A nonlocal boundary condition for domain truncation in frequency-domain Helmholtz problems

Robert Kirby[1] Andreas Klöckner[2] Ben Sepanski[3]
[1]Baylor University
[2]University of Illinois at Urbana-Champaign
[3]University of Texas at Austin
25 March 2021

Helmholtz scattering

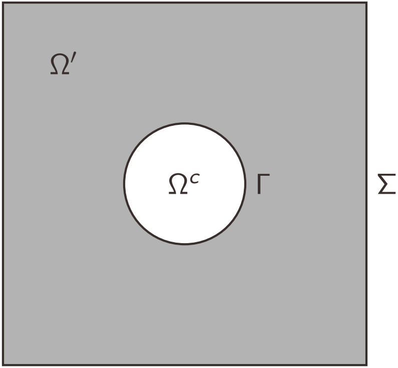
A new boundary condition

Quo vadis, Firedrake? (Future work)

Thanks to...

- NSF 1525697, 1909176
- The U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0021110
- Luke Olson (UIUC)

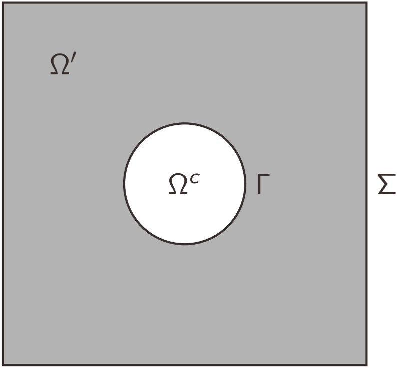
Model problem



Real problem:

$$
\begin{aligned}
-\Delta u + \kappa^2 u &= 0, \quad \mathbb{R}^d \backslash \Omega^c \\
-\frac{\partial u}{\partial n} &= f, \quad \Gamma \\
\lim_{r\to\infty} r^{\frac{d-1}{2}} \left(\tfrac{\partial u}{\partial r} - i\kappa u\right) &= 0,
\end{aligned}
$$

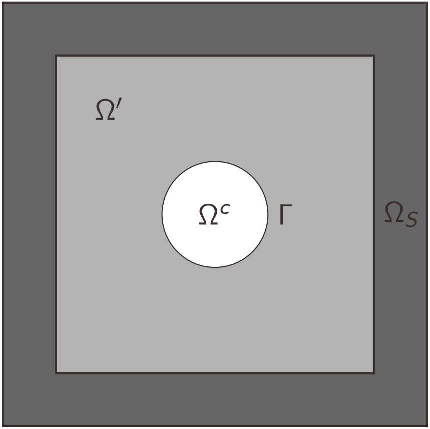
Model problem



Computational problem:

$$
\begin{aligned}
-\Delta u + \kappa^2 u &= 0, \quad \Omega' \\
-\frac{\partial u}{\partial n} &= f, \quad \Gamma \\
?????\quad & \Sigma
\end{aligned}
$$

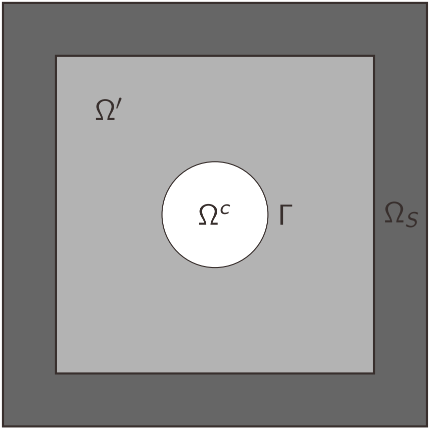
Perfectly Matched Layers (PML)



Computational problem:

$$
\begin{aligned}
-\nabla \cdot \beta(x) \nabla u + \kappa^2 u &= 0, \quad \Omega' \\
-\frac{\partial u}{\partial n} &= f, \quad \Gamma \\
u &= 0, \quad \Sigma
\end{aligned}
$$

Perfectly Matched Layers (PML)

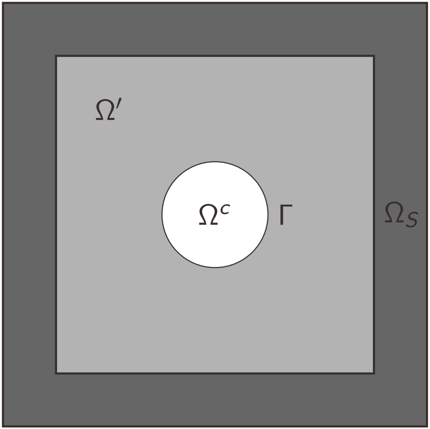


Computational problem:

$$
\begin{aligned}
-\nabla \cdot \beta(x)\nabla u + \kappa^2 u &= 0, \quad \Omega' \\
-\frac{\partial u}{\partial n} &= f, \quad \Gamma \\
u &= 0, \quad \Sigma
\end{aligned}
$$

- $\beta = I$ in Ω'

Perfectly Matched Layers (PML)

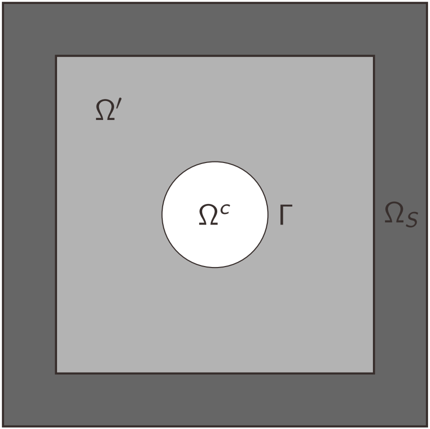


Computational problem:

$$
\begin{aligned}
-\nabla \cdot \beta(x) \nabla u + \kappa^2 u &= 0, \quad \Omega' \\
-\frac{\partial u}{\partial n} &= f, \quad \Gamma \\
u &= 0, \quad \Sigma
\end{aligned}
$$

- $\beta = I$ in Ω'
- β is complex-valued, eats waves in Ω_S

Perfectly Matched Layers (PML)

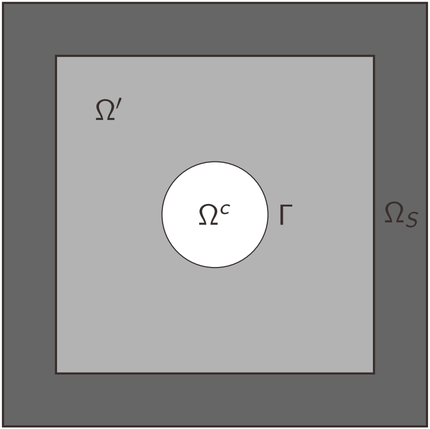


Computational problem:

$$\begin{aligned} -\nabla \cdot \beta(x) \nabla u + \kappa^2 u &= 0, \quad \Omega' \\ -\frac{\partial u}{\partial n} &= f, \quad \Gamma \\ u &= 0, \quad \Sigma \end{aligned}$$

- $\beta = I$ in Ω'
- β is complex-valued, eats waves in Ω_S
- Solution is right in Ω'

Perfectly Matched Layers (PML)



Computational problem:

$$\begin{aligned} -\nabla \cdot \beta(x) \nabla u + \kappa^2 u &= 0, \quad \Omega' \\ -\frac{\partial u}{\partial n} &= f, \quad \Gamma \\ u &= 0, \quad \Sigma \end{aligned}$$

- $\beta = I$ in Ω'
- β is complex-valued, eats waves in Ω_S
- Solution is right in Ω'
- **Solvers are a pain!**

Integral form of the solution

With $\mathcal{K}$ the Green's function, the *true* solution satisfies:

$$u(x) = \int_\Gamma \left(\frac{\partial}{\partial n}\mathcal{K}(x-y)\right) u(y) - \left(\frac{\partial}{\partial n}u(y)\right)\mathcal{K}(x-y)dy,$$

where

Integral form of the solution

With $\mathcal{K}$ the Green's function, the *true* solution satisfies:

$$u(x) = \int_\Gamma \left(\frac{\partial}{\partial n}\mathcal{K}(x-y)\right) u(y) - \left(\frac{\partial}{\partial n} u(y)\right) \mathcal{K}(x-y) dy,$$

where

$$D(u)(x) = \mathrm{PV} \int_\Gamma \left(\frac{\partial}{\partial n}\mathcal{K}(x-y)\right) u(y) dy,$$

Integral form of the solution

With $\mathcal{K}$ the Green's function, the *true* solution satisfies:

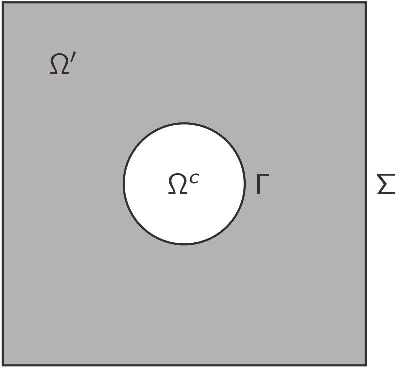
$$u(x) = \int_\Gamma \left(\frac{\partial}{\partial n}\mathcal{K}(x-y)\right) u(y) - \left(\frac{\partial}{\partial n}u(y)\right)\mathcal{K}(x-y)dy,$$

where

$$D(u)(x) = \mathrm{PV}\int_\Gamma \left(\frac{\partial}{\partial n}\mathcal{K}(x-y)\right) u(y)dy,$$

$$S(u)(x) = \int \mathcal{K}(x-y)u(y)dy$$

Model problem



Real problem

$$
\begin{aligned}
-\Delta u + \kappa^2 u &= 0, \quad \Omega' \\
-\frac{\partial u}{\partial n} &= f, \quad \Gamma \\
\left(i\kappa - \tfrac{\partial}{\partial n}\right)\left(u - D(u) + S(f)\right) &= 0, \quad \Sigma
\end{aligned}
$$

Variational form

$$a(u, v) = F(v)$$

for all $v \in H^1(\Omega')$,

Variational form

$$a(u, v) = F(v)$$

for all $v \in H^1(\Omega')$, where

$$a(u, v) = (\nabla u, \nabla v) - \kappa^2 (u, v) - i\kappa \langle u, v \rangle_\Sigma + \left\langle \left(i\kappa - \frac{\partial}{\partial n} \right) D(u), v \right\rangle_\Sigma$$

Variational form

$$a(u,v) = F(v)$$

for all $v \in H^1(\Omega')$, where

$$a(u,v) = (\nabla u, \nabla v) - \kappa^2 \, (u,v) - i\kappa \langle u, v\rangle_\Sigma + \langle \left(i\kappa - \tfrac{\partial}{\partial n}\right) D(u), v\rangle_\Sigma$$

and

$$F(v) = \langle f, v\rangle_\Gamma + \langle \left(i\kappa - \tfrac{\partial}{\partial n}\right) S(f), v\rangle_\Sigma$$

Note

Note

- Not DtN (Keller/Givoli): No Fourier series

Note

- Not DtN (Keller/Givoli): No Fourier series
- Not Poincare-Steklov operator: No singular integrals.

Note

- Not DtN (Keller/Givoli): No Fourier series
- Not Poincare-Steklov operator: No singular integrals.

BUT still requires layer potentials on boundary.

Theory

Theory

- a is a bounded bilinear form on $H^1 \times H^1$

Theory

- a is a bounded bilinear form on $H^1 \times H^1$
- F is a bounded linear functional on H^1

Theory

- a is a bounded bilinear form on $H^1 \times H^1$
- F is a bounded linear functional on H^1
- Gårding inequality. There exist M and an $\alpha > 0$ such that

$$\mathrm{Re}(a(u,u)) + M\|u\|^2 \geq \alpha \|u\|^2_{H^1(\Omega)}.$$

Theory

- a is a bounded bilinear form on $H^1 \times H^1$
- F is a bounded linear functional on H^1
- Gårding inequality. There exist M and an $\alpha > 0$ such that

$$\mathrm{Re}(a(u,u)) + M \|u\|^2 \geq \alpha \|u\|^2_{H^1(\Omega)}.$$

- For $h \leq h_0$, we have optimal-order H^1 and L^2 error estimates.

Linear system $A\mathrm{x} = \mathrm{b}$

Separate out FEM/transmission from layer potential terms:

$$A = A^{L} + A^{NL}$$

Linear system $A\mathrm{x} = \mathrm{b}$

Separate out FEM/transmission from layer potential terms:

$$A = A^L + A^{NL}$$

- A^L is built with Firedrake

Linear system $A\mathrm{x} = \mathrm{b}$

Separate out FEM/transmission from layer potential terms:

$$A = A^L + A^{NL}$$

- A^L is built with Firedrake
- A^{NL} is evaluated matrix-free

Linear system $A\mathrm{x} = \mathrm{b}$

Separate out FEM/transmission from layer potential terms:

$$A = A^{L} + A^{NL}$$

- A^{L} is built with Firedrake
- A^{NL} is evaluated matrix-free
  - Firedrake – pytential marshalling

Linear system $A\mathrm{x} = \mathrm{b}$

Separate out FEM/transmission from layer potential terms:

$$A = A^L + A^{NL}$$

- A^L is built with Firedrake
- A^{NL} is evaluated matrix-free
  - Firedrake – pytential marshalling
  - Pytential constructs layer potential via FMM, evaluates at edge quad points

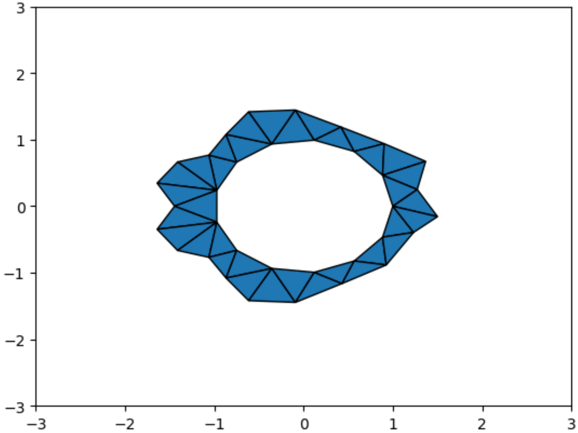
Linear system $A\mathrm{x} = \mathrm{b}$

Separate out FEM/transmission from layer potential terms:

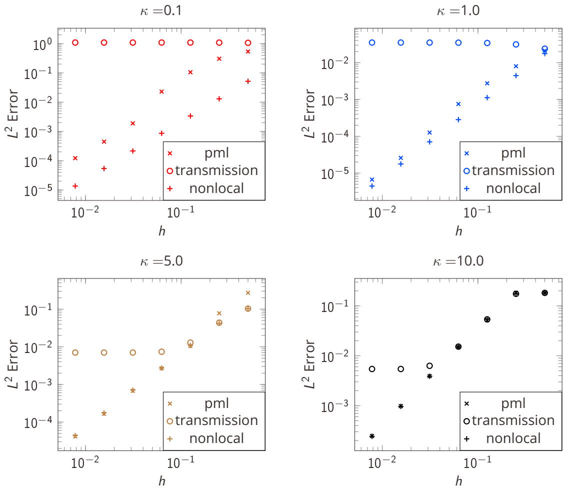
$$A = A^L + A^{NL}$$

- A^L is built with Firedrake
- A^{NL} is evaluated matrix-free
  - Firedrake – pytential marshalling
  - Pytential constructs layer potential via FMM, evaluates at edge quad points
  - Firedrake integrates against test function

Marshalling issue



Numerical results



Preconditioning

Flexible GMRES requires right-preconditioning:

$$A\left(P^{-1}\mathsf{y}\right) = \mathsf{b}; \quad \mathsf{y} = P\mathsf{x}$$

Preconditioning

Flexible GMRES requires right-preconditioning:

$$A\left(P^{-1}\mathsf{y}\right) = \mathsf{b}; \quad \mathsf{y} = P\mathsf{x}$$

Our choice of preconditioner...

Preconditioning

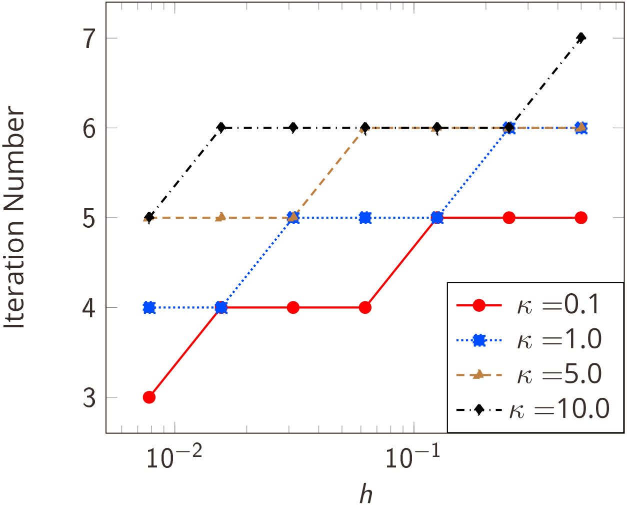
Flexible GMRES requires right-preconditioning:

$$A\left(P^{-1}\mathrm{y}\right)=\mathrm{b};\quad \mathrm{y}=P\mathrm{x}$$

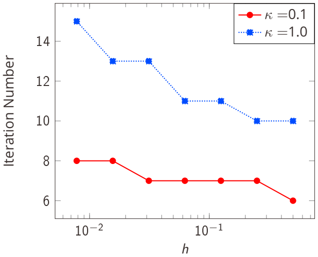
Our choice of preconditioner...

$$P=A^{L}$$

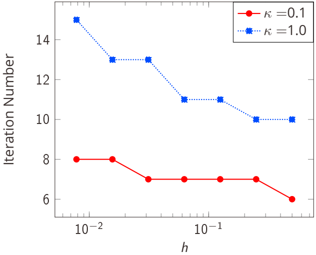
Using LU on A^L



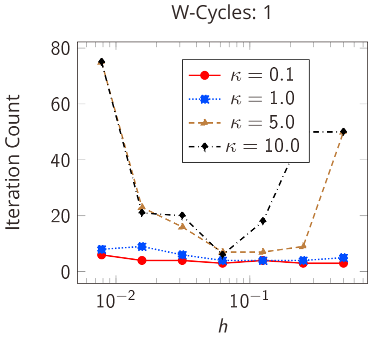
Scalable – gamg?

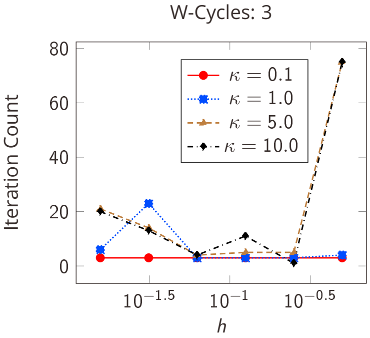


Scalable – gamg?



And explodes for higher κ.


W-Cycles: 1
Iteration Count
80
60
40
20
0
10^{-2}
10^{-1}
h
$\kappa = 0.1$
$\kappa = 1.0$
$\kappa = 5.0$
$\kappa = 10.0$


W-Cycles: 3
Iteration Count
80
60
40
20
0
$10^{-1.5}$
10^{-1}
$10^{-0.5}$
h
$\kappa = 0.1$
$\kappa = 1.0$
$\kappa = 5.0$
$\kappa = 10.0$

Had NUFL of vaporware?

Can we do this?

More generally

More generally

- Extend UFL with nonlocal operators: Layer/volume potentials, fractional derivatives, etc

More generally

- Extend UFL with nonlocal operators: Layer/volume potentials, fractional derivatives, etc
- Compiler should segregate local and nonlocal terms:

More generally

- Extend UFL with nonlocal operators: Layer/volume potentials, fractional derivatives, etc
- Compiler should segregate local and nonlocal terms:
  - Matrix free, hybrid, HSS, etc

More generally

- Extend UFL with nonlocal operators: Layer/volume potentials, fractional derivatives, etc
- Compiler should segregate local and nonlocal terms:
  - Matrix free, hybrid, HSS, etc
  - TSFC + pytential/sumpy/loopy

More generally

- Extend UFL with nonlocal operators: Layer/volume potentials, fractional derivatives, etc
- Compiler should segregate local and nonlocal terms:
  - Matrix free, hybrid, HSS, etc
  - TSFC + pytential/sumpy/loopy
- Extend solver infrastructure