Offline Software Update

Daniel Brandenburg

Monday, April 26, 2021

Updates to BFC

- Forward Tracking:
  - `FwdTrack` : runs forward tracking, looks for config in local dir
- Fast Simulators:
  - `FstFastSim` - Runs the Forward Silicon Tracker Fast Simulator
  - `FttFastSim` - Runs the Forward sTGC Tracker Fast Simulator
- Slow Simulators (Skeletons)
  - `FstSlowSim` - Runs the Forward Silicon Tracker Fast Simulator (NOT in yet)
  - `FttSlowSim` - Runs the Forward sTGC Tracker Fast Simulator (NOT in yet)
- sTGC Offline chain:
  - `FttDat` - runs the raw data reader, produced/fills StEvent collection
  - `FttQA` - runs the QA Maker for the sTGC offline chain

Running Tracking code in BFC

1. Make/copy a config.xml (into current dir)

2. Run on one Pythia file:

```
root4star -l 'bfc.C(
        10, // number of events
        "fzin dev2021 StEvent evout geantout ReverseField agml usexgeom bigbig
FwdTrack", // chain options
        "/gpfs01/star/pwg/youqi/runPythia/out/5.fzd" // input (simulation) file
)'
```

NOTE: Still waiting for these updates in DEV, should be in by end of the week, need to send updates to Gene

Running sTGC Offline code in BFC

1. Checkout code: https://github.com/jdbrice/stgc-offline.git
2. `cd stgc-offline`, build with `cons`
3. Run on one DAQ file:

```
root4star -b -q -l
       'bfc.C(
              5000,
              "in, db, FttDat, FttQA",
              "stgc_2021_03_31_05_34.daq"
       )'
```

This runs just the raw data reader and QA maker, others will

FST Geometry and Misalignment

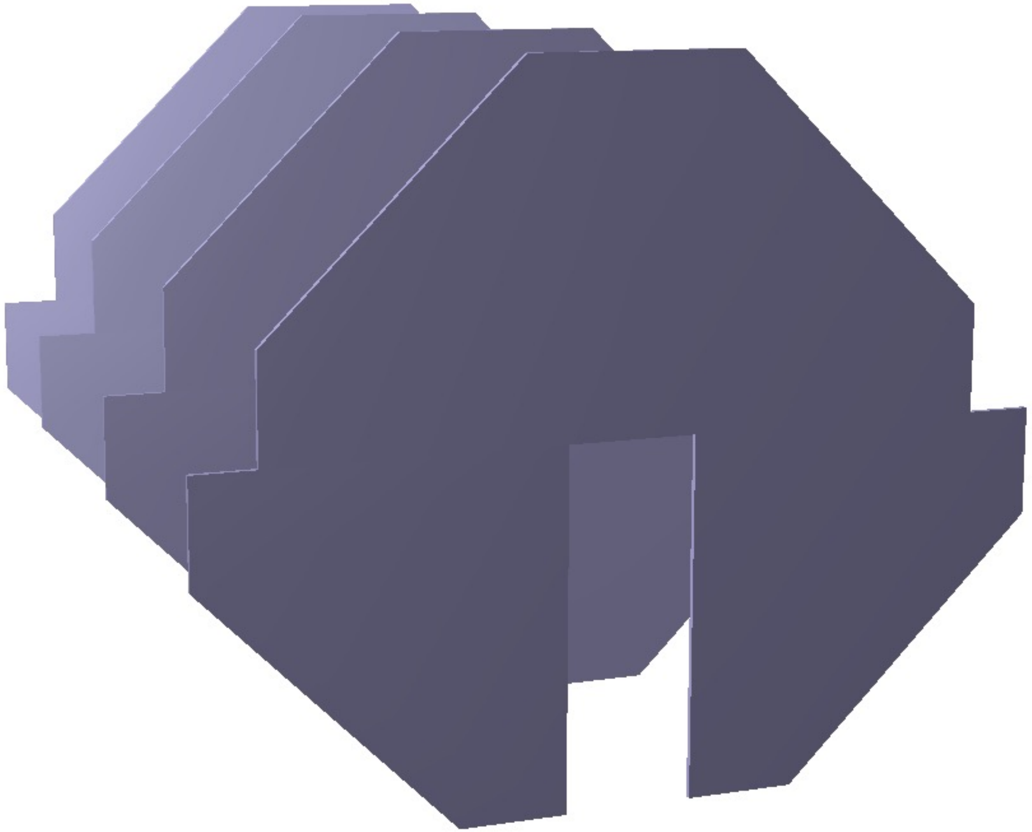
- Like HFT, we need to prepare FST geometry for misalignment:
  - Geometry must be re-organized (to allow unique sensor, unique mis-alignment)
  - TODO: Setup misalignment tables
  - Examples (from PXL) in: `$STAR/StarDb/Geometry/pxl/*.C`

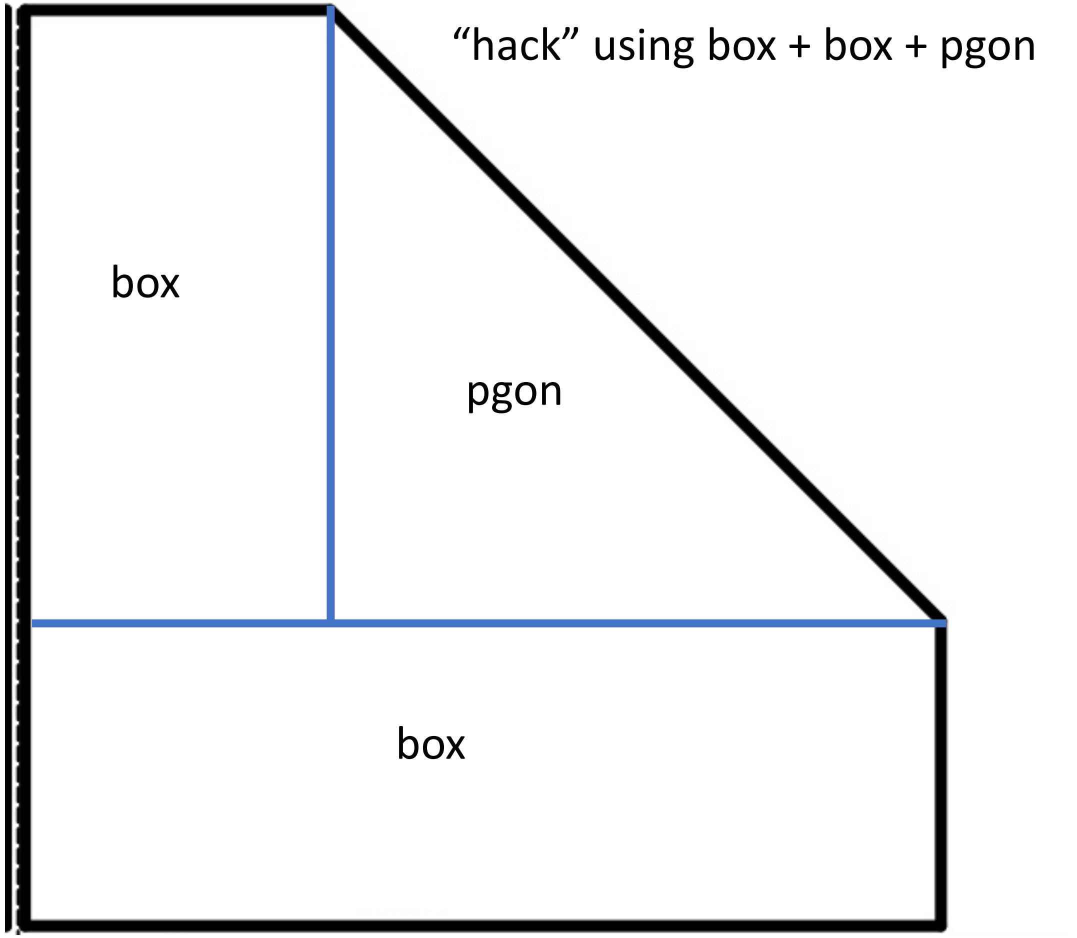
- Plan for Geometry
- Freeze existing geometry as a reference
- Setup 2 new FST geometries
  1. Re-organized full detail geometry (for misalignment)
  2. Simplified geometry for GENFIT

sTGC Geometry

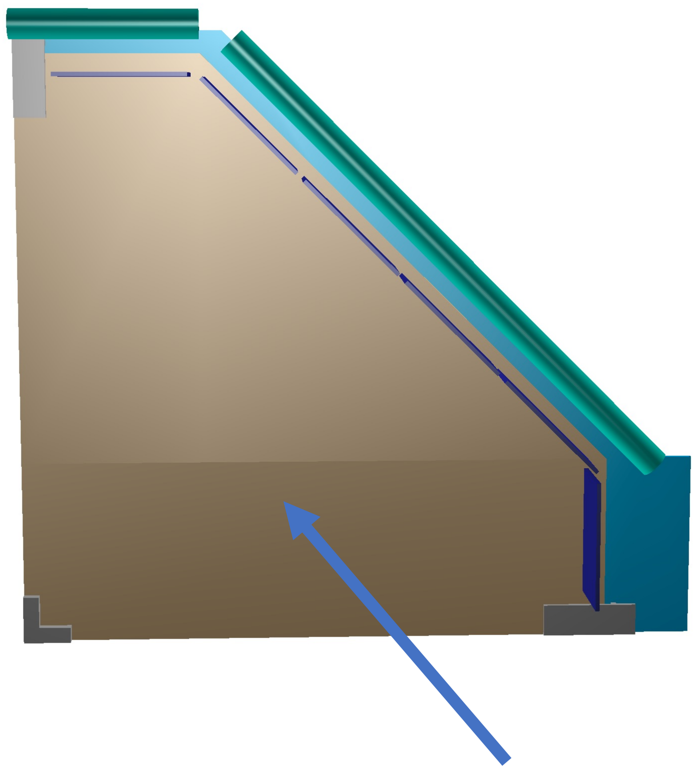
- Last f2f meeting: ROOT pentagon geometry using `TGeoXTru`

sTGC geometry using TGeoXtru in ROOT

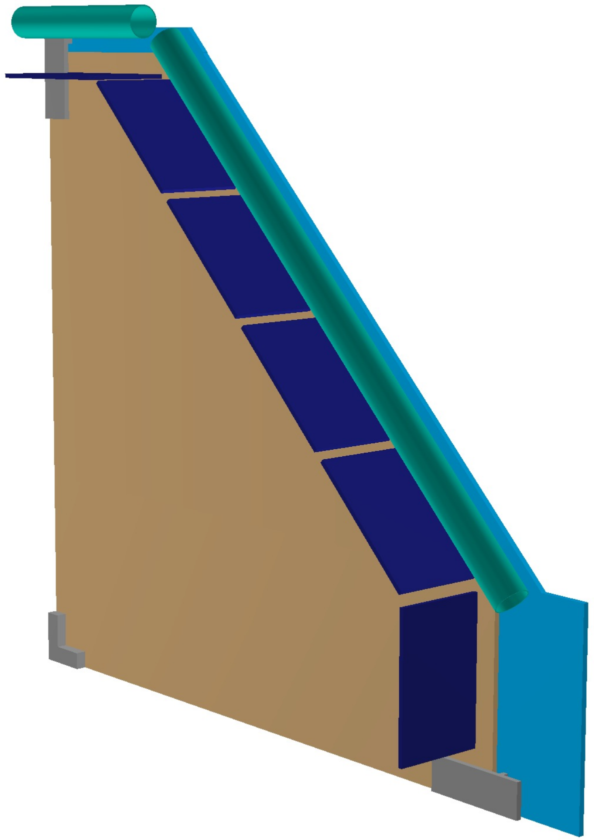




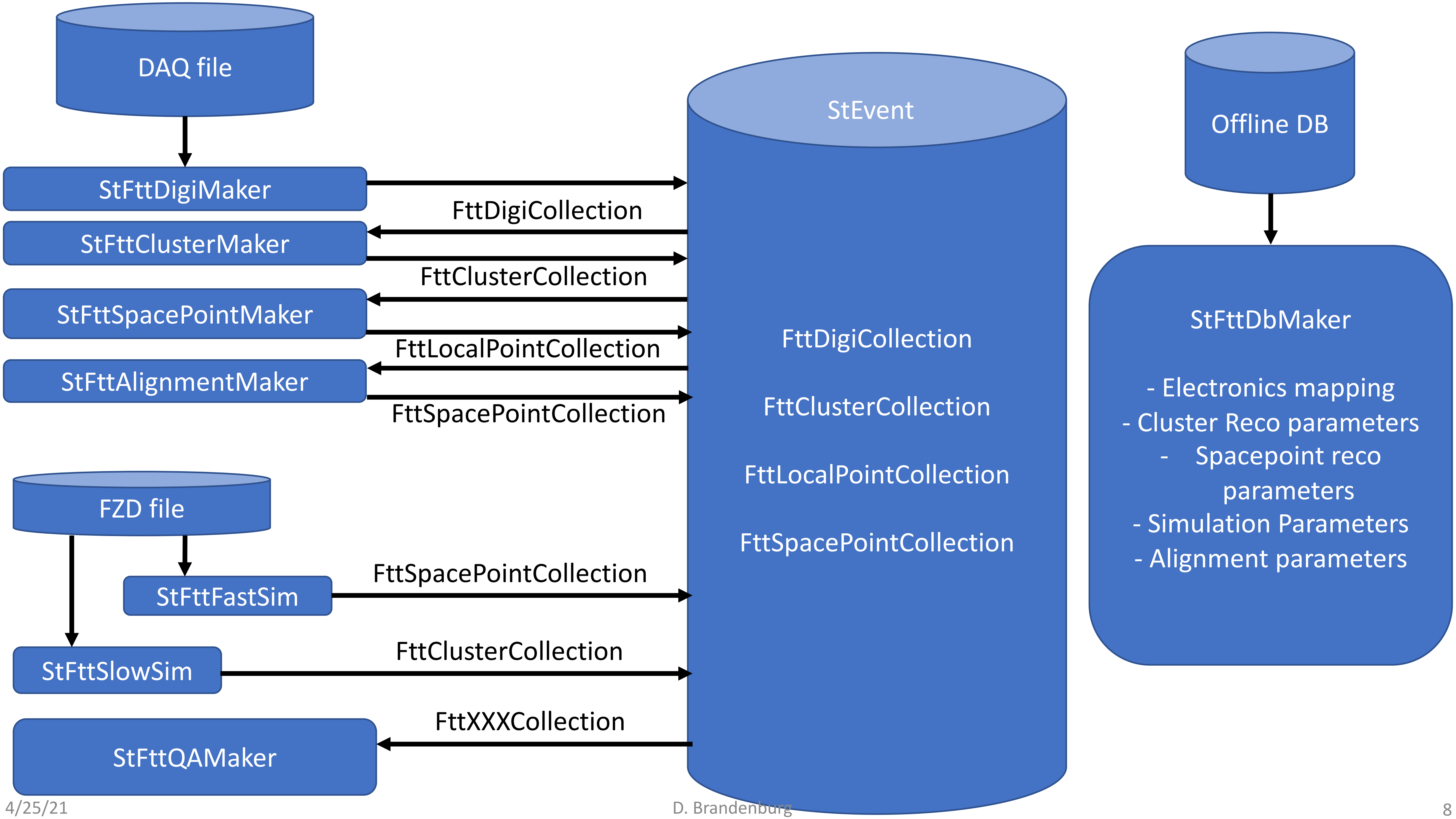
Update ROOT sTGC Geometry



Edge between box & pgon



- ROOT TGeo geometry
- Uses box + box + pgon for pentagonal shape
- Jason will convert to AgML
- Doesn't expect box+pgon edge to be issue. Will add tests to check for this
- Nicole (BNL post-doc) helping with sTGC geometry
- We are making a very simple geometry for GENFIT, will test if needed for sTGC


DAQ file
StFttDigiMaker
FttDigiCollection
StFttClusterMaker
FttClusterCollection
StFttSpacePointMaker
FttLocalPointCollection
StFttAlignmentMaker
FttSpacePointCollection
FZD file
StFttFastSim
FttSpacePointCollection
StFttSlowSim
FttClusterCollection
StFttQAMaker
FttXXXCollection
StEvent
FttDigiCollection
FttClusterCollection
FttLocalPointCollection
FttSpacePointCollection
Offline DB
StFttDbMaker
- Electronics mapping
- Cluster Reco parameters
- Spacepoint reco parameters
- Simulation Parameters
- Alignment parameters

sTGC Offline Chain

LEGEND:
DONE
SKELETON (WIP)
TODO

- FttDigiMaker
  - Read DAQ files, unpack raw data, write to StEvent::FttDigiCollection
- FttClusterMaker
  - Read raw data, apply channel mapping
  - Run clustering algorithm
  - Write clusters to StEvent::FttClusterCollection
- FttSpacePointMaker
  - Read in clusters, make 2D according to X,Y,U strip mapping
  - Optionally, apply ghost hit rejection
  - Write Space Points (+Cov matrix) to StEvent for tracking
- FttFastSim
  - Directly simulate 2D space points from GEANT input
- FttSlowSim
  - Simulate electronic level signals – write into a format that is compatible with cluster finder (need to decide about mapping)
- FttDbMaker
  - Read from the DB tables that store sTGC related settings
  - Will be filled in as we finish other makers

- FttAlignmentMaker
  - Map the local hits to global (STAR) coordinate system
  - Apply any transformations (rotations, translation, deformations)
- FttQaMaker
  - Create detailed QA Histograms for all steps
  - Similar to the online plots, but allow QA at all levels (Cluster, space point, reco)
  - Online plot level QA now
- StEvent structures
  - FttDigiCollection
  - FttClusterCollection
  - FttSpacePointCollection

sTGC Database makers

- Interface with database is used for “parameters”
  - Allows run-by-run or Run (Lik2 Run20…) based parameters
- Needed for:
  - Electronic mapping
  - Cluster reco parameters
  - Spacepoint reco parameters
  - Ghost hit rejection parameters
  - ….
- Manpower : 2 weeks from HepDB shifters
- Goal:

1. Discuss needed parameters (start with electronic mapping, more tomorrow)
2. Implement classes and debug (read parameters from local files)
3. Depending on progress, setup DB structure with Dmitry

Vertex Finding + Fitting

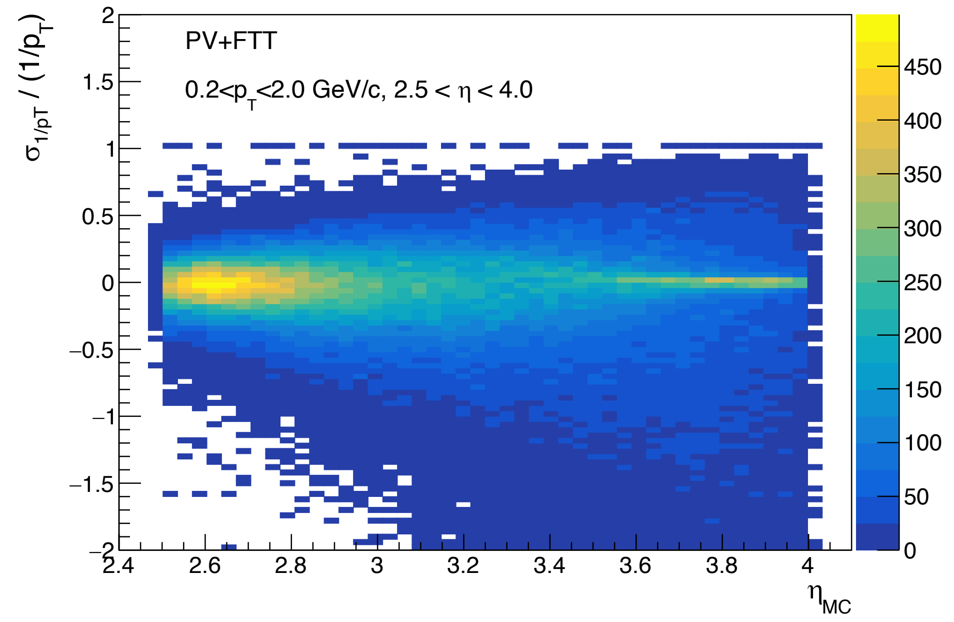
- **RAVE - an Open, Extensible, Detector-Independent Toolkit for Reconstruction of Interaction Vertices**
- W. Waltenberger and F. Moser, *2006 IEEE Nuclear Science Symposium Conference Record*, 2006, pp. 104-109, doi: 10.1109/NSSMIC.2006.356117.

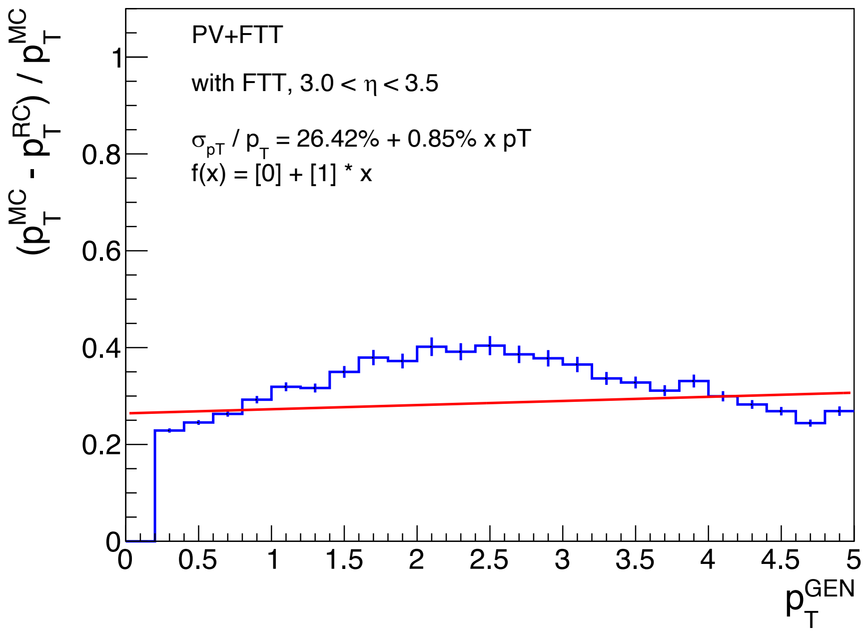
- http://www.hephy.at/project/ilc/talks/07_ALCPG_Fnal/Mitaroff_rave.pdf

- Was hoping to show first plots, but still need some work.
- Will help to have installed on RCF – working with Dmitry S on this.
- Also need a rebuild of GENFIT once RAVE is installed

Tracking Mysteries

- In past f3f meeting I showed (also shown by gavin)
- Tracking with FTT only showed strange “improvement” in resolution at high pT
- Strong eta dependence
- At that time I wasn’t sure what was causing this





Tracking with FTT only

- Both Gavin and I used the `fast_track.xml` config with

```xml
<TrackFitter refitSi="false" mcSeed="true"  >
    <Vertex sigmaXY="0.05" sigmaZ="5.0" includeInFit="true" />
</TrackFitter>
```

- `mcSeed="true"` - use the MC pT (blurred) as seed for fitting
- I implemented this to test GENFIT (speed, etc.) with ideal settings + good starting place
- ISSUE : sometimes seed fit χ^2 etc. was already good enough to "converge" and finish fit – since space points have little constraining power
- In these cases a "real fit" without MC seed would fail

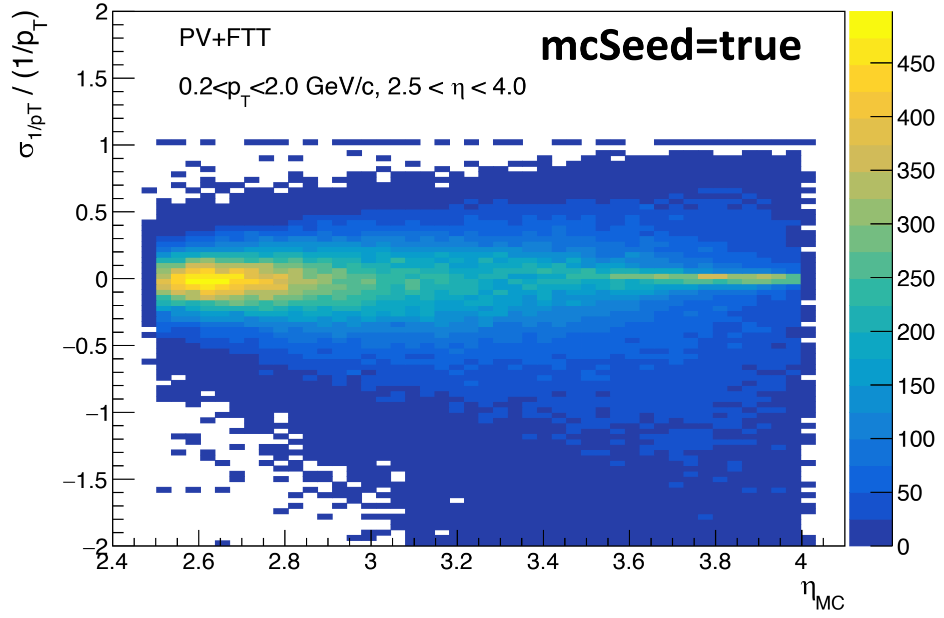
Tracking with FTT only

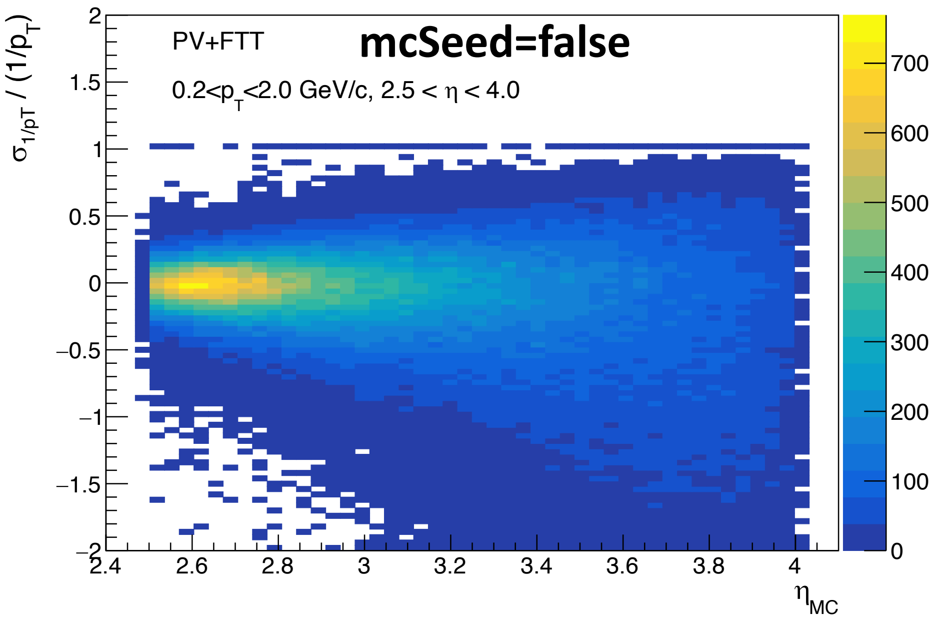
- Simple change to the `fast_track.xml` config with

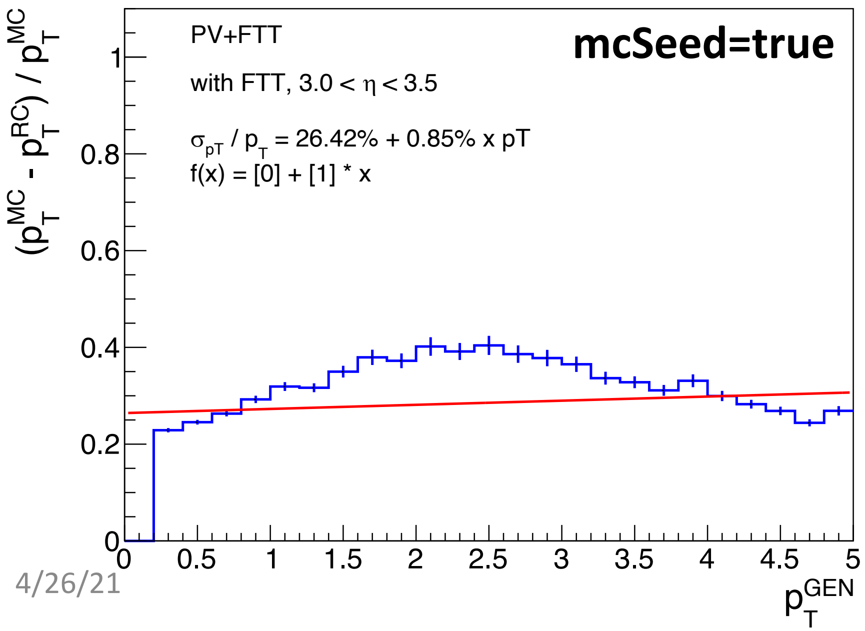
```
<TrackFitter refitSi="false" mcSeed="false"  >
    <Vertex sigmaXY="0.05" sigmaZ="5.0" includeInFit="true" />
</TrackFitter>
```

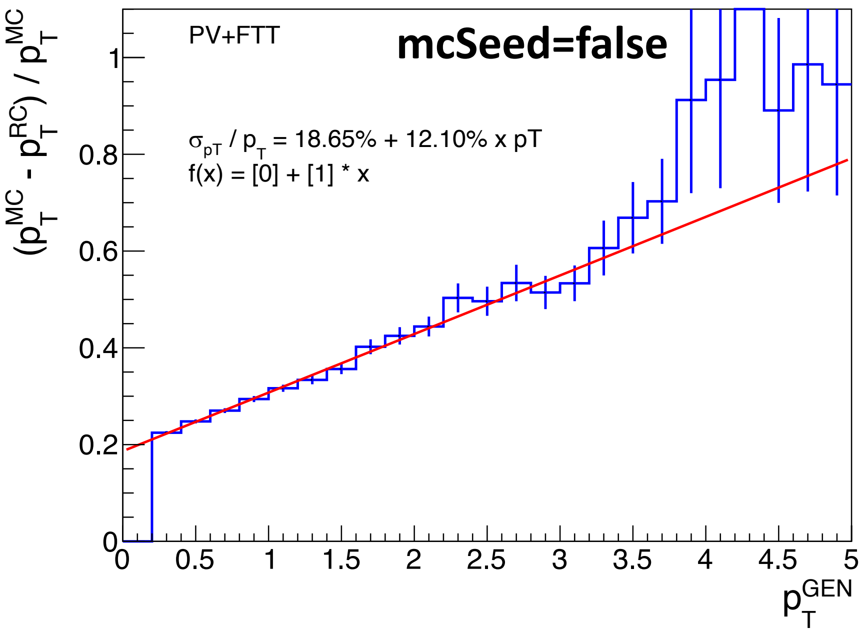
- `mcSeed="false"` - this means the seed is determined from the space points in sTGC – based on 4 simple circle fits (to each set of 3 hits)
- This is what we can do in data

Tracking Mysteries







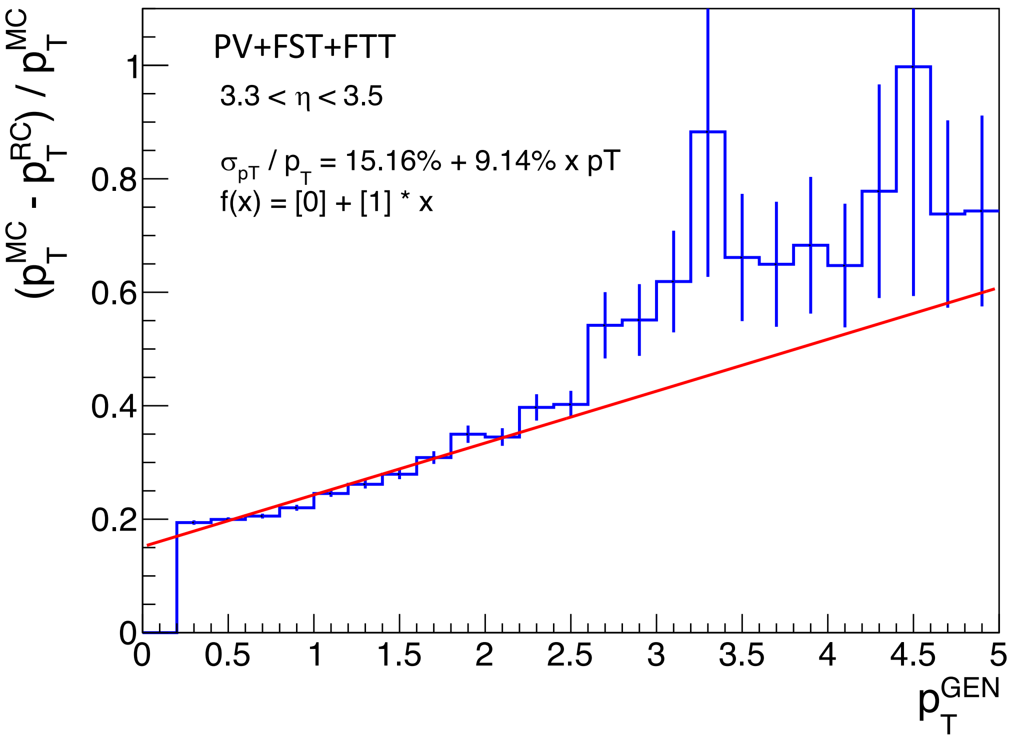


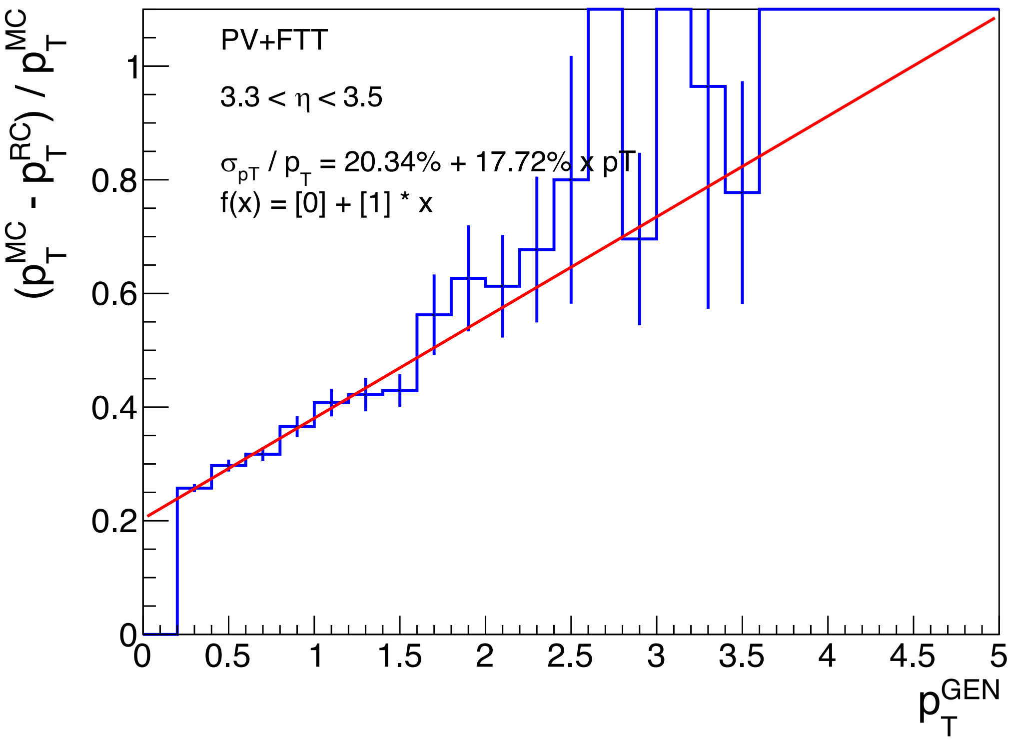
With “real” fit seed the FTT only case shows that the resolution has strong pT dependence

- as expected

Update comparison w/FST

- At high Eta, FST in tracking improves the momentum resolution significantly
  - These are single muon "events" PV=500μm



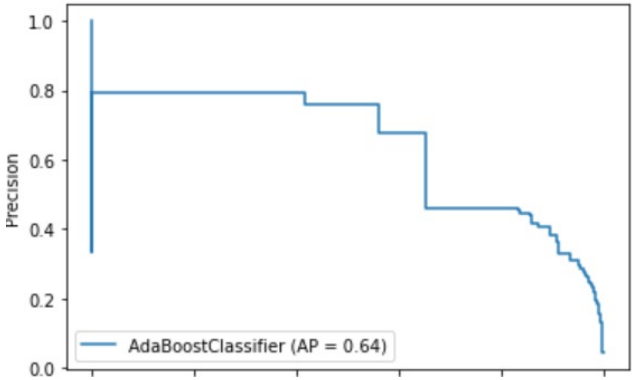


BDT-Based Track Seed Finding

- Youqi (Yale) has been working on BDT-based Cellular automata track finding
- Trained in python scikit-learn : previously stuck getting into ROOT
- Last f2f meeting, Hongwei pointed us to example BDT->ROOT

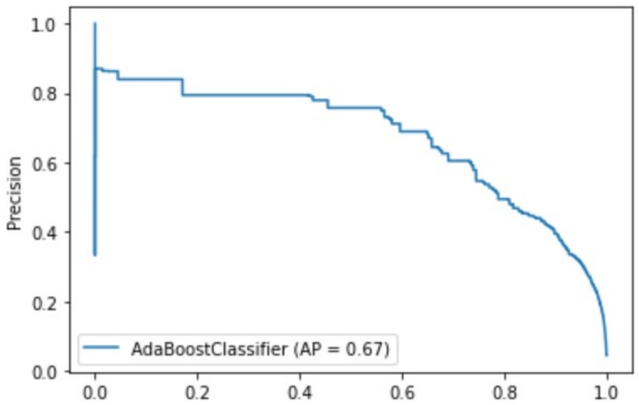
```
              precision    recall  f1-score   support

        fake       0.98      0.99      0.99   2718792
        real       0.76      0.56      0.64    128367

    accuracy                           0.97   2847159
   macro avg       0.87      0.78      0.81   2847159
eighted avg       0.97      0.97      0.97   2847159

rea under ROC curve: 0.9745
```

```
Precision-recall curve, https://scikit-learn.org/stable/(
isp = plot_precision_recall_curve(bdt2, X2_test, y2_test)
```




```
        fake       0.98      0.99      0.99   2718792
        real       0.76      0.56      0.64    128367

    accuracy                           0.97   2847159
   macro avg       0.87      0.78      0.81   2847159
weighted avg       0.97      0.97      0.97   2847159

Area under ROC curve: 0.9756
```

```
In [12]: # Precision-recall curve, https://scikit-learn.org/stab
         disp = plot_precision_recall_curve(bdt2, X2_test, y2_te
```



- BDT-based track finding shows promising performance
- Need to test in full tracking code
- BDT is trained on existing 2-hit and 3-hit “Criteria”

BDT-Based Track Seed Finding

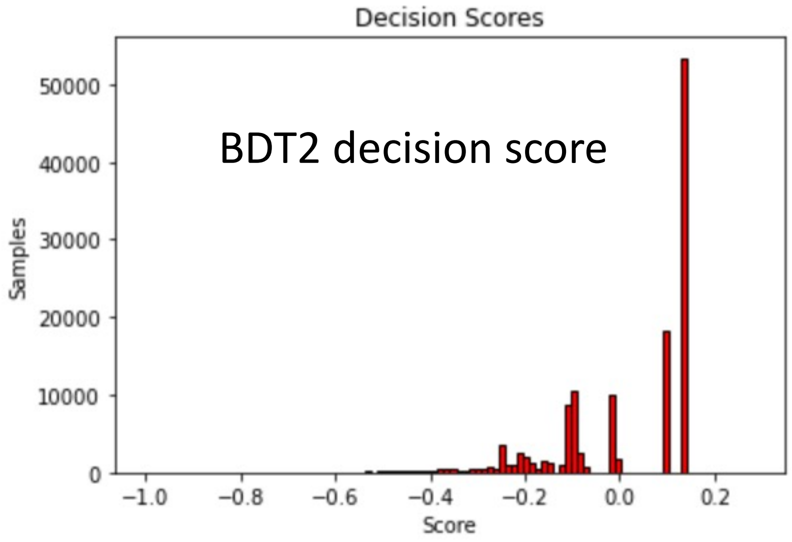
- Youqi implemented ROOT TMVA code to read BDT trained in python

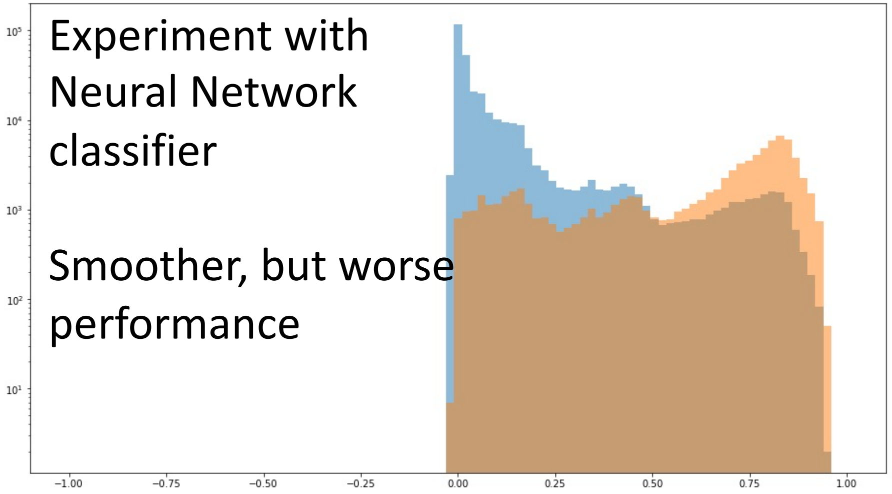
```
<TrackFinder nIterations="1">
        <Iteration nPhiSlices="1"> <!-- Options for first iteration -->
            <SegmentBuilder>
                <Criteria name="Crit2_BDT" min="0.0" max="10.0" weights="bdt2.xml" />
            </SegmentBuilder>

            <ThreeHitSegments>
                <Criteria name="Crit3_BDT" min="0.0" max="10.0" weights="bdt3.xml"/>
            </ThreeHitSegments>
        </Iteration>
        ...
</TrackFinder>
```

- I added new classes to tracking code, to handle the TMVA and interface with the cellular automata.

BDT Stability and Follow-up Checks

- BDT “score” for pure signal (real tracks) shows spikey structure





- Expect smooth/continuous “score” – need to understand this more
- Next tasks:
  - Identify which “Criteria” cause the spikey BDT score
  - Select optimal score threshold based on full tracking performance

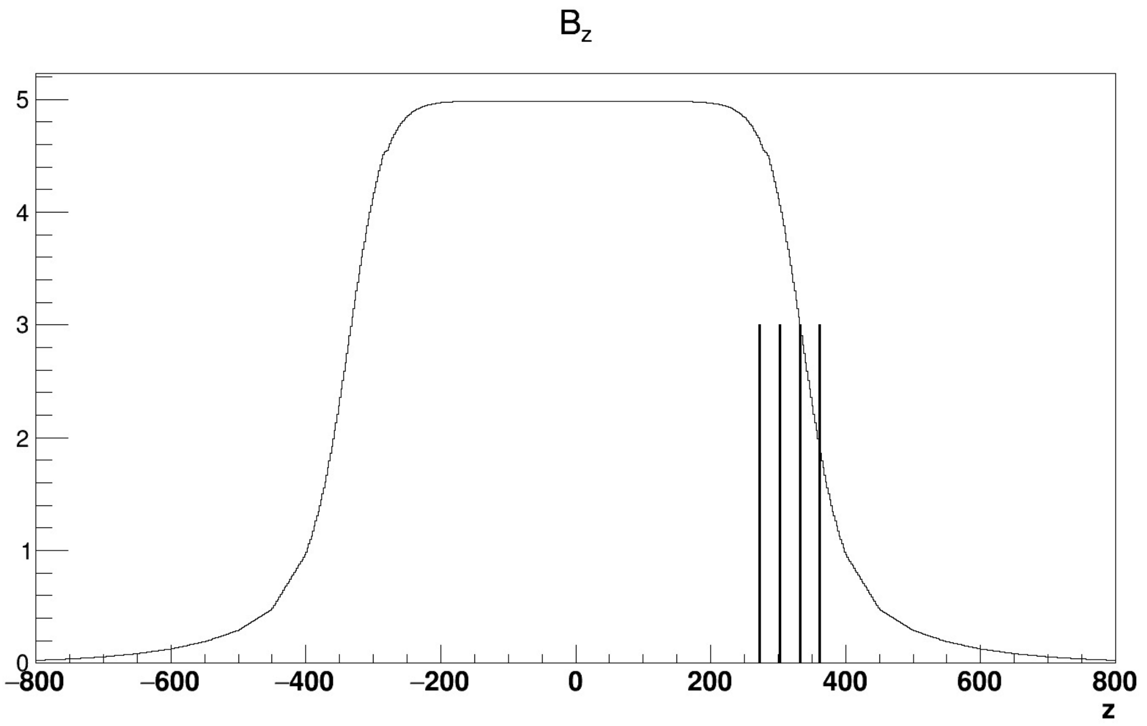
Magnetic Field Tasks

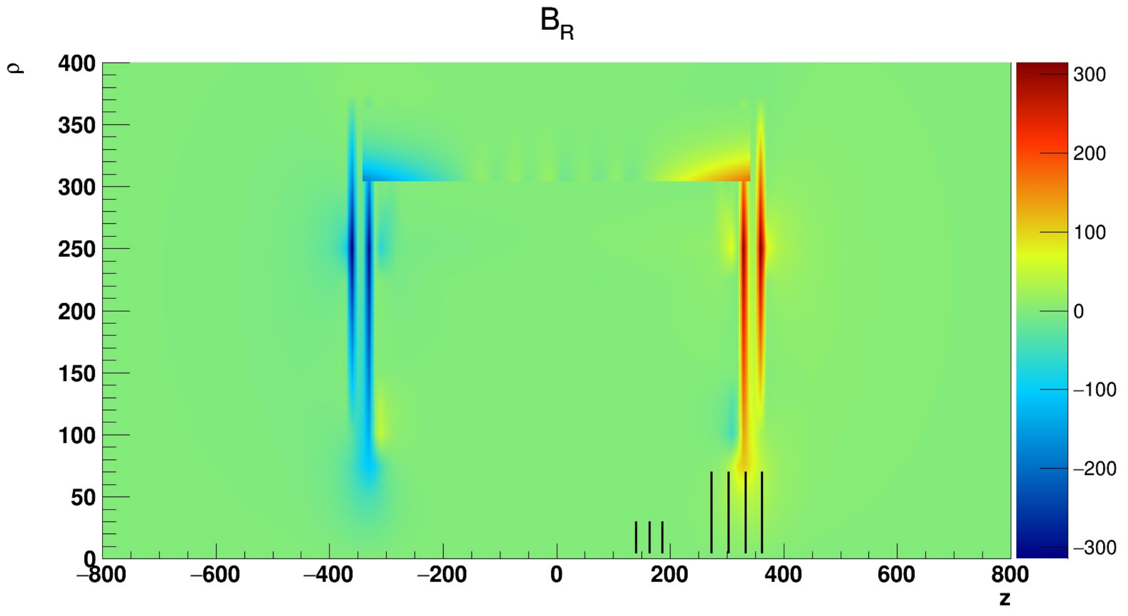
- Yevheniia will work on this (as a HEPDb shift) – with Grigory helping/reviewing
- Example code here: https://gist.github.com/jdbrice/a27c8c6ae7c0eff0165acd95dd063efe
- Lookup field using `StMagF::Agufld` interface
  - This code seems slow, it takes ~1m to export full field, >20-100M files (depending on resolution)

- GENFIT tracking looks-up the mag field many times per track fit – potential source of slowness

Magnetic Field Tasks

- Magnetic field is mostly well behaved around tracking detectors
- IDEA: implement a parameterization / in-memory lookup table for B-field in needed region





- GOAL:

1. improve tracking speed
2. Test sensitivity to “slightly wrong” B-Field

Summary and Tasks

- Recently Completed
  - Update BFC to allow forward tracking + tracking sim
  - sTGC offline chain (most parts at least in skeleton)
  - Update sTGC geometry using pgon (ROOT only)
  - Tracking Updates
    - More studies of track finding in Pythia (Youqi) + added BDT classes for track finding CA
    - Resolve
- Work in progress
  - FST geometry – (Simplified for GENFIT, full details with misalignment)
  - FTT geometry – new geometry with pgon implemented (Jason will convert to AgML)
    - Nicole (BNL post-doc) helping with simplified model for GENFIT, will test if needed