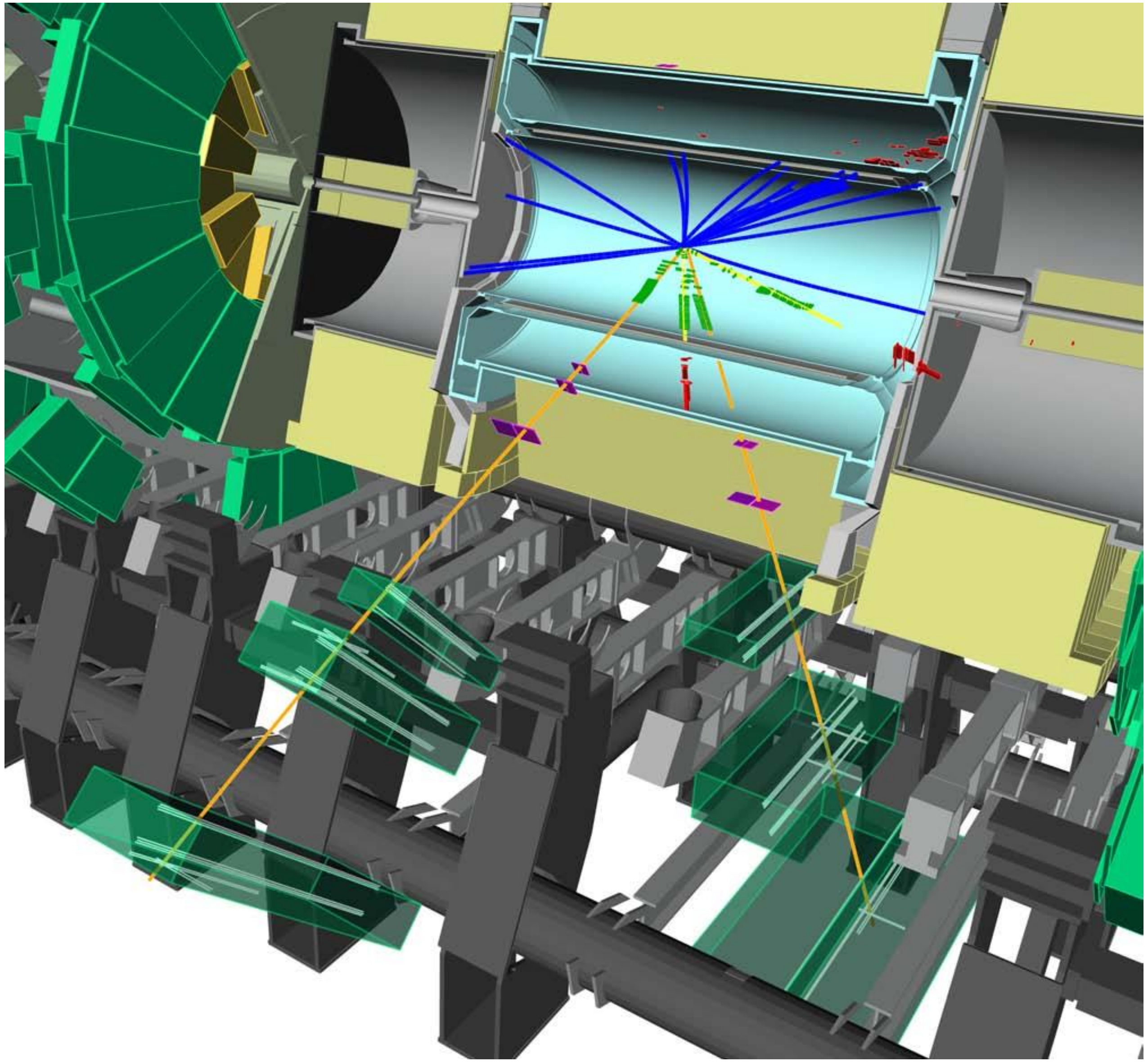
Simulation of large detector systems

Joe Boudreau
University of Pittsburgh

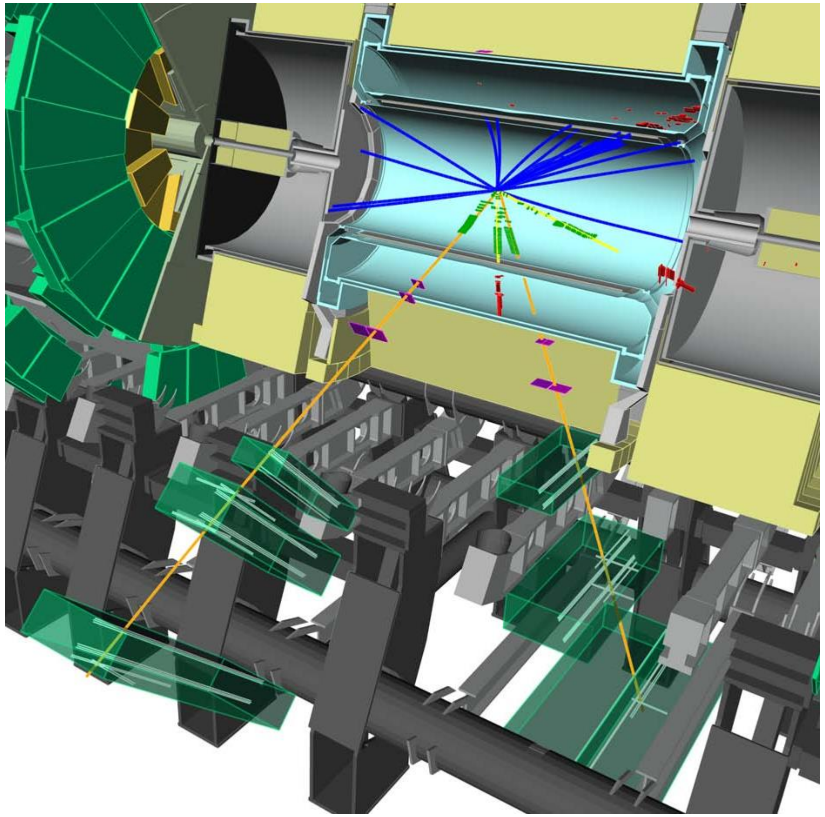
Software training workshop
Chandigarh India
Sep 11-15 2025

Simulation: a story.

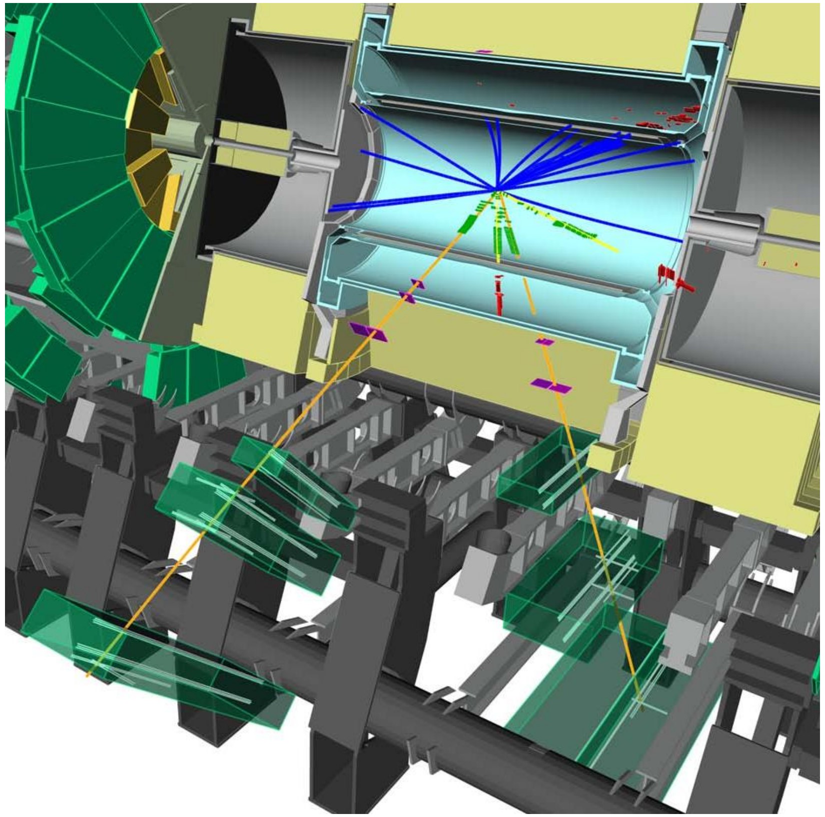
It's 2008.



Simulated digital signals left behind by a simulated particles from a simulated collision:



- The simulation starts with collision of two protons at a random position within the *beam spot*.
- Quarks, gluons within the proton have random relativistic momenta.
- Two such partons are chosen randomly.
- Their color & flavor are also chosen randomly.
- A fundamental process leading to multiple final state particles is chosen; the angles and energies of these particles also chosen at random, the probability is computed.
- In the case of this event the process is $H^0 \rightarrow Z^0 Z^0 \rightarrow e^+ e^- \mu^+ \mu^-$ The kinematics follows distributions computed from Feynman diagrams and other techniques, which are *sampled*.
- The final state particles, electron and muon, are propagated in a deterministic way through a magnetic field until material is encountered.



Simulated digital signals left behind by a simulated particles from a simulated collision:

- The muons, being unstable, are given an opportunity to decay.
- Before that happens they, like the electrons, will interact with the material:
  - Continuously as they traverse.
  - During discrete encounters at random positions.
- The energy that they leave behind is the basis for simulated detector response.
  - Green dots along the yellow lines: straw tracker hits.
- Electrons incident on heavy material in the calorimeter undergo further fundamental processes that quickly multiply the number of particles. Their energy and emission angles are sampled from probability distributions.

Expected Performance of the ATLAS Experiment
Detector, Trigger and Physics
The ATLAS Collaboration
CERN-OPEN-2008-020
ISBN 978-92-9083-321-5
December 2008
Volume I: Performance
Volume II: Standard Model, Top Quark, B-Physics
Volume III: Higgs Boson, Supersymmetry
Geneva
2009

The analysis of simulated data fills three volumes.

… an 1800 page study of synthetic LHC data.

- Many of the simulated particles are produced in abundance today.
- Others have been excluded by dedicated searches.
- Much of the work also appears in Ph.D theses.
- The impact of this work: *the computing infrastructure was immediately ready for data.*

Detector Simulation: Why It Matters

01

Design of Experimental Apparatus

1) Evaluate physics performance of different choices (subsys, integrated system).
2) Follow aging effects and other failures as experiment goes into operation.

02

Design and support Physics Analyses

1) Develop new analysis strategies before real data is available.
2) Debug, characterize, and optimize a measurement on simulated data (blind analysis) before actual measurement.

03

Development of reconstruction algorithms.

Design algorithms for track reconstruction, jet reconstruction, missing ET reconstruction. Evaluate efficiency, resolution, resource requirements. Design trigger algorithms, evaluating efficiency and required bandwidth. Design for CPU, GPU, FPGAs, new platforms.

04

Test of processing chain from trigger to data distribution.

The platform reconstructing, distributing, and analyzing the multi-petabyte dataset is very large computing infrastructure ("the grid"). It is stress-tested before datataking. Even the data acquisition system has been tested with simulated events!

05

Interpretation of data

The simulation chain (including also event reconstruction) reproduces the convolution of detector response with physics and its dependency on fundamental parameters (mass, width, coupling, new particles) is to be compared to data.

06

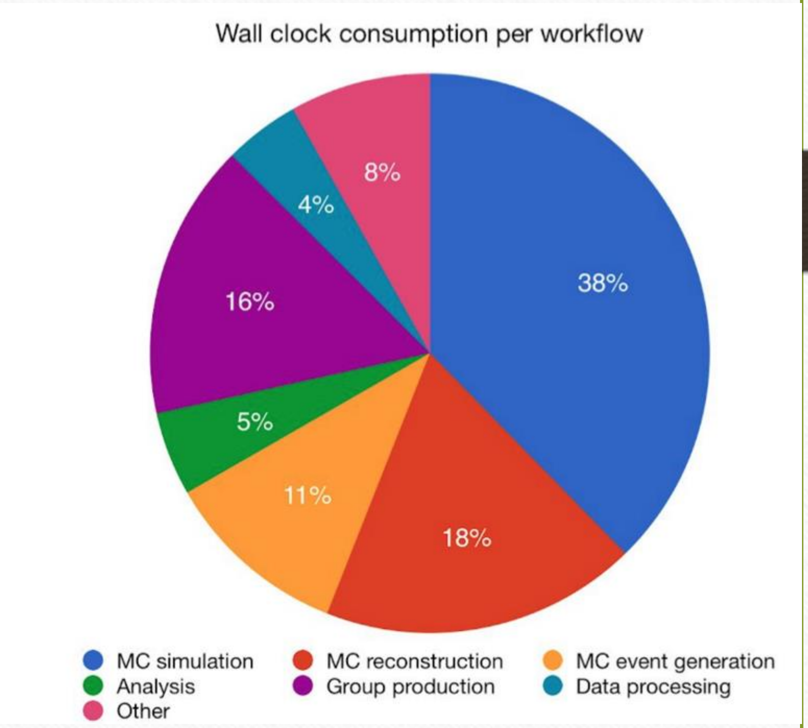
Estimation of Systematic Uncertainties

Uncertainties on the factors such as detector response or physics modeling that affect analysis are evaluated using simulation.

Simulation plays a critical role over the full life-cycle of an experiment.

Cost of Detector Simulation

- An event in ATLAS takes between 60s and 90s to simulate.
- Each year we generate about 45 billion events.
- This adds up to a substantial fraction of the computing resources ➔
- A rough estimate of the cost: $25 Million/Year (ATLAS alone).



ATLAS CPU hours used by various activities in 2018 [link]

Properties

More than 1M cores are used for simulation in ATLAS.

The performance and stability of the simulation code is continuously scrutinized and maintained by a group of dedicated experts.

Apart from some high-level dressing in python, workhorse for simulation is C++ code.

Development of this code is a partnership between physicist, detector physicists, and software engineers.

Simulation runs on CPU farms running the linux operating system.

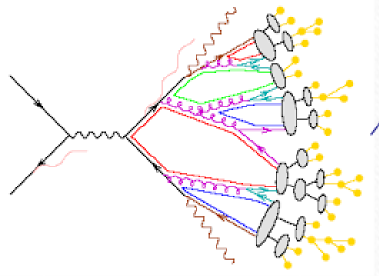
The operating system version is updated very infrequently for stability.

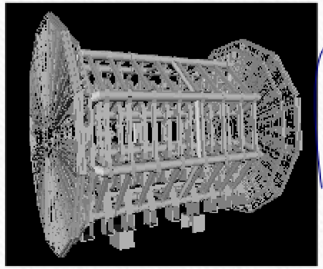
Consequence: little opportunity for hands-on work in this training workshop, mostly it will be my hands doing the work in front of you….

There is always a need for more people to help with simulation in ATLAS (I assume also in CMS LHCb ..)

But more on how to train yourself for a role in simulation is coming in subsequent slides.

Anatomy of a simulation





Physics List:
*
*
*

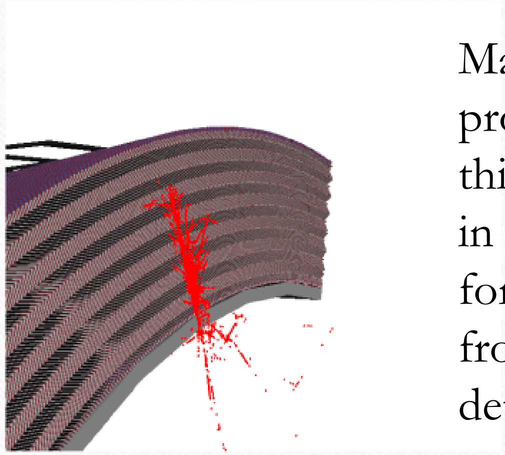
Cross sections,
Decay rates
Branching fractions..

Inputs

Simulation

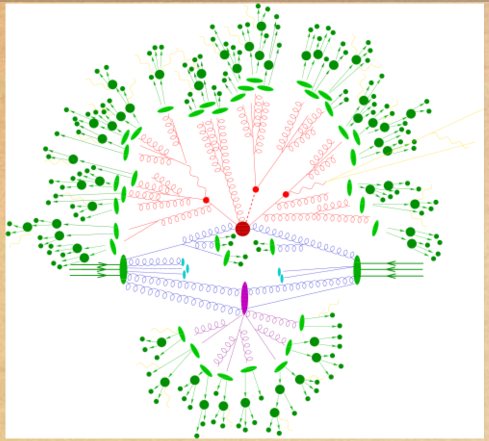
Customizations

Energy deposits (hits)
Electronic signals (digits)
Performance metrics (scoring)
…

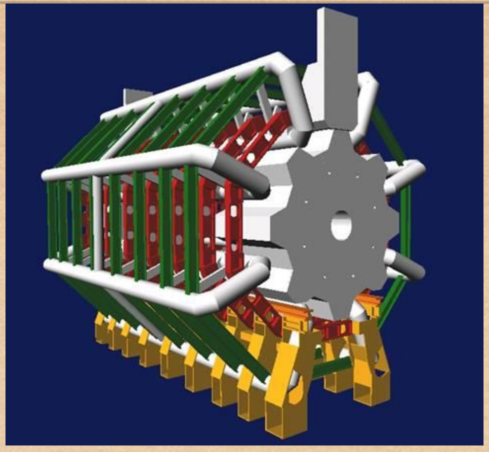


Mature sim. programs output this information in the same format as that from the detector.

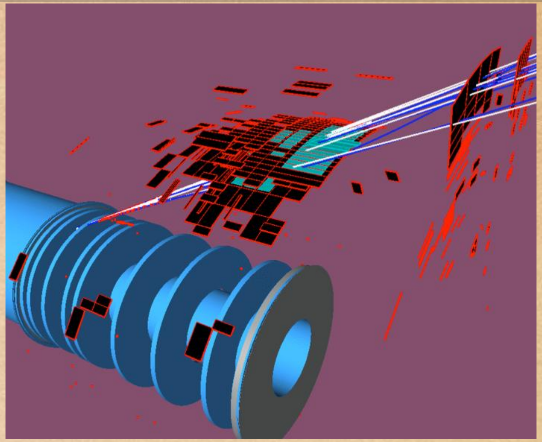
What goes into the simulation?



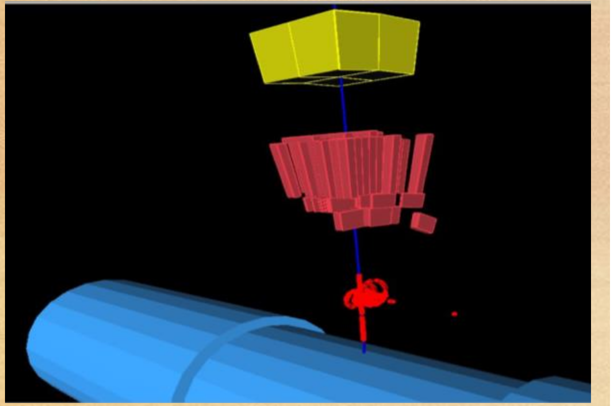
1. Simulation of a pp collision. Based on quantum field theory, developed by theorists.



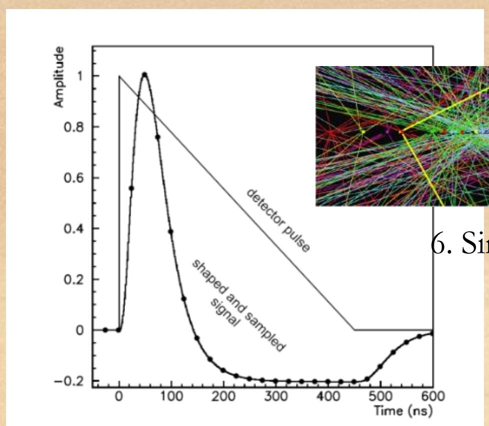
2. Complete and detailed software description of the detector geometry



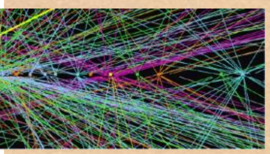
3. Charged particle propagation, collision detection, and a complete library of physics processes: e.g. showering, scattering, and energy loss.



4. Simulation of particle interactions w/ sensitive elements of the detectors

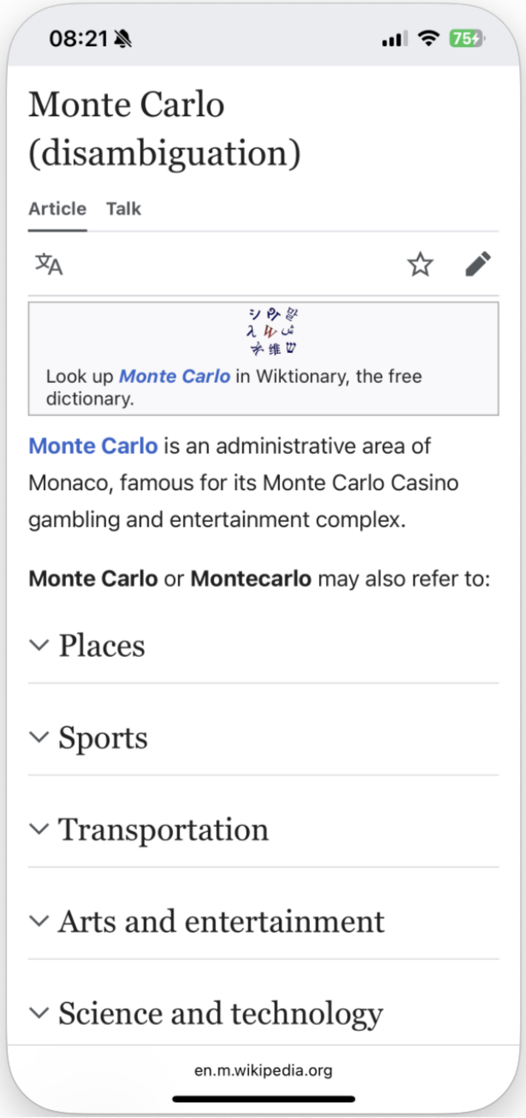


5. Simulation of electronics response



6. Simulation of *pileup*.

7. Simulation of the trigger



Monte Carlo (disambiguation)

Look up *Monte Carlo* in Wiktionary, the free dictionary.

Monte Carlo is an administrative area of Monaco, famous for its Monte Carlo Casino gambling and entertainment complex.

Monte Carlo or **Montecarlo** may also refer to:

- Places
- Sports
- Transportation
- Arts and entertainment
- Science and technology

en.m.wikipedia.org

What is Monte Carlo?

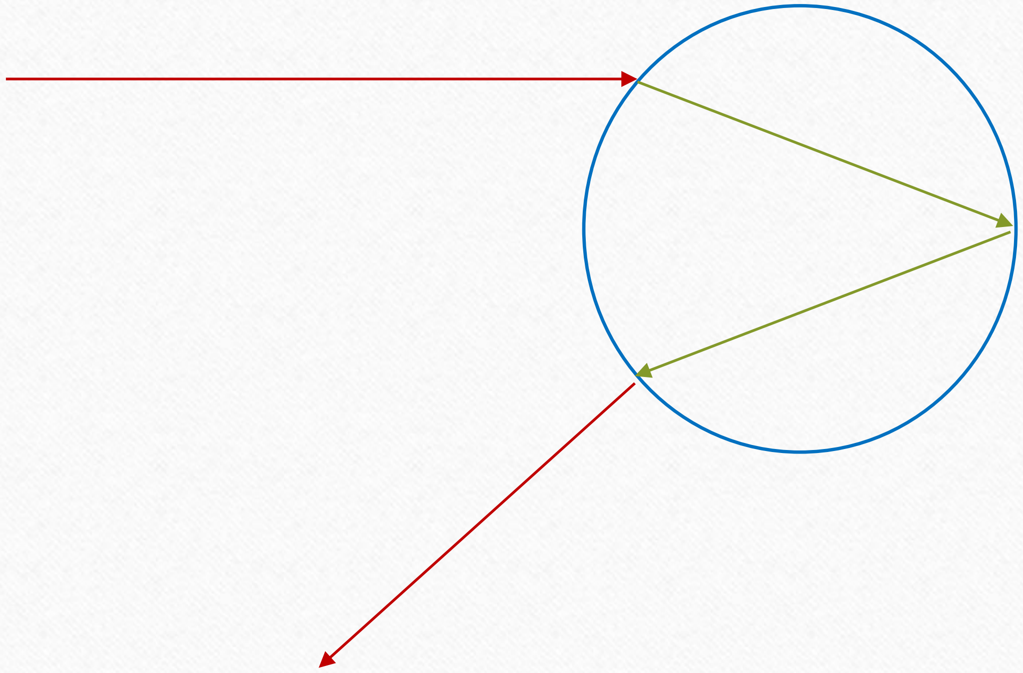
- Viewpoint 1: Monte Carlo is a technique for performing numerical quadrature using sequences of random numbers.
- Viewpoint 2: Monte Carlo is simulation of a system that evolves through a series of random processes.
- Holistic viewpoint: When simulating a system which evolves through a series of random processes, one is secretly computing the value of an integral, generally speaking without knowing which integral is being computed.
- To illustrate, let's look at a problem in radiation transport.

Example: scattering of light from a water droplet.

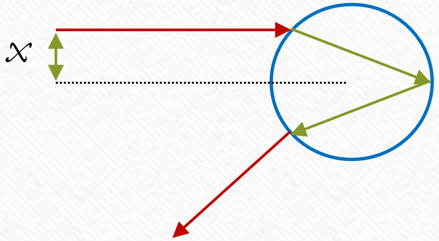
Compute the cross section for light scattering after a refraction, a reflection, and another refraction for a light wave of given wavelength that strikes a droplet of water.

(Owing to the dispersion of water, this is the process on which a rainbow depends).

In other words, compute the cross section for the incident light to appear in the rainbow.



Analytically



$$d\sigma = 2\pi x dx$$

differential cross section for anything to happen.

$T(x|n)$: transmission coefficient

$R(x|n)$: reflection coefficient

differential cross section for scattering into the rainbow

$$d\sigma\left(n\right) = 2\pi\left[T\left(x|n\right)\right]^2 R\left(x|n\right) x dx$$

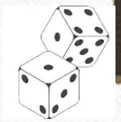
$$\sigma\left(n\right) = \int_0^a 2\pi\left[T\left(x|\, n\right)\right]^2 R\left(x|n\right) x dx$$

not an easy integral…

Monte Carlo Method

- generate a photon uniformly over a circular region of radius *a*
- calculate the angle of incidence at the surface of the raindrop.
- Calculate the reflection coefficient; throw the photon away if it is reflected.

- Follow the photon to the back surface, compute the new reflection coefficient, this time *require* reflection.



- Follow the photon to the front surface, compute the reflection coefficient, throw the photon away if reflected.

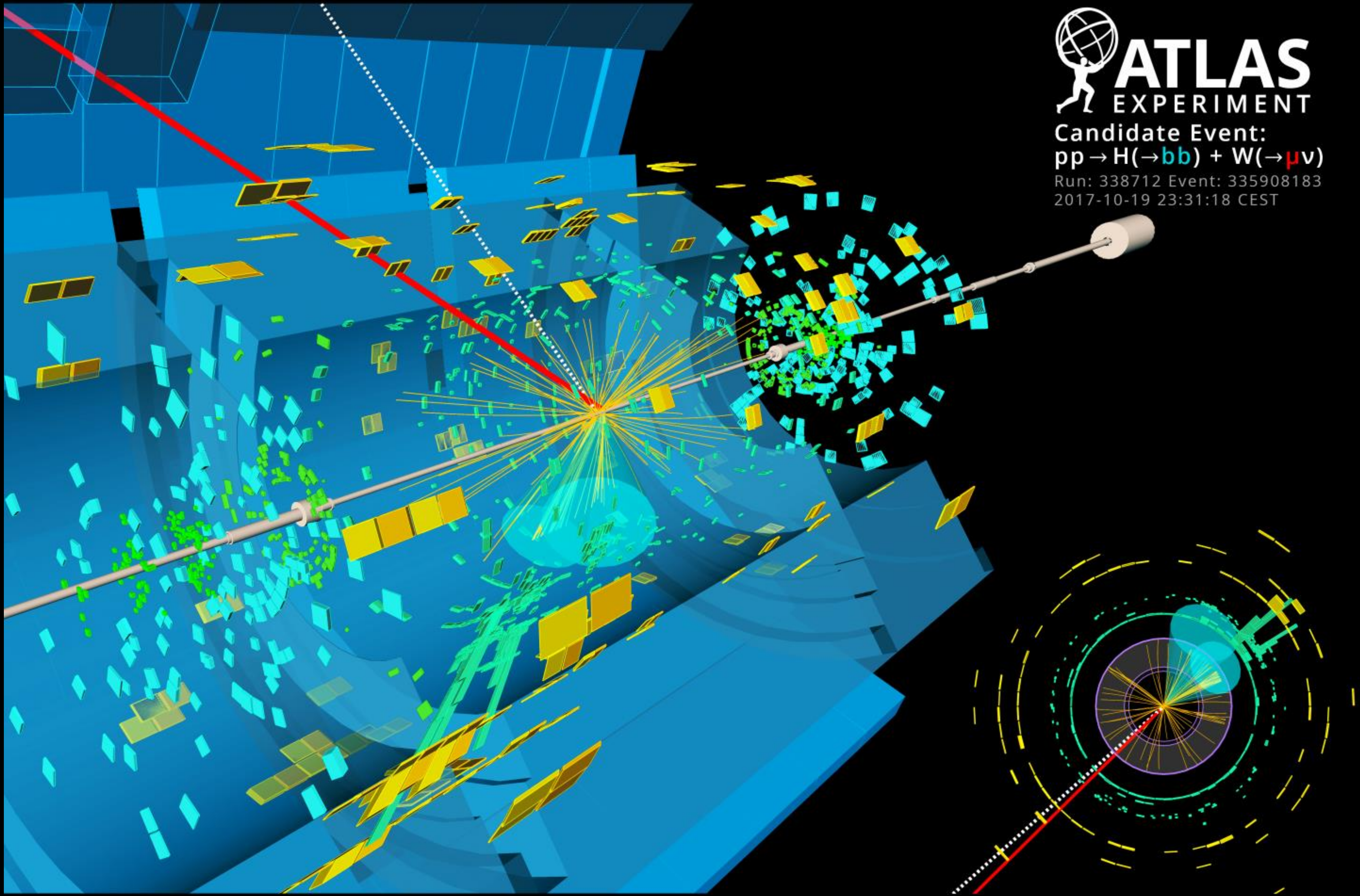
- The cross section is the area cross sectional area of the raindrop times the fraction of photons remaining.

$$\sigma\left(n\right)=\int_{0}^{a}2\pi\left[T\left(x\right|n)\right]^{2}R\left(x|n\right)xdx$$

$$\begin{aligned}\left[T\left(x|n\right)\right]^{2}R\left(x|n\right) &= \frac{8\left(1-x^{2}\right)\left(n^{2}-x^{2}\right)}{\left(\sqrt{1-x^{2}}+\sqrt{n^{2}-x^{2}}\right)^{2}\left(n\sqrt{1-x^{2}}+\sqrt{1-x^{2}/n^{2}}\right)^{2}}\times\\ &\left\{\left(\frac{\sqrt{1-x^{2}}-\sqrt{n^{2}-x^{2}}}{\sqrt{1-x^{2}}+\sqrt{n^{2}-x^{2}}}\right)^{2}+\left(\frac{n\sqrt{1-x^{2}}-\sqrt{1-x^{2}/n^{2}}}{n\sqrt{1-x^{2}}+\sqrt{1-x^{2}/n^{2}}}\right)^{2}\right\}\end{aligned}$$

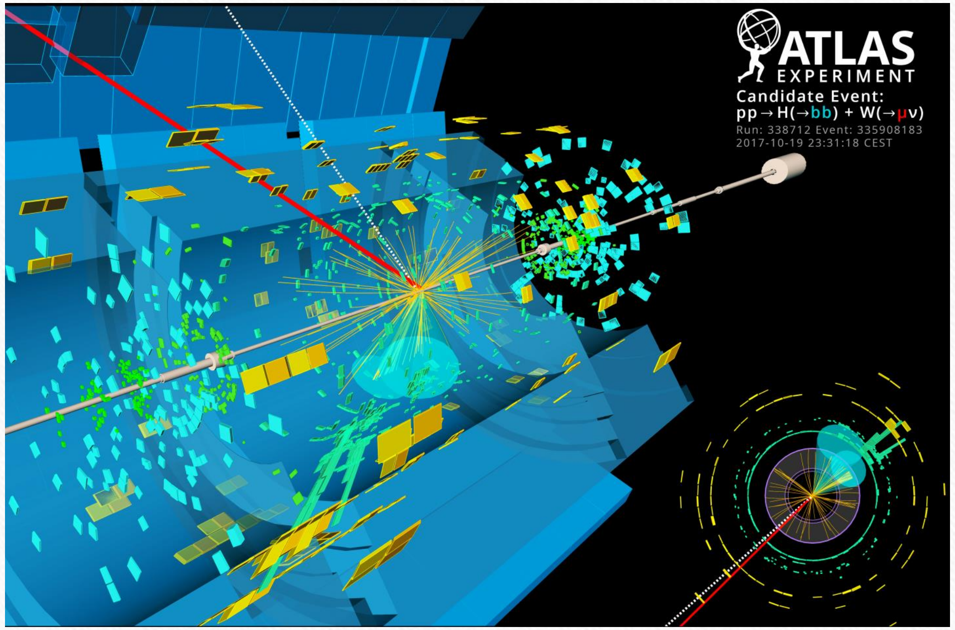
Even formulating this integral is difficult for this simple now imagine a more complicated setup….

NEXT



ATLAS
EXPERIMENT

Candidate Event:
$pp \to H(\to bb) + W(\to \mu\nu)$
Run: 338712 Event: 335908183
2017-10-19 23:31:18 CEST



When you calculate the rate of any process through simulation you are integrating.

The integral will never every be formulated.

However certain formal considerations of general Monte Carlo methods are useful to be aware of:

1. Order of convergence is ½. This means that the error on the integral goes as $1/\sqrt{N}$
2. Secondly,….

$$\bar{x} = \int f(x)\,\rho(x)\,dx$$

The error on the Monte Carlo estimate of this quantity is something divided by $\sqrt{N}$. You can control that "something"….

$$\sigma(n) = \int_0^a 2\pi\,[T(x|\,n)]^2\,R(x|n)\,x dx$$

By deciding which piece of this integral is the normalized probability distribution and which is the $f(x)$

choice 1

$$f(x) = 2\pi\,[T(x)]^2\,R(x)\,ax$$
$$\rho(x) = 1/a$$

choice 2

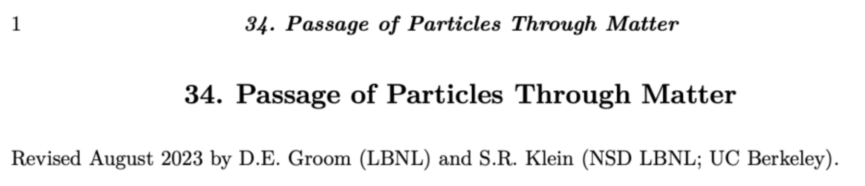
$$f(x) = 2\pi\,[T(x)]^2\,R(x)\,\frac{a^2}{2}$$
$$\rho(x) = 2x/a^2$$

choice 2 ("unweighted") has lower variance than choice 1 ("weighted")

Principal Physics Processes

- Ionization energy loss.
- Bremsstrahlung
- Electron-positron pair production
- Multiple Coulomb scattering
- Hadronic interactions with nuclei
- Particle decays

from pdg.lbl.gov:



1 *34. Passage of Particles Through Matter*

34. Passage of Particles Through Matter

Revised August 2023 by D.E. Groom (LBNL) and S.R. Klein (NSD LBNL; UC Berkeley).

Ionization Energy Loss

- Applies to charged particles but not neutral particles.
- Arises from the interaction between particle and electrons in the material.
- Can give rise to detectable signals in gas, noble liquids, scintillator, silicon, the origin of most charged particle detection technology.

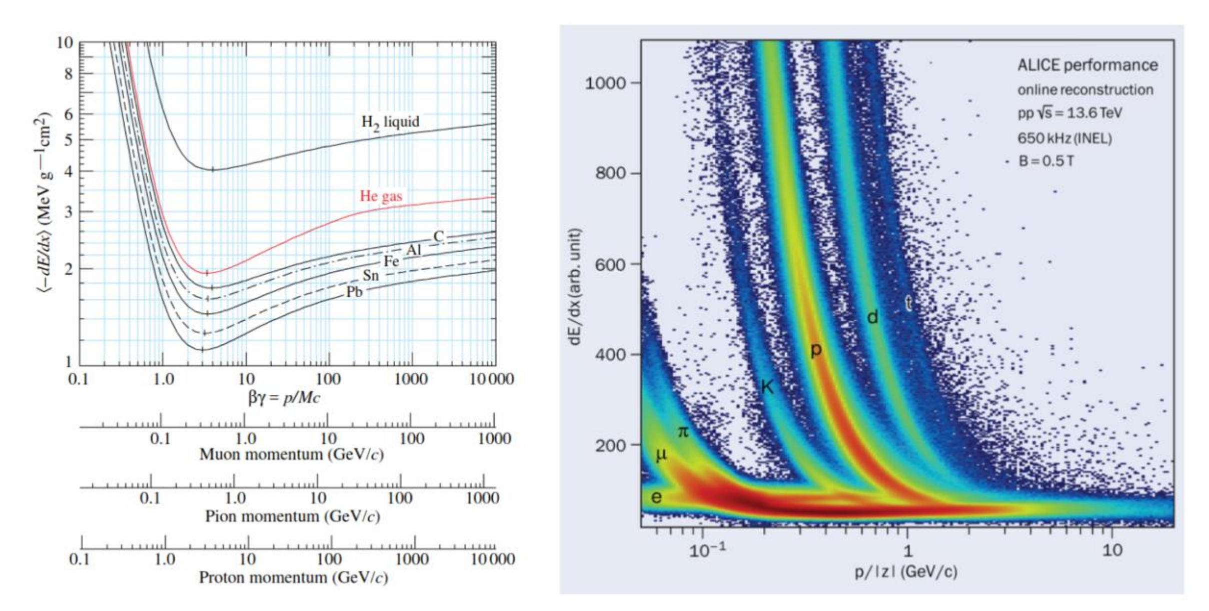
Follows the Bethe-Bloch formula:

$$\langle -\frac{dE}{dx} \rangle \;=\; 2\pi r_e^2 m_e \left[\rho \frac{N_A}{A} \right] \frac{Zz^2}{\beta^2} \left[\frac{1}{2} \ln \frac{2 m_e \beta^2 \gamma^2 W_{max}}{I^2} - \beta^2 - \delta\left(\beta\gamma\right) \right]$$

- Fluctuations about the mean described by Landau distribution.

r_e	classical electron radius
m_e	electron mass
ρ	mass density
N_A	Avogadro's number
A	Atomic mass
Z	Atomic number
β, γ	Velocity parameters
W_{max}	Maximum kinetic energy
I	Effective ionization energy
δ	Correction term

Ionization energy loss



Depends on type of charged particle.

Depends strongly on the density of the material, but also electronic structure.

Determinant factor for:

- efficiencies of detector systems.
- range of soft particles in matter.
- charged particle identification.

Bremsstrahlung

- Bremsstrahlung applies to the electron, also very high energy muons.
- It involves collisions with the nucleus, not the electrons, in the material.
- As in a collision between ping-pong ball and bowling ball, the electron suffers a large change in direction, the nucleus is little affected.
- The represents accelerated charge and radiation is emitted.
- The effect depends on the radiation length X_0 with a **strong dependency on nuclear charge Z**, signifying the distance over which an electron energy reduced by 1/e.

$$\frac{1}{X_0} = 4\alpha r_e^2 \left[\frac{\rho N_A}{A}\right] \left\{Z^2 \left[\underline{L_{rad} - f}\right] + \underline{Z L'_{rad}}\right\}$$

coefficients w/ softer
dependence on Z

Electron-Positron Pair Production

- In matter (the heavier the better) photons with energy above 1 MeV convert to electron positron pairs. These are colinear, have the energy of the original photon.

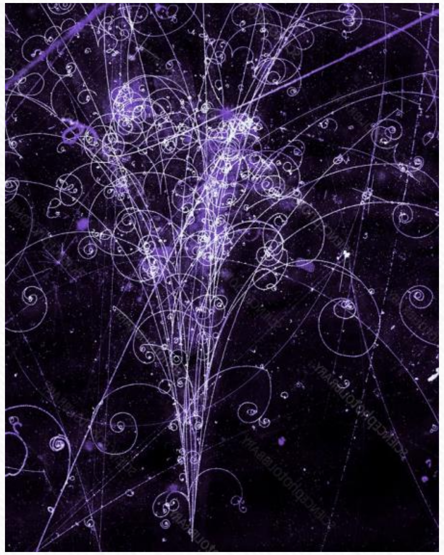
$$\frac{d\sigma}{dx} = \frac{A}{X_0 N_A}\left(1 - \frac{4}{3}x\,(1-x)\right)$$

x: the fraction of energy carried by the electron.

$$\sigma = \frac{7}{9}\frac{A}{\rho X_0 N_A}$$

$$\frac{1}{n\sigma} = \frac{9}{7}X_0$$

And so the mean free path for pair production is $\lambda = \frac{9}{7}X_0$



Multiple Coulomb Scattering

- Charged particles heavier than the electron also scatter from nuclei but being more massive their path is less disrupted, photons are not brem'd as with electrons.
- Nonetheless small deviations build up after multiple Coulomb interactions take place incoherently over a path length x. This disrupts the direction of tracks in dense matter without being a source of energy loss.

$$\theta_0 = \frac{13.6\ \mathrm{MeV}}{\beta c p} z \sqrt{\frac{x}{X_0}} \left(1 + 0.038 \ln \frac{x z^2}{X_0 \beta^2}\right)$$

- Primary effect is on track reconstruction.

Hadronic interactions with nuclei

- Hadronic processes involve the interaction between hadrons (charged and neutral) and the nuclei.
- Hadronic processes exemplify discrete (vs continuous) processes: either the hadron strikes a nucleus (the force is short-range) or not.
- When it strikes the nuclei, it creates a shower of secondaries, including protons, neutrons, charged and neutral pions.
- Neutral pions lead to clumps of electromagnetic energy.
- Other hadrons propagate through the material and may interact further.
- There are no simple formula to describe this, only models incorporating measured and predicted cross sections.

Geant4

- Geant4 (“G4”) is a toolkit for simulating the interaction of particles with a detector system. It is a successor to the obsolete FORTRAN packaged called GEANT3 (“G3”) workhorse of detector simulation in the last century.
- It is a C++ class library and the framework for all modern detector simulation.
- Also refers to the international collaboration, with leadership centered at CERN, which developed the software in the late 1990s and which continues to develop and support it.
- They run periodic tutorials in GEANT4 software development lasting 1 week.
- The next one in Catania, Sicily, October 7-10, registration deadline Sep 21:
- https://agenda.infn.it/e/XIII-International-Geant4-School
- Past tutorials here: https://geant4.web.cern.ch/collaboration/events?past

`fullSimLight`

- `fullSimLight` is a simulation program with a command line interface that uses G4 as the simulation engine. It was developed within ATLAS.
- It reads in:
  - Geometry (format: SQLite file)
  - Events (HepMC3 ASCII, or generated internally with Pythia).
- It really was designed with a mind to studying elements of a larger simulation and debugging; today it fully serves that role.
- Incidentally, it is quite good for small demonstrations such as those that I am going to do today.

Demonstration #1:

- Read in a geometry file and visualize it.
- In this demonstration we will demonstrate Geant4 geometry visualization.

```
fullSimLight -i -g TRT.db
```

The TRT is the ATLAS transition radiation detector, a straw tracking chamber.

- Some commands to try at the prompt:

```
/vis/geometry/list
/vis/drawVolume ! ! 1
/vis/drawVolume ! ! 2
```

Demonstration #2

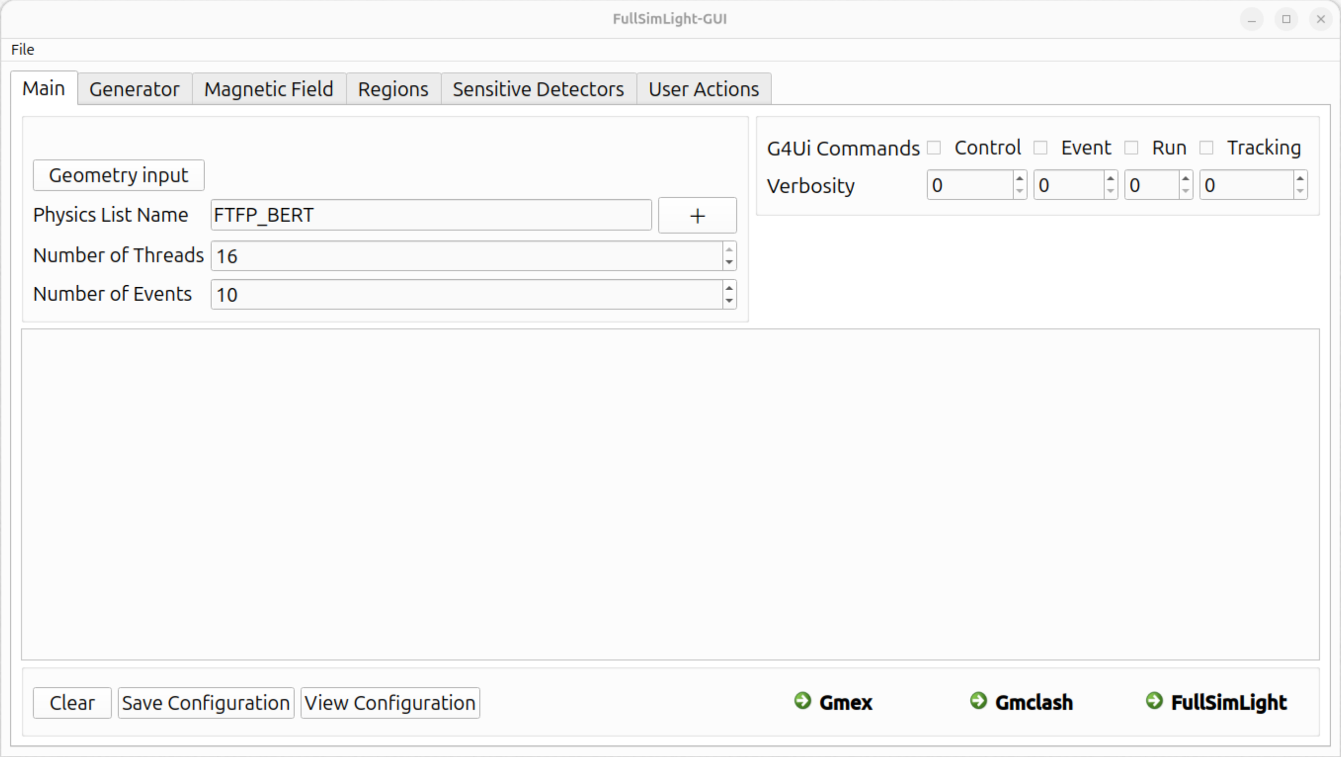
```
fullSimLight –i –g muonSystem.db –m muonVis.g4
```

`fsl`

- `fullSimLight` has a few options and even more g4 commands that may be given in a macro file.
- This is cumbersome, so we invented a graphical user interface (GUI)
- Called `fsl`, it can be used to generate the configuration file, or to run directly.

- It is also going to be helpful today.

Demonstration #3



Physics List Name: FTFP_BERT
Number of Threads: 16
Number of Events: 10
G4Ui Commands: ☐ Control ☐ Event ☐ Run ☐ Tracking
Verbosity: 0 0 0 0

In this demonstration we specify the geometry input and select an event.

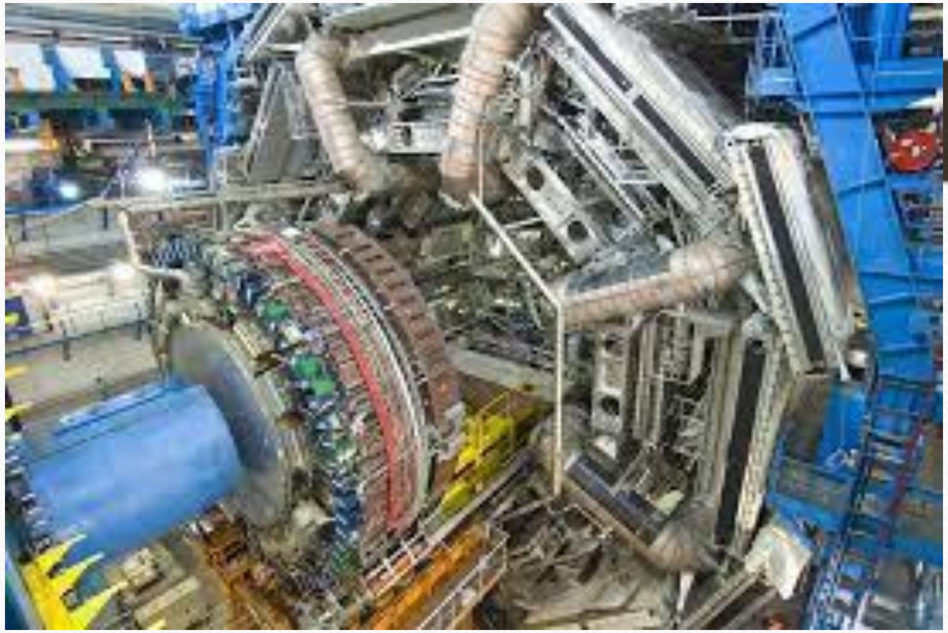
We then run fullSimLight from the GUI.

We can also run the gmex tool….

Demonstration #4 gmex: the geometry explorer

A little more on the topic of geometry.

- Position and composition of all elements must be correct.
- Density is most important for energy loss, Compton scattering, and hadronic interactions.
- Nuclear charge most important for multiple scattering, bremsstrahlung, pair production.
- Nuclear isotopes are important if the radiation field is to be calculated.
- Material upstream of the calorimeter has an impact on shower shapes and therefore on electron/photon identification and energy measurement.

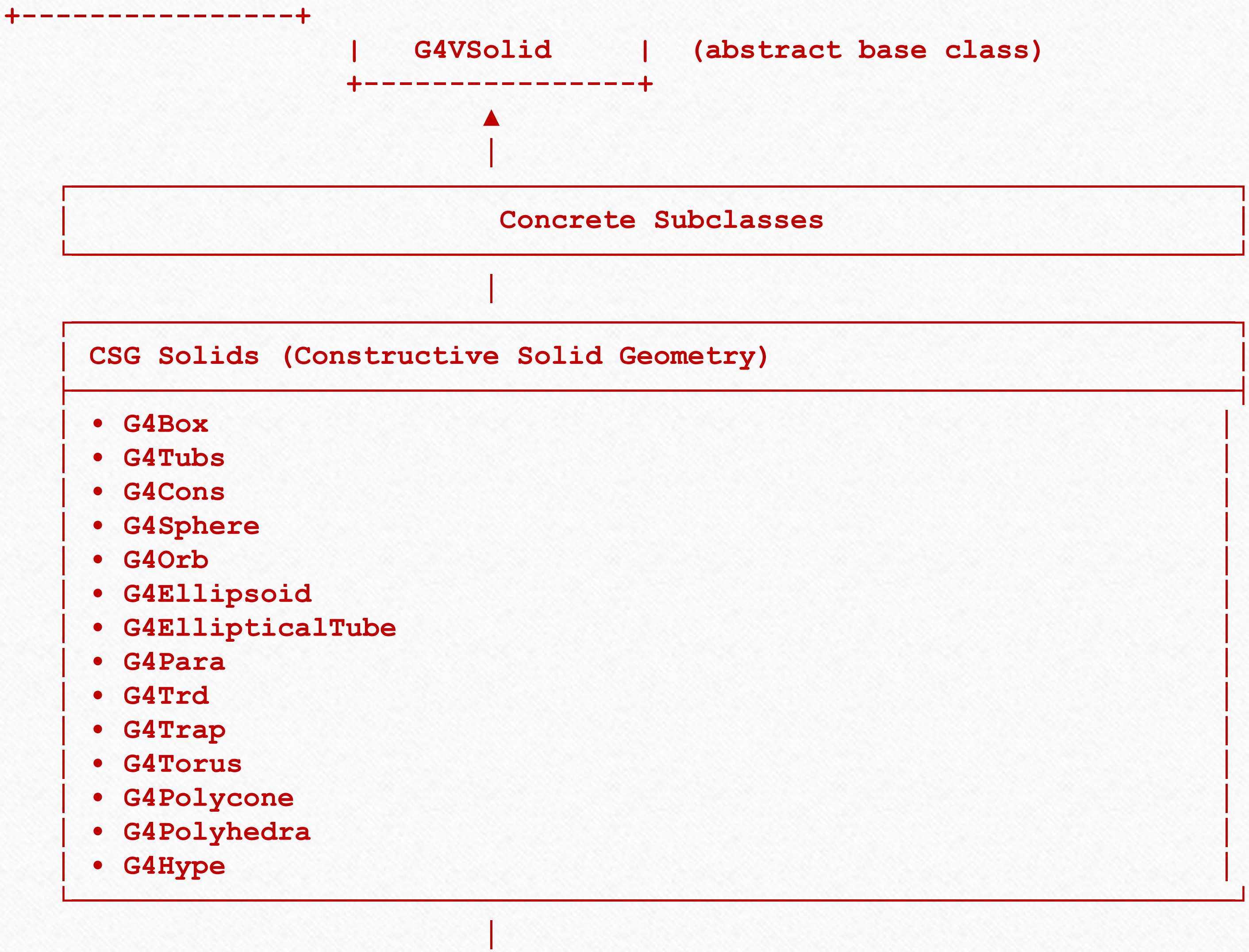


```
G4double z, a, density;
G4String name, symbol;
G4int ncomponents, natoms;

a = 1.01*g/mole;
G4Element* elH  = new G4Element(name="Hydrogen",symbol="H" , z= 1., a);

a = 16.00*g/mole;
G4Element* elO  = new G4Element(name="Oxygen"  ,symbol="O" , z= 8., a);

density = 1.000*g/cm3;
G4Material* H2O = new G4Material(name="Water",density,ncomponents=2);
H2O->AddElement(elH, natoms=2);
H2O->AddElement(elO, natoms=1);
```

The Geometry in Geant4 is built by composition:

- Materials composed of Elements.
- “Logical volumes” composed of Solids plus materials
- ”Physical volumes” composed of logical volumes and their position, orientation.

- These entities are then assembled into a directed acyclic graph (DAG).

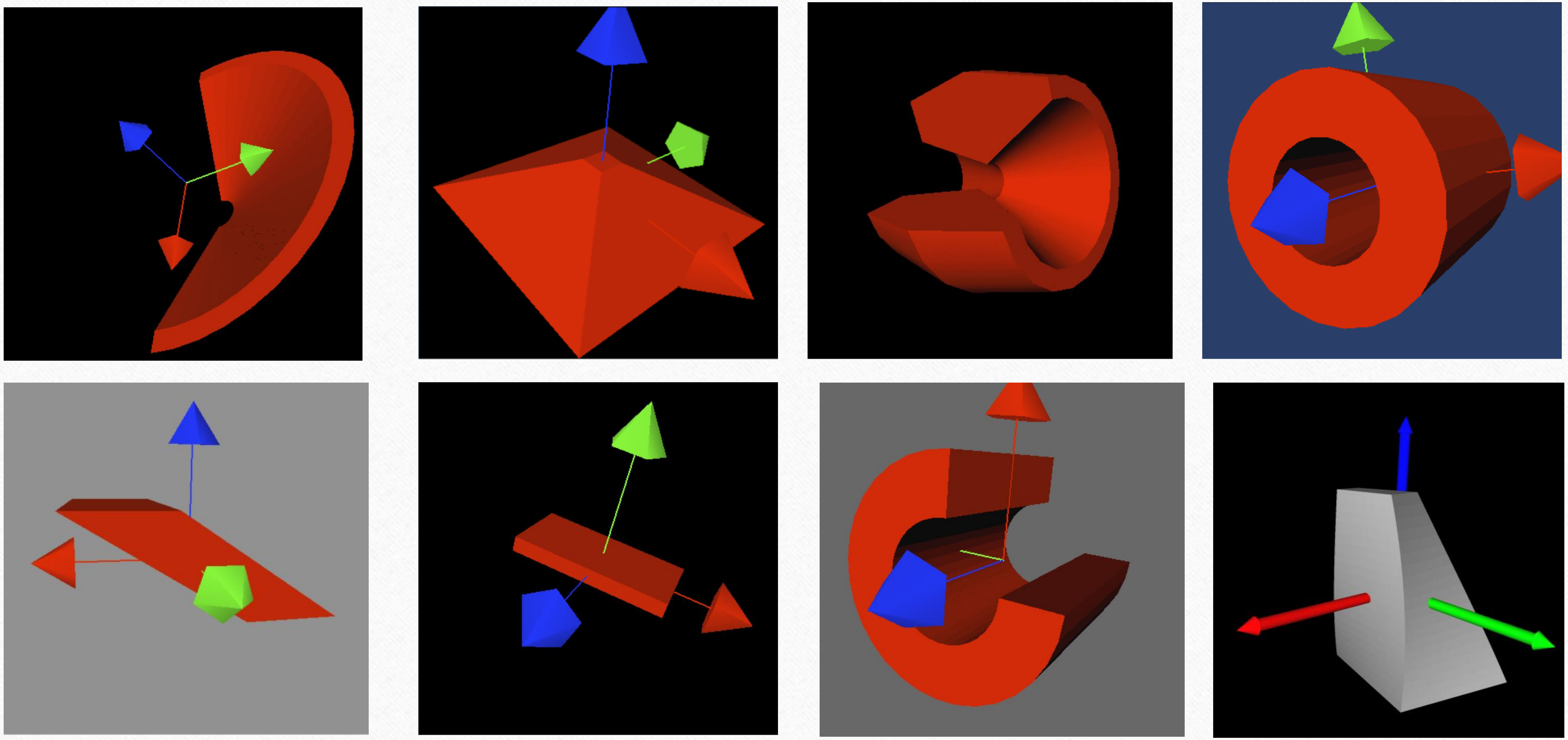


G4VSolid (abstract base class)

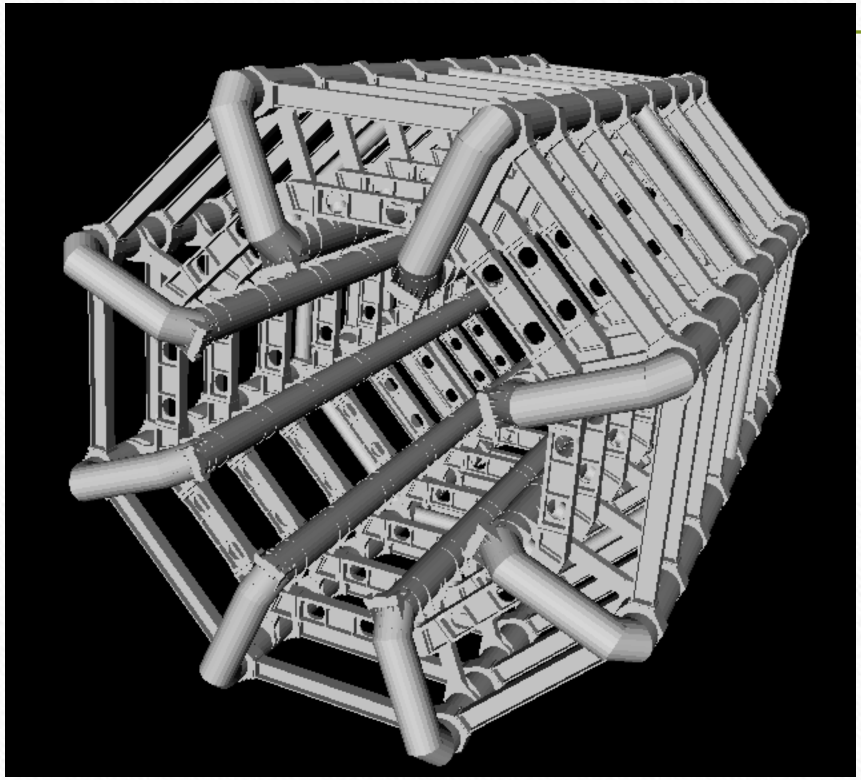
Concrete Subclasses

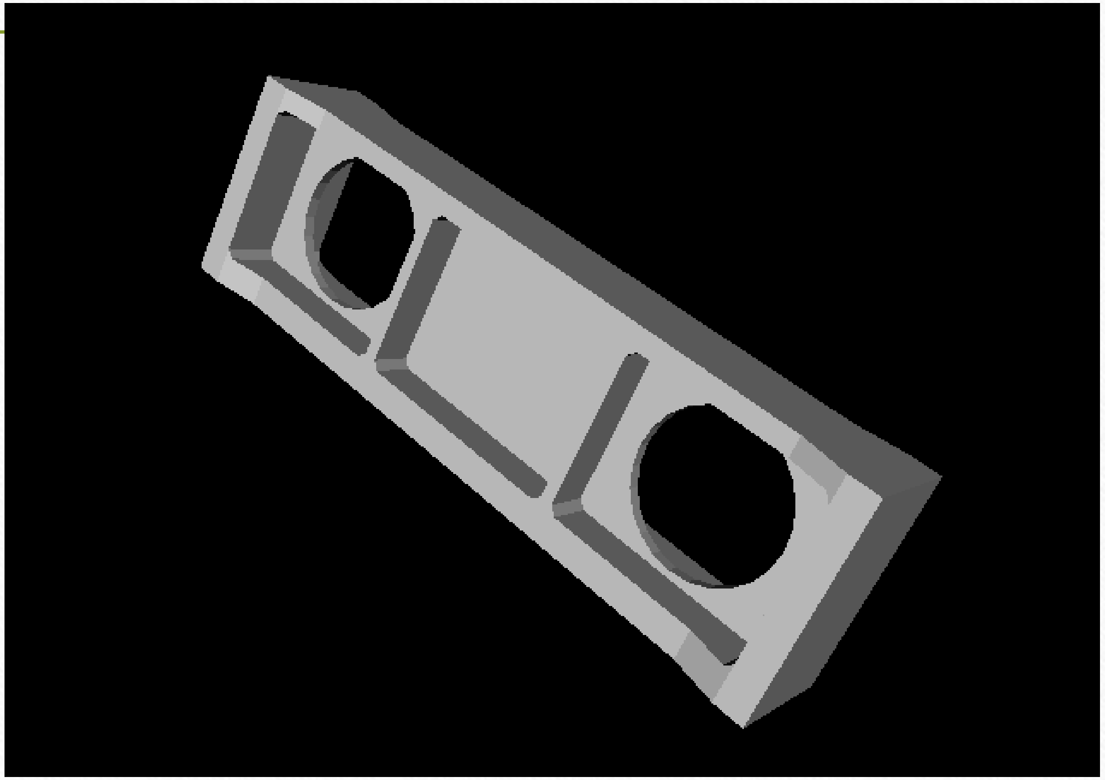
CSG Solids (Constructive Solid Geometry)

- G4Box
- G4Tubs
- G4Cons
- G4Sphere
- G4Orb
- G4Ellipsoid
- G4EllipticalTube
- G4Para
- G4Trd
- G4Trap
- G4Torus
- G4Polycone
- G4Polyhedra
- G4Hype



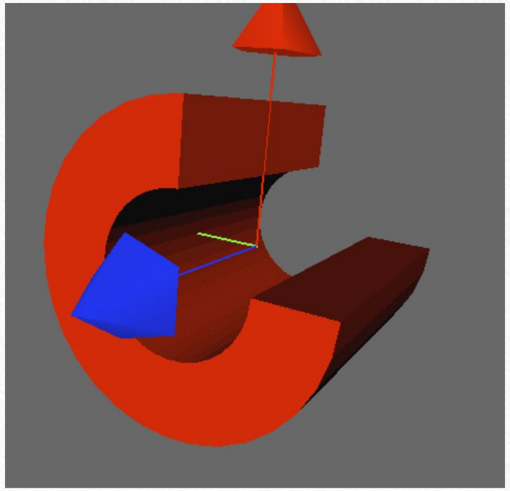
Boolean volumes





Shapes: how to create

```
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);
```

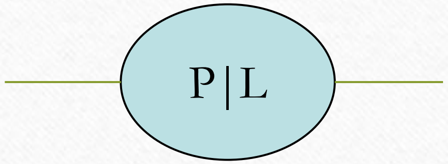


Volumes: create & place

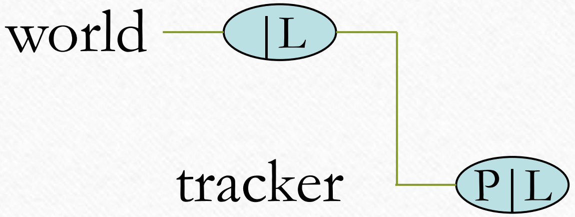
```
G4LogicalVolume* worldLog;

G4LogicalVolume* trackerLog = new G4LogicalVolume(trackerTube, Al, "Tracker");

G4VPhysicalVolume* trackerPhys
  = new G4PVPlacement(0,                                    // no rotation
                      G4ThreeVector(pos_x, pos_y, pos_z),   // translation position
                      trackerLog,                           // its logical volume
                      "Tracker",                            // its name
                      worldLog,                             // its mother (logical) volume
                      false,                                // no boolean operations
                      0);                                   // its copy number
```

Let me represent the volume as a "Physical/Logical" node when so assembled.

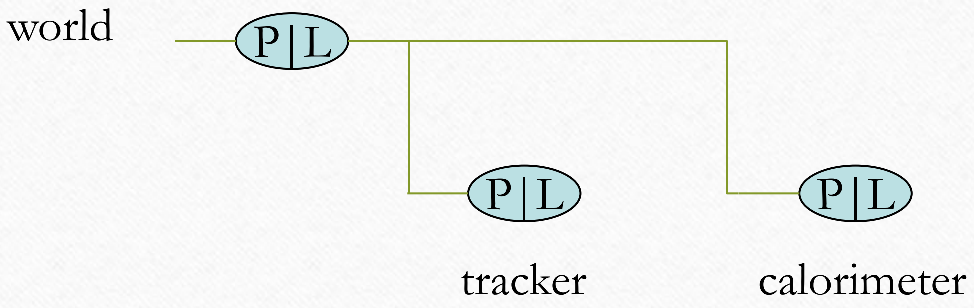
- The physical volume may connect to a mother.
- The logical volume has a solid, a material and children.



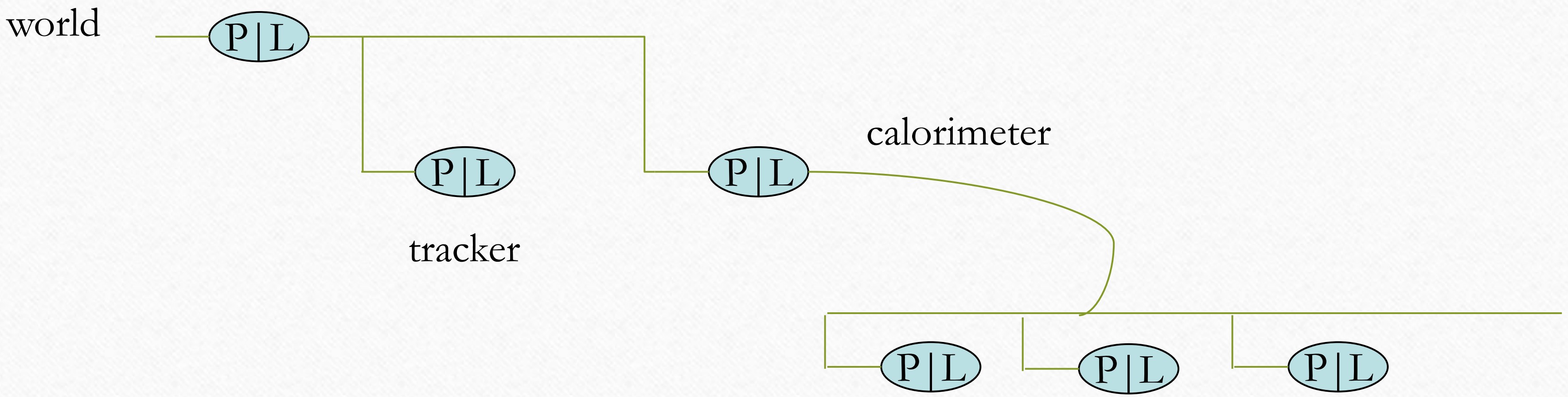
You will have done this:



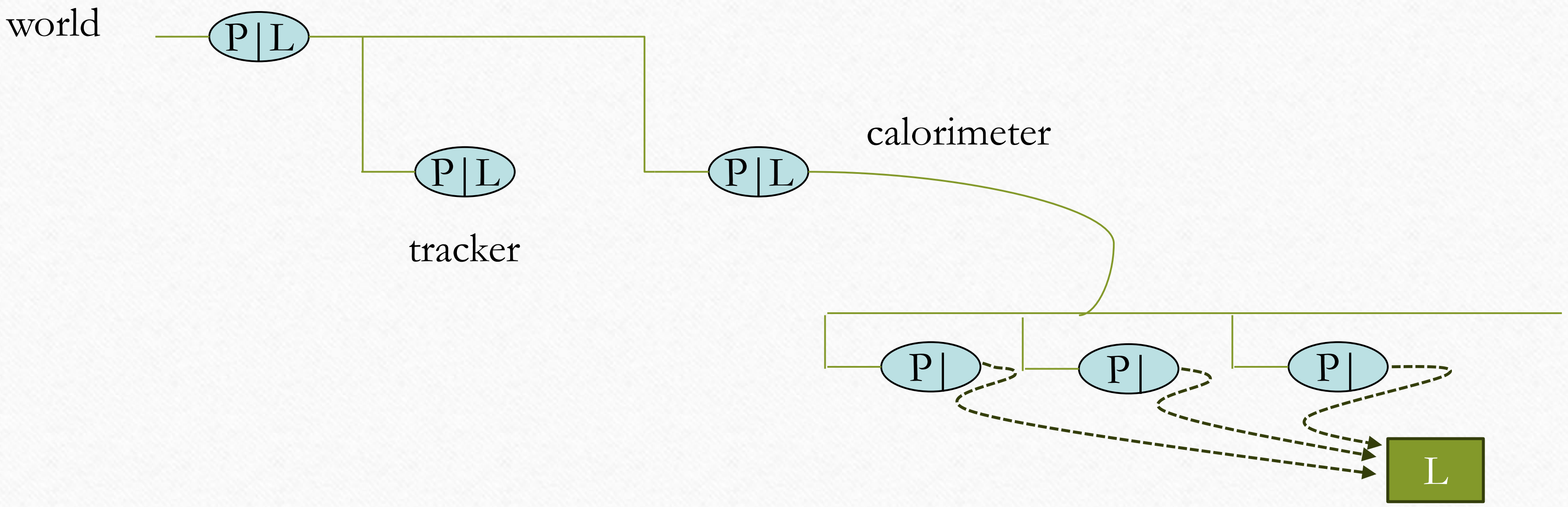
Do this as often as needed.



Nest as deeply as required.

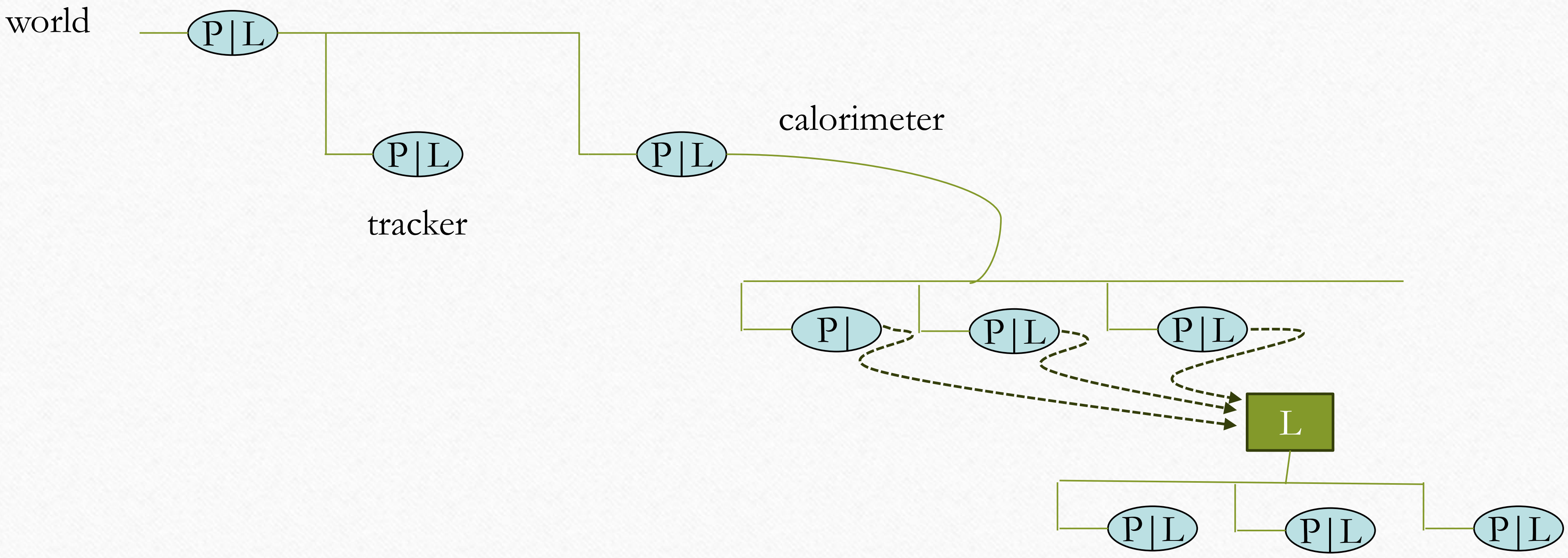


Share instances of Logical Volumes, save memory.



(Logical Volumes can share Solids, too).

Share entire subtrees.

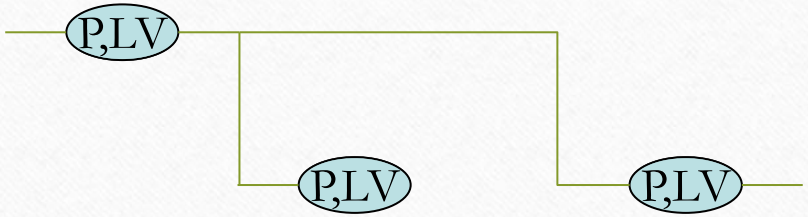


Geometry: used throughout the offline

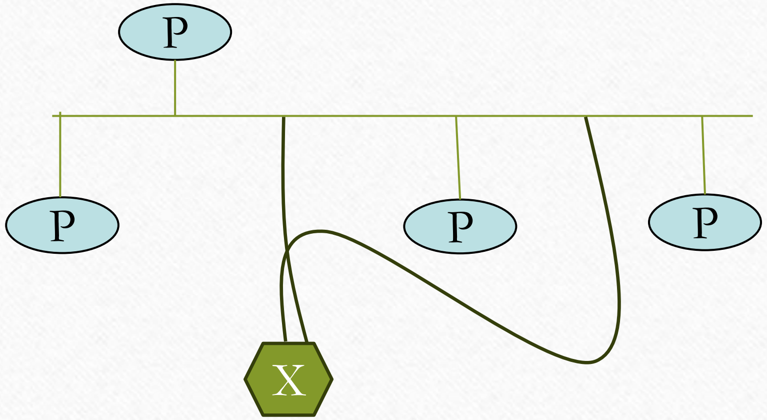
- Geometry is used in simulation, but also reconstruction event visualization, and more.
- Why not use Geant4 geometry for all of that?
  - Each volume contains a lot of overhead to serve the purposes of simulation
  - One always has to we the benefit of functionality against the cost of external dependencies.
- In ATLAS we use GeoModel for a general purpose description.
  - The geometry input for the geomodel explorer (gmex) is GeoModel.
  - The geometry input for fullSimLight or fsl is Geomodel
  - And there are additional tools
- See geomodel.web.cern.ch

Scene graph approach to geometry

Geant4



GeoModel: only one kind of volume.
properties are inferred from place in graph.
no voxel tables. This is just a data layer.



It’s designed to be light, portable, persistifiable.
Faithful representation from/to G4 geometry

Equation of motion.

- The equation of particles depends on:
  - Charge of the particle.
  - Whether or not there is a magnetic field.
  - Whether or not energy is lost in the particular medium.
- In the most complicated case, the ordinary differential equation:

$$\begin{aligned} \frac{d\vec{r}}{ds} &= \frac{1}{mc\beta\gamma}\vec{p} \\ \frac{d\vec{p}}{ds} &= \frac{q}{mc\beta\gamma}\left(\vec{p}\times\vec{B}\left(\vec{r}\right)\right) - \frac{\vec{p}}{mc\beta^2\gamma}\left|\frac{dE}{dx}\right| \end{aligned}$$

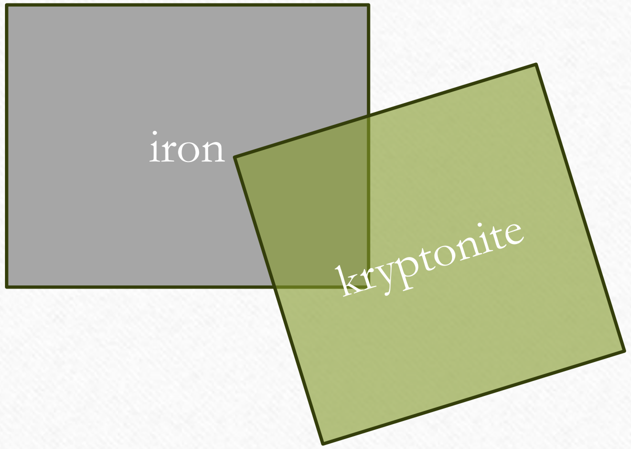
is solved using an ordinary differential equation solver (i.e. a Runge-Kutte integrator). This is the easy part.

The hard part: navigation

- The hard part is to determine how far the particle goes before “something happens”,   either an interaction or a decay or…the crossing of a geometric boundary.
- In particular the last step is difficult when many volumes are present.
  - Which of the many surfaces will be crossed by the current trajectory?
  - To aid in the decision, each logical volume is subdivided into “voxels” (voxelization); this takes place during geometry initialization. The voxels keep track of their neighbors.
  - It can be tuned somewhat for special circumstances.

Integrity of the geometry description

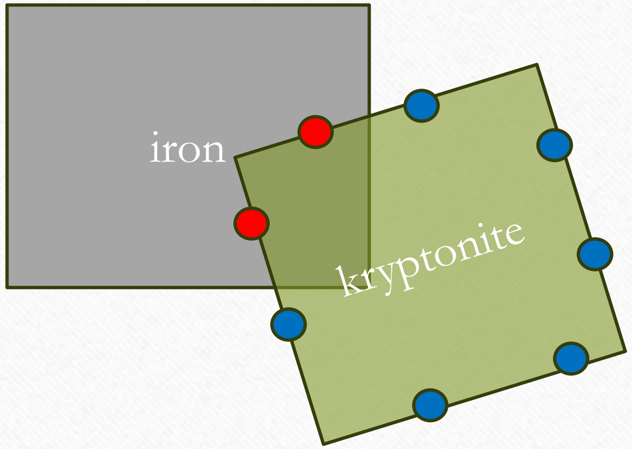
- Imagine a geometry overlap in some region.



- The manner in which a particle should interact with its surroundings is undefined.
- The particle will get lost during navigation.
- Geant4 will detect this.
- But we can purge the geometry of such issues with a specialized procedure.

Clash detection

- Two services provide by Geant 4 solids aid in detecting geometry clashes:
    - method to generate points on the surface of a solid.
    - method to determine whether a point is inside or outside a volume.
    - this can detect both parent/daughter and sister/sister volume clashes.



Demonstration # 4

- fsl ; use kitchenSink; clash detect in fsl

Tracking

- Geant4 tracks particles through the geometry with the aid of a stepper:
  - integrates the equations of motion step-by-step
  - applies continuous and discrete physics processes.
    - continuous processes may affect the equations of motion.
    - discrete processes (nuclear interaction, particle decay) may limit the step.
  - if the step is not limited by a discrete process, it may be limited by a geometrical surface.

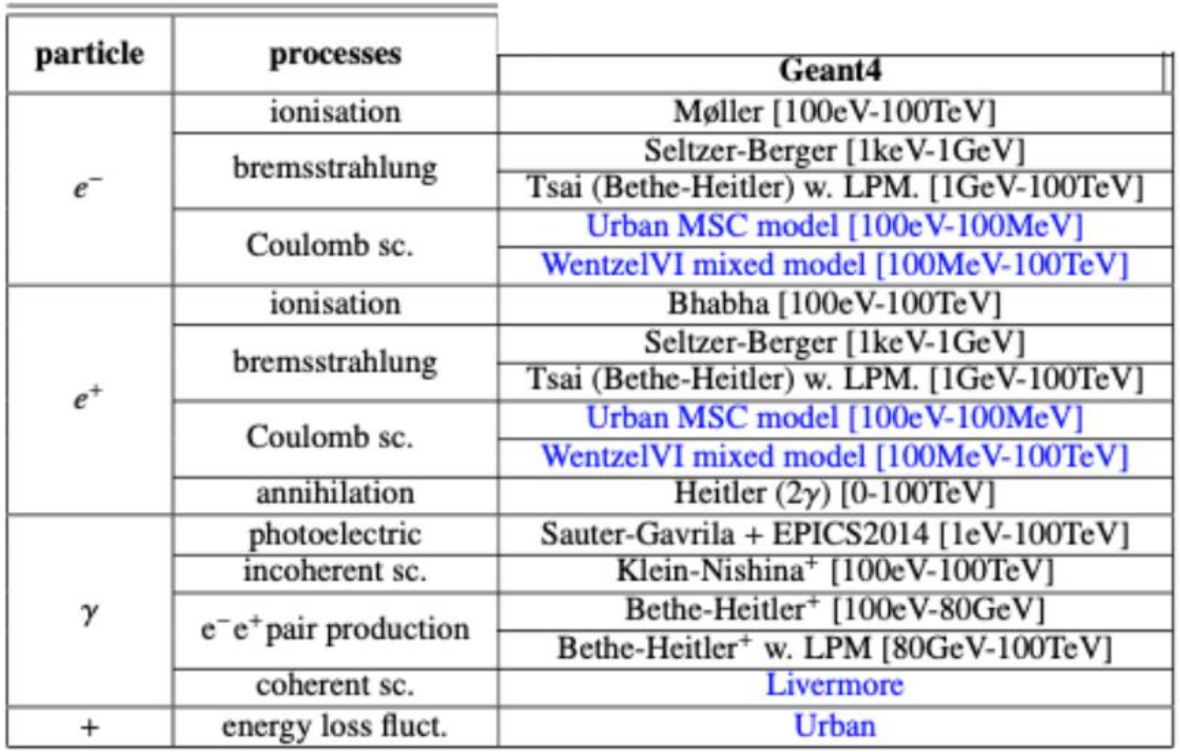
Physics processes.

- Particle interactions with matter and decay are implemented in Geant4 by ”Physics Processes” (class `G4VProcess`)
- For continuous processes the effects are applied during the step.
- For discrete processes, the effect is applied after the step is taken.
- The cross section must be known for either process
  - Geant4 comes with extensive tables of measured or computed cross sections.
  - Interaction length from $\lambda = 1/(n\sigma)$ where n is number density.

Physics Lists

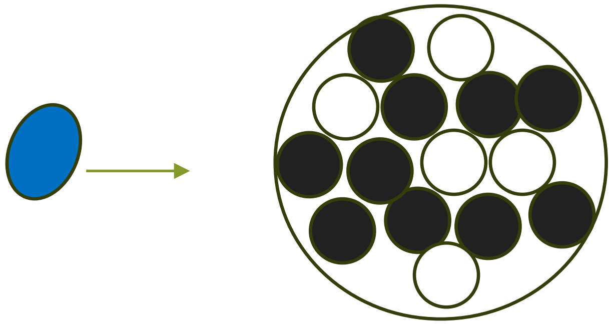
- Prepackaged lists are usually used but you can customize.
  - For collider physics the “Bertini cascade model” is usually used.
  - Other lists are more appropriate for low energy and medical physics.
  - Electromagnetic, weak, hadronic and optical processes included.

Electromagnetic processes

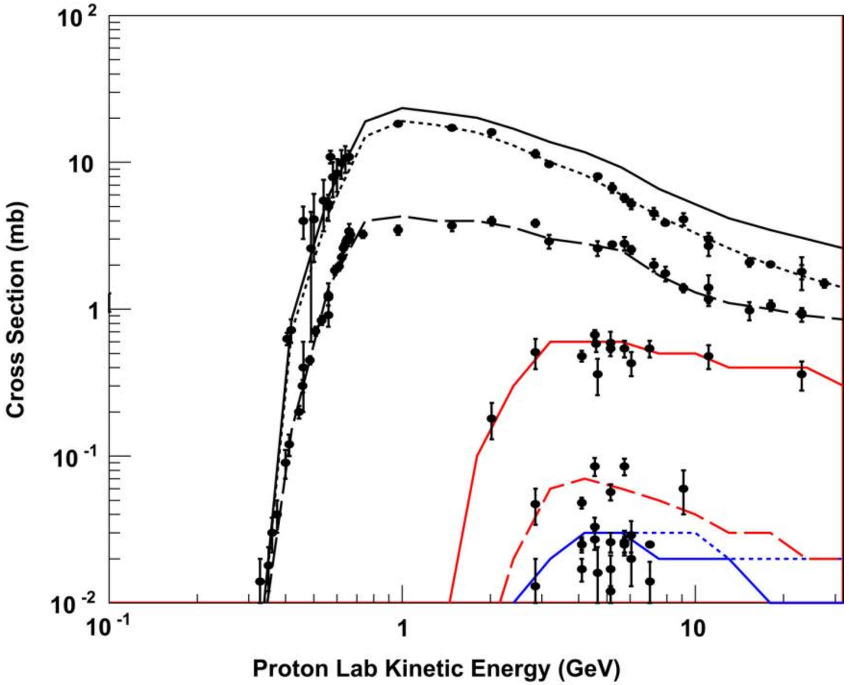


particle	**processes**	**Geant4**
e^-	ionisation	Møller [100eV-100TeV]
	bremsstrahlung	Seltzer-Berger [1keV-1GeV]
		Tsai (Bethe-Heitler) w. LPM. [1GeV-100TeV]
	Coulomb sc.	Urban MSC model [100eV-100MeV]
		WentzelVI mixed model [100MeV-100TeV]
e^+	ionisation	Bhabha [100eV-100TeV]
	bremsstrahlung	Seltzer-Berger [1keV-1GeV]
		Tsai (Bethe-Heitler) w. LPM. [1GeV-100TeV]
	Coulomb sc.	Urban MSC model [100eV-100MeV]
		WentzelVI mixed model [100MeV-100TeV]
	annihilation	Heitler (2γ) [0-100TeV]
γ	photoelectric	Sauter-Gavrila + EPICS2014 [1eV-100TeV]
	incoherent sc.	Klein-Nishina$^+$ [100eV-100TeV]
	e^-e^+pair production	Bethe-Heitler$^+$ [100eV-80GeV]
		Bethe-Heitler$^+$ w. LPM [80GeV-100TeV]
	coherent sc.	Livermore
+	energy loss fluct.	Urban

Hadronic Physics



- Elastic, inelastic, capture, processes modeled.
- A cascade develops as the particle strikes nucleons within the nucleus. Cross sections from experimental tabulation or calculation.
- This leaves the nucleus in an excited state, it decays by fission or deexcitation.
- Nuclear reactions induced by photons or leptons.



The “Geantino”

- The simplest particle to track is the geantino. It is not a real particle, but simply a software tool.
- It comes in two species, charged and neutral.
- It is tracked through the detector but undergoes no physics processes (beyond response to the magnetic field in case of the charges species).
- This is not useful for anything, right?
- Reply: Not for the moment…

Demonstration #4: Muon Interactions with the detector.

```
fullSimLight –i –g calorimeter.db –m muonVis.g4
```

Demonstration #5: Electron Interactions with the detector.

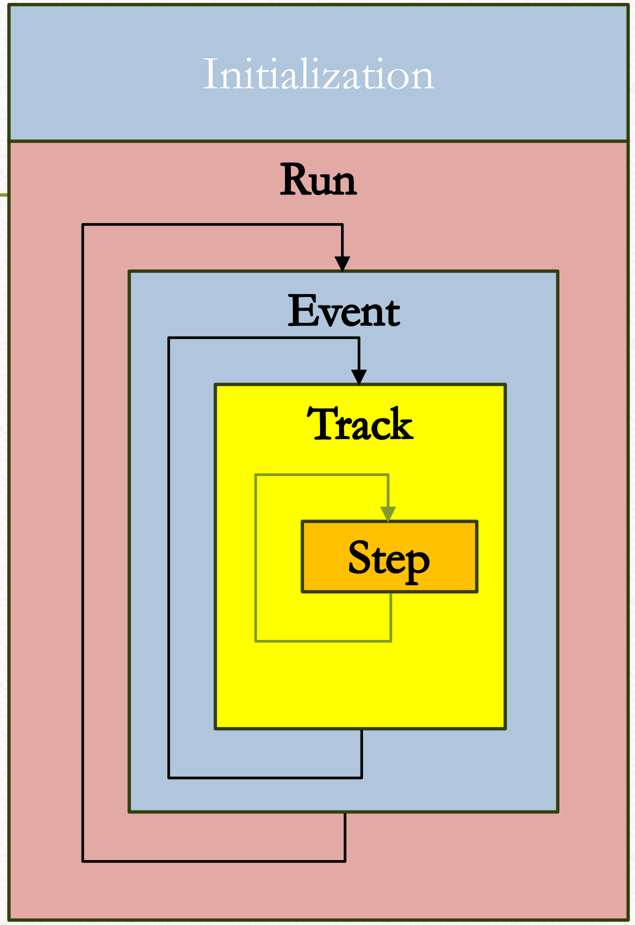
```
fullSimLight –i –g calorimeter.db –m electronVis.g4
```

Demonstration #6: Pion Interactions with the detector.

```
fullSimLight –i –g calorimeter.db –m pionVis.g4
```

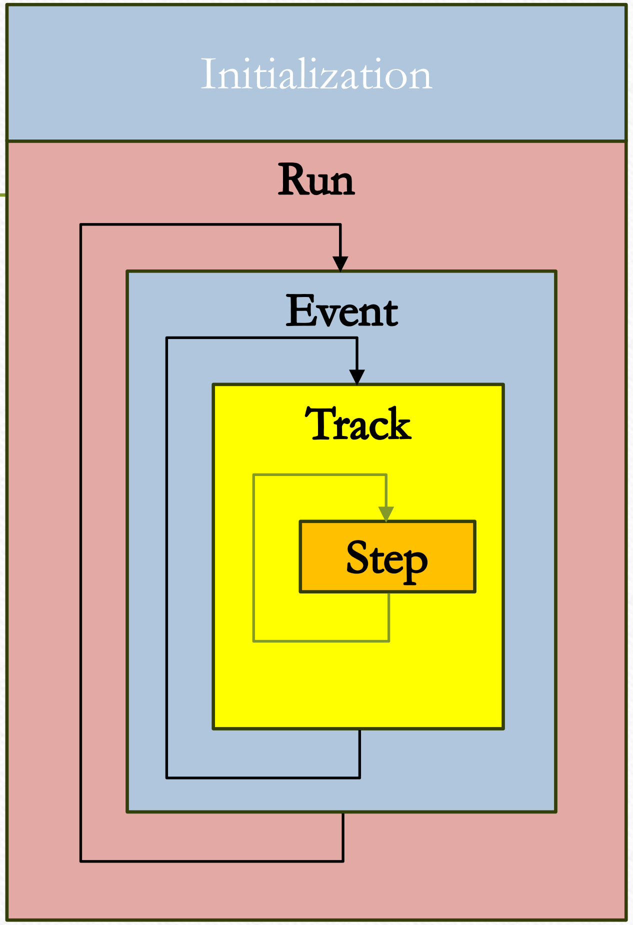
Structure of a Geant4 program

- A Geant4 program does its work within a main loop called an event loop which you do not control. You simply start the loop.
- You customize what happens in the loop by scheduling certain actions at certain times.
- This is done at initialization time, when you construct the geometry.



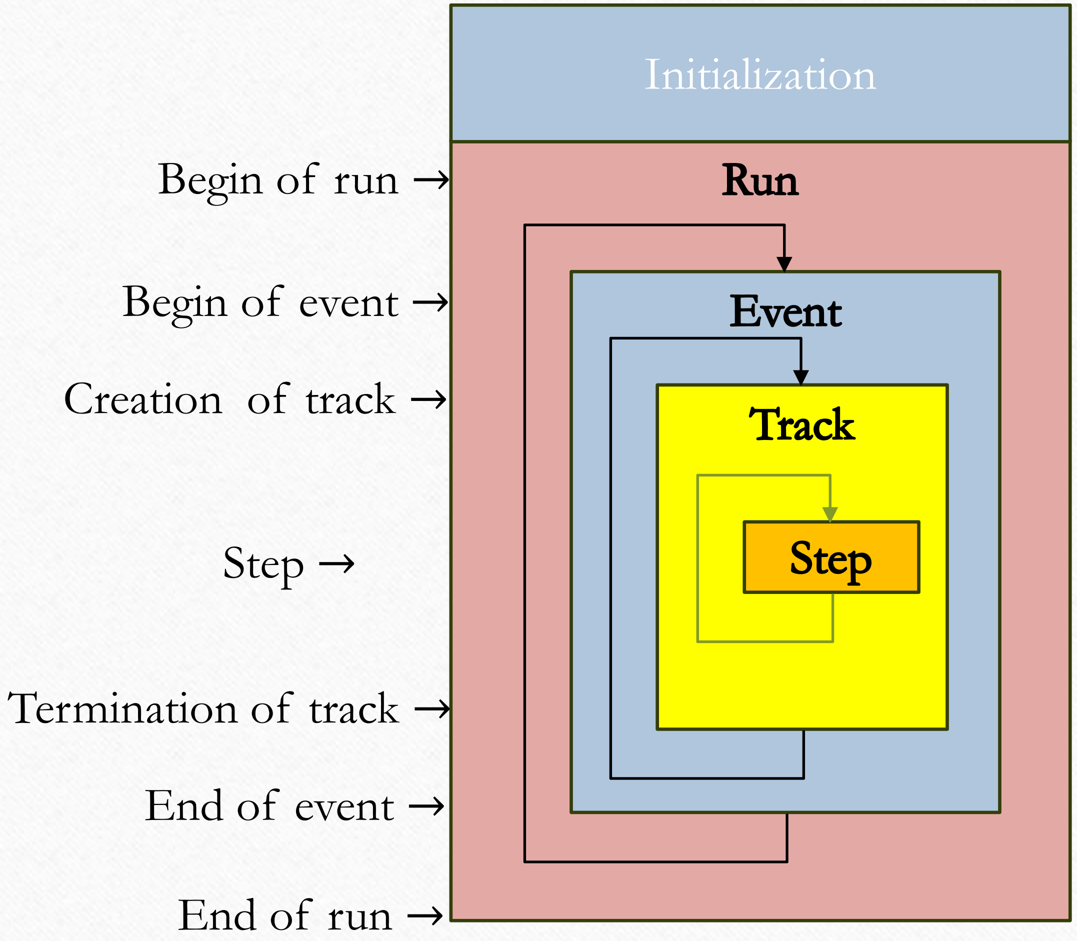
Structure of a Geant4 program

- A run is a collection of events, corresponding to collisions in a simulated detector or the firing of a “particle gun"
- Each event has tracks, both primary and secondary.
- Each tracks has multiple steps, under the control of a stepping manager (class).



User Actions

- It is possible to schedule an interuption of processing at various points.
- This done by:
  - Defining your own UserAction class.
  - Instantiating it.
  - Declaring the object to Geant4.



Begin of run →

Begin of event →

Creation of track →

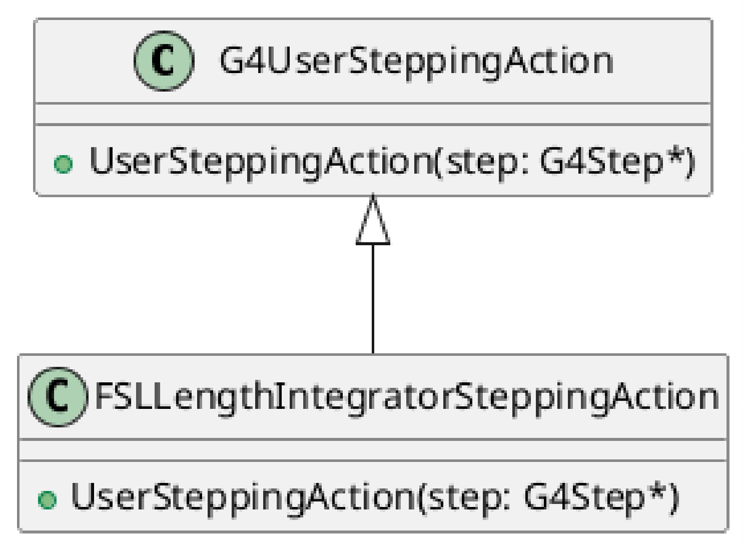
Step →

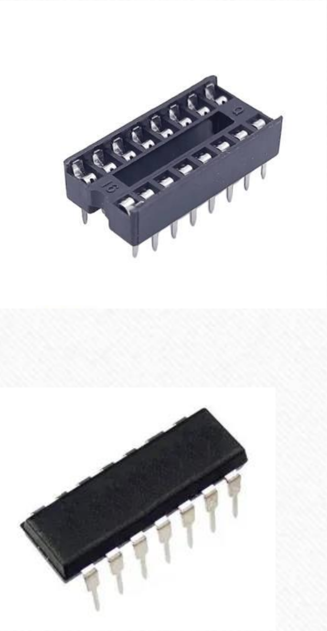
Termination of track →

End of event →

End of run →

Geant 4 and polymorphic design.


C
G4UserSteppingAction
UserSteppingAction(step: G4Step*)
C
FSLLengthIntegratorSteppingAction
UserSteppingAction(step: G4Step*)



In a nutshell: one designs an interface to a whole family of classes.

One can then design classes which adhere to the interface.

The system is robust against extension because extension does not require modification.

The system works with classes that have not yet been invented.

Important user-extensible classes in Geant4

- G4VUserDetectorConstruction: builds the detector (geometry and more)
- G4UserRunAction
- G4UserEventAction
- G4UserTrackingAction
- G4UserSteppingAction
- G4UserStackingAction
- G4VUserActionInitialization: Attaches these actions to a scheduler.

Demonstration #8

```
gmgeantino –g TRT.db –m geantino_TRT.g4 –o trt.root
plotGeantino.py
```

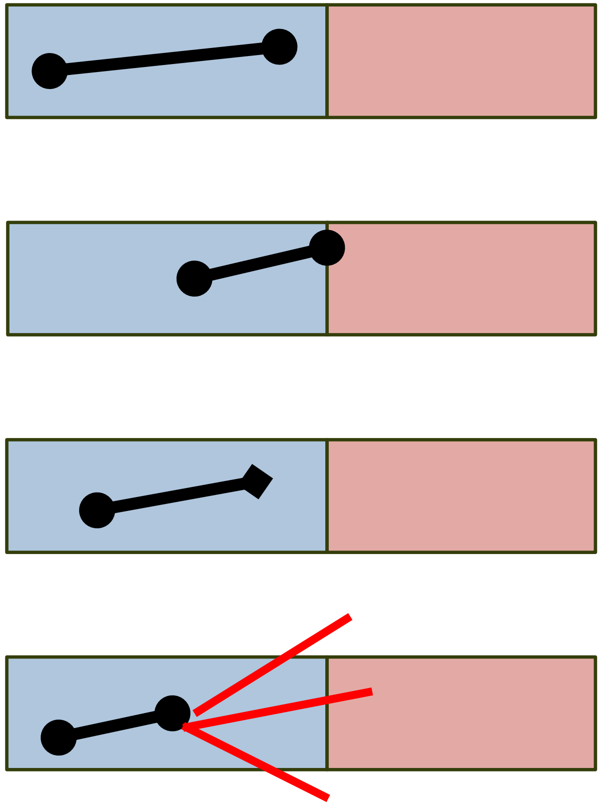
Mapping the material: `gmgeantino`

- `gmgeantino` is a command line geant4 utility.
  - It throws geantinos at the detector from aparticle gun.
  - It interrupts every step of the stepper and records the amount of material (X_0 and λ_{int}) is traverses vs. the detector.
  - It then makes a histogram of the material, a kind of "X-ray" of the detector.
  - This is important as a diagnostic tool.
    - Diagnose the simulation.
    - Diagnose the engineering!! (Too much material is very bad for energy measurement)
  - Let's start this now while I explain how this is done.

Stepping actions:

- To collect information during stepping:
  - Define a stepping action.
  - Instantiate a stepping action.
  - Add it to an object called a Run Manager before the event loop.
    - This means pass a pointer.

Types of step



Step terminates in current volume.

Step terminates at geometric boundary.

Particle ranges out, track is dead.

Discrete interaction or particle decay.

Information from a `G4Step`

- Track position
- Track momentum
- Energy loss along the track.
- Volume at the beginning of track.
- Volume at the end of the track.

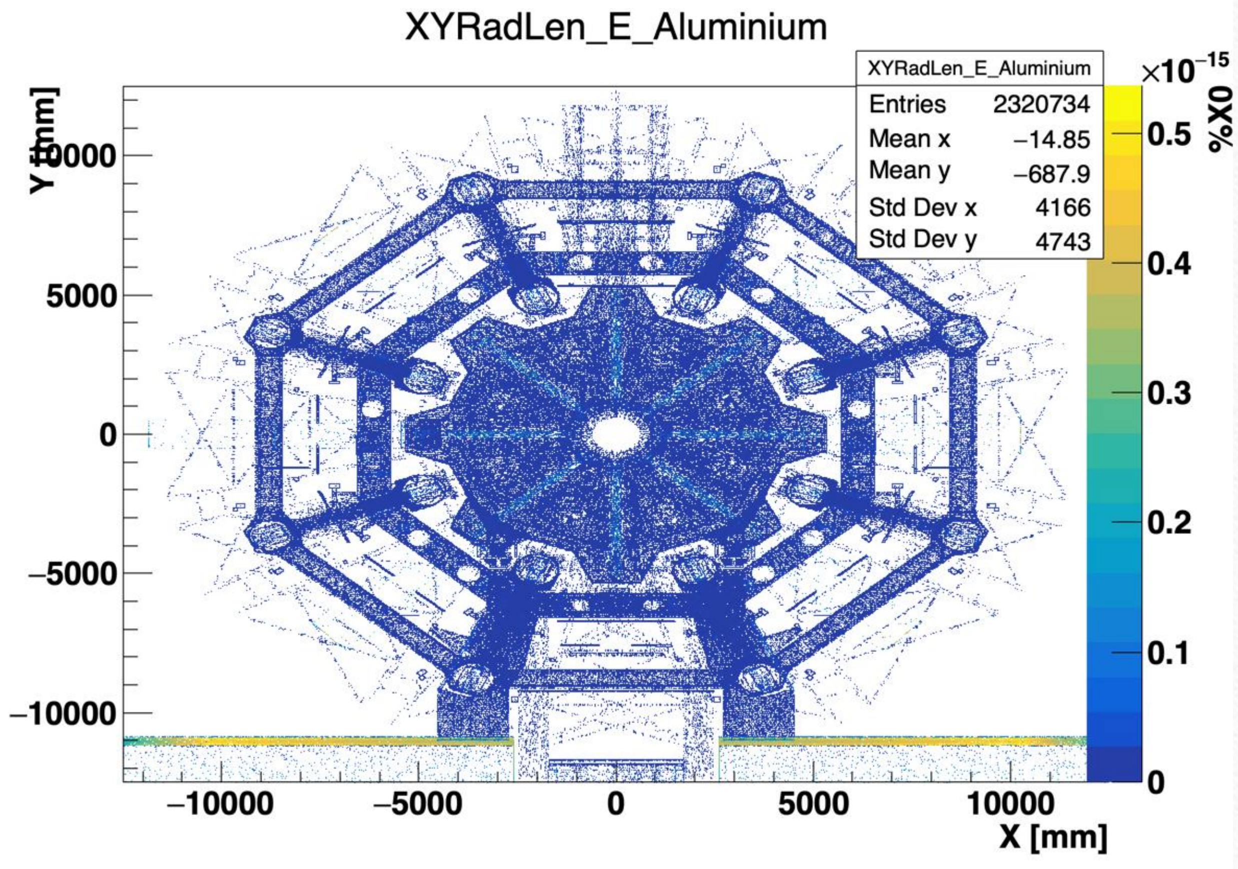
Demo #8 harvest results

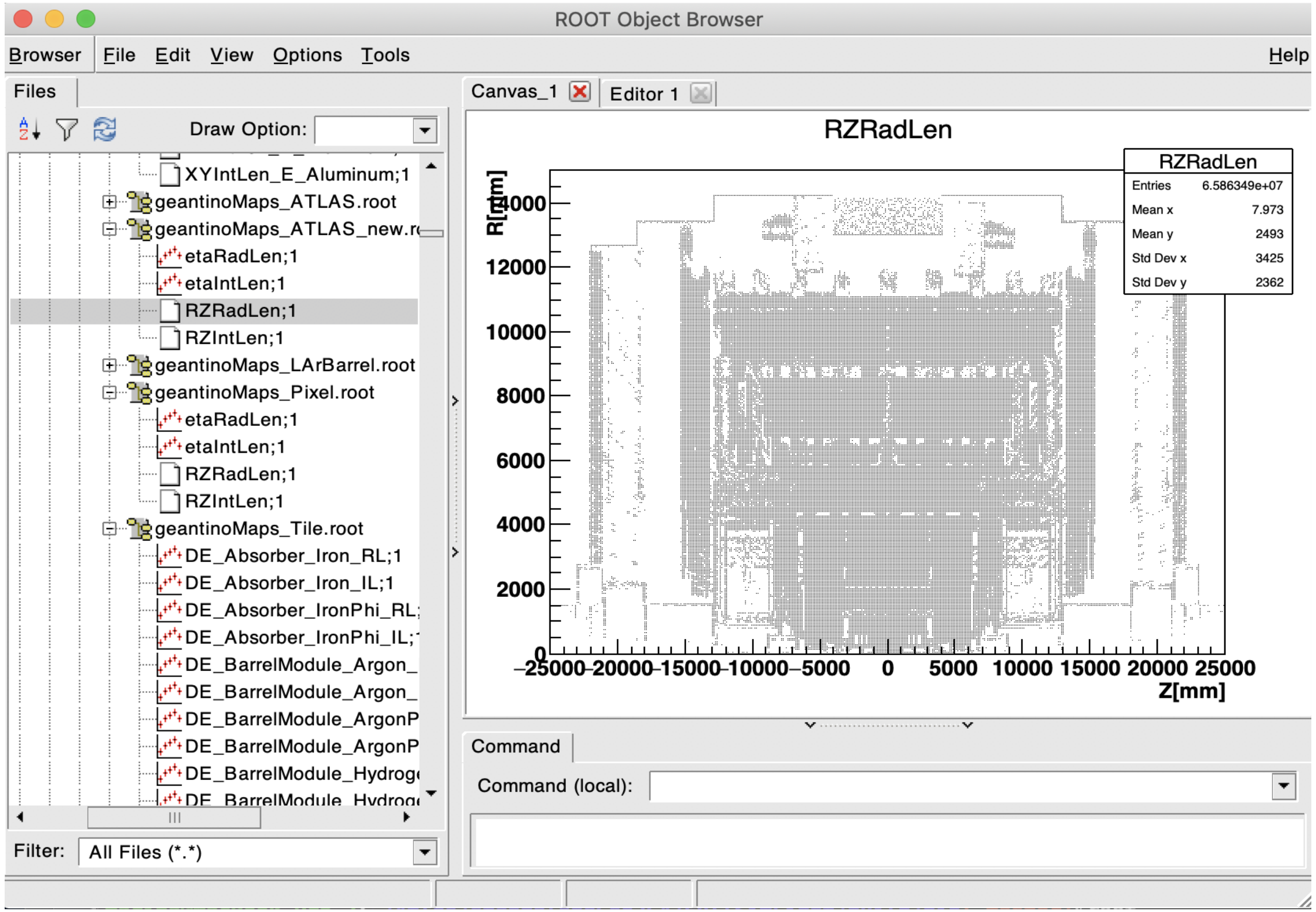
plotGeantino.py

An option to gmclash also allows to do a two-dimensional analog of geantino scans.

I don’t do this in front of you today since my laptop lacks the memory resources.

Here is an image that was so generated on another computer.

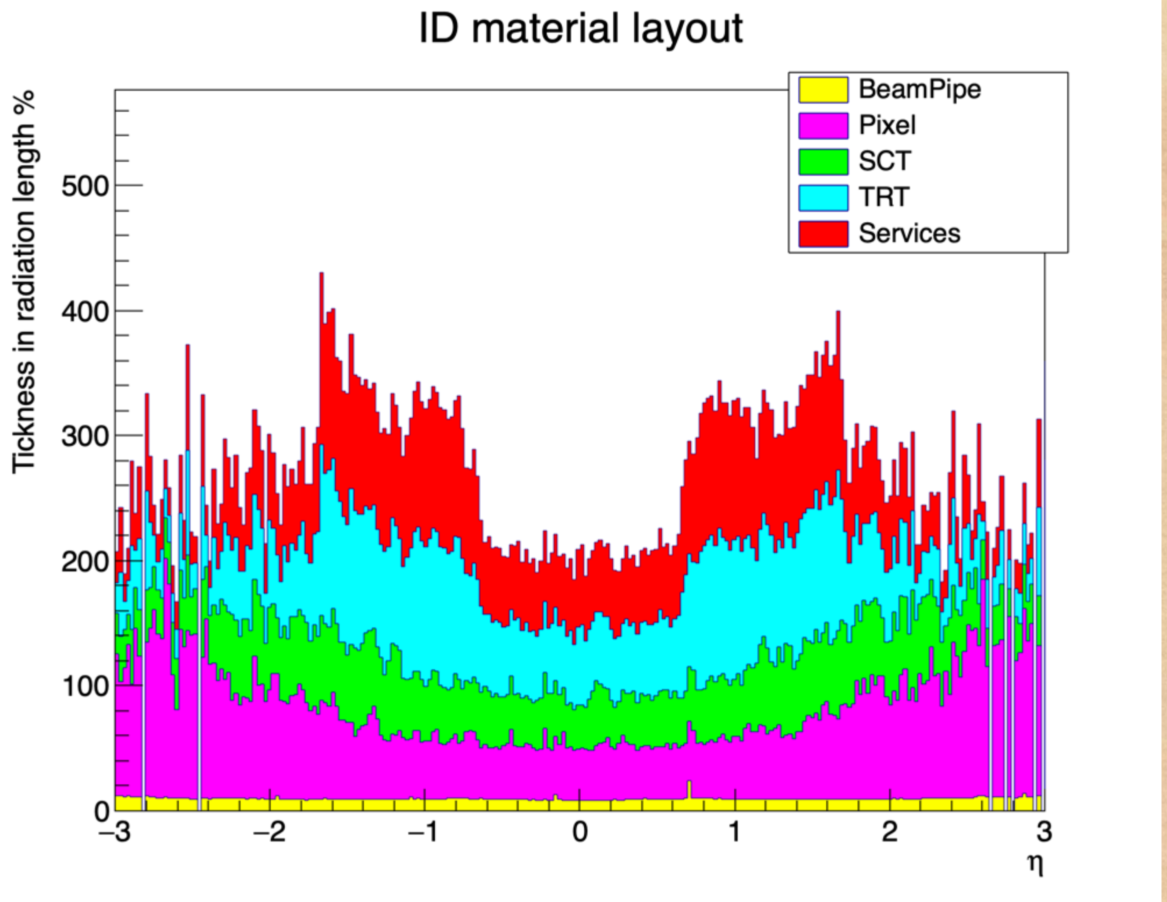




RZRadLen

Entries	6.586349e+07
Mean x	7.973
Mean y	2493
Std Dev x	3425
Std Dev y	2362

Geantino geometry maps: gmgeantino



```
./gmgeantino -g BeamPipe.db
./gmgeantino -g Pixel.db
./gmgeantino -g SCT.db
./gmgeantino -g TRT.db
./gmgeantino -g InDetServMat.db
```

Material distribution expressed in units of X_0 (Radiation Length) as a function of η for the ATLAS ID. The breakdown indicates the contributions of external services and of individual sub-detectors, including services in their active volume (plot reproduced from paper)

```
 //-------------------------------------------------------------------------
 // Accumulate results from one step
 //-------------------------------------------------------------------------

void FSLLengthIntegratorSteppingAction::UserSteppingAction(const G4Step* aStep)
 {
     double stepl = aStep->GetStepLength();

     const G4TouchableHistory* touchHist = aStep->GetPreStepPoint()->GetTouchable();

     G4LogicalVolume* lv = touchHist->GetVolume()->GetLogicalVolume();
     std::string volName = lv->GetName();

     G4Material* mat = lv->GetMaterial();
     double radl = mat->GetRadlen();
     double intl = mat->GetNuclearInterLength();

     double thickstepRL = stepl/radl *100;
     double thickstepIL = stepl/intl;
```

User actions in FullSimLight/FSL

- In `FullSimLight/FSL` the UserAction extensions are not only linked with the program, but **dynamically** linked.
- This means that a compiled, ready to run executable can gain new functionality.
- This is called a plugin, and it's not too hard to implement.
- It relies on a system call called `dlopen` and `dlsym`.

Demonstration #9

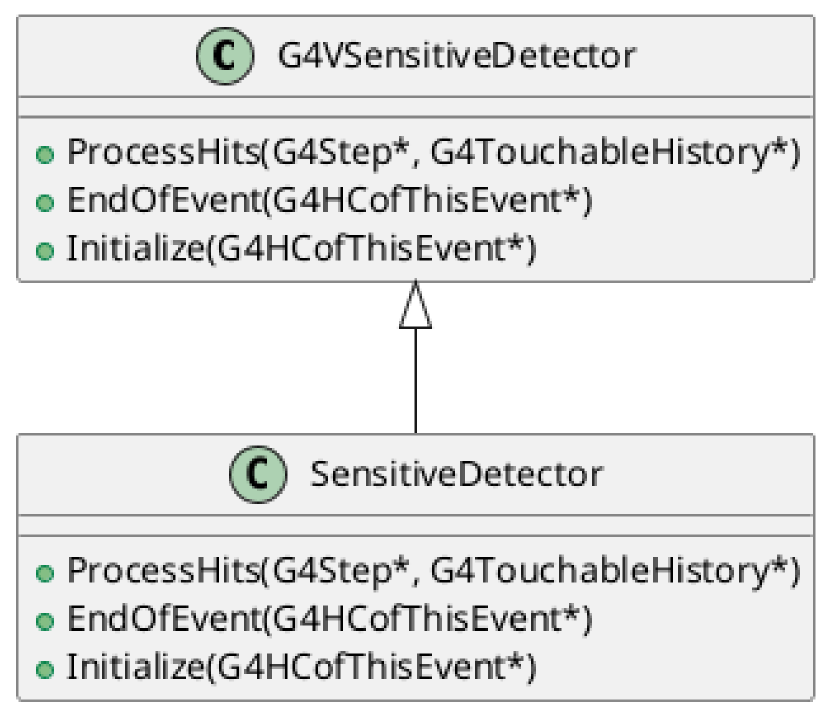
fsl with action plugins.

fsl with action plugins and MC input.

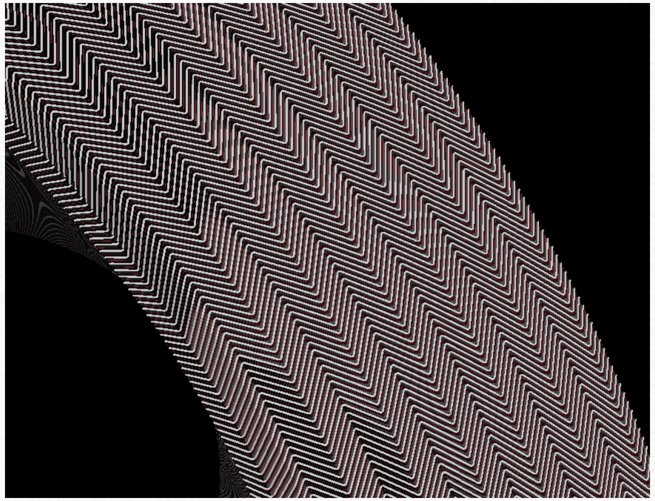
fsl with action plugins and a big event.

Sensitive Detectors

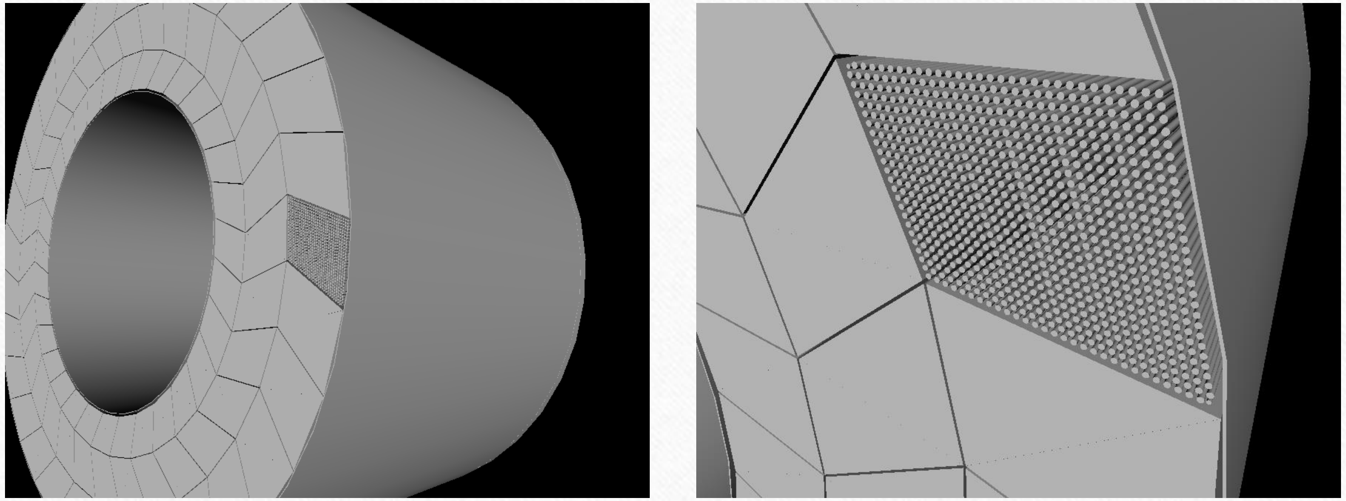
- "SensitiveDetectors" are the most important plugin because this is what you use to get the primary output (hits) out of the simulation!


G4VSensitiveDetector
ProcessHits(G4Step*, G4TouchableHistory*)
EndOfEvent(G4HCofThisEvent*)
Initialize(G4HCofThisEvent*)
SensitiveDetector
ProcessHits(G4Step*, G4TouchableHistory*)
EndOfEvent(G4HCofThisEvent*)
Initialize(G4HCofThisEvent*)

We distinguish sensitive volumes



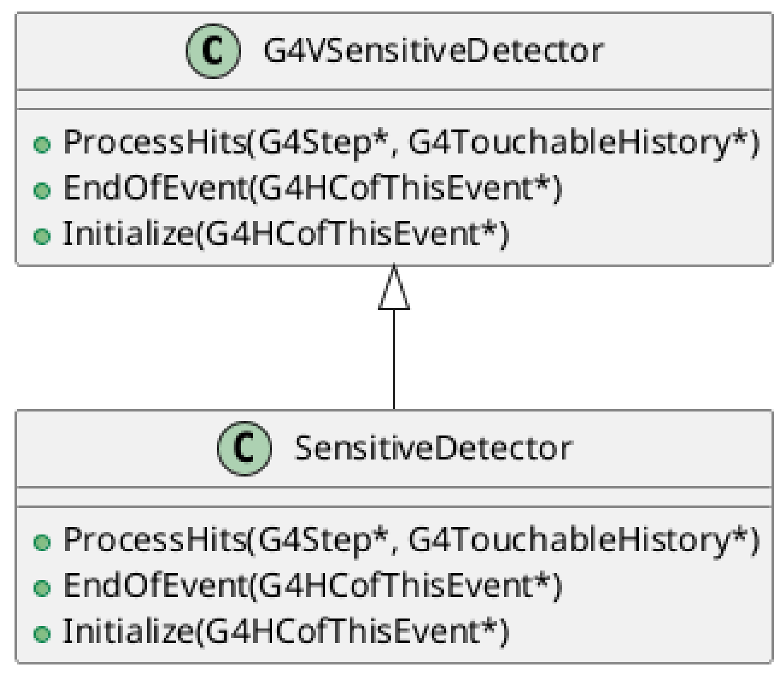
Liquid Argon in the gap



Xenon gas mixture in the straw tubes

Procedure

- Define a specialized hit to contain specialized information about what happens in the volume. For you to implement!
    - Flash of light?
    - Ionization?
    - Creation of electron-hole pairs?
- Define a SensitiveDetector that extracts or develops this information from the Step, and stores it in a hit collection.
- Instantiate it. Attach it (set a pointer) to a volume.
- Geant4 does the rest, invoking it when a particle enters a sensitive region.


C
G4VSensitiveDetector
ProcessHits(G4Step*, G4TouchableHistory*)
EndOfEvent(G4HCofThisEvent*)
Initialize(G4HCofThisEvent*)
C
SensitiveDetector
ProcessHits(G4Step*, G4TouchableHistory*)
EndOfEvent(G4HCofThisEvent*)
Initialize(G4HCofThisEvent*)

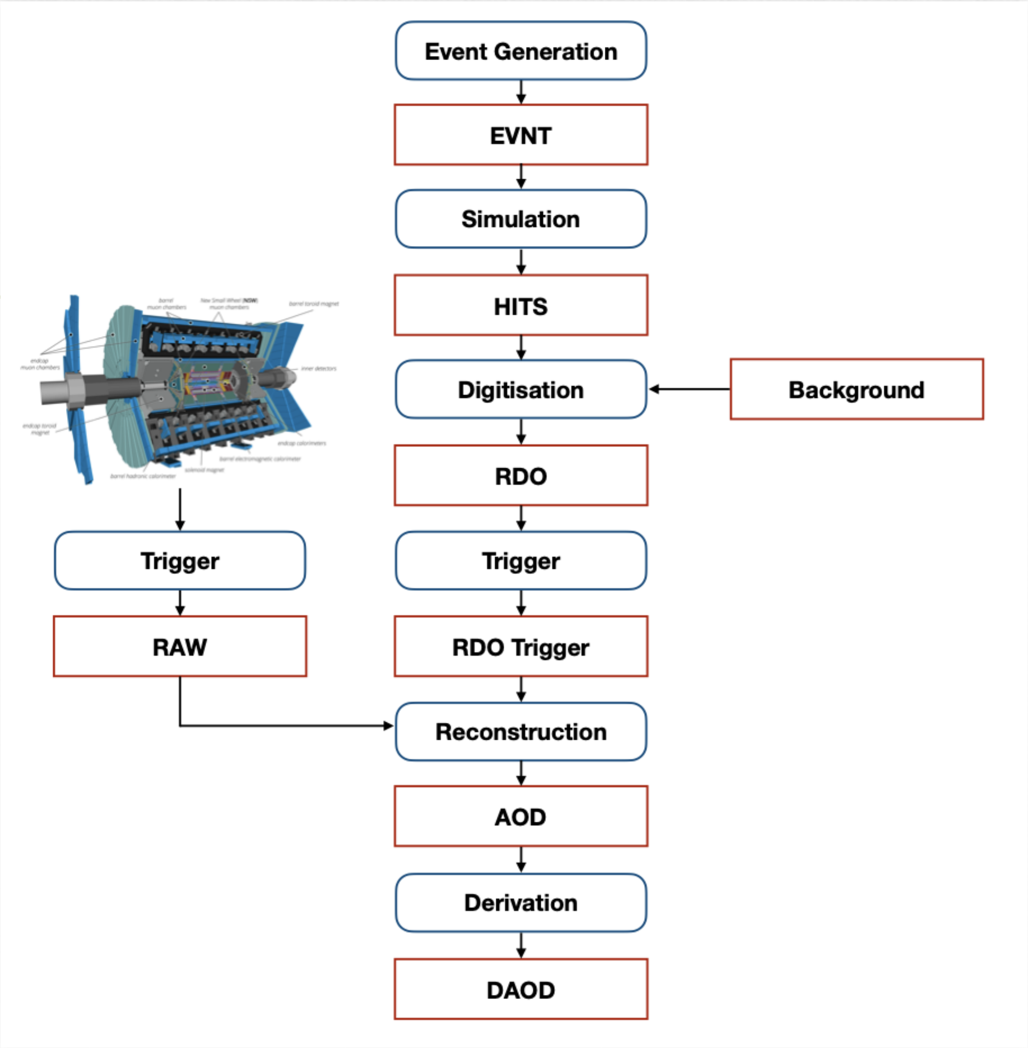
Demonstration #10

fsl w/ TRT.db and add sensitive detector

Simulation in Practice

One small link in the chain.

- Simulation is the most CPU intensive part of the chain but it is not the only part.
- Digitization: applies a software emulation of signal processing.
- Pileup events are superimposed on the hard scattering.
- After a simulation of the trigger selection the simulated data is formatted in the same way as data from the detector
- It is ingested in the same way by reconstruction software (except that truth information is also passed along)



Event Generation

EVNT

Simulation

HITS

Digitisation

Background

RDO

Trigger

RDO Trigger

Reconstruction

AOD

Derivation

DAOD

Trigger

RAW

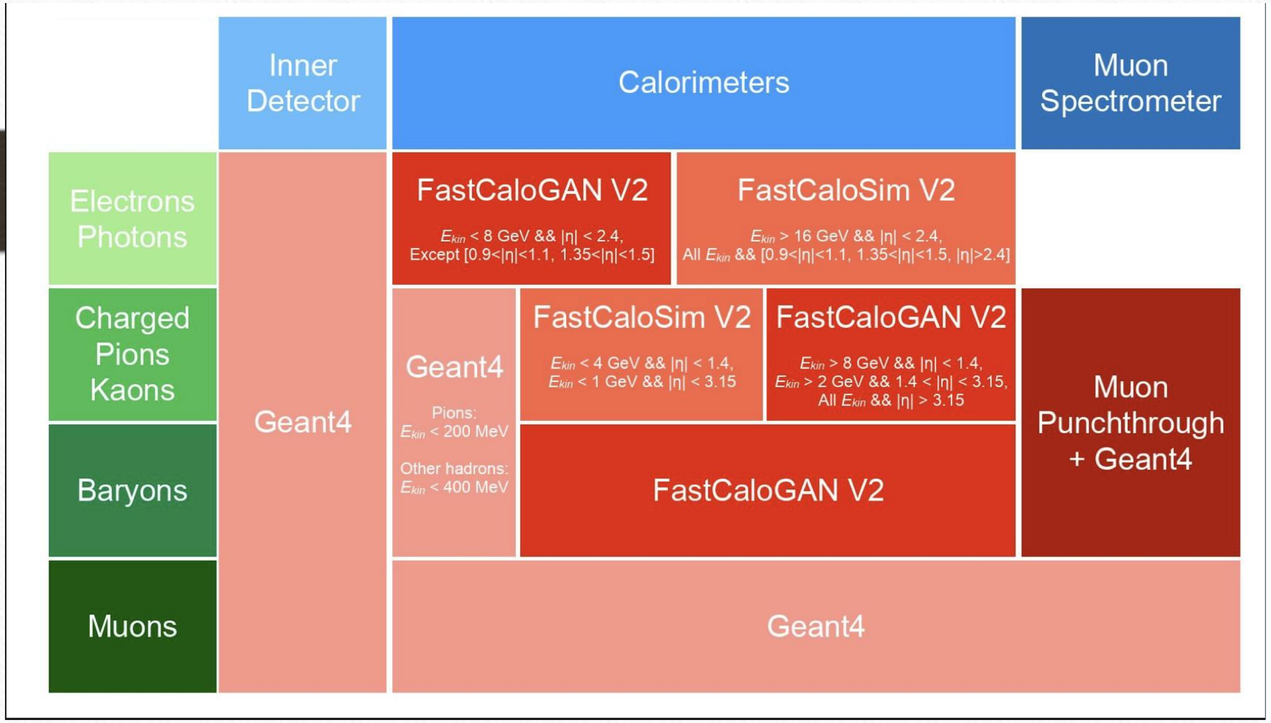
Fast simulation

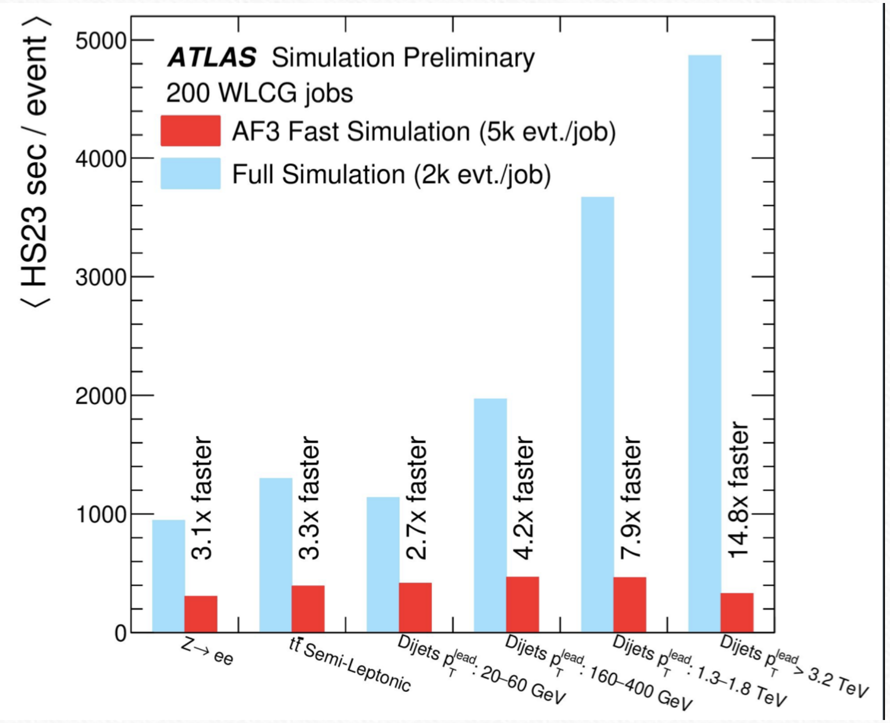
- ~ 2/3 of ATLAS simulation uses parameterization of showers in the calorimeter.
- These methods are trained on full simulation.

- FastCaloSim:
  - Parameterization of shower longitudinal and transverse shapes

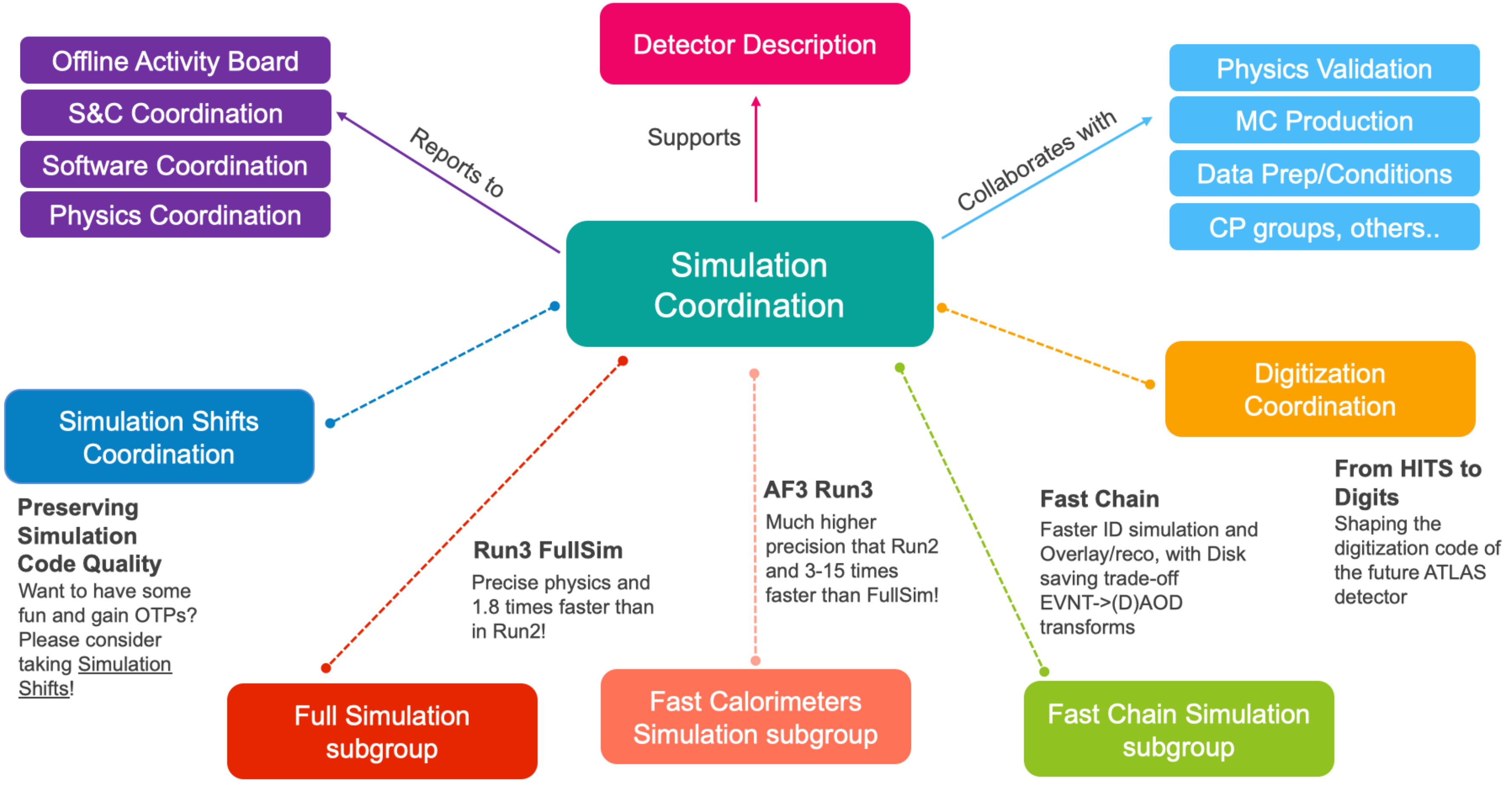
- FastCaloGAN
  - Uses deep generative models to generate calorimeter showers.
  - Uses ~600 GANs to simulate each particle type over a wide momentum range.

Fast Simulation


Inner Detector
Calorimeters
Muon Spectrometer
Electrons Photons
Charged Pions Kaons
Baryons
Muons
Geant4
FastCaloGAN V2
E_{kin} < 8 GeV && |η| < 2.4,
Except [0.9<|η|<1.1, 1.35<|η|<1.5]
FastCaloSim V2
E_{kin} > 16 GeV && |η| < 2.4,
All E_{kin} && [0.9<|η|<1.1, 1.35<|η|<1.5, |η|>2.4]
Geant4
Pions: E_{kin} < 200 MeV
Other hadrons: E_{kin} < 400 MeV
FastCaloSim V2
E_{kin} < 4 GeV && |η| < 1.4,
E_{kin} < 1 GeV && |η| < 3.15
FastCaloGAN V2
E_{kin} > 8 GeV && |η| < 1.4,
E_{kin} > 2 GeV && 1.4 < |η| < 3.15,
All E_{kin} && |η| > 3.15
FastCaloGAN V2
Muon Punchthrough + Geant4
Geant4



Simulation is a large-scale software endeavor



Offline Activity Board

S&C Coordination

Software Coordination

Physics Coordination

Reports to

Detector Description

Supports

Physics Validation

MC Production

Data Prep/Conditions

CP groups, others..

Collaborates with

Simulation Coordination

Simulation Shifts Coordination

Preserving Simulation Code Quality
Want to have some fun and gain OTPs? Please consider taking Simulation Shifts!

Digitization Coordination

From HITS to Digits
Shaping the digitization code of the future ATLAS detector

Run3 FullSim
Precise physics and 1.8 times faster than in Run2!

AF3 Run3
Much higher precision that Run2 and 3-15 times faster than FullSim!

Fast Chain
Faster ID simulation and Overlay/reco, with Disk saving trade-off EVNT->(D)AOD transforms

Full Simulation subgroup

Fast Calorimeters Simulation subgroup

Fast Chain Simulation subgroup

Thank you!

Backup

He didn’t get excited at all. He took the this-is-unheard-of-but-not-really-serious attitude of a street fakir whose mechanical dingus flops during a demonstration.