IDEFIX DAYS

(c) Bruno Maillard

PNST
Programme National Soleil-Terre

PNPS
Programme National de Physique Stellaire

MHDiscs
Revealing the dynamics of planet-forming discs

IPAG
Institut de Planétologie et d'Astrophysique de Grenoble

erc
European Research Council
Established by the European Commission

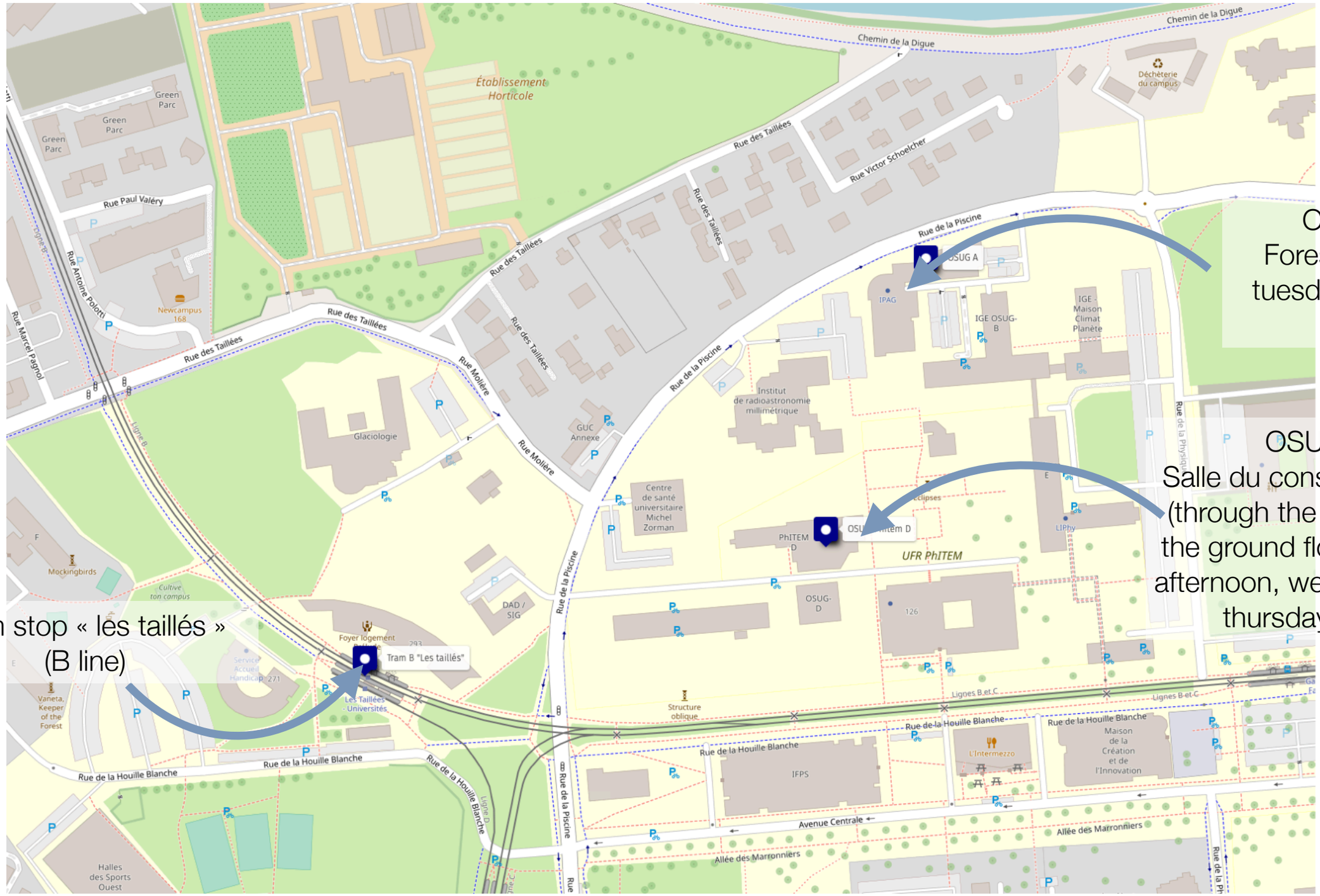
PNP

PNHE
Programme national hautes énergies

The plan for the first Idefix days

- Mornings: Scientific presentation (20” + 10” questions)
- Afternoon: tutorials
  - First day: basics, how to setup your problem, how to configure and compile on several architecture
  - Second day: visualisation with several tools (yt, nonos, paraview), debugging and profiling
  - Third day: how to contribute to Idefix
- Wednesday evening: conference dinner.

Logistics



OSUG A
Forestini room: tuesday morning only

OSUG D
Salle du conseil, first floor (through the museum on the ground floor): tuesday afternoon, wednesday and thursday all day

Tram stop « les taillés » (B line)

IDEFIX: a performance portable code for multi-physics astrophysical fluid dynamics

« small, smart, fast & French! »

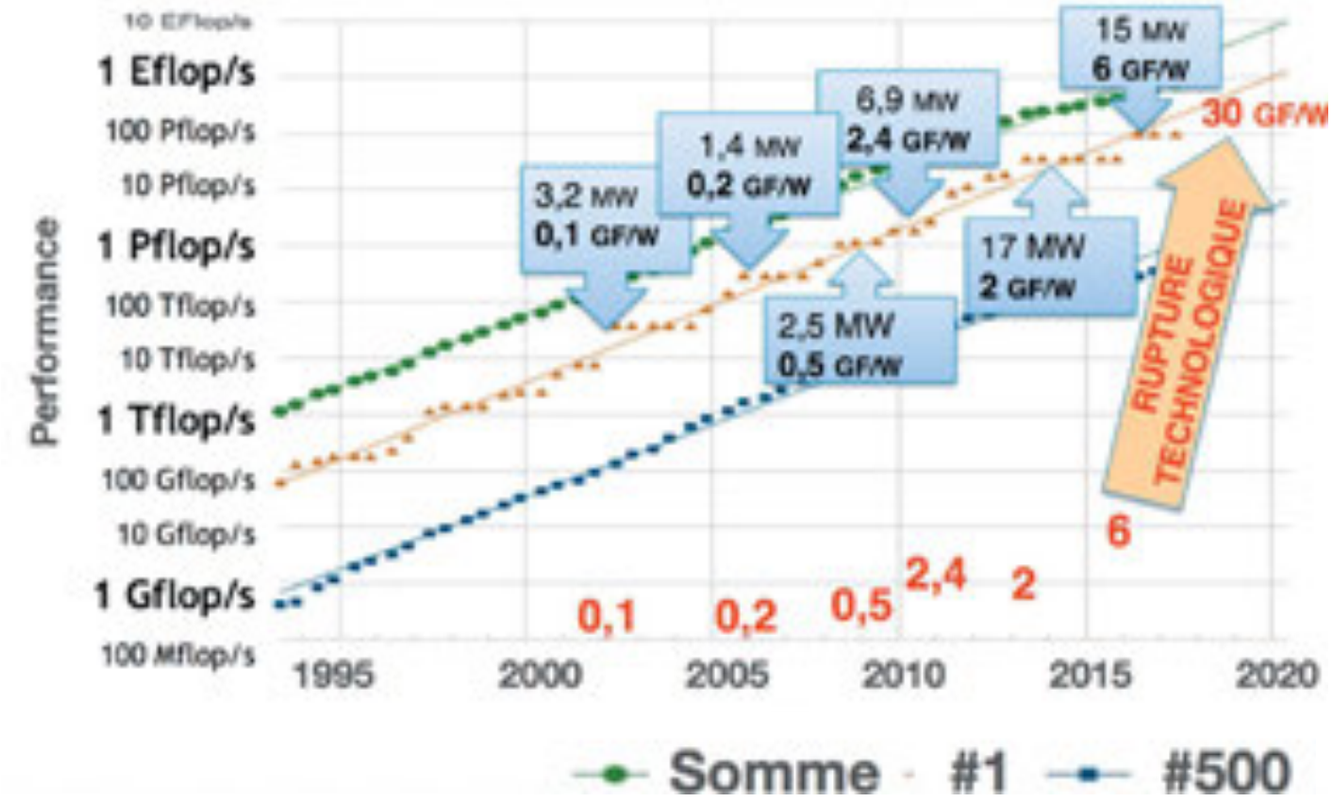
Geoffroy Lesur

Towards exascale

An electrical power issue

- The current average power consumption of a x86 processor is ~2W/Gflop/s
- an Exaflop machine would need about 500 MW of power=1 french nuclear reactor
- The « socially acceptable » power consumption is 30 MW

→ need for energy efficient architectures



=?

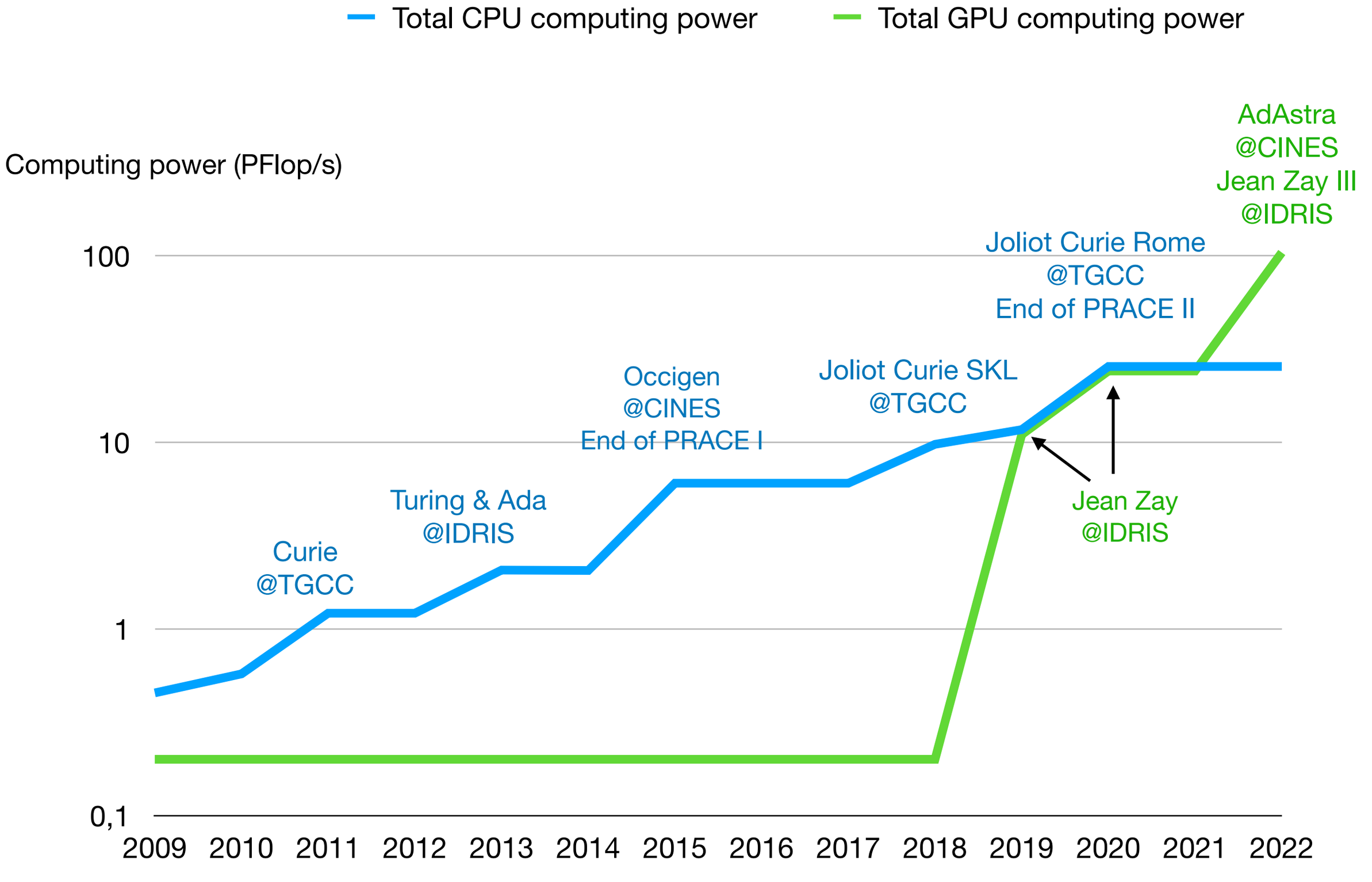
France: The Villani report

Des moyens de calcul pour la recherche

Les instituts de recherche en IA doivent pouvoir disposer d'outils de calcul qui leur permettent de rivaliser avec les moyens quasi-illimités des grands acteurs privés. C'est pourquoi notre mission propose la mise en place d'un supercalculateur conçu spécifiquement pour les applications d'IA, dédié aux chercheurs et à leurs partenaires économiques dans le cadre de projets communs.

As it turns out, IA applications usually runs best on graphical process units (GPUs)

Computing capacity in France (GENCI)



We need to be able to use GPUs and other hybrid architectures to run on future HPC facilities

Motivations for a performance portable code

- What is Performance portability?
  - (Re)write your code once, and run efficiently everywhere
  - High-level approach: hide hardware details to the physicist (if possible)

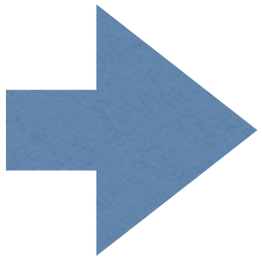
Multicore: ARM, AMD, Intel
ManyCore: NVidia, AMD GPUs

Our original wish list

Given that the project was initially financed in part by an ERC focusing on protoplanetary discs, we should be able to address:

- Supersonic flows (winds, jets)
- Magnetohydrodynamic (MHD) & Nonideal MHD terms (cold plasmas)
- Non-cartesian geometry (polar, at the very least)
- Some sort of mesh refinement

This already exists on the market: Athena, Pluto, AMRVac to name a few



Let’s port Pluto (we’ve been using it for 10 years…)

To port or to rewrite from scratch?

- Our original idea was to port progressively PLUTO by rewriting loops, but…

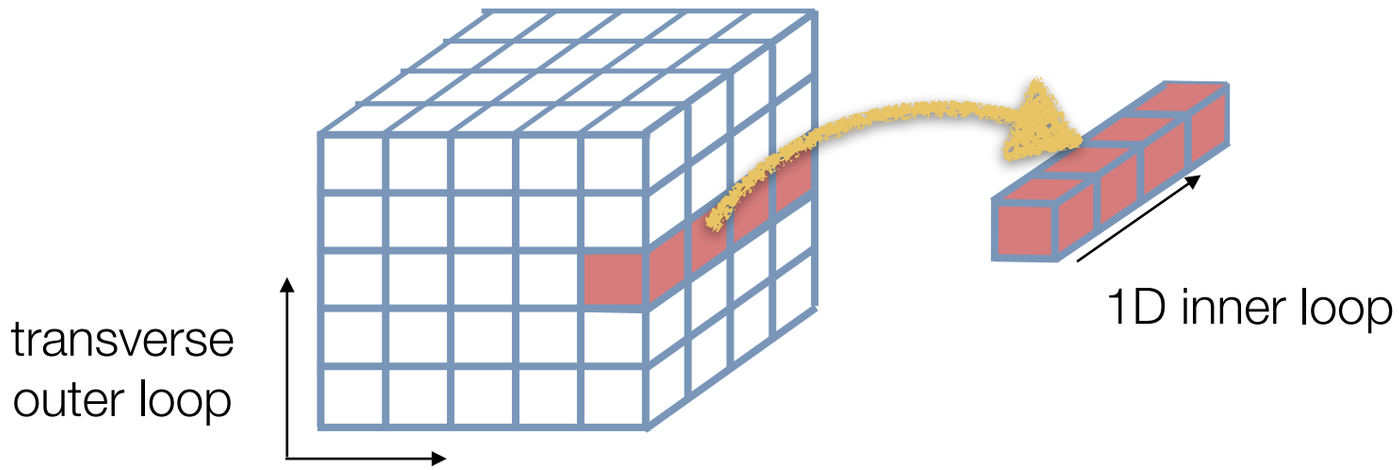
```
BOX_TRANSVERSE_LOOP(&sweepBox, k,j,i){
/* ---------------------------------------------------------
   2a. Copy data to 1D arrays
   --------------------------------------------------------- */
  [...........]
/* ---------------------------------------------------------
   2b. Compute L/R states
   --------------------------------------------------------- */
  CheckNaN (stateC->v, 0, ntot-1, "stateC->v");
  States  (&sweep, nbeg - 1, nend + 1, grid);

/* ---------------------------------------------------------
   2c. Solve Riemann problem
   --------------------------------------------------------- */
  d->fluidRiemannSolver (&sweep, nbeg-1, nend, Dts->cmax, grid);
  #if NSCL > 0
  AdvectFlux (&sweep, nbeg-1, nend, grid);
  #endif
/* ---------------------------------------------------------
   2d. Compute right hand side side
   --------------------------------------------------------- */
  RightHandSide (&sweep, Dts, nbeg, nend, dt, grid);
/* ---------------------------------------------------------
   2e. Update conservative solution array,

       U += dt*R
   --------------------------------------------------------- */
  for ((*ip) = nbeg; (*ip) <= nend; (*ip)++) {
    NVAR_LOOP(nv) Uc[k][j][i][nv] += sweep.rhs[*ip][nv];
  }

  [...........]
}
```

outer loop on the perpendicular directions



each function is working on a small 1D array
[very efficient on CPUs with small cache lines]

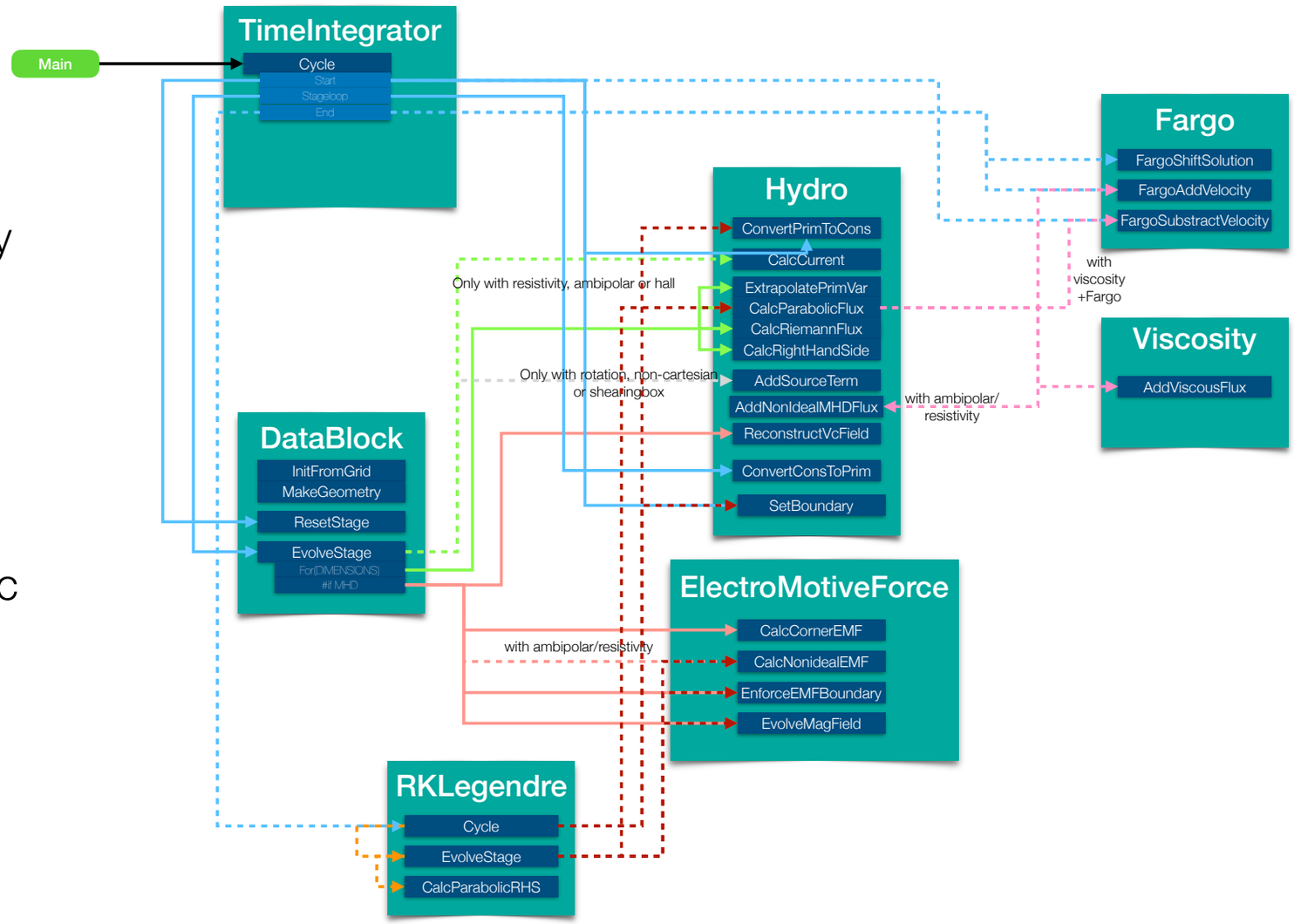
Difficult to make such a nested loop structure efficient on hybrid machines

Need to rewrite the core of PLUTO hydro solver
→ Rewrite everything from scratch, leveraging C++ features

Idefix

- Idefix is technically a new code (written from scratch), that implements a multistep, finite-volume Godunov algorithm in C++ 17, relying on Kokkos meta-programming objects.
  - 1st, 2nd or 3rd order in time (using Runge-Kutta TVD time integrators)
  - 1st, 2nd or 3rd order spatial reconstruction
  - Constrained transport for the MHD module
- Project started in March 2020 with Soufiane Baghdadi (IR IPAG)

- Inputs, outputs and data structures are very similar to Pluto: simplified setup portability
- Strong encapsulation: each physical module/algorithm is a C++ object with its own constructor/destructor
- Method paper early 2023, public release available



Code compilation

Portability & simplicity

We use `Cmake` to configure and compile Idefix

- Idefix features can be enabled (MPI parallelization, MHD, …)
- Kokkos backend automatically available to Idefix
- Simplifies portability to new architectures
- Possibility to use a user-friendly graphical interface: `ccmake`

on a laptop (MacOS)

```
lesurg@geoffsmacbook OrszagTang3D % cmake $IDEFIX_DIR -DIdefix_MHD=ON
-- The C compiler identification is AppleClang 12.0.5.12050022
-- The CXX compiler identification is AppleClang 12.0.5.12050022
[…]
-- Built-in Execution Spaces:
--     Device Parallel: NoTypeDefined
--     Host Parallel: NoTypeDefined
--       Host Serial: SERIAL
-- Kokkos Devices: SERIAL, Kokkos Backends: SERIAL
-- Idefix final configuration
--     MHD: ON
--     MPI: OFF
--     Version: v0.9.0-3-ga1ad803
--     Problem definitions: 'definitions.hpp'
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/lesurg/Documents/src/idefix/test/MHD/OrszagTang3D
lesurg@geoffsmacbook OrszagTang3D % make
Consolidate compiler generated dependencies of target kokkoscore
[  1%] Building CXX object build/kokkos/core/src/CMakeFiles/kokkoscore.dir/impl/Kokkos_CPUDiscovery.cpp.o
…..
```

on a cluster with Nvidia GPUs

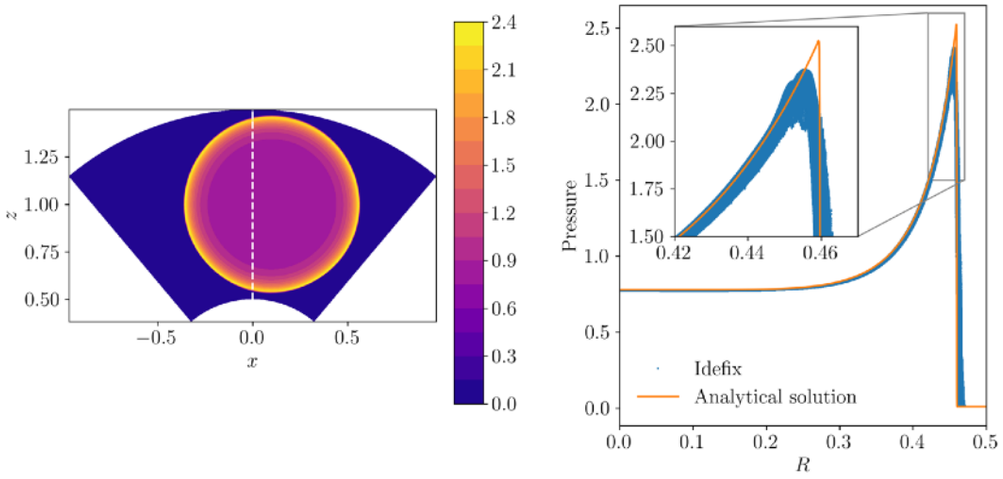
```
lesurg@f-dahu:$ cmake $IDEFIX_DIR -DIdefix_MHD=ON -DKokkos_ENABLE_CUDA=ON -DKokkos_ARCH_VOLTA70=ON
-- The C compiler identification is GNU 9.2.0
-- The CXX compiler identification is GNU 9.2.0
[….]
-- Built-in Execution Spaces:
--     Device Parallel: Kokkos::Cuda
--     Host Parallel: NoTypeDefined
--       Host Serial: SERIAL
--
-- Architectures:
--  VOLTA70
-- Found CUDAToolkit: /applis/dahu/cuda/cuda-10.1/include (found version "10.1.243")
-- Kokkos Devices: CUDA;SERIAL, Kokkos Backends: CUDA;SERIAL
-- Idefix final configuration
--     MHD: ON
--     MPI: OFF
--     Version: v0.9.0-4-gb669403
--     Problem definitions: 'definitions.hpp'
-- Configuring done
-- Generating done
-- Build files have been written to: /home/lesurg/src/idefix/test/MHD/OrszagTang3D
lesurg@f-dahu:~/src/idefix/test/MHD/OrszagTang3D$ make
Scanning dependencies of target kokkoscore
[  1%] Building CXX object build/kokkos/core/src/CMakeFiles/kokkoscore.dir/impl/Kokkos_CPUDiscovery.cpp.o
……
```

(some) Standard tests

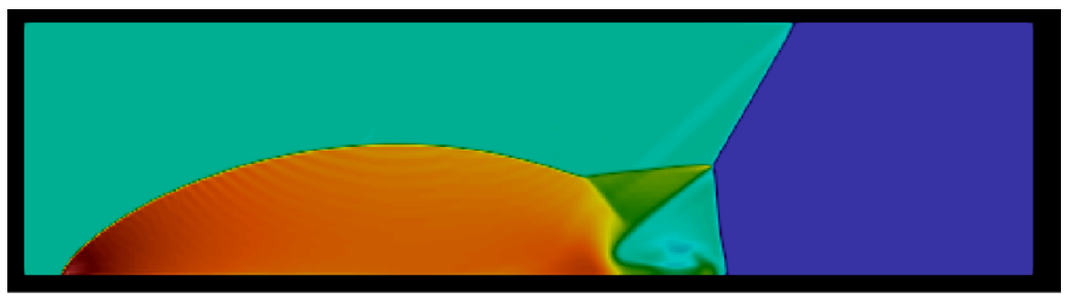
Hydrodynamics

Blast wave

(propagating through the axis singularity in spherical geometry)

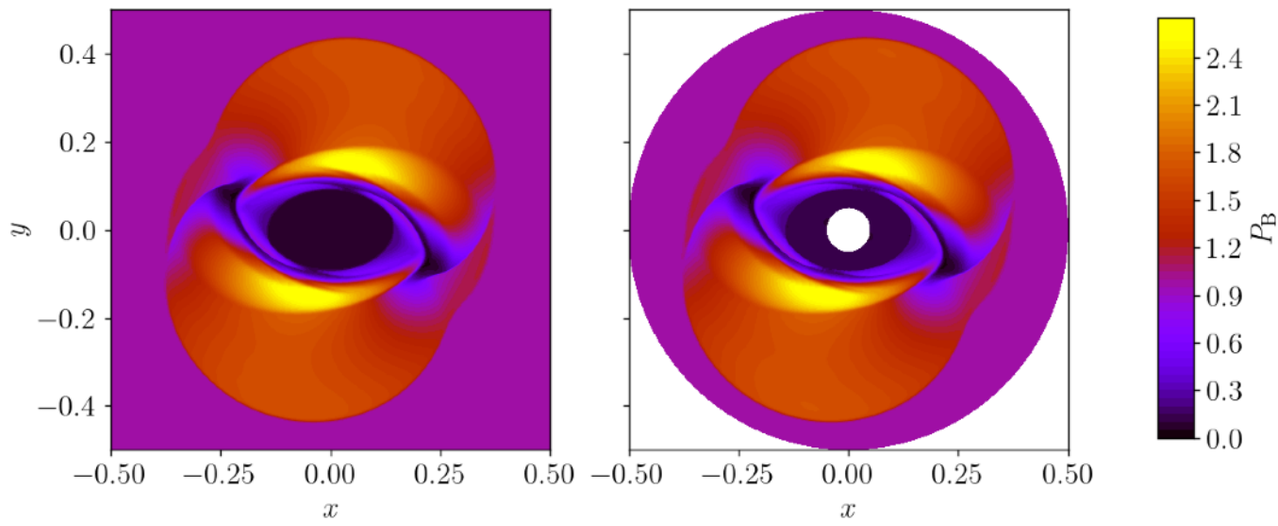


2D Mach reflection test

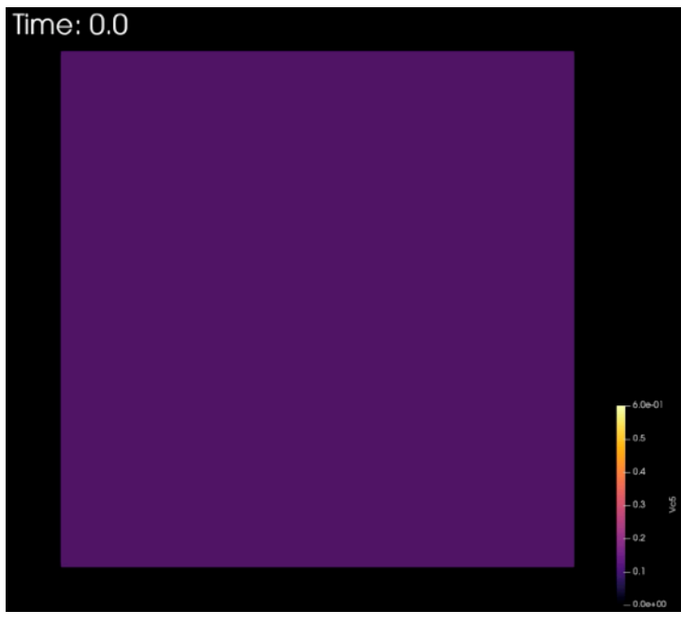


Magneto-Hydrodynamics

Magnetised rotor in various geometries



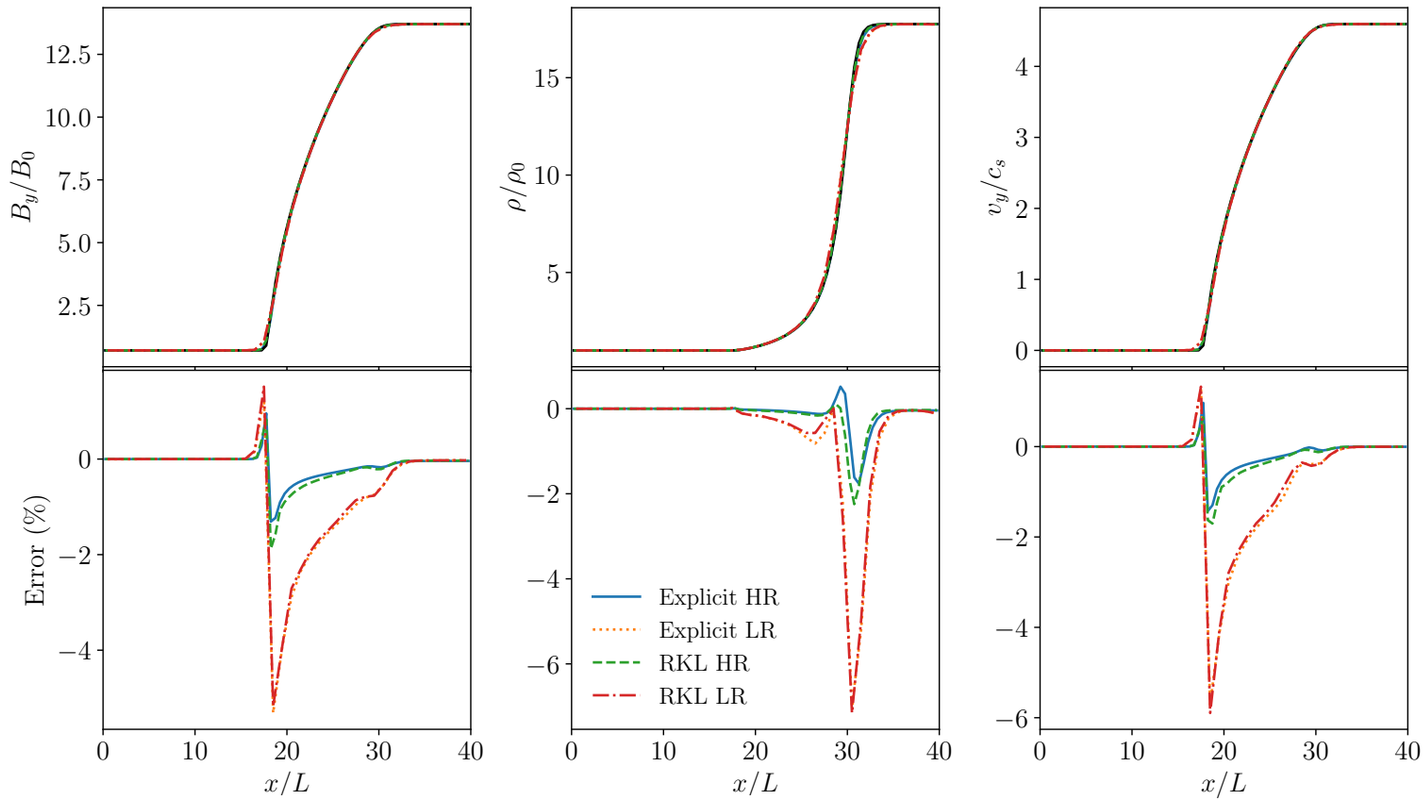
Orszag-Tang vortex



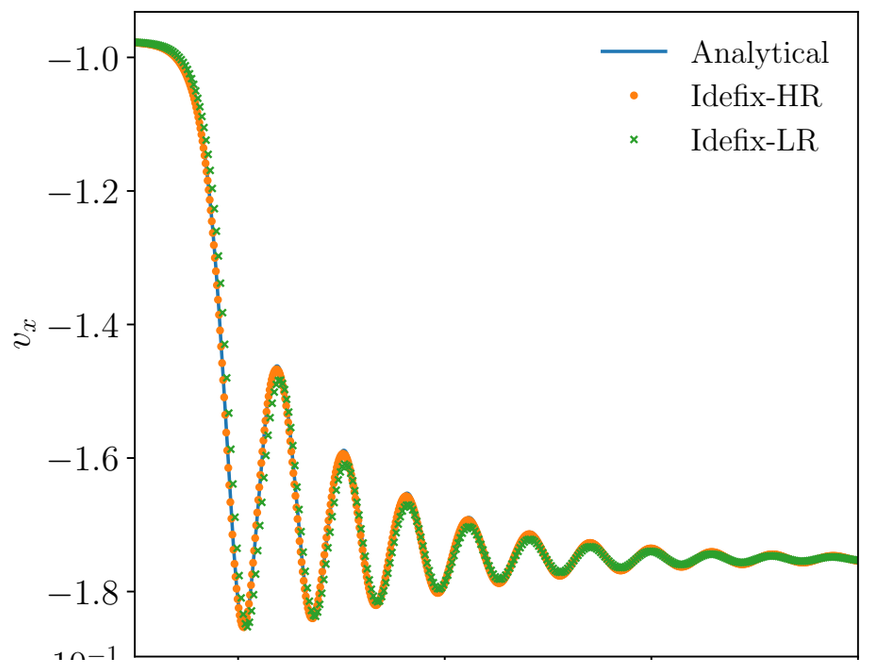
All tests are automatically compiled, run and checked every night (Gitlab CI/CD)

Non-ideal MHD tests

Ambipolar C-shock



Hall-dominated shock



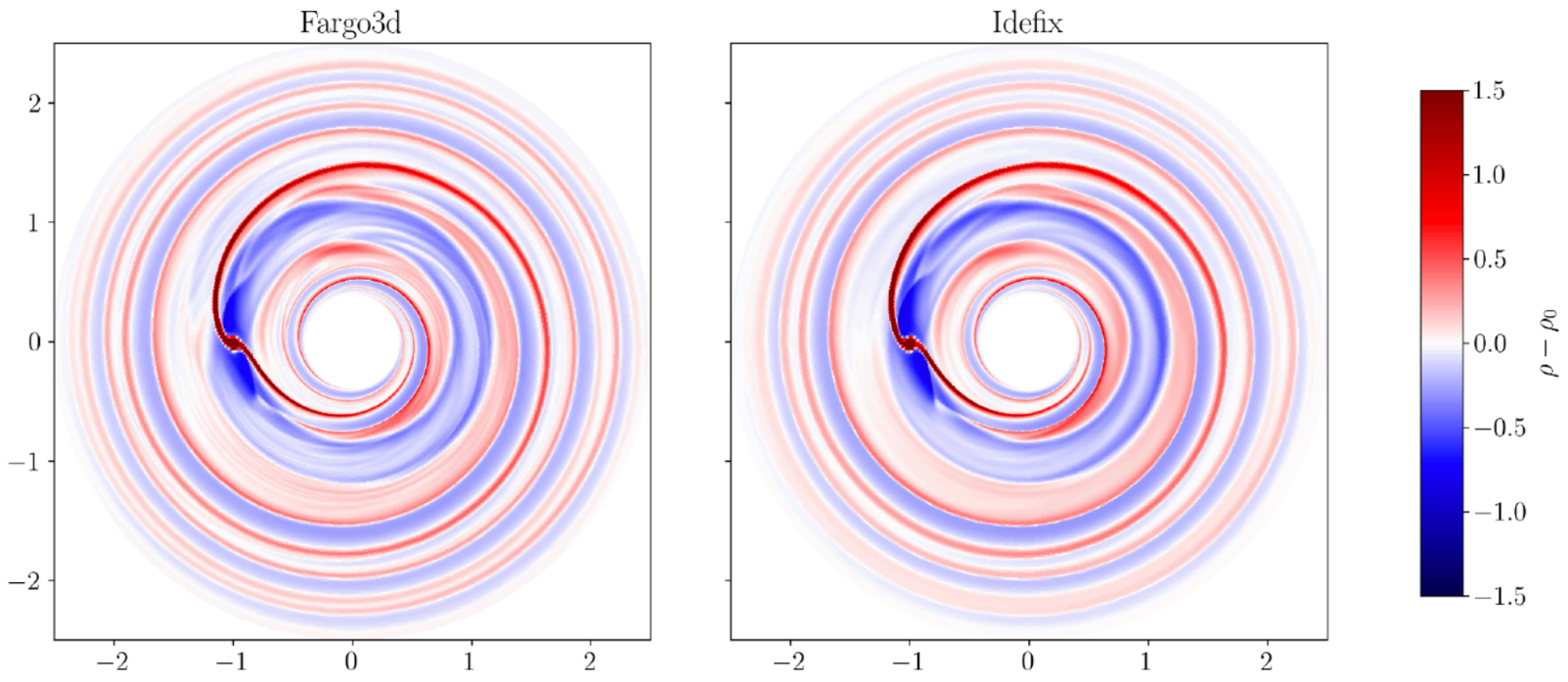
All tests are automatically compiled, run and checked every night (Gitlab CI/CD)

Planet-disc interaction

More tests

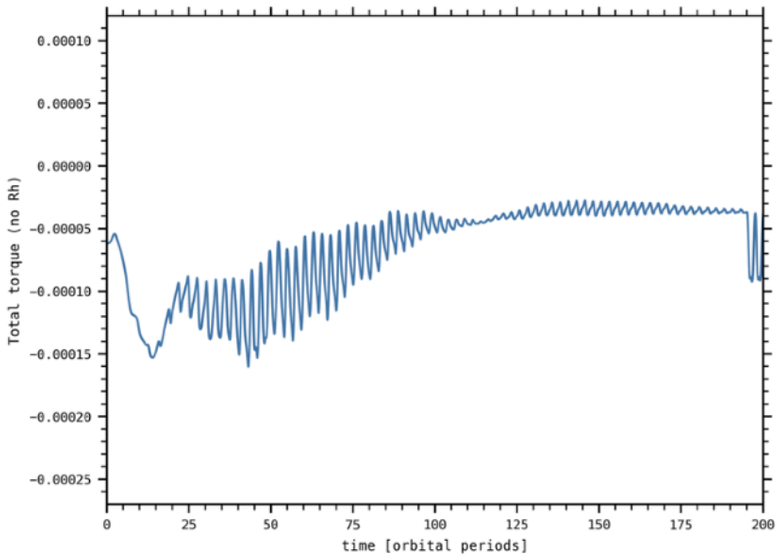
[Gaylor Wafflard-Fernandez]

- Fargo3D vs Idefix

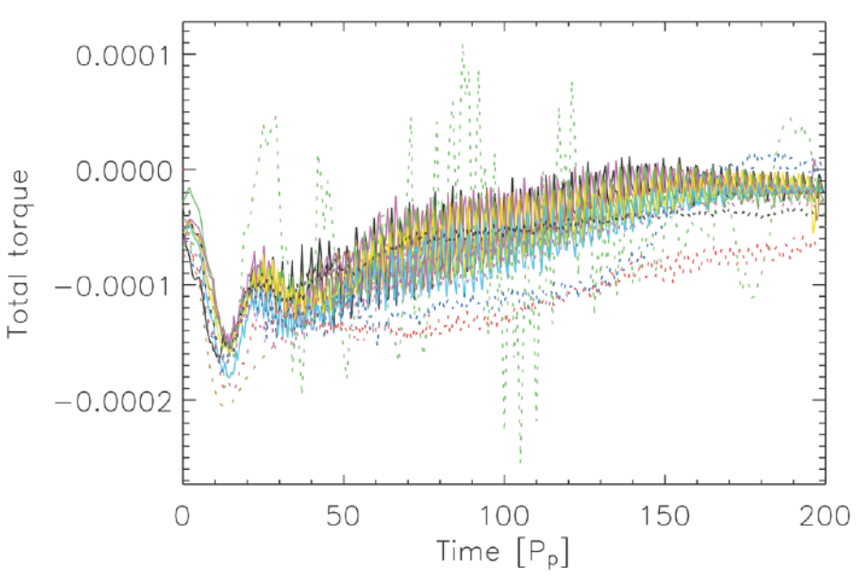


- Planet torque comparison [de Val-Borro 2006]

Idefix



de Val-Borro+ 2006



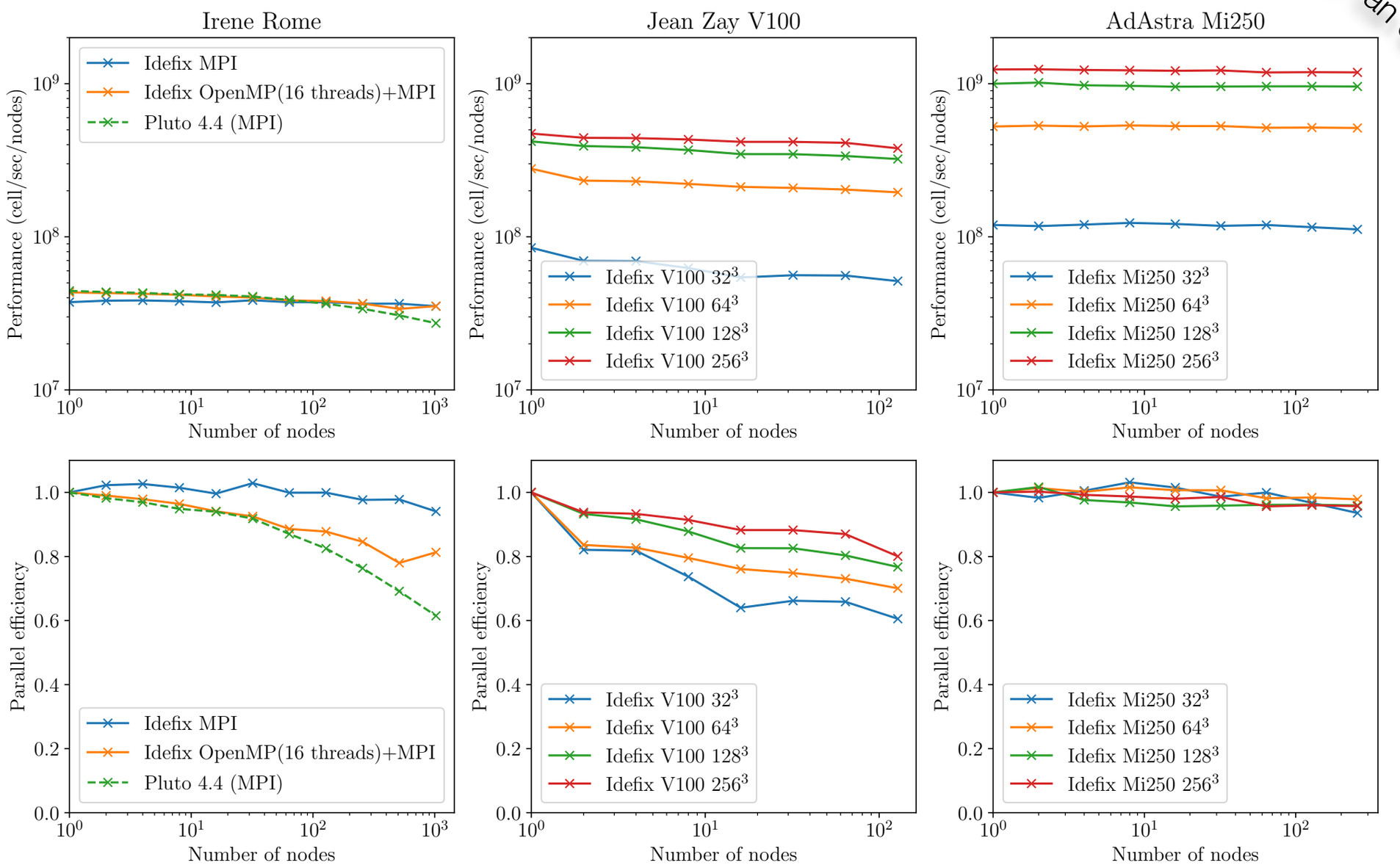
All tests are automatically compiled, run and checked every night (Gitlab CI/CD)

Parallel performances

Grand Challenge Adastra

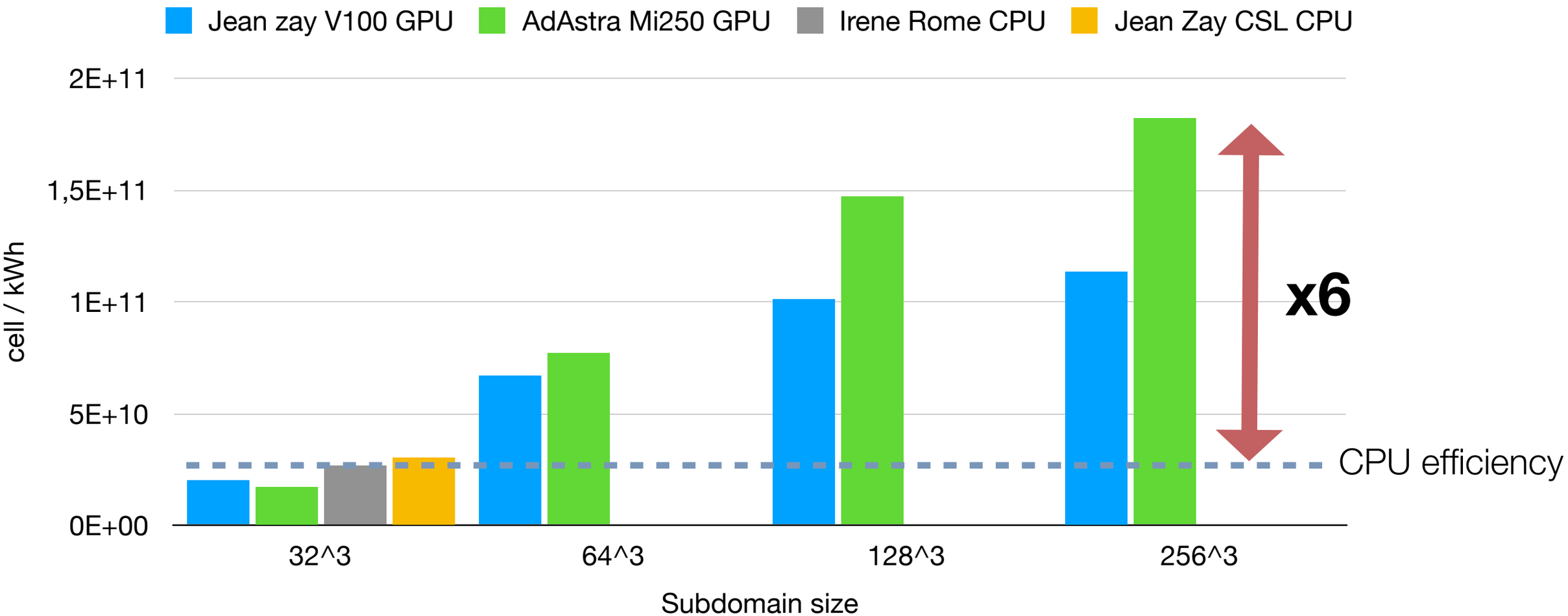
[M. Van den Bossche]

Test: 3D MHD Orszag-Tang vortex



- Similar performance to Pluto 4.3 on CPUs → performance portability is a success!
- 1 V100 node>10 Rome nodes, 1Mi250 node~30 Rome nodes
- Fairly good scaling (75% efficiency on 131072 CPUs, 80% efficiency on 512 V100s, 96% on 1024 Mi250)

Energy efficiency



Running the same problem on a GPU is up to 6 times more energy efficient

but

Strong dependence on the domain size!

Idefix status & roadmap

Feature	Status
HD & MHD	✓
Multiple Riemann solvers (Lax, HLL, HLLC/D, Roe)	✓
Geometry (cartesian, cylindrical, spherical, polar)	✓
Non-ideal MHD (Ohmic, Ambipolar, Hall)	✓
MPI, MPI+OpenMP, MPI+Cuda, MPI+HIP	✓
Checkpointing & restart	✓
Super time-stepping (RKL scheme: viscosity, ambipolar diffusion, Ohmic resistivity)	✓
Orbital advection (FARGO)	✓
Radiative transfer	✗
Dust (particle approach)	✓ (not yet public)
Dust (fluid approach)	✓
Self gravity	✓

IDEFIX

- Designed to work on hybrid machines using Kokkos but also keep performances on standard x86 CPUs
- Keep the (astro) physicist life as simple as possible with code that « looks like C »
- Tested (and works) on x86, Nvidia GPUs, AMD GPUs & ARM architectures (A64FX)
- Energetically efficient
- Your playground for the next 3 days!