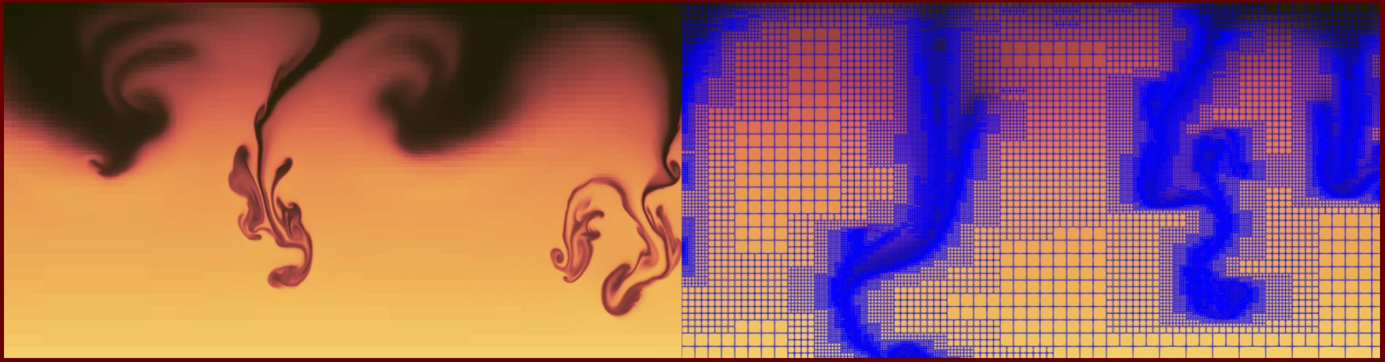
dyablo-Whole Sun/Whole Star

A new simulation code on AMR grids for the simulation of the Sun and solar-like stars on exascale architectures

Maxime Delorme (maxime.delorme@cea.fr)

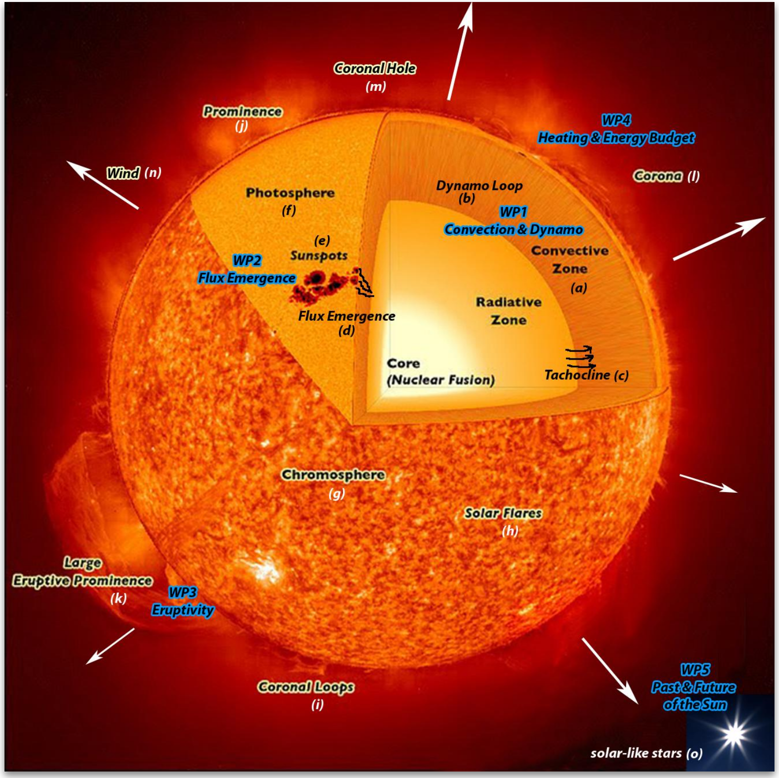
Atelier codes PNPS - Meudon - 30/06/2022

Collaborateurs: Allan-Sacha Brun, Arnaud Durocher, Pierre Kestener, Antoine Strugarek



LDE3
Irfu - CEA Saclay
Institut de recherche sur les lois fondamentales de l'Univers
cea
erc
WHOLE

Yet another code ?

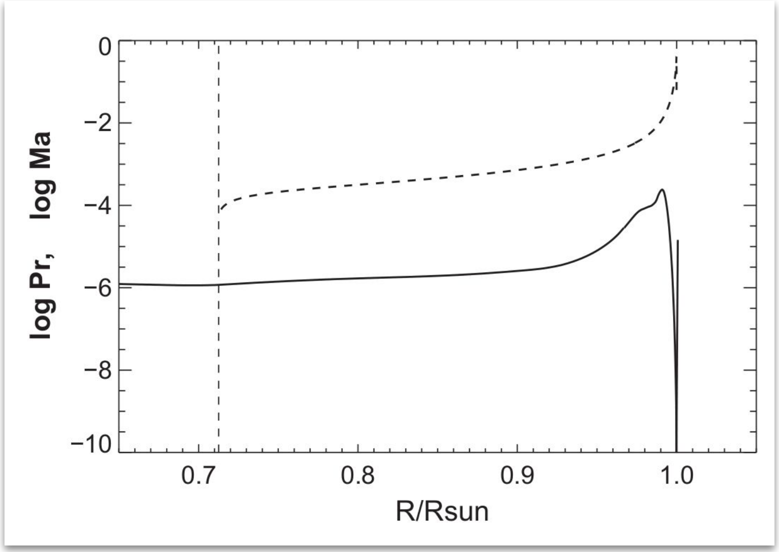


Source: [Whole Sun website]

Shopping list :

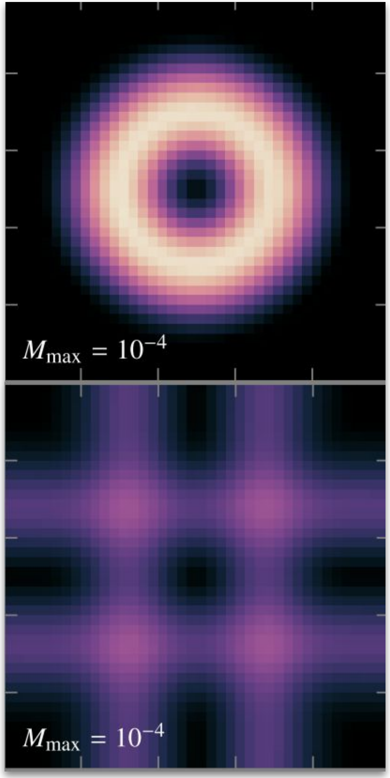
- All-Mach compressible hydrodynamics
- Non-ideal MHD
- Diffusion : viscosity, thermal conduction,
- Rotation
- Radiative transfer
- Non-ideal EOS
- Cartesian and Spherical geometries
- Adaptive mesh-refinement
- “Exascale”-ready
- [Well-balanced, ambipolar-diffusion, etc.]

Why is it so difficult ?



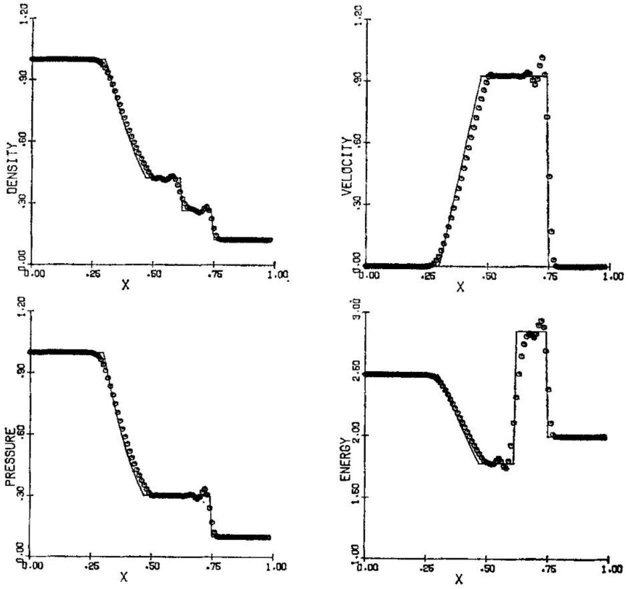
Freytag et al 2012

Low Mach end



Miczek et at 2015

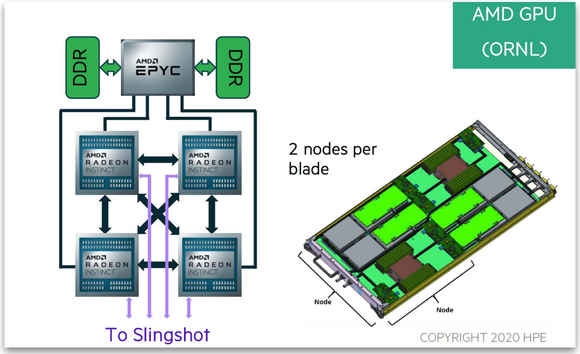
High Mach end



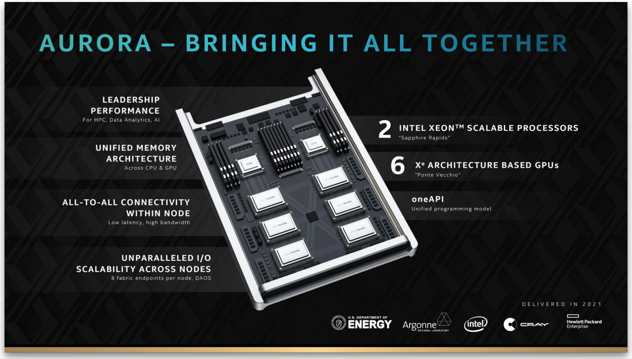
Sod 1978

A case for performance portability

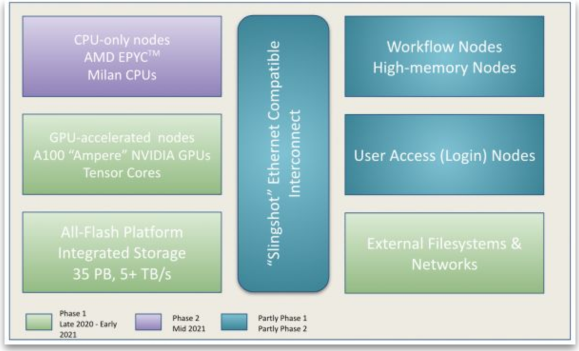
Frontier : AMD CPUs + AMD GPUs

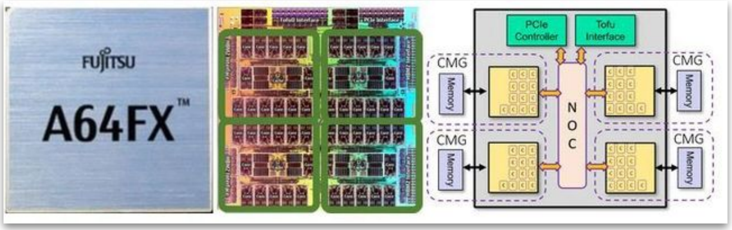


Aurora : Intel CPUs + Intel GPUs

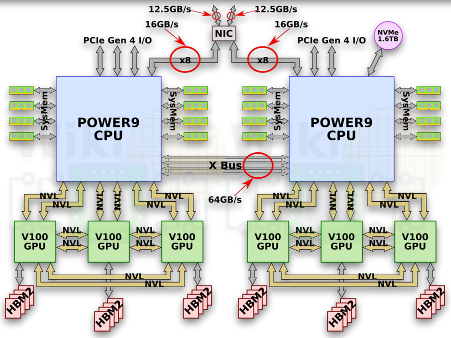


Perlmutter : AMD CPUs + NVidia GPUs





Fugaku : Fujitsu CPUs



Summit : IBM CPUs + NVidia GPUs

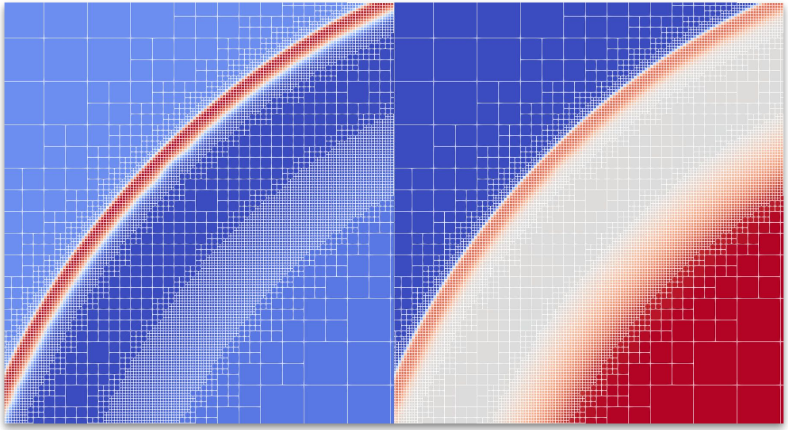
Modern architectures are diverse and require adaptation and portability

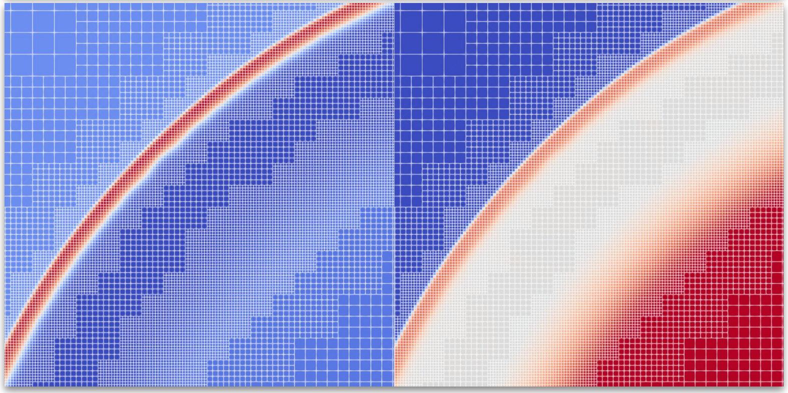
AMR ?

Adaptive **M**esh **R**efinement :

- Allocates more points in interesting[definition needed] regions
- Allows to fit large problems in memory
- Many flavors :
  - **Cell-based**
  - **Block-based**
  - Patch-based
- Main challenges :
  - More difficult algorithmics
  - More complex numerical schemes
  - Difficult to parallelize
  - Usually slower than regular grids
  - What's a sensible refinement criterion ?

Cell-based AMR





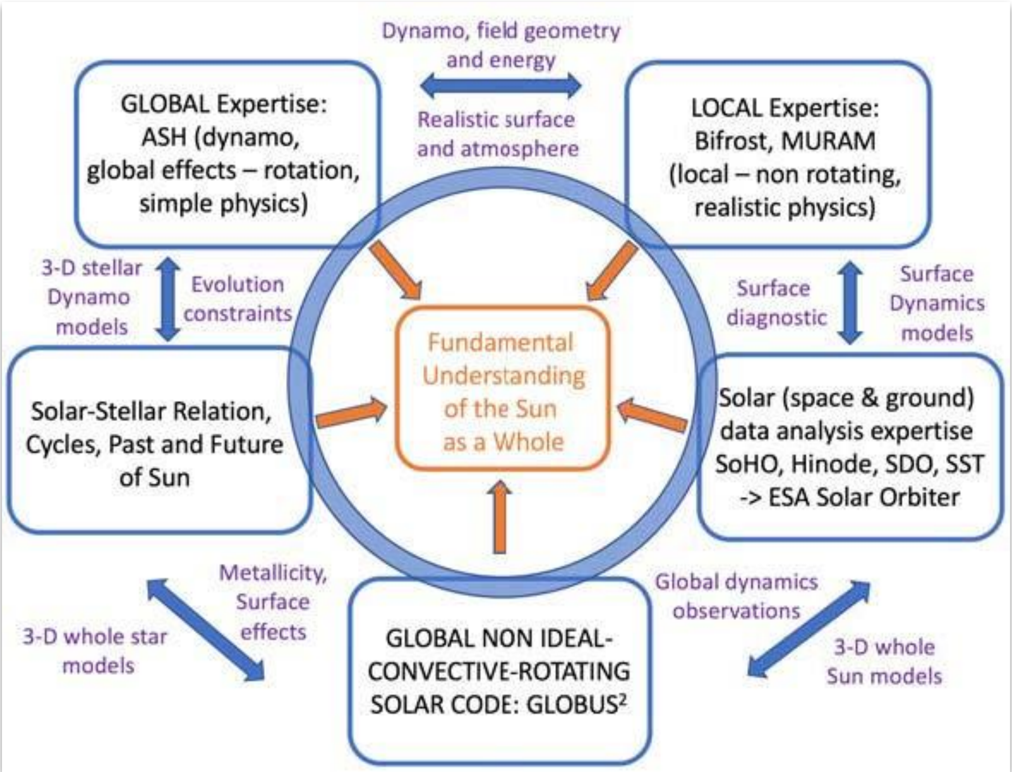
Block-based AMR

And a lot of very good other reasons

Incentive:

- **Global simulations of the Sun**
  - Radiative zone → Corona
- **Multi-scale/multi-physics dynamics**
  - Large variation of temporal and spatial scales
  - Different regimes corresponding to different regions
- **Modularity and ease of use**
  - Testing and implementing new physics
- **Performance portability**
  - Being able to run and be efficient on “““any””” cluster

New code = Modern algorithmics + modern numerical methods



Source: [Whole Sun website]

Whole Sun: design goals and wishlist [2022]

Physics

- **Objective:** Global simulation of the Sun and solar-like stars, from the radiative interior to the corona
- **Ingredients:** MHD, viscosity, gravity, thermal conduction, radiative-transfer, rotation, all-Mach

Numerical methods

- Geometry: Adaptive mesh refinement, multiple geometries
- Finite-volumes, with godunov-type method, multiple solvers (muscl-hancock, rk2/rk3, euler)
- Explicit integration of sources (purely explicit, STS, RKL) or IMEX methods

Software engineering

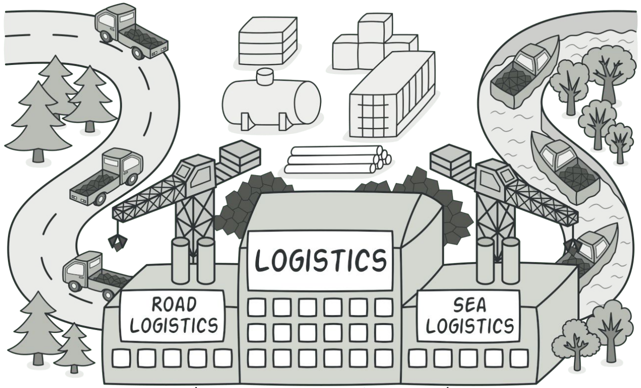
- Performance portable: MPI + shared parallelism
- *"Separation of Concerns"*: Generic AMR tree traversals/reductions
- Modularity: Plugins and factories system

SoC : Separation of Concerns

"We all have a specific job"

- **Physicists do physics**
  - Corollary #1 : Physicists don't do Software engineering, code optimization, GPU code, [...]
  - Corollary #2 : The parts of the code physicists modify should :
    - 1. Have access to simple interfaces to implement/add functionalities
    - 2. Hide all the complexities of the algorithmic machinery
    - 3. Avoid as many side effects as possible, especially on performance.

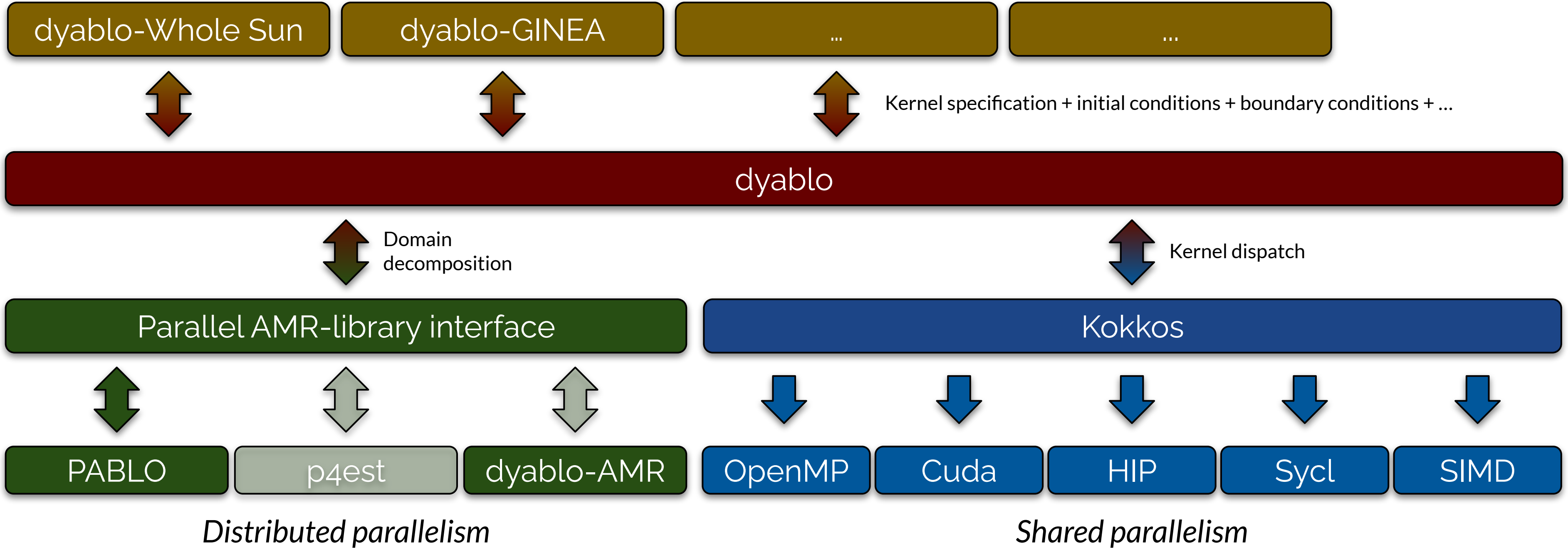
Plugin system and Factories



(Source : Refactoring Guru)

- Abstraction of common parts of the code
- **Factory** : Let the system create the right object at startup
- **Plugin** : Factory + Concrete Products
  - (M)HD solver, Parabolic Terms, Parabolic Solver, Refinement method, IO methods, etc.

dyablo: a high-performance AMR framework



Historically, DYABLO := DYnamics Adaptive mesh refinement CFD applications with PABLO

Available upon request at on gitlab

dyablo-Whole Sun: current state [2022]

Physics

- **Objective:** Global simulation of the Sun, from the radiative interior to the corona
- **Ingredients:** **MHD, viscosity, gravity, thermal conduction**, radiative-transfer, rotation, **all-Mach**

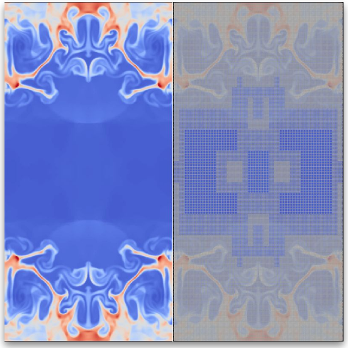
Numerical methods

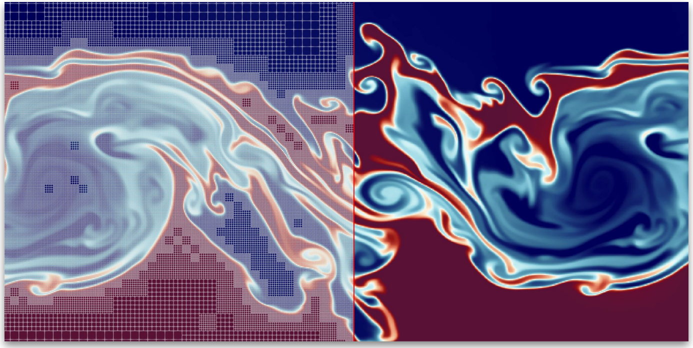
- Geometry: **Adaptive mesh refinement**, multiple geometries
- Finite-volumes, with godunov-type method, multiple solvers (**muscl-hancock, rk2**/rk3, **euler**)
- Explicit integration of sources (**purely explicit**, STS, RKL) or IMEX methods

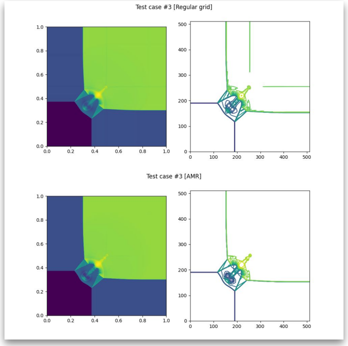
Software engineering

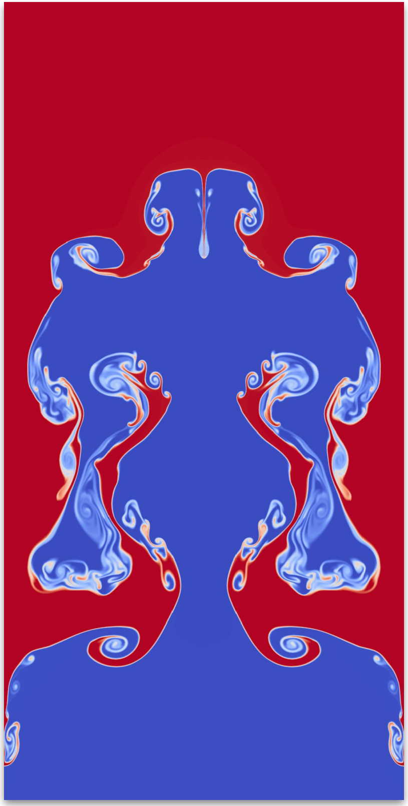
- Performance portable: **MPI + shared parallelism** [CPU intel/AMD; GPU Nvidia]
- *Separation of Concerns*: **Generic AMR tree traversals/reductions**
- Modularity: **Plugins and factories system**

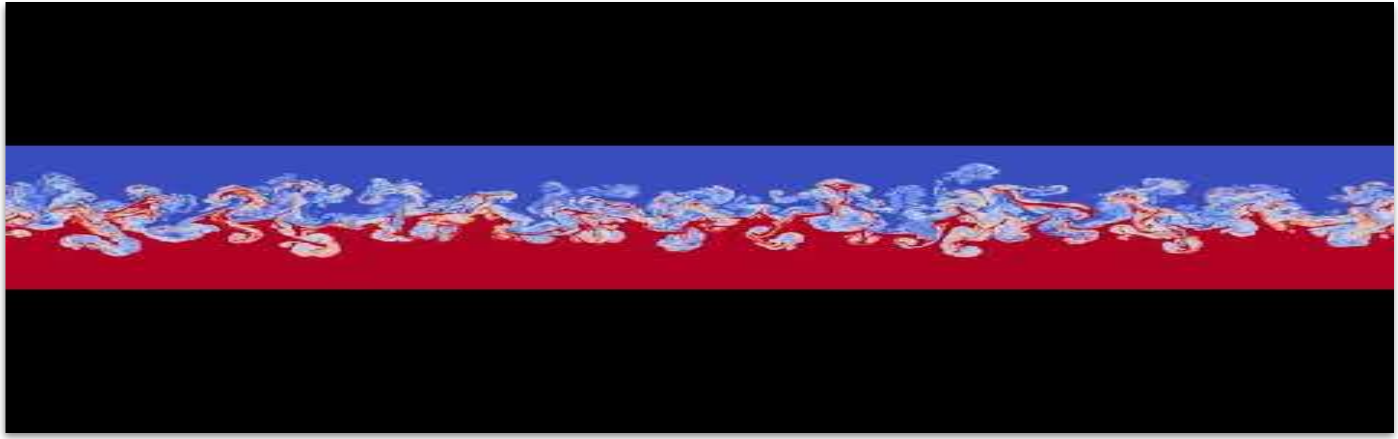
dyablo-Whole Sun: Hydrodynamics tests











Convective hydrodynamics benchmark

Setup

- Inspired from Hurlburt 1984, **Cattaneo et al 1991**, Brummell et al. 1996 and 2002

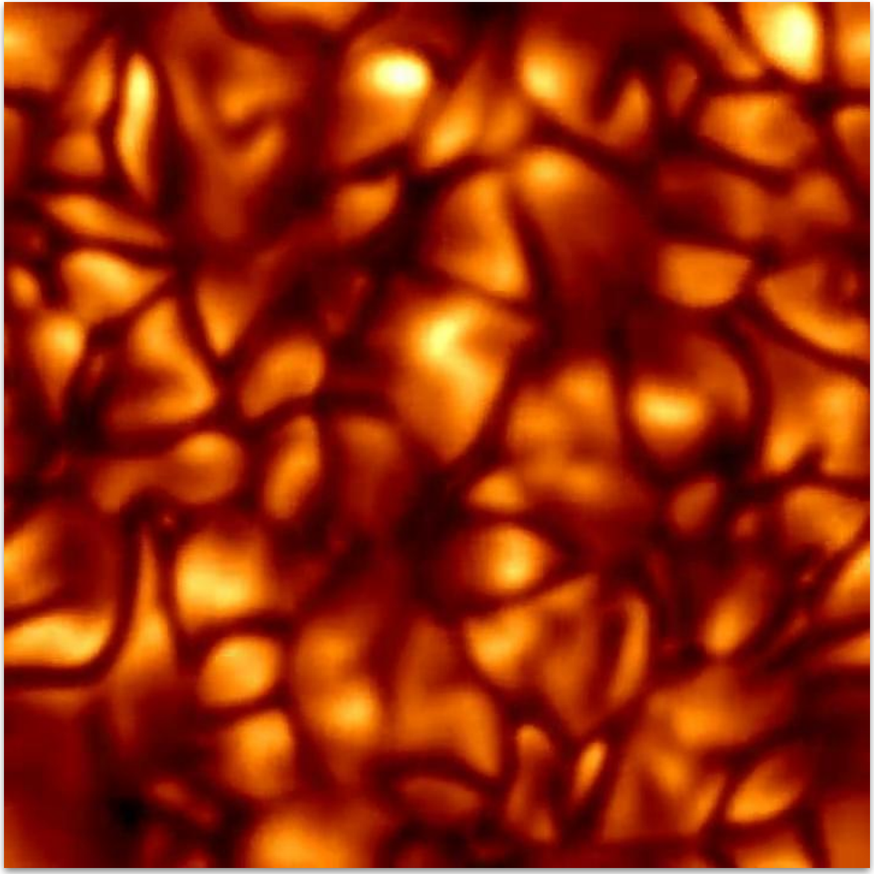
TURBULENT COMPRESSIBLE CONVECTION

FAUSTO CATTANEO, NICHOLAS H. BRUMMELL, AND JURI TOOMRE

Joint Institute for Laboratory Astrophysics and Department of Astrophysics, Planetary, and Atmospheric Sciences, University of Colorado, Boulder, CO 80309-0440

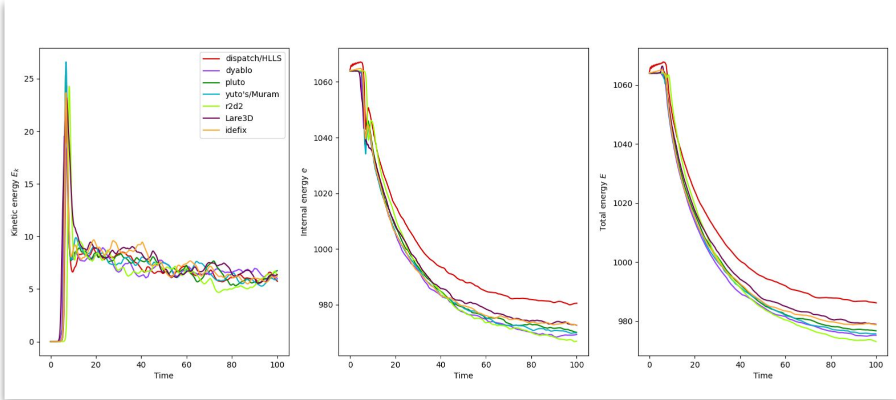
- **Ingredients**: Compressible hydrodynamics, viscosity, gravity and thermal conduction
- **Domain:** Convective near-surface slab. Highly stratified spanning multiple density scale-heights.
  - Horizontal dimension spans 4 times the vertical dimension
  - Fixed grid resolution: 256^2x64
  - ICs: Polytropic model, hydrostatic equilibrium, random perturbation on pressure
  - Vertical BCs: FT, stress-free impenetrable walls
- Benchmark inputs:
  - Stratification θ
  - Prandtl number σ
- 9 codes involved : dedalus, dispatch, dyablo, hps, idefix, lare3d, muram, pluto, r2d2

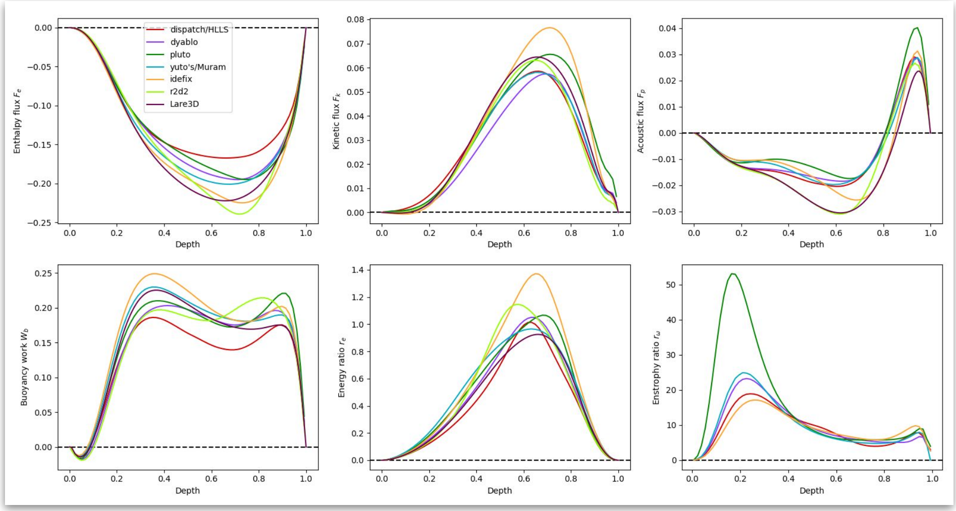
Convective hydrodynamics benchmark



Ran on 16 AMD-Epyc CPUs (Souleu)

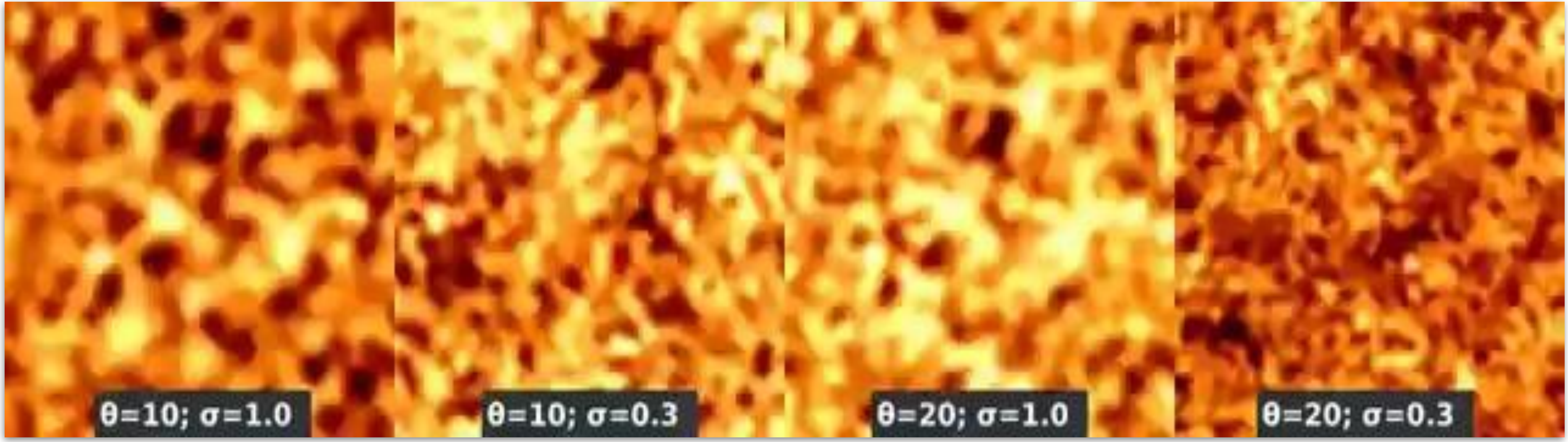
Convection benchmark





Increasing Ra

Horizontal cuts at z=0.1



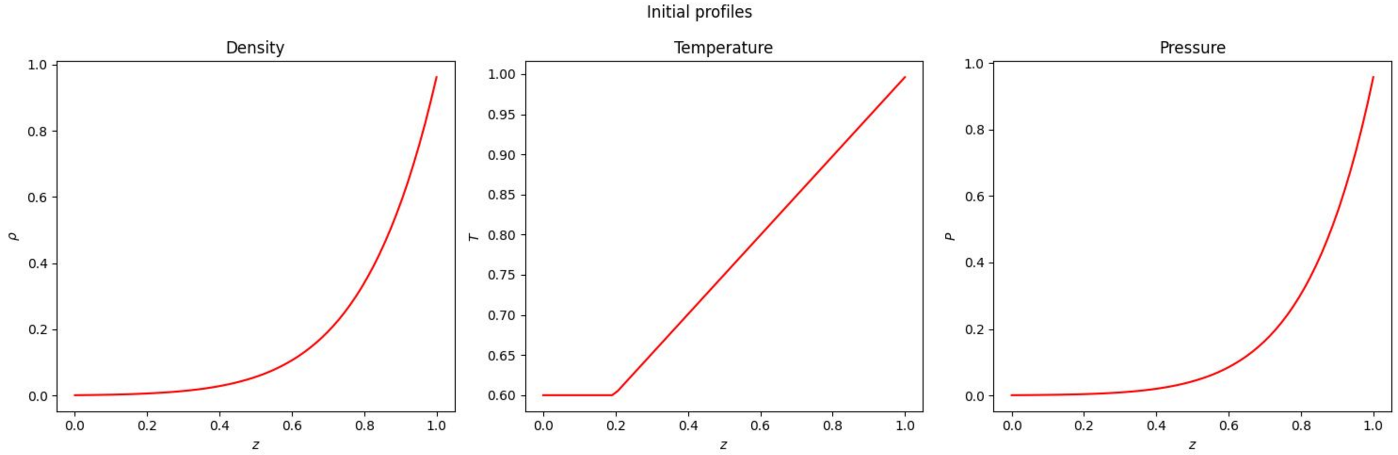
Ra = 4.9×10^4 | Ra = 1.6×10^5 | Ra = 3.6×10^5 | Ra = 1.6×10^6

Ran on 16 AMD-Epyc CPUs (Souleu)

Surface cooling driven convection benchmark

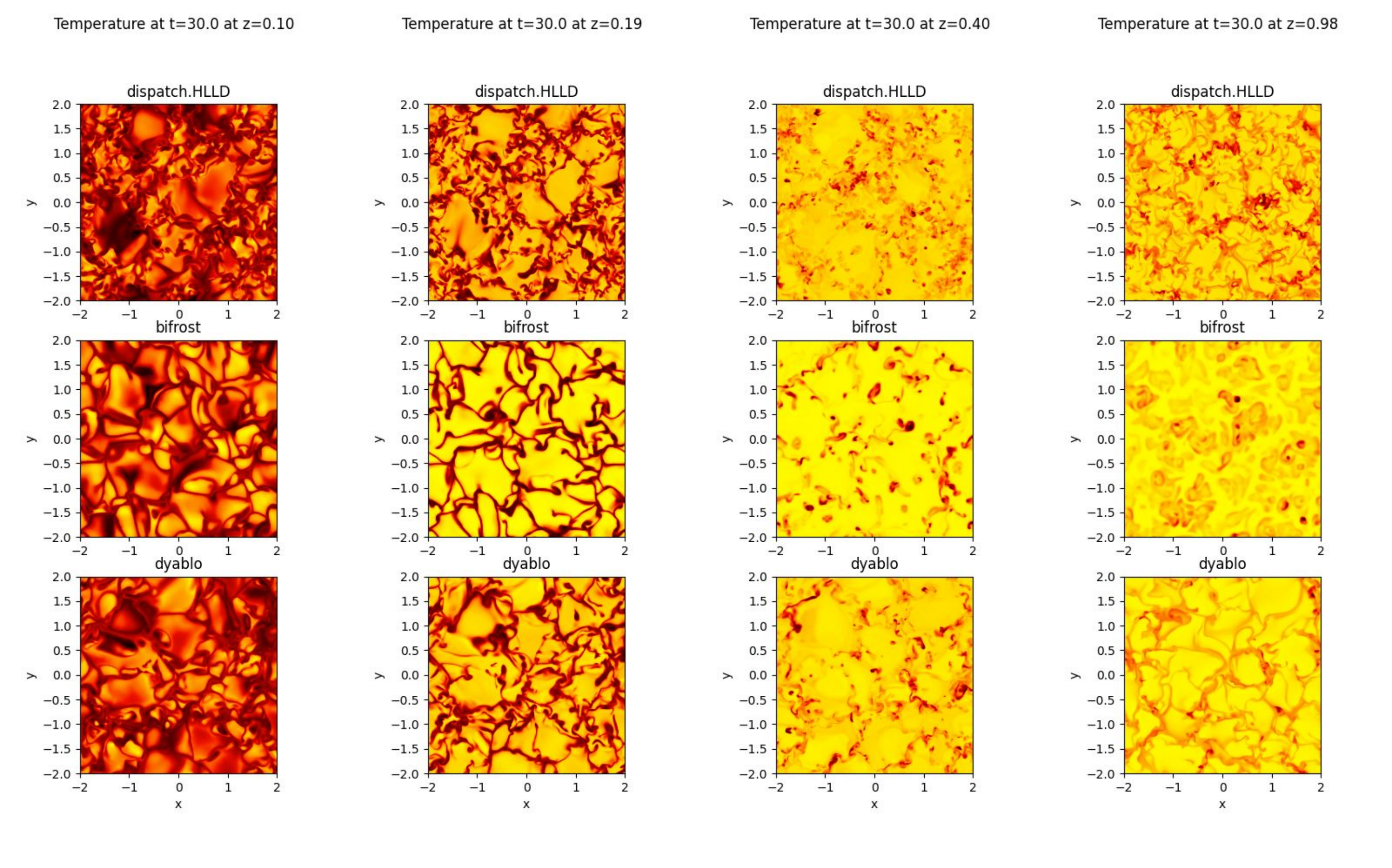
Setup

- Derived by Åke Nordlund in the context of Whole-Sun. Coordinated by Mikolaj Szydlarski
- **Ingredients**: Compressible hydrodynamics + Newtonian cooling
- **ICs** :
  - Polytropic model from the base of the convection zone to the cooling layer,
  - Constant temperature above
  - Deterministic perturbation to trigger instability
- Participating codes : bifrost, dispatch, dyablo, (CO)-Mancha



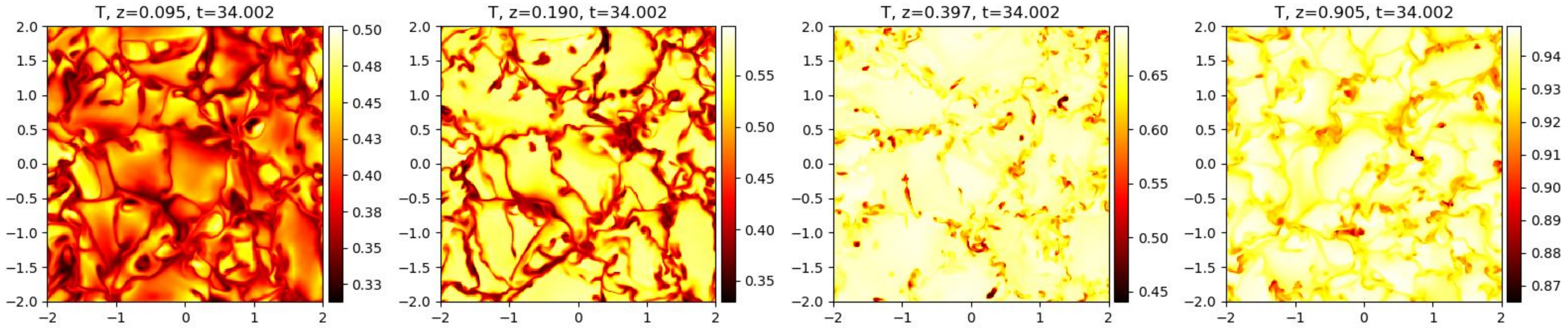
Surface cooling driven convection benchmark

Runs



Surface cooling driven convection benchmark

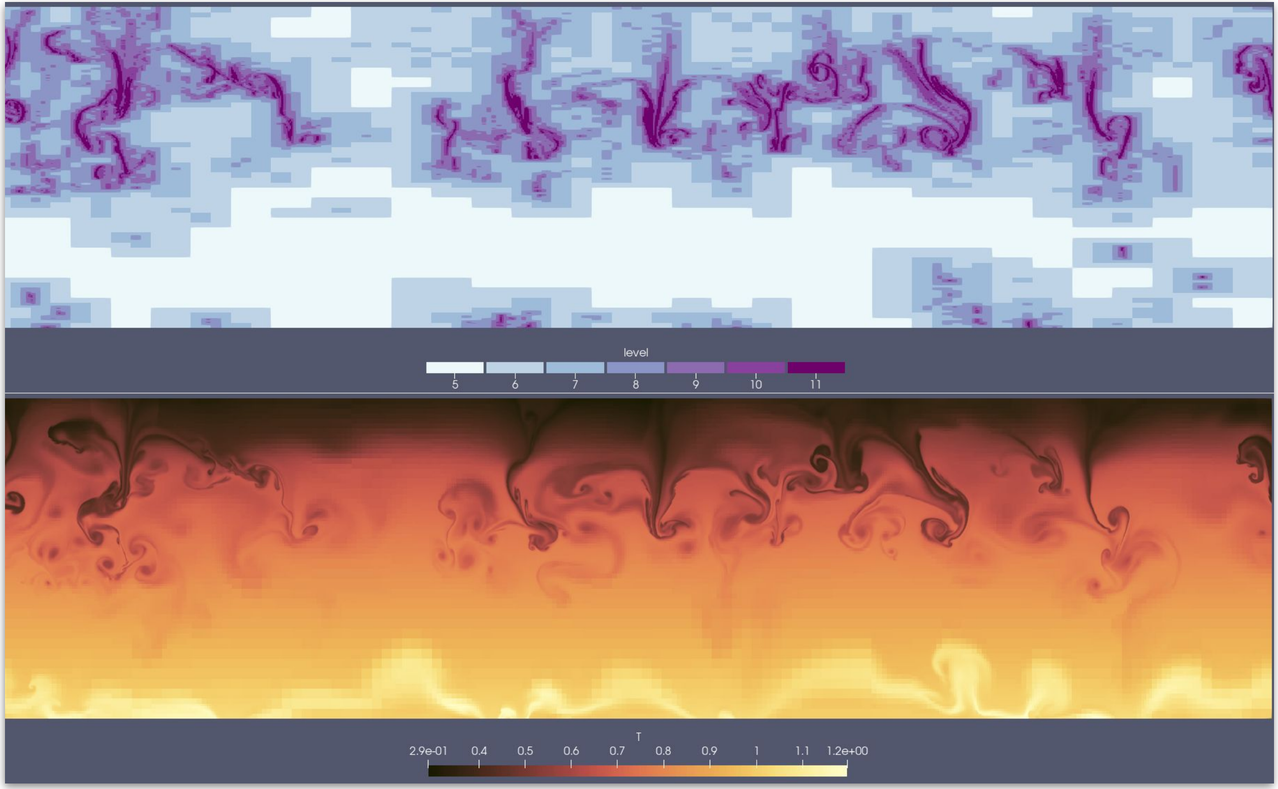
Runs



Ran on 4 AMD-Epyc CPUs (Souleu)

Surface cooling driven convection benchmark

2d AMR Runs



Ran on a laptop GPU

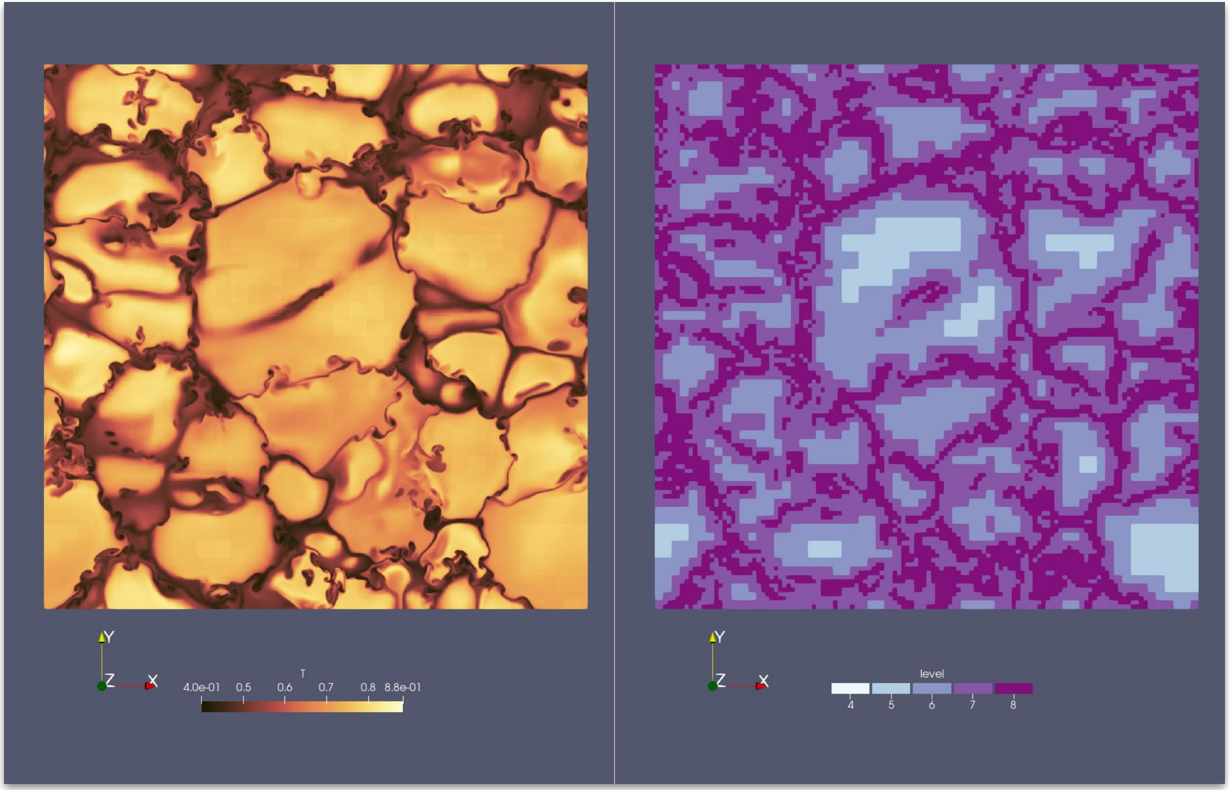
Base resolution:
128x32

Max resolution:
8192x2048

Blocks:
4x1

Surface cooling driven convection benchmark

3d AMR Runs [base level of fixed run is 6]



Ran on 8 Nvidia v100 (Irene)

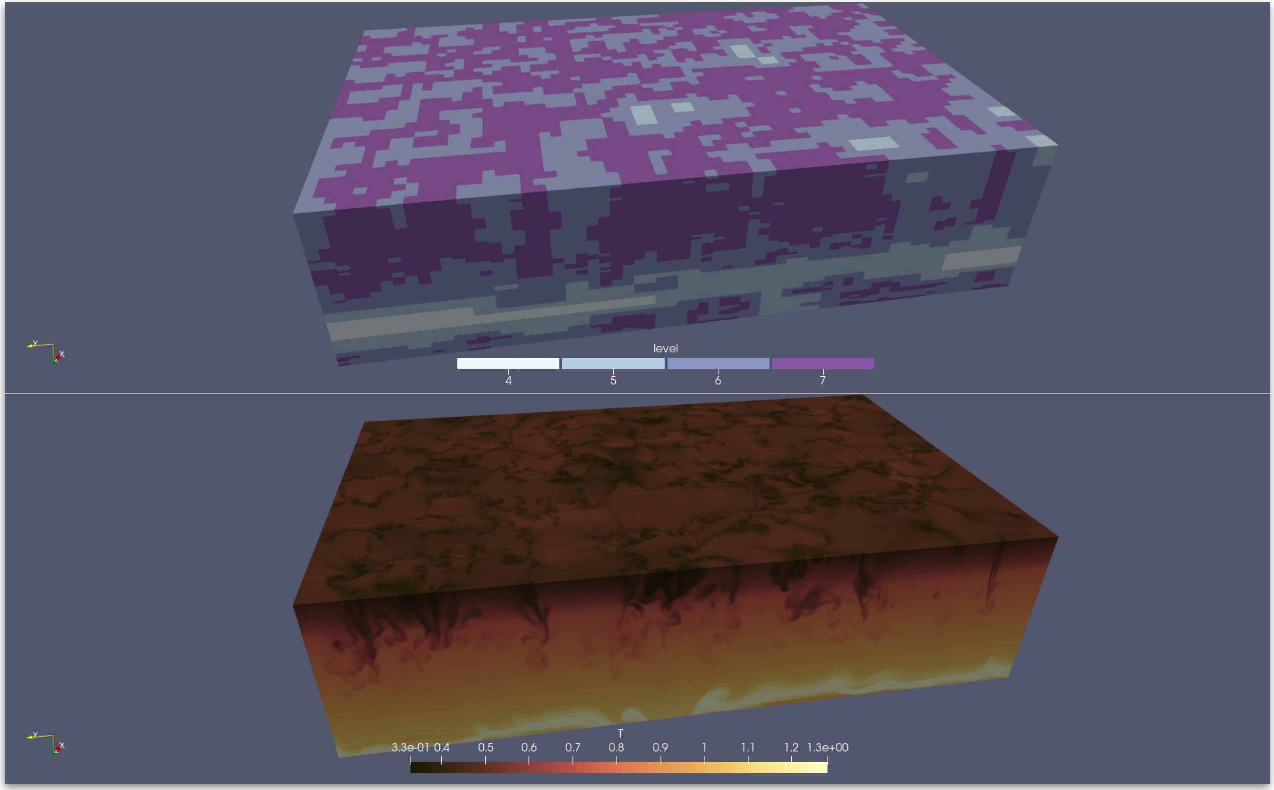
Base resolution:
64x64x16

Max resolution:
1024x1024x256

Blocks:
4x4x1

Surface cooling driven convection benchmark

AMR Runs [base level of fixed run is 6]



Ran on 8 Nvidia v100 (Irene)

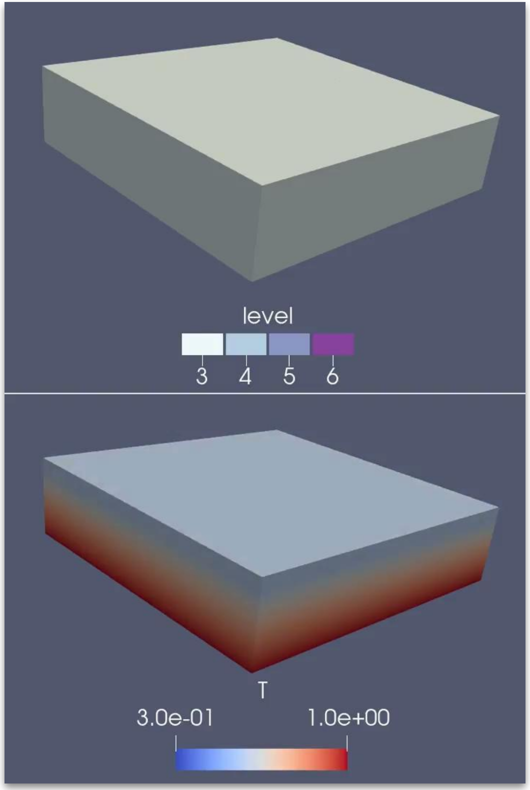
Base resolution:
64x64x16

Max resolution:
512x512x128

Blocks:
4x4x1

Surface cooling driven convection benchmark

AMR Runs [base level of fixed run is 4]



Ran on 8 Nvidia v100 (Irene)

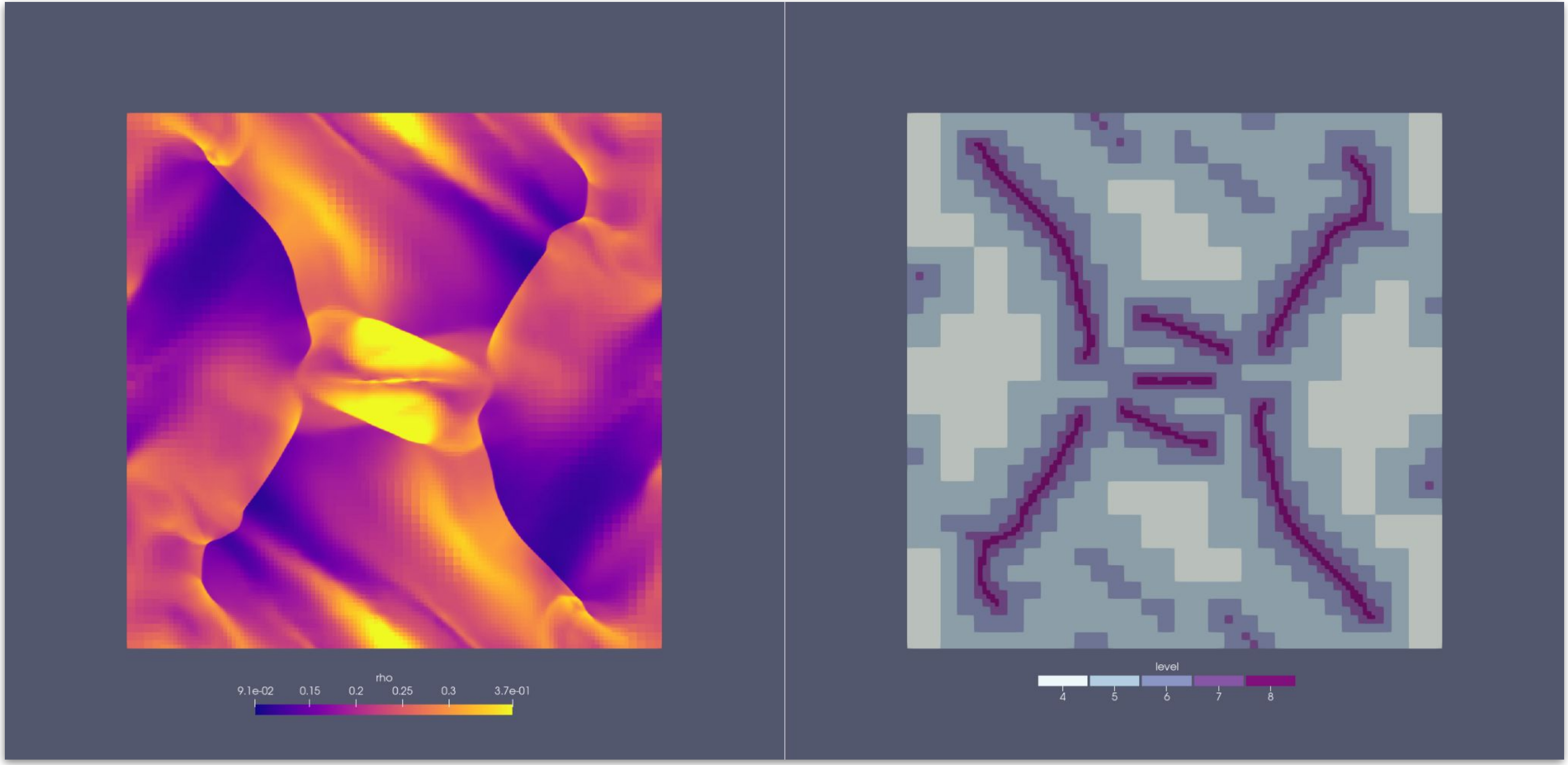
Base resolution:
128x128x32

Max resolution:
1024x1024x256

Blocks:
16x16x4

MHD

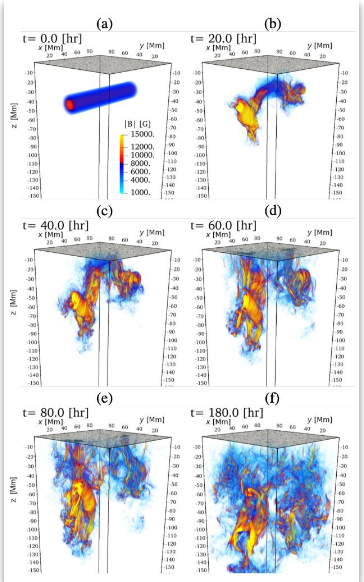
Orszag-Tang with AMR

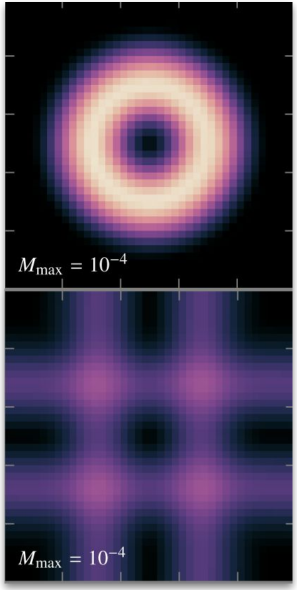


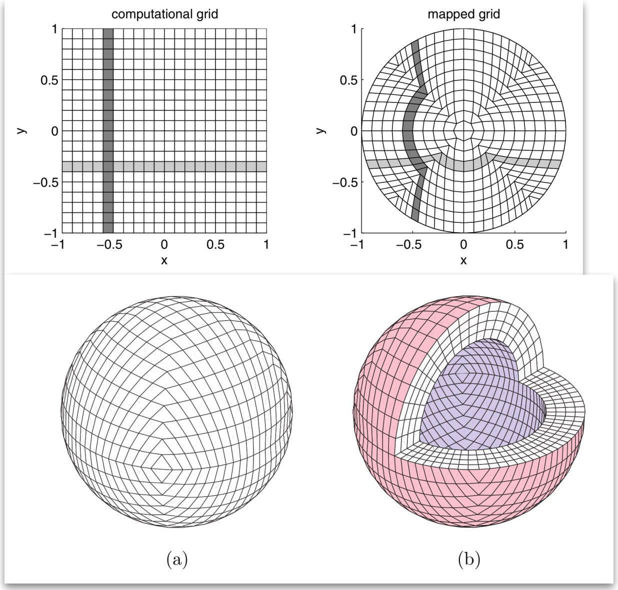
MHD + Well-balanced + All-Mach scheme (Tremblin et al, in prep)

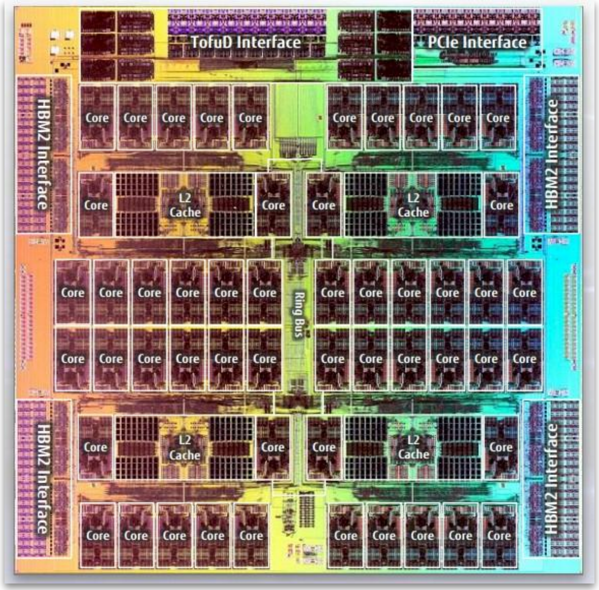
dyablo-Whole Sun:

What’s next ?









[Hotta et al 2020](#) [Miczek et at 2015](#) [Calhoun et. al 2008](#)

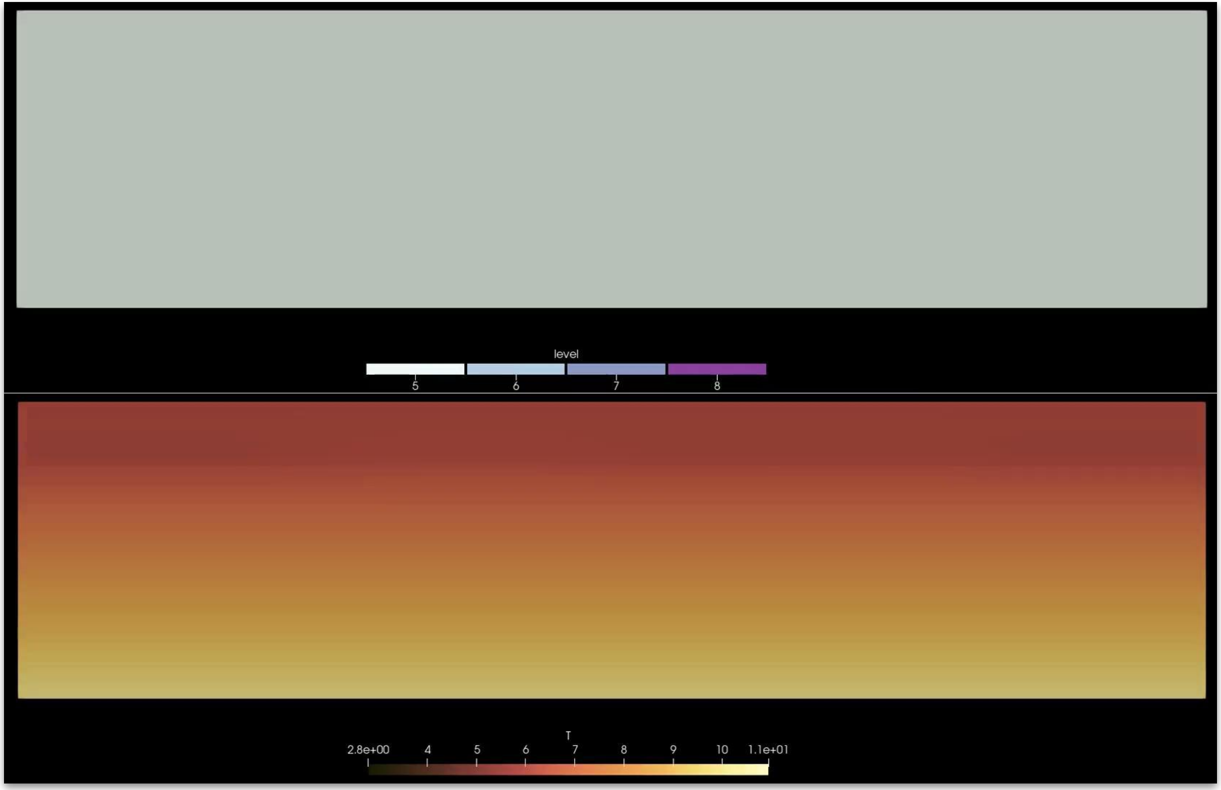
(+ Tons of debugging/improvements/testing)

Thank you
for your attention

Questions ?

Surface cooling driven convection benchmark

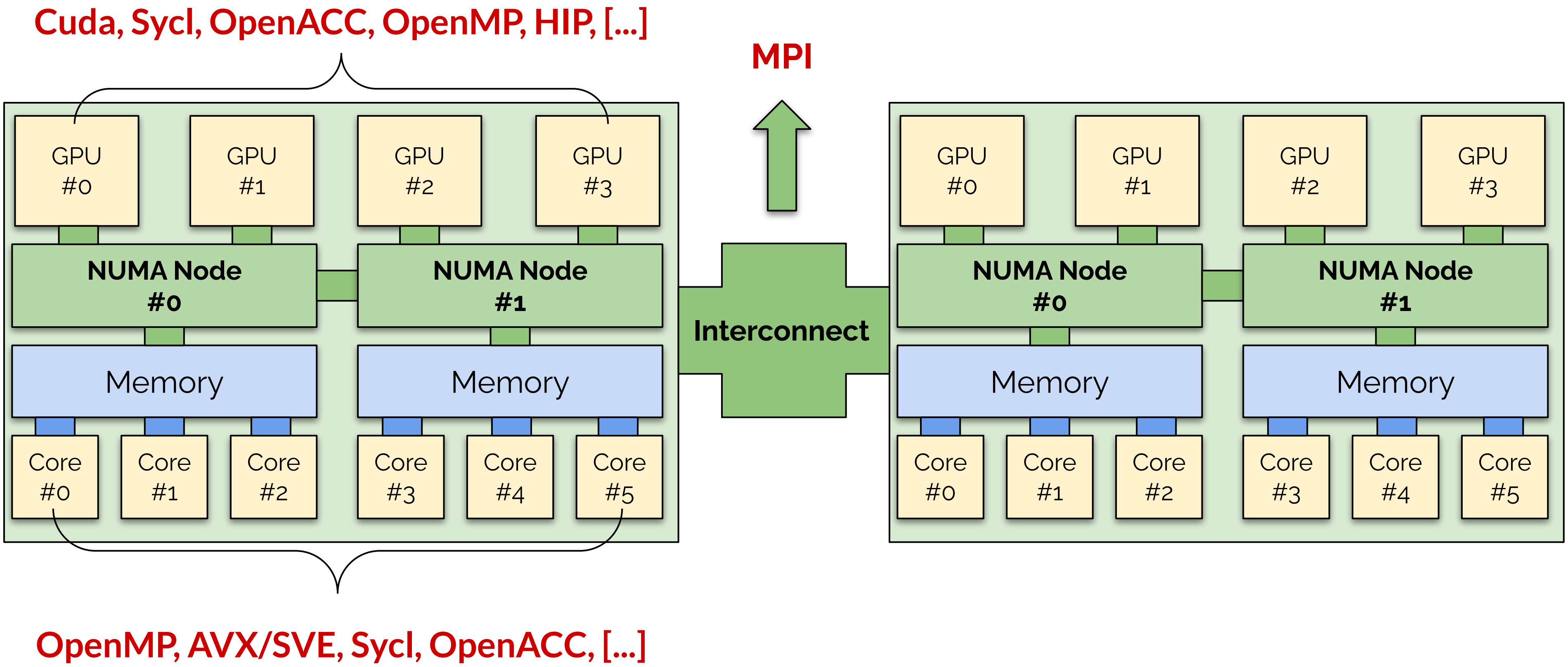
AMR Runs (2d)



Base resolution:
128x32

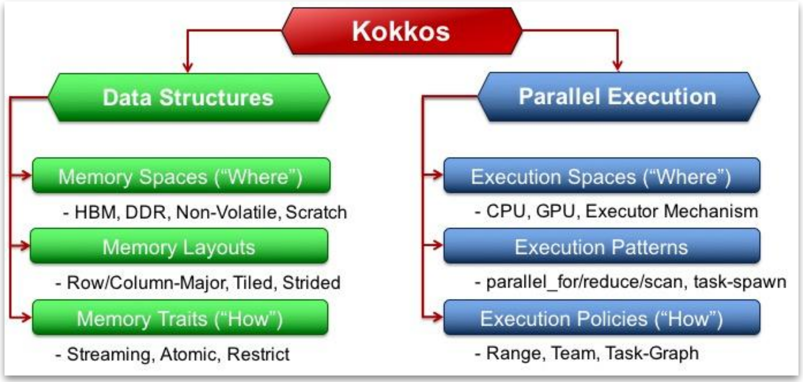
Max resolution:
1024x256

The scope of performance portability



Kokkos: performance portability in C++

A solution to heterogeneous systems

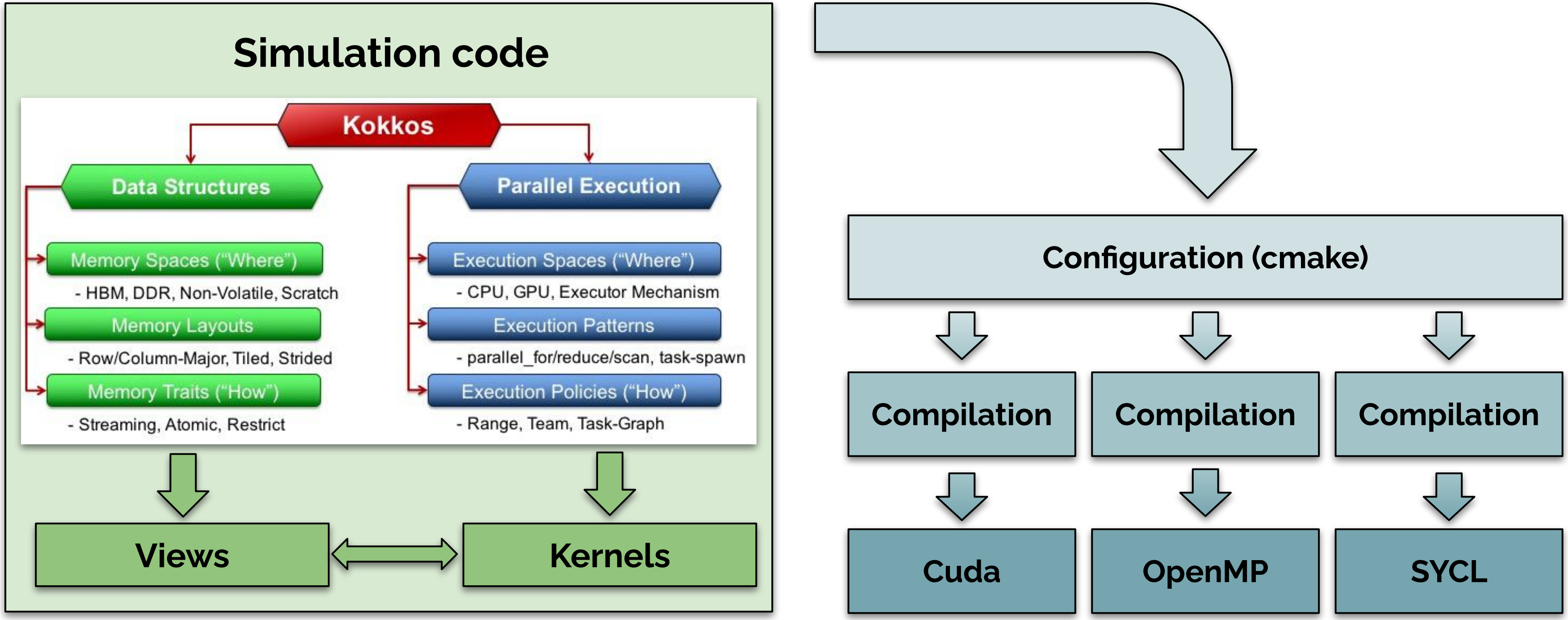


- Open-source modern C++ metaprogramming library
- Developer picks the memory structure, the type of algorithm and provides computation kernels
- Kokkos provides backends to automatically adapt the code to target architectures with minimum overhead

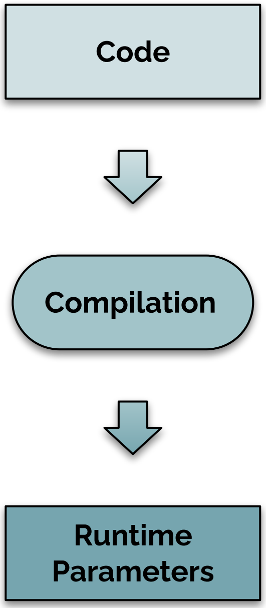
https://github.com/kokkos/kokkos

Carter Edwards, H., Trott, C., Sunderland, D., "Kokkos: Enabling manycore performance portability through polymorphic access patterns", Journal of Parallel and Distributed Computing, 2014

Using the Kokkos ecosystem



Plugins/Factory example




```cpp
class ParabolicUpdate_implicit {
  ParabolicUpdate_implicit() {
    /** Constructor :
     * initialising object,
     * reading required parameters, etc.
     **/
  }

  void update(
    /** Parameters required for parabolic update **/)
  {
    // Code of the update
  }
};
FACTORY_REGISTER( dyablo::muscl_block::ParabolicUpdateFactory,
                  dyablo::muscl_block::ParabolicUpdate_explicit,
                  "ParabolicUpdate_implicit")
```

```cpp
class ParabolicUpdate_explicit {
  ParabolicUpdate_explicit() {
    /** Constructor :
     * initialising object,
     * reading required parameters, etc.
     **/
  }

  void update(
    /** Parameters required for parabolic update **/)
  {
    // Code of the update
  }
};
FACTORY_REGISTER( dyablo::muscl_block::ParabolicUpdateFactory,
                  dyablo::muscl_block::ParabolicUpdate_explicit,
                  "ParabolicUpdate_explicit")
```

```ini
[parabolic]
thermal_conduction=ParabolicUpdate_explicit
viscosity=ParabolicUpdate_explicit
uniform_kappa=true
viscosity_type=dynamic
uniform_viscosity_coefficient=true
```

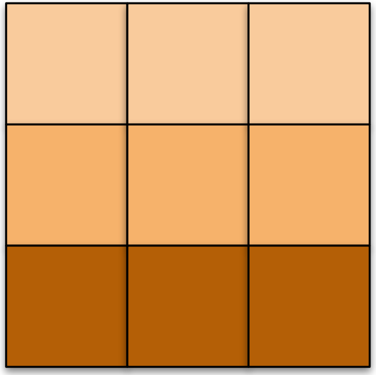
```ini
[parabolic]
thermal_conduction=ParabolicUpdate_implicit
viscosity=ParabolicUpdate_explicit
uniform_kappa=true
viscosity_type=dynamic
uniform_viscosity_coefficient=true
```

AMR-Cycle

AMR cycle step	PABLO backend	Hashmap backend
Cell marking	On device + transfer	On device*
Mesh adaptation	On host	On device*
Mesh remapping	On device	On device
Load balancing	On host	On device

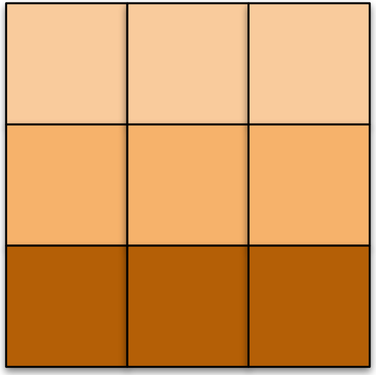
* CPU <-> GPU transfers due to backward compatibility

Hydrostatic equilibrium refinement disaster

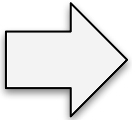


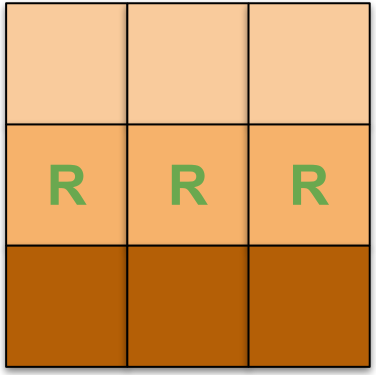
In hydrostatic equilibrium

Hydrostatic equilibrium refinement disaster



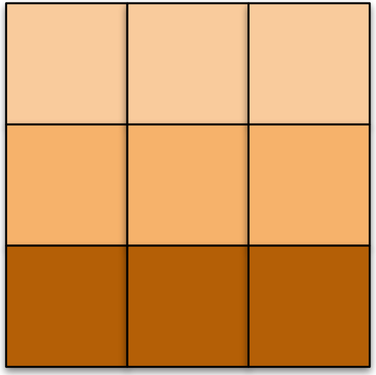
In hydrostatic equilibrium



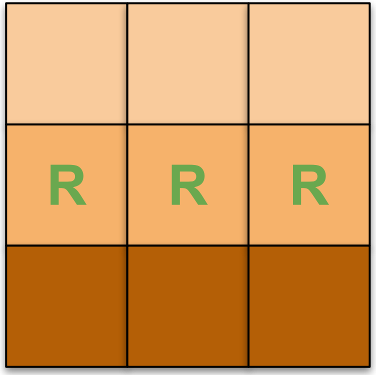


Marking for refinement

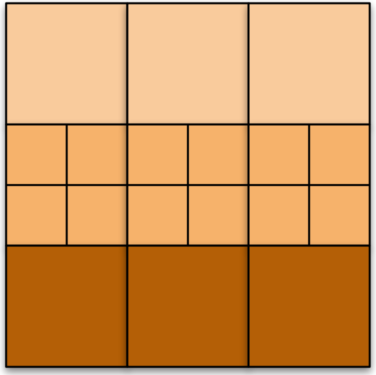
Hydrostatic equilibrium refinement disaster



In hydrostatic equilibrium



Marking for refinement



Hydrostatic equilibrium refinement disaster