T8CODE
SCALABLE AND MODULAR AMR FOR ALL ELEMENT SHAPES

Johannes Holke, AMR25 03.09.2025
DLR Institute of Software Technology (SC)
High-performance Computing (HPC) | Scalable adaptive mesh refinement (AMR)

Ole Albers; Niklas Böing; Lukas Dreyer; Sandro Elsweijer; David Knapp; Lena Plötzke; Prasanna Ponnusamy; Thomas Spenke; Johannes Markert; AG Carsten Burstedde; Ioannis Lilikakis; Florian Becker; et. al.

T8CODE
SCALABLE AND MODULAR AMR FOR ALL ELEMENT SHAPES

Johannes Holke, AMR25 03.09.2025
DLR Institute of Software Technology (SC)
High-performance Computing (HPC) | Scalable adaptive mesh refinement (AMR)

Ole Albers; Niklas Böing; Lukas Dreyer; Sandro Elsweijer; David Knapp; Lena Plötzke; Prasanna Ponnusamy; Thomas Spenke; Johannes Markert; AG Carsten Burstedde; Ioannis Lilikakis; Florian Becker; et. al.

The AMR team at DLR-SC

Dr. Johannes Holke

Dr. Prasanna Ponnusamy

Dr. Thomas Spenke

David Knapp (PhD)

Lukas Dreyer (PhD, Uni Hannover)

Sandro Elsweijer (PhD with Uni Bonn)

Niklas Böing (PhD)

Lena Plötzke (PhD)

Ole Albers (PhD planned)

Students

Janot George
Antje Henric-Petri
Faouzi Homsani
Lena Radmer
Tu Nguyen Xuan

Disclaimer

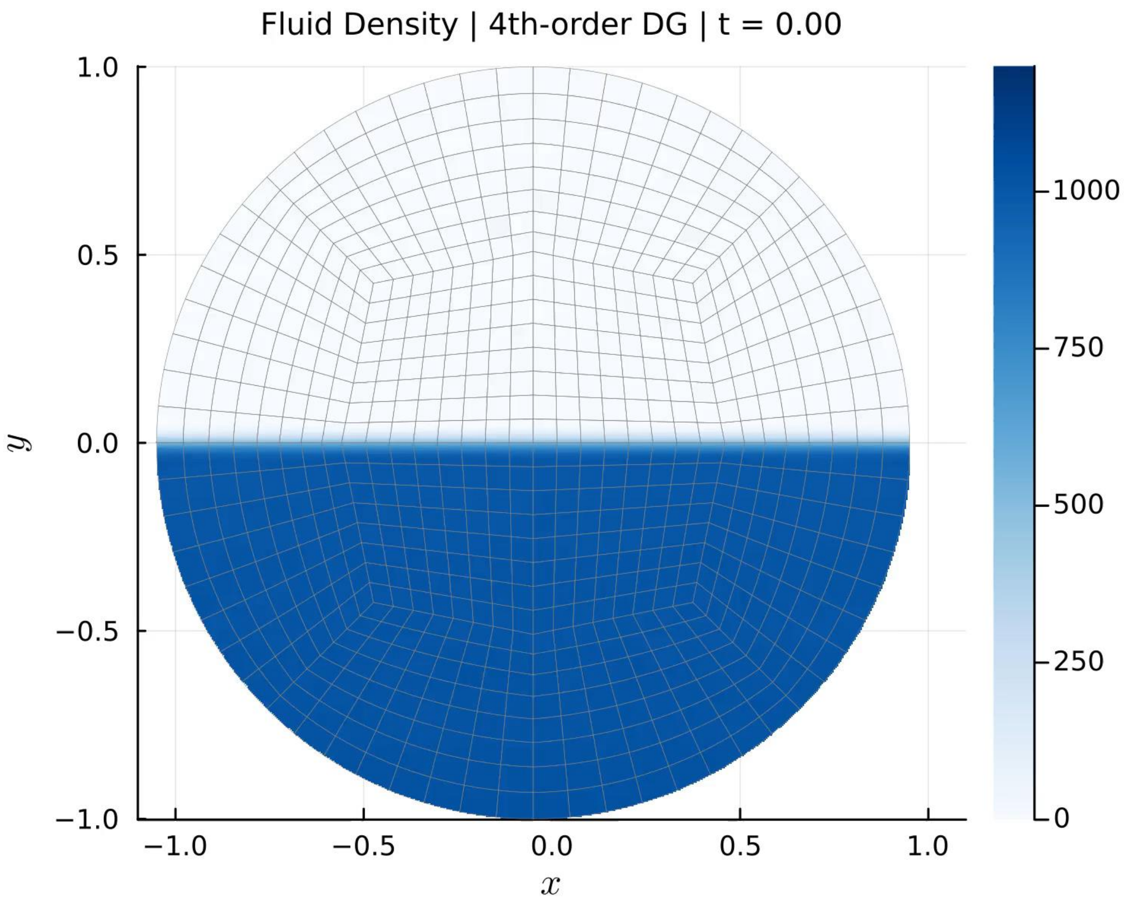
This talk has

99% mesh handling
1% PDEs

Modular tree-based AMR

We have seen a lot of tree-based AMR using space-filling curves

- Memory efficient
- Fast
- Scale to >100.000s cores

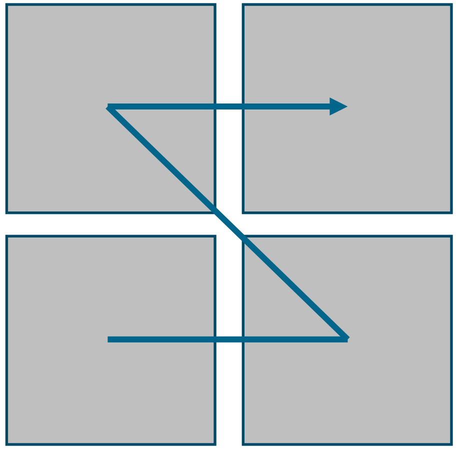


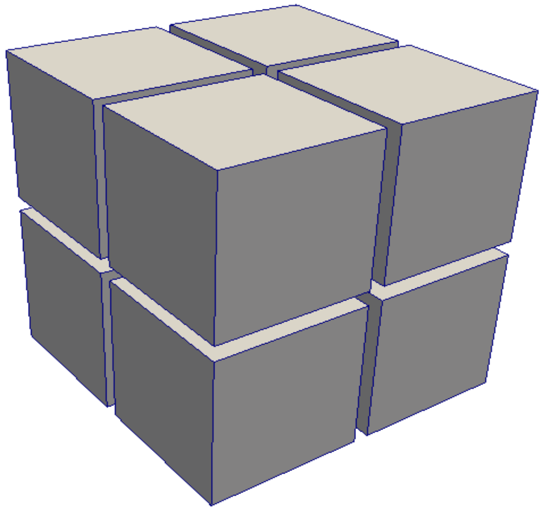
Hydrogen tank sloshing, DLR HYTAZER, Trixi.jl + t8code, Johannes Markert

Modular tree-based AMR

Historically these were **limited to quads/cubes** (with some notable exceptions)

and a **single type of SFC at a time**.





Modular tree-based AMR

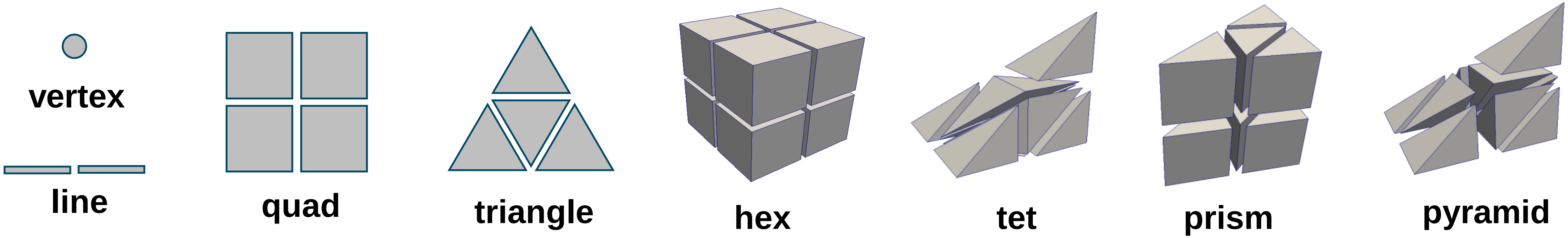
Complex geometries and **industrial use cases** need
tetrahedra, prisms, pyramids
Unstructured meshes:
- runtime and memory intensive
- do not s

t8code
Extend tree-based AMR to **all* element shapes** and **different SFCs/refinement patterns!**

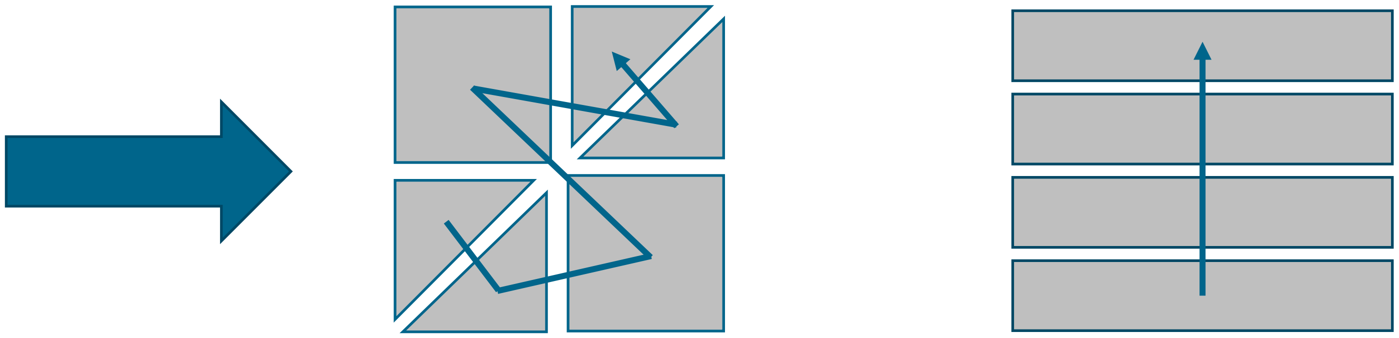
Image source: https://home.centaursoft.com/applications/aerospace/high-lift-aircraft/, DLR

*all refinement patterns?

- Currently implemented:



- Extend with your favourite SFC!



Modular tree-based AMR

High-Level

Mesh…
Adapt
Partition
2:1 Balance
Iterate
Search
Face Neighbor
…

Call when needed

Low-Level

Element...
Get Level
Get Shape
Construct Children
Construct Parent
Construct Neighbor
…

Implement once

Implement for each
Shape (tri, tet, quad, hex, prism, …)
with Refinement pattern/SFC (Morton, Peano, …)

Modular tree-based AMR

Example: refining the mesh.

Instead of:

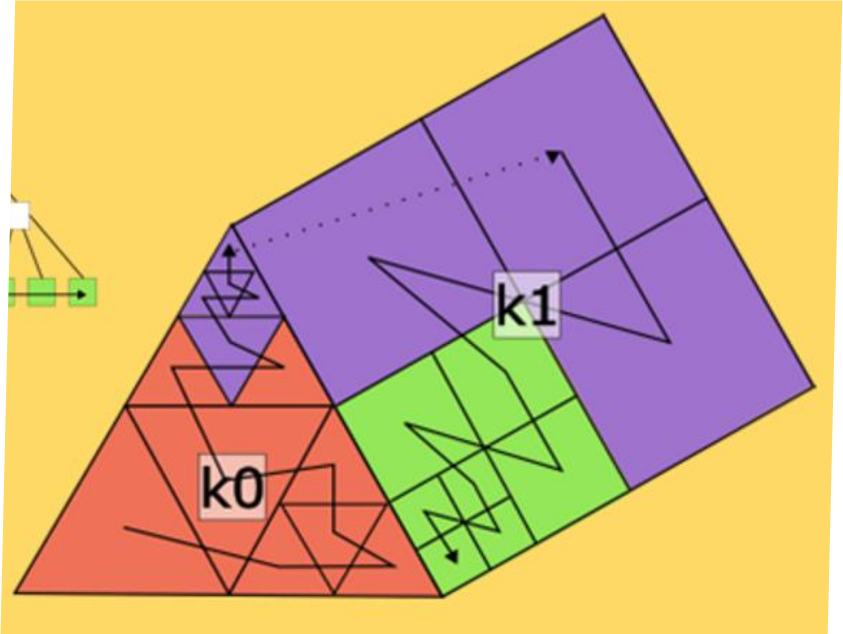
```
if (refine_callback (quad)) {
  const quad_type *children;
  Construct_quad_children (quad, children);
  Append (new_quads, 4, children);
}
```

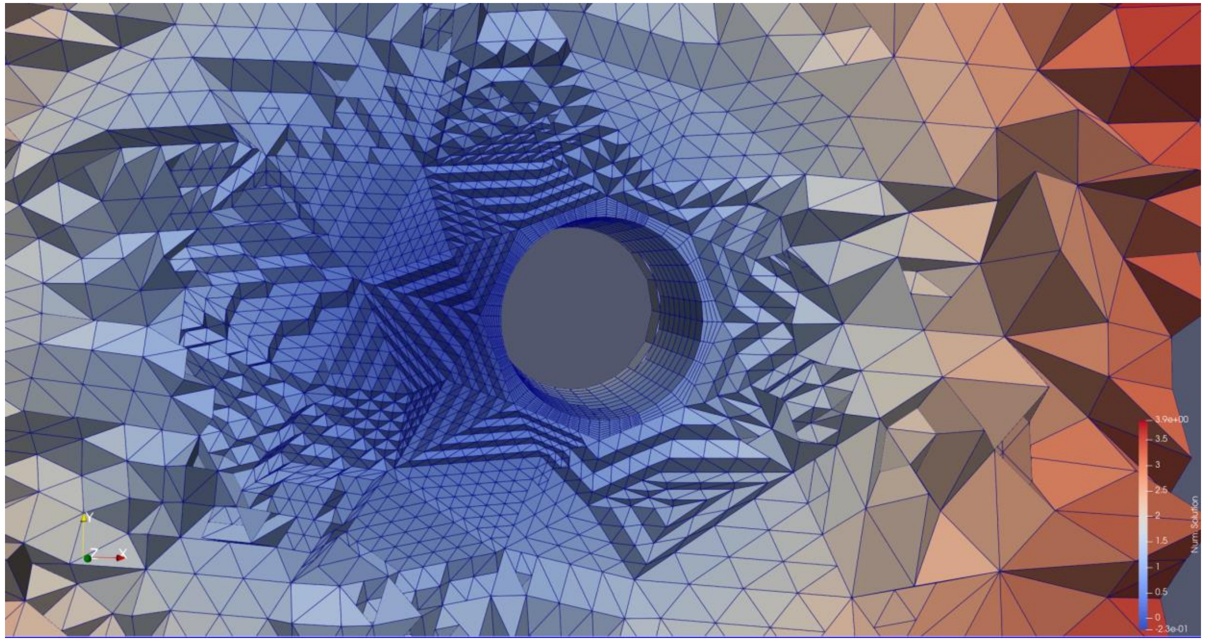
We do:

```
if (refine_callback (element)) {
  const int num_children =
      scheme->get_num_children (tree_class, element);
  const t8_element_t * children =
    scheme->construct_children (tree_class, element);
  Append (new_elements, num_children, children);
}
```

Modular tree-based AMR

A **single implementation** for Adapt, Partition, …, for **any element shape**





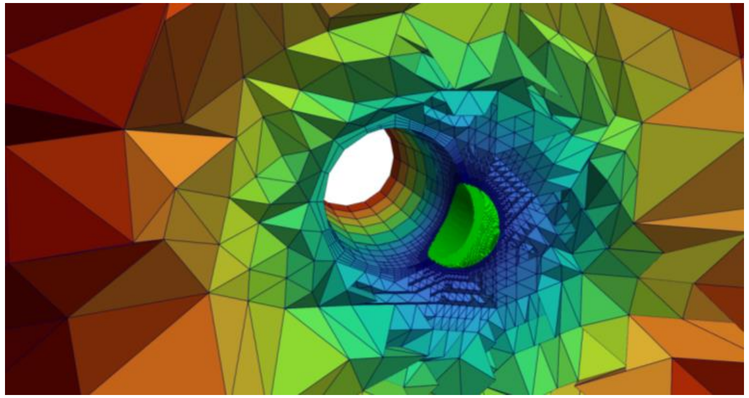
All with the performance and scalability of tree-based AMR!

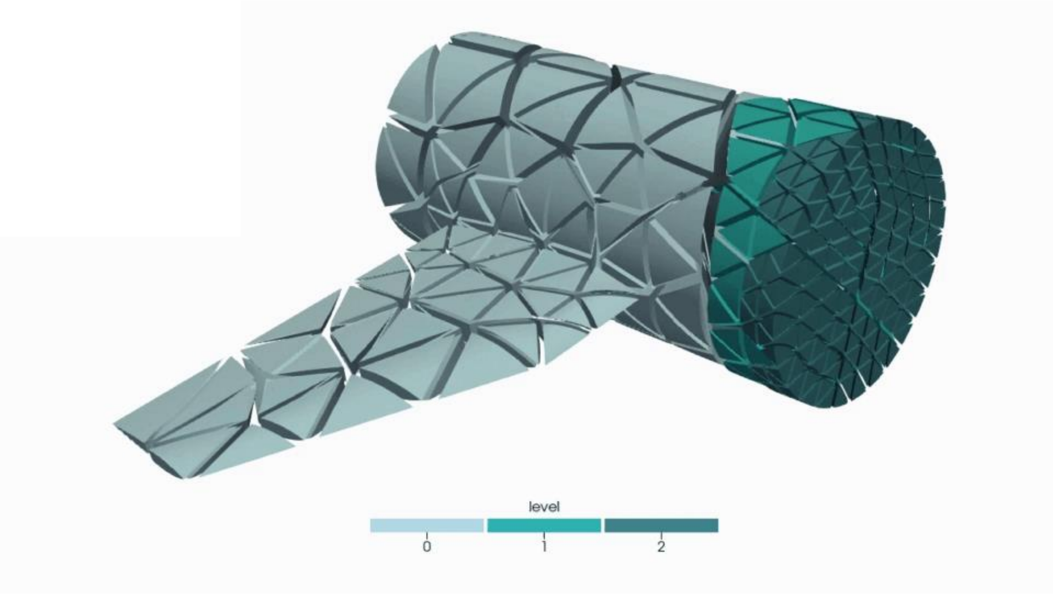
t8code („tetcode“)

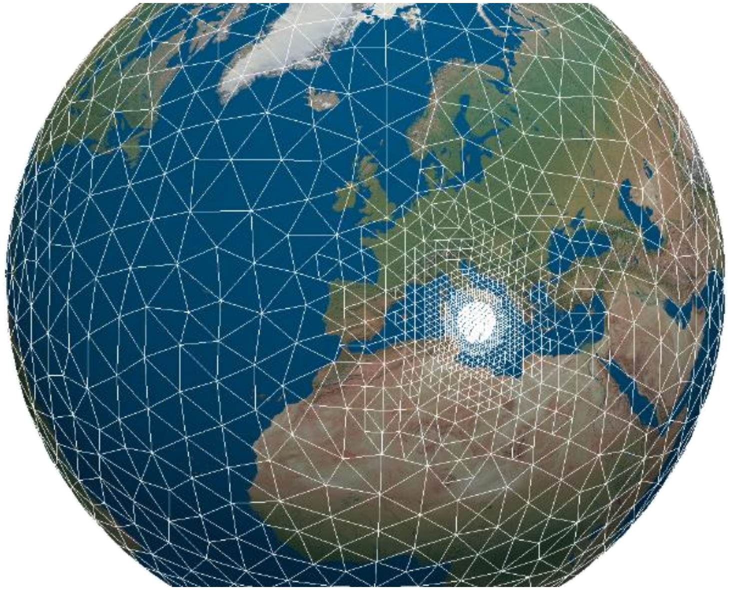
DLR-AMR / t8code Public | Starred 231

DLR

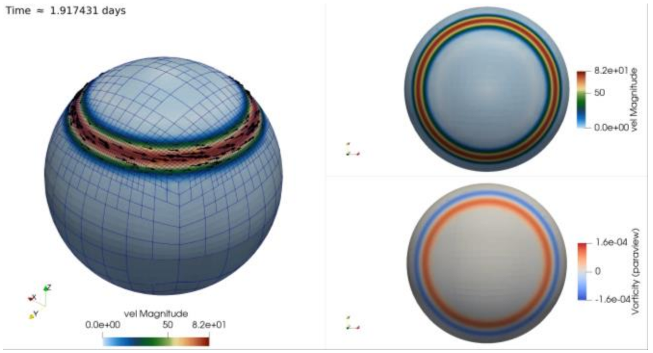
- **Parallel** management of **adaptive meshes** and **data**
- Refine/Coarsen, load-balancing, interpolating, ghost elements, …
- Flexible & **modular thirdparty library**, controlled via callbacks
- **Open Source**
- C/C++ and MPI
- Scales up to **1 mio. MPI** ranks and **1 trillion elements** (> 90% eff.)
- 10 maintainers, 40 contributors, active user community



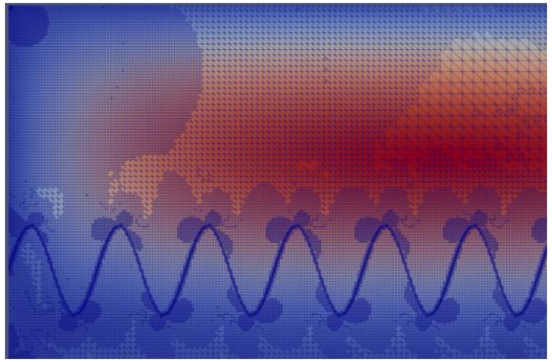




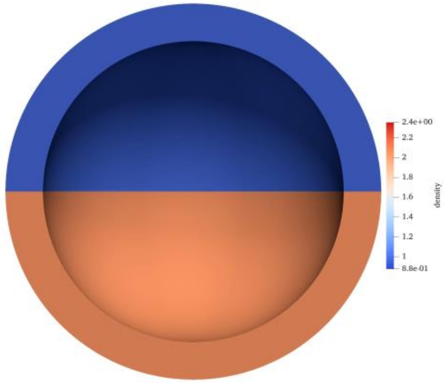
t8code – some applications



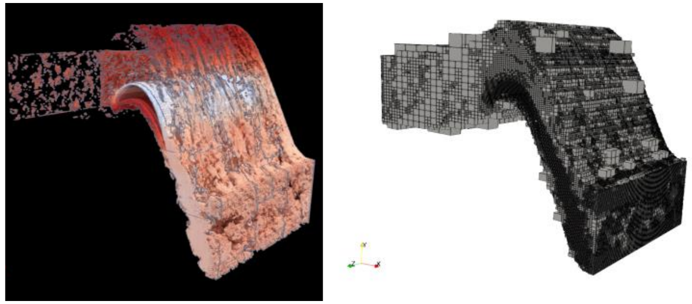
Climate-Chemistry simulations, Trixi.jl + MESSy,
(Johannes Markert, Chiara Hergl, Thomas Spenke)



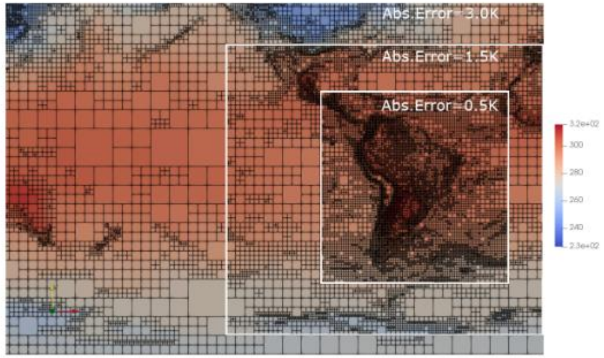
deal.ii coupling (Lukas Dreyer)



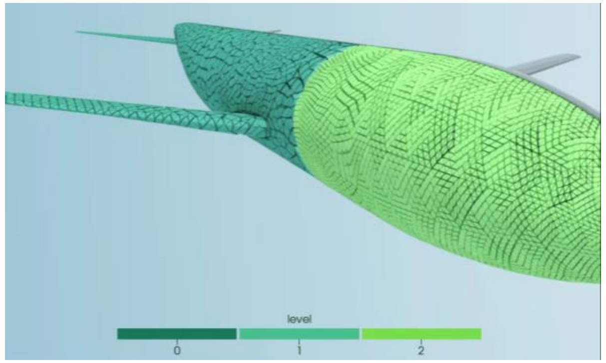
CFD on GPUs (Maël Karembe)



Large scale data analyis, Paraview Plugin
(David Knapp)

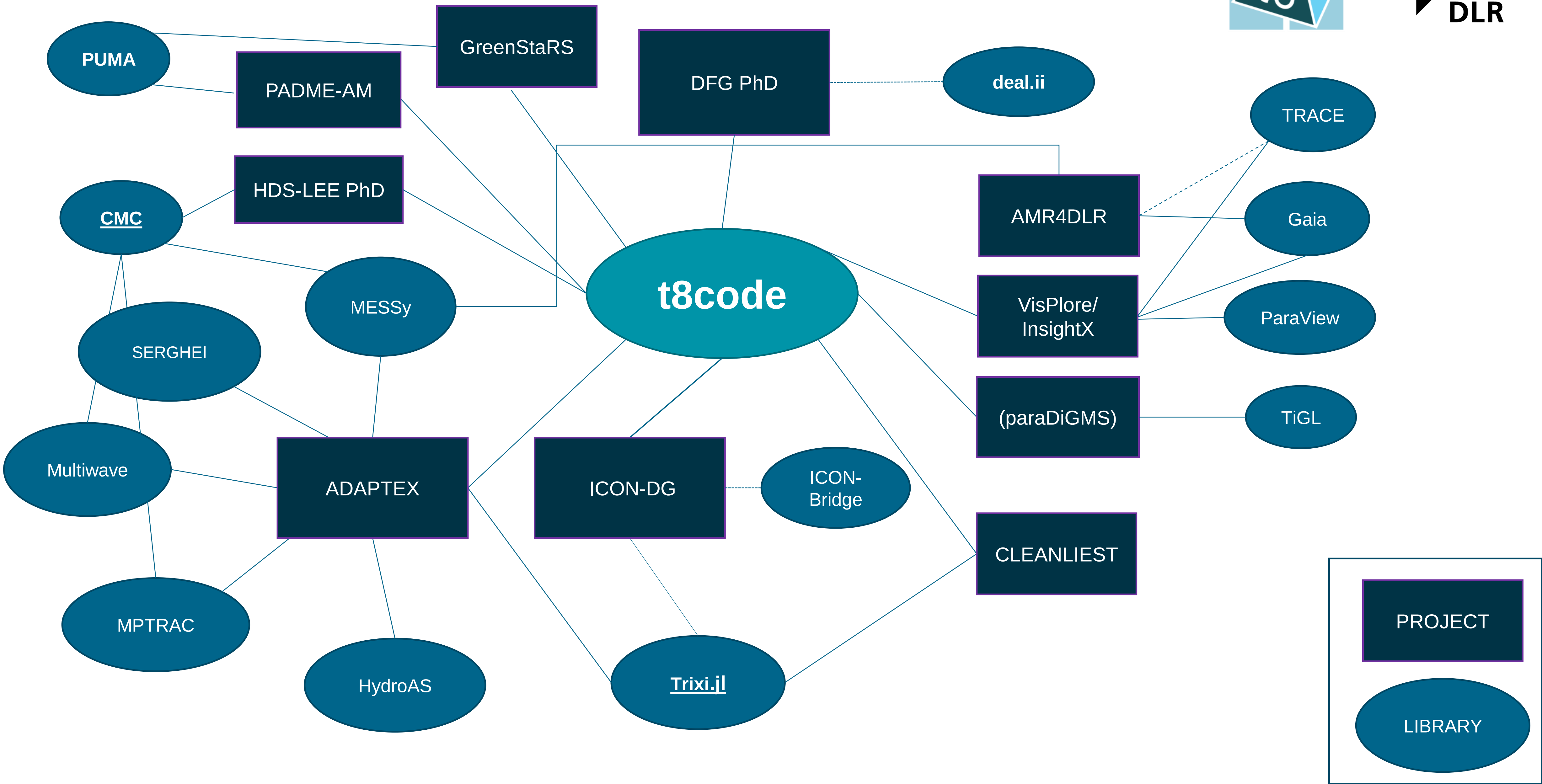


Data compression (Niklas Böing)



CAD aware refinement (Sandro Elsweijer)

t8code – projects and application libraries



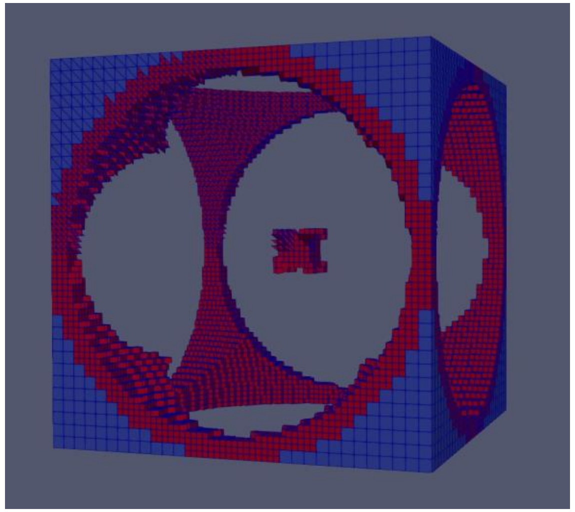
And now, some cool stuff

We were forced to make the high-level algorithms more flexible and robust (changing number of children, changing shape of elements, etc.).

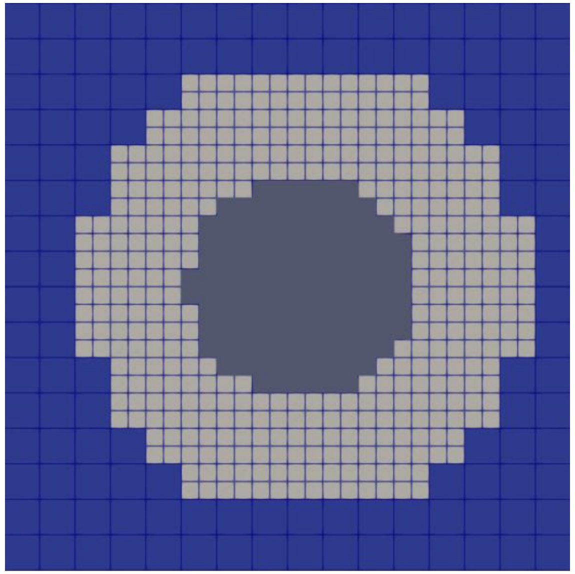
This allows us now to implement „non-standard“ features.

Cutting holes - prototype
Ioannis Lilikakis (now at FZJ)

- Embedding obstacles in the mesh
- Growing meshes

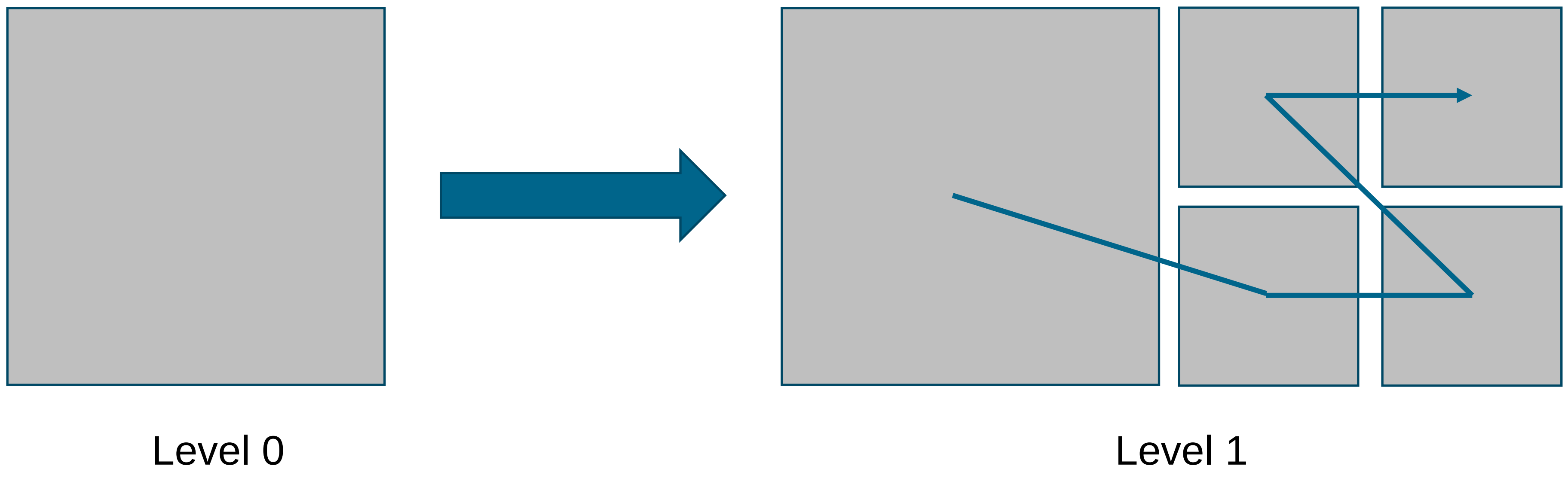


```
if (refine_callback (element) == REMOVE) {
  const int num_children = 0;
}
```



Multilevel SFC
WIP by Sandro Elsweijer with PUMA

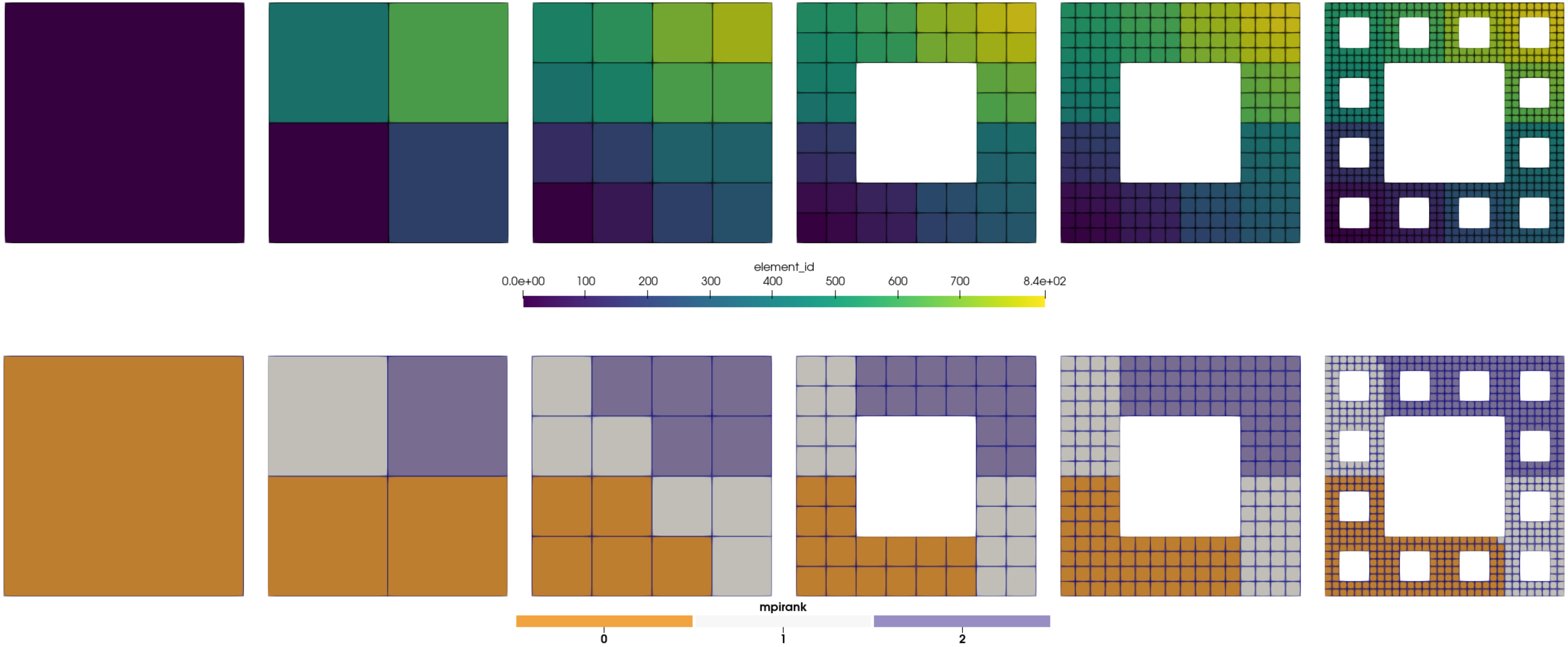
- Sometimes, we need a hierarchy of mesh elements
- We could use multiple forests, but:
  - Interpolation across forests and processes requires careful bookkeeping
- What if all elements are just part of the SFC?



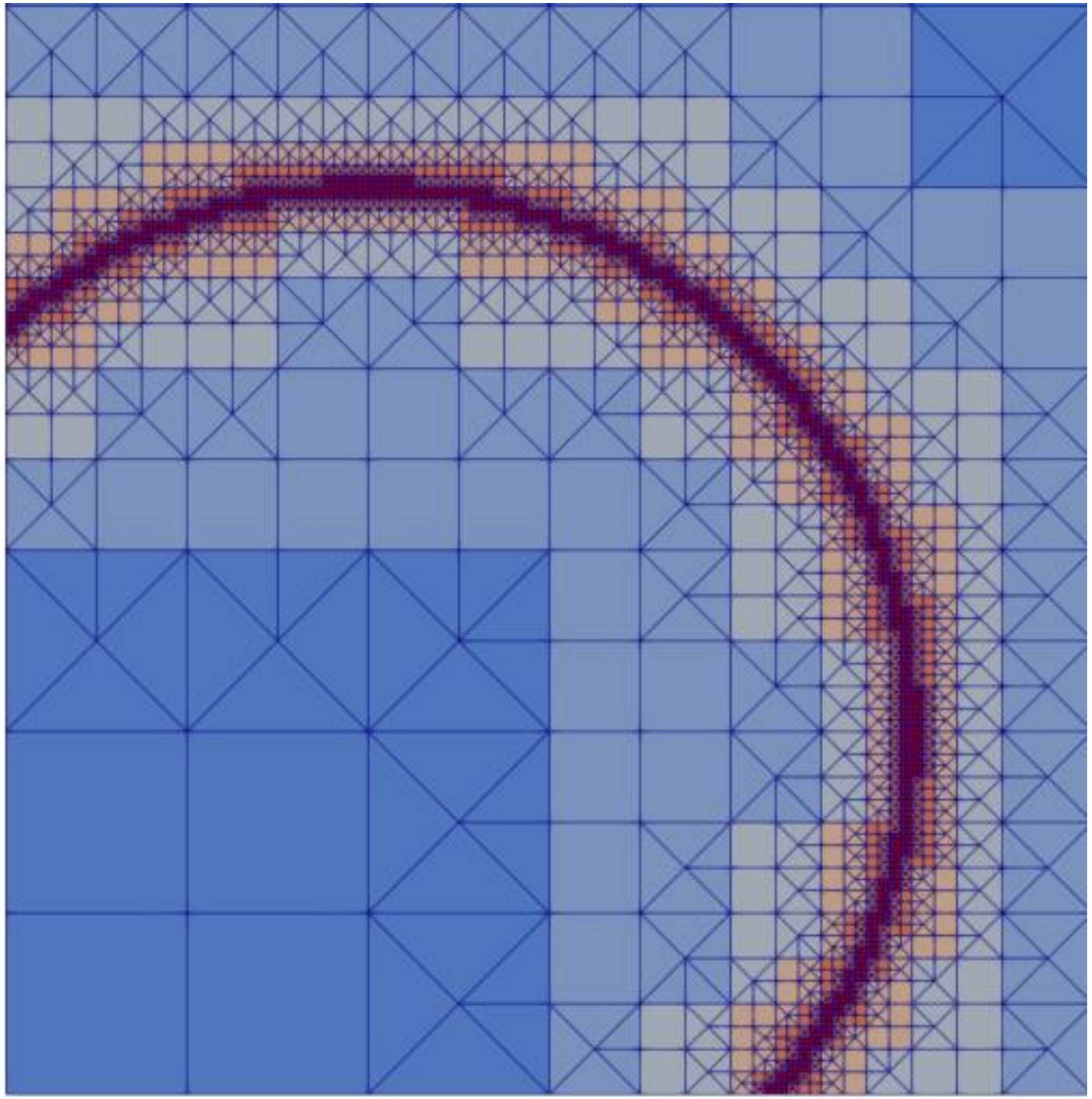
Multilevel SFC
WIP by Sandro Elsweijer with PUMA

These are leaves of a single forest!

Level 0 Level 1 Level 2 Level 3 Level 4 Level 5



Further cool stuff - subelements



This is a **tree-based mesh with a space-filling curve**.
We see **one single quad tree**.

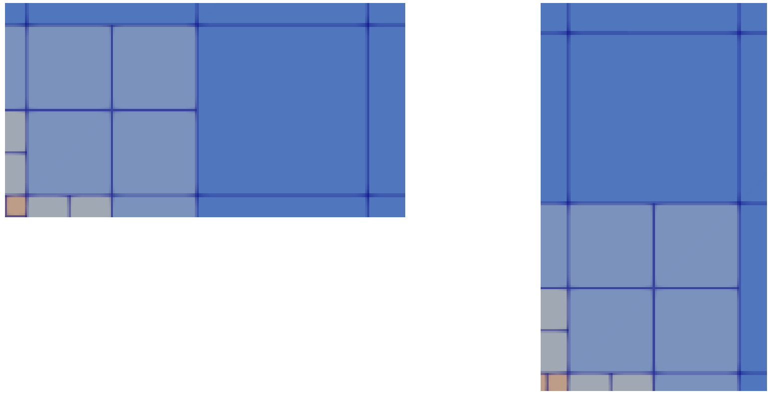
Subelements – resolving hanging nodes
Florian Becker

With standard elements

- We could do different refinement patterns, but…

- An SFC cannot change its behavior at will

„A level X element with Index Y allways has to refine the same way“



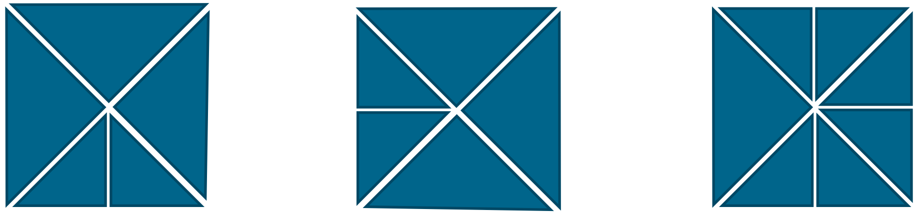
Resolving hanging nodes require
different refinement patterns
for the **same element**
depending on surrounding elements

Subelements – resolving hanging nodes

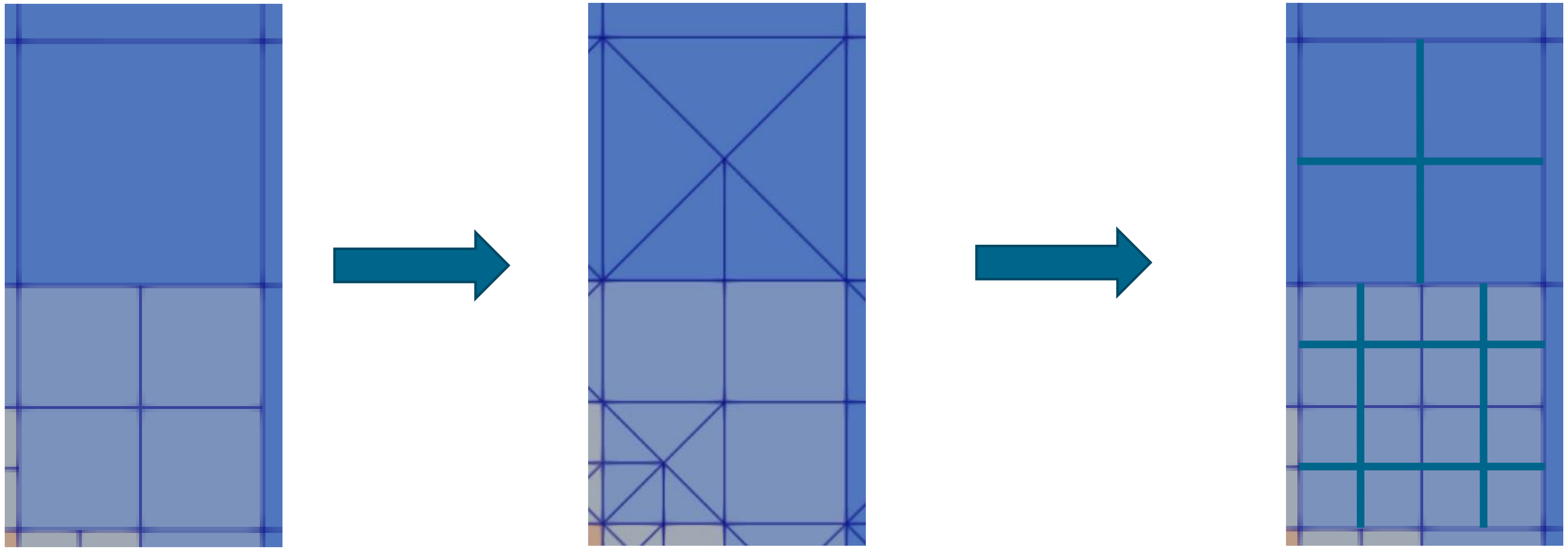
Florian Becker

Idea of Subelements:

- After refinement, do **whatever you want**



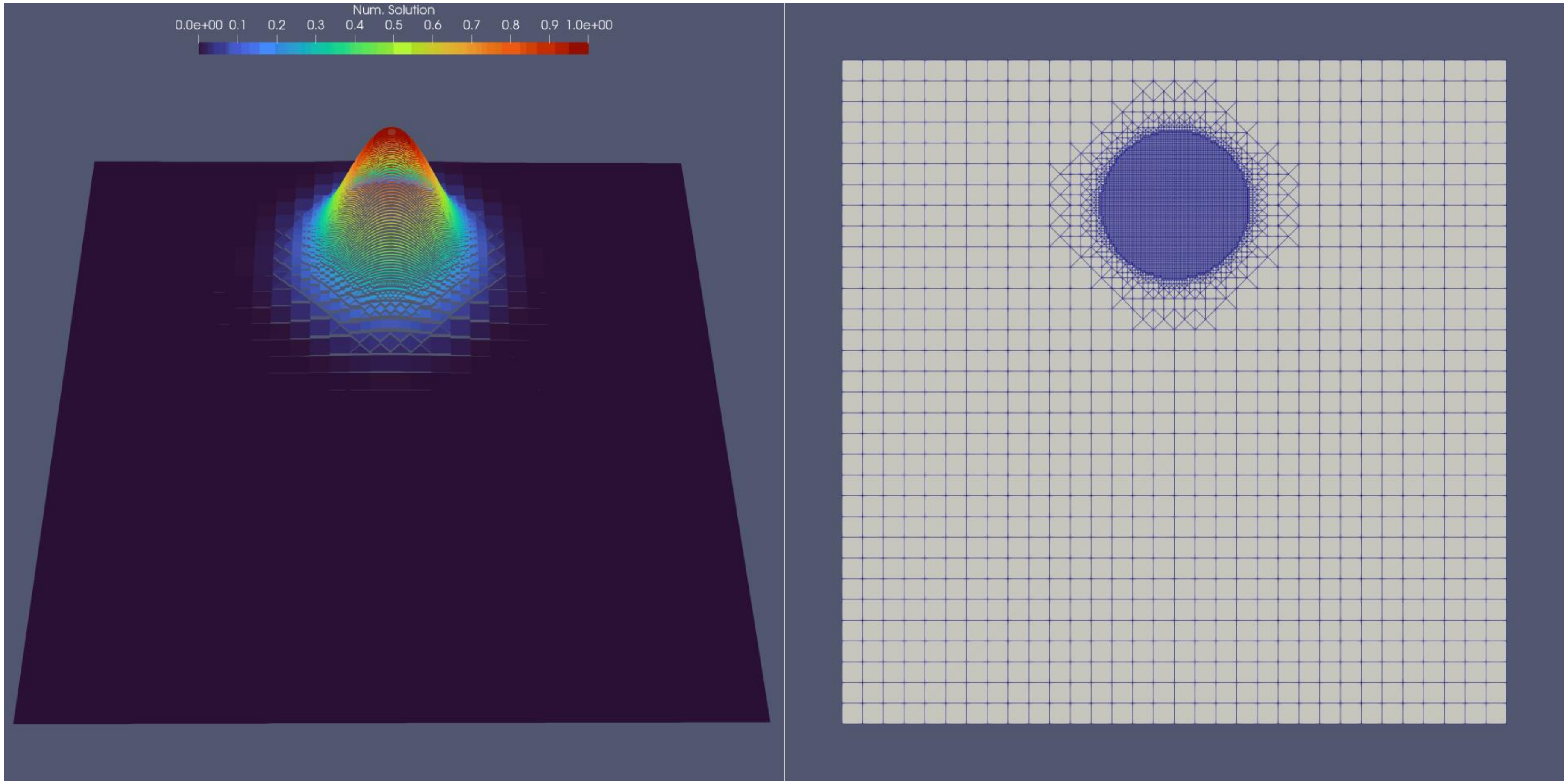
- Before refinement, remove subelements



Subelements – 2D prototype

Florian Becker

We implemented full hanging node resolution for 2D quads with it:

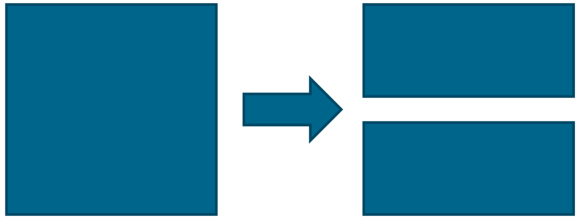


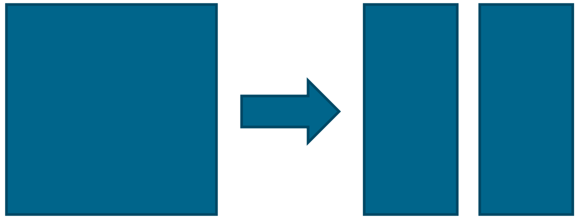
3D hexes (Tabea Leistikow) and other element shapes currently work in progress by Lena Plötzke

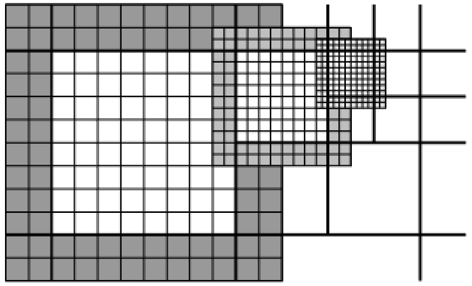
Subelements – What next?

Your imagination is the limit!

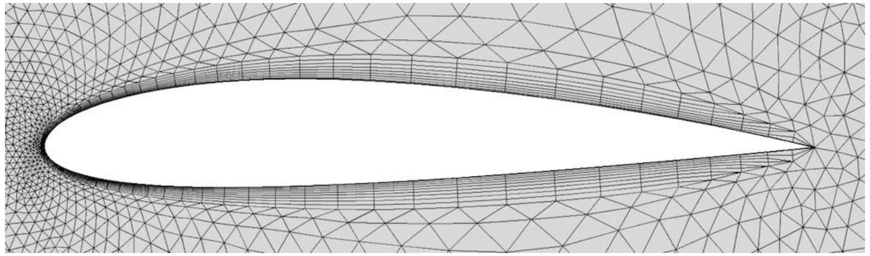
- Anisotropic refinement
- Uniform subgrids for GPUs
- Y+ Boundary layers
- Your ideas?



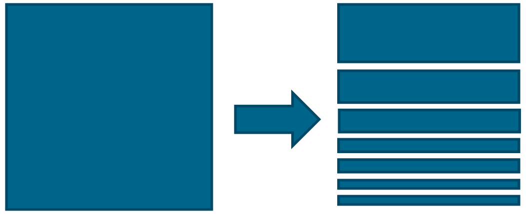




Donna Calhoun et. Al.



https://www.comsol.fr/blogs/your-guide-to-meshing-techniques-for-efficient-cfd-modeling/

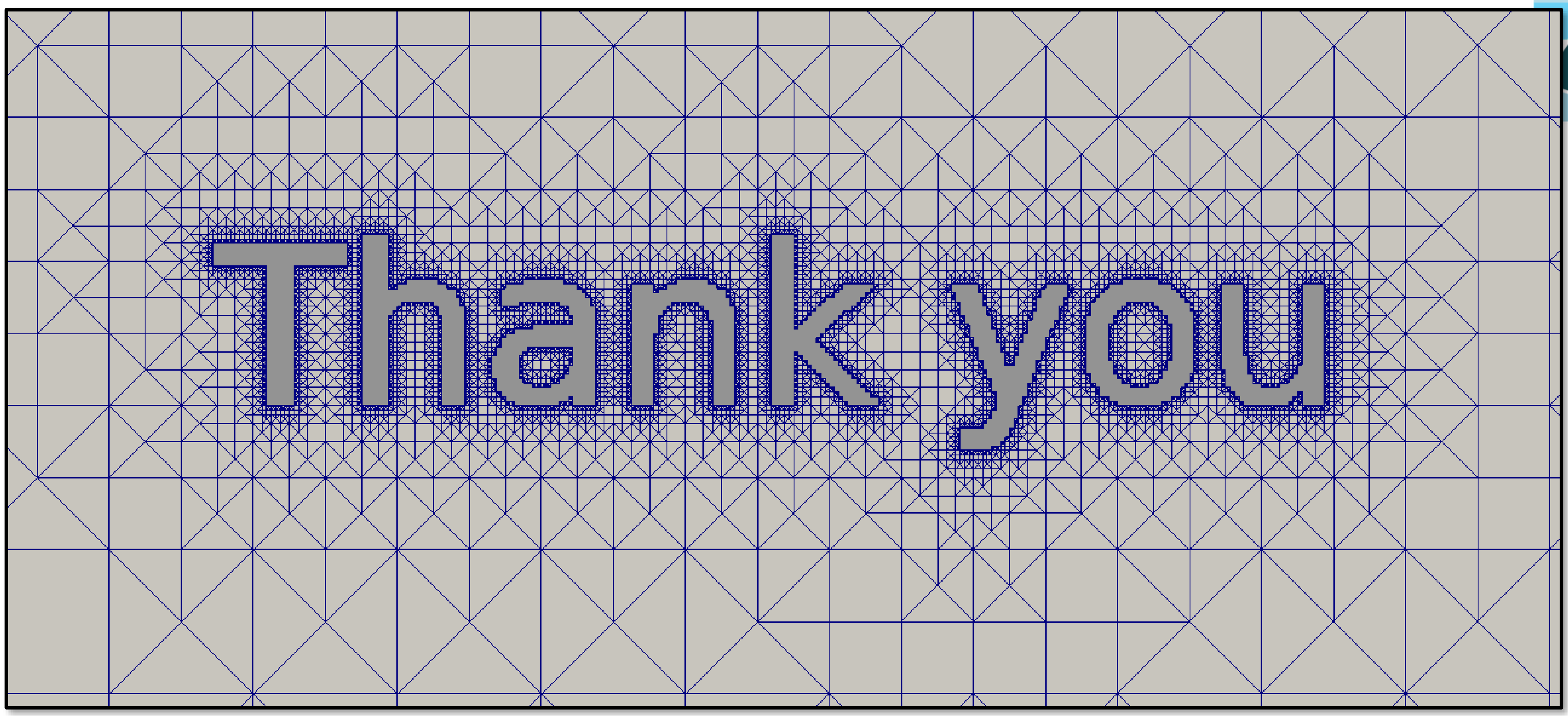


Some of our plans

- Full coupling with deal.ii
- Better interfaces
- Formalize and fully integrate subelements
    - 3D hanging node resolution
    - GPU subgrids
- Enhance GPU support
- Geometry optimization/AD workflows

Some of our Challenges

- Code maintenance (Bug fixing, Code review, User support)
- C++ modernization effort



www.github.com/dlr-amr/t8code

t8code @ AMR25:

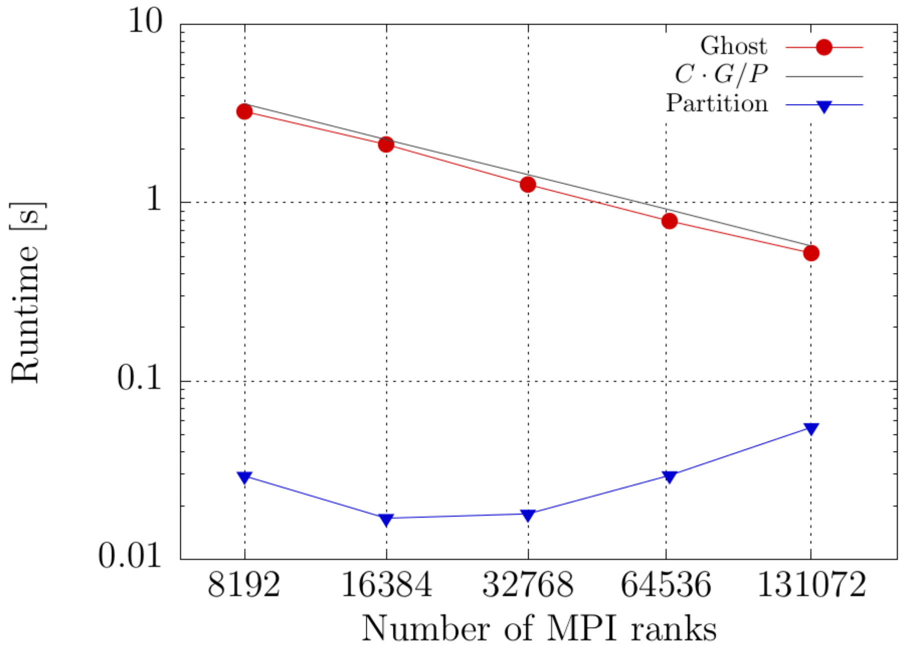
Lukas Dreyer

Space-filling curves for scalable hybrid adaptive mesh refinement

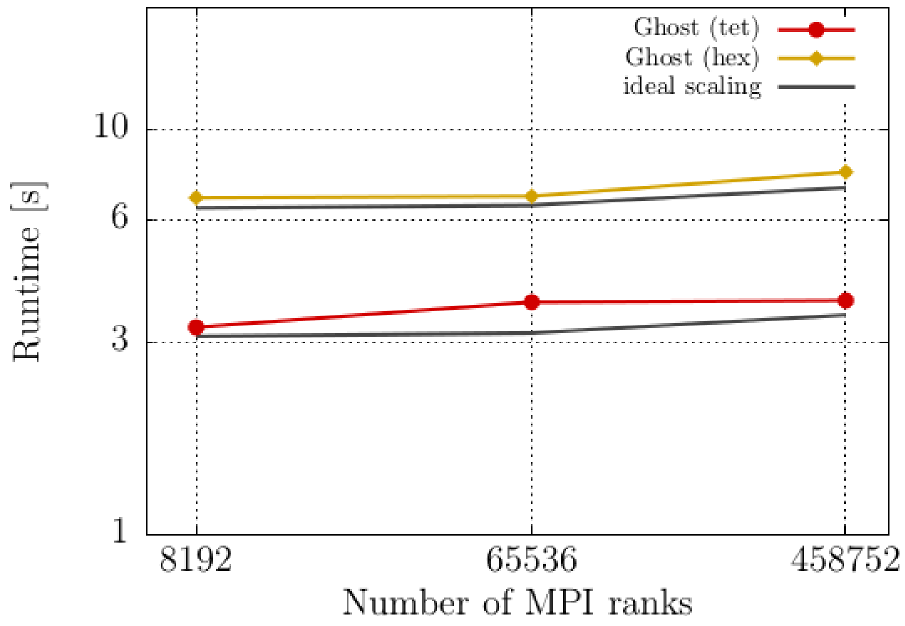
Niklas Böing

Tree-Based Adaptive Data Reduction Techniques for Scientific Simulation Data.

Performance



Strong scale 1.9 billion tets



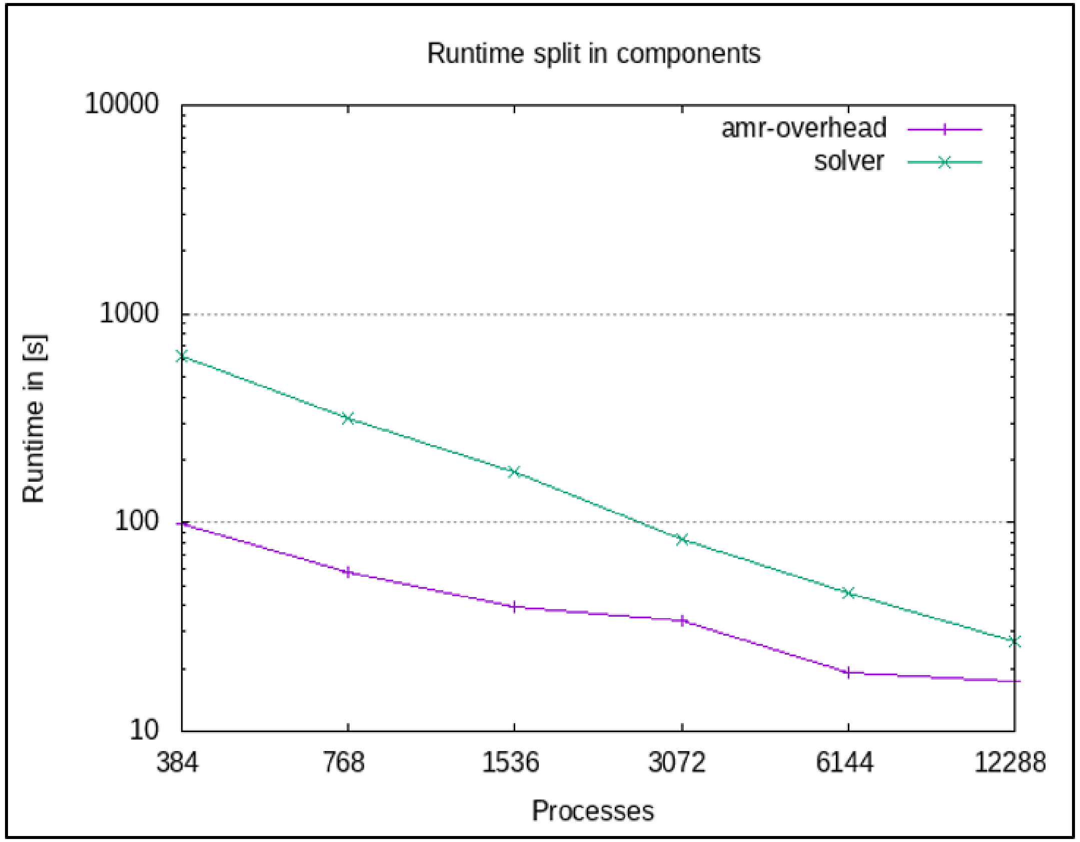
Weak scale, up to 142 billion elements

1 trillion element mesh:

#processes	#elements	#elements/process	ghost
98,304	1,099,511,627,776 ~ 1.1e12	11,184,811	1.43 s

Overhead of AMR?

DG solver on JUWELS



The Local Discontinuous Galerkin Method for the Advection-Diffusion Equation on adaptive meshes
Master's Thesis by Lukas Dreyer at Uni Bonn

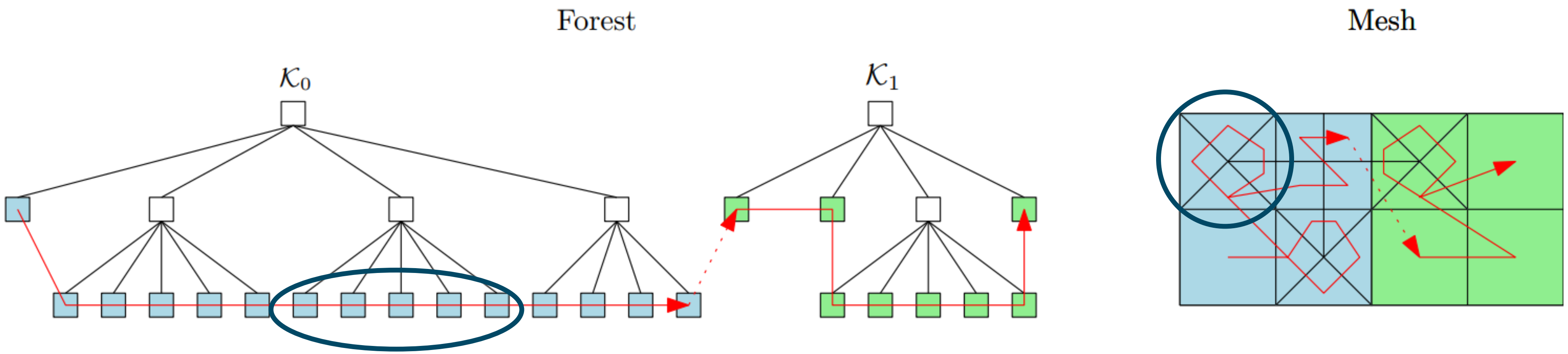
10-15%
mesh management

	Runtime	Error	#DOFs
Uniform 3D	7057s	1.3e-3	16.777.216
Adaptive 3D	561s	1.5e-3	~1.920.000

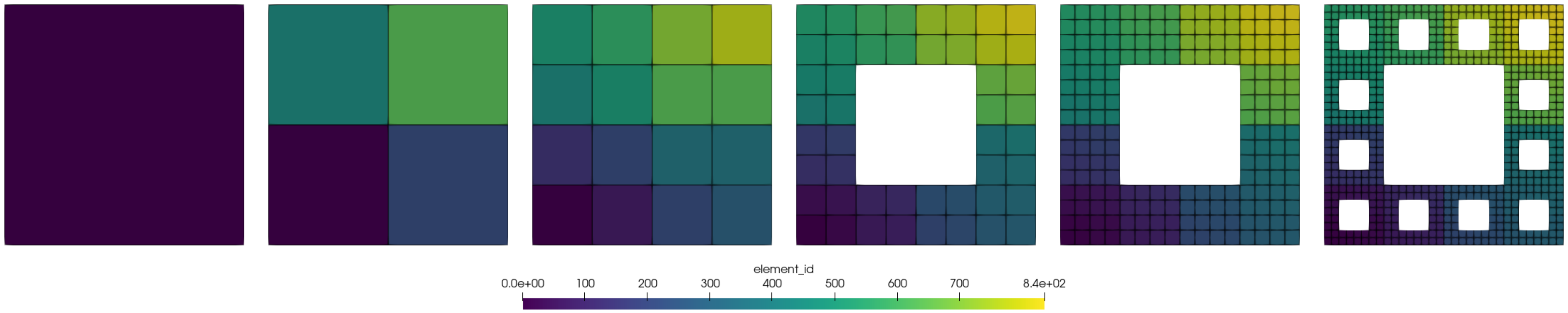
12.6x speedup
8.7x less DOFs

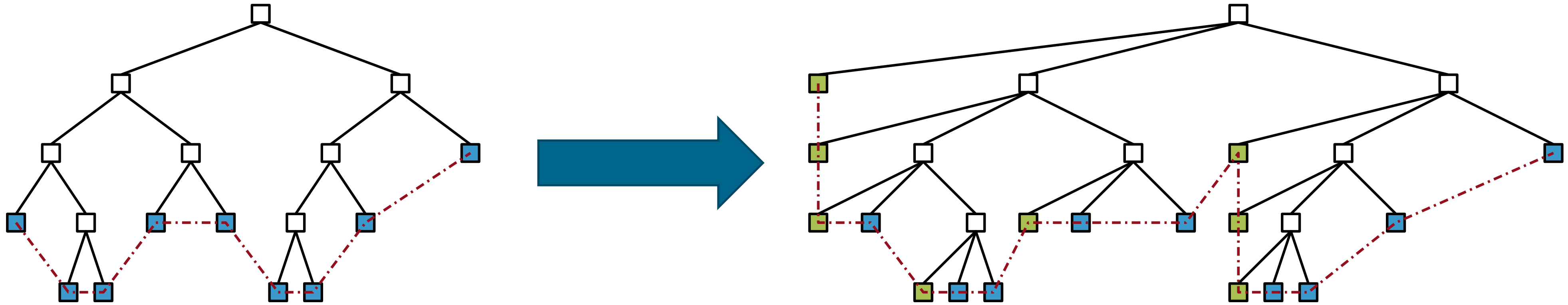
Subelements – resolving hanging nodes
Florian Becker

- Subelements have **same level and SFC index** as their „parent“ element plus an additional **subelement ID**
- Subelements look like elements to the outer world
  - They implement a subset of low-level algorithms
  - Iteration, ghost elements, etc.



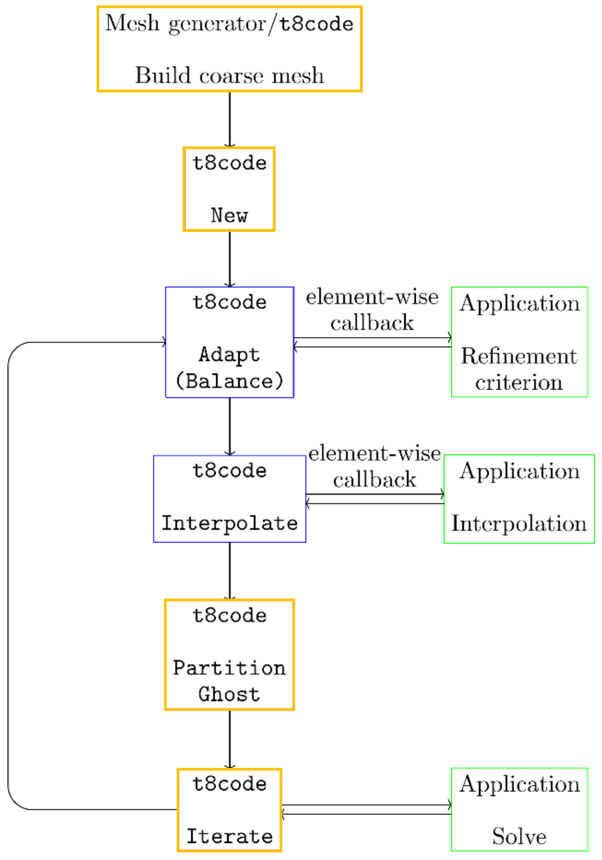
Multilevel SFC
WIP by Sandro Elsweijer





What does the simulation have to implement?

t8code Simulation code



Adaptiv

Application can freely speficy how to

- **Adapt** the mesh
- **Interpolate** the data
- **Solve** the equation