Universiteit Utrecht

Faculteit Bètawetenschappen

Principal Component Analysis of elliptic flow fluctuations at $\sqrt{s_{NN}} = 2.76$ TeV in PbPb collisions at ALICE

Bachelor thesis

D.J.W. Verweij

Study: Natuur- en Sterrenkunde

Supervisors:

prof. dr. R.J.M.Snellings
Institute for Subatomic Physics Utrecht

J. Margutti MSc
Institute for Subatomic Physics Utrecht

June, 2016

It is a very unlikely series of events, but it is possible.

– Robin Banks, *narrator Mythbusters*

Abstract

Event-by-event flow fluctuations have long been a recognized influence on anisotropic flow measurements and the cause of factorization breaking of two-particle correlations. A Principle Component Analysis (PCA) provides a measure for anisotropic flow and quantifies these underlying fluctuations. PCA produces leading modes, which are comparable with flow harmonics (v_n), and sub-leading modes, which relate to different causes of event-by-event flow fluctuations. PCA of elliptic flow and its fluctuations is presented at $\sqrt{s_{NN}} = 2.76$ TeV in PbPb collisions for two-particle azimuthal correlations as a function of pseudorapidity (η) and transverse momentum (p_T) with data from the ALICE detector at the LHC. The results are presented for multiple centrality windows. The leading modes and the first sub-leading modes are used to present factorization breaking effects for both p_T- and η-dependence. Also a comparison is made between PCA and the accepted Q-cumulants method for two-particle correlations. It can be concluded that PCA can be used to determine elliptic flow and, moreover, gives new information about event-by-event effects on flow. These effects can be attributed to hydrodynamic processes, namely the event-by-event initial density fluctuations and torque effects due to forward-backward moving particles.

Contents

Chapter 1

Introduction

Anisotropic flow, v_n is a key observable in the field of heavy-ion collisions[1][2]. Even though azimuthal anisotropies have been measured since the very first high energy nuclear collisions, they became a popular observable when large, significant in-plane elliptic flow, v_2, was first discovered at RHIC[3]. Due to the fact that anisotropies are created in the early stages of the system, it gives unique information about the first few fm/c after the collision, where the Quark-Gluon Plasma (QGP) is thought to exist.

Anisotropic flow is defined as the Fourier coefficients of the azimuthal dependence of the invariant particle yield with respect to the reaction plane[4]:

$$\frac{2\pi}{N}\frac{dN}{d\phi} = 1 + 2\sum_{n=1}^{\infty} v_n(p_T,\eta)\cos(n(\phi - \Psi_n(p_T,\eta))) \tag{1.1}$$

Current techniques of measuring anisotropic flow coefficients, such as analysis with Q-cumulants[5][6] and the event-plane method[7], are affected by multiplicity and flow fluctuations in a non trivial way, making it difficult to decouple their effect. The latter method also forces one to approximate the reaction plane, which cannot be measured directly. This results in approximations for the reaction plane and incomplete use of all the information available. Furthermore, as a result of event-by-event fluctuations of the shape of the initial energy density, anisotropic flow fluctuates. This event-by-event flow fluctuation is still a matter of debate, even though it has long been recognized to have an important role in nuclear collisions[8]. These fluctuations can be the cause of small factorization effects, which were found in several experiments [9][10][11].

A new way to use all information has been introduced in [12] and further examined at CMS in [13], where the flow model is used that, varying per event, particles are emitted with a certain probability distribution dependent upon transverse momentum, p_T, pseudorapidity[1], η and azimuthal angle, ϕ. The single particle distribution with $d\mathbf{p} \equiv dp_T\, d\eta\, d\phi$ can then be written as:

$$\frac{dN}{d\mathbf{p}} = \sum_{n=-\infty}^{+\infty} V_n(p_T,\eta)e^{in\phi} \tag{1.2}$$

and the same applies for the pair distribution:

$$\left\langle \frac{dN_{pairs}}{d\mathbf{p}_1 d\mathbf{p}_2} \right\rangle = \sum_{n=-\infty}^{+\infty} V_{n\Delta}(p_{T,1},p_{T,2},\eta_1,\eta_2)e^{in(\phi_1-\phi_2)} \tag{1.3}$$

The brackets denote an average over every event. The advantage of two- (or multiple-) particle correlations is the suppression of non-flow[2] correlations, due to the collective

[1] Pseudorapidity $\eta = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$ with θ the angle between the beam axis and the particles trajectory

[2] Non-flow correlations are correlations, which cannot be attributed to collective behavior. Examples are resonance decays, jet fragmentation or Bose-Einstein correlations[14].

nature of anisotropic flow. Here correlations caused by non-flow are left out, since these become negligible small for large systems. The resulting Fourier coefficients for $V_{n\Delta}$ become a covariance matrix.

$$V_{n\Delta}(p_{T,1}, p_{T,2}, \eta_1, \eta_2) = \langle V_n(p_{T,1}, \eta_1) V_n^*(p_{T,2}, \eta_2) \rangle \tag{1.4}$$

This thesis uses a Principal Component Analysis[12] (PCA) to fully exploit all the hidden information in the correlation matrix $V_{n\Delta}$. This gives both p_T- and η-dependences for anisotropic flow and, for this thesis especially, elliptic flow (v_2). Furthermore, the eigenmodes and eigenvalues of $V_{n\Delta}$ reveal multiple modes, where the leading mode is the anisotropic flow and on top of that the sub-leading modes contain new information on the initial state and conditions of the collision. By analyzing ALICE data these eigenmodes can be determined and compared to the flow analysis with Q-cumulants for both p_T- and η-dependence on the same data.

Chapter 2

Experimental Setup

ALICE [15] (A Large Ion Collider Experiment) is a dedicated heavy-ion experiment at CERN's Large Hadron Collider in Geneva. It was designed to handle the large particle densities reached in the most head-on PbPb collisions. Furthermore, it can measure particle transverse momentum down to $p_T^{min} \approx 0.15$ GeV/c. Among the many subdetectors of ALICE, the ones used in the current analysis are the V0 detectors, the Inner Tracking System (ITS) and the Time Projection Chamber (TPC). The V0 detectors are used for centrality[1] determination using the mean multiplicity. ITS tracks particle paths and decay vertexes. Lastly, the TPC is the main tracking detector at midrapidity with excellent particle identification (PID) capabilities.

For this thesis, a sample of 10M PbPb collisions at $\sqrt{s_{NN}} = 2.76$ TeV is used. The data was collected by ALICE in 2010. The events included in the analysis are up to centrality 60%.

Tracks are selected requiring a minimum number of 70 TPC clusters for $p_T > 1.5$ GeV/c and a minimum of 100 TPC clusters for $p_T > 20$ GeV/c after first tracking iteration[16]. Further selection and computation is executed with AliRoot[2]. The corresponding code can be found in Appendix C. Only the tracks in the region $|\eta| < 0.8$ and $0.2 < p_T < 5.0$ GeV/c are selected.

A correction for the non-uniform azimuthal acceptance is made (see equations 3.8, 3.11 and 3.15), though it contributes for less than 0.5% of the final value of v_2. This small contribution is expected, since the detectors have full nominal azimuthal acceptance. Statistical uncertainties for the PCA are calculated by analyzing subparts of the data and subsequently determining the root mean squared error of v_2. Tracking efficiencies are not corrected for in the final results, but have been estimated to contribute by a few percent maximum.

[1] Centrality is used for the categorization of nuclear collisions, where centrality is parametrized by the impact parameter **b**, the distance between the two colliding nuclei. As **b** is not a direct observable, the centrality can be experimentally inferred from the multiplicity of the produced hadrons[14].

[2] For more information see: AliRoot installation

Chapter 3

Analysis Technique

For this thesis, two methods are used to analyze the data from ALICE mentioned above. Firstly, the Q-cumulants method from [5] and [6] is used to measure the elliptic flow coefficient, v_2. Secondly, the principal component analysis (PCA) is used, as discussed in [12]. The two analyses are expected to provide two different measurements of the elliptic flow coefficient, in particular of its dependence on p_T and η at different centralities. In [13] it was found that the two analyses should give two values, which are connected to each other:

$$v_n^{PCA}(p_T, \eta) \sim \sqrt{N_{pairs}} v_n^Q(p_T, \eta) \tag{3.1}$$

However, it's still not clear whether the values are expected to coincide. Moreover, the PCA provides multiple sub-leading modes, which provide more information on the underlying event-by-event flow fluctuations. In the equations in this chapter, n is used to characterize the different coefficients of anisotropic flow. However, this thesis only looks at elliptic flow ($n = 2$). In consideration of the readability and reusability of the equations, n is not inserted.

3.1 Q-cumulants

In this thesis, the analysis for two-particle azimuthal correlations from [5] is used. Among the available methods, this one shows the most resemblance to the calculations used in the PCA. Higher particle correlations are neglected, since the PCA is also based on two-particle correlations. The errors are calculated with the methods from [6].

3.1.1 Reference Flow

Firstly, we calculate a reference flow averaged for each event over the complete range of p_T or η.

$$\langle 2 \rangle = \frac{\sum_{i,j=1}^{M} e^{in(\phi_i - \phi_j)} - M}{M(M-1)} \tag{3.2}$$

Here, M is the number of tracks and $\langle 2 \rangle$ is the reference flow for a *single*-event. The sum is more commonly noted as $|Q_n|^2$, which explains the name Q-cumulants. $|Q_n|^2$ is usually referred to as the flow vector[2] and can also be rewritten to:

$$\sum_{i,j=1}^{M} e^{in(\phi_i - \phi_j)} = e^{in\phi_i} e^{-in\phi_j} \tag{3.3}$$

$$= \Big(\cos n\phi_i + i \sin n\phi_i \Big) \cdot \Big(\cos n\phi_j - i \sin n\phi_j \Big) \tag{3.4}$$

$$\begin{aligned} &= \cos n\phi_i \cos n\phi_j + \sin n\phi_i \sin n\phi_j \\ &\quad + \; i \left(\sin n\phi_i \cos n\phi_j - \cos n\phi_i \sin n\phi_j \right) \end{aligned} \tag{3.5}$$

Note that the imaginary part can be ignored. Next the average over all events is calculated.

$$\langle\langle 2\rangle\rangle \equiv \frac{\sum\limits_{Events} \left(w_{\langle 2\rangle}\right)_i \langle 2\rangle_i}{\sum\limits_{Events} \left(w_{\langle 2\rangle}\right)_i} \tag{3.6}$$

$$= \frac{\sum\limits_{Events} M_i(M_i-1)\langle 2\rangle_i}{\sum\limits_{Events} M_i(M_i-1)} \tag{3.7}$$

Weights are added to ensure minimal fluctuations due to different event multiplicities. After adding an azimuthal correction to correct for the non-uniform acceptance of the detector, the reference flow is obtained. The double brackets indicate an average over all tracks and over all events.

$$Reference\ Flow = \langle\langle 2\rangle\rangle - \left[\langle\langle \cos n\phi_1\rangle\rangle^2 + \langle\langle \sin n\phi_1\rangle\rangle^2\right] \tag{3.8}$$

3.1.2 Differential Flow

The calculations for differential flow are analogous to those for the reference flow. The only difference is that differential flow is calculated per bin, instead of over all bins. Therefore the variable m is needed, which is the number of tracks in a certain bin. The differential flow averaged for each event becomes:

$$\langle 2'\rangle = \frac{\sum\limits_{i=1}^{m} {\sum\nolimits'}_{j=1}^{M} e^{in(\psi_i-\phi_j)}}{m(M-1)} \tag{3.9}$$

The second sum is taken over all $j \neq i$. Here ψ_i and ϕ_j are the azimuthal angles of particles in the desired bin and all bins, respectively. Averaging over all events gives the uncorrected differential flow $\langle\langle 2'\rangle\rangle$.

$$\langle\langle 2'\rangle\rangle \equiv \frac{\sum\limits_{Events} m_i(M_i-1)\langle 2'\rangle_i}{\sum\limits_{Events} m_i(M_i-1)} \tag{3.10}$$

Again non-uniform azimuthal detector acceptance is corrected for by subtracting correction terms. This results in the following final differential flow.

$$Differential\ Flow = \langle\langle 2'\rangle\rangle - [\langle\langle \cos n\psi_1\rangle\rangle\langle\langle \cos n\phi_2\rangle\rangle + \langle\langle \sin n\psi_1\rangle\rangle\langle\langle \sin n\phi_2\rangle\rangle] \tag{3.11}$$

Where ψ_1 is averaged over all tracks and all events in the corresponding bin and ϕ_2 is averaged over all bins. The anisotropic flow, v_n, can now easily be calculated from the differential and reference flow.

$$v_n = \frac{DifferentialFlow}{\sqrt{ReferenceFlow}} \tag{3.12}$$

3.2 Principle Component Analysis

The mathematical concept of PCA is to construct independent components, which, together, explain the correlations between two particles, $V_{n\Delta}$ and thereby the flow fluctuations per event. In a more mathematical way this is denoted as:

$$V_{n\Delta}(p_{T,1}, p_{T,2}, \eta_1, \eta_2) \approx \sum_{\alpha=1}^{k} V_n^{(\alpha)}(p_{T,1}, \eta_1) V_n^{(\alpha)*}(p_{T,2}, \eta_2) \tag{3.13}$$

For $k = 1$, equation 3.13 gives only one mode for the correlation, which is the usual anisotropic flow. However, for $k > 1$ factorization breaking takes place and new modes are added, which also contribute to $V_{n\Delta}$. These new components may reveal information about flow fluctuations. Therefore the correlations between particles need to be calculated to determine the components.

3.2.1 Construction of the correlation matrix

The detector acceptance is divided into N_b bins for either p_T or η. The assigned number of bins is not only relevant for the logical distribution of entries per bin, but also determines the size of the covariance matrix calculated below and therefore the number of modes which are calculated, since $k \leq N_b$. A new flow vector Q_n is determined corresponding to the flow model sketched in chapter 1.

$$Q_n(p_T, \eta) = \frac{1}{2\pi \Delta p_T \Delta\eta} \sum_{j=1}^{m} e^{in\phi_j} \tag{3.14}$$

The first component is a normalization with Δp_T and $\Delta\eta$ the range in transverse momentum and pseudorapidity space for a specific bin[1]. m is again the number of particles in a particular bin. Subsequently, the correlations between different bins $V_{n\Delta}$ is calculated. For all bins, this gives a covariance matrix $V_{n\Delta} : \{N_b \times N_b\}$.

$$\begin{aligned} V_{n\Delta}(p_T^a, p_T^b, \eta^a, \eta^b) = &\langle Q_n(p_{T,1}, \eta_1) Q_n^*(p_{T,2}, \eta_2)\rangle - \frac{\langle M(p_{T,1}, \eta_1)\rangle\, \delta_{p_{T,1}, p_{T,2}} \delta_{\eta_1, \eta_2}}{(2\pi \Delta p_T \Delta\eta)^2} \\ &- \langle Q_n(p_{T,1}, \eta_1)\rangle \langle Q_n^*(p_{T,2}, \eta_2)\rangle \end{aligned} \tag{3.15}$$

The second term subtracts self correlations on the diagonal of the matrix. The last term corrects for a non-uniform detector acceptance by deducting the average Q-vectors[17]. The brackets indicate the average over all events. The resulting matrix is positive semidefinite by construction, when only the first term is taken into account, since it results from an inner product. This means all eigenvalues are positive. After considering the other two terms, some eigenvalues become negative due to non-flow correlations.

3.2.2 Calculation of eigenmodes

Next step is diagonalizing $V_{n\Delta}$, which gives the eigenvalues and eigenmodes.

$$V_{n\Delta}(p_T^a, p_T^b, \eta^a, \eta^b) = \sum_{\alpha} \lambda^{(\alpha)} \psi^{(\alpha)}(p_{T,1}, \eta_1) \psi^{(\alpha)}(p_{T,2}, \eta_2) \tag{3.16}$$

Here $0 < \alpha \leq N_b$,eigenvalues are ordered from largest to smallest ($\lambda^{(1)} > \lambda^{(2)} > \dots$) and $\psi^{(\alpha)}(p_T, \eta)$ is the corresponding eigenvector. Comparing equations 3.13 and 3.16, quickly learns that there is a way to express $V_n^{(\alpha)}(p_T, \eta)$ as a function of the eigenvalues and eigenmodes.

$$V_n^{(\alpha)}(p_T, \eta) \equiv \sqrt{\lambda^{(\alpha)}} \psi^{(\alpha)}(p_T, \eta) \tag{3.17}$$

The downside to this equation is that it requires eigenvalues to be positive. Therefore non-flow correlations cannot be calculated by this analysis. To achieve a value for anisotropic flow for one particle, $V_n^{(\alpha)}$ is divided by the average multiplicity.

$$v_n^{(\alpha)}(p_T, \eta) \equiv \frac{V_n^{(\alpha)}(p_T, \eta)}{\langle m(p_T, \eta)\rangle} \tag{3.18}$$

Just as in equation 3.9, $m(p_T, \eta)$ is the multiplicity per bin. Evaluation of this equation shows that for $n = 0$ relative multiplicity fluctuations can be studied and that for this thesis, $n = 2$ gives the various components for elliptic flow.

3.2.3 Factorization breaking

The small factorization breaking effects[9], which were briefly touched upon in chapter 1 can be conveniently explained by the PCA components. A measure for this factorization breaking is introduced in [18].

$$r_n(p_T^a, p_T^b) \equiv \frac{V_{n\Delta}(p_T^a, p_T^b)}{\sqrt{V_{n\Delta}(p_T^a, p_T^a) V_{n\Delta}(p_T^b, p_T^b)}} \tag{3.19}$$

[1]This is the bin width and either 4.8 or 1.6, depending on which dependence is assessed.

r_n can take values between -1 and 1 and is ± 1 for factorization (i.e. $k = 1$ in 3.13). Implementing equation 3.16 with only $\alpha = 1, 2$ and taking a Taylor approximation for $v_n^{(1)} \gg v_n^{(2)}$, gives:

$$r_n(p_T^a, p_T^b) \approx 1 - \frac{1}{2}\left(\frac{v_n^{(2)}(p_T^a)}{v_n^{(1)}(p_T^a)} - \frac{v_n^{(2)}(p_T^b)}{v_n^{(1)}(p_T^b)}\right)^2 \tag{3.20}$$

Naturally, this equation also holds true for η-dependence. Therefore PCA provides a natural way of explaining and quantifying factorization breaking in nuclear collissions.

Chapter 4

Results

4.1 Comparison between PCA and Q-cumulants

Fig.4.1 shows the ratio between the principal component analysis and Q-cumulants analysis of the second flow coefficient (elliptic flow) for both p_T and η at $\sqrt{s_{NN}} = 2.76$ TeV in high-multiplicity PbPb collisions. Only 20-30% centrality is shown[1]; in all other centralities the ratios are compatible. For PCA the outcome of equation 3.18 with $\alpha = 1$ and $n = 2$ is used. For the Q-cumulants approach, equation 3.12 is taken with $n = 2$.

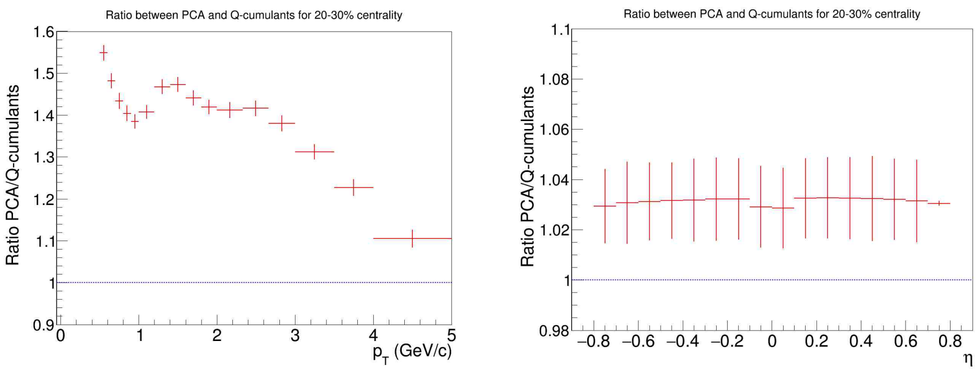


Figure 4.1 – Ratio between PCA and Q-cumulants analysis outcome for v_2 as a function of p_T (left) and η (right) in multiple centrality windows at $\sqrt{s_{NN}} = 2.76$TeV.

From Fig. 4.1, it can be seen that both analyses display the same dependence, differing less than 6% and 60% as a function of η and p_T, respectively. The two analyses do not necessarily have to give the same result, as underlying flow fluctuations are taken into account in different ways. Nevertheless, a close resemblance is expected and also found.

4.2 PCA of pseudorapidity

Leading and sub-leading components of the PCA are shown in Fig.4.2 for central (0-10%) up to peripheral (50-60%) collisions with their η-dependences in the $|\eta| < 0.8$ window at a binsize of 0.1. This results in a 16×16-matrix, corresponding to the η-bins and accordingly, 16 eigenmodes. In general, there is a strong ordering between leading and sub-leading eigenvalues: $\lambda^{(1)}/\lambda^{(2)} \sim 4$.

[1] All centralities can be found in Appendix B

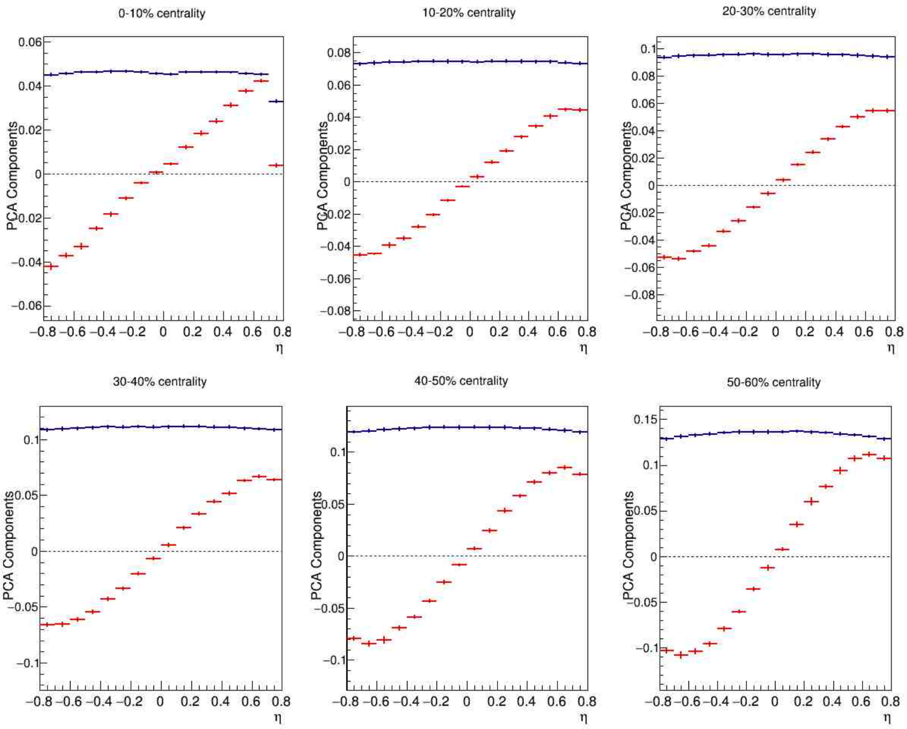


Figure 4.2 – Leading ($\alpha = 1$) and sub-leading ($\alpha = 2$) modes of the principal component analysis of elliptic flow ($n = 2$) for multiple centrality windows as a function of pseudorapidity at $\sqrt{s_{NN}} = 2.76$ TeV for PbPb collisions. The error bars correspond to statistical uncertainties.

From Figures 4.2 and 4.3 it is clear that the principal components are orthogonal, since the modes are sinusoidal with increasing number of nodes: $v_2^{(\alpha)} \sim \sin((\alpha - 1)\eta))$.

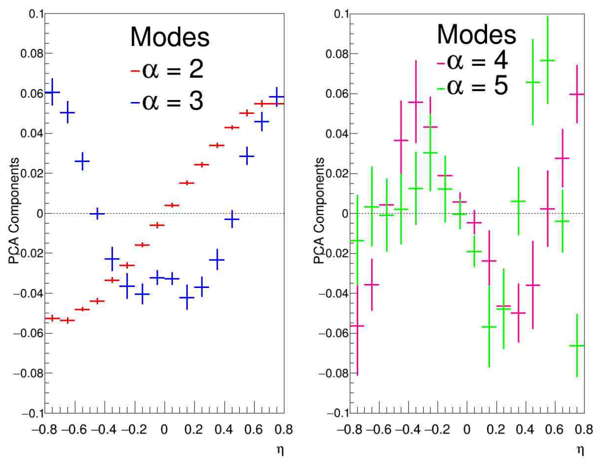


Figure 4.3 – Multiple ($2 \leq \alpha \leq 5$) modes of the principal component analysis of elliptic flow for the 20-30% centrality range as a function of pseudorapidity. The error bars are statistical uncertainties.

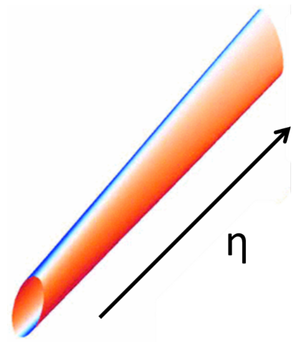


Figure 4.4 – Schematic representation of torque effects due to forward/backward moving particles adapted from [19]. The principal axes of the initial elliptical volume change from negative to positive pseudorapidity.

For $\alpha \geq 3$, eigenvalues show an ordering of $\lambda^{(\alpha)}/\lambda^{(\alpha+1)} \sim 1.005$, which is small. This can be attributed to the contribution of non-flow at small values of relative pseudorapidity $\Delta\eta$. As can be seen in Fig.4.3, flow components for $\alpha \geq 3$ become less distinct due to increasingly large statistical errors. Therefore they are left out in further discussion. A full list of eigenvalues for η can be found in table A.2 in Appendix A. For η-dependence, no eigenvalues are negative in this case.

Equation 3.13 doesn't assign a sign, which means a choice has to be made regarding parity for $v_2^{(\alpha)}$. In Fig.4.2 $v_2^{(2)}$ is chosen positive for positive pseudorapidity. The asymmetric rapidity shape of $\alpha = 2$ is attributed to event-by-event torque effects due to initial fluctuations of the nuclei densities and the preference for forward (backward) moving particles to keep moving forward (backward) after collision[19]. This is schematically represented in Fig.4.4. This results in a difference of principal axis of the transverse momentum distributions of the detected particles as a function of η. Therefore the sub-leading component changes sign from negative to positive pseudorapidity. With this component, the torque effect can be measured experimentally.

4.3 PCA of transverse momentum

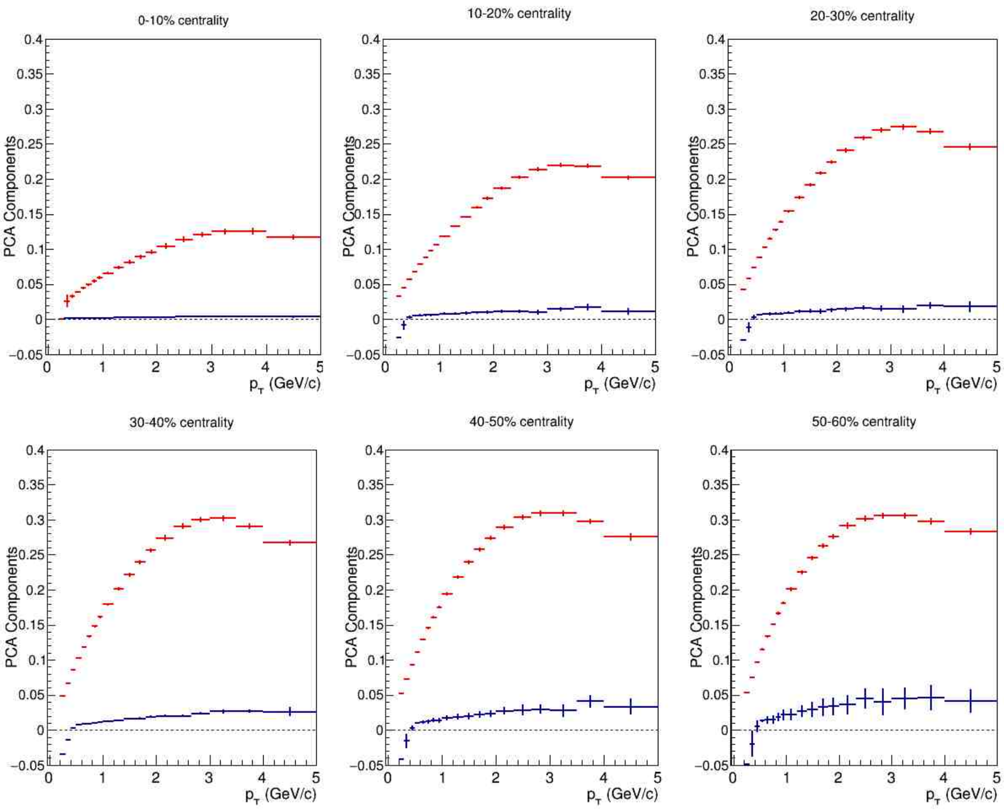


Figure 4.5 – Leading ($\alpha = 1$) and sub-leading ($\alpha = 2$) modes of the principal component analysis of elliptic flow ($n = 2$) for multiple centrality windows as a function of transverse momentum at $\sqrt{s_{NN}} = 2.76$ TeV for PbPb collisions. Bins for low p_T are smaller due to higher multiplicity. The error bars correspond to statistical uncertainties.

In Fig.4.5 the first two components ($\alpha = 1, 2$) are shown for the PCA as a function of p_T for centrality windows between 0-60%. The p_T-range from 0.2 to 5 GeV is considered with the conventional choice of bin size, which minimizes multiplicity variations between the bins. This leads to a 19×19-correlationmatrix with 19 corresponding eigenvalues.

Depending on centrality, 3-5 modes are unusable as a result of negative eigenvalues (table A.1 in Appendix A). The origin of these modes is not fully understood. The eigenvalues show for $\alpha \geq 2$ a strong ordering for all centrality ranges: $\lambda^{(\alpha)}/\lambda^{(\alpha+1)} \sim 1.65$. This suggests a long range correlation in Δp_T, as opposed to what is the case for pseudo-rapidity. The ratio between the leading and sub-leading modes (~ 30) indicate that p_T-dependence of elliptic flow is less influenced by flow fluctuations than for η. Furthermore, $v_2^{(2)}$ increases for large p_T and more peripheral collisions, just as its leading mode.

This can be attributed to event-by-event fluctuations in the initial density distributions. These density inhomogenities cause the anisotropic flow angles to vary with p_T[10][11]. The magnitude of $v_2^{(2)}$ gives a measure of this variance, which is 5-15% for high p_T according to Fig.4.5. As can be seen in Fig.4.6, statistical errors increase for higher modes, but nevertheless for $2 \leq \alpha \leq 5$ the same dependence is visible for all centralities (only 20-30% is shown).

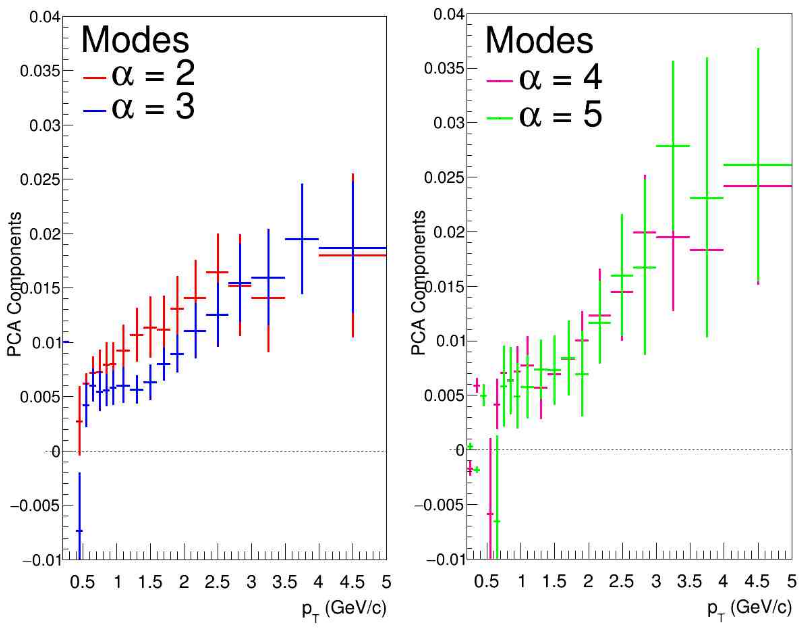


Figure 4.6 – Multiple ($2 \leq \alpha \leq 5$) modes of the principal component analysis of elliptic flow for the 20-30% centrality range as a function of transverse momentum. The error bars are statistical uncertainties.

4.4 Factorization breaking

The factorization breaking coefficient, r_2, from equation 3.20 quantifies the influences of sub-leading modes on flow fluctuations. As can be seen in Fig.4.7, factorization breaking takes place for $\eta \neq 0$ and increases with η for all centralities. This is expected, since Fig. 4.2 shows that $v_2^{(2)}$ is significant for large $|\eta|$. There is no clear dependence between the magnitude of r_2 and the centrality. It seems that for more peripheral collisions, the factorization breaking is larger. However, the centrality window 0-10% contradicts this. Nonetheless, the shape of the function agrees with cumulative measures of the torque in [19], which used Monte Carlo simulations.

In Fig.4.8, it can be seen that r_2 becomes significantly different from 1 for large p_T. This corresponds with the findings in Fig.4.5, where $v_2^{(2)}$ increases with p_T. Just as in Fig.4.7, r_2 seems to become smaller for larger centralities with the exception of the most head-on collisions (0-10%). The statistical errors are also very large in the 0-10%

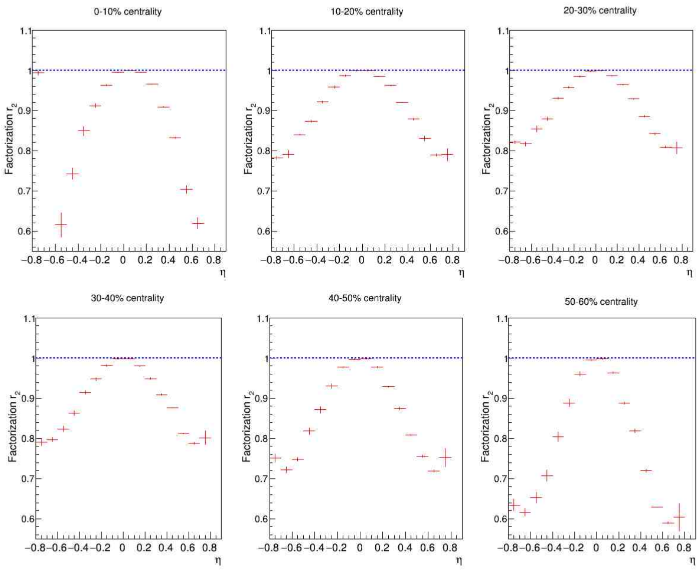


Figure 4.7 – Factorization breaking coefficient r_2 as a function of η for centralities 0-60%. The thin bars correspond to the error propagation of the statistical uncertainties of the variables in equation 3.20.

centrality. Therefore it can be concluded that these results are not fully usable (at least in the 0-10% centrality window) and more research is needed to get a clear view of the factorization breaking calculations with PCA. Nevertheless, the shape and magnitudes correspond with theoretical models in [10][2] of PbPb collisions at $\sqrt{s_{NN}} = 2.76$ TeV.

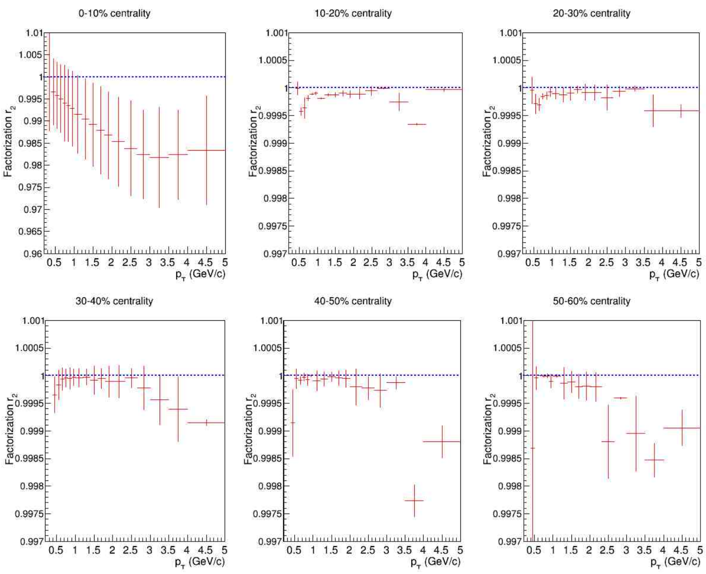


Figure 4.8 – Factorization breaking coefficient r_2 as a function of p_T for centralities 0-60%. The thin bars correspond to the error propagation of the statistical uncertainties of the variables in equation 3.20.

[2]In this paper MC-Glauber and MC-KLN models are used.

Chapter 5

Conclusion

PCA uses all the information available from two-particle correlations and may thus provide a new way to understand anisotropic flow and its inherent fluctuations. Factorization breakdown of flow can now be explained by the interplay between the leading and subleading modes. The latter reveal, for both transverse momentum and pseudorapidity, a significant effect, which can be attributed to two physical processes in the initial state. These processes are the density inhomogenities of the two colliding nuclei and torque effects on the principal axis of the matter after collision. They can now be measured and examined in further research. Also a way to quantify the factorization breaking is given, which flows directly from the components of the PCA.

To summarize, principal components analysis may give an unique insight into the fluctuations of anisotropic flow in heavy-ion collisions, which, although being identified since several years, influence the measurements of flow coefficients in a way which is still under debate. PCA consists of a new way to calculate anisotropic flow with the same precision as previous methods and provides further observables, the sub-leading modes, to quantify its fluctuations. Although the interpretation of PCA results is still not fully understood, this analysis may open the way to numerous new experimental and theoretical programs investigating flow in heavy-ion collisions.

Appendix A

Tables with eigenvalues

Table A.1 – Eigenvalues for p_T

	0-10%	10-20%	20-30%	30-40%	40-50%	50-60%
0	254.329	7.13575	5.4371	3.02831	1.28898	0.426696
1	4.82651	0.262705	0.170246	0.105673	0.0599403	0.0299446
2	0.26827	0.157493	0.103236	0.0571476	0.0307422	0.0136895
3	0.162848	0.0973572	0.0615172	0.0331436	0.0178535	0.00754277
4	0.0976929	0.0580559	0.0348233	0.0202198	0.010328	0.00421407
5	0.0593541	0.0377274	0.021579	0.0117402	0.00611057	0.00266755
6	0.0366464	0.022406	0.013738	0.00765769	0.00342419	0.00162917
7	0.0237037	0.0148542	0.00857553	0.00522276	0.00205772	0.000848191
8	0.0164859	0.0113767	0.00637626	0.00371307	0.00149081	0.000418766
9	0.0123858	0.0073828	0.00429559	0.00177498	0.00094343	0.000356879
10	0.00680651	0.00523759	0.00266756	0.00129795	0.000738205	0.000234678
11	0.0036665	0.00245202	0.0014237	0.000432613	0.000286046	9.63654e-05
12	0.000819913	0.000793954	0.000504419	0.000243401	0.000118029	$5.50477{\cdot}10^{-5}$
13	0.000468831	0.000495992	0.000368334	$5.92212{\cdot}10^{-5}$	$7.22258{\cdot}10^{-5}$	$1.96098{\cdot}10^{-5}$
14	0.000276377	0.000266515	0.000178831	$4.12965{\cdot}10^{-5}$	$1.84579{\cdot}10^{-5}$	$-1.50037{\cdot}10^{-7}$
15	0.000132403	0.000165013	$6.02127{\cdot}10^{-5}$	$-9.383{\cdot}10^{-6}$	$1.58659\text{e}{\cdot}10^{-6}$	$-6.06382{\cdot}10^{-6}$
16	$1.61483{\cdot}10^{-5}$	$-1.19513{\cdot}10^{-5}$	$1.02768\text{e}{\cdot}10^{-5}$	$-3.12189{\cdot}10^{-5}$	$-2.62834{\cdot}10^{-5}$	$-9.57577{\cdot}10^{-6}$
17	$-3.535{\cdot}10^{-5}$	$-6.08836{\cdot}10^{-5}$	$-3.06824{\cdot}10^{-5}$	-0.000145367	$-4.27281\text{e}{\cdot}10^{-5}$	$-1.56544{\cdot}10^{-5}$
18	-0.000281056	-0.000293957	-0.000155489	-0.00018148	$-8.46634{\cdot}10^{-5}$	$-6.22198{\cdot}10^{-5}$

Table A.2 – Eigenvalues for η

	0-10%	10-20%	20-30%	30-40%	40-50%	50-60%
0	4.30922	5.22043	3.89613	2.21133	0.999875	0.379517
1	1.42226	0.973763	0.658106	0.428748	0.258765	0.145058
2	1.32845	0.896259	0.605666	0.394358	0.239905	0.135148
3	1.3263	0.883966	0.594723	0.387293	0.234823	0.131798
4	1.3236	0.881328	0.591983	0.384083	0.232477	0.130395
5	1.31854	0.880426	0.591091	0.382939	0.232101	0.129829
6	1.3158	0.878087	0.588432	0.380747	0.230835	0.129222
7	1.31184	0.877339	0.58817	0.380299	0.229897	0.129149
8	1.30868	0.87655	0.585348	0.37923	0.229211	0.128699
9	1.30287	0.874389	0.584903	0.378694	0.228967	0.128487
10	1.29787	0.872756	0.583601	0.37775	0.228552	0.12811
11	1.29431	0.87083	0.582601	0.377259	0.228389	0.12802
12	1.28989	0.868432	0.581742	0.376398	0.227708	0.127852
13	1.26567	0.866306	0.580212	0.375922	0.227569	0.127705
14	1.25743	0.857387	0.577004	0.374105	0.22617	0.126917
15	0.315656	0.849372	0.570981	0.369998	0.223991	0.125792

Appendix B

Ratios between PCA and Q-cumulants

Figure B.1 – Ratios between PCA and Q-cumulants for p_T-dependence for the 0-60% centrality window.

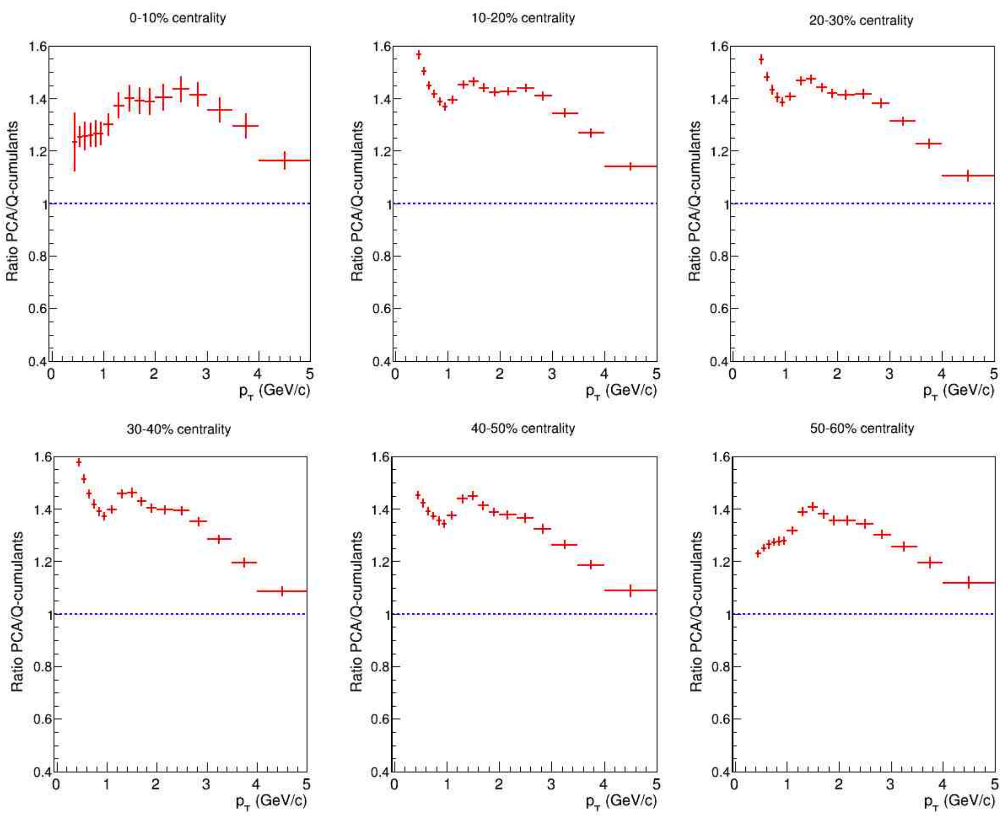
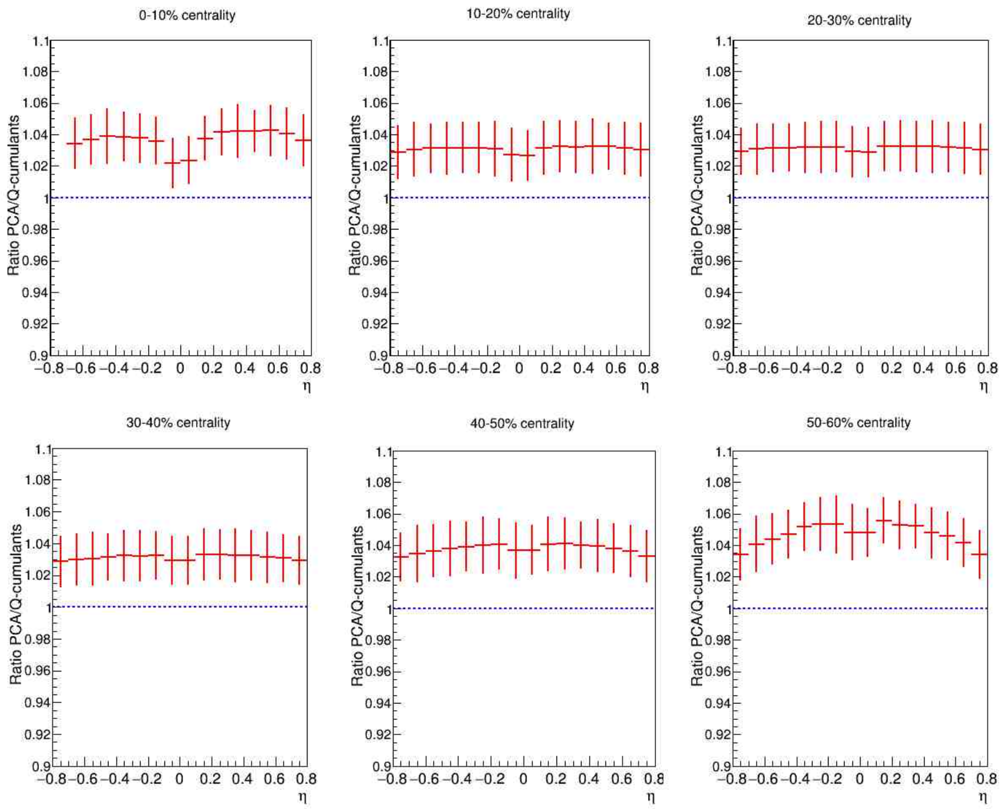


Figure B.2 – Ratios between PCA and Q-cumulants for η-dependence for the 0-60% centrality window.



Appendix C

Analysis code

C.1 PCA Code

Only the code for transverse momentum is shown, but for pseudorapidity the code is the same. The calculations come from [12] and [13].

```
// Constants
const Int_t sigmaevents = 1000000; // Number of events in one subcalculation
const Int_t sigmaN = 16; // Number of subcalculations to determine the statistical variation

// Parameters
Double_t Q1Re[nPtBins] = {}, // Real part Q-vector per event
           Q1Im[nPtBins] = {}, // Imaginary part Q-vector per event
           Q1ReTo[nPtBins][nCentralityRanges] = {}, // Real part Q-vector total
           Q1ImTo[nPtBins][nCentralityRanges] = {}, // Imaginary part Q-vector total
           sigmaQ1ReTo[nPtBins][sigmaN][nCentralityRanges] = {}, // Real part Q-vector subtotal
           sigmaQ1ImTo[nPtBins][sigmaN][nCentralityRanges] = {}, // Imaginary part Q-vector subtotal
           sigmaV2Delta[nCentralityRanges][sigmaN][nPtBins*nPtBins] = {}, // Correlationmatrix for subcalculation
           SubEllipticFlow[nPtBins][nPtBins][sigmaN][nCentralityRanges] = {}, // Elliptic flow for subcalculation
           V2Delta[nCentralityRanges][nPtBins*nPtBins] = {}, // Correlationmatrix
           normalization[nPtBins][nPtBins], // Normalization
           flow[nPtBins][nPtBins][nCentralityRanges] = {}; // Elliptic flow
Int_t  QTo[nPtBins] = {}, // Multiplicity counter per event
           Multiplicity[nPtBins][nCentralityRanges] = {}, // Total Multiplicity
           sigmaMultiplicity[nPtBins][sigmaN][nCentralityRanges] = {}, // Total Multiplicity per subcalculation
           sigma = 0; //Counter for subcalculations

  // Normalization determination
for (Int_t r = 0; r<nPtBins; r++) {
  for (Int_t c = 0; c<nPtBins; c++) {
    normalization[r][c] = TMath::Power(2*pi*PtRange*EtaRange,2);
  }
}

// For-loop over the events (n)
for(Int_t n = 0; n < nEvents; n++) {
  inTree->GetEntry(n);
  if (n%sigmaevents==0 && n!=0)
    sigma += 1;

  // Centrality binning
  Double_t centrality = eventData->v0Cent;
  if (centrality>=60)
    continue;
  Int_t binnc = (centrality/10);

  // For-loop over the Tracks (i)
  for(Int_t i = 0; i < nTracks; i++) {

        // Track selection and pt binning
    if (Ti->pt>5)
      continue;
    if (Ti->eta > 0.8 || Ti->eta <-0.8)
      continue;
    for (Int_t k = 1; k <= nPtBins; k++) {
      if (Ti->pt < PtBin[k] && Ti->pt > PtBin[k-1])
        Int_t binn = k-1;
    }

      Q1Re[binn] += TMath::Cos(2*Ti->phi);
      Q1Im[binn] += TMath::Sin(2*Ti->phi);
      Q1ReTo[binn][binnc] += TMath::Cos(2*Ti->phi);
      Q1ImTo[binn][binnc] += TMath::Sin(2*Ti->phi);
      QTo[binn] += 1;
      sigmaQ1ReTo[binn][sigma][binnc] += TMath::Cos(2*Ti->phi);
      sigmaQ1ImTo[binn][sigma][binnc] += TMath::Sin(2*Ti->phi);
  } // end of loop over i

  // Correlationmatrix
  for (Int_t r = 0; r<nPtBins; r++) {
    for (Int_t c = 0; c<nPtBins; c++) {
      if (r==c) { // Multiplicity correction
        V2Delta[binnc][r*nPtBins+c] += (Q1Re[r]*Q1Re[c] + Q1Im[r]*Q1Im[c] - QTo[r])/normalization[r][c];
        sigmaV2Delta[binnc][sigma][r*nPtBins+c] += (Q1Re[r]*Q1Re[c] + Q1Im[r]*Q1Im[c] - QTo[r])/normalization[r][c];
      }
      else {
        V2Delta[binnc][r*nPtBins+c] += (Q1Re[r]*Q1Re[c] + Q1Im[r]*Q1Im[c])/normalization[r][c];
        sigmaV2Delta[binnc][sigma][r*nPtBins+c] += (Q1Re[r]*Q1Re[c] + Q1Im[r]*Q1Im[c])/normalization[r][c];
      }
    } // end of loop over c
  } // end of loop over r
```

```
      // empty Q-vectors
      for (Int_t m = 0; m<nPtBins; m++) {
          Multiplicity[m][binnc] += QTo[m];
          sigmaMultiplicity[m][sigma][binnc] += QTo[m];
          Q1Re[m] = 0;
          Q1Im[m] = 0;
          QTo[m] = 0;
      } // end of loop over m
    } // end of loop over n

    // For-loop over the centrality ranges (ce)
    for(Int_t ce = 0; ce < nCentralityRanges; ce++) {
      for (Int_t r = 0; r<nPtBins; r++) { //Azimuthal correction on correlationmatrix
        for (Int_t c = 0; c<nPtBins; c++) {
          V2Delta[ce][r*nPtBins+c] = V2Delta[ce][r*nPtBins+c]/nEvents-(Q1ReTo[r][ce]*Q1ReTo[c][ce]+Q1ImTo[r][ce]*Q1ImTo[c
        ][ce])/(nEvents*nEvents);
          for (Int_t s = 0; s<(sigmaN-1); s++) {
            sigmaV2Delta[ce][s][r*nPtBins+c] = sigmaV2Delta[ce][s][r*nPtBins+c]/sigmaevents-(sigmaQ1ReTo[r][s][ce]*
        sigmaQ1ReTo[c][s][ce]+sigmaQ1ImTo[r][s][ce]*sigmaQ1ImTo[c][s][ce])/(sigmaevents*sigmaevents*normalization[r][c]);
               }
          sigmaV2Delta[ce][sigmaN-1][r*nPtBins+c] = sigmaV2Delta[ce][sigmaN-1][r*nPtBins+c]/sigmaevents-(sigmaQ1ReTo[r][
        sigmaN-1][ce]*sigmaQ1ReTo[c][sigmaN-1][ce]+sigmaQ1ImTo[r][sigmaN-1][ce]*sigmaQ1ImTo[c][sigmaN-1][ce])/((nEvents%
        sigmaevents)*(nEvents%sigmaevents)*normalization[r][c]);
        }
      }

      // Subcalculations to determine the variation in Elliptic flow
      for (Int_t s = 0; s<sigmaN; s++) {
        TMatrixDSym sigmacorrelationmatrix(nPtBins,sigmaV2Delta[ce][s]);
        TMatrixDSymEigen sigmaeigenmatrix(sigmacorrelationmatrix);
        TVectorD sigmaeigenval = sigmaeigenmatrix.GetEigenValues();
        TMatrixD sigmaeigenvec = sigmaeigenmatrix.GetEigenVectors();
        for (Int_t m = 0; m<nPtBins; m++) {
          for (Int_t a = 0; a<nPtBins; a++) {
            if (sigmaeigenval(a)>=0 && sigmaMultiplicity[m][s][ce]>0) { //negative eigenvalues are mostly due to non-flow
              if (s!=(sigmaN-1))
                SubEllipticFlow[a][m][s][ce] = (TMath::Sqrt(sigmaeigenval(a)) * sigmaeigenvec(m,a) * sigmaevents)/(
        sigmaMultiplicity[m][s][ce]);
              else
                SubEllipticFlow[a][m][s][ce] = (TMath::Sqrt(sigmaeigenval(a)) * sigmaeigenvec(m,a) * (nEvents%sigmaevents)
        )/(sigmaMultiplicity[m][s][ce]);
            } // end of if
          } // end of loop over a
          SubEllipticFlow1[ce]->Fill(PtBin[m],SubEllipticFlow[0][m][s][ce]);
          SubEllipticFlow2[ce]->Fill(PtBin[m],SubEllipticFlow[1][m][s][ce]);
          SubEllipticFlow3[ce]->Fill(PtBin[m],SubEllipticFlow[2][m][s][ce]);
          SubEllipticFlow4[ce]->Fill(PtBin[m],SubEllipticFlow[3][m][s][ce]);
          SubEllipticFlow5[ce]->Fill(PtBin[m],SubEllipticFlow[4][m][s][ce]);
        } // end of loop over m
      } // end of loop over s

      // Matrix eigenvector calculations
      TMatrixDSym correlationmatrix(nPtBins,V2Delta[ce]);
      TMatrixDSymEigen eigenmatrix(correlationmatrix);
      TVectorD eigenval = eigenmatrix.GetEigenValues();
      TMatrixD eigenvec = eigenmatrix.GetEigenVectors();

      // Elliptic Flow calculations
      for (Int_t m = 0; m<nPtBins; m++) {
        for (Int_t a = 0; a<nPtBins; a++) {
          if (eigenval(a)>=0 && Multiplicity[m][ce]>0) { //negative eigenvalues are mostly due to non-flow
            flow[a][m][ce] = (TMath::Sqrt(eigenval(a)) * eigenvec(m,a) * nEvents)/(Multiplicity[m][ce]);
          } // end of if
        } // end of loop over a
        EllipticFlowComponent1[ce]->SetBinContent(m+1,flow[0][m][ce]);
        EllipticFlowComponent2[ce]->SetBinContent(m+1,flow[1][m][ce]);
        EllipticFlowComponent3[ce]->SetBinContent(m+1,flow[2][m][ce]);
        EllipticFlowComponent4[ce]->SetBinContent(m+1,flow[3][m][ce]);
        EllipticFlowComponent5[ce]->SetBinContent(m+1,flow[4][m][ce]);
        EllipticFlowComponent1[ce]->SetBinError(m+1,SubEllipticFlow1[ce]->GetBinError(m));
        EllipticFlowComponent2[ce]->SetBinError(m+1,SubEllipticFlow2[ce]->GetBinError(m));
        EllipticFlowComponent3[ce]->SetBinError(m+1,SubEllipticFlow3[ce]->GetBinError(m));
        EllipticFlowComponent4[ce]->SetBinError(m+1,SubEllipticFlow4[ce]->GetBinError(m));
        EllipticFlowComponent5[ce]->SetBinError(m+1,SubEllipticFlow5[ce]->GetBinError(m));
      } // end of loop over m
    } // end of loop over ce
```

C.2 Q-cumulants code

Again only the code for transverse momentum is shown, but with minimum effort the code for pseudorapidity can be obtained. The code is based on the calculations in [5] and [6].

```
Double_t  Q1Re = 0, //Real part Q-vector reference flow
              Q1Im = 0, // Imaginary part Q-vector reference flow
              qpt1Re[nPtBins] = {}, // Real part Q-vector differential flow
              qpt1Im[nPtBins] = {}; // Imaginary part Q-vector differential flow
              azirefre[nCentralityRanges] = {}, //Real part azimuthal correction reference flow
              azirefim[nCentralityRanges] = {}, //Imaginary part azimuthal correction reference flow
              azirefM[nCentralityRanges] = {}, // Multiplicity counter azimuthal correction reference flow
              aziref[nCentralityRanges] = {}, // Azimuthal correction reference flow
              azidifre[nPtBins][nCentralityRanges] = {}, // Real part azimuthal correction differential flow
              azidifim[nPtBins][nCentralityRanges] = {}, // Imaginary part azimuthal correction differential flow
              azidifM[nPtBins][nCentralityRanges] = {}, // Multiplicity counter all events azimuthal correction
    differential flow
              azidif[nPtBins][nCentralityRanges] = {}; // Azimuthal correction differential flow
Int_t   Q1M = 0, // Multiplicity counter per event reference flow
              qpt1M[nPtBins] = {}; // Multiplicity counter per event differential flow

// Sums needed to calculate errors and weights
Double_t sum1[nPtBins][nCentralityRanges] = {},
           sum2[nPtBins][nCentralityRanges] = {},
           sum3[nPtBins][nCentralityRanges] = {},
           sum4[nPtBins][nCentralityRanges] = {},
           sum5[nPtBins][nCentralityRanges] = {},
           sumw1[nPtBins][nCentralityRanges] = {},
           sumw2[nPtBins][nCentralityRanges] = {},
           sumw3[nPtBins][nCentralityRanges] = {},
           sumw4[nPtBins][nCentralityRanges] = {},
           sumw5[nPtBins][nCentralityRanges] = {};

// For-loop over the events (n)
for(Int_t n = 0; n < nEvents; n++) {

  inTree->GetEntry(n);
  // Centrality binning
  Double_t centrality = eventData->v0Cent;
  if (centrality >= 60)
    continue;
  Int_t binnc = (centrality/10);

  // For-loop over the Tracks (i)
  for(Int_t i = 0; i < nTracks; i++) {

    // Track selection and pt binning
    if (Ti->pt>5)
      continue;
    if (Ti->eta > 0.8 || Ti->eta <-0.8)
      continue;
    for (Int_t k = 1; k <= nPtBins; k++) {
      if (Ti->pt < PtBin[k] && Ti->pt > PtBin[k-1])
        Int_t binn = k-1;
    }

    Q1Re += TMath::Cos(2*Ti->phi);
    Q1Im += TMath::Sin(2*Ti->phi);
    Q1M += 1;
    qpt1Re[binn] += TMath::Cos(2*Ti->phi);
    qpt1Im[binn] += TMath::Sin(2*Ti->phi);
    qpt1M[binn] += 1;

  } // end of loop over i

  if (Q1M > 1) {
    Double_t differential[nPtBins]; // differential flow
    Double_t reference = (Q1Re*Q1Re + Q1Im*Q1Im)/(Q1M*Q1M-Q1M); // reference flow
    azirefre[binnc] += Q1Re;
    azirefim[binnc] += Q1Im;
    azirefM[binnc] += Q1M;

    // For-loop over the bins (m)
      for (Int_t m = 0; m<nPtBins; m++) {
      ReferencePlot[binnc]->Fill(PtBins[m],reference,Q1M*Q1M);
      if (qpt1M[m] > 1) {
        azidifre[m][binnc] += qpt1Re[m];
        azidifim[m][binnc] += qpt1Im[m];
        azidifM[m][binnc] += qpt1M[m];
        differential[m] = (qpt1Re[m]*Q1Re + qpt1Im[m]*Q1Im)/(qpt1M[m]*Q1M-qpt1M[m]);
        DifferentialPlot[binnc]->Fill(PtBins[m],differential[m],qpt1M[m]*Q1M-qpt1M[m]);
        } // end of if
    } // end of loop over m
      } // end of if

  // empty q-vectors
  Q1Re = 0;
  Q1Im = 0;
  Q1M = 0;
  for (Int_t m = 0; m<nPtBins; m++) {
    qpt1Re[m] = 0.;
    qpt1Im[m] = 0.;
    qpt1M[m] = 0.;
  }
} // end of loop over n

// For-loop over the centrality ranges (c)
  // Loop is needed to add the azimuthal correction
for(Int_t c = 0; c < nCentralityRanges; c++) {

  if (azirefM[c] > 0)
    aziref[c] = (azirefre[c]*azirefre[c])/(azirefM[c]*azirefM[c]) + (azirefim[c]*azirefim[c])/(azirefM[c]*azirefM[c]);
  else
    aziref[c] = 0;
  Double_t Reference[nCentralityRanges]; // Total Reference Flow
  Reference[c] = ReferencePlot[c]->GetBinContent(3);
  Reference[c] = Reference[c] - aziref[c];

  // For-loop over the bins (m)
```

```
    for (Int_t m = 0; m<nPtBins; m++) {
      if(azidifM[m][c] > 0 && azirefM[c] > 0)
        azidif[m][c] = (azidifre[m][c]*azirefre[c])/(azidifM[m][c]*azirefM[c]) + (azidifim[m][c]*azirefim[c])/(azidifM[m
      ][c]*azirefM[c]);
      else
        azidif[m][c] = 0;
      Double_t Differential[nCentralityRanges]; // Total Differential Flow
      Differential[c] = DifferentialPlot[c]->GetBinContent(m);
      Differential[c] = Differential[c] - azidif[m][c];
      TotalDifferentialPlot[c]->SetBinContent(m+1,Differential[c]);
      TotalReferencePlot[c]->SetBinContent(m+1,Reference[c]);
    } // end of loop over m
  } // end of loop over c

  // For-loop over the events (n)
  for(Int_t n = 0; n < nEvents; n++) {

    inTree->GetEntry(n);
    Double_t centrality = eventData->v0Cent;
    if (centrality >= 60)
      continue;
    Int_t binnc = (centrality/10);

    // For-loop over the Tracks (i)
    for(Int_t i = 0; i < nTracks; i++) {

      // Track selection and pt binning
      if (Ti->pt>5)
        continue;
      if (Ti->eta > 0.8 || Ti->eta <-0.8)
        continue;
      for (Int_t k = 1; k <= nPtBins; k++) {
        if (Ti->pt < PtBin[k] && Ti->pt > PtBin[k-1])
          Int_t binn = k-1;
      }
        Q1Re += TMath::Cos(2*Ti->phi);
        Q1Im += TMath::Sin(2*Ti->phi);
        Q1M += 1;
        qpt1Re[binn] += TMath::Cos(2*Ti->phi);
        qpt1Im[binn] += TMath::Sin(2*Ti->phi);
        qpt1M[binn] += 1;
    } // end of loop over i
    if (Q1M > 1) {
      Double_t differential[nPtBins]; // Differential flow per event
      Double_t difweight[nPtBins]; // weight of differential flow
      Double_t refweight = Q1M*Q1M; // weight of reference flow
      Double_t reference = (Q1Re*Q1Re + Q1Im*Q1Im)/(Q1M*Q1M-Q1M); // Reference flow per event

      // For-loop over the bins (m)
      for (Int_t m = 0; m<nPtBins; m++) {
        if (qpt1M[m] > 1) {
          differential[m] = (qpt1Re[m]*Q1Re + qpt1Im[m]*Q1Im)/(qpt1M[m]*Q1M-qpt1M[m]);
          difweight[m] = qpt1M[m]*Q1M-qpt1M[m];
          // calculation of the sums needed for the errors
          sum2[m][binnc] += (difweight[m])*(differential[m]-(TotalDifferentialPlot[binnc]->GetBinContent(m)))*(
      differential[m]-(TotalDifferentialPlot[binnc]->GetBinContent(m)));
          sum5[m][binnc] += (difweight[m])*(differential[m]);
          sumw1[m][binnc] += (difweight[m])*(difweight[m]);
          sumw2[m][binnc] += difweight[m];
          sum1[m][binnc] += (refweight)*(reference-(TotalReferencePlot[binnc]->GetBinContent(m)))*(reference-(
      TotalReferencePlot[binnc]->GetBinContent(m)));
          sum3[m][binnc] += (refweight)*(difweight[m])*(reference)*(differential[m]);
          sum4[m][binnc] += (refweight)*(reference);
          sumw3[m][binnc] += (refweight)*(refweight);
          sumw4[m][binnc] += refweight;
          sumw5[m][binnc] += (difweight[m])*(refweight);
                } // end of if
      } // end of loop over m
        } // end of if
    // empty q-vectors
    Q1Re = 0;
    Q1Im = 0;
    Q1M = 0;
    for (Int_t m = 0; m<nPtBins; m++) {
      qpt1Re[m] = 0.;
      qpt1Im[m] = 0.;
      qpt1M[m] = 0.;
    } // end of loop over m
  } // end of loop over n

  // For-loop over the centrality ranges (c)
  for(Int_t c = 0; c < nCentralityRanges; c++) {

    // For-loop over the bins (m)
    for (Int_t m = 0; m<nPtBins; m++) {
      Double_t V2[nCentralityRanges]; // Elliptic flow
      Double_t error2[nPtBins][nCentralityRanges]; // Variation
      Double_t error[nPtBins][nCentralityRanges]; // Standard deviation
      if (TMath::Sqrt((TotalReferencePlot[c]->GetBinContent(m))) > 0 && (TotalDifferentialPlot[c]->GetBinContent(m)) !=
       0) {
        if (sumw4[m][c]*(sumw4[m][c]*sumw4[m][c]-sumw3[m][c])!=0 && (sum3[m][c]*sumw4[m][c]*sumw2[m][c]-sumw5[m][c]*sum4
      [m][c]*sum5[m][c])!= 0 && (sumw4[m][c]*sumw2[m][c]-sumw5[m][c])*sumw4[m][c]*sumw2[m][c]!=0){
          error2[m][c] = ((TotalDifferentialPlot[c]->GetBinContent(m))*(TotalDifferentialPlot[c]->GetBinContent(m))*
      sumw3[m][c]*sum1[m][c]) / (4*(TotalReferencePlot[c]->GetBinContent(m))*(TotalReferencePlot[c]->GetBinContent(m))*(
      TotalReferencePlot[c]->GetBinContent(m))*sumw4[m][c]*(sumw4[m][c]*sumw4[m][c]-sumw3[m][c])) + (sumw1[m][c]*sum2[m
      ][c]) / ((TotalReferencePlot[c]->GetBinContent(m))*sumw2[m][c]*(sumw2[m][c]*sumw2[m][c]-sumw1[m][c])) - ((
      TotalDifferentialPlot[c]->GetBinContent(m))*(sum3[m][c]*sumw4[m][c]*sumw2[m][c]-sumw5[m][c]*sum4[m][c]*sum5[m][c])
      ) / ((TotalReferencePlot[c]->GetBinContent(m))*(TotalReferencePlot[c]->GetBinContent(m))*(sumw4[m][c]*sumw2[m][c]-
      sumw5[m][c])*sumw4[m][c]*sumw2[m][c]);
          error[m][c] = TMath::Sqrt(TMath::Abs(error2[m][c]));
          V2[c] = (TotalDifferentialPlot[c]->GetBinContent(m))/TMath::Sqrt((TotalReferencePlot[c]->GetBinContent(m)));
          EllipticFlowPlot[c]->SetBinContent(m+1,V2[c]);
          EllipticFlowPlot[c]-> SetBinError(m+1,error[m][c]);
        } // end of if for error2
      } //end of if for Ref>0
    } // end of loop over m
  } // end of loop over c
```

Bibliography

[1] J.Y. Ollitrault, "Anisotropy as a signature of transverse collective flow", Phys. Rev. D **46**, 229 (1992).

[2] S.A. Voloshin,A.M. Poskanzer, R.Snellings, "Collective phenomena in non-central nuclear collisions", (2008), [arXiv:0809.2949v2[nucl-ex]].

[3] K.H. Ackermann *et al.*, "Elliptic Flow in Au+Au collisions at $\sqrt{s_{NN}} = 130$ GeV",Phys. Rev. Lett. **86**, 402 (2001), [arXiv:nucl-ex/0009011].

[4] S. Voloshin and Y. Zhang, "Flow Study in Relativistic Nuclear Collisions by Fourier expansion of Azimuthal Particle Distributions", Z.Phys. C **70**, 665 (1996),[arXiv:hep-ph/9407282].

[5] A. Bilandzic, R. Snellings and S. Voloshin, "Flow analyis with cumulants: direct calculations", Phys. Rev. C. **83**, 044913 (2011), [arXiv:1010.0233 [nucl-ex]].

[6] A. Bilandzic, R. Snellings and S. Voloshin, "Flow analyis with cumulants", (2010).

[7] A.M. Poskanzer and S.A. Voloshin, "Methods for analyzing anisotropic flow in relativistic nuclear collisions", Phys. Rev. C. **58**, 1671 (1998), [arXiv:nucl-ex/9805001].

[8] M. Luzum and J.Y. Ollitrault, "Eliminating experimental bias in anisotropic-flow measurements of high-energy nuclear collisions", Phys.Rev. C **87** , 044907 (2013), [arXiv:1209.2323v3 [nucl-ex]].

[9] ALICE Collaboration, "Harmonic decomposition of two-particle angular correlations in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV", Phys. Lett. B **708**, 249 (2012) [arXiv:1109.2501 [nucl-ex]].

[10] U.W. Heinz, Z. Qiu and C. Shen, "Evidence for transverse momentum and pseudorapidity dependent event plane fluctuations in PbPb and pPb collisions", Phys. Rev. C **87**, 034913 (2013), [arXiv:1503.01692v2 [nucl-ex]].

[11] CMS Collaboration, "Fluctuating flow angles and anisotropic flow measurements", Phys. Rev. C **92**, 034911(2015), [arXiv:1302.3535v2 [nucl-th]].

[12] S.R. Bhalerao, J.-Y Ollitrault, S. Pal, and D.Teaney, "Principal component analysis of event-by-event fluctuations", Phys. Rev. Lett. **114**, 152301 (2015),[arXiv:1410.7739v2 [nucl-th]].

[13] CMS Collaboration, "Principal Component Analysis of two-particle azimuthal correlations in PbPb and pPb collisions at CMS," CMS-PAS-HIN-15-010.

[14] U. Heinz and R. Snellings, "Collective flow and viscosity in relativistic heavy-ion collisions", Ann. Rev. Nucl. Part. Sci. **63**, 123 (2013), [arXiv:1301.2826v1 [nucl-th]].

[15] ALICE Collaboration, "Performance of the ALICE Experiment at the CERN LHC", Int. J. Mod. Phys. A **29**, 1430044 (2014), [arXiv:1402.4476[nucl-ex]].

[16] "2010 version hybrid selection", (2010), [ALICE > Hybrid Tracks].

[17] A. Bilandzic, C.H. Cristensen, K. Gulbrandsen, A. Hansen and Y. Zhou, "Generic framework for anisotropic flow analyses with multi-particle azimuthal correlations", Phys. Rev. C **89**, 064904 (2014), [arXiv:1312.3572v2 [nucl-ex]].

[18] F.G. Gardim, F. Grassi, M. Luzum and J.Y. Ollitrault, "Breaking of factorization of two-particle correlations in hydrodynamics", Phys. Rev. C **87**, 031901(R) (2013), [arXiv:1211.0989v2 [nucl-th]].

[19] P. Bozek, W. Broniowski and J. Moreira, "Torqued fireballs in relativistic heavy-ion collisions", Phys. Rev. C **83**, 034911 (2011), [arXiv:1011.3354 [nucl-th]].