Introduction to Geant4

Dr. Kavita Lalwani (Assistant Professor)
Department of Physics, MNIT Jaipur
Email: kavita.phy@mnit.ac.in

For Beginners: Basic Questions

Q1. What we can do with Geant4?

Q 2. From where I should start with Geant4?

Q 3. How can we create detector Geometry in Geant4?

Q 4. How can I generate the beam of proton OR electron in Geant4?

For Experts: Advance Questions?

Q1. How can we develop complex geometry such as Specific Detector (example: Calorimeter, RPC) in Geant4?

Q 2. How can we implement specific Physics List OR Physics Model ?

Q3 How can we generate differential distribution of flux/energy OR isotropic uniform spectrum of flux (eg. galactic cosmic radiation) in Geant4?

Outline

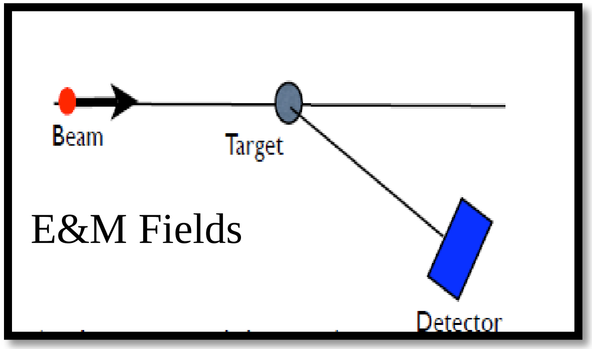
- Geant4 and its important features
- Geant4 Class Structure
- Method to construct a Detector Geometry
- Beam generation
- Physics List and Physics Processes
- Geant4 working flow chart
- Examples
- Applications

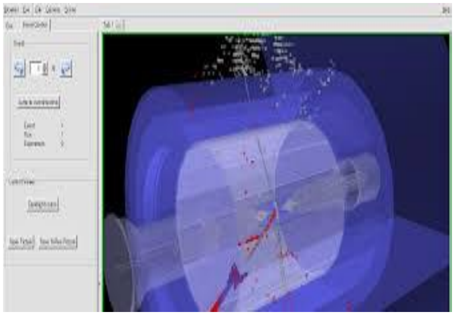
Note: This presentation is prepared with the training material of Geant4.

What we do with Geant4 & ROOT Softwares in EHEP?

1 Design New Experiment

Monte Carlo Simulation Tool (Geant4)



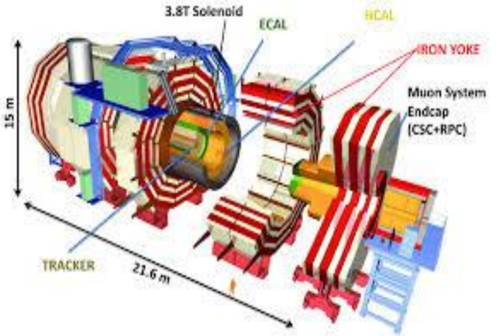


Belle II at SuperKEKB, Japan

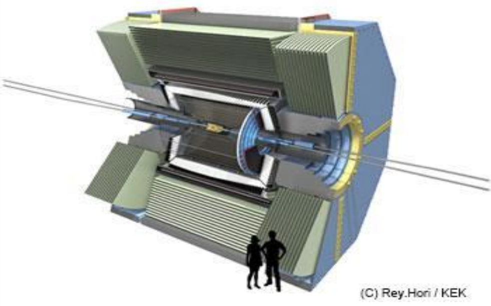
Use of Geant4 Simulation

- Design the New experiment.
- Optimize the detector geometry.
- Study the expected background and radiation level.
- Generate large amount of signal and background events for use physics analysis to study signal/background separation.
- Find Calibration Constants, efficiency, resolution study.

2 Choose any Exp. you want to participate



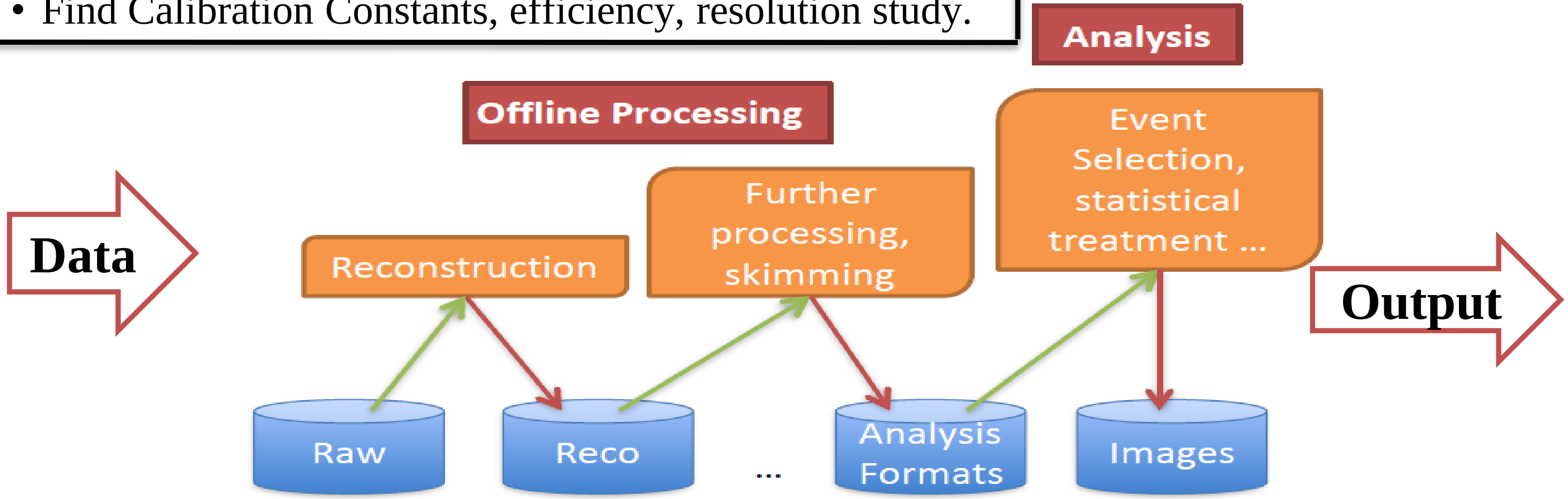
CMS Experiment at LHC, CERN



Belle II at SuperKEKB, Japan

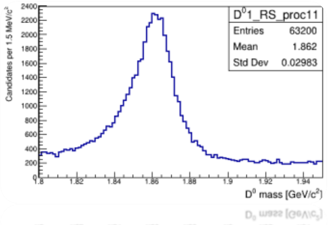
Data

3 Data Analysis: ROOT Framework



Ntuples/Histograms

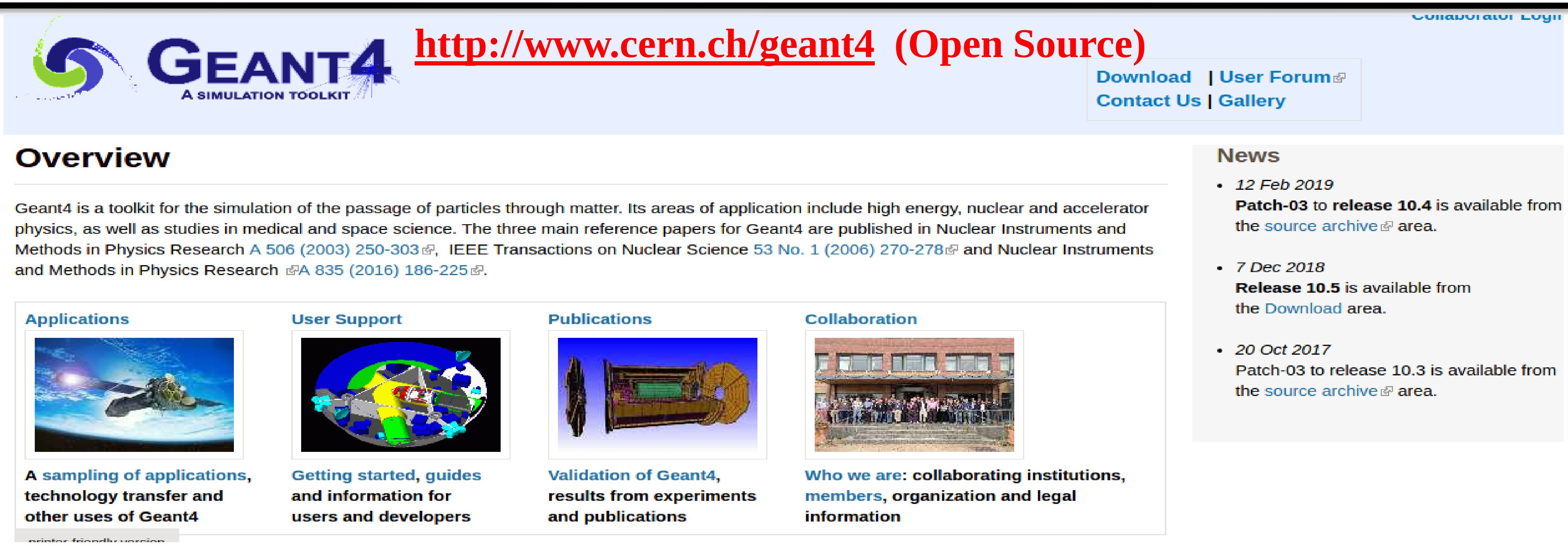
Particle Energy
Particle Momentum
Particle Tracking
Finding Vertex Info.
Particle Type



Monte Carlo Simulation tool: Geant4 (Geometry and Tracking)

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

- ○ GEANT4 is written in C++
- ○ Geant4 is mainly used for High energy Physics.
- ○ **Applications:** Accelerator physics, Medical as well as Space applications.
- ○ Using GEANT4 tool, user can build a simulation program for specific application.



http://www.cern.ch/geant4 (Open Source)

Collaborator Login

Download | User Forum
Contact Us | Gallery

Overview

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 three main reference papers for Geant4 are published in Nuclear Instruments and Methods in Physics Research A 506 (2003) 250-303, IEEE Transactions on Nuclear Science 53 No. 1 (2006) 270-278 and Nuclear Instruments and Methods in Physics Research A 835 (2016) 186-225.

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

printer friendly version

News

- *12 Feb 2019*
  Patch-03 to **release 10.4** is available from the source archive area.
- *7 Dec 2018*
  Release 10.5 is available from the Download area.
- *20 Oct 2017*
  Patch-03 to release 10.3 is available from the source archive area.

Important Features of Geant4

- Detector Geometry
- Materials involved
- Particle beam generation
- Generation of events
- Particle Tracking through matter and EM fields
- Response of detector components
- Storage of events and tracks
- Visualization
- Analysis of data

Users may construct stand-alone applications

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

Detector Construction in Geant4

(Define Materials and Geometries)

Three Volumes:

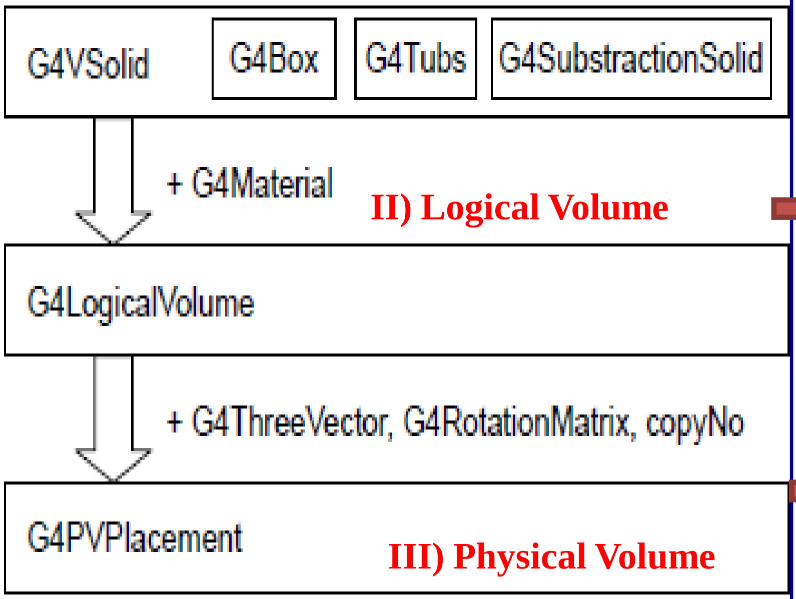
- **Geometrical (**G4VSolid) volumes specify shape and size.

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

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

G4VUserDetectorConstruction class

Flow Chart

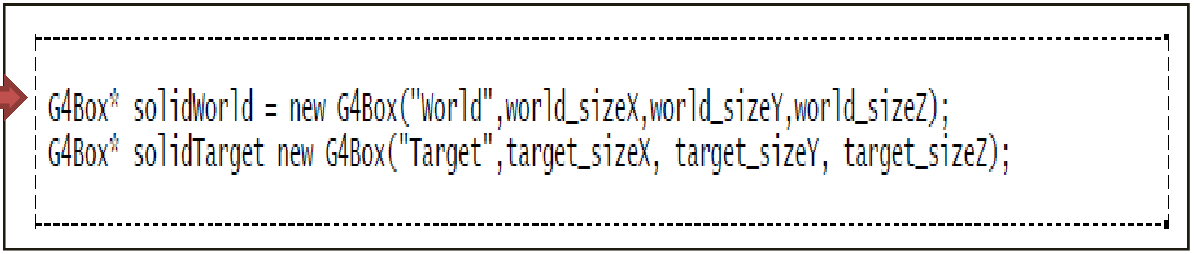


I) Geometrical Volume

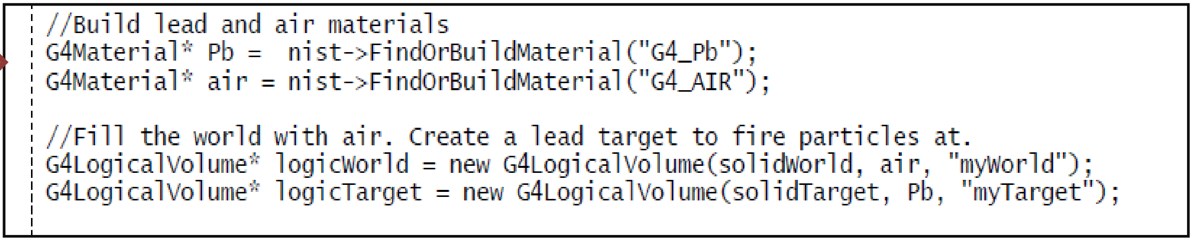
II) Logical Volume

III) Physical Volume

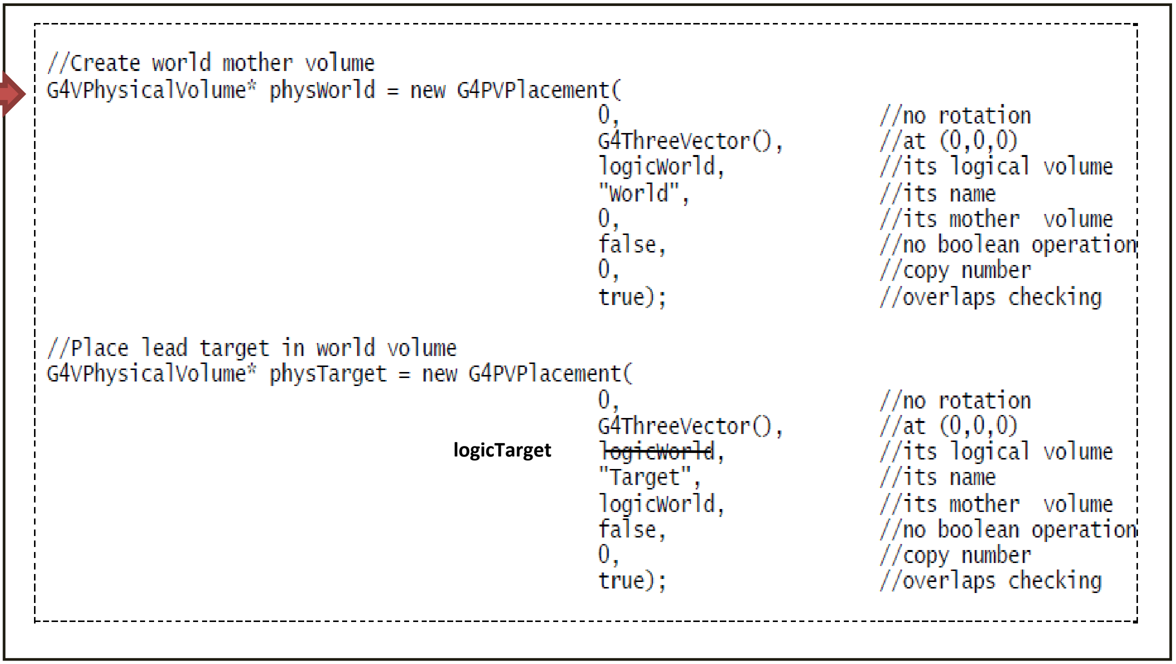
Code



```
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
```

Examples of Solid in Geant4:

Geant4 geometry module supports variety of representations of shapes.

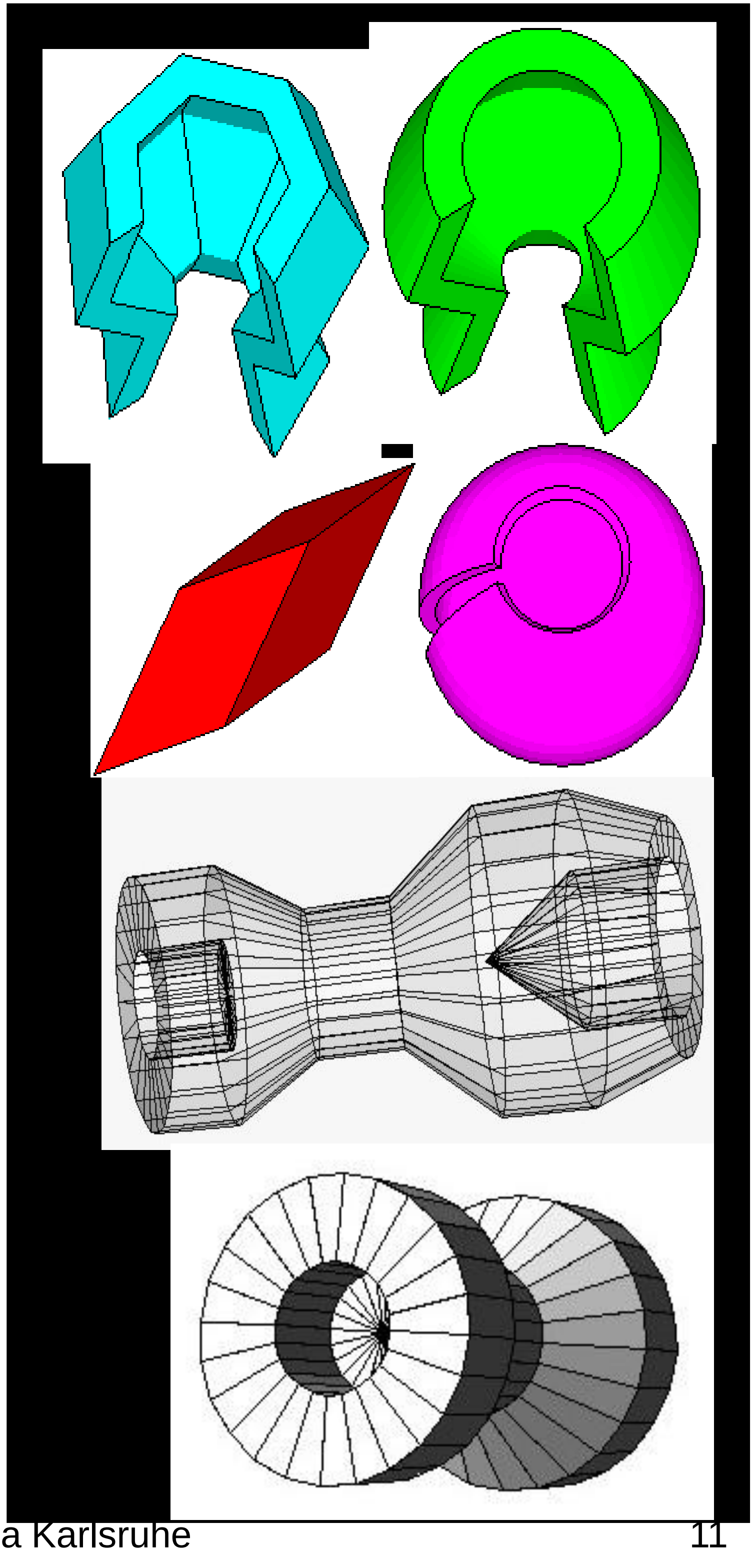
- CSG (Constructed Solid Geometry) solids
  - G4Box, G4Tubs, G4Cons, G4Trd, …
  - Analogous to simple GEANT3 CSG solids

- Specific solids (CSG like)
  - G4Polycone, G4Polyhedra, G4Hype, …
- BREP (Boundary REPresented) solids

  - G4BREPSolidPolycone, G4BSplineSurface, …

  - Any order surface

- Boolean solids
  - G4UnionSolid, G4SubtractionSolid, …



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:

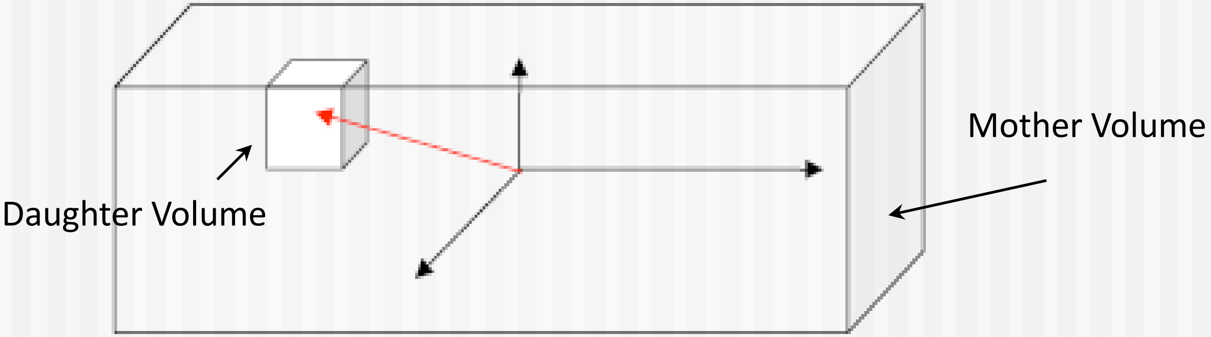
```
G4Material* H2O =
  new G4Material("Water", density=1.000*g/cm3,n_comps=2);

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

Compounds are mixture 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);
```

Placement
(Physical Volume)

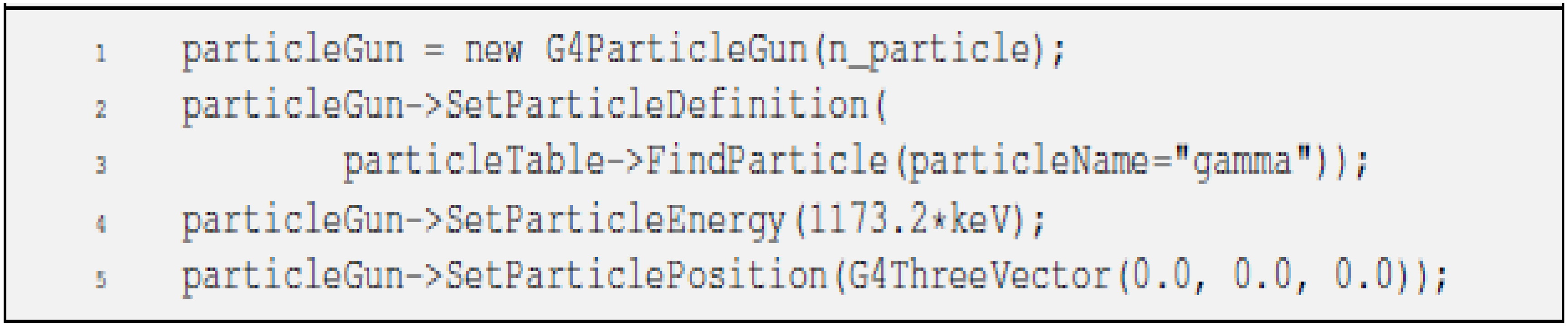
- Mother and daughter volumes
  - A volume is placed in its mother volume
    - Position and rotation of the daughter volume is described with respect to the local coordinate system of the mother volume
    - The origin of the mother's local coordinate system is at the center of the mother volume
  -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



Beam Generation in Geant4

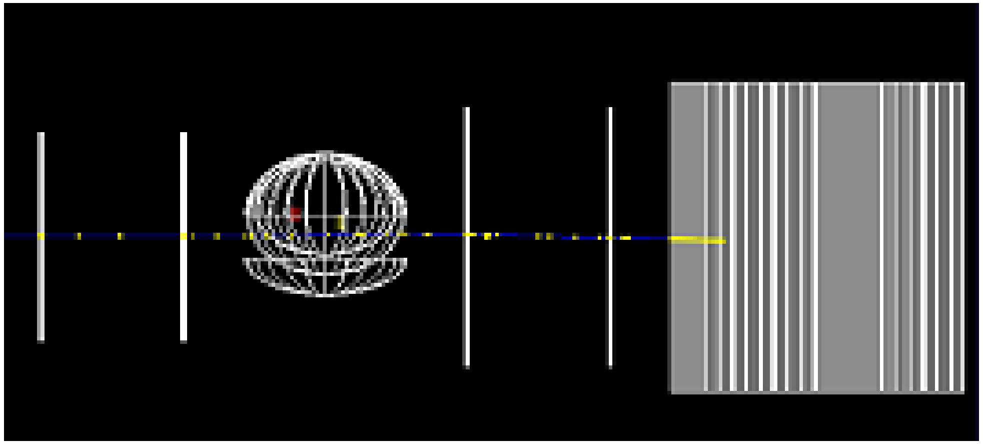
Primary Event Generator Action Class

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));
```

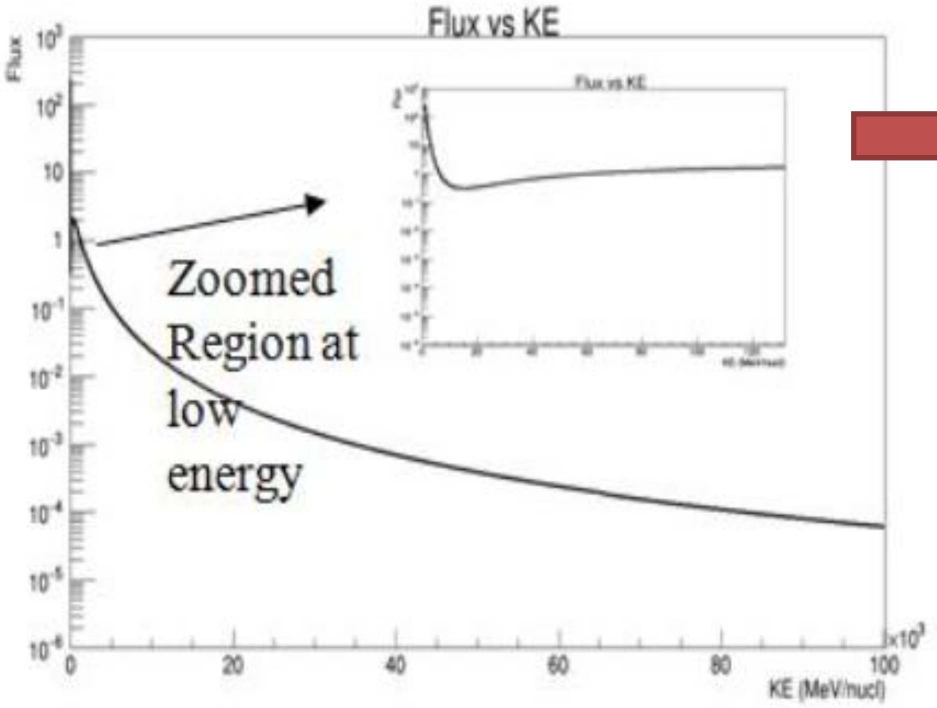
Proton Pencil Beam in PCT (Geant4 Visualization)



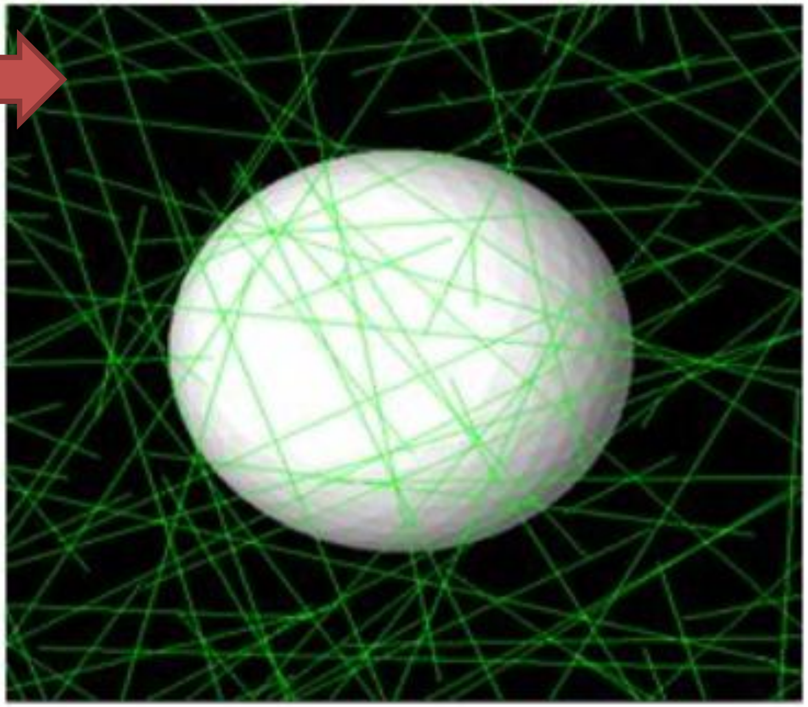
GPS (General Particle Source)

Example: Simulation of Proton Galactic Cosmic Ray Spectrum in Geant4

Input GCR spectrum thrown on Spherical detector geometry



Zoomed Region at low energy



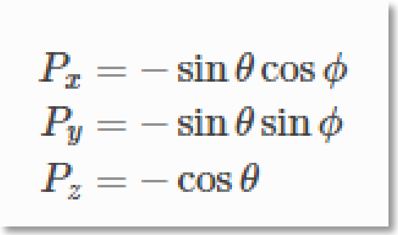
General Particle Source (GPS)

- The G4GeneralParticleSource (GPS) is part of the Geant4 toolkit for Monte-Carlo, high-energy particle transport.
- Specifically, it allows the specifications of the **spectral, spatial and angular distribution** of the primary source particles

Angular Distribution

The **angular distribution** is used to control the directions in which the particles emanate from/incident upon the source point.
In general there are three main choices, isotropic, cosine-law or user-defined.

The top-level /gps/direction command uses direction **cosines** to specify the primary particle direction, as follows:



$$P_x = -\sin\theta\cos\phi$$
$$P_y = -\sin\theta\sin\phi$$
$$P_z = -\cos\theta$$

Energy Distribution

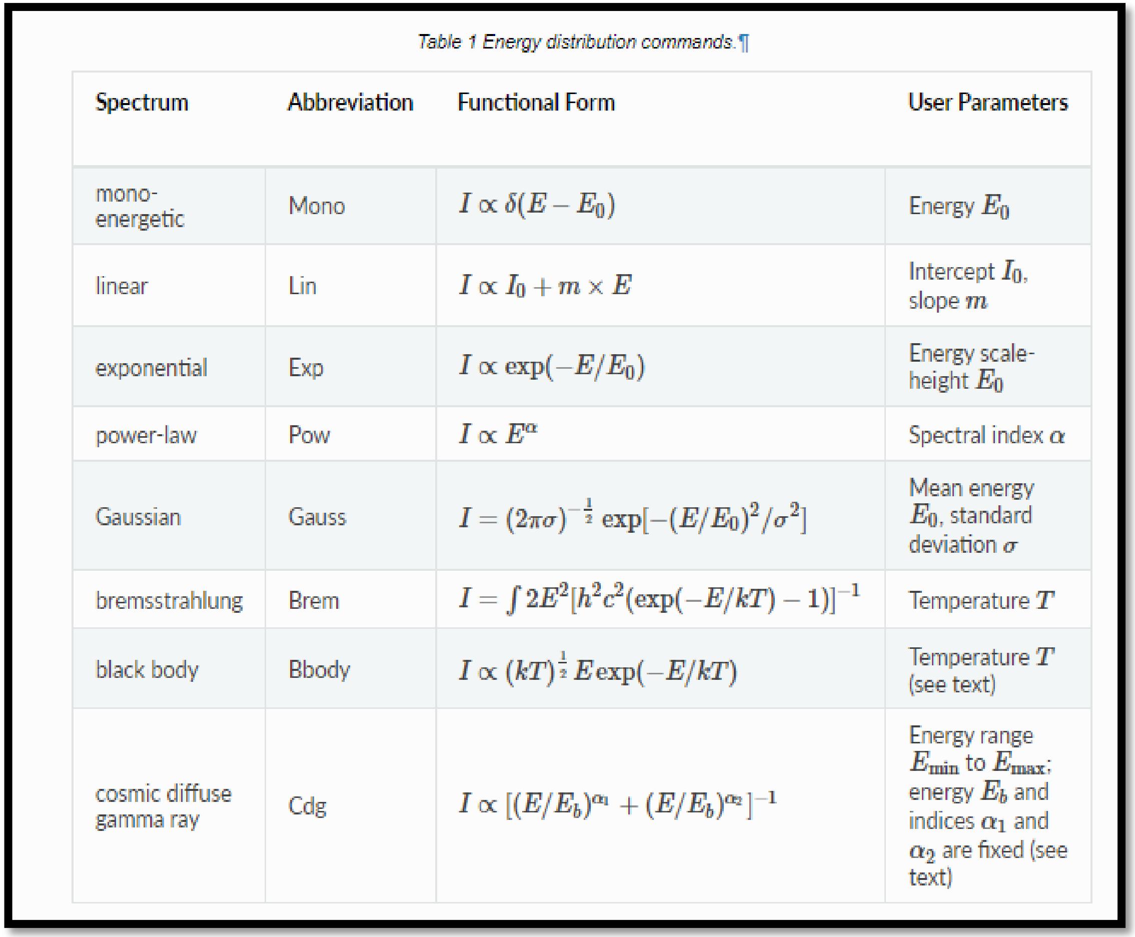


Table 1 Energy distribution commands.¶

Spectrum	Abbreviation	Functional Form	User Parameters
mono-energetic	Mono	$I \propto \delta(E - E_0)$	Energy E_0
linear	Lin	$I \propto I_0 + m \times E$	Intercept I_0, slope m
exponential	Exp	$I \propto \exp(-E/E_0)$	Energy scale-height E_0
power-law	Pow	$I \propto E^{\alpha}$	Spectral index α
Gaussian	Gauss	$I = (2\pi\sigma)^{-\frac{1}{2}} \exp[-(E/E_0)^2/\sigma^2]$	Mean energy E_0, standard deviation σ
bremsstrahlung	Brem	$I = \int 2E^2[h^2c^2(\exp(-E/kT) - 1)]^{-1}$	Temperature T
black body	Bbody	$I \propto (kT)^{\frac{1}{2}} E \exp(-E/kT)$	Temperature T (see text)
cosmic diffuse gamma ray	Cdg	$I \propto [(E/E_b)^{\alpha_1} + (E/E_b)^{\alpha_2}]^{-1}$	Energy range E_{min} to E_{max}; energy E_b and indices α_1 and α_2 are fixed (see text)

Physics Processes in Geant4 (Overview)

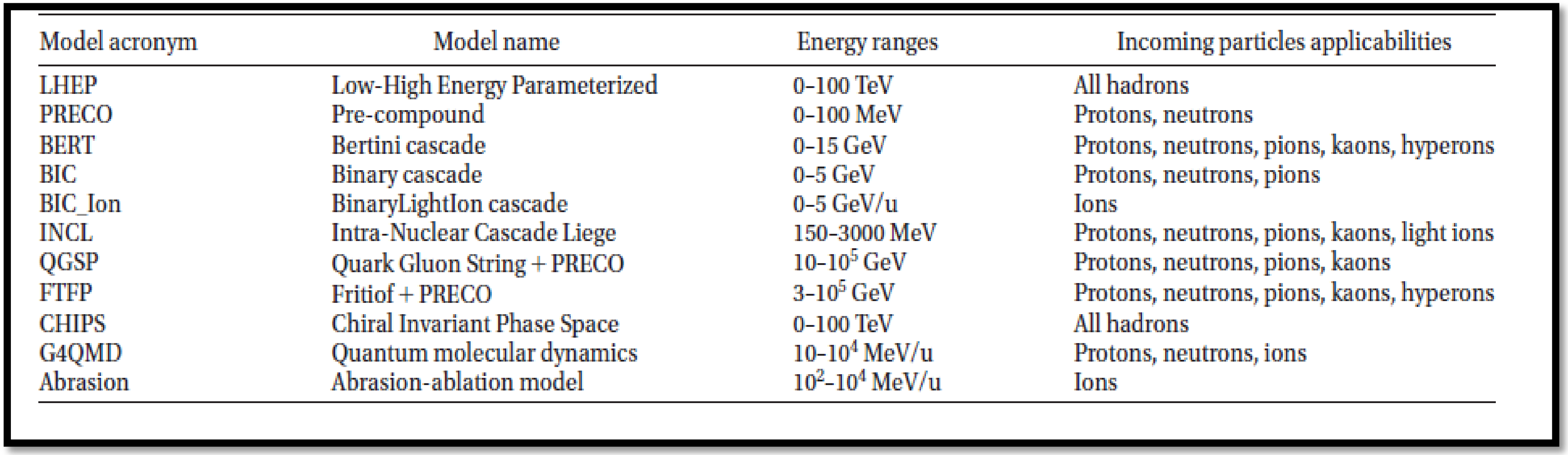
✔Geant4 provides a wide variety of physics components for use in simulation.

✔ 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.

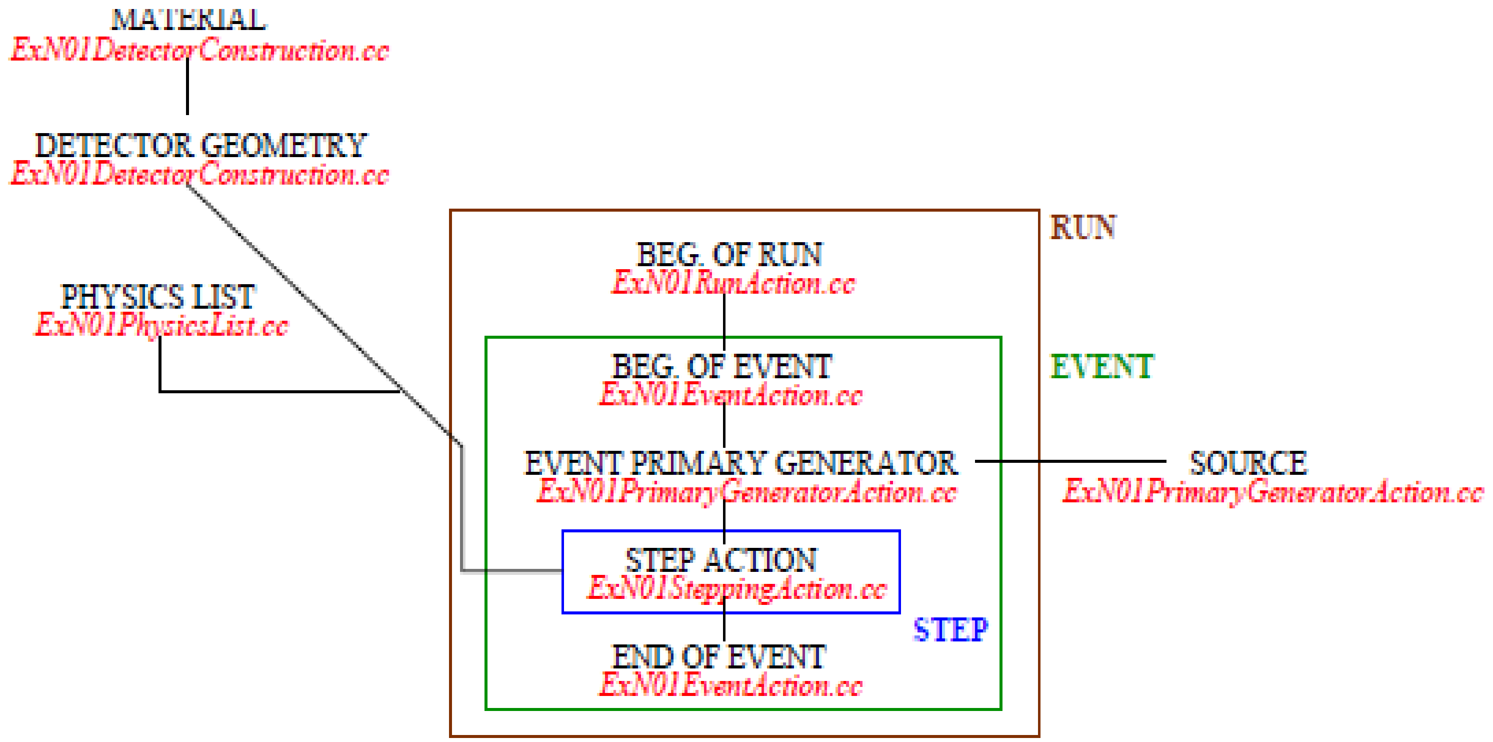
✔Processes are grouped into

- Electromagnetic
- Hadronic
- Decay



Model acronym	Model name	Energy ranges	Incoming particles applicabilities
LHEP	Low-High Energy Parameterized	0–100 TeV	All hadrons
PRECO	Pre-compound	0–100 MeV	Protons, neutrons
BERT	Bertini cascade	0–15 GeV	Protons, neutrons, pions, kaons, hyperons
BIC	Binary cascade	0–5 GeV	Protons, neutrons, pions
BIC_Ion	BinaryLightIon cascade	0–5 GeV/u	Ions
INCL	Intra-Nuclear Cascade Liege	150-3000 MeV	Protons, neutrons, pions, kaons, light ions
QGSP	Quark Gluon String + PRECO	10–10^5 GeV	Protons, neutrons, pions, kaons
FTFP	Fritiof + PRECO	3-10^5 GeV	Protons, neutrons, pions, kaons, hyperons
CHIPS	Chiral Invariant Phase Space	0–100 TeV	All hadrons
G4QMD	Quantum molecular dynamics	10–10^4 MeV/u	Protons, neutrons, ions
Abrasion	Abrasion-ablation model	10^2–10^4 MeV/u	Ions

Flow Chart in Geant4



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.

Visualization of Detector Geometry

Visualization Drivers:

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

What can be Visualized?

- Geometrical components
- Particle trajectories

Geant4 Examples

From where I should start in Geant4?

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: CMS Hadron Calorimeter;
- Advanced Medical applications
- Hadron Therapy
- LINAC

See many more updated examples in Geant4 webpage

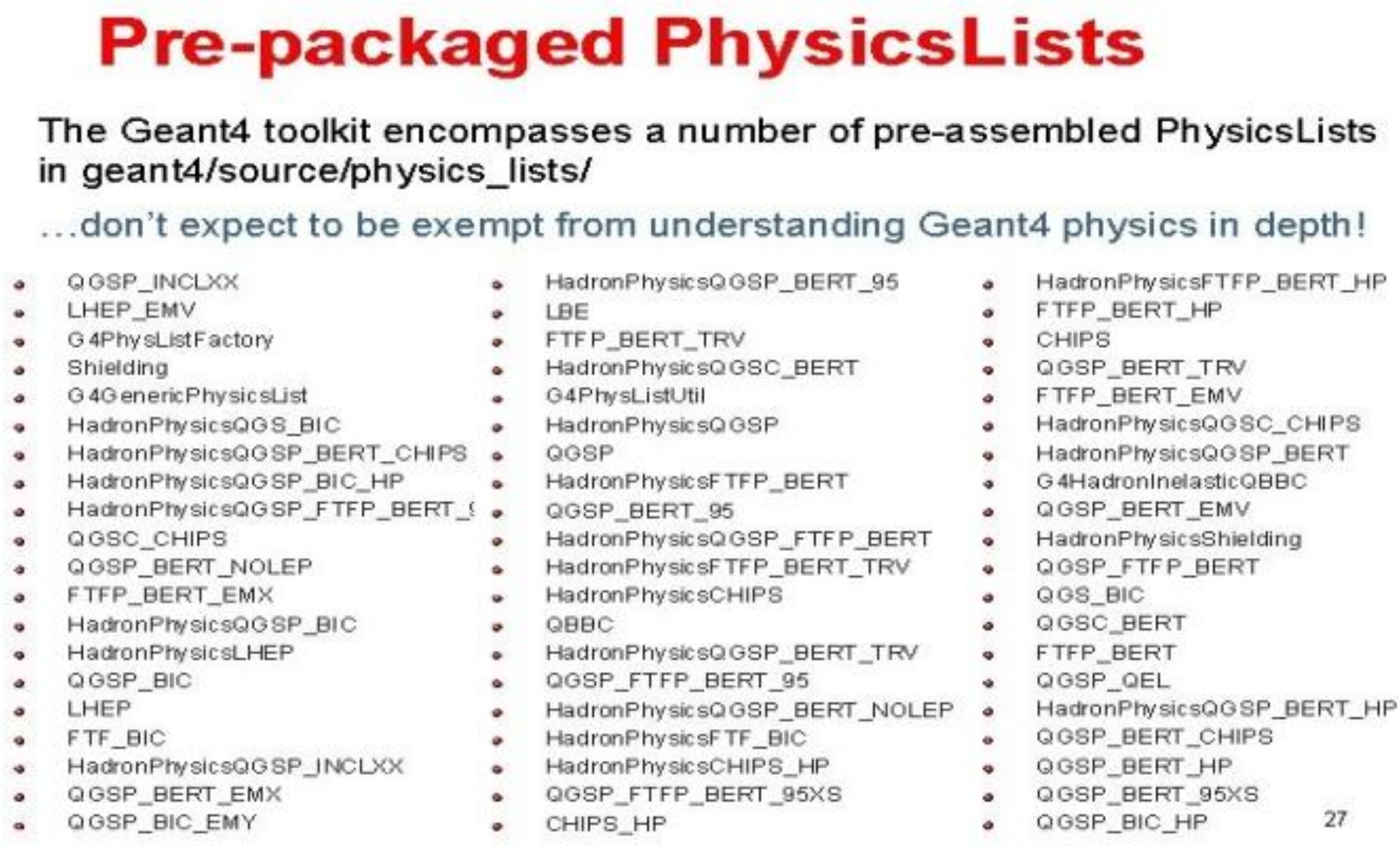
Physics List/Processes in Detail

Q. **Why do we need Physics list ?**

Answer: The concept of a physics list comes because GEANT4 can not offer a single modeling algorithm to cover the entire energy domain from zero to the TeV scale, for all known processes and particles.

To set up the physics environment in Geant4, we need to choose
(a) particle and (b) physics process that will be assigned to each particle.

Q How many physics lists are there in Geant4 ?



Pre-packaged PhysicsLists

The Geant4 toolkit encompasses a number of pre-assembled PhysicsLists in geant4/source/physics_lists/

...don't expect to be exempt from understanding Geant4 physics in depth!

- QGSP_INCLXX
- LHEP_EMV
- G4PhysListFactory
- Shielding
- G4GenericPhysicsList
- HadronPhysicsQGS_BIC
- HadronPhysicsQGSP_BERT_CHIPS
- HadronPhysicsQGSP_BIC_HP
- HadronPhysicsQGSP_FTFP_BERT_
- QGSC_CHIPS
- QGSP_BERT_NOLEP
- FTFP_BERT_EMX
- HadronPhysicsQGSP_BIC
- HadronPhysicsLHEP
- QGSP_BIC
- LHEP
- FTF_BIC
- HadronPhysicsQGSP_INCLXX
- QGSP_BERT_EMX
- QGSP_BIC_EMY
- HadronPhysicsQGSP_BERT_95
- LBE
- FTFP_BERT_TRV
- HadronPhysicsQGSC_BERT
- G4PhysListUtil
- HadronPhysicsQGSP
- QGSP
- HadronPhysicsFTFP_BERT
- QGSP_BERT_95
- HadronPhysicsQGSP_FTFP_BERT
- HadronPhysicsFTFP_BERT_TRV
- HadronPhysicsCHIPS
- QBBC
- HadronPhysicsQGSP_BERT_TRV
- QGSP_FTFP_BERT_95
- HadronPhysicsQGSP_BERT_NOLEP
- HadronPhysicsFTF_BIC
- HadronPhysicsCHIPS_HP
- QGSP_FTFP_BERT_95XS
- CHIPS_HP
- HadronPhysicsFTFP_BERT_HP
- FTFP_BERT_HP
- CHIPS
- QGSP_BERT_TRV
- FTFP_BERT_EMV
- HadronPhysicsQGSC_CHIPS
- HadronPhysicsQGSP_BERT
- G4HadronInelasticQBBC
- QGSP_BERT_EMV
- HadronPhysicsShielding
- QGSP_FTFP_BERT
- QGS_BIC
- QGSC_BERT
- FTFP_BERT
- QGSP_QEL
- HadronPhysicsQGSP_BERT_HP
- QGSP_BERT_CHIPS
- QGSP_BERT_HP
- QGSP_BERT_95XS
- QGSP_BIC_HP

Physics processes in Geant4

EM physics

- Low energy processes can be used from 100eV to PeV.

2. Weak interaction physics

- Decay of subatomic particles
- Radio-active decay of nuclei

3. Hadronic physics

- Pure strong interaction valid from **0eV to TeV**

- High precision neutron package valid from **thermal energies to ~20MeV**

How to create a physics list ?

There are three options to create a physics list :-

1. **Create and inherit it from G4VUserPhysicsList**

- In this, specify all particles needed
- For each particle specify processes.
- Due to this a large number of lines of code is there and difficult to solve the problems.

2. **Create and inherit from G4VModularPhysicsList**

- This allows to use existing physics constructors
- Can construct own constructors

3. **Reuse pre-packaged physics list directly via G4PhysicsListFactory**

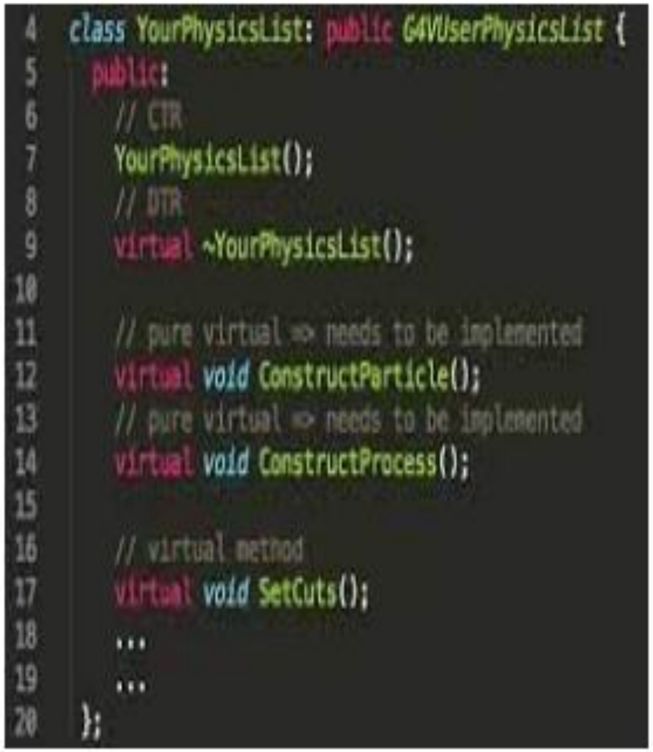
Physics list in Geant4 (continue..)

- It is the basic Geant4 physics list interface.
- All physics list derived from this base class.

- In this user implement two virtual methods:-

1. constructParticle( ) :- using this we can create all particles needed in the simulation and along with secondary particles possibly created in simulation.

2. ConstructProcess( ) :-
using this we have to assign specific process to each particle.
User can implement the SetCuts( ) method (optional)

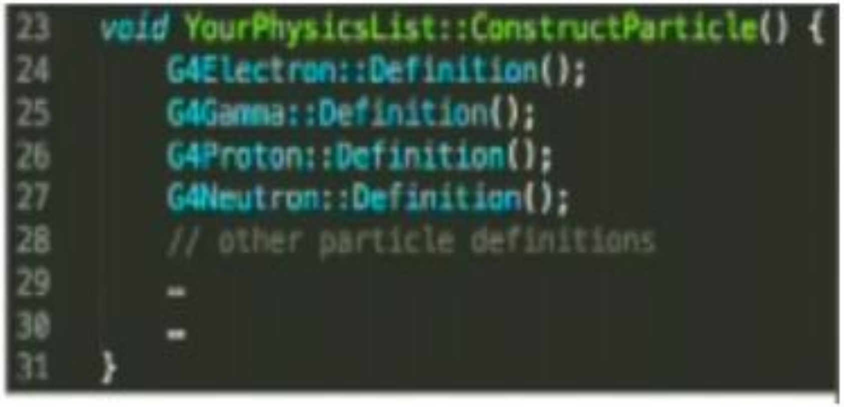


```
class YourPhysicsList: public G4VUserPhysicsList {
  public:
    // CTR
    YourPhysicsList();
    // DTR
    virtual ~YourPhysicsList();

    // pure virtual => needs to be implemented
    virtual void ConstructParticle();
    // pure virtual => needs to be implemented
    virtual void ConstructProcess();

    // virtual method
    virtual void SetCuts();
    ...
    ...
 };
```

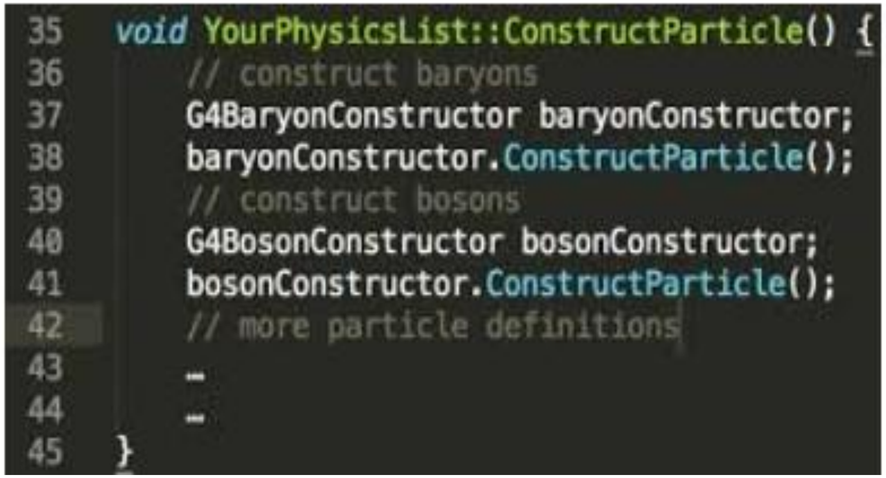
Physics list in Geant4 (continue..)

G4VUserPhysicsList:CreateParticles()

- **Construct particles individually one by one**
- - many particles in G4
- - particle classes
- Gluons, quarks, di-Quarks
- Leptons
- Mesons
- Baryons
- Ions
- Others
- **Construct particles by using helpers**
- - helpers are under particles
- Leptons
- Baryons
- Mesons
- Ions
- Short-lived
- Excited nucleons



```
void YourPhysicsList::ConstructParticle() {
    G4Electron::Definition();
    G4Gamma::Definition();
    G4Proton::Definition();
    G4Neutron::Definition();
    // other particle definitions
    …
    …
}
```

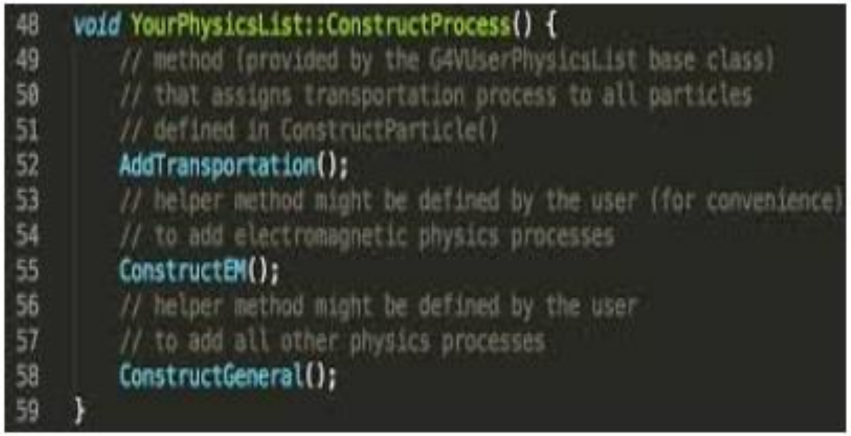


```
void YourPhysicsList::ConstructParticle() {
    // construct baryons
    G4BaryonConstructor baryonConstructor;
    baryonConstructor.ConstructParticle();
    // construct bosons
    G4BosonConstructor bosonConstructor;
    bosonConstructor.ConstructParticle();
    // more particle definitions
    …
    …
}
```

Physics list in Geant4 (continue..)

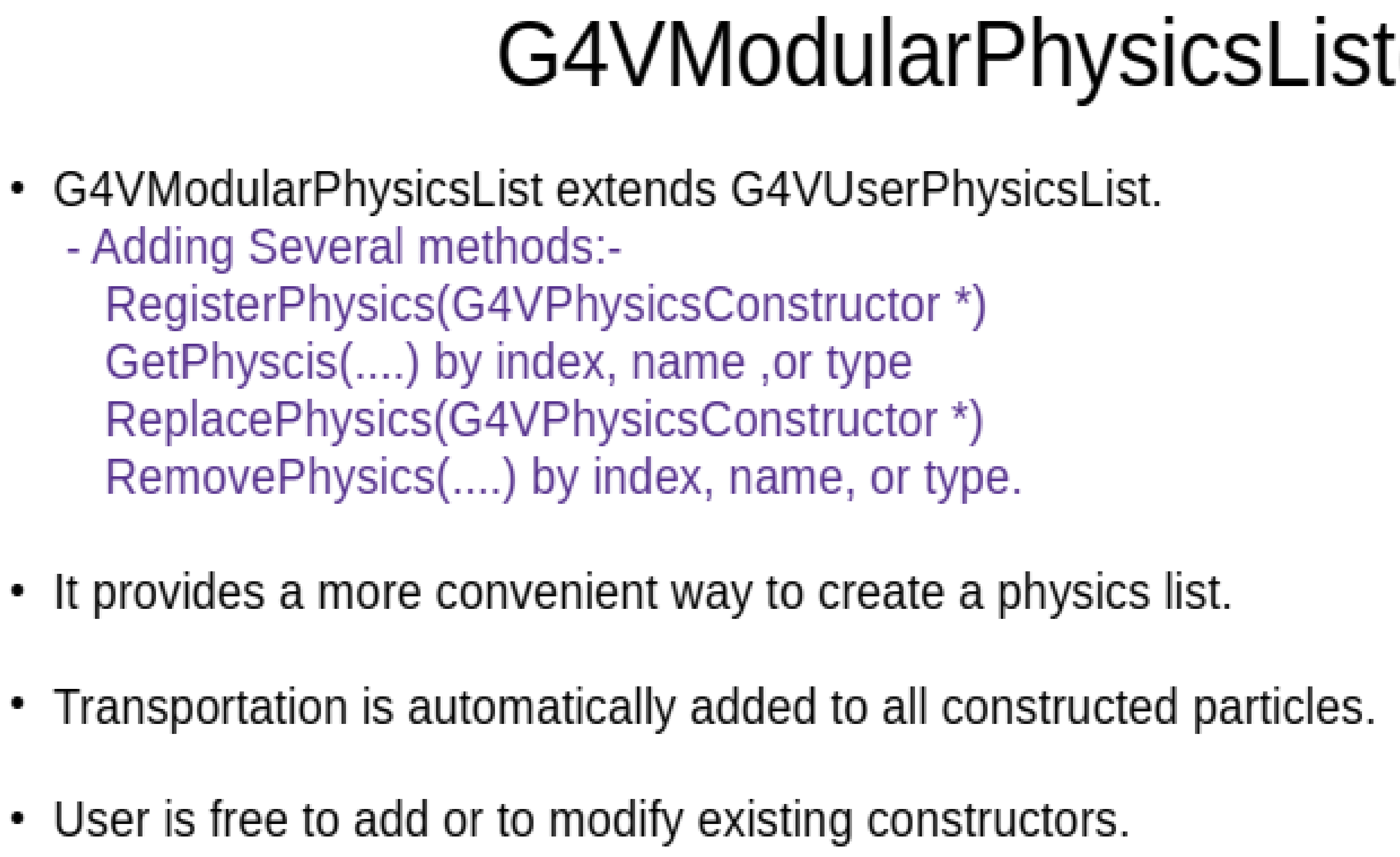
G4VUserPhysicsList:constructProcess()

- A process in Geant4 **describes reaction probability** and creates final state of interaction.
- GeneralConstruct method split into components for EM and hadronic etc.
- Transportation must be added.



```
void YourPhysicsList::ConstructProcess() {
    // method (provided by the G4VUserPhysicsList base class)
    // that assigns transportation process to all particles
    // defined in ConstructParticle()
    AddTransportation();
    // helper method might be defined by the user (for convenience)
    // to add electromagnetic physics processes
    ConstructEM();
    // helper method might be defined by the user
    // to add all other physics processes
    ConstructGeneral();
}
```

Physics list in Geant4 (continue..)



G4VModularPhysicsList()

- G4VModularPhysicsList extends G4VUserPhysicsList.
  - Adding Several methods:-
    RegisterPhysics(G4VPhysicsConstructor *)
    GetPhyscis(....) by index, name ,or type
    ReplacePhysics(G4VPhysicsConstructor *)
    RemovePhysics(....) by index, name, or type.
- It provides a more convenient way to create a physics list.
- Transportation is automatically added to all constructed particles.
- User is free to add or to modify existing constructors.

For more detail, please visit Geant4 webpage and try to implement your own physics list

Applications

Example: 1) Simulation of Space Radiation Environment

Space Radiation Environment

- The space radiation consists of charged particles and heavy ions, which come from the burst of solar wind created at the surface of the sun.

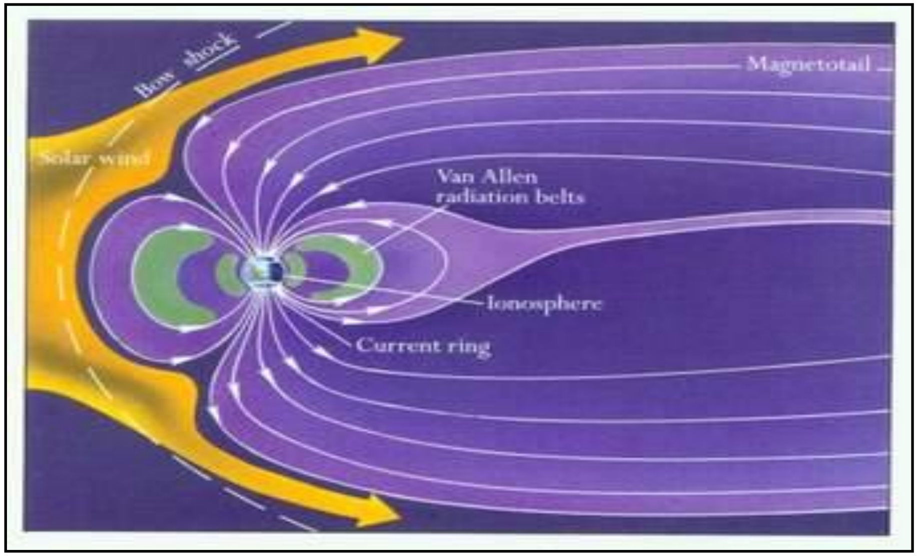
- The interaction of the particles and the magnetic field of earth occur and particles get deflected.

- Still some particles manage to enter the magnetosphere and become trapped.

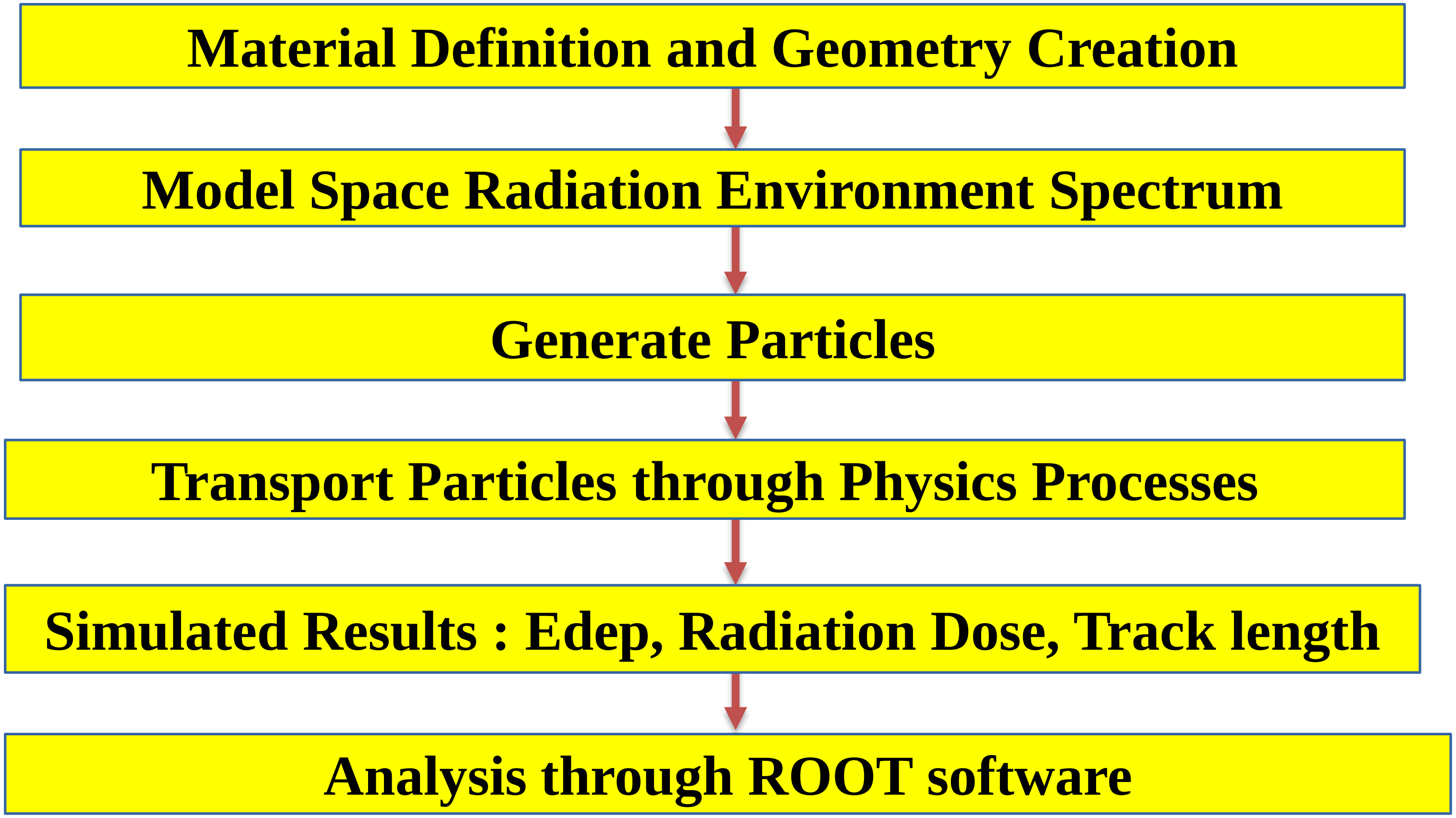
- These trapped particles are contained in two doughnut-shaped magnetic rings around earth called the Van Allen radiation belts.

- Three main components of space radiations:
  - Trapped particles
  - **Galactic Cosmic Rays (GCRs)**
  - **Solar Particle Events (SPEs)**

The energies of space radiation lies in the range of eV to the order of few TeV.



Flow chart to Model Space Radiation Environment in Geant4



Material Definition and Geometry Creation

Model Space Radiation Environment Spectrum

Generate Particles

Transport Particles through Physics Processes

Simulated Results : Edep, Radiation Dose, Track length

Analysis through ROOT software

Energy Deposited

- When beam of charged particles enter the detector material, they loss their energy via coulomb interactions with electrons and nucleus of the atom.

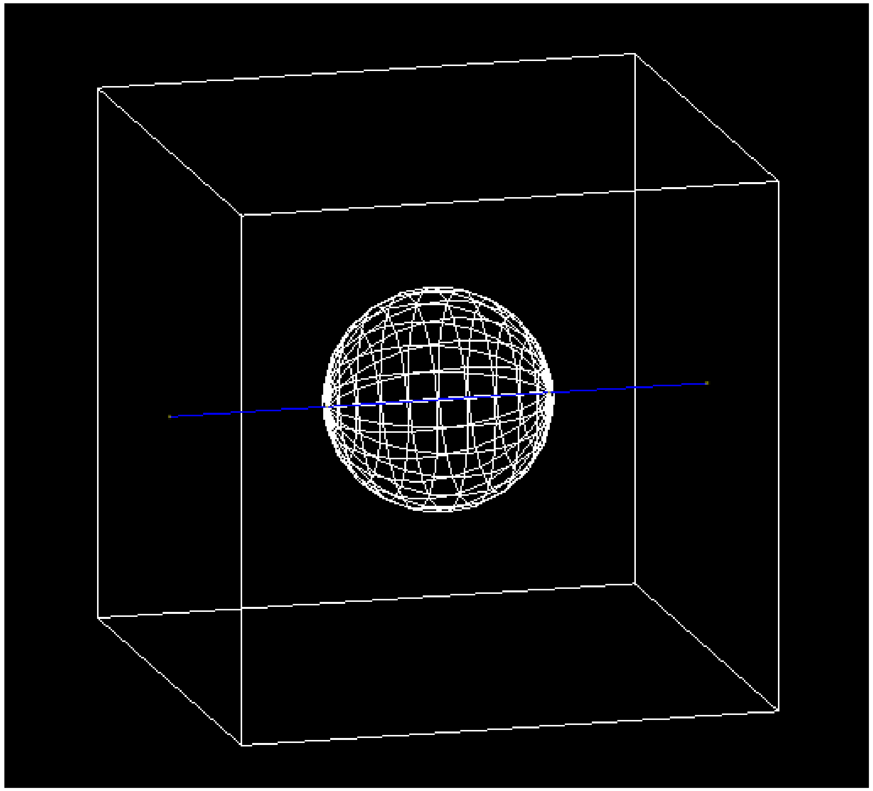
- The energy loss by charged particles in the detector material is measured through energy deposited, which is given by Bethe-Bloch formula:

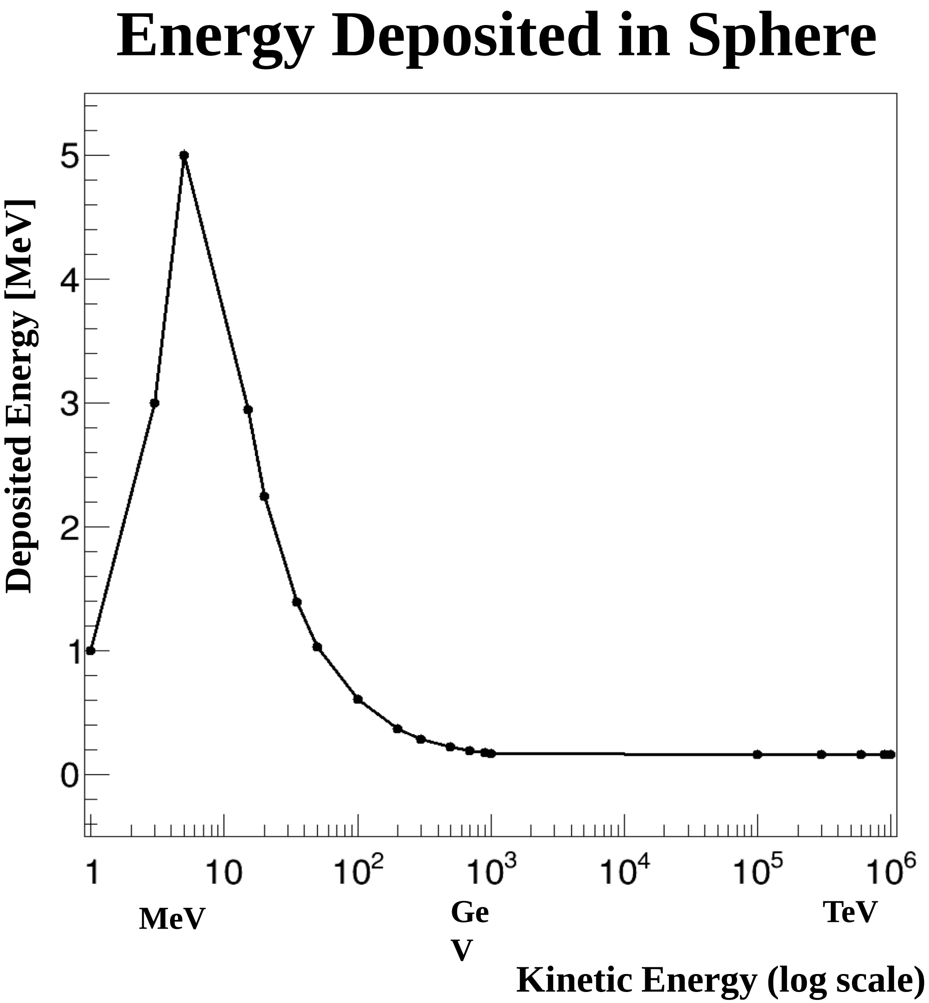
$$\boxed{\frac{S}{\rho} = -\frac{dE}{\rho dx} = 4\pi N_A r_e{}^2 m_e c^2 \frac{Z}{A}\frac{z^2}{\beta^2}\left[\ln\frac{2m_e c^2\beta^2}{I\cdot(1-\beta^2)} - \beta^2 - \frac{\delta}{2} - \frac{C}{Z}\right]}$$

➡ dE/dx is proportional to the inverse square of the incoming particle's velocity, and the square of its ion charge z.

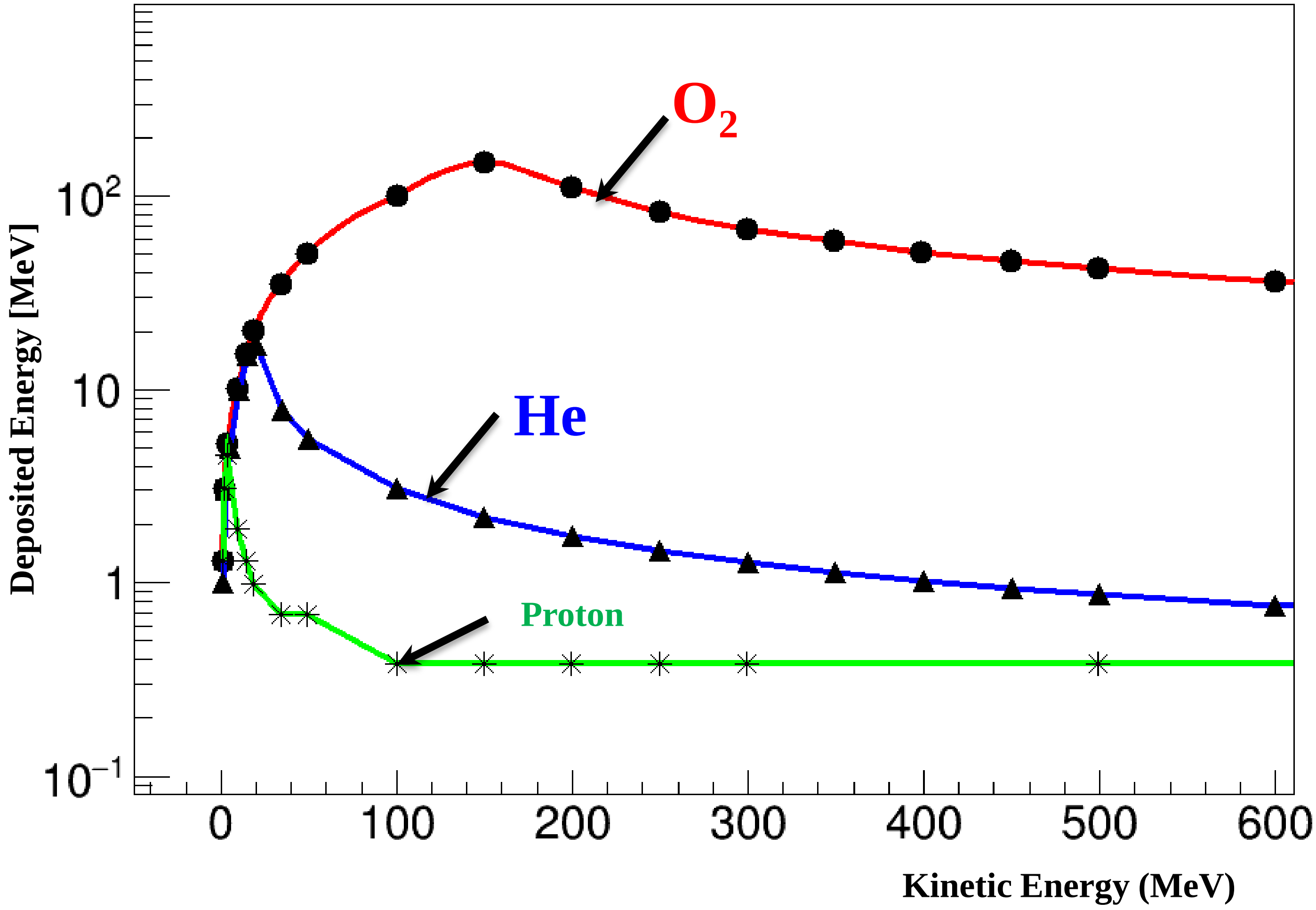
Visualization of Detector Geometry

Geometry: Sphere
Size: Radius = 414 μm
Material: Water
Incoming Beam: Proton





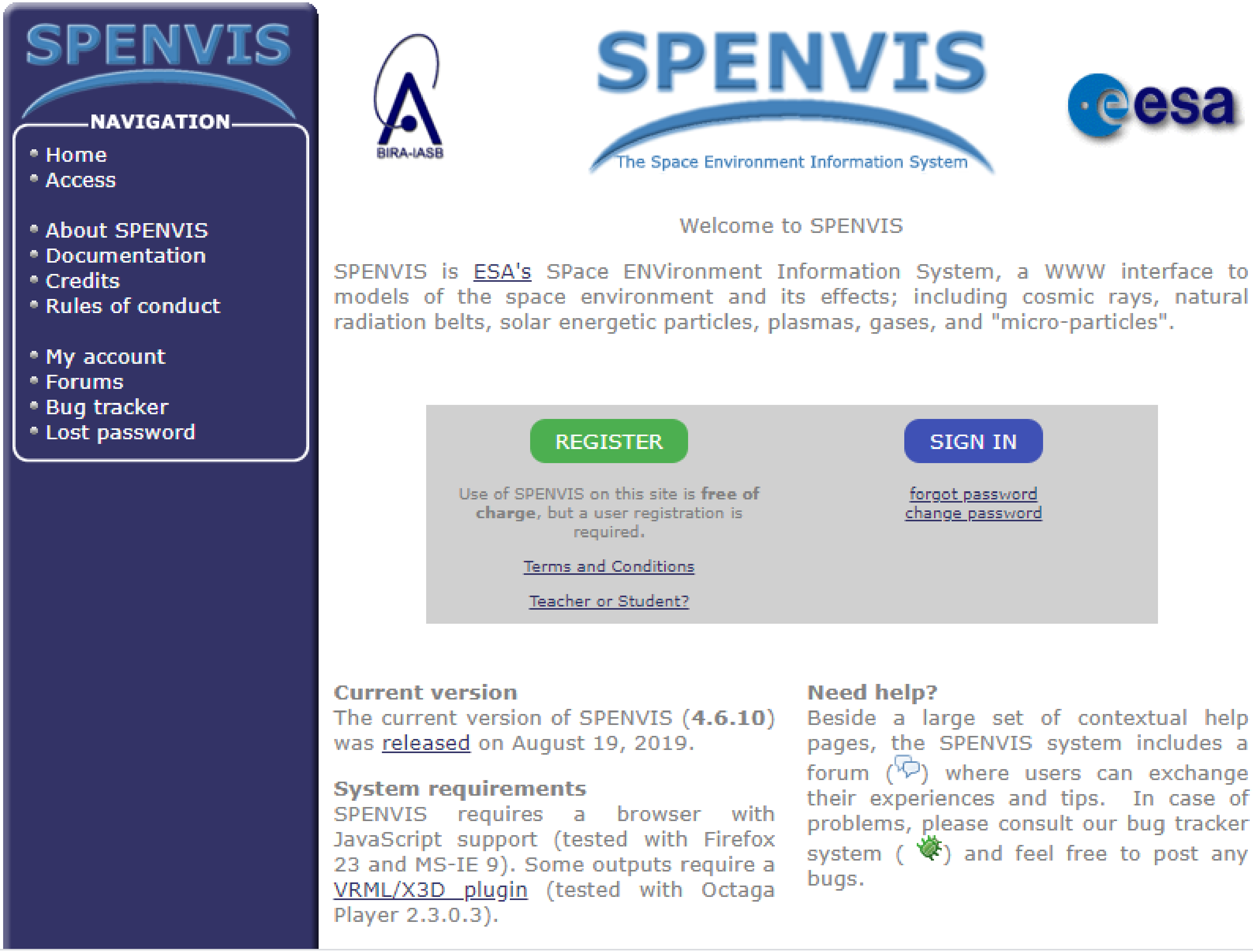
Effect of Charge on Energy Deposited



Geant4 based Online Software Tool: SPENVIS

For Space Applications

https://www.spenvis.oma.be/



Welcome to SPENVIS

SPENVIS is ESA's SPace ENVironment Information System, a WWW interface to models of the space environment and its effects; including cosmic rays, natural radiation belts, solar energetic particles, plasmas, gases, and "micro-particles".

Use of SPENVIS on this site is **free of charge**, but a user registration is required.

Current version
The current version of SPENVIS (**4.6.10**) was released on August 19, 2019.

System requirements
SPENVIS requires a browser with JavaScript support (tested with Firefox 23 and MS-IE 9). Some outputs require a VRML/X3D plugin (tested with Octaga Player 2.3.0.3).

Need help?
Beside a large set of contextual help pages, the SPENVIS system includes a forum ( ) where users can exchange their experiences and tips. In case of problems, please consult our bug tracker system ( ) and feel free to post any bugs.

2) GDML File in Geant4

GEOMETRY DESCRIPTION MARKUP LANGUAGE

Geant4 can Convert CAD Files to GDML Files, Create New GDML Geometries

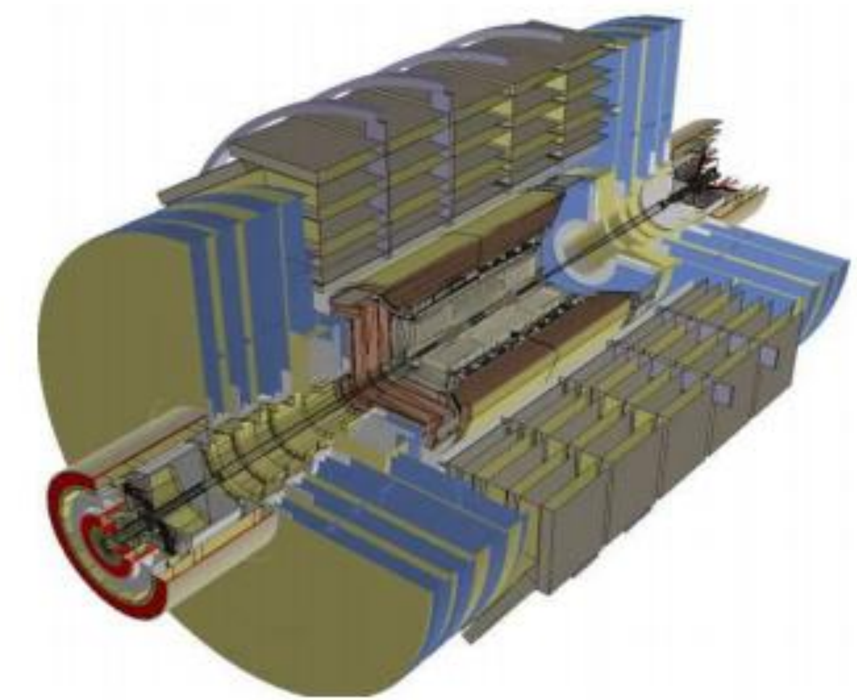


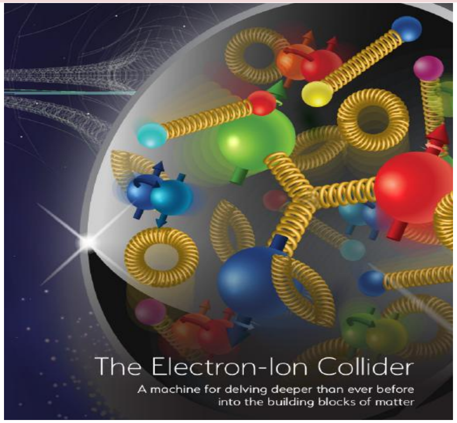
Figure 5: CMS detector visualized using ROOT from an automatically generated GDML file.

You can start with existing Examples: examples/extended/gdml

2) Electron Ion Collider (Future Facility in EHEP)

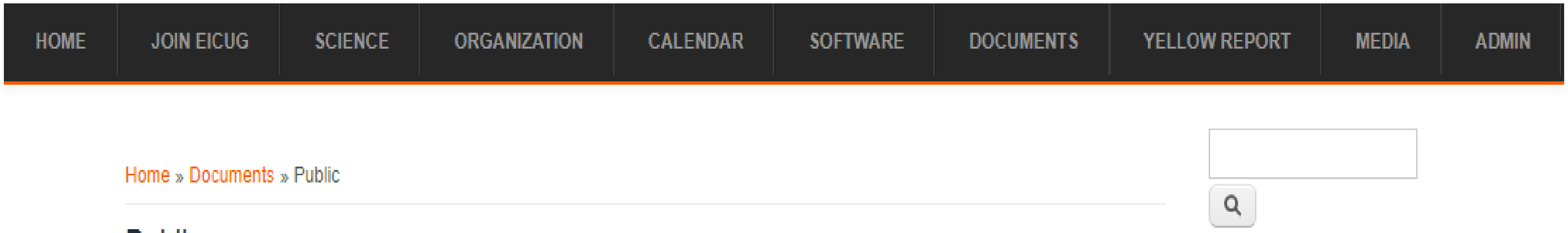
@BNL, USA

http://eicug.org/web/content/eic-software


The Electron-Ion Collider
A machine for delving deeper than ever before into the building blocks of matter

Electron-Ion Collider User Group

The world's most powerful microscope for studying the "glue" that binds the building blocks of visible matter.



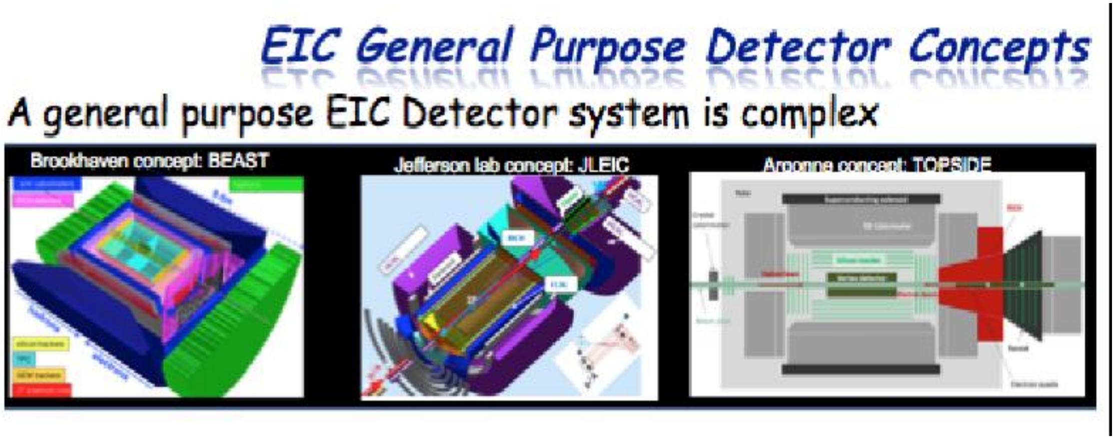
Home » Documents » Public

Public

EICUG Documentation and Publication

Documents produced by EIC-Community

- Memo EIC Physics and Detector Conceptual Development (2019)
- EIC White Paper (2nd edition 12/2014)
- European Strategy report on the EIC accelerator (2018)
- European Strategy report on the EIC physics (2018)
- Report on the 2010 joint BNL/INT/Jlab program on the science case for an Electron-Ion Collider (2010)
- EIC Brochure

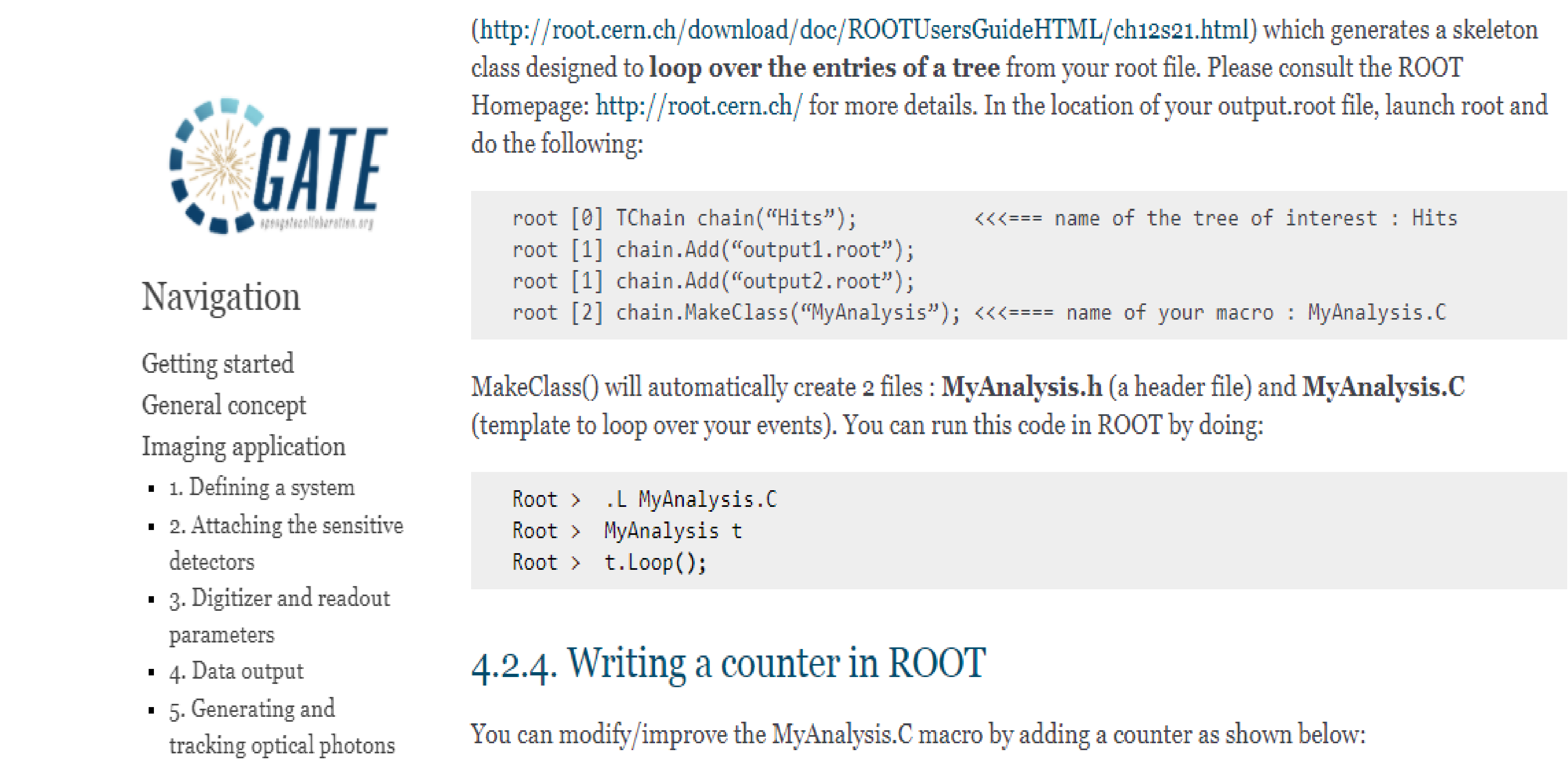


EIC General Purpose Detector Concepts

A general purpose EIC Detector system is complex

3) Geant4 based Online Software Tool: GATE

https://opengate.readthedocs.io/en/latest/introduction.html



Navigation

Getting started

General concept

Imaging application

- 1. Defining a system
- 2. Attaching the sensitive detectors
- 3. Digitizer and readout parameters
- 4. Data output
- 5. Generating and tracking optical photons
- 6. Compton camera imaging simulations: CCMod

Radiotherapy and dosimetry applications

Thermal therapy application

(http://root.cern.ch/download/doc/ROOTUsersGuideHTML/ch12s21.html) which generates a skeleton class designed to **loop over the entries of a tree** from your root file. Please consult the ROOT Homepage: http://root.cern.ch/ for more details. In the location of your output.root file, launch root and do the following:

```
root [0] TChain chain(“Hits”);          <<<=== name of the tree of interest : Hits
root [1] chain.Add(“output1.root”);
root [1] chain.Add(“output2.root”);
root [2] chain.MakeClass(“MyAnalysis”); <<<==== name of your macro : MyAnalysis.C
```

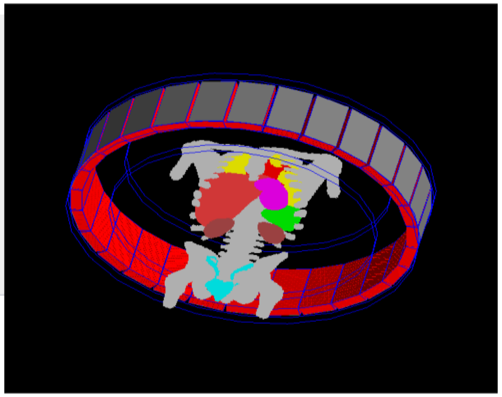
MakeClass() will automatically create 2 files : **MyAnalysis.h** (a header file) and **MyAnalysis.C** (template to loop over your events). You can run this code in ROOT by doing:

```
Root >  .L MyAnalysis.C
Root >  MyAnalysis t
Root >  t.Loop();
```

4.2.4. Writing a counter in ROOT

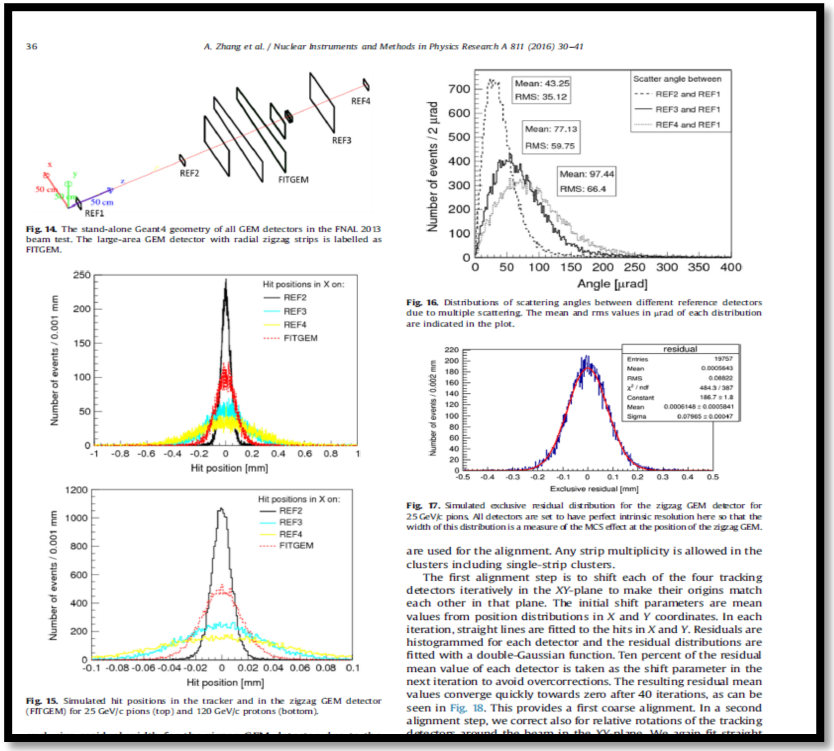
You can modify/improve the MyAnalysis.C macro by adding a counter as shown below:

```
void MyAnalysis::Loop()
{
if (fChain == 0) return;
Long64_t nentries = fChain->GetEntriesFast();
Long64_t nbytes = 0, nb = 0;
Int_t num_INITIAL = 0;
Int_t num_DETECTED = 0;
```



Scientific Publications available using Geant4 & ROOT

EHEP Detector Paper (Geant4 & ROOT)



36 A. Zhang et al. / Nuclear Instruments and Methods in Physics Research A 811 (2016) 30–41

Fig. 14. The stand-alone Geant4 geometry of all GEM detectors in the FNAL 2013 beam test. The large-area GEM detector with radial zigzag strips is labelled as FITGEM.

Fig. 15. Simulated hit positions in the tracker and in the zigzag GEM detector (FITGEM) for 25 GeV/c pions (top) and 120 GeV/c protons (bottom).

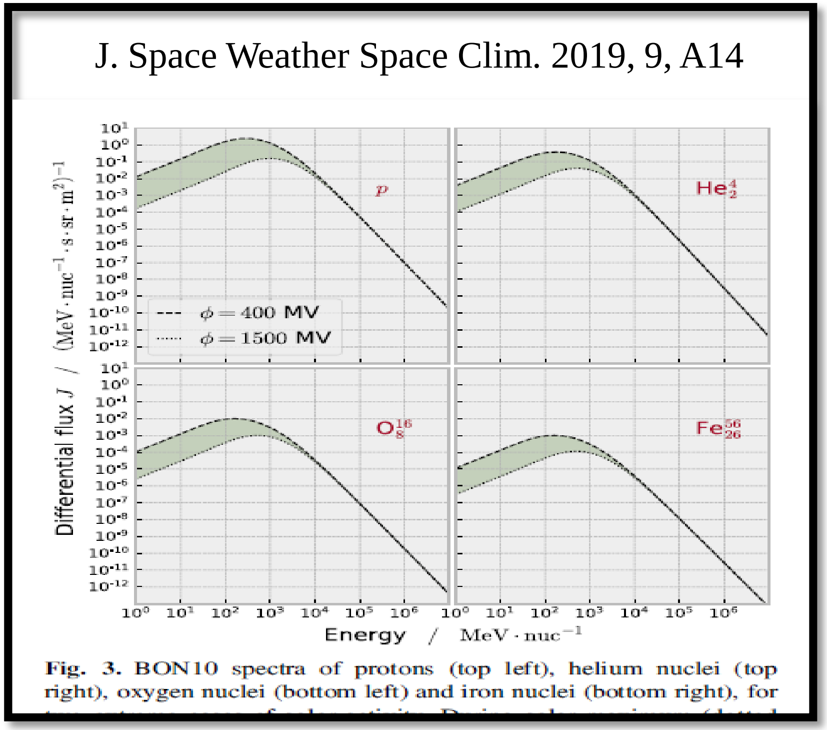
Fig. 16. Distributions of scattering angles between different reference detectors due to multiple scattering. The mean and rms values in μrad of each distribution are indicated in the plot.

Fig. 17. Simulated exclusive residual distribution for the zigzag GEM detector for 25 GeV/c pions. All detectors are set to have perfect intrinsic resolution here so that the width of this distribution is a measure of the MCS effect at the position of the zigzag GEM.

are used for the alignment. Any strip multiplicity is allowed in the clusters including single-strip clusters.

The first alignment step is to shift each of the four tracking detectors iteratively in the *XY*-plane to make their origins match each other in that plane. The initial shift parameters are mean values from position distributions in *X* and *Y* coordinates. In each iteration, straight lines are fitted to the hits in *X* and *Y*. Residuals are histogrammed for each detector and the residual distributions are fitted with a double-Gaussian function. Ten percent of the residual mean value of each detector is taken as the shift parameter in the next iteration to avoid overcorrections. The resulting residual mean values converge quickly towards zero after 40 iterations, as can be seen in Fig. 18. This provides a first coarse alignment. In a second alignment step, we correct also for relative rotations of the tracking

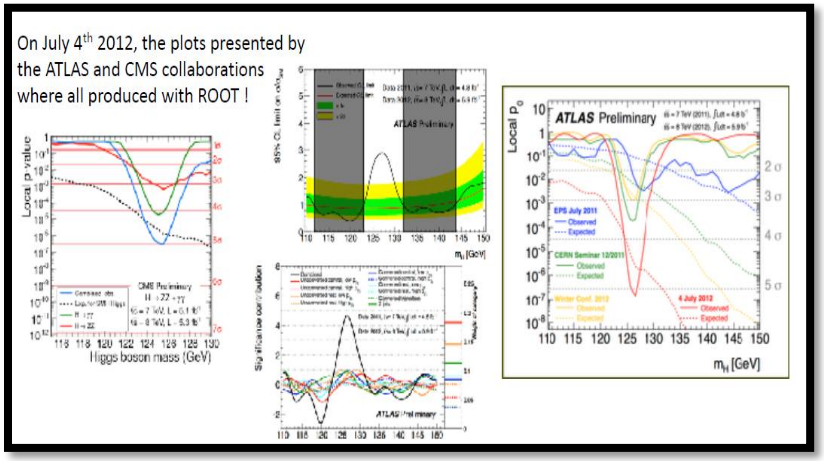
Geant4 paper for Space Applications



J. Space Weather Space Clim. 2019, 9, A14

Fig. 3. BON10 spectra of protons (top left), helium nuclei (top right), oxygen nuclei (bottom left) and iron nuclei (bottom right), for

Higgs Boson Discovery !



On July 4th 2012, the plots presented by the ATLAS and CMS collaborations where all produced with ROOT !

Geant4 for Proton Computed Tomography

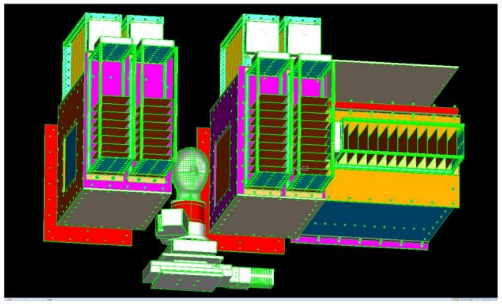
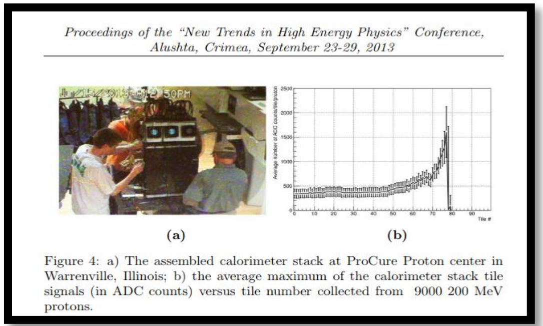


Fig.14. pCT Test Beam Setup



Proceedings of the "New Trends in High Energy Physics" Conference, Alushta, Crimea, September 23-29, 2013

Figure 4: a) The assembled calorimeter stack at ProCure Proton center in Warrenville, Illinois; b) the average maximum of the calorimeter stack tile signals (in ADC counts) versus tile number collected from 9000 200 MeV protons.

A Detector for Proton Computed Tomography G.Blazey, .K. Lalwani et al, FERMILAB-CONF-13-554-E

Many more Publications

References

Geant4:

www.geant4.cern.ch
https://geant4.web.cern.ch/support/training_courses

Geant4 Forum
https://geant4-forum.web.cern.ch/

Youtube lectures
https://www.youtube.com/watch?v=bk7eTGXq_K8

Our tutorials are available at Github here

Quiz

Q.1: Which volume is used to define the material in Geant4?

a) Physical b) Solid c) Placement b) None of them

Q.2: Which is the visualization driver mostly used in Geant4?

a) HepRep
b) DAWN
c) RayTracer
d) OpenGL

Q.3: Particle beam in Geant4 is generated using

a) Mandatory Classes
b) Optional Classes
c) a) and b) both are correct
d) none of them

Quiz

4. **From which class we can define a material directly?**
(a) G4AnalysisManager
(b) G4NistManager
(c) G4Element
(d) G4Material

5. **Which is not type of a Physics List?**
(a) G4VUserPhysicsList
(b) G4PhysicsList
(c) G4VModularPhysicsList
(d) G4PhysicsListFactory

6**. Which statement is correct?**
(a) a event is a collection of runs
(b) a step is a collection of runs
(c) a run is a collection of events
(d) a step is a collection of events

7. **Which class is used to define GPS?**
(a) G4Particle
(b) G4ParticleGun
(c) G4GeneralParticleSource
(d) G4Source

Hands-on Session

July 16, 2021