Vertex seeding at high pile up

*Cléo E. Nicollin**
CERN, Geneva, Switzerland

Abstract
This report presents an investigation into vertex seeding algorithms under high pile-up conditions, with a focus on integrating a time parameter into the Gaussian Track Density (GTD) model. The aim is to improve vertex reconstruction precision by incorporating 4D tracking in the high-luminosity environment of the Large Hadron Collider. Through algorithm comparisons and parameter optimization, the project highlights improvements in CPU efficiency without sacrificing physics performance. Initial results suggest that the time parameter integration provides promising advancements, with potential applications in future collider experiments.

Keywords
CERN Summer Student report; tracking; ACTS; vertex seeder.

1 Introduction

The European Organization for Nuclear Research, or CERN, is a global leader in particle physics, pioneering significant discoveries through the Large Hadron Collider (LHC). The ATLAS collaboration, one of CERN's major experiments, is crucial in probing the fundamental particles of matter. As the LHC transitions from Run 3 to Run 4, substantial upgrades will transform it into the High-Luminosity LHC (HL-LHC), boosting luminosity to $7.5 \times 10^{34}\text{cm}^{-2}\text{s}^{-1}$ by 2027 [1], significantly increasing proton-proton collision rates and the potential for new discoveries (ATLAS Collaboration, "Expected Performance").

*Wellesley College, MA, USA

Figure 1: Schematic representation of the HL-LHC timeline. Fig from High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report V. 0.1. Nov. 2017. doi:10.23731/CYRM-2017-004.

To maintain its performance under these intense conditions, the ATLAS experiment will undergo critical upgrades, essential for managing the increased "pileup"—where multiple collisions occur quasi-simultaneously, complicating event detection (CITE). The ability of ATLAS to accurately reconstruct particles from these complex collisions is a key focus as it prepares for the HL-LHC environment.

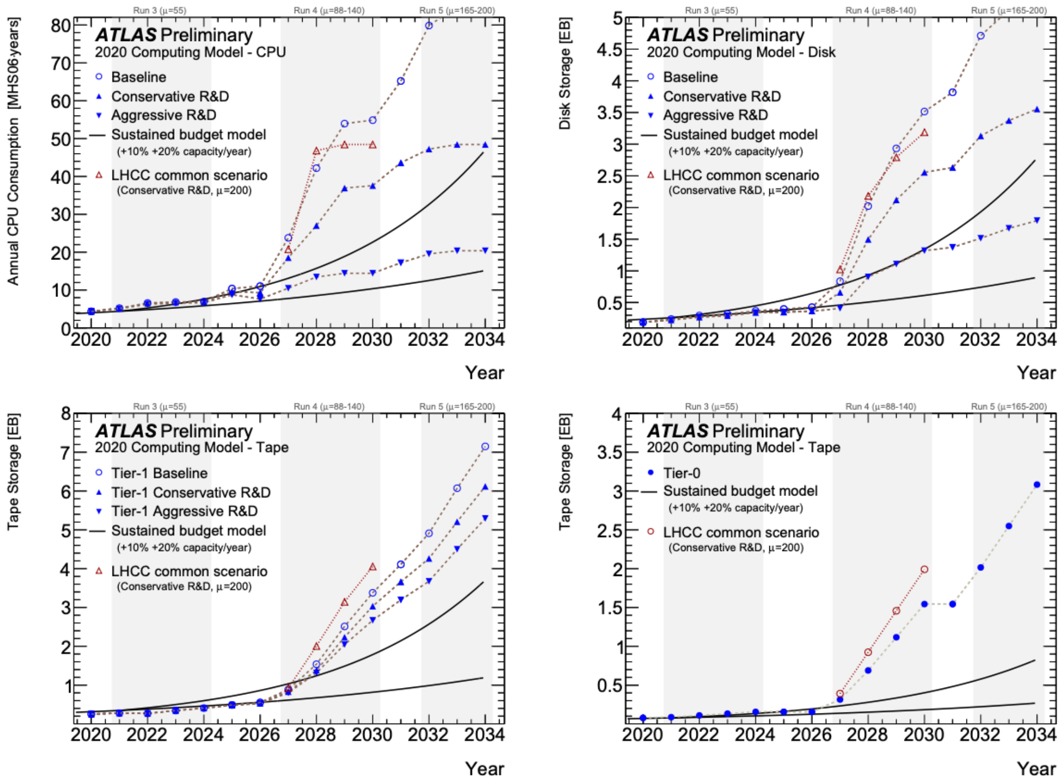


Figure 2: Caption from og paper! ATLAS HL-LHC Computing Conceptual Design Report !: Estimated CPU, disk and tape (at the Tier-1 and Tier-0) resources needed for the years 2020 to 2034 under the different scenarios described in the text. The solid lines indicate annual improvements of 10% and 20% in the capacity of new hardware for a given cost, assuming a sustained level of annual investment. The blue dots with the brown dashed lines represent the three ATLAS scenarios following the current LHC schedule. The red open triangles indicate the Conservative R&D scenario under an assumption of the LHC reaching hµi = 200 in 2028.

Central to these upgrades is the Acts Common Tracking Software (ACTS), a high-performance toolkit designed to provide efficient and precise tracking capabilities for ATLAS. As the LHC advances into the High-Luminosity era, accurate particle trajectory reconstruction amidst vastly increased collision rates becomes essential. ACTS addresses this by offering a flexible, modular framework adaptable to the evolving needs of the ATLAS detector. ACTS is optimized to handle the extreme pileup levels anticipated during the HL-LHC era. Its advanced algorithms are capable of reconstructing tracks from the massive data generated during each collision, ensuring that ATLAS maintains its precision in particle identification. The software's architecture is both detector-agnostic and future-proof, allowing adaptation to upgrades in ATLAS or different detector configurations. Furthermore, ACTS integrates seamlessly with multi-threaded and distributed computing environments, which is crucial for processing the enormous datasets expected from the HL-LHC. It also provides essential tools for event simulation, tracking geometry, and visualization, supporting the development of new tracking strategies under high-luminosity conditions. The successful implementation of ACTS is expected to significantly enhance ATLAS's ability to analyze complex events, making it a crucial enhancement for the future success of the collaboration's physics program [2] [3].

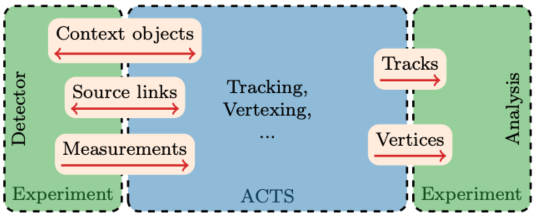
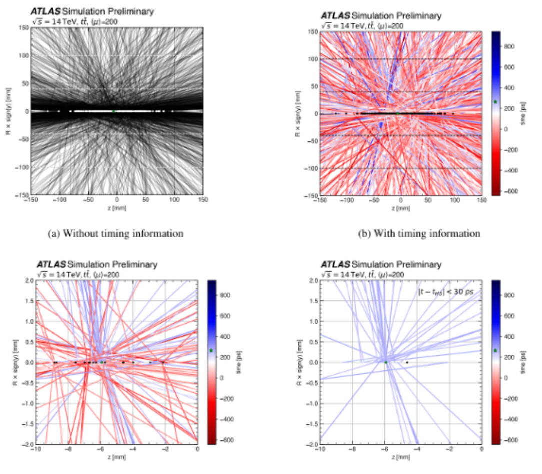


Figure 3: Caption from of paper ! ACTS arXiv:2106.13593v1! Example of the integration of ACTS into an experiment's software framework. The experiment- and detector-specific code (green) is expected to handle lowlevel data preparation and provide SourceLinks and Measurements as input to ACTS algorithms. ACTS provides tracks and vertices as output for further experiment-specific reconstruction and analysis.

2 Motivations

4D tracking is an exciting development in high-energy physics, adding an essential dimension to the analysis of particle collisions by incorporating time information alongside spatial data. This approach helps scientists better understand complex events in particle detectors. Traditional tracking systems rely mostly on spatial information, which can lead to challenges when trying to distinguish between events that occur very close together in space, especially under conditions of high pile-up where many particle interactions happen quasi-simultaneously.

One of the major limitations of current tracking systems is the difficulty in accurately assigning particle tracks to the correct collision event when multiple events overlap. This can lead to errors in the reconstruction of physics objects, such as b-jets, and in identifying rare processes, particularly those involving long-lived particles that decay at some distance from the interaction point. 4D tracking addresses this by adding precise timing information to each particle track, allowing scientists to more clearly differentiate between events that are close in space but separated in time.



(a) Without timing information

(b) With timing information

(c) With timing information, spatial zoom

(d) With timing information, spatial zoom and time cut

Figure 4: Enter Caption

Additionally, 4D tracking can improve the performance of seeding algorithms, which are used to initiate the track reconstruction process. By incorporating time information, these algorithms can more effectively filter out noise and irrelevant data, reducing the computational load and improving the accuracy of the tracks that are reconstructed. This enhancement is particularly important as we move into the era of the High-Luminosity LHC, where the number of simultaneous interactions per collision is expected to increase significantly.

3 Summer Specific Work

This summer has been a deep dive into the world of algorithm performance, both from a computational and a physics standpoint. After spending some time installing ACTS and all its dependencies using CMake and various scripts, it was time to look into the performance of the seeder algorithms. To do so, we would run the algorithm with specific grid parameters, get the CPU and physics results, and compare between different parameters and pile-up. This process has revealed a lot about the potential of these algorithms in real-world scenarios. The ability to evaluate both the computational efficiency and the accuracy in the physics domain has been enlightening, offering valuable insights into where improvements can be made.

In addition to exploring algorithmic performance, I expanded my skills across several programming languages and tools. This project demanded a solid understanding of C++, Python, and the use of the Terminal, along with an introduction to Root, which was new to me. It was a rewarding challenge to integrate these technologies, and I can confidently say my proficiency in each has grown significantly over the course of the summer.

3.1 Algorithm Comparisons

This summer's project was focused on evaluating the performance of algorithms from both computational and physics perspectives. Initially, simulations were run on a local machine, with a Python script operating in the background to collect data. However, the script required user input at various stages, which created the need for continuous monitoring and limited the ability to multitask or run additional processes efficiently. On top of this, the seeder algorithms consumed a large amount of memory, significantly slowing down the machine and making it difficult to work effectively. At one point, the system became so sluggish that daily restarts were necessary just to maintain functionality.

To overcome these limitations, the simulations were moved to an LXPLUS machine, which provided a significant boost in performance. The ability to run multiple programs simultaneously allowed for much more data to be processed in a shorter time. Data outputs were stored in ROOT files, enabling a more structured approach to data management and analysis.

By automating data collection through Python and leveraging the enhanced computational power of LXPLUS, the project achieved far greater efficiency. This approach not only resulted in a more robust dataset but also allowed for deeper insights into the performance of the algorithms under different conditions. It demonstrated the crucial role that proper resource management and high-performance computing play in large-scale research and development projects.

Let's take a closer look at the results and their significance. Figure 5 is one of the final plots that we produced for this project. It represents a lot of information both on the physics and computer side, with the main objective being to show that the implementation of time is indeed beneficial.

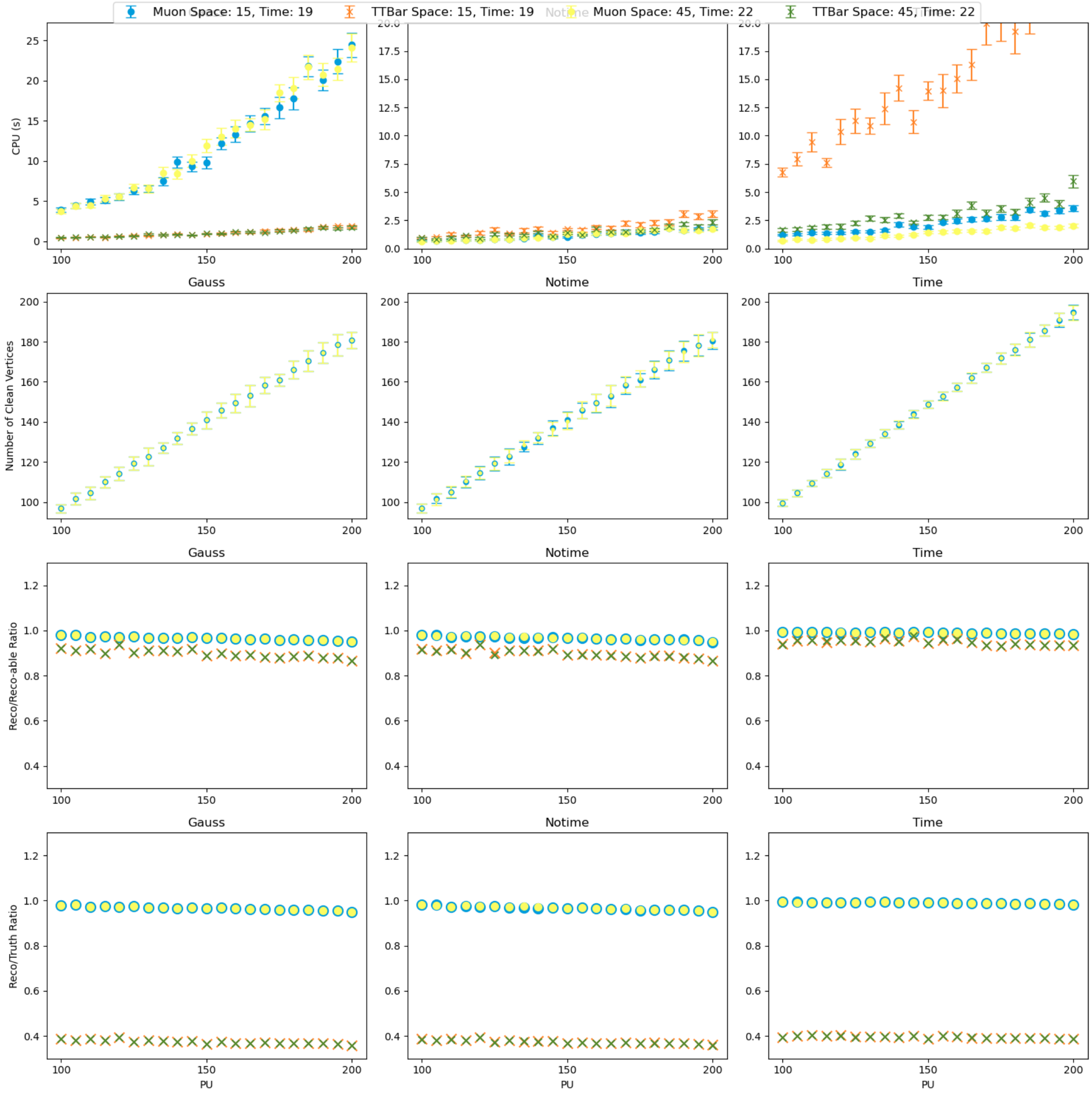


Figure 5: From left to right: First column represents the Gauss Seeder algorithm's results, the second column is the Adaptative Multi Vertex Finder (AMVF) algorithm without time implementation and its performance and the third column is the AMVF algorithm with time implemented and its results. Each color corresponds to a different grid configuration and/or type of events. The main result observed with this plot is that the implementation of time does provide a better CPU time (first line) while also keeping the physics performance quite the same.

One of the first parts of the project was about discovering the ACTS implementations and their performance, this included looking into the grid implementation (AMVF) and investigating its parameters. The grid has two main parameters; time and space. We can look at how those parameters affect the performance of the algorithm by changing them and running the whole process. From Figure 5, we can compare the default parameters (orange) to some new parameters (green) for TTbar events. With the newly chosen parameters we can observe a significant difference, especially in the performance of the algorithm with time implemented. The new parameters provide a better CPU while still maintaining the physics performance compared to the default parameters. It was not investigated 'why' those parameters were better but could be a interesting study.

3.2 Gaussian Time Implementation

The objective of this part of the project was to incorporate a time parameter into the existing *Gaussian Track Density (GTD)* model used for vertex reconstruction. The time parameter allows for a more precise estimation of vertex positions by considering temporal uncertainties alongside spatial ones, which is specifically important in high-luminosity environments.

3.2.1 Extending the Covariance Matrix

The original implementation of the GTD model used a 2x2 covariance matrix to represent uncertainties in the transverse (d_0) and longitudinal (z_0) impact parameters. To include time, the covariance matrix was extended to a 3x3 matrix. This new matrix incorporates not only the variances for d_0, z_0, and time, but also the covariances between them: σ_{dz}, σ_{dt}, and σ_{zt}. By extending the matrix, the model can now account for correlations between the spatial and temporal measurements, which is crucial for precise track density calculations.

The updated covariance matrix was implemented as follows:

```
Eigen::Matrix3d covMatrix;
covMatrix << covDD, covDZ, covDT,
             covDZ, covZZ, covZT,
             covDT, covZT, covTT;
```

This ensures that the time component is fully integrated into the model's uncertainty representation.

3.2.2 Updating Track Density Calculations

With the covariance matrix updated, the track density calculations needed to account for time-based terms. The *constant term*, which reflects the base value of the Gaussian function, was modified to include contributions from the time parameter. Similarly, the *linear term*, which describes how the density changes with respect to z, was updated to factor in the time component. The *quadratic term*, responsible for the spread of the density function, remained largely unchanged, as it primarily deals with spatial uncertainty.

For example, the constant term was updated as follows:

$$c_0 = -\frac{1}{2}\left(d_0^2\sigma_{zz} + z_0^2\sigma_{dd} + t^2\sigma_{tt} + 2d_0z_0\sigma_{dz} + 2d_0t\sigma_{dt} + 2z_0t\sigma_{zt}\right) / \det(\Sigma) \tag{1}$$

These changes should ensure that the Gaussian track density function now incorporates the time parameter when calculating density, making the vertex reconstruction more robust. However, this may be one of the reasons this new implementation does not produce results. This part could be investigated further.

3.2.3 Handling Time in the InputTrack Structure

Initially, the `InputTrack` structure did not contain a member for `time`, which led to complications when trying to extract temporal information. To address this, we either modified the `InputTrack` structure to include `time` directly or updated the *Extractor* function to handle time appropriately. Once time was included, it was extracted and passed through the same density calculations as the spatial parameters, ensuring consistent integration throughout the GTD process.

```
const double time = boundParams.parameters()[BoundIndices::eBoundTime];  // Extract time
```

By ensuring that `time` was available and properly extracted, the model was able to handle temporal data in addition to spatial information.

3.2.4 Integrating Time into GaussianTrackDensityStore

This would be the next step to the implementation. After handling time in the InputTrack structure as well as implementing it over different functions the integration would be complete. However, because of time restrictions, this part of the project is still underway and may even be the biggest.

By extending the covariance matrix, updating the track density calculations, and integrating time into the `InputTrack` and `GaussianTrackDensityStore` structures, we could potentially successfully incorporate the time parameter into the Gaussian Track Density model. This enhancement would allow for a more accurate reconstruction of vertices by considering both spatial and temporal uncertainties, leading to improved performance in high-luminosity environments such as those encountered at the LHC.

3.3 First Results and Conclusion

The introduction of 4D tracking, with the integration of time as an additional parameter in the Gaussian Track Density model, represents a significant step forward in the precision of vertex reconstruction. The updated covariance matrix and the revised density calculations are key in incorporating this temporal dimension, offering a more comprehensive understanding of particle interactions in high-luminosity environments.

Initial results demonstrate that this enhancement would provide improvements in CPU efficiency while maintaining the accuracy of physics performance, as observed in the algorithm comparisons (refer to Figure 5). With the transition to the LXPLUS machine and automation of data collection, it was possible to handle larger datasets and process more simulations efficiently, leading to better insights into the computational and physics performance of the algorithms.

Looking forward to Run 4, the implementation of ACTS with time parameters is expected to enhance the scalability and accuracy of vertex reconstruction. The combination of spatial and temporal uncertainties will be essential in future high-energy physics experiments. Although there are still areas for further investigation—such as optimizing parameters and completing the full integration of time into the GaussianTrackDensityStore—these early results provide a promising foundation for the ongoing development and application of 4D tracking technology.

4 Outlook

Throughout this journey, I've learned so much—not just from a physics perspective but also in terms of life lessons. Coming into this, I wasn't entirely sure about my future or whether I would find my place in research, but CERN and the team have been incredibly welcoming, making me feel like I belong. Although I'm sad to be leaving, this experience has given me the motivation to return after further study. The project itself has been incredibly fulfilling, even though it's outside my usual area of expertise. Despite the challenges, I enjoyed every part of it and look forward to seeing its continuation in future years. I hope to one day hear about its successes in the news—granted, in scientific news, but still! It's such a cool project, and I believe more people should know about it. The combination of learning on the job, engaging in lectures, and being constantly involved with new tasks has been immensely rewarding. Presenting my research at the poster session was particularly valuable, as it helped me articulate my findings and understand them more deeply, allowing me to push the work further. This experience has only strengthened my desire to continue in this field, and I can't wait to see where the project goes next.

Ackowledgements

I would like to express my sincere gratitude to my supervisors for their invaluable guidance and support throughout this project. A special thanks goes to the ACTS contributors for their continuous efforts in developing and improving the framework, which made this research possible. I would also like to acknowledge the Summer Student team for their collaboration and insightful discussions, which greatly enriched my experience and contributed to the success of this work.

Bibliography

[1] ATLAS Collaboration. Expected performance of the atlas detector under different high-luminosity lhc conditions. *ATL-PHYS-PUB-2021-023*, 2021.

[2] ATLAS Collaboration. Investigating the impact of 4d tracking in atlas beyond run 4. *ATL-PHYS-PUB-2023-023*, 2023.

[3] Xiaocong Ai et al. A common tracking software project. *https://arxiv.org/abs/2106.13593*, 2021.