

DYABLO: A NEW GENERAL-PURPOSE GPU-ACCELERATED HYDRODYNAMICAL CODE FOR ASTROPHYSICS

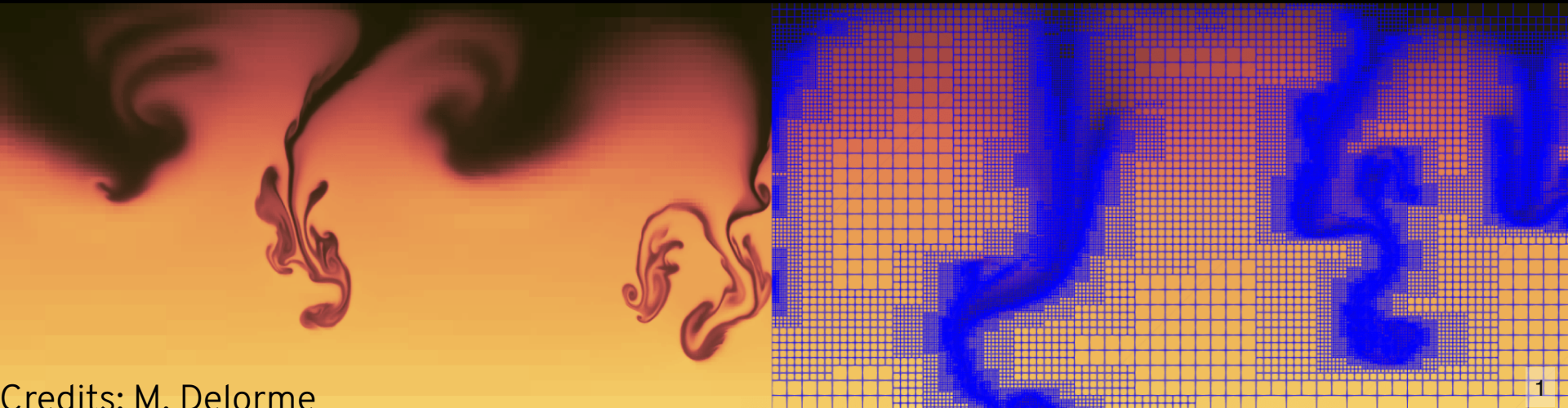
Corentin Cadiou¹ (cadiou@iap.fr)

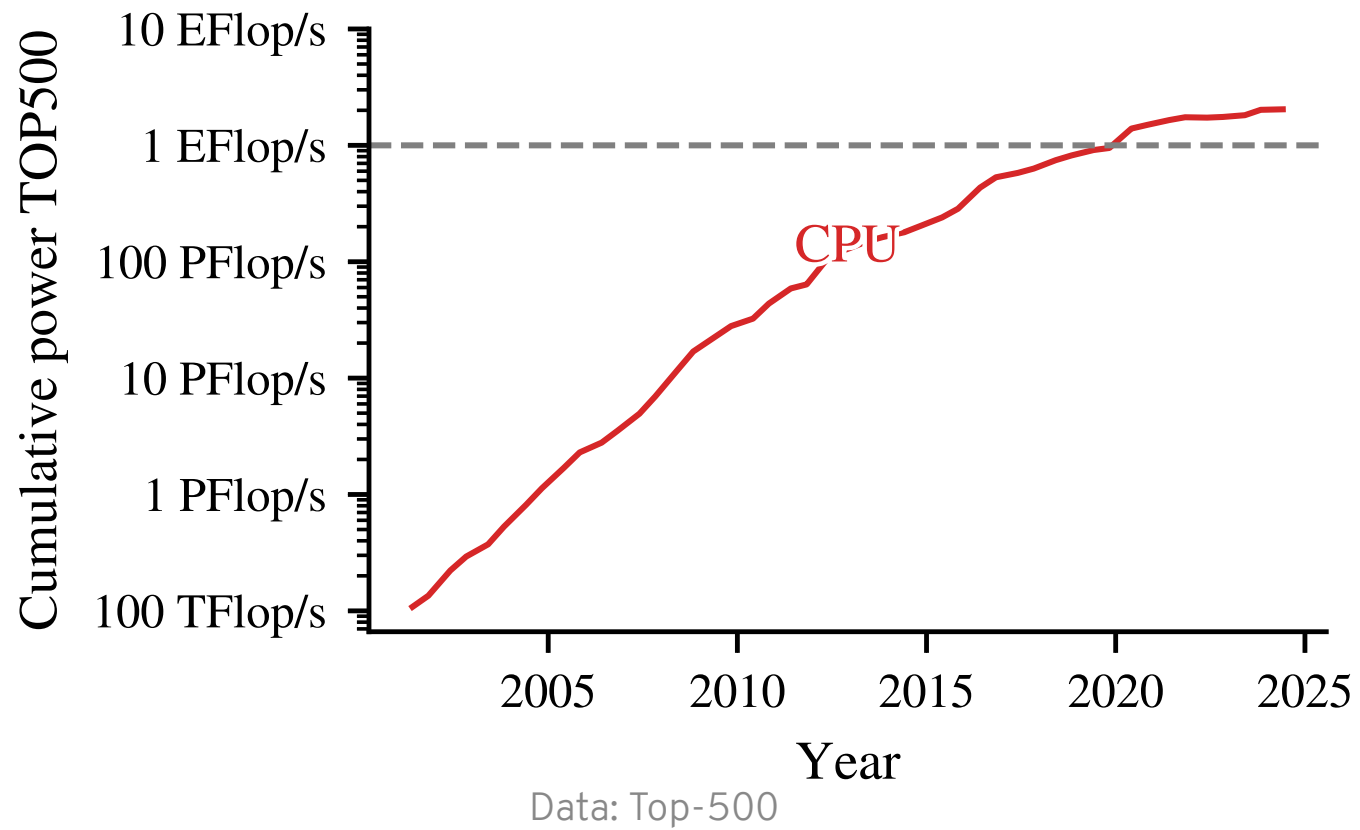
Lead developers: Maxime Delorme (CEA), Arnaud Durocher (CEA)

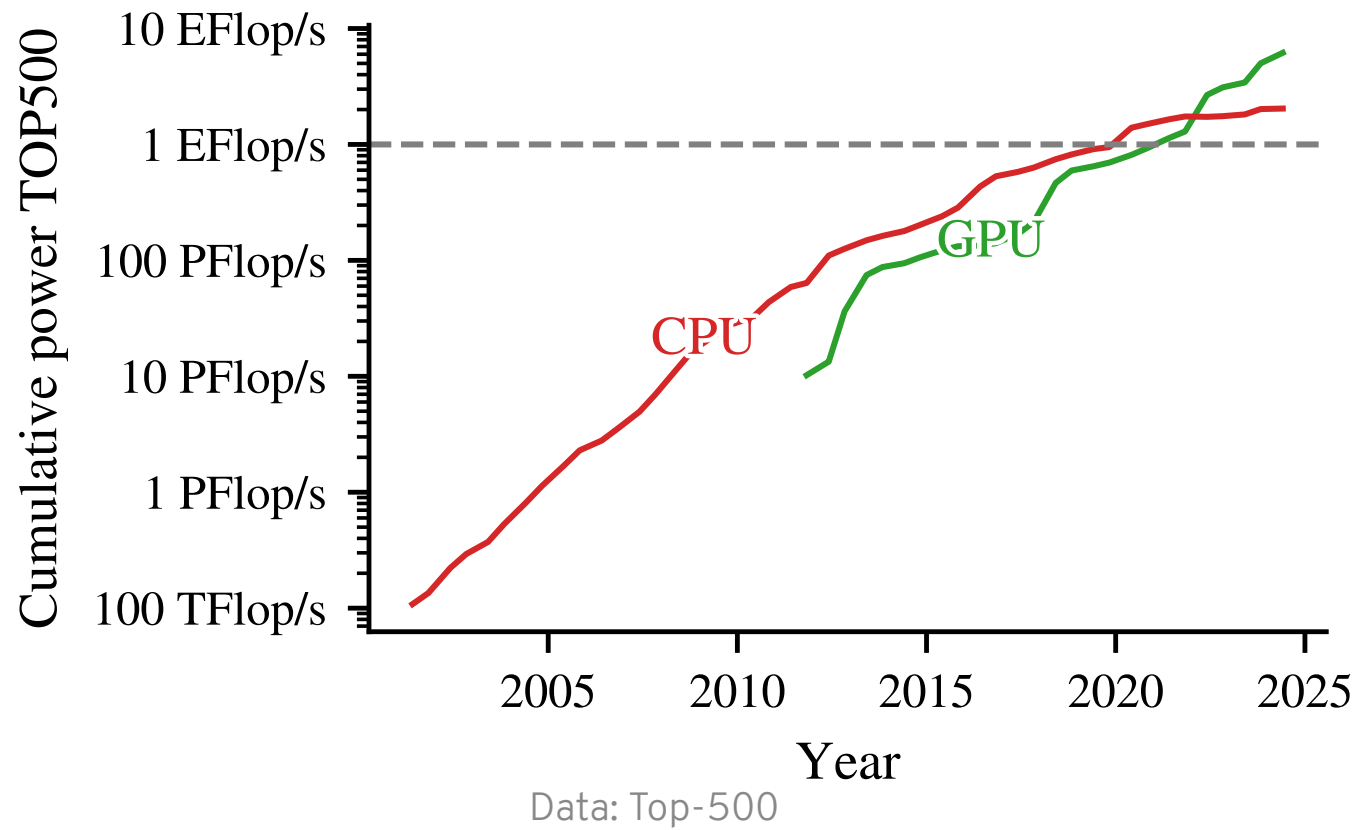
Scientific collaborators: Dominique Aubert (ObAS), Grégoire Doebele (CEA), Olivier Marchal (ObAS), San Han (IAP), Guillaume Tcherniatsky (IAP), Harley Katz (U. Chicago)



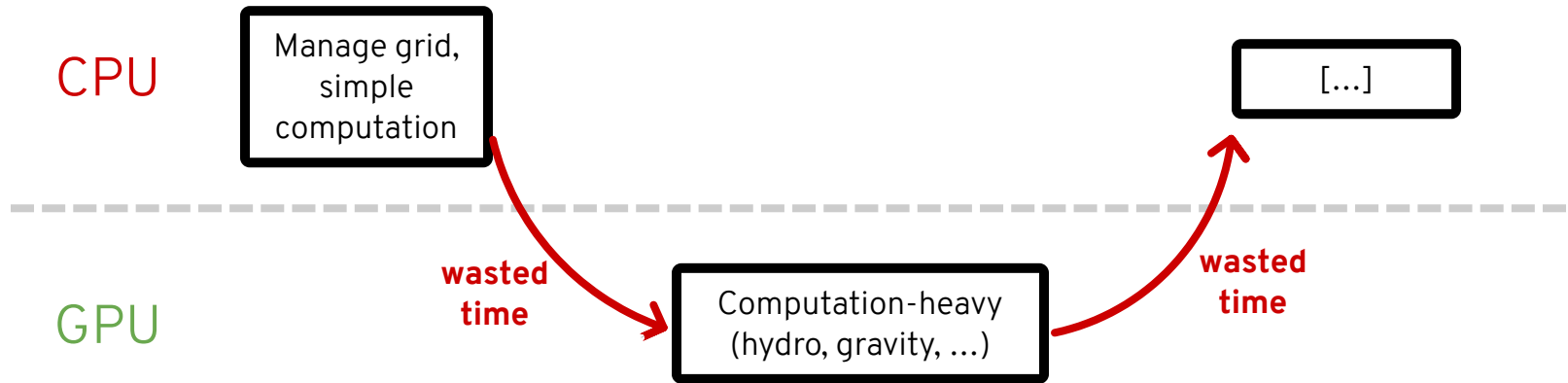
¹Institut d'Astrophysique de Paris, CNRS, France







Typical approach: offloading

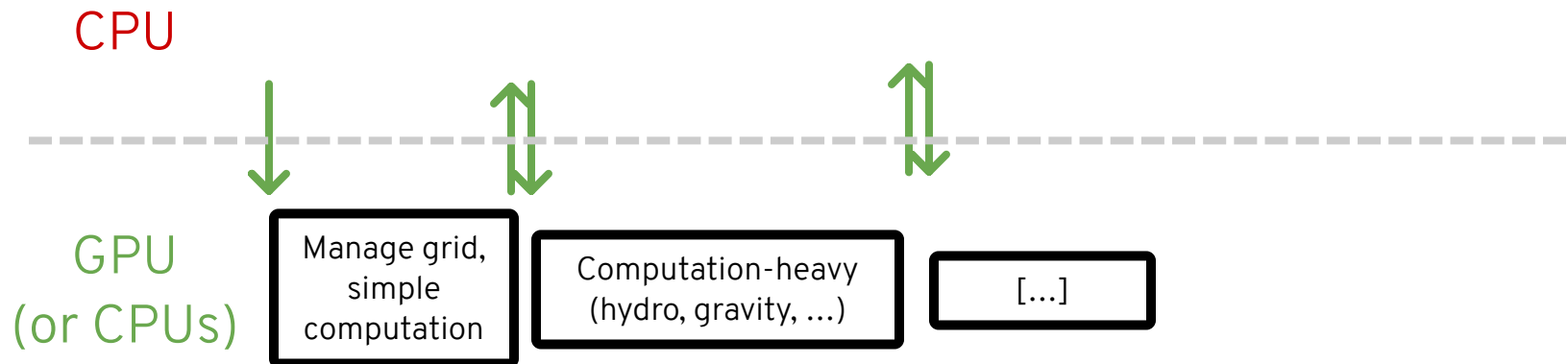


Dyablo's approach:

Amdahl's law: latency kills gains of parallelisation

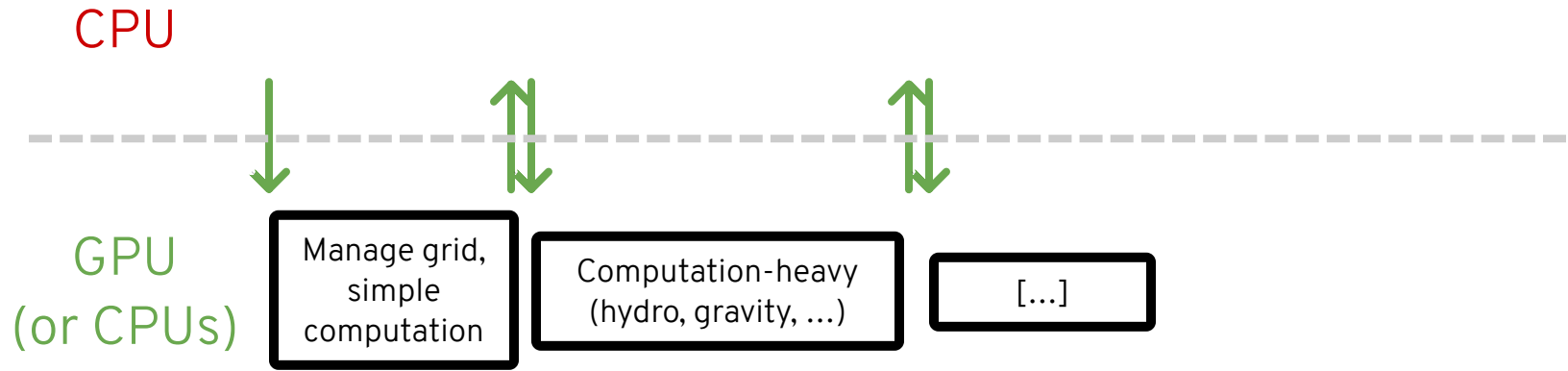
Typical approach: offloading

Dyablo's approach: “true” GPU computing, CPU as a puppeteer



Typical approach: offloading

Dyablo's approach: “true” GPU computing, CPU as a puppeteer



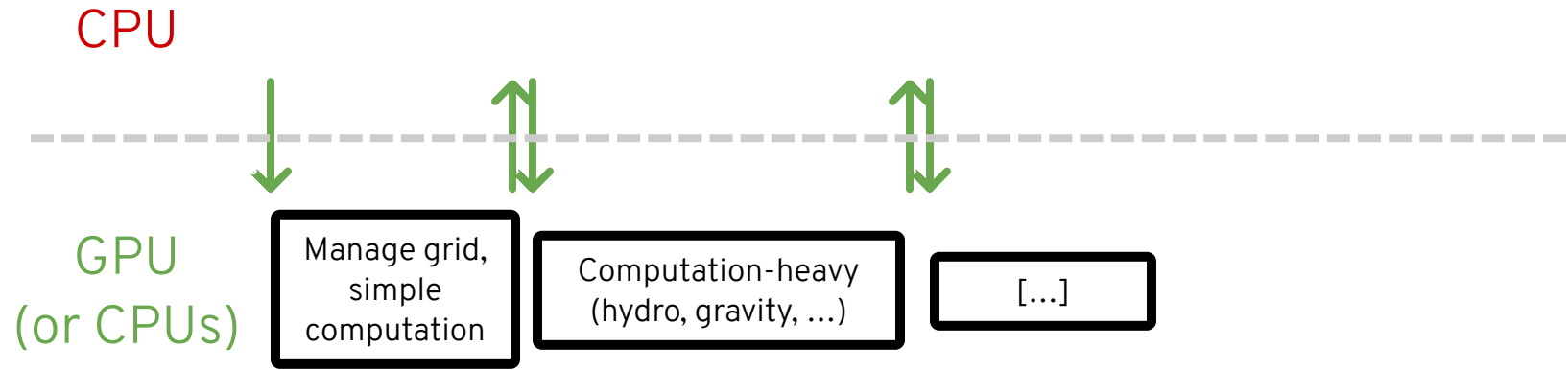
1. SEPARATION OF CONCERNS

Physicists: write subgrid models

Computer scientists: optimize the code

Typical approach: offloading

Dyablo's approach: “true” GPU computing, CPU as a puppeteer



1. SEPARATION OF CONCERNS

Physicists: write subgrid models

Computer scientists: optimize the code

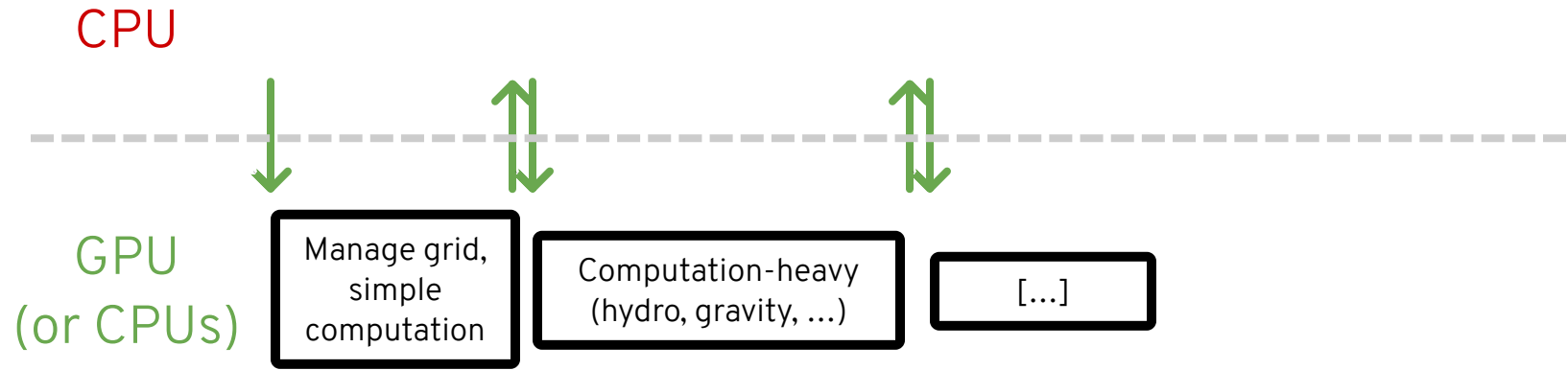
2. ABSTRACTION

Physics: independent of data layout, parallelization, ...

Code should provide high-level APIs

Typical approach: offloading

Dyablo's approach: “true” GPU computing, CPU as a puppeteer



1. SEPARATION OF CONCERNS

Physicists: write subgrid models

Computer scientists: optimize the code

2. ABSTRACTION

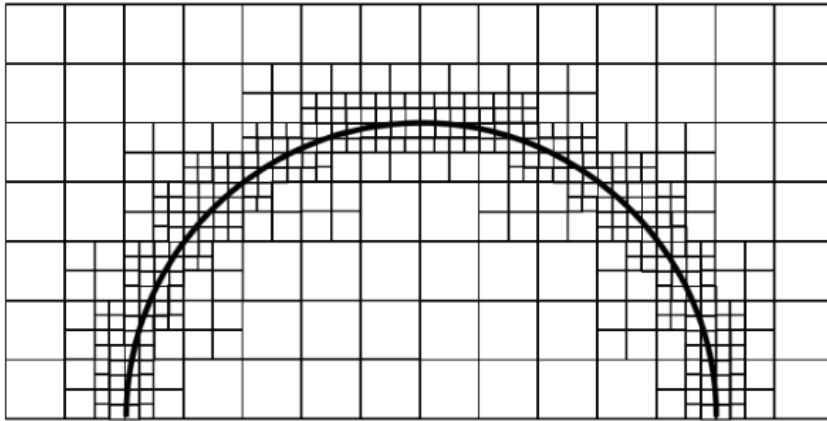
Physics: independent of data layout, parallelization, ...

Code should provide high-level APIs

3. MODULARITY

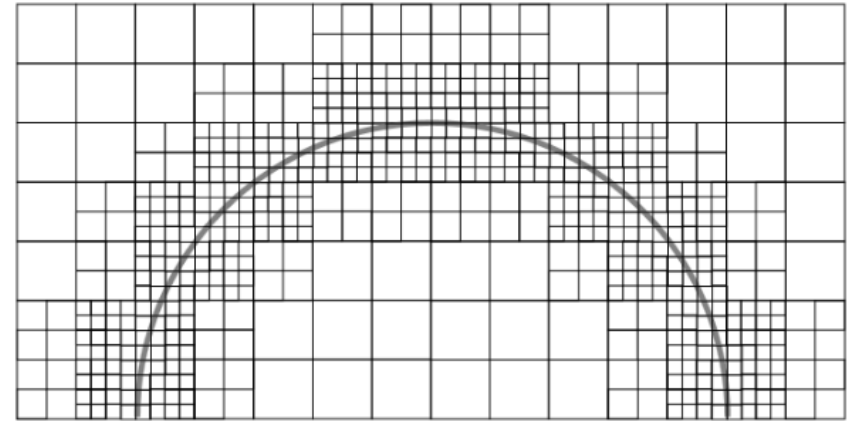
Implementations & models should be interchangeable depending on needs

DYABLO: OVERALL DESIGN



Cell-based AMR with 382 cells

Traditional oct-based AMR



Block-based AMR with 596 cells

Collection of $N \times M \times L$ blocks

Cell or block-based
Adaptive **M**esh **R**efinement

DYABLO: OVERALL DESIGN



```
1 [...]
2
3 // Memory abstracted
4 Kokkos::View<double**> U(Ncell,
   Nfields);
5
6 // Parallelism abstracted
7 Kokkos::parallel_for(Ncell,
   KOKKOS_LAMBDA(size_t iCell) {
8     U(iCell, IRho) += ...
9 });
10
11 [...]
```



or



or



Single node / GPU

DYABLO: OVERALL DESIGN



```
1 [...]
2
3 // Memory abstracted
4 Kokkos::View<double**> U(Ncell,
   Nfields);
5
6 // Parallelism abstracted
7 Kokkos::parallel_for(Ncell,
   KOKKOS_LAMBDA(size_t iCell) {
8     U(iCell, IRho) += ...
9 });
10
11 [...]
```

Single node / GPU



or



or

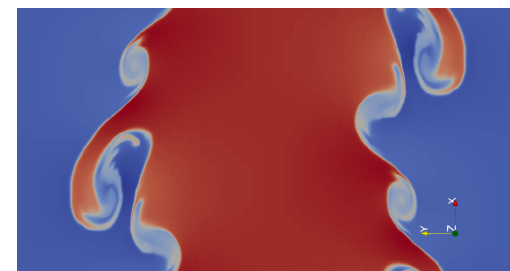


DYABLO: OVERALL DESIGN



DYABLO: IMPLEMENTED PHYSICS

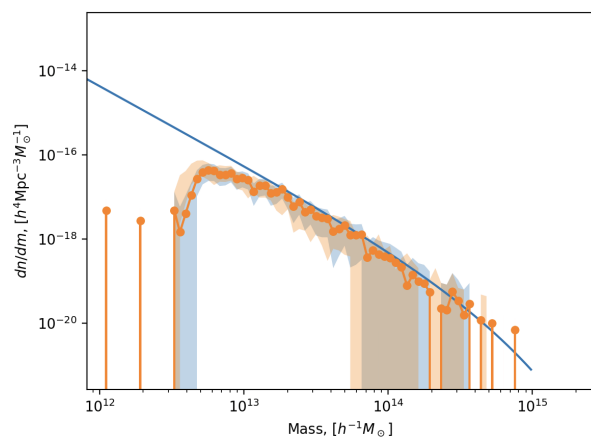
- Block-based adaptive mesh-refinement,
- Hydrodynamics / MHD / radiative transfer
- (Self-)Gravity
- Particles,
- Diffusion terms: thermal conduction, viscosity,
- Cosmology: Comoving coordinates,
 - Ongoing –
- *Star formation and feedback,*
- *(Non-equilibrium) thermochemistry,*
- *Cosmic rays, ...*



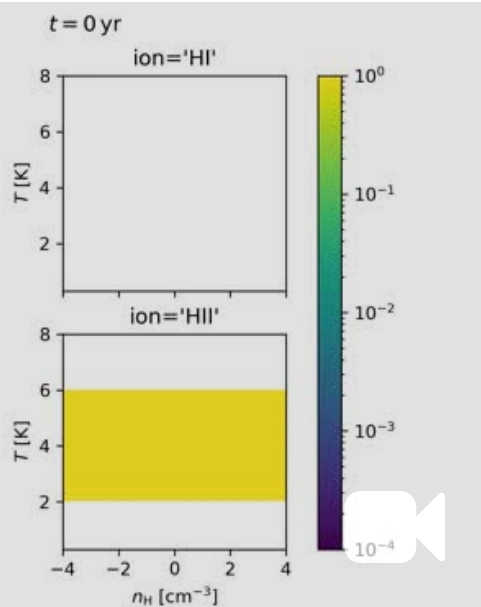
KH instability (Delorme)

<https://www.youtube.com/embed/P40OxuMQjFA?enablejsapi=1>

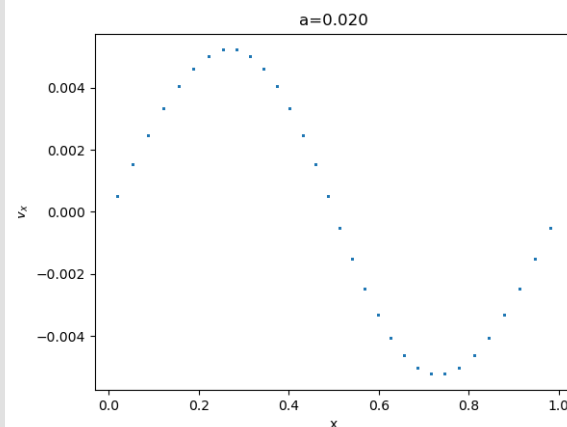
Solar convection (Delorme)



Halo mass function (Aubert & Marchal)



Thermochem. (Cadiou & Katz)

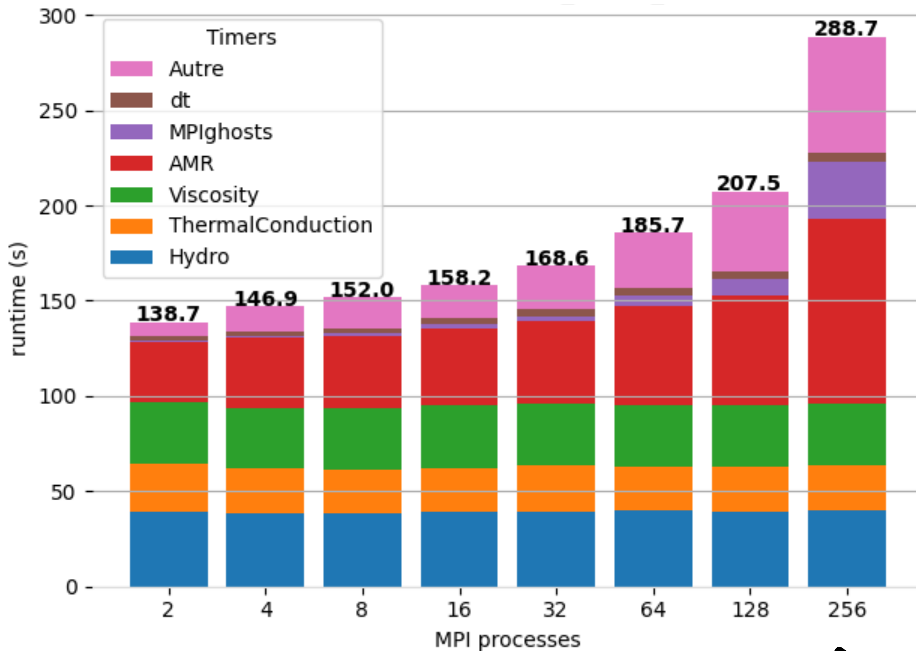


Zeldovich pancake (Aubert & Marchal)

DYABLO: HOW DOES IT SCALE?

– weak scaling: solar convection

Ad Astra CPU (2 x AMD Epyc Genoa 96 cores)

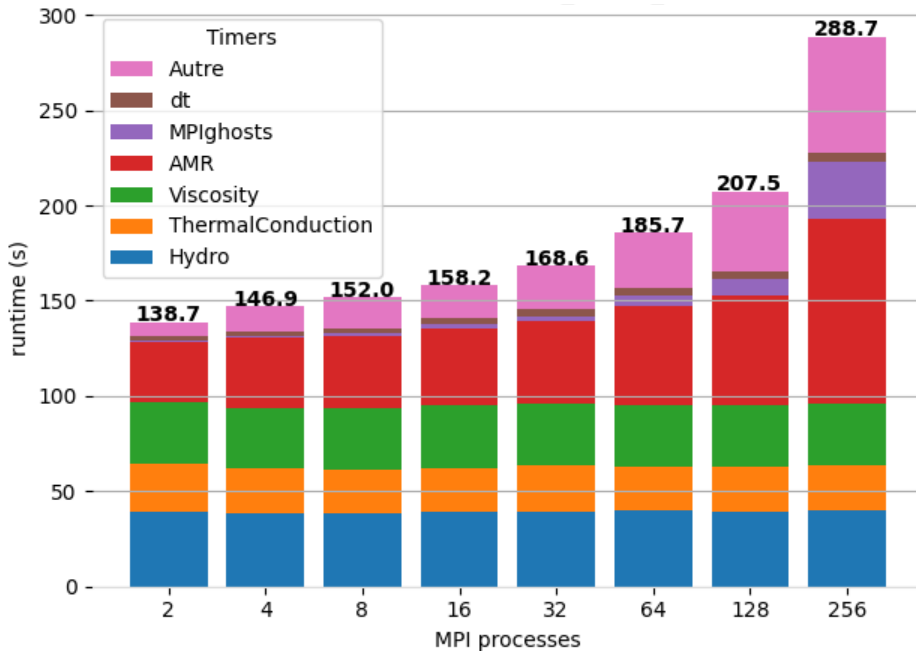


24,576 cores

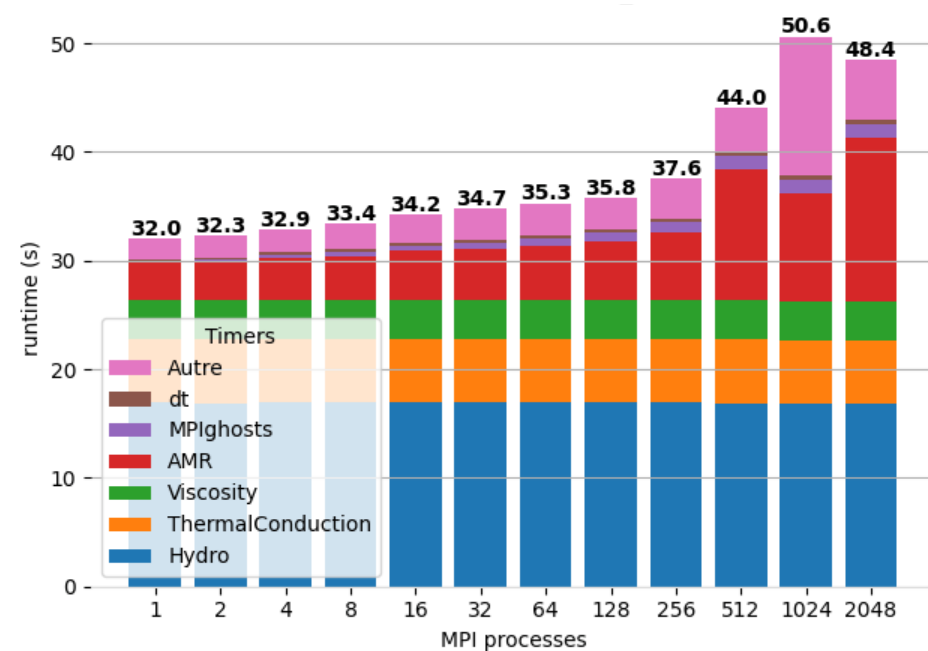
DYABLO: HOW DOES IT SCALE?

– weak scaling: solar convection

Ad Astra CPU (2 x AMD Epyc Genoa 96 cores)



Ad Astra GPU (4 AMD MI250 x2 GCD)

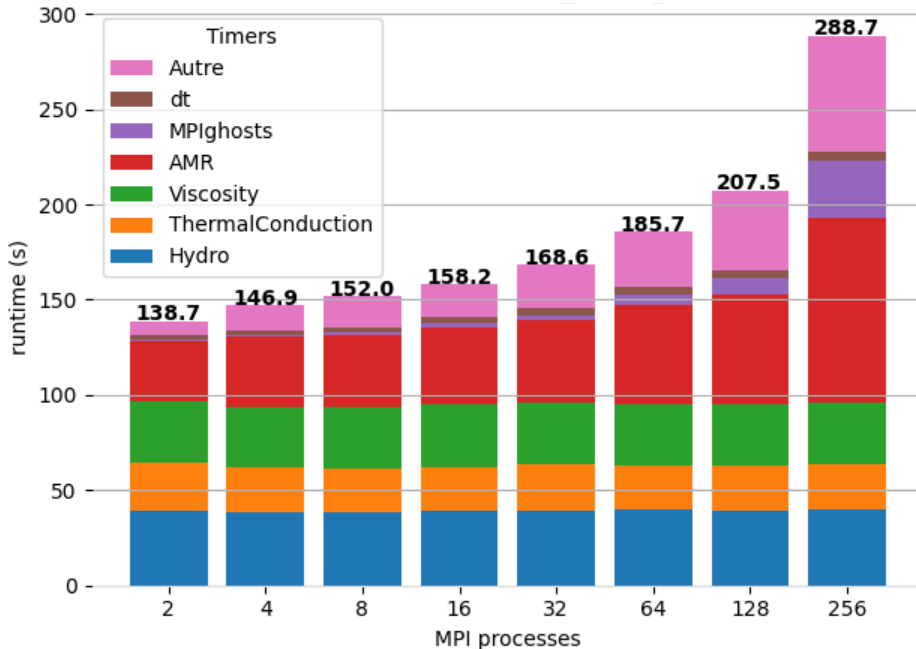


Note: some updates in the code between
left & right, raw perfs **cannot** be
compared directly

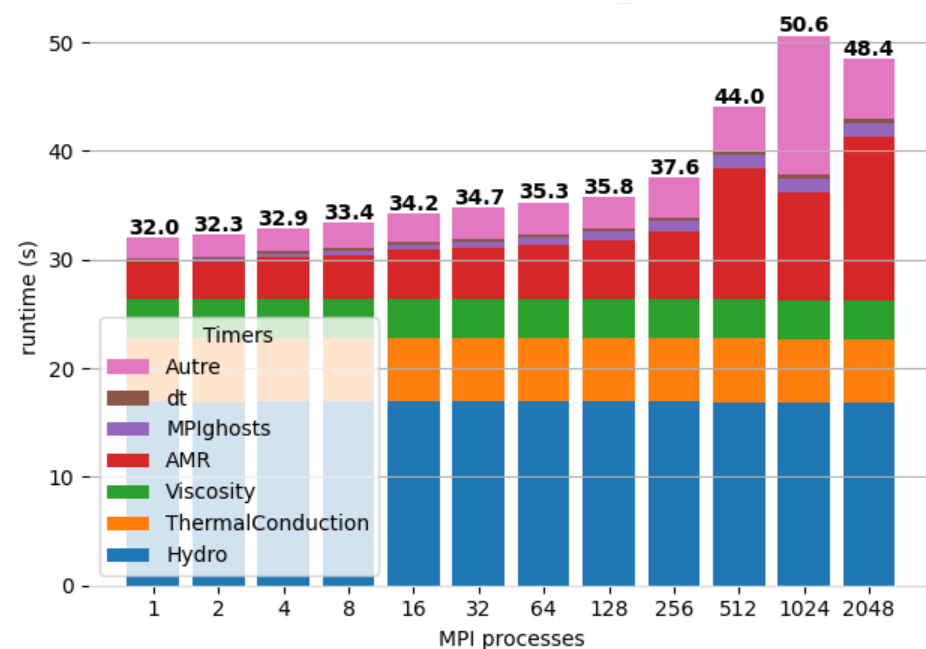
DYABLO: HOW DOES IT SCALE?

– weak scaling: solar convection

Ad Astra CPU (2 x AMD Epyc Genoa 96 cores)



Ad Astra GPU (4 AMD MI250 x2 GCD)



Note: some updates in the code between
left & right, raw perfs **cannot** be
compared directly

Hydro only: ~ 200 Mcell/s

AthenaK ~ 1000 Mcell/s, Stone+24

Cholla-MHD ~ 200 Mcell/s, Caddy+24,

AREPO-RT ~ 1 Mcell/s, Zier+24

Shamrock ~ 10 Mcell/s, David-Cl  ris+25

DYABLO: A NEW GENERAL-PURPOSE GPU-ACCELERATED HYDRODYNAMICAL CODE FOR ASTROPHYSICS

Physics: hydrodynamics, MHD, RT, gravity, cosmology, ...

Efficient scaling up to 10,000+ CPUs, 1000+ GPUs

Modular and future-proof code



[Dyablo-HPC/Dyablo](https://github.com/Dyablo-HPC/Dyablo)

Corentin Cadiou (cadiou@iap.fr)

