Feasibility study of fluctuations in net-proton, and net-charge multiplicity distributions in the CBM experiment

A report Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF SCIENCE

by

DIPTANIL ROY

to the

School of Physical Sciences

National Institute of Science Education and Research

Bhubaneswar

November 18, 2018

ACKNOWLEDGEMENTS

ABSTRACT

We study the feasibility of measuring cumulants of net-proton, and net-charge multiplicity distributions in Au-Au collisions at $E_{lab} = 10$ AGeV in the upcoming Compressed Baryonic Matter (CBM) experiment. The ratio of these cumulants are related to the correlation length of the system, and hence are sensitive to the location of the predicted QCD critical point at finite baryon chemical potentials. The bin width corrected cumulants have been presented for net-proton and net-charge multiplicity distributions, and efficiency corrected results have been presented for net-proton multiplicity distributions only.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Compressed Baryonic Matter experiment (CBM), planned at Facility for Antiproton and Ion Research (FAIR) in Darmstadt, aims at investigating the nuclear matter phase diagram in the region of high net baryonic densities and moderate temperatures. This is complimentary to the experiments at LHC and RHIC energies, which explore the Quantum Chromodynamics (QCD) phase diagram in the transition region between quark-gluon-plasma (QGP) and hadron gas, at vanishing baryon chemical potentials, and high temperatures [1]. A schematic of the QCD phase diagram is shown in Fig.1. At $\mu_B = 0$, it has been found theoretically that there occurs
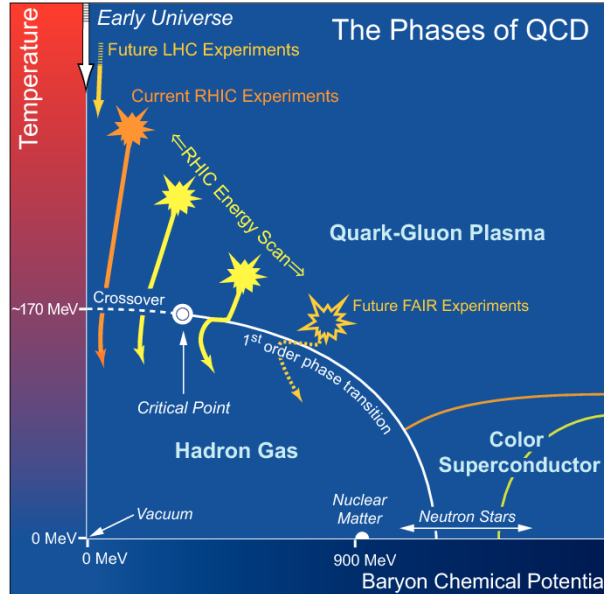
Figure 1.1: Schematic of the QCD Phase Diagram [2]

1

a cross-over transition between hadronic and quark-gluon phases at high temperatures [3]. Recent model calculations have also predicted a QCD critical point (CP) and a first order phase boundary between quark-gluon and hadronic phases at finite μ_B [4]. The experimental discovery of a first order phase transition or a critical point in the QCD phase diagram would significantly enhance our understanding of strong interactions, along with the knowledge of neutron stars, and origin of hadron masses. Among the primary goals of the CBM experiment are the search for the predicted first order phase transition between hadronic matter and partonic matter and the search for the critical point.

A good signature of a phase transition and a CP are non-monotonic variations of observables related to the moments of the distributions of conserved quantities such as net-baryon, net-charge, and net-strangeness number with $\sqrt{s_{NN}}$. The moments are related to the correlation length (ξ) of the system [6], and since the system stays for finite time and is of finite size, it will not diverge, however it will take large values. Following the Beam Energy Scan (BES) at RHIC, the first measurements of the beam energy ($\sqrt{s_{NN}} = 7.7 - 200 \ GeV/c$) and collision centrality dependence of the mean(M), standard deviation (σ), skewness(S), and kurtosis (κ) have been reported for both net charge [5] and net proton multiplicity distributions [6] in Au-Au collisions. These observables have shown a centrality and beam energy dependence for the net-proton distributions, as shown in Fig.1.3, which is neither predicted by non-CP transport model calculations, nor by the Hadron Resonance Gas Model (HRG). For net-charge distributions, σ^2/M is shown to monotonically increase, while the remaining observables have a weak dependence of centrality and beam energy. Within the uncertainties, no non-monotonic behavior had been observed in the products of moments of net-charge as a function of collision energy. The aim of BES-II in RHIC and CBM experiment in FAIR is to look for the non-monotonic behaviour in these

2
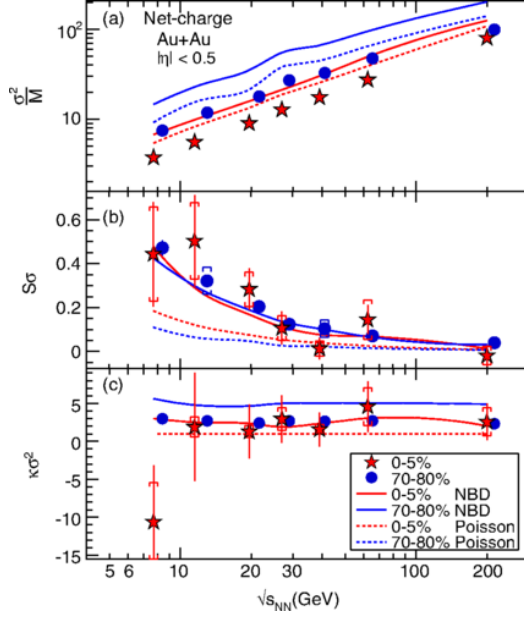
Figure 1.2: Beam Energy Dependence of a) σ^2/M, b)$S\sigma$, and c)$\kappa\sigma^2$, , after all corrections, for most central (0- 5%) and peripheral (70-80%) bins from Au-Au collisions at RHIC. This is taken from [5].
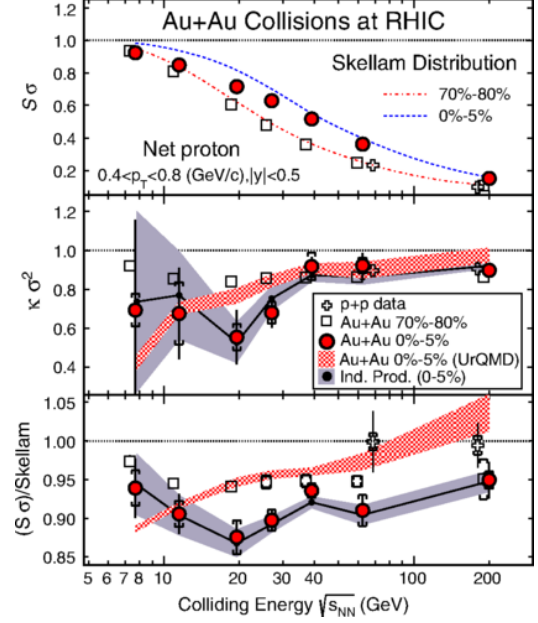
Figure 1.3: Beam Energy Dependence of $S\sigma$, and $\kappa\sigma^2$, , after all corrections, for most central (0- 5%) and peripheral (70-80%) bins from Au-Au and p-p collisions at RHIC. This is taken from [6].

observables by making measurements with high statistical precision, associated with the existence of a critical point.

The organisation of this thesis is as follows. In Chapter 2, the CBM detector system is described in brief. Then, in Chapter 3, event-by-event fluctuations are briefly discussed. In Chapter 4, important formulae for the analysis of cumulants of net-proton and net-charge distribution are stated and/or derived. In Chapter 5, the analysis techniques are discussed, and bin width corrected cumulants are presented for both net-charge and net-proton multiplicity distributions. Finally, the efficiency correction technique is discussed for cumulants of net-proton multiplicity distributions.

Chapter 2

The Compressed Baryonic Matter (CBM) Experiment

The Facility for Antiproton and Ion Research (FAIR) is a new accelerator complex currently under construction at the GSI Helmholtz Centre for Heavy Ion Research in Darmstadt, Germany. The Compressed Baryonic Matter experiment [7] is a heavy-ion fixed target experiment, which is being built in FAIR. Heavy nuclei beams will be provided to CBM using two synchrotrons, the SIS100 and the SIS300. The 100 and 300 signifies the bending power $(B.r)$ of the magnets used. SIS100 has a strength of 100 Tm, while SIS300 has a strength of 300 Tm. The available kinetic beam energy per nucleon depends essentially on the bending power $B.r$ provided by the dipole magnets.

$$E/A = \sqrt{(0.3 \times B.r \times Z/A)^2 + m^2} - m \qquad (2.1)$$

where Z and A are the charge and atomic number of the ion respectively, B is the magnetic field, r is the radius of the synchrotron, and m the mass of the nucleon. Table 2.1 shows the proposed kinetic energy per nucleon for different ion species in SIS100 and SIS300 setup.

The CBM experiment will investigate nucleus-nucleus collisions with energy ranging from 2 $AGeV$ to 45 $AGeV$ in the lab frame. It will complement the BES program of RHIC (where centre of mass energy ranges from 7.7 to 200 GeV). The experiment at CBM is fixed-target, designed to run at extremely high interaction rates of upto 10 MHz [9]. It will be capable of measuring both hadrons and leptons produced in

Table 2.1: Ion species and their kinetic energy per nucleon for a beam rigidity of 100 Tm at the SIS100 and 300 Tm at the SIS300. The table has been reproduced from [8].

Beam	Z	A	E/A GeV SIS100	E/A GeV SIS300
p	1	1	29	89
d	1	2	14	44
Ca	20	40	14	44
Ni	28	58	13.6	42
In	49	115	11.9	37
Au	**79**	**197**	**11**	**35**
U	92	238	10.7	34

the nuclear collisions. The CBM detector system will accept polar emission angles between 2.5 and 25 degrees and has full azimuthal coverage. The setup contains the following sub-detectors [8]:

1. Micro-Vertex Detector (MVD): The MVD consists of four layers of ultra-thin and highly-granulated monolithic silicon sensors which are located close to the target. The MVD is needed to reconstruct the primary collision vertex. It will also secondary vertices with high precision which are required for the reconstruction of open charm (D_0, D^* etc.). The angular coverage for MVD is 2.5 deg - 25 deg.

2. Silicon Tracking System (STS): The STS consists of up to 8 tracking layers of silicon detectors. They are located downstream of the target at distances between 30 and 100 cm in a magnetic dipole field of 1 Tm bending power. The task of the STS is to provide track reconstruction and momentum determination of charged particles. The required momentum resolution is of the order of $\Delta p/p = 1\%$. The angular coverage for STS is 2.5 deg - 25 deg.

3. Ring Imaging Cherenkov Detector (RICH): The RICH detector comprises a CO_2
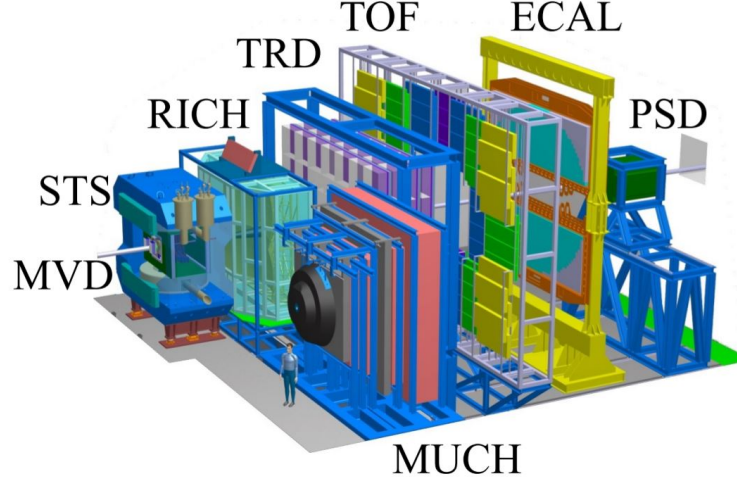
Figure 2.1: The CBM Detector Setup. The components of electron and muon setup are both shown in the diagram.

radiator and a UV photon detector realized with multi-anode photomultipliers for electron identification. It is designed to provide identification of electrons and suppression of pions in the momentum range below 8 GeV/c. The angular coverage for RICH is 2.5 deg - 25 deg.

4. Transition Radiation Detector (TRD): Three Transition Radiation Detector stations each consisting of 34 detector layers will serve for particle tracking and for the identification of electrons and positrons with $p > 1.5\ GeV/c$ ($\gamma > 1000$). Each layer consists of a radiator where the transition radiation is produced by electrons, and of a gaseous detector in which the deposited energy of charged particles and the transition radiation can be measured. The angular coverage for TRD is 2.7 deg - 44.7 deg.

5. Muon Chamber System (MuCh): Muons will be measured with an active hadron absorber system consisting of iron layers and muon tracking chambers (MuCh). The final design of the muon detector system consists of 6 hadron absorber layers

and 18 gaseous tracking chambers located in triplets behind each absorber slab. The angular coverage for MuCh is 5.7 deg - 25 deg.

6. Time of Flight Detector (TOF): Charged hadron identification will be performed by a time-of-flight (TOF) measurement with a wall of resistive plate chambers (RPC) located at a distance of 10 m behind the target. A micro-strip detector provides the start signal for the TOF measurement. The equations pertaining to the time of flight measurements are derived in Chapter 4. The angular coverage for TOF is 2.5 deg - 25 deg.

7. Electromagnetic Calorimeter (ECAL) and Projectile Spectator Detector (PSD): ECAL provides information on photons and neutral particles in selected regions of phase space. PSD is needed for the determination of the collision centrality and the orientation of the reaction plane.

   CBM will run in two different setups : *Electron setup* and *Muon setup.* The electron setup will comprise of MVD, STS, RICH, TRD, TOF, ECAL, and PSD detectors. In the muon setup, the RICH detector will be replaced by MuCh for muon detection. A summary of the required detectors for different particle measurements is presented in Table 2.2.

Table 2.2: Observables and required detectors. 'x' denotes that the detectors can detect the mentioned observables. The table has been taken from [8].

Observables	MVD	STS	RICH	MuCh	TRD	TOF	ECAL	PSD
π, K, p		x	x		x	x		x
Hyperons		x			x	x		x
Electrons	x	x	x		x	x		x
Muons		x		x		x		x
Photons							x	x
Photons via $e^{\pm}$ conversion	x	x	x		x	x		x

Chapter 3

Event by event fluctuations

In this thesis, the aim is to look at event-by-event fluctuations of conserved quantities like net baryon number, net charge, and net strangeness, in the CBM detector setup. Theoretically, it has been predicted that at high temperatures, there is a cross-over from hadronic phase to quark-gluon-plasma at $\mu_B = 0\ MeV$ [3]. At larger μ_B, a first order phase transition is predicted between quark-gluon and hadronic phases, which ends in a critical point (CP). Several other studies have reported that the CP region is unlikely to be found below $\mu_B = 200\ MeV$ [10].

It has also been observed in several experiments that the value of baryon chemical potential μ_B increases with decreasing $\sqrt{s}$. In fact, the following parameterisation of μ_B vs $\sqrt{s}$ has been reported [11].

$$\mu_B(\sqrt{s}) = \frac{1.308 \text{ GeV}}{1 + 0.273 \text{ GeV}^{-1} \times \sqrt{s}} \tag{3.1}$$

This is easy to understand, as with increasing s, the amount of entropy(heat) generated grows, while the net baryon number is limited by that of the initial nuclei [12]. The CBM experiment would look for the experimental signatures of a first order phase transition by colliding Au ions at various $\sqrt{s_{NN}}$. The variations of observables which are related to the moments of net baryon, net-charge, and net-strangeness number with respect to the beam energy are predicted to be good signatures of phase transitions and a critical point [13].

In Lattice QCD and HRG models, the susceptibilities are expressed in terms of the Taylor coefficients in the expansion of the dimensionless pressure (P/T^4), with

respect to normalised chemical potential (μ_i/T). [14]. Here i can represent baryon (B), charge (q), and strangeness (S).

$$P/T^4 \;=\; \frac{1}{VT^3}\ln[Z(V,T,\mu_i)] \tag{3.2}$$

$$\chi_i^n(T,\mu_i) \;=\; \left.\frac{\partial^n(P/T^4)}{\partial(\mu_i/T)^n}\right|_T \tag{3.3}$$

In a grand canonical ensemble, the nth order cumulant of multiplicity distributions are connected to the corresponding susceptibilities by

$$C_{n,i} = VT^3\chi_q^n(T,\mu_i) \tag{3.4}$$

where V denotes the volume of the system. The susceptibilities exhibit singularity near the critical point of the system.

While relating the susceptibilities to the moments, a volume term appears (Eq. 3.4.). This makes it difficult to compare different centralities, and different systems of collisions. However, the ratio of moments like $\frac{C_3}{C_2}$, $\frac{C_4}{C_2}$, are devoid of the volume term, and it has been seen from lattice QCD calculations that these products also change rapidly near the critical point. Some of the relations are listed below

$$S\sigma \;=\; \frac{C_3}{C_2} = \frac{\chi^{(3)}}{\chi^{(2)}} \tag{3.5}$$

$$\kappa\sigma^2 \;=\; \frac{C_4}{C_2} = \frac{\chi^{(4)}}{\chi^{(2)}} \tag{3.6}$$

$$\frac{\kappa\sigma}{S} \;=\; \frac{C_4}{C_3} = \frac{\chi^{(4)}}{\chi^{(3)}} \tag{3.7}$$

The moments have also been shown to be directly related to the correlation length (ξ) of the system. Some of the relations are listed below.

$$C_2 \sim V\xi^2 \,\big|\, C_3 \sim V\xi^{9/2} \,\big|\, C_4 \sim V\xi^7 \tag{3.8}$$

The most general dependence of moments on correlation lengths is given by $C_n \sim V\xi^{\frac{5}{2}n-3}$ [15]. Hence, the higher moments of distributions of conserved quantities are studied as they have a stronger dependence on ξ.

In an experiment however, it is often difficult to capture all baryons. For example, neutrons which are almost equally produced in heavy-ion collisions go completely undetected. It thus stands to question how one can practically measure variations of net baryon number. It has been reported that the proton number susceptibility almost completely reflects the singularity of the baryon number susceptibility near the critical point, and the effect of isospin fluctuations on the shape of net baryon distributions is small [16]. Hence the net proton multiplicity is a good proxy for the net baryon multiplicity.

In the next chapter, the relevant formulae for the calculations of cumulants of net-proton, and net-charge multiplicity distributions are discussed.

Chapter 4

Important formulae and Derivations

4.1 Kinematic Variables

There are certain kinematic variables which are often used in high energy physics. They are briefly discussed in this section.

1. **Center of mass energy** $(\sqrt{s})$: This is defined as $s = (\mathbf{p_1} + \mathbf{p_2})^2 = (\mathbf{p_3} + \mathbf{p_4})^2$ for a typical 2+2 scattering with particles 1 and 2 incoming, and particles 3 and 4 outgoing, where $\mathbf{p_i}$ denotes the 4-momentum, and m_i is the mass of the ith particle. It reduces to $s = m_1^2 + m_2^2 + 2E_1E_2 - 2\vec{p_1}.\vec{p_2}$, where $\vec{p_i}$ is the 3-momentum, E_i is the energy of the ith particle, and for a fixed target experiment like CBM, since $\vec{p_2} = 0$, it finally takes the form $s = m_1^2 + m_2^2 + 2E_1m_2$. $\sqrt{s}$ is typically a measure of the total 'energy' available in a system.

2. **Minkowski metric**: It is given by $g_{\mu\nu} = \text{diag}(1, -1, -1, -1)$.

3. **Transformation matrix**: For our purposes, the incoming beam is taken to be in z-direction, hence the matrix reduces to $\left(\begin{smallmatrix} \gamma & -\beta\gamma \\ -\beta\gamma & \gamma \end{smallmatrix}\right)$ for the t and z components of the 4-vector only.

4. **Rapidity**: It is defined as follows:

$$y = \frac{1}{2}\ln\frac{E + p_z}{E - p_z} = \frac{1}{2}\ln\frac{1 + \beta}{1 - \beta} \tag{4.1}$$

Here $\beta = \frac{p}{E}$.

Rapidity is additive under boost from one frame to another, therefore $y_{\text{new}} = y_{\text{old}} + \frac{1}{2} \ln \frac{1+\beta}{1-\beta} = y_{\text{old}} + y_{\text{boost}}$.

5. **Pseudorapidity**: It is defined as follows:

$$\eta = \frac{1}{2} \ln \frac{p + p_z}{p - p_z} \tag{4.2}$$

Using the definition $\cos\theta = \frac{p_z}{p}$, we have

$$\eta = -\ln(\tan\frac{\theta}{2}) \tag{4.3}$$

When $|\vec{p}| >> m$, $y \approx \eta$.

It is convenient to define rapidity for phenomenological calculations since the mass of the desired particle is known. However, in experiments, it is more convenient to use pseudorapidity as a typical detector or detector component covers some well-defined θ region with respect to the beam axis.

On a side note, for a beam of particles of mass m travelling with kinetic energy E_{kin} in the lab frame, $E_{lab} = E_{kin} + m$ and $p = \sqrt{E_{lab}^2 - m^2}$.

Then, $\beta_{CM} = \frac{p}{E_{lab}+m}$.

4.2 Cumulants

1. **Cumulant generating function**: For any random variable X, the CGF is defined as $G(t) = \log(< e^{tX} >)$.

2. **Cumulants**: The cumulants are defined by the derivatives of the CGF. The nth order cumulant C_n is given by

$$C_n = G^n(0) = \frac{\partial^n G(t)}{\partial t^n}\bigg|_{t=0} \tag{4.4}$$

3. **Raw Moments**: The nth order raw moment for a distribution N is defined as $m_n = <N^n>$ where $<>$ denotes the average.

4. **Central Moments**: The nth order central moment for a distribution N is defined as follows:

$$\mu_n = \langle (N - <N>)^n \rangle \tag{4.5}$$

Central moments are related to the nth order cumulants (C_n) as follows:

$$C_n = \mu_n - \sum_{m=2}^{n-2} \binom{n-1}{m-1} C_m \mu_{n-m} \tag{4.6}$$

For our analysis, we used the relations between raw moments and cumulants for ease of calculation. The cumulants upto 4th order are listed below in terms of raw moments.

$$
\begin{aligned}
C_1 &= m_1 \\
C_2 &= m_2 - m_1^2 \\
C_3 &= m_3 - 3m_2 m_1 + 2m_1^3 \\
C_4 &= m_4 - 4m_3 m_1 - 3m_2^2 + 12m_2 m_1^2 - 6m_1^4
\end{aligned}
\tag{4.7}
$$

5. **Variance**: The second central moment (σ^2) is used to describe the width of a distribution.

6. **Skewness**: Skewness, the normalised third central moment, denoted by S, is a measure of the asymmetry of the distribution. It is defined by $S = \frac{C_3}{C_2^{3/2}}$. A distribution can be negative or left skewed if the left tail is longer, and positive or right skewed if the right tail is longer.

7. **Kurtosis**: Kurtosis, denoted by κ, is a measure of the peakness of the distri-bution. It is defined by $\kappa = \frac{C_4}{C_2^2}$. A normal distribution has $\kappa = 0$. For any
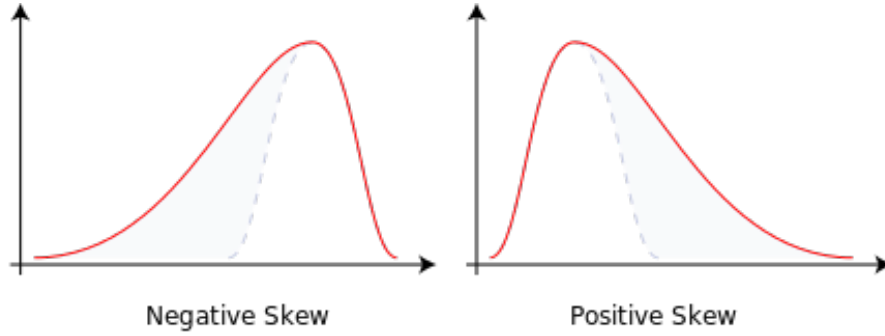
Figure 4.1: Left-skewed and Right-skewed distributions respectively. The image is shared under *Creative Commons License*, and hence is reproduced here.

arbitrary distribution, if $\kappa < 0$, it means the distribution produces fewer and less extreme outliers than does the normal distribution. Similarly, if $\kappa > 0$, the distribution produces more outliers than does the normal distribution. Example of the first kind is a *uniform distribution*, while example of the second kind is a *laplace distribution*.

8. **Properties of Cumulants**:

   8.1. *Translationally invariant*: $C_n(X + c) = C_n(X)$ where X is any random variable and c is any real number.

   8.2. *Homegenous*: $C_n(cX) = c^n C_n(X)$ where X is any random variable and c is any real number.

   8.3. *Additivity*: For independent random variables X and Y,

   $$C_n(X + Y) = C_n(X) + C_n(Y) \tag{4.8}$$

9. **Errors on Cumulants**: There are several methods to find out statistical errors in a sample and its related quantities. The most commonly used methods in such analysis are *Bootstrap*, *Delta Theorem*, and *Subgroup Method*. In this analysis, we primarily use "*Delta Theorem*" which provides us with an analytical function
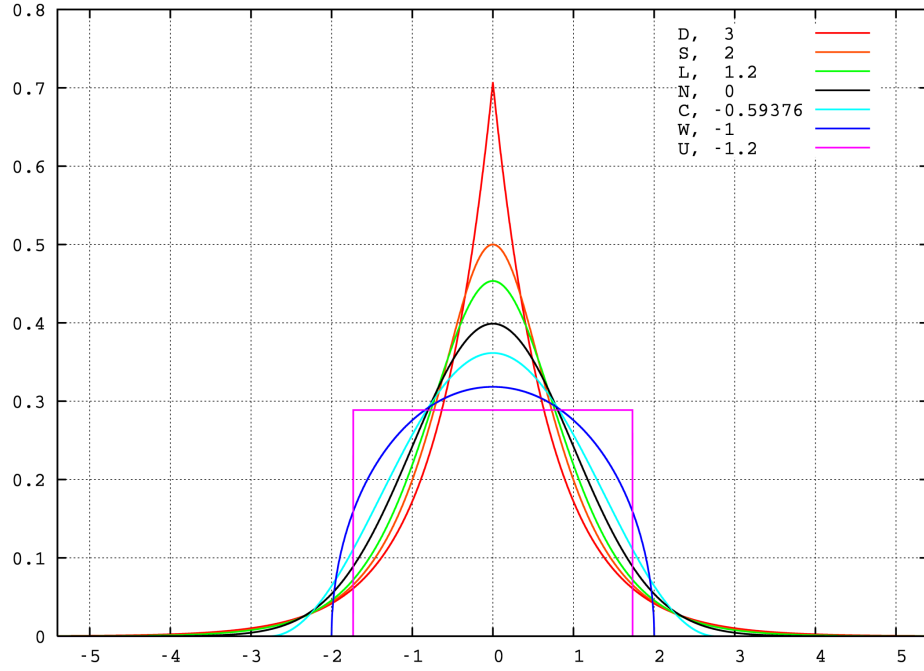
Figure 4.2: Value of Kurtosis for some standard distributions like Laplace Double Exponential, Hyperbolic Secant, Logistic, Normal, Raised Cosine, Wigner Semicircle, and Uniform. The image is shared under *Creative Commons License*, and hence is reproduced here.

to calculate the error.

Statement of Delta Theorem

Given a sequence of random variables X_n such that $\sqrt{n}[X_n - \theta] \rightarrow N(0, \sigma^2)$ where θ, and σ are finite, and $N(0, \sigma^2)$ represents the standard normal distribution, $\sqrt{n}[g(X_n) - g(\theta)] \rightarrow N(0, \sigma^2[g'(\theta)]^2)$, where g is any function for which $g'(\theta)$ exists and is non-zero.

For our case, we can take X_n to be a constant sequence X, and $\theta = \mu$. Then, the statement simplifies to

$$X \sim N(\mu, \frac{\sigma^2}{n}) \implies g(X) \sim N(g(\mu), \frac{\sigma^2}{n}[g'(\mu)]^2) \tag{4.9}$$

To find out the errors, we need to find the variances and covariances of the

central moments in the sample $(hat\mu_i)$. The general formulae are given in terms of the population central moments (μ_i) below.

$$\text{var}(\hat{\mu}_r) = \frac{1}{n}(\mu_{2r} - \mu_r^2 + r^2\mu_2\mu_{r-1}^2 - 2r\mu_{r-1}\mu_{r+1}) \qquad (4.10)$$

$$\text{cov}(\hat{\mu}_r, \hat{\mu}_q) = \frac{1}{n}(\mu_{r+q} - \mu_r\mu_q + rq\mu_2\mu_{r-1}\mu_{q+1} - q\mu_{r+1}\mu_{q-1}) \qquad (4.11)$$

From Equation 4.10, we have the distribution of sample moment $\hat{\mu}_2$ given by

$$\hat{\mu}_2 = \hat{\sigma}^2 \sim N\left(\sigma^2, \frac{\mu_4 - \sigma^4}{n}\right) \qquad (4.12)$$

Now, consider the function $g(x) = \sqrt{x}$. Using Delta Theorem on $\hat{\sigma}^2$, we have

$$\hat{\sigma} \sim N\left(\sigma, \frac{\mu_4 - \sigma^4}{4\sigma^2 n}\right) \qquad (4.13)$$

This gives the variance in μ_2.

Errors in $C_1, C_2, \& C_3$ are trivial and are given by the variances of $\mu_1, \mu_2, \& \mu_3$ respectively. The error in C_4 is slightly more involved and is discussed below for completeness.

$$C_4 = \mu_4 - 3\mu_2^2$$

$$dC_4 = d\mu_4 - 6\mu_2 d\mu_2$$

$$\text{var}(C_4) = \text{var}(\mu_4) - 12\mu_2\text{cov}(\mu_4, \mu_2) + 36\mu_2^2\text{var}(\mu_2)$$

$$\implies \text{var}(C_4) = \frac{1}{n}\mu_8 - 12\mu_6\mu_2 - 8\mu_5\mu_3 - \mu_4^2 + 48\mu_4\mu_2^2 + 64\mu_3^2\mu_2 - 36\mu_2^4 \qquad (4.14)$$

Chapter 5

Analysis and Preliminary Results

5.1 Simulation

We have performed simulation for net-proton and net-charge fluctuations in the CBM framework. The tools used for the simulation are CBMROOT, FairSoft, and Fair-Root. The Monte-Carlo generator used for generating the events is UltraRelativistic Quantum Molecular Dynamics (UrQMD) [17, 18].

UrQMD

UrQMD is an acronym for "Ultra Relativistic Quantum Molecular Dynamics". It is a transport model for simulating heavy ion collisions in the low to medium energy range, like those found in SIS and RHIC. A microscopic dynamics description of heavy ion collisions is usually based on transport theory. Here a sequence of propagations of particles is simulated numerically, and the inputs include cross sections, the two-body potentials, and decay widths. UrQMD runs on various UNIX-based computing platforms, and is designed as a multipurpose tool for studying a wide variety of heavy ion related effects including fluctuations of conserved quantities [17, 18]. For our analysis, **UrQMD v3.3** has been used. UrQMD produces the data in ASCII format, which is then converted to a "*.root*" file using the sofware UNIGEN (United Generators). This file serves as the input to CBM detector setup in GEANT. The simulation chain is as follows.

The UrQMD data file is passed through the CBM detector setup in GEANT, using the routine '*run_mc.C*, and a root file containing the detector response, generated

using monte-carlo methods is the output. The output file, which we name as '*mc.root*' is used as an input to the routine '*run_reco.C*' for reconstructing the tracks. The output file after reconstruction, named as '*reco.root*' is finally used for the analysis. Fig 5.2 shows a typical event display simulated for $Au - Au$ collision at $E_{lab} = 10$ $AGeV$.
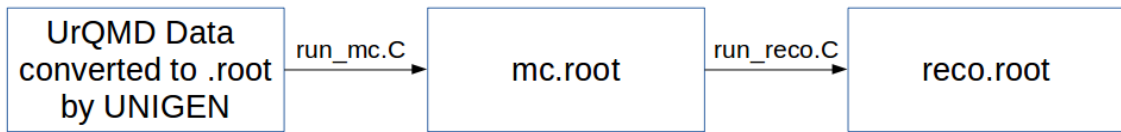
Figure 5.1: Block diagram of the simulation chain for the analysis.

5.2   Analysis Tools

The analysis has been done using CBMROOT and some associated softwares which are listed below. The CBM electron setup for SIS100 energies has been used as the detector setup for this analysis. The SIS100 electron setup consists of MVD, STS, RICH, TRD, and TOF detectors. Nearly 3 million events of Au-Au collisions at $E_{lab} = 10$ $AGeV$ have been used for the analysis. The entire simulation and analysis was performed on the HPC cluster located in GSI, Darmstadt. Some important details about the analysis tools are listed below:

1. **Event Generator** : *UrQMD 3.3*

2. **Versions of CBMROOT & associated software**

   2.1. CBMROOT - **MAY2018**

   2.2. FairSoft - **Mar17**

   2.3. FairRoot - **v-17.03a**
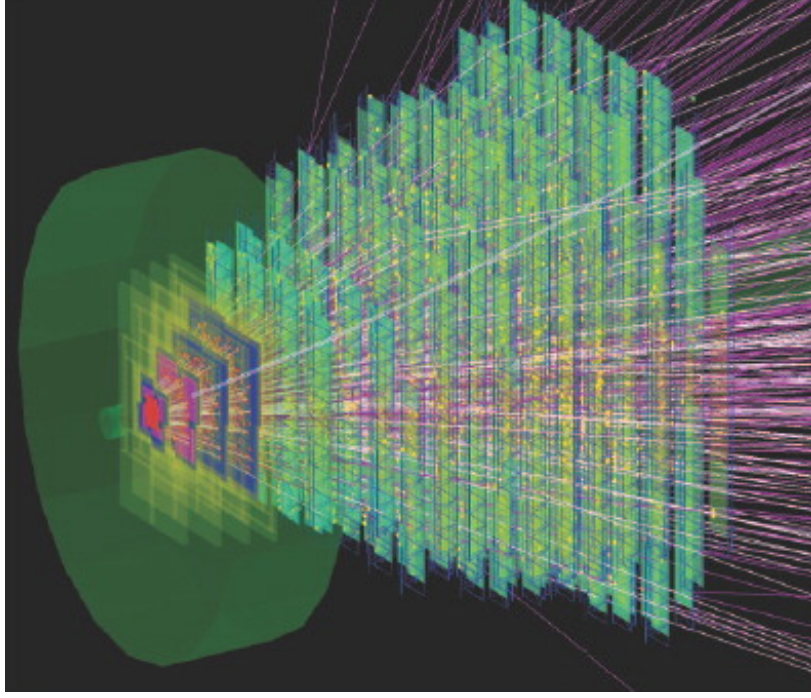
Figure 5.2: Simulation of a central collision of two gold nuclei at a beam energy of 10A GeV. This figure has been reproduced from [19].

3. **Detector Setup**: ***sis100_electron***.

4. **System**: **Au-Au** collisions ($Z_{Au} = 79$; $A_{Au} = 197$)

5. **Impact Parameter**: $b \sim 0 - 14 fm$

6. **Energy**: $E_{lab} = 10$ AGeV $\implies \sqrt{s_{NN}} = 4.72$ GeV.

7. **Number of events**: ~ 3 million minimum bias

5.3   Analysis Cuts

To select the charged particle tracks with better reconstruction, and to avoid secondaries, various cuts are applied.

1. **Hits in STS**: The STS detector has 8 consecutive panels, located along the

beam axis. To ensure that the momentum of the track is correctly estimated, we have chosen tracks which have more than 6 hits in the STS detector system. This is shown in Fig 5.3.
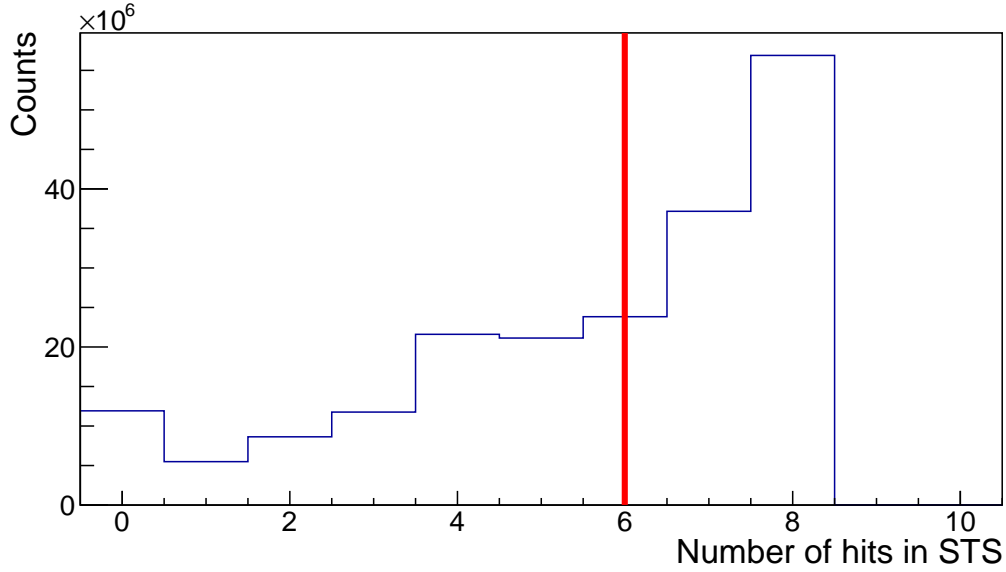
Figure 5.3: The distribution of number of hits in STS detector. Tracks having more than 6 hits out of 8 in the detector have been selected for analysis.

2. **Hits in MVD**: The MVD detector has 4 consecutive panels, located along the beam axis. Along with STS, MVD helps in reconstructing the track vertex. We have chosen tracks which have a minimum of 2 hits in the MVD detector system. This is shown in Fig. 5.4.

3. **Distance of Closest Approach (DCA)**: The position of every collision (in cartesian coordinates) is recorded by MVD and STS. After reconstruction, every track is also assigned a position, which denotes the point where it originated. All primary tracks in an event should essentially originate from the event vertex, however there is a spread in the vertices of the tracks in an event. Distance of closest approach, or DCA refers to the absolute scalar difference between
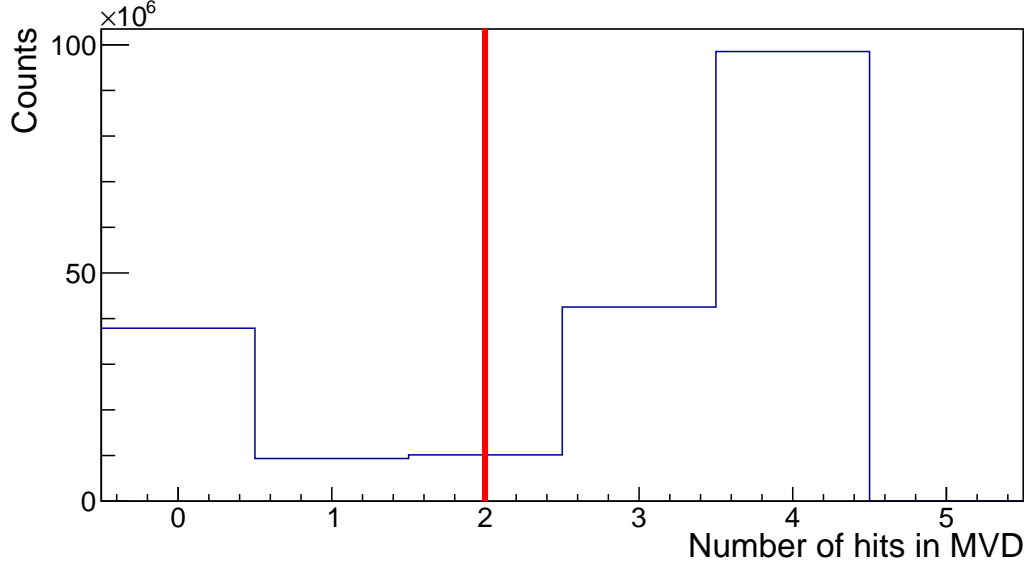
Figure 5.4: The distribution of number of hits in MVD detector. Tracks having more than 2 hits out of 4 in the detector have been selected for analysis.

the vertices of the tracks and the vertex of the event they belong to. For our analysis, we have selected tracks for which the z-component of the DCA (dca_z) is less than 0.2 *cm*, and the transverse component ($dca_t = \sqrt{dca_x^2 + dca_y^2}$) is less than 1 *cm*. as shown in Fig. 5.5. This helps us in getting rid of secondary tracks.

4. **Pseudorapidity**: The CBM detector setup has a polar angle coverage of 2.5 deg to 25 deg. Using Equation 4.3, we thus have set a limit of $\eta \simeq 1.5$ corresponding to $\theta = 25$ deg and $\eta \simeq 3.8$ corresponding to $\theta = 2.5$ deg, as shown in Fig. 5.6.

5. **Hits in TOF**: The Time-of-Flight detector is further downstream as compared to the STS and MVD. Since we have to identify protons and antiprotons in this analysis, we have only chosen tracks that have at least one hit in TOF.
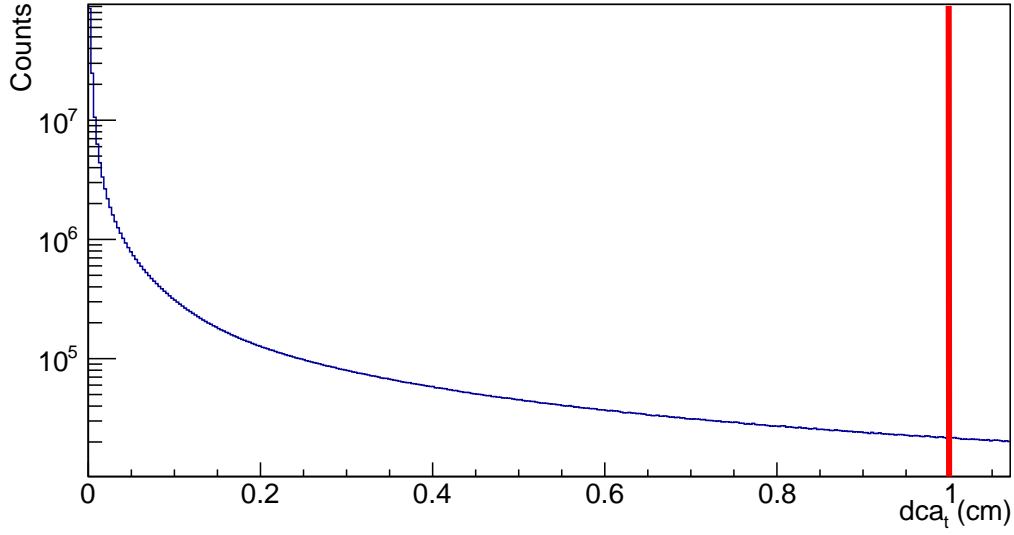
Figure 5.5: The distance of closest approach in transverse plane (dca_t). A dca_t cut is applied at 1 cm ($dca_t < 1$ cm).
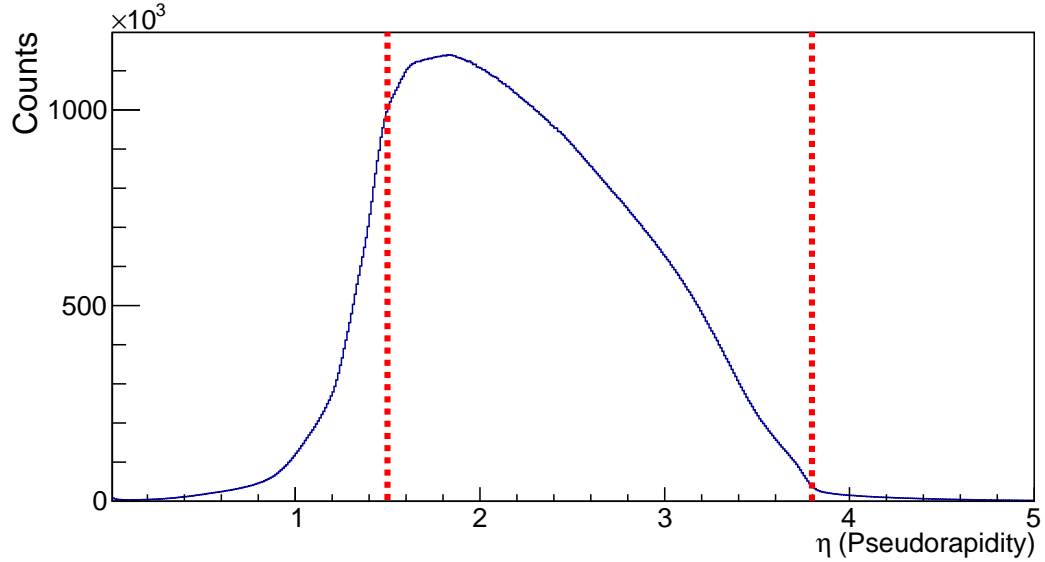
Figure 5.6: The pseudorapidity distribution of all charged particles. The particles in the region $1.5 < \eta < 3.8$ have been chosen for analysis, based on the coverage of the detector.

22

5.4 Net-Proton Fluctuation

The data from the following detectors have been used for this analysis.

1. **Micro Vertex Detector** for primary collision vertex, and track reconstruction.

2. **Silicon Tracking System** for track reconstruction, and momentum determination.

3. **Time-of-Flight detector** for particle identification

5.4.1 Particle Identification

Particle identification in CBM is done using time-of-flight measurements available from the TOF detector. TOF records the length(l) a track has covered before it is detected in TOF, and the time taken (t_{flight}) to reach the TOF detector from the micro-strip detector which measures the start time near the vertex.

$$\beta = \frac{l}{t_{flight}} = \frac{p}{E} \tag{5.1}$$

$$\beta = \frac{p}{\sqrt{p^2 + m^2}}$$

$$\therefore m^2 = p^2 \left(1 - \frac{1}{\beta^2}\right) \tag{5.2}$$

Equations 5.1 and 5.2, in conjunction, give the m^2 which we use to identify the particle species. Fig. 5.7 shows $1/\beta$ vs momentum multiplied by charge ($q * p$) for charged hadrons in Au-Au collision at $\sqrt{s_{NN}} = 4.72$ GeV. The bands correspond to π, K and protons (from bottom) respectively, and are well separated for $p < 3$ GeV/c. In top panel of Fig. 5.8, m^2 is plotted as a function of momentum multiplied by charge ($q*p$) and in the bottom panel, 1D projection of m^2 is shown in the analysed momentum range of $0.2 < p_T < 2$ GeV/c. The protons and antiprotons are selected using m^2 cut of $0.6 < m^2 < 1.2$ GeV^2/c^4, in the rapidity range $1.1 < y < 2.0$ and momentum range

$0.2 < p_T < 2.0$ GeV/c. The $y - p_T$ acceptance of proton and antiproton is shown in Fig 5.9, and the box in the plot shows the region of analysis.
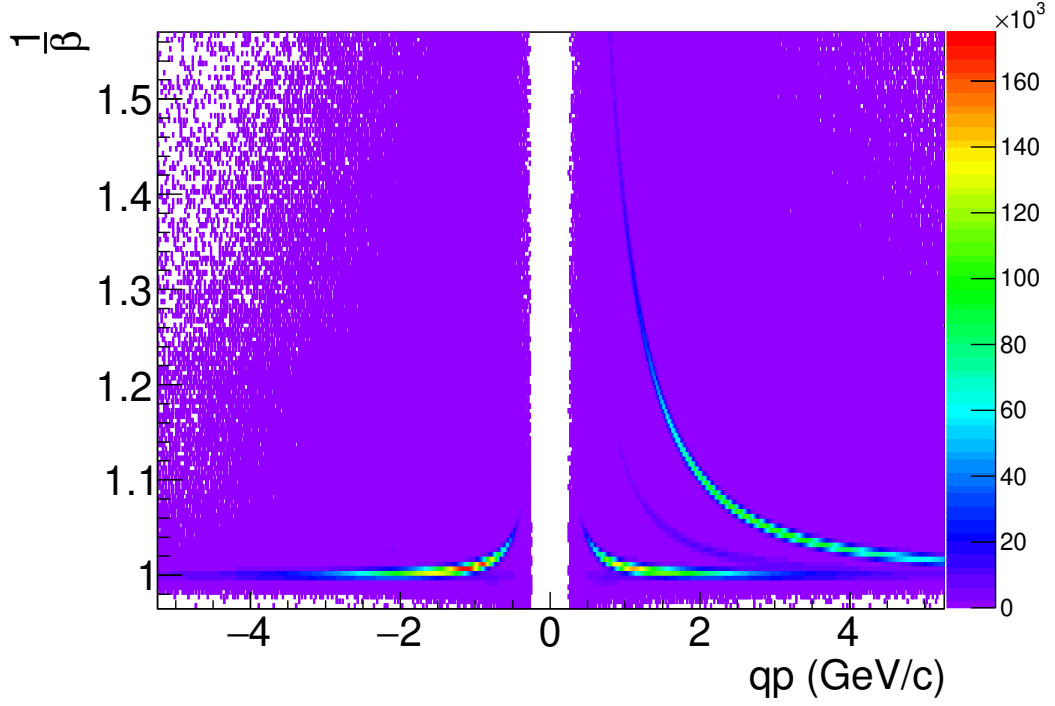
Figure 5.7: $\frac{1}{\beta}$ vs momentum multiplied by charge. The three bands on the positive axis (from bottom) belong to π^+, K^+, and protons respectively. The two bands on the negative axis (from bottom) belong to π^- and K^- respectively. There are very few antiprotons produced at $E_{lab} = 10$ AGeV, as is clearly seen from the absence of a band opposite to that of protons.
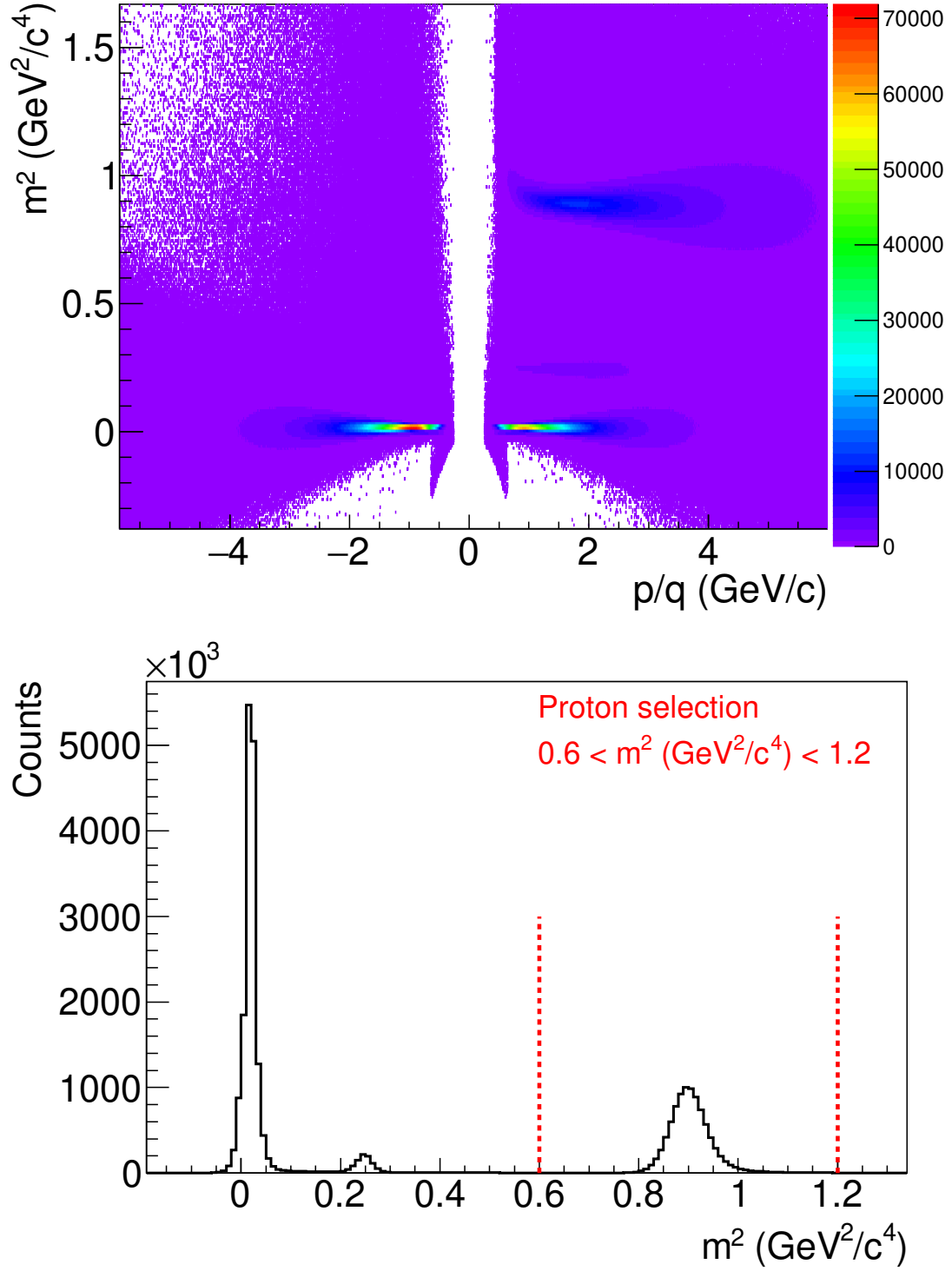
Figure 5.8: (Top Panel) Mass-squared vs Momentum/Charge. Protons(Antiprotons) are chosen using the cut $0.6 < m^2 < 1.2$ GeV^2/c^4. (Bottom Panel) 1D projection of m^2 in momentum range $0.2 < p_T < 2$ GeV/c.
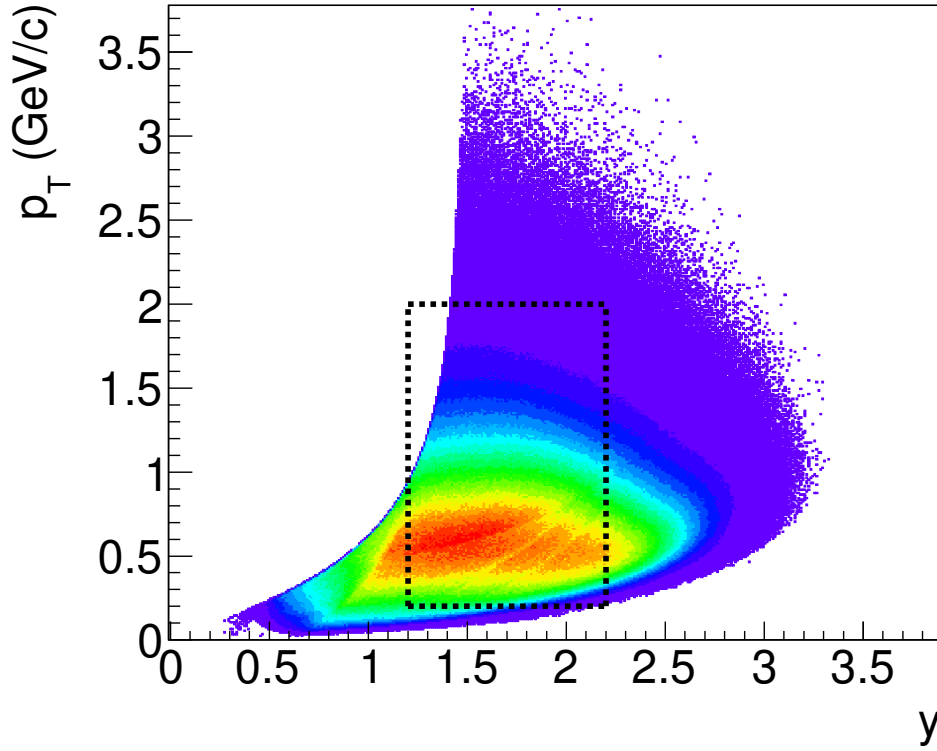
Figure 5.9: Transverse momentum (p_T) vs Rapidity(y) for (anti)protons. In this analysis, we have chosen (anti)protons in the region $1.1 < y < 2$ in the rapidity range and $0.2 < p_T < 2.0$ GeV/c in the momentum range, shown by the rectangular region.

5.4.2   Centrality Selection

Nuclei are extended objects, and hence the volume of the interacting region depends on the impact parameter (b) of the collision. It is defined as the distance between the centres of the two colliding nuclei. For two nuclei of radii r_1 and r_2, the impact parameter can range from 0 to $r_1 + r_2$. For Au-Au collision, the impact parameter ranges from $0 - 14 \ fm$. However, in a real experiment, the impact parameter is not an accesible parameter, hence a new quantity, called centrality is defined based on the impact parameter, and the cross-section of the interaction. The centrality is usually defined as a percentage of the total nuclear interaction cross section σ.

The centrality percentile for a $A - A$ collision with an impact parameter b is defined as [20, 27]

$$c(b) = \frac{\int_0^b d\sigma/db' db'}{\int_0^\infty d\sigma/db' db'} \tag{5.3}$$

Since number of particles produced in a collision is directly related to the cross-section, in experiments, centrality classes are defined by the same. The centrality selection is done by the multiplicity of charged particles which is referred as Reference Multiplicity (RefMult). To avoid the autocorrelation, RefMult is measured using charged tracks with $m^2 < 0.4 \ \text{GeV}^2/\text{c}^4$. Fig 5.10 shows the RefMult distribution for the net-proton analysis. The analysis is performed in 9 centrality classes : 0 - 5 %, 5 - 10 %, 10 - 20 %, 20 - 30 %, 30 - 40 %, 40 - 50 %, 50 - 60 %, 60 - 70 %, and 70 - 80 %. The number of events in each centrality class are listed in Table 5.1.
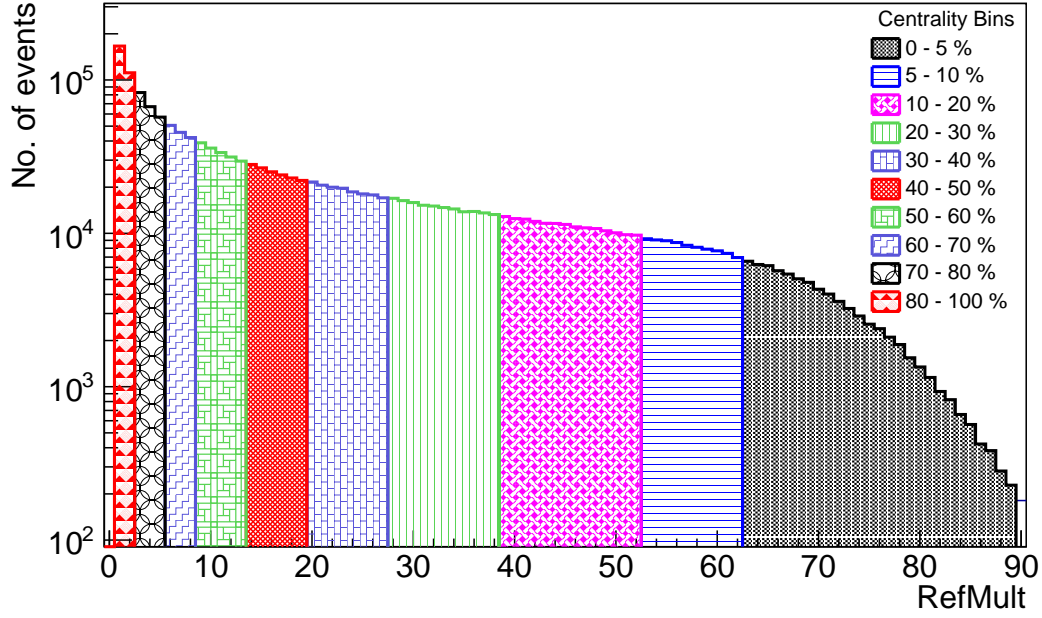
Figure 5.10: RefMult distribution in Net-proton analysis. The different coloured bands correspond to the different centrality classes. $0 - 5\%$ represents the most central collisions, while $70 - 80\%$ represents the most peripheral collisions used in our analysis.

Table 5.1: Number of events in each centrality class in net-proton analysis in Au-Au collision at $E_{lab} = 10$ AGeV.

Centrality (%)	No. of Events
0 - 5	76.1k
5- 10	82.3k
10 - 20	157.0k
20 - 30	163.1k
30 - 40	153.5k
40 - 50	149.2k
50 -60	169.5k
60 -70	138.3k
70 - 80	208.2k

5.4.3 Cumulants

Once the centrality classes are determined, the proton and antiproton distributions for each centrality class are measured. The multiplicity distribution of protons and antiprotons are shown in Fig. 5.11. The net-proton ($\Delta N_p = N_p - N_{\bar{p}}$) multiplicity distributions in various centrality classes are shown in Fig. 5.12. The cumulants and central moments upto 4th order are calculated for net-proton multiplicity distributions using Eqn. 4.7. The first cumulant C_1 of net-proton distribution decreases as we go from central to peripheral.This can be understood by the decrease of particle production in peripheral collisions as compared to central collisions. Various corrections are applied on the measured cumulants, and they are discussed in the next section.

5.4.4 Corrections to Cumulants

Centrality Bin Width Effect

Particle production is dependent on the energy of collision, as well as the centrality. Choosing the smallest centrality windows become important so that we can reduce the fluctuations arising due to the selection. The inherent fluctuations are those arising from the difference in geometry within the centrality bin. A centrality bin scans a range of charged particle multiplicity with different cross sections. This introduces geometrical fluctuations which need to be controlled. Selection of narrow centrality bins helps to get rid of inherent fluctuations within a centrality bin. In our analysis, the smallest accesible centrality windows are the multiplicity bins, each one unit wide. Bin width correction is done by taking the weighted average of the observables [22] in multiplicity bins.

$$X = \frac{\sum_i n_i X_i}{\sum_i n_i} = \sum_i \omega_i X_i \qquad (5.4)$$
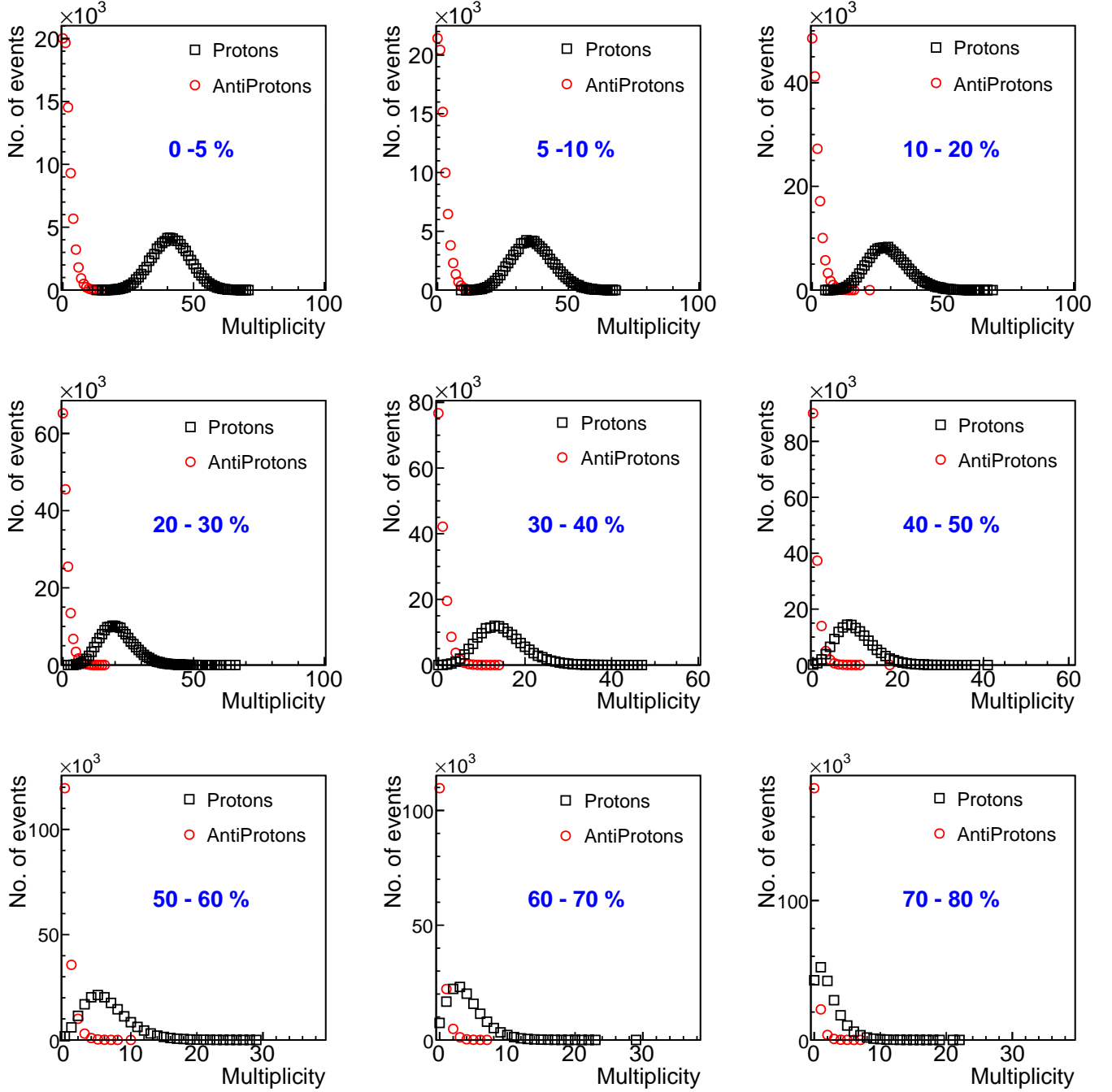
Figure 5.11: Multiplicity distribution of protons and antiprotons in different centrality classes in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c and $E_{lab} = 10$ AGeV.
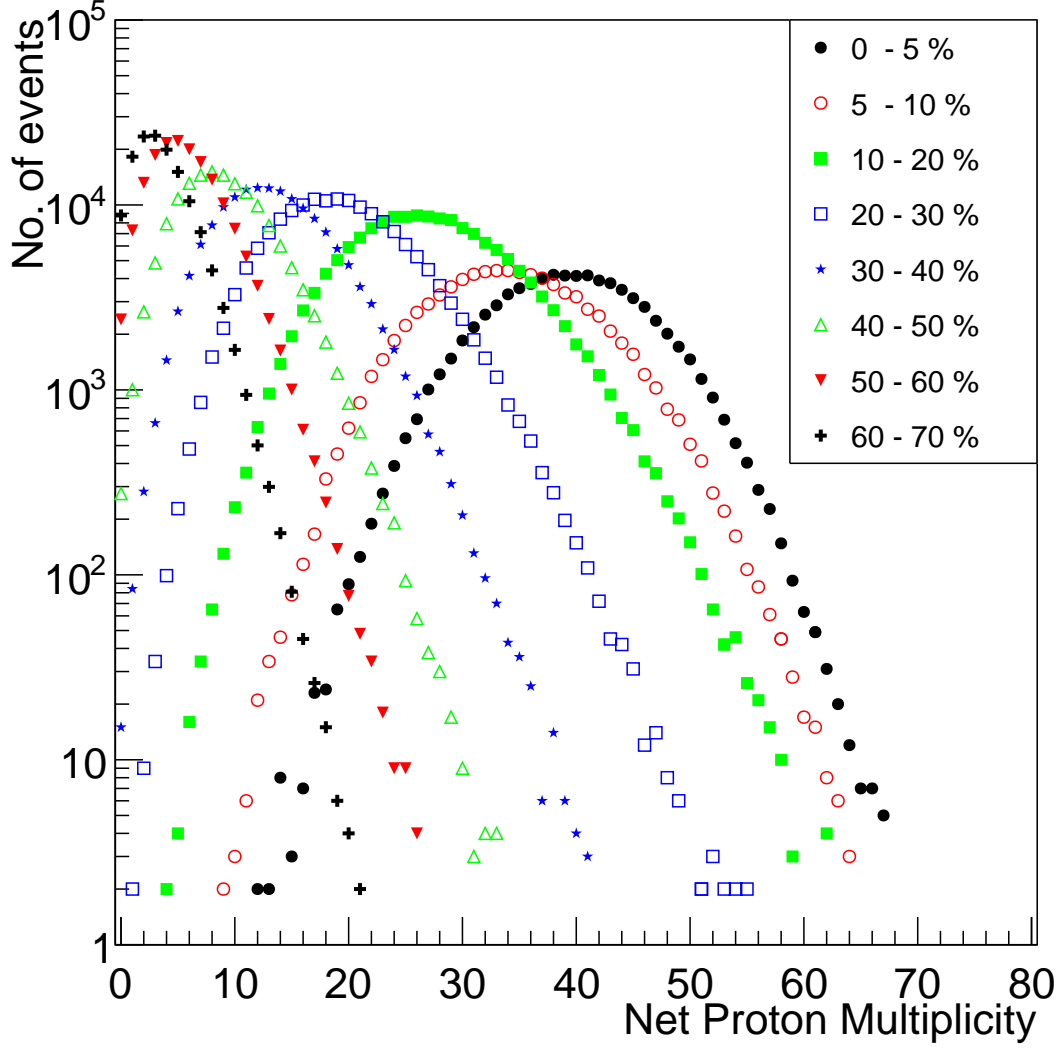
Figure 5.12: Net proton multiplicity distributions for different centrality classes in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c and $E_{lab} = 10$ AGeV.

where the index i runs over each multiplicity bin, X_i represents the moments or the cumulants in the ith multiplicity bin, n_i represents the number of particles in the ith multiplicity bin, and $\sum_i \omega_i = 1$. The cumulants before and after bin width correction are shown in Fig. 5.13. The statistical errors for the cumulants have been calculated using *Delta Theorem* which has been discussed in Section 4.2.
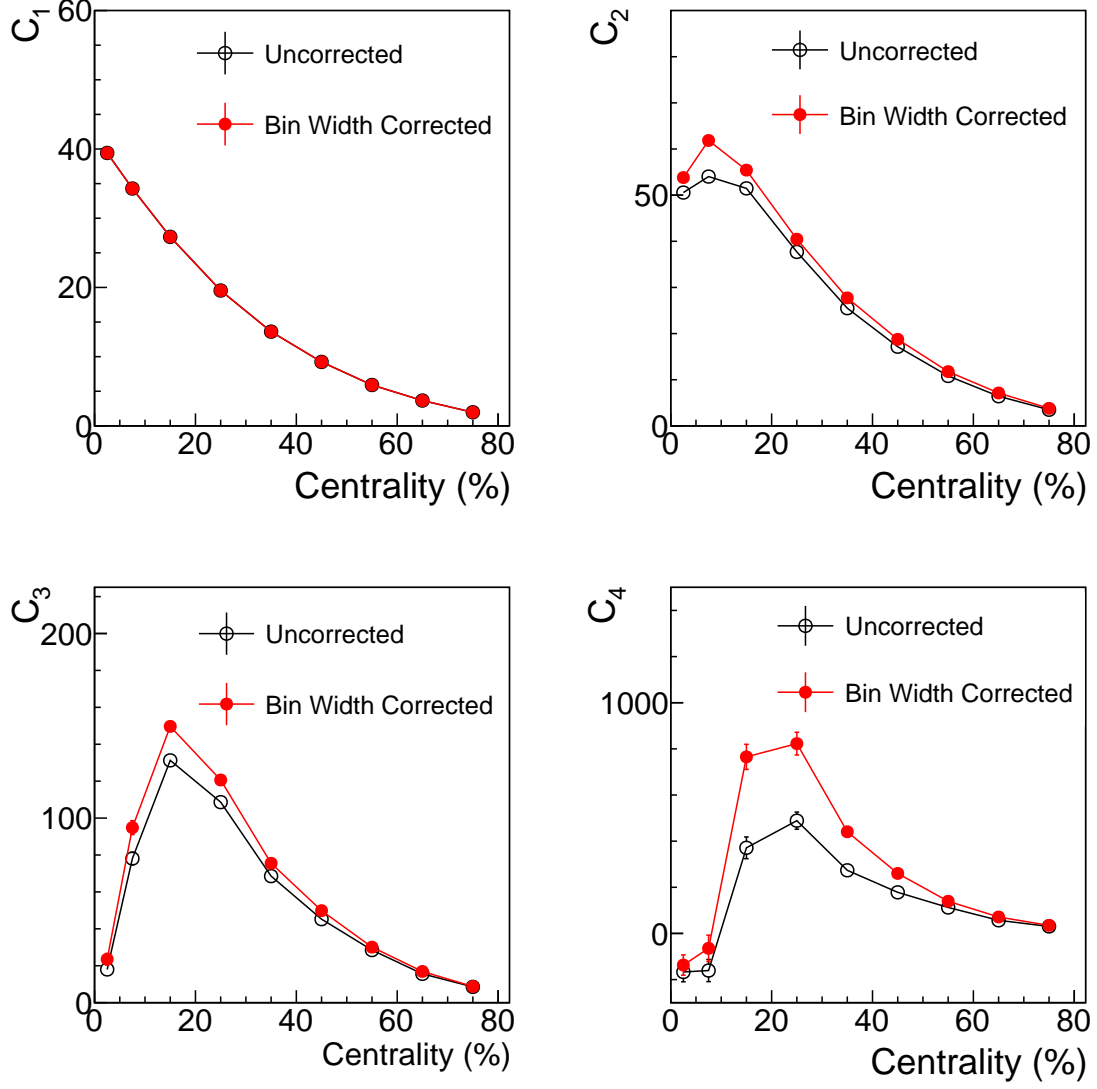
Figure 5.13: The first four cumulants of net-proton multiplicity distributions before and after bin width correction in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c, and $E_{lab} = 10$ AGeV.

Efficiency Correction

In real detectors, we don't have 100% efficiency of detection, which implies we lose out on the information from the actual collision. Thus, we need to estimate the effect of detector efficiency on the observable. Generally, the effect of efficiency (ϵ) can be

modelled by a binomial distribution on the actual number of particles in an event.

Let the actual number of particles of a certain species produced event-by-event, N, have a distribution $\tilde{N}$. Let the efficiency of the detector be ϵ, and let the detector acceptance be modelled by a binomial distribution denoted by $B(N, \epsilon)$. The distribution of the number of particles of that species which are detected, n is then given by

$$P(n) = B(N, \epsilon)\tilde{N} \tag{5.5}$$

The Moment-Generating function is then given by

$$M(t) = (1 + (e^t - 1)\epsilon)^N \tag{5.6}$$

The moments can thus be calculated by evaluating the derivatives of $M(t)$ at $t = 0$. By inverting the equations for the moments of the distribution of the detected particles, we can thus get the moments of the distribution of the actual particles. The first four moments are listed below:

$$
\begin{aligned}
\langle N \rangle &= \frac{\langle n \rangle}{\epsilon} \\
\langle N^2 \rangle &= \frac{\langle n^2 \rangle - \langle n \rangle(1 - \epsilon)}{\epsilon^2} \\
\langle N^3 \rangle &= \frac{\langle n^3 \rangle - 3\langle n^2 \rangle(1 - \epsilon) + \langle n \rangle(1 - \epsilon)(2 - \epsilon)}{\epsilon^3} \\
\langle N^4 \rangle &= \frac{\langle n^4 \rangle - 6\langle n^3 \rangle(1 - \epsilon) + \langle n^2 \rangle(1 - \epsilon)(11 - 7\epsilon) - \langle n \rangle(1 - \epsilon)(6 - (6 - \epsilon)\epsilon)}{\epsilon^4}
\end{aligned}
\tag{5.7}
$$

where $< n^i >$ is the ith order raw moment for the distribution of detected particles, and $< N^i >$ is the ith order raw moment for the distribution of incident particles. The new cumulants defined using these moments, should in principle give us the actual cumulants of (anti) proton multiplicity distributions. From Eqn. 4.8, we know that cumulants of independent distributions add up. Therefore, using the above formalism on both protons and antiprotons, we get the individual cumulants which are then subtracted to get the cumulants of net-proton.
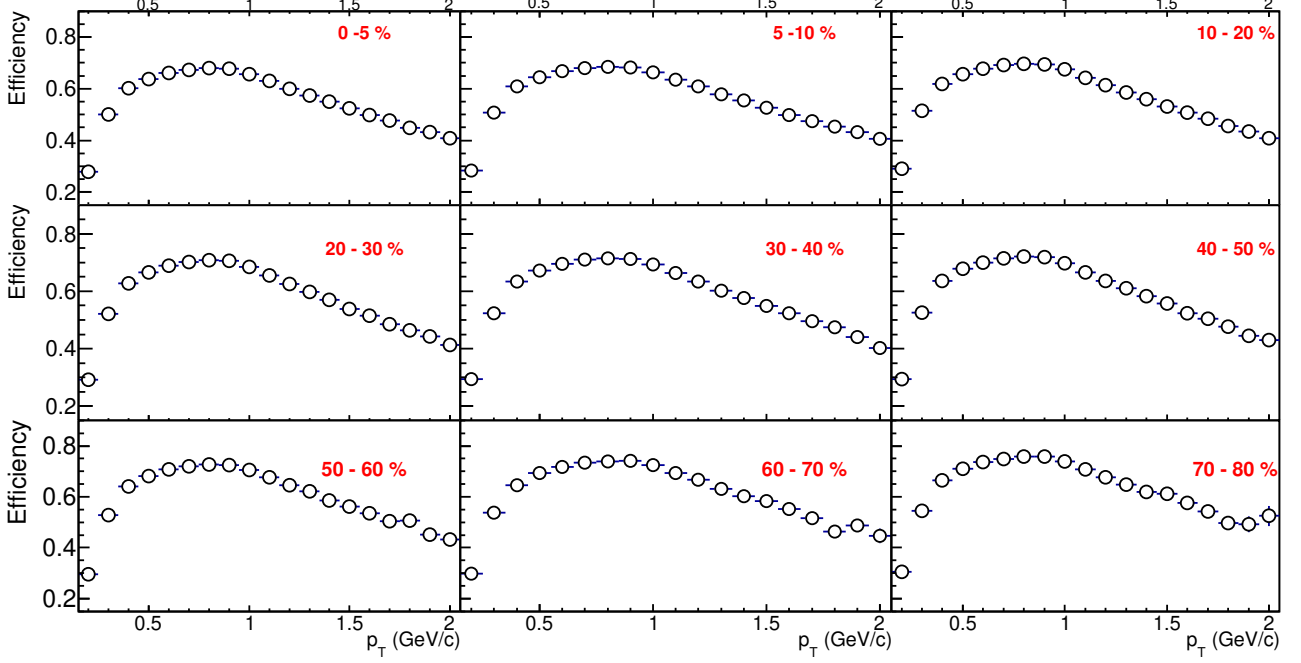
Figure 5.14: Efficiency vs transverse momentum (p_T) in different centrality bins in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c and $E_{lab} = 10$ AGeV.

For our analysis, we have defined efficiency in the following manner.

$$\text{Efficiency} = \frac{\text{Number of detected (anti)protons which are true (anti)protons}}{\text{Number of Incident (anti)protons}} \quad (5.8)$$

This definition also takes into account the purity of reconstruction of (anti)protons, however in our analysis, the purity is found to be over 90% in most of the centrality classes.

Fig 5.14 shows the efficiency as a function of the transverse momentum in the range $0.2 < p_T < 2$ GeV/c, for different centrality classes. Due to the low statistics, the efficiency of protons is used for the correction. Fig. 5.15 shows the p_T integrated values of efficiency as a function of centrality. The efficiency increases from central to peripheral collisions, which can be understood by the fact that in highly dense environments, such as in central collisions, the detector tends to miss out on detecting

a lot more particles than in peripheral collisions. The efficiency correction is done in each multiplicity bin, before it is bin width corrected. The efficiency corrected cumulants are shown in Fig. 5.16 and have been compared with the cumulants of the incident net-proton distribution in Fig. 5.17. There are visible disagreement between the incident and measured net proton cumulants, which suggests that the method is not complete or has not been applied properly in this analysis. It warrants further attention.
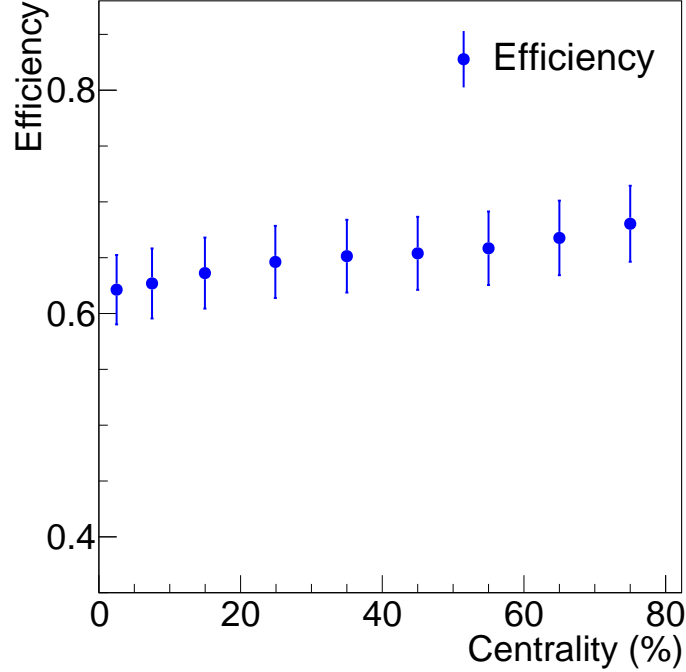
Figure 5.15: p_T integrated efficiency as a function of centrality in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c and $E_{lab} = 10$ AGeV. The errors have been fixed at 5% for this analysis, following the procedure in [6].
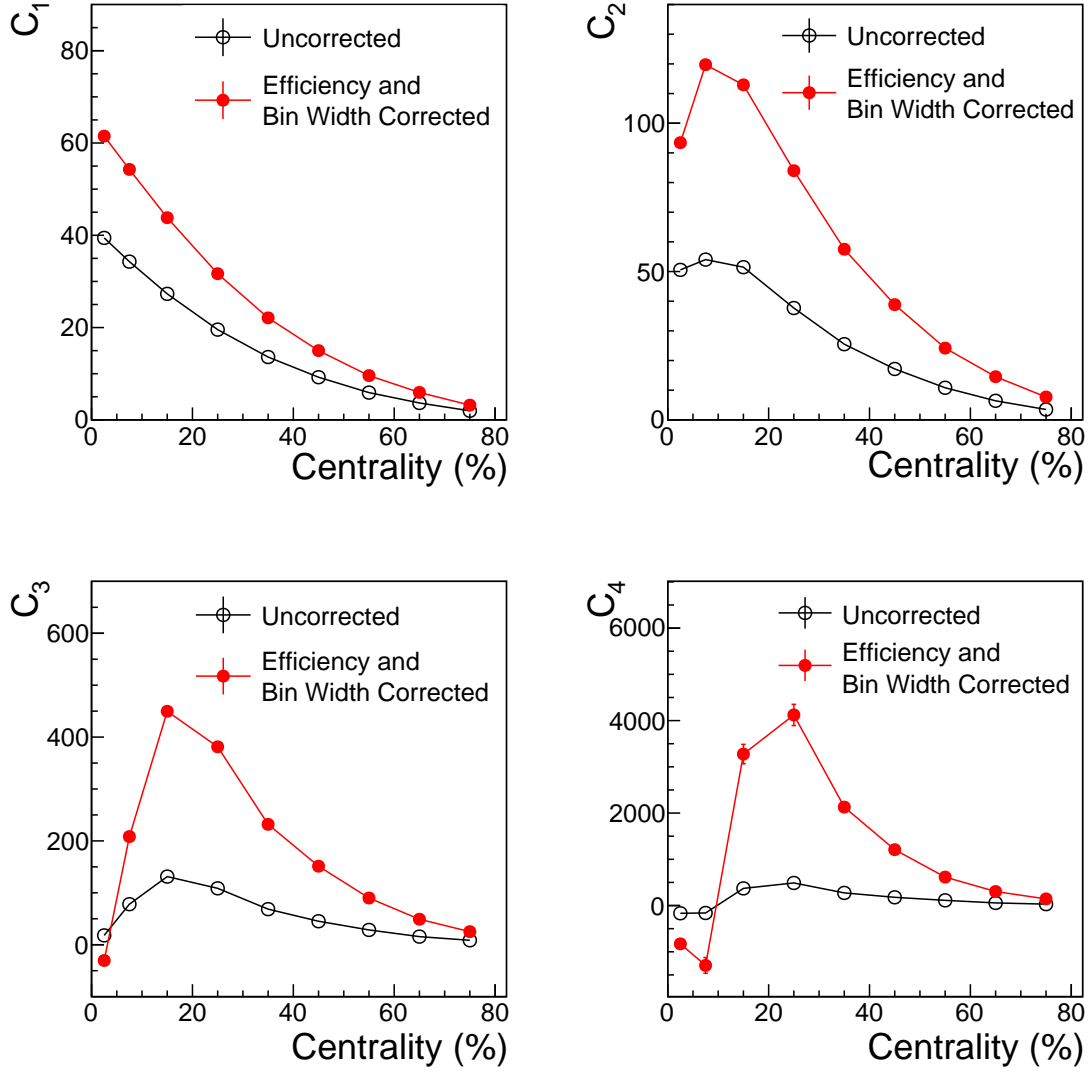
Figure 5.16: Net proton cumulants in various centrality classes before and after efficiency and bin width correction in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c and $E_{lab} = 10$ AGeV.
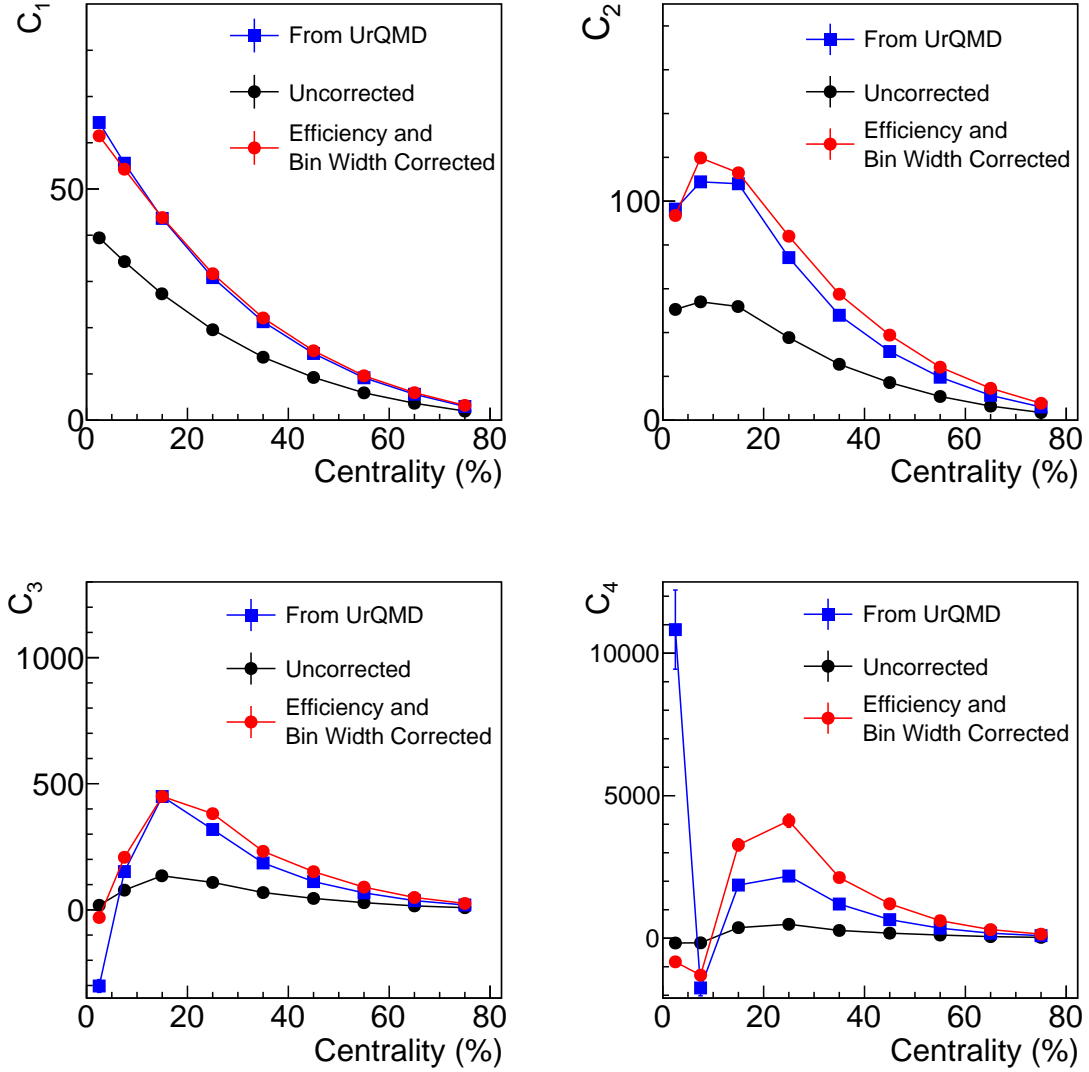
Figure 5.17: Net proton cumulants in various centrality classes before and after efficiency and bin width correction in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c and $E_{lab} = 10$ AGeV. For comparision, the cumulants of the incident net-proton distribution are also shown.

37

5.5 Net-Charge Fluctuation

The following detectors have been used for this analysis.

1. **Micro Vertex Detector** for primary collision vertex, and track reconstruction.

2. **Silicon Tracking System** for track reconstruction, and momentum determination.

The cuts used for the analysis of net-charge distribution have been discussed in section 5.3. Since all the charged particles in the pseudorapidity acceptance region are selected for analysis, we divide the pseudorapidity distribution into two distinct regions for analysis, and reference multiplicity (RefMult) selection. The RefMult is chosen in the range $1.5 < \eta < 1.8$ & $2.8 < \eta < 3.8$, while the particles in the range $1.8 < \eta < 2.8$ are used for net-charge fluctuation measurements. The relevant regions are shown in Fig. 5.19.

The RefMult is shown in Fig. 5.20. The number of events in each centrality class is listed in Table 5.2. The multiplicity distribution of positive and negative charges in every centrality bin is shown in Fig. 5.21. The cumulants have been corrected for bin width effects, but not for efficiency. This will be done later.
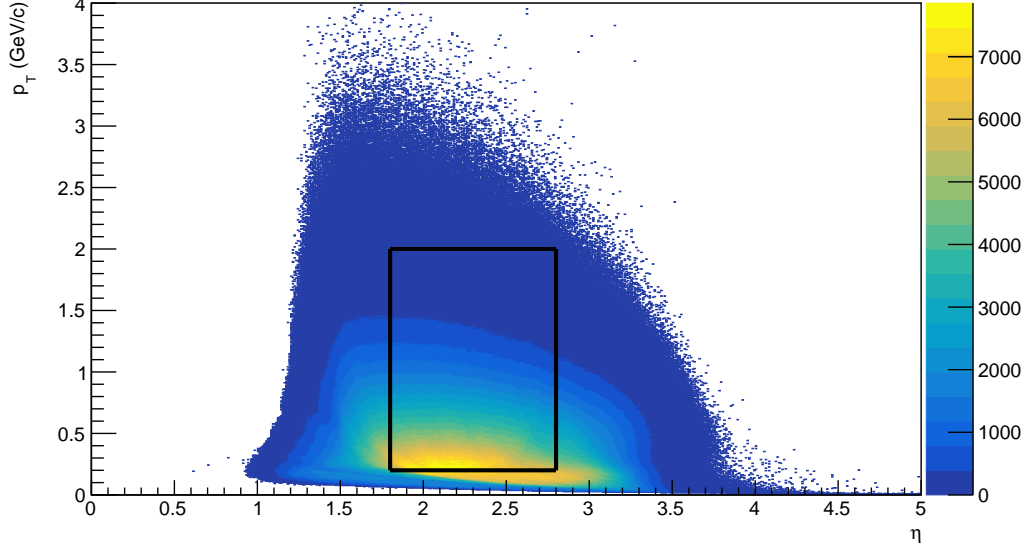
Figure 5.18: The distribution of pseudorapidity with transverse momentum p_T for all charged particles in the η range 1.5-3.8, p_T range 0.2 -2 GeV/c and $E_{lab} = 10$ AGeV. The tracks within the rectangular region have been chosen for analysis.
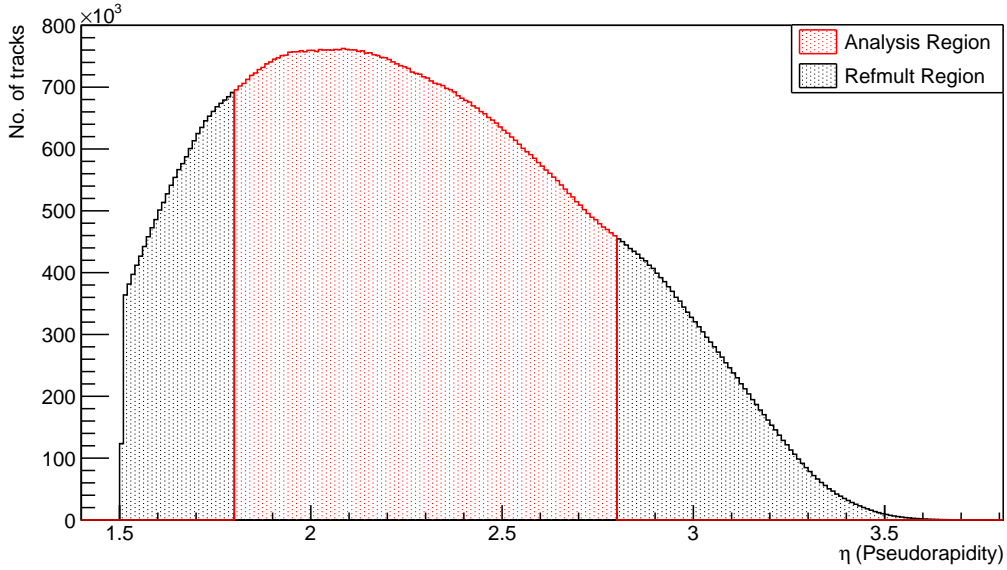
Figure 5.19: Pseudorapidity distribution of all charged particles in the η range 1.5-3.8, p_T range 0.2 -2 GeV/c, and $E_{lab} = 10$ AGeV. The *Refmult* region is used for constructing centrality classes.
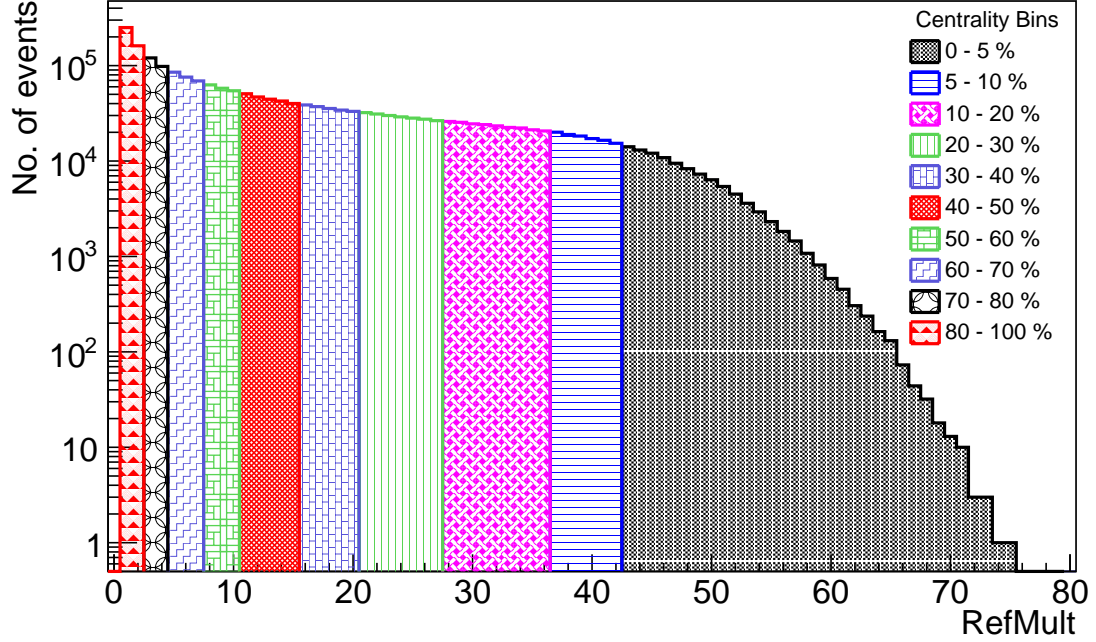
Figure 5.20: RefMult distribution for net-charge analysis in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c, and $E_{lab} = 10$ AGeV. $0 - 5\%$ represents the most central collisions, while $70 - 80\%$ represents the most peripheral collisions used in our analysis.

Table 5.2: Number of events in each centrality bin in net-charge analysis for 2.78 million events of Au-Au collision at $E_{lab} = 10$ AGeV

Centrality (%)	No. of Events
0 - 5	107.6k
5- 10	106.3k
10 - 20	210k
20 - 30	204.4k
30 - 40	178.7k
40 - 50	224.8k
50 -60	175.2k
60 -70	230.2k
70 - 80	218.5k
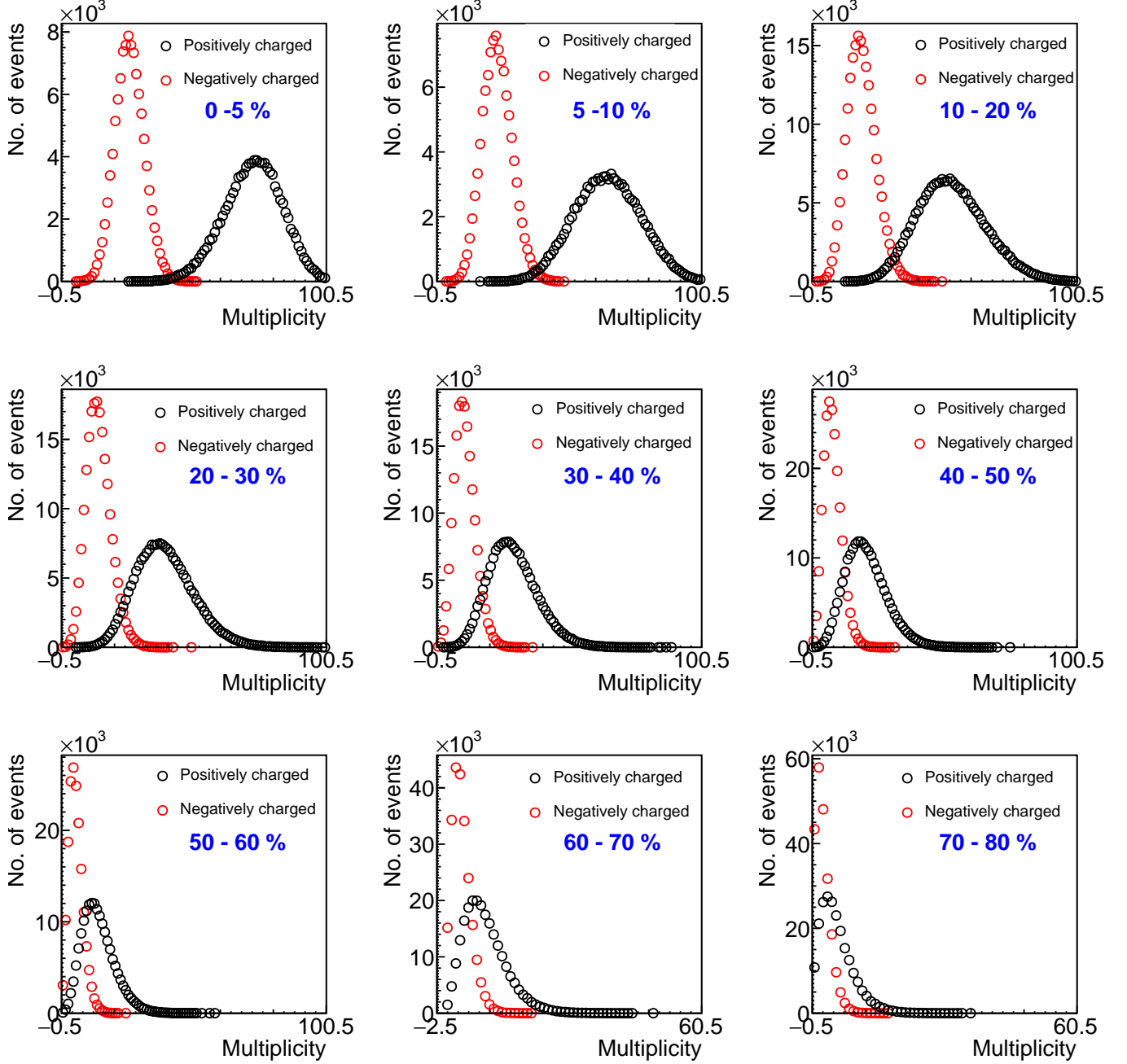
Figure 5.21: Multiplicity distribution of positive and negative charged particles in different centrality classes in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c, and $E_{lab} = 10$ AGeV.

41
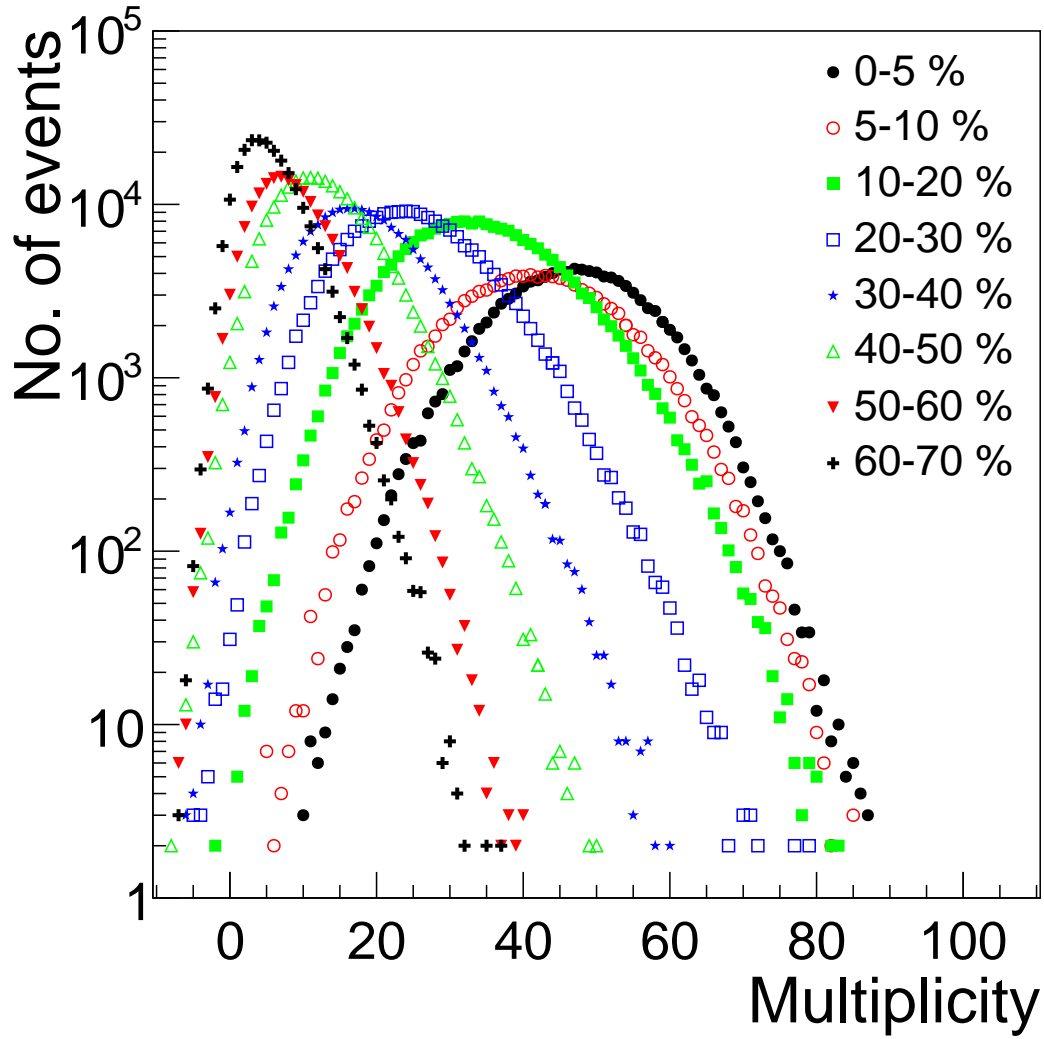
Figure 5.22: Net-charge multiplicity distribution in different centrality bins in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c, and $E_{lab} = 10$AGeV.
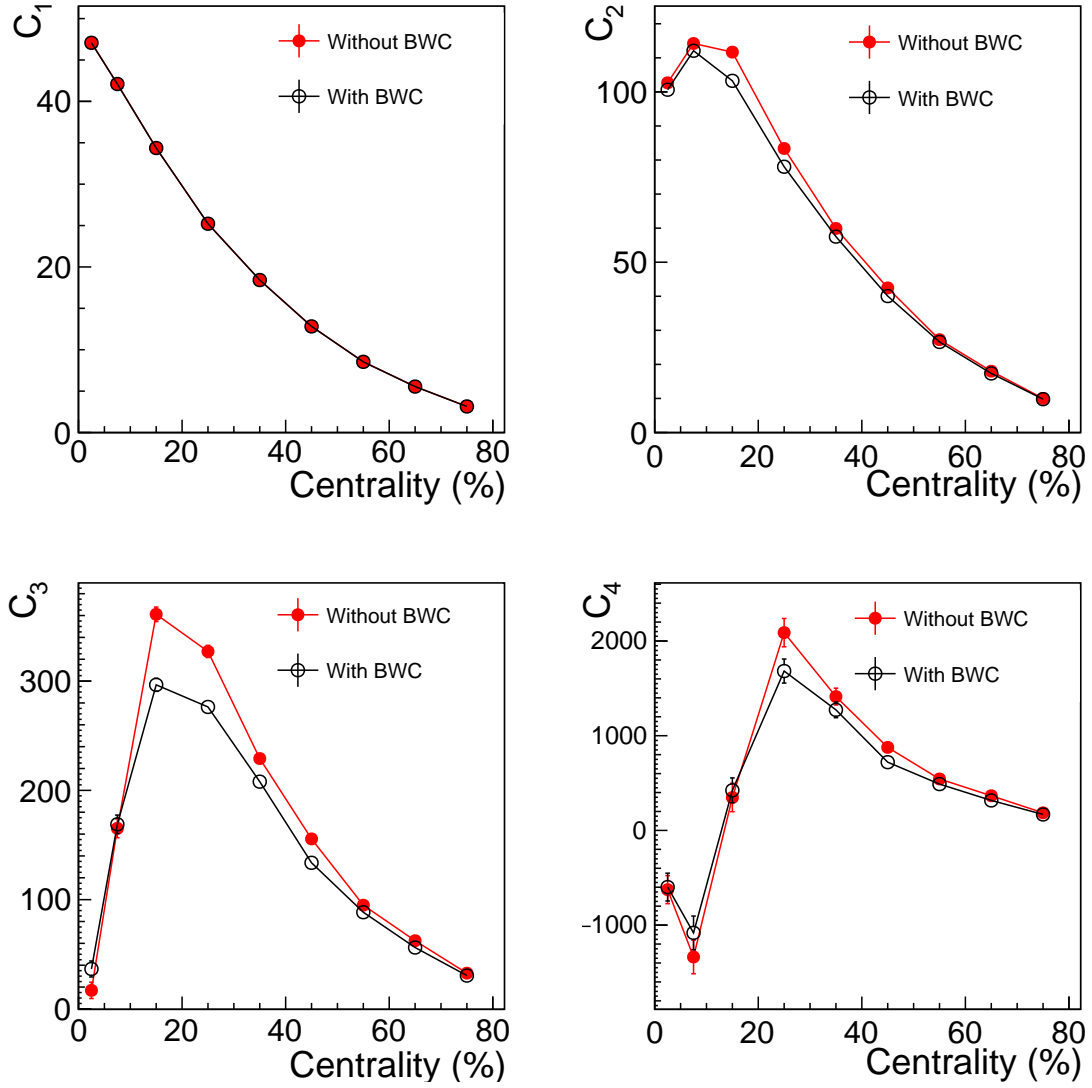
Figure 5.23: Net-charge fluctuations in various centrality classes before and after bin width correction in the η range 1.5-3.8, y range 1.2-2.2, p_T range 0.2 -2 GeV/c, and $E_{lab} = 10$ AGeV.

5.6 Outlook

In the next phase of this thesis, we will work on the following topics.

1. The efficiency correction for the cumulants using binomial distribution didn't produce expected results, and the corrected cumulants differ from the cumulants of the original distribution. We will look into possible reasons for the same.

2. **Efficiency Correction by Unfolding**: We will also correct for efficiency effects using the method of unfolding [23]. The Bayes unfolding algorithm of the RooUnfold package is being used to remove the effects of detection efficiency to determine the true distributions [24]. The method is event-generator dependent, and also needs large statistics to provide correct results. We will compare the results using the two methods.

3. **Analysis at different SIS100 energies**: We will perform the net-charge and net-proton analysis at other SIS100 energies, ranging from 2 AGeV to 11 AGeV to see how the cumulants vary with collision energy.

4. **Fluctuations of net kaon distribution**: We will also perform the same analysis for net kaon distributions. Net kaon is a proxy for net strangeness, and as discussed in Chapter 3, the cumulants of its distribution are related to the correlation length of the system [25].

5. **Spin alignment of vector mesons**: In non-central high energy heavy ion collision, large magnetic field, and large angular momentum are expected to be produced. ALICE [26] and STAR [27] are looking for signatures of these effects by studying the angular distributions of the decay daughters of vector mesons (K^* and ϕ) in Pb-Pb and Au-Au collisions respectively. In presence of

polarisation, it is seen that the angular distributions become non-uniform, and thus ρ_{00}, the zeroth element of the spin density matrix differs from $1/3$, the no spin-alignment scenario. Any deviation from $\rho_{00} = 1/3$ leads to net polarisation of vector mesons. We will look at the spin alignment of K^* and ϕ in Au-Au collisions at SIS100 energies.

References

[1] Giubellino, P., *Nucl. Phys. A*, **715**, 441 (2003).

[2] Schaefer, Bernd-Jochen et al. *Prog.Part.Nucl.Phys.*, **62**, (2009).

[3] Aoki, Y. et al, *Nature*, **443**, 675 (2006).

[4] Ejiri, S., *Phys. Rev. D*, **78**, 074507 (2008).

[5] Adamczyk, L., et al. (STAR Collaboration), *Phys. Rev. Lett.*, **113**, 092301 (2014).

[6] Adamczyk, L., et al. (STAR Collaboration), *Phys. Rev. Lett.*, **112**, 032302 (2014).

[7] Senger, P. and the Cbm Collaboration, *J. Phys.: Conf. Ser.*, **50** 357 (2006).

[8] Hohne, C. et al.: Part V CBM Experiment. Lect. Notes Phys. 814, 849972 (2011).

[9] Heuser, J.M., *Nuclear Physics A*, **830**, 563c566c (2009).

[10] Aggarwal, M. M., et al., *Phys. Rev. Lett.*, **105**, 022302(2010).

[11] Cleymans, J., Oeschler, H., Redlich, K., Wheaton, S., *Phys. Rev. C*, **73**, 034905 (2006).

[12] Stephanov, M. A., *International Journal of Modern Physics A*,**20**, 4387-4392 (2005).

[13] Koch, V., Majumder, A. and Randrup, J., *Phys. Rev.Lett.*, **95**, 182301 (2005).

[14] Gupta, S., Mohanty, B., *Science*, **332**, 1525-1528 (6037).

[15] Athanasiou, C., et al., *Phys. Rev. D*, **82**, 074008 (2010).

[16] Hatta, Y., and Stephanov, M. A., *Phys. Rev. Lett.*, **91**, 102003 (2003).

[17] Bass, S.A et al., *Prog. Part. Nucl. Phys.*, **41**, 225370 (1998).

[18] Bleicher, M. et al., *J. Phys.*, **G25**, 18591896 (1999).

[19] Senger, P., Herrmann, N., *Nuclear Physics News*, **28**, 23-27 (2018).

[20] Miller, M., Reygers, K., Sanders, S. J. and Steinberg, P., *Ann. Rev. Nucl. Part. Sci.*, **57**, 205(2007).

[21] Abelev, B. et al. arXiv:1301.4361v3 [nucl-ex].

[22] Mukherjee, M. et al., *J. Phys. G: Nucl. Part. Phys.*, **43**, 085102 (2016).

[23] Garg, P. et al. J.Phys. G40, 055103 (2013).

[24] Adye, T. [arXiv:1105.1160 [physics.data-an]]. RooUnfold package.

[25] Adamczyk, L., et al. (STAR Collaboration), *Phys. Lett. B*, **785**, 551-560 (2018).

[26] Mohanty, B., *EPJ Web of Conferences*, **171**, 16008 (2018).

[27] Abelev, B. I., et al. [STAR Collaboration], *Phys. Rev. C*, **77**, 061902 (2008).

Appendix A

$(run\_mc.C)$

```
// ————————————————————————————————————————————
//
// Macro for standard transport simulation using UrQMD input and GEANT3
//
// V. Friese    22/02/2007
//
// Version 2016−02−05
//
// For the setup (geometry and field), predefined setups can be chosen
// by the second argument. A list of available setups is given below.
// The input file can be defined explicitly in this macro or by the
// third argument. If none of these options are chosen, a default
// input file distributed with the source code is selected.
//
// ————————————————————————————————————————————


void run_mc(Int_t nEvents = 2,
const char* setupName = "sis100_electron",
//                      const char* setupName = "sis100_debug",
//                      const char* setupName = "sis100_hadron",
//                      const char* setupName = "sis100_muon_jpsi",
//                      const char* setupName = "sis100_muon_lmvm",
const char* inputFile = "")
{

// ════════════════════════════════════════════
//         Adjust this part according to your requirements

// ———— Environment   ————————————————————————
TString myName = "run_mc";  // this macro's name for screen output
TString srcDir = gSystem->Getenv("VMCWORKDIR");  // top source directory
// ————————————————————————————————————————————


// ————  In− and output file names  ————————————————
TString inFile = "";  // give here or as argument; otherwise default is taken
TString outDir  = "data/";
TString outFile = outDir + setupName + "_test.mc.root";
TString parFile = outDir + setupName + "_params.root";
TString geoFile = outDir + setupName + "_geofile_full.root";
// ————————————————————————————————————————————
```

```
// ——— Logger settings ————————————————————————————————————————
TString logLevel     = "INFO";
TString logVerbosity = "LOW";
// ——————————————————————————————————————————————————————————————


// ——— Define the target geometry ——————————————————————————————
//
// The target is not part of the setup, since one and the same setup can
// and will be used with different targets.
// The target is constructed as a tube in z direction with the specified
// diameter (in x and y) and thickness (in z). It will be placed at the
// specified position as daughter volume of the volume present there. It is
// in the responsibility of the user that no overlaps or extrusions are
// created by the placement of the target.
//
TString  targetElement   = "Gold";
Double_t targetThickness = 0.025;  // full thickness in cm
Double_t targetDiameter  = 2.5;    // diameter in cm
Double_t targetPosX      = 0.;     // target x position in global c.s. [cm]
Double_t targetPosY      = 0.;     // target y position in global c.s. [cm]
Double_t targetPosZ      = 0.;     // target z position in global c.s. [cm]
Double_t targetRotY      = 0.;     // target rotation angle around the y axis [deg]
// ——————————————————————————————————————————————————————————————


// ——— Define the creation of the primary vertex ——————————————————
//
// By default, the primary vertex point is sampled from a Gaussian
// distribution in both x and y with the specified beam profile width,
// and from a flat distribution in z over the extension of the target.
// By setting the respective flags to kFALSE, the primary vertex will always
// at the (0., 0.) in x and y and in the z centre of the target, respectively.
//
Bool_t smearVertexXY = kTRUE;
Bool_t smearVertexZ  = kTRUE;
Double_t beamWidthX  = 0.1;  // Gaussian sigma of the beam profile in x [cm]
Double_t beamWidthY  = 0.1;  // Gaussian sigma of the beam profile in y [cm]
// ——————————————————————————————————————————————————————————————


// In general, the following parts need not be touched
// ══════════════════════════════════════════════════════════════


// ——————  Timer  ————————————————————————————————————————————————
TStopwatch timer;
timer.Start();
// ——————————————————————————————————————————————————————————————
```

```
// ——————      Debug option    ————————————————————————————————
gDebug = 0;
// ————————————————————————————————————————————————————————————


// ——————      Remove old CTest runtime dependency file    ——————————————————
TString depFile = Remove_CTest_Dependency_File(outDir, "run_mc" , setupName);
// ————————————————————————————————————————————————————————————



// ——————      Create simulation run    ——————————————————————————————————
FairRunSim* run = new FairRunSim();
run−>SetName("TGeant3");               // Transport engine
run−>SetOutputFile(outFile);           // Output file
run−>SetGenerateRunInfo(kTRUE);        // Create FairRunInfo file
// ————————————————————————————————————————————————————————————


// ——————      Logger settings    ——————————————————————————————————————
FairLogger::GetLogger()−>SetLogScreenLevel(logLevel.Data());
FairLogger::GetLogger()−>SetLogVerbosityLevel(logVerbosity.Data());
// ————————————————————————————————————————————————————————————


// ——————      Load the geometry setup    ——————————————————————————————
std::cout << std::endl;
TString setupFile = srcDir + "/geometry/setup/setup_" + setupName + ".C";
TString setupFunct = "setup_";
setupFunct = setupFunct + setupName + "()";
std::cout << "−I−_" << myName << ":_Loading_macro_" << setupFile << std::endl;
gROOT−>LoadMacro(setupFile);
gROOT−>ProcessLine(setupFunct);
// You can modify the pre−defined setup by using
// CbmSetup::Instance()−>RemoveModule(ESystemId) or
// CbmSetup::Instance()−>SetModule(ESystemId, const char*, Bool_t) or
// CbmSetup::Instance()−>SetActive(ESystemId, Bool_t)
// See the class documentation of CbmSetup.
// ————————————————————————————————————————————————————————————


// ——————      Input file    ————————————————————————————————————————————
std::cout << std::endl;
TString defaultInputFile = srcDir + "/input/urqmd.auau.10gev.centr.root";
if ( inFile.IsNull() ) {  // Not defined in the macro explicitly
if ( strcmp(inputFile, "") == 0 ) {  // not given as argument to the macro
inFile = defaultInputFile;
}
else inFile = inputFile;
```

51

```
}
std::cout << "-I-_" << myName << ":_Using_input_file_" << inFile << std::endl;
// ————————————————————————————————————————————————————————


// ——————      Create media      ————————————————————————————————————
std::cout << std::endl;
std::cout << "-I-_" << myName << ":_Setting_media_file" << std::endl;
run->SetMaterials("media.geo");          // Materials
// ————————————————————————————————————————————————————————


// ——————      Create and register modules    ————————————————————————————
std::cout << std::endl;
TString macroName = gSystem->Getenv("VMCWORKDIR");
macroName += "/macro/run/modules/registerSetup.C";
std::cout << "Loading_macro_" << macroName << std::endl;
gROOT->LoadMacro(macroName);
gROOT->ProcessLine("registerSetup()");
// ————————————————————————————————————————————————————————


// ——————      Create and register the target    ————————————————————————————
std::cout << std::endl;
std::cout << "-I-_" << myName << ":_Registering_target" << std::endl;
CbmTarget* target = new CbmTarget(targetElement.Data(),
targetThickness,
targetDiameter);
target->SetPosition(targetPosX, targetPosY, targetPosZ);
target->SetRotation(targetRotY);
target->Print();
run->AddModule(target);
// ————————————————————————————————————————————————————————


// ——————      Create magnetic field    ————————————————————————————————
std::cout << std::endl;
std::cout << "-I-_" << myName << ":_Registering_magnetic_field" << std::endl;
CbmFieldMap* magField = CbmSetup::Instance()->CreateFieldMap();
if ( ! magField ) {
std::cout << "-E-_run_sim_new:_No_valid_field!";
return;
}
run->SetField(magField);
// ————————————————————————————————————————————————————————


// ——————      Create PrimaryGenerator    ————————————————————————————————
std::cout << std::endl;
std::cout << "-I-_" << myName << ":_Registering_event_generators" << std::endl;
FairPrimaryGenerator* primGen = new FairPrimaryGenerator();
```

```
// --- Uniform distribution of event plane angle
primGen->SetEventPlane(0., 2. * TMath::Pi());
// --- Get target parameters
Double_t tX = 0.;
Double_t tY = 0.;
Double_t tZ = 0.;
Double_t tDz = 0.;
if ( target ) {
target->GetPosition(tX, tY, tZ);
tDz = target->GetThickness();
}
primGen->SetTarget(tZ, tDz);
primGen->SetBeam(0., 0., beamWidthX, beamWidthY);
primGen->SmearGausVertexXY(smearVertexXY);
primGen->SmearVertexZ(smearVertexZ);
//
// TODO: Currently, there is no guaranteed consistency of the beam profile
// and the transversal target dimension, i.e., that the sampled primary
// vertex falls into the target volume. This would require changes
// in the FairPrimaryGenerator class.
// ———————————————————————————————————————————————

// Use the CbmUnigenGenrator for the input
CbmUnigenGenerator*  uniGen = new CbmUnigenGenerator(inFile);
primGen->AddGenerator(uniGen);
run->SetGenerator(primGen);
// ———————————————————————————————————————————————


// —————  Run initialisation  —————————————————————————————
std::cout << std::endl;
std::cout << "-I-_" << myName << ":_Initialise_run" << std::endl;
run->Init();
// ———————————————————————————————————————————————


// —————  Runtime database  ——————————————————————————————
std::cout << std::endl << std::endl;
std::cout << "-I-_" << myName << ":_Set_runtime_DB" << std::endl;
FairRuntimeDb* rtdb = run->GetRuntimeDb();
CbmFieldPar* fieldPar = (CbmFieldPar*) rtdb->getContainer("CbmFieldPar");
fieldPar->SetParameters(magField);
fieldPar->setChanged();
fieldPar->setInputVersion(run->GetRunId(),1);
Bool_t kParameterMerged = kTRUE;
FairParRootFileIo* parOut = new FairParRootFileIo(kParameterMerged);
parOut->open(parFile.Data());
rtdb->setOutput(parOut);
rtdb->saveOutput();
rtdb->print();
// ———————————————————————————————————————————————
```

```
// ———      Start run   —————————————————————————————
std::cout << std::endl << std::endl;
std::cout << "-I- " << myName << ": Starting run" << std::endl;
run->Run(nEvents);
// —————————————————————————————————————————————————


// ———      Finish   ———————————————————————————————
run->CreateGeometryFile(geoFile);
timer.Stop();
Double_t rtime = timer.RealTime();
Double_t ctime = timer.CpuTime();
std::cout << std::endl << std::endl;
std::cout << "Macro finished successfully." << std::endl;
std::cout << "Output file is "    << outFile << std::endl;
std::cout << "Parameter file is " << parFile << std::endl;
std::cout << "Geometry file is "  << geoFile << std::endl;
std::cout << "Real time " << rtime << " s, CPU time " << ctime
<< "s" << std::endl << std::endl;
// —————————————————————————————————————————————————
```

Appendix B

($run\_reco.C$)

```
// ————————————————————————————————————————————————
//
// Macro for digitisation and reconstruction of MC events with standard settings
//
// "old", event−by−event simulation and reconstruction chain.
// For time−based simulation, look at run_digi_tb.C
// and run_reco_tb_digi.C and run_reco_tb_track.C.
//
// This macro combines digitisation and reconstruction by using the
// macro modules digitize.C and reconstruct.C.
//
// The input file (...mc.root) can be chosen directly in the macro (line 42).
// or through the third argument to the macro call.
// If neither is done, the standard input for the regular tests is used.
//
// V. Friese    24/02/2006
// Version      18/03/2017 (V. Friese)
//
// ————————————————————————————————————————————————


void run_reco(Int_t nEvents = 2,
const char* setupName = "sis100_electron",
const char* inputFile = "")
{

// ================================================================
//             Adjust this part according to your requirements

// ——— Logger settings ————————————————————————————
TString logLevel     = "INFO";
TString logVerbosity = "LOW";
// ————————————————————————————————————————————————


// ———— Environment ——————————————————————————————
TString myName = "run_reco";  // this macro's name for screen output
TString srcDir = gSystem->Getenv("VMCWORKDIR");  // top source directory
// ————————————————————————————————————————————————


// ———— In− and output file names —————————————————————
```

```
TString inFile = ""; // give here or as argument; otherwise default is taken
TString outDir  = "data/";
TString outFile = outDir + setupName + "_test.eds.root";  // Output file (reco)
TString parFile = outDir + setupName + "_params.root";     // Parameter file
// ————————————————————————————————————————————————


// ————— Remove old CTest runtime dependency file ——————————————
TString depFile = Remove_CTest_Dependency_File(outDir, "run_reco", setupName);
// ————————————————————————————————————————————————


// ————— Load the geometry setup  ——————————————————————————
std::cout << std::endl;
TString setupFile = srcDir + "/geometry/setup/setup_" + setupName + ".C";
TString setupFunct = "setup_";
setupFunct = setupFunct + setupName + "()";
std::cout << "-I-_" << myName << ": _Loading_macro_" << setupFile << std::endl;
gROOT->LoadMacro(setupFile);
gROOT->ProcessLine(setupFunct);
// You can modify the pre-defined setup by using
// CbmSetup::Instance()->RemoveModule(ESystemId) or
// CbmSetup::Instance()->SetModule(ESystemId, const char*, Bool_t) or
// CbmSetup::Instance()->SetActive(ESystemId, Bool_t)
// See the class documentation of CbmSetup.
// ————————————————————————————————————————————————


// ————— Parameter files as input to the runtime database  ——————————
std::cout << std::endl;
std::cout << "-I-_" << myName << ": _Defining_parameter_files_" << std::endl;
TList *parFileList = new TList();
TString geoTag;

// - TRD digitisation parameters
if ( CbmSetup::Instance()->GetGeoTag(kTrd, geoTag) ) {
TObjString* trdFile = new TObjString(srcDir + "/parameters/trd/trd_" + geoTag + ".dig
parFileList->Add(trdFile);
std::cout << "-I-_" << myName << ": _Using_parameter_file_"
<< trdFile->GetString() << std::endl;
}

// - TOF digitisation parameters
if ( CbmSetup::Instance()->GetGeoTag(kTof, geoTag) ) {
TObjString* tofFile = new TObjString(srcDir + "/parameters/tof/tof_" + geoTag + ".dig
parFileList->Add(tofFile);
std::cout << "-I-_" << myName << ": _Using_parameter_file_"
<< tofFile->GetString() << std::endl;
TObjString* tofBdfFile = new TObjString(srcDir + "/parameters/tof/tof_" + geoTag + ".
parFileList->Add(tofBdfFile);
std::cout << "-I-_" << myName << ": _Using_parameter_file_"
```

56

```
       << tofBdfFile->GetString() << std::endl;
}
// —————————————————————————————————————————————————

// In general, the following parts need not be touched
// ═════════════════════════════════════════════════


// ———  Timer  ——————————————————————————————————————
TStopwatch timer;
timer.Start();
// —————————————————————————————————————————————————


// ———  Debug option  ———————————————————————————————
gDebug = 0;
// —————————————————————————————————————————————————


// ———  Input file  —————————————————————————————————
std::cout << std::endl;
TString defaultInputFile = "data/";
defaultInputFile = defaultInputFile + setupName + "_test.mc.root";
if ( inFile.IsNull() ) {  // Not defined in the macro explicitly
if ( strcmp(inputFile, "") == 0 ) {  // not given as argument to the macro
inFile = defaultInputFile;
}
else inFile = inputFile;
}
std::cout << "-I- " << myName << ": Using input file " << inFile << std::endl;
// —————————————————————————————————————————————————


// ———  FairRunAna  —————————————————————————————————

FairRunAna *run = new FairRunAna();
FairFileSource* inputSource = new FairFileSource(inFile);
run->SetSource(inputSource);
run->SetOutputFile(outFile);
run->SetGenerateRunInfo(kTRUE);
run->SetGenerateRunInfo(kTRUE);
Bool_t hasFairMonitor = Has_Fair_Monitor();
if (hasFairMonitor) FairMonitor::GetMonitor()->EnableMonitor(kTRUE);
// —————————————————————————————————————————————————


// ———  Logger settings  ————————————————————————————
FairLogger::GetLogger()->SetLogScreenLevel(logLevel.Data());
FairLogger::GetLogger()->SetLogVerbosityLevel(logVerbosity.Data());
// —————————————————————————————————————————————————
```

```
// ——— Mc Data Manager ————————————————————————
CbmMCDataManager* mcManager=new CbmMCDataManager("MCManager", 1);
mcManager->AddFile(inFile);
run->AddTask(mcManager);
// ————————————————————————————————————————————————


// ——— Digitisers ————————————————————————
std::cout << std::endl;
TString macroName = gSystem->Getenv("VMCWORKDIR");
macroName += "/macro/run/modules/digitize.C";
std::cout << "Loading macro " << macroName << std::endl;
gROOT->LoadMacro(macroName);
gROOT->ProcessLine("digitize()");
// ————————————————————————————————————————————————



// ——— Reconstruction tasks ————————————————————————
std::cout << std::endl;
macroName = srcDir + "/macro/run/modules/reconstruct.C";
std::cout << "Loading macro " << macroName << std::endl;
gROOT->LoadMacro(macroName);
Bool_t recoSuccess = gROOT->ProcessLine("reconstruct()");
if ( ! recoSuccess ) {
std::cerr << "-E-" << myName << ": error in executing " << macroName
<< std::endl;
return;
}
std::cout << "-I-" << myName << ": " << macroName << " excuted successfully"
<< std::endl;
// ————————————————————————————————————————————————



// ——— Parameter database ————————————————————————
std::cout << std::endl << std::endl;
std::cout << "-I- " << myName << ": Set runtime DB" << std::endl;
FairRuntimeDb* rtdb = run->GetRuntimeDb();
FairParRootFileIo* parIo1 = new FairParRootFileIo();
FairParAsciiFileIo* parIo2 = new FairParAsciiFileIo();
parIo1->open(parFile.Data(),"UPDATE");
rtdb->setFirstInput(parIo1);
if ( ! parFileList->IsEmpty() ) {
parIo2->open(parFileList, "in");
rtdb->setSecondInput(parIo2);
}
// ————————————————————————————————————————————————



// ——— Run initialisation ————————————————————————
std::cout << std::endl;
std::cout << "-I- " << myName << ": Initialise run" << std::endl;
run->Init();
```

58

```
// ————————————————————————————————————————————————————————————

rtdb->setOutput(parIo1);
rtdb->saveOutput();
rtdb->print();

// ———      Start run    ————————————————————————————————————
std::cout << std::endl << std::endl;
std::cout << "-I-_" << myName << ":_Starting_run" << std::endl;
run->Run(0, nEvents);
// ————————————————————————————————————————————————————————————


// ———      Finish     ————————————————————————————————————
timer.Stop();
Double_t rtime = timer.RealTime();
Double_t ctime = timer.CpuTime();
std::cout << std::endl << std::endl;
std::cout << "Macro_finished_succesfully." << std::endl;
std::cout << "Output_file_is_" << outFile << std::endl;
std::cout << "Parameter_file_is_" << parFile << std::endl;
std::cout << "Real_time_" << rtime << "_s,_CPU_time_" << ctime << "_s"
<< std::endl;
std::cout << std::endl;
std::cout << "_Test_passed" << std::endl;
std::cout << "_All_ok_" << std::endl;
// ————————————————————————————————————————————————————————————
```