Multi-Domain Solutions of PDEs Posed on Perforated Domains

Miranda Boutilier

In collaboration with Konstantin Brenner and Victorita Dolean

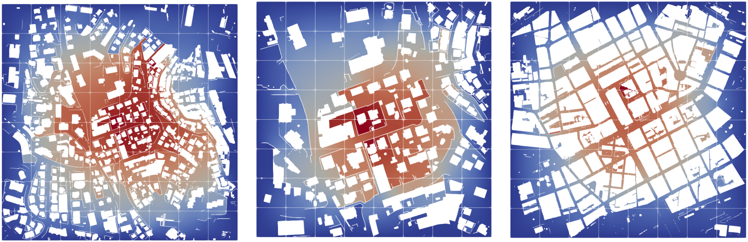
Université Côte d'Azur, LJAD, CNRS

Strasbourg PDE Seminar, December 5th, 2023

Outline

Motivation

- Efficiently solve problems on perforated domains.
  - Numerous holes representing buildings and walls in urban data;
  - Can be considered a heterogeneous domain with coefficients $0, 1$.
  - Expect corner singularities
  - Want to avoid global fine-scale solve.
- We begin with the linear Poisson equation before moving to nonlinear problems (Diffusive Wave model).
- Applications: flood modelling in urban areas.



Model PDE: Linear

- D: Open simply connected polygonal domain in $\mathbb{R}^2$;
- $(\Omega_{S,k})_k$: Finite family of perforations in D;
- $\Omega_S = \bigcup_k \Omega_{S,k}$ and $\Omega = D \setminus \overline{\Omega_S}$.

$$\begin{cases} -\Delta u = f & \text{in} \quad \Omega, \\ \dfrac{\partial u}{\partial n} = 0 & \text{on} \quad \partial\Omega \cap \partial\Omega_S, \\ u = 0 & \text{on} \quad \partial\Omega \setminus \partial\Omega_S. \end{cases}$$

With a P1 finite element discretization, this discretely becomes the linear system

$$\mathbf{Au} = \mathbf{f}.$$

Domain Decomposition Approach

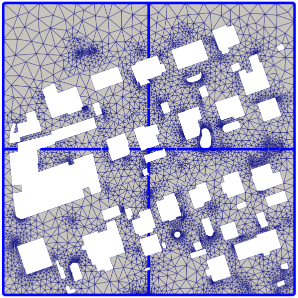
- 'Divide and conquer': Break up problem into subdomains;
- Two levels of discretization: 'Coarse' and 'fine';
- Local subdomain solves can be done in parallel;
- Can use overlapping Schwarz methods as iterative solver or as preconditioner for Krylov;

Idea: Solve model problem on each subdomain locally, with boundary conditions taken from adjacent subdomains when possible.

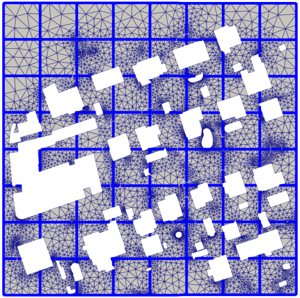
Coarse-cell conforming triangulation

Mesh generation process:

- Larger $N \rightarrowtail$ more basis functions, larger coarse matrix ;
- Triangulate after nonoverlapping coarse cell partitioning Ω'_j;
- Overlap subdomains by layers of triangles for RAS.



2×2 subdomains

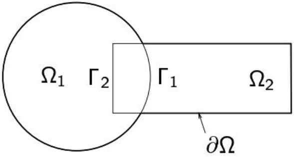


8×8 subdomains

Alternating Schwarz Introduction for $\mathcal{L}u = f$: 2 subdomains

Continuously, the Schwarz iteration is given by

$$\begin{aligned} \mathcal{L}u_1^{n+1} &= f && \text{in} \quad \Omega_1 && \mathcal{L}u_2^{n+1} = f && \text{in} \quad \Omega_2 \\ u_1^{n+1} &= u_2^n && \text{on} \quad \Gamma_1 && u_2^{n+1} = u_1^{n+1} && \text{on} \quad \Gamma_2 \end{aligned}$$



- $\Gamma_1 = \partial\Omega_1 \cap \Omega_2$, $\Gamma_2 = \partial\Omega_2 \cap \Omega_1$.
- Solve on Ω_1, use information from Ω_1 as boundary condition for the solve on Ω_2, etc.

Parallel Schwarz Introduction for $\mathcal{L}u = f$: 2 subdomains

Continuously, the local classical additive Schwarz iteration is given by

$$\begin{aligned} \mathcal{L}u_1^{n+1} &= f \quad &\text{in} \quad &\Omega_1 \qquad & \mathcal{L}u_2^{n+1} &= f \quad &\text{in} \quad &\Omega_2 \\ u_1^{n+1} &= u_2^n \quad &\text{on} \quad &\partial\Omega_1 \cap \Omega_2 \qquad & u_2^{n+1} &= u_1^n \quad &\text{on} \quad &\partial\Omega_2 \cap \Omega_1 \end{aligned}$$

Extending the iteration to multiple subdomains, the algorithm is given by the following:

$$\begin{aligned} \mathcal{L}u_j^{n+1} &= f \quad &\text{in} \quad &\Omega_i \\ u_j^{n+1} &= u_i^n \quad &\text{on} \quad &\partial\Omega_j \cap \Omega_i \end{aligned}$$

for $j = 1, \ldots, N$ and j such that $\partial\Omega_i \cap \Omega_j$ is non-empty.

- At each iteration, use information from adjacent subdomains at **previous** iteration $\rightarrow$ parallel iteration.

Algebraic Form

Algebraically, the global stationary (RAS) iteration becomes

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \left(\sum_{j=1}^{N} \mathbf{R}_j^T \mathbf{D}_j (\mathbf{R}_j \mathbf{A} \mathbf{R}_j^T)^{-1} \mathbf{R}_j \right) (\mathbf{f} - \mathbf{A}\mathbf{u}^n)$$

and the preconditioned system is given by

$$\left(\sum_{j=1}^{N} \mathbf{R}_j^T \mathbf{D}_j (\mathbf{R}_j \mathbf{A} \mathbf{R}_j^T)^{-1} \mathbf{R}_j \right) \mathbf{A}\mathbf{u} = \left(\sum_{j=1}^{N} \mathbf{R}_j^T \mathbf{D}_j (\mathbf{R}_j \mathbf{A} \mathbf{R}_j^T)^{-1} \mathbf{R}_j \right) \mathbf{f}$$

- $\mathbf{R}_j$: Boolean restriction matrices for Ω_j;
- $\mathbf{D}_j$: Partition of unity matrices (deal with overlap);
- $\mathbf{R}_j$ notation allows for global iteration, algebraic definition, overlapping subdomains.

1D example- Restriction, partition of unity matrices

Given set of indices $\mathcal{N} = \{0, 1, 2, 3, 4\}$: partitioned into $\mathcal{N}_1 = \{0, 1, 2, 3\}$ and $\mathcal{N}_2 = \{2, 3, 4\}$, restriction and partition of unity matrices are given as

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{R}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{D}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix} \qquad \mathbf{D}_2 = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

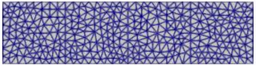
- Satisfies $\mathbf{I} = \sum_{j=1}^{2} \mathbf{R}_j^T \mathbf{D}_j \mathbf{R}_j$.

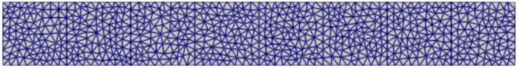
Need for coarse correction

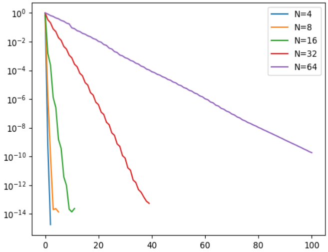
- Coarse corrections allows for global communication between all subdomains.
- Coarse correction (two-level methods) necessary for scalability for large number of subdomains.
- Generally, without coarse correction: Iterations scale with N.

Numerical Comparison: Without coarse correction

- Weak Scalability: fixed subdomain and fine triangulation size, keep $\frac{H}{h}$ constant.
- Shown on homogeneous 2D domain (subdomains in 1 dimension).







(Some) existing overlapping Schwarz coarse spaces

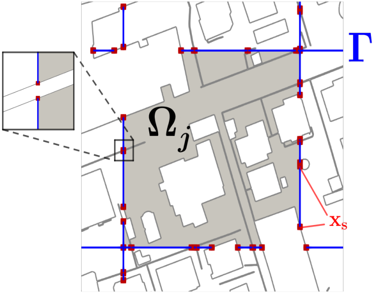
- Nicolaides: Piecewise constant by subdomain;
- Spectral spaces (eigenvalue problems): DtN, GenEO, SHEM (spectrally enriched MSFEM);
- Energy-minimizing spaces: GDSW, AGDSW, RGDSW;
- Multi-scale FEM: MsFEM
  - Numerically compute harmonic basis functions.
  - Used to approximate solution on coarse grid, but can use as DD coarse space!

Choice of coarse space

- Idea: want to take advantage of a-priori location of perforations (buildings/walls);
- Want robustness with respect to perforation size/location (even along subdomain interfaces);
- Want to choose a coarse space with approximation properties to improve convergence;
- Choose: Local harmonic basis functions occuring at intersection of a perforation with the coarse skeleton.
  - Think of as 'enriching' MsFEM coarse space.
  - Based on nonoverlapping subdomains.

Coarse grid nodes for coarse space basis functions

- Nonoverlapping skeleton: $\Gamma = \bigcup_{j\in\{1,\ldots,N\}} \partial\Omega'_j$;
- $(e_k)_{k=1,\ldots,N_e}$: Partitioning of Γ;
  - each "coarse edge" e_k is an open planar segment;
- Set of coarse grid nodes: $\bigcup_{k=1,\ldots,N_e} \partial e_k$
- $(\phi_s)_{s\in\{1,\ldots,N_x\}}$: Locally harmonic basis functions for each coarse grid node.
- $\#$ of coarse grid nodes is automatically generated.

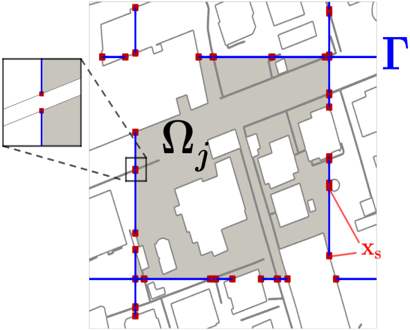


Basis functions: boundary conditions

For each coarse grid node $\mathbf{x}_s$, define $g_s : \Gamma \rightarrow [0,1]$ as: for $i = 1, \ldots N_{\mathbf{x}}$,

$$g_s(\mathbf{x}_i) = \begin{cases} 1, & s = i, \\ 0, & s \neq i, \end{cases}$$

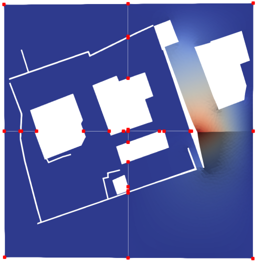
- g_s is linearly extended on the remainder of Γ.
- Can also include higher-order polynomials on coarse edges.



Basis functions: Harmonic local solutions

For all nonoverlapping $\left(\Omega'_j\right)_{j\in\{1,\dots,N\}}$ and $s = 1, \dots, N_{\mathbf{x}}$, to obtain $\phi_{s,j} = \phi_s|_{\Omega_j}$, solve

$$\begin{cases} -\Delta\phi_{s,j} = 0 & \text{in} \quad \Omega'_j, \\ -\dfrac{\partial\phi_{s,j}}{\partial n} = 0 & \text{on} \quad \partial\Omega'_j \cap \partial\Omega_S, \\ \phi_{s,j} = g_s & \text{on} \quad \partial\Omega'_j \setminus \partial\Omega_S. \end{cases}$$



- $\mathrm{supp}(\phi_s) = \{\bigcup_j \Omega'_j \mid \mathbf{x}_s$ is a coarse grid node belonging to $\partial\Omega'_j\}$.
- Continuously, the coarse space is given by $V_H = \mathrm{span}\{\phi_s\}$.
- Discretely, columns of coarse matrix $\mathbf{R}_0^T$ are the discrete harmonic basis functions.

2-level RAS iteration: N Subdomains

Combine (multiplicitavely) the 1-level RAS iteration

$$M_{RAS,1}^{-1} = \sum_{j=1}^{N} \mathbf{R}_j^T \mathbf{D}_j (\mathbf{R}_j \mathbf{A} \mathbf{R}_j^T)^{-1} \mathbf{R}_j$$

with the coarse approximation

$$M_0^{-1} = \mathbf{R}_0^T (\mathbf{R}_0 \mathbf{A} \mathbf{R}_0^T)^{-1} \mathbf{R}_0.$$

and solve

$$\mathbf{u}^{n+\frac{1}{2}} = \mathbf{u}^n + M_{RAS,1}^{-1}(\mathbf{f} - \mathbf{A}\mathbf{u}^n),$$
$$\mathbf{u}^{n+1} = \mathbf{u}^{n+\frac{1}{2}} + M_0^{-1}(\mathbf{f} - \mathbf{A}\mathbf{u}^{n+\frac{1}{2}}),$$

- $\mathbf{R}_j$: Correspond to overlapping subdomains.

The 2-level preconditioner for Krylov

Combine (additively) the 1-level RAS iteration

$$M_{RAS,1}^{-1} = \sum_{j=1}^{N} \mathbf{R}_j^T \mathbf{D}_j (\mathbf{R}_j \mathbf{A} \mathbf{R}_j^T)^{-1} \mathbf{R}_j$$

with the coarse approximation

$$M_0^{-1} = \mathbf{R}_0^T (\mathbf{R}_0 \mathbf{A} \mathbf{R}_0^T)^{-1} \mathbf{R}_0.$$

to give

$$M_{RAS,2}^{-1} = M_0^{-1} + M_{RAS,1}^{-1}.$$

and solve

$$M_{RAS,2}^{-1} \mathbf{A} \mathbf{u} = M_{RAS2}^{-1} \mathbf{f}.$$

Approximation properties: Multiscale approximation

Discretely, given

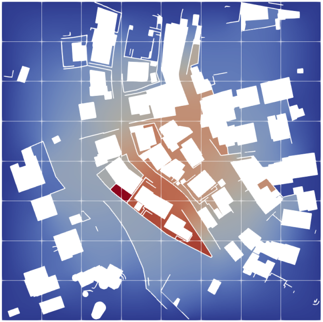
$$M_0^{-1} = \mathbf{R}_0^T(\mathbf{R}_0\mathbf{A}\mathbf{R}_0^T)^{-1}\mathbf{R}_0.$$

the coarse approximation is the solution of

$$\mathbf{u}_H = M_0^{-1}\mathbf{f}.$$

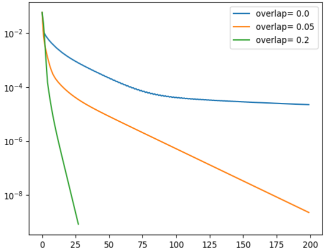
- Can use u_H as initial iterate for iteration, Krylov methods.

Linear Numerical Results: Iterative+Krylov, real data set

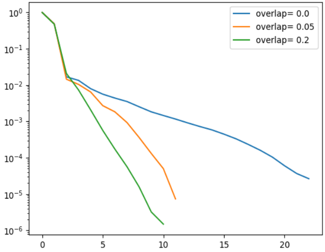


- Compare iterative to Krylov with various overlap values;
- Multiple singularities and no analytical solution available.

Numerical Results: Iterative RAS (Real data)



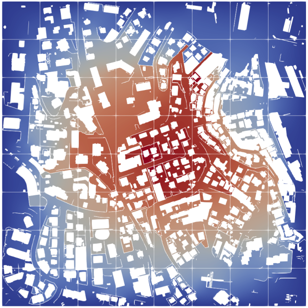
Iterative



Krylov

- Fast convergence with Krylov acceleration.
- As expected, faster convergence with larger overlap.

Experiment 3: Krylov Scalability, large real data set



$\approx$ 300K DOFS in FE triangulation.

- Want to show scalability:
- "Strong" scalability tests: Keep model domain and h constant, vary N.

Numerical Results: Krylov (table)

	Trefftz		
	it.		dim. (rel)
N	min	$\frac{H}{20}$	
16	56	22	400 (16.0)
64	56	26	880 (10.9)
256	59	30	1912 (6.6)
1024	61	28	4253 (3.9)

- Relative dimension (rel): Compared to would-be homogeneous domain, $\frac{\dim(R_0)}{(\sqrt{N}+1)^2}$.
- Relative dimension reduces as N increases;
- Trefftz-like space produces scalable, accelerated iterations.

Nonlinear Problem: Diffusive Wave model

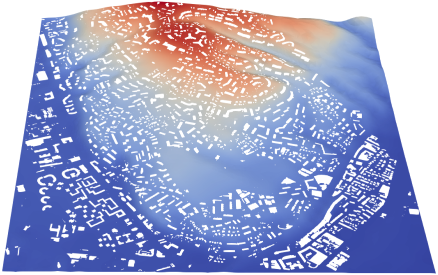
$$\left\{ \begin{array}{rcllll} \partial_t u + \mathrm{div}\mathcal{F}(x,u,\nabla u) & = & f & \text{in} & \Omega, & \\ \mathcal{F}(u)\cdot \mathbf{n} & = & 0 & \text{on} & \text{on} & \partial\Omega \cap \partial\Omega_S, \\ u & = & g & \partial\Omega \setminus \partial\Omega_S. & & \end{array} \right.$$

$$\mathcal{F}(x,u,\nabla u) = h(u, z_b(\mathbf{x}))^{\alpha} ||\nabla u||^{p-2} \nabla u,$$

- $z_b(\mathbf{x})$: Bathymetry;
- $h(u, z_b(\mathbf{x})) = \max(u - z_b(\mathbf{x}), 0)$: Water depth;
- κ: Friction coefficient;
- $\alpha > 1, 1 < p \leq 2$.

Forming Realistic Problem

- Realistic bathymetry /topography of Nice, France: 5m data;
- Rainfall data (source term): Can be taken from previous flood events (rain gauge data);
- Discretization of Problem: FEM/FV Hybrid with upwinding.



Discretization

We obtain the nonlinear system

$$F(U^n) = \frac{1}{\Delta t} M(U^n - U^{n-1}) + K(U^n) = 0, \quad (1)$$

where M is the (lumped) mass-matrix.

- Time derivative is computed via backward-Euler;
- $K(U^n)$ is discretization of nonlinear term (FEM/FVM);
- Perform upwinding on $h(u, z_b(\mathbf{x}))^\alpha$ term (due to degeneracy);
- Adaptive time-stepping may be necessary for Newton's method.

Nonlinear Preconditioning

Goal: instead of $F(U) = 0$, solve $N(F(U)) = 0$.

- $N(v) = 0 \rightarrow v = 0$;
- $N(F(v))$ straightforward to compute.

Recall from linear problem $\rightarrow$ fixed point iteration leads to a well-suited preconditioner.
Idea: From some fixed point iteration

$$U^{n+1} = P(U^n), \tag{2}$$

solve $\mathcal{F}(U) = P(U) - U = 0$ via nonlinear solve.

- $\mathcal{F}(U) = 0$ is preconditioned nonlinear system.

Nonlinear RAS iteration

Similarly to the linear problem, use local subdomain solves and glue together to form fixed-point iteration.

$$U^{n+1} = \sum_j R_j^T D_j G_j(U^n), \tag{3}$$

where $G_j(U^n)$ is the solution of

$$R_j F(R_j^T G_j(U^n) + (I - R_j^T R_j)U^n)) = 0. \tag{4}$$

- Local subproblems are solved via Newton with negligible cost;
- Local solves can be done in parallel.

RASPEN

As mentioned, solve $\mathcal{F}(U) = \sum_j R_j^T D_j G_j(U) - U = 0$ (RASPEN). via Newton.

- an "improvement" from ASPIN, converges in the overlap;
- Inner nonlinear solves allow for computation of exact Jacobian $\nabla\mathcal{F}$, or specifically the matrix-vector product $\nabla\mathcal{F}v$ for some v.

RASPEN: Computation of Jacobian

Recall equation for local nonlinear solves:

$$R_j F(R_j^T G_j(U^n) + (I - R_j^T R_j)U^n)) = 0.$$

Taking the derivative of this equation, we obtain

$$\nabla G_j(U^n) = R_j - [R_j \nabla F(U^n) R_j^T]^{-1} R_j \nabla F(U^n);$$

This gives

$$\begin{aligned} \nabla \mathcal{F}(U^n) &= \nabla(U^n - \sum_j R_j^T D_k \nabla G_j(U^n)) \\ &= \sum_j R_j^T D_j [R_j \nabla F(U^n) R_j^T]^{-1} R_j \nabla F(U^n) \end{aligned}$$

- $R_j \nabla F(U^n) R_j^T$, $\nabla F(U^n)$ can be reused from local nonlinear solves.

One-level RASPEN

The algorithm, for each time step, is given by: For outer iteration $n = 0, \dots,$ to convergence,

- Solve $\widehat{U}^n = \sum_j R_j^T D_j G_j(U^n)$ by gluing local solutions;
- Set $\mathcal{F}(U) = U - \widehat{U}^n$;
- Solve $U^{n+1} = U^n - [\nabla\mathcal{F}(U^n)]^{-1}\mathcal{F}(U^n)$ via GMRES, where $\nabla\mathcal{F}(U^n)$ is assembled as a linear operator.

Two-level RASPEN

While there are many different ways to choose the coarse correction (including FAS !cite inspired by Multigrid, we add the coarse correction multiplicitavely, with a discrete matrix R_0.

The algorithm, for each time step, given by: For outer iteration $n = 0, \ldots,$ to convergence,

- solve local subproblems $R_j F(R_j^T G_j(U^n) + (I - R_j^T R_j)U^n)) = 0$ for $G_j(U^n)$;
- Set $\widehat{U}^n = \sum_j R_j^T D_j G_j(U^n)$ by gluing local solutions;
- Solve coarse problem $R_0 F(\widehat{U}^n - R_0^T c_o^n) = 0$ for c_0^n;
- Set $\mathcal{F}(U^n) = U^n - \widehat{U}^n + R_0^T c_o^n$;
- Solve $U^{n+1} = U^n - [\nabla \mathcal{F}(U^n)]^{-1} \mathcal{F}(U^n)$ via GMRES, where $\nabla \mathcal{F}(U^n)$ is assembled as a linear operator.

Coarse Galerkin Approximation

Coarse Galerkin Formulation: solve

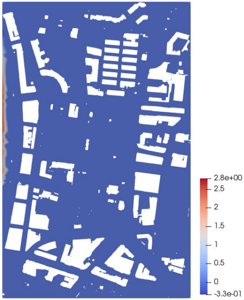
$$\mathbf{R}_0 F(\mathbf{R}_0^T u_H^n) = 0 \tag{5}$$

for each time step via Newton.

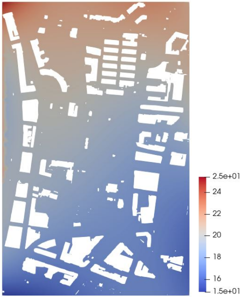
- Residual still takes global vector as input, but input vector is sparse;
- Much more efficient than global Newton solve (cheaper outer iterations).

Setup example model problem

- Excessive water flow coming from Paillon river in Nice, France;
- Dirichlet boundary conditions with initial condition $u_0 > z_b$ at leftmost boundary (river).
- $\alpha = \frac{3}{2}, p = 2$ (ignoring gradient term), 0 source term.

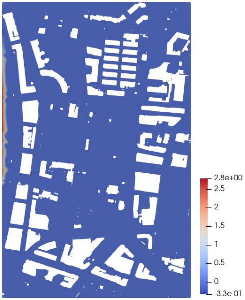


initial h

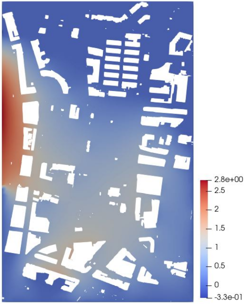


initial u

Solution at final time (h)



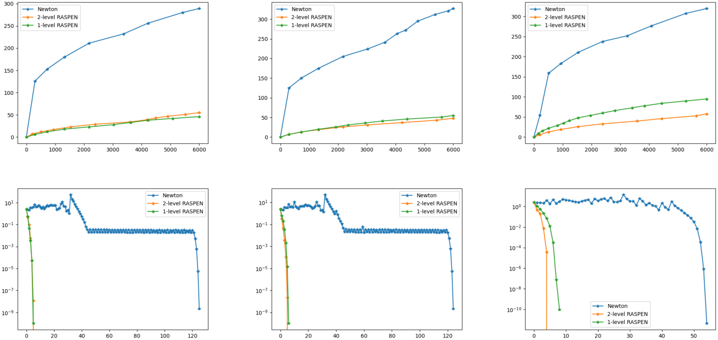
$t = 0$



$t = t_f$

- Effect of z_b is visible.

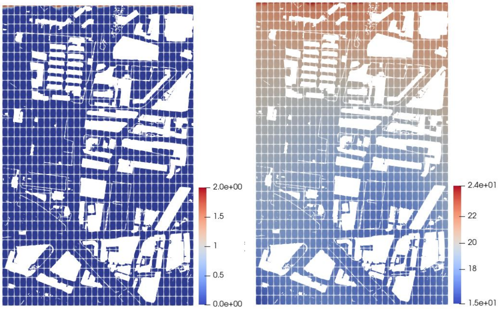
Numerical results

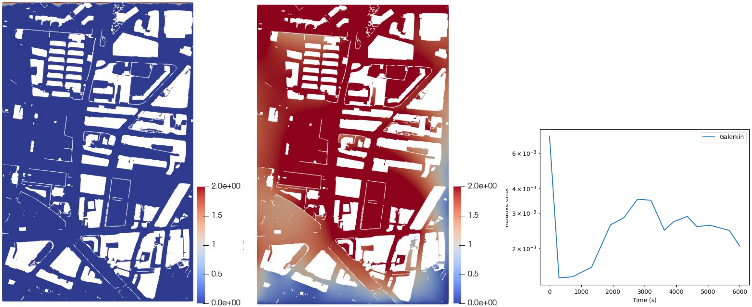


- Left to right: N=2, 4,8.
- Top: cumulative iterations over time. Bottom: convergence history for first time step.
- $\mathrm{dt}_0 = 5$ minutes, with increasing/decreasing by $\sqrt{2}$ depending on convergence.

Setup example model problem: Comparison of Coarse Galerkin and Newton

- Excessive water flow coming from the top of the domain with Dirichlet boundary conditions;
- $\alpha = \frac{3}{2}, p = 2$ (ignoring gradient term).
- Comparison between Coarse Galerkin and Newton.





- Left to right: solution h at initial time, solution h at final time, error between coarse Galerkin and Newton over time.
- Runtimes are 370 seconds (coarse Galerkin), 3614 seconds (Newton).
- Coarse Galerkin method gives acccurate solution

Closing Remarks

- We have presented a novel Trefftz coarse space that can be used to approximate the fine-scale solution;

Closing Remarks

- We have presented a novel Trefftz coarse space that can be used to approximate the fine-scale solution;
- The space can also be used in combination with Schwarz methods to achieve fine-scale accuracy.

Closing Remarks

- We have presented a novel Trefftz coarse space that can be used to approximate the fine-scale solution;
- The space can also be used in combination with Schwarz methods to achieve fine-scale accuracy.
- For the nonlinear problem, nonlinear preconditioning can be used in a similar manner to Krylov acceleration (accelerating a fixed-point iteration);

Closing Remarks

- We have presented a novel Trefftz coarse space that can be used to approximate the fine-scale solution;
- The space can also be used in combination with Schwarz methods to achieve fine-scale accuracy.
- For the nonlinear problem, nonlinear preconditioning can be used in a similar manner to Krylov acceleration (accelerating a fixed-point iteration);
- Performing the coarse Galerkin method is cheap, easy to implement, and reasonably accurate.

Funding Acknowledgement

This work has been supported by ANR Project Top-up (ANR-20-CE46-0005).

We also thank Métropole Nice Côte d'Azur for the given data.

Thank you for your time!