Computing the Cosmos:

the role of numerical methods in astrophysics.

Jim Stone
Institute for Advanced Study
Princeton, NJ

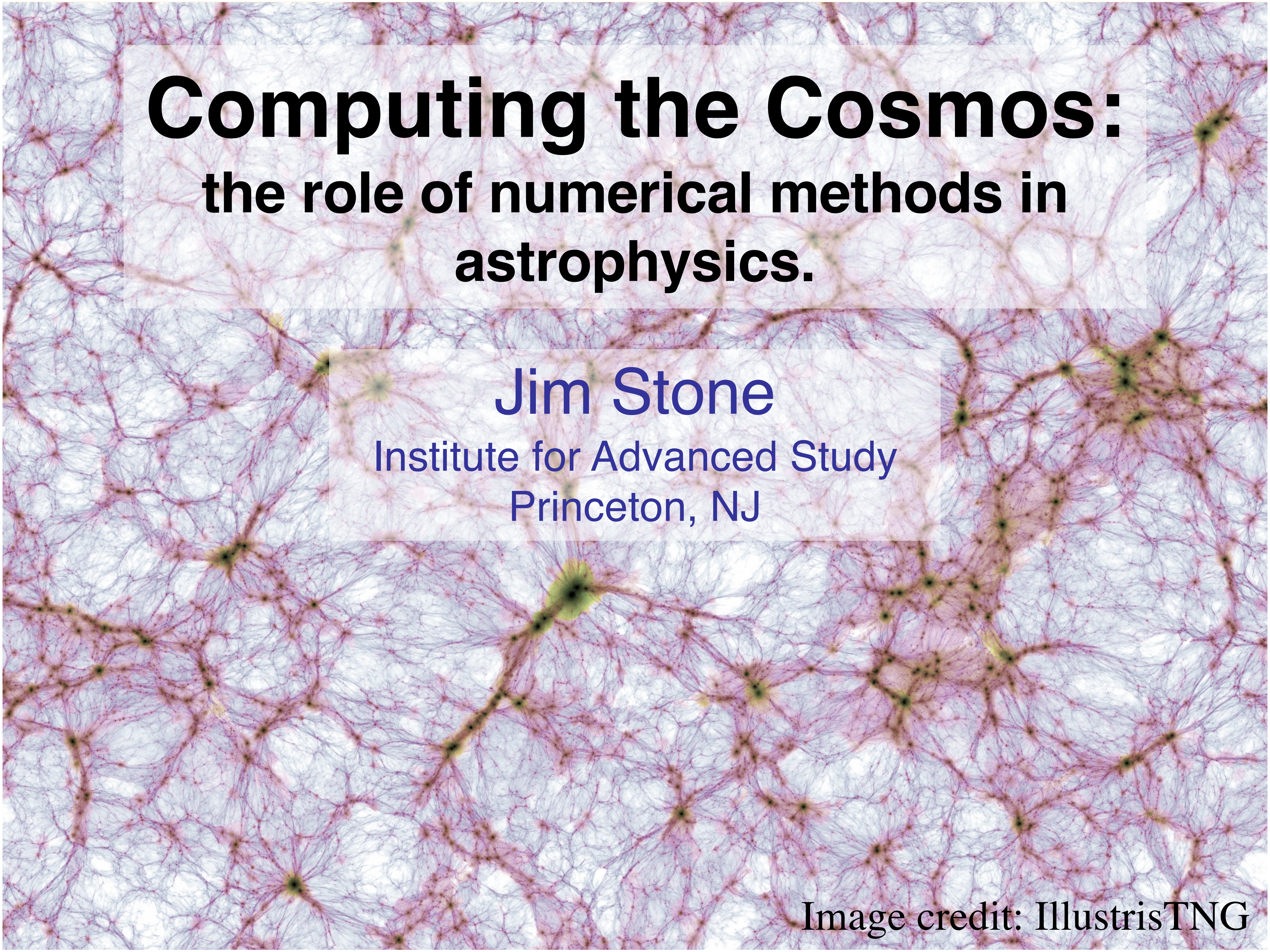
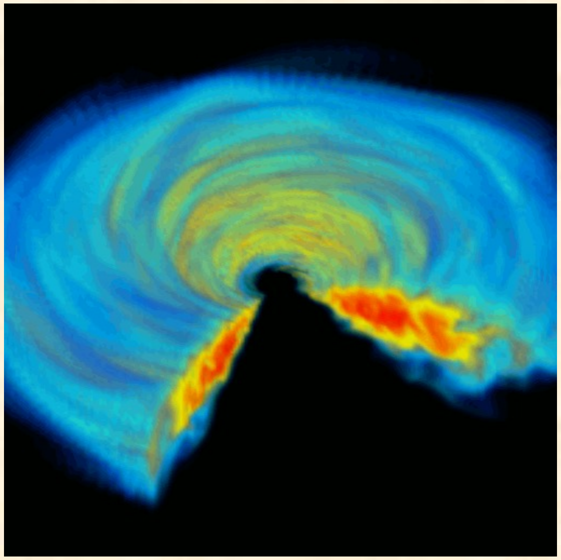


Image credit: IllustrisTNG

In remembrance of John Hawley
(1958-2021)

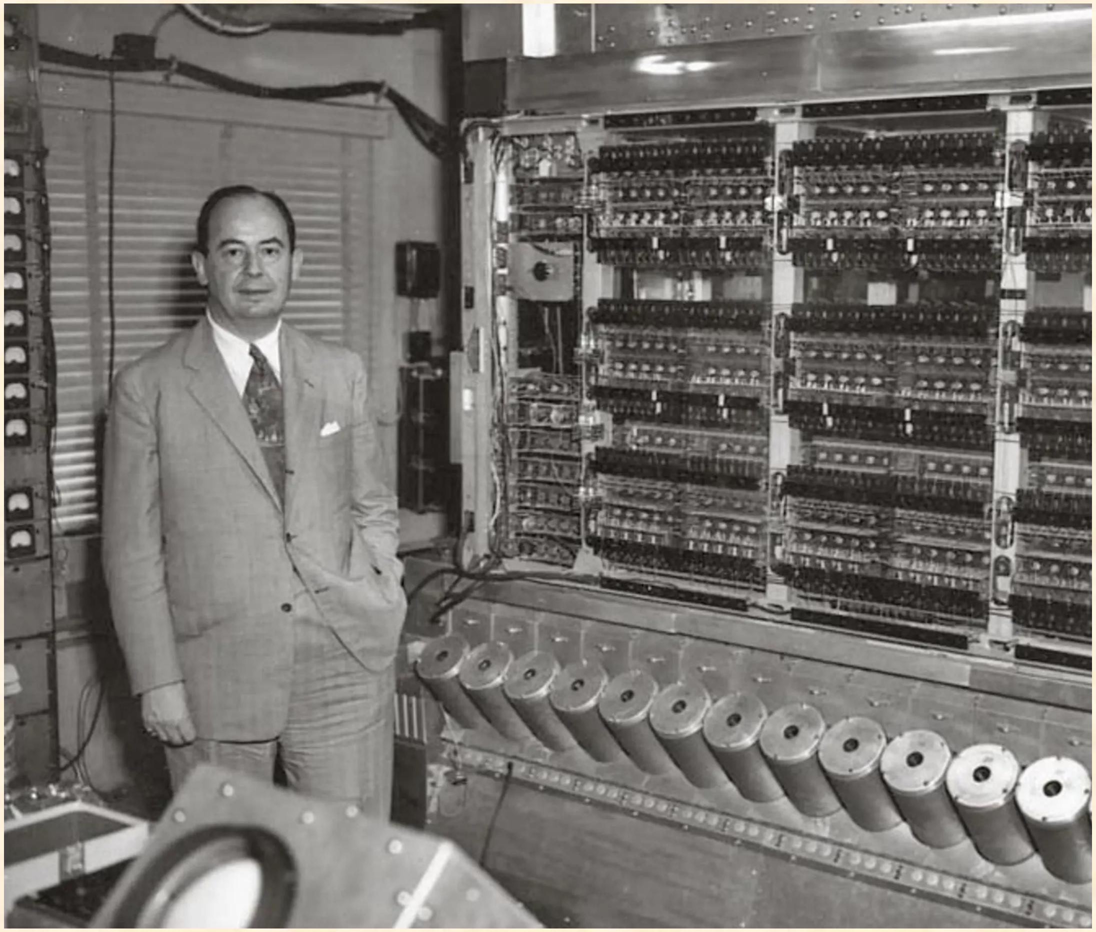


Hawley et al. 2001

Outline

- Rise of the machines
- Success stories
- Simulation versus reality
- Software challenges
- Future Directions

The Beginning: introduction of stored program computers (c1950)



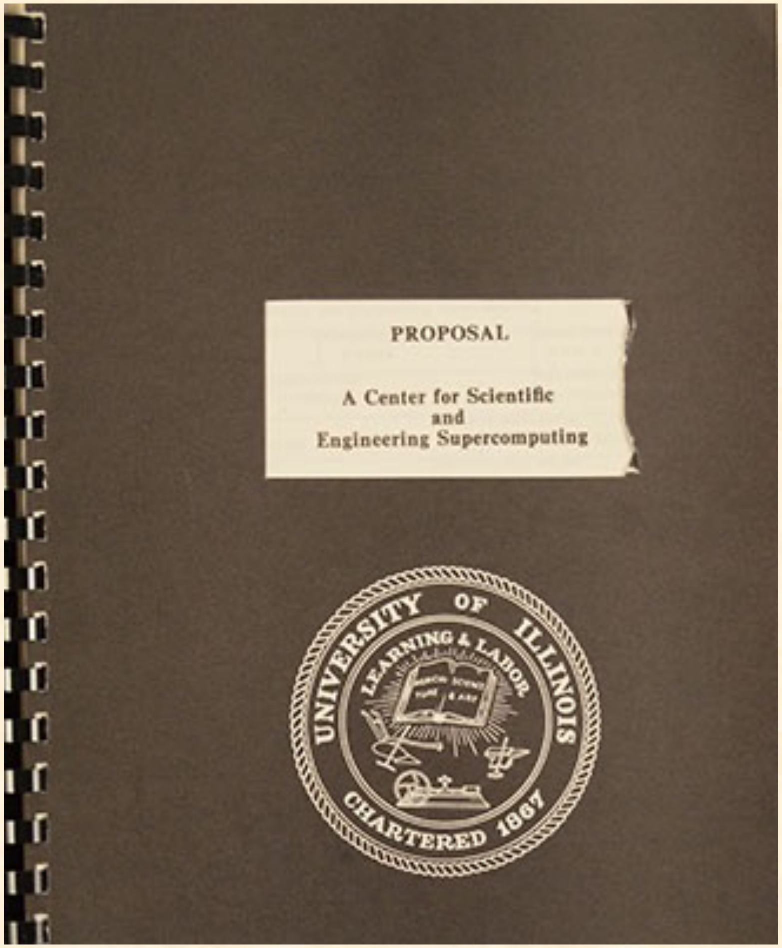
John von Neumann and the IAS computer (c1951);
see George Dyson Turing's Cathederal

ILLIAC I: first computer built and owned by a university (1952-1962)

University of Illinois archives

NCSA: scientific computing for all

Black proposal (1983)



PROPOSAL

A Center for Scientific
and
Engineering Supercomputing

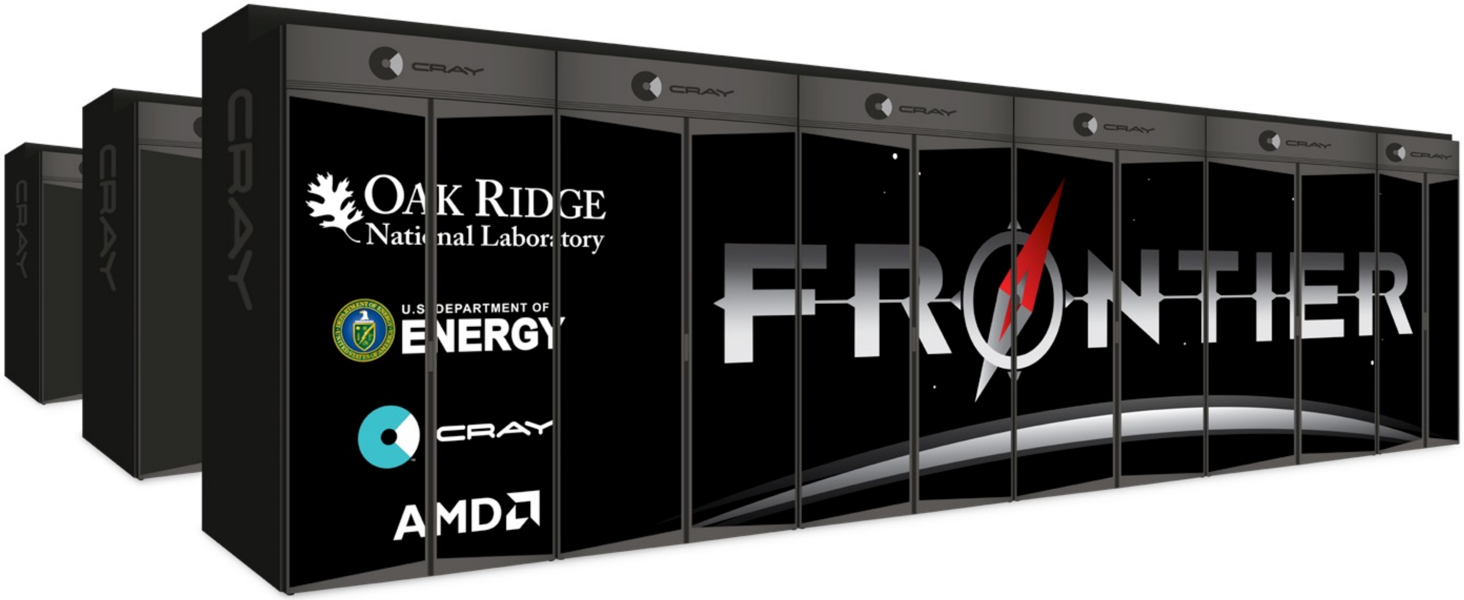
UNIVERSITY OF ILLINOIS
LEARNING & LABOR
CHARTERED 1867

Larry Smarr and Cray X-MP (1986)

[800 Mflops, 64MB]

Today: exascale has arrived

OLCF “Frontier” system, available July 2022



1.5 Eflops, about 10^9x faster than Cray X-MP

Even more important: better algorithms

In every discipline, the speed of algorithms that achieve a given level of accuracy has improved more than the speed of computers

- Simple example: Column 8 of *Programming Pearls*[1]
- Pattern matching code, 4 different algorithms (n^3, n^2, $n \log n$, n)

Problem size n	$1.3n^3$	$10n^2$	$47n \log_2 n$	48n
10^3	1.3 secs	10 msecs	0.4 msecs	0.05 msecs
10^4	22 mins	1 sec	6 msecs	0.5 msecs
10^5	15 days	1.7 min	78 msecs	5 msecs
10^6	41 yrs	2.8 hrs	0.94 secs	48 msecs
10^7	41 millenia	1.7 wks	11 secs	0.48 secs

The speed-up provided by the correct choice of algorithm far outweighs the speed-up provided by the hardware.

10^{13}x faster than $O(n^3)$ algorithm

---

[1] Jon Bentley, *Programming Pearls*

Success Stories: what is computation good for?

Computation had led to enormous progress on many problems:

- Stellar astrophysics
- Gravitational physics
- Accretion flows
- Core-collapse supernovae
- Pulsar magnetospheres
- Star and planet formation
- Large scale structure and galaxy formation
- Etc., etc., etc.

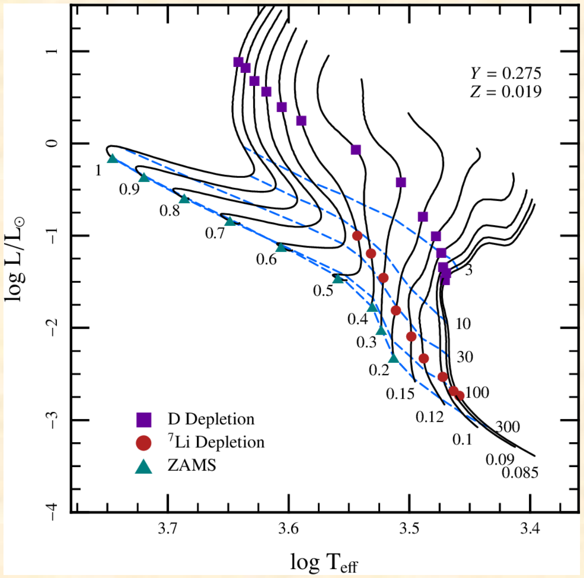
Success Stories:
what is computation good for?

Computation had led to enormous progress on many problems:

- Stellar astrophysics
- Gravitational physics
- Accretion flows
- Core-collapse supernovae
- Pulsar magnetospheres
- Star and planet formation
- Large scale structure and galaxy formation
- Etc., etc., etc.

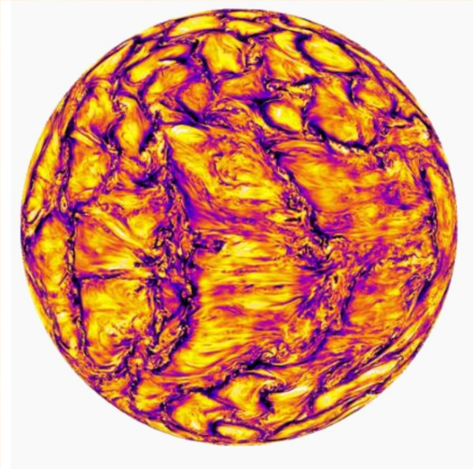
Stellar astrophysics

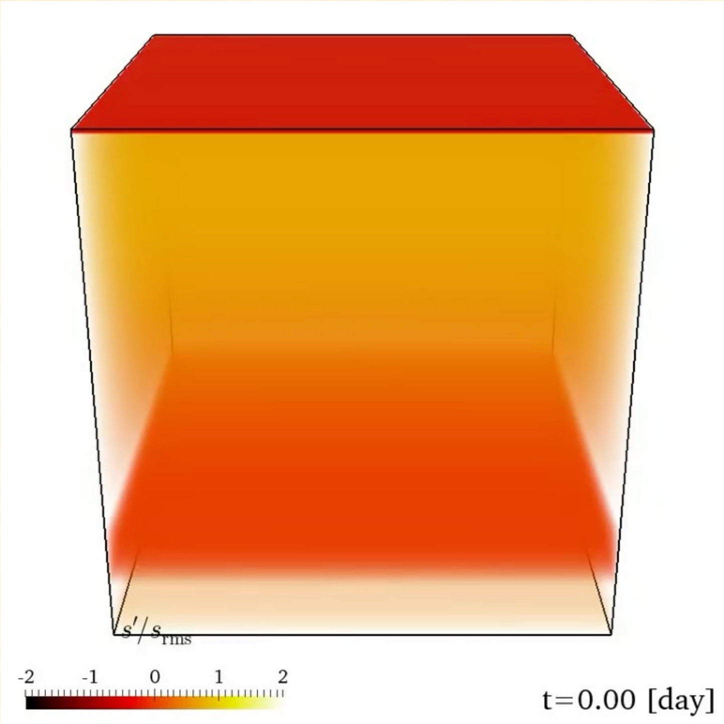
- Solving equations of stellar structure was important application for first generation supercomputers.
- Now, open-source stellar evolution codes like MESA widely available.



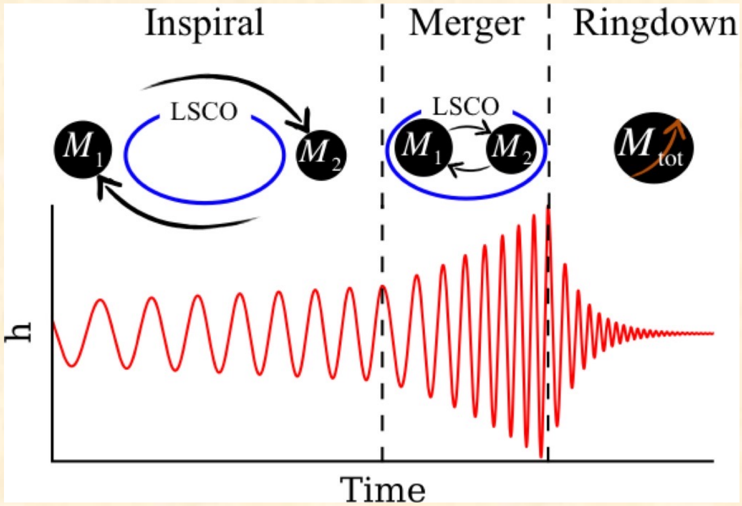
Stellar evolution tracks computed by MESA

- Stellar hydrodynamic calculations enable exploration of convection, rotation, and dynamo (Hotta et al 2016; 2019).





Black hole mergers



Since1950s numerical methods for solving Einstein equations have been developed to explore dynamical merger regime.

First calculation of head-on collision
(Smarr Sources of Grav. Rad. (1978), p268

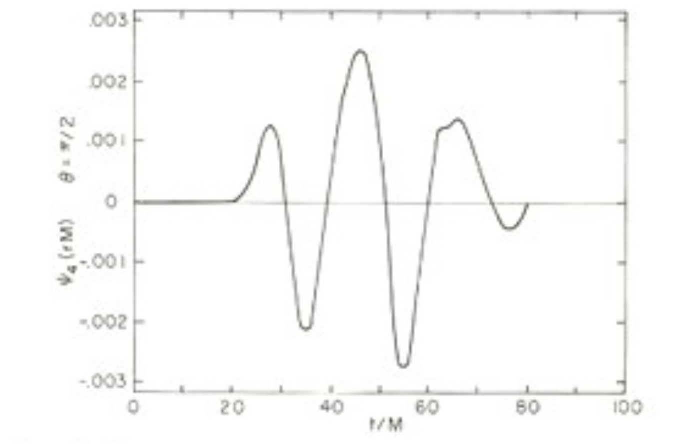
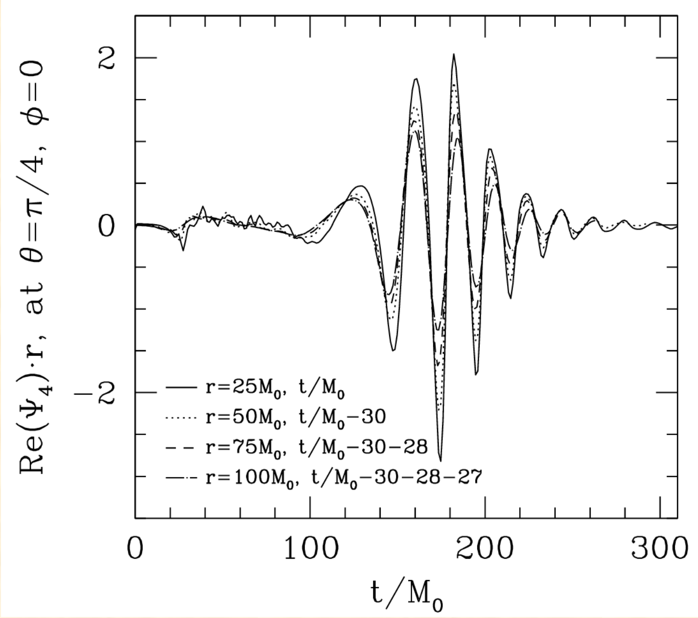
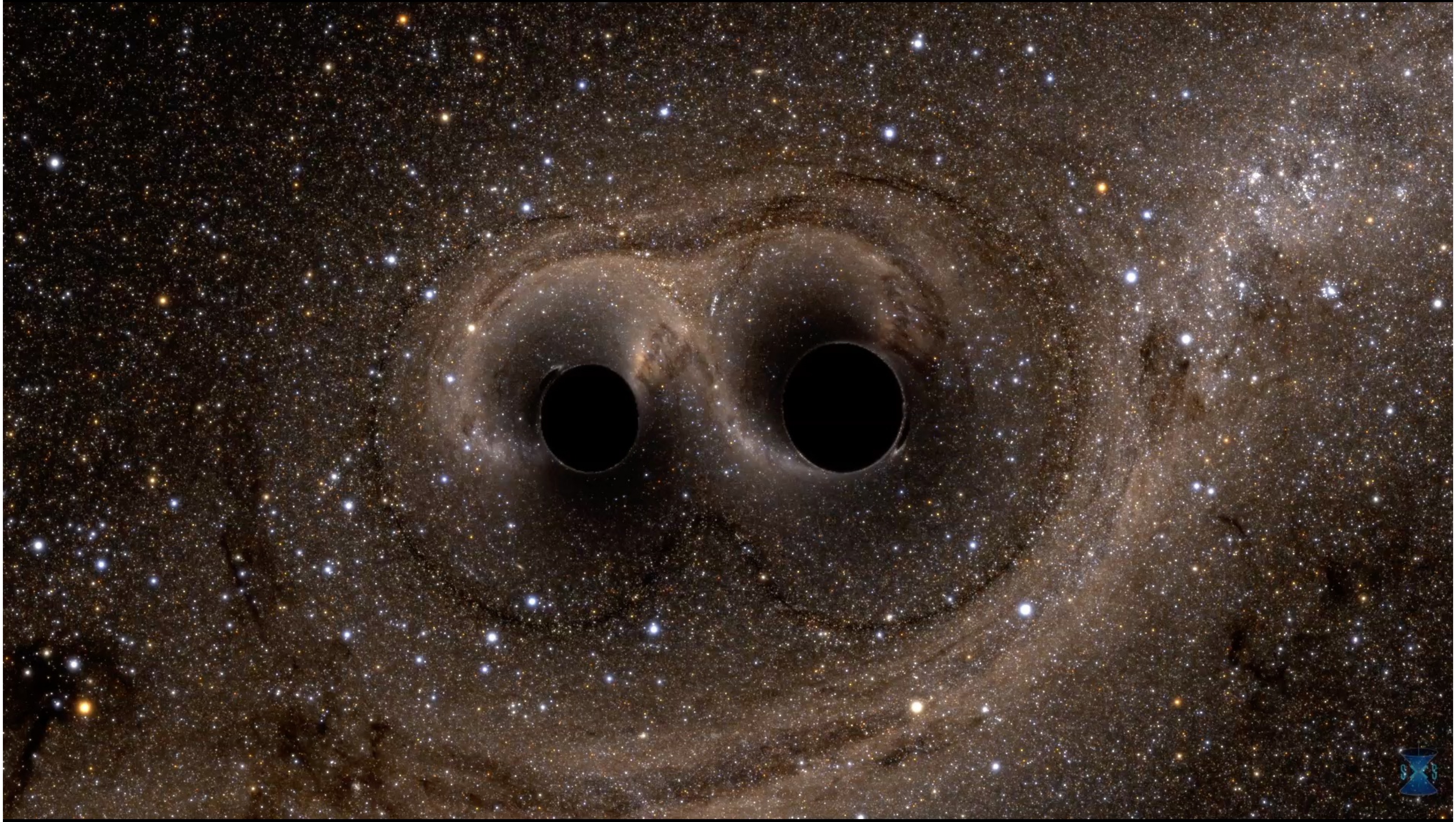


Figure 3. The curvature $\psi_4 \cdot rM$ in the equatorial plane crossing the 2-sphere at $r = 25M$ as a function of time. This is for the two black hole collision Run II.



First calculation of 3D inspiral
(Pretorius 2005)

Stunning visualizations convey far more information…



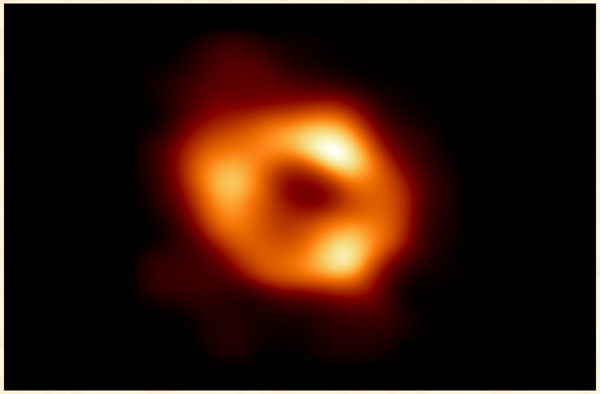
Movie from SXS collaboration: www.black-holes.org

Black hole accretion

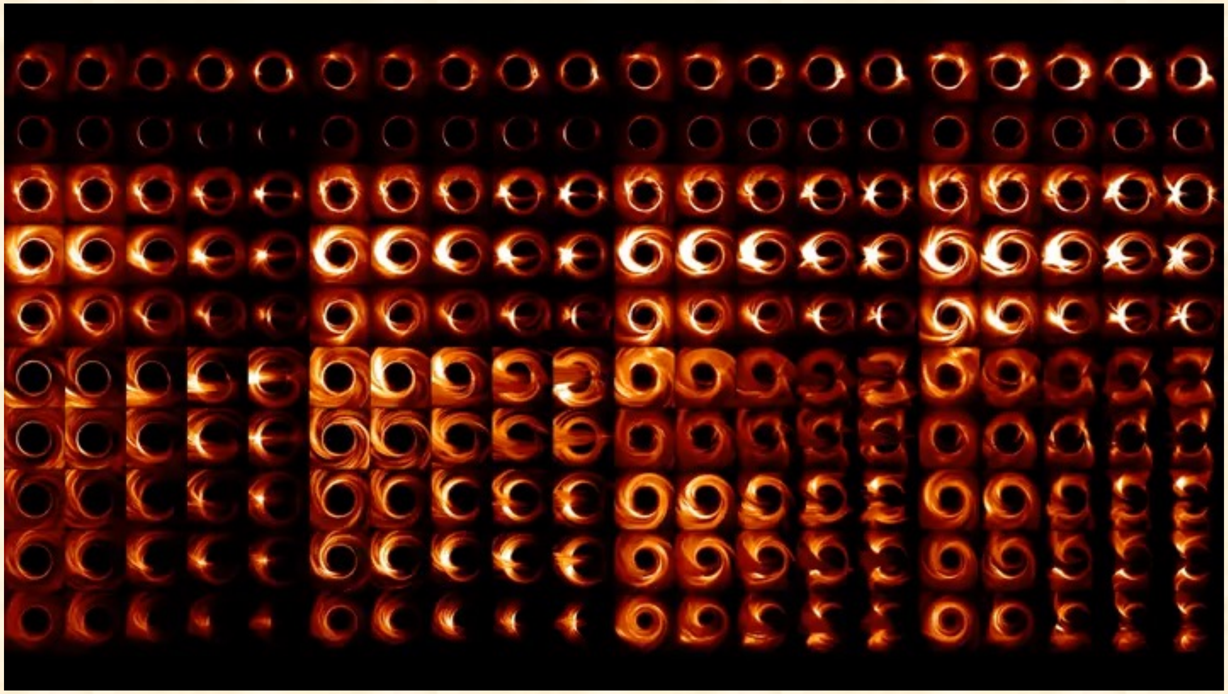
Most luminous sources are powered by accretion:

- Quasars and X-ray binaries in high state: GR + MHD + radiation transport
- Low-luminosity AGN (M87 and Sgr A*): GR + MHD + kinetic effects

To interpret EHT images, thousands of GR-MHD models of low-luminosity AGN have been run producing millions of synthetic images.



Event Horizon Telescope image of Sgr A*

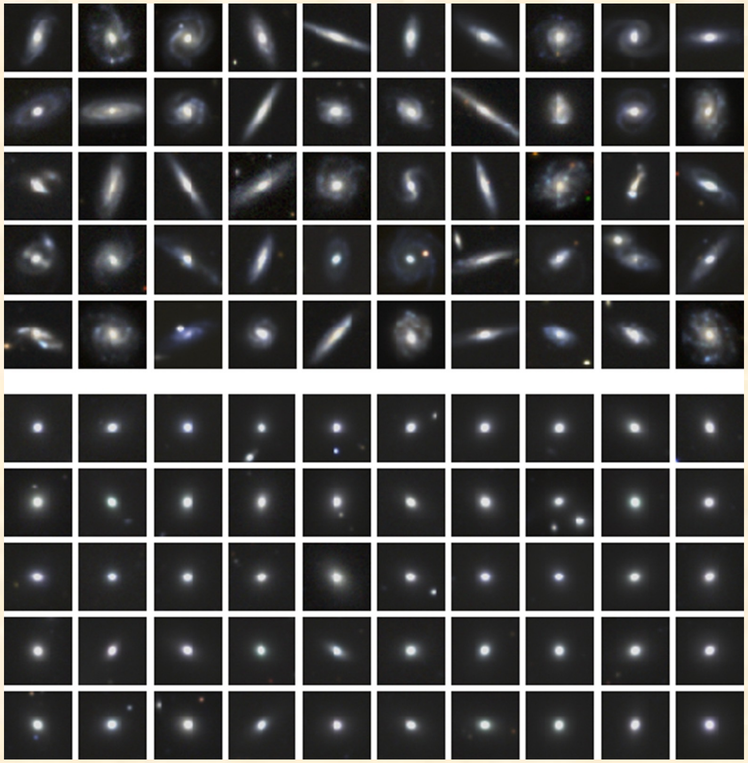


Visualization: Ben Prather, UIUC.
Image library: EHT Theory Working Group.

Let us not forget: Observation and data science

- Modern telescopes such as Vera Rubin Observatory will generate ~20TB data per night.
- Computational methods such as machine learning have emerged as important tools for analysis.

Galaxy classification in DESI



Khan et al. (2019)

Finding rare events, e.g. SN

Simulation versus reality

At most basic level, numerical methods are simply mathematical tools for finding approximate solutions to complex systems of equations, e.g.

- ODEs describing stellar structure
- Hyperbolic PDEs describing fluid dynamics
- Mixed system representing Einstein equations

Consistent, convergent, and robust algorithms developed using tools of numerical analysis.

With appropriate algorithms, focus is generating and understanding solutions as applied to astrophysical systems.

reality

At most basic level, numerical methods are simply mathematical tools for finding approximate solutions to complex systems of equations, e.g.

- ODEs describing stellar structure
- Hyperbolic PDEs describing fluid dynamics
- Mixed system representing Einstein equations

Consistent, convergent, and robust algorithms developed using tools of numerical analysis.

With appropriate algorithms, focus is generating and understanding solutions as applied to astrophysical systems.

Simulation

In some cases, astrophysical systems involve uncertain or complex physics that cannot be modeled directly

- Galaxy formation
  - Sub-grid models for star formation and AGN feedback
- Compact objects
  - Equation of state for nuclear matter, neutrino flavor mixing
- Radiation hydrodynamics
  - Non-LTE opacities for gas in non-equilibrium ionization

In these cases, numerical methods enter realm of *simulation*.

Must balance what can be learned from solutions of

- model system represented by exact equations (numerical experiments)
- approximate system with parametrized physics (simulations)

(These same challenges face analytic theory.)

Tremendous challenge to develop physics-based sub-grid prescriptions.
But won't discuss further here.

Verification and validation

Important to confirm that numerical methods are accurate!

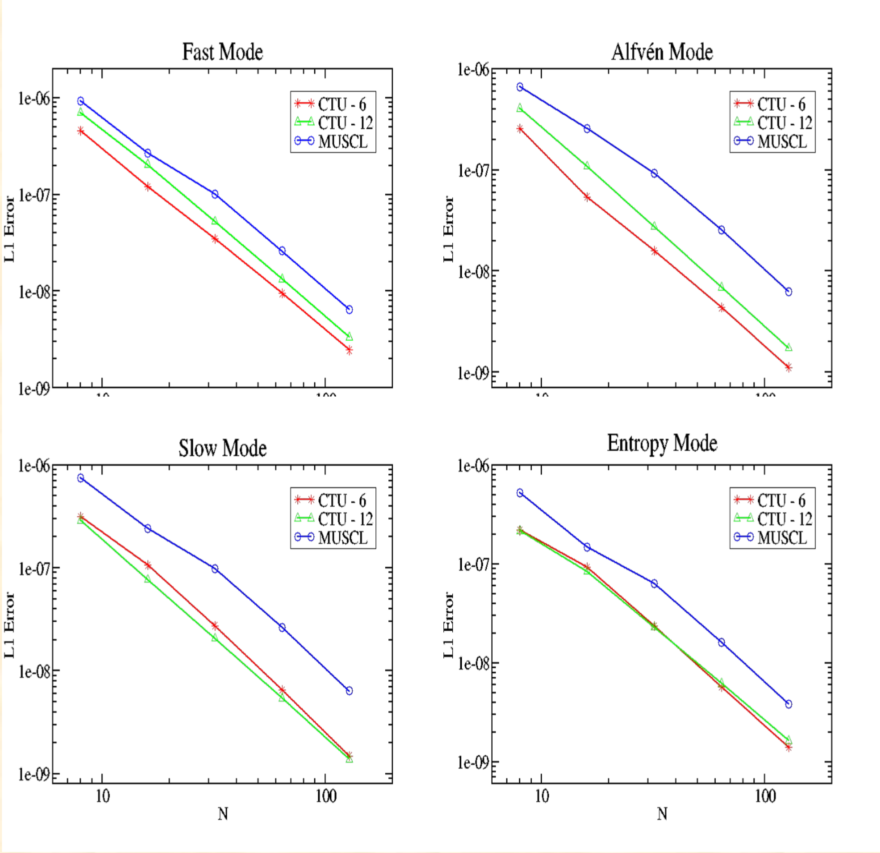
Methods must continuously be tested against:

- analytic solutions (verification),
- experiments or observation (validation),
- other methods,
- the application itself.

Testing against known solutions

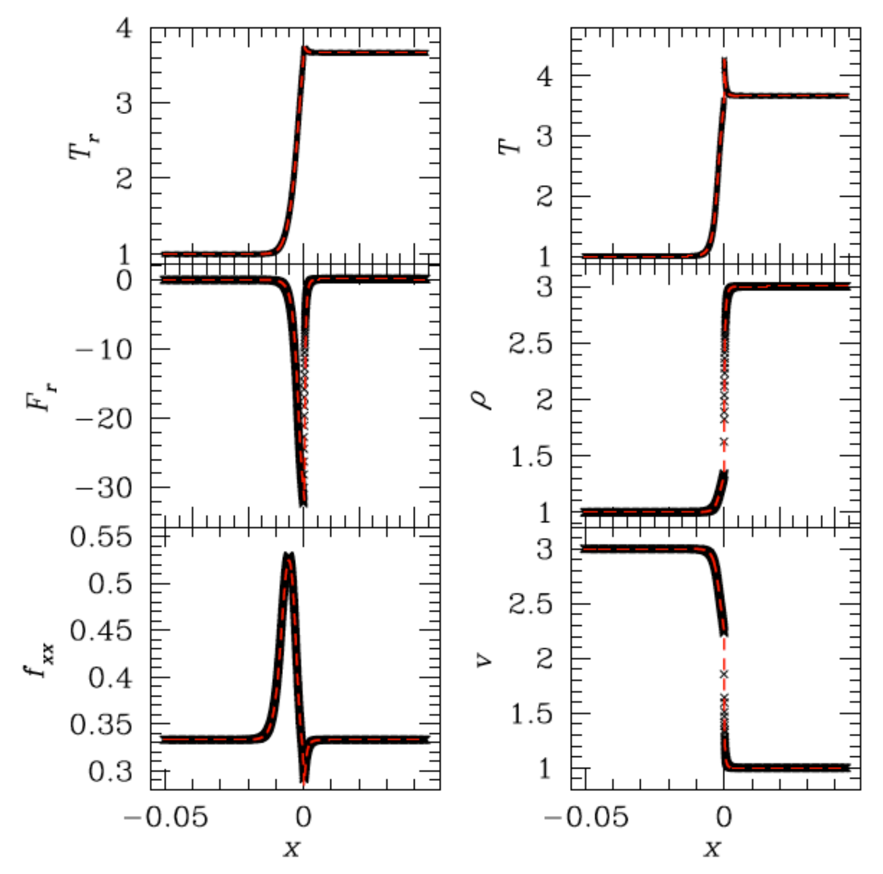
Challenge is finding such solutions.

MHD:
linear wave convergence



Stone+ 2012

Radiation hydrodynamics:
Structure of radiation shock

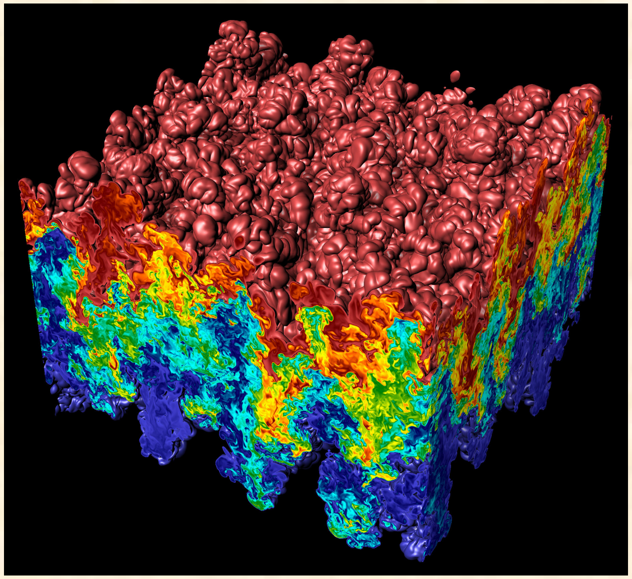


Jiang+ 2014

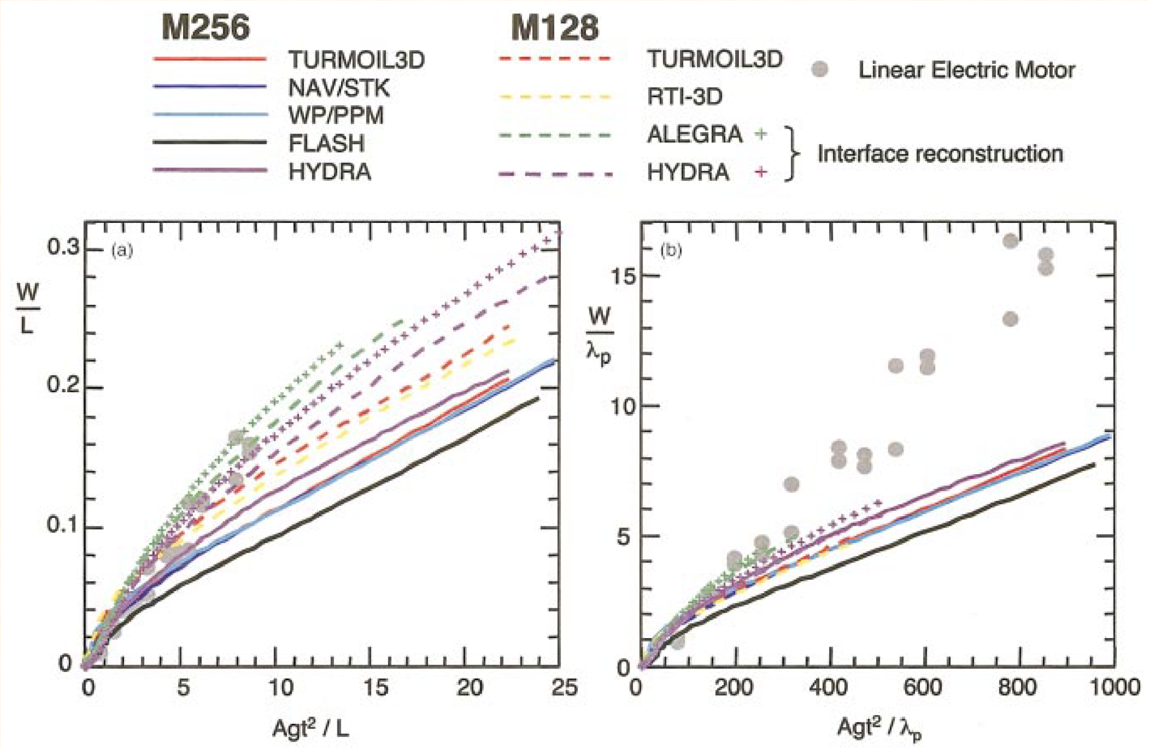
Red line=reference sol'n.
Points=numerical sol'n

Testing against experiments

For example, hydrodynamic Rayleigh-Taylor instability

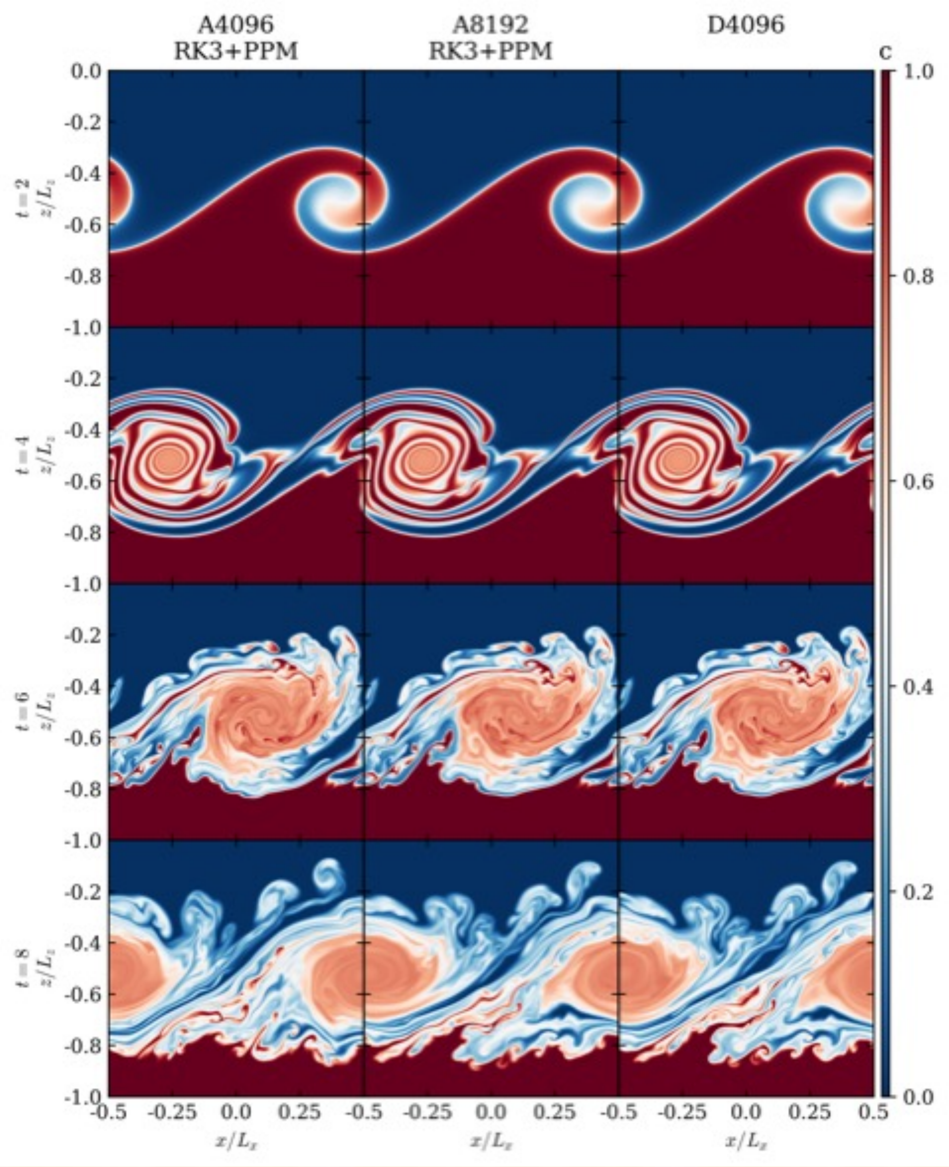


Width of mixing layer



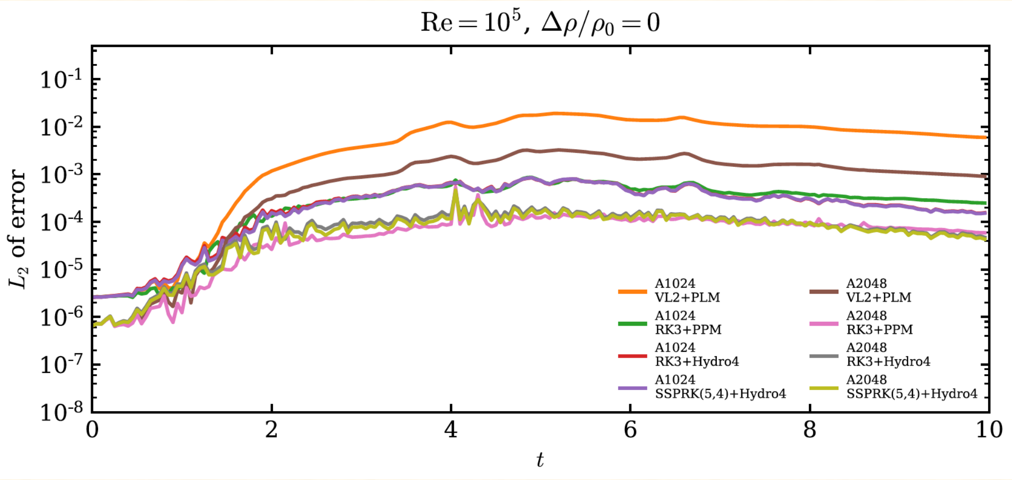
Dimonte et al (2004)

Testing against other codes



Comparison of Athena++ and Dedalus on non-linear Kelvin-Helmholtz instability with explicit dissipation.

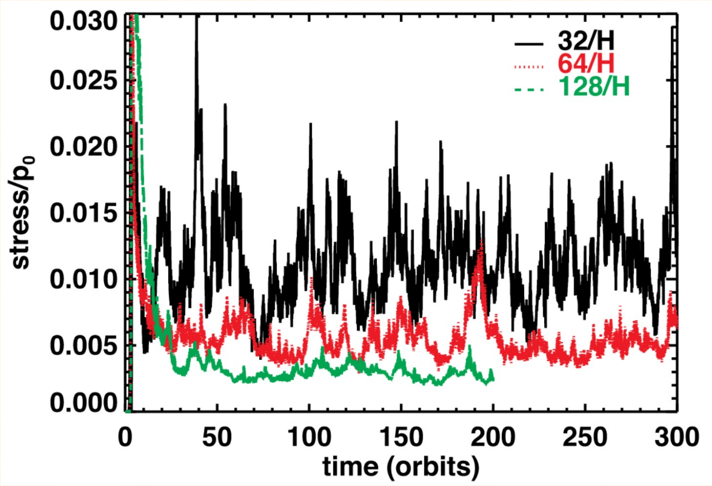
Athena++ reproduces spectral code results at 2x resolution and 1/2 the cost.



Testing against the application

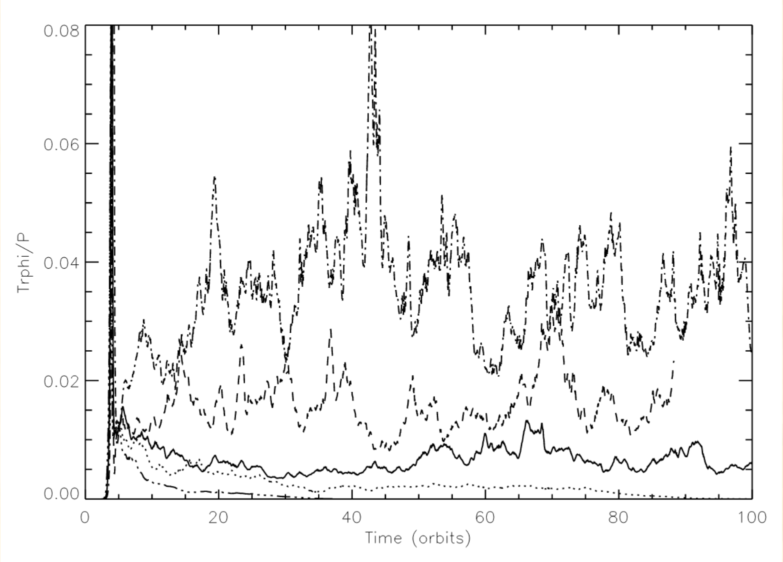
Can always test convergence of properties of solution. *Sometimes, this reveals new physics.*

For example, Fromang & Papaloizou (2007) found stress in zero-net-flux unstratified shearing-box simulations of the MRI decreases with resolution.

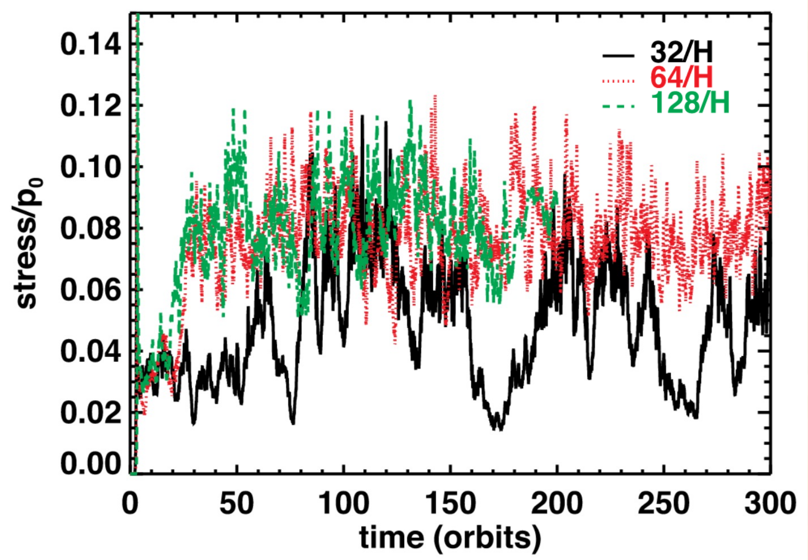


Fromang et al (2007) showed that adding explicit resistivisty and viscosity leads to converged solutions, and that the amplitude of MRI turbulence depends on the magnetic Prandtl number.

Re_M=12,500, $P_m = \nu/\eta$ = 1,2,4,8,16

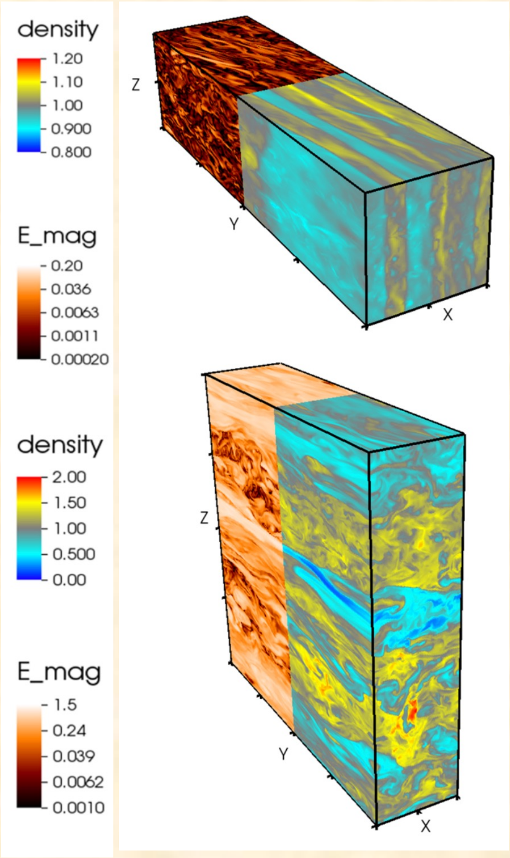


However, increasing aspect ratio L_Z/L_X gives convergence, *even without explicit dissipation* (Shi et al. 2016)



Result related to properties of nonlinear MHD dynamo driven by MRI, similar to shear-driven dynamos (Yousef et al. 2008).

We don’t yet understand the details…



Software challenges

Developing software is a significant challenge due to:

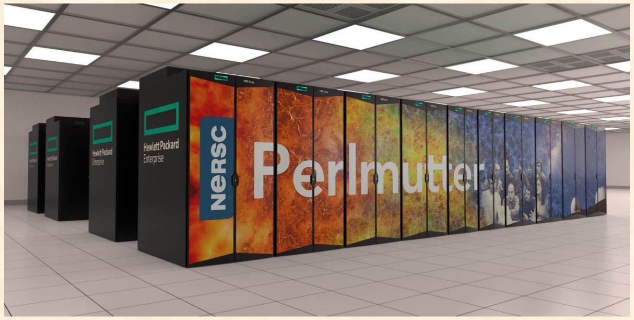
- Heterogeneous architectures
- Complex algorithms
- Data volumes

Heterogenous architectures

Largest open-science computers adopt a wide variety of designs

Fugaku (Japan):
1.6M ARM processors, 48 cores each

Summit (USA)
10k IBM Power9 processors +
28k NVidia V100 GPUs



Perlmutter (USA)
200k AMD EPYC cores +
6k NVidia A100 GPUS

Frontier (ORNL) will use AMD GPUs, and Aurora (ALCF) will use Intel GPUs (cannot be programmed with CUDA).

Performance portability

Most application scientists do not have the resources to support different versions of the same code that runs on different machines.

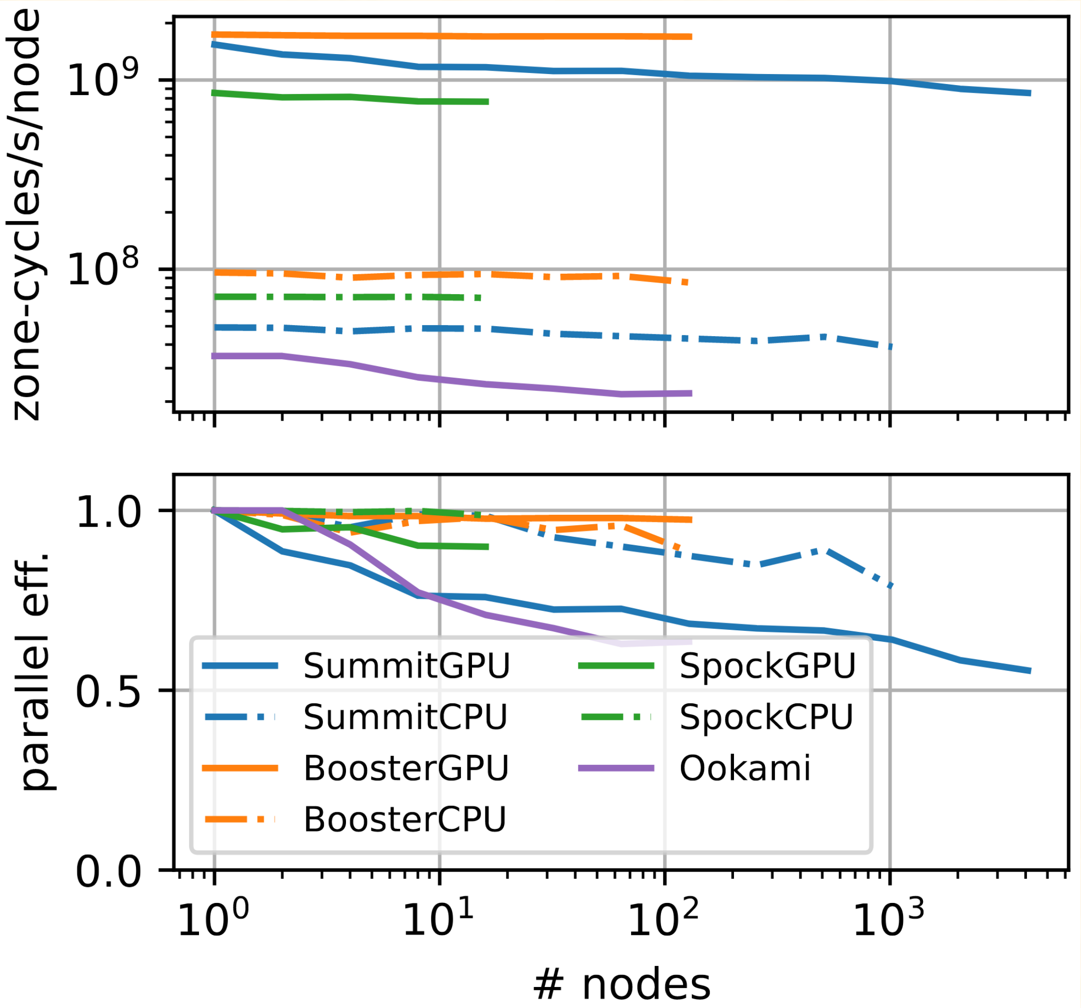
Need a single programming environment supported by all vendors, e.g.

- OpenMP
- OpenCL
- Kokkos

Kokkos (Trott et al. 2021):
C++ library that provides abstractions for multiple execution and data spaces. Interfaces with CUDA, HIP, SYCL, HPX, OpenMP, and C++.

Parthenon: Open source implementation of Athena++ AMR framework in Kokkos

Weak scaling of Athena++ hydrodynamics with Parthenon:



Summit GPU = Nvidia V100
Booster GPU = Nvidia A100
Spock GPU = AMD MI100
Ookami = Fujitsu ARM

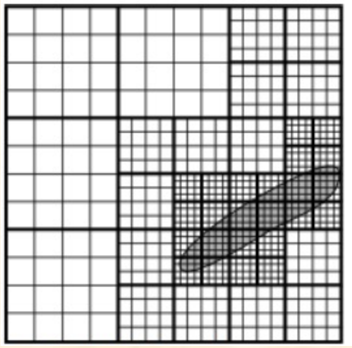
Grete et al (2022)

https://github.com/lanl/parthenon

Modern codes are increasingly complex

They involve sophisticated algorithms like

- adaptive mesh refinement



e.g. block-based
AMR organized into
oct-tree

- dynamic task scheduling
- hybrid parallelization (MPI and OpenMP)
- solvers for mixed equations (elliptic, parabolic, hyperbolic)

It is becoming increasingly difficult for graduate students to write code from scratch as part of their thesis.

Open-source codes are becoming increasingly important, but must still encourage innovation.

Future Directions

Future progress will likely be made in a number of directions:

- Ultra-high-resolution
- New physics
- Better algorithms
- Novel approaches

Ultra-high resolution

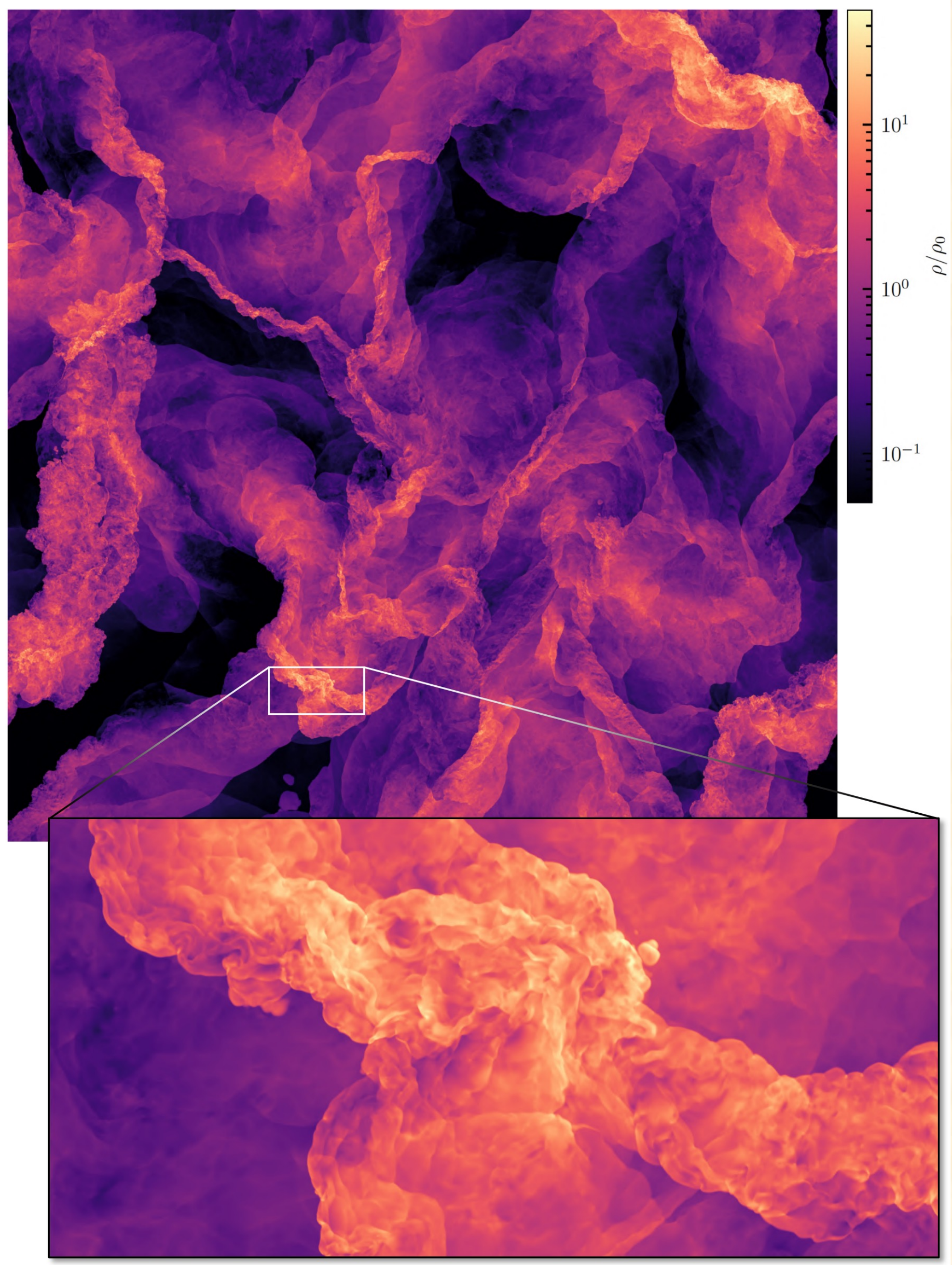
Example: driven hydrodynamic turbulence with a resolution of $10{,}048^3$ to resolve the sonic scale: $l_s \sim L/M^2$

(some theories of gravitational fragmentation predict only gas inside sonic scale can collapse).

Required 50M core hours spread over roughly 1 year to run.

Could now be run in a few hours *but this requires codes than run at exascale.*

Federrath et al (2022)

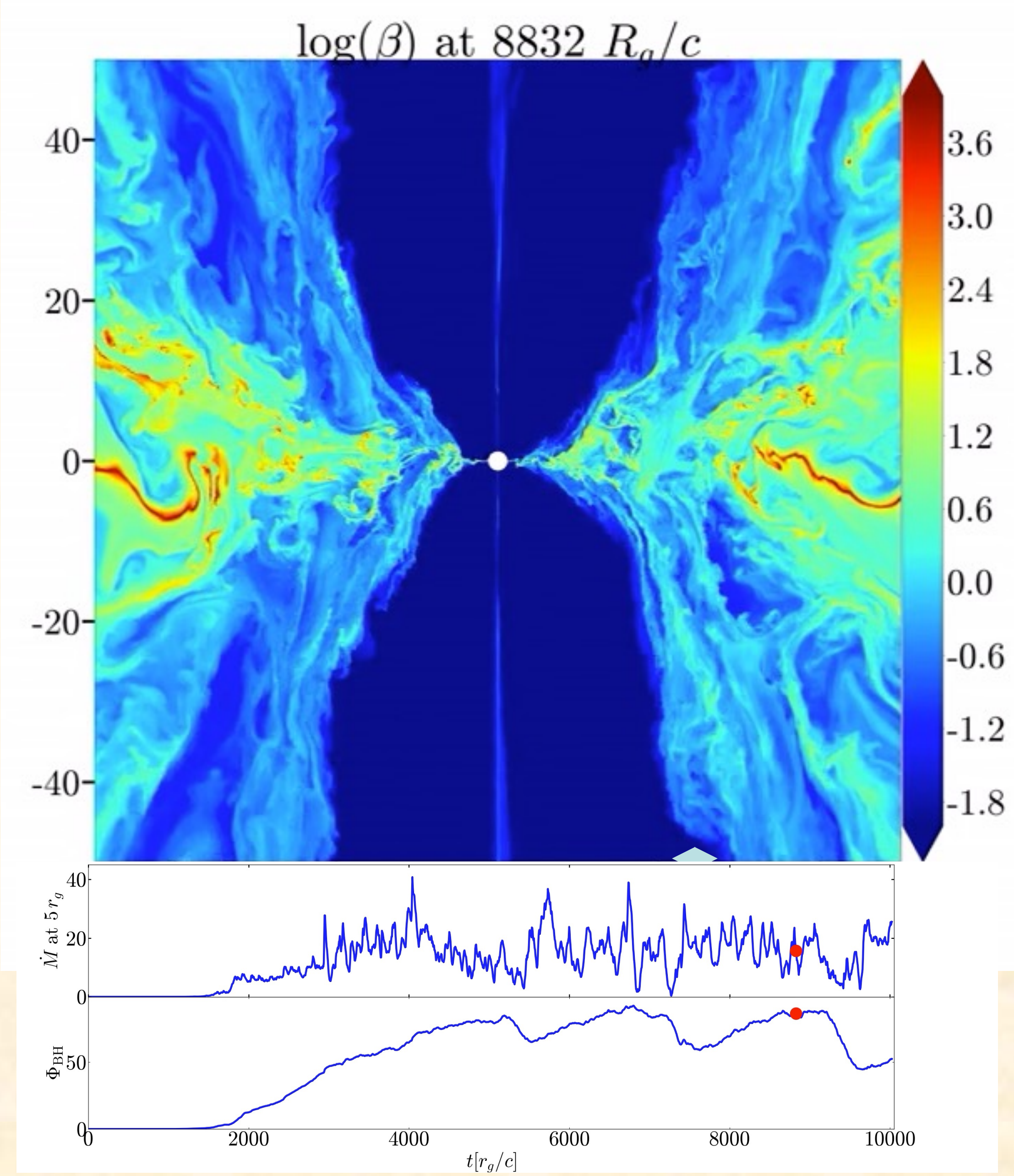


Zooming into Event Horizon with GRMHD

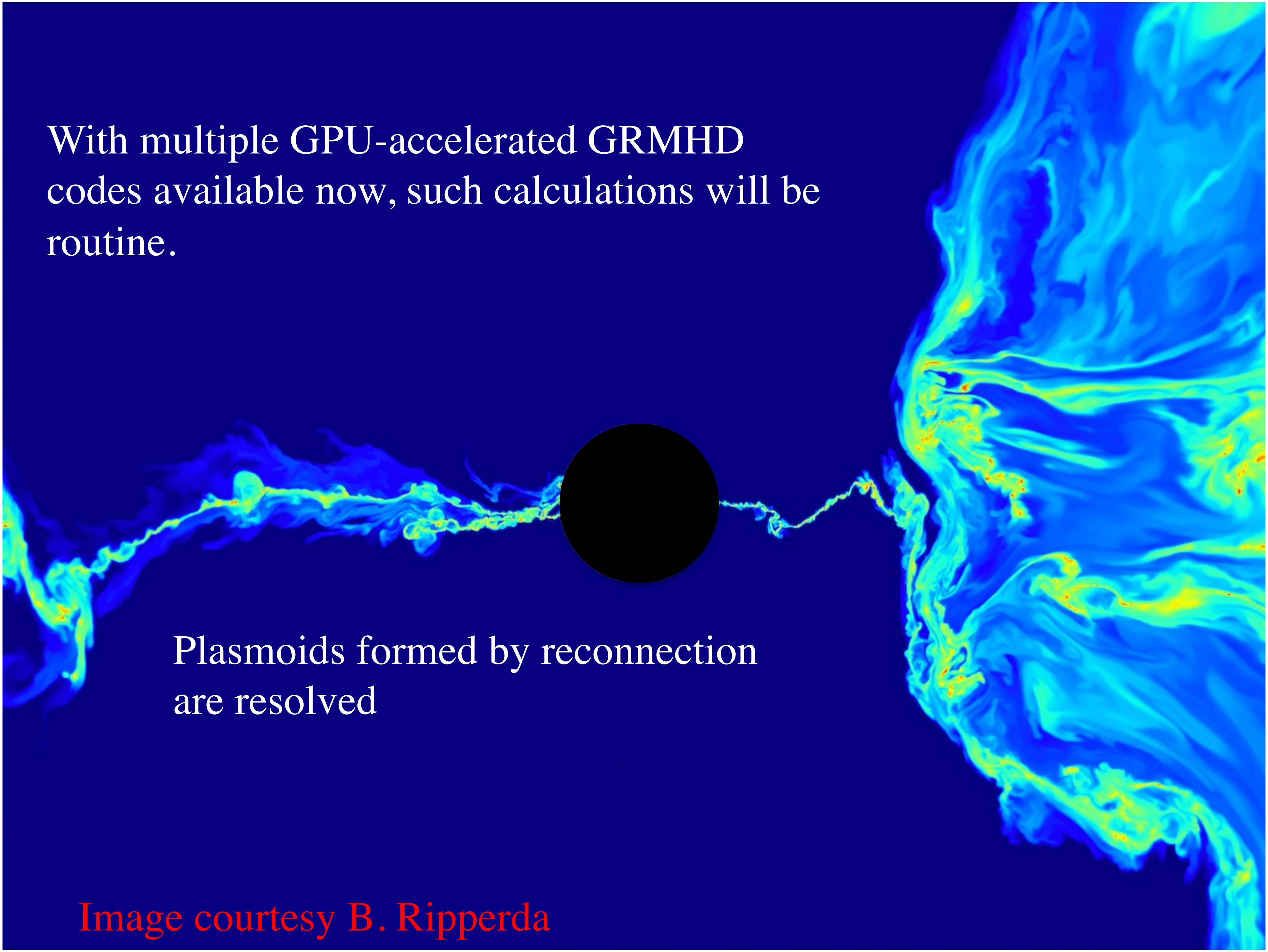
Computed using GPU-accelerated HAM-R code.

5000x2500x2500 effective resolution at highest level of refinement.

Made possible by one-billion core hour allocation on GPU-accelerated DOE machine.



[**Ripperda** et al, ApJL, 2022]



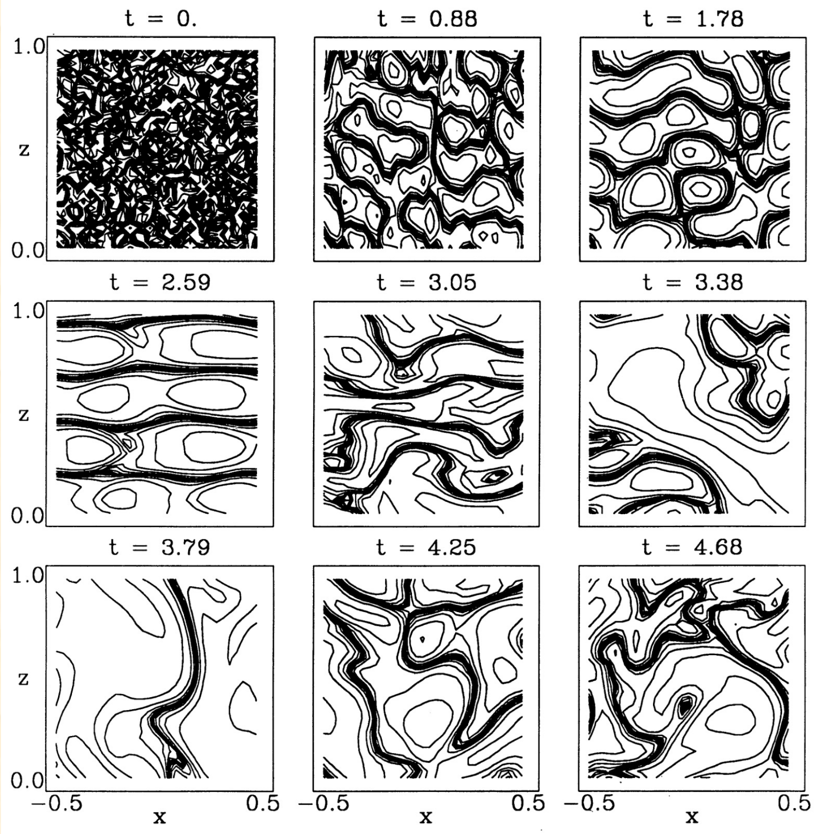
With multiple GPU-accelerated GRMHD codes available now, such calculations will be routine.

Plasmoids formed by reconnection are resolved

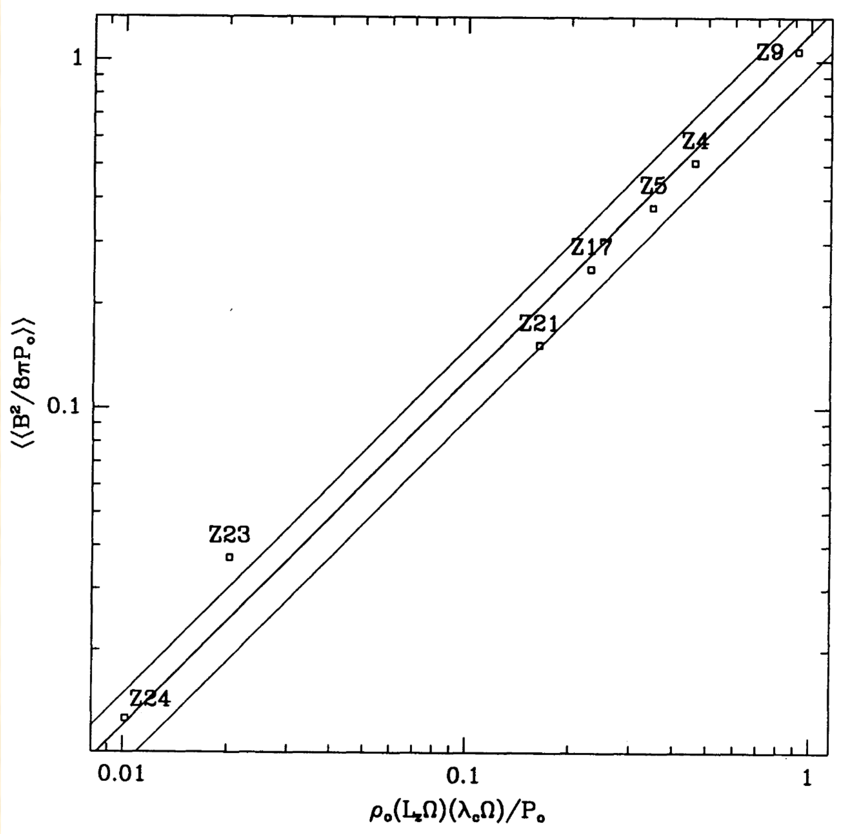
Image courtesy B. Ripperda

Not everything requires ultra-high resolution

Example: Original shearing box simulations of MRI captured the essential results at a resolution of 31x63x31 (Hawley, Gammie, Balbus 1995)



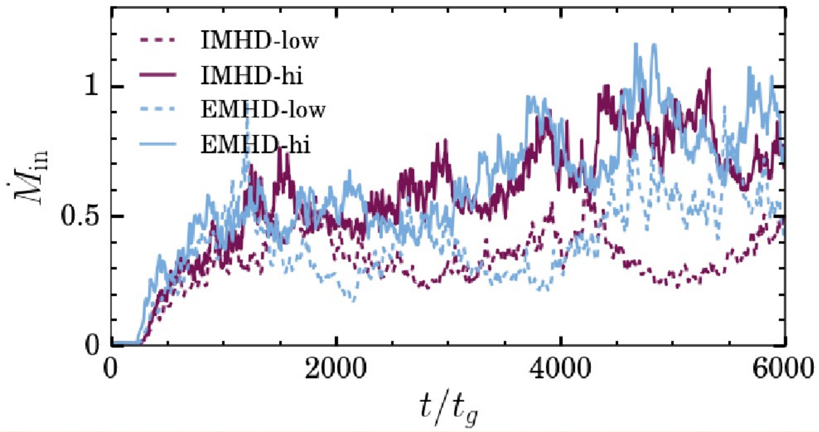
MHD turbulence in the nonlinear regime



Saturation predictor with mean B_Z

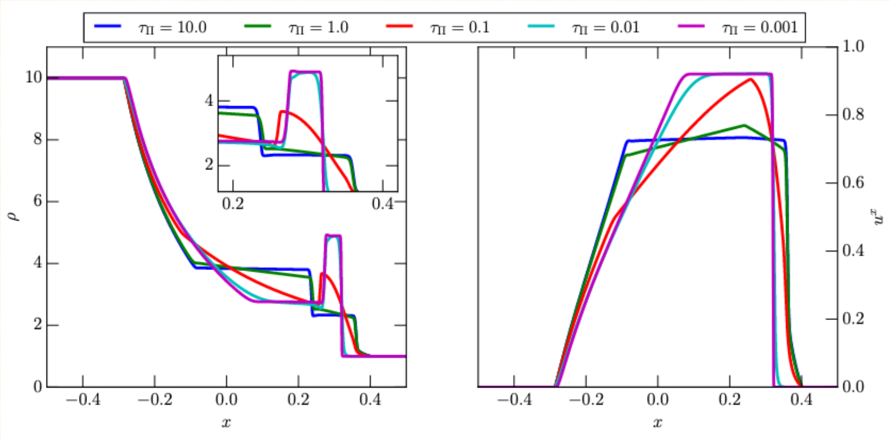
Additional physics

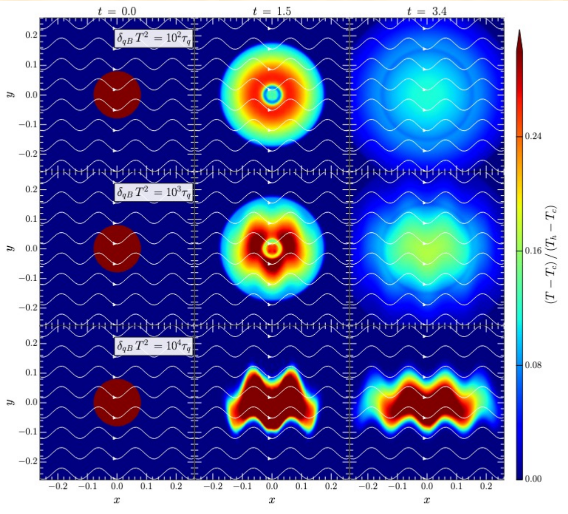
Example: Adding kinetic effects (viscosity, heat conduction) to GR-MHD models of low-luminosity accretion flows.



First results from Foucart et al. (2018) show kinetic effects do not make substantive difference. Results use `grim` code of Chandra et al (2017)

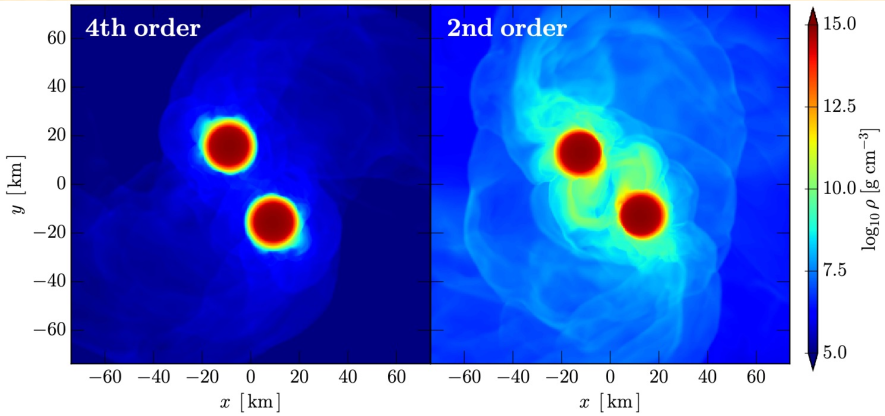
Still, important to develop improve methods for kinetic MHD in relativistic two-temperature plasmas, e.g. (Most & Noronha 2021).

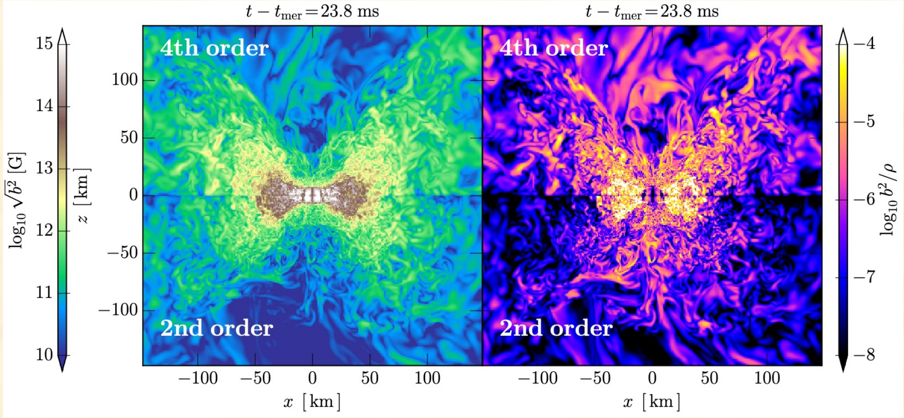




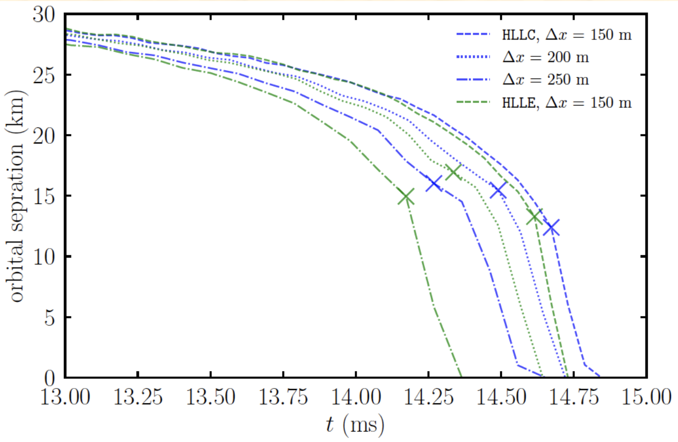
Improved algorithms (e.g. mergers)

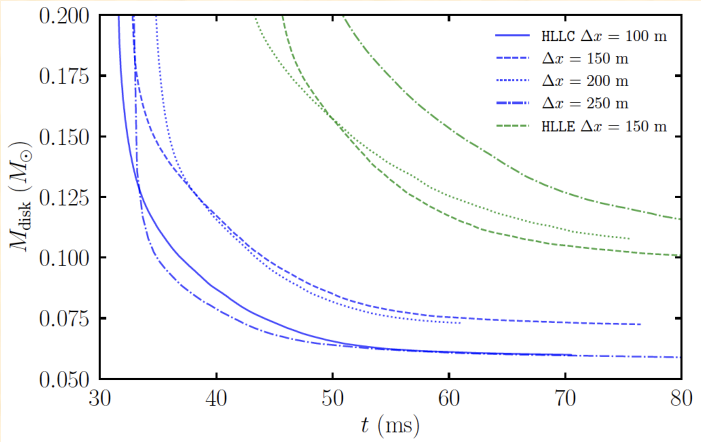
Higher-order methods for fluid dynamics (Most et al. 2020)





More accurate Riemann solvers (White et al. 2016; Kiuchi et al. 2022)





Significant differences between HLLE and HLLD

Improved algorithms (radiation)

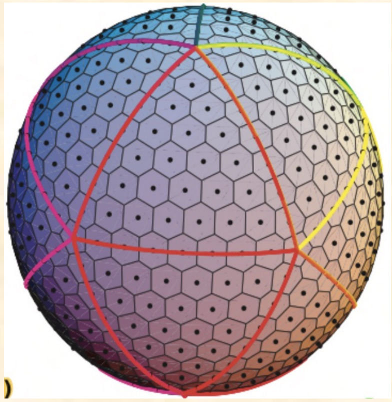
Solving radiation transport equation directly in curved spacetime challenging.

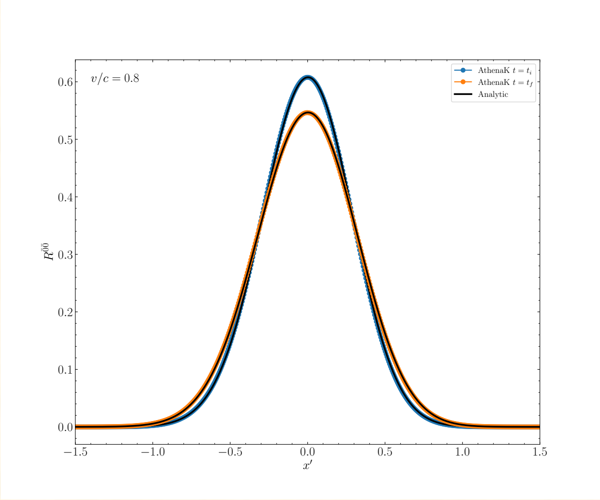
Most work to date has adopted moment methods with an assumed closure.

Recently, both Monte-Carlo and finite-volume methods have been developed.

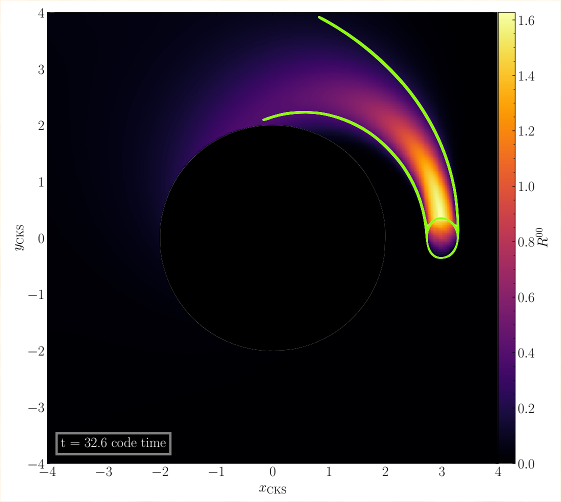
For example, in FV method of White et al, (in prep)

- Specific intensity is discretized in angle using geodesic mesh.
- Radiation-matter coupling included via implicit solution in co-moving frame.





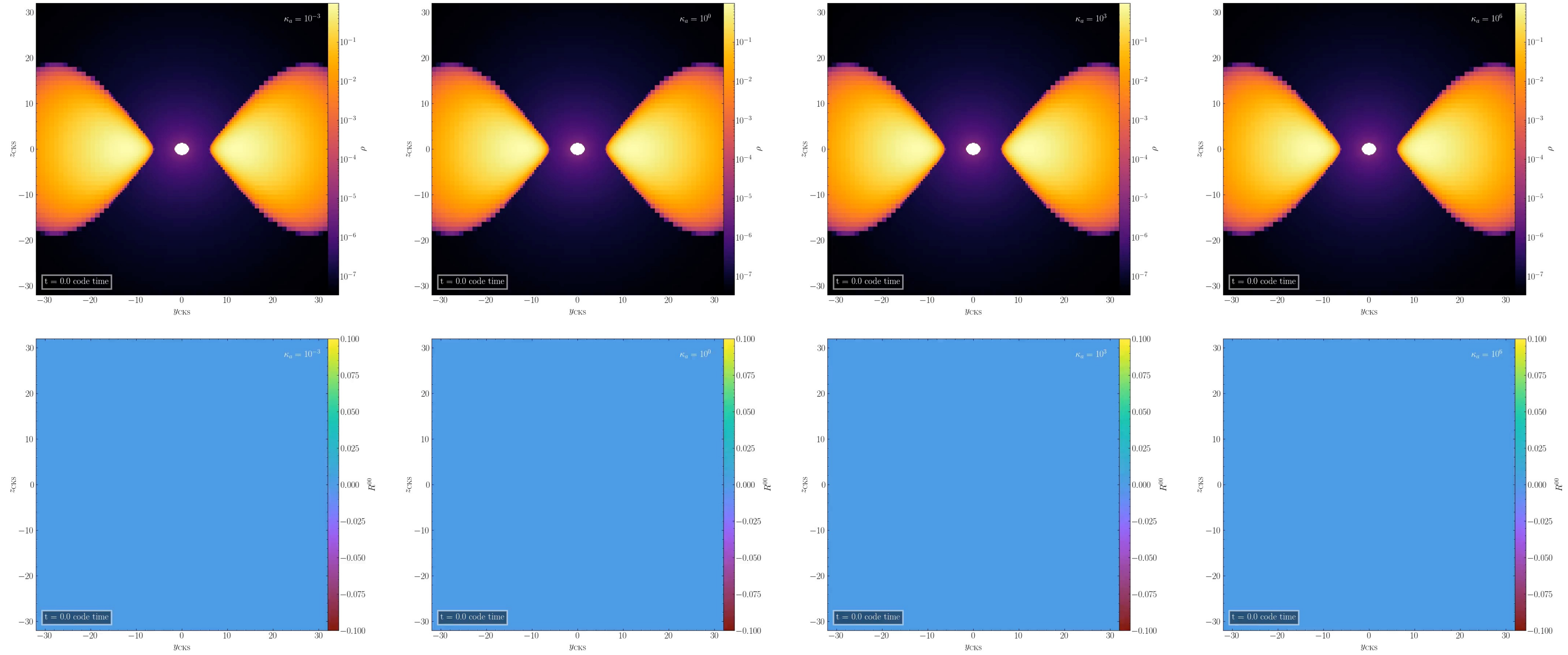
Dynamic diffusion test



Beam test near Kerr black hole

Radiation GR accretion flows computed with full transport method.

Hydrodynamic, spinning BH, Cartesian Kerr-Schild coordinates with 4 levels of mesh refinement, 64^3 base grid, 92-angle geodesic mesh. Run with Kokkos version of code on single A100 GPU in 8 hours.



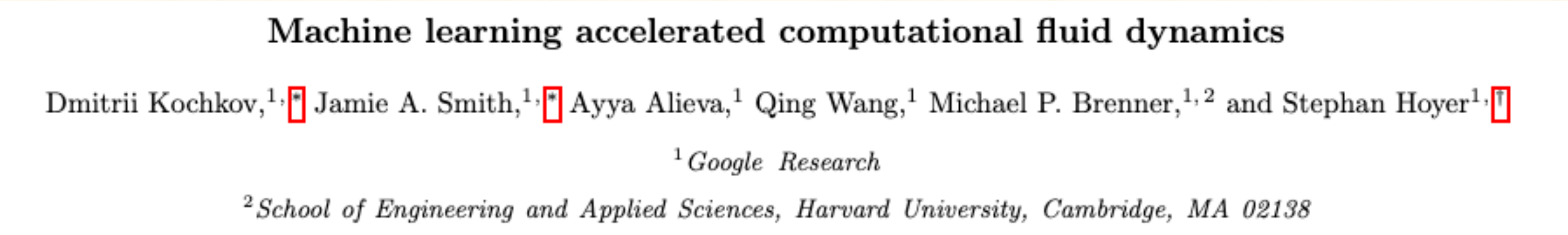
$\tau = 10^{-3}$ $\tau = 1$ $\tau = 10^{3}$ $\tau = 10^{6}$

Initial $P_{rad}/P_{gas} \sim 1$

Mullen et al., in prep.

Novel approaches

Considerable interest using ML to accelerate computation by learning sub-grid model for low-resolution that gives same results as high-resolution:



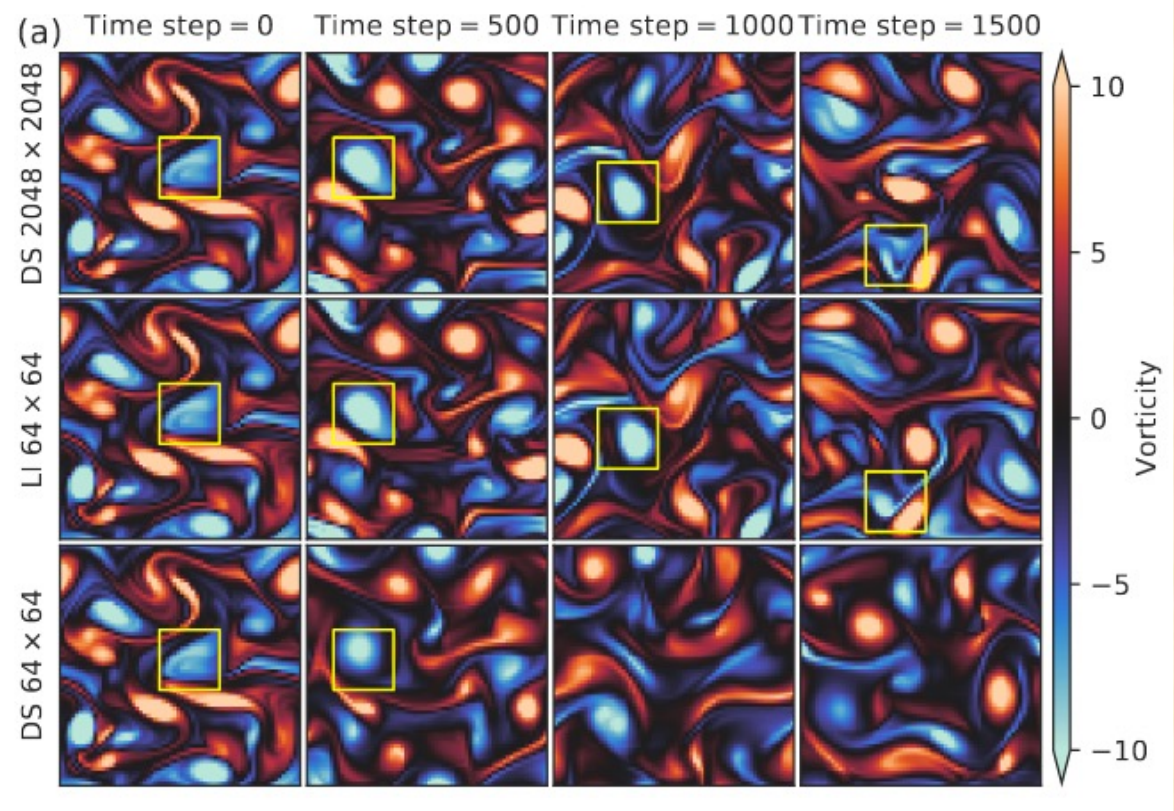
Machine learning accelerated computational fluid dynamics

Dmitrii Kochkov,[1,*] Jamie A. Smith,[1,*] Ayya Alieva,[1] Qing Wang,[1] Michael P. Brenner,[1,2] and Stephan Hoyer[1,†]

[1] *Google Research*

[2] *School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138*

Driven turbulence in 2D incompressible and viscous flow:



Summary

Numerical methods are an essential tool for theoretical astrophysics. Nearly every question relies on computation at some level for solution.

Developing software tools to tackle future problems faces many challenges:

- Complexity of programming modern HPC systems.
- Developing accurate algorithms to treat new physics.
- Developing software that includes full range of physics needed.

We can expect many exciting results in the near future:

- NS mergers including accurate treatment of microphysics.
- Spectra and light curves of radiation-dominated accretion flows using realistic opacities.
- Etc, etc.