Freeze-Out in the Multi-Module Model of Ultra-Relativistic Heavy Ion Collisions

Cand.scient. Thesis
Øyvind Heggø Hansen
June 19, 2003

Advisor: Prof. L. P. Csernai

Department of Physics
University of Bergen
Norway

Contents

Chapter 1

Introduction

Heavy ion research can be divided into different energy domains. The *relativistic* domain covers collision energies between 100 AMeV (energy per nucleon) and 10 AGeV. Studies in this energy range probe the properties of the nuclear equation of state (EOS) and associated collective phenomena, such as compressibility and flow patterns. There is also a connection to astrophysics, especially neutron stars and supernova explosions, where similar densities exist.

At energies above 10 AGeV we enter the domain of *ultra-relativistic* heavy ion collisions (uRHIC), where the rest energy of the colliding matter is very small compared to the momentum. The main focus of the uRHIC research is to study the formation of quark-gluon plasma (QGP). Being a separate phase of matter where single quarks and gluons exist in a deconfined state, QGP is believed to have existed in the very dense early universe, only milliseconds after the big bang.

The principle of the experiments is simple: Nuclei of heavy elements such as Pb and Au are accelerated to velocities very close the speed of light and then brought to collide.

In most collisions the two nuclei will not hit each other head on, and in that case we can define two regions in the collision system: The participant region is the part of the system where two nuclei hit each other and intermix, while the spectator region is the matter that don't take a direct part in the collision. In the participant region the matter almost immediately becomes extremely hot and dense, forming a 'fireball'. If the energy density exceeds the critical energy density for liberation of the quarks from the hadrons the belief is that the matter in the fireball will transform into the QGP-phase.

The threshold for QGP formation depends on both the energy density and the chemical potential of the nuclear matter (see Figure 1.1 for an illustration). Thus, it is also believed that some neutron stars contain QGP because of the very high baryonic chemical potential existing at their core.

The top experimental facilities for uRHIC research are CERN's Super Proton Synchrotron (SPS), which will soon cease its operation, and new and more powerful Relativistic Heavy Ion Collider (RHIC), which started to operate at the end of year 2000 at Brookhaven National Laboratory.

An even more powerful accelerator, the Large Hadron Collider (LHC), with about 30 times higher collision energy than RHIC, is currently under construction at CERN and is scheduled for completion in the second half of this decade.

In experiments the QGP-phase will only endure for a very short moment, around 10 fm/c. The fireball will expand and cool by the force of its internal pressure and the matter will *hadronize* (transition back into the hadronic phase). The continuing expansion eventually leads to the total disintegration of the fireball into a swarm of particles that will be picked up by the detectors. Most of them are secondary particles (mesons and baryon/antibaryon-pairs) formed in the collision.

In February 2000 CERN announced at a special seminar that they had created QGP at the SPS. However, many scientists believe that CERN only created a mixed phase of hadronic matter and QGP. This doubt can exist because there isn't any practical, immediate and unambiguous way of telling from experimental data if the QCD phase transition did indeed take place, as the QGP itself is long gone by the time of detection.

Some observable clues of QGP existence have been identified though. These include, but are not limited to, the following:

- We should observe more strange particles compared to a QGP-less collisions, due to lower production threshold of strange quarks in the QGP phase.

- Transverse flow patterns should show specific azimuthal anisotropy because the collective properties of QGP are different from those of hadronic matter.

- Less J/Ψ particles should be observed because of color screening in the QGP.

- Analysis of two-particle correlations can probe the age and size of the source (fireball) at the time of particle creation. QGP has more degrees of freedom for storage and release of energy, and thus the source should last longer than a comparable hadronic source. Since the size of the system can be probed we can also work out its energy density and see if it is above the threshold energy of quark deconfinement.

- Different vector mesons decay into lepton pairs of different energies. Usually observations should show a clear peak around the mass of a vector meson, but at high energies, where we believe QGP is created, the peaks must appear to be smeared out.

- Photon emission is highly dependent on the temperature of the source, and thus we should be able to detect photons emitted from the very hot QGP.

A thorough theoretical study is necessary to understand the signatures of QGP. But modeling heavy ion collisions in theoretical physics is challenging, because from experiment we only know the beginning and end states, while all the complex and exotic physics that take place in between is practically impossible to probe directly in experiments. Some assumptions have to be made, and the various models currently in use are built on different theoretical foundations that all have their shortcomings, consequently each model has its advantages and disadvantages, and none of them can claim to offer a complete description of the reaction.

1.1 Structure of the Thesis

In this text we concentrate on fluid dynamics, one of the oldest means of simulating high energy nuclear collisions, that for several reasons remains popular even today. We will work with a three-dimensional fluid dynamical solution coupled with a separate phenomenological initial state model. Fluid dynamics provides a continuum description of the collision, and we will try to make an interface that allows us to translate the results into particle distributions that are comparable to experimental results.

- Chapter 2 presents the basic physical quantities involved in relativistic fluid mechanics along with the most common variables of relativistic heavy ion physics. In Chapter 3 we then present an efficient three-dimensional solution method for fluid mechanics along with the EOS of hadronic- and quark-matter.

- In Chapter 4 we examine the construction of an initial state for the hydrodynamical solution, and present a recent initial state-model that incorporates QGP creation.

- We then move on to discuss the final parts of the reaction, where the matter undergoes a phase transition from fluid to gas. This is described in the so-called freeze-out process, which is defined to take place across a three-dimensional hypersurface in the space-time. Recent advances in the analytical description of freeze-out across a propagating surface is presented in Chapter 5.

- The main goal of the work performed by the author of this text was the determination of a detailed freeze-out surface to be used with the three-dimensional solution of the hydrodynamics. To reap the benefits of a three-dimensional solution it is very important to have a realistic description of the freeze-out process. So we developed a method that, given the results of a particular hydrodynamical code run, will tailor a surface based on the properties of the matter in the fluid phase. The method was implemented in a computer program, and an example surface was produced. This work is presented in Chapter 6.

I have tried to write this thesis to the level of advanced undergraduate students. Basic knowledge of particle and nuclear physics, statistical and thermal physics and relativity is required, but the theory and terminology that is particular to the field of heavy ion physics is for the most part introduced as we go along and, with the possible exception of a few sections, prior experience in the field is not a requirement for the understanding.

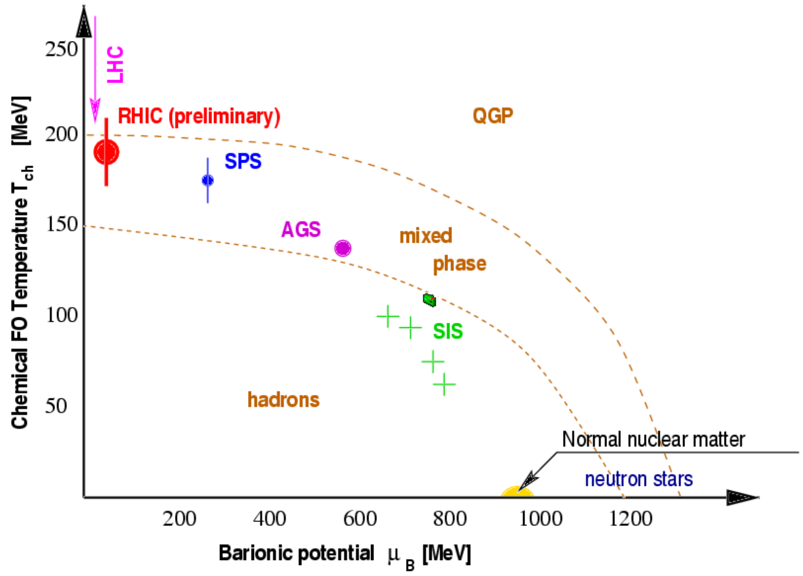

Figure 1.1: QCD phase diagram. The RHIC (preliminary) and LHC (theoretical) points are close zero baryonic chemical potential. In contrast to this neutron stars have near zero temperature and high baryonic chemical potential.

Chapter 2

Basic Definitions and Terminology

In trying to make this text as self contained and complete as possible we will first review the foundation that all the subsequent chapters rest on. This review is based on [1], which should be consulted for a more thorough exposition of the subject matter. In this chapter, and throughout the rest of this text we will use the (very practical) convention $c = \hbar = 1$, which is reflected in the units we use.

2.1 Elementary Relativistic Kinetic Theory

Kinetic theory utilizes both microscopic and macroscopic variables. The microscopic are the mass, energy, momentum and position of the particles, while the macroscopic are the collective properties of a macroscopic system of matter, such as flow velocity, density, temperature and other thermodynamical variables. A macroscopic system is taken to be a composite system containing a statistically significant number of microscopic constituents, but can still be very small compared to an everyday object. A heavy nucleus contains a couple of hundred nucleons, and thus two colliding nuclei form a system of circa five hundred nucleons. In addition to this we can have several thousand particles produced in the collision, depending on the collision energy. This constitutes an ensemble large enough to make statistical mechanics applicable, even though it differs a great deal from the typical everyday applications of statistical mechanics.

Relativistic Momentum and Velocity Throughout this text we assume that the reader is familiar with the basics of special relativity. The following only serves to introduce the notation standard that is used in this text.

- The four-momentum of a particle is defined as $p^\mu = (p^0, \vec{p}) = (E, \vec{p})$, where $p^0 = E$ is the energy of the particle. Taking the length of the four momentum we get:

$$p_\mu p^\mu \equiv \sum_\mu p_\mu p^\mu = m^2, \qquad \mu = 0, 1, 2, 3. \tag{2.1}$$

This also demonstrates the convention of dropping the summation sign when taking scalar products of four-vectors. The p^μ are called the *contravariant* components, and the $p_\mu = (E, -\vec{p})$ are called the *covariant* components. They are related as

$$p_\mu = g_{\mu\nu} p^\nu , \tag{2.2}$$

where $g_{\mu\nu} = \mathrm{diag}(1, -1, -1, -1)$ is the metric tensor.

- The four-velocity u^μ is a unit vector pointing in the direction of the motion in space-time. The three-velocity is defined as $\vec{v} = \vec{p}/p^0$ and by using $\gamma = 1/\sqrt{(1-\vec{v}^2)}$ we get

$$u^\mu = (\gamma, \gamma\vec{v}) \qquad \text{and} \qquad u_\mu = (\gamma, -\gamma\vec{v}), \tag{2.3}$$

for the contra- and covariant components respectively.

- General four-vectors are aggregates of four quantities, one related to the time coordinate, and one related to each of the three space coordinates. A four-vector q may be *space-like*, $q_\mu q^\mu < 0$, or *time-like*, $q_\mu q^\mu > 0$. Time-like four-vectors reside inside the lightcone and space-like four-vectors reside outside of the lightcone and there is no Lorenz transformation that can turn one type of four-vector into the other. Causally related events can always be joined in space-time by a line that has a time-like tangent four-vector.

Coordinate System and Rapidity In heavy ion physics it is customary to use a special coordinate system where the spatial z-axis is aligned with the direction of the accelerator's particle beam. Many collisions will be off center, and we define the *impact vector* $\vec{b}$ as the component orthogonal to the z-axis of the vector connecting the centers of the beam particle and the target particle. $\vec{b}$ is a two dimensional vector. Its direction is usually denoted as the x-direction. The plane spanned by the axes x and z is called the *reaction plane* of the collision.

The *rapidity*, denoted y, is a generalization of of the velocity. While the velocity is limited by the speed of light, $v \in (-c, c) = (-1, 1)$, the rapidity has no such limitation and may take all values; $y \in (-\infty, \infty)$. The definition of rapidity is

$$y \equiv \mathrm{arcth}(v_\parallel) = \mathrm{arcth}\left(\frac{p_\parallel}{p^0}\right) = \frac{1}{2}\ln\left(\frac{p^0 + p_\parallel}{p^0 - p_\parallel}\right). \tag{2.4}$$

For small velocities $y \approx v_\parallel$.

After a collision a particle is moving with a velocity $\vec{v}$ in some direction. It is then customary to give its phase space position by the coordinates $(y, \vec{p_\perp}/m)$, where m is the mass of the particle. In the non-relativistic limit $(y, \vec{p_\perp}/m) \to (v_\parallel, v_\perp)$. The momentum vector can be decomposed as $p^\mu = (p^0, p_\parallel, \vec{p_\perp})$. We also note that rapidities are additive under Lorenz transformations.

Macroscopic Quantities Common macroscopic quantities include the following: *(i)* local density $n(\vec{r}, t) = n(x)$. This quantity is not an invariant scalar because three-volume is not invariant under Lorenz transformations. *(ii)* Particle current $\vec{j}(\vec{r}, t) = \vec{j}(x)$ across unit area per unit time, *(iii)* particle four flow

$N^\mu = (n(x), \vec{j}(x))$ and *(iv)* particle distribution in the six-dimensional phase space (μ-space), which gives the number of particles, N, in a phase space volume element

$$f(x,p): \qquad N = f(x,p)\triangle^3 x \triangle^3 p. \tag{2.5}$$

In a given reference frame the density and current can be expressed in terms of the distribution function:

$$n(x) = \int d^3p f(x,p), \tag{2.6}$$

$$\vec{j}(x) = \int d^3p \vec{v} f(x,p). \tag{2.7}$$

We know that $\vec{v} = \frac{\vec{p}}{p^0}$, and thus the particle four flow may be expressed as

$$N^\mu = \int \frac{d^3p}{p^0} p^\mu f(x,p). \tag{2.8}$$

The Energy-Momentum Tensor The energy-momentum tensor (see [2]) is another important quantity that gives us the entire 'picture' of the energy and momentum of a given piece of matter. In terms of the microscopic distribution function it can be written as

$$T^{\mu\nu} = \int \frac{d^3p}{p^0} p^\mu p^\nu f(x,p), \tag{2.9}$$

while with macroscopic variables it is expressed as

$$T^{\mu\nu} = (e+P)u^\mu u^\nu + P g^{\mu\nu}, \tag{2.10}$$

where e is the energy density and P is the pressure of the piece of matter we are considering, and $w = (e+P)$ is the enthalpy density. Note that this version of the energy-momentum tensor does not include fields and potential energy, nor does it include dissipation (entropy producing processes). If dissipation cannot be neglected we need to add a dissipative term, so that the energy-momentum tensor can be decomposed into two parts:

$$T^{\mu\nu} = T^{\mu\nu(0)} + T^{\mu\nu(1)}, \tag{2.11}$$

where $T^{\mu\nu(0)}$ is the reversible(non-dissipative) energy-momentum tensor Eq. 2.10 and $T^{\mu\nu(1)}$ is the irreversible part, which again contains contributions from viscous stress and heat flow.

Local Rest Frame It is always important to keep in mind what reference frame we are using. Since the systems we are dealing with are fluids and gases, different parts of the system can move with different velocities, and the definition of the reference frame becomes a bit complicated at times.

The Local Rest Frame (LR) is the reference frame where the flow velocity $u^\mu = u^\mu_{LR} = (1,0,0,0)$. u^μ is always time-like, so there always exists a Lorenz transformation that leads to the LR.

In fluid dynamics there are two definitions of the LR:

Eckart's LR In Eckart's approach we tie the local rest frame to conserved charges, like baryon charge. We define a unit vector in the direction of the particle four flow as

$$u^\mu = \frac{N^\mu}{(N_\nu N^\nu)^{1/2}}. \tag{2.12}$$

If we decompose the four components of the flow velocity, $u^\mu = (\gamma, \gamma\vec{v})$, we can see that the three-vector of the flow velocity, $\vec{v}$, is parallel to the particle current $\vec{j}$ according to Eckart's definition. N^μ is always a time-like four-vector, and consequently there is no flow in spatial directions in the LR when Eckart's definition is used.

Eckarts's local rest frame is ill-suited for the description of high temperature matter flow. This is because such matter is dominated by radiation and mesons, and the baryon density is low and unrepresentative of the physical qualities of the matter. This is the case in ultra-relativistic heavy ion collisions, and was also the case in the early universe. Also a coherent flow of particles and antiparticles would yield vanishing flow according to Eckart's definition.

Landau's LR Landau's approach ties the local rest frame to the energy flow. In the LR the spatial components of the energy flow should vanish,

$$T^{0i}_{LR} = T^{i0}_{LR} = 0, \qquad i = 1, 2, 3. \tag{2.13}$$

Since the energy flow four-vector is $T^{\mu\nu}u_\nu$ and should be parallel to u^μ we get

$$\triangle_{\sigma\mu} T^{\mu\nu} u_\nu = 0, \tag{2.14}$$

where

$$\triangle^{\mu\nu} \equiv g^{\mu\nu} - \frac{u^\mu u^\nu}{u^\sigma u_\sigma} \tag{2.15}$$

is the orthogonal projection of the flow velocity. This definition shows that u^μ is the normalized eigenvector of $T^{\mu\nu}$, since $T^{\mu\nu}u_\nu$ is parallel to u^μ. Consequently

$$u^\mu = constant \times T^{\mu\nu} u_\nu. \tag{2.16}$$

From the normalization of u^μ the constant is $(u_\rho T^{\rho\nu} u_\nu)^{-1/2}$.

Two More Important Macroscopic Quantities We can now define the *invariant scalar density*

$$n \equiv N^\mu u_\mu \tag{2.17}$$

as the density in the local rest frame:

$$n = N^0_{LR}. \tag{2.18}$$

The *invariant scalar energy density* e is defined as:

$$e \equiv u_\mu T^{\mu\nu} u_\nu, \tag{2.19}$$

so that in the local rest frame

$$e = T^{00}_{LR}. \tag{2.20}$$

2.2 Perfect Fluid Mechanics

Fluid mechanics, or hydrodynamics, is a macroscopic theory, so the fluid is considered to be a continuous medium. A small volume element in such a fluid is still considered large compared to the elementary molecules of the fluid, so that it in itself is a continuous distribution of matter.

We also consider the fluid to be in local thermal equilibrium, so that all the thermodynamical quantities are defined at any given point.

Perfect fluid dynamics concerns itself with non-viscous or *ideal* fluids. It starts out with the conservation of particle current (in the application on heavy ions only baryons are conserved) and of energy and momentum. These conservation laws can be expressed as

$$N^{\mu}{}_{,\mu} = 0 \tag{2.21}$$

and

$$T^{\mu\nu}_{,\mu} = 0, \tag{2.22}$$

where we have introduced the notation $_{,\mu} = \partial_\mu = \partial/\partial x_\mu$. We may also express these conservation laws in a different form. Let us use $u^\mu = (\gamma, \gamma\vec{v})$, $w = e + P$, $T^{ik} = w\gamma^2 v_i v_k + P\delta_{ik}$ and $T^{00} = T_{00} = (e+Pv^2)\gamma^2$. By introducing the *apparent density*,

$$\mathcal{N} \equiv n\gamma, \tag{2.23}$$

and the *momentum current density* $\vec{\mathcal{M}}$ and the *apparent energy density* $\mathcal{E}$,

$$\vec{\mathcal{M}} \equiv T^{0i} = w\gamma^2\vec{v} \tag{2.24}$$

$$\mathcal{E} \equiv T^{00} = (e + Pv^2)\gamma^2, \tag{2.25}$$

the continuity equation and the energy and momentum conservation can be expressed as

$$(\partial_t + \vec{v}\mathrm{grad})\mathcal{N} = -\mathcal{N}\mathrm{div}\vec{v}, \tag{2.26}$$

$$(\partial_t + \vec{v}\mathrm{grad})\vec{\mathcal{M}} = -\vec{\mathcal{M}}(\mathrm{div}\vec{v}) - \mathrm{grad}P, \tag{2.27}$$

and

$$(\partial_t + \vec{v}\mathrm{grad})\mathcal{E} = -\mathcal{E}\mathrm{div}\vec{v} - \mathrm{div}(P\vec{v}), \tag{2.28}$$

respectively. Note that Eq. 2.27 is the well known Euler equation (see e.g. [3]).

Relativistic perfect fluid dynamics also requires an equation of state (EOS) of the matter in order to work. The quantities $\mathcal{N}, \vec{\mathcal{M}}$ and $\mathcal{E}$ are not directly related to the EOS, but must be obtained by solving the set of equations Eq. 2.23, Eq. 2.24 and Eq. 2.25.

It can easily be shown that entropy never increases in perfect fluid dynamics. This is of course a departure from the general truth about real macroscopic physical processes, but the relativistic generalization of viscous fluid dynamics is problematic, as it may yield unstable solutions [4].

It is also interesting to note that numerical solution methods introduce so-called numerical viscosity into perfect fluid mechanics. This phenomenon is a result of the finite resolution of the solver and is absent in analytical calculations.

Fluid mechanical models are popular in heavy ion physics because they let us test different EOS and see their effect on the collective properties of the matter. The solutions are transparent, and usually in good qualitative agreement

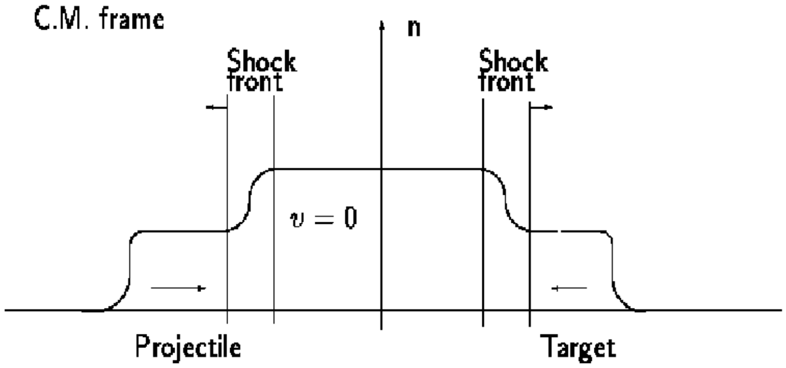

Figure 2.1: Density profile of two nuclei colliding supersonically, seen in the center-of-mass (CM) frame. The density increase happens in the narrow shock fronts. Figure reproduced from [1].

with experiments. Still there is an ongoing controversy about the equilibrium assumption, and non-equilibrium models such as parton cascades and molecular dynamics (which usually make some physical assumptions themselves) can give better quantitative accuracy of many measurables.

Solutions of perfect fluid dynamics applicable to heavy ion collisions include simple analytical solutions in one or two spatial dimensions such as shock waves and detonation waves and the phenomenological models of Bjorken and Landau, that will be discussed in Section 4.1. These models make many simplifying assumptions, but their analytical solubility provides us with valuable insight with a small computational cost.

There are also several different numerical solution algorithms, like the particle-in-cell method, our method of choice, that is described in Section 3.1. Such algorithms allows us to perform detailed three-dimensional simulations of heavy ion collisions.

2.3 Shock Waves

Shocks are discontinuities where the properties of the matter change in a very fast or even sudden manner. In the volume where this change takes place the matter forms a shock wave that propagates through matter at a speed faster than the local speed of sound (Figure 2.1). A good example of a shock wave is the Mach-cone created at the nose of supersonic aircraft, that consists of air with higher density and pressure than the surroundings, and can be felt on the ground as a quick peak in the otherwise atmospheric pressure ('sonic boom').

In perfect fluid mechanics shocks are infinitely sharp, and we consider them to happen across a surface (front) in space-time. We describe this surface in terms of its unit normal four-vector Λ^μ, which can be space-like or time-like.

In the following '0' denotes the initial state and '1' denotes the final state. We also use the notation $[A] = A_1 - A_0$, where A is a physical quantity. Orthogonally to the front we should observe conservation of particle number and the energy-momentum tensor:

$$[T^{\mu\nu}\Lambda_\mu] = 0 \tag{2.29}$$

and

$$[N^\mu \Lambda_\mu] = 0. \tag{2.30}$$

These relations are known as the relativistic Rankine-Hugoniot equations [5].

The Equation of the Raleigh Line By taking the parallel projection of the energy-momentum tensor to the surface normal, $[T^{\mu\nu}\Lambda_\nu] = 0$, we can derive the equation of the Raleigh line:

$$j^2 = \frac{[P](\Lambda^\mu \Lambda_\mu)}{[X]}, \tag{2.31}$$

Where w is the enthalpy density and $X \equiv w/n^2$ is the generalized specific volume. In the relativistic domain the $[P, X]$ plane is used instead of the $[P, V]$ plane. The equation of the Raleigh line is a straight line in the $[P, X]$ plane, connecting the initial state with the possible end states. The strength of a shock is characterized by j.

The Taub Adiabat The equation of the *Taub adiabat* or shock adiabat is expressed as

$$[P] = \frac{[wX]}{(X_1 + X_0)}. \tag{2.32}$$

Like a normal adiabat this equation describes a curve in the $[P, X]$ plane. This curve depends on the final state EOS and on the initial state, and the equation then gives the locus of the possible final state P and X values. The Taub adiabat differs from the Poisson (standard) adiabat in its dependence on both the initial and final states of the matter. If the two states have the same EOS the Taub adiabat goes through both the initial and final points in the $[P, X]$ plane.

If the EOS of the beginning and the end state differs, the Taub adiabat does not go through the initial point, and we speak of detonation, deflagration and condensation waves. The change in EOS may happen due to phase transitions, chemical changes, etc.

Another notable and useful equation gives the relative speed of matter on the two sides of the front:

$$v_{01} = \frac{v_0 - v_1}{1 - v_0 v_1/c^2} = \sqrt{\frac{(P_1 - P_0)(e_1 - e_0)}{(e_0 + P_1)(e_1 + P_0)}}. \tag{2.33}$$

Chapter 3

Numerical Fluid Dynamics

One usually assume that after a short relaxation period from the initial impact, relativistic heavy ion collisions can be validly described by one-fluid mechanics, especially if QGP is formed, as interactions are then strong and frequent. In addition the matter is very dense, meaning trouble for many transport theoretical models that are often assuming binary collisions only. We can use a numerical algorithm to simulate the further evolution of the fluid in three dimensions, allowing us to try out different EOS to see their effect on the collective flow patterns.

It is necessary to limit the region where we apply the fluid dynamical model, though. The initial state can not be described within the idealized world of fluid mechanics, and neither can the dilute final state. The expanding system will ultimately be so sparse that it is sensible to break up the fluid, which is a continuous matter distribution, into a particle distribution that can be processed using a statistical particle production model to produce results comparable to experimental data. The initial and final states are discussed in Chapter 4 and Chapter 5 respectively.

The two main approaches to solving the fluid dynamical equations are the Eulerian and the Lagrangian. They both utilize a grid in space as their computational frame. In the Eulerian approach the grid is fixed in space and the fluid can move relative to the grid. The Lagrangian approach fixes the grid to the fluid itself, and the cells change shape and size as the fluid moves about.

3.1 The Particle-in-Cell(PIC) Method

The particular solution algorithm that we have used for our computations is the particle-in-cell one-fluid code that was developed by Amsden and Harlow at Los Alamos National Laboratory as a fast and efficient algorithm with some necessary drawbacks in accuracy. It's formulation is given in [6]. It is actually a combination of the Eulerian and Lagrangian approaches: Its grid is fixed in space, but the computation also relies on fictitious numerical marker particles, which should not be confused with real elementary particles. The marker particles are set up so that they share the conserved charges of the real particles amongst them, but the number of marker particles is greater than than the corresponding number of real particles.

The collective quantities N^μ and $T^{\mu\nu}$ of the fluid are defined in the Eulerian grid and the marker particles are used to account for transfer of energy, momentum and charge between cells. At each time step the following operations are performed:

- Compute the energy, momentum and charge densities for each cell. This is done by solving eqs. 2.21 and 2.22:

$$\partial_\mu N^\mu = 0, \qquad \partial_\mu T^{\mu\nu} = 0$$

- Distribute the computed quantities evenly between the fictitious marker particles in the cell. Then assign a velocity to each marker particle by interpolating among the velocities of the neighboring cells in the grid. If a marker particle crosses a cell border, its energy, momentum and charge are subtracted from the old cell and added to the new cell. It should be noted that the velocities of the marker particles are not related to the momentum and energy, but rather decided by the interpolation.

The use of marker particles 'quantizes' the transfer of physical quantities between cells. This technique becomes a problem at low densities, such as at the late and final stages of the reaction, when the amount of marker particles contained in a cell may be very small. We can preempt this problem by breaking up the fluid before density gets too low.

3.2 Equation of State

As mentioned earlier the EOS is an important factor in any relativistic fluid calculation. For the work described in this text we have used relatively simple choices of EOS, and a phenomenological description of the phase transition. Still, the essential qualitative properties of the matter are represented.

3.2.1 Bag Model EOS

The EOS currently used in the our version of the PIC code is the MIT bag model, one of the simplest equations of state for QGP. It assumes the existence of a *pertubative vacuum*. This is different from the physical vacuum of the laboratory because it limits the freedom of movement within it. The quarks and gluons are kept in the pertubative vacuum(a 'bag'), and we can never observe free quarks and gluons as we are outside the confines the 'bag'.

The strength of the QCD coupling decreases at high energies. The bag model uses the so-called *bag constant* to parameterize the containment of the quarks and gluons, and this decreases asymptotically as the energy density increases. The EOS has the following following formulation:

$$\begin{aligned}
e_q(T,\mu_q) &= \frac{\pi^2}{15}(N_c^2 - 1 + \frac{7N_cN_f}{4})T^4 + \frac{N_cN_f}{2}(T^2\mu_q^2 + \frac{\mu_q^4}{2\pi^2}) + B, \\
P_q(T,\mu_q) &= \frac{1}{3}(e_q(T,\mu_q) - B) - B, \\
n_q(T,\mu_q) &= \left(\frac{\partial P_q}{\partial \mu_q}\right)_T = \frac{N_cN_f}{3\pi^2}(\mu_q^3 + \pi^2T^2\mu_q),
\end{aligned} \tag{3.1}$$

where μ_q is the quark chemical potential, n_b is the baryon charge density in the QCD phase and B is the bag constant.

3.2.2 Hadronic EOS

A simple EOS for the hadronic phase was presented in [8]. It takes contributions from three different terms, using the baryon density n and the temperature T as independent variables. The corresponding thermodynamical potential is the Helmholtz free energy, $f = e - Ts$, where e is the energy density and s is the entropy density. The EOS takes the form:

$$f_{\mathrm{H}}(n,T) = f_{\mathrm{Boltzmann}}(n,T) + f_{\mathrm{compr}}(n,K) + f_{\mathrm{meson}}(T), \tag{3.2}$$

where K is an isothermal compressibility:

$$K = 9\left(\frac{\partial P}{\partial n}\right)_{n=n_0,T=0}. \tag{3.3}$$

Temperatures are high enough to ignore Fermi-Dirac statistics for nucleons, so we use Boltzmann statistics in the first term:

$$f_{\mathrm{Boltzmann}}(n,T) = nm + nT\left\{\ln\left[\frac{n}{g_N}\left(\frac{2\pi}{mT}\right)\right]\right\}. \tag{3.4}$$

For the compression term there are a lot of different parametrizations, such as linear, quadratic, Sierk-Nix and Grant-Kapusta. Consult [8] for details on these parameterizations.

Only pions are included in the meson term, since these are by far the most commonly produced mesons. For massless pions we have

$$f_{\mathrm{meson}}(T) = -\frac{\pi^2}{30}T^4, \tag{3.5}$$

a term that becomes important at higher temperatures when the baryon density is small.

3.2.3 Mixed Phase

The phase equilibrium between hadronic matter and QGP has to fulfill the Gibbs criteria, namely that

$$T_{\mathrm{H}} = T_{\mathrm{QGP}}, \qquad P_{\mathrm{H}} = P_{\mathrm{QGP}} \qquad \text{and} \qquad \mu_{\mathrm{H}} = \mu_{\mathrm{QGP}}.$$

The last of these requirements doesn't apply in the baryon free case. In [8] the Maxwell construction(see e.g. [10]) was used to construct a phenomenological phase transition, that combined with a hadronic EOS makes for a complete coverage of the phase diagram. During the mixed phase of a slow phase transition the volume fractions of the pure phases are

$$\alpha_{\mathrm{Q}}(T) = \frac{n - n_{\mathrm{cH}}(T)}{n_{\mathrm{cQ}}(T) - n_{\mathrm{cH}}(T)}, \qquad \alpha_{\mathrm{H}}(T) = 1 - \alpha_{\mathrm{Q}}(T).$$

The thermodynamical quantities of the mixed phase are then:

$$P_{\mathrm{M}}(T) = P_{\mathrm{c}}, \qquad \mu_{\mathrm{M}}(T) = \mu_{\mathrm{c}}(T),$$
$$e_{\mathrm{M}}(n,T) = \alpha_{\mathrm{H}} e_{\mathrm{H}}(n_{\mathrm{cH}}(T),T) + \alpha_{\mathrm{Q}} e_{\mathrm{Q}}(n_{\mathrm{cQ}}(T),T),$$
$$s_{\mathrm{M}}(n,T) = \alpha_{\mathrm{H}} s_{\mathrm{H}}(m_{\mathrm{cH}}(T),T) + \alpha_{\mathrm{Q}} s_{\mathrm{Q}}(n_{\mathrm{cQ}}(T),T).$$

3.2.4 Supercooled QGP EOS

The phenomenological phase transition presented in the previous section does not take into account the possibility that the matter may become supercooled and eventually undergo a fast phase transition from that state.

The number of degrees of freedom of the matter decreases as the QGP hadronizes, and consequently there should be a reduction of the entropy density, seemingly a violation of the second law of thermodynamics.

Two solutions to this problem have been proposed. One is the suggestion that a QGP source has a lower 'explosivity' compared to a hadronic matter source, leading to a prolonged first order phase transition from the expanding system. Thus, the increase in the volume of the system will make up for the reduction of the entropy density.

Such a hadronization would require a time of $50-100$fm [11, 12]. By analysis of correlations among identical pions, that yield information about the space-time distribution of the source (HBT-effect), we can test the hypothesis. If it is true, there should be an observable peak in the R_{side}/R_{out} ratio [13, 14]. Current results from RHIC do not support this slow hadronization scenario.

The other solution, which appears more attractive, is a supercooling scenario; the QGP expands into a supercooled (unstable) state, and hadronization happens fast as a direct crossover without any intermediate mixed phase. This hypothesis can be checked in hydrodynamical models using a supercooled QGP EOS as input.

A simple phenomenological EOS that allows supercooling has been suggested in [15, 16] for baryon-free QGP with two quark flavours:

$$P(T) = \frac{1}{3}e(T) - bT, \qquad e(T) = e_{\mathrm{SB}}, \tag{3.6}$$

where e and T are connected by the relation:

$$e = T\frac{dP}{dT} - P \tag{3.7}$$

This is called the spinoidal EOS because it produces a local minimum in the pressure profile.

Chapter 4

The Initial State

The early stages of ultra relativistic heavy ion collisions are for the most part poorly understood. The strong interactions, high density and lack of equilibrium make it a dynamically complex situation. Fluid mechanical models that consider all the nuclear matter as one fluid are insufficient to describe the initial stage. But they are not alone in this respect. As a matter of fact no existing model of any kind can describe the initial state completely and unambiguously.

If one still want to use fluid dynamics, the two- or three-fluid models [17, 18, 19] should be considered. Here we divide the fluid into different components, each described by a separate EOS. The target and projectile are represented as different fluids, becoming intermixed as the collision develops. But such models have a set of problems of their own. The determination of drag, and friction terms is difficult. Another is problem is the mixing of the components later on: When the use of to or three different equations of state no longer makes a realistic or practical model the fluids should mix together, but determination of the transfer term can be problematic. Phase transitions also become problematic as the different spatially overlapping components may be in different phases.

The use of non-equilibrium models, like ultra-relativistic quantum molecular dynamics (UrQMD) [20] or parton cascades (for example MPC [21]) is also problematic. Such models often assume a dilute gas where only binary collisions take place. But the system is heavily Lorenz contracted making its density and energy density very high. Many of the models were originally developed for much lower energies than what is available in today's experiments. The modeling of particle interactions have to be heavily modified, and this is complicated and computationally time consuming. Thus many approximations and assumptions have to be made

Pertubative Quantum Chromo Dynamics(pQCD) would have been the ideal model for describing these highly energetic systems, but unfortunately it is inapplicable for ultra-relativistic heavy ion reactions at current experimental energies.

4.1 Simple Phenomenological Models

A few simple phenomenological solutions of fluid mechanics have been developed. Originally both of the following models were developed for p+p-collisions, but they are also applicable to *central* heavy ion collisions.

4.1.1 Landau's Model

Landau's model [22, 23] is applicable at energies in the range $E_{lab} = 10 - 100$ AGeV. At these highly relativistic energies both the projectile and target will be Lorenz-contracted to such a degree that we can treat them as flat disks, propagating in the z-direction with an orientation transverse to it.

Because of the high energy and density the mean free paths are small compared to the characteristic length. This means that the relaxation time is short, and we can assume local thermal equilibrium from the very beginning, thereby avoiding to deal with a complicated non-equilibrium situation.

The model starts by assuming complete stopping of the 'disks' at the moment of impact. When equilibrium is established the system starts to expand in the z-direction and cool, making it a $1+1$ (space + time) dimensional model at this stage. When the system has reached a size in the z-direction comparable to its transverse diameter, we let the expansion enter the second stage, where we also consider transverse expansion. This makes the model $2 + 1$-dimensional at the second stage.

We choose to use as an initial condition the disks of matter comprised of the heavy ions in the center-of-mass frame. The system is Lorenz-contracted by the factor $\gamma_{c.m.}$, thus it is expected that the initial volume of the system is $V_0 = V_{rest}/\gamma_{c.m.}$. When the system undergoes a large expansion the initial energy distribution becomes irrelevant, and it is sufficient to assume that the initial energy density is constant over V_0:

$$e_0 = E_{c.m.}/V_0.$$

The solution of Landau's model is rather involved, so we skip it here. It can be done quasi-analytically or numerically, as shown in [1].

4.1.2 Bjorken's Model

Bjorken's model [24], like Landau's model, assumes that the colliding nuclei are highly Lorenz-contracted disks. However it is intended for even higher energies, $e_{c.m.} > 100$ AGeV. At such energies the assumption is that the projectile and target move right through each other and continue outbound without slowing down (transparency).

During the expansion stage we focus on the mid-rapidity region (the rapidity region between target and projectile rapidities). In this region the rapidity distribution has the nice feature of being invariant to Lorenz boosting in the beam-direction.

By using the Lorenz boost invariance and the disk geometry we can derive several simple results. The density of charged particles, together with the energy density, pressure, temperature etc. depends only on the proper time, τ. By using data from p+p collisions we can estimate the charged particle density and initial energy density. If the matter is thermalized at $t_0 = 1\text{fm/c}$ the initial energy density is

$$e_0 \approx \frac{1\text{GeV}}{t_0 d_0^2} \approx 1\text{Gev/fm}^3,$$

where d_0 is the average surface per nucleon. Thus the initial condition becomes $e(\tau_0) = e_0$. We can also derive an equation for the further one dimensional development that turns out to be

$$\frac{\partial e}{\partial \tau} = -\frac{e+P}{\tau}. \tag{4.1}$$

We know that in perfect fluids the total entropy never increases, that is $S^{\mu}_{,\mu} = 0$, $S^{\mu} = su^{\mu}$ where s is the entropy density in the proper frame. From this requirement we get the equation

$$\frac{\partial s}{\partial \tau} = -\frac{s}{\tau}, \tag{4.2}$$

which solves into

$$s(\tau) = s(\tau_0)\frac{\tau_0}{\tau}.$$

For the ultra-relativistic case we can use the Stefan-Boltzmann EOS, and we have $e = 3P$. We can then integrate Eq. 4.1. The solution is

$$e(\tau) = e(\tau_0)\left(\frac{\tau}{\tau_0}\right)^{-4/3}.$$

The projectile and the target can exchange color charge during their interpenetration. This leads to the creation of a chromoelectric (color-charge) field in the volume between them, often compared to the electric field between condenser plates. Due to self interaction this field stays confined in the transverse direction, and it has the geometry of a string. Like other strings it also has a tension, and the energy density due to this is about 1 GeV/fm. When these strings decay they produce quarks and gluons, but the volume occupied by the strings stays baryon free because of the baryon conservation. The strings stretch between single constituents, and if several constituents interact at the same time there will be more strings, or equivalently a stronger field.

The number of strings and entropy per rapidity unit increases with the number of hadron-hadron collisions, which again increases proportionally with the surface of the target and projectile orthogonal to the beam direction. We get the estimate:

$$\left(\frac{dS}{dy}\right)_{\mathrm{A+A}} = \frac{r_0^2 A^{2/3}\pi}{d_0^2}\left(\frac{dS}{dy}\right)_{\mathrm{p+p}}.$$

We assume that the entropy and the pion multiplicity are proportional, and from this we can find that the pion multiplicity is proportional to $A^{2/3}$. This estimate is somewhat bigger as not only the leading baryons, but also the subsequent ones contribute to the collision.

4.2 Transverse Flow, Elliptic Flow, Tilted Disk and the Third Flow Component

Collective flow patterns are produced by the pressure in the colliding matter. They have usually been decomposed into two parts: Directed transverse flow in the reaction plane, denoted v_1 and often called 'side-splash', and enhanced

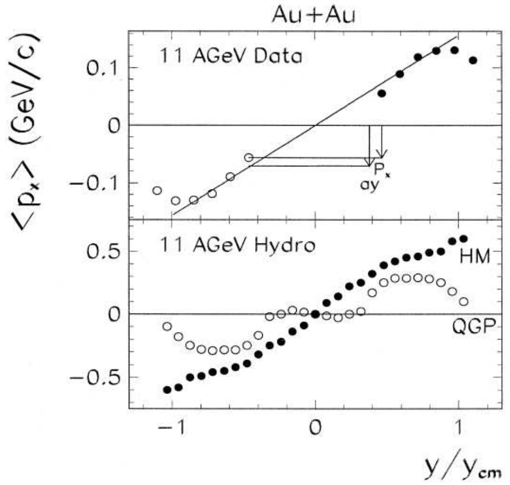

Figure 4.1: In the upper plot we notice that $P_x(y)$ deviates from the straight line behavior ay around the center-of-mass. The softening is parameterized by the quantity $S \equiv |ay - P_x(y)|/|ay|$. The lover plot compares fluid dynamical solutions with hadronic and QGP EOS. For the QGP solution we see a strong softening, especially at small rapidities. Figure reproduced from [25].

emission of particles out of the reaction plane at center of mass rapidities, the so-called squeeze-out effect.

Transverse flow patterns are a good probe of changes in the EOS. The QGP EOS is softer than the hadronic one (it can accommodate a higher density at the same pressure and temperature). This results in a pressure drop at the phase transition to QGP, the so-called 'soft point', that immediately shows up in the transverse momentum spectra. This can be seen in the frequently used p_x vs. y diagrams like Figure 4.1, where we clearly see that the transverse flow deviates from from straight line behavior.

When QGP is formed, rapid equilibration takes place, the stopping is stronger than expected and Landau's model becomes applicable. The disk will expand in the direction of the largest pressure gradient, which for central collisions is forward and backward along the beam axis. The measured rapidity spectrum will not be a fully Gaussian shape, and this can be interpreted as a bounce back effect, in accordance with the assumptions of the Landau model, rather than as transparency. There is, however, no way to distinguish the two effects from each other when collisions are central.

In non-central collisions with a small impact parameter the initial disk will be tilted, and the geometry of the expansion will be anisotropic. The direction of fastest expansion will not be aligned with the beam direction. Though it is 'in-plane', it will point in directions opposite to the usual directed transverse flow. This leads to the creation of a *third flow component* or, as it is also called, *antiflow*.

The third flow component, as first pointed out by L.P. Csernai and D. Röhrich in [25], can be seen in p_x vs. y plots as small negative values at low CM rapidities. It is a feature seen in almost all known fluid dynamical calculations that use a QGP EOS. On the other hand it is not seen in calculations with a hadronic EOS.

Pressure and transparency are independent of each other, so transparency can not explain the deviation of the flow from the beam axis. The explanation of the antiflow is that it arises from the large pressure gradient in the Lorenz contracted tilted disk. At the same time as antiflow is strengthened by the Lorenz contraction at increasing energies, the primary directed flow v_1, is weakened. Together the two flow components form the *elliptic flow*, usually denoted v_2.

As the system expands, v_2 decreases, a property that makes it especially useful as a messenger of the physical properties of the matter in the early evolution of the collision, as it is most sensitive in this region. It requires re-scattering within the matter to convert the initial spatial anisotropy into momentum anisotropy, and v_2 is therefore also a measure of the degree of thermalization in the system. Furthermore the centrality and energy dependence of v_2 are believed to be connected to the QCD phase transition, and the soft point should be observed in the collective flow patterns.

Experimental studies at the Alternating Gradient Synchrotron (AGS) and the SPS have shown that the v_2 is 'in-plane', and for pions from peripheral collisions it increases with collision energy. Fluid dynamics predict stronger elliptic flow than any other model, and the new experimental data from RHIC seem to saturate the theoretical expectations [26, 27]. Because the fluid mechanics assume local thermal equilibrium, as opposed to other models that predict weaker elliptic flow, there is reason to conclude that thermalization occurs early at today's highest experimental energies.

4.3 The Need For a Multi-Module Model

As we mentioned in the beginning of this chapter, the initial impact and relaxation time of an ultra-relativistic heavy ion collision can not be well described by a fluid dynamics. Still, fluid dynamics works very well as soon as the matter has assumed local thermal equilibrium, and we would still like to use it at that time. Because the different stages requires the use of very different physical descriptions, the best way to make a complete description of the reaction is therefore to divide the collision into distinct stages and use the most suitable model at each stage. The output of the first stage model will then serve as the input of the next and so on.

The multi-module model [28] uses a sharp three-dimensional hypersurface, $\sigma^\mu(x)$ in the space-time as the interface between the different modules. As matter crosses this surface we make sure that physical conservation laws are adhered to, and that entropy is non-decreasing.

4.4 The Effective String Rope Model

Standard hadronic string models don't apply to heavy ion collisions. There have been numerous models attempting to patch up this problem by introducing new energetic objects like 'string ropes' [30, 31], 'quark clusters' [32] and 'fused strings' [33] to describe the abundant formation of massive hadrons

The effective string rope model was developed by V. Magas, L. P. Csernai and D. D. Strottman for the description of the energy, pressure and flow velocity distributions of the initial stages of collisions. It does so in the framework of classical Yang-Mills theory. Built on the model presented in [34], basically a Bjorken model enhanced with an effective field that takes baryon recoil into account, it assumes a larger field strength than the original model, while also adhering exactly to all conservation laws. We don't describe hadronization within the framework of the model, but instead use the model to create initial conditions for a fluid dynamical solution.

Here we will just try to give a simple outline of this model for the sake of the completeness of the text. For a more detailed formulation the reader should consult [29, 28].

As usual the $[x, z]$-plane is the reaction plane and the beam is aligned with the z-direction. We create a grid in the $[x, y]$-plane. The nucleus-nucleus collision is described as the sum of the independent streak-streak collisions, corresponding to the same transverse coordinates, $\{x_i, y_i\}$. Baryon recoil for both target and projectile arises from the acceleration of partons in an effective field $F^{\mu\nu}$ created by the interaction, and considered to be an effective Abelian field, but the most important qualities of non-Abelian fields, such as self interaction, are still reflected in the model. The field is described by a phenomenological parameter with a value that is determined from experiment.

When streaks cross each other they create a *string rope* that has a *string tension* σ. We assume that the first touch of two streaks occurs at the time $t = 0$, when the string tension is zero. At the time $t = t_0$ the streaks have completely interpenetrated each other, and the string tension will have reached a constant value of of around 10GeV/fm.

Within each streak we form one flux tube between the target and projectile, with uniform σ throughout its length. This is a one-dimensional object, and only the length is allowed to change during the evolution. The target and projectile start moving away from each other at the time $t = t_0$ and the flux tube's energy increases linearly with the expansion. This energy is acquired by slowing down the projectile and target until they stop and start accelerating backward, at the time $t = t_{turn}$, making the system act like an oscillator. This is obviously a departure from the real behavior of colliding nuclei, and we have to make a few assumptions to solve the problem.

As the model has no inherent hadronization mechanism, there is no way the strings can dispose of the energy from the string tension through the natural channels of decay and breakup.

We assume that after stopping the string's expansion, and after its decay, the result is one streak of length $\triangle l_f$ with homogeneous distribution of energy density, e_f, and baryon charge, ρ_f. The final values for the energy density, baryon density and rapidity should be determined from conservation laws.

We said that the streak is moving like one object with a rapidity y_f, but quite interestingly this rapidity may differ from the rapidity of the center-of-mass of the system. If all of the elements of the system move with the same rapidity in one reference frame, then the center-of-mass also moves with that rapidity. However upon transforming to another reference system the different parts of the system may not necessarily move with the same rapidity anymore. This is a problem we see whenever pressure is present.

Thus the assumption of the final streak moving like one object oversimplifies the situation. This can be improved by letting the final streak expand with the rapidity field.

4.4.1 Construction of Initial State for Fluid Mechanics

In [28] a code was used for computing the initial conditions to be used by our fluid dynamical code from the effective string rope model. The computations were performed for an Au+Au collision at $\epsilon_0 = 65$AGeV. The results are also presented in [29]. The end result is given when the expansion of the system stops, and the matter is a locally equilibrated disk of QGP matter. This is assumed to occur at a time τ, that should be larger than the time of the final streak formation.

In Fig. 3 of [29] we are presented the results with an impact parameter of $b = 0.5(r_1 + r_2)$ where $(r_1 + r_2) = R_{au}$. The resulting initial state for the PIC-code turns out to be a disk tilted at about a 45°. Thus, the the direction of fastest expansion deviates from the beam axis, but stays in the reaction plane. This opens up the possibility of observing a third flow component, as discussed in Section 4.2.

Fig. 4 of [29] presents the same system, but with a smaller impact parameter of $b = 0.25(r_1 + r_2)$. This yields a larger energy density, while the disk is still tilted, although less than it was with the larger impact parameter. The peak of the energy distribution reaches $E_{max} \approx 50 - 90\mathrm{GeV/fm}^3$ for these more central collisions. This is more than predicted by the simple Bjorken model, but that is to be expected as the QGP volume is much smaller than the cylinder of the Bjorken model.

Chapter 5

Basics of Freeze-Out

In the late stages of the hydrodynamical solution matter becomes dilute and interactions less frequent. Artifacts of the particular solution algorithm we use may start to manifest themselves, and it becomes appropriate and necessary to start treating the matter as a gas rather than as a fluid. The process where particles stop interacting and reach the detectors can be modeled in kinetic transport theory and use of drain terms in fluid mechanics or by using the simpler *freeze-out* (FO) description.

Here we will only look at freeze-out. The review in this chapter is based on [36, 37, 38, 39, 40].

The basic idea of FO is that the matter passes through a layer in space-time where its physical properties undergo a change. During the *thermal* FO elastic collisions in the matter come to an end. This can be understood as a phase transition from fluid to ideal (noninteracting) gas.

The layer where the FO takes place may be defined either as a three dimensional hypersurface in four-space or as a full fledged four-dimensional space-time domain. The former definition corresponds to a sudden freeze out without any internal dynamics, while the latter is used to describe a supposedly more realistic gradual FO.

The hypersurface is the simplest description. This surface, called the FO surface, is characterized by its normal four-vector $d\sigma_\mu$. In addition to the specification of the FO surface we need to know the pre FO physical quantities. The transition between the pre FO- and and post FO-states is shock-like, and is described by the equations of the Raleigh line and the Taub adiabat from Section 2.3. Since the thickness of the front is zero there is no need for a kinetic description of the FO process.

For a more realistic model the FO-layer should have a thickness on the order of a mean free path, where the matter gradually changes its properties through particle scattering internal to this layer, making a kinetic description of the FO itself a necessity. The construction of a model then becomes a more complicated undertaking.

There is also another kind of freeze-out that is frequently discussed in the literature; the *chemical* freeze-out, in which inelastic collisions among particles cease, in effect freezing the abundances of the different hadronic species. Chemical FO may occur before or at the same time as the thermal FO; hydrodynamical models usually assume a simultaneous chemical and thermal FO, even though physical reasoning tells us that the chemical FO should generally happen at a higher temperature than thermal FO, as elastic collisions generally require more energy than inelastic ones. Both chemical and thermal FO may also possibly happen at different temperatures for different hadronic species, but we do not take that into consideration here.

5.1 Conservation Laws Across a Discontinuity

In the following discussion we are making these assumptions: *(i)* The FO surface is a zero-thickness three-dimensional hypersurface in the four dimensional space-time, *(ii)* the pre FO matter is in local thermal equilibrium so that all macroscopic parameters are defined by its EOS, and *(iii)* the post FO matter is an ideal gas, but not necessarily an equilibrated one.

Based on our detailed knowledge of the matter pre FO, provided by the solution of the fluid mechanics, we should now be able to use conservation laws and an assumed microscopic post FO distribution to determine post FO parameters. We can then go on and compute measurables like flow patterns, one- and two-particle spectra.

We know the invariant number of conserved particles crossing an element $d\sigma_\mu$ on the FO surface is

$$dN = N^\mu d\sigma_\mu. \tag{5.1}$$

By performing a surface integral we see that the total number of particles crossing the FO surface is:

$$N = \int_S N^\mu d\sigma_\mu.$$

Inserting the kinetic definition Eq. 2.8 of N^μ,

$$N^\mu = \int \frac{d^3p}{p^0} p^\mu f_{FO}(x,p),$$

into Eq. 5.1, we obtain the Cooper-Frye formula [35]:

$$E\frac{dN}{d^3p} = \int f_{FO}(x,p) p^\mu d\sigma_\mu, \tag{5.2}$$

that gives us the particle distribution in momentum space. Here f_{FO} is the post FO phase space distribution.

Knowing the pre FO baryon current N_0^μ, and energy-momentum tensor $T_0^{\mu\nu}$ we can calculate the post FO quantities with the help of conservation laws. The baryon current and energy-momentum tensor change in a discontinuous manner over the FO surface, so the conservation laws are expressed as

$$[N^\mu d\sigma_\mu] = 0, \tag{5.3}$$

and

$$[T^{\mu\nu} d\sigma_\mu] = 0, \tag{5.4}$$

which are identical to the relativistic Rankine-Hugoniot equations Eq. 2.29 and 2.30.

Earlier FO treatments often ignored these conservation laws, but they are very important in turning the often significant latent heat of the pre FO state into observable collective effects in the post FO matter. Due to these laws we may for example observe that flow velocity is higher in the post FO state than in the pre FO state.

In addition to the conservation laws above we must also check that the entropy is non-decreasing in the FO-process,

$$[S^\mu d\sigma_\mu] \geq 0 \qquad \text{or} \qquad \frac{S^{(1)\mu} d\sigma_\mu}{S^{(0)\mu} d\sigma_\mu} \geq 1, \tag{5.5}$$

where for all distributions the entropy is given by the formula:

$$S^\mu = -\int \frac{d^3p}{p^0} p^\mu f_{FO}(x,p)[\ln\{(2\pi)^3 f(x,p)\} - 1]. \tag{5.6}$$

This check acts as a safeguard against unphysical solutions.

5.2 Post-FO Phase Space Distributions

5.2.1 The Jüttner Distribution

We prefer to use well known distributions for the post FO state. The Jüttner distribution [41, 42],

$$f^{Juttner}(p) = \frac{1}{(2\pi\hbar)^3} \exp\left(\frac{\mu - p^\mu u_\mu}{T}\right), \tag{5.7}$$

is the relativistic generalization of the Maxwell–Boltzmann distribution. Here μ is the chemical potential, which can be determined by normalization of the distribution function.

In ultra-relativistic collisions it is reasonable to treat the post FO matter as a relativistic ideal Boltzmann gas by using the Jüttner distribution. However this approach has a flaw when combined with a zero-thickness FO-layer.

5.2.2 The Cut Jüttner Distribution

The FO surface may have either time-like or space-like normal vectors. In the literature there is some confusion about these terms, but here we will use the convention that a time-like FO surface is one that has a time-like normal four-vector. For a FO surface with a time-like normal, $d\sigma_\mu d\sigma^\mu = (d\sigma)^2$, corresponding to a sudden change in the matter properties in a finite volume, we always have

$$p^\mu d\sigma_\mu > 0,$$

as both p^μ and $d\sigma_\mu$ are time-like vectors. For describing the FO-process over the part of the surface with a time-like normal we can use the methods for detonation and shock fronts, and the equation of the Raleigh line, Eq. 2.31, will always yield a real current.

If $d\sigma_\mu$ is space-like, $d\sigma_\mu d\sigma^\mu = -(d\sigma)^2$, corresponding to a propagating discontinuity, we note that p^μ may point in both the post- and pre FO directions. Thus $p^\mu d\sigma_\mu$ can now be both positive and negative. This implies that with a thermal post FO distribution there may exist particles that are forbidden to

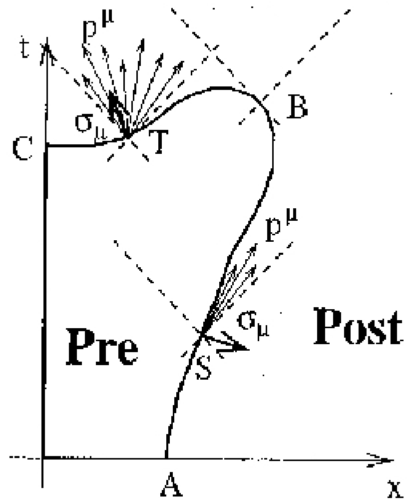

Figure 5.1: Flow across the FO surface. Notice that in the space-like domain it is possible that the flow is directed into the pre FO domain. Reproduced from [44]

freeze out through the surface element $d\sigma_\mu$, and imaginary currents results from applying the equation of the Raleigh line. This is illustrated in Fig. 5.1.

In general the FO surface has both time-like and space-like parts. A completely time-like FO can only be assumed in the case of a very rapid phase transition from supercooled QGP, so it is necessary to do a closer analysis of the space-like FO.

When integrating Eq. 5.2 a negative contribution can be interpreted as a current directed back into the pre FO region. On the pre FO side $p^\mu d\sigma_\mu$ may have both signs because p^μ is unrestricted as we assume thermal equilibrium. If the FO-layer is of finite thickness we can interpret the negative contributions as particles scattering back into the pre FO domain driven by the internal dynamics of the FO-layer. However, such back scatterings cannot occur in a zero-width FO surface as there are no particle interactions in the post FO domain.

A solution, first suggested in [43], is to specify a simple modification to the Jüttner distribution describing the post FO state, namely:

$$f^*_{FO}(x,p,d\sigma^\mu) = f^{Juttner}(x,p)\Theta(p^\mu d\sigma_\mu) = \frac{\Theta(p^\mu d\sigma_\mu)}{(2\pi\hbar)^3}\exp\left(\frac{\mu - p^\mu u_\mu}{T}\right), \quad (5.8)$$

where $\Theta(p^\mu d\sigma_\mu)$ is the sign function. This new distribution excludes all the negative momentum contributions, at the cost of producing a sharp cut in the otherwise Gaussian shape of the distribution.

5.2.3 An Improvement Upon the Cut Jüttner Distribution

K. Tamošiūnas and L.P. Csernai have proposed an improvement of the cut Jüttner distribution in [45]. The basic goal is to eliminate the sharply cut distribution and replace it with with a smoother one that still keeps the advantages of the cut Jüttner. The motivation for doing this is that sharp cuts are in general seldom seen in physics, or in nature for that matter, and they are therefore considered somewhat unphysical when appearing in a model.

The way to smooth out the cut is to take the standard Jüttner distribution and subtracting from it another Jüttner distribution with negative velocity in the *rest frame of the front* (RFF):

$$f_{new} = f_R^{Juttner} - f_L^{Juttner} = \frac{1}{(2\pi\hbar)^3}\left(\exp\frac{\mu - p^\mu u_\mu^R}{T} - \exp\frac{\mu - p^\mu u_\mu^L}{T}\right)\Theta(p^\mu d\sigma_\mu) \tag{5.9}$$

where $u_\mu^R = (\gamma, \gamma v, 0, 0)$ and $u_\mu^L = (\gamma, -\gamma v, 0, 0)$ are taken in the RFF.

5.3 Some Important Reference Frames

The FO-front can move in the positive or negative x-direction depending on the reference frame, or it can be transformed into its own local rest frame, the *rest frame of the front* (RFF). For a space-like surface $d\sigma_\mu = (0,1,0,0)d\sigma$ in the RFF. In other reference frames $d\sigma_\mu = \gamma(v,1,0,0)d\sigma$, where $\gamma = 1/\sqrt{1-v^2}$. The velocity parameter $v = d\sigma_0/d\sigma_x = -t^x/t^0$ is frame dependent and can take on both positive and negative values; to use it we have to select a given reference frame.

Another important reference frame is the one co-moving with the peak of the post FO invariant momentum distribution: the *rest frame of the gas*(RFG). This should not be confused with the local rest frame of the post FO matter since the post FO phase space distribution is not spherically symmetric in momentum space. We define v in the RFG frame as

$$v \equiv \left.\frac{d\sigma_0}{d\sigma_x}\right|_{\mathrm{RFG}}.$$

We can also obtain the same value of v by taking the velocity of the peak of the post FO distribution, u^μ_{RFG}, in the RFF. We then get:

$$v = \left.\frac{u^x_{\mathrm{RFG}}}{u^0_{\mathrm{RFG}}}\right|_{\mathrm{RFF}},$$

and this means that in the RFF the peak four-velocity has the form $u^\mu_{\mathrm{RFG}} = \gamma(1,v,0,0)|_{\mathrm{RFF}}$.

5.4 Conserved Currents for Cut Jüttner Distribution

Consider the case where f_{FO} is a cut Jüttner distribution (Eq. 5.8). We can then obtain several important quantities:

The Baryon Current: The baryon four-current N^μ can be evaluated by inserting the cut Jüttner distribution into the definition Eq. 2.8. We obtain a time-component N^0 as well as a space-component N^x. Performing the calculation in the post FO RFG frame we get

$$\begin{aligned}
N^0 &= \frac{\tilde{n}}{4}\left\{vA + a^2 j[(1+j)K_2(a) - \mathcal{K}_2(a,b)] + j\frac{b^3v^3}{3}e^{-b}\right\} \\
&\xrightarrow{m=0} \tilde{n}(\mu,T)\frac{v+1}{2}, \\
N^x &= \frac{\tilde{n}}{8}[(1-v^2)A - a^2e^{-b}] \\
&\xrightarrow{m=0} \tilde{n}(\mu,T)\frac{1-v^2}{4},
\end{aligned} \tag{5.10}$$

where $j = \mathrm{sign}(v)$, $\tilde{n} = 8\pi T^3 e^{\mu/T}(2\pi\hbar)^{-3}$, $a = m/T$, so that $\tilde{n}(\mu,T) = \tilde{n}a^2K_2(a)/2$ is the invariant scalar density of the symmetric Jüttner gas, $b = a/\sqrt{1-v^2}$, $v = d\sigma_0/d\sigma_x$, $A = (2+2b+b^2)e^{-b}$ and

$$\mathcal{K}_n(z,w) \equiv \frac{2^n(n)!}{2n!}z^{-n}\int_w^\infty dx(x^2-z^2)^{n-1/2}e^{-x}, \tag{5.11}$$

i.e. , $\mathcal{K}_n(z,z) = K_n(z)$.

The baryon current may be Lorenz transformed into the Eckart local rest frame (ELR) of the post FO matter, which moves with velocity $u^\mu = N^\mu/(N^\nu N_\nu)^{1/2} = \gamma_E(1,v_E,0,0)|_{\mathrm{RFG}}$ in the RFG or into the RFF, where $d\sigma_\mu = (0,1,0,0)d\sigma$, and the velocity of the RFG is $u^\mu_{E,\mathrm{RFG}} = \gamma(1,v,0,0)|_{\mathrm{RFF}}$. Then the Eckart flow velocity of the matter represented by the cut Jüttner distribution viewed from the RFF is $u^\mu_E = \gamma_c(1,v_c,0,0)|_{\mathrm{RFF}}$, where $v_c = (v+v_E)/(1+vv_E)$.

The proper density, which is the density in the ELR, is

$$n(\mu,T,v) = \sqrt{N^\nu N_\nu}. \tag{5.12}$$

The proper density of the cut Jüttner distribution is reduced compared to the proper density of the full Jüttner distribution.

The Energy-Momentum Tensor: In the post FO RFG the energy-momentum tensor has the components:

$$\begin{aligned}
T^{00} &= \frac{3\tilde{n}T}{2}\left\{\frac{ja^2}{2}\left\{(1+j)\left[K_2(a) + \frac{a}{3}K_1(a)\right]\right.\right. \\
&\quad \left.\left. - \left[\mathcal{K}_2(a,b) + \frac{a}{3}\mathcal{K}_1(a,b)\right]\right\} + Bv\right\}, \\
T^{0x} &= \frac{3\tilde{n}T}{4}\left\{(1-v^2)B - \frac{a^2}{6}(b+1)e^{-b}\right\}, \\
T^{xx} &= \frac{\tilde{n}T}{2}\left\{j\frac{a^2}{2}[(1+j)K_2(a) - \mathcal{K}_2(a,b)] + v^3B\right\}, \\
T^{zz} = T^{yy} &= \frac{3\tilde{n}T}{4}\left\{v\left(1-\frac{v^2}{3}\right)B + \frac{ja^2}{3}[(1+j)K_2(a)\right. \\
&\quad \left. - \mathcal{K}_2(a,b)] - \frac{va^2}{6}(b+1)e^{-b}\right\},
\end{aligned} \tag{5.13}$$

where $B = (1 + b + b^2/2 + b^3/6)e^{-b}$. We can transform this energy-momentum tensor into the Landau local rest frame (LLR) of the post FO matter. The LLR moves with velocity u_L^μ in the RFG. We can also transform it into the RFF, where $d\sigma_\mu = (0, 1, 0, 0)d\sigma$. Both of the flow velocities u_L^μ and u_E^μ (Landau and Eckart respectively) can be transformed to the frame where we want to evaluate the conservation laws Eq. 5.3 and Eq. 5.4. The parameters of the post FO distribution can then be determined so that they satisfy the conservation law.

When mass is zero the components of the energy-momentum tensor in the RFG can be simplified:

$$T^{00} = 3\tilde{n}T(v+1)/2, \qquad T^{0x} = 3\tilde{n}T(1-v^2)/4,$$
$$T^{xx} = \tilde{n}T(v^3+1)/2, \qquad T^{zz} = T^{yy} = (T^{00} - T^{xx})/2.$$

The Entropy Current In the RFG frame the entropy current becomes

$$\begin{aligned} S^0 &= \frac{\tilde{n}}{4}\left\{\left(1-\frac{\mu}{T}\right)vA + 6vB + \left(1-\frac{\mu}{T}\right)a^2 j[(1+j)K_2(a)\right. \\ &\quad \left. -\mathcal{K}_2(a,b)] + ja^2[(1+j)K_1(a) - \mathcal{K}_1(a,b)]\right\} \\ &\overset{m=0}{\longrightarrow} \frac{\tilde{n}(\mu,T)}{2}(v+1)\left(4-\frac{\mu}{T}\right), \\ S^x &= \frac{\tilde{n}}{8}\left[(1-v^2)\left(1-\frac{\mu}{T}\right)A + 6(1-v^2)B - a^2\left(2+b-\frac{\mu}{T}\right)e^{-b}\right] \\ &\overset{m=0}{\longrightarrow} \frac{\tilde{n}(\mu,T)}{4}(1-v^2)\left(4-\frac{\mu}{T}\right). \end{aligned} \tag{5.14}$$

Note that in the $m = 0$ limit the vectors S^μ and N^μ are parallel to each other.

5.5 Analytical Solution of the Freeze-Out Problem

When solving the freeze-out problem we do not assume that the flow and the normal of the FO surface are parallel. This means that we have to take into consideration the different possible directions of these two quantities.

We label the pre FO side by '0' and the post FO side '1'. In the LR frame $u^\mu = (1, 0, 0, 0)$, and we can choose the x-direction in this frame to point into the FO-direction, so that $d\sigma_\mu = \gamma_0(v_0, 1, 0, 0)d\sigma_\mu$. Assuming we know the FO hypersurface we also know v_0, and we know the three parameters $N_0^\mu d\sigma_\mu$, $T_0^{0\mu} d\sigma_\mu$ and $T_0^{x\mu} d\sigma_\mu$.

We need these values in the post FO RFG. The spatial direction of the gas flow will not change because we assume that the FO front is isotropic in its own $[y, z]$ plane. Thus, in the RFG the peak flow parameter is $u^\mu_{\mathrm{RFG}} = (1, 0, 0, 0)|_{\mathrm{RFG}}$, and the FO normal is $n_\mu = \gamma(v, 1, 0, 0)$. Note that $v \neq v_0$. The parameter v determines the post FO peak flow parameter in the RFF, $u^\mu_{\mathrm{RFG}} = \gamma(1, v, 0, 0)d\sigma|_{\mathrm{RFF}}$ (where $d\sigma_\mu = (0, 1, 0, 0)d\sigma_\mu|_{\mathrm{RFF}}$).

The conservation laws Eq. 5.3 and Eq. 5.4 give us three non-vanishing equations in the RFG,

$$[N^\mu d\sigma_\mu] = 0, \qquad [T^{0\mu} d\sigma_\mu] = 0, \qquad [T^{x\mu} d\sigma_\mu] = 0.$$

These can be used to determine the unknown parameters of the post FO cut Jüttner distribution; v, T and μ. The first equation is an invariant scalar, but the other two are components of a four-vector and should be transformed into the RFG. We need to evaluate both the pre FO and post FO quantities in the same reference frame, so the pre FO quantities must be transformed into the RFG as well. We can determine these by using the fluid dynamical form of $T^{\mu\nu}$, Eq. 2.10, seen from the RFG. From this frame the pre FO flow velocity is given by the difference of the pre FO and post FO flow velocities; $u_0^\mu = \gamma_{0R}(1, v_{0R}, 0, 0)|_{\mathrm{RFG}}$, where $v_{0R} = (v_0 - v)/(1 - v_0 v)$.

In the $m = 0$ limit an analytical solution is possible. The continuity equation leads to the equation

$$Q^{-2}(v+1)^3 + v - 1 = 0, \tag{5.15}$$

where $Q^{-1} = Q_1^{-1}(\mu, T) = \tilde{n}(\mu, T)/(4 n_0 \gamma_0 v_0)$, which gives us a third order equation and can be solved analytically for v. The energy equation $[T^{0\mu} d\sigma_\mu] = 0$ leads to the same equation with a different coefficient: $Q^{-1} = Q_2^{-1}(\mu, T) = 3T\tilde{n}(\mu, T)/(4 e_0 \gamma_0 v_0)$.

These two equations can only have the same unique solution for v if the coefficients Q_1 and Q_2 are equal, and we get

$$T = \frac{1}{3}\frac{e_0}{n_0}. \tag{5.16}$$

The solutions of both third order equations yield the same expression:

$$v = v_{3rd}(\mu) = Q^{2/3}\{[1 + \sqrt{1 + Q^2/27}]^{1/3} + [1 - \sqrt{1 + Q^2/27}]^{1/3}\} - 1. \tag{5.17}$$

By dividing $[T^{0\mu} d\sigma_\mu] = 0$ by $[T^{x\mu} d\sigma_\mu] = 0$ we obtain another third order equation for v:

$$R_0 v^3 + 3v^2 + 3(2 - R_0)v + 3 - 3R_0 = 0 \tag{5.18}$$

Here $R_0 = e_0 v_0 / p_0$. The analytical solution of this equation yields three roots, but only one of them is physically meaningful: $v = 2 - 3/R_0$. Now we can obtain μ by inserting v into Eq. 5.15.

The possibility of this simple analytical solution is a consequence of the fact that in the $m = 0$ limit the cut of the Jüttner distribution is made along central cones in the RFG, which then distribute the energy and the baryon charge in exactly the same proportions.

5.5.1 Example of the Solution

A small example solution was worked out in [37]. The pre FO state was in local equilibrium and described by the the bag-model EOS, Eq. 3.1, so that all pre FO parameters was known, while the post FO state was a massless cut Jüttner gas.

Physical solutions exist only for positive initial velocities v_0. Since we are using the cut Jüttner distribution the post FO baryon current is obviously always positive in the RFF (otherwise there would be no FO), and therefore the pre FO current and v_0 should also be positive.

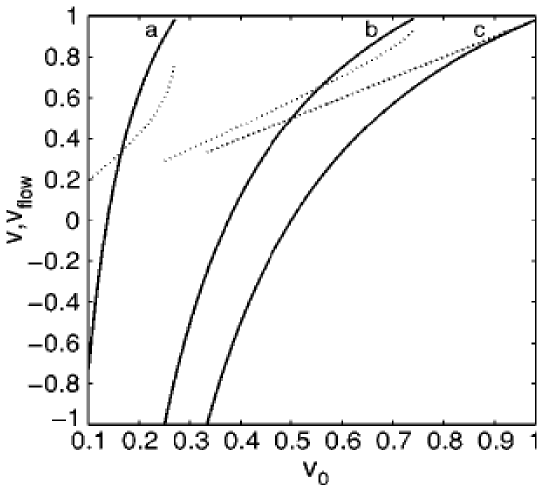

Figure 5.2: Change of velocities in freeze-out from QGP to cut Jüttner gas. The final velocity parameters(full lines) are plotted versus the initial flow velocity of the measured in RFF for case (a) $n_0 = 1.2\mathrm{fm}^{-3}, T_0 = 60\mathrm{MeV}, \Lambda_B = 225\mathrm{MeV}$, (b) $n_0 = 0.1\mathrm{fm}^{-3}, T_0 = 60\mathrm{MeV}, \Lambda_B = 80\mathrm{MeV}$ and (c) $n_0 = 1.2\mathrm{fm}^{-3}, T_0 = 60\mathrm{MeV}, \Lambda_B = 0\mathrm{MeV}$. Reproduced from [37].

From Fig. 5.2 we see that the the pre FO and post FO flow velocities differs. For small initial velocities the post FO velocities a approach zero, but for moderate velocities like $v = 0.3$–0.7, the difference between pre FO and post FO flow velocities may be significant. With $v_0 = 0.2$ the post FO flow velocity increases to $v_{\mathrm{flow}} = 0.4$ while the post FO parameter velocity increases to $v = 0.6$. These velocity changes are due to the conservation laws Eq. 2.22 and Eq. 2.21 converting the latent heat of the QGP into kinetic energy. The latent heat itself stems from the bag constant.

In connection to the flow velocity increase, the baryon density decreases in the FO as seen in Fig. 5.3.

As mentioned in Section 5.1 only the solutions that lead to an increase in entropy can be considered physical. In Fig. 5.4 the ratio of post FO and pre FO entropy currents are plotted for three different cases.

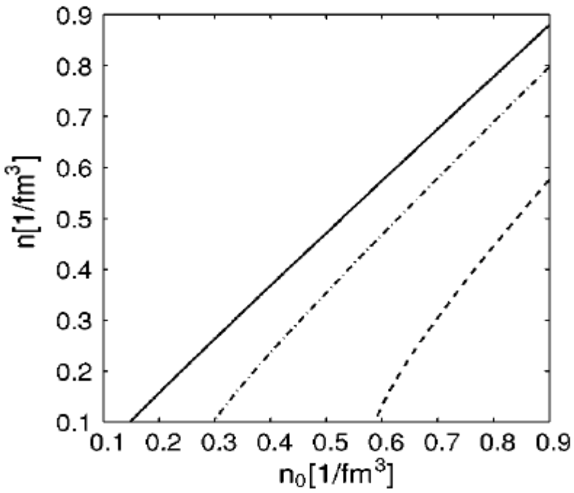

Figure 5.3: Final baryon density, n, as a function of initial baryon density in QGP to massless cut Jüttner gas. The baryon density decreases for the cases $v_0 = 0.5, T_0 = 50$MeV, (a) $\Lambda_B = 80$MeV (full line), (b) $\Lambda_B = 120$MeV (dash dotted line) and (c)$\Lambda_B = 160$MeV (dashed line). Reproduced from [37].

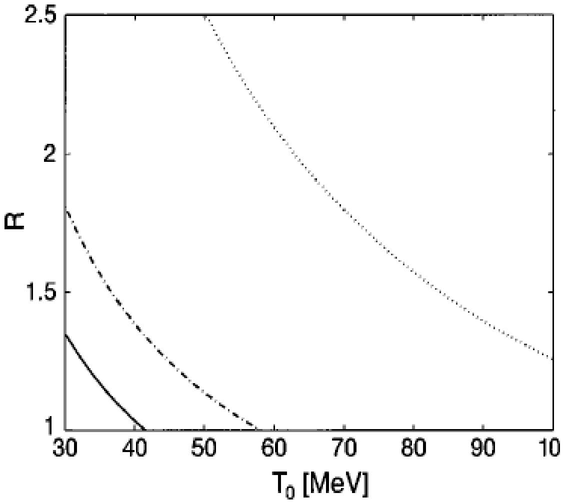

Figure 5.4: Ratio of post FO and pre FO entropy currents transverse to the FO front. Freeze-out can only be physically realized if this ratio is larger than unity. The three cases in the figure are (a) $n_0 = 0.1fm^3$, $v_0 = 0.5$, $\Lambda_B = 80$MeV (full line), (b) $n_0 = 0.5fm^3$, $v_0 = 0.5$, $\Lambda_B = 80$MeV (dashed line), and (c) $n_0 = 1.2fm^3$, $v_0 = 0.5$, $\Lambda_B = 225$MeV (dotted line). Reproduced from [37].

Chapter 6

Freeze-Out in Numerical Fluid Dynamics

In the previous chapter's analysis we assumed all the time that the FO surface was known. We now turn to the problem of defining the FO surface itself, along with the task of implementing FO in a three-dimensional numerical solution.

Previous calculations often didn't determine the FO surface explicitly. Instead one used the assumption that the normal of the FO surface is parallel to the flow, $d\sigma^\mu = u^\mu$. There is no physical justification for such an assumption, and it does in fact lead to an unrealistic surface. In numerical calculations this approach gives a ragged surface, because it is joined by vertical surfaces at cell borders.

Our goal in this chapter is to dispose of this assumption and determine the FO surface, explicitly for the PIC hydrodynamical solution.

6.1 Freeze-Out Criterion

6.1.1 Isotherm Freeze-Out

An important consideration of any practical FO-scheme is the criterion that decides whether or not the matter freezes out. This is the core of the problem of deciding where to place the FO hypersurface in space-time in relation to the matter, and there is more than one way around this problem.

From a kinetic point of view the FO should take place when the mean free path is larger than the characteristic size of the system or when the local expansion rate out-paces the rescattering rate. However these criteria deal with information not contained in the hydrodynamical model, so a kinetic description of the FO would have to be developed.

The output of the PIC code is formatted as a list of the fluid cells in the late stages of the computation. This list is repeated for several selected timesteps so that we can also follow the time development of the fluid. For each fluid cell we are given the density, energy density, pressure, entropy, temperature and flow velocity in each of the three spatial directions in in the CM frame. By iterating through the cells in space and time we can select cells for freeze-out.

Probably the simplest criterion, and certainly a very popular one, is to introduce a critical temperature T_c, so that the FO occurs over an isothermal surface, σ^μ, where $T(\sigma^\mu) = T_c$. Since we know the temperature of all the fluid cells we can easily implement an algorithm that searches through all cells and chooses those that are as near as possible to this critical temperature as candidates for freeze-out. When we know which cells that are freezing out we have in effect determined $\sigma_\mu(x)$.

The chosen temperature should, of course, reflect experimental reality, the critical temperature can be extracted from experiment by combining information about the spectrum of a single particle species and its Bose-Einstein correlations (see for example [46]). Values at CERN are in the range $T_c \sim 100 - 140$MeV.

Unfortunately, the assumption of an isothermal FO is not very accurate, and it is also quite ill fit for our particular hydrodynamical results as the temperature exhibits large variations from cell to cell and in the central region the temperature never gets low enough to warrant a FO decision based on a temperature criterion. If we are to use this criterion we will have to supplement it with other assumptions.

6.1.2 The Appearance of Negative Pressure

Studying the output of the hydrodynamical code one notices that late in the evolution negative pressures start showing up in the outermost cells of the system, forming a 'halo' surrounding the system, as seen in Fig. 6.1.

This phenomenon is not unphysical, but rather an effect of the bag constant used in the EOS, Eq. 3.1, that produces negative pressure at low temperatures. The negative pressure has the effect of pulling the matter together, producing a 'skin effect', and it leads to clustering at the edge of the system.

Negative pressure is also indicative of supercooling. So if we want the freeze-out process to model a first order (slow) phase transition (Section 3.2.4) we should preferably stop the hydrodynamical evolution before negative pressure starts to appear.

6.1.3 Simultaneous Chemical and Thermal FO?

It has been observed [47] that at the hadronic composition of the final state is governed by the requirement that $< E > / < N >= 1$GeV, that is the average energy per hadron should be approximately 1GeV in the rest frame of the produced system, irrespective of its other features.

Interestingly enough this rule fits remarkably well with data of hadronic abundances from both SPS, AGS and GSI/SIS, even though the beam energy between these facilities varies from 200AGeV to 1AGeV and nuclei of different elements have been used. If we assume simultaneous chemical and thermal freeze-out, we can implement this rule in an algorithm for computing the FO surface. This would probably be a better choice than using a simple critical temperature.

The simultaneous FO scenario may be the simplest, although it may not always yield entirely accurate results. Some consequences of introducing separate thermal and chemical freeze-outs are discussed in [48]: In heavy ion collisions we expect that distinct critical temperatures for chemical and thermal FO should exist because inelastic collisions generally need higher CM energy than elastic

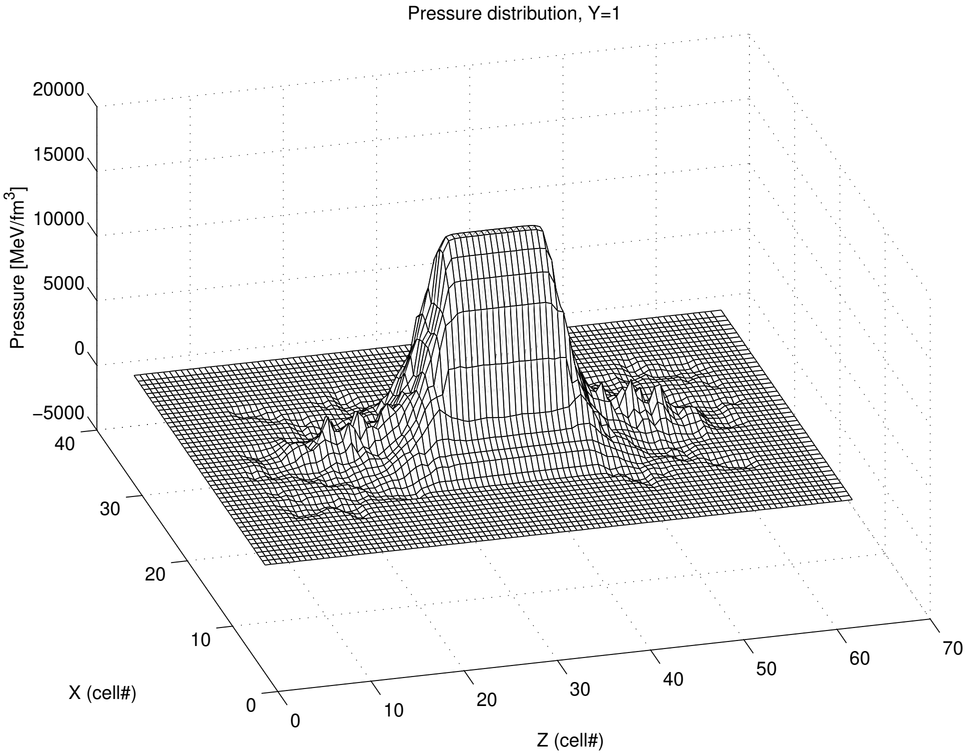

Figure 6.1: Graphical representation of the hydrodynamical pressure, taken at at a single timestep and plotted as a function of the x and z coordinates for a two dimensional 'slice' at the center of the system. The surroundings are normalized to zero pressure in this figure, and we can clearly see the negative pressure as a dip at the edge of the matter. Also notice that the pressure peaks very high in the central parts of the system.

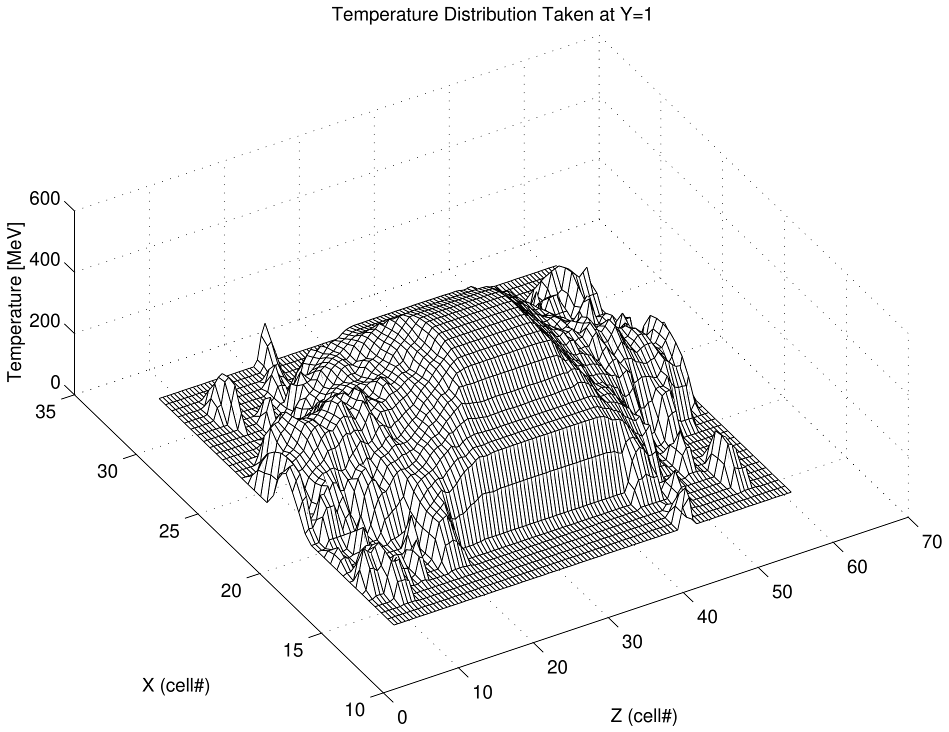

Figure 6.2: A typical pre FO temperature distribution for a two-dimensional slice in the $[x, z]$-plane taken at the center of the system. Significant temperature fluctuations are clearly visible around the edges along with several 'islands' of matter not connected to the main system. We also note that the cells in the central region have temperatures in the range of 300-400 MeV, much higher than the typical experimentally determined critical temperatures of $T_c \sim 100 - 140$MeV.

collisions. As mentioned earlier, hydrodynamical models usually ignore this fact, but an early chemical FO can have a significant effect on the collective matter properties, and these effects become important if the difference between the two FO-temperatures is large (i.e. 110 and 180 MeV).

If we have a rapid FO from supercooled QGP, we may assume that chemical and thermal FO happens simultaneously, at a constant time hypersurface. This is a very attractive possibility as we can then defend the use of the simple, idealized FO-theory.

6.2 Determination of the Freeze-Out Surface

6.2.1 General Considerations

There are different levels of refinement available to us in the implementation of FO in the PIC code. Ideally, the FO should occur in a self-consistent way as an integral part of the hydrodynamical code, iterating over all cells at every timestep taking into account loss of matter due to the FO through the use of drain terms. The FO itself could be defined either in terms of a FO surface or, in a more sophisticated model, by the use of kinetic theory.

Both of these approaches become complicated in practice, as we would have to work out the dynamics of the phase transition, and in addition relate it to the hydrodynamical code in a correct way. It would be necessary to keep track of the future evolution of cells undergoing FO and thereby removal of matter, as they could be refilled with matter by the hydrodynamical evolution later on.

In cells with a space-like FO matter could rescatter into the pre FO domain during the FO itself, and propagate in time to undergo a second FO. In [21] there is further discussion of the difficulties of implementing a self consistent freeze out, and on the limitations of the FO approach itself. The most important problem is that the assumed local equilibrium is not a solution of Boltzmann transport theory, and the FO through a hypersurface can not be adjusted to give the same results consistent with a kinetic description.

Given all these theoretical and practical problems, the FO-algorithm could be very complicated. So, in trying to keep things simple, we give up on the idea of making the FO self-consistent, at least in the current version of the model.

The PIC code solution fluctuates a good deal on a cell-by-cell basis, so that two neighboring cells can have very different values of temperature, pressure and other physical quantities. There are also sometimes gaps or islands in the matter distribution, as can be seen in Figure 6.2. There are also thin tails of matter sticking out at places, apparently caused by artifacts in the initial state, as discussed in [28]. We also observe that cells at at the edges of the system are relatively cold and frequently shows negative pressure, while at the same time the more central cells have temperatures of several hundred MeV.

Given this material it is difficult, without making simplifying assumptions, to come up with a simple physical FO-criterion that yields a reasonably smooth FO surface.

In the general case the FO surface has both time-like and space-like regions. As explained in Chapter 5 the theoretical framework used for description of the FO differs between these two regions. As we have seen above, the numerical hydrodynamical solution tends to fluctuate, and this would cause a FO surface constructed from the uncritical use of a simple FO-criterion to fluctuate between being time-like and space-like, making the calculations complicated.

To simplify the situation we make the requirement that the FO surface should have a time-like normal everywhere. Then there is no causal connection between neighboring cells on the surface, but they are still connected through the initial conditions, so they should freeze out across the same FO surface.

This assumption is really only applicable for a rapid FO in a large system. Still, the results from the PIC code indicates that the fluctuations appear mostly at the edges, while the more central region that account for the largest fraction of the matter appear to have less variations in temperature and pressure from cell to cell. When neighboring cells have approximately the same physical characteristics we can assume they should undergo FO at approximately the same time, yielding a time-like FO surface.

6.2.2 Constructing the Freeze-Out Surface

Based on the constraint that FO is completely time-like we developed an algorithm that, given the output of a PIC-code run, constructs a smoothly changing boundary for the four-dimensional volume where the FO takes place and then for every cell extracts the normal $d\sigma^\mu$ of the FO surface, defined in the CM frame of the system.

First, let us keep in mind that the data we have available from the hydrodynamical solution comes in the form of cells in a three-dimensional eulerian grid.

This means that every cell has up to 26 neighboring cells filled with matter. Matter moves between cells, and we generally have a new distribution at every timestep. So the task at hand is to define a FO surface for every cell in terms of this changing system.

The surface is made time-like through selecting cells for freeze-out in such a way that whenever a cell freezes out at a given timestep all its neighboring cells should freeze out no later than at the following timestep and no earlier than at the previous timestep. That means that no more than one timestep will separate the freeze out of neighboring cells.

The hypersurface is time-like as long as we have $\|d\vec{\sigma}/d\sigma^0\| < 1$, where $d\vec{\sigma} = (d\sigma^1, d\sigma^2, d\sigma^3)$. The dimensions of a cell are $dx = dy = 1.444$ fm and $dz = 0.1444$ fm. The duration of a timestep is $dt = 0.03804$ fm/c. Thus, a difference of one timestep in the time of FO among neighboring cells is well within the requirements for a time-like FO surface.

Some cells have few or no neighbors, and such cells, that most frequently appear at the exterior of the system, are forced to freeze out at the earliest opportunity. These cells are the result of artifacts of the PIC code and the initial state, and the matter contained in them generally moves away from the main body body of the system at a high speed, so there is little harm done in freezing them out early.

We also require that the hypersurface should be closed surface, surrounding the whole system. If it has any gaps matter may escape from the system without undergoing freeze-out, violating the conservation laws. To prevent gaps from forming we iterate from the border and inward. When a cell is frozen out it is removed from the system, so that we are guaranteed that a given cell only freezes out once. This means that the FO surface will be a layer of cells that always surrounds all parts of the system, shrinking as matter is removed. Thus, the cells that are selected for freeze out at a given timestep are those that fulfill the particular FO criterion that we enforce plus additional cells in the outermost layer required to close the FO hypersurface.

Using this tactic we obtain a series of FO hypersurfaces, one for each timestep, so that all the cells in the system will undergo FO eventually. Although being sudden at each cell, this FO is gradual for the system as a whole. Now we have determined when every cell freezes out and we have the FO surface defined in terms of cell positions, so we can begin finding the normal of the surface at every cell.

Now we make a third requirement, namely that the normal of the FO surface should be smoothly changing from cell to cell. We start out by taking a row of cells along a direction, for example the z-direction, and determine for every cell at which timestep itself and two or three of its neighbors on each side appear on the FO surface, determining the chords (in the $[z, t]$-plane) between the position of the cell and the positions of the neighbors we choose to compare it to. This computation is done in the real units of length and time as the axes have different scales. To counteract the fluctuating nature of the FO surface we take the average value of the chords and *redefine* the FO surface to be tangent to this average chord at every cell position. This process is repeated for all the rows of cells along each of the three spatial directions.

When we now know the tangent of the FO surface for each cell it becomes a trivial operation to compute the normal. This computation is described in Appendix A.

6.2.3 Example of the New Surface

The algorithm described in the previous section was implemented in FORTRAN and the complete code is located in Appendix A. This program is a module separate from the PIC code, working only on the final output of this. This implies that even though the FO gradually bleeds matter off the system, the hydrodynamical evolution will not be influenced by matter lost in the ongoing FO, as it would be in a more sophisticated model.

As an illustration of our new method we made an example surface. This was based on PIC code simulation of a $65 + 65$AGeV Au+Au collision with an impact parameter $b = 0.1(r_{Au})$. The initial state was provided by the effective string rope model, as discussed in Section 4.4. The hydrodynamical evolution was stopped at timestep 260, corresponding to a duration of 9.89fm/c, with a write-out of physical parameters being made for all timesteps between 250 and 260. At this time there were already some negative pressure cells showing up around the edges. We then ran the freeze-out code, starting at timestep 250. In Fig. 6.3 we have plotted the temperature distribution of this system at timestep 250, with slices taken in each of the three spatial planes.

As FO criterion we used a critical temperature $T_c = 140$MeV together with the requirement that any cell that has a negative pressure should undergo freeze-out. The FO is complete in 7-10 timesteps, corresponding to a real time duration of 0.27-0.38 fm/c.

To visualize the results we plotted some sections of the resulting FO surface. Figures 6.4 and 6.5 shows some examples of how this new FO surface looks in different planes. We notice a slight along the beam direction. This is caused by the choice of a finite impact parameter. We also see that all changes in the FO normal are smooth. The surface also has a shape in the three dimensional space. This is a side effect of it not being a flat surface in the four-dimensional space-time, but these surface normals are not identical to the true three dimensional surface normal of the volume.

6.3 Solving the Freeze-Out Problem

Given the FO surface it is possible to solve the FO-problem and evaluate measurables. This work is in progress, and is being performed by A. Nyiri. There are still unresolved problems though, so we are not yet ready to present results. The following only serves to outline the necessary steps of the solution.

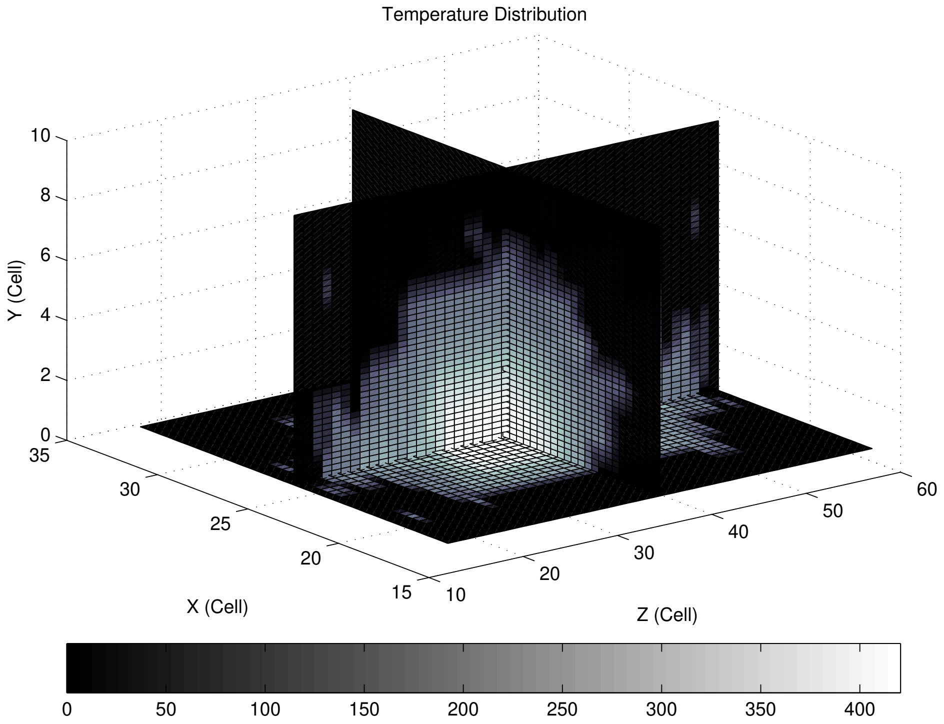

Figure 6.3: Here we have visualized the hydrodynamical system in three dimensions. The system is symmetrical around the $[x, z]$-plane, so only half of it is drawn here. We see that temperatures at the core peak at around 400, MeV while the temperatures at the edges are in the range $100 - 200$ MeV.

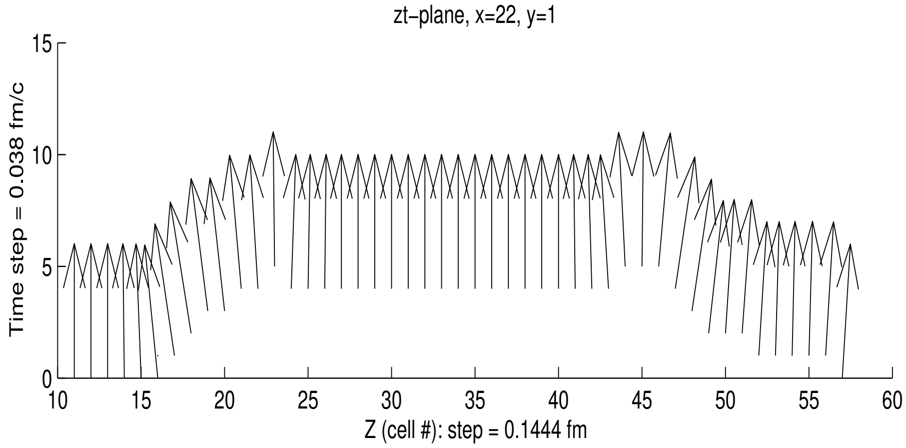

Figure 6.4: This figure shows how the FO surface looks along a row of cells along the beam direction, where cells freeze out at different timesteps. Only cells on the FO surface are drawn, represented by their normal vector. The time-components have been scaled down to better illustrate how bumps in the surface are handled, as the normal is dominated by the time component. We see that directions of the surface normal only varies slightly between neighboring cells. Also notice the slight asymmetry in the beam direction, most likely caused by the finite impact parameter

On the pre FO side, again labeled '0', the quantities provided by the hydrodynamical solution are all defined in in the computational frame, which is identical to the CM frame of the system. The density, energy density and pressure are invariant scalars, while the three flow components has to be Lorenz transformed into an appropriate reference frame. The FO surface that was computed in section 6.2.2 is also defined in the the CM frame.

The first step of the solution is to Lorenz transform the FO surface and the components of the flow velocity into the RFF. For a time-like surface the RFF is the frame where $d\sigma^\mu = (1,0,0,0)d\sigma$. To determine the state of the matter on the post FO side if the front we must solve a set of equations.

The connection between the initial and final states is given by eqs. 2.31 and 2.32,

$$j^2 = \frac{[P]}{[X]} = (n_0\gamma_0 v_0)^2 \qquad \text{and} \qquad [P] = \frac{[wX]}{(X_1 + X_0)},$$

where $X \equiv w/n^2$. The post FO thermal parameters appearing in these equations are determined by the EOS of the post FO matter. The continuity relation $[j] = 0$ can be expressed as

$$n_0\gamma_0 v_0 = n_1\gamma_1 v_1. \tag{6.1}$$

Finally outgoing entropy current must be non-decreasing across the front:

$$s_0\gamma_0 v_0 \leq s_1\gamma_1 v_1 \tag{6.2}$$

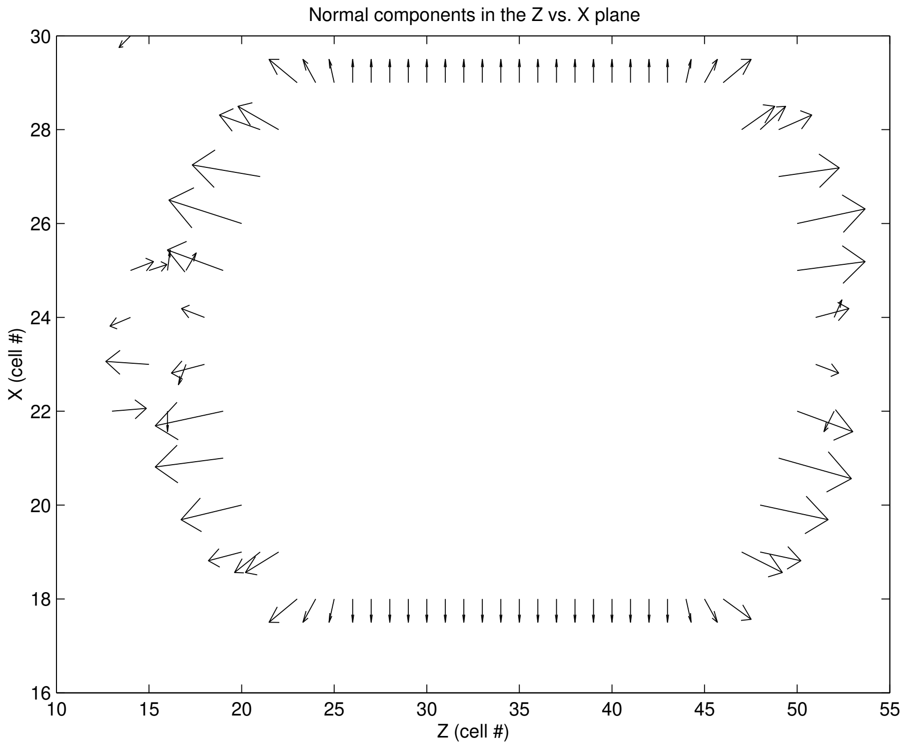

Figure 6.5: This figure illustrates the spatial shape of the FO surface, taken in the $[x, z]$-plane. The z- and x-axes use different scales, making the z-components appear to be larger than the x-components, but they are really of the same order of magnitude.

Evaluation of Measurables When the thermal parameters of the post FO state have been determined, we may evaluate measurables. These should be evaluated in the experimental detection frame, which frequently is identical to the CM frame. Therefore the post FO parameters have to be transformed back into this frame.

In [1] the definition of several measurables are presented. Unfortunately the formulae in that textbook uses the assumption that $d\sigma^\mu = u^\mu$. As mentioned in the beginning of this chapter this assumption is unjustified. The modification of the formulae for the case of a more general $d\sigma^\mu$ is straightforward, a least in principle.

Rapidity Distribution The most frequently evaluated single particle measurable is the the rapidity distribution, defined as

$$\frac{dN}{dy} = \sum_{\text{cell}} \frac{dN_{\text{cell}}}{dy}, \tag{6.3}$$

where for each cell we have

$$\frac{dN_{\text{cell}}}{dy} = \gamma V_{\text{cell}} \int \frac{d^3p}{p^0} \frac{d}{dy} p^\mu d\sigma_\mu f(x,p). \tag{6.4}$$

Here γV is the proper volume of the cell and $f(x,p)$ is the post FO particle distribution.

Transverse Momentum Distribution The transverse momentum distribution is another frequently measured quantity. The contribution of each cell to the baryon transverse momentum distribution is

$$\frac{dN_{\text{cell}}}{p_\perp dp_\perp} = \gamma V_{\text{cell}} \int \frac{d^3p}{p^0} \frac{d}{p_\perp dp_\perp} p^\mu d\sigma_\mu f(x,p) \tag{6.5}$$

The solutions of these integrals that were presented in [1] can not be used for our more general FO surface, so we have to perform a new solution to these integrals, but the new solutions wil not differ very much from the old ones.

Two Particle Interferometry Having detemined the FO surface normal, we can also evaluate the outcome of two particle correlation measurements. These are are defined in a standard way [49] based on the 'source function' $S(x,\vec{k})$, which is a single particle distribution that contains the space-time position of each emitted particle. We can assume that each particle is emitted on the FO hypersurface, so $S(x,\vec{k})$ can be approximated as a source on a sharp hypersurface However the two particle correlations are sensitive to the space-time configuration of emission. Thus, this approximation is probably not the best for this purpose as the space-time evolution of our new FO surface is rather artificial.

Long Range Multi-Particle Correlations Flow patterns are also collective correlation observables. Their determination are usually based on the single particle observables, and we can therefore probably use the hypersurface approximation of the FO with satisfying results.

In conclusion the determination of the freeze-out hypersurface and its normal vector is an important contribution to the accurate evaluation of many measurables in heavy ion physics, and in the multi module model in particular, as this model now for the first time has a realistic freeze-out description.

Chapter 7

Conclusions

In this text we have looked at fluid mechanical modeling of relativistic heavy ion collisions with an emphasis on the FO-process, and especially the determination of the FO surface. The particular fluid mechanical model we used was the PIC code, introduced in Chapter 3, an efficient three dimensional numerical solution algorithm for perfect relativistic fluid dynamics.

In Chapter 4 we explained the need for a separate model of the pre equilibrium state that exists at the very beginning of collisions. The concept of the multi-module model was introduced, where the pre equilibrium initial stage, the local equilibrium intermediate stage and the dilute statistical final stage each get their own model and are interfaced through hypersurfaces.

The effective string rope-model was presented and used to compute initial conditions for the PIC code. The initial condition turned out to be a tilted disc of matter for non-central collisions. We also discussed flow patterns, and their importance as messengers from the early stage of the collision, and argued that equilibration happens very quickly in today's ultra-relativistic experiments.

Chapter 5 reviewed the FO-process, where the continuum description of the matter is translated into a description in terms of microscopic particle distributions, while in the process all physical conservation laws are adhered to. Special emphasis was given to recent advances in the analytical treatment of space-like freeze-out. It was demonstrated that the flow velocity of the matter may change significantly in the phase transition as a result of latent heat in the QGP being converted into kinetic energy in the hadronic phase. This underlines the importance of a correct freeze-out treatment.

We then turned to implementing FO in the numerical fluid code in Chapter 6. The determination of the FO surface, σ^μ, is the main original result presented in this text, and it is an important part of a joint effort to make a precise and reliable description of measurables. Previous results often assumed that the FO surface normal $d\sigma^\mu$ was simply parallel to the flow velocity u^μ at every cell, making the FO surface ragged as there is no joining of the surface between neighboring cells. Compared to such surfaces our new surface is clearly an improvement.

There are several difficulties to the task of determining the FO surface, such as the choice of a reasonable global FO-criterion and the need to treat both space-like and time-like FO. The possibility of post FO-matter rescattering into the pre FO phase and then freezing out at a later time should also be avoided.

Fundamental theoretical obstacles are also also present, such as the need for a kinetic description of the FO-dynamics.

Because of all these difficulties we placed constraints on how our FO surface should be allowed to behave. This allowed us to obtain a surface that allows the whole system to bleed off matter and freeze-out gradually while each cell undergoes a sudden FO-process that can be treated by integrating the Cooper-Frye formula cell-by-cell.

For the first time we are able to evaluate measurables from the multi-module model through a custom tailored FO surface. Earlier results from the multi-module model (two-module model in [28]) were able to reproduce essential qualitative features of QGP, like antiflow, but had to ignore the effects a detailed FO has on the flow patterns, like thermal smearing and flow changes due to the phase transition.

While the new surface has a nice and smooth shape without any sharp edges or discontinuities it is probably not too realistic in other respects. The assumption of a purely time-like FO is a strong approximation even for large systems, and the the FO also becomes rather arbitrary as the matter is forced to freeze out even if it doesn't fulfill the FO-criteria. The FO is also likely to fail in reproducing two-particle spectra as these are highly dependent on the space-time evolution of the source. So in conclusion the current FO surface should only be viewed as a first step toward realistic FO-treatment, and work should go on to construct a more sophisticated algorithm. This work could also hopefully be aided by improvements in the other modules.

Appendix A

Computer Code (FORTRAN)

This is the computer code that was used to generate the freeze-out-normal. In addition to computing the FO surface normal for every cell it also contains functionality for exporting data to plotting programs, and it can also produce simple visualizations of the matter distribution and of the FO surface.

The Formula for Determination of the Normal Components The program initializes the normal components to $d\sigma^\mu = (1, 0, 0, 0)$. The spatial components of are then computed as

$$d\sigma^1_{\text{cell}} = \sin\left[\arctan\left(< \frac{\triangle t}{\triangle x} >_{\text{cell}} \right)\right], \tag{A.1}$$

$$d\sigma^2_{\text{cell}} = \sin\left[\arctan\left(< \frac{\triangle t}{\triangle y} >_{\text{cell}} \right)\right], \tag{A.2}$$

$$d\sigma^3_{\text{cell}} = \sin\left[\arctan\left(< \frac{\triangle t}{\triangle z} >_{\text{cell}} \right)\right], \tag{A.3}$$

where $< \frac{\triangle t}{\triangle x} >_{\text{cell}}$ is the average of the chords along the x-direction between the given cell and its neighbors on the FO surface, and correspondingly for the other spatial directions. The motivation for using arctan to determine the angle is that, for reasons of simplicity, $\triangle x, \triangle y$ and $\triangle z$ are not explicitly assigned as variables in the program.

The time directed component of $d\sigma^\mu$ is computed in an iterative manner, where it is modified in three steps, one for each of the spatial components:

$$d\sigma^0_{\text{new,cell}} = d\sigma^0_{\text{old,cell}} \cdot \sin\left(\frac{\arccos(d\sigma^1)}{d\sigma^0_{\text{old,cell}}}\right), \tag{A.4}$$

and correspondingly for $d\sigma^2$ and $d\sigma^3$.

```
c
c     Input:
c
c     unit 10, table output from hydro calculation.
c     contains density, pressure, energy density, temperature, entropy
c     and velocities for all the fluid cells at specific times late in
c     the hydro evolution.
c
c     Output:
c
c     fo_plot: contains various simple ASCII-plots meant to give a graphical
c              representation of the FO-surface.
c
c     fo_data: a new table containg cell data from unit 10, but only for
c              those cells that are involved in the freeze out. In addition
c              it contains the normal of the freeze out surface for all the cells.
c
c     fo_normal: contains a listing of the four normal components for every cell
c
c     xz_temp.dat: temperature data that are to be used by matlab
c              in plotting and such
c
c     xz_press.dat: pressure data that are to be used by matlab in plotting
c
c     -----
c
      program fo

      integer ncyc, ncyc2, keqost, m, n, l, kcell,
     x     ltype, nindex, cycles
     x     , findmaxk, findmink, findmini, findmaxi,
     x     findmaxj, findminj, itimes, jtimes, ktimes, neighbors
c
c     the following contain neighbor-info for cells on the outer surface
c
     x     nextkimin, lastkimin, nextkimax, lastkimax,
     x     nextjimin, lastjimin, nextjimax, lastjimax,
     x     nextjkmin, lastjkmin, nextkjmax, lastjkmax,
     x     nextkjmin, lastkjmin, nextjkmax, lastkjmax,
     x     jminvalue, jmaxvalue, iminvalue, imaxvalue,
     x     kminvalue, kmaxvalue

      real rlmbda, B, totn, fdnst, fentrp, fpress, fvx,
     x     fvy, fvz, ftemp, maxtmp, frzout, dx, dy, dz, dt,ednst, slope,
     x     slope1, slope2, y_comp, z_comp, t_comp, x_comp, xzaspectr,
     x     ztaspectr

      logical occup, isfo, removed
c
c     Assumed critical temperature of the FO
c

      parameter(frzout=140.0)

c
c     Here are the parameters that decides the size of the FO-system.
c     They should be set to reflect the size of the hydro system, otherwise
c     the program might crash.
c
      parameter(isize=45)
      parameter(jsize=15)
      parameter(ksize=85)
      parameter(nsize=7)
c
c     Cell sizes in [fm] from the the hydro code!
c

      parameter(dx=1.444)
      parameter(dy=1.444)
      parameter(dz=0.144438)
c
c     Size of the timestep, in [fm/c]
c

      parameter(dt=0.03804)
      dimension  kcell(isize, jsize, ksize), ltype(isize, jsize,ksize,
     x     nsize), cycles(nsize), nextkimin(jsize, ksize, nsize),
     x     lastkimin(jsize, ksize, nsize), nextkimax(
     x     jsize,ksize, nsize), lastkimax(jsize, ksize, nsize),
     x     nextjimin(jsize, ksize, nsize), lastjimin(
     x     jsize,ksize, nsize), nextjimax(jsize, ksize, nsize),
     x     lastjimax(jsize, ksize, nsize), nextjkmin(isize, jsize,
     x     nsize), lastjkmin(isize, jsize, nsize), nextjkmax(
     x     isize,jsize, nsize), lastjkmax(isize, jsize, nsize),
     x     nextkjmin(isize, ksize,
     x     nsize), lastkjmin(isize, ksize, nsize), nextkjmax(
     x     isize,ksize, nsize), lastkjmax(isize, jsize, nsize),
     x     itimes(isize), jtimes(jsize), ktimes(ksize),
```

```
     x     jminvalue(isize), jmaxvalue(isize), kminvalue(isize),
     x     kmaxvalue(isize), iminvalue(jsize), imaxvalue(jsize)

      dimension fdnst(isize, jsize, ksize, nsize), fentrp(isize,
     x     jsize, ksize, nsize), ednst(isize, jsize, ksize, nsize),
     x     fpress(isize, jsize, ksize, nsize), t_comp(isize, jsize,
     x     ksize, nsize),
     x     fvx(isize, jsize, ksize, nsize),fvy(isize, jsize, ksize,
     x     nsize),fvz(isize, jsize, ksize, nsize),ftemp(isize,
     x     jsize, ksize, nsize),
     x     x_comp(isize, jsize,
     x     ksize, nsize), y_comp(isize, jsize, ksize, nsize),
     x     z_comp(isize, jsize, ksize, nsize)

c
c     occup   - cell is occupied by matter
c     isfo    - cell is frozen out
c     removed - matter in cell has been removed from the system
c               after the FO
c
      dimension occup(isize, jsize, ksize, nsize), isfo(isize, jsize,
     x     ksize, nsize), removed(isize, jsize, ksize, nsize)

c     Initialization of the logicals
c
      do 40 i=1, isize
      do 50 j=1, jsize
      do 60 k=1, ksize
      do 70 n=1, nsize
         occup(i, j, k, n) = .false.
         isfo(i, j, k, n) = .false.
         removed(i, j, k, n) = .false.
 70   enddo
 60   enddo
 50   enddo
 40   enddo

c
c     Aspect ratios between time and space axes
c
      xtaspectr=dt/dx
      ytaspectr=dt/dy
      ztaspectr=dt/dz

      open (unit=10, file='fort.10', status='old')
      open (unit=11, file='fo_plot.txt', status='unknown')
      open (unit=12, file='fo_data.txt', status='unknown')
      open (unit=13, file='xz_temp.dat', status='unknown')
      open (unit=14, file='xz_press.dat', status='unknown')
      open (unit=15, file='fo_normal.txt', status='unknown')
      open (unit=16, file='xyz_temp.dat', status='unknown')
      open (unit=17, file='xyz_press.dat', status='unknown')


c
c     Reading from unit 10 using 3 nested do loops
c
      write(*, *) 'Reading from fort.10 ...'

      nindex=0
      do 300 m=1, nsize
      read(10, 135, err=300, end=300) totn, keqost
 135  format(14x, f8.3, 11x, i3)
      nindex = nindex+1  ! This variable increases by one for each cycle


      do 280 l=1,3
      read(10, 145, end=280, err=280) ncyc, ncyc2
 145  format(// 8x, i3, i2)
      cycles(nindex) = ncyc

      do 250 n=1, 10000
         read(10, 100, end=300, err=280) i, j, k, ijk, aednst,
     x        dnst, press, entrp, vx, vy, vz, temp
 100     format(4x, 3i3, i6, f9.4, f9.5, 2f9.3, 1x, 3f7.4, 4x, f5.1)

c----------------------  error checking --------------------------


         if(i.gt.isize) then
            write(12, *) 'i to big'
            go to 999  ! the end
         endif
         if(j.gt.jsize) then
```

```
            write(12, *) 'j too big'
            go to 999
         endif
         if(k.gt.ksize) then
            write(12, *) 'k too big'
            go to 999
         endif

c---------------------------------------------------------------------
c     Here we choose the cells that are frozen out and place them in
c     the spatial grid at the same positions as in the hydro
c     calculation. Later on we will decide which cells are crossed by
c     the fo-surface.
c---------------------------------------------------------------------

         occup(i, j, k, nindex) = .true.
c
c     the following line is the main FO-criterion. It can be changed
c     to accomodate any needs, but cells will also be frozen out even
c     when they dont fulfill this criterion if they happen to be on the
c     outermost layer in the three dimensional system
c
         if ((press .le. 0.0) .or. (temp .lt. frzout)) then
            isfo(i, j, k, nindex) = .true.
         endif
         kcell(i, j, k) = ijk
         ednst(i, j, k, nindex) = aednst
         fdnst(i, j, k, nindex) = dnst
         fpress(i, j, k, nindex) = press
         fentrp(i, j, k, nindex) = entrp
         fvx(i, j, k, nindex) = vx
         fvy(i, j, k, nindex) = vy
         fvz(i, j, k, nindex) = vz
         ftemp(i, j, k, nindex) = temp
         ltype(i, j, k, nindex) = ncyc2

 250  end do                    ! inner
 280  end do  ! middle
      read(10, 135) totn, keqost
 300  end do  ! outer

c$$$      call matlabxz(occup, removed, isfo, 1, 1, isize, jsize
c$$$     x     , ksize, nsize, ftemp, fdnst)
c----------------------------------------------------------------

      write(*, *) 'Analyzing data ...'
c
c
c     Here we decide when cells should freeze out and be
c     removed from the system. The main requirement is to avoid gaps
c     in the surface in both time and space, this means that
c
c     1) if a cell freezes out all its neighbors should also
c     freeze out within the following timestep and
c
c     2) there should be no holes in the three-dimensional hypersurface,
c     otherwise matter could escape from the system without undergoing FO
c
c----------------------------------------------------------------



c
c     But first we eliminate cells that has three neighbors or less. Such cells
c     frequently appear in tails sticking out of the main body of the
c     system, and we freeze them out and remove them at the first timestep.
c
      do 340 i=2, isize-1
      do 350 j=2, jsize-1
      do 360 k=2, ksize-1

         if (neighbors(i, j, k, 1, occup, isize,
     x        jsize, ksize, nsize) .le. 3) then
            isfo(i, j, k, 1) = .true.
            removed(i, j, k, 1) = .true.
         endif
 360   enddo
 350   enddo
 340   enddo

      do 700 n=1, nsize
      do 600 j=1, jsize
c
```

```
c     Finding the the minimum and maximum k-indices for the given xz-slice
c
      kmin = ksize
      kmax = 0
      do 400, i=1, isize
         newkmax = findmaxk(i, j, n, occup, isize, jsize, ksize, nsize)
         newkmin = findmink(i, j, n, occup, isize, jsize, ksize, nsize)
         if(newkmin .lt. kmin) kmin = newkmin
         if(newkmax .gt. kmax) kmax = newkmax
 400  enddo

      lastkimax(j, kmin, n) = 0
      lastkimin(j, kmin, n) = ksize

      do 450 k=kmin, kmax
c
c     Finding minimum and maximum i for the current and the subsequent k-vector
c
         imin = isize
         imax = 0
         imax = findmaxi(j, k, n, occup, isize, jsize, ksize, nsize)
         imin = findmini(j, k, n, occup, isize, jsize, ksize, nsize)
         nextkimax(j, k, n)=findmaxi(j, k+1, n, occup, isize,jsize,
     x         ksize, nsize)
         nextkimin(j, k, n)=findmini(j, k+1, n, occup, isize,jsize,
     x        ksize, nsize)

c
c     Finding minimum and maximum j for the current xy-slice
c
         jmin = jsize
         jmax = 0
         do 405, i=1, isize
         newjmax = findmaxj(i, k, n, occup, isize, jsize, ksize, nsize)
         newjmin = findminj(i, k, n, occup, isize, jsize, ksize, nsize)
         if(newjmin .lt. jmin) jmin = newjmin
         if(newjmax .gt. jmax) jmax = newjmax
         nextkjmax(i, k, n)=findmaxi(j, k+1, n, occup, isize,jsize,
     x         ksize, nsize)
         nextkjmin(i, k, n)=findmini(j, k+1, n, occup, isize,jsize,
     x        ksize, nsize)
 405     enddo

         lastjimax(j+1, k, n) = imax
         lastjimin(j+1, k, n) = imin
         nextjimax(j, k, n) = findmaxi(j+1, k, n, occup,
     x        isize,jsize, ksize, nsize)
         nextjimin(j, k, n) = findmini(j+1, k, n, occup,
     x           isize,jsize, ksize, nsize)


c
c     Forcing FO at the ends
c
         if(lastkimin(j, k, n) .gt. lastkimax(j, k, n)) then
            do 420 ii=imin, imax
               kk = findmink(ii,j, n, occup, isize, jsize, ksize, nsize)
               isfo(ii, j, kk, n) = .true.
               removed(ii, j, kk, n) = .true.
               kminvalue(ii) = kk
               ltype(ii, j, kk, n) = 2
 420        enddo
         else if(nextkimin(j, k, n) .gt. nextkimax(j, k, n)) then
            do 425 ii=imin, imax
               kk = findmaxk(ii,j, n, occup, isize, jsize, ksize, nsize)
               isfo(ii, j, kk, n) = .true.
               removed(ii, j, kk, n) = .true.
               kmaxvalue(ii) = kk
               ltype(ii, j, kk, n) = 2
 425        enddo
         endif

         do 435 ii=imin, imax
            jj = findmaxj(ii, k,n, occup, isize, jsize, ksize, nsize)
            isfo(ii, jj, k, n) = .true.
            removed(ii, jj, k, n) = .true.
            jmaxvalue(ii) = jj
            lastkjmax(ii, k+1, n) = jj
            ltype(ii, jj, k, n) = 2
 435     enddo
c
c     Force FO and removal for outermost cells
c
```

```
           removed(imax, j, k, n) = .true.
           removed(imin, j, k, n) = .true.
           isfo(imax, j, k, n) = .true.
           isfo(imin, j, k, n) = .true.
           ltype(i, j, k, n) = 2
c
c     Preparing for next iteration
c
           lastkimax(j, k+1, n) = imax
           lastkimin(j, k+1, n) = imin
           lastjimax(i, j+1, k) = imax
           lastjimin(i, j+1, k) = imin

           do 440 i1=1, isize/2
           if ((imax-i1) .lt. (imin+i1)) go to 450
c
c     FO and remove interior cells that meet FO-criterion
c
           if (isfo(imax-i1, j, k, n)) removed(imax-i1, j, k, n) = .true.
           if (isfo(imin+i1, j, k, n)) removed(imin+i1, j, k, n) = .true.

c
c    Here we make forced FO and removal to eliminate gaps in the hypersurface
c

c
c     First we check for k vs. i
c
           if (nextkimax(j, k, n) .gt. nextkimin(j, k, n)
     x        .and. lastkimax(j, k, n) .gt. lastkimin(j, k, n)) then

           if (imax .gt. nextkimax(j, k, n)
     x        .and.  .not. isfo(imax-i1, j, k, n)
     x        .and. imax-i1-nextkimax(j, k, n) .ge. 1) then
              isfo(imax-i1, j, k, n) = .true.
              removed(imax-i1, j, k, n) = .true.
           endif
           if(imax .gt. lastkimax(j, k, n)
     x        .and. .not. isfo(imax-i1, j, k, n)
     x        .and. imax-i1-lastkimax(j, k, n) .ge. 1) then
              isfo(imax-i1, j, k, n) = .true.
              removed(imax-i1, j, k, n) = .true.
           endif

           if (imin .lt. nextkimin(j, k, n)
     x        .and. .not. isfo(imin+i1, j, k, n)
     x        .and. nextkimin(j, k, n)-imin-i1 .ge. 1) then
              isfo(imin+i1, j, k, n) = .true.
              removed(imin+i1, j, k, n) = .true.
           endif
           if(imin .lt. lastkimin(j, k, n)
     x        .and. .not. isfo(imin+i1, j, k, n)
     x        .and. lastkimin(j, k, n)-imin-i1 .ge. 1) then
              isfo(imin+i1, j, k, n) = .true.
              removed(imin+i1, j, k, n) = .true.
           endif
           endif

           if (nextjimax(j, k, n) .gt. nextjimin(j, k, n)
     x        .and. lastjimax(j,k,n) .gt. lastjimin(j, k,n)) then
c
c     Here we check for j vs. i , if we find a gap it will be closed
c

           if (imax .gt. nextjimax(j, k, n)
     x        .and.   .not. isfo(imax-i1, j, k, n)
     x        .and. imax-i1-nextjimax(j, k, n) .ge. 1) then
              isfo(imax-i1, j, k, n) = .true.
              removed(imax-i1, j, k, n) = .true.
           endif
           if (imax .gt. lastjimax(j, k, n)
     x        .and. .not. isfo(imax-i1, j, k, n)
     x        .and. imax-i1-lastjimax(j, k, n) .ge. 1) then
              isfo(imax-i1, j, k, n) = .true.
              removed(imax-i1, j, k, n) = .true.
           endif
           if (imin .lt. nextjimin(j, k, n)
     x        .and. .not. isfo(imin+i1, j, k, n)
     x        .and. nextjimin(j, k, n)-imin-i1 .ge. 1) then
              isfo(imin+i1, j, k, n) = .true.
              removed(imin+i1, j, k, n) = .true.
           endif
           if (imin .lt. lastjimin(j, k, n)
     x        .and. .not. isfo(imin+i1, j, k, n)
```

```
     x        .and. lastjimin(j, k, n)-imin-i1 .ge. 1) then
            isfo(imin+i1, j, k, n) = .true.
            removed(imin+i1, j, k, n) = .true.
         endif
         endif
 440  enddo
 445  continue
 450  enddo                       ! end k

      if(imin .le. imax) then
      do 550 i=imin, imax

         if (j .eq. 1) then
            lastjkmin(i, j, n) = ksize
            lastjkmax(i, j, n) = 0
         endif

         do 515 kk=kmin, kmax
            jj = findmaxj(i, kk, n, occup, isize, jsize, ksize, nsize)
            isfo(i, jj, kk, n) = .true.
            removed(i, jj, kk, n) = .true.
 515     enddo

         nextjkmax(i, j, n) = findmaxk(i, j+1, n, occup,
     x        isize,jsize, ksize, nsize)
         nextjkmin(i, j, n) = findmink(i, j+1, n, occup,
     x        isize, jsize, ksize, nsize)
         lastjkmax(i, j+1, n) = kmax
         lastjkmin(i, j+1, n) = kmin

         do 540 k1=1, ksize/2
         if ((kmax-k1) .le. (kmin+k1)) go to 550
c
c     Check for j vs. k
c
         if (nextjkmax(i, j, n) .gt. nextjkmin(i, j, n)
     x        .and. lastjkmax(i, j, n) .gt. lastjkmin(i, j, n)) then

         if ((kmax .gt. nextjkmax(i, j, n))
     x        .and.   (.not. isfo(kmax-k1, j, k, n))
     x        .and. kmax-k1-nextjkmax(i, j, n) .ge. 1) then
            isfo(kmax-i1, j, k, n) = .true.
            removed(kmax-i1, j, k, n) = .true.
         endif
         if((kmax .gt. lastjkmax(i, j, n))
     x        .and. (.not. isfo(kmax-k1, j, k, n))
     x        .and. kmax-k1-lastjkmax(i, j, n) .ge. 1) then
            isfo(kmax-k1, j, k, n) = .true.
            removed(kmax-k1, j, k, n) = .true.
         endif

         if ((kmin .lt. nextjkmin(i, j, n))
     x        .and. (.not. isfo(kmin+k1, j, k, n))
     x        .and. nextjkmin(i, j, n)-kmin-k1 .ge. 1) then
            isfo(kmin+k1, j, k, n) = .true.
            removed(kmin+k1, j, k, n) = .true.
         endif
         if((kmin .lt. lastjkmin(i, j, n))
     x        .and. (.not. isfo(kmin+k1, j, k, n))
     x        .and. lastjkmin(i, j, n)-kmin-k1 .ge. 1) then
            isfo(kmin+k1, j, k, n) = .true.
            removed(kmin+k1, j, k, n) = .true.
         endif
         endif

 540     enddo
 550     enddo
         endif
 600  enddo                       ! end j

c
c     Here we remove forever cells that are frozen out
c
      do 690 ii=1, isize
      do 680 jj=1, jsize
      do 670 kk=1, ksize
         nn=n
         if (removed(ii, jj, kk, nn)) then
            do 660 nn=n+1, nsize
               occup(ii, jj, kk, nn) = .false.
               isfo(ii, jj, kk, nn) = .false.
 660        enddo
         endif
 670  enddo
```

```
 680  enddo
 690  enddo


 700  enddo  ! end n
c---------------------------------------------------------------------
c
c     Compute the components of the normal in the z and time-direction
c     for all cells on the FO-surface
c
c---------------------------------------------------------------------
      do 760 n=1, nsize
      do 750 i=1, isize
      do 740 j=1, jsize
      do 730 k=1, ksize
         if(occup(i,j,k,n)) then
c
c     initialize the normal components
c
            t_comp(i, j, k, n) = 1.0
            x_comp(i, j, k, n) = 0.0
            y_comp(i, j, k, n) = 0.0
            z_comp(i, j, k, n) = 0.0
         endif
 730  enddo
 740  enddo
 750  enddo
 760  enddo
c
c     First make corrections of t and z components
c
      do 950 i=1, isize
      do 940 j=1, jsize

      kmin = ksize
      kmax = 0

      do 800 k=1, ksize
         ktimes(k) = 0  ! initialize the timestep vector
 800  enddo
c
c     Find the timestep where FO occurs for each cell in a row along
c     the z-axis
c
      do 840 n=1, nsize

         newkmin = findmink(i, j, n, occup, isize, jsize,ksize, nsize)
         newkmax = findmaxk(i, j, n, occup, isize, jsize,ksize, nsize)
         if(newkmax .gt. kmax) kmax = newkmax
         if(newkmin .lt. kmin) kmin = newkmin
 840  enddo

      do 850 k=1, ksize
         do 845 n=nsize, 1, -1
            if(isfo(i, j, k, n) .and. occup(i,j,k,n)) ktimes(k) = n
 845     enddo
 850  enddo
c      write(14, *) (ktimes(kk), kk=1, ksize)

      do 900 k=kmin, kmax

         if(ktimes(k) .ne. 0) then

c
c     We now compute the tangent of the FO-surface for a given cell
c

c
c     Average up to three cells on the left side
c
            if(k .eq. kmin) then
               dv1 = ztaspectr
            else if(ktimes(k-3) .ne. 0 .and. ktimes(k-2) .ne. 0 .and.
     x           ktimes(k-1) .ne. 0 .and. k .gt. kmin+2) then
               dv1=dt*((ktimes(k)-ktimes(k-1))/dz+
     x              (ktimes(k)-ktimes(k-2))/(2*dz)
     x              +(ktimes(k)-ktimes(k-3))/(3*dz))/3
            else if(ktimes(k-2) .ne. 0 .and. ktimes(k-1) .ne.0
     x              .and. k .gt. kmin+1) then
               dv1=0.5*((ktimes(k)-ktimes(k-1))/dz+
     x              (ktimes(k)-ktimes(k-2))/(2*dz))*dt
            else
               dv1 = (ktimes(k)-ktimes(k-1))/dz
            endif
```

```
c
c     .. and on the right side
c
            if(k .eq. kmax) then
               dv2 = -ztaspectr
            else if(ktimes(k+3) .ne. 0 .and. ktimes(k+2) .ne. 0 .and.
     x              ktimes(k+1) .ne. 0 .and. k .lt. kmax-2) then
               dv2=dt*((ktimes(k+1)-ktimes(k))/dz+
     x              (ktimes(k+2)-ktimes(k))/(2*dz)
     x              +(ktimes(k+3)-ktimes(k))/(3*dz))/3
            else if(ktimes(k+2) .ne. 0 .and. ktimes(k+1) .ne. 0
     x              .and. k .lt. kmax-1) then
               dv2=0.5*dt*((ktimes(k+1)-ktimes(k))/dz+
     x              (ktimes(k+2)-ktimes(k))/(2*dz))
            else
               dv2= dt*(ktimes(k+1)-ktimes(k))/dz
            endif
c
c     check for errors
c
            if (abs(dv1) .gt. 1.0) then
               dv1=0.0
            endif

            if (abs(dv2) .gt. 1.0) then
               dv2=0.0
            endif
c
c     Here we find the components of the normal vector of the FO-surface
c
         z_comp(i, j, k, ktimes(k)) = sin(atan((-dv2-dv1)/2))
         t_comp(i, j, k, ktimes(k)) = t_comp(i, j, k, ktimes(k))
     x        *sin(acos(z_comp(i, j, k, ktimes(k))))
c     x           /t_comp(i, j, k, jtimes(j))))

      endif

 900  enddo                     ! end k
 940  enddo                     ! end j
 950  enddo                     ! end i
c
c     Do the same for t and x components
c
      do 1200 j=1, jsize
      do 1190 k=1, ksize

      do 1000 i=1, isize
         itimes(i) = 0
 1000 enddo

      imin = isize
      imax = 0
c
c     Find timesteps where FO occurs
c
      do 1040 n=1, nsize
         newimin = findmini(j, k, n, occup, isize, jsize, ksize, nsize)
         newimax = findmaxi(j, k, n, occup, isize, jsize, ksize, nsize)
         if (newimax .gt. imax) imax = newimax
         if (newimin .lt. imin) imin = newimin
 1040 enddo

      do 1050 i=1, isize
         do 1045 n=nsize, 1, -1
            if(isfo(i, j, k, n) .and. occup(i,j,k,n)) itimes(i) = n
 1045    enddo
 1050 enddo

      do 1100 i=imin, imax

         if(itimes(i) .ne. 0) then
c
            if(i .eq. imin) then
               dv1 = xtaspectr
            elseif(itimes(i-2) .ne. 0 .and. itimes(i-1) .ne. 0
     x              .and. i .gt. imin+1) then
               dv1=0.5*dt*((itimes(i)-itimes(i-1))/dx+
     x              (itimes(i)-itimes(i-2))/(2*dx))
            else
               dv1 = dt*(itimes(i)-itimes(i-1))/dx
            endif

c
c     right side
```

```
c
            if(i .eq. imax) then
               dv2 = -xtaspectr
            elseif(itimes(i+2) .ne. 0 .and. itimes(i+1) .ne. 0
     x           .and. i .gt. imax-1) then
               dv2 = 0.5*dt*((itimes(i+1)-itimes(i))/dx+
     x               (itimes(i+2)-itimes(i))/(2*dx))
            else
               dv2 = dt*(itimes(i+1)-itimes(i))/dx
            endif
c
c     If we have a space like tangent there is something wrong and we just assign zero
c     value for the tangent of that cell
c
            if (abs(dv1) .gt. 1.0) then
               dv1=0.0
            endif

            if (abs(dv2) .gt. 1.0) then
               dv2=0.0
            endif

            x_comp(i, j, k, itimes(i)) = sin(atan((-dv2-dv1)/2))
            t_comp(i, j, k, itimes(i)) = t_comp(i, j, k, itimes(i))
     x            *sin(acos(x_comp(i, j, k, itimes(i))
     x            /t_comp(i, j, k, itimes(i))))
         endif

 1100 enddo
 1190 enddo
 1200 enddo

      do 1400 i=1, isize
      do 1390 k=1, ksize

      do 1215 j=1, jsize
         jtimes(j) = 0
 1215 enddo

      jmin = jsize
      jmax = 0
c
c     Find timesteps where FO occurs
c
      do 1240 n=1, nsize
         newjmin = findminj(i, k, n, occup, isize, jsize, ksize, nsize)
         newjmax = findmaxj(i, k, n, occup, isize, jsize, ksize, nsize)
         if (newjmax .gt. jmax) jmax = newjmax
         if (newjmin .lt. jmin) jmin = newjmin
 1240 enddo

      do 1250 j=1, jsize
         do 1245 n=nsize, 1, -1
            if(isfo(i, j, k, n) .and. occup(i,j,k,n)) jtimes(j)=n
 1245    enddo
 1250 enddo

      do 1300 j=jmin, jmax

         if(jtimes(j) .ne. 0) then

            if(j .eq. 1 ) then
               dv1 = 0.0
            elseif(j .eq. jmin .and. j .ne. 1) then
               dv1 = ytaspectr
            elseif(jtimes(j-3) .ne. 0 .and. itimes(j-2).ne.0
     x           .and. itimes(j-1) .ne. 0 .and. j .gt. jmin-2) then
               dv1 =dt*((jtimes(j)-jtimes(j-1))/dy+
     x             (jtimes(j)-jtimes(j-2))/(2*dy)
     x             +(jtimes(j)-jtimes(j-3))/(3*dy))/3
            elseif(j .gt. jmin+1 .and. jtimes(j-2) .ne. 0 .and.
     x             jtimes(j-1) .ne. 0) then
               dv1 = 0.5*dt*((jtimes(j)-jtimes(j-1))/dy+
     x             (jtimes(j)-jtimes(j-2))/(2*dy))
            elseif(j-1 .ge. jmin .and. j .le. jmax) then
               dv1 = dt*(jtimes(j)-jtimes(j-1))/dy
            endif

             if(j .eq. jmax) then
               dv2 = -ytaspectr
            else if(jtimes(j+3) .ne. 0 .and. jtimes(j+2) .ne. 0 .and.
     x          jtimes(j+1) .ne. 0 .and. j .lt. jmax-2) then
               dv2 = dt*((jtimes(j+1)-jtimes(j))/dy+
```

```
     x             (jtimes(j+2)-jtimes(j))/(2*dy)
     x             +(jtimes(j+3)-jtimes(j))/(3*dy))/3
            else if(jtimes(j+2) .ne. 0 .and. jtimes(j+1) .ne. 0
     x             .and. j .lt. jmax-1) then
               dv2 = 0.5*dt*((jtimes(j+1)-jtimes(j))/dy+
     x             (jtimes(j+2)-jtimes(j))/(2*dy))
            else
               dv2=dt*(jtimes(i+1)-jtimes(i))/dy
            endif


            if (abs(dv1) .gt. 1.0) then
               dv1=0.0

            endif
            if (abs(dv2) .gt. 1.0) then
               dv2=0.0
            endif

            y_comp(i, j, k, jtimes(j)) = sin(atan((-dv2-dv1)/2))
            t_comp(i, j, k, jtimes(j)) = t_comp(i, j, k, jtimes(j))
     x          *sin(acos(y_comp(i, j, k, jtimes(j))
     x          /t_comp(i, j, k, jtimes(j))))
         endif

 1300 enddo
 1390 enddo
 1400 enddo
c
c     Final Writeout starts here:
c
      write(11, 2000) frzout
      write(12, 2000) frzout
      write(11, *) 'Cellsize and timestep: '
      write(12, *) 'Cellsize and timestep: '
      write(11, 3000) dx, dy, dz, dt
      write(12, 3000) dx, dy, dz, dt


      write(12, *) 'x  y  z   t   ijk      e        n        p       s
     x      vx      vy      vz      T     type cycle    time'
      write(12, *) '                    [GeV/fm^3][1/fm^3]
     x                            [MeV]                [fm/c]    '
c      write(15, *) ' n  i  j  k     Vn        Vx        Vy         Vz
c     x    Length'

c
c     Writing the final fo_data including normal-vector to unit 12
c
      do 1880 n=1, nsize
      do 1860 i=1, isize
      do 1840 j=1, jsize
      do 1820 k=1, ksize
         if(isfo(i,j,k,n)) then
         write(12, 4000) i, j, k, n,  kcell(i,j,k), ednst(i, j, k, n),
     x     fdnst(i, j, k, n), fpress(i, j, k, n), fentrp(i, j, k, n),
     x     fvx(i, j, k, n) , fvy(i, j, k, n), fvz(i, j, k, n),
     x     ftemp(i,j,k,n), ltype(i, j, k, n), cycles(n), cycles(n)*dt
         endif
         if(isfo(i,j,k,n)) then
         write(15, 5000) n, i, j, k, t_comp(i, j, k, n),
     x      x_comp(i, j, k, n), y_comp(i, j, k, n), z_comp(i, j, k, n),
     x      t_comp(i, j, k, n)**2 + x_comp(i, j, k, n)**2
     x      +(y_comp(i, j, k, n))**2+(z_comp(i, j, k, n))**2,
     x      ftemp(i,j,k,n)
         endif
 1820  enddo
 1840  enddo
 1860  enddo
 1880  enddo


 2000 format(1x, 'FO-temperature is ', f6.1,' MeV.')
 3000 format(' dx = ', f5.3,  '   dy = ', f5.3, '  dz = ',f8.6, '
     x dt = ',f7.5, //)
 4000 format(4i3, ' ',i6, f9.4, f9.5, 2f9.3, 1x, 3f7.4, 4x, f5.1,
     x        4x, i2, 3x, i3, 3x, f8.4, f8.4, f9.6)
 5000 format(4i3, 1x, f9.6, 1x, f9.6, 1x, f9.6, 1x, f9.6, 1x, f9.6,
     x     1x, f5.1)

      write(*,*) 'done!'
 999  end                        !end of main program
```

```
c-----------------------------------------------------------------------
c
c     The following functions search for minimum or maximum occupied
c     positions for a given vector
c
c-----------------------------------------------------------------------


c
c     Minimum occupied k-index, returns ksize if nothing is found
c
      integer function findmink(i, j, n, occup, isize, jsize,
     x     ksize, nsize)
      common // mink
      integer k1, i, j, n, isize, jsize, ksize, nsize
      logical occup(isize, jsize, ksize, nsize)

      findmink=0
      do 10 k1=1, ksize
         if(occup(i, j, k1, n)) go to 20
 10   enddo
 20   continue
      findmink = k1
      return
      end

c
c     Maximum occupied k-index, returns 0 if nothing is found
c
      integer function findmaxk(i, j, n, occup, isize, jsize,
     x     ksize, nsize)
      common // maxk
      integer k1, i, j, n, isize, jsize, ksize, nsize
      logical occup(isize, jsize, ksize, nsize)

      findmaxk=0
      do 10 k1=ksize, 1, -1
         if(occup(i, j, k1, n)) go to 20
 10   enddo
 20   continue
      findmaxk = k1
      return
      end

c
c     Minimum occupied i-index, returns isize if nothing is found
c
      integer function findmini(j, k, n, occup, isize, jsize,
     x     ksize, nsize)
      integer i1, j, k, n, isize, jsize, ksize, nsize
      logical occup(isize, jsize, ksize, nsize)

      findmini=0
      do 10 i1=1, isize
         if(occup(i1, j, k, n)) go to 20
 10   enddo
 20   continue
      findmini = i1
ccc      write(*,*) 'imin', findmini
      return
      end

c
c     Maximum occupied i-index, returns 0 if nothing is found
c
      integer function findmaxi(j, k, n, occup, isize, jsize,
     x     ksize, nsize)
      integer i1, j, k, n, isize, jsize, ksize, nsize
      logical occup(isize, jsize, ksize, nsize)

      findmaxi = 0
      do 10 i1=isize, 1, -1
         if(occup(i1, j, k, n)) go to 20
 10   enddo
 20   continue
      findmaxi = i1
      return
      end
c
c     Minimum occupied j-index, returns isize if nothing is found
c
      integer function findminj(i, k, n, occup, isize, jsize,
     x     ksize, nsize)
```

```
      integer j1, i, k, n, isize, jsize, ksize, nsize
      logical occup(isize, jsize, ksize, nsize)

      findminj=0
      do 10 j1=1, jsize
         if(occup(i, j1, k, n)) go to 20
 10   enddo
 20   continue
      findminj = j1
      return
      end

c
c     Maximum occupied j-index, returns 0 if nothing is found
c
      integer function findmaxj(i, k, n, occup, isize, jsize,
     x     ksize, nsize)
      integer j1, i, k, n, isize, jsize, ksize, nsize
      logical occup(isize, jsize, ksize, nsize)

      findmaxj = 0
      do 10 j1=jsize, 1, -1
         if(occup(i, j1, k, n)) go to 20
 10   enddo
 20   continue
      findmaxj = j1
      return
      end

c
c     Determines the number of neighbors for a given cell, maximum
c     is 26
c
      integer function neighbors(i, j, k, n, occup, isize, jsize,
     x     ksize, nsize)
      integer isize, jsize, ksize, nsize
      logical occup(isize, jsize, ksize, nsize)

      neighbors = 0
      if(occup(i, j-1, k, n)) neighbors = neighbors+1
      if(occup(i, j+1, k, n)) neighbors = neighbors+1
      if(occup(i+1, j-1, k, n)) neighbors = neighbors+1
      if(occup(i+1, j, k, n)) neighbors = neighbors+1
      if(occup(i+1, j+1, k, n)) neighbors = neighbors+1
      if(occup(i-1, j-1, k, n)) neighbors = neighbors+1
      if(occup(i-1, j, k, n)) neighbors = neighbors+1
      if(occup(i-1, j+1, k, n)) neighbors = neighbors+1
      if(occup(i, j, k+1, n)) neighbors = neighbors+1
      if(occup(i, j-1, k+1, n)) neighbors = neighbors+1
      if(occup(i, j+1, k+1, n)) neighbors = neighbors+1
      if(occup(i+1, j-1, k+1, n)) neighbors = neighbors+1
      if(occup(i+1, j, k+1, n)) neighbors = neighbors+1
      if(occup(i+1, j+1, k+1, n)) neighbors = neighbors+1
      if(occup(i-1, j-1, k+1, n)) neighbors = neighbors+1
      if(occup(i-1, j, k+1, n)) neighbors = neighbors+1
      if(occup(i-1, j+1, k+1, n)) neighbors = neighbors+1
      if(occup(i, j, k-1, n)) neighbors = neighbors+1
      if(occup(i, j-1, k-1, n)) neighbors = neighbors+1
      if(occup(i, j+1, k-1, n)) neighbors = neighbors+1
      if(occup(i+1, j-1, k-1, n)) neighbors = neighbors+1
      if(occup(i+1, j, k-1, n)) neighbors = neighbors+1
      if(occup(i+1, j+1, k-1, n)) neighbors = neighbors+1
      if(occup(i-1, j-1, k-1, n)) neighbors = neighbors+1
      if(occup(i-1, j, k-1, n)) neighbors = neighbors+1
      if(occup(i-1, j+1, k-1, n)) neighbors = neighbors+1
      return
      end

c------------------------ subroutine matlabxz --------------------------------
c
c     Makes tables of temperature and pressure on a two dimensional slice
c
c-----------------------------------------------------------------------------
      subroutine matlabxz(occup, removed, isfo, j, n, isize, jsize
     x     , ksize, nsize, ftemp, fdnst)

      integer j, n, isize, jsize, ksize, nsize ,
     x     findmink, findmaxk, findmini, findmaxi
      real ftemp, fdnst
      logical isfo, occup, removed
      dimension isfo(isize, jsize, ksize, nsize), occup(isize, jsize,
     x     ksize, nsize), removed(isize, jsize, ksize, nsize),
     x     ftemp(isize, jsize, ksize, nsize), fdnst(isize, jsize,
     x     ksize, nsize)
```

```
      mink = ksize
      maxk = 0
      do 50 i=1, isize
         newmin = findmink(i ,j, n, occup, isize, jsize, ksize, nsize)
         newmax = findmaxk(i, j, n, occup, isize, jsize, ksize, nsize)
         if (newmin .lt. mink) mink = newmin
         if (newmax .gt. maxk) maxk = newmax
 50   enddo

      mini = isize
      maxi = 0
      do 90 k=1, ksize
         newmini = findmini(j, k, n, occup, isize, jsize, ksize, nsize)
         newmaxi = findmaxi(j, k, n, occup, isize, jsize, ksize, nsize)
         if(newmini .lt. mini) mini = newmini
         if(newmaxi .gt. maxi) maxi = newmaxi
 90   enddo


      do 860 i=mini, maxi
      do 820 k=mink, maxk
c
c     writing temperatures to unit 13
c
      if(i .gt. 0 .and. i .lt. isize .and.
     x        k .gt. 0 .and. k .lt. ksize) then
c$$$        if(removed(i, j, k, n)) then
c$$$           write(13, 5000) i, k, frzout
         if(occup(i, j, k, n)) then
            write(13, 5000) i, k, ftemp(i, j, k, n)
         else
            write(13, 5000) i, k, 0.0
         endif

      endif

 820  enddo
 860  enddo
      do 960 i=mini-5, maxi+5
      do 920 k=mink-5, maxk+5
c
c     writing pressure to unit 14
c
      if(i .gt. 0 .and. i .lt. isize .and.
     x        k .gt. 0 .and. k .lt. ksize) then

         if(occup(i, j, k, n)) then
            write(14, 6000) i, k, fdnst(i, j, k, n)
         else
            write(14, 6000) i, k, 0.0
         endif

      endif

 920  enddo
 960  enddo



 5000 format(2i3, 1x, f7.1)
 6000 format(2i3, 1x, f9.3)


      end


c----------------------- subroutine matlab3D-------------------------------
c
c     Makes a tables of temperature and pressure in a three dimensional region
c
c--------------------------------------------------------------------------
      subroutine matlab3D(occup, removed, isfo, n, isize, jsize
     x     , ksize, nsize, ftemp, fpress)

      integer n, isize, jsize, ksize, nsize,
     x     findmink, findmaxk, findmini, findmaxi, findminj, findmaxj
      real ftemp, fpress
      logical isfo, occup, removed
      dimension isfo(isize, jsize, ksize, nsize), occup(isize, jsize,
     x     ksize, nsize), removed(isize, jsize, ksize, nsize),
```

```
     x     ftemp(isize, jsize, ksize, nsize), fpress(isize, jsize,
     x     ksize, nsize)

      mink = ksize
      maxk = 0
      do 60 i=1, isize
      do 50 j=1, jsize
         newmin = findmink(i ,j, n, occup, isize, jsize, ksize, nsize)
         newmax = findmaxk(i, j, n, occup, isize, jsize, ksize, nsize)
         if (newmin .lt. mink) mink = newmin
         if (newmax .gt. maxk) maxk = newmax
 50   enddo
 60   enddo

      mini = isize
      maxi = 0
      do 90 k=1, ksize
      do 80  j=1, jsize
         newmini = findmini(j, k, n, occup, isize, jsize, ksize, nsize)
         newmaxi = findmaxi(j, k, n, occup, isize, jsize, ksize, nsize)
         if(newmini .lt. mini) mini = newmini
         if(newmaxi .gt. maxi) maxi = newmaxi
 80   enddo
 90   enddo

      minj = jsize
      maxj = 0
      do 120 k=1, ksize
      do 110 i=1, isize
         newminj = findminj(i, k, n, occup, isize, jsize, ksize, nsize)
         newmaxj = findmaxj(i, k, n, occup, isize, jsize, ksize, nsize)
         if(newminj .lt. minj) minj = newminj
         if(newmaxj .gt. maxj) maxj = newmaxj
 110  enddo
 120  enddo

      do 860 i=mini, maxi
      do 840 j=1, maxj
      do 820 k=mink, maxk

        if(removed(i, j, k, n)) then
            write(16, 5000) i, j, k, 0.0
         else if(occup(i, j, k, n)) then
            write(16, 5000) i, j, k, ftemp(i, j, k, n)
         else
            write(16, 5000) i, j, k, 0.0
         endif
 820  enddo
 840  enddo
 860  enddo


      do 960 i=mini, maxi
      do 940 j=1, maxj
      do 920 k=mink, maxk
c
c     writing pressure to unit 17
c
      if(i .gt. 0 .and. i .lt. isize .and. j .lt. jsize
     x        .and. k .gt. 0 .and. k .lt. ksize) then

         if(occup(i, j, k, n)) then
            write(17, 6000) i, j, k, fpress(i, j, k, n)
         else
            write(17, 6000) i, j, k, 0.0
         endif

      endif

 920  enddo
 940  enddo
 960  enddo

 5000 format(3i3, f7.1)
 6000 format(3i3, f9.3)


      end

c
c     The following subroutines produce simple ASCII plots for
c     the pupose of visual evaluation
c
```

```
c------------------------- Subroutine Plotxz -------------------------------
c
c     Makes a simple visual representation of a slice of the system
c
c     INPUT:
c     isfo, isize, jsize, ksize, nsize are the same as elsewhere
c     in the program
c
c     j, n:   decides at which y-value and timestep to take the plot
c
c-------------------------------------------------------------------------
      subroutine plotxz(occup, removed, isfo, j, n, isize, jsize
     x     , ksize, nsize,cycles, ltype, x_comp, z_comp)

      integer j, n, isize, jsize, ksize,
     x     nsize, lcells, cycles, ltype
     x     ,findmink, findmaxk, findmini, findmaxi
      real x_comp, z_comp
      logical isfo, occup, removed
      character line(1:85)

      dimension isfo(isize, jsize, ksize, nsize), occup(isize, jsize,
     x     ksize, nsize), removed(isize, jsize, ksize, nsize),
     x     ltype(isize, jsize, ksize, nsize),
     x     x_comp(isize, jsize, ksize, nsize),
     x     z_comp(isize, jsize, ksize, nsize)

      dimension cycles(nsize)

c     Search for the FO-cells with the smallest and biggest
c     k-values to minimize the sixe of the plot
c
c
      mink = ksize
      maxk = 0
      do 50, i=1, isize
         newmin = findmink(i ,j, n, occup, isize, jsize, ksize, nsize)
         newmax = findmaxk(i, j, n, occup, isize, jsize, ksize, nsize)
         if (newmin .lt. mink) mink = newmin
         if (newmax .gt. maxk) maxk = newmax
 50   enddo
      mini = isize
      maxi = 0
      do 90, k=1, ksize
         newmini = findmini(j, k, n, occup, isize, jsize, ksize, nsize)
         newmaxi = findmaxi(j, k, n, occup, isize, jsize, ksize, nsize)
         if(newmini .lt. mini) mini = newmini
         if(newmaxi .gt. maxi) maxi = newmaxi
 90   enddo

      write(11, *) ' Shows occupation of cells in the xz-plane for a cho
     xsen y and timestep'
      write(11, 100)  j, cycles(n)
 100  format( 1x, ' Y = ', i2, '   Cycle = ', i3)
      write(11, *) ' X = cell on the FO-hypersurface'
      write(11, *) ' * = cell that is occupied with matter, but is not f
     xrozen out'
      write(11, '(//)')
c
c     Writing the ASCII plot
c
      write(11, 200) (k, k=mink, maxk)
      do 130 i=maxi, mini, -1
      do 120 k=1, 85
         if(occup(i, j, k, n) .and.
     x          (.not. removed(i, j, k, n))) then
            line(k) = '*'
         else if(z_comp(i, j, k, n) .lt. 0.0 .and. x_comp(i, j, k, n)
     x        .gt. 0.0 .and. occup(i, j, k, n)) then
            line(k) = '\134'
         elseif(z_comp(i, j, k, n) .gt. 0.0 .and. x_comp(i, j, k, n)
     x          .gt. 0.0 .and. occup(i, j, k, n)) then
            line(k) = '/'
         elseif(z_comp(i, j, k, n) .eq. 0.0 .and.
     x          occup(i, j, k, n)) then
            line(k) = '|'
         elseif(x_comp(i, j, k, n) .eq. 0.0 .and.
     x          occup(i, j, k, n)) then
            line(k) = '-'
         elseif(z_comp(i, j, k, n) .lt. 0.0 .and. x_comp(i, j, k, n)
     x          .lt. 0.0 .and. occup(i, j, k, n)) then
            line(k) = '/'
```

```
          elseif(z_comp(i, j, k, n) .gt. 0.0 .and. x_comp(i, j, k, n)
     x           .lt. 0.0 .and. occup(i, j, k, n)) then
             line(k) = '\134'
          else
             line(k) = ' '
          endif
 120  enddo
      write(11, 300) i, (line(kk), kk=mink, maxk)
 130  enddo
 300  format(1x, i3, 2x,  80(A, '  '))

      write(11, 200) (k, k=mink, maxk)
 200  format(4x, 70i3)
      write(11, *) ' '

      end


c------------------------- Subroutine plotzt -------------------------------
c
c     Makes a visual represention of the FO-surface in z vs. time plots
c
c     INPUT:
c     isfo, isize, jsize, ksize, nsize and cycles are the same as elsewhere
c     in the program
c
c---------------------------------------------------------------------------


      subroutine plotzt(occup, isfo, removed, isize, jsize, ksize,
     x      nsize, j, cycles, z_comp, t_comp)

      integer i, j, k, n, mink, maxk, isize, jsize, ksize, nsize, cycles
     x     , findmink, findmaxk
      real x_comp, t_comp
      logical isfo, occup, removed
      character line(1:80)

      dimension isfo(isize, jsize, ksize, nsize), occup(isize, jsize,
     x     ksize, nsize), removed(isize, jsize, ksize, nsize),
     x     z_comp(isize, jsize, ksize, nsize), t_comp(isize, jsize,
     x     ksize, nsize)
      dimension cycles(nsize)

      do 140 i=22, 22
      write(11, *) ' '
      write(11, *) ' Plot of k vs. time for some chosen values of x and
     xy.'
      write(11, 100) i, j
 100  format(2x, 'X=', i3, '  Y =', i3)
      write(11, *) ' X = cell on the FO-hypersurface'
      write(11, *) ' * = cell that is occupied with matter, but is not f
     xrozen out'
      write(11, *) ' '
c
c     Finding the minimum and maximum occupied k-indices
c
      mink = ksize
      maxk = 0
      do 70 n=1, nsize
         newmin = findmink(i, j, n, occup, isize, jsize, ksize, nsize)
         newmax = findmaxk(i, j, n, occup, isize, jsize, ksize, nsize)
         if (newmin .lt. mink) mink = newmin
         if (newmax .gt. maxk) maxk = newmax
 70   enddo

      write(11, 200) (k, k=mink, maxk)
      do 130 n=nsize, 1, -1
      do 120 k=1, 80
         if(occup(i, j, k, n) .and.
     x           (.not. removed(i, j, k, n))) then
            line(k) = '*'
         else if(z_comp(i, j, k, n) .lt. 0.0 .and. t_comp(i, j, k, n)
     x        .gt. 0.0 .and. occup(i, j, k, n)) then
            line(k) = '\134'
         elseif(z_comp(i, j, k, n) .gt. 0.0 .and. t_comp(i, j, k, n)
     x           .gt. 0.0 .and. occup(i, j, k, n)) then
            line(k) = '/'
         elseif(z_comp(i, j, k, n) .eq. 0.0 .and.
     x           occup(i, j, k, n)) then
            line(k) = '|'
         elseif(t_comp(i, j, k, n) .eq. 0.0 .and.
     x           occup(i, j, k, n)) then
            line(k) = '-'
         elseif(z_comp(i, j, k, n) .lt. 0.0 .and. t_comp(i, j, k, n)
```

```
     x           .lt. 0.0 .and. occup(i, j, k, n)) then
            line(k) = '/'
         elseif(z_comp(i, j, k, n) .gt. 0.0 .and. t_comp(i, j, k, n)
     x           .lt. 0.0 .and. occup(i, j, k, n)) then
            line(k) = '\134'
         else
            line(k) = ' '
         endif
 120  enddo
      write(11, 300) cycles(n), (line(k), k=mink, maxk)
 130  enddo
 300  format(1x, i3, 2x,  80(A, '  '))
c
c     Writing the the k-cell numbers
c
      write(11, 200) (k, k=mink, maxk)
 200  format(4x, 70i3)
      write(11, *) ' '

 140  enddo

      end

c------------------------- Subroutine plotxt -------------------------------
c
c     Makes a visual represention of the FO-surface in x vs. time plots
c
c     INPUT:
c     isfo, isize, jsize, ksize, nsize and cycles are the same as elsewhere
c     in the program
c
c---------------------------------------------------------------------------

      subroutine plotxt(occup, isfo, removed, isize, jsize, ksize,
     x      nsize, j, cycles)

      integer mini, maxi, j, isize, jsize, ksize, nsize, cycles
     x     , findmini, findmaxi
      logical isfo, occup, removed
      character line(1:80)

      dimension isfo(isize, jsize, ksize, nsize), occup(isize, jsize,
     x     ksize, nsize), removed(isize, jsize, ksize, nsize)

      dimension cycles(nsize)
c
c     Search for the FO-cells with the smallest and biggest
c     k-values to minimize the width of the plot
c


c
c     Writing the ASCII plot, for different values of i
c
      do 140 k=20,30
      write(11, *) ' '
      write(11, *) ' Plot of x vs. time for some chosen values of k and
     xy.'
      write(11, 100) k, j
 100  format(2x, 'Z=', i3, '  Y=', i3)
      write(11, *) ' X = cell on the FO-hypersurface'
      write(11, *) ' * = cell that is occupied with matter, but is not f
     xrozen out'
      write(11, *) ' '
c
c     Finding the minimum and maximum occupied i-indices
c
      mini = isize
      maxi = 0
      do 70, n=1, nsize
         newmini = findmini(j, k, n, occup, isize, jsize, ksize, nsize)
         newmaxi = findmaxi(j, k, n, occup, isize, jsize, ksize, nsize)
         if(newmini .lt. mini) mini = newmini
         if(newmaxi .gt. maxi) maxi = newmaxi
 70   enddo

      write(11, 200) (i, i=mini, maxi)
      do 130 n=nsize, 1, -1
      do 120 i=1, isize
         if(isfo(i, j, k, n) .and. occup(i, j, k, n)) then
            line(i) = 'X'
         elseif(occup(i, j, k, n) .and.
     x           (.not. removed(i, j, k, n))) then
            line(i) = '*'
         elseif(removed(i, j, k, n)) then
```

```
              line(i) = ' '
           else
              line(i) = ' '
           endif
 120    enddo
        write(11, 300) cycles(n), (line(ii), ii=mini, maxi)
 130    enddo
 300    format(1x, i3, 2x,  50(A, '  '))
c
c      Writing the the i-cell numbers
c
       write(11, 200) (i, i=mini, maxi)
 200   format(4x, 50i3)
       write(11, *) ' '

 140   enddo
       end
```

Bibliography

[1] L.P. Csernai, "Introduction to Relativistic Heavy Ion Collisions", John Wiley & Sons(1994).

[2] L.D. Landau, E.M. Lifschitz, "The Classical Theory of Fields".

[3] L.D. Landau, E.M. Lifschitz, "Fluid Mechanics".

[4] T. S. Olson, W. A. Hiscock, Phys. Rev. **D 41**, 3687 (1990).

[5] A. H. Taub, Phys. rev. **74**, 328 (1948).

[6] F. H. Harlow, A.A. Amsden, J. R. Nix, J. Comp. Phys., **20**, 119 (1976).

[7] L.D. Landau, E.M. Lifschitz, "Statistical Physics".

[8] A. K. Holme, E. F. Staubo, L. P. Csernai, E. Osnes, D. Strottman, Phys. Rev. **D 40**, 3735 (1989).

[9] N. S. Amelin, E. F. Staubo, L. P. Csernai, V. D. Toneev, K. K. Gudima, D. Strottmann, Phys. Rev. Lett. **67**, 1523 (1991).

[10] K. Huang, "Statistical Mechanics", John Wiley & Sons (1966).

[11] L. P. Csernai, J. I. Kapusta, Phys. Rev **D 46**, 1379 (1992).

[12] L. P. Csernai, J. I. Kapusta, Phys. Rev. Lett **69**, 737 (1992).

[13] D. H. Rischke, M. Gyulassy, Nucl. Phys. **A 608**, 479 (1996).

[14] S. Soff, S. A. Bass, A. Dumitru, Phys. Rev. Lett. **86**, 3981 (2001).

[15] C. G. Kallman, Phys. lett **134B**, 363 (1984).

[16] M. I. Gorenstein, O. A. Moglievsky, Z. Phys, C **30**, 161 (1988).

[17] A. A. Amsden, A. S. Goldhaber, F. H. Harlow, J. R. Nix, Phys. Rev. **C 17**, 2080 (1978).

[18] L. P. Csernai, I. Lowas, J. Mahruhn, A. Rosenhauer, J. Zimanyi, W. Greiner, Phys. Rev. **C 26**, 149 (1982).

[19] J. Brachmann, S. Soff, A. Dumitru, H. Stöcker, J. A. Mahruhn, W. Greiner, D. H. Rische, L. Bravina, Phys. Rev. **C 61**, 024909 (2000).

[20] S. A. Bass et al. Prog. Part. Nucl. Phys. **41**, 225 (1998).

[21] D. Molnár, M. Gyulassy, Phys. Rev. **C 62**, 054907-1 (2000).

[22] L.D. Landau, Izv. Akad. Nauk SSSR, **17**, 51 (1953).

[23] L.D. Landau, S.Z. Belenkij, Usp. Phys. Nauk. **56**, 309 (1956), Nuovo Cimento Suppl, **3**, 15 (1956).

[24] J.D. Bjorken, Phys. Rev. **D 27**, 140 (1983).

[25] L.P. Csernai, D. Röhrich, Phys. Lett. **B 458** 454 (1999).

[26] STAR Collaboration, K. H. Ackerman *et al.*, Phys. Rev. Lett. **86**, 402 (2001).

[27] STAR Collaboration, C. Adler *et al.*, Phys. Rev. Lett. **87**, 182301 (2001).

[28] V. K. Magas "Multi Module Model for Ultra-Relativistic Heavy Ion Collisions", Ph. D. Thesis, University of Bergen (2001).

[29] V.K. Magas, L.P. Csernai, D.D. Strottman, Phys. Rev. **C 64** 014901 (2001).

[30] T. S. Biró, H. B. Nielsen, J. Knoll, Nucl. Phys. **B 245**, 449 (1984).

[31] H. Sorge, Phys. Rev. **C 52**, 3291 (1952).

[32] K. Werner, J. Aichelin, Phys. Rev. Lett. **76**, 1027 (1996).

[33] N. S. Amelin, M. A. Braun, C. Pajares, Phys. Lett. **B 306**, 312 (1993).

[34] M. Gyulassi, L. P. Csernai, Nucl. Phys. **A 460** 723 (1986).

[35] F. Cooper, G. Frye, Phys. Rev. **D 10**, 186 (1974).

[36] Zs. Lazar, V.K. Magas, L.P. Csernai, H. Stöcker and W. Greiner Phys. Rev. **C 59** 388 (1999).

[37] Cs. Anderlik, L.P. Csernai, F. Grassi, W. Greiner, Y. Hama, T. Kodama, Zs. Lazar, V.K. Magas and H. Stöcker, Phys. Rev. **C 59** 3309 (1999).

[38] V.K. Magas, Cs. Anderlik, L.P. Csernai, F. Grassi, W. Greiner, Y. Hama, T. Kodama, Zs. Lazar, and H. Stöcker, Heavy Ion Phys. **9** 193 (1999).

[39] Cs. Anderlik, L.P. Csernai,F. Grassi, W. Greiner, Y. Hama, T. Kodama, Zs. Lazar, V.K. Magas and H. Stöcker, Phys. Lett. **B 459** 33 (1999).

[40] V.K. Magas, Cs. Anderlik, L.P. Csernai, F. Grassi, W. Greiner, Y. Hama, T. Kodama, Zs. Lazar, and H. Stöcker, Nucl. Phys. **A 661** 596 (1999).

[41] F. Jüttner, Ann.Phys. und Chemie, **34**, 856 (1911).

[42] I. Abonyi, Acta Phys. Hungarica, **23**, 247 (1967).

[43] K.A. Bugaev, Nucl. Phys. **A 606**, 559 (1996).

[44] Zs. I. Lázár, "Theoretical Study of Fluctuations and Discontinuities in Fluid Dynamics", Ph. D. Thesis, University of Bergen (1999).

[45] K. Tamošiūnas, "Change of the Matter Properties Across Idealized Freeze Out Hypersurface", M.Phil. Thesis, University of Bergen (2001).

[46] U. A: Wiedeman *et al.*, Nucl. Phys. **A 638**, 475c-478c (1998).

[47] J. Cleymans, K. Redlich, Phys. Rev. Lett. **81** 5284 (1998).

[48] N. Arbex, F. Grassi, Y. Hama, O. Socolowski, Phys. Rev. **C 64** 064906 (2001).

[49] C. Y. Wong, "Introduction to High-Energy Heavy Ion Collisions", World Scientific (1994).