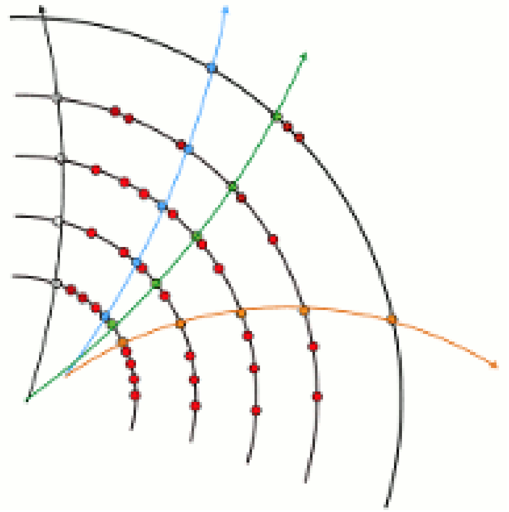
ACTS project status and FCC integration

Julia Hrdinka (TU Wien)
On behalf of the ACTS & FCCSW Team



Motivation

FCC(-hh) design study: track reconstruction is crucial for benchmark detector studies

LHC track reconstruction software

- Well tested & high performant code (10^{10} events with 10^{3} tracks/event)

Current R&D and development ongoing for HL-LHC

- Challenging (pile up) environment: 140-200
- Requires substantial updates to algorithmic code
- Adapt to new developments in computing hardware (concurrency)
- Long-term maintainance of the software

=> FCC should profit from development

A Common Tracking Software - ACTS

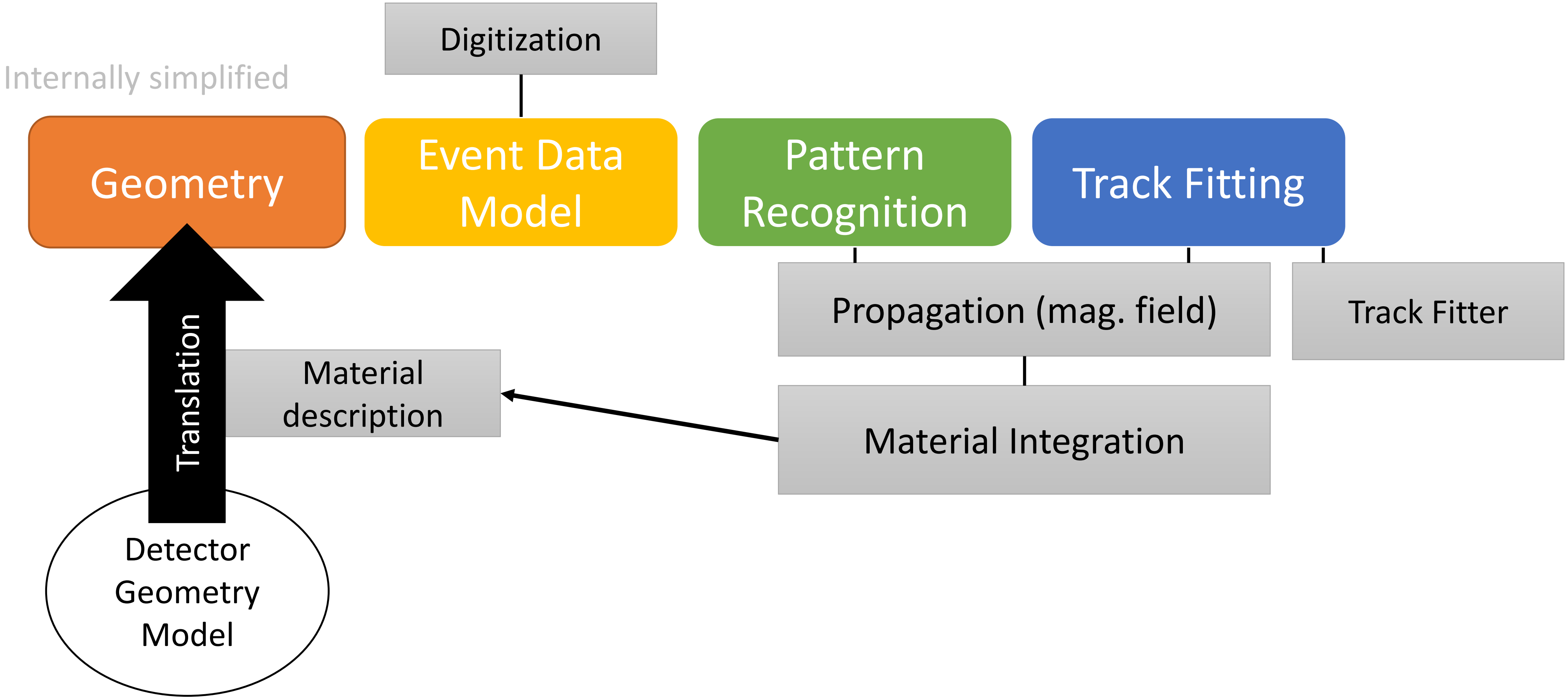
Starting point: ATLAS track reconstruction software

- a-common-tracking-sw https://gitlab.cern.ch/acts/a-common-tracking-sw
  - Core package – contains base components of tracking code
  - Framework and experiment independent
  - Minimal dependencies: Eigen, Boost
  - Modular – users can extend with their implementations
  - Plugins for experiment specific parts
- acts-test-fw https://gitlab.cern.ch/acts/acts-test-fw
  - Test framework (mimics Gaudi) using the core package
  - Examples and testing of core

  => For testing & development only

Find our homepage:
http://acts.web.cern.ch/ACTS/index.php
Subsribe to our mailing lists:
acts-users@cern.ch
Weekly Meetings
https://indico.cern.ch/category/7968/

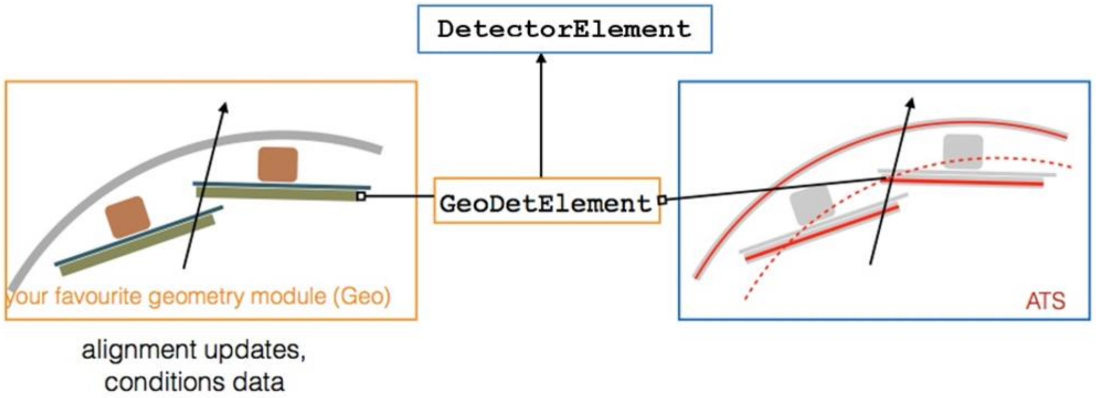
Track Reconstruction components



Geometry

ACTS provides plugin mechanism for different geometry back-ends

- ➢ Automated translation into ACTS geometry
    - DD4hepPlugin (FCC)
    - TGeoPlugin (ROOT)
- ➢ Direct link between underlying geometry and ACTS geometry



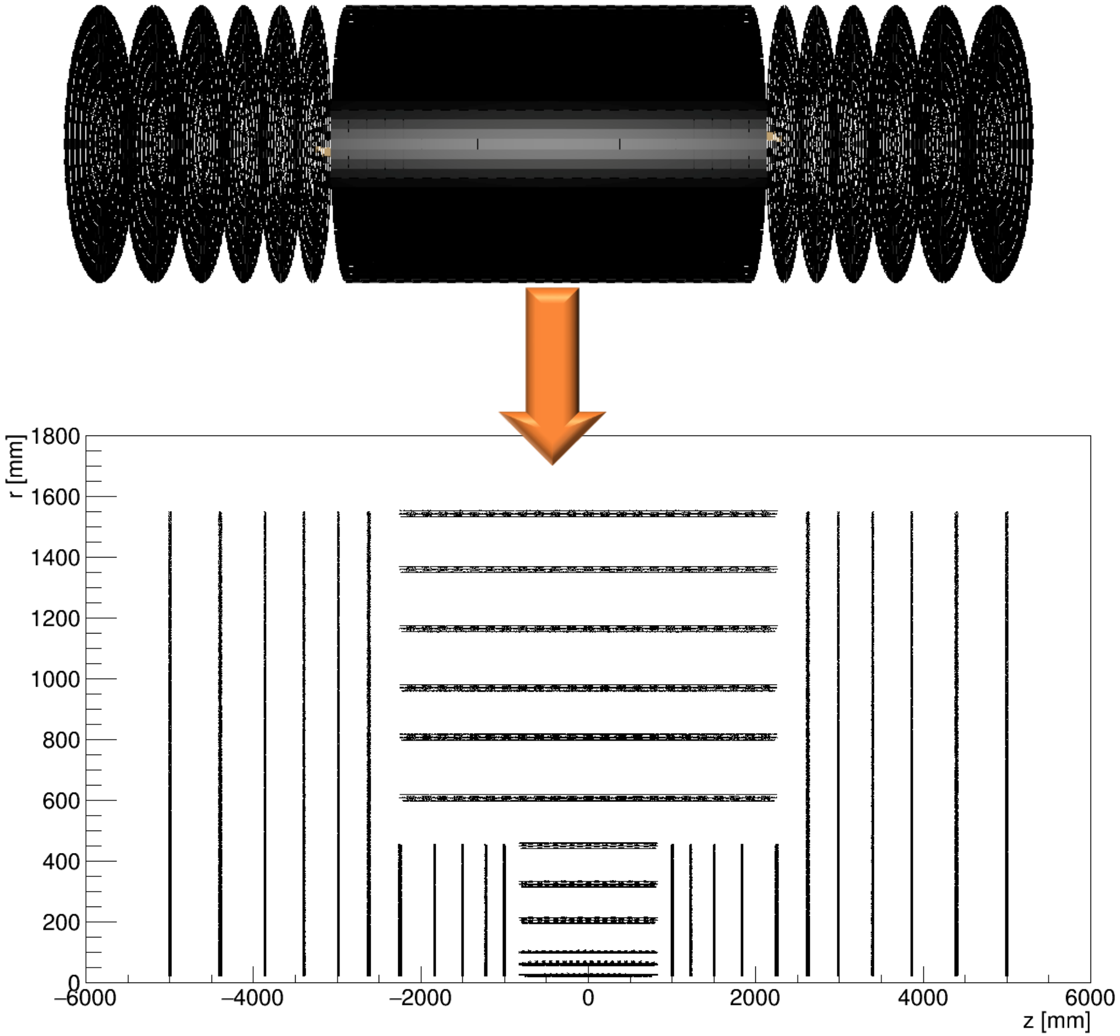
Fast but accurate material description of tracker needed for reconstruction & fast simulation.

- ➢ Automated Material transcript from full simulation geometry

Consistent geometry input in FCCSW

One common geometry source for all simulation types + reconstruction

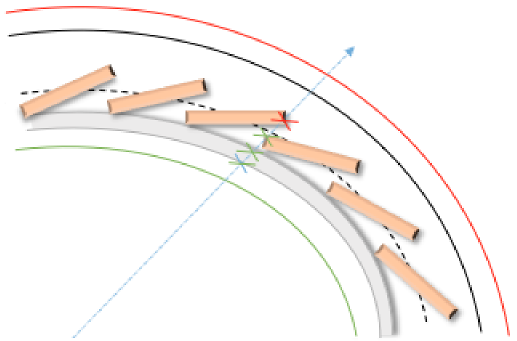
- FCChh Tracker was designed with TkLayout tool (see talks of V.Volkl, Z.Drasal)
- Exported to xml file readable by DD4hep
- Automatic translation to ACTS using the DD4hepPlugin

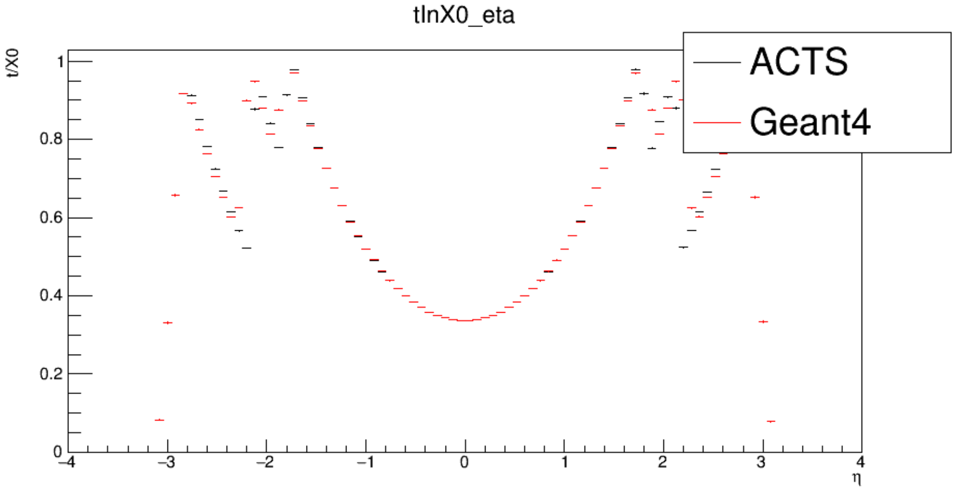


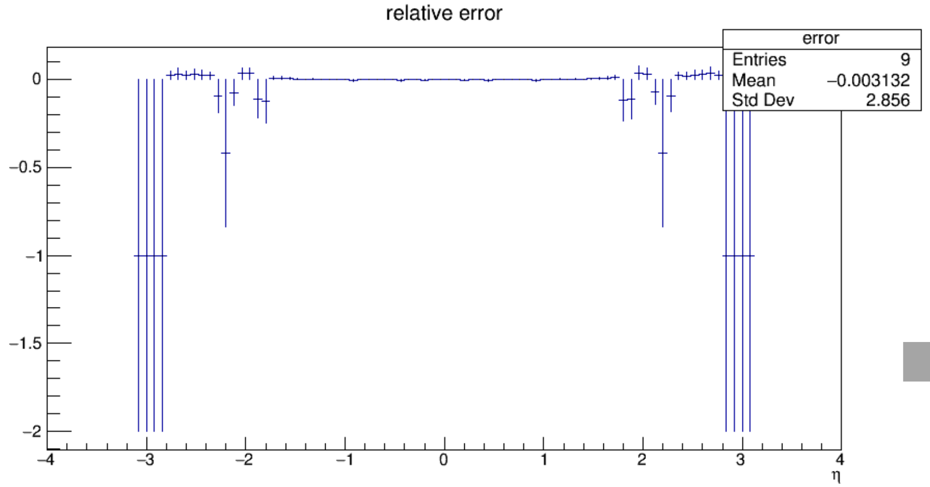
DD4hep

ACTS

Material mapping





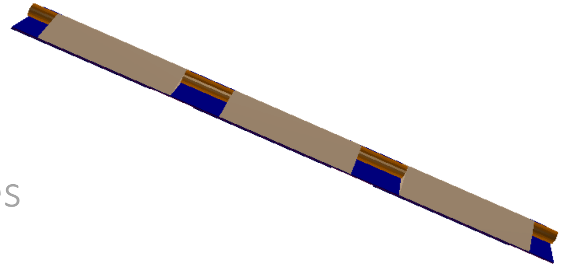


For complicated geometries:

<u>Automated mapping algorithm</u>

- Mapping onto surface based ACTS geometry
- Described on 2D grid
- User decides granularity of grid and which surfaces should carry material
- Description consistent at the order of 1%

using test detector
with complex modules



Event Data Model

ACTS algorithms are targeted to be EDM independent

- once fully completed, one should be able to use ACTS with a custom EDM

ACTS also provides an EDM which you are free to use: **optimized for CPU performance**

Eigen-3.2.7, g++ 4.9.2 –O3, 100M operations

	Dynamic sized	Fixed size	converted
M(2,5) x v(5)	0.154546s	0.00747539s	0.0116945s
M(5,2) x M(2,5)	0.217145s	0.031977s	0.0326164s

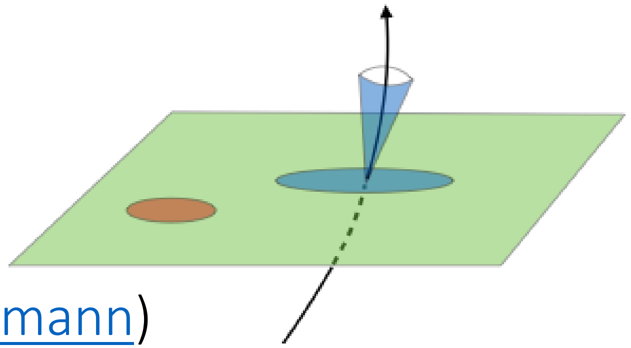
→ Copy dynamic sized matrix/vector into fixed sized implementation

$\Rightarrow$ Fixed sized operations 8-20 times faster

Two base classes: Measurement & TrackParametrization

- fixed sized
- Parameters & size can be customized => template implementation

$$default : (l_0, l_1, \varphi, \theta, \frac{q}{p})$$

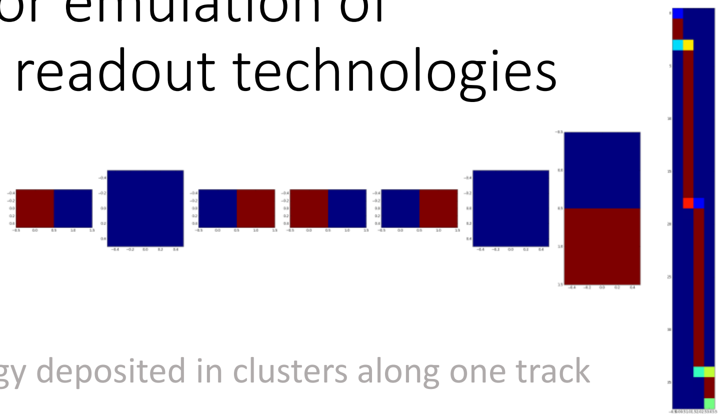


=> Direct translation to and from FCC EDM (PODIO, see talk of J. Lingemann)

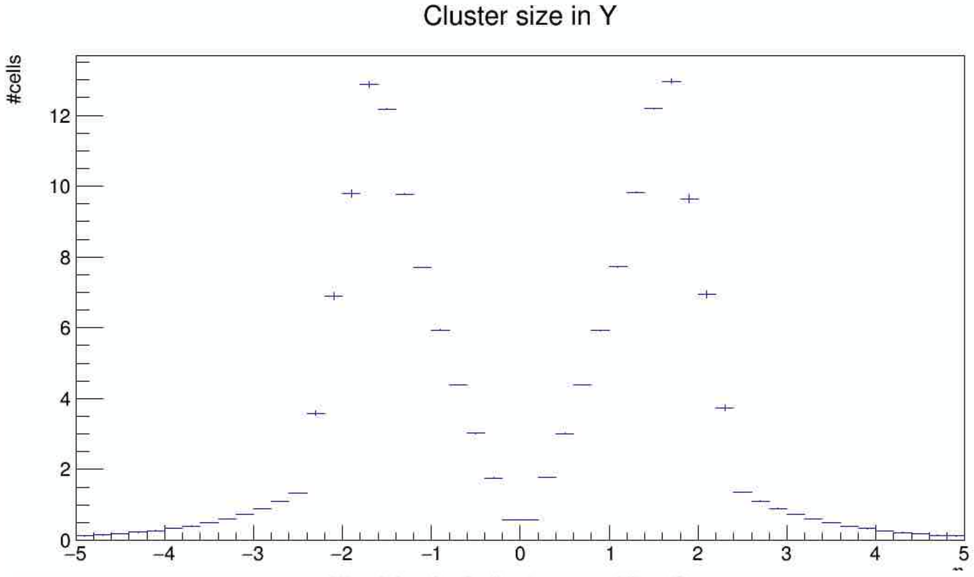
Digitization

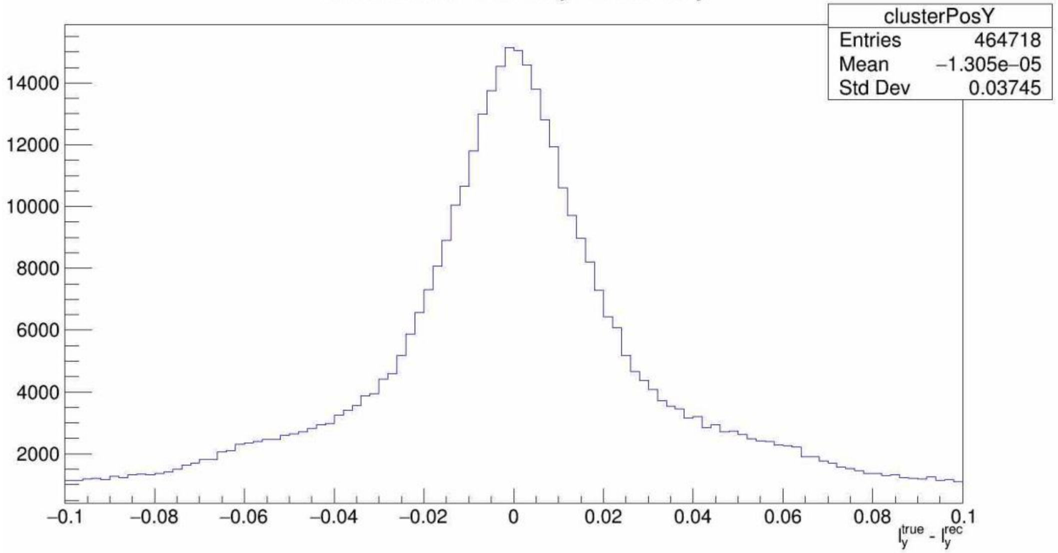
Geometric digitisation

- Imported module from ATLAS
- Calculates cluster sizes and path lengths in individual pixels
  - Validated against ATLAS Geant4 simulation
- Takes Lorentz angle into account
- Allows for emulation of different readout technologies



Energy deposited in clusters along one track





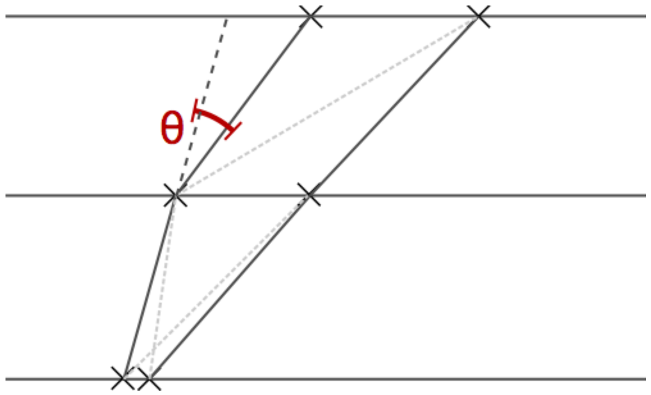
Simple detector from DD4hep geometry input

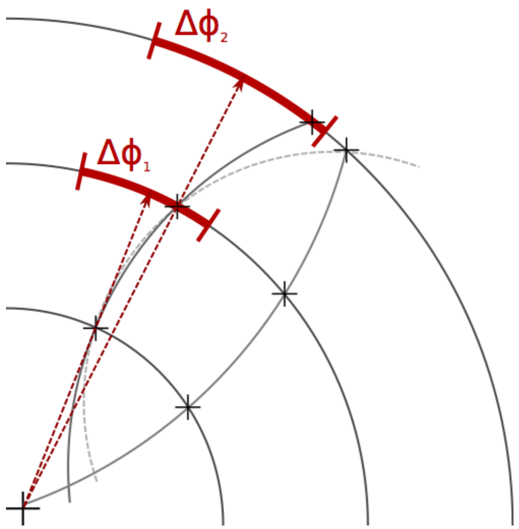
Pattern Recognition

seed -> track candidate -> reconstructed track

Track finding modules development has recently started

- Seeding
  - Combination of measurements that are likely to be part of a particle track
  - Provide a direction for the combinatorial track finder
- ACTS seeding
  - ATLAS seeds consist of 3 measurements compatible with a helix traversing the interaction region
  - Independent of # of measurements per seed & detector geometry
  - Current ACTS seeding plugin prototype is for cylindrical detector
  - ATLAS seeding optimizations to be included in ACTS example implementation





Seeding studies

Two different hit container geometries are studied

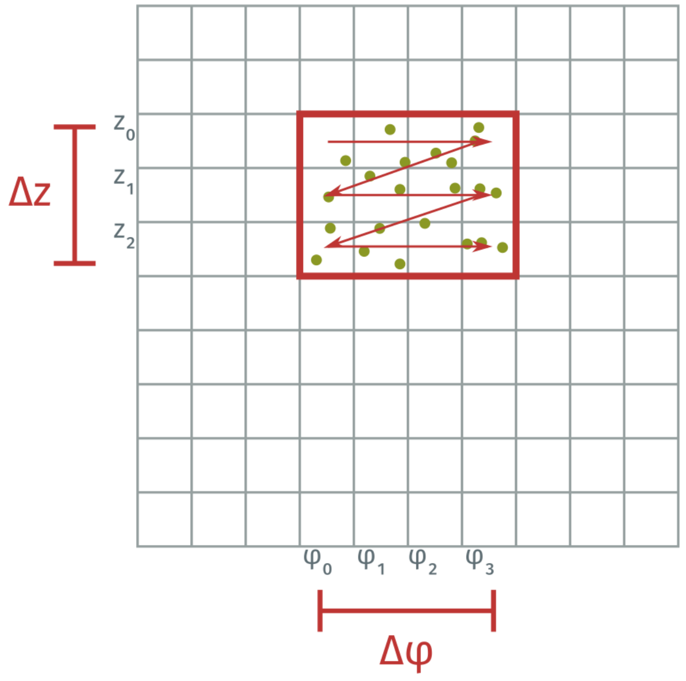
- ➢ 2D Binning (φ,z) + ordering in r (ATLAS)
- ➢ 3D Binning (φ,z,r)

⇒ Test which memory layouts are most efficient

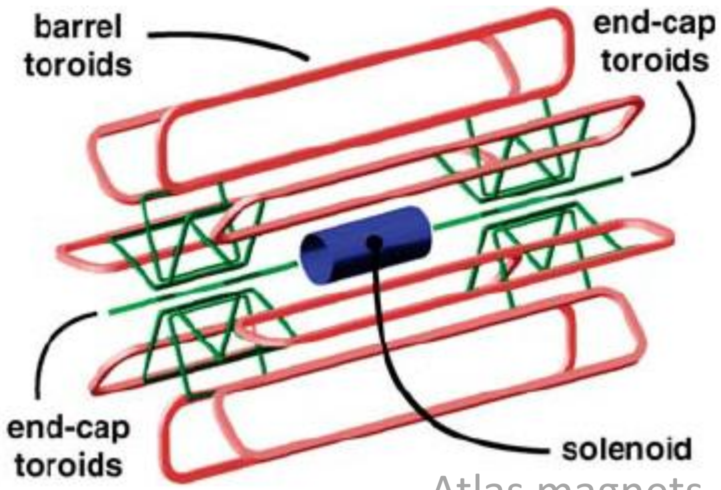
Seeding optimizations

- ➢ Runtime of tracking linear with number of seeds
  - ▪ Reduce number of seeds
    - – Minimum pT cut
    - – Cut on maximum distance from interaction region
    - – Cut on kink of tracks
    - – Neural network based classification
  - ▪ ATLAS finds about 60 times more seeds than final tracks
  - ▪ Future plan to revive parallel track finding approaches

Algorithm automatically returns hits in a given region



Track fitting



Atlas magnets

Extrapolation & Propagation

- Mathematical propagation & material effects separated
  - Plug in different propagators
  - Plug in different material effects integration
- Runge Kutta propagator usable with costum magnetic field service (template argument)
- ACTS provides default magnetic field service
  - Possibility to read in magnetic field from txt or csv file (FCC field map can be read in)
  - Propagation through complex magnetic fields possible
  - ATLAS has a lot of experience with complex magnetic fields

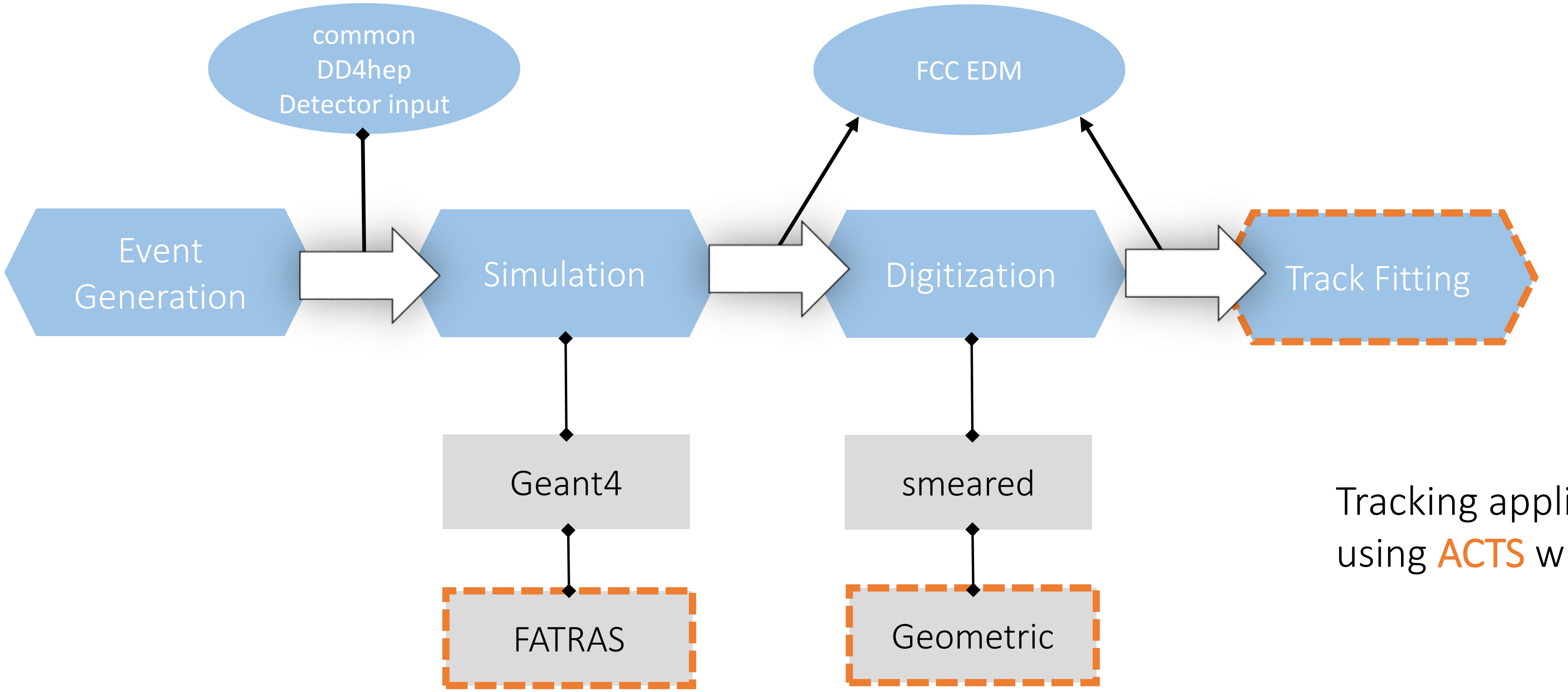
First Kalman fitter prototype is implemented

- Gain matrix formalism
- Hole finding on the fly (Extrapolator gives that for free)
- Common backbone with Gaussian Sum Filter

FCCSW - ACTS integration

ACTS integration in next FCCSW release

- Latest ACTS release is in synch with FCC development
- First application tests inside FCCSW (see talk of V. Voelkl)

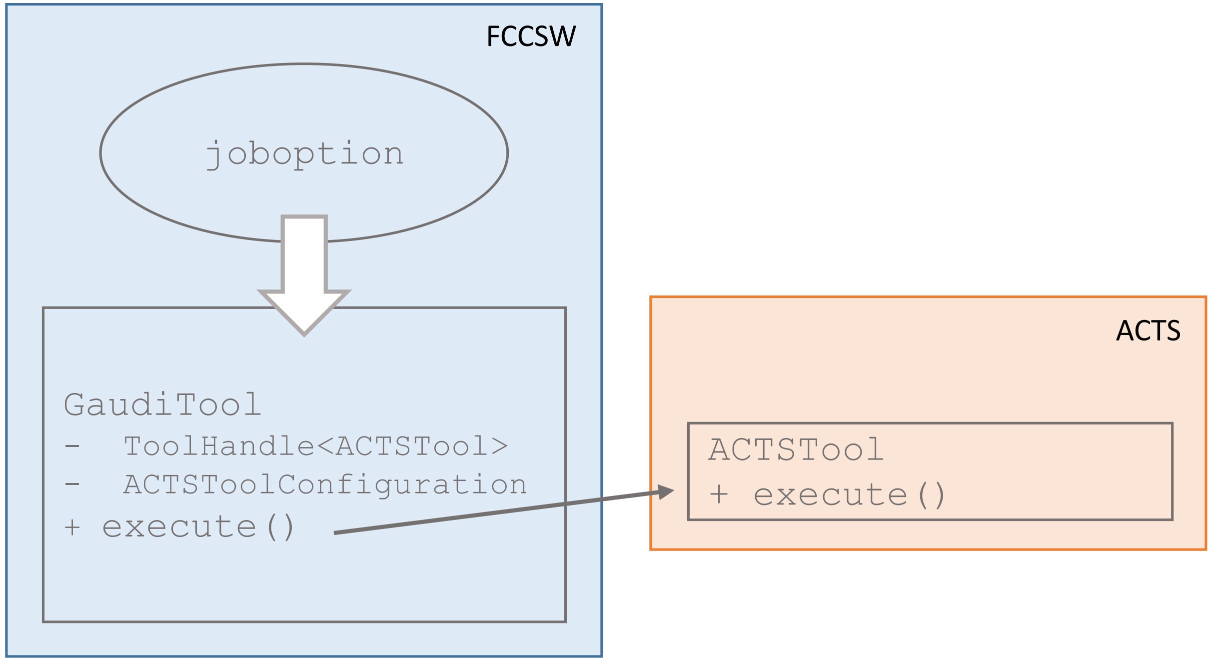


Tracking application
using ACTS within FCCSW

Integration into FCCSW

FCCSW Event processing framework: Gaudi

- Gaudi wrappers (Services, Tools, Algorithms) configured by python job option file internally using ACTS
  - FCC **magnetic field service** – interface to ACTS BField
  - **Geometric digitizaion** algorithm – using ACTS digitizaion
  - **Track fitting** algorithm – using ACTS track fitting tools
  - **Fast track simulation** (Fatras) as a simulation option – embedded in Geant4 simulation Kernel



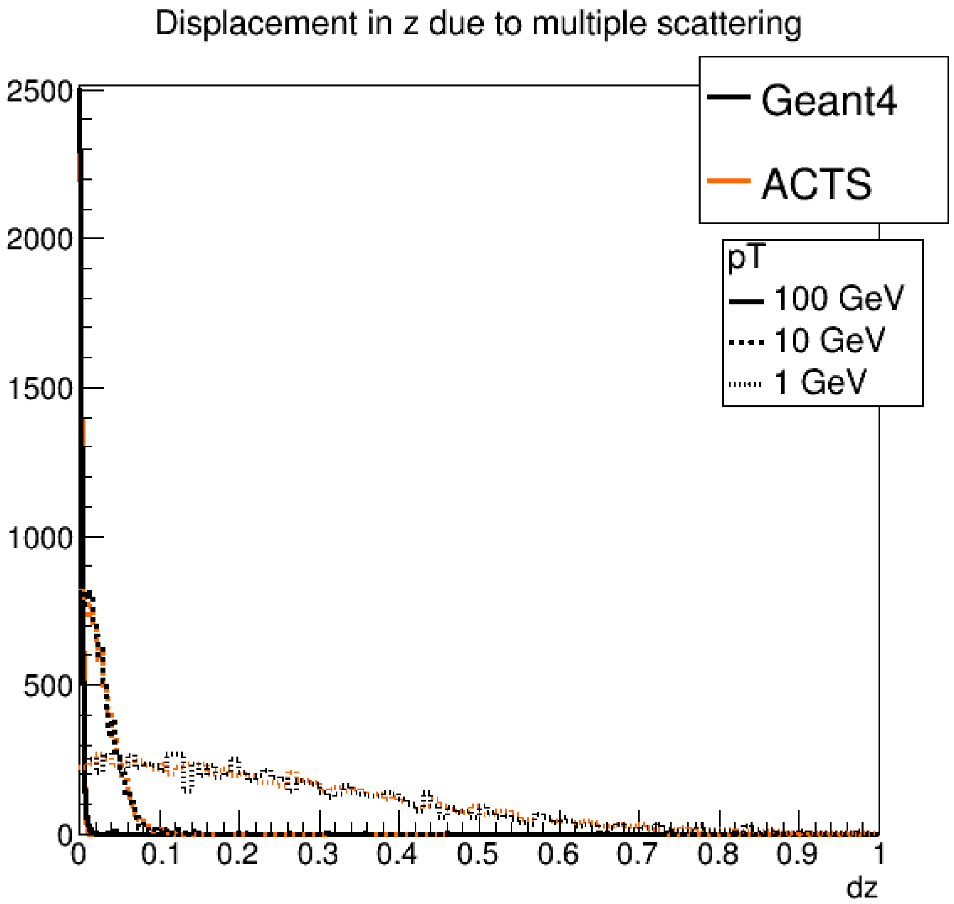
Fast Track Simulation (Fatras)

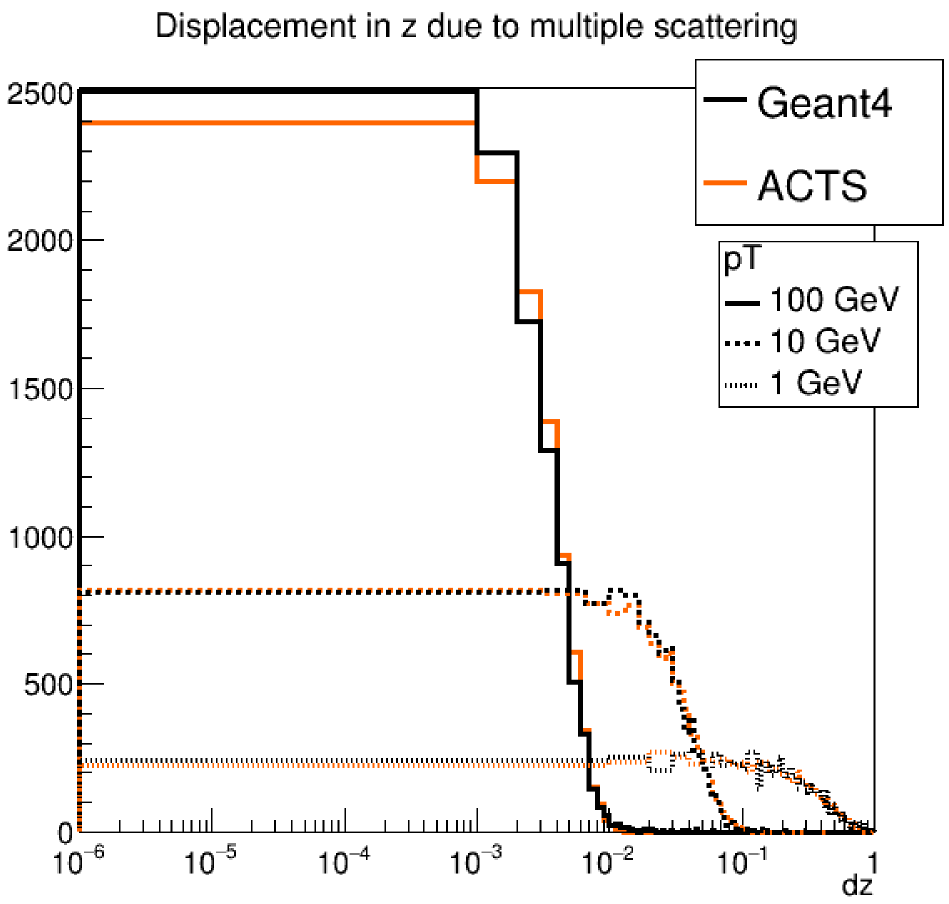
Fast Simulation based on the the simplified reconstruction geometry

- Extrapolation + material effects
- Comparison of material interactions
  - Test of quality of material description and interaction in ACTS with respect to Geant4
- Allows for first timing estimates for parts of the FCC-hh reconstruction
  - Extrapolation of O(10k) particles through FCC-hh detector

Material integration validation I

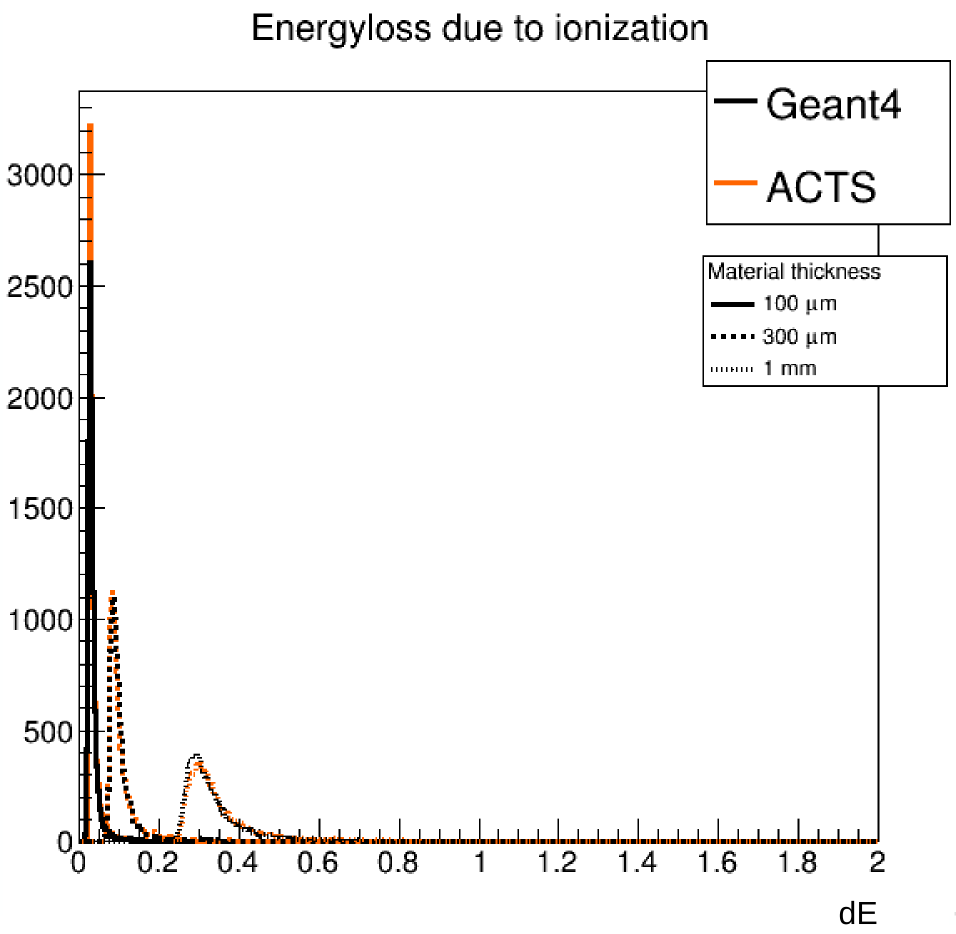
Muon, Cylindrical silicon layer with 0.1 mm thickness

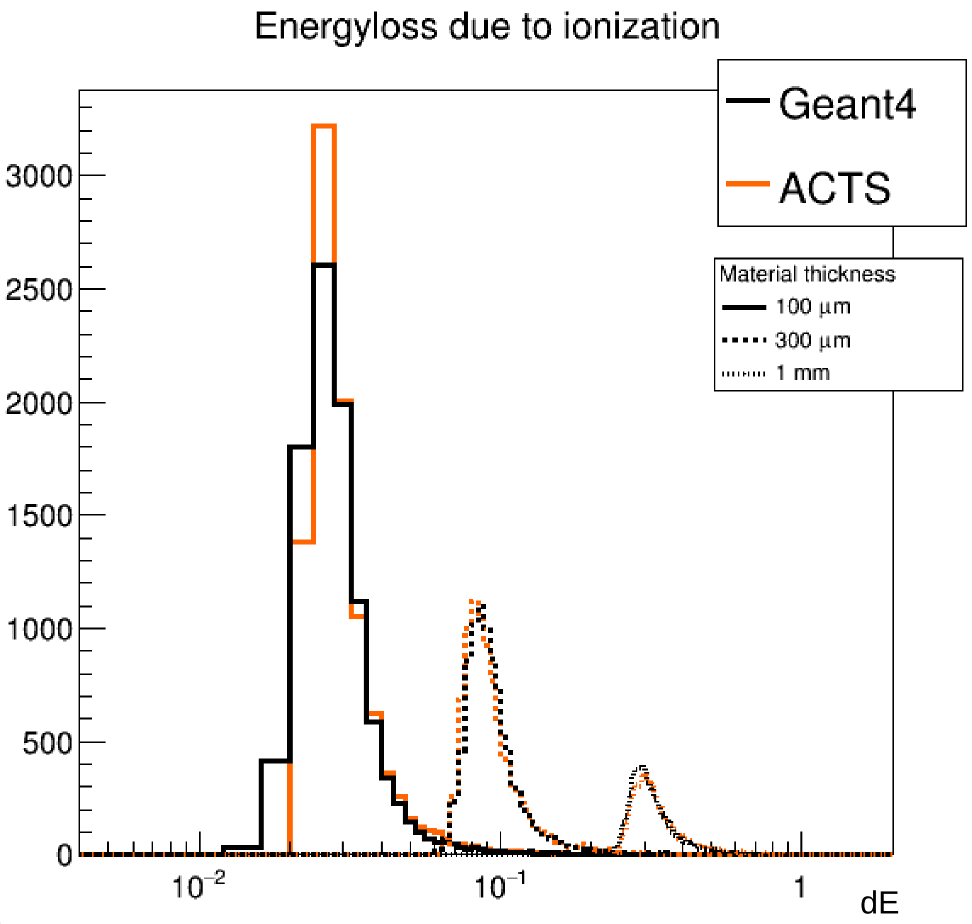




Material integration validation II

Muon 1GeV, Cylindrical silicon layer



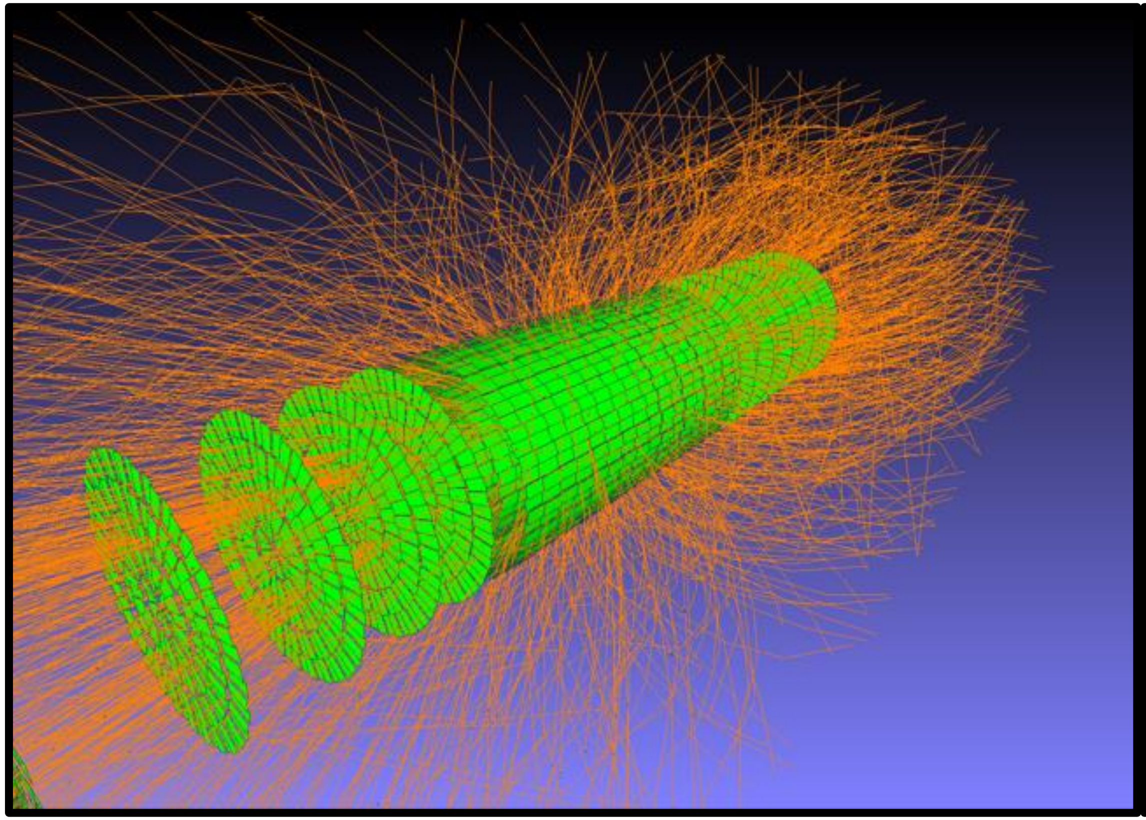


High multiplicity & timing tests

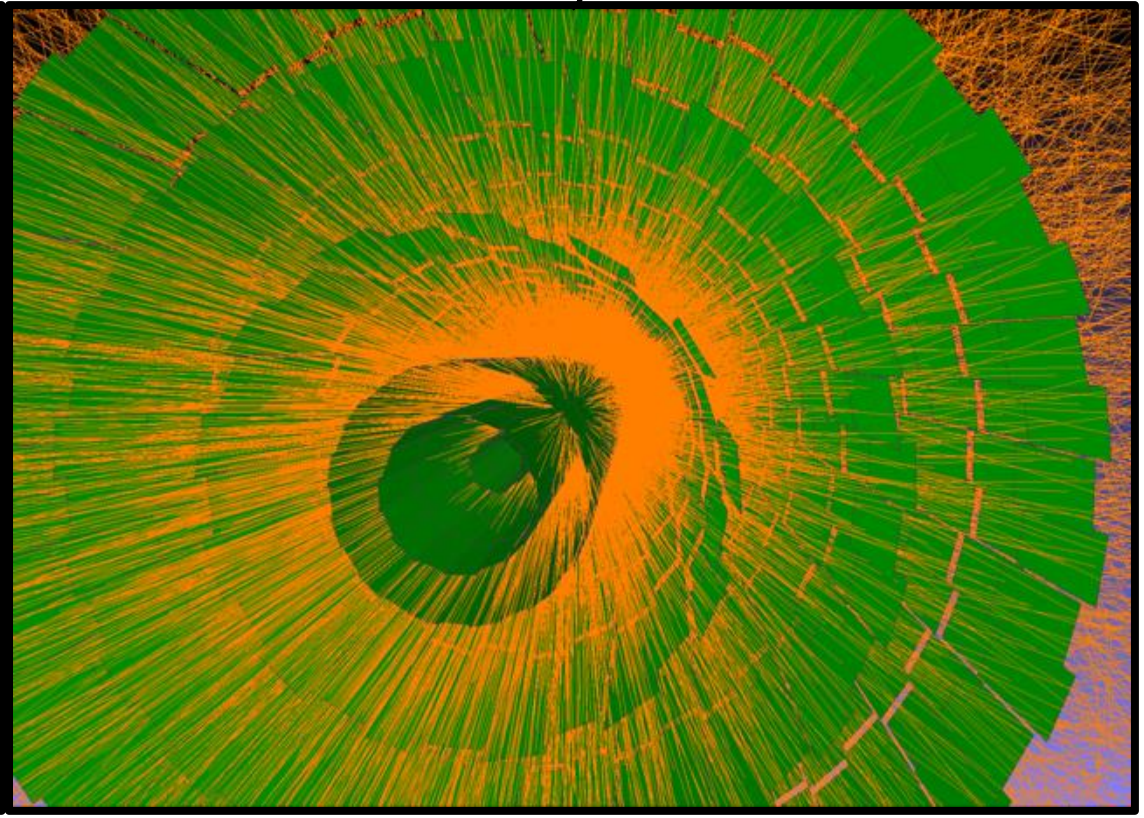
- Events generated with Pythia and overlaid to a gg->H event
- FATRAS simulation w/o material effects
- Using current FCChh detector

~3s/event simulation time for particles > 900 MeV

$\mu = 200$



$\mu = 1000$



Concurrency

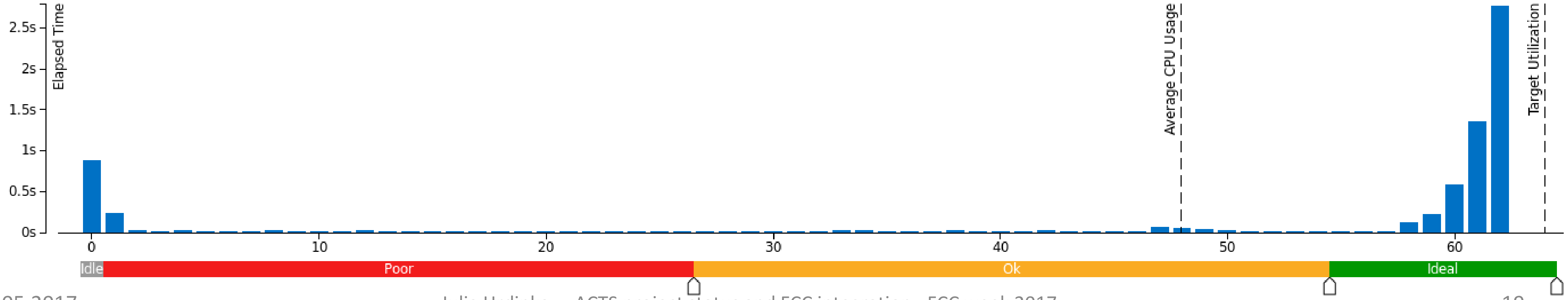
Present and future hardware moves towards concurrent computing

- Needed in order to deal with future high pile up environments
- **Testing ACTS algorithms on concurrent event processing**
  - Done inside ACTS mini test framework
  - Based on OpenMP, Number of threads can be set
- Predictive extrapolation example
  - FATRAS fast simulation without material effects
  - Testing the load of up to 64 threads
  - Shared geometry and magnetic field

Intel Xeon e5-2698 v3, 2 sockets
32 Cores, 2 threads per core
64 Processors(cpu's)

CPU Usage Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU usage value.



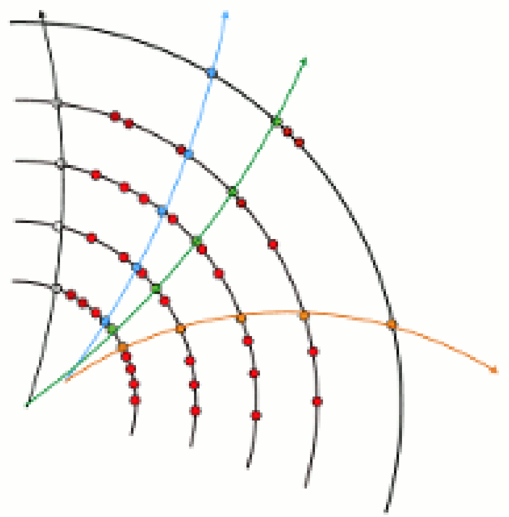
Conclusion, Timescale and Outlook

Pattern recognition is the last corner stone missing in ACTS for first FCC-hh track reconstruction tests

Geometry	Event Data Model	Pattern Recognition	Track Fitting
done, validated	done, validated	started	done, validation ongoing

- ATLAS demonstrator with ACTS planned for autumn 2017
- Pattern recognition for ATLAS/FCC-hh should be highly interchangeable
- Full track reconstruction studies for FCC-hh foreseen by end of the year
- Should give time to validate the CDR assumption, based on simplified tools

Back up

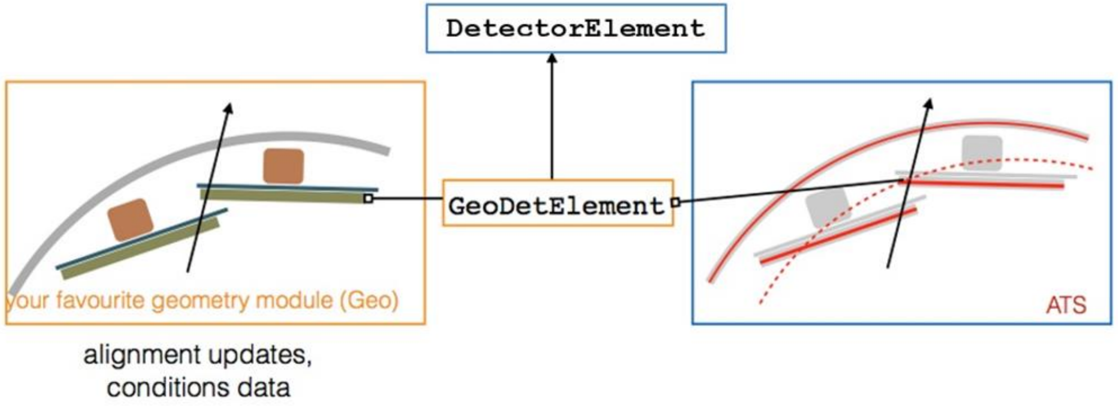


Geometry - Basics

Tracking geometry = Simplified geometry describing sensitive material + approximated material setup

Core of the Geometry: `Surface` class

- represents detector element
  - connection to describing geometry via `DetectorElement`
  - base for measurement and parametrization



- layers extend surfaces
- volumes are enclosed by boundary surfaces

Dynamic vs. fixed size matrices

- ATS is using Eigen as algebra library
- What is the performance penalty when using dynamic instead of fixed size matrices?

1) Eigen-3.2.7, g++ 4.9.2, 1M operations

	Dynamic sized	Fixed size	converted
M(2,5) x v(5)	1.7277s	0.797514s	1.45323s
M(5,2) x M(2,5)	3.53559s	2.67478s	3.48556s

2) Eigen-3.2.7, g++ 4.9.2 –O3, 100M operations

	Dynamic sized	Fixed size	converted
M(2,5) x v(5)	0.154546s	0.00747539s	0.0116945s
M(5,2) x M(2,5)	0.217145s	0.031977s	0.0326164s

=> optimization compiler flags give huge speedup, fixed size operations are a factor 8-20 faster

Optimize Event Data Model

- Use your own parameter definitions: **define plugin**
  - Measurement mapping functions need to be provided
  - Jacobian matrices need to be provided
- Number of parameters variable
- Make use of fixed size vector/matrices whenever possible
- Concrete measurements are of different C++ types
  - Common base class stores concrete measurement in single vector
  - Measurement base class rely on dynamically sized vectors/matrizes
  - Boost::variant keep performance benefit from using fixed size matrix operations while allowing to treat different measurements uniformely