THE GPLUTO CODE FOR ASTROPHYSICAL PLASMAS

Agnese Costa[1] on behalf of the PLUTO group

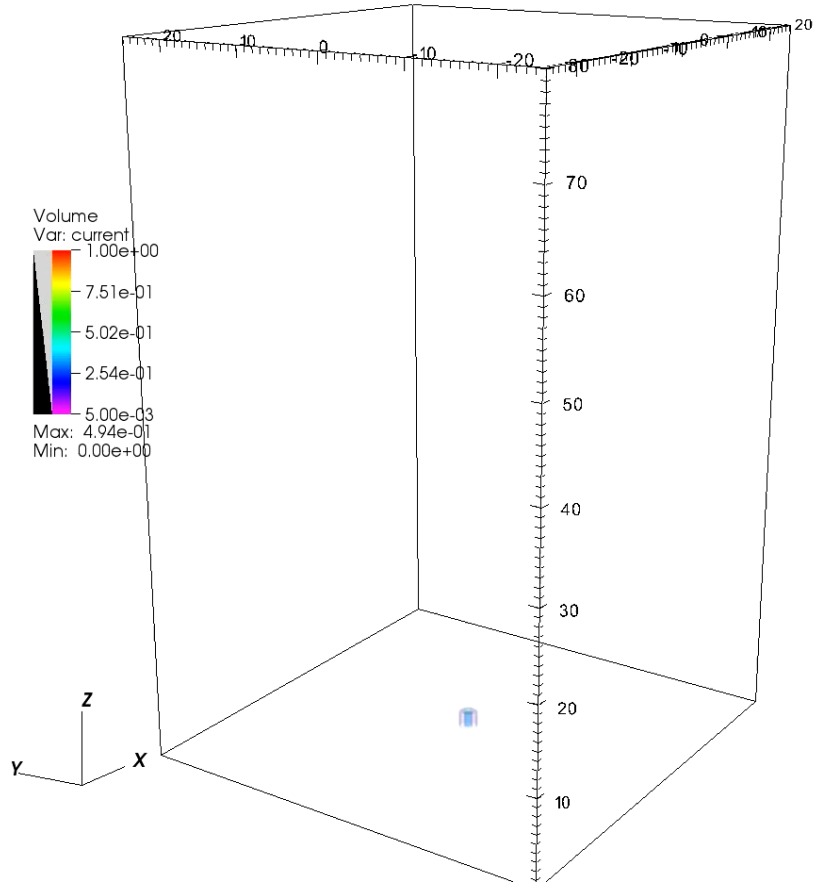
Contributors: A. Mignone, M. Rossazza, A. Suriano, A. Sciaccaluga, D. Mukherjee, G. Bodo, P. Rossi, V. Berta, S. Truzzi, M. Bugli

1: INAF – OA Torino

USC-C General Assembly – Trieste, 2026

THE PLUTO CODE

- PLUTO[1,2] is a finite volume Godunov-type modular parallel code providing a multi-physics and multi-algorithm framework for solving the hyperbolic/parabolic equations of gas and plasma dynamics in astrophysics.
- Target: multidimensional compressible plasma with high-Mach numbers
- Freely distributed at https://plutocode.ph.unito.it/ (v. 4.4)



[1]Mignone et al. ApJS (2007), 170, 228-242; [2]Mignone et al, ApJS (2012), 198, 7

THE PLUTO CODE

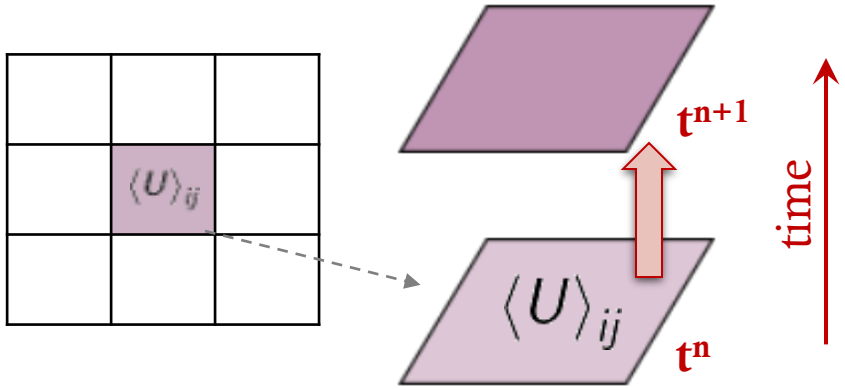
- PLUTO is (primarily) an Eulerian code, solving conservation laws on a fixed / adaptive grid

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) && = 0 && \text{(Mass cons.)}\\
&\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot \left[\rho \mathbf{u}\mathbf{u} - \frac{\mathbf{B}\mathbf{B}}{4\pi} + \left(p + \frac{\mathbf{B}^2}{8\pi}\right)\right] && = 0 && \text{(Momentum cons.)}\\
&\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(E + p + \frac{\mathbf{B}^2}{8\pi}\right)\mathbf{u} - \frac{(\mathbf{u}\cdot\mathbf{B})}{4\pi}\mathbf{B}\right] && = 0 && \text{(Energy cons.)}\\
&\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\boldsymbol{u}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{u}) && = 0 && \text{(Mag. flux cons.)}
\end{aligned}
$$

- PLUTO is shock-capturing, relying on finite volume formalism, evolving **volume averages**

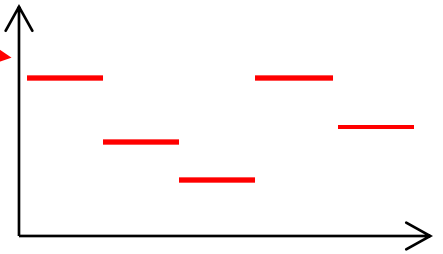
$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0 \quad \rightarrow \quad \frac{d\,\langle U\rangle_{ij}}{dt} + \int_{ij} \mathbf{F} \cdot d\mathbf{S} = 0$$

$$\langle U\rangle_{ij} = \frac{1}{\Delta V}\int_{ij} U(x,y)\,dx\,dy\,, \qquad U = \{\rho,\ \rho\mathbf{v},\ E,\ \mathbf{B}\}$$



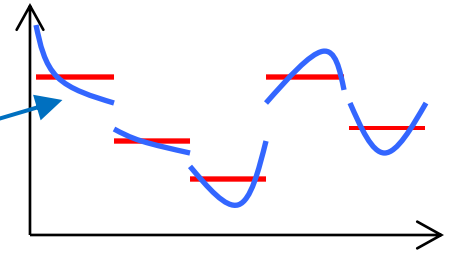
CODE STRUCTURE

$$\langle \mathbf{U} \rangle_i^n$$



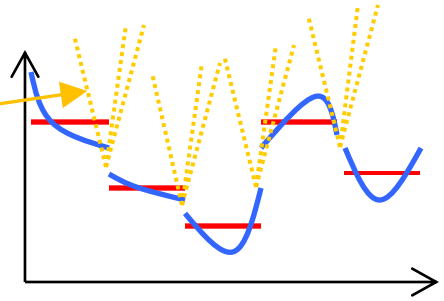
- The solution is obtained by iterating the update of **volume averages** of physical quantities to the next time step
- Three step iterative structure:
  1. **Reconstruct** **interface values** from zone averages using a high-order non-oscillatory poliynomial:

$$\begin{cases} \mathbf{U}^L_{i+\frac{1}{2}} = \lim_{x \to x^-_{i+\frac{1}{2}}} \mathbf{U}_i(x)\,, \\ \mathbf{U}^R_{i+\frac{1}{2}} = \lim_{x \to x^+_{i+\frac{1}{2}}} \mathbf{U}_{i+1}(x)\,, \end{cases}$$



  2. **Solve** Riemann problems at cell interfaces to obtain **interface flux**
  3. **Update** in time conserved variables:

$$\frac{d\langle \mathbf{U} \rangle}{dt} = -\frac{1}{\Delta \mathcal{V}} \sum_{\text{faces}} \mathbf{F} \cdot \hat{\mathbf{n}}\, dA + \langle \mathbf{S} \rangle$$

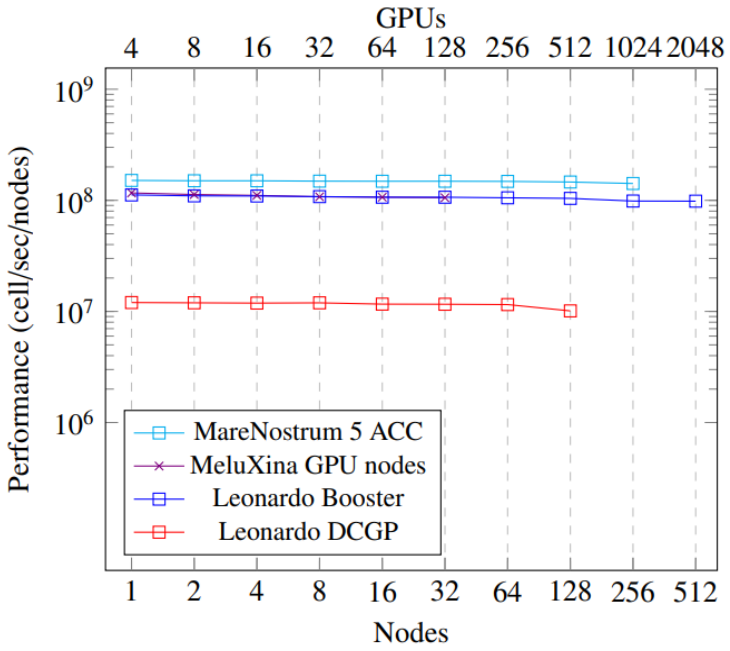


THE PLUTO LEGACY: PLUTO

gPLUTO is the new GPU-accelerated version of the PLUTO code targeting exascale facilities and new generation hardware, available at https://gitlab.com/PLUTO-code/gPLUTO

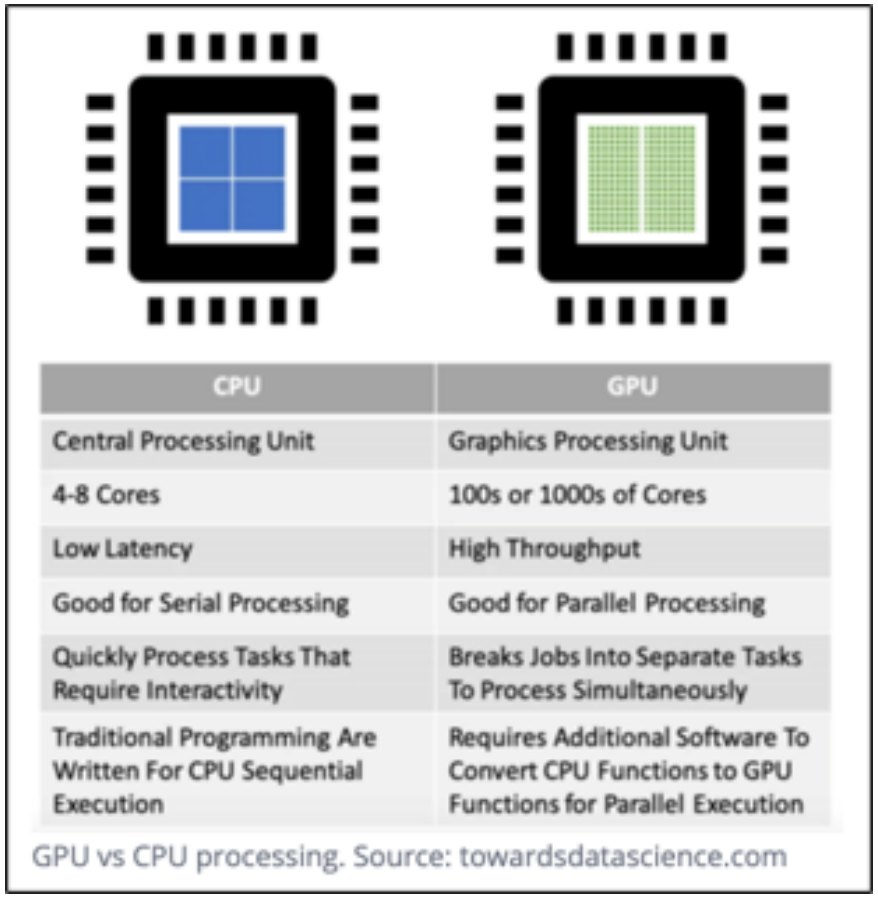
- C++ & OpenACC, OpenMP (high-level directive-based programming models developed by NVIDIA/OpenMP) chosen as programming paradigms

- **Current status**
  - 80% of the static-grid code ported to GPUs
  - Deployed on all JU systems with both CPUs and GPUs
  - Extensive code revision



- The standard version of PLUTO will be maintained although it will not receive major upgrades

GPU PORTING



CPU	GPU
Central Processing Unit	Graphics Processing Unit
4-8 Cores	100s or 1000s of Cores
Low Latency	High Throughput
Good for Serial Processing	Good for Parallel Processing
Quickly Process Tasks That Require Interactivity	Breaks Jobs Into Separate Tasks To Process Simultaneously
Traditional Programming Are Written For CPU Sequential Execution	Requires Additional Software To Convert CPU Functions to GPU Functions for Parallel Execution

GPU vs CPU processing. Source: towardsdatascience.com

Model: compute and data directives for accelerating loop and data transfer from CPU to GPU

```
#pragma acc enter data copyin (A,B)
// ... Code where A ad B are used in GPU computations
#pragma acc exit data delete (A,B)
```

```
#pragma acc parallel loop vector
for (i = 0; i < N; i++){
  // Loop body
  // ... Things to do here ...
}
```

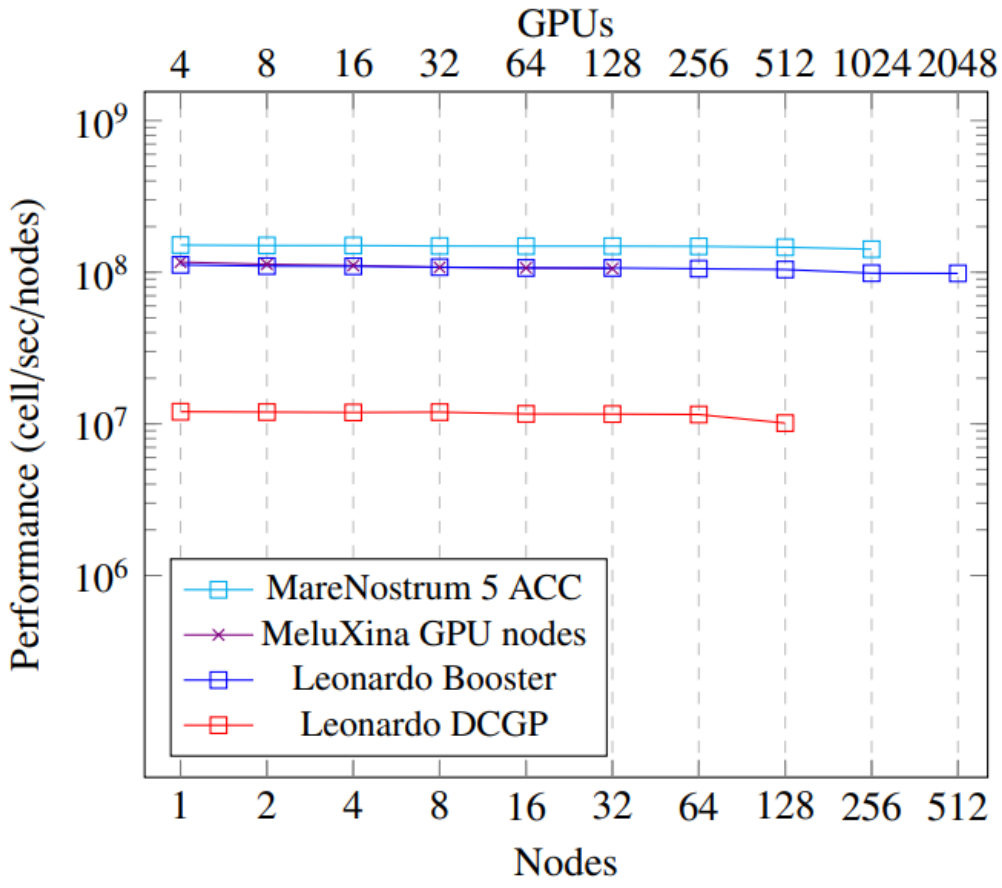
Keypoints:

- Data locality-> computational intensive code all on GPUs
- Private variables -> allocated individually
- Coalesced access to memory -> different array ordering

ACCELERATION PERFORMANCE: CPU VS GPU

3D ORSZAG-TANG MHD TEST PROBLEM, OPENACC

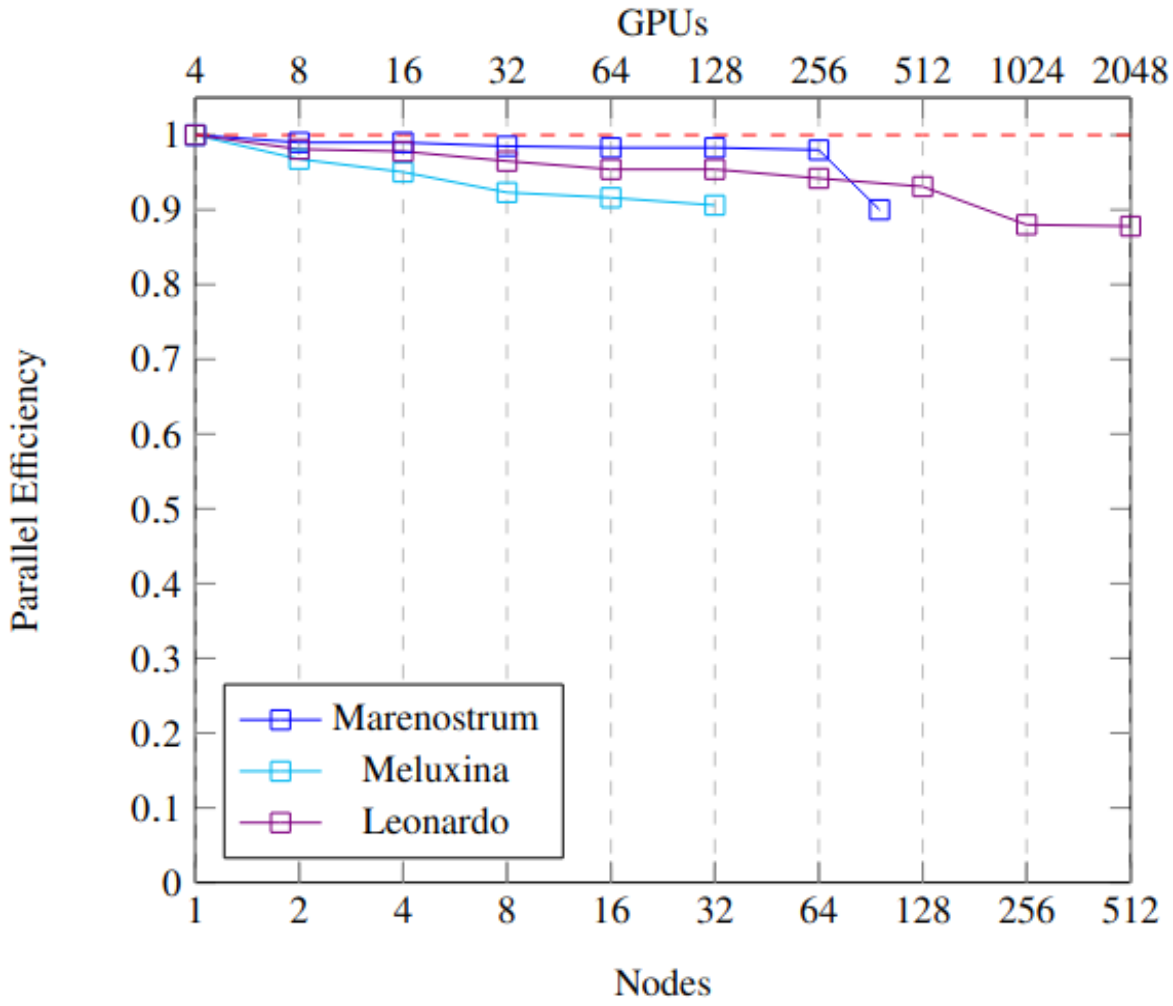
- **System**: Leonardo Booster, MareNostrum, MeluXina vs Leonardo DCGP
- **Application**: performance at increasing number of nodes at fixed resolution per node
- **Speedup**: about 10



Rossazza+26

WEAK SCALING: PARALLEL EFFICIENCY

- **System**: Leonardo / MeluXina / MareNostrum
- **Max #cores used**: 2048 GPUs (512 nodes)
- **Scaling type**: weak
- **Performance**: >85%



Rossazza+26

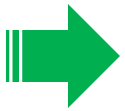
OPENACC VS OPENMP

- OpenMP added to provide GPU support for non-NVIDIA vendors
- Mini-app tested on local workstation and LUMI (40% speedup with respect to OpenACC)
- Optimized for AMD GPUs using Unified Memory
- Raised the collapse number from 2->3 (removed inner level of parallelism)

```
      *** PREVIOUS ACC only version **
      ********************************

#pragma acc parallel loop collapse(2) \
        present (…)
for (k = kbeg; k <= kend; k++){
for (j = jbeg; j <= jend; j++){

  long int offset = ni*(j+nj*k)
  ...
  <array offsetting>
  #pragma acc loop
  for (i = ibeg; i <= iend; i++){
    <code>
    ...
```



```
            *** ACC + OMP ADAPTED ***
            *************************

#pragma acc parallel loop collapse(3) \
        present (…)

#pragma omp target teams distribute \
        parallel for collapse(3) present (…)
for (k = kbeg; k <= kend; k++){
for (j = jbeg; j <= jend; j++){
for (i = ibeg; i <= iend; i++){
  long int offset = ni*(j+nj*k)
  ...
  <array offsetting>
  <code>
  ...
```

PLUTO & GPLUTO: PHYSICS MODULARITY

		PLUTO →	gPLUTO
Basic physics	HD	**Yes**	**Yes**
	MHD	**Yes**	**Yes**
	RHD	**Yes**	**Yes**
	RMHD	**Yes**	**Yes**
	ResRMHD	**Yes**	**Yes**
Dissipation properties	Viscosity	**Yes**	**No** (planned)
	Thermal conduction	**Yes**	**No** (planned)
	Resistivity / Hall MHD	**Yes**	**No** (in progress)
EoS / Thermodynamics	Ideal / isothermal gas	**Yes**	**Yes**
	Synge gas	**Yes**	**Yes**
	Non constant gamma	**Yes**	**No**
	Cooling	**Yes**	**Yes**
Particles	Cosmic Rays	**Yes**	**No** (planned)
	Dust	**Yes**	**No** (in progress)
	Lagrangian Particles	**Yes**	**Yes**

PLUTO & GPLUTO: ALGORITHM MODULARITY

		PLUTO →	gPLUTO
Time stepping	Hancock / Characteristic tracing RK 2,3,4	**Yes** **Yes**	**No** **Yes**
Reconstruction	Linear Parabolic WENO3 / WENO5 Monotonicity Preserving (5th)	**Yes** **Yes** **Yes** **Yes**	**Yes** **Yes** **Yes** **Yes**
Riemann solver	Two shock Roe TVDLF / HLL(C/D) GFORCE	**Yes** **Yes** **Yes** **Yes**	**Yes** **Yes** **Yes** **Yes**
High order method	4th order Finite Volume	***Dev***	**Yes**
$\nabla \cdot B = 0$	8 Wave Generalized Lagrangian Multiplier (GLM) Constrained Transport	**Yes** **Yes** **Yes**	**Yes** **Yes** **Yes**
Poisson solver		**No**	**Yes**
FARGO	(Fast Advection Scheme for Rotating Objects)	**Yes**	**Yes**
ShearingBox	(Local model for differentially rotating disk)	**Yes**	**No**

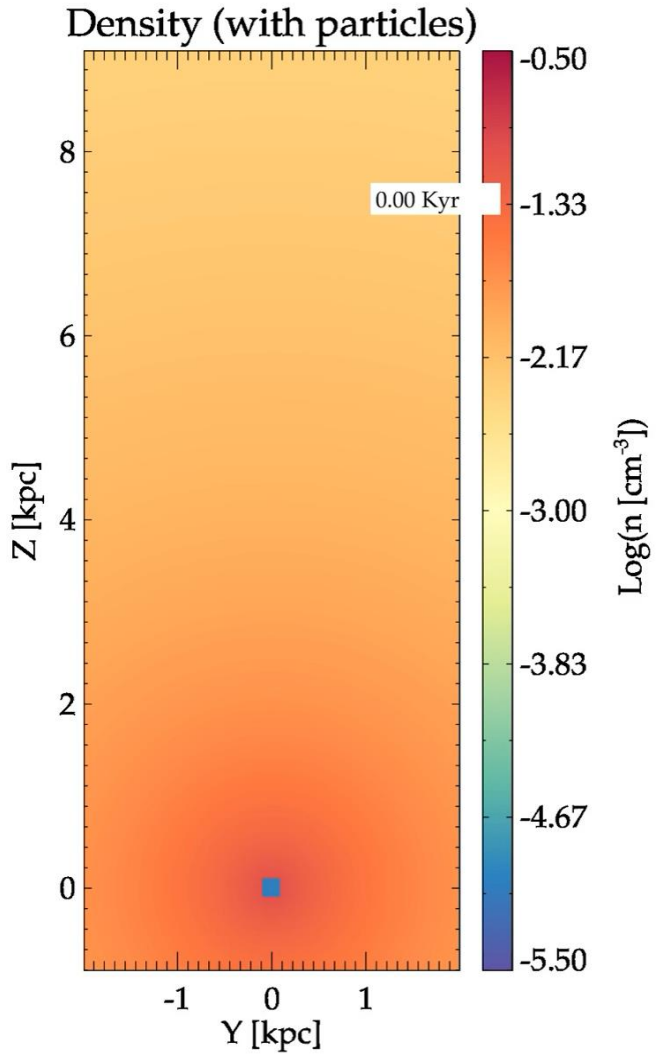
PLUTO & GPLUTO: GRID, I/O, COMMUNICATION

		PLUTO →	gPLUTO
Geometry	Cartesian Cylindrical Spherical	**Yes** **Yes** **Yes**	**Yes** **Yes** **Yes**
Non-uniform grid	Fixed transformations Mapped grids	**Yes** **No**	**No** **Yes**
Adaptive Mesh Refinement		**Yes**	***Dev***
Output file format	Vtk Binary HDF5	**Yes** **Yes** **Yes**	**Yes** **Yes** **Yes**
Input	Restart Externally produced data files	**Yes** **Yes**	**Yes** **Yes**
Parallel communication (MPI / NCCL)	Synchronous Asynchronous	**Yes** **No**	**Yes** **Yes**

LAGRANGIAN PARTICLE MODULE

Hybrid framework to bridge the gap between micro- and macro-scale plasma descriptions (Vaidya+18; Mukherjee+21).

- Eulerian RMHD + sub-grid Lagrangian prescription for Non-Thermal Particles: advection + spectral evolution (adiabatic and radiative losses, shock acceleration, magnetic reconnection) + beaming.
- Consistent modelling of observables: emission and polarization, maps, spectra and lightcurves.
- Fundamental to exploit future observational facilities (CTA, SKA...)



PORTING THE LP MODULE

OPENACC

Re-design of the particle structure: for N_p particles of *n* parameters we define *n* arrays for easy array-like access

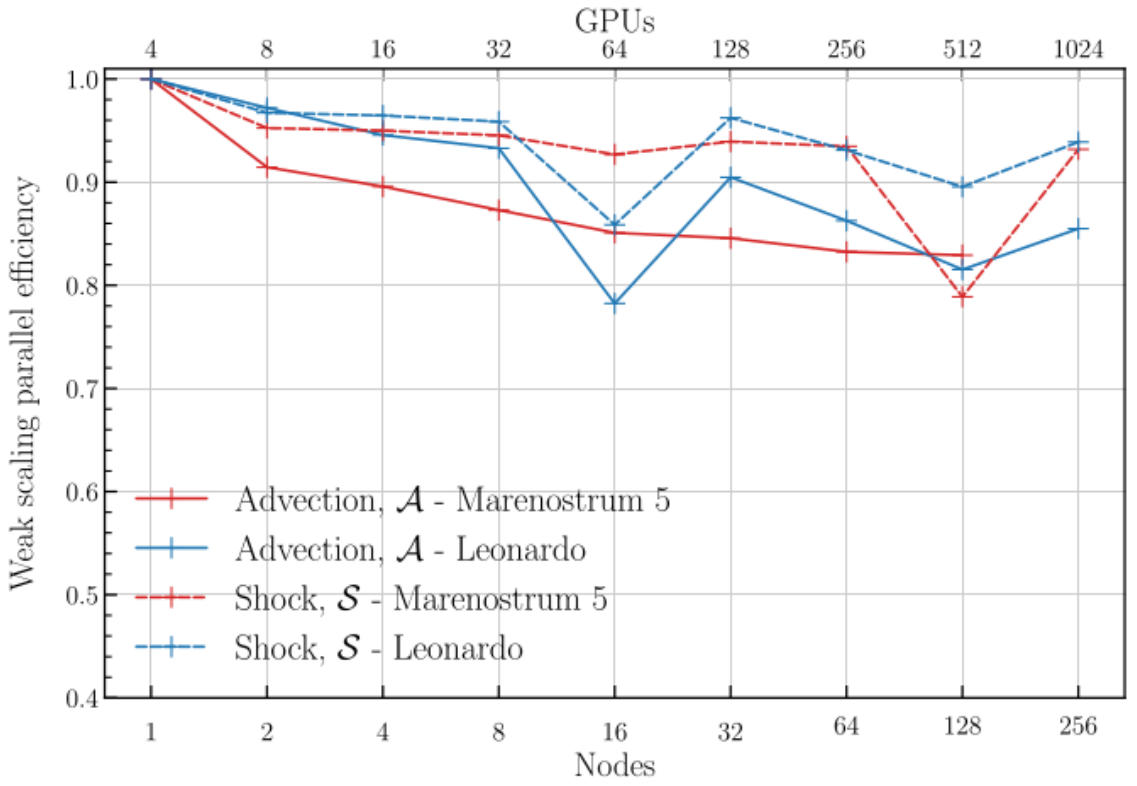
- Parallelization is based on the underlying domain decomposition
- **Challenges**:
  - **Data locality**
  - **Coalesced access to memory**: re-ordering arrays after particle creation/deletion
  - **Minimize memory (de)allocation:** chunked memory allocation

Suriano+26a

PORTING THE LP MODULE: WEAK SCALING

2 TESTS: ADVECTION AND SHOCK

- **System**: Leonardo / MareNostrum
- **Max #cores used**: 1024 GPUs (256 nodes)
- **Scaling type**: weak
- **Performance**: >80% with MareNostrum

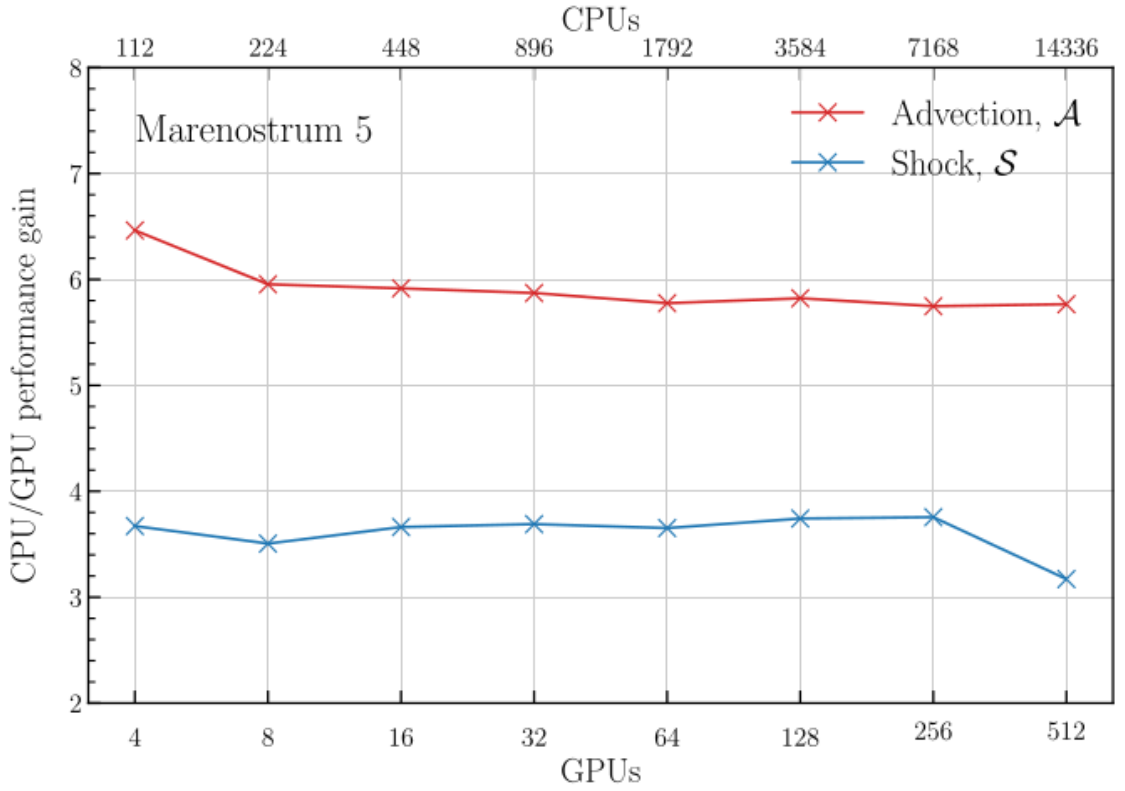


Suriano+26a

PORTING THE LP MODULE: WEAK SCALING

2 TESTS: ADVECTION AND SHOCK

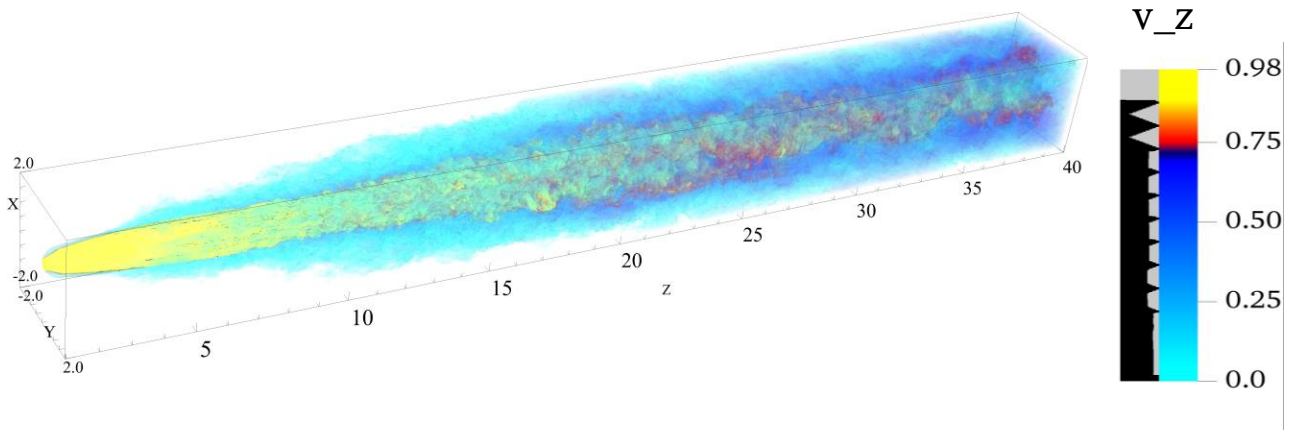
- **System**: Leonardo / MareNostrum
- **Max #cores used**: 1024 GPUs (256 nodes)
- **Scaling type**: weak
- **Performance**: >80% with MareNostrum
- **Speedup**: >3

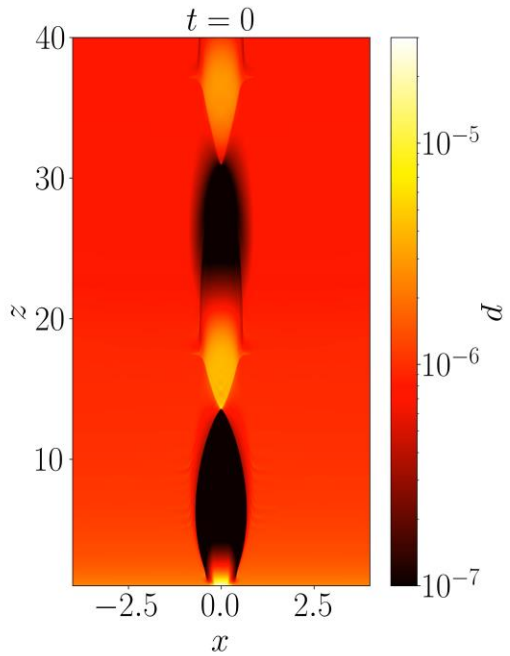


Suriano+26a

3 SCIENTIFIC CASES

SCIENTIFIC CASE: JET INSTABILITIES





Costa+24;25

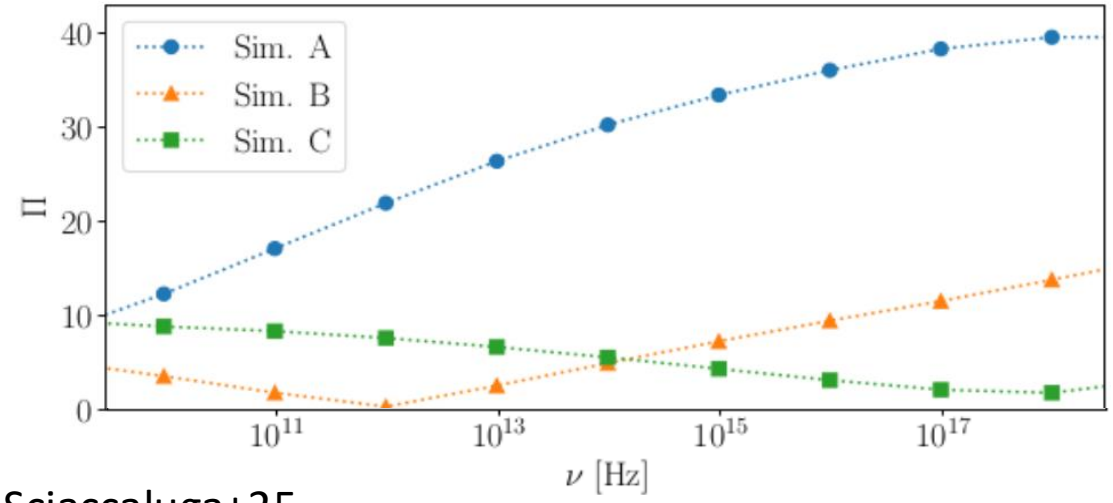
2D vs 3D RHD simulations of instabilities in recollimated jets

- Low-power, low-magnetized jets unstable downstream of recollimation
- Instabilities lead to: dissipation, spine-sheath configuration, entrainment, fast deceleration
- Interpretation of jet deceleration in FROs, while higher power FRI jets decelerate on larger scales

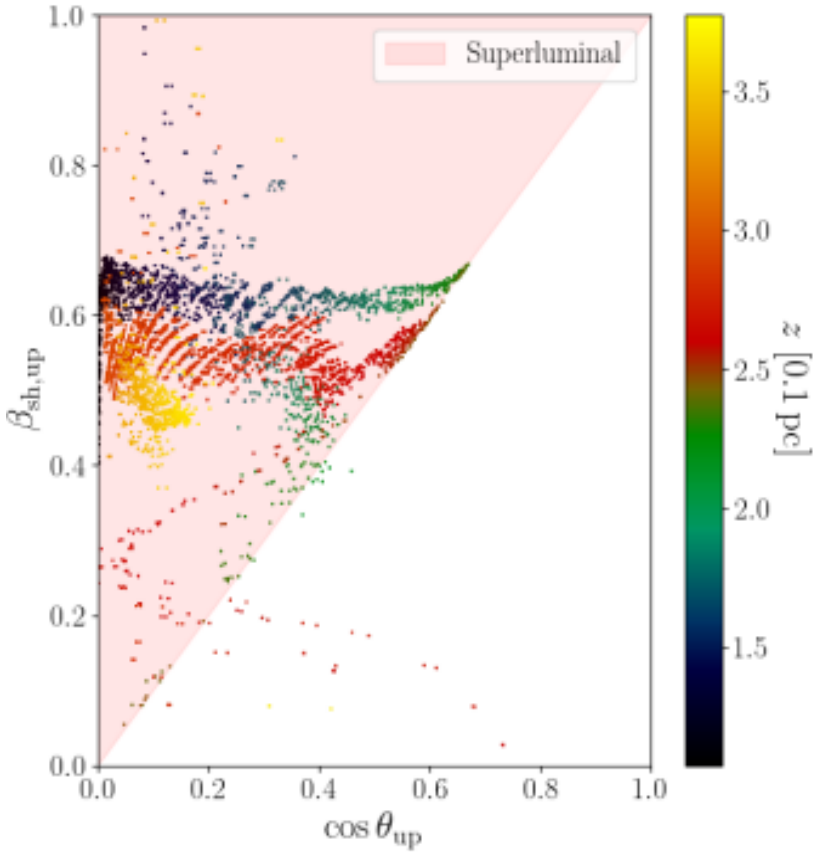
SCIENTIFIC CASE: SHOCK ACCELERATION

Target: synchrotron polarization from recollimated jets

- Small scale turbulence required for efficient shock acceleration
- Chromaticity (Sim.B) matches data of (E)HBL

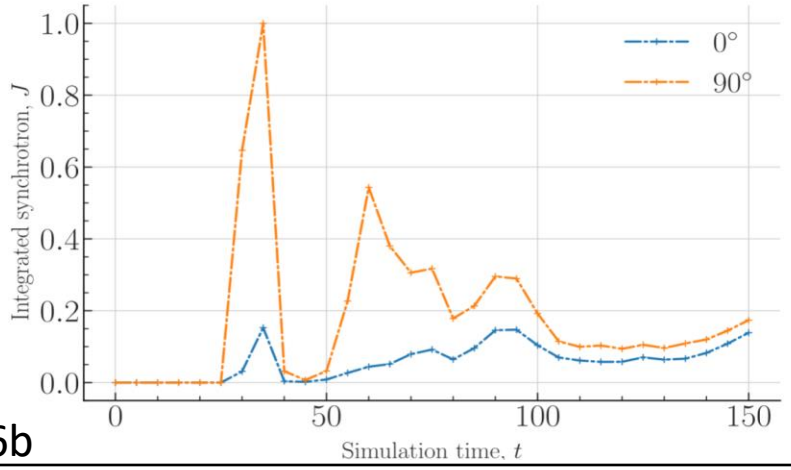


Sciaccaluga+25

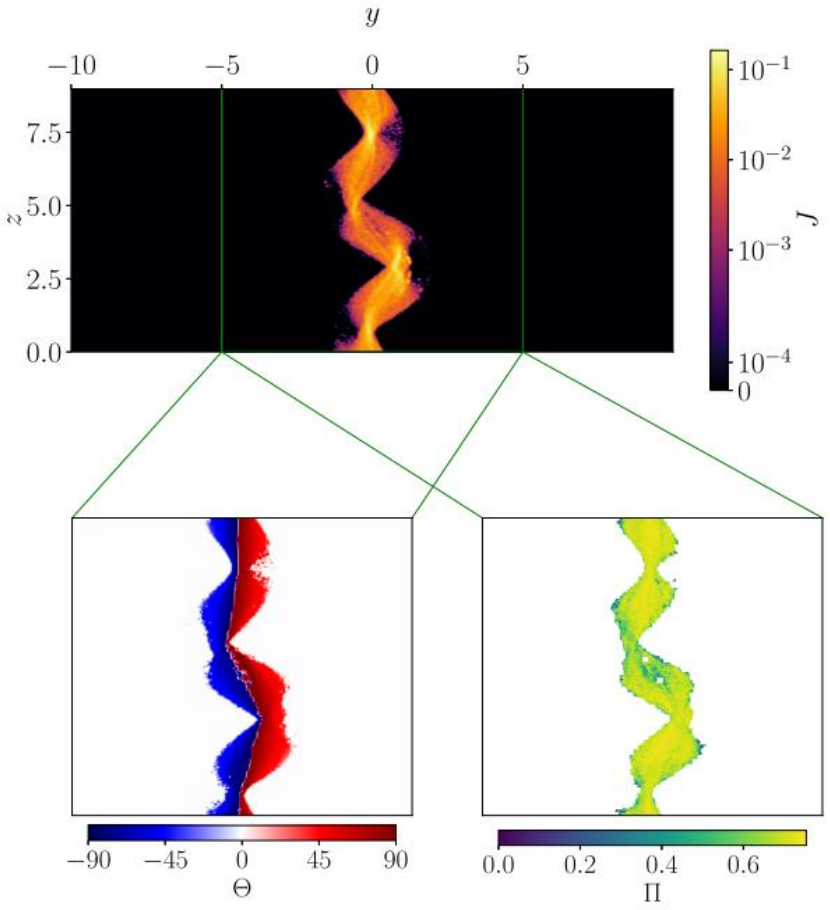


SCIENTIFIC CASE: MAGNETIC RECONNECTION

- Magnetic reconnection effects newly included in a hybrid RMHD + LP framework
- Sampling local field quantities to estimate steepness, efficiency, and γ_max
- Fundamental for distinguishing MR from DSA models in astrophysical sources



Suriano+26b



THANK YOU!