Universiteit Utrecht

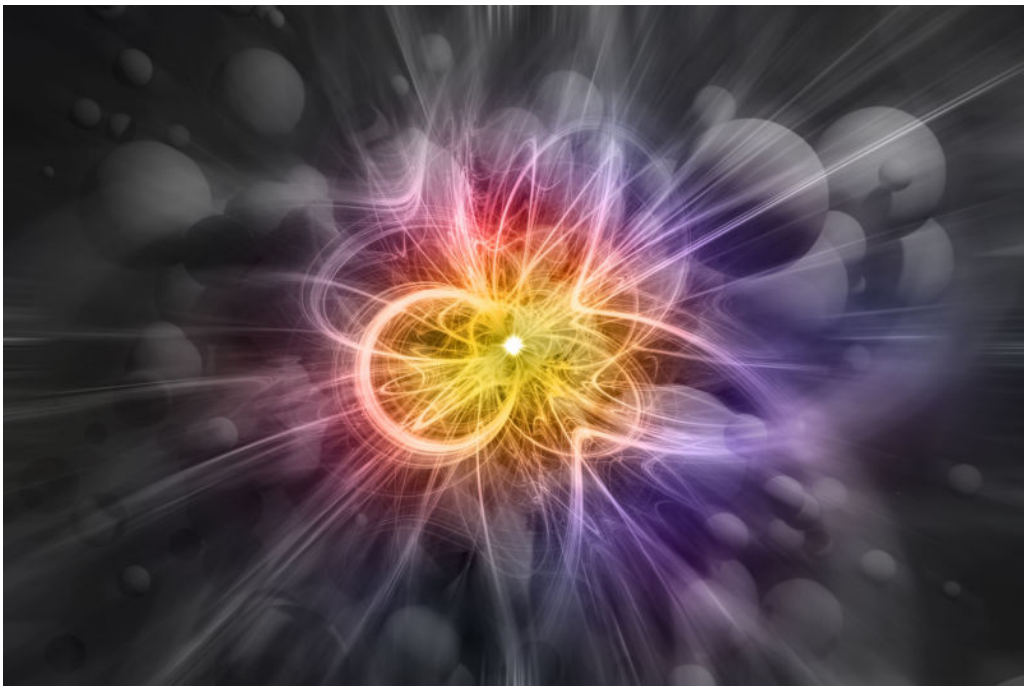
Natuur- en Sterrenkunde (TWIN)

Simulation of symmetric cumulants in heavy-ion collisions

How theory and experiments can support each other to understand the quark-gluon plasma

Bachelor Thesis

Robin van Bijleveld



Supervisors:

Prof. dr. Raimond Snellings
Institute for Subatomic Physics

Dr. Umut Gürsoy
Institute for Theoretical Physics

MSc Govert Nijs
Institute for Theoretical Physics

January 2020

Abstract

The quark-gluon plasma (QGP) is a fluid of free quarks and gluons that exists under the extreme conditions of high temperature and/or high pressure. This plasma can be investigated by performing high energy heavy-ion collisions. These collisions can also be simulated by computer models, which will be done at a centre of mass energy of $\sqrt{s_{NN}} = 5.02$ TeV in this research. These models contain parameters that one would like to know, e.g. parameters for the shear viscosity η/s of the QGP. After the collision, there are different stages that have to be described. One of these stages is the QGP, which can be described by hydrodynamics. We will see how the physics of the different stages works and how it can be implemented in a computer program, from the initial collision to the final particles that hit the detectors. We will then describe how relevant physical information can be extracted from these detections. It will turn out that the momentum distribution transverse to the collision axis will not be isotropic, which can be characterised by the flow coefficients v_n, of which the elliptic flow v_2 will be the most important one. We will also introduce the Symmetric Cumulants, an observable that is not yet implemented in the currently used program. They measure the correlation between the different flow coefficients. We will look at different system sizes to get a complete view of this observable.

The information about different observables can be compared to experimental data. By using Bayesian analysis, one can obtain a good guess for the values of the different parameters. In this way, we can investigate the quark-gluon plasma and the other important physics of a heavy-ion collision. The hope is that the simulated Symmetric Cumulants are sensitive for the chosen set of parameters, so that this additional observable will constrain the parameters further.

Table of Contents

Introduction

The **quark-gluon plasmsa** (QGP) is a state of matter in quantum chromodynamics (QCD), as illustrated in the phase diagram of figure 1. The horizontal axis reads the baryon chemical potential, but this can be read as baryon density, as the two are related to each other. It exists under the conditions of high temperature and/or high density or, equivalently, pressure. In this state, hadrons do not exist in the way we are used to: the **quarks** that make up these hadrons are **deconfined** and can therefore exist on their own. Normally, they are confined due to the strong nuclear force and form hadrons. We could have mesons ($q\bar{q}$) or baryons (qqq), which are all colour-neutral. Colour can be exchanged by the force carriers, the **gluons**. Deconfinement takes place, because the strong coupling between quarks becomes small in the case of a QGP. We can indeed see in figure 2 that the coupling constant of the strong interaction α_s decreases with increasing momentum transfer in an interaction. On the other hand, confinement occurs for small momentum transfer.

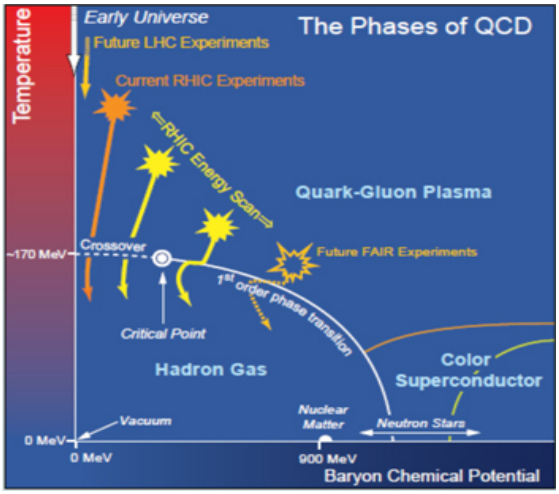


Figure 1: A schematic of the phase diagram of QCD. It contains the different possible phases.

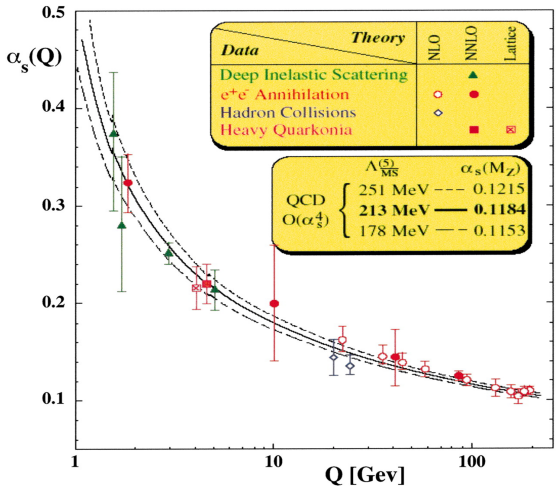


Figure 2: Measurement of the coupling constant of the strong interaction α_s as a function of the momentum transfer Q [1].

It is thought by scientists that a QGP existed in the very early universe, just after the big bang. Only a few microseconds later, the system began to cool down and it became possible for individual particles such as protons and neutrons to form. Nowadays, some scientists think that neutron stars contain a phase similar to a QGP. This is due to the very high density in these stars. As they are often very far away from the earth, it is hard to investigate these stars. Luckily, similar conditions needed to form such a plasma can also be obtained in the lab. For this we need high-energy nuclear collisions and these can be performed in particle accelerators such as the Large Hadron Collider (LHC) at CERN and the Relativistic Heavy-Ion Collider (RHIC) at Brookhaven National Laboratory. A **heavy-ion collision** is an example of such collisions. A heavy-ion is an electrically charged particle that is usually heavier than a He-4 nucleus. After the collision, there is a period where the system is in a quark-gluon plasma phase.

It is believed that a quark-gluon plasma can be described macroscopically using relativistic hydrodynamics. This idea has already existed for more than fifty years and became more prominent in the late 1970's by performed fixed-target heavy-ion collisions [2]. It led to evidence of the formation of dense matter that has a collective expansion. This is what makes a QGP special: it is neither a bunch of hadrons, nor a free gas, as the quarks are strongly coupled to each other in the liquid.

Besides from experiments, heavy-ion collisions can also be studied by using computer programs. This possibility will be explored and investigated in the next chapters. The simulations can then be compared to experimental data, which we will also do.

Chapter 1

Preliminaries

Before we can really start, it is good to get familiarised with some important concepts. In this thesis, we use natural units, i.e. $c = \hbar = k_B = 1$, and the mostly negative metric tensor $g_{\mu\nu} = \eta_{\mu\nu} := \text{diag}(1, -1, -1, -1)$ for flat Minkowski space. Throughout the thesis, we will consider two colliding heavy ions. They will eventually form an ensemble of particles. We can assign a momentum $\vec{p} = \left(p_x, p_y, p_z\right)^T$, mass m, and energy $E = \sqrt{p^2 + m^2}$ to each particle. We will use the convention that the beam of the two colliding particles is in the z-direction, such that the transverse plane will be the xy-plane. There are also some important variables or observables that will be mentioned here and used throughout the thesis.

The system can undergo a **Lorentz transformation**, i.e. a boost with velocity v. For a four-vector a^μ, we have that

$$a'^\mu = \Lambda^\mu{}_\nu \, a^\nu, \qquad \text{with } \Lambda^\mu{}_\nu = \begin{pmatrix} \gamma & 0 & 0 & -v\gamma \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -v\gamma & 0 & 0 & \gamma \end{pmatrix}, \tag{1.1}$$

when we deal with a boost in the z-direction: $\vec{v} = v\hat{e}_z$.

There is a notion of **transverse momentum**, which is defined by

$$p_t := \sqrt{p_x^2 + p_y^2}, \tag{1.2}$$

such that $p_x = p_t \cos\varphi$ and $p_y = p_t \sin\varphi$, with φ the azimuthal angle with respect to a so-called **reaction plane**. This plane is schematically shown in figure 1.1. The angle between this reaction plane and the laboratory frame is called the reaction plane angle and is denoted by Ψ_R.

We will also come across the **rapidity**, defined by

$$y := \frac{1}{2} \ln\left(\frac{E + p_z}{E - p_z}\right). \tag{1.3}$$

This quantity indicates more or less the direction of the particle. If the particle's direction is primarily in the xy-plane, then p_z will be small and y will tend to zero. If, in the other extreme case, the direction is essentially in the $\pm z$-direction (i.e. $E \approx \pm p_z$), then y will tend to $\pm\infty$. The rapidity is a useful quantity, as the difference of the rapidity of two different particles is invariant under a Lorentz transformation, i.e. [3]

$$\begin{aligned} y' = y + \frac{1}{2} \ln\left(\frac{1 - v}{1 + v}\right) = y - \text{arctanh}(v); \\ y_1' - y_2' = y_1 - y_2, \end{aligned} \tag{1.4}$$

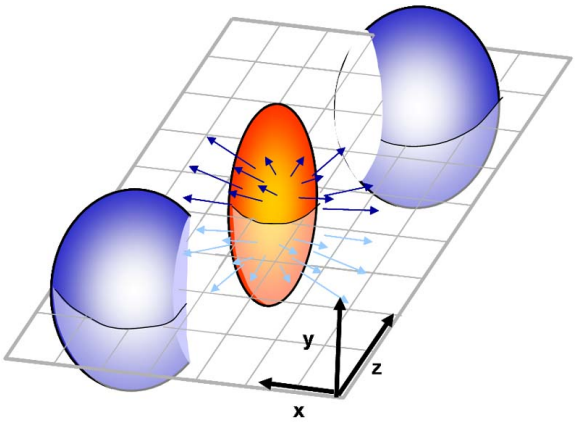


Figure 1.1: Schematic view of the reaction plane as two particles or nuclei collide. The red part is supposed to be the quark-gluon plasma formed after the collision.

with v the velocity of the associated boost in the z-direction. Despite this advantage, the rapidity is not an easy quantity to compute, as one needs to know both the angle w.r.t. the beam and the mass of the particle. It is difficult to measure the latter precisely and that measurement is avoided by introducing the **pseudorapidity** η. For highly energetic particles, the two quantities are almost the same, but the pseudorapidity is easier to

compute. If we thus assume that the particles are highly energetic (i.e. $p \gg m$), we come to the definition of the pseudorapidity[1] [3]:

$$\eta := -\ln\left(\tan\frac{\theta}{2}\right) \tag{1.5}$$

$$= -\frac{1}{2}\ln\left(\frac{\frac{1}{2}-\frac{1}{2}\cos\theta}{\frac{1}{2}+\frac{1}{2}\cos\theta}\right) = \frac{1}{2}\ln\left(\frac{p(1+\cos\theta)}{p(1-\cos\theta)}\right) = \frac{1}{2}\ln\left(\frac{p+p_z}{p-p_z}\right), \tag{1.6}$$

with θ the angle between the trajectory of the particle and the beam pipe, i.e. $p_z = p\cos\theta$. It is indeed easier to compute η as it only depends on the polar angle θ and not on the mass anymore. Note that for massless particles we have that $y = \eta$.

Another important concept is **centrality**. It is of importance because many observables are plotted with the centrality on the horizontal axis. It is determined by the impact parameter b, as shown in figure 1.2, and is normally given in terms of percentages. This parameter b represents the distance between the two centres of the colliding nuclei, so it is related to the overlap between the particles. The centrality is proportional to b^2 [4] and related to the multiplicity of participating particles (the **participants**): a higher centrality results in a smaller multiplicity. We call a collision **central** (i.e. 0% centrality) if $b \approx 0$ and **peripheral** (i.e. 100% centrality) if $b \approx R_1 + R_2$, with R_i the radius of nucleus i. Observables can also be plotted with $N_{\text{trk}}^{\text{off}}$ on the horizontal axis. It represents the amount of trajectories of charged particles, which is based on the multiplicity of charged particles. It takes into account all charged particles that are in some p_t-range and η-range.[2] In this way, we see that centrality and $N_{\text{trk}}^{\text{off}}$ are equivalent. However, it is sometimes easier and more convenient to use $N_{\text{trk}}^{\text{off}}$ in an experimental setup. Simulations have to be done under the exact same conditions, so we adopt it here.

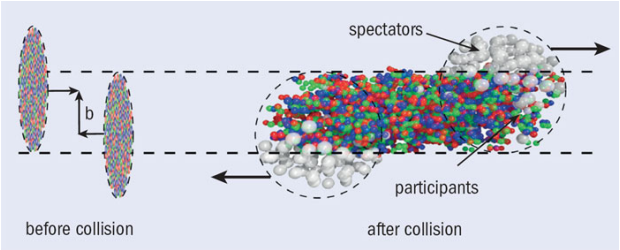


Figure 1.2: Illustration of the impact parameter b and a schematic view of the collision.

The spectators illustrated in figure 1.2 are the particles that do not participate in the collision. They move more or less in the same direction as before the collision, unaltered. They will not be of relevance in this thesis.[3] We also see in the figure that the colliding nuclei are represented by flat disks. This is due to **Lorentz contraction**:

$$L' = L\sqrt{1-v^2} =: \frac{L}{\gamma},$$

as the nuclei move with a relativistic velocity v. The rest length is denoted by L and it is shortened via γ to a length L'.

By manipulation of equation (1.3), we see that $y = \operatorname{arctanh}(p_z/E)$. But then we also have that

$$p_t^2 + m^2 = E^2 - p_z^2 = E^2\left(1-\tanh(y)^2\right) = E^2\frac{1}{\cosh(y)^2}.$$

We can now express E and p_z in terms of transverse momentum and rapidity by

$$E = \sqrt{p_t^2+m^2}\cosh(y), \tag{1.7}$$

$$p_z = \sqrt{p_t^2+m^2}\sinh(y), \tag{1.8}$$

and for the momentum four-vector we can write, for $p_t \gg m$,[4] that

$$p^\mu = \begin{pmatrix} E \\ \vec{p} \end{pmatrix} = p_t \begin{pmatrix} \cosh(y) \\ \cos(\varphi) \\ \sin(\varphi) \\ \sinh(y) \end{pmatrix}. \tag{1.9}$$

[1] This is done by a Taylor expansion around $m/p \ll 1$.
[2] The appropriate ranges will be mentioned in Chapter 4 when we discuss the results.
[3] This does not mean that they are useless. Spectators induce electromagnetic effects which can be studied [5].
[4] This is a valid assumption, as we will restrict ourselves in the simulation to particles within a range $-\eta_{\max} < \eta < \eta_{\max}$, with $\eta_{\max}$ in the order of magnitude of 1. As argued for y, this means that p_z is small, so that $p \approx p_t$.

Chapter 2

Simulation of heavy-ion collisions

We argued in the introduction that a quark-gluon can be described hydrodynamically. This is different from how one would normally describe what would happen after a collision of (subatomic) particles. In this chapter, we will develop the theory of relativistic hydrodynamics to describe the plasma. However, the plasma is not the only stage that exists after a heavy-ion collision. The different stages are shown in figure 2.1 and we will shortly describe them here. Later in this chapter, we will explain these stages in more detail and also argue how these stages are implemented in the simulation program. For a full explanation with all the mathematical and physical details, I refer to [6] and most of the explanation of the simulation is done with this reference in mind. However, the following sections contain enough information to understand the process.

The two nuclei collide at $\tau = 0$. There is a short period of free streaming. At $\tau \sim 1$ fm, we can treat the quark-gluon plasma in a hydrodynamic way. The system will cool down over time and at $\tau \sim 10$ fm, we have a conversion to hadrons: a hadron resonance gas. This gas is the low temperature phase of quantum chromodynamics, as seen in figure 1. At a certain moment, chemical freeze-out will occur. This means that the particles are in their final stage and will not decay into other particles anymore before they reach the detectors. After that, there is the kinetic freeze-out after which the particles will not interact with each other anymore. The result is an ensemble of particles that can move around without bumping into other particles. These final particles will be detected, from which one can deduce physical information. This analysis is treated in Chapter 3.

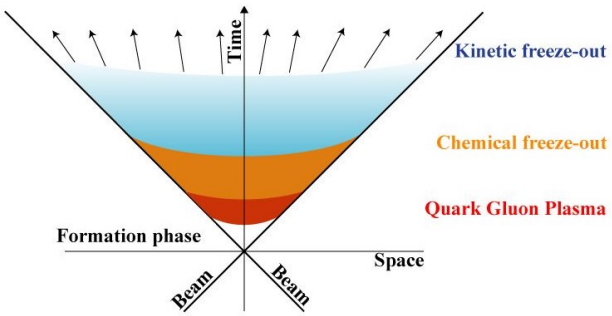


Figure 2.1: Schematic spacetime diagram for a heavy-ion collision with indicated stages.

2.1 Relativistic hydrodynamics

In order to describe a quark-gluon plasma hydrodynamically, we need a relativistic theory of **hydrodynamics**. It is based on the conservation of the so-called **energy-momentum tensor** $T^{\mu\nu}$, which is the conserved current that corresponds to a symmetry in the spacetime translation. To build this tensor, we go step by step. It is built by an expansion of gradients, assuming that the fluid is close to **thermal equilibrium**. This means that we allow small gradients in the velocity and temperature.[5] First we consider an ideal fluid, which corresponds to $T_0^{\mu\nu}$. We do not take into account any gradients, because there are no dissipative effects by definition of an ideal fluid. We can only make use of the hydrodynamic degrees of freedom, which give rise to two Lorentz scalars $\mathcal{E}$ and $\mathcal{P}$ [7], which will turn out to be related to each other. We also use the metric tensor $g^{\mu\nu}$ and the

[5] Small can be a somewhat arbitrary notion. It means that it is much smaller than the typical scale of QCD, which is about the mass of a proton: 1 GeV.

four-velocity u^μ, defined by

$$u^\mu := \frac{dx^\mu}{d\tau} = \gamma_v \begin{pmatrix} 1 \\ \vec{v} \end{pmatrix}, \tag{2.1}$$

with τ the proper time and $\gamma_v := (1-v^2)^{-1/2}$. Note that $u_\mu u^\mu = 1$. As $T^{\mu\nu}$ is symmetric,[6] the most general expression we can come up with is a symmetric superposition of the quantities just mentioned. The tensor will then look like

$$T_0^{\mu\nu} = \mathcal{E}\left(c_0 g^{\mu\nu} + c_1 u^\mu u^\nu\right) + \mathcal{P}\left(c_2 g^{\mu\nu} + c_3 u^\mu u^\nu\right), \tag{2.2}$$

for $c_i \in \mathbb{R}$, with $i = 0, 1, 2, 3$ and $\mathcal{E}$ and $\mathcal{P}$ some variables that we will address in a moment. These prefactors c_i can be determined by looking at $T_0^{\mu\nu}$ in the local rest frame. In this frame, we know that the energy-momentum tensor for an isotropic fluid looks like $\mathrm{diag}(\epsilon, p, p, p)$ and $u^\mu = \left(1, \vec{0}\right)^T$. Here ϵ plays the role of a local energy density and p the local pressure. This requires a small computation, and the result yields that

$$T_0^{\mu\nu} = \epsilon u^\mu u^\nu - p\Delta^{\mu\nu}, \tag{2.3}$$

with $\Delta^{\mu\nu} := g^{\mu\nu} - u^\mu u^\nu$. Note that $\Delta^{\mu\nu} u_\nu = 0$. We can also see now that our unknown variables $\mathcal{E}$ and $\mathcal{P}$ are just the local energy density and the local pressure, respectively.

In principle, $T_0^{\mu\nu}$ has 10 independent components and we only have four equations, namely

$$\partial_\mu T_0^{\mu\nu} = 0, \tag{2.4}$$

as $T_0^{\mu\nu}$ is a conserved quantity. By equation (2.3), it follows that there are only five independent variables, namely ϵ, p and three components of u^μ.[7] However, we still only have four equations, so the system is not closed. The fifth equation that will close this system of equations is the **equation of state** (E.o.S.), which is the pressure as a function of the energy density:

$$p = p(\epsilon). \tag{2.5}$$

This E.o.S. can be a small and elegant expression. For example, in cosmology it takes the form $p = w\epsilon$ [8], with $w \in \mathbb{R}$. If we have a gas of highly relativistic particles, we have that $w = \frac{1}{3}$, and for nonrelativistic matter, we have $w = 0$. In some cases however, it will be a bit more complex. For heavy-ion collisions, one has to do a lattice calculation, which is an advanced topic in quantum field theory. It is out of the scope of this thesis and we will just assume that an expression for the equation of state can be found.

So far, we have not taken into account that there could be dissipative (i.e. viscous) effects, which are caused by gradients in the energy-momentum tensor. If we do this, there is a notion of local temperature and local fluid velocity that are not uniquely defined out of equilibrium, due to these viscous effects. This means that there are different choices of local rest frames of the fluid. For example, we can define many local temperature fields

$$\{T(x^\mu) \,|\, T \text{ a local temperature field}\},$$

which will differ from each other by gradients of hydrodynamic variables, with the constraint that all these fields have to approach the same value once in equilibrium. We could redefine our Lorentz scalars as

$$\mathcal{E} = \epsilon(T) + f_{\mathcal{E}}(\partial T, \partial u), \tag{2.6}$$
$$\mathcal{P} = p(T) + f_{\mathcal{P}}(\partial T, \partial u). \tag{2.7}$$

Redefinitions of fields like $T(x^\mu)$ and u^μ are often called a **frame choice** and look like

$$T(x^\nu) \mapsto T'(x^\nu) = T(x^\nu) + \delta T(x^\nu) \tag{2.8}$$
$$u^\mu(x^\nu) \mapsto u'^\mu(x^\nu) = u^\mu(x^\nu) + \delta u^\mu(x^\nu), \tag{2.9}$$

where $\delta T(x^\nu)$ and $\delta u^\mu(x^\nu)$ are first order in derivatives. We can redefine them as we want, as long as it does not change the energy-momentum tensor. This introduces constraints, and one can deduce that $\mathcal{E} = \epsilon$ and $\mathcal{P} = p - \zeta \nabla_\mu u^\mu$, with ζ a coefficient that turns out to represent the bulk viscosity. This procedure also introduces a term $\pi^{\mu\nu}$ that will be related to the shear viscosity η. A more elaborate explanation and computation is done in [9], but the main result yields (to first order) that the energy-momentum tensor can be written as[8] [10]

$$T^{\mu\nu} = T_0^{\mu\nu} + \pi^{\mu\nu} - \Pi\Delta^{\mu\nu} = \epsilon u^\mu u^\nu - (p + \Pi)\Delta^{\mu\nu} + \pi^{\mu\nu}, \tag{2.10}$$

[6] $T^{\mu\nu}$ can be derived from varying the action with respect to the metric $g^{\mu\nu}$. Because the metric is symmetric, $T^{\mu\nu}$ will be symmetric as well: $T^{\mu\nu} \propto \frac{\delta S}{\delta g_{\mu\nu}}$.

[7] u^μ has four components, but it is constrained by the normalisation condition $u^\mu u_\mu = 1$.

[8] This equation is the relativistic analogue to the Navier-Stokes equation.

with $\pi^{\mu\nu} := 2\eta\sigma^{\mu\nu}$ and $\Pi := -\zeta\nabla_\mu u^\mu$. We used that $\nabla^\mu := \Delta^{\mu\nu}\partial_\nu$, while $\sigma^{\mu\nu}$ is symmetric and traceless and given by

$$\sigma^{\mu\nu} := \frac{1}{2}\left(\nabla^\mu u^\nu + \nabla^\nu u^\mu\right) - \frac{1}{3}\Delta^{\mu\nu}\nabla_\lambda u^\lambda =: \nabla^{\langle\mu} u^{\nu\rangle}, \tag{2.11}$$

where $\nabla^{\langle\mu} u^{\nu\rangle}$ is the short notation for the symmetric and traceless version of $\nabla^\mu u^\nu$. The shear tensor $\pi^{\mu\nu}$ is symmetric and traceless by construction and transverse to u_ν, i.e. $\pi^{\mu\nu}u_\nu = 0$ [11]. This means that $\pi^{\mu\nu}$ has five independent components. The two quantities for the viscosity are usually given as a dimensionless fraction like η/s and ζ/s, with s the entropy density. A QGP is often called a **perfect fluid**, due to its low shear viscosity. Indeed, experiments show that η/s is very small. It even tends to be at the lower bound for strongly coupled plasma, as conjectured by string theory [12, 13]:

$$\frac{\eta}{s} = \frac{1}{4\pi}. \tag{2.12}$$

The dissipative effects in the first order energy-momentum tensor is not the general result. In fact, the general dissipative part is an expansion in powers of gradients of hydrodynamic variables like u^μ. So equation (2.10) only contains this effect up to first order. This first order theory, however, poses a problem of causality and stability of the solutions [14]. Acausality means that the propagation speed is greater than the speed of light: two events are not in causal contact. Müller, Israel and Stewart proposed a solution [15, 16]:

$$\begin{aligned} D\pi^{\mu\nu} &= \frac{1}{\tau_\pi}\Big(- \pi^{\mu\nu} - \eta\sigma^{\mu\nu} + \cdots \Big), \\ D\Pi &= \frac{1}{\tau_\Pi}\Big(- \Pi - \zeta\nabla_\mu u^\mu + \cdots \Big), \end{aligned} \tag{2.13}$$

where $D := u^\mu\nabla_\mu$ and τ_Π and τ_π can be thought of as the relaxation time and have corresponding dimensions. These formulas imply that the viscous terms are calculated in the form of a differential equation that is to be solved for $\pi^{\mu\nu}$ and Π. Equation (2.13) can be made more precise by adding higher order terms in gradients of the velocity [17]. For second order, the energy-momentum tensor will look like equation (2.10), but with the difference that the viscous part contains more terms. These additional terms are partly determined by coupling constants between the shear and bulk viscosity. To get the equations of motion, we will need to solve, together with an equation of state:

$$\partial_\mu T^{\mu\nu} = 0. \tag{2.14}$$

2.2 From collision to quark-gluon plasma

We will now explain how the whole process from a heavy-ion collision to the final detected particles works. First, we choose the kind of particles we want to collide, e.g. two lead nuclei. The simulation program then begins by generating the events given the initial conditions. These initial conditions, as shown in Appendix D, are chosen in such a way that the **centre of mass energy** of two nucleons is $\sqrt{s_{NN}} = 5.02$ TeV. It is defined as

$$\sqrt{s_{NN}} := \sqrt{p_\mu p^\mu} = \sqrt{(E_1 + E_2)^2 - (\vec{p}_1 + \vec{p}_2)^2}. \tag{2.15}$$

This energy mainly depends on the nucleon-nucleon cross section σ_{NN} and a normalisation constant mentioned in Appendix D.

We can not immediately describe the system hydrodynamically. This is because it does not describe the formation of the plasma: it only describes the time evolution. Also, hydrodynamics requires thermal equilibrium and that is certainly not the case right after the collision. That is why there is this short timegap $\Delta\tau \sim 1$ fm between the collision and the hydrodynamic description. In order to describe the system with hydrodynamics, we need to solve equation (2.14). The solution to a differential equation needs initial conditions in order to be defined uniquely. This means that we first need to go from the collision of particles to a scalar field for the energy density or entropy density.[9] There are different models that predict the initial conditions, e.g. the Glauber model and the TRENTo model [18]. In this research, TRENTo (Reduced Thickness Event-by-event Nuclear Topology) is the used model, so we discuss it now. This model finds a function f that is proportional to the entropy density and this function turns out to be related to the thickness of participating particles. The factor that converts this proportional sign to an equal sign is a parameter for the simulation. We first consider a collision between two protons A and B with impact parameter b along the x-direction. We then determine whether they collide and that probability is given by

$$P_{\text{coll}} = 1 - \exp\left[-\sigma_{gg}\int\left(\int\rho_A dz\int\rho_B dz\right)dxdy\right], \tag{2.16}$$

[9] The entropy density is directly related to the energy density by the equation of state and the laws of thermodynamics.

where $\rho_A := \rho_{\text{proton}}(x - b/2, y, z)$ and $\rho_B := \rho_{\text{proton}}(x + b/2, y, z)$. The effective parton-parton cross section is denoted by σ_{gg}.[10] The particles that collide are the participants. We then define the **fluctuated thickness** for each proton by

$$T_A(x, y) := w_A \int \rho_A(x, y, z) dz, \tag{2.17}$$

where we took proton A without loss of generality and w_A is a weight coming from the Gamma distribution $\Gamma(k, k)$:

$$P_k(w_A) = \frac{k^k}{\Gamma(k)} w_A^{k-1} e^{-k w_A}. \tag{2.18}$$

This implies that it has a mean of $k/k = 1$. These weights produce multiplicity fluctuations as is observed in experiments. To this end, one can tweak k such that theory and experiment coincide. In practice, it often occurs that the system is actually a collision between nuclei, which contain more than one nucleon. We can produce a set of positions of the different nucleons, usually done by sampling from a distribution. One often uses the Woods-Saxon distribution [19]:

$$\rho(r) = \frac{\rho_0}{1 + \exp\left(\frac{r-R}{a}\right)}, \tag{2.19}$$

with ρ_0 the density at the centre of the nucleus, R the nuclear radius and a the skin depth. These values can be obtained experimentally from electron scattering experiments. Equation (2.17) can be modified to be valid for bigger systems, namely

$$T_A(x, y) = \sum_{i=1}^{N_{\text{part}}} (w_A)_i \int \rho_A(x - x_i, y - y_i, z - z_i) dz =: \sum_{i=1}^{N_{\text{part}}} (w_A)_i T_p(x - x_i, y - y_i), \tag{2.20}$$

with N_{part} the amount of participants and T_p the nucleon thickness function. We again chose A without loss of generality and these labels A and B now distinguish between the two colliding objects. The transverse profile of the nucleon shape is often chosen to be Gaussian shaped:

$$T_p(x, y) = \frac{1}{2\pi w^2} \exp\left(-\frac{x^2 + y^2}{2w^2}\right), \tag{2.21}$$

with w the nucleon width. Finally, we can define the **reduced thickness** T_R, which is the function we are actually looking for:

$$f = T_R(p, T_A, T_B) := \left(\frac{T_A^p + T_B^p}{2}\right)^{1/p}, \tag{2.22}$$

which depends on T_A and T_B, i.e. on the fluctuated thickness of both colliding nuclei. In conclusion, we have found a function that can bring us from the initial participants to an entropy density, which was our goal.

TRENTo is active right after the collision, say at $\tau_0 = 0^+$. However, it is still too early to let hydrodynamics do its thing. The system is not yet in local thermal equilibrium right after the collision. The stage between $\tau = \tau_0$ and the start of hydrodynamics is called the **pre-equilibrium** stage. It is not necessarily needed, as one could start with hydrodynamics right away [20]. However, one should probably change the initial conditions to compensate for any pre-equilibrium evolution. As there has to be enough time for the collective flow to build up, we want to start early with the hydrodynamic description of the system (namely at $\tau \sim 1$ fm). We therefore need this pre-equilibrium stage to get the system into equilibrium quickly. The simplest way to model this quick expansion is by introducing **free streaming**, of which the duration is indicated by τ_{fs}. This model assumes that the particles are noninteracting and massless. After the time τ_{fs}, the system will instantaneously go to a hydrodynamic description with strong coupling. This is of course not the most physical description and it can be improved by a more gradual build-up of the strength of the coupling or by having a weakly-coupled system in the beginning. However, for our intents and purposes, the free streaming works rather well. The TRENTo model returns an entropy density, but we want to consider particles, i.e. a particle density $n(x, y)$, to implement free streaming. Luckily, these densities have the same units and we can therefore think of these particles as carriers of some amount of entropy units. So this does not contradict what we just did.

[10] The cross section σ_{gg} is chosen in such a way that the total proton-proton cross section is the same as the inelastic nucleon-nucleon cross section σ_{NN}.

As the particles are assumed to be massless during free streaming, they move with the speed of light. They also move along straight trajectories as the particles are noninteracting. We can write down the energy-momentum tensor for a time $\tau > \tau_0$ [21]:

$$T^{\mu\nu}(x,y) = \frac{1}{\tau}\int_0^{2\pi} \hat{p}^\mu \hat{p}^\nu n(x - \Delta\tau\cos\varphi, y - \Delta\tau\sin\varphi)d\varphi, \tag{2.23}$$

with $\Delta\tau = \tau - \tau_0$ and $\hat{p}^\mu = p^\mu/p_t$. We assume **boost invariance** [22], so we can say that $\eta = 0$. As $y = \eta$ for massless particles, we see that $\hat{p}^\mu\hat{p}^\nu$ can be written as (recall Chapter 1)

$$\hat{p}^\mu\hat{p}^\nu = \begin{pmatrix} 1 & \cos\varphi & \sin\varphi & 0 \\ \cos\varphi & \cos^2\varphi & \cos\varphi\sin\varphi & 0 \\ \sin\varphi & \cos\varphi\sin\varphi & \sin^2\varphi & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.24}$$

At time $\tau = \tau_{\text{fs}}$, we have to match equation (2.23) to equation (2.10). We do a small computation where we use the results from section 2.1 that $u_\mu u^\mu = 1$ and that $\Delta^{\mu\nu}$ and $\pi^{\mu\nu}$ are transverse to the four-velocity, i.e. $A^{\mu\nu}u_\nu = 0$ for such a transverse tensor $A^{\mu\nu}$:

$$\begin{aligned} T^{\mu\nu}u_\nu &= \left(\epsilon u^\mu u^\nu - (p + \Pi)\,\Delta^{\mu\nu} + \pi^{\mu\nu}\right) u_\nu \\ &= \epsilon u^\mu. \end{aligned} \tag{2.25}$$

This is just a matter of solving for the eigenvalue and the eigenvector. Naturally, only physical solutions, i.e. timelike eigenvectors, are allowed to be solutions to this equation and we obtain ϵ and u^μ. After this, the pressure p can be calculated using the equation of state. Next, we can calculate the bulk viscosity Π:

$$\begin{aligned} \Delta_{\mu\nu}T^{\mu\nu} &= \epsilon\Delta_{\mu\nu}u^\mu u^\nu - (p+\Pi)\Delta_{\mu\nu}\Delta^{\mu\nu} + \Delta_{\mu\nu}\pi^{\mu\nu} \\ &= -3(p+\Pi), \end{aligned} \tag{2.26}$$

as $\pi^{\mu\nu}$ is traceless and transverse and $g_\mu{}^\mu = 4$. In other words, Π can be calculated by

$$\Pi = -\frac{1}{3}\Delta_{\mu\nu}T^{\mu\nu} - p. \tag{2.27}$$

Finally, we can calculate the shear viscosity tensor $\pi^{\mu\nu}$ by

$$\pi^{\mu\nu} = T^{\mu\nu} - \epsilon u^\mu u^\nu + (p+\Pi)\Delta^{\mu\nu}. \tag{2.28}$$

Now that we have all information about the initial conditions for the energy-momentum tensor, we can use the hydrodynamic description of the system. For this, there are still some things left to be modelled. For the shear and bulk viscosity as function of the temperature T, the following ansatzes are made:

$$\left(\frac{\eta}{s}\right)(T) = (\eta/s)_{\text{min}} + (\eta/s)_{\text{slope}}\,(T - T_c)\left(\frac{T}{T_c}\right)^{(\eta/s)_{\text{crv}}}, \tag{2.29}$$

$$\left(\frac{\zeta}{s}\right)(T) = \frac{(\zeta/s)_{\text{max}}}{1 + \left(\frac{T-(\zeta/s)_{T_0}}{(\zeta/s)_{\text{width}}}\right)^2}, \tag{2.30}$$

where the different coefficients are parameters of the simulation and T_c is the transition temperature from the plasma to the hadron resonance gas. For this gas, there is a different parameter, namely $(\eta/s)_{\text{hrg}}$. This is a constant that describes the shear viscosity in the hadronic phase while describing the system hydrodynamically.

Due to numerical inaccuracy that can occur and grow over time, it can be difficult to require that $\pi^\mu{}_\mu = 0 = \pi^{\mu\nu}u_\nu$. We therefore choose a numerical 'zero' $\xi_0 \ll 1$, such that

$$\begin{aligned} \pi^\mu{}_\mu &\leq \xi_0\sqrt{\pi^{\mu\nu}\pi_{\mu\nu}}, \\ \pi^{\mu\nu}u_\nu &\leq \xi_0\sqrt{\pi^{\mu\nu}\pi_{\mu\nu}}, \end{aligned} \tag{2.31}$$

where $\sqrt{\pi^{\mu\nu}\pi_{\mu\nu}} = \text{Tr}\left(\pi^2\right)$ is used as a measure for the magnitude of the components of $\pi^{\mu\nu}$. There is another requirement for this shear viscosity tensor, namely that it is smaller than the ideal part of the energy-momentum tensor $T_0^{\mu\nu}$. The reason for this is driven by consistency in the theoretical framework [23]. Numerically, the following is required:

$$\pi^{\mu\nu}\pi_{\mu\nu} \ll T_0^{\mu\nu}T_{0\mu\nu} = \epsilon^2 + 3p^2. \tag{2.32}$$

To make this happen, we define the parameter $\rho_{\max} \ll 1$ such that

$$\sqrt{\pi^{\mu\nu}\pi_{\mu\nu}} \leq \rho_{\max}\sqrt{\epsilon^2 + 3p^2}. \tag{2.33}$$

The condition of equation (2.33) can sometimes be violated. There is a systematic treatment against these large viscous terms that stabilises the code, but does not affect the important physics [23]. It can be seen as making local sharp jumps in the flow profile smoother and is done by

$$\pi^{\mu\nu} \mapsto \hat{\pi}^{\mu\nu} := \pi^{\mu\nu}\frac{\tanh\rho}{\rho}, \tag{2.34}$$

with

$$\rho := \max\left\{\frac{\sqrt{\pi^{\mu\nu}\pi_{\mu\nu}}}{\rho_{\max}\sqrt{\epsilon^2+3p^2}},\ \frac{\pi^{\mu}{}_{\mu}}{\xi_0\rho_{\max}\sqrt{\pi^{\mu\nu}\pi_{\mu\nu}}},\ \frac{\pi^{\mu\nu}u_{\nu}}{\xi_0\rho_{\max}\sqrt{\pi^{\mu\nu}\pi_{\mu\nu}}}\right\}.$$

Normally, ρ will be quite small and then $\tanh\rho/\rho$ will be very close to 1 and we have our normal system. However, if it happens that ρ becomes large (because conditions (2.31) and (2.33) are not fulfilled), ρ can become larger than 1 and the viscous tensor will be suppressed due to $\tanh\rho/\rho < 1$.

2.3 Particlisation and afterburner

Over time, the temperature and density of the system decrease due to the expansion. There will be a point in time after which hydrodynamics is not the best way anymore to describe the system. At low temperatures, it is best described via Boltzmann transport models. For this to happen, the continuous hydrodynamic system has to be converted to a discrete system of individual particles. This process is called **particlisation** and is performed at a temperature T_{switch}. This temperature has to be chosen such that both descriptions are equally valid and give the same result in that regime. It defines a hypersurface σ in four dimensional spacetime. The process of particlisation of species s is governed by the Cooper-Frye formula [24]:

$$E\frac{d^3N_s}{d^3p} = \frac{g_s}{(2\pi)^3}\int_{\sigma} f_s(\vec{x},p)p^{\mu}d^3\sigma_{\mu}, \tag{2.35}$$

where the amount of particles of species s is indicated by N_s and g_s is the degeneracy of that species, d^3p is an abbreviation for $d^3p = dp_x dp_y dp_z$, $f_s(\vec{x},p)$ is the grand-canonical boosted particle distribution and $d\sigma_{\mu}$ is the normal to σ. This formula gives the particle distribution of species s in momentum space. If the system is in equilibrium, one would have for the particle distribution function that

$$f_{\text{eq}}(\vec{x},p) = \frac{1}{\exp\left[p_{\mu}u^{\mu}(\vec{x})/T(\vec{x})\right] \pm 1}, \tag{2.36}$$

where the $+1$ belongs to the Fermi-Dirac distribution (i.e. for fermions) and the -1 belongs to the Bose-Einstein distribution (i.e. for bosons). The chemical potential μ is set to zero in the simulations. A system that is described by viscous hydrodynamics is a bit out of equilibrium and the particle distribution is therefore modified by adding some δf to the equilibrium distribution. This has to be done to make sure that $T^{\mu\nu}$ is continuous across the transition of system description. What this looks like is not well known, although there exist some ansatzes [25]. If $p^{\mu}d\sigma_{\mu} < 0$, it means that the particle travels backwards in the fluid. This effect is difficult to model, and as it only applies for a small percentage of the total amount of particles, this effect is neglected and corresponding particles are discarded.

The particle distribution function is not only modified due to non-equilibrium effects. Another important aspect is the **resonance width** of particles. Instead of having a mass m_0, the pole mass, the particle can have a different mass, while still being the same particle. These resonance particles are in an excited state and have short half lives. Normally for a density of particles with mass m_0, one would have

$$n = \frac{g}{(2\pi)^3}\int f(m_0,p)d^3p, \tag{2.37}$$

with $f(m_0,p) = \left(\exp\left[\sqrt{m_0^2+p^2}/T\right] \pm 1\right)^{-1}$. If, however, this energy can vary via the mass by a probability distribution $\mathcal{P}(m)$, we obtain a different particle distribution, namely

$$f(p) = \int \mathcal{P}(m)f(m,p)dm = \int \frac{\mathcal{P}(m)}{\exp\left(\sqrt{m^2+p^2}/T\right) \pm 1}dm. \tag{2.38}$$

The distribution $\mathcal{P}(m)$ can be assumed to take the form

$$\mathcal{P}(m) \propto \frac{\Gamma(m)}{(m-m_0)^2+\Gamma(m)^2/4}, \qquad \text{with } \Gamma(m) := \Gamma_0\sqrt{\frac{m-m_{\min}}{m_0-m_{\min}}}, \tag{2.39}$$

where $m_{\min}$ is a threshold mass: the total mass of the lightest decay products. Naturally, $\mathcal{P}(m)$ is normalised to 1 for $m_{\min} < m < m_{\max} := m_0 + 4\Gamma_0$.

Equation (2.35) gives a probability distribution of the momentum. To produce an ensemble of discrete particles, one can sample those particles from the distribution. When this whole process is done, the hadronic system is described by a Boltzmann transport model. This simulates all the dynamics, such as decay or scattering. It solves the Boltzmann equation

$$\frac{df_i(x,p)}{dt} = \mathcal{C}_i(x,p), \tag{2.40}$$

where f_i is the distribution function of species i and $\mathcal{C}_i$ is the source term, which takes into account the collisions involving species i. The model that is used as **afterburner** for our purposes is UrQMD (Ultra-relativistic Quantum Molecular Dynamics) [26, 27]. It solves equation (2.40) by propagating the particles along straight-line trajectories and samples the collisions by using stochastics. The full relativistic treatment is a bit too involved to show here, as all particles have a different internal clock. This causes the relative distances between particles to be different. One would have to construct a relative distance that is Lorentz invariant, which can be done by some Minkowski inner product. Here, a somewhat intuitive reasoning will be given. The particles are represented by a Gaussian shaped density distribution. All these particles induce a potential and interact with each other. Due to electric charge, there is a Coulomb potential. There is also a Skyrme potential that models the binding and saturation and a Yukawa potential that takes the strong nuclear force into account. A collision takes place if

$$d_{\text{trans}} \leq \sqrt{\frac{\sigma_{\text{tot}}}{\pi}}, \tag{2.41}$$

with σ_{tot} the total cross section, interpreted geometrically as an area. The decay channels for the different particles are known and can be used to this extent. When the chemical interactions, such as decays, are done, **chemical freeze-out** is reached: the particles are stable.[11] Some time after that, the particles will not interact with each other anymore, which means that **kinetic freeze-out** has been reached. Again some time later, the particles will hit a detector, and we can start to analyse the detections, which will be done in the next chapter.

[11] Stable in the sense that the lifetime of the hadrons is long enough to reach the detectors.

Chapter 3

Analysis of heavy-ion collisions

In the previous chapter, we have developed a theory to describe the whole process of a heavy-ion collision and the development of the system until the final particles are detected. In this chapter, we will discuss how one can deduce important physical information of the system from particle detections. It will be important to quantify the collective behaviour that was shortly mentioned in the introduction and the observed anisotropy in the transverse momentum distribution. This will be the first topic. Then we will discuss the Symmetric Cumulants, which is an observable that will be explored in great detail in this thesis. The last section is dedicated to the question how one can obtain information about the initial conditions of the system, such as the shear viscosity η/s for the quark-gluon plasma, by using particle detectors.

3.1 Anisotropy

If two (elementary) particles collide, the momentum distribution in the transverse plane will be **isotropic**, i.e. uniformly distributed in this plane. We could also look at heavy-ion collisions. The colliding particles are composed of a number of nucleons. If all collisions are independent, we will just have a superposition of the individual collisions and the momentum distribution will still be isotropic in the transverse plane. There is another case, which we saw in the previous chapter: the system can be in thermal equilibrium. Then the collisions are not independent [28]. Instead, particles can have an influence on each other, causing an anisotropy in the transverse momentum distribution induced by the initial geometry. The (an)isotropy of the initial geometry is characterised by

$$\epsilon_x(b) := \frac{\langle y^2 - x^2 \rangle}{\langle y^2 + x^2 \rangle}, \tag{3.1}$$

where $\langle \cdot \rangle$ denotes an average over the participants. Sometimes, the initial geometry is characterised by the different eccentricity coefficients

$$\epsilon_n e^{in\Phi_n} := -\frac{\langle r^n e^{in\varphi} \rangle}{r^n}, \tag{3.2}$$

with Φ_n the initial symmetry plane angle.

The anisotropy of the initial geometry can be translated to the transverse momentum distribution via the pressure gradient. This initial geometry and the gradient is illustrated in figure 1.1 and figure 3.1. The latter also shows how the plasma transforms over time. The gradient in the x-direction is greater than the gradient in the y-direction, because the distance between the centre of the plasma and the surrounding vacuum is smaller. This causes the plasma to propagate quicker in the x-direction than in the y-direction.

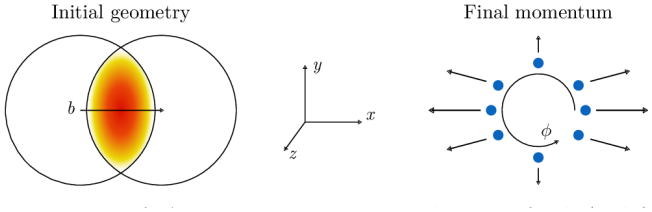


Figure 3.1: Illustration of the overlap region with impact parameter b in the transverse plane and the expansion of the QGP over time [6].

The consequence of this kind of expansion is an anisotropic azimuthal transverse momentum distribution, which essentially is given by the transformation of coordinates of equation (1.9) [29]:

$$E\frac{d^3N}{d^3p} = \frac{d^2N}{2\pi p_t dp_t dy}\left(1 + 2\sum_{n=1}^{\infty} v_n \cos(n(\varphi - \Psi_R))\right), \tag{3.3}$$

where N is the amount of particles. The other symbols have already been explained in Chapter 1. A full derivation of this Fourier series with coefficients v_n is done in Appendix A. The right hand side of equation (3.3) can be viewed as the amount of particles N binned by p_t per unit rapidity y. The Fourier coefficients, also called the harmonics or the **flow coefficients**, of this Fourier series expansion (3.3) can be calculated in the following way:

$$v_n(p_t, y) = \langle \cos(n(\varphi - \Psi_R)) \rangle_p, \tag{3.4}$$

where $\langle\cdot\rangle_p$ denotes an average over all detected particles in an event. The first harmonic is also called the directed flow. It is basically an overall shift of the distribution. The second harmonic is the most important one and it is called the **elliptic flow**.

3.2 Cumulants

In principle, the reaction plane angle Ψ_R is not known, so we cannot compute v_n by equation (3.4). There is a way to estimate this reaction plane in a somewhat intuitive way, namely by the usual way to compute an angle of a position (x, y) with respect to the x-axis: $\theta = \arctan(y/x)$. We can estimate Ψ_R depending on the n^{th} harmonic by [30]

$$\Psi_R(n) \approx \Psi_n := \frac{1}{n}\arctan\left(\frac{\sum_{i=1}^{M}\sin(n\varphi_i)}{\sum_{i=1}^{M}\cos(n\varphi_i)}\right), \tag{3.5}$$

with M the number of detected particles in an event, and φ_j the azimuthal angle of particle j.

A way to avoid this approximation is using azimuthal correlations. In this way, we do not need the approximation in equation (3.5). To develop this theory, we first define a **flow vector**[12] $Q_n \in \mathbb{C}$ as

$$Q_n := \sum_{j=1}^{M} e^{in\varphi_j}. \tag{3.6}$$

This definition of Q_n also has an immediate generalisation by using weights, i.e.

$$Q_n := \sum_{j=1}^{M} w_j^k e^{in\varphi_j}, \tag{3.7}$$

with w_j the weight of particle j and $k \geq 0$. In light of this complex way of expressing the flow vector, we can say that

$$v_n(p_t, y) = \left\langle e^{in(\varphi - \Psi_R)} \right\rangle_p. \tag{3.8}$$

We can indeed argue, by symmetry of particle production w.r.t. the reaction plane, that the imaginary part of equation (3.8) has to be zero, as $\sin(\cdot)$ is an odd function. So this way of writing v_n is equivalent with equation (3.4).

From now on, we will omit the subscript in $\langle\cdot\rangle_p$. If not indicated otherwise, we mean by single brackets an average over the particles in an event and double brackets indicate an additional average over all events, with the appropriate weights as defined in the next section.

3.2.a Cumulant method

In practice, we want to quantify the (anisotropic) flow of a QGP by calculating the flow coefficients. However, we also have so-called **nonflow** in the system.[13] Anisotropic flow is the result of correlations between all particles in the fluid, because it is in (approximate) equilibrium. On the other hand, nonflow is a result of the correlation between only a few particles. The decay of a particle could be an example, as this involves only a few particles.

[12] It is a vector in the sense that it has two components (real and imaginary part). It is just standard terminology and is adopted here.

[13] The program does not simulate nonflow coming from jets, which is the main part of nonflow. However, we use the described method anyway, because the simulation has to do the exact same thing as is done in experiments.

Another example would be the formation of jets with a back-to-back topology. This nonflow causes a bias in the flow we want to measure. To this end, **multi-particle correlation** methods have been developed. The more particles we take into account at once, the less bias we have from nonflow in the flow estimate. We will denote this multi-particle correlation by $v_n\{k\}$, with $k \in \mathbb{N}$. So $v_n\{k\}$ is based on the correlation between k particles. The strength of this method lies in the fact that a higher order correlation removes all lower order correlations. After some important definitions, we will argue this result. Due to this fact, the effects of nonflow can be greatly reduced. Because a big part of nonflow is due to processes that involve only a few particles, we already largely reduce the bias of nonflow if we use 4-particle correlations. For explanatory purposes, we only look at 2- and 4-particle correlations. The result for higher order correlations is a straightforward generalisation. First, we define a single-event average 2- and 4-particle correlation as [31]

$$\langle 2 \rangle := \left\langle e^{in(\varphi_1 - \varphi_2)} \right\rangle = \frac{1}{P_{M,2}} \sum_{\substack{i,j=1 \\ i \neq j}}^{M} e^{in(\varphi_i - \varphi_j)}, \tag{3.9}$$

$$\langle 4 \rangle := \left\langle e^{in(\varphi_1 + \varphi_2 - \varphi_3 - \varphi_4)} \right\rangle = \frac{1}{P_{M,4}} \sum_{\substack{i,j,k,l=1 \\ i \neq j \neq k \neq l}}^{M} e^{in(\varphi_i + \varphi_j - \varphi_k - \varphi_l)}, \tag{3.10}$$

where $P_{M,k} = \frac{M!}{(M-k)!}$. We can continue to average this single-event average over all events:

$$\langle\langle 2 \rangle\rangle = \left\langle \left\langle e^{in(\varphi_1 - \varphi_2)} \right\rangle \right\rangle = \frac{\sum_{i=1}^{N} (W_{\langle 2 \rangle})_i \langle 2 \rangle_i}{\sum_{i=1}^{N} (W_{\langle 2 \rangle})_i}, \tag{3.11}$$

$$\langle\langle 4 \rangle\rangle = \left\langle \left\langle e^{in(\varphi_1 + \varphi_2 - \varphi_3 - \varphi_4)} \right\rangle \right\rangle = \frac{\sum_{i=1}^{N} (W_{\langle 4 \rangle})_i \langle 4 \rangle_i}{\sum_{i=1}^{N} (W_{\langle 4 \rangle})_i}, \tag{3.12}$$

with N the amount of events. In these formulas, $W_{\langle k \rangle}$ denotes the **event weight**. They are used to minimise the statistical spread. This is achieved by [29]

$$W_{\langle 2 \rangle} := M(M-1), \tag{3.13}$$

$$W_{\langle 4 \rangle} := M(M-1)(M-2)(M-3). \tag{3.14}$$

When one uses non-unit weights, these weights generalise to

$$\mathcal{M}_{abcd\ldots} := \sum_{\substack{i,j,k,l=1 \\ i \neq j \neq k \neq l}}^{M} w_i^a w_j^b w_k^c w_l^d \cdots. \tag{3.15}$$

For unit weights, we indeed get back equations (3.13) and (3.14).

In the end, we want to have a reliable formula for v_n. The formulas above for the k-particle correlation will help us. First, we need the concept of a genuine k-particle correlation, which we will call the **k-particle cumulant**. These are defined by $c_n\{2k\}$. There is a short derivation of these cumulants in Appendix B. As shown in [32], we can express $c_n\{2k\}$ in terms of the averages defined above:

$$c_n\{2\} = \langle\langle 2 \rangle\rangle, \tag{3.16}$$

$$c_n\{4\} = \langle\langle 4 \rangle\rangle - 2 \langle\langle 2 \rangle\rangle^2. \tag{3.17}$$

Moreover, we have that the flow coefficients are related to these cumulants, as shown in Appendix B:

$$v_n\{2\} = \sqrt{c_n\{2\}}, \tag{3.18}$$

$$v_n\{4\} = \sqrt[4]{-c_n\{4\}}. \tag{3.19}$$

If there would be no nonflow and no statistical fluctuations, we would get back what one would expect:

$$v_n\{2\} = \sqrt{\left\langle \left\langle e^{in(\varphi_1 - \varphi_2)} \right\rangle \right\rangle} = \sqrt{\left\langle \left\langle e^{in[(\varphi_1 - \Psi_R) - (\varphi_2 - \Psi_R)]} \right\rangle \right\rangle} = \sqrt{\left\langle \left\langle e^{in(\varphi_1 - \Psi_R)} \right\rangle \left\langle e^{-in(\varphi_2 - \Psi_R)} \right\rangle \right\rangle} = v_n, \tag{3.20}$$

and similarly for the higher order correlations.

We can now argue that higher order correlations remove the lower order ones. The form of equation (3.17) already suggests that there is a subtraction going on. We will give a demonstration for the second harmonic. Assume that we have in each event $M/2$ pairs of particles and the particles of each pair have collinear momenta. These pairs are emitted with a random orientation, such that there is no flow. In total, there are $M(M-1)/2$ pairs, of which $M/2$ are correlated. This results in $\langle 2 \rangle = 1/(M-1)$. As each event has the same amount of particles by assumption, we also have that $c_2\{2\} = \langle\langle 2 \rangle\rangle = 1/(M-1)$. By equation (3.18), we have that $v_2\{2\} \sim 1/\sqrt{M}$. This was under the assumption that there is no flow, but only azimuthal correlations. This also shows that nonflow cannot be neglected, because $v_2 \neq 0$ even though we assumed that there is no flow. Similarly, we have that $c_2\{4\}$ only has a nonflow contribution of $1/M^3$, while $\langle\langle 4 \rangle\rangle$ and $\langle\langle 2 \rangle\rangle$ have a contribution of $1/M^2$ and $1/M$ respectively [33]. This indeed shows that $c_2\{4\}$ removes lower order correlations and suppresses nonflow.

Figure 3.2 shows how the most important flow harmonic, i.e. the elliptic flow, depends on the centrality. For small centrality, there is much overlap between the colliding nuclei, so there is not a good initial elliptic geometry. This results in a rather small anisotropy. This anisotropy increases as the centrality increases, because the initial geometry becomes more suited to develop a nice anisotropy. However, for even larger centrality, the elliptic flow decreases again, because the QGP medium does not survive long enough for these peripheral collisions and thermal equilibrium cannot be established. Besides, such a peripheral collision is nothing more than a collision between two nucleons and will therefore have an isotropic transverse momentum distribution. The figure also shows that there is a notable difference between the different order correlations. Larger order correlations tend to decrease the total elliptic flow. This is indeed expected, because nonflow is being suppressed in this method.

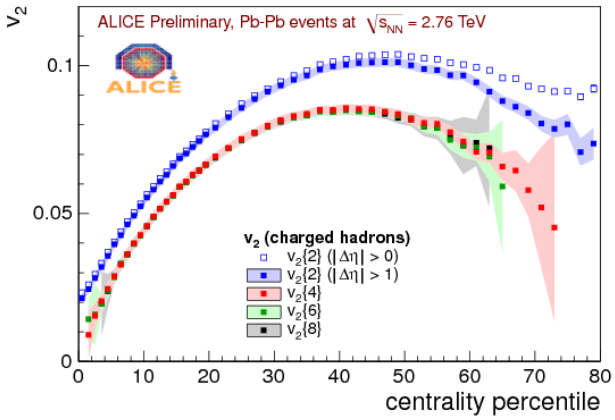


Figure 3.2: Elliptic flow measured for lead-lead collisions at $\sqrt{s_{NN}} = 2.76$ TeV as a function of centrality [34].

So far, we have not used our knowledge about the flow vector. This can be done as follows:

$$|Q_n|^2 = Q_n Q_n^* = \sum_{i,j=1}^{M} e^{in(\varphi_i - \varphi_j)} = M + \sum_{\substack{i,j=1 \\ i \neq j}}^{M} e^{in(\varphi_i - \varphi_j)} = M + P_{M,2} \langle 2 \rangle . \tag{3.21}$$

Rearranging, this yields

$$\langle 2 \rangle = \frac{|Q_n|^2 - M}{M(M-1)}. \tag{3.22}$$

So by only using the flow vector, we can completely determine $v_n\{2\}$ using relations (3.11) and (3.18). Similarly, we can calculate $\langle 4 \rangle$ using the flow vector [29]:

$$\langle 4 \rangle = \frac{|Q_n|^4 + |Q_{2n}|^2 - 2\,\mathrm{Re}[Q_{2n} Q_n^* Q_n^*] - 4(M-2)|Q_n|^2 + 2M(M-3)}{M(M-1)(M-2)(M-3)}. \tag{3.23}$$

3.2.b Symmetric Cumulants

An observable that will be of great use for this research is the **Symmetric Cumulant**, which is defined, for $m \neq n$, as [6, 35]

$$\mathrm{SC}(m,n) := \langle\langle \cos(m(\varphi_1 - \varphi_3) + n(\varphi_2 - \varphi_4)) \rangle\rangle - \langle\langle \cos(m(\varphi_1 - \varphi_2)) \rangle\rangle \langle\langle \cos(n(\varphi_1 - \varphi_2)) \rangle\rangle \tag{3.24}$$

$$\approx \langle v_m^2 v_n^2 \rangle - \langle v_m^2 \rangle \langle v_n^2 \rangle , \tag{3.25}$$

where the approximate sign is replaced by an equal sign if there are no nonflow effects. The terms in the definition of $\mathrm{SC}(m,n)$ can be computed via the flow vector:

$$\langle\cos(n(\varphi_1-\varphi_2))\rangle = \frac{1}{P_{M,2}}\left(|Q_n|^2 - M\right), \tag{3.26}$$

$$\begin{aligned}\langle\cos(m(\varphi_1-\varphi_3)+n(\varphi_2-\varphi_4))\rangle = \frac{1}{P_{M,4}}\Big[&|Q_m|^2|Q_m|^2 - 2\,\mathrm{Re}(Q_{m+n}Q_m^*Q_n^*) - 2\,\mathrm{Re}(Q_mQ_{m-n}^*Q_n^*)\\ &+ |Q_{m+n}|^2 + |Q_{m-n}|^2 - (M-4)\left(|Q_m|^2+|Q_n|^2\right) + M(M-6)\Big].\end{aligned} \tag{3.27}$$

The approximate formula for $\mathrm{SC}(m,n)$ in equation (3.25) helps to get a feeling for these Symmetric Cumulants. Namely, if we look at the definition of the **covariance** of two stochastic variables X and Y, then this is defined as [36]

$$\mathrm{Cov}(X,Y) := \langle XY\rangle - \langle X\rangle\langle Y\rangle. \tag{3.28}$$

So the Symmetric Cumulant is nothing more than a measure of how v_n^2 and v_m^2 are correlated. Analogously with the correlation coefficient ρ in statistics, we can define the normalised Symmetric Cumulant $\mathrm{sc}(m,n)$ by

$$\mathrm{sc}(m,n) := \frac{SC(m,n)}{\langle\langle\cos(m(\varphi_1-\varphi_2))\rangle\rangle\langle\langle\cos(n(\varphi_1-\varphi_2))\rangle\rangle} \tag{3.29}$$

$$\approx \frac{\langle v_m^2v_n^2\rangle - \langle v_m^2\rangle\langle v_n^2\rangle}{\langle v_m^2\rangle\langle v_n^2\rangle}. \tag{3.30}$$

Naturally, we want to have the most precise results. That is why only equation (3.24) and (3.29) are used. They take into account the nonflow contributions, which are of relevance and cannot be neglected as argued before. This observable is implemented in the simulation program and will be investigated in great detail in Chapter 4.

3.3 Bayesian analysis

In the end, the research about the quark-gluon plasma is done to better understand what is going on and how it behaves. We can understand this phase better by knowing the parameters of the system, which give information about e.g. the shear viscosity. Bayesian statistics can help determine these. It uses experimental data to improve the probability distribution of the different parameters. We therefore do not think about parameters as fixed values, but rather as a probability distribution. The set of parameters is varied, which makes the analysis a time-consuming one, because one would have to run the whole simulation for each set of parameters. Machine learning is used to optimise this process, but it still requires a lot of computation time. Such an analysis is done using a supercomputer.

Let us denote the set of parameters we want to know by $x = (x_1, x_2, \ldots, x_n)$ and the experimental data by y. We can then define a prior distribution of x, namely $\mathcal{P}(x)$. This is based on prior knowledge about the parameters, e.g. some parameters cannot be negative. We also define the conditional probability $\mathcal{P}(y \mid x)$, which tells something about the quality of the fit to experimental data, given the initial parameters x. Of interest is then $\mathcal{P}(x \mid y)$, i.e. the posterior distribution of the parameters given the experimental data. These quantities are linked via **Bayes' theorem**:

$$\mathcal{P}(x \mid y) \propto \mathcal{P}(y \mid x)\mathcal{P}(x). \tag{3.31}$$

We can go back to the distribution of one specific variable by

$$\mathcal{P}(x_i \mid y) = \int \mathcal{P}(x \mid y)dx_1\cdots\widehat{dx_i}\cdots dx_n, \tag{3.32}$$

where $\widehat{dx_i}$ means that we omit integration with respect to x_i. Normally, the posterior distribution will be a better guess of the distribution of the parameters than the prior distribution. From there, one can deduce a best guess that can act as the value of the parameter. Besides the experimental data, we also want to use model data, i.e. data from simulations. Let $\mathcal{D}$ denote the collection of experimental and model data. We can then replace y by $\mathcal{D}$ in equation (3.31). Now $\mathcal{P}(\mathcal{D} \mid x)$ quantifies the compatibility of the model calculations, performed with a particular set of parameters, with the experimental data. Let y_e be the vector of experimental data, which comes from measuring the hypothetical true values with measurement errors. These errors are of statistical and systematic nature. Similarly, one can define a vector of modelled data, with uncertainties that come from e.g. discretising a continuous system. It is a deviation from modelled data with unlimited precision. These vectors can be written as

$$y_e = y_e^{\mathrm{true}} + \epsilon_e, \qquad \text{with} \quad \epsilon_e \sim \mathcal{N}(0,\Sigma_e) \tag{3.33}$$

$$y_m(x) = y_m^{\mathrm{ideal}}(x) + \epsilon_m, \qquad \text{with} \quad \epsilon_m \sim \mathcal{N}(0,\Sigma_m), \tag{3.34}$$

where Σ_e and Σ_m are covariance matrices. We now assume that there exists a $x_\star$, such that the model data match the experimental data, i.e. $y_e^{\text{true}} = y_m^{\text{ideal}}(x_\star)$, then

$$y_e = y_m(x_\star) + \epsilon, \qquad \text{with} \quad \epsilon \sim \mathcal{N}(0, \Sigma), \tag{3.35}$$

where $\Sigma = \Sigma_e + \Sigma_m$, as the difference of two normal distributions is again a normal distribution with modified mean and variance. Relation (3.35) implies that we can write $\mathcal{P}(\mathcal{D} \mid x)$ as [6]

$$\mathcal{P}(\mathcal{D} \mid x) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left[-\frac{1}{2}\left(y_m(x) - y_e\right)^T \Sigma^{-1} \left(y_m(x) - y_e\right)\right]. \tag{3.36}$$

This procedure works well if we compare simulations to experimental data. However, we will see in the next chapter that there is not always experimental data available. A way to work around this is by guessing these data points. For example, the Symmetric Cumulants can be calculated for different sets of parameters. These parameters have to be compatible with the outcome of a Bayesian analysis that is done with experimental data that is available. This produces a band of possible outcomes when we plot the Symmetric Cumulants. An educated guess can then be made for the experimental data. If the band is narrow, it means that the observable is hardly sensitive for a change of parameters.[14] The observable will probably not constrain the parameters very well then. If there is a large band on the other hand, it indicates that the observable is sensitive to varying the parameters. Experimentalists could then consider to actually perform these collisions to acquire real experimental data for a better analysis that can actually constrain the parameters.

This finishes the discussion about the Bayesian analysis. Although we have a posterior distribution that is only proportional to a product of distributions, it suffices. With just the relative distributions, one can extract enough information about the (range of the) most likely value, which is eventually what is desired.

[14] Narrow is quite an arbitrary notion. One could say that a band is narrow if the size is comparable to the error bars.

Chapter 4

Results

Now that we have established the theory of simulating heavy-ion collisions and what one can do with the information that comes from particle detections, we can do some actual simulations. The main goal will be to determine the Symmetric Cumulants, as discussed in section 3.2.b. For starters, it is important that we can match results of a simulation with actual experimental results that already exist. It will be a good indicator whether the code is correctly written. The code is shown in Appendix E. If that seems to be the case, we continue with actual research. We will investigate the behaviour of adding a **substructure** to the nucleons. This means that the nucleons are not seen as one energy blob. It is instead taken into account that the nucleons consist of smaller constituents, with all an energy distribution of their own. Because of the large nucleon-nucleon centre of mass energy, the amount of constituents is modelled to be larger than the regular three (e.g. a proton consists of two up quarks and a down quark), as there is a cloud of virtual particles in the nucleon that interacts with the constituents. It is expected that the results from proton-lead collisions are sensitive to this additional structure, whereas results from lead-lead collisions are hardly influenced due to the large size of the colliding nuclei. Unless explicitly stated otherwise, we always take substructure into account. All the simulations are done at a centre of mass energy $\sqrt{s_{NN}} = 5.02$ TeV.

Later, other kinds of heavy-ion collisions are investigated, such as xenon-xenon and oxygen-oxygen collisions. With these results at hand, we have calculated the Symmetric Cumulants for different system sizes. As the Symmetric Cumulants are not yet investigated experimentally for XeXe and OO collisions, there will be no comparisons with experiments, but rather an estimation of what it would look like. We look at different system sizes, as some phenomena could be less pronounced in some systems that do appear in other kinds of systems.

It is good to note that there is not much experimental data available for the Symmetric Cumulants. Comparisons are sometimes done not under exactly the same conditions, so a translation between experimental and modelled data should be made. It is the hope that the Symmetric Cumulants are useful to constrain the initial conditions of the model. Constraining can be done by doing a Bayesian analysis, as discussed in section 3.3. If, from the theoretical point of view, there is enough evidence that the Symmetric Cumulants constrain further the initial conditions as we examine different sizes of systems, experiments can be done to investigate this in further detail. It is therefore important that the simulations are done accurately, in order to judge whether or not to do experiments with for example OO-collisions. As oxygen is a light nucleus, one would acquire a lot of new data of a different regime than lead. The used initial conditions can be found in Appendix D.

4.1 Lead-lead collisions

We first want to check whether experimental data can be reproduced. This is done by a simulation of 10,000 PbPb collisions. The results of $\mathrm{SC}(4,2)$ and $\mathrm{SC}(3,2)$ will be compared to data from the ALICE experiment at the LHC, with $\sqrt{s_{NN}} = 2.76$ TeV [35]. The result is demonstrated in figure 4.1. For this, we used that $|\eta| < 2.4$ and $200 < p_t < 5000$ GeV, similar to the experimental data.[15] First of all, the order of magnitudes are the same. Next, the trend of experimental and modelled data are comparable; $\mathrm{SC}(3,2)$ is negative and $\mathrm{SC}(4,2)$ is positive. The harmonics v_2 and v_3 are driven by the initial geometry, respectively ϵ_2 and ϵ_3. As ϵ_2 and ϵ_3 are anti-correlated, this results in a negative $\mathrm{SC}(3,2)$. Meanwhile, v_4 is driven by v_2, so $\mathrm{SC}(4,2)$ is positive [37]. The curves have similar behaviour if we take into account that the centre of mass energy is different. As the simulation is done with a larger centre of mass energy than in the experiment, it makes sense that the deviation from zero is different. The standard deviation of the simulation is quite a bit larger than from experiments, as much more data is acquired in experiments. It takes too long for practical reasons to gather the same statistics

[15] The experiment used a smaller eta range ($|\eta| < 0.8$), but as we assume the system to be approximately boost invariant, the eta range is made a bit larger. This is to gain more statistics.

in a simulation. From this analysis, we can conclude that the code is functional and accurate, as it gives results that are consistent with experiments.

We observe that the error bars for the Symmetric Cumulants grow for higher centrality bins. This is because there are less particles per event in these bins as the collisions become more peripheral. So there is less statistics available in the high centrality bins. We could introduce a bias in the generated events, such that more peripheral collisions are simulated more often. This is however not yet available in the current simulation program.

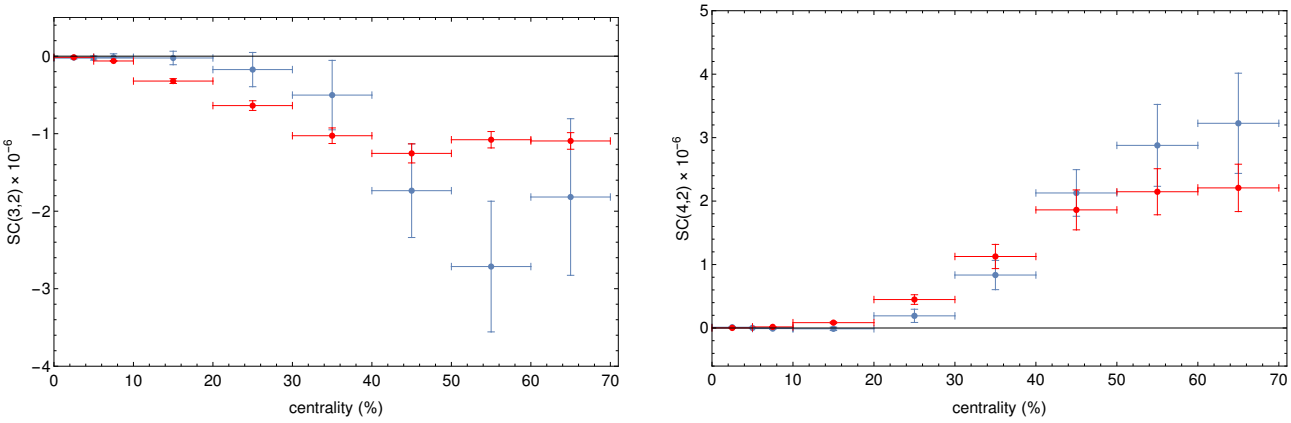


Figure 4.1: SC(3, 2) (left) and SC(4, 2) (right) as a function of centrality in 5.02 TeV lead-lead collisions. The blue data points come from the simulation and the red ones from experiments with $\sqrt{s_{NN}} = 2.76$ TeV [35].

4.2 Relevance of substructure

Each nucleon consists of three quarks. We can choose to model a nucleus as a bunch of nucleons without underlying substructure. We then have a Gaussian shaped energy distribution for each nucleon and the nucleus consists of some superposition of all these energy distributions. Although this is not the most realistic case, it is easier to implement in the program than taking into account a substructure for each nucleon. For large systems, such as PbPb, adding a substructure will likely not change the results significantly. However for smaller systems, such as a proton that collides with a lead nucleus, a substructure could make a significant difference.

Figure 4.2 shows the results for SC(3, 2) and SC(4, 2) as a function of $N_{\text{trk}}^{\text{off}}$ in pPb collisions, where we have not taken any substructure into account. We have done 100,000 simulations, where $N_{\text{trk}}^{\text{off}}$ is determined by using $0.4 < p_t < 10,000$ GeV and $|\eta| < 2.4$ as cuts.[16] For the calculation of the Symmetric Cumulants, the used cuts are $0.3 < p_t < 3000$ GeV and $|\eta| < 2.4$. The red data points come from experimental data [38] and one immediately sees that simulation and experiment do not coincide. This is a sign that neglecting the substructure of nucleons is not a reasonable assumption.

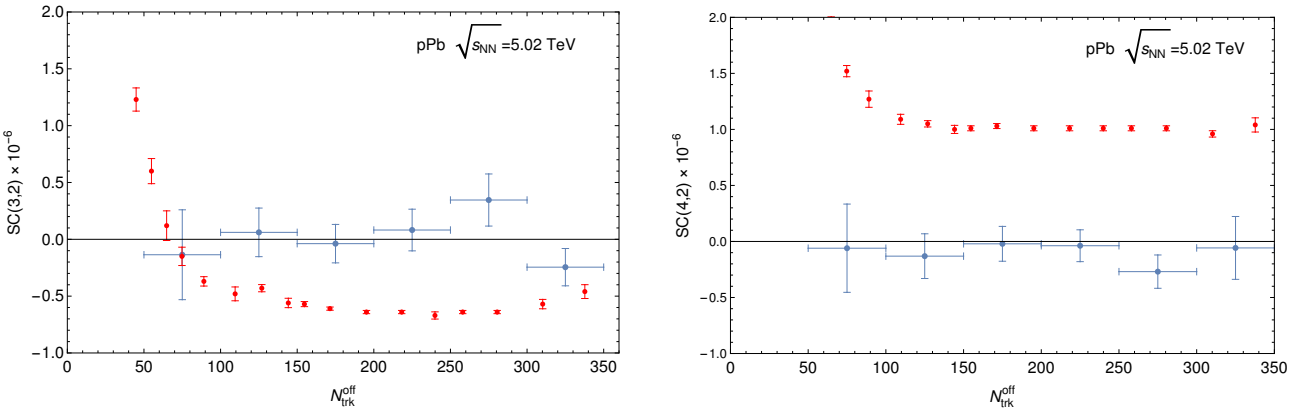


Figure 4.2: SC(3, 2) (left) and SC(4, 2) (right) as a function of the amount of offline tracks in 5.02 TeV proton-lead collisions. The simulation is done without substructure of the nucleons. The data points in red are the experimental data [38] and the data points in blue come from the simulation.

[16] The experiment uses $p_t > 0.4$ GeV. As we have to supply an upper bound for p_t to the simulation program, we chose a safe upper bound of 10, 000 GeV.

We would then like to confirm that simulations can match the experiments when we do take a substructure into account. Unfortunately, a problem arises when that is done. A nucleon is not a single Gaussian energy distribution anymore, but rather a superposition of Gaussian energy distributions that come from the quarks. It becomes less likely that the collision between all the constituents of two colliding nucleons is head-on, so the chances of maximal entropy production are slim. So in general, less entropy will be produced in a collision and therefore there is no data for the higher values of $N_{\text{trk}}^{\text{off}}$. This is supported by figure 4.3, where we clearly see that for high $N_{\text{trk}}^{\text{off}}$, there are no charged particles.

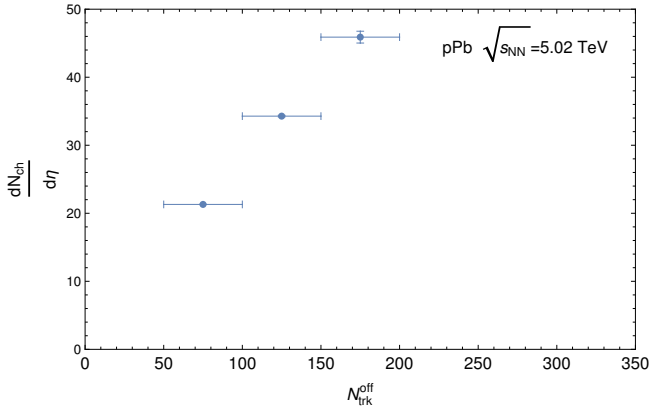


Figure 4.3: The amount of charged particles in 5.02 TeV proton-lead collisions as a function of $N_{\text{trk}}^{\text{off}}$.

4.3 Symmetric Cumulants for OO and XeXe collisions

We will now investigate the behaviour of the Symmetric Cumulants in oxygen-oxygen and xenon-xenon collisions. As there is no experimental data available of the Symmetric Cumulants for these kind of collisions, we will just show the results from the simulations. We start off with OO collisions, which is a relatively small system. We performed 100,000 collisions and the same p_t- and η-cuts are used as for pPb collisions. The results are shown in figure 4.4 and 4.5.

We continue with XeXe collisions. Xenon nuclei are larger than oxygen nuclei, but a bit smaller than lead nuclei. We performed 2,500 XeXe collisions and we again used the same cuts as for pPb collisions. The results are shown in figure 4.6 and 4.7.

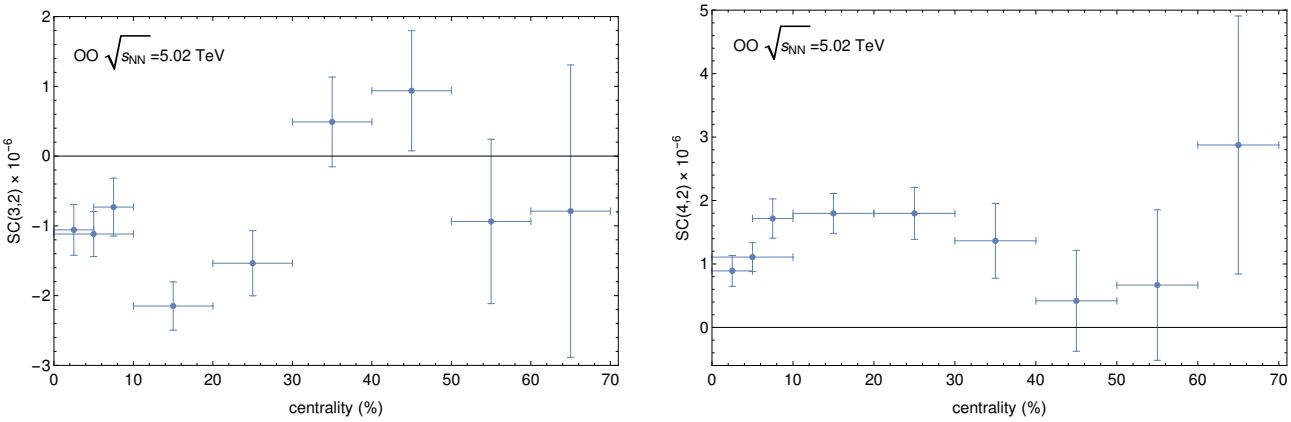


Figure 4.4: SC(3, 2) (left) and SC(4, 2) (right) as a function of centrality in 5.02 TeV oxygen-oxygen collisions.

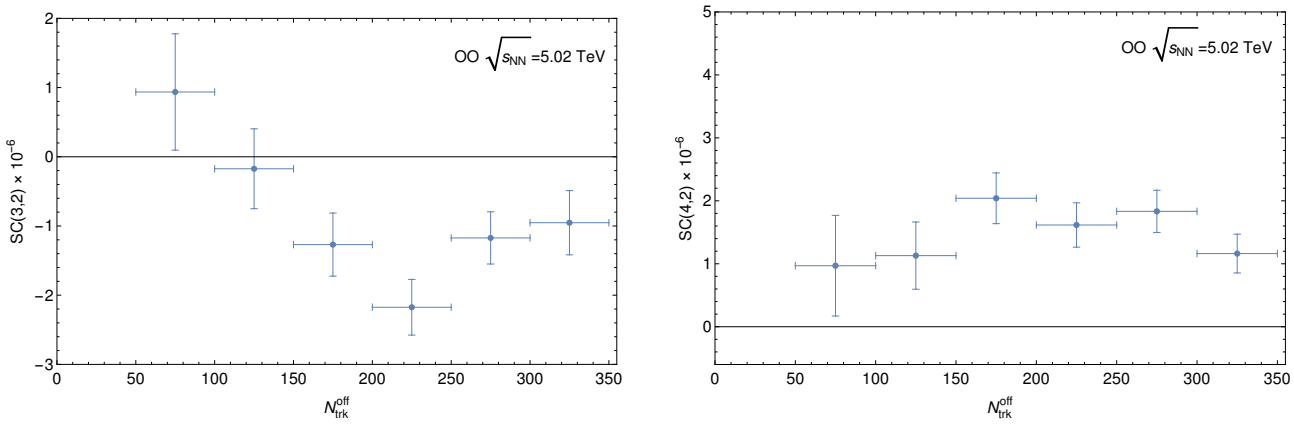


Figure 4.5: SC(3, 2) (left) and SC(4, 2) (right) as a function of the amount of offline tracks in 5.02 TeV oxygen-oxygen collisions.

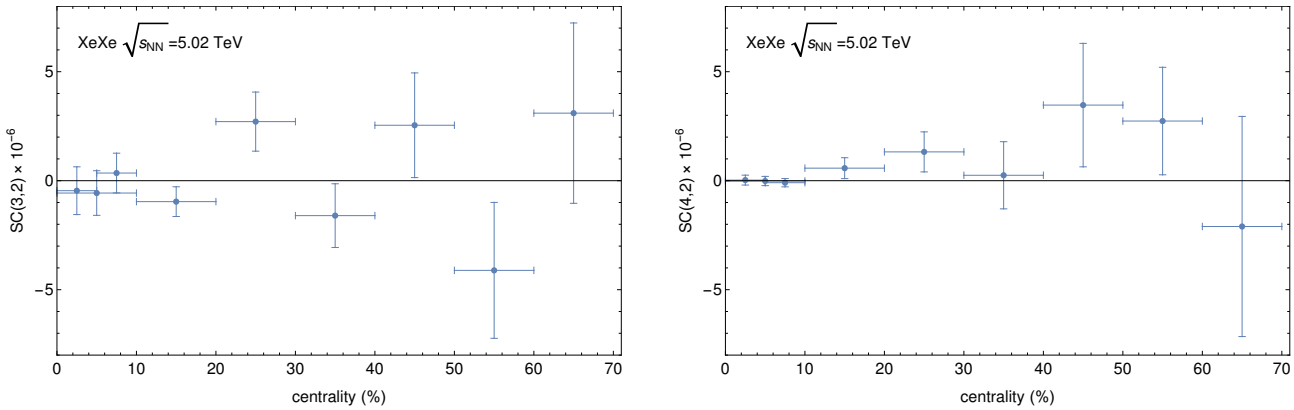


Figure 4.6: SC(3, 2) (left) and SC(4, 2) (right) as a function of centrality in 5.02 TeV xenon-xenon collisions.

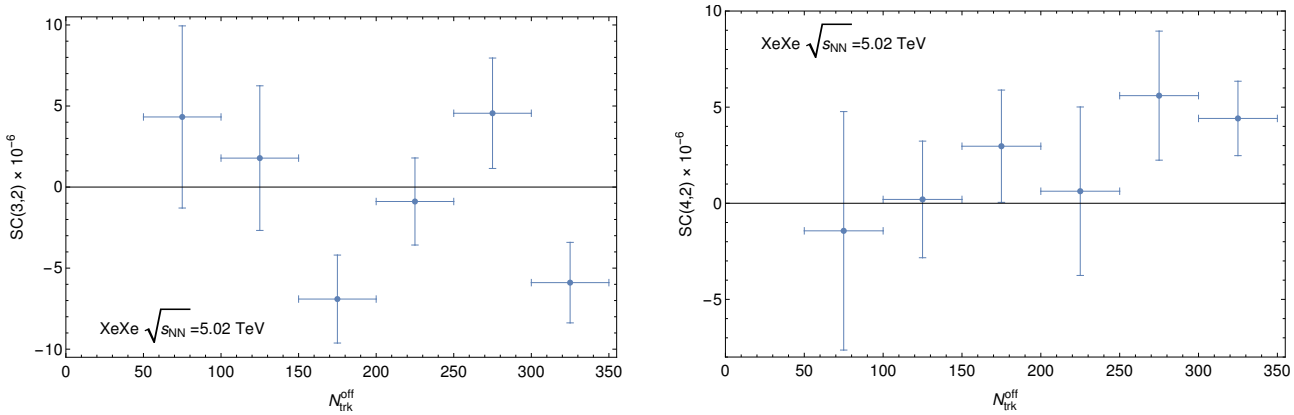


Figure 4.7: SC(3, 2) (left) and SC(4, 2) (right) as a function of the amount of offline tracks in 5.02 TeV xenon-xenon collisions.

Chapter 5

Concluding page

5.1 Conclusion

In the previous chapter we have looked at the Symmetric Cumulants as a new observable in the current simulation program. From section 4.1 we concluded that there is no apparent reason to believe that the code to compute the Symmetric Cumulants is wrong. It produces results that are consistent with experimental data.

In section 4.2, we saw that there is the need to introduce a substructure of the nucleons, which is physically also more realistic. We could not reproduce the experimental results when we do not model substructure. However, adding a substructure posed a different problem. We still could not reproduce the experimental results, because there is an absence of high $N_{\text{trk}}^{\text{off}}$. Experimentally, these higher $N_{\text{trk}}^{\text{off}}$ are measured however. From this we can conclude that the used set of parameters is not accurate enough. Perhaps, the modelled width and amount of quarks play a role. This is also a good reason why the Symmetric Cumulants are investigated for different kind of systems. The PbPb system is insensitive for the added substructure, while it turns out that it does matter for another system. Different systems therefore can behave differently.

At last, the behaviour of the Symmetric Cumulants in OO and XeXe collisions are shown in section 4.3. In some cases the pattern is more evident than in other cases. For OO collisions, one can conclude that 100,000 simulated collisions are really necessary to produce significant results that are not just consistent with zero. In particular, when the Symmetric Cumulants are plotted against $N_{\text{trk}}^{\text{off}}$, the trend is obvious. For XeXe collisions it is a bit harder. This is mainly because of the relatively small amount of simulations, namely 2,500. We did not do more, as simulations for big systems is a time-consuming activity. If the amount of simulations increases with a factor N, then the errors approximately decrease with a factor $1/\sqrt{N}$. In order to produce results that could be relevant for further research, the errors have to decrease at least by a factor of 2, meaning that a data set of 10,000 events could suffice. The amount of simulations that have to be done is important information for a Bayesian analysis in the future. This will come back in the outlook.

5.2 Summary and outlook

In this thesis, I tried to give a rather complete overview of the theory of heavy-ion collisions. We started off with the description of a quark-gluon plasma, which is a state of matter in QCD. It can be investigated if one performs a high-energy nuclear collision, i.e. a heavy-ion collision. The QGP is one of the stages that exist after such a collision and it can be described by relativistic hydrodynamics. After some time, the temperature has decreased enough to describe the system as a hadron resonance gas, i.e. a gas of individual particles. Eventually, these particles will be detected by particle detectors. These detections give information about the particles and from that we can extract useful physical quantities. We quantified the observed anisotropy in the transverse momentum distribution by introducing flow coefficients v_n. We looked for a reliable formula that can compute them. We then looked at the Symmetric Cumulants and implemented this observable in the simulation program. We produced results that are consistent with experimental data and found out that a substructure of the nucleons can drastically change the outcome of the observable. We noticed that we could improve the results if we introduce a bias in the generated events, such that more peripheral collisions are simulated more often, but that is for the future. At last, we simulated OO and XeXe collisions. We could not compare the results with experimental data, as there are none. However, it is useful to have this simulated data for a future Bayesian analysis. As this analysis is time-consuming, it is important to know how many simulations are sufficient for significant results. If it turns out that the outcome of the Symmetric Cumulants for OO and/or XeXe collisions is sensitive to the used parameters, experimentalists can consider to actually perform these experiments. We then would have a reliable source of data, instead of an educated guess. This helps to further constrain the parameters and to better understand the behaviour and properties of the quark-gluon plasma.

Appendix A

Fourier series and derivation of the flow coefficients v_n

Take a distribution function, i.e. **a probability density function** (p.d.f.), $f\colon \mathbb{R}/2\pi\mathbb{Z} \to \mathbb{R}$. This function $f(x)$ is 2π periodic. In our case, f will be the azimuthal transverse momentum distribution. Formally, one can always write down a power series of a function, even though it is possible that it does not converge at all. Let us write f as a Fourier series [39, Chapter 5], i.e.

$$f(x) = \sum_{k\in\mathbb{Z}} c_k e^{ikx}, \tag{A.1}$$

with $c_k \in \mathbb{C}$ for all $k \in \mathbb{Z}$. These coefficients are given by

$$c_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-ikx}dx. \tag{A.2}$$

A first constraint on c_k is given by the fact that f maps to the real numbers. This means that

$$\sum_{k\in\mathbb{Z}} c_k e^{ikx} = f(x) \overset{!}{=} f(x)^* = \sum_{k\in\mathbb{Z}} c_k^* e^{-ikx}, \tag{A.3}$$

and as e^{ikx} is orthogonal to e^{ilx} for $k \neq l$, we must have that $c_{-k} = c_k^*$ for all $k \in \mathbb{Z}$. Next, we can rewrite $f(x)$ a bit to get it into a more familiar form:

$$\begin{aligned}
f(x) = \sum_{k\in\mathbb{Z}} c_k e^{ikx} &= \sum_{k=-\infty}^{-1} c_k e^{ikx} + c_0 + \sum_{k=1}^{\infty} c_k e^{ikx} \\
&= c_0 + \sum_{k=1}^{\infty} \left(c_{-k}e^{-ikx} + c_k e^{ikx}\right) \\
&= c_0 + \sum_{k=1}^{\infty} \left[\left(c_k e^{ikx}\right)^* + c_k e^{ikx}\right] \\
&= c_0 + 2\sum_{k=1}^{\infty} \operatorname{Re}\left(c_k e^{ikx}\right) \\
&= c_0 + 2\sum_{k=1}^{\infty} |c_k| \operatorname{Re}\left(e^{i(kx+\varphi_k)}\right) \\
&= c_0 + 2\sum_{k=1}^{\infty} |c_k| \cos(kx+\varphi_k),
\end{aligned} \tag{A.4}$$

for some $\varphi \in [0, 2\pi)$. This already looks a lot like (3.3). Next, we calculate c_0. We now use the fact that f is a p.d.f., so if it is integrated over all space, it will result in 1.[17] This means that

$$c_0 = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)dx = \frac{1}{2\pi}. \tag{A.5}$$

[17] We can assume that f is already normalised. If not, we can redefine it as $f \mapsto \left(\int_{-\pi}^{\pi} f(x)dx\right)^{-1} f$.

We can relabel some variables and rescale[18] the coefficients to get the more common relation

$$f(\varphi) = \frac{1}{2\pi}\left(1 + 2\sum_{n=1}^{\infty} v_n \cos(n(\varphi - \Psi_n))\right). \tag{A.6}$$

Notice that we now have Ψ_n in our expression and not Ψ_R. This is indeed a difference, but this does not affect the experimental measurements [40]. That is because we do not know Ψ_R and Ψ_n a priori.

We will now proof that equation (3.4) is indeed the right expression. This requires a computation [41]:

$$\begin{aligned}
\langle\cos(n(\varphi - \Psi_n))\rangle &= \int_{-\pi}^{\pi} \cos(n(\varphi - \Psi_n)) f(\varphi) d\varphi \qquad (A.7)\\
&= \frac{1}{2\cdot 2\pi}\int_{-\pi}^{\pi}\left[e^{in(\varphi-\Psi_n)} + e^{-in(\varphi-\Psi_n)}\right]\left(1 + \sum_{k=1}^{\infty} v_k\left[e^{ik(\varphi-\Psi_k)} + e^{-ik(\varphi-\Psi_k)}\right]\right) d\varphi \\
&= \frac{1}{4\pi}\int_{-\pi}^{\pi} e^{in(\varphi-\Psi_n)} + e^{-in(\varphi-\Psi_n)} + \sum_{k=1}^{\infty} v_k\Big(e^{i(k+n)\varphi}e^{-i(n\Psi_n+k\Psi_k)} + \\
&\quad e^{i(n-k)\varphi}e^{-i(n\Psi_n-k\Psi_k)} + e^{i(k-n)\varphi}e^{i(n\Psi_n-k\Psi_k)} + e^{-i(n+k)\varphi}e^{i(n\Psi_n+k\Psi_k)}\Big) d\varphi \qquad (A.8)\\
&= \frac{1}{4\pi}\left(2\pi\delta_{n,k}e^{-i(n\Psi_n-k\Psi_k)}v_k + 2\pi\delta_{k,n}e^{-i(n\Psi_n-k\Psi_k)}v_k\right) \\
&= \frac{1}{4\pi}\left(2\pi v_n + 2\pi v_n\right) \\
&= v_n.
\end{aligned}$$

Here we used that integrating a complex exponential over the domain $[-\pi, \pi)$ results in zero if $|n + k| \neq 0$ or $|n - k| \neq 0$. If $|n - k|$ results in zero, the integral contributes 2π to the result, while $|n + k|$ can never become zero, as $n, k \geq 1$. We see that the expression for the n^{th} harmonic v_n is indeed given by equation (3.4).

[18] Notice that this factor $\frac{1}{2\pi}$ comes from the definition of the coefficients as in equation (A.2).

Appendix B

Generating functions for k-particle cumulants

We will not give a complete proof of the way the cumulants are defined, but it will be more a line of reasoning to show where they come from. Therefore, many results are just stated without proof.

There is a very elegant way to express these cumulants as detailed by [32, 42]. We will use the formalism of a **generating function**. In general, generating functions transform problems about sequences to problems about functions. We know a lot about manipulating functions, so this transformation can be of great help. For each event, we define $G_n \colon \mathbb{C} \to \mathbb{R}$ by

$$G_n(z) := \prod_{j=1}^{M} \left(1 + \frac{z^* e^{in\varphi_j} + z e^{-in\varphi_j}}{M}\right). \tag{B.1}$$

Next, we can average over all events to get $\langle G_n(z) \rangle$ and expand this in powers of z and z^*. The coefficients of this power series yield multiparticle azimuthal correlations of all orders. Now we can define

$$\mathcal{C}_n(z) := M \left(\langle G_n(z) \rangle^{1/M} - 1\right), \tag{B.2}$$

which we can also expand in powers of z and z^*:

$$\mathcal{C}_n(z) \ = \sum_{k,l} \frac{z^{*k} z^l}{k!\, l!} c_n\{k, l\}. \tag{B.3}$$

Equation (B.3) defines the cumulants. Of special interest are the coefficients for which $k = l$, because they are linked to v_n. So we define $c_n\{2k\} := c_n\{k, k\}$ and after some manipulation of functions, we obtain the following approximation [32]:

$$\mathcal{C}_n(z) \ \approx \ln I_0(2v_n|z|), \tag{B.4}$$

with I_0 a modified Bessel function of the first kind. We can match this to equation (B.3) after converting (B.4) into a power series in z and z^*. Coefficients can be matched and one then concludes that

$$v_n\{2\}^2 = \quad c_n\{2\}, \tag{B.5}$$

$$v_n\{4\}^4 = -c_n\{4\}, \tag{B.6}$$

which can be continued for higher order correlations.

Appendix C

Computations needed for the program

This appendix is added to explain certain parts of the code that is written. The code is shown in Appendix E. The main formulas are already discussed in Chapter 3, but this appendix gives an explanation why it is implemented in the way it is. The code is presented in Appendix E.

C.1 Rewriting product of complex numbers

To compute the quantities in equation (3.27), we need the real part of a product of three complex numbers. As a complex number is represented in the code by two real numbers, i.e. the real and imaginary part, instead of one complex number, we have to implement this manually. Generally, let $u, v, w \in \mathbb{C}$ such that $u = a + bi$, $v = c + di$, $w = e + fi$ with $a, b, c, d, e, f \in \mathbb{R}$. Then the real part of the product uvw is:

$$\begin{aligned} \mathrm{Re}(uvw) &= \mathrm{Re}\left[(a+bi)(c+di)(e+fi)\right] \\ &= \mathrm{Re}\left[(ac-bd+(ad+bc)i)(e+fi)\right] \\ &= (ac-bd)e-(ad+bc)f. \end{aligned} \tag{C.1}$$

C.2 Standard deviation

To compute the **variance** (also called the standard deviation or the error) of the Symmetric Cumulants $SC(m,n)$ we need the covariance matrix $C \in M_n(\mathbb{R})$ for stochastic variables $x_1, x_2, ..., x_n$, which is written as

$$C := \begin{pmatrix} \sigma_{x_1,x_1} & \sigma_{x_1,x_2} & \cdots & \sigma_{x_1,x_n} \\ \sigma_{x_2,x_1} & \sigma_{x_2,x_2} & \cdots & \sigma_{x_2,x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{x_n,x_1} & \sigma_{x_n,x_2} & \cdots & \sigma_{x_n,x_n} \end{pmatrix}. \tag{C.2}$$

The covariance σ_{x_i,x_j} has two important properties, namely:

(i) $\sigma_{x_i,x_j} = \sigma_{x_j,x_i}$ (symmetry),

(ii) $\sigma_{x_j,x_j} = \sigma^2_{x_j}$, the regular standard deviation squared.

Now consider a function f depending on the stochastic variables $x_1, x_2, ..., x_n$. The standard deviation is then given by [43]

$$\sigma_f^2 = \begin{pmatrix} \frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_n} \end{pmatrix} C \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}. \tag{C.3}$$

In the case of the Symmetric Cumulants we have that $f(X,Y,Z) = SC(m,n) = (X - YZ)(m,n)$. Its derivatives are easily found. The same method can be applied for the normalised Symmetric Cumulants $sc(m,n)$.

Appendix D

Parameters used for resulting plots

D.1 Parameterlist

We first give a full list of the used parameters with a short explanation where they come from.

Parameter list		
Parameter name	Symbol	Explanation
trentodmin	$d_{\min}$	minimal distance between nucleons
trentwow	w	width of nucleon in fm
trentosigmann	σ_{NN}	cross-section NN in mb
trentosigmafluct	-	multiplicity fluctuation standard deviation
trentop	p	p as in equation (2.22)
trentonorm	Norm	proportionality factor from thickness to entropy density
trentofreestreamingtime	τ_{fs}	time for free streaming
trentocouplingstrength	-	coupling strength during free streaming
trentonc	-	amount of constituents in nucleon
trentov	-	width of constituent of a nucleon in fm
numlatticesites	-	amount of lattice sites
latticesize	-	size of the lattice
rhomax	$\rho_{\max}$	$\rho_{\max}$ as in equation (2.33)
xi0	ξ_0	ξ_0 as in equation (2.31)
cflconstant	$\frac{\Delta t}{\Delta x}$	modified lattice spacing
shearhrg	$(\eta/s)_{\text{hrg}}$	η/s in de hadron resonance gas
shearcrv	$(\eta/s)_{\text{crv}}$	$(\eta/s)_{\text{crv}}$ as in equation (2.29)
shearmin	$(\eta/s)_{\text{min}}$	$(\eta/s)_{\text{min}}$ as in equation (2.29)
shearslope	$(\eta/s)_{\text{slope}}$	$(\eta/s)_{\text{slope}}$ as in equation (2.29)
shearrelaxationtime	τ_π	τ_π as in equation (2.13)
bulkmax	$(\zeta/s)_{\max}$	$(\eta/s)_{\max}$ as in equation (2.30)
bulkT0	$(\zeta/s)_{T_0}$	$(\eta/s)_{T_0}$ as in equation (2.30)
bulkwidth	$(\zeta/s)_{\text{width}}$	$(\zeta/s)_{\text{width}}$ as in equation (2.30)
bulkrelaxationtime	τ_Π	τ_Π as in equation (2.13)
deltapipiovertaupi	$\frac{\delta_{\pi\pi}}{\tau_\pi}$	parameter for second order hydrodynamics
phi7overpressure	$\varphi_7 p$	parameter for second order hydrodynamics
taupipiovertaupi	$\frac{\tau_{\pi\pi}}{\tau_\pi}$	parameter for second order hydrodynamics
lambdapiPiovertaupi	$\frac{\lambda_{\pi\Pi}}{\tau_\pi}$	parameter for second order hydrodynamics
deltaPiPiovertauPi	$\frac{\delta_{\Pi\Pi}}{\tau_\Pi}$	parameter for second order hydrodynamics
lambdaPipiovertauPi	$\frac{\lambda_{\Pi\pi}}{\tau_\Pi}$	parameter for second order hydrodynamics
freezeouttemp	T_{switch}	temperature that indicates change of system description
rapidityrange	$\lvert\eta_{\max}\rvert$	rapidity range for which particles will be counted

Table D.1: Parameter list of the parameters that are used to determine the system with its physical meaning.

D.2 Values used parameters

The following values for the parameters are used. If we do not take substructure into account, we do not need trentonc and trentov.

TRENTo:
trentodmin=0
trentow=0.89
trentosigmann=70
trentosigmafluct=1.1
trentop=0.03
trentonorm=20.8
trentofreestreamingtime=0.59
trentocouplingstrength=0
trentonc=6
trentov=0.52

secondorderhydro:
numlatticesites=100
latticesize=25
rhomax=10
xi0=0.1

finitedifferencesolver2/ musclsolverktminmodmidpoint:
cflconstant=0.08

LatticeEOStempdep:
shearhrg=0.06
shearmin=0.06
shearslope=2
shearcrv=0.05
shearrelaxationtime=5
bulkmax=0.015
bulkT0=0.154
bulkwidth=0.02
bulkrelaxationtime=0.2
deltapipiovertaupi=1.33333333333333333333333333333333
phi7overpressure=0.128571
taupipiovertaupi=1.42857
lambdapiPiovertaupi=1.2
deltaPiPiovertauPi=0.66666666666666666666666666666666
lambdaPipiovertauPi=1.6

cooperfryehadronizer:
freezeouttemp=155
rapidityrange=4

Appendix E

Code Symmetric Cumulants

```
#ifndef SC_H
#define SC_H

#include "measuredquantity.h"

class SC : public MeasuredQuantity{
private:
    int Orderm;
    int Ordern;
    bool Charged;
    int ParticleClass;

    int NumParticles;
    long double ReQm;
    long double ImQm;
    long double ReQn;
    long double ImQn;
    long double ReQmplusn;
    long double ImQmplusn;
    long double ReQmminusn;
    long double ImQmminusn;
    long double RHS11; // first part of equation with average over particles
    long double RHS21; // second part of equation with average over particles
    long double RHS31; // third part of equation with average over particles

    void WriteSpecificSettings(ofstream& OutputFileIn);
public:
    SC(bool* CreationSuccess, int OrdermIn, int OrdernIn, bool ChargedIn, int ParticleClassIn);
    ~SC();
    bool Reset();
    bool Setup();
    void AddInput(int NumParticlesInParticleArray, Particle** ParticleArray, long double*
        AuxiliaryQuantitiesIn);
    long double GetSubMeasurement(int Number);
    long double GetSubMeasurementWeight(int Number);
    bool DoFinalComputations(ofstream& FileStream, ofstream& MathematicaFileStream, long double
        LowerCentralityBound, long double UpperCentralityBound, bool CentralityBins, AverageQuantity*
        AverageQuantityIn);
};

#endif
```

```cpp
#include "SC.h"
#include "averagequantity.h"
#include <math.h>
#include <iostream>
using namespace std;

SC::SC(bool* CreationSuccess, int OrdermIn, int OrdernIn, bool ChargedIn, int ParticleClassIn) : MeasuredQuantity(){
    Orderm = OrdermIn;
    Ordern = OrdernIn;
    Charged = ChargedIn;
    ParticleClass = ParticleClassIn;

    Name = "SC";
    Name += to_string(Orderm);
    Name += to_string(Ordern);
    Description = "SC(";
    Description += to_string(Orderm) + ",";
    Description += to_string(Ordern) + ") of ";
    if(Charged){
        Name += "charged";
        Description += "charged";
    }
    else
        Description += "all";
    if(ParticleClass == 0)
        Description += " particles.";
    else if(ParticleClass == 1){
        Name += "pion";
        Description += " pions.";
    }
    else if(ParticleClass == 2){
        Name += "kaon";
        Description += " kaons.";
    }
    else if(ParticleClass == 3){
        Name += "proton";
        Description += " protons.";
    }
    else if(ParticleClass == 4){
        Name += "neutron";
        Description += " neutrons.";
    }
    else if(ParticleClass == 5){
        Name += "lambda";
        Description += " lambdas";
    }
    else if(ParticleClass == 6){
        Name += "xi";
        Description += " xis";
    }
    else if(ParticleClass == 7){
        Name += "omega";
        Description += " omegas";
    }

    NumParameters = 3;
    ParameterNames = new string[NumParameters];
    ParameterDescriptions = new string[NumParameters];
    Parameters = new long double[NumParameters];
    if(ParameterNames == 0 || ParameterDescriptions == 0 || Parameters == 0){
        *CreationSuccess = false;
        return;
    }
    ParameterNames[0] = "lowerptbound";
```

```
65      ParameterNames[1] = "upperptbound";
66      ParameterNames[2] = "etamax";
67      ParameterDescriptions[0] = "Lower pt boundary for which particles will be counted.";
68      ParameterDescriptions[1] = "Upper pt boundary for which particles will be counted.";
69      ParameterDescriptions[2] = "Maximum absolute value of eta for which particles will be counted.";
70
71      *CreationSuccess = true;
72  }
73
74  SC::~SC(){
75      return;
76  }
77
78  void SC::WriteSpecificSettings(ofstream& OutputFileIn){
79      return;
80  }
81
82  bool SC::Reset(){
83      NumParticles = 0;
84      ReQm = 0.;
85      ImQm = 0.;
86      ReQn = 0.;
87      ImQn = 0.;
88      ReQmplusn = 0.;
89      ImQmplusn = 0.;
90      ReQmminusn = 0.;
91      ImQmminusn = 0.;
92      RHS11 = 0.;
93      RHS21 = 0.;
94      RHS31 = 0.;
95
96      return true;
97  }
98
99  bool SC::Setup(){
100     NumSubMeasurements = 3;
101     SubMeasurementNames = new string[3];
102     if(SubMeasurementNames == 0){
103         cout << "Error: failed to allocate memory." << endl;
104         return false;
105     }
106     //SC(m,n) = X - Y * Z
107     SubMeasurementNames[0] = "RHS11"; //first part RHS of equation with average over particles
108     SubMeasurementNames[1] = "RHS21"; //second part RHS of equation with average over particles
109     SubMeasurementNames[2] = "RHS31"; //third part RHS of equation with average over particles
110
111     return true;
112 }
113
114 void SC::AddInput(int NumParticlesInArray, Particle** ParticleArray, long double*
        AuxiliaryQuantitiesIn){
115     for(int i = 0; i < NumParticlesInArray; i++){
116         if(Charged){
117             if(!ParticleArray[i]->IsCharged())
118                 continue;
119         }
120         if(ParticleClass == 1){
121             if(!ParticleArray[i]->IsPion())
122                 continue;
123         }
124         else if(ParticleClass == 2){
125             if(!ParticleArray[i]->IsKaon())
126                 continue;
127         }
128         else if(ParticleClass == 3){
129             if(!ParticleArray[i]->IsProton())
```

```
130                 continue;
131         }
132         else if(ParticleClass == 4){
133             if(!ParticleArray[i]->IsNeutron())
134                 continue;
135         }
136         else if(ParticleClass == 5){
137             if(!ParticleArray[i]->IsLambda())
138                 continue;
139         }
140         else if(ParticleClass == 6){
141             if(!ParticleArray[i]->IsXi())
142                 continue;
143         }
144         else if(ParticleClass == 7){
145             if(!ParticleArray[i]->IsOmega())
146                 continue;
147         }
148 
149         long double ParticlepT = ParticleArray[i]->pT();
150         long double Particleeta = ParticleArray[i]->eta();
151         if(ParticlepT < Parameters[0] || ParticlepT > Parameters[1] || fabs(Particleeta) > Parameters
                [2])
152             continue;
153         // calculate flow vector
154         long double ParticlePhi = ParticleArray[i]->phi();
155         long double ReQmAddition = cos((long double)(Orderm) * ParticlePhi);
156         long double ImQmAddition = sin((long double)(Orderm) * ParticlePhi);
157         long double ReQnAddition = cos((long double)(Ordern) * ParticlePhi);
158         long double ImQnAddition = sin((long double)(Ordern) * ParticlePhi);
159         long double ReQmplusnAddition = cos(((long double)(Orderm + Ordern)) * ParticlePhi);
160         long double ImQmplusnAddition = sin(((long double)(Orderm + Ordern)) * ParticlePhi);
161         long double ReQmminusnAddition = cos(((long double)(Orderm - Ordern)) * ParticlePhi);
162         long double ImQmminusnAddition = sin(((long double)(Orderm - Ordern)) * ParticlePhi);
163 
164         NumParticles++;
165         ReQm += ReQmAddition;
166         ImQm += ImQmAddition;
167         ReQn += ReQnAddition;
168         ImQn += ImQnAddition;
169         ReQmplusn += ReQmplusnAddition;
170         ImQmplusn += ImQmplusnAddition;
171         ReQmminusn += ReQmminusnAddition;
172         ImQmminusn += ImQmminusnAddition;
173     }
174 
175     long double NumParticleslong = (long double)NumParticles;
176     RHS11 = (ReQm * ReQm + ImQm * ImQm) * (ReQn * ReQn + ImQn * ImQn);
177     RHS11 -= 2. * (ImQmplusn * ImQn * ReQm - ImQm * ImQn * ReQmplusn + ImQm * ImQmplusn * ReQn + ReQm
            * ReQmplusn * ReQn);
178     RHS11 -= 2. * (-ImQmminusn * ImQn * ReQm + ImQm * ImQn * ReQmminusn + ImQm * ImQmminusn * ReQn +
            ReQm * ReQmminusn * ReQn);
179     RHS11 += ReQmplusn * ReQmplusn + ImQmplusn * ImQmplusn;
180     RHS11 += ReQmminusn * ReQmminusn + ImQmminusn * ImQmminusn;
181     RHS11 -= (NumParticleslong - 4.) * (ReQm * ReQm + ImQm * ImQm + ReQn * ReQn + ImQn * ImQn);
182     RHS11 += NumParticleslong * (NumParticleslong - 6.);
183     RHS11 /= NumParticleslong * (NumParticleslong - 1.) * (NumParticleslong - 2.) * (NumParticleslong -
             3.);
184     RHS21 = (ReQm * ReQm + ImQm * ImQm - NumParticleslong) / (NumParticleslong * (NumParticleslong -
            1.));
185     RHS31 = (ReQn * ReQn + ImQn * ImQn - NumParticleslong) / (NumParticleslong * (NumParticleslong -
            1.));
186 }
187 
188 long double SC::GetSubMeasurement(int Number){
189     if(NumParticles == 0)
```

```
190         return 0.;
191     else if(Number == 0)
192         return RHS11;
193     else if(Number == 1)
194         return RHS21;
195     else if(Number == 2)
196         return RHS31;
197     else
198         return 0.;
199 }
200
201 long double SC::GetSubMeasurementWeight(int Number){
202     long double NumParticleslong = (long double)NumParticles;
203     if(NumParticles == 0)
204         return 0.;
205     else if(Number == 0)
206         return NumParticleslong * (NumParticleslong - 1.) * (NumParticleslong - 2.) * (NumParticleslong
              - 3.);
207     else if(Number == 1 || Number == 2)
208         return NumParticleslong * (NumParticleslong - 1.);
209     else
210         return 0.;
211 }
212
213 bool SC::DoFinalComputations(ofstream& FileStream, ofstream& MathematicaFileStream, long double
      LowerCentralityBound, long double UpperCentralityBound, bool CentralityBins, AverageQuantity*
      AverageQuantityIn){
214     if(!FileStream.is_open() || !MathematicaFileStream.is_open())
215         return false;
216
217     FileStream << "# " << Name << ":" << endl;
218     FileStream << "# Format: " << Name << " = #1 \u00b1 #2 (stat.)." << endl;
219     MathematicaFileStream << "{{\"" << Name << "\",\"" << LowerCentralityBound << "to" <<
          UpperCentralityBound;
220     if(CentralityBins)
221         MathematicaFileStream << "central\",\"centrality [%]";
222     else
223         MathematicaFileStream << "ntrkoff\",\"\\!\\(\\*SubsuperscriptBox[\\(N\\),\\(trk\\),\\(off\\)
              ]\\)";
224     MathematicaFileStream << "\",\"SC(" << Orderm << "," << Ordern << ")";
225     MathematicaFileStream << "\",\"";
226     if(ParticleClass == 1)
227         MathematicaFileStream << "\\[Pi]";
228     else if(ParticleClass == 2)
229         MathematicaFileStream << "K";
230     else if(ParticleClass == 3)
231         MathematicaFileStream << "p";
232     else if(ParticleClass == 4)
233         MathematicaFileStream << "n";
234     else if(ParticleClass == 5)
235         MathematicaFileStream << "\\[CapitalLambda]";
236     else if(ParticleClass == 6)
237         MathematicaFileStream << "\\[CapitalXi]";
238     else if(ParticleClass == 7)
239         MathematicaFileStream << "\\[CapitalOmega]";
240     else
241         MathematicaFileStream << "total";
242
243     MathematicaFileStream << "\"},{Around[" << ConvertNumberToMathematicaFormat(0.5 * (
          UpperCentralityBound + LowerCentralityBound)) << "," << ConvertNumberToMathematicaFormat(0.5 *
           (UpperCentralityBound - LowerCentralityBound)) << "],Around[";
244
245     long double RHS1 = AverageQuantityIn->GetSubMeasurement(0);
246     long double RHS2 = AverageQuantityIn->GetSubMeasurement(1);
247     long double RHS3 = AverageQuantityIn->GetSubMeasurement(2);
248     long double SCmn = RHS1 - RHS2 * RHS3;
```

```
249
250     long double Error1 = AverageQuantityIn->GetSubMeasurementError(0);
251     long double Error2 = AverageQuantityIn->GetSubMeasurementError(1);
252     long double Error3 = AverageQuantityIn->GetSubMeasurementError(2);
253     long double Cov12 = AverageQuantityIn->GetSubMeasurementCovariance(0, 1);
254     long double Cov13 = AverageQuantityIn->GetSubMeasurementCovariance(0, 2);
255     long double Cov23 = AverageQuantityIn->GetSubMeasurementCovariance(1, 2);
256     long double ErrorOfSCmn = Error1 * Error1 - 2. * Cov13 * RHS2 + Error3 * Error3 * RHS2 * RHS2 - 2.
            * Cov12 * RHS3 + 2. * Cov23 * RHS2 * RHS3 + Error2 * Error2 * RHS3 * RHS3;
257
258     FileStream << SCmn << "\t" << sqrt(ErrorOfSCmn) << endl;
259     FileStream << "\n" << endl;
260     if(ErrorOfSCmn < 0. || Error1 == -1. || Error2 == -1. || Error3 == -1.){
261         MathematicaFileStream << "Missing[],Missing[]]}}";
262     }
263     else{
264         ErrorOfSCmn = sqrt(ErrorOfSCmn);
265         MathematicaFileStream << ConvertNumberToMathematicaFormat(1000000. * SCmn) << "," <<
                ConvertNumberToMathematicaFormat(1000000. * ErrorOfSCmn) << "]}}";
266     }
267
268     // normalised SC
269     MathematicaFileStream << ",{{\"" << "small" << Name << "\",\"" << LowerCentralityBound << "to" <<
            UpperCentralityBound;
270     if(CentralityBins)
271         MathematicaFileStream << "central\",\"centrality [%]";
272     else
273         MathematicaFileStream << "ntrkoff\",\"\\!\\(\\*SubsuperscriptBox[\\(N\\),\\(trk\\),\\(off\\)
                ]\\)";
274     MathematicaFileStream << "\",\"sc(" << Orderm << "," << Ordern << ")";
275     MathematicaFileStream << "\",\"";
276     if(ParticleClass == 1)
277         MathematicaFileStream << "\\[Pi]";
278     else if(ParticleClass == 2)
279         MathematicaFileStream << "K";
280     else if(ParticleClass == 3)
281         MathematicaFileStream << "p";
282     else if(ParticleClass == 4)
283         MathematicaFileStream << "n";
284     else if(ParticleClass == 5)
285         MathematicaFileStream << "\\[CapitalLambda]";
286     else if(ParticleClass == 6)
287         MathematicaFileStream << "\\[CapitalXi]";
288     else if(ParticleClass == 7)
289         MathematicaFileStream << "\\[CapitalOmega]";
290     else
291         MathematicaFileStream << "total";
292
293     MathematicaFileStream << "\"},{Around[" << ConvertNumberToMathematicaFormat(0.5 * (
            UpperCentralityBound + LowerCentralityBound)) << "," << ConvertNumberToMathematicaFormat(0.5 *
             (UpperCentralityBound - LowerCentralityBound)) << "],Around[";
294
295     long double smallSCmn = SCmn / (RHS2 * RHS3);
296     long double Der1 = 1. / (RHS2 * RHS3);
297     long double Der2 = - RHS1 / (RHS2 * RHS2 * RHS3);
298     long double Der3 = - RHS1 / (RHS3 * RHS3 * RHS2);
299     long double ErrorOfsmallSCmn = Der1 * Der1 * Error1 * Error1 + 2. * Der1 * (Der2 * Cov12 + Der3 *
            Cov13) + Der2 * Der2 * Error2 * Error2 + 2. * Der2 * Der3 * Cov23 + Der3 * Der3 * Error3 *
            Error3;
300
301     FileStream << smallSCmn << "\t" << sqrt(ErrorOfsmallSCmn) << endl;
302     FileStream << "\n" << endl;
303
304     if(ErrorOfsmallSCmn < 0. || Error1 == -1. || Error2 == -1. || Error3 == -1.){
305         MathematicaFileStream << "Missing[],Missing[]]}}";
306     }
```

```
307     else{
308         ErrorOfsmallSCmn = sqrt(ErrorOfsmallSCmn);
309         MathematicaFileStream << ConvertNumberToMathematicaFormat(smallSCmn) << "," <<
                ConvertNumberToMathematicaFormat(ErrorOfsmallSCmn) << "]}}";
310     }
311     return true;
312 }
```

References

[1] S. Bethke. α_s 2002. *Nuclear Physics B - Proceedings Supplements*, 121:74–81, Jun 2003.

[2] Ulrich Heinz and Raimond Snellings. Collective flow and viscosity in relativistic heavy-ion collisions. *Annual Review of Nuclear and Particle Science*, 63(1):123–151, Oct 2013.

[3] Ed Daw. Special relativity, lecture 7 rapidity and pseudorapidity, 2011/2012.

[4] Wojciech Broniowski and Wojciech Florkowski. Geometric relation between centrality and the impact parameter in relativistic heavy-ion collisions. *Physical Review C*, 65(2), Jan 2002.

[5] V. Ozvenchuk, A. Rybicki, A. Szczurek, A. Marcinek, and M. Kiełbowicz. Spectator induced electromagnetic effects in heavy-ion collisions and space-time-momentum conditions for pion emission, 2019.

[6] Jonah E. Bernhard. Bayesian parameter estimation for relativistic heavy-ion collisions, 2018.

[7] Paul Romatschke. New developments in relativistic viscous hydrodynamics. *International Journal of Modern Physics E*, 19(01):1–53, 2010.

[8] Barbara Ryden. *Introduction to Cosmology*. Pearson Education, Inc., 2nd edition, 2017.

[9] Pavel Kovtun. Lectures on hydrodynamic fluctuations in relativistic theories. *Journal of Physics A: Mathematical and Theoretical*, 45(47):473001, nov 2012.

[10] Derek A Teaney. Viscous hydrodynamics and the quark gluon plasma. In *Quark-Gluon Plasma 4*, pages 207–266. World Scientific, 2010.

[11] Huichao Song. Causal viscous hydrodynamics for relativistic heavy ion collisions, 2009.

[12] Pavel K Kovtun, Dan T Son, and Andrei O Starinets. Viscosity in strongly interacting quantum field theories from black hole physics. *Physical review letters*, 94(11):111601, 2005.

[13] Miklos Gyulassy. The qgp discovered at rhic. In *Structure and Dynamics of Elementary Matter*, pages 159–182. Springer, 2004.

[14] Fábio S Bemfica, Marcelo M Disconzi, and Jorge Noronha. Nonlinear causality of general first-order relativistic viscous hydrodynamics, 2019.

[15] L Del Zanna, V Chandra, G Inghirami, V Rolando, A Beraudo, A De Pace, G Pagliara, A Drago, and F Becattini. Relativistic viscous hydrodynamics for heavy-ion collisions with echo-qgp. *The European Physical Journal C*, 73(8):2524, 2013.

[16] Rudolf Baier, Paul Romatschke, Dam Thanh Son, Andrei O Starinets, and Mikhail A Stephanov. Relativistic viscous hydrodynamics, conformal invariance, and holography. *Journal of High Energy Physics*, 2008(04):100, 2008.

[17] G. S. Denicol, S. Jeon, and C. Gale. Transport coefficients of bulk viscous pressure in the 14-moment approximation. *Physical Review C*, 90(2), Aug 2014.

[18] J. Scott Moreland, Jonah E. Bernhard, and Steffen A. Bass. Alternative ansatz to wounded nucleon and binary collision scaling in high-energy nuclear collisions. *Phys.Rev.*, C92(1):011901, 2015.

[19] Q.Y. Shou, Y.G. Ma, P. Sorensen, A.H. Tang, F. Videbæk, and H. Wang. Parameterization of deformed nuclei for glauber modeling in relativistic heavy ion collisions. *Physics Letters B*, 749:215–220, Oct 2015.

[20] Kevin Dusling and Shu Lin. Dilepton production from a viscous qgp. *Nuclear Physics A*, 809(3-4):246–258, Sep 2008.

[21] Wojciech Broniowski, Wojciech Florkowski, Mikolaj Chojnacki, and Adam Kisiel. Free-streaming approximation in early dynamics of relativistic heavy-ion collisions. *Physical Review C*, 80(3), Sep 2009.

[22] James D Bjorken. Highly relativistic nucleus-nucleus collisions: The central rapidity region. *Physical review D*, 27(1):140, 1983.

[23] Chun Shen, Zhi Qiu, Huichao Song, Jonah Bernhard, Steffen Bass, and Ulrich Heinz. The iebe-vishnu code package for relativistic heavy-ion collisions. *Computer Physics Communications*, 199:61–85, 2016.

[24] Matthew Luzum and Hannah Petersen. Initial state fluctuations and final state correlations in relativistic heavy-ion collisions. *Journal of Physics G: Nuclear and Particle Physics*, 41(6):063102, 2014.

[25] Piotr Bożek. Bulk and shear viscosities of matter created in relativistic heavy-ion collisions. *Physical Review C*, 81(3):034909, 2010.

[26] Steffen A Bass, Mohamed Belkacem, Marcus Bleicher, Mathias Brandstetter, L Bravina, Christoph Ernst, Lars Gerland, M Hofmann, S Hofmann, Jens Konopka, et al. Microscopic models for ultrarelativistic heavy ion collisions. *Progress in Particle and Nuclear Physics*, 41:255–369, 1998.

[27] Marcus Bleicher, E Zabrodin, Christian Spieles, Steffen A Bass, Christoph Ernst, Sven Soff, L Bravina, Mohamed Belkacem, Henning Weber, Horst Stöcker, et al. Relativistic hadron-hadron collisions in the ultra-relativistic quantum molecular dynamics model. *Journal of Physics G: Nuclear and Particle Physics*, 25(9):1859, 1999.

[28] Jean-Yves Ollitrault. Anisotropy as a signature of transverse collective flow. *Physical Review D*, 46(1):229, 1992.

[29] Ante Bilandzic, Raimond Snellings, and Sergei Voloshin. Flow analysis with cumulants: Direct calculations. *Physical Review C*, 83(4):044913, 2011.

[30] Arthur M Poskanzer and Sergei A Voloshin. Methods for analyzing anisotropic flow in relativistic nuclear collisions. *Physical Review C*, 58(3):1671, 1998.

[31] Ante Bilandzic. *Anisotropic flow measurements in ALICE at the large hadron collider*. PhD thesis, Utrecht University, 2012.

[32] Nicolas Borghini, Phuong Mai Dinh, and Jean-Yves Ollitrault. Flow analysis from multiparticle azimuthal correlations. *Physical Review C*, 64(5):054901, 2001.

[33] Nicolas Borghini, Phuong Mai Dinh, and Jean-Yves Ollitrault. Thermalization and elliptic flow at rhic, 2001.

[34] Ilya Selyuzhenkov. Anisotropic flow and other collective phenomena measured in pb-pb collisions with alice at the lhc. *Progress of Theoretical Physics Supplement*, 193:153–158, 2012.

[35] Jaroslav Adam, Dagmar Adamová, Madan M Aggarwal, G Aglieri Rinella, Michelangelo Agnello, Nikita Agrawal, Zubayer Ahammed, Shamim Ahmad, Sang Un Ahn, Salvatore Aiola, et al. Correlated event-by-event fluctuations of flow harmonics in pb-pb collisions at $\sqrt{s_{NN}} = 2.76$ tev. *Physical review letters*, 117(18):182301, 2016.

[36] H.P. Lopuhaä L.E. Meester F.M. Dekking, C. Kraaikamp. *A Modern Introduction to Probability and Statistics*. Springer, 2010.

[37] Georges Aad, B Abbott, J Abdallah, O Abdinov, R Aben, M Abolins, OS AbouZeid, H Abramowicz, H Abreu, R Abreu, et al. Measurement of the correlation between flow harmonics of different order in lead-lead collisions at s nn= 2.76 tev with the atlas detector. *Physical Review C*, 92(3):034903, 2015.

[38] CMS collaboration et al. Observation of correlated azimuthal anisotropy fourier harmonics in pp and ppb collisions at the lhc, 2017.

[39] E.P. van den Ban. *Lecture notes Functions and Series*. Mathematical Institute, Utrecht University, June 2017.

[40] B Alver and G Roland. Collision-geometry fluctuations and triangular flow in heavy-ion collisions. *Physical Review C*, 81(5):054905, 2010.

[41] Alexander Frank. Analyses of anisotropic flow and symmetry planes using multiparticle correlation techniques in alice at the large hadron collider. Bachelor thesis, Technical University of Munich, Sept 2016.

[42] Nicolas Borghini, Phuong Mai Dinh, and Jean-Yves Ollitrault. Flow analysis from cumulants: A practical guide, 2001.

[43] Peter van Capel. *Data acquisition and applied analysis*. Julius Institute, Utrecht University, 2015/2016.