SAMURAI

Loïc Gouarin
HPC@Maths team
CMAP / CNRS - Ecole polytechnique

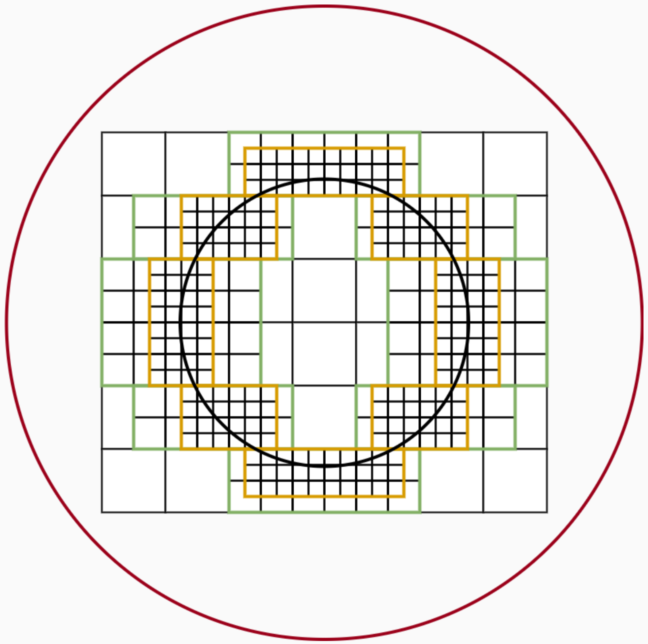
PATCHED-BASED REPRESENTATION

Advantages

- compact data structure
- rectangular zones work well for tiling and optimizing caches
- efficient to work with neighbors and the different levels

Disadvantages

- requires more cells than necessary
    - grid hierarchy: overlapping
    - larger refinement zone



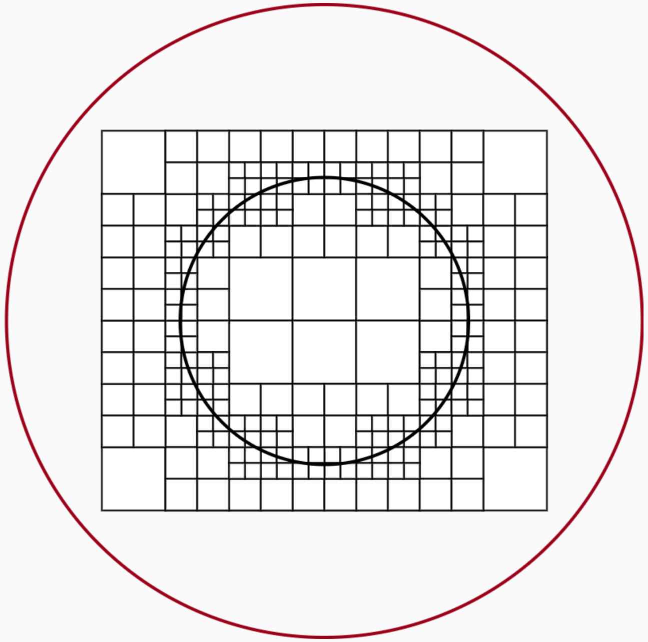
CELL-BASED REPRESENTATION

Advantages

- no grids hierarchy
- keep only needed cells

Disadvantages

- more difficult to find the neighbors
- more difficult to work with the different levels
- tree-like structure
- more difficult to add ghost cells hierarchy



ADAPTIVE METHODS

AMR

Advantages

- based on heuristic criteria (gradient, second derivative, …)
- easy to implement

Disadvantages

- physical problem understanding is important
- add potentially more cells that needed

MRA

Advantages

- based on wavelet decomposition
- error control
- independent of the physical problem
- only needed cells

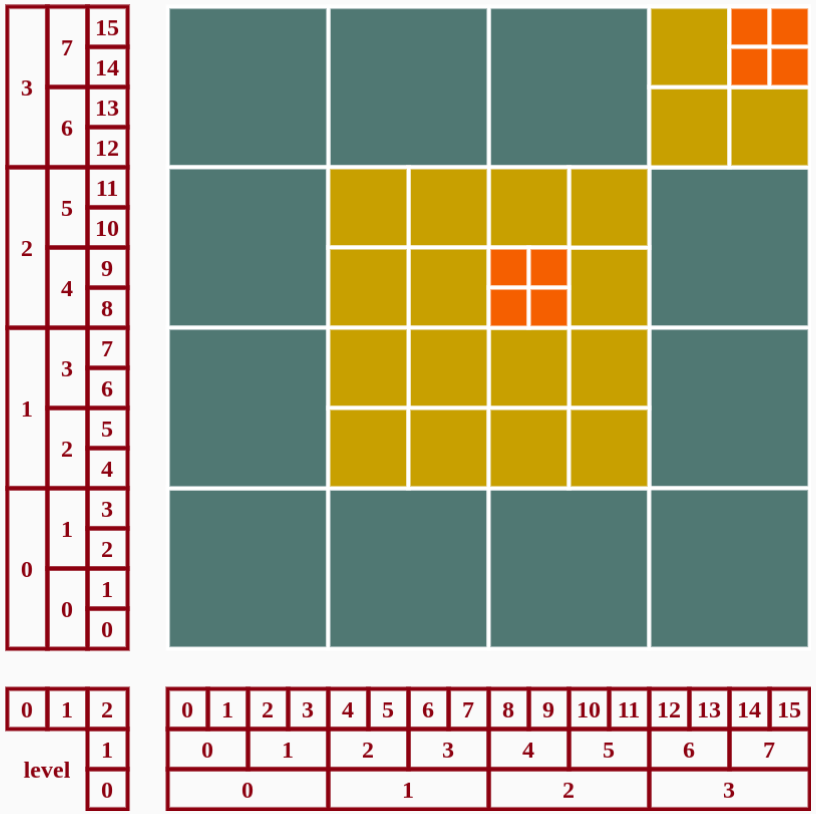
Disadvantages

- difficu[illegible]ement with the current data structures

DESIGN PRINCIPLES

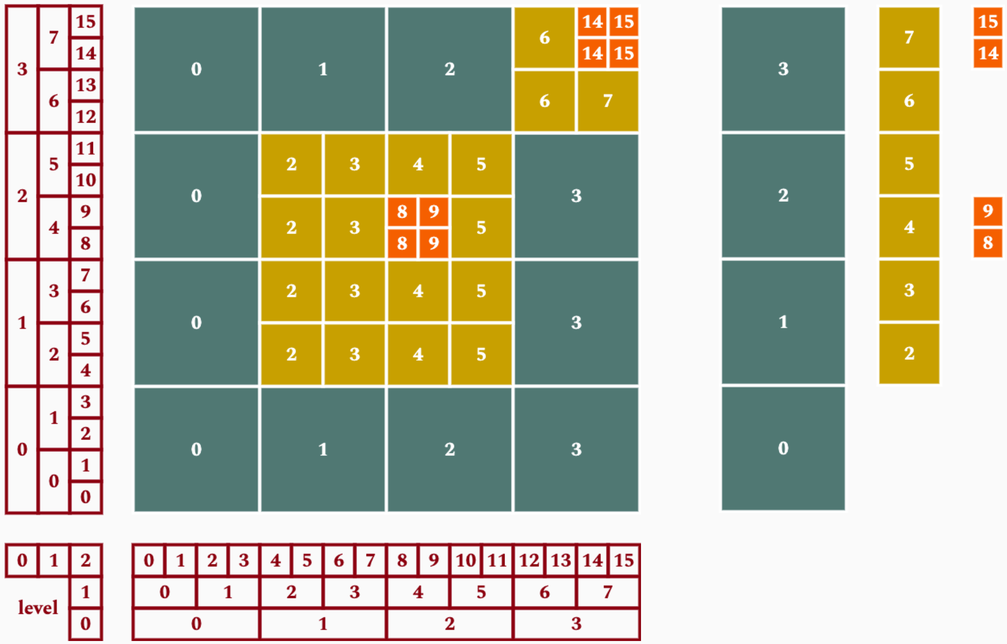
- Compress the mesh according to the level-wise spatial connectivity along each Cartesian axis.
- Achieve fast look-up for a cell into the structure, especially for parents and neighbors.
- Maximize the memory contiguity of the stored data to enable caching and vectorization.
- Manage and facilitate complex operations between meshes (numerical schemes, MRA, ...).

AN OVERVIEW OF THE DATA STRUCTURE



[start ,end) @ offset

AN OVERVIEW OF THE DATA STRUCTURE



Level 0

x: [0, 4[, [0, 1[, [3, 4[, [0, 1[, [3, 4[, [0, 3[

y: [0, 4[

y-offset: [0, 1, 3, 5, 6]

Level 1

x: [2, 6[, [2, 6[, [2, 4[, [5, 6[, [2, 6[, [6, 8[, [6, 7[

y: [2, 8[

y-offset = [0, 1, 2, 4, 5, 6, 7]

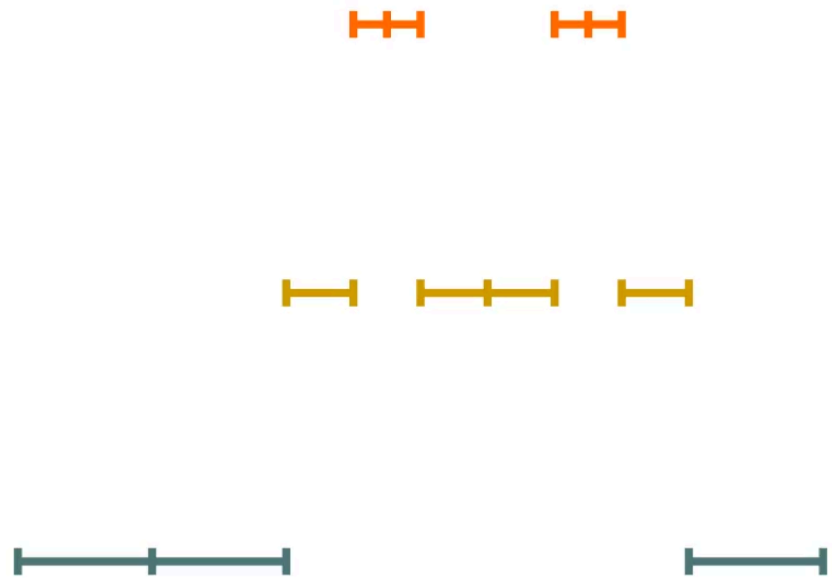
Level 2

x: [8, 10[, [8, 10[, [14, 16[, [14, 16[

y: [8, 10[, [14, 16[

y-offset: [0, 1, 2, 3, 4]

IDENTIFY THE DIFFERENT TYPES OF CELLS

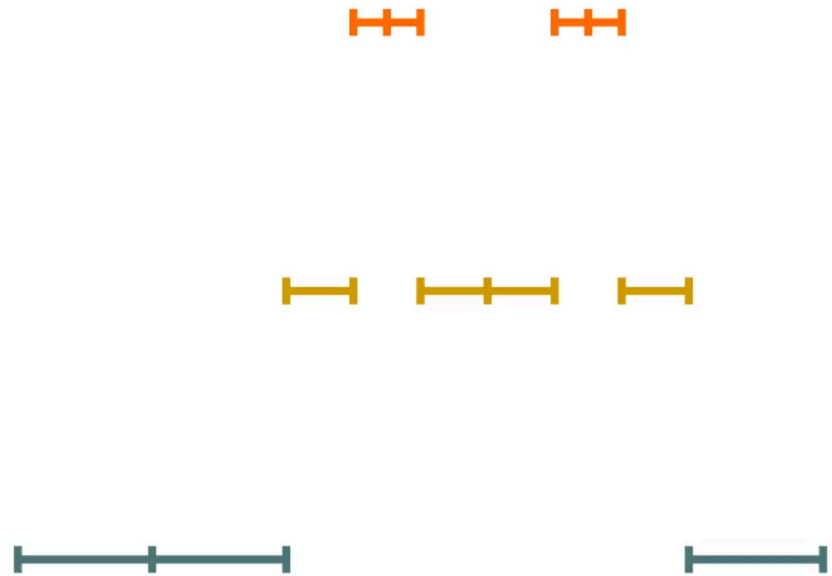


Level 0 $\rightarrow$ $[0, 2[$ $[5, 6[$

Level 1 $\rightarrow$ $[4, 5[$ $[6, 8[$ $[9, 10[$

Level 2 $\rightarrow$ $[10, 12[$ $[16, 18[$

IDENTIFY THE DIFFERENT TYPES OF CELLS

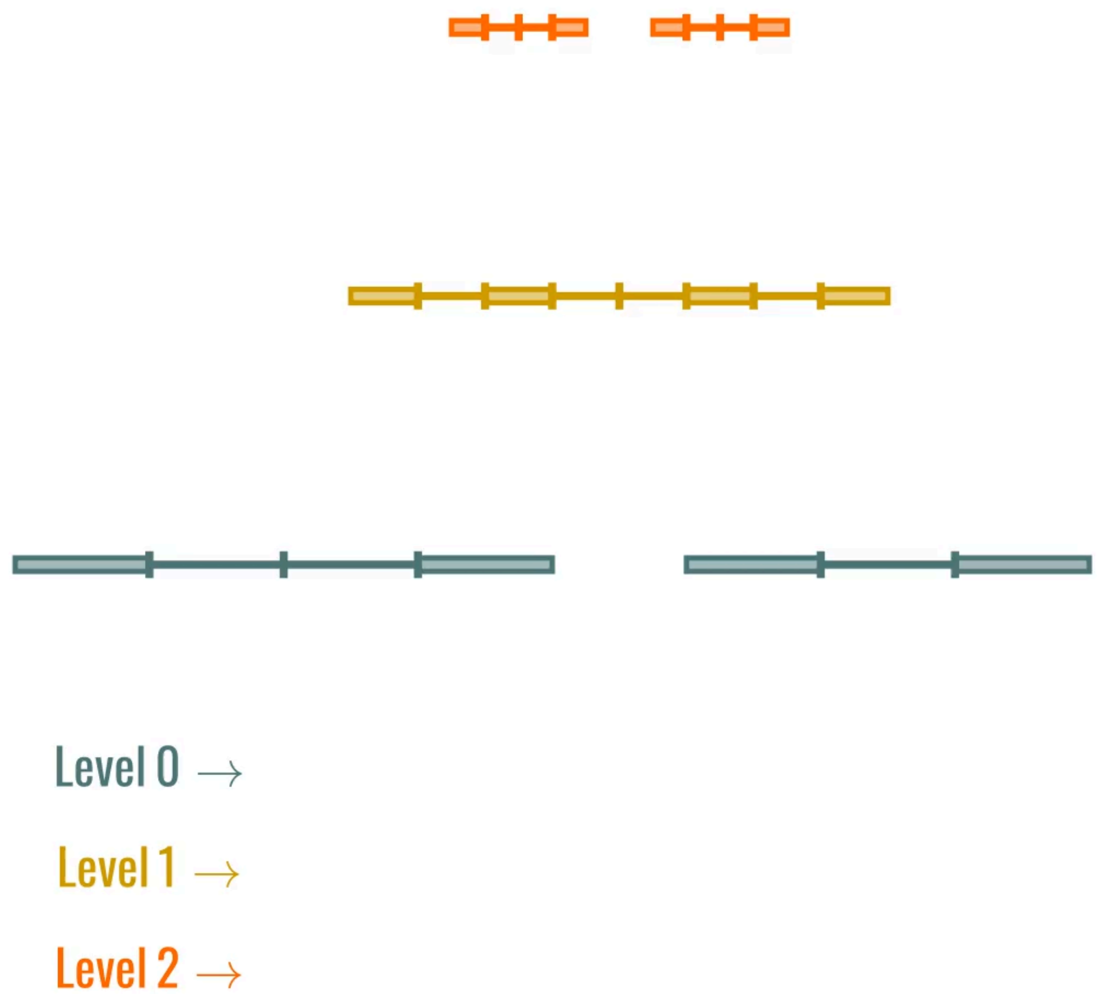


Level 0 $\rightarrow$ $[0, 2[$ $[5, 6[$

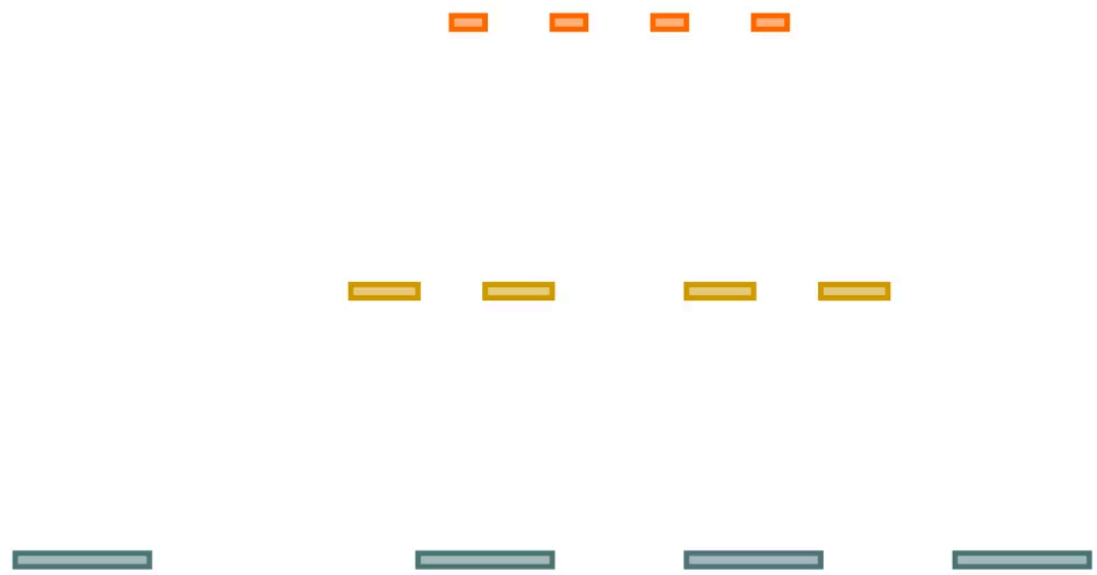
Level 1 $\rightarrow$ $[4, 5[$ $[6, 8[$ $[9, 10[$

Level 2 $\rightarrow$ $[10, 12[$ $[16, 18[$

IDENTIFY THE DIFFERENT TYPES OF CELLS



IDENTIFY THE DIFFERENT TYPES OF CELLS

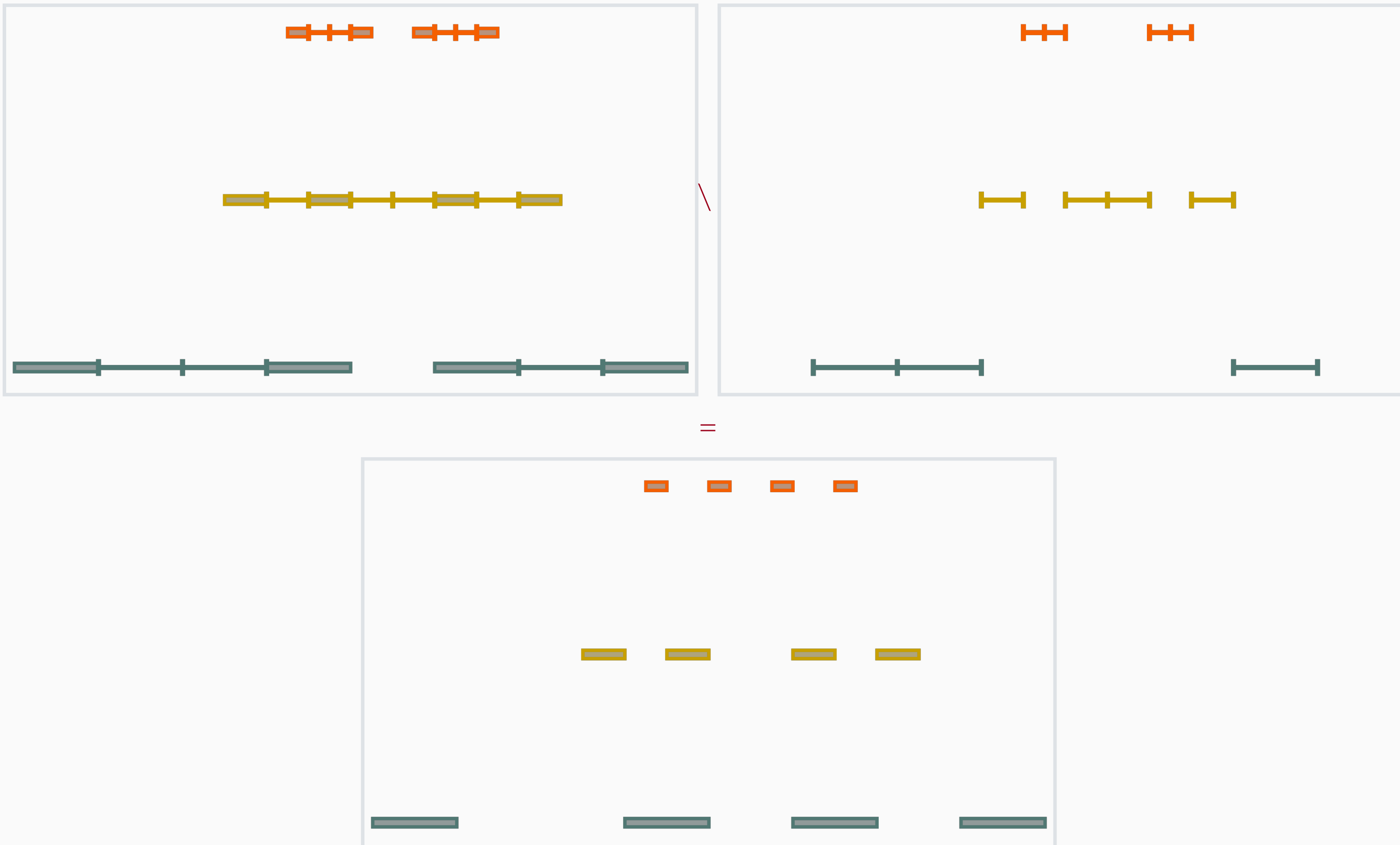


Level 0 $\rightarrow$ $[-1, 0[$ $[2, 3[$ $[4, 5[$ $[6, 7[$

Level 1 $\rightarrow$ $[3, 4[$ $[5, 6[$ $[8, 9[$ $[10, 11[$

Level 2 $\rightarrow$ $[9, 10[$ $[12, 13[$ $[15, 16[$ $[18, 19[$

ALGEBRA OF SETS



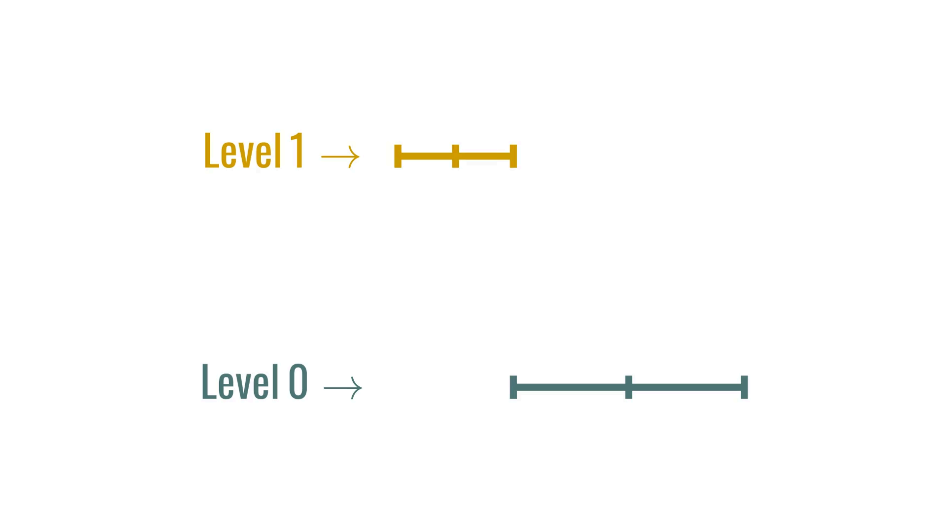
ALGEBRA OF SETS

The search of an admissible set is recursive. The algorithm starts from the last dimension (y in 2d, z in 3d,...).

The available operators in samurai are for now

- the intersection of sets,
- the union of sets,
- the difference between two sets,
- the translation of a set,
- the projection of a set.

USAGE EXAMPLE: FIND JUMPS



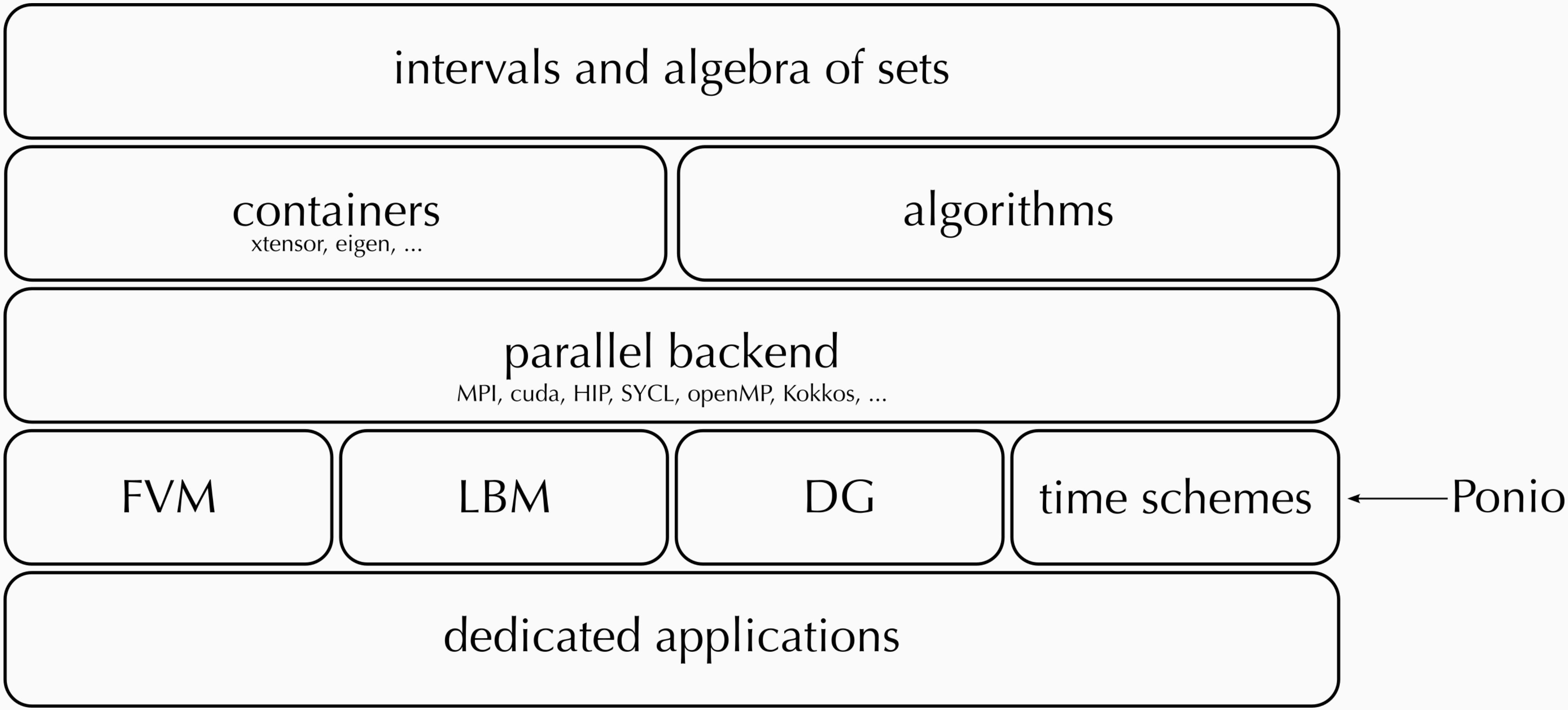

```cpp
auto jump_set = intersection(
                    translate(mesh[1], {1}).on(0),
                    mesh[0]
                );
```

USAGE EXAMPLE: THE PROJECTION OPERATOR

```cpp
auto proj_set = intersection(mesh[level], mesh[level + 1])
          .on(level);

proj_set([&](auto i)
{
    u(level, i) = 0.5*(u(level+1, 2*i) + u(level+1, 2*i+1));
});
```


ROADMAP


intervals and algebra of sets
containers
xtensor, eigen, ...
algorithms
parallel backend
MPI, cuda, HIP, SYCL, openMP, Kokkos, ...
FVM
LBM
DG
time schemes
Ponio
dedicated applications

PARALLELISM IN SAMURAI

CPU

- MPI implementation
- Algebra of set gives the data to exchange
- Various load balancing strategies (WIP)
  - SFC
    - Hilbert
    - Morton
  - diffusion
  - Metis
  - Scotch

GPU

- POC (3 months of work)
  - Two strategies
    - the mesh and the algebra of set keep on the CPU
    - the mesh and an adapted algebra is on the GPU
  - Identify the bottlenecks and the difficulties

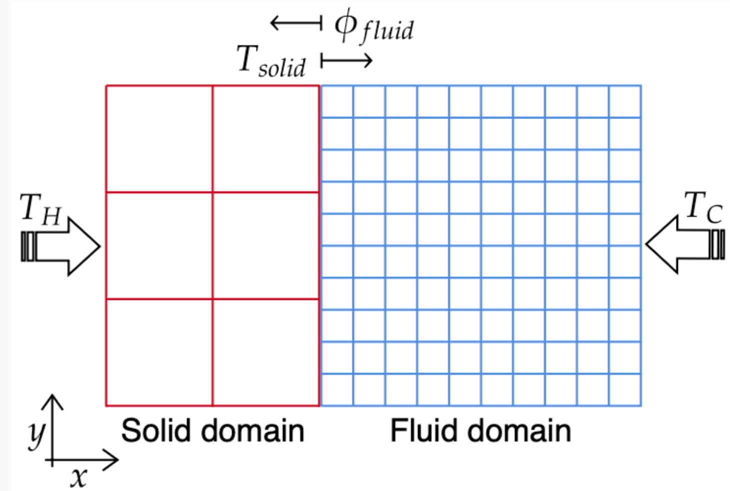
Future collaboration with NVIDIA

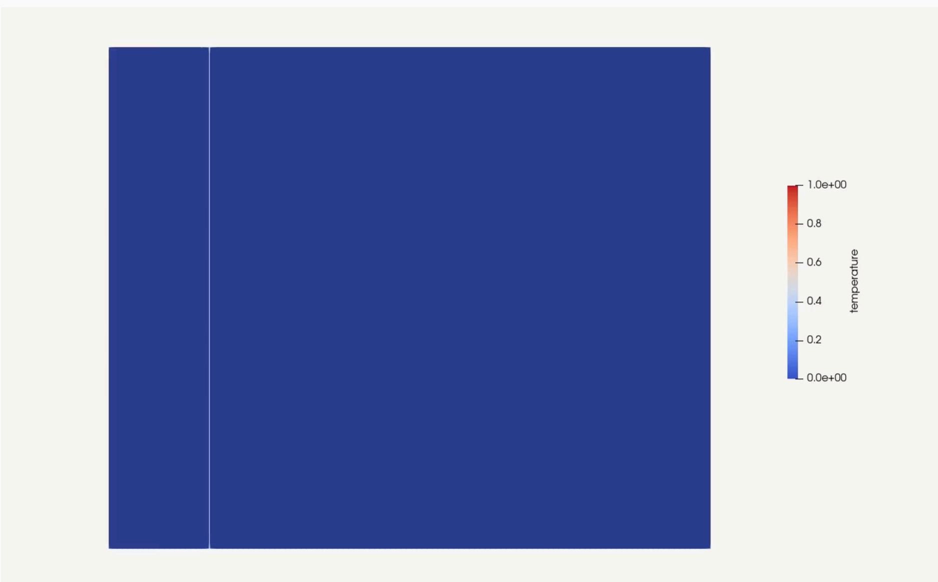
NUMPEX INTERACTIONS

HIGH-ORDER SPACE-TIME COUPLING HPC LIBRARY

Project Overview

- Development of a high-order space-time coupling HPC library for multiphysics applications
- Numerical analysis of the multistep coupling scheme in terms of precision and stability
- Implemented in CwIPI² C++ library





Key Achievements

- Dynamic time step adaptation capability
- Demonstrated on conjugate heat transfer test case (solid-fluid)
- Uses dimensionless equations for both systems

Antoine Simon - PhD thesis funded by NumPEx in collaboration with ONERA

A. Asad, L. François, L. Gouarin, M. Massot

Exa-MA discussion session 26 June 2026

HIGH-RESOLUTION METHODS WITH ERROR CONTROL

Context

- Multiscale reaction-diffusion-convection PDEs (combustion, stiff systems)
- Low-Lewis-number mixtures (hydrogen): thermo-diffusive instabilities
- Need an *a priori* control of errors both in **time** and **space**

Approach

- ImEx (Implicit/Explicit) additive Runge-Kutta schemes
- Controlled operator coupling instead of operator splitting
- Couple ImEx integrators with adaptive multiresolution (`samurai`, `ponio`)
- Discontinuous Galerkin + multiwavelets for local h-p control

Objective

- New generation of high-resolution methods with error control
- New generation of open-source simulation codes
- Applications: hydrogen risk (CEA), plasma & solar physics, nonlinear chemistry

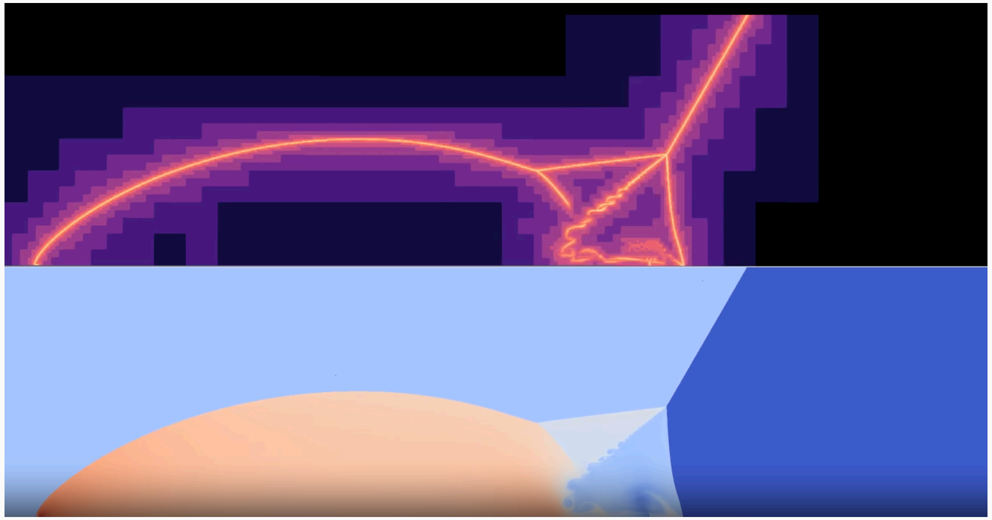
Framework

- M2 internship $\longrightarrow$ PhD thesis: **Fadel Kicha**
- Funded by NumPEx – EXA-MA project
- CMAP / HPC@Maths & EM2C
- L. Gouarin, J. Massot, M. Massot, L. Series, C. Tenaud

EXA-DI INTERACTION - GT BLOCK-STRUCTURED AMR

Project overview

- Identify open source software to compare
- Define physical problem suitable for AMR benchmarking
- Numerical scheme selection
- Establish performance metrics



Key achievements

- First tests with Euler equations
- 4 selected software
  - Dyablo
  - samurai
  - AthenaK
  - Astro

Future Work

- Define reference solutions (double mach reflection, Riemann problems,...)
- Compare accuracy and performance of the different AMR methods and codes

EXA-DOST INTERACTION - AMR I/O

Objective

- Identify I/O requirements for AMR libraries
- Define a common I/O strategy for both AMR libraries
- Allow data compression without or with loss

Methodology

- Bibliography on wavelet compression
- Analysis of various formats for AMR data

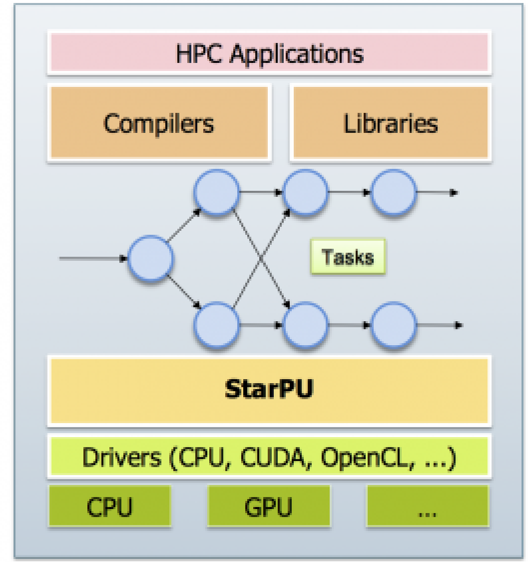
Sylvain Joube (Post-doc - 24 months - started Nov. 2025)

J. Bigot, M. Delorme, A. Durocher, L. Gouarin, M. Stauffert

EXA-SOFT INTERACTION - TASK-BASED PARALLELISM

Objective

- Express `samurai`'s adaptive computations as a StarPU task graph
- Target asynchronous and heterogeneous (CPU/GPU) execution
- Improve load balancing across refinement levels



Methodology

- Identify the right task granularity from the interval-based data structure (per level / per sub-domain)
- Delegate scheduling and dependencies to the StarPU runtime

Involved Person

- **Théo Grandsart** (M2 internship - PhD funding secured)
- Nathalie Furmento
- Loic Gouarin
- Marc Massot
- Raymond Namyst
- Pierre-André Wacrenier

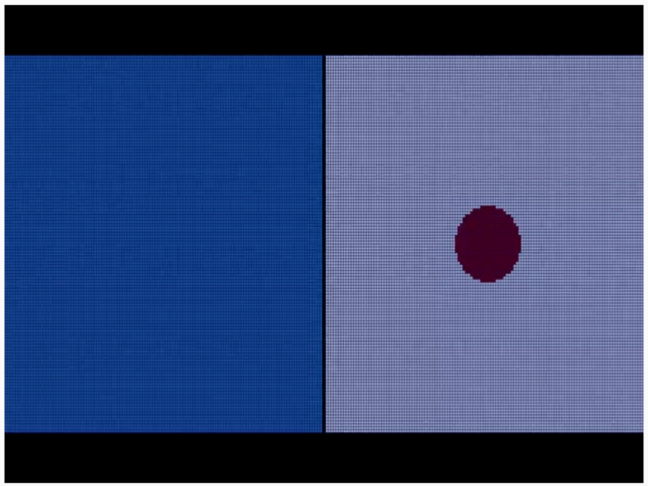
HPC Maths

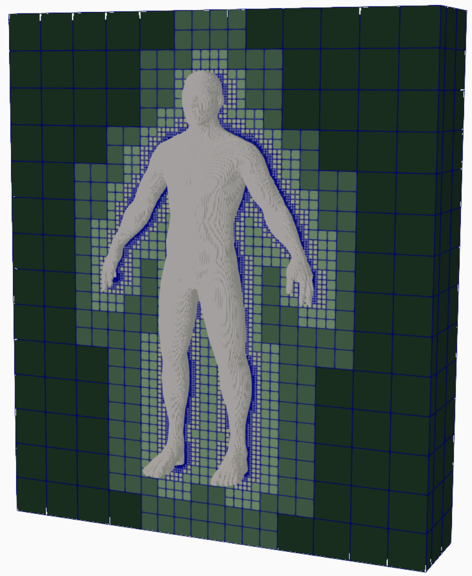
PEOPLE INVOLVED

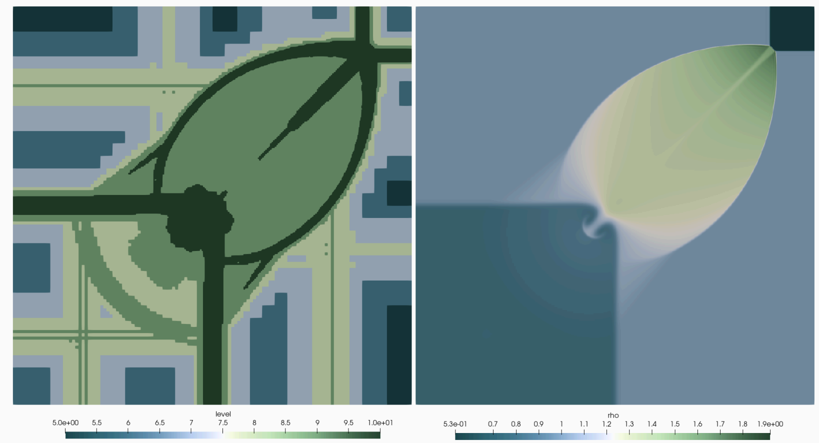
- 4 core developers
- users and developers communities
- Various collaborations and research programs

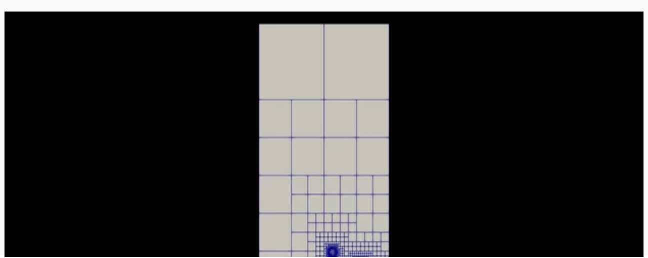
REAL-WORLD APPLICATIONS

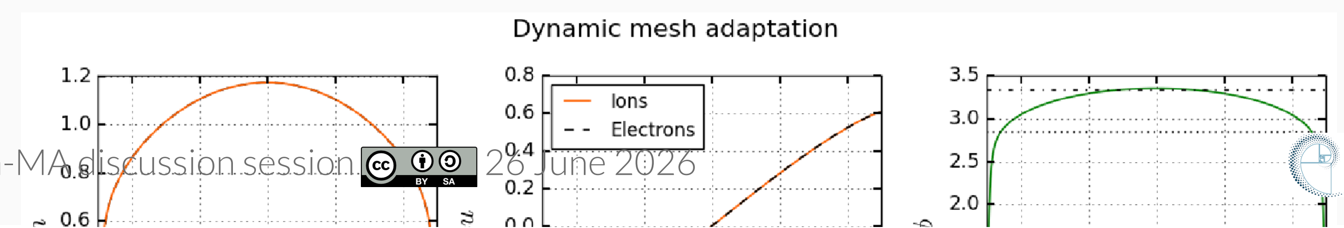
- Interfacial flow simulation
- Simulation analysis on the Hydrogen risk
- Plasma discharges simulations
- Lithium battery simulation
- Combustion DNS
- Code coupling simulations






level
rho




Dynamic mesh adaptation
Ions
Electrons

Samurai

https://github.com/hpc-maths/samurai

Hands-on samurai

https://hpc-maths.github.io/2025-hands-on-samurai

