# Integration of the ACTS track reconstruction toolkit in the ATLAS software for HL-LHC operations

*Carlo* Varni[1,*], on behalf of the ATLAS Collaboration

[1]University of California at Berkeley

**Abstract.** In view of the High-Luminosity LHC era the ATLAS experiment is carrying out an upgrade campaign which foresees the installation of a new all-silicon Inner Tracker (ITk) and the modernization of the reconstruction software. Track reconstruction will be pushed to its limits by the increased number of proton-proton collisions per bunch-crossing and the number of read-out channels of the ITk detector. In order to remain within CPU budgets while retaining high physics performance, the ATLAS Collaboration plans to use A Common Tracking Software (ACTS), an experiment-independent toolkit for track reconstruction. The migration to ACTS involves the redesign of the track reconstruction components as well as the ATLAS Event Data Model (EDM), aiming at a thread-safe and maintainable software. In this contribution, the current status of the ACTS integration for the ATLAS ITk track reconstruction is presented, with emphasis on the improvements of the track reconstruction software and the implementation of the ATLAS EDM.

## 1 Introduction

The High Luminosity Large Hadron Collider (HL-LHC) is expected to deliver luminosity of $5.0 - 7.5 \times 10^{34}$ cm$^{-2}$s$^{-1}$ and an average of up to 200 simultaneous proton-proton (pp) interactions per bunch crossing (pile-up or $\langle\mu\rangle$). To cope with these challenging running conditions, the ATLAS experiment [1] engaged in extensive Phase-II detector and software upgrade programmes [2] that foresee the installation of a new all-silicon Inner Tracker (ITk) [3] and a complete redesign of its track reconstruction software, to meet the tracking performance requirements for the ATLAS physics programme while remaining within the CPU budget.

The new ITk detector will provide a pseudo-rapidity[1] coverage up to $|\eta| = 4.0$ and it will consist of an inner pixel and an outer strip system, as shown in Figure 1: the pixels to provide precision measurements as close as possible to the interaction point; the strips to achieve accurate tracking at larger radii. The ITk detector has been designed to record a minimum of nine measurements per track and aims at reconstructing charge particle trajectories with $p_{\mathrm{T}} > 1$ GeV in the entire pseudo-rapidity coverage.

---

[*] e-mail: carlo.varni@cern.ch

[1] ATLAS uses a right-handed coordinate system with its origin at the nominal interaction point (IP) in the centre of the detector and the $z$-axis along the beam pipe. The $x$-axis points from the IP to the centre of the LHC ring, and the $y$-axis points upwards. Polar coordinates $(r, \phi)$ are used in the transverse plane, $\phi$ being the azimuthal angle around the $z$-axis. The pseudo-rapidity is defined in terms of the polar angle $\theta$ as $\eta = -\ln\tan(\theta/2)$ and is equal to the rapidity $y = \frac{1}{2}\ln\left(\frac{E+p_z}{E-p_z}\right)$ in the relativistic limit. Angular distance is measured in units of $\Delta R \equiv \sqrt{(\Delta y)^2 + (\Delta\phi)^2}$.

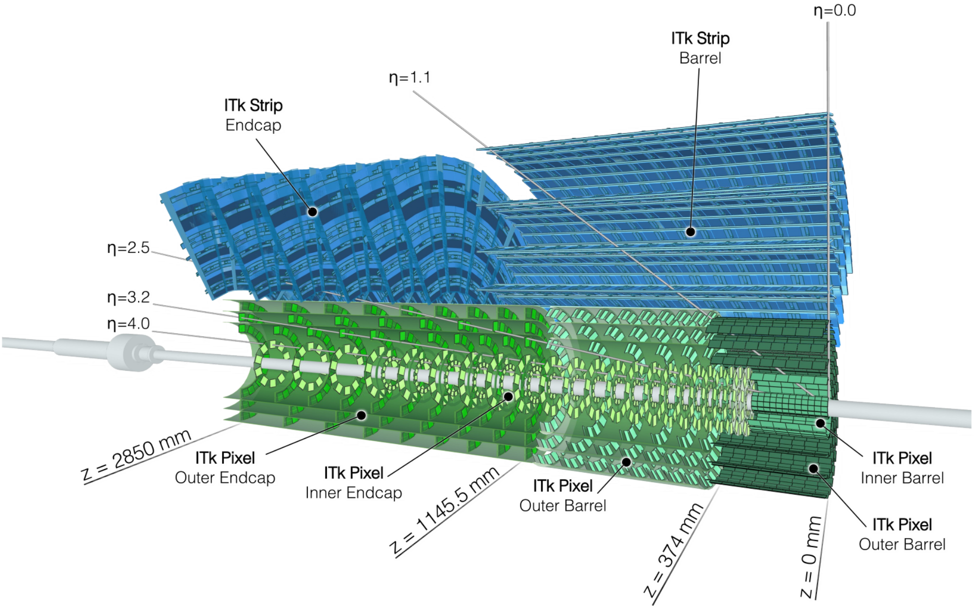

**Figure 1.** Cutaway view of layout of one end of the ITk detector [4] to be installed in ATLAS. In green the inner pixel system, in blue the outer strip system. Specific subcomponents of these systems are also highlighted.

Track reconstruction constitutes the most computationally intensive element of event reconstruction due to its combinatorial nature, currently representing 40% of the event reconstruction CPU budget at $\langle\mu\rangle \sim 50$ [5]. Its complexity scales with the number of charged particles, which is determined by the number of simultaneous interactions per event in pp collisions: the number of tracks scales linearly with pile-up, while the execution time scales approximately quadratically. In addition to that, the ITk detector design will substantially increase both the pseudo-rapidity acceptance and the number of read-out channels with respect to the current ATLAS inner detector. Under these conditions, the required CPU resources for event reconstruction will increase substantially and exceed the available computing budget by a factor of two [6]. This motivated the ATLAS Collaboration to upgrade its software by adopting A Common Tracking Software (ACTS) [7] for track reconstruction.

This documents presents a brief overview of the ACTS project in section 2, followed by a short description of how ATLAS performs track reconstruction in section 3. Finally, a description of the integration of ACTS into the ATLAS reconstruction software is provided in section 4, with a particular focus on what the integration required. A comparison of the track reconstruction performance on the ITk detector, both in terms of physics results and CPU time, between the current ATLAS track reconstruction software and a modified version of it using ACTS will be presented. In the rest of this document, these will be referred to as ATLAS and ACTS reconstructions respectively. Since the integration effort is still in progress, this document will show the current state-of-the-art as of the end of 2024.

## 2 A Common Tracking Software

ACTS is a generic, framework- and experiment-independent toolkit for charged particle trajectory reconstruction, resulting from the collaboration of international developers from different high energy physics (HEP) experiments. This open-source project, written in modern C++ 20 standards, started at CERN in 2016 and is currently being explored as a software solution by several experiments, such as sPHENIX [8–10], Lohengrin [11], FASER [12], ATLAS [13] and many others. The aim of the ACTS project is to provide production-ready implementations of state-of-the-art tracking methods and to serve as an algorithmic testbed, supporting Machine Learning methods and computing accelerators. The ACTS code has been designed to be thread-safe, customizable, extendable and agnostic to detector technologies, designs and frameworks. Particular care has been spent on CPU optimization, both in terms of data structures and algorithm designs.

## 3 Track Reconstruction in ATLAS

Track reconstruction with the ITk detector [5] proceeds by sequential steps. The first one consists of grouping signals from adjacent read-out channels in both the pixel and strip detectors. These grouped signals, referred to as clusters, are then converted into three-dimensional space representations – space points – in the global reference frame of the detector. For the pixel detector one cluster is used for local-to-global transformation; for the strip detector two clusters are necessary. A subsequent step identifies triplets of space points that are compatible with a helix trajectory, called track seeds. An iterative Combinatorial Kalman Filter (CKF) [14] then extends these seeds into track candidates by collecting all the clusters in the detector that are compatible with the estimated trajectory, basing the decision on a $\chi^2$ criterion. The final step is represented by the ambiguity solver, which resolves the overlaps between the resulting track candidates, rejects the low-quality trajectories and performs a track re-fit procedure to extract the final track parameters. These steps constitute a tracking pass.

The complete track reconstruction in ATLAS is composed of multiple tracking passes, each pass tuned for reconstructing specific topologies of tracks. The primary pass aims at reconstructing tracks stemming from the primary interaction, while secondary passes are optimized to increase the acceptance to other particles: tracks resulting from photon conversion; tracks stemming from displaced vertices; or tracks with low $p_\mathrm{T}$. The tracking passes are run sequentially, and each pass uses detector hits that have not been assigned to any track reconstructed by previous passes. The reconstruction algorithms provided by ACTS are flexible enough to be tuned for all these use cases. While the integration of the ACTS toolkit in the ATLAS software applies to all these tracking passes, the results presented by this document pertain only to the primary pass.

## 4 ACTS integration into the ATLAS software

Extensive work is required to fully integrate ACTS into an experiment reconstruction software. In ATLAS, a new Event Data Model (EDM) [15] has been defined to connect ATLAS and ACTS representations of the core objects; an ACTS-based geometry description of the ITk detector has been implemented; and reconstruction algorithms have been restructured to use ACTS. Changes to geometry and tracking algorithms also have to be implemented in muon reconstruction. To validate the integration of each ACTS component, dedicated track reconstruction workflows have been defined that foresee an ATLAS reconstruction sequence with only one or some of its components modified to use ACTS. This assures a direct comparison between the legacy ATLAS and ACTS versions of the same algorithms.

ACTS periodically provides stable releases of the software to be deployed to the experiments. These releases may include new features, code refactoring, as well as bug fixes and may require the experiment software to be adapted in case of changes to the public API. To avoid the creation of ACTS releases incompatible with ATLAS, a solid testing infrastructure has been put in place, which tests every single commit in ACTS to ensure the experiment code always works properly. Since updates on both ACTS and ATLAS code may be necessary in the development phase, the continuous iteration between the experiment developers and the ACTS core software team is a key factor in the successful integration effort, which has greatly gained by the fact that the majority of the ACTS developers are also ATLAS members. Having developers working in both environments assures a quick development cycle, with an efficient process of developing and integrating features necessary to meet ATLAS-specific requirements.

### 4.1 Event Data Model

ACTS possesses an internal representation of the core objects that are used by its routines and that are coherently propagated through all its reconstruction components. At the same time, the HEP experiments have their own representations of the same objects: for the ATLAS experiment, the xAOD data format is widely used. ACTS makes no assumptions about the experiment's EDM model, nor does it impose a specific representation, but it provides a public EDM to connect the two. This public EDM is responsible for instructing ACTS on how to interact with the customer EDM by requiring the experiment to implement dedicated classes that satisfy a specific API, but whose internal implementation is up to the user. Figure 2 shows a conceptual scheme of the EDM model adopted. This structure allows the ACTS routine to keep working on the internal ACTS EDM, while retrieving the required information from the experiment EDM. Further details can be found in [16].

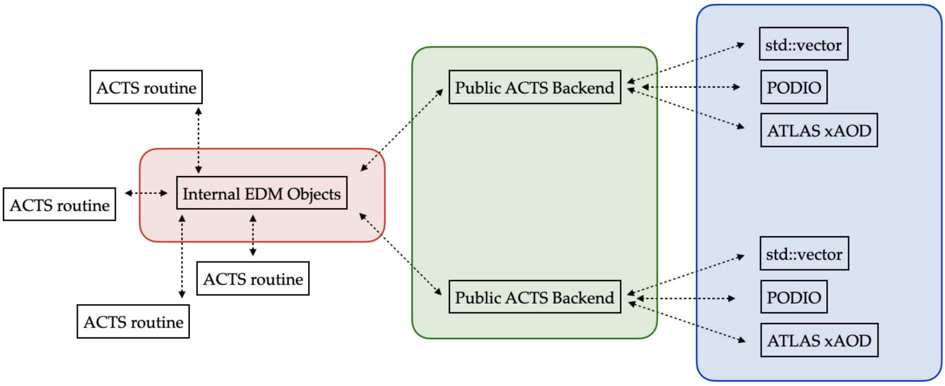

**Figure 2.** Design of the EDM model, which comprises three elements: an internal ACTS EDM (left red box), an experiment-specific EDM (right blue box) and a public EDM that implements a well-defined API (middle green box) and that instructs ACTS how to interact with the experiment representations of the core objects.

### 4.2 Track Reconstruction Algorithms

ACTS has been successfully integrated in all the ATLAS track reconstruction algorithms. The focus is now on tuning their performance.

For the cluster creation component, ACTS deploys an implementation based on the Hoshen–Kopelman algorithm [17]. Given how the problem is formulated – i.e. grouping signals from adjacent read-out channels – only one solution is possible for an event with a given hit content. As such, the comparison with the ATLAS counterpart in both the pixel and strip systems produces a 100% agreement as shown in Figure 3, which compares the resulting clusters obtained by processing the same events.

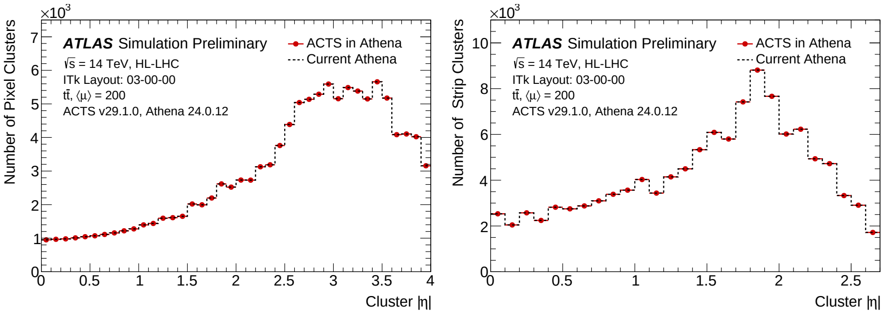

**Figure 3.** Average number of reconstructed pixel (left) and strip (right) clusters per event as a function of the cluster pseudo-rapidity. These quantities have been computed running on the same set of events using the legacy ATLAS software and a modified version of it using the ACTS toolkit. Identical physics results are obtained.

For triplet finding, ACTS provides a re-implementation of the same algorithm used by ATLAS, relying on geometric considerations to first find doublets of space points, and then build triplets by iterating over all the possible doublet combinations. Since the implementations have the same logic, the two algorithms produce the same outputs, provided their configuration is the same. This is shown in Figure 4, where the seeding efficiencies obtained by the ATLAS and ACTS reconstructions are shown to be the same. Efficiency is defined as the ratio between the number of reconstructed seeds matched to truth particles and the total number of selected truth particles. A seed is matched to a truth particle if the percentage of its measurements belonging to that same truth particle is more than 50%. Other alternative algorithms are available in ACTS, but not covered in this document [18].

For the CKF, ACTS adopts a different strategy with respect to the existing one in ATLAS. ACTS has a branching mechanism in case multiple compatible clusters are found on the same detector element, and it will produce tracks with a good quality, as it performs a full Kalman Fit with the correct treatment of interactions with the full detector material and magnetic field maps. This removes the need to perform a downstream track re-fit procedure to extract the final track parameters, as currently done in ATLAS. Consequently, the ACTS ambiguity resolution stage will only focus on removing overlaps and reduce the track duplicate rate. This is expected to provide a major CPU time reduction with respect to the legacy ATLAS software.

Figure 5 shows the track reconstruction efficiency obtained with the ACTS CKF implementation. Efficiency is defined as the ratio between the number of reconstructed tracks matched to truth particles and all selected truth particles. A reconstructed track is matched to

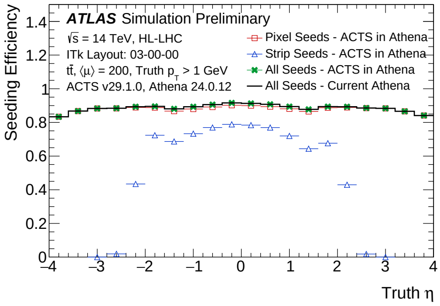

**Figure 4.** Efficiency of the triplet finding as a function of the pseudo-rapidity of the associated truth particle. Results are obtained running on the same set of events using the current ATLAS software and a modified version of it using the ACTS toolkit. Identical results are obtained. For ACTS, seeding efficiency is also shown separately for pixels and strips.

a truth particle if the matching probability is larger than 50%, where the matching probability is provided by the fraction of measurements associated to the same truth particle [3].

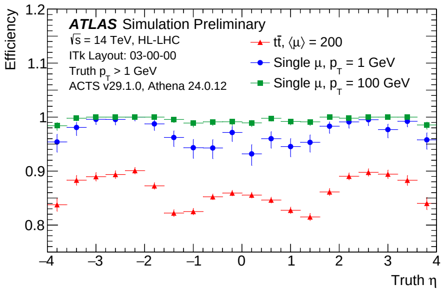

**Figure 5.** Tracking efficiency as a function of the pseudo-rapidity of the associated truth particle: semileptonic $t\bar{t}$ events at $\langle\mu\rangle = 200$; and for single muons with $p_\mathrm{T}$ of 1 GeV and 10 GeV without pile-up. Results are obtained running a version of the ATLAS software using ACTS.

### 4.3 CPU optimization

The on-going optimization campaign focuses on reducing the CPU time consumption of the ACTS reconstruction algorithms. The aim of this effort is to achieve faster reconstruction algorithms, while retaining compatible physics performance with the current ATLAS tracking. The current effort is focused on the track finding step (triplet finding and CKF) and Figure 6 shows the single-thread CPU time evolution when reconstructing the same set of events with an ACTS reconstruction sequence relative to the ATLAS reconstruction sequence. The figure shows that a reduction in CPU time of a factor of 5.4 has been achieved by a long list of improvements. These improvements are the results of several changes: new features in ACTS, EDM design, algorithm logic, and algorithm configurations. Details of the entries in the plot can be found in [19].

A solid monitoring infrastructure is in place to check the CPU time impact of any commit in the ACTS software. Additional investigations into how to improve the ACTS track reconstruction deployed in ATLAS will also be performed in 2025, with focus on both algorithm implementation and configuration.

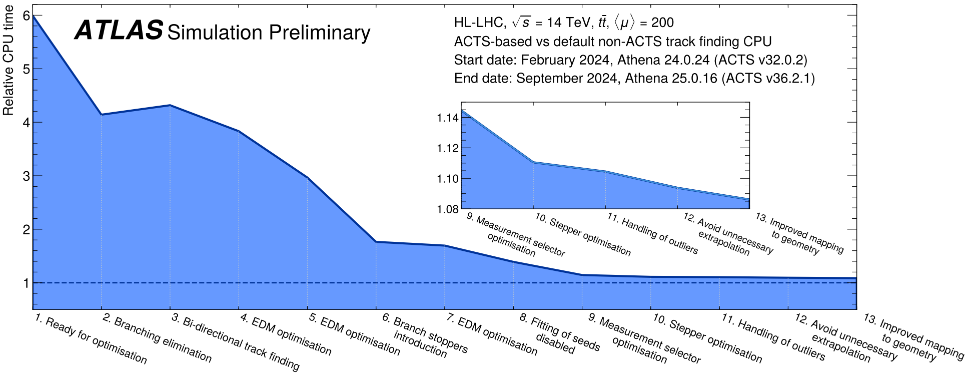

**Figure 6.** Incremental decrease of the CPU time of the ACTS reconstruction sequence, following a long list of developments in 2024. The plot only accounts for track finding (triplet seeding and CKF), and it is relative to the current ATLAS counterparts.

## 5 Conclusions

The development and deployment of an efficient and maintainable track reconstruction software is a key component for future HEP experiments. Within the ATLAS experiment, the ACTS toolkit has been introduced to provide high-level track reconstruction tools. The current status of the integration of ACTS in the ATLAS ITk track reconstruction software has been presented, and the magnitude of this enterprise has been shown. For every component of the track reconstruction sequence, the ACTS implementation has been described and compared with its ATLAS counterpart. Now that initial studies have shown that ACTS obtains identical or comparable physics results, the next goal is to increase its speed. Significant improvements in CPU consumption obtained from an initial effort in 2024 have been described, and this optimization campaign will continue in 2025, with the ultimate goal of having ACTS-based software provide the same or better results as the current ATLAS tracking software, but much faster.

## Acknowledgements

This work was supported by the National Science Foundation under Cooperative Agreements OAC-1836650 and PHY-2323298 and the U.S. Department of Energy's Office of Science, Office of High Energy Physics, under Contract No. DE-AC02-05CH11231.

## References

[1] ATLAS Collaboration, *The ATLAS experiment at the CERN Large Hadron Collider*, Journal of Instrumentation **3**, S08003 (2008), https://dx.doi.org/10.1088/1748-0221/3/08/S08003

[2] ATLAS Collaboration, *ATLAS Phase-II Upgrade Scoping Document*, CERN-LHCC-2015-020, LHCC-G-166 (2015), https://cds.cern.ch/record/2055248

[3] ATLAS Collaboration, *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*, CERN-LHCC-2017-021, ATLAS-TDR-030 (2017), https://dx.doi.org/10.17181/CERN.FOZZ.ZP3Q

[4] ATLAS Collaboration, *Expected Tracking Performance of the ATLAS Inner Tracker at the High-Luminosity LHC* (2024), https://arxiv.org/abs/2412.15090

[5] ATLAS Collaboration, *Software Performance of the ATLAS Track Reconstruction for LHC Run 3*, Computing and Software for Big Science **8**, 9 (2024), https://doi.org/10.1007/s41781-023-00111-y

[6] ATLAS Collaboration, *ATLAS Software and Computing HL-LHC Roadmap*, CERN-LHCC-2022-005, LHCC-G-182 (2022), https://cds.cern.ch/record/2802918

[7] X. Ai *et al.*, *A Common Tracking Software Project*, Computing and Software for Big Science **6**, 8 (2022), https://doi.org/10.1007/s41781-021-00078-8

[8] A. Adare *et al.*, *An Upgrade Proposal from the PHENIX Collaboration* (2015), https://arxiv.org/abs/1501.06197

[9] J. D. Osborn, *Requirements, Status, and Plans for Track Reconstruction at the sPHENIX Experiment* (2020), https://arxiv.org/abs/2007.00771

[10] J. D. Osborn *et al.*, *Implementation of ACTS into sPHENIX Track Reconstruction*, Computing and Software for Big Science **5**, 23 (2021), http://dx.doi.org/10.1007/s41781-021-00068-w

[11] P. Bechtle *et al.*, *A Proposal for the Lohengrin Experiment to Search for Dark Sector Particles at the ELSA Accelerator* (2024), https://arxiv.org/abs/2410.10956

[12] FASER Collaboration, *FASER: ForwArd Search ExpeRiment at the LHC* (2019), https://arxiv.org/abs/1901.04468

[13] ATLAS Collaboration, *Expected Tracking Performance of the ATLAS Inner Tracker at the HL-LHC*, ATL-PHYS-PUB-2019-014 (2019), https://cds.cern.ch/record/2669540

[14] R. Frühwirth, *Application of Kalman filtering to track and vertex fitting*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **262**, 444 (1987), https://doi.org/10.1016/0168-9002(87)90887-4

[15] A. Buckley *et al.*, *Implementation of the ATLAS Run 2 event data model*, J. Phys.: Conf. Ser. **664**, 072045 (2015), http://dx.doi.org/10.1088/1742-6596/664/7/072045

[16] P. Gessinger, *Flexible, robust and minimal-overhead Event Data Model for track reconstruction in ACTS*, EPJ Web of Conferences **295**, 03015 (2024), https://doi.org/10.1051/epjconf/202429503015

[17] J. Hoshen and R. Kopelman, *Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm*, Phys. Rev. B. **14**, 3438 (1976), https://dx.doi.org/10.1103/PhysRevB.14.3438

[18] C. Allaire *et al.*, *CTD2023: Seed Finding in the ACTS Software Package: Algorithms and Optimizations.*, Connecting The Dots Workshop 2023 (CTD2023), Zenodo (2024), https://dx.doi.org/10.5281/zenodo.14246396

[19] ATLAS Collaboration, *Initial improvements to the CPU performance for the ACTS-based track reconstruction software for ATLAS during the HL-LHC*, ATL-PHYS-PUB-2024-017 (2024), https://cds.cern.ch/record/2912217