SPONSORED BY THE

Federal Ministry of Education and Research

EP-R&D
Programme on Technologies for Future Experiments

FSP ATLAS
Erforschung von Universum und Materie

23rd Gentner Day

Detray - Heterogeneous Tracking Geometry Description and Navigation

Joana Niermann[1,2]

supervised by Andreas Salzburger[1], Stan Lai[2]

on behalf of the `detray` developers (Beomki Yeo[3,4], Attila Krasznahorkay[1], Stephen Swatman[1,5])

26.04.2023

[1] CERN
[2] II. Physikalisches Institut, Georg-August-Universität Göttingen
[3] Department of Physics, University of California
[4] Lawrence Berkeley National Laboratory
[5] University of Amsterdam, Amsterdam, The Netherlands

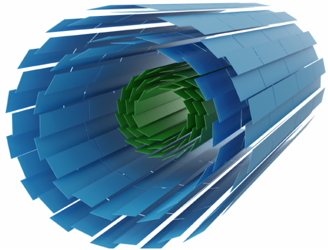
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

CERN

Motivation

ACTS - A Common Tracking Software

- Efficient, thread-safe `C++` implementation of track reconstruction tools.
- Detector agnostic *tracking geometry* description.
- Investigated by e.g. ATLAS, FASER, sPHENIX



Bringing Tracking Software to GPUs

- Different GPU backends, e.g. CUDA or SYCL
- Polymorphic geometry cannot be used in kernels
- Vector-of-vector data structures difficult to move to device memory system

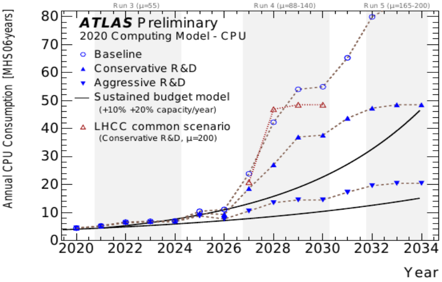
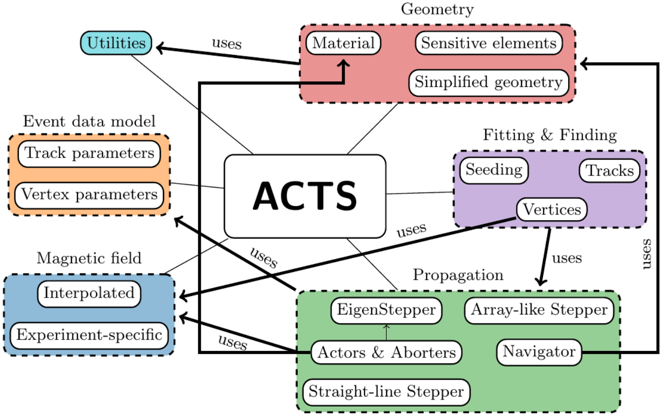


Image: sPHENIX silicon tracker in ACTS (top)

ACTS R&D Project

- **traccc:** Main GPU demonstrator: Event data model, Fitting & Finding
- **covfie:** Vector field library, used for Magnetic Field
- **detray:** Geometry, Propagation
- **vecmem:** Memory management between host and device
- **algebra-plugins:** Switch linear algebra implementation



Source: code available at https://github.com/acts-project/

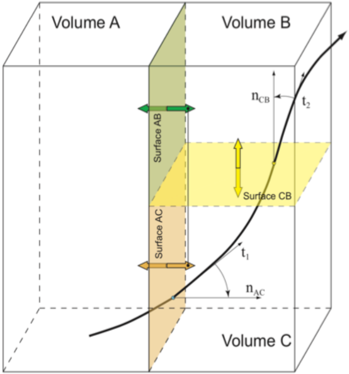
The detray Project

Tracking Geometry Building Blocks

- **Volumes** subdivide the detector into smaller navigation regions.
- **Surfaces** as core building blocks.
- A **Grid** as volume and surface finder accelerator.
- Read tracking geometry from ACTS

Design Goals

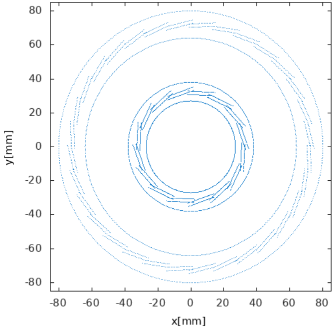
- Use classes on host (CPU) and device (GPU)
- Geometry classes without runtime polymorphism
- Flat container structure, using `vecmem` library
- Index based data linking, no pointers



resulting navigation
through the boundary portals

`detray` Detector Class

- Holds all geometry and magnetic field data
- Performs the container moves between host and device
- Provides interface to the tracking geometry data
- Testbed geometry modeled after pixel component of ACTS *generic detector*.



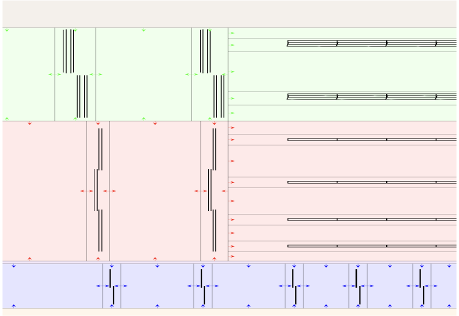
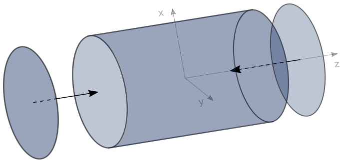
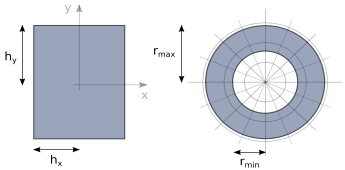


Image: The ACTS Open Data Detector implementation https://github.com/acts-project/acts/pull/1039 (right)

Geometry Description

- **Volumes**: defined by their boundary surfaces
- **Surfaces**: Placed by affine transformations and defined by boundary masks
- **Masks**: Defined by a shape type.
  Specify local coordinates and extent of surfaces.
- **Portals**: Special surfaces that tie volumes together through index links.
- **Material:** Homogeneous *slabs* or *rods* of parametrized material. Many predefined materials available.





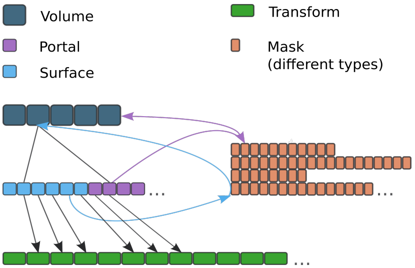
No abstract classes: Every type needs its own container. Solved by compile-time *unrolling* of tuple containers.

`detray` Container Structure

In ACTS: (For now) Jagged memory layout of volumes containing layers, containing surfaces.

Linking by Index

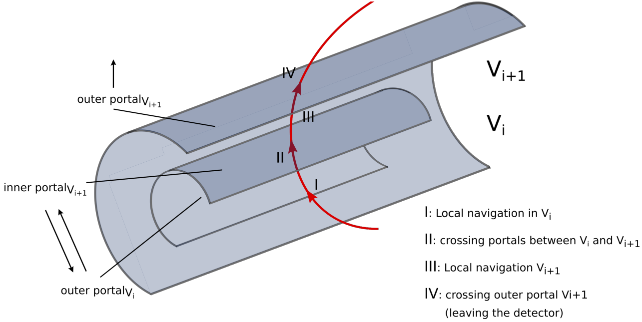
- Volume (descriptors) keep links into surface container (acceleration data structures).
- Surface (descriptors) keep indices into the transform, mask and material containers.
- Portal masks link to adjacent volume.
- Sensitive/Passive surface masks link back to mother volume.



The geometry data structures are built host-side and the memory allocation strategy is determined by *vecmem memory resources*.

Participants

- **Propagator:** runs the propagation loop: Calls stepper, navigator and the actors.
- **Navigator:** Moves between detector volumes and finds distance to next candidate surface.
- **Stepper:** Transports the track parameters and corresponding covariance matrix through magnetic field.
- **Actors/Aborters:** Extend propagation with various functionality (e.g. watch termination criteria).



Geometry Navigation

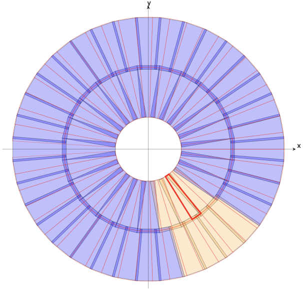
Surface Candidate Cache

- *Trust levels* determine update method:
- ***Full trust***: Track state still consistent, do nothing.
- ***High trust***: Only update the current next target surface.
- ***Fair trust***: Update all candidate surfaces and sort again.
- ***No trust***: (Re-)initialize current volume, i.e. fill cache from local neighborhood and sort.

⇒ Stepper/actors can lower trust level to trigger navigation update.

Local Navigation in a Volume

- Accelerator data structures provide surface neighbourhood lookups.
- Navigate local neighborhood, before reassuming inter-volume navigation.
- In principle: Any kind of accelerator data structure possible.



Propagation Loop

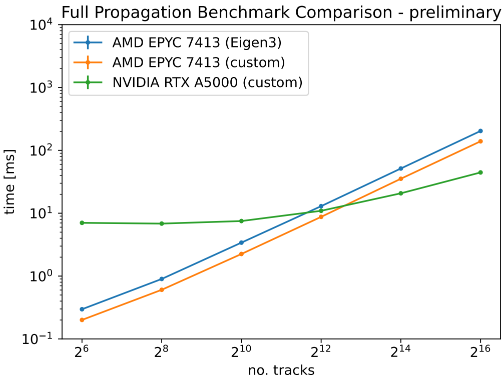
```
// initialize the propagation
navigator.init(propagation);

run_actors(propagation._actor_states,
           propagation);

// run while propagation has hearbeat
while (propagation.heartbeat) {

  stepper.step(propagation);

  navigator.update(propagation);

  run_actors(propagation.actor_states,
             propagation);

  navigator.update(propagation);
}
```

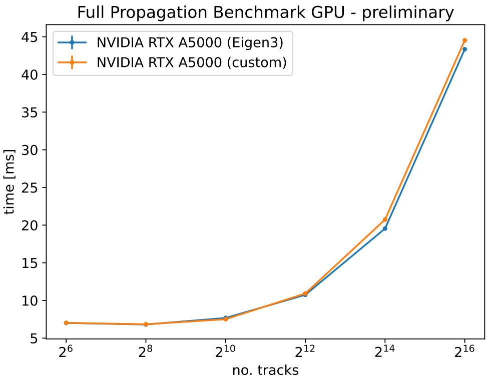
Actor Mechanism

- E.g. *aborters, material interactor, random scatterer* . . .
- Can be plugged in at compile time.
- Perform various tasks in every step
- Possible to *observe* other actors ⇒ call tree

⇒ Schedules one track per thread, currently.

Preliminary Benchmarks



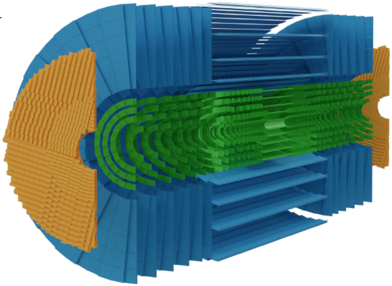


Source: tag v0.29.0 (https://github.com/acts-project/detray)

Summary and Outlook

Status

- Testbed geometry modelled after ACTS generic detector's pixel detector
- Uses COVFIE library for inhomogeneous B-field description (WIP)
- Adaptive Runge-Kutta-Nyström algorithm for field integration
- Transport of track parametrization and covariance through (in-)homogeneous B-field
- Simple material description with material interactions



Major on-going Developments

- Integration of navigation accelerator data structures (grid collections are available on device, but not yet used)
- Read existing tracking geometry implementations from ACTS (e.g. ATLAS ITk)

Acknowledgements

This work benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments (CERN-OPEN-2018-006)[1].

This work has been sponsored by the Wolfgang Gentner Programme of the German Federal Ministry of Education and Research (grant no. 05E18CHA).

This work was funded by the NSF under Cooperative Agreement OAC-1836650.

References

[1] M. Aleksa *et al.*, "Strategic R&D Programme on Technologies for Future Experiments," CERN, Geneva, Tech. Rep., Dec. 2018, CERN-OPEN-2018-006. [Online]. Available: https://cds.cern.ch/record/2649646.

[2] *CUDA C++ Programming Guide*, Oct. 2022. [Online]. Available: https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html (visited on 10/24/2022).

[3] J. Nickolls *et al.*, "Scalable Parallel Programming with CUDA: Is CUDA the Parallel Programming Model That Application Developers Have Been Waiting For?" *Queue*, vol. 6, no. 2, pp. 40–53, Mar. 2008. DOI: 10.1145/1365490.1365500. [Online]. Available: https://doi.org/10.1145/1365490.1365500.

[4] N. Bell *et al.*, "Thrust: A productivity-oriented library for cuda," in *GPU Computing Gems Jade Edition*, ser. Applications of GPU Computing Series, Boston: Morgan Kaufmann, 2012, pp. 359–371. DOI: https://doi.org/10.1016/B978-0-12-385963-1.00026-5.

[5] J. Myrheim *et al.*, "A fast runge-kutta method for fitting tracks in a magnetic field," *Nucl. Instrum. Methods*, vol. 160, no. 1, pp. 43–48, 1979. DOI: https://doi.org/10.1016/0029-554X(79)90163-0.

[6] L. Bugge and J. Myrheim, "Tracking and track fitting," *Nuclear Instruments and Methods*, vol. 179, no. 2, pp. 365–381, 1981, ISSN: 0029-554X. DOI: https://doi.org/10.1016/0029-554X(81)90063-X.

[7] E. Lund, L. Bugge, I. Gavrilenko, *et al.*, "Track parameter propagation through the application of a new adaptive runge-kutta-nyström method in the atlas experiment," *Journal of Instrumentation*, vol. 4, no. 04, P04001, Apr. 2009. DOI: 10.1088/1748-0221/4/04/P04001.

[8] C. Allaire *et al.*, *OpenDataDetector*, gitlab, version v1, 2021. DOI: 10.5281/zenodo.4674402. [Online]. Available: https://gitlab.cern.ch/acts/OpenDataDetector/.

Backup

Heterogeneous Computing Model

Implementation in `detray`

- Goal: outsource many-track propagation to device.
- Need to handle host-device memory transfers.
- Core classes templated on `STL` vs. vecmem containers.
- The geometry data structures are built host side and memory allocation strategy is determined by *vecmem memory resources*.

```
#include <vecmem/containers/vector.hpp>

// Transform store using managed memory
vecmem::cuda::managed_memory_resource mng_mr;

// Build with host vector type
transform_store<vecmem::vector> store(mng_mr);

// Get store view object
auto sv = detray::get_data(store);

// Run the kernel
test_kernel<<<block_dim, thread_dim>>>(sv);
```

```
#include <vecmem/containers/device_vector.hpp>

// Kernel-side construction
__global__ void test_kernel(store_view sv) {
  // Build with device vector type
  transform_store<vecmem::device_vector> store(sv);

  // Do something
}
```

Parameter and Error Propagation

Track state parametrization: global $(x, y, z, t, v_x, v_y, v_z, q/p)$, local $(\mathrm{loc}_0, \mathrm{loc}_1, \varphi, \theta, q/p, t)$

Field Integration

- No track solution in closed form inhomogeneous magnetic field.
- Numeric Integration: Runge-Kutta-Nyström algorithm (4-th order).
- Takes distance to next target surface and adjusts step-size according to integration error.
- Magnetic field map interpolation to get field vectors at arbitrary positions (covfie library).

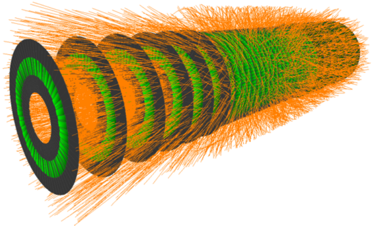
Covariance Transport and Material Interaction

- Do covariance transport: Transform initial covariance estimate with coordinate transform/RK-transport Jacobians.
- Called at every material surface to add material effects to covariance.
- Takes energy loss effects and multiple scattering into account.

$\Rightarrow$ Both implemented as actors.

Kaggle TrackML challenge

- Generic Tracking Detector design
- Reduced physics list in fastsim
- But: afterwards dataset was used for further tracking R&D



⇒ Provide simplified generic, but more realistic dataset!

Next level: The Open Data Detector [8]

- More realistic detector description
- 4 layer Pixel detector
- Short- and Long-Strip detector
- Detector mounting, cables, cooling . . .

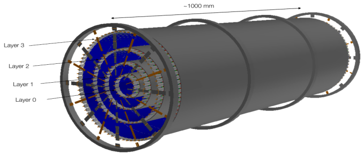


Image: (top) https://sites.google.com/site/trackmlparticle/