How to retain detailed, per-event information in hits
- Other ‘user’ classes
  - How to customise the simulation at each step, or each track start/end
- Configuring your application at run-time
  - How to create your own ‘user commands’

- Multithreading (MT) – days 1 & 2
  - How does Geant4 Multi-threading work ?
  - What do I need to do to adapt my application to work with MT ?
  - How to write out results

Days 3 & 5 - Speeding up your simulation

- Generally there are different methods to get **faster results**. The most popular are
  - '**bias**' the simulation to spend time on 'important' particles, and mostly **ignore** all the rest
  - Replace the detailed simulation with **samples** from the same or similar distributions
  - Create (train) a model which generates similar results – either parameterising or using ML

- What is fast simulation and how to 'hook' it into Geant4 ? (Day 4)
  - How to replace detailed (step by step) simulation with estimated/sampled results
  - Event biasing chooses to focus the simulation time/computation on 'important' parts of the
  - Are Machine Learning methods used ? How about other methods ?

- What to do if a simulation wants to estimate 'first order' quantities, and many / most particles don't contribute ?
  - Event biasing is a set a techniques to choose which tracks to emphasize or eliminate, and get correct results fast
  - Some techniques are general, others involve the implementations of physics processes
  - This will be in Day 5.

User Interfaces (UI) and Visualisation

- For the first time we will cover UI and Visualisation in this course
- Geant4 comes with several User Interface options
  - both text driven – the ‘tcsh-like’ offers history of previous commands, copying and editing
  - Graphical: primarily Qt (with menus created from Geant4 and user ’messengers’)
- Visualisation allows one to display
  - The geometry of the detector or setup
  - The trajectory of tracks (either curved trajectories in field, or simpler lines)
  - Cutaways of a setup
  - Field lines of magnetic field (in example)
- There is a rich set of visualisation options with different purposes
  - Output to files (ascii, eps)
  - Output to external programs to render (‘dawn’ , gMocren, ..)
  - ‘Online’ direct output to screen (e.g. OpenGL or VTK)
- To benefit from these there are requirements
  - We list the options now, and
  - Ask (poll) how many have installed some key options

UI and Visualisation: requirements

- For features of UI and Vis:
  - A build with interaction and graphics
  - If possible, with Qt
    - Installing
      - MacOS: **brew install qt**
        - For Qt5: **brew unlink qt** (to avoid linking to Qt6)
      - Linux: **apt install qtbase5-dev**
      - Windows: Binaries from Qt web site (qt.io)
    - Building—an example cmake command:
- In the Homework/questions
  - Try UI and Vis features
  - Explore a GDML file?
    - Requires Xerces
    - **brew install xerces-c**

```
cmake -Wno-dev --log-level=ERROR \
-DCMAKE_INSTALL_PREFIX=`pwd` \
-DGEANT4_USE_GDML=ON \
-DGEANT4_USE_QT=ON \
-DCMAKE_PREFIX_PATH=\
"$(brew --prefix qt@5)" \
-DGEANT4_ENABLE_TESTING=ON \
-DGEANT4_USE_FREETYPE=ON \
-G Xcode \
~/Geant4/geant4
```

Needed libraries – for reference

- Installing X11
  - Linux: native
  - Windows: X11 not available, but special drivers use the native windowing system.
  - MacOS: Xquartz
- Installing Qt
  - Linux: **apt install qtbase5-dev**
  - Windows: Binaries from Qt web site (qt.io)
  - MacOS: **brew install qt**
    - For Qt5: **brew unlink qt** (to avoid linking to Qt6)
- Installing VTK
  - Linux: **apt install vtk**
  - Windows: **conda install vtk**
  - MacOS: **brew install vtk**
- Installing Coin3D (Open Inventor)
  - Coind3D (Open Inventor): See https://www.coin3d.org

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

Specialised physics – which require further depth

- Atomic de-excitation in EM physics – details are not discussed
  - Yet it is present for use in the most sensitive applications in medical physics from imaging to therapy, eg calibration of gas chambers
- ‘DNA’ physics models, for liquid water or other specific material (eg models of chromosomes)
- Phonons
- Channeling

How to ask questions - suggestions

- All: please try to use MatterMost to ask questions (even in the room)

- This lets our ‘coordinator’ know – he/she will look for a good time to raise the question.
  - If it needs a quick answer (eg yes/no), he/she may answer it
- Let him/her know
  - if you prefer to ask it yourself
  - whether you are in person or remote.

- For a followup question, please raise your hand in the room or online
  - Or, if needed, just ask your question (e.g. if remote un-mute and talk)

- If you are trying to follow, but are lost don’t hesitate – ask for clarification!
  - Others likely will share your question or difficulty

- Eventually if time becomes tight we may need to delay questions to the discussion afterwards