ACTS Vertexing

BASTIAN SCHLAG

CERN / JGU MAINZ

The ACTS Vertexing Suite

Several ATLAS (primary) vertexing algorithms reimplemented in ACTS
+ numerically validated w.r.t. ATLAS physics performance

Vertex (Seed) Finder:

- Iterative Vertex Finder (IVF)
- Adaptive Multi-Vertex Finder (AMVF)
- Z-Scan Vertex Finder
- Gaussian Track Density Vertex Finder
- Grid Density Vertex Finder

ATLAS Run-2
ATLAS Run-3

Vertex Fitter:

- Full-Billoir Vertex Fitter
- Adaptive Multi-Vertex Fitter

Utilities:

- Track linearizer, impact point estimator, annealing tool, …

- Modern C++
- Thread-safe, modular design
- Fast
- Very well documented

AMVFInfo.hpp	add missing std::map include	18 days ago
AdaptiveMultiVertexFin...	save deletedSeedTracks in vector and pass to seedFinder sta...	9 days ago
AdaptiveMultiVertexFin...	bug fix, track density removal now working	9 days ago
AdaptiveMultiVertexFitt...	remove propagator options from IP estimator configs and cre...	17 days ago
AdaptiveMultiVertexFitt...	remove propagator options from IP estimator configs and cre...	17 days ago
DummyVertexFitter.hpp	merge VertexFitterOptions and VertexFinderOptions to Verte...	17 days ago
FsmwMode1dFinder.hpp	FsmwMode1dFinder and TrackToVertexIPEstimator added	11 months ago
FullBilloirVertexFitter.hpp	add method to retrieve maximum density value	9 days ago
FullBilloirVertexFitter.ipp	remove BoundParsDim	15 days ago
GaussianGridTrackDen...	addTrack returns pair of zBin and trackGrid to allowing cachi...	9 days ago
GaussianGridTrackDen...	addTrack returns pair of zBin and trackGrid to allowing cachi...	9 days ago
GaussianTrackDensity....	remove unnecessary const keyword in vector of BoundParam...	last month
GridDensityVertexFind...	save deletedSeedTracks in vector and pass to seedFinder sta...	9 days ago
GridDensityVertexFind...	use constraint vertex correctly in grid density finder and han...	8 days ago
HelicalTrackLinearizer....	remove propagator options from HelicalTrackLinearizer confi...	17 days ago
HelicalTrackLinearizer.i...	remove BoundParsDim	15 days ago
ImpactPoint3dEstimato...	remove propagator options from IP estimator configs and cre...	17 days ago
ImpactPoint3dEstimato...	remove propagator options from IP estimator configs and cre...	17 days ago
IterativeVertexFinder.hpp	introduce vertex finder state for consistency reasons to all v...	9 days ago
IterativeVertexFinder.ipp	introduce vertex finder state for consistency reasons to all v...	9 days ago
KalmanVertexTrackUpd...	remove SpacePointDim	15 days ago
KalmanVertexTrackUpd...	remove SpacePointDim	15 days ago
KalmanVertexUpdater.h...	bug fixes: remove auto keyword where Eigen doesnt like it	last month
KalmanVertexUpdater.ipp	bug fixes: remove auto keyword where Eigen doesnt like it	last month
LinearizedTrack.hpp	remove BoundParsDim	15 days ago
LinearizerConcept.hpp	Apply suggestion to Core/include/Acts/Vertexing/LinearizerC...	17 days ago
TrackAtVertex.hpp	bug fixes: remove auto keyword where Eigen doesnt like it	last month
TrackDensity.hpp	remove unnecessary const keywords in set	last month
TrackDensityVertexFin...	introduce vertex finder state for consistency reasons to all v...	9 days ago
TrackDensityVertexFin...	introduce vertex finder state for consistency reasons to all v...	9 days ago
TrackToVertexIPEstimat...	remove propagator options from IP estimator configs and cre...	17 days ago
TrackToVertexIPEstimat...	remove propagator options from IP estimator configs and cre...	17 days ago
Vertex.hpp	bug fixes: remove auto keyword where Eigen doesnt like it	last month
Vertex.ipp	bug fixes: remove auto keyword where Eigen doesnt like it	last month
VertexFinderConcept.h...	introduce vertex finder state for consistency reasons to all v...	9 days ago
VertexFitterConcept.hpp	merge VertexFitterOptions and VertexFinderOptions to Verte...	17 days ago
VertexingError.hpp	add more error cases to vertexing error	9 months ago
VertexingOptions.hpp	merge VertexFitterOptions and VertexFinderOptions to Verte...	17 days ago
ZScanVertexFinder.hpp	introduce vertex finder state for consistency reasons to all v...	9 days ago
ZScanVertexFinder.ipp	introduce vertex finder state for consistency reasons to all v...	9 days ago

The ACTS Vertexing Suite

- Easy to set up & use (in backup: example of running ACTS vertexing in <40 lines of code)

  → Easily employable in any reconstruction framework

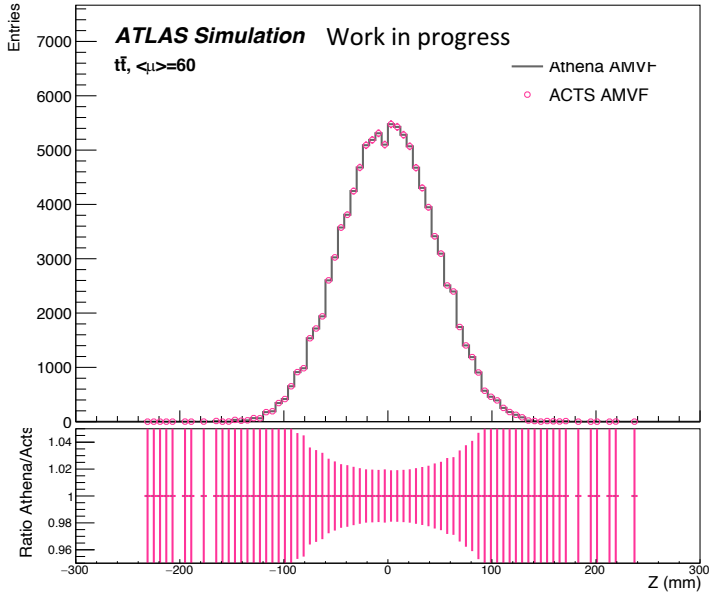
- Fully integrated in ATLAS reconstruction framework Athena
  - Physics performance validated w.r.t. ATLAS primary vertexing tools (see next slides)
  - Single- and multi-threaded execution modes validated
  - Excellent CPU performance

  → ACTS primary vertexing will be default for ATLAS Run-3 data taking

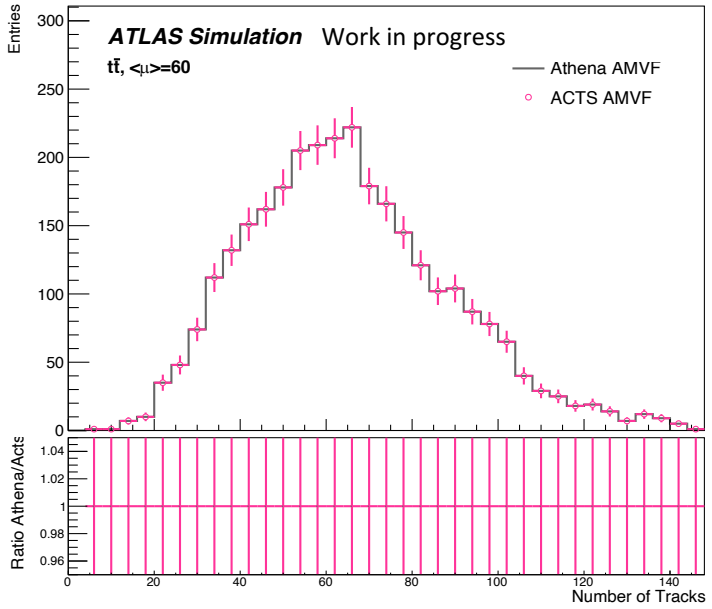
ACTS – AMVF Validation (Single-Threaded)

- **Perfect agreement** between ACTS AMVF (single-threaded) and ATLAS AMVF
- Only small numerical fluctuations on all tested variables (Note: different extrapolator/propagator methods are used)
- Validation on different sample, here: 3k $t\bar{t}$ **(**$< \mu >= 60$**)** events

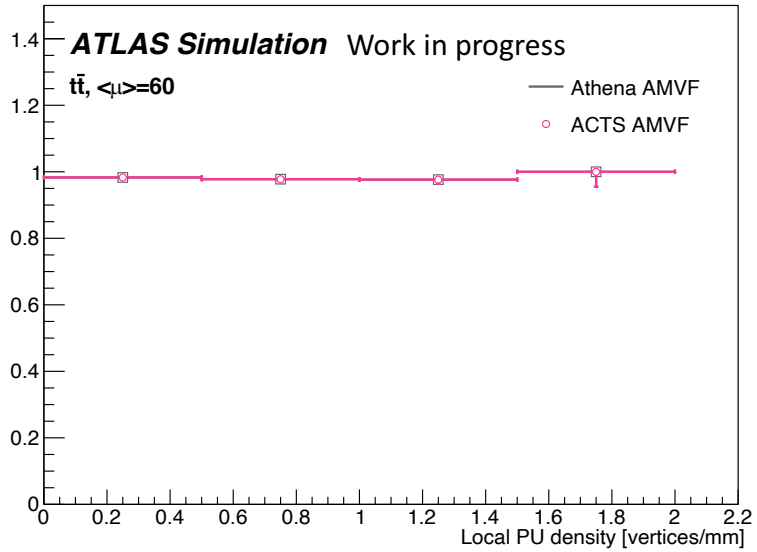
All vertices – z-positions



All vertices – number of tracks



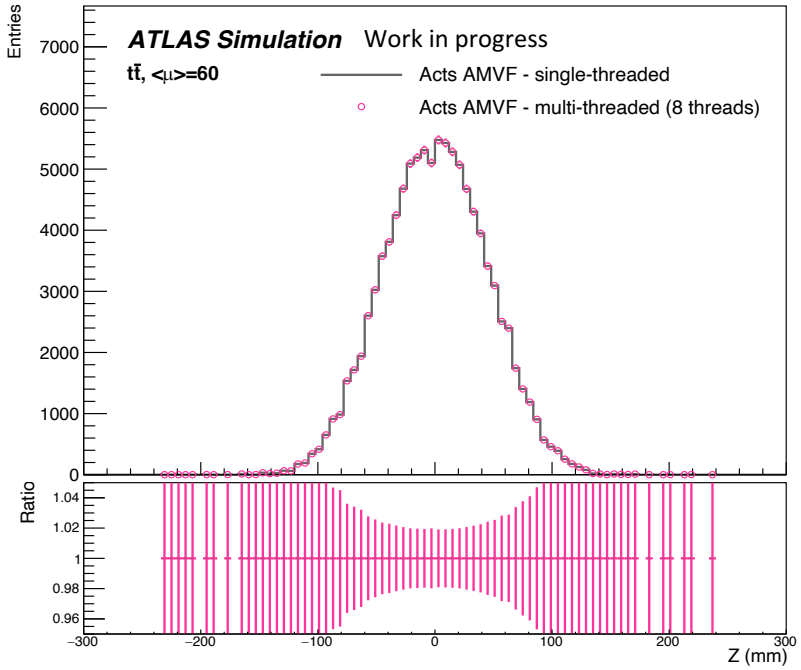
Hard scatter – selection efficiency



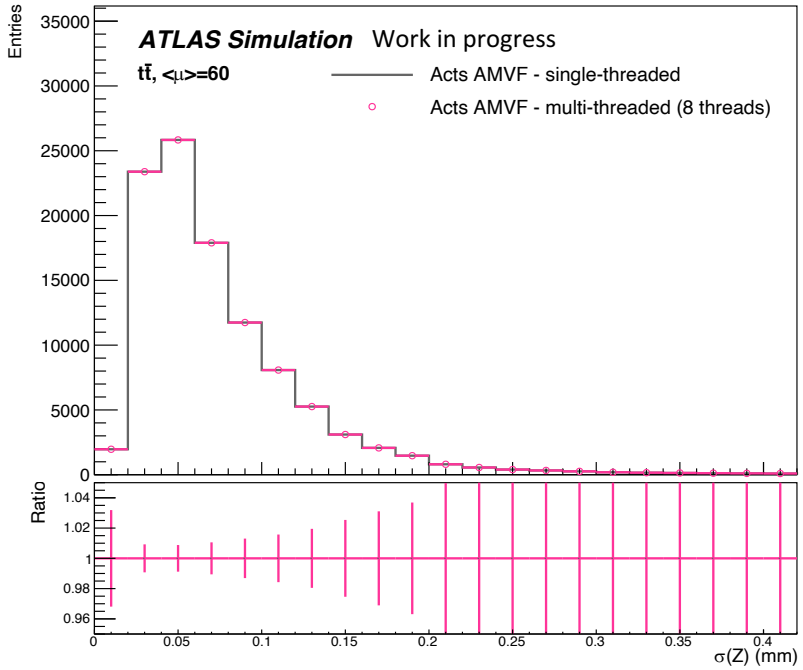
ACTS – AMVF Validation (Multi-Threaded)

- **<u>Multi-threaded execution</u> (on 8 threads) validated**
- **Perfect agreement** between ACTS AMVF (multi-threaded) and ACTS AMVF (single-threaded)
- Validation on 3k $\boldsymbol{t\bar{t}}$ **(**$\boldsymbol{< \mu >= 60}$**)** events

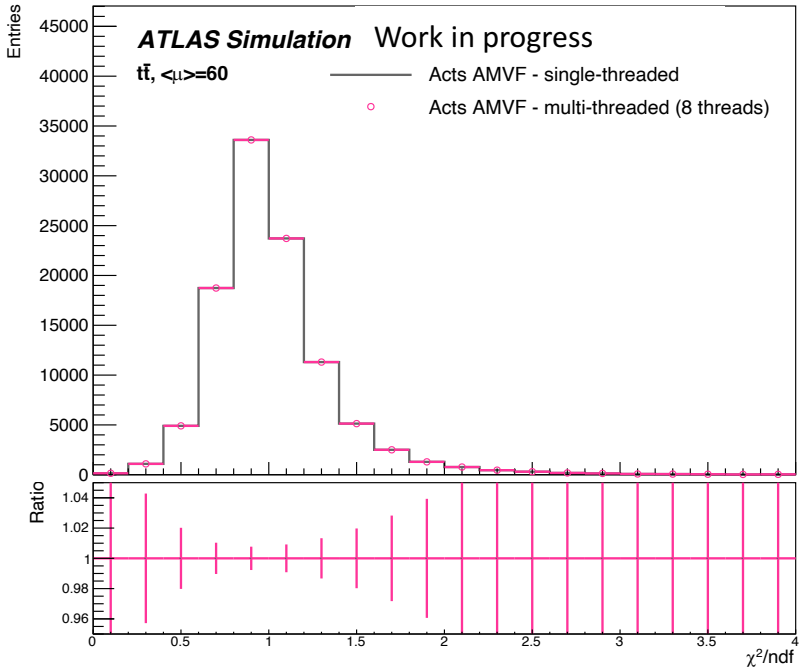
All vertices – z-positions



All vertices – z-position error



All vertices – reduced χ^2

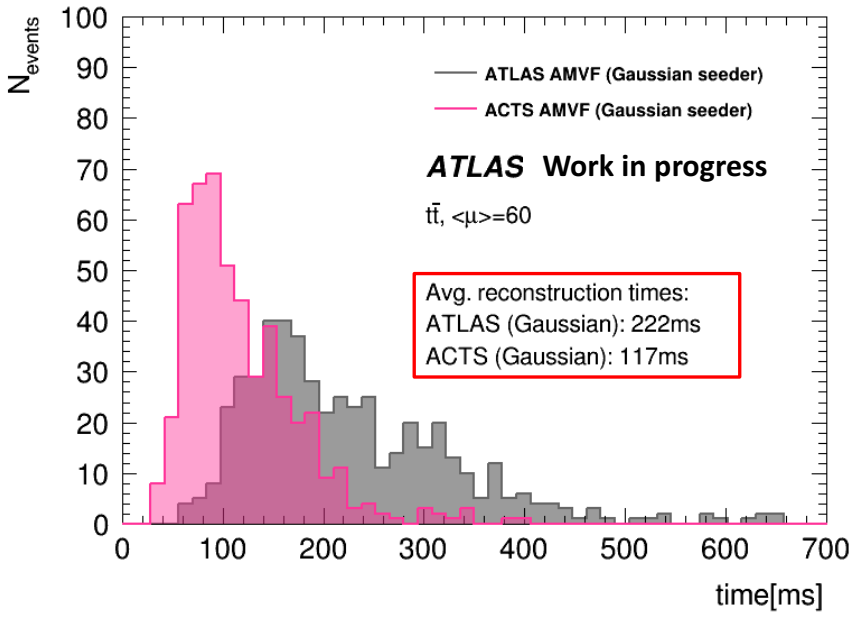


ACTS – AMVF CPU Performance – Single-Threaded

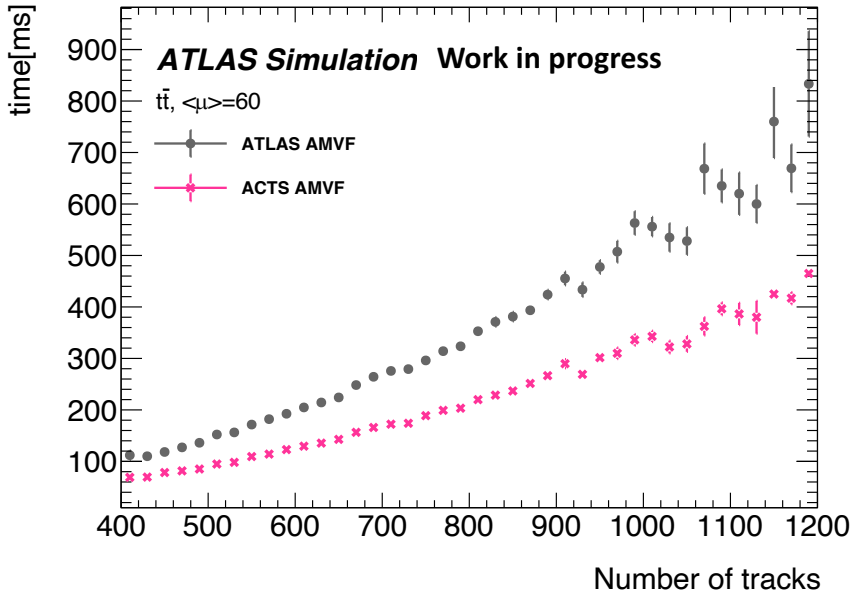
- ACTS AMVF shows overall speed-up vs. ATLAS AMVF (including EDM conversion): **2x faster!** (both single-threaded)
- Higher speed-ups for high track multiplicities
- No algorithmic changes, only code improvements

→ same physics performance, 2x faster + entirely thread-safe

PV reconstruction time per event



PV reconstruction time vs. number of accepted tracks



Current Vertexing Developments in ACTS

Generalization of track linearization using the ACTS::Propagator:

$$\vec{q} = \vec{q}(\vec{r}, \vec{p}) = A\vec{r} + B\vec{p} + \vec{c}_0$$

Retrieve dedicated Jacobians from ACTS::Propagator

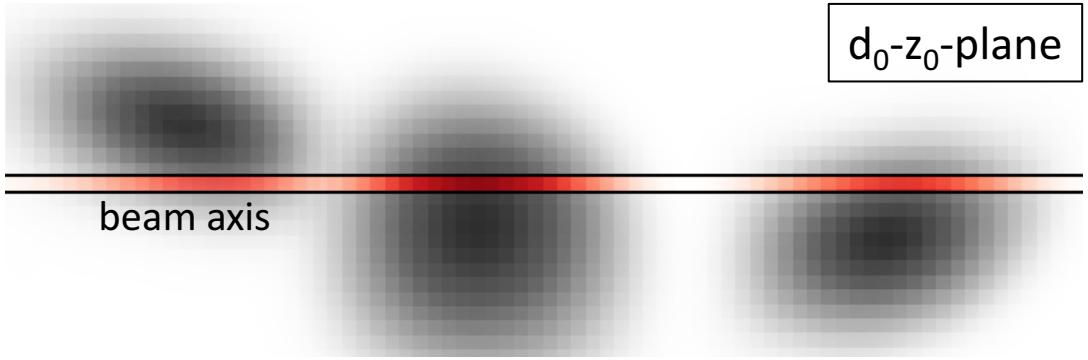
- No assumption of helical track parameters anymore
- Vertex fitter more robust in all detector regions
- Harmonize primary and secondary vertexing with common math kernels
- Fully integrated time propagation in ACTS Vertex fitting with time information possible
- → Currently WIP and contributions are always welcome

Gaussian Grid Track Density Vertex Seed Finder:

- Model track as 2-dim Gaussian density grid in d_0-z_0-plane

  → calculate only track contribution along beam axis (red)
- Superimpose all tracks and find maximum along beam axis

Extremely fast in iterative approaches



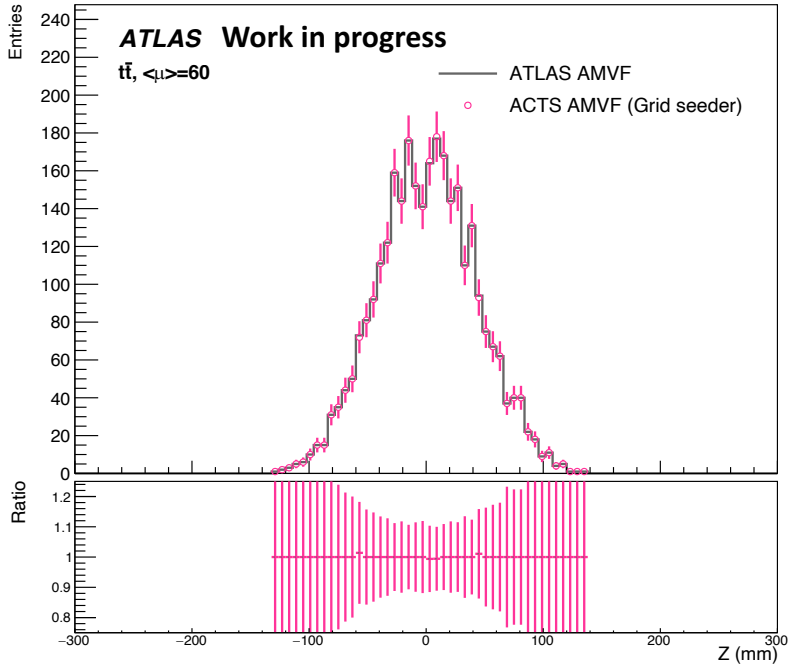
Example: Track density representations of 3 single tracks

ACTS – Gaussian Grid Seed Finder Performance

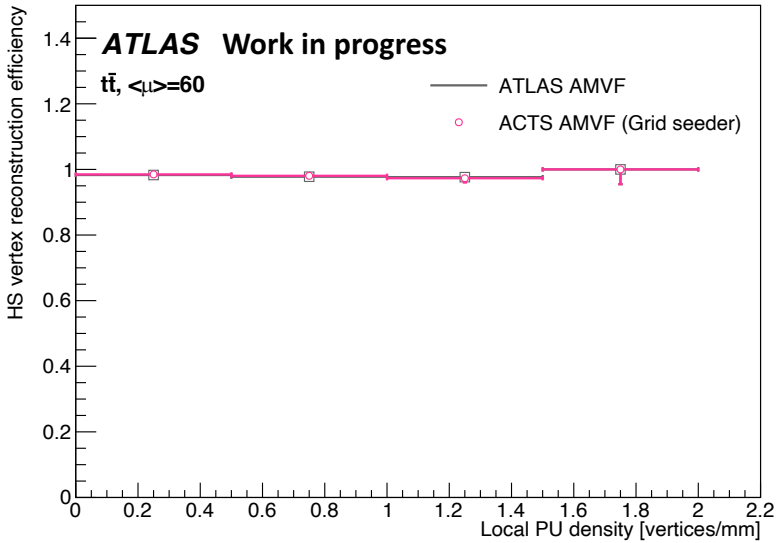
- Grid seed finder itself up to 40x faster than ATLAS default seed finder
- AMVF with Grid seeder **almost 3x faster** vs. ATLAS on 3k $t\bar{t}$ ($< \mu >= 60$) events
- AMVF with Grid seeder shows **similar physics performance** to ATLAS AMVF

  → Excellent physics & CPU performance

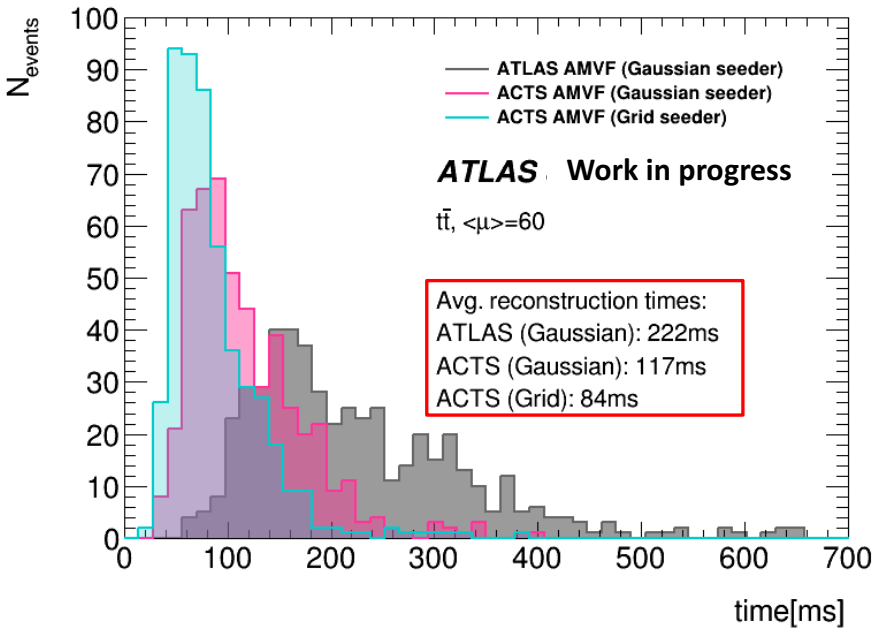
HS vertices – z-positions



HS reconstruction efficiency



PV reconstruction time per event



Summary & Outlook

- **Modern, fast & MT-capable** vertexing suite implemented in ACTS
- Easy to integrate in any reconstruction framework

  → fully integrated in ATLAS reconstruction framework and set as default primary vertexing tool
- **Validated** in **single-threaded** and **multi-threaded** execution mode
- **2-3x faster** than original implementations
- **Generalization of track linearization** work ongoing
  - Harmonize vertex fitting for primary and secondary vertexing + include time information
  - Any contribution is very welcome → let us know if you are interested
- **New seed finder** available: Excellent physics & CPU performance

bastian.schlag@cern.ch

BACKUP

Employing the ACTS Vertexing

Input to ACTS Vertexing:

ACTS::BoundParameters

or

Arbitrary user-defined track type:

```
// Dummy user-defined InputTrack type
struct InputTrack {
  InputTrack(const BoundParameters& params) : m_parameters(params) {}

  /* Store additional information here */

  // Method to return BoundParameters
  const BoundParameters& parameters() const { return m_parameters; }

 private:
  BoundParameters m_parameters;
};
```

Only requirement for user-defined track:

std::function which returns ACTS::BoundParameters

```
// Create a custom std::function to extract BoundParameters from
// user-defined InputTrack
std::function<BoundParameters(InputTrack)> extractParameters =
    [](InputTrack params) { return params.parameters(); };
```

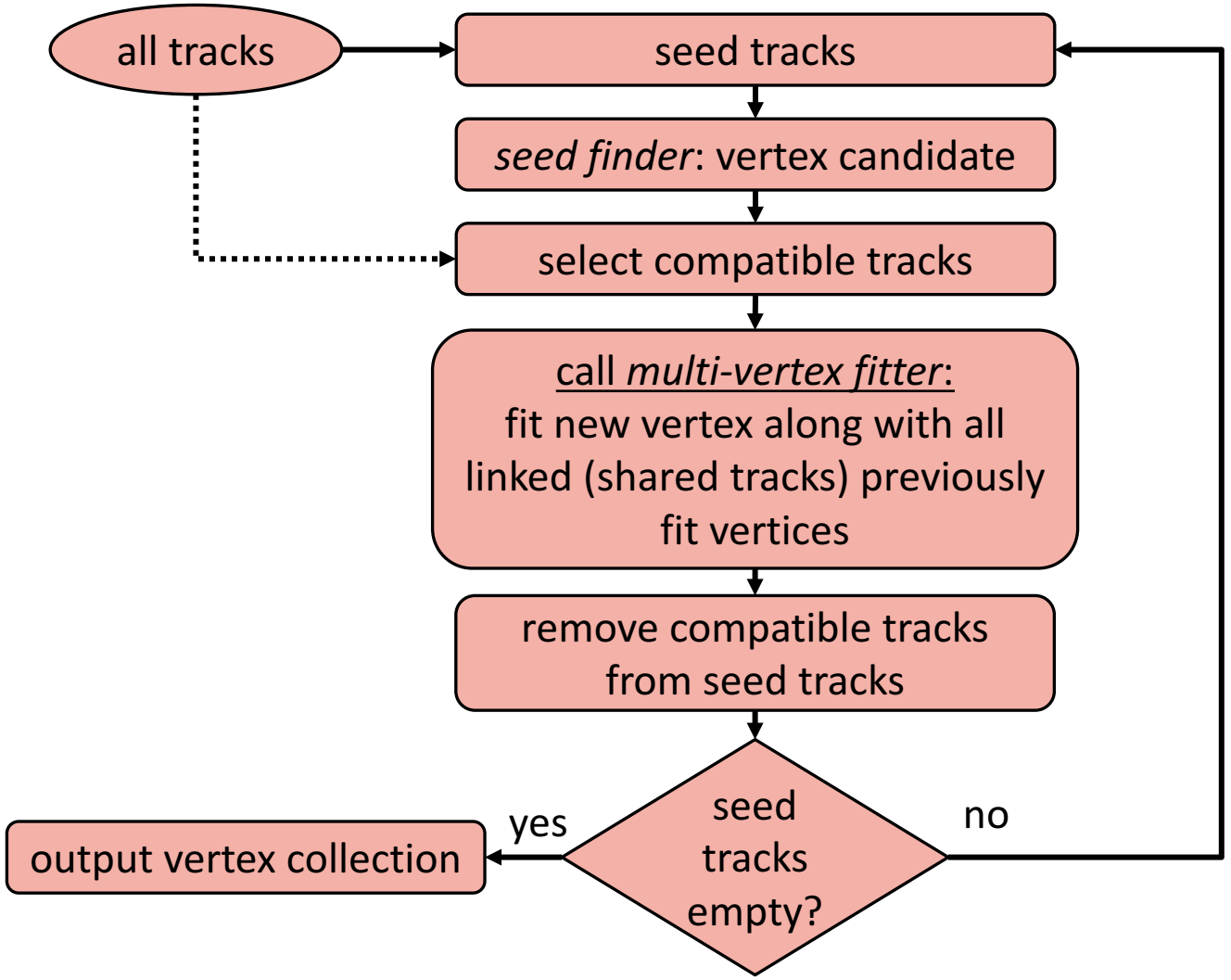
Set up and run the ACTS vertexing in < 40 lines of code

```
// Set up B-Field and Propagator
ConstantBField bField(0.0, 0.0, 1_T);
EigenStepper<ConstantBField> stepper(bField);
auto propagator = std::make_shared<Propagator>(stepper);

// Linearizer for user defined InputTrack type
Linearizer::Config linearizerCfg(bField, propagator);
Linearizer linearizer(linearizerCfg);

using BilloirFitter = FullBilloirVertexFitter<InputTrack, Linearizer>;
BilloirFitter::Config vertexFitterCfg;
BilloirFitter bFitter(vertexFitterCfg, extractParameters);

using IPEstimator = ImpactPointEstimator<InputTrack, Propagator>;
IPEstimator::Config ipEstimatorCfg(bField, propagator);
IPEstimator ipEstimator(ipEstimatorCfg);

using ZScanSeedFinder = ZScanVertexFinder<BilloirFitter>;
ZScanSeedFinder::Config seedFinderCfg(ipEstimator);
ZScanSeedFinder sFinder(seedFinderCfg, extractParameters);

using VertexFinder = IterativeVertexFinder<BilloirFitter, ZScanSeedFinder>;
VertexFinder::Config cfg(bFitter, linearizer, std::move(sFinder), ipEstimator);
VertexFinder finder(cfg, extractParameters);
VertexingOptions<InputTrack> vertexOptions(geoContext, magFieldContext);
```

Run vertex finder on track collection

```
std::vector<const InputTrack*> tracksPtr = getAllTracks();
auto finderResult = finder.find(tracksPtr, vertexOptions);
if (finderResult.ok()) {
  std::vector<Vertex<InputTrack>> allVertices = *finderResult;
}
```

Adaptive Multi-Vertex Finder (AMVF)



Gaussian Track Density Seed Finder:

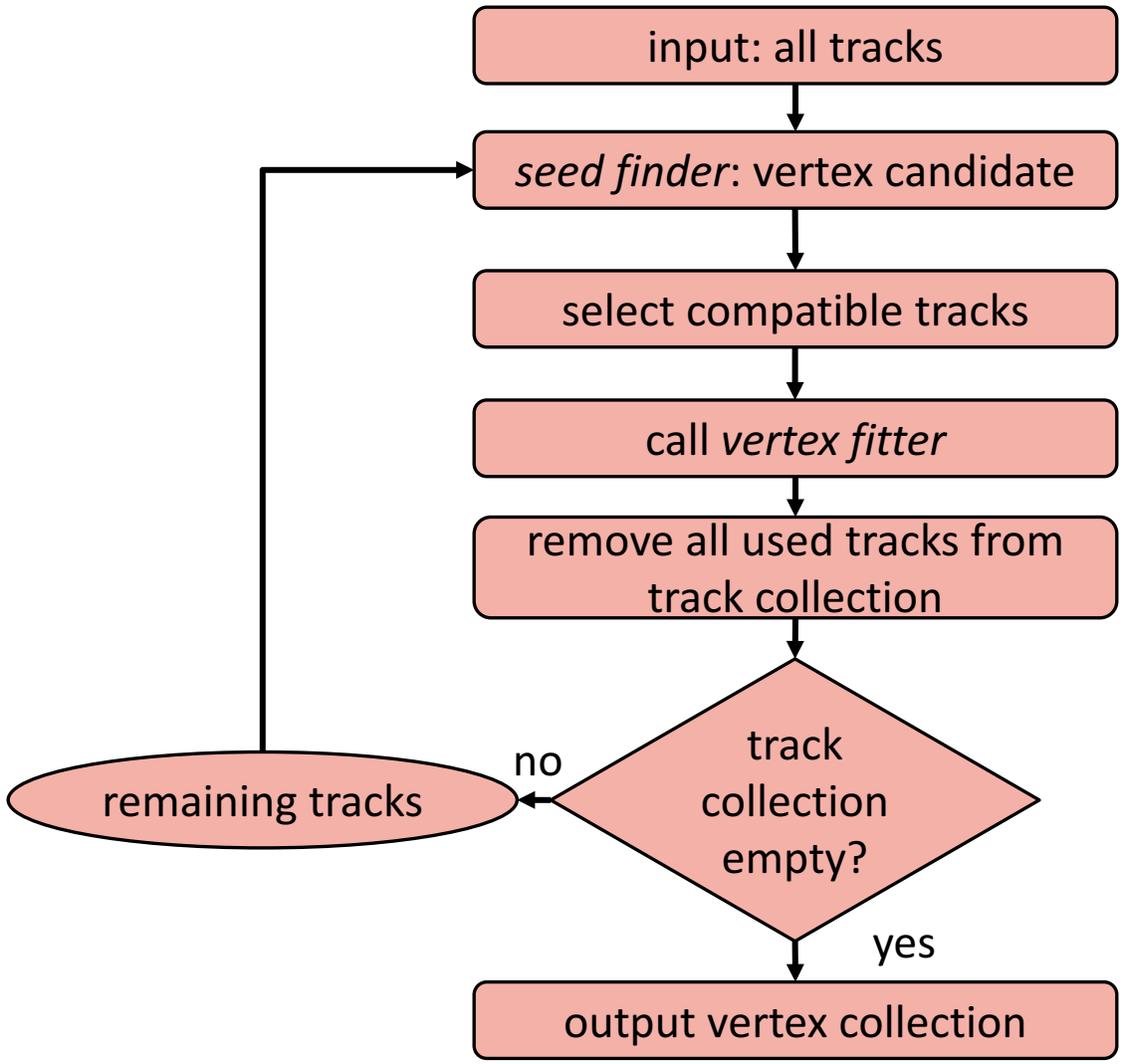
- model each track as 2-dim Gaussian distribution in d_0-z_0-plane around (d_0, z_0)
- find z value of highest track density along z-axis

Adaptive Multi-Vertex Fit:

- weighted adaptive Kalman filter using deterministic annealing scheme
- subject to beamspot and seed constraint
- simultaneous refit of all vertices connected through a chain of vertices and tracks, with weights:

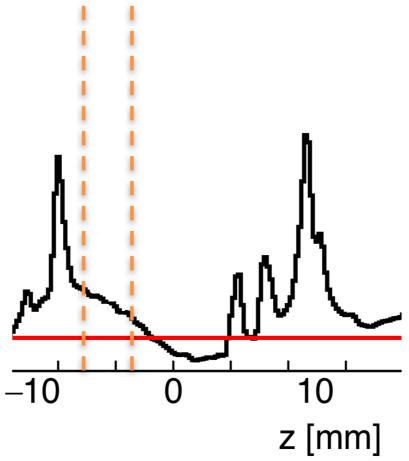
$$\omega_i(\chi_i^2, T) = \frac{e^{-\frac{1}{2}\chi_i^2/T}}{\sum_j e^{-\frac{1}{2}\chi_j^2/T} + e^{-\frac{1}{2}\chi_0^2/T}}$$

Iterative Vertex Finder (IVF)



ZScanVertexFinder:

- find mode value of all z_0 track parameters
- vertex candidate at position (z_0, 0, 0)



Iterative fitting-after-finding approach:

- iteratively find vertex and fit with compatible tracks
- single track always associated to at most one vertex
- tracks removed from pool after fitting