Large-scale embedded domain simulations by means of the AgFEM method

Francesc Verdugo

CIV-PRESTO, Toulouse, FR, 2020-09-23

CIMNE

LSSC

Prof. S. Badia

Dr. A.F. Martín

MSc. E. Neiva

Dr. F. Verdugo

MONASH University

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

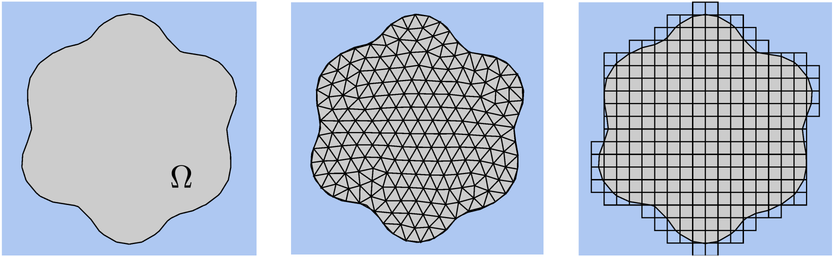
CIMNE

EXCELENCIA SEVERO OCHOA

What are Embedded Finite Element methods?

Embedded Finite elements

CutFEM, Finite Cell Method, AgFEM, X-FEM, ...



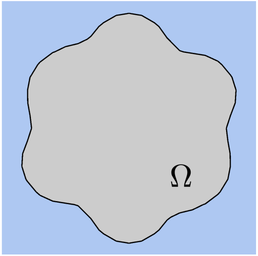
body-fitted mesh

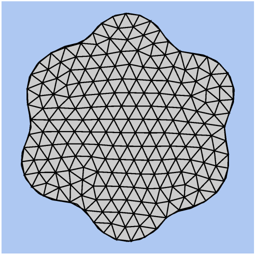
unfitted mesh

✔ Simplified mesh generation

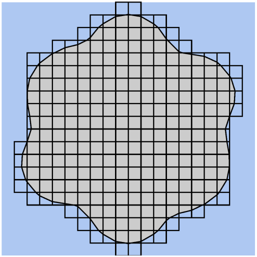
Embedded Finite elements

CutFEM, Finite Cell Method, AgFEM, X-FEM, ...





body-fitted mesh



unfitted mesh

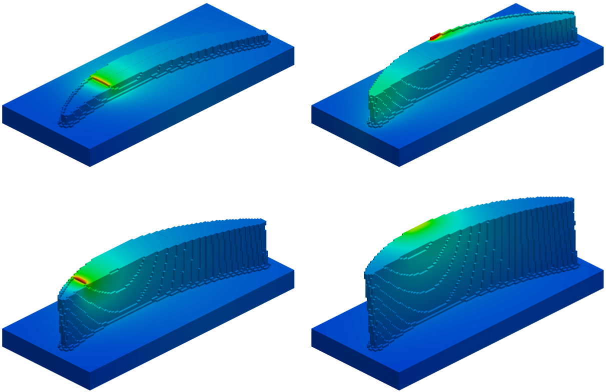
✔ Simplified mesh generation

✘ Dirichlet BC? ✘ Numerical integration? ✘ ill-conditioning? (this talk)

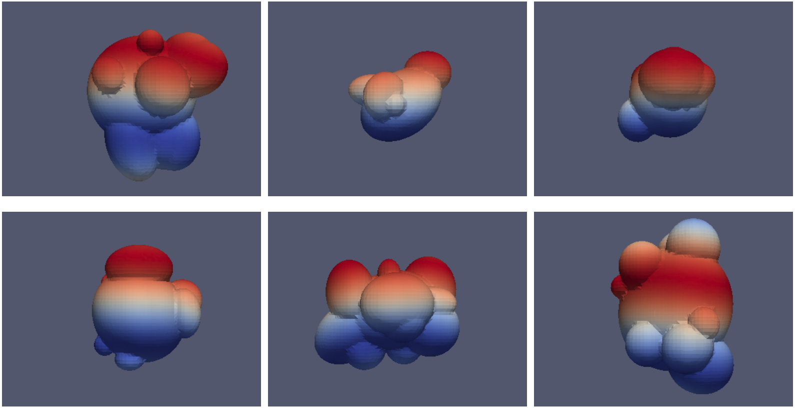
3D printing simulation

EMUSIC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690725



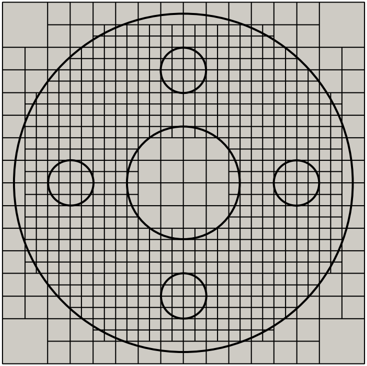
UQ with stochastic geometries



Exascale Quantification of Uncertainties for Technology and Science Simulation

Distributed simulation pipeline

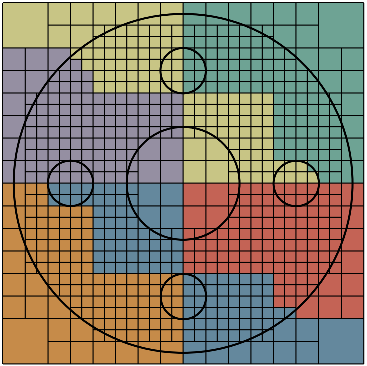
1. Unfitted (adaptive) Cartesian grids (p4est)



Distributed simulation pipeline

1. Unfitted (adaptive) Cartesian grids (p4est)

2. Partition using space filling-curves (p4est)



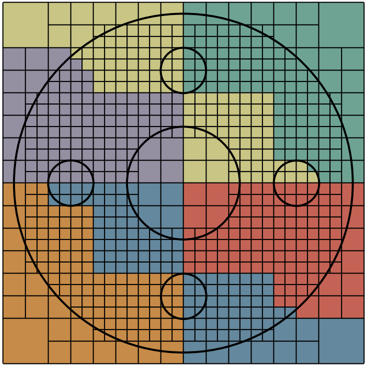
Distributed simulation pipeline

1. Unfitted (adaptive) Cartesian grids (p4est)

2. Partition using space filling-curves (p4est)

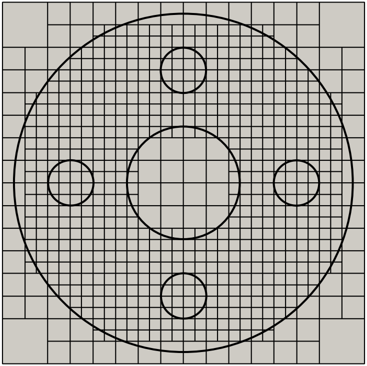
3. Embedded FEM (AgFEM)

4. AMG linear solver (petsc)



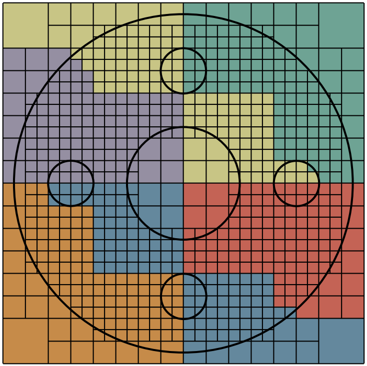
Unfitted methods at large scales: pros and cons

✔ Highly scalable mesh generation based on octrees (e.g. `p4est`)



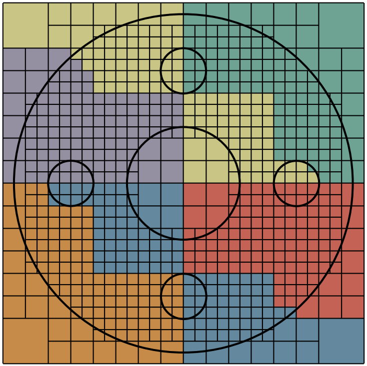
Unfitted methods at large scales: pros and cons

- ✔ Highly scalable mesh generation based on octrees (e.g. `p4est`)
- ✔ Highly scalable mesh partition with space-filling curves (Parmetis not needed)



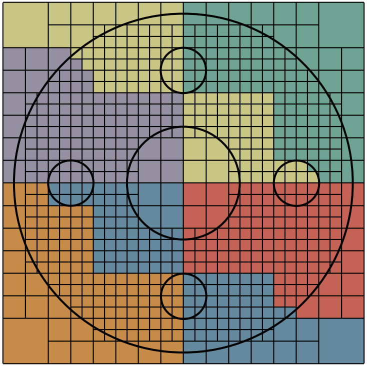
Unfitted methods at large scales: pros and cons

- ✔ Highly scalable mesh generation based on octrees (e.g. `p4est`)
- ✔ Highly scalable mesh partition with space-filling curves (Parmetis not needed)
- ✔ Highly scalable adaptive mesh refinement + load balancing



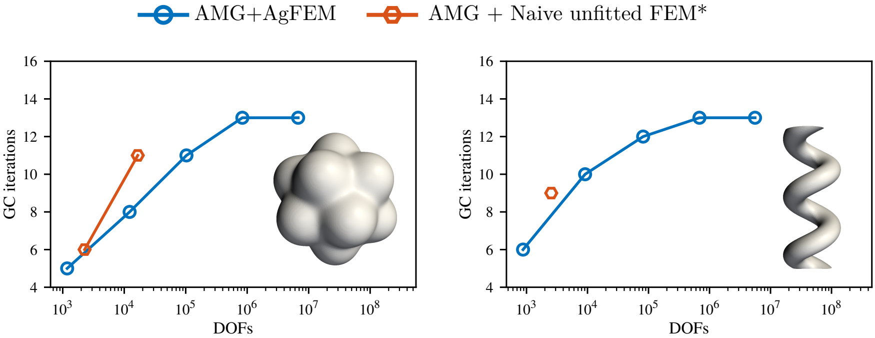
Unfitted methods at large scales: pros and cons

- ✔ Highly scalable mesh generation based on octrees (e.g. `p4est`)
- ✔ Highly scalable mesh partition with space-filling curves (Parmetis not needed)
- ✔ Highly scalable adaptive mesh refinement + load balancing
- ✘ Not guaranteed that highly scalable linear solvers keep their optimal properties for cut elements.



Petsc CG + AMG preconditioner on unfitted meshes

Poisson equation (weak scaling test with 5 meshes)

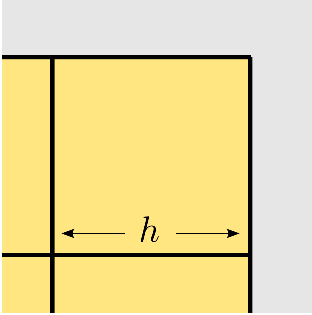


* Nitsche BCs + modified integration in cut cells

Why linear solvers are affected by cut cells?

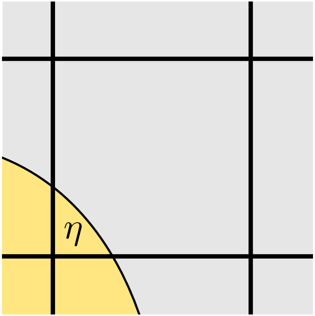
Condition number estimates (Poisson Eq.)

(a) Body-fitted case



$k_2(A) \sim h^{-2}$

(b) Naive unfitted FEM



$k_2(A) \sim |\eta|^{-(2p+1-2/d)}$

"small cut cell problem"

Possible remedy: Fix the linear solver [1]

Embedded FEM

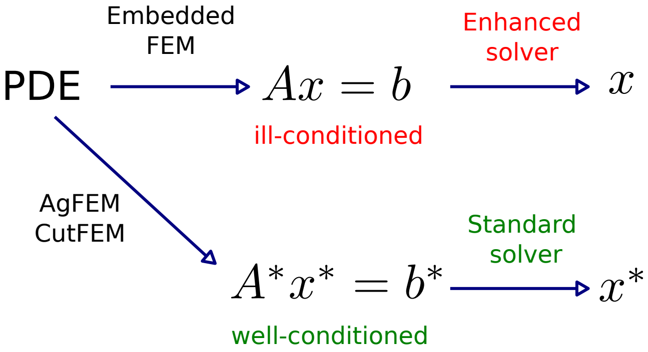
Enhanced solver

PDE $\longrightarrow$ $Ax = b$ $\longrightarrow$ x

ill-conditioned

[1] [S. Badia, F. Verdugo. Robust and scalable domain decomposition solvers for unfitted finite element methods. *Journal of Computational and Applied Mathematics* (2018)]

Possible remedy: Enhanced FE formulation



Agenda

1. The AgFEM method (serial case)

2. Parallel implementation

3. Performance of parallel AgFEM + AMG solvers

Agenda

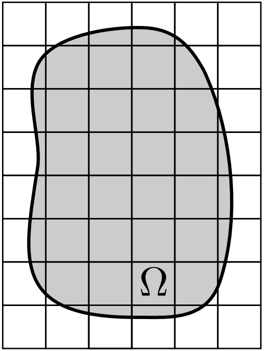
1. The AgFEM method (serial case)

2. Parallel implementation

3. Performance of parallel AgFEM + AMG solvers

AgFEM method for the Poisson Eq.

$$\left.\begin{aligned} -\Delta u &= f \quad \text{in} \quad \Omega \\ u &= u^{\mathrm{D}} \quad \text{on} \quad \partial\Omega \end{aligned}\right\}$$



Dirichlet BCs

Nitsche's Method

Find $u^h \in V^h$ such that

$$a(v, u^h) = l(v) \quad \forall v \in V^h \text{ (} V_h \text{ does not vanish on } \partial\Omega\text{)}$$

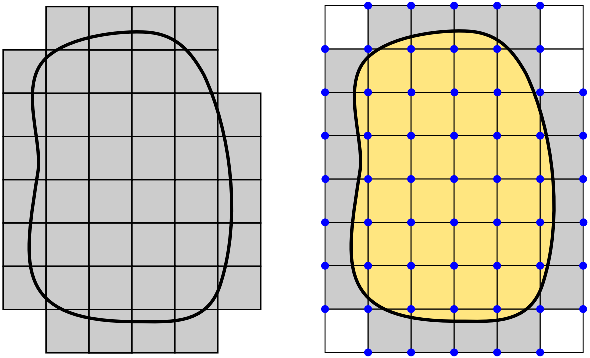
where

$$a(v,u) := \int_\Omega \nabla v \cdot \nabla u - \int_{\partial\Omega} (v \nabla u) \cdot n + \int_{\partial\Omega} \beta\, v\, u - \int_{\partial\Omega} (u \nabla v) \cdot n$$

$$l(v) := \int_\Omega v\, f + \int_{\partial\Omega} \beta v u^D - \int_{\partial\Omega} (u^{\mathrm{D}} \nabla v) \cdot n$$

Starting point: "naive" FE space

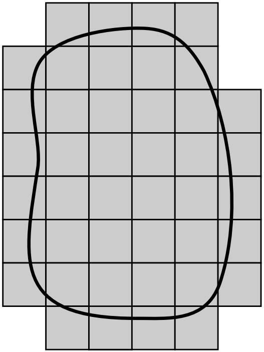
$$V_h^{\mathrm{std}} := \{u \in C^0(\Omega^{\mathrm{act}}) \ : \ u|_K \in Q^p(K) \; \forall K \in \mathcal{T}^{\mathrm{act}}\}$$



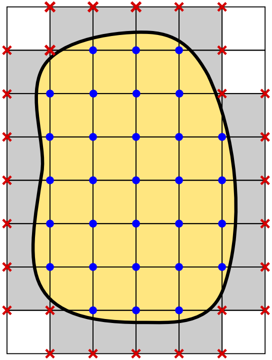
$\mathcal{T}^{\mathrm{act}}, \ \Omega^{\mathrm{act}}$ $\qquad$ V_h^{std}

Starting point: "naive" FE space

$$V_h^{\mathrm{std}} := \{u \in C^0(\Omega^{\mathrm{act}}) \ : \ u|_K \in Q^p(K) \ \forall K \in \mathcal{T}^{\mathrm{act}}\}$$



$\mathcal{T}^{\mathrm{act}}, \ \Omega^{\mathrm{act}}$

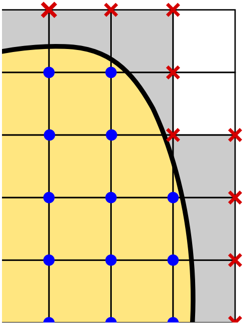


V_h^{std}

Aggregated FE space

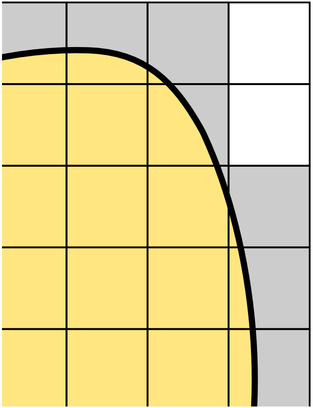
Basic idea: improve conditioning by removing problematic DOFs

$$V_h^{\mathrm{agg}} := \left\{ u \in V_h : \quad u_{\times} = \sum_{\bullet \in \mathrm{masters}(\times)} C_{\times \bullet} u_{\bullet} \quad \forall \times \in \mathcal{P} \right\}$$

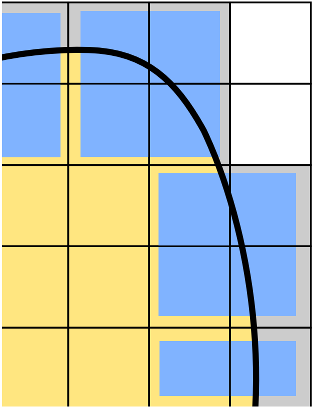


- ● *well-posed dofs*
- × *problematic dofs ($\mathcal{P}$)*

Definition of constraints via cell aggregates

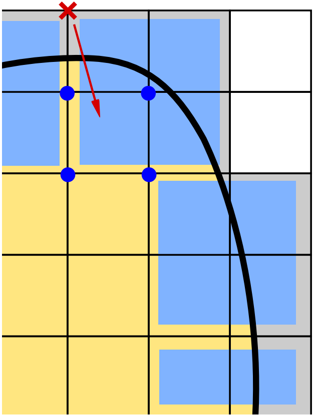


Definition of constraints via cell aggregates



1. Generate cell aggregates
   (1 interior cell + several cut cells)

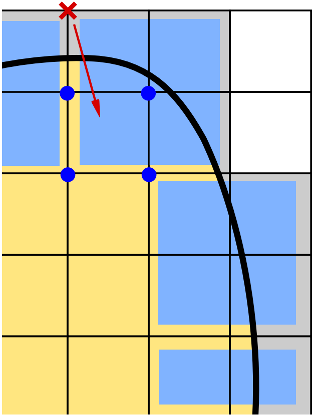
Definition of constraints via cell aggregates



1. Generate cell aggregates
   (1 interior cell + several cut cells)

2. Define dof to root cell map $\mathrm{root}(\times)$
   via the aggregates

Definition of constraints via cell aggregates



1. Generate cell aggregates
   (1 interior cell + several cut cells)

2. Define dof to root cell map $\mathrm{root}(\times)$
   via the aggregates

3. Define constraints:

$$u_{\times} = \sum_{\bullet \in \mathrm{dofs}(\mathrm{root}(\times))} \phi_{\bullet}^{\mathrm{root}(\times)}(x_{\times}) u_{\bullet}$$

Results for the unfitted aggregated FEM (Poisson Eq.)[2]

$$\kappa(\mathbf{A}) \leq c_1 h^{-2} \quad \text{(Condition number bound)}$$

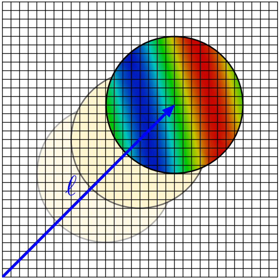
$$\beta \leq c_2 h^{-2} \quad \text{(Nitsche's penalty coef.)}$$

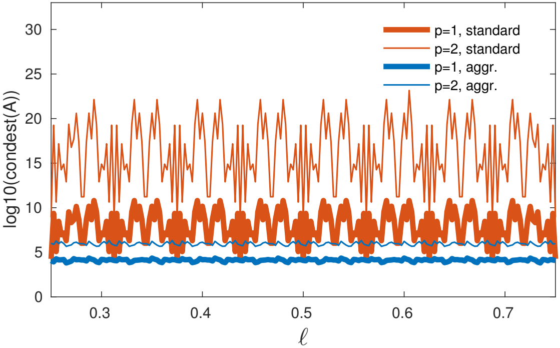
$$\|u - u_h\|_{H^1(\Omega)} \leq c_3 h^p \quad \text{(Optimal convergence order)}$$

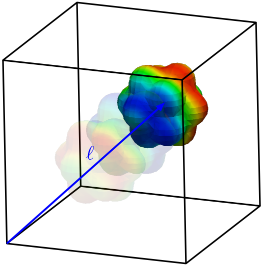
$$\|u - u_h\|_{L^2(\Omega)} \leq c_4 h^{p+1} \quad \text{(Optimal convergence order)}$$

and others (inverse/trace inequalities, bound of aggregate size, bound of the extended solution, ...)

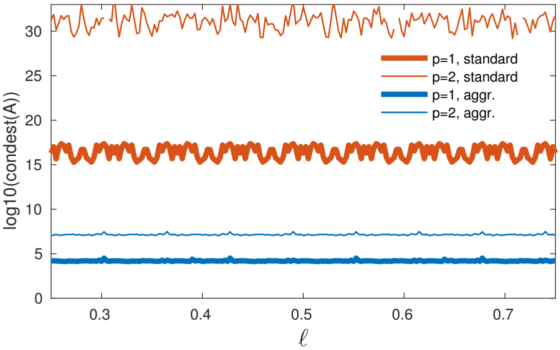
[2] [Badia, Verdugo, Martín. The aggregated unfitted finite element method for elliptic problems. Comput. Methods Appl. Mech. Eng. (2018).]


ℓ


p=1, standard
p=2, standard
p=1, aggr.
p=2, aggr.
log10(condest(A))
0
5
10
15
20
25
30
0.3
0.4
0.5
0.6
0.7
ℓ

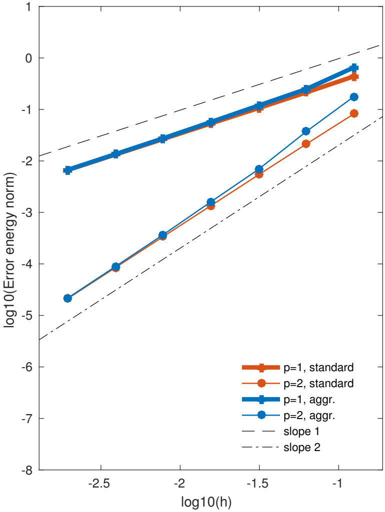


ℓ

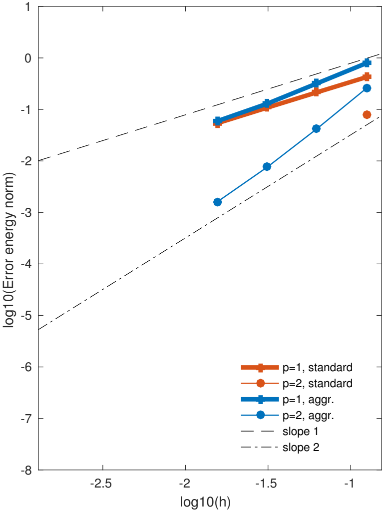


p=1, standard
p=2, standard
p=1, aggr.
p=2, aggr.
log10(condest(A))
0
5
10
15
20
25
30
0.3
0.4
0.5
0.6
0.7
ℓ

Convergence test

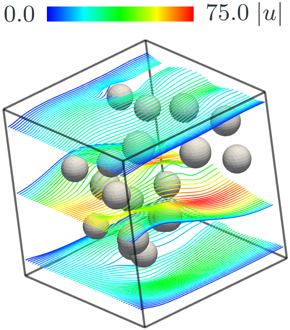


(a) 2D



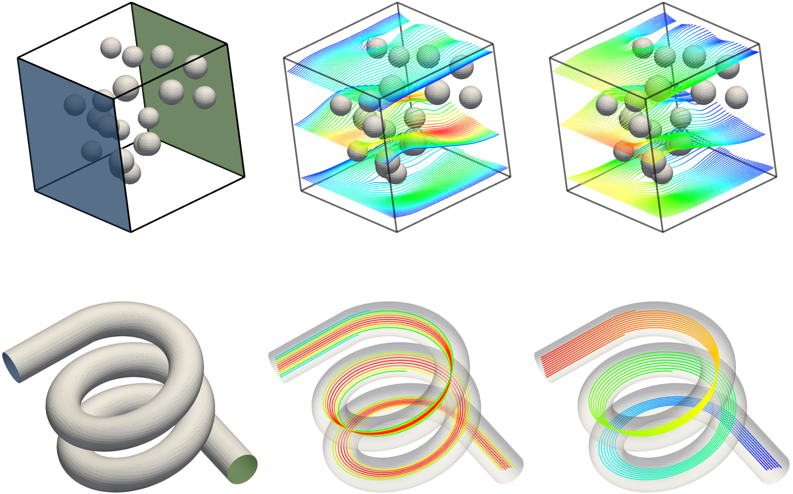
(b) 3D

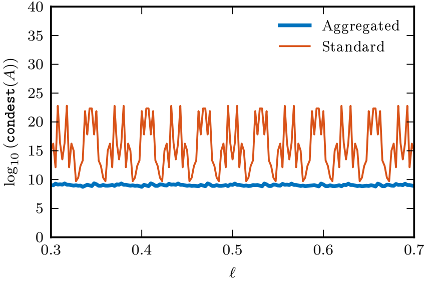
Extension to the Stokes problem[3]

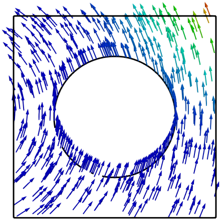


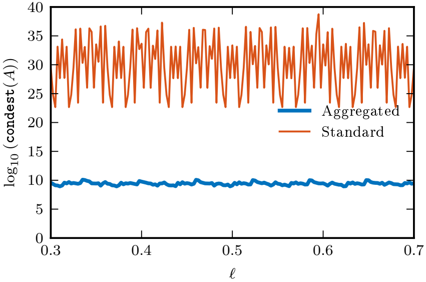
$$\left.\begin{aligned} -\Delta u + \nabla p = f \quad & \text{in } \Omega \\ \nabla \cdot u = 0 \quad & \text{in } \Omega \\ u = 0 \quad & \text{on } \Gamma_{\mathrm{D}} \\ (\nabla u - pI) \cdot n = g \quad & \text{on } \Gamma_{\mathrm{N}} \end{aligned}\right\}$$

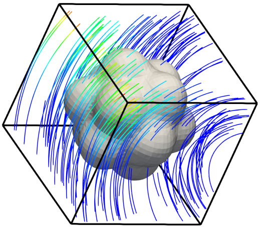
[3][Badia, Martín, Verdugo. Mixed aggregated finite element methods for the unfitted discretization of the stokes problem. SIAM J. Sci. Comput., 40(6). 2018.]

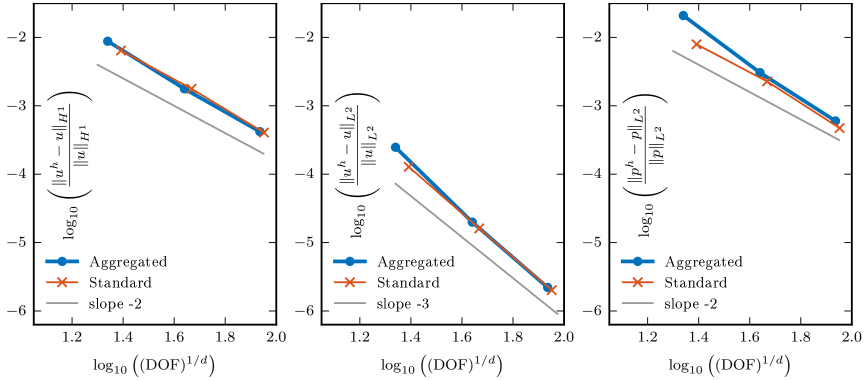



Aggregated
Standard
$\log_{10}(\texttt{condest}(A))$
0
5
10
15
20
25
30
35
40
0.3
0.4
0.5
0.6
0.7
ℓ

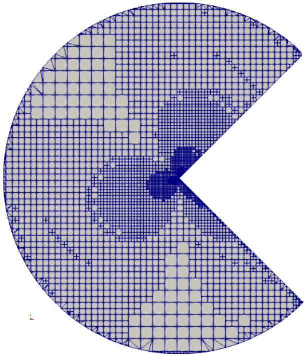


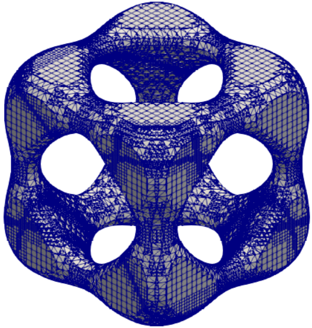

Aggregated
Standard
$\log_{10}(\texttt{condest}(A))$
0
5
10
15
20
25
30
35
40
0.3
0.4
0.5
0.6
0.7
ℓ




−2
−3
−4
−5
−6
1.2
1.4
1.6
1.8
2.0
$\log_{10}\left(\frac{\|u^h - u\|_{H^1}}{\|u\|_{H^1}}\right)$
$\log_{10}\left(\frac{\|u^h - u\|_{L^2}}{\|u\|_{L^2}}\right)$
$\log_{10}\left(\frac{\|p^h - p\|_{L^2}}{\|p\|_{L^2}}\right)$
$\log_{10}\left((\mathrm{DOF})^{1/d}\right)$
Aggregated
Standard
slope -2
slope -3

Extension to non-conforming meshes [4]





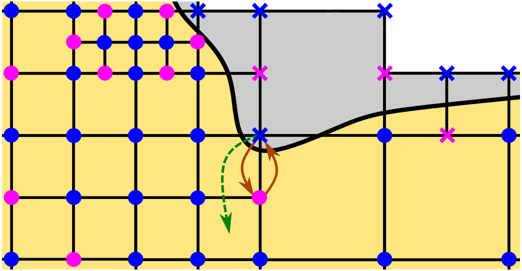
[4][S. Badia, A.F. Martín, E. Neiva, and F. Verdugo. The aggregated unfitted finite element method on parallel tree-based adaptive meshes. Submitted 2020]

Combining AgFEM + hanging node constrains

Well-posed DOFs = Interior DOFs

↓

Cyclic constraints



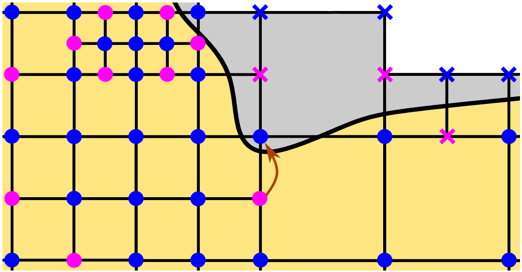
- well-posed free
- well-posed hanging
- × ill-posed free
- × ill-posed hanging

Combining AgFEM + hanging node constrains

Well-posed DOFs = DOFs with local support on interior cells

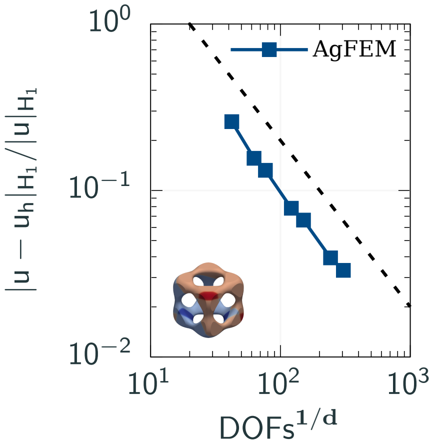
↓

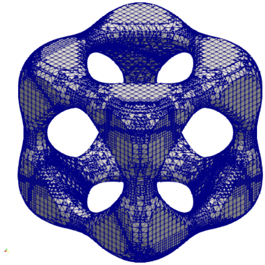
Resolvable constraints



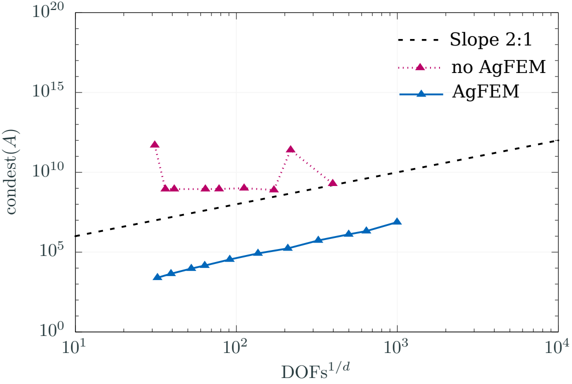
- ● well-posed free
- ● well-posed hanging
- × ill-posed free
- × ill-posed hanging

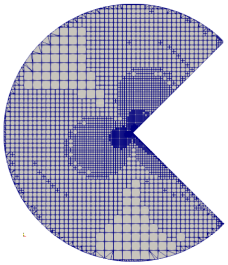
Convergence (Poisson Eq.)





Condition number (Poisson Eq.)





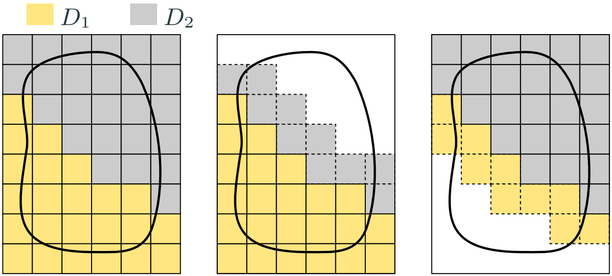
Agenda

1. The AgFEM method (serial case)

2. Parallel implementation

3. Performance of parallel AgFEM + AMG solvers

Domain decomposition setup

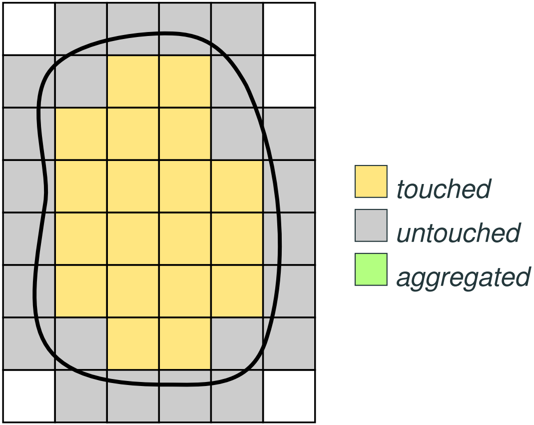


(a) $\mathcal{D} = \{D_1, D_2\}$. **(b)** View from D_1. **(c)** View from D_2

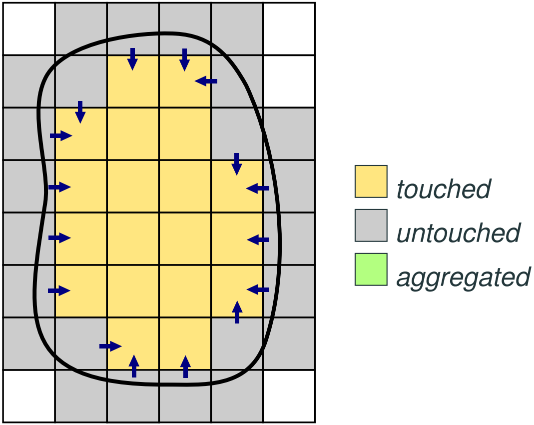
Main phases to be parallelized:

- Cell Aggregation
- Imposition of constraints

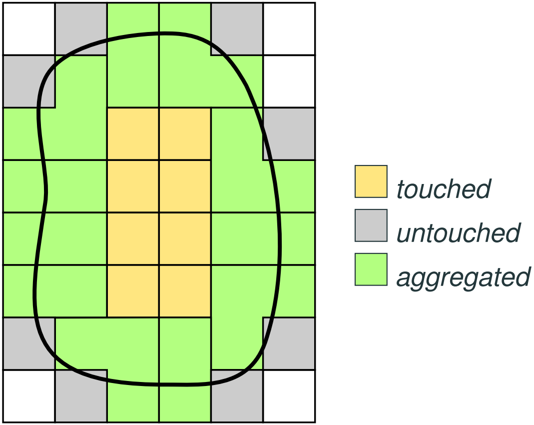
Cell aggregation (serial)



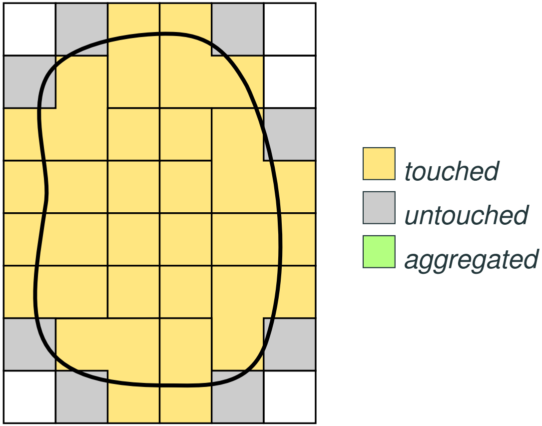
Cell aggregation (serial)



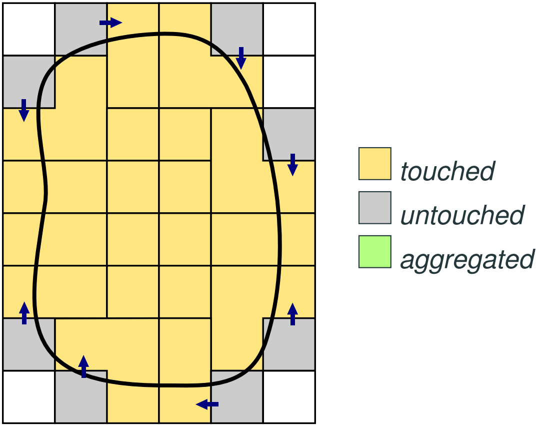
Cell aggregation (serial)



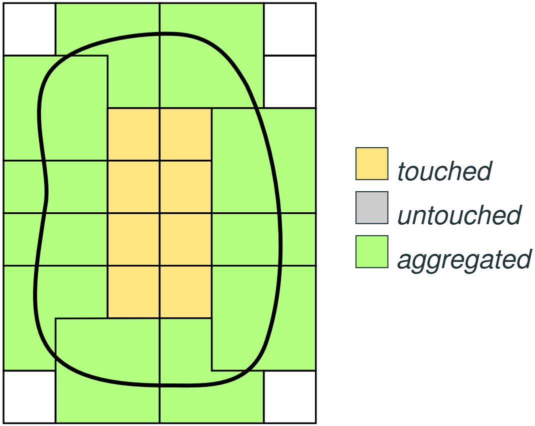
Cell aggregation (serial)



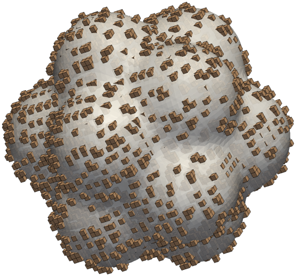
Cell aggregation (serial)

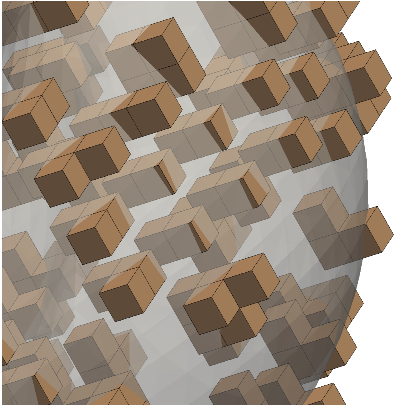


Cell aggregation (serial)

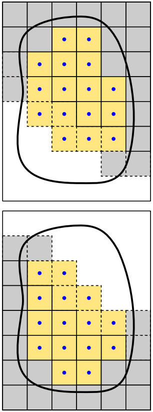


Aggregates in 3D





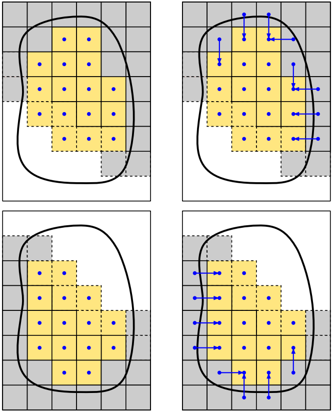
Cell aggregation (parallel)



(a) Step 1.

✔ Standard nearest neighbor communication to determine root cells

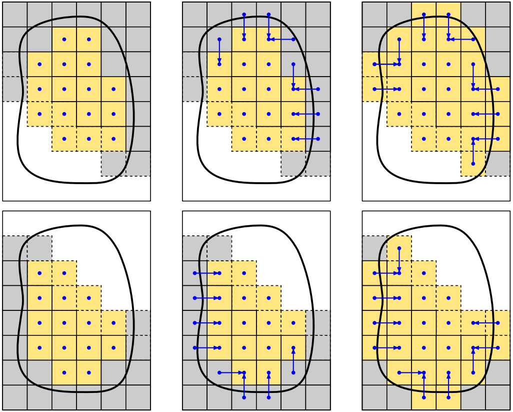
Cell aggregation (parallel)



(a) Step 1. **(b)** Step 2.

✔ Standard nearest neighbor communication to determine root cells

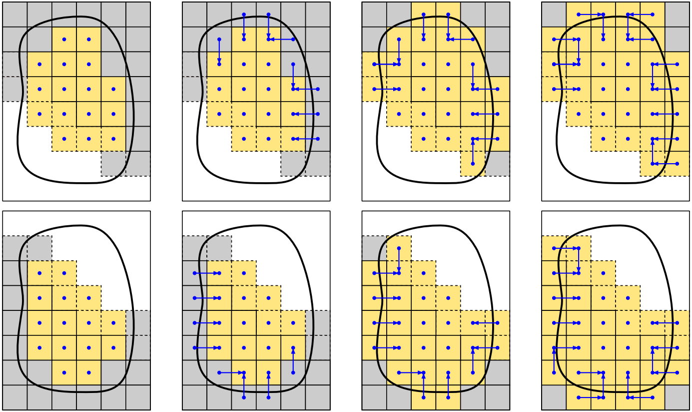
Cell aggregation (parallel)



(a) Step 1. (b) Step 2. (c) Comm.

✔ Standard nearest neighbor communication to determine root cells

Cell aggregation (parallel)



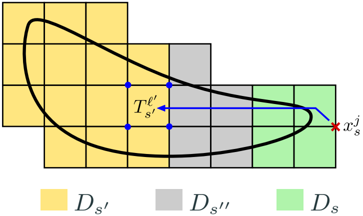
(a) Step 1. **(b)** Step 2. **(c)** Comm. **(d)** Step 3.

✔ Standard nearest neighbor communication to determine root cells

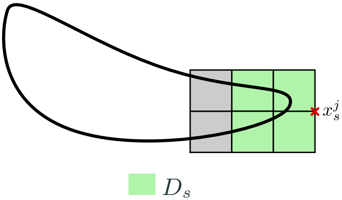
Parallel imposition of constraints

✘ Standard layer of ghost cells not sufficient

$$u_{\times} = \sum_{\bullet \in \mathrm{dofs}(\mathrm{root}(\times))} \phi_{\bullet}^{\mathrm{root}(\times)}(x_{\times}) u_{\bullet}$$



(a) Dof to root cell map

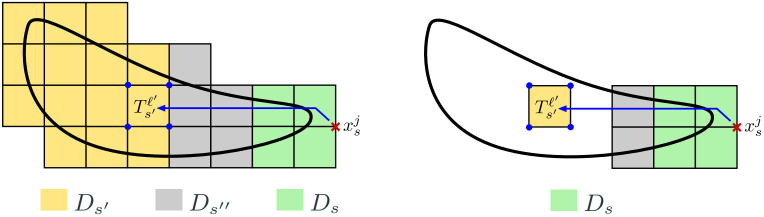


(b) View from D_s

Parallel imposition of constraints

✔ We import extra ghost cells if needed

$$u_{\times} = \sum_{\bullet \in \mathrm{dofs}(\mathrm{root}(\times))} \phi_{\bullet}^{\mathrm{root}(\times)}(x_{\times}) u_{\bullet}$$



(a) Dof to root cell map

(b) View from D_s

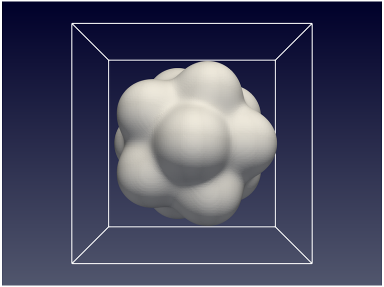
Agenda

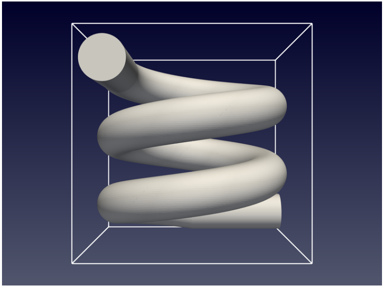
1. The AgFEM method (serial case)

2. Parallel implementation

3. Performance of parallel AgFEM + AMG solvers

Weak scaling test setup



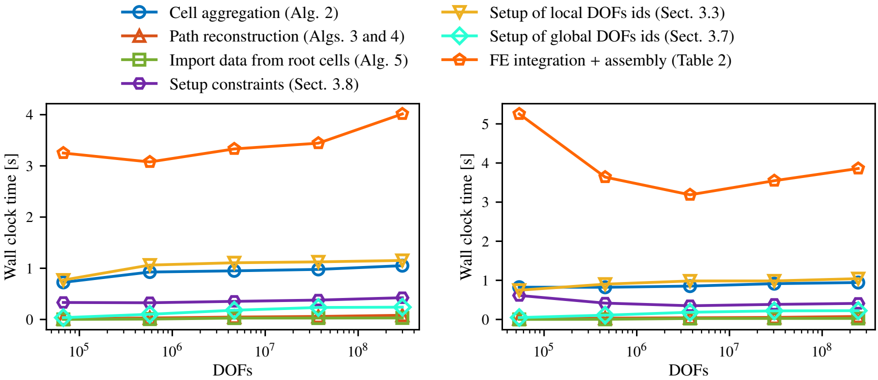


- Poisson eq.
- AgFEM and "naive" unfitted FEM
- Linear solver: PCG from Petsc
- Preconditioner: smooth aggregation AMG from Petsc (GAMG)
- Up to 16K cores and 1000M background cells

Computed at Mare Nostrum 4

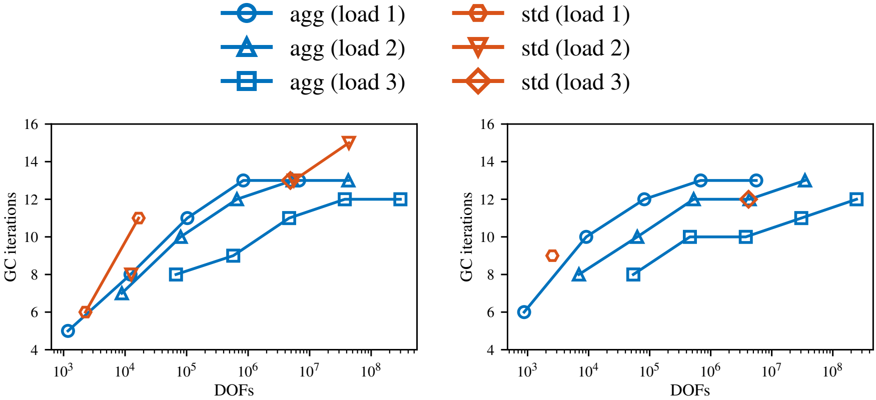
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Weak scalability analysis of AgFEM



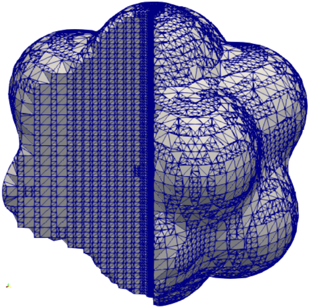
(a) Popcorn **(b)** Spiral

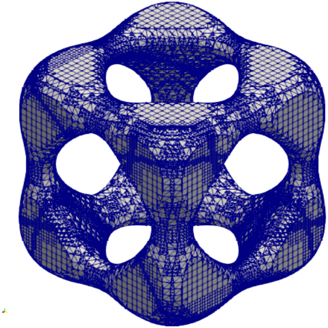
Algorithmic weak scalability linear solver



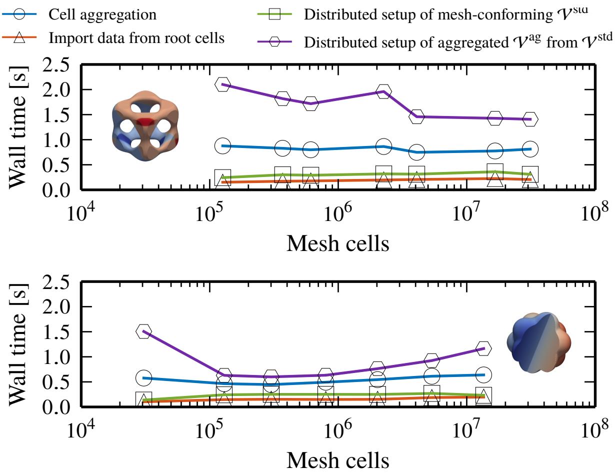
(a) Popcorn **(b)** Spiral

Weak scalability analysis of h-adaptive AgFEM

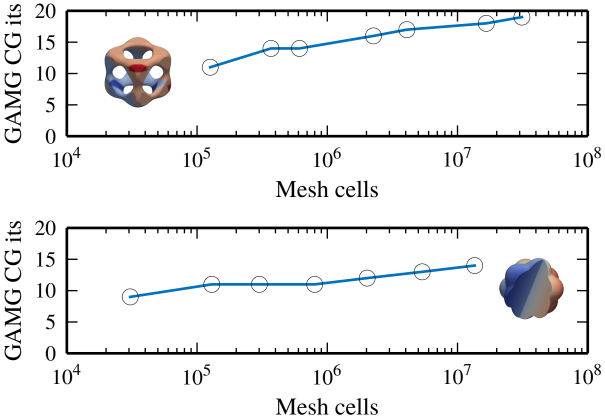




Weak scalability of main AgFEM phases



Algorithmic weak scalability linear solver



Conclusions

- ✔ Embedded FEM simplifies mesh generation and partitioning
- ✘ ... but can destroy the scalability of linear solvers

- ✔ AgFEM allows
- ✔ ... to recover the optimal scaling of linear solver
- ✔ ... while keeping the optimal discretization order

For more details:

- S. Badia, A.F. Martín, E. Neiva, and F. Verdugo. The aggregated unfitted finite element method on parallel tree-based adaptive meshes. *Submitted*. 2020. ArXiv 2006.05373.

- F. Verdugo, A. F. Martin, and S. Badia. Distributed-memory parallelization of the aggregated unfitted finite element method. *Comput. Methods Appl. Mech. Eng.*, 357. 2019.

- S. Badia, A.F. Martín, F. Verdugo. Mixed aggregated finite element methods for the unfitted discretization of the stokes problem. *SIAM J. Sci. Comput.*, 40(6). 2018.

- S. Badia, F. Verdugo, A.F. Martín. The aggregated unfitted finite element method for elliptic problems. *Comput. Methods Appl. Mech. Eng.*, 336. 2018.