GENERAL INTRODUCTION

Training course “Monte Carlo simulation for Micro- and Nanodosimetry”
Karlsruhe, 25-26 October 2011

V. Ivanchenko
adaptation of the original lecture of
Makoto Asai (SLAC)
Outline

- General introduction and brief history
- Highlights of user applications
- Geant4 license
- Geant4 kernel
  - Basic concepts and kernel structure
  - User classes

25-26 October, 2011, Geant4 Introduction
Geant4 – Its history

- Dec ’94 - Project start
- Apr ’97 - First alpha release
- Jul ’98 - First beta release
- Dec ’98 - First Geant4 public release - version 1.0
- Dec 19th, ’08 - Geant4 version 9.2 release
- Dec 18th, ’09 - Geant4 version 9.3 release
  - Sep 24th, ’10 - Geant4 9.3-patch02 release
- Dec 17th, ’10 - Geant4 version 9.4 release
  - June 24th, ’11 - Geant4 9.4-patch02 release
- We currently provide one public release every year.
  - Beta releases are also available.
  - Release announcements on Collaboration Web pages and through the announcement mailing list

25-26 October, 2011, Geant4 Introduction
In order to meet wide variety of requirements from various application fields, a large degree of functionality and flexibility are provided.

Geant4 has many types of geometrical descriptions to describe most complicated and realistic geometries
- CSG, BREP and Boolean solids
- Placement, replica, divided, parameterized, reflected and grouped
- XML interface

Everything is open to the user - user may become a developer
- Choice of physics processes/models
- Choice of GUI/Visualization/persistency/histogramming technologies
Collaborators also from non-member institutions, including
Budker Inst. of Physics
IHEP Protvino
MEPHI Moscow
Pittsburg University

25-26 October, 2011, Geant4 Introduction
DEPARTMENT OF PHYSICS UNIVERSITY OF BRITISH COLUMBIA VANCOUVER  V6T 1Z1 CANADA

25-26 October, 2011, Geant4 Introduction


GEANT4 - A simulation toolkit

View at publisher  Set feed
HIGHLIGHTS OF USERS APPLICATIONS

To provide you some ideas how Geant4 would be utilized...
BaBar at SLAC is the pioneer experiment in HEP in use of Geant4

- Started in 2000
- Simulated $\sim 2 \times 10^{10}$ events so far
- Produced at 20 sites in North America and Europe
Geant4 now used by all LHC detectors

- ATLAS, CMS – greatest detectors
- LHCb, ALICE – large specific detectors

25-26 October, 2011, Geant4 Introduction
Events with > 50000 particles/event in detector acceptance

Albert De Roeck (CERN)
Boulby Mine dark matter search Prototype Simulation

One High Energy event

Courtesy of H. Araujo, A. Howard, IC London
Example: Helical Channel

72 m long solenoidal + dipole field with wedge absorbers and thin cavities

Published in proc. of PAC 2001
(Fermilab-Conf-01-182-T)

\[ B_{xy} = B_T \cos \left( \frac{2p}{L} z \right) \]
\[ B_z = B_0 \]

Other simulations:
- Alternate Solenoid Channel (sFoFo), published in proceedings of PAC2001 and Feasibility Study II for a Neutrino Factory at BNL (2001)
- Bent Solenoid Channel, presented at Emittance Exchange Workshop, BNL 2000
- Low Frequency r.f. Cooling Channel, presented at International Cooling Experiment Workshop, CERN 2001
- Cooling Experiment (MICE) Simulation (in progress)
• Launch December 1999
• Perigee 7000 km
• Apogee 114000 km
• Flight through the radiation belts

• Chandra X-ray observatory, with similar orbit, experienced unexpected degradation of CCDs
• Possible effects on XMM?
• First mission simulated with Geant4
γ astrophysics

γ-ray bursts

Typical telescope:
- Tracker
- Calorimeter
- Anticoincidence
- γ conversion
- electron interactions
- multiple scattering
- δ-ray production
- charged particle tracking
PlanetoCosmics
Geant4 simulation of Cosmic Rays in planetary Atmo-/Magneto- spheres

Geant4 Simulation of the Propagation of Cosmic Rays through the Earth’s Atmosphere

Physikalisches Institut, University of Bern, CH-3012 Bern, Switzerland

Cutoff Rigidities vs position
PlanetoCosmics
Mars field and atmosphere

- NASA Mars-GRAM2001 model

25-26 October, 2011, Geant4 Introduction
Geometry examples of GATE applications

- Multi-ring PET
- Triple-head gamma camera

D. Strul
IPHE Lausanne

S. Staelens
Uni Ghent
GEANT4 based proton dose calculation in a clinical environment: technical aspects, strategies and challenges
Screen shots of gMocren

25-26 October, 2011, Geant4 Introduction
The Geant4 License

In response to user requests for clarification of Geant4’s distribution policy, the collaboration established a license.

- Makes clear the user’s wide-ranging freedom to use, extend or redistribute Geant4, even as part of some for-profit venture.
- Simple enough that you can read and understand it.

http://cern.ch/geant4/license/
Geant4 License Highlights

- Establishes the Geant4 copyright
- Prohibits others from claiming that they are Geant4

- If you develop something in or based on Geant4 and give it away, Geant4 can have it for free, too
- Any documentation you produce must refer to Geant4
- You cannot patent the parts already written by the collaboration

- We don't claim that it works, and we're not responsible if it doesn't
BASIC CONCEPTS
AND KERNEL STRUCTURE
Internal unit system used in Geant4 is completely hidden not only from user’s code but also from Geant4 source code implementation.

Each hard-coded number must be multiplied by its proper unit.

\[
\text{radius} = 10.0 \times \text{cm}; \\
\text{kineticE} = 1.0 \times \text{GeV};
\]

To get a number, it must be divided by a proper unit.

\[
\text{G4cout} \ll \text{eDep} / \text{MeV} \ll \text{“[MeV]”} \ll \text{G4endl};
\]

Most of commonly used units are provided and user can add his/her own units.

By this unit system, source code becomes more readable and importing / exporting physical quantities becomes straightforward.

For particular application, user can change the internal unit to suitable alternative unit without affecting to the result.
G4cout, G4cerr

- G4cout and G4cerr are ostream objects defined by Geant4.
  - G4endl is also provided.
    
    ```
    G4cout << "Hello Geant4!" << G4endl;
    ```

- Some GUls are buffering output streams so that they display print-outs on another window or provide storing / editing functionality.
  - The user should not use std::cout, etc.

- The user should not use std::cin for input. Use user-defined commands provided by intercoms category in Geant4.
  - Ordinary file I/O is OK.
Terminology

- **Step** – the smallest unit of Geant4 simulation, a particle is transported from one point to another
- **Trajectory and TrajectoryPoint** – collection of steps and step points
- **Process** – the physics that happens along a step
- **Track** – a snapshot of a particle at some point along its path (not the same as trajectory)
- **Event** – a collection of info from tracks and particle trajectories
- **Run** – a collection of events
Geant4 consists of 17 categories.

- Independently developed and maintained by WG(s) responsible to each category.
- Interfaces between categories (e.g. top level design) are maintained by the global architecture WG.

**Geant4 Kernel**

- Handles run, event, track, step, hit, trajectory.
- Provides frameworks of geometrical representation and physics processes.
Run in Geant4

- As an analogy of the real experiment, a run of Geant4 starts with “Beam On”.
- Within a run, the user cannot change
  - detector setup
  - settings of physics processes
- Conceptually, a run is a collection of events
  - A run consists of one event loop.
- At the beginning of a run, geometry is optimized for navigation and cross-section tables are calculated according to materials appear in the geometry and the cut-off values defined.
- G4RunManager class manages processing a run, a run is represented by G4Run class or a user-defined class derived from G4Run.
  - A run class may have a summary results of the run.
- G4UserRunAction is the optional user hook.
An event is the basic unit of simulation in Geant4.

At beginning of processing, primary tracks are generated. These primary tracks are pushed into a stack.

A track is popped up from the stack one by one and “tracked”. Resulting secondary tracks are pushed into the stack.

- This “tracking” lasts as long as the stack has a track.

When the stack becomes empty, processing of one event is over.

**G4Event** class represents an event. It has following objects at the end of its (successful) processing.

- List of primary vertices and particles (as input)
- Hits and Trajectory collections (as output)

**G4EventManager** class manages processing an event. **G4UserEventAction** is the optional user hook.
Track is a snapshot of a particle.
- It has physical quantities of current instance only. It does not record previous quantities.
- Step is a “delta” information to a track. Track is not a collection of steps. Instead, a track is being updated by steps.

Track object is deleted when
- it goes out of the world volume,
- it disappears (by e.g. decay, inelastic scattering),
- it goes down to zero kinetic energy and no “AtRest” additional process is required, or
- the user decides to kill it artificially.

No track object persists at the end of event.
- For the record of tracks, use trajectory class objects.

G4TrackingManager manages processing a track, a track is represented by G4Track class.

G4UserTrackingAction is the optional user hook.
At the end of each step, according to the processes involved, the state of a track may be changed.
- The user can also change the status in UserSteppingAction.
- Statuses shown in green are artificial, i.e. Geant4 kernel won’t set them, but the user can set.

- **fAlive**
  - Continue the tracking.

- **fStopButAlive**
  - The track has come to zero kinetic energy, but still AtRest process to occur.

- **fStopAndKill**
  - The track has lost its identity because it has decayed, interacted or gone beyond the world boundary.
  - Secondaries will be pushed to the stack.

- **fKillTrackAndSecondaries**
  - Kill the current track and also associated secondaries.

- **fSuspend**
  - Suspend processing of the current track and push it and its secondaries to the stack.

- **fPostponeToNextEvent**
  - Postpone processing of the current track to the next event.
  - Secondaries are still being processed within the current event.
Step in Geant4

- Step has two points and also “delta” information of a particle (energy loss on the step, time-of-flight spent by the step, etc.).
- Each point knows the volume (and material). In case a step is limited by a volume boundary, the end point physically stands on the boundary, and it logically belongs to the next volume.
  - Because one step knows materials of two volumes, boundary processes such as transition radiation or refraction could be simulated.
- G4SteppingManager class manages processing a step, a step is represented by G4Step, and G4StepPoint classes.
- G4UserSteppingAction is the optional user hook.
Step Status

- Status is attached to each G4StepPoint to show how step was determined
  - Use PostStepPoint to get status of current step
  - PreStepPoint has status of previous step

- Step status codes:
  - fWorldBoundary: step at edge of world volume
  - fGeomBoundary: step limited by a volume boundary other than the world
  - fAtRestDoItProc, fAlongStepDoItProc, fPostStepDoItProc: step is limited by one three types of process
  - fUserDefinedLimit: step limited by user
  - fUndefined: step not defined yet
Track does not keep its trace. No track object persists at the end of event.

- **G4Trajectory** is the class which copies some of G4Track information. **G4TrajectoryPoint** is the class which copies some of G4Step information.
  - G4Trajectory has a vector of G4TrajectoryPoint.
  - At the end of event processing, G4Event has a collection of G4Trajectory objects. 
    - /tracking/storeTrajectory must be set to 1.

- Keep in mind the distinction.
  - G4Track $\leftrightarrow$ G4Trajectory, G4Step $\leftrightarrow$ G4TrajectoryPoint

- Given G4Trajectory and G4TrajectoryPoint objects persist till the end of an event, you should be careful not to store too many trajectories.
  - E.g. avoid for high energy EM shower tracks.

- G4Trajectory and G4TrajectoryPoint store only the minimum information.
  - You can create your own trajectory / trajectory point classes to store information you need. 
    G4VTrajectory and G4VTrajectoryPoint are base classes.
A particle in Geant4 is represented by three layers of classes.

- **G4Track**
  - Position, geometrical information, etc.
  - This is a class representing a particle to be tracked.

- **G4DynamicParticle**
  - "Dynamic" physical properties of a particle, such as momentum, energy, spin, etc.
  - Each G4Track object has its own and unique G4DynamicParticle object.
  - This is a class representing an individual particle.

- **G4ParticleDefinition**
  - "Static" properties of a particle, such as charge, mass, life time, decay channels, etc.
  - G4ProcessManager which describes processes involving to the particle
  - All G4DynamicParticle objects of same kind of particle share the same G4ParticleDefinition.
Geant4 tracking is general.

- It is independent to
  - the particle type
  - the physics processes involving to a particle

- It gives the chance to all processes
  - To contribute to determining the step length
  - To contribute any possible changes in physical quantities of the track
  - To generate secondary particles
  - To suggest changes in the state of the track
    - e.g. to suspend, postpone or kill it.
Geant4 has six application states.

- **G4State_PreInit**
  - Material, Geometry, Particle and/or Physics Process need to be initialized/defined

- **G4State_Idle**
  - Ready to start a run

- **G4State_GeomClosed**
  - Geometry is optimized and ready to process an event

- **G4State_EventProc**
  - An event is processing

- **G4State_Quit**
  - (Normal) termination

- **G4State_Abort**
  - A fatal exception occurred and program is aborting
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 have to be physically meaningful
  - And the stepping invokes physics to realize them
- This physics is attached to the G4Track, let’s see how.

25-26 October, 2011, Geant4 Introduction
From **G4Track** to processes

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

- Momentum, pre-assigned decay…

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

- Holds pointers to processes

- Implementing physics processes

**G4Track**

**G4DynamicParticle**

**G4ParticleDefinition**

**G4ProcessManager**

**Process_1**

**Process_2**

**Process_3**

25-26 October, 2011, Geant4 Introduction
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
The main program

- Geant4 does not provide the `main()`.
- In your `main()`, you have to
  - Construct G4RunManager (or your derived class)
  - Set user mandatory classes to RunManager
    - G4VUserDetectorConstruction
    - G4VUserPhysicsList
    - G4VUserPrimaryGeneratorAction
- You can define VisManager, (G)UI session, optional user action classes, and/or your persistency manager in your `main()`.
Geant4 (User) Interface and Applications

**Analysis Tools**
- AIDA Interface
  - JAS3
  - ROOT
- Python binding
  - PAIDA
  - ROOT-Python

**GUI Tools**
- MOMO; Java-based tools
  - for editing Geometry/Physics List
- interactive session (GAG)
- OpenScientist; interactive environment

**Python Interface**
- C++ classes are directly bridged

**GUI**
- Qt
- Motif
- Java (GAG)

**Interfaces**
- Python as software component bus

**Pythonized Applications**
- Dynamic configuration of user apps
- GUIs / web apps

**User Applications (C++)**
- Pythonized Applications
  - Dynamic configuration of user apps
  - GUIs / web apps

**Terminal Front End**
- Batch
  - macro script
- simple command-line
  - tcsh-like shell

**GUI Tools**
- Python as software component bus

**Python Front End**
- >> import Geant4

**Intercoms**
- C++ classes are directly bridged
Practical Usage (G4UIExecutive)

- **G4UIExecutive** is available since 9.3 release.
  - Convenient class for selecting a UI session according to environment variables, G4UI_USE_XXX.
    - TCSH, XM, WIN32, QT, Ulterminal (default)
    - Select a session type by the order above

- In your main(),

  ```cpp
  #include "G4UIExecutive.hh"

  G4UIExecutive* ui = new G4UIExecutive(argc, argv);
  ui->SessionStart();
  delete ui;
  
  More practical implementation, see main() in n
  
  25-26 October, 2011, Geant4 Introduction
Batch Mode

- A Geant4 simulation can be executed in a batch mode.
  - A macro file consists of a series of UI commands
  - A macro file can be specified as an argument.
    \$ task2a myrun.mac >& myrun.log (csh)
    \# task2a myrun.mac > myrun.log 2>&1 (bash)

- To enable batch mode,
  - In your main(),
    G4UImanager* UI = G4UImanager::GetUIpointer();
    G4String command = "/control/execute ";
    G4String fileName = argv[1];
    UI->applyCommand(command+fileName);
LET US START EXERCISES

HTTP://GEANT4.IN2P3.FR/SPIP.PHP?RUBRIQUE6&LANG=EN
GEANT4
DOCUMENTATION

Slides are prepared by D.H. Wright (SLAC)
List of Main Documents and Tools

- **User Documents**
  - Application Developers' Guide
  - Installation Guide
  - Toolkit Developer Guide
  - Examples
  - Physics Reference Manual

- **User Aids**
  - Linux Crossed Reference (LXR) source code browser
  - HyperNews User Forum
  - Bug report system
Installation guide


- List of required software
  - C++ compiler, CLHEP, GNU make, Geant4 toolkit
  - choices for visualization software

- How to install on Linux

- Tips for installing on Windows
Joseph Perl’s (SLAC) installation guides

- Easier to use instructions
  http://geant4.slac.stanford.edu/installation
  - Installing Geant4 on Linux
  - Installing Geant4 on Mac
  - Installing Geant4 on Windows

- Tutorials for 3 most commonly used visualization systems
  http://geant4.slac.stanford.edu/Presentations/vis
  - Geant4 Visualization Tutorial using OpenGL
  - Geant4 Visualization Tutorial using HepRApp
  - Geant4 Visualization Tutorial using DAWN

25-26 October, 2011, Geant4 Introduction
Application Developer guide


- Introduces new users to the Geant4 toolkit

- Describes the most useful tools

- Describes how to set up and run a simulation application

- Intended as an overview of the toolkit, not an exhaustive treatment. For more details:
  - Physics Reference Manual: description of Physics models available in Geant4
  - Toolkit Developers Guide: in order to understand Geant4 more deeply

25-26 October, 2011, Geant4 Introduction
A description of the object-oriented design of the Geant4 toolkit
- class diagrams (some UML, some other)
- philosophy behind design choices

A guide for users who want to extend the functionality of Geant4
- adding new solids, modifying the navigator, creating new fields, etc.
A rich set of examples ready-to-run is delivered with Geant4 in $G4INSTALL/examples

Described in Users’ Guide for Application Developers

Ideal to learn how to use Geant4

3 categories
- Novice: basic functionalities of Geant4
- Extended: specific functionalities
- Advanced: full simulation of realistic use cases
A reference for toolkit users and developers who wish to consult the underlying physics of an interaction.

Presents the theoretical formulation, model or parameterization of the physics interactions provided by Geant4.
LXR code browser

- **URL:**
  http://www-geant4.kek.jp/LXR

- **Search entire Geant4 source tree by**
  - filename (e.g. G4Track.hh)
  - text
  - identifier

- **Results:** a source file fully hyper-linked to classes and methods
  - tells where classes and methods are defined
  - also where they are referenced

- **Recently added a doxygen version:**
  - http://www-geant4.kek.jp/Reference
HyperNews user forum

- URL: http://hypernews.slac.stanford.edu/HyperNews/geant4/cindex
- See also top of Geant4 home page
- Discuss problems with other users, post questions for experts, etc.
- 22 forums roughly based on Geant4 categories
- 4 forums for specific application areas (education, medicine, space, industry)
- New forums may be requested by users
- To join: click on “New member” at top of page and fill out form

25-26 October, 2011, Geant4 Introduction
Thank you for your attention!
Let us have a coffee