A fully adaptive, high-order, fast Poisson solver for complex two-dimensional geometries

Dan Fortunato

Flatiron Institute

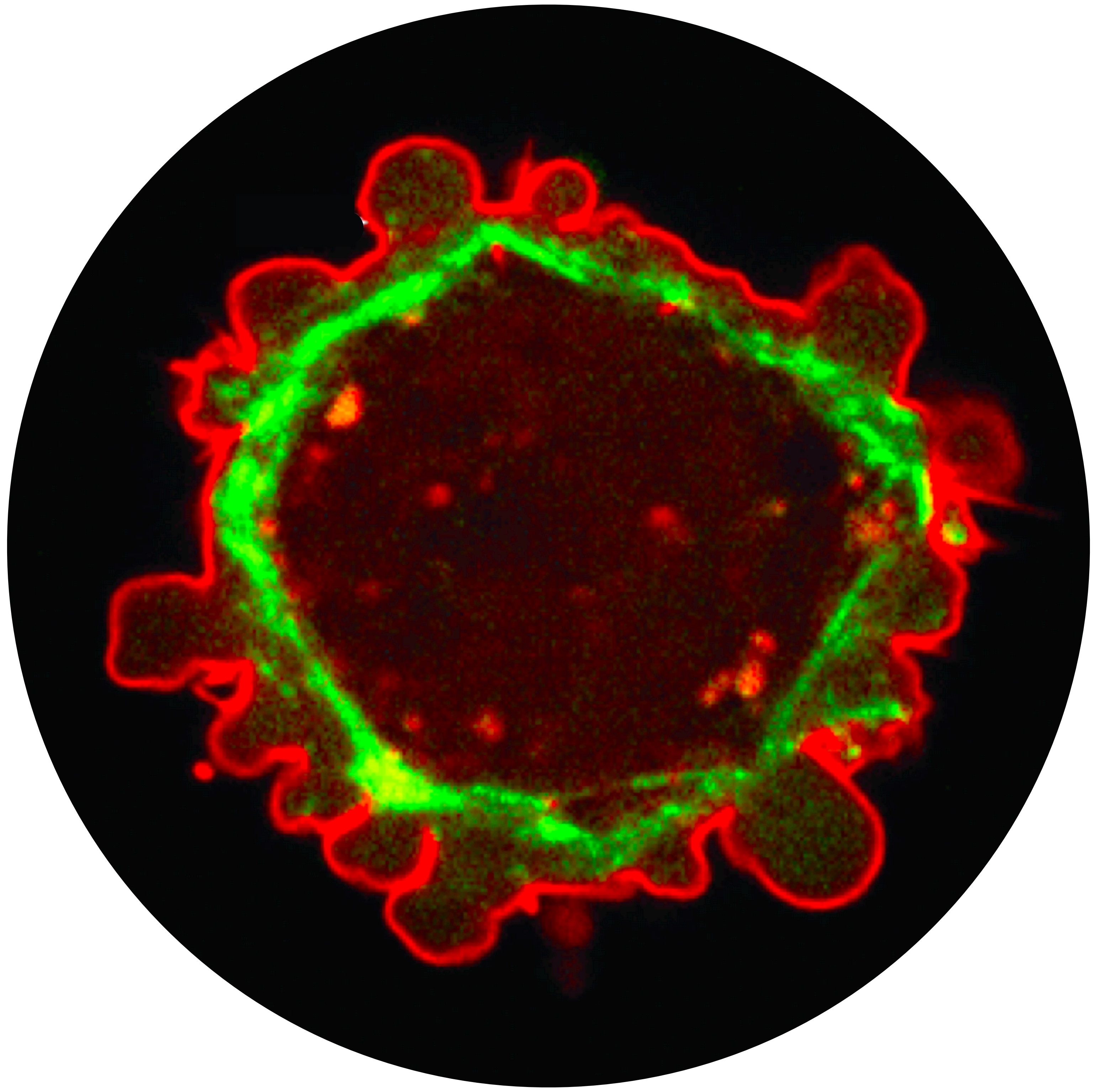
David Stein

FLATIRON INSTITUTE

Center for Computational Mathematics

Alex Barnett

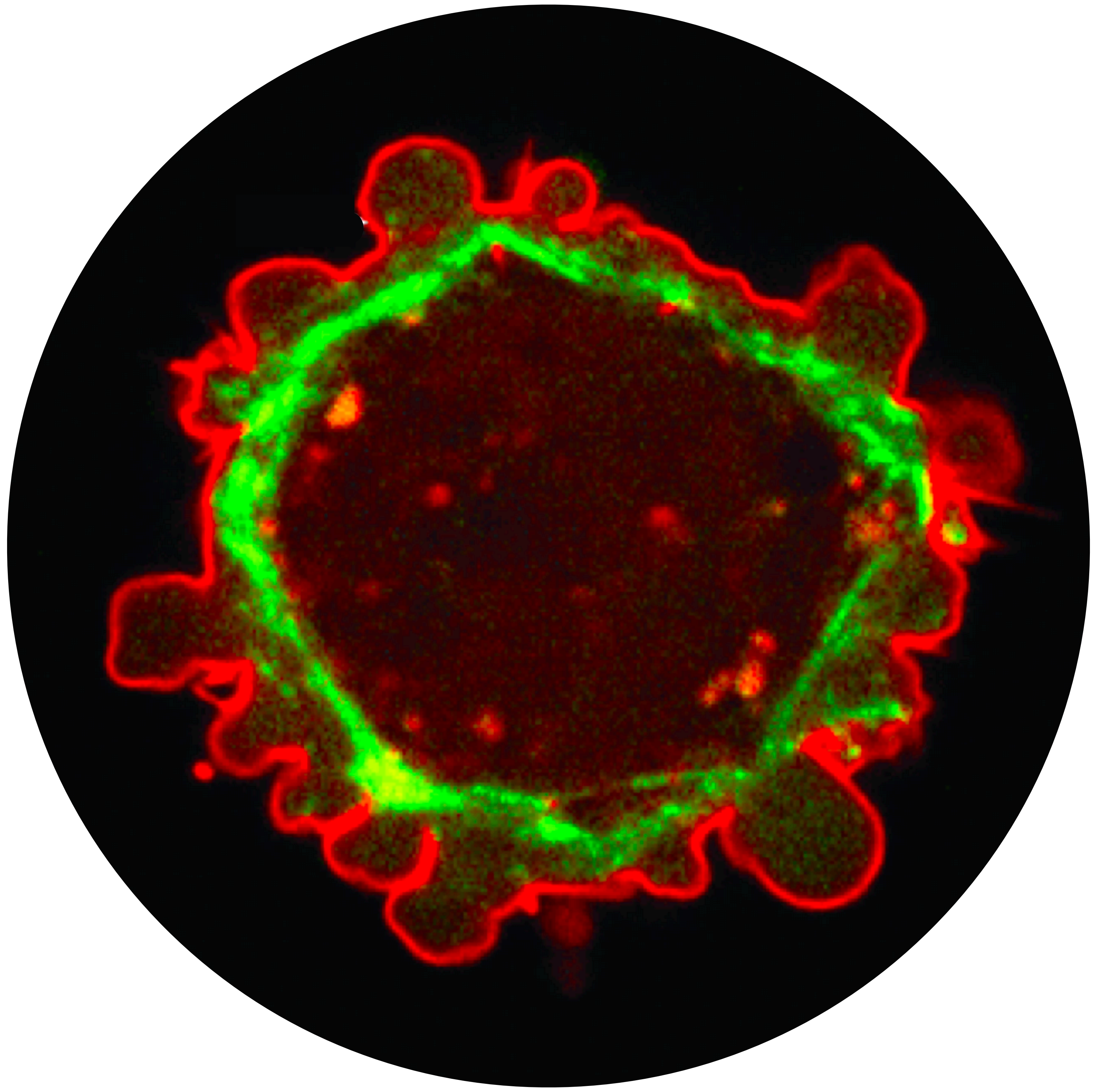
Geometry can drive the dynamics of physical phenomena, but can pose a numerical challenge.



[Charras, 2008]

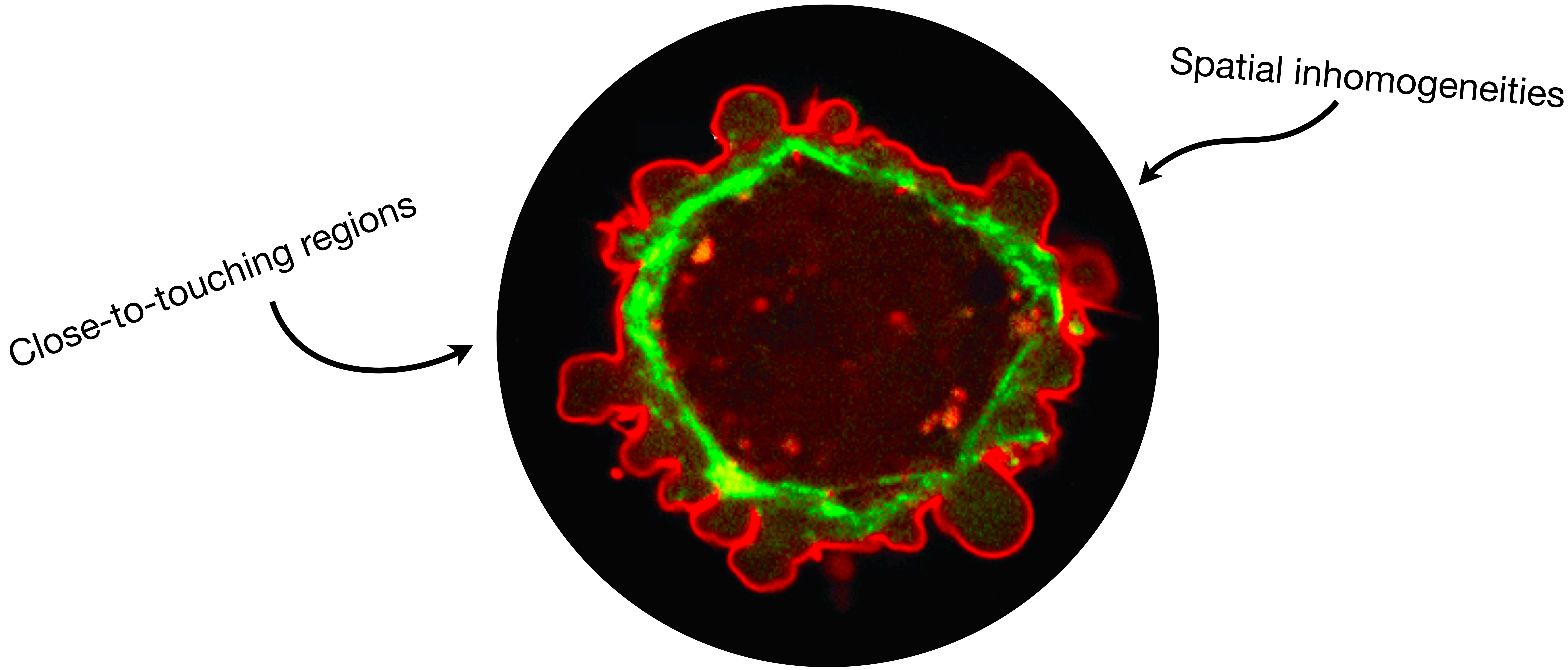
Geometry can drive the dynamics of physical phenomena, but can pose a numerical challenge.

Close-to-touching regions



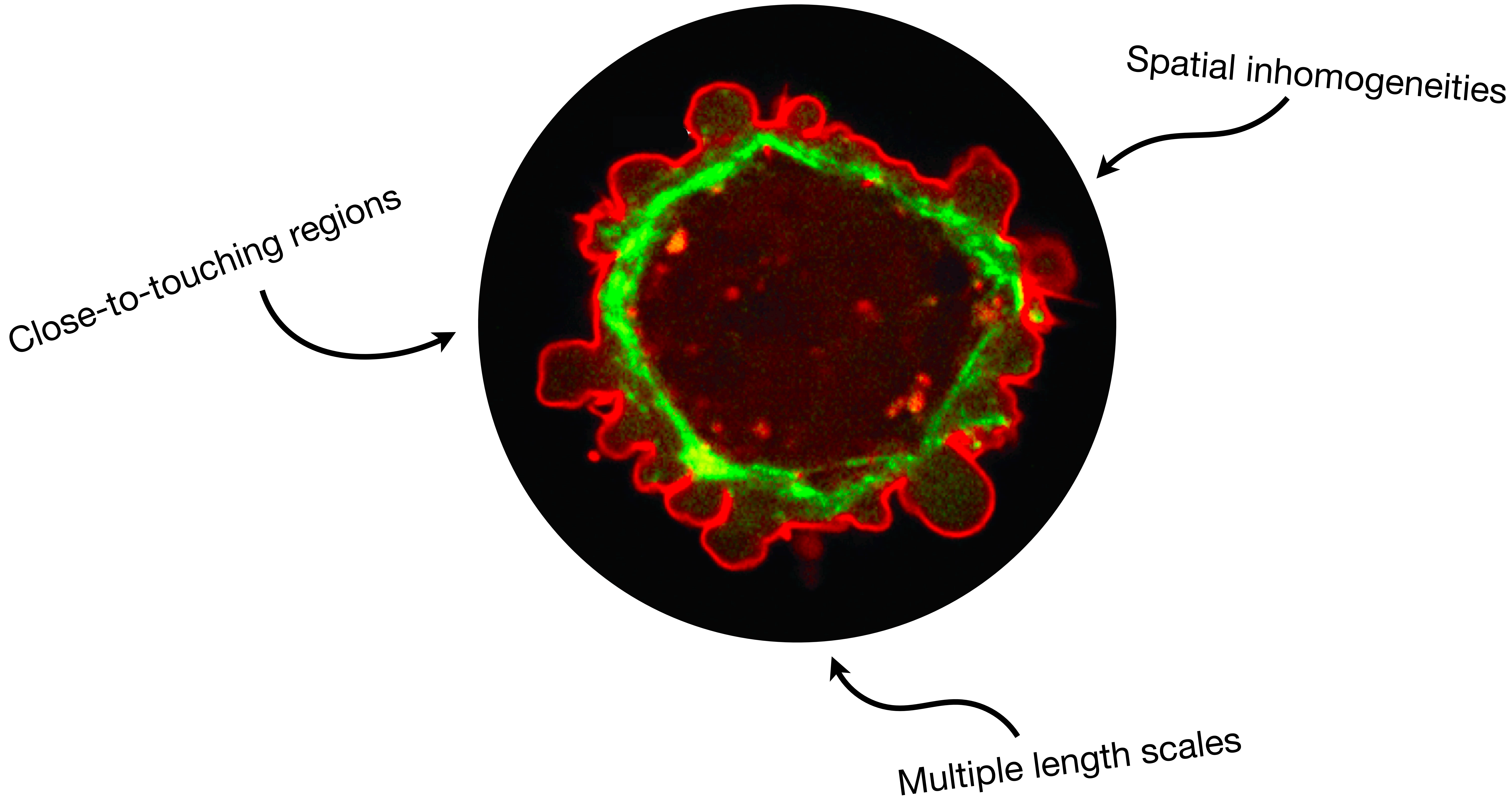
[Charras, 2008]

Geometry can drive the dynamics of physical phenomena, but can pose a numerical challenge.



[Charras, 2008]

Geometry can drive the dynamics of physical phenomena, but can pose a numerical challenge.



[Charras, 2008]

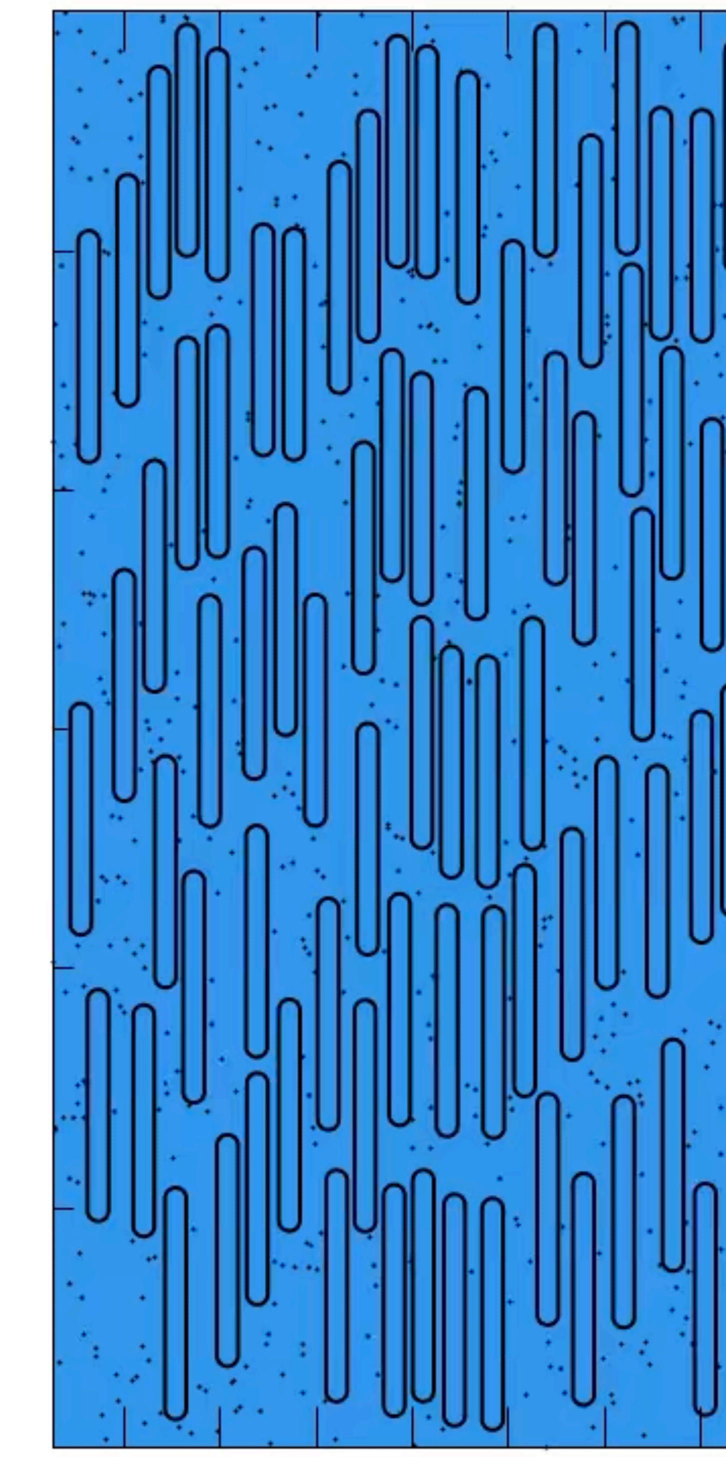
Inhomogeneous PDEs
Introduction

Many applications involve solving an inhomogeneous elliptic BVP.

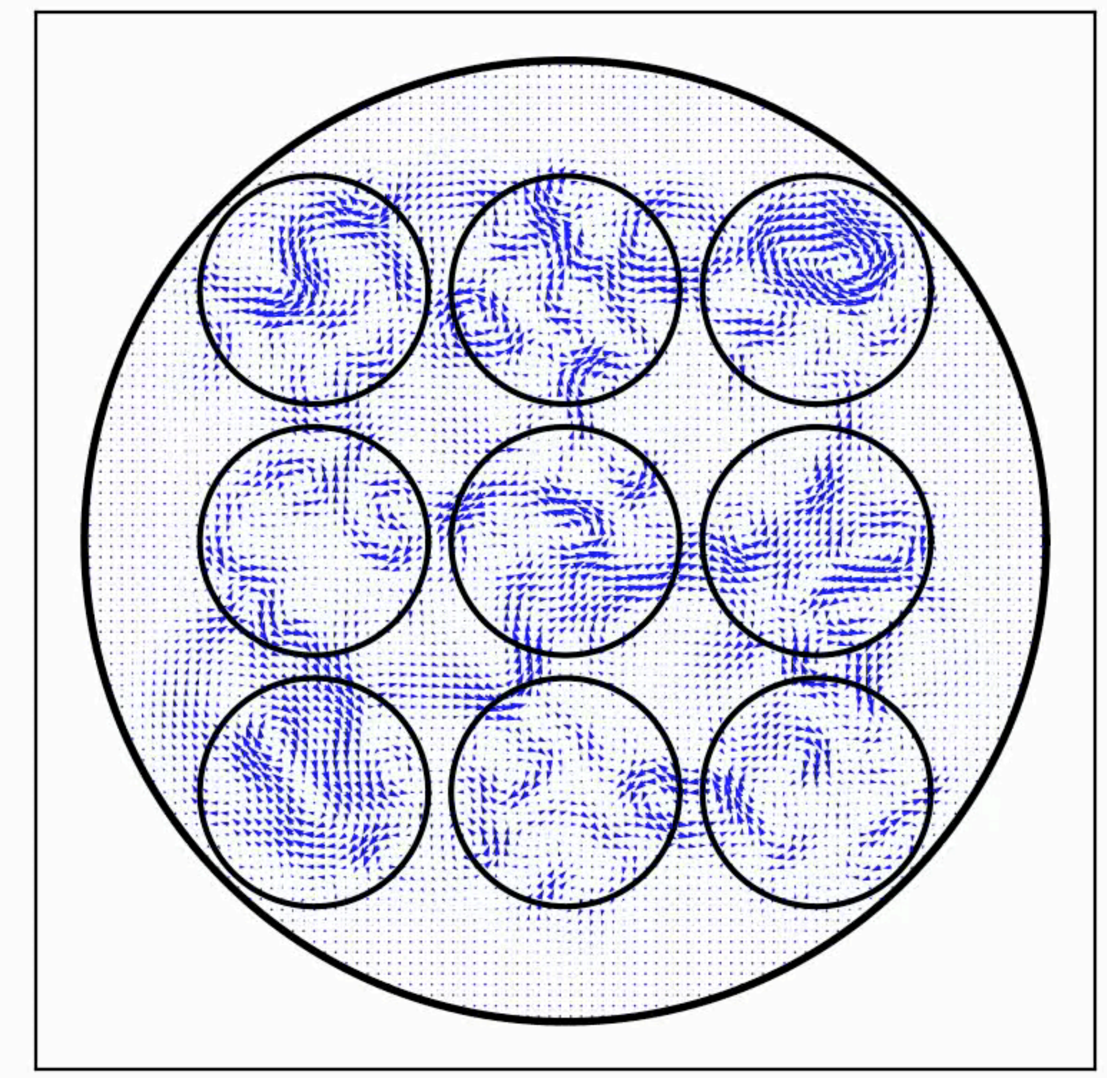
$$Lu = f \quad \text{in } \Omega \qquad \text{(e.g. Poisson, Helmholtz, Stokes, ...)}$$

$$Bu = g \quad \text{on } \Gamma \qquad \text{(e.g. Dirichlet, Neumann, Robin, ...)}$$

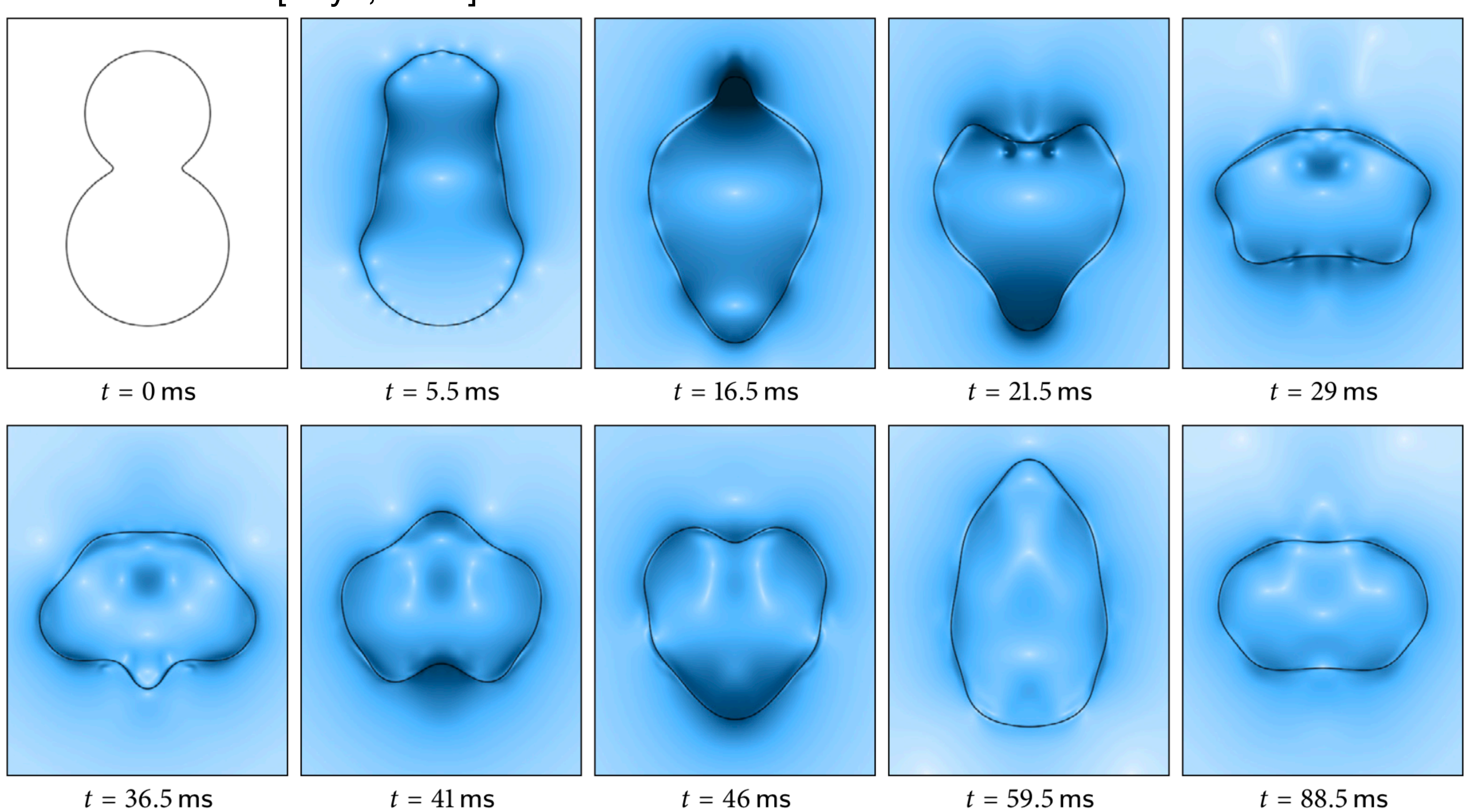
Fluid-structure interaction
[Rycroft et al., 2020]



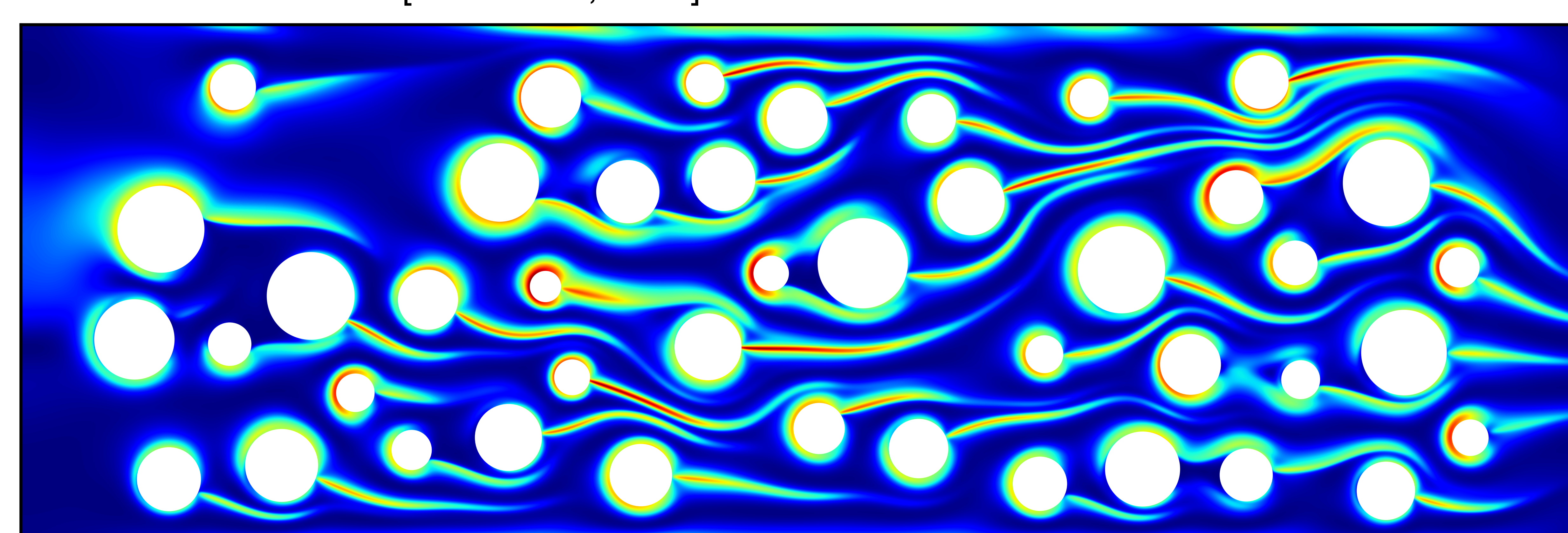
Active droplets [Stein, 2021]



Bubble collision [Saye, 2017]



Non-Newtonian fluids [Stein et al., 2019]



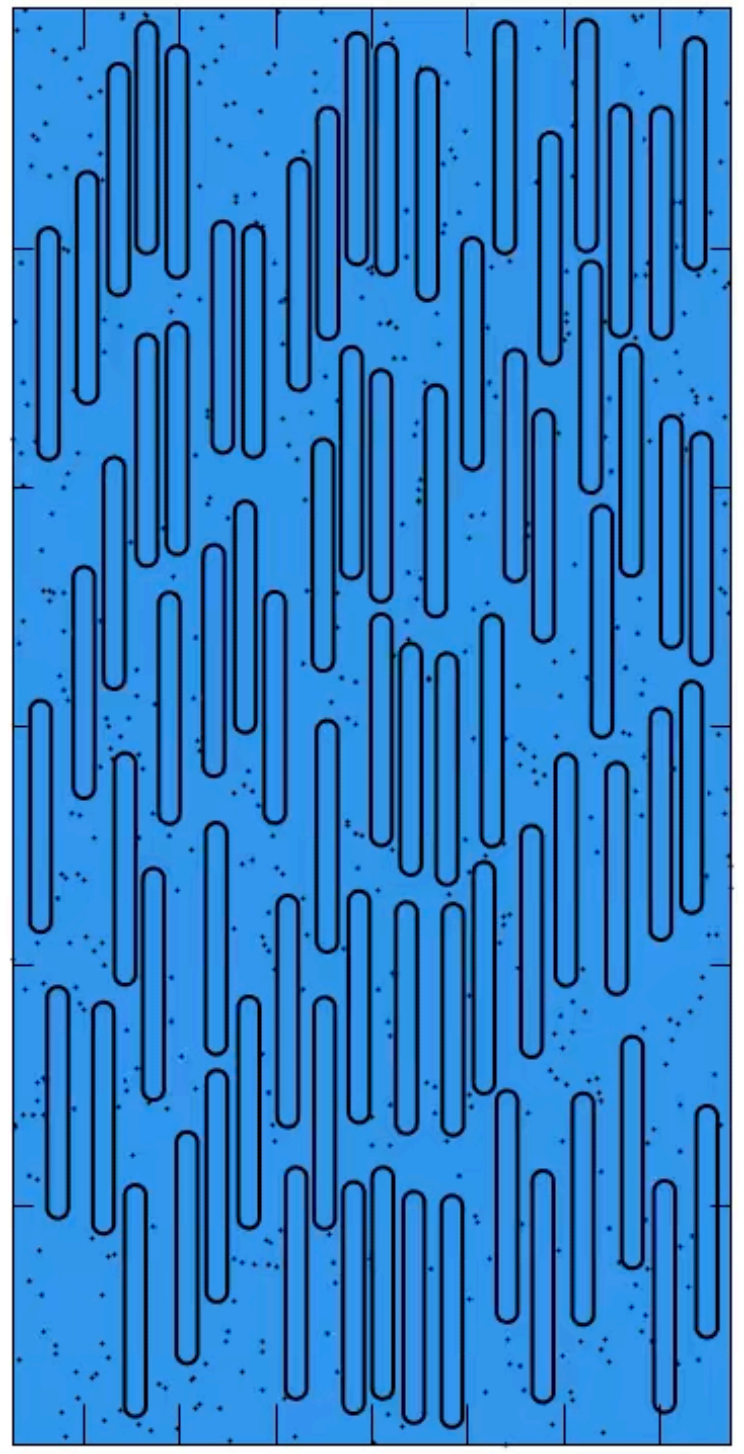
Inhomogeneous PDEs
Introduction

Many applications involve solving an inhomogeneous elliptic BVP.

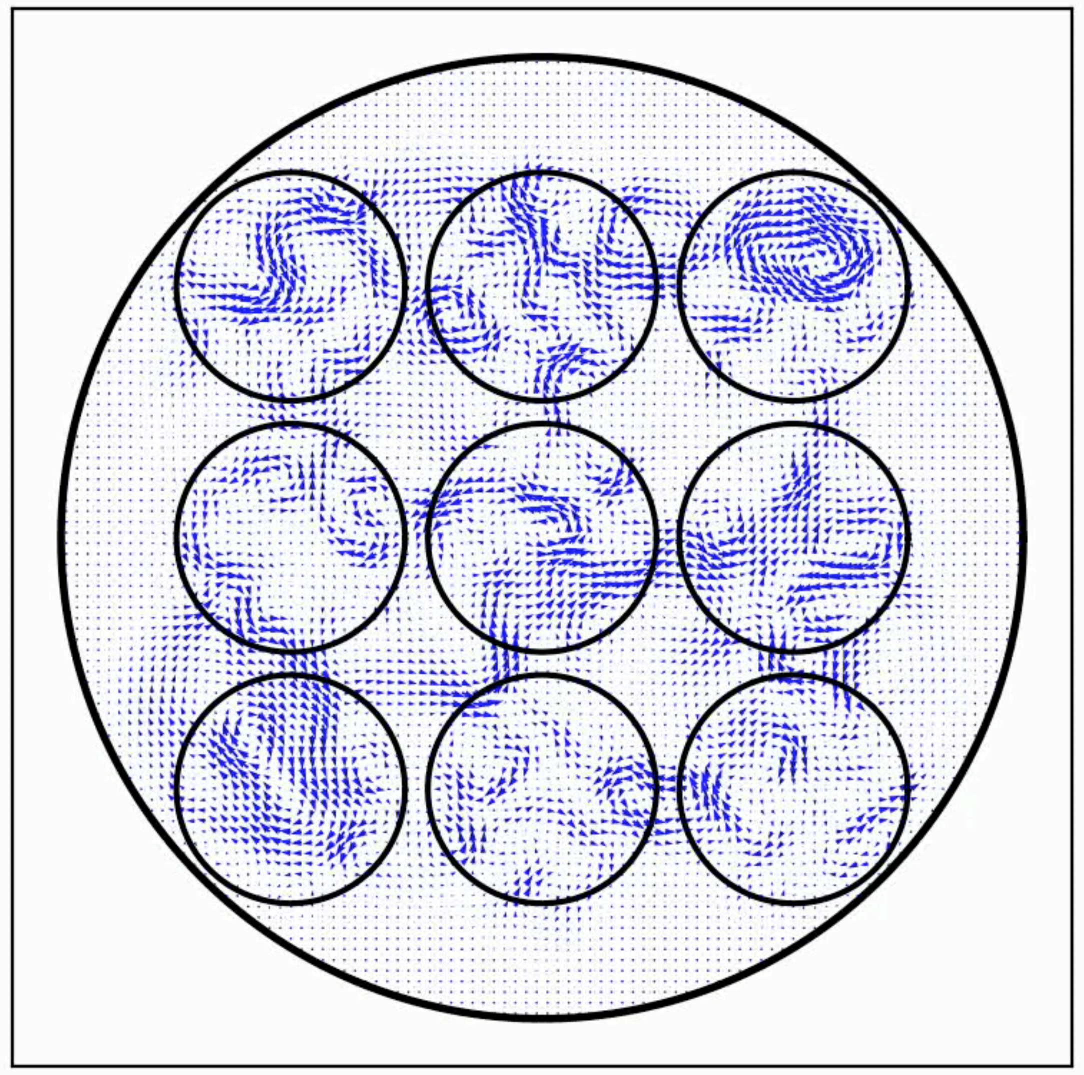
$$Lu = f \quad \text{in } \Omega \qquad \text{(e.g. Poisson, Helmholtz, Stokes, ...)}$$

$$Bu = g \quad \text{on } \Gamma \qquad \text{(e.g. Dirichlet, Neumann, Robin, ...)}$$

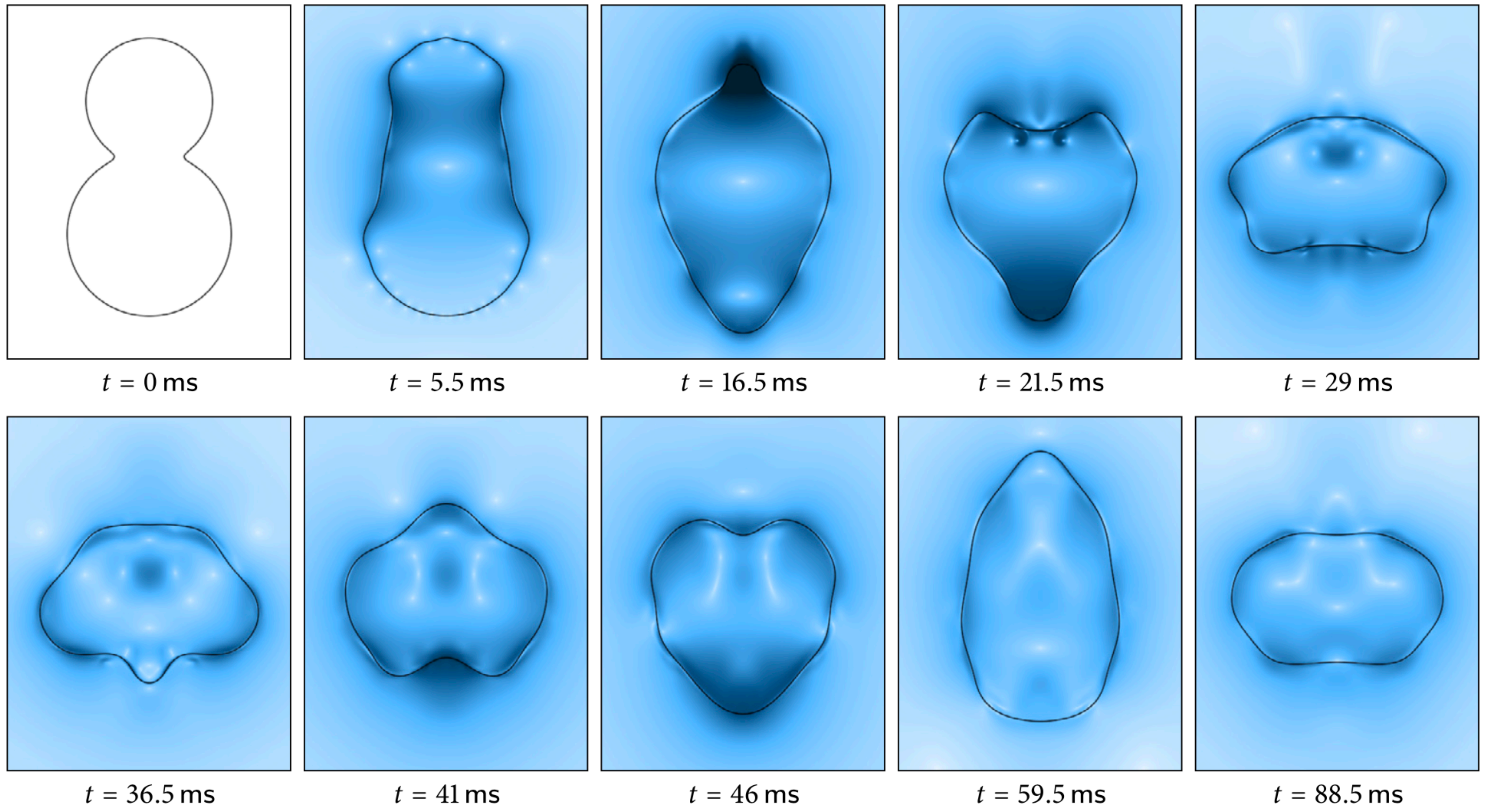
Fluid-structure interaction [Rycroft et al., 2020]



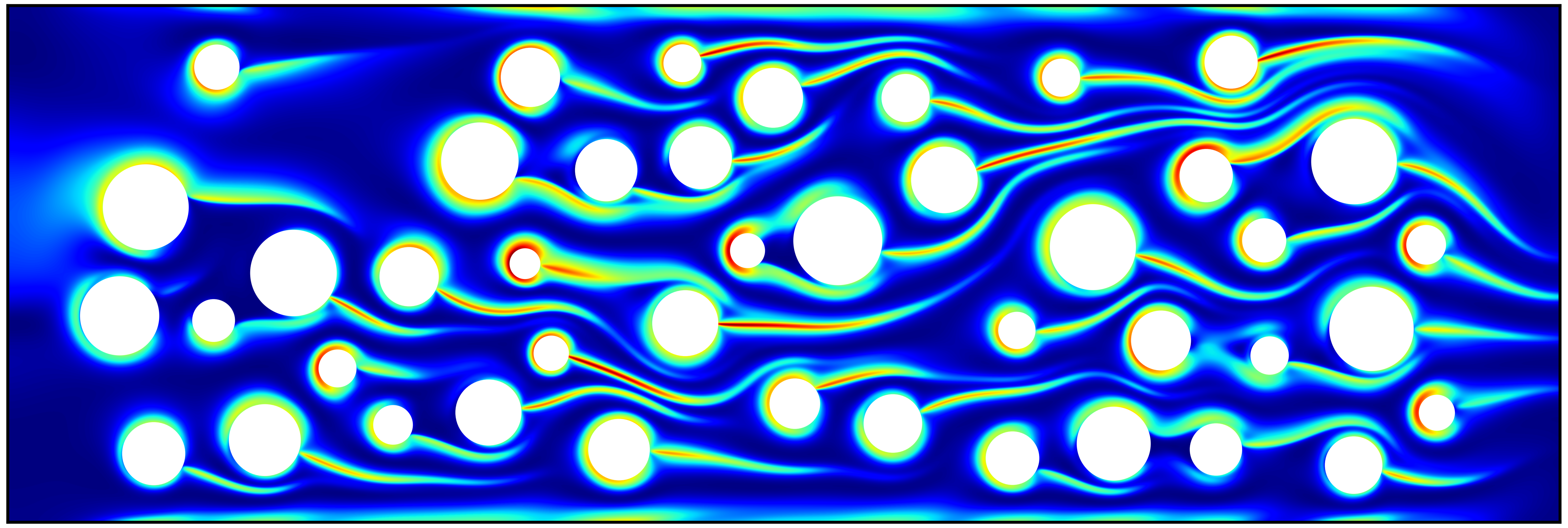
Active droplets [Stein, 2021]



Bubble collision [Saye, 2017]



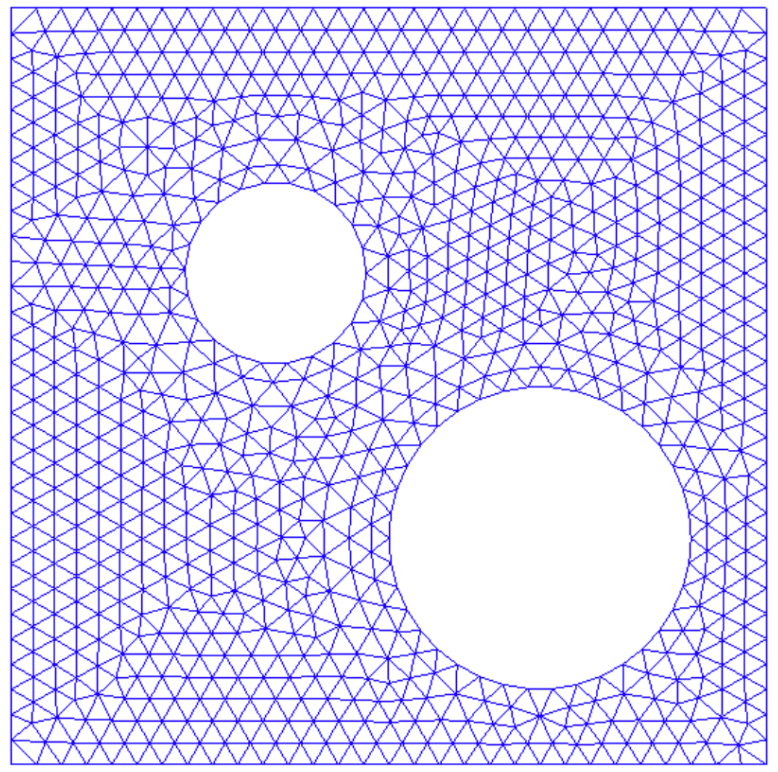
Non-Newtonian fluids [Stein et al., 2019]

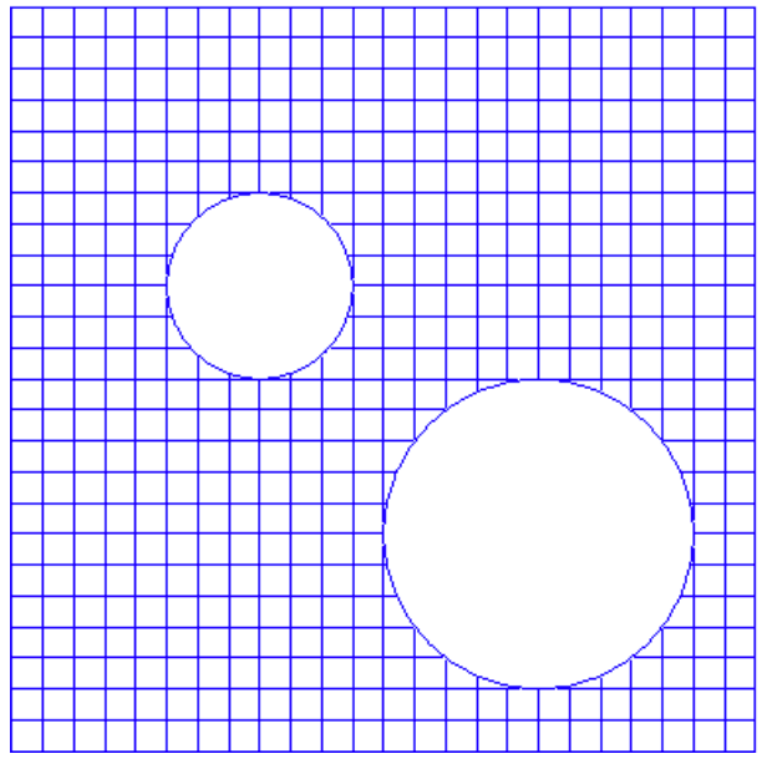


Inhomogeneous PDEs

Many approaches for inhomogeneous BVP

Traditional


Conforming


Cut-cell

- Mesh generation (or cut-cell generation)
- Directly discretize with FEM, FDM, SEM, ...

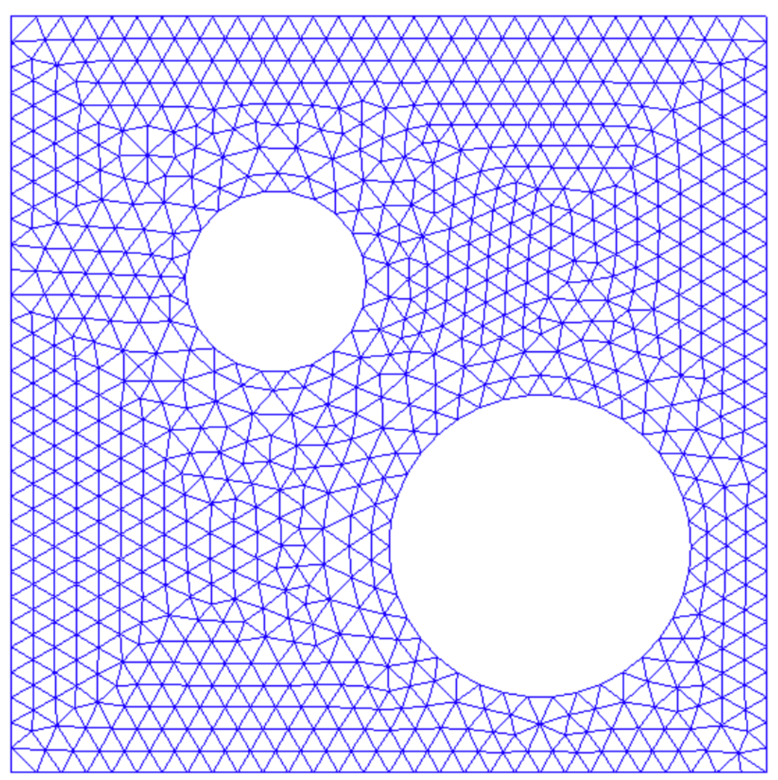
$$Ax = b$$

Discretize on a mesh.
Solve linear system for volume DoFs.

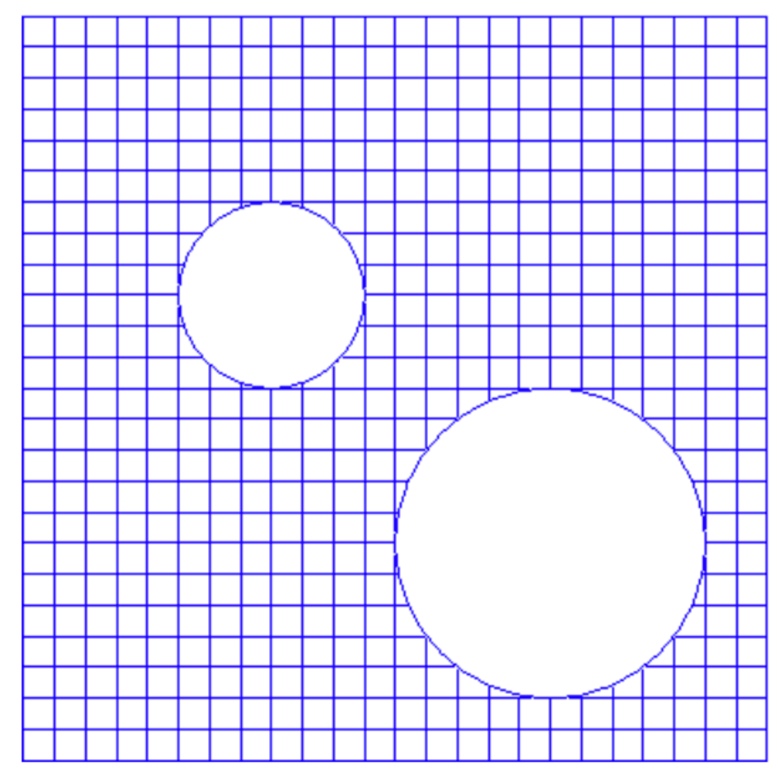
Inhomogeneous PDEs

Many approaches for inhomogeneous BVP

Traditional



Conforming



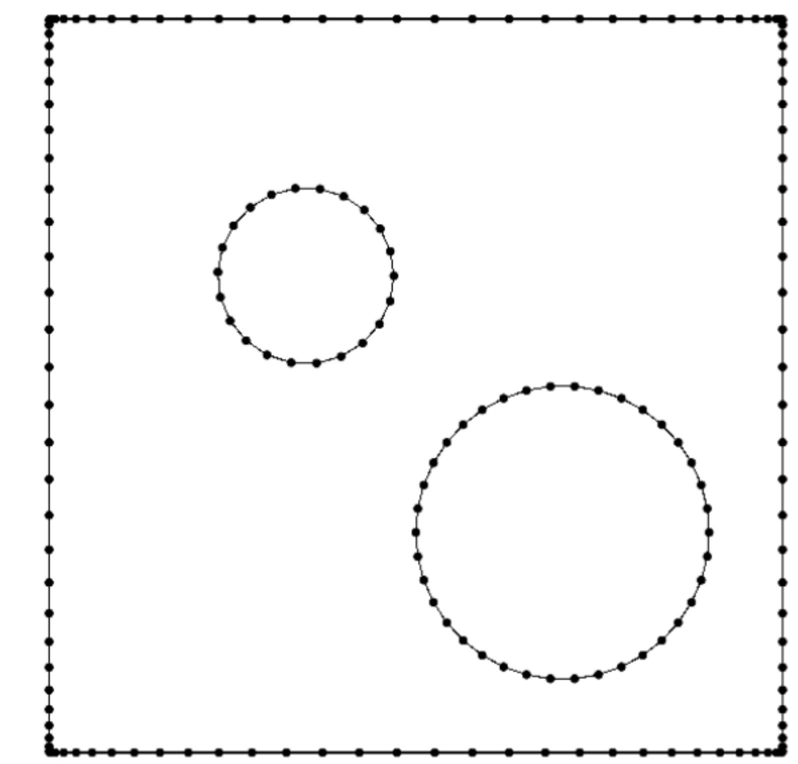
Cut-cell

- Mesh generation (or cut-cell generation)
- Directly discretize with FEM, FDM, SEM, …

$$Ax = b$$

Discretize on a mesh.
Solve linear system for volume DoFs.

Analysis-based



- Write solution as $u = v + w$

particular solution *homogeneous solution*

- Find **some** (any!) function v such that

$$Lv = f \quad \text{in } \Omega$$

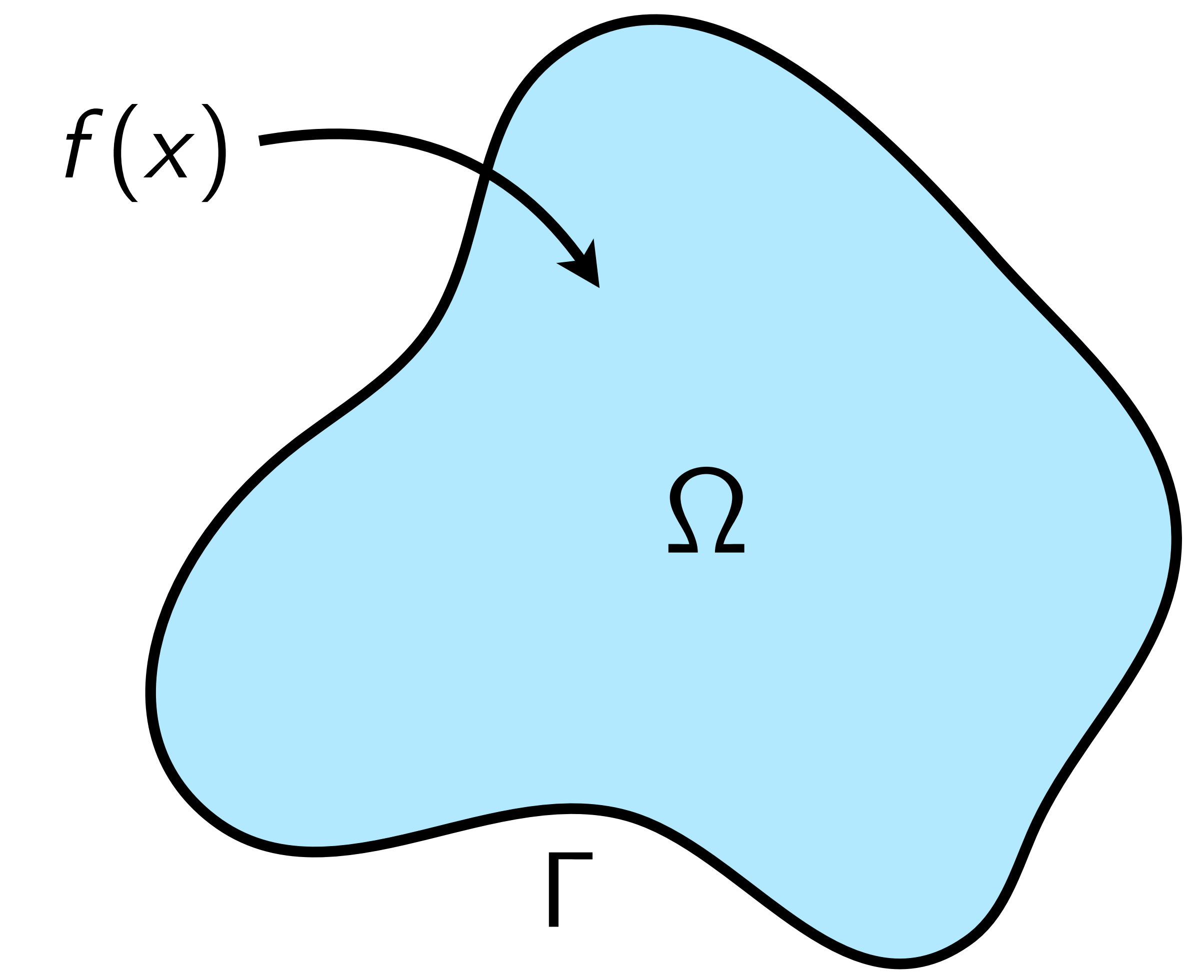
No solve required. Convolve with Green's function.

- Compute u_H to satisfy boundary conditions:

$$\begin{aligned} Lw &= 0 && \text{in } \Omega \\ Bw &= g - Bv && \text{on } \Gamma \end{aligned}$$

Solve linear system for boundary DoFs using BIE.

In many applications $f(x)$ is only given inside Ω. So we have a few options...

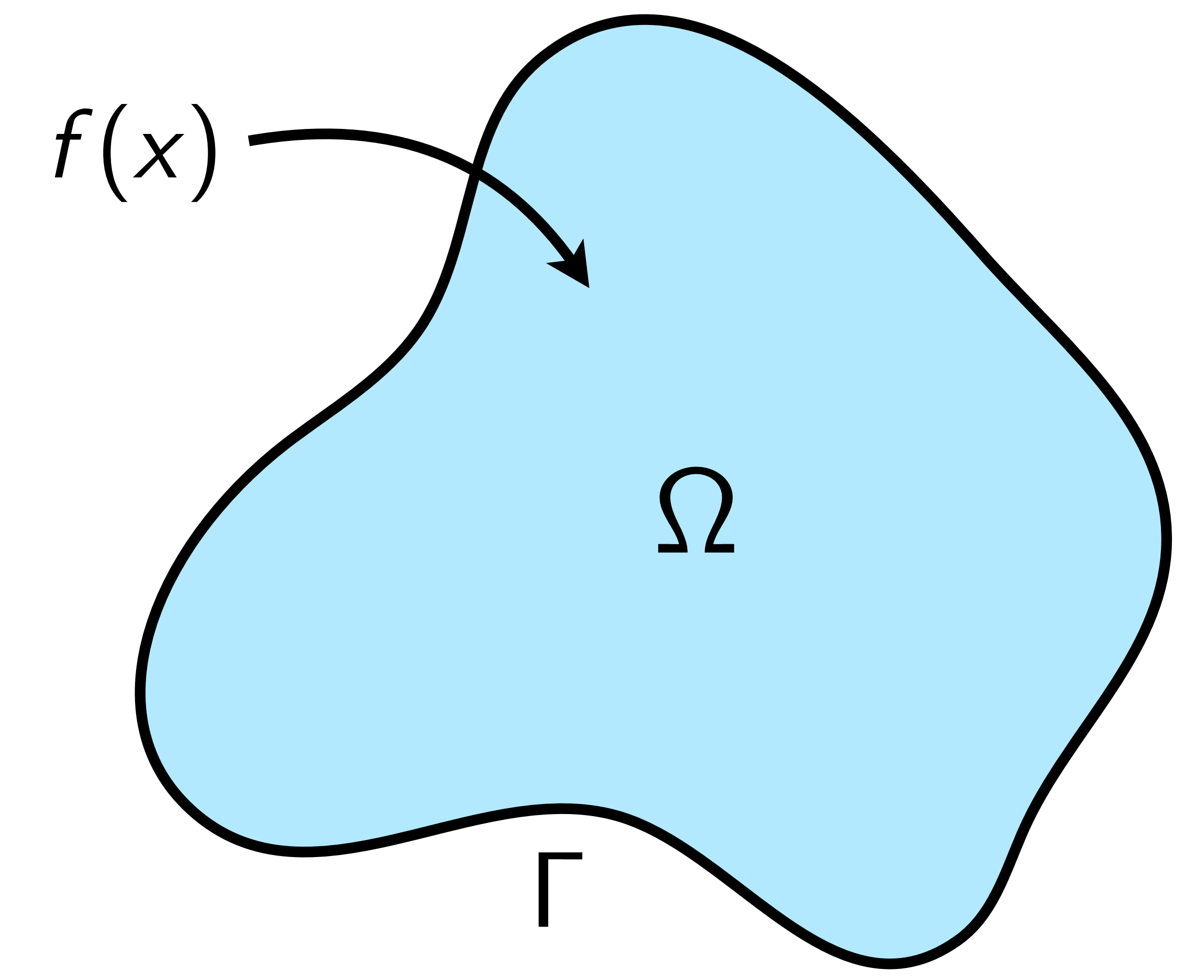


In many applications $f(x)$ is only given inside Ω. So we have a few options...

- Build accurate quadrature scheme over Ω (e.g. adaptive boxes with cut cells near boundary) and compute

$$v(x) = \int_\Omega \Phi(x, y)\, f(y)\, dy \qquad \left(\text{Poisson: } \Phi(x, y) = \frac{1}{2\pi} \log |x - y| \right)$$

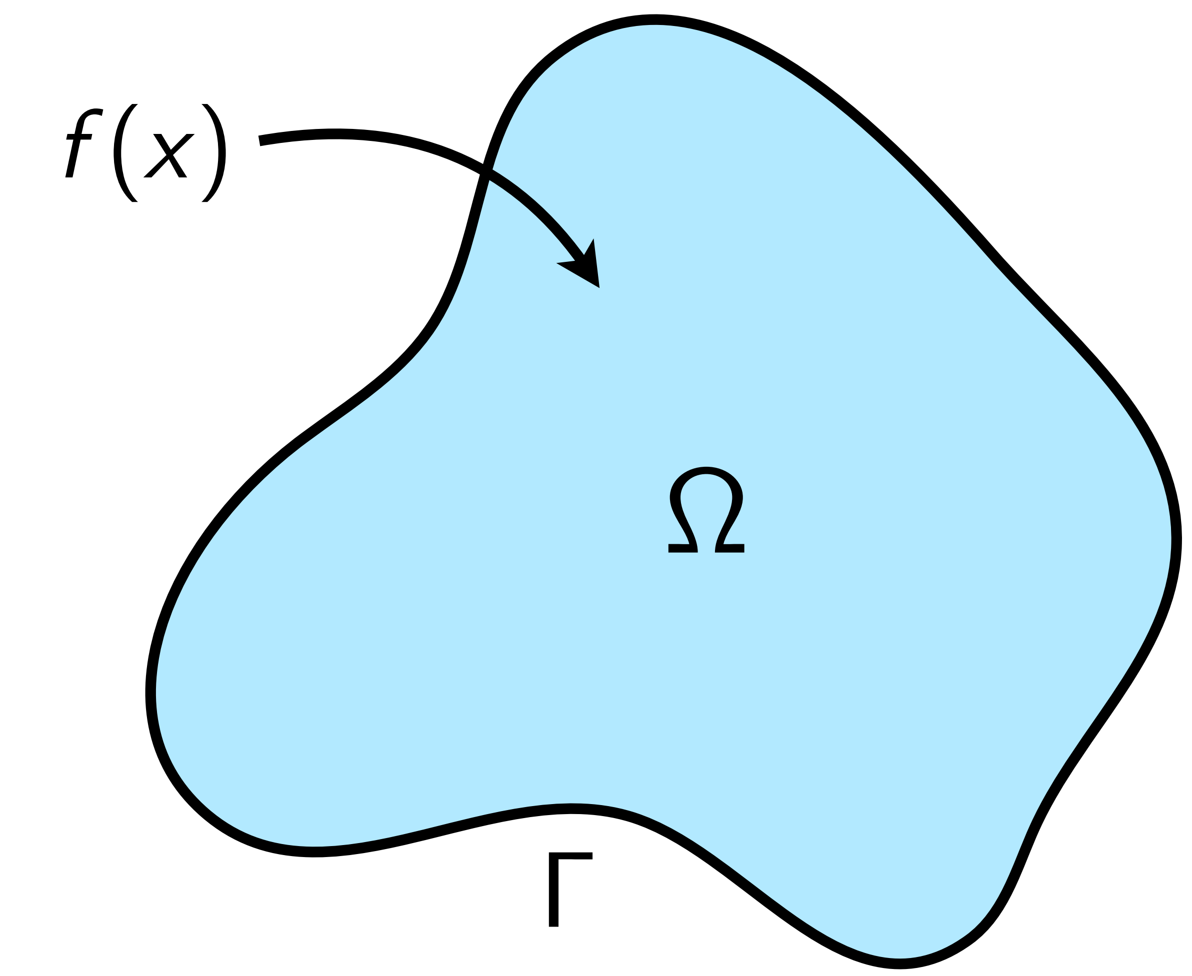
free space fundamental solution



In many applications $f(x)$ is only given inside Ω. So we have a few options...

- Build accurate quadrature scheme over Ω (e.g. adaptive boxes with cut cells near boundary) and compute

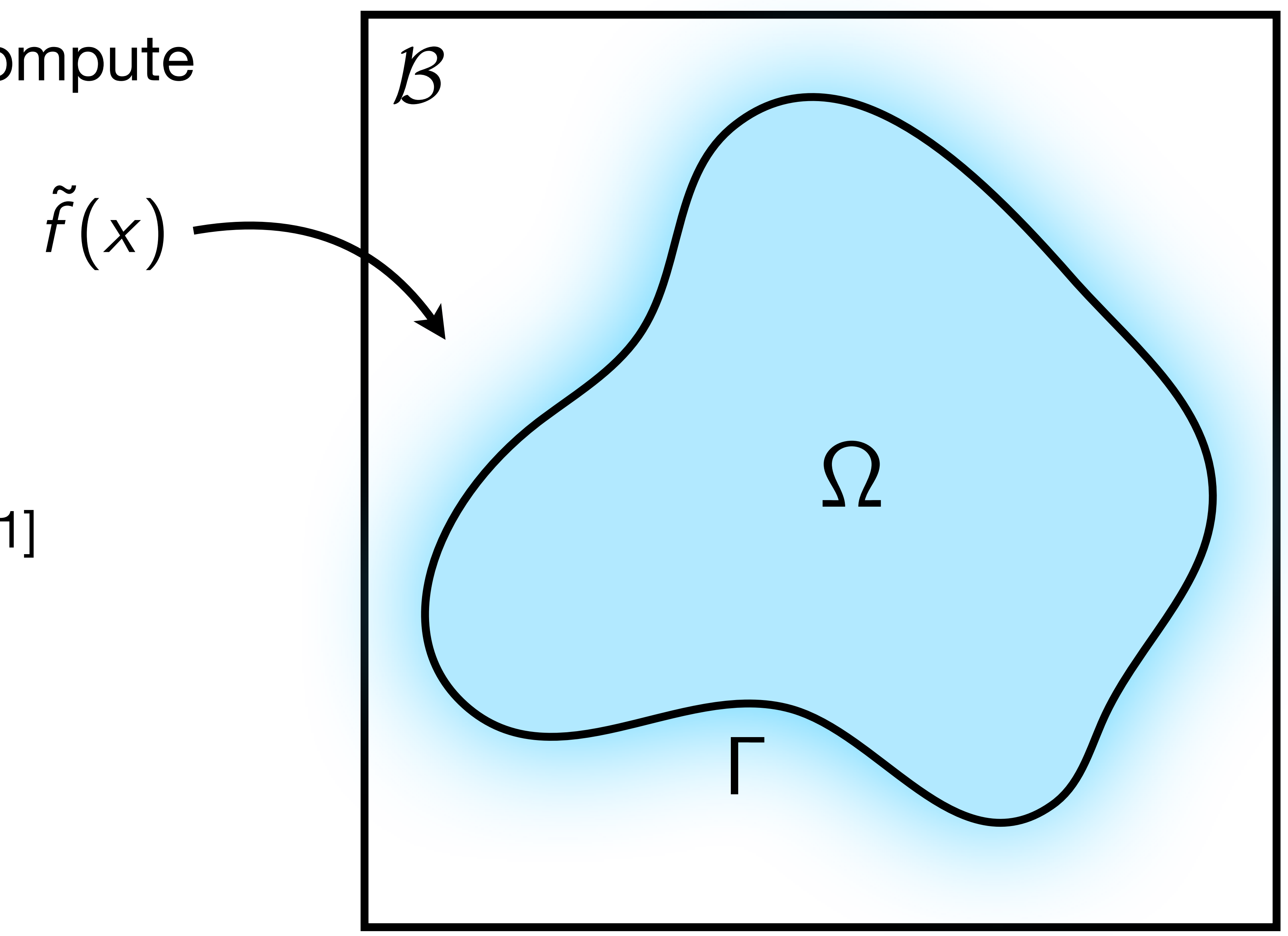
$$v(x) = \int_{\Omega} \underbrace{\Phi(x,y)}_{\text{free space fundamental solution}} f(y)\, dy \qquad \left(\text{Poisson: } \Phi(x,y) = \frac{1}{2\pi} \log|x-y| \right)$$



- Extend f to $\tilde{f}$ outside Ω ("function extension"). Adaptively resolve $\tilde{f}$ and compute

$$v(x) = \int_{\mathcal{B}} \Phi(x,y)\, \tilde{f}(y)\, dy$$

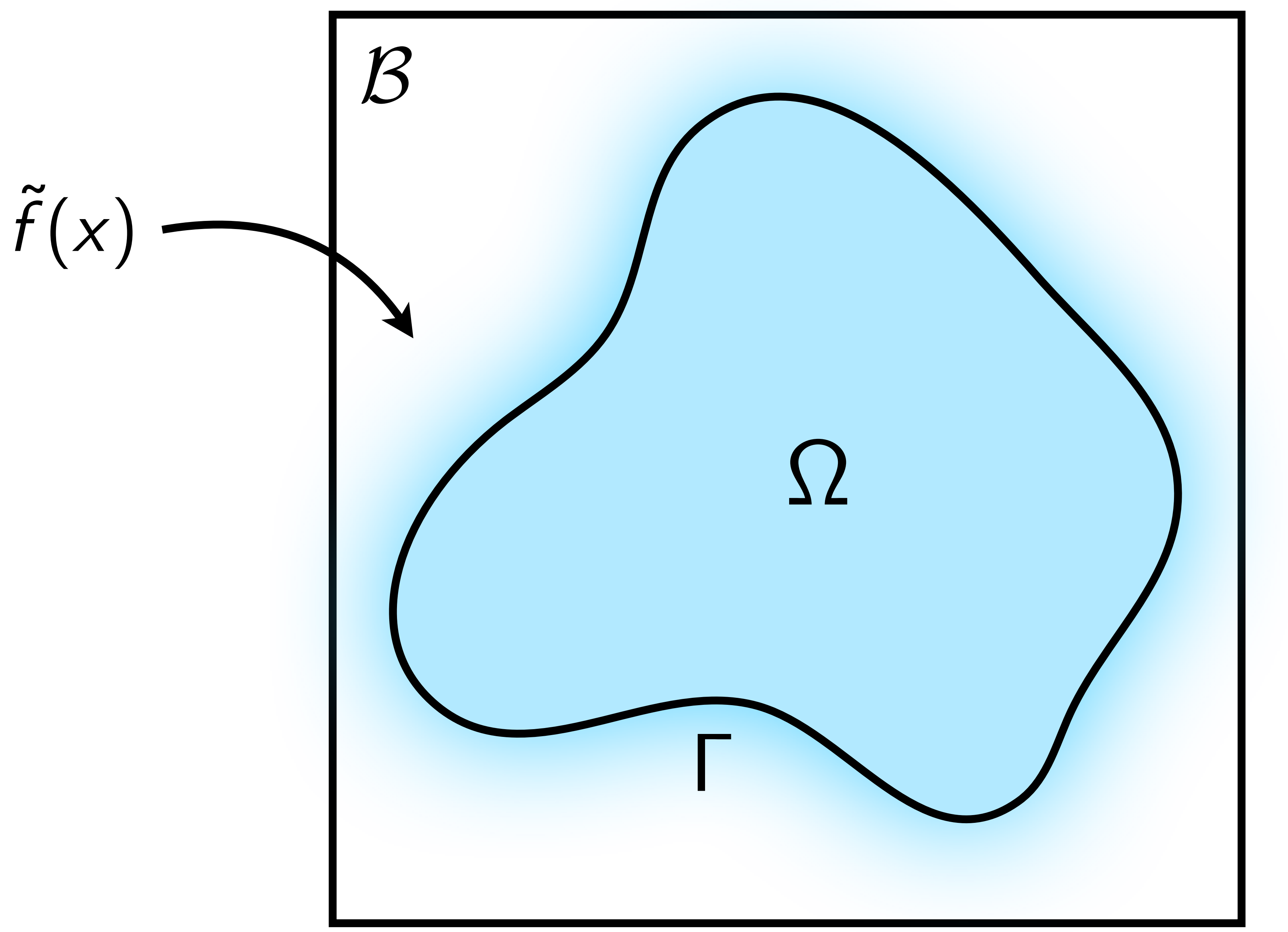
- Quadrature tables for boxes can be **precomputed**.
- Apply FMM to adaptive quadtree of $\tilde{f}$ ("box code"). [Ethridge & Greengard, 2001]
- Want v as smooth as u so no effort is wasted.



Function extension

Prior work

- Finite difference extension, second-order accurate [Mayo, 1984]
- Fourier continuation [Bruno & Lyon, 2010], [Bruno & Paul, 2020]
- Immersed boundary smooth extension [Stein, Guy, & Thomases, 2015]
- C^k polyharmonic extension + box code, fourth-order accurate [Askham & Cerfon, 2017]
- Partition of unity extension [Fryklund, Lehto, & Tornberg, 2018], [Fryklund & Greengard, 2022]

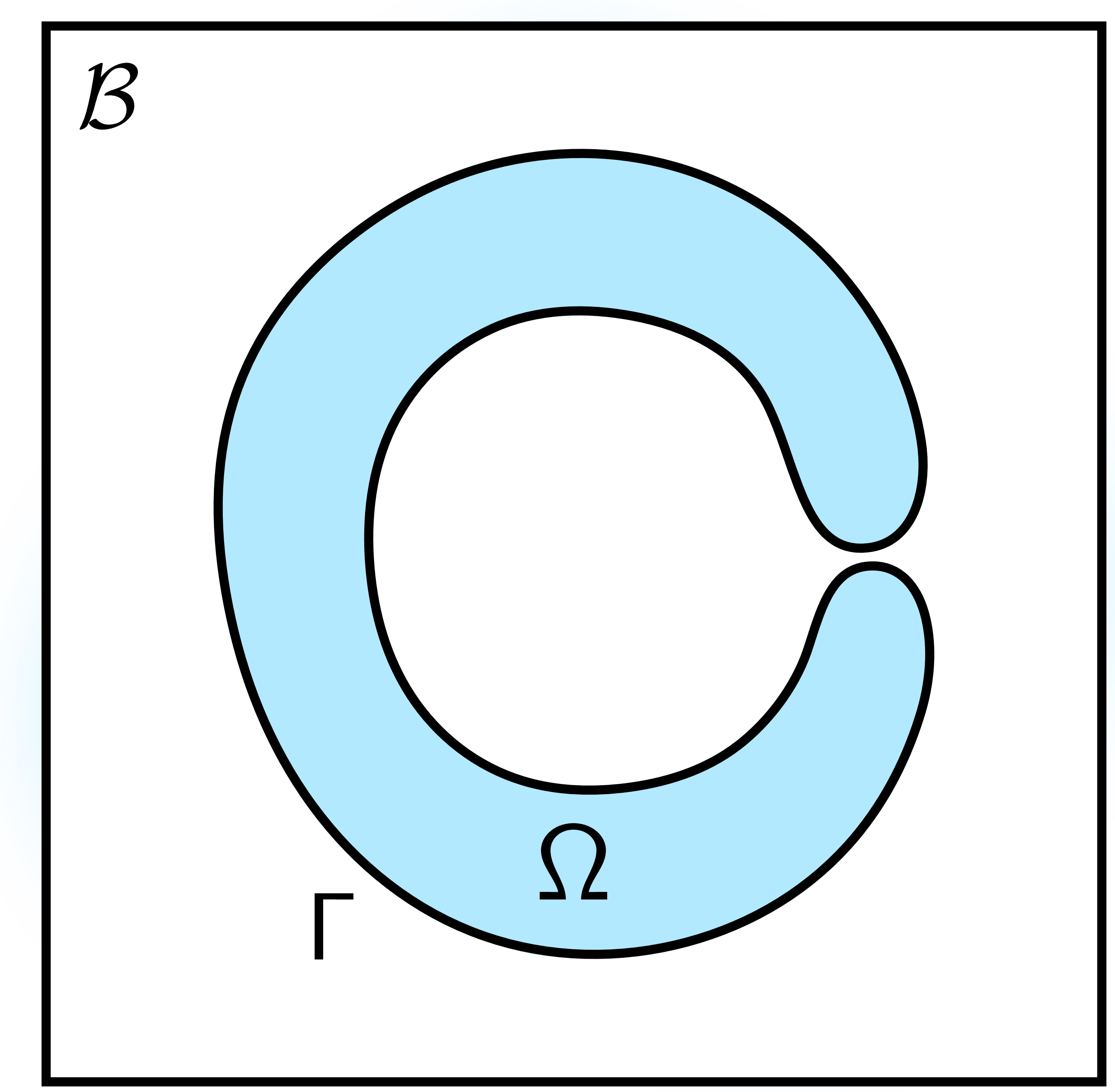


Function extension

Prior work

- Finite difference extension, second-order accurate [Mayo, 1984]
- Fourier continuation [Bruno & Lyon, 2010], [Bruno & Paul, 2020]
- Immersed boundary smooth extension [Stein, Guy, & Thomases, 2015]
- C^k polyharmonic extension + box code, fourth-order accurate [Askham & Cerfon, 2017]
- Partition of unity extension [Fryklund, Lehto, & Tornberg, 2018], [Fryklund & Greengard, 2022]

$\tilde{f}(x)$...?



Function extension

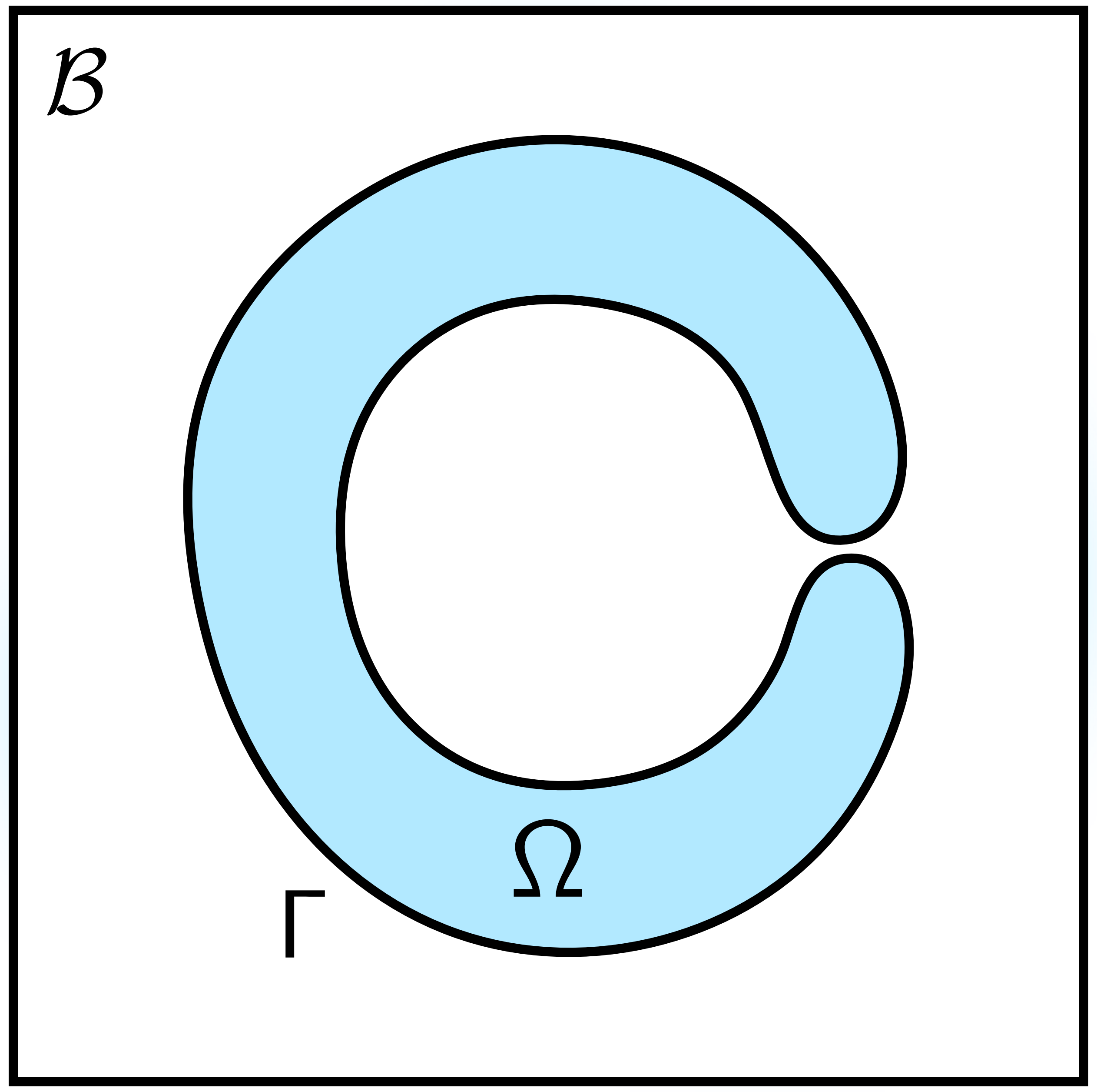
Prior work

- Finite difference extension, second-order accurate [Mayo, 1984]
- Fourier continuation [Bruno & Lyon, 2010], [Bruno & Paul, 2020]
- Immersed boundary smooth extension [Stein, Guy, & Thomases, 2015]
- C^k polyharmonic extension + box code, fourth-order accurate [Askham & Cerfon, 2017]
- Partition of unity extension [Fryklund, Lehto, & Tornberg, 2018], [Fryklund & Greengard, 2022]

*Smoothly extending functions in a **robust** way is **challenging**!*

- multiscale geometry
- multiscale data
- close-to-touching regions

$\tilde{f}(x)$...?



Function extension

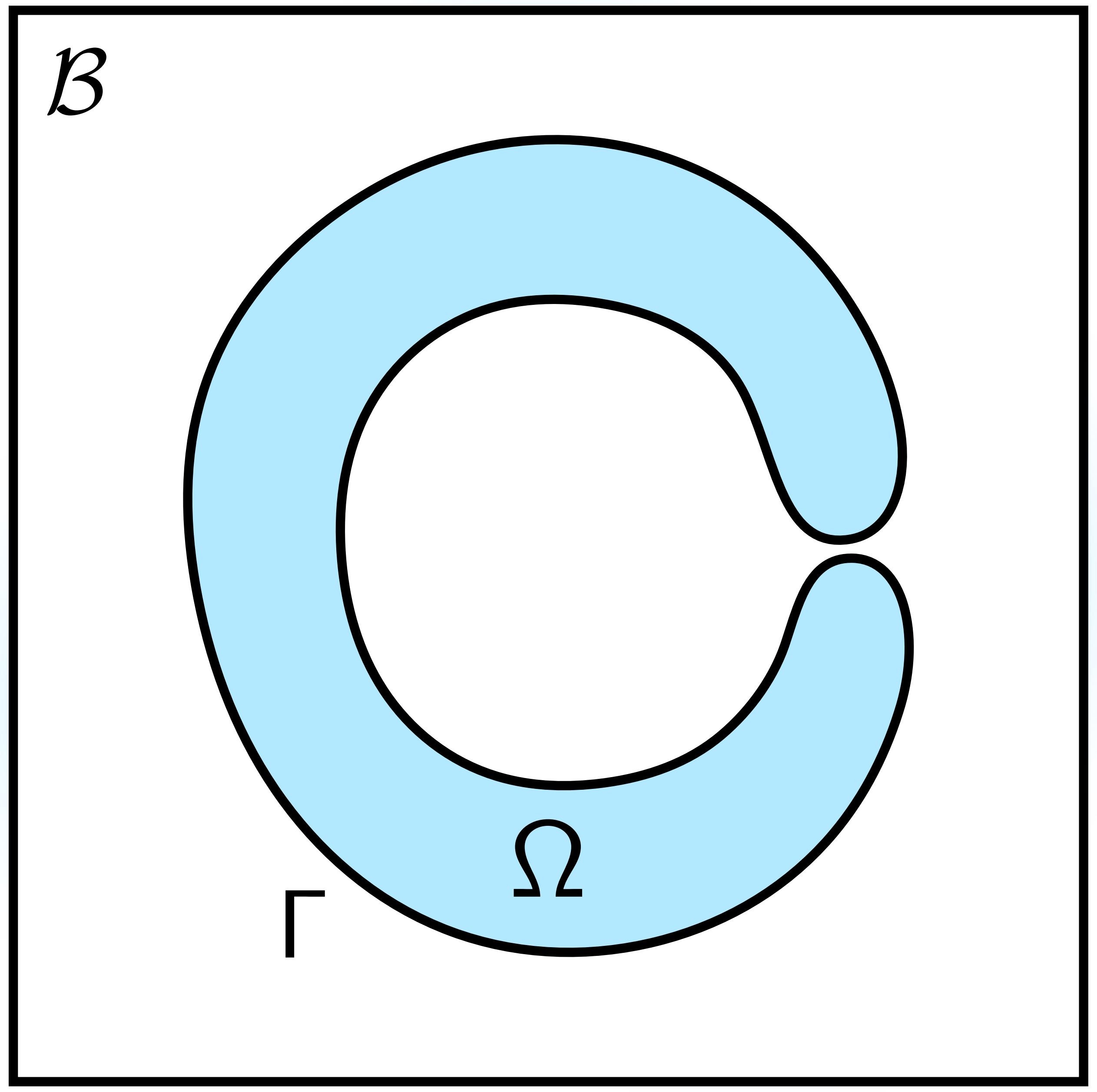
Prior work

- Finite difference extension, second-order accurate [Mayo, 1984]
- Fourier continuation [Bruno & Lyon, 2010], [Bruno & Paul, 2020]
- Immersed boundary smooth extension [Stein, Guy, & Thomases, 2015]
- C^k polyharmonic extension + box code, fourth-order accurate [Askham & Cerfon, 2017]
- Partition of unity extension [Fryklund, Lehto, & Tornberg, 2018], [Fryklund & Greengard, 2022]

*Smoothly extending functions in a **robust** way is **challenging**!*

- multiscale geometry
- multiscale data
- close-to-touching regions

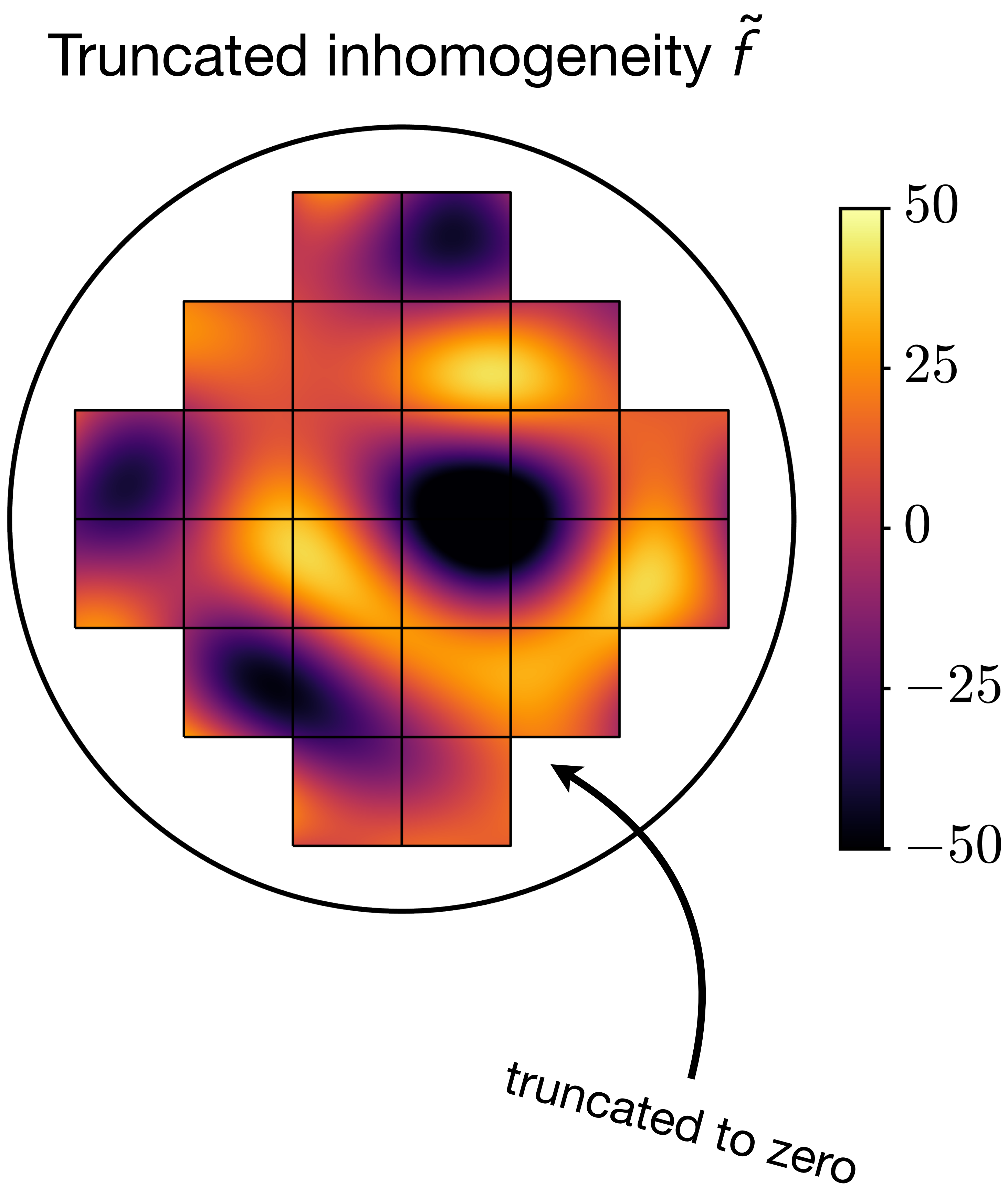
$\tilde{f}(x)$...?



Q: What if we truncate instead of extend?

$$v(x) = \int_{\mathcal{B}} \Phi(x, y)\, \tilde{f}(y)\, dy$$

$$v(x) = \int_{\mathcal{B}} \Phi(x, y)\, \tilde{f}(y)\, dy$$

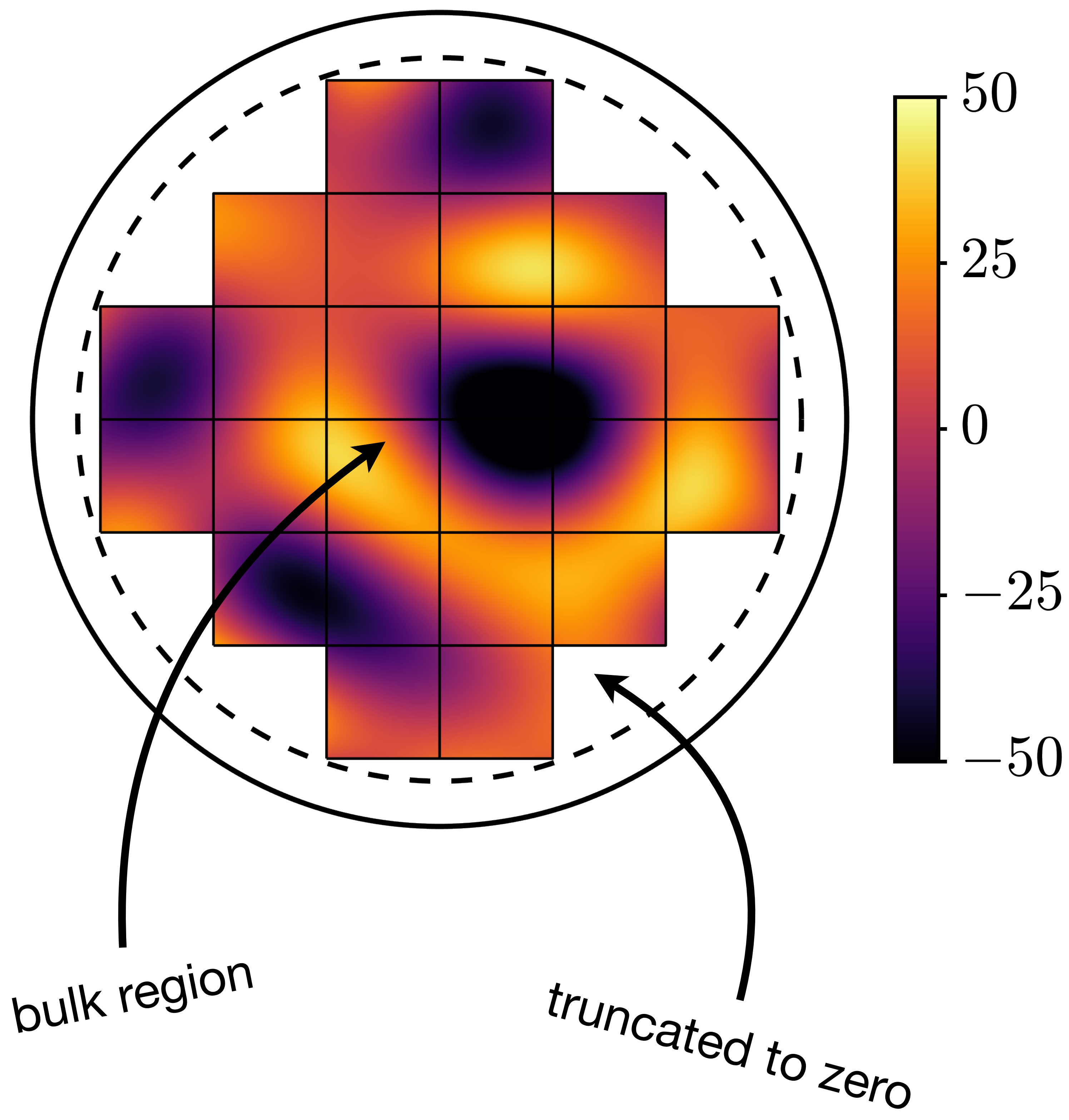


Potential theory

Truncating a volume potential

$$v(x) = \int_{\mathcal{B}} \Phi(x, y)\, \tilde{f}(y)\, dy$$

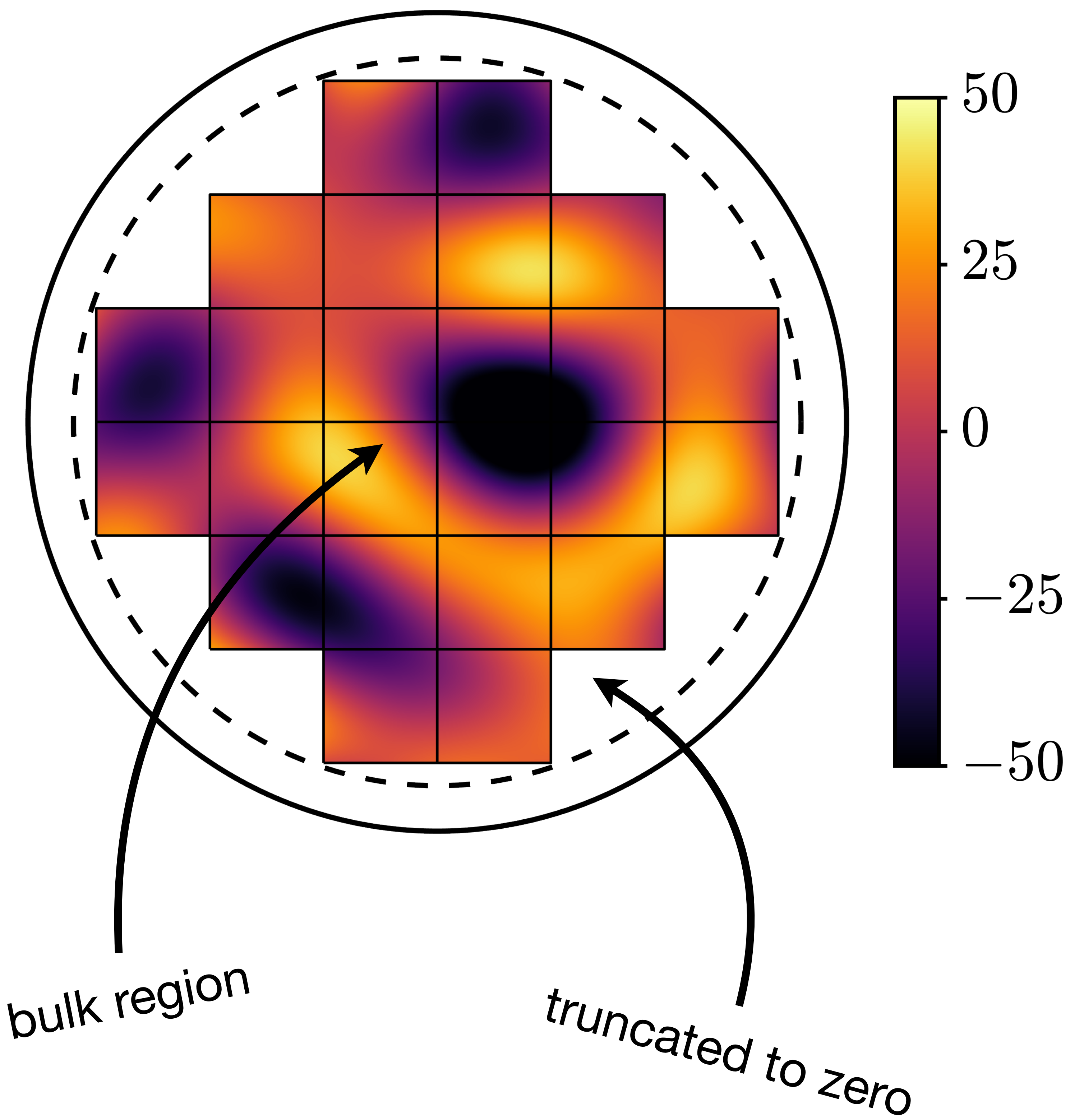
Truncated inhomogeneity $\tilde{f}$

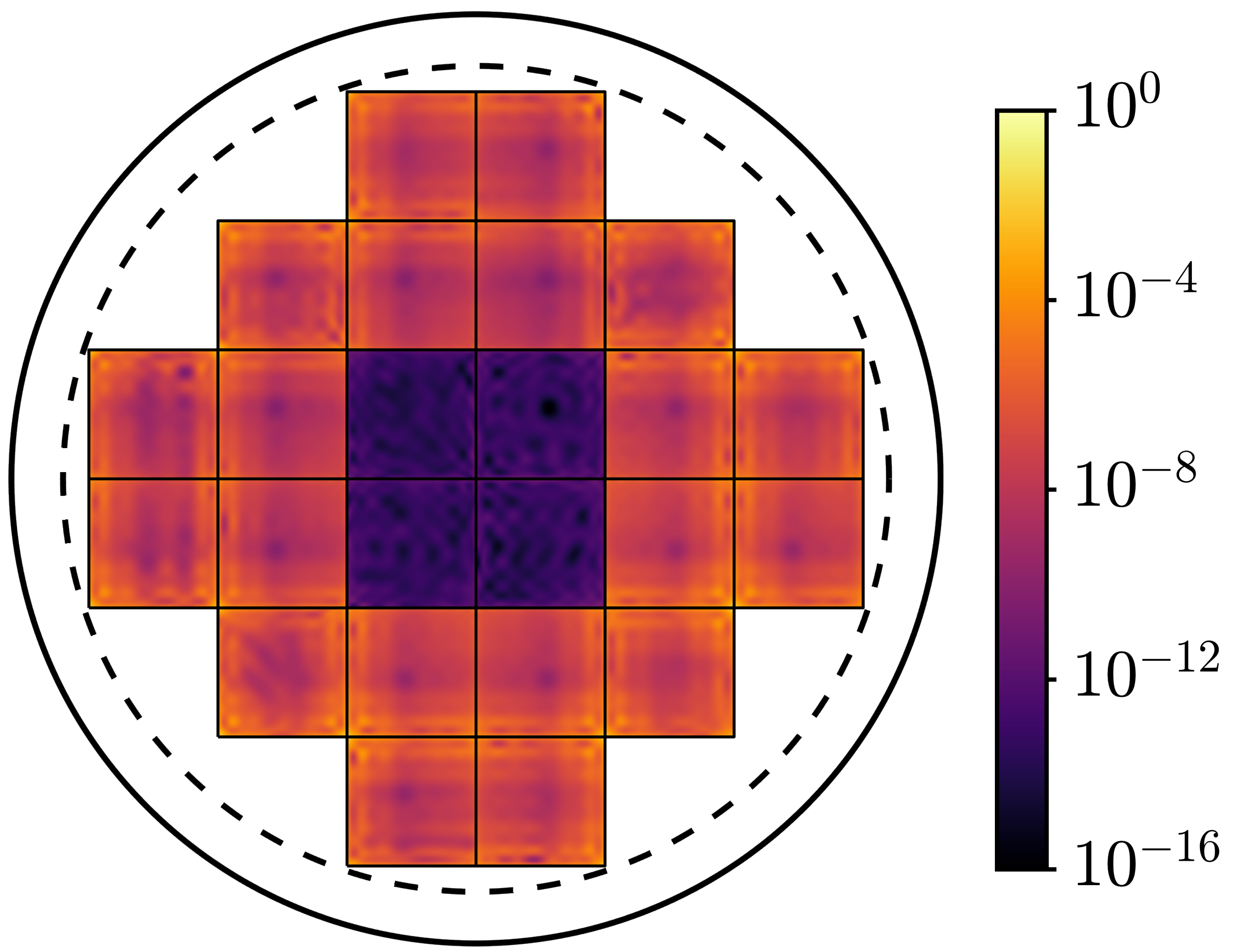


Potential theory

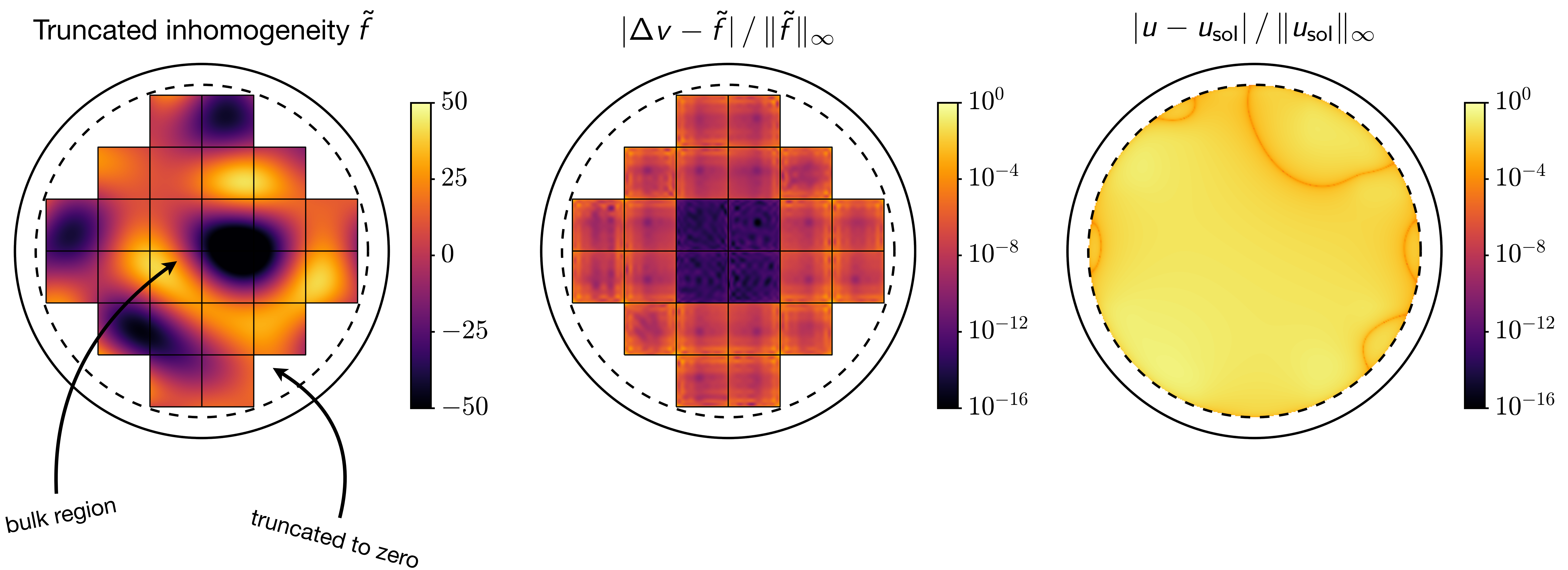
Truncating a volume potential

$$v(x) = \int_{\mathcal{B}} \Phi(x, y)\, \tilde{f}(y)\, dy$$



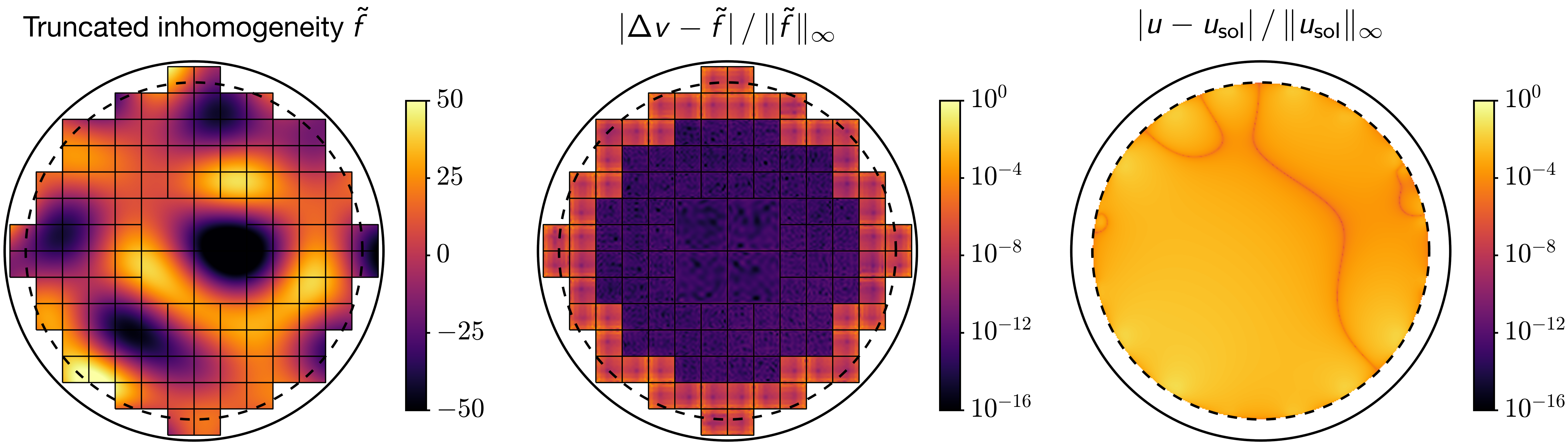


$$v(x) = \int_{\mathcal{B}} \Phi(x,y)\, \tilde{f}(y)\, dy$$



Truncating a volume potential

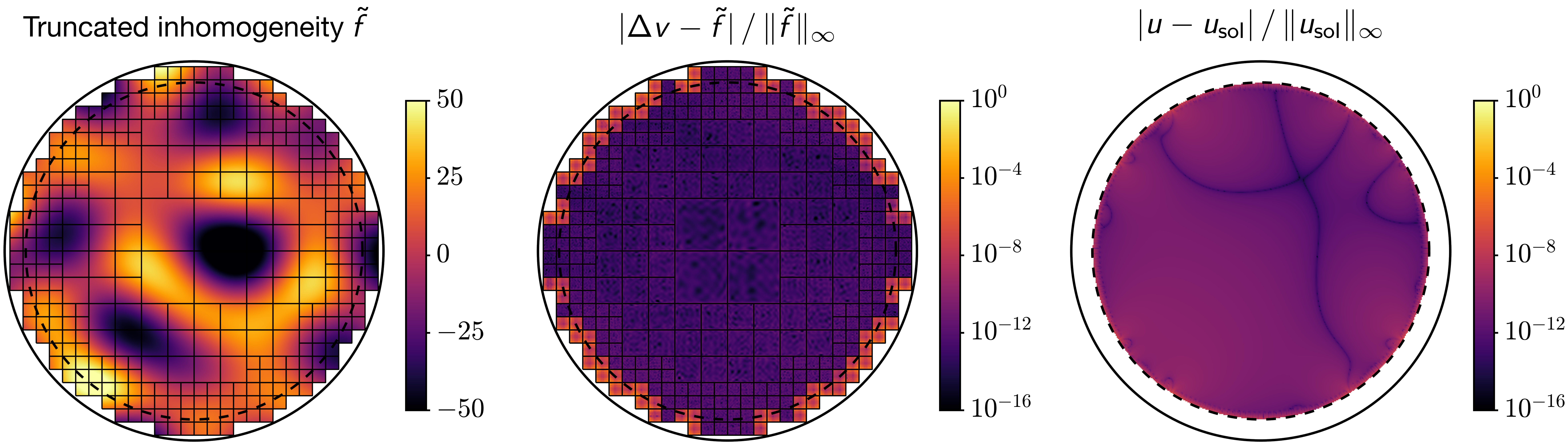
$$v(x) = \int_{\mathcal{B}} \Phi(x, y)\, \tilde{f}(y)\, dy$$



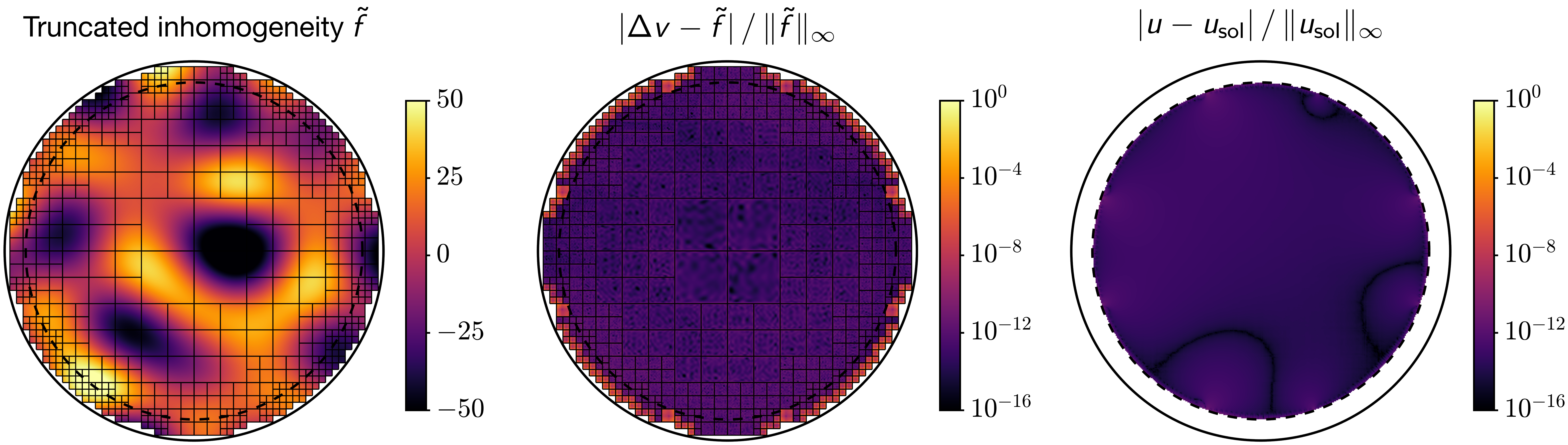
Potential theory

Truncating a volume potential

$$v(x) = \int_{\mathcal{B}} \Phi(x, y)\, \tilde{f}(y)\, dy$$

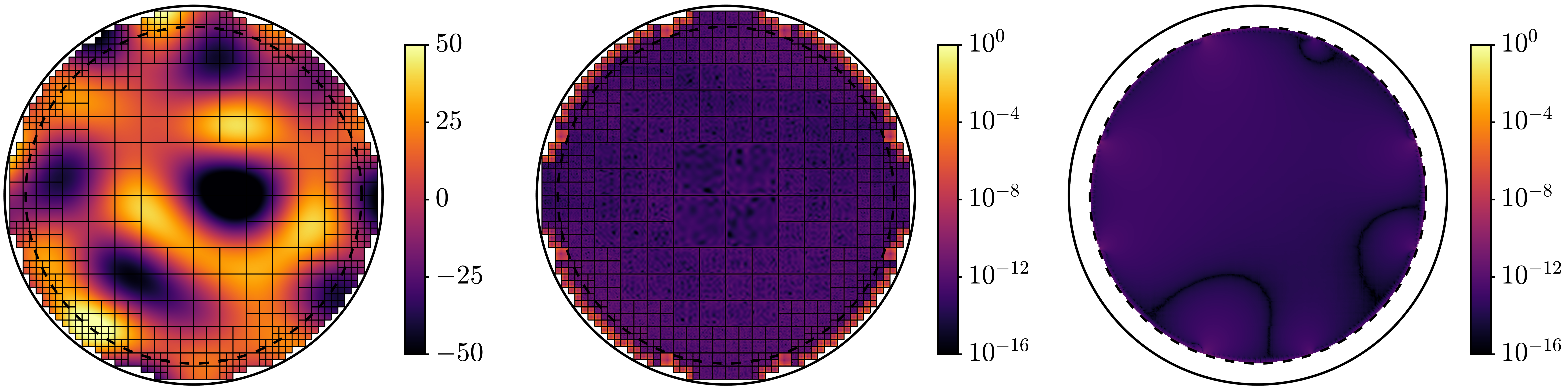


$$v(x) = \int_{\mathcal{B}} \Phi(x, y)\, \tilde{f}(y)\, dy$$



$$v(x) = \int_{\mathcal{B}} \Phi(x, y)\, \tilde{f}(y)\, dy$$

Truncated inhomogeneity $\tilde{f}$ | $|\Delta v - \tilde{f}| \,/\, \|\tilde{f}\|_\infty$ | $|u - u_{\mathrm{sol}}| \,/\, \|u_{\mathrm{sol}}\|_\infty$

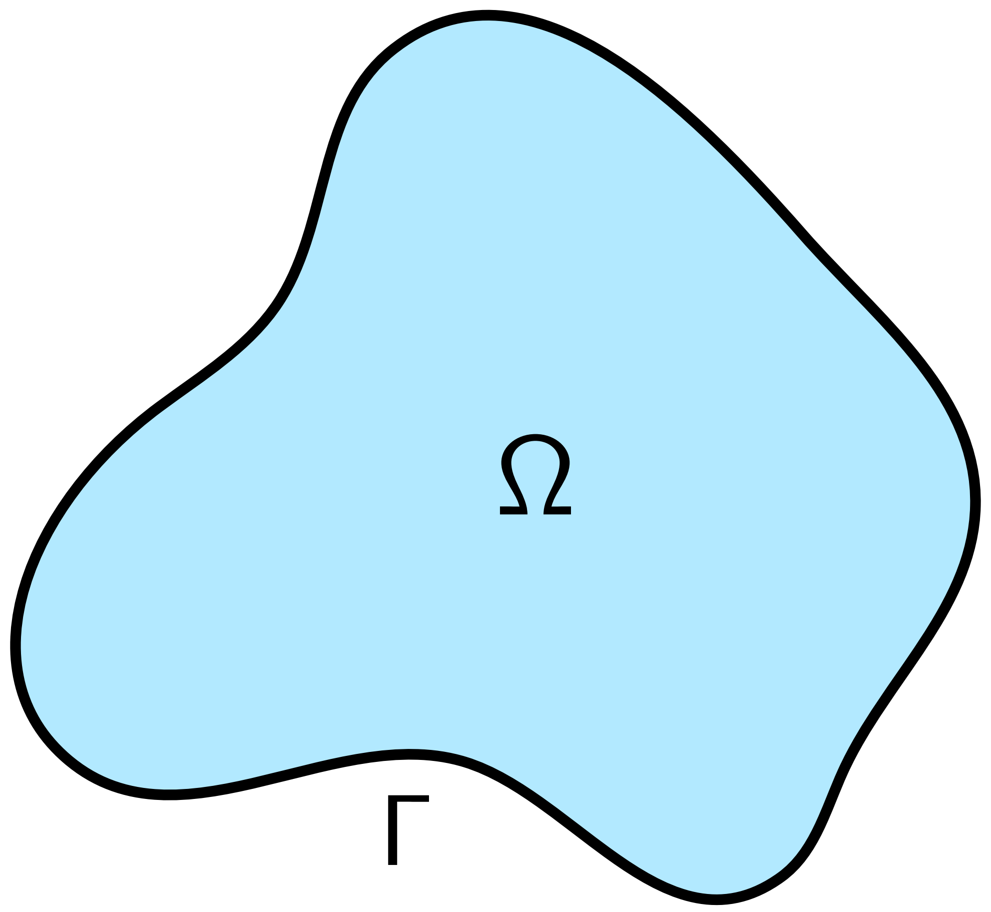


Theorem (F., Stein, & Barnett)

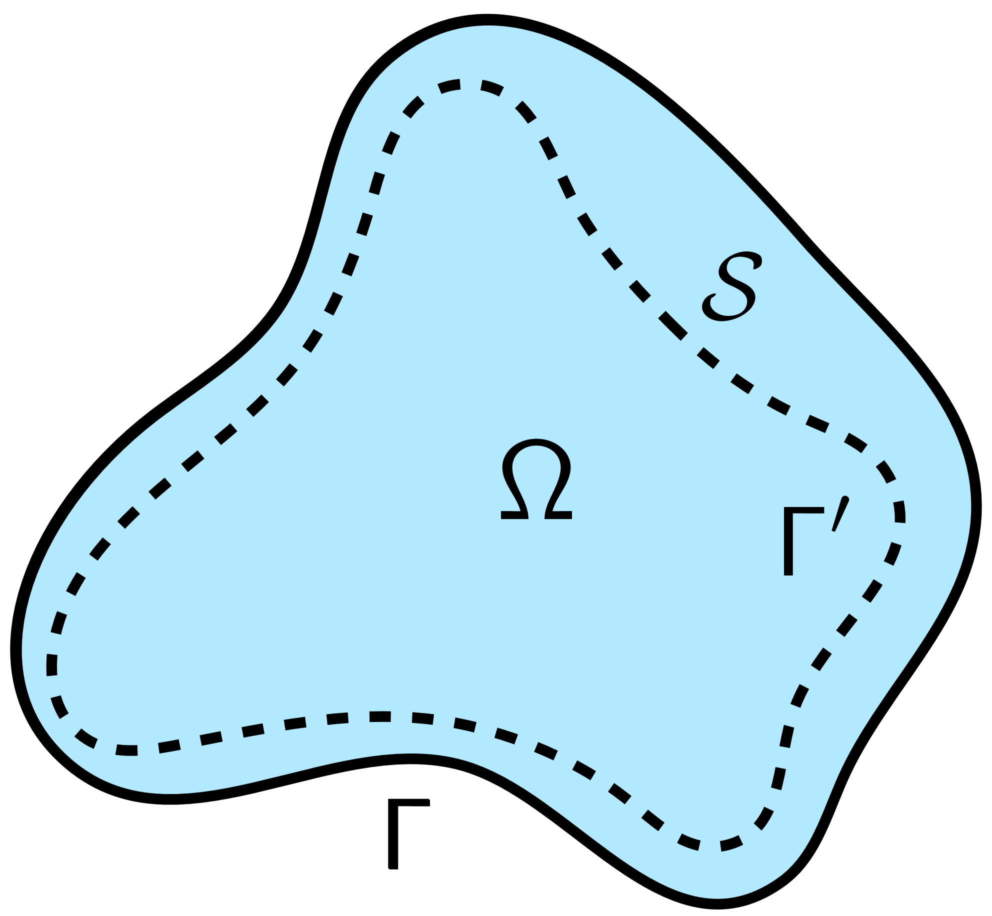
Let the truncated region $\operatorname{supp}(f - \tilde{f})$ *be well separated from each bulk box. Then the volume potential v induced by $\tilde{f}$ is as smooth on each bulk box as that induced by the original f.*

Inhomogeneous PDEs

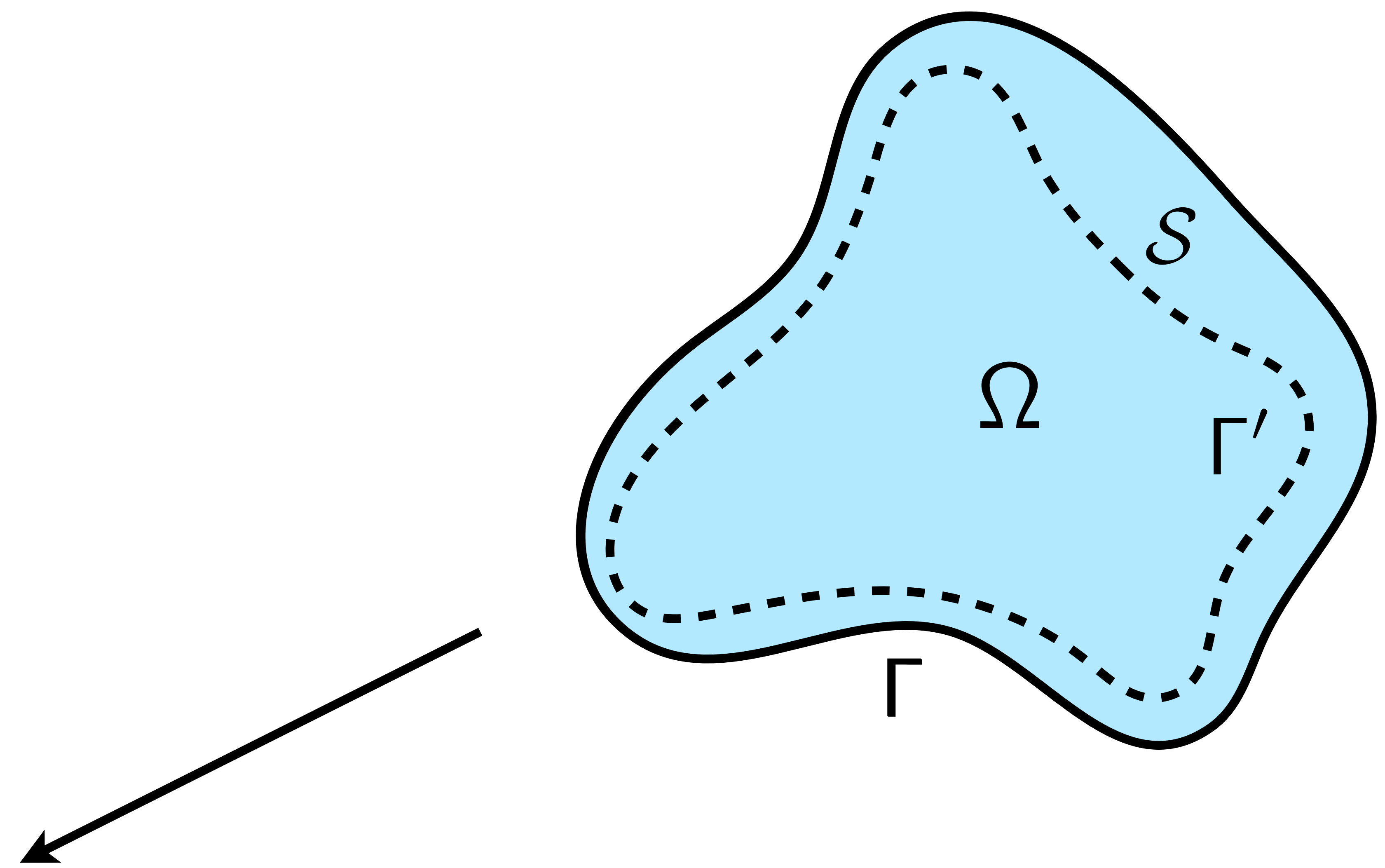
Our approach

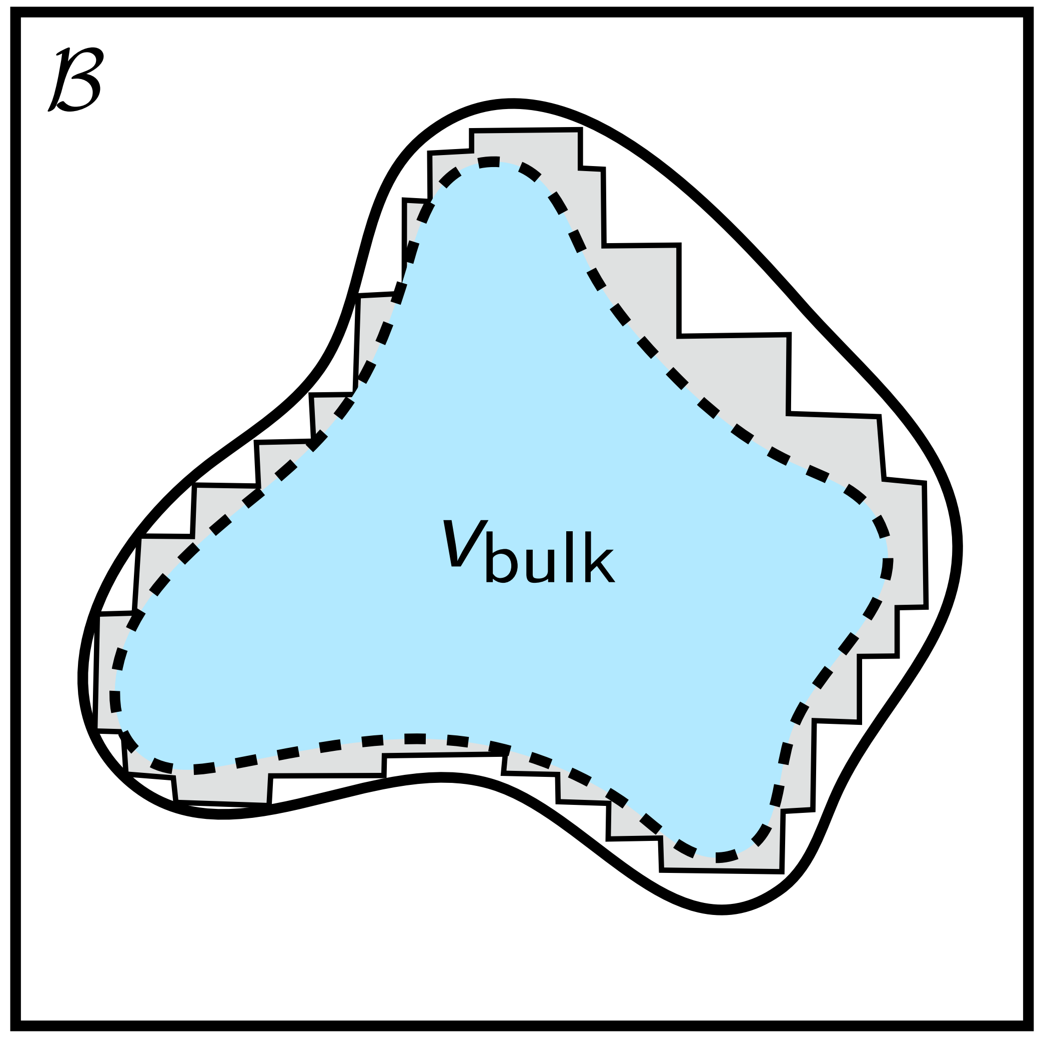


Inhomogeneous PDEs
Our approach



Inhomogeneous PDEs
Our approach

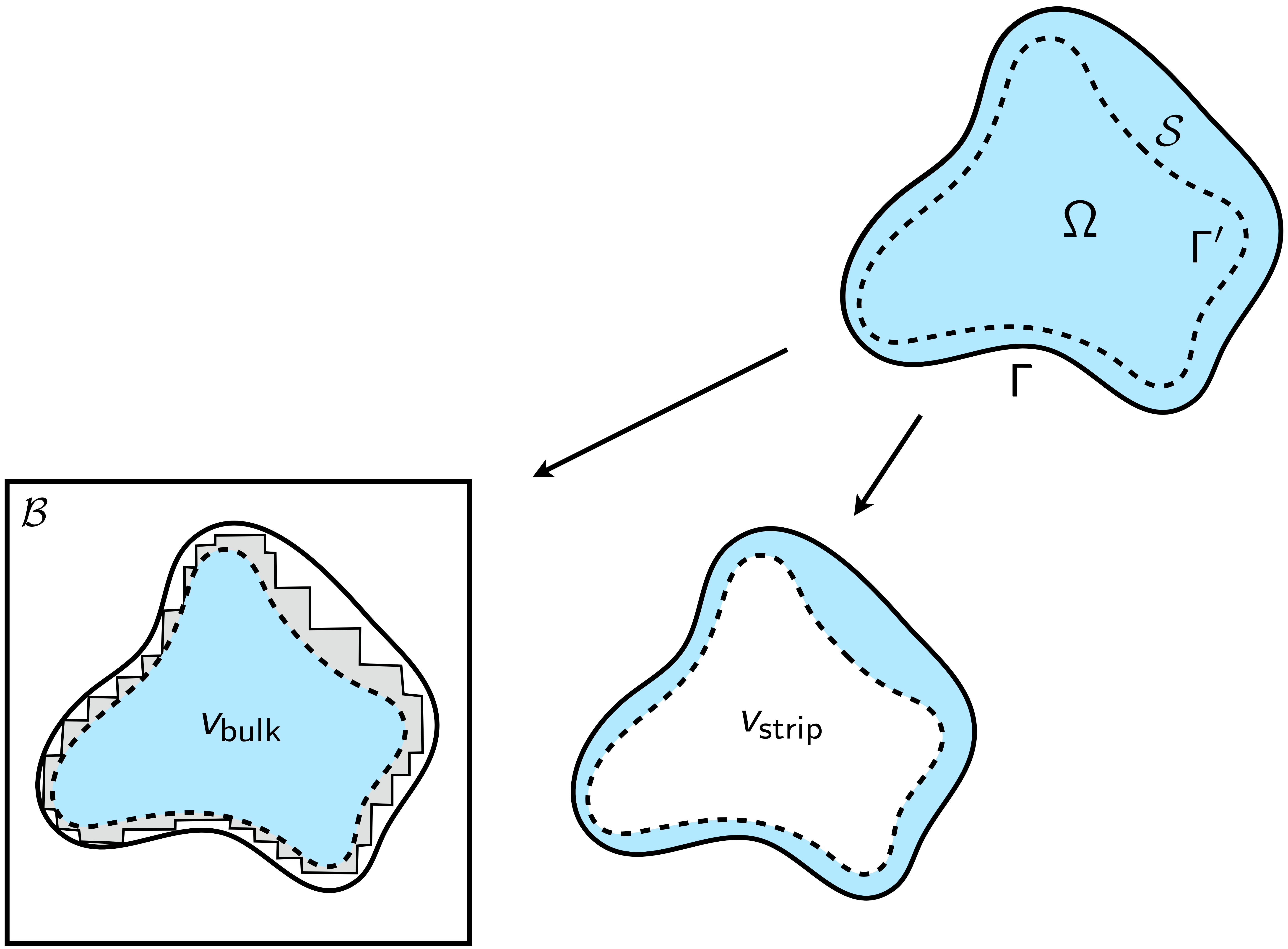




Truncated volume potential

Inhomogeneous PDEs

Our approach

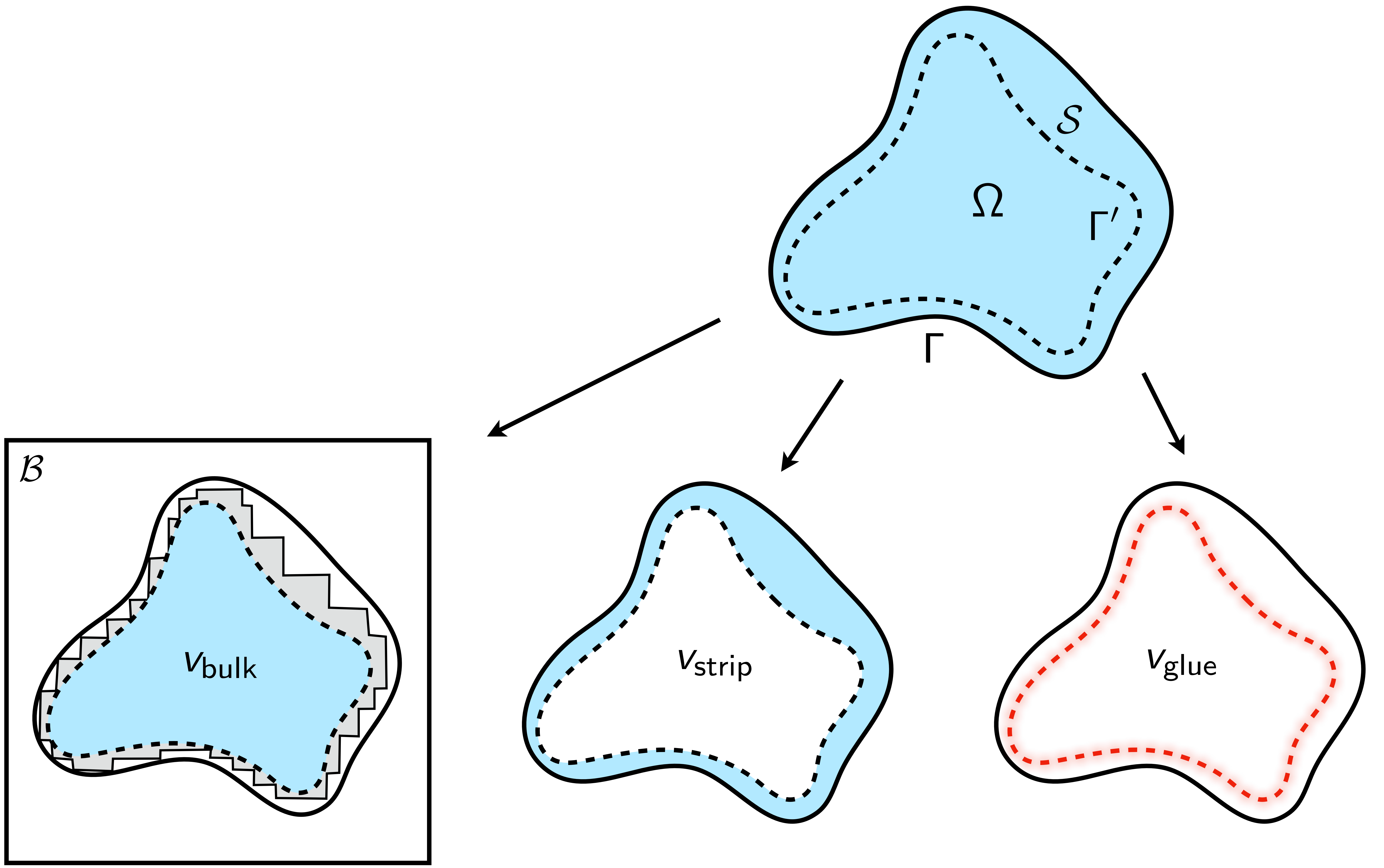


Truncated volume potential

Strip solve

Inhomogeneous PDEs

Our approach



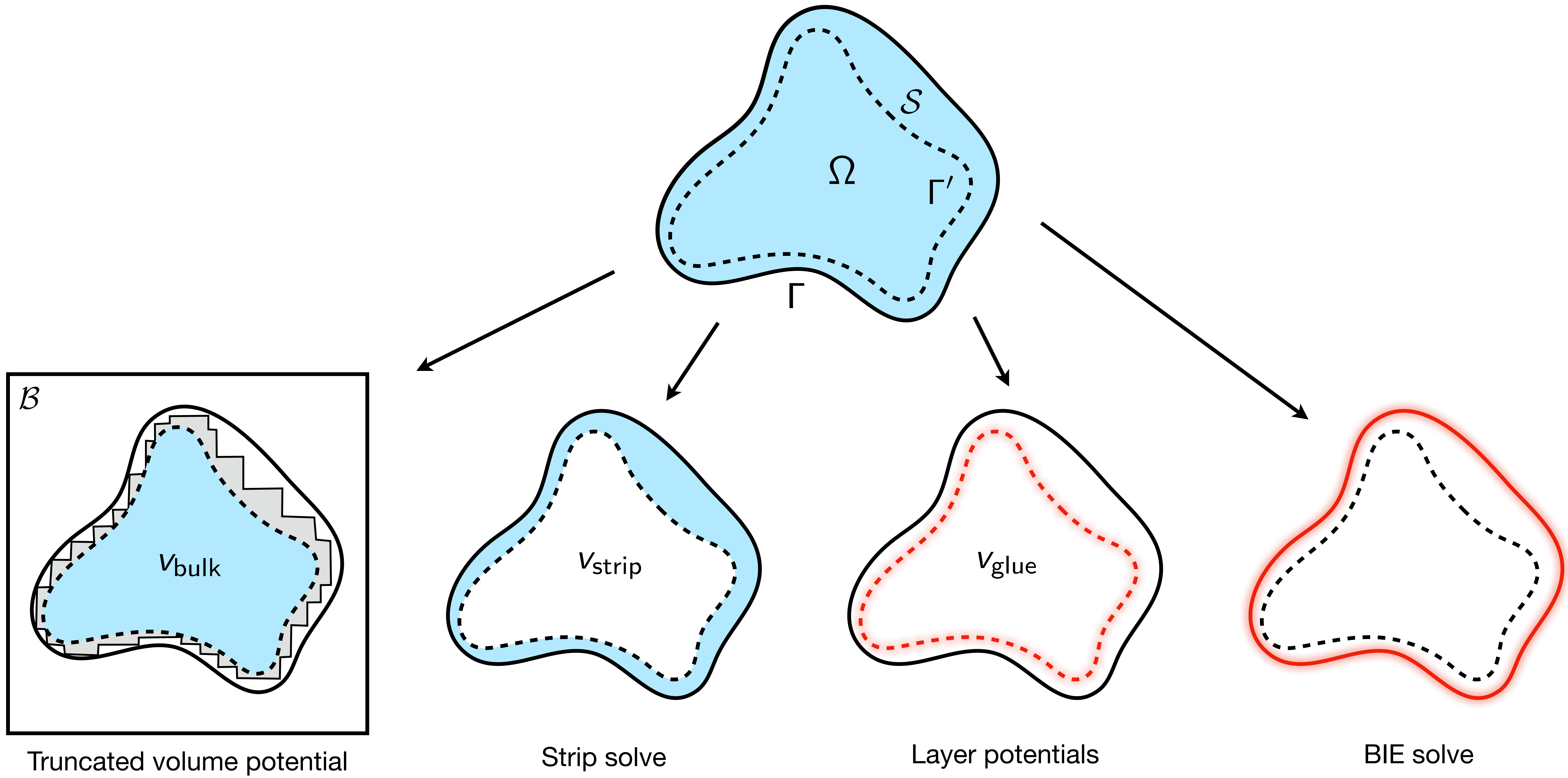
Truncated volume potential

Strip solve

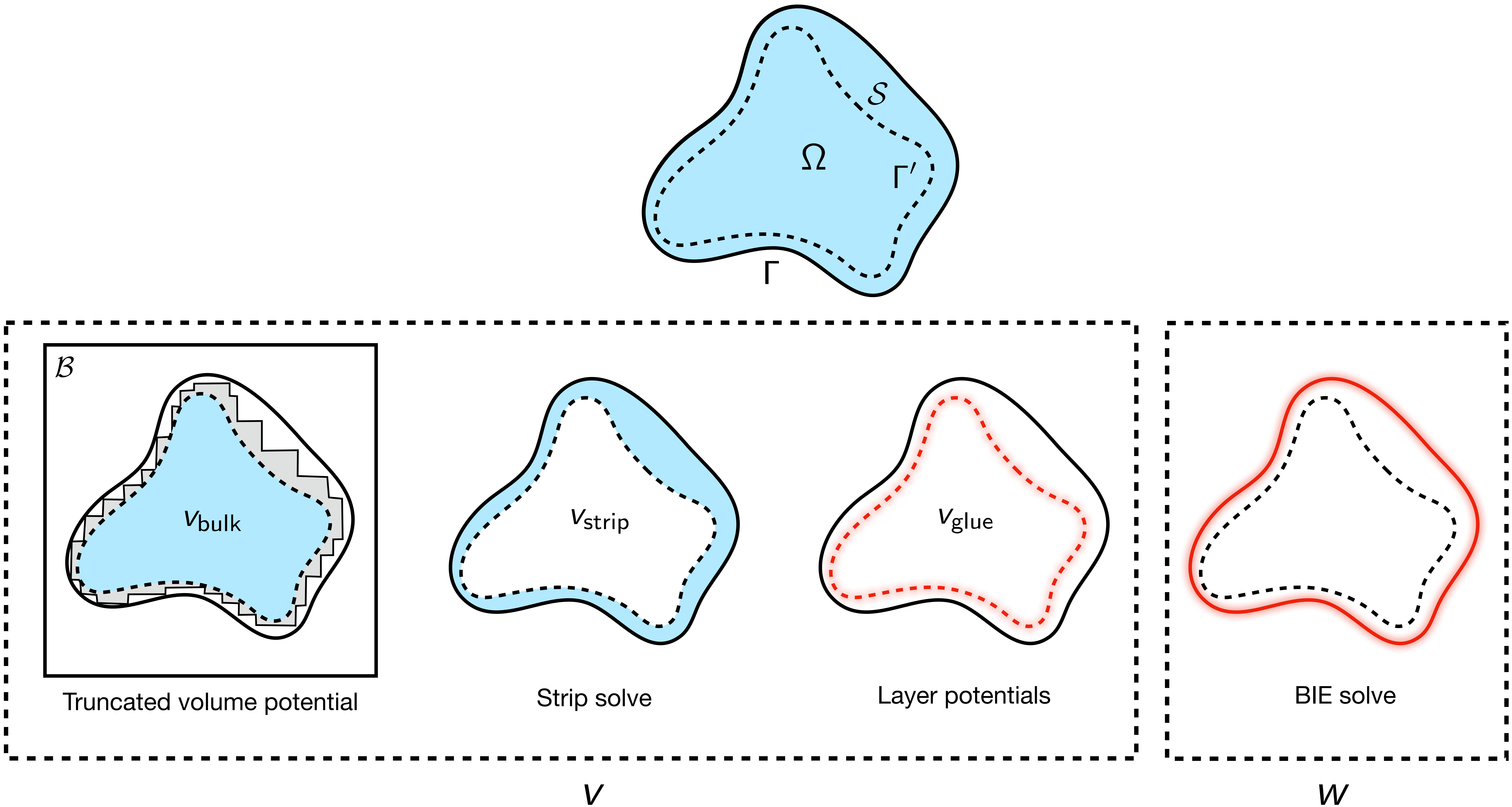
Layer potentials

Inhomogeneous PDEs

Our approach

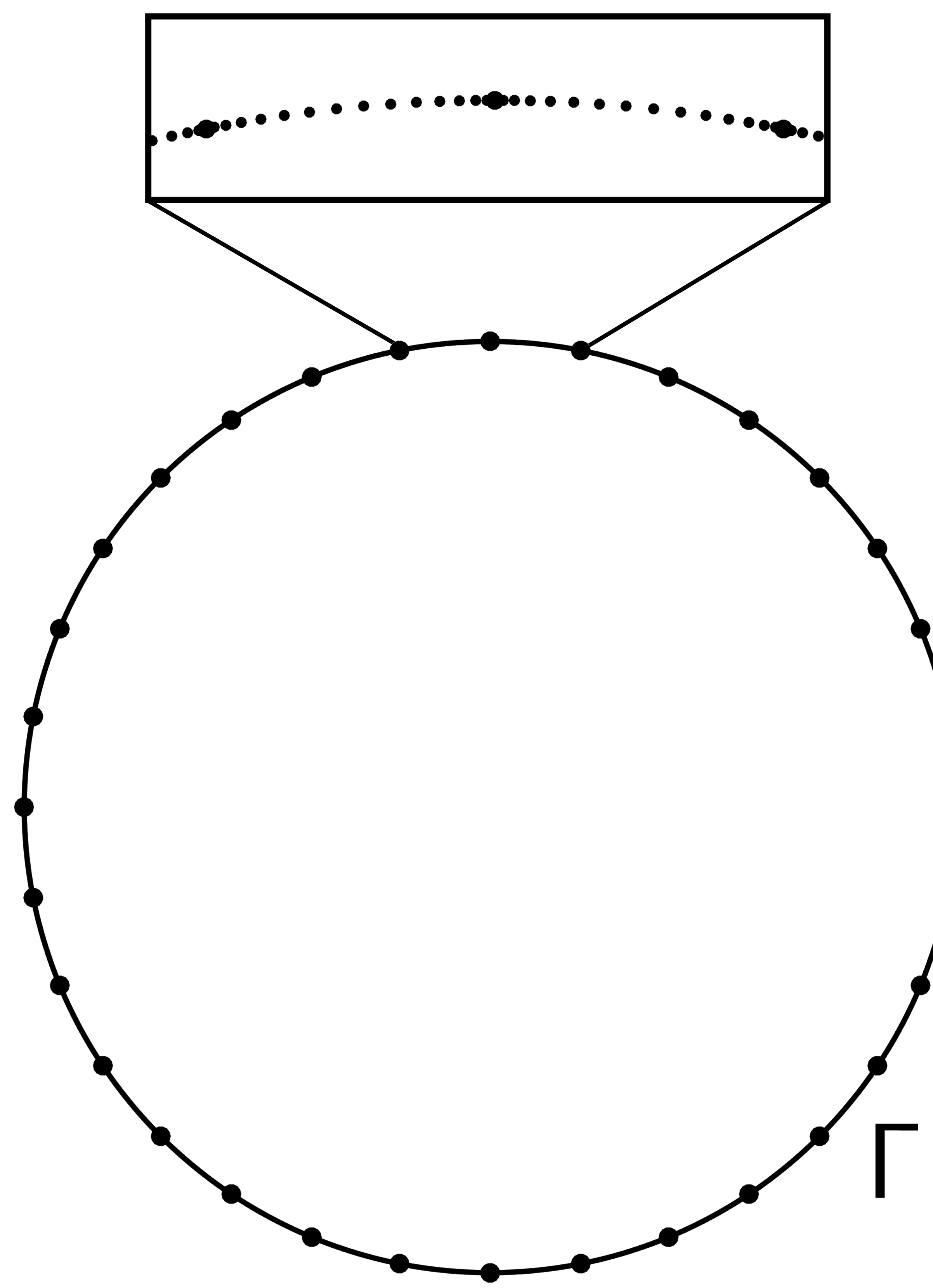


Inhomogeneous PDEs
Our approach



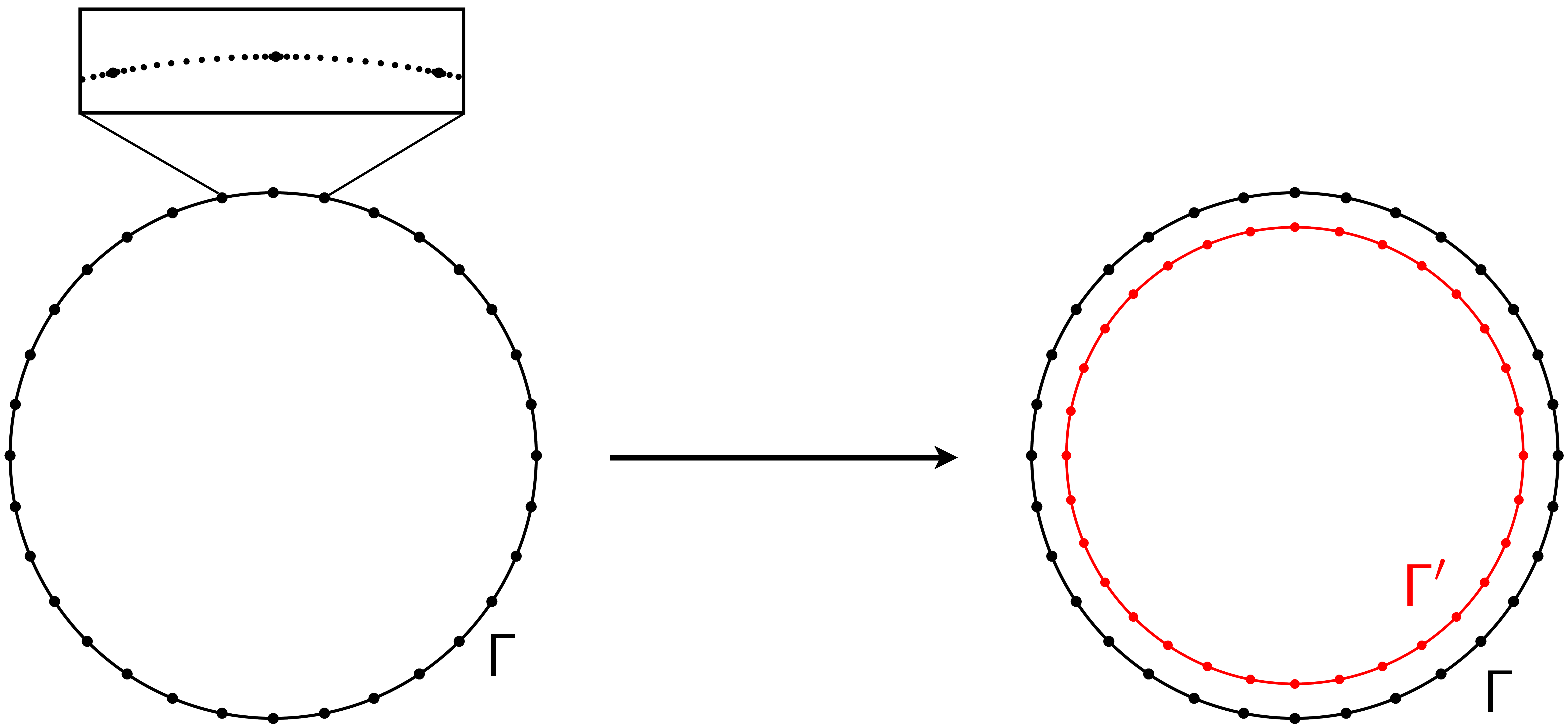
Defining the truncation region

What happens on an adaptive geometry?



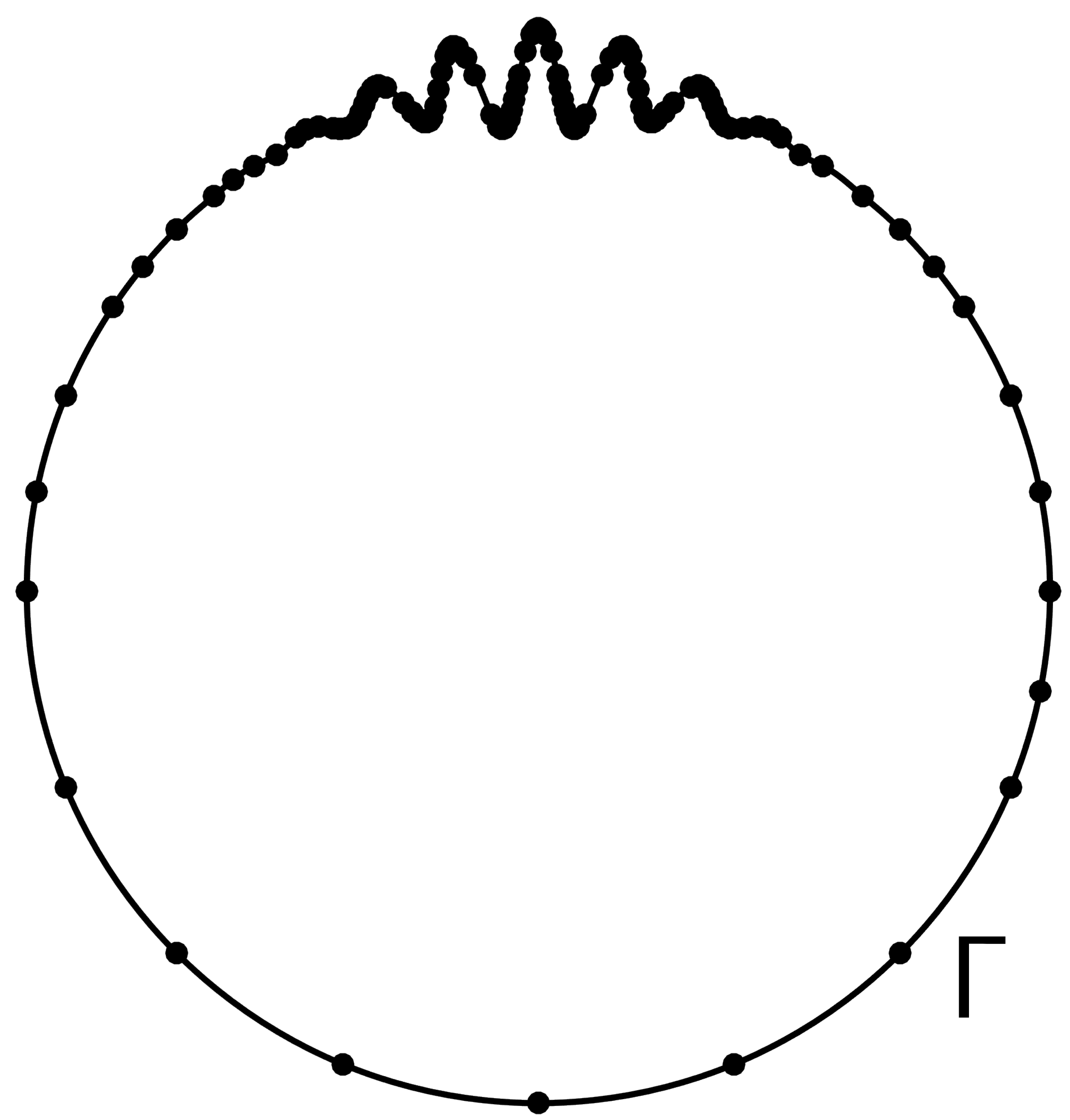
Defining the truncation region

What happens on an adaptive geometry?



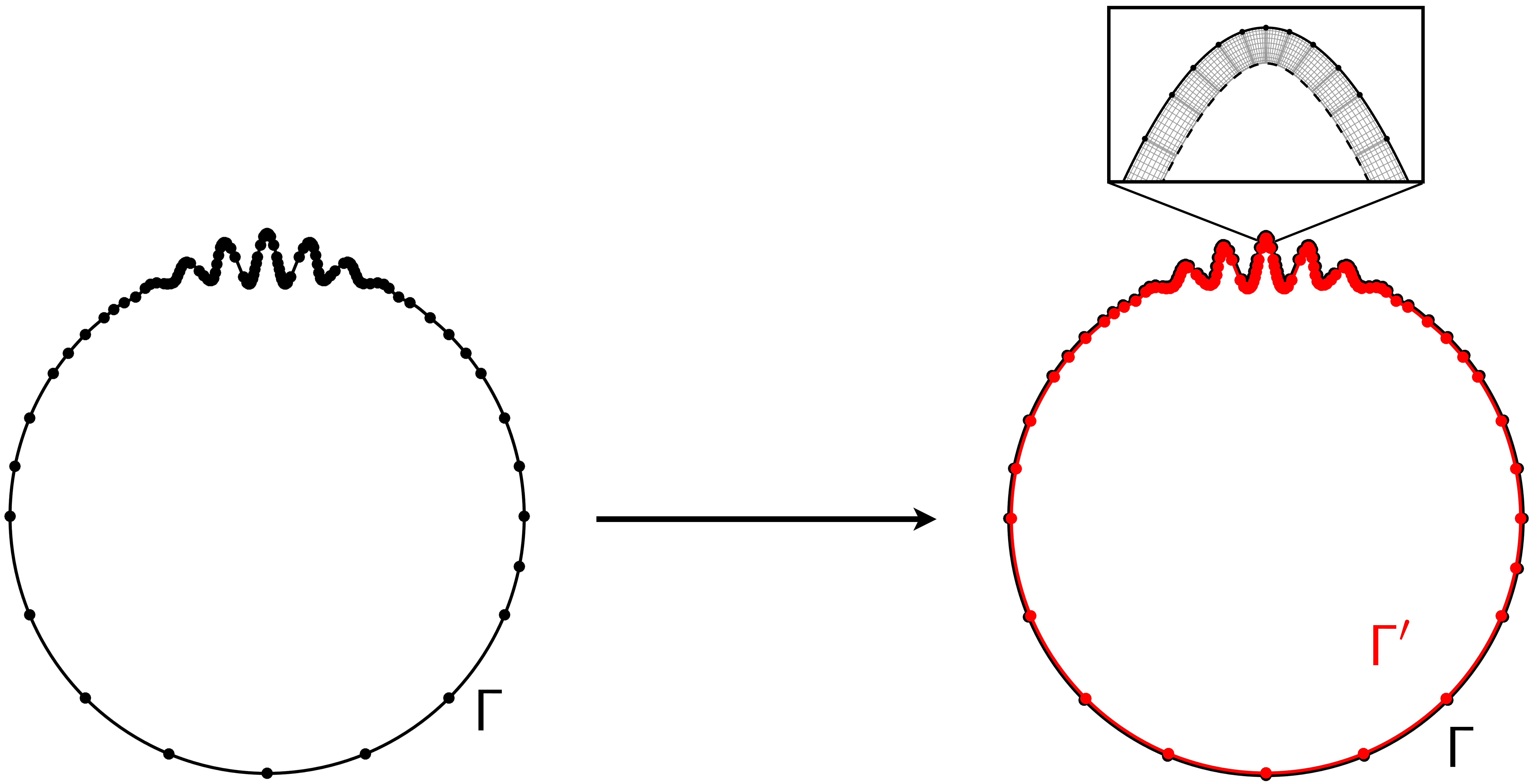
Defining the truncation region

What happens on an adaptive geometry?



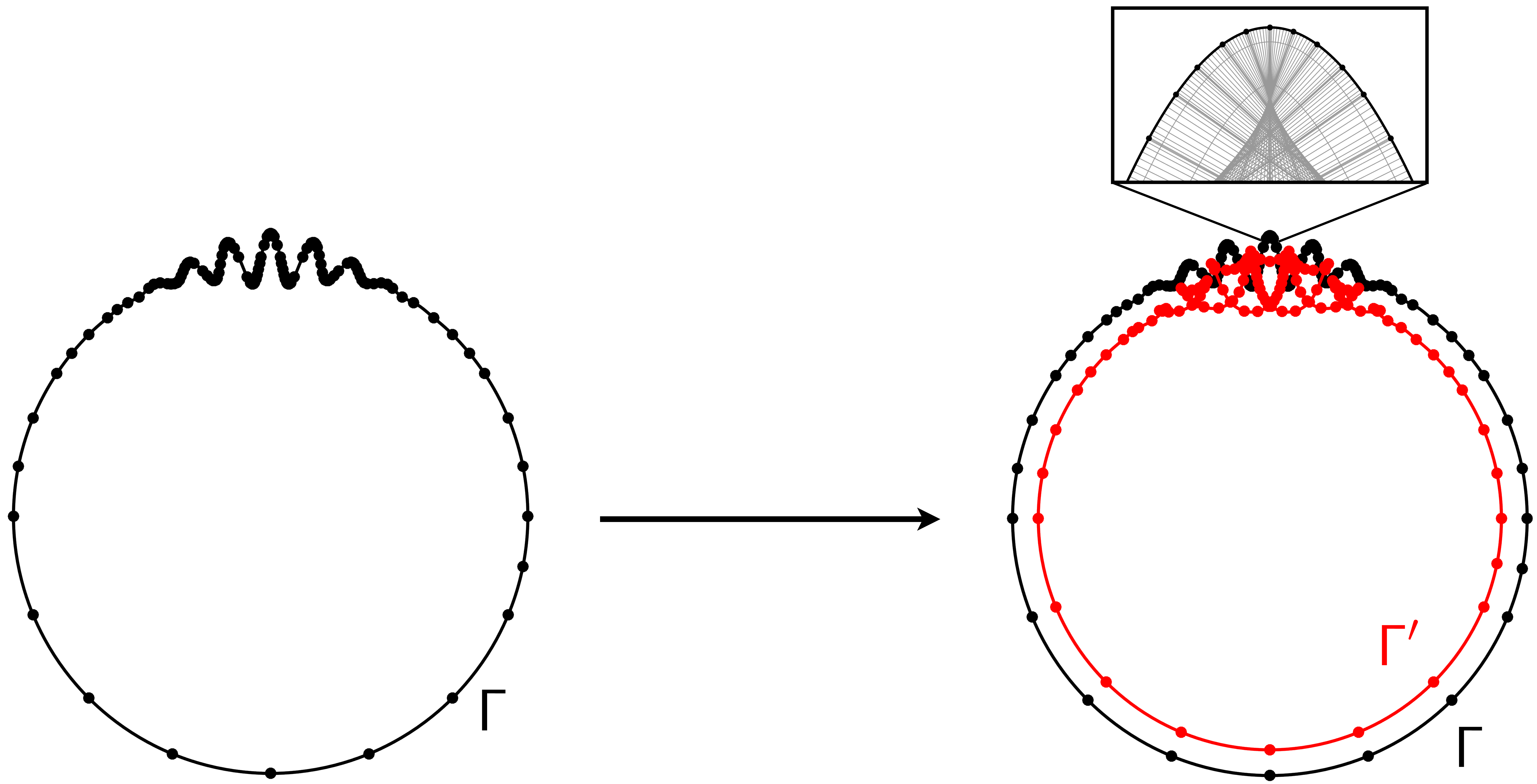
Defining the truncation region

What happens on an adaptive geometry?



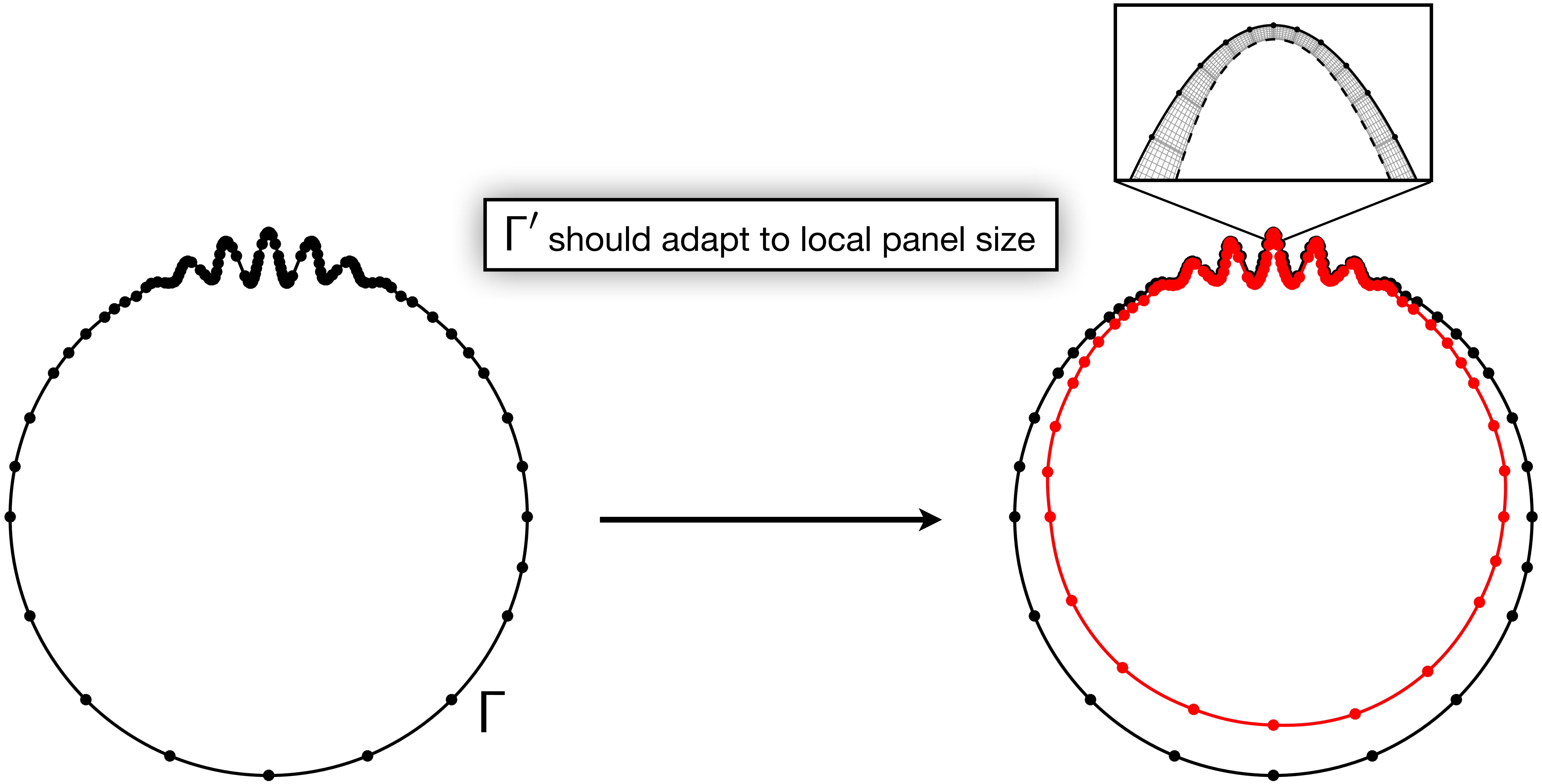
Defining the truncation region

What happens on an adaptive geometry?



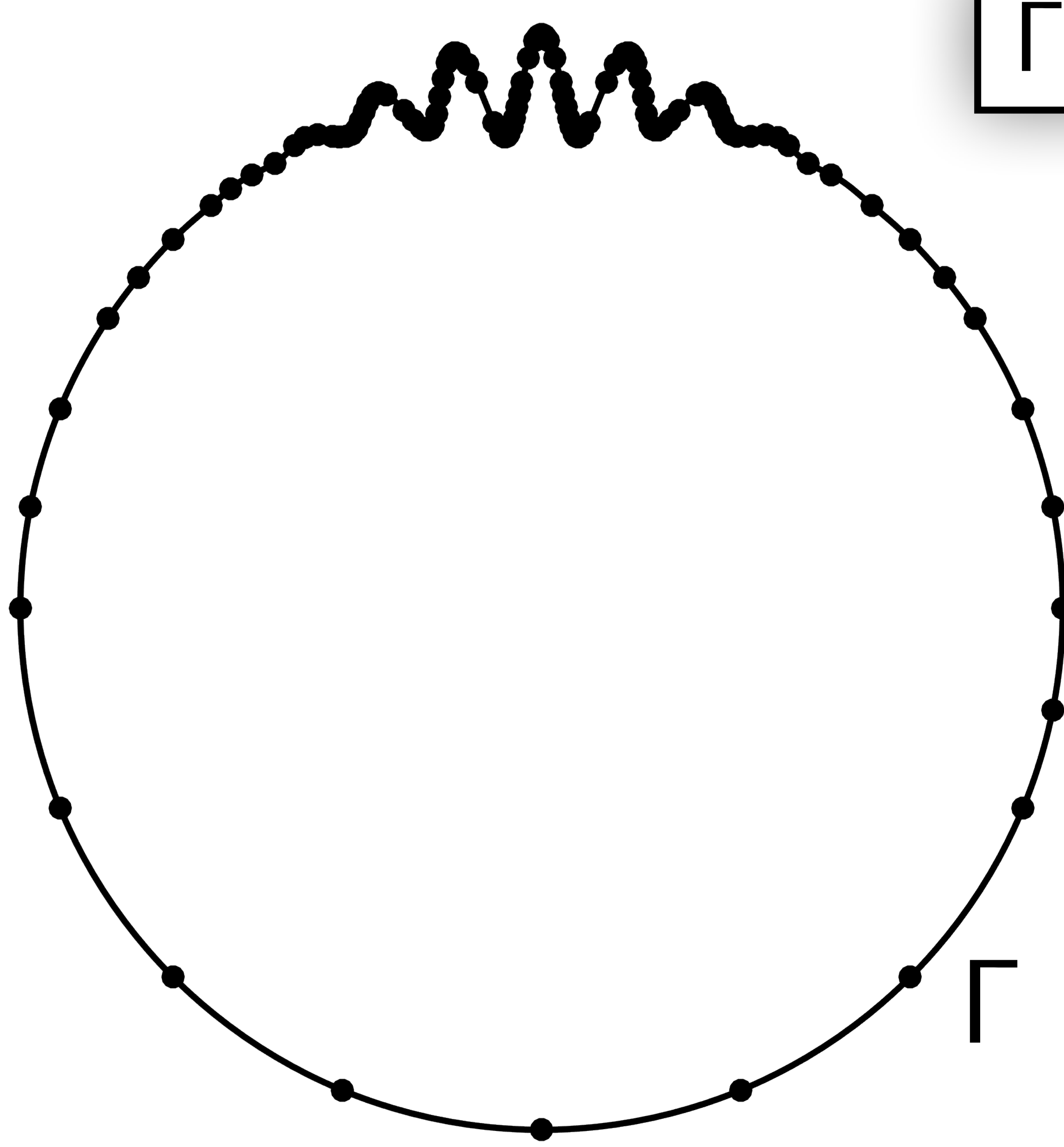
Defining the truncation region

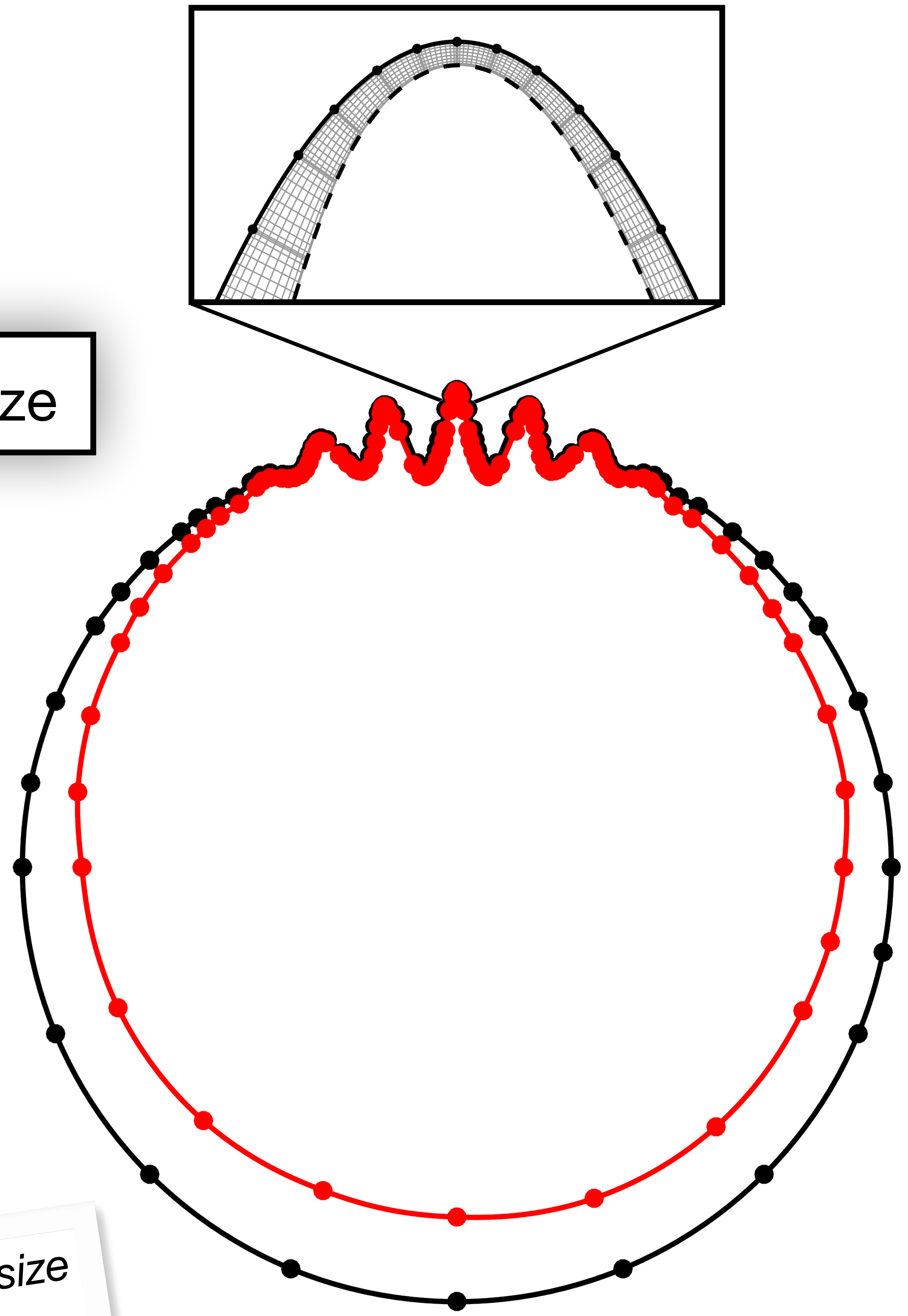
What happens on an adaptive geometry?



Defining the truncation region

What happens on an adaptive geometry?





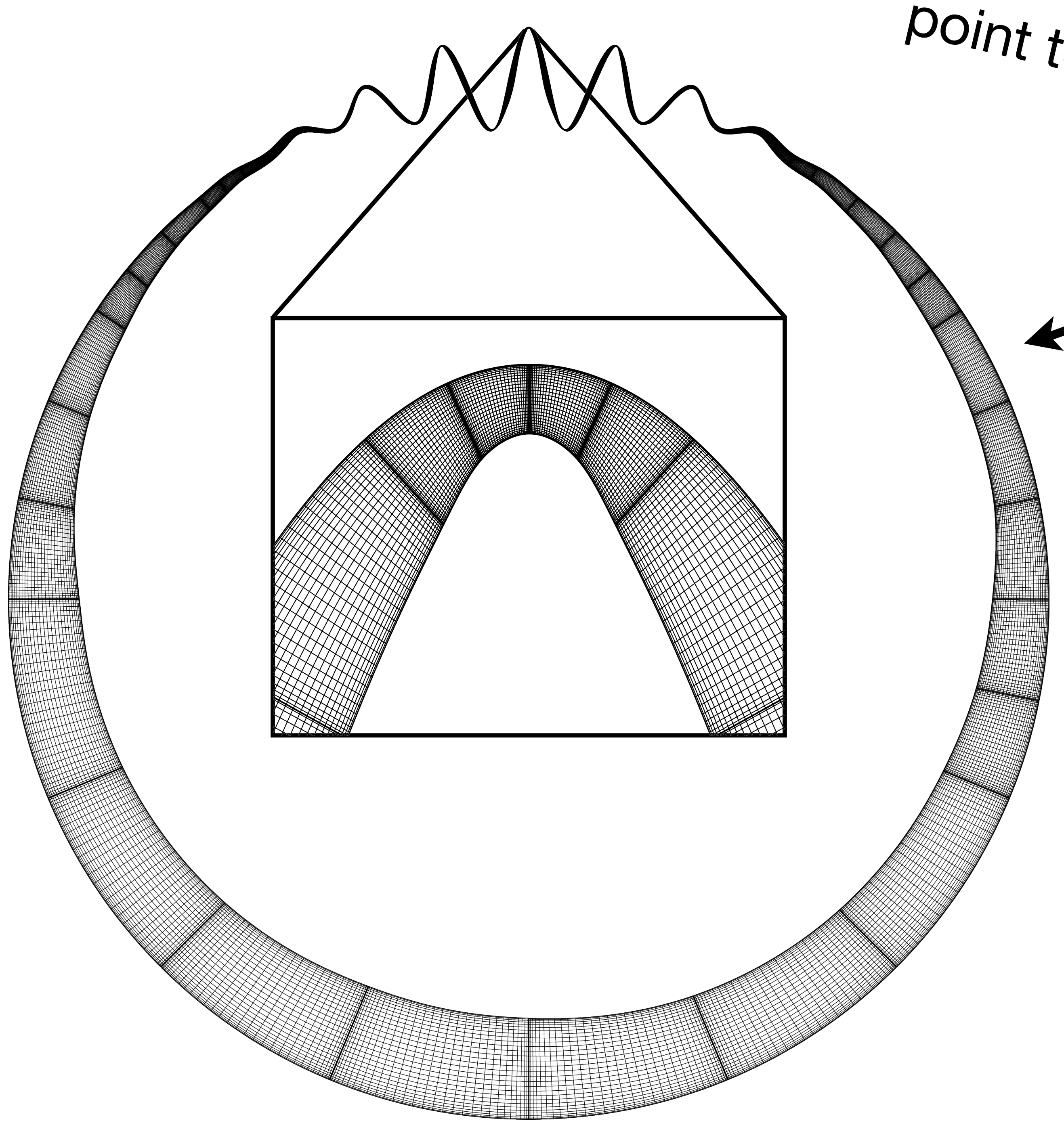
Γ' should adapt to local panel size

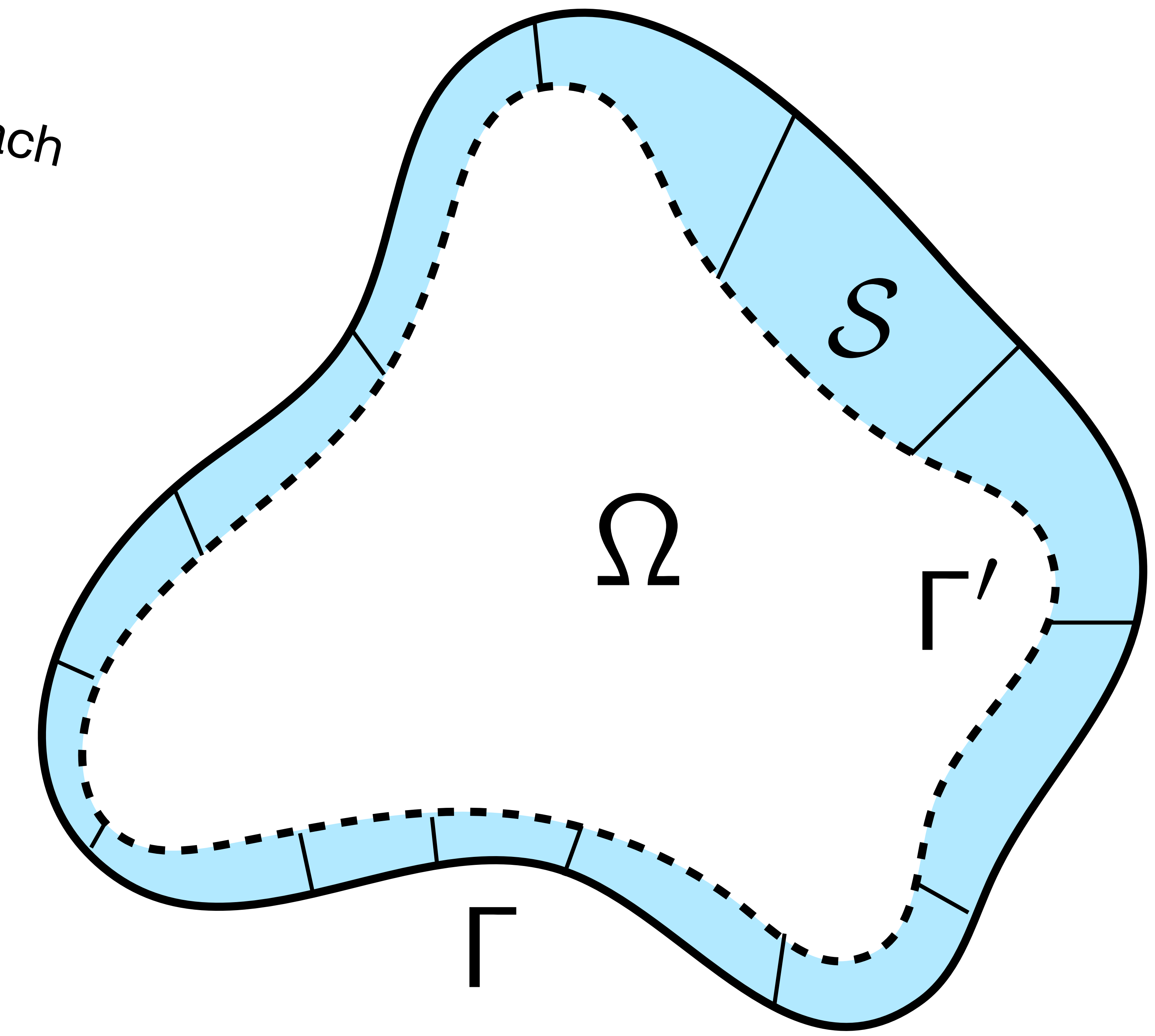
1. Define piecewise width from local panel size
2. Smooth each junction
3. Blend together using matched asymptotics
4. Perturb in the normal direction

$$\begin{aligned} \Delta v_{\text{strip}}(\boldsymbol{x}) &= f(\boldsymbol{x}), & \boldsymbol{x} &\in \mathcal{S}, \\ v_{\text{strip}}(\boldsymbol{x}) &= 0, & \boldsymbol{x} &\in \Gamma \cup \Gamma'. \end{aligned}$$

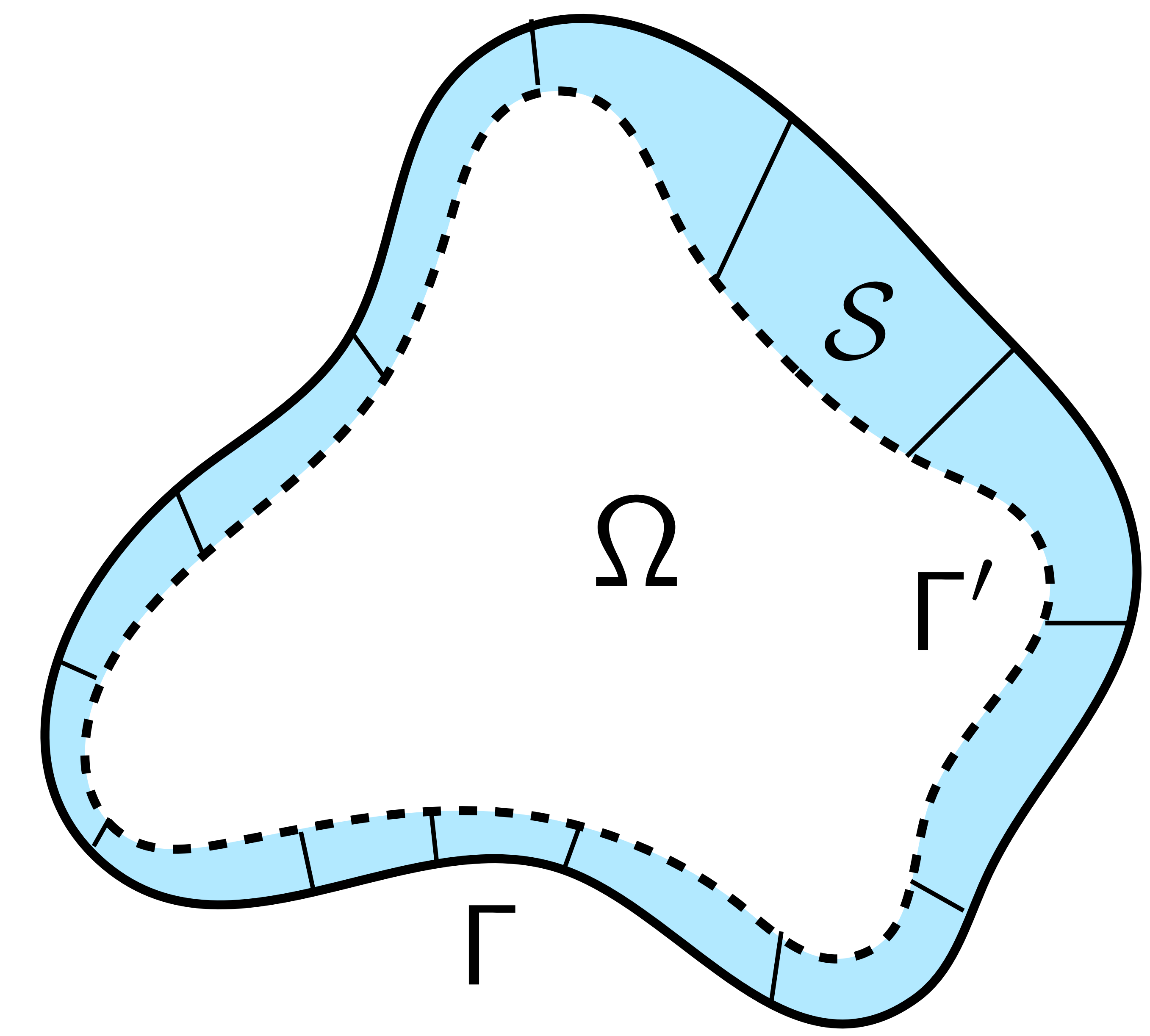
We use a spectral collocation method in $\mathcal{S}$ and solve using the “1D” HPS method.

Jacobian is numerically computed at each point to form differentiation matrices.





We use the hierarchical Poincaré–Steklov scheme to build a **fast direct solver** in $\mathcal{S}$.



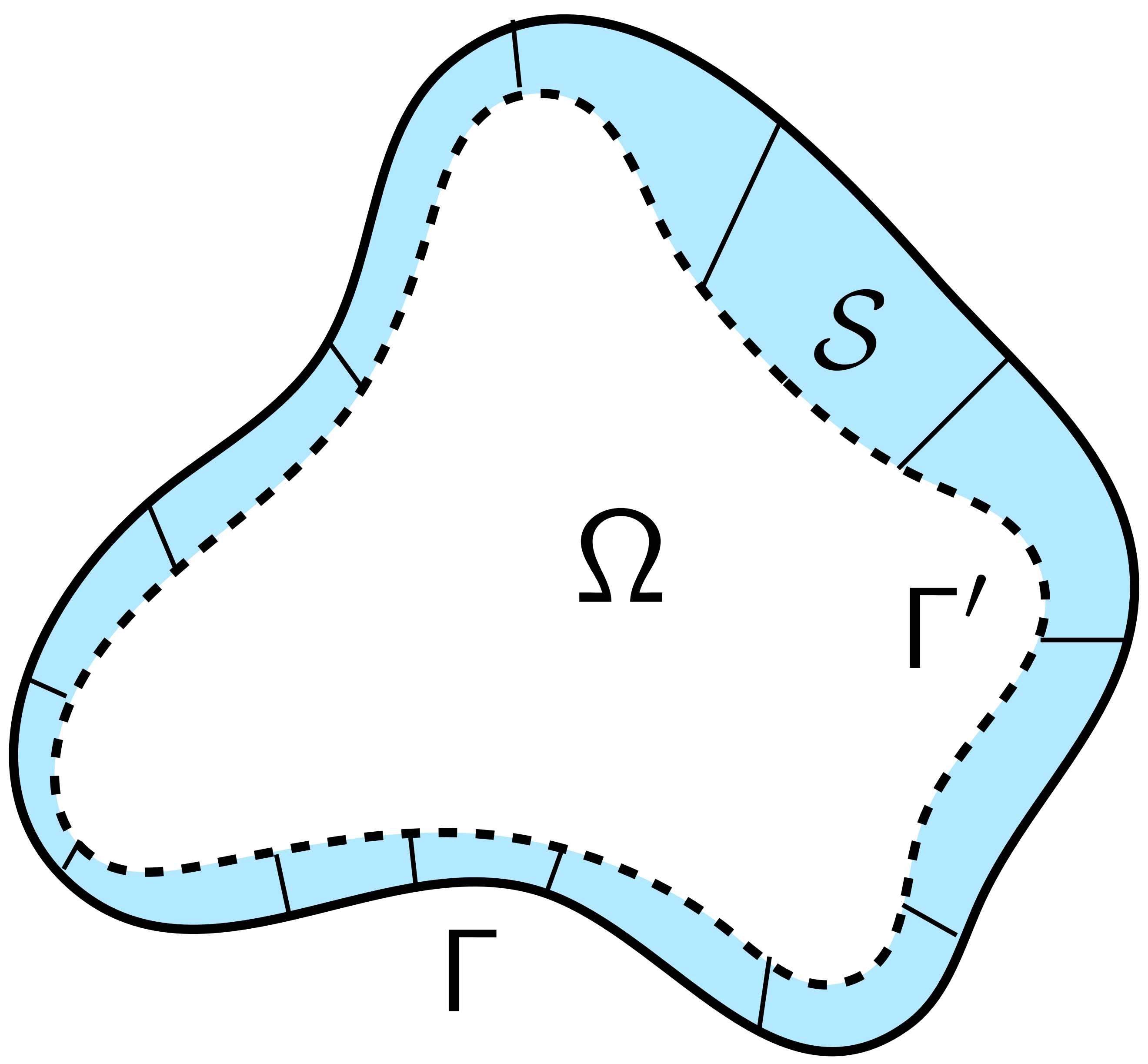
[Martinsson, 2013]
[Gillman & Martinsson, 2014]

We use the hierarchical Poincaré–Steklov scheme to build a **fast direct solver** in $\mathcal{S}$.

Given an inhomogeneity f:

① On each element, compute:

- Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
- Dirchlet-to-Neumann map: $DtN \in \mathbb{R}^{4n \times 4n}$



[Martinsson, 2013]
[Gillman & Martinsson, 2014]

Solving the strip problem
A fast direct solver for the strip

We use the hierarchical Poincaré–Steklov scheme to build a **fast direct solver** in $\mathcal{S}$.

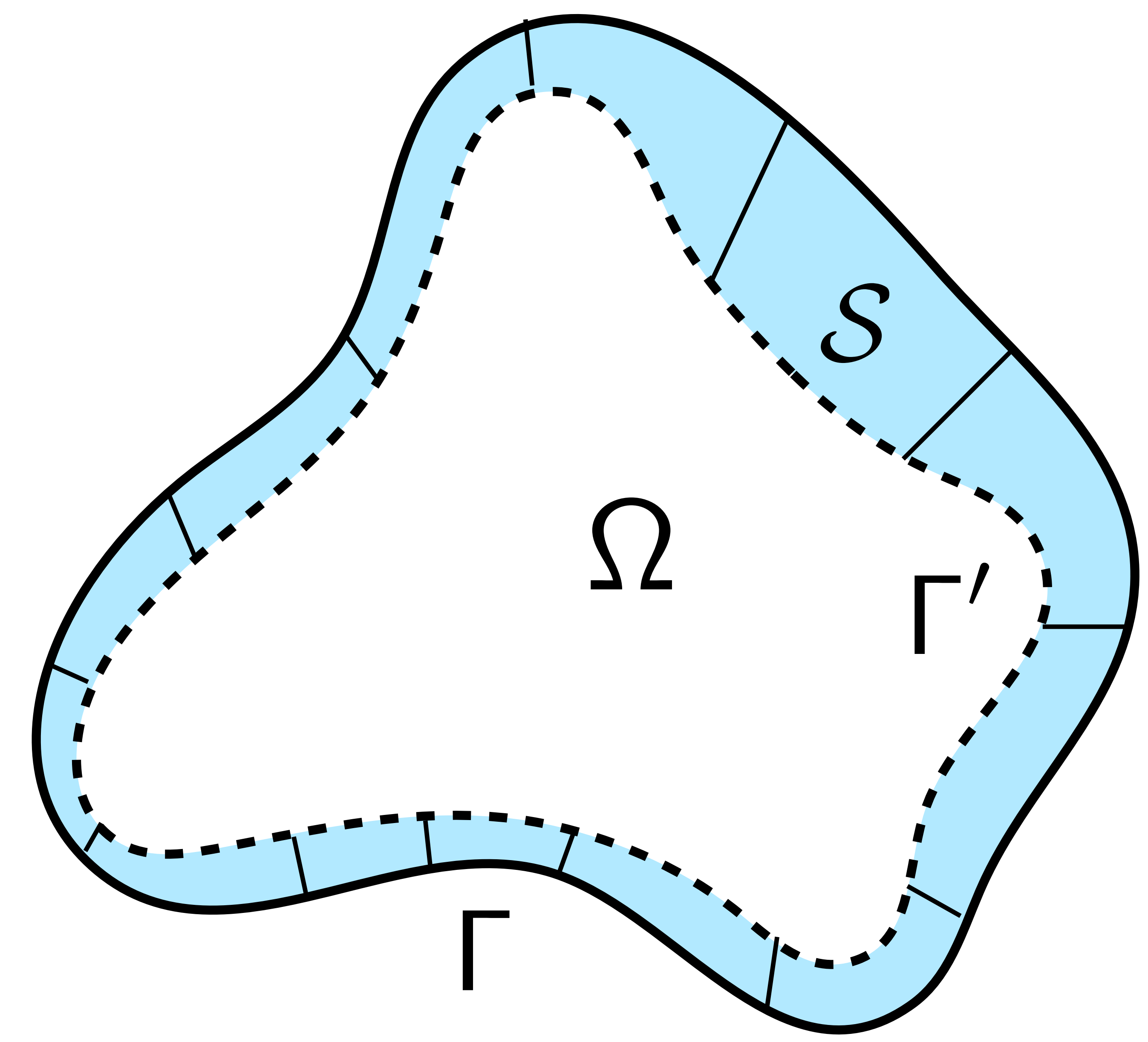
Given an inhomogeneity f:

① On each element, compute:

- Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
- Dirchlet-to-Neumann map: $DtN \in \mathbb{R}^{4n \times 4n}$

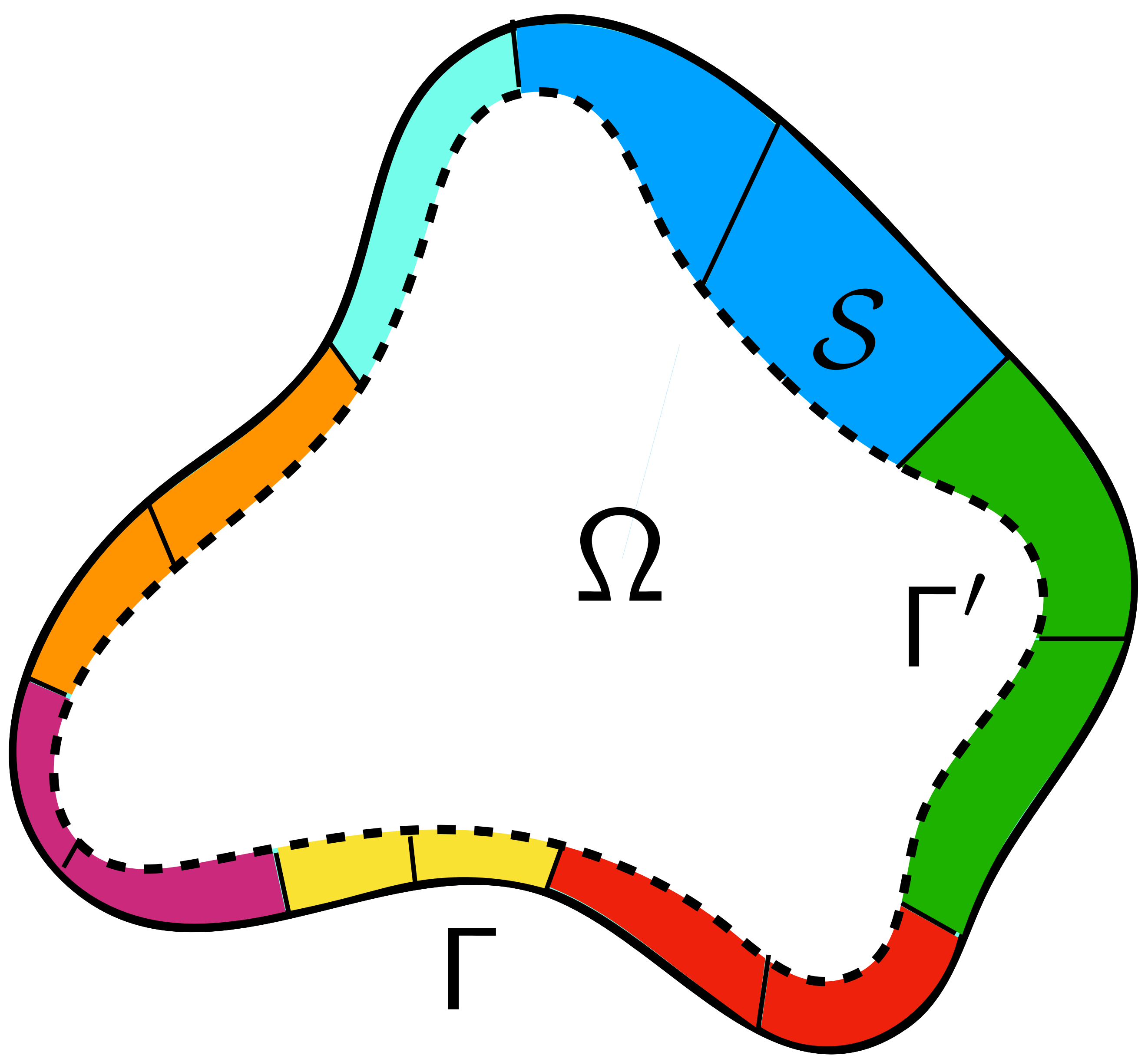
② Merge adjacent elements pairwise

- Compute S and DtN on parent via Schur complement



[Martinsson, 2013]
[Gillman & Martinsson, 2014]

We use the hierarchical Poincaré–Steklov scheme to build a **fast direct solver** in $\mathcal{S}$.



Given an inhomogeneity f:

1. On each element, compute:
   - Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Dirchlet-to-Neumann map: $DtN \in \mathbb{R}^{4n \times 4n}$

2. Merge adjacent elements pairwise
   - Compute S and DtN on parent via Schur complement

[Martinsson, 2013]
[Gillman & Martinsson, 2014]

Solving the strip problem
A fast direct solver for the strip

We use the hierarchical Poincaré–Steklov scheme to build a **fast direct solver** in $\mathcal{S}$.

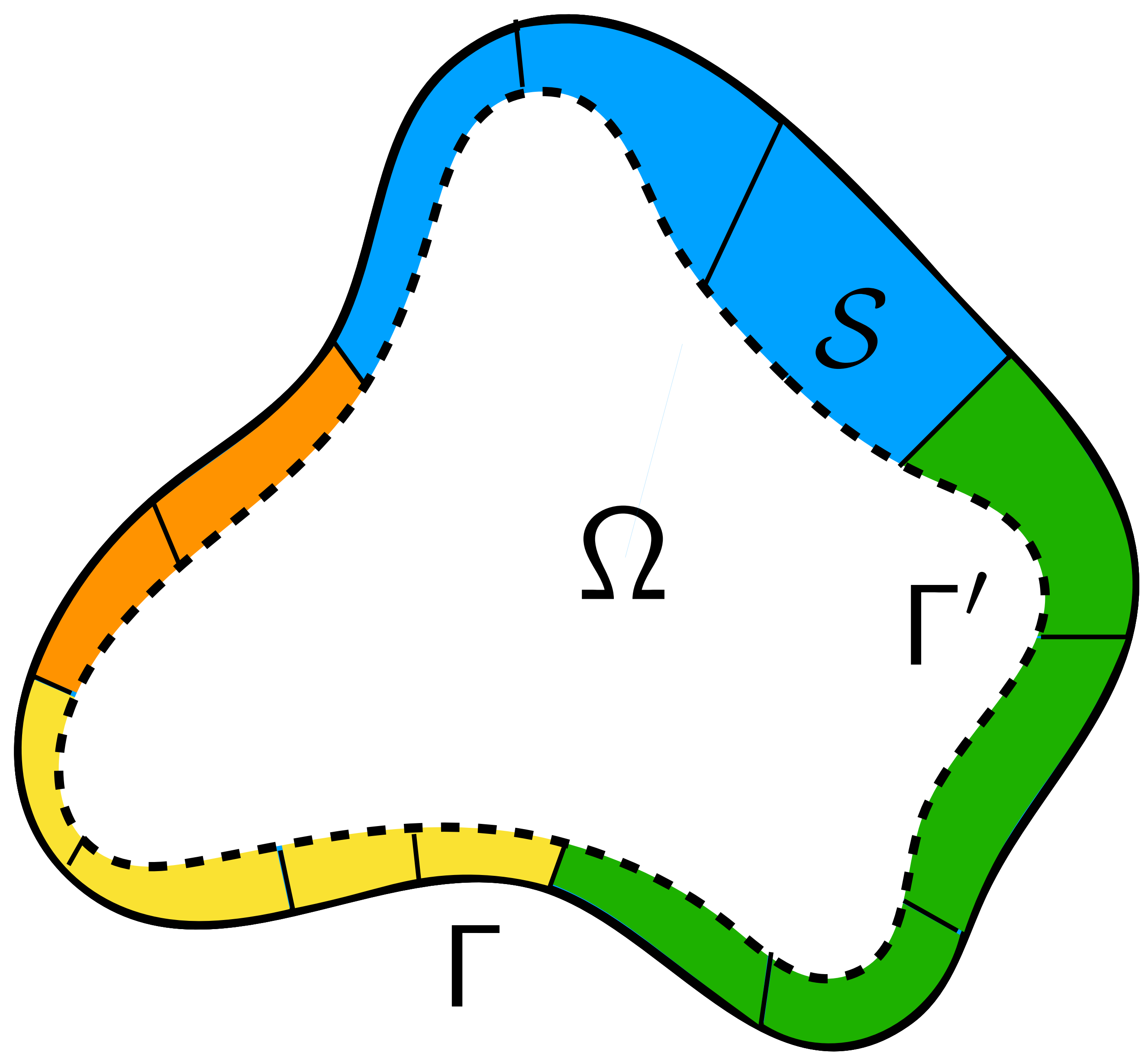
Given an inhomogeneity f:

① On each element, compute:

- Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
- Dirchlet-to-Neumann map: $DtN \in \mathbb{R}^{4n \times 4n}$

② Merge adjacent elements pairwise

- Compute S and DtN on parent via Schur complement



[Martinsson, 2013]
[Gillman & Martinsson, 2014]

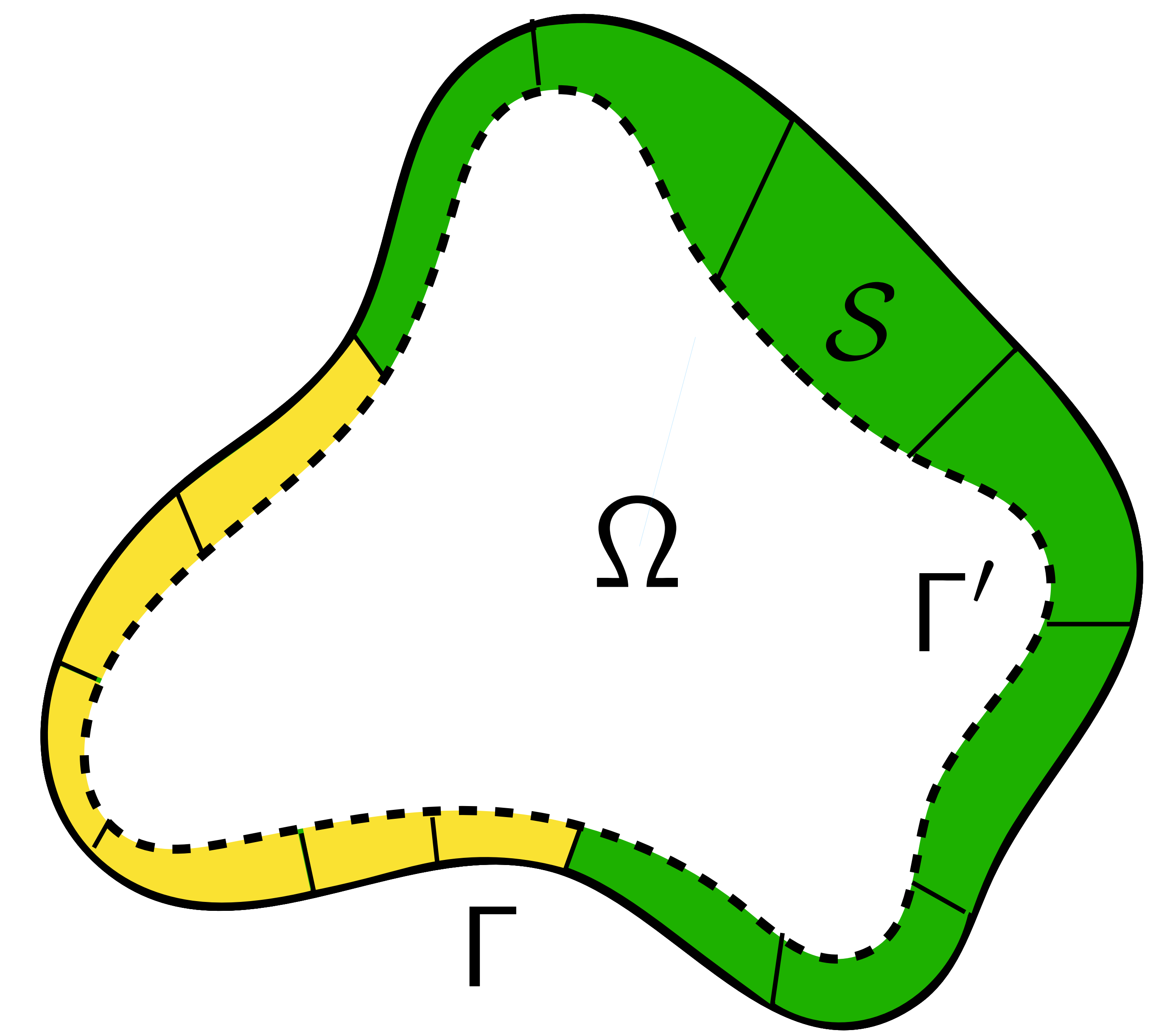
Solving the strip problem
A fast direct solver for the strip

We use the hierarchical Poincaré–Steklov scheme to build a **fast direct solver** in $\mathcal{S}$.

Given an inhomogeneity f:

1. On each element, compute:
   - Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Dirchlet-to-Neumann map: $DtN \in \mathbb{R}^{4n \times 4n}$

2. Merge adjacent elements pairwise
   - Compute S and DtN on parent via Schur complement



[Martinsson, 2013]
[Gillman & Martinsson, 2014]

Solving the strip problem

A fast direct solver for the strip

We use the hierarchical Poincaré–Steklov scheme to build a **fast direct solver** in $\mathcal{S}$.

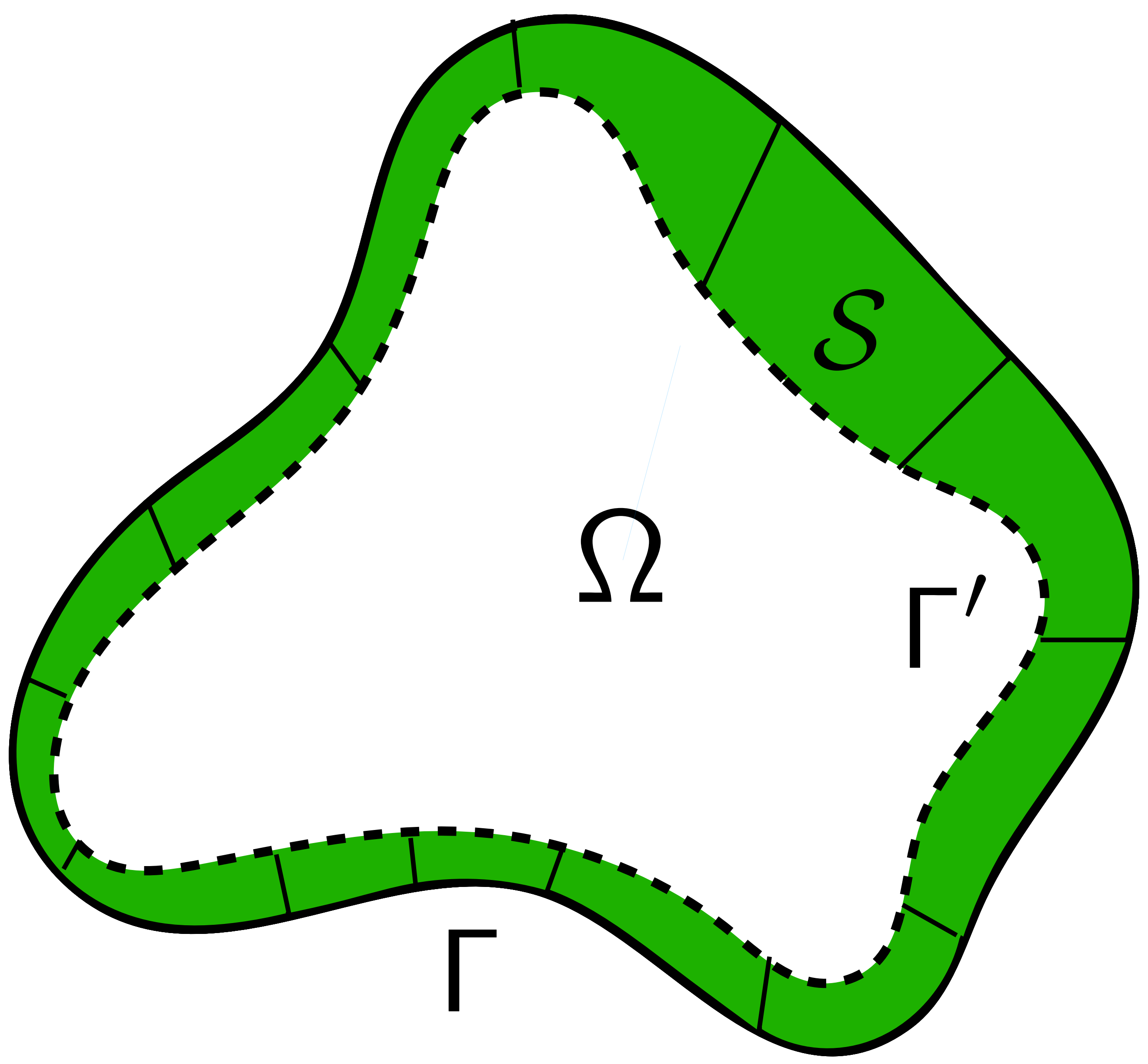
Given an inhomogeneity f:

① On each element, compute:

- Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
- Dirchlet-to-Neumann map: $DtN \in \mathbb{R}^{4n \times 4n}$

② Merge adjacent elements pairwise

- Compute S and DtN on parent via Schur complement



[Martinsson, 2013]
[Gillman & Martinsson, 2014]

We use the hierarchical Poincaré–Steklov scheme to build a **fast direct solver** in $\mathcal{S}$.

Given an inhomogeneity f:

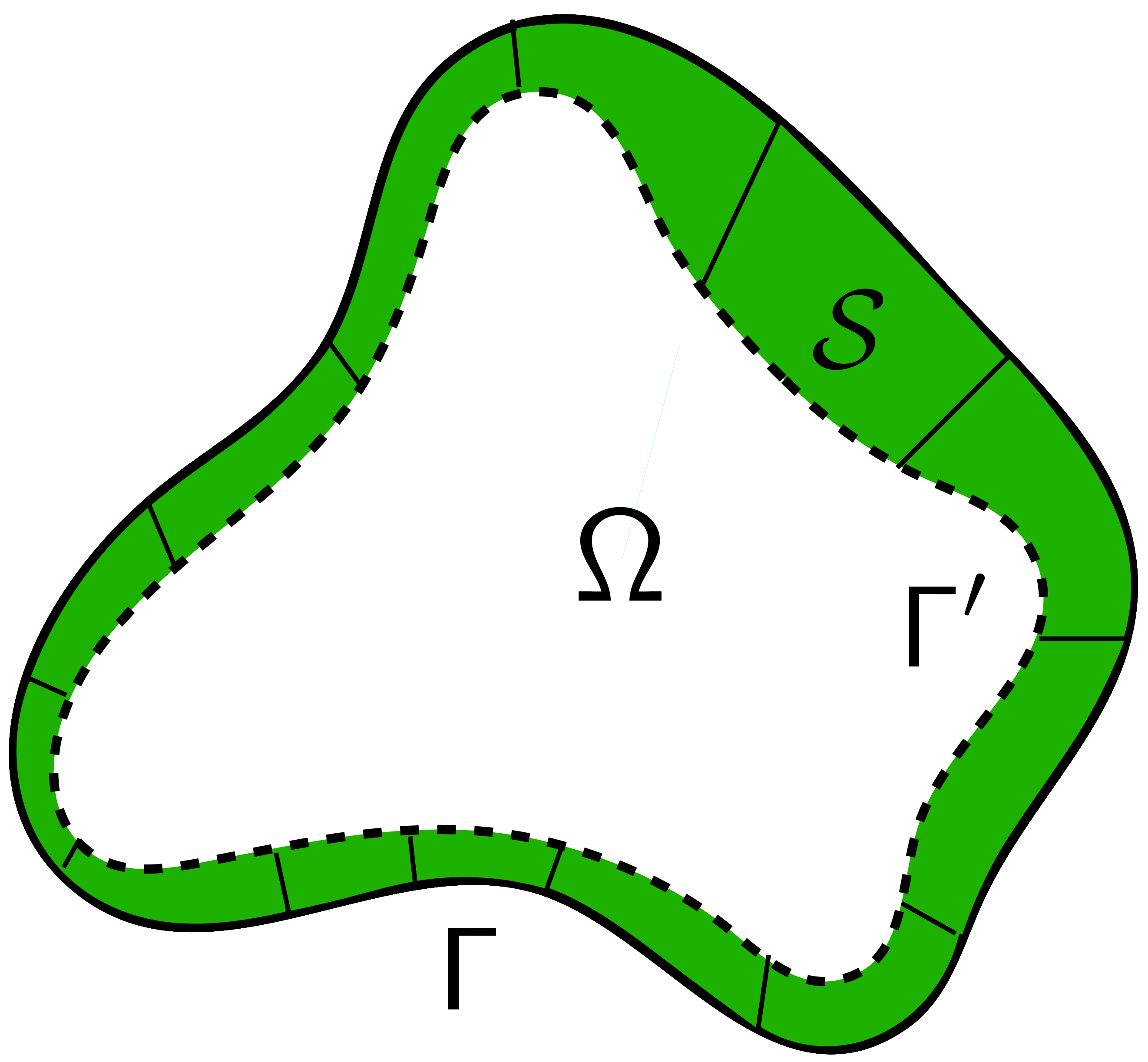
① On each element, compute:

- Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
- Dirchlet-to-Neumann map: $DtN \in \mathbb{R}^{4n \times 4n}$

② Merge adjacent elements pairwise

- Compute S and DtN on parent via Schur complement

③ Recursively apply S, starting from known boundary conditions at the top level.



[Martinsson, 2013]
[Gillman & Martinsson, 2014]

Solving the strip problem

A fast direct solver for the strip

We use the hierarchical Poincaré–Steklov scheme to build a **fast direct solver** in $\mathcal{S}$.

Given an inhomogeneity f:

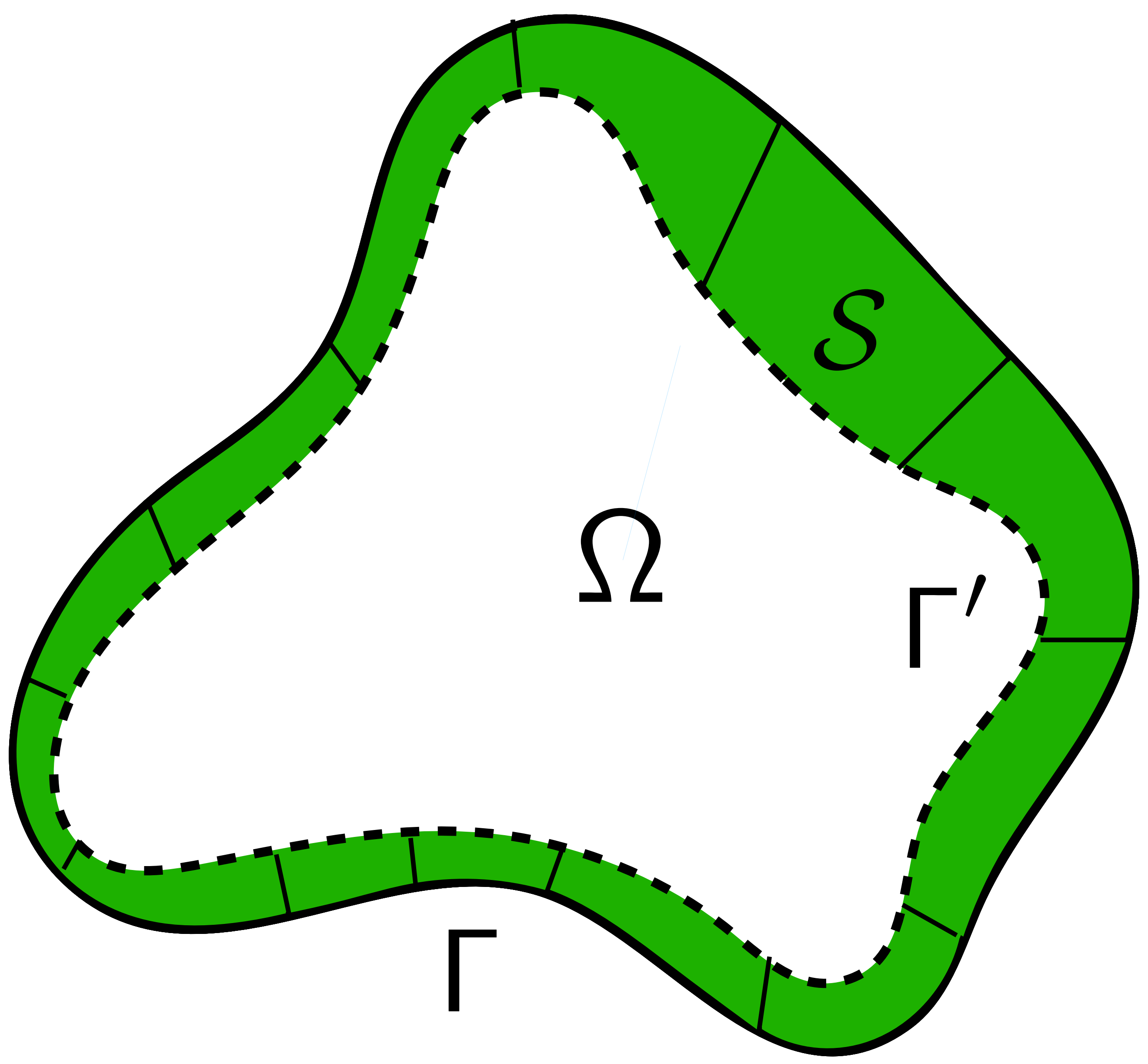
① On each element, compute:

- Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
- Dirchlet-to-Neumann map: $DtN \in \mathbb{R}^{4n \times 4n}$

② Merge adjacent elements pairwise

- Compute S and DtN on parent via Schur complement

③ Recursively apply S, starting from known boundary conditions at the top level.



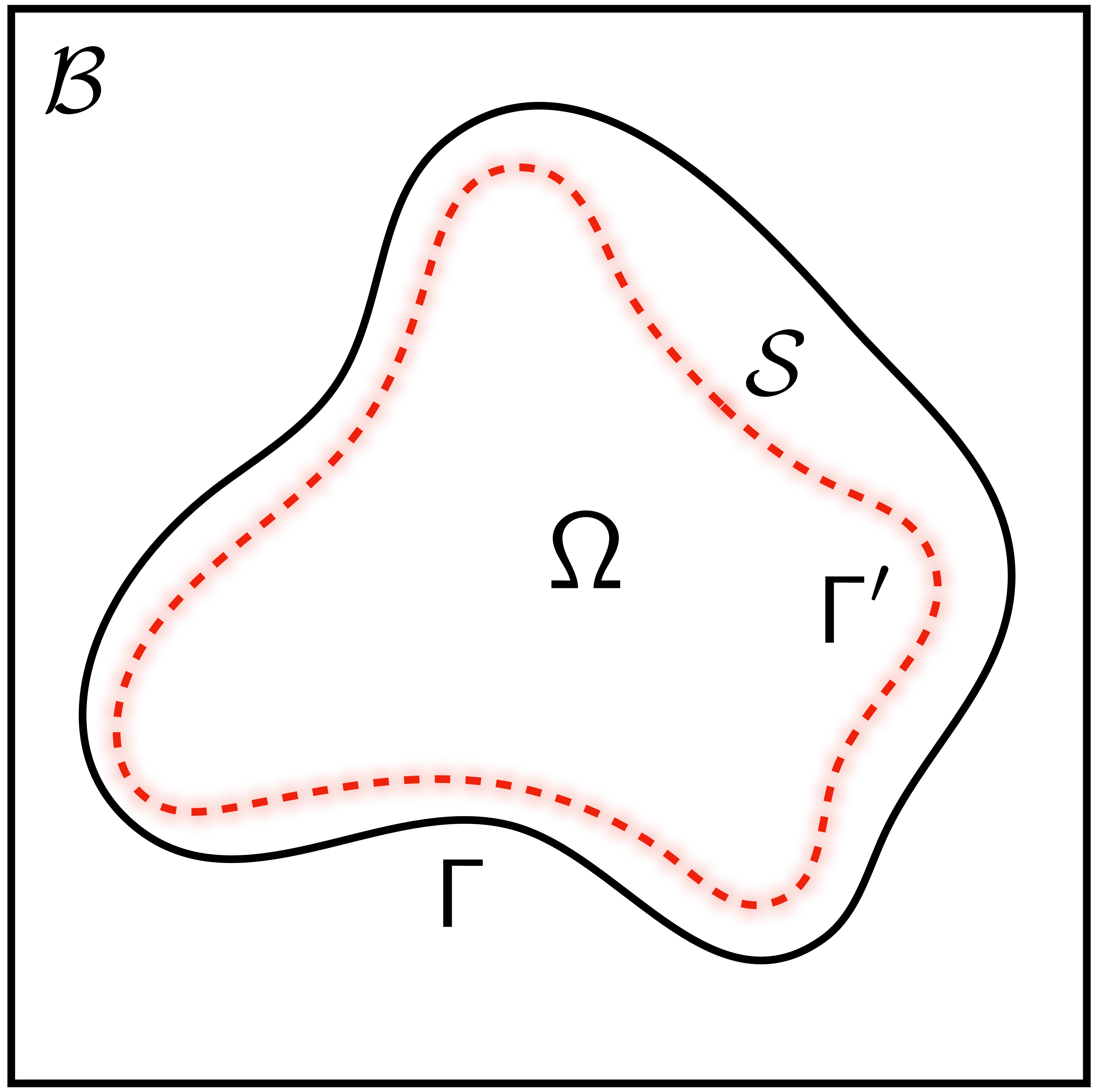
Cost: $\underset{①}{\mathcal{O}(p^6 n_{\text{panel}})} + \underset{②}{\mathcal{O}(p^3 n_{\text{panel}})} + \underset{③}{\mathcal{O}(p^2 n_{\text{panel}})} = \mathcal{O}(n_{\text{panel}})$

We typically use p = 16 on each panel and upsample the SEM grid to 2p = 32.

[Martinsson, 2013]
[Gillman & Martinsson, 2014]

Patching the solutions
Layer potentials

The solutions v_{bulk} and v_{strip} may differ in value and normal derivative along Γ'.



Patching the solutions

Layer potentials

The solutions v_{bulk} and v_{strip} may differ in value and normal derivative along Γ'.

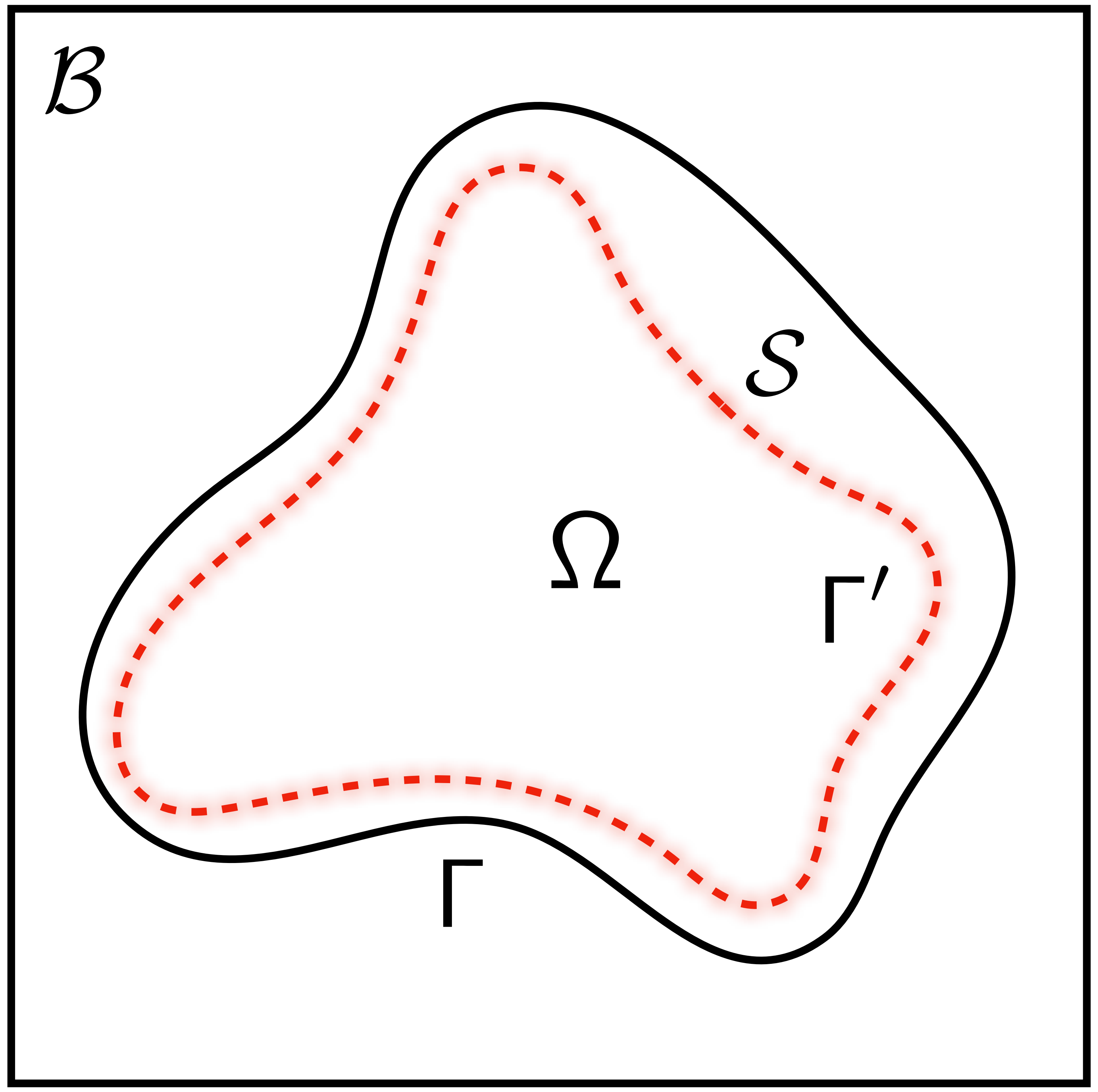
The single- and double-layer potentials

$$S[\sigma](\boldsymbol{x}) = \int_{\Gamma'} \Phi(\boldsymbol{x},\boldsymbol{y})\,\sigma(\boldsymbol{y})\,d\boldsymbol{y} \qquad D[\tau](\boldsymbol{x}) = \int_{\Gamma'} \frac{\partial \Phi(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{n_y}}\,\tau(\boldsymbol{y})\,d\boldsymbol{y}$$

are harmonic and satisfy the jump relations

$$S[\sigma]\big|_{\Gamma'}^{+} - S[\sigma]\big|_{\Gamma'}^{-} = 0 \qquad D[\tau]\big|_{\Gamma'}^{+} - D[\tau]\big|_{\Gamma'}^{-} = \tau$$

$$\partial_{\boldsymbol{n}} S[\sigma]\big|_{\Gamma'}^{+} - \partial_{\boldsymbol{n}} S[\sigma]\big|_{\Gamma'}^{-} = -\sigma \qquad \partial_{\boldsymbol{n}} D[\tau]\big|_{\Gamma'}^{+} - \partial_{\boldsymbol{n}} D[\tau]\big|_{\Gamma'}^{-} = 0$$



Patching the solutions
Layer potentials

The solutions v_{bulk} and v_{strip} may differ in value and normal derivative along Γ'.

The single- and double-layer potentials

$$S[\sigma](\boldsymbol{x}) = \int_{\Gamma'} \Phi(\boldsymbol{x},\boldsymbol{y})\,\sigma(\boldsymbol{y})\,d\boldsymbol{y} \qquad D[\tau](\boldsymbol{x}) = \int_{\Gamma'} \frac{\partial\Phi(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{n}_{\boldsymbol{y}}}\,\tau(\boldsymbol{y})\,d\boldsymbol{y}$$

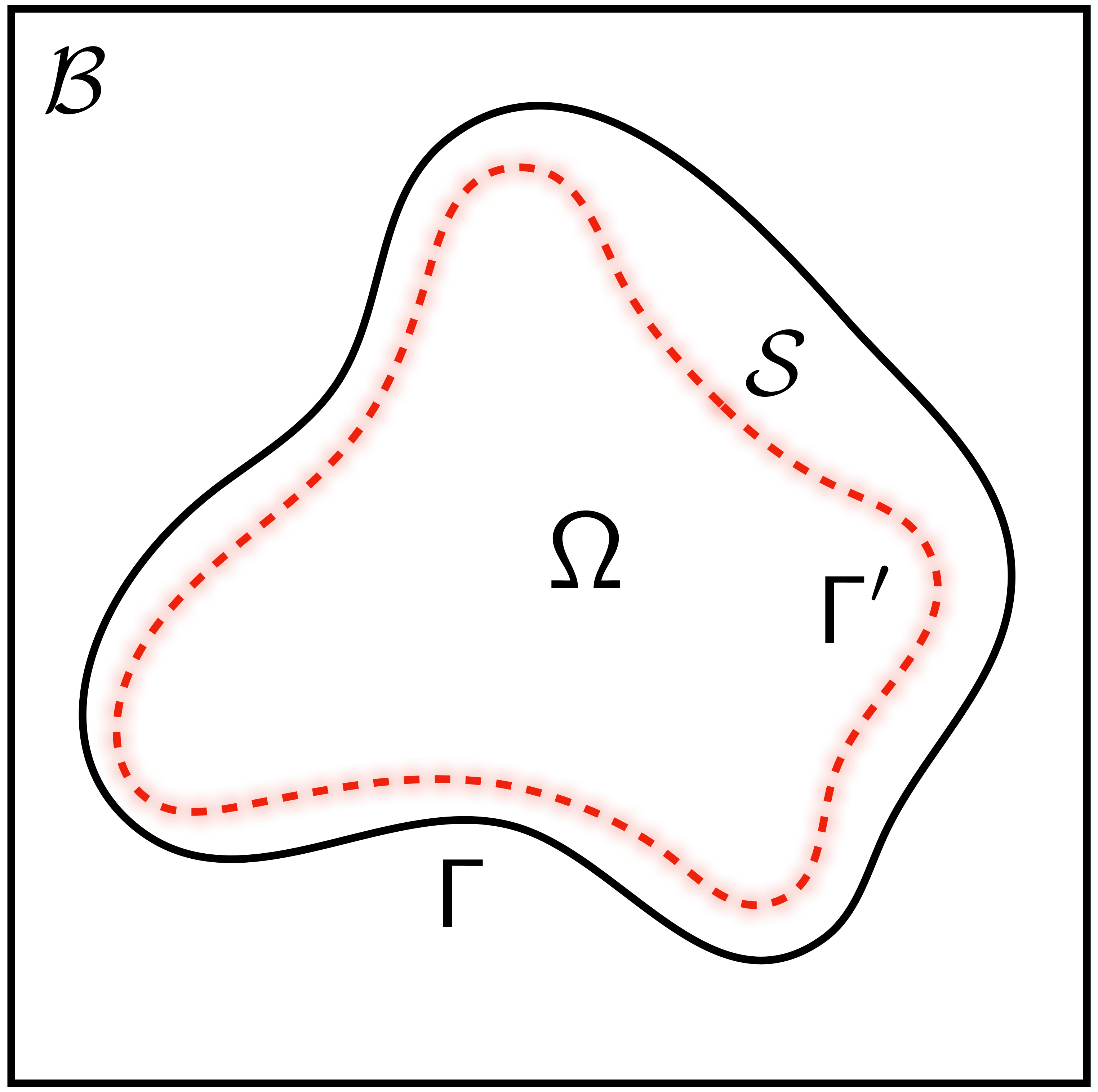
are harmonic and satisfy the jump relations

$$S[\sigma]\big|_{\Gamma'}^{+} - S[\sigma]\big|_{\Gamma'}^{-} = 0 \qquad D[\tau]\big|_{\Gamma'}^{+} - D[\tau]\big|_{\Gamma'}^{-} = \tau$$

$$\partial_{\boldsymbol{n}} S[\sigma]\big|_{\Gamma'}^{+} - \partial_{\boldsymbol{n}} S[\sigma]\big|_{\Gamma'}^{-} = -\sigma \qquad \partial_{\boldsymbol{n}} D[\tau]\big|_{\Gamma'}^{+} - \partial_{\boldsymbol{n}} D[\tau]\big|_{\Gamma'}^{-} = 0$$

So define $v_{\text{glue}} = S[\sigma] - D[\tau]$ with

$$\sigma = \partial_{\boldsymbol{n}} v_{\text{bulk}}|_{\Gamma'} - \partial_{\boldsymbol{n}} v_{\text{strip}}|_{\Gamma'} \qquad \tau = v_{\text{bulk}}|_{\Gamma'} - v_{\text{strip}}|_{\Gamma'}$$



Patching the solutions
Layer potentials

The solutions v_{bulk} and v_{strip} may differ in value and normal derivative along Γ'.

The single- and double-layer potentials

$$S[\sigma](\boldsymbol{x}) = \int_{\Gamma'} \Phi(\boldsymbol{x},\boldsymbol{y})\,\sigma(\boldsymbol{y})\,d\boldsymbol{y} \qquad D[\tau](\boldsymbol{x}) = \int_{\Gamma'} \frac{\partial\Phi(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{n}_{\boldsymbol{y}}}\,\tau(\boldsymbol{y})\,d\boldsymbol{y}$$

are harmonic and satisfy the jump relations

$$S[\sigma]\big|_{\Gamma'}^{+} - S[\sigma]\big|_{\Gamma'}^{-} = 0 \qquad D[\tau]\big|_{\Gamma'}^{+} - D[\tau]\big|_{\Gamma'}^{-} = \tau$$

$$\partial_{\boldsymbol{n}} S[\sigma]\big|_{\Gamma'}^{+} - \partial_{\boldsymbol{n}} S[\sigma]\big|_{\Gamma'}^{-} = -\sigma \qquad \partial_{\boldsymbol{n}} D[\tau]\big|_{\Gamma'}^{+} - \partial_{\boldsymbol{n}} D[\tau]\big|_{\Gamma'}^{-} = 0$$

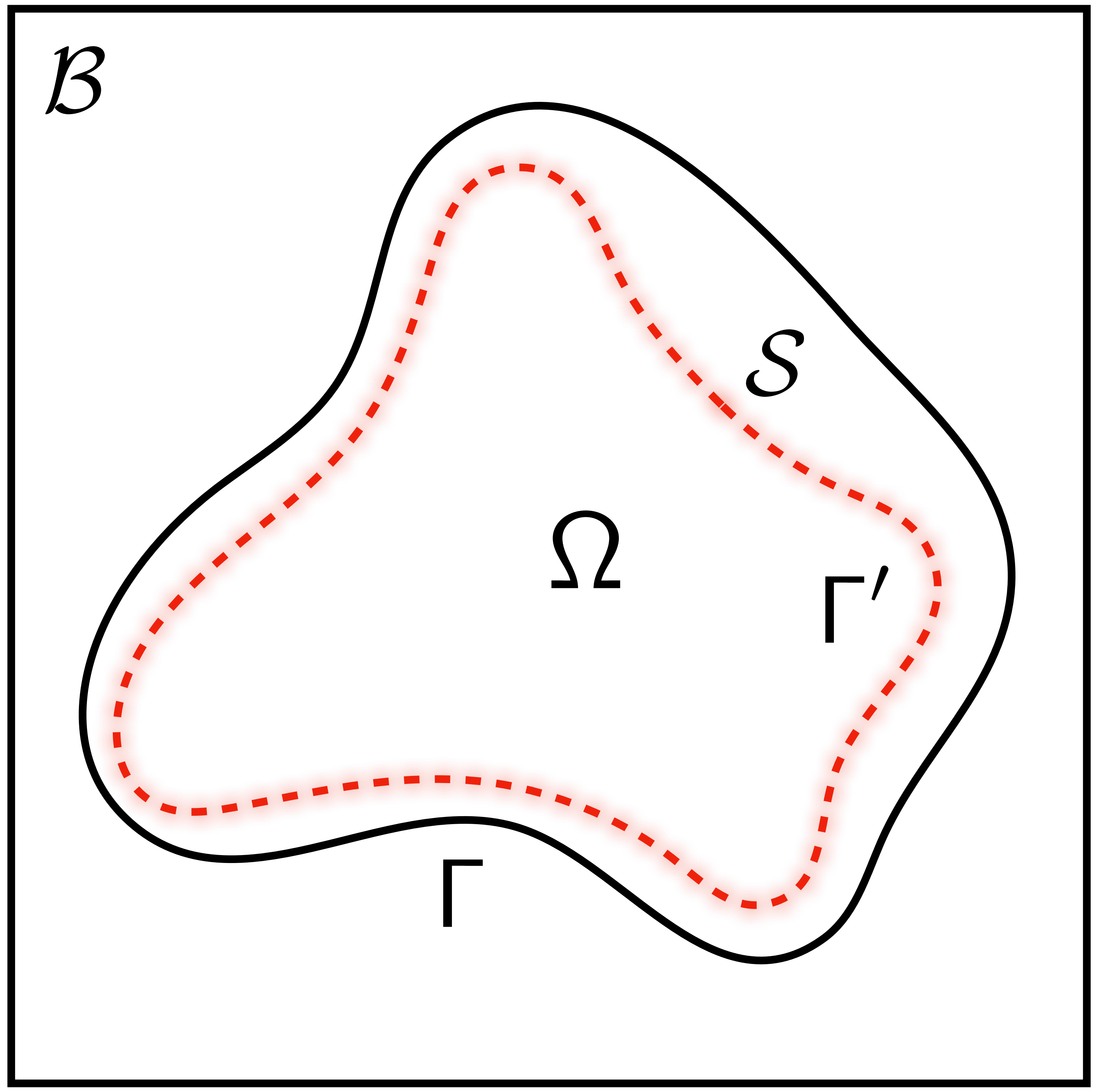
So define $v_{\text{glue}} = S[\sigma] - D[\tau]$ with

$$\sigma = \partial_{\boldsymbol{n}} v_{\text{bulk}}|_{\Gamma'} - \partial_{\boldsymbol{n}} v_{\text{strip}}|_{\Gamma'} \qquad \tau = v_{\text{bulk}}|_{\Gamma'} - v_{\text{strip}}|_{\Gamma'}$$

Then

$$v = \begin{cases} v_{\text{bulk}} + v_{\text{glue}} & \text{in } \Omega \setminus \mathcal{S} \\ v_{\text{strip}} + v_{\text{glue}} & \text{in } \mathcal{S} \end{cases}$$

satisfies $\Delta v = f$ in Ω and is smooth across Γ'.



Correcting for boundary data
Homogeneous solution

Finally, the boundary conditions may not be satisfied.

A homogeneous correction may computed using standard BIE techniques:

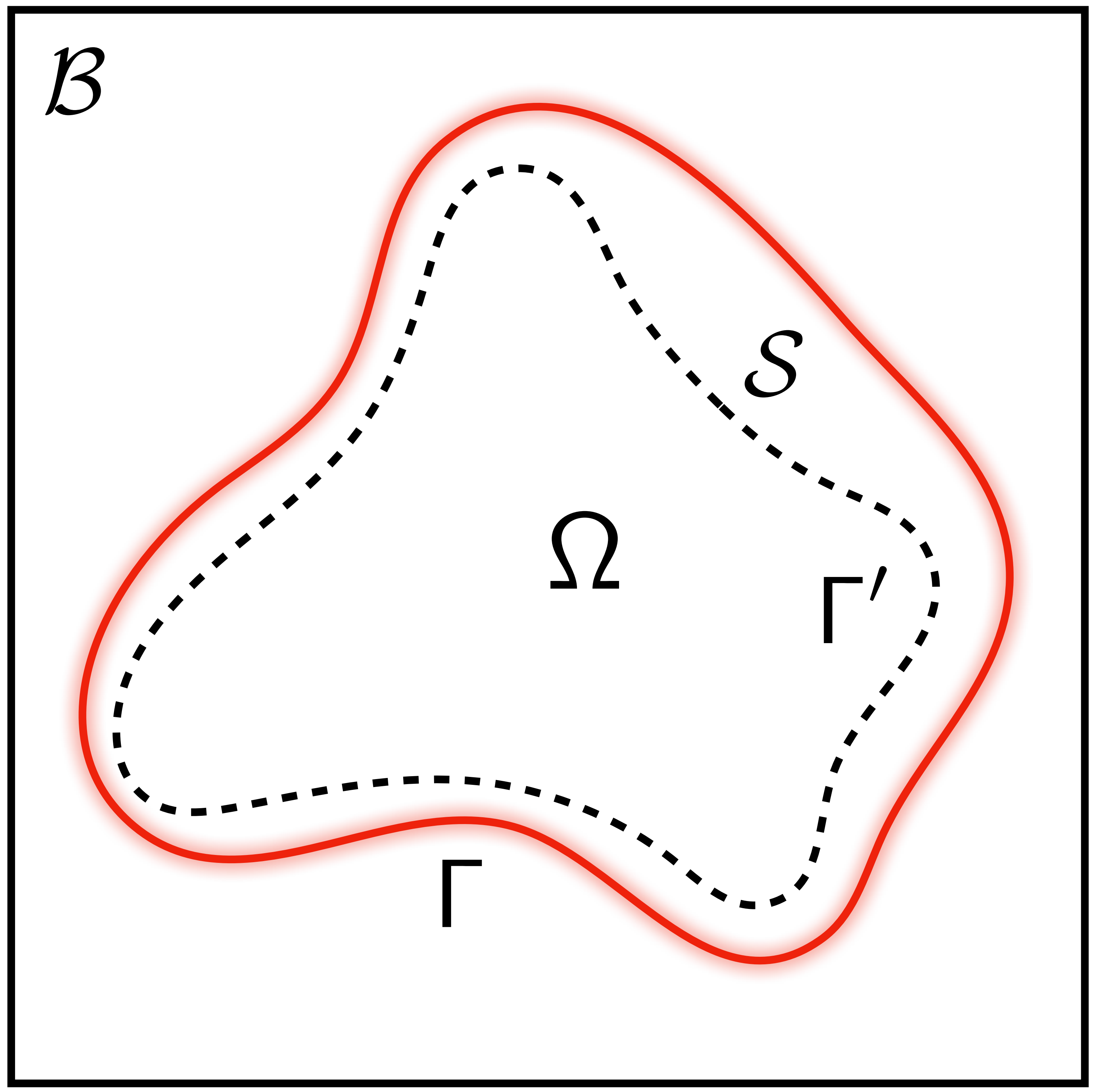
$$\begin{aligned} \Delta w &= 0 && \text{in } \Omega \\ w &= g - v_{\text{glue}}\big|_{\Gamma} && \text{on } \Gamma \end{aligned}$$

Then,

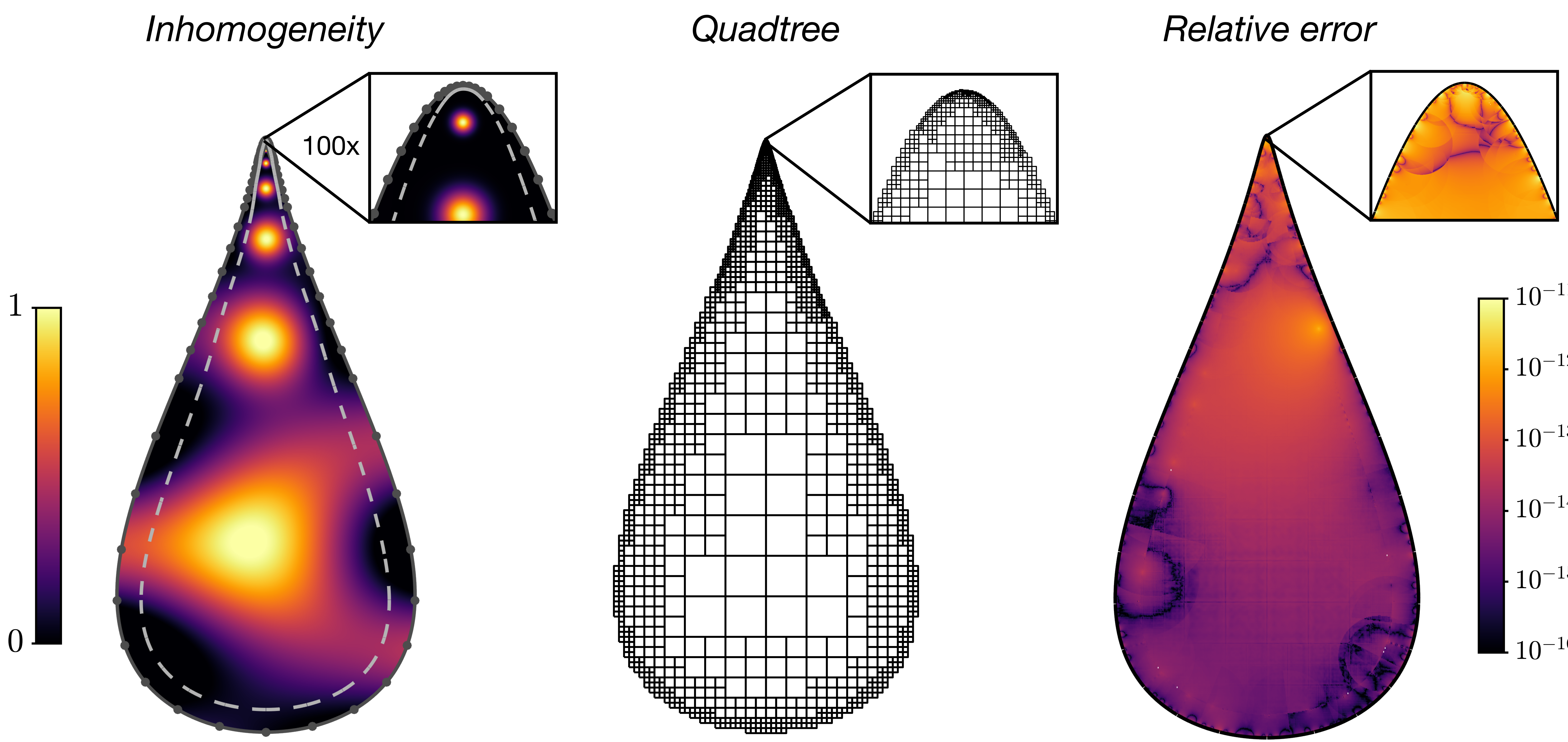
$$u = \begin{cases} v_{\text{bulk}} + v_{\text{glue}} + w & \text{in } \Omega \setminus \mathcal{S} \\ v_{\text{strip}} + v_{\text{glue}} + w & \text{in } \mathcal{S} \end{cases}$$

satisfies

$$\begin{aligned} \Delta u &= f && \text{in } \Omega \\ u &= g && \text{on } \Gamma \end{aligned}$$



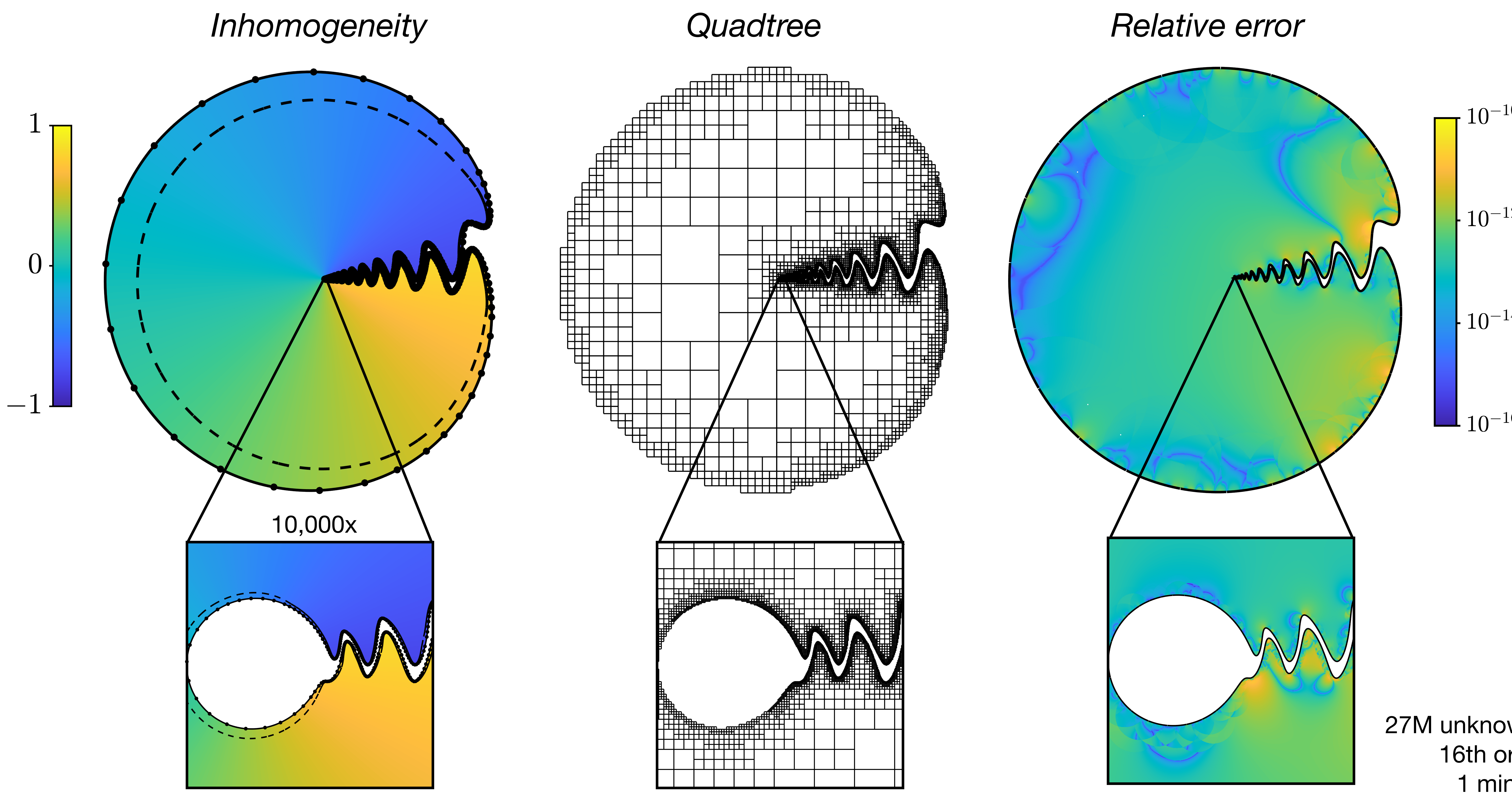
Example
A rounded cusp



750k unknowns
16th order
2 seconds

Example

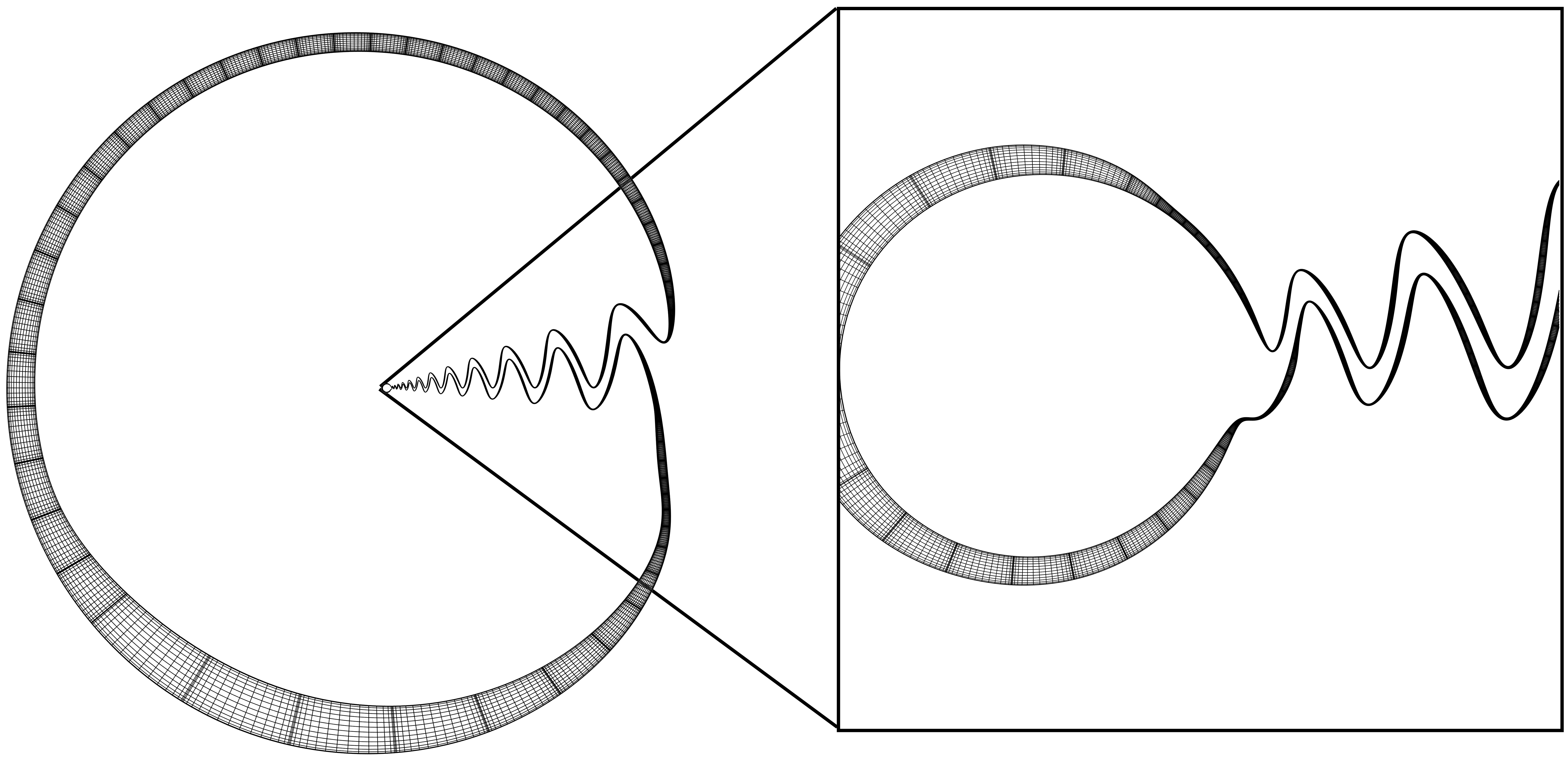
A multiscale inhomogeneity and close-to-touching geometry



27M unknowns
16th order
1 minute

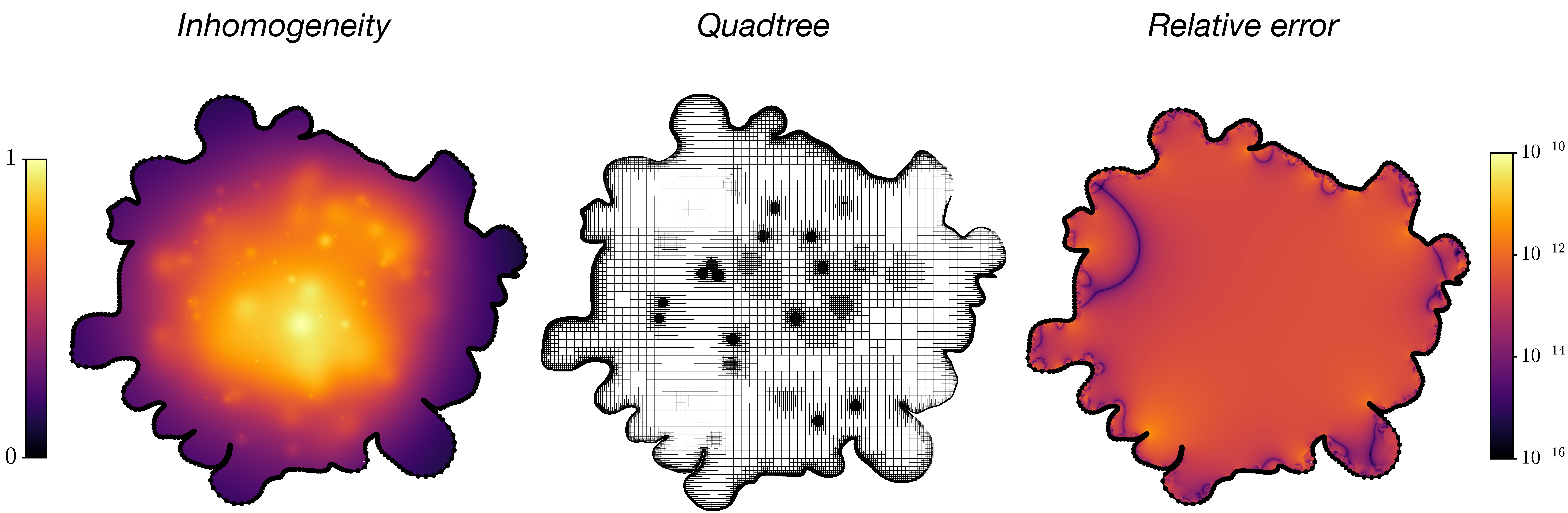
Example

A multiscale inhomogeneity and close-to-touching geometry



Example

Cell blebbing geometry extract from microscopy



12M unknowns
16th order
30 seconds

Conclusion

David Stein

FLATIRON INSTITUTE
Center for Computational Mathematics

Alex Barnett

Conclusion

- Function extension can be challenging in multiscale geometries. Our scheme avoids extension.
- Particular solutions can be computed adaptively and efficiently using truncated volume potentials.
- Surface-conforming truncation idea designed to extend to 3D.

David Stein

FLATIRON INSTITUTE
Center for Computational Mathematics

Alex Barnett