Scalability analysis of the distributed-memory implementation of the Aggregated unfitted Finite Element Method (AgFEM)

Alberto F. Martín*, Santiago Badia, Francesc Verdugo, Eric Neiva

MWNDEA 2020, Melbourne, Australia, 12/02/2020

MONASH University

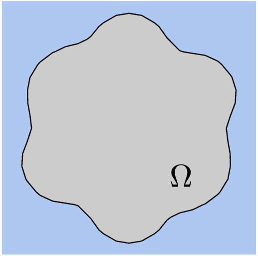
UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

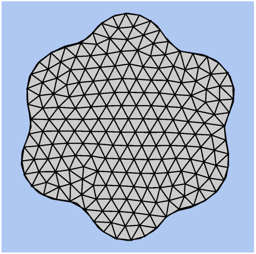
CIMNE EXCELENCIA SEVERO OCHOA

LSSC

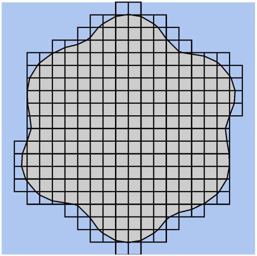
Embedded Finite elements

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





body-fitted mesh

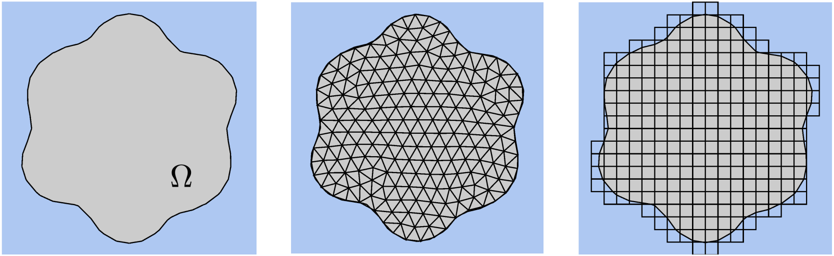


unfitted mesh

✔ Simplified mesh generation

Embedded Finite elements

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



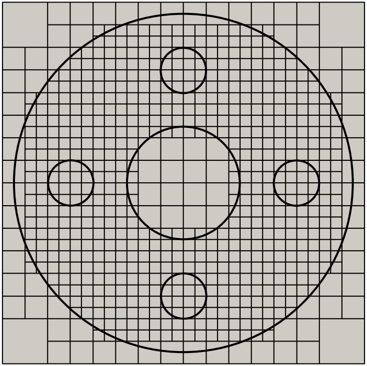
body-fitted mesh unfitted mesh

✔ Simplified mesh generation

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

Parallel distributed-memory simulation pipeline

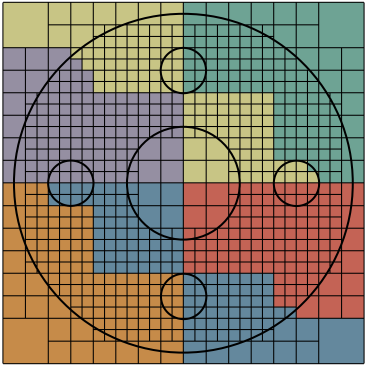
1. Unfitted (adaptive) Cartesian grids (p4est)



Parallel distributed-memory simulation pipeline

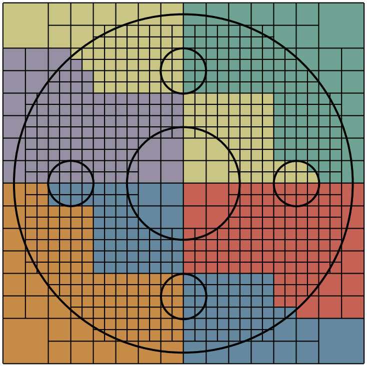
1. Unfitted (adaptive) Cartesian grids (p4est)

2. Partition using space filling-curves (p4est)



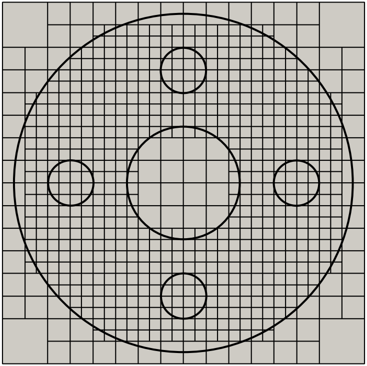
Parallel distributed-memory simulation pipeline

1. Unfitted (adaptive) Cartesian grids (p4est)
2. Partition using space filling-curves (p4est)
3. Unfitted FE discretization (AgFEM)
4. AMG linear solver (PETSc)



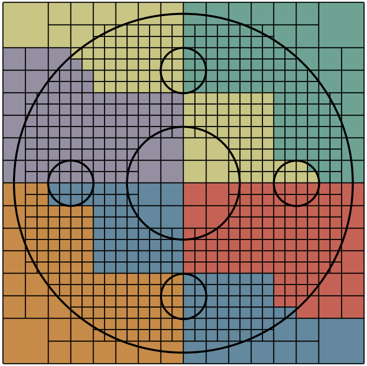
Unfitted methods at large scales: pros and cons

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



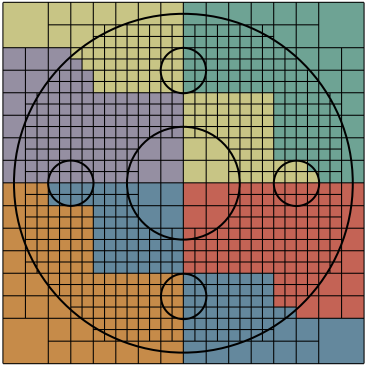
Unfitted methods at large scales: pros and cons

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



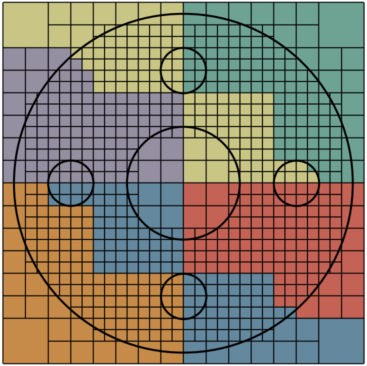
Unfitted methods at large scales: pros and cons

- ✔ 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



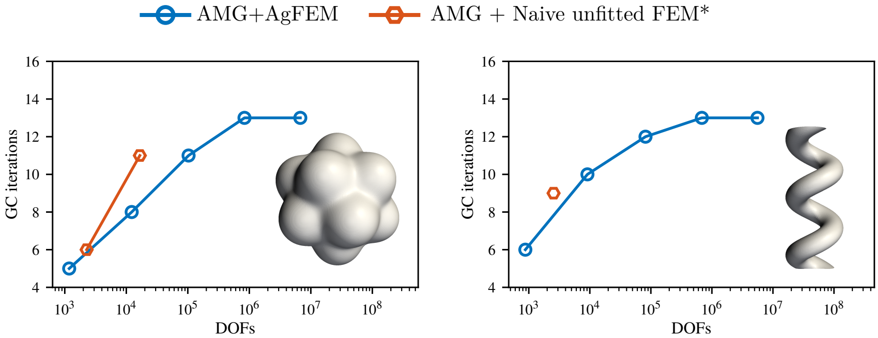
Unfitted methods at large scales: pros and cons

- ✔ 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)

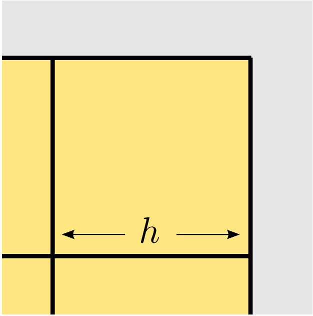


* Nitsche BCs + modified integration in cut cells

Why linear solvers are affected by cut cells?

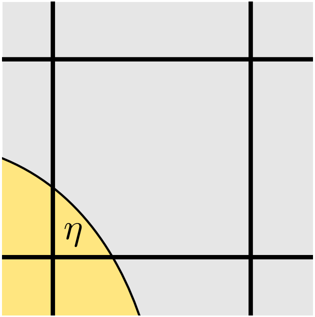
Condition number estimates (Poisson Eq.)

(a) Body-fitted case



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

(b) Naive unfitted FEM



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

"small cut cell problem"

Possible remedies

Fix the linear solver

Taylor your parallel solver to deal with $k_2(A) \sim |\eta|^{-(2p+1-2/d)}$

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

Fix the linear system (this talk)

Enhance the unfitted FE method so that $k_2(A) \sim h^{-2}$

Use a standard scalable solver

Examples: CutFEM, AgFEM

Agenda

1. The AgFEM method (serial case)

2. Parallel implementation

3. Performance of parallel AgFEM + AMG solvers

Agenda

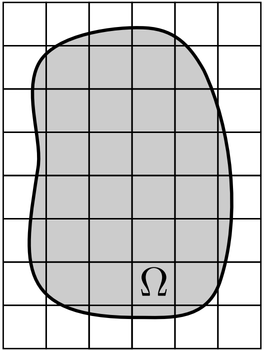
1. The AgFEM method (serial case)

2. Parallel implementation

3. Performance of parallel AgFEM + AMG solvers

AgFEM method for the Poisson Eq.

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



Weak imposition of Dirichlet BCs

Nitsche's Method

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

$$a_h(u^h, v^h) = l_h(v^h) \quad \forall v^h \in V^h \quad (v_h \text{ does not vanish on } \partial\Omega!)$$

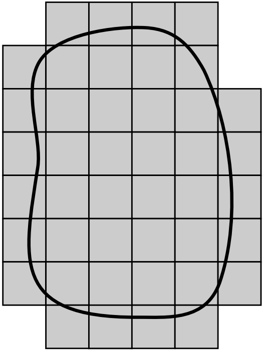
where

$$a_h(u^h, v^h) := \sum_{K \in \mathcal{T}_h^{\text{act}}} \int_{K \cap \Omega} \nabla u^h \cdot \nabla v^h - \sum_{F \in \mathcal{T}_h^{\text{act}} \cap \partial\Omega} \int_F (\nabla u^h \cdot n) v^h$$
$$- \sum_{F \in (\mathcal{T}_h^{\text{act}} \cap \partial\Omega)} \int_F u^h (\nabla v^h \cdot n) + \sum_{F \in (\mathcal{T}_h^{\text{act}} \cap \partial\Omega)} \beta h_F^{-1} \int_{\partial\Omega} u^h \, v^h$$
$$l_h(v^h) := \sum_{K \in \mathcal{T}_h^{\text{act}}} \int_{K \cap \Omega} f \, v^h + \sum_{F \in (\mathcal{T}_h^{\text{act}} \cap \partial\Omega)} \beta h_F^{-1} \int_F u^D v^h - \sum_{F \in (\mathcal{T}_h^{\text{act}} \cap \partial\Omega)} \int_F u^{\text{D}} (\nabla v^h \cdot n)$$

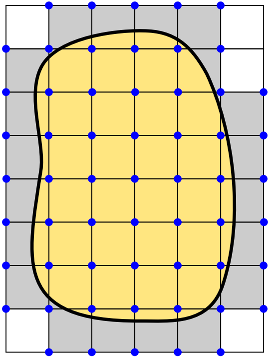
The key feature of AgFEM is the definition of the discrete space V_h

Starting point: "naive" FE space

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



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

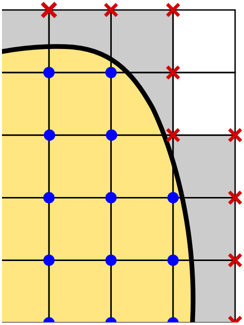


V_h^{std}

Aggregated FE space

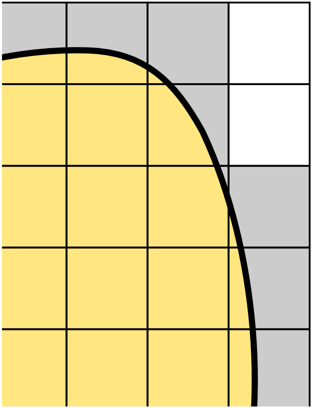
Basic idea: improve conditioning by removing problematic DOFs

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

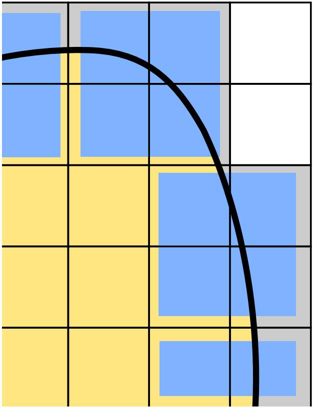


- • *well-posed dofs*
- × *problematic dofs ($\mathcal{P}$)*

Definition of constraints via cell aggregates

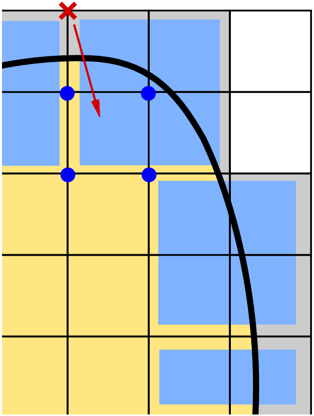


Definition of constraints via cell aggregates



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

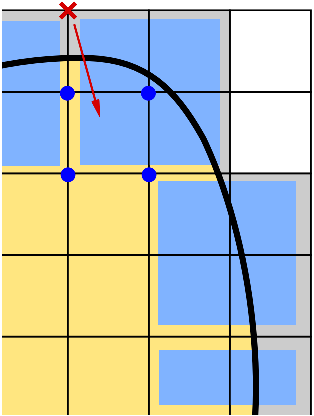
Definition of constraints via cell aggregates



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.)[1]

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

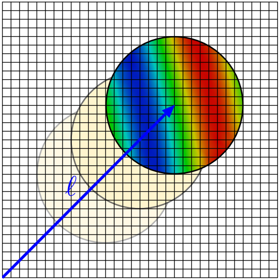
$$\beta \leq c_2 h^{-2} \quad \text{(Nitsche's penalty coef.)}$$

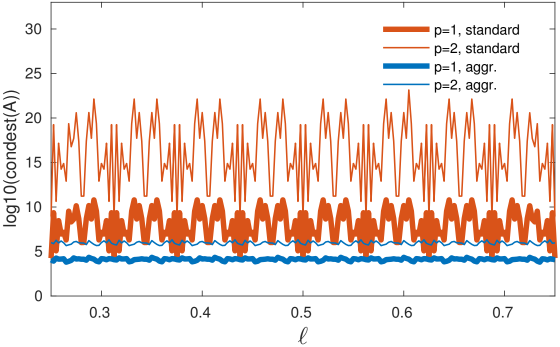
$$\|u - u_h\|_{H^1(\Omega)} \leq c_3 h^p \quad \text{(Optimal convergence order)}$$

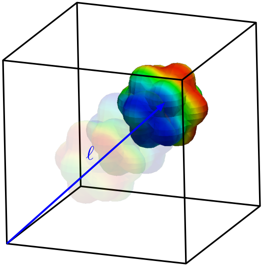
$$\|u - u_h\|_{L^2(\Omega)} \leq c_4 h^{p+1} \quad \text{(Optimal convergence order)}$$

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

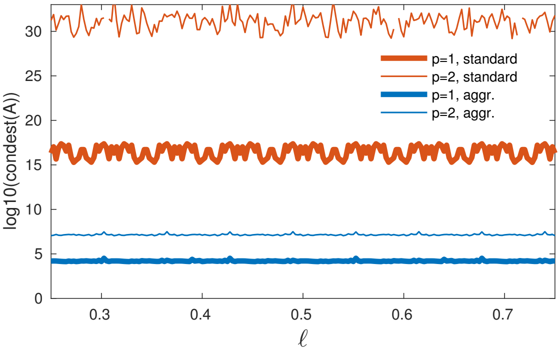
[1] [Badia, Verdugo, Martín. The aggregated unfitted finite element method for elliptic problems. Comput. Methods Appl. Mech. Eng. (2018).]


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

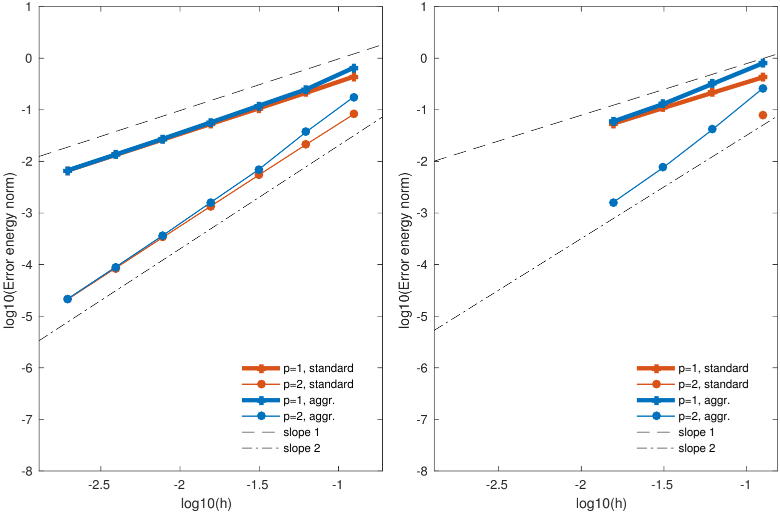


ℓ



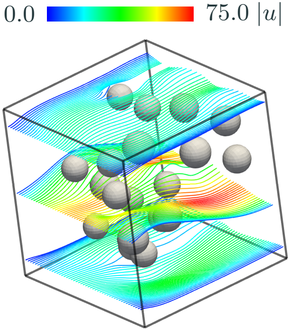
p=1, standard
p=2, standard
p=1, aggr.
p=2, aggr.
log10(condest(A))
0
5
10
15
20
25
30
0.3
0.4
0.5
0.6
0.7
ℓ

Convergence test



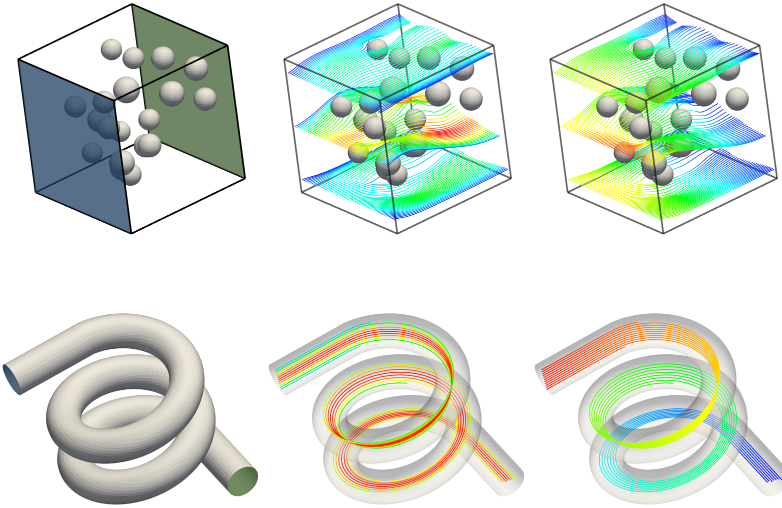
(a) 2D　　**(b)** 3D

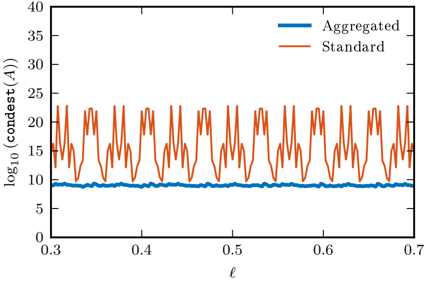
Extension to the Stokes problem[2]

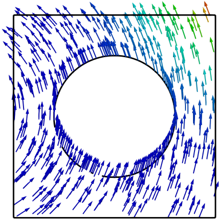


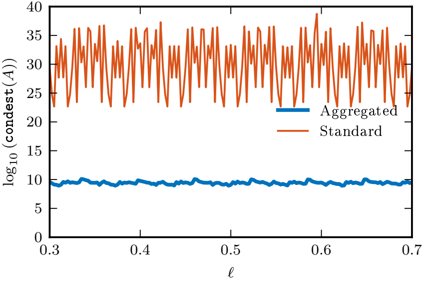
$$\left.\begin{aligned} -\Delta u + \nabla p = f &\quad \text{in } \Omega \\ \nabla \cdot u = 0 &\quad \text{in } \Omega \\ u = 0 &\quad \text{on } \Gamma_{\mathrm{D}} \\ (\nabla u - pI) \cdot n = g &\quad \text{on } \Gamma_{\mathrm{N}} \end{aligned}\right\}$$

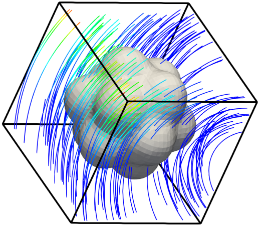
[2][Badia, Martín, Verdugo. Mixed aggregated finite element methods for the unfitted discretization of the stokes problem. SIAM J. Sci. Comput., 40(6). 2018.]

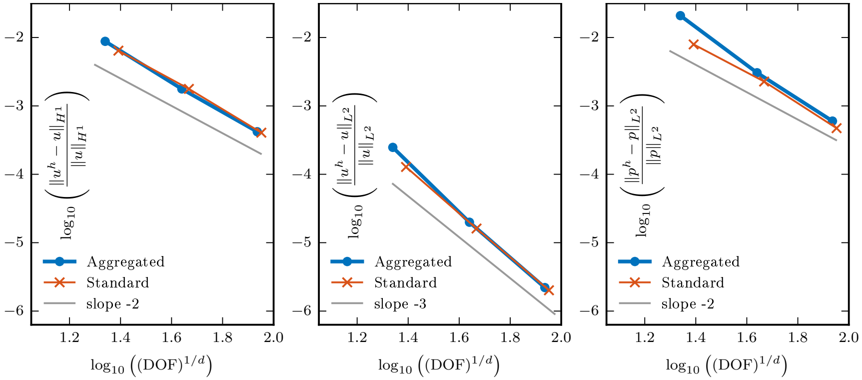



Aggregated
Standard
$\log_{10}(\texttt{condest}(A))$
ℓ
0
5
10
15
20
25
30
35
40
0.3
0.4
0.5
0.6
0.7




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





−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

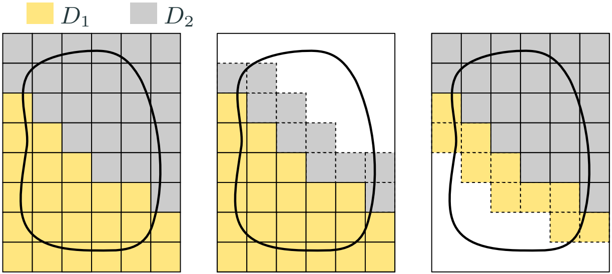
Agenda

1. The AgFEM method (serial case)

2. Parallel implementation

3. Performance of parallel AgFEM + AMG solvers

Parallel mesh distribution

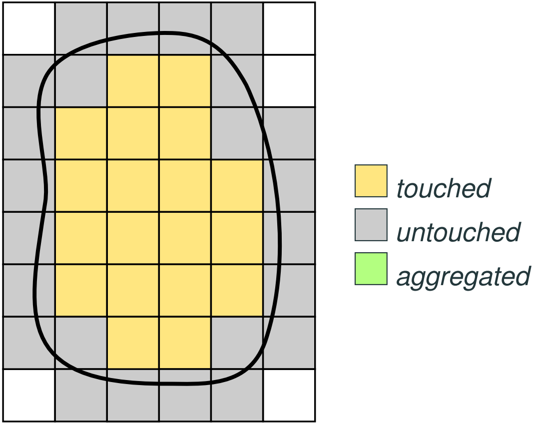


(a) $\mathcal{D} = \{D_1, D_2\}$. **(b)** View from D_1. **(c)** View from D_2

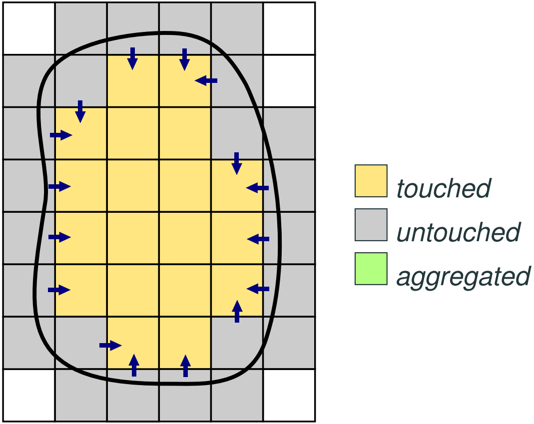
Main phases to be parallelized:

- Cell Aggregation
- Imposition of constraints

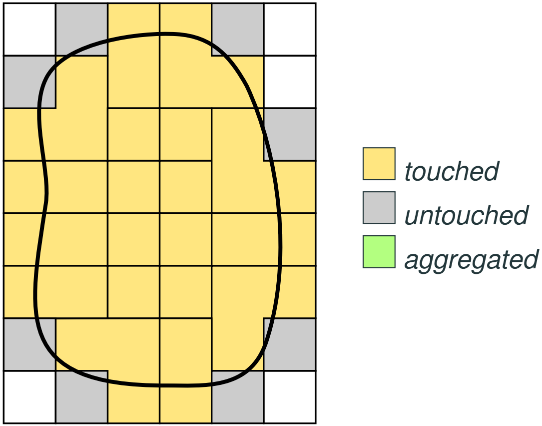
Cell aggregation (serial)



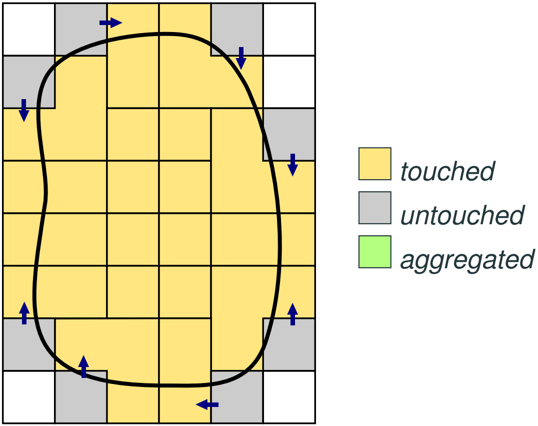
Cell aggregation (serial)



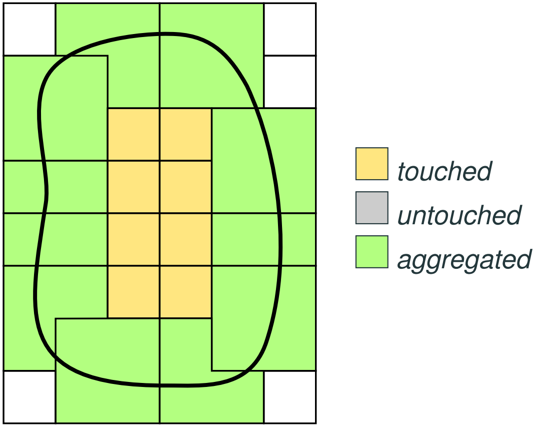
Cell aggregation (serial)



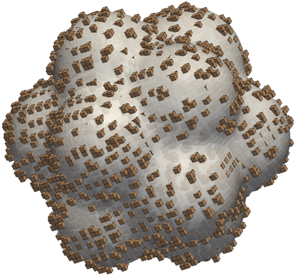
Cell aggregation (serial)

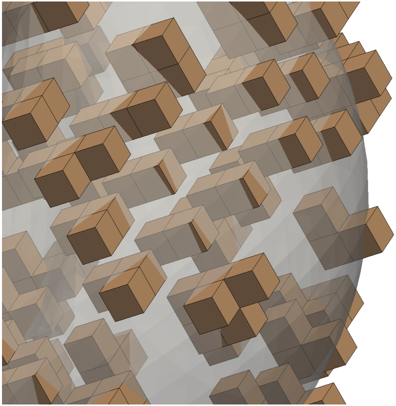


Cell aggregation (serial)

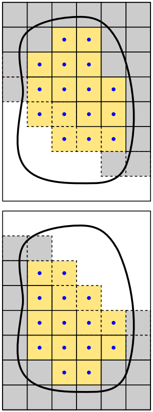


Aggregates in 3D





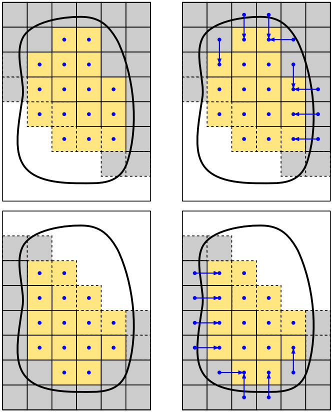
Cell aggregation (parallel)



(a) Step 1.

✔ Standard nearest neighbor communication to determine root cells

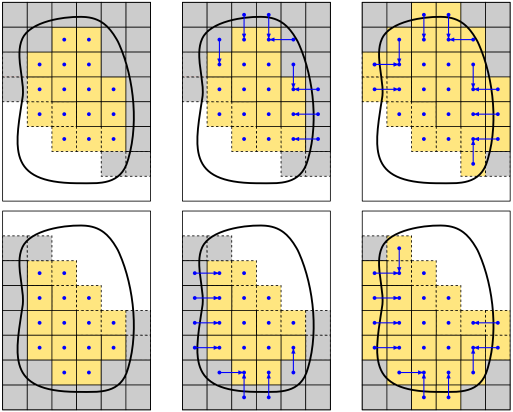
Cell aggregation (parallel)



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

✔ Standard nearest neighbor communication to determine root cells

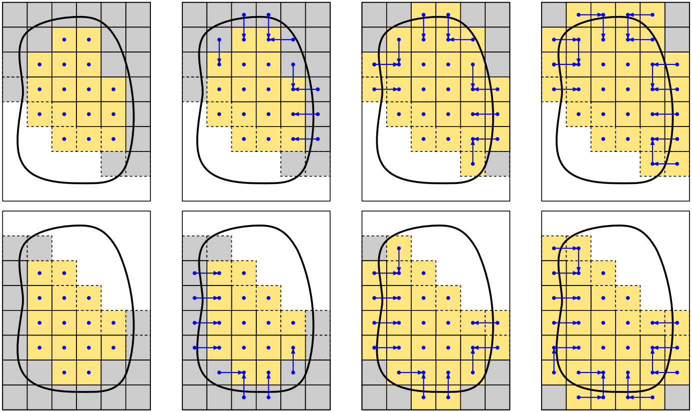
Cell aggregation (parallel)



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

✔ Standard nearest neighbor communication to determine root cells

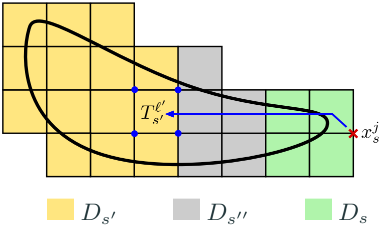
Cell aggregation (parallel)



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

✔ Standard nearest neighbor communication to determine root cells

Parallel imposition of constraints



✘ Subdomain-local constraints not even possible in some cases

✔ At the end, only standard neighbor communication required

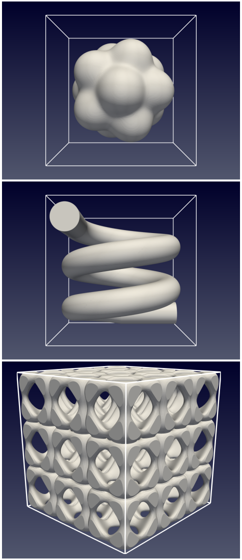
Agenda

1. The AgFEM method (serial case)

2. Parallel implementation

3. Performance of parallel AgFEM + AMG solvers

Weak scaling test setup

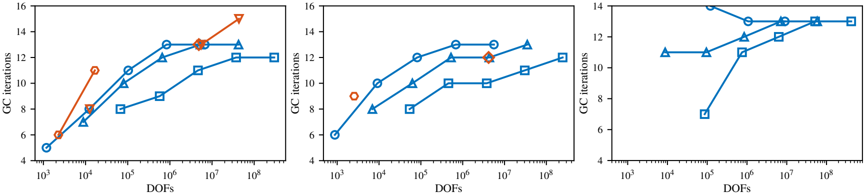


- Poisson eq.
- AgFEM vs "naive" unfitted FEM
- Linear solver:
  PCG from PETSc
- Preconditioner:
  smooth aggregation AMG from PETSc (GAMG)
- Up to 16K cores and 1000M background cells

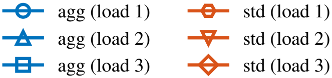
Computed at Mare Nostrum 4

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

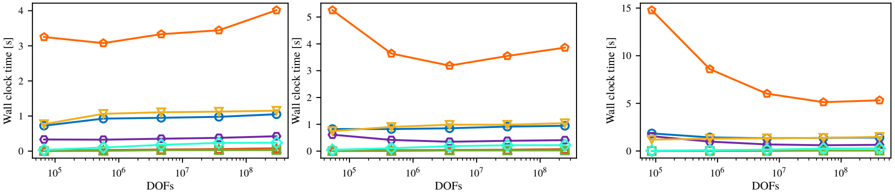
Number of PCG iterations (weak scaling)



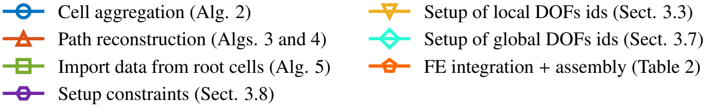
(a) Popcorn (b) Spiral (c) Swiss Cheese



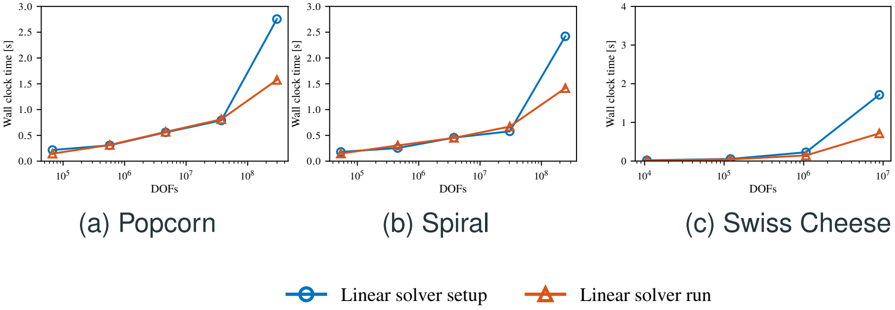
Computational time (secs) AgFEM stages (weak scaling)



(a) Popcorn (b) Spiral (c) Swiss Cheese



Computational time (secs) AMG solver (weak scaling)



(a) Popcorn (b) Spiral (c) Swiss Cheese

- Setup degradation (11.94x CPU time for 4,448x larger problem)
- Similar for standard FEM in a box (2.50x for 355.26x)

Conclusions

- ✔ Embedded FEM enables scalable octree-based meshes
- ✘ ... 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, see papers:

F. Verdugo, A.F. Martín, S. Badia. Distributed-memory parallelization of the aggregated unfitted finite element method. *CMAME,* 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. *CMAME,* 336, 2018.

FEMPAR

`https://github.com/fempar/fempar`