GPU Optimized fluid dynamics for heavy-ion collisions

Dennis Bazow
The Ohio State University

With:
U. Heinz and M. Strickland

Quark Matter 2017

THE OHIO STATE UNIVERSITY

JETSCAPE

JETSCAPE Mission

- Theoretical understanding of Jet measurements provide details into the internal structure of the QGP, but requires sophisticated numerical modeling
- Develop software package that:
  - Requires integration of numerical simulations of various (and competing) physical models for the initial state, dynamical medium evolution and the medium modification of jets
  - Use of advanced statistical methods in order to compare theory calculations with experimental data

See the JETSCAPE website: http://jetscape.wayne.edu/jetscape/Overview

JETSCAPE software framework

- Open source
- Will consist of numerical models to simulate:
  - Wave function of the incoming nuclei
  - Fluid dynamical evolution of the plasma
  - Transport and modification of jets
- Use Gaussian Process emulators based on Bayesian statistics to systematically determine model parameters that accurately predict the experimental observables

Will focus here in this talk.

Numerically expensive. Use graphic processing units (GPUs) to accelerate the process

GPU vs. CPU

- CPU is optimized for sequential serial code performance
- GPU has massively parallel architecture designed to optimize the performance of a simultaneous execution of thousands of tasks

Model		Intel Xeon Phi	GeForce GTX Titan Z
Processor cores		61	5760
Clock speed (MHz)	Core	1238	705
	Turbo/Boost	1333	876
Memory Configuration	Size (GB)	16	12.288
	Bandwidth (GB/s)	352	672
Processing power (GFLOPS)	Single precision	2416.58	8121.6
	Double precision	1208.29	2707.2

What algorithms benefit from parallelism?

Simplest case: Embarrassingly parallel

Addition of two vectors a and b of length N

For j = 1:N; c(j)=a(j)+b(j)

↓ Idealized case

Launch N concurrent threads.

Data is split up so that every addition operation is performed simultaneously

How does this help hydro codes?
Lets first review the algorithm.

Fluid dynamic stage of a heavy ion collision

Solve the conservation laws and relaxation equations

$$\partial_\mu T^{\mu\nu} = 0 \ ,$$

$$\tau_\Pi \dot{\Pi} + \Pi = -\zeta\theta + \mathcal{J} \ ,$$

$$\tau_\pi \dot{\pi}^{\langle\mu\nu\rangle} + \pi^{\mu\nu} = 2\eta\sigma^{\mu\nu} + \mathcal{J}^{\mu\nu} \ .$$

Use equations from Denicol, Niemi, Molnar, Rischke, PRD 85 114047

In flux conservative form using curvilinear coordinates:

$$\partial_\tau \mathbf{q} + \partial_i (v_i \mathbf{q}) = \mathbf{J} \qquad i \in \{x, y, \eta_s\} \qquad \mathbf{q} \equiv \begin{pmatrix}\hat{T} & \vec{\pi}\end{pmatrix}^T$$

Source terms

$$\hat{T} \equiv \begin{pmatrix} T^{\tau\tau} & T^{\tau x} & T^{\tau y} & T^{\tau\eta} \end{pmatrix}$$

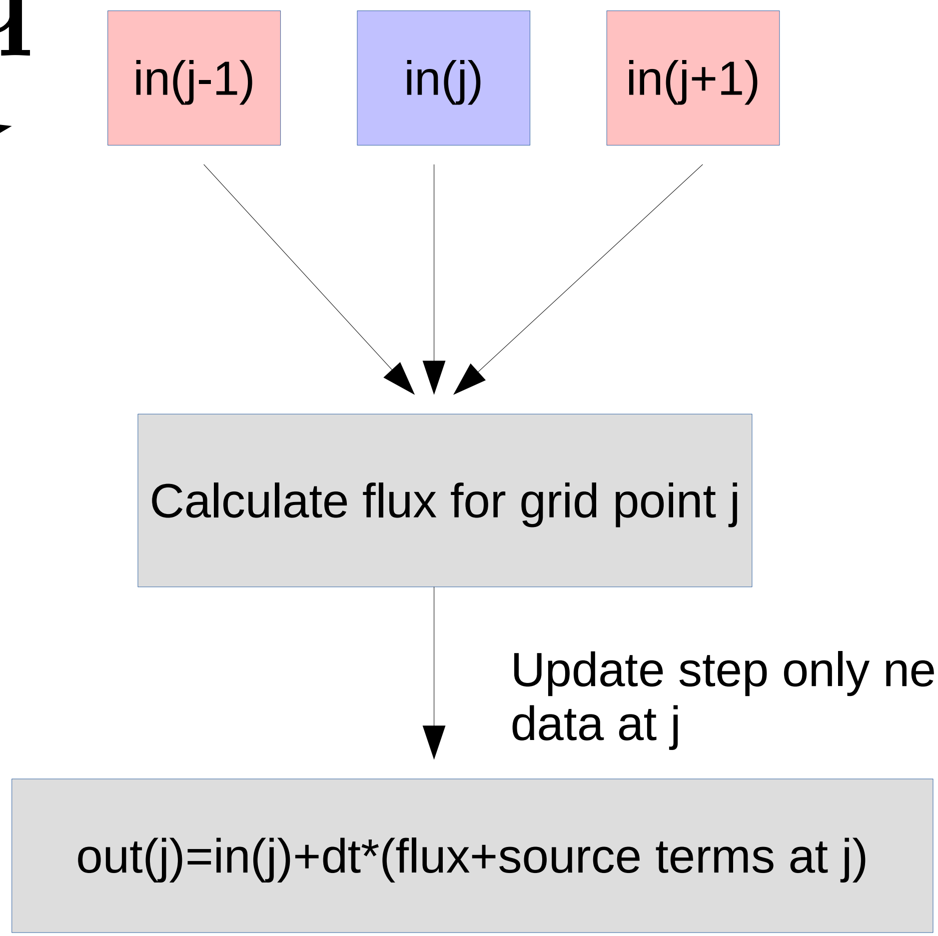
First row of energy-momentum tensor

$$\vec{\pi} \equiv \begin{pmatrix} \pi^{\tau\tau} & \pi^{\tau x} & \pi^{\tau y} & \pi^{\tau\eta} & \pi^{xx} & \pi^{xy} & \pi^{x\eta} & \pi^{yy} & \pi^{y\eta} & \pi^{\eta\eta} & \Pi \end{pmatrix}$$

Center differences + Euler step (1d)

$$\mathrm{in}, \mathrm{out} \in \mathbf{q}$$

Input data



Update step only needs to write to output data at j

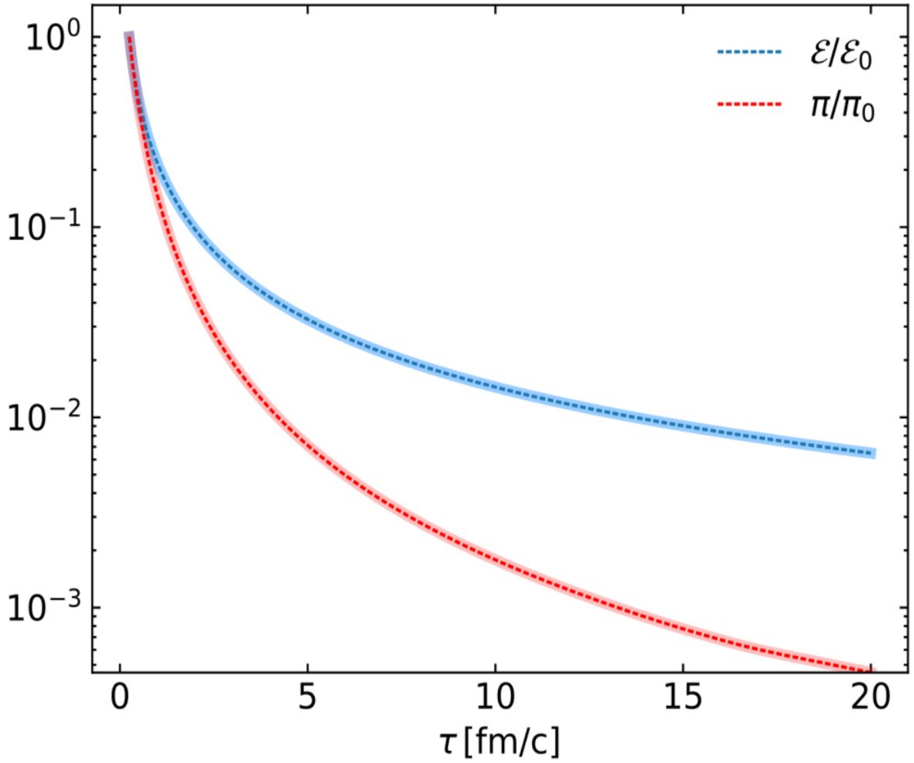
To update grid point j only need to read in neighboring cell data

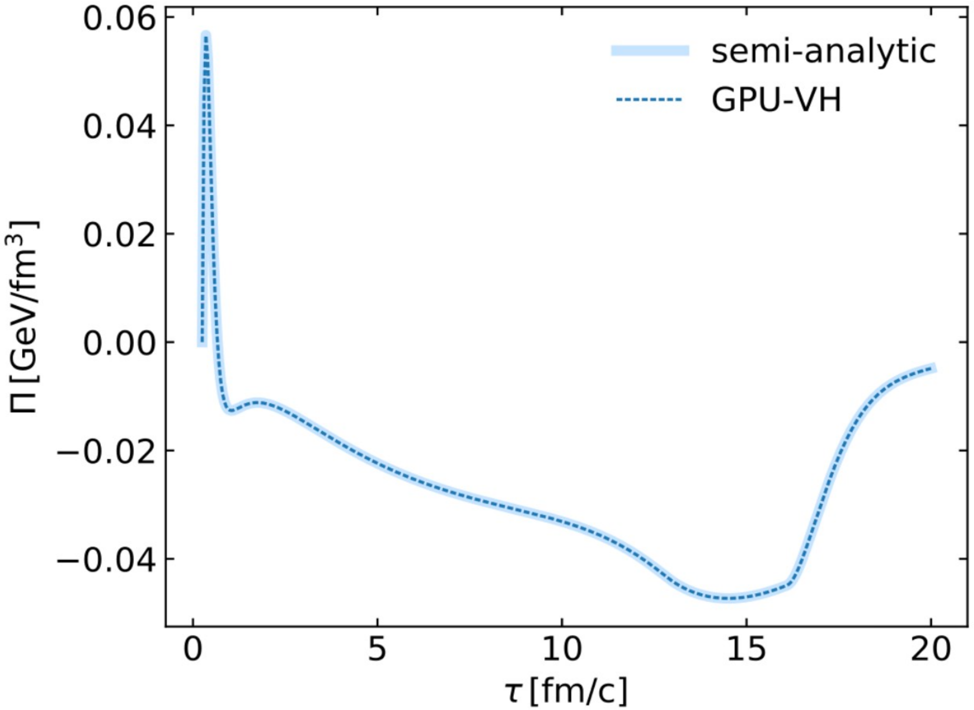
3+1d viscous hydro code GPU-VH

- Solve fluid dynamic equations using the Kurganov-Tadmor algorithm
  - Highly accurate for equations governing shock, discontinuities and large gradient phenomena
- On GPUs using CUDA C
- Fluctuating ICs, shear and bulk, QCD EoS
- Validate the code using (semi-)analytic tests

Bazow&Heinz&Strickland, arxiv:1608.06577

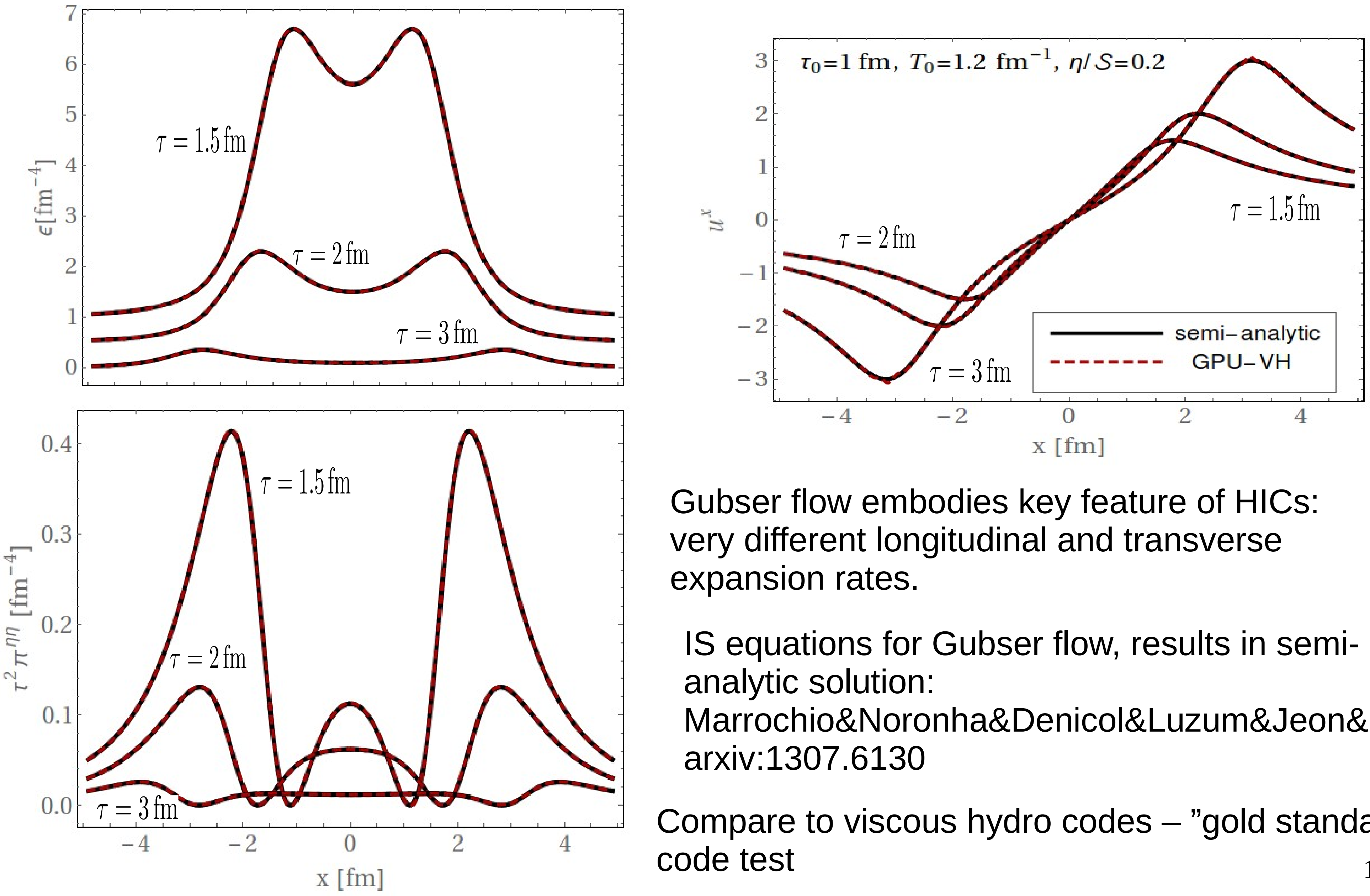
Nonconformal Bjorken flow test





Compare against semi-analytic solution which is a set of 3 coupled ODEs that can be solved to very high precision

Gubser test



Gubser flow embodies key feature of HICs: very different longitudinal and transverse expansion rates.

IS equations for Gubser flow, results in semi-analytic solution: Marrochio&Noronha&Denicol&Luzum&Jeon&Gale, arxiv:1307.6130

Compare to viscous hydro codes – ”gold standard” code test

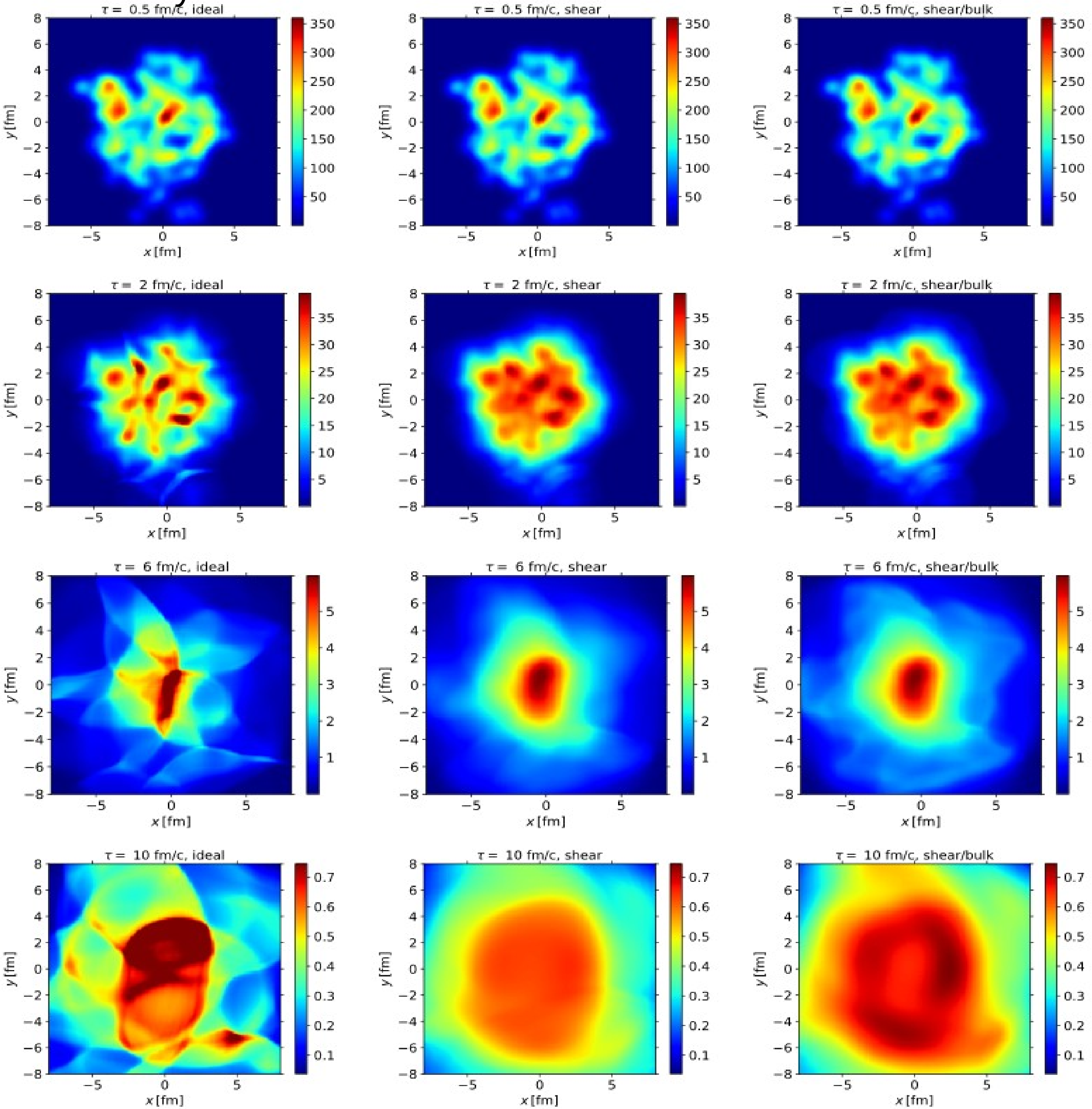
Ideal hydro | shear | shear+bulk

$\tau = 0.5\,\mathrm{fm/c}$

$\tau = 2\,\mathrm{fm/c}$

$\tau = 6\,\mathrm{fm/c}$

$\tau = 10\,\mathrm{fm/c}$



← MC Glauber initial condition

Hydro started at $\tau_0 = 0.5$ fm/c

Evolution of the energy density in the transverse plane for:

Ideal hydro (left),

Shear effects (middle),

Shear and bulk effects (right)

Performance benchmarks

Number of grid points	C/CPU (ms/step)	CUDA/GPU (ms/step)	Speedup
$128 \times 128 \times 32$	7145	63	112
$128 \times 128 \times 64$	13937	123	112
$128 \times 128 \times 128$	30717	244	125
$256 \times 256 \times 32$	25934	236	109
$256 \times 256 \times 64$	57387	472	121
$256 \times 256 \times 128$	129239	939	137
$256 \times 256 \times 256$	268448	1865	143

GPU: GeForce GTX 980 Ti (2816 cores)

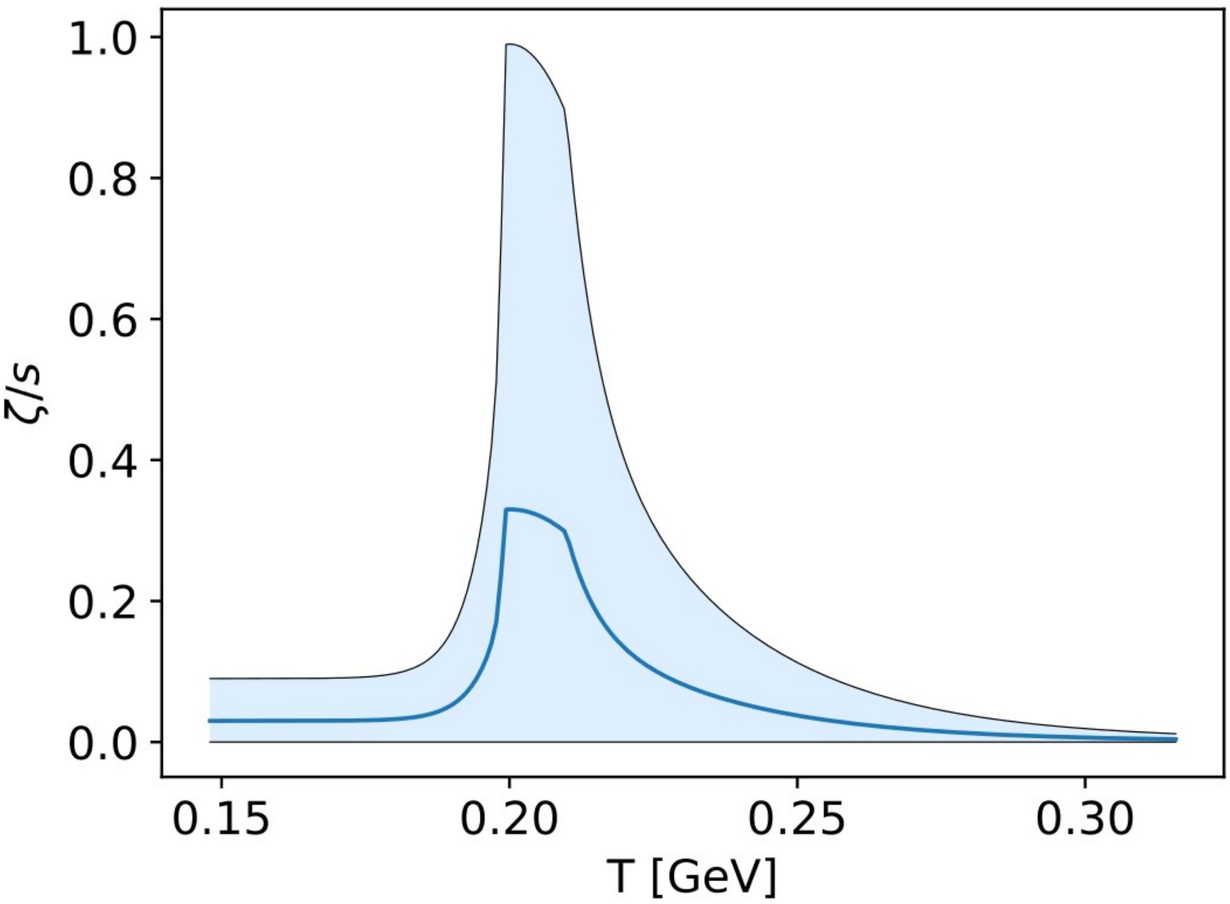
Host: 2.6 GHz Intel Xeon CPU E5-2697 v3

Conclusion/outlook

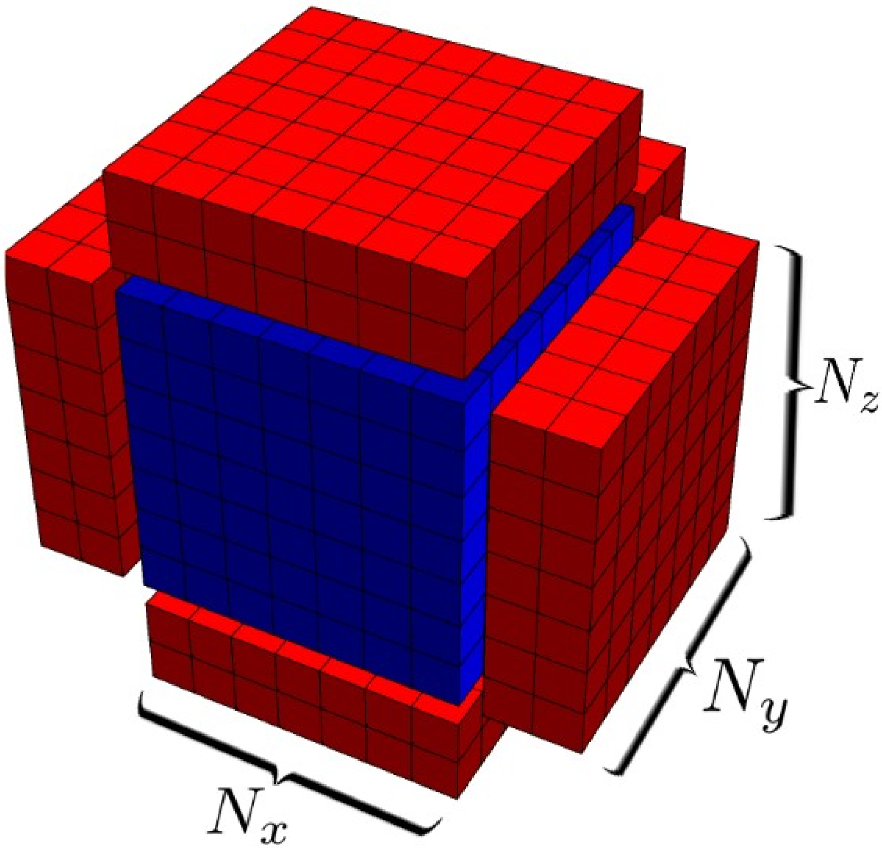
- Developed a highly efficient implementation of fluid dynamics on GPUs
- O(100) speedup over highly optimized serial CPU implementation of our code
  - Can simulate 256^3 grid size with 1000 time steps in 30 min.
  - Can simulate 1000 events on a small cluster of 4 GPUs in ~4 days.
- Necessary in order to perform a full 3+1d statistical parameter extraction
- Open source, will be downloadable from JETSCAPE git repo

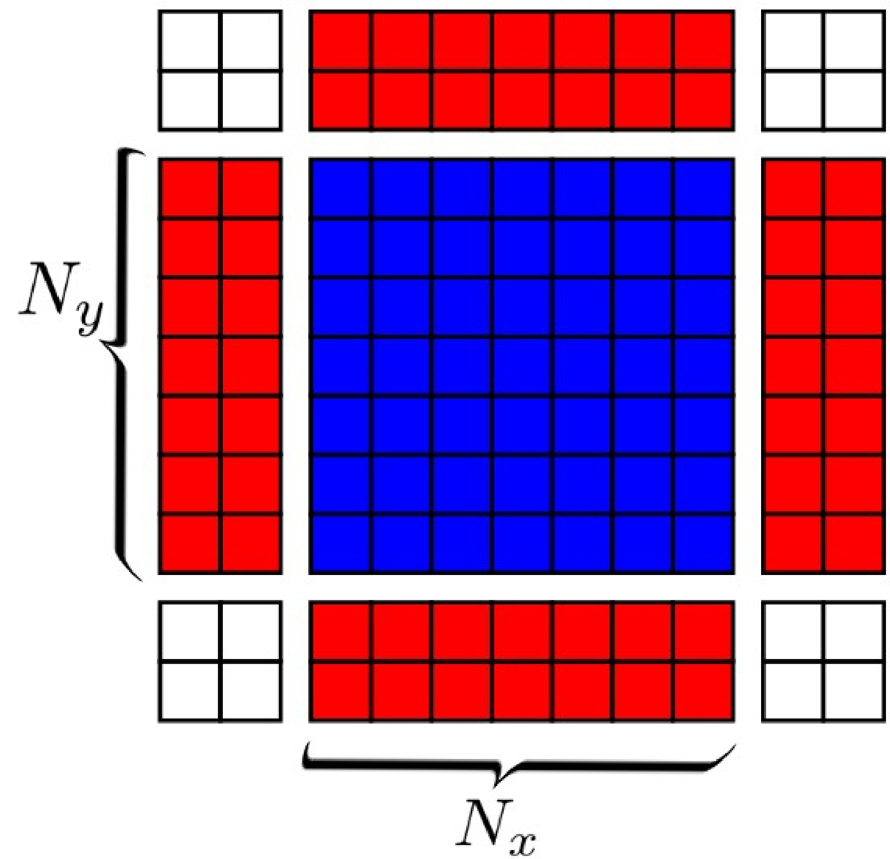
Backups

Bulk viscosity



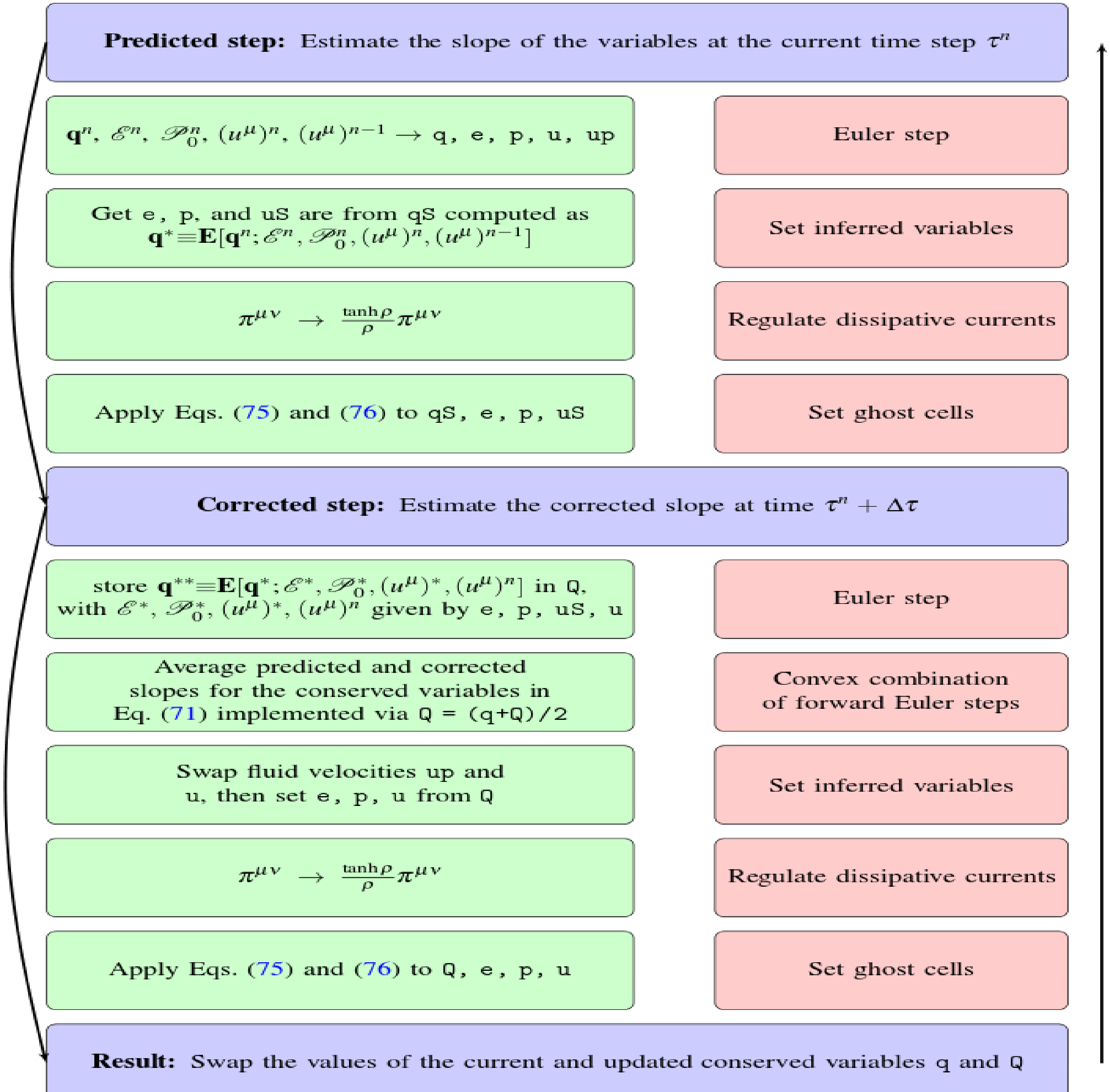
GPU-VH computational grid





Red cells are ghost cells that are added to allow the information in the cells on the edge of the blue region (physical volume of the simulation) to be evolved in time with the same set of instructions (using the fluxes at the boundary) as for interior cells

Program flow chart for the two-step RK algorithm



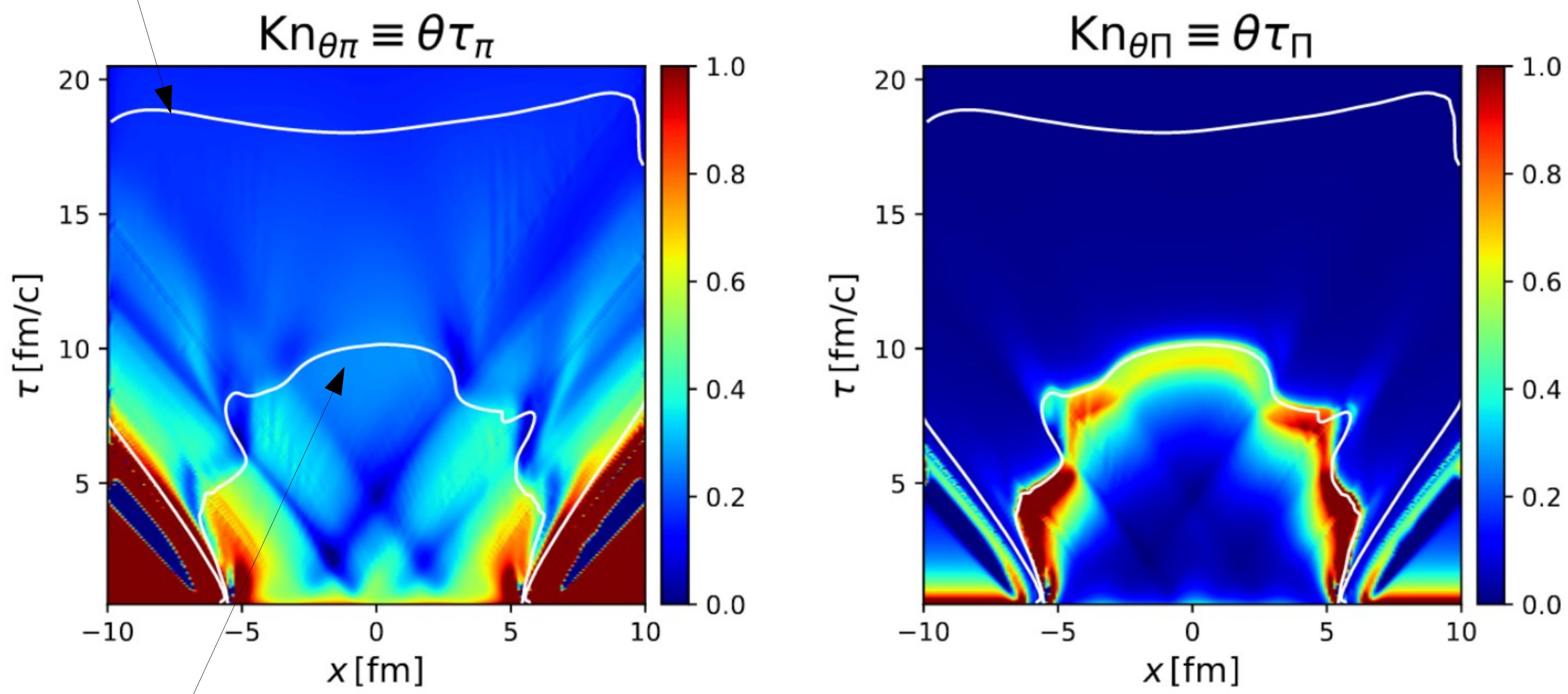
What is the validity of hydrodynamics in Heavy ion collisions?

$$\tau_\Pi \dot{\Pi} + \Pi = -\zeta\theta + \mathscr{J} + \mathscr{K} + \mathscr{R} ,$$
$$\tau_n \dot{n}^{\langle\mu\rangle} + n^\mu = \kappa I^\mu + \mathscr{J}^\mu + \mathscr{K}^\mu + \mathscr{R}^\mu ,$$
$$\tau_\pi \dot{\pi}^{\langle\mu\nu\rangle} + \pi^{\mu\nu} = 2\eta\sigma^{\mu\nu} + \mathscr{J}^{\mu\nu} + \mathscr{K}^{\mu\nu} + \mathscr{R}^{\mu\nu} .$$

- $\mathcal{J}, \mathcal{J}^\mu, \text{ and } \mathcal{J}^{\mu\nu}$ are $O(\mathrm{Kn}\, R^{-1})$
- $\mathcal{K}, \mathcal{K}^\mu, \text{ and } \mathcal{K}^{\mu\nu}$ are $O(\mathrm{Kn}^2)$
- $\mathcal{R}, \mathcal{R}^\mu, \text{ and } \mathcal{R}^{\mu\nu}$ are $O(R^{-2})$

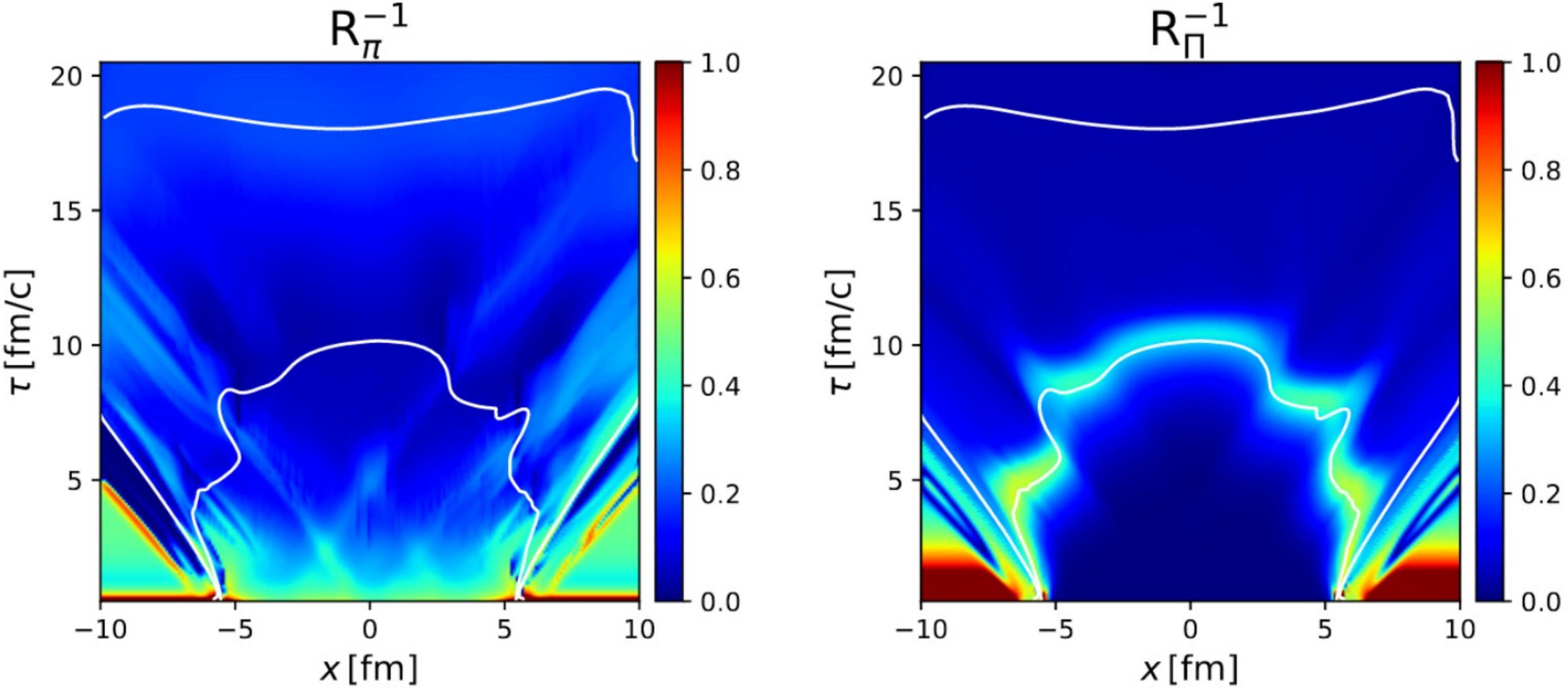
Freezeout surface at
$T_c = 155\,\mathrm{MeV}$

Knudsen number



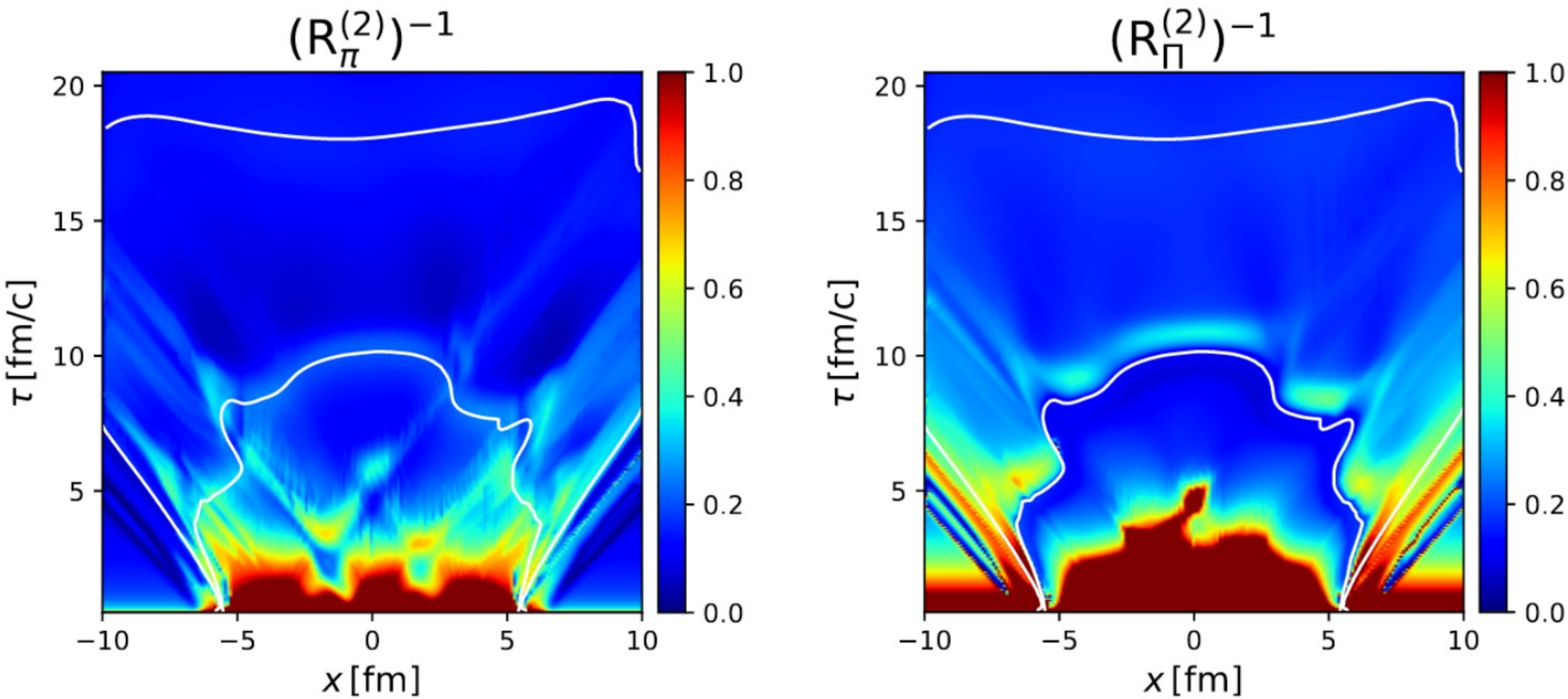
Isothermal contour at $T_c = 200\,\mathrm{MeV}$

Inverse Reynolds number



$$R_\pi^{-1} \equiv \frac{\sqrt{\pi^{\mu\nu}\pi_{\mu\nu}}}{\mathscr{P}_0}, \quad R_\Pi^{-1} \equiv \frac{|\Pi|}{\mathscr{P}_0}.$$

Second order inverse Reynolds number

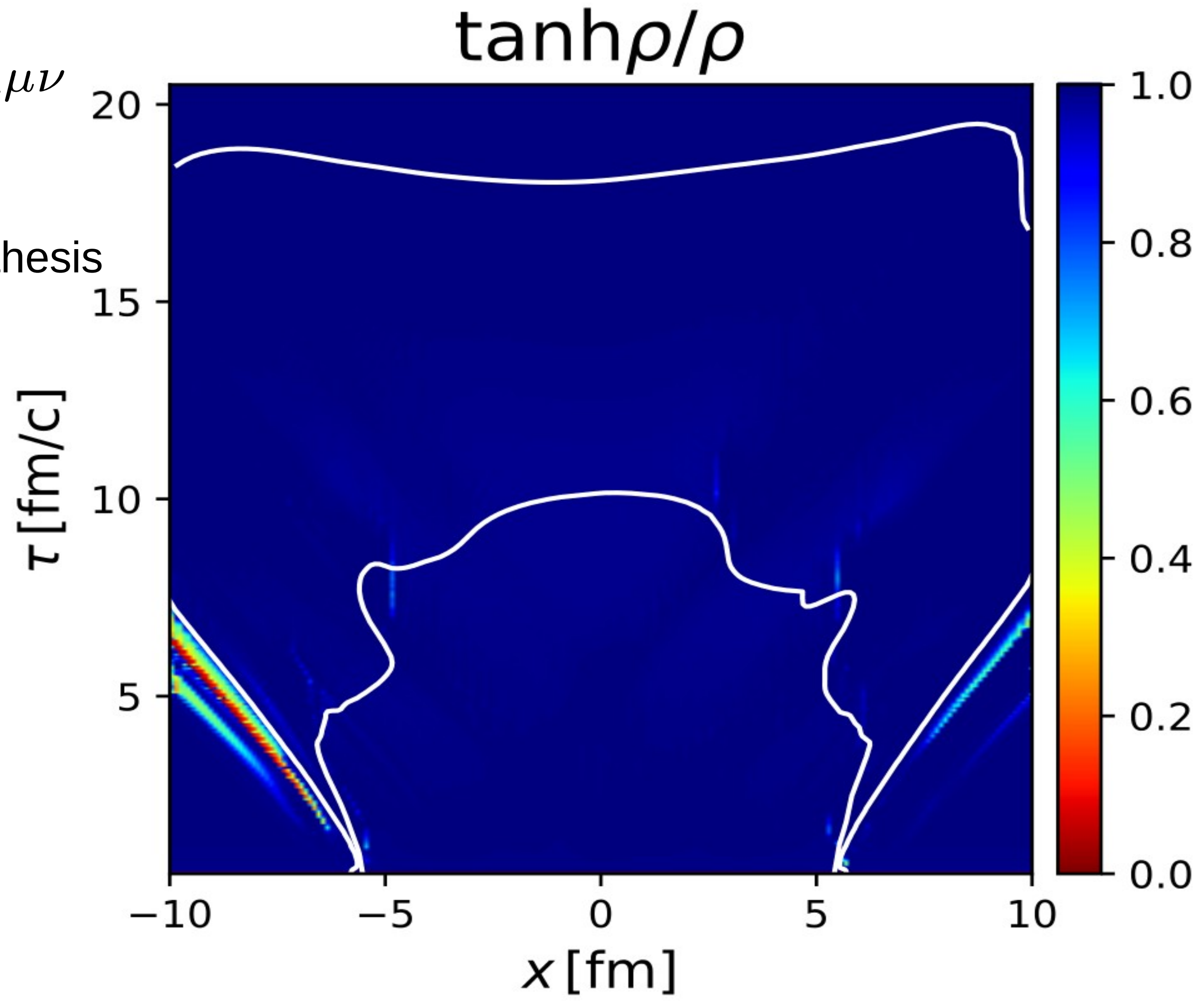


$$(\mathrm{R}_{\pi}^{(2)})^{-1} \equiv \frac{\sqrt{\mathscr{J}^{\mu\nu}\mathscr{J}_{\mu\nu}}}{2\eta\sqrt{\sigma^{\mu\nu}\sigma_{\mu\nu}}}, \quad (\mathrm{R}_{\Pi}^{(2)})^{-1} \equiv \frac{|\mathscr{J}|}{\zeta|\theta|}.$$

Shear stress regularization

$$\pi^{\mu\nu} \rightarrow \frac{\tanh\rho}{\rho}\pi^{\mu\nu}$$

See Chun Shen's PhD thesis



Graphic card specs

		Clock speeds (MHz)		Memory Configuration		Processing power (GFLOPS)	
Model	Processor Cores	Core	Memory	Size (GB)	Bandwidth (GB/s)	Single precision	Double precision
GeForce GTX 560M	192	775	2500	3.076	60	595.2	N/A
GeForce GTX 980 Ti	2816	1000	7012	6.144	336	5632	176
Tesla K20M	2496	706	2600	5.120	208	3524	1175