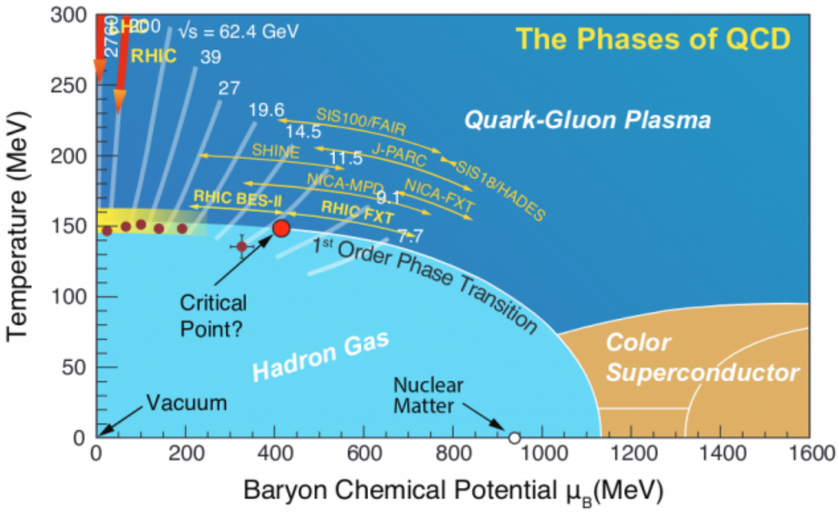
Efficient solver for relativistic hydrodynamics with implicit Runge-Kutta method

Nathan Touroux[1]
Marlene Nahrgang, Masakiyo Kitazawa, Koichi Murase

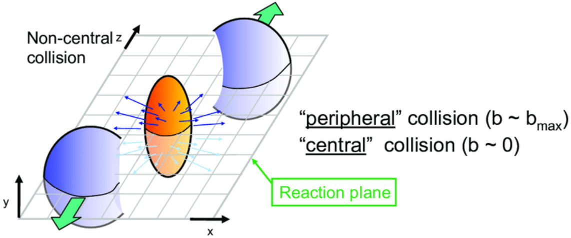
October 31, 2024

[1]Subatech, Osaka University, YITP

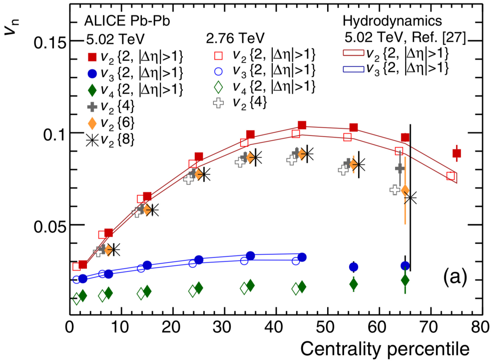
QCD phase diagram



Why hydrodynamics?



Almond shape → elliptic flow



J. Adam et al, PRL 116, 132302 (2016)

Relativistic hydrodynamics

Energy–momentum tensor conservation:

$$\partial_{;\,\mu} T^{\mu\nu} = \partial_\mu T^{\mu\nu} + \Gamma^{\mu}{}_{\alpha\mu} T^{\alpha\nu} + \Gamma^{\nu}{}_{\alpha\mu} T^{\mu\alpha} = 0$$

$$T^{\mu\nu} = \epsilon u^\mu u^\nu - (P + \Pi)\Delta^{\mu\nu} + \pi^{\mu\nu}$$

$$\Delta_{\mu\nu} = g_{\mu\nu} - u_\mu u_\nu, \qquad u^\mu u_\mu = 1,$$

Israel-Stewart equations:

$$u^\lambda \partial_{;\lambda}\Pi = -\frac{\Pi - \Pi_{NS}}{\tau_\Pi} - \frac{4}{3}\Pi \partial_{;\lambda} u^\lambda$$

$$\langle u^\lambda \partial_{;\lambda}\pi^{\mu\nu} \rangle = -\frac{\pi - \pi^{\mu\nu}_{NS}}{\tau_\pi} - \frac{4}{3}\pi \partial_{;\lambda} u^\lambda$$

$$\Pi_{NS} = -\zeta \partial_{;\lambda} u^\lambda,$$

$$\pi^{\mu\nu}_{NS} = \eta \left(\Delta^{\mu\lambda}\partial_{;\lambda} u^\nu + \Delta^{\nu\lambda}\partial_{;\lambda} u^\nu \right) - \frac{2}{3}\eta \Delta^{\mu\nu}\partial_{;\lambda} u^\lambda,$$

$$\tau_\pi = \tau_\Pi = \frac{5\eta}{\epsilon + P}$$

Relativistic Hydrodynamics

Rewriting the equations:

$$\partial_\tau \left(\tau T^{\tau\nu}_{\mathrm{id}}\right) = -\partial_x \left(\tau T^{x\nu}\right) - \partial_y \left(\tau T^{y\nu}\right) - \partial_\eta \left(T^{\eta\nu}\right) + A(\partial_\tau u^\nu, \partial_\tau \Pi, \partial_\tau \pi^{\tau\nu}) \quad (1)$$

$$\partial_\tau \left(u^\tau \pi^{\mu\nu}\right) = -\partial_x \left(u^x \pi^{\mu\nu}\right) - \partial_y \left(u^y \pi^{\mu\nu}\right) - \partial_\eta \left(u^\eta \pi^{\mu\nu}\right) + B(\partial_\tau u^\nu, \partial_i u^\nu) \quad (2)$$

$$\partial_\tau \left(u^\tau \Pi\right) = -\partial_x \left(u^x \Pi\right) - \partial_y \left(u^y \Pi\right) - \partial_\eta \left(u^\eta \Pi\right) + C(\partial_\tau u^\nu, \partial_i u^\nu) \quad (3)$$

where

$$A(\partial_\tau u^\nu, \partial_\tau \Pi, \partial_\tau \pi^{\tau\nu}) = I^\nu_T - \partial_\tau \left(\tau \left[-\Pi \Delta^{\tau\nu} + \pi^{\tau\nu}\right]\right), \quad (4)$$

$$B(\partial_\tau u^\nu, \partial_i u^\nu) = -\frac{\tilde{\pi}^{\mu\nu} - \tilde{\pi}^{\mu\nu}_{\mathrm{NS}}}{\tau_\pi} - \frac{1}{3}\tilde{\pi}^{\mu\nu}\left(\tilde{\partial}_\lambda \tilde{u}^\lambda - 4\frac{\tilde{u}^\tau}{\tau}\right) \quad (5)$$

$$- \left(\tilde{u}^\nu \tilde{\pi}^{\mu\beta} + \tilde{u}^\mu \tilde{\pi}^{\nu\beta}\right) \tilde{u}^\lambda \tilde{\partial}_{;\lambda} \tilde{u}_\beta + \frac{\tilde{u}^\eta}{\tau} I^{\mu\nu}_\pi, \quad (6)$$

$$C(\partial_\tau u^\nu, \partial_i u^\nu) = -\frac{\tilde{\Pi} - \tilde{\Pi}_{\mathrm{NS}}}{\tau_\Pi} - \frac{1}{3}\tilde{\Pi}\left(\tilde{\partial}_\lambda \tilde{u}^\lambda - 4\frac{\tilde{u}^\tau}{\tau}\right), \quad (7)$$

Numerical scheme

Hydro equations with the dynamical variables $y = (\tau T^{\tau\nu}_{\mathrm{id}}, u^\tau \pi^{\mu\nu}, u^\tau \Pi)$:

$$\partial_t y = f(t, y)$$

Kurganov-Tadmor space discretization: $f(t, y) = f_{KT}(t, y) + O(\Delta x^2)$

- **Independent of time discretization**
- Numerical diffusion $\propto \Delta x^2$
- Flux limiter $\rightarrow$ avoid numerical oscillations

Runge-Kutta time discretization

$$y(t + \Delta t) = y(t) + \Delta t \sum_{j=1}^{n} b_j k_j$$

$$k_i = f(t, y(t) + \Delta t \sum_{j=1}^{n} a_{ij} k_j) \qquad \text{“}\vec{K} = \vec{F}(\vec{K})\text{”}$$

Extending Runge-Kutta

Considering that

$$k_i = \left(\partial_\tau \left(\tau T_{\mathrm{id}}^{\tau\nu}\right), \partial_\tau \left(u^\tau \pi^{\mu\nu}\right), \partial_\tau \left(u^\tau \Pi\right)\right), \tag{8}$$

we can hope to express the right-hand-side time derivatives by extending Runge-Kutta as either:

implicit

$$k_i = f(t, y(t) + \Delta t \sum_{j=1}^{n} a_{ij} k_j, k_i)$$

explicit

$$k_1 = f(t, y(t) + \Delta t, k_n(t - \Delta t))$$
$$k_2 = f(t, y(t) + \Delta t a_{21} k_1, k_1)$$
$$k_3 = f(t, y(t) + \Delta t(a_{31} k_1 + a_{32} k_2), k_2)$$
$$\ldots$$

Right-hand-side time derivatives partial solving

By considering $\partial_\tau \left(\tau T^{\tau\nu}_{\mathrm{id}}\right)$, $\partial_\tau \left(u^\tau \pi^{\mu\nu}\right)$ and $\partial_\tau \left(u^\tau \Pi\right)$ known, we can express $\partial_\tau u^\nu$, $\partial_\tau \Pi$ and $\partial_\tau \pi^{\tau\nu}$[2]:

$$\partial_\tau \pi^{\tau\nu} = \frac{\partial_\tau \left(u^\tau \pi^{\tau\nu}\right) - \pi^{\tau\nu}\partial_\tau u^\tau}{u^\tau} \tag{9}$$

$$\partial_\tau \Pi = \frac{\partial_\tau \left(u^\tau \Pi\right) - \Pi\partial_\tau u^\tau}{u^\tau} \tag{10}$$

$$\partial_\tau u^\tau = -\frac{u^x\partial_\tau u^x + u^y\partial_\tau u^y + u^\eta\partial_\tau u^\eta}{u^\tau} \tag{11}$$

$$\begin{pmatrix}\partial_\tau \epsilon & \partial_\tau u^x & \partial_\tau u^y & \partial_\tau u^\eta\end{pmatrix}^{\mathrm{T}} = \mathbb{M}^{-1}\partial_\tau T^{\tau\nu}_{\mathrm{id}} \tag{12}$$

$$\partial_\tau T^{\tau\nu}_{\mathrm{id}} = \frac{\partial_\tau \left(\tau T^{\tau\nu}_{\mathrm{id}}\right) - T^{\tau\nu}_{\mathrm{id}}}{\tau} \tag{13}$$

where the 4 by 4 matrix $\mathbb{M} = \left[\frac{\partial \left(\partial_\tau T^{\tau\nu}_{\mathrm{id}}\right)}{\partial \begin{pmatrix}\partial_\tau \epsilon & \partial_\tau u^x & \partial_\tau u^y & \partial_\tau u^\eta\end{pmatrix}}\right]$ can be inverted analytically.

[2] Inspired by rfh code (Koichi Murase)

Choice of Runge-Kutta coefficients

Accuracy order:

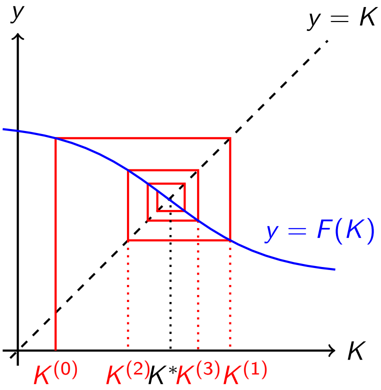
$$||y^*(t) - y(t)|| < C\Delta t^p$$

Second order choices:

	Heun	Gauss-Legendre 1 (GL1)
Type	Explicit	Implicit
Stage S	2	1
Order p	2	2
$\begin{array}{c\|c} c_n & a_{nm} \\ \hline & b_m \end{array}$	$\begin{array}{c\|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 0.5 & 0.5 \end{array}$	$\begin{array}{c\|c} 0.5 & 0.5 \\ \hline & 1 \end{array}$

Implicit solver

Solve implicit equation "$\vec{K} = \vec{F}(\vec{K})$" by the fixed-point solver



$$\vec{K}^{(l+1)} = \vec{F}(\vec{K}^{(l)})$$

$$\vec{K}^{(0)} = \vec{0} \text{ or last time step solution}$$

Solve iteratively

$$\vec{K}^{(0)} \to \vec{K}^{(1)} \to \vec{K}^{(2)} \to \ldots \to \vec{K}^{(l+1)}$$

Local optimization

1. Update all cells once to obtain $\vec{K}^{(1)}$
2. check the convergence in every cell $[\bullet]_j$:

$$\left\|[\vec{F}(\vec{K}^{(l)})]_j - [\vec{K}^{(l)}]_j\right\| < e\frac{\langle T^{\tau\tau}\rangle}{\Delta\tau}\left(\frac{\Delta\tau}{\Delta x}\right)^{(p+1)}$$

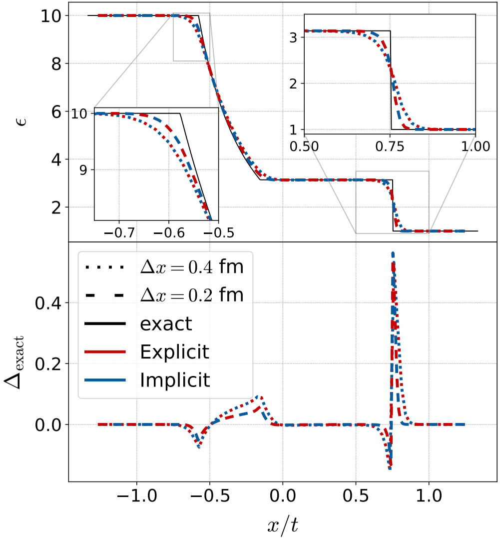
3. To obtain $\vec{K}^{(l+1)}$ $(l+1 \geq 2)$, only update a cell if itself or any surrounding cells does not satisfy the threshold
4. Repeat (2) and (3) until all cells satisfy the threshold

This dramatically reduces the computation cost.

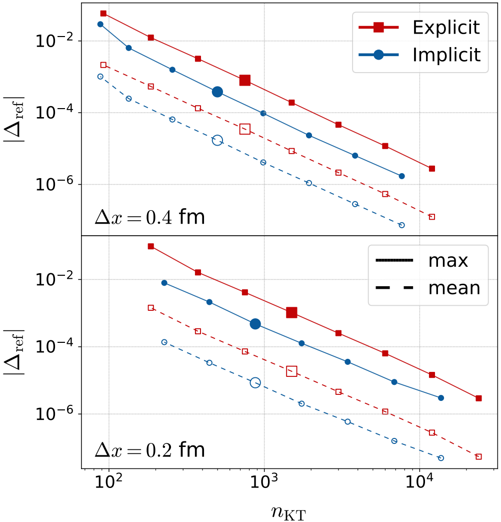
Disclaimer: This partial update breaks conservation and leads to inconsistencies, but this is controlled within the error threshold.

1D ideal Riemann problem

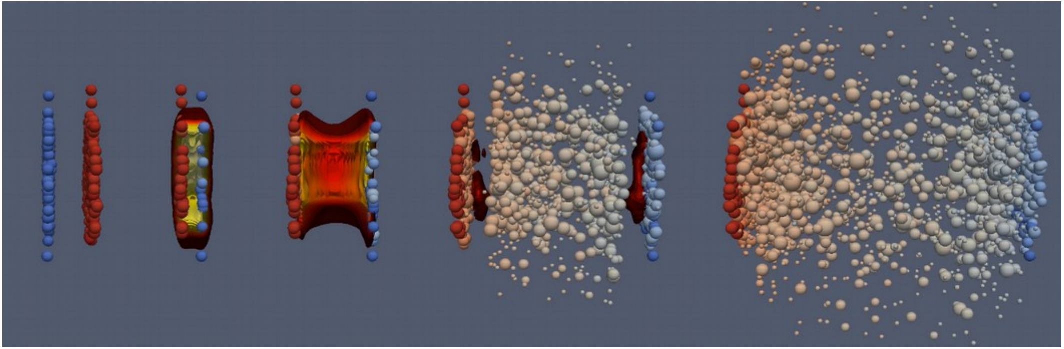
Vanishing Δt comparison



Error-cost comparison



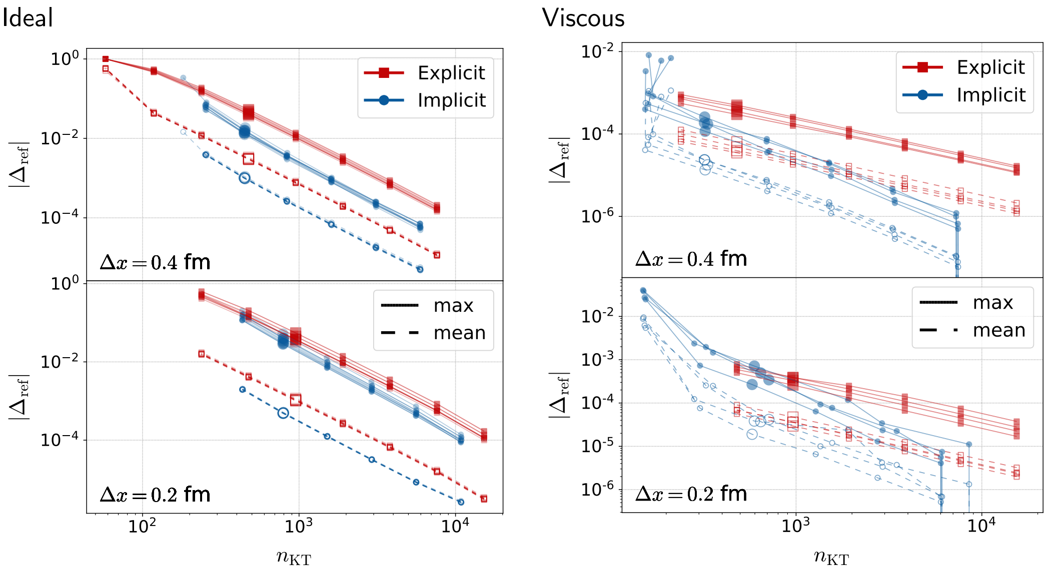
Heavy ion collisions evolution



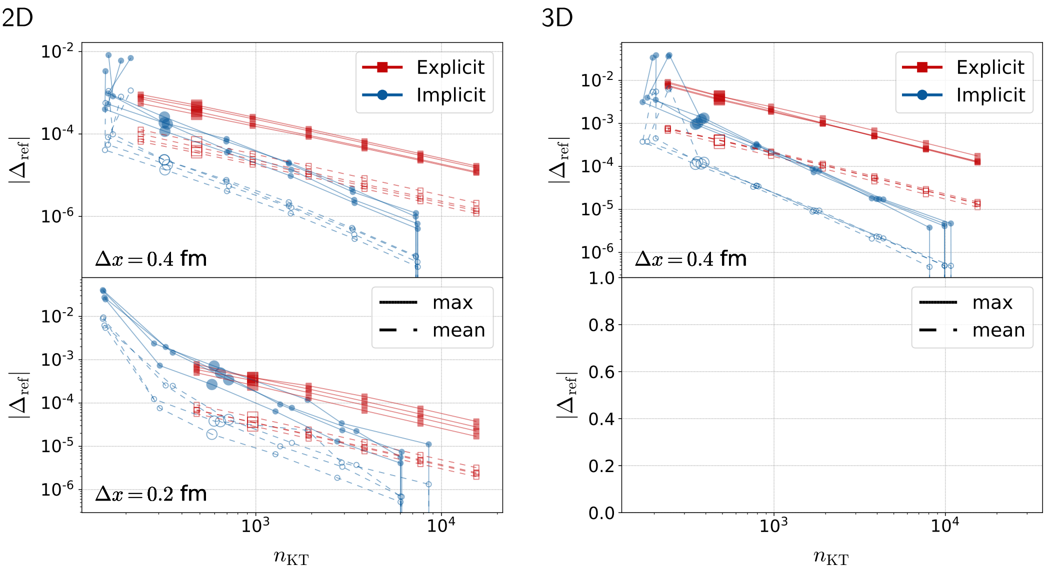
Collision	thermalization Pre-hydrodynamics	QGP Hydrodynamics	Particlization Cooper-Frye	Decay/free-streaming Cascade/Kinetic
Trento	skipped	ImplHydro	frzout	UrQMD

Parameters from: PRC 101, 024911 (2020)

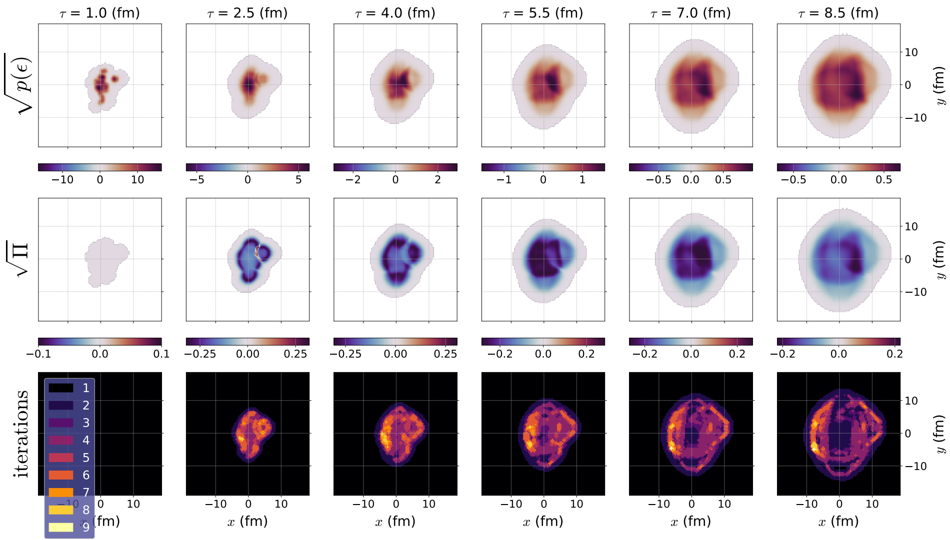
Trento initial conditions: 2D ideal vs viscous



Trento initial conditions: viscous 2D vs 3D

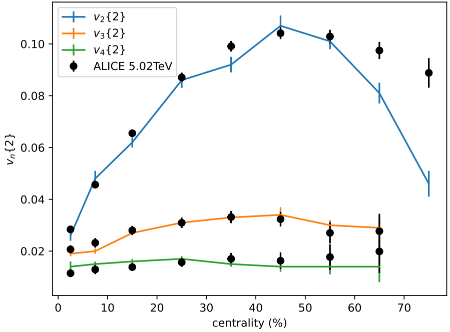


Trento initial conditions: viscous 3D

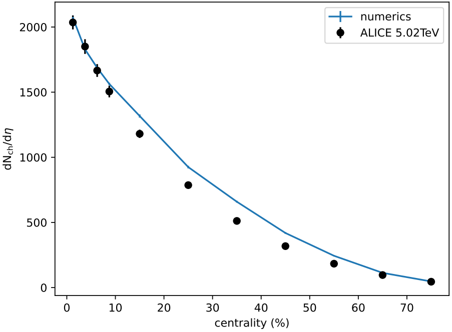


Physical results

Elliptic flow



Particle spectra



Conclusion

Achievement

- Implicit method can be more efficient and accurate than explicit
- First true 2$^{\text{nd}}$ order time integrator for viscous hydrodynamics
- Detects stiffness through convergence of the fixed-point iterator

Outlook

- Addapt Δt depending on convergence for unconditional stability
- Including other charges (baryon number, electric charge, ...)
- Including fluctuations

ImplHydro:

- arXiv:2306.12696
- Open source: `https://github.com/xayon40-12/ImplHydro.git`

Thank you for your attention

Enforcing positivity of the effective pressure

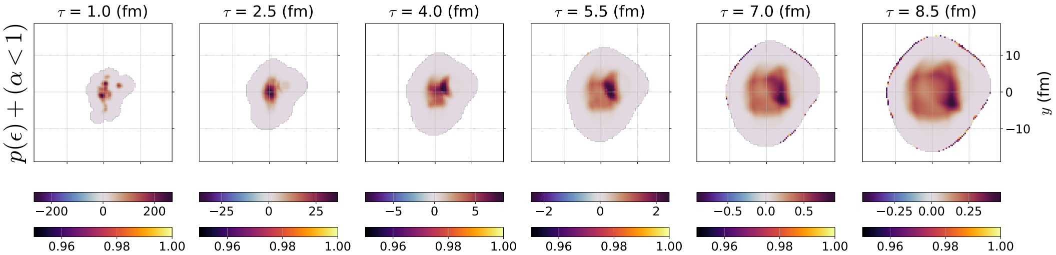
We need to garanty that

$$\epsilon + P(\epsilon) + \Pi + \lambda_{\min} \geq 0, \tag{14}$$

where $\lambda_{\min} = \min_\mu\{\lambda^\mu\}$ is the smallest of the eigenvalues λ^μ of the shear tensor $\pi^{\mu\nu}$. If it is not the case, we rescale the bulk pressure and shear tensor as

$$\pi^{\mu\nu} \leftarrow \alpha\pi^{\mu\nu} \qquad \alpha = -\frac{\epsilon + P(\epsilon)}{\Pi + \lambda_{\min}}$$

$$\Pi \leftarrow \alpha\Pi$$



Right hand side space derivatives

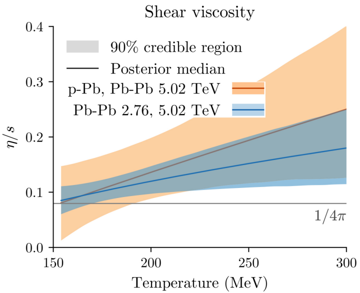
In order to approximate the space derivatives $\partial_i u^\mu$ in A, B and C, we first reconstruct the values of the fields at the boundaries of cells using the cubic interpolation (here in the x direction by considering that j indexes the x coordinate):

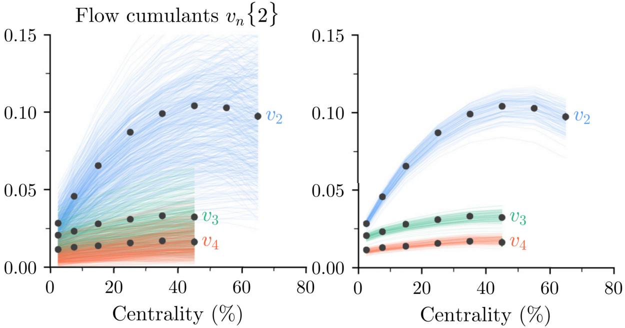
$$[u^\mu]_{j\pm1/2} \approx -\frac{1}{16}[u^\mu]_{j\pm1/2-3/2} + \frac{9}{16}[u^\mu]_{j\pm1/2-1/2} + \frac{9}{16}[u^\mu]_{j\pm1/2+1/2} - \frac{1}{16}[u^\mu]_{j\pm1/2+3/2} \quad (15)$$

and then perform the finite difference to approximate

$$\partial_x u^\mu \approx \frac{[u^\mu]_{j+1/2} - [u^\mu]_{j-1/2}}{\Delta x}. \quad (16)$$

Simulation framework





PRC 101, 024911 (2020)

KT 2^{nd} order algorithm

$$KT[f,\rho(f),y,\Delta x]_i = \frac{H_{i+1/2} - H_{i-1/2}}{\Delta x}$$

$$H_{i+1/2} = \frac{f(y^+_{j+1/2}) + f(y^-_{j+1/2})}{2} + \frac{a_{i+1/2}}{2}\left(y^+_{j+1/2} - y^-_{i+1/2}\right)$$

$$a_{i+1/2} = max\left\{\rho\left(\frac{\partial f}{\partial y}\left(y^+_{i+1/2}\right)\right), \rho\left(\frac{\partial f}{\partial y}\left(y^-_{i+1/2}\right)\right)\right\}$$

$$y^+_{i+1/2} = y_{i+1} - \frac{\Delta x}{2}(\partial_x y)_{i+1}$$

$$y^-_{i+1/2} = y_i + \frac{\Delta x}{2}(\partial_x y)_i$$

$$(\partial_x y)_i = \text{minmod}\left(\theta\frac{y_i - y_{i-1}}{\Delta x}, \frac{y_{i+1} - y_{i-1}}{2\Delta x}, \theta\frac{y_{i+1} - y_i}{\Delta x}\right)$$

$$1 \leq \theta \leq 2$$

needs $y_{i-2}, y_{i-1}, y_i, y_{i+1}, y_{i+2}$.