Introduction to Geant4
(Geometry and Tracking Software)

Kavita Lalwani
Assistant Professor
Department of Physics
MNIT Jaipur, India

Objective & Prerequisites

Objective of this session is to first introduce you about the Geant4 simulations and then give you hands on training for the same.

Prerequisites:

Basic knowledge of:

- **Detectors** used in Nuclear Physics & High Energy Physics (HEP) experiments.
- **Different particles radiations and their interaction** with detector material.
- **Basics of C++**

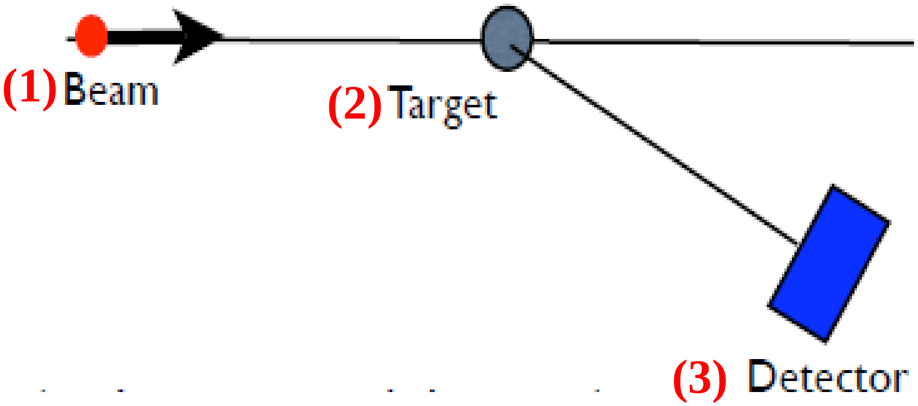
Book: Techniques for Nuclear and Particle Physics Experiments by William R. Leo

Plan for Geant4 Tutorial Based Class

Sr. No.	Plan	Outline
Part I	Lecture (30 min)	• Introduction to Geant4 • Installation • Class Structure • How to build a very simple application in Geant4, which includes: Detector construction Physics processes Particle generation
Part II	Hands on session (30 min)	• Basic Examples • Exercises

Why do we need 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 compare the experimental results in order to make sure there is no error in analysis.

What is GEANT4 ?

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)

Developed and maintained by GEANT4 Collaboration

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

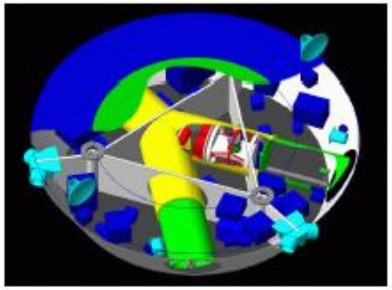
Geant 4

Geant4 is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The two main reference papers for Geant4 are published in *Nuclear Instruments and Methods in Physics Research* A 506 (2003) 250-303, and *IEEE Transactions on Nuclear Science* 53 No. 1 (2006) 270-278.

Applications

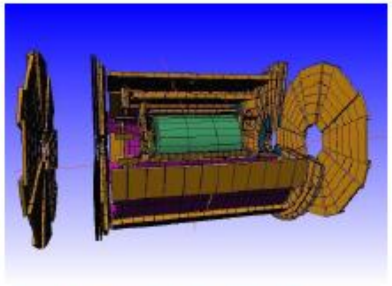
A sampling of applications, technology transfer and other uses of Geant4

User Support



Getting started, guides and information for users and developers

Publications



Validation of Geant4, results from experiments and publications

Collaboration

Who we are: collaborating institutions, members, organization and legal information

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

Installation of Geant4

Perquisites for Geant4

Which Operating System?

We need a Desktop/Laptop having OS

- Linux

- Windows

Prerequisite Libraries for Geant4: g++-4.9, Make, **Cmake-3.3 or higher,** Xerces-c++, Opengl, Qt4.8, motif UI, x11, libxmu-dev, libxmu-header

How to Install Geant4?

Two Source Files: **CLHEP** and **Geant4** **(see installation report)**

A) CLHEP is a C++ library that provides utility classes for numerical programming, random number generation, and linear algebra, specifically targeted for high energy physics simulation and analysis software.

latest source file: [clhep-2.3.3.2.tgz]()

http://proj-clhep.web.cern.ch/proj-clhep/clhep23.html

- Create directory CLHEP
- Create directory build inside CLHEP
- Inside build, use command

Where we want to build G4lib

cmake -DCMAKE_INSTALL_PREFIX=/home/user/CLHEP/2.3.3.2/CLHEP- install /home/user/CLHEP/2.3.3.2/CLHEP

$ make -jN

$ make test

$ make install

Source directory: contains the CMakeLists.txt file.

where N is the number of parallel jobs you require

- This command will generate CLHEP-install and following three directories inside "CLHEP-install": **Bin, Lib, Include**

- Set environment in .bashrc file

Set environment variables in .bashrc

export CLHEP_DIR=/home/user/CLHEP/2.3.3.2/CLHEP-install/

export CLHEP_INCLUDE_DIR=${CLHEP_DIR}/include/

export CLHEP_LIBRARY=${CLHEP_DIR}/lib/

ExportLD_LIBRARY_PATH=${CLHEP_LIBRARY}:${LD_LIBRARY_PATH}

export PATH=$CLHEP_DIR/bin/:$PATH

Tested on Ubuntu 14.04 & Ubuntu 16

How to Install Geant4?

B) Geant4

latest source file: geant4.10.02.p02 tar file

https://geant4.web.cern.ch/geant4/support/download.shtml

Create a directory build inside Geant4

Inside build, use command

This is the method of building Geant4 using cmake

1) $ cmake\ -DCMAKE_INSTALL_PREFIX=/home/user/Geant4/geant4.10.02.p02-install \ -DGEANT4_USE_QT=ON -DGEANT4_USE_OPENGL_X11=ON \ -DGEANT4_USE_RAYTRACER_X11=ON -DGEANT4_USE_GDML=ON \ -DGEANT4_USE_SYSTEM_CLHEP=ON \ -DCLHEP_INCLUDE_DIR=/home/user/CLHEP/2.3.3.2/CLHEP- \ install/include/ -DCLHEP_LIBRARY=/home/user/CLHEP/2.3.3.2/CLHEP-\ install/lib/libCLHEP.so -DGEANT4_INSTALL_EXAMPLES=ON \ /home/user/Geant4/geant4.10.02.p02

Where we want to build G4lib

Source directory: contains the CMakeLists.txt file.

---

2) $ make –jN (2)
$ make test

CMake generated Makefiles support parallel builds, so can set N suitable for the number of cores on your machine (e.g. on a dual core processor, you could set N to 2).

---

3) $ make install
This command will create directory Geant4-install.

---

4) Set environment variables in .bashrc
source /home/user/Geant4/geant4.10.02.p02-install/bin/geant4.sh

Class Structure in Geant4

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

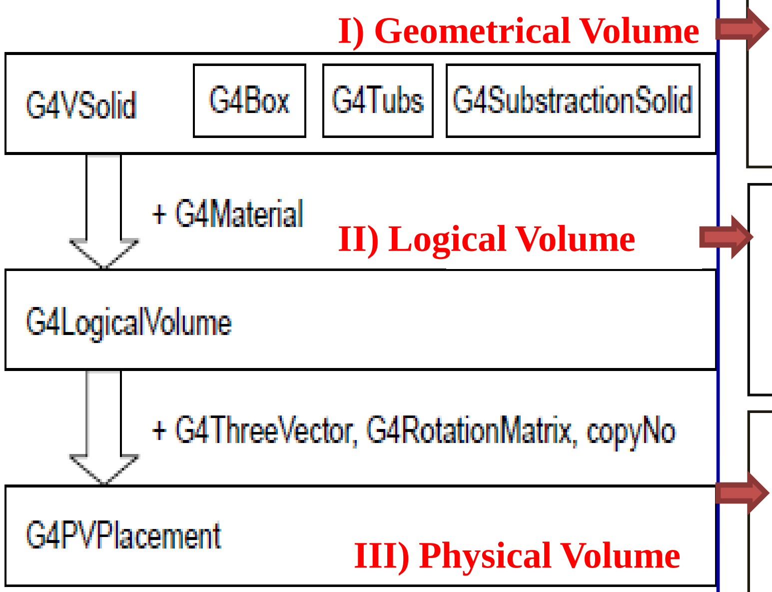
(Define Materials and Geometries)

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

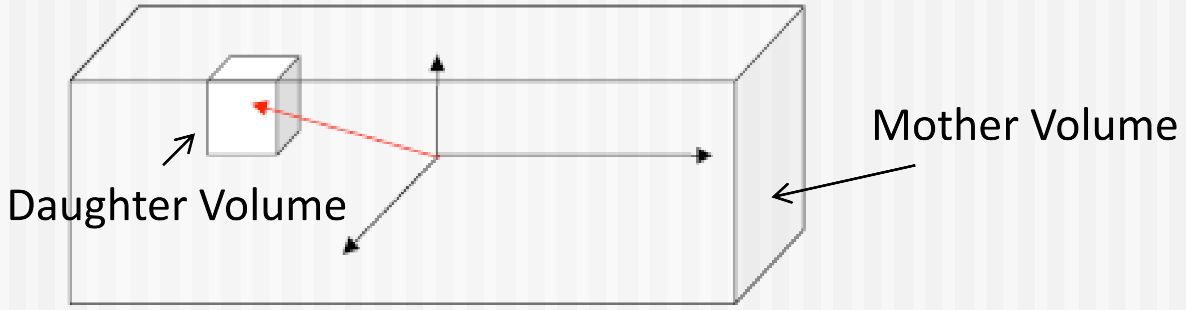
```
G4Box* solidWorld = new G4Box("World",world_sizeX,world_sizeY,world_sizeZ);
G4Box* solidTarget new G4Box("Target",target_sizeX, target_sizeY, target_sizeZ);
```

```
//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
    - Daughter volumes cannot protrude from the mother volume
    - Daughter volumes cannot overlap
  - 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

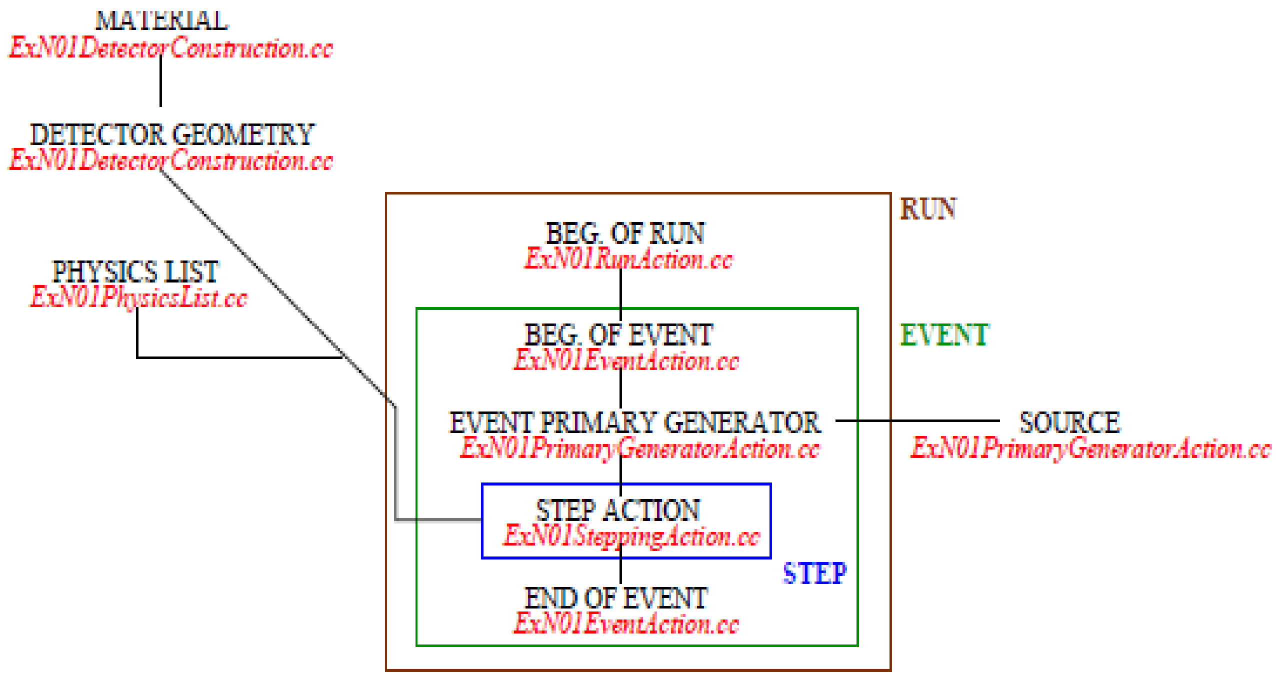
- Electromagnetic
- Hadronic
- Decay

Q1. How photon interact with detector material?

Example: Electromagnetic Processes Registration for Photon

```
void MyPhysicsList::ConstructProcess()
{
  // Define transportation process
  AddTransportation();
  // electromagnetic processes
  ConstructEM();
}
void MyPhysicsList::ConstructEM()
{
   //  Get the process manager for gamma
  G4ParticleDefinition* particle = G4Gamma::GammaDefinition();
  G4ProcessManager* pmanager = particle->GetProcessManager();

  // Construct processes for gamma
  G4PhotoElectricEffect * thePhotoElectricEffect = new G4PhotoElectricEffect();
  G4ComptonScattering * theComptonScattering = new G4ComptonScattering();


  // Register processes to gamma's process manager
  pmanager->AddDiscreteProcess(thePhotoElectricEffect);
  pmanager->AddDiscreteProcess(theComptonScattering);

}
```

Summary of Different Part of the User Code



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

Advanced Examples

- Simulation of real experimental set-up, for eg. Hadron Therapy

Basic Examples

- Example B1
  - Simple geometry with a few solids
  - Geometry with simple placements (G4PVPlacement)
  - Scoring total dose in a selected volume user action classes
  - Geant4 physics list (QBBC)
- Example B2
  - Simplified tracker geometry with global constant magnetic field
  - Geometry with simple placements (G4PVPlacement) and parameterisation (G4PVParameterisation)
  - Scoring within tracker via G4 sensitive detector and hits
  - Geant4 physics list (FTFP_BERT) with step limiter
  - Started from novice/N02 example
- Example B3
  - Schematic Positron Emitted Tomography system
  - Geometry with simple placements with rotation (G4PVPlacement)
  - Radioactive source
  - Scoring within Crystals via G4 scorers
  - Modular physics list built via builders provided in Geant4
- Example B4
  - Simplified calorimeter with layers of two materials
  - Geometry with replica (G4PVReplica)
  - Scoring within layers in four ways: via user actions (a), via user own object (b), via G4 sensitive detector and hits (c) and via scorers (d)
  - Geant4 physics list (FTFP_BERT)
  - Histograms (1D) and ntuple saved in the output file

Visualization of Detector Geometry

Visualization Driver:

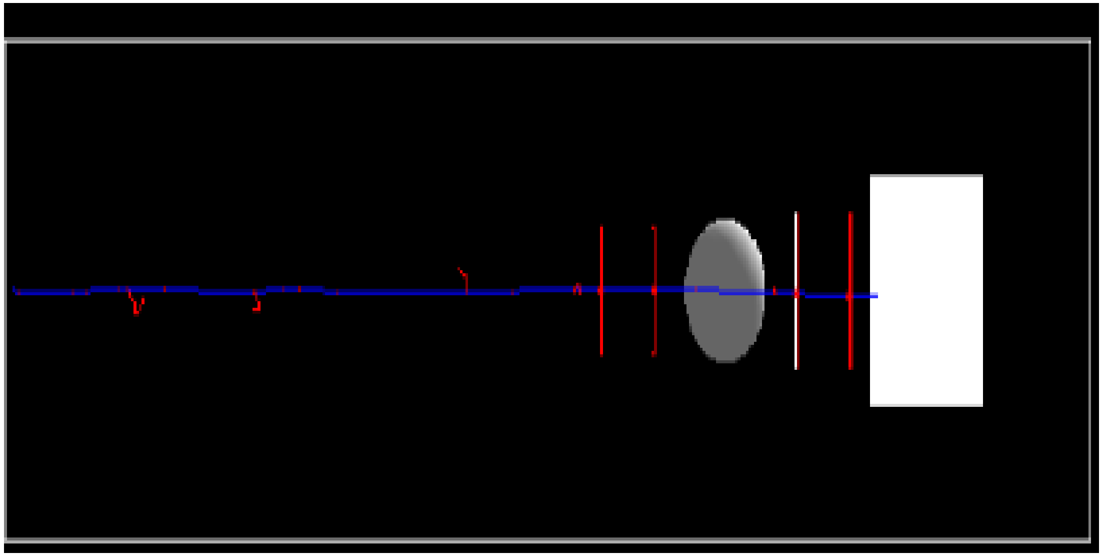
- **OpenGL**
- OpenInventor
- HepRep
- DAWN
- VRML
- RayTracer
- gMocren
- ASCIITree

What can be Visualized?

- Geometrical components
- Particle trajectories

Design and Development of Proton Computed Tomography (pCT) Set up using Geant4

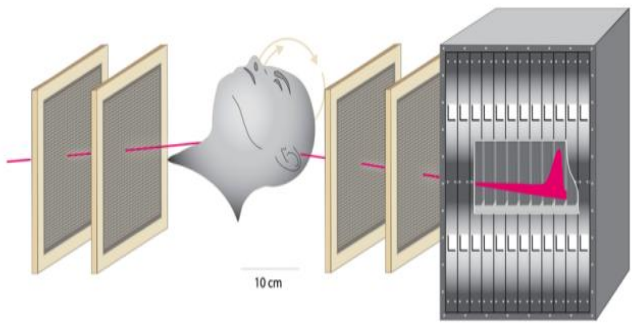
In collaboration with Northern Illinois University (NIU), Fermi lab, USA and Delhi university, India.



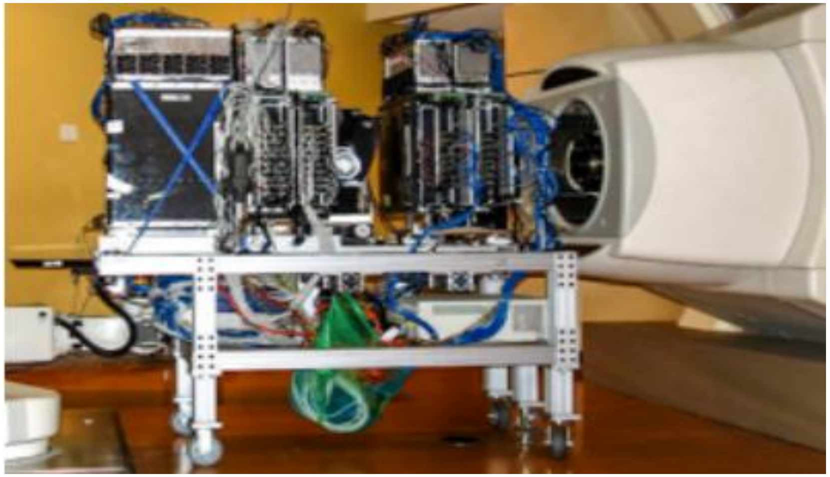
Design and Development of pCT Set up using Geant4

In collaboration with Northern Illinois University, Fermi lab, USA and Delhi University, India.

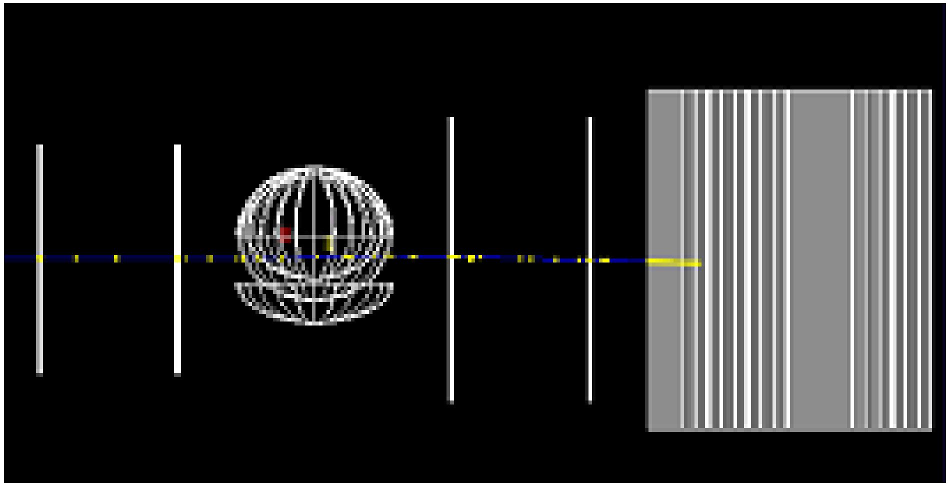
- Participation in the design optimization of Proton Computed Tomography (pCT) set-up.
- The facility is established at NIU.



Sketch for pCT set-up



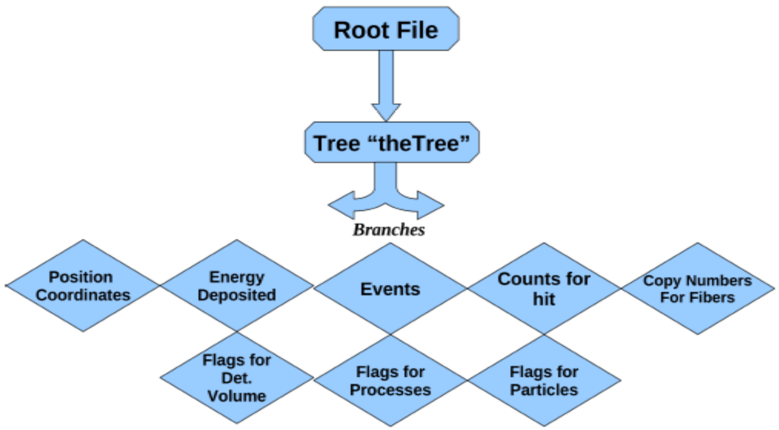
Full pCT detector with test beam



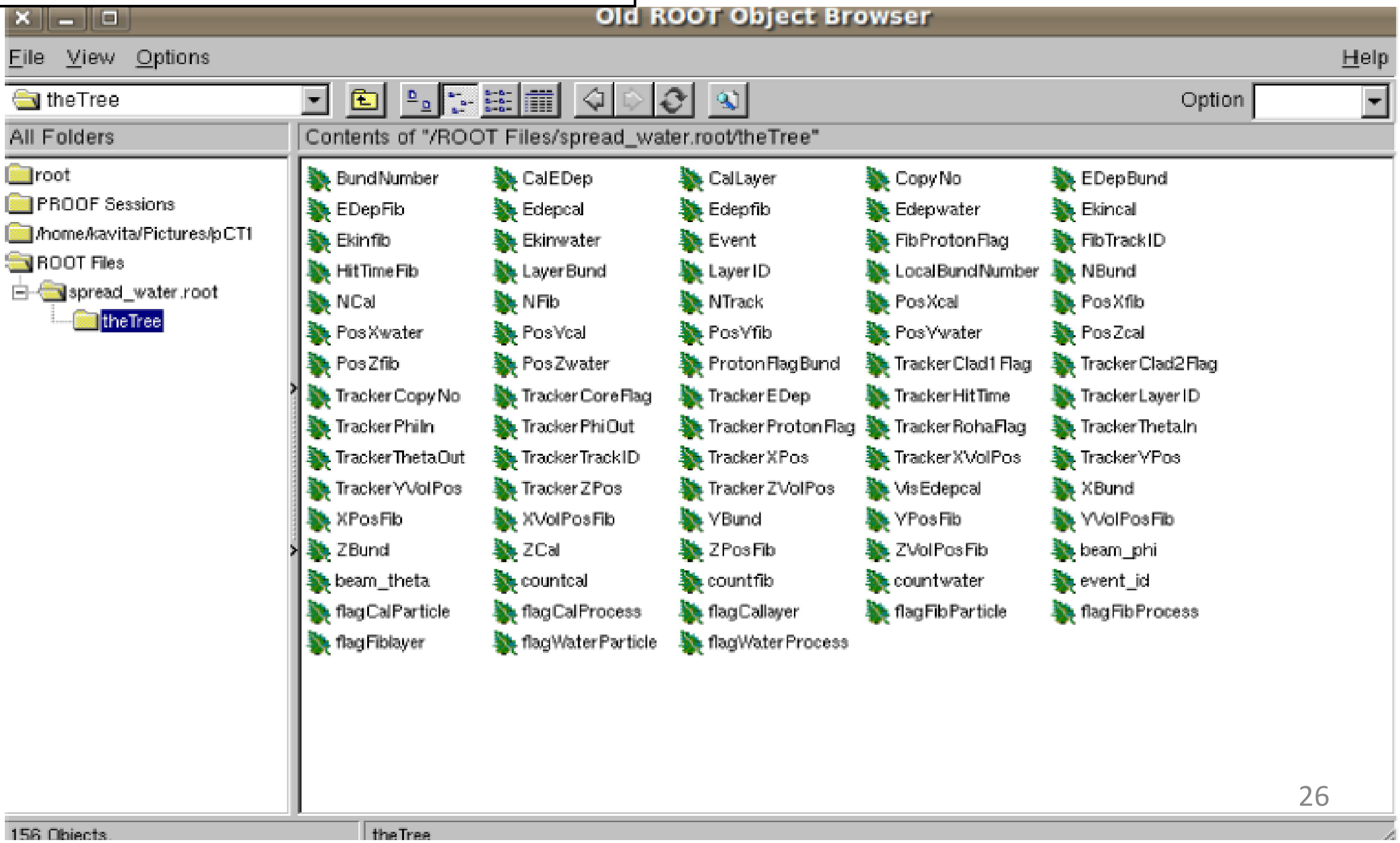
pCT in Geant4 Monte Carlo simulations

1. Development of a proton Computed Tomography (pCT) scanner at NIU , S. A. Uzunyan, K. Lalwani et al..,Proceedings of the “New Trends in High Energy Physics” Conference, Alushta, Crimea, September 23-29, 2013.
2. Development of Proton Computed Tomography (pCT) Detector for Cancer Therapy, Kavita Lalwani for the pCT collaboration ,Proceedings of the DAE Symp. on Nucl. Phys. 57 (2012)

Analysis using ROOT

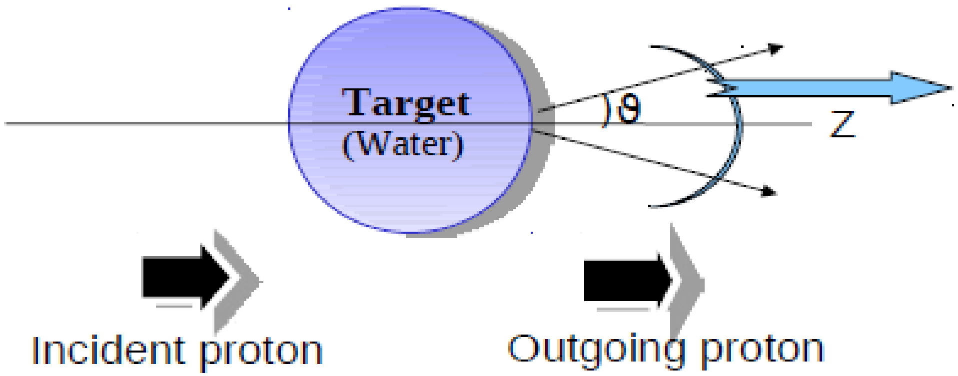


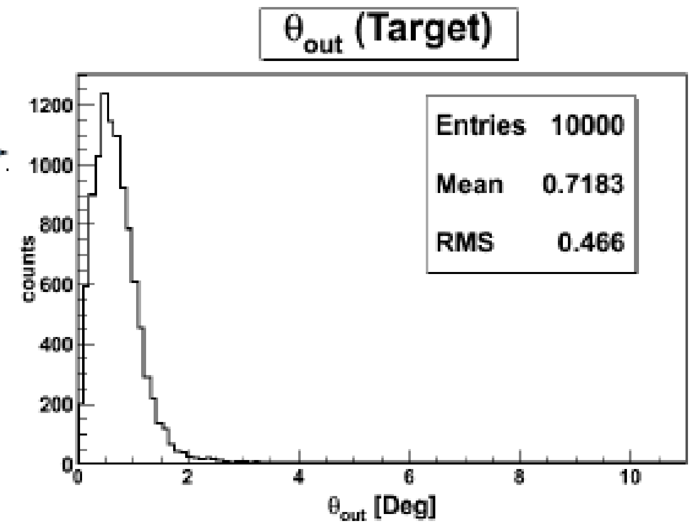
Using Analysis Manager Class

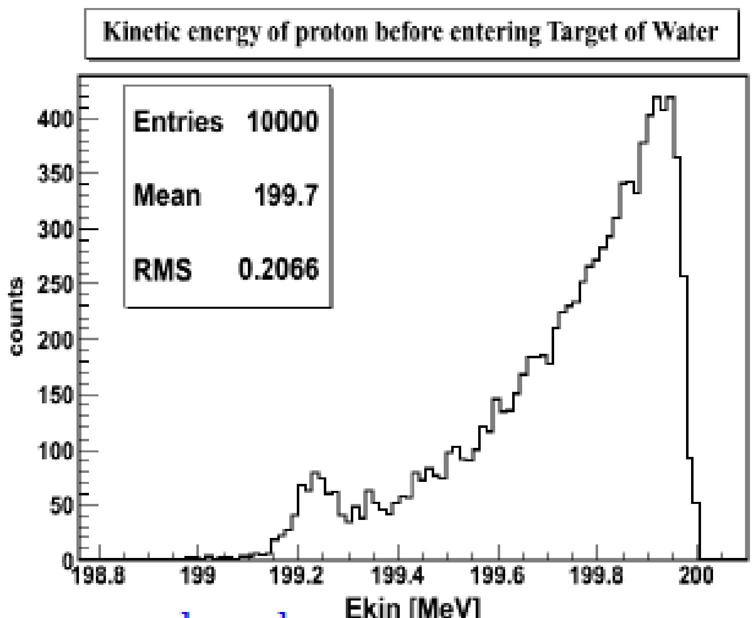


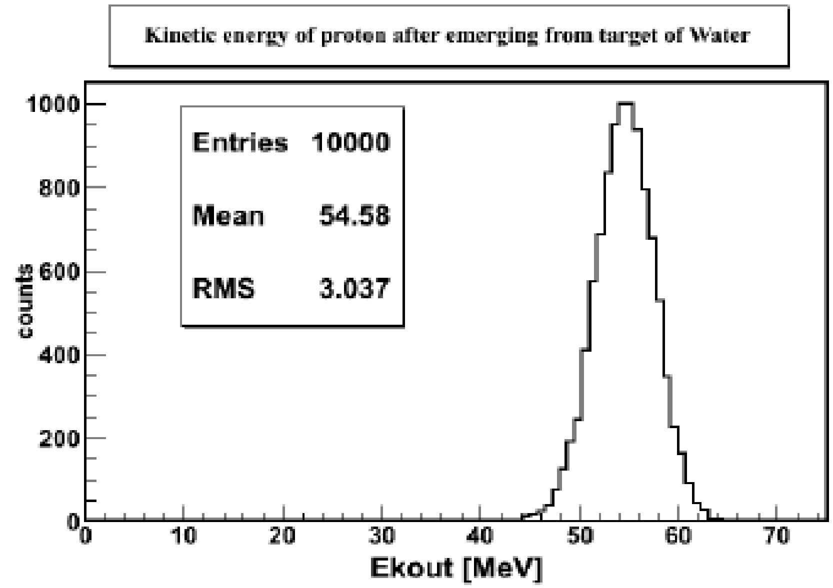
Incident and outgoing variables*

Events generated: 10k



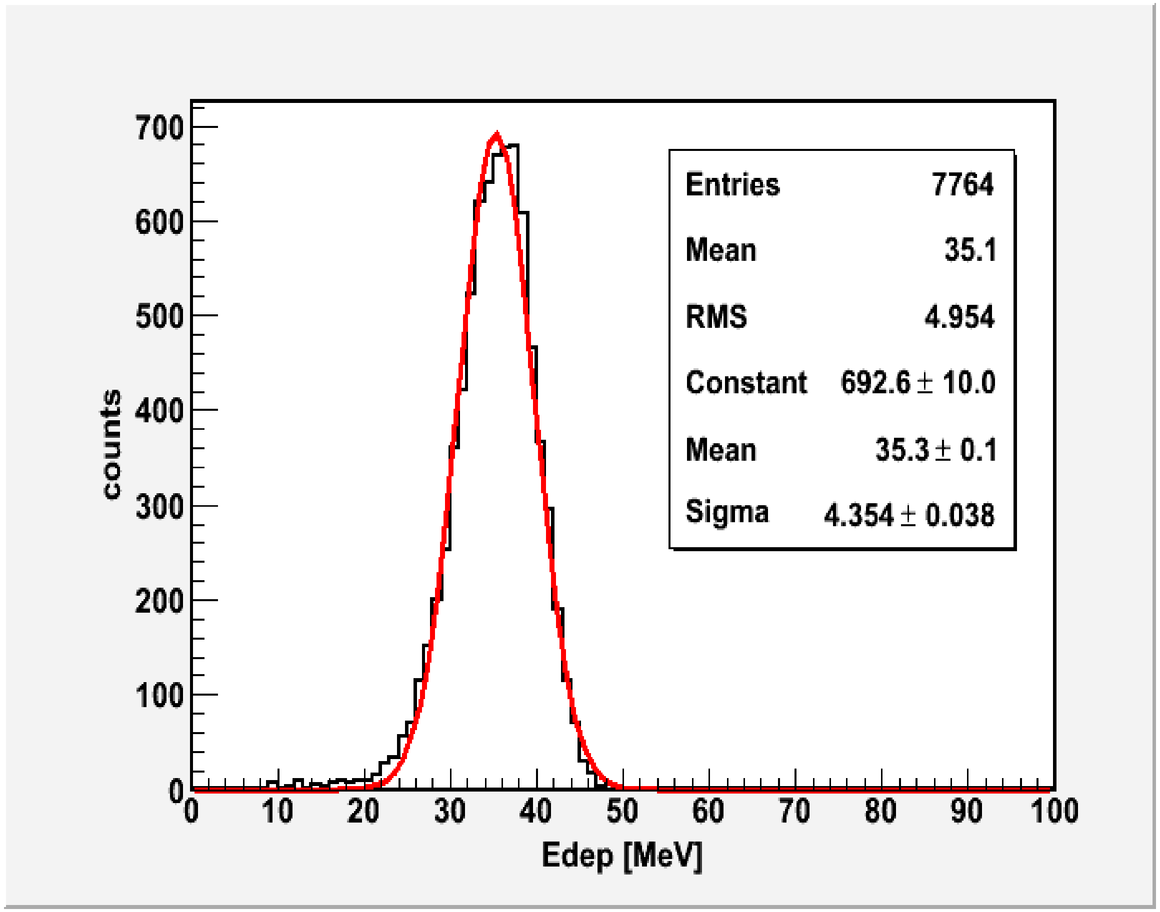






*Energy and angle

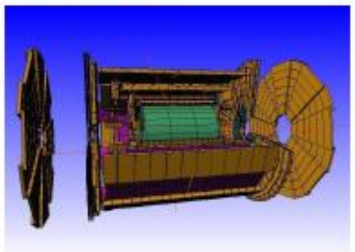
Deposited Energy in Calorimeter



Geant 4

Applications

High Energy Physics

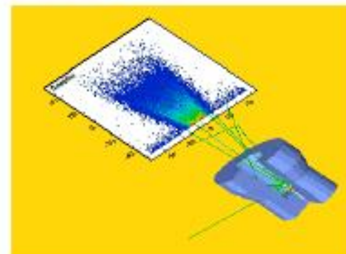


>>more high energy physics

Space and Radiation

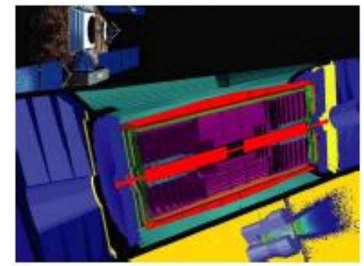
>>more space and radiation

Medical



>>more medical

Technology Transfer



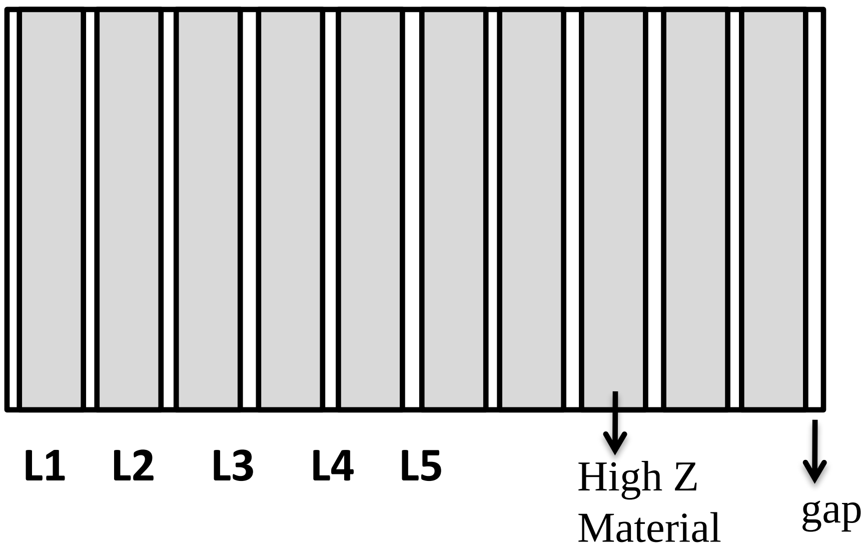
>>more technology transfer

High-Energy Physics experiments: detector design & tracking
Cosmic rays
Neutrino physics
Dosimetry
Radiotherapy
Biological damage studies
Assessment of radiation damage to the electronics of satellites
Study of the radiation environment of planets

Part II: Hands-on Session (30 min)

Exercise I) Construct Calorimeter Geometry using G4VDetectorConstructionClass

- The calorimeter is a box made up of a given number of layers.
- A layer consists of an absorber plate and a gap.
- The layer is replicated.



Layer (LX) consists of Absorber + Gap

Exercise II) Generate beam of proton (pencil & cone shape) and incident it on Calorimeter you designed in Ex. I using G4VPrimaryGeneration Class

Exercise III) Energy Deposited in Absorber (Analysis via ROOT)

Total energy in calorimeter?

Parameters for Exercise I & II

Beam (Proton)
Pencil beam
kinetic energy: 200 MeV

Calorimeter

Scintillator Tiles (Polystyrene ~ $[C_6H_5CHCH_2]_n$
Area: 26 cm x 26 cm
Numbers: 100
Density: 1.032 g/cm3
Width: 0.3cm

Two layers of Tyvek
Width: 0.1mm
Each tile will have Tyvek (Polyethylene ~ [CH2]n , density = 0.920 g/cm3) on its front and back face

Can you do exercise I & II ?

Ex. I) Design detector geometry of calorimeter

- Look for the existing examples: Basics, extended, advanced ?
- Go back to the Slide no?

- **How to Build and Run the Example (B4):**

Step 1: Make g4work dir
Step 2: cd g4work
Step 3: mkdir build_B4
Step 4 cd build_B4
Step 5: Copy example directory "B4" from /home/kavita/Final_Soft/Geant4/geant4.10.02.p02-install/share/Geant4-10.2.2/examples/basics/B4 to "g4work/build_B4" directory
Step 6: Now inside build_B4 you have B4 directory
Step 7: inside build_B4, use **command 1**
$ cmake -DGeant4_DIR=/Final_Soft/Geant4/geant4.10.02.p02-install/lib/Geant4-10.2.0/ /home/kavita/g4work/build_B4/B4 (Source dir)

You will see the below message on terminal (screen1)

Screen 1

kavita@kavita:~/g4work/build_B4$ cmake -DGeant4_DIR=/Final_Soft/Geant4/geant4.10.02.p02-install/lib/Geant4-10.2.0/ /home/kavita/g4work/build_B4/B4

Here, the first argument points CMake to our install of Geant4. Specifically, it is the directory holding the Geant4Config.cmake file that Geant4 installs to help CMake find and use Geant4.

The second argument is the path to the *source directory* of the application we want to build. Here it's just the B4 directory.

```
-- The C compiler identification is GNU 4.8.4
-- The CXX compiler identification is GNU 4.8.4
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found CLHEP Version 2.3.3.2
-- Found CLHEP: /lib/libCLHEP.so (Required is at least version "2.3.3.2")
-- Configuring done
-- Generating done
-- Build files have been written to: /home/kavita/g4work/build_B4
```

Command 2

$ cmake -j 6 B4

You will see below message on terminal (screen 2)

Screen 2

-- Configuring done

-- Generating done

-- Build files have been written to:

/home/kavita/g4work/build_B4

Command 3

$ make install

you will see the message (below) on screen 3

```
Screen 3
[ 87%] Building CXX object B4d/CMakeFiles/exampleB4d.dir/src/B4RunAction.cc.o
[ 90%] Building CXX object B4d/CMakeFiles/exampleB4d.dir/src/B4dActionInitialization.cc.o
[ 93%] Building CXX object B4d/CMakeFiles/exampleB4d.dir/src/B4dDetectorConstruction.cc.o
[ 96%] Building CXX object B4d/CMakeFiles/exampleB4d.dir/src/B4dEventAction.cc.o
[100%] Linking CXX executable exampleB4d
[100%] Built target exampleB4d
Install the project...
-- Install configuration: ""
-- Installing: /usr/local/bin/exampleB4a
```

Visualization of Geometry (OpenGL)

(Without beamOn)

This will generate an executable (green colour) inside build_B4/B4a/**exampleB4a**
cd B4/B4a
run **command 4**
./example4a

Calorimeter layers (gap and absorber) = 10

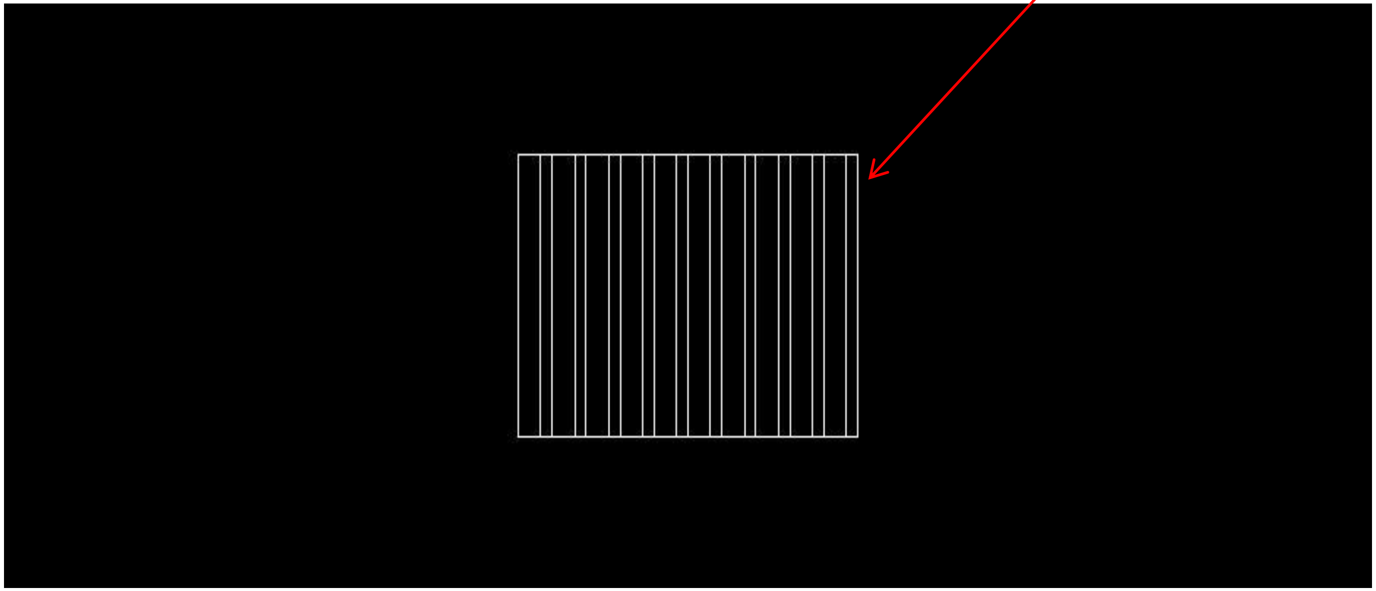


Figure: Calorimeter without beamOn

You can save it by right clicking on and save as (give file name.jpg)

Visualization of Geometry (Open GL)

(With beamOn)

Inside 4a directory, you see vis.mac file
and add line in vis.mac file:
/run/beamOn 10
Here “10” is the number of events
./example4B

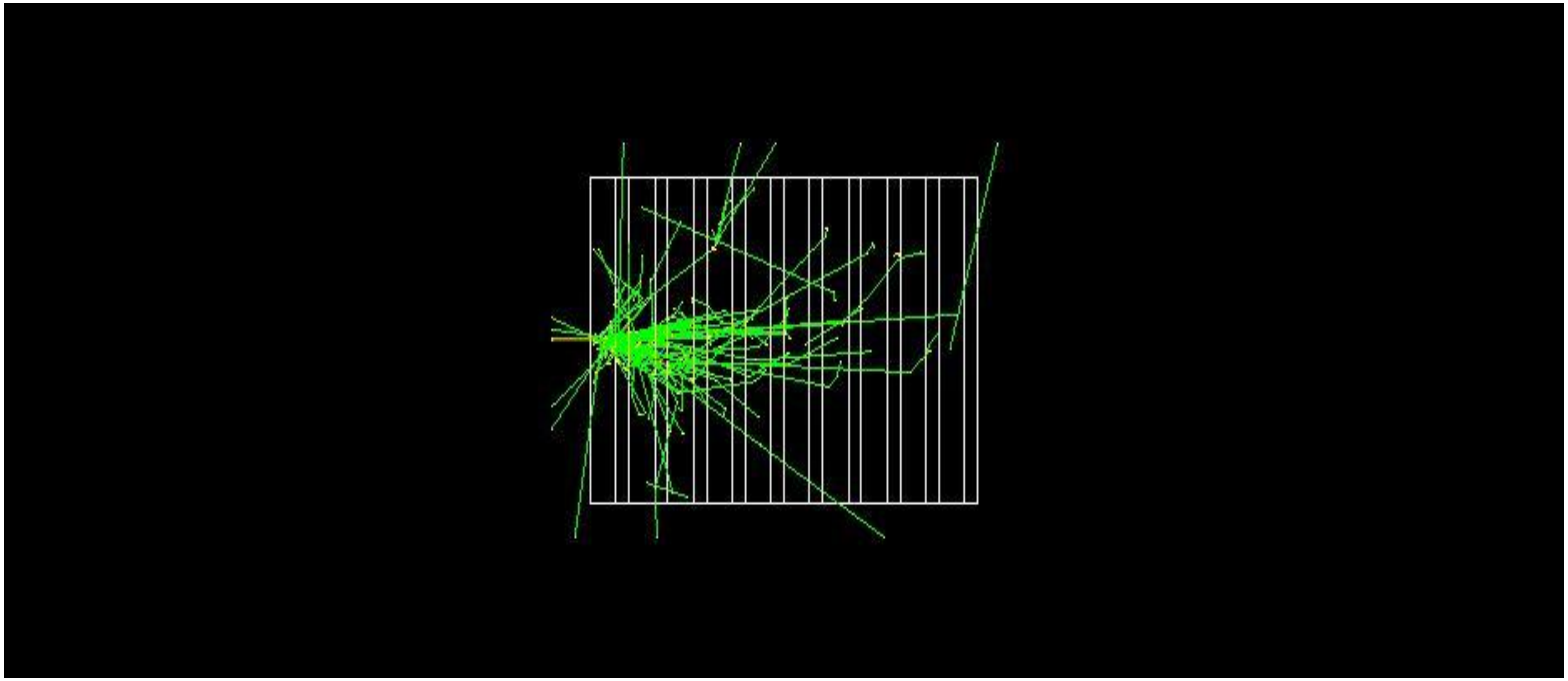


Figure: Calorimeter with beamOn

Analysis using ROOT (Analysis Manager Class)

Section II) How to run the simulation with external macro file (run1.mac) inside g4work/build_B4/B4

use command

~/g4work/build_B4/B4a$ **./exampleB4a -m run1.mac**

```
**************************************************************
 Geant4 version Name: geant4-10-02-patch-02    (17-June-2016)
                      Copyright : Geant4 Collaboration
                      Reference : NIM A 506 (2003), 250-303
                            WWW : http://cern.ch/geant4
**************************************************************
<<< Geant4 Physics List simulation engine: FTFP_BERT 2.0
Using Root
Visualization Manager instantiating with verbosity "warnings (3)"...
Visualization Manager initialising...
Registering graphics systems...
You have successfully registered the following graphics systems.
Current available graphics systems are:
ASCIITree (ATree)
DAWNFILE (DAWNFILE)
G4HepRep (HepRepXML)
G4HepRepFile (HepRepFile)
RayTracer (RayTracer)
=================================================================
### Run 0 starts.
... open Root analysis file : B4.root - done
--> Event 0 starts
```

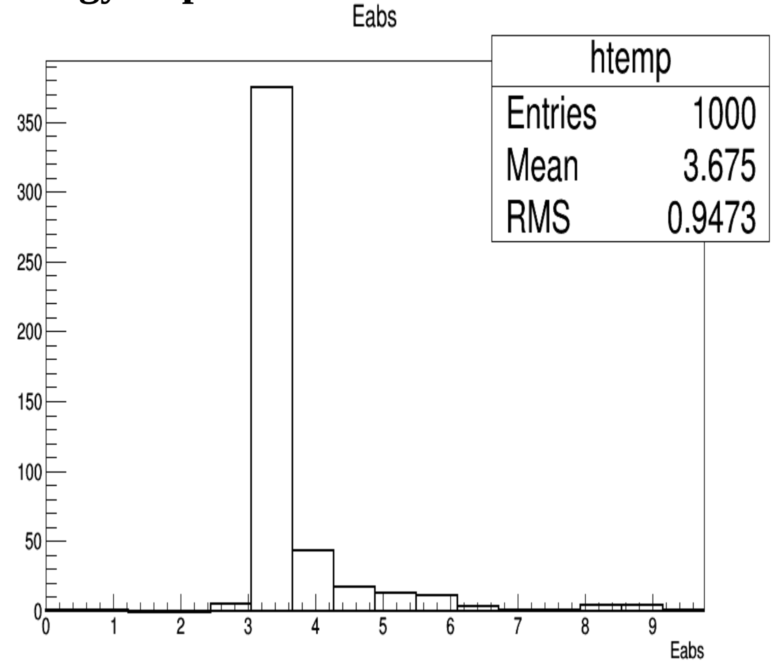
Continue....

```
... open Root analysis file : B4.root - done
--> Event 0 starts
*********************************************************************************************************
* G4Track Information:   Particle = e-,   Track ID = 1,   Parent ID = 0
*********************************************************************************************************
Step#    X(mm)    Y(mm)    Z(mm) KinE(MeV)  dE(MeV) StepLeng TrackLeng  NextVolume ProcName
    0        0        0      -90        50        0        0         0       World initStep
    1        0        0      -75        50     50.1       15        15        Abso    Transportation
    2   0.0728  -0.0411      -74      46.1      1.2    0.974        16  Abso    eBrem
    3    0.157   0.0469    -73.8      45.7    0.282    0.244      16.2   Abso eBrem
    4    0.163    0.061    -73.8      17.6   0.0678    0.028      16.2     Abso eBrem
    5    0.174   0.0925    -73.8      11.6   0.0495   0.0562      16.3     Abso eBrem
    6    0.661     1.32    -73.1      9.42     1.98     1.64      17.9     Abso eBrem
    7   0.0235    0.995    -74.3      6.74     2.24     1.85      19.8     Abso eBrem
    8    0.108     1.26      -75      5.44      1.3     1.12      20.9    World Transportation
    9    0.108     1.26      -75      5.44        0        0      20.9    Abso Transportation
   10    0.131     1.25      -75      5.38   0.0544    0.046        21     Abso msc
   11    0.205     1.25    -74.8      5.04     0.34    0.178      21.1     Abso msc
   12    0.186     1.16    -74.7      4.79    0.252    0.178      21.3     Abso msc
   13    0.215      1.1    -74.6      4.35      0.1   0.0819      21.4     Abso eBrem
   14    0.333     0.97    -74.5      4.11    0.236    0.226      21.6     Abso msc
   15    0.459    0.786    -74.6      3.81    0.304    0.297      21.9     Abso msc
   16    0.533    0.752    -74.7      2.33    0.167    0.138      22.1     Abso eIoni
   17    0.672    0.733    -74.7      2.11    0.221    0.178      22.2     Abso msc
   18    0.735    0.696    -74.8      1.89    0.218     0.18      22.4     Abso msc
   19    0.693    0.531    -74.8      1.67     0.22    0.178      22.6     Abso msc
   20    0.765    0.459    -74.8      1.52     0.15    0.178      22.8     Abso msc
   21    0.758    0.499    -74.7     0.622    0.902    0.178      22.9     Abso msc
   22    0.744    0.489    -74.7      0.34    0.147    0.132      23.1     Abso eBrem
```

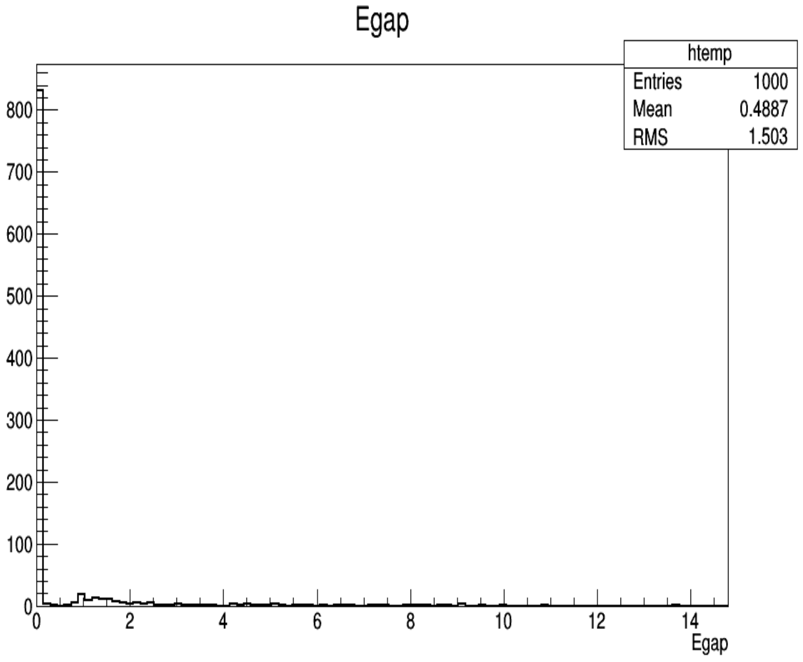
Analysis using ROOT

Different variables are stored in tree format for 1000 Events

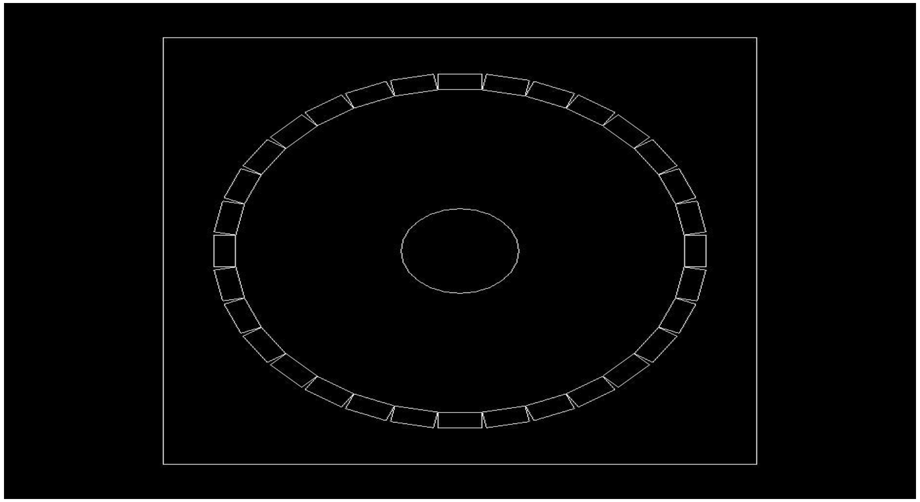
Energy Deposited in absorber of Calorimeter

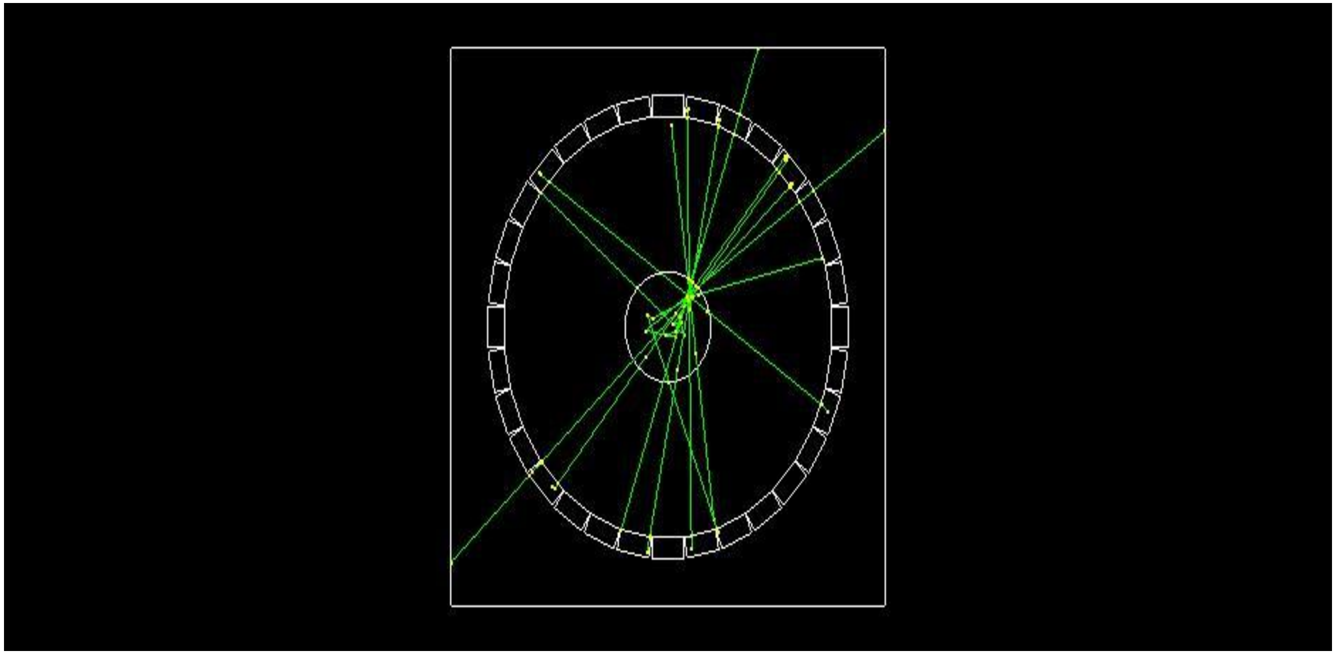


Energy Deposited in gap of Calorimeter



Lets run Example B3
and see Output





Ready for Exercises I, II , III

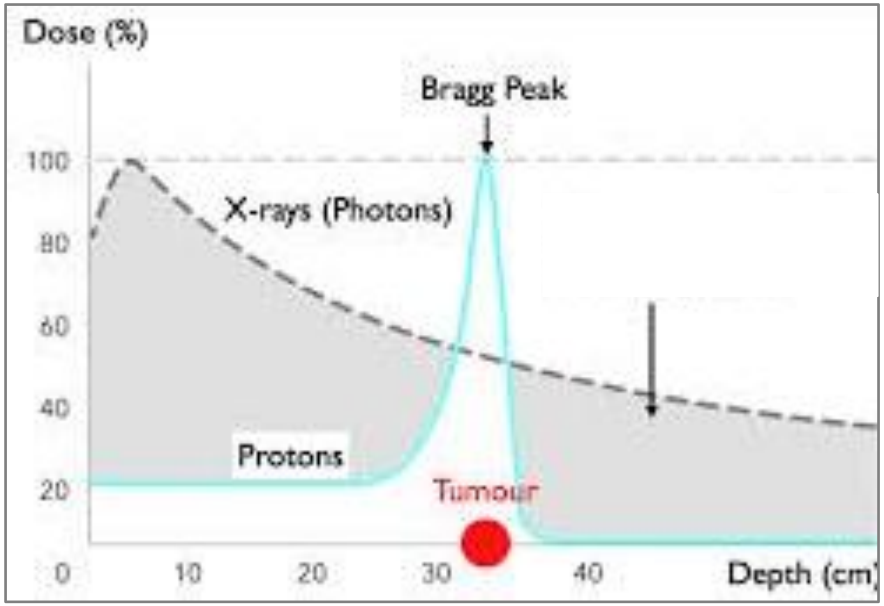
Thank You

Proton Therapy

✓The main advantage of using protons is the "Bragg peak": the way protons interacts with matter, they will give much more energy at a certain depth (Bragg peak)

✓ We can tune the energy of incoming proton beam and other design parameters, so that the Bragg peak coincide with the tumor.

- **Minimum Energy Dose**
- **Precise Patient Positioning as compared to X CT**



Principle of Proton CT (Proton Computed Tomography)

- Proton looses its energy mainly through two processes: Ionization & excitation of atomic electrons in medium

- Mean energy loss (Bethe Bloch formula)

$$\boxed{-\frac{dE}{dx}(\boldsymbol{r}) = \eta_e(\boldsymbol{r})\, F(I(\boldsymbol{r}), E(\boldsymbol{r}))} \longrightarrow \boxed{\int_S \eta_e(\boldsymbol{r})\,dx = \int_{E_{out}}^{E_{in}} \frac{dE}{F(I_{water}, E)}}$$

where $\eta_e = \dfrac{\rho_e}{\rho_{e,water}}$

- L.H.S: Integral of relative volume electron density along the proton path S
  In R.H.S, E(in) is the incident proton energy and E(out) is the proton energy after traversing the object.

- Integrated relative volume electron density can be calculated based on the knowledge of in- and out-going proton energy (Geant4 simulations)