BSQ Hydrodynamics and Challenges with a 4D equation of state

Collaborators:
*Dekrayat **Almaalol**, Travis **Dore**,*
*Debora **Mroczek**, Jordi **Salinas San Martin**,*
*Willian **Serenone**, Lydia **Spychalla**,*
*Patrick **Carzon**, Matthew **Sievert**,*
*Fernando **Gardim**, and Jacquelyn **Noronha-Hostler***

Reference: arXiv:2405.09648 [nucl-th]

Christopher Plumberg
Pepperdine University

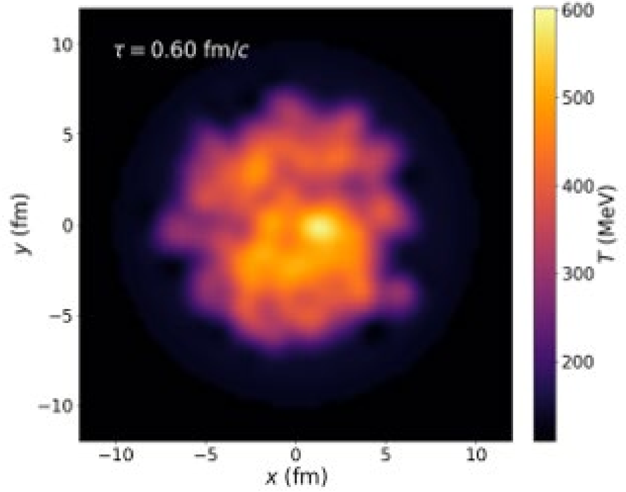
CPOD 2024
Lawrence Berkeley National Laboratory, Berkeley, California
May 20-24, 2024

BSQ Simulations

– **Equations of motion**:

$\nabla_\mu T^{\mu\nu} = 0$

(*energy-momentum conservation*)



– **Propagate densities**:

e (*energy density*) or

s (*entropy density*)

– Equations of motion closed by **equation of state** (EoS): $P = P(T)$

– **Need to know** T coordinate for a **given** e/s point

BSQ Simulations

– **Equations of motion**:

$$\nabla_\mu T^{\mu\nu} = 0$$

(*energy-momentum conservation*)

$$\nabla_\mu J_i^\mu = 0 \quad (i = B, S, Q)$$

(*charge conservation*)

– **Propagate densities**:

e (*energy density*) or

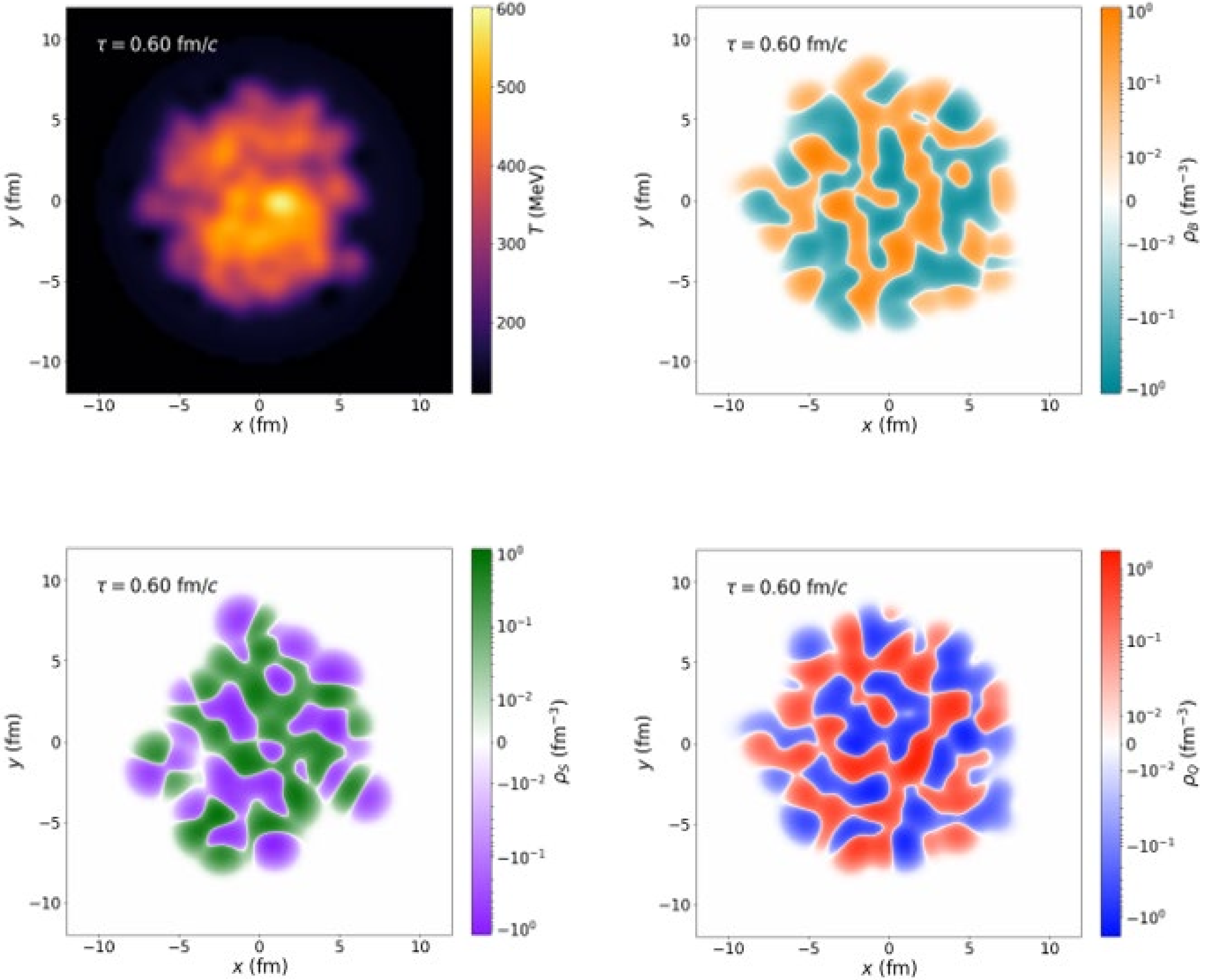
s (*entropy density*)

ρ_B (*baryon density*)

ρ_S (*net strangeness density*)

ρ_Q (*electric charge density*)

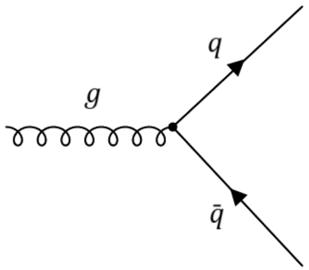
– Equations of motion closed by **equation of state** (EoS): $P = P(T, \{\mu_i\})$

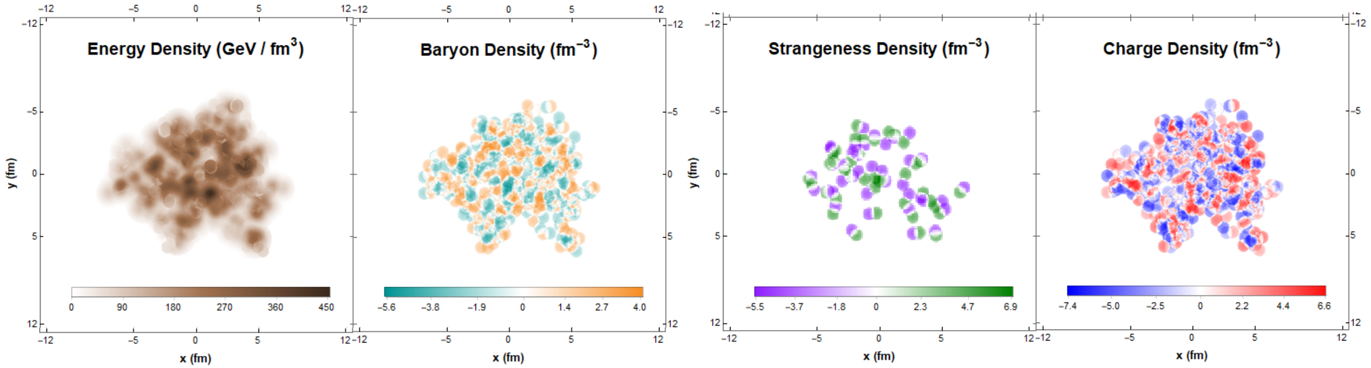


– **Need to know** (T, μ_B, μ_S, μ_Q) coordinates for a **given** $(e/s, \rho_B, \rho_S, \rho_Q)$ point

BSQ in a nutshell

- ***Introducing new code for BSQ hydrodynamics:***
  CCAKE = Conserved ChArges with hydrodynamiK Evolution
- Initial conditions generated by ***ICCING***:
  gluon splitting in CGC framework to generate *local charge fluctuations*





[arXiv:1911.12454 [nucl-th]]

- Hydrodynamic evolution equations use *minimal Israel-Stewart* formalism
  - **new** equations of motion for conserved charges
  - **new** transport coefficients (in addition to shear, bulk, etc.)
  - currently assume ideal evolution in charge sector (i.e., no diffusion)

BSQ Thermodynamics

This work: lattice QCD EoS given by Taylor expansion of pressure in powers of chemical potentials

$$\frac{P(T,\mu_B,\mu_Q,\mu_S)}{T^4} = \sum_{i,j,k} \frac{1}{i!j!k!}\chi_{i,j,k}^{BQS} \left(\frac{\mu_B}{T}\right)^i \left(\frac{\mu_Q}{T}\right)^j \left(\frac{\mu_S}{T}\right)^k$$

Claudia Ratti 2018 *Rep. Prog. Phys.* **81** 084301

J. Noronha-Hostler, P. Parotto, C. Ratti, J. Stafford PRC 100 (2019),
A. Monnai et al., PRC 100 (2019)

"Susceptibilities" $\chi_{i,j,k}^{BQS}$ functions of temperature; matched to lattice QCD at high T and hadron resonance gas at low T

Charge/entropy densities obtained by taking derivatives w.r.t. P
Energy density obtained using Gibbs' relation

How do we invert given set of densities for corresponding phase diagram coordinates?

Root-finding in the equation of state

Goal: obtain $(T_0, \mu_{B,0}, \mu_{S,0}, \mu_{Q,0})$ from $(e_0, \rho_{B,0}, \rho_{S,0}, \rho_{Q,0})$

Given:

$$\begin{aligned} e &= e(T, \vec{\mu}) \\ \rho_B &= \rho_B(T, \vec{\mu}) \\ \rho_S &= \rho_S(T, \vec{\mu}) \\ \rho_Q &= \rho_Q(T, \vec{\mu}) \end{aligned}$$

→

Solve:

$$\begin{aligned} e_0 &= e(T_0, \vec{\mu}_0) \\ \rho_{B,0} &= \rho_B(T_0, \vec{\mu}_0) \\ \rho_{S,0} &= \rho_S(T_0, \vec{\mu}_0) \\ \rho_{Q,0} &= \rho_Q(T_0, \vec{\mu}_0) \end{aligned}$$

Construct interpolants from table of equation-of-state (e.g., LQCD) data

Couple to multi-dimensional rootfinder (e.g., via GSL library)

Current default functionality of `CCAKE`

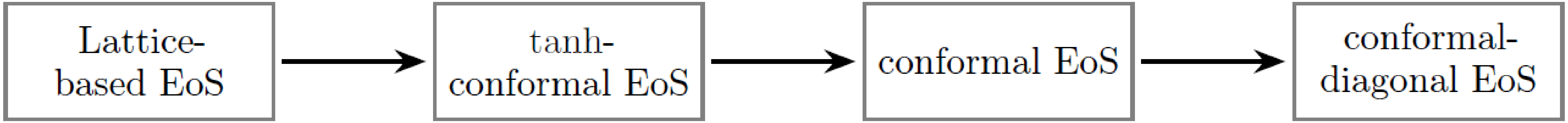
Alternative: Delaunay interpolation + k-d trees (see back-up slides)

What if the numerical inversion fails?

- Obviously the ideal is that this never happens
- This can happen for any of a number of reasons:
  - The true solution **may exist outside** the current grid
  - There may not be ***any* solution** for the chosen equation of state
  - There may be **multiple "correct" solutions**
- Q: *When this happens in hydrodynamics,*
  how should we close the equations of motion?
- A: Supplement with an alternative "back-up" EoS at $\vec{\mu} = 0$
  which closely approximates primary EoS

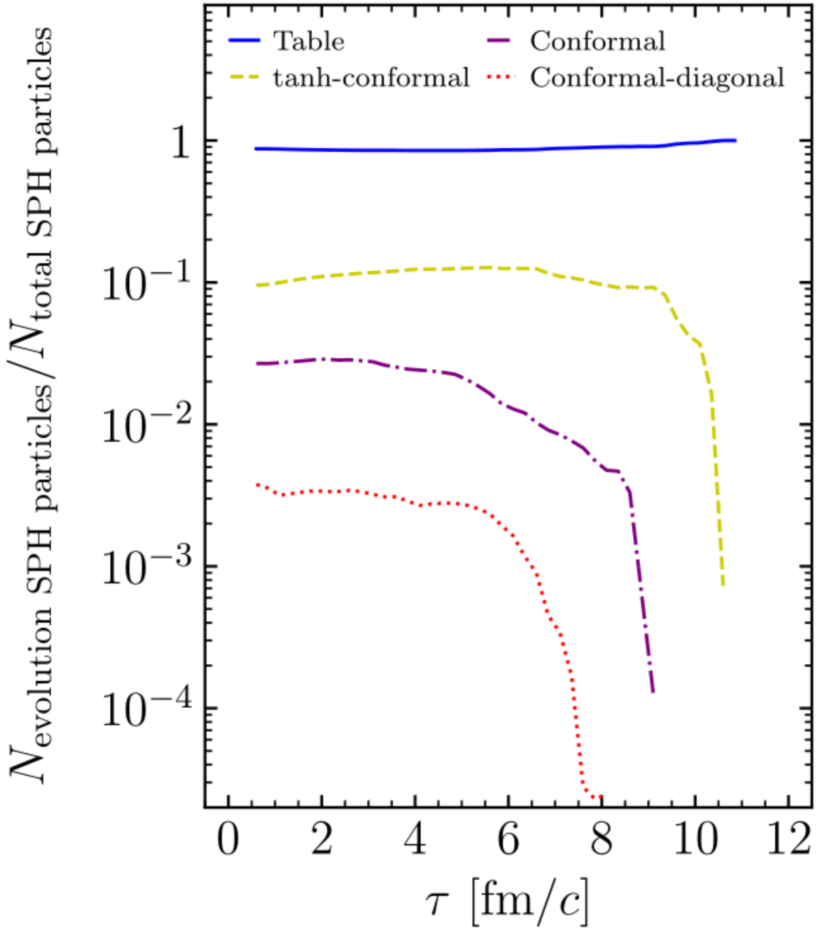
“Back-up” Equations of State

- If the preferred (read: *tabulated lattice QCD*) EoS fails to yield a unique solution, then “fall back” to an alternative EoS which can provide a solution



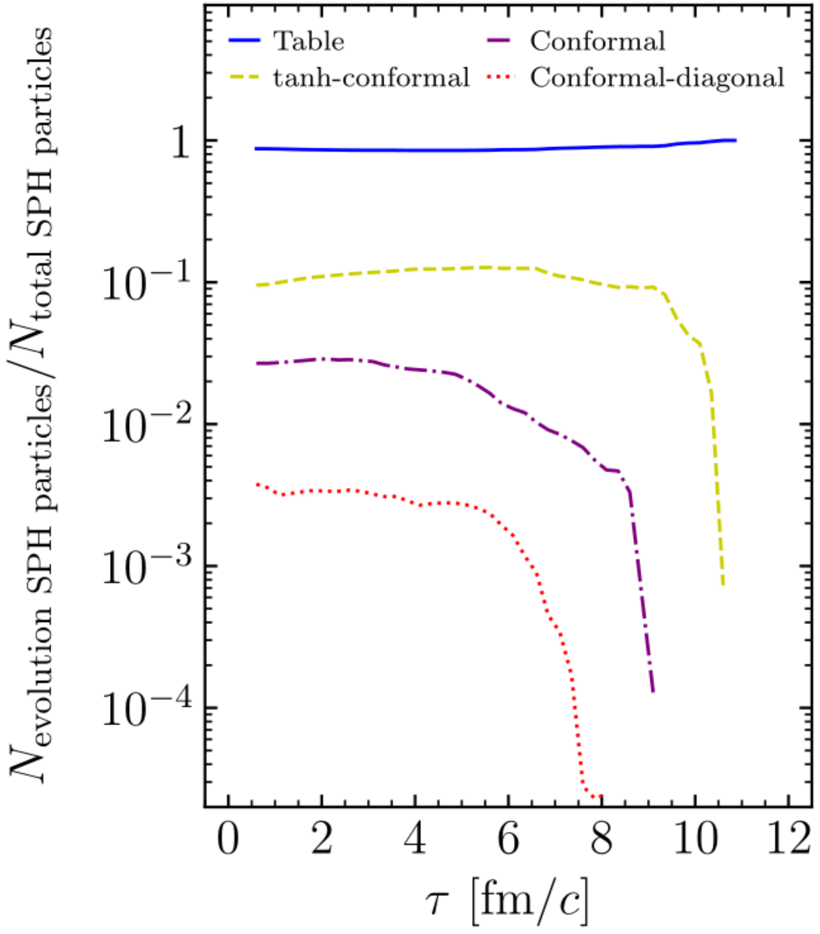
- Available back-ups:
  - “Tanh-conformal” EoS provides better approximation to lattice at $\vec{\mu} = 0$
  - Conformal EoS
  - Conformal-diagonal EoS (guarantees existence of solution)
- Explicit parametrizations in backup slides

Comparison: lattice vs. "back-up" EoSs

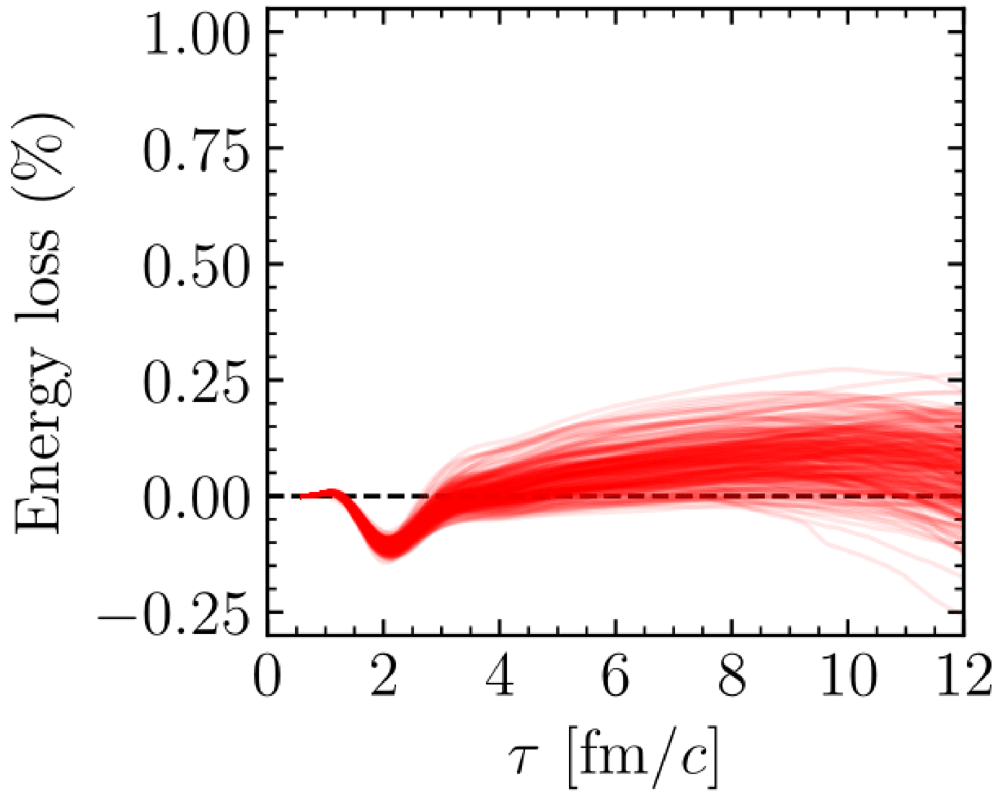


Relative fraction of fluid cells
in respective EoSs (not frozen-out)

Comparison: lattice vs. “back-up” EoSs

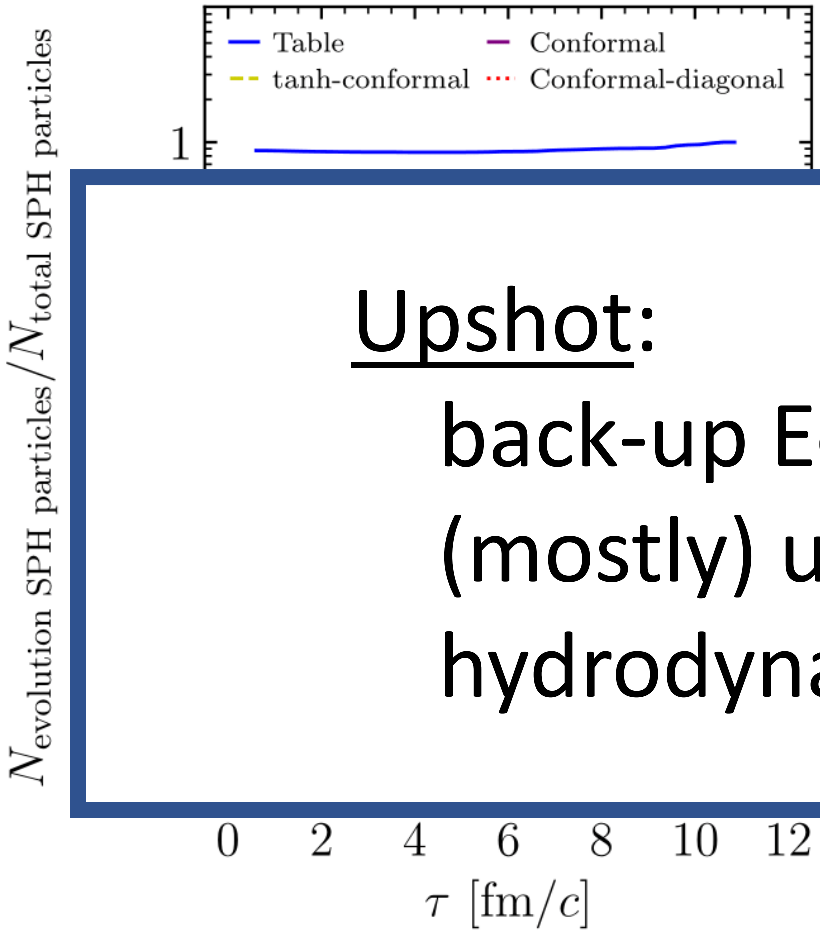


Relative fraction of fluid cells in respective EoSs (not frozen-out)

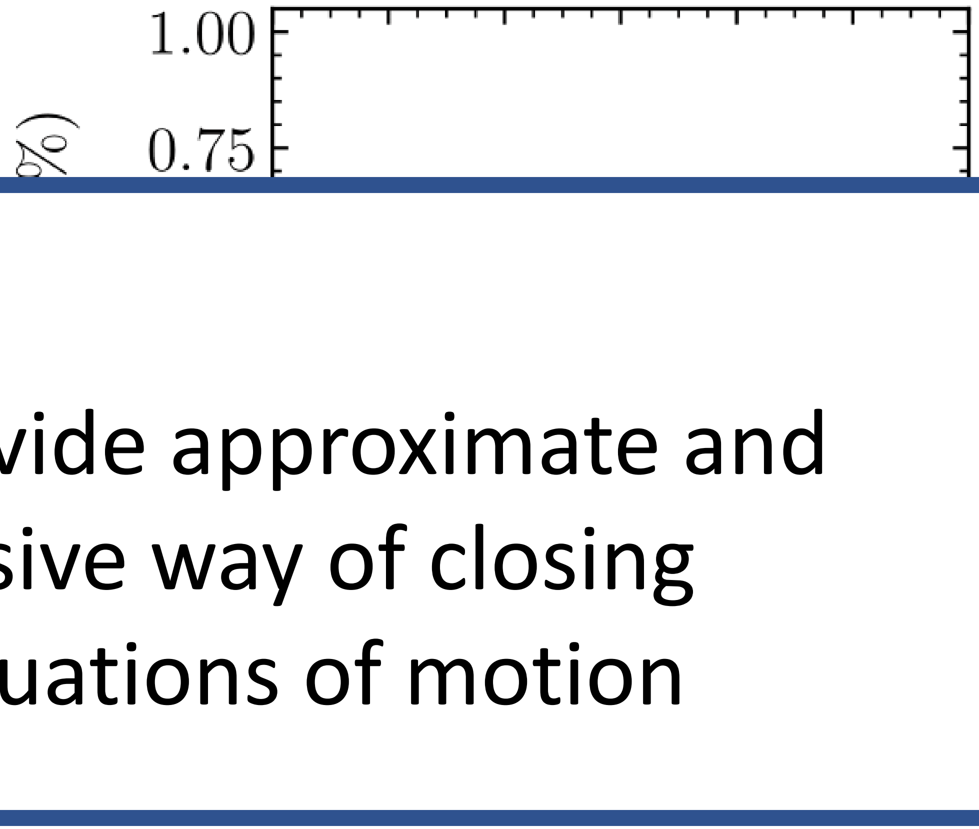


Switching to back-up EoSs produces small violations of energy conservation; integrated violations $\lesssim 0.25\%$

Comparison: lattice vs. “back-up” EoSs



Relative fraction of fluid cells
in respective EoSs (not frozen-out)



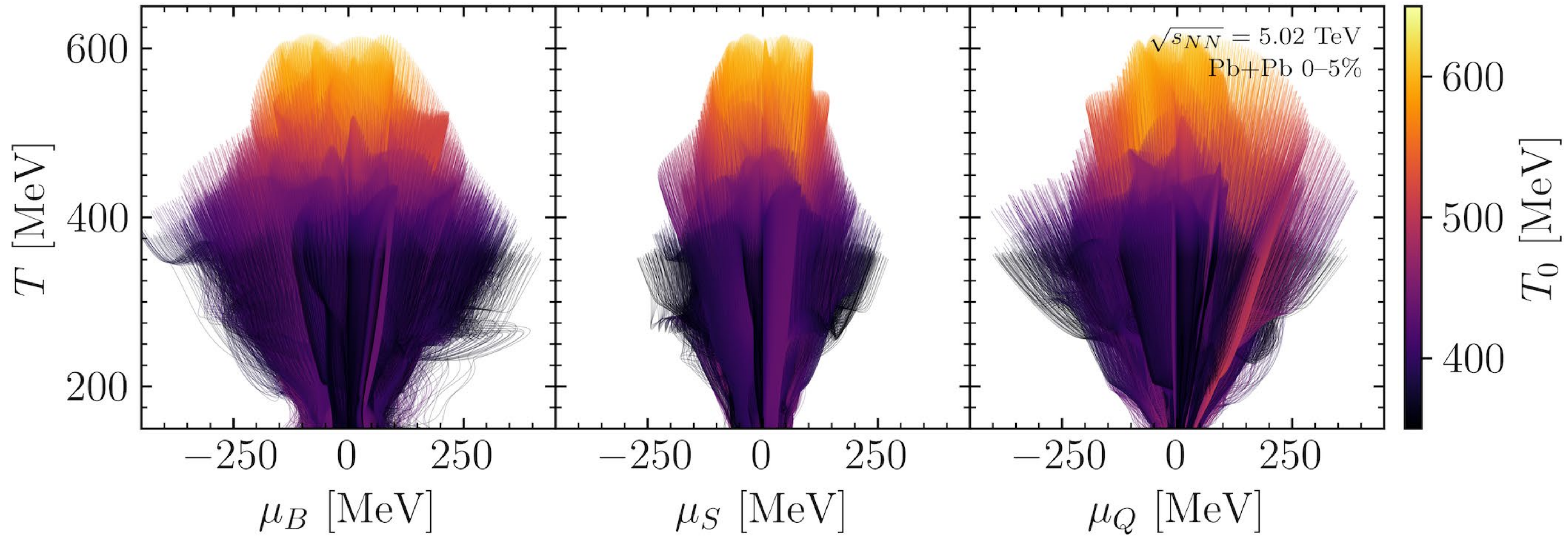
Switching to back-up EoSs produces small violations of energy conservation; integrated violations $\lesssim 0.25\%$

Upshot:

back-up EoSs provide approximate and (mostly) unobtrusive way of closing hydrodynamic equations of motion

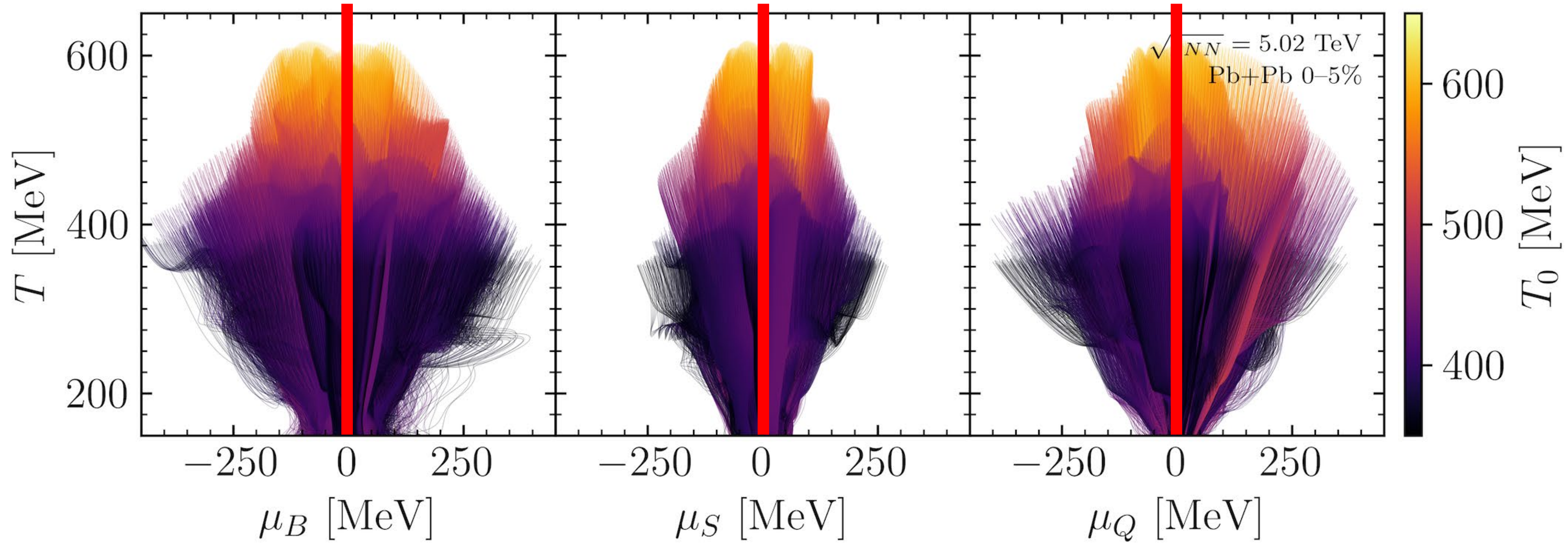
A few results

Phase diagram trajectories (0-5%)



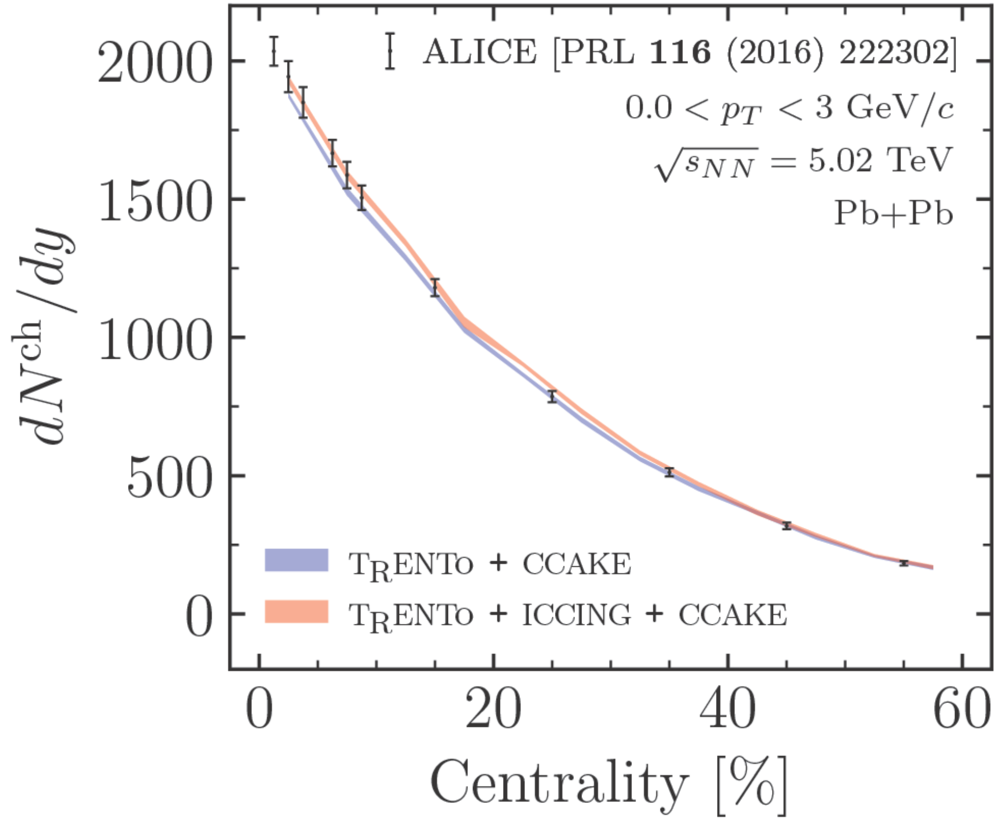
- $\vec{\mu}$ magnitudes initially reach up to 200-400 MeV, even at LHC
- <u>Large fluctuations</u> in $\vec{\mu}$, <u>average</u> still consistent with zero
- Magnitudes still O(50-100 MeV) near freeze out

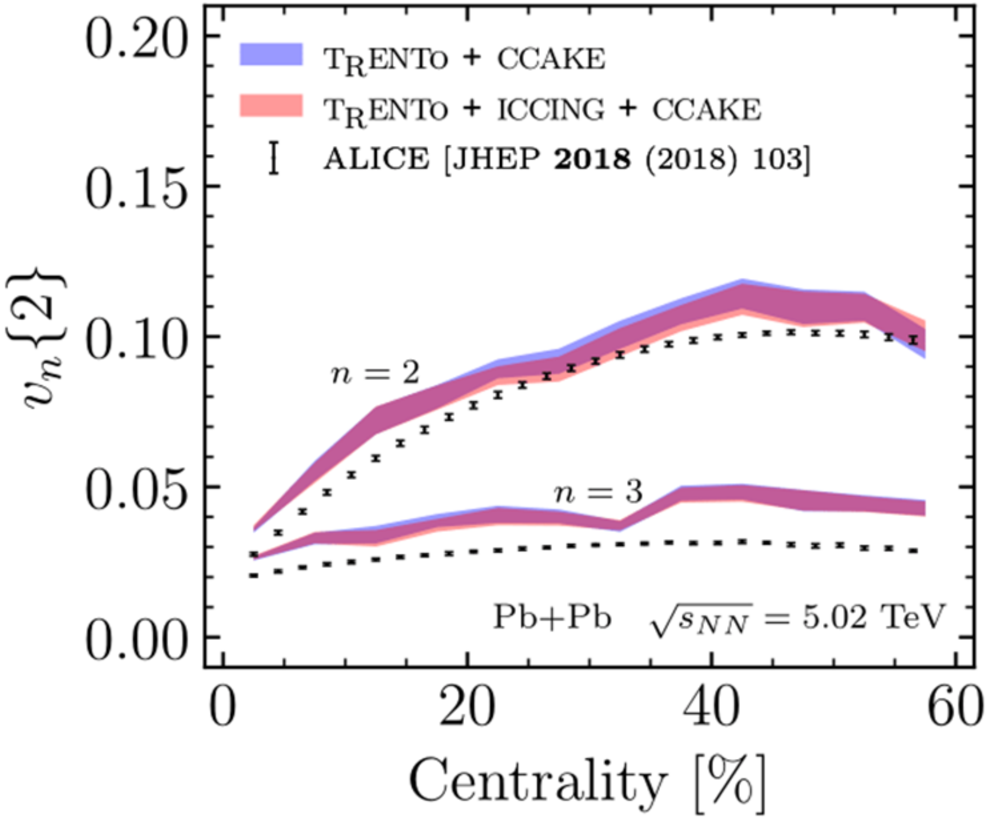
Phase diagram trajectories (0-5%)



- $\vec{\mu}$ magnitudes initially reach up to 200-400 MeV, even at LHC
- Large fluctuations in $\vec{\mu}$, average still consistent with zero
- Magnitudes still O(50-100 MeV) near freeze out, but $|\langle \vec{\mu}_{\mathrm{FO}} \rangle| \leq 0.5$ MeV

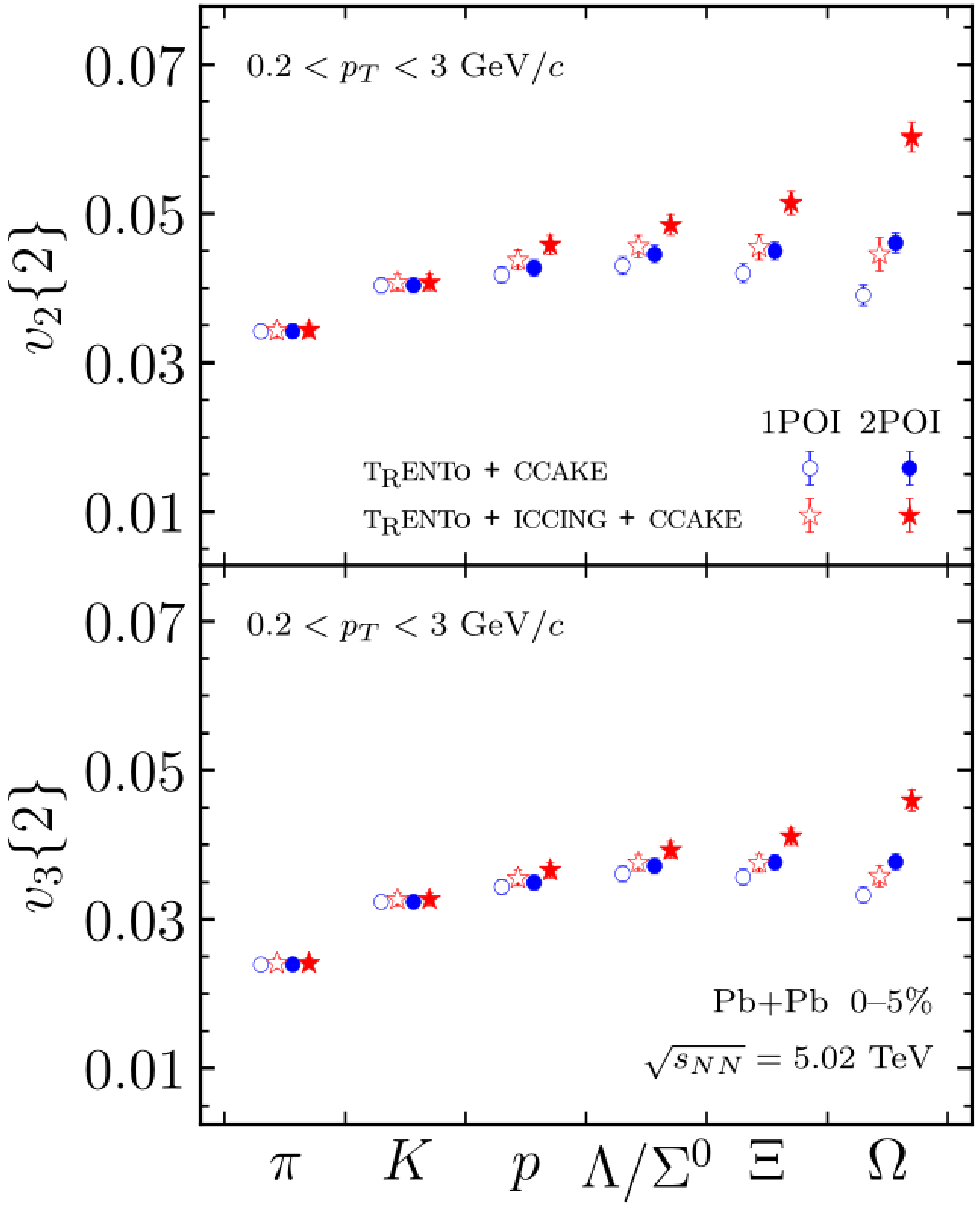
Observables





- All charged particle pseudorapidity densities, flow reproduced well, *with* or *without* initial state charge fluctuations
- Note: $\langle \rho_B \rangle = \langle \rho_S \rangle = \langle \rho_Q \rangle = 0$

Observables



- Effects of charge fluctuations become visible with one or two particles of interest (POIs – *see definitions in backup slides*)
- Especially sensitive in strange sector because of larger fluctuations
- Opportunity to probe effects of charge fluctuations directly(!)

Summary

- ***New*** hydrodynamics code with conserved charges: CCAKE
- Available for download here:
  https://github.com/the-nuclear-confectionery/CCAKE
- Back-up EoSs used to stabilize hydrodynamic evolution
- Charge fluctuations can reach large values *even at LHC energies*
- Good description of "all charged particles" observables,
  nPOI flow observables sensitive to charge fluctuations (n=1,2)
- Open challenges
  - Finding fast and stable ways to implement multi-dimensional EoS
  - Improved treatment of BSQ initial conditions, transport coefficients, etc.

Thank you!

Backup slides

“Back-up” EoS #1: “Tanh-conformal”

- Definition:

$$p_{\rm tc}(T,\mu_B,\mu_S,\mu_Q) = \frac{A_0 T_0^4}{2}\left[1+\tanh\left(\frac{T-T_c}{T_s}\right)\right]\left(\left(\frac{T}{T_0}\right)^2+\left(\frac{\mu_B}{\mu_{B,0}}\right)^2+\left(\frac{\mu_S}{\mu_{S,0}}\right)^2+\left(\frac{\mu_Q}{\mu_{Q,0}}\right)^2\right)^2$$

- Scale parameters determined to mimic tabulated EoS at high T as closely as possible:

$$A_0 \equiv p_{T,0}/T_{\rm scale}^4$$
$$T_0 \equiv 1~{\rm fm}^{-1}$$
$$\mu_{X,0} \equiv \frac{A_0^{1/4} T_0 \mu_{X,\max}}{\sqrt{\sqrt{p_{X,\max}}-\sqrt{p_{T,0}}}},$$
$$(X = B,\, S\,, Q)$$

where

$$p_{T,0} \equiv p_{\rm table}(T_{\rm scale},0,0,0)$$
$$p_{B,\max} \equiv p_{\rm table}(T_{\rm scale},\mu_{B,\max},0,0)$$
$$p_{S,\max} \equiv p_{\rm table}(T_{\rm scale},0,\mu_{S,\max},0)$$
$$p_{Q,\max} \equiv p_{\rm table}(T_{\rm scale},0,0,\mu_{Q,\max})$$

- Three additional parameters in tanh(): $T_{\rm scale} = 165~{\rm MeV},\ T_c = 220~{\rm MeV},\ T_s = 120~{\rm MeV}$

“Back-up” EoS #2: “Conformal”

- Definition:

$$p_{\mathrm{c}}(T, \mu_B, \mu_S, \mu_Q) = A_0 T_0^4 \left(\left(\frac{T}{T_0} \right)^2 + \left(\frac{\mu_B}{\mu_{B,0}} \right)^2 + \left(\frac{\mu_S}{\mu_{S,0}} \right)^2 + \left(\frac{\mu_Q}{\mu_{Q,0}} \right)^2 \right)^2 ,$$

- Not the most general (any quartic combinations are acceptable)
- Scale parameters determined as in “Tanh-conformal”

“Back-up” EoS #3: “Conformal-diagonal”

- Definition:

$$p_{\rm cd}(T,\mu_B,\mu_S,\mu_Q) = A_0 T^4 \left(\left(\frac{T}{T_0}\right)^4 + \left(\frac{\mu_B}{\mu_{B,0}}\right)^4 + \left(\frac{\mu_S}{\mu_{S,0}}\right)^4 + \left(\frac{\mu_Q}{\mu_{Q,0}}\right)^4 \right),$$

- Scale parameters determined similarly to “Tanh-conformal”/“Conformal”
- One can prove

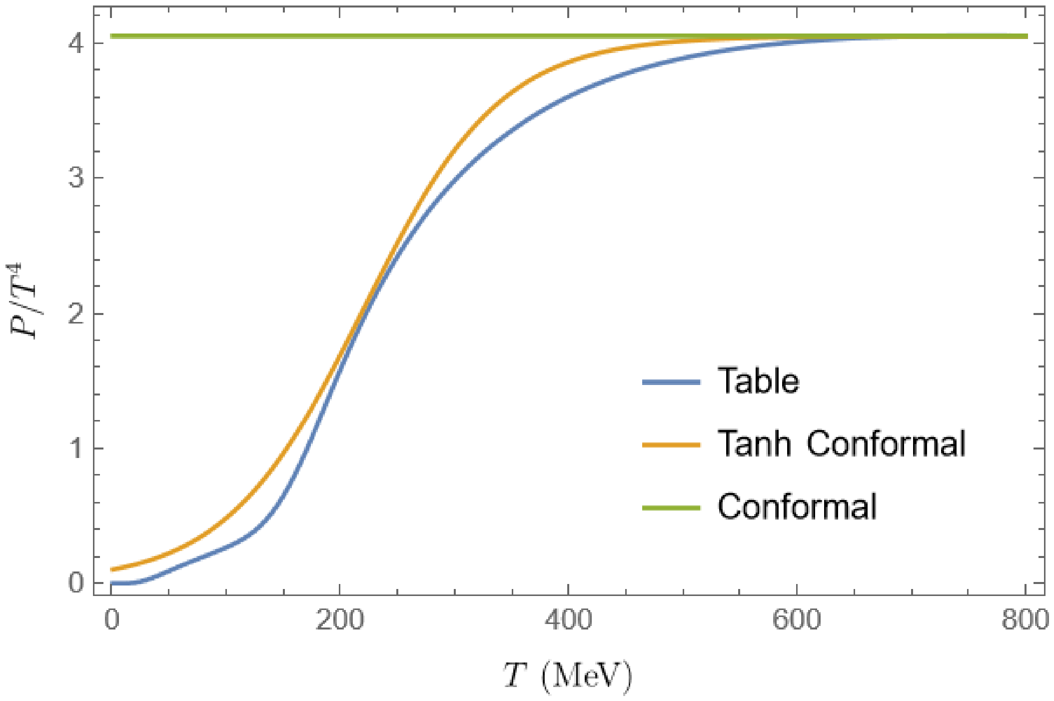
$$e \geq e_{\min}\left(\vec{\rho}\right) = \frac{3}{4 \cdot 2^{2/3}\left(A_0 T_0^4\right)^{1/3}} \left(\left(\mu_{B,0}\left|\rho_B\right|\right)^{4/3} + \left(\mu_{S,0}\left|\rho_S\right|\right)^{4/3} + \left(\mu_{Q,0}\left|\rho_Q\right|\right)^{4/3} \right)$$

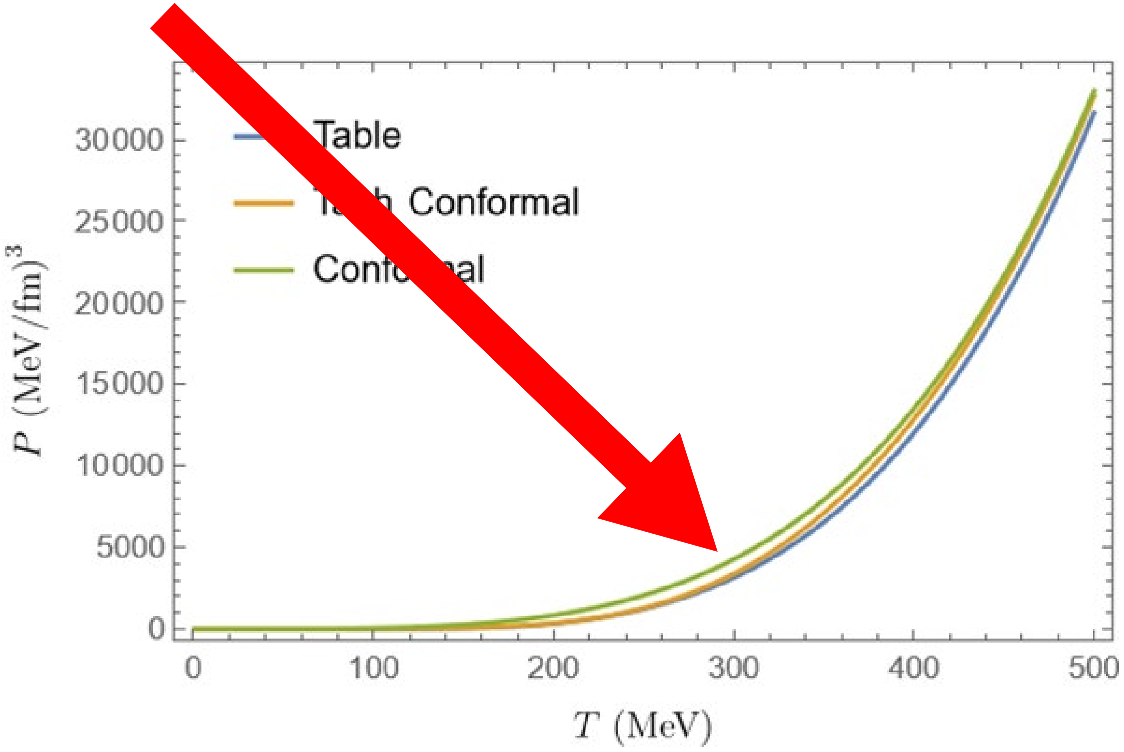
  is a necessary and sufficient condition for given set of (e,ρ_B,ρ_S,ρ_Q) to have a real solution

- If one propagates (s,ρ_B,ρ_S,ρ_Q), then a real solution is always guaranteed

Comparison: lattice vs. “back-up” EoSs

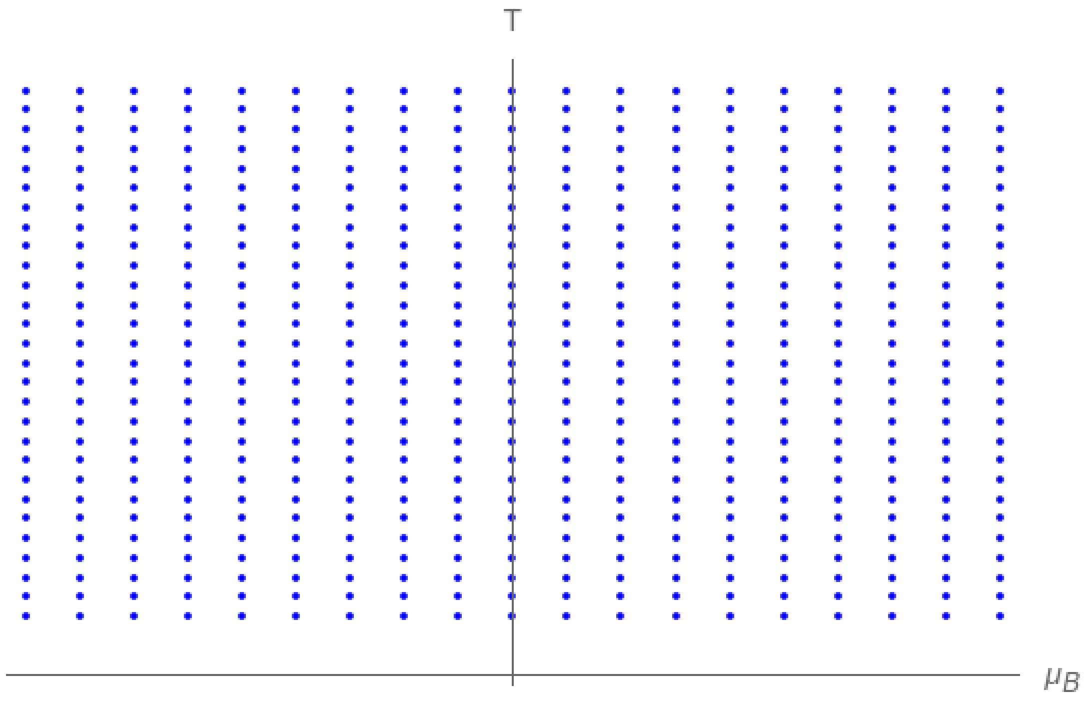
Switching to back-up EoSs produces small violations of energy conservation

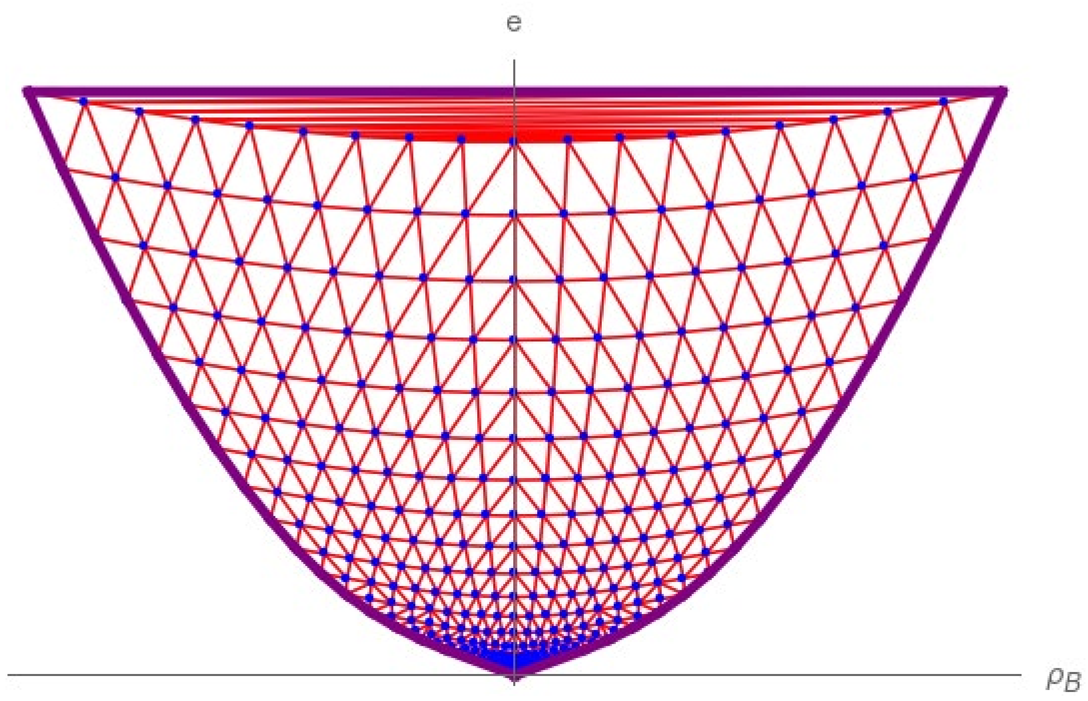




- Conformal-diagonal reduces to Conformal when $\mu_{\mathrm{B,S,Q}} = 0$
- Total energy depends on both energy and pressure
- Total integrated violations below $\sim 0.5\%$

Alternate Strategy: Delaunay interpolation



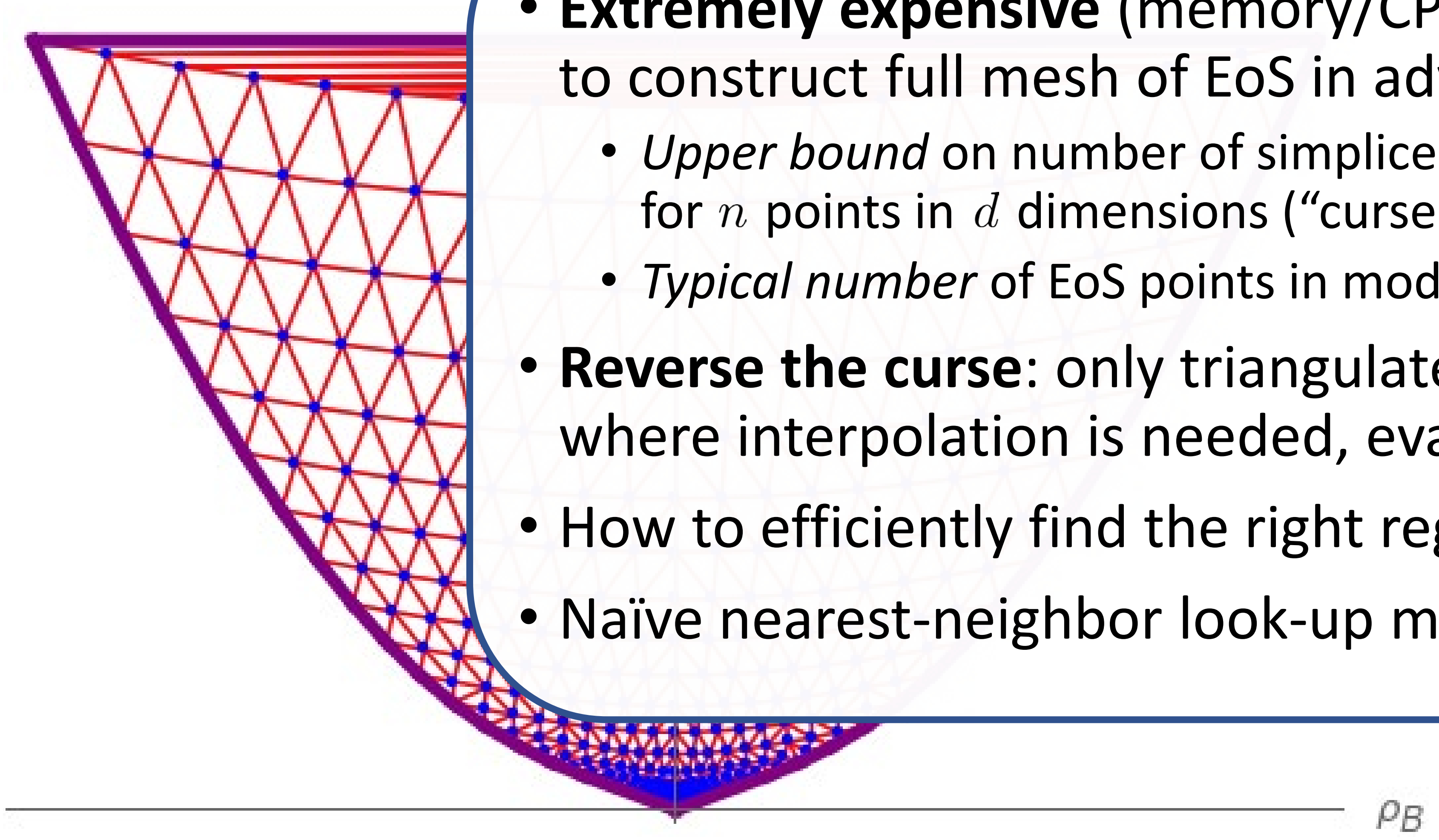


$$\text{Uniform } T\text{–}\mu_B \text{ grid} \quad \not\Rightarrow \quad \text{uniform } e\text{–}\rho_B \text{ grid}$$

– Perform linear interpolation on Delaunay triangulation of scattered density points

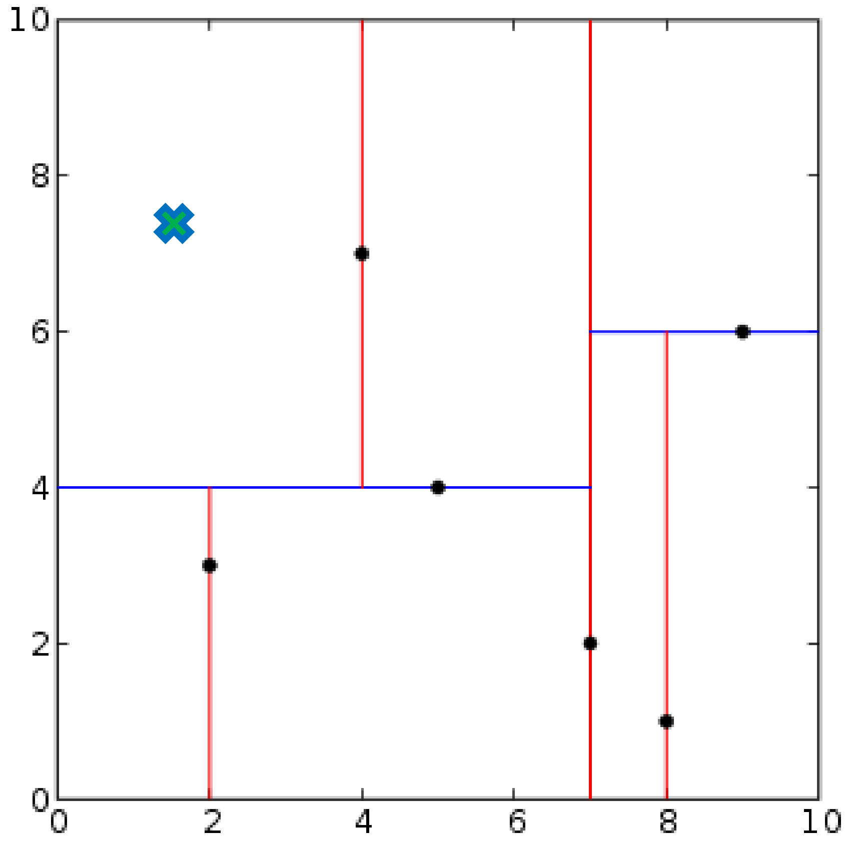
– Only defined inside **convex hull** (bold line)

Constructing the Delaunay mesh

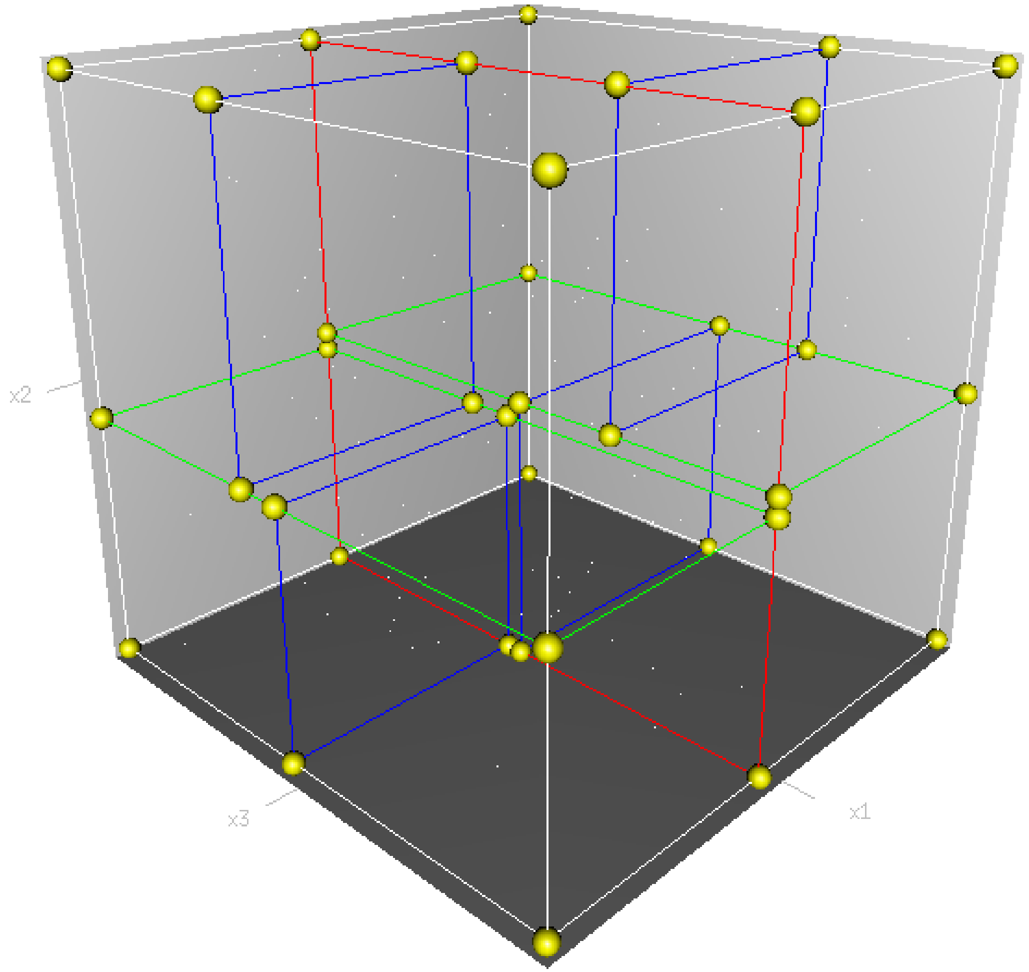


- **Extremely expensive** (memory/CPU time) to construct full mesh of EoS in advance
  - *Upper bound* on number of simplices grows like $O\left(n^{\lceil d/2 \rceil}\right)$, for n points in d dimensions ("curse of dimensionality")
  - *Typical number* of EoS points in modest grid: $O\left(10^5 - 10^7\right)$ in 4D
- **Reverse the curse**: only triangulate the region where interpolation is needed, evaluated at runtime
- How to efficiently find the right region to triangulate?
- Naïve nearest-neighbor look-up may be *very* inefficient

Finding closest simplex efficiently: *k*-d trees



2D example



3D example

A rough algorithm

Compute (T, μ_B, μ_S, μ_Q) distributions on scattered $(e, \rho_B, \rho_S, \rho_Q)$ grids

Identify convex hull inside of which density interpolation is defined

Build k-d trees of density grids

For given densities $(e_0, \rho_{B,0}, \rho_{S,0}, \rho_{Q,0})$:

– *locate* containing / neighboring simplices

– *construct* Delaunay triangulation

– *evaluate* unique linear interpolant at input densities

Download: `https://github.com/astrophysicist87/eos_delaunay_demo` (see backup slides)

Code demo: Delaunay $(e, \rho_B, \rho_S, \rho_Q)$ interpolator

```
//========================================
// read path to input file from command line
string path_to_file = string(argv[1]);

//========================================
// set up EoS object
cout << "Initializing equation of state "
              "interpolator:" << endl;
cout << "  --> reading in equation of state "
              "table from: " << path_to_file << endl;
eos_delaunay EoS( path_to_file );

//========================================
// vectors to store input densities and
// interpolated result for (T,muB,muQ,muS)
vector<double> result(4, 0.0);
vector<double> point({ 5754.35,  0.00231029,
                       0.351709, 0.378919   }); // just an example
```

Code demo: Delaunay $(e, \rho_B, \rho_S, \rho_Q)$ interpolator

```
//========================================
// call the interpolator
const size_t n_repeat = 1000;
cout << "Calling the interpolator "
     << n_repeat << " times for test point: \n"
                    "  --> {e,B,S,Q} = {"
     << point[0] << " MeV/fm^3," << point[1] << " 1/fm^3, "
     << point[2] << " 1/fm^3, "  << point[3] << " 1/fm^3}" << endl;

// multiple calls to improve timing estimate
for (size_t i = 0; i < n_repeat; i++)
  EoS.interpolate(point, result);
```

Invocation: `$ ./interpolate_ebsq eos.dat`

```
================================================================================
= Code:     Equation of state interpolator
= Purpose:  Performance and closure tests of Delaunay interpolator
= Author:   Christopher Plumberg
= Contact:  plumberg@illinois.edu
= Date:     April 28, 2022
================================================================================

Initializing equation of state interpolator:
  --> reading in equation of state table from: eos.dat
       - read in 1000000 lines.
       - read in 2000000 lines.
       - read in 3000000 lines.
       - read in 4000000 lines.
       - read in 5000000 lines.
       - read in 6000000 lines.
       - read in 7000000 lines.
  --> check minima and maxima:
       - e:     0.0151087    729992      [MeV/fm^3]
       - B:     -25.56       25.56       [1/fm^3]
       - S:     -65.1234     65.1234     [1/fm^3]
       - Q:     -37.5927     37.5927     [1/fm^3]
  --> setting up kd-trees: finished in 7.63744 seconds!

Calling the interpolator 1000 times for test point:
  --> {e,B,S,Q} = {5754.35 MeV/fm^3,0.00231029 1/fm^3, 0.351709 1/fm^3, 0.378919 1/fm^3}
  --> exact result (units MeV):           252.5    52.5    52.5    52.5
  --> interpolated result (units MeV):    252.448    52.5715    52.571    52.5597

Average time to interpolate 0.00512986 s.
Total runtime: 56.6377 s.
```

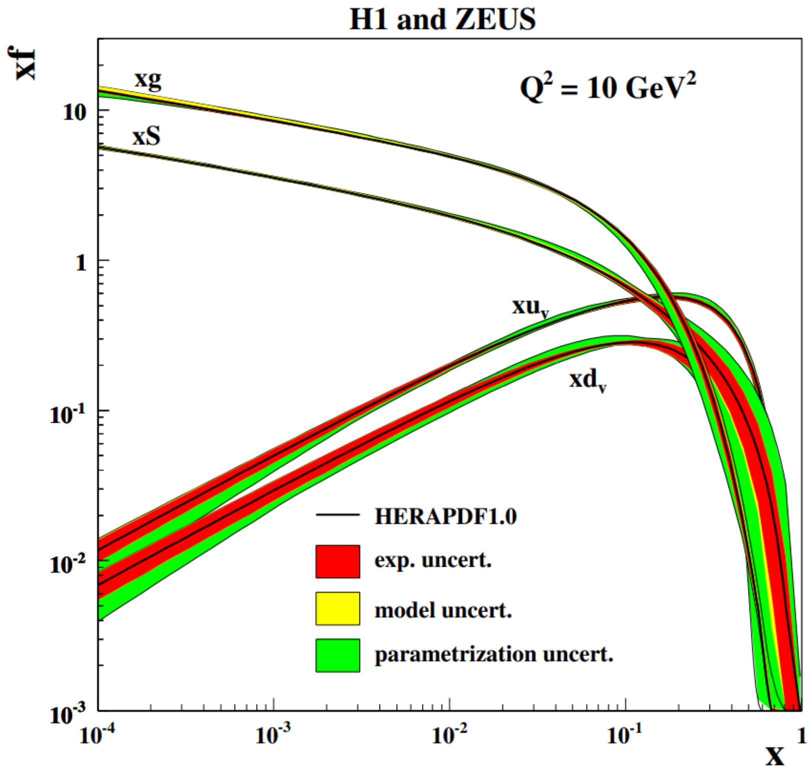
Moderate grid size

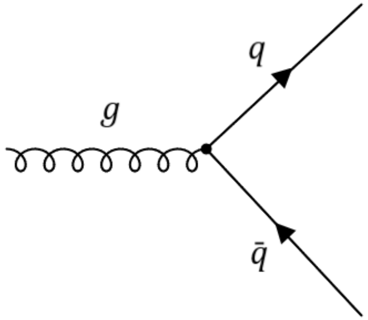
Reasonable accuracy

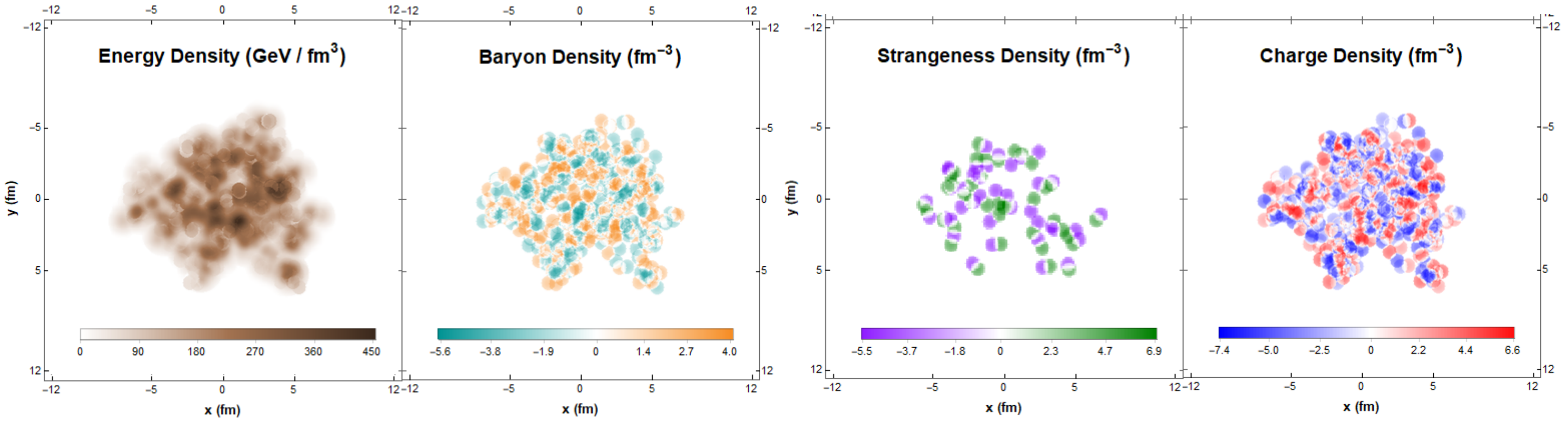
~ 200 solutions$/s$

BSQ Initial Conditions

- CCAKE accepts **ICCING** initial conditions (**I**nitial **C**onserved **C**harges **I**n **N**uclear **G**eometry)
- ICCING relies on the gluon-saturated initial state at mid-rapidity to determine probabilities for gluon splitting to quark pairs
- Use color-glass condensate (CGC) framework to generate *local charge fluctuations*







[arXiv:1911.12454 [nucl-th]]

BSQ Evolution

Israel-Stewart fluid dynamics

Dekrayat Almaalol, Travis Dore, Jacquelyn Noronha-Hostler [arXiv:2209.11210 [hep-th]]

$$S^\mu = su^\mu - \sum_q^{B,S,Q} \alpha_q n_q^\mu - \frac{1}{2}u^\mu \left(\beta_\Pi \Pi^2 + \beta_\pi \pi^{\mu\nu}\pi_{\mu\nu} + \sum_q^{B,S,Q} \beta_n^{qq'} n_q^\mu n_\mu^{q'} \right) - \sum_q^{B,S,Q} \left(\gamma_{n\Pi}^q n_q^\mu \Pi + \gamma_{n\pi}^q n_q^\nu \pi_\nu^\mu \right) - \frac{1}{2}(u^\nu \beta_{\Pi\pi} \Pi \pi_{\mu\nu})$$

equilibrium | 1^{st}-order term | 2^{nd}-order terms | 2^{nd}-order coupling terms

Second law of thermodynamics

$$\partial_\mu S^\mu = \frac{\beta_0}{2\eta}\pi_{\mu\nu}\pi^{\mu\nu} + \frac{\beta_0}{\zeta}\Pi^2 + \frac{1}{\kappa_{qq'}} n_\mu^q n_{q'}^\mu \geq 0$$

Hydrodynamic modeling with multiple conserved charges introduces a host of **new transport coefficients** characterizing charge diffusion, shear-diffusion couplings, etc.

NS Transport coefficients

$$\eta, \zeta, \kappa_{qq'}$$

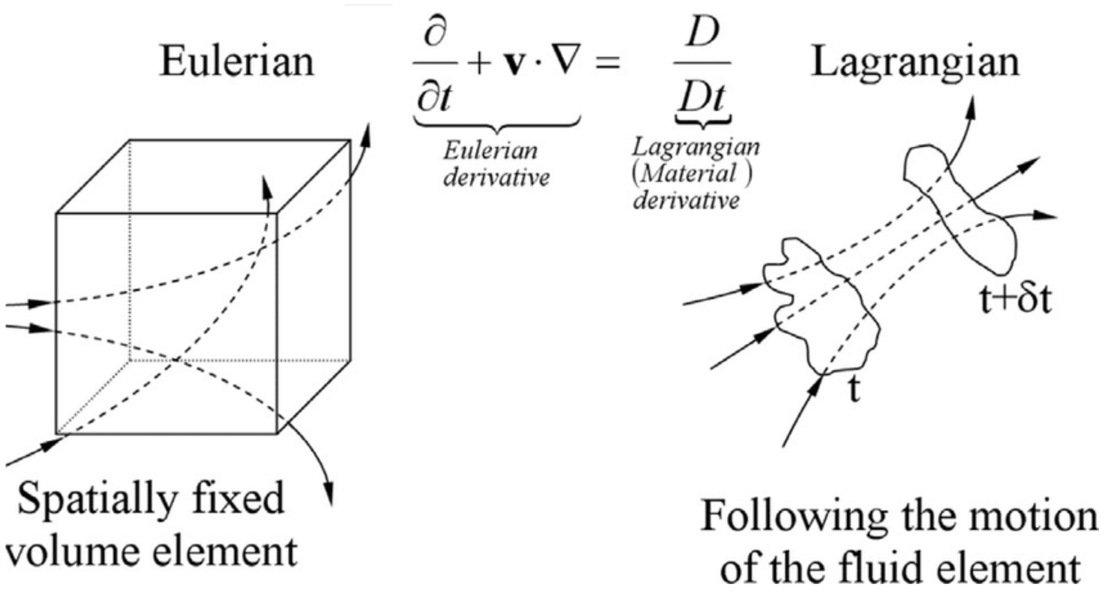
2^{nd} order Transport coefficients

$$\beta_\Pi, \beta_\pi, \gamma_{n\Pi}, \gamma_{n\pi}, \beta_{qq'}$$

Fotakis et al, 2203.11549 [nucl-th]

Slide credit: Dekrayat Almaalol

Smoothed Particle Hydrodynamics (SPH)

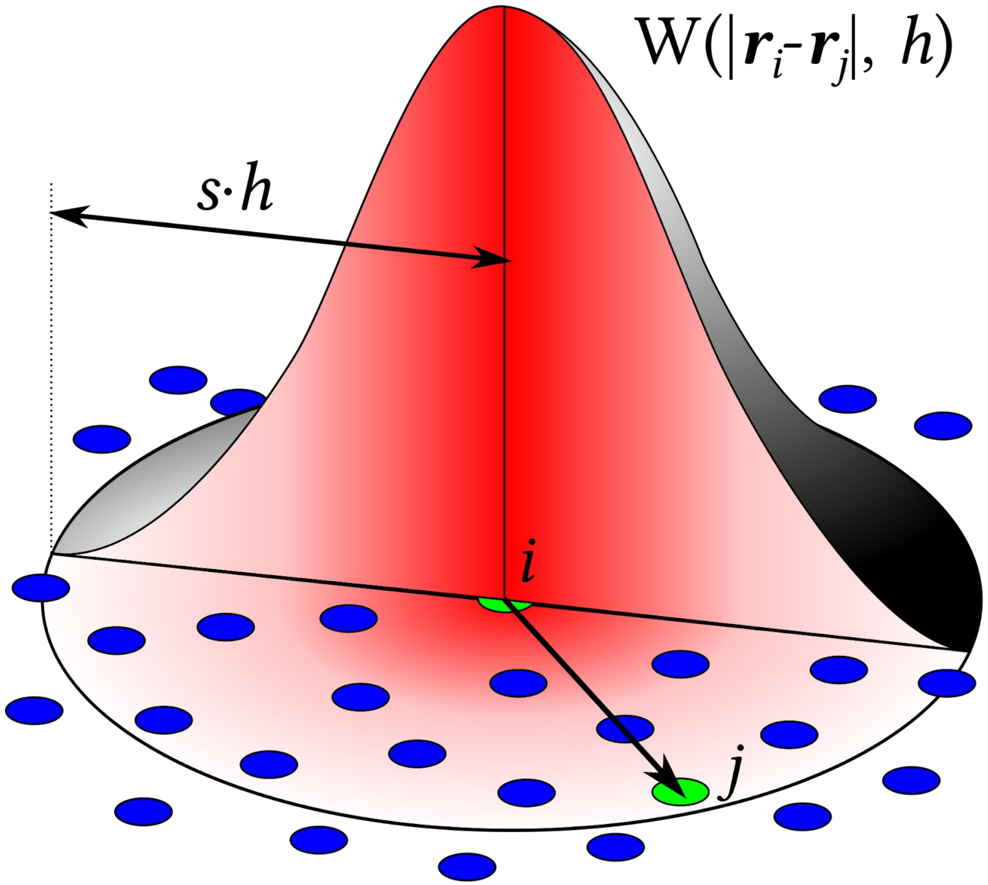


Grid-based hydrodynamics

Smoothed particle hydrodynamics

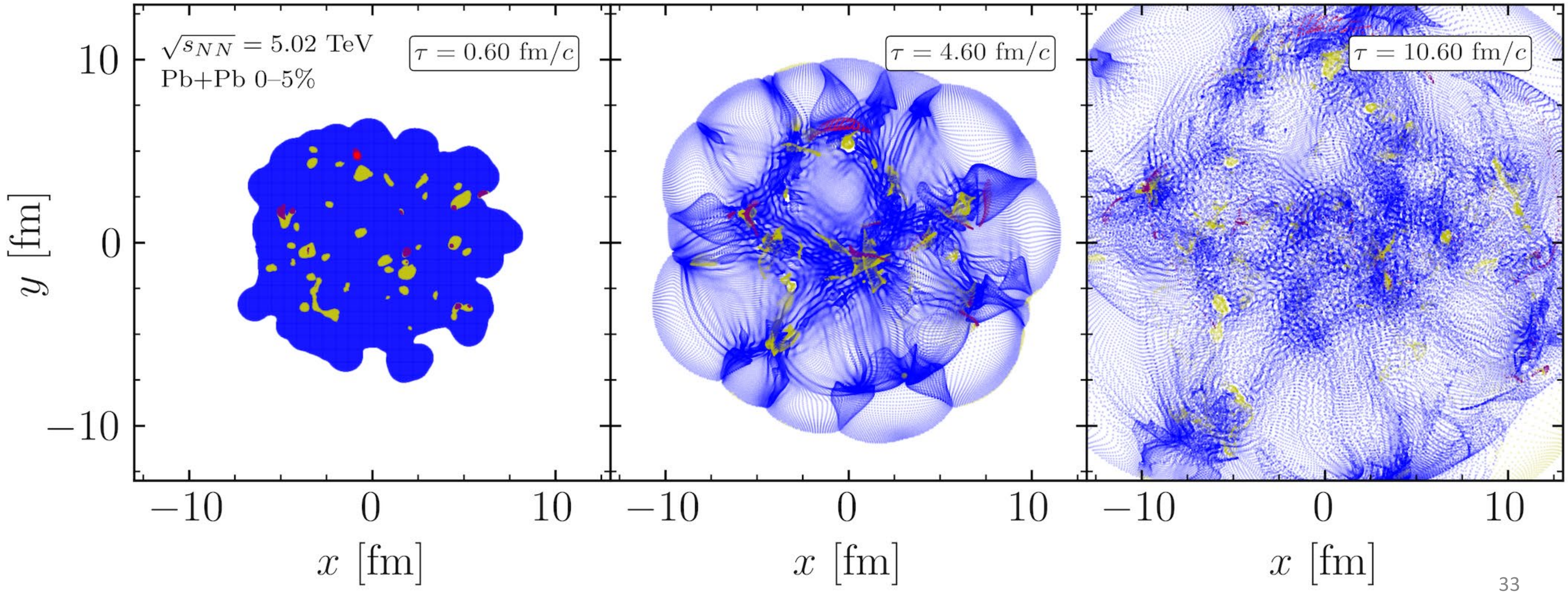
Conservation laws built-in by construction

Kernel function W imposes coarse-graining onto set of fictitious 'SPH particles'



EoS types in hydro

- Typical (central) Pb+Pb event showing EoS for each fluid cell
- **Blue: Table**
- **Green: Tanh-conformal**
- **Purple: Conformal**
- **Red: Conformal-diagonal**

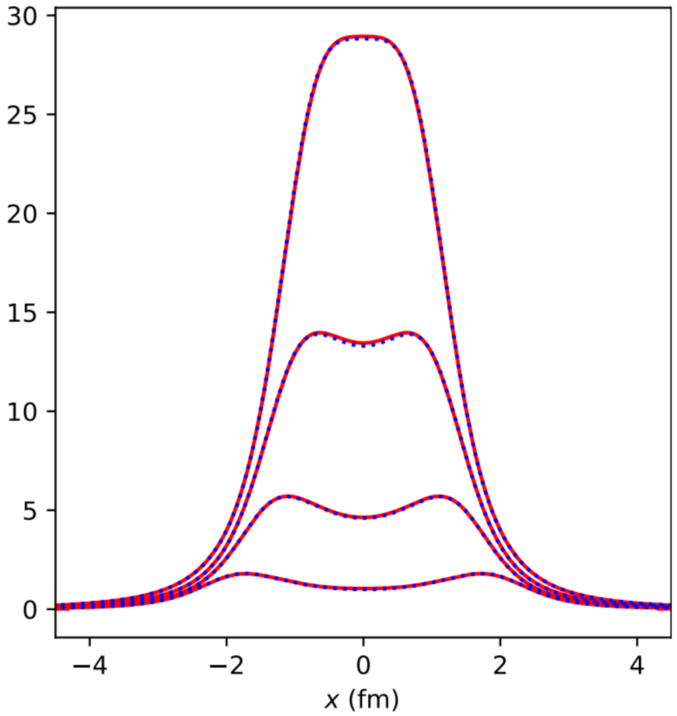


Gubser checks

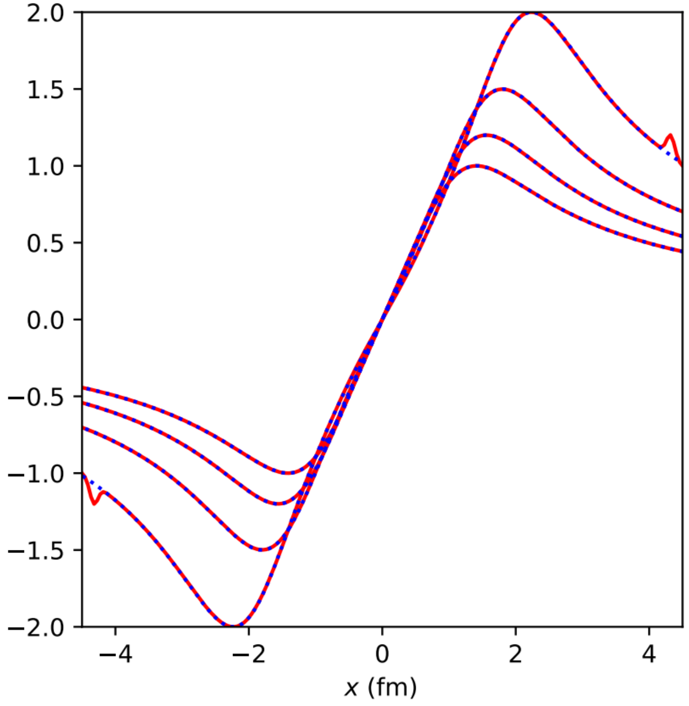
Blue (dotted): exact

Red (solid): CCAKE

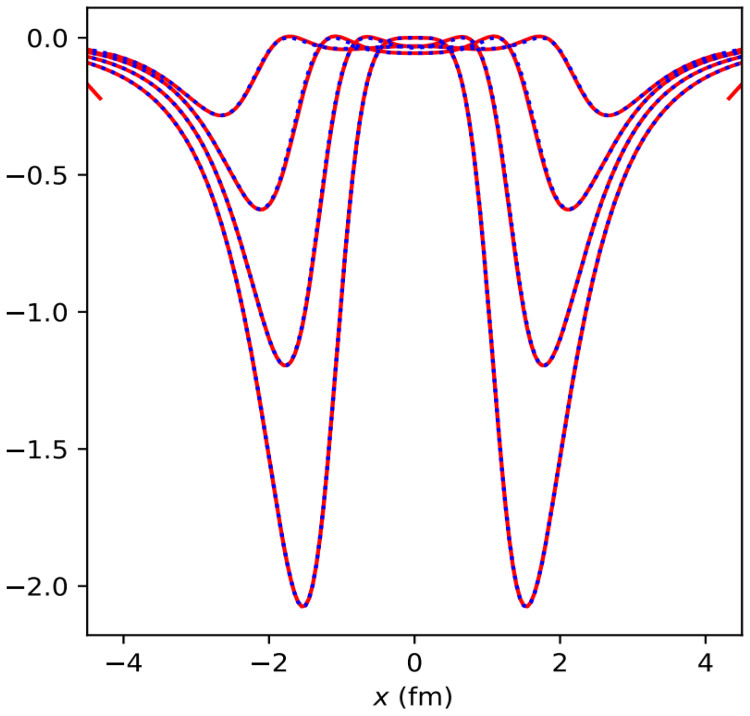
$\tau = 1.0,\ 1.2,\ 1.5,\ 2.0\ \mathrm{fm}/c$



Energy density: $e\ (\mathrm{fm}^{-4})$

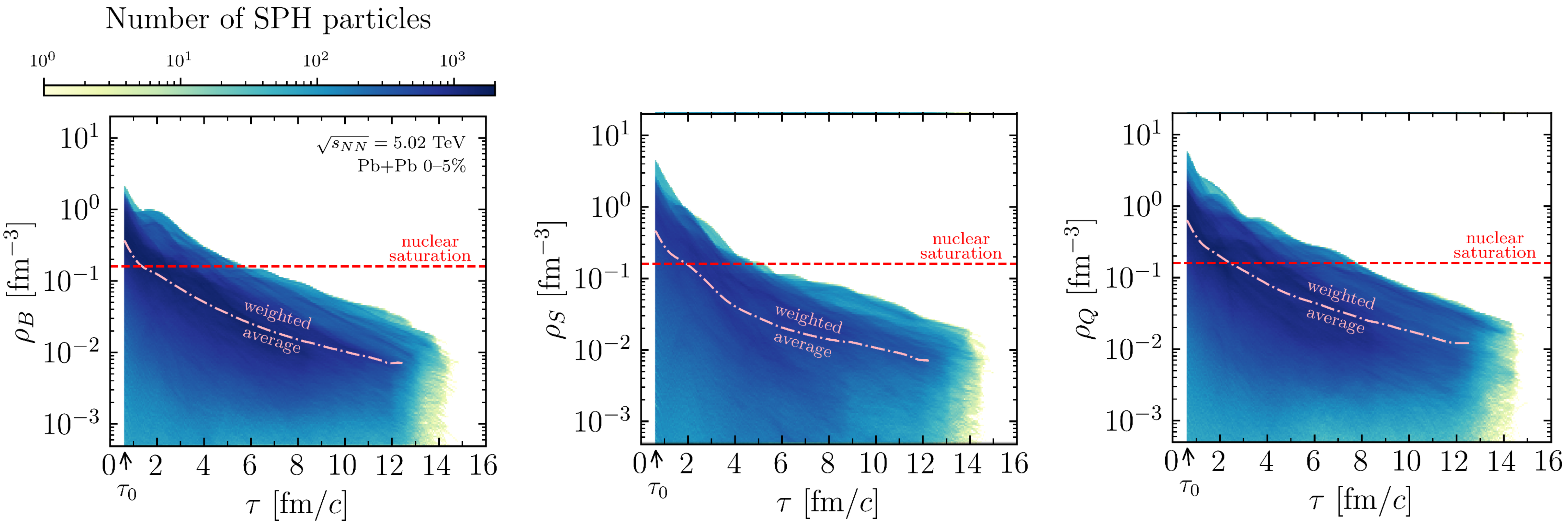


Flow velocity: u^r



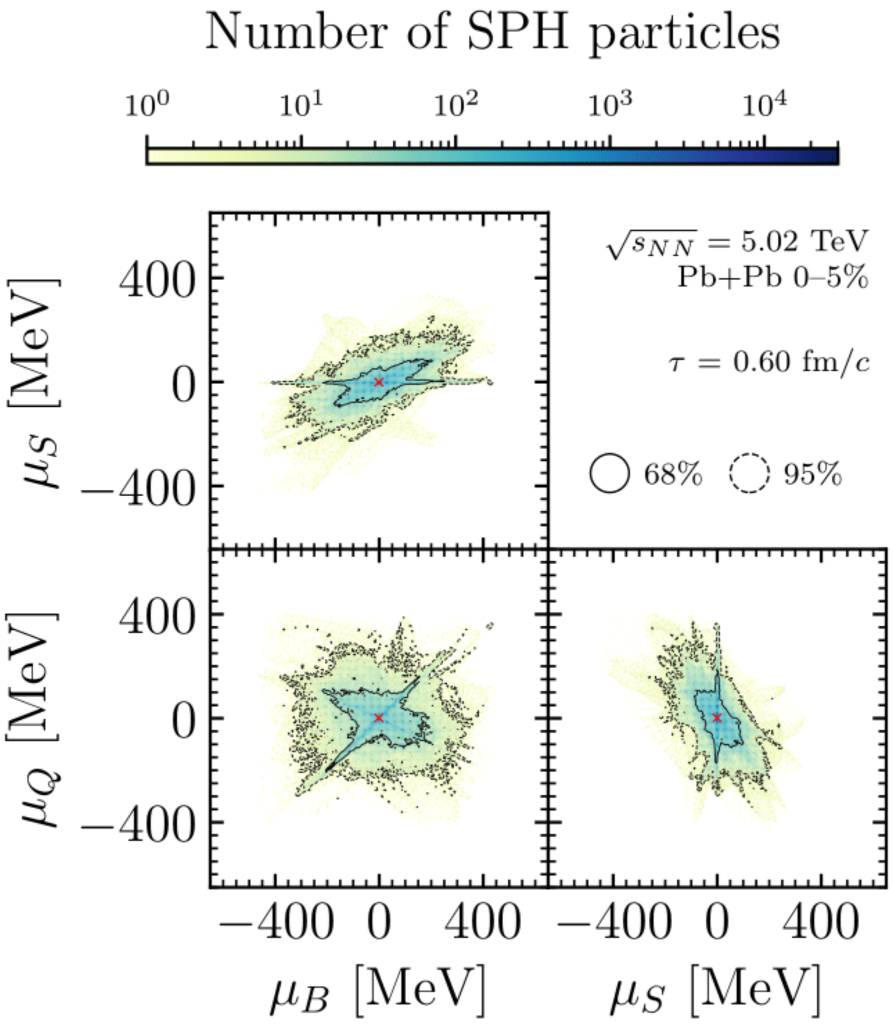
Shear stress: $\pi^{xx}\ (\mathrm{fm}^{-4})$

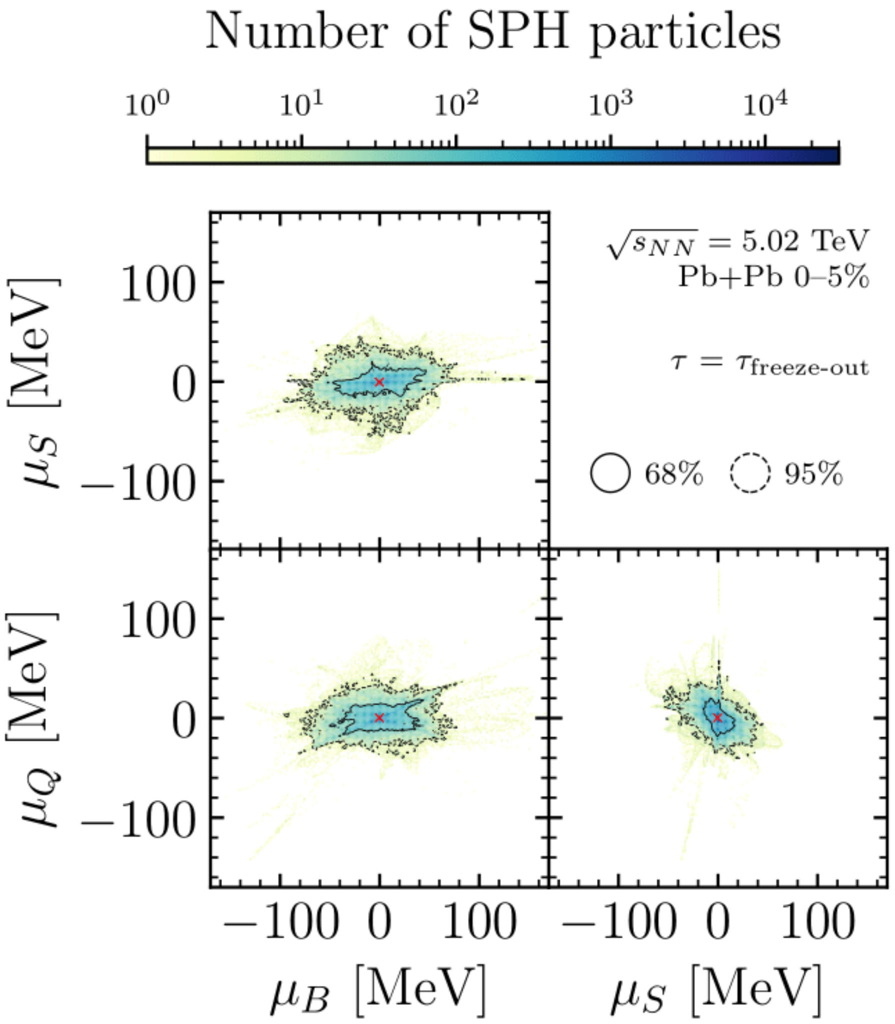
Density distributions (0-5)%



- Density scale set by initial charge fluctuations
- Prospect of constraining wide swath of QCD phase diagram using current (and future) HI experiments

Chemical potential correlations





- Correlations reflect charge combinations on (anti)quarks with different flavors
- Large initial and final spread in chemical potentials
- Averages consistent zero throughout evolution

nPOI Flow Observables: Definitions

nth-order flow vector: $V_n = v_n e^{in\Psi_n}$

All charged particles flow: $v_n\{2\} = \sqrt{\langle v_n^2 \rangle}$

1POI flow:

$$
\begin{aligned}
v_n^{\mathrm{1POI}}\{2\} &= \frac{\langle V_n \left(V_n'\right)^* \rangle}{v_n\{2\}} \\
&= \frac{\langle v_n v_n' \cos n \left(\Psi_n - \Psi_n'\right) \rangle}{v_n\{2\}}
\end{aligned}
$$

2POI flow:

$$
v_n^{\mathrm{2POI}}\{2\} = \langle V_n' \left(V_n'\right)^* \rangle = \sqrt{\langle \left(v_n''\right)^2 \rangle}
$$

Each prime ($'$) denotes additional POI