



The Astra code

an Accelerated Spectral code for TuRbulent plAsmas

Geoffroy Lesur, Institut de Planétologie et d'Astrophysique de Grenoble

Rationale

Astro code techniques comparison

Rationale

Astro code techniques comparison

	Finite Volume (Dyablo, Ramses, Idefix)
Approach	Discretize the solution on elementary volumes
✓	<ul style="list-style-type: none">- Shock capturing capabilities- Simple to parallelize- Adapted to AMR
✗	<ul style="list-style-type: none">- average convergence ($O(N^2)$)- Very dissipative in subsonic flows- Boundary conditions often « messy »

Rationale

Astro code techniques comparison

	Finite Volume (Dyablo, Ramses, Idefix)	Smoothed Particle Hydrodynamics (Phantom, Shamrock)
Approach	Discretize the solution on elementary volumes	Discretize the solution on Lagrangian particles
✓	<ul style="list-style-type: none">- Shock capturing capabilities- Simple to parallelize- Adapted to AMR	<ul style="list-style-type: none">- AMR « for free »- Open boundary conditions are simple to implement- CFL condition less containing
✗	<ul style="list-style-type: none">- average convergence ($O(N^2)$)- Very dissipative in subsonic flows- Boundary conditions often « messy »	<ul style="list-style-type: none">- poor convergence ($O(\sqrt{N})$)- Shocks require some care- Numerical noise can become problematic

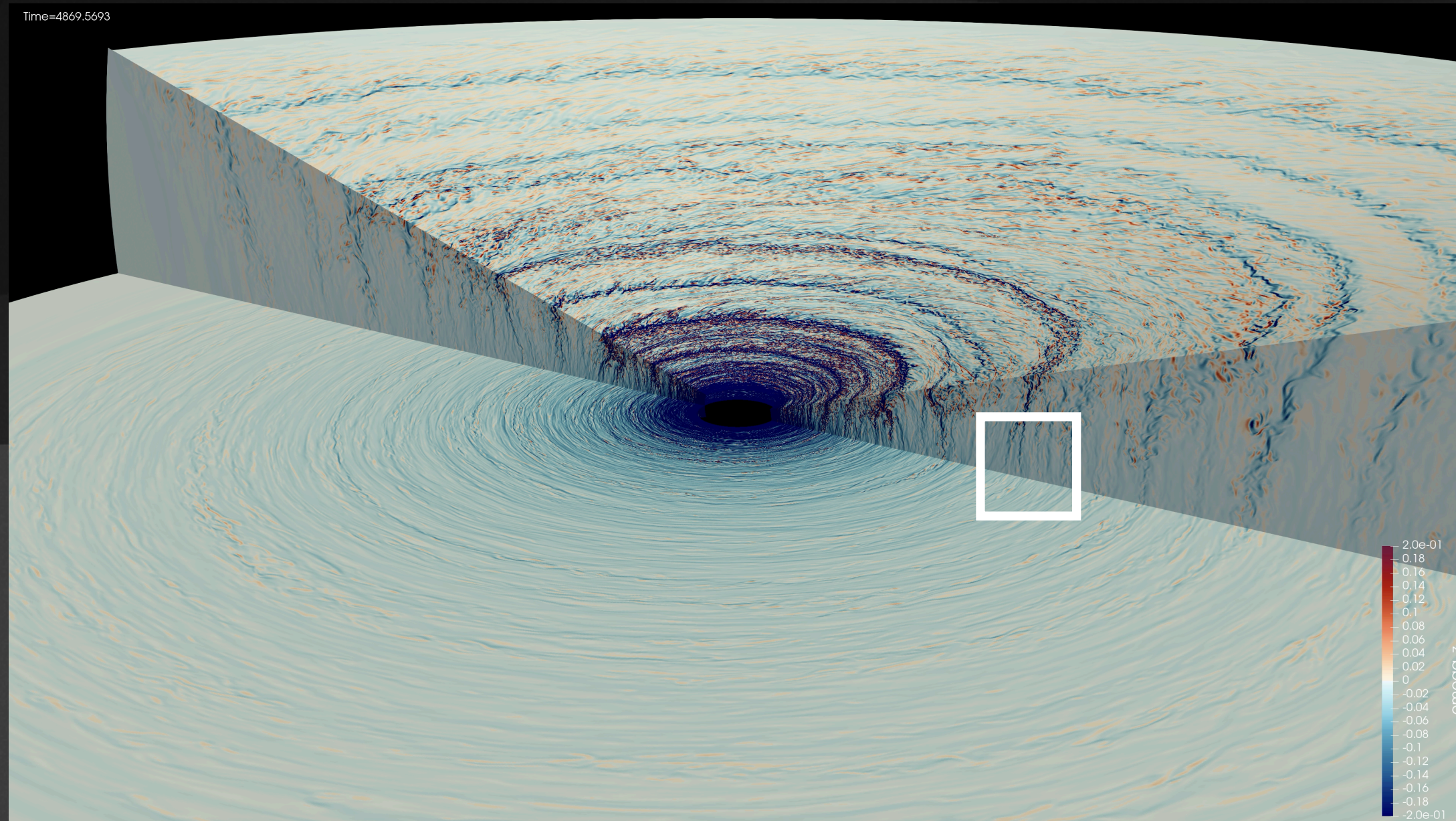
Rationale

Astro code techniques comparison

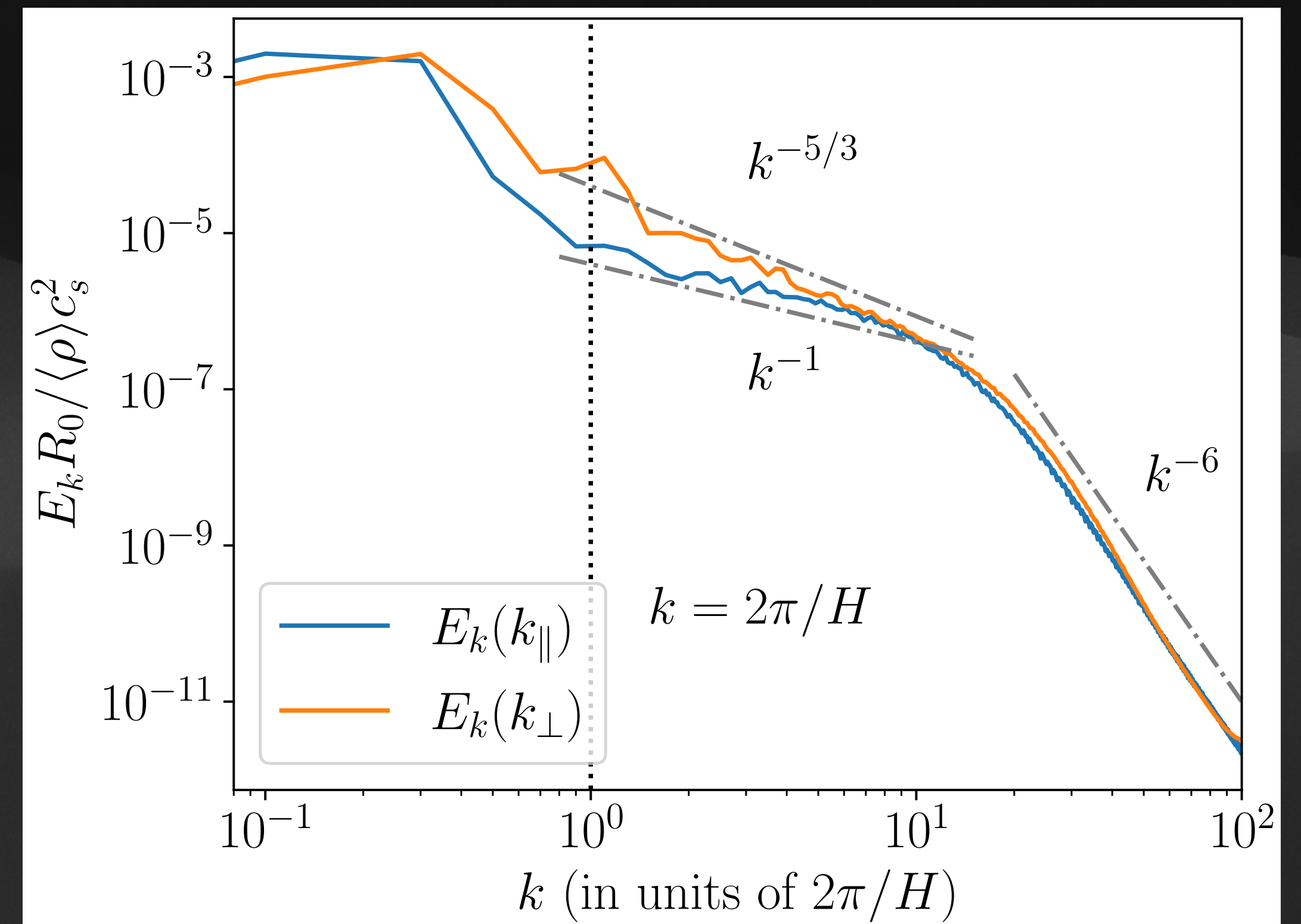
	Finite Volume (Dyablo, Ramses, Idefix)	Smoothed Particle Hydrodynamics (Phantom, Shamrock)	Spectral (Ash, Magic, Snoopy)
Approach	Discretize the solution on elementary volumes	Discretize the solution on Lagrangian particles	Discretize the solution on a « well chosen » basis
✓	<ul style="list-style-type: none">- Shock capturing capabilities- Simple to parallelize- Adapted to AMR	<ul style="list-style-type: none">- AMR « for free »- Open boundary conditions are simple to implement- CFL condition less containing	<ul style="list-style-type: none">- Fast convergence ($O(e^N)$)- algorithmically simple- No numerical dissipation
✗	<ul style="list-style-type: none">- average convergence ($O(N^2)$)- Very dissipative in subsonic flows- Boundary conditions often « messy »	<ul style="list-style-type: none">- poor convergence ($O(\sqrt{N})$)- Shocks require some care- Numerical noise can become problematic	<ul style="list-style-type: none">- hard to parallelize efficiently- Boundary conditions embedded in the basis

Practical example

Vertical Shear Instability with finite volume methods



[Lesur+2025]
Idefix code (finite volume)
- 4000^3 effective resolution,
- high order reconstruction



turbulent cascade

numerical dissipation

A decade in resolution is « lost » by numerical dissipation

The Snoopy code

- * MHD equations solved in the sheared frame
- * Use the Fourier basis to compute derivatives: $\partial_{x'} \leftrightarrow ik_x(t)$...
- * Compute non linear terms using a pseudo spectral representation
- * 3rd order low storage Runge-Kutta integrator
- * OpenMP and / or MPI parallelization
- * Written in C

- * Advantages:
 - * Shearing waves are computed exactly (natural basis)
 - * Exponential convergence when resolution is increased
 - * Magnetic flux conserved to machine precision
 - * Sheared frame & incompressible approximation: no CFL constrain due to the background sheared flow / sound speed.
 - * Very weak numerical dissipation: tight control on physical dissipation processes
- * Disadvantages:
 - * Slower than finite differences for the same resolution (number of real grid points)
 - * Shocks / discontinuities can't be treated spectrally (Gibbs oscillations)
 - * Strongly parallel codes are not very efficient



17 years ago...

Algorithm

Example: Incompressible hydro case

- Solve the Navier-Stokes equation
(free to choose among a variety of time integrators)

$$\frac{\partial \hat{v}_k}{\partial t} = -ik \cdot \widehat{(v \otimes v)}_k - ik \hat{P}_k - \nu k^2 \hat{v}_k$$

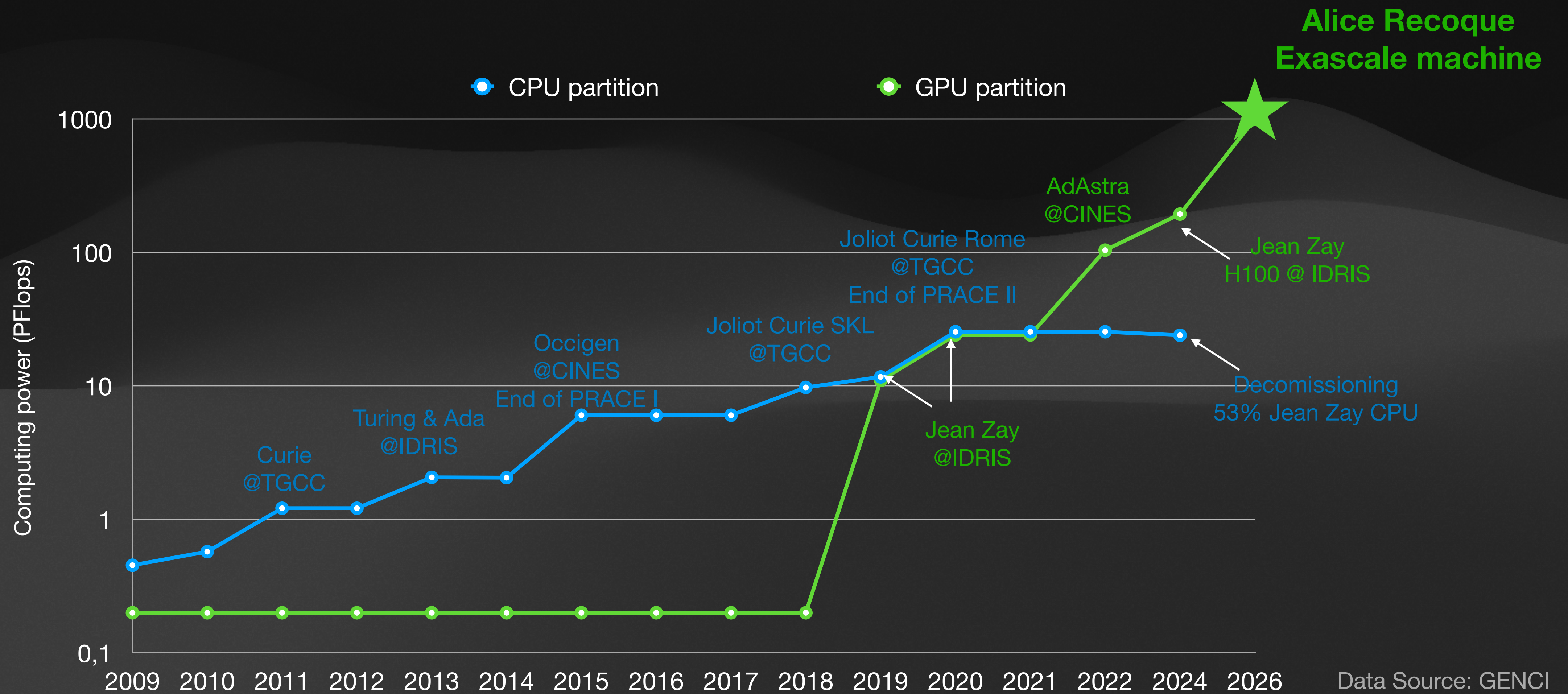
complex amplitude of velocity mode k computed in real space, and transformed back to Fourier space Pressure gradient viscous term

- Pressure computed from incompressibility condition

$$k \cdot \hat{v}_k = 0 \quad \longrightarrow \quad P_k = -\frac{k \cdot \left[k \cdot \widehat{(v \otimes v)}_k \right]}{k^2}$$

All of the derivatives become algebraic products in Fourier space
Nonlinear terms require FFTs and iFFT


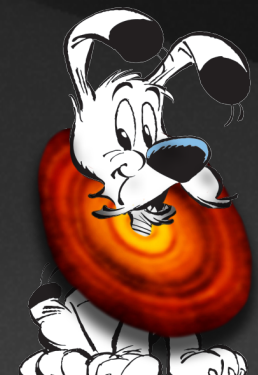
Leveraging new HPC architectures



The capability to use accelerated machines is now mandatory to have access to the largest French (and European) supercomputers

The ASTRA code

Accelerated Spectral code for TuRbulent plAsmas

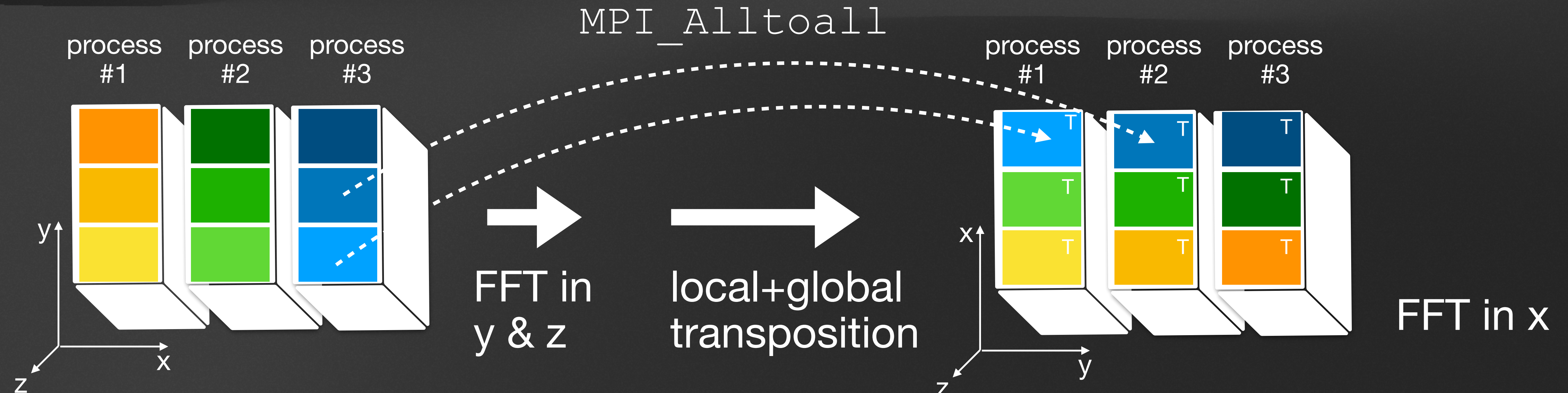
- Easy-to-use spectral code, based on
 - Snoopy algorithm (ANSI C, spectral CPU) 
 - Idefix data structures (C++20, finite volume, GPU+CPU). 
- C++20, Kokkos-based (performance portability):
same code runs both on your laptop's CPU and on 100s of GPUs
- Uses Fourier basis in cartesian boxes (periodic/shear-periodic).
Relies on Kokkos-FFT
- All-is-a-plugin approach, using generic C++ class template (from time-integrators to equation right-hand side)
- Python interface to define initial conditions and customized « on the fly » outputs
- Automatic multi-architecture validation (CI/CD)

Feature	Status
3D incompressible HD & MHD, compressible HD	public
Ohmic resistivity, viscosity, hyperdiffusion	public
Multiple time integrator (Euler, RK2, LS-RK3)	public
Non-ideal MHD (Ambipolar, Hall)	coming soon
Boussinesq stratification	coming soon
MPI, MPI+OpenMP, MPI+Cuda, MPI+HIP, MPI+Sycl	public
Large-scale shear	public
Python interface « astrapy »	public
Dust (particle approach)	planned
Dust (fluid approach)	coming soon

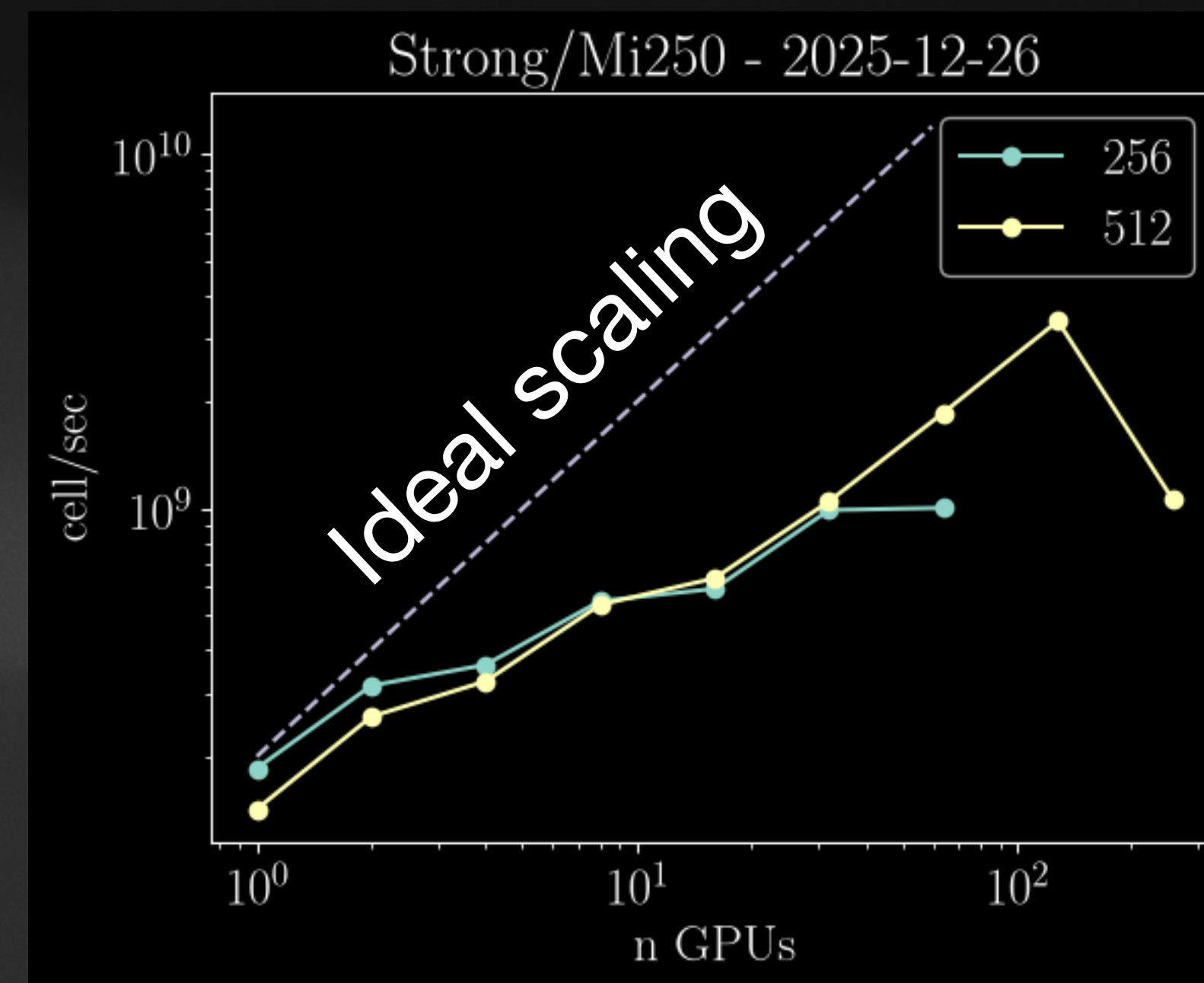
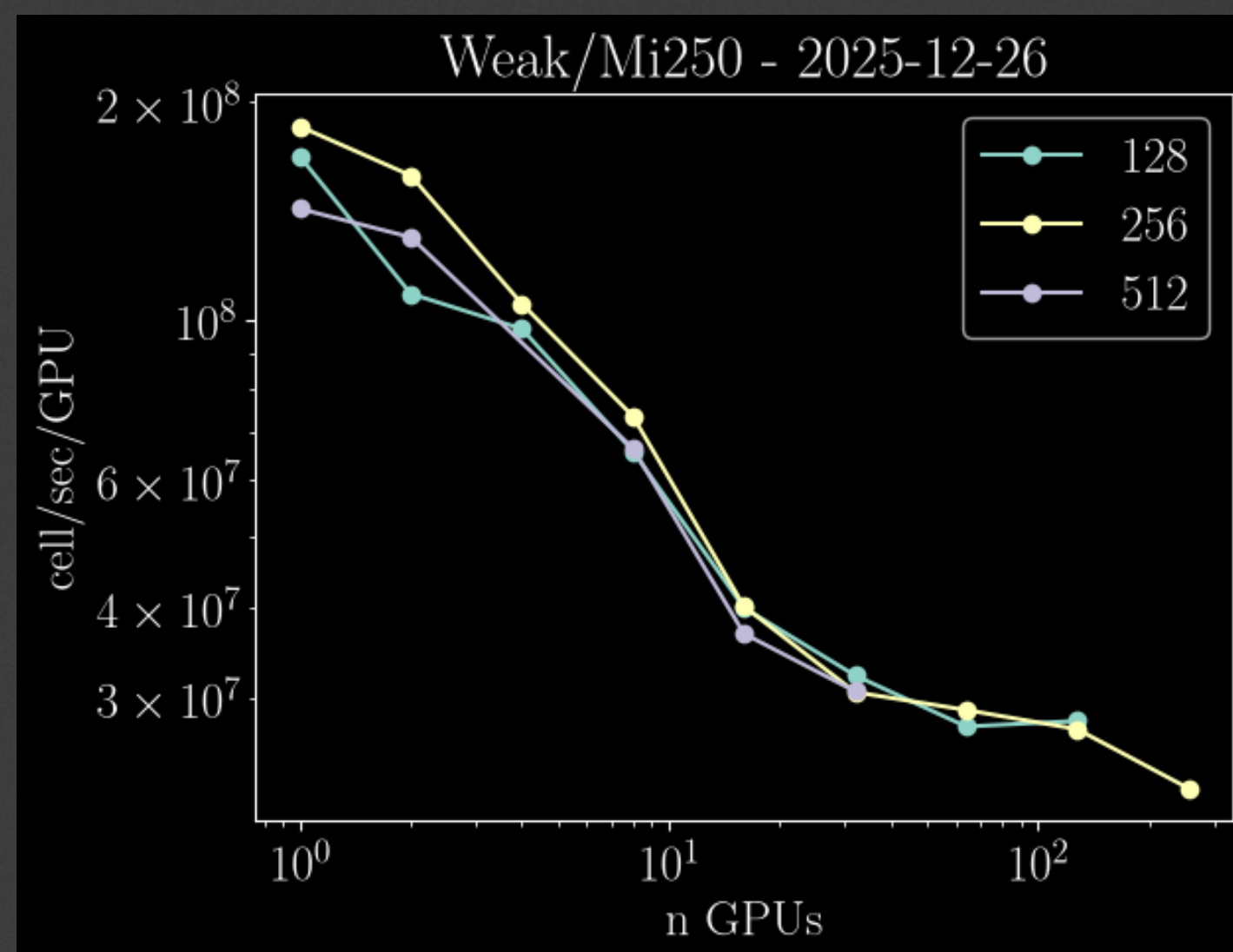
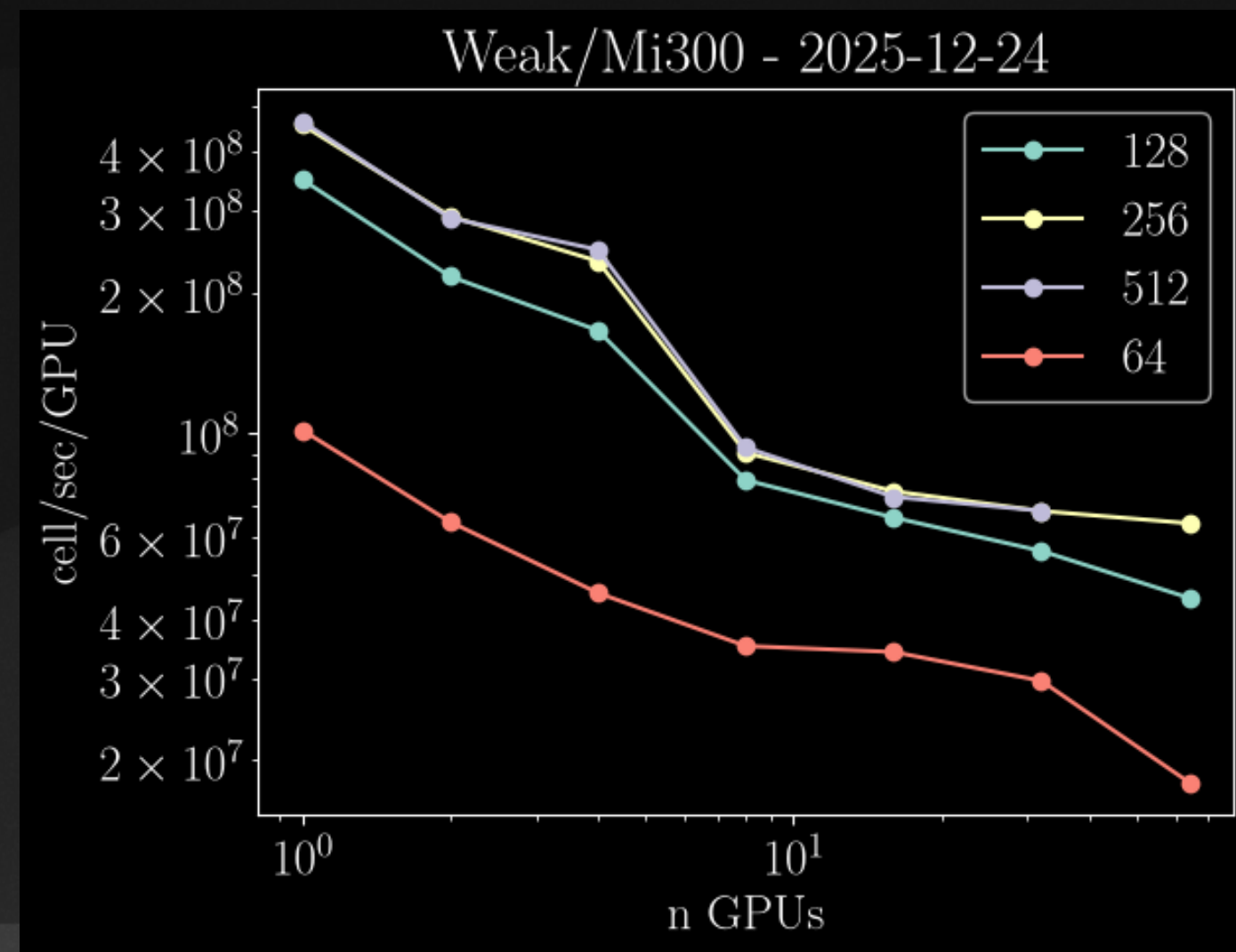
Parallelization

Implementing MPI+GPU parallel FFTs

- Neither Kokkos nor Kokkos-fft support multi GPU FFTs
- ASTRA uses a slab domain decomposition, with a global transposition to perform distributed FFTs



Performances & Scaling

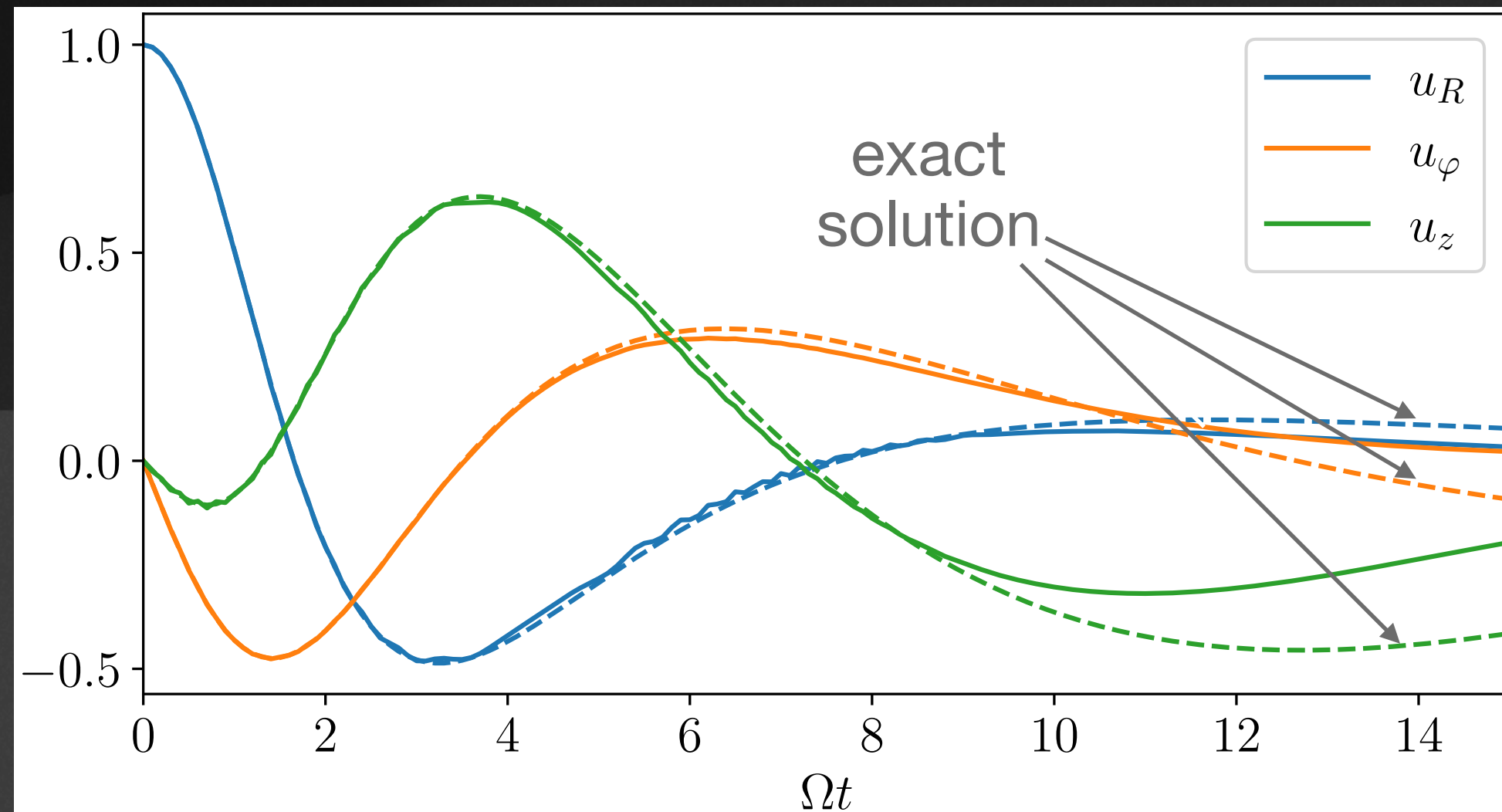
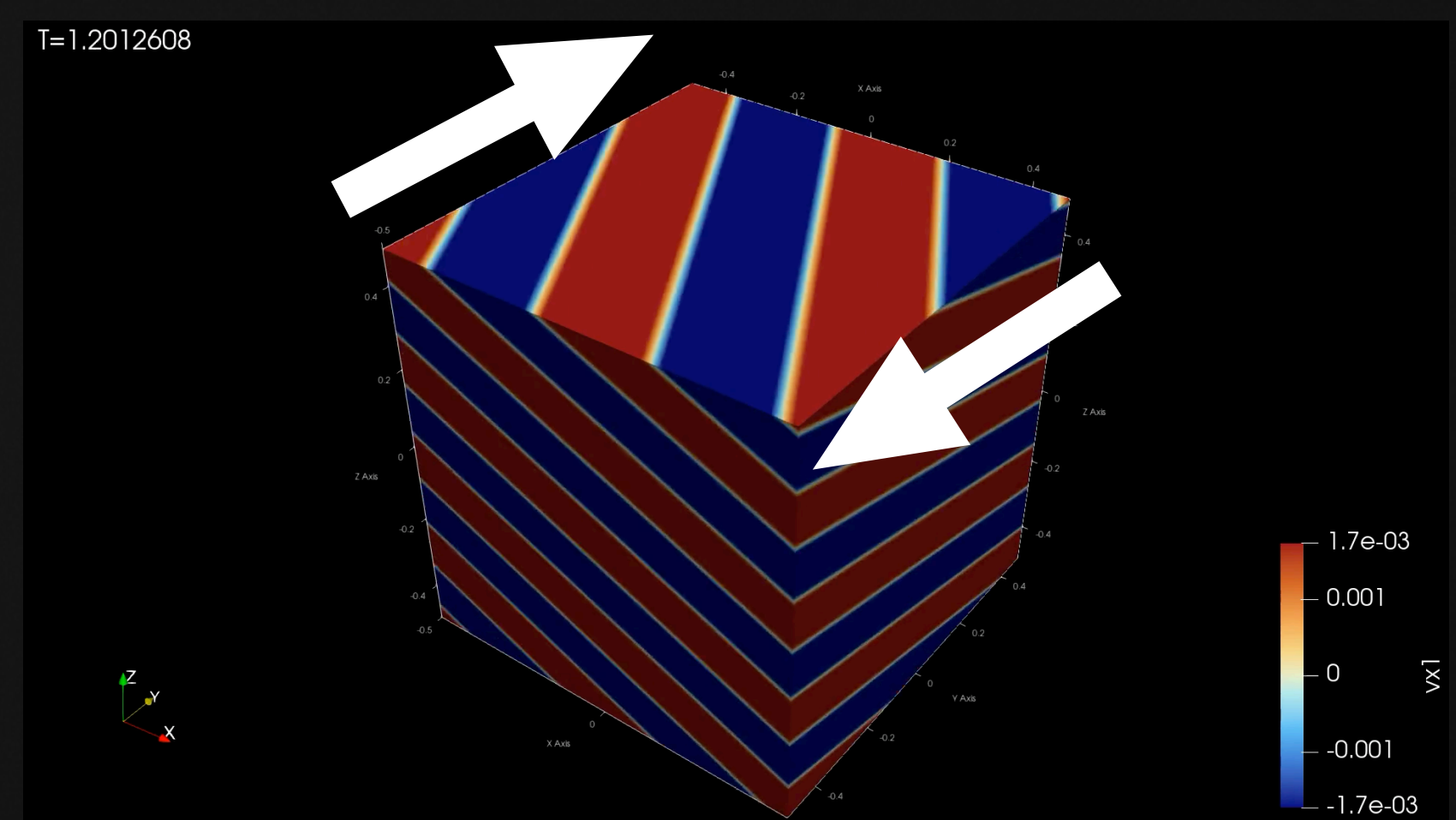


Scaling is « decent » for a spectral code (remember FFT is $\mathcal{O}(n \log(n))$!)

Better performances could be achieved with specialized MPI_Alltoall

Testing

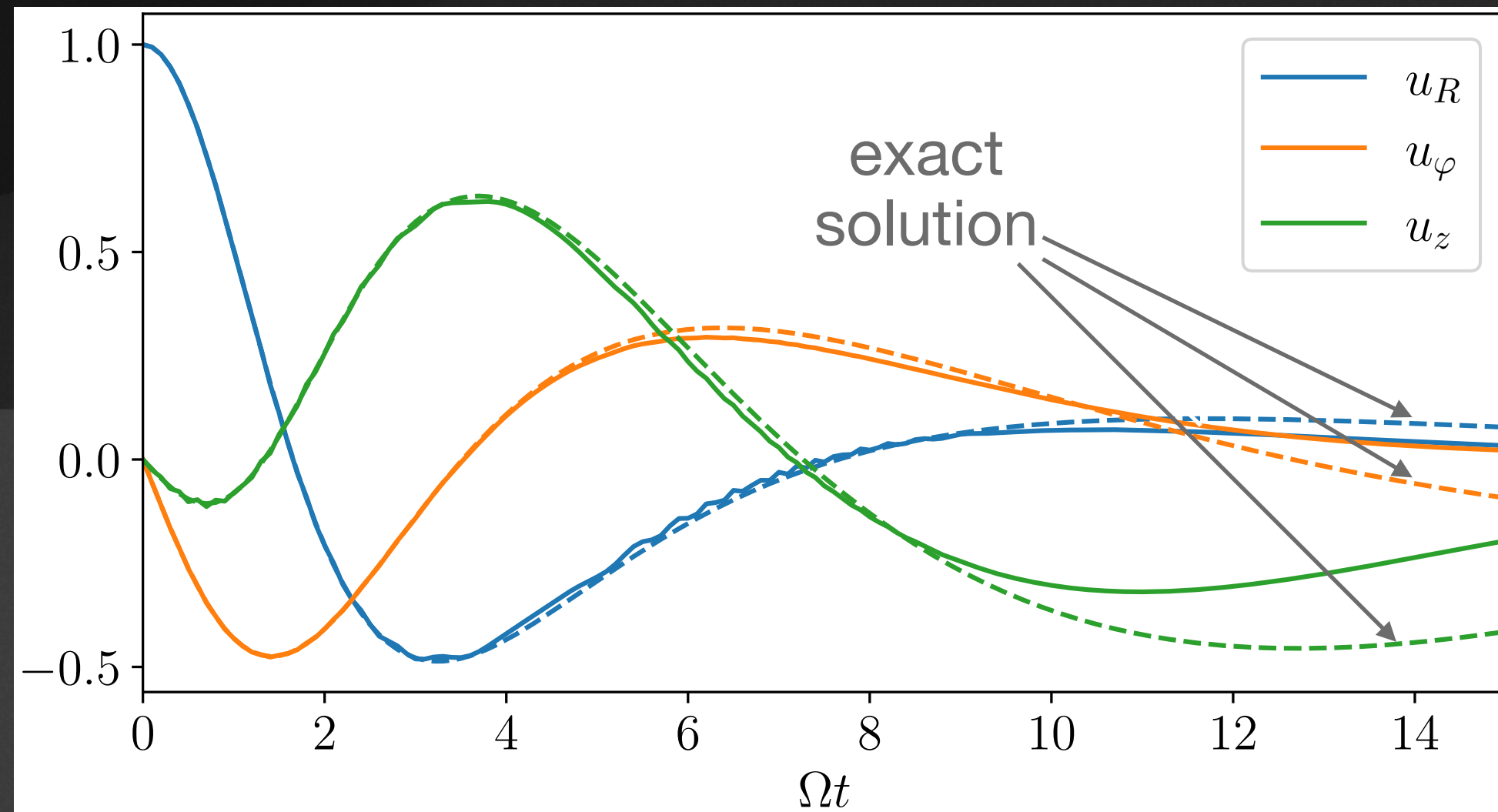
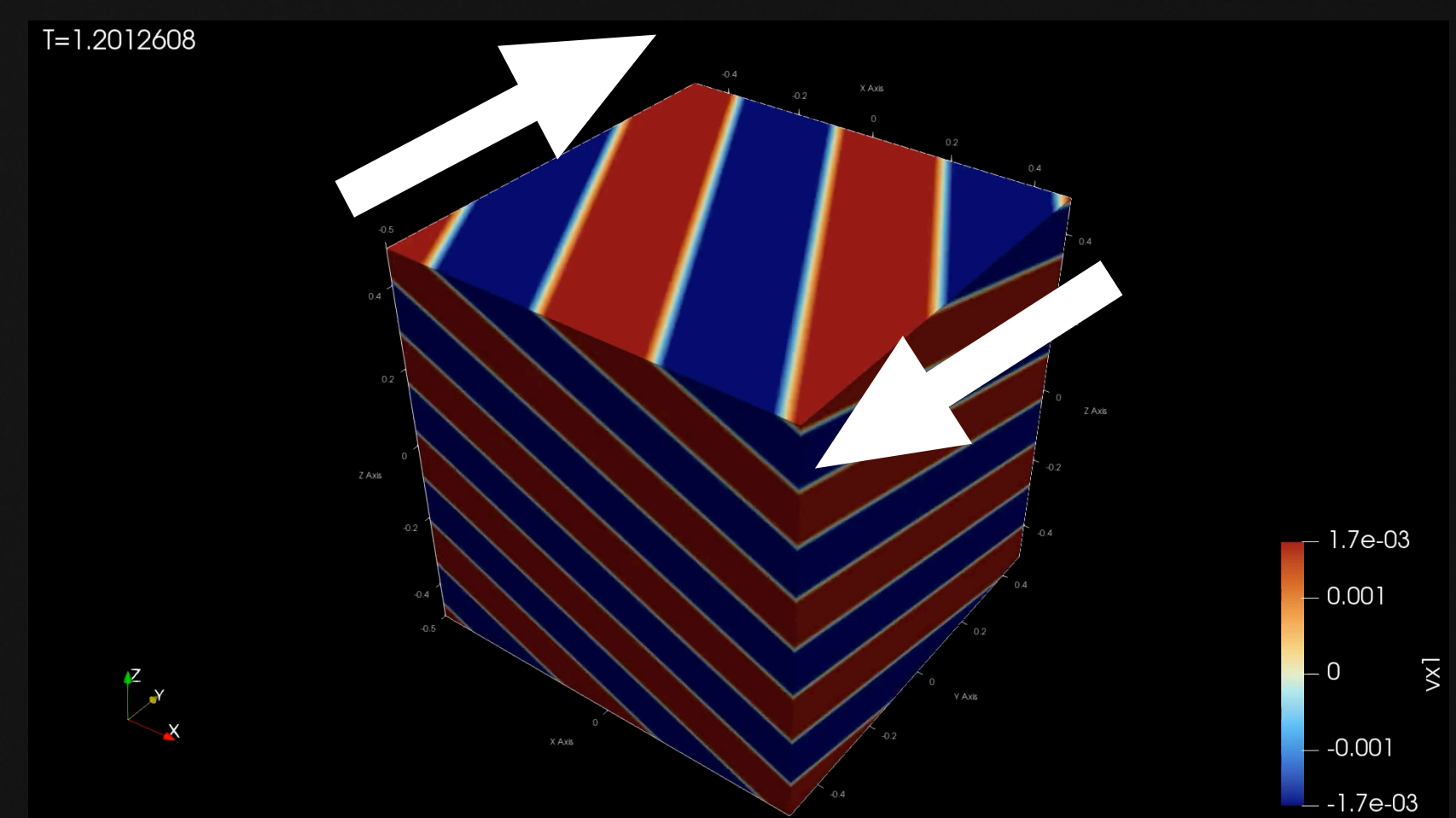
Shearing wave test [Balbus & Hawley 2006]



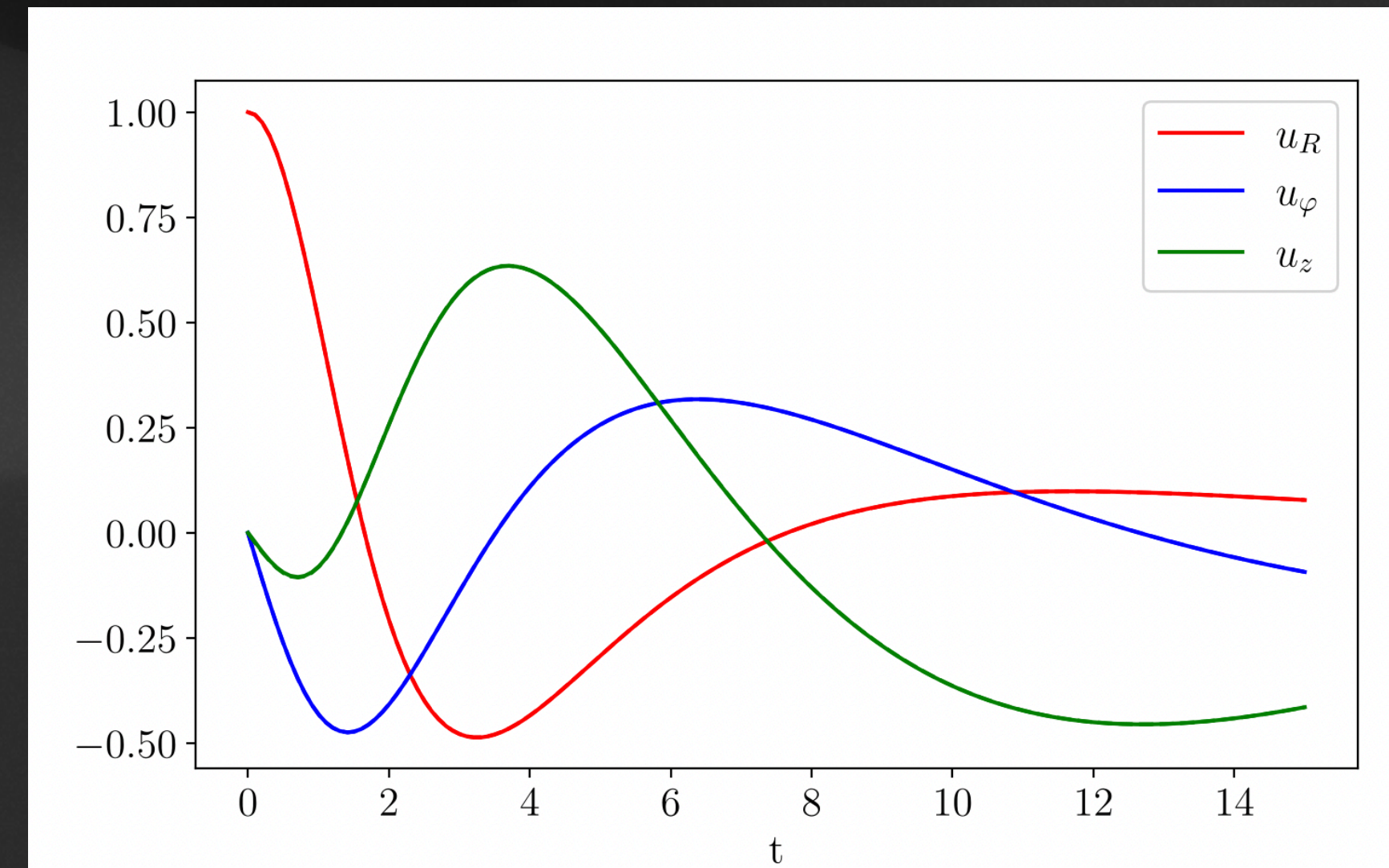
Idefix, finite volume, 256^3
[Lesur+2023]

Testing

Shearing wave test [Balbus & Hawley 2006]



Idefix, finite volume, 256^3
[Lesur+2023]



Astra, spectral, 64^3
(relative error $< 10^{-3}$)

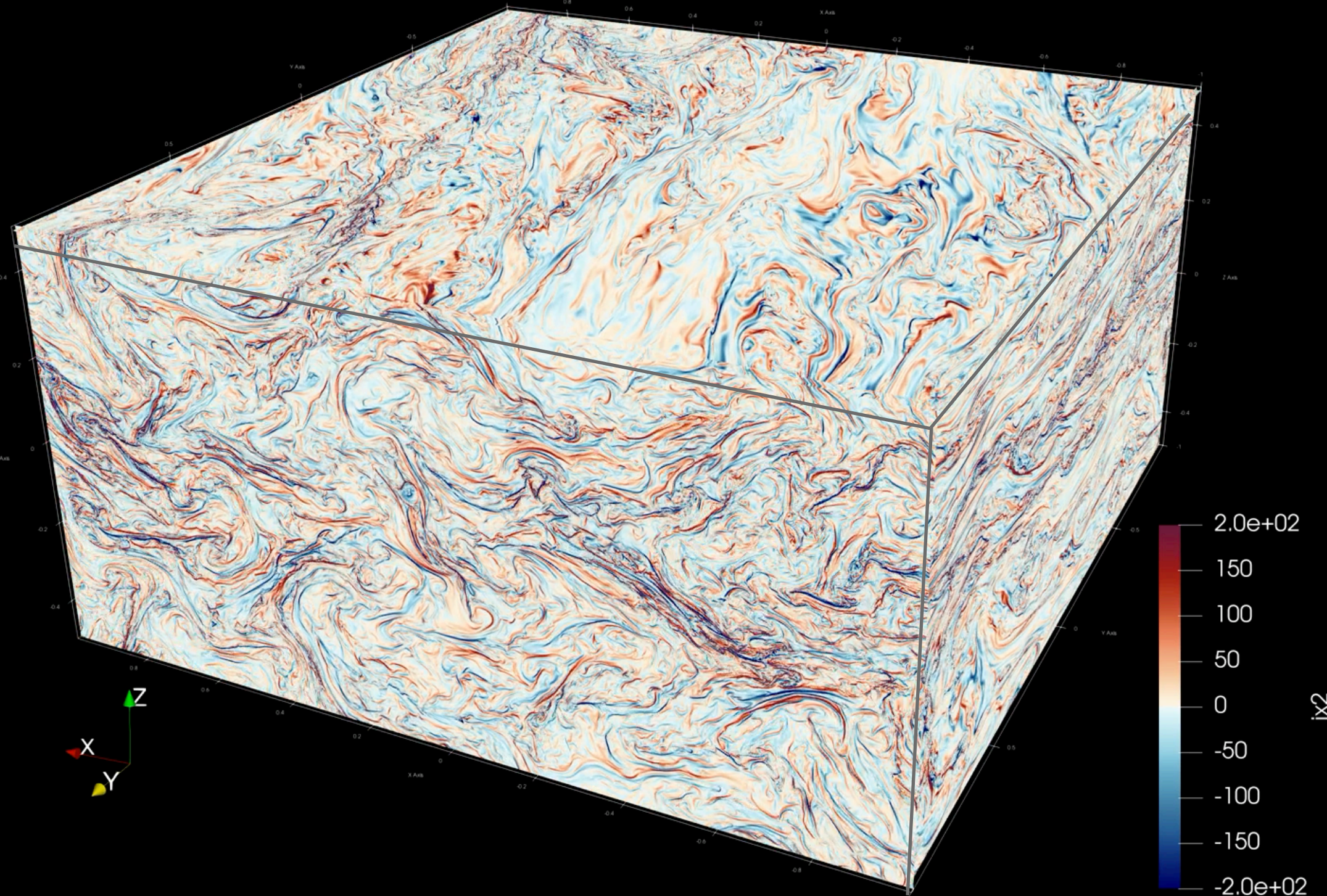
Better results with 4x less resolution (runtime divided by 200)

Example: Magneto-rotational instability

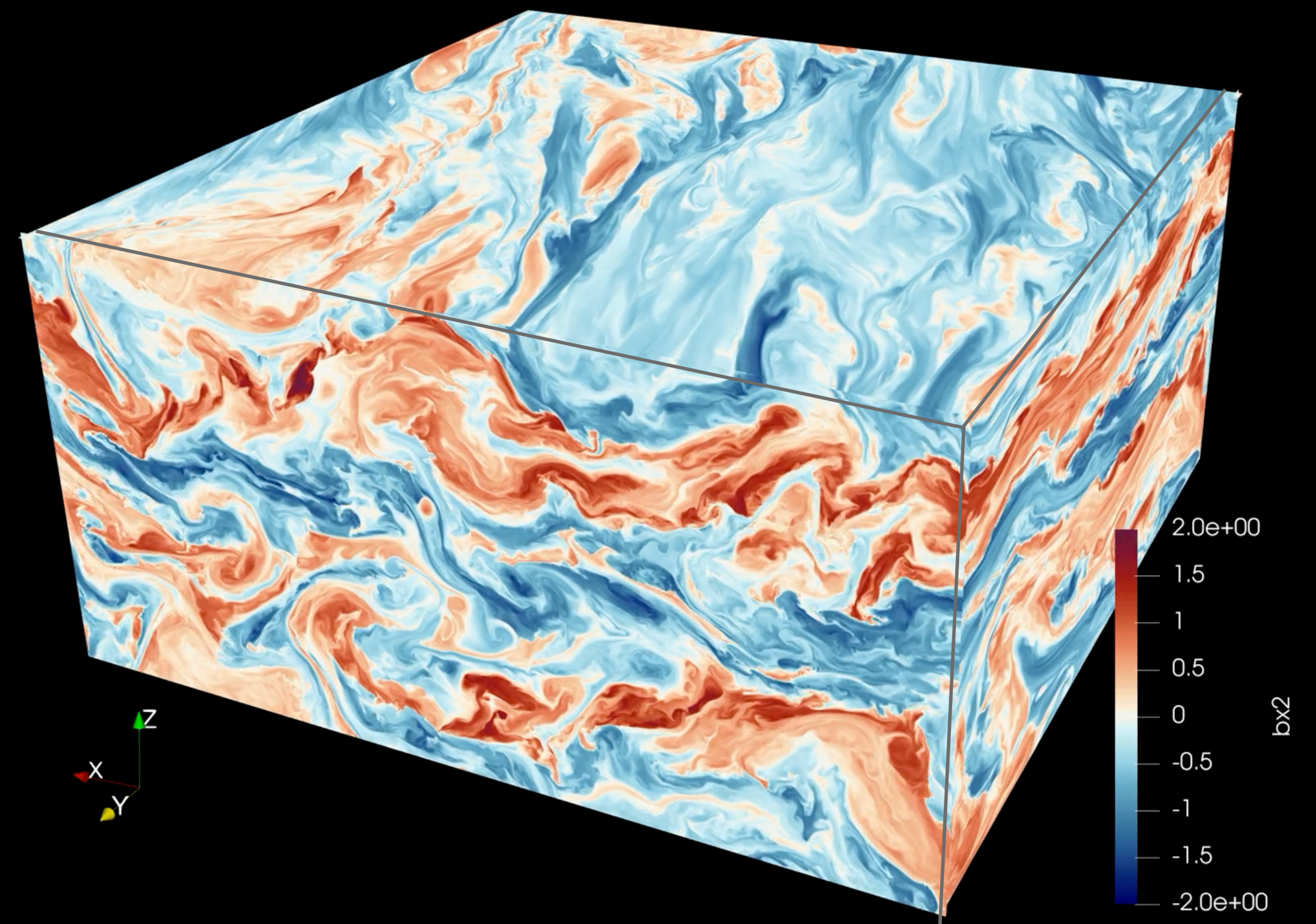
Saturation at very high Reynolds number $[2048^3]$

$t=16.510029$

electrical current



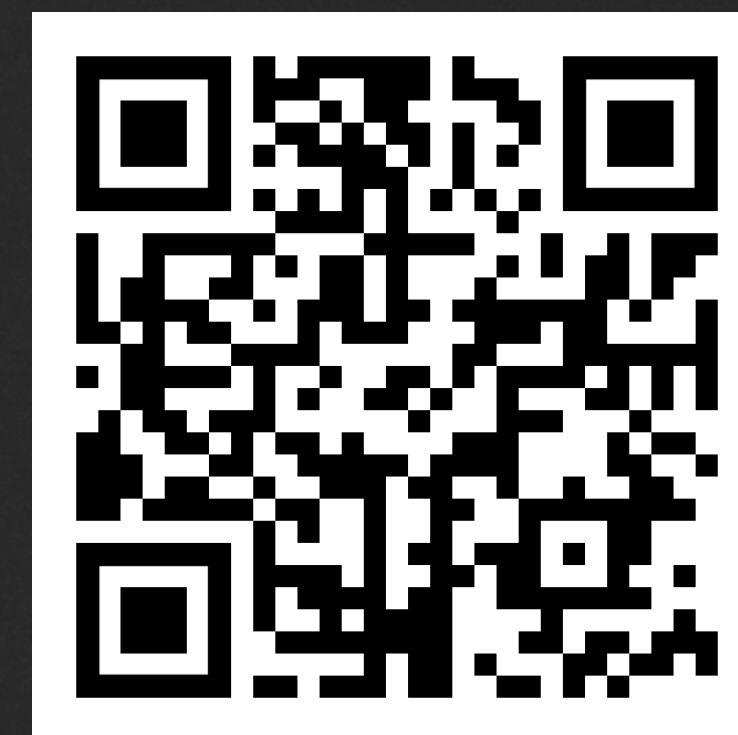
toroidal magnetic field



3000h on 64 Mi300 GPU

Conclusion

- Astra is a spectral (Fourier) code for (compressible) hydro and MHD.
- C++20, Kokkos-based with Python embeddings.
- Fast convergence (compared to FV and SPH).
- Performance portable, from laptop CPUs to GPU clusters.
- Available now: github.com/glesur/astra



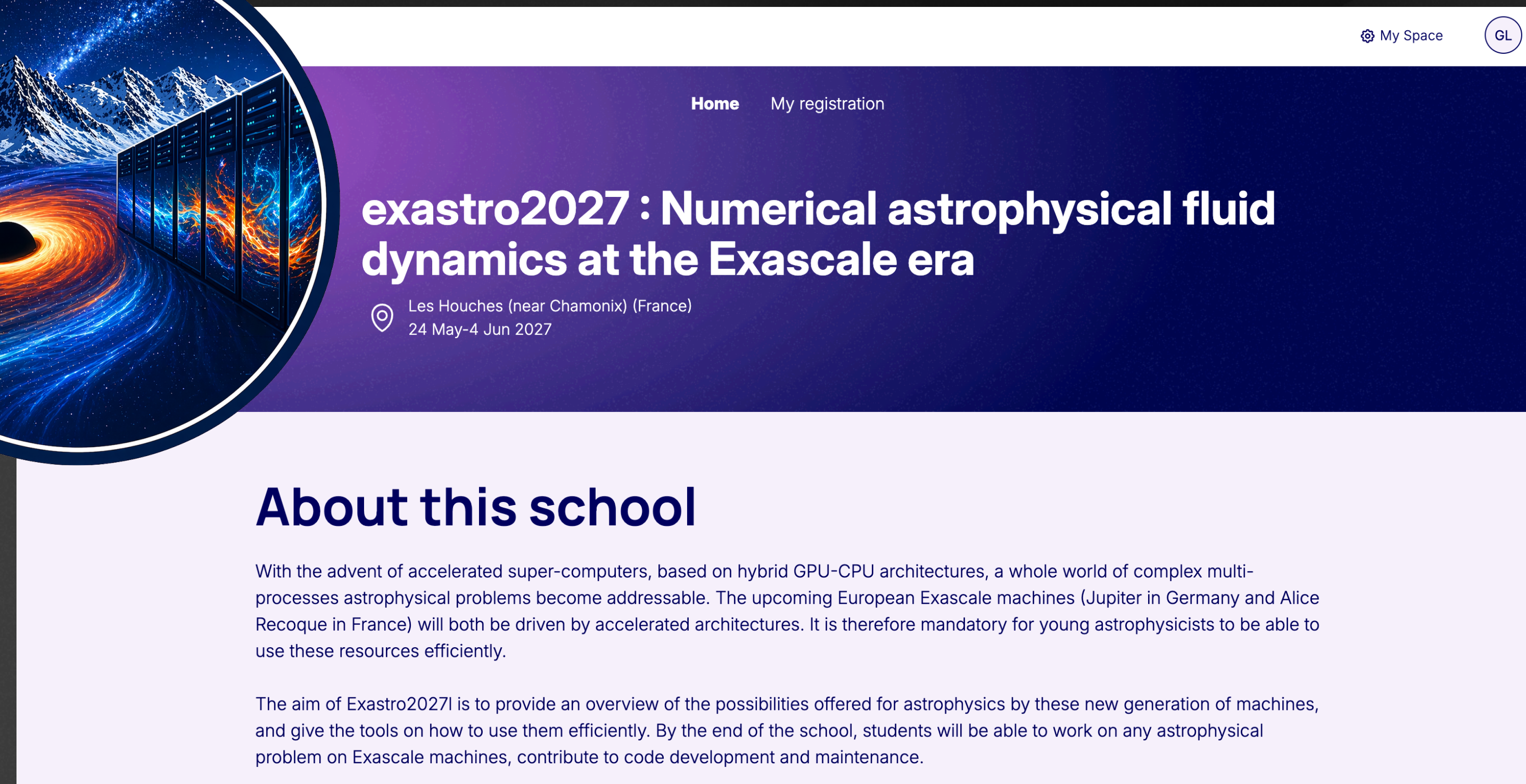
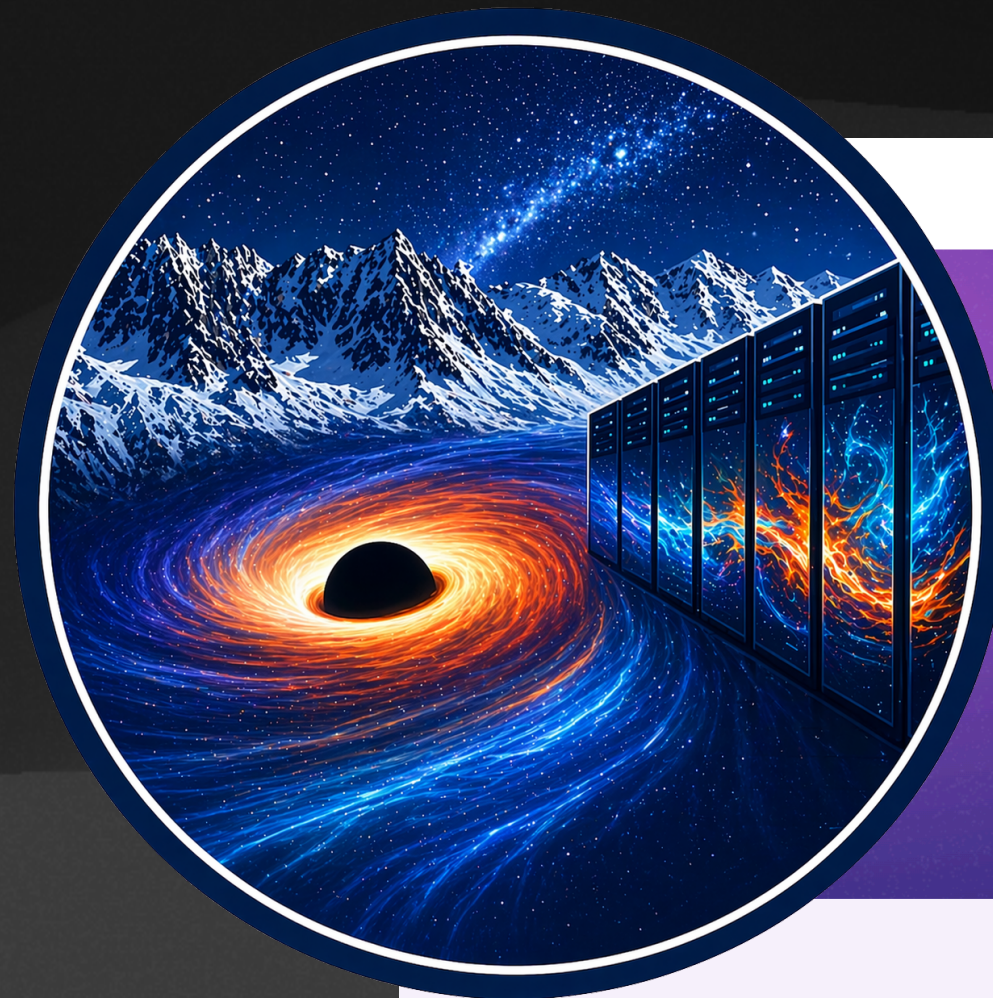
Announcement

Exastro2027 School of physics

May 24 - June 4 2027, Les Houches



exastro2027.sciencesconf.org



The screenshot shows a website interface for the event. At the top right, there is a 'My Space' link and a user profile icon labeled 'GL'. Below this is a navigation bar with 'Home' and 'My registration' links. The main header features the event title 'exastro2027 : Numerical astrophysical fluid dynamics at the Exascale era' in white text on a dark blue background. Below the title, a location pin icon indicates 'Les Houches (near Chamonix) (France)' and the dates '24 May-4 Jun 2027'. The main content area has a white background with the heading 'About this school' in bold blue text. Below this heading are two paragraphs of text describing the school's focus on exascale computing and its goals for students.

Registration opens Sep. 1st 2026