

Primary Vertex Reconstruction in the ATLAS Experiment at LHC

G Piacquadio¹, K Prokofiev², A Wildauer³

¹ Albert-Ludwigs-Universität Freiburg, 3 Hermann-Herder-Strasse, D-79104 Freiburg, Germany

² Department of Physics and Astronomy, University of Sheffield, Hicks building, Hounsfield road, Sheffield, S37RH, UK

³ CERN, European Laboratory for Particle Physics, Geneva-23, CH-1211, Switzerland

Abstract. In the harsh environment of the Large Hadron Collider at CERN (design luminosity of $10^{34} \text{ cm}^{-2}\text{s}^{-1}$) efficient reconstruction of the signal primary vertex is crucial for many physics analyses. Described in this paper are primary vertex reconstruction strategies implemented in the ATLAS software framework Athena. The implementation of the algorithms follows a very modular design based on object oriented C++ and the use of abstract interfaces. This guarantees the easy use and exchange of different vertex fitters and finders which are considered for a given analysis. Such a modular approach relies on a dedicated Event Data Model for vertex reconstruction. The data model has been developed alongside the reconstruction algorithms. Its design is presented in detail. The performance of the implemented primary vertex reconstruction algorithms has been studied on a variety of Monte Carlo samples and results are presented.

1. Introduction

The Large Hadron Collider, which is currently under construction at CERN, is designed to collide protons with the energy of 14 TeV. During the initial years, the accelerator will operate at the luminosity of $2 \cdot 10^{33} \text{ cm}^{-2}\text{s}^{-1}$ (low luminosity mode), reaching $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ (high luminosity mode) at the later stages. The number of proton-proton interactions per bunch crossing is distributed according to the Poisson distribution with averages of 4.6 and 24 for low and high luminosity modes, respectively. Each signal event triggered and reconstructed in the ATLAS detector will thus be superimposed with several low- p_T proton-proton interactions, commonly denoted as minimum bias events.

The interaction region is described by a Gaussian with the standard deviation of 5.6 cm in the direction of the beam and $15 \mu\text{m}$ in the perpendicular plane. While for many physics analyses this knowledge is enough, for some it is not sufficient. The precise reconstruction of physics processes such as $H \rightarrow 4l$, $H \rightarrow \gamma\gamma$, identification of b - and τ -jets, reconstruction of exclusive b -decays and the measurement of lifetimes of long living particles require a precise knowledge of the primary vertex. It is therefore essential to reconstruct all the primary vertices in the bunch crossing and identify the one which is related to the signal process.

The primary vertex reconstruction framework implemented in the ATLAS Athena [1] environment allows the reconstruction of the primary vertices in both low and high luminosity regimes of the LHC. The positions of the primary proton-proton interactions are reconstructed and the signal collision is identified with high efficiency.

2. Reconstruction of Primary vertices

Compared to minimum bias events, signal events usually have a higher track multiplicity and transverse momentum. In some analyses however this is not true and this selection may introduce important biases. In Fig. 1 and 2 typical distributions of the Z -impact parameters of simulated charged tracks reconstructed in the ATLAS tracker are presented for two physics channels: $H(130) \rightarrow 4l$ and $H(130) \rightarrow \gamma\gamma$. It should be noted that in the case of $H(130) \rightarrow \gamma\gamma$ decay,

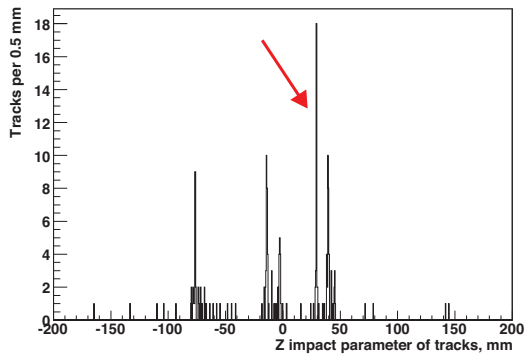


Figure 1. Distribution of the Z impact parameters of reconstructed tracks in a typical simulated $H(130) \rightarrow 4l$ event in the low luminosity regime at LHC. The cluster of tracks, corresponding to the signal primary vertex is marked with an arrow.

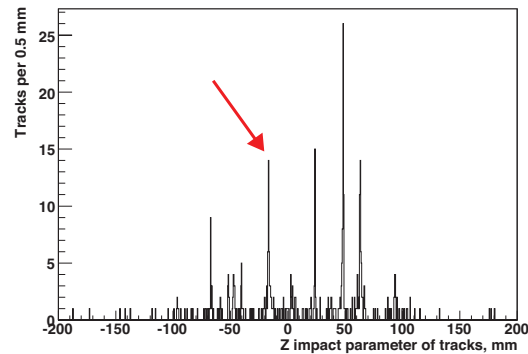


Figure 2. Distribution of the Z impact parameters of reconstructed tracks in a typical simulated $H(130) \rightarrow \gamma\gamma$ event in the low luminosity regime at LHC. The cluster of tracks, corresponding to the signal primary vertex is marked with an arrow.

the cluster with the highest track multiplicity does not correspond to the signal event. In this and similar cases it is essential to reconstruct all the primary vertices in a given bunch crossing. The selection of the signal can be done using some external information, specific to the type of the event, such as photons, leptons, jets and other observables.

The reconstruction of primary vertices can generally be subdivided in two stages:

- Primary vertex finding: association of reconstructed tracks to a particular vertex candidate.
- Vertex fitting: reconstruction of the actual vertex position and its covariance matrix, estimate of the quality of the fit, refit of the incident tracks.

It is evident that often these two stages are not distinguishable from each other. The algorithms exhibiting both the “fitting-after-finding” and “finding-through-fitting” approaches are implemented in the ATLAS Athena framework. Whatever the internal differences of these algorithms are, they are all implemented in a common user-friendly interface and based on the same Event Data Model.

3. Event Data Model

The Athena Event Data Model (EDM) for vertex reconstruction is designed using an object-oriented approach. It is composed of core classes, providing the basic vertexing objects, and further extensions of them, intended to be used for specific vertexing applications. The core EDM classes are:

- **Vertex:** Stores 3-vector of vertex position defined in the global ATLAS Detector reference frame.

- **RecVertex:** Inherits from the *Vertex* class. Stores covariance matrix, χ^2 and number of degrees of freedom of the vertex fit.
- **VxTrackAtVertex:** This class represents a track in a vertex fit. It contains a link to the original trajectory, reconstructed in the ATLAS Tracker, the original track parameters and the track parameters after refitting the track with the knowledge of the vertex. The access to the χ^2 , the number of degrees of freedom of the track refit and optionally to the track weight with respect to the vertex is also provided.
- **MVFXTrackAtVertex:** This class is an extension of the *VxTrackAtVertex*, which is specially designed to be used in the Multi Vertex Fit (see below). In addition to the usual track data, it provides detailed information about association of track to vertices.

Depending on the physics case, different representations of the reconstructed vertex candidates are used. All of them however inherit from the same *VxCandidate* class, which provides the basic information about a reconstructed vertex. The properties of these classes are the following:

- **VxCandidate:** Basic description of a reconstructed vertex. This class stores a *RecVertex* and a vector of *VxTrackAtVertex*'s, thus providing information on the vertex itself and on the tracks used in the vertex fit.
- **ExtendedVxCandidate:** An extension of the *VxCandidate*, which allows to store the full covariance matrix of the fit, including correlations between the parameters of the tracks fitted to the vertex.
- **MVFXCandidate:** Representation of a vertex candidate in a Multi Vertex Fit. A vector of *MVFXTrackAtVertex* objects is stored and in addition it contains information needed only during the multi vertex fit.

4. Common Interfaces

The common interface for the primary vertex reconstruction in ATLAS Athena was implemented using an object-oriented approach. All structural components of fitters and finders are implemented independently. An abstract interface is provided for each of these components, assuming each of them may have more than one implementation and making an easy exchange of tools between algorithms possible. Some of the core abstract base classes are:

- **IVertexFinder:** An abstract base class for the implementation of primary vertex finders. Concrete implementations analyze a track collection and return a set of reconstructed primary vertices (i.e. a vector of *VxCandidates*).
- **IVertexFitter:** An abstract base class for the implementation of vertex fitters. The fit method operates on a set of reconstructed tracks and, optionally, on a starting point for the fit and on a vertex constraint (typically the beam spot). A *VxCandidate* is returned.
- **IVertexSeedFinder:** An abstract base class for the algorithms estimating the starting point of the vertex fit, given a set of tracks. The starting point is returned using the *Vertex* class.
- **IVertexLinearizedTrackFactory:** A base class for storing the parameters of the linearized measurement equation, representing the dependence of the track parameters on the vertex position and on the track momentum at vertex.
- **IVertexUpdater:** An abstract base class for an iterative called tool, updating the vertex estimate with one track at the time. Concrete implementations allow to add or remove a single track to or from a *VxCandidate*.
- **IVertexSmoother:** An abstract base class for the implementation of algorithms which update the state parameters of all tracks fitted to a vertex with the knowledge of the vertex position.

- **IVertexTrackUpdater:** An abstract base class for the implementation of algorithms updating the state of a single track with the knowledge of the vertex position.

5. Primary Vertex Finders

As mentioned in the Sec. 2, both “fitting-after-finding” and “finding-through-fitting” approaches are implemented in the ATLAS offline software framework for primary vertex reconstruction. A good example of the first type of strategy is the *InDetPriVxFinder*. The reconstruction of primary vertices starts with a pre-selection of tracks compatible with the expected bunch-crossing region. The selected tracks are then ordered according to the value of their z -impact parameter and track clusters in z projection are searched for using a sliding window approach. The obtained clusters are regarded as independent primary vertex candidates. Each of these candidates is then reconstructed using one of the provided vertex fitters and cleaned iteratively for the outlying tracks. The χ^2 between the vertex estimate and the trajectory in question is calculated. The trajectories for which the probability of the χ^2 is less than 8 % are considered to be outliers. These trajectories are then rejected and the candidate is refitted. The procedure is repeated until no incompatible tracks are left or the cluster size becomes too small to continue.

In Fig. 3 a collaboration diagram, showing the work principle of the *InDetPriVxFinder* is presented. The use of abstract interfaces in the implementation is also shown.

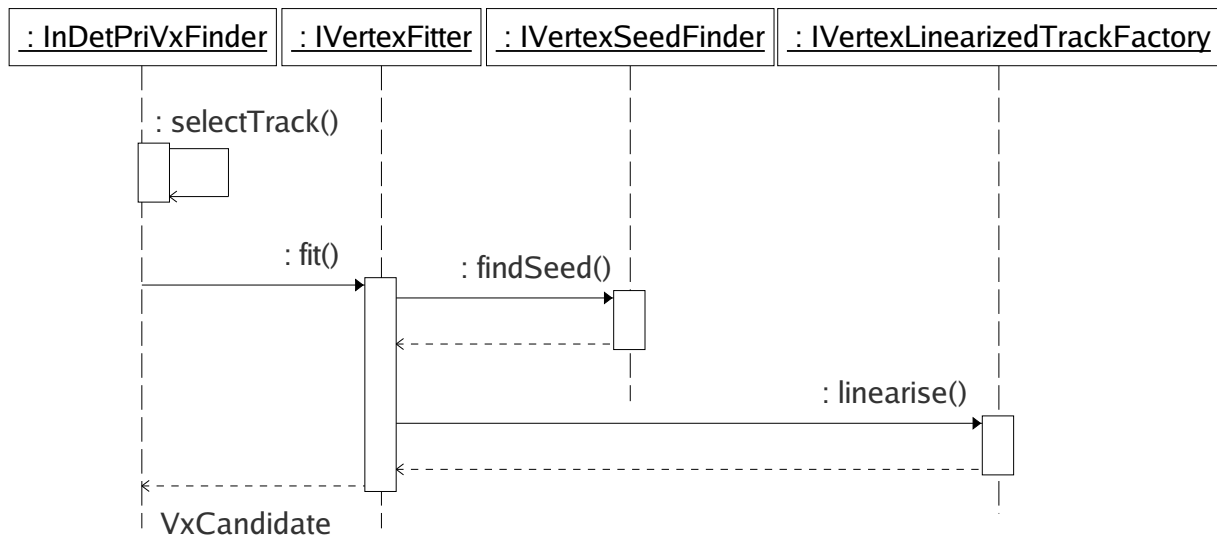


Figure 3. A collaboration diagram showing the work principle of the *InDetPriVxFinder* and the use of abstract interfaces in the implementation of the code.

An important property of the *InDetPriVxFinder* is that the number of reconstructed primary vertices is completely determined at the seeding stage. Moreover, once a track is rejected from a given vertex candidate, it is never used in any other cluster.

A better way of dealing with the outlying tracks is possible with the “finding-through-fitting” approach. An example of such a strategy implemented in the ATLAS Athena framework is the *InDetAdaptiveMultiPriVxFinder*. As in the case of the *InDetPriVxFinder*, the reconstruction starts with a pre-selection of tracks originating from the bunch crossing region. A single vertex seed is created out of this set of tracks. The vertex candidate is then reconstructed using the *AdaptiveMultiVertexFitter* [2]. The tracks which were considered to be outliers during the first fit are used to create a new vertex seed. At the next iteration a simultaneous fit of two vertices is performed. Each track is then down-weighted with respect to the two vertices. The number

of vertex candidates is growing iteration after iteration and the vertices are competing with each other in order to attain more tracks. An annealing procedure is applied to this process: the assignment of tracks to vertices is becoming harder with iterations. As a consequence, at the end of the fit all tracks which do not enter any vertex with a probability of χ^2 higher than 1 % have no influence on the fit result. The collaboration diagram, showing the principle of operation of the *InDetAdaptiveMultiPriVxFinder* is presented in Fig. 4.

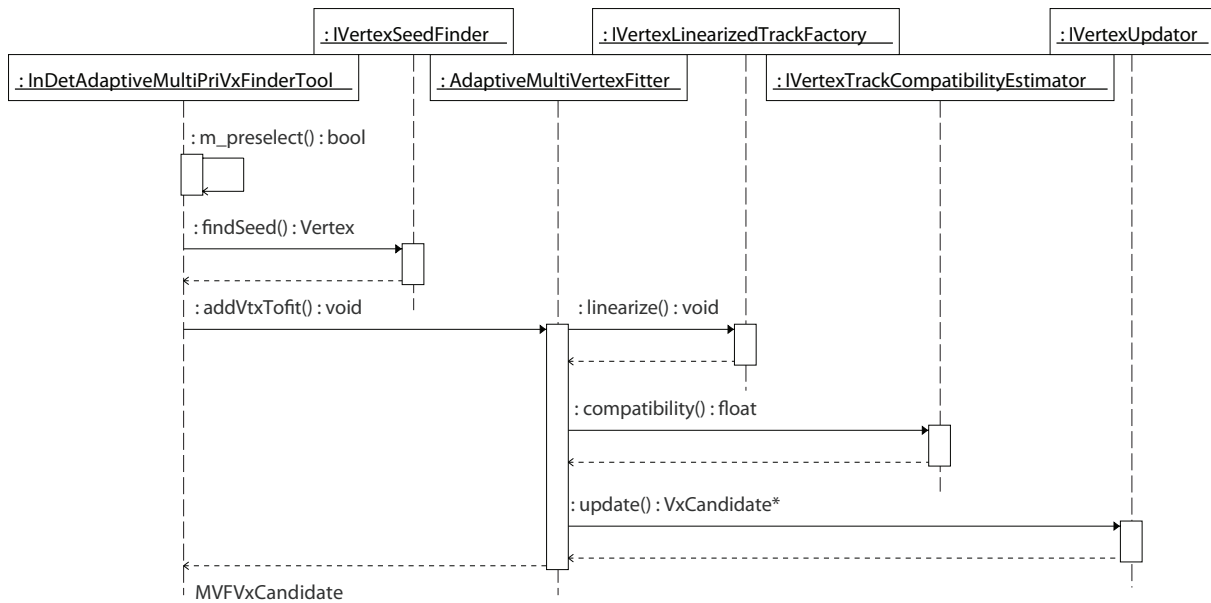


Figure 4. A simplified collaboration diagram showing the work principle of the Adaptive Multi Vertex Finder and the use of abstract interfaces in the implementation of the code. After track selection, iteratively a new vertex is seeded and the Adaptive Multi Vertex Fitter is called, which relies on concrete implementations of track linearization, fast estimation of the compatibility of the track to the vertex and Kalman update step.

6. Vertex Fitters

As mentioned above, the vertex fitting is treated separately from the vertex finding in the ATLAS offline software. The vertex fitters are therefore implemented as standalone tools with common abstract interfaces (see Sec. 4) and can be used for both primary and secondary vertex fitting. Several implementations, exhibiting different approaches to the problem are now discussed.

The first package for vertex fitting, which was implemented in ATLAS Athena was the *TrkVertexBilloirTools*. This package consists of two algorithms, which follow the approach proposed in the work of P.Billoir [3]. In both the Billoir *Fast* and *Full Vertex Fitter* the equations of motion of a charged particle in the magnetic field are approximated with their first-order Taylor expansions in terms of $\left(\frac{q}{p}\right)^1$.

The *Fast Vertex Fitter* drops any correlations of the vertex position with the track momenta, neglecting this dependence in the measurement equation. In contrary, the *Full Vertex Fitter*

¹ The parametrization of trajectories of charged particles in a magnetic field used in ATLAS Athena is in general similar to the perigee parametrization of P.Billoir [3]. However, several important differences, such as sign and angular range conventions, exist. Among others, the ratio of the charge of the particle to its full momentum $\left(\frac{q}{p}\right)$ is used instead of the transverse curvature.

takes these correlations into account, allowing to obtain the refitted parameters of tracks at the reconstructed vertex position.

A more modular approach to the vertex fitting is realized in the *SequentialVertexFitter*. This algorithm implements the conventional Kalman Filter for the vertex fitting as described in [4]. Contrary to the *TrkVertexBilloirTools*, the equations of motion are not approximated but the full analytical solution is used to compute Jacobians of the measurement equation. In addition, the *SequentialVertexSmoother*, which allows to refit the parameters of incident trajectories with the knowledge of the reconstructed vertex position is implemented as a separate tool.

A robust version of the above algorithm is the *AdaptiveVertexFitter*. It is an iterative re-weighted Kalman Vertex Fitter, where each track is down-weighted according to its compatibility to the actual vertex position [5]. The dependence of the weighting factor on the iteration number of the fit is determined by a thermodynamic annealing procedure. The assignment of tracks to a vertex candidate thus becomes stronger with iterations and the outliers are efficiently discarded.

7. Performance

Disclaimer: All the reconstruction efficiencies and coordinate resolutions presented below should be considered as *ATLAS preliminary* results which are intended to show the performance of the primary vertex reconstruction framework.

The performance of the ATLAS primary vertex reconstruction framework was tested with two types of Monte Carlo events: $t\bar{t}$ and $H(130) \rightarrow 4l$. The samples were digitized with low luminosity pile up using Athena version 12.0.6. A misaligned detector geometry with a realistic description of the distorted material distribution is used. In Figs. 5, 6, 7 and 8 the distributions of residuals of x and z coordinates of primary vertices reconstructed with *InDetAdaptiveMultiPriVxFinder* are presented. It can be seen that all the distributions are

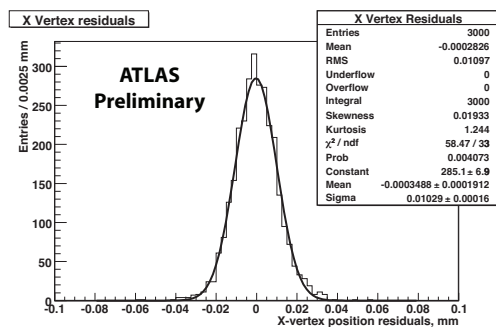


Figure 5. Distribution of residuals of the x coordinates of signal primary vertices reconstructed in $t\bar{t}$ events using the Adaptive Multi Primary Vertex Finder.

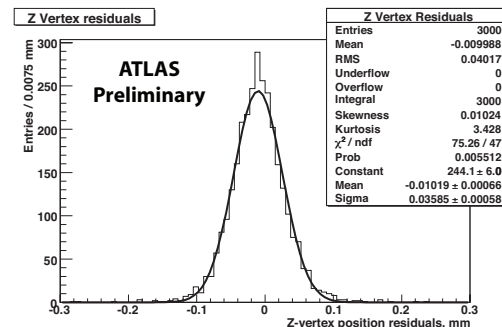


Figure 6. Distribution of residuals of the z coordinates of signal primary vertices reconstructed in $t\bar{t}$ events using the Adaptive Multi Primary Vertex Finder.

reasonably well approximated with a Gaussian function. The x -coordinate resolution (transverse to the beam axis) is about $10 \mu\text{m}$ for both channels² and the z -coordinate resolution (along the beam axis) is $36 \mu\text{m}$ to $40 \mu\text{m}$. The distributions of pulls were also studied for both channels and their Gaussian widths were found to be close to unity, indicating correct estimation of errors.

In Figs. 9, 10, 11 and 12 the distribution of residuals of x and z coordinates of primary vertices reconstructed with *InDetPriVxFinder* in combination with the *BilloirFastVertexFitter* are presented. It can be noted that while the coordinate resolutions in the transverse plane

² The standard deviation of a Gaussian fit to the distribution will be quoted as resolution hereafter.

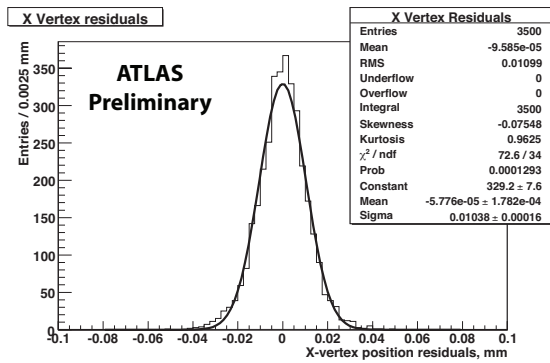


Figure 7. Distribution of residuals of the x coordinates of signal primary vertices reconstructed in $H(130) \rightarrow 4l$ events using the Adaptive Multi Primary Vertex Finder.

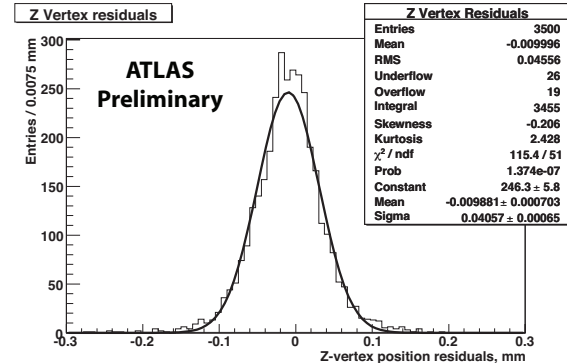


Figure 8. Distribution of residuals of the z coordinates of signal primary vertices reconstructed in $H(130) \rightarrow 4l$ events using the Adaptive Multi Primary Vertex Finder.

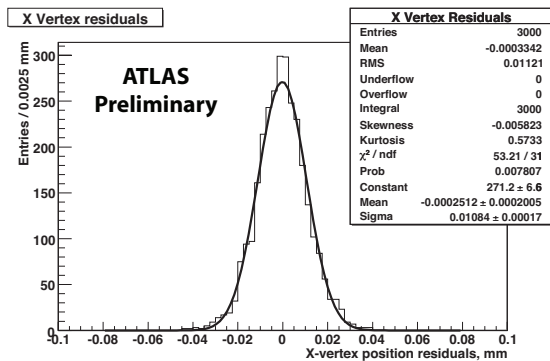


Figure 9. Distribution of residuals of the x coordinates of signal primary vertices reconstructed in $t\bar{t}$ events using the Inner Detector Primary Vertex Finder and the Fast Billoir Vertex Fitter.

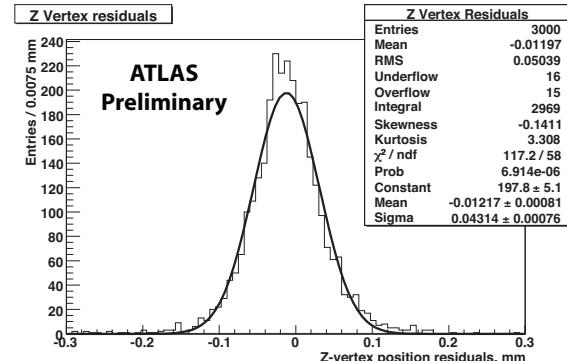


Figure 10. Distribution of residuals of the z coordinates of signal primary vertices reconstructed in $t\bar{t}$ events using the Inner Detector Primary Vertex Finder and the Fast Billoir Vertex Fitter.

are similar to those obtained with the *InDetAdaptiveMultiPriVxFinder*, the resolution in the longitudinal direction is degraded by several microns. This can be explained partly by the mathematical assumptions made in the *BilloirFastVertexFitter* and partly by the precision of the *InDetPriVxFinder* discussed in Sec. 5.

In Tab. 1 the efficiencies of the primary vertex finding with the two discussed approaches are presented. The signal primary vertex is considered to be correctly reconstructed if it is located within 100 μm from the simulated signal primary vertex.

As expected, the performance of the *InDetPriVxFinder* combined with *BilloirFastVertexFitter* is lower than in the case of *InDetAdaptiveMultiPriVxFinder*.

8. Conclusions

The primary vertex reconstruction framework is implemented in the ATLAS offline software. The framework is capable of incorporating different approaches to primary vertex finding and

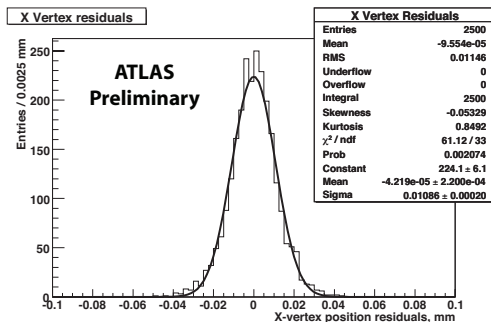


Figure 11. Distribution of residuals of the x coordinates of signal primary vertices reconstructed in $H(130) \rightarrow 4l$ events using the Inner Detector Primary Vertex Finder and the Fast Billoir Vertex Fitter.

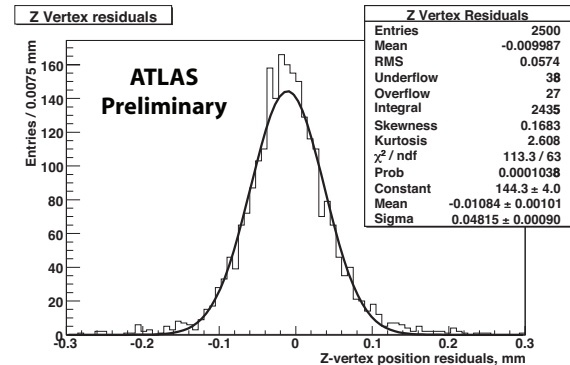


Figure 12. Distribution of residuals of the z coordinates of signal primary vertices reconstructed in $H(130) \rightarrow 4l$ events using the Inner Detector Primary Vertex Finder and the Fast Billoir Vertex Fitter.

Table 1. Efficiencies of primary vertex reconstruction with *InDetPriVxFinder* and *InDetAdaptiveMultiPriVxFinder* for $t\bar{t}$ and $H(130) \rightarrow 4l$ channels.

Channel	<i>InDetPriVxFinder</i>	<i>InDetAdaptiveMultiPriVxFinder</i>
$t\bar{t}$	$92.6 \pm 1.8 \%$	$96.9 \pm 1.8 \%$
$H(130) \rightarrow 4l$	$88.2 \pm 1.8 \%$	$94.1 \pm 1.6 \%$

vertex fitting due to the implementation of a common Event Data Model and common abstract interfaces. The interfaces are made sufficiently modular so that the different tools can easily be exchanged between different algorithms. The test of the framework was performed using Monte Carlo events passed through the full simulation of the ATLAS Detector, showing an excellent coordinate resolution and a high reconstruction efficiency.

Acknowledgments

We would like to express our gratitude to the ARTEMIS Research Training Network for the financial support of this study. The participation in the CHEP 2007 conference was partly funded from the EU FP6 Research Training Network grant MRTN-CT-2006-035657 Artemis. We would like to thank also Gareth Brown from the University of Manchester for his substantial help during the preparation of the conference presentation and the current proceedings.

References

- [1] The ATLAS Collaboration **URL:** <https://twiki.cern.ch/twiki/bin/view/Atlas/WorkBook>
- [2] R Frühwirth, W Waltenberger proc. CHEP 2004, Interlaken, Switzerland.
- [3] P Billoir, S Qian Nucl. Ins. and Meth. in Phys. Res. A311(1992) 139-150
- [4] R Frühwirth, Nucl. Ins. and Meth. 225(1984) 352
- [5] R Frühwirth et al. Nucl. Ins. and Meth. in Phys. Res. A502(2003) 699