THERMINATOR 2 [1]

authors: M. Ch., Adam Kisiel, Wojciech Florkowski, Wojciech Broniowski

“New make, new models, new mission”

Mikołaj Chojnacki
Institute of Nuclear Physics
Polish Academy of Sciences
Kraków, Poland

VI Workshop on Particle Correlations and Femtoscopy
14-18 September 2010

[1] http://www.ifj.edu.pl/dept/no4/nz41/therminator2

Outlook

- What *(or who?)* is `THERMINATOR 2`,
- Physics behind the code,
- How to use `THERMINATOR 2`,
- Possible extensions of `THERMINATOR 2`,
- FAQ: How to add my own model, and how can we help you in doing so.

THERMINATOR 2 *a.k.a.* TH2

THERMal heavy-**I**o**N** gener**ATOR** version **2**

- TH2 is a Monte-Carlo event generator, designed with only one purpose ...

THERMINATOR 2 *a.k.a.* TH2

THERMal heavy-**I**o**N** gener**ATOR** version **2**

- TH2 is a Monte-Carlo event generator, designed with only one purpose ... to study particle production in relativistic heavy-ion collisions.

THERMINATOR 2 *a.k.a.* TH2

THERMal heavy-**IoN** gener**ATOR** version **2**

- TH2 is a Monte-Carlo event generator, designed with only one purpose ... to study particle production in relativistic heavy-ion collisions.
- TH2 consists of several thermal models of particle production, with single freeze-out.

THERMINATOR 2 *a.k.a.* TH2

THERMal heavy-IoN generATOR version 2

- TH2 is a Monte-Carlo event generator, designed with only one purpose ... to study particle production in relativistic heavy-ion collisions.
- TH2 consists of several thermal models of particle production, with single freeze-out.
- Basic tasks of TH2:
    - generation of stable particles and unstable resonances at the chosen hypersurface. The local phase-space density of particles is given by the statistical distribution factors,

THERMINATOR 2 *a.k.a.* TH2

THERMal heavy-**I**o**N** gener**ATOR** version **2**

- TH2 is a Monte-Carlo event generator, designed with only one purpose ... to study particle production in relativistic heavy-ion collisions.
- TH2 consists of several thermal models of particle production, with single freeze-out.
- Basic tasks of TH2:
  - generation of stable particles and unstable resonances at the chosen hypersurface. The local phase-space density of particles is given by the statistical distribution factors,
  - subsequent space-time evolution and decays of hadronic resonances in cascades,

THERMINATOR 2 *a.k.a.* TH2

THERMal heavy-**I**o**N** gener**ATOR** version **2**

- TH2 is a Monte-Carlo event generator, designed with only one purpose ... to study particle production in relativistic heavy-ion collisions.
- TH2 consists of several thermal models of particle production, with single freeze-out.
- Basic tasks of TH2:
  - generation of stable particles and unstable resonances at the chosen hypersurface. The local phase-space density of particles is given by the statistical distribution factors,
  - subsequent space-time evolution and decays of hadronic resonances in cascades,
  - calculation of various physical observables such as: p_T- or m_T-spectra, flow coefficients e.g. v_1, v_2, v_4, femtoscopy: two-particle correlation function, HBT radii, and many more.

THERMINATOR 2

What's new in version 2

- support for user-defined hypersurfaces — interface to various hydro codes, introduced hypersurface input format compatible with XML version 1.0,

THERMINATOR 2

What's new in version 2

- support for user-defined hypersurfaces — interface to various hydro codes, introduced hypersurface input format compatible with XML version 1.0,
- added two new hydro-based freezeout models: `Lhyquid2DBI` (boost-invariant) and `Lhyquid3D` (full 3+1), that use the output form our perfect-fluid hydrodynamics codes,

THERMINATOR 2

What's new in version 2

- support for user-defined hypersurfaces — interface to various hydro codes, introduced hypersurface input format compatible with XML version 1.0,
- added two new hydro-based freezeout models: `Lhyquid2DBI` (boost-invariant) and `Lhyquid3D` (full 3+1), that use the output form our perfect-fluid hydrodynamics codes,
- easy modification, or creation of new freeze-out models with:
  - space-time dependent thermal parameters i.e. temperature, chemical potentials or fugacities
  - shear and bulk viscosity corrections

THERMINATOR 2

What's new in version 2

- support for user-defined hypersurfaces — interface to various hydro codes, introduced hypersurface input format compatible with XML version 1.0,
- added two new hydro-based freezeout models: `Lhyquid2DBI` (boost-invariant) and `Lhyquid3D` (full 3+1), that use the output form our perfect-fluid hydrodynamics codes,
- easy modification, or creation of new freeze-out models with:
  - space-time dependent thermal parameters i.e. temperature, chemical potentials or fugacities
  - shear and bulk viscosity corrections
- library of "standard" sets of hypersurfaces that describe the Au+Au data at RHIC

THERMINATOR 2

What's new in version 2

- support for user-defined hypersurfaces — interface to various hydro codes, introduced hypersurface input format compatible with XML version 1.0,
- added two new hydro-based freezeout models: `Lhyquid2DBI` (boost-invariant) and `Lhyquid3D` (full 3+1), that use the output form our perfect-fluid hydrodynamics codes,
- easy modification, or creation of new freeze-out models with:
  - space-time dependent thermal parameters i.e. temperature, chemical potentials or fugacities
  - shear and bulk viscosity corrections
- library of "standard" sets of hypersurfaces that describe the Au+Au data at RHIC
- separate code to carry out the analysis of the HBT correlation

THERMINATOR 2

What's new in version 2

- support for user-defined hypersurfaces — interface to various hydro codes, introduced hypersurface input format compatible with XML version 1.0,
- added two new hydro-based freezeout models: `Lhyquid2DBI` (boost-invariant) and `Lhyquid3D` (full 3+1), that use the output form our perfect-fluid hydrodynamics codes,
- easy modification, or creation of new freeze-out models with:
  - space-time dependent thermal parameters i.e. temperature, chemical potentials or fugacities
  - shear and bulk viscosity corrections
- library of "standard" sets of hypersurfaces that describe the Au+Au data at RHIC
- separate code to carry out the analysis of the HBT correlation
- shell scripts, cluster calculation — tested with `TORQUE`

Physics behind THERMINATOR 2

statistical hadronization

The key element of the success behind the statistical approach is the inclusion of the full list of hadronic resonances.

- TH2 includes 381 particles form the PDG
- resonance decay via two or three-body decay channels
- decay channel database file has over 1800 entries

Physics behind THERMINATOR 2

Cooper-Frye formalism

The number of hadrons being produced on the freeze-out hypersurface Σ^μ

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

where

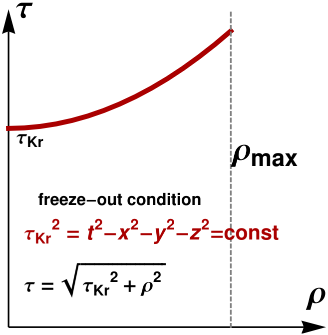
$$f(p \cdot x) = \left\{ exp \left[\frac{p_\mu u_\mu - (B\mu_B + I_3 \mu_{I_3} + S\mu_s)}{T} \right] \pm 1 \right\}^{-1}$$

is the phase-space distribution of particles, and $d\Sigma_\mu$ is the 3-dimensional element of the freeze-out hypersurface.

Physics behind THERMINATOR 2

Hydro inspired models

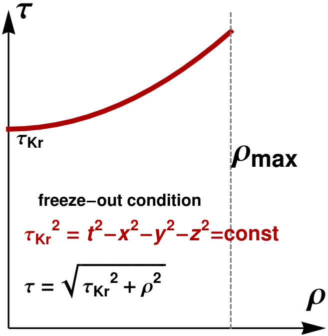
- Krakow Single Freeeze-Out Model



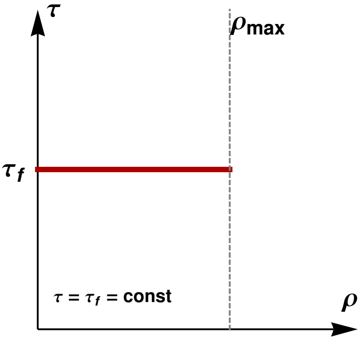
Physics behind THERMINATOR 2

Hydro inspired models

- Krakow Single Freeeze-Out Model



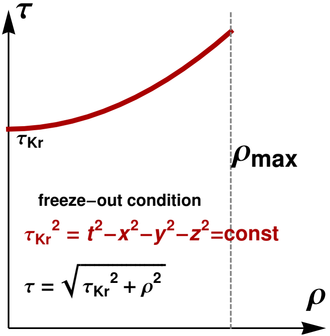
- Blast-Wave



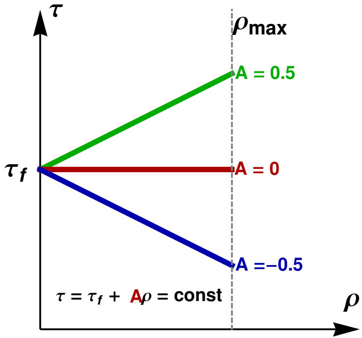
Physics behind THERMINATOR 2

Hydro inspired models

- Krakow Single Freeeze-Out Model



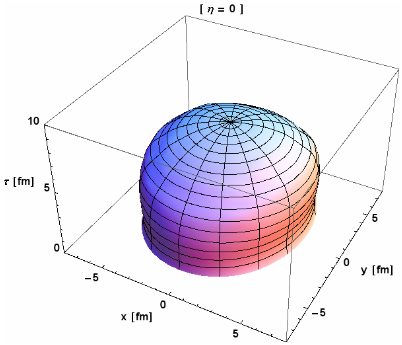
- Blast-Wave and A-class Blast-Wave [BWA]



Physics behind THERMINATOR 2

Hydro based models

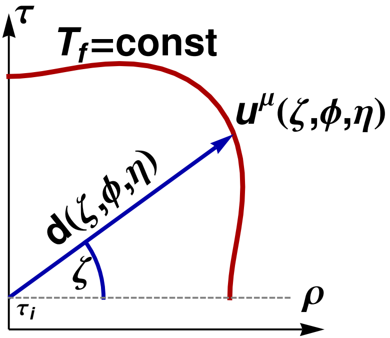
- Lhyquid2DBI



Physics behind THERMINATOR 2

Hydro based models

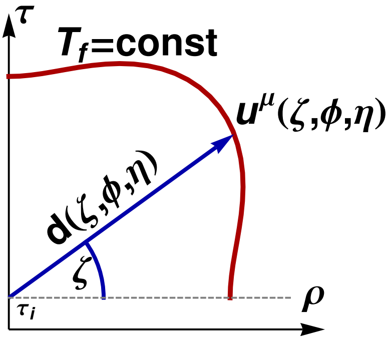
- Lhyquid2DBI



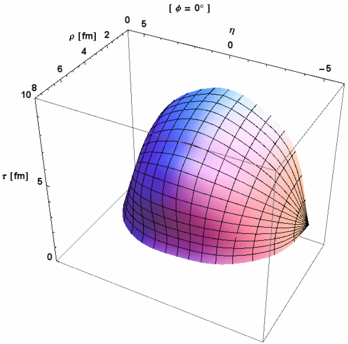
Physics behind THERMINATOR 2

Hydro based models

- Lhyquid2DBI



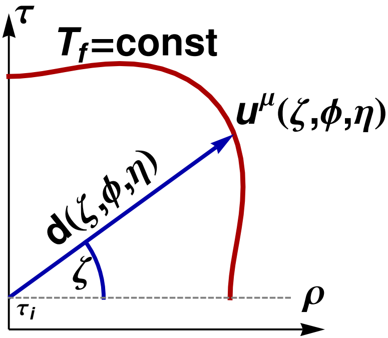
- Lhyquid3D



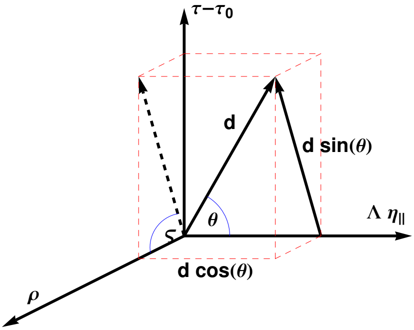
Physics behind THERMINATOR 2

Hydro based models

- Lhyquid2DBI



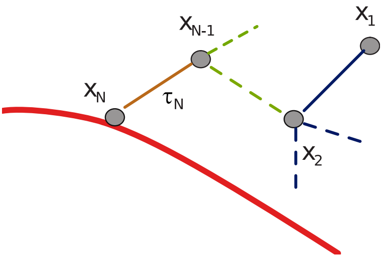
- Lhyquid3D



Physics behind THERMINATOR 2

decay of resonances in cascades

- incorporated all of the four-**** and three-*** resonances,
- excluded all single-* resonances, and practically all double-** resonances listed in the Particle Data Tables



- resonances decay after some time τ with probability $\Gamma \exp(-\Gamma\tau)$, where $1/\Gamma$ is the lifetime,
- the decay is calculated in the rest frame of the resonance,
- two-body and three-body decays are implemented according to the branching ratios taken from the PDG tables,
- products of the decay are boosted again to the hypersurface-element frame,

Physics behind THERMINATOR 2

information stored in ROOT event files

Each event-entry in the `event`*XXX*`.root` file has the following information for each particle

- space-time coordinates (t, x, y, z),
- mass, energy and momentum (E_p, p_x, p_y, p_z)
- PDG code of the particle, its parent and the primordial particle
- particle index in cascade — full decay history

Physics behind THERMINATOR 2

information stored in ROOT event files

Each event-entry in the `event`*XXX*`.root` file has the following information for each particle

- space-time coordinates (t, x, y, z),
- mass, energy and momentum (E_p, p_x, p_y, p_z)
- PDG code of the particle, its parent and the primordial particle
- particle index in cascade — full decay history

ROOT scripts produce figures based on the information in the `event`*XXX*`.root` files

- p_T- or m_T-spectra,
- rapidity or pseudorapidity distribution,
- flow coefficients e.g. v_1, v_2 or even v_4,
- average values of p_T, v_2, multiplicity,

Physics behind THERMINATOR 2

information stored in ROOT event files

Each event-entry in the `event`*XXX*`.root` file has the following information for each particle

- space-time coordinates (t, x, y, z),
- mass, energy and momentum (E_p, p_x, p_y, p_z)
- PDG code of the particle, its parent and the primordial particle
- particle index in cascade — full decay history

ROOT scripts produce figures based on the information in the `event`*XXX*`.root` files

- p_T- or m_T-spectra,
- rapidity or pseudorapidity distribution,
- flow coefficients e.g. v_1, v_2 or even v_4,
- average values of p_T, v_2, multiplicity,

The HBT code calculates the nominator and the denominator of the two-particle correlation function and stores in separete files.

Fitting program (*not included in the official distribution*) extracts the HBT radia.

How to use TH2

system requirements and compilation

TH2 requires:

- C++ compiler (*gcc*, etc.)
- ROOT libraries and include files

How to use TH2

system requirements and compilation

TH2 requires:

- C++ compiler (*gcc*, etc.)
- ROOT libraries and include files

How to get TH2 running (**BETA version**)

- dowload & unpack source package therminator2-beta-1.tar.gz form `http://www.ifj.edu.pl/dept/no4/nz41/therminator2/`
- compile with `make`,
- default installation `make install`.

How to use TH2

manipulate settings

The out-of-the-box run:

- by default TH2 is set to `Lhyquid2DBI` freeze-out model with the hypersurface describing RHIC top energy data for 20-30% in centrality. Generates 500 events.
- event file (`event000.root`), file holding the average multiplicity of primordial particles (`fmultiplicity_XXXXXXXX.dat`) are stored in the subdirectory `./events/lhyquid2dbi-rhicAuAu200c2030Ti455ti025Tf145/`
- figures and correlation function will be stored also in that directory.

How to use TH2

manipulate settings

The out-of-the-box run:

- by default TH2 is set to `Lhyquid2DBI` freeze-out model with the hypersurface describing RHIC top energy data for 20-30% in centrality. Generates 500 events.
- event file (`event000.root`), file holding the average multiplicity of primordial particles (`fmultiplicity_XXXXXXXX.dat`) are stored in the subdirectory `./events/lhyquid2dbi-rhicAuAu200c2030Ti455ti025Tf145/`
- figures and correlation function will be stored also in that directory.

Having fun with the basic settings of TH2:

- main program (`therminator.ini`): more events, different freeze-out model, +more technical settings,
- freeze-out model (`./fomodel/lhyquid2dbi.ini` or any other i.e. `cracow.ini`, `blastwave.ini`): model parameters, choice of hypersurface, etc.

Possible extentions of TH2

modification of existing models

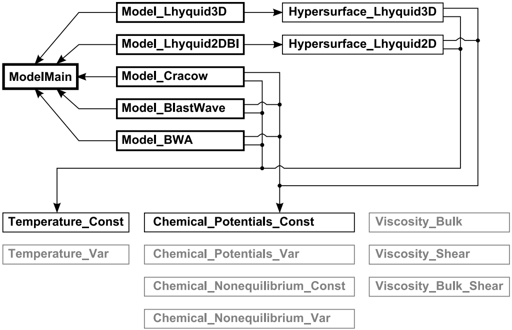
C++ class derivation diagram

space-time dependent thermodynamic variables

- temperature T
- chemical potentials μ_B, μ_{I_3}, μ_S
- fugacity Γ with $\lambda_{q,s,c}$, $\gamma_{q,s,c}$

viscosity correction for

- bulk,
- shear,
- both bulk and shear



FAQ: How to add my own freeze-out model

and how can we help you in doing so

- User has output from his hydro code [hypersurface]:
  - save hydro-output in our XML format [in case you like our models],

FAQ: How to add my own freeze-out model

and how can we help you in doing so

- User has output from his hydro code [hypersurface]:
    - save hydro-output in our XML format [in case you like our models],
    - provide such output with description, so we can prepare an interface for you,
    - tell us how to parametrize your hypersurface data.
    - we will build a Model_<*your-model-name*> class in C++ and add it to the code,

FAQ: How to add my own freeze-out model

and how can we help you in doing so

- Let us assume that the user has basic knowledge of C++
  - create a c++ class Model_*<your-model-name>*.cxx that has a function `double GetIntegrand(ParticleType* aPartType);` defined. This function returns a value of the Cooper-Frye integrand and generates x^μ and p^μ of a particle of a given type.
  - user-model may be derived form already defined classes like `Hypersurface2D`, `Hypersurface3D`, `Temperature`, `Chemical` or `Viscosity`
  - define the volume of the `mHyperCube` — for Monte-Carlo procedure (variable defined in the `ModelMain` class),
  - not obligatory: CRC-32 `mHash` value for the model parameters, character string `mDescription` with the model current parameters
  - make your model choosable by adding in a choice in `Integrator` class and main program,
  - compile debug and enjoy!

FAQ: How to add my own freeze-out model

and how can we help you in doing so

- e-mail us if you have problems or just want to consult:

Mikolaj.Chojnacki@ifj.edu.pl

Hypersurface file format

XML version 1.0

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE HYPERSURFACE [

<HYPERSURFACE filename="rhicAuAu200c2030Ti455ti025Tf145" version="1.0">
<PARAMETERS type="model">
  <PARAM name="Tau_i" unit="fm">0.25</PARAM>
  <PARAM name="Temperature" unit="MeV">145.</PARAM>
  <PARAM name="Mu_B" unit="MeV">28.5</PARAM>
  <PARAM name="Mu_I3" unit="MeV">-0.9</PARAM>
  <PARAM name="Mu_S" unit="MeV">6.9</PARAM>
</PARAMETERS>
<PARAMETERS type="experimental">
<PARAMETERS type="hydro code EoS">
<PARAMETERS type="hydro code initial conditions">
<PARAMETERS type="hydro code equation">
<!-- Included VECTOR3D names : Distance, FluidVt, FluidPhi -->
<VECTOR3D name="Distance" unit="GeV^-1">
  <AXIS name="Zeta">
    <PAR name="min" unit="rad">0.</PAR>
    <PAR name="max" unit="rad">1.5708</PAR>
    <PAR name="pts">91</PAR>
  </AXIS>
  <AXIS name="Phi">
  <AXIS name="">
  <DATA points="8281">
34.869887481893976      34.90422611298513      35.00716959428196
  </DATA>
</VECTOR3D>
```