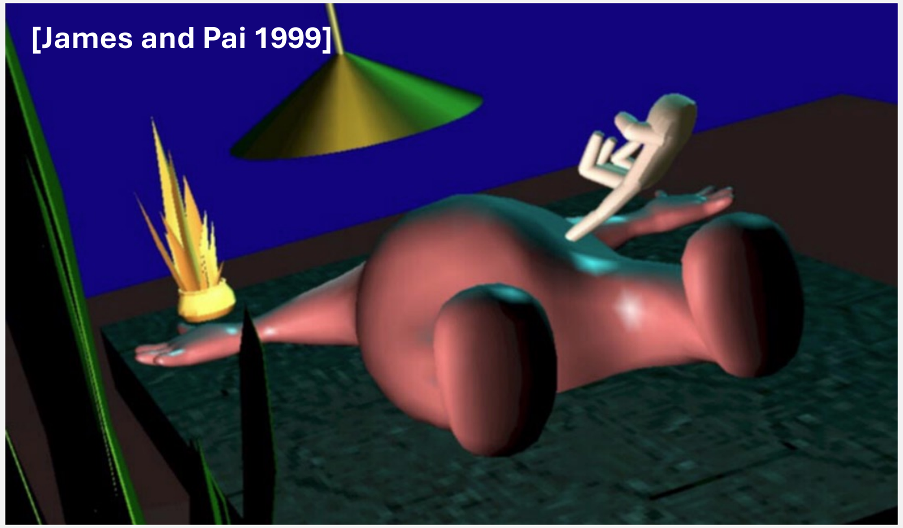
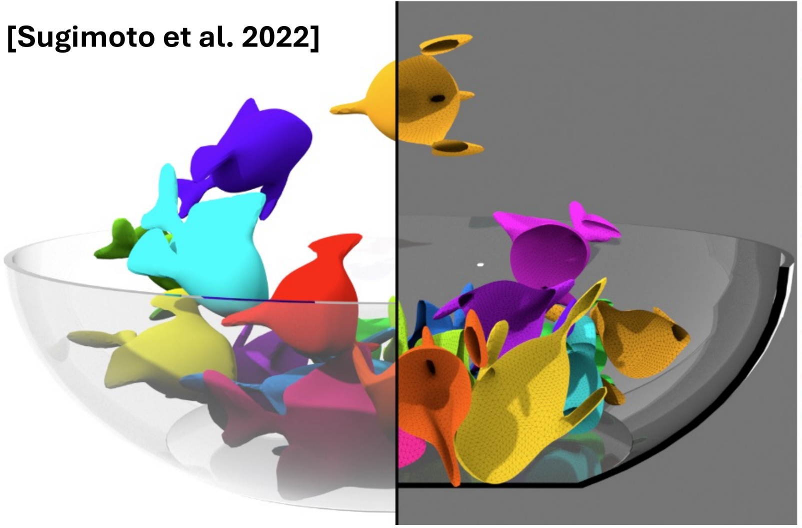
SIGGRAPH 2025

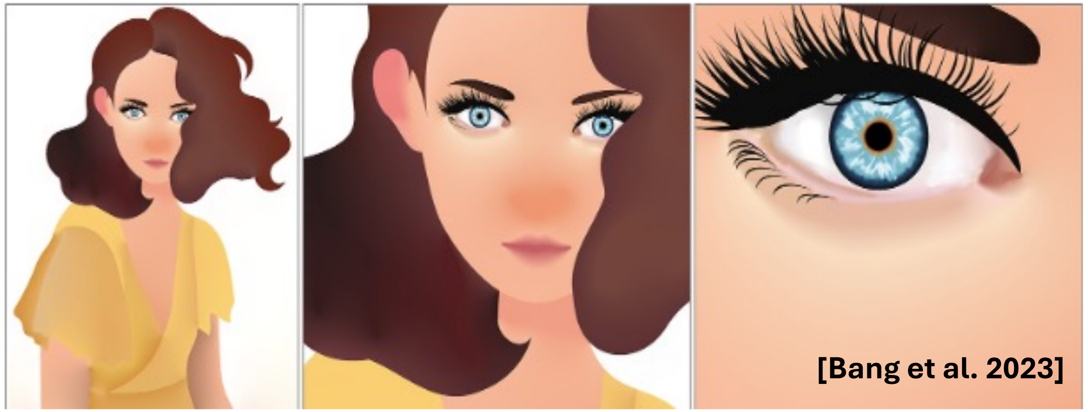
LIGHTNING-FAST BOUNDARY ELEMENT METHODS

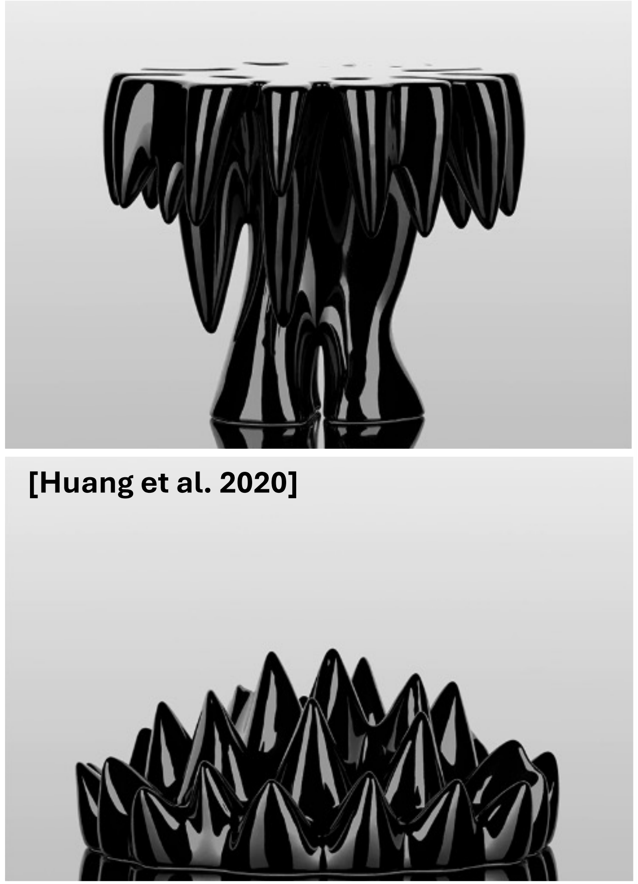
JIONG CHEN — Inria
FLORIAN SCHÄFER — Georgia Tech
MATHIEU DESBRUN — Inria, École Polytechnique, IP Paris


[James and Pai 1999]


[Sugimoto et al. 2022]


[Xia et al. 2020]


[Bang et al. 2023]


[Huang et al. 2020]


[Umetani et al. 2016]

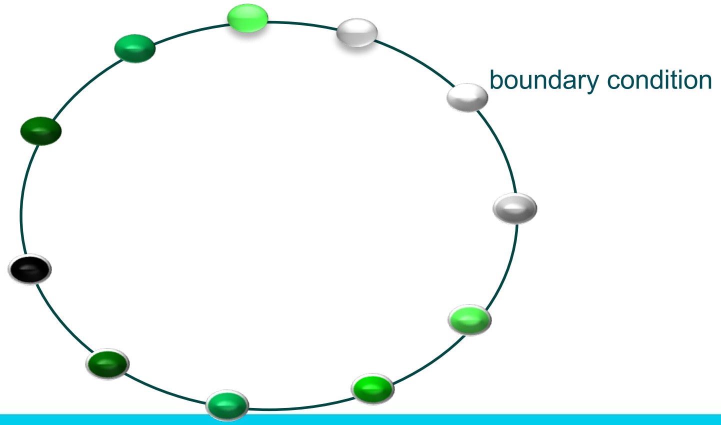

A
B
C
[Da et al. 2016]

RECAP BEM

- Boundary Element Method (BEM)
  - Turn volumetric differential equations into **boundary integral equations (BIE)**
  - No need for volumetric tessellation, slower growth of the problem size
  - Works for infinite large domains

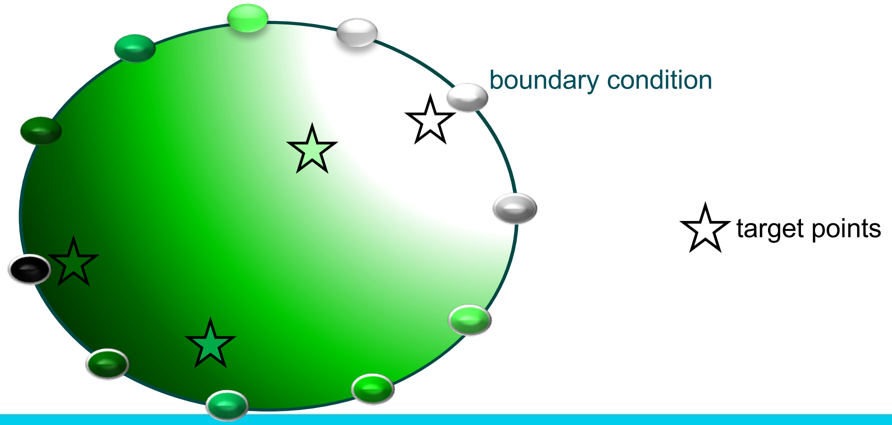
RECAP BEM

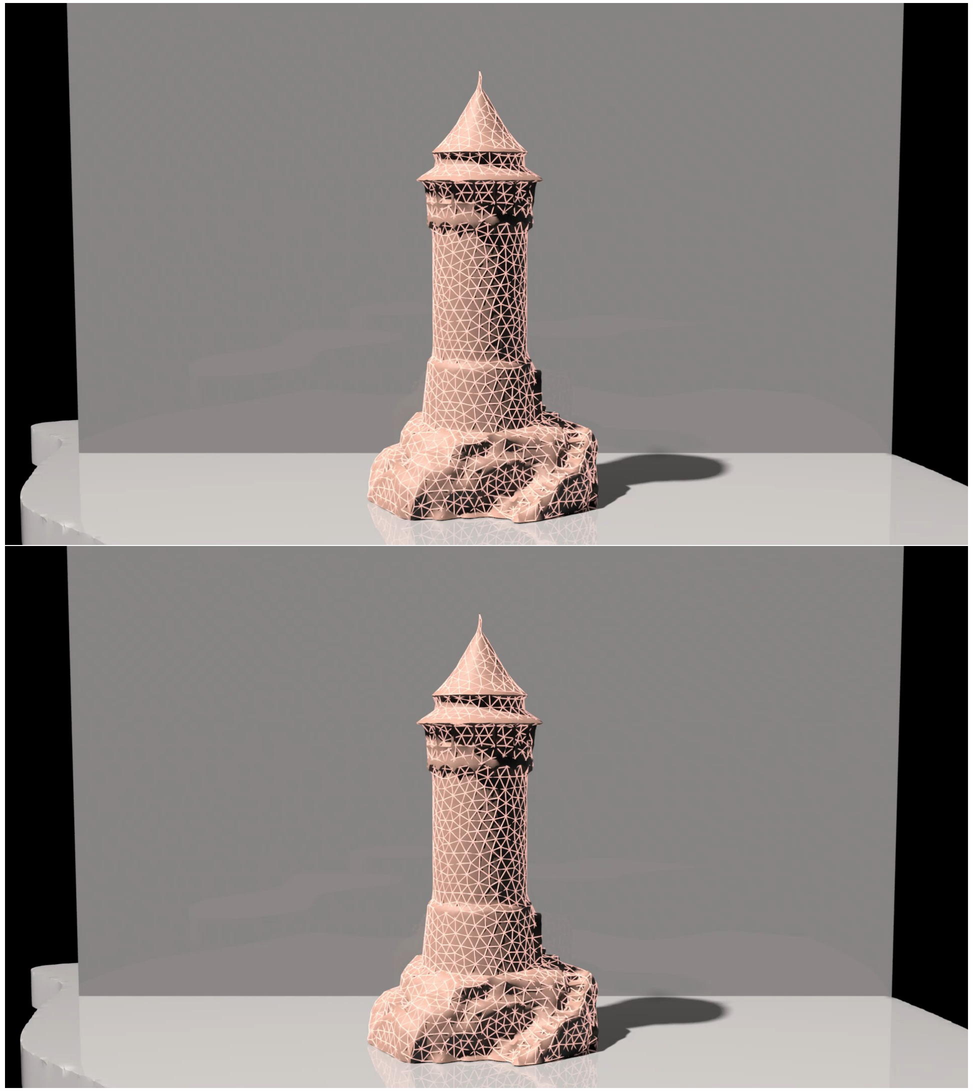
- Boundary Element Method (BEM)
  - Turn volumetric differential equations into **boundary integral equations (BIE)**
  - No need for volumetric tessellation, slower growth of the problem size
  - Works for infinite large domains
- Two stages of BEM
  - **SOLVE** for **unknown** boundary data from **given** boundary conditions
    - E.g., boundary charges producing an electric potential field



RECAP BEM

- Boundary Element Method (BEM)
  - Turn volumetric differential equations into **boundary integral equations (BIE)**
  - No need for volumetric tessellation, slower growth of the problem size
  - Works for infinite large domains
- Two stages of BEM
  - **SOLVE** for **unknown** boundary data from **given** boundary conditions
    - E.g., boundary charges producing an electric potential field
  - **INTERPOLATE / EXTRAPOLATE** the solution at arbitrary target points from boundary data





BOTTLENECK: FINDING BOUNDARY DATA

PDE

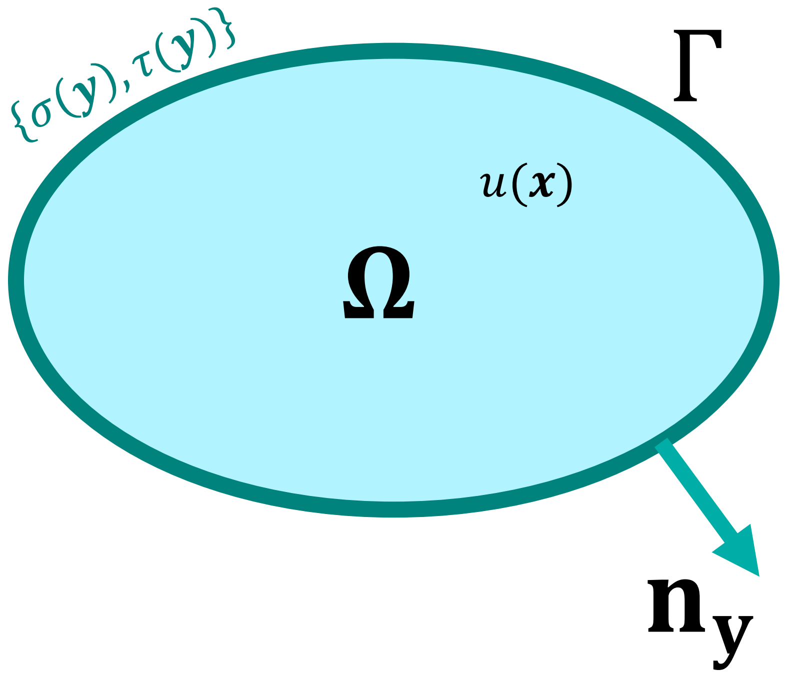
$$\mathbf{x} \in \Omega, \quad \Delta u = 0$$

Representation of the solution

$$\mathbf{x} \in \mathbb{R}^d \backslash \Gamma, \; u(\mathbf{x}) = \underbrace{\int_\Gamma \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}_\mathbf{y}} \sigma(\mathbf{y}) \, \mathrm{d}A_\mathbf{y}}_{\text{Double-layer potential}} - \underbrace{\int_\Gamma G(\mathbf{x}, \mathbf{y}) \tau(\mathbf{y}) \, \mathrm{d}A_\mathbf{y}}_{\text{Single-layer potential}}$$

BIE

$$\mathbf{x} \in \Gamma, \; u(\mathbf{x}) = b(\mathbf{x}) \text{ or } \partial_\mathbf{n} u(\mathbf{x}) = g(\mathbf{x})$$



BOTTLENECK: FINDING BOUNDARY DATA

PDE

$$\mathbf{x} \in \Omega, \quad \Delta u = 0$$

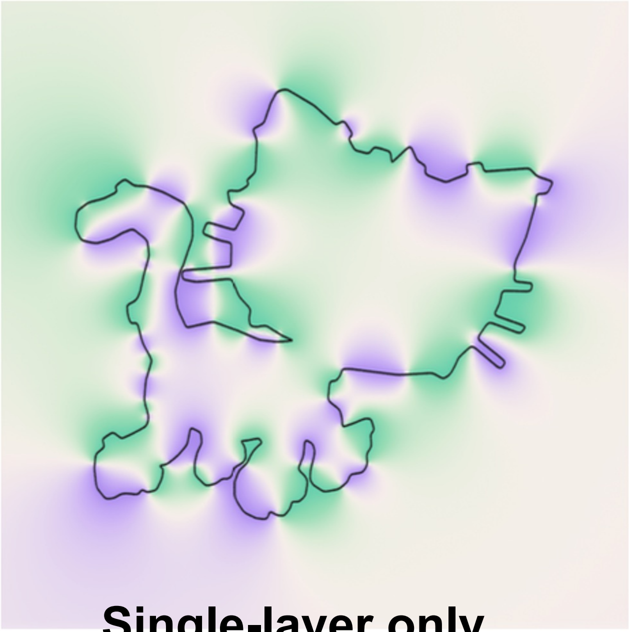
Representation of the solution

$$\mathbf{x} \in \mathbb{R}^d \backslash \Gamma, \; u(\mathbf{x}) = \underbrace{\int_\Gamma \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}_\mathbf{y}} \sigma(\mathbf{y}) \, \mathrm{d}A_\mathbf{y}}_{\text{Double-layer potential}} - \underbrace{\int_\Gamma G(\mathbf{x},\mathbf{y}) \tau(\mathbf{y}) \, \mathrm{d}A_\mathbf{y}}_{\text{Single-layer potential}}$$

BIE

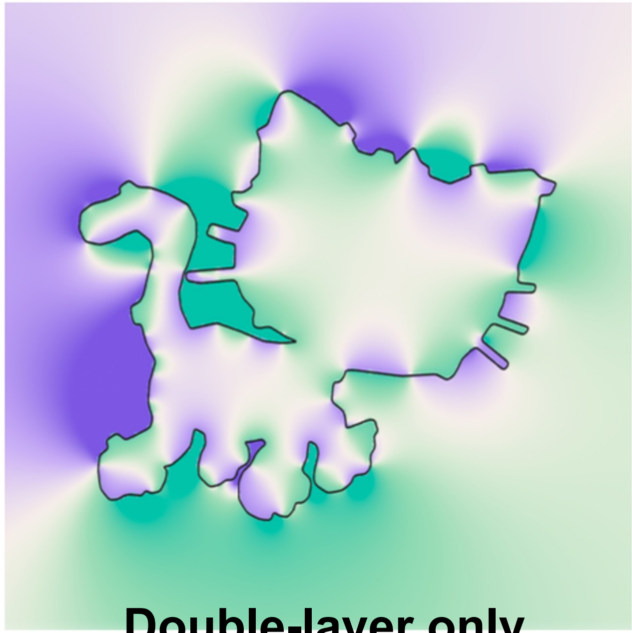
$$\mathbf{x} \in \Gamma, \; u(\mathbf{x}) = b(\mathbf{x}) \text{ or } \partial_\mathbf{n} u(\mathbf{x}) = g(\mathbf{x})$$

$$[u(\mathbf{x})]_\Gamma = 0$$



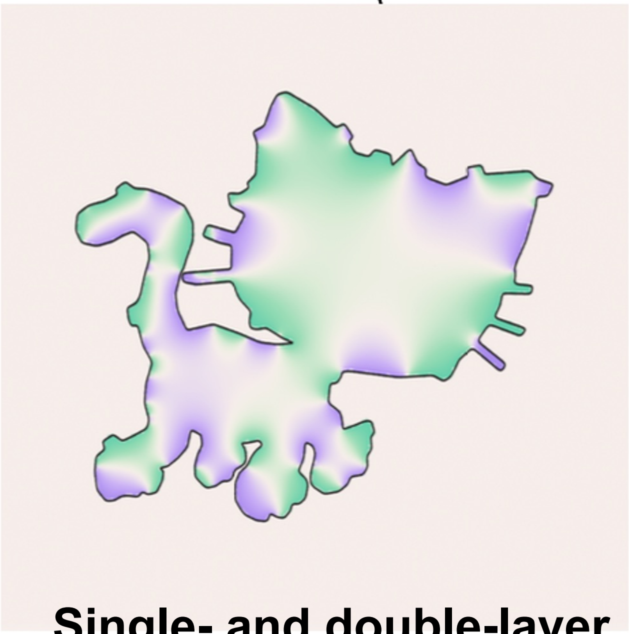
Single-layer only

$$[u(\mathbf{x})]_\Gamma = \sigma(\mathbf{x})$$



Double-layer only

$$u(\mathbf{x})|_{\mathbf{x} \in \mathbb{R}^d \backslash \Omega} = 0$$



Single- and double-layer

NUMERICAL CHALLENGES OF SOLVING BIES

BIE

$\mathbf{x} \in \Gamma,\ u(\mathbf{x}) = b(\mathbf{x})$ or $\partial_{\mathbf{n}} u(\mathbf{x}) = g(\mathbf{x})$

Linear system

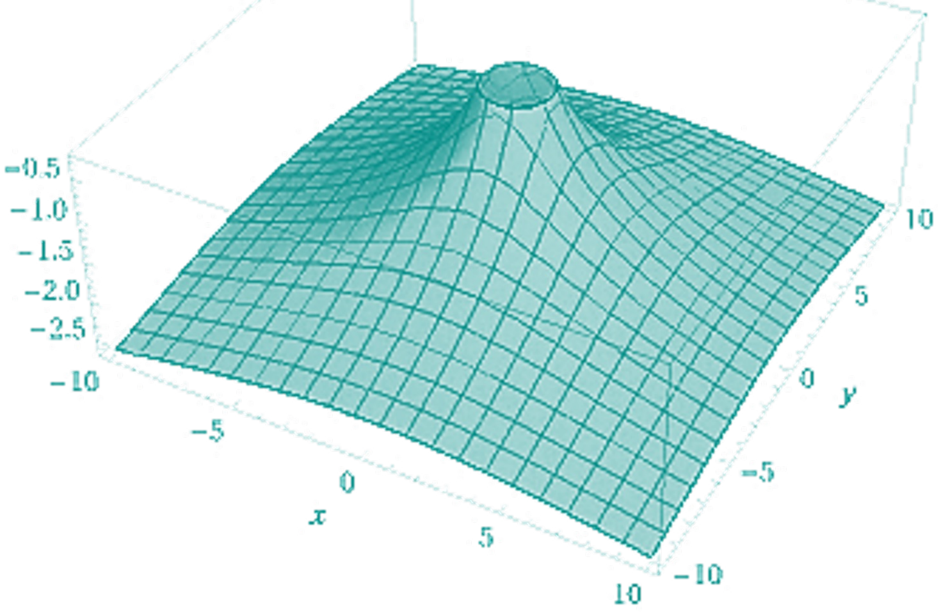
$$\boldsymbol{Ks} = \boldsymbol{b}$$

- **The linear system is always dense**
  - Green's functions have non-zero values everywhere
  - Storing the entire system matrix is impossible for big problems
    - 70G for 100k boundary samples; assembly time is large too!
  - Direct solvers have cubic complexity
- **The linear system is often ill-conditioned***
  - High-frequency vibrations in σ get smoothed out after integration
    - So very different σ's map to similar b, meaning that the BIE is almost degenerate
  - Iterative solvers often struggle to converge
    - multigrid approaches too memory hungry, H-matrices too inaccurate

In practice, BIE of ~25K unknowns in recent graphics papers…

There has to be a better way…

Main culprit:
smoothness of the Green's function



*Fredholm integral equation of the first kind

$$\mathbf{x} \in \Gamma,\ \int_{\Gamma} G(\mathbf{x},\mathbf{y})\sigma(\mathbf{y})\ \mathrm{d}A_{\mathbf{y}} = b(\mathbf{x})$$

SYMMETRIC CASE: INVERSE CHOLESKY FACTORIZATION

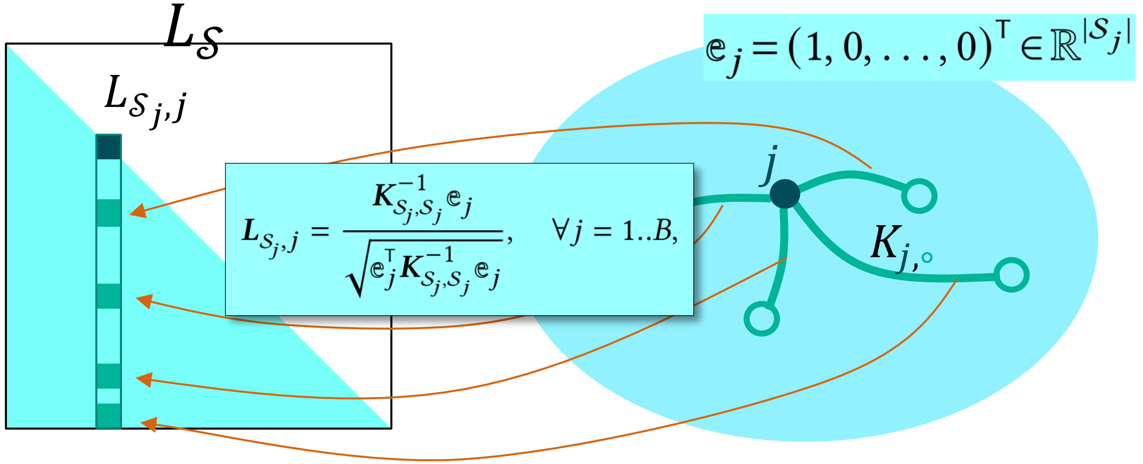
SIGGRAPH 2025
Vancouver+ 10-14 August

- [Chen et al. 2024] computed inverse Cholesky factors to accelerate PCG

$$Ks = b \Rightarrow \boxed{K^{-1} \approx L_S L_S^T} \Rightarrow s \approx L_S L_S^T b$$

- Kaporin's construction for L_S [Kaporin 1994]

$$L_{S_j,j} = \frac{K^{-1}_{S_j,S_j} \mathbb{e}_j}{\sqrt{\mathbb{e}_j^\mathsf{T} K^{-1}_{S_j,S_j} \mathbb{e}_j}}, \qquad \forall j = 1..B,$$



- Properties
  - **Massively parallel:** each column of L_S is computed independently of others. Perfect for GPUs!
  - **Memory efficient:** no need to assemble the global BIE matrix.
  - **Stable:** no breakdowns will occur
  - **Variational interpretation(s):** minimizing Kaporin's condition number*, KL-divergence, and a constrained quadratic form

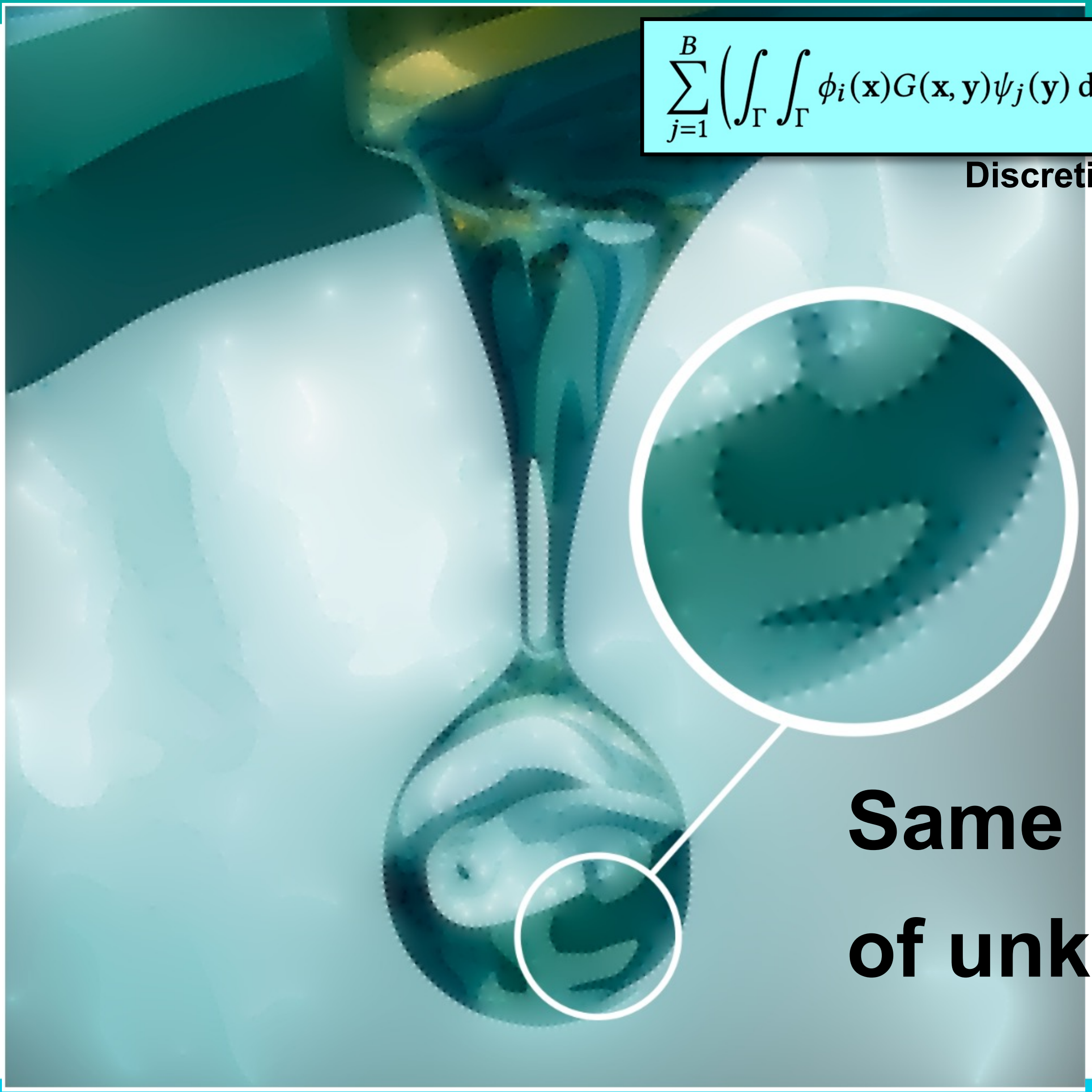
$$* \quad \kappa_{\mathrm{Kap}}(M) = \frac{1}{B} \frac{\mathrm{tr}(M)}{\det(M)^{1/B}}$$

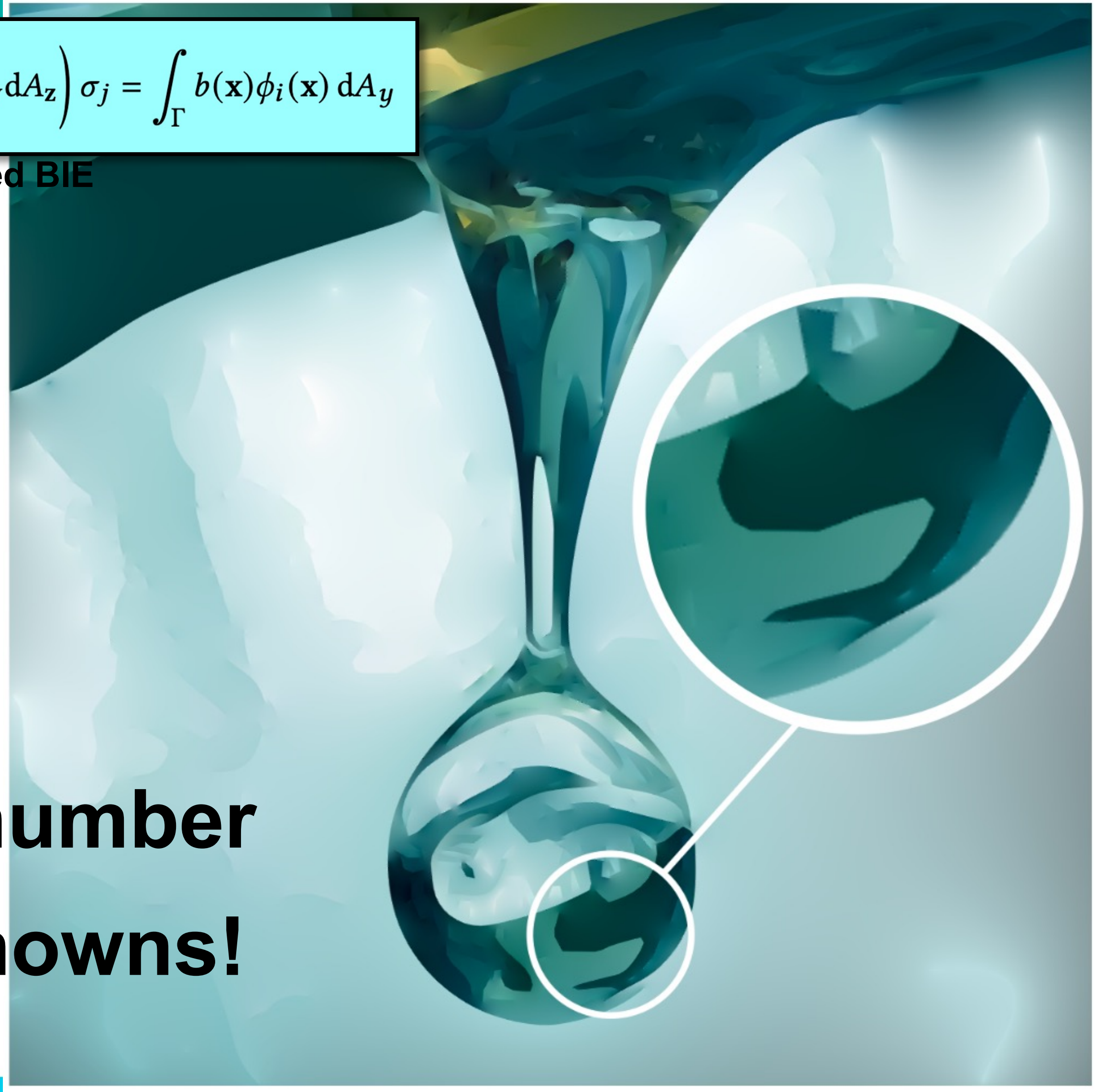
Last year: $\boldsymbol{\phi_i = \psi_i = \delta(x - x_i)}$
Symmetric, meshless approach

This year: $\boldsymbol{\phi_i \neq \psi_i}$
Asymmetric, more general approach

$$\sum_{j=1}^{B}\left(\int_\Gamma\int_\Gamma \phi_i(\mathbf{x})G(\mathbf{x},\mathbf{y})\psi_j(\mathbf{y})\,\mathrm{d}A_\mathbf{y}\mathrm{d}A_\mathbf{z}\right)\sigma_j = \int_\Gamma b(\mathbf{x})\phi_i(\mathbf{x})\,\mathrm{d}A_y$$

Discretized BIE





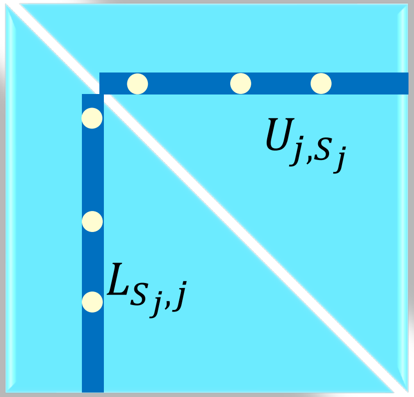
Same number of unknowns!

ASYMMETRIC CASE: INVERSE LU FACTORIZATION

- Solve the least-squares problem $\boldsymbol{K}^T\boldsymbol{K}\boldsymbol{s} = \boldsymbol{K}^T\boldsymbol{b}$?
- We leverage an inverse LU factorization to precondition BIE matrices

$$Ks = b \Leftarrow \boxed{K^{-1} \approx L_S U_S} \Rightarrow s \approx L_S U_S b$$

- Generalizing Kaporin's construction



$$\begin{cases} \mathbf{L}_{S_j,j} = \dfrac{\mathbf{G}_{S_j,S_j}^{-1}\mathbb{e}_j}{\mathbb{e}_j^{\mathsf{T}}\mathbf{G}_{S_j,S_j}^{-1}\mathbb{e}_j}, \\ \mathbf{U}_{j,S_j}^{\mathsf{T}} = \mathbf{G}_{S_j,S_j}^{-\mathsf{T}}\mathbb{e}_j, \end{cases}$$

Forgoing symmetry opens the door to a variety of BIEs with diverse discretization choices.

REORDERING & SPARSITY PATTERN

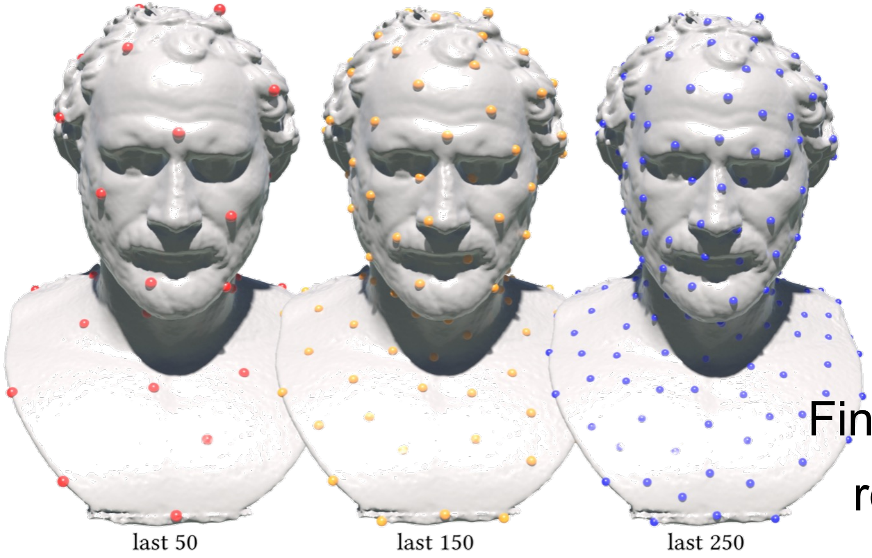
SIGGRAPH 2025
Vancouver+ 10-14 August

REORDERING

- Goal: **evenly** distributing point samples
  - Farthest point sampling, i.e., coarse-to-fine

$$i_k = \underset{q}{\operatorname{argmax}} \min_{p \in \{0,k-1\}} \operatorname{dist}(\boldsymbol{y}_q, \boldsymbol{y}_{i_p}),$$

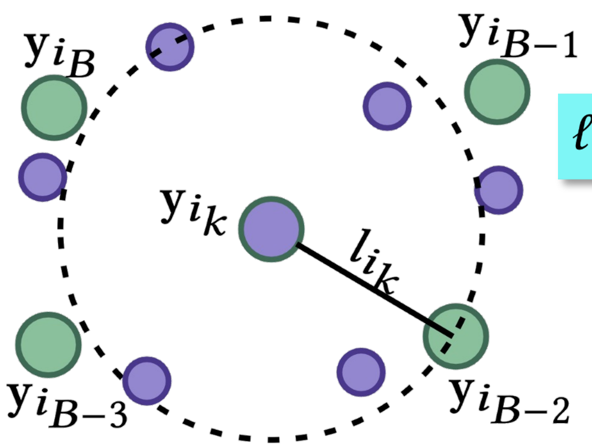
  - Reverse it $\boldsymbol{P} = \{i_{B-1}, \ldots, i_1, i_0\}$, i.e., fine-to-coarse



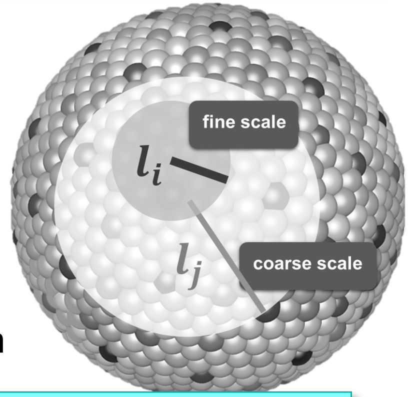
Fine-to-coarse reordering

SPARSITY PATTERN

- Capturing those **"important"** nonzero fill-ins
  - Length scale returned in coarse-to-fine ordering



$$\ell_{i_k} = \min_{p \in \{0,k-1\}} \operatorname{dist}(\boldsymbol{y}_{i_k}, \boldsymbol{y}_{i_p})$$

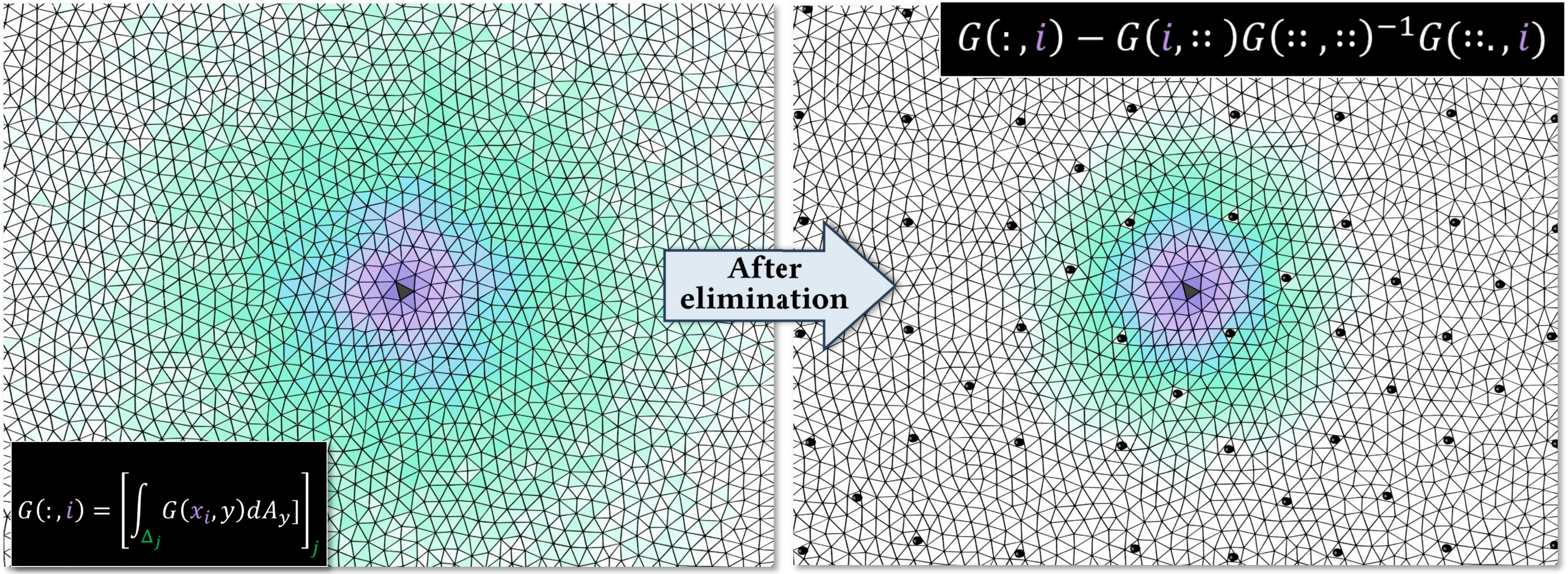


  - Lower-triangular, multiscale sparsity pattern

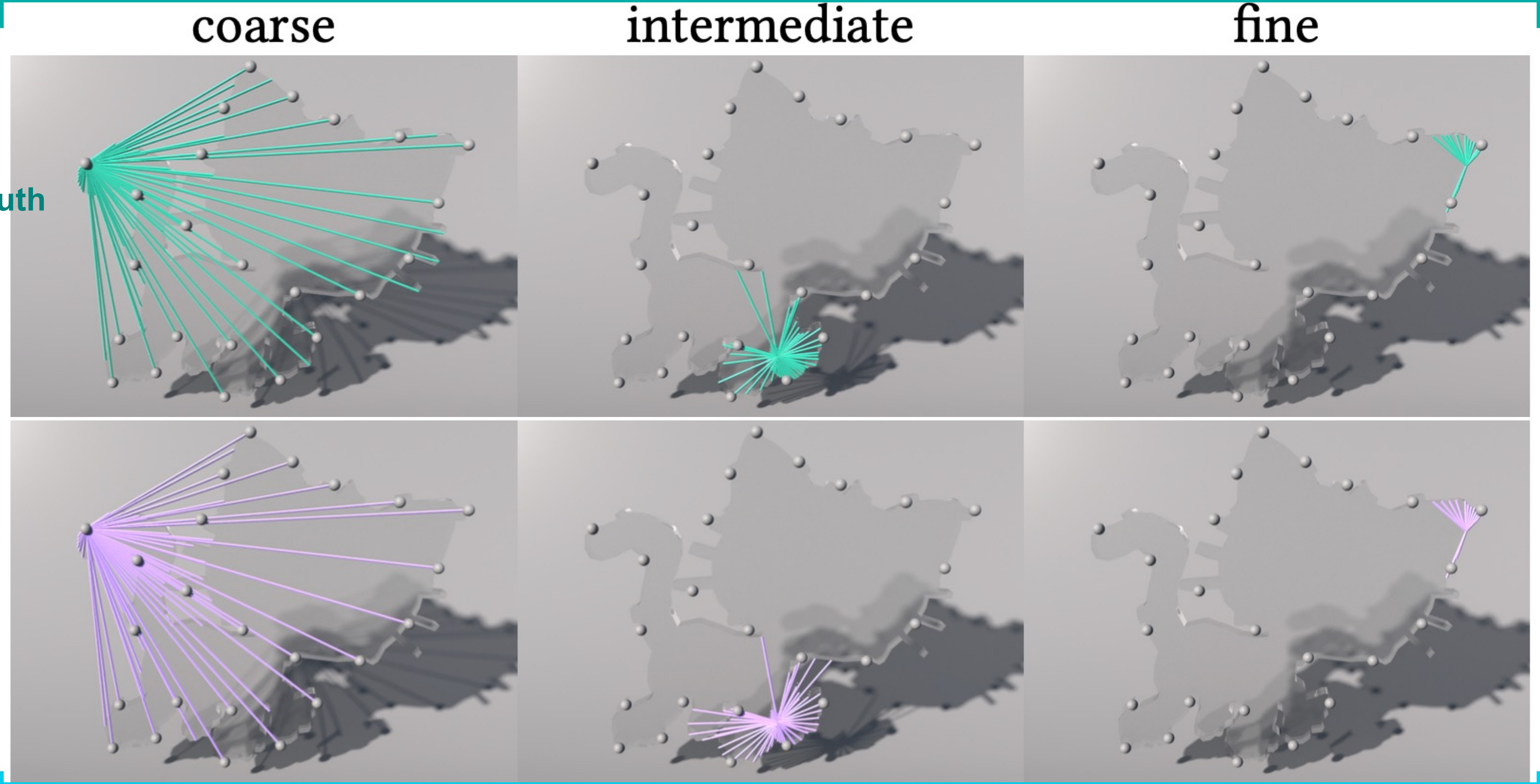
$$S := \{(i,j) \mid i \geq j \text{ and } \operatorname{dist}(x_i, x_j) \leq \rho \min(\ell_i, \ell_j)\}$$

THE BASIS OF EFFECTIVE SPARSITY: SCREENING EFFECT

- Statistical description of the **screening effect**
  - A stochastic process with smooth kernels implies long-range correlations between point samples
  - Conditioning a smooth process on values near a target point weakens the target's correlation with more distant points



PROOF OF CONCEPT

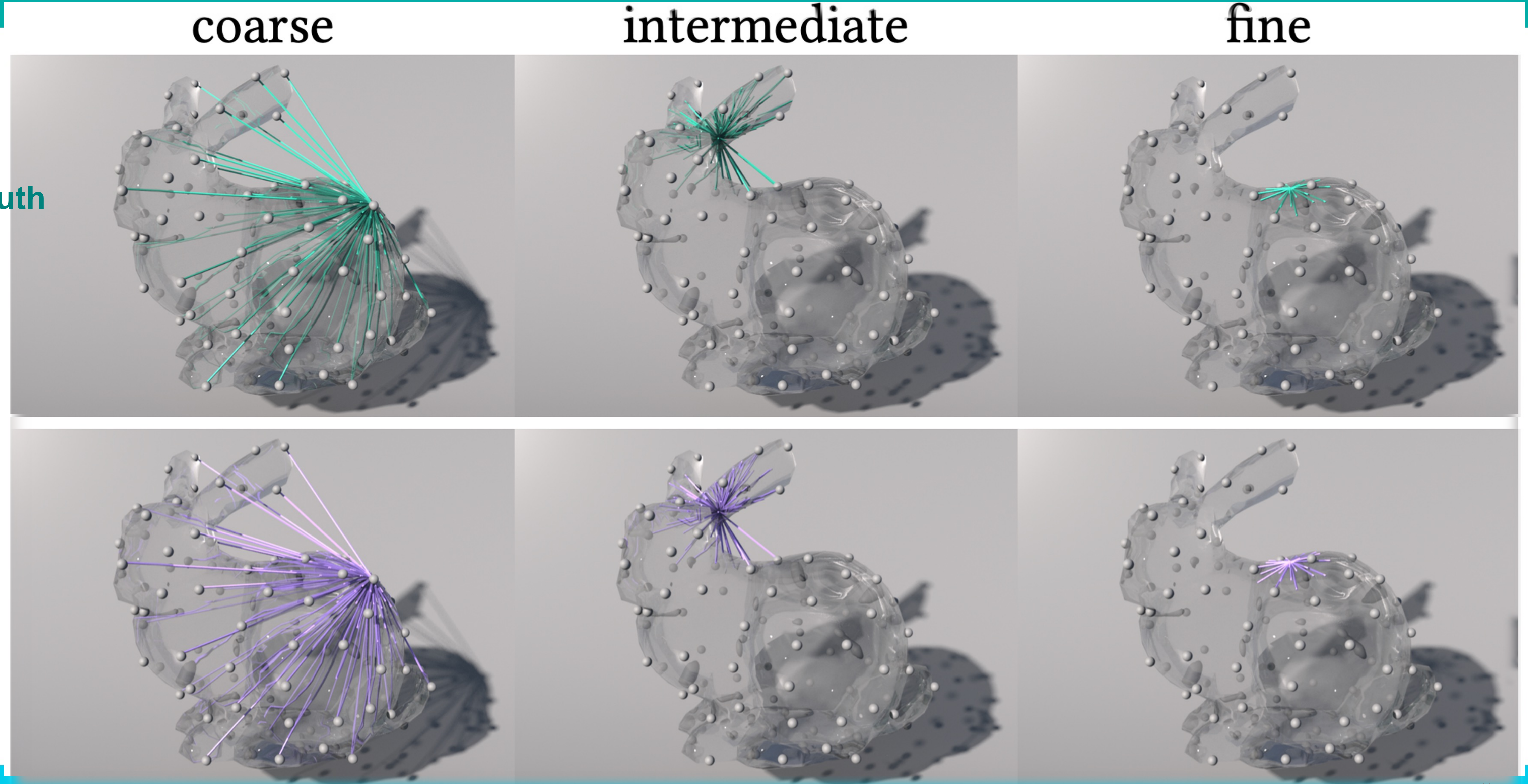


Ground-truth pattern

Max-min pattern

PROOF OF CONCEPT

SIGGRAPH 2025
Vancouver+ 10-14 August



Ground-truth pattern

Max-min pattern

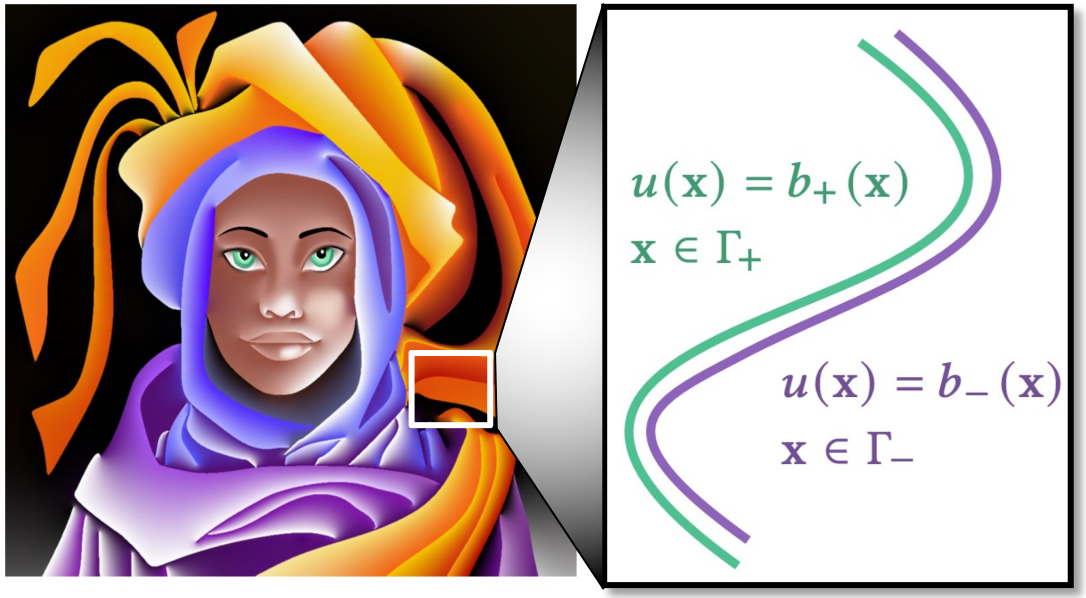
AN INTERESTING PARADOX

- Smoothness of the Green's function responsible for all the numerical challenges
- … but also key to solve these problems
  - because the information provided by nearby points renders that of distant points redundant
  - proper reordering disentangles the complex correlations between points

RESULTS

DIFFUSION CURVES

[Orzan et al. 2008]



Solution representation

$$\mathbf{x} \in \mathbb{R}^2 \backslash \Gamma, \quad u(\mathbf{x}) = \int_\Gamma G(\mathbf{x}, \mathbf{y})\sigma(\mathbf{y})\mathrm{d}A_{\mathbf{y}}.$$

BIE

$$\mathbf{x} \in \Gamma, \quad \int_\Gamma G(\mathbf{x}, \mathbf{y})\sigma(\mathbf{y})\, \mathrm{d}A_{\mathbf{y}} = b(\mathbf{x}),$$

Fredholm integral equation of the first kind

- Diffusing van Gogh's *"Irises"*
  - **6.6M** boundary elements
  - **64M** pixels in total
- Our inverse LU precond.
  - **20 iterations** to reach error below **0.001**
  - Cost **15 mins**
- Jacobi precond.
  - **2.1 days** to reach the same level of error, **200x** slower

MAGNETOSTATICS

[Ni et al. 2024]

$$\begin{cases} \nabla \cdot \mu_0 (\mathbf{H}_\Omega + \mathbf{M}) = 0, \\ \nabla \times \mathbf{H}_\Omega = 0, \end{cases}$$

$$\mathbf{H}_\Omega(\mathbf{x}) = -\nabla u(\mathbf{x})$$

Solution representation

$$\boxed{u(\mathbf{x}) = \int_\Gamma G(\mathbf{x}, \mathbf{y}) \sigma(\mathbf{y}) \mathrm{d}A_\mathbf{y}.}$$

BIE

$$\mathbf{x} \in \Gamma, \quad \frac{2+\chi}{2\chi} \boxed{\sigma(\mathbf{x})} + \int_\Gamma \boxed{\frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}_\mathbf{x}}} \sigma(\mathbf{y}) \, \mathrm{d}A_\mathbf{y} = \mathbf{H}_{\mathrm{ext}} \cdot \mathbf{n}.$$

Fredholm integral equation of the second kind

BIE

$$\mathbf{x} \in \Gamma, \quad \int_\Gamma G(\mathbf{x}, \mathbf{y}) \sigma(\mathbf{y}) \, \mathrm{d}A_\mathbf{y} = b(\mathbf{x}),$$

Fredholm integral equation of the first kind

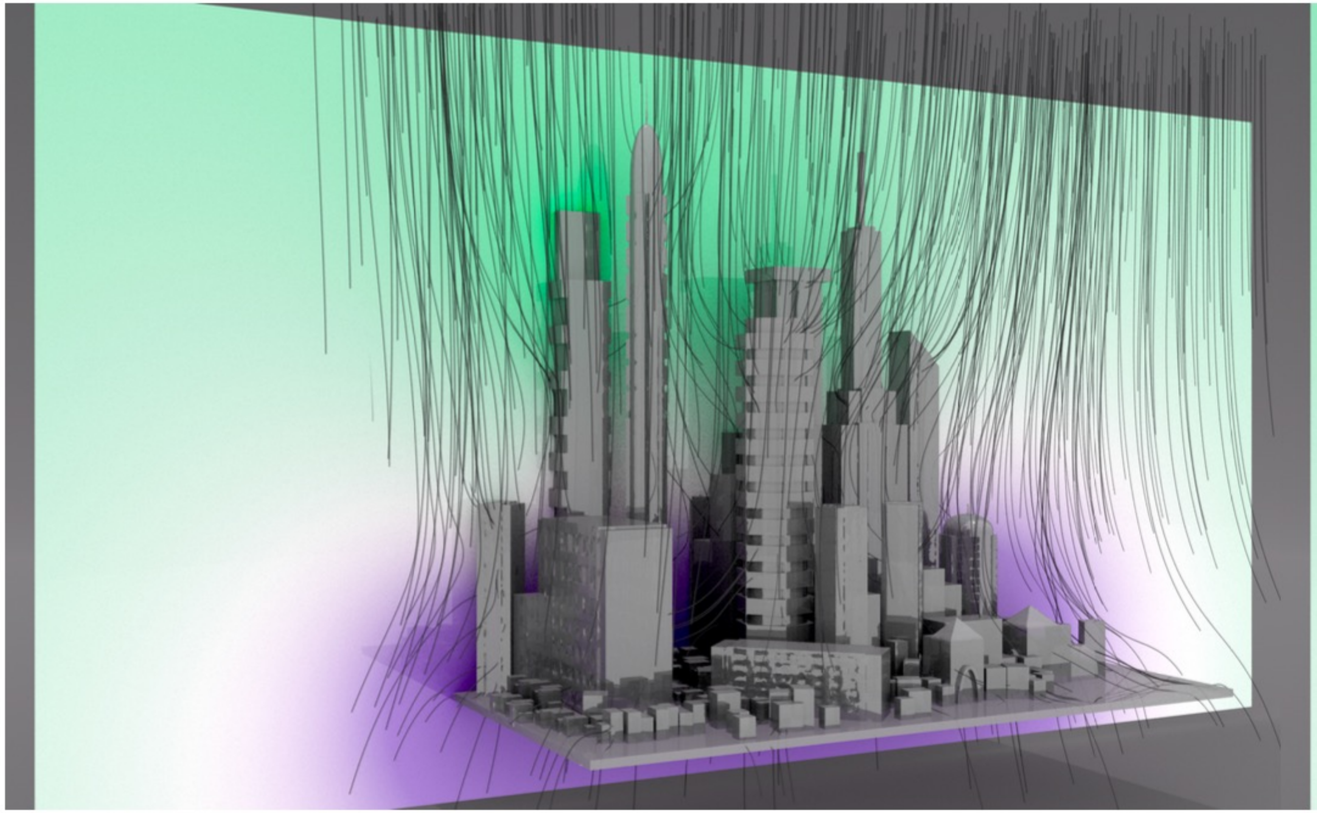
MAGNETOSTATICS ON NON-SMOOTH GEOMETRY

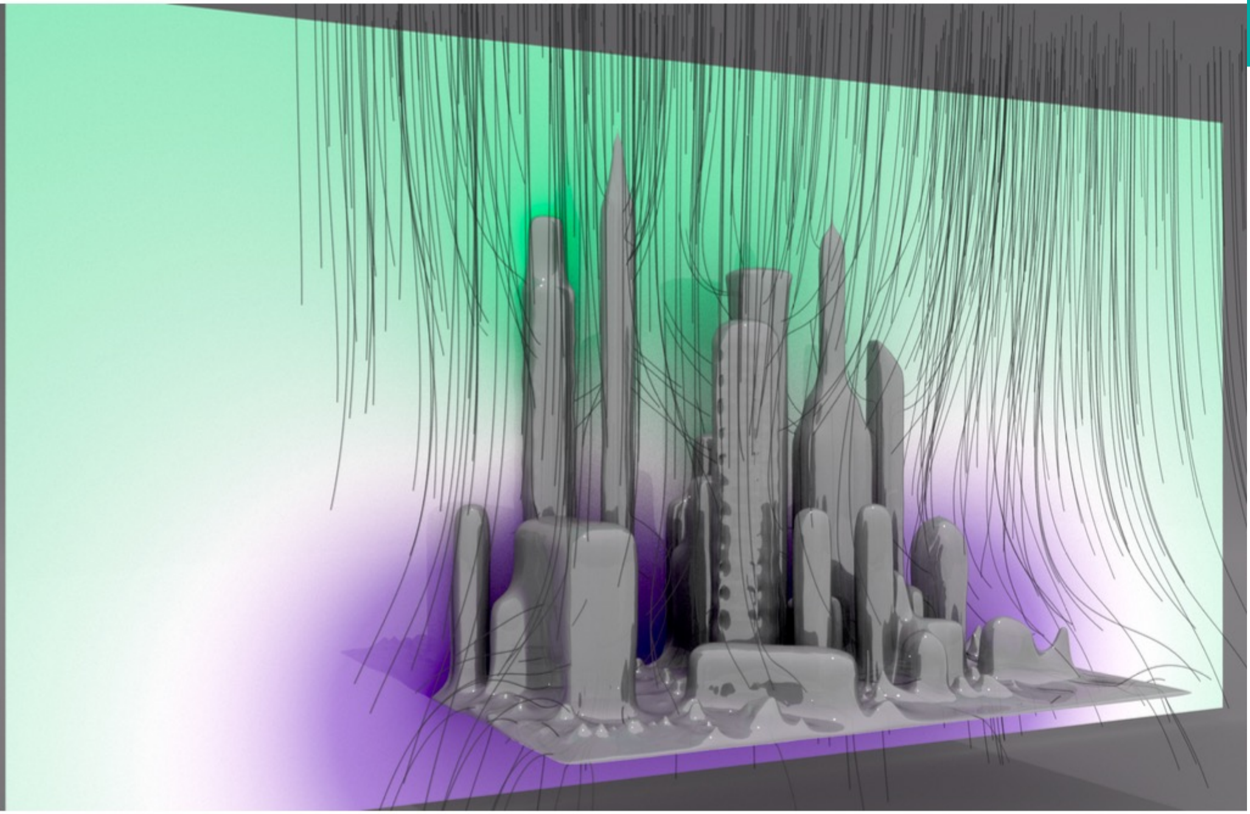
CAREFUL

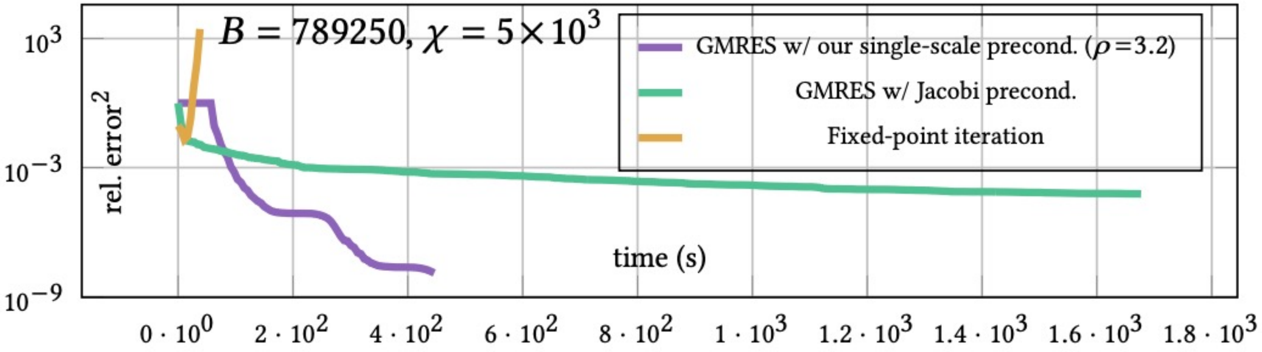
Screening effect

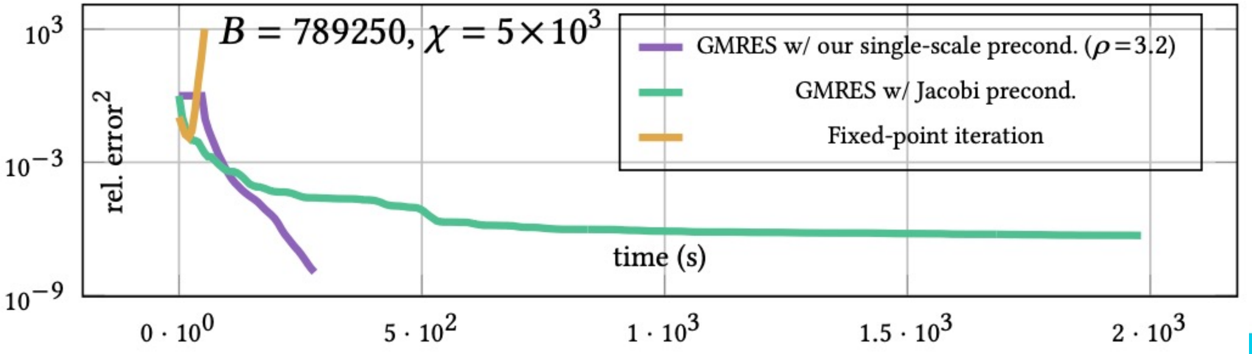
much weaker!!

MAGNETOSTATICS ON NON-SMOOTH GEOMETRY

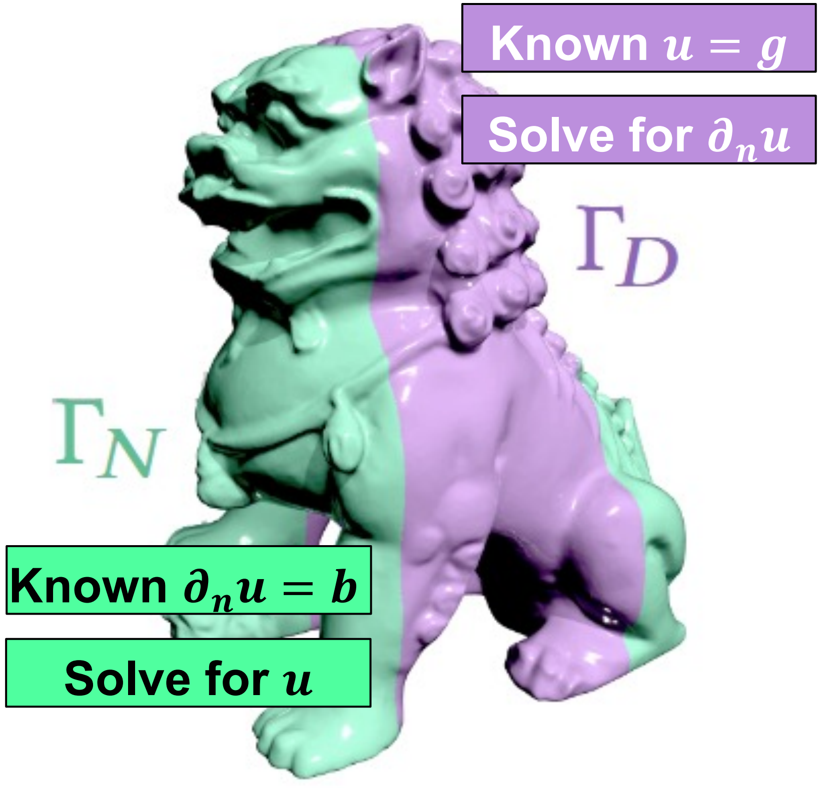








MIXED BOUNDARY CONDITION



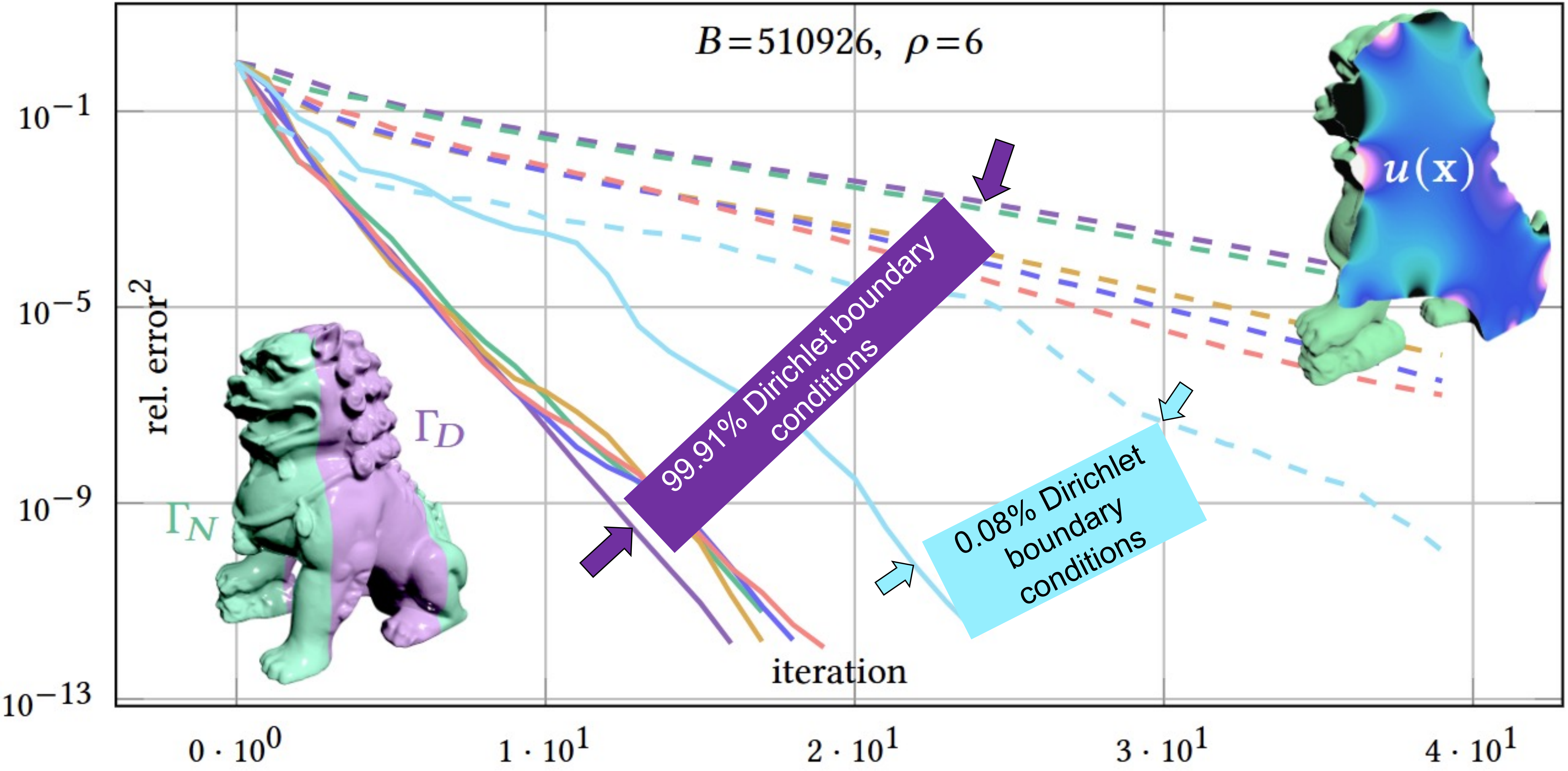
Solution representation

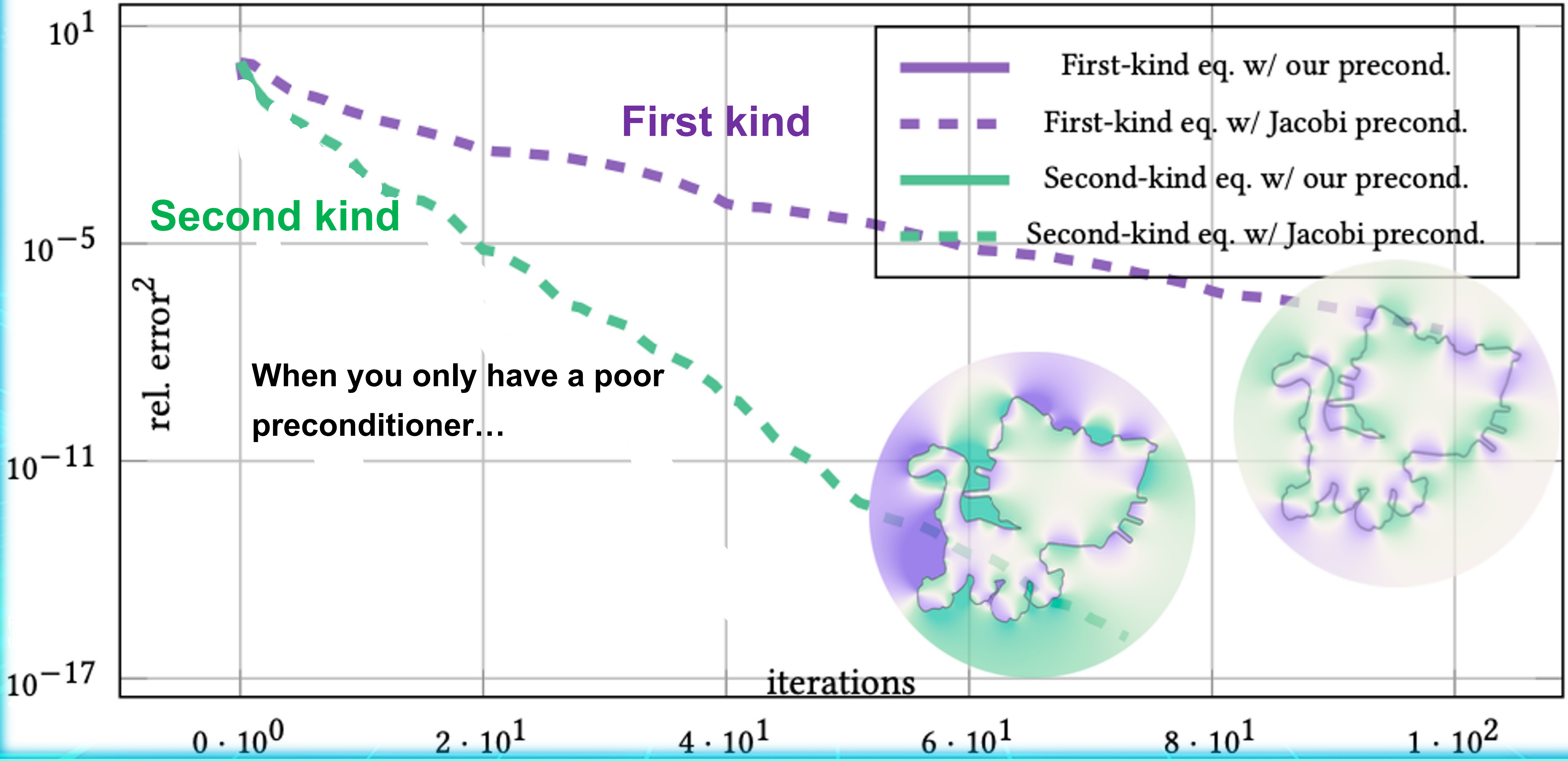
$$u(\mathbf{x}) = -\int_{\Gamma} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}_{\mathbf{y}}} u(\mathbf{y})\, \mathrm{d}A_{\mathbf{y}} + \int_{\Gamma} G(\mathbf{x},\mathbf{y}) \frac{\partial u(\mathbf{y})}{\partial \mathbf{n}_{\mathbf{y}}}\, \mathrm{d}A_{\mathbf{y}}$$

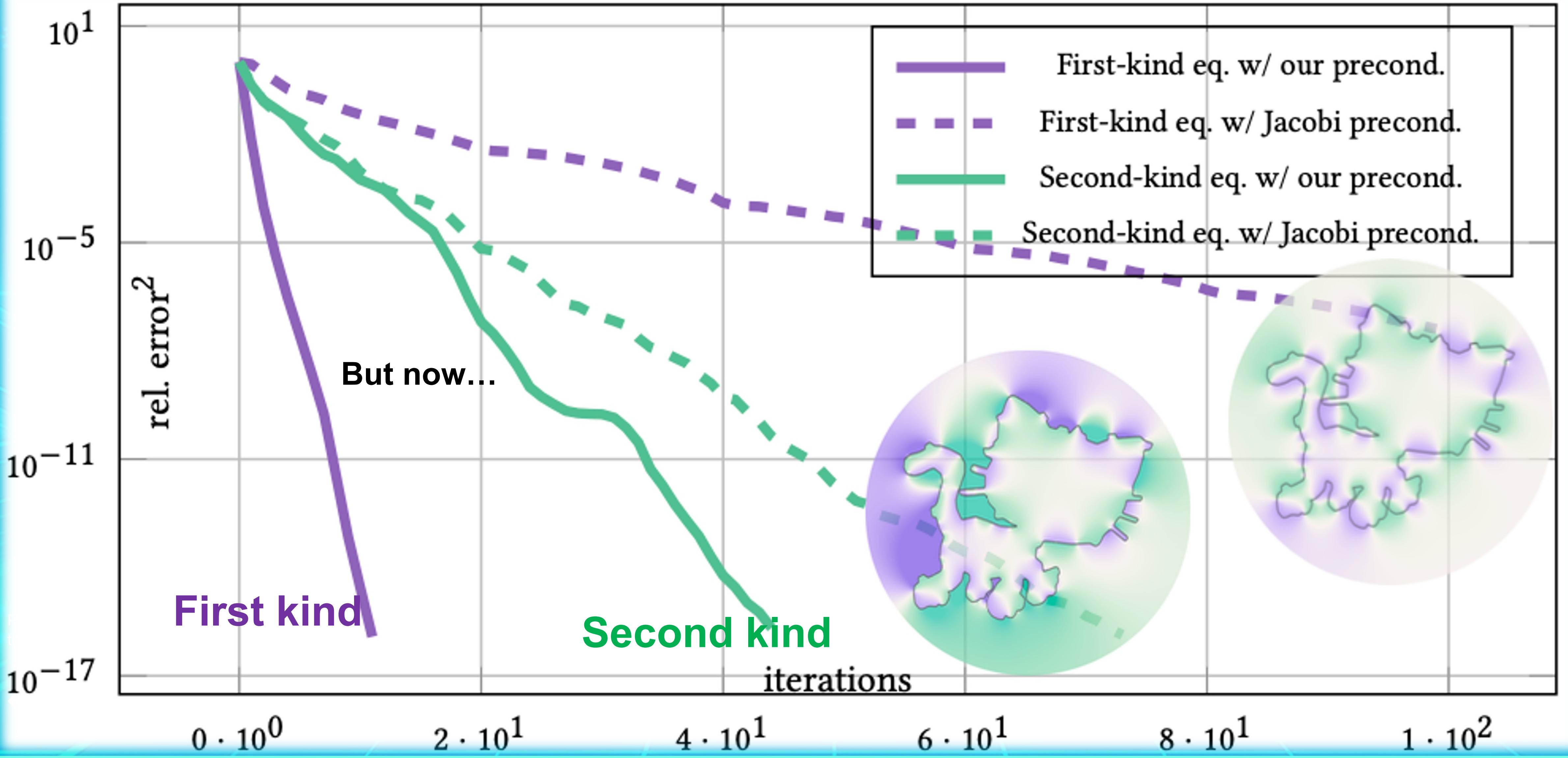
BIE

$$\frac{1-\chi_D(\mathbf{x})}{2} u(\mathbf{x}) + \int_{\Gamma_N} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}_{\mathbf{y}}} u(\mathbf{y}) \mathrm{d}A_{\mathbf{y}} - \int_{\Gamma_D} G(\mathbf{x},\mathbf{y}) \frac{\partial u(\mathbf{y})}{\partial \mathbf{n}_{\mathbf{y}}} \mathrm{d}A_{\mathbf{y}}$$
$$= -\frac{\chi_D(\mathbf{x})}{2} b(\mathbf{x}) - \int_{\Gamma_D} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}_{\mathbf{y}}} b(\mathbf{y}) \mathrm{d}A_{\mathbf{y}} + \int_{\Gamma_N} G(\mathbf{x},\mathbf{y}) g(\mathbf{y}) \mathrm{d}A_{\mathbf{y}}$$

MIXED BOUNDARY CONDITIONS

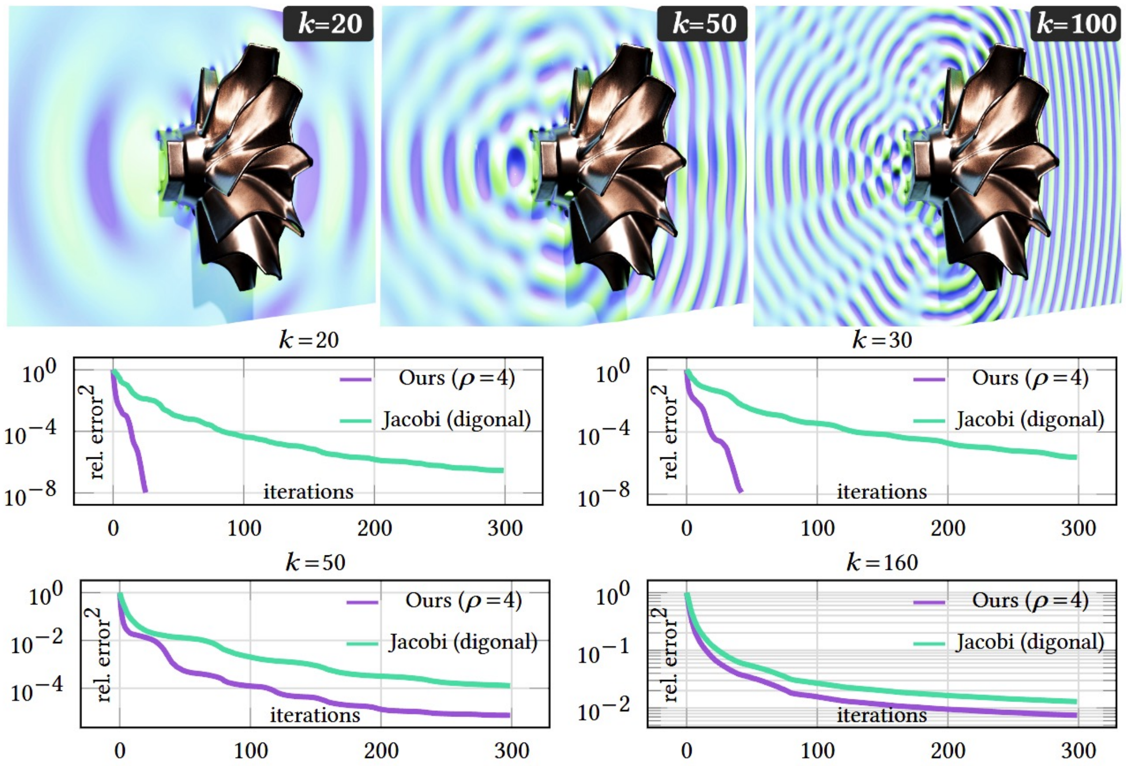



First-kind eq. w/ our precond.
First-kind eq. w/ Jacobi precond.
Second-kind eq. w/ our precond.
Second-kind eq. w/ Jacobi precond.
First kind
Second kind
When you only have a poor preconditioner…
rel. error^2
10^1
10^{-5}
10^{-11}
10^{-17}
iterations
$0 \cdot 10^0$
$2 \cdot 10^1$
$4 \cdot 10^1$
$6 \cdot 10^1$
$8 \cdot 10^1$
$1 \cdot 10^2$


First-kind eq. w/ our precond.
First-kind eq. w/ Jacobi precond.
Second-kind eq. w/ our precond.
Second-kind eq. w/ Jacobi precond.
rel. error^2
10^1
10^{-5}
10^{-11}
10^{-17}
iterations
$0 \cdot 10^0$
$2 \cdot 10^1$
$4 \cdot 10^1$
$6 \cdot 10^1$
$8 \cdot 10^1$
$1 \cdot 10^2$
But now…
First kind
Second kind

LIMITATION AND FUTURE WORK

- Debased efficiency due to weakened screening effects
  - Screening effect hinges on the smoothness of the kernel functions
  - Certain cases that reduce the smoothness of the kernel
    - High-frequency Helmholtz equation
    - Addition of a positive diagonal matrix, i.e., $\int_\Gamma \partial G + \alpha Id$
    - Mix of different kernels, e.g., BIE for mixed boundary conditions
- Future work
  - Explore more effective strategies for above issues
  - Extension to least-squares problems for rectangular systems
  - Boundary-only or meshless methods for nonlinear PDEs



Thanks !