

THERMINATOR 2: THERMal heavy IoN generATOR 2[★]

Mikołaj Chojnacki^a, Adam Kisiel^{b,c}, Wojciech Florkowski^{a,d},
Wojciech Broniowski^{a,d},

^a*The H. Niewodniczański Institute of Nuclear Physics, Polish Academy of Sciences, PL-31342 Cracow, Poland*

^b*Physics Department, CERN, CH-1211 Geneve 23, Switzerland*

^c*Faculty of Physics, Warsaw University of Technology, PL-00661 Warsaw, Poland*

^d*Institute of Physics, Jan Kochanowski University, PL-25406 Kielce, Poland*

Abstract

We present an extended version of THERMINATOR, a Monte Carlo event generator dedicated to studies of the statistical production of particles in relativistic heavy-ion collisions. The increased functionality of the code contains the following features: The input of any shape of the freeze-out hypersurface and the expansion velocity field, including the 3+1 dimensional profiles, in particular those generated externally with various hydrodynamic codes. The hypersurfaces may have variable thermal parameters, which allows for studies departing significantly from the mid-rapidity region, where the baryon chemical potential becomes large. We include a library of standard sets of hypersurfaces and velocity profiles describing the RHIC Au+Au data at $\sqrt{s_{NN}} = 200$ GeV for various centralities, as well as those anticipated for the LHC Pb+Pb collisions at $\sqrt{s_{NN}} = 5.5$ TeV. A separate code, FEMTO-THERMINATOR, is provided to carry out the analysis of femtoscopic correlations which are an important source of information concerning the size and expansion of the system. We also include several useful scripts that carry out auxiliary tasks, such as obtaining an estimate of the number of elastic collisions after the freeze-out, counting of particles flowing back into the fireball and violating causality (typically very few), or visualizing various results: the particle p_T -spectra, the elliptic flow coefficients, and the HBT correlation radii. We also investigate the problem of the back-flow of particles into the hydrodynamic region, as well as estimate the elastic rescattering in terms of trajectory crossings. The package is written in C++ and uses the CERN ROOT environment.

Key words: relativistic heavy-ion collisions, statistical hadronization, Monte Carlo event generator, LHC, RHIC, SPS, FAIR, NICA

PACS: 25.75.-q, 25.75.Dw, 25.75.Ld

Program summary

Title of the program: THERMINATOR 2

Catalog identifier:

Program web page:

<http://therminator2.ifj.edu.pl/>

Licensing provisions: none

Computer: any computer with a C++ compiler and the CERN ROOT environment, ver. 5.26 or later, tested with Intel Core2 Duo CPU E8400 @ 3 GHz, 4 GB RAM

Operating system under which the program has been tested:

Linux Ubuntu 10.10 x64 (gcc 4.4.5) ROOT ver. 5.26

Programming language used: C++ with the CERN ROOT libraries, BASH shell

No. of lines in distributed program, including test data:

13000 code, 300 configuration, 400000 input files (hypersurface library)

No. of bytes in distributed program, including test data: 2.7 MB

Distribution format: tar.gz

Nature of physical problem: particle production via statistical hadronization in relativistic heavy-ion collisions

Method of solution: Monte-Carlo simulation, analyzed with ROOT

Restrictions concerning the complexity of the problem: none

Memory required to execute with typical data:

— 30 MB `therm2_events`,

— 150 MB `therm2_femto`

Typical running time: default configuration at 3 GHz

— primordial multiplicities 70 min (calculated only once per case),

— 8 min/500 events,

— 10 min – draw all figures,

— 25 min/one k_T bin in the HBT analysis with 5000 events

Typical data file size: default configuration

— 45 MB/500 events

— 35 MB/correlation file (one k_T bin)

— 45 kB/fit file (projections and fits)

* Supported in part by the Polish Ministry of Science and Higher Education, grant No. N N202 263438 and Foundation for Polish Science.

Email addresses: Mikolaj.Chojnacki@ifj.edu.pl (Mikołaj Chojnacki), kisiel@if.pw.edu.pl (Adam Kisiel), Wojciech.Florkowski@ifj.edu.pl (Wojciech Florkowski), Wojciech.Broniowski@ifj.edu.pl (Wojciech Broniowski).

1 Introduction

We introduce an updated and largely extended version of **THERMINATOR** [1], the **THERMal heavy IoN generATOR**, created to carry out the statistical hadronization in relativistic heavy-ion collisions. Numerous successful analyses have been performed with the help of our code over the last few years [2–63], contributing to better understanding of the involved evolution of the hot and dense system created in ultra-relativistic heavy-ion collision.

THERMINATOR 2 is a Monte Carlo generator written in C++ and using the standard CERN **ROOT** [64] environment. That way, apart from model applications, the code can be easily adapted for purposes directly linked to experimental data analysis, detector modeling, or estimates for the heavy-ion experiments at RHIC, LHC, SPS, FAIR, or NICA.

Originally, **THERMINATOR** was designed to perform the hadronic freeze-out on simple boost-invariant hypersurfaces, such as the Cracow single-freeze-out model [65, 66] and the Blast-Wave model [67–69]. Meanwhile, the code has evolved into a versatile tool, where the freeze-out profile and the expansion velocity field of any shape can be implemented, allowing application to all approaches based on statistical hadronization on a specified hypersurface. The present functionality of the code includes the following new features:

- (1) An implementation of any shape of the freeze-out hypersurface and the expansion velocity field is possible now, including the 2+1 and 3+1 dimensional profiles, in particular those generated externally with various codes for perfect hydrodynamics [22, 70–81].
- (2) The hypersurfaces may have space-time dependent thermal parameters, which allows for studies of non-boost-invariant systems. In particular, one may depart significantly from the mid-rapidity region to the fragmentation regions where the baryon chemical potential becomes large [8, 45].
- (3) The package includes a library of “standard” sets of hypersurfaces and velocity profiles, which describe the Au+Au data [22] at the highest RHIC energy $\sqrt{s_{NN}} = 200$ GeV for various centralities. We also provide the hypersurfaces and velocity profiles anticipated [23] for the LHC Pb+Pb collisions at $\sqrt{s_{NN}} = 5.5$ TeV, which prior to studies with the real data can be used for the modeling of detectors.
- (4) A separate code, the **FEMTO-THERMINATOR**, is provided to carry out the analysis of the femtoscopic correlations. These correlations are an important source of complementary information concerning the size and expansion of the system, which should be reproduced in a realistic description of the data [5, 22, 26].

We also provide several scripts that carry out auxiliary tasks:

- (1) Counting the number of particles flowing back into the fireball, thus violating causality (for realistic freeze-out profiles this number is negligible, below 1%).
- (2) An estimate of the number of elastic collisions (trajectory crossings) after the freeze-out. For realistic cases this number is small (~ 1.5 for pions scattering off other pions), supporting the single-freeze-out approximation.
- (3) Visualization of the results: the particle p_T -spectra, the elliptic flow coefficient v_2 , or the HBT correlation radii.

We have also made an effort to make the package more user-friendly, providing numerous examples of usage in typical situations.

Other codes implementing the thermal description of hadronization have also been developed recently, to mention **SHARE** [82, 83] and **THERMUS** [84], computing the particle abundances, **DRAGON** [28], a Monte Carlo generator of particle production from a fragmented fireball, or **HYDJET++** [53, 85, 86], a heavy-ion event generator with hydrodynamics and jets. A code similar to **THERMINATOR**, **Fast Hadron Freeze-out Generator**, has been presented in Refs. [87, 88].

2 Statistical hadronization

For the detailed description of the physics behind **THERMINATOR 2** the reader is referred to the original paper [1]. Here we only wish to stress that the statistical approach to heavy-ion collisions has proved to be very useful for a very broad range of observables. With few physical input parameters, such as the temperature, chemical potentials, size, and the velocity of the collective flow, the models describe the observed particle abundances [65, 68, 69, 89–107], the transverse-momentum spectra [66], the balance functions [108, 109], the elliptic flow coefficient [110, 111], and femtoscopic observables [5, 22, 26]. A uniform description of the spectra, the elliptic flow coefficient v_2 , and the HBT data achieved in [22] explained for the first time the so-called “RHIC HBT puzzle”.

The key element of the success behind the statistical approach is the inclusion of the full list [112] of hadronic resonances, whose number grows rapidly according to the Hagedorn hypothesis [113–118]. At the rather high temperature of the freeze-out, ~ 140 – 165 MeV, the resonances contribute very significantly to the observables. **THERMINATOR 2** uses the particle data tables in the universal form originally included in the **SHARE** [82] package, and incorporates the four-**** and three-*** resonances.

All particles are created at the freeze-out hypersurface according to the Cooper-

Frye formula [119]. The two- and three-body decays of resonances, proceeding in cascades, produce the stable particles observed in the detectors. **THERMINATOR** offers the full space-time information on positions and velocities of the produced particles. As a Monte Carlo event generator written in **C++** in the CERN **ROOT** [64] environment, **THERMINATOR** can be straightforwardly interfaced to the standard software routinely used in the data analysis of relativistic heavy-ion collisions at the LHC, RHIC, SPS, FAIR or NICA. The inclusion of experimental acceptance and kinematic cuts poses no difficulty in a Monte Carlo generator.

2.1 Cooper-Frye formalism

Our starting point for the analysis of different models is the famous Cooper-Frye formula [119]. It expresses the number of hadrons being produced on the freeze-out hypersurface Σ^μ by the following integral

$$N = (2s + 1) \int \frac{d^3p}{(2\pi)^3 E_p} \int d\Sigma_\mu(x) p^\mu f(x, p), \quad (1)$$

where

$$f(p \cdot u) = \left\{ \exp \left[\frac{p_\mu u^\mu - (B\mu_B + I_3\mu_{I_3} + S\mu_S + C\mu_C)}{T} \right] \pm 1 \right\}^{-1} \quad (2)$$

is the phase-space distribution function of the particles (the stable ones and resonances), and $d\Sigma^\mu$ is the three-dimensional element of the freeze-out hypersurface. The latter may be calculated with the help of the formula known from the differential geometry [120]

$$d\Sigma_\mu = \varepsilon_{\mu\alpha\beta\gamma} \frac{\partial x^\alpha}{\partial \alpha} \frac{\partial x^\beta}{\partial \beta} \frac{\partial x^\gamma}{\partial \gamma} d\alpha d\beta d\gamma, \quad (3)$$

where $\varepsilon_{\mu\alpha\beta\gamma}$ is the Levi-Civita tensor ($\varepsilon_{0123} = +1$) and the variables α , β , and γ are used to parametrize the three-dimensional freeze-out hypersurface submerged in the four-dimensional Minkowski space. The quantity $d\Sigma^\mu$ may be interpreted as the four-vector normal to the hypersurface with the norm equal to the “volume of the hypersurface element”. Explicitly,

$$d\Sigma_0 = \begin{vmatrix} \frac{\partial x}{\partial \alpha} & \frac{\partial x}{\partial \beta} & \frac{\partial x}{\partial \gamma} \\ \frac{\partial y}{\partial \alpha} & \frac{\partial y}{\partial \beta} & \frac{\partial y}{\partial \gamma} \\ \frac{\partial z}{\partial \alpha} & \frac{\partial z}{\partial \beta} & \frac{\partial z}{\partial \gamma} \end{vmatrix} d\alpha d\beta d\gamma, \quad (4)$$

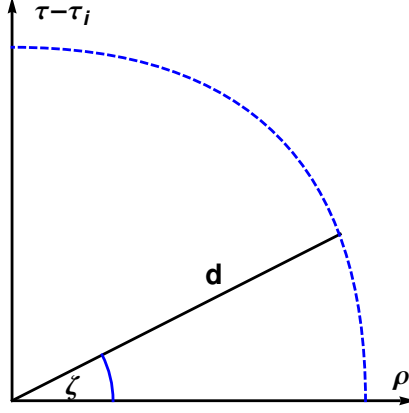


Fig. 1. The system of coordinates used to parametrize the boost-invariant freeze-out hypersurfaces obtained from the 2+1 hydrodynamic codes. The dashed line shows the freeze-out points with the fixed values of the azimuthal angle ϕ .

while the remaining components are obtained via cyclic permutations of t , z , y , and z . In the following subsections we shall explicitly provide $d\Sigma_\mu$ in the form appropriate for various considered cases. The formulas will be directly used to generate the hadrons with the Monte-Carlo method.

The quantities appearing in the expressions below have the following physical interpretation:

- thermodynamic quantities on the freeze-out hypersurface:
 - T - temperature,
 - μ_B - baryon chemical potential,
 - μ_{I_3} - isospin chemical potential,
 - μ_S - strange chemical potential,
 - μ_C - charmed chemical potential,
- space-time coordinates:
 - $\tau = \sqrt{t^2 - z^2}$ - longitudinal proper time,
 - $\rho = \sqrt{x^2 + y^2}$ - distance in the transverse plane,
 - ϕ - azimuthal angle,
 - $Y_s = \frac{1}{2} \ln((t + z)/(t - z))$ - space-time rapidity,
- coordinates and parameters used to specify the freeze-out hypersurface:
 - τ_i - the initial proper time for hydrodynamics,
 - τ_f - the final proper time,
 - ζ - angle in $\rho - \tau$ plane, see Fig. 1
 - θ - angle between the Y_s -axis and the direction determined by the space time point $(\tau_i, 0, 0, 0)$ and the point on the freeze-out hypersurface, see Fig. 2,
 - $d = d(\zeta, \phi, \theta)$ - distance from the space-time point $(\tau_i, 0, 0, 0)$ to the point

on the freeze-out hypersurface, see Fig. 2,

- flow characteristics of the fluid element:
 - u_x, u_y - transverse components of the four-velocity,
 - $v_T = \sqrt{v_x^2 + v_y^2}$ - magnitude of the transverse velocity,
 - ϕ_f - azimuthal angle of the transverse velocity,
 - Y_f - rapidity of the fluid element,
- properties of a particle generated on the freeze-out hypersurface:
 - s - spin,
 - m - mass,
 - p_T - transverse momentum,
 - $m_T = \sqrt{m^2 + p_T^2}$ - transverse mass,
 - ϕ_p - azimuthal angle of the transverse momentum,
 - Y - rapidity,

In the case of freeze-out hypersurfaces obtained in the framework of the perfect-fluid hydrodynamics the distribution function in (1) is the Fermi-Dirac or Bose-Einstein distribution and the number of hadrons produced is given by the formula

$$N = (2s + 1) \int \frac{dY p_T dp_T d\phi_p}{(2\pi)^3} \int \frac{d\Sigma_\mu p^\mu}{\exp(-\frac{B\mu_B + I_3\mu_{I_3} + S\mu_S + C\mu_C}{T}) \exp(\frac{p_\mu u^\mu}{T}) \pm 1}. \quad (5)$$

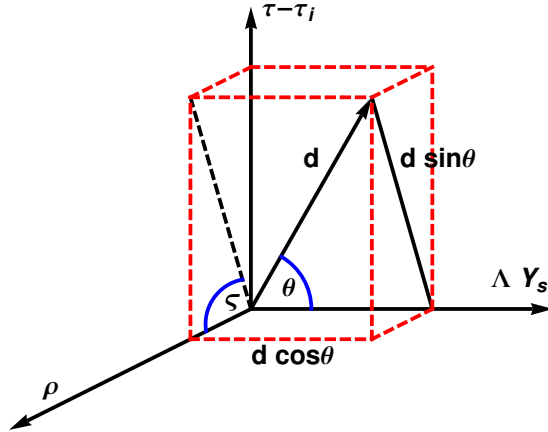


Fig. 2. The system of coordinates used to parametrize the freeze-out hypersurfaces obtained from the 3+1 hydrodynamic codes, see Eqs. (6) and (7).

2.2 Freeze-out in 3+1-dimensional perfect-fluid hydrodynamics

First, we consider the most general case where both the freeze-out hypersurface and the hydrodynamic flow at freeze-out are obtained from the 3+1 perfect-fluid hydrodynamic code, with no symmetries assumed. In this situation the freeze-out hypersurface is parametrized in the following way [35],

$$\begin{aligned}
t &= (\tau_i + d(\zeta, \phi, \theta) \sin \theta \sin \zeta) \cosh \frac{d(\zeta, \phi, \theta) \cos \theta}{\Lambda}, \\
x &= d(\zeta, \phi, \theta) \sin \theta \cos \zeta \cos \phi, \\
y &= d(\zeta, \phi, \theta) \sin \theta \cos \zeta \sin \phi, \\
z &= (\tau_i + d(\zeta, \phi, \theta) \sin \theta \sin \zeta) \sinh \frac{d(\zeta, \phi, \theta) \cos \theta}{\Lambda},
\end{aligned} \tag{6}$$

where d is a function of the parameters ϕ, ζ , and θ , which is obtained numerically from the hydrodynamic code, see Fig. 2. The parameter Λ is a scale used to change the dimensionless space-time rapidity Y_s into a dimensional quantity (for example one may choose $\Lambda = \rho_{\max}/Y_s^{\max}$), namely

$$\begin{aligned}
Y_s &= Y_s(\zeta, \phi, \theta) = \frac{d(\zeta, \phi, \theta) \cos \theta}{\Lambda}, \\
\tau &= \tau(\zeta, \phi, \theta) = \tau_i + d(\zeta, \phi, \theta) \sin \theta \sin \zeta.
\end{aligned} \tag{7}$$

In addition, we express the particle four-momentum and the fluid four-velocity in terms of particle and fluid rapidities, respectively, which leads to the expression

$$p_\mu u^\mu = \sqrt{1 + u_x^2 + u_y^2} [m_T \cosh(Y_f - Y) - p_T(u_x \cos \phi_p + u_y \sin \phi_p)]. \tag{8}$$

The calculation of the integration measure in the Cooper-Frye gives

$$\begin{aligned}
d\Sigma_\mu p^\mu = d^2 \sin \theta \left\{ \frac{(\tau_i + d \sin \zeta)}{\Lambda} \left[\frac{\partial d}{\partial \zeta} \cos \zeta \left(-m_T \cos \zeta \cosh \left(Y - \frac{d \cos \theta}{\Lambda} \right) \right. \right. \right. \\
\left. \left. + p_T \sin \zeta \cos(\phi - \phi_p) \right) + \left(d \sin \theta - \frac{\partial d}{\partial \theta} \cos \theta \right) \right. \\
\left. \times \left(m_T \sin \zeta \cosh \left(Y - \frac{d \cos \theta}{\Lambda} \right) + p_T \cos \zeta \cos(\phi - \phi_p) \right) \right. \\
\left. + \frac{\partial d}{\partial \phi} p_T \sin(\phi - \phi_p) \right] \\
\left. + \left(d \cos \theta + \frac{\partial d}{\partial \theta} \sin \theta \right) \sin \theta \sinh \left(Y - \frac{d \cos \theta}{\Lambda} \right) \right\}. \quad (9)
\end{aligned}$$

Using (7) in (9) one obtains a more compact form

$$\begin{aligned}
d\Sigma_\mu p^\mu = \frac{d \sin \theta}{\Lambda} d\tau \left[\frac{\partial d}{\partial \zeta} \cos \zeta (-m_T \cos \zeta \cosh(Y - Y_s) \right. \\
+ p_T \sin \zeta \cos(\phi - \phi_p)) + \cos \zeta \sin \theta \left(d \sin \theta - \frac{\partial d}{\partial \theta} \cos \theta \right) \\
\times (m_T \sin \zeta \cosh(Y - Y_s) + p_T \cos \zeta \cos(\phi - \phi_p)) \\
+ \frac{\partial d}{\partial \phi} p_T \sin(\phi - \phi_p) \\
\left. + \frac{\Lambda}{\tau} \left(d \cos \theta + \frac{\partial d}{\partial \theta} \sin \theta \right) \sin \theta \sinh(Y - Y_s) \right]. \quad (10)
\end{aligned}$$

2.3 Freeze-out in 2+1-dimensional boost-invariant hydrodynamics

For boost-invariant systems, considered in approximate treatment of the heavy-ion collisions, the fluid rapidity is equal to space-time rapidity, $Y_f = Y_s$, and the function d depends only on ϕ and ζ . In this case we find [18]

$$p_\mu u^\mu = \frac{1}{\sqrt{1 - v_T^2}} (m_T \cosh(Y - Y_s) - v_T p_T \cos(\phi - \phi_p)) \quad (11)$$

and

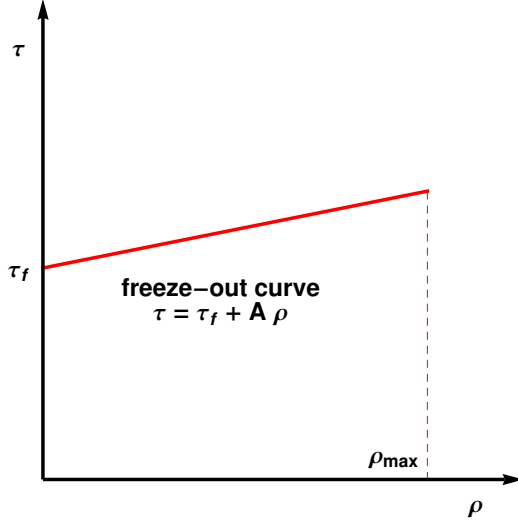


Fig. 3. The freeze-out curves considered in the modified Blast-Wave model [5].

$$\begin{aligned}
d\Sigma_\mu p^\mu = d\tau & \left[\frac{\partial d}{\partial \zeta} \cos \zeta \left(-m_T \cos \zeta \cosh(Y - Y_s) + p_T \sin \zeta \cos(\phi - \phi_p) \right) \right. \\
& + d \cos \zeta \left(m_T \sin \zeta \cosh(Y - Y_s) + p_T \cos \zeta \cos(\phi - \phi_p) \right) \\
& \left. + \frac{\partial d}{\partial \phi} p_T \sin(\phi - \phi_p) \right]. \quad (12)
\end{aligned}$$

2.4 Blast-wave models

In this Section we discuss one of the most popular parameterizations of the freeze-out hypersurface — the Blast-Wave model. In its standard form, the model is boost-invariant and cylindrically symmetric. Moreover, the assumption is made that the freeze-out happens at a constant value of the proper time

$$\tau(\zeta) = \tau_f = \text{const.} \quad (13)$$

In order to get a broader applicability, we generalize this condition to the formula

$$\tau(\rho) = \tau_f + A\rho, \quad (14)$$

where A is a constant that describes the slope of the freeze-out curve in the Minkowski space, see Fig. 3. With $A > 0$ ($A < 0$) we may consider the freeze-out scenarios where the outer parts of the system freeze-out later (earlier). Of course, with $A = 0$ we reproduce the standard Blast-Wave parametrization. In this case we use

$$p_\mu u^\mu = \frac{1}{\sqrt{1 - v_T^2}} (m_T \cosh(Y_s - Y) - v_T p_T \cos(\phi - \phi_p)) \quad (15)$$

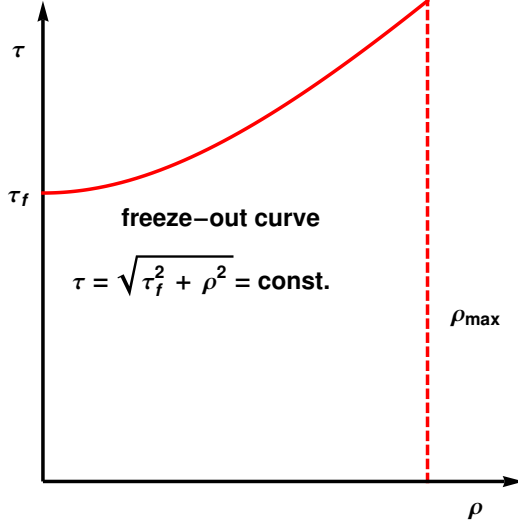


Fig. 4. The freeze-out curve assumed in the Cracow model.

and

$$d\Sigma_\mu p^\mu = (\tau_f + A \rho) \rho (m_T \cosh(Y_s - Y) - A p_T \cos(\phi - \phi_p)). \quad (16)$$

The freeze-out conditions defined by Eq. (14) with different values of A were studied in [5].

2.5 Cracow model

Another boost-invariant and cylindrically symmetric freeze-out model [66] assumes that the hypersurface is defined by the condition,

$$x^\mu x_\mu = t^2 - x^2 - y^2 - z^2 = \tau^2 - \rho^2 = \tau_f^2 = \text{const.}, \quad (17)$$

see Fig. 4, and the fluid four-velocity is proportional to the space-time position

$$u^\mu = \gamma(1, \mathbf{v}) = \frac{x^\mu}{\tau_f} = \frac{t}{\tau_f} \left(1, \frac{\mathbf{x}}{t}\right). \quad (18)$$

Note that the physical interpretation of the parameter τ_f is now slightly different from that used in the previous Section.

The calculation of the volume element of the freeze-out hypersurface shows that it is proportional to u^μ , as in the model of Siemens and Rasmussen [121], so we may write

$$p_\mu u^\mu = \frac{1}{\tau_f} \left(\sqrt{\tau_f^2 + \rho^2} m_T \cosh(Y_s - Y) - \rho p_T \cos(\phi - \phi_p) \right) \quad (19)$$

and

$$d\Sigma_\mu p^\mu = \tau_f \rho p_\mu u^\mu. \quad (20)$$

2.6 Resonance decays

The two- and three-body resonance decays are implemented in exactly the same manner as in the original version of the code [1]. The format of the tables from SHARE [82] is presented in Appendix A.

3 Structure of THERMINATOR 2

THERMINATOR is written in C++ and uses the CERN ROOT environment – *An object-oriented data analysis framework* [64].

3.1 Modules and their functions

The code consists of several modules performing subsequent tasks. First, the information on particles and their decays is read by `Parcer.cxx` and stored by `ParticleDB.cxx`. Then `EventGenerator.cxx` conducts the generation of events, which consists of several tasks. `Integrator.cxx` computes the average multiplicity of primordial particles corresponding to a given model and parameters. Next, `Event.cxx` generates the event, first producing the primordial particles with the multiplicity given by the Poisson distribution from `Integrator.cxx`, and then calling `ParticleDacayer.cxx` to carry out the decays of resonances. Finally, the event is written to the output file. The corresponding block structure is shown in Fig. 5. The organization of the input and output files is shown in Fig. 6.

3.2 File structure

After unpacking the THERMINATOR 2 package in the directory `therminator2` one obtains the directory structure presented in Fig.7. The main directory of THERMINATOR 2 contains the following files:

`Doxygen` – Doxygen [122] configuration,
`events.ini` – THERMINATOR 2 configuration used by `therm2_events`,

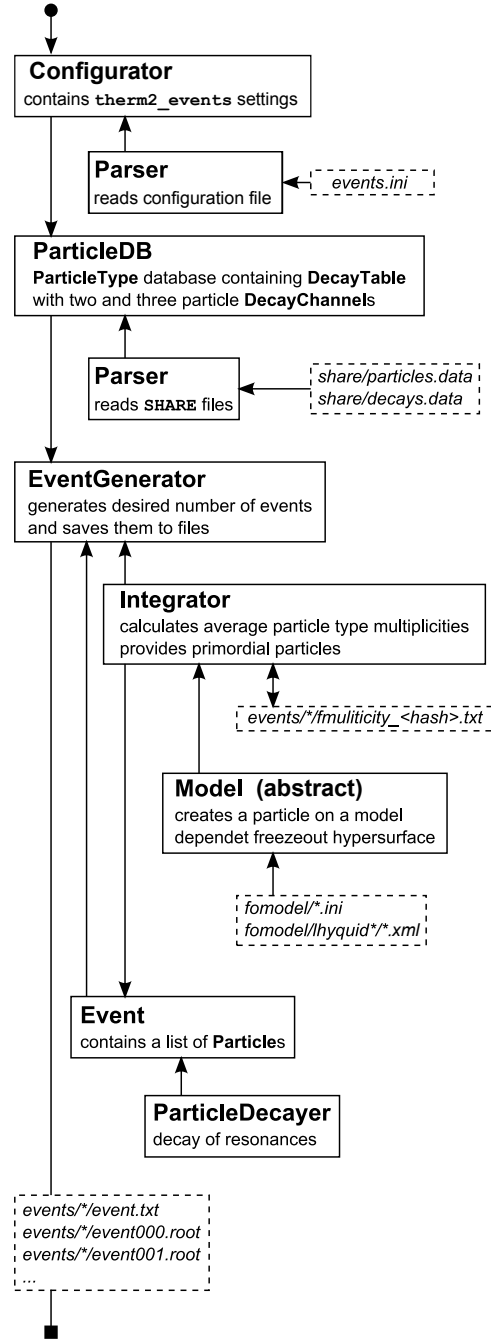


Fig. 5. Visual concept of the **therm2_events** program. Solid line boxes represent classes or a groups of classes with their names in bold font. Dashed rectangular boxes represent input or output files. The arrows symbolize the communication between classes.

femto.ini – THERMINATOR 2 configuration used by **therm2_femto**,
hbtfit.ini – THERMINATOR 2 configuration used by **therm2_hbtfit**,
Makefile – make configuration,

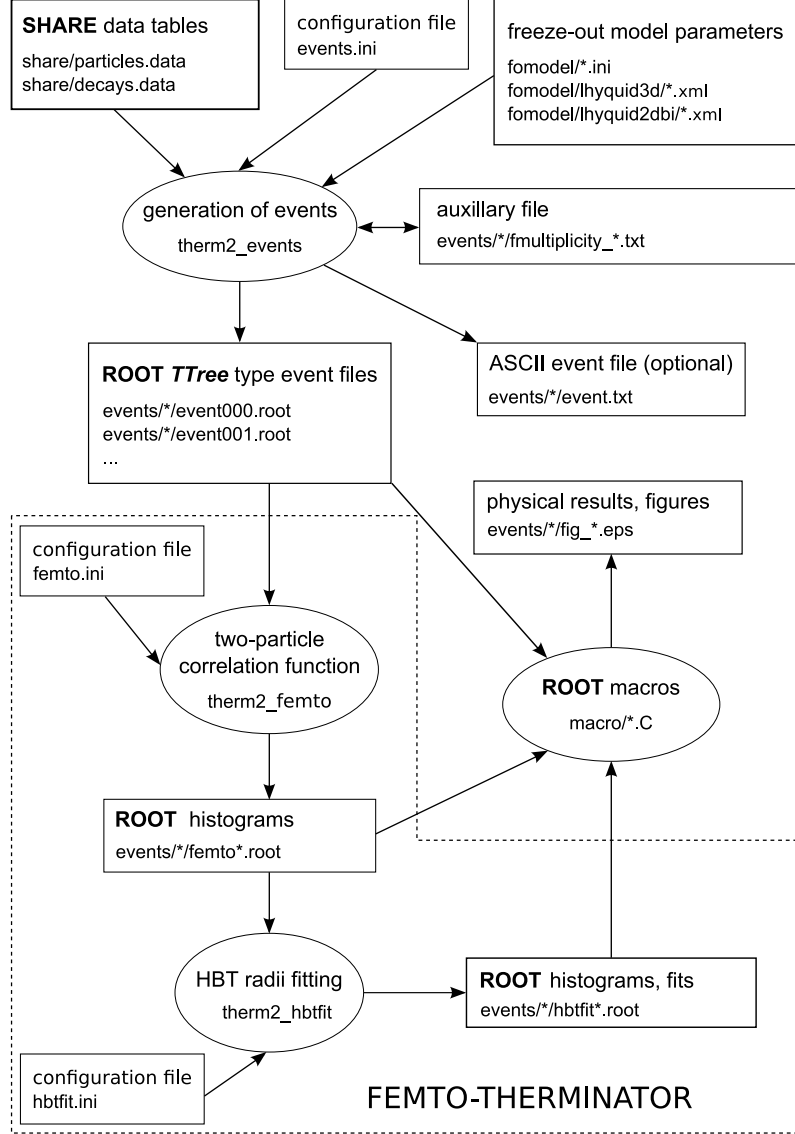


Fig. 6. Organization of input and output files. Rectangular boxes represent files, ovals correspond to codes or scripts. The arrows indicate reading or writing. The part enclosed in the dashed line corresponds to the **FEMTO-THERMINATOR** package.

`manual.html` – reference manual in HTML format (created with `make doc`),
 redirection to `doc/html/index.html`,
`manual.pdf` – reference manual in PDF format (created with `make doc`),
 symbolic link to `doc/latex/refman.pdf`,
`runtorque.sh` – bash shell script for TORQUE [123],
`runall.sh` – bash shell script for the full calculation,
`runevents.sh` – bash shell script for generating events,
`runfigure.sh` – bash shell script for generating figures,
`runhbt.sh` – bash shell script for the femtoscopic analysis,
`therm2_events` – program generating events (created with `make`),
`therm2_femto` – program for two-particle correlations (created with `make`),

```

-therminator2/
├── addons/
├── build/
│   ├── doxygen/
│   ├── include/
│   ├── obj/
│   └── src/
├── doc/
├── events/
├── fomodel/
│   ├── lhyquid2dbi/
│   └── lhyquid3d/
├── macro/
└── share/

```

Fig. 7. Directory structure of the THERMINATOR 2 package.

`therm2_hbtfit` – program for HBT analysis (created with `make`).

The other directories are:

`therminator2/build/doxygen/`

– contains Doxygen input files used to generate the reference manual,

`therminator2/build/include/`

– contains header files (`*.h`) with the declarations of classes and structures, as well as the general declaration file `THGlobal.h`,

`therminator2/build/obj/`

– contains the compiled source files (`*.o`) and the binaries `therm2_events` and `therm2_femto` (created with `make`),

`therminator2/build/src/`

– contains the source code files for the classes (`*.cxx`) and the programs `therm2_events.cxx` and `therm2_femto.cxx`,

`therminator2/doc/`

– contains the Doxygen reference manuals in the HTML and LaTeX formats (created with `make doc`),

`therminator2/events/`

– contains the generated event files, figures, correlation function files, etc. (the contents of the `events/` directory and the subdirectory structure depends on the selected model and its settings).

The default subdirectories for each model are as follows:

`therminator2/event/krakow/` – the Krakow Single Freeze-Out model,

`therminator2/event/blastwave/` – the Blast-Wave model,

`therminator2/event/bwa/` – the Blast-Wave A-class model,

therminator2/event/lhyquid2dbi-*/ – the Lhyquid 2+1D Boost-Invariant hydro,
 therminator2/event/lhyquid3d/¹ – the Lhyquid 3+1D hydro.

The files stored in the `events/*/` subdirectory:

`event*.root` – event file,
`fmultiplicity*.txt` – primordial particle multiplicity,
`fig*.eps` – figure in EPS format,
`fig*.png` – figure in PNG format,
`fig*.xml` – figure data in XML format,
`fig*.C` – ROOT macro to generate a figure,
`femto*.root` – two particle correlation function,
`hbtfit*.root` – fit to the two particle correlation function,
`events.ini` – event generator configuration file,
`femto.ini` – two-particle correlation configuration file,
`hbtfit.ini` – HBT fit configuration file,
`model_config.ini` – freeze-out model configuration file (the file name depends on the chosen model, e.g. `lhyquid2dbi.ini`).

File naming conventions are as follows: The `event*.root` files are always stored in a sequence beginning with `event000.root`, and subsequently with the file counter incremented by one. The `fmultiplicity*.txt` file has an eight-character hex-base number representing the CRC-32 identification number calculated from the model parameters, e.g., `fmultiplicity_10D57D490.txt` for the default case of RHIC with the centrality 20-30%. The `fig*.*` files have the name inherited from the corresponding ROOT macros (with the word “figure” shortened to “fig”). The `femto*.root` and `fmfit*.root` filenames depend on the particle pair type, average pair momentum, and the inclusion of particles from resonances, e.g., `femtopipi0200a.root` for the pion-pion pairs with average $k_T = 200$ MeV, and with resonance decays indicated with “a”.

`therminator2/fomodel/`

– contains the freeze-out model configuration files (`*.ini`) and freeze-out hypersurfaces data in the XML format (`*.xml`). The five freeze-out model configuration files are supplied:

`blastwave.ini` – for Blast-Wave model,
`bwa.ini` – for modified Blast-Wave A-Class model,
`krakow.ini` – for Krakow Single Freeze-out model,
`lhyquid2dbi.ini` – for 2+1 boost-invariant hydrodynamics,
`lhyquid3d.ini` – for 3+1 hydrodynamics.

¹ THERMINATOR 2 distribution does not include hypersurfaces for this case. They can be downloaded from the project homepage <http://therminator2.ifj.edu.pl>.

The freeze-out hypersurface input files for RHIC at $\sqrt{s_{NN}} = 200$ GeV and various centralities have the following names:

```
lhyquid2dbi/RHICAuAu200c0005Ti500ti025Tf145.xml (for  $c = 0-5$  %),
lhyquid2dbi/RHICAuAu200c0510Ti491ti025Tf145.xml (5-10 %),
lhyquid2dbi/RHICAuAu200c1020Ti476ti025Tf145.xml (10-20 %),
lhyquid2dbi/RHICAuAu200c2030Ti455ti025Tf145.xml (20-30 %),
lhyquid2dbi/RHICAuAu200c3040Ti429ti025Tf145.xml (30-40 %),
lhyquid2dbi/RHICAuAu200c4050Ti398ti025Tf145.xml (40-50 %),
lhyquid2dbi/RHICAuAu200c5060Ti354ti025Tf145.xml (50-60 %),
lhyquid2dbi/RHICAuAu200c6070Ti279ti025Tf145.xml (60-70 %).
```

The name encodes the colliding nuclei, the collision energy (in MeV), the centrality class, the initial central temperature T_i (in MeV), the starting time used in hydrodynamics t_i (in fm/c), and the freeze-out temperature T_f (in MeV). In an analogous way, the freeze-out hypersurface input files for the LHC collisions at $\sqrt{s_{NN}} = 5.5$ TeV and various centralities are:

```
lhyquid2dbi/LHCPbPb5500c0005Ti500ti100Tf145.xml (for  $c = 0-5$  %),
lhyquid2dbi/LHCPbPb5500c0510Ti500ti100Tf145.xml (5-10 %),
lhyquid2dbi/LHCPbPb5500c1020Ti500ti100Tf145.xml (10-20 %),
lhyquid2dbi/LHCPbPb5500c2030Ti500ti100Tf145.xml (20-30 %),
lhyquid2dbi/LHCPbPb5500c3040Ti500ti100Tf145.xml (30-40 %),
lhyquid2dbi/LHCPbPb5500c4050Ti500ti100Tf145.xml (40-50 %),
lhyquid2dbi/LHCPbPb5500c5060Ti500ti100Tf145.xml (50-60 %),
lhyquid2dbi/LHCPbPb5500c6070Ti500ti100Tf145.xml (60-70 %).
```

`therminator2/macro/`

– contains the ROOT macro files for generating figures in the event analysis:

```
figure_disteta.C – distribution in pseudorapidity,
figure_distmt.C – distribution in transverse mass,
figure_distpt.C – distribution in transverse momentum,
figure_distpt_exotic.C – distribution in transverse momentum of less
abundant particles (e.g., those with strangeness),
figure_distpt_pion.C – pions' anatomy in transverse momentum (contri-
butions from individual resonance decays to pion spectra),
figure_distpt_proton.C – protons' anatomy in transverse momentum (con-
tributions from individual resonance decays to proton spectra),
figure_distrap.C – distribution in rapidity,
figure_distrap_prim.C – primordial-particle distribution in rapidity,
figure_hsret.C – primordial-particle distribution in  $(\rho, \eta, \tau)$ ,
figure_hsxyt.C – primordial-particle distribution in  $(x, y, \tau)$ ,
figure_xpx.C – position-momentum correlation,  $x - p_x$ ,
figure_v2pt.C – differential elliptic flow at midrapidity,
figure_v4pt.C – differential  $v_4$  coefficient at midrapidity,
figure_v2rap.C – integrated elliptic flow vs. rapidity.
```

Other ROOT macros used by the `figure*.C` files are:

```
events2chain.C – reads event files,
hist2xml.C – converts histogram data into the XML format,
```

`model2legend.C` – creates figure legends displaying the model parameters.

`therminator2/share`

– contains the SHARE [82] database files:

`decays.data` – particle decay channels,

`particles.data` – particle properties.

`therminator2/addons`

– contains an auxiliary program (estimate of the elastic rescattering):

`coll.mk` – makefile,

`collcount.cxx` – program estimating elastic collisions per pion.

3.3 *.ini files

The main configuration file `events.ini` (Table 1) is used by `therm2_events` program to select the model, number of generated events, type of output, file and directory names, etc. Depending on the value of `FreezeOutModel`, the program reads additional information from an appropriate model settings file `*.ini`. The file is located in the *FreezeOutDir* directory.

The model configuration file (Table 2) contains the parameters of a particular model, e.g., the configuration file for the default **FreezeOutModel = Lhyquid2DBI** is `fomodel/lhyquid2dbi.ini`.

The variable `IntegrateSamples`, used to obtain the integrand ranges and the particle average multiplicities, is set to a large value (5000000 by default). This leads to accurate results, but increases the running time of the first stage of `therm2_events`. The user may decrease this value, however, care is needed in the inspection of results. A very large value of `NumberOfEvents`, above 20000, is needed to achieve stable HBT results. A low value (about 1000) is sufficient to obtain the p_T spectra. Of course, the running time is proportional to `NumberOfEvents`.

4 Storage of events

The particles are stored in the computer memory as collections corresponding to one event. Once the whole event is generated (all primordial particles are produced and all unstable particles have been sequentially decayed), the event is written down to the output file and erased from the memory. There are two types of output: plain ASCII text and ROOT file.

4.1 Plain text output file

The plain text output has the advantage of having the complete information and being easily readable by a human-being or a computer program. It is, however, not efficient in terms of disk-space usage and data-access time. The structure of the plain text file is the following:

- The first two lines form the file header containing a description of the `EVENT_ENTRY`, see further in text.
- Each event in the output file consists of an event header containing the `EVENT_ID` (a unique number generated with the CRC-32 algorithm from the system time and event sequence number) and number of particles in the event.
- The lines that follow the event header are the particle entries. Each particle is represented by a single line.

An example of a few lines of the test output:

```
# THERMINATOR 2 text output
#<EVENT_ENTRY>eid fathereid pid fatherpid rootpid decayed mass e px py pz t x y z</EVENT_ENTRY>
#<EVENT_ID>0x65679F6</EVENT_ID>
#<NO_OF_PARTICLES>9684</NO_OF_PARTICLES>
0 -1 9001 9001 9001 0 2.350000e+00 5.423351e+00 5.105272e-01 8.400604e-01 -4.787891e+00
   3.012708e+01 3.797786e+00 1.937673e+00 -2.789553e+01
1 -1 5218 5218 5218 1 2.250000e+00 5.035699e+00 1.177745e+00 -3.007674e-01 -4.337997e+00
   2.272015e+01 2.773765e+00 -1.740429e+00 -1.985397e+01
```

The particle event entry columns contain the following information:

- **eid**, the particle sequence number in the event,
- **fathereid**, the sequence number of the parent,
- **pid**, Particle Data Group (PDG) identification number,
- **fatherpid**, PDG identification number of the parent,
- **rootpid**, PDG identification number of the original primordial particle starting the decay sequence,
- the decay flag (1 if particle has decayed, 0 if not),
- mass and components of the four momentum: **e**, **px**, **py**, and **pz** (in GeV),
- space-time coordinates of the creation point: **t**, **x**, **y** and **z** (in fm/c).

The information on **eid**, **fathereid**, together with **pid** and **fathereid** can be used to trace back the resonance cascade from the last stable particle up to its primordial particle born on the freeze-out hypersurface. Particles with **fathereid** = -1 are primordial particles with **pid** = **fathereid** = **rootpid**.

4.2 ROOT file

The information stored in the ROOT file is slightly different from the one stored in the text file and it is grouped in three separate **TTree** type objects.

The tree called **particles** consists of one branch named **particle**. The branch is filled with particle entries (leaves) – one leaf per particle. This tree stores information on particles from 500 events (by default). The particle leaf consist of a class called **ParticleCoor** that stores the following particle properties:

data type	name	description
Float_t	mass	mass
Float_t	t, x, y, z	space-time coordinates (in fm/c)
Float_t	e, px, py, pz	energy and momentum coordinates (in GeV)
Int_t	decayed	decay flag
Int_t	pid	PDG identification number
Int_t	fatherpid	parent PDG number
Int_t	rootpid	root (primordial) particle PDG number
Int_t	eid	sequence number in the event
Int_t	fathereid	parent sequence number in the event
UInt_t	eventid	CRC-32 unique id of the event

The data types used here are the ROOT data types.

The events tree called **events** consists of one branch named **event**. The branch is filled with event entries (leaves) – one leaf per event, by default 500 entries per file. The event leaf consist of a structure called **StructEvent** that stores the information on the number of particles in each event and the event id number:

data type	name	description
UInt_t	eventID	unique ID of the event generated with CRC-32 algorithm
UInt_t	entries	number of particle entries in this event
UInt_t	entriesprev	number of previous entries, 0 by default.

The **entriesprev** variable is filled with 0 and is not used at this stage. It is used in the classes that read a sequence of event files for analysis.

The parameters tree called **parameters** consists of eight branches that store the following information :

branch name	data type	description
IntegrateSample	UInt_t	number of Monte-Carlo integration samples
Randomize	UInt_t	random seed switch (1-random seed, 0-constant seed)
TimeStamp	Char_t	date and time of this run
ModelID	UInt_t	id number of the chosen freeze-out model
ModelName	Char_t	name of the freeze-out model
ModelHash	Char_t	CRC-32 hash made of the freeze-out model parameters
ModelDescription	Char_t	model parameter description
ModelParameters	<i>Model_t</i>	structure containing model parameters.

The structure **Model_t** is the default freeze-out model structure that does not contain any parameter information. Each freeze-out model has a separate structure defined and saved to the file. The detailed information can be found in the `therminator2/build/include/StructModel.h` file. As an example, the structure for the default freeze-out model is presented as follows:

data type	name	description
Float_t	RapPRange	rapidity range
Float_t	RapSRange	spatial rapidity range
Float_t	TauI	initial proper time
Float_t	TempF	freeze-out temperature
Float_t	MuB	baryochemical potential
Float_t	MuI	isospin potential
Float_t	MuS	strangeness potential
Float_t	MuC	charm potential
Float_t	CollidingEnergy	CMS beam energy
Float_t	CentralityMin	lower limit of centrality
Float_t	CentralityMax	higher limit of centrality
Float_t	ImpactParameter	impact parameter
Float_t	TempI	initial central temperature
Char_t	DeviceName[30]	name of the device, e.g. RHIC, LHC
Char_t	CollidingSystem[30]	colliding nuclei, e.g. Au, Pb.

5 Installation and sample runs

5.1 Downloading the latest version

THERMINATOR 2 is distributed in a form of a .tar.gz archive containing the C++ sources, scripts, configuration files, and the data files. The latest package can be downloaded from the website

<http://therminator2.ifj.edu.pl/> .

Prior to compiling and running **THERMINATOR 2**, the user has to install the

following programs:

- C++ compiler,
- ROOT ver. 5.26 or later.

The latest version of the ROOT package is available at <http://root.cern.ch/>. The simplest way to proceed is to follow the instructions below. In this way one first creates a directory called `therminator2`. Then, the package is saved and extracted,

```
mkdir therminator2
cd therminator2/
wget therminator2.ifj.edu.pl/therminator2-latest.tar.gz
tar xzf therminator2-latest.tar.gz
```

In place of “latest” the actual number appears. The information about the version of the package is contained in the `version` file.

5.2 *Compilation*

One compiles THERMINATOR 2 with the command

```
make
```

After a successful compilation, the binary files are present in the main directory.

5.3 *Doxygen documentation*

The source files of THERMINATOR 2 are documented with **Doxygen** to generate the web-page and the PDF files with detailed information on the classes, variables, and files. The user must have the Doxygen package installed on the system (latest version can be obtained from www.doxygen.org/). The documentation is created with the command

```
make doc
```

In the main distribution directory, two files are created: `manual.html`, which redirects the web browser to the main documentation page, and `manual.pdf`, which is a symbolic link to the PDF reference manual.

5.4 A test run

A test run is executed with with the script

```
./runall.sh
```

This run displays all basic functionality of the package, including the femto-scopic analysis, producing the plots with the physical results, etc. It corresponds to Au+Au collisions at the top RHIC energy and centrality 20-30%. The 2D Boost-Invariant hydrodynamics is used as the input. By default, 5000 events are used. The test run takes several hours to complete (4 h 20 min on a 3 GHz processor). At the end the events and figures are contained in the `events/lhyquid2dbi-RHICAuAu200c2030Ti455ti025Tf145/` directory.

5.5 Running

Now we describe in a greater detail the elements (codes and figure macros) used in the script of the previous subsection.

The main code of THERMINATOR 2 is `therm2_events`, which generates the events. It has been set by default (see the description of the `*.ini` files) to generate 500 events for the case of the RHIC top energy and centrality 20-30%. The 2D Boost-Invariant hydrodynamics is used. In order to perform the default run, issue the command

```
./therm2_events
```

This creates the file `eventnnn.root` in the subdirectory `therminator2/events/lhyquid2dbi-RHICAuAu200c2030Ti455ti025Tf145` together with a text file `fmultiplicity_10D57D490.txt` that holds the primordial particle multiplicities. The number identifier `nnn` is 000 if no event files are present in the subdirectory, whereas it is one unit larger from the file with highest identifier if the files are already present.

The script

```
./runevents.sh
```

generates as many events as requested by the `NumberOfEvents` parameter in the `events.ini` file. The user may add more events by repeating the `./run-events.sh` command.

To generate via a ROOT macro the figure showing the p_T spectra, issue the command

```
root -x './macro/figure_distpt.C("./events/
      lhyquid2dbi-RHICAuAu200c2030Ti455ti025Tf145/",n)'
```

where n is the number of event files used, starting from `event000.root`. The single and double quotation marks are necessary to pass the parameters to the macro. The macro creates two files in the events subdirectory: `fig_distpt.eps` and `fig_distpt.xml`. The first one is the EPS graphics file, the second one contains the data from the histogram in a form of the XML file.

The creation of all figures from ROOT macros can be done automatically by issuing the command

```
./runfigure.sh ./events/lhyquid2dbi-RHICAuAu200c2030Ti455ti025Tf145/
```

All executables, i.e. binaries and shell scripts, produce help information when called with parameter `-h` or `--help`, for instance,

```
./therm2_events -h
./runfigure --help
```

5.6 Auxiliary files

After the program `therm2_events` is started, the file `event_*.tmp` is created in the main directory of the THERMINATOR 2 distribution. This file has two lines with the information about the subdirectory with the event files generated by the current run and the total number of event files located in that directory, including files from the previous runs. The name of the temporary file ends with the shell parent process ID (PPID), i.e. the PID number of the shell script that has started the `therm2_events` process. The auxiliary `event_*.tmp` file is used by scripts in order to be able to run multiple THERMINATOR 2 processes at once, e.g., on a computer cluster.

6 Scripts to generate physical results

We provide several ROOT scripts that may be used to generate physical results. These scripts read the chain of `event*.root` files and produce figures and ROOT figure macros. To work properly, the scripts must read from the sequence of event files generated by THERMINATOR 2, starting from `event000.root`. This means that the user should not accidentally delete event files from the middle of a sequence.

The transverse-momentum spectra are obtained by running the `figure_sp.C` script in ROOT. To use n event files, the user executes the following command (from the `therminator2` directory):

```
root -x './macro/figure_distpt.C("./events/
lhyquid2dbi-RHICAuAu200c2030Ti455ti025Tf145/",n)'
```

where n is the requested number of the event files to be processed.

For the elliptic flow coefficient, v_2 , the user should run

```
root -x './macro/figure_v2pt.C("./events/
lhyquid2dbi-RHICAuAu200c2030Ti455ti025Tf145/",n)'
```

This computes the elliptic flow coefficient, v_2 , as a function of the transverse momentum, p_T , in the reaction plane set by the hydrodynamics. The script for the fourth-order Fourier flow coefficient, v_4 , called `figure_v4pt.C`, is run analogously.

7 HBT analysis

This section describes the new functionality of THERMINATOR 2, allowing to compute the HBT correlation radii with the two-particle method.

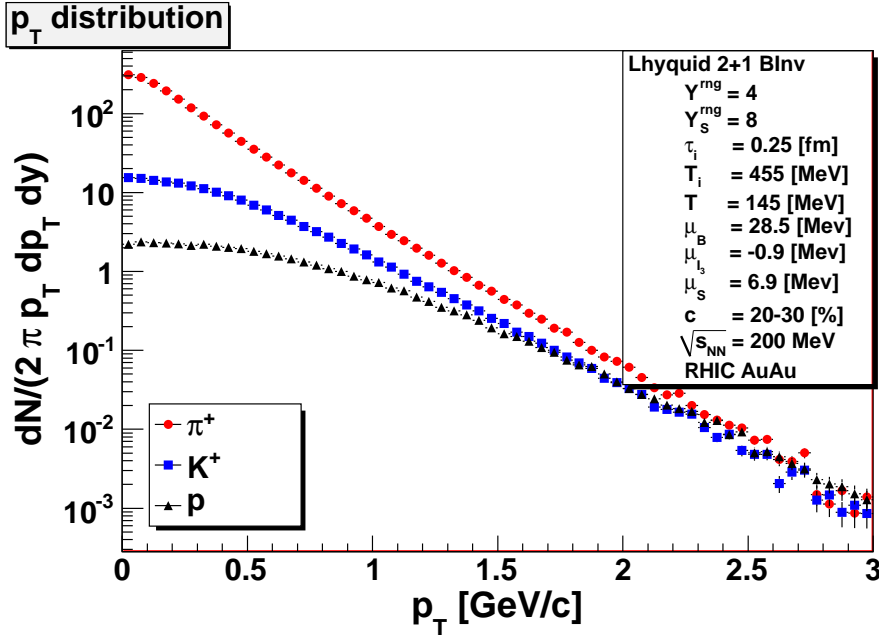


Fig. 8. Transverse-momentum spectra of π^+ , K^+ , and protons for Au+Au collisions at $\sqrt{s_{NN}} = 200$ GeV and the centrality class 20-30%, `fig_distpt.eps` (protons from Λ decays are excluded).

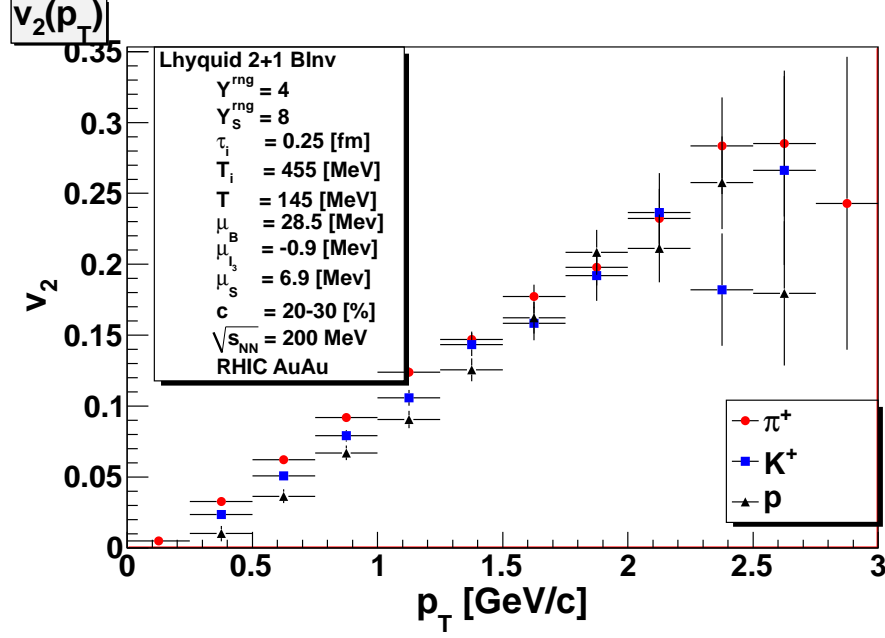


Fig. 9. The elliptic flow coefficient, v_2 , for π^+ , K^+ , and protons at the RHIC top energy $\sqrt{s_{NN}} = 200$ GeV and the centrality class 20-30%, `fig_v2pt.eps`.

7.1 Method

The method of carrying out the femtoscopic analysis is the same as described in detail in Ref. [5]. For integrity of this paper, we present below the main points.

Consider the two-particle distribution expressed by the two-particle emission function,

$$W_2(\vec{p}_1, \vec{p}_2) = E_{p_1} E_{p_2} \frac{dN}{d^3 p_1 d^3 p_2} \int S(x_1, x_2, p_1, p_2) d^4 x_1 d^4 x_2. \quad (21)$$

The correlation function is then defined as

$$C(\vec{p}_1, \vec{p}_2) = \frac{W_2(\vec{p}_1, \vec{p}_2)}{W_1(\vec{p}_1) W_1(\vec{p}_2)}, \quad (22)$$

where

$$W_1(\vec{p}) = E_p \frac{dN}{d^3 p} = \int d^4 x S(x, p), \quad (23)$$

with $S(x, p)$ being the source emission function.

We define the momentum difference of the particles,

$$q = (q_0, \vec{q}) = (E_{p_1} - E_{p_2}, \vec{p}_1 - \vec{p}_2), \quad (24)$$

the sum of their momenta,

$$P = (P_0, \vec{P}) = (E_{p_1} + E_{p_2}, \vec{p}_1 + \vec{p}_2), \quad (25)$$

and the average momentum of the pair,

$$k = (k_0, \vec{k}) = \frac{1}{2} (E_{p_1} + E_{p_2}, \vec{p}_1 + \vec{p}_2). \quad (26)$$

For the particles with equal masses we use the notation

$$\vec{q}_{\text{inv}} = 2\vec{k}^*. \quad (27)$$

The space and time separations of the members of the pair are $\vec{r} = \vec{r}_1 - \vec{r}_2$ and $\Delta t = t_1 - t_2$. When calculated in the Pair Rest Frame (PRF), they are denoted as \vec{r}^* and Δt^* . Both \vec{k}^* and \vec{r}^* appear as the arguments of the wave function, since PRF is the natural reference frame for the representation of the wave-function structure.

In general, the HBT analysis may be performed in any reference frame. One determines the correlation function as a function of the relative momentum components in the selected frame. Then the inverse widths of the correlation functions yields the size parameters of the system in this frame. Here we use the standard Bertsch-Pratt decomposition [124,125] of the mean and relative three-momenta into the three components. The *long* axis coincides with the beam axis, the *out* axis is determined by the direction of the average transverse momentum of the pair, denoted later by \vec{k}_T , and the *side* direction is perpendicular to the other two axes. Following the RHIC experiments, we choose to perform the analysis in the longitudinal co-moving system (LCMS), which is defined as a system where $k_{\text{long}} = 0$. In this work the notation is used in which the values in PRF are denoted by an asterisk, while the values without asterisk are defined in LCMS.

By definition, in a Monte-Carlo method the numerical equivalent of the integrals (21,23) is the summation over particles or pairs of particles generated by the Monte-Carlo procedure. The numerical calculation of the correlation functions is done in bins, which may be expressed with the help of the function

$$\delta_{\Delta}(\vec{p}) = \begin{cases} 1 & \text{if } |p_x| \leq \frac{\Delta}{2}, |p_y| \leq \frac{\Delta}{2}, |p_z| \leq \frac{\Delta}{2} \\ 0 & \text{otherwise.} \end{cases} \quad (28)$$

Then the correlation function may be expressed as

$$C(\vec{q}, \vec{k}) = \frac{\sum_i \sum_{j \neq i} \delta_{\Delta}(\vec{q} - \vec{p}_i + \vec{p}_j) \delta_{\Delta}(\vec{k} - \frac{1}{2}(\vec{p}_i + \vec{p}_j)) |\Psi(\vec{k}^*, \vec{r}^*)|^2}{\sum_i \sum_j \delta_{\Delta}(\vec{q} - \vec{p}_i + \vec{p}_j) \delta_{\Delta}(\vec{k} - \frac{1}{2}(\vec{p}_i + \vec{p}_j))}. \quad (29)$$

Various analyses of the HBT correlations use different approximations for the full pionic wave function, Ψ . In the non-interacting system, or in the interacting but non-relativistic case, the motion of the center of mass can be separated and one deals with the relative motion only. The simplest relative wave function ignores all dynamical interactions and has the form

$$\Psi = \frac{1}{\sqrt{2}}(e^{i\vec{k}^* \cdot \vec{r}^*} + e^{-i\vec{k}^* \cdot \vec{r}^*}), \quad (30)$$

where symmetrization over the two identical particles has been performed. Therefore $|\Psi|^2 = 1 + \cos(2\vec{k}^* \cdot \vec{r}^*)$.

Correlation functions calculated according to (29) represent the ideal Bose-Einstein correlation functions. They are also very useful in model studies, because they can be calculated analytically for simple Gaussian emission functions.

In addition, in the applied method *mixing* of particles is introduced when evaluating the correlation functions. The purpose of generating mixed events is to break all correlations other than those originating from femtoscopy, for instance the correlations due to resonances. The details are explained carefully in Sect. V of [48]. For the case of identical charged particles this mixing technique is innocuous, as there are no resonances producing two identical same-charge particles. In addition, the mixing method increases the statistics, which is highly desirable in our simulations. The calculation without mixing is not implemented in the present code.

The inclusion of Coulomb effects is described in [5].

7.2 Calculation of correlation functions extraction of the HBT radii

The correlation functions are obtained through a numerical implementation of Eqs. (29) and (30) (no Coulomb effects). As described in the previous sections, particles generated by **THERMINATOR 2** are grouped into events, as in experiment. In each event every charged pion is combined with every other pion of the same charge. For each pion pair, $|\Psi|^2$ is calculated and added to the numerator of Eq. (29) in a bin corresponding to the pair's q_{out} , q_{side} and

q_{long} . At the same time, 1 is added to the denominator of Eq. (29) in the corresponding bin. The resulting ratio yields the correlation function.

By making a proper selection of single pions and pairs of pions one may study the correlation functions as functions of various variables. For instance, taking into account the pairs of particles within a certain total momentum range only, one obtains femtoscopic information for a given k_T .

Usually, the single-particle emission function is postulated to be a static 3-dimensional ellipsoid with a Gaussian density profile

$$S(\vec{x}, \vec{p}) = N \exp \left(-\frac{x_{\text{out}}^2}{2R_{\text{out}}^2} - \frac{x_{\text{side}}^2}{2R_{\text{side}}^2} - \frac{x_{\text{long}}^2}{2R_{\text{long}}^2} \right). \quad (31)$$

This source function is static, i.e., it does not depend on particle momentum. In this case the integral leads to the well known formula:

$$C(k_{\perp}, q_{\text{out}}, q_{\text{side}}, q_{\text{long}}) = 1 + \lambda \exp \left[-R_{\text{out}}^2(k_{\perp})q_{\text{out}}^2 - R_{\text{side}}^2(k_{\perp})q_{\text{side}}^2 - R_{\text{long}}^2(k_{\perp})q_{\text{long}}^2 \right]. \quad (32)$$

The quantities R_{out} , R_{side} and R_{long} , known as the “HBT radii”, are the widths of the Gaussian approximation. It is important to emphasize that formula (32) is commonly used to fit the experimental data and to represent the results of the model calculations, although the experimental or model emission functions are frequently far from gaussians. In the simple fitting examples given in this paper we do not address this problem. More detailed discussion of such effects can be found in [5].

7.3 *therm2_femto and therm2_fithbt programs*

The program that implements the two-particle femtoscopic formalism described above, see Eq. (29), is called **therm2_femto**. The syntax of the command is:

```
./therm2_hbt <KTBIN> <EVENT_DIR> <EVENT_FILES> [FEMTO_INI] [PPID]
```

where the parameter **KTBIN=0,1,2,3** selects the transverse-momentum bin of the pair, **EVENT_DIR** is the directory where the **event*.root** files are stored, and **EVENT_FILES** is the number of the files to be taken. Parameter **FEMTO_INI** (optional) is the name configuration file. This parameter is by default is set to **femto.ini**. The **PPID** parameter (optional) is the system’s process ID number used by the shell scripts. By default it is equal to 0.

The k_T bins correspond to the following ranges of the pair momentum:

```
0 - (0.15, 0.25) GeV/c,
1 - (0.25, 0.35) GeV/c,
2 - (0.35, 0.45) GeV/c,
3 - (0.45, 0.6) GeV/c.
```

The output of the `therm2_femto` code consists of the ROOT files `femto*.root`, with histograms of the numerator and denominator,

which can be used to calculate the correlation function. In addition we provide the projections of the pair emission function S on the three components of the pair relative separation r calculated in LCMS. They are saved as 2D histograms, with the magnitude of relative momentum q_{inv} at the other axis.

The `therm2_hbtfite` program is an implementation of the fitting procedure, which uses the simplest fitting functional form (32). The syntax of the commands is:

```
therm2_hbtfite <CORR_FILE> [HBTFIT_INI] [PPID]
```

The first parameter `CORR_FILE` is the file containing the numerator and denominator histograms. The second parameter (optional) is set by default to `./fithbt.ini`. It is the name of the configuration file, containing the fitting parameters and switches. The third parameter (optional) is the process ID number, used by shell scripts.

The naming convention for the parameters is as follows: The label `[_fit]` indicates a parameter to be fitted. The name of the parameter, e.g., `Rout`, is assigned its initial value from which the fit starts. The `*FitType` flag defines the type of fitting. If it is set to `fixed`, then this value is fixed to the starting value in the fitting procedure. If `limit` is given, the value can vary in the range given by the `*Min` and `*Max` parameters. Finally, if it is equal to `free` than the value can be fitted without constraints.

The five parameters, `Norm` (set to 1), `Lambda`, `Rout`, `Rside`, and `Rlong`, correspond to the parameters of Eq. (32): the overall normalization, the λ parameter, and the three HBT radii (R_{out} , R_{side} , and R_{long}). The parameter `MaxFitRange` contains the range of q values in which the fitting is performed.

The output of the `therm2_hbtfite` program is a text file `hbtradii.txt`. It contains the values of the fitted parameters and statistical errors. Subsequent runs of the program append information to this file. The macro `macro/figure_hbtradii.C` reads that information and creates a plot of the HBT radii. The `therm2_hbtfite` program also produces the output ROOT file `hbtfite*.root` which should be inspected in order to judge the quality of the fit. It contains the one-dimensional slices of the calculated correlation function, as well as the corresponding slices

of the fitted function.

The histograms are named in the following way:

`<Type>_<Direction>_<Width>`

where **Type** can be **CFproj** for the correlation function slices or **FITptoj** for the fitted function slices. **Direction** is one of **out**, **side** or **long**, while **Width** can have a value of 1 - corresponding to a slice which has a width of 1 bin in the other two directions, 2 - a slice with the width of 4 bins and 3 - a slice with the width of 10 bins. The slices of the correlation function and the fitted function for the same direction and the same width should be plotted on top of each other in order to confirm that the fitting procedure correctly captured the width of the correlation. This is done by the macro `macro/figure_fitfemto.C`.

We warn the user that obtaining statistically stable results for the HBT radii requires sometimes a very large number of events, such as 20000 or even 50000. This requires an appropriate change of the parameter **NumberOfEvents** in the `events.ini` file, leading to long execution times.

The running sequence of the femtosopic analysis has been automatized in the shell script `./runhbt.sh`. The user should simply execute

```
./runhbt.sh
```

which in addition generates the physical plots. An example is presented in Fig. 10 and Fig. 11.

8 Yielding auxiliary results

In this Section we present new analyses, possible with **THERMINATOR 2**. In fact, they relate to fundamental physical aspects of the freeze-out model assumed in our approach.

8.1 *Backflow*

The first effect concerns the so-called back-flow problem [126–128]. It originates from the fact that for typical freeze-out hypersurfaces generated with hydrodynamics, particles generated with the Cooper-Frye formula may flow back into the hydrodynamics domain, thus violating causality. A typical remedy is the removal (used also in **THERMINATOR 2**) of the back-flowing particles

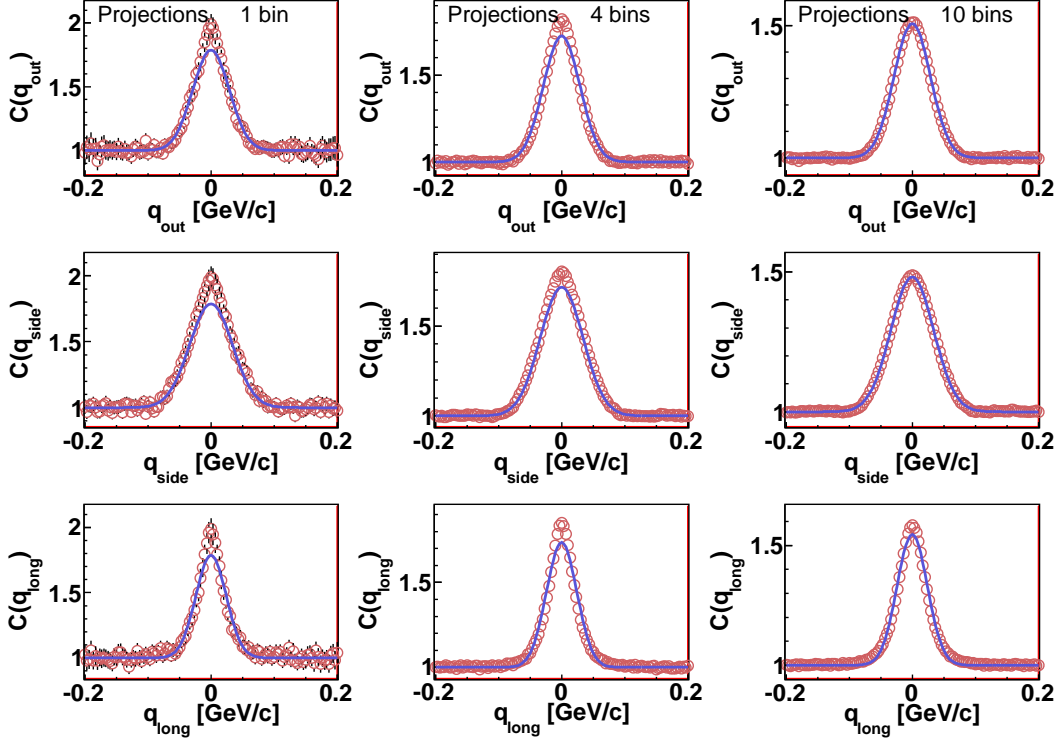


Fig. 10. Projections of the correlation function (symbols) calculated with the default settings for pions with k_T in the (0.15, 0.25 GeV) bin. Au+Au, $\sqrt{s_{NN}} = 200$ GeV, centrality class 20-30%, 50000 events (fig_cfprojpipi0a.eps). The lines show the fit projected the same way.

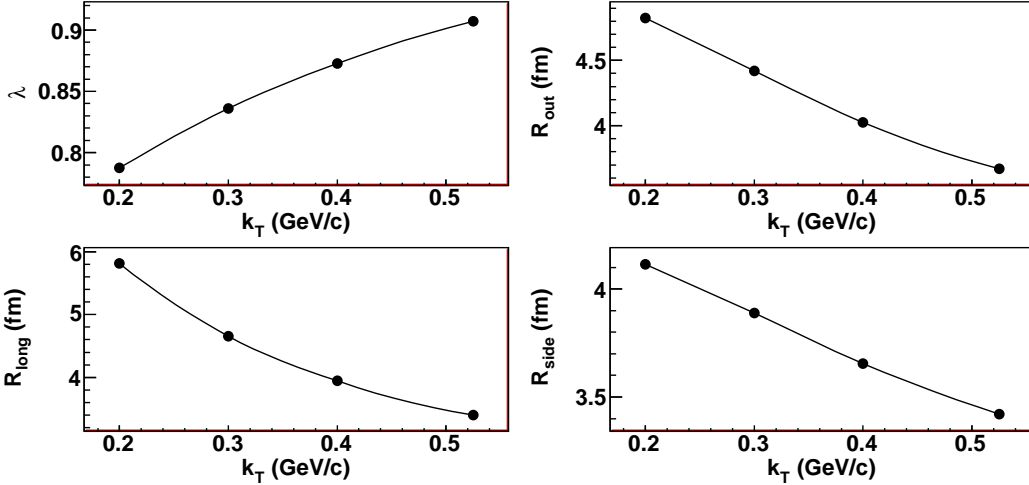


Fig. 11. The HBT radii and λ as a function of the average pair k_T calculated with the default settings, Au+Au, $\sqrt{s_{NN}} = 200$ GeV, centrality class 20-30%, 50000 mixed events (fig_hbtradii.eps).

via the constraint $p^\mu d\Sigma_\mu > 0$, where $d\Sigma_\mu$ is normal to the hypersurface. To see how big is the back-flow effect, we compute the back-flowing particles requesting that $p^\mu d\Sigma_\mu < 0$, and compare the result to the case without this

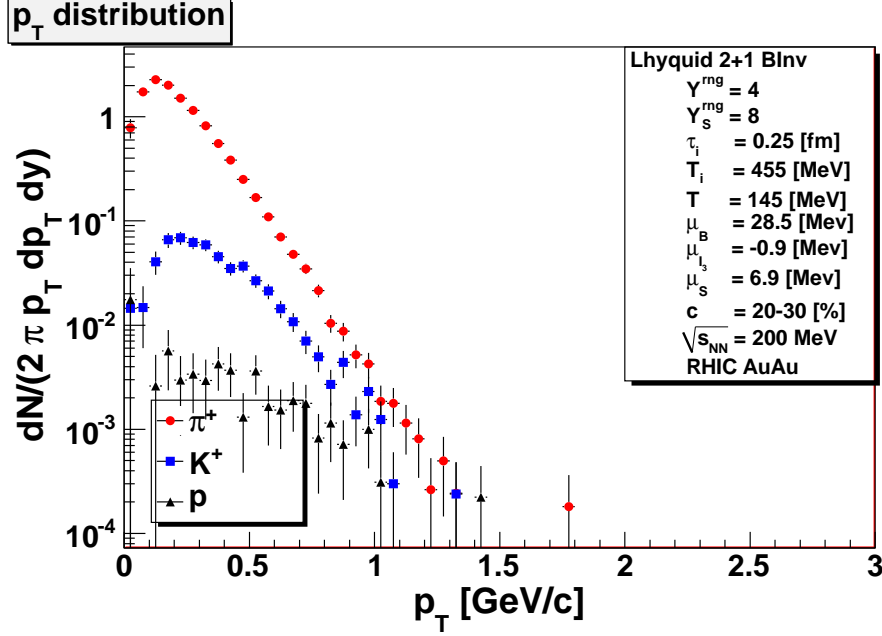


Fig. 12. Same as Fig. 8 for the back-flowing particles, 5000 events.

constraint, i.e. including all particles.

To prepare the binary file of `therm2_events` generating the back-flowing particles only, the user should run (this feature is implemented for the `Lhyquid2dbi` and `Lhyquid3d` models only)

```
make all BACK_FLOW=1
```

and then execute

```
./therm2_events
```

The user should take care of the fact that the directory specified with `EventSubDir` does not include the `fmultiplicity_*.txt` file, such that the multiplicities are recomputed for the present case of the back-flowing particles.

The result is shown in Fig. 12. Comparing with Fig. 8, we note that the back-flow effect is very small (less than 0.5% for the pions and even less for more massive particles) in our realistic calculation. This smallness is caused by large transverse velocity of the fluid, which inhibits the thermal motion of particles in the opposite direction [22].

8.2 Elastic rescattering

The other considered effect concerns the elastic rescattering after freeze-out. In our single-freeze-out approximation it is assumed that the chemical and

thermal freeze-outs occur simultaneously. As a matter of fact, this is not the case, as the inelastic processes, responsible for the chemical equilibrium, have generally a smaller cross section than the elastic processes. In order to estimate how good this approximation is, one may consider a simple estimate of elastic rescattering based on trajectory crossings.

Consider two particles, born at the freeze-out hypersurface at coordinates (t_1, \vec{x}_1^0) and (t_2, \vec{x}_2^0) . Their trajectories are

$$\vec{x}_1(t) = \vec{x}_1^0 + (t - t_1)\vec{v}_1, \quad \vec{x}_2(t) = \vec{x}_2^0 + (t - t_2)\vec{v}_2, \quad (33)$$

and the closest approach occurs at

$$t_m = -\frac{(\vec{v}_1 - \vec{v}_2) \cdot (\vec{x}_1^0 - \vec{x}_2^0)}{(\vec{v}_1 - \vec{v}_2) \cdot (\vec{v}_1 - \vec{v}_2)}. \quad (34)$$

Since this time cannot be earlier than t_1 or t_2 , we define $t_0 = \max(t_m, t_1, t_2)$. The distance squared at the closest approach is $d^2 = (\vec{x}_1(t_0) - \vec{x}_2(t_0))^2$. The collision occurs if, geometrically, $\pi d^2 < \sigma_{\text{el}}(s)$, where $\sigma_{\text{el}}(s)$ is the elastic cross section for the Mandelstam variable s .

We have analyzed in detail the case of a charged pion scattering elastically off the other pions. The elastic cross section is computed with the help of the realistic parametrization of the pion-pion phase shifts from [129]. The result is that for most-central collisions the pion scatters elastically off the other pion on the average 1.8 times, while for the mid-central and peripheral collisions this value drops below 1. This shows that for the pions the effects of elastic rescattering after freeze-out are small and our approximation is appropriate.

The user may carry out this analysis by running

```
cd addons
cp <dest_dir>/event000.root .
make -f coll.mk
./collcount
```

where `<destdir>` denotes the directory with the previously generated event files for the model and centrality of interest.

The code `collcount.cxx` may also serve as a simple example of an “off-line” analysis with the THERMINATOR 2 events. Its structures (e.g., concerning the reading of events) may be directly imitated in similar applications.

9 Conclusion

We hope that **THERMINATOR 2** will continue to be a useful and versatile tool for the heavy-ion community. In particular, the possibility of implementing the general freeze-out conditions (read from external files) allows the user to investigate in detail the predictions of the hydrodynamic approach followed with statistical hadronization (implemented in the single-freeze-out scenario). This approach was very successful in reproducing, in a uniform way, the RHIC data [22] and the LHC data at $\sqrt{s_{NN}} = 2.76$ TeV [130–132]. It was also very helpful in a variety of analyses [2–63]. Therefore, it has proved to be realistic and may serve as a practical tool in other physical studies or in detector modeling.

A new part of the package, **FEMTO-THERMINATOR**, carries out the femtoscopic analysis using the two-particle method. This is of practical importance, as complete analyses of the heavy-ion data typically consist of the studies of spectra, flow, and femtoscopy – all are now implemented in **THERMINATOR 2**.

We note that a small modification of the approach presented in this paper, accounting for viscosity effects which draw the distribution functions away from the thermal equilibrium, may be included in the straightforward way. That way the code can be used as an afterburner in the increasingly popular hydrodynamic studies with viscosity effects [19, 134–148]. Fugacities, reflecting incomplete chemical equilibrium [133] of the system, can now be also incorporated. This allows for model studies of this phenomenon.

We have made a considerable effort to make the structure of the code clear and straightforward to adapt for other applications. The program is documented in detail, also via the Doxygen utility. Numerous **ROOT** macros, allowing to carry out various tasks such as the calculation and presentation of physical results, are supplied with the package.

We foresee the following further extensions and applications of the program:

- (1) Statistical hadronization for *viscous* hydrodynamics – in this case the Cooper-Frye formula requires a straightforward modification of the form of the distribution function f .
- (2) Implementation of more advanced fitting procedures of the femtoscopic correlation functions, for instance imaging [149]. We note that even at present the output from **FEMTO-THERMINATOR** contains all necessary information for such analyses (the 3-dimensional correlation functions).
- (3) Analysis of the triangular and multipole flow, analysis of the two-particle correlations other than those due to the Bose-Einstein statistics.
- (4) Implementation of a more general form (*e.g.*, negative binomial) of the multiplicity distributions. This will allow for a realistic modeling of event-

- by-event multiplicity fluctuations.
- (5) Conservation laws. A modification of the particle-generation algorithm might include (approximate) conservation of the global charges and the transverse momentum.
 - (6) Interface to other formats for the files parameterizing hypersurfaces will be developed according to demand of potentially interested users.

Acknowledgments

We thank Piotr Bożek for providing the files with hypersurfaces from the 3+1-dimensional perfect hydrodynamics.

A Particle data files (`particles.data` and `decays.data`)

The structure of input files containing the information on the particle properties and their decays is exactly the same as in the original code [1] — the form inherited from `SHARE` [82]. To maintain integrity, we provide here the most basic information. The input file `particles.data` contains the properties of particles such as mass, width, spin, isospin, the quark contents, and the Monte Carlo identification number. The file has the format:

```
name mass width spin I I3 q s aq as c ac MC
```

where:

name the particle label used in the program,
mass mass in GeV,
width width in GeV,
spin spin,
I isospin,
I3 3rd component of isospin,
q, s number of light/strange quarks,
aq, as number of light/strange antiquarks,
c, ac number of charmed/anticharmed quarks,
MC particle's identification number, where available corresponding to the standard Monte Carlo particle identification convention [150].

The file `decays.data` contains the information on particle decay channels in the format:

```
Nameparent Namedaughter1 Namedaughter2 Namedaughter32 BR C-G?(0/1)
```

where **BR** denotes the branching ratio of the decay and **C-G** refers to whether the branching ratio should be multiplied by a Clebsch–Gordan coefficient (**0**: no, **1**: yes). Normally, we use **1** for the two-body decays and **0** for the three-body decays, where the full (isospin-dependent) branching ratio BR is provided. The entry BR may be used to control the feed-down from the weak decays, in particular setting it to 0 would switch off the particular decay channel.

² Appears for the three-body decays only.

B General definitions

Variables used to describe the properties of a particle:

ParticleType.h	symbol	definition
mNumber		particle type number
mName		particle name
mMass	m	mass
mGamma	Γ	width
mSpin	s	spin
mBarionN	B	baryon number
mI	I	isospin
mI3	I_3	3rd component of isospin
mStrangN	S	strangness
mCharmN	C	charm
mNq	n_q	number of u and d quarks
mNaq	$n_{\bar{q}}$	number of \bar{u} and \bar{d} quarks
mNs	n_s	number of s quarks
mNas	$n_{\bar{s}}$	number of \bar{s} quarks
mNc	n_c	number of c quarks
mNac	$n_{\bar{c}}$	number of \bar{c} quarks
mPDGCode		particle MC number according to PDG [150]
mMaxIntegrand		maximum value of the Cooper-Frye integrand
mMultiplicity		average multiplicity
mDecayChannelCount2		number of two-body decay channels
mDecayChannelCount3		number of three-body decay channels.

Variables used to calculate the Cooper-Frye integrand in the `Model_*.cxx` files. See function `GetIntegrand(ParticleType* aPartType)`.

in program	symbol	definition
dSigmaP	$p \cdot d\Sigma$	particle's four momentum projected on the hypersurface element
PdotU	$p \cdot u$	particle's energy in the fluid local rest frame
Statistics	± 1	+1 (for fermions), -1 (for bosons)
<i>space-time</i>		
(Xt, Xx, Xy, Xz)	(t, x, y, z)	Cartesian coordinates
Tau	τ	proper time
Rho	ρ	transverse size
PhiS	ϕ	azimuthal angle
RapS	Y_s	space-time rapidity
<i>momentum</i>		
(Pe, Px, Py, Pz)	(E, p_x, p_y, p_z)	Cartesian coordinates
Mt	m_T	transverse mass
Pt	p_T	transverse momentum
PhiP	ϕ_p	transverse-momentum angle
RapP	Y	rapidity
<i>fluid element</i>		
Ux, Uy	u_x, u_y	transverse four-velocity
Vt	v_T	transverse velocity
PhiF	ϕ_f	transverse-velocity angle
RapF	Y_f	fluid rapidity.

C Hydro-inspired models

Common parameters for hydro-inspired models:

ini file	in program	symbol	description
RapPRange	mRapPRange	Y^{rng}	rapidity range
RapSRange	mRapSRange	Y_s^{rng}	spatial-rapidity range
RhoMax	mRhoMax	ρ^{max}	firecylinder transverse size
Temperature	mTemp	T	freeze-out temperature
MuB	mMuB	μ_B	baryon chemical potential
MuI	mMuI	μ_{I_3}	isospin chemical potential
MuS	mMuS	μ_S	strange chemical potential
MuC	mMuC	μ_C	charm chemical potential.

Integration ranges

$$\left. \begin{array}{l} Y_s \in \left[-\frac{Y_s^{\text{rng}}}{2}, \frac{Y_s^{\text{rng}}}{2}\right] \\ \rho \in [0, \rho_{\text{max}}] \\ \phi \in [0, 2\pi] \end{array} \right| \left. \begin{array}{l} Y \in \left[-\frac{Y^{\text{rng}}}{2}, \frac{Y^{\text{rng}}}{2}\right] \\ p_T \in [0, \infty) \\ \phi_p \in [0, 2\pi] \end{array} \right.$$

C.1 Cracow Single Freeze-out Model

Model specific parameters:

fomodel/krakow.ini	in program	symbol	description
TauC	mTauC	τ_f	Cracow proper time.

Definition of the proper time used in the Cracow model

$$\tau_f = \sqrt{t^2 - x^2 - y^2 - z^2} = \sqrt{\tau^2 + \rho^2}. \quad (\text{C.1})$$

C.2 Blast-Wave

Model specific parameters:

fomodel/blastwave.ini	in program	symbol	description
Tau	mTau	τ	proper time
VelT	mVt	v_T	transverse velocity of the fluid element.

C.3 Blast-Wave A-Class

Model specific parameters:

fomodel/bwa.ini	in program	symbol	description
Tau	mTau	τ	proper time
VelT	mVt	v_T	transverse velocity of the fluid element
ParA	mA	A	the A parameter in Eq. (14)
Delay	mDelay	λ	emission delay

If the `events.ini` file has the **FreezeOutModel** option set to: **BWAVLinear**, **BWAVLinearDelay**, or **BWAVLinearFormation**, the value of the fluid transverse velocity has a linear form

$$\tilde{v}_T = \frac{\rho/\rho_{\text{max}}}{v_T + \rho/\rho_{\text{max}}}. \quad (\text{C.2})$$

Additionally, for the models **BWAVTDelay** and **BWAVLinearDelay** the particle's

creation time is shifted randomly according to the exponential distribution

$$f(t) = e^{-\frac{t-t_0}{\lambda}}, \quad (\text{C.3})$$

where t_0 is the original time of the particle's creation.

The model `BWAVLinearFormation` incorporates the particle's formation time and shifts the position of the particle,

$$\tilde{x} = x + \frac{p_x}{E} t, \quad \tilde{y} = y + \frac{p_y}{E} t, \quad \tilde{z} = z + \frac{p_z}{E} t.$$

D Freeze-out models based on 2+1 and 3+1 hydrodynamic calculations

Common parameters in hydro-based models:

lhyquid*.ini	in program	symbol	description
RapidityRange	mRapRange	Y^{rng}	rapidity range
*.xml file			
<PARAMETER>			
Tau_i	mTauI	τ_i	initial proper time
Temperature	mTemp	T	freeze-out temperature
Mu_B	mMuB	μ_B	baryon chemical potential
Mu_I	mMuI	μ_{I_3}	isospin chemical potential
Mu_S	mMuS	μ_S	strange chemical potential
Mu_C	mMuC	μ_C	charm chemical potential
device	mDeviceName		accelerator's name, e.g., RHIC, LHC
colliding_system	mCollidingSystem		colliding nuclei, e.g., Au, Pb
colliding_energy	mCollidingEnergy	$\sqrt{s_{NN}}$	center-of-mass energy
centrality_min	mCentralityMin	c_{min}	centrality lower limit
centrality_max	mCentralityMax	c_{max}	centrality higher limit
impact_parameter	mImpactParameter	b	impact parameter
temperature_at_center	mTempl	T_i	central initial temperature
<VECTOR3D>			
Distance	mDistance	$d(\zeta, \phi, \theta)$	distance to the hypersurface element
DistanceDZeta	mDistanceDZeta	$\frac{\partial d}{\partial \zeta}$	ζ -derivative of d
DistanceDPhi	mDistanceDPhi	$\frac{\partial d}{\partial \phi}$	ϕ -derivative of d .

Phase-space integration ranges:

$$Y_s \in \left[-\frac{Y_s^{\text{rng}}}{2}, \frac{Y_s^{\text{rng}}}{2} \right] \text{ or } \left. \begin{array}{l} \theta \in [\theta_{\min}, \theta_{\max}] \\ \zeta \in [\zeta_{\min}, \zeta_{\max}] \\ \phi \in [\phi_{\min}, \phi_{\max}] \end{array} \right| \left. \begin{array}{l} Y \in \left[-\frac{Y^{\text{rng}}}{2}, \frac{Y^{\text{rng}}}{2} \right] \\ p_T \in [0, \infty) \\ \phi_p \in [0, 2\pi]. \end{array} \right.$$

Note that in the **Lhyquid3D** case the space-time rapidity Y_s is parametrized by the θ angle, $Y_s = (d/\Lambda) \cos \theta$.

D.1 Lhyquid 3D

Model specific parameters:

*.xml file	in program	symbol	description
<PARAMETER>			
Lambda	mLambda	Λ	conversion constant in $Y_s = (d/\Lambda) \cos \theta$
<VECTOR3D>			
DistanceDTheta	mDistanceDTheta	$\frac{\partial d}{\partial \theta}$	θ -derivative of d
FluidUx	mFluidUx	$u_x(\zeta, \phi, \theta)$	fluid four-velocity x -component
FluidUy	mFluidUy	$u_y(\zeta, \phi, \theta)$	fluid four-velocity y -component
FluidRapidity	mFluidRapidity	$Y_f(\zeta, \phi, \theta)$	fluid rapidity.

D.2 Lhyquid 2D boost-invariant

Model specific parameters:

lhyquid2dbi.ini	in program	symbol	description
RapSRange	mRapSRange	Y_s^{rng}	spatial-rapidity range
*.xml file			
<VECTOR3D>			
FluidVt	mFluidVt	$v_T(\zeta, \phi)$	fluid transverse velocity
FluidPhi	mFluidPhi	$\phi_f(\zeta, \phi)$	fluid velocity azimuthal angle.

References

- [1] A. Kisiel et al., Comput. Phys. Commun. 174 (2006) 669, nucl-th/0504047.
- [2] W. Broniowski et al., Phys. Lett. B635 (2006) 290, nucl-th/0510033.
- [3] S. Pratt and D. Schindel, AIP Conf. Proc. 828 (2006) 430, nucl-th/0511010.
- [4] A. Andronic, P. Braun-Munzinger and J. Stachel, Nucl. Phys. A772 (2006) 167, nucl-th/0511071.

- [5] A. Kisiel, W. Florkowski and W. Broniowski, Phys. Rev. C73 (2006) 064902, nucl-th/0602039.
- [6] M. Chojnacki and W. Florkowski, Phys. Rev. C74 (2006) 034905, nucl-th/0603065.
- [7] W. Florkowski et al., Acta Phys. Polon. B37 (2006) 3381, nucl-th/0609054.
- [8] B. Biedron and W. Broniowski, Phys. Rev. C75 (2007) 054905, nucl-th/0610083.
- [9] O. Utyuzh, G. Wilk and Z. Wlodarczyk, Braz. J. Phys. 37 (2007) 708, hep-ph/0610408.
- [10] V.V. Begun et al., Phys. Rev. C76 (2007) 024902, nucl-th/0611075.
- [11] O. Utyuzh, G. Wilk and Z. Wlodarczyk, Phys. Rev. D75 (2007) 074030, hep-ph/0702073.
- [12] D.A. Brown et al., Phys. Rev. C76 (2007) 044906, 0705.1337.
- [13] B. Baeuchle, (2007), 0706.4252.
- [14] PHENIX, R. Vertesi, (2007), 0706.4409.
- [15] P. Danielewicz, (2007), 0707.0377.
- [16] V.V. Begun, Phys. Atom. Nucl. 71 (2008) 1813, 0711.2912.
- [17] B. Tomasik et al., Acta Phys. Polon. Supp. 1 (2008) 513, 0712.0563.
- [18] M. Chojnacki et al., Phys. Rev. C78 (2008) 014905, 0712.0947.
- [19] P. Bozek, Phys. Rev. C77 (2008) 034911, 0712.3498.
- [20] PHENIX, S. Afanasiev et al., Phys. Rev. Lett. 100 (2008) 232301, 0712.4372.
- [21] Y. Akiba, Prog. Theor. Phys. Suppl. 174 (2008) 88.
- [22] W. Broniowski et al., Phys. Rev. Lett. 101 (2008) 022301, 0801.4361.
- [23] W. Florkowski et al., Acta Phys. Polon. B39 (2008) 1555, 0804.0974.
- [24] PHENIX, R.A. Lacey, J. Phys. G35 (2008) 104139, 0805.1352.
- [25] P. Chung and P. Danielewicz, Phys. Atom. Nucl. 71 (2008) 1552, 0807.4892.
- [26] A. Kisiel et al., (2008), 0808.3363.
- [27] NA49, C. Alt et al., (2008), 0809.1445.
- [28] B. Tomasik, Comput. Phys. Commun. 180 (2009) 1642, 0806.4770.
- [29] P. Bozek, Phys. Rev. C79 (2009) 054901, 0811.1918.
- [30] PHOBOS, B. Alver et al., (2008), 0812.1172.

- [31] W. Broniowski et al., Phys. Rev. C80 (2009) 034902, 0812.3393.
- [32] W. Broniowski et al., Acta Phys. Polon. B40 (2009) 979, 0812.4935.
- [33] W. Florkowski et al., Acta Phys. Polon. B40 (2009) 1093, 0901.1251.
- [34] Z. Chajecski, Acta Phys. Polon. B40 (2009) 1119, 0901.4078.
- [35] P. Bozek and I. Wyskiel, Phys. Rev. C79 (2009) 044916, 0902.4121.
- [36] X. Luo et al., Phys. Lett. B673 (2009) 268, 0903.0024.
- [37] P. Bozek and I. Wyskiel, Phys. Rev. C79 (2009) 044916.
- [38] J. Noronha-Hostler et al., (2009), 0906.3960.
- [39] W. Broniowski, M. Chojnacki and L. Obara, Phys. Rev. C80 (2009) 051902, 0907.3216.
- [40] W. Florkowski et al., Nucl. Phys. A830 (2009) 821c, 0907.3592.
- [41] P. Bozek and I. Wyskiel, (2009), 0909.2354.
- [42] M. Hauer and S. Wheaton, Phys. Rev. C80 (2009) 054915, 0909.2431.
- [43] R. Ryblewski and W. Florkowski, (2009), 0910.0985.
- [44] W. Broniowski et al., (2009), 0910.3585.
- [45] BRAHMS, F. Videbaek, Nucl. Phys. A830 (2009) 43c, 0907.4742.
- [46] P. Bozek, (2009), 0911.2392.
- [47] A. Kisiel and D.A. Brown, (2009), 0901.3527.
- [48] A. Kisiel, (2009), 0909.5349.
- [49] A. Kisiel and T.J. Humanic, (2009), 0908.3830.
- [50] W. Florkowski and R. Ryblewski, (2009), 0912.3451.
- [51] U.W. Heinz, (2009), 0901.4355.
- [52] P. Bozek and I. Wyskiel, (2009), 0903.3129.
- [53] I.P. Lokhtin et al., (2009), 0910.5129.
- [54] J. Noronha-Hostler, J. Noronha and C. Greiner, J. Phys. G37 (2010) 094062, 1001.2610.
- [55] X.F. Luo et al., J. Phys. G37 (2010) 094061, 1001.2847.
- [56] P. Bozek and I. Wyskiel, Phys. Rev. C81 (2010) 054902, 1002.4999.
- [57] STAR, M.M. Aggarwal et al., Phys. Rev. Lett. 105 (2010) 022302, 1004.4959.
- [58] M. Hauer, (2010), 1008.1990.

- [59] J. Noronha-Hostler and C. Greiner, (2010), 1008.5075.
- [60] P. Bozek and I. Wyskiel-Piekarska, (2010), 1009.0701.
- [61] A. Bilandzic, R. Snellings and S. Voloshin, (2010), 1010.0233.
- [62] C. Lizhu et al., (2010), 1011.0712.
- [63] P. Bozek, W. Broniowski and J. Moreira, (2010), 1011.3354.
- [64] R. Brun and F. Rademakers, Nucl. Instrum. Meth. A389 (1997) 81, (<http://root.cern.ch>).
- [65] W. Florkowski, W. Broniowski and M. Michalec, Acta Phys. Polon. B33 (2002) 761, nucl-th/0106009.
- [66] W. Broniowski and W. Florkowski, Phys. Rev. Lett. 87 (2001) 272302, nucl-th/0106050.
- [67] P. Danielewicz and Q.b. Pan, MSUCL-848.
- [68] E. Schnedermann, J. Sollfrank and U.W. Heinz, Phys. Rev. C48 (1993) 2462, nucl-th/9307020.
- [69] F. Retiere and M.A. Lisa, Phys. Rev. C70 (2004) 044907, nucl-th/0312024.
- [70] D. Teaney, J. Lauret and E.V. Shuryak, (2001), nucl-th/0110037.
- [71] T. Hirano and K. Tsuda, Phys. Rev. C66 (2002) 054905, nucl-th/0205043.
- [72] P.F. Kolb and U. Heinz, (2003), in Quark-Gluon Plasma 3, edited by R.C. Hwa and X.-N. Wang (World Scientific, Singapore, 2004), p. 634, nucl-th/0305084.
- [73] P. Huovinen, (2003), in Quark-Gluon Plasma 3, edited by R.C. Hwa and X.-N. Wang (World Scientific, Singapore, 2004), p. 600, nucl-th/0305064.
- [74] E.V. Shuryak, Nucl. Phys. A750 (2005) 64, hep-ph/0405066.
- [75] K.J. Eskola et al., Phys. Rev. C72 (2005) 044904, hep-ph/0506049.
- [76] Y. Hama et al., Nucl. Phys. A774 (2006) 169, hep-ph/0510096.
- [77] T. Hirano et al., Phys. Lett. B636 (2006) 299, nucl-th/0511046.
- [78] P. Huovinen and P.V. Ruuskanen, Ann. Rev. Nucl. Part. Sci. 56 (2006) 163, nucl-th/0605008.
- [79] T. Hirano et al., J. Phys. G34 (2007) S879, nucl-th/0701075.
- [80] C. Nonaka and S.A. Bass, Phys. Rev. C75 (2007) 014902, nucl-th/0607018.
- [81] P. Huovinen and P. Petreczky, (2009), 0912.2541.
- [82] G. Torrieri et al., Comput. Phys. Commun. 167 (2005) 229, nucl-th/0404083.
- [83] G. Torrieri et al., Comput. Phys. Commun. 175 (2006) 635, nucl-th/0603026.

- [84] S. Wheaton and J. Cleymans, Comput. Phys. Commun. 180 (2009) 84, hep-ph/0407174.
- [85] I.P. Lokhtin et al., Comput. Phys. Commun. 180 (2009) 779, 0809.2708.
- [86] I.P. Lokhtin et al., (2009), 0903.0525.
- [87] N.S. Amelin et al., Phys. Rev. C74 (2006) 064901, nucl-th/0608057.
- [88] N.S. Amelin et al., Phys. Rev. C77 (2008) 014903, 0711.0835.
- [89] P. Koch and J. Rafelski, South Afr. J. Phys. 9 (1986) 8.
- [90] J. Cleymans and H. Satz, Z. Phys. C57 (1993) 135, hep-ph/9207204.
- [91] J. Sollfrank et al., Z. Phys. C61 (1994) 659.
- [92] P. Braun-Munzinger et al., Phys. Lett. B344 (1995) 43, nucl-th/9410026.
- [93] P. Braun-Munzinger et al., Phys. Lett. B365 (1996) 1, nucl-th/9508020.
- [94] T. Csorgo and B. Lorstad, Phys. Rev. C54 (1996) 1390, hep-ph/9509213.
- [95] J. Cleymans et al., Z. Phys. C74 (1997) 319, nucl-th/9603004.
- [96] J. Rafelski, J. Letessier and A. Tounsi, Acta Phys. Polon. B27 (1996) 1037, nucl-th/0209080.
- [97] J. Rafelski, J. Letessier and A. Tounsi, Acta Phys. Polon. B28 (1997) 2841, hep-ph/9710340.
- [98] F. Becattini, J. Phys. G23 (1997) 1933, hep-ph/9708248.
- [99] G.D. Yen and M.I. Gorenstein, Phys. Rev. C59 (1999) 2788, nucl-th/9808012.
- [100] J. Cleymans and K. Redlich, Phys. Rev. Lett. 81 (1998) 5284, nucl-th/9808030.
- [101] M. Gazdzicki and M.I. Gorenstein, Acta Phys. Polon. B30 (1999) 2705, hep-ph/9803462.
- [102] M. Gazdzicki, Nucl. Phys. A681 (2001) 153, hep-ph/9910363.
- [103] P. Braun-Munzinger, I. Heppe and J. Stachel, Phys. Lett. B465 (1999) 15, nucl-th/9903010.
- [104] J. Cleymans and K. Redlich, Phys. Rev. C60 (1999) 054908, nucl-th/9903063.
- [105] F. Becattini et al., Phys. Rev. C64 (2001) 024901, hep-ph/0002267.
- [106] P. Braun-Munzinger et al., Phys. Lett. B518 (2001) 41, hep-ph/0105229.
- [107] W. Broniowski, A. and W. Florkowski, Acta Phys. Polon. B33 (2002) 4235, hep-ph/0209286.
- [108] W. Florkowski, W. Broniowski and P. Bozek, J. Phys. G30 (2004) S1321, nucl-th/0403038.

- [109] P. Bozek, W. Broniowski and W. Florkowski, *Acta Phys. Hung.* A22 (2005) 149, [nucl-th/0310062](#).
- [110] W. Broniowski, A. Baran and W. Florkowski, *AIP Conf. Proc.* 660 (2003) 185, [nucl-th/0212053](#).
- [111] W. Florkowski, W. Broniowski and A. Baran, (2004), [nucl-th/0412077](#).
- [112] Particle Data Group, C. Amsler et al., *Phys. Lett.* B667 (2008) 1.
- [113] R. Hagedorn, *Nuovo Cim. Suppl.* 3 (1965) 147.
- [114] R. Hagedorn and J. Ranft, *Nuovo Cim. Suppl.* 6 (1968) 169.
- [115] R. Hagedorn, Invited talk at NATO Advanced Study Workshop on Hot Hadronic Matter: Theory and Experiment, Divonne-les-Bains, France, 27 Jun - 1 Jul 1994.
- [116] W. Broniowski and W. Florkowski, *Phys. Lett.* B490 (2000) 223, [hep-ph/0004104](#).
- [117] W. Broniowski, (2000), [hep-ph/0008112](#).
- [118] W. Broniowski, W. Florkowski and L.Y. Glozman, *Phys. Rev.* D70 (2004) 117503, [hep-ph/0407290](#).
- [119] F. Cooper and G. Frye, *Phys. Rev.* D10 (1974) 186.
- [120] C.W. Misner, K.S. Thorne and J.A. Wheeler, (W.H. Freeman, San Francisco, 1973) 1279 p .
- [121] P.J. Siemens and J.O. Rasmussen, *Phys. Rev. Lett.* 42 (1979) 880.
- [122] <http://www.stack.nl/~dimitri/doxygen/>.
- [123] <http://www.clusterresources.com/products/torque-resource-manager.php>.
- [124] G. Bertsch, M. Gong and M. Tohyama, *Phys. Rev.* C37 (1988) 1896.
- [125] S. Pratt, *Phys. Rev.* D33 (1986) 1314.
- [126] K.A. Bugaev, *Nucl. Phys.* A606 (1996) 559, [nucl-th/9906047](#).
- [127] K.A. Bugaev and M.I. Gorenstein, (1999), [nucl-th/9903072](#).
- [128] K.A. Bugaev, *Phys. Rev. Lett.* 90 (2003) 252301, [nucl-th/0210087](#).
- [129] G. Colangelo, J. Gasser and H. Leutwyler, *Nucl. Phys.* B603 (2001) 125, [hep-ph/0103088](#).
- [130] The ALICE, K. Aamodt et al., (2010), 1011.3914.
- [131] P. Bozek, (2010), 1012.5927.
- [132] P. Bozek, (2011), 1101.1791.

- [133] J. Letessier and J. Rafelski, Phys. Rev. C59 (1999) 947, hep-ph/9806386.
- [134] A. Muronga, Phys. Rev. Lett. 88 (2002) 062302, nucl-th/0104064.
- [135] D. Teaney, Phys. Rev. C68 (2003) 034913, nucl-th/0301099.
- [136] R. Baier, P. Romatschke and U.A. Wiedemann, Phys. Rev. C73 (2006) 064903, hep-ph/0602249.
- [137] R. Baier and P. Romatschke, Eur. Phys. J. C51 (2007) 677, nucl-th/0610108.
- [138] P. Romatschke and U. Romatschke, Phys. Rev. Lett. 99 (2007) 172301, 0706.1522.
- [139] A.K. Chaudhuri, Phys. Rev. C74 (2006) 044904, nucl-th/0604014.
- [140] U.W. Heinz, H. Song and A.K. Chaudhuri, Phys. Rev. C73 (2006) 034904, nucl-th/0510014.
- [141] P. Bozek, Acta Phys. Polon. B39 (2008) 1375, 0711.2889.
- [142] H. Song and U.W. Heinz, Phys. Rev. C77 (2008) 064901, 0712.3715.
- [143] P. Bozek, Acta Phys. Polon. B39 (2008) 1539, 0803.4447.
- [144] H. Song and U.W. Heinz, (2008), 0812.4274.
- [145] M. Luzum and P. Romatschke, Phys. Rev. C78 (2008) 034915, 0804.4015.
- [146] P. Bozek, Acta Phys. Polon. B40 (2009) 987, 0901.2272.
- [147] P. Bozek, (2009), 0911.2397.
- [148] P. Romatschke, (2009), 0902.3663.
- [149] D.A. Brown and P. Danielewicz, Phys. Lett. B398 (1997) 252, nucl-th/9701010.
- [150] Particle Data Group, K. Hagiwara et al., Phys. Rev. D66 (2002) 010001.

Table 1

The contents of the file `events.ini`

```

# -----|-----|
#   FreezeOutModel      |      model parameter file      |
# -----|-----|
#   KrakowSFO           | ./fomodel/cracow.ini           |
#   BlastWave           | ./fomodel/blastwave.ini       |
#   BWAvt               | ./fomodel/bwa.ini             |
#   BWAvtDelay          | ./fomodel/bwa.ini             |
#   BWAvtLinear          | ./fomodel/bwa.ini             |
#   BWAvtLinearDelay     | ./fomodel/bwa.ini             |
#   BWAvtLinearFormation | ./fomodel/bwa.ini             |
#   Lhyquid3D           | ./fomodel/lhyquid3d.ini       |
#   Lhyquid2DBI         | ./fomodel/lhyquid2dbi.ini     |
# -----|-----|

# Name of the freeze-out model
# available: see table above
# default: Lhyquid2DBI
FreezeOutModel = Lhyquid2DBI

# Custom freeze-out model ini file [not used by default]
# default:
# FreezeOutModelINI = ./fomodel/lhyquid2dbi.ini

[Event]
# Number of events to generate
# default: 5000
NumberOfEvents = 5000

# Event output file format
# available: root, root&text, text
# default: root
EventFileType = root

[Primordial]
# Distribution of primordial particles multiplicity
# available: Poisson
# default: Poisson
MultiplicityDistribution = Poisson

# Number of samples used in determination of primordial multiplicity and max. integrand value
# default: 5000000
IntegrateSamples = 5000000

[Random]
# Start each event with a new random seed taken from current time (1) or do a constant seed (0)
# default: 1
Randomize = 1

[Directories]
# Directory with SHARE input files
# default: ./share/
ShareDir = ./share/

# Directory with Freeze-Out Model parameter files
# default: ./fomodels/
FreezeOutDir = ./fomodel/

# Directory with ROOT macro files *.C
# default: ./macro/
MacroDir = ./macro/

# Directory to write the events
# default: ./events/
EventDir = ./events/

[Logging]
# Log file - save information on number of events runned, destination and time
# default: terminator.log
LogFile = terminator.log

```

Table 2

Parameters of fomodel/lhyquid2dbi.ini

```
#####
#           Lhyquid 2+1 Boost Invariant           #
#####

# Rapidity range
# default: 4.0
RapPRange = 4.0

# Spatial rapidity range
# default: 8.0
RapSRange = 8.0

# location of the hydro-code output containing the definition of the hypersurface,
# velocity profile, thermodynamic properties etc.
# default: lhyquid2dbi/RHICAuAu200c2030Ti455ti025Tf145.xml
FreezeFile = lhyquid2dbi/RHICAuAu200c0005Ti500ti025Tf145.xml

# subdirectory to store events of this model.
# If option is hashed the EventSubDir will be made form FreezeFile name (replace '/' with '-')
# and remove ".xml")
# default: lhyquid2dbi-RHICAuAu200c2030Ti455ti025Tf145/
#EventSubDir = lhyquid2dbi-RHICAuAu200c2030Ti455ti025Tf145/
```

Table 3

Contents of the femto.ini configuration file.

```
[Pair]
# Type of particle pairs to correlate
# default: pion-pion, kaon-kaon
PairType = pion-pion

[Cuts]
# Particle time cut [fm]
# default: 500.0
TimeCut = 500.0

[Event]
# Number of events to mix
# default: 20
EventsToMix = 20

[Switches]
# Use only primordial particles (yes) or all particles from resonance decays (no)
# default: no
EnableOnlyPrimordial = no

# Enable source histograms
# default: no
EnableSourceHistograms = no

[Logging]
# Log file - save information on number of events runned, destination and time
# default: therminator.log
LogFile = therminator.log
```

Table 4

Contents of the `hbtfit.ini` file.

```
[Norm_Fit]
# Normalization initial value
# default:      1.0
Norm = 1.0
# Fit type
# default:      fixed
NormFitType = fixed
[Lambda_Fit]
# Lambda initial value
# default:      1.0
Lambda = 1.0
# Fit type
# default:      free
LambdaFitType = free
[Rout_Fit]
# R_out initial value
# default:
Rout = 5.0
# Fit type
# default:      limit
RoutFitType = limit
# Rout limits
# default:
RoutMin = 2.0
RoutMax = 7.0
[Rside_Fit]
# Rside initial value
# default:
Rside = 5.0
# Fit type
# default:      limit
RsideFitType = limit
# Rout limits
# default:
RsideMin = 2.0
RsideMax = 7.0
[Rlong_Fit]
# Rlong initial value
# default:
Rlong = 7.0
# Fit type
# default:      limit
RlongFitType = limit
# Rout limits
# default:
RlongMin = 4.0
RlongMax = 14.0
[QRange]
# Maximum range of q_out, q_side, q_long
#default:      0.15
MaxFitRange = 0.15
[Histograms]
# Numerator histogram name
# default:      cnuma
Numerator = cnuma
# Denominator histogram name
# default:      cdena
Denominator = cdena
[Logging]
# Log file
# default:      therminator.log
LogFile = therminator.log
```