Hands-on Session on GEANT4
(Geometry and Tracking Software)

Kavita Lalwani
Assistant Professor
Department of Physics
MNIT Jaipur

Outline

Part I: Introduction (10 min)

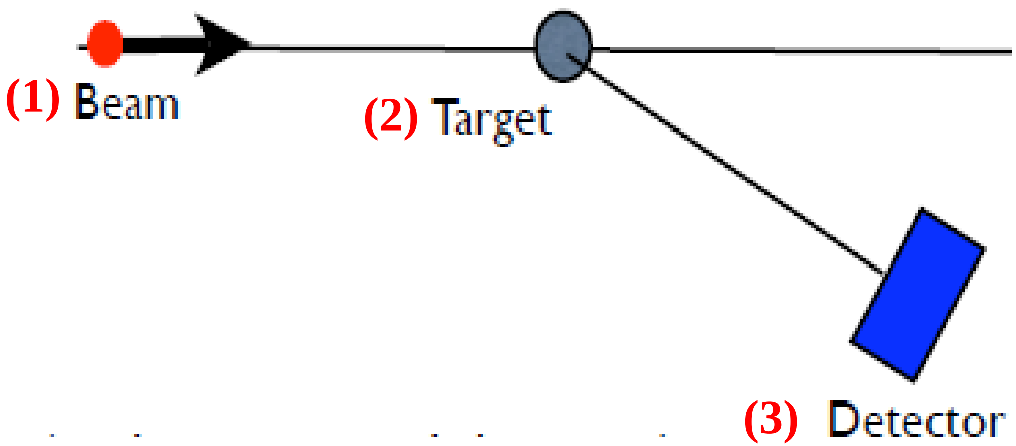
- Use of Detector Simulation
- Geant4 and its important features
- Method to construct a Detector Geometry
- Geant4 Class Structure
- Geant4 working flow chart
- Examples

Part II: Hands-on Session (50 min)

- Create Geometry: Box, sphere, tube, calorimeter, hodoscope (stack of scintillators), drift chamber
- How to rotate the Geometry
- How to generate beam of particles
- Store the deposited energy in Ntuples

Use of Detector Simulation

Experiment in Nuclear Physics/High Energy Physics



(4) E&M Fields
(5) Physics interactions
(in the target and detector)

- To design the experimental setup.
- To optimize the detector geometry.
- To study the expected background and radiation level.
- To generate large amounts of signal and background events for use in physics analysis once data comes to study signal/background separation.
- To derive directly calibrations, efficiencies, resolutions for high level objects in cases where data are biased or not available.

GEANT4 is *a toolkit for the simulation of the passage of particles through matter.*

- Used in nuclear physics, high energy physics, accelerator physics, medical as well as space applications.

- Using GEANT4 tool, user can build a simulation program for specific application.

- Developed from GEANT3 (Fortran). GEANT4 is written in C++ (Object-Oriented programming)

http://www.cern.ch/geant4 (Free Software)

Important Features of Geant4

- Geometry of the system
- Materials involved
- Particles
- Generation of events
- Tracking through matter and EM fields
- Response of detector components
- Generation of event data
- Storage of events and tracks
- Visualization
- Analysis of data
- Users may construct stand-alone applications

User Documentation

https://geant4.web.cern.ch/support/user_documentation

Introduction to Geant4

Installation Guide: For setting up Geant4 in your computing environment

User's Guide: For Application Developers
The first part of the document provides a step-by-step tutorial in the use of Geant4.
The second part describes the usage of the toolkit for practical applications.

User's Guide: For Toolkit Developers:
This document is for those who want to contribute to the extension of the functionality to the Geant4 toolkit - for example, to add a new physics process, to add a new particle, etc.

Units in Geant4

- Length : m/cm
- Time : ns
- Energy : MeV
- Angle : Radian
- Temperature : Kelvin
- All are defined inside
$CLHEP_BASE_DIR/Units/CLHEP/Units/SystemOfUnits.h

How to construct a Detector in GEANT4

Description of detector
Derive your own concrete class from
G4VUserDetectorConstruction abstract base class.

1) Construct all necessary materials
2) Define shapes/solids
3) Define logical volumes
4) Place volumes of your detector geometry
5) Associate (magnetic) field to geometry (optional)
6) Instantiate sensitive detectors / scorers and set them to corresponding logical volumes (optional)
7) Define visualization attributes for the detector elements (optional)
8) Define regions (optional)

Define Materials In GEANT4

Single Element

```
G4Element* hydrogen =
  new G4Element("Hydrogen", symbol="H", z=1, a=1.01*g/mole);

G4Element* oxygen =
  new G4Element("Oxygen", symbol="O", z=8, a=16.00*g/mole);
```

Assemble elements into molecule:

```
G4Material* H2O =
  new G4Material("Water", density=1.000*g/cm3,n_comps=2);

G4int number_of_atoms;
H2O->AddElement(hydrogen, number_of_atoms=2);
H2O->AddElement(oxygen, number_of_atoms=1);
```

Compounds are mixtures of elements not bound into molecules and can be specified

```
G4Element* nitrogen  =
  new G4Element("Nitrogen",symbol="N",z= 7., a=14.01*g/mole);
G4Element* oxygen  =
  new G4Element("Oxygen",symbol="O",z=8., a=16.00*g/mole);

G4Material* Air =
  new G4Material("Air",density=1.290*mg/cm3,ncomp=2);
  Air->AddElement(nitrogen, fracMass=70.0*perCent);
  Air->AddElement(oxygen, fracMass=30.0*perCent);
```

Class Structure in Geant4

User Classes

Mandatory

G4VUserDetectorConstruction
Define materials and geometries

G4VUserPhysicsList
Define processes and production threshold (s)

G4VUserPrimaryGeneratorAction
Generation of Primary event

Action Classes (Optional)

G4UserRunAction

G4UserEventAction

G4UserTrackingAction

G4UserSteppingAction

G4VUserDetectorConstruction class

(Define Materials and Geometries)

Three Volumes:

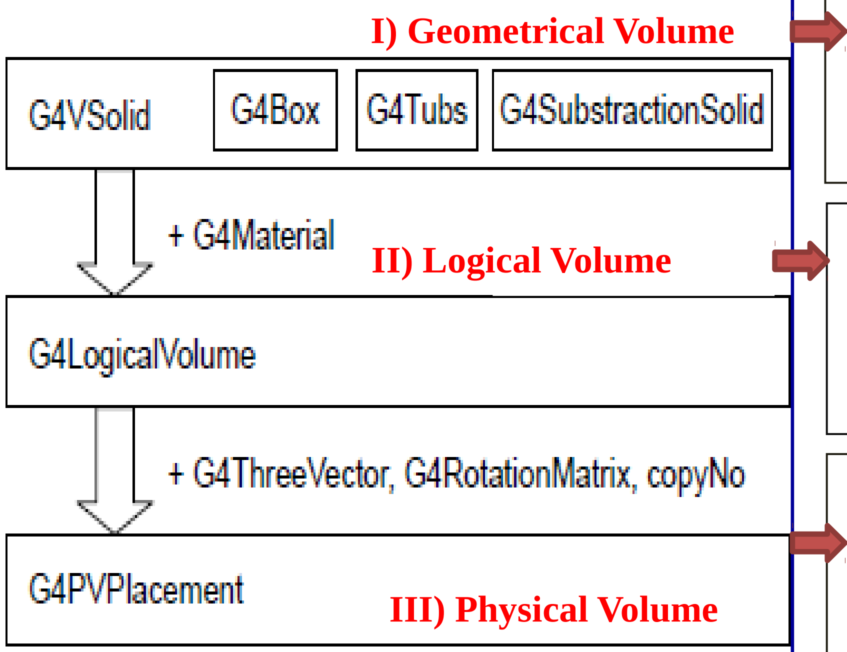
- **Geometrical (**G4VSolid) volumes specify shape and size.

- **Logical** (G4LogicalVolume) volumes specify materials.

- **Physical (**G4VPhysicalVolume) volumes specify placements in the setup.

G4VUserDetectorConstruction class

Flow Chart



Code

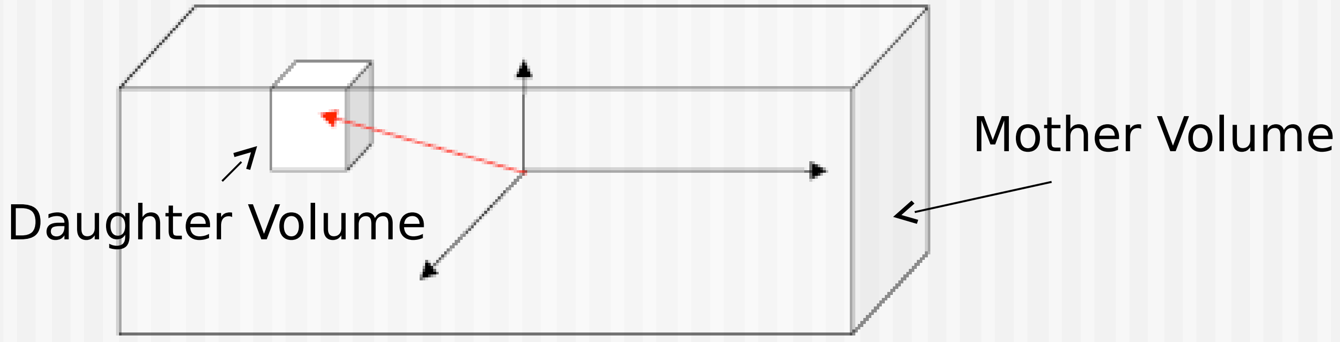
```
G4Box* solidWorld = new G4Box("World",world_sizeX,world_sizeY,world_sizeZ);
G4Box* solidTarget new G4Box("Target",target_sizeX, target_sizeY, target_sizeZ);
```

```
//Build lead and air materials
G4Material* Pb =  nist->FindOrBuildMaterial("G4_Pb");
G4Material* air = nist->FindOrBuildMaterial("G4_AIR");

//Fill the world with air. Create a lead target to fire particles at.
G4LogicalVolume* logicWorld = new G4LogicalVolume(solidWorld, air, "myWorld");
G4LogicalVolume* logicTarget = new G4LogicalVolume(solidTarget, Pb, "myTarget");
```

```
//Create world mother volume
G4VPhysicalVolume* physWorld = new G4PVPlacement(
                                        0,                  //no rotation
                                        G4ThreeVector(),    //at (0,0,0)
                                        logicWorld,         //its logical volume
                                        "World",            //its name
                                        0,                  //its mother  volume
                                        false,              //no boolean operation
                                        0,                  //copy number
                                        true);              //overlaps checking

//Place lead target in world volume
G4VPhysicalVolume* physTarget = new G4PVPlacement(
                                        0,                  //no rotation
                                        G4ThreeVector(),    //at (0,0,0)
                            logicTarget logicWorld,         //its logical volume
                                        "Target",           //its name
                                        logicWorld,         //its mother  volume
                                        false,              //no boolean operation
                                        0,                  //copy number
                                        true);              //overlaps checking
```

Placement
(Physical Volume)

- Mother and daughter volumes
  - A volume is placed in its mother volume
    - Position and rotation of the daughter volume is described with respect to the local coordinate system of the mother volume
    - The origin of the mother's local coordinate system is at the center of the mother volume
  - One or more volumes can be placed in a mother volume



G4VUserPrimaryGeneratorAction

(primary particle generation)

Set particle type, position and energy of beam

```
particleGun = new G4ParticleGun(n_particle);
particleGun->SetParticleDefinition(
        particleTable->FindParticle(particleName="gamma"));
particleGun->SetParticleEnergy(1173.2*keV);
particleGun->SetParticlePosition(G4ThreeVector(0.0, 0.0, 0.0));
```

G4VUserPhysicsList

✓Geant4 provides a wide variety of physics components for use in simulation.

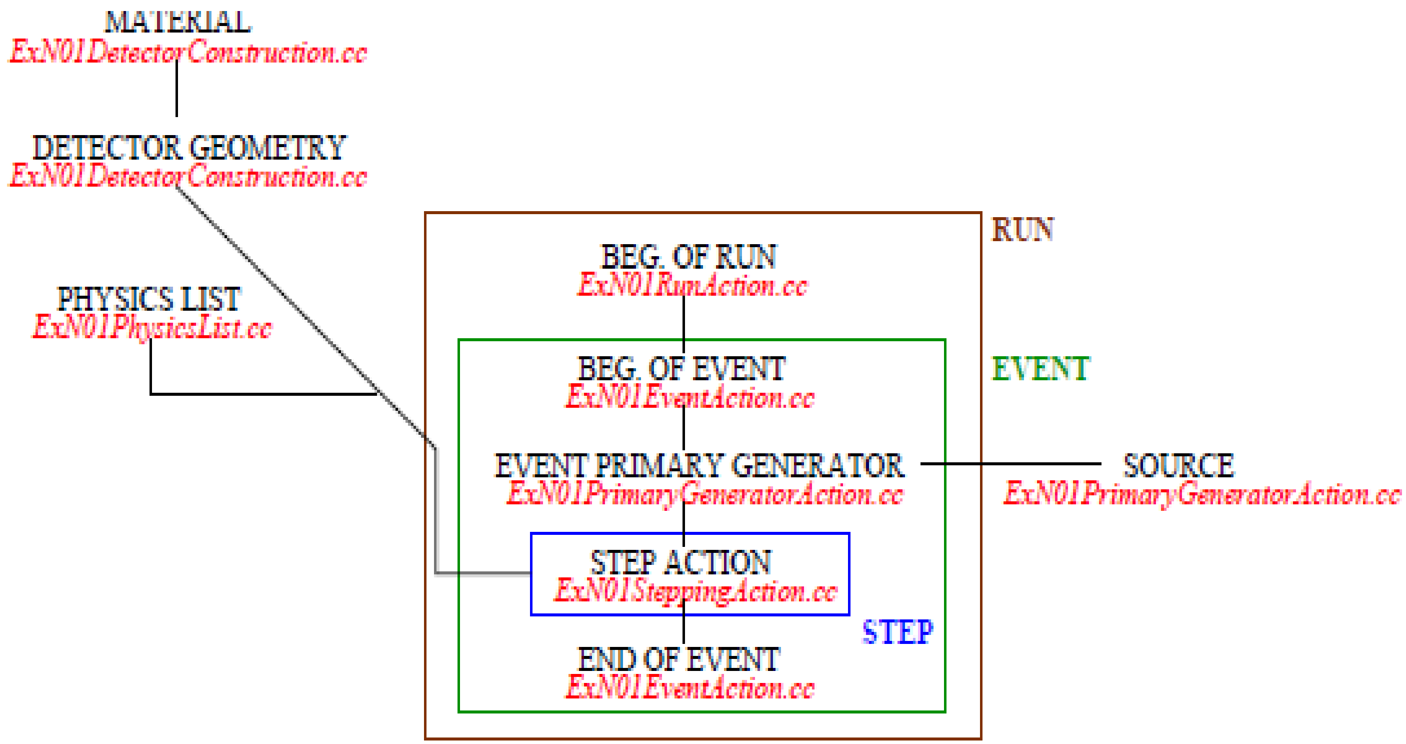
✓ Physics components are coded as processes.

- A process is a class which tells a particle how to interact with Detector material.
- User may write his own processes derived from Geant4 process.

✓Processes are grouped into

  - Electromagnetic
  - Hadronic
  - Decay

Flow Chart for GEANT4 Working



Run

- In Geant4, a run is a collection of events which share the same detector and physics conditions.
- A run consists of one event loop.
- At the beginning of a run, geometry is optimized and cross- section tables are calculated according to materials appear in the geometry and the cut-off values defined.
- A run starts with BeamOn() method of G4RunManager.

Event

- An event is the basic unit of simulation in Geant4.
- 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)

Track

- Track is a snapshot of a particle.
- G4TrackingManager manages processing a track, a track is represented by G4Track class.

Geant4 Examples (Three)

Basic Examples

- Simple Geometry with few solids
- Calorimeter
- Tracker etc

Extended Examples

- Demonstration of Geant4 specific usage
- Electromagnetic
- Event Generator Classes
- Analysis
- Medical Applications

Advanced Examples

- Simulation of real experimental set-up, for eg. Hadron Therapy; CMS Hadron Calorimeter; Medical applications LINAC, etc.

Visualization of Detector Geometry

Visualization Driver:

- **OpenGL**
- OpenInventor
- HepRep
- DAWN
- VRML
- RayTracer
- gMocren
- ASCIITree

What can be Visualized?

- Geometrical components
- Particle trajectories

Steps: How to Define a Detector Geometry

1. **Create a Simple Volume: Box**

To create a simple box, you need to define its name and its extent along each of the Cartesian axes.

```
G4double world_hx = 3.0*m;
G4double world_hy = 1.0*m;
G4double world_hz = 1.0*m;
G4Box* worldBox
= new G4Box("World", world_hx, world_hy, world_hz);
```

This creates a box named “World” with the extent from -1.0 meter to +1.0 meter along the Z axis, from -1.0 to 1.0 meters in Y, and from -3.0 to 3.0 meters in X.

Note that the G4Box constructor takes as arguments the halves of the total box size.

Steps: How to Define a Detector Geometry (Continue..)

2. Create Logical Volume

To create a logical volume, you must start with a solid and a material. So, using the box created above, you can create a simple logical volume filled with argon gas

```
G4LogicalVolume* worldLog= new G4LogicalVolume(worldBox, Ar, "World");
```

Its solid **Material** **Name**

This logical volume is named “World”.

Note: How to Define Material (Will Discuss Next after creating the Box)

Steps: How to Define a Detector Geometry (continue..)

3. **Place a Volume**

You create a physical volume starting with your logical volume.
A physical volume is simply a placed instance of the logical volume.
This instance must be placed inside a mother logical volume.

```
G4VPhysicalVolume* WorldPV
  = new G4PVPlacement(
          0,              // no rotation
          G4ThreeVector(),  // at (0,0,0)
          WorldLog,         // its logical volume
          "World",          // its name
          0,              // its mother  volume
          false,            // no boolean operation
          0,              // copy number
          fCheckOverlaps);  // checking overlaps
```

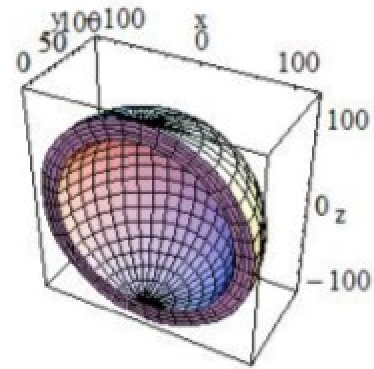
Steps: How to Define a Detector Geometry (Tube)

```
G4double innerRadius = 0.*cm;
G4double outerRadius = 60.*cm;
G4double hz = 25.*cm;
G4double startAngle = 0.*deg;
G4double spanningAngle = 360.*deg;

G4Tubs* trackerTube = new G4Tubs("Tracker",
                                 InnerRadius,
                                 outerRadius,
                                 hz,
                                 StartAngle,
                                  spanningAngle);
```

This creates a full cylinder, named “Tracker”, of radius 60 centimeters and length 50 cm (the hz parameter represents the half length in Z).

How to Create a Sphere ?

```
G4Sphere(const G4String& pName,
             G4double   pRmin,
             G4double   pRmax,
             G4double   pSPhi,
             G4double   pDPhi,
             G4double   pSTheta,
             G4double   pDTheta )
```



In the picture:
`pRmin = 100, pRmax = 120, pSPhi = 0*Degree, pDPhi = 180*Degree, pSTheta = 0 Degree, pDTheta = 180*Degree`

to obtain a solid with name `pName` and parameters:

`pRmin`	Inner radius
`pRmax`	Outer radius
`pSPhi`	Starting Phi angle of the segment in radians
`pDPhi`	Delta Phi angle of the segment in radians
`pSTheta`	Starting Theta angle of the segment in radians
`pDTheta`	Delta Theta angle of the segment in radians

Hands-on session