GEANT4:
Introduction and examples of use for accelerator physics

ARNAUD HUBER

IMPORTANT NOTE !!!

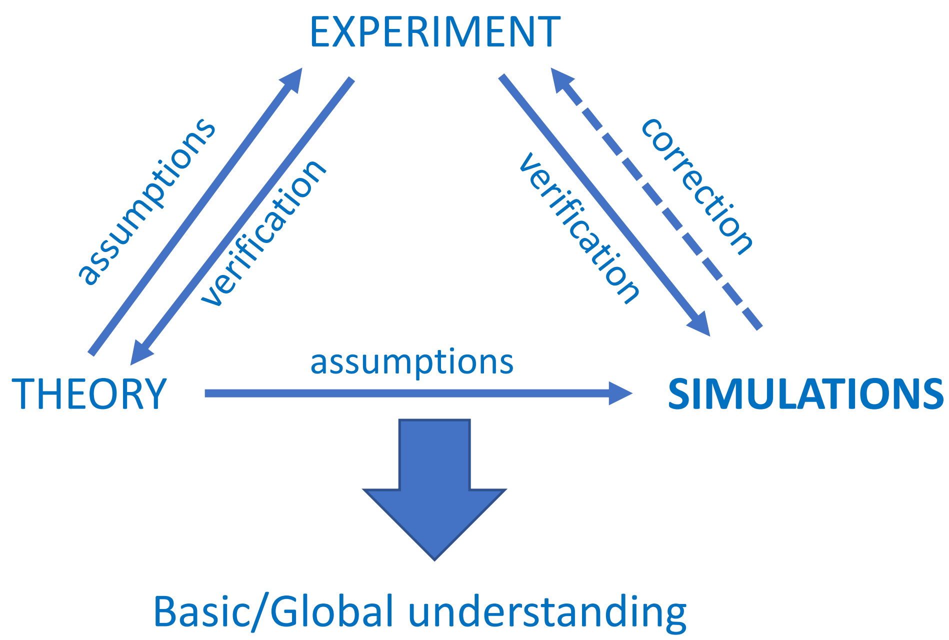
- **I'm not a developper of GEANT4**
- **Just an active user**
- **Many information taken from GEANT4 lectures formation**

RESUME

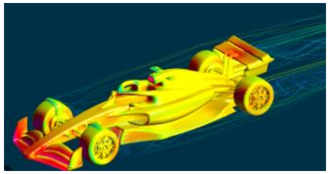
- **Introduction**
- **Monte-Carlo technique**
- **GEANT4**
- **Examples of use in accelerators physics**
- **Conclusions**

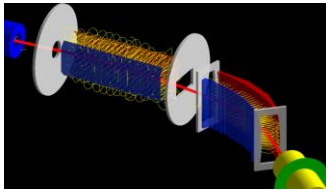
Introduction

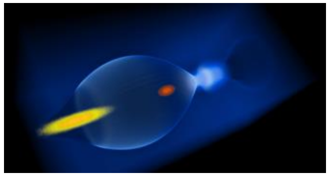
- **Simulations = essential part of the science today**



- **Simulations ?**
  - **Computational Fluid Dynamics (FLUENT, …)**
  - **Particle-In-Cell (SMILEI, …)**
  - **Monte-Carlo (GEANT4, …)**

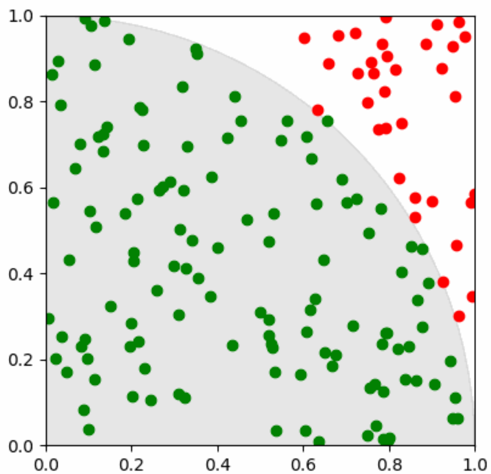


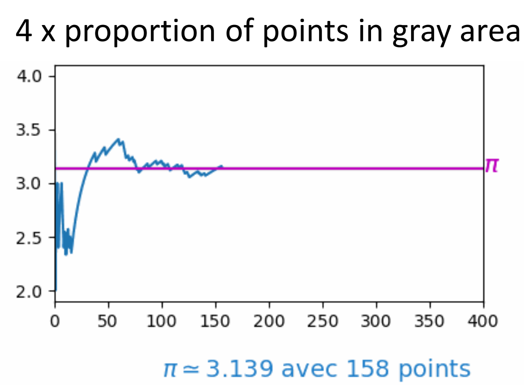




Monte-Carlo technique

- **Monte-Carlo ? :**
  - (computational) method that relies on the use of random sampling and probability statistics to obtain numerical results for solving deterministic or probabilistic problems until convergence is achieved
  - Give an approximate solution to a problem which is too big , too hard, too irregular for deterministic mathematical approach





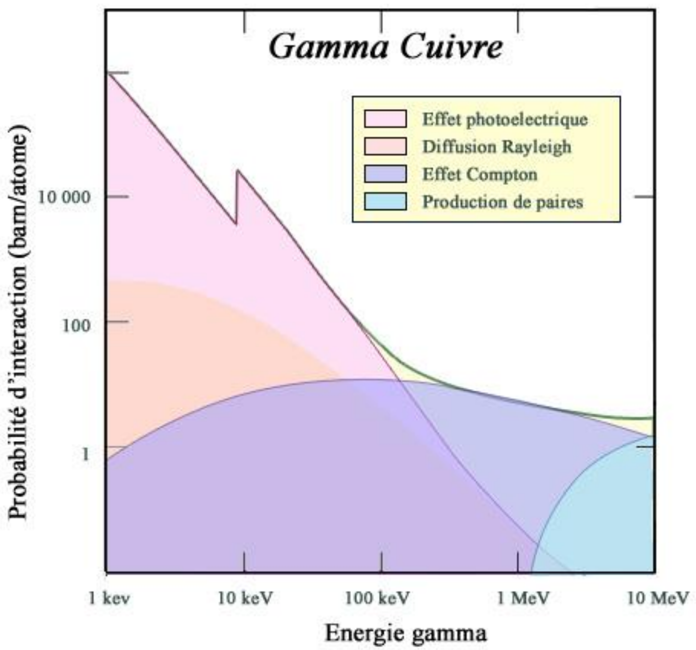
$$\pi\ estimation = \frac{Nb\ green\ point}{Nb\ red\ point}$$

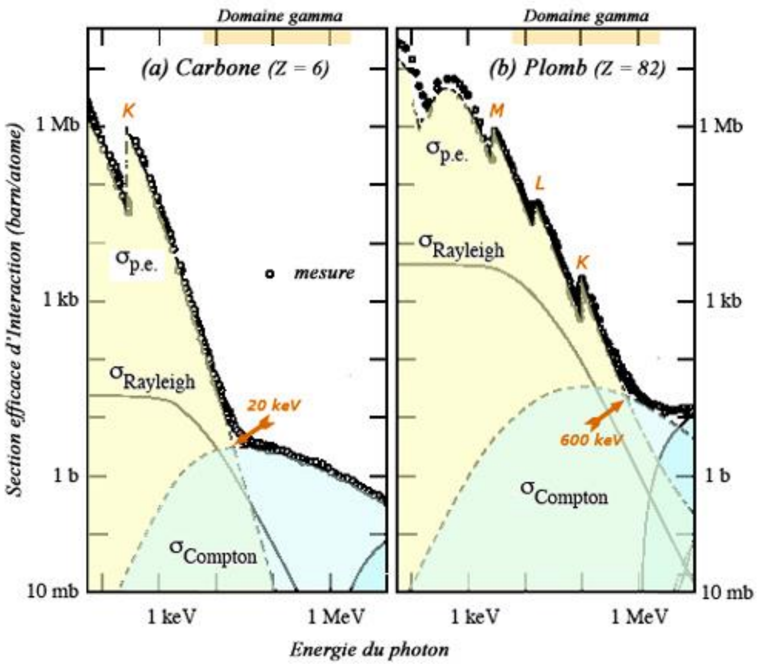
$$Area\ disk\ quarter = \frac{\pi}{4}$$

monteCarlo (education.fr)

Monte-Carlo technique

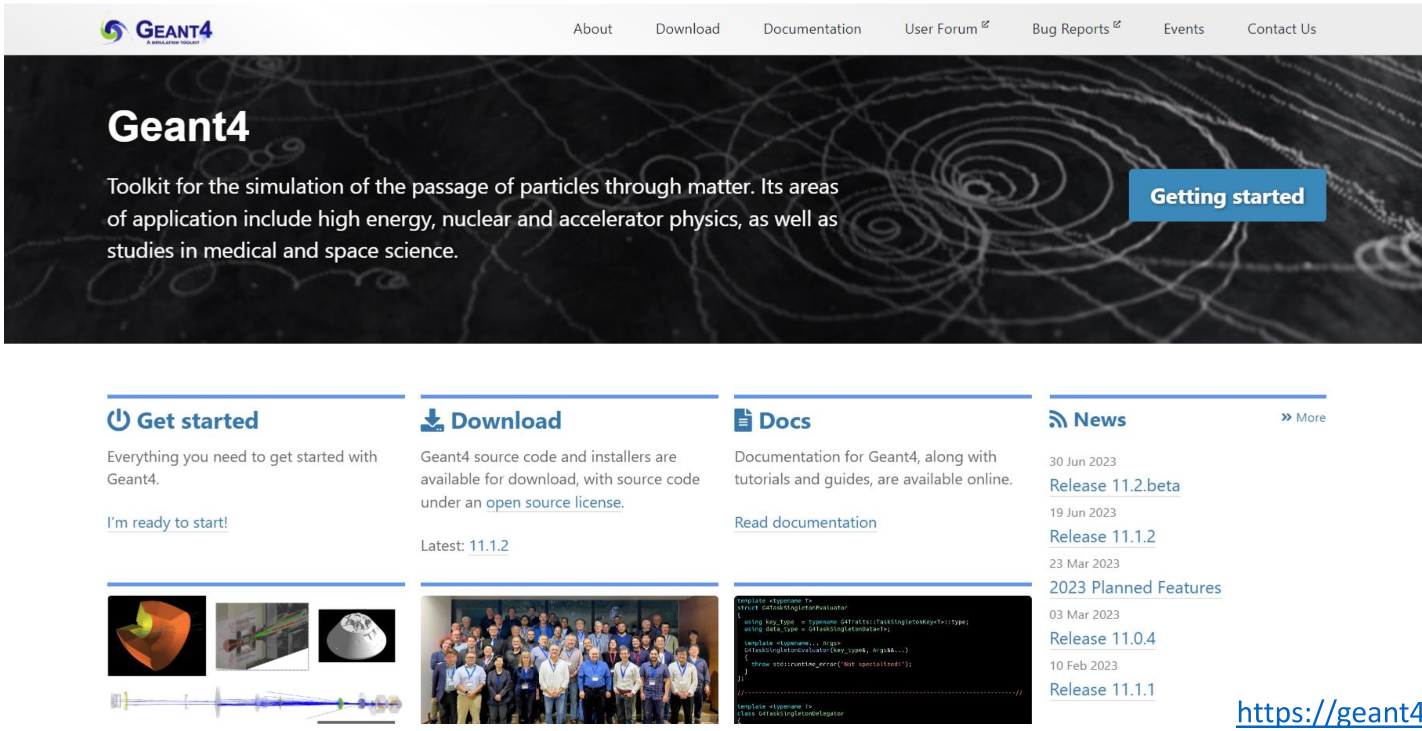
- **Monte-Carlo ? :**
  - (computational) method that relies on the use of random sampling and probability statistics to obtain numerical results for solving deterministic or probabilistic problems until convergence is achieved
  - Give an approximate solution to a problem which is too big , too hard, too irregular for deterministic mathematical approach





GEANT4

- **GEometry ANd Tracking**

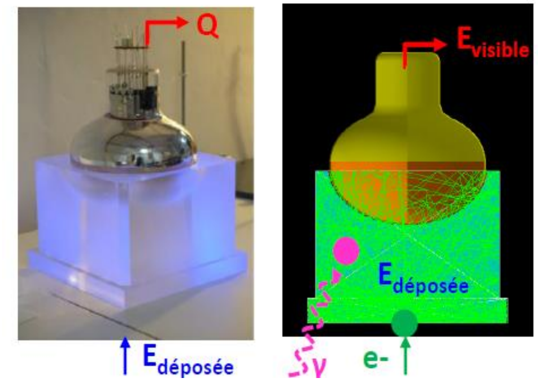


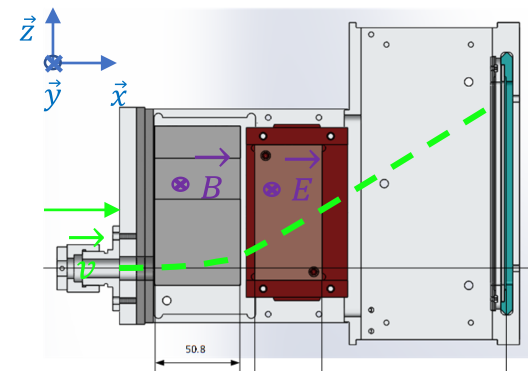
https://geant4.web.cern.ch/

- **Code, documentations, publications available on the web page**
- **Regular tutorial courses worldwide**
- **Active community forum (https://geant4-forum.web.cern.ch/)**

GEANT4

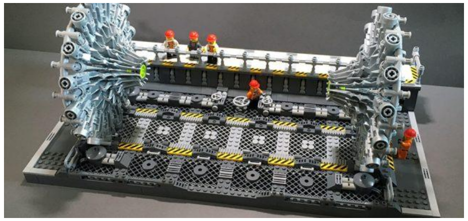
- **Overview :**
  - **C++**
  - **Object oriented**
  - **Open Source**
  - **Toolkit i.e. collection of tools**
    - **Geant4 defaults model DOES NOT EXIST !!!**
    - **MUST provide the necessary information, choose the GEANT4 tools**

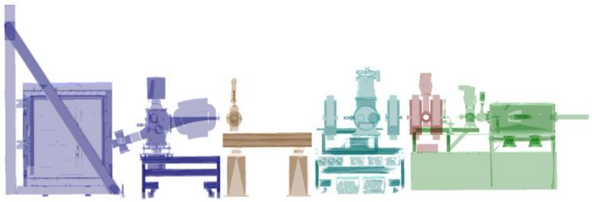




GEANT4 provides some features (building blocks)

Users construction in order to describe the problematic

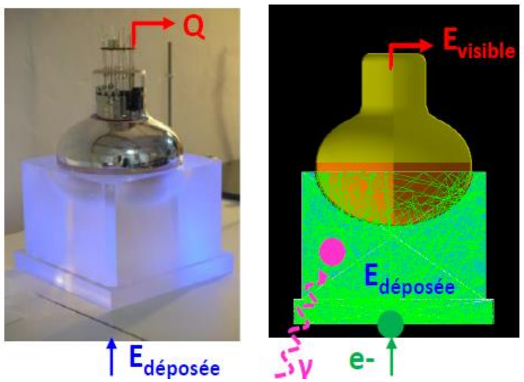


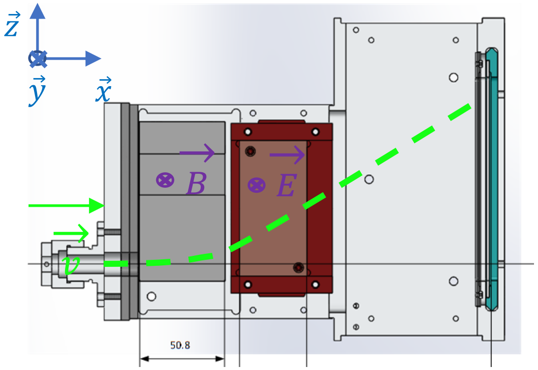


Many examples provided :

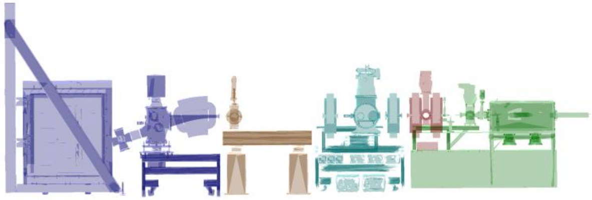
https://geant4-userdoc.web.cern.ch/Doxygen/examples_doc/html/index.html

- **Overview :**
  - **C++**
  - **Object oriented**
  - **Open Source**
  - **Toolkit i.e. collection of tools**
    - **Geant4 defaults model DOES NOT EXIST !!!**
    - **MUST provide the necessary information, choose the GEANT4 tools**





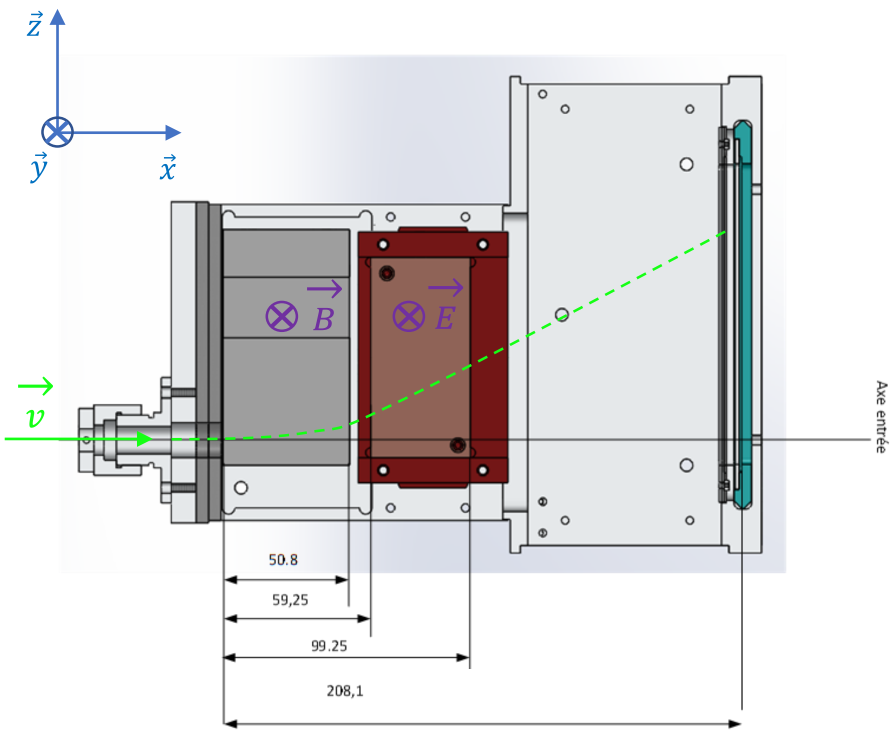
- **Basics :**
  - **Obligation :**
    - **Geometry information**
    - **Primary particles**
    - **Physics models**
  - **Possibility :**
    - **Visualization**
    - **Output files generation**

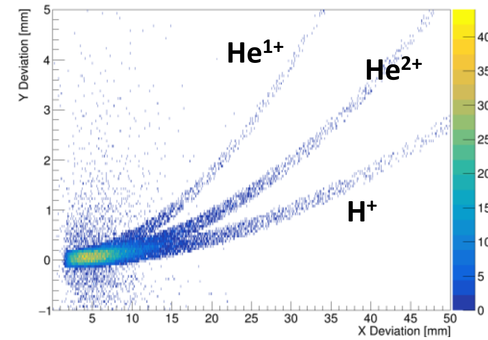


GEANT4 – Example 1

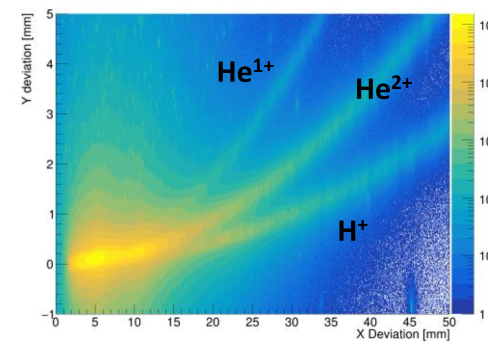
Thomson Parabola :

- Pre-experiment :
  - Useful to define the TP's parameters
- Post-experiment :
  - Useful for the TP's calibration

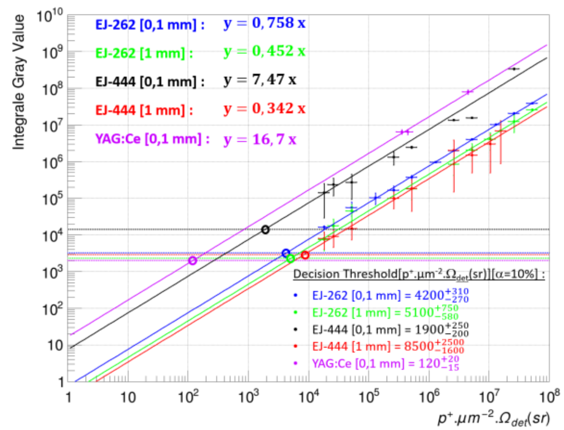




IP like



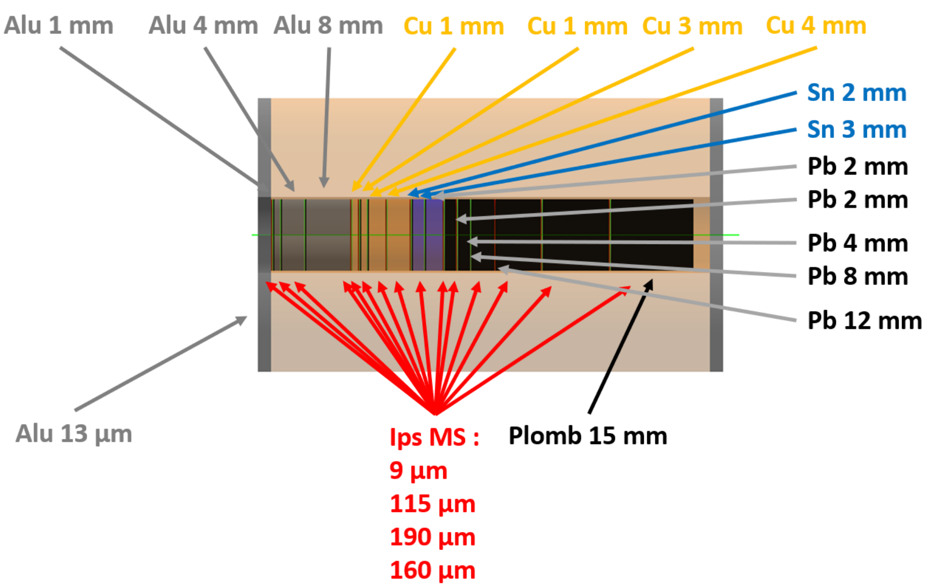
0,1mm plastic scintillator + CMOS

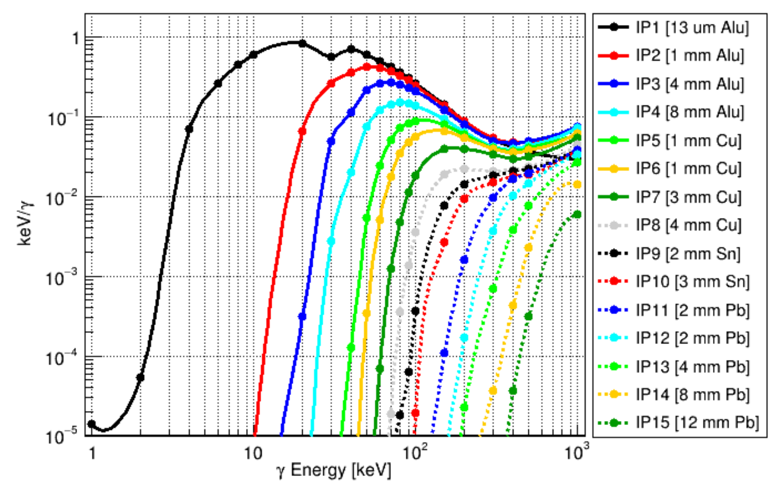


Bremsstrahlung cannon simulation :

- Pre-experiment :
  - Useful to define the filters according to the experimental parameters

- Post-experiment :
  - Reconstruction of initial spectrum with « a priori » assumptions

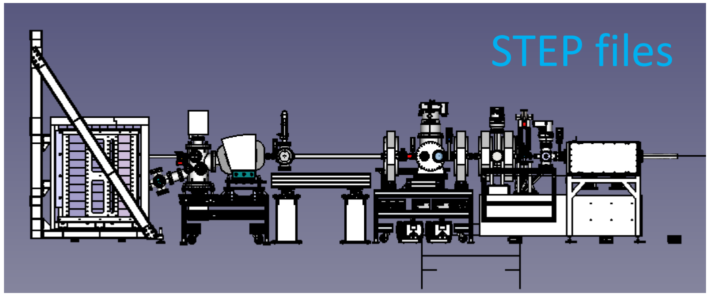




GEANT4 – Example 3

Collimator studies for the PALLAS experiment:

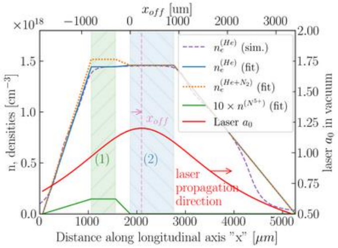
- **Define a collimator slits (in front of e- spectrometer) :**
  - **Clean the beam…**
  - **Without too much noise generation**
  - **Without material activation**



Massive fast PIC simulations *K.Cassou slide*

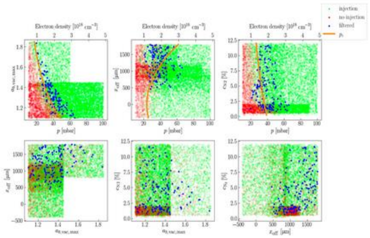
Using Smilei Smilei with envelope approximation and low particle per cell (Drobniak, Baynard, Bruni, et al. 2023):

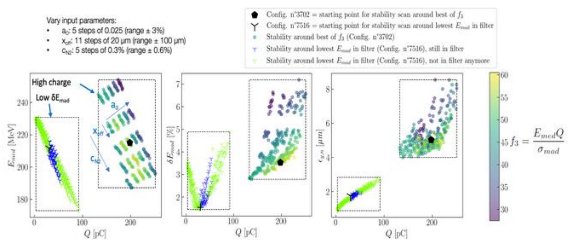
- 4 input parameters $[a_0, p, c_{N_2}, x_{off}]$
- 5 Random Scans (RS)
- 5 x 2401 = **12 005** configurations
- 5 x 4 h < 1 day



4 output parameters [E_{mean}, dE_{mean}, Q, epsb]

Unique dataset => surrogate model for beam parameter prediction (*V. Kubitskyi G. Kane*)





Simulations Team Work :

PIC simulations
Particles generation input
MC Simulations

ML model directly in GEANT4

GEANT4 – Other questions

- **Output files (analysis) ?**
  - **G4AnalysisManager tools**
  - **CSV, Txt, ROOT (Tree, Ntuples, …) files by user choice**

- **MultiThreading possible ?**
  - **Yes**
  - https://geant4-ed-project.pages.in2p3.fr/geant4-ed-web/docs/multithreading-1.pdf.fr

- **Easy to install ?**
  - **Yes with IN2P3 package Virtual Machine (https://geant4.lp2ib.in2p3.fr/)**

Download the current release of Geant4 Virtual Machine

Current Stable Release :

- **VM du 13/03/2024 :**
  Geant4.11.2.1=> PC Windows, Mac Intel processors, for Vmware Workstation/Fusion
  Geant4.11.2.1 =>PC Windows, Mac Intel processors, for VirtualBox
  Geant4.11.2.1 => Mac Apple silicon chips, for Vmware Fusion
  readme

Previous Releases :

- VM du 20/12/2023 :
  Geant4.11.2.0=> PC Windows, Mac Intel processors, (with Vmware Player/Fusion)

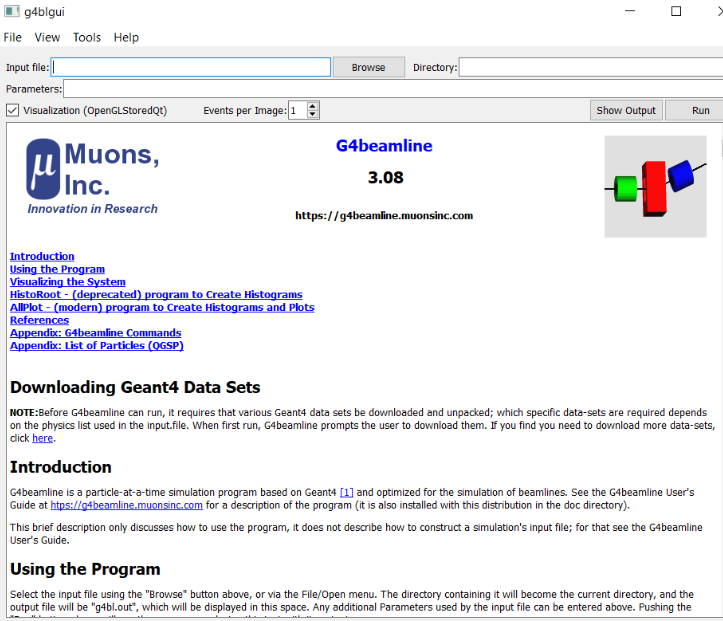
G4Beamline

- **Single-particle tracking program based on GEANT4**
- **Specifically designed for beamline simulation but can also be used for other systems**

- **Very friendly :**
  - **No C++ programming required**
  - **Linux, Windows, Mac**
  - **1 ASCII file (system & simulation)**
  - **Advance visualization capabilities**
  - **Plots & histograms easily generated**
  - **Common beamline elements already implemented**
  - **Well-documented Users Guide**

- **High realistic simulations :**
  - **Full power & accuracy of GEANT4**
  - **Any GEANT4 physics list can be used**
  - **Implementation of beamline elements details possible (ex : Field map)**

- **Already used by accelerators physicist :**
  - **> 500 people used it**
  - **MICE, Muon Accelerator Program, potential anti-proton experiments at Fermilab, …**



delerue@lal.in2p3.fr

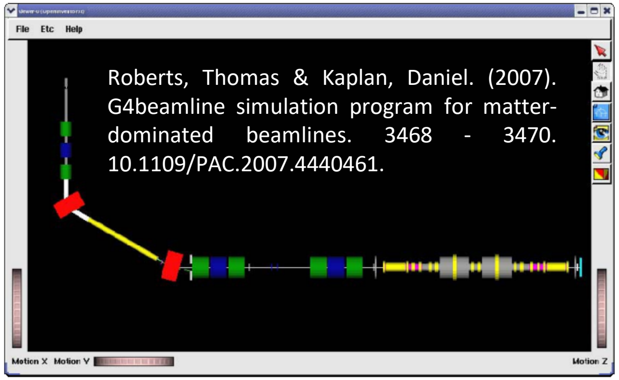
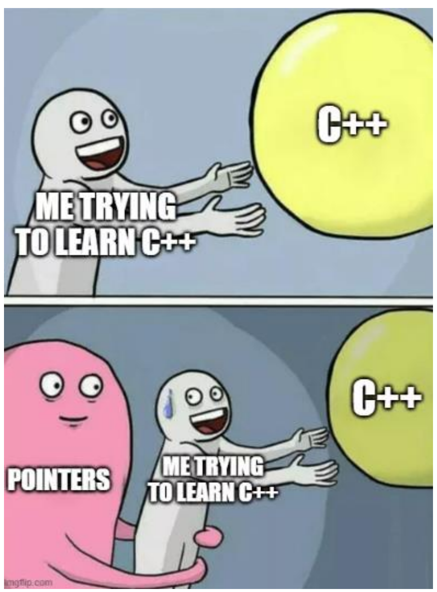
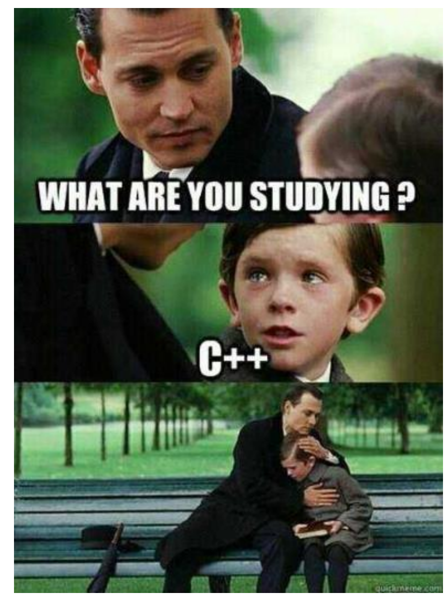


Figure 2: The MICE muon beamline and cooling channel [3] – a detailed and realistic simulation using G4beamline. Quadrupoles are green (HF) and blue (HD), dipoles are red, solenoids are yellow, beam pipes are gray, vacuum chambers are white, the two RF cavities are gray, and the calorimeter at the end is light blue. The pion production target is at the top left inside the ISIS synchrotron (not shown).

Conclusions

C++ programming is no longer an excuse to do GEANT4 simulations !!!

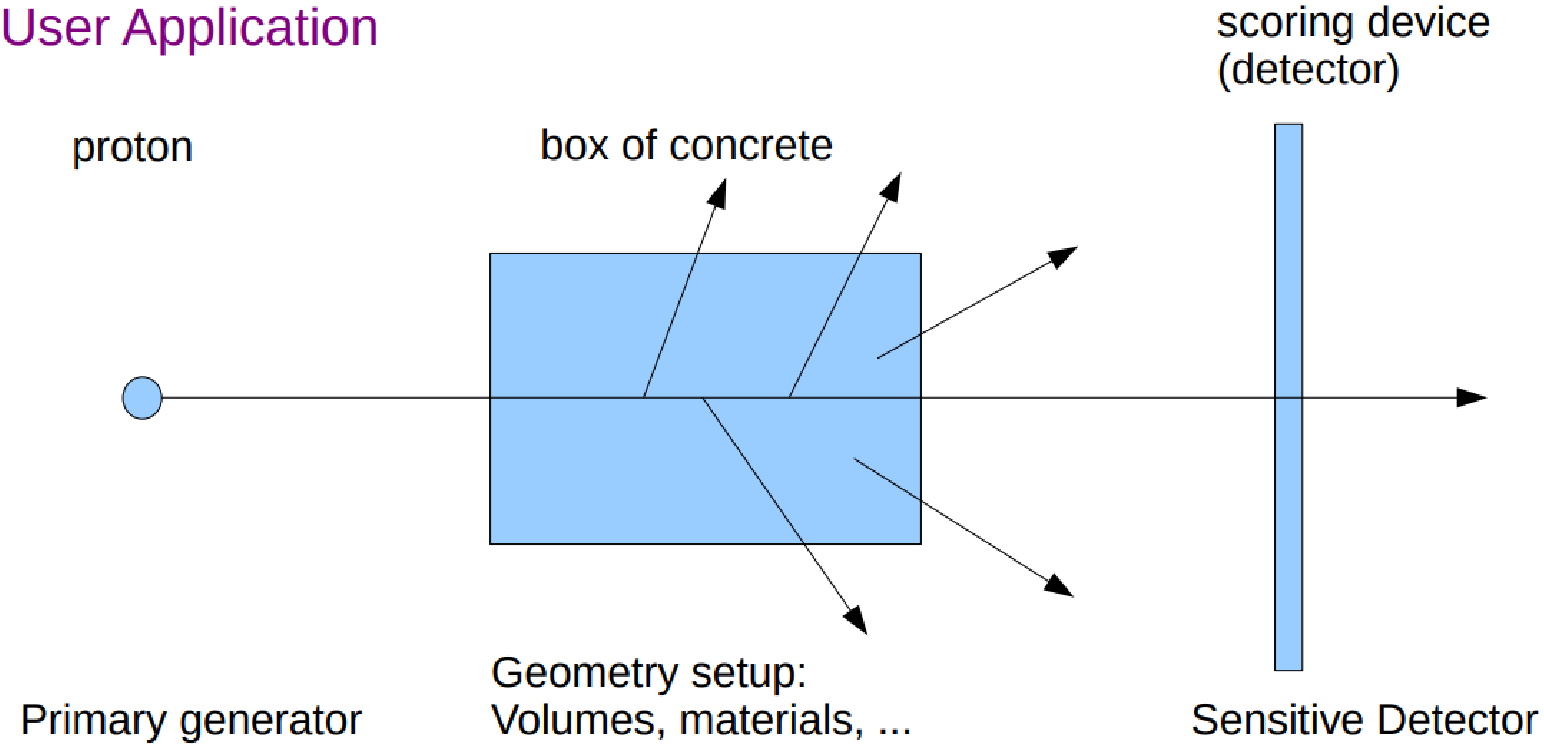




THANKS FOR YOUR ATTENTION

GEANT4 - kernel

User Application



Geant4

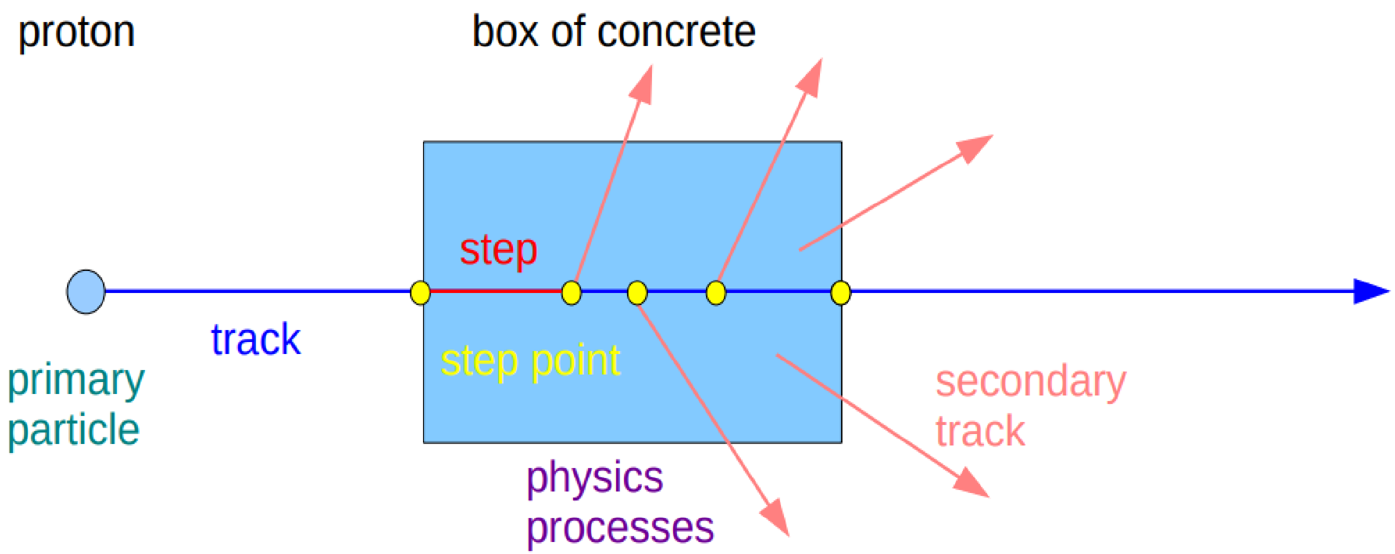
Users have first to define their experimental setup via Geant4 toolkit classes

I. Hrivnacova @ Geant4 IN2P3 and ED PHENIICS Tutorial, 2023, IJCLab

GEANT4 - kernel

User Application

scoring device
(detector)



Primary generator

Geometry setup:
Volumes, materials, ...

Sensitive Detector

Geant4

Geant4 then tracks the defined primary particles and let them interact with the materials present in geometry

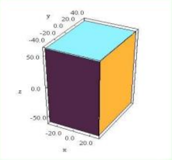
I. Hrivnacova @ Geant4 IN2P3 and ED PHENIICS Tutorial, 2023, IJCLab

GEANT4 - Geometry

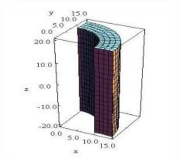
1) Start with its shape & size ➡ Solid
   - Shapes and sizes

     (Box 3 x 5 x 7 cm, sphere r = 8m)
2) Add properties: ➡ Logical volume
   - Material
   - Magnetic/electric
   - Make it sensitive
   - e.t.c
3) Place it in another volume ➡ Physical volume
   - Placement and rotation
     - Just once
     - Repeatedly

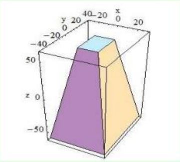
```
G4Box(const G4String& name, // name
      G4double hx,    // x half size
      G4double hy,    // y half size
      G4double hz);   // z half size
```



```
G4Tubs(const G4String& name, // name
      G4double rmin,  // inner radius
      G4double rmax,  // outer radius
      G4double hz,    // z-half length
      G4double sphi,  // starting Phi
      G4double dphi); // segment angle
```



```
G4Trd(const G4String& name,  // name
      G4double dx1,   // x half size at -dz
      G4double dx2,   // x half size at +dz
      G4double dy1,   // y half size at -dz
      G4double dy2,   // y half size at +dz
      G4double hz);.  // z half size
```



```
G4LogicalVolume(
  G4VSolid* solid,
  G4Material* material,
  const G4String& name,
  G4FieldManager* fieldManager = 0,
  G4VSensitiveDetector* sd = 0,
  G4UserLimits* userLimits = 0,
  G4bool optimise = true)
```

optional arguments

```
G4PVPlacement(
  G4RotationMatrix* rotation,         // rotation
  const G4ThreeVector& translation,   // translation
  G4LogicalVolume* currentLV,         // volume being placed
  const G4String& name,               // physical volume name
  G4LogicalVolume* motherLV,          // mother logical volume
  G4bool many,                        // not used
  G4int copyNumber,                   // position (copy) number
  G4bool surfaceCheckk = false);      // option to activate
                                      // overlap checking
```

Choosing a Physics List

- Which physics list to use depends on the use-case
- It is convenient and recommended to start with one of the reference physics lists
  - They are routinely validated and updated with each release
  - These should be considered only as starting points which you may need to validate for your needs
- If you need more specialized physics lists you may:
  - Use the G4PhysicsListFactory to build by physics constructor names (expert +)
  - Handle directly physics list with methods like (expert ++)
  - Write your physics constructor to implement your specialized process (expert ++++)
  - Write your own (expert n×'+', with n >> 1)
- There are currently **23** reference physics lists, of which **11** are used in production:
  - **FTFP_BERT, FTFP_BERT_HP, FTFP_BERT_ATL**
  - **QGSP_BERT, QGSP_BERT_HP**
  - **QGSP_BIC, QGSP_BIC_AllHP, QGSP_BIC_HP**
  - **Shielding, ShieldingLEND**
  - **NuBeam**

http://geant4.in2p3.fr/IMG/pdf_PhysicsLists.pdf

Geant4 PHENIICS & ANF IN2P3 Tutorial, 22 – 26 May 2023, Orsay

GEANT4 – Primary particles

Constructor (ie, called once)

```
MyPrimaryGeneratorAction::MyPrimaryGeneratorAction()
{
    G4int n_particle = 1;
    fparticleGun  = new G4ParticleGun(n_particle);

    // default particle kinematic
    G4ParticleTable* particleTable = G4ParticleTable::GetParticleTable();
    G4ParticleDefinition* particle = particleTable->FindParticle("gamma");
    fparticleGun->SetParticleDefinition(particle);
    fparticleGun->SetParticleMomentumDirection(G4ThreeVector(0.,0.,1.));
    fparticleGun->SetParticleEnergy(100.*MeV);
    fparticleGun->SetParticlePosition(G4ThreeVector(0.,0.,-50*cm));
}
```

Called at each event start

```
void MyPrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
{
    fparticleGun->GeneratePrimaryVertex(anEvent);
}
```

How to define our primary particle to shoot ?

1st example : ParticleGun

```
void G4ParticleGun::GeneratePrimaryVertex(G4Event* evt)
{
  if(particle_definition==0) return;

  // create a new vertex
  G4PrimaryVertex* vertex = new G4PrimaryVertex(particle_position,particle_time);

  // create new primaries and set them to the vertex
  G4double mass =  particle_definition->GetPDGMass();
  for( G4int i=0; i<NumberOfParticlesToBeGenerated; i++ ){
    G4PrimaryParticle* particle = new G4PrimaryParticle(particle_definition);
    particle->SetKineticEnergy( particle_energy );
    particle->SetMass( mass );
    particle->SetMomentumDirection( particle_momentum_direction );
    particle->SetCharge( particle_charge );
    particle->SetPolarization(particle_polarization.x(), particle_polarization.y(), particle_polarization.z());
    vertex->SetPrimary( particle );
  }

  evt->AddPrimaryVertex( vertex );
}
```

Sample code of G4ParticleGun clas
It is defined in geant4 : you don't ha
to provide it ! But just use it (see aft

Geant4 PHENIICS & ANF IN2P3 Tutorial, 22 – 26 May 2022, Orsay

GEANT4 – Primary particles

G4GeneralParticleSource (GPS)

- A more advanced implementation of G4VPrimaryGenerator
- It uses G4SingleParticleSource
  - Itself a G4VPrimaryGenerator
  - And which is an extended version of G4ParticleGun, allowing particles to be shoot according to distributions
- GPS Relies on the concept of "source"
  - The source emits the primary particles;
    - Of a given particle type
  - Sources can be combined with relative intensities to form a more advanced source.
    - Eg: built an Am/Be neutron + gamma source
- A source emits primary particles randomly according to
  - Position distribution
    - Ie the "source" distribution (point-like, surface, 3D…)
  - Energy, angular spectra
    - Built-in (uniform, exponential, gaussian, etc.)
    - Or user defined (providing an histogram-like data)
- Sources can be biased to enhance some phase space regions
  - And related statistical weight is provided

2nd example : General Particle Source

How to define our primary particle to shoot ?

```
MyPrimaryGeneratorAction::PrimaryGeneratorAction()
{
    fgps  = new G4GeneralParticleSource();
}

void MyPrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
{
    fgps->GeneratePrimaryVertex(anEvent);
}
```

Macro file commands:

/gps/particle proton	Shooting protons
/gps/pos/type Point	Point-like source
/gps/pos/centre 1. 2. 1. cm	Source position
/gps/ang/type iso	Isotropic source
/gps/energy 2. MeV	Protons energy