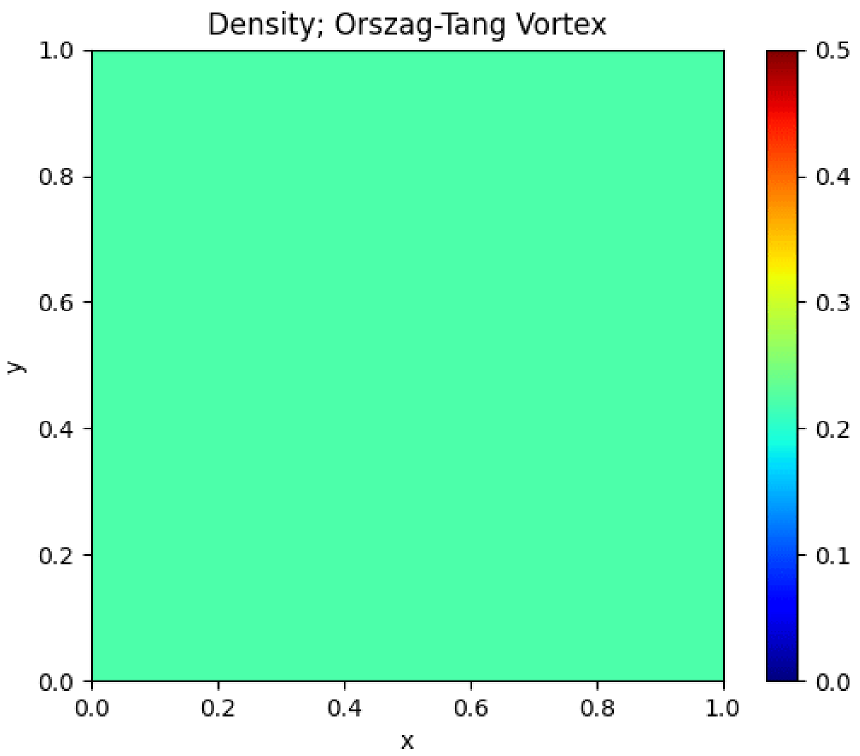
jf1uids

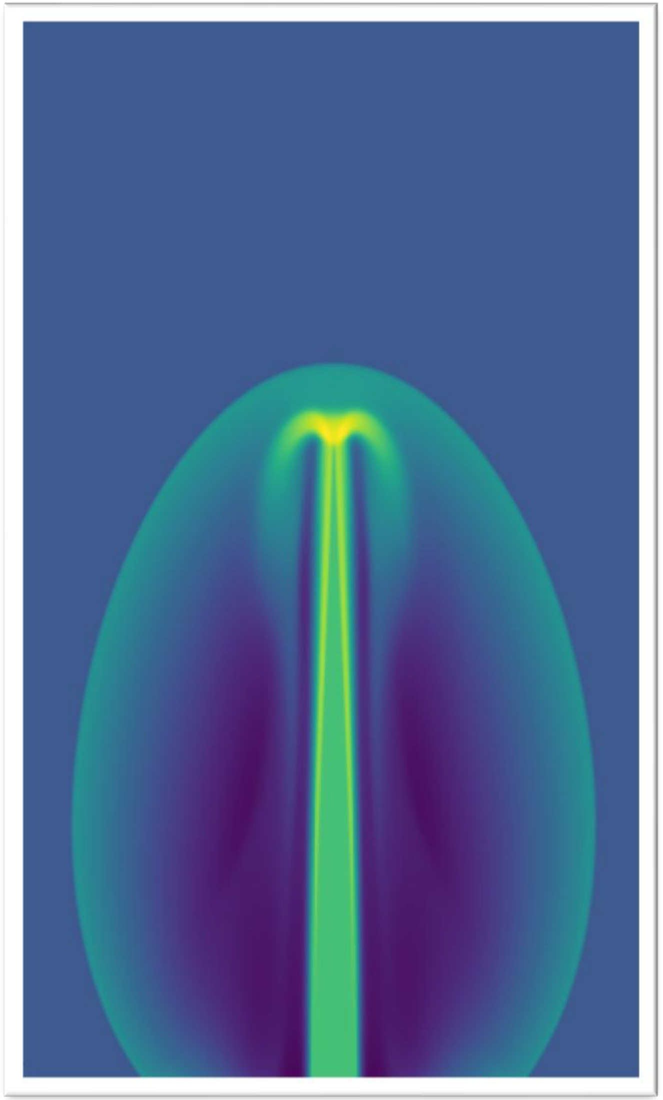
towards differentiable magnetohydrodynamics on exascale hardware



https://github.com/leo1200/jf1uids/

Features of jf1uids

- differentiable, GPU-ready fluid simulator written in JAX
- inherently divergence free approach to MHD (based on (Pang and Wu, 2024))
- novel (near-)energy-conserving scheme for self-gravity stable at discontinuities
- conservative geometric formulation for radially symmetric problems
- physics modules in development: stellar wind, cosmic rays, cooling, …
- fully open source



there is not differentiable, GPU-ready MHD code yet

		AREPO	ATHENA	SHAMROCK	JAXFLUIDS
exascale GPU hardware e.g. Jupiter in Jülich with NVIDIA GH200 Grace Hopper Chips with H100 GPUs	**runs on the GPU?** Sycl? PyTorch? Kokkos? JAX?	✗	✓	✓	✓
differentiable enable SciML applications	**differentiable**	✗	✗	✗	✓
astrophysics	**mhd, self-gravity, ...**	✓	✓	not yet	✗
accessibility	**open source**	~	✓	not yet	✓

~~finite differencing~~

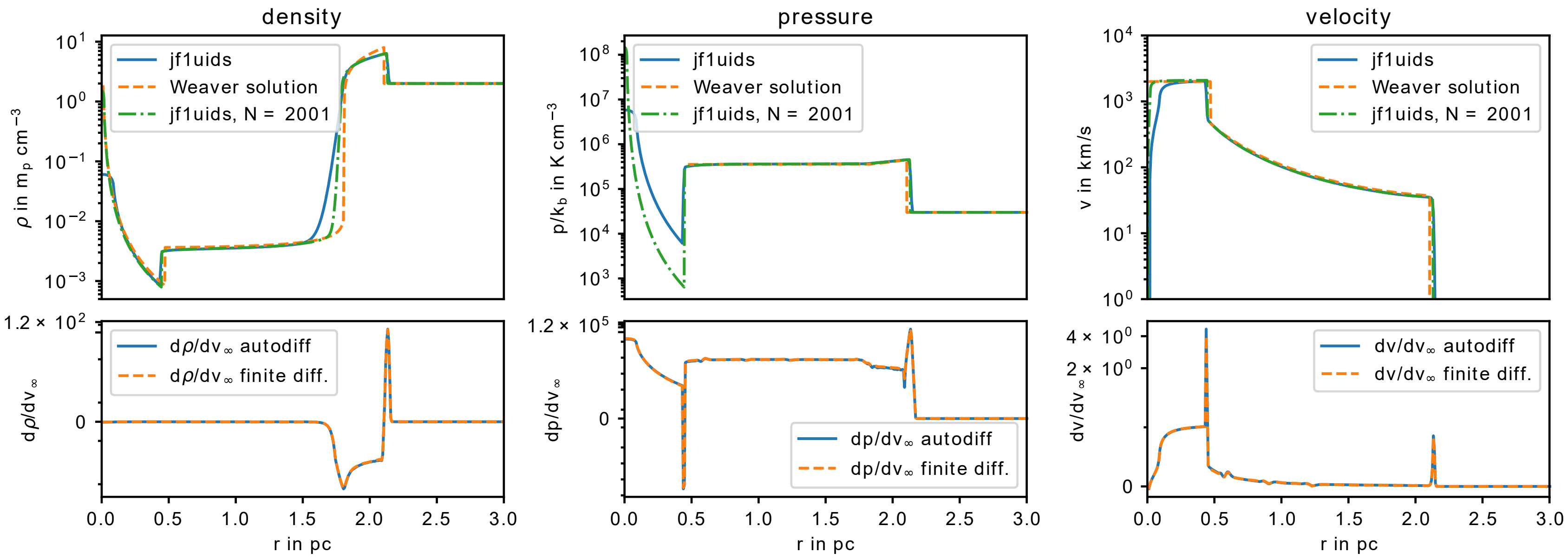
Why (automatic) differentiability?

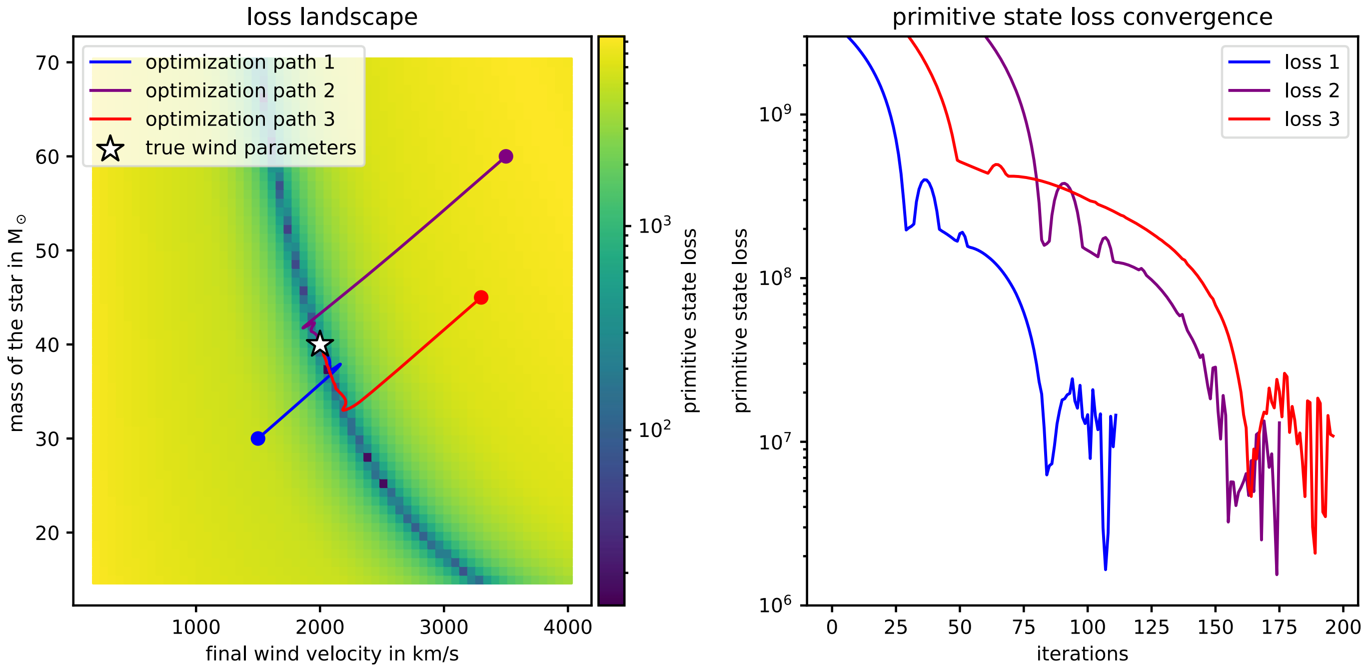
~~continuous adjoint method~~

calculate gradients through the simulator, based on the differentiability of the building blocks provided by the numerical framework → reverse-mode autodiff

- inverse modeling by optimization
- improved simulation-based inference (Zeghal et al., 2022; Holzschuh et Thuerey, 2024))
- corrector in the loop (Um et. al, 2021)

Case Study I: Recovering Physical Parameters

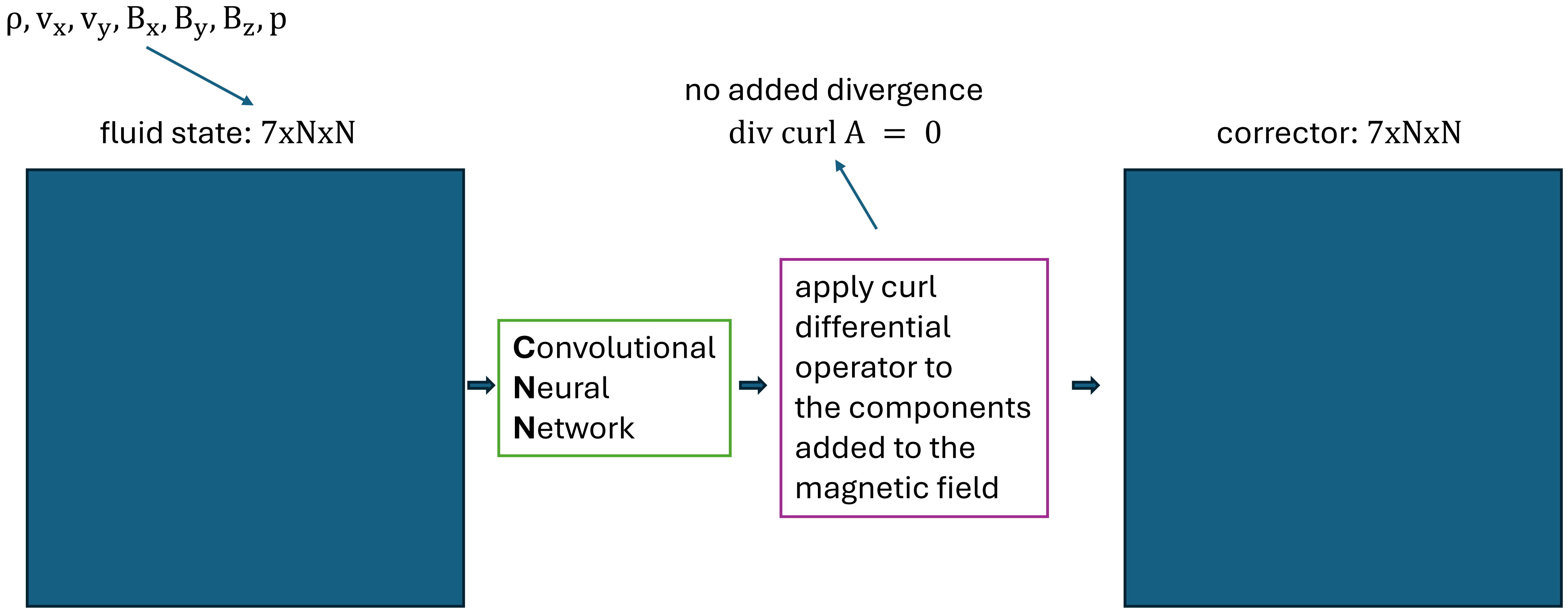




radial 1d stellar wind simulation

aim: retrieve stellar mass and terminal wind velocity from the simulation output

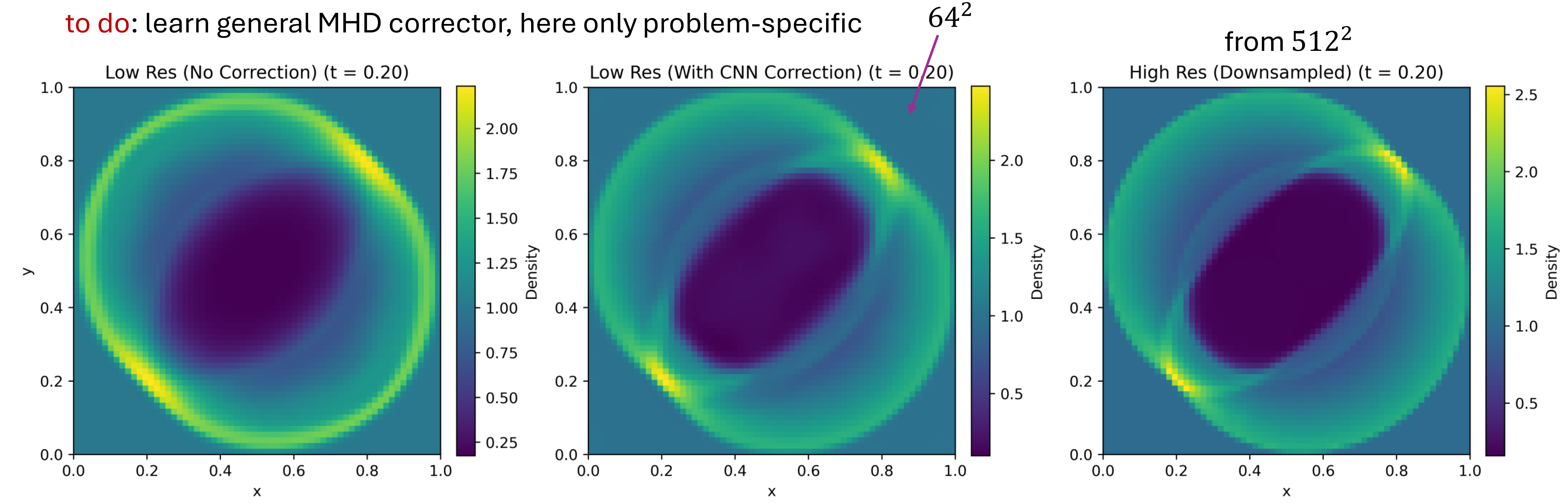
Case Study II: MHD with Corrector in the Loop

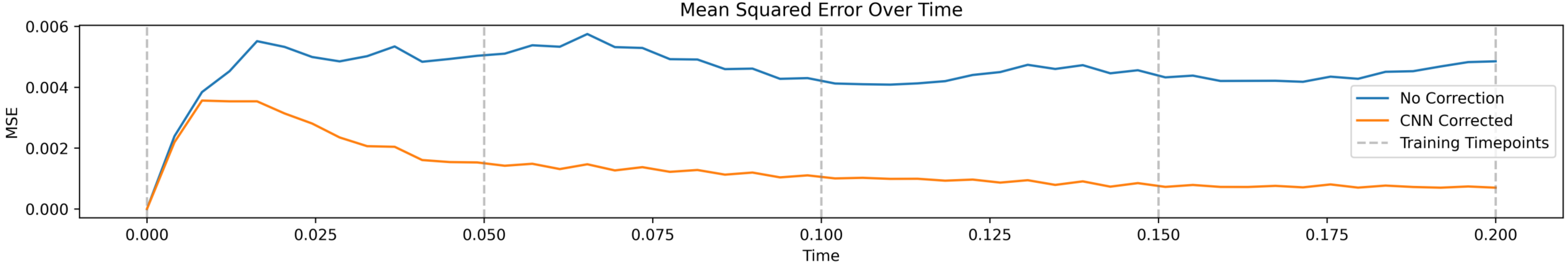


correction step after every timestep: fluid state = fluid state + dt corrector

Only if we train through the differentiable simulator, can the corrector account for the simulator-corrector interaction!

to do: learn general MHD corrector, here only problem-specific





Interesting Numerics I: Divergence Free MHD

Idea: Split MHD equations into a coupled hydrodynamic and magnetic system.

$$\mathbf{U}^{n+1} = S_A^{\frac{\Delta t}{2}} \circ S_B^{\Delta t} \circ S_A^{\frac{\Delta t}{2}} \mathbf{U}^n$$

magnetic

hydro

div curl … = 0

finite difference magnetic update (self-consistently also evolves the gas velocity field (and energy)):

Lorentz force

$$S_B \quad \begin{cases} \rho \partial_t \mathbf{v} = -\mathbf{B} \times \operatorname{curl} \mathbf{B}, \\ \partial_t \mathbf{B} = \operatorname{curl}(\mathbf{v} \times \mathbf{B}), \end{cases}$$

implicit System!, if it holds: div curl … = 0 ✅

R = (**B**, **v**), implicit 2nd order scheme, fixed point iteration

$$\mathbf{R}^{n+1} = \mathbf{R}^n + \Delta t \boldsymbol{\Psi}\left(\frac{\mathbf{R}^n + \mathbf{R}^{n+1}}{2}\right)$$

in practice only a few iterations

based on Pang and Wu, 2024

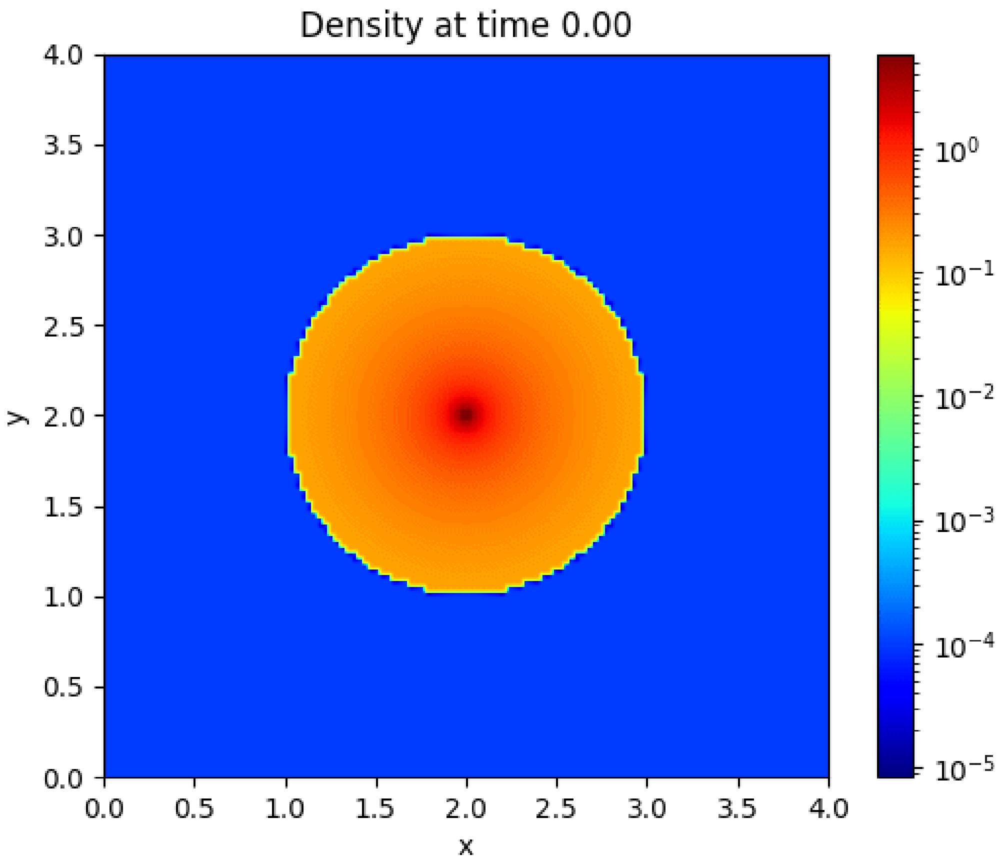
Interesting Numerics II: Improved Self-Gravity

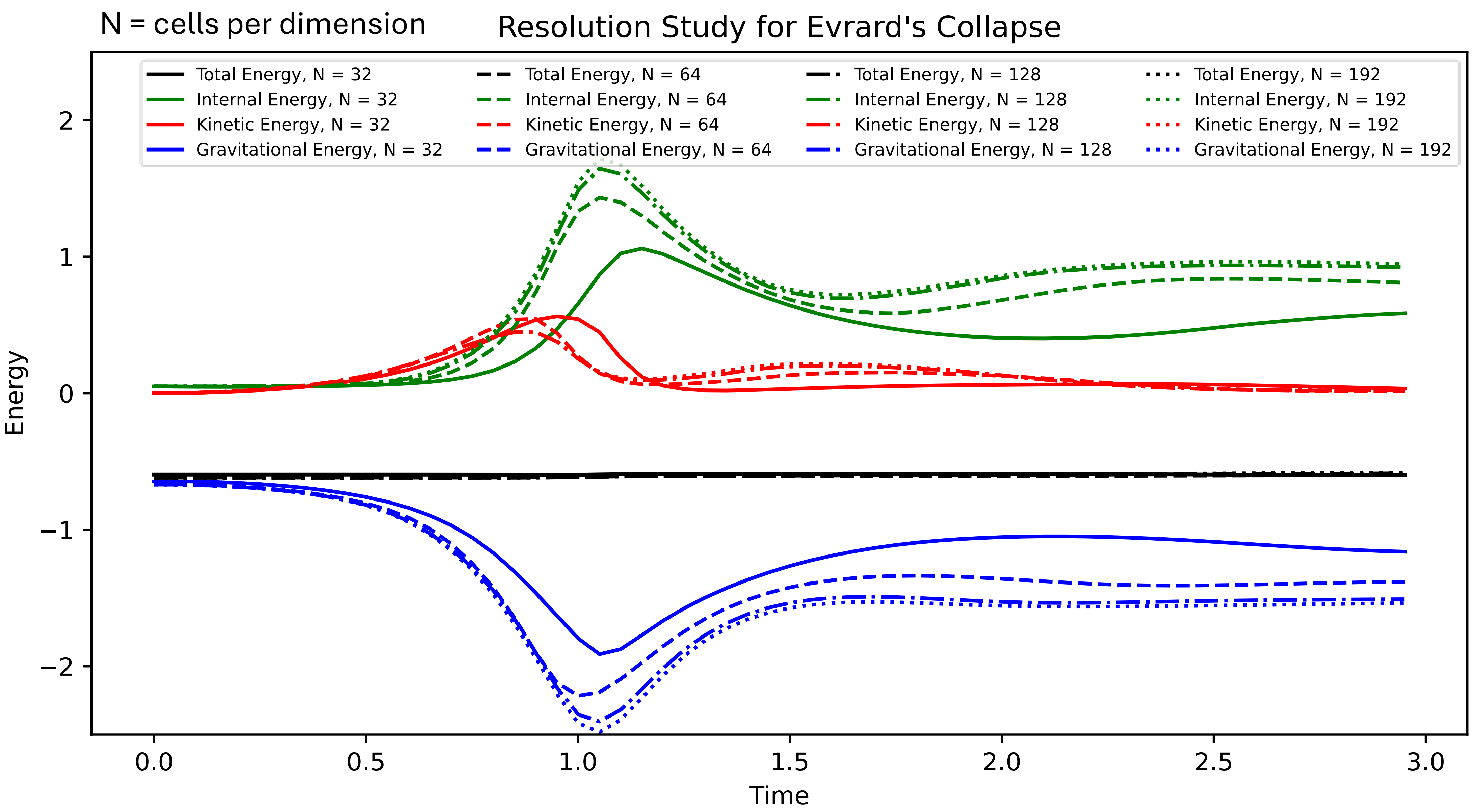
Which flux is moving in the field, how to subtract energy?

a) use the centered fluxes (e.g. in Pluto) – problem: energy not conserved because the cell centered bulk fluxes are not the actual Riemann fluxes moved in the potential

b) use the Riemann fluxes for the energy update (as used in ATHENA) – problem in the standard approach: both cells do half of the work irrespecive of their energy content, leading to negative pressures at discontinuities

our novel approach: only the cell the net flux is coming from pays with its energy (or do a more advanced split of the Riemann flux)

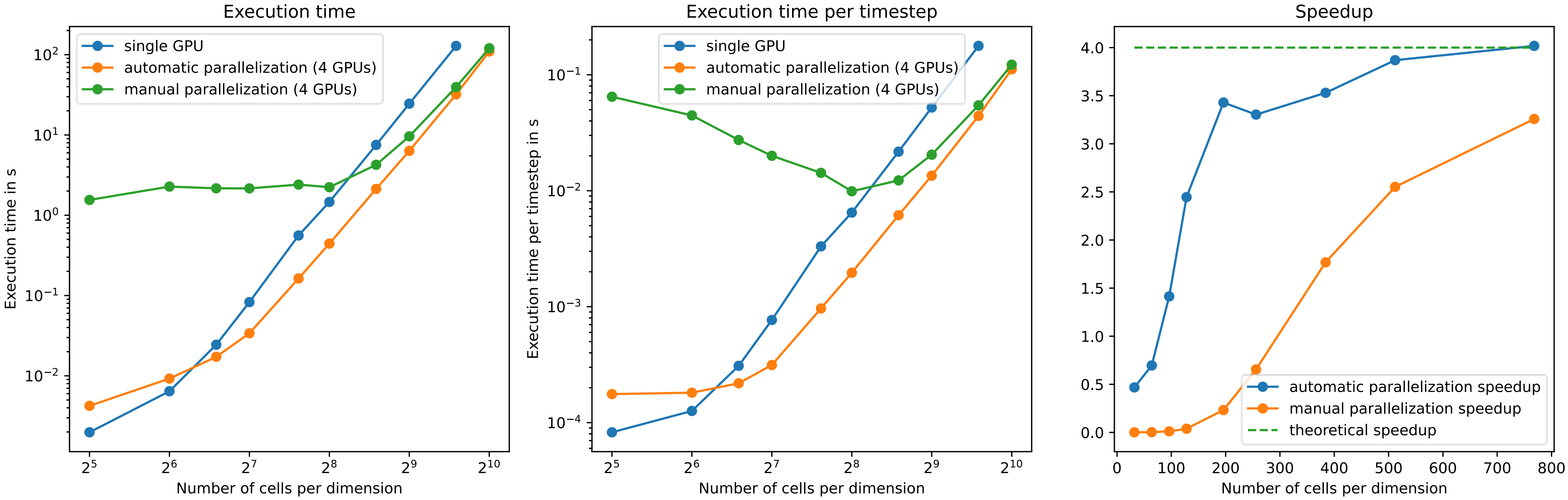




N = cells per dimension
Resolution Study for Evrard's Collapse
Total Energy, N = 32
Internal Energy, N = 32
Kinetic Energy, N = 32
Gravitational Energy, N = 32
Total Energy, N = 64
Internal Energy, N = 64
Kinetic Energy, N = 64
Gravitational Energy, N = 64
Total Energy, N = 128
Internal Energy, N = 128
Kinetic Energy, N = 128
Gravitational Energy, N = 128
Total Energy, N = 192
Internal Energy, N = 192
Kinetic Energy, N = 192
Gravitational Energy, N = 192
Energy
2
1
0
−1
−2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Time

Preliminary Scaling Results

on simpler test code tinyfluids (https://github.com/leo1200/tinyfluids)

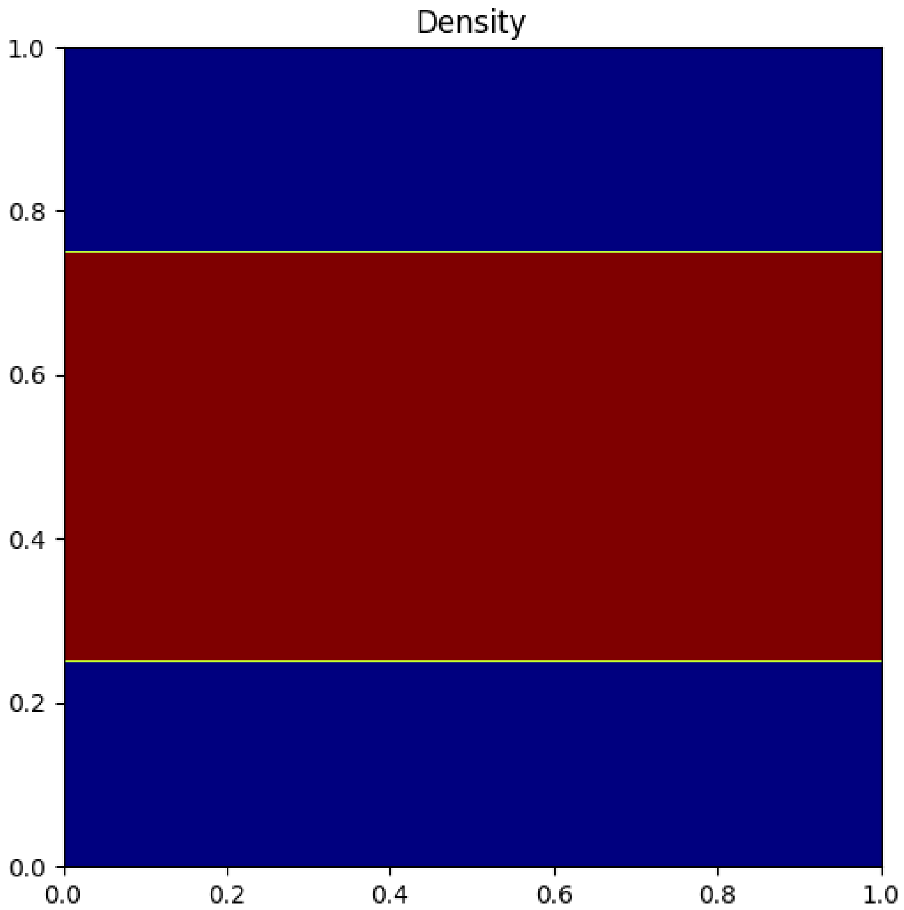


Literature

- Pang and Wu, 2024: https://arxiv.org/abs/2410.05173
- Zeghal et al., 2022: https://arxiv.org/abs/2207.05636
- Holzschuh et Thuerey, 2024: https://arxiv.org/abs/2410.22573
- Um et. al, 2021: https://arxiv.org/abs/2007.00016

jf1uids

towards differentiable magnetohydrodynamics on exascale hardware



https://github.com/leo1200/jf1uids/