Thermal-FIST package

Volodymyr Vovchenko (University of Houston)

2023 MUSES Collaboration Meeting, UIUC

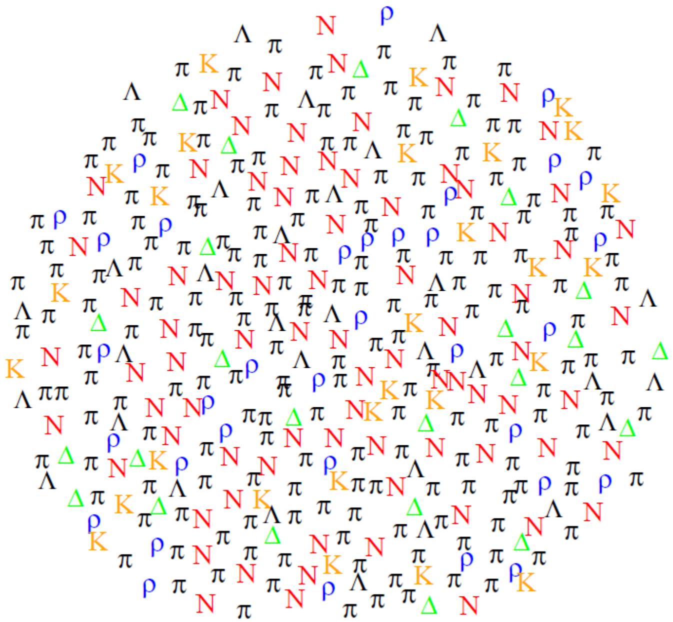
May 17, 2023

Hadron resonance gas (HRG)

HRG: Equation of state of hadronic matter as a multi-component (non-)interacting gas of known hadrons, resonances, and light nuclei

$$\ln Z \approx \sum_{i\in M,B} \ln Z_i^{id} = \sum_{i\in M,B} \frac{d_i V}{2\pi^2} \int_0^\infty \pm p^2 dp \ln\left[1 \pm \exp\left(\frac{\mu_i - E_i}{T}\right)\right]$$

Grand-canonical ensemble: $\mu_i = b_i\mu_B + q_i\mu_Q + s_i\mu_S$ ***chemical equilibrium***



© J. Cleymans

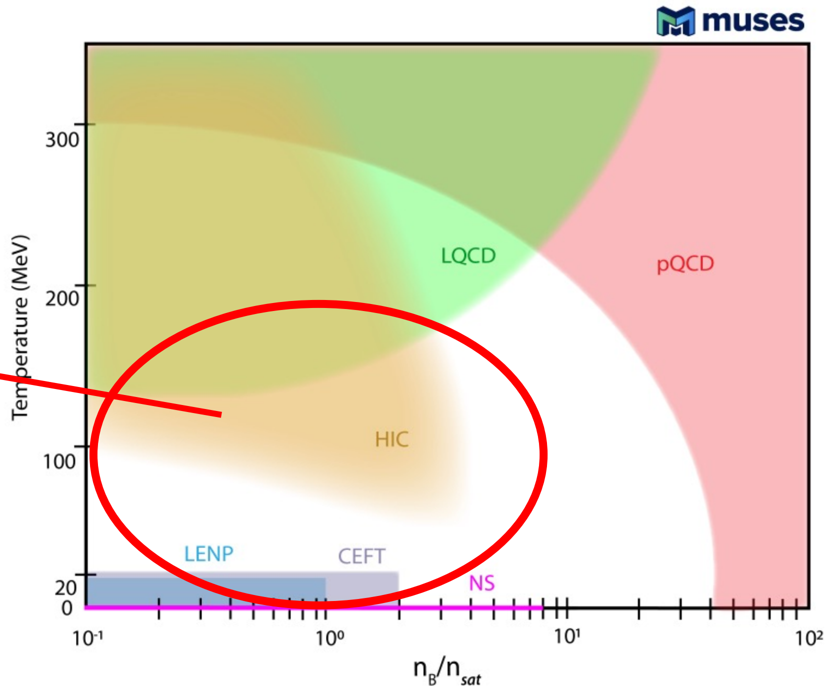
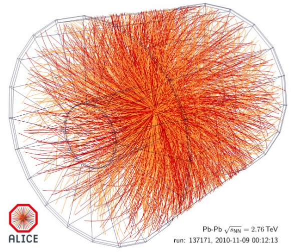
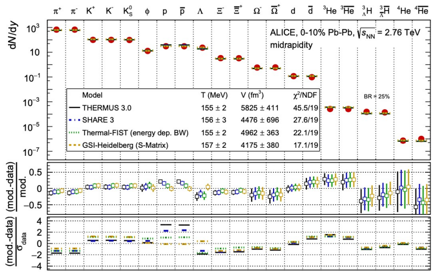


Figure from 2303.17021

HRG model applications

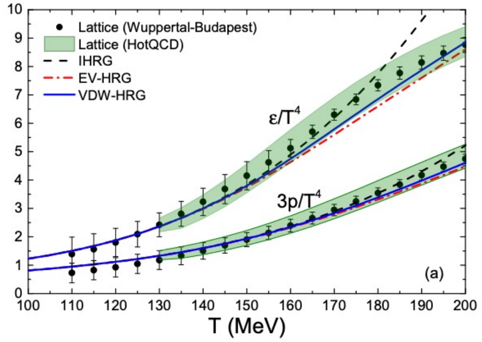
- Heavy-ion collisions
  - Hadrochemistry (chemical freeze-out)
  - Fluctuations of conserved charges



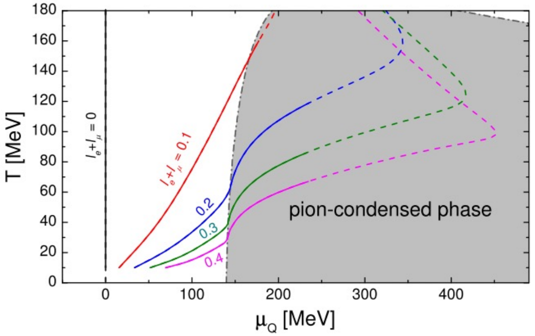


ALICE Coll. (2022)

- Lattice QCD context
  - Understanding the degrees of freedom
  - Equation of state, susceptibilities, partial pressures



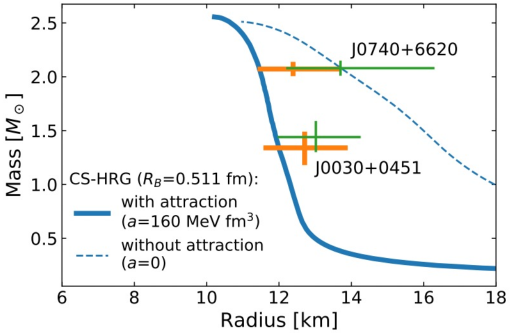
VV et al., PRL (2017)



VV et al., PRL (2021)

- Early universe
  - Modeling QCD contribution to cosmic EoS
  - Finite isospin density

- Neutron-star matter
  - Extending to include non-resonant interactions
  - Hadronic part of the CMF model

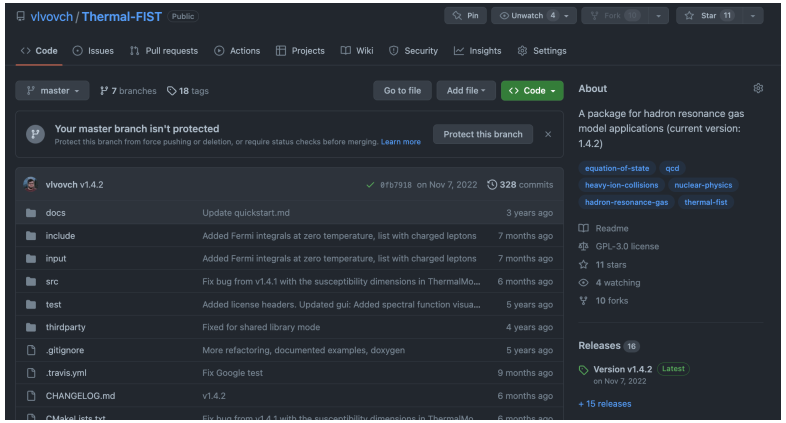


Fujimoto et al., 2109.06799

Natural block for MUSES

What is Thermal-FIST?

Thermal-FIST* (current version: v1.4.2) [**VV**, H. Stoecker]



Open-source C++ package for general-purpose HRG model analysis

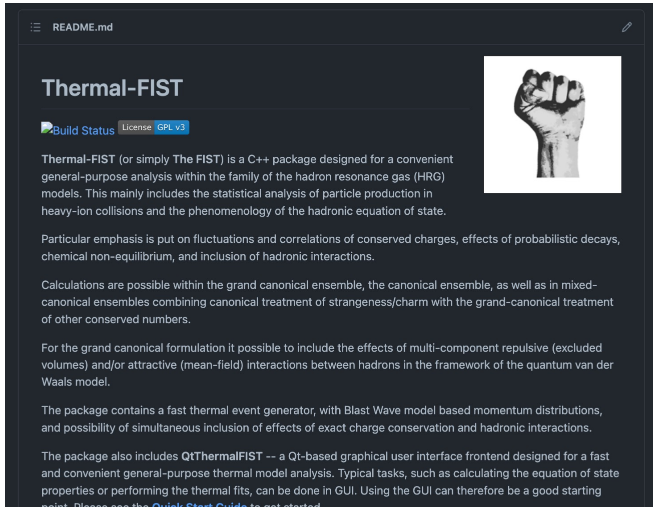
Cross-platform (Linux, Mac, Windows) through **cmake**

License: GPL-3.0

GitHub: https://github.com/vlvovch/Thermal-FIST

physics manual: Comput. Phys. Commun. 244, 295 (2019)

- **2014-2018:** Initial development and applications (closed source)
- **June 2018:** First public release (v0.6)
- **Jan 2019:** Code documentation and CPC article release (v1.0)
- **2019-2022:** Incremental upgrades
- **Soon(?):** Version 1.5 with new features (dense matter EoS, cosmology)



*Thermal, Fast and Interactive Statistical Toolkit

HRG model aspects in Thermal-FIST

- Equation of state and related properties
  - thermodynamics, hadron yields and fluctuations
- Extensions of the base HRG model
  - finite resonance widths
  - repulsive (excluded volume) and van der Waals interactions (criticality)
  - (non-)conserved charges fluctuations and correlations
  - partial chemical equilibrium
- Heavy-ion applications
  - thermal fits
  - small systems and canonical effects
  - Monte Carlo event generator
  - partial chemical equilibrium
  - light nuclei
- Other applications
  - Neutron star matter
  - Early universe (cosmic EoS)

Thermal-FIST structure

- Core library (libThermalFIST)
  - Ideal (base) HRG model (HRGBase)
  - Interacting HRG model (HRGEV/HRGVDW)
  - Partial chemical equilibrium (HRGPCE)
  - Monte Carlo mode (HRGEventGenerator)
  - Thermal fits (HRGThermalFit)
- Graphical user interface (QtThermalFIST)
  - Based on Qt5
  - Wrapper around libThermalFIST
- Sample console applications
  - Essentially just C++ macros linking to libThermalFIST

External dependencies:

- Eigen library for linear algebra (header-only, built-in)
- Minuit2 (built-in, i.e. ROOT **not needed**)
- Qt5 (for GUI only)

**Thermal, Fast and Interactive Statistical Toolkit*

Using Thermal-FIST

Installation using **git** and **cmake**

```
# Clone the repository from GitHub
git clone https://github.com/vlvovch/Thermal-FIST.git
cd Thermal-FIST

# Create a build directory, configure the project with cmake
# and build with make
mkdir build
cd build
cmake ../
make

# Run the GUI frontend
./bin/QtThermalFIST

# Run the test calculations from the paper
./bin/examples/cpc1HRGTDep
./bin/examples/cpc2chi2
./bin/examples/cpc3chi2NEQ
./bin/examples/cpc4mcHRG
```

Quick start guide *Documentation* *Physics manual*

Using Thermal-FIST: Console mode

```cpp
#include "HRGBase.h"
#include "HRGEV.h"
#include "HRGFit.h"
#include "HRGVDW.h"

#include "ThermalFISTConfig.h"

using namespace std;

#ifdef ThermalFIST_USENAMESPACE
using namespace thermalfist;
#endif


// Temperature dependence of HRG thermodynamics at \mu = 0
// Three variants of the HRG model:
// 1. Ideal HRG: <config> = 0
// 2. EV-HRG with constant radius parameter r = 0.3 fm for all hadrons (as in 1412.5478): <config> = 1
// 3. QvdW-HRG with a and b for baryons only, fixed to nuclear ground state (as in 1609.03975): <config> = 2
// Usage: cpc1HRGTDep <config>
int main(int argc, char *argv[])
{
  // Particle list file
  // Here we will use the list from THERMUS-2.3, for comparing the results with THERMUS-2.3
  string listname = string(ThermalFIST_INPUT_FOLDER) + "/list/thermus23/list.dat";

  // Alternative: use the default PDG2014 list
  //string listname = string(ThermalFIST_INPUT_FOLDER) + "/list/PDG2014/list.dat";

  // Create the hadron list instance and read the list from file
  ThermalParticleSystem TPS(listname);

  // Which variant of the HRG model to use
  int config = 0;
```

```cpp
if (config == 0) // Ideal HRG
{
  model = new ThermalModelIdeal(&TPS);

  printf("#Calculating thermodynamics at \\mu = 0 in Id-HRG model\n");

  modeltype = "Id-HRG";
}
else if (config == 1) // EV-HRG, r = 0.3 fm, to reproduce 1412.5478
{
  model = new ThermalModelEVDiagonal(&TPS);
  // Set r = 0.3 fm for each hadron in the list
  double rad = 0.3;
  for (int i = 0; i < model->TPS()->ComponentsNumber(); ++i)
    model->SetRadius(i, rad);

  printf("#Calculating thermodynamics at \\mu = 0 in EV-HRG model with r = %lf fm\n", rad);

  modeltype = "EV-HRG";
}
else if (config == 2) // QvdW-HRG, to reproduce 1609.03975
{
  model = new ThermalModelVDWFull(&TPS);

  // vdW parameters, for baryon-baryon, antibaryon-antibaryon ONLY, otherwise zero
  double a = 0.329; // In GeV*fm3
  double b = 3.42;  // In fm3
```

Link to **libThermalFIST** and write a C++ macro doing whatever calculation you want

The most flexible way of using the code

git submodule is useful

MUSES use case: write a wrapper for **libThermalFIST**?

Using Thermal-FIST: Jupyter notebooks

jupyter Xeus Cling

Interactive notebooks through Jupyter (xeus kernel and ROOT-cling)*

jupyter FitExample Last Checkpoint: несколько секунд назад (unsaved changes)   Logout

File Edit View Insert Cell Kernel Widgets Help   Trusted   C++14

Run   Code

Initialize and run the fitter

In [7]:
```
// Set chemical potentials to zero
model.SetBaryonChemicalPotential(0.0);
model.SetElectricChemicalPotential(0.0);
model.SetStrangenessChemicalPotential(0.0);
model.SetCharmChemicalPotential(0.0);
model.FillChemicalPotentials();

// Initialize the fitter
ThermalModelFit fitter(&model);

// Do not fit muB, it is zero at LHC
fitter.SetParameterFitFlag("muB", false);

// Pass the data to the fitter
fitter.SetQuantities(dataPbPb010);

// Perform the fit
ThermalModelFitParameters fitResult = fitter.PerformFit(false);
```

Print the fitted parameters and the χ^2

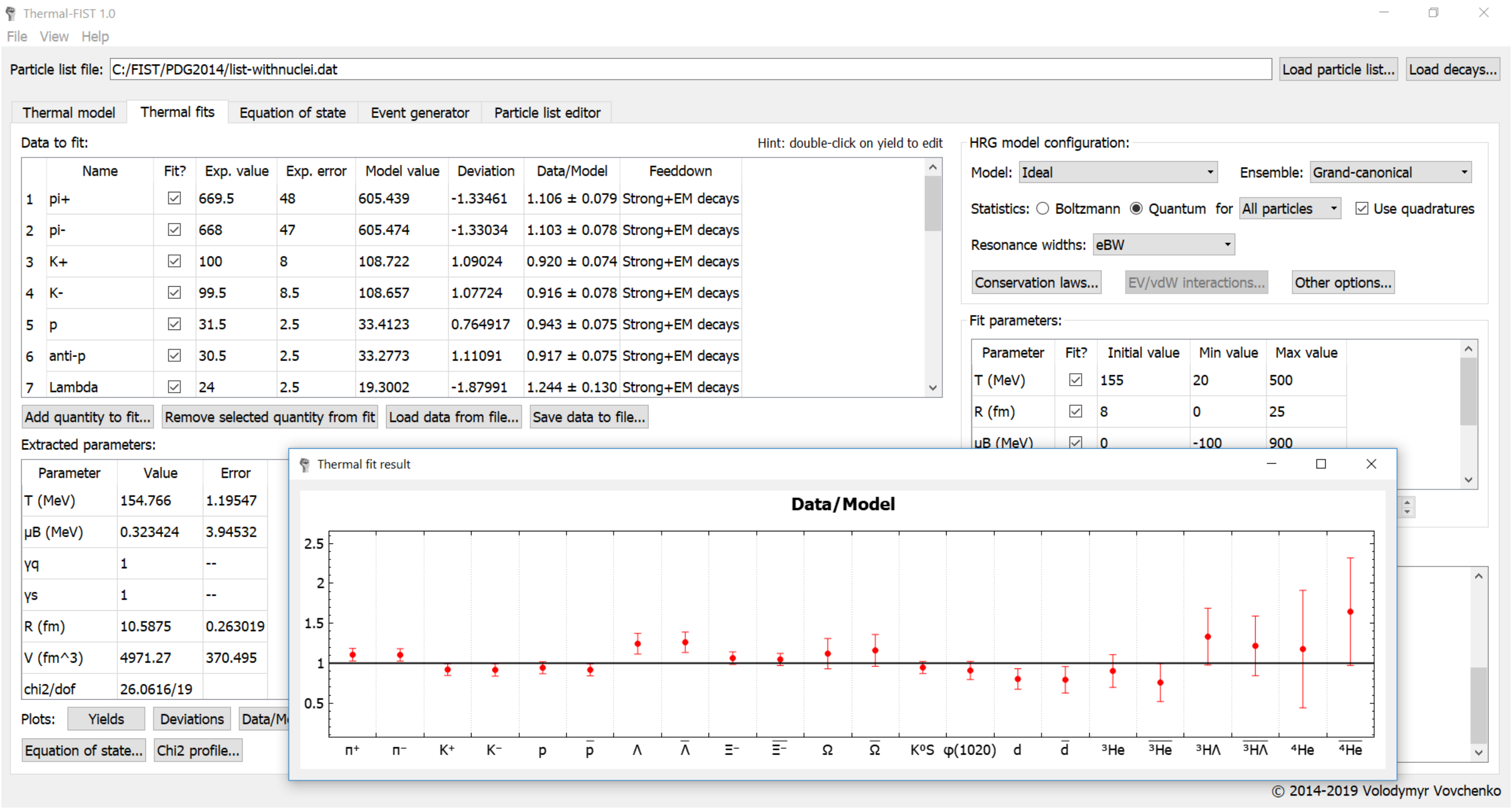
In [8]:
```
cout << "Extracted parameters:" << endl;
cout << setw(15) << "T [MeV]" << " = " << setw(15) << 1.e3 * fitResult.T.value << " +- " << 1.e3 * fitResult.T.error <<
cout << setw(15) << "R [fm]" << " = " << setw(15) << fitResult.R.value << " +- " << fitResult.R.error << endl;
cout << setw(15) << "chi2/dof" << " = " << setw(15) << fitResult.chi2 << "/" << fitResult.ndf << endl;
```

```
Extracted parameters:
        T [MeV] =          155.28 +- 2.78665
         R [fm] =         10.3342 +- 0.545205
       chi2/dof =         15.3832/6
```

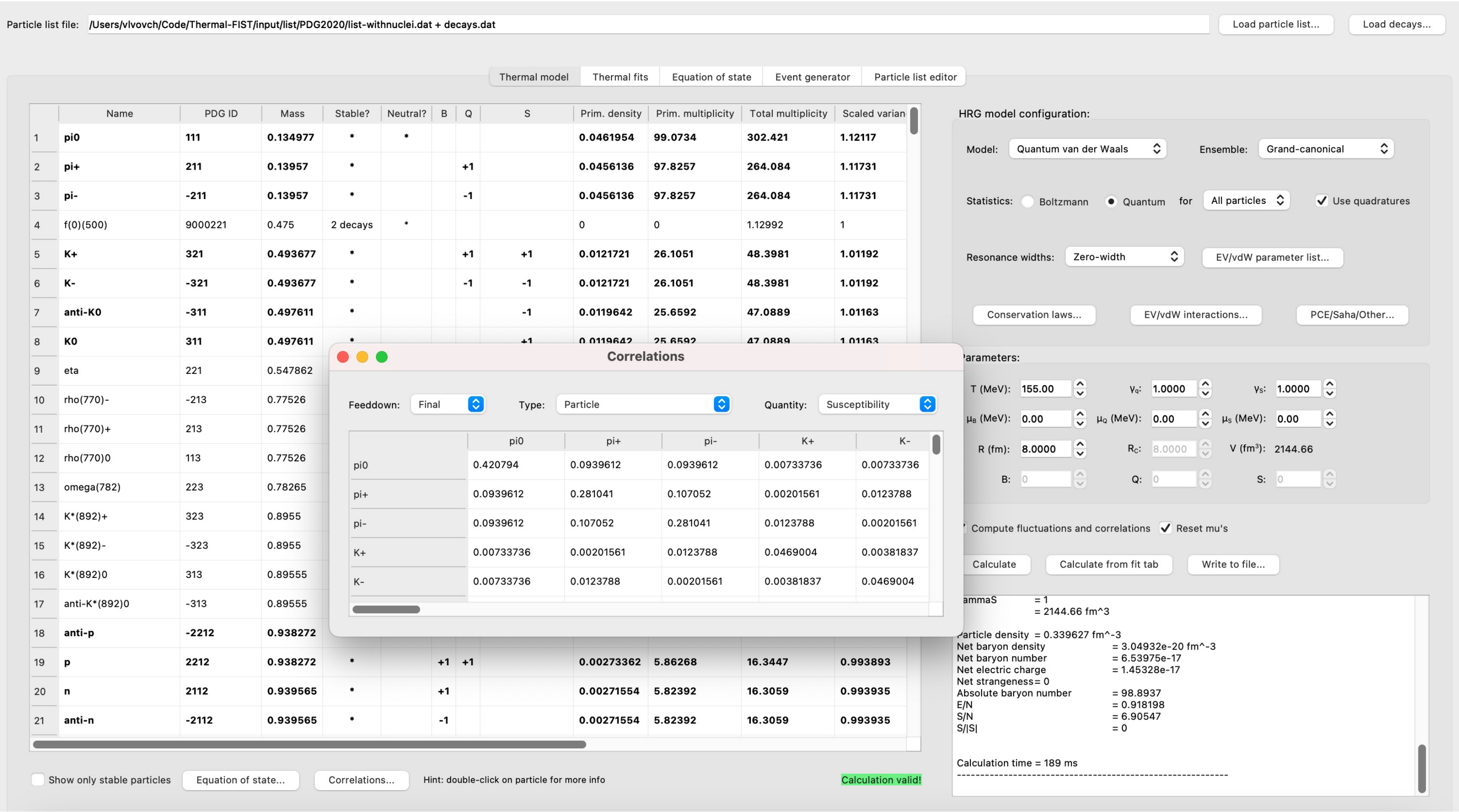
*Since version 1.2.1, example at github.com/vlvovch/FIST-jupyter

Using Thermal-FIST: GUI

Graphical user interface for *general-purpose* HRG model applications

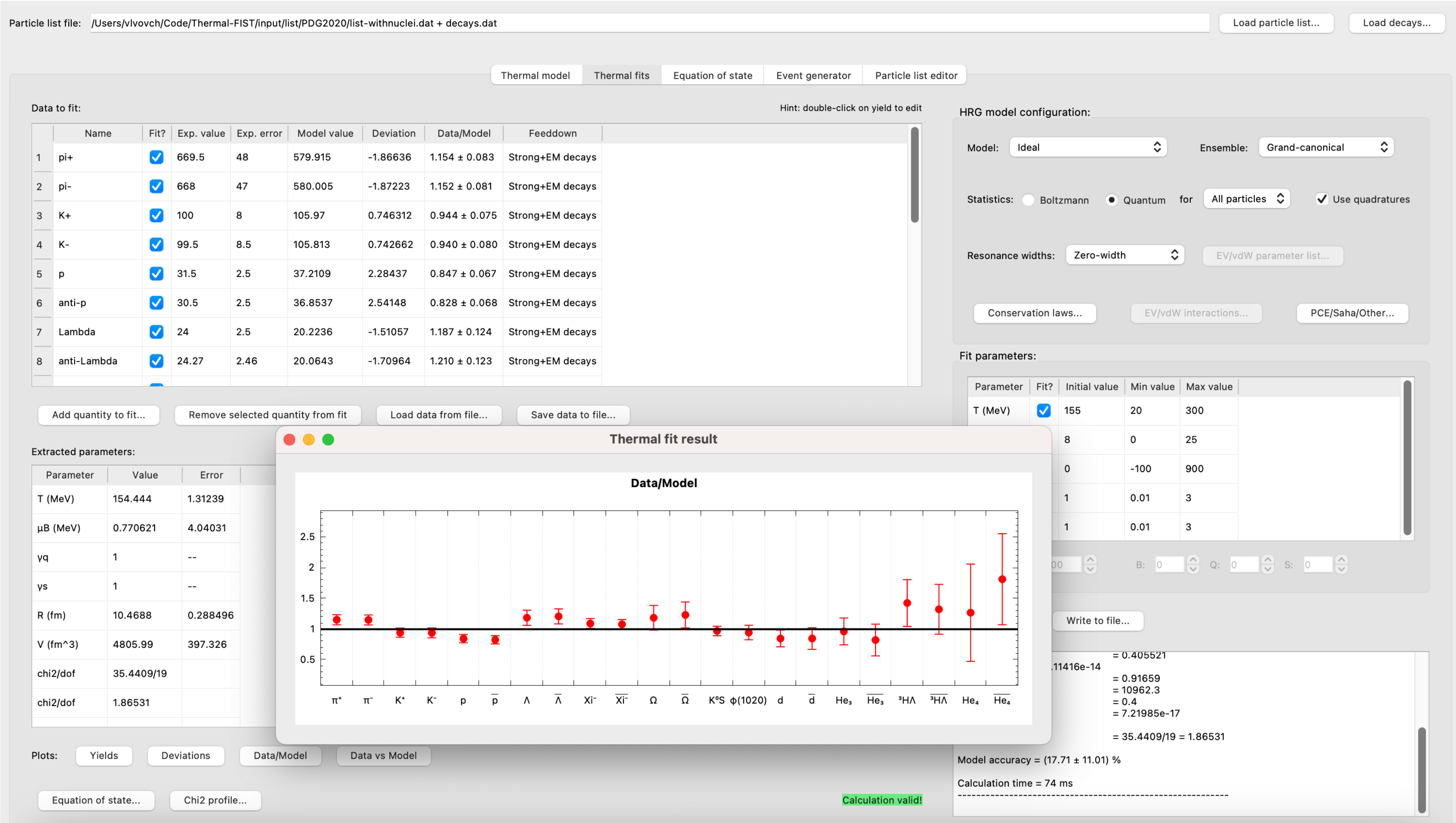


Thermal-FIST and HRG model equation of state



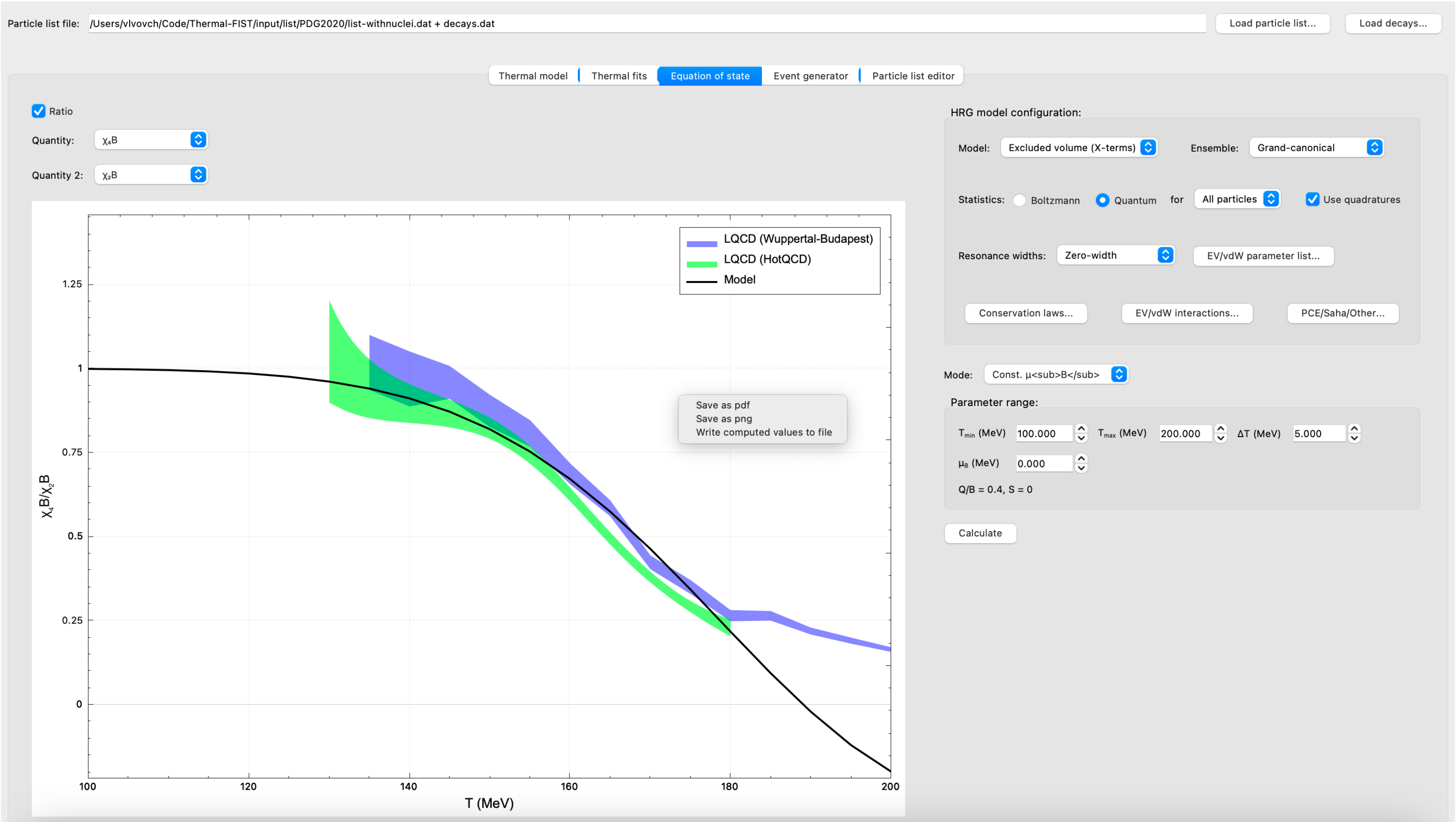
- Base calculation is at fixed T & $\mu_{B,Q,S}$ (alternatively at fixed s/n_B, Q/B, S/|S|)
- Thermodynamic functions, hadron abundances, feeddown, correlations and fluctuations

Thermal-FIST and Thermal-FITS



- Extract chemical freeze-out parameters from heavy-ion hadron abundances
- χ^2 minimization

Thermal-FIST and equation of state

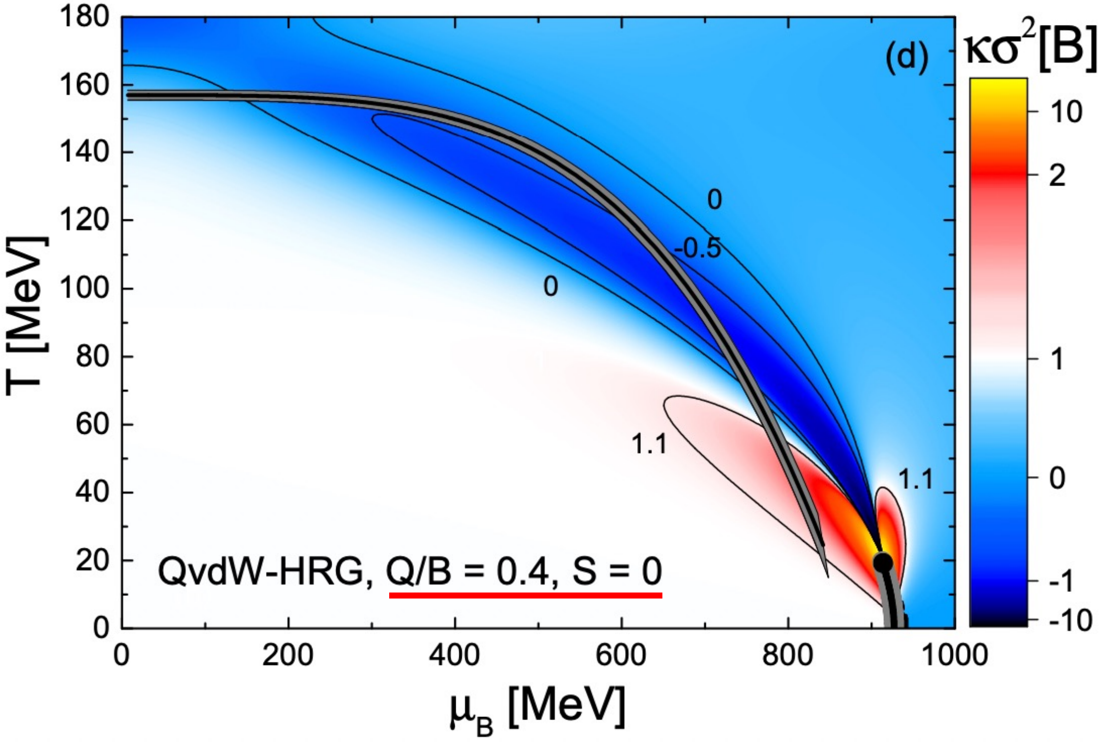


- Compute HRG model quantities along a fixed T, μ_B, or μ_B / T
- Impose conservation laws [e.g. strangeness neutrality (heavy-ions) or charge neutrality (neutron stars)]

Thermal-FIST and equation of state

Console mode provides more flexibility

baryon number



electric charge

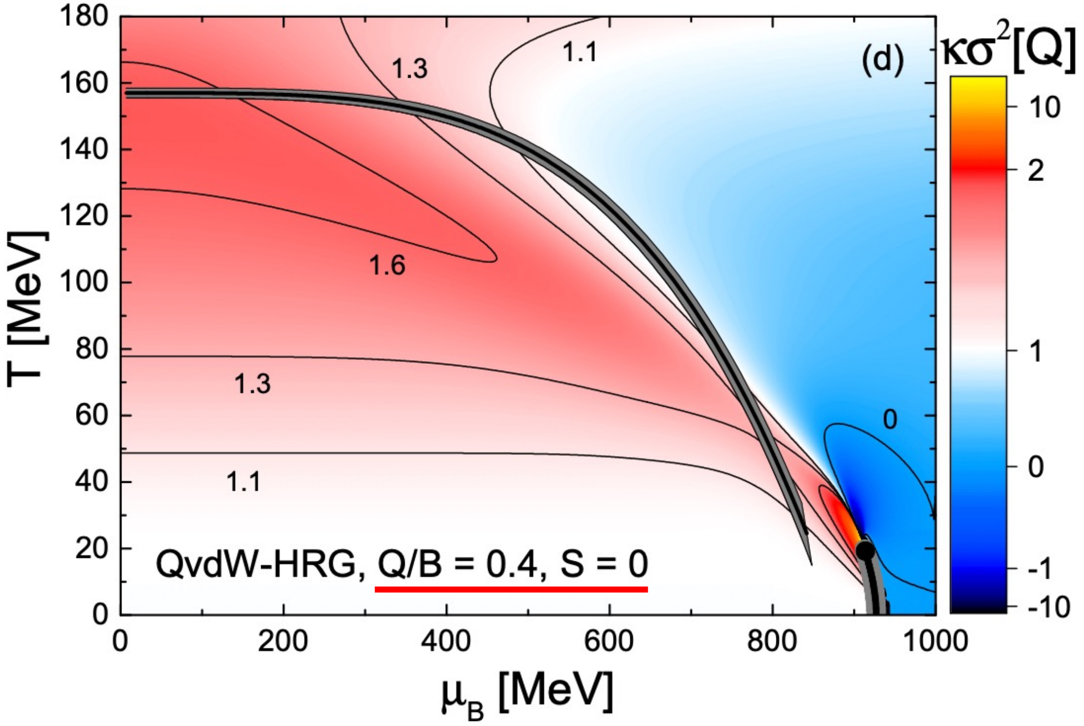
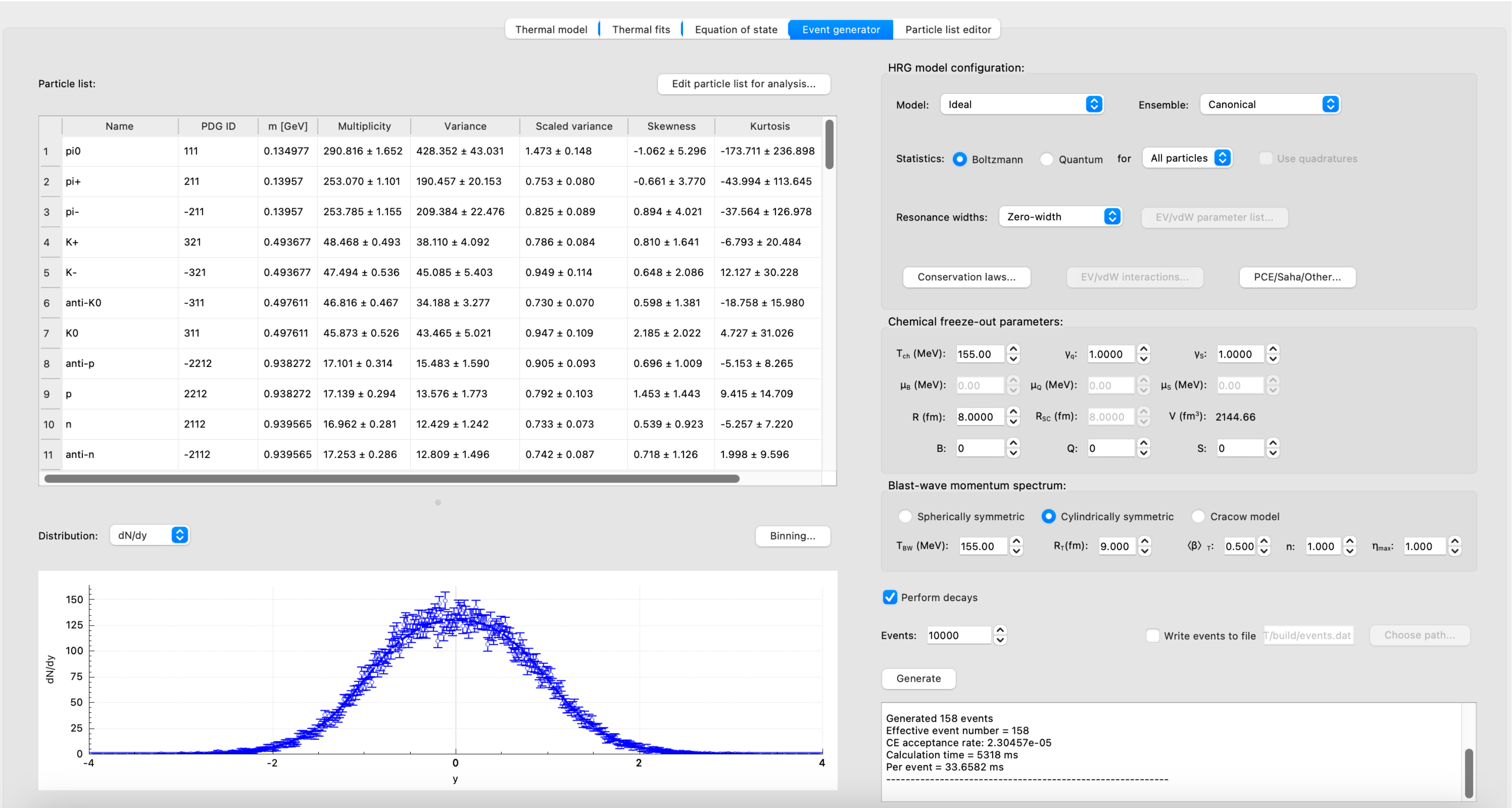


Figure from R.V. Poberezhnyuk et al., PRC 99, 024907 (2019)

Thermal-FIST and HRG event generator



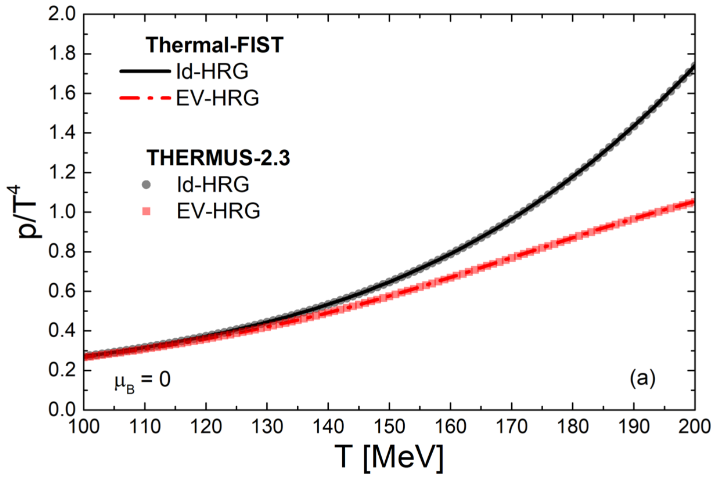
- Monte Carlo sampling of hadron abundances, momenta, and coordinates
- Superimposed on blast-wave flow velocity profile
- Realistic modeling of acceptance effects, especially for correlations and fluctuations

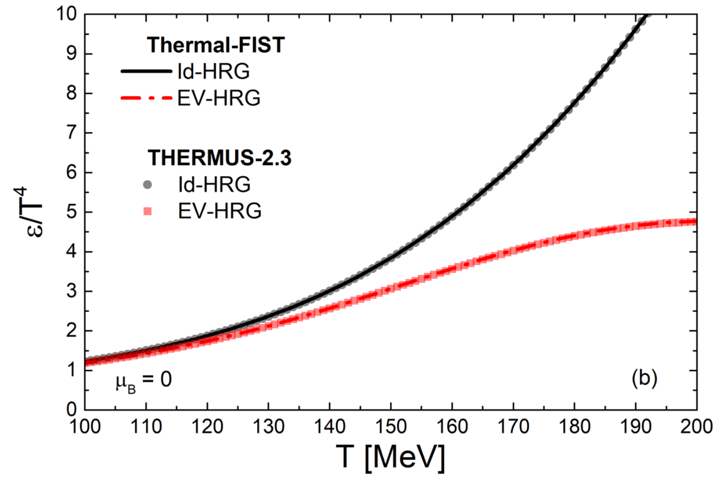
Thermal-FIST in THERMUS mode: cross-check

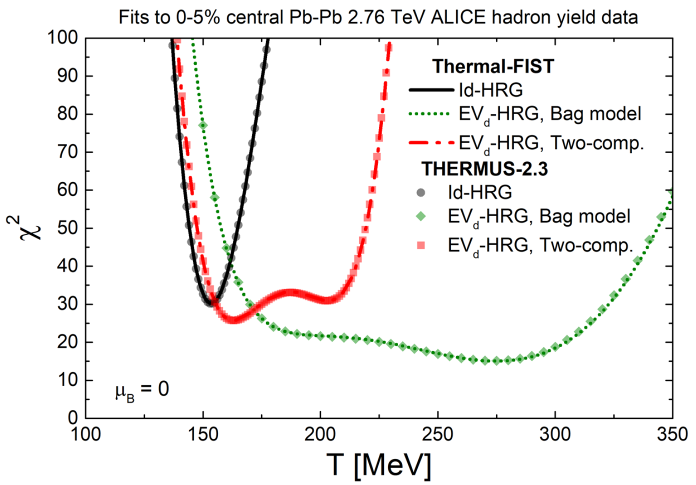
THERMUS* is an early open-source implementation of some HRG model features

[S. Wheaton, J. Cleymans, B. Hippolyte, et al.]

Use exactly the same input (particle list, finite widths, and excluded volume parameters) and compare







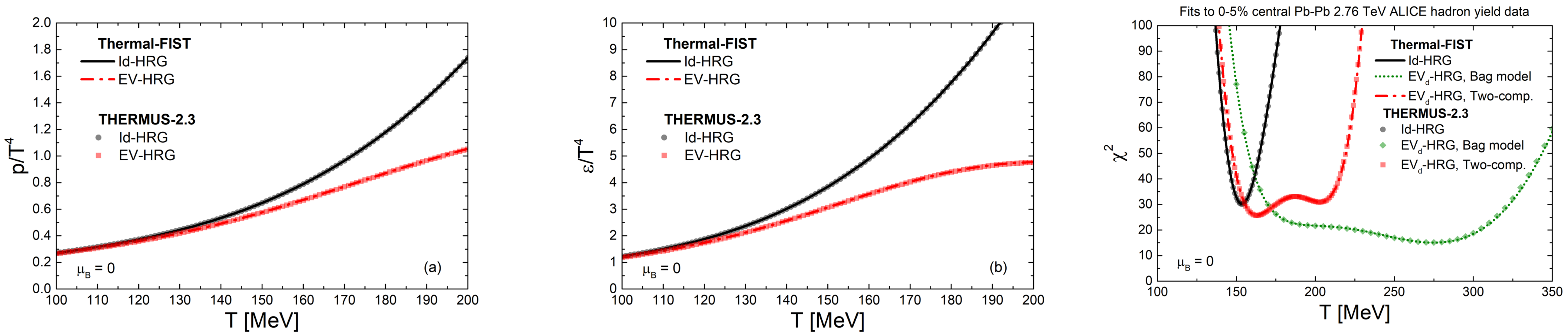
*https://github.com/thermus-project/THERMUS

Thermal-FIST in THERMUS mode: cross-check

THERMUS* is an early open-source implementation of some HRG model features

[S. Wheaton, J. Cleymans, B. Hippolyte, et al.]

Use exactly the same input (particle list, finite widths, and excluded volume parameters) and compare



FIST: Fist **IS T**hermus

Rigorous unit testing still to be implemented

*https://github.com/thermus-project/THERMUS

Interactions in HRG

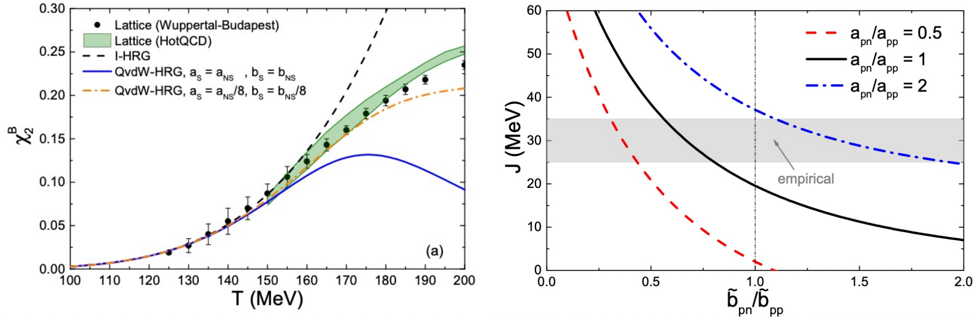
Interacting HRG can be more smoothly connected to other QCD phases than ideal HRG

e.g. the crossover in T direction, Albright, Kapusta, Young, PRC (2014)

Thermal-FIST incorporates van der Waals interactions in the most general form

$$p(T, n_1, \ldots, n_h) = \sum_i \frac{T\, n_i}{1 - \sum_j \tilde{b}_{ji}\, n_j} - \sum_{i,j} a_{ij}\, n_i\, n_j$$

- Separate excluded volume b_{ij} for each pair i,j of species
- Separate mean field a_{ij} for each pair i,j of species
- So far very little explored!
- E.g. indications for flavor-dependent parameters from
  - Lattice QCD susceptibilities [Karthein et al., 2107.00588]
  - Symmetry energy
  - Neutron-star matter EoS [Fujimoto et al., 2109.06799]



VV, Motornenko, Alba, et al., 1707.09215

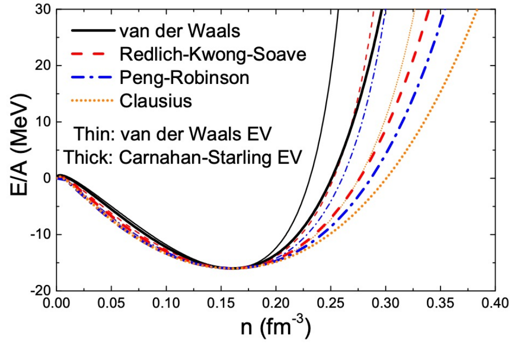
Interactions in HRG: Beyond van der Waals

Standard van der Waals gives too stiff EoS beyond the saturation density

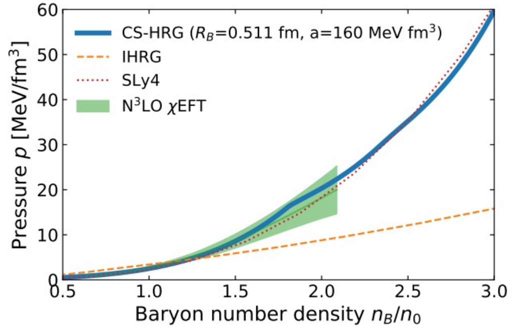
Beyond vdW:

- Generalized (non-linear) excluded volume
  - Carnahan-Starling (CS)
- Density-dependent mean-field
  - Real gases
  - Skyrme
  - VDF model

A. Sorensen, V. Koch, PRC 104, 034904 (2021)

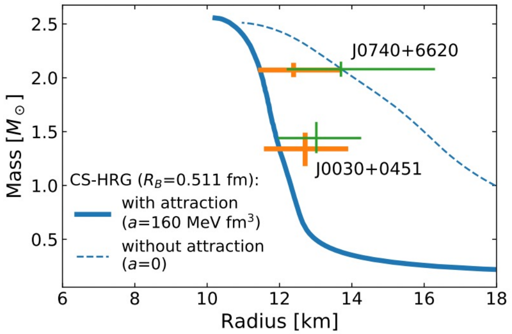


VV, PRC 96, 015206 (2017)



Helps soften the EoS in the cold & dense regime, making it easier to match with others

Available out-of-the-box in FIST in next version (already present in **devel** branch on github), adding leptons into the list one can do neutron-star matter



Fujimoto et al., 2109.06799

Another extension: pion interactions and condensation at finite isospin density

VV et al., PRL (2021)

Thermal-FIST & MUSES: Summary

- Thermal-FIST is an open-source implementation of the HRG model equation of state with many knobs
  - Particle lists, interaction parameters, and other settings easily customizable
  - Provides EoS properties (averages as well as susceptibilities) at given T & $\mu_{B,Q,S}$
  - Works both under heavy-ion and neutron star regimes

- Standalone C++ implementation with minimal external dependencies
  - Only the base library **libThermalFIST** really needed to be built and linked against
  - Integration into MUSES with a wrapper?

- Interaction parameters still need to be constrained

Thanks for your attention!