FullSimLight: ATLAS Standalone Geant4 Simulation

CHEP 2019

Adelaide

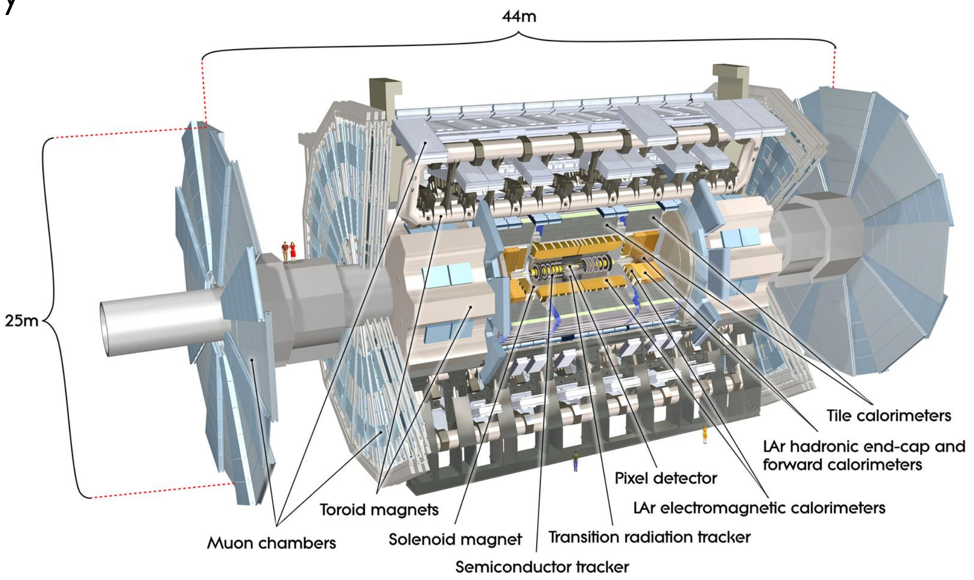
5th November 2019

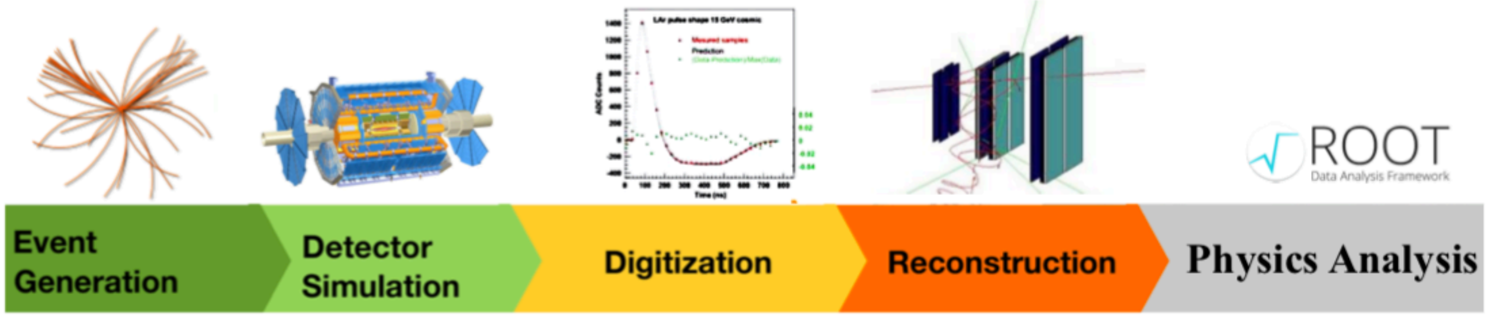
Marilena Bandieramonte
Riccardo Maria Bianchi
Joseph Boudreau

ATLAS offline software: the Full Chain

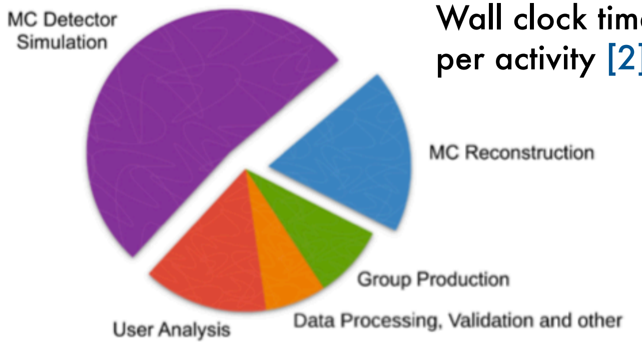
- The **Athena**[1] framework is the main software infrastructure used by the ATLAS collaboration:

1. Virtual particles are generated with specific energies and directions
   Input: particle definition file **Output**: EVGEN file
2. The interaction between these particles and the detector is simulated
   Input: EVGEN file **Output**: HITS file
3. The ATLAS detector output is calculated.
   Input: HITS file **Output**: RDO file
4. Times and voltages are reconstructed into tracks and energy deposits
   Input: RDO file **Output**: ESD file
5. Only keep the most important data from the last step
   Input: ESD file **Output**: AOD file





Wall clock time per activity [2]



The ATHENA framework

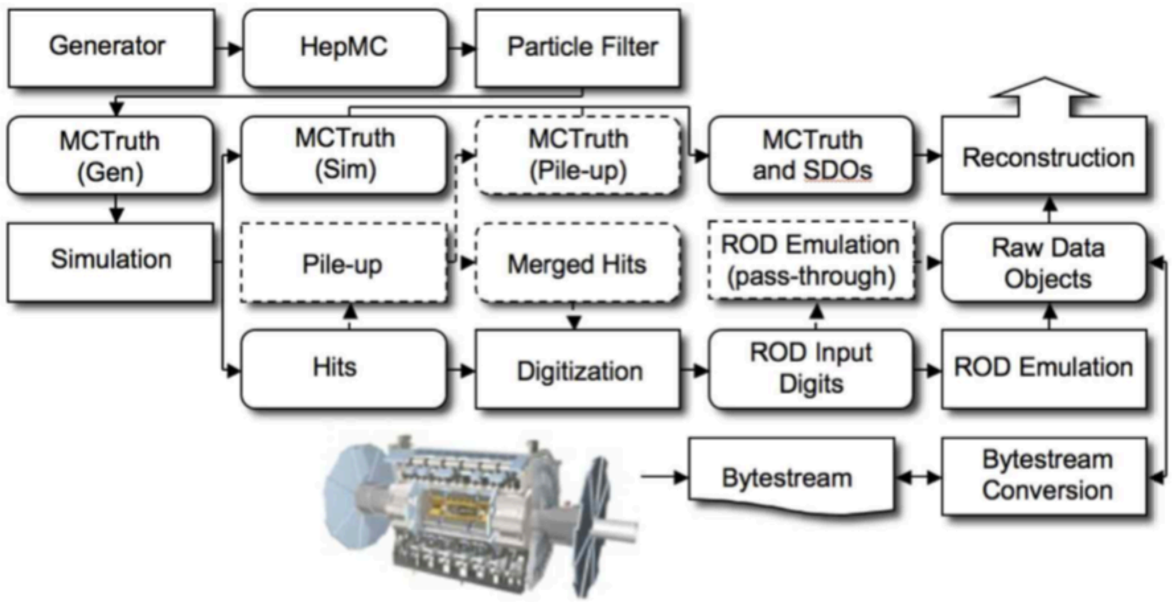
- **Athena** is based on the **Gaudi** [3] framework which was initially developed by the LHCb collaboration:
  - Gaudi supports a variety of applications through base classes and basic functionality
- Athena framework takes an **input dataset** and runs in sequence through all the events, processing them **one by one (in MT mode - one event per thread)**
  - The loop over each individual event is called the **event loop**
- Athena manages the **event loop** and the **sequence** in which different algorithms are executed inside the event loop (the Athena `AlgSequence`)
- *Athena-based programs* are **configured** depending on the required data flow of the algorithms involved in the sequence
- *Athena* uses **Python as front-end language:**
  - **Jobs are configured** through script files, so called **jobOptions:**
    - Provided as an **argument** to an Athena shell command line
    - Generated by **job transform** wrapper scripts

The ATLAS detector simulation application

- **The simulation software** for the **ATLAS** Experiment at the Large Hadron Collider is being used for **large-scale production of events** on the LHC Computing Grid (but also on HPC and clouds)

- This simulation requires **many components**, from the **generators** that simulate particle collisions, through **packages simulating the response** of the various detectors and triggers.

- All of these components come together under the **ATLAS simulation infrastructure**[4]
  - very complex framework
  - that manages many components



Towards a standalone ATLAS Full Simulation

- **Decoupling the simulation process from the experiment infrastructure** can be useful for a number of tasks
  - debugging of new features
  - validation of multithreaded vs sequential simulation code
  - optimization of algorithms for HPCs
  - could also be used to measure overhead of the framework

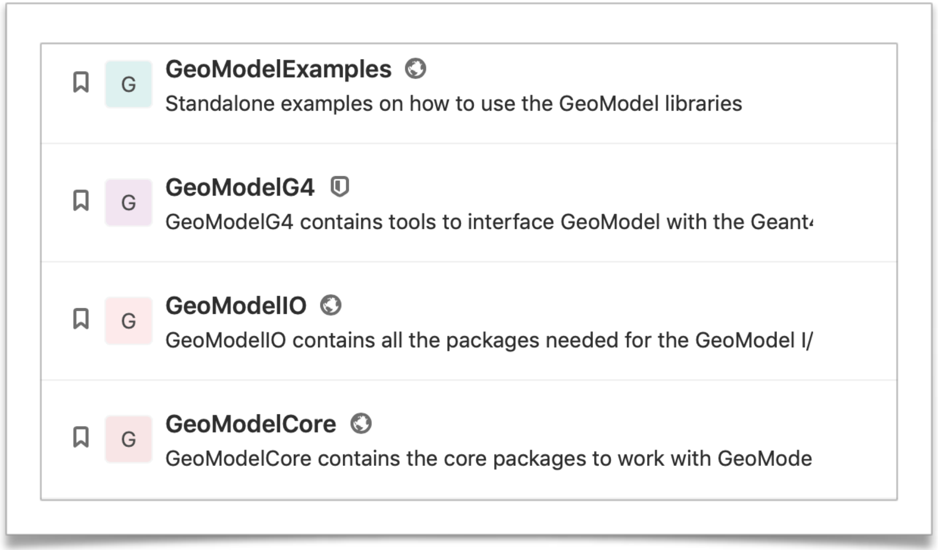
- In order to have a **standalone simulation** relevant **tools** and **data** must be **extracted** from the main framework

- **Basic building blocks** required for this kind of work:

  - Standalone access to **ATLAS geometry** — *we have it! thanks to standalone* ***GeoModel*** [5]
  - Tool to convert GeoModel geometry to Geant4 geometry - **GeoModel2G4 converter**
  - Standalone access to **ATLAS magnetic field**
  - **Need to decide how to handle I/O** :
    - Events handling
    - Conditions and Alignments
    - HITS - Sensitive Detectors

Standalone GeoModel and Geometry Persistency

- The initial **motivation** was to have a **light version of VP1** [6], a powerful visualization tool integrated in ATHENA that can access and visualise all ATLAS data:
  - Could be run outside of Athena framework
  - Not only on *CC7 (Centos7), *SLC6, but also on other platforms (i.e. macOS)
  - Can be useful for physics analysis users
- For that purpose **ATLAS geometry should be accessible outside of Athena** framework
- GeoModel is the detector description toolkit used in ATLAS:
  - "standard" shapes/nodes, are then customized with parameters taken from an "online" **Geometry DB (Oracle-based)**
  - geometry built on demand at **run-time** and kept only **in memory** (no persistent copy)



NOW, in addition:

- packages to **read-in** and **write-out** the built geometry:
  - **persistified** version of the **full GeoModel tree** as described in Athena, after the application of the parameters taken from the Geometry DB
- **GeoModel libraries can be build outside Athena**
  - **Ease the development** of new code and its **use** in standalone applications
  - **GeoModel** is used in Athena as an **externally compiled package**
- **A full toolkit suite** based on GeoModel, including **visualization** of detector geometry and **input/output** tools

GeoModel2G4 converter

- This package is meant to convert the ATLAS geometry (read and built with GeoModelKernel) into Geant4 geometry
  - **2 ATHENA packages are needed for this conversion: GeoMaterial2G4** and **Geo2G4**

Geo2G4

Dependencies:

GaudiKernel - *removed*
Control/AthenaBaseComps - *removed*
Control/AthenaKernel - *removed*
Control/SGTools - *removed*
Control/StoreGate - *removed*
DetectorDescription/GeoModel/GeoModelInterfaces – *removed*
DetectorDescription/GeoModel/GeoSpecialShapes – *to be extracted from Athena (removed for now)*
DetectorDescription/GeoModel/GeoModelUtilities – → *extracted from Athena to GeoMaterial2G4*
DetectorDescription/GeoPrimitives – *removed (also from TFPersistification and TGeoWrite)*
Simulation/G4Atlas/G4AtlasInterfaces - *removed*
Simulation/G4Atlas/G4AtlasTools - *removed*
Simulation/G4Sim/SimHelpers - *removed*
Simulation/G4Utilities/GeoMaterial2G4 → *extracted from Athena*

 # External dependencies:

- Boost - *removed*
- CLHEP - *needed for the conversion to G4*
- Geant4
- ROOT - *removed*
- XercesC - *removed*
- GeoModel

GeoMaterial2G4

Dependencies:

Control/AthenaKernel - *removed*
GaudiKernel - *removed*
Control/AthenaBaseComps - *removed*
DetectorDescription/GeoModel/GeoModelUtilities
→ *extracted from Athena to GeoMaterial2G4*

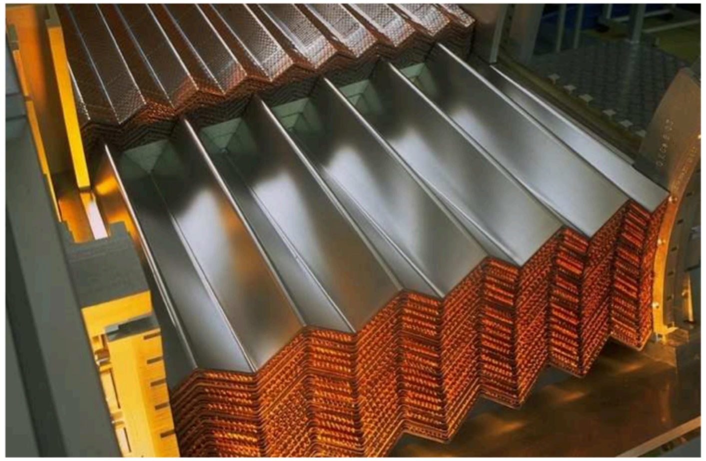
 # External dependencies:

Geant4
CLHEP - *removed*
GeoModel

Complexity of ATLAS geometry: EMEC

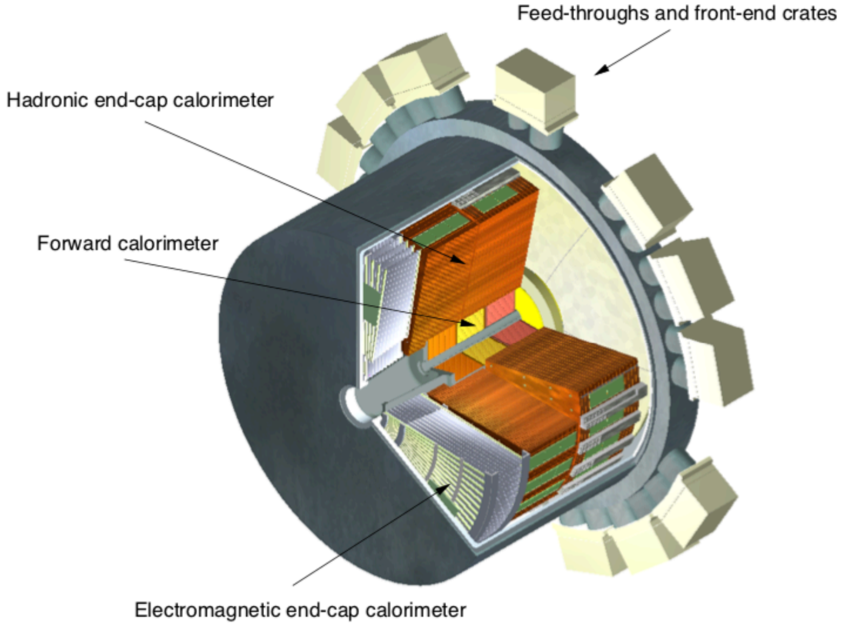
- The **EMEC (ElectroMagnetic End-Cap) [7]** is a lead-liquid argon sampling calorimeter with interleaved accordion-shaped absorbers and electrodes.

- The geometry has been chosen of accordion type because:
  - it allows very good hermeticity
  - increases the absorber rigidity which improve their relative positioning and hence contributes to the response uniformity.
  - minimizes inductances in the signal paths



Picture of an electromagnetic end-cap module during stacking, showing the accordion structure of the ATLAS EM calorimeters.

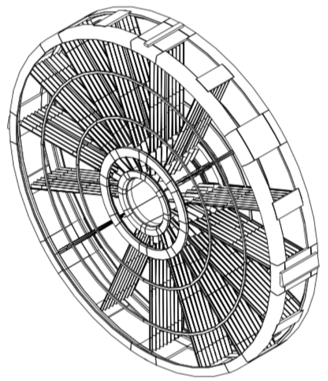
- It has been implemented as a **custom solid** ***(GeoSpecialShape)*** in GeoModel and in Geant4



Cut-away view of an end-cap cryostat showing the positions of the three end-cap calorimeters. The outer radius of the cylindrical cryostat vessel is 2.25 m and the length of the cryostat is 3.17 m.

Complexity of ATLAS geometry: special shape

The **ATLAS EMEC** consists of a fan-like structures positioned radially like a wheel spokes and serving as absorbers and electrodes.

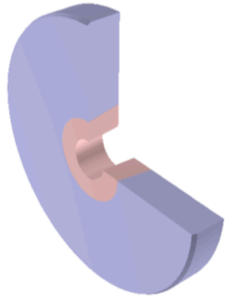


- Custom implementation of the **EMEC (GeoSpecialShapes)** calculations take ~15% simulation time

```
atlas_subdir( GeoSpecialShapes )
atlas_depends_on_subdirs( PUBLIC
                          Control/SGTools
                          Control/StoreGate
                          PRIVATE
                          Control/AthenaKernel
                          Control/CxxUtils
                          Database/RDBAccessSvc
                          DetectorDescription/GeoModel/
GeoModelInterfaces
                          DetectorDescription/GeoModel/
GeoModelUtilities
                          GaudiKernel )
```

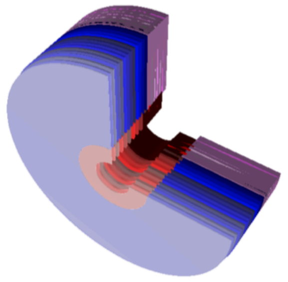
`LArWheelSolid`

- Bounding Shape is `G4Polycone`
- Implies the coordinate system with $Z = 0$ at front face — in use everywhere



`LArWheelSliceSolid`

- A slice per half-wave
- Each slice has an unique number id
- Navigation from slice to slice at Geant4 level



Recent developments from A. Sukharev [8]

FullSimLight: an ATLAS Full Simulation prototype

FullSimLight: an ATLAS Full Simulation prototype

Detector construction

The detector is
built from
the persistified
ATLAS geometry

FullSimLight: an ATLAS Full Simulation prototype

Detector construction

The detector is built from the persistified ATLAS geometry

A constant magnetic field can be set through a macro

Magnetic Field

FullSimLight: an ATLAS Full Simulation prototype

Detector construction

The detector is built from the persistified ATLAS geometry

Magnetic Field

A constant magnetic field can be set through a macro

Primaries Generator

Primary generators is a particle gun that gen. particles at position (0,0,0) with options*

FullSimLight: an ATLAS Full Simulation prototype

Detector construction

The detector is built from the persistified ATLAS geometry

Magnetic Field

A constant magnetic field can be set through a macro

Primaries Generator

Primary generators is a particle gun that gen. particles at position (0,0,0) with options*

*# Number of primaries per event (default [1, 10])
 # Primary particle energy (default [1 GeV, 100 GeV])
 # Primary particle direction (default isotropic distribution)
 # Primary particle type (currently e-, e+ and gamma)

FullSimLight: an ATLAS Full Simulation prototype

Detector construction

The detector is built from the persistified ATLAS geometry

Magnetic Field

A constant magnetic field can be set through a macro

Primaries Generator

Primary generators is a particle gun that gen. particles at position (0,0,0) with options*

Scoring

A set of observables** is collected during the simulation per primary-particle

*# Number of primaries per event (default [1, 10])
 # Primary particle energy (default [1 GeV, 100 GeV])
 # Primary particle direction (default isotropic distribution)
 # Primary particle type (currently e-, e+ and gamma)

FullSimLight: an ATLAS Full Simulation prototype

ATLAS EXPERIMENT

CERN

Detector construction

The detector is built from the persistified ATLAS geometry

Magnetic Field

A constant magnetic field can be set through a macro

Primaries Generator

Primary generators is a particle gun that gen. particles at position (0,0,0) with options*

Scoring

A set of observables** is collected during the simulation per primary-particle

*# Number of primaries per event (default [1, 10])
Primary particle energy (default [1 GeV, 100 GeV])
Primary particle direction (default isotropic distribution)
Primary particle type (currently e-, e+ and gamma)

**Mean energy deposit
Mean charged and neutral step lengths
Mean number of steps made by charged and neutral particles
Mean number of secondary e-, e+ and gamma particles

FullSimLight: an ATLAS Full Simulation prototype

Detector construction

The detector is built from the persistified ATLAS geometry

Magnetic Field

A constant magnetic field can be set through a macro

Primaries Generator

Primary generators is a particle gun that gen. particles at position (0,0,0) with options*

Scoring

A set of observables** is collected during the simulation per primary-particle

Physics List

The Physics List can be specified as an input argument with a flag(e.g.FTFP_BERT)

*# Number of primaries per event (default [1, 10])
 # Primary particle energy (default [1 GeV, 100 GeV])
 # Primary particle direction (default isotropic distribution)
 # Primary particle type (currently e-, e+ and gamma)

**Mean energy deposit
 Mean charged and neutral step lengths
 Mean number of steps made by charged and neutral particles
 Mean number of secondary e-, e+ and gamma particles

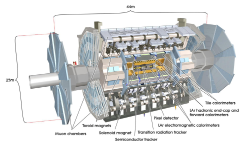
FullSimLight: an ATLAS Full Simulation prototype

- **FullSimLight** is a Geant4 based simulation of the full ATLAS detector.
- The application can run both in **sequential** and **multithreaded** mode, with a flag in the macro (MT version requires Geant4 to be compiled with the flag -`DGEANT4_BUILD_MULTITHREADED=ON`)
- The **detector construction** supports also **GDML** geometry files (in this case Geant4 has to be compiled with GDML support with `–DGEANT4_USE_GDML=ON`) - the ATLAS .gdml file includes the whole ATLAS geometry except the special shape
- Some **observables** are collected during the event and the result is reported at the end of each event for each primary particle that was transported
- The **final report shows**:
  - the run time
  - the primary generator and magnetic field settings
  - the total number of events and primary particles transported
  - per-primary type simulation statistics

FullSimLight: an ATLAS Full Simulation prototype

FullSimLight: an ATLAS Full Simulation prototype

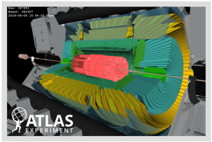
FullSimLight: an ATLAS Full Simulation prototype



Access the persistified ATLAS geometry

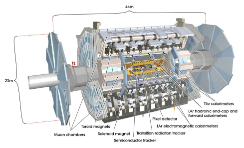
Standalone GeoModel

FullSimLight: an ATLAS Full Simulation prototype



Convert GeoModel geometry to Geant4

GeoModel2G4



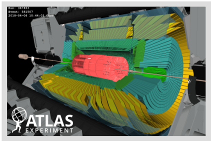
Access the persistified ATLAS geometry

Standalone GeoModel

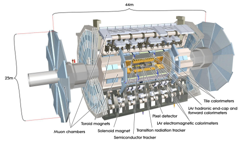
FullSimLight: an ATLAS Full Simulation prototype

GEANT4 A SIMULATION TOOLKIT

Geant4 application



Convert GeoModel geometry to Geant4



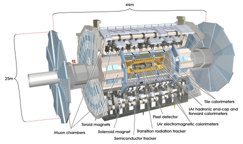
Access the persistified ATLAS geometry

Standalone GeoModel

GeoModel2G4

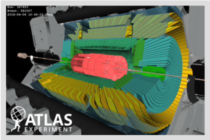
Geant4

FullSimLight: an ATLAS Full Simulation prototype



Access the persistified ATLAS geometry

Standalone GeoModel

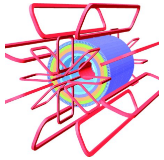


Convert GeoModel geometry to Geant4

GeoModel2G4

Geant4 application

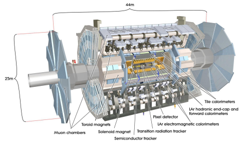
Geant4



Transport in real field

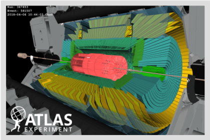
ATLAS magnetic field

FullSimLight: an ATLAS Full Simulation prototype



Access the persistified ATLAS geometry

Standalone GeoModel



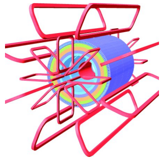
Convert GeoModel geometry to Geant4

GeoModel2G4

GEANT4
A SIMULATION TOOLKIT

Geant4 application

Geant4



Transport in real field

ATLAS magnetic field

Alignments?
Input from Ev Generators?

?

~~*Hits production*~~

FullSimLight as geometry debugging tool

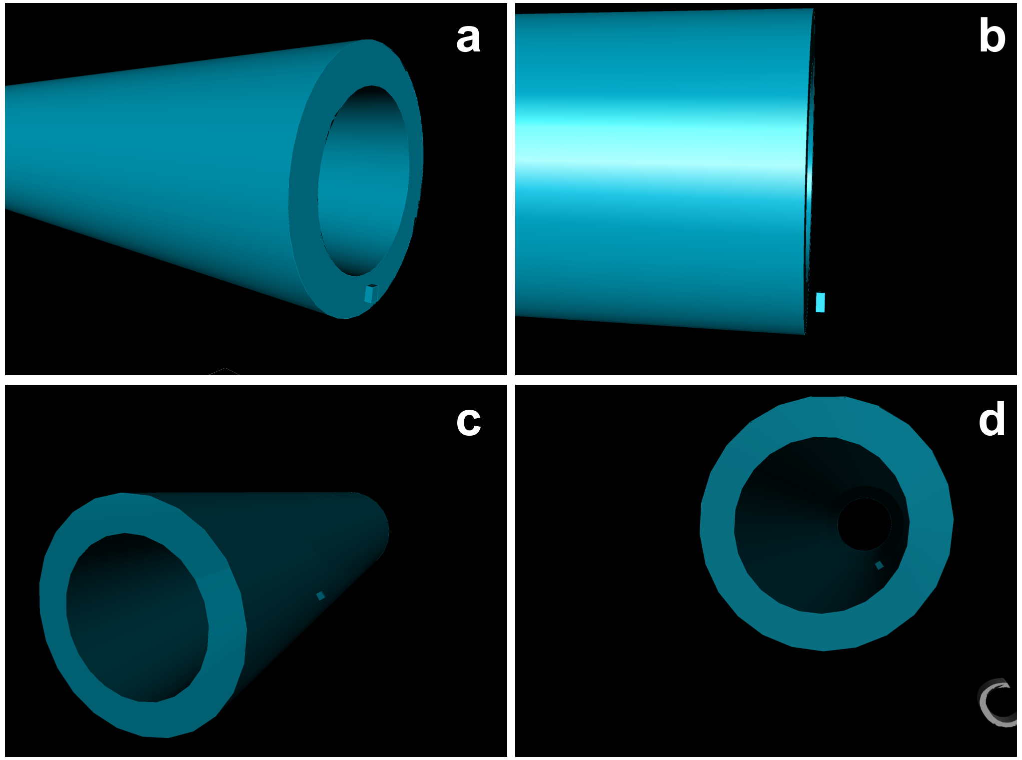
FullSimLight can be a useful tool for many use cases:

- **Geometry debugging** purposes:
  - When we first plugged-in the dump of the persistified ATLAS geometry, we discovered some bugs (clashing of geometries or misplacement of geometries)
- Another idea is to study the possibility to use **parallel geometries** [9]:
  - i.e. event biasing, scoring of radiation, and/or the creation of hits in detailed readout structures studies

```
-------- G4Exception-START ---------
*** G4Exception : GeomMgt0002
      issued by :
G4SmartVoxelHeader::BuildNodes()
PANIC! - Overlapping daughter with mother
volume.
         Daughter physical volume
Brl1570_TrolleyA
         is entirely outside mother
logical volume Env21A_PP0 !!
*** Fatal Exception *** core dump ***
 **** Track information is not available
at this moment
 **** Step information is not available at
this moment

-------- G4Exception-END -------


*** G4Exception: Aborting execution ***
Abort trap: 6
```



Volume rendering images realised with VP1Light [10], showing the daughter volume outside the mother volume (a-b) and mother and daughter volumes after the bug was fixed (c-d).

Summary

- We have a first working prototype of **FullSimLight**: it decouples MonteCarlo simulation from the **ATLAS** experiment infrastructure

- The relevant features have been extracted from the Athena framework to produce a **first standalone simulation application**

- This lightweight prototype is meant to ease **debugging operations** on the Geant4 side and to allow early testing of new Geant4 releases:
  - not only : It has been already useful as geometry debugging tool

- **Simplify studies of Geant4 tracking and physics processes**, including on novel architectures

- It will also ease **optimization studies and R&D activities related to HPC development**: i.e. the possibility to offload partially/totally the simulation to GPUs/Accelerators without having to port the whole experimental infrastructure.

Thanks for your attention.

Marilena Bandieramonte
marilena.bandieramonte@cern.ch

Backup slides