



# Thermal-FIST package

Volodymyr Vovchenko (University of Houston)

*2023 MUSES Collaboration Meeting, UIUC*



**May 17, 2023**

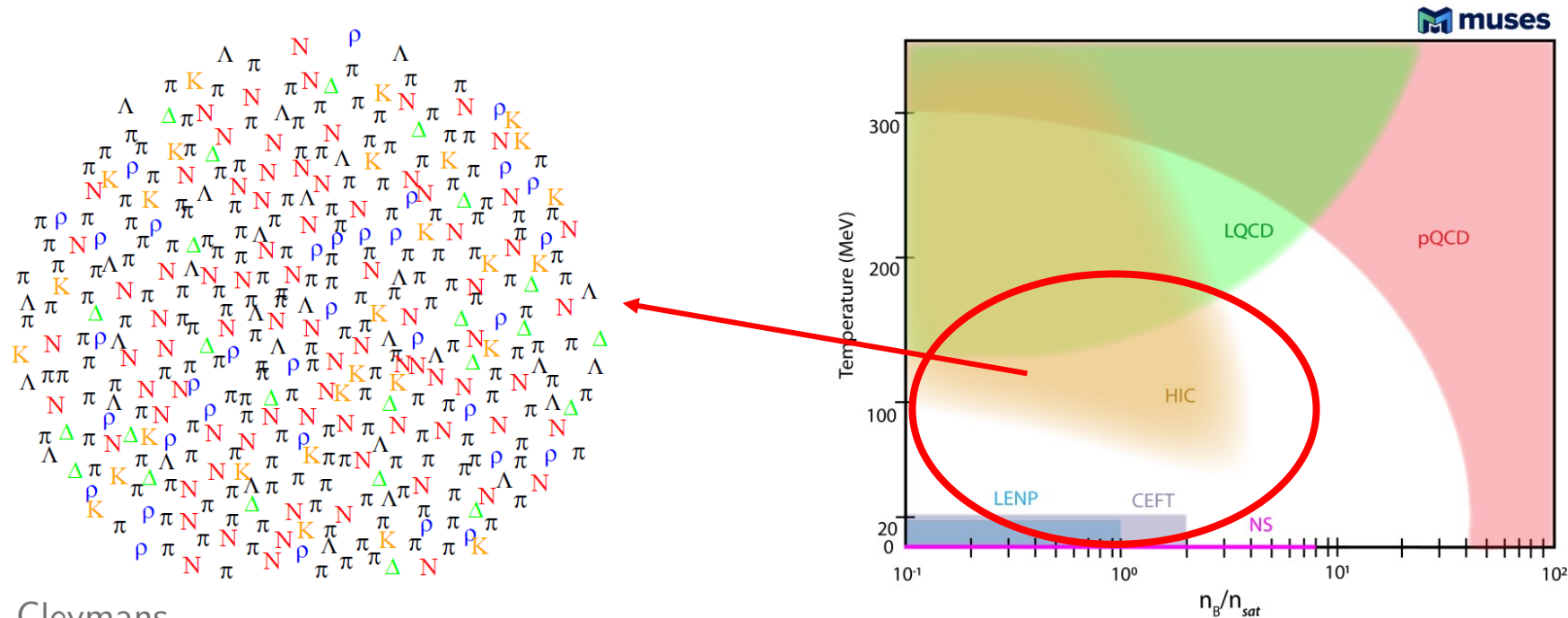


# Hadron resonance gas (HRG)

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

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

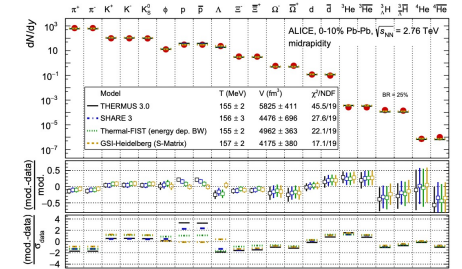
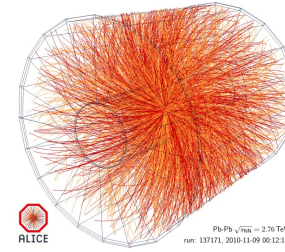
**Grand-canonical ensemble:**  $\mu_i = b_i \mu_B + q_i \mu_Q + s_i \mu_S$  *chemical equilibrium*



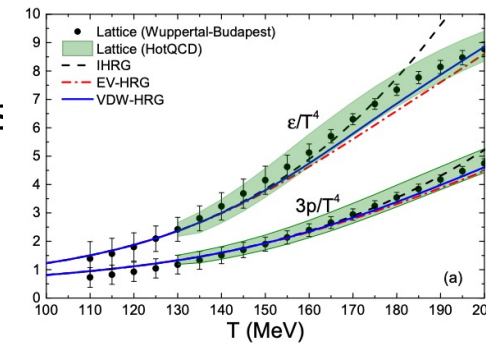
# HRG model applications

- Heavy-ion collisions
  - Hadrochemistry (chemical freeze-out)
  - Fluctuations of conserved charges
- Lattice QCD context
  - Understanding the degrees of freedom
  - Equation of state, susceptibilities, partial pressures
- 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

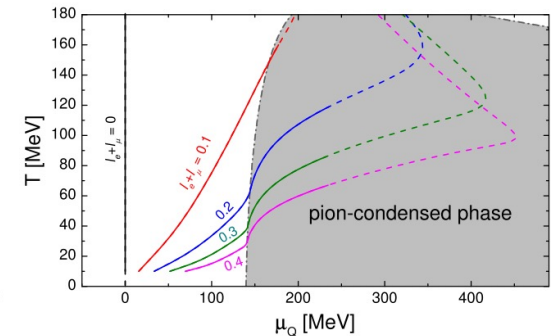
Natural block for MUSES



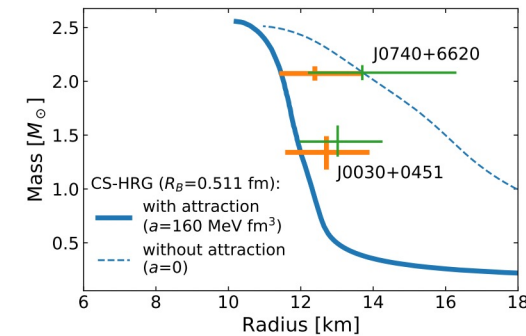
ALICE Coll. (2022)



VV et al., PRL (2017)



VV et al., PRL (2021)



Fujimoto et al., 2109.06799

# What is Thermal-FIST?



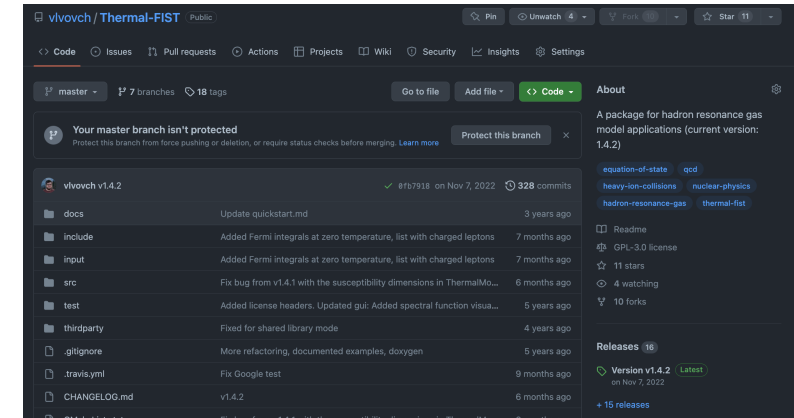
**Thermal-FIST\*** (current version: v1.4.2) [VV, H. Stoecker]

Open-source C++ package for general-purpose HRG model analysis  
*Cross-platform* (Linux, Mac, Windows) through **cmake**

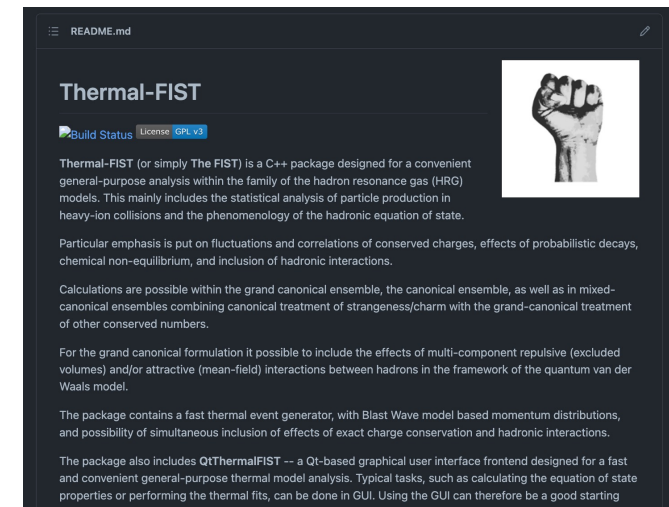
**License:** GPL-3.0

**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)



# 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/HRGV DW)
  - 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)

# 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
```

# Using Thermal-FIST: Console mode

```
#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;
}
```

```
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  $\text{GeV}^3$ 
    double b = 3.42; // In  $\text{fm}^3$ 
}
```

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



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

The screenshot shows a Jupyter notebook interface. At the top, it says 'jupyter FitExample Last Checkpoint: несколько секунд назад (unsaved changes)'. There is a 'Logout' button and a 'Trusted' status indicator. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The toolbar has icons for file operations, a 'Run' button, and a 'Code' dropdown menu. The notebook content is divided into two cells. The first cell is titled 'Initialize and run the fitter' and contains C++ code for setting chemical potentials, initializing the fitter, and performing a fit. The second cell is titled 'Print the fitted parameters and the  $\chi^2$ ' and contains C++ code for printing the results. The output of the second cell shows the extracted parameters: T [MeV] = 155.28 +- 2.78665, R [fm] = 10.3342 +- 0.545205, and chi2/dof = 15.3832/6.

```
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  $\chi^2$ 

In [8]: cout << "Extracted parameters:" << endl;
cout << setw(15) << "T [MeV]" << " = " << setw(15) << 1.e3 * fitResult.T.value << " +- " << 1.e3 * fitResult.T.error << endl;
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](https://github.com/vlvovch/FIST-jupyter)

# Using Thermal-FIST: GUI

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

Thermal-FIST 1.0  
File View Help

Particle list file: C:/FIST/PDG2014/list-withnuclei.dat Load particle list... Load decays...

Thermal model Thermal fits Equation of state Event generator Particle list editor

Data to fit: Hint: double-click on yield to edit

	Name	Fit?	Exp. value	Exp. error	Model value	Deviation	Data/Model	Feeddown
1	pi+	<input checked="" type="checkbox"/>	669.5	48	605.439	-1.33461	1.106 ± 0.079	Strong+EM decays
2	pi-	<input checked="" type="checkbox"/>	668	47	605.474	-1.33034	1.103 ± 0.078	Strong+EM decays
3	K+	<input checked="" type="checkbox"/>	100	8	108.722	1.09024	0.920 ± 0.074	Strong+EM decays
4	K-	<input checked="" type="checkbox"/>	99.5	8.5	108.657	1.07724	0.916 ± 0.078	Strong+EM decays
5	p	<input checked="" type="checkbox"/>	31.5	2.5	33.4123	0.764917	0.943 ± 0.075	Strong+EM decays
6	anti-p	<input checked="" type="checkbox"/>	30.5	2.5	33.2773	1.11091	0.917 ± 0.075	Strong+EM decays
7	Lambda	<input checked="" type="checkbox"/>	24	2.5	19.3002	-1.87991	1.244 ± 0.130	Strong+EM decays

Add quantity to fit... Remove selected quantity from fit Load data from file... Save data to file...

HRG model configuration:

Model: Ideal Ensemble: Grand-canonical

Statistics:  Boltzmann  Quantum for All particles  Use quadratures

Resonance widths: eBW

Conservation laws... EV/vdW interactions... Other options...

Fit parameters:

Parameter	Fit?	Initial value	Min value	Max value
T (MeV)	<input checked="" type="checkbox"/>	155	20	500
R (fm)	<input checked="" type="checkbox"/>	8	0	25
$\mu_B$ (MeV)	<input checked="" type="checkbox"/>	0	-100	900

Extracted parameters:

Parameter	Value	Error
T (MeV)	154.766	1.19547
$\mu_B$ (MeV)	0.323424	3.94532
$\gamma_q$	1	--
$\gamma_s$	1	--
R (fm)	10.5875	0.263019
V (fm <sup>3</sup> )	4971.27	370.495
chi2/dof	26.0616/19	

Plots: Yields Deviations Data/M

Equation of state... Chi2 profile...

Thermal fit result

Data/Model

© 2014-2019 Volodymyr Vovchenko

# Thermal-FIST and HRG model equation of state

Particle list file: /Users/vlvovch/Code/Thermal-FIST/input/list/PDG2020/list-withnuclei.dat + decays.dat

Thermal model Thermal fits Equation of state Event generator Particle list editor

Name	PDG ID	Mass	Stable?	Neutral?	B	Q	S	Prim. density	Prim. multiplicity	Total multiplicity	Scaled varian
1	pi0	111	0.134977	*	*			0.0461954	99.0734	302.421	1.12117
2	pi+	211	0.13957	*		+1		0.0456136	97.8257	264.084	1.11731
3	pi-	-211	0.13957	*		-1		0.0456136	97.8257	264.084	1.11731
4	f(0)(500)	9000221	0.475	2 decays	*			0	0	1.12992	1
5	K+	321	0.493677	*		+1	+1	0.0121721	26.1051	48.3981	1.01192
6	K-	-321	0.493677	*		-1	-1	0.0121721	26.1051	48.3981	1.01192
7	anti-K0	-311	0.497611	*			-1	0.0119642	25.6592	47.0889	1.01163
8	K0	311	0.497611	*			+1	0.0119642	25.6592	47.0889	1.01163
9	eta	221	0.547862	*							
10	rho(770)-	-213	0.77526	*							
11	rho(770)+	213	0.77526	*							
12	rho(770)0	113	0.77526	*							
13	omega(782)	223	0.78265	*							
14	K*(892)+	323	0.8955	*							
15	K*(892)-	-323	0.8955	*							
16	K*(892)0	313	0.89555	*							
17	anti-K*(892)0	-313	0.89555	*							
18	anti-p	-2212	0.938272	*							
19	p	2212	0.938272	*		+1	+1	0.00273362	5.86268	16.3447	0.993893
20	n	2112	0.939565	*		+1		0.00271554	5.82392	16.3059	0.993935
21	anti-n	-2112	0.939565	*		-1		0.00271554	5.82392	16.3059	0.993935

HRG model configuration:

Model: Quantum van der Waals Ensemble: Grand-canonical

Statistics: Boltzmann Quantum for All particles Use quadratures

Resonance widths: Zero-width EV/vdW parameter list...

Conservation laws... EV/vdW interactions... PCE/Saha/Other...

Parameters:

T (MeV): 155.00  $V_B$ : 1.0000  $V_S$ : 1.0000

$\mu_B$  (MeV): 0.00  $\mu_Q$  (MeV): 0.00  $\mu_S$  (MeV): 0.00

R (fm): 8.0000  $R_C$ : 8.0000 V (fm<sup>3</sup>): 2144.66

B: 0 Q: 0 S: 0

Compute fluctuations and correlations  Reset mu's

Calculate Calculate from fit tab Write to file...

$\text{gammaS} = 1$   
 $= 2144.66 \text{ fm}^3$   
 particle density = 0.339627 fm<sup>-3</sup>  
 Net baryon density = 3.04932e-20 fm<sup>-3</sup>  
 Net baryon number = 6.53975e-17  
 Net electric charge = 1.45328e-17  
 Net strangeness = 0  
 Absolute baryon number = 98.8937  
 E/N = 0.918198  
 S/N = 6.90547  
 S/|S| = 0

Calculation time = 189 ms

Correlations

Feeddown: Final Type: Particle Quantity: Susceptibility

	pi0	pi+	pi-	K+	K-
pi0	0.420794	0.0939612	0.0939612	0.00733736	0.00733736
pi+	0.0939612	0.281041	0.107052	0.00201561	0.0123788
pi-	0.0939612	0.107052	0.281041	0.0123788	0.00201561
K+	0.00733736	0.00201561	0.0123788	0.0469004	0.00381837
K-	0.00733736	0.0123788	0.00201561	0.00381837	0.0469004

Show only stable particles Equation of state... Correlations... Hint: double-click on particle for more info Calculation valid!

- 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

Particle list file: /Users/vlvovch/Code/Thermal-FIST/input/list/PDG2020/list-withnuclei.dat + decays.dat Load particle list... Load decays...

Thermal model Thermal fits Equation of state Event generator Particle list editor

Data to fit: Hint: double-click on yield to edit

	Name	Fit?	Exp. value	Exp. error	Model value	Deviation	Data/Model	Feeddown
1	pi+	<input checked="" type="checkbox"/>	669.5	48	579.915	-1.86636	1.154 ± 0.083	Strong+EM decays
2	pi-	<input checked="" type="checkbox"/>	668	47	580.005	-1.87223	1.152 ± 0.081	Strong+EM decays
3	K+	<input checked="" type="checkbox"/>	100	8	105.97	0.746312	0.944 ± 0.075	Strong+EM decays
4	K-	<input checked="" type="checkbox"/>	99.5	8.5	105.813	0.742662	0.940 ± 0.080	Strong+EM decays
5	p	<input checked="" type="checkbox"/>	31.5	2.5	37.2109	2.28437	0.847 ± 0.067	Strong+EM decays
6	anti-p	<input checked="" type="checkbox"/>	30.5	2.5	36.8537	2.54148	0.828 ± 0.068	Strong+EM decays
7	Lambda	<input checked="" type="checkbox"/>	24	2.5	20.2236	-1.51057	1.187 ± 0.124	Strong+EM decays
8	anti-Lambda	<input checked="" type="checkbox"/>	24.27	2.46	20.0643	-1.70964	1.210 ± 0.123	Strong+EM decays

Add quantity to fit...
Remove selected quantity from fit
Load data from file...
Save data to file...

HRG model configuration:

Model:  Ensemble:

Statistics:  Boltzmann  Quantum for   Use quadratures

Resonance widths:  EV/vdW parameter list...

Conservation laws...
EV/vdW interactions...
PCE/Saha/Other...

Fit parameters:

Parameter	Fit?	Initial value	Min value	Max value
T (MeV)	<input checked="" type="checkbox"/>	155	20	300
8	<input type="checkbox"/>	0	0	25
0	<input type="checkbox"/>	-100	-100	900
1	<input type="checkbox"/>	0.01	0.01	3
1	<input type="checkbox"/>	0.01	0.01	3

Write to file...

```

= 0.405521
11416e-14
= 0.91659
= 10962.3
= 0.4
= 7.21985e-17
= 35.4409/19 = 1.86531

```

Model accuracy = (17.71 ± 11.01) %  
Calculation time = 74 ms

Equation of state... Chi2 profile... Calculation valid!

Extracted parameters:

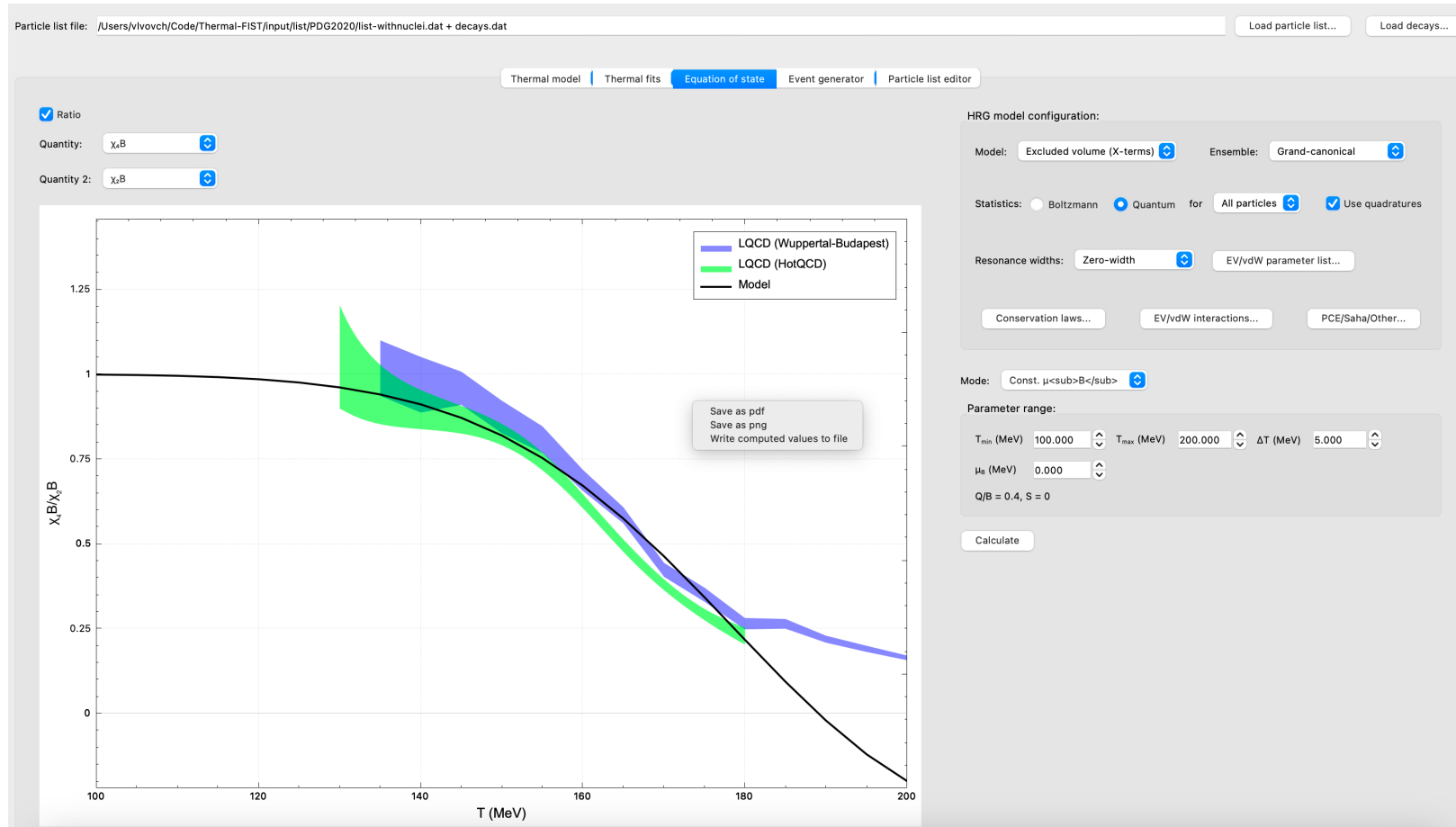
Parameter	Value	Error
T (MeV)	154.444	1.31239
μB (MeV)	0.770621	4.04031
vq	1	--
ys	1	--
R (fm)	10.4688	0.288496
V (fm^3)	4805.99	397.326
chi2/dof	35.4409/19	
chi2/dof	1.86531	

Plots:  Yields  Deviations  Data/Model  Data vs Model

Thermal fit result

- Extract chemical freeze-out parameters from heavy-ion hadron abundances
- $\chi^2$  minimization

# Thermal-FIST and equation of state



- Compute HRG model quantities along a fixed  $T$ ,  $\mu_B$ , or  $\mu_B / T$
- 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

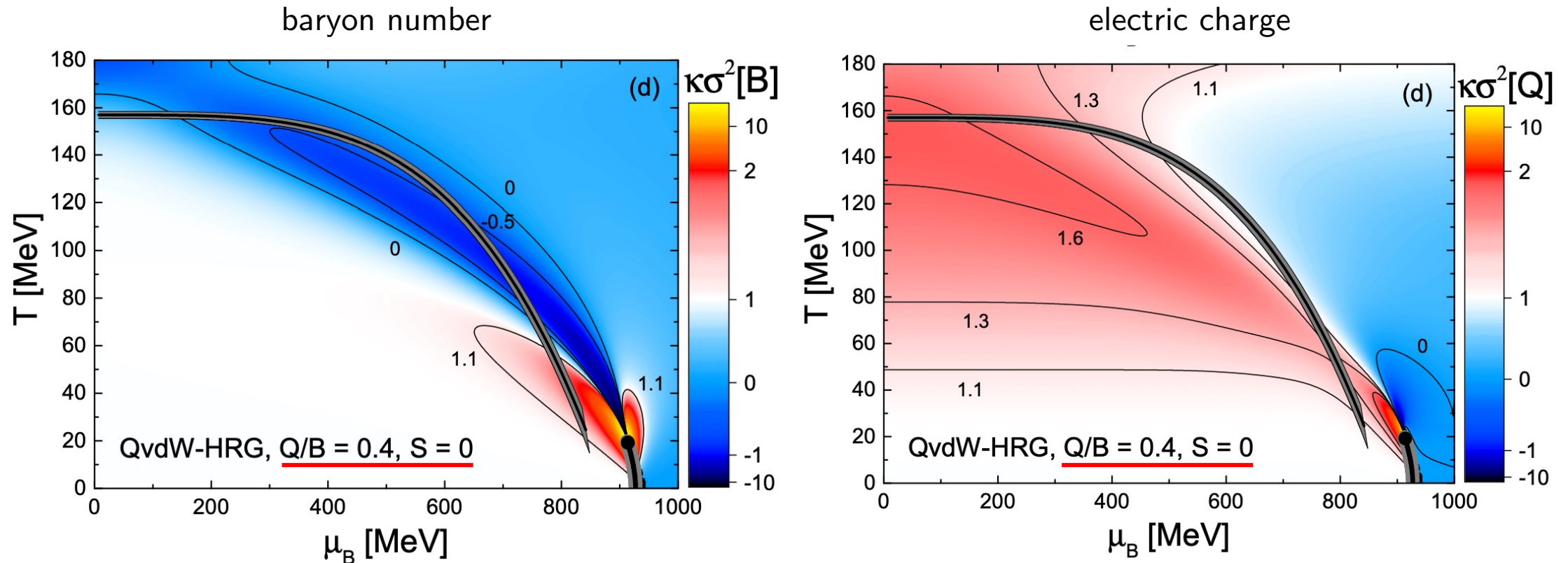


Figure from R.V. Poberezhnyuk et al., PRC 99, 024907 (2019)

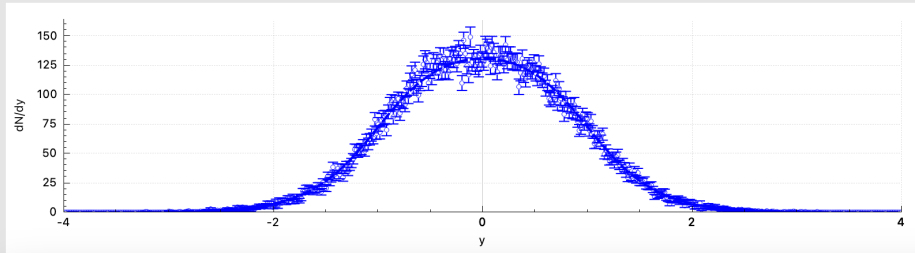
# Thermal-FIST and HRG event generator

Thermal model | Thermal fits | Equation of state | **Event generator** | Particle list editor

Particle list: Edit particle list for analysis...

	Name	PDG ID	m [GeV]	Multiplicity	Variance	Scaled variance	Skewness	Kurtosis
1	pi0	111	0.134977	290.816 ± 1.652	428.352 ± 43.031	1.473 ± 0.148	-1.062 ± 5.296	-173.711 ± 236.898
2	pi+	211	0.13957	253.070 ± 1.101	190.457 ± 20.153	0.753 ± 0.080	-0.661 ± 3.770	-43.994 ± 113.645
3	pi-	-211	0.13957	253.785 ± 1.155	209.384 ± 22.476	0.825 ± 0.089	0.894 ± 4.021	-37.564 ± 126.978
4	K+	321	0.493677	48.468 ± 0.493	38.110 ± 4.092	0.786 ± 0.084	0.810 ± 1.641	-6.793 ± 20.484
5	K-	-321	0.493677	47.494 ± 0.536	45.085 ± 5.403	0.949 ± 0.114	0.648 ± 2.086	12.127 ± 30.228
6	anti-K0	-311	0.497611	46.816 ± 0.467	34.188 ± 3.277	0.730 ± 0.070	0.598 ± 1.381	-18.758 ± 15.980
7	K0	311	0.497611	45.873 ± 0.526	43.465 ± 5.021	0.947 ± 0.109	2.185 ± 2.022	4.727 ± 31.026
8	anti-p	-2212	0.938272	17.101 ± 0.314	15.483 ± 1.590	0.905 ± 0.093	0.696 ± 1.009	-5.153 ± 8.285
9	p	2212	0.938272	17.139 ± 0.294	13.576 ± 1.773	0.792 ± 0.103	1.453 ± 1.443	9.415 ± 14.709
10	n	2112	0.939565	16.962 ± 0.281	12.429 ± 1.242	0.733 ± 0.073	0.539 ± 0.923	-5.257 ± 7.220
11	anti-n	-2112	0.939565	17.253 ± 0.286	12.809 ± 1.496	0.742 ± 0.087	0.718 ± 1.126	1.998 ± 9.596

Distribution: dN/dy Binning...



HRG model configuration:

Model: Ideal Ensemble: Canonical

Statistics:  Boltzmann  Quantum for All particles  Use quadratures

Resonance widths: Zero-width EV/vdW parameter list...

Conservation laws... EV/vdW interactions... PCE/Saha/Other...

Chemical freeze-out parameters:

$T_{ch}$  (MeV): 155.00  $V_c$ : 1.0000  $V_s$ : 1.0000

$\mu_B$  (MeV): 0.00  $\mu_s$  (MeV): 0.00  $\mu_c$  (MeV): 0.00

$R$  (fm): 8.0000  $R_{sc}$  (fm): 8.0000  $V$  (fm<sup>3</sup>): 2144.66

$B$ : 0  $Q$ : 0  $S$ : 0

Blast-wave momentum spectrum:

Spherically symmetric  Cylindrically symmetric  Cracow model

$T_{BW}$  (MeV): 155.00  $R_f$  (fm): 9.000  $\langle\beta\rangle$ : 0.500  $n$ : 1.000  $\eta_{max}$ : 1.000

Perform decays

Events: 10000  Write events to file T/build/events.dat Choose path...

Generate

Generated 158 events  
Effective event number = 158  
CE acceptance rate: 2.30457e-05  
Calculation time = 5318 ms  
Per event = 33.6582 ms

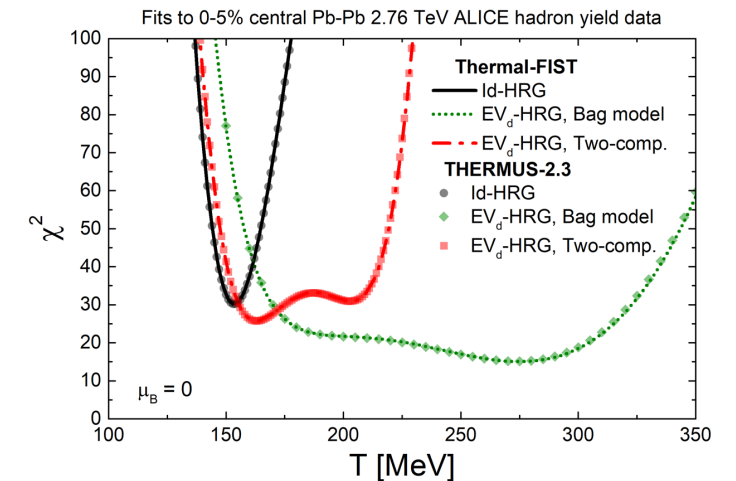
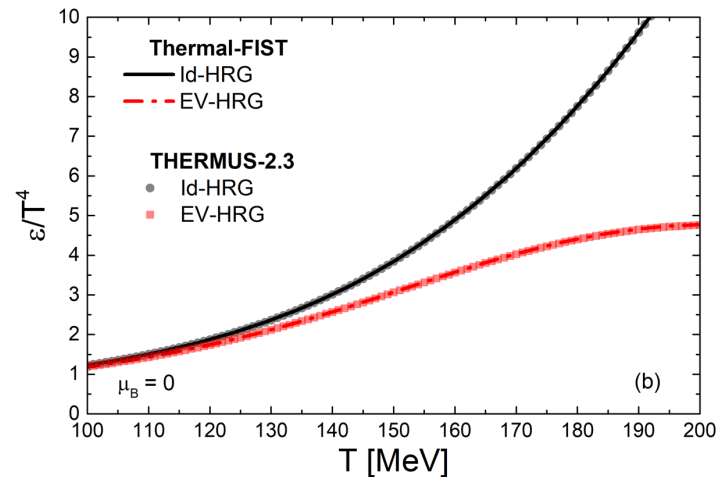
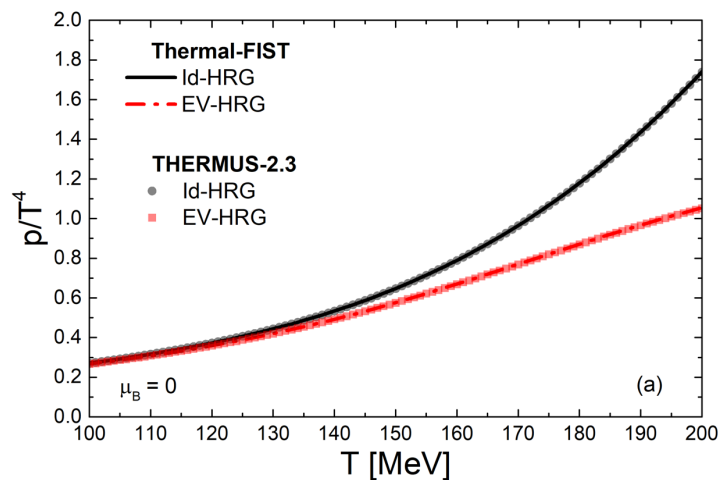
- Monte Carlo sampling of hadron abundances, momenta, and coordinates
- Superimposed on blast-wave flow velocity profile
- Realistic modeling of acceptance effects, especially for correlations and fluctuations

# 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



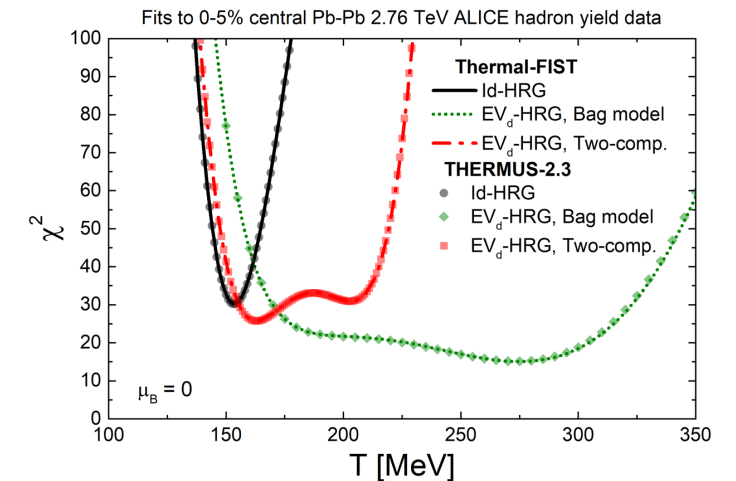
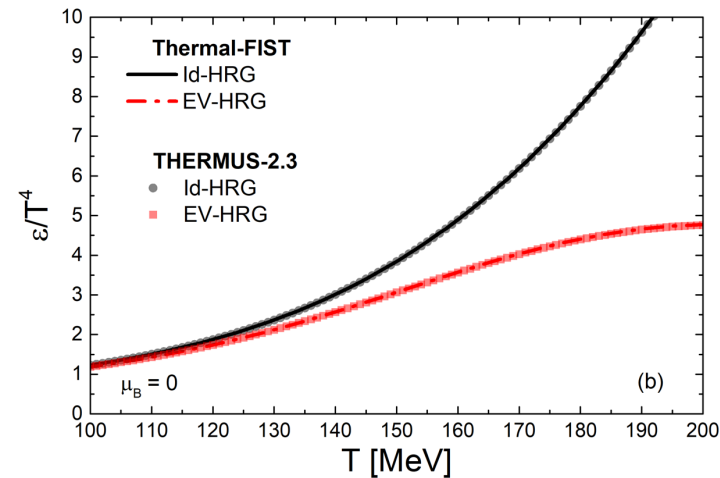
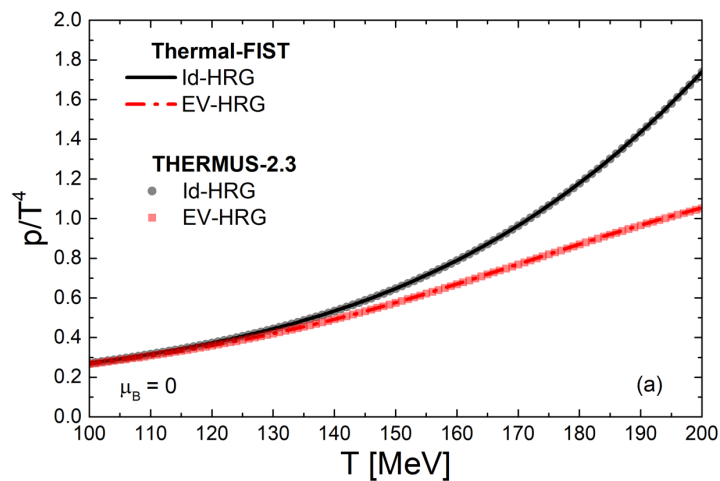


# Thermal-FIST in THERMUS mode: cross-check

**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 Thermus**

Rigorous unit testing still to be implemented

# Interactions in HRG

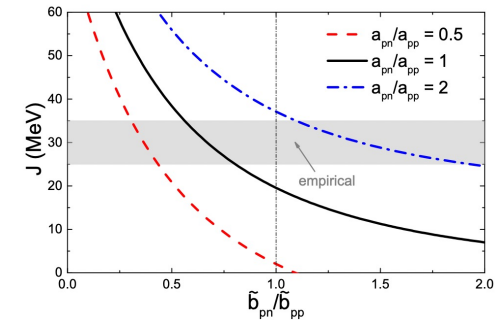
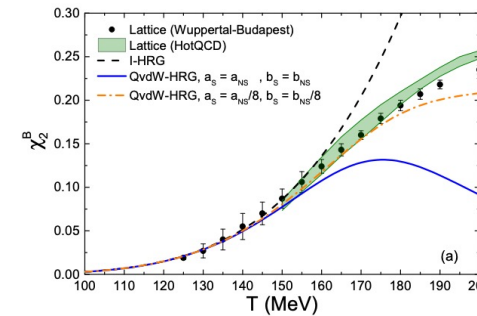
Interacting HRG can be more smoothly connected to other QCD phases than ideal HRG

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

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

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

- 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

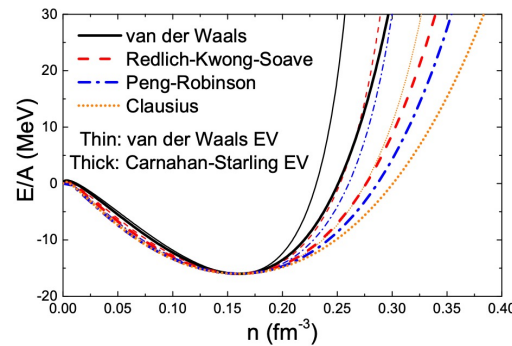
A. Sorensen, V. Koch, PRC 104, 034904 (2021)

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

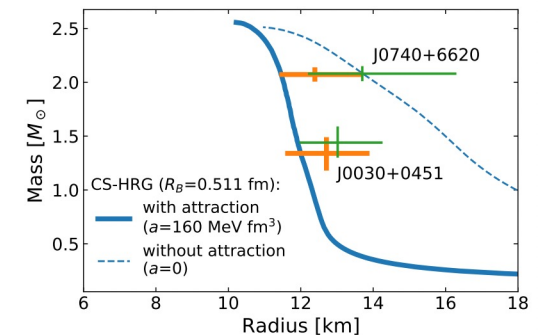
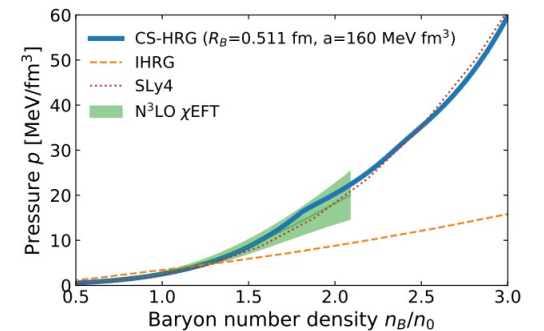
Available out-of-the-box in FIST in next version (already present in **devel** branch on github), adding leptons into the list one can do neutron-star matter

Another extension: pion interactions and condensation at finite isospin density

VV et al., PRL (2021)



VV, PRC 96, 015206 (2017)



Fujimoto et al., 2109.06799

# 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!**