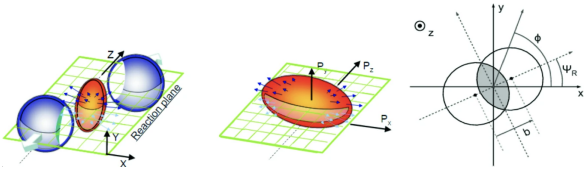
Event plane reconstruction in O2Physics

Junlee Kim[1]

October 17, 2024
O2 Tutorial 4.0

1. CERN

Event plane reconstruction



- Flow coefficients can be extracted using the reaction plane of ψ_n
  - $v_n = \langle \cos\left[n(\varphi - \psi_n)\right] \rangle$
  - Not possible to access ψ_n in experiment.
- Alternatively, the event plane (Ψ_n) is reconstructed for n-th modulations using the Q-vector
  - $Q_{n,x} = \sum_i \omega_i \cos(n\varphi_i)$ and $Q_{n,y} = \sum_i \omega_i \sin(n\varphi_i)$
    - ω_i: weight factor of each component, detector amplitude or p_{T} ($v_n \propto p_{\mathrm{T}}$)
    - Gain equalization for FIT detector with a factor of $(M_{\mathrm{tot}}^{\mathrm{avg}}/M_i^{\mathrm{avg}})$
  - $\Psi_n = (1/n)\arctan\left(Q_{n,y}/Q_{n,x}\right)$
  - $v_n^{\mathrm{obs}} = \langle \cos\left[n(\varphi - \Psi_n)\right] \rangle$
- The resolution of the event plane can be calculated with 3-sub event method
  - $\mathcal{R}_n = \langle \cos(n(\Psi_n^A - \psi_n)) \rangle \approx \sqrt{\dfrac{\langle \cos(n(\Psi_n^A - \Psi_n^B)) \rangle \langle \cos(n(\Psi_n^A - \Psi_n^C)) \rangle}{\langle \cos(n(\Psi_n^B - \Psi_n^C)) \rangle}}$
  - Individual booking for $\langle \cos(n(\Psi_n^A - \Psi_n^B)) \rangle$, $\langle \cos(n(\Psi_n^A - \Psi_n^C)) \rangle$, and $\langle \cos(n(\Psi_n^B - \Psi_n^C)) \rangle$.
  - $v_n^{\mathrm{cor}} = v_n^{\mathrm{obs}}/\mathcal{R}_n$

- Data Model: `Common/DataModel/Qvectors.h`
  - `DECLARE_SOA_COLUMN(QvecDetRe(Im)Vec, qvecDetRe(Im)Vec, std::vector<float>);`
  - `DECLARE_SOA_TABLE(QvectorFT0CVecs, "AOD", "QVECTORSFT0CVEC", qvec::IsCalibrated, qvec::QvecFT0CReVec, qvec::QvecFT0CImVec, qvec::SumAmplFT0C);`
  - Support to save multiple harmonic orders for different subsystems
- Table Producer: `Common/TableProducer/qVectorsTable.cxx`
  - Main piece to calculate and correct Q-Vectors.
  - Gain equalization and Q-Vector calibration
  - A few key configurable parameters to be emphasized.
- Helper: `Common/Core/EventPlaneHelper.h`
  - Provide relevant functions
- Validation Task: `Common/Tasks/qVectorsCorrection.cxx`
- Example: `PWGLF/Tasks/Strangeness/lambdapolarization.cxx`

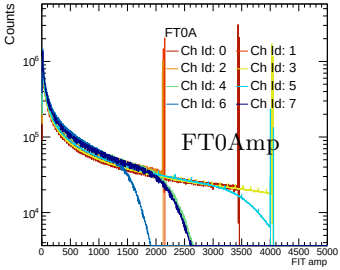
Preparation

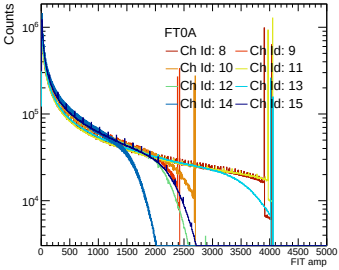
- Gain equalization is required to ensure that every channel provides the same signal amplitude for a traversing charged particle.
- Require to have amplitude distribution for each channel
  `histosQA.fill(HIST("FV0Amp"), ampl, FV0AchId);` (L.350, L.395)
- Average amplitudes of each channel and average amplitude of all channels to calculate ($M_{\mathrm{tot}}^{\mathrm{avg}}/M_{i}^{\mathrm{avg}}$)
- Upload via `ccdb.storeAsTFileAny(VECTOR, ccdbInternalPath.c_str() metadata, sor, eor);`
- `cfgGainEqPath` in Table Producer: CCDB path for accessing gain equalization constants

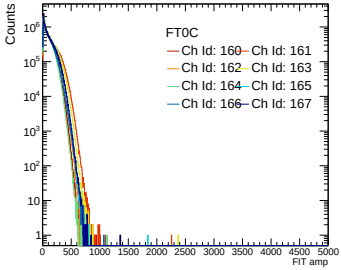
Equalization

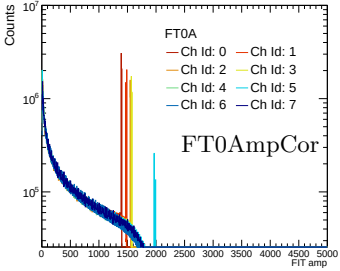
- Load gain via `auto objft0Gain = ccdb->getForTimeStamp<std::vector<float>>(fullPath, timestamp);`
- Validate the process with `histosQA.fill(HIST("FT0AmpCor"), ampl / FT0RelGainConst[FT0CchId], FT0CchId);`

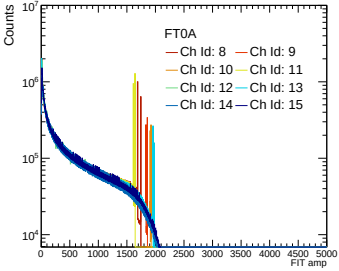
Equalized amplitude distribution

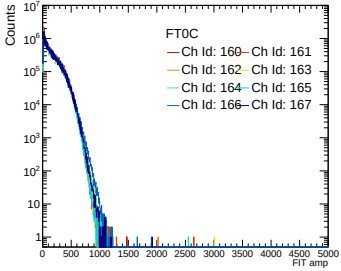




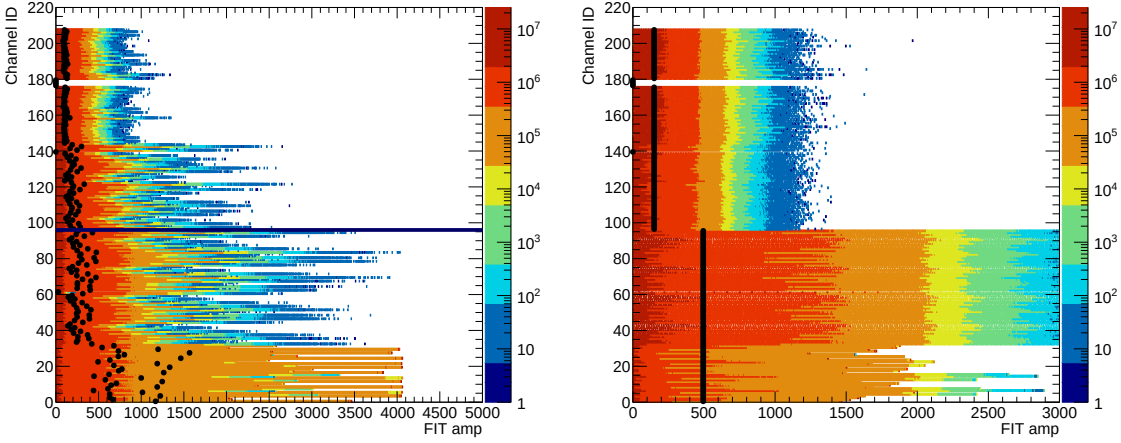








Raw amplitude distribution



- Gain equalization has been done for FT0C and FT0A (the same treatment for V0A and V0C in Run2)

Preparation

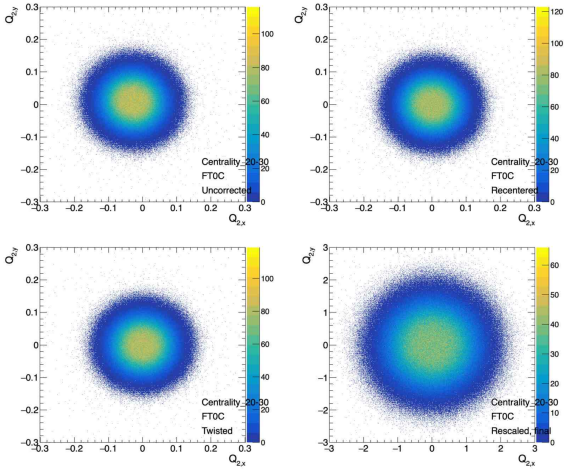
- Uncorrected 2-D scatter plots of $Q_{n,x}$ and $Q_{n,y}$.
- Generate correction constants from the scatter plot based on functions in `Common/Core/EventPlaneHelper.h`
  - TH3 with 6 (calibration step) $\times$ 7 (subsystems) $\times$ 90 (centrality intervals of 1%) bin size
- Upload constants in the CCDB (L.246)

Correction

- Load gain via `auto objqvec = ccdb->getForTimeStamp<TH3F>(fullPath, timestamp);`
- Validate the process with `Common/Tasks/qVectorsCorrection.cxx`

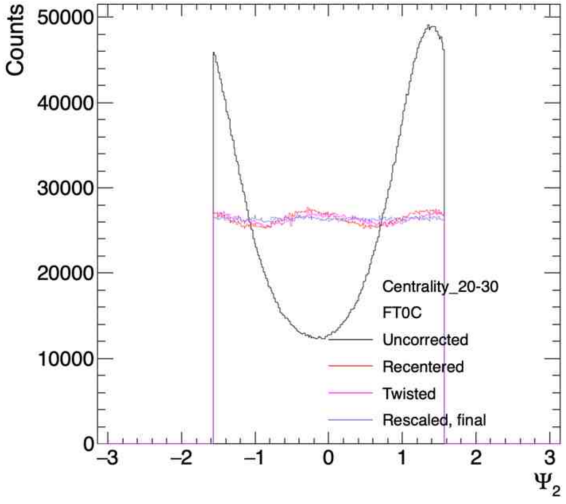
Q-Vector correction

- Scatter plot of $Q_{n,x}$ and $Q_{n,y}$ would be isotropic around the origin.
  - Uniform event plane distribution
- In reality, $(Q_{n,x}, Q_{n,y})$ distribution is not centered and elongated.
  - **Recentering** corrects the center position
  - **Twist** corrects the possible rotation
  - **Rescaling** corrects the possible deformation
- Prerequisite: gain equalization



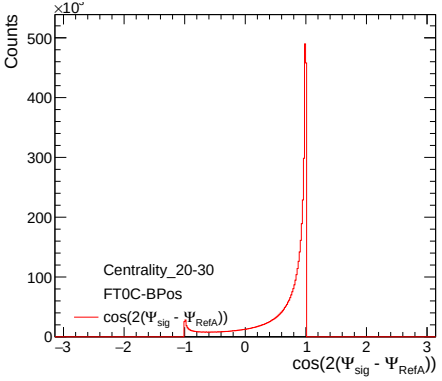
Event plane distribution

- $\Psi_2 = (1/2)\arctan\left(Q_{2,y}/Q_{2,x}\right)$
- Fully corrected distribution is uniform as expected.
- Validation with `Common/Tasks/qVectorsCorrection.cxx`

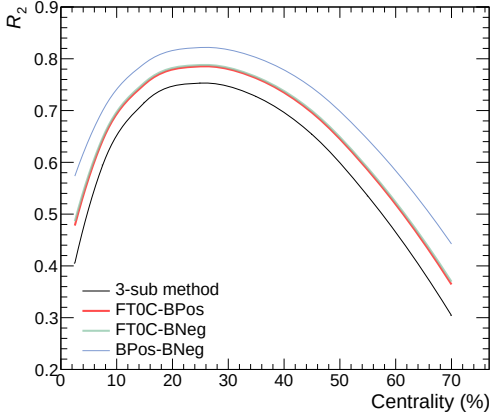


The resolution of the event plane

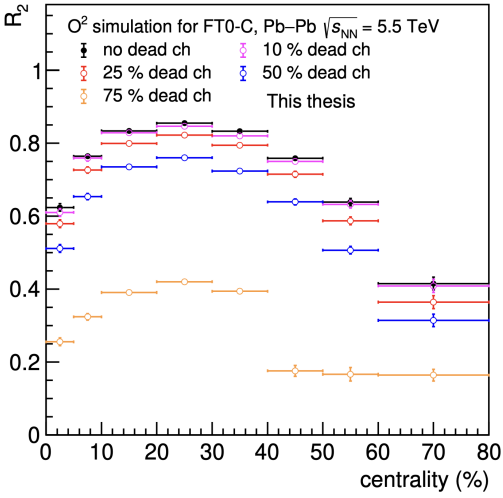
- $\mathcal{R}_n = \langle \cos(n(\Psi_n^A - \psi_n)) \rangle \approx \sqrt{\frac{\langle \cos(n(\Psi_n^A - \Psi_n^B)) \rangle \langle \cos(n(\Psi_n^A - \Psi_n^C)) \rangle}{\langle \cos(n(\Psi_n^B - \Psi_n^C)) \rangle}}$
- $v_n^{\mathrm{cor}} = v_n^{\mathrm{obs}} / \mathcal{R}_n$
- Individual booking for $\langle \cos(n(\Psi_n^A - \Psi_n^B)) \rangle$, $\langle \cos(n(\Psi_n^A - \Psi_n^C)) \rangle$, and $\langle \cos(n(\Psi_n^B - \Psi_n^C)) \rangle$
  - The figure shows $\langle \cos(n(\Psi_n^A - \Psi_n^B)) \rangle$
- The average value of distributions is used to evaluate the resolution
- histogram generation with `Common/Tasks/qVectorsCorrection.cxx`



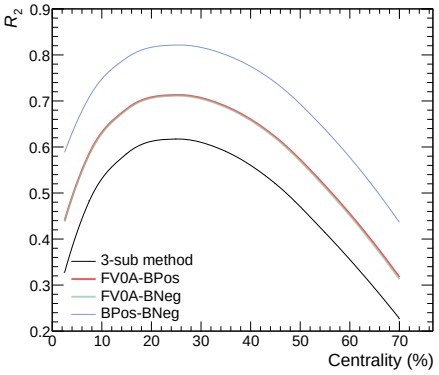
Event plane resolution for FT0C

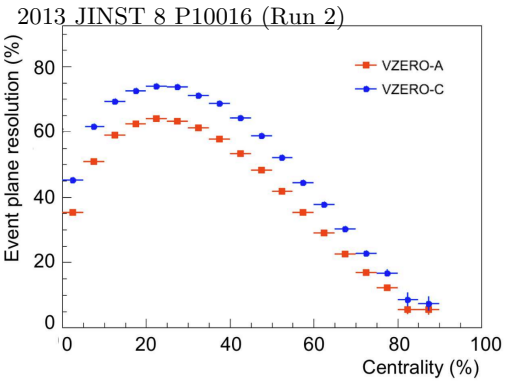


CERN-THESIS-2023-094

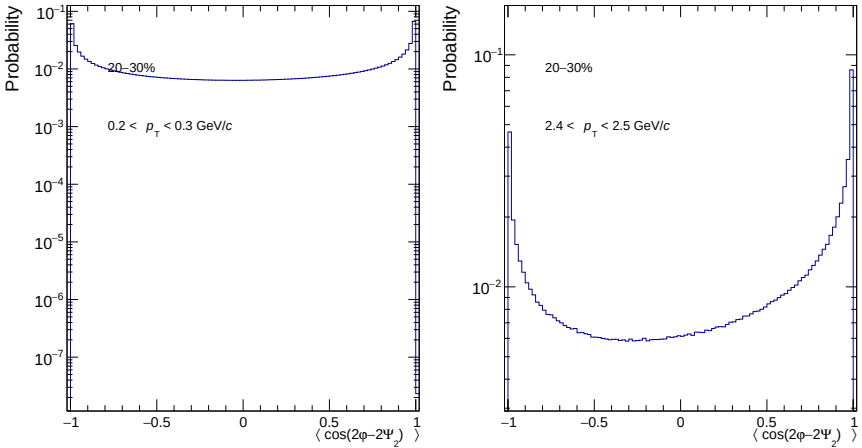


Event plane resolution for FV0A

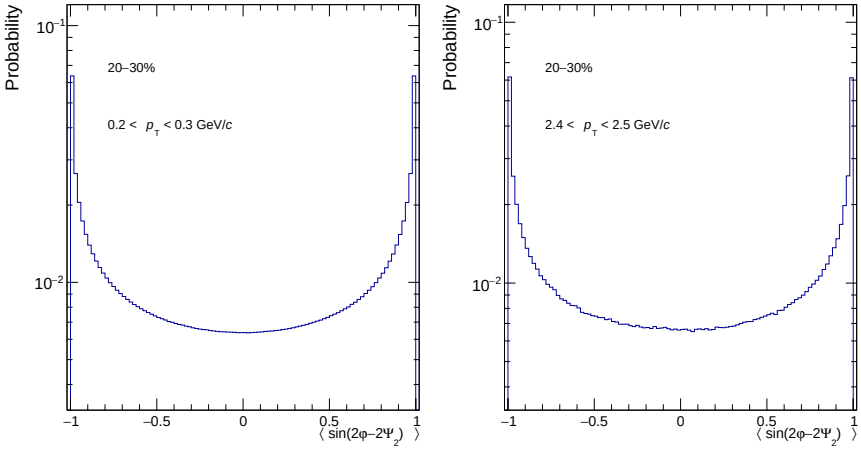




- Comparable event plane resolution between Run 2 and this work

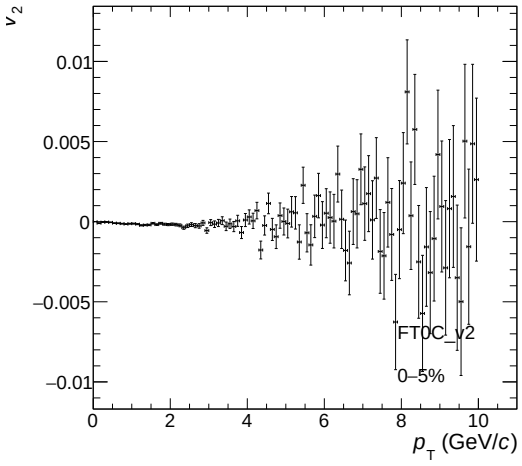


- Average of the histogram to be signal

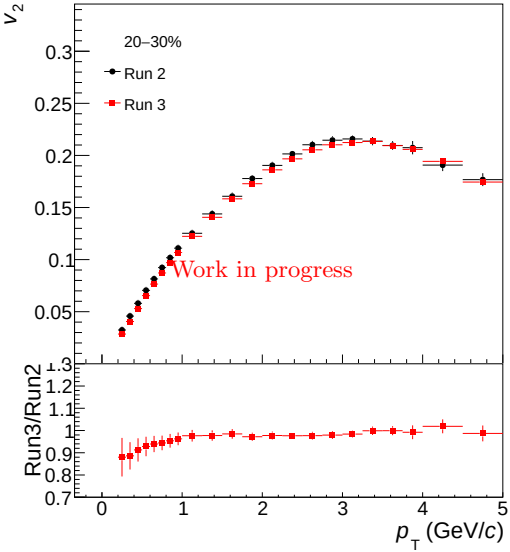


- Average of the histogram to examine possible bias
- $\langle \sin[n(\varphi - \Psi_n)] \rangle = 0 \rightarrow$ well decomposed to cosine

- $\langle \sin [n(\varphi - \Psi_n)] \rangle = 0$ throughout all measured p_{T} $\rightarrow$ no bias is expected



- Discrepancy especially in low p_{T}
- Bias from high occupancy?
- Better agreement with updated centrality calibration in pass4?
- There are still open questions on the measurement.



Exercises

Before doing this make sure that you are in the O2Physics with `alienv enter O2Physics::latest`
Main code to work in this time: `Tutorials/PWGCF/EventPlane/src/qVectorstutorial.cxx`
Constants for gain equalization and Q-Vector corrections are prepared at `Users/j/junlee/Qvector/Pass4`
You need to update tutorial code with `ninja Tutorials/PWGCF/EventPlane/install`
Get the A02D.root, macro, and configuration for running the task

- Checking gain equalization
  - See `http://alice-ccdb.cern.ch/browse/Users/j/junlee/Qvector/Pass4` and insert path to conf.
  - Plot and compare FT0Amp and FT0AmpCor
- Checking Q-Vector corrections
  - Q-Vectors with each calibration step can be obtained with `cfgCorrLevel`
  - Plot 2-D Q-Vectors for each step from 1 to 4
    (`cfgCorrLevel`=0 corresponds to the Q-Vector before gain eq.)
- Checking event plane distribution
  - Plot event plane distribution and check if it is uniform
- Measuring v_2 and correct it for event plane resolution
  - Event plane resolution with 3-sub event method: need to book more histograms
  - Book another histogram for sine term to see if any acceptance bias
  - Correct v_2 with the event plane resolution
- Draw higher-order such like v_3 and v_4

BACKUP