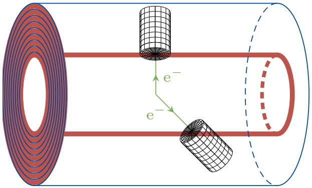
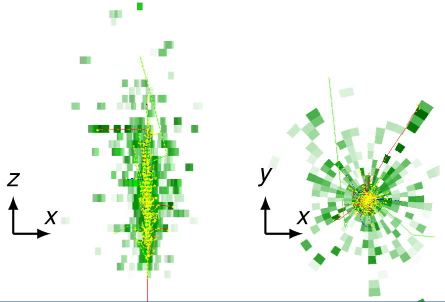
Updates on Par04 and fast simulation

Anna Zaborowska
Dalila Salamani

Geant4 Collaboration Week, 25.09.2023

Geant4 example Par04

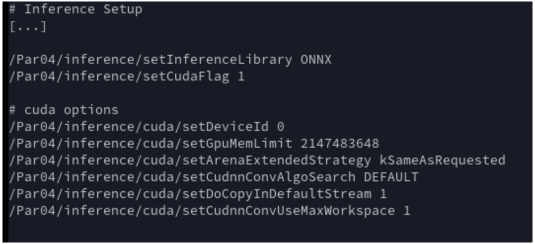




- Extended example Par04 shows how to use Machine Learning (ML) models within GEANT4.
- Distributed with a Variational Autoencder (VAE) model of showers used in fast simulation, updated to a recent version.
- Demonstrates how to incorporate inference libraries: ONNX runtime, pyTorch (new since last release), lwtnn.

Geant4 example Par04: GPU support

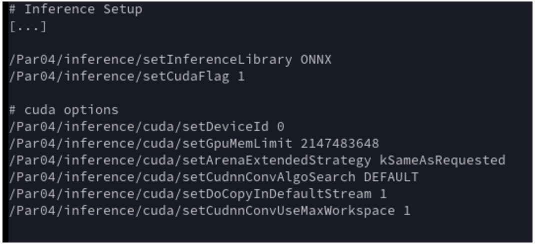
- In the coming release will allow to run inference with ONNX runtime on a GPU (a choice done by UI command).
- No extensive tests performed so far, but with a single-shower inference this may not show performance gain.



```
# Inference Setup
[...]

/Par04/inference/setInferenceLibrary ONNX
/Par04/inference/setCudaFlag 1

# cuda options
/Par04/inference/cuda/setDeviceId 0
/Par04/inference/cuda/setGpuMemLimit 2147483648
/Par04/inference/cuda/setArenaExtendedStrategy kSameAsRequested
/Par04/inference/cuda/setCudnnConvAlgoSearch DEFAULT
/Par04/inference/cuda/setDoCopyInDefaultStream 1
/Par04/inference/cuda/setCudnnConvUseMaxWorkspace 1
```

Geant4 example Par04: GPU support

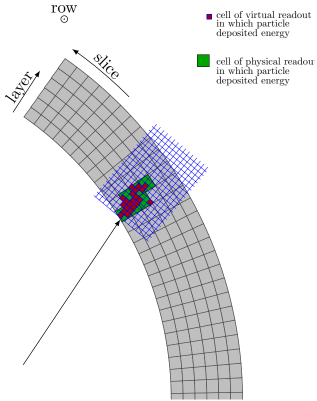
- In the coming release will allow to run inference with ONNX runtime on a GPU (a choice done by UI command).
- No extensive tests performed so far, but with a single-shower inference this may not show performance gain.
- Possibility to use `G4VFastSimulationModel::Flush()` to explore batching (although not expecting more than $\mathcal{O}(10)$).



```
# Inference Setup
[...]

/Par04/inference/setInferenceLibrary ONNX
/Par04/inference/setCudaFlag 1

# cuda options
/Par04/inference/cuda/setDeviceId 0
/Par04/inference/cuda/setGpuMemLimit 2147483648
/Par04/inference/cuda/setArenaExtendedStrategy kSameAsRequested
/Par04/inference/cuda/setCudnnConvAlgoSearch DEFAULT
/Par04/inference/cuda/setDoCopyInDefaultStream 1
/Par04/inference/cuda/setCudnnConvUseMaxWorkspace 1
```

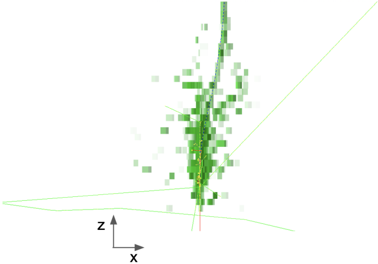
Geant4 example Par04: realistic readout

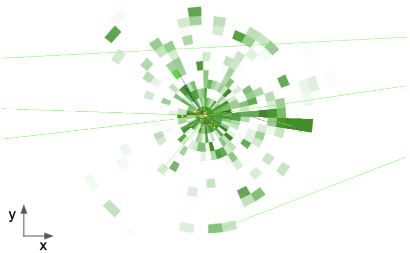
- Cylindrical scoring mesh for energy deposits is key in obtaining same showers independently of position in the detector.
- Recent additions of physical detector readout for performance benchmarking (for ML model speed-up calculation, to account the overhead time on top of inference time)
- Serves also an example of how to place back fast-simulation hits (even if they fall inside absorbers) via parallel world geometry.


row
layer
slice
🟥 cell of virtual readout in which particle deposited energy
🟩 cell of physical readou in which particle deposited energy

Geant4 example Par04: datasets

- Full simulation source of datasets:
  - on zenodo: High Granularity Electromagnetic Calorimeter Shower Images
    DOI 10.5281/zenodo.6082201
  - used as dataset 2 and 3 in CaloChallenge:
    - dataset 2: 6’480 voxels DOI 10.5281/zenodo.6366271
    - dataset 3: 40’500 voxels DOI 10.5281/zenodo.6366324





Calo Challenge calochallenge.github.io

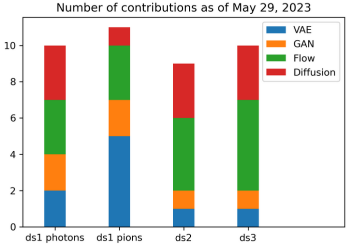
Michele Faucci Giannelli, Gregor Kasieczka, Claudius Krause, Ben Nachman, Dalila Salamani, David Shih and Anna Zaborowska

- Lots of on-going effort with ML models for fast shower simulation, but with **no place for comparison or discussions** of different methods.
- This challenge was released with three datasets with increasing dimension of input: first one comes from ATLAS open data, **the 2nd and 3rd from Par04**.
- Set of **common datasets** and **common validation metrics** allows to productively discuss different approaches.

Calo Challenge calochallenge.github.io

Michele Faucci Giannelli, Gregor Kasieczka, Claudius Krause, Ben Nachman, Dalila Salamani, David Shih and Anna Zaborowska

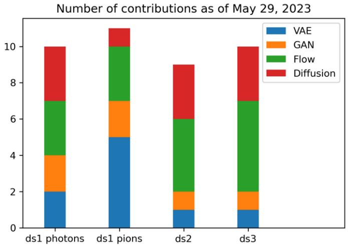
- Lots of on-going effort with ML models for fast shower simulation, but with **no place for comparison or discussions** of different methods.
- This challenge was released with three datasets with increasing dimension of input: first one comes from ATLAS open data, **the 2nd and 3rd from Par04**.
- Set of **common datasets** and **common validation metrics** allows to productively discuss different approaches.



Calo Challenge calochallenge.github.io

Michele Faucci Giannelli, Gregor Kasieczka, Claudius Krause, Ben Nachman, Dalila Salamani, David Shih and Anna Zaborowska

- Lots of on-going effort with ML models for fast shower simulation, but with **no place for comparison or discussions** of different methods.
- This challenge was released with three datasets with increasing dimension of input: first one comes from ATLAS open data, **the 2nd and 3rd from Par04**.
- Set of **common datasets** and **common validation metrics** allows to productively discuss different approaches.

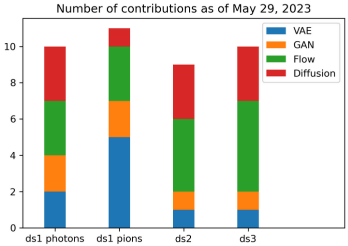


- No winner is expected, it's a compilation of alternative solutions, often with different architectures.

Calo Challenge calochallenge.github.io

Michele Faucci Giannelli, Gregor Kasieczka, Claudius Krause, Ben Nachman, Dalila Salamani, David Shih and Anna Zaborowska

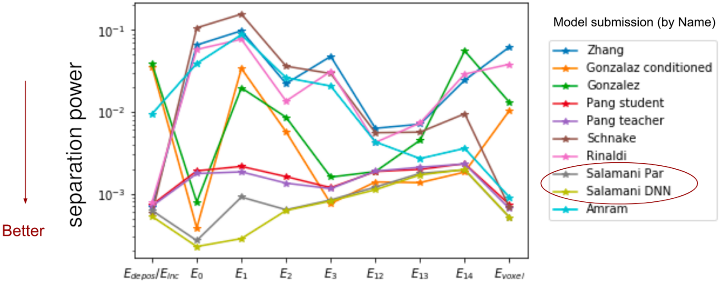
- Lots of on-going effort with ML models for fast shower simulation, but with **no place for comparison or discussions** of different methods.
- This challenge was released with three datasets with increasing dimension of input: first one comes from ATLAS open data, **the 2nd and 3rd from Par04**.
- Set of **common datasets** and **common validation metrics** allows to productively discuss different approaches.



- No winner is expected, it's a compilation of alternative solutions, often with different architectures.
- Calo Challenge Workshop held on 30-31 May 2023 in Frascati gathered many contributors and was a place for fruitful discussions, and first benchmarks.
- Final benchmarking and paper write-up is in progress.

Calo Challenge: VAE model

Variational autoencoder (VAE) model developed for ATLAS pion dataset by Dalila Salamani.
It seems to be very attractive, both comparing shower observables (histograms),



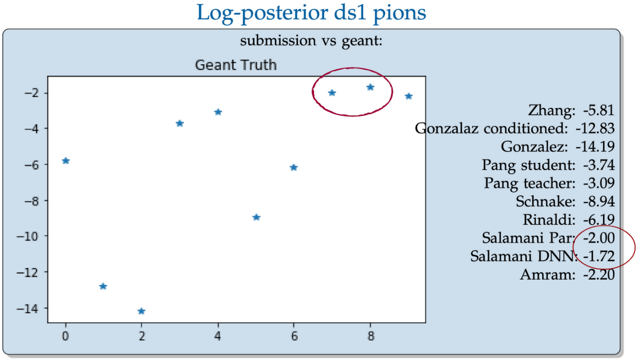
Calo Challenge: VAE model

Variational autoencoder (VAE) model developed for ATLAS pion dataset by Dalila Salamani.

It seems to be very attractive, both comparing shower observables (histograms), as well as the binary classifier (single cell comparison).

Currently being implemented within ATLAS.

Better



Log-posterior ds1 pions

submission vs geant:

Geant Truth

Zhang:	-5.81
Gonzalaz conditioned:	-12.83
Gonzalez:	-14.19
Pang student:	-3.74
Pang teacher:	-3.09
Schnake:	-8.94
Rinaldi:	-6.19
Salamani Par:	-2.00
Salamani DNN:	-1.72
Amram:	-2.20

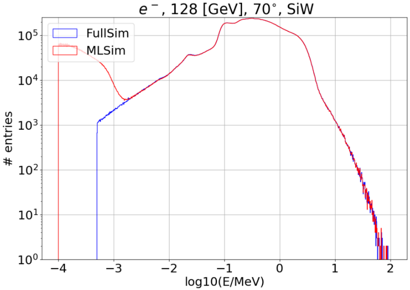
Calo Challenge: transformer-based model

Also developed in EP-SFT by Piyush Raikwar, in cooperation with CERN openlab and IBM
Transformer architecture is powerful, proposed in 2017 (10.48550/arXiv.1706.03762) and gained lots of attention recently (chatGPT, DALL-E, ...)

Calo Challenge: transformer-based model

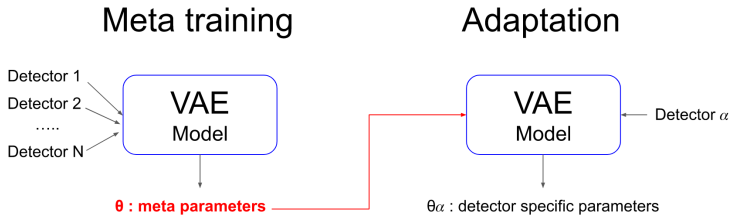
Also developed in EP-SFT by Piyush Raikwar, in cooperation with CERN openlab and IBM
Transformer architecture is powerful, proposed in 2017 (10.48550/arXiv.1706.03762) and gained lots of attention recently (chatGPT, DALL-E, ...)

- The main objective is to better model the cell energy distribution
- Final goal is to train a model on showers from various detectors to allow it to learn representations, with quick adaptation phase to each experiment



not yet a final result form a generative model, a interim prototype

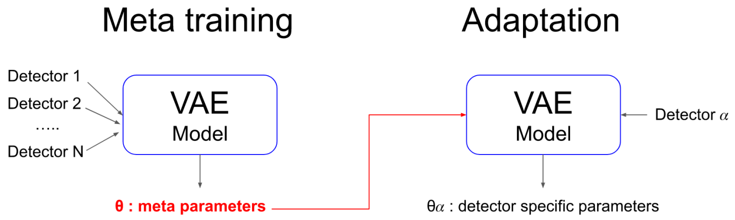
MetaHEP



Finalised publication *MetaHEP: Meta learning for fast shower simulation of high energy physics experiments* 10.1016/j.physletb.2023.138079.

- Covered at the last Geant4 Collaboration Week by Dalila

MetaHEP



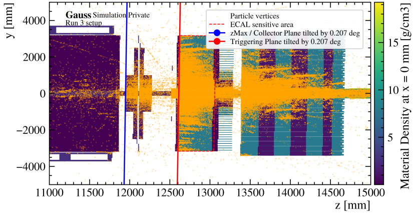
Finalised publication *MetaHEP: Meta learning for fast shower simulation of high energy physics experiments* 10.1016/j.physletb.2023.138079.

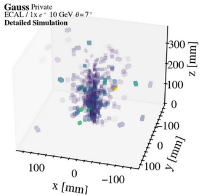
- Covered at the last Geant4 Collaboration Week by Dalila
- Extensive instruction on how to use MetaHEP for any detector described at our website: g4fastsim.web.cern.ch/
- Documentation based on Par04 and the first experiment-framework implementation: in key4HEP simulation toolkit k4SimGeant4
- Inference part of Par04 was also integrated into DD4hep by DESY group (Frank Gaede et al) as DDML

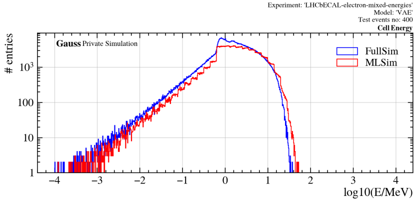
Implementation within Gaussino for the LHCb

work done by Michal Mazurek

- Implementation of Par04 approach in Gaussino, use on LHCb's calorimetry;
- Many cavaets discovered on the way, as expected in realistic conditions;
- Implementation ready to be used, first model (VAE) is being tested;
- Very general implementation, allows to use MetaHEP, or any models submitted to CaloChallange;



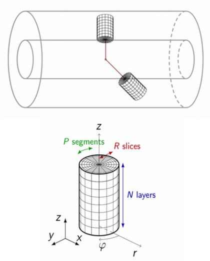




Preliminary results by Michal

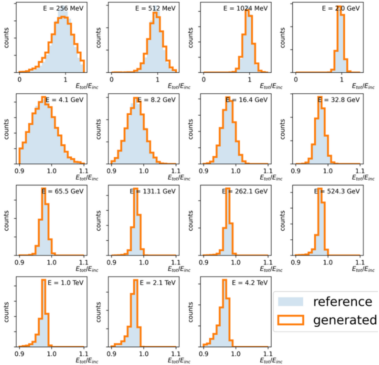
First look on MetaHEP for ATLAS

MetaHEP : Adaptation



$R_xP_xN=18_x50_x45$

Artificially placing the ATLAS voxels into R_xP_xN voxels and padding with 0



CaloChallenge ATLAS Dataset 1 photons

Naturally, a proper “cylinder” should come directly from simulation, but looks promising!

Classical parameterisation

- Most fast-sim R&D is ML-driven;
- But ML still not widely used in production;

Classical parameterisation

ATLAS simulation

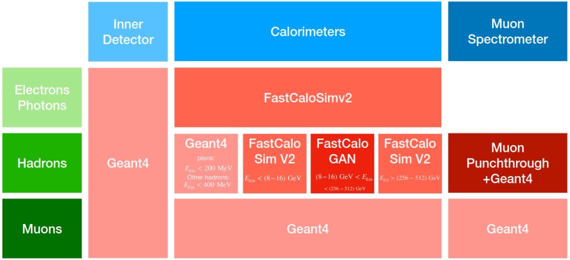


Figure 17: The configuration of the different tools used for AtlFast3, which depends on the particle type, the detector and the particle energy.

10.1007/s41781-021-00079-7

- Most fast-sim R&D is ML-driven;
- But ML still not widely used in production;

FastCaloSimV2-based classical shower simulation

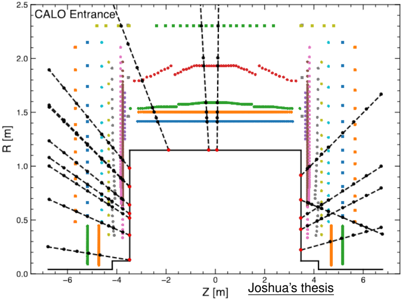
work done by Joshua Falco Beirer

- ATLAS moves to Geant4 fast-sim hooks;
- Extraction of classical parameterisation to a package only Geant4-dependant;

FastCaloSimV2-based classical shower simulation

work done by Joshua Falco Beirer

- ATLAS moves to Geant4 fast-sim hooks;
- Extraction of classical parameterisation to a package only Geant4-dependant;
- Getting it ATLAS independent starting with extrapolation to calo layers;

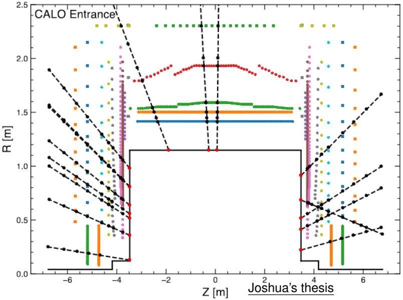


Joshua's thesis

FastCaloSimV2-based classical shower simulation

work done by Joshua Falco Beirer

- ATLAS moves to Geant4 fast-sim hooks;
- Extraction of classical parameterisation to a package only Geant4-dependant;
- Getting it ATLAS independent starting with extrapolation to calo layers;
- Next moving to extraction of parameterisation to be ATLAS geometry independent;

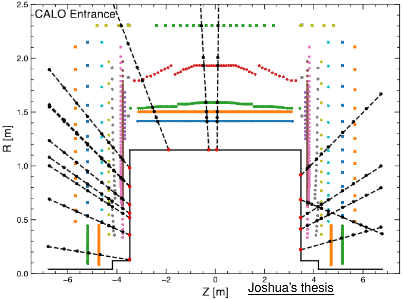


Joshua's thesis

FastCaloSimV2-based classical shower simulation

work done by Joshua Falco Beirer

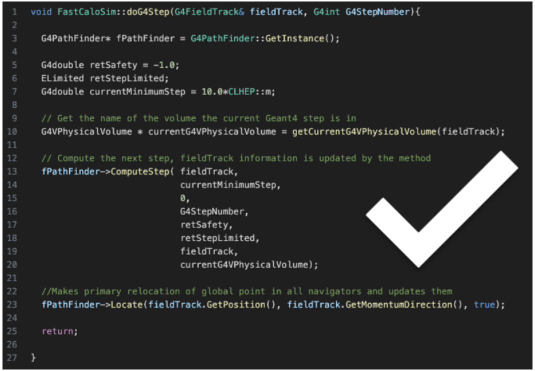
- ATLAS moves to Geant4 fast-sim hooks;
- Extraction of classical parameterisation to a package only Geant4-dependant;
- Getting it ATLAS independent starting with extrapolation to calo layers;
- Next moving to extraction of parameterisation to be ATLAS geometry independent;



Joshua's thesis

- First tests to validate against on ATLAS geometry, then move towards some future detector (or Open Data Detector)

Track extrapolation

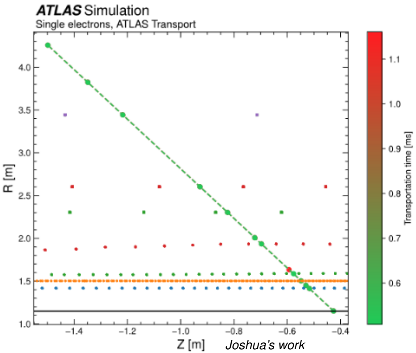


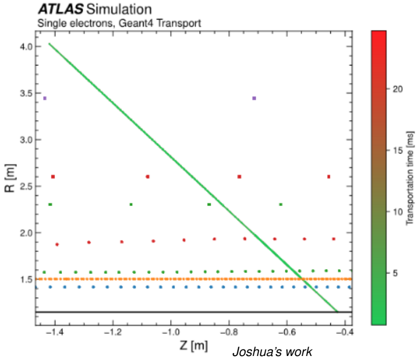
```
void FastCaloSim::doG4Step(G4FieldTrack& fieldTrack, G4int G4StepNumber){

  G4PathFinder* fPathFinder = G4PathFinder::GetInstance();

  G4double retSafety = -1.0;
  ELimited retStepLimited;
  G4double currentMinimumStep = 10.0*CLHEP::m;

  // Get the name of the volume the current Geant4 step is in
  G4VPhysicalVolume * currentG4VPhysicalVolume = getCurrentG4VPhysicalVolume(fieldTrack);

  // Compute the next step, fieldTrack information is updated by the method
  fPathFinder->ComputeStep( fieldTrack,
                            currentMinimumStep,
                            0,
                            G4StepNumber,
                            retSafety,
                            retStepLimited,
                            fieldTrack,
                            currentG4VPhysicalVolume);

  //Makes primary relocation of global point in all navigators and updates them
  fPathFinder->Locate(fieldTrack.GetPosition(), fieldTrack.GetMomentumDirection(), true);

  return;

}
```

Joshua's work

- Track extrapolation using `G4PathFinder` (inspired by Par02 example).
- The most primitive method to try (to assess accuracy wrt ATLAS tracking): loop over `ComputeStep` method with `G4FieldTrack` as input to each iteration, encoding information of current state of particle
- Invoking `Locate` method of `G4PathFinder` to update position
- First naive visual validation indicates identical results to transportation with ATLAS tracking tools

Track extrapolation: results as expected

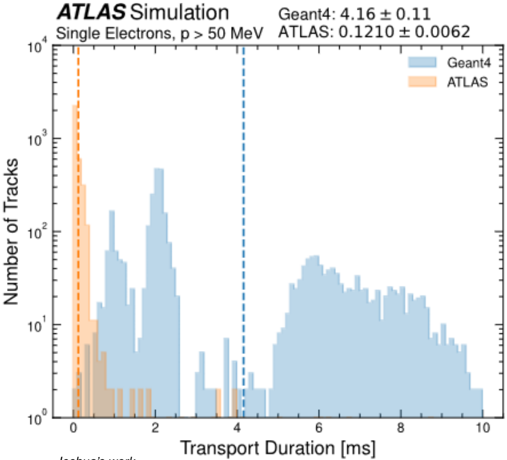




- ATLAS Transport: creates points only for $\mathcal{O}(12)$ intersections with calorimeter layers: **fast* and good approximation**
- GEANT4 Transport: due to many boundaries extrapolation results in many points along the particle trajectory: **slow and very precise (to an extend that we don't need)**

Track extrapolation: first naive approach

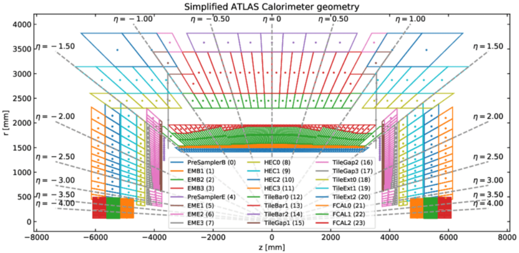
First approach confirms end-goal results, but as expected it's unnecessarily slow.



Joshua's work

Track extrapolation: current approach

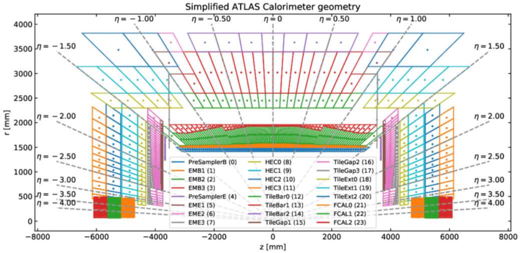
- Clearly the volume boundaries are our limit;
- Use of simplifed geometry is a natural direction;



ATLAS, 10.1051/epjconf/202024510003

Track extrapolation: current approach

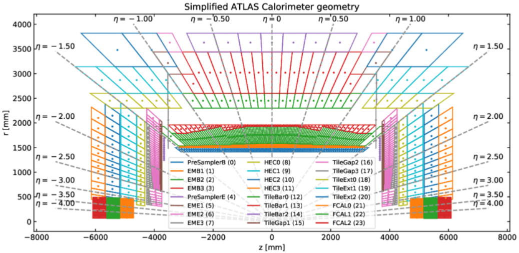
- Clearly the volume boundaries are our limit;
- Use of simplifed geometry is a natural direction;
- Registering parallel world not exactly a great solution in the existing production environment;



ATLAS, 10.1051/epjconf/202024510003

Track extrapolation: current approach

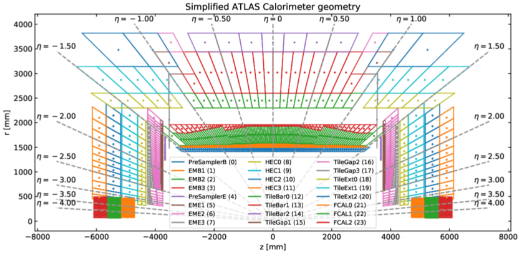
- Clearly the volume boundaries are our limit;
- Use of simplifed geometry is a natural direction;
- Registering parallel world not exactly a great solution in the existing production environment;
- But: parallel world read from GDML just for the sake of our model, unregistered? (Thanks John Ap for this hint!)



ATLAS, 10.1051/epjconf/202024510003

Track extrapolation: current approach

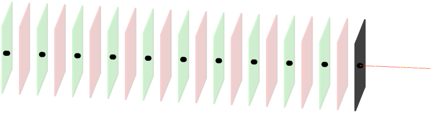
- Clearly the volume boundaries are our limit;
- Use of simplifed geometry is a natural direction;
- Registering parallel world not exactly a great solution in the existing production environment;
- But: parallel world read from GDML just for the sake of our model, unregistered? (Thanks John Ap for this hint!)
- Goal for ATLAS example is to replicate their reco geometry;



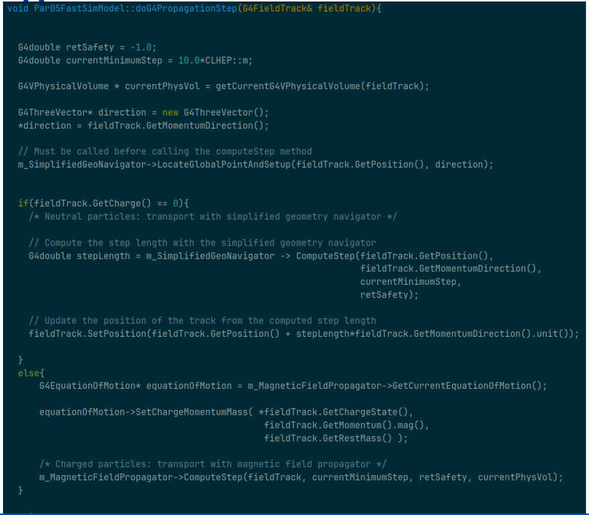
ATLAS, 10.1051/epjconf/202024510003

Track extrapolation: Par05 prototype

- We prototype in a standalone G4 aplication called Par05 as possibly could be a useful example of a simplified geometry (let's see how it evolves);



red = mass world, no points are created here

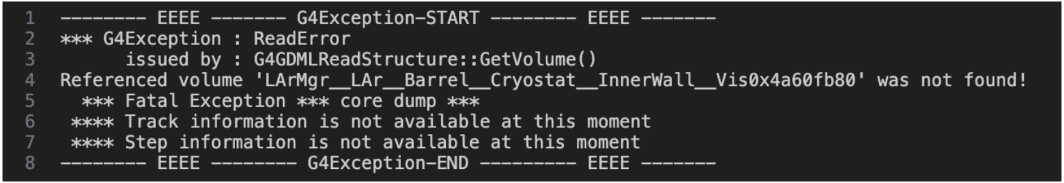


```
void Par05FastSimModel::doG4PropagationStep(G4FieldTrack& fieldTrack){


  G4double retSafety = -1.0;
  G4double currentMinimumStep = 10.0*CLHEP::m;

  G4VPhysicalVolume * currentPhysVol = getCurrentG4VPhysicalVolume(fieldTrack);

  G4ThreeVector* direction = new G4ThreeVector();
  *direction = fieldTrack.GetMomentumDirection();

  // Must be called before calling the computeStep method
  m_SimplifiedGeoNavigator->LocateGlobalPointAndSetup(fieldTrack.GetPosition(), direction);


  if(fieldTrack.GetCharge() == 0){
    /* Neutral particles: transport with simplified geometry navigator */

    // Compute the step length with the simplified geometry navigator
    G4double stepLength = m_SimplifiedGeoNavigator -> ComputeStep(fieldTrack.GetPosition(),
                                                                fieldTrack.GetMomentumDirection(),
                                                                currentMinimumStep,
                                                                retSafety);

    // Update the position of the track from the computed step length
    fieldTrack.SetPosition(fieldTrack.GetPosition() + stepLength*fieldTrack.GetMomentumDirection().unit());

  }
  else{
      G4EquationOfMotion* equationOfMotion = m_MagneticFieldPropagator->GetCurrentEquationOfMotion();

      equationOfMotion->SetChargeMomentumMass( *fieldTrack.GetChargeState(),
                                                fieldTrack.GetMomentum().mag(),
                                                fieldTrack.GetRestMass() );

      /* Charged particles: transport with magnetic field propagator */
      m_MagneticFieldPropagator->ComputeStep(fieldTrack, currentMinimumStep, retSafety, currentPhysVol);
  }
```

Track extrapolation: back to Athena

Unfortunately the same recipe applied in ATLAS framework: Athena, does not work...

Something seems to make `G4GDMLParser` unable to read in gdml files.



```
-------- EEEE ------- G4Exception-START -------- EEEE -------
*** G4Exception : ReadError
      issued by : G4GDMLReadStructure::GetVolume()
Referenced volume 'LArMgr__LAr__Barrel__Cryostat__InnerWall__Vis0x4a60fb80' was not found!
  *** Fatal Exception *** core dump ***
 **** Track information is not available at this moment
 **** Step information is not available at this moment
-------- EEEE -------- G4Exception-END --------- EEEE -------
```

Summary and Outlook

On-going work on finalisation of many of the topics:

- Par04 changes must be moved from our development repo to geant-dev;
- Many topics on the ML models: Calo Challenge finalisation, next evolution, ...
- Exciting results from LHCb who tests our (and in future Calo Challenge community!) models;

Summary and Outlook

On-going work on finalisation of many of the topics:

- Par04 changes must be moved from our development repo to geant-dev;
- Many topics on the ML models: Calo Challenge finalisation, next evolution, ...
- Exciting results from LHCb who tests our (and in future Calo Challenge community!) models;

More long-term plans:

- Implementation of MetaHEP for other experiments than LHCb;
- Development of ML models based on transformers, applicable within above-mentioned implementations to frameworks;
- Helping to decouple the ATLAS classical parameterisation to act as a standalone package;