Parallel Concepts

Christina Stöcker (stoecker@caesar.de),
Dr. Angel Ribalta (ribalta@caesar.de),
Simon Vey (vey@caesar.de),*
Dr. Axel Voigt (voigt@caesar.de)

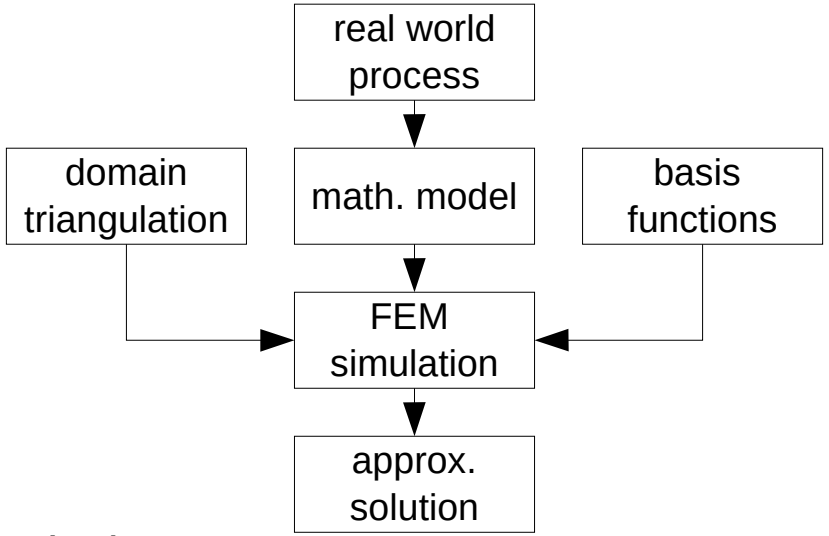
research center caesar – crystal growth group
Ludwig-Erhard-Allee 2
53175 Bonn, Germany

What is AMDiS?

AMDiS (Adaptive MultiDimensional Simulations)

- C++ library for the numerical solution of PDEs of 2nd order

$$\partial_t u - \nabla \cdot A \nabla u + b \cdot \nabla u + cu = f$$



  - Implements the finite element method (FEM)
  - Local mesh adaption based on local error estimates
  - Enables
    - Computation in 1d, 2d, 3d
    - Coupling of problems (same or different dimension)
    - Solution of PDE systems
  - Design goals: high level of abstraction, generality, extensibility, efficiency

c aes ar

crystal growth group

http://www.caesar.de/amdis.html

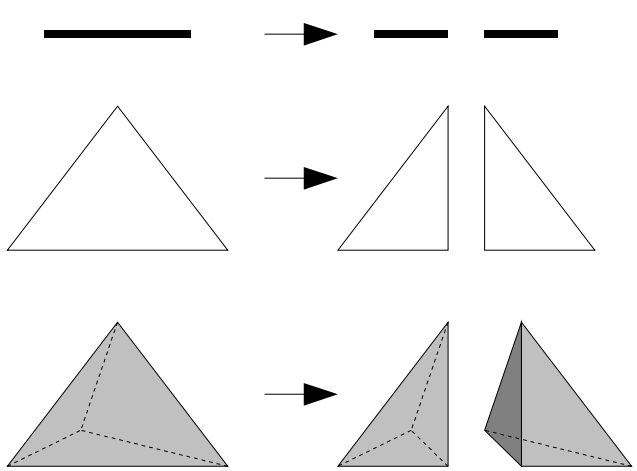
Adaptivity

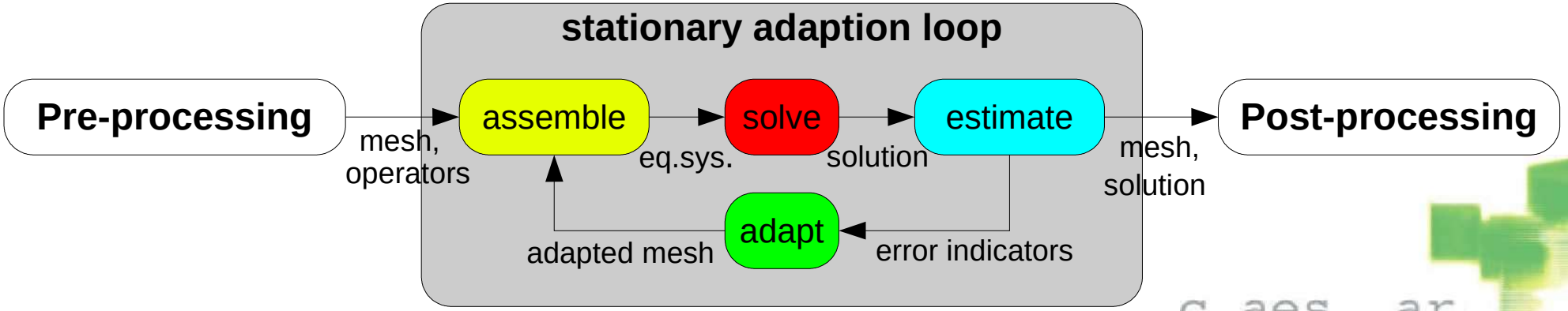
Idea

Reach given error tolerance with minimal effort

1. Start with a **coarse macro triangulation**
2. Calculate solution (assemble, solve)
3. Estimate local and global **error indicators**
4. If tolerance not reached
   -> refine elements with large errors
   -> Goto 2

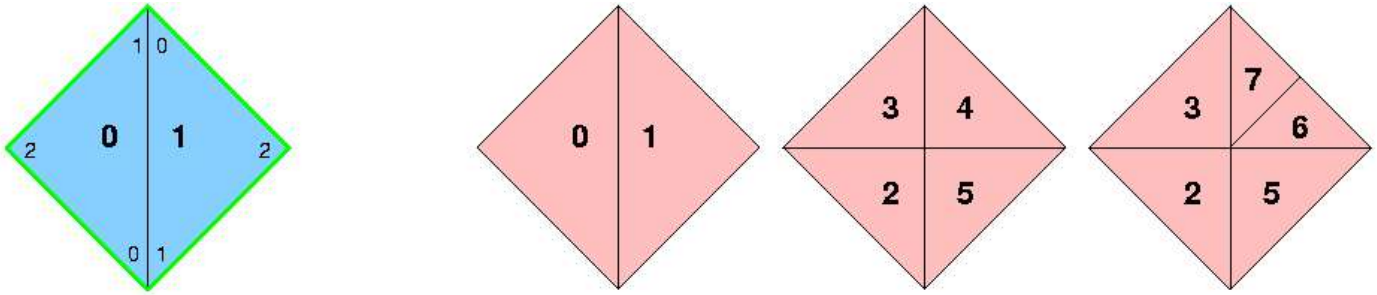
Bisectioning of simplices:

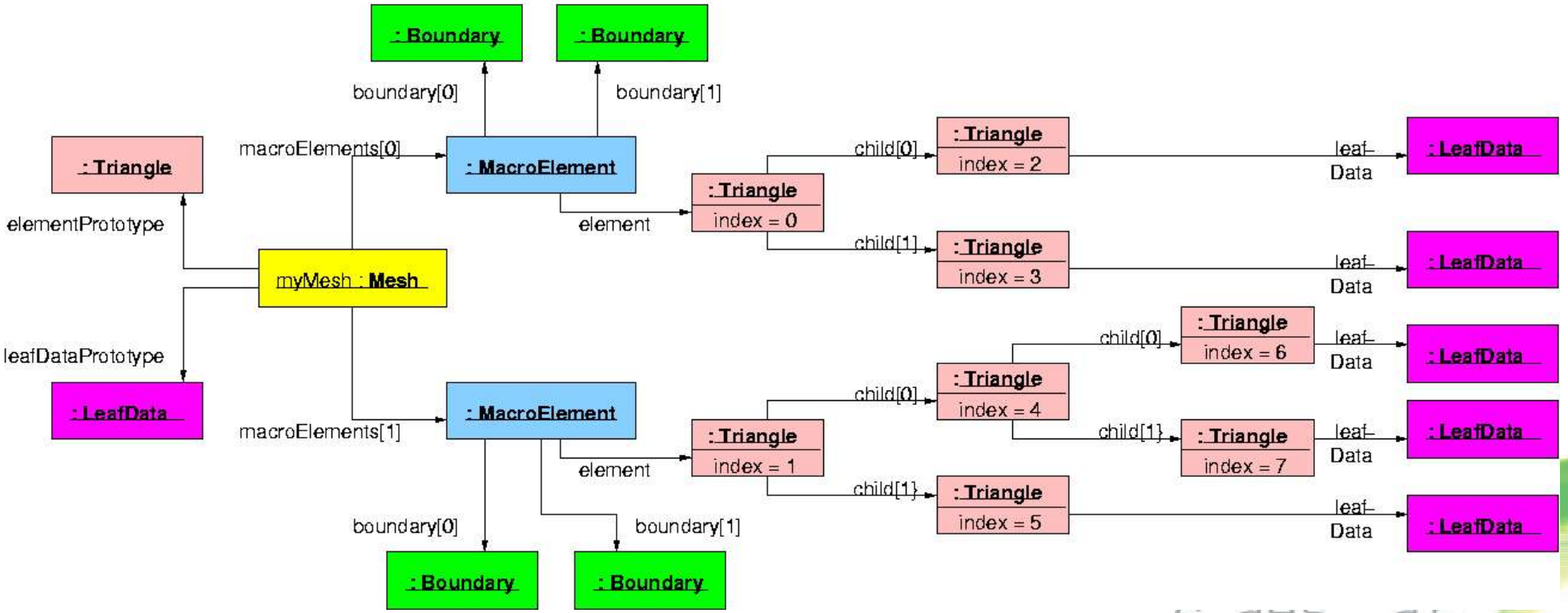




Hierarchical mesh structure

Example instance diagram:





Classical adaptive Domain Decomposition

- DD tries to balance the load for **current mesh**
- Each processor computes **on it's domain only**

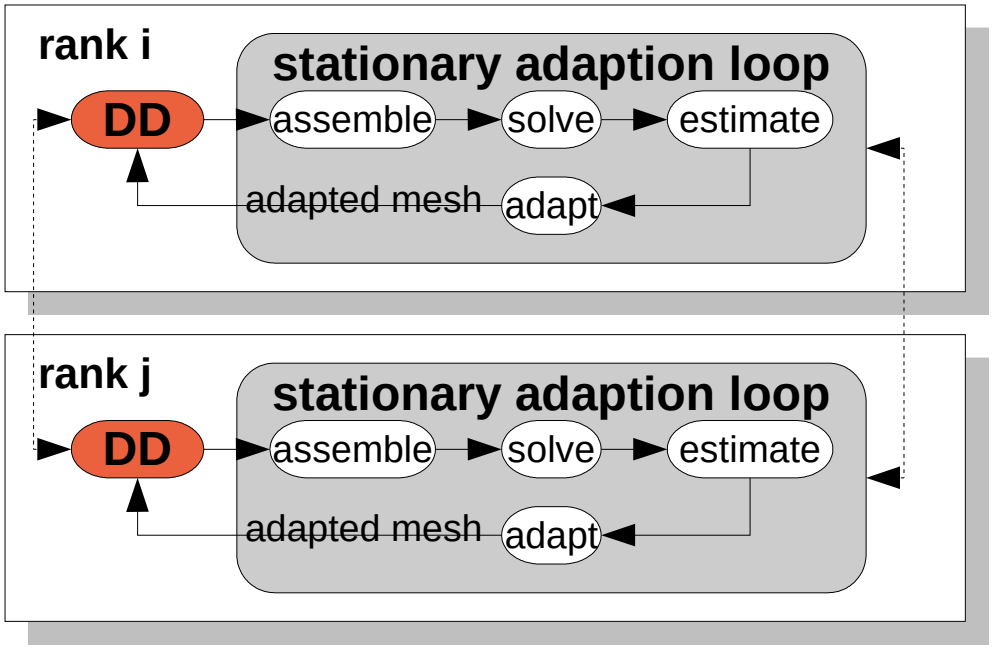
=> Communication at domain boundaries

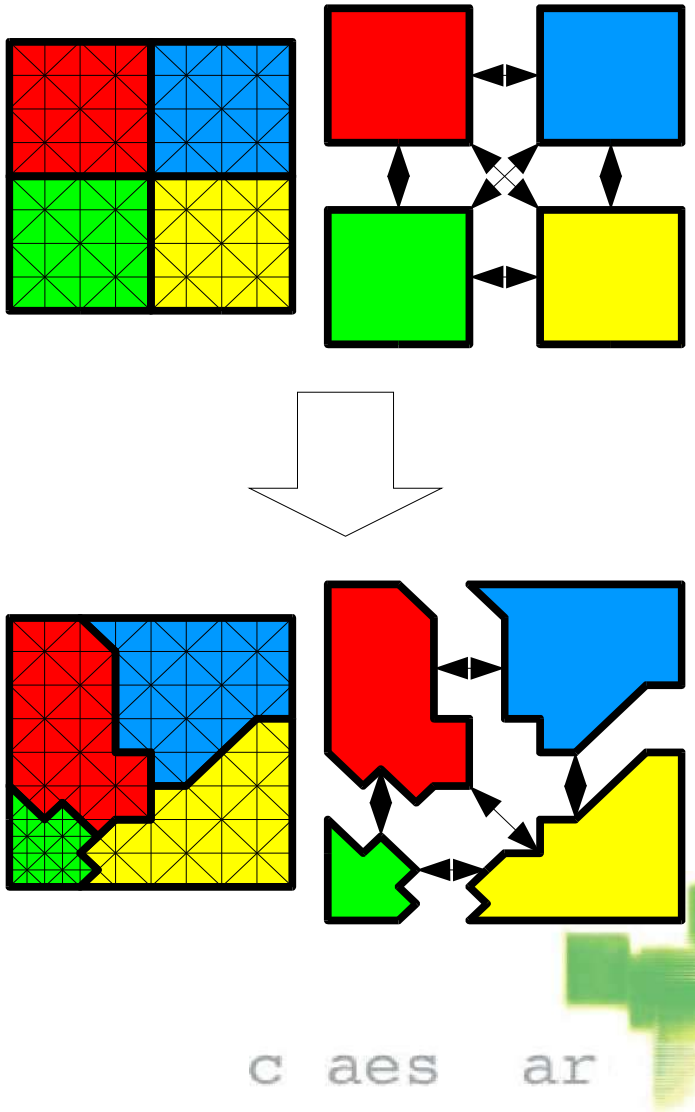
=> Adaption of many software components

- After mesh adaption -> new load balanced DD!

=> Communication needed

  - to calculate DD
  - to redistribute mesh elements





http://www.caesar.de/amdis.html

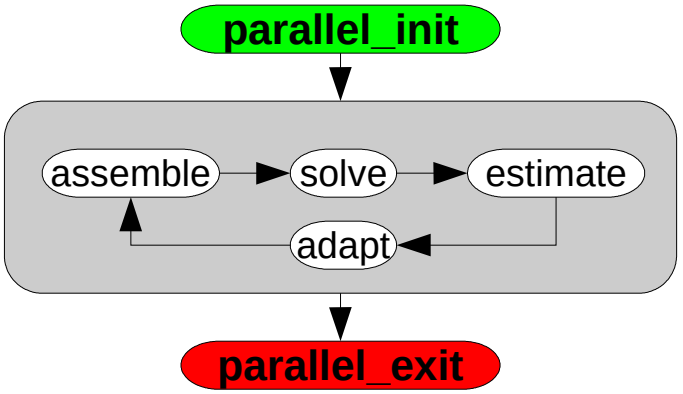
crystal growth group

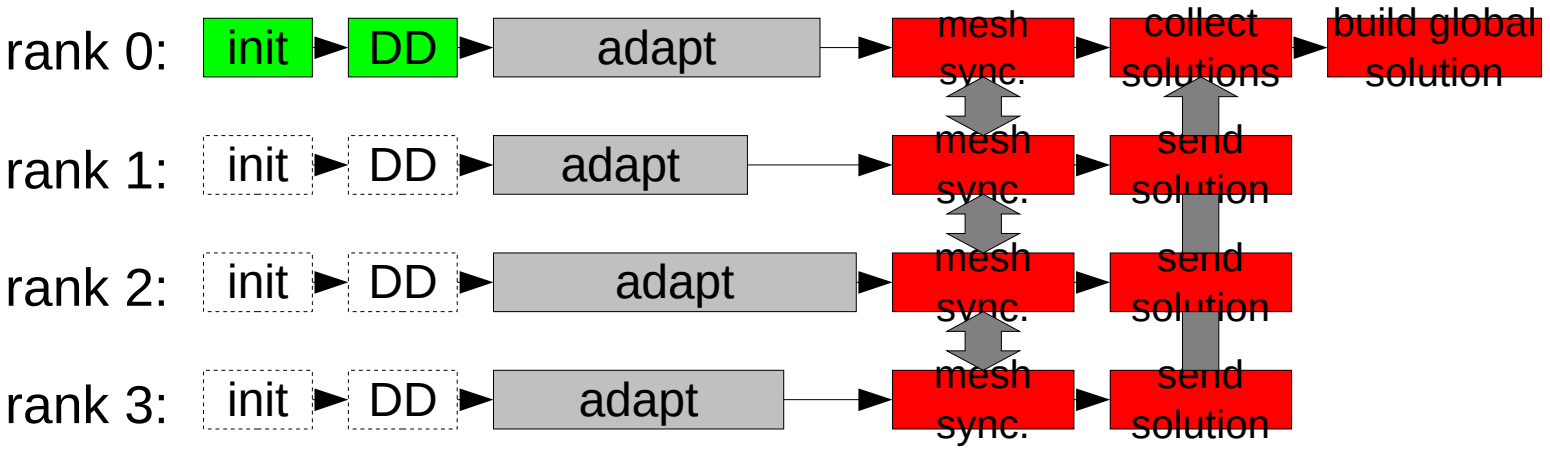
Parallel adaptive meshing in AMDiS

(1) • Solve problem on rel. coarse mesh.
• Estimate element errors.
• Create partitions with approx. equal error.

(2) • Each processor solves problem on whole domain.
• Refinement allowed only in own partition.

(3) • Create composite mesh.
• Create global solution.





Literature: *"A new paradigm for parallel adaptive meshing algorithms"*, R. E. Bank and M. Holst, SIAM Rev. 2003

Error-weighted spectral bisectioning

Problem: S – set of mesh elements, $n = |S|$, w_1, w_2 - weights

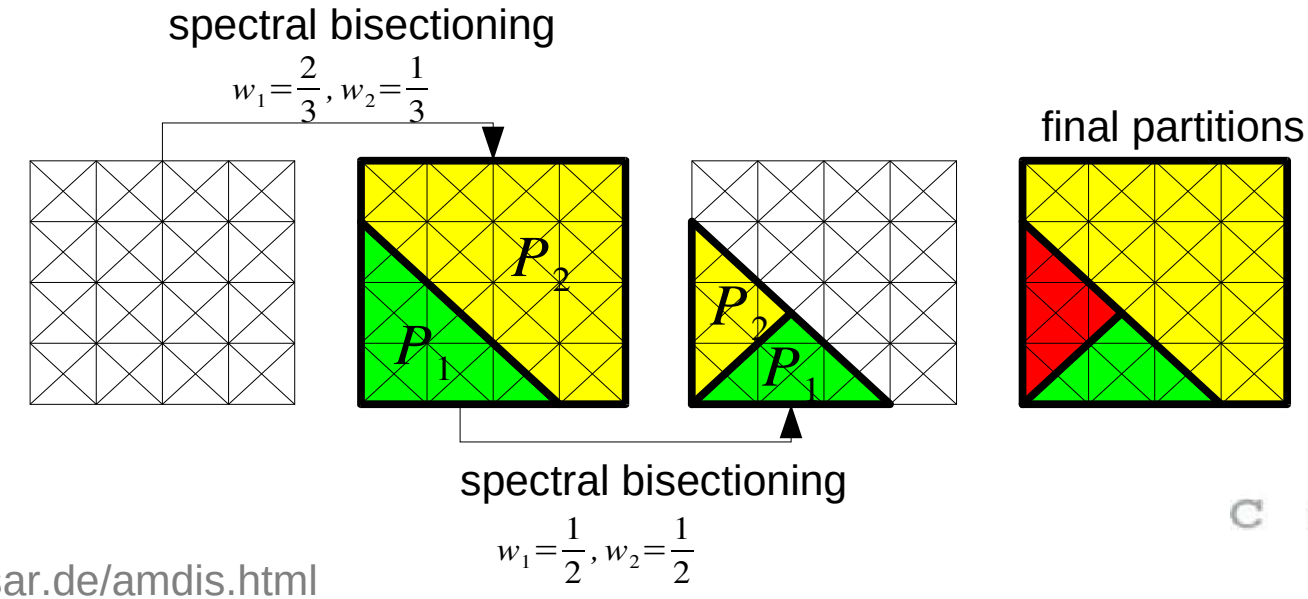
Find sets P_1 and P_2 s.t. (a) $P_1 \cup P_2 = S \quad \wedge \quad P_1 \cap P_2 = \emptyset$

(b) $|C|$ is minimized with
$C = \{(\sigma_i, \sigma_j) \mid \sigma_i$ and σ_j are neighbour elements and belong to different partitions$\}$

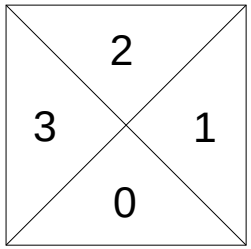
(c) P_1 and P_2 are connected

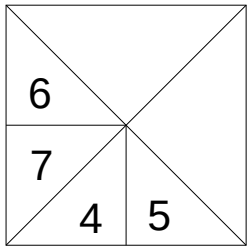
(d) $w_1 \cdot error(P_1) \approx w_2 \cdot error(P_2)$

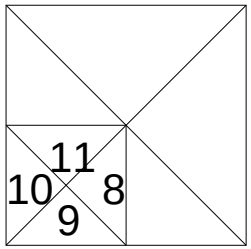
Example for 3 partitions:

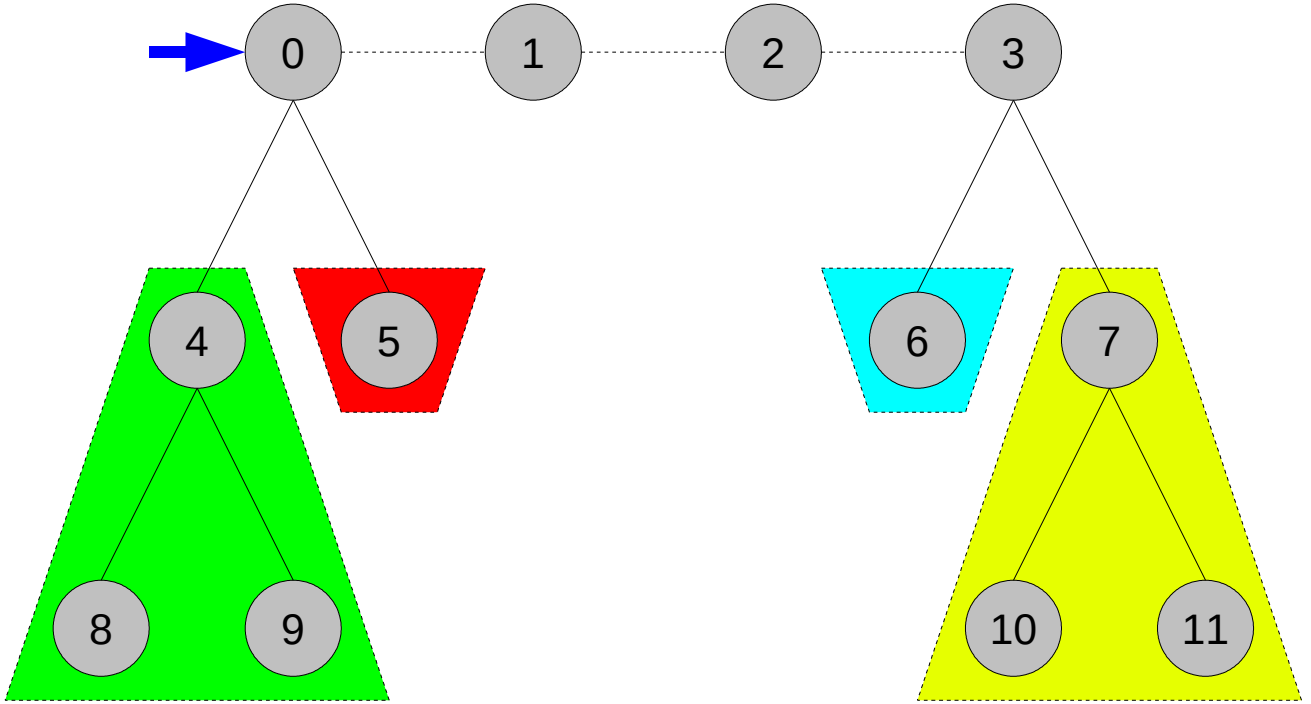


Mesh structure code









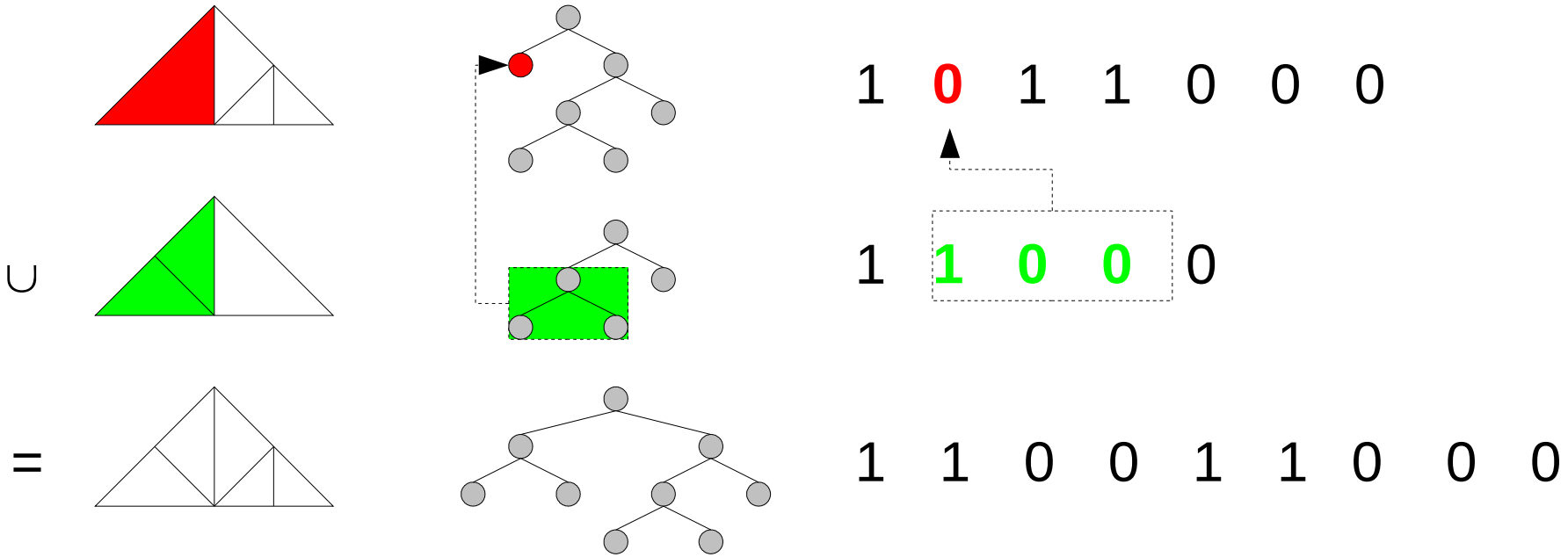
Pre-order traverse: 0 4 8 9 5 1 2 3 6 7 10 11

Has children? 1 1 0 0 0 0 0 1 0 1 0 0

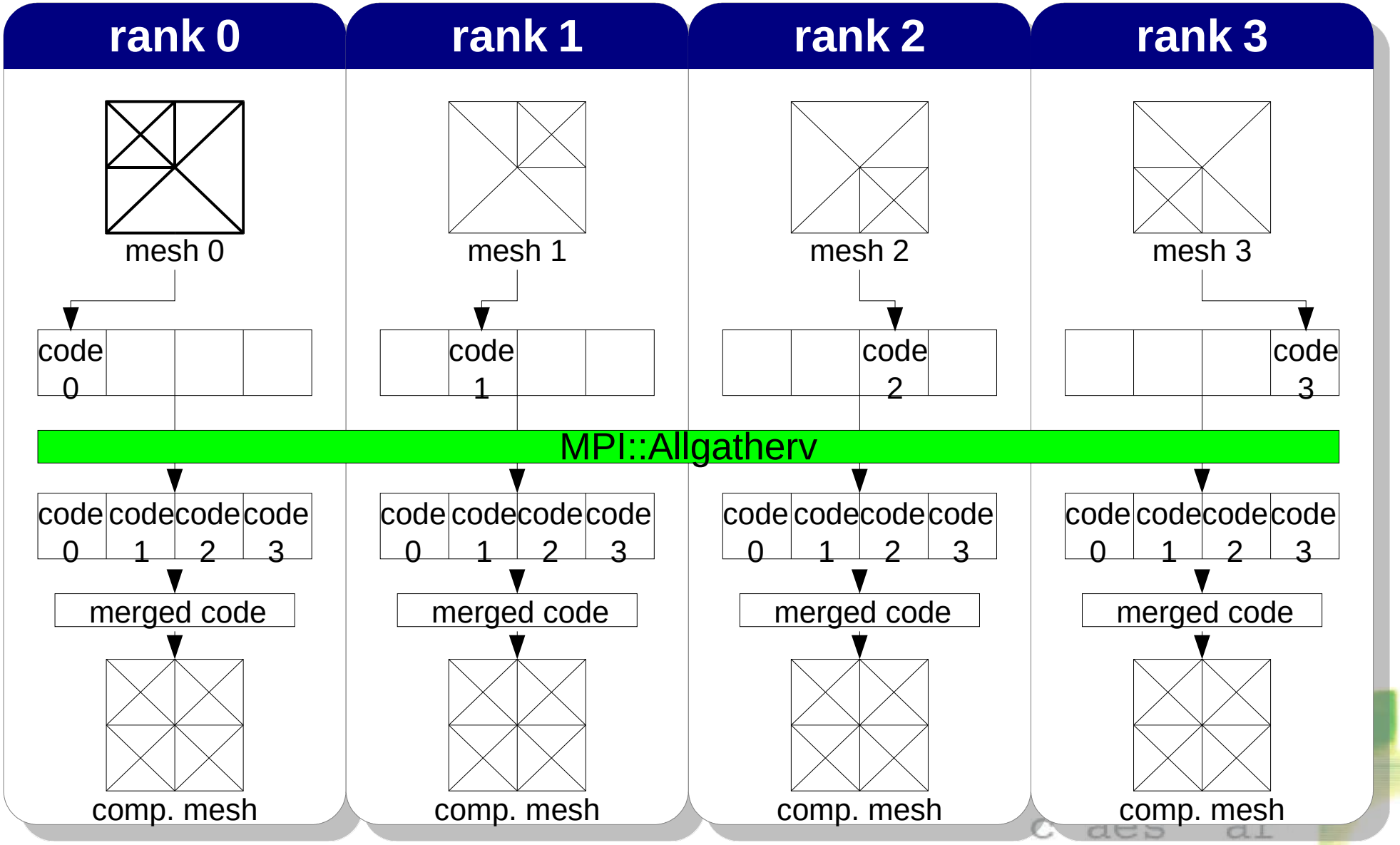
decimal: 24 0 0 20

Mesh synchronization

Merging codes:



Mesh synchronization (2)



Partition of Unity

Building global solution by Partition of Unity method:

$\{\Omega_i\}$: open covering of Ω

u_i : approximate solution on local domain Ω_i

ϕ_i : continuous partition of unity subordinate to the covering $\{\Omega_i\}$

$$\begin{aligned} \phi_i(x) &\geq 0 && \forall\, x \in \Omega \\ \phi_i(x) &= 0 && \forall\, x \notin \Omega_i \\ \sum \phi_i(x) &= 1 && \forall\, x \in \Omega \end{aligned}$$

Global solution:

$$u_G(x) = \sum \phi_i(x) u_i(x)$$

Partition of Unity (2)

Theorem:

h_i : edge size of mesh i on Ω_i $\qquad h := max\, h_i$

H_i : edge size of mesh i on $\Omega \setminus \Omega_i$ $\qquad H := max\, H_i$

Assume $u \in H^2(\Omega)$, then

$$\|u - u_G\|_{H^1} \leq C(h + H^2)$$

In particular, if $H \leq \sqrt{h}$ then

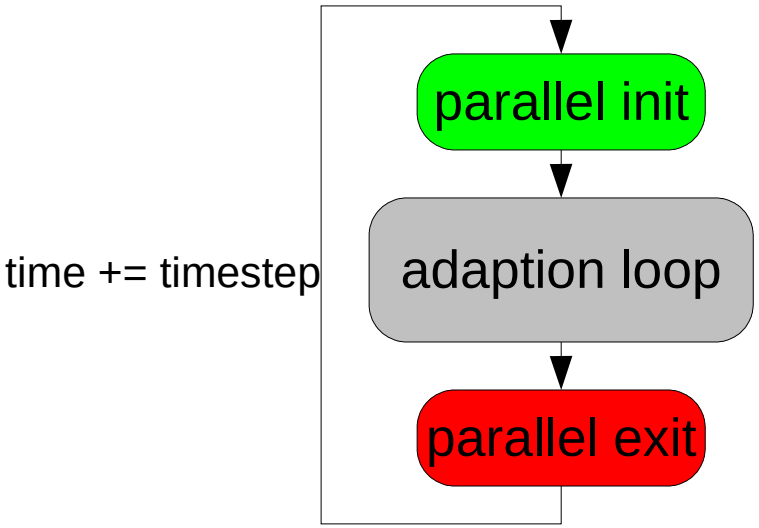
$$\|u - u_G\|_{H^1} \leq C(h)$$

Time dependent problems

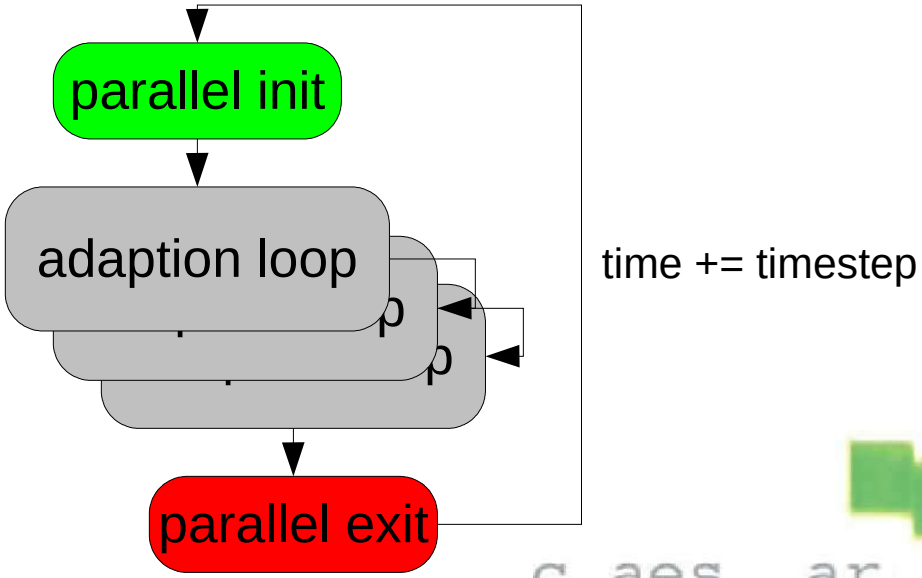
Repartitioning:

(1) Store composite mesh structure code of last timestep
(2) Go back to macro triangulation
(3) Do initial iterations for the new timestep
(4) Compute new DD
(5) Refine processors partition according to structure code

after every timestep:

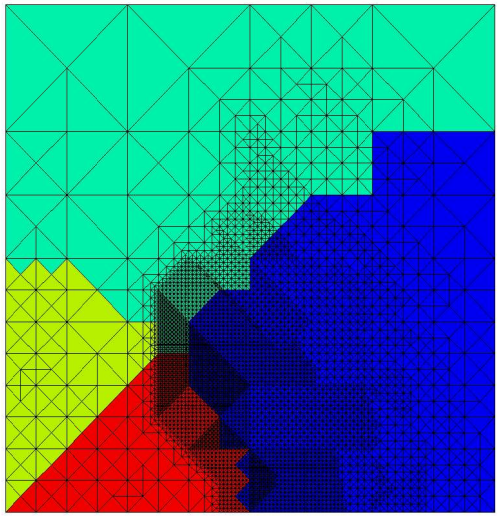


after every i-th timestep:



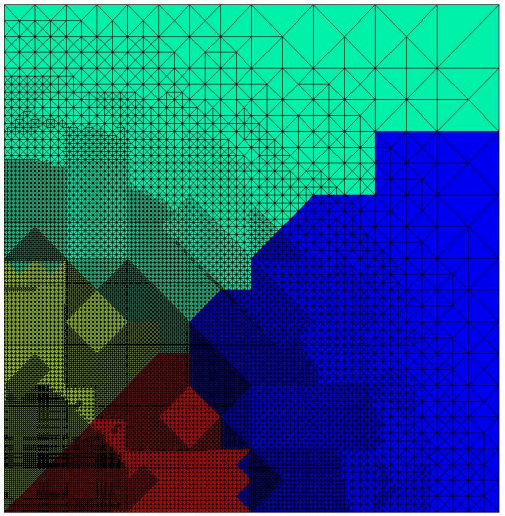
Example repartitioning

final mesh rank 0, timestep 1

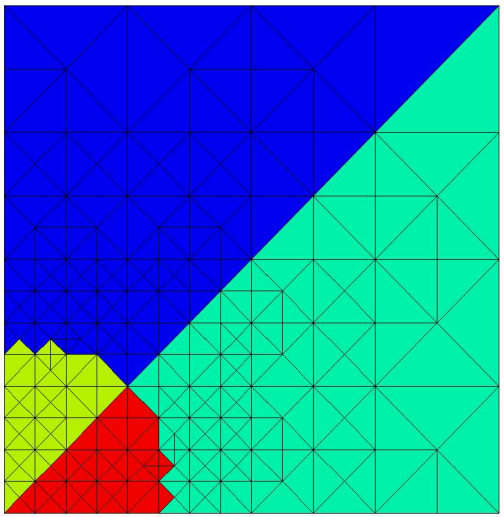


mesh sync.

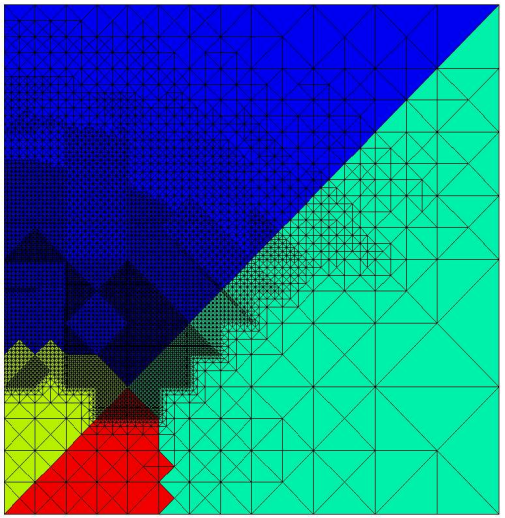
composite mesh timestep 1



repartitioning



refine



DD timestep 2

start mesh rank 0, timestep 2

Example problem

Example problem

Poisson equation:

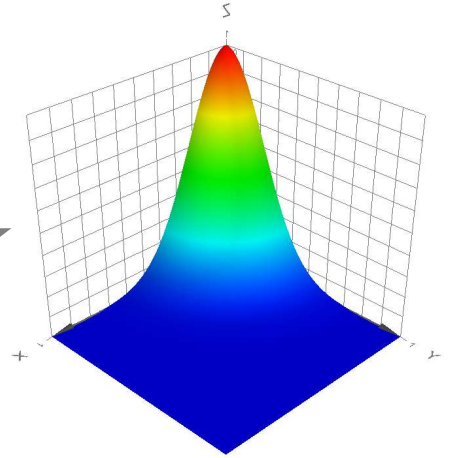
$$-\Delta u(x) = f(x) \quad x \in \Omega$$
$$u(x) = g(x) \quad x \in \partial\Omega$$

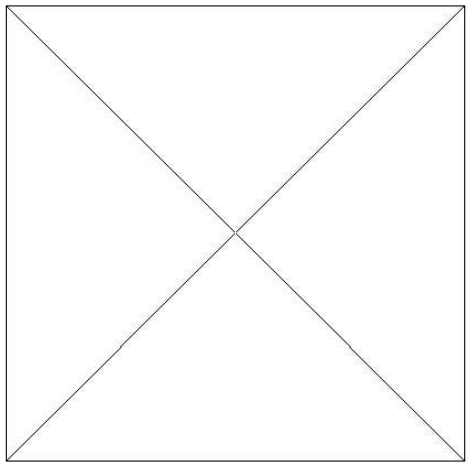
$$f(x) = -(400\,x^2 - 20 \cdot dim)\,e^{-10x^2} \qquad g(x) = e^{-10x^2}$$

Solution:

$$u(x) = e^{-10x^2}$$

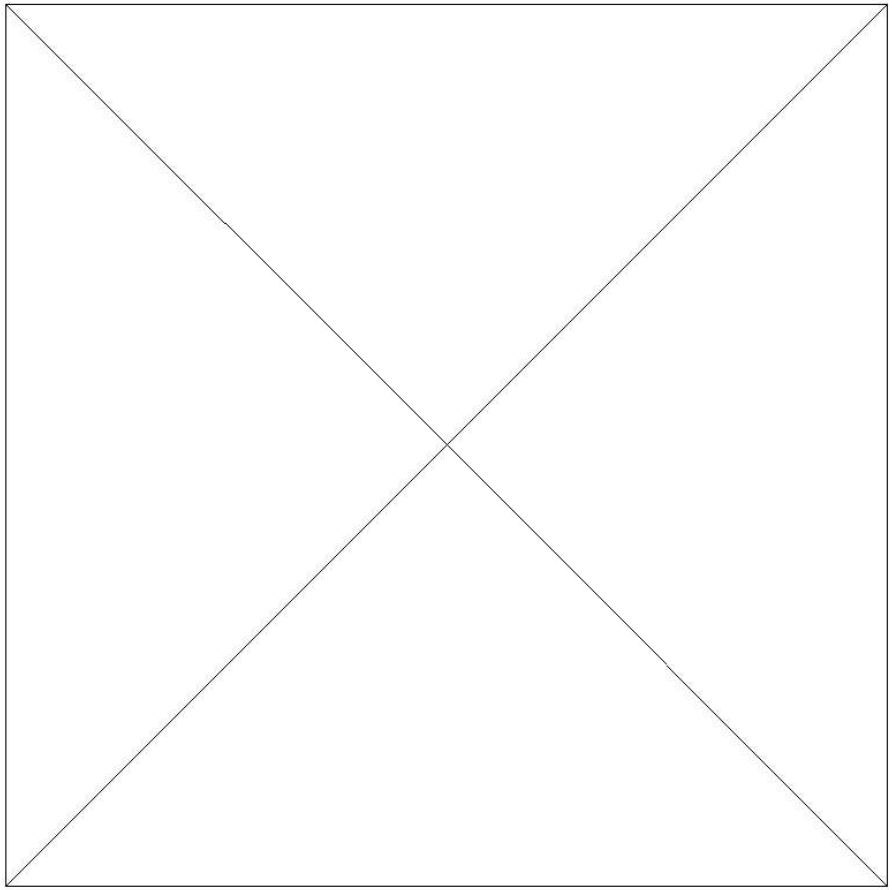
solution for dim = 2



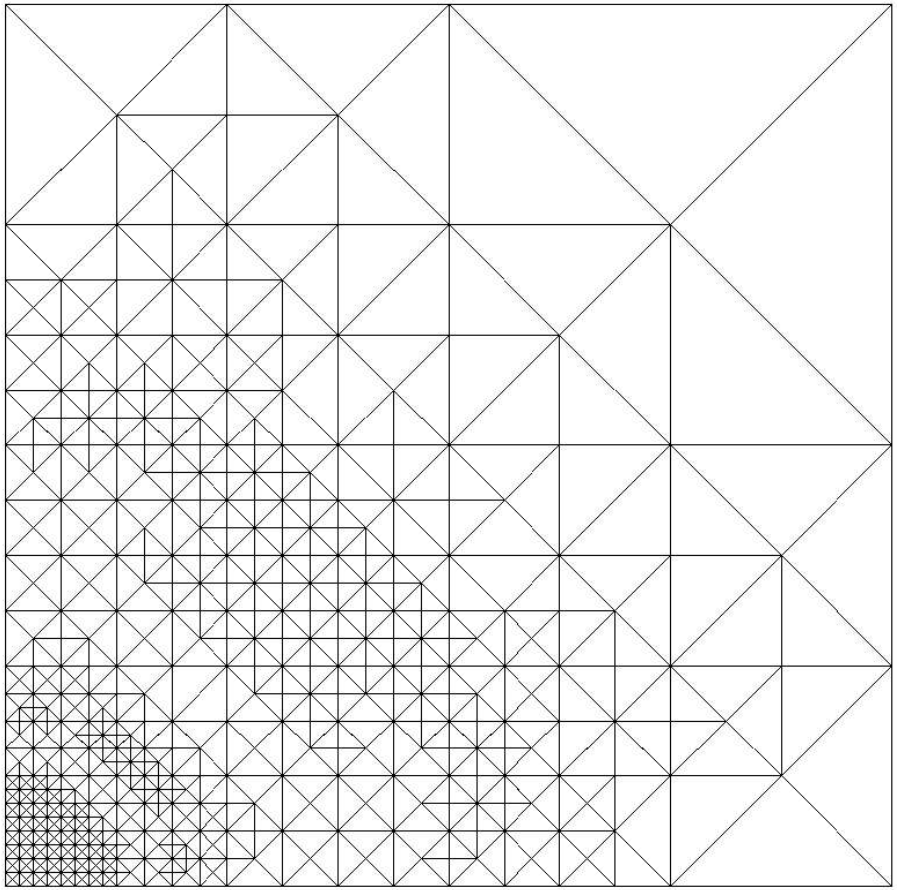


macro triangulation

Partitioning

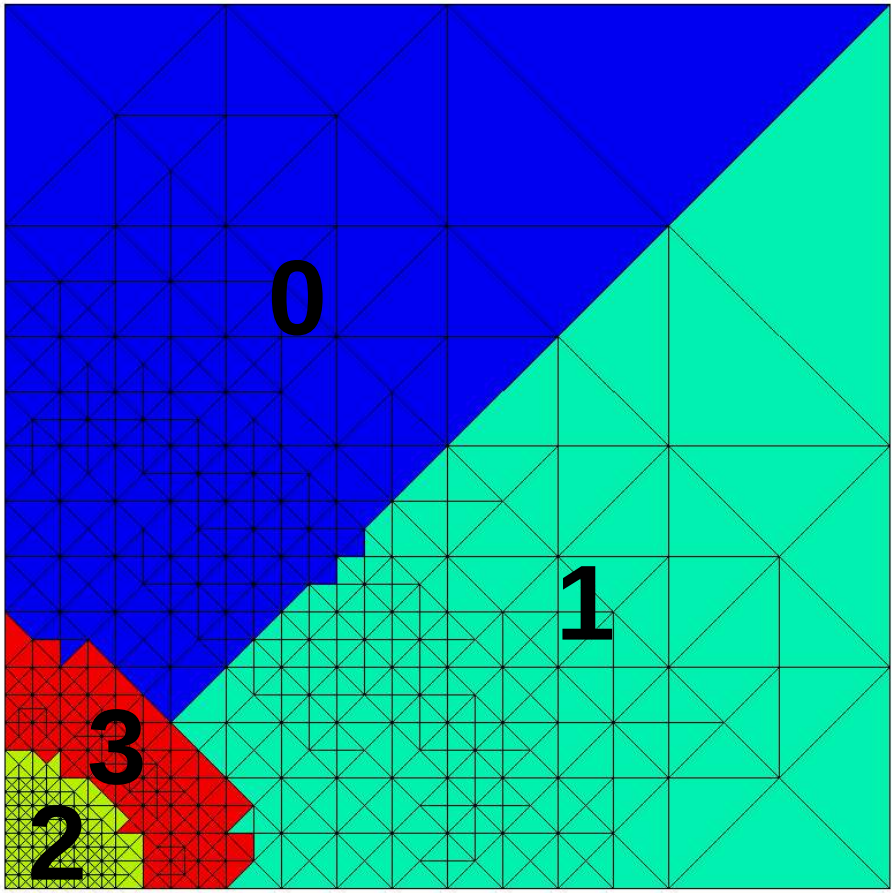


macro triangulation

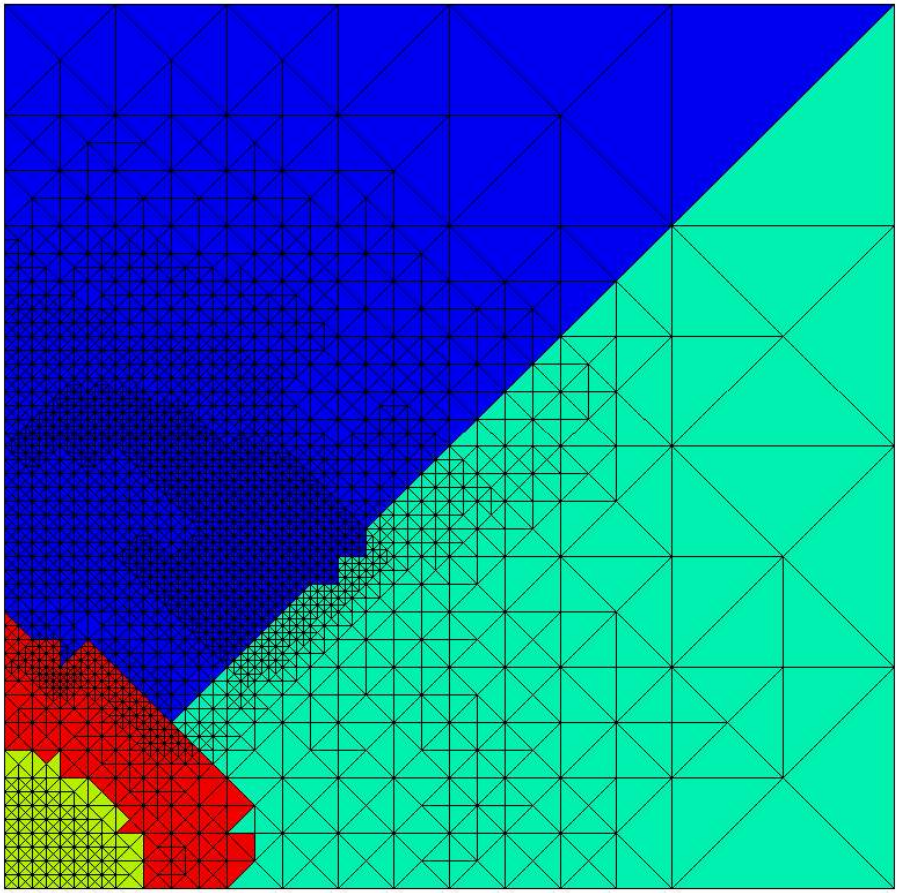


mesh after 6 initial iterations

Partitioning

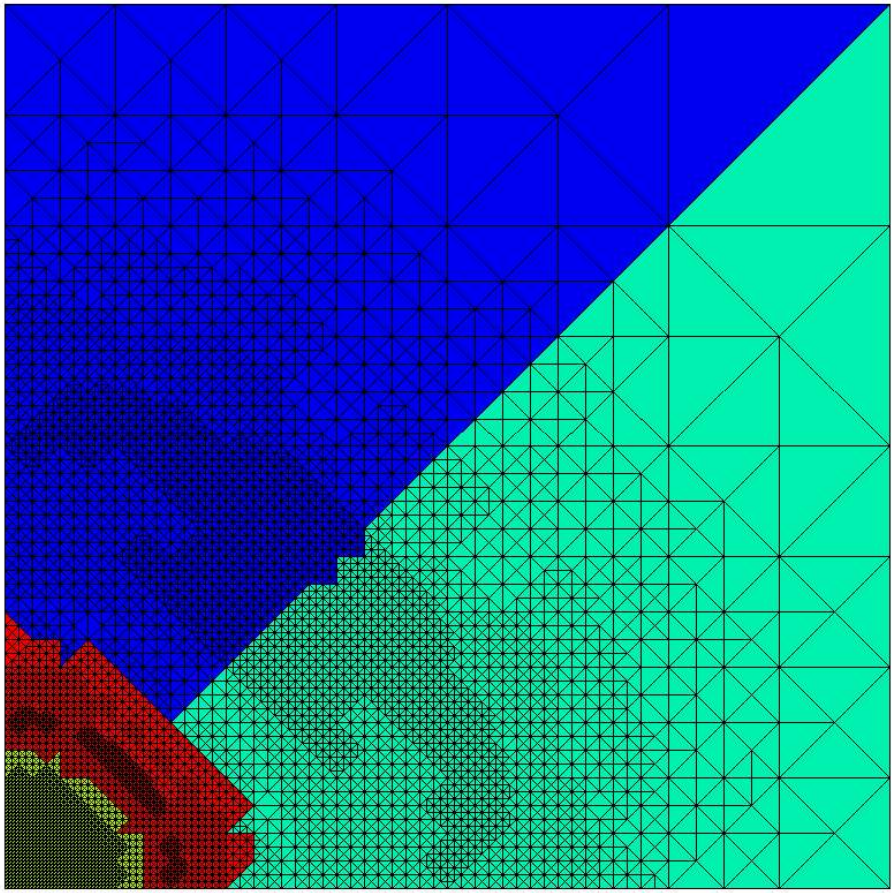


partitioning after 6 initial iterations

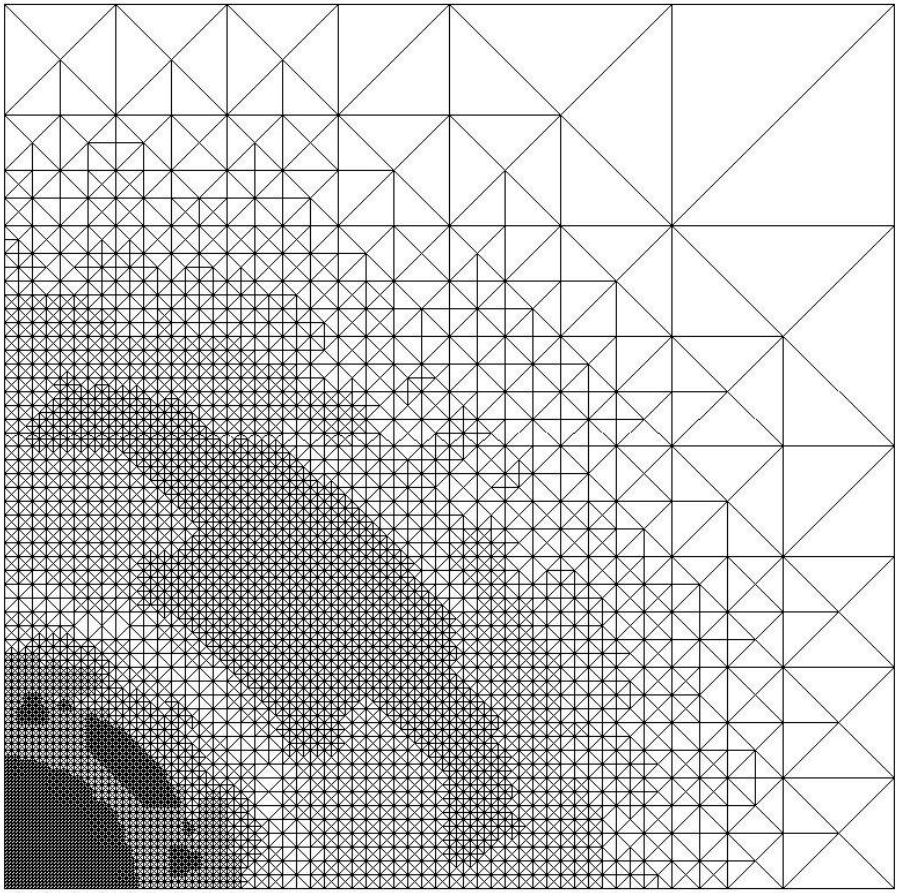


final rank0 mesh

Composite mesh and corresonding sequential mesh



after mesh synchronization

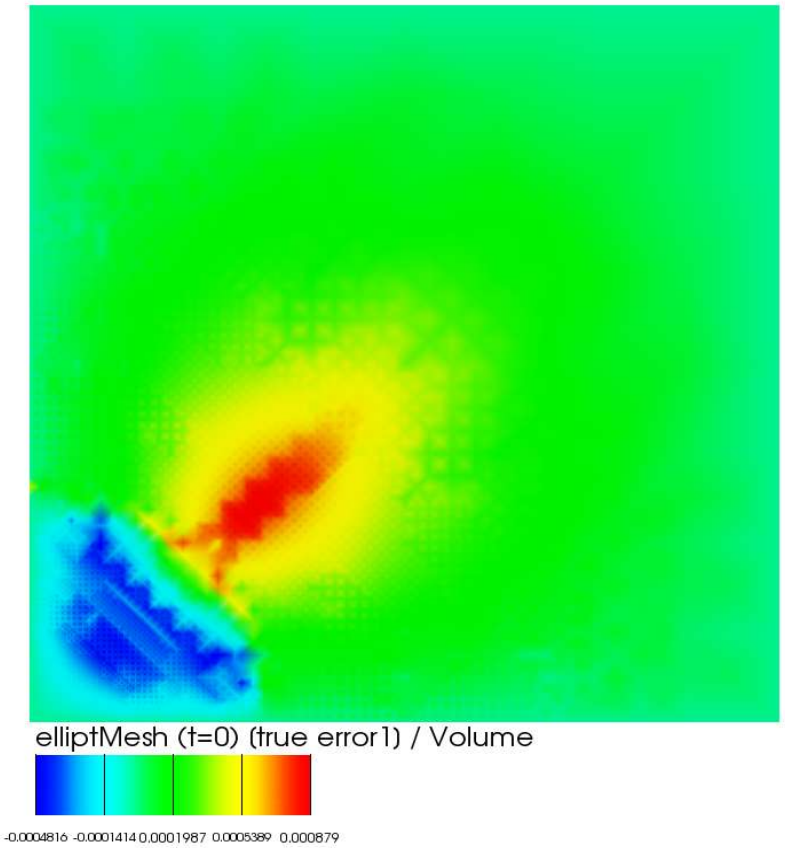


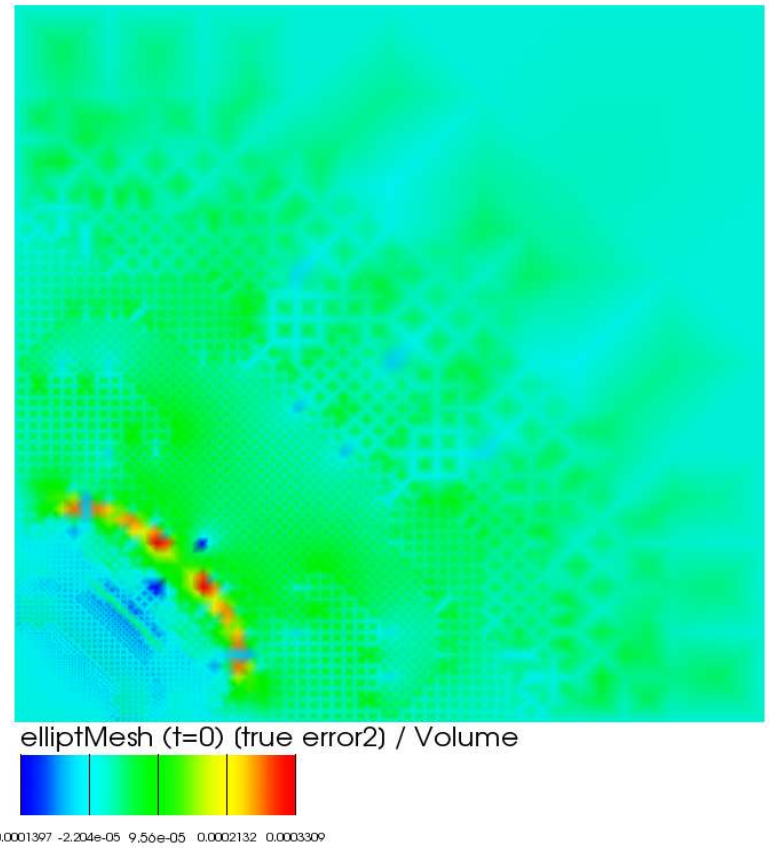
sequential mesh

c aes ar

crystal growth group

Pointwise error





after PU

total error: $2.67 \cdot 10^{-2}$

after final solve

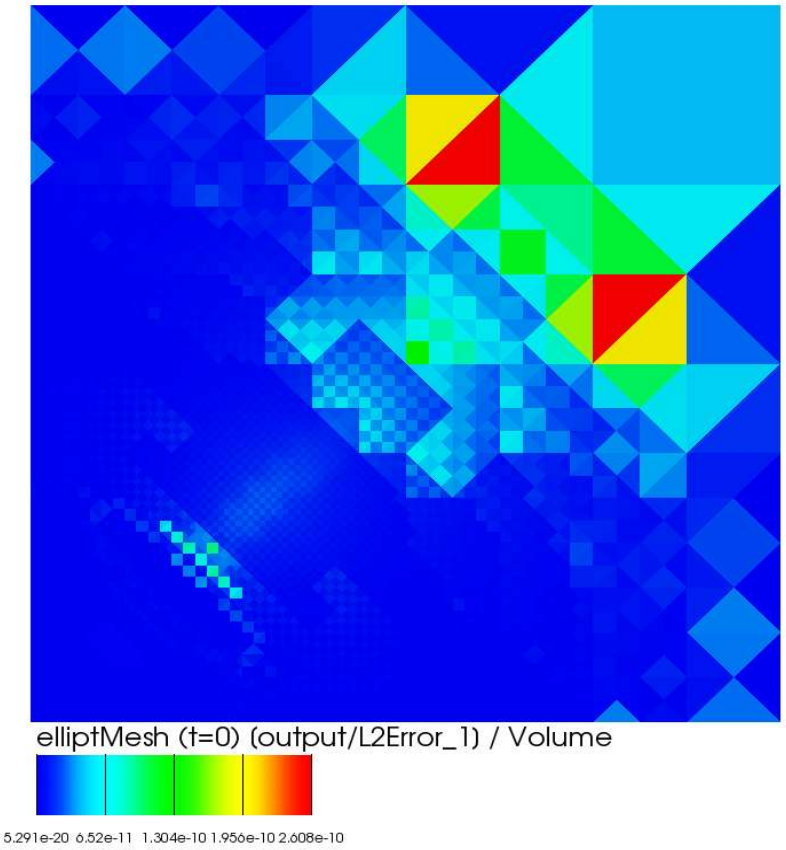
$3.69 \cdot 10^{-3}$

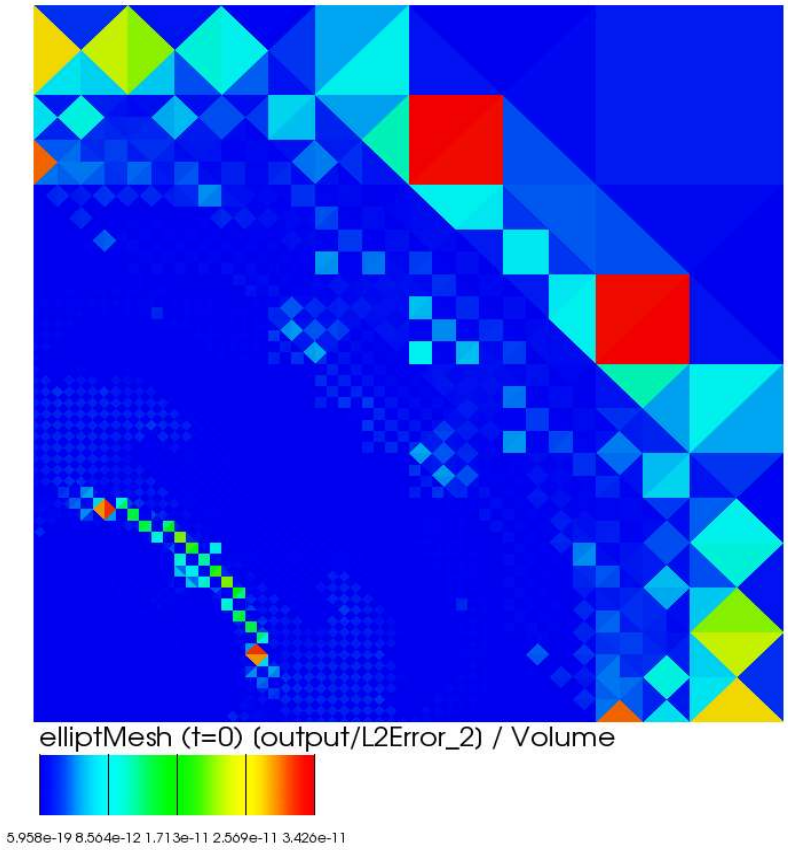
c aes ar

crystal growth group

http://www.caesar.de/amdis.html

L2 error





after PU

total error: $2.67 \cdot 10^{-4}$

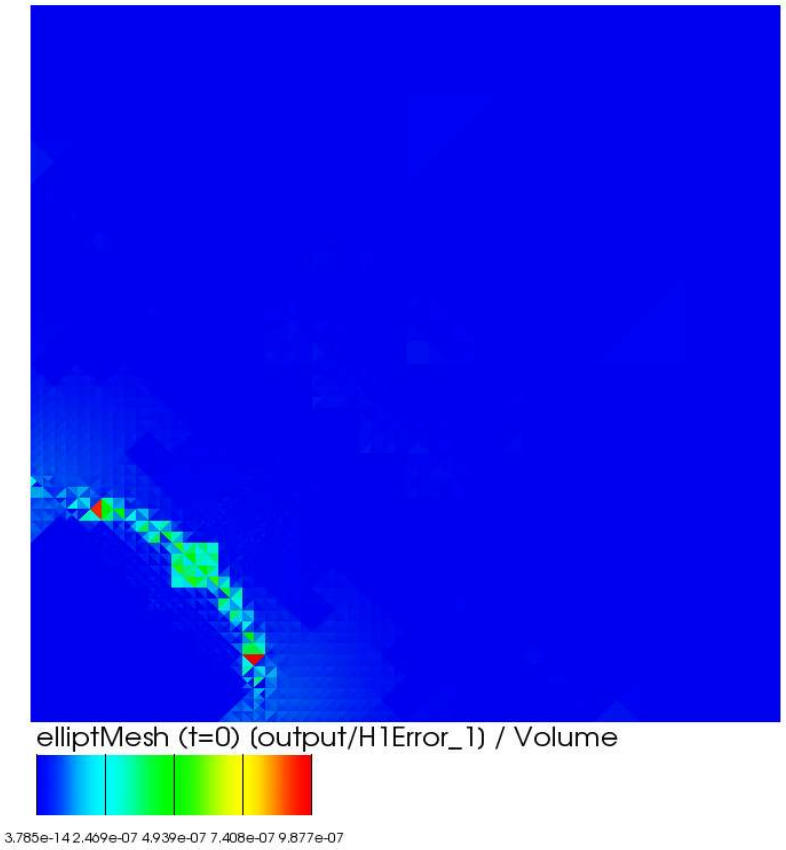
after final solve

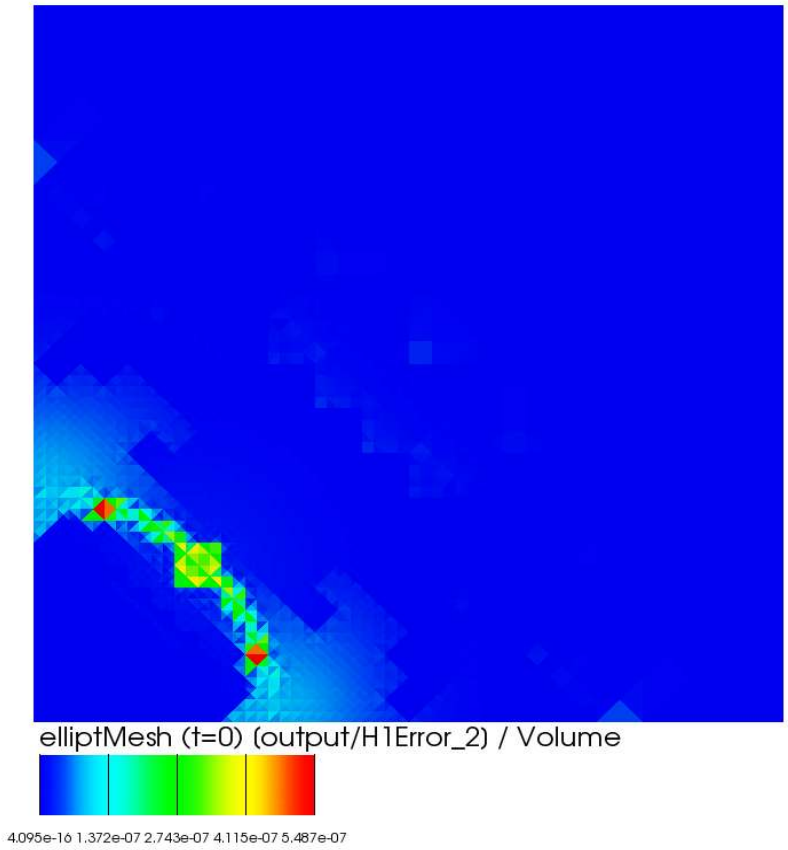
$7.01 \cdot 10^{-5}$

c aes ar

crystal growth group

H1 error





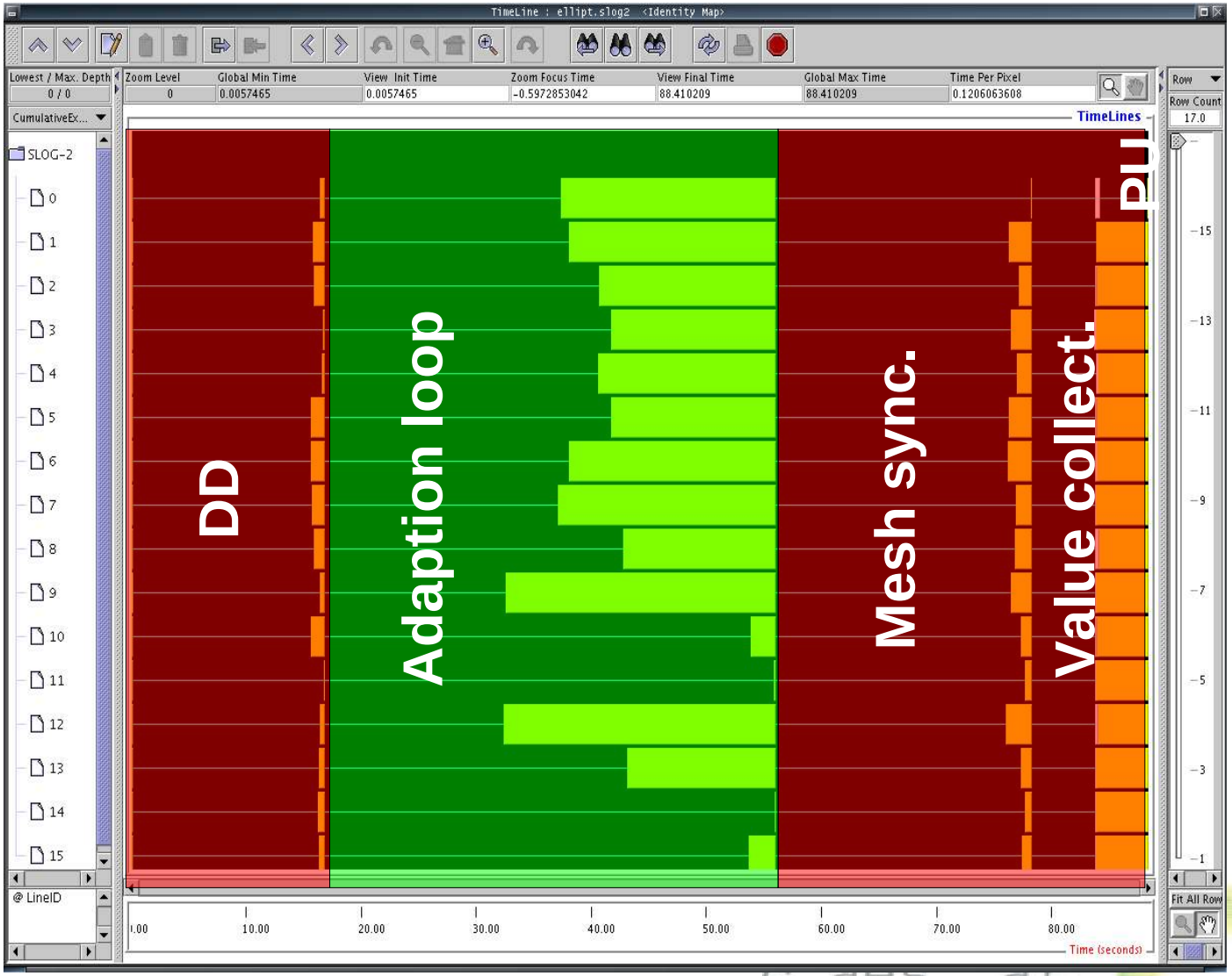
	after PU	after final solve
total error:	$1.19 \cdot 10^{-2}$	$1.15 \cdot 10^{-2}$

c aes ar

crystal growth group

Speed-up and load balance

#procs	Time [sec]	Speed-Up
1	358	1
2	235	1.52
4	147	2.44
8	120	2.98



"sequential part"

"parallel part"

Conclusion

- Parallel adaptive meshing succesfully implemented in AMDiS
- Partitioning by error-weighted spectral bisectioning
- Global solution built by Partition of Unity
- Time dependent problems manageable

- Speed-up and load balance not yet optimal

Further work:

- Performance improvement
- Goal-oriented error estimates