Representing Misalignments of the STAR Geometry Model using AgML

Jason C Webb, Jérôme Lauret, Victor Perevotchikov, Dmitri Smirnov and Gene Van Buren
Brookhaven National Laboratory
E-mail: jwebb@bnl.gov

Abstract.
The STAR Heavy Flavor Tracker (HFT) was designed to provide high-precision tracking for the identification of charmed hadron decays in heavy-ion collisions at RHIC. It consists of three independently mounted subsystems, providing four precision measurements along the track trajectory, with the goal of pointing decay daughters back to vertices displaced by less than 100 microns from the primary event vertex. The ultimate efficiency and resolution of the physics analysis will be driven by the quality of the simulation and reconstruction of events in heavy-ion collisions. In particular, it is important that the geometry model properly accounts for the relative misalignments of the HFT subsystems, along with the alignment of the HFT relative to STARs primary tracking detector, the Time Projection Chamber (TPC).

The Abstract Geometry Modeling Language (AgML) provides a single description of the STAR geometry, generating both our simulation (GEANT 3) and reconstruction geometries (ROOT). AgML implements an ideal detector model, while misalignments are stored separately in database tables. These have historically been applied at the hit level. Simulated detector hits are projected from their ideal position along the track's trajectory, until they intersect the misaligned detector volume, where the struck detector element is calculated for hit digitization. This scheme has worked well as hit errors have been negligible compared with the size of sensitive volumes. The precision and complexity of the HFT detector require us to apply misalignments to the detector volumes themselves. In this paper we summarize the extension of the AgML language and support libraries to enable the static misalignment of our reconstruction and simulation geometries, discussing the design goals, limitations and path to full misalignment support in ROOT/VMC-based simulation.

1. Introduction
Charm quarks are created through hard scattering early in heavy-ion collisions due to its diminished cross section for any secondary scattering. They experience the full evolution of the system, and can therefore provide important insights into whether the heavy quarks interact differently with the medium than the light quarks. Experiments, such as the Solenoidal Tracker at RHIC (STAR), are thus motivated to identify open-charm hadrons to address this important question. These hadrons cannot be detected directly, due to their short decay lengths. They must be reconstructed from their decay products, such as $D^0 \rightarrow K^+\pi^-$. Combinatoric backgrounds pose significant challenges to these measurements, as illustrated in figure 1. In order to achieve the required background suppression in the high multiplicity environment, topological reconstruction, where the decay products are projected to a vertex displaced less than 100 $\mu$m
from the primary collision, must be employed. This requires precision hardware capable of \( \sim 50 \mu m \) or better pointing resolution to the primary vertex. It also requires precision software, capable of reconstructing tracks with the required resolution and simulating the deviations on that order from the ideal geometry of the detector model. The requirements on our track reconstruction software are discussed in a separate contribution to this proceedings[1]. In this paper we summarize the effort to equip simulation in STAR to satisfy the misalignment needs of the heavy flavor program.

![Figure 1](image1.png)

**Figure 1.** Invariant mass of \( K\pi \) pairs, topologically reconstructed in 200 GeV AuAu collisions. The HFT suppresses combinatoric backgrounds by over four orders of magnitude (inset).

2. **The STAR Detector**

For the 2014-2016 physics runs at RHIC, the STAR experiment added four precision layers of silicon tracking to supplement the Time Projection Chamber (TPC)[2] and achieve the necessary resolution for the heavy flavor program. The layout of the detector is illustrated in figure 2. The HFT [3] consists of three subsystems of silicon detectors, providing four layers of increasing precision as we dive from the TPC (\( \sim 1 \) mm hit resolution) to the vertex. These include: (1) The Silicon Strip Detector (SSD) [4]: fast, double-sided (stereoscopic) silicon strips at radii of \( \approx 22 \) cm; (2) Intermediate Silicon Tracker (IST): fast, single-sided silicon pads at radii of \( \approx 14 \) cm; and (3) the Pixel Detector (PXL) [5]: two layers of \( 20.7 \times 20.7\mu m \) Monolithic Active Pixel Sensors (MAPS) at radii of \( \approx 8 \) cm and \( \approx 2.8 \) cm.

![Figure 2](image2.png)

**Figure 2.** The STAR detector includes four major subsystems dedicated to tracking, with increasing precision as one dives in towards the beamline.
3. Misalignment

3.1. Hit-Level Misalignment

Historically, STAR has accounted for misalignments at the hit digitization stage, rather than by moving the physical volumes during the simulation. In this scheme, hits are generated by GEANT 3[6] saving both the hit and track parameters with respect to the ideal position of the detector volume. The helical track model is evaluated at this point, and used to project from the ideal hit position along the trajectory of the track until it intercepts the misaligned detector. The struck detector element is then updated, based on the track’s projection to the misaligned position. In the event that the track projects outside of the acceptance of the detector, the hit is dropped from further consideration. Otherwise, it is assigned to the detector element crossed at the misaligned detector position. This method has performed adequately during the TPC-era with comparatively large millimeter scale hit resolution, but suffers from notable deficiencies as we approach the 10’s of micron scale resolution of the HFT. Typical misalignments are on a scale of significance compared to the size of the active sensors. This introduces an inefficiency, as tracks which miss the ideally aligned detector do not leave a hit in the sensor, even if the track passes through its true misaligned position. Additionally, for those tracks which are in the acceptance of both the ideal and misaligned geometry, energy loss and multiple scattering will be applied at the ideal position. This alters the trajectory of the track, potentially changing the misaligned hit location by a significant amount. The end result is that this technique severely underestimates tracking efficiency in the HFT (see figure 3), which undermines the intended physics program.

![Figure 3](image)

**Figure 3.** Ratio of HFT tracking efficiency to TPC tracking efficiency, for ideal simulations (red) and misaligned simulations using digitization stage misalignment (blue).

3.2. Design and Implementation

The HFT physics program requires simulations which account for the many small misalignments of the active layers in order to reach the ultimate accuracy needed in determining its efficiency. Thus, we need a framework in which we can associate the numerous misalignment parameters, stored in the STAR database, with their corresponding volumes in the geometry hierarchy. This framework needs to support both our existing GEANT 3 simulation package starsim, and our reconstruction codes (and any future VMC application) which utilize the ROOT/TGeo geometry package[7]. Both the GEANT 3 and ROOT geometry models in STAR are compiled from a single source code, implemented in the Abstract Geometry Markup Language (AgML)[8]. Extending the existing AgML framework to support applying misalignment parameters was the logical path forward.
In both GEANT 3 and ROOT, the placement and orientation of a daughter volume with respect to its mother volume can be described by a general transformation matrix:

\[
T = \begin{bmatrix}
  r_{00} & r_{01} & r_{02} & t_0 \\
  r_{10} & r_{11} & r_{12} & t_1 \\
  r_{20} & r_{21} & r_{22} & t_2 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]

To transform from the local coordinates of a daughter volume \(x_{\text{daughter}}\) to the global coordinates \(x_{\text{global}}\) consists of multiplying out all of the transformation matrices in the path of the daughter volume:

\[
x_{\text{global}} = T_N \cdots T_k \cdots T_1 x_{\text{daughter}} \tag{1}
\]

We are interested in introducing misalignments to the geometry to one, or more, volumes in a given path. For example, we want to be able to add a translation and rotation to the ideal position of the volume \(k\), and all of its descendants, relative to its position in its mother volume – changing the position and orientation of both the dead material and active sensors in simulation. This is performed by inserting an additional transformation matrix \(M_k\), before the transformation of the \(k\) volume.

\[
x_{\text{global}} = T_N \cdots M_k T_k \cdots T_1 x_{\text{daughter}} \tag{2}
\]

Essentially, we can misalign a volume in the simulation with respect to its mother volume by multiplying the original transformation matrix by a misalignment matrix \(M_k\) on the left. Another use case is when we want to misalign a volume (and its descendants) with respect to its ideal position within the mother volume. In other words, we want to apply an additional translation/rotation after it has been placed in the mother volume. This is achieved by multiplying the transformation by the misalignment matrix on the right:

\[
x_{\text{global}} = T_N \cdots T_k M_k \cdots T_1 x_{\text{daughter}}. \tag{3}
\]

The AgML[8] language provides a rich syntax for placing volumes, which we have extended to support the application of misalignment matrices either before or after the ideal placement of the volume. A \texttt{Placement} block declares which volume is placed in which mother volume, fully specifying the transformation matrix. This is done by applying a series of user-defined rotations and translations, in sequence, to a transformation matrix held in the AgML support library. This functionality was extended to support misalignments by the addition of a \texttt{Misalign} command to the AgML syntax, and the addition of an interface to the STAR database in the concrete geometry support library. The \texttt{Misalign} tag specifies the name and row of a database table, enabling the independent misalignment of different copies of each detector volume. Additionally, the misalignment tag allows the user to specify whether the misalignment occurs relative to the ideal position of the volume (right multiplication) or relative to the position of the mother volume (left multiplication). This is illustrated in the listing of figure 4, which describes the placement of the pixel sectors PXLA within the main pixel mother volume PXMO. The misalignment tags specify two tables to be applied. The \texttt{pxlSectorOnHalf} table is applied to the sectors relative to their ideal orientation, i.e. after the rotation and translation operations have been performed. The \texttt{pxlHalfOnPxl} table is applied with the group option, which performs misalignments relative to the mother volumes orientation (left multiplication). This permits a group of co-moving detector elements to have a common misalignment applied, relative to the orientation of their mother volume.
<!-- Place pixel sectors in pixel mother -->
<Placement volume="PXLA" in="PXMO" konly="MANY"
    x="xpos" y="ypos" z="zpos"
    ncopy="sector"/>

<Rotation alphaz="alpha_z"/>
<Rotation alphax="alpha_x"/>
<Rotation alphay="alpha_y"/>

<!-- Ideal rotation after translation -->
<Placement volume="PXLA" in="PXMO" konly="MANY"
    x="xpos" y="ypos" z="zpos"
    ncopy="sector"/>

<!-- Ideal rotation after translation -->
<Rotation alphaz="alpha_z"/>
<Rotation alphax="alpha_x"/>
<Rotation alphay="alpha_y"/>

<!-- Misalignment sectors 1-5 and 6-10 relative to mother volume -->
<Misalign table="Geometry/pxl/pxlHalfOnPx1"
    row="(sector-1)/5" opts="group" />

<!-- Misalignments of each sector relative to its ideal position -->
<Misalign table="Geometry/pxl/pxlSectorOnHalf"
    row="sector-1" />

</Placement>

Figure 4. At left, the AgML code responsible for placing pixel sectors within the top-level pixel mother volume. At right, the AgML source code is compiled into a geometry construction library which steers the AgML support library’s creation of the concrete geometry model. Misalignment tables in the database are applied by the support library.

4. Proof-of-principle
In order to test the misalignment framework, a test geometry for the pixel detector was created, enabling misalignments of the pixel sectors as noted in the listing of figure 4. A set of misalignment parameters were applied to each of the pixel sectors, and cosmic ray muons were simulated with the STAR magnetic field off. Then, the standard internal alignment procedure, developed to extract the true positions of the sectors and ladders of the pixel detector, was applied to the simulated data. The internal alignment process is an iterative procedure. Track candidates are formed from pairs of points in two layers of a PXL sector, and a straight line is extrapolated to one of the three sectors on the opposite side of the beamline. Residuals between the extrapolated track and hits in the opposing sectors are accumulated. From these residuals, a set of displacements and rotations are calculated and applied to the reconstruction code for the next iteration (holding a reference sector, sector one, fixed). The procedure is repeated, applying the displacements and rotations from the previous iteration, until the residuals converge to a minimum value. Figure 5 shows the hit residuals in sector 6 for the local-x coordinate compared with a straight-line extrapolation of hits in sector 2 for the first (left) and last (right) iteration of the internal alignment procedure. After ≈ 10 iterations, the procedure is observed to converge to the correct positions for all misaligned sectors. Figure 6 shows the final residuals for all sectors in the local-x, -y and -z coordinates. Residual widths are comparable to those observed in real cosmic ray events.

5. Summary and Path Forward
In order to achieve the full potential of the HFT physics program, simulations must be performed using geometry models which closely reflect the true positioning of the detector elements. The AgML language has been extended to support the misalignment of the HFT detector volumes in STAR simulations, and can be applied to other STAR subsystems, such as the TPC, Time-
Figure 5. Residuals in sector 6 with respect to track segments identified in sector 2. Left plot shows the residuals in the x coordinate at the first iteration in the inner and outer pixel layers. The right shows the same for the final iteration. The x-axis ranges from -1 to +1 cm on the left, and -0.2 to +0.2 cm on the right.

Figure 6. Resolution achieved in all sectors of the pixel detector, indicating the precision with which the internal alignment procedure converges to the input values of the detector segments. The framework connects the ideal geometry model, stored in the AgML source code, to the misalignment tables stored in the database. This ensures that a single geometry implementation can be used to describe varying conditions in the experimental hall. The code supports our existing GEANT 3 based simulations, allowing for the misalignment of pixel sectors, and the IST and SST ladder assemblies. The code is currently under review and testing, using simulated cosmic rays in a misaligned detector geometry and applying the standard HFT alignment software to the data. For determining efficiencies, the misalignment framework will allow simulations embedded into real data to replace the less robust data-driven methods which has been employed in the interim, enabling precise physics
measurements from a precision detector.

Acknowledgments
This work was supported by the Office of Nuclear Physics within the U.S. Department of Energy’s Office of Science.

References
[1] Webb J C et al. STAR Reconstruction Improvements for Tracking with the Heavy Flavor Tracker, these proceedings.
[2] Anderson M et al. 2003 Nucl. Instr. and Meth. A 499 699.
[3] Qiu H, et al. 2014 Nucl. Phys. A 931 1141-1146.
[4] Arnold L et al. 2003 Nucl. Instr. and Meth. A 499 652.
[5] Greiner L et al. 2011 Nucl. Instr. and Meth. A 650 68.
[6] Brun R, Bruyant F, Maire M, McPherson A C and Zanarini P, CERN-DD-EE-84-1.
[7] Brun R, Gheata A and Gheata M 2003 Nucl.Instrum. Meth. A 502, 676.
[8] Webb J C et al. 2012 J. Phys. Conf. Ser. 396 022058.