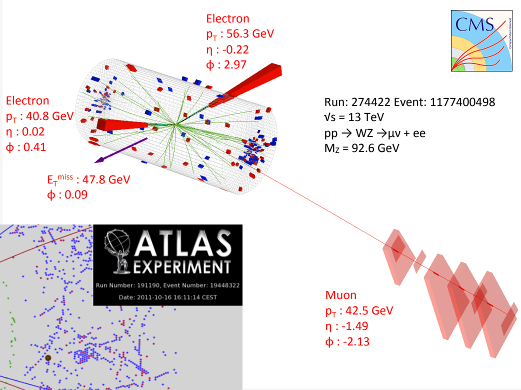
ACTS: from ATLAS software towards a common track reconstruction software

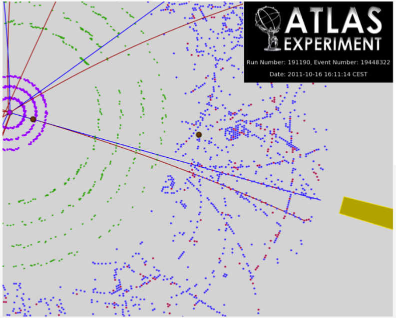
CHEP2016

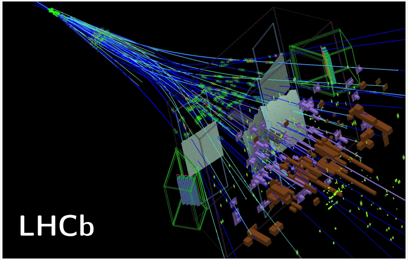
Christian Gumpert
on behalf of the ACTS team

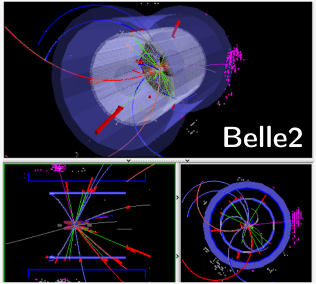
12$^{\text{th}}$ October 2016

Tracking is everywhere





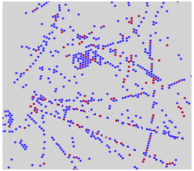




Track reconstruction in a nutshell

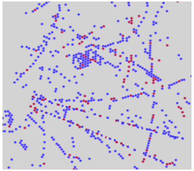
- track = trajectory of (charged) particle through the detector
- hit = energy deposition of particle in the detector
- one possible approach to track reconstruction involves two steps:

Track reconstruction in a nutshell



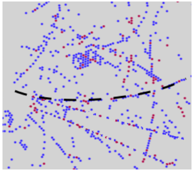
- track = trajectory of (charged) particle through the detector
- hit = energy deposition of particle in the detector
- one possible approach to track reconstruction involves two steps:
  1. track finding
     - find "hits" in the detector which are assumed to belong to one track
     - tailored pattern recognition algorithms to fight large combinatorical background
     - efficient seed finding requires detailed knowledge about the detector

Track reconstruction in a nutshell

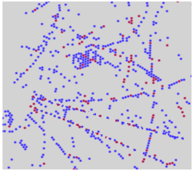


- track = trajectory of (charged) particle through the detector
- hit = energy deposition of particle in the detector
- one possible approach to track reconstruction involves two steps:
  1. track finding
     - find "hits" in the detector which are assumed to belong to one track
     - tailored pattern recognition algorithms to fight large combinatorical background
     - efficient seed finding requires detailed knowledge about the detector
  2. track fitting
     - determine track parameter values from a given set of hits
     - well-known mathematical tools (e.g. χ^2 minimisation, Kalman filter)
     - require knowledge about "propagation" of track through the detector

     $\Rightarrow$ depends on magnetic field and material interactions

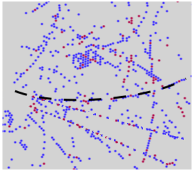


Track reconstruction in a nutshell



- track = trajectory of (charged) particle through the detector
- hit = energy deposition of particle in the detector
- one possible approach to track reconstruction involves two steps:
  1. track finding
     - find "hits" in the detector which are assumed to belong to one track
     - tailored pattern recognition algorithms to fight large combinatorical background
     - efficient seed finding requires detailed knowledge about the detector
  2. track fitting
     - determine track parameter values from a given set of hits
     - well-known mathematical tools (e.g. χ^2 minimisation, Kalman filter)
     - require knowledge about "propagation" of track through the detector

     $\Rightarrow$ depends on magnetic field and material interactions



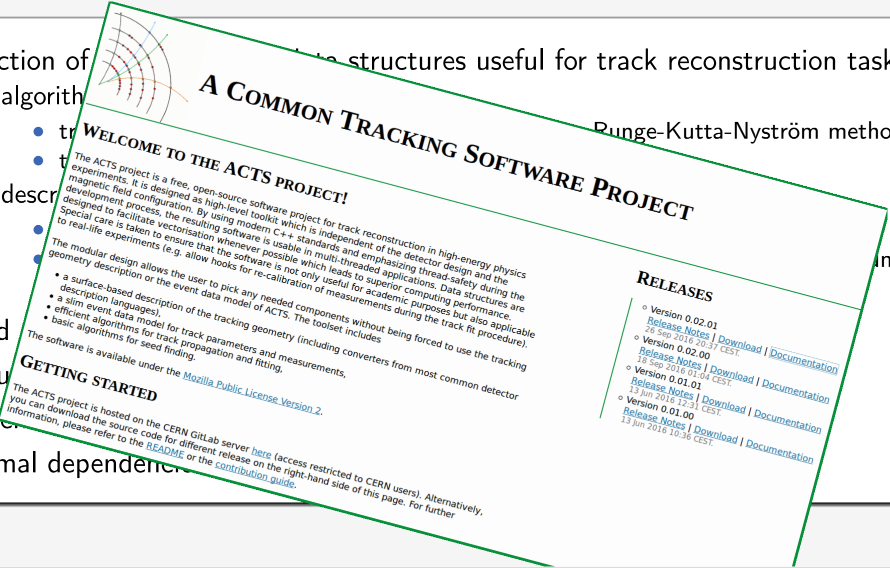
same for all HEP experiments

ACTS: A Common Tracking Software [acts.web.cern.ch/ACTS]

- collection of algorithms and data structures useful for track reconstruction tasks:
  - algorithmic tools:
    - track propagation with covariance matrix (adaptive Runge-Kutta-Nyström method)
    - track fitting (Kalman Filter, χ^2 fitter, Gaussian Sum Filter)
  - descriptive tools:
    - surface-based description of tracking geometry with embedded navigation
    - converters for many common geomerty description languages (e.g. DD4Hep, gdml, TGeo)
    - simple classes for track parameters and measurements
- based on the ATLAS offline tracking software
- modular and customizable design $\Rightarrow$ *"pick and adjust"*
- written in `C++14` using `cmake` as build system
- minimal dependencies: only `boost` and `Eigen`

ACTS: A Common Tracking Software [acts.web.cern.ch/ACTS]

- collection of ... structures useful for track reconstruction tasks:
  - algorith...
    - tr... Runge-Kutta-Nyström method)
    - t...
  - descr...
    - ...
    - ... ml, TGeo)
- based
- modu...
- writt...
- minimal dependen...



A Common Tracking Software Project

Welcome to the ACTS project!

The ACTS project is a free, open-source software project for track reconstruction in high-energy physics experiments. It is designed as high-level toolkit which is independent of the detector design and the magnetic field configuration. By using modern C++ standards and emphasizing thread-safety during the development process, the resulting software is usable in multi-threaded applications. Data structures are designed to facilitate vectorisation whenever possible which leads to superior computing performance. Special care is taken to ensure that the software is not only useful for academic purposes but also applicable to real-life experiments (e.g. allow hooks for re-calibration of measurements during the track fit procedure).

The modular design allows the user to pick any needed components without being forced to use the tracking geometry description or the event data model of ACTS. The toolset includes

- a surface-based description of the tracking geometry (including converters from most common detector description languages),
- a slim event data model for track parameters and measurements,
- efficient algorithms for track propagation and fitting,
- basic algorithms for seed finding.

The software is available under the Mozilla Public License Version 2.

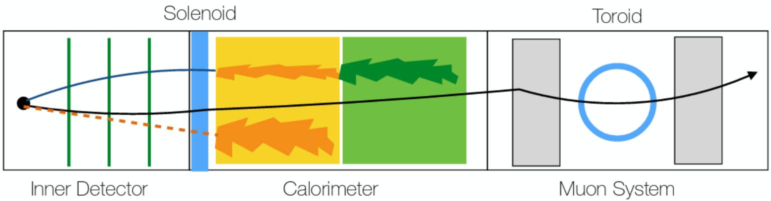
Getting Started

The ACTS project is hosted on the CERN GitLab server here (access restricted to CERN users). Alternatively, you can download the source code for different release on the right-hand side of this page. For further information, please refer to the README or the contribution guide.

Releases

- Version 0.02.01
  Release Notes | Download | Documentation
  26 Sep 2016 20:37 CEST
- Version 0.02.00
  Release Notes | Download | Documentation
  18 Sep 2016 01:04 CEST
- Version 0.01.01
  Release Notes | Download | Documentation
  13 Jun 2016 12:31 CEST
- Version 0.01.00
  Release Notes | Download | Documentation
  13 Jun 2016 10:36 CEST

We do not re-invent the wheel!



- ATLAS offline tracking = optimal starting point for common tracking
- two precision tracking systems having:
  - very different magnetic field setups → field-agnostic parameterisation
  - very different detector technologies → technology-agnostic high-level tracking
  - very different dimensions → re-calibration on demand
  - some lump of material in between → integration of calorimeter into tracking

⇒ ATLAS had to solve the common tracking problem already

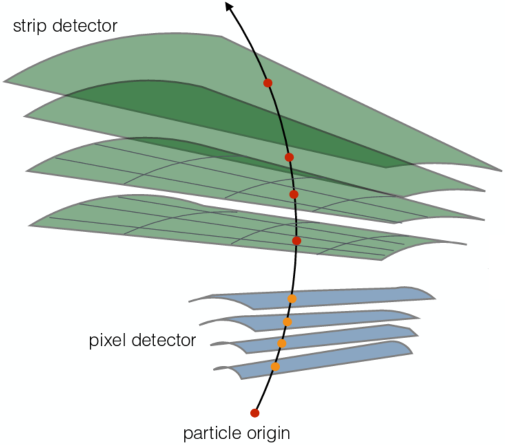
- a lot of experience/expertise went into a major tracking speed-up campaing during LS1 ⇒ highly-optimised code
- uses state-of-the-art mathematical methods (e.g. covariance transport)

Surface-based description of the tracking geometry

- tracking geometry = simplified version of full detector geometry
- most tracking detectors characterized by low material budget and sensitive detector elements at discrete locations

$\Rightarrow$ can be described by a set of surfaces

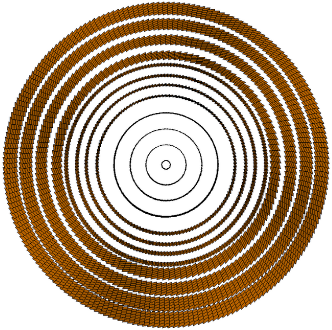
- optimized for fast navigation and propagation through the detector by linking surfaces and volumes
- detector material is mapped onto surfaces
- plugin for creating material maps from Geant4 simulation under developement

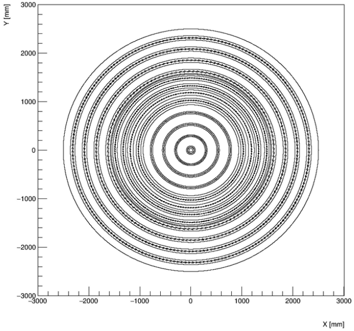


Converters for different geometry description languages

- many different formats for describing your detector geometry available
- ACTS provides plugins to read the most common ones:
  DD4Hep, ROOT's `TGeo*`, gdml and we are happy to add more to the list.

DD4Hep detector model





hit positions from ACTS tracking geometry

Track parameters and measurements

- track parametrized on a surface by five parameters: $(l_0, l_1, \phi, \theta, q/p)$
- measurements described in local 2D coordinate system of surface

$\Rightarrow$ algorithms (e.g. Kalman fitter) work on small-sized vectors/matrices

```
class TrackParameters
{
  Eigen::Matrix<double, 5, 1> m_parameters;
  Eigen::Matrix<double, 5, 5> m_covariance;

  ...
};
```

fixed-sized matrices

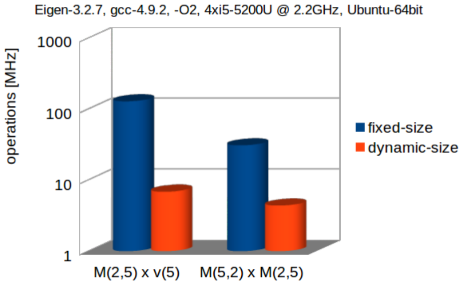
```
class MeasurementBase
{
  virtual Eigen::Matrix<double, Eigen::Dynamic, 1>
  get_parameters() const = 0;

  virtual Eigen::Matrix<double, Eigen::Dynamic, Eigen::Dynamic>
  get_covariance() const = 0;

  ...
};
```

dynamically-sized matrices

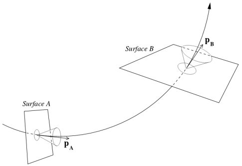
depends on actual type of the measurement



- large performance difference between fixed- and dynamic-sized matrices for small dimensions in `Eigen`
- ACTS uses a variant-based approach to handle measurements which allows to always use fixed-size matrix operations

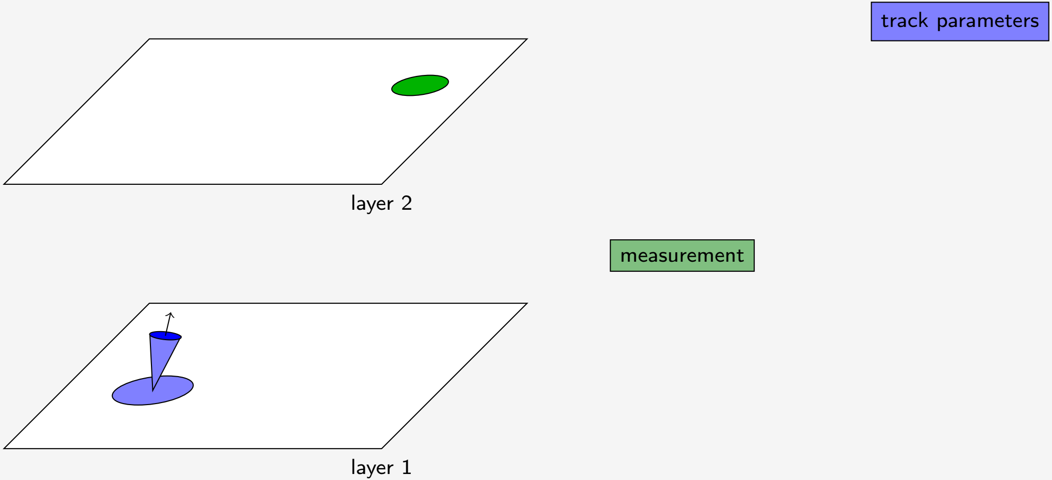
Design of the track propagator interface

- fundamental task of track propagator: solve equations of motion for particle in inhomogenious magnetic field and possibly with material interactions: $\vec{x} \to \vec{x}'$
- CPU bottleneck: transport of covariance matrix of track parameters requires knowledge of jacobian $\mathbf{J}_{ij} = \frac{dx'_i}{dx_j} \Rightarrow \mathbf{C}' = \mathbf{J}\mathbf{C}\mathbf{J}^{\mathrm{T}}$

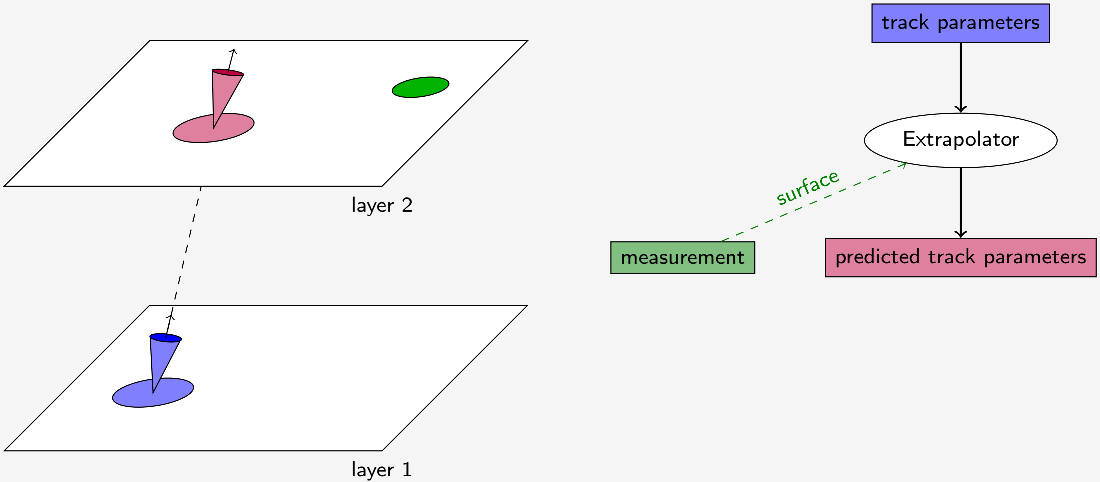


- design considerations:
  - choose optimal parameterization for solving equations of motions
  - exploit special detector properties (e.g. symmetries, field-free regions)
  - specialize calculations for your track parameters (e.g. to benefit from vectorization)
  - extended functionality, not only propagate track parameters but also gather information (e.g. material)
  - support custom stopping conditions
- ACTS allows you to plugin in your specialized/optimized code
- default: adaptive Runge-Kutta propagator with Nyström method used in ATLAS

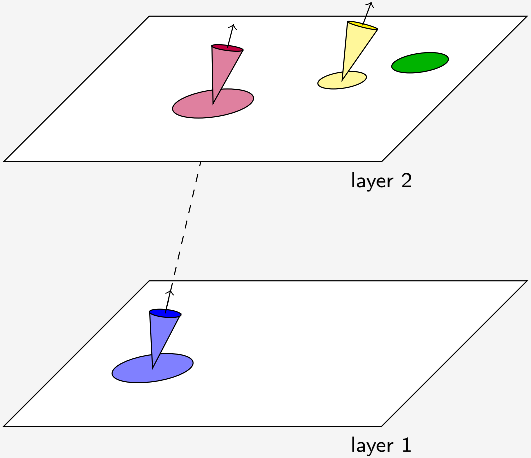
Track fitting: Kalman filter – simple approach

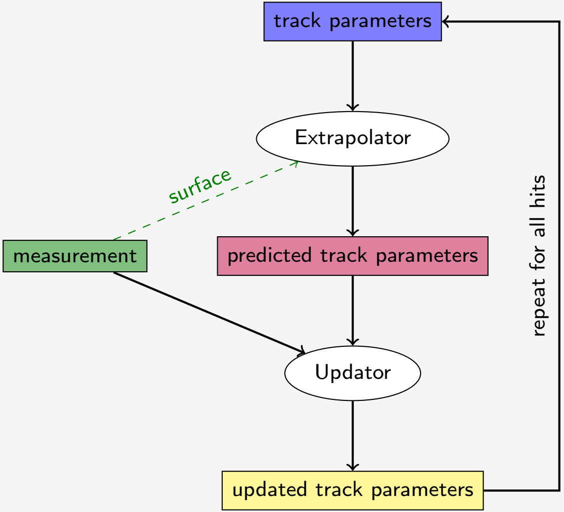


Track fitting: Kalman filter – simple approach

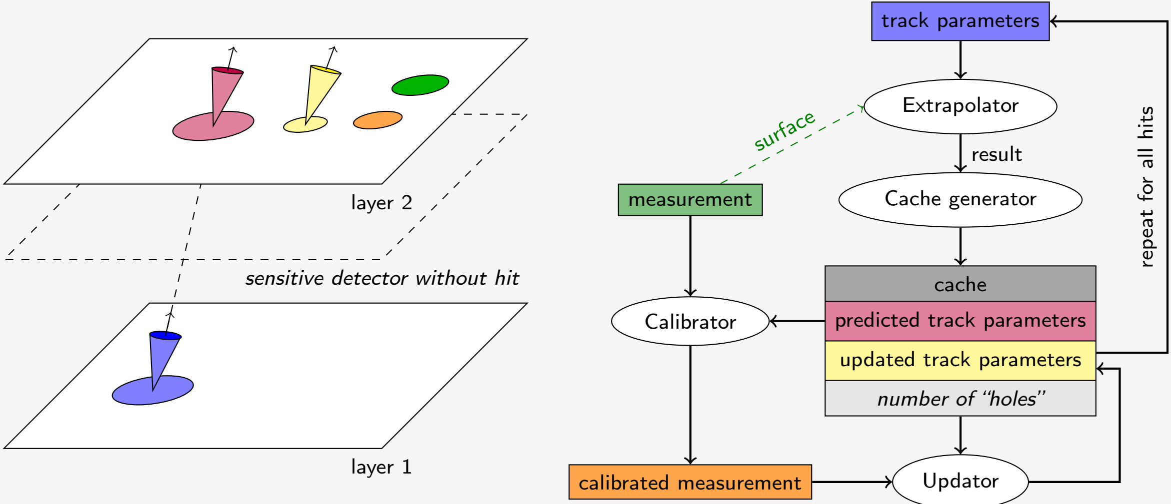


Track fitting: Kalman filter – simple approach



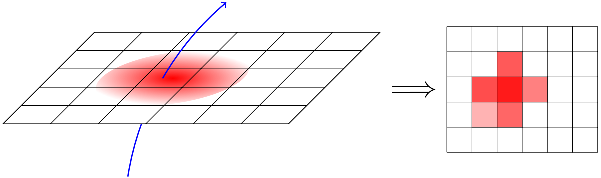


Track fitting: Kalman filter – ACTS implementation



Further features and future plans

- simple, helix-based seed finding algorithm for barrel-endcap type detector layouts
- digitization module to generate hit clusters from hit positions using sensor segmentation



$\Rightarrow$ ACTS can be used as fast track simulator

- extend list of track fitters to include χ^2-based track fitter and Gaussian Sum Filter
- provide vectorized versions for propagator and fitter to operate on "packs" of tracks

Summary

- ACTS is an open-source and framework-independent track reconstruction tool set based on the ATLAS offline tracking software which delivered outstanding performance

⇒ benefits from more than seven years of experience

- design is focused on excellent computing performance with the provided event data model while maintaining the possibility to adapt to your own code
- future developments include tools for seed finding, specialised multi-track propagators using vectorization, and more track fitters
- ACTS is actively developed, your feedback and contribution is highly welcome
- if interested, sign up on acts-users -at- cern.ch,
  have a look at our webpage [acts.web.cern.ch/ACTS] or
  browse the code on GitLab