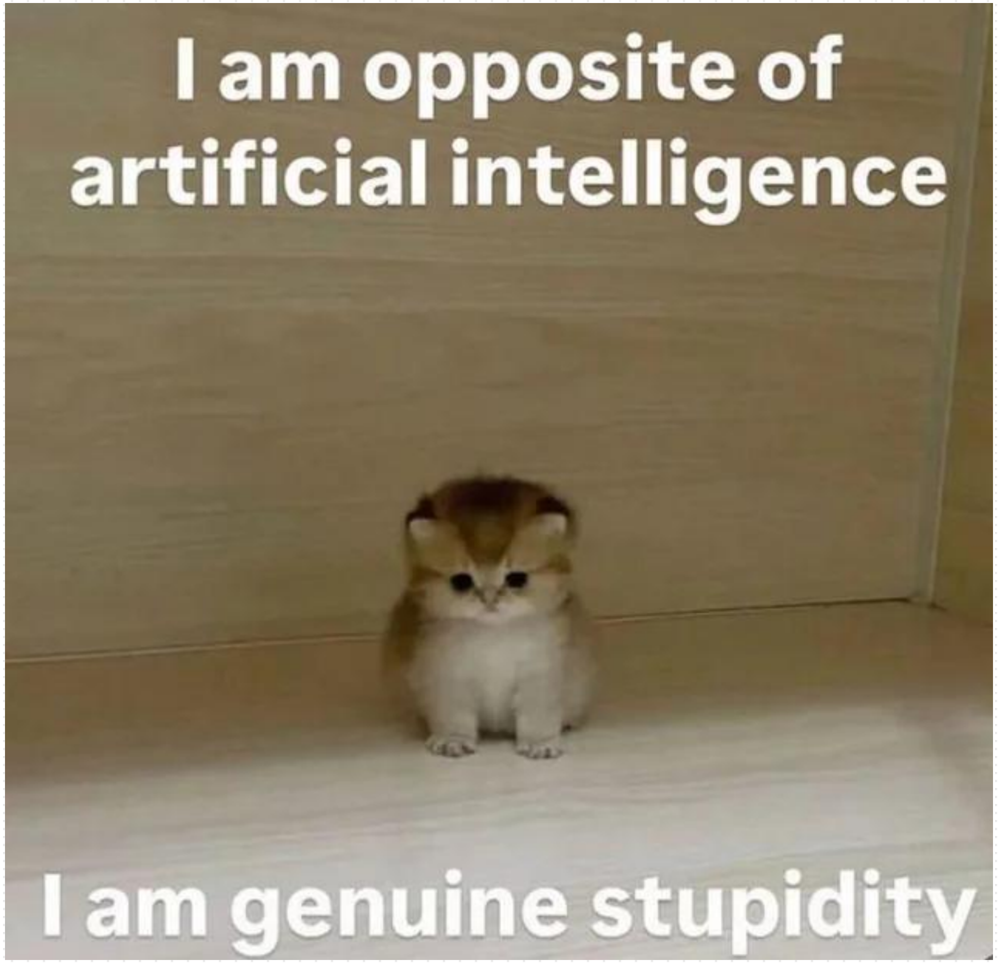
INSTITUTE FOR ADVANCED STUDY

GRMHD in AthenaK

Jacob Fields

2026 AthenaK Summer School

A disclaimer

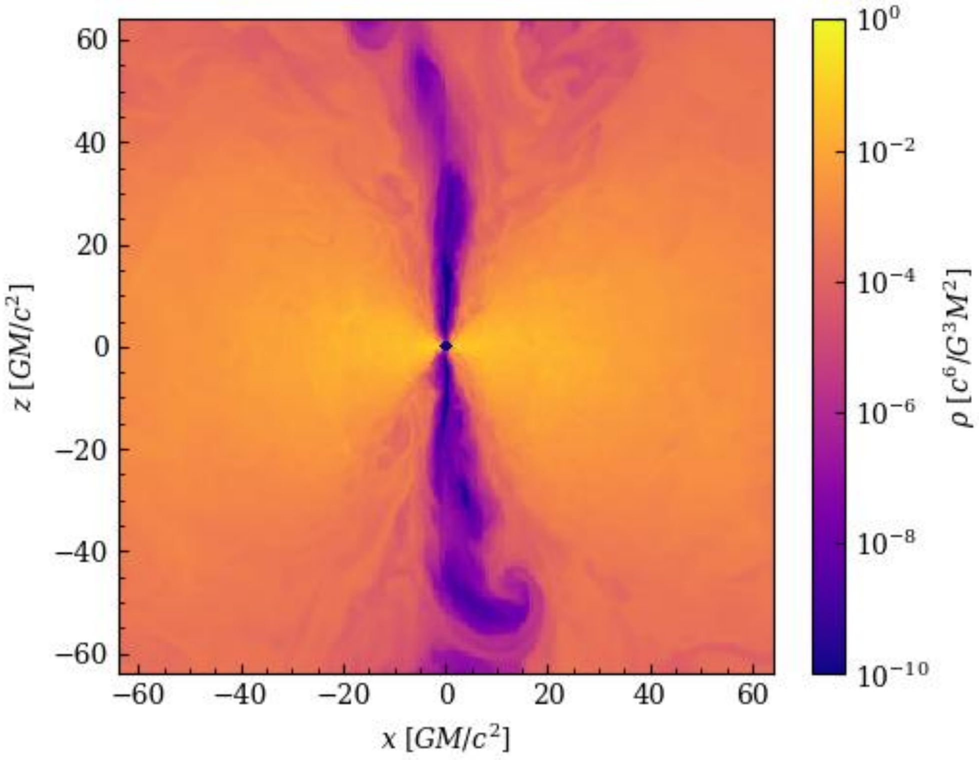


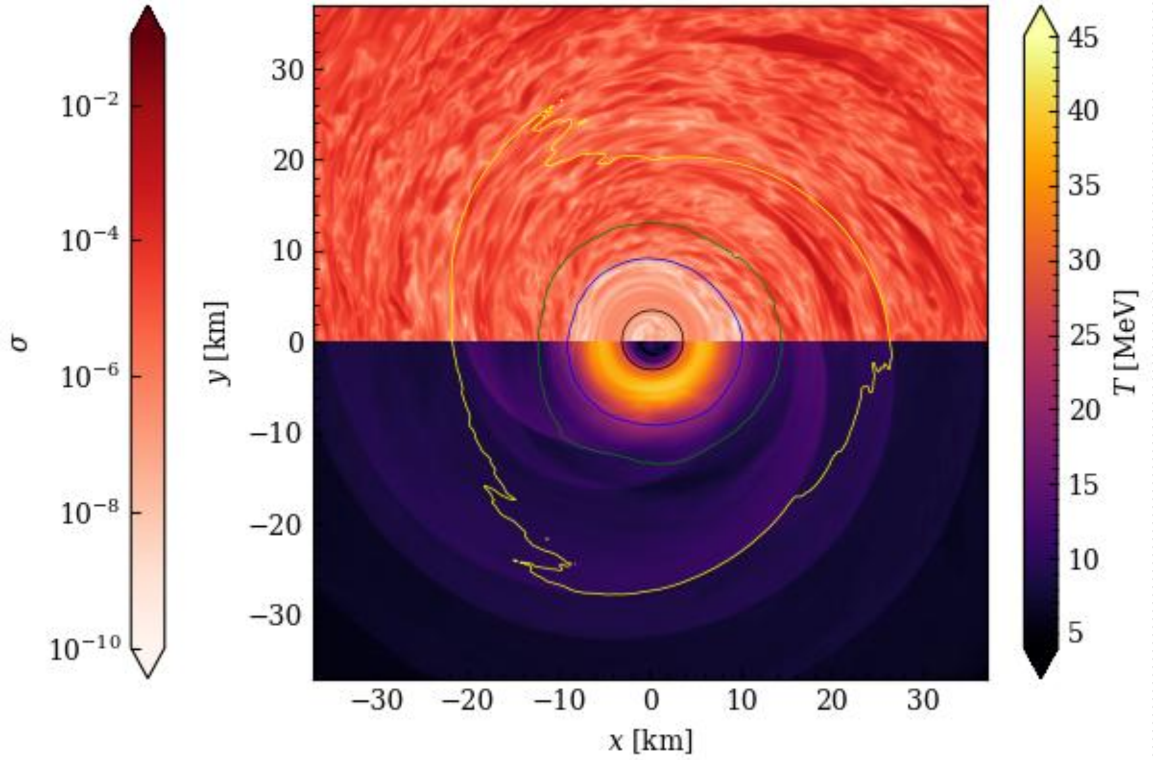
Challenges with GRMHD

Where Newtonian MHD breaks down

- Relativistic velocities: $v \gtrsim 0.1c$
- Gravity is strong: $\frac{GM}{c^2} \gtrsim 0.1r$
- Relativistic energies: $\rho\epsilon \sim \rho c^2, B^2 \sim \rho c^2$

What kinds of problems need GRMHD?





GRMHD equations

- Conservation of baryon number:

$$\nabla_\mu(\rho u^\mu) = 0$$

- Conservation of stress-energy:

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

$$T^{\mu\nu} = (\rho h + b^2)u^\mu u^\nu + \left(P + \frac{b^2}{2}\right)g^{\mu\nu} - b^\mu b^\nu$$

- Gauss-Faraday law:

$$\nabla_\mu(* F)^{\mu\nu} = 0,$$

$$(* F)^{\mu\nu} = b^\mu u^\nu - b^\nu u^\mu$$

Flux-conservative GRMHD equations

- Baryon mass conservation:

$$\partial_t(\sqrt{-g}\rho u^0) + \partial_i\left(\sqrt{-g}\rho u^i\right) = 0,$$

- Momentum and energy conservation:

$$\partial_t\left(\sqrt{-g}T^t_\mu\right) + \partial_i\left(\sqrt{-g}T^i_\mu\right) = \frac{1}{2}\sqrt{-g}T^{\alpha\beta}\partial_\mu g_{\alpha\beta}$$

- Faraday's law:

$$\partial_t\left(\sqrt{-g}\hat{B}^j\right) + \partial_i(\sqrt{-g}\left(b^j u^i - b^i u^j\right) = 0,$$
$$\hat{B}^i = (*F)^{it}$$

- Gauss's law:

$$\partial_i\left(\sqrt{-g}\hat{B}^i\right) = 0$$

3+1 form

$$\partial_t \widetilde{D} + \partial_i\left(\alpha \widetilde{D}\, \hat{v}^i\right) = 0,$$
$$\partial_t \tilde{S}_j + \partial_i\left(\tilde{S}_j \hat{v}^i - \frac{\alpha b_j \tilde{B}^i}{W} + \alpha \tilde{P}_{\rm tot}\delta^j_i\right) = \frac{1}{2}\alpha \tilde{S}^{ik}\partial_j \gamma_{ik} + \tilde{S}_i \partial_j \beta^i - \tilde{E}\partial_j \alpha,$$
$$\partial_t \tilde{\tau} + \partial_i\left(\tilde{\tau}\hat{v}^i - \frac{\alpha^2 b^0 \tilde{B}^i}{W} + \alpha \tilde{P}_{\rm tot} v^j\right) = \alpha K_{ij}\tilde{S}^{ij} - \tilde{S}^i \partial_i \alpha,$$
$$\partial_t \tilde{B}^j + \partial_i\left(\alpha\left[\tilde{B}^j \hat{v}^i - \tilde{B}^i \hat{v}^j\right]\right) = 0,$$

where

$$\tilde{A} = \sqrt{\gamma} A, \qquad D = \rho W, \qquad S_i = (\rho h W^2 + B^2) v_i - \left(B^l v_l\right) B_i,$$
$$\tau = \rho h W^2 + B^2 - P_{\rm tot} - D, \qquad B^i = n_\mu (*F)^{\mu i}, \qquad \hat{v}^i = u^i / W$$

Solving GRMHD numerically

- Still flux-conservative, can just use FV + CT
  - Reconstruct primitive variables to interfaces
  - Solve Riemann problem at interfaces
  - Compute electric fields at edges
  - Add source terms
  - Integrate with RK solver with a stable timestep
  - Compute primitives from conserved variables
- But some of these steps look a little different

Some things made simpler by GRMHD

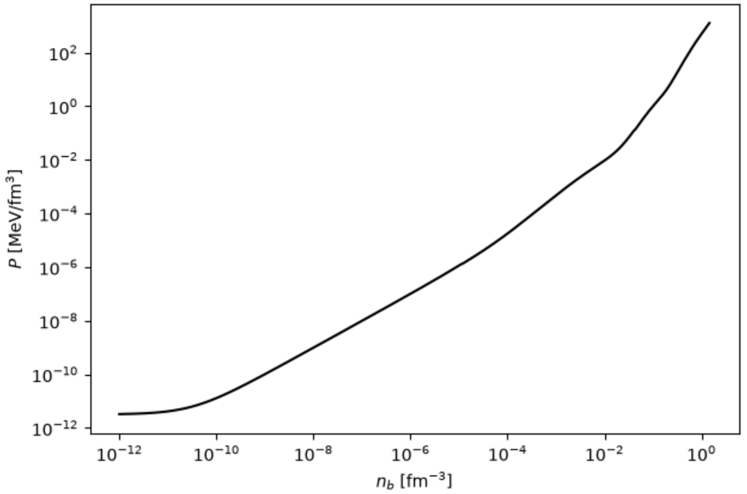
- Gravity is now hyperbolic! (See Manuela and Hengrui's lectures)
    - Arguably easier to solve numerically[1]
- Simpler time step control
    - Maximum characteristic speed, fluid speed never exceeds c
    - Maximum stable timestep $\Delta t \rightarrow \frac{C\Delta x}{c}$ as $v \rightarrow c$
- Curvilinear and non-inertial coordinates baked into formalism[2]

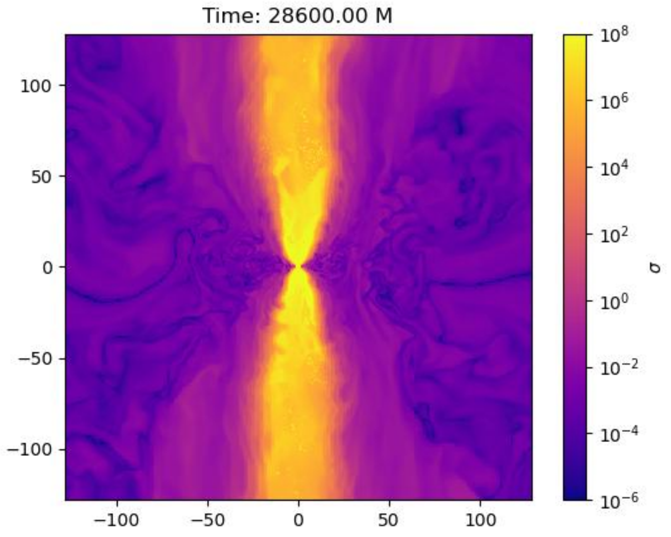
Some things made harder by GRMHD

- Basically everything else.
- Relationship between conserved and primitive variables:
  - $(D, S_i, \tau) \to (\rho, v^i, P)$ does not have general closed-form solution
  - More constraints on physical state, difficult to detect and repair a priori
- Ill-conditioned at high speeds
  - $W^2 = (1 - v^2)^{-1} \to \infty$ as $v \to 1$
- Characteristic structure more complicated
  - More difficult to construct accurate Riemann solvers
  - Full GRMHD decomposition suitable for numerical implementation unknown until recently (Teukolsky, 2025)*

Additional problem-specific challenges

- Binary neutron stars:
  - Nuclear EOS needed
  - Initial data is hard
    - Nonlinear constraints
    - Orbits not quite Keplerian
- Accretion disks
  - $\frac{b^2}{\rho} \gg 1$, numerically challenging
  - Resolving energetics is challenging





Some problems have easy solutions

- $v \to 1$ is ill-conditioned
  - Use $W v^i$ wherever possible
- Characteristic structure
  - Use reasonable approximations for wave speeds, more diffusive solvers
- Complex initial data
  - Rely on well-tested and robust external libraries

Some are harder

- Nuclear EOS
    - Tabulated EOS performs poorly on GPUs
- Conserved-to-primitive problem
    - Need complicated root-solving procedure
    - Need a way to handle unphysical states
- Strongly magnetized regime
    - Code becomes unstable in force-free limit

Nuclear EOS

- 3D table in (n, Y_e, T)
  - n and T spaced logarithmically
  - Quality of table depends on underlying model, interpolation procedure, etc.
- Three major performance pain points:
  1. Random memory accesses
  2. Often need $T(n, Y_e, P)$ or $T(n, Y_e, e)$; requires root solve of table
  3. Logs and exponents are expensive

Floating-point Representation

Diagram: Hammond, **JF**, Miller, and Barker, *ApJS* **277**, 2 (2025) (2501.05410)

- Floating point … $> 0)$
- $m \in [1/2,1)$ is … xponent
- Therefore

$$\log_2 x = p + \log_2 m$$

Only thing that needs to be calculated

Can be pulled straight from the floating-point number

Not-quite-transcendental functions

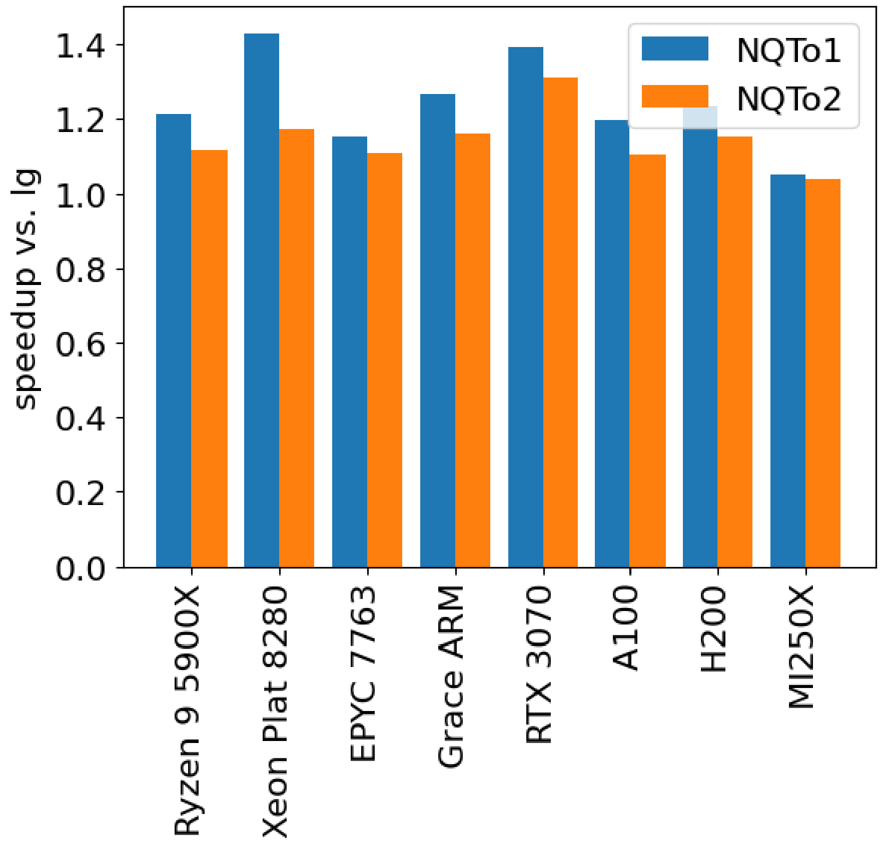
- "Log-like" scaling property of table more important than actual scaling
- Can use an approximation for $\log_2 m$
  - Needs to be C^1
  - Needs to be fast
  - Should be exactly invertible

$$\log_2 x \approx f(x) = -\frac{4}{3}(m-2)(m-1) + p$$

$$m = \frac{1}{2}\left(3 - \sqrt{1 - 3f(m)}\right)$$

NQT functions (cont.)

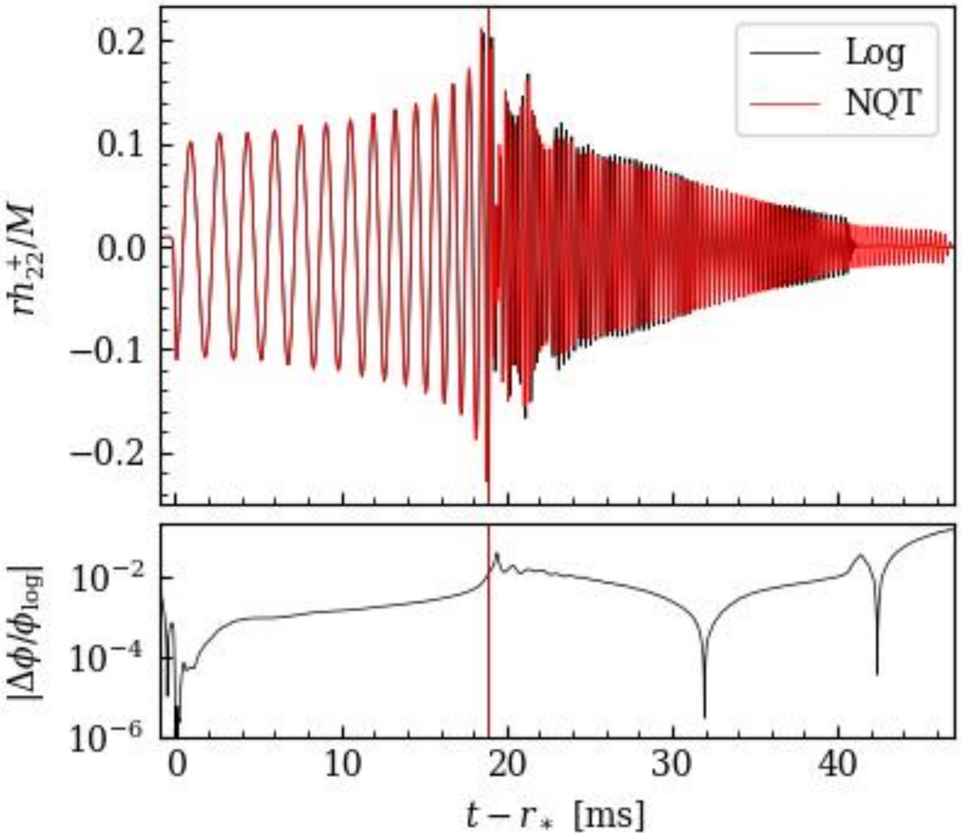
- Implementation in AthenaK based on bit hacks for optimal performance
- Table performance improves on all architectures, sometimes up to 30%
- Differs from log-based interpolation by at most a few percent



Hammond, **JF**, Miller, and Barker, *ApJS* **277**, 2 (2025) (2501.05410)

Effects on Waveforms

- NQT-scaled EOS slightly different
  - BNS waveform very sensitive to EOS
- Comparison run at low resolution shows accumulated phase difference is <1% during inspiral
- Changing resolution produces a bigger shift in merger time



JF, PhD Thesis (2025)

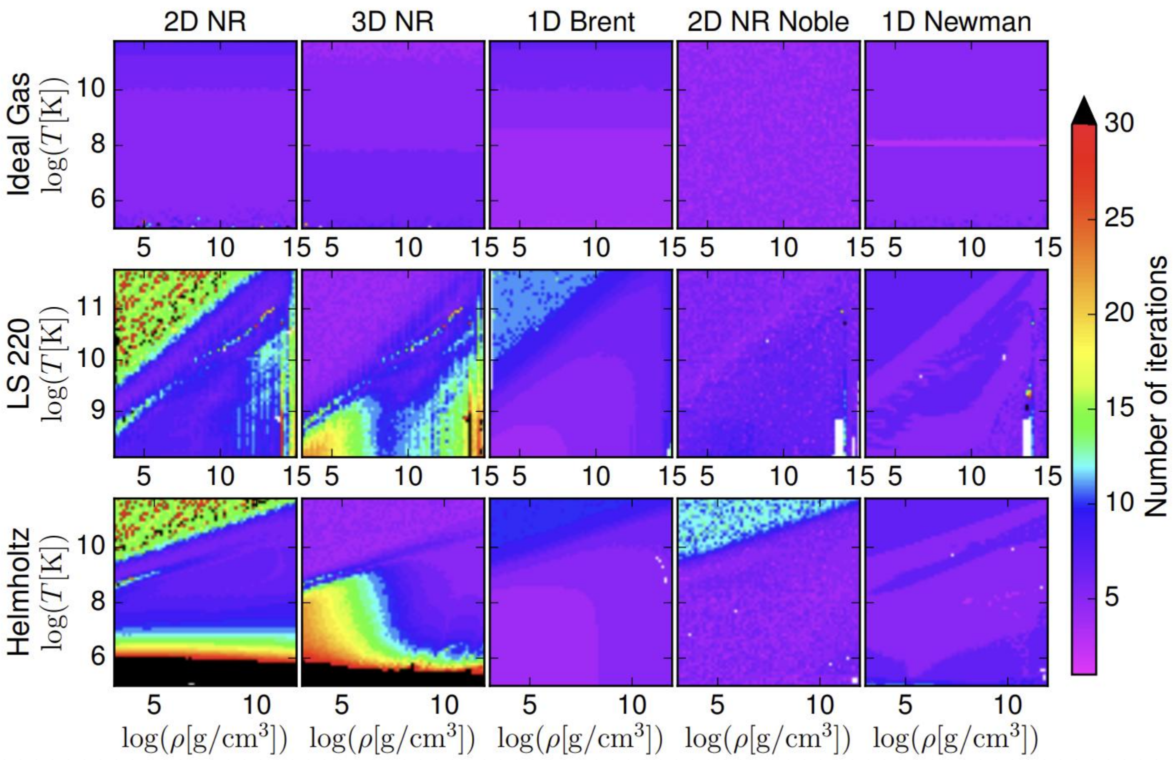
Conserved-to-primitive inversion

- Need to know (ρ, v^i, P) to compute fluxes
- Evolved variables are (D, S_i, τ) or $(\rho u^0, T^t_\mu)$
- What makes $(D, S_i, \tau) \rightarrow (\rho, v^i, P)$ hard?
  - $W^2 = (1 - v^2)^{-1}$ couples directions together
  - S_i depends on EOS via h
- Can generally only be solved iteratively
  - Fast algorithms usually not robust, robust algorithms usually not fast
  - EOS not always smooth => derivative methods not reliable

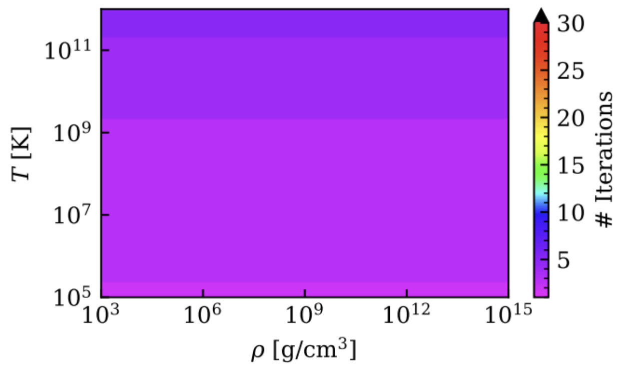
PrimitiveSolver

- General EOS framework
  - Built-in support for a variety of EOS types
  - Easy to add new EOS
- Implements modified version of C2P inversion from Kastaun *et al.* (2021) (the RePrimAnd algorithm)
  - Start with guess for $\mu = \frac{1}{hW} \in (0, \frac{1}{h_{\mathrm{min}}})$
  - Use μ to find and subtract off B-field component
  - Solve hydro problem for ρ and e (or ϵ), refine μ guess, rinse and repeat
  - Robust for high W and b^2/ρ.

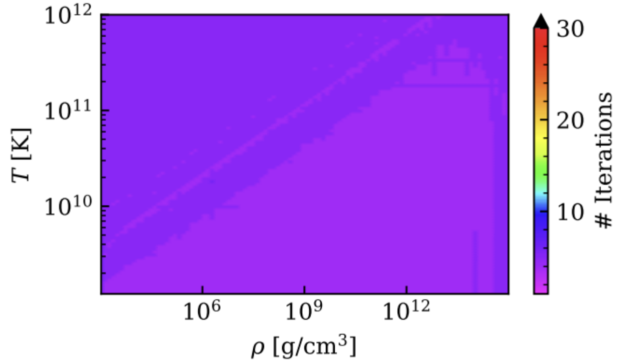
PrimitiveSolver comparison



Siegel *et al.* (2018)



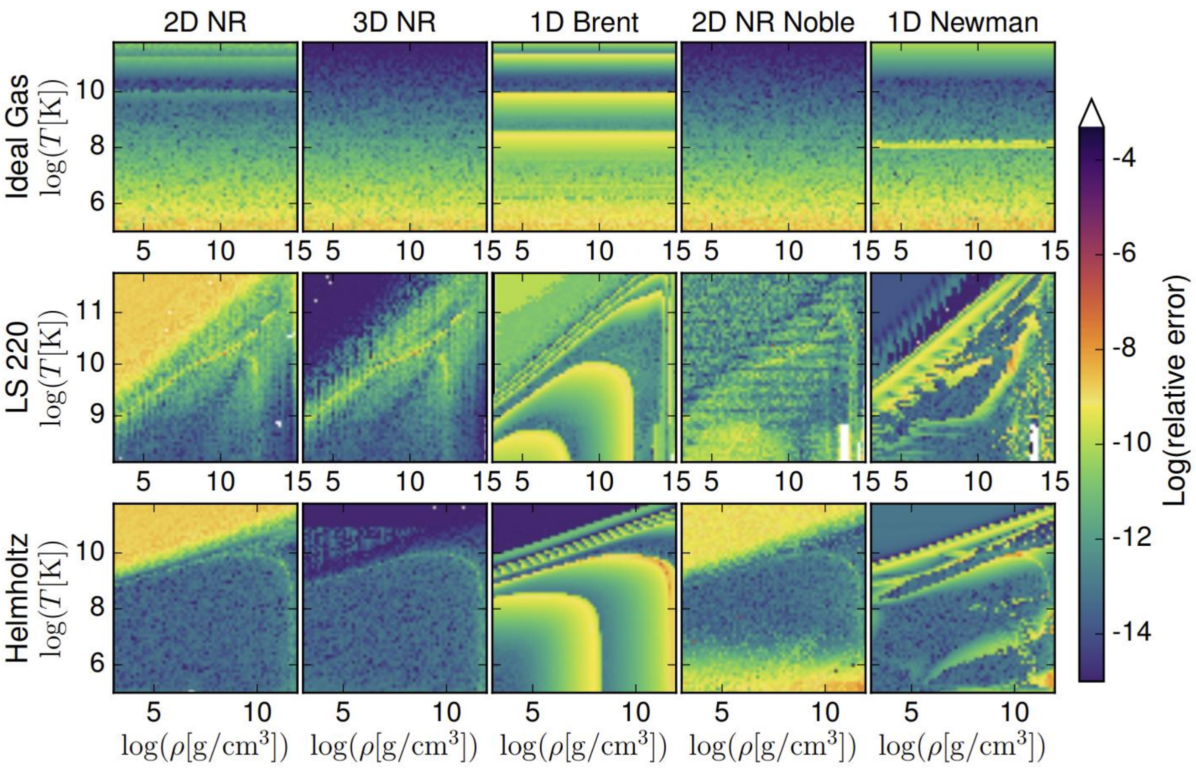
Ideal Gas



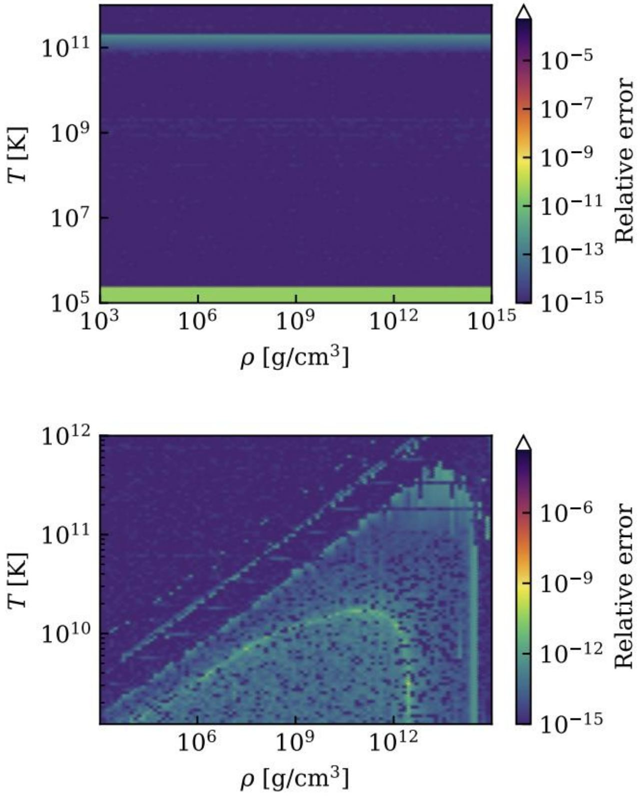
LS220

PrimitiveSolver (AthenaK)

PrimitiveSolver comparison (cont.)



Siegel *et al.* (2018)



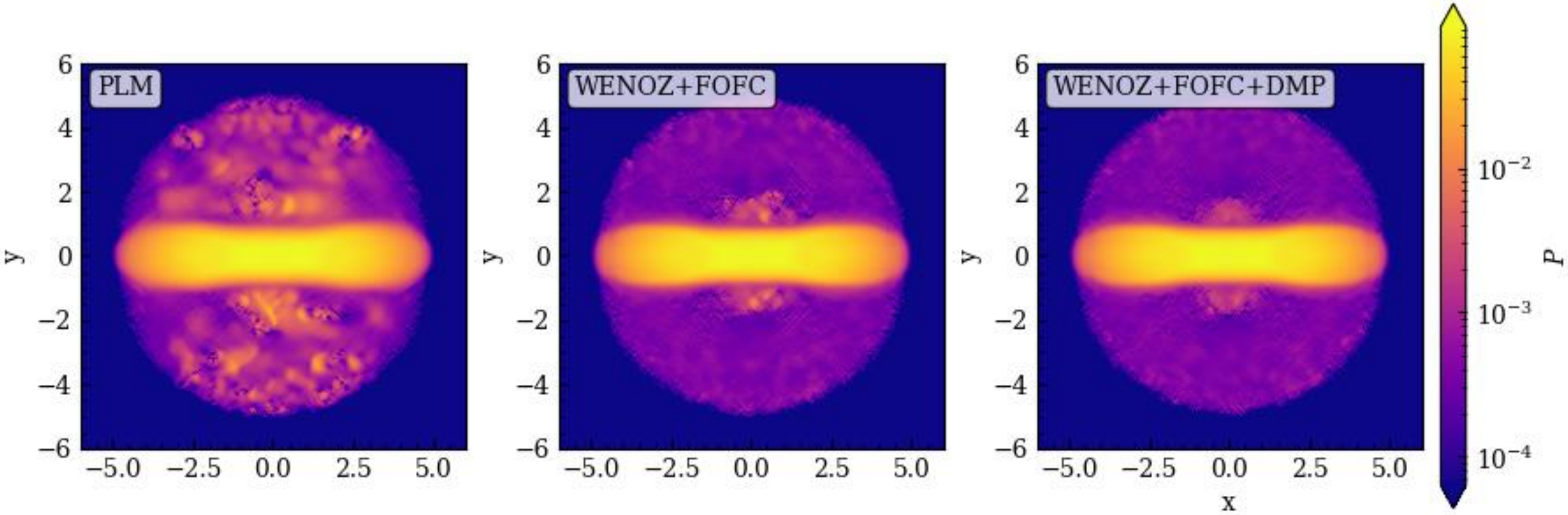
Ideal Gas

LS220

PrimitiveSolver (AthenaK)

A word on magnetization

- C2P is half the battle: PrimitiveSolver/RePrimAnd is reliable up to extreme magnetizations
- First-order flux correction handles the rest
- Test: strongly magnetized blast wave (Komissarov *et al*., 1999)



Using GRMHD in AthenaK

Enabling GRMHD

- Coordinates block:
  - Enable general relativity
  - Set spin (used for CKS spacetime)
  - Enable or disable excision
- Can also use GR solver in flat space; add `minkowski=true` flag to block
- Enables HARM-like solver; only supports CKS/flat space

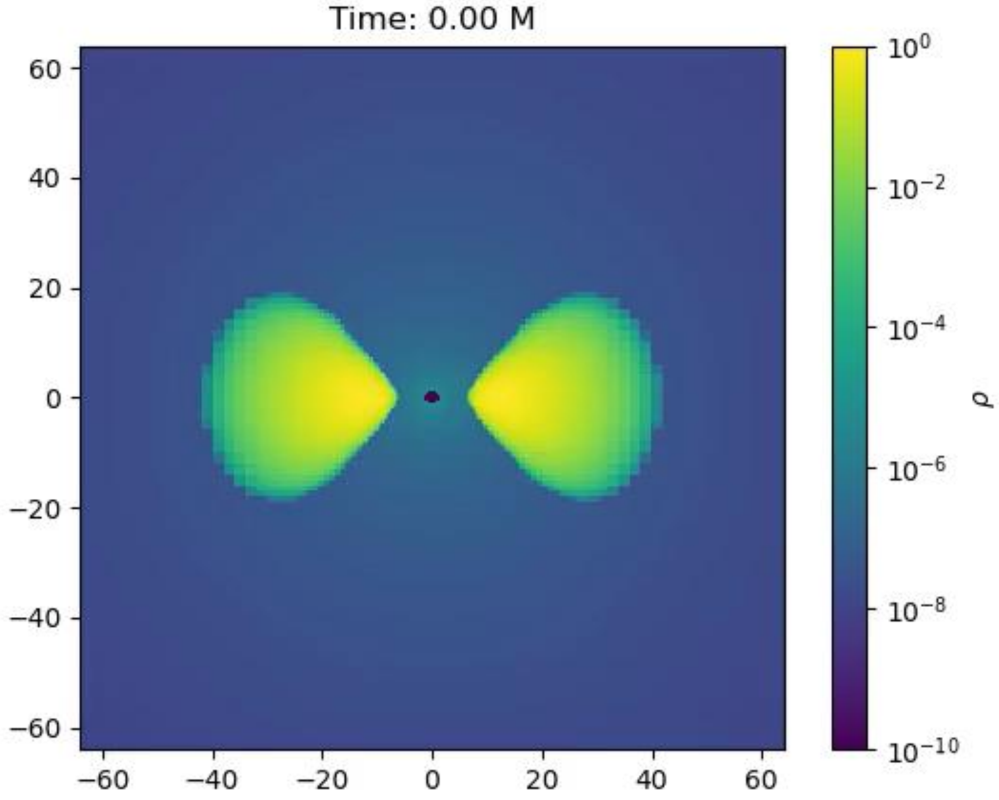
```
<coord>
general_rel = true        # general relativity
m           = 1.0
a           = 0.0
excise      = false
```

Example: black hole accretion disk

- Compile with `-DPROBLEM=fluids/gr_torus`
- Choose black hole spin and torus configuration (see `inputs/grmhd` for examples)
- Support for both Fishbone-Moncrief and Chakrabarti torii initial conditions

```
<coord>
general_rel = true       # general relativity
a           = 0.9375     # black hole spin a (0 <= a/M < 1)
excise      = true       # excise r_ks <= 1.0
dexcise     = 1.0e-10    # density inside excision
pexcise     = 0.333e-12  # pressure inside excision

<mhd>
eos         = ideal      # EOS type
reconstruct = ppm4       # spatial reconstruction method
rsolver     = hlle       # Riemann-solver to be used
dfloor      = 1.0e-10    # floor on density rho
pfloor      = 0.333e-12  # floor on gas pressure p_gas
gamma       = 1.33333333333333333  # ratio of specific heats Gamma
fofc        = true       # Enable first order flux correction
gamma_max   = 20.0       # Enable ceiling on Lorentz factor

<problem>
fm_torus   = true   # Fishbone & Moncrief
r_edge     = 6.0    # radius of inner edge of disk
r_peak     = 12.0   # radius of pressure maximum; use l instead if negative
tilt_angle = 0.0    # angle (deg) to incl disk spin axis relative to BH spin in dir of x
potential_beta_min = 100.0  # ratio of gas to magnetic pressure at maxima (diff locations)
potential_cutoff   = 0.2    # amount to subtract from density when calculating potential
potential_rho_pow = 1.0     # dependence of the vector potential on density rho
rho_min   = 1.0e-5    # background on rho given by rho_min ...
rho_pow   = -1.5      # ... * r^rho_pow
pgas_min  = 0.333e-7  # background on p_gas given by pgas_min ...
pgas_pow  = -2.5      # ... * r^pgas_pow
rho_max   = 1.0       # if > 0, rescale rho to have this peak; rescale pres by same factor
l         = 0.0       # const ang. mom. per unit mass u^t u_phi; only used if r_peak < 0
pert_amp  = 2.0e-2    # perturbation amplitude
user_hist = true      # enroll user-defined history function
```

Example (cont.)



Enabling Valencia/dynamical GRMHD

- Add ADM or Z4c block
- Specify EOS and error policy
- Swap out `pfloor` with `tfloor`
- Optional:
  - Flooring threshold
  - Some additional options for FOFC (usually unnecessary)
- Supports arbitrary spacetimes

```
<mhd>
# General options
eos             = ideal      # EOS type
reconstruct = ppmx       # spatial reconstruction method
rsolver         = hlle       # Riemann solver to be used
dfloor          = 1.0e-10    # floor on density rho
gamma           = 2.0        # ratio of specific heats Gamma
fofc            = true

# DynGRMHD specific
dyn_eos         = ideal      # EOS type
dyn_error       = reset_floor # error policy
tfloor          = 1.0e-8
dthreshold      = 1.00       # Threshold for flooring
enforce_maximum = false

<adm>
```

Supported EOSs

- Ideal gas
$$e(n,T) = nT/(\Gamma - 1)$$
- Piecewise polytropic EOS:
$$e(n,T) = K_i n^{\Gamma_i}/(\Gamma_\mathrm{i} - 1) + nT/(\Gamma_\mathrm{th} - 1)$$
- Hybrid tabulated EOS:
$$e(n,T) = e_c(n) + nT/(\Gamma_\mathrm{th} - 1)$$
- Microphysical tabulated EOS:
$$e(n, Y_e, T)$$
- Very easy to add more via PrimitiveSolver

Tabulated EOS support

- PyCompOSE format; can use any EOS in the CompOSE database*
- Enable "not-quite-transcendental" functions for better performance (`mhd/use_NQT=true`)
- Set `mhd/eos_policy=compose`, `mhd/nscalars=1`, and `mhd/table=<table_name>`
- Many pgens (TOV, BNS, etc.) also require 1D table slice for initial data

*Your mileage may vary

BNS/BHNS initial data

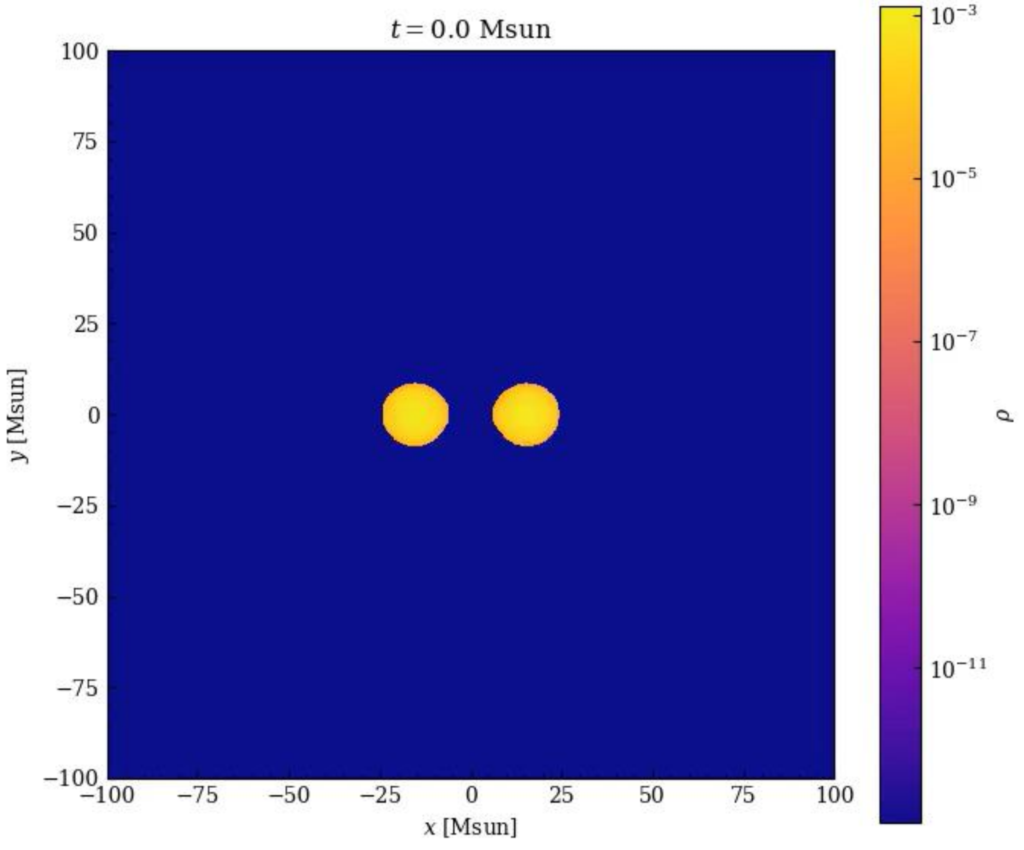
- Jim’s comment that AthenaK requires no external libraries comes with a caveat: all BNS runs require external initial data libraries
  - LORENE: classic pseudospectral code, only reliable for quasicircular BNS
  - FUKA: modern code with support for BBH, BHNS, BNS, arbitrary spins, etc.
  - Elliptica: another modern code with support for BBH, BHNS, BNS, arbitrary spins, etc.
  - sgrid: another BNS library, more modern than LORENE but with fewer features than Elliptica or FUKA

Example: LORENE BNS

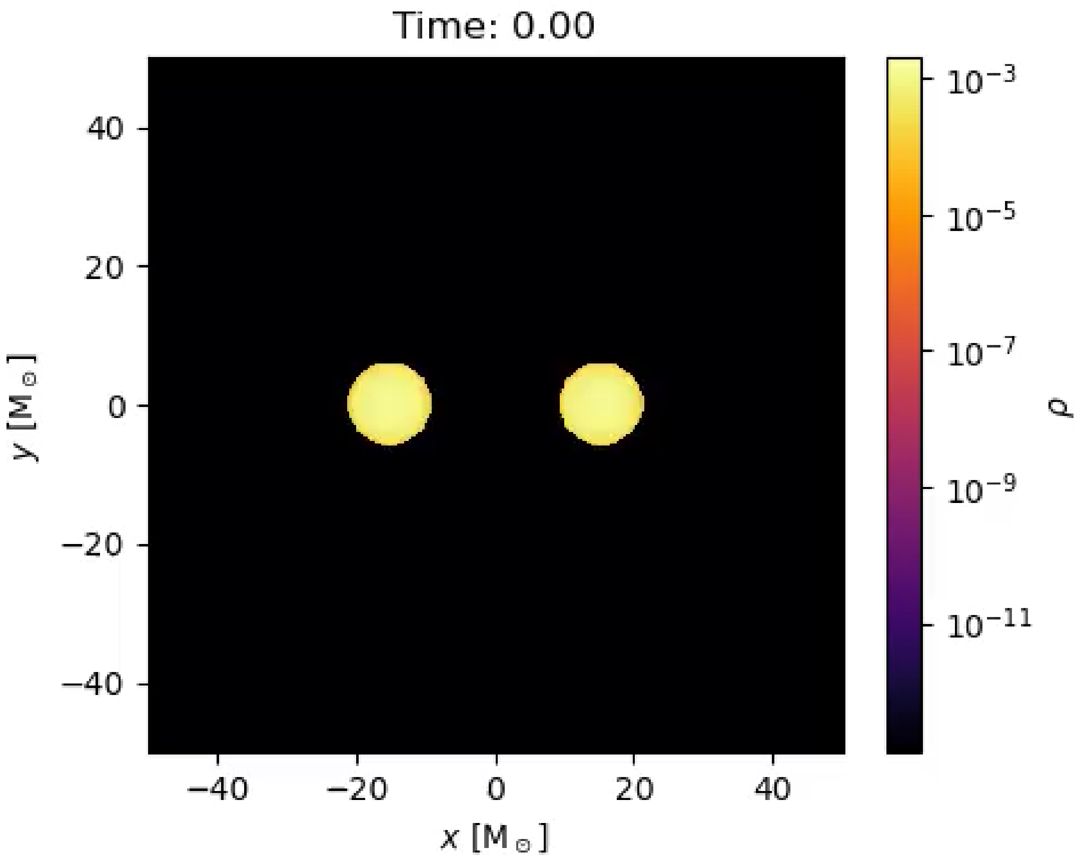
- Compile with `-DPROBLEM=dyn_grmhd/lorene/lorene_bns`
- Provide path to initial data
- Provide 1D EOS information if needed
- Set up BNS trackers
- Set up GW extraction

```
# <z4c>
# Gravitational wave extraction
nrad_wave_extraction = 3
extraction_radius_1  = 200
extraction_radius_2  = 400
extraction_radius_3  = 600
waveform_dt          = 2.5

# Neutron star tracker.
nco         = 2
co_0_type   = NS
co_0_x      = -15.236509409120682
co_0_radius = 10.0
co_1_type   = NS
co_1_x      =  15.236509409120682
co_1_radius = 10.0
tracker_mode = walk

<problem>
# Lorene options
initial_data_file = path_to_lorene.res # Path to LORENE ID
rho_cut = 1e-6 # Cut to atmosphere below this density
b_max = 1e12   # Magnetic field strength in G
r_0_current = 5.0 # Location of current loop for dipole field

# EOS options
kappa = 123.6133 # K for P = K*\rho^\gamma
```

Example (cont.)



Ideal Gas



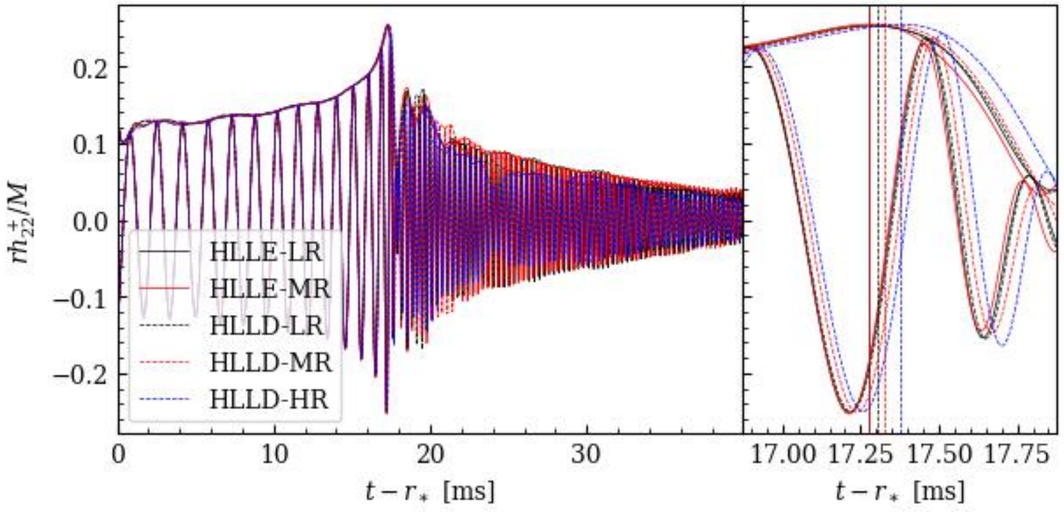
SFHo EOS (tabulated)

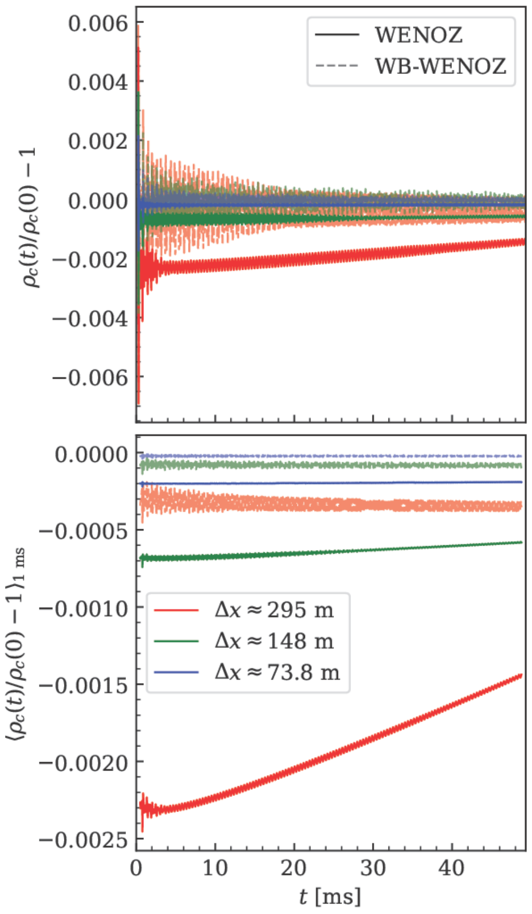
Some options for robustness

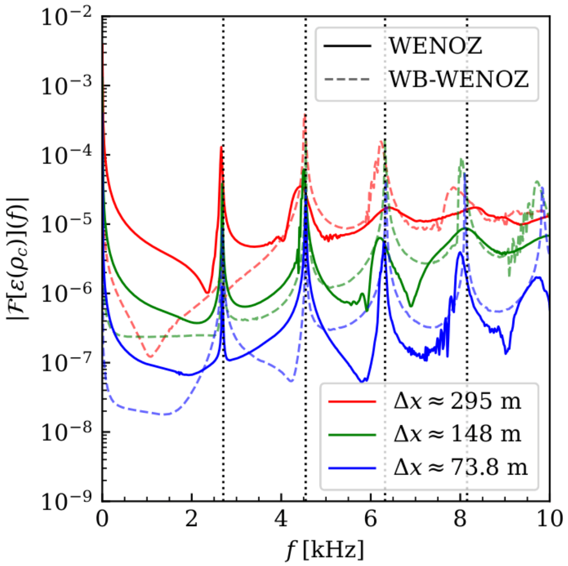
- Enable first-order flux correction: `mhd/fofc=true`
- Set a ceiling on W: `mhd/gamma_max=…`
- Enforce a maximum on B^2/D (DynGRMHD only): `mhd/max_bsq=…`
- Use FOFC to enforce relaxed maximum principle (DynGRMHD only, not usually needed): `mhd/enforce_maximum=true`, `dmp_M=…` ($M \geq 1$)

New developments coming soon

- HLLD for GRMHD
- Well-balanced scheme for hydrostatic systems







Summary

- AthenaK has two GRMHD solvers
    - HARM-like solver (CKS spacetimes)
    - Valencia/3+1 solver (arbitrary spacetimes)
- State-of-the-art methods for speed and stability
    - Robust C2P algorithm
    - First-order flux correction
    - Optionally read NQT-scaled tabulated EOS