(Geant4) Simulation And You

Zach Marshall

MC & Simulation Tutorial

28 Sept 2015

Example Time!

- Usually we'll use *job transforms* like you did this morning. But *job options* are not that scary, and you should get to see them too!

- Go to a clean directory. **ALWAYS ALWAYS ALWAYS** a clean directory when you run an athena job!! Setup:

```
asetup 19.2.4.10,here
```

- Now grab the job options we want to run

```
cp /afs/cern.ch/user/z/zmarshal/public/Tutorial/jobOptions.G4Atlas.py .
```

- Run the job!

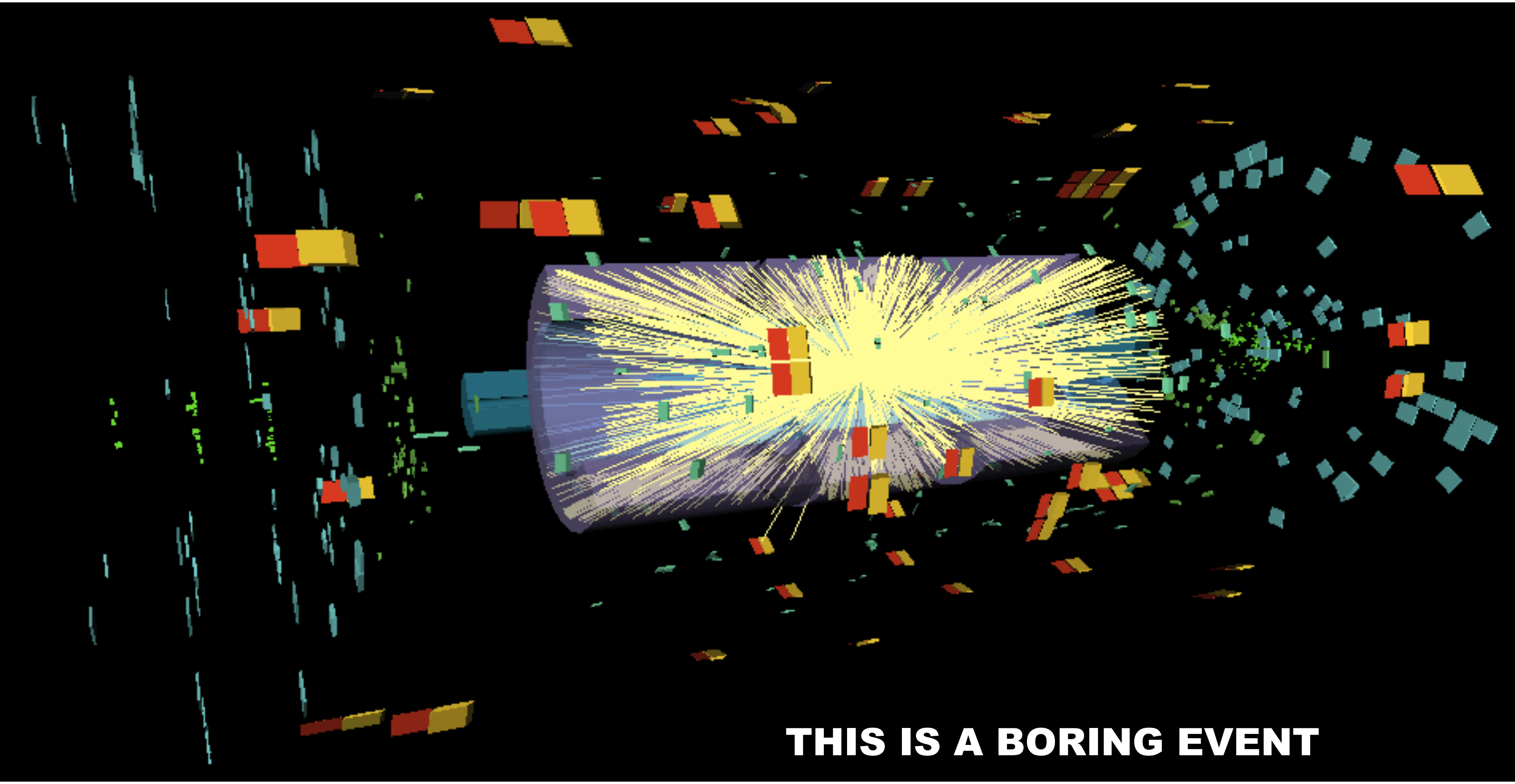
```
athena jobOptions.G4Atlas.py >& test.log &
```

- Get ready, cause this is a real job, and it takes a ***while*** to run.
- While that thing runs, I'm going to tell you everything you need to know about simulation

Why Simulate Anything

- We can (usually) only build one detector
  - What will we miss because of our detector design?
  - How would a slightly different detector affect things?
  - How will the detector stand up to radiation damage?
- Most detectors only measure voltages, currents, and times
  - It's an *interpretation* to say that such-and-such a particle caused such-and-such a signature in the detector
  - We can use simulation to correct our observables and understand our (in)efficiencies
- There is only one right answer in nature
  - What would new physics look like in our detector?
  - Could we find it under realistic conditions?
  - What are the biggest problems, and how do we ease them?
- A good simulation is the way to demonstrate to the world that you understand your detector and the physics you are studying

Write Me a Simulation for This:



Simulation Basics

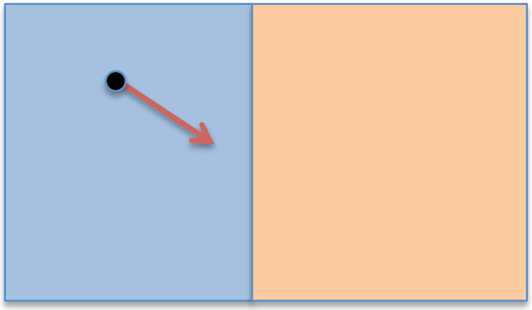
1. Break the problem up as much as possible
   - Do you understand *all the steps* of the system?
2. For each piece of the problem, write some code
   - Did you remember *all the effects* for each step?
3. Spend enough time on each piece that you get the *accuracy that you need* out of them
   - Not a moment longer!!
4. Cross your fingers and press the button

- There are two general approaches for a detector simulation
- We can simulate every little detail along the way
  - Usually we use Geant4 (GEometry ANd Tracking)
- We can go straight for the final state
  - "A pion will look like such-and-such"
  - Smear things directly
  - These are *always* home-brewed
- In ATLAS we try to do both
  - Géraldine will tell you more about the second option in ATLAS

Primer on Simulation

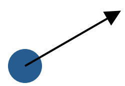
arXiv:1005.4568

- To Geant4, every problem looks like this:



- It has the concepts of:
  - Particle (if it isn't standard model, G4 has no idea about it!)
  - Material (you define everything except the elements)
  - Magnetic field (you define it at every point)
  - Physics process (you get to pick from their list!)
- It is *only a toolbox* – it's up to you to put the pieces together
  - Don't expect it to be any smarter than you are.
  - Saying "we do the simulation with Geant4" is like saying "We do the analysis with ROOT." It's true, but it's not enough to explain anything.

Simulation Step 1



π^-

Question 1: What am I looking at, and what can it do?

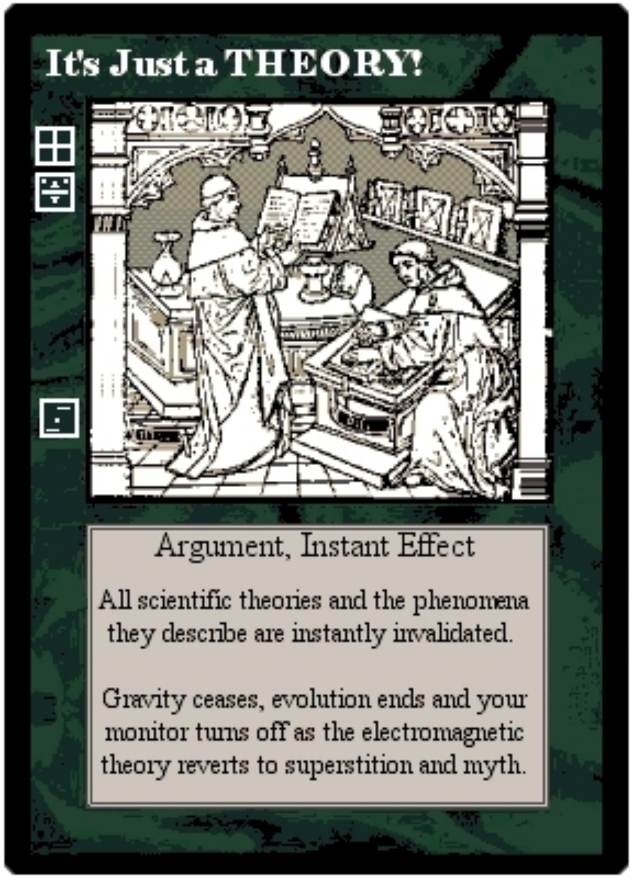
- Simulation is agnostic about the generator; It *only* cares about final state (stable) particles
- What is "final state" *depends on the experiment!!* We use ct<10 mm.
- Remember that your generator does not know *anything* about your detector geometry or magnetic field. So you don't want it to handle anything that would move through a detector element or bend significantly in a field!

Grab a particle from the 'stack' and see what type of particle it is

If necessary, figure out where it is

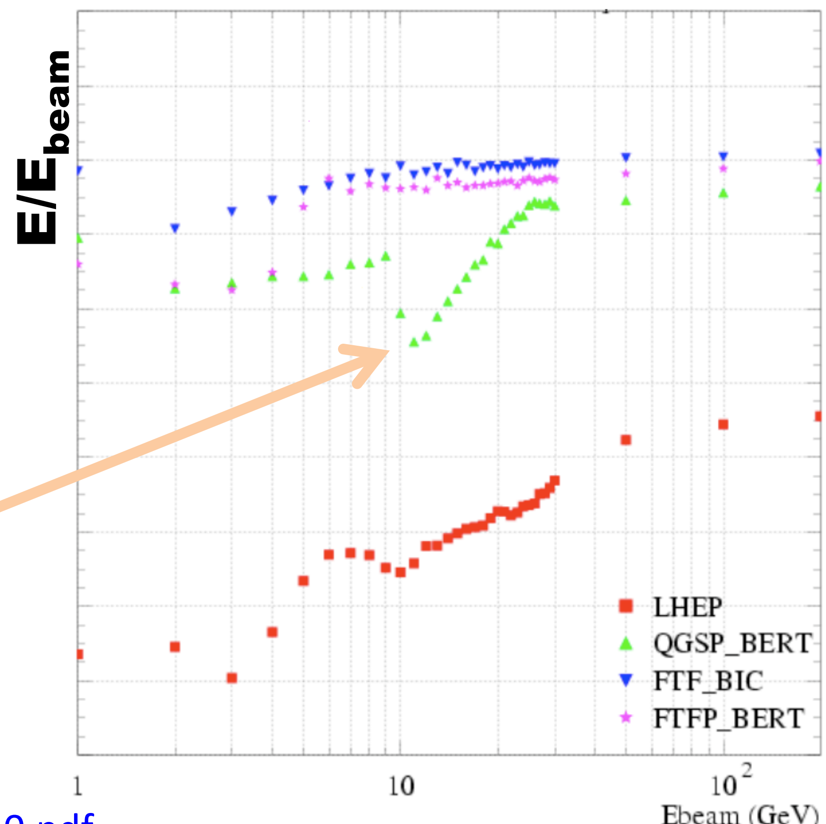
Numerical Models

- But numerical models are not!
  - All processes become *discrete*, including "transportation"
- Use phenomenological models tuned to experimental data
  - Never solve a Lagrangian!!
- Some interactions are easy
  - Photon conversion
- Some are hard to model
  - The nucleus gets its own simulation!
- Some have a variety of models
  - ATLAS covers keV to TeV physics:
    - **>9** orders of magnitude!!
  - Good in an energy range
  - Transitions can be problematic!!



Physics Models in Reality

- We use collections of models (good in an energy ranges and for a particle type) called "physics lists"
  - We use FTFP_BERT : Fritiof with a precompound model and the Bertini intra-nuclear cascade model (note: EM physics isn't even a part of the name – it's too easy).
  - Not all lists are created equal…
  - Right: tile cal test beam response
  - Note the jumps in energy…

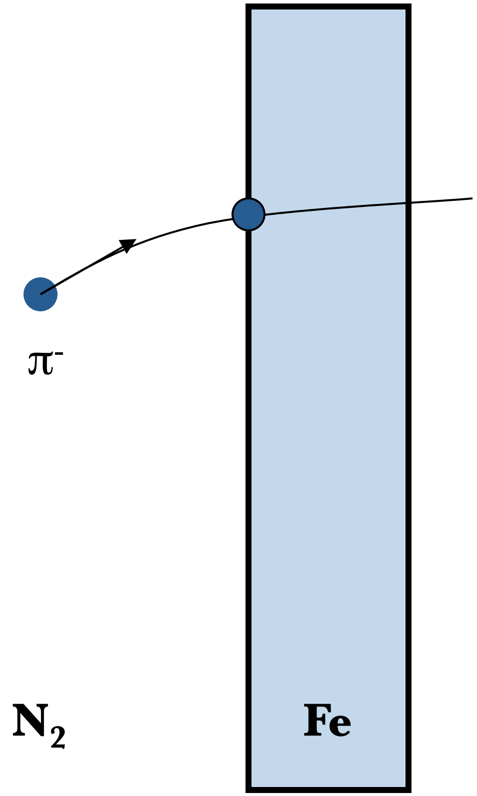


FTFP_BERT=

QGSP_BERT=

http://lcgapp.cern.ch/project/docs/noteStatusHadronic2010.pdf

Simulation Step 2

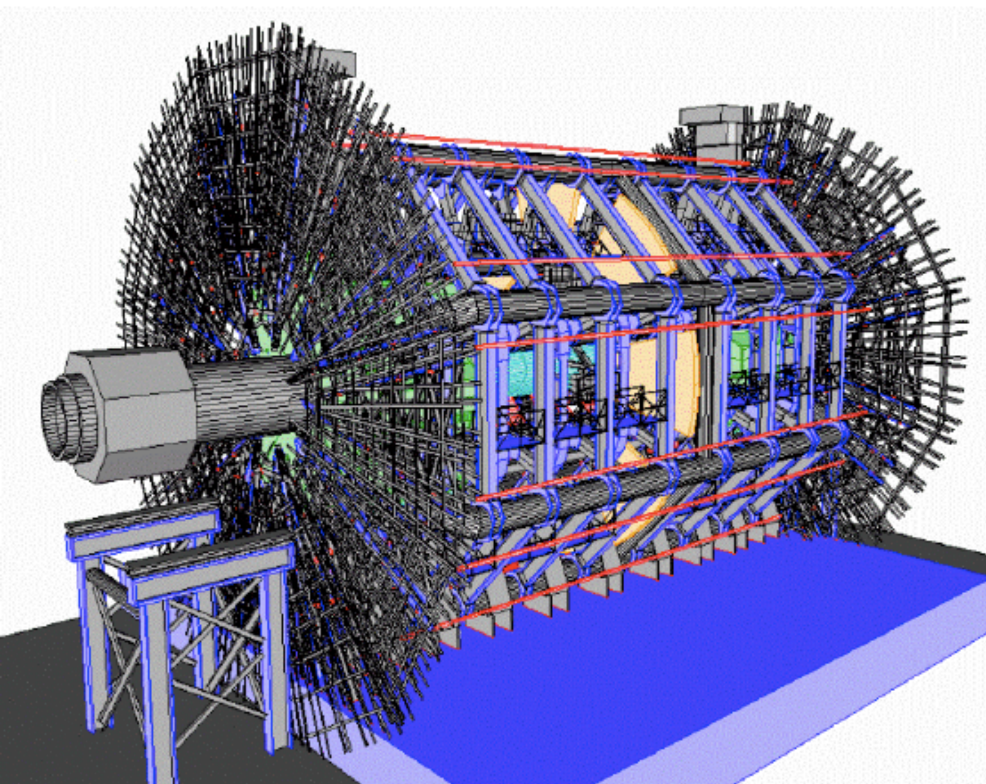


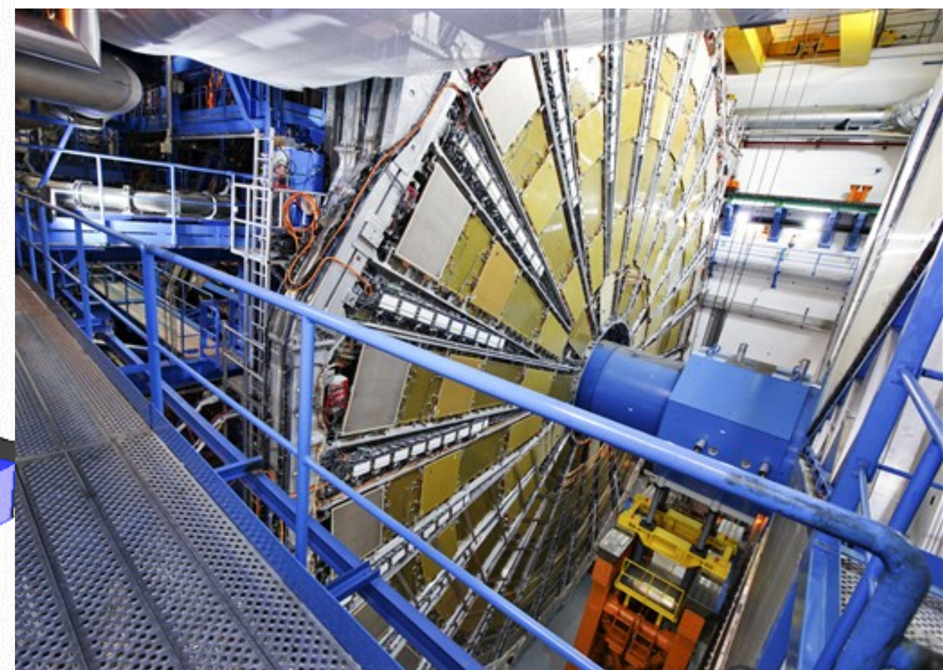
Question 2: How far may I go?

Never move farther than a volume boundary - the physics could change!!

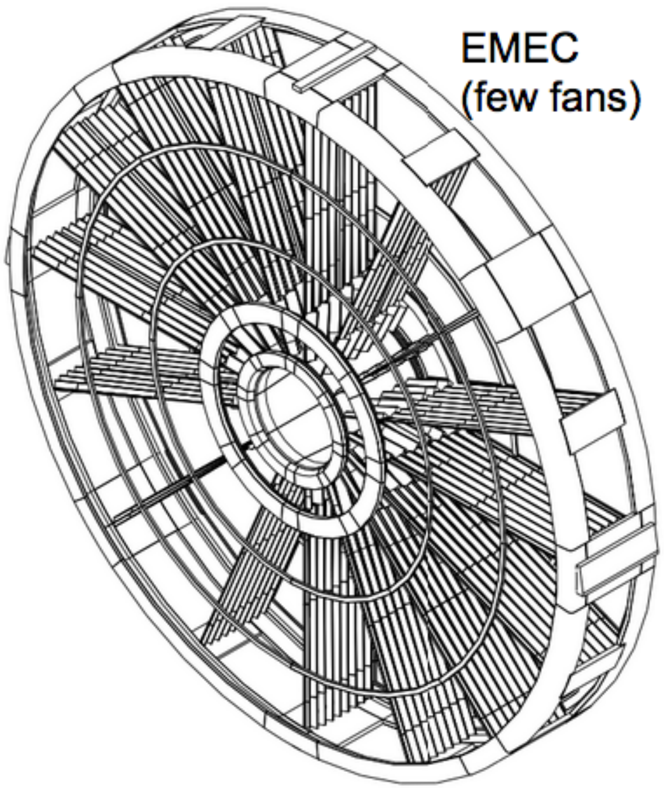
Detector Descriptions

- Need a consistent detector description
  - We are still making things more 'realistic'
    - But some things aren't worth worrying about
  - Once you are done: *just weigh the detector!*
  - Of course, with collisions, we can get fancier…

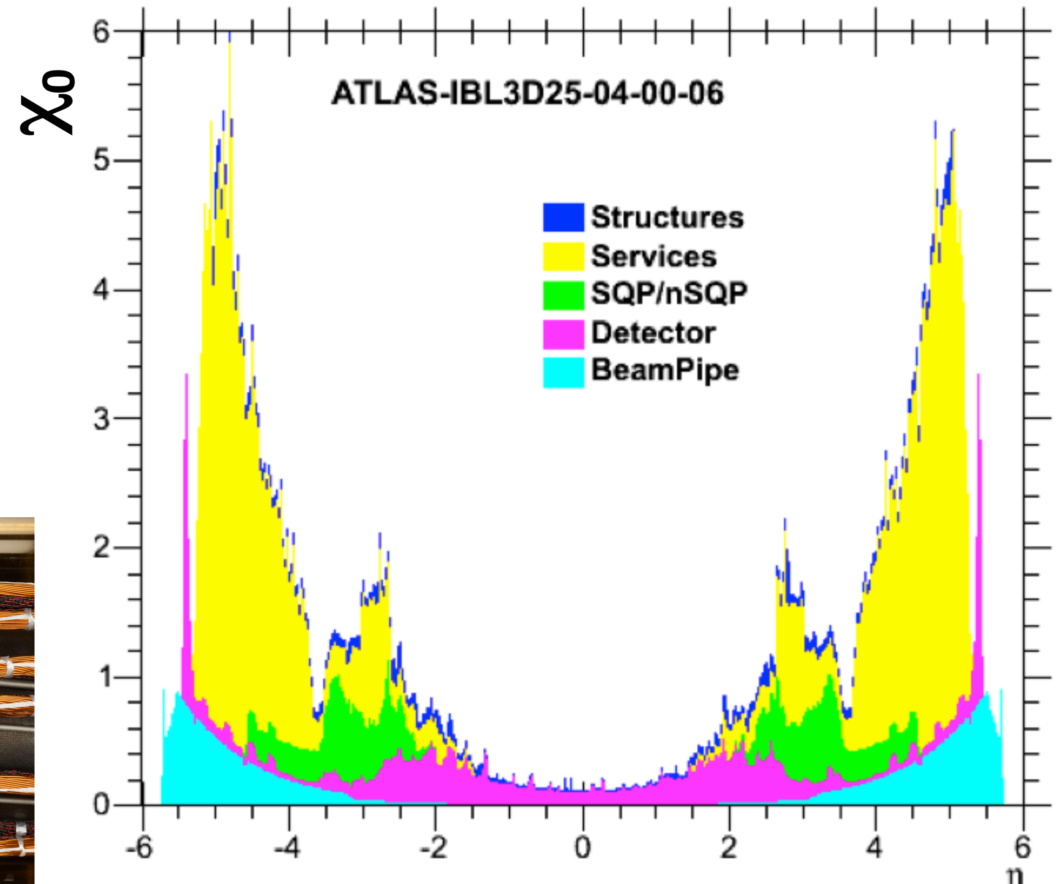




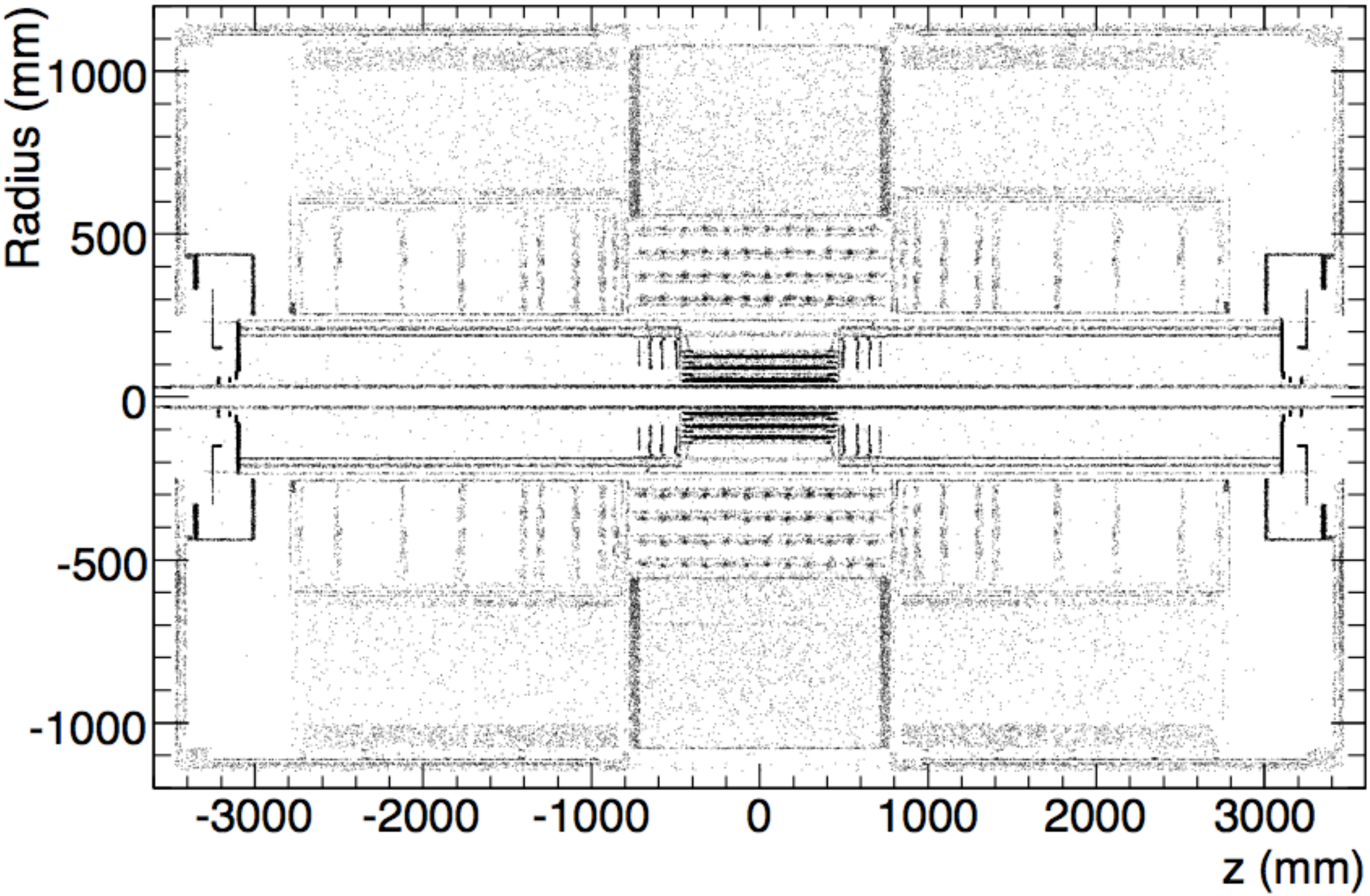
Detector Geometry



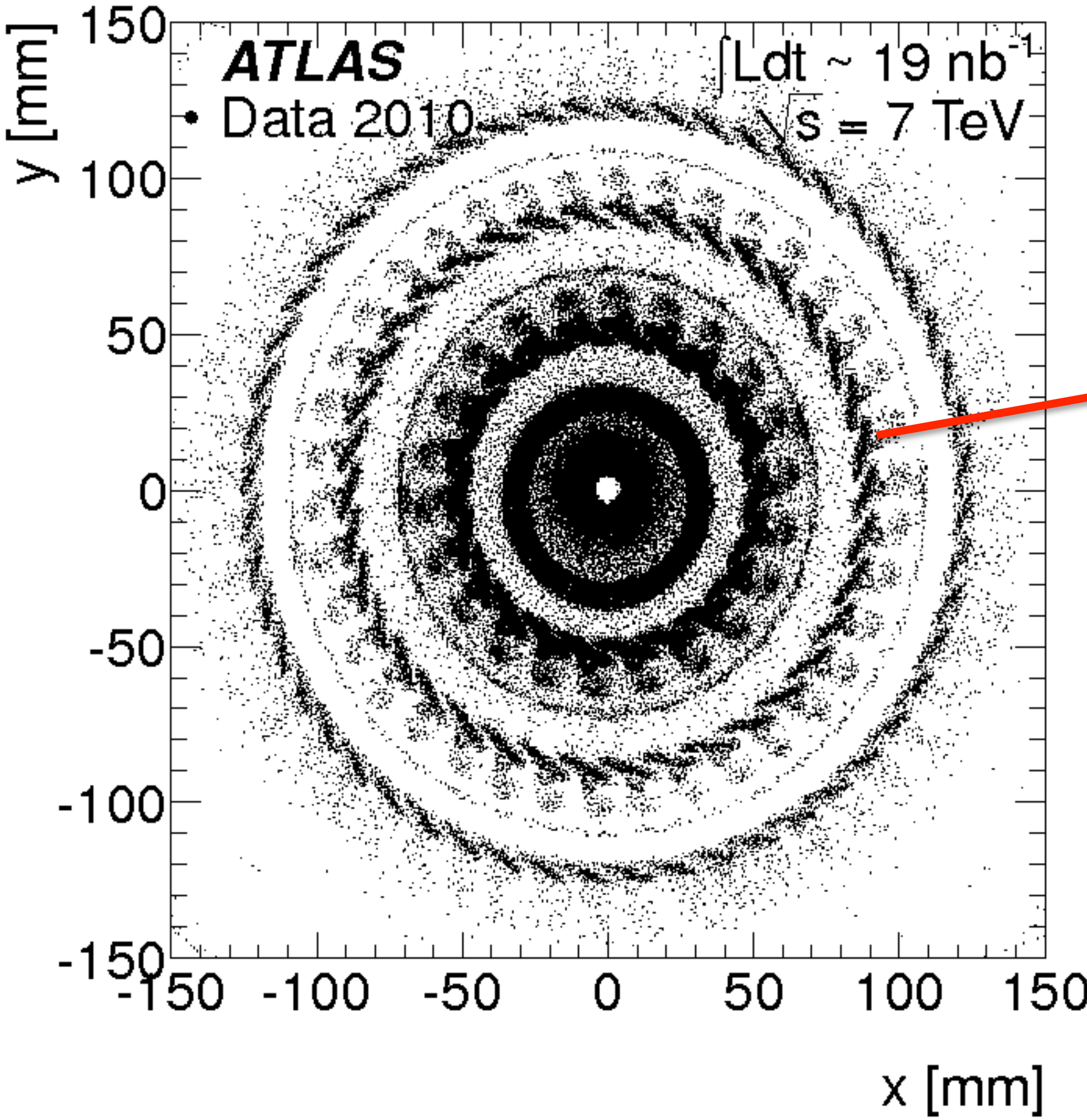
- Lots of ways to look at the detector geometry – we'll see some in the exercises ☺

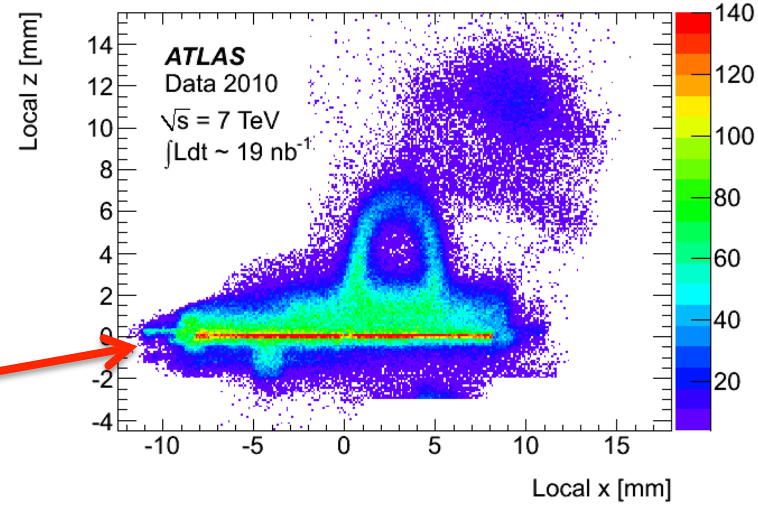


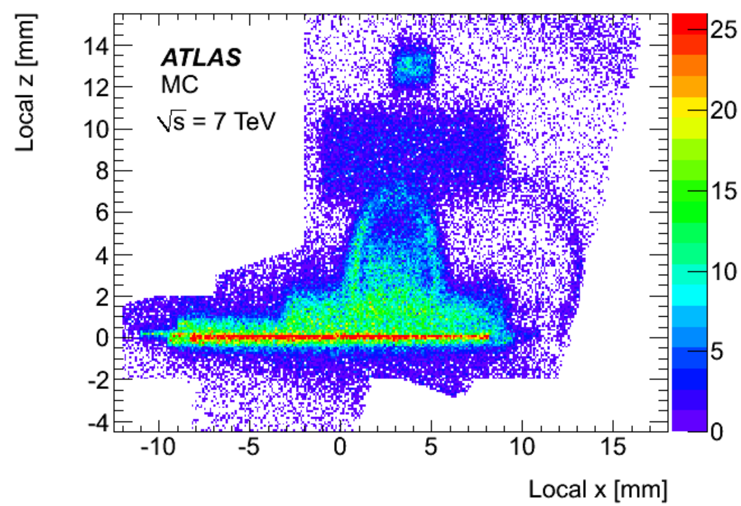
Photon Conversions



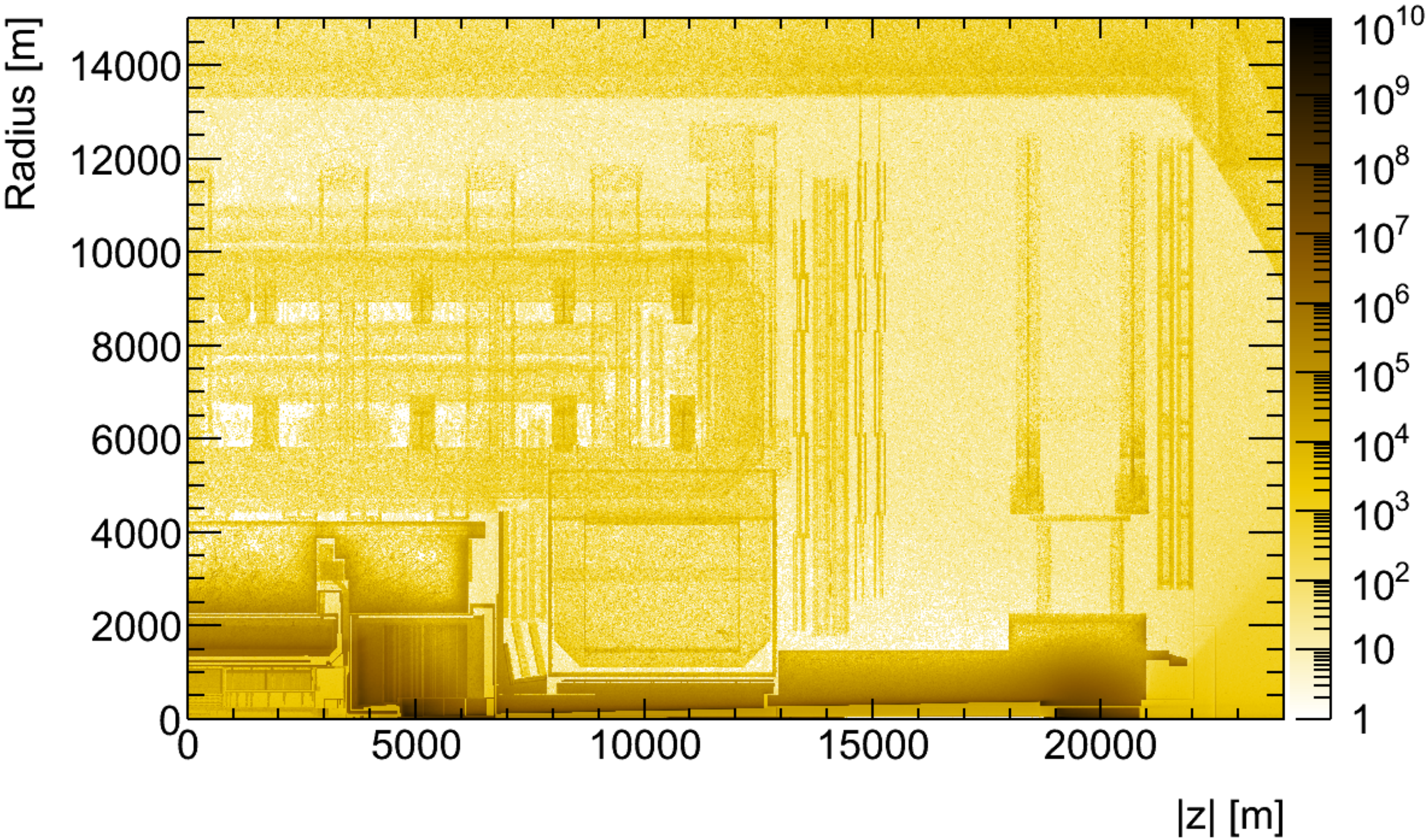
Hadronic Interactions





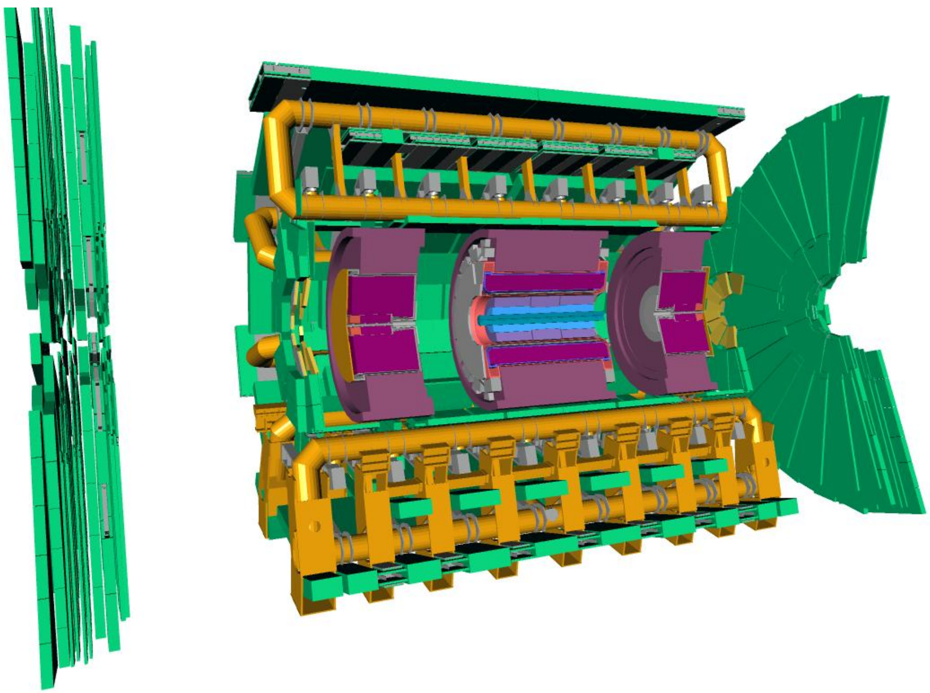


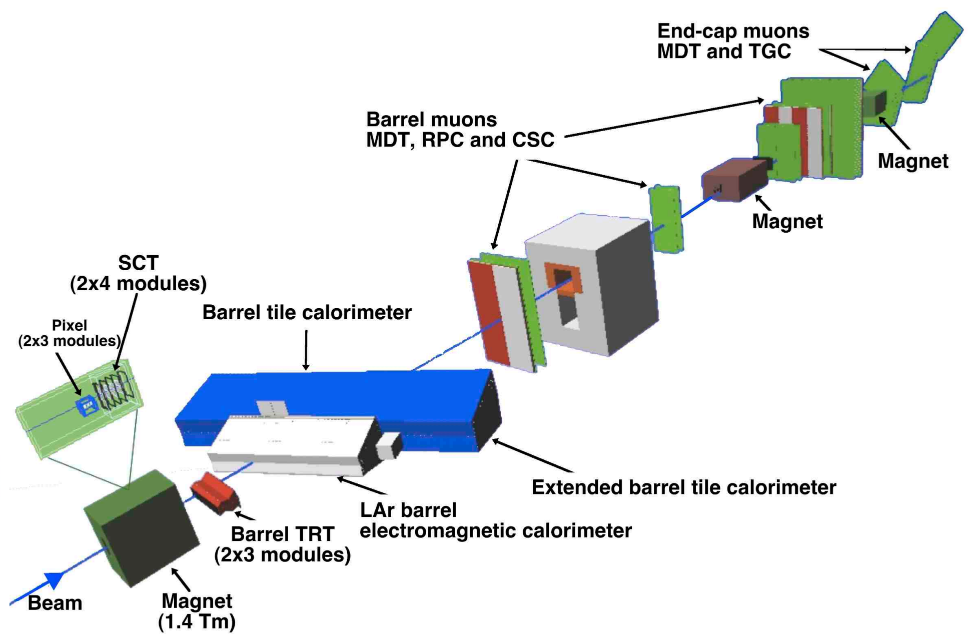
The Full Detector



Strange Geometries

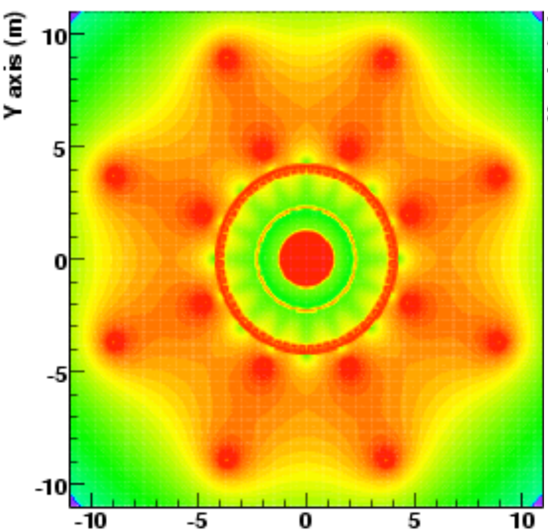
- We have commissioning layouts for cosmic ray data taking
  - See the calorimeter and muon wheel positions on the left there?
- We have layouts for test beams, IBL, NSW, Lucid, ZDC, ALFA
- If you want to simulate it, you had better write a geometry!

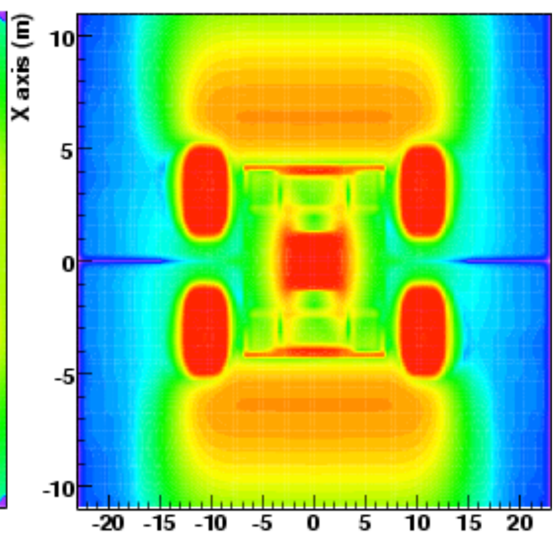


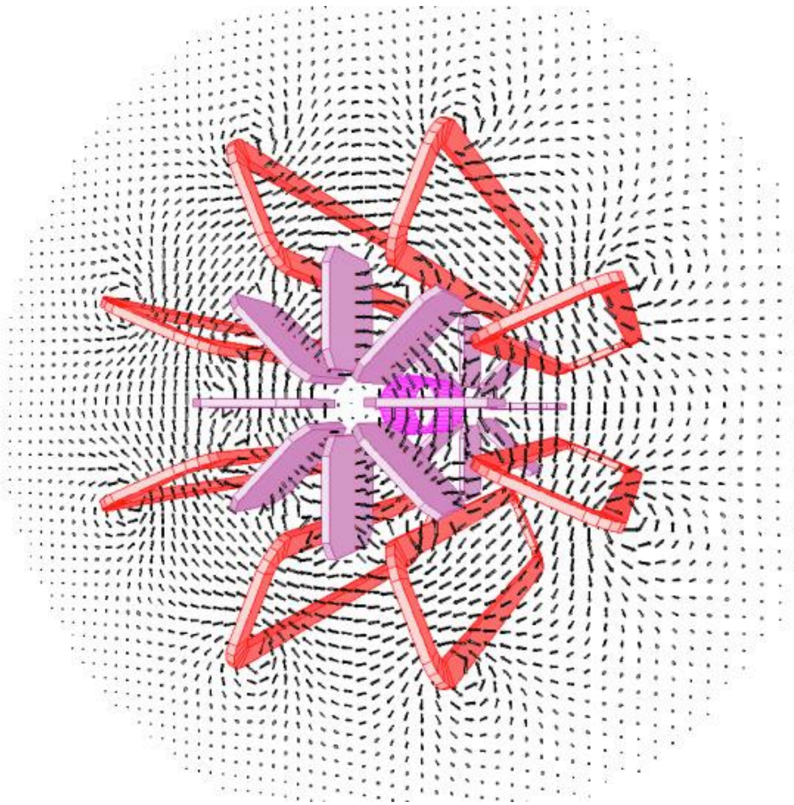


The Magnetic Field

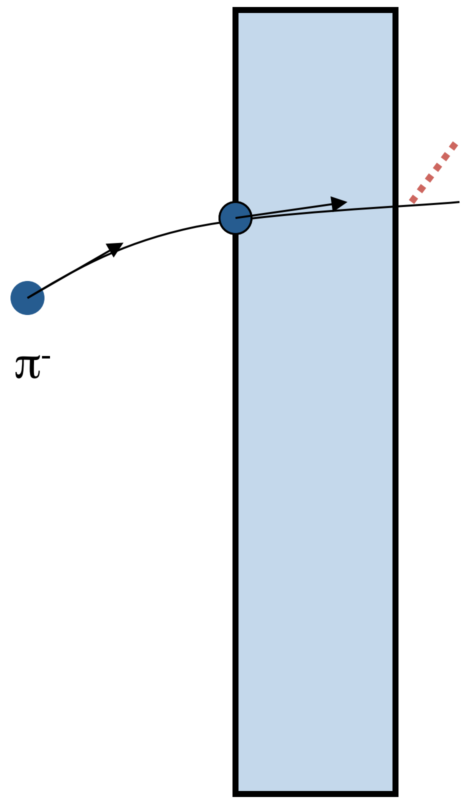
- No hope for simulating on the fly - must have a *map*
  - Do they match *reality*?
  - Do they match the *geometry*? (I always love finding a magnetic field map that has the wrong solenoid position…
- ATLAS keeps a handful of hall cubes on the detector to measure the field
  - Parameterizes elsewhere







Simulation Step 3



Question 3: What will happen next?

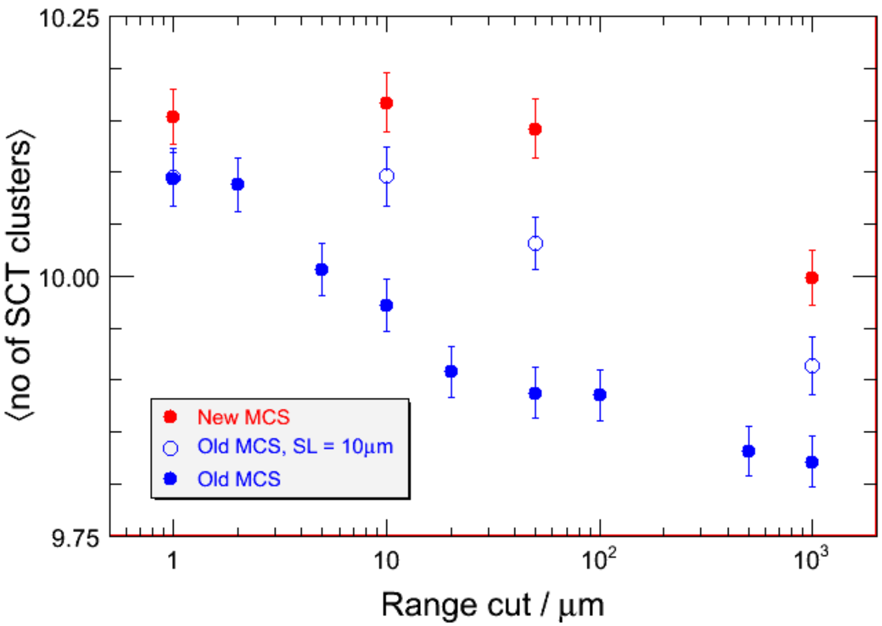
Check on all physics models. For a pion, this means multiple scattering, bremsstrahlung, nuclear interactions, decay, ionization…

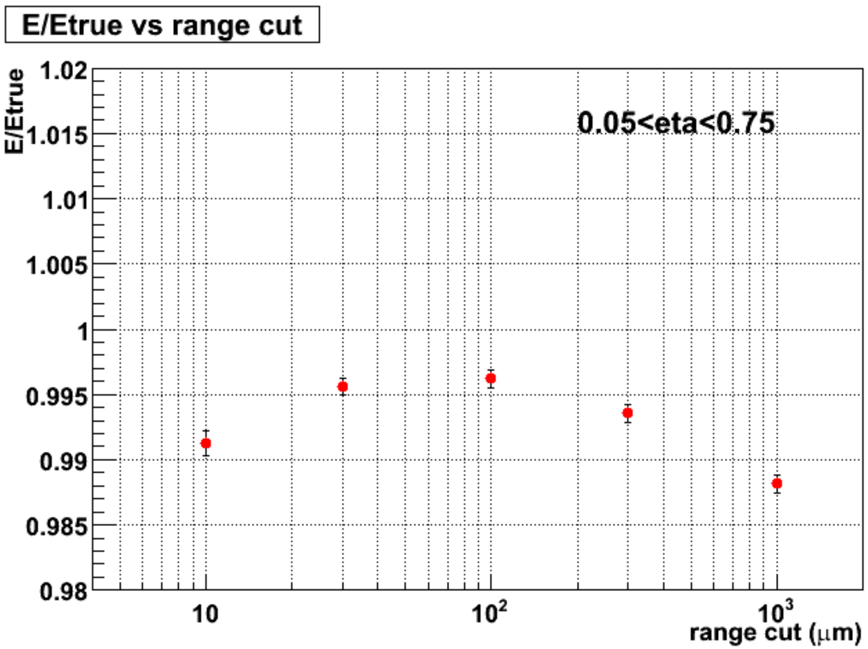
Add any new particles to the stack

Adjust energy and momentum accordingly

Making More Particles

- If there is an interaction, generate all the secondaries
  - EM processes have a "range cut" – $e^+/e^-/\gamma$ that have an average range below that *distance* are never created. Standard cut is 1mm (most of ATLAS)
  - Incorrect setting of this range cut caused a major production problem in the CSC days (for those of you old enough to remember…)
  - Now we are meant to be far less sensitive to it!
- In general, the models of what secondaries are produced in an interaction are well-educated guesses



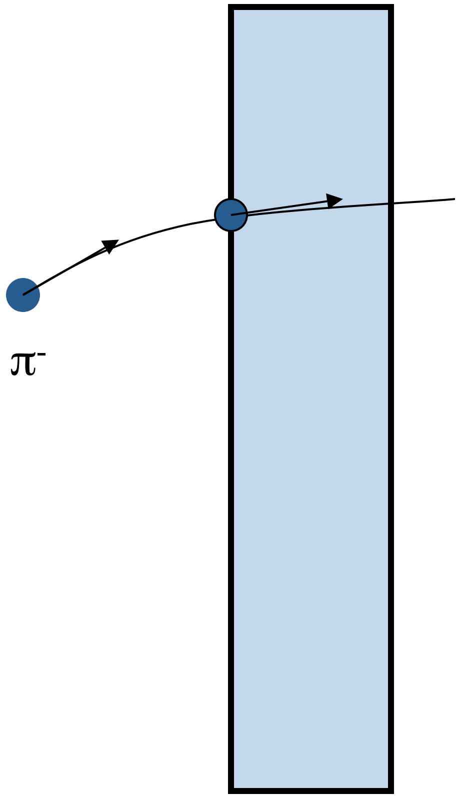


What Really Happens

- Each iteration is called a “step”
- The number of steps dictates the speed of the simulation
  - The best code changes get us ~3-10% speed ups; the best physics changes get us 20-50% speed ups
  - Sim time goes with energy, not E_T, so it is very important how far forward you simulate particles – but you need to be careful to not harm detector response!
- Most time just moving stuff around in the calorimeter
  - Here is the table for 50 ttbar events
- 50% of your simulation time is spent moving $e^{\pm}$ and γ below 10 MeV

Process	**Inner Detector**	**calorimeter**	**Muon System**
Transportation	1.76×10^6	1.46×10^7	2.31×10^5
MSC	2.31×10^5	1.48×10^7	5,200
Photoelectric Effect	6,760	1.37×10^6	2.32×10^5
Compton Scattering	14,800	1.66×10^6	5.03×10^5
Ionization	1.03×10^5	4.81×10^6	9.71×10^5
bremsstrahlung	6,060	1.22×10^6	1.92×10^5
Conversion	416	86,800	18,100
Annihilation	271	87,000	18,500
Decay	212	1,670	402
Other Hadronic Interaction	2,190	6.66×10^5	1.23×10^5
Other Process	426	25,400	5,720
Total	2.13×10^6	3.93×10^7	2.69×10^6

Simulation Step 4



Question 4: Anything else to do?

The user is allowed a hook at the end of this “step” to perform any necessary action, for example to **make a record of energy deposition**

All particles are tracked to zero energy or their exit of your world (this is different from the old days of simulation…)

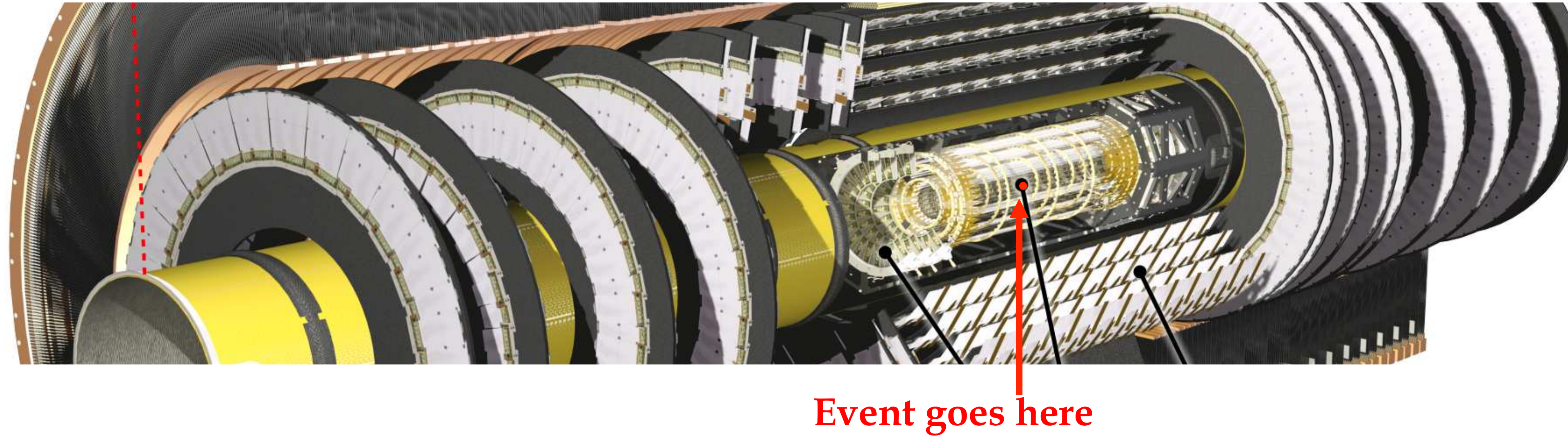
Truth in ATLAS

- Of course, the most interesting interactions in the detector need to be saved to understand how the reconstruction does
- ATLAS combines several approaches to ensure everything that needs to be saved, is
  - Records of high-energy particles crossing several boundaries
    - Outer edges of the tracker, calorimeter, and muon system
  - Records of the *vertices* of the most interesting interactions
    - High energy photons from muons in the calorimetry - key for calorimeter energy correction studies
    - Just about every high-energy interaction in the inner detector we can think of - critical for track efficiency corrections
  - Records of the "true" energy deposited in some volumes
- At the end (in reco) these different records are stitched back together into what ever is required
  - Track truth is not a concept in G4
  - This is not easy!!

https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/MonteCarloTruthTaskForce

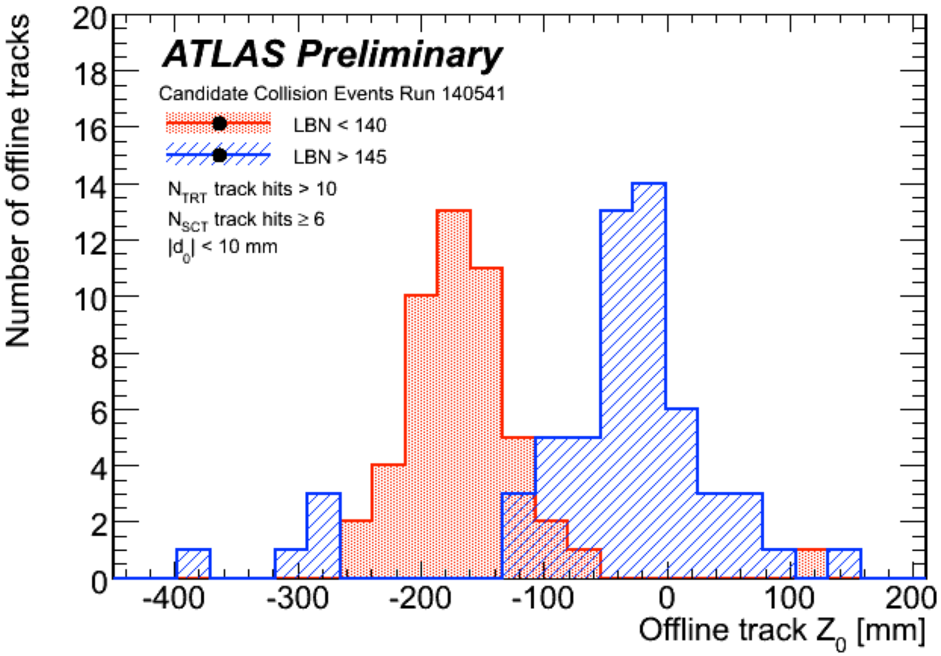
Working from EVNT

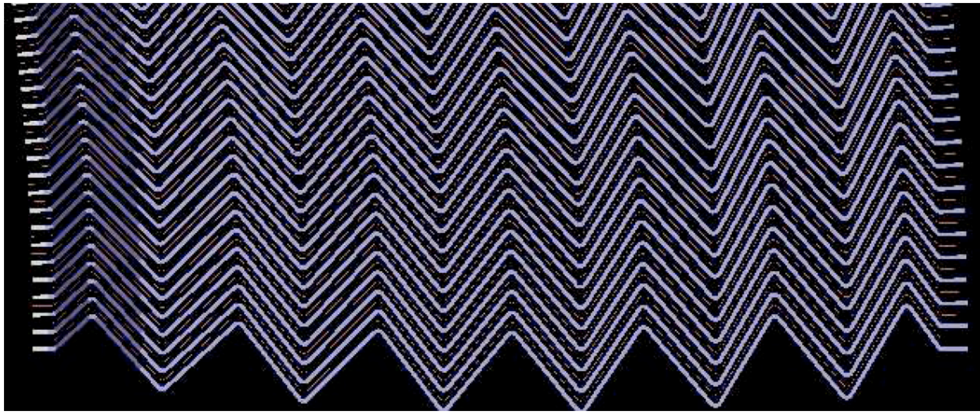
- Pick your beam spot size
  - This is the first place *conditions* enter the MC chain
- Put down the event and begin the loop
- Move everything until it stops
  - Positrons annihilate; Neutrons are captured; Muons can decay; They may also exit the cavern…



Conditions in MC

- ATLAS has to deal with many types of "conditions"
  - Beam spot, misalignments, which parts are masked or disabled…
- To save time and energy, conditions are applied as late as possible
  - Channels can be masked long after simulation finishes!
  - But moving the interaction region is a whole other story… We must get the position of the beamspot right already in the simulation!
- Some coarse misalignments already have to be applied in the simulation
  - We have to be VERY careful that things don't overlap! G4 could get lost…





Exaggerated (10x) sagging in the EM barrel calorimeter

Ok finally…

BACK TO THAT EXAMPLE

Job Options Part 1

```
## Algorithm sequence
from AthenaCommon.AlgSequence import AlgSequence
topSeq = AlgSequence()
```

Boilerplate (don't worry about it)

```
## Detector flags
from AthenaCommon.DetFlags import DetFlags
DetFlags.ID_setOn()
DetFlags.Calo_setOn()
DetFlags.Muon_setOn()
DetFlags.Truth_setOn()
```

Turn on or off bits of the detector

```
## Global conditions tag
from AthenaCommon.GlobalFlags import jobproperties
jobproperties.Global.ConditionsTag = "OFLCOND-RUN12-SDR-21"
```

Detector conditions
(always copy this from somewhere)

```
## AthenaCommon flags
from AthenaCommon.AthenaCommonFlags import athenaCommonFlags
athenaCommonFlags.PoolEvgenInput = ['/afs/cern.ch/atlas/offline/ProdData/16.6.X/16.6.7.Y/ttbar_muplusjets-pythia6-7000.evgen.pool.root']
athenaCommonFlags.PoolHitsOutput = "test.HITS.pool.root"
athenaCommonFlags.EvtMax = 3
```

Input file, output file, number of events

```
## Simulation flags
from G4AtlasApps.SimFlags import simFlags
simFlags.load_atlas_flags()
```

We are using the ATLAS detector ☺

Job Options Part 2

```
## Layout tags: see simFlags.SimLayout for allowed values
## Set a specific layout tag:
# See http://twiki.cern.ch/twiki/bin/view/AtlasComputing/AtlasGeomDBTags
simFlags.SimLayout='ATLAS-R2-2015-03-01-00'
simFlags.RunNumber = 222510
```

Which *version* of the detector?

```
## Set the LAr parameterization
#simFlags.LArParameterization = 3
simFlags.CalibrationRun.set_Off()
```

Turn on “frozen showers”
(wait for Géraldine’s talk)
Turn off calibration hits

```
## Use verbose G4 tracking
#from G4AtlasApps import callbacks
#simFlags.InitFunctions.add_function("postInit", callbacks.use_verbose_tracking)
```

See what G4 is *really* doing!

```
## Dump the input event that you've read
#from TruthExamples.TruthExamplesConf import DumpMC
#topSeq += DumpMC()
```

Print out what is in the input file

```
## Add the G4 sim to the alg sequence after the generator
from G4AtlasApps.PyG4Atlas import PyG4AtlasAlg
topSeq += PyG4AtlasAlg()
print topSeq
```

More boilerplate
(don’t worry about this)

Check Out that Log File!

- ***Never be afraid of a log file***
- Stay calm and look for something you recognize
- The first loooong part of *this* log file is just going piece by piece through the detector, building it and getting it ready
- You can find the version of Geant4 we use:

```
Geant4 version Name: geant4-09-06-patch-04    (30-January-2015)
```

- You can find all the physics being initialized:

```
<<< Geant4 Physics List simulation engine: FTFP_BERT 2.0
conv:   for  gamma    SubType= 14
      Lambda table from 1.022 MeV to 7 TeV in 77 bins, spline: 1
      ===== EM models for the G4Region  DefaultRegionForTheWorld ======
        BetheHeitler :  Emin=        0 eV    Emax=       80 GeV
     BetheHeitlerLPM :  Emin=       80 GeV   Emax=        7 TeV
```

- You can find all 600+ materials in ATLAS!

```
Index : 16     used in the geometry : Yes     recalculation needed : No
 Material : Iron
 Range cuts        :  gamma  1 mm     e-  1 mm     e+  1 mm  proton 1 mm
 Energy thresholds :  gamma  20.6438 keV    e-  1.29592 MeV    e+  1.21169 MeV proton 100 keV
```

- And the bad news about "full" simulation:

```
ChronoStatSvc       INFO Time User   : Tot= 12[min]                             #=  1
```

Transforms

- What we run in production is pretty similar!

```
Sim_tf.py
  --simulator=MC12G4
  --inputEvgenFile=/afs/cern.ch/atlas/offline/ProdData/16.6.X/16.6.7.Y/ttbar_muplusjets-pythia6-7000.evgen.pool.root
  --outputHITSFile=test.HITS.pool.root
  --maxEvents=1
  --randomSeed=10
  --geometryVersion=ATLAS-R2-2015-03-01-00
  --conditionsTag=OFLCOND-MC12-SIM-00
```

- I'll bet you can recognize all of those arguments.
- As we'll talk about tomorrow, when in doubt, ALWAYS ALWAYS ALWAYS copy a production setup
  - You can find the MC15 setups here: https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/AtlasProductionGroupMC15a

It’s the same underneath

- That log file *looks* much more complicated than the other to start, but what it’s really doing is this:

```
EVNTtoHITS 00:35:17 Py:Athena            INFO including file "runargs.EVNTtoHITS.py"
EVNTtoHITS 00:35:17 Py:Athena            INFO including file "SimuJobTransforms/skeleton.EVGENtoHIT_ISF.py"
EVNTtoHITS 00:35:17 Py:Athena            INFO including file "SimuJobTransforms/skeleton.EVGENtoHIT.py"
EVNTtoHITS 00:35:17 Py:Athena            INFO including file "SimuJobTransforms/CommonSkeletonJobOptions.py"
```

- That first thing is “just” a python file with what you had on the command line from the last slide
- The next three things are called the “skeleton”, and their job is to take the commands in the run arguments and put them into the properties *just like in the job options files.*
- That’s actually true for *all* job transforms. You have to figure out where to look, but you can *always* boil it down to a call to “athena” + “runargs.Something.py” + “SomeTransformSkeleton.py”
- Bonus: you can get the skeleton file by just doing:

```
get_files SimuJobTransforms/skeleton.EVGENtoHIT_ISF.py
```

Next Things to Try

- Turn on frozen showers!
  - How much faster is the job?
- Turn on “verbose tracking”
  - Whoa, look at that output!!
- Dig through the log file and see what you can learn
  - How big is our beam spot? ☺
  - What “truth saving strategies” are enabled? Can you guess what each one is doing?
  - Any materials or physics processes you’re surprised to see included in there?
  - Any WARNINGs that you see? (We still haven’t fixed these, so you’re not expected to, but you can still find them ☺)
  - How many tracks, secondaries, etc are there?