VERONA: A GPU-accelerated special relativistic hydrodynamics (SRHD) code for astrophysical applications

Piotr Płonka[1], **Mateusz Kapusta**[2]

prof. dr hab. Agnieszka Janiuk[1]

[1]Center for Theoretical Physics PAS

[2]Astronomical Observatory of the University of Warsaw

Tuesday, April 14, 2026

Motivation for developing VERONA

- The project started one year ago by two Master's students.
- The idea was to develop our own **GPU-accelerated** special relativistic hydrodynamics (SRHD) code written in the `Julia` programming language, using `CUDA.jl` and `KernelAbstractions.jl`, with **CUDA-aware MPI** communication.
- SRHD codes are widely used to model astrophysical phenomena, such as **jet breakout**, which we aim to simulate at ultra-high resolution.
- The code architecture is designed to enable **future extensions** to SRMHD and GRMHD.

Special relativistic hydrodynamics

Special relativistic hydrodynamics (SRHD) is based on two local conservation laws, supplemented by an equation of state:

- Covariant conservation of the rest-mass current:

$$\nabla_\mu J^\mu = \nabla_\mu(\rho u^\mu) = 0 \tag{1}$$

- Covariant conservation of energy and momentum:

$$\nabla_\mu T^{\mu\nu} = 0, \qquad T^{\mu\nu} = \rho h\, u^\mu u^\nu + p\, \eta^{\mu\nu} \tag{2}$$

  where $h = 1 + \epsilon + p/\rho$ is the specific enthalpy.

- Ideal-gas equation of state:

$$p = (\Gamma - 1)\rho\epsilon \tag{3}$$

Hyperbolic nature of SRHD equations

We use the **Valencia-type** conservative formulation of the SRHD equations, in which the system can be written as a hyperbolic conservation law:

$$\partial_t \boldsymbol{F}^0(\boldsymbol{P}) + \partial_i \boldsymbol{F}^i(\boldsymbol{P}) = 0, \tag{4}$$

where $\boldsymbol{F}^0$ is the vector of conserved variables, $\boldsymbol{F}^i$ are the flux vectors in the spatial directions, and $\boldsymbol{P}$ denotes the primitive variables. These are defined as

$$\boldsymbol{F}^0 = \begin{pmatrix} D \\ S_j \\ \tau \end{pmatrix}, \qquad \boldsymbol{F}^i = \begin{pmatrix} Dv^i \\ S_j v^i + p\,\delta^i{}_j \\ \tau v^i + p v^i \end{pmatrix}, \qquad \boldsymbol{P} = \begin{pmatrix} \rho \\ v^1 \\ v^2 \\ v^3 \\ \epsilon \end{pmatrix}. \tag{5}$$

Here, the primitive variables are: ρ – rest-mass density, v^i – fluid three-velocity, and ϵ – specific internal energy.

Numerical Method implemented in VERONA

- VERONA is **conservative** and **shock-capturing** code, using the HLLE method for flux calculations:

$$\boldsymbol{F}^i = \frac{c_{\min}\,\boldsymbol{F}_r^i\left(\boldsymbol{P}_r\right) + c_{\max}\,\boldsymbol{F}_l^i\left(\boldsymbol{P}_l\right) - c_{\max}c_{\min}\left(\boldsymbol{F}_r^0\left(\boldsymbol{P}_r\right) - \boldsymbol{F}_l^0\left(\boldsymbol{P}_l\right)\right)}{c_{\max} + c_{\min}} \tag{6}$$

- For spatial reconstruction, we use the **WENO-Z** scheme, with fallback to **MINMOD** reconstruction and, if needed, to a **first-order piecewise-constant** state.
- The conserved-to-primitive conversion is performed using a **1D Brent root-finding method** (combining inverse quadratic interpolation, secant method, and bisection) to solve the following equation:

$$f(z) \equiv \tau + D - z + \frac{\gamma - 1}{\gamma}\left[z\left(1 - \frac{S^2}{z^2}\right) - D\sqrt{1 - \frac{S^2}{z^2}}\right] = 0 \tag{7}$$

Kernels configuration of VERONA

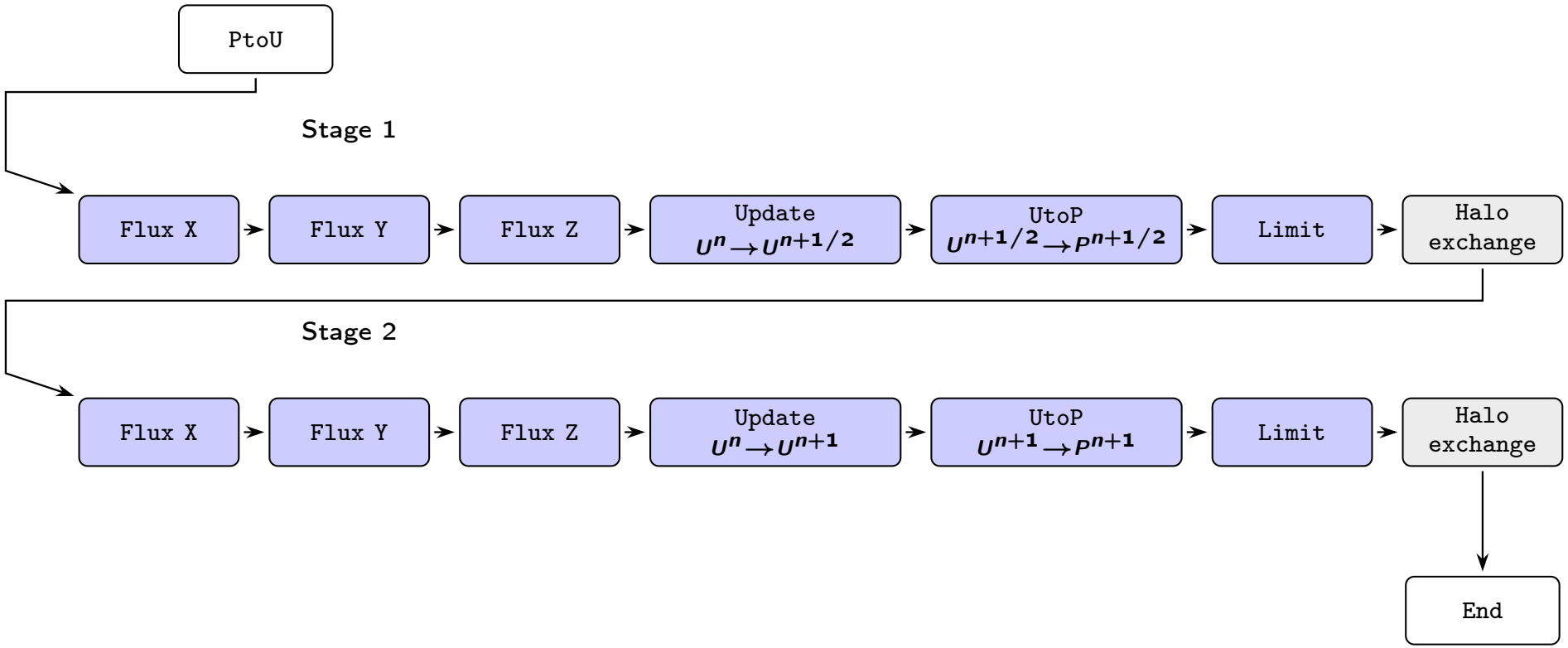


Figure 1: Two-stage RK2 (midpoint method) kernel pipeline in VERONA.

GPU configuration of VERONA

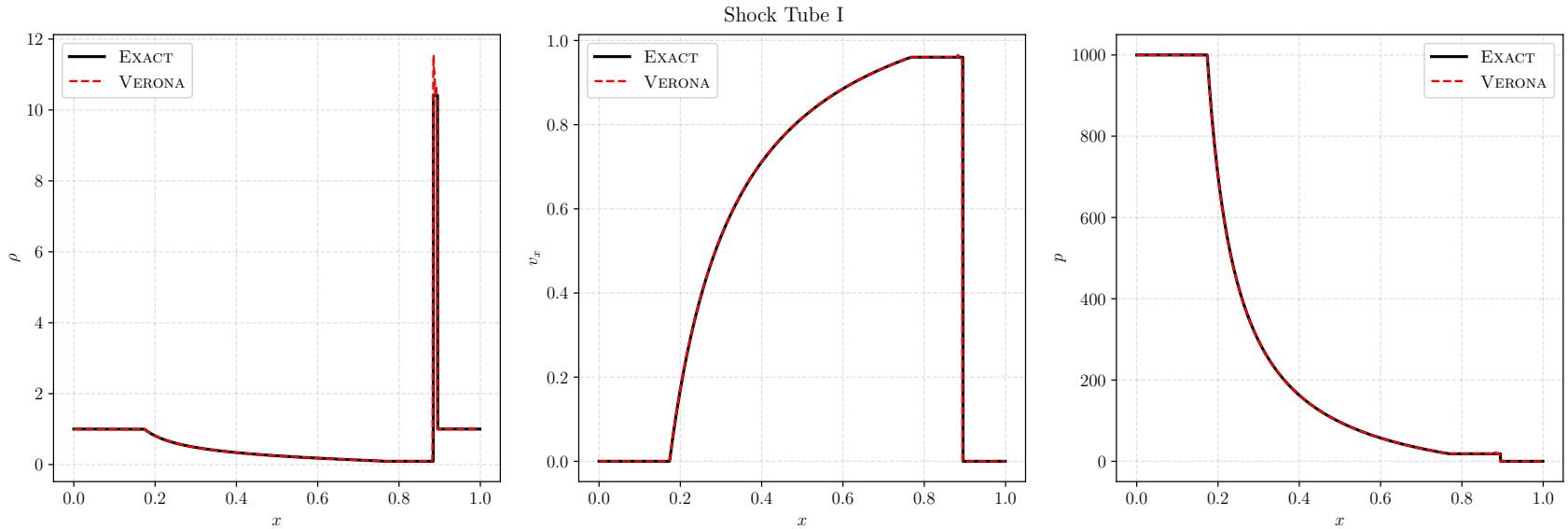
- **Halo synchronization** is performed with **non-blocking CUDA-aware MPI** communication using `MPI.Isend/MPI.Irecv` on GPU buffers.
- Output is saved with **parallel HDF5/MPI-IO**. First, the local data is copied from GPU memory to CPU memory, and then all MPI ranks write their parts together into **one global HDF5 file**.
- **Shared memory** is used in `Fluxes` and `UtoP`, to store temporary data during the calculation.
- **Kernel times** are dominated by `Flux X/Y/Z` (91.9% of GPU time), followed by `UtoP` (5.7%), `Update` (1.5%), and `Limit` (0.6%).

Solver validation: Thompson shock tubes

- We validate the VERONA solver by comparing **numerical** results with the **exact** solution for shock-tube tests.

Test	Side	ρ	p	v_x
Shock Tube I ($\Gamma = 5/3$)	L	1.0	10^3	0
	R	1.0	10^{-2}	0

Table 1: The discontinuity is located at $x_0 = 0.5$ in the domain $x \in [0, 1]$.



Weak Scaling of VERONA

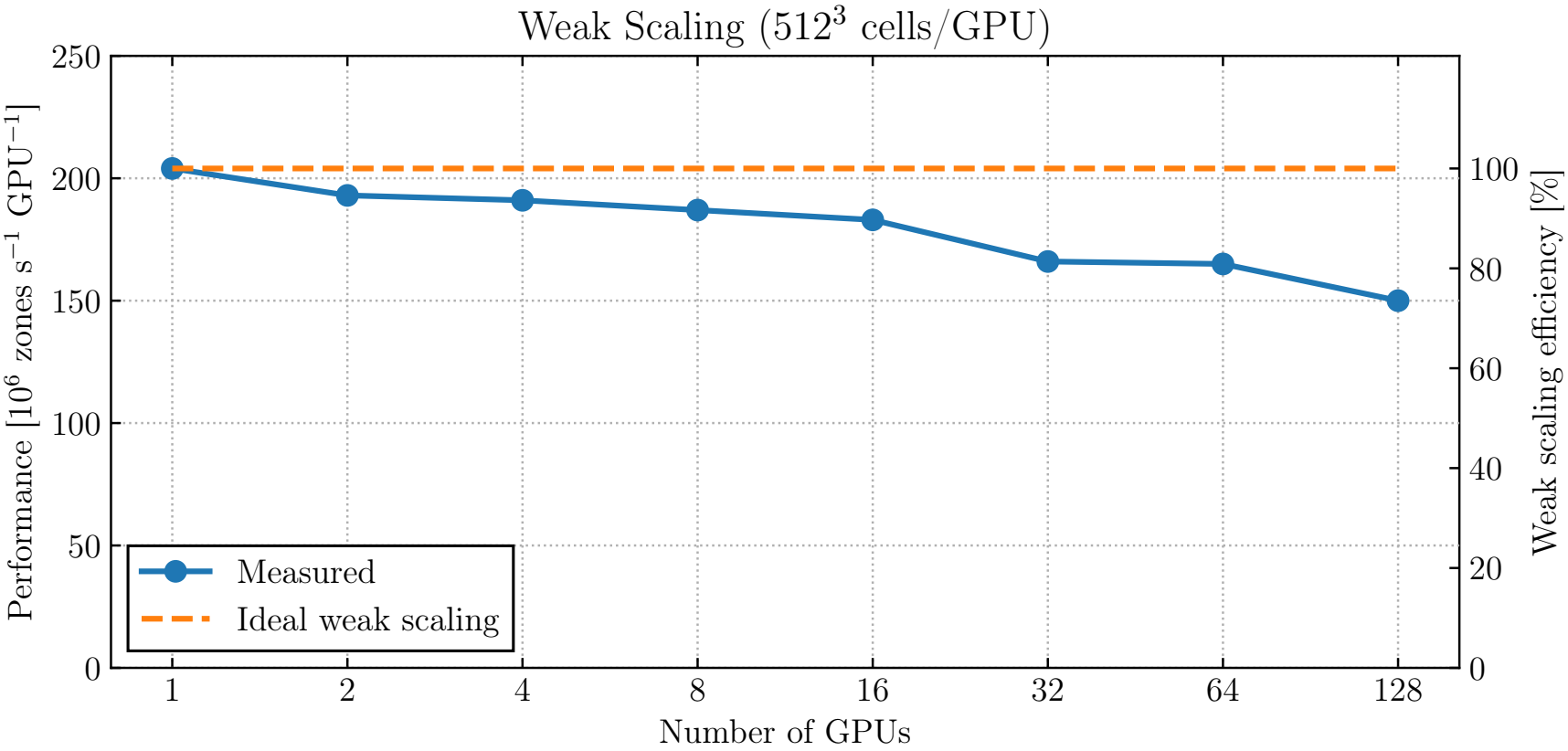


Figure 2: Weak scaling results for VERONA with 512^3 cells per GPU. The code maintains good scalability up to 128 GPUs, reaching about 74% weak scaling efficiency at the largest configuration.

Strong Scaling of VERONA

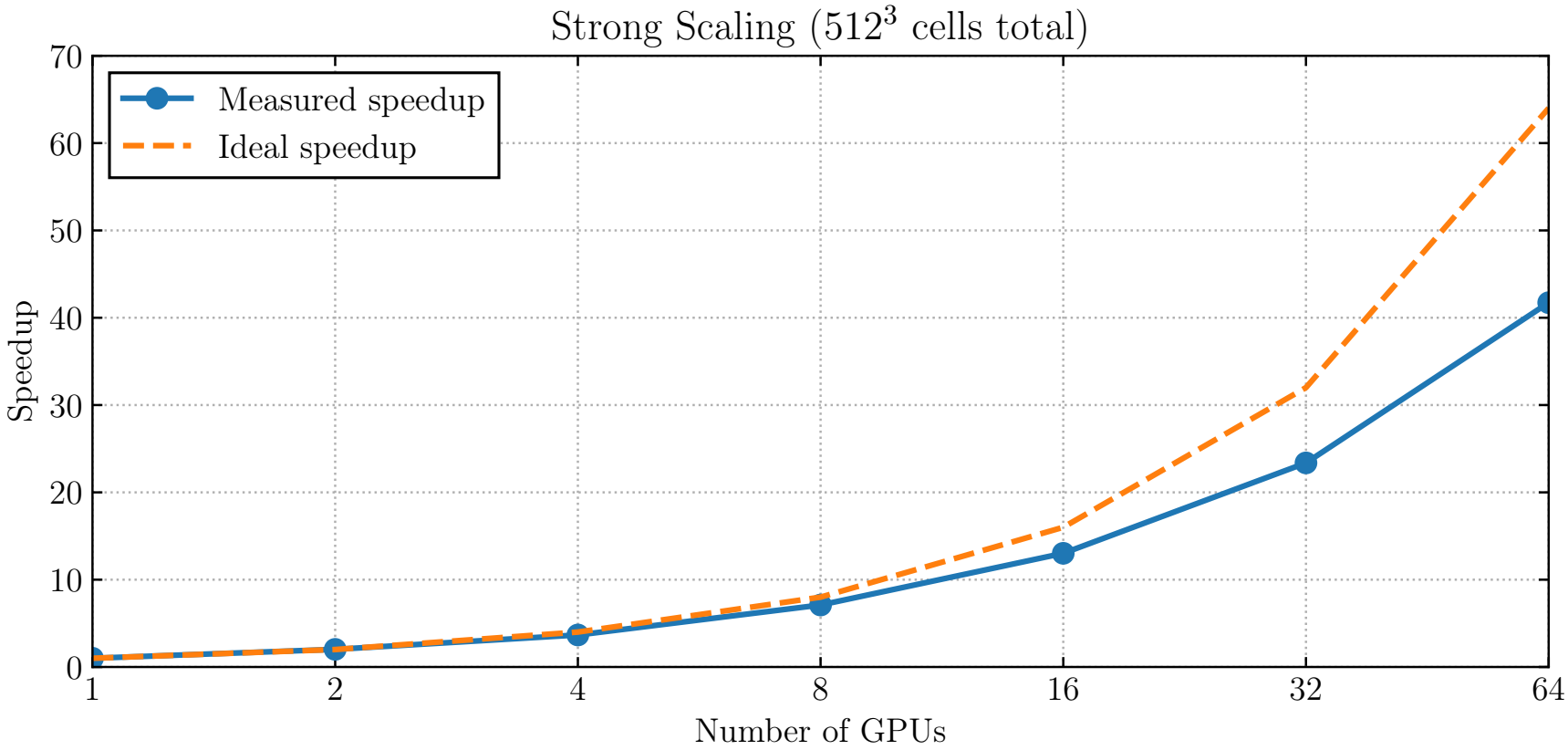


Figure 3: Strong scaling results for VERONA for a fixed global grid of 512^3 cells. The measured speedup, normalized to the single-GPU case, is compared with the ideal linear scaling.

Astrophysical application: jet breakout

- A **relativistic jet** is injected with initial Lorentz factor $\Gamma = 10$ into a progenitor star of radius $R_\star = 4 \times 10^{10}\,\mathrm{cm}$ (Wolf–Rayet star).
- The resolution is set to $N_x \times N_y \times N_z = 4096 \times 2048 \times 2048$, which is presumably the **largest** ever used for jet breakout without AMR.

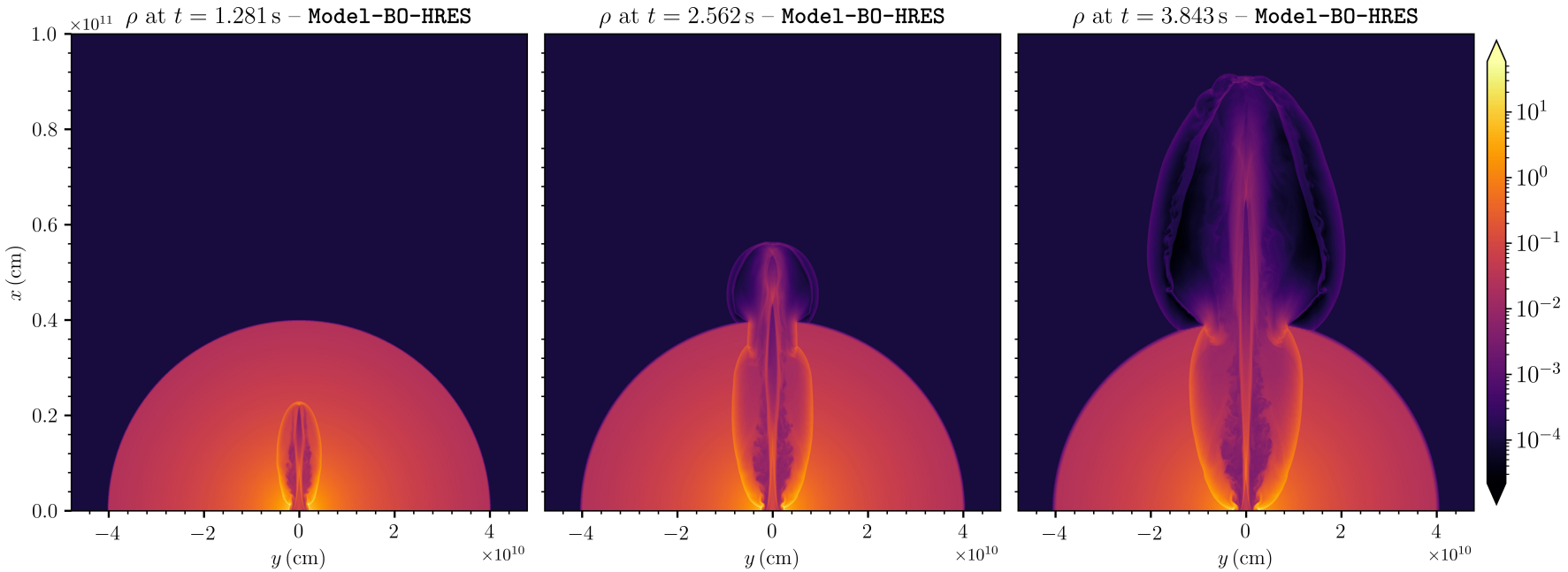


Figure 4: A relativistic jet propagates through a star. After crossing the star, the jet breaks out into surrounding space and material around the jet forms a cocoon.

Astrophysical application: jet breakout

- We obtained a **successful breakout** with time $t_{\rm bo} \sim 2.5\,{\rm s}$.
- In three-dimensional simulations, high resolution reduces numerical dissipation and enables the capture of **small-scale structures**.

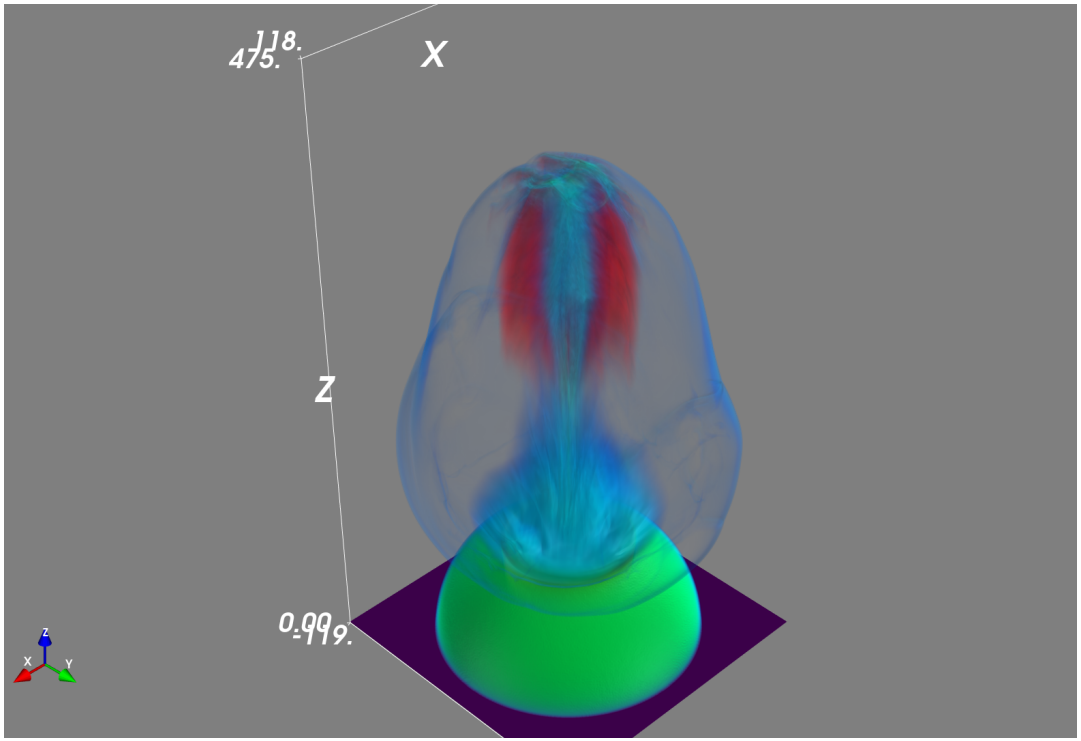


Figure 5: Three-dimensional visualisation of a jet breakout simulation. The red color indicates a highly relativistic outflow with $\Gamma > \sqrt{2}$.

Future Prospects for VERONA: SRMHD and GRMHD

- We have developed a preliminary version of a **special relativistic magnetohydrodynamics** (SRMHD) module in VERONA, using the **constrained transport** method ($\nabla \cdot \boldsymbol{B} = 0$).
- Future developments may include an extension to **general relativistic magnetohydrodynamics** (GRMHD).

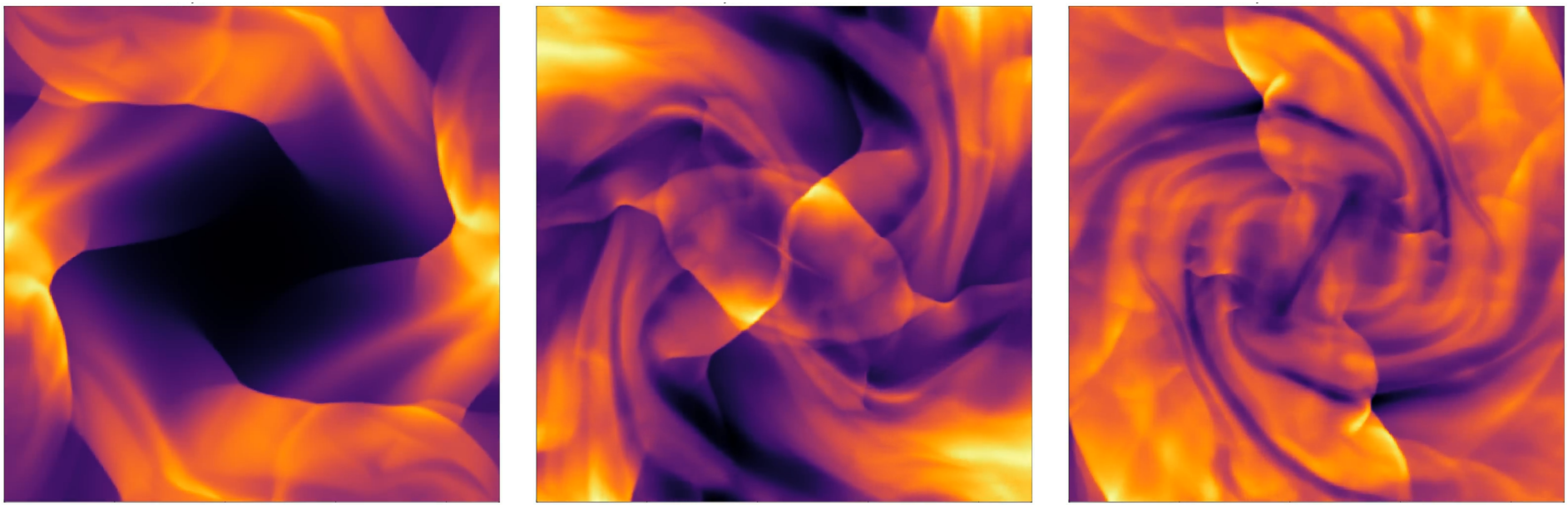


Figure 6: Orszag–Tang vortex test using the SRMHD version of VERONA. Panels show the mass density ρ.

Conclusions on VERONA

- We achieved performance exceeding $2 \times 10^8\,\mathrm{zones\,s^{-1}}$ in a realistic application on a NVIDIA A100 (40GB), using WENO-Z and RK2, making our code **among the fastest**, while offering significantly improved **maintainability** thanks to `Julia`.
- Thanks to access to the PLGrid supercomputing infrastructure, this initially **dormitory project** started by two students evolved into a high-performance scientific code.
- The code is publicly available:

Thank You for Your Attention!

All calculations presented in this work were performed on the **Athena supercomputer** at ACK Cyfronet AGH.