FASER2 :

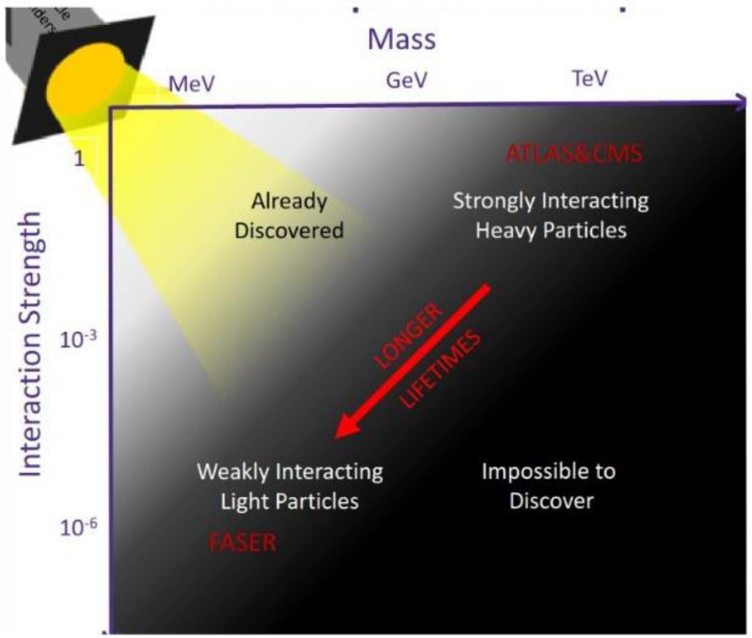
Development of a tracking software with ACTS and detector design studies

Olivier SALIN

UNIVERSITY OF OXFORD

Forward physics motivation

pp collisions at the LHC produce many weakly coupled particles in the forward direction
Currently being missed by colliders detectors at the LHC

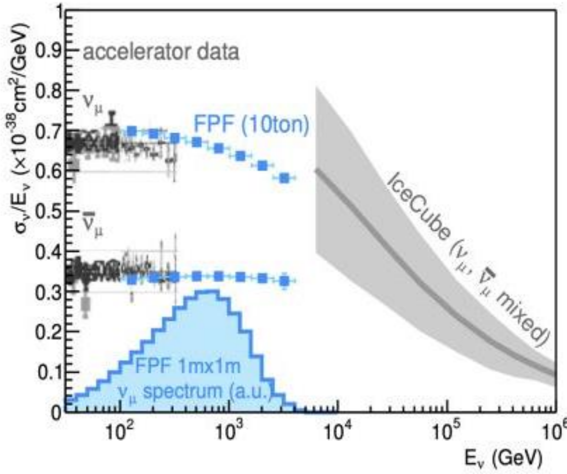


BSM Feebly Interacting Particles :

- Long-lived Particles (LLP)
- Axion-Like-Particles (ALPs)
- Heavy Neutral Lepton (HNL)

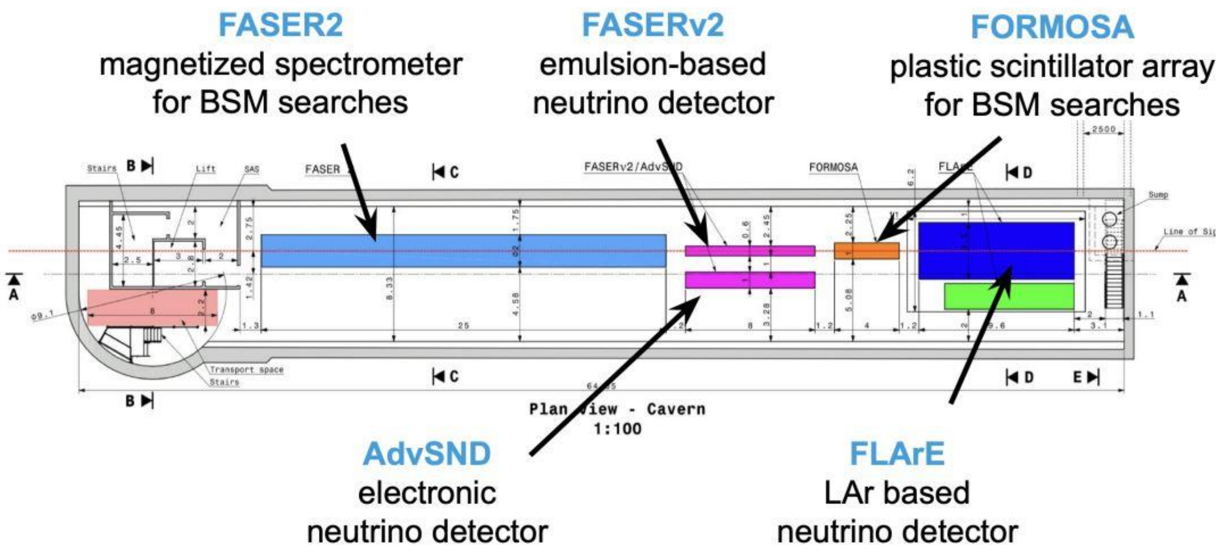
Collider neutrino:

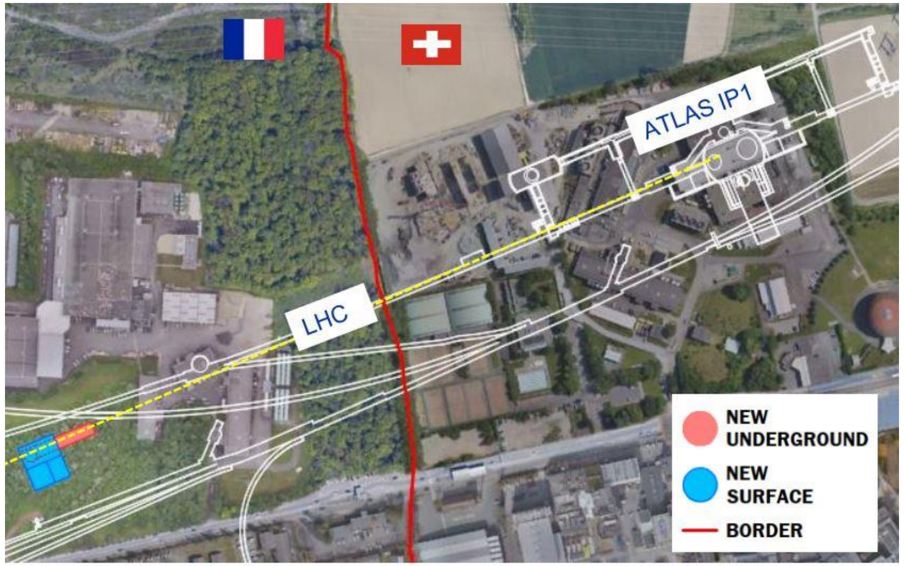
- Collider neutrinos are produced from the weak decays of mesons
- Highest energy neutrinos produced in forward direction
- Complementary to existing neutrino experiments



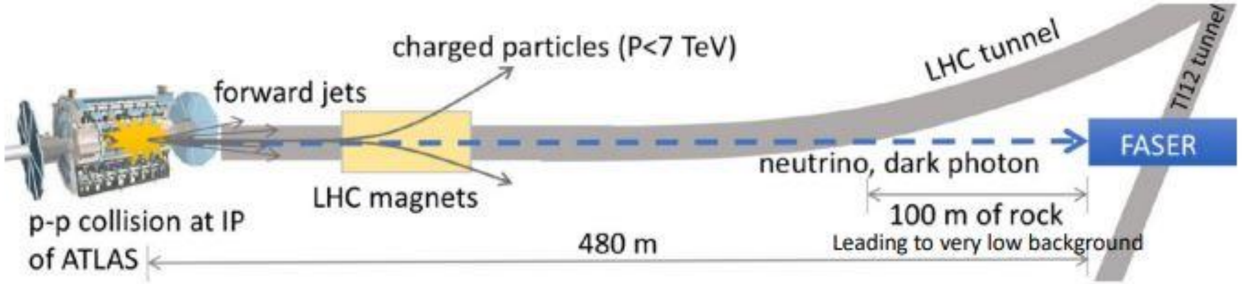
Forward Physics Facility (FPF)

- Forward Physics facility proposed cavern to host many forward experiments at the HL-LHC
- The cavern is 65 m-long, 9 m-wide/high located 620m away from ATLAS IP



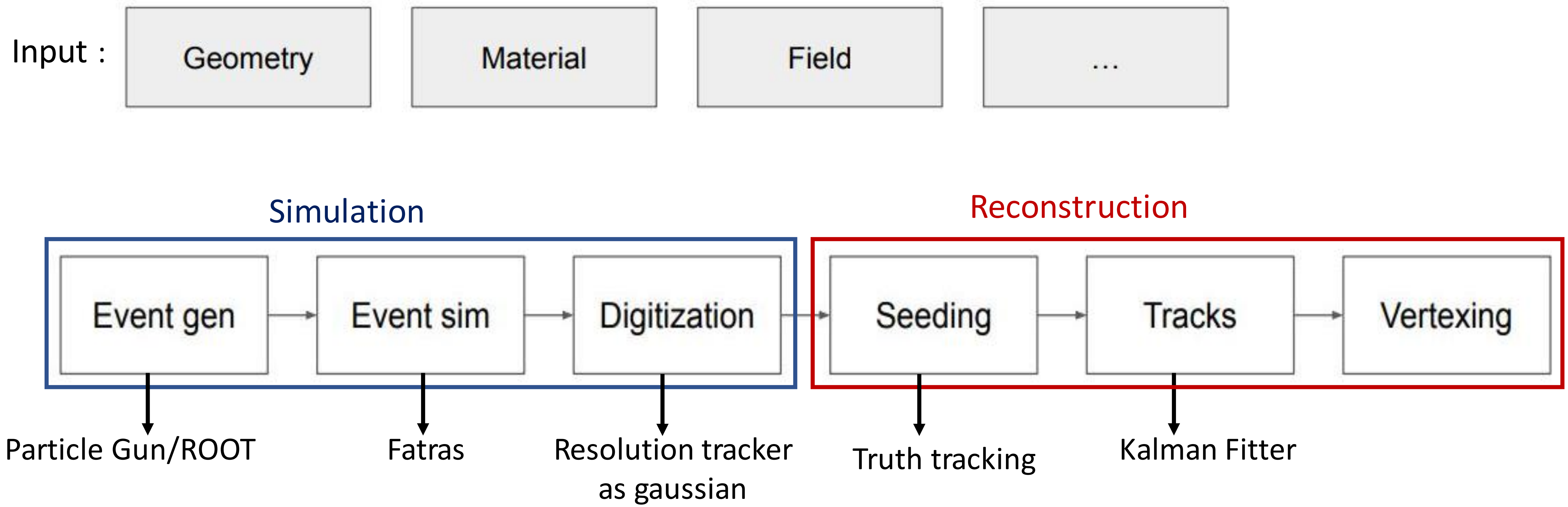


Physic process:



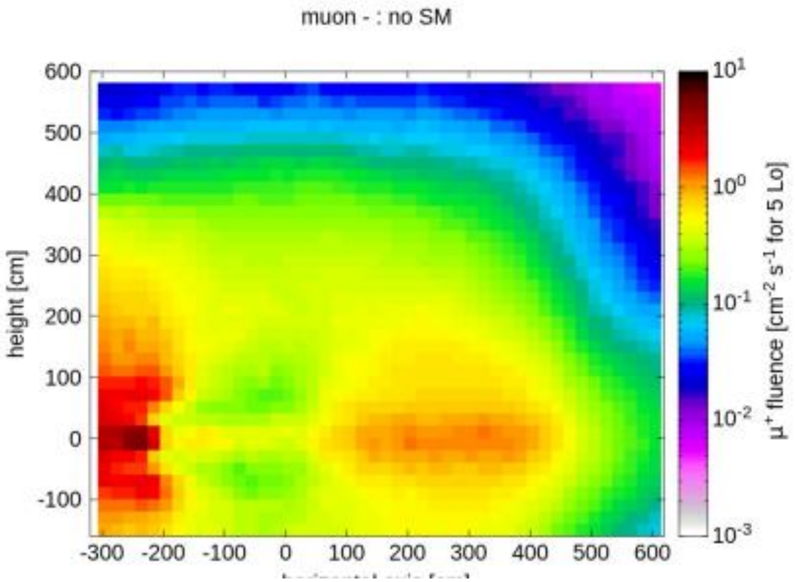
ACTS track reconstruction algorithm

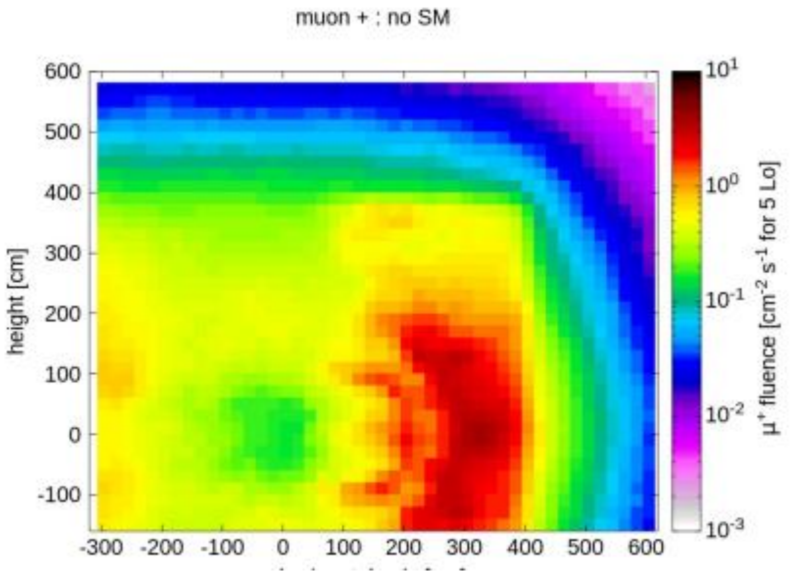
Python binding function with the simulation and reconstruction algorithm: [here]()



FASER2 : Muon Background studies

- Muons from ATLAS IP are the main background for most FPF signals and will define trigger rates
- FLUKA results for the HL-LHC are 0.6 Hz/cm^2 at LOS, much larger at some locations ~1 m from the LOS
  - Easy to veto for most FPF signals.
  - This is a small rate for LHC experiments, but a large rate for neutrino/DM experiments!





More details about the FPF/FASER-2 background : [here]

FASER2 : Muon Background studies

The muon flux has been calcuated at the FPF by FLUKA simulation the results can be found in a .dat/.txt format

The files can be found here:

.dat files V2: link (files 5Go each for Nm (μ^-) and Pm (μ^+))

.txt files V1 : here

- V1 and V2 of the files have different statistics, not same normalisation factors
- Need to read the ReadMe file carefully to understand, especially normalisation

FASER2 : Muon Background studies

Script to convert .dat and .txt files in root files using ROOT macros: [here]

Details about the functions:

This directory contains ROOT macros to read the output of FLUKA simulations and produce histograms that can be used as input for background studies.

The first two scrips are meant to be run at CERN (`lxplus`):

- `extract_fluence_data.C` : reads the `.txt` FLUKA output files and converts them into ROOT TTrees. Fluence data is directly available only for muons, but convers the entire FPF cavern.
- `extract_flux_data.C` : reads the `.dat` FLUKA output files and converts them into ROOT TTrees. Flux data contains complete description of all muons and neutrons crossing the entrance of the FPF cavern. It can be used to extract the flux for these particle species, but also other distributions.

The other scripts are used to produce the inputs histograms for the background studies. These are saved in a ROOT filed and accessed by the generator class in Geant4.

- `make_muon_fluence_histos.C` : produces 2D projections of the muon fluence rates, with a focus on the FPF entrance.
- `make_muon_flux_histos.C` : produces the muon flux at the FPF entrance, binned in slices along the x-axis.
- `make_neutron_flence_histos.C` : produces the 2D xy projection of the neutron fluence rate at the FPF entrance.
- `make_neutron_flux_histos.C` : produces the neutron flux at the FPF entrance, binned in slices along the x-axis.

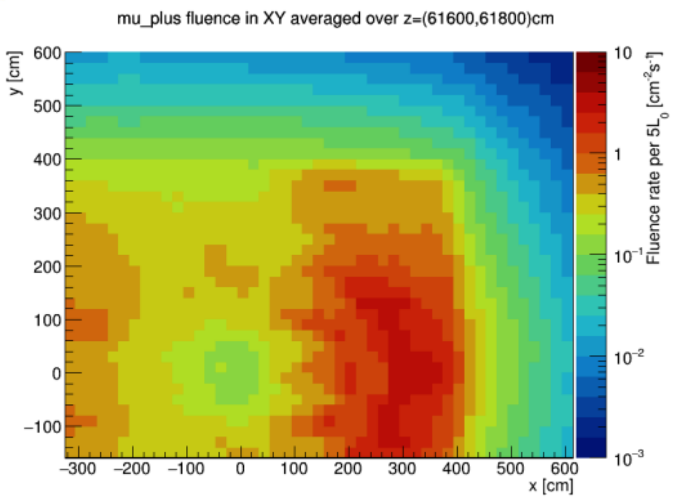
FASER2 : Muon Background studies

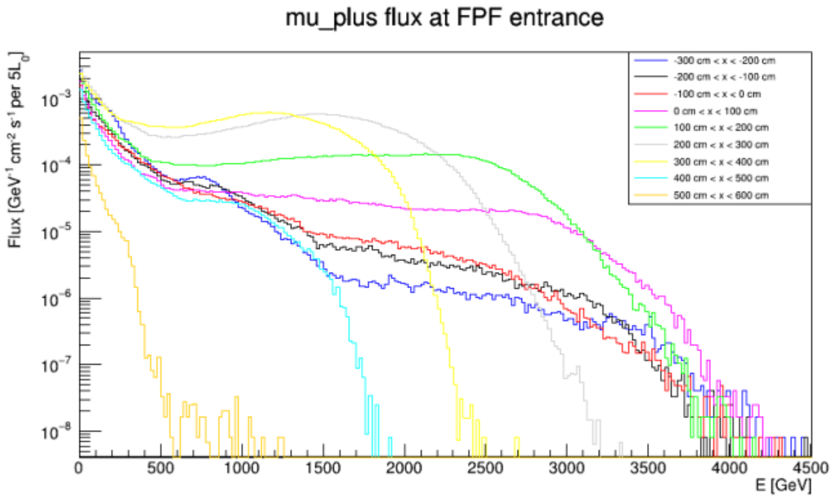
Goal: Study the background muon fluxes for FASER-2:

- Use of the muon background for track based alignment of the tracking stations
- Quantify effect of the Veto
- Impact of the background on the invariant mass plot ?

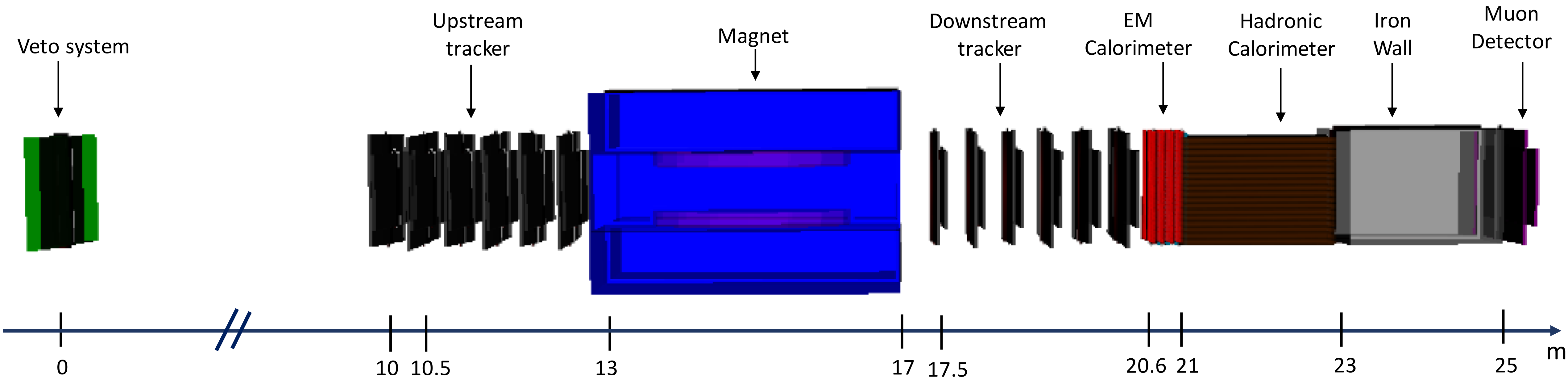
Good starting point:

- Getting used to ROOT and macros : (tutorial, can ask anyone PP department for help)
- Manipulating the background muons root files (here)
- Aim to redo the plots from this presentation : here (slides 9,10, 12)





FASER2 : Baseline geometry

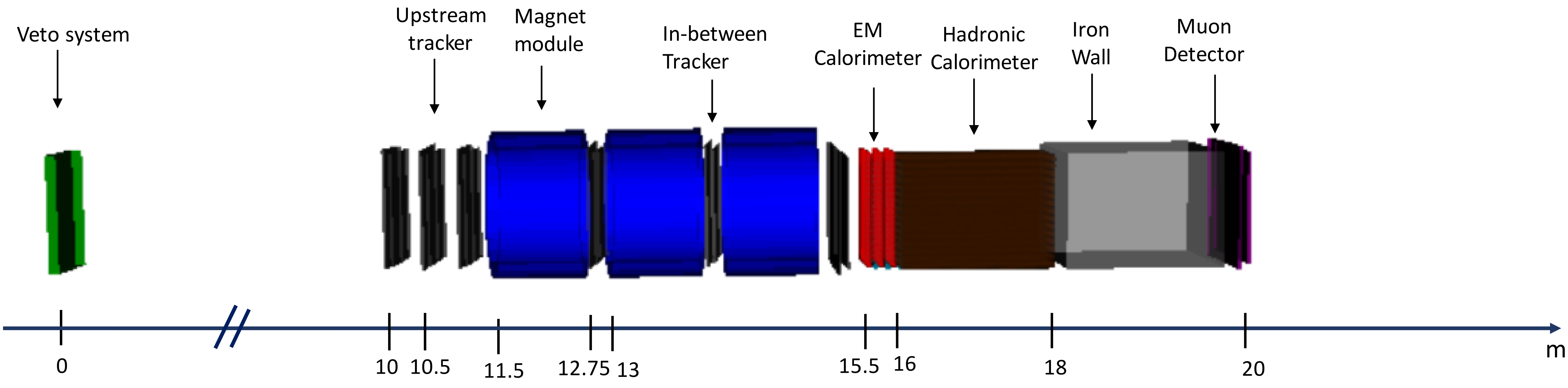


Veto system : 1 m X 2 m X 100 mm

Upstream tracker : 1 m X 3 m X 40 mm
Downstream tracker : 1 m X 3.5 m X 40 mm

Magnet : 1 m X 3 m X 4 m

FASER2 : Alternative geometry



Veto system : 1.6 m X 1.6 m X 100 mm

Upstream tracker : 1.6 m X 1.6 m X 40 mm
In-between tracker : 1.6 m X 1.6 m X 40 mm
Downstream tracker : 1.6 m X 2.5 m X 40 mm

Magnet module : diameter = 1.6 m X z =1.25 m (0.625 Tm)

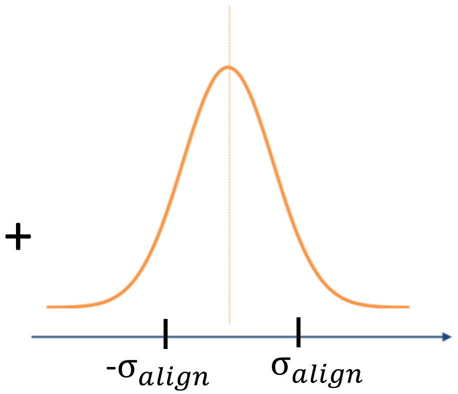
Alignment studies

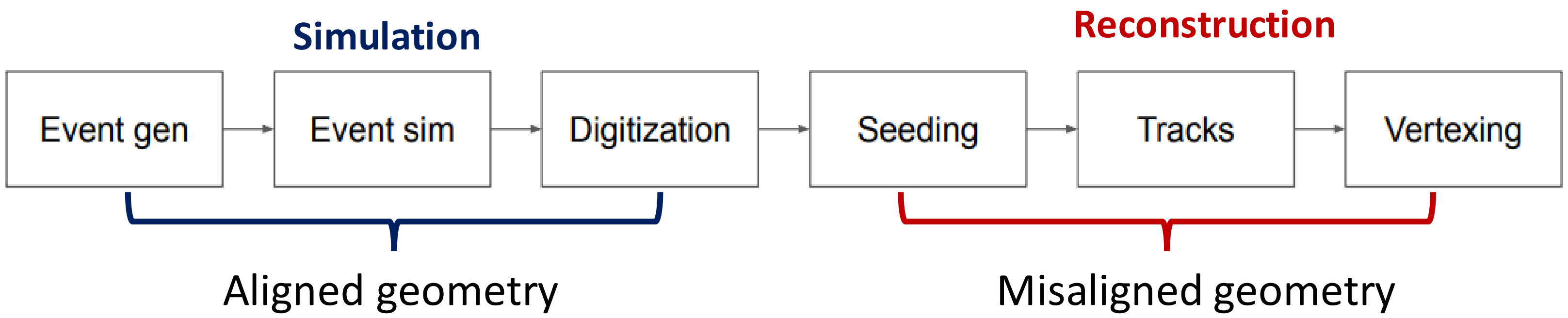
Alignment of the detector is expected one of the main contributor to the mass resolution
Study of the expected effect of misalignment on mass, momentum resolution plots

Implementation of the misalignment in the tracking:

Transformation to tracking station applied:

- Misalignment in translation = Aligned position +

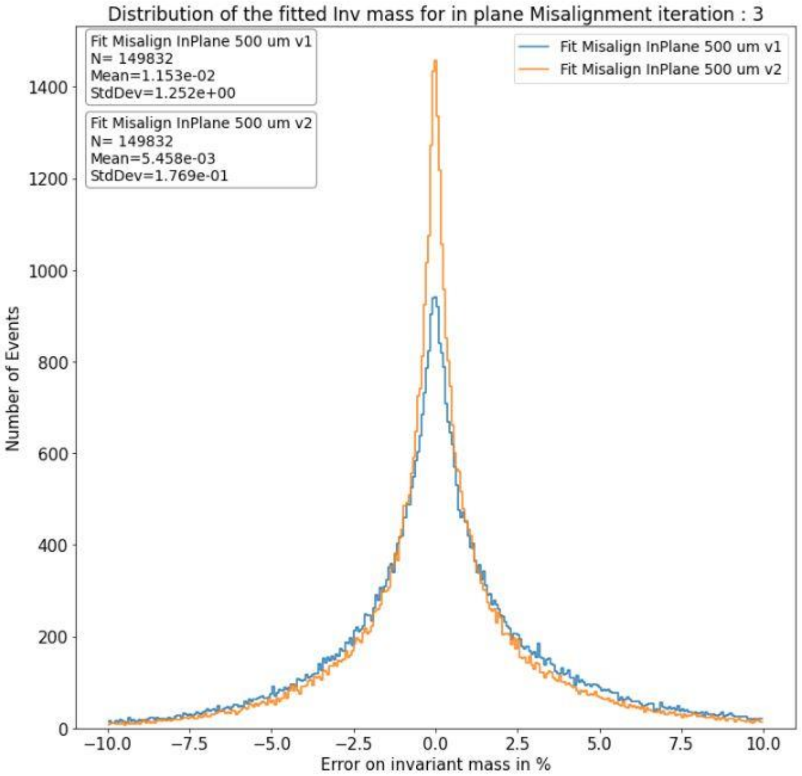




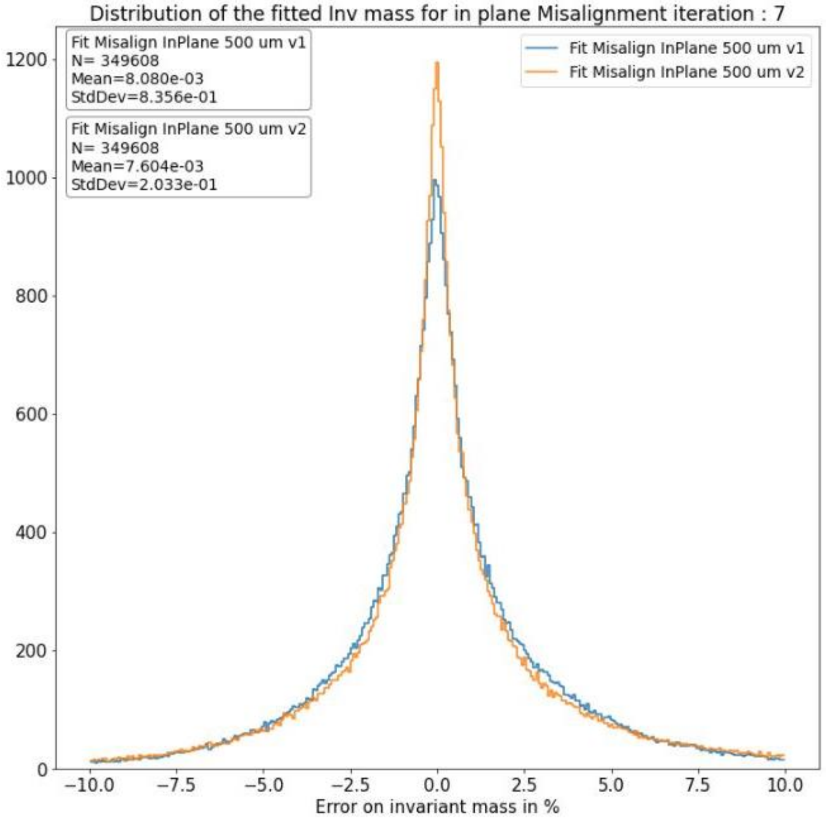
Misalignment studies: Multiple iteration

- Statistical averaging with same values of misalignment to get rid of effect of randomness
- Compare two different set with same value of misalignment

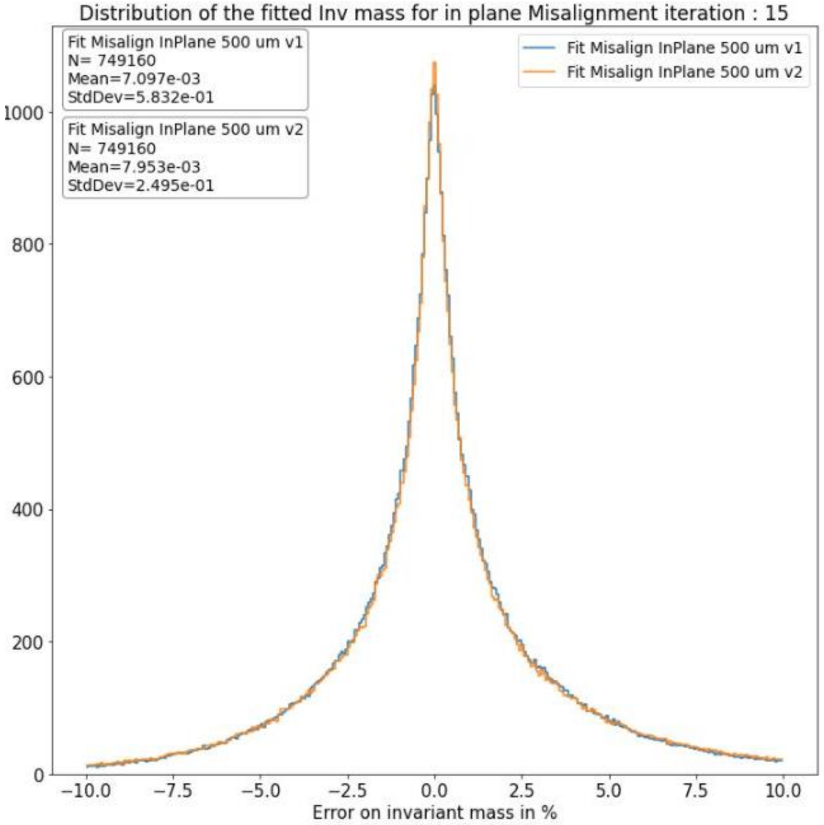
3 Iterations



7 Iterations

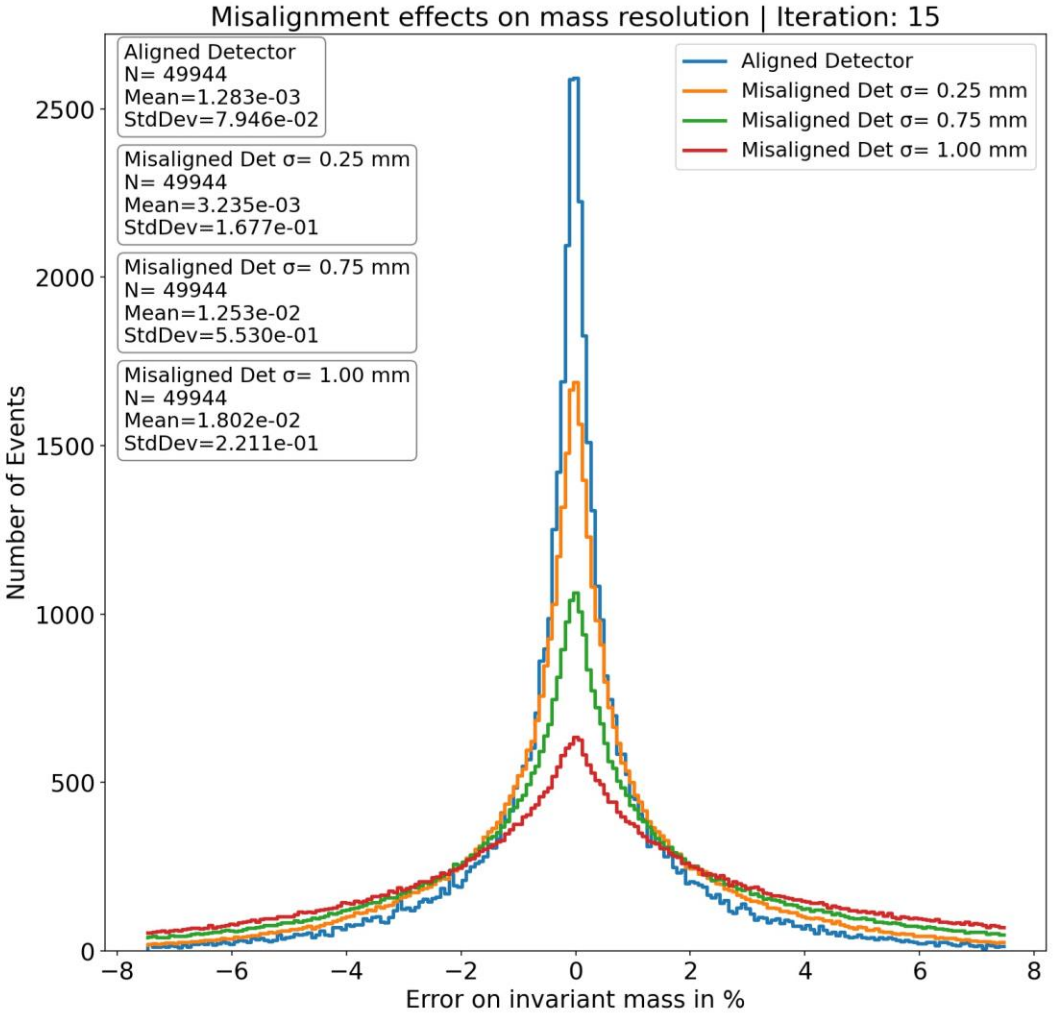


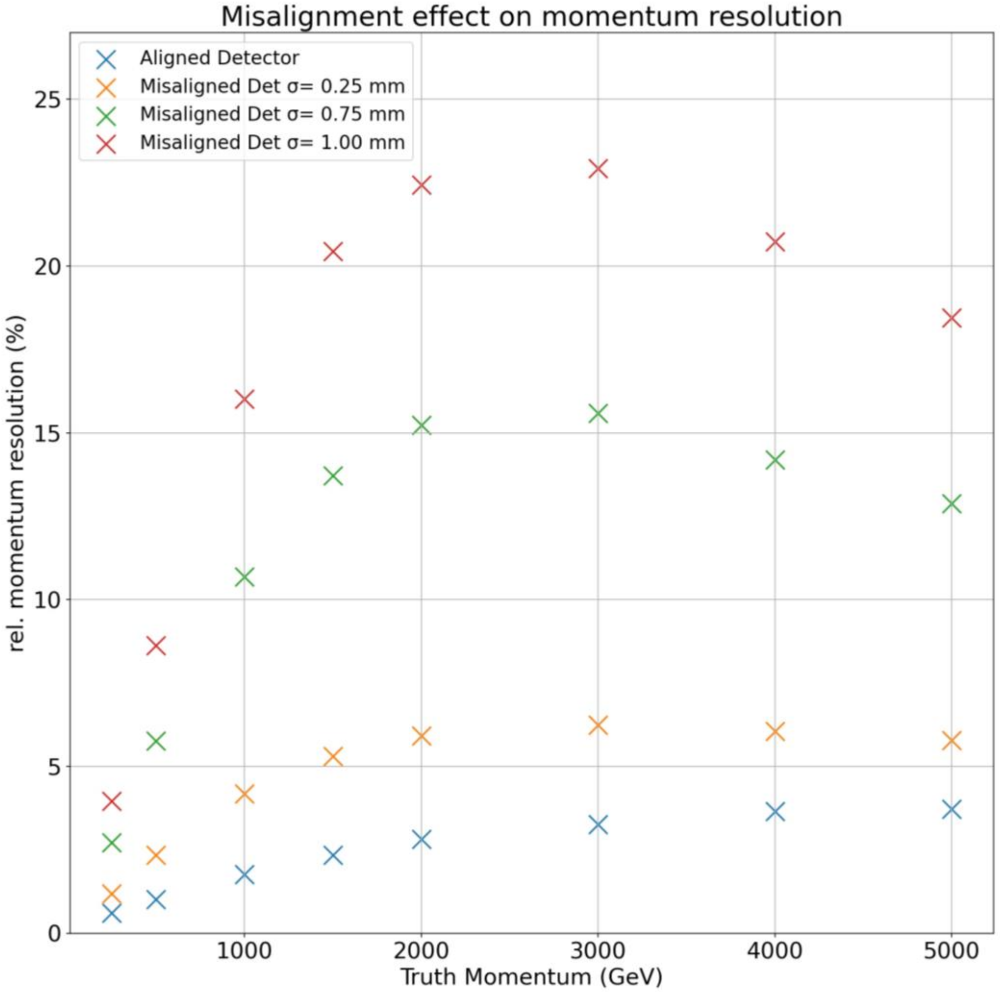
15 Iterations



Misalignment studies: Multiple iteration

Statistical averaging with 15 iteration of same misalignment to lessen effect of randomness on results





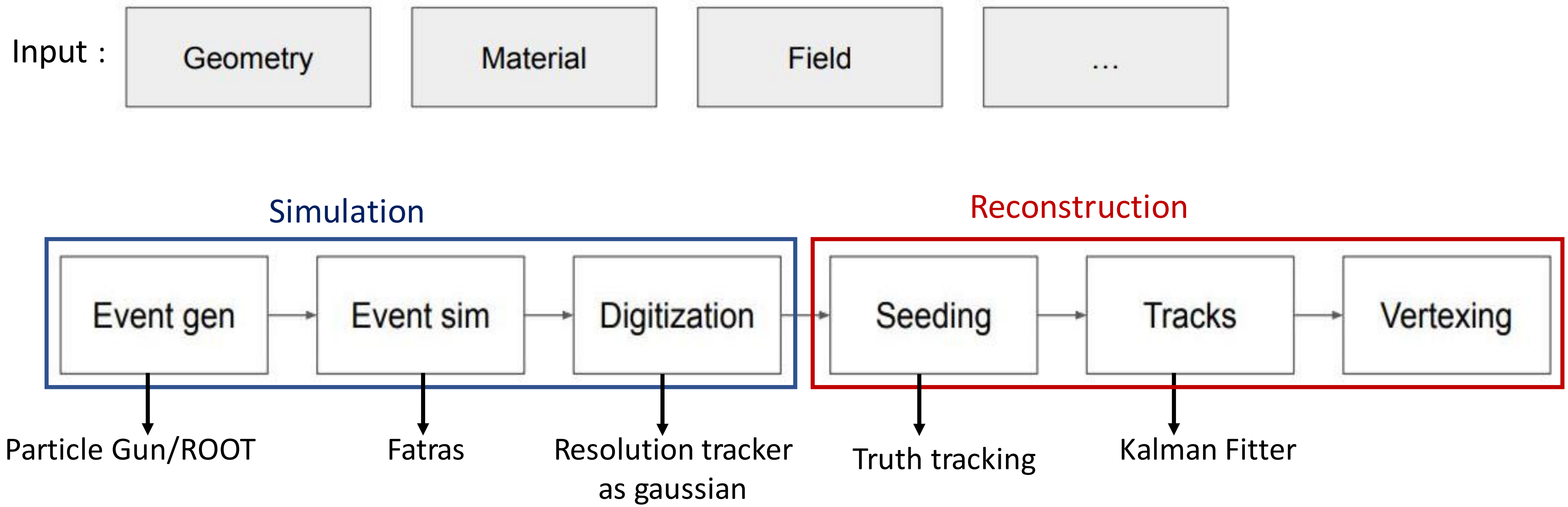
Misalignment > 750 µm starts to have significant impact on mass resolution

Expert expect maximum misalignment should be around 250 µm with mechanical precision

ACTS Technical details and implementation

ACTS track reconstruction algorithm

Python binding function with the simulation and reconstruction algorithm: [here]

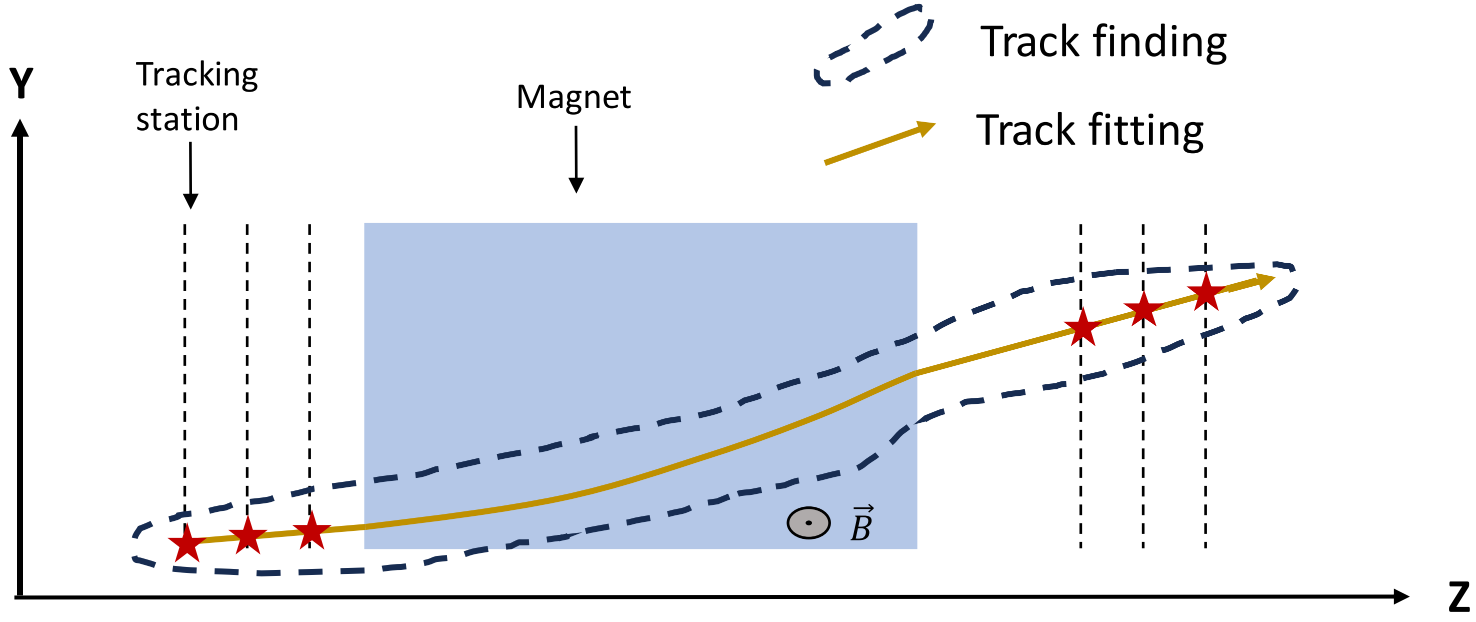


ACTS code in a nutshell

- The code of ACTS can be found in the github page
- Installation step (that works) is described here
- Code is divided in two part:
  - ❑ 95 % written in C++ to describe Core (Geometry, Magnetic Field, Propagator,, Track Fitting algo, …) and Plugging
  - ❑ 5% of the code is a python binding interface to operate the simulation and track reconstruction (easy to change parameter)
    Example : Truth tracking
  - ❑ Every modification to the C++ code must be compiled to be used

Tracking in a nutshell

Processus:



$$p\ [\text{GeV}] = 0.3\ \times B[T] \times R[\text{m}]$$

Implementation of ACTS for FASER2

Main difficulties that needed to be overcame during the implementation:

- ACTS main features more tuned toward cylindrical, colliders detector
- Numerical uncertainty in track fitting for η > 6

Implementation process:

pyg4ometry
Build geometry of setup

→

FASER2 geometry description
GDML → ACTS

→

FASER2 SciFi tracker material (1% X0)
ACTS material mapping

↓

Rotation of the axis : Z→X
- Rotation of the geometry and Event Data
- Needed to adapt forward experiment to ACTS and to avoid pole at high η

FORESEE[8]
Forward BSM Monte Carlo

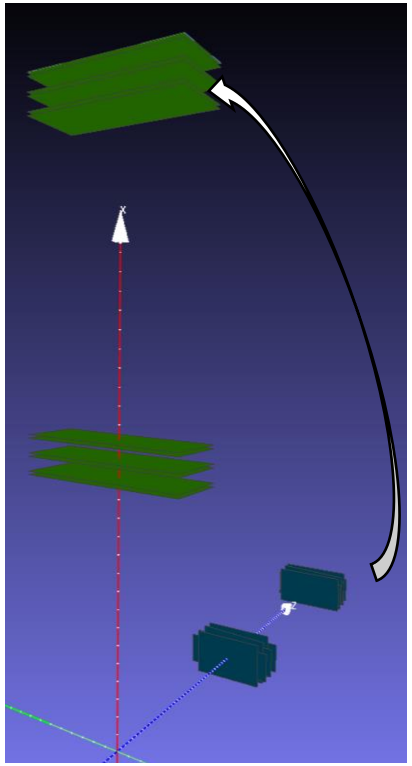
→

Event Data for FASER2
HepMC → Root

↑ (to Rotation of the axis)

→

ACTS
Simulation
Track reconstruction



ACTS track reconstruction algorithm: Geometry

Python binding function with the simulation and reconstruction algorithm: here

Input : Geometry | Material | Field | ...

Geometry: Telescope Detector allows to create rectangular layers

```
detector, trackingGeometry, decorators = acts.examples.TelescopeDetector.create(
    bounds=[1000, 3000], positions=[10000,  10500, 11000, 19500, 20000,20500], binValue=0,thickness=4, offsets=[0,0],
```

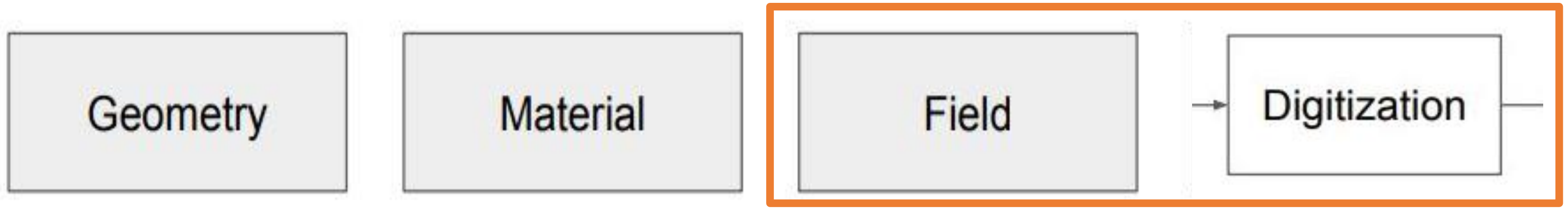
- Parameters easy to modify:
  - Dimension of the detector : Boundary, thickness, shape
  - Position of the trackers
  - BinValue=0 (Rotation of the detector to the X axis)
  - Offset of the detector

- Material: Parameter of the material implemented in Telescope Detector creation
  - Density
  - Radiation Length
  - Interaction length

ACTS track reconstruction algorithm

Python binding function with the simulation and reconstruction algorithm: [here]

Input :



Field: Restricted BField

```
field = acts.RestrictedBField(acts.Vector3(0* u.T, 0, 1.0 * u.T))
```

- Field Strength easy to modify in the python binding

- Volume hardcoded to the C++ code, each modification to volume (CuboidVolumeBounds) and the position (Translation3) must mean compiling the code again (I need to change that)

- **Digitization**: Parameter that simulate the resolution of the detector by a Gaussian

```
"value" : {
  "smearing" : [
    {"index" : 0, "mean" : 0, "stddev" : 0.1, "type" : "Gauss"},
    {"index" : 1, "mean" : 0, "stddev" : 0.1, "type" : "Gauss"}
  ]
```

ACTS track reconstruction: **Output**

3 Main root file output from the Track reconstruction algorithm:

Tracksummary_fitter → Results from the track reconstruction with reconstructed and truth values for:

- Q/P
- Theta
- Phi angle

Trackstates_fitter → Results from the track reconstruction for each tracking layers (reco and truth):

- Position of the hits in X, Y, Z
- Q/P, angles theta, phi

Performance_vertexing → Results from the vertex reconstruction with reconstructed and truth values for:

- Truth position of the vertex
- Reco position of the vertex
- Covariance between parameters

Example of Analysis of those output can be found here: Written in jupyter notebook format

ACTS results: Long-lived Particles searches