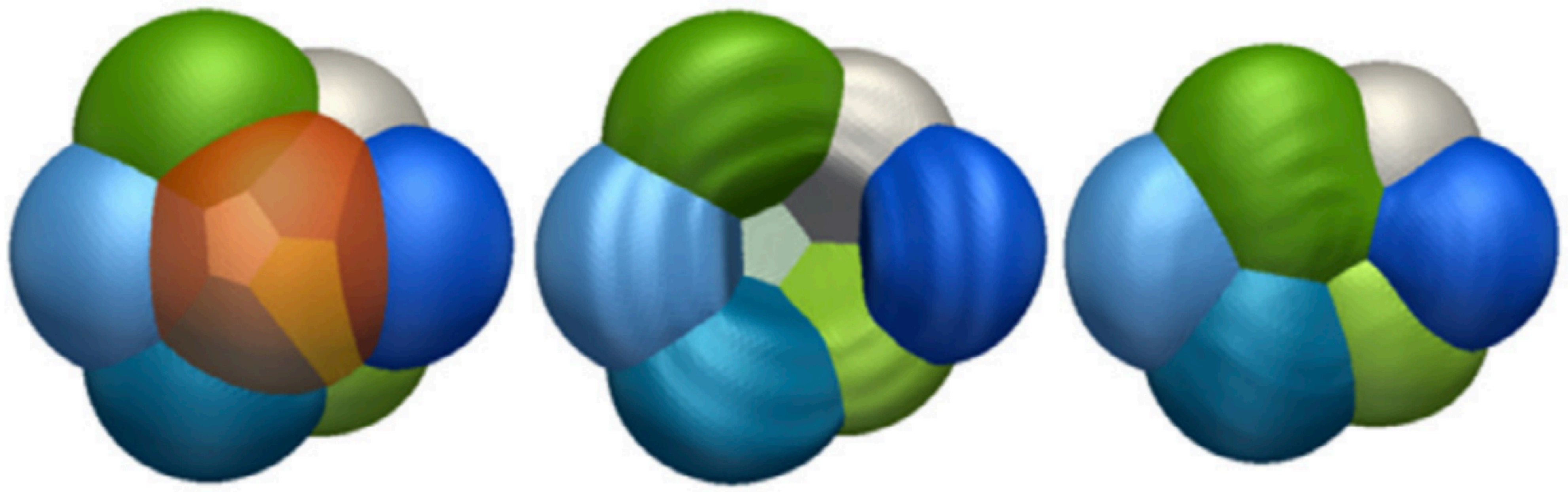
Fast algorithms for biological processes on surfaces

Dan Fortunato

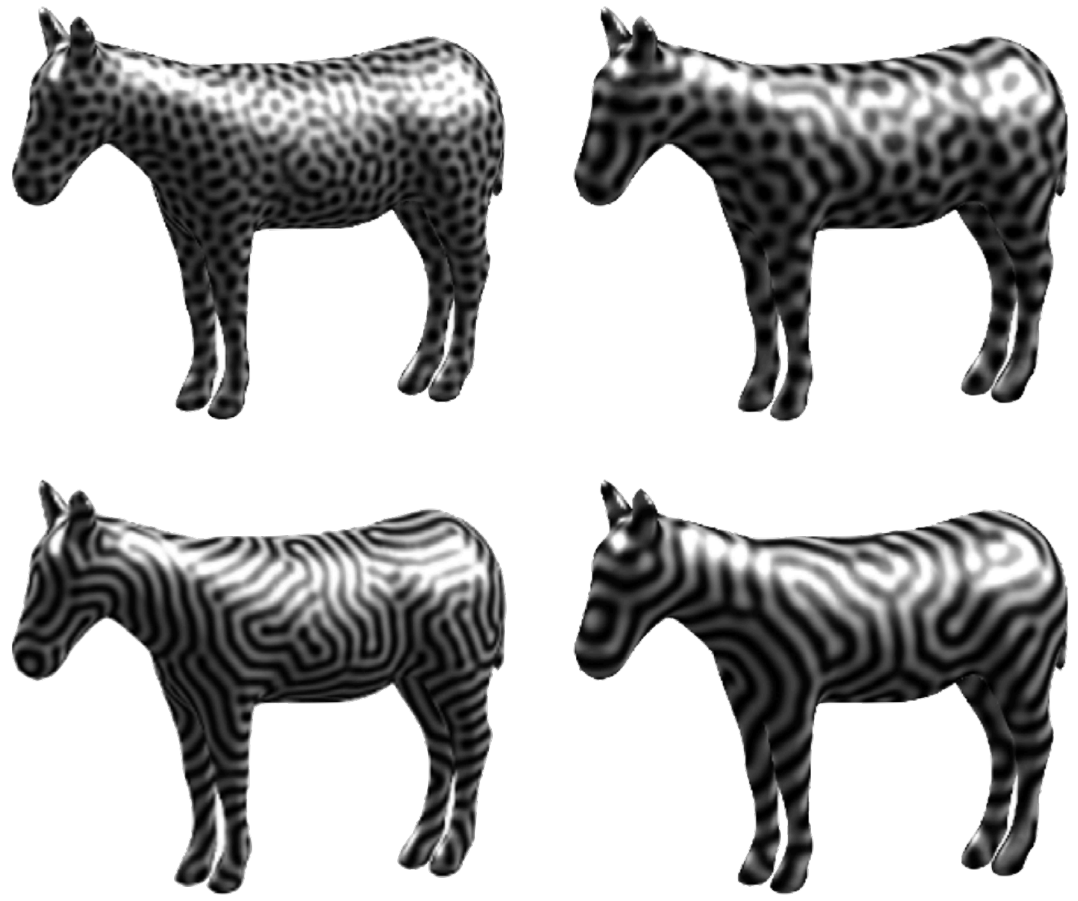
Flatiron Institute

Surface-bound phenomena arise in many applications.

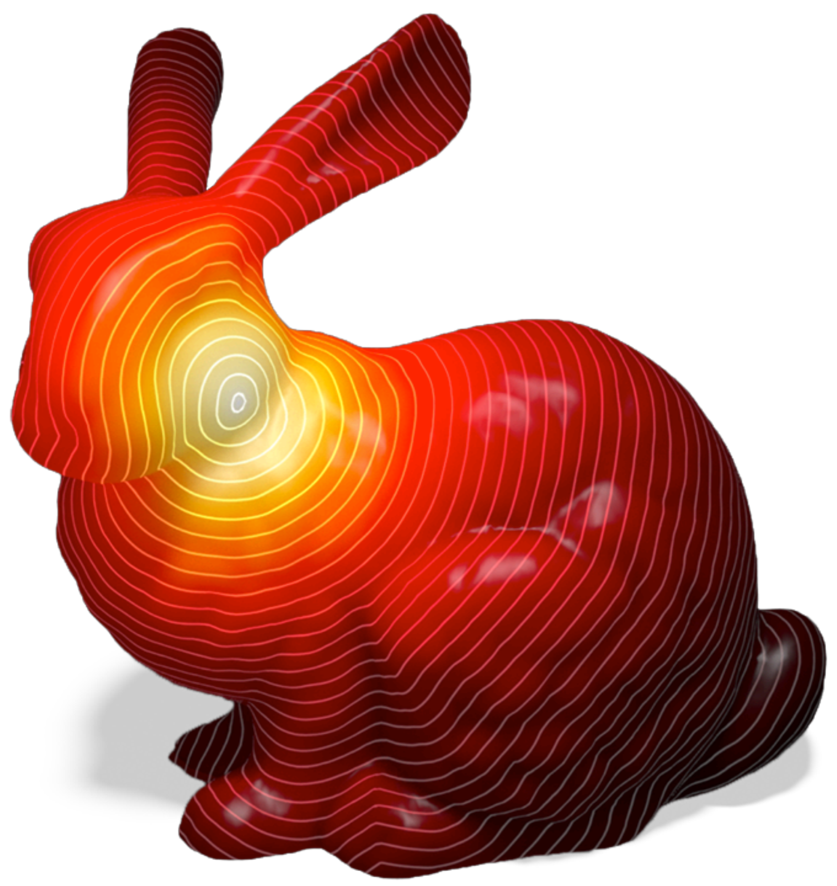
Thin-film hydrodynamics [Saye, 2016]



Pattern formation [Jeong, 2017]

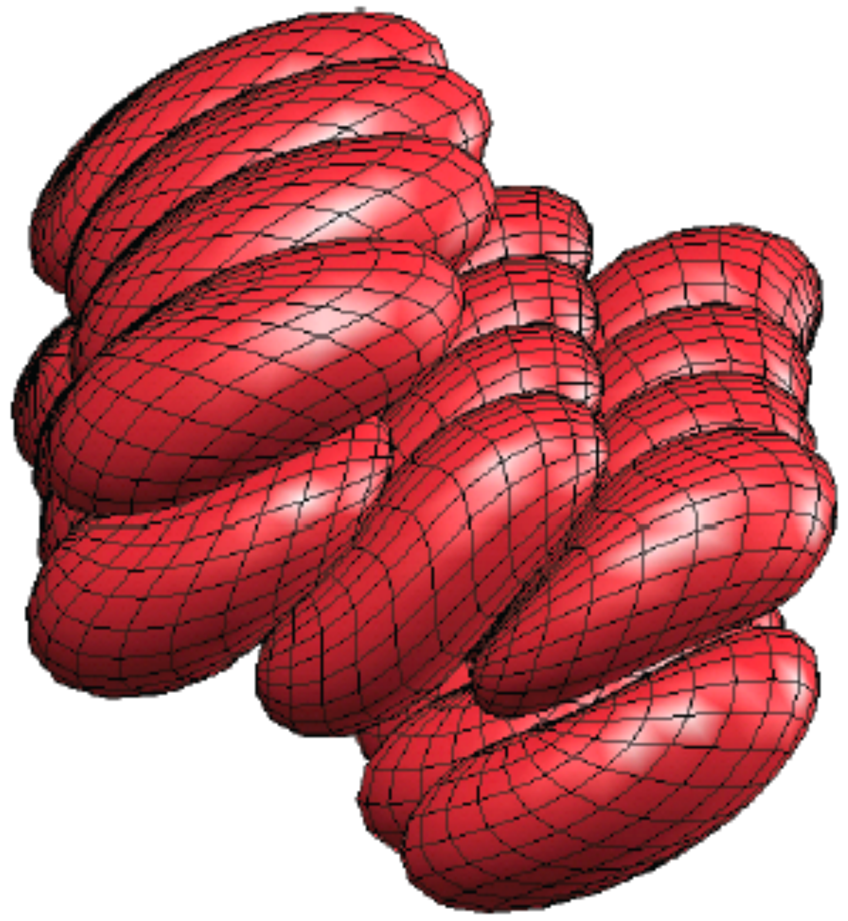


Geodesic distance [Crane et al., 2017]

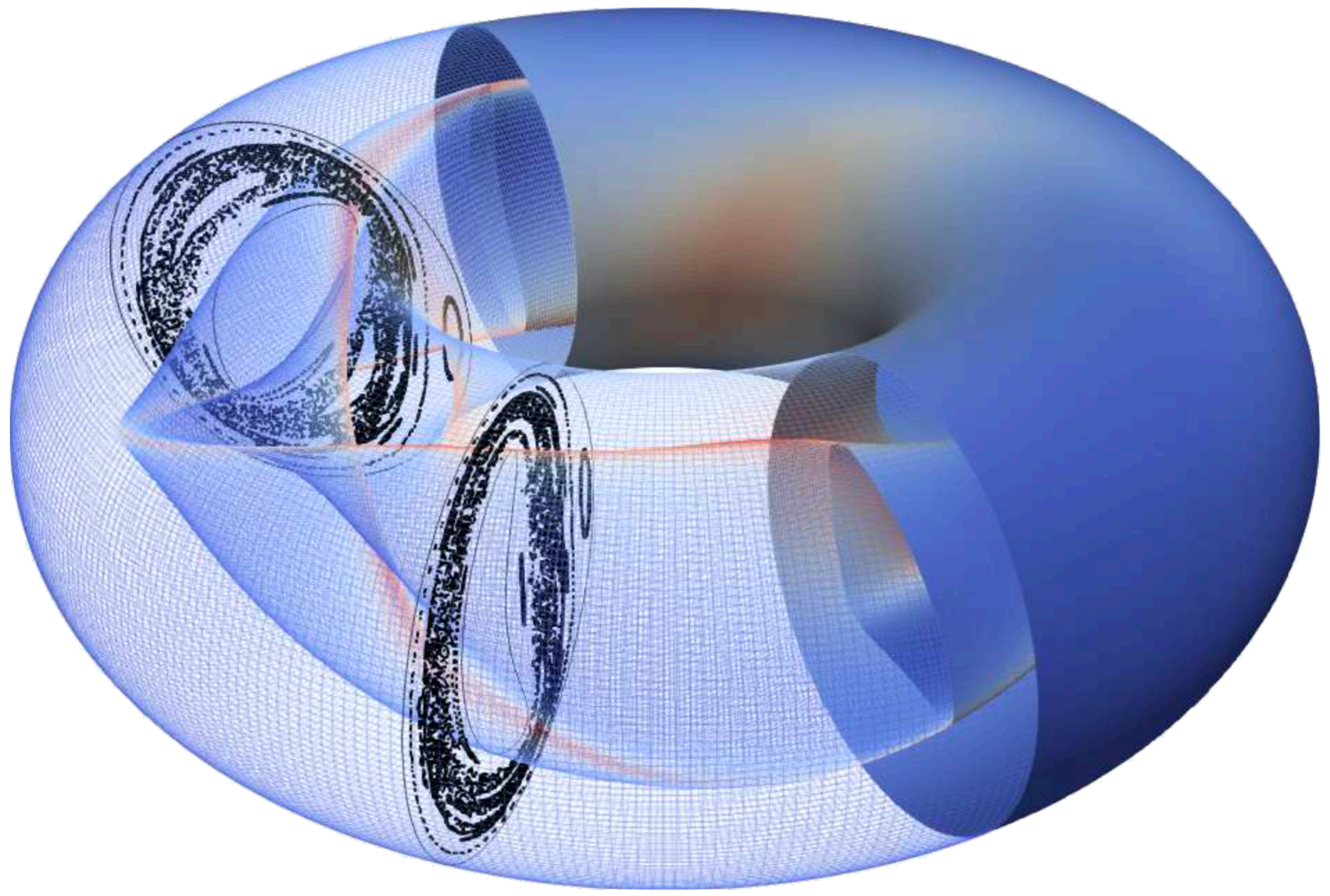


Embryonic furrow development [Image from HHMI]

Vesicle flows [Veerapaneni et al., 2011]



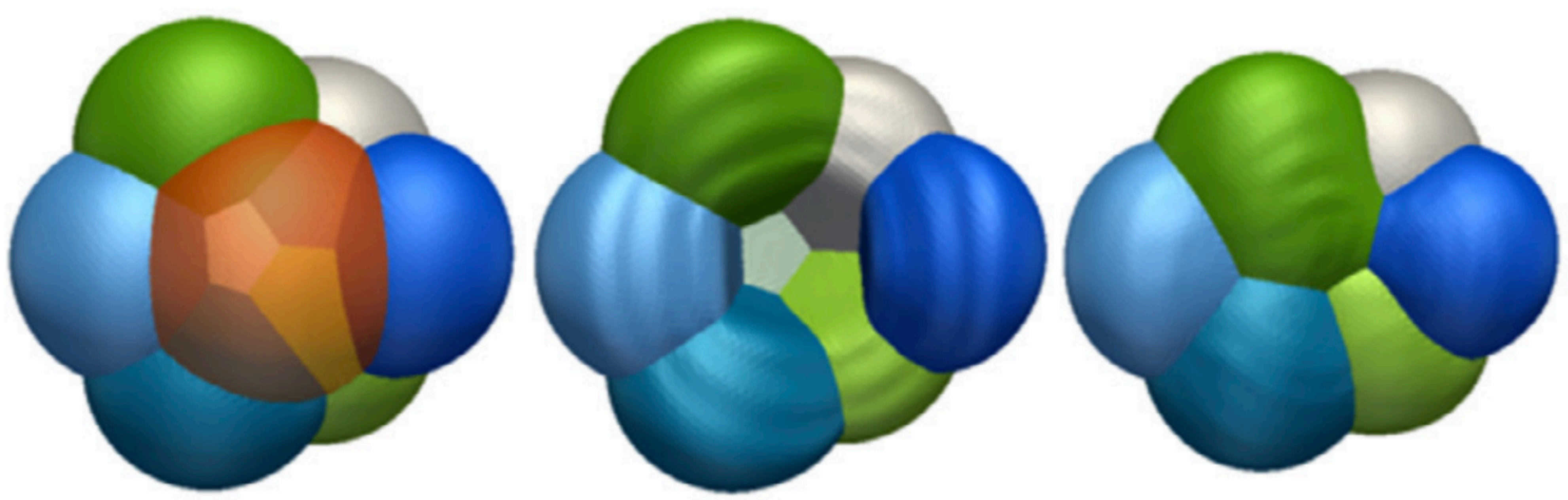
Stellarator design [Malhotra et al., 2019]



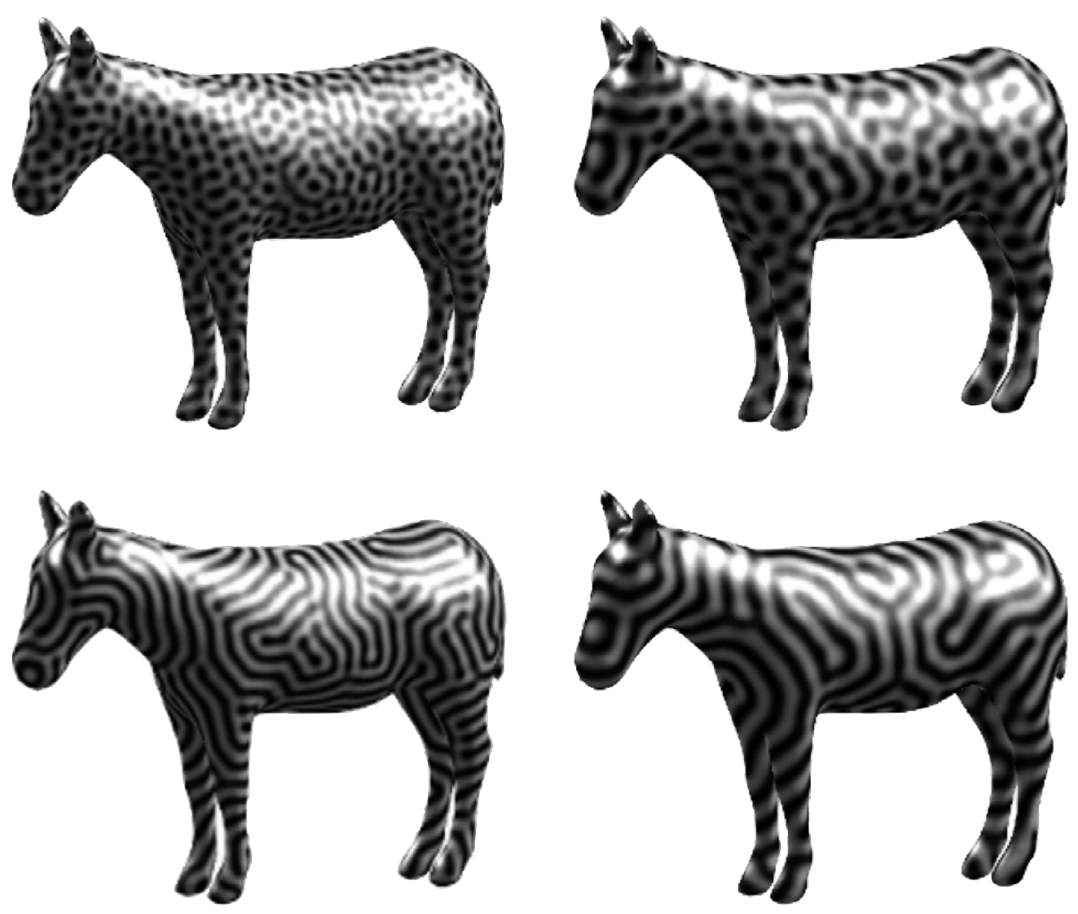
Introduction
Surface PDEs

Surface-bound phenomena arise in many applications.

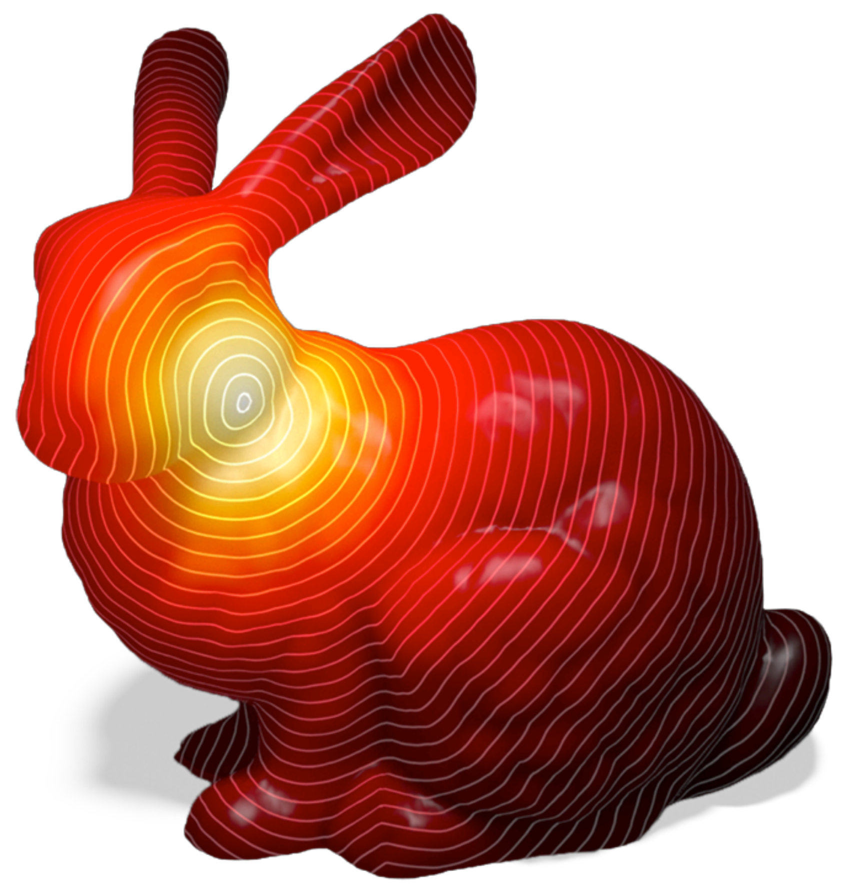
Thin-film hydrodynamics [Saye, 2016]



Pattern formation [Jeong, 2017]

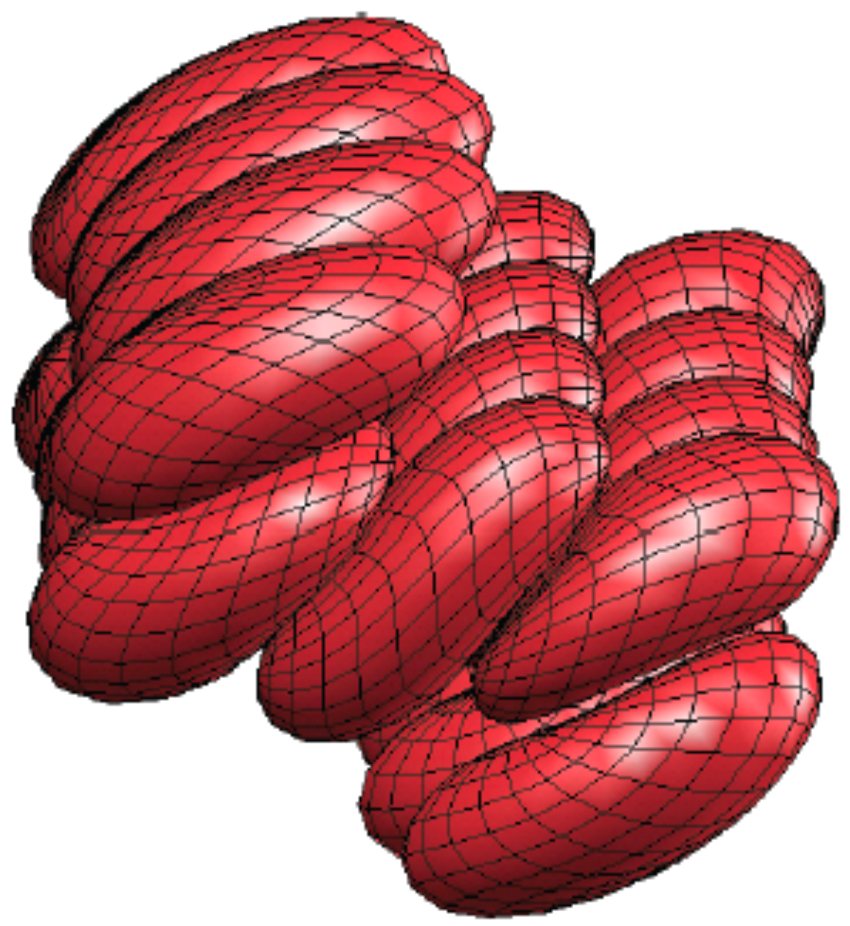


Geodesic distance [Crane et al., 2017]

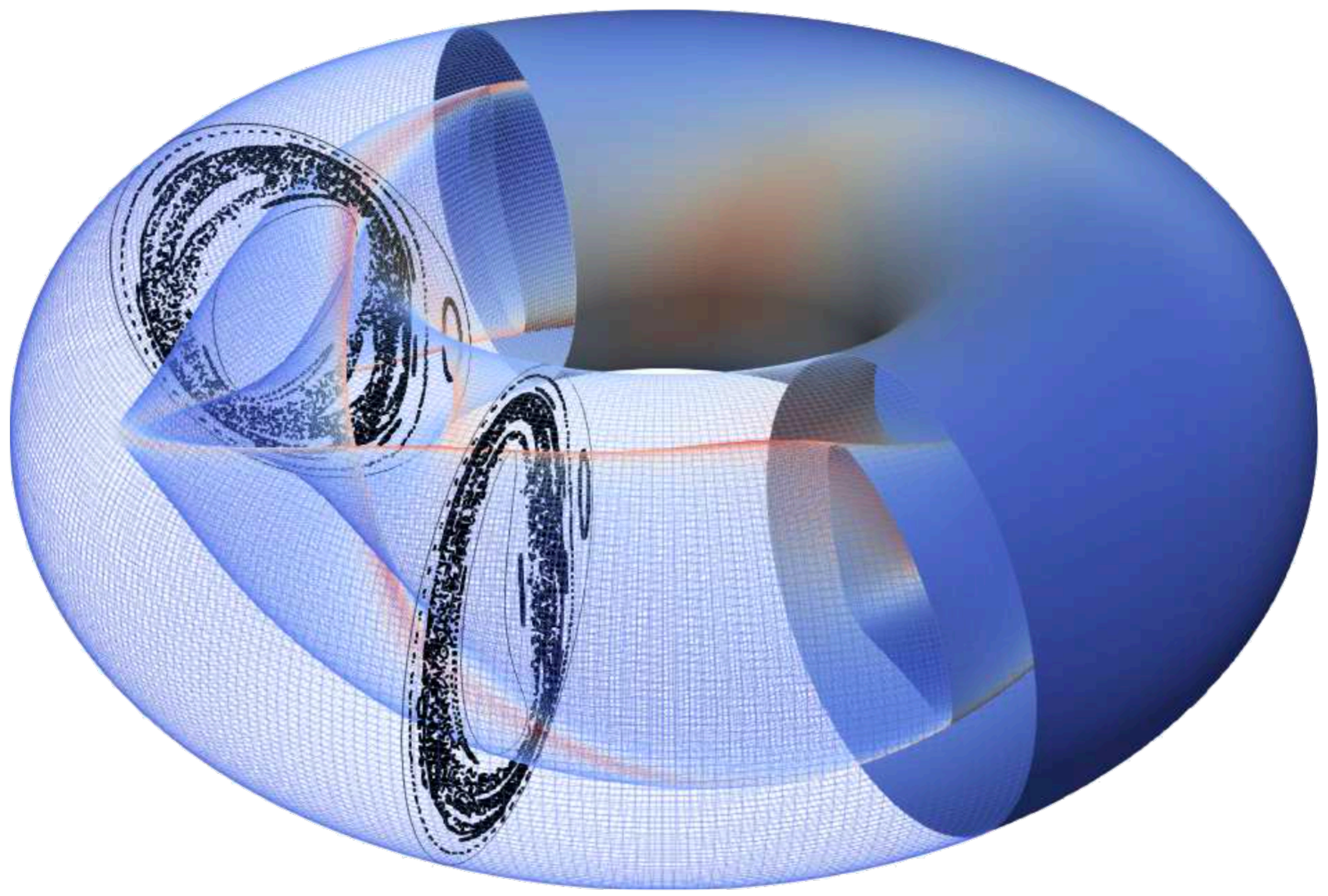


Embryonic furrow development [Image from HHMI]

Vesicle flows [Veerapaneni et al., 2011]



Stellarator design [Malhotra et al., 2019]



Surface PDEs describe the dynamics of such phenomena.

Steady-state problem

$$\mathcal{L}_\Gamma u(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma$$

- Laplace–Beltrami
- convection–diffusion
- steady Stokes

Time-dependent problem

$$\frac{\partial u}{\partial t} = \underbrace{\mathcal{L}_\Gamma u}_{\text{Linear}} + \underbrace{\mathcal{N}(u)}_{\text{Nonlinear}} \quad \text{on } \Gamma$$

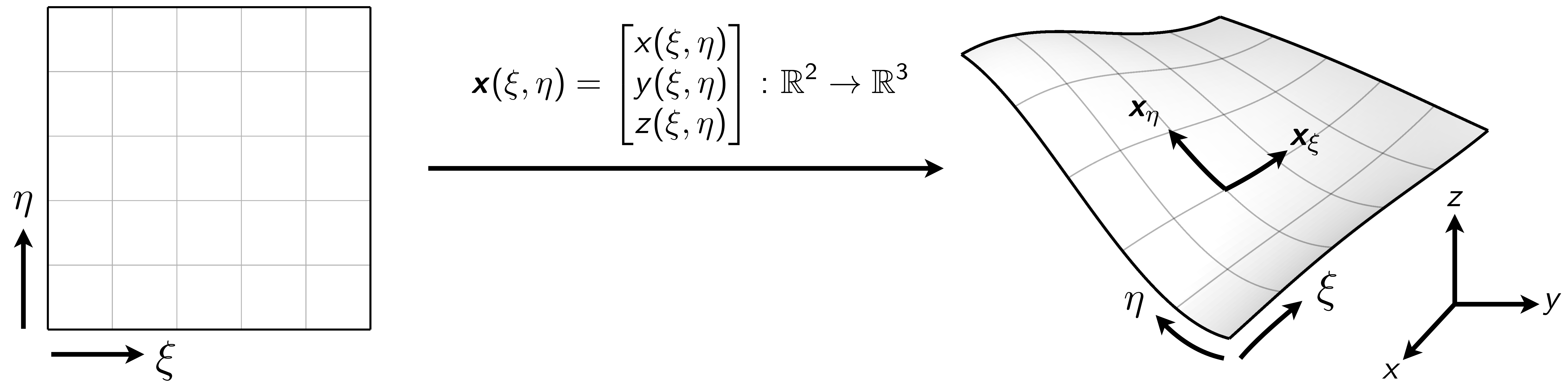
- reaction–diffusion
- heat
- Navier–Stokes

Implicit time discretization:

$$(I - \Delta t \mathcal{L}_\Gamma)\, u^{k+1} = u^k + \Delta t \mathcal{N}(u^k)$$

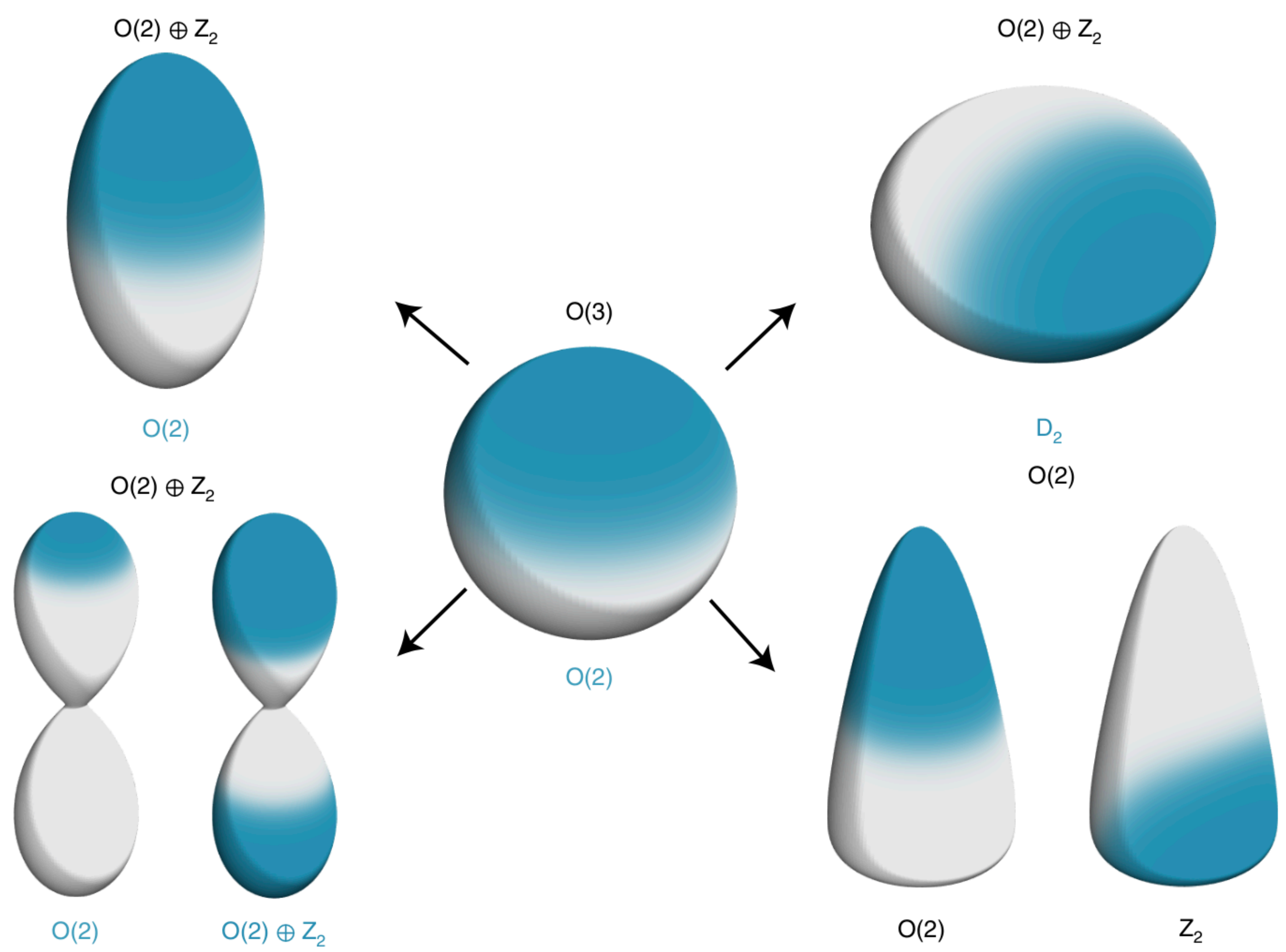
Model surface PDE: $\nabla_\Gamma \cdot (\boldsymbol{A}(\boldsymbol{x}) \nabla_\Gamma u(\boldsymbol{x})) + \nabla_\Gamma \cdot (\boldsymbol{b}(\boldsymbol{x}) u(\boldsymbol{x})) + c(\boldsymbol{x}) u(\boldsymbol{x}) = f(\boldsymbol{x})$

(+ BCs if surface is not closed)

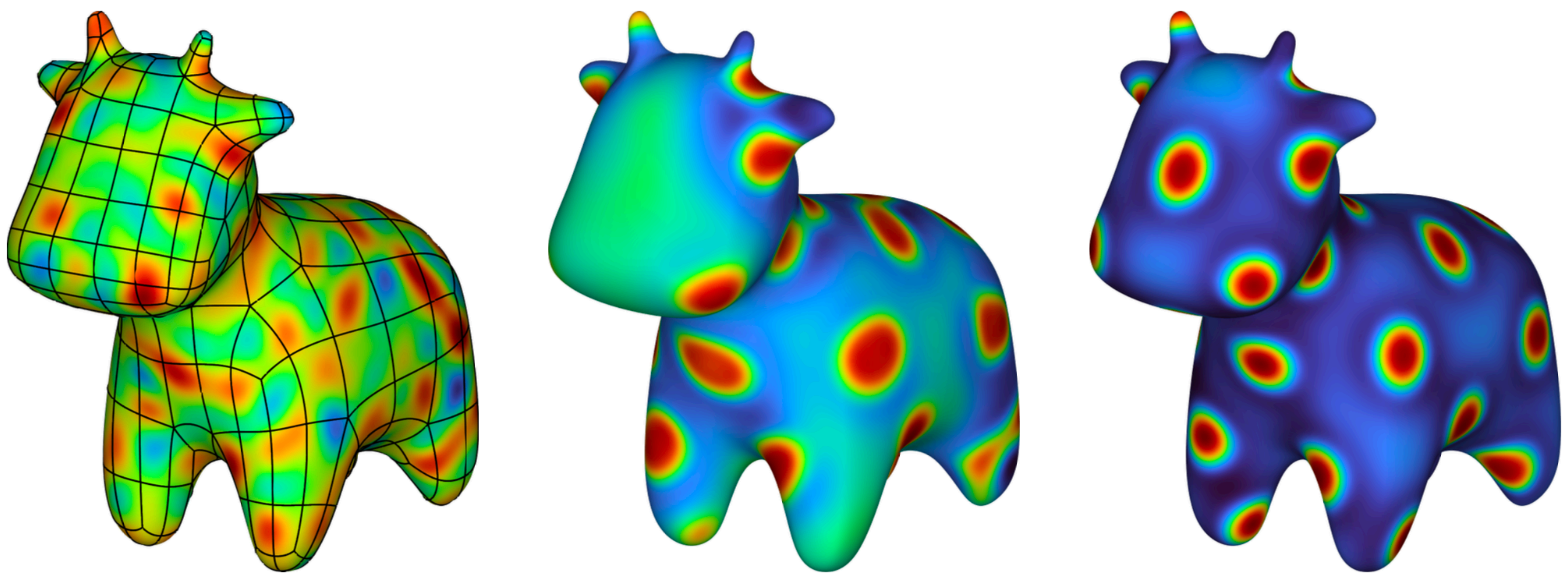


- Metric tensor $g = \begin{bmatrix} \boldsymbol{x}_\xi \cdot \boldsymbol{x}_\xi & \boldsymbol{x}_\xi \cdot \boldsymbol{x}_\eta \\ \boldsymbol{x}_\eta \cdot \boldsymbol{x}_\xi & \boldsymbol{x}_\eta \cdot \boldsymbol{x}_\eta \end{bmatrix}$ encodes how lengths and angles change along surface.
- Surface gradient: $\nabla_\Gamma u = [\boldsymbol{x}_\xi \;\; \boldsymbol{x}_\eta] \; g^{-1} \nabla_{\xi\eta} u$ $\qquad \partial_x^\Gamma = \boldsymbol{e}_x \cdot \nabla_\Gamma$
- Surface divergence: $\nabla_\Gamma \cdot \boldsymbol{u} = \frac{1}{\sqrt{\det g}} \nabla_{\xi\eta} \cdot \left(\sqrt{\det g} \, \boldsymbol{u} \right)$ $\qquad \partial_y^\Gamma = \boldsymbol{e}_y \cdot \nabla_\Gamma$
- Laplace–Beltrami: $\Delta_\Gamma u = \nabla_\Gamma \cdot \nabla_\Gamma u = \frac{1}{\sqrt{\det g}} \nabla_{\xi\eta} \cdot \left(\sqrt{\det g} \, g^{-1} \nabla_{\xi\eta} u \right)$ $\qquad \partial_z^\Gamma = \boldsymbol{e}_z \cdot \nabla_\Gamma$

A fast spectral method for cell polarization on axisymmetric surfaces



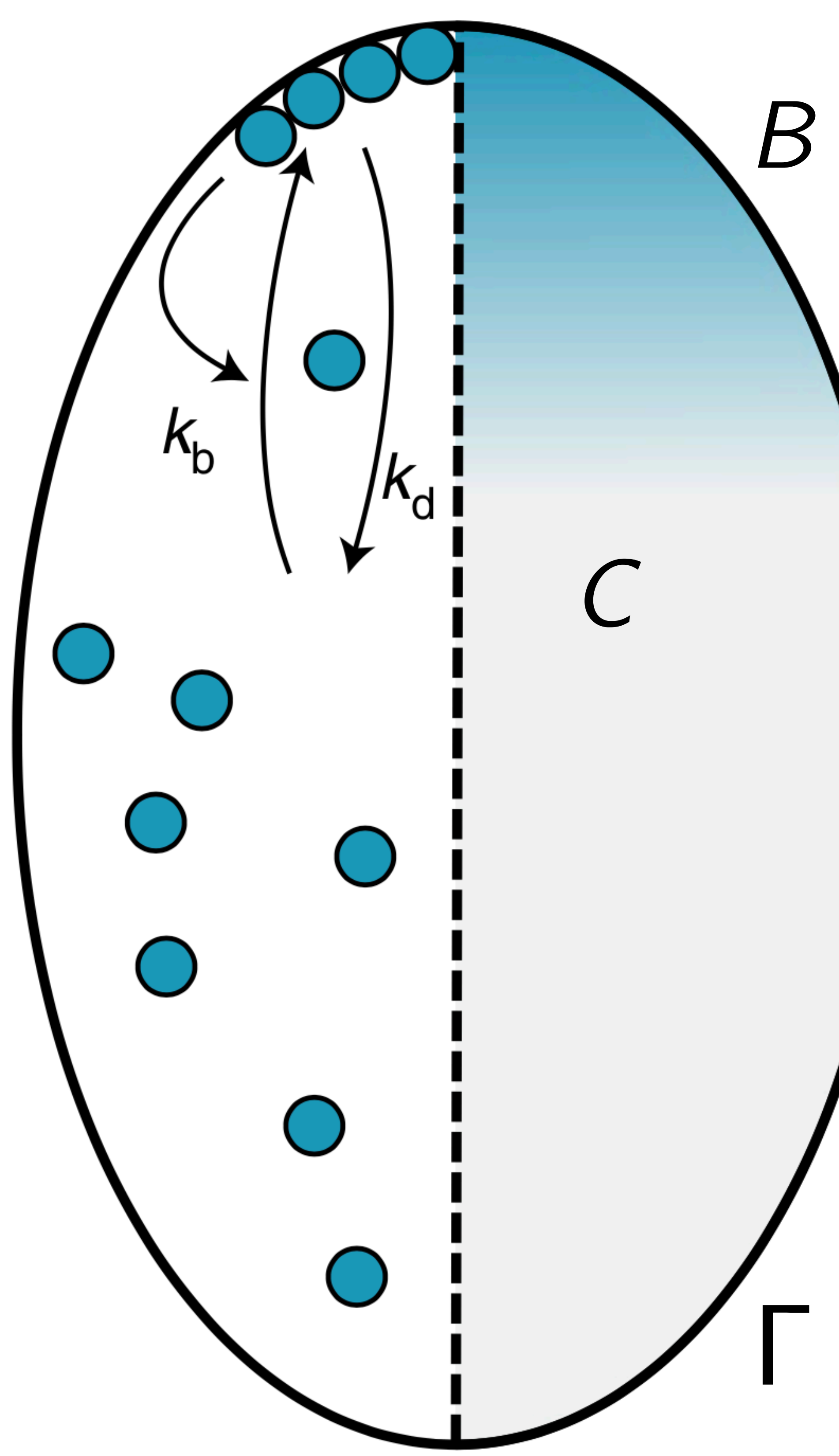
A fast high-order solver for general surface PDEs



Cell polarization

A bulk-surface model

$$\begin{cases} \dfrac{\partial B}{\partial t} = D_B \Delta_\Gamma B - k_d B + k_b \left(\beta + \dfrac{B^\nu}{G^\nu + B^\nu}\right) C & \textit{on surface} \\ \dfrac{\partial C}{\partial t} = D_C \Delta C & \textit{in bulk} \\ D_C \left(\nabla C \cdot \boldsymbol{n}\right) = k_d B - k_b \left(\beta + \dfrac{B^\nu}{G^\nu + B^\nu}\right) C & \textit{on boundary} \end{cases}$$

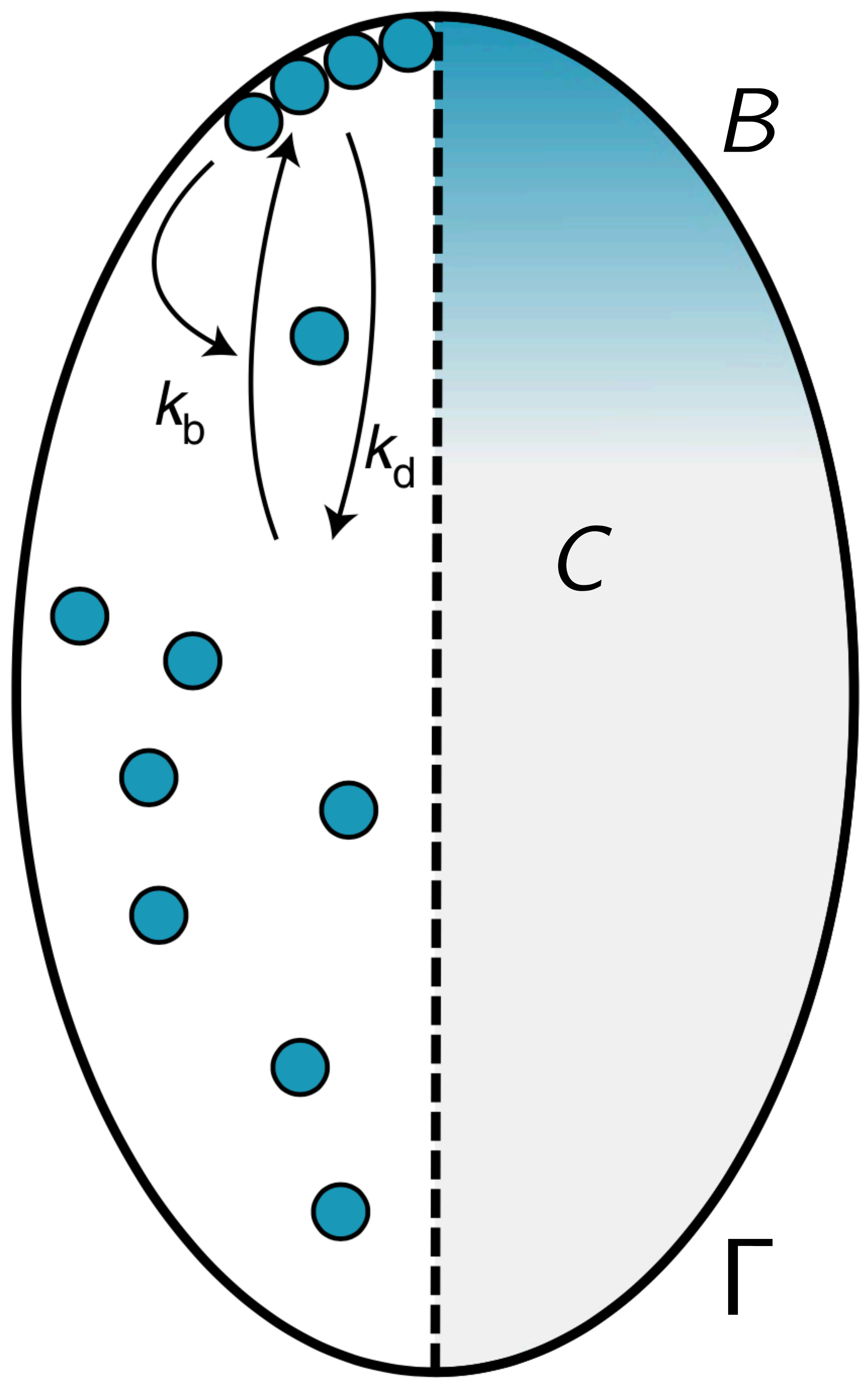


[Mori et al., 2008]
[Diegmiller et al, 2018]

Cell polarization

A bulk-surface model

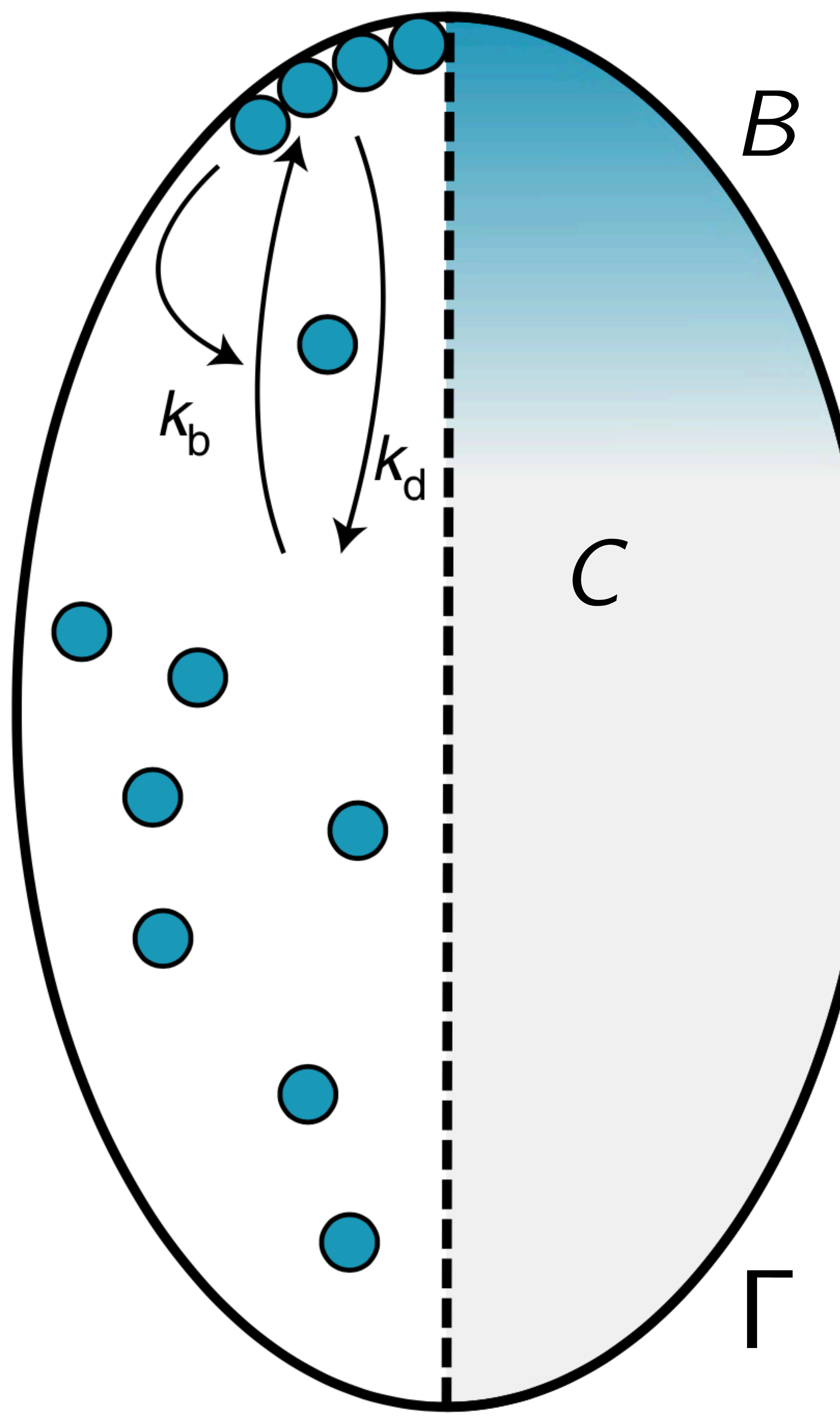
$$\begin{cases} \dfrac{\partial B}{\partial t} = \underbrace{D_B \Delta_\Gamma B}_{\text{surface diffusion}} - \underbrace{k_\mathrm{d} B}_{\text{unbinding}} + \underbrace{k_\mathrm{b}\left(\beta + \dfrac{B^\nu}{G^\nu + B^\nu}\right) C}_{\text{cooperative binding}} & \textit{on surface} \\ \dfrac{\partial C}{\partial t} = \underbrace{D_C \Delta C}_{\text{bulk diffusion}} & \textit{in bulk} \\ D_C \left(\nabla C \cdot \boldsymbol{n}\right) = k_\mathrm{d} B - k_\mathrm{b}\left(\beta + \dfrac{B^\nu}{G^\nu + B^\nu}\right) C & \textit{on boundary} \end{cases}$$



[Mori et al., 2008]
[Diegmiller et al, 2018]

$$
\begin{cases}
\dfrac{\partial B}{\partial t} = \underbrace{D_B \Delta_\Gamma B}_{\text{surface diffusion}} - \underbrace{k_\mathrm{d} B}_{\text{unbinding}} + \underbrace{k_\mathrm{b} \left(\beta + \dfrac{B^\nu}{G^\nu + B^\nu}\right) C}_{\text{cooperative binding}} & \textit{on surface} \\
\dfrac{\partial C}{\partial t} = \underbrace{D_C \Delta C}_{\text{bulk diffusion}} & \textit{in bulk} \\
D_C \left(\nabla C \cdot \boldsymbol{n}\right) = k_\mathrm{d} B - k_\mathrm{b} \left(\beta + \dfrac{B^\nu}{G^\nu + B^\nu}\right) C & \textit{on boundary}
\end{cases}
$$

- Conservation of mass: $\int_\Gamma B \, ds + \int_\Omega C \, dV = \text{constant}$
- Bulk diffusion faster than surface diffusion: $D_C \gg D_B$
- Cooperative reaction kinetics



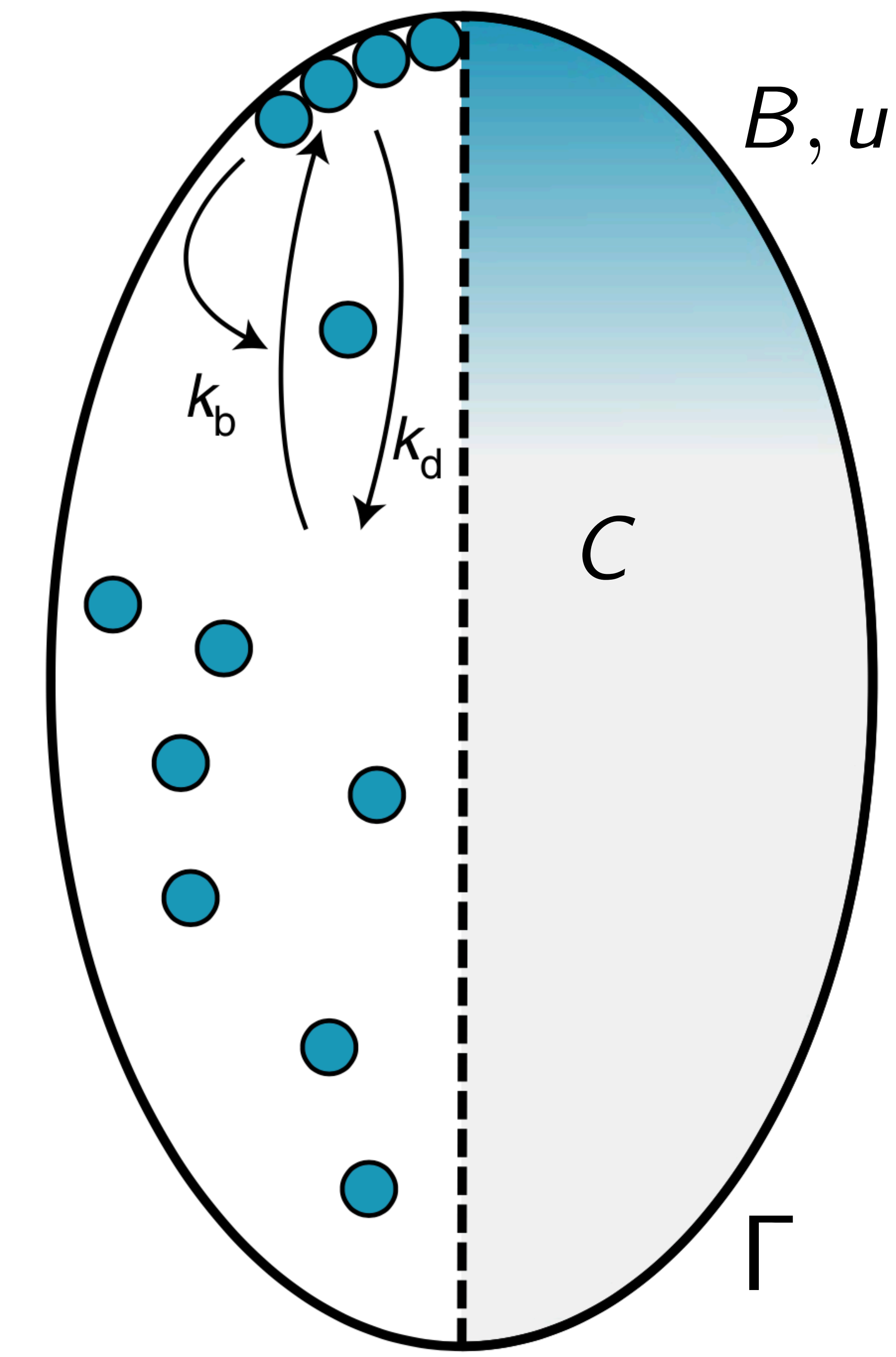
[Mori et al., 2008]
[Diegmiller et al, 2018]

$$\begin{cases} \dfrac{\partial B}{\partial t} = \underbrace{D_B \Delta_\Gamma B}_{\text{surface diffusion}} - \underbrace{k_d B}_{\text{unbinding}} + \underbrace{k_b \left(\beta + \dfrac{B^\nu}{G^\nu + B^\nu}\right) C}_{\text{cooperative binding}} & \text{on surface} \\ \dfrac{\partial C}{\partial t} = \underbrace{D_C \Delta C}_{\text{bulk diffusion}} & \text{in bulk} \\ D_C \left(\nabla C \cdot \boldsymbol{n}\right) = k_d B - k_b \left(\beta + \dfrac{B^\nu}{G^\nu + B^\nu}\right) C & \text{on boundary} \end{cases}$$

- Conservation of mass: $\int_\Gamma B\, ds + \int_\Omega C\, dV = \text{constant}$
- Bulk diffusion faster than surface diffusion: $D_C \gg D_B$
- Cooperative reaction kinetics

Assuming infinitely fast bulk diffusion, we can simplify and nondimensionalize:

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u - u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u\, ds\right) \quad \text{on surface}$$



[Mori et al., 2008]
[Diegmiller et al, 2018]

Cell polarization

A bulk-surface model

surface diffusion

$$\begin{cases} \dfrac{\partial B}{\partial t} = D_B \Delta_\Gamma B - k_\mathrm{d} B + k_\mathrm{b}\left(\beta + \dfrac{B^\nu}{G^\nu + B^\nu}\right) C & \textit{on surface} \\[2ex] \dfrac{\partial C}{\partial t} = D_C \Delta C & \textit{in bulk} \\[2ex] D_C\left(\nabla C \cdot \boldsymbol{n}\right) = k_\mathrm{d} B - k_\mathrm{b}\left(\beta + \dfrac{B^\nu}{G^\nu + B^\nu}\right) C & \textit{on boundary} \end{cases}$$

unbinding

cooperative binding

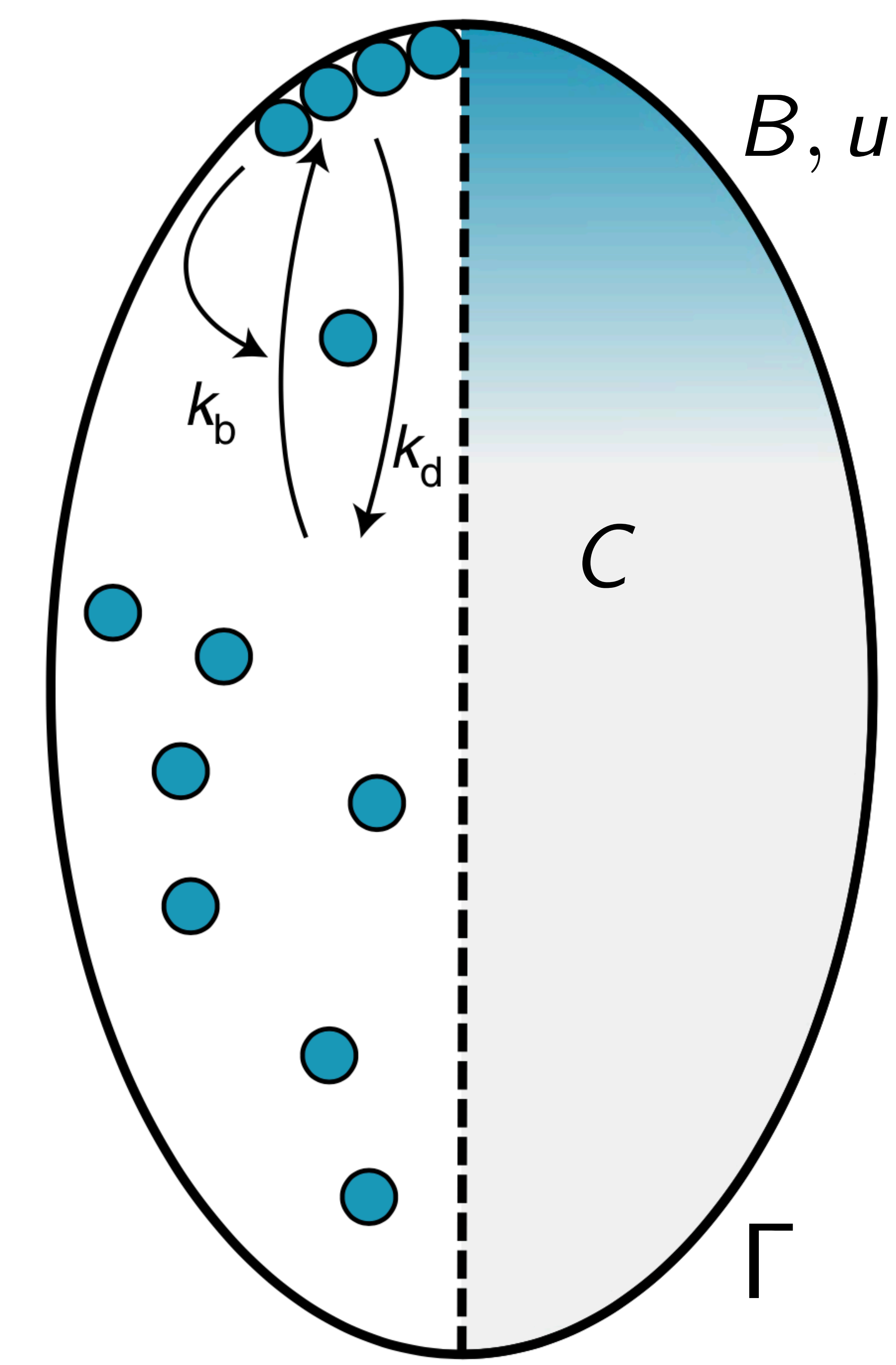
bulk diffusion

- Conservation of mass: $\int_\Gamma B\, ds + \int_\Omega C\, dV = \text{constant}$
- Bulk diffusion faster than surface diffusion: $D_C \gg D_B$
- Cooperative reaction kinetics

Assuming infinitely fast bulk diffusion, we can simplify and nondimensionalize:

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u - u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u\, ds\right) \quad \textit{on surface}$$

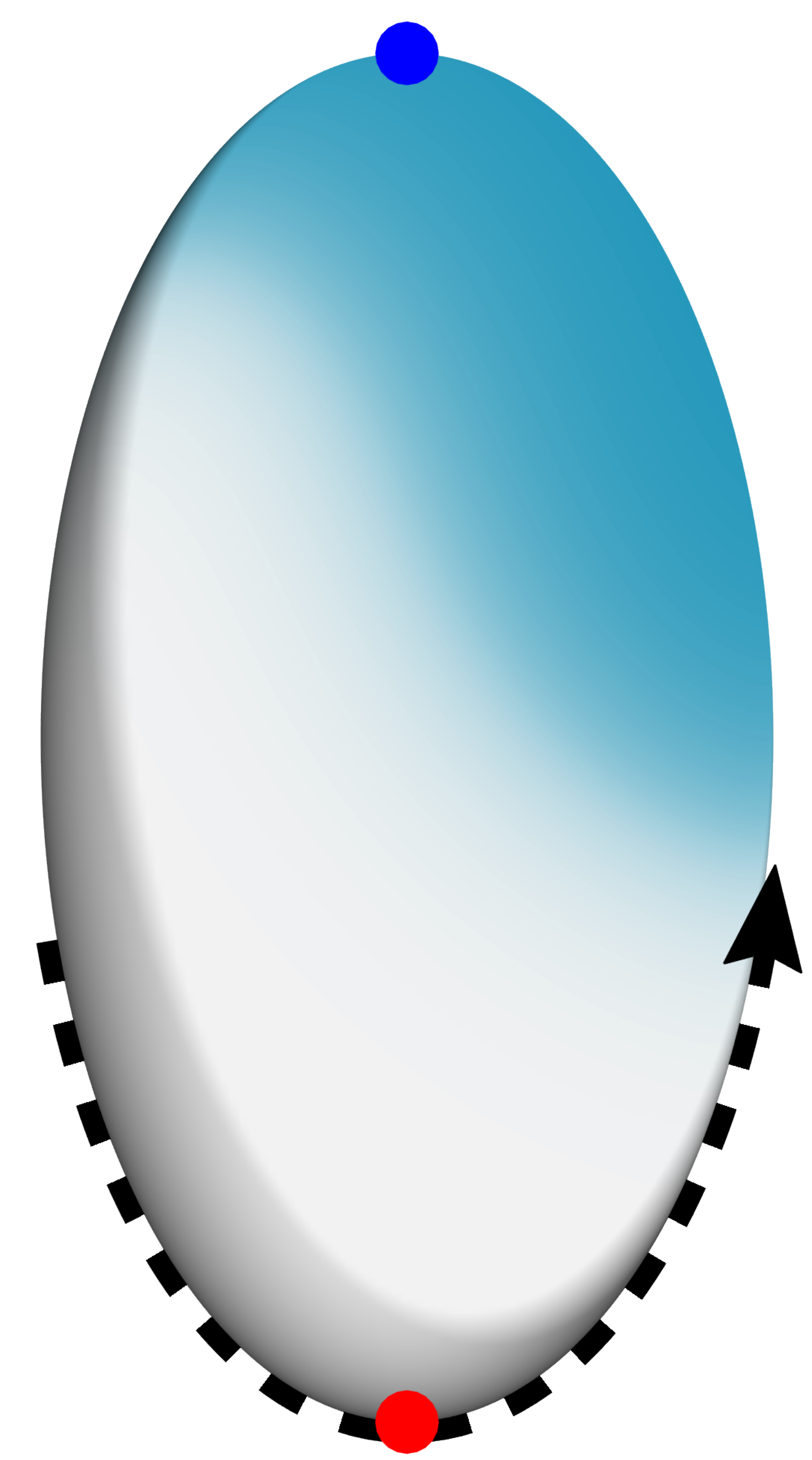
nonlocal movement via bulk



[Mori et al., 2008]
[Diegmiller et al, 2018]

Surface discretization
The double Fourier sphere method

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u - u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u \, ds\right) \quad \text{on } \Gamma$$

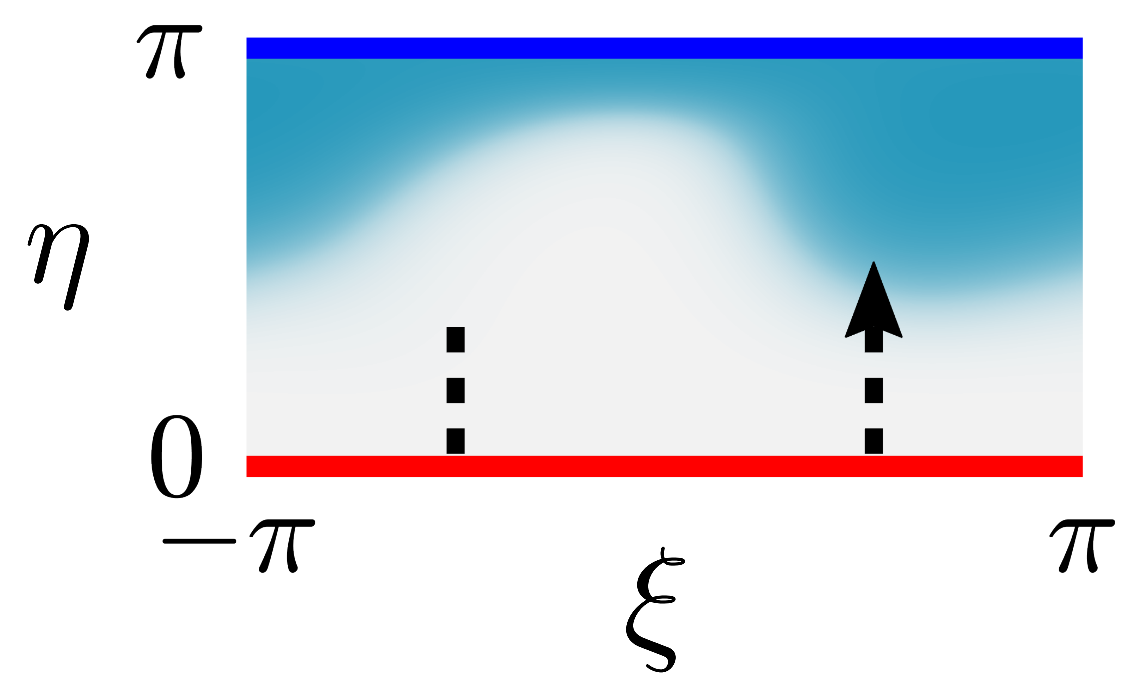


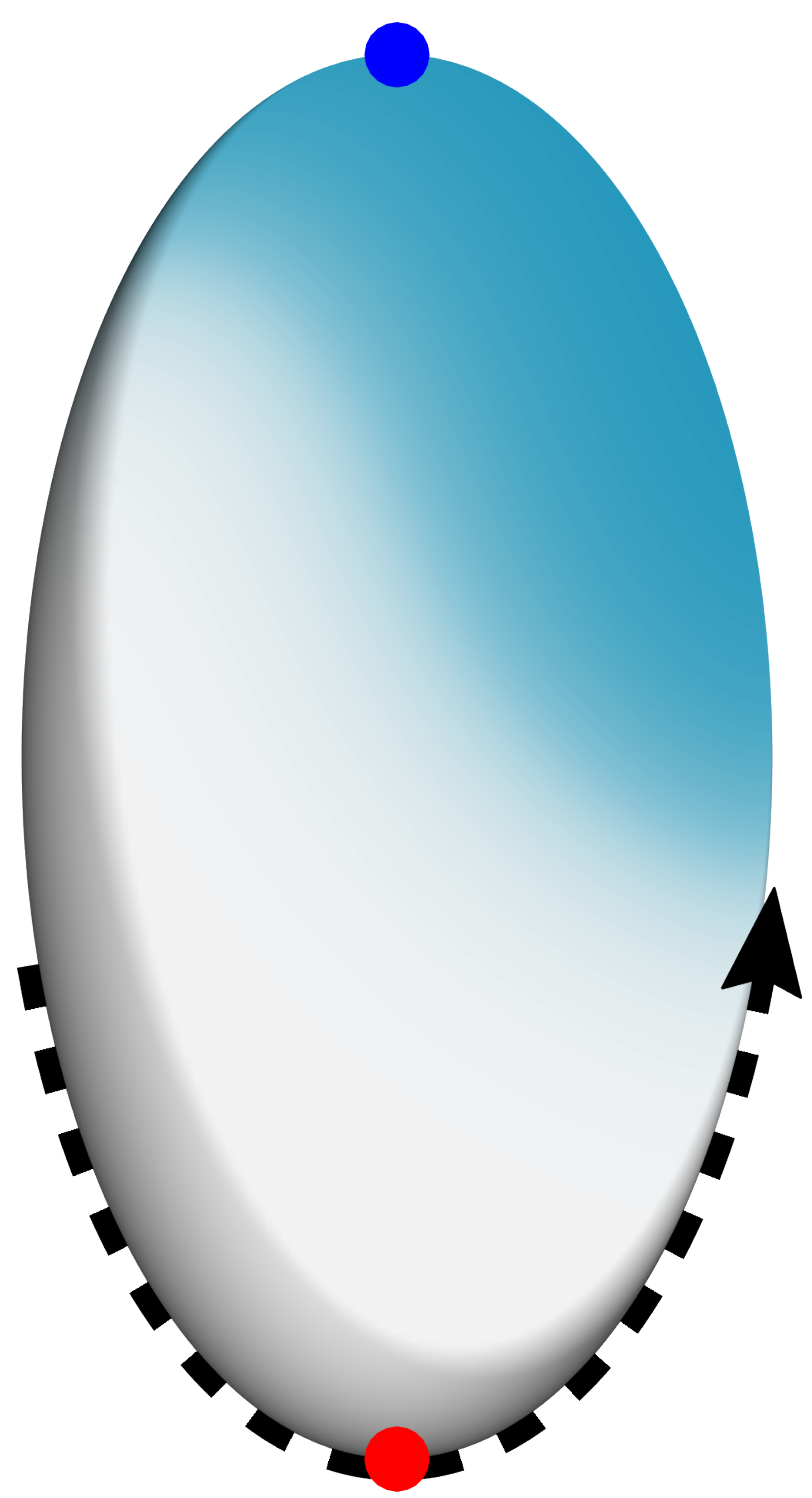
[Merilees, 1973]
[Orszag, 1974]

Surface discretization
The double Fourier sphere method

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u - u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u\, ds\right) \quad \text{on } \Gamma$$

Longitude-latitude coordinates

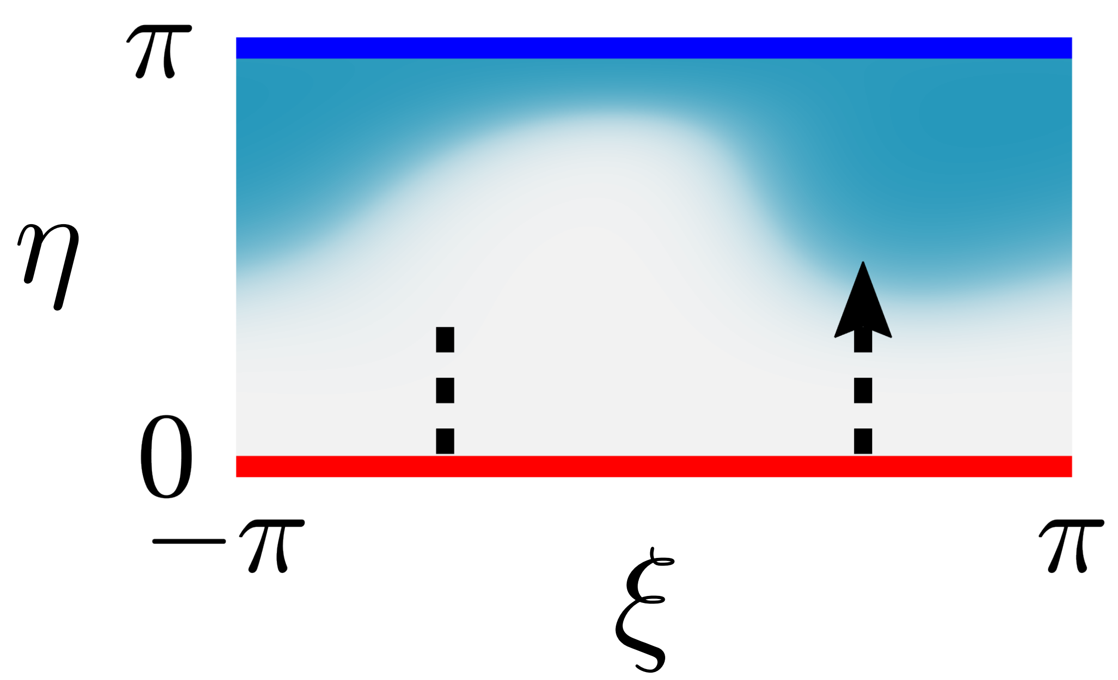




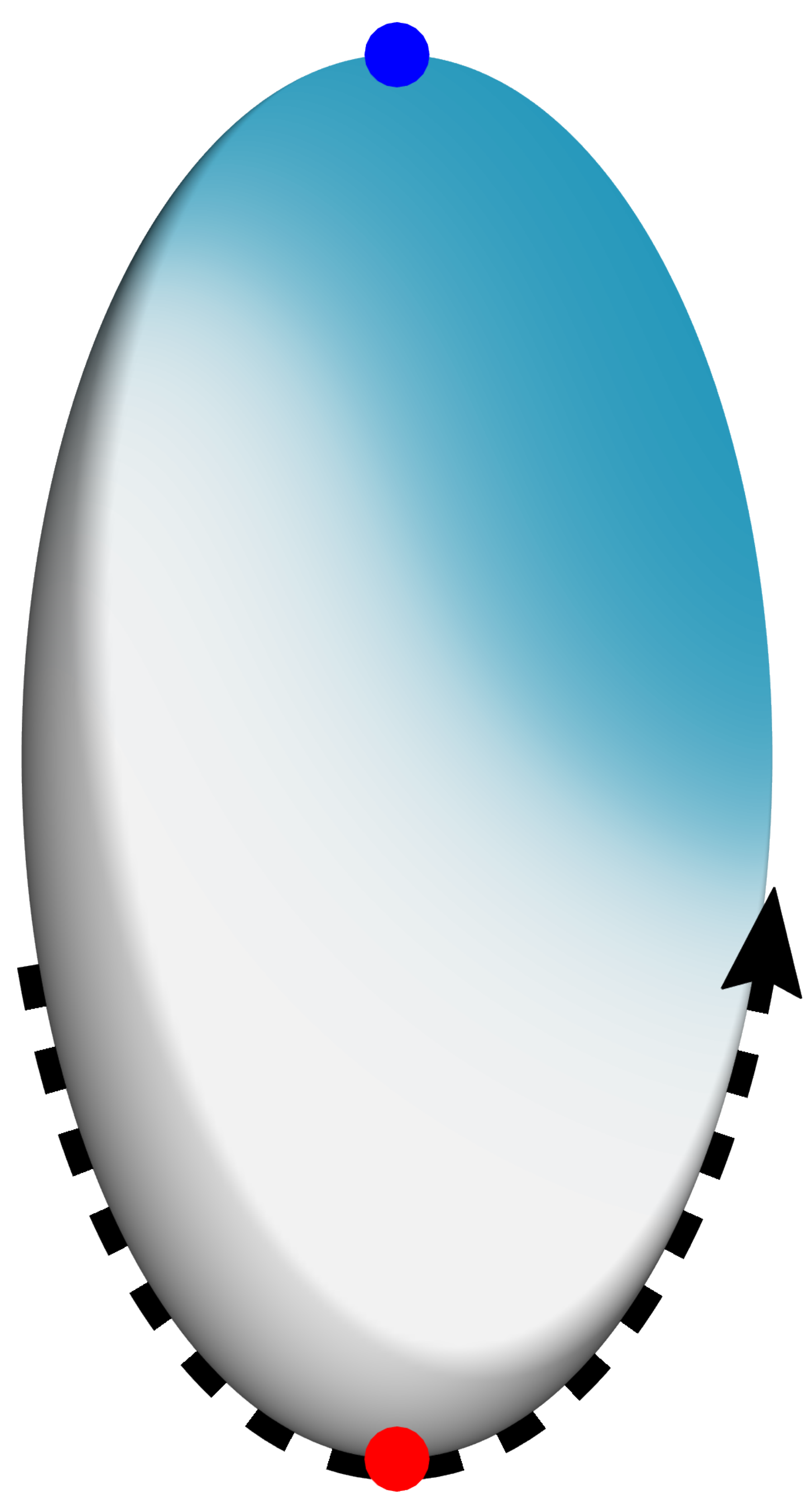
[Merilees, 1973]
[Orszag, 1974]

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u - u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u \, ds\right) \quad \text{on } \Gamma$$

Longitude-latitude coordinates



Only periodic in one dimension
What to do at pole...?

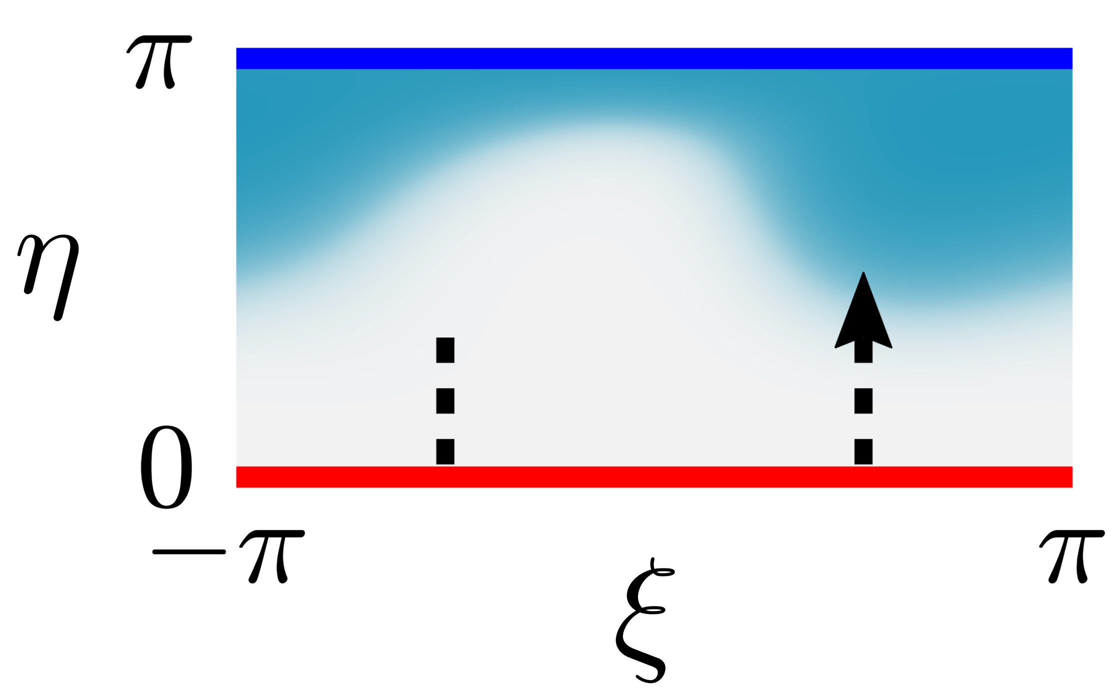


[Merilees, 1973]
[Orszag, 1974]

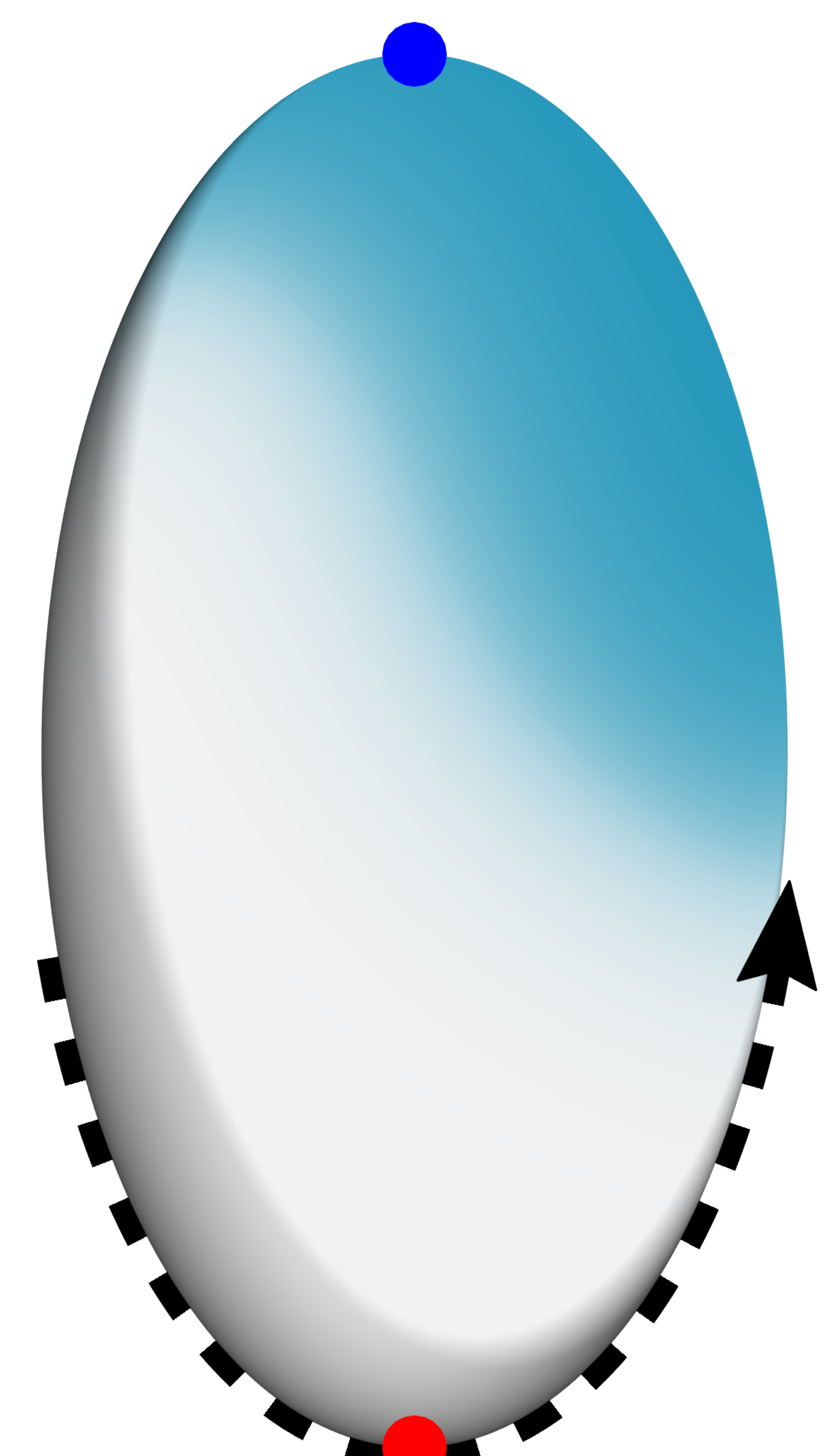
Surface discretization
The double Fourier sphere method

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u - u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u\, ds\right) \quad \text{on } \Gamma$$

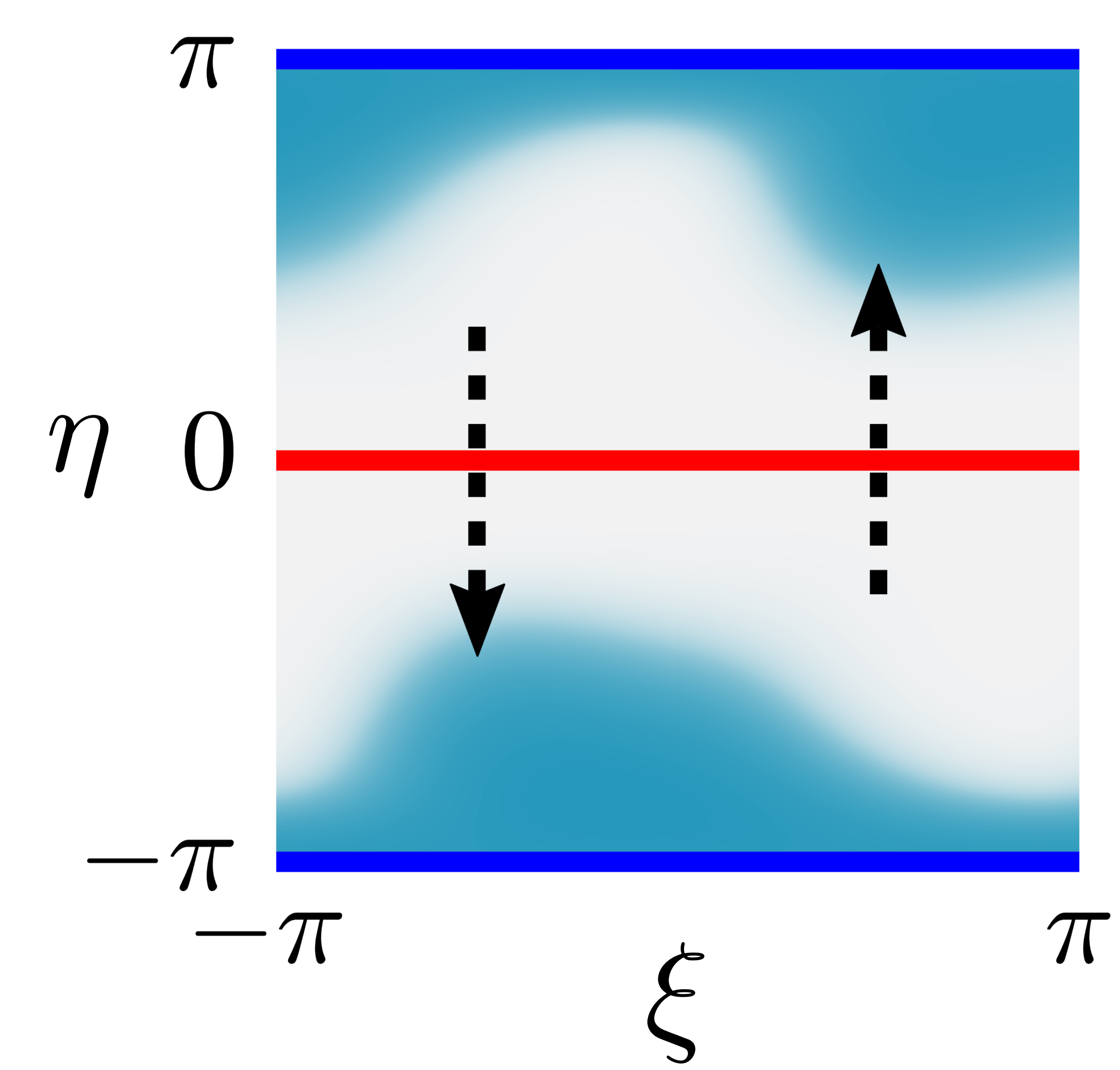
Longitude-latitude coordinates



Only periodic in one dimension
What to do at pole...?



"Doubled up" coordinates



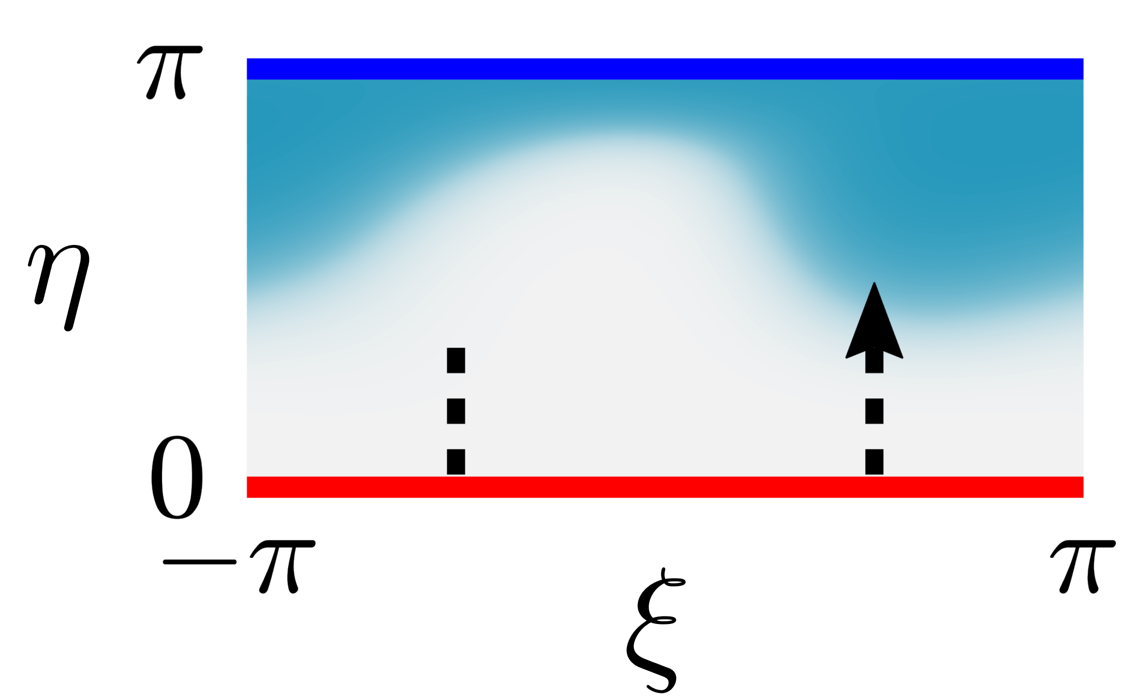
[Merilees, 1973]
[Orszag, 1974]

Surface discretization

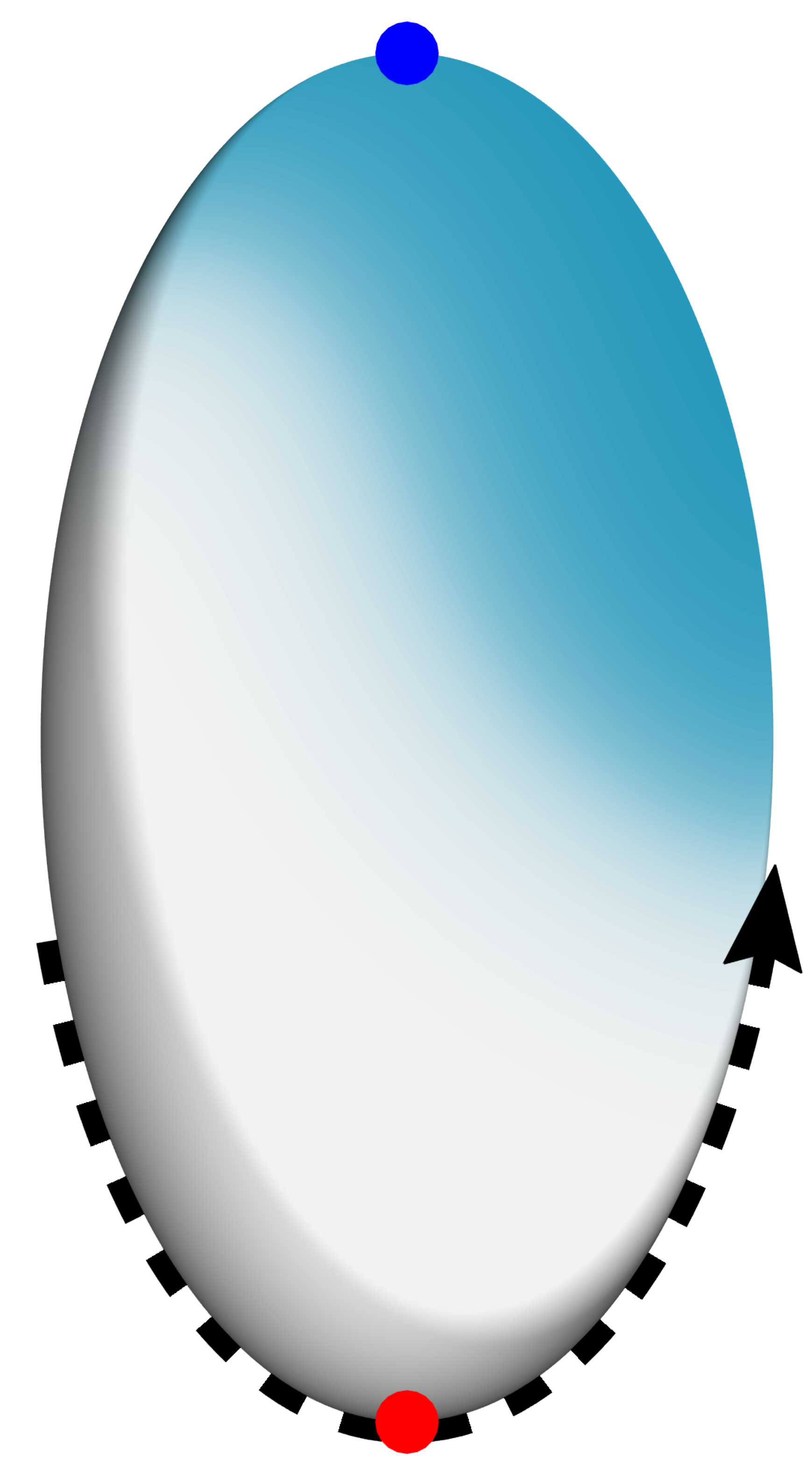
The double Fourier sphere method

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u - u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u\, ds\right) \quad \text{on } \Gamma$$

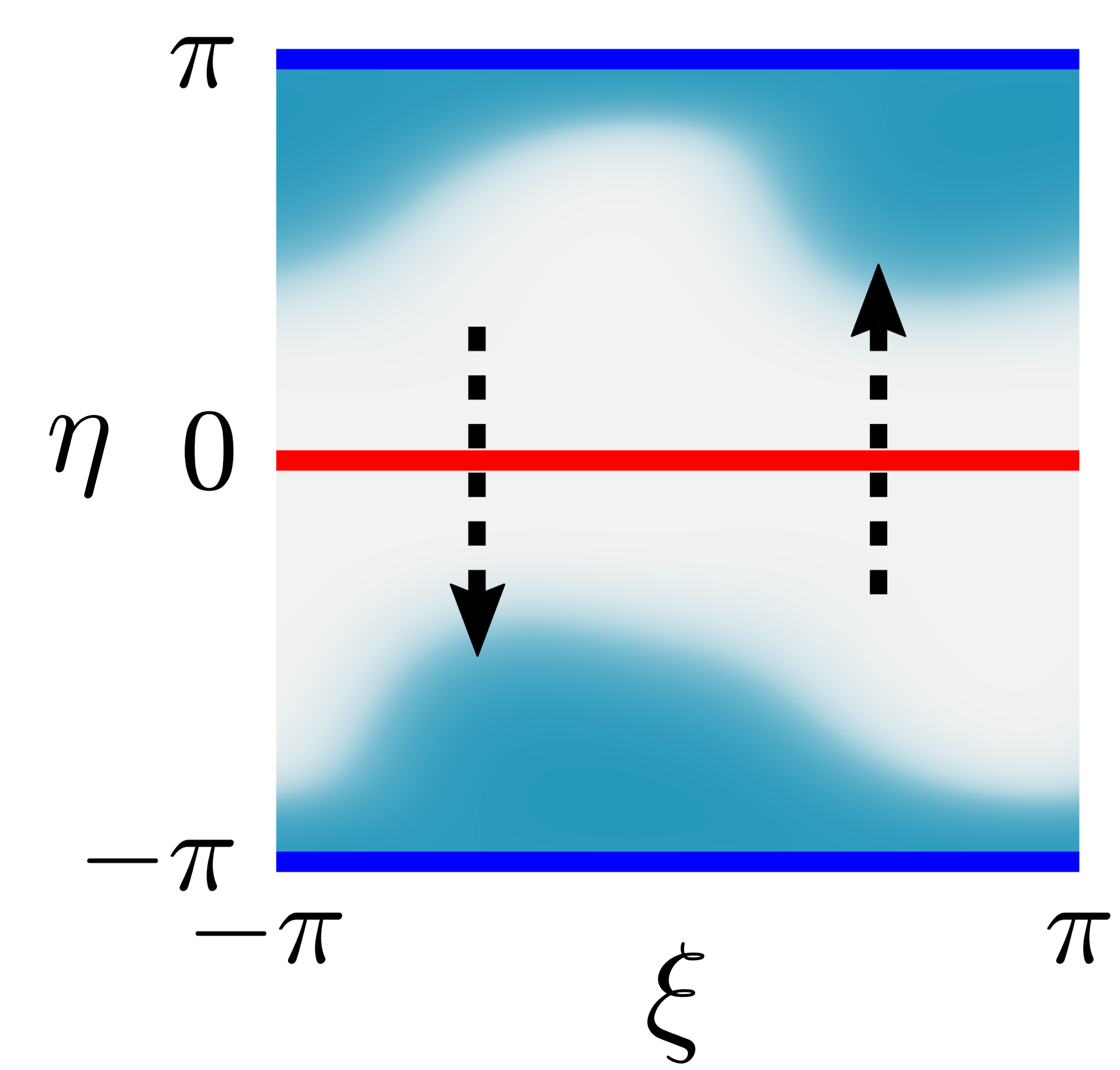
Longitude-latitude coordinates



Only periodic in one dimension
What to do at pole…?



“Doubled up” coordinates



Periodic in both dimensions!
Discretize with Fourier series

[Merilees, 1973]
[Orszag, 1974]

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u - u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u\, ds\right) \quad \text{on } \Gamma$$

More explicitly, we “double up” according to:

$$\tilde{u}(\xi,\eta) = \begin{cases} u(\xi,\eta), & (\xi,\eta) \in [-\pi,\pi] \times [0,\pi], \\ u(\xi+\pi,-\eta) & (\xi,\eta) \in [-\pi,0] \times [-\pi,0], \\ u(\xi-\pi,-\eta) & (\xi,\eta) \in [0,\pi] \times [-\pi,0]. \end{cases}$$

Surface discretization
The double Fourier sphere method

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u - u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u\, ds\right) \quad \text{on } \Gamma$$

More explicitly, we “double up” according to:

$$\tilde{u}(\xi,\eta) = \begin{cases} u(\xi,\eta), & (\xi,\eta) \in [-\pi,\pi] \times [0,\pi], \\ u(\xi+\pi,-\eta) & (\xi,\eta) \in [-\pi,0] \times [-\pi,0], \\ u(\xi-\pi,-\eta) & (\xi,\eta) \in [0,\pi] \times [-\pi,0]. \end{cases}$$

Then we can discretize using Fourier series:

$$\tilde{u}(\xi,\eta) \approx \sum_{j=-m/2}^{m/2-1} \sum_{k=-n/2}^{n/2-1} \tilde{U}_{jk} e^{ij\eta} e^{ik\xi}, \qquad (\xi,\eta) \in [-\pi,\pi]^2$$

Surface discretization

The double Fourier sphere method

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u - u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u\, ds\right) \quad \text{on } \Gamma$$

More explicitly, we "double up" according to:

$$\tilde{u}(\xi,\eta) = \begin{cases} u(\xi,\eta), & (\xi,\eta) \in [-\pi,\pi] \times [0,\pi], \\ u(\xi+\pi,-\eta) & (\xi,\eta) \in [-\pi,0] \times [-\pi,0], \\ u(\xi-\pi,-\eta) & (\xi,\eta) \in [0,\pi] \times [-\pi,0]. \end{cases}$$

Then we can discretize using Fourier series:

$$\tilde{u}(\xi,\eta) \approx \sum_{j=-m/2}^{m/2-1} \sum_{k=-n/2}^{n/2-1} \tilde{U}_{jk} e^{ij\eta} e^{ik\xi}, \qquad (\xi,\eta) \in [-\pi,\pi]^2$$

On an axisymmetric surface, the nonlocal term only interacts with the zero mode:

$$\int_\Gamma \tilde{u}\, ds \approx \sum_{j=-m/2}^{m/2-1} w_j \tilde{U}_{j0}$$

Surface discretization
The double Fourier sphere method

$$\frac{\partial u}{\partial t} = \underbrace{\delta^2 \Delta_\Gamma u}_{\mathcal{L}_\Gamma u} \underbrace{- u + \left(\beta + \frac{u^\nu}{\gamma^\nu + u^\nu}\right)\left(1 - \alpha \int_\Gamma u \, ds\right)}_{\mathcal{N}(u)} \quad \text{on } \Gamma$$

More explicitly, we "double up" according to:

$$\tilde{u}(\xi,\eta) = \begin{cases} u(\xi,\eta), & (\xi,\eta) \in [-\pi,\pi] \times [0,\pi], \\ u(\xi+\pi,-\eta) & (\xi,\eta) \in [-\pi,0] \times [-\pi,0], \\ u(\xi-\pi,-\eta) & (\xi,\eta) \in [0,\pi] \times [-\pi,0]. \end{cases}$$

Then we can discretize using Fourier series:

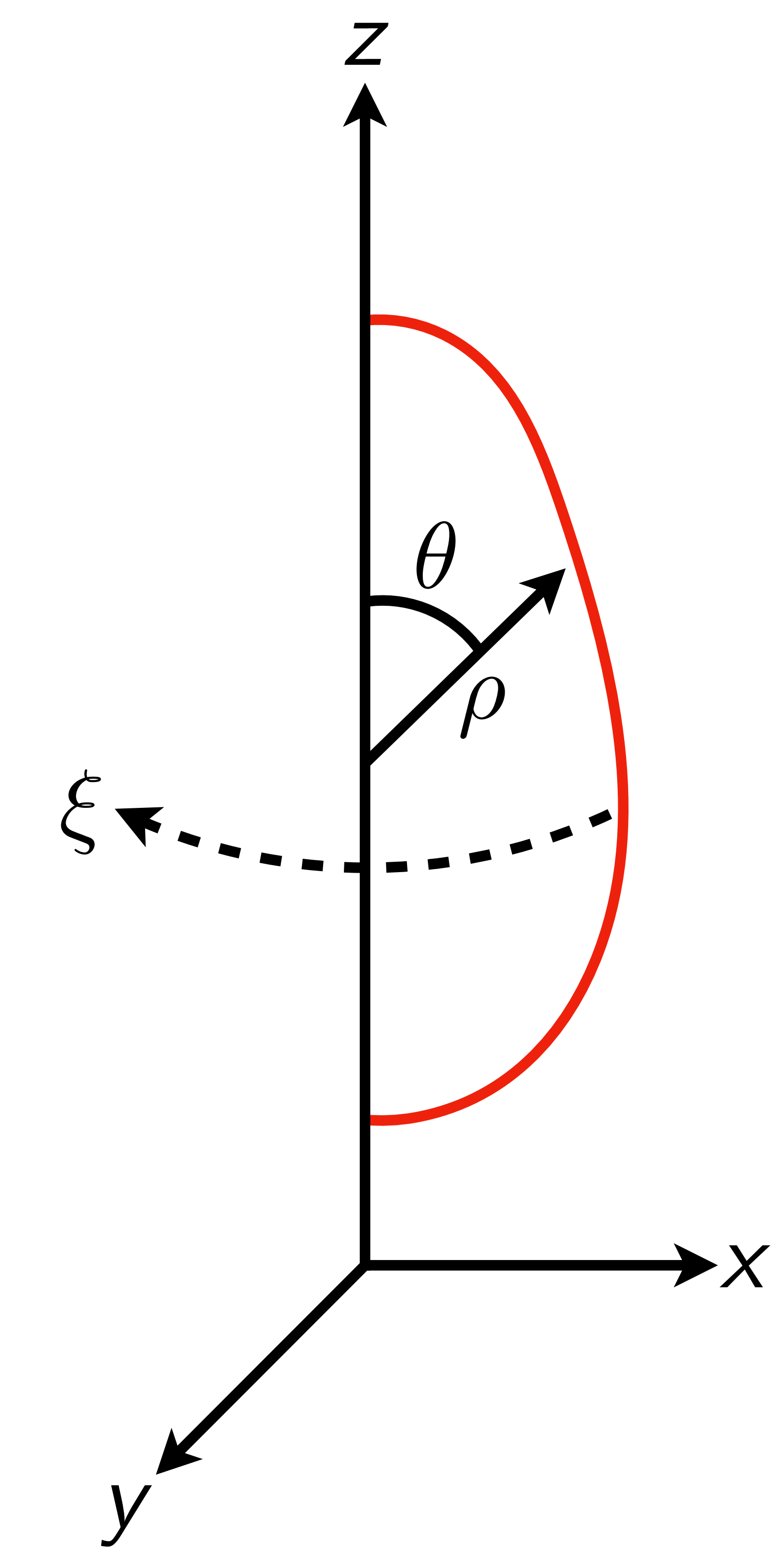
$$\tilde{u}(\xi,\eta) \approx \sum_{j=-m/2}^{m/2-1} \sum_{k=-n/2}^{n/2-1} \tilde{U}_{jk} e^{ij\eta} e^{ik\xi}, \qquad (\xi,\eta) \in [-\pi,\pi]^2$$

On an axisymmetric surface, the nonlocal term only interacts with the zero mode: $\displaystyle\int_\Gamma \tilde{u} \, ds \approx \sum_{j=-m/2}^{m/2-1} w_j \tilde{U}_{j0}$

We use an IMEX scheme in time, with diffusion treated implicitly and reaction explicitly.
At each time step, solve, e.g.:

$$(I - \Delta t \mathcal{L}_\Gamma)\, \tilde{u}^{k+1} = \tilde{u}^k + \Delta t \mathcal{N}(\tilde{u}^k)$$

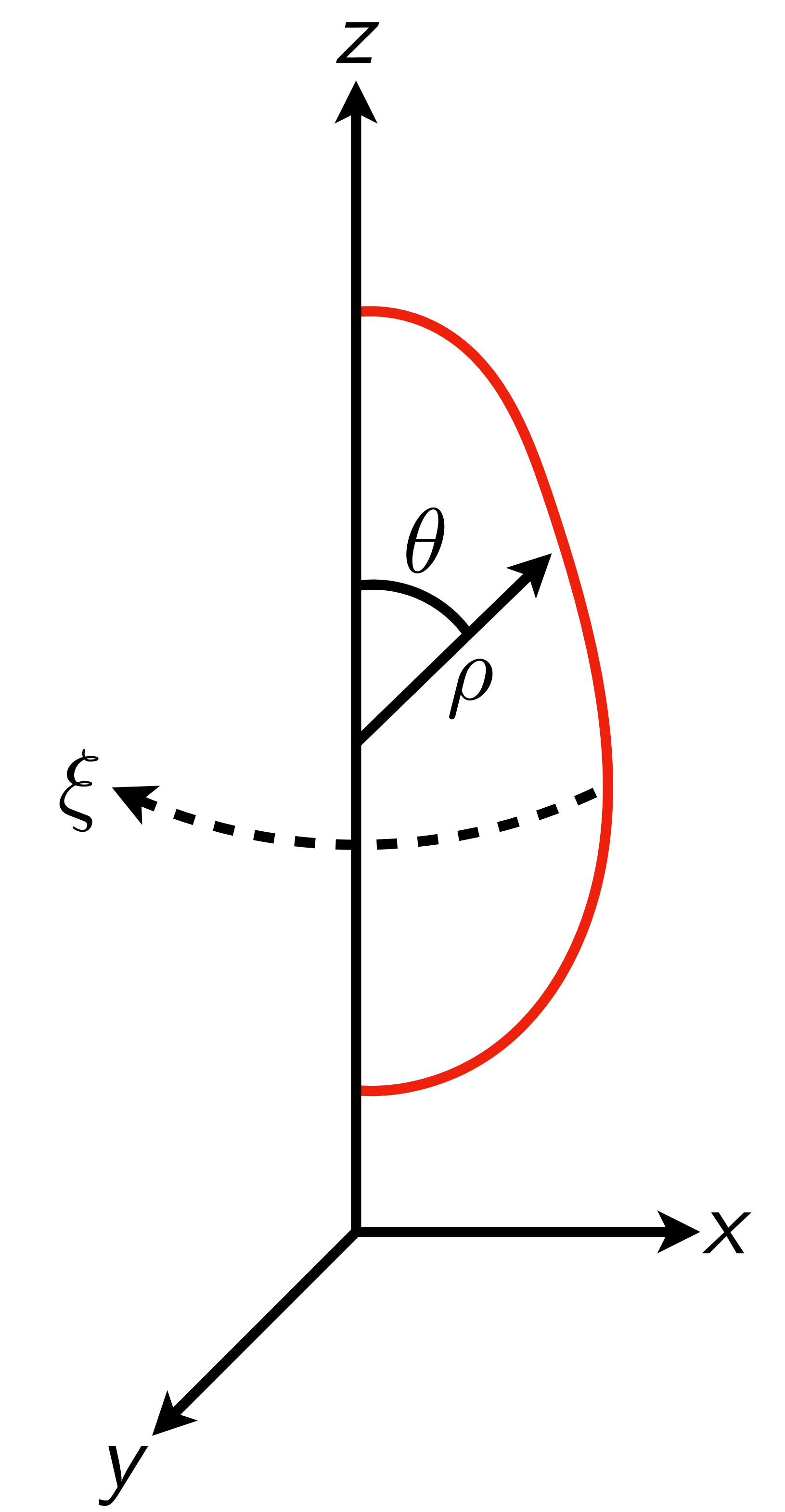
$$\Delta_\Gamma \tilde{u} - c^2 \tilde{u} = f \quad \text{on } \Gamma$$



$$x(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\cos\xi$$
$$y(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\sin\xi$$
$$z(\xi,\eta) = \rho(\eta)\cos\theta(\eta)$$

$$\Delta_\Gamma \tilde{u} - c^2 \tilde{u} = f \quad \text{on } \Gamma$$

Substitute in generating curve: $\tilde{A}(\eta)\dfrac{\partial^2 \tilde{u}}{\partial \eta^2} + \tilde{B}(\eta)\dfrac{\partial \tilde{u}}{\partial \eta} + \tilde{C}(\eta)\dfrac{\partial^2 \tilde{u}}{\partial \xi^2} - \tilde{D}(\eta)c^2\tilde{u} = \tilde{D}(\eta)\tilde{f}$

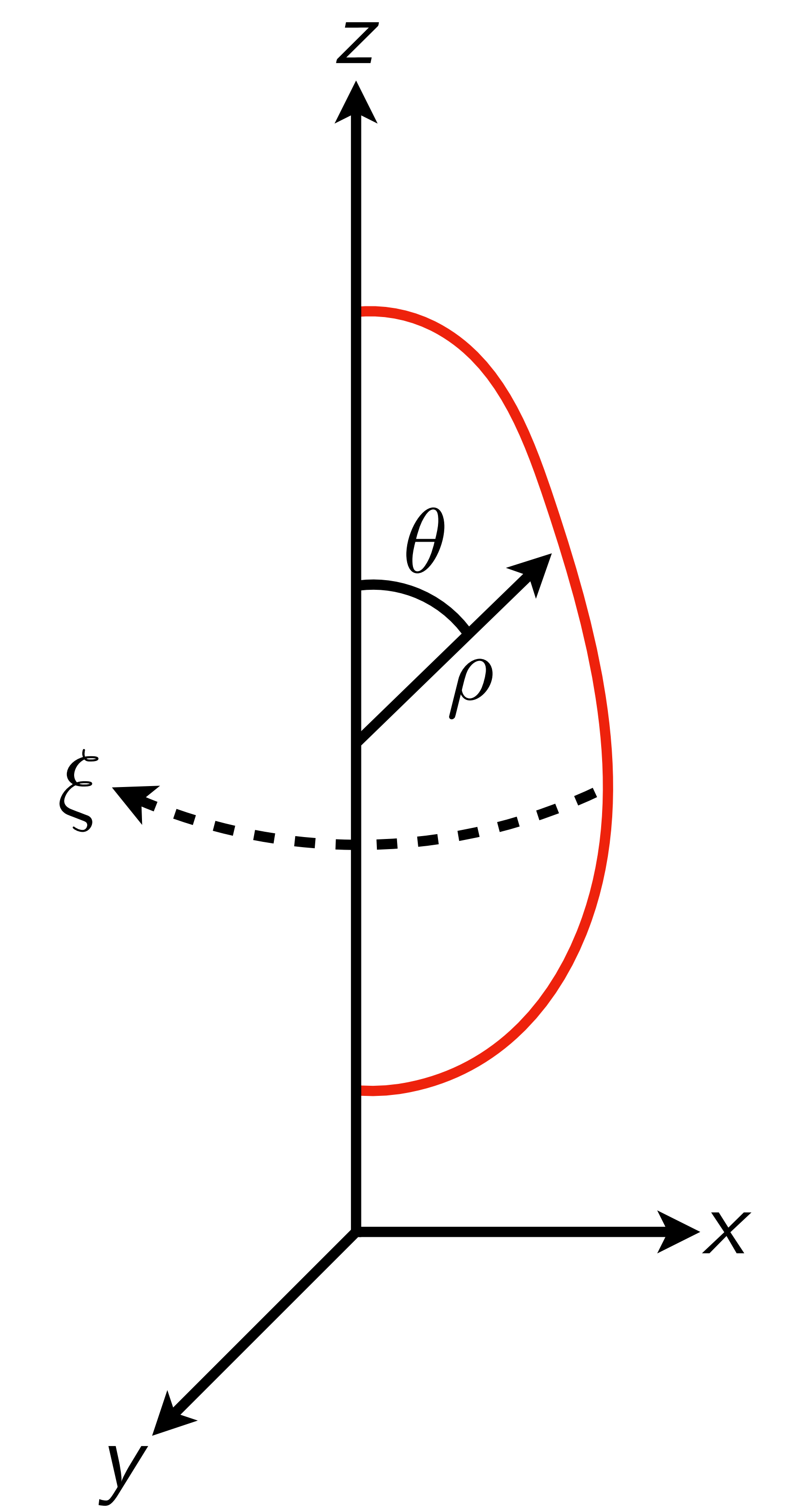


$$x(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\cos\xi$$
$$y(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\sin\xi$$
$$z(\xi,\eta) = \rho(\eta)\cos\theta(\eta)$$

$$\Delta_\Gamma \tilde{u} - c^2 \tilde{u} = f \quad \text{on } \Gamma$$

Substitute in generating curve: $\tilde{A}(\eta)\frac{\partial^2 \tilde{u}}{\partial \eta^2} + \tilde{B}(\eta)\frac{\partial \tilde{u}}{\partial \eta} + \tilde{C}(\eta)\frac{\partial^2 \tilde{u}}{\partial \xi^2} - \tilde{D}(\eta)c^2\tilde{u} = \tilde{D}(\eta)\tilde{f}$

Discretize using Fourier series: $\left(M_{\tilde{A}}D_m^2 + M_{\tilde{B}}D_m - c^2 M_{\tilde{D}}\right)\tilde{U} + M_{\tilde{C}}\tilde{U}D_n^2 = M_{\tilde{D}}\tilde{F}.$



$$x(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\cos\xi$$
$$y(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\sin\xi$$
$$z(\xi,\eta) = \rho(\eta)\cos\theta(\eta)$$

DFS on axisymmetric surfaces
A fast modified Laplace–Beltrami solver

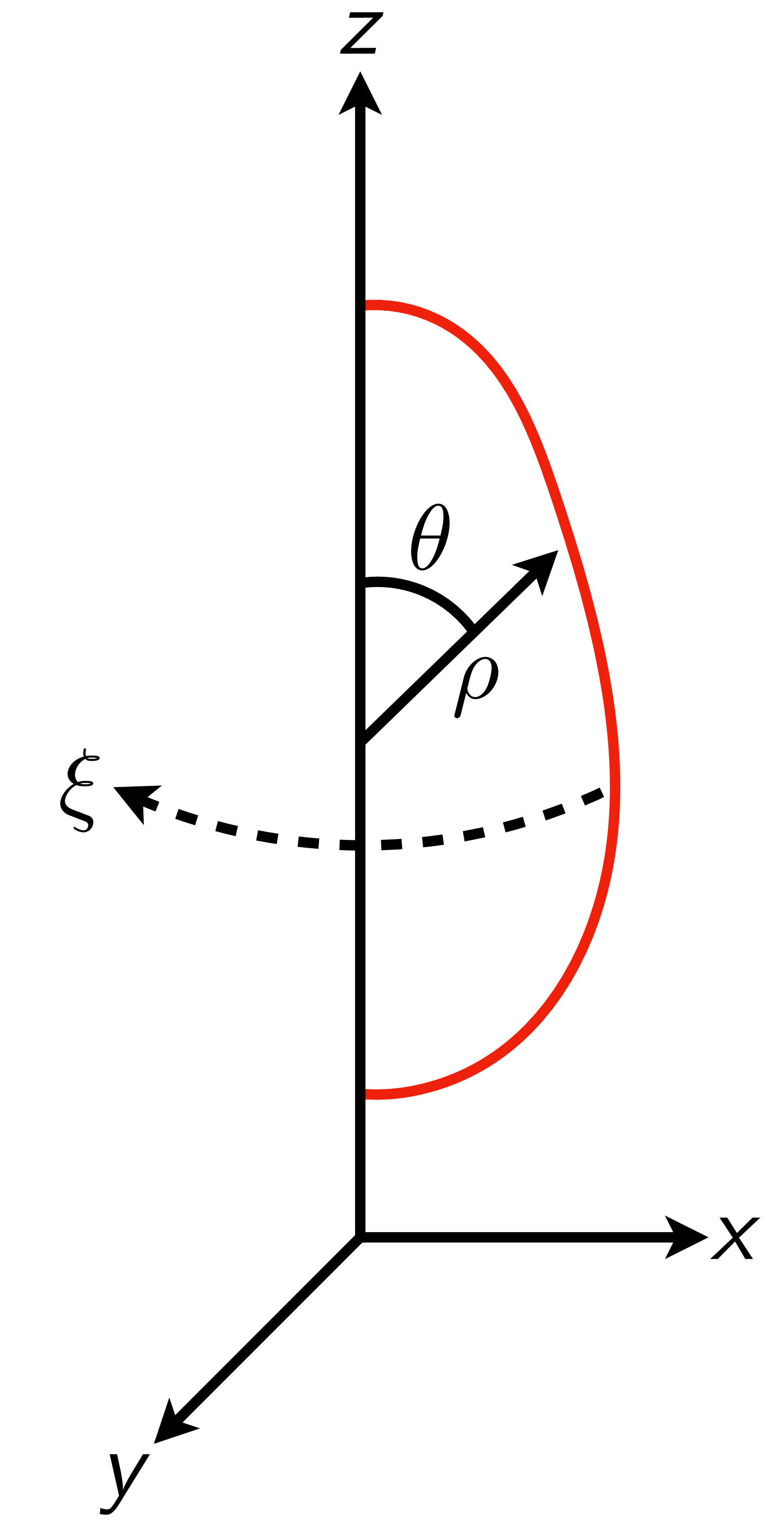
$$\Delta_\Gamma \tilde{u} - c^2 \tilde{u} = f \quad \text{on } \Gamma$$

Substitute in generating curve: $$\tilde{A}(\eta)\frac{\partial^2 \tilde{u}}{\partial \eta^2} + \tilde{B}(\eta)\frac{\partial \tilde{u}}{\partial \eta} + \tilde{C}(\eta)\frac{\partial^2 \tilde{u}}{\partial \xi^2} - \tilde{D}(\eta)c^2\tilde{u} = \tilde{D}(\eta)\tilde{f}$$

Pseudospectral multiplication

Discretize using Fourier series: $$\left(M_{\tilde{A}} D_m^2 + M_{\tilde{B}} D_m - c^2 M_{\tilde{D}}\right)\tilde{U} + M_{\tilde{C}}\tilde{U}D_n^2 = M_{\tilde{D}}\tilde{F}.$$

Spectral differentiation



$$x(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\cos\xi$$
$$y(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\sin\xi$$
$$z(\xi,\eta) = \rho(\eta)\cos\theta(\eta)$$

DFS on axisymmetric surfaces

A fast modified Laplace–Beltrami solver

$$\Delta_\Gamma \tilde{u} - c^2 \tilde{u} = f \quad \text{on } \Gamma$$

Substitute in generating curve: $\tilde{A}(\eta)\dfrac{\partial^2 \tilde{u}}{\partial \eta^2} + \tilde{B}(\eta)\dfrac{\partial \tilde{u}}{\partial \eta} + \tilde{C}(\eta)\dfrac{\partial^2 \tilde{u}}{\partial \xi^2} - \tilde{D}(\eta)c^2\tilde{u} = \tilde{D}(\eta)\tilde{f}$

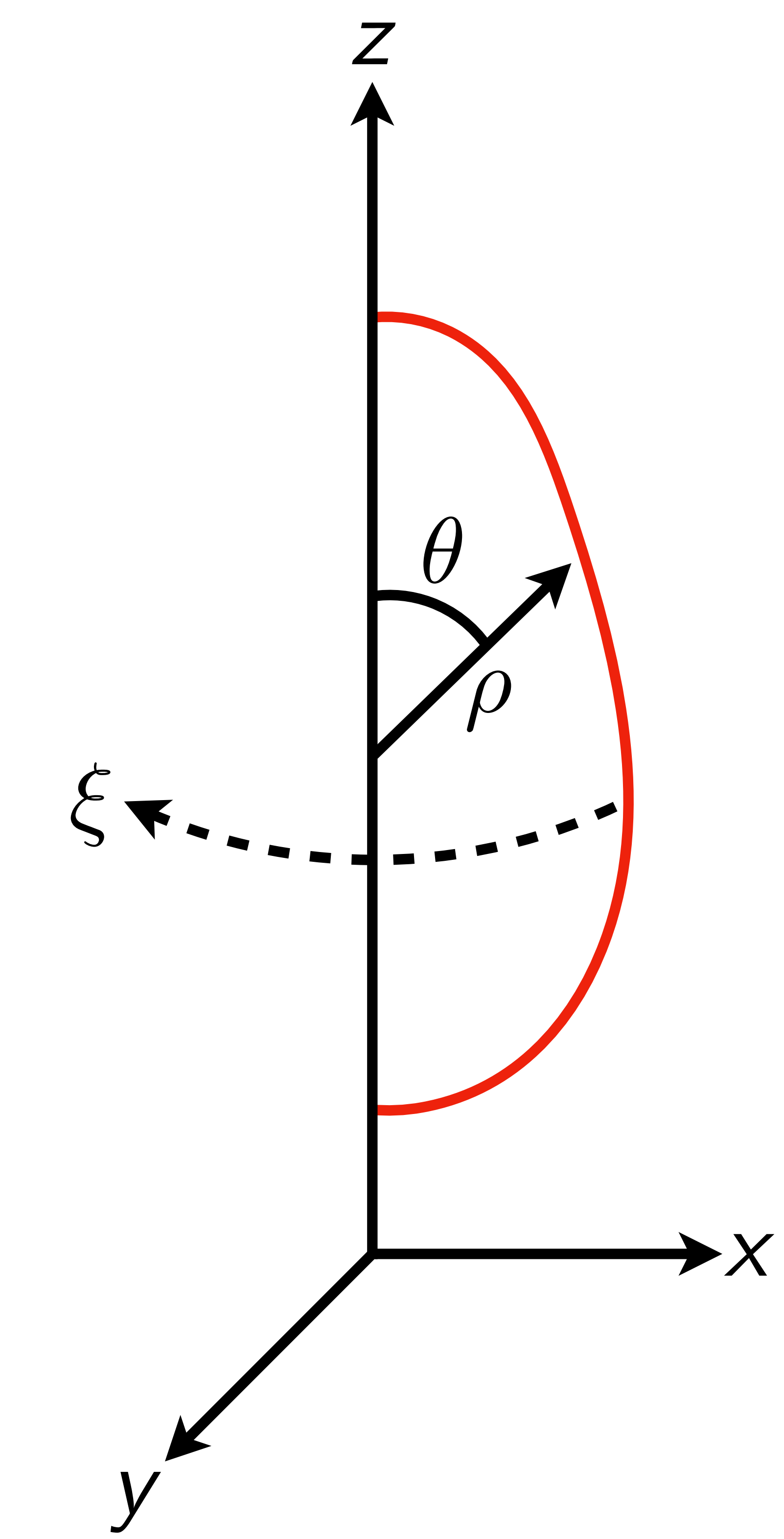
Pseudospectral multiplication

Discretize using Fourier series: $\left(M_{\tilde{A}}D_m^2 + M_{\tilde{B}}D_m - c^2 M_{\tilde{D}}\right)\tilde{U} + M_{\tilde{C}}\tilde{U}D_n^2 = M_{\tilde{D}}\tilde{F}.$

Spectral differentiation

Matrix equation decouples azimuthally into n banded linear systems of size $m \times m$:

$$\left(M_{\tilde{A}}D_m^2 + M_{\tilde{B}}D_m - c^2 M_{\tilde{D}} + M_{\tilde{C}}(D_n^2)_{kk}\right)\tilde{U}_{:,k} = M_{\tilde{D}}\tilde{F}_{:,k},$$



$$x(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\cos\xi$$
$$y(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\sin\xi$$
$$z(\xi,\eta) = \rho(\eta)\cos\theta(\eta)$$

DFS on axisymmetric surfaces

A fast modified Laplace–Beltrami solver

$$\Delta_\Gamma \tilde{u} - c^2 \tilde{u} = f \quad \text{on } \Gamma$$

Substitute in generating curve: $\tilde{A}(\eta)\dfrac{\partial^2 \tilde{u}}{\partial \eta^2} + \tilde{B}(\eta)\dfrac{\partial \tilde{u}}{\partial \eta} + \tilde{C}(\eta)\dfrac{\partial^2 \tilde{u}}{\partial \xi^2} - \tilde{D}(\eta)c^2\tilde{u} = \tilde{D}(\eta)\tilde{f}$

Pseudospectral multiplication

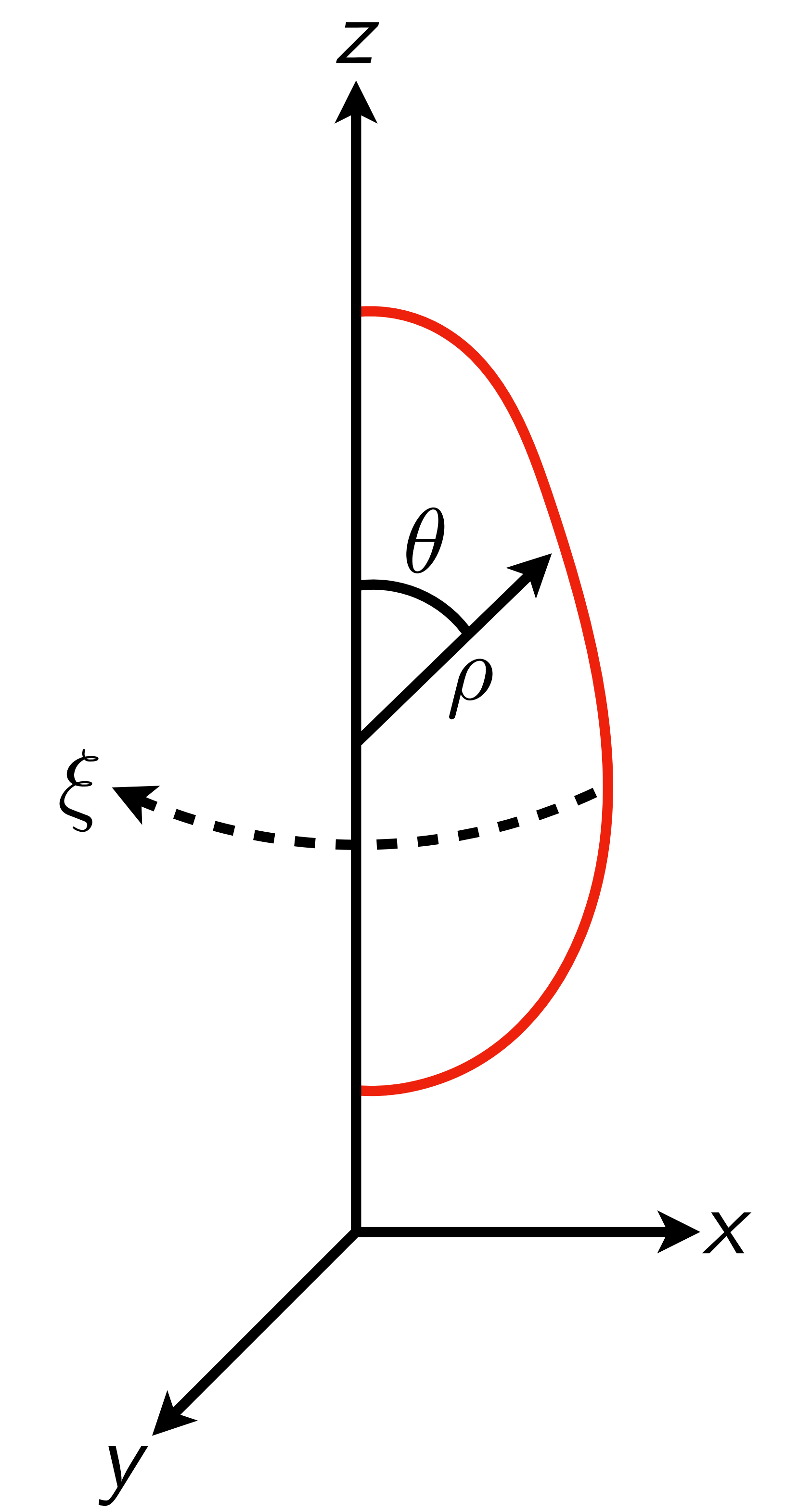
Discretize using Fourier series: $\left(M_{\tilde{A}} D_m^2 + M_{\tilde{B}} D_m - c^2 M_{\tilde{D}}\right)\tilde{U} + M_{\tilde{C}}\tilde{U}D_n^2 = M_{\tilde{D}}\tilde{F}.$

Spectral differentiation

Matrix equation decouples azimuthally into n banded linear systems of size $m \times m$:

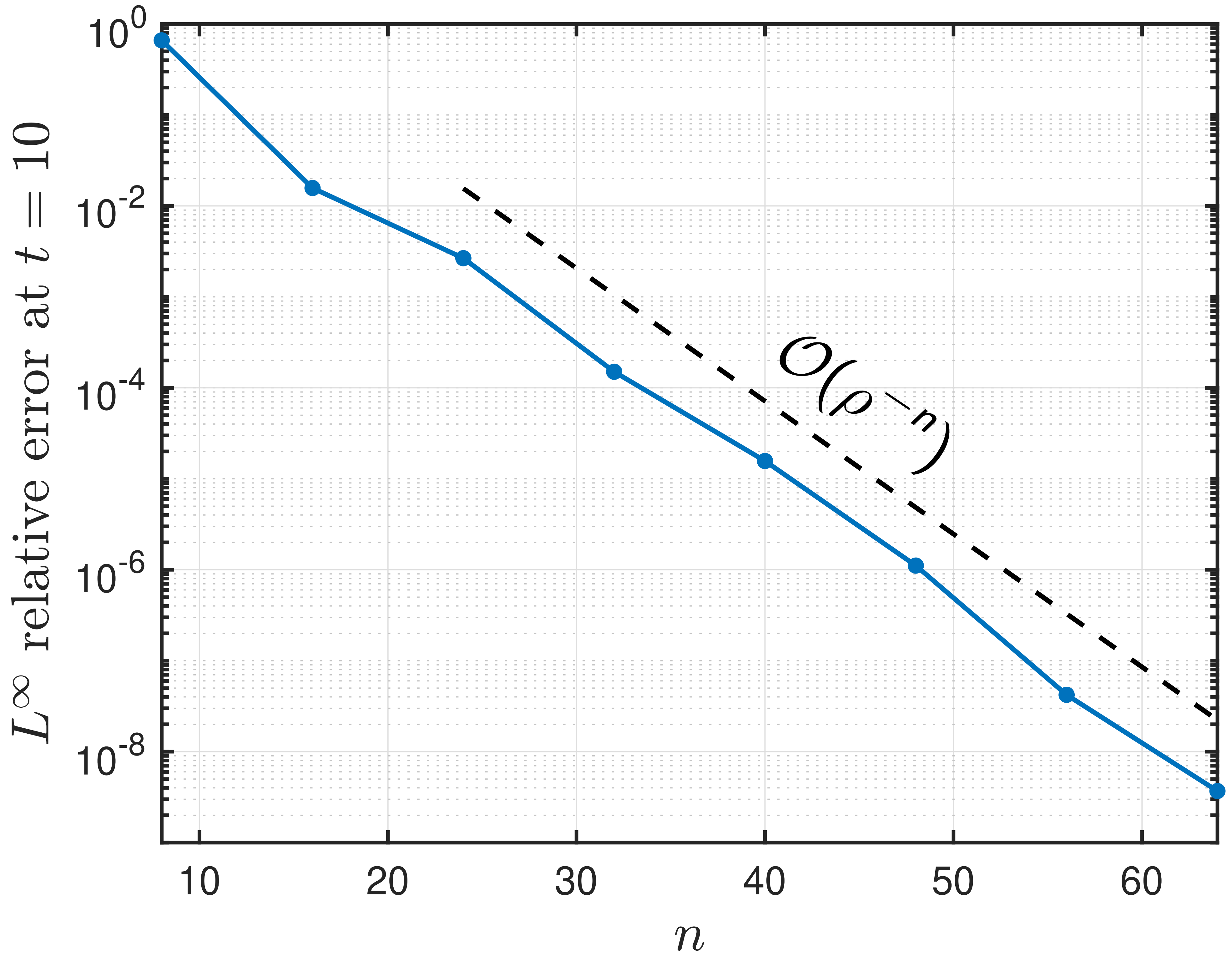
$$\left(M_{\tilde{A}} D_m^2 + M_{\tilde{B}} D_m - c^2 M_{\tilde{D}} + M_{\tilde{C}}(D_n^2)_{kk}\right)\tilde{U}_{:,k} = M_{\tilde{D}}\tilde{F}_{:,k},$$

Overall solver complexity: $\mathcal{O}(mn \log mn)$



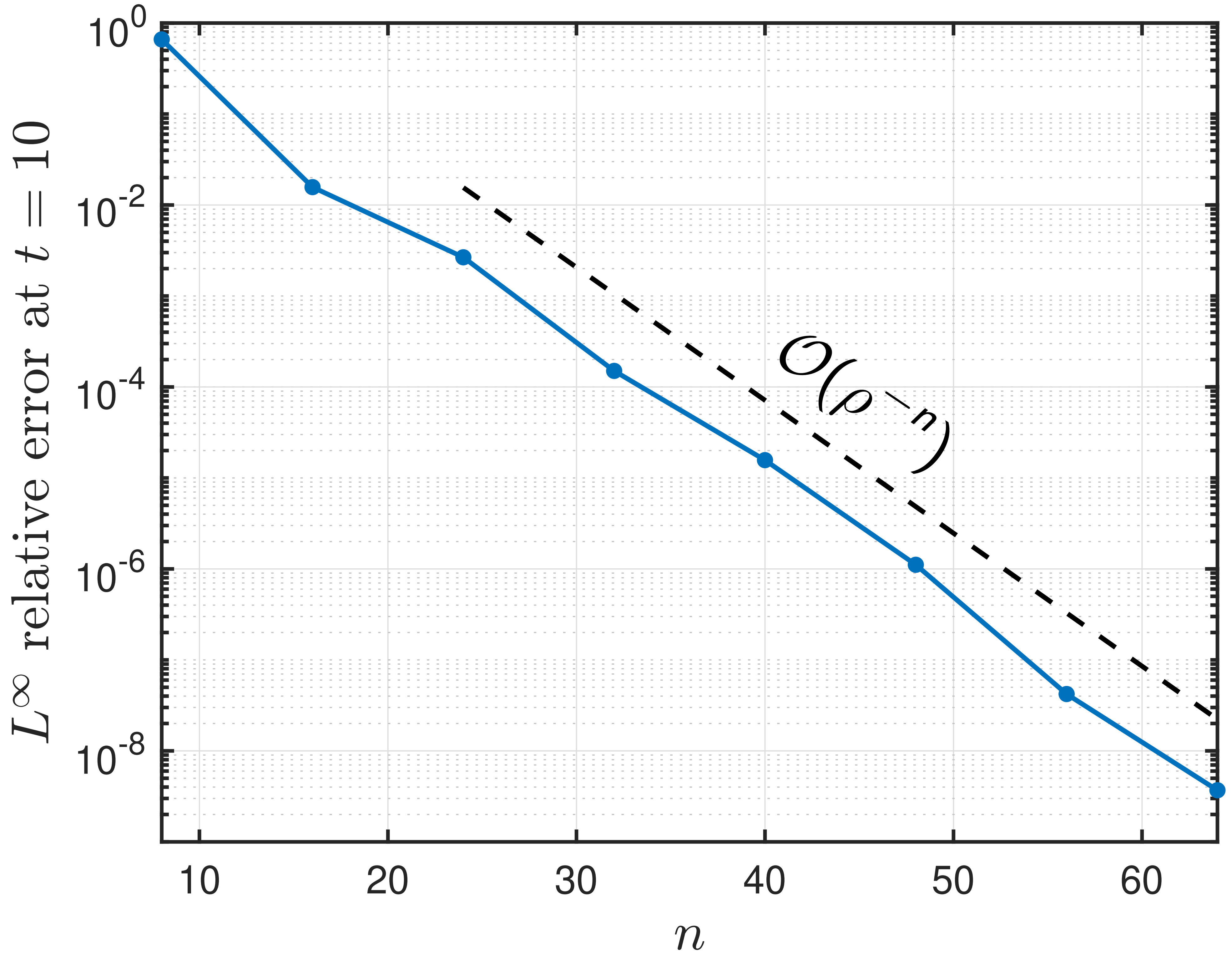
$$x(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\cos\xi$$
$$y(\xi,\eta) = \rho(\eta)\sin\theta(\eta)\sin\xi$$
$$z(\xi,\eta) = \rho(\eta)\cos\theta(\eta)$$

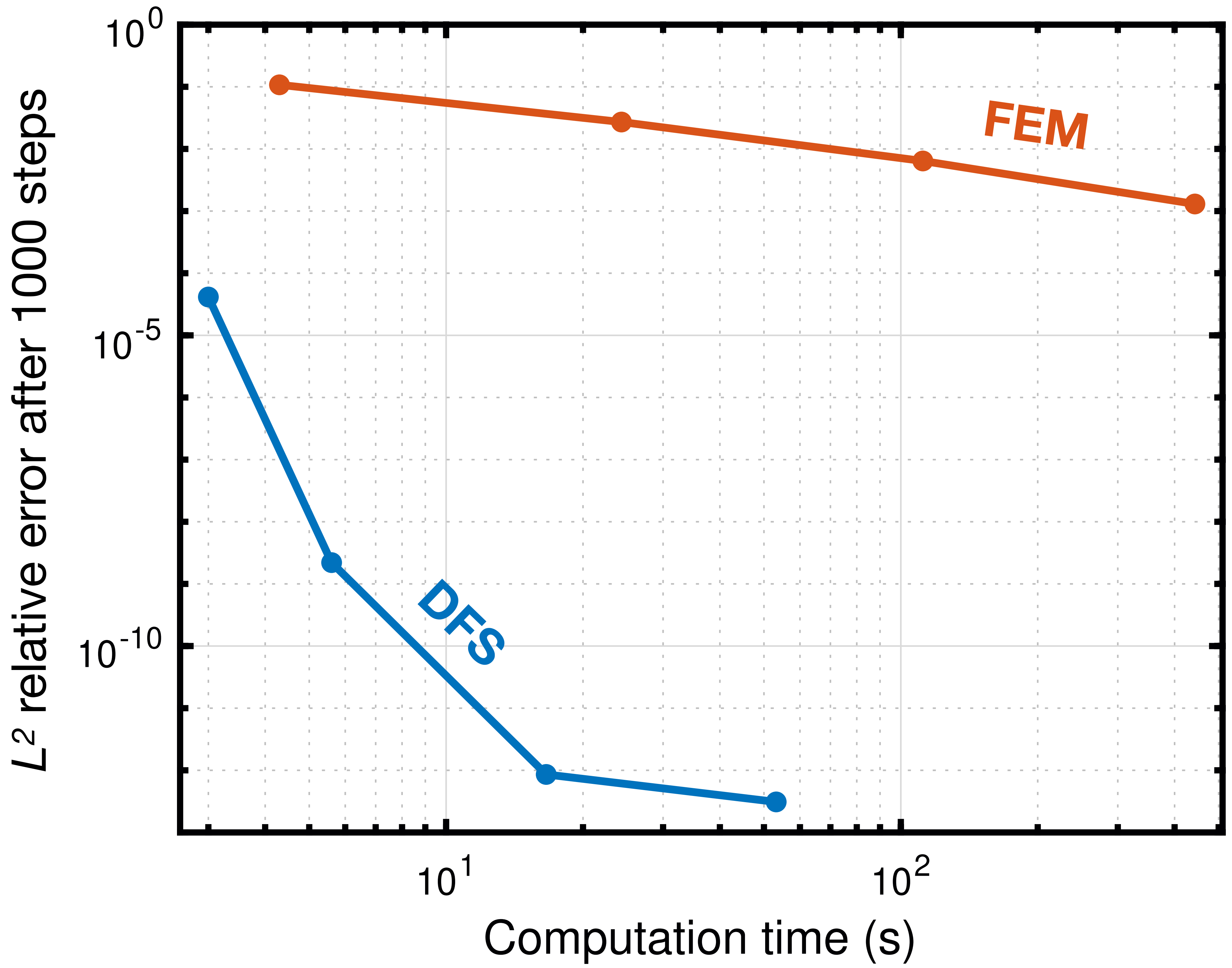
For smooth surfaces, the DFS-accelerated IMEX scheme obtains spectral accuracy, i.e., exponential convergence in space.



DFS on axisymmetric surfaces
Numerical results

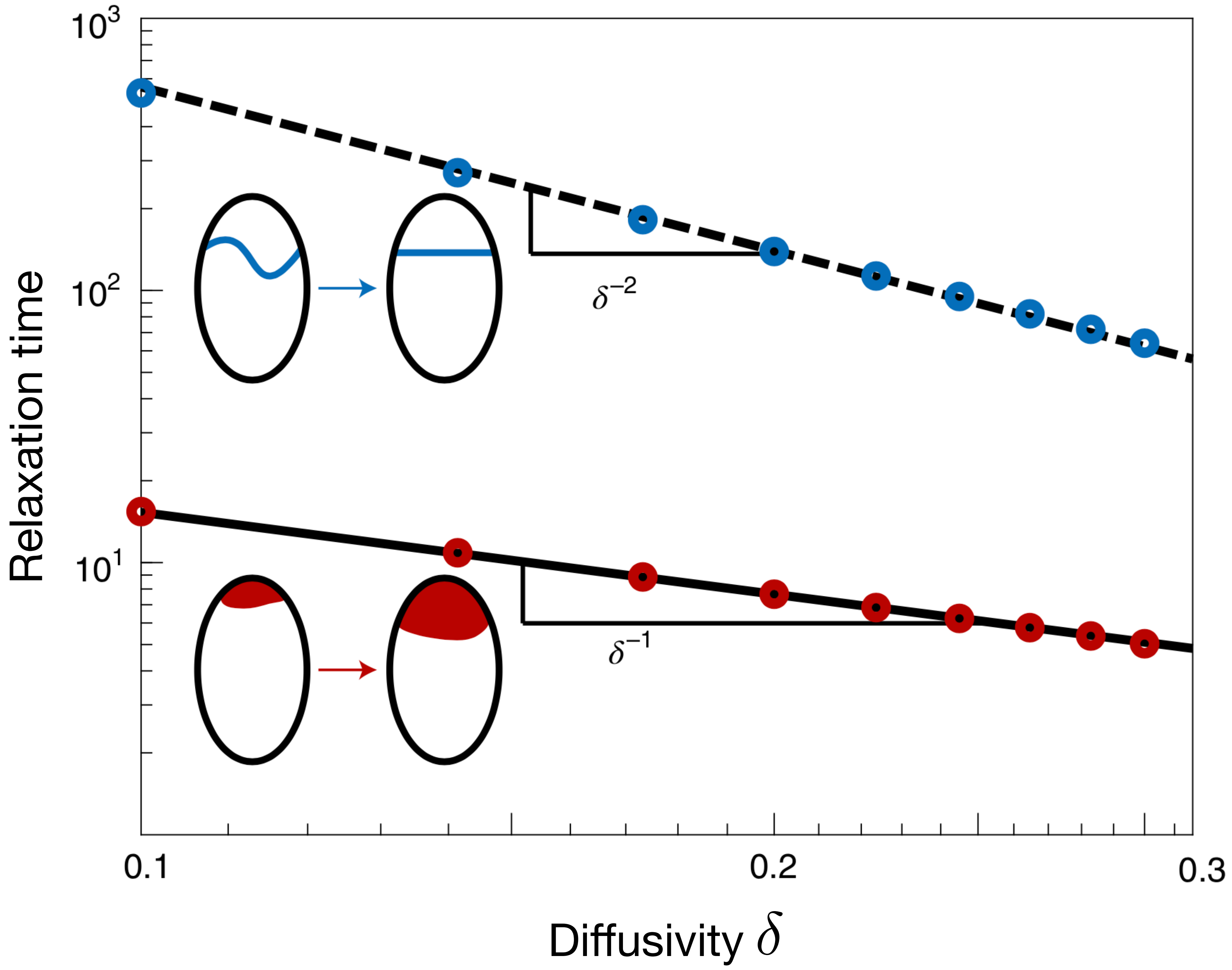
For smooth surfaces, the DFS-accelerated IMEX scheme obtains spectral accuracy, i.e., exponential convergence in space… and provides a significant speedup over FEM.



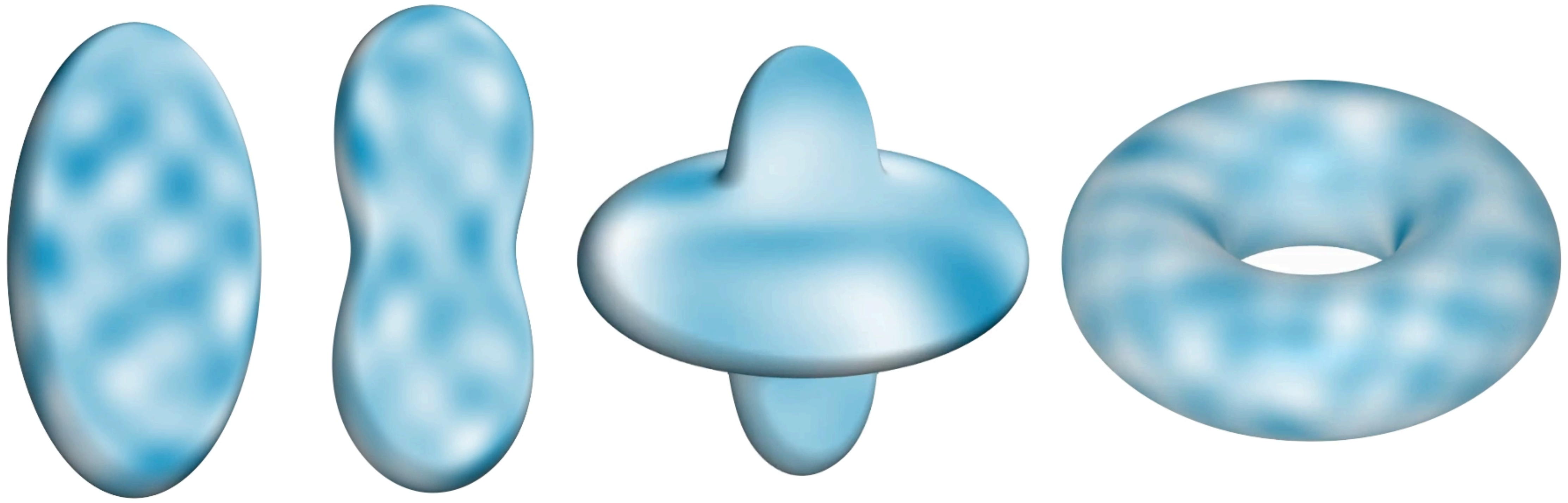


Cell polarization
A hierarchy of timescales

Surface curvature can drive cell polarization. Fast solvers allowed us to explore how symmetry is broken over a wide range of geometries and characterize three distinct timescales.



$\mathcal{O}(1)$	Local reaction kinetics
$\mathcal{O}(\delta^{-1})$	Formation of cap with fixed area
$\mathcal{O}(\delta^{-2})$	Interface minimization via geodesic curvature flow

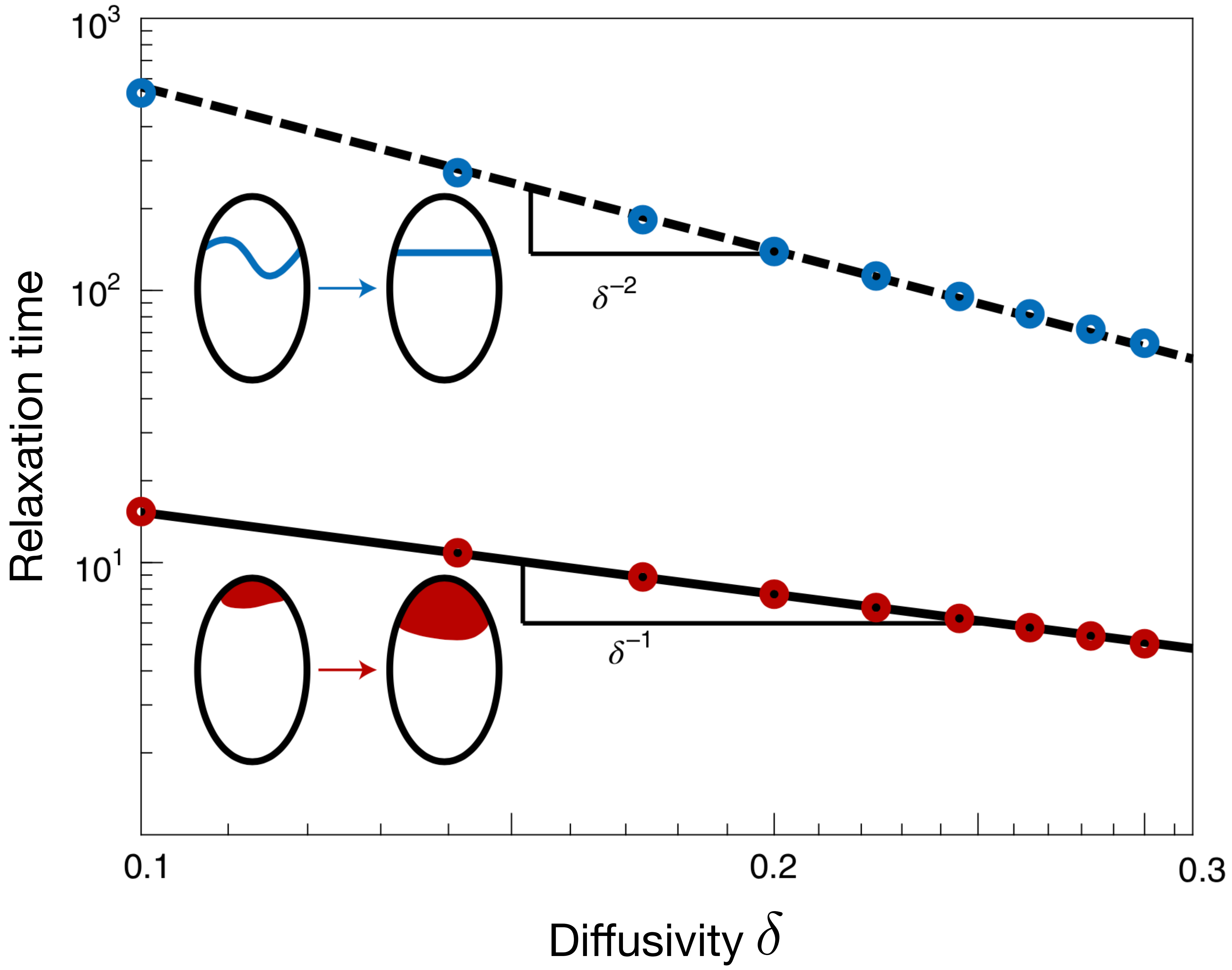


Pearson Miller Cyrill Muratov Leslie Greengard Stas Shvartsman

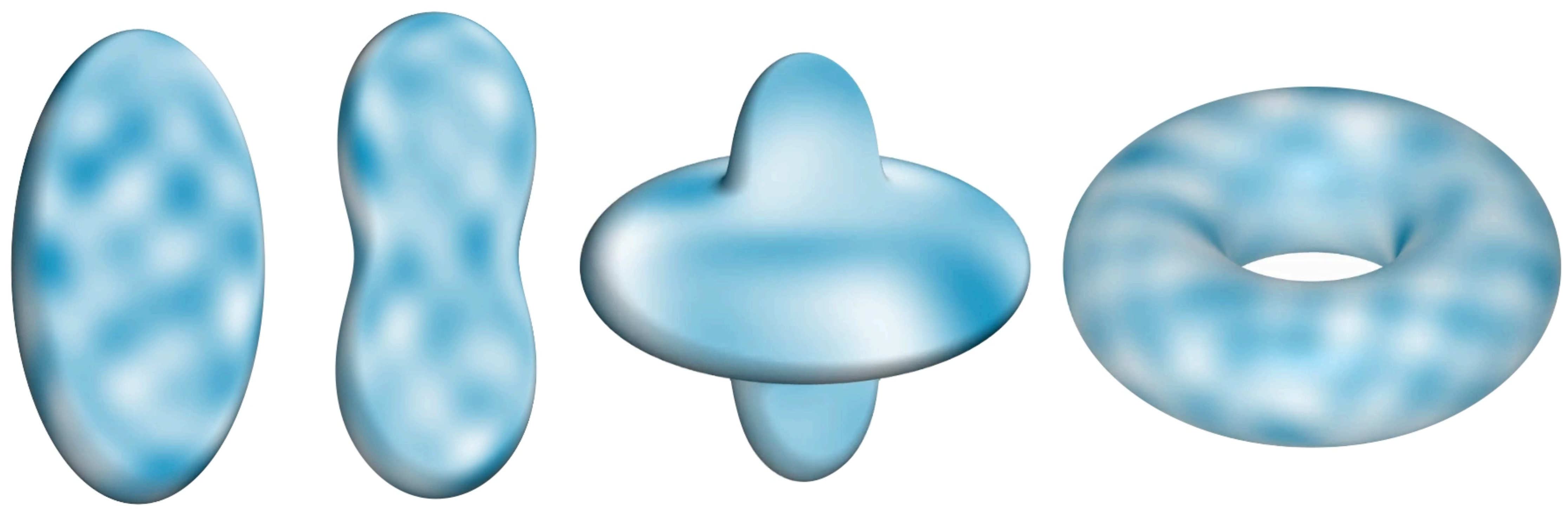
[Miller*, F.*, Muratov, Greengard, & Shvartsman, 2022]
[Miller, F., Novaga, Shvartsman, & Muratov, 2023]

Cell polarization
A hierarchy of timescales

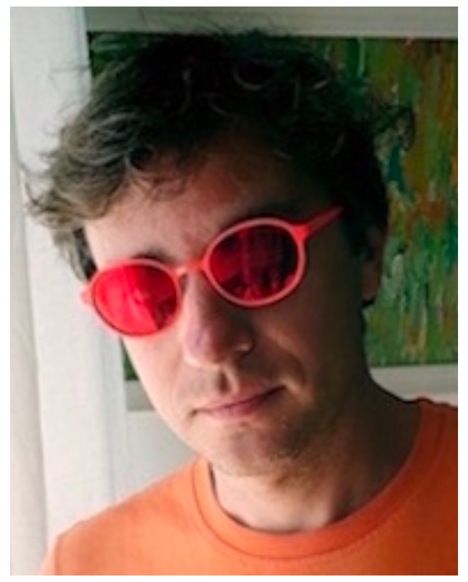
Surface curvature can drive cell polarization. Fast solvers allowed us to explore how symmetry is broken over a wide range of geometries and characterize three distinct timescales.



$\mathcal{O}(1)$	Local reaction kinetics
$\mathcal{O}(\delta^{-1})$	Formation of cap with fixed area
$\mathcal{O}(\delta^{-2})$	Interface minimization via geodesic curvature flow



Pearson Miller



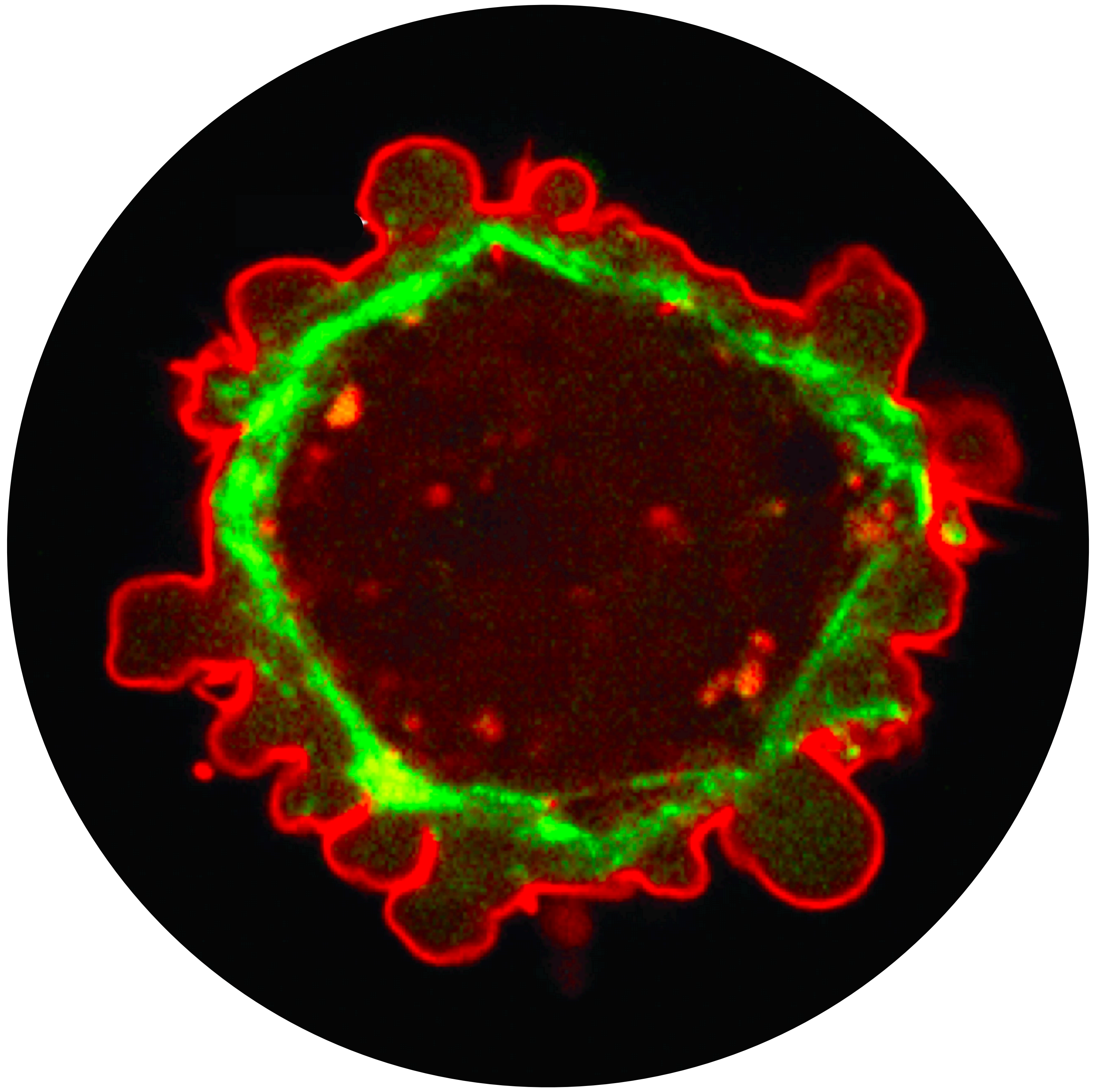
Cyrill Muratov

Leslie Greengard

Stas Shvartsman

[Miller*, F.*, Muratov, Greengard, & Shvartsman, 2022]
[Miller, F., Novaga, Shvartsman, & Muratov, 2023]

What about reaction-diffusion processes on more complicated surfaces?



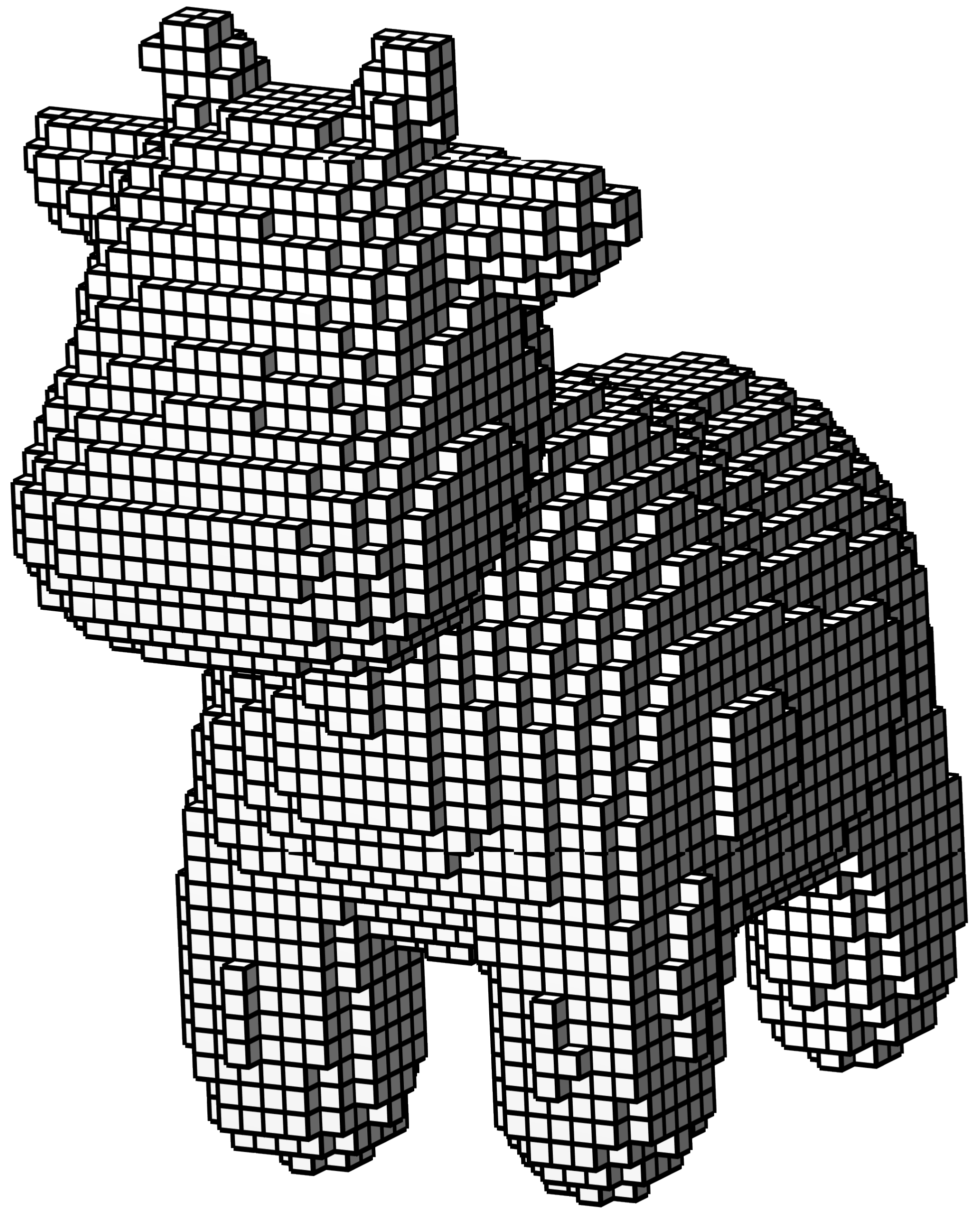
Is high-order accuracy still possible?

[Charras, 2008]

Surface representation

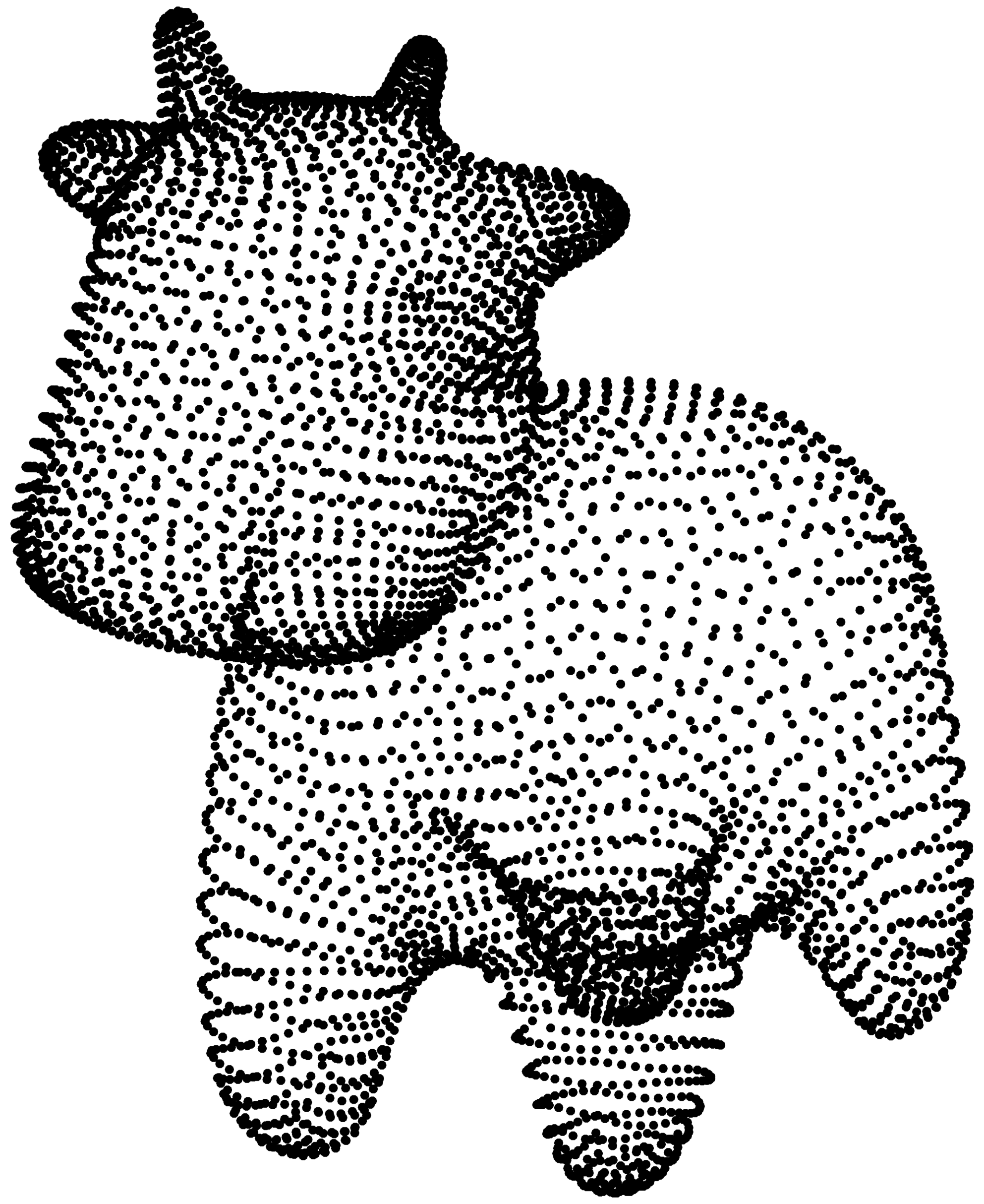
Many ways to represent a surface

Embedded grid



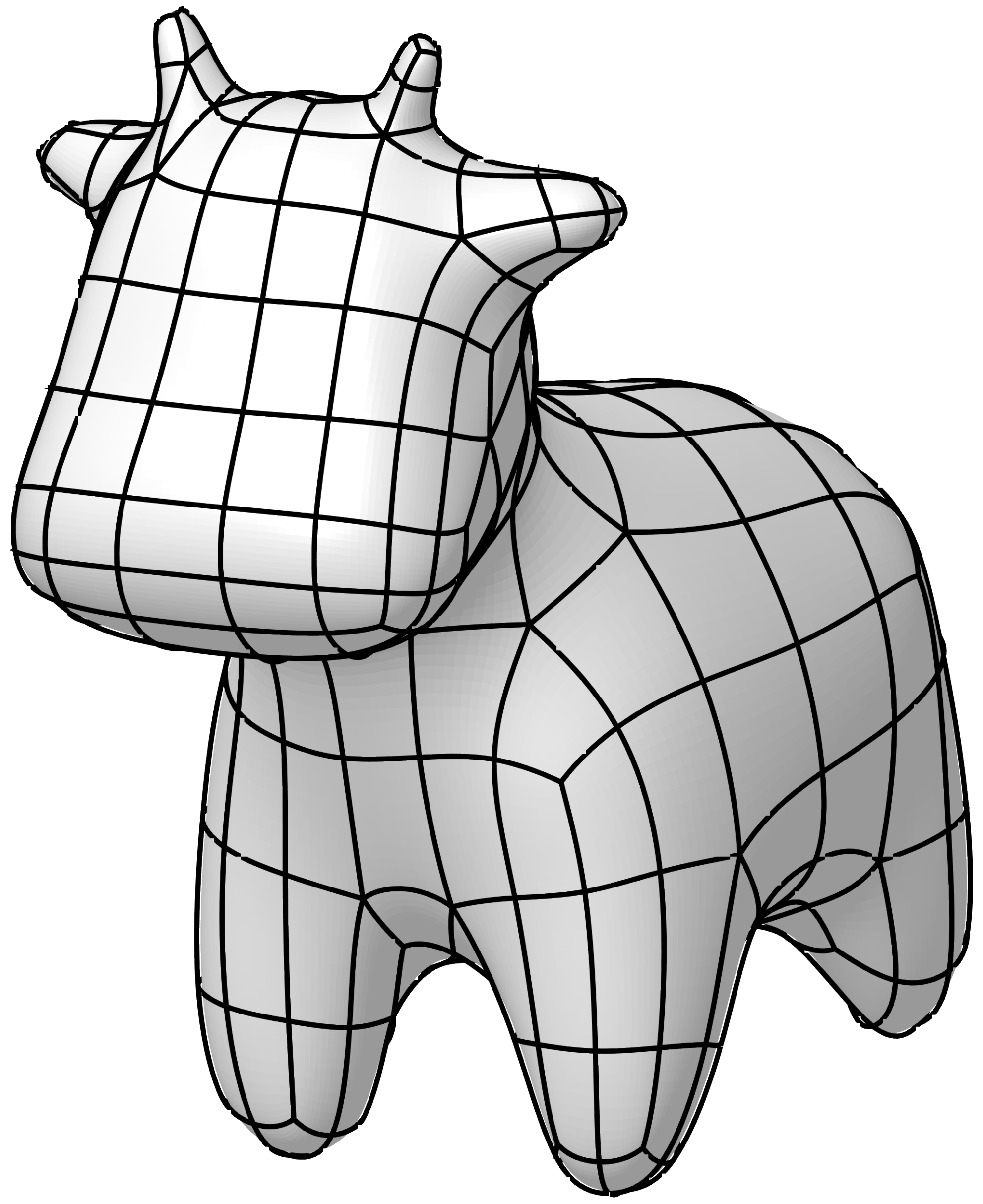
- Closest point methods
  (Macdonald, Greer, Ruuth, …)
- Level set methods
  (Sethian, Osher, …)

Unstructured point cloud



- Radial basis functions
  (Fornberg, Piret, Wendland, Wright, …)

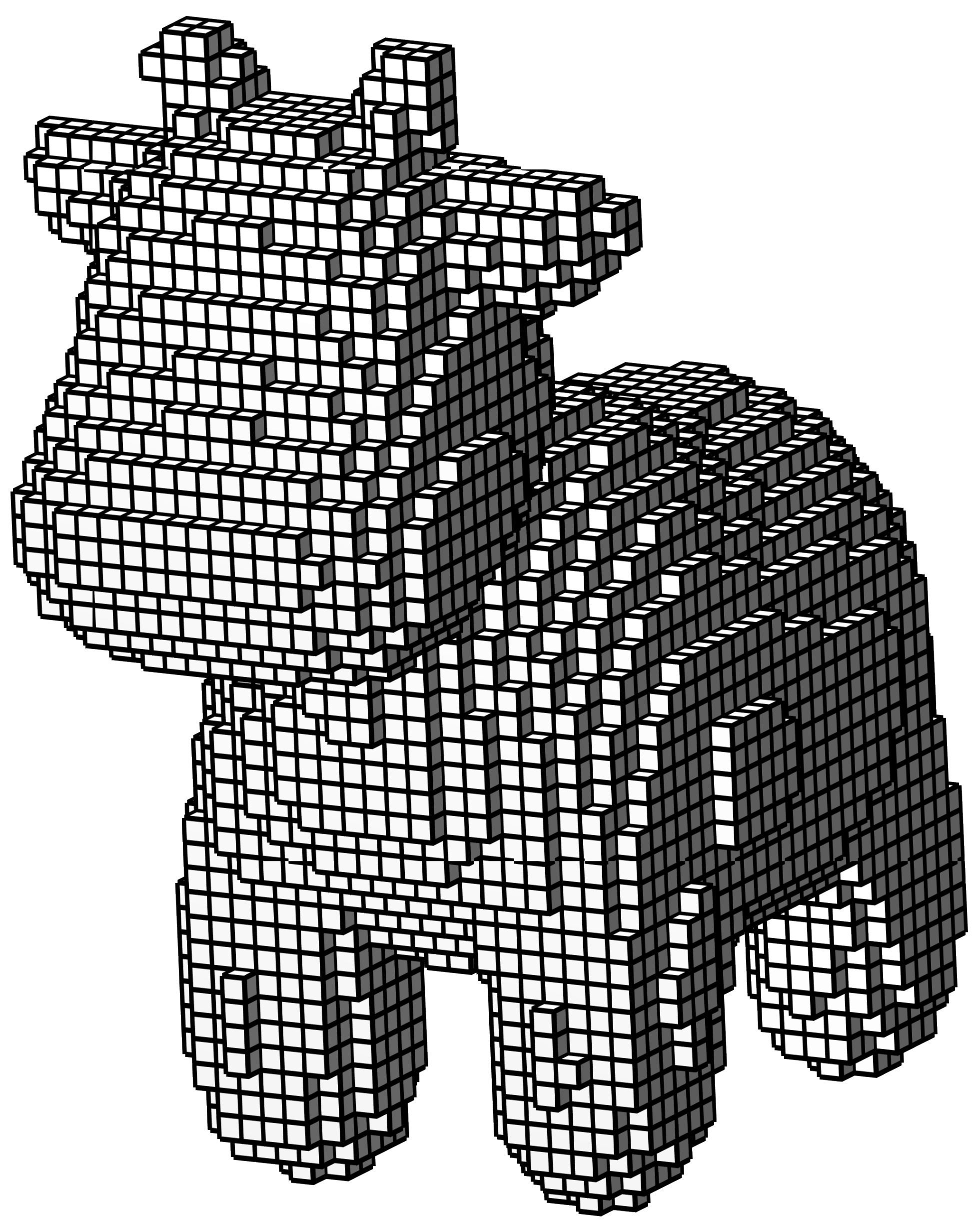
Surface mesh



- Finite element methods
  (Dziuk, Elliott, …)
- Integral equation methods
  (O'Neil, Goodwill, Rachh, …)

Surface representation
Many ways to represent a surface

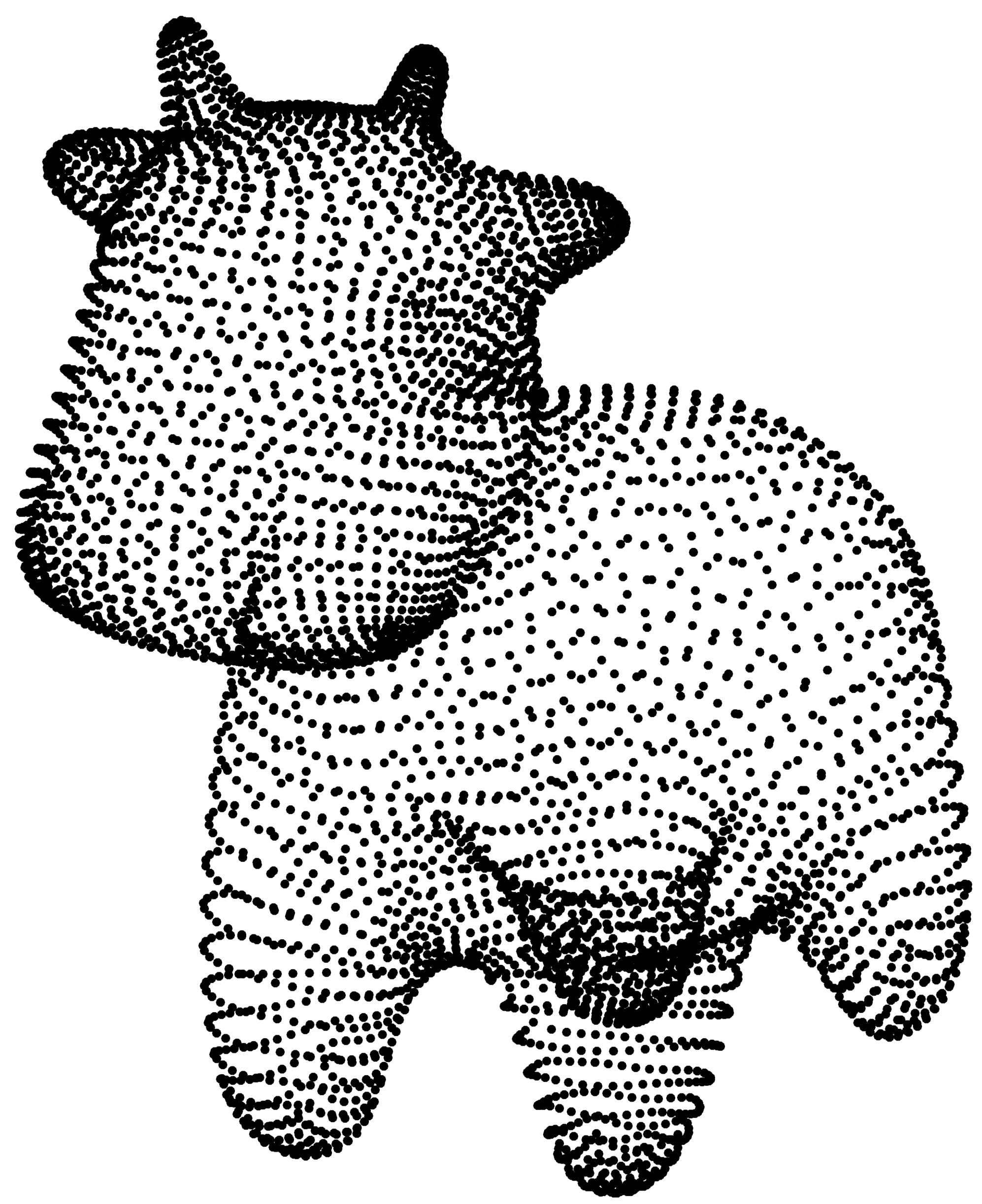
Embedded grid



- Closest point methods
  (Macdonald, Greer, Ruuth, …)
- Level set methods
  (Sethian, Osher, …)

high order → more volume DoFs

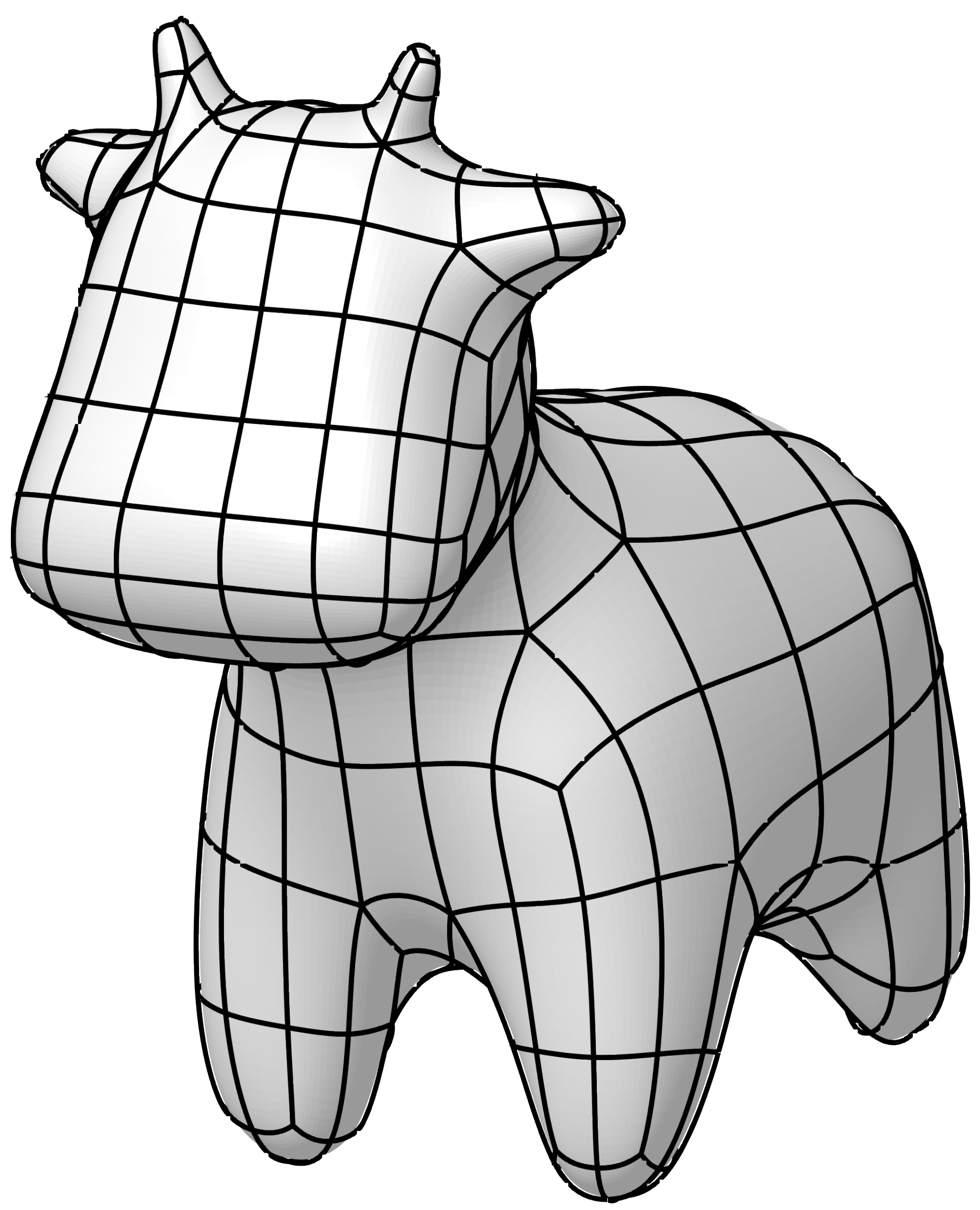
Unstructured point cloud



- Radial basis functions
  (Fornberg, Piret, Wendland, Wright, …)

high order → ill-conditioned

Surface mesh

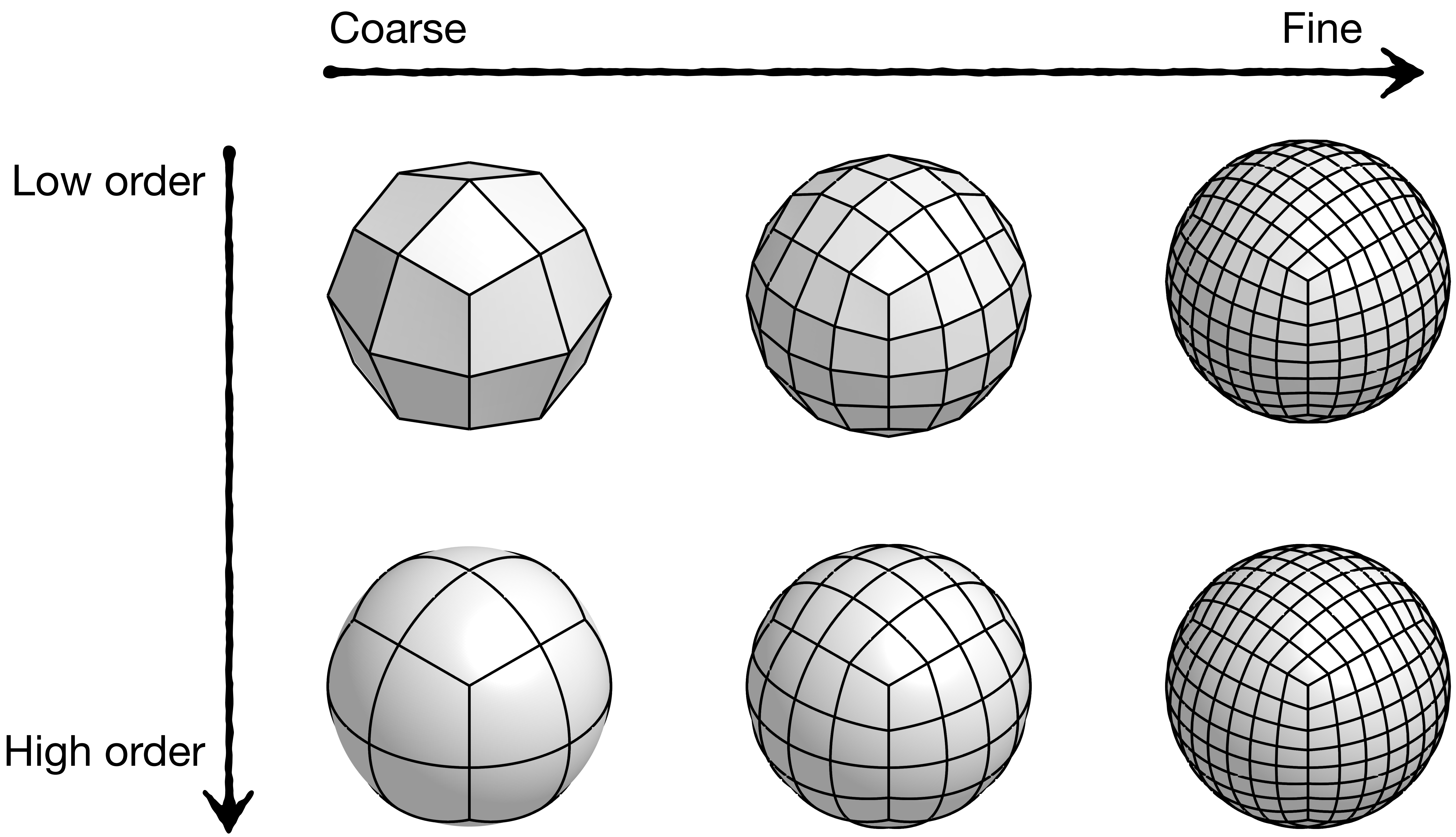


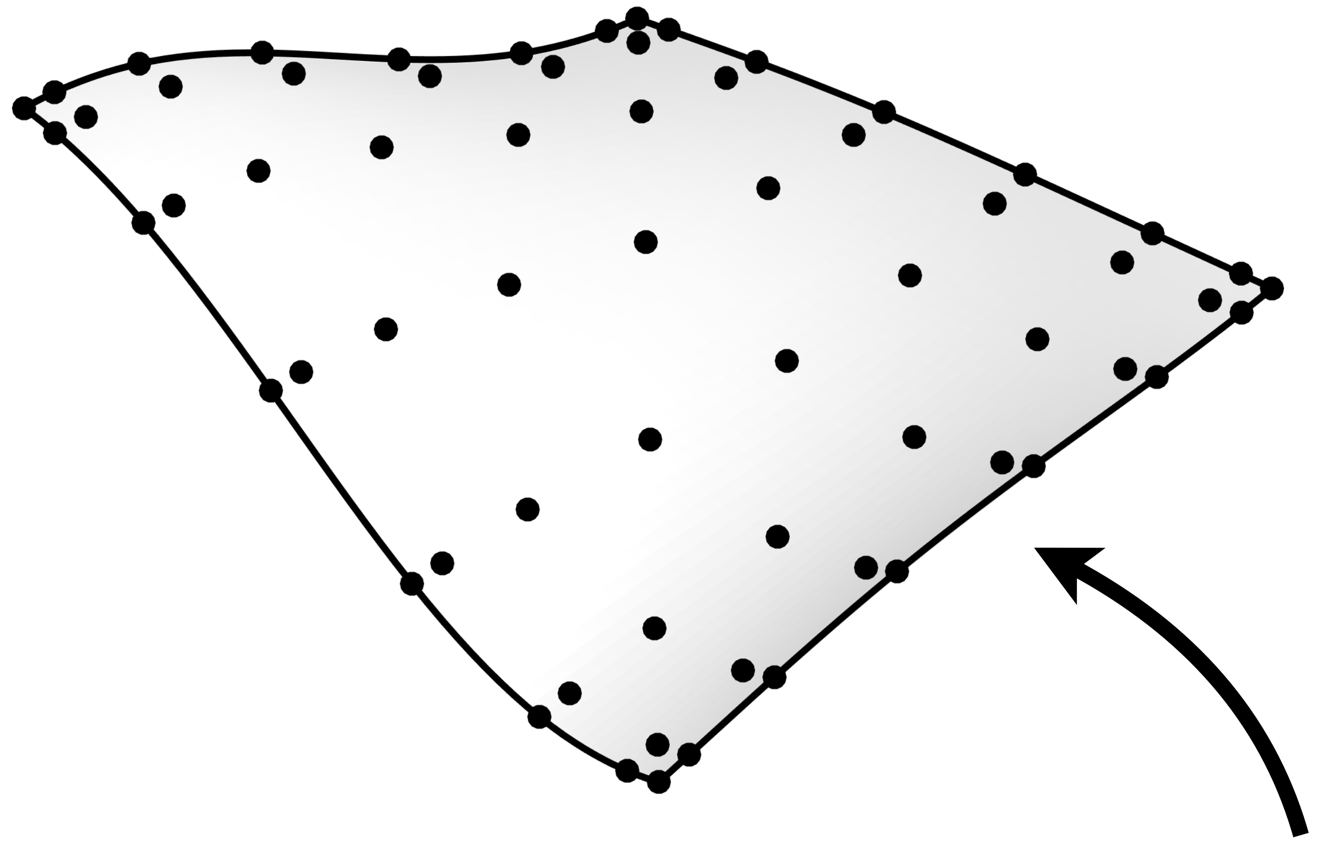
- Finite element methods
  (Dziuk, Elliott, …)
- Integral equation methods
  (O'Neil, Goodwill, Rachh, …)

high order → dense fill-in

Surface representation
Low-order vs. high-order

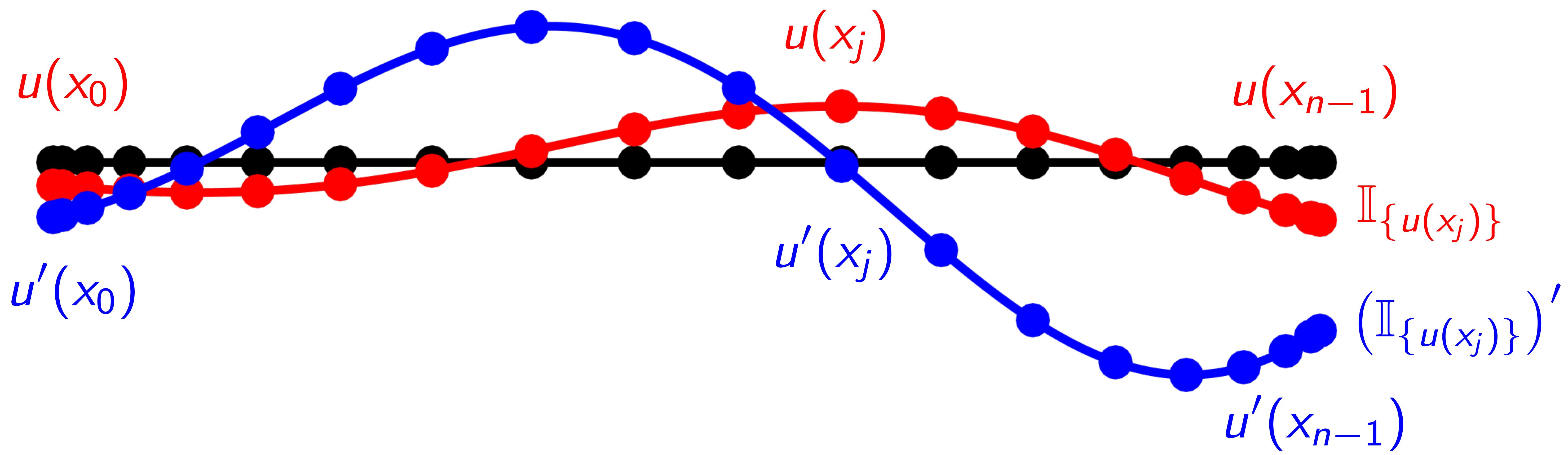
- Meshes are a good choice for CAD-compatibility. We use high-order quadrilateral patches.
- High-order elements allow faster convergence to solution.
- Coordinate maps of a patch are discretized via tabulation at Chebyshev nodes.





High-order discretization
Spectral collocation

- Function values also stored at Chebyshev nodes.
- Derivatives and metric information (e.g. Jacobian) computed via spectral differentiation.

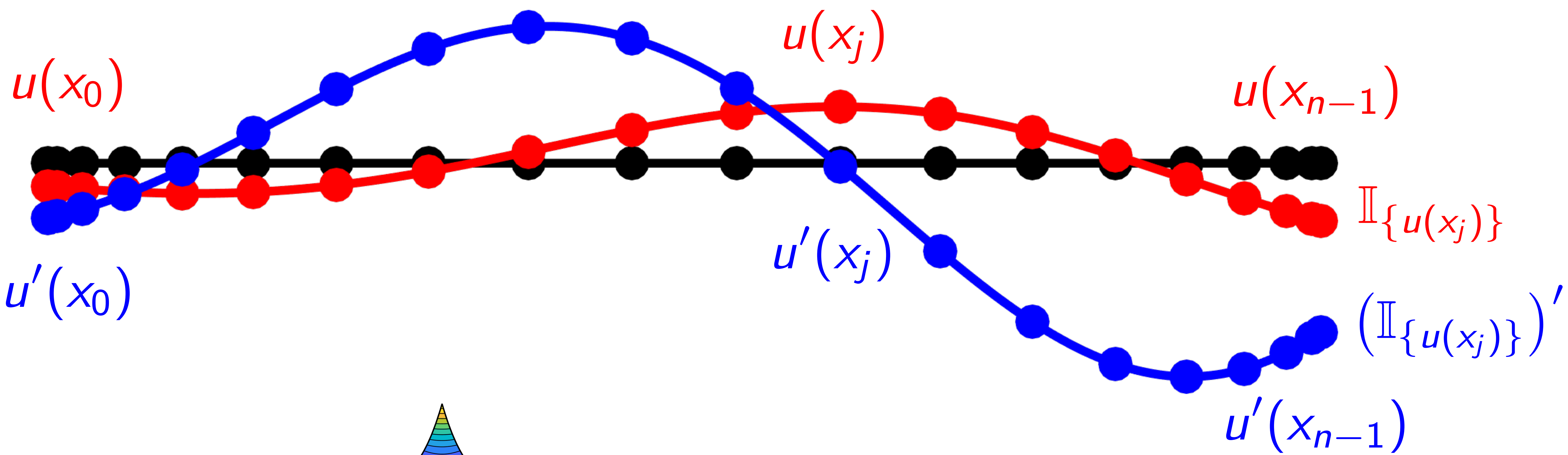


$$D\,u_j \mapsto u_j'$$

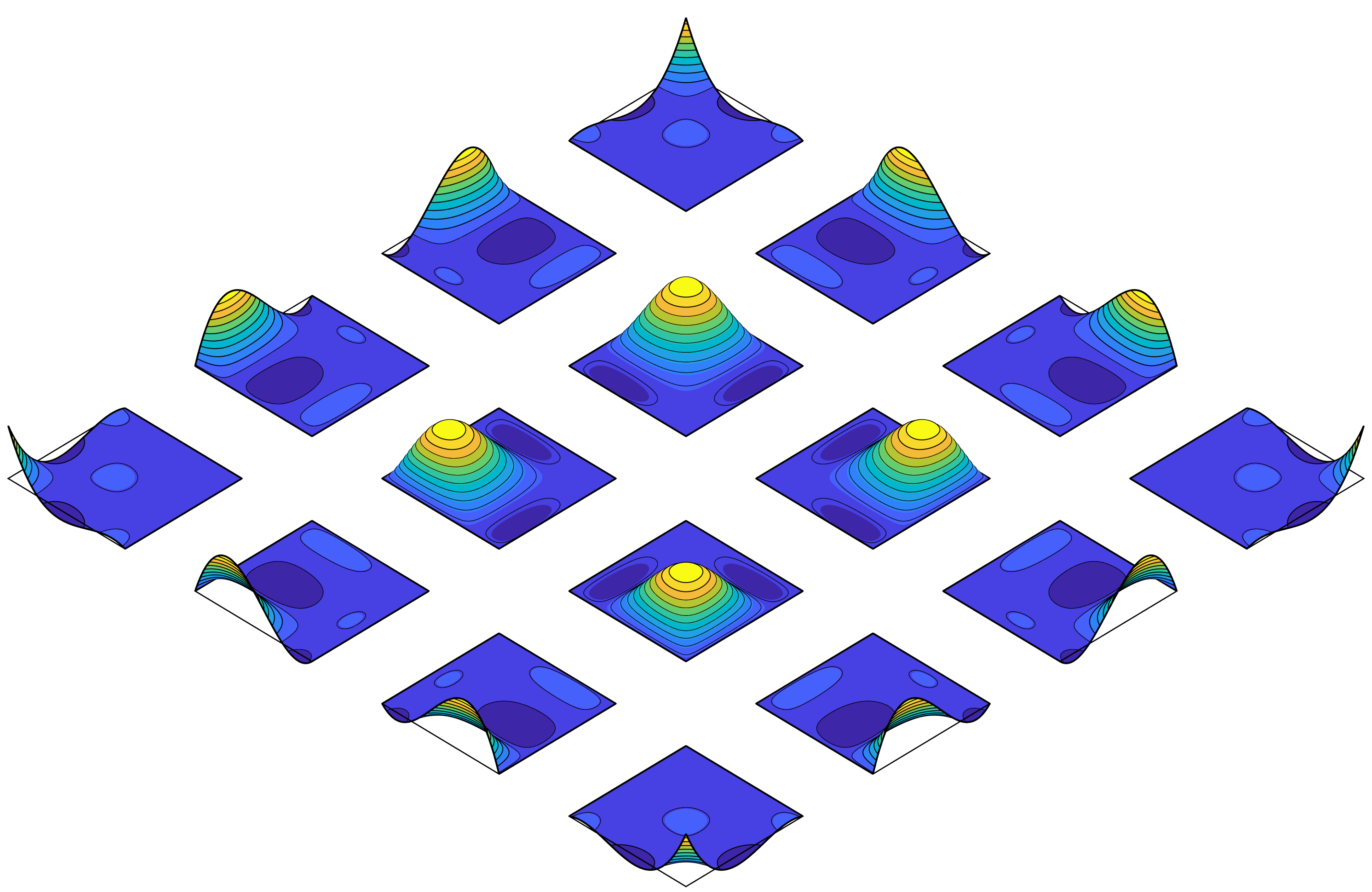
[Trefethen, 2000]
[Trefethen, 2013]

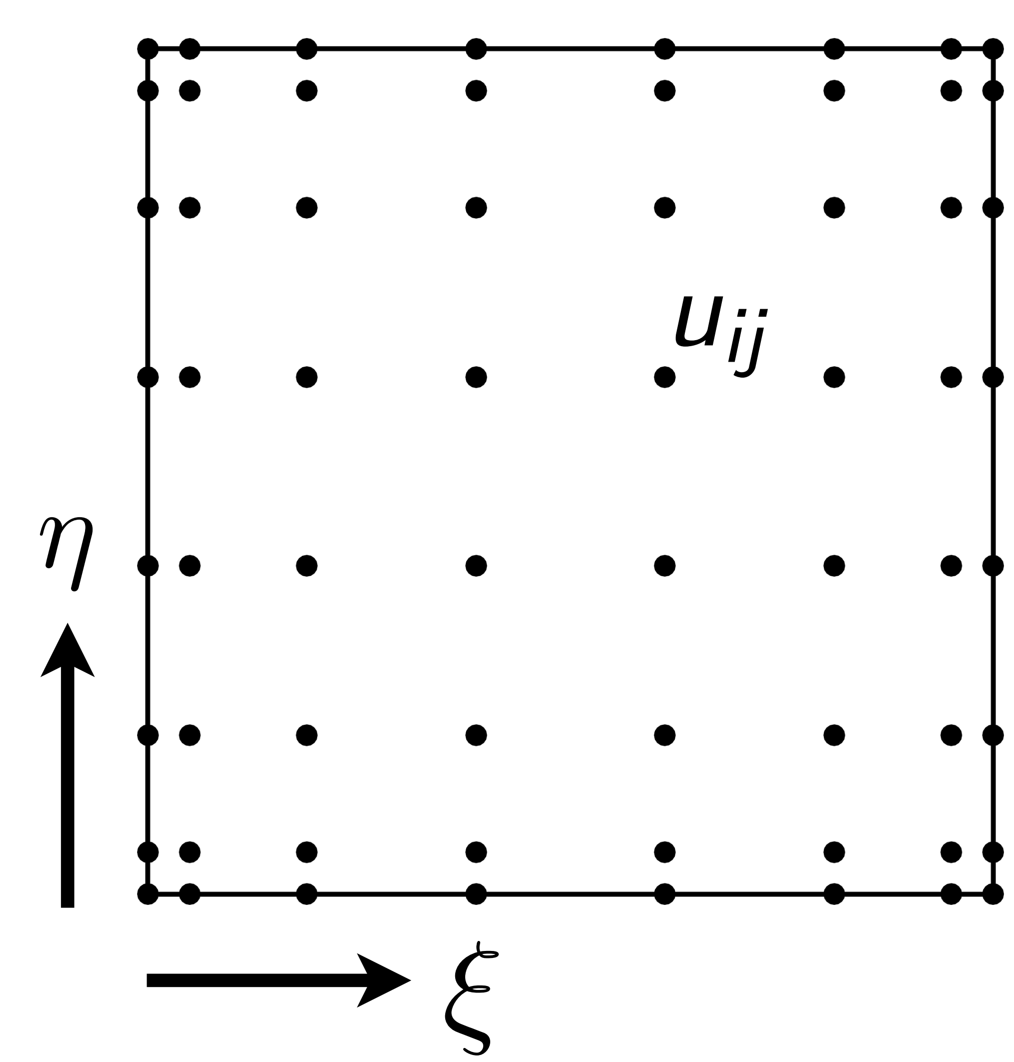
High-order discretization
Spectral collocation

- Function values also stored at Chebyshev nodes.
- Derivatives and metric information (e.g. Jacobian) computed via spectral differentiation.



$$D\mathbf{u}_j \mapsto \mathbf{u}'_j$$





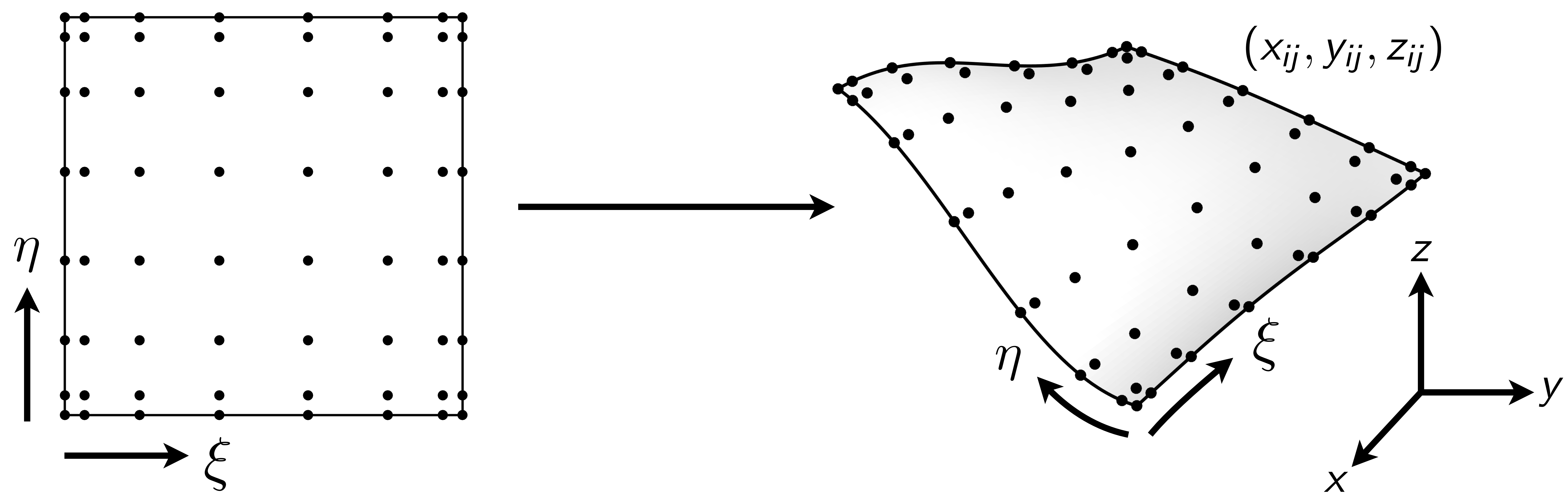
$$(D \otimes I)\mathbf{u}_{ij} \mapsto \partial \mathbf{u}_{ij}/\partial \xi$$
$$(I \otimes D)\mathbf{u}_{ij} \mapsto \partial \mathbf{u}_{ij}/\partial \eta$$

[Trefethen, 2000]
[Trefethen, 2013]

High-order discretization

Spectral collocation on a surface

- PDE is discretized through spectral differentiation and pointwise multiplication.



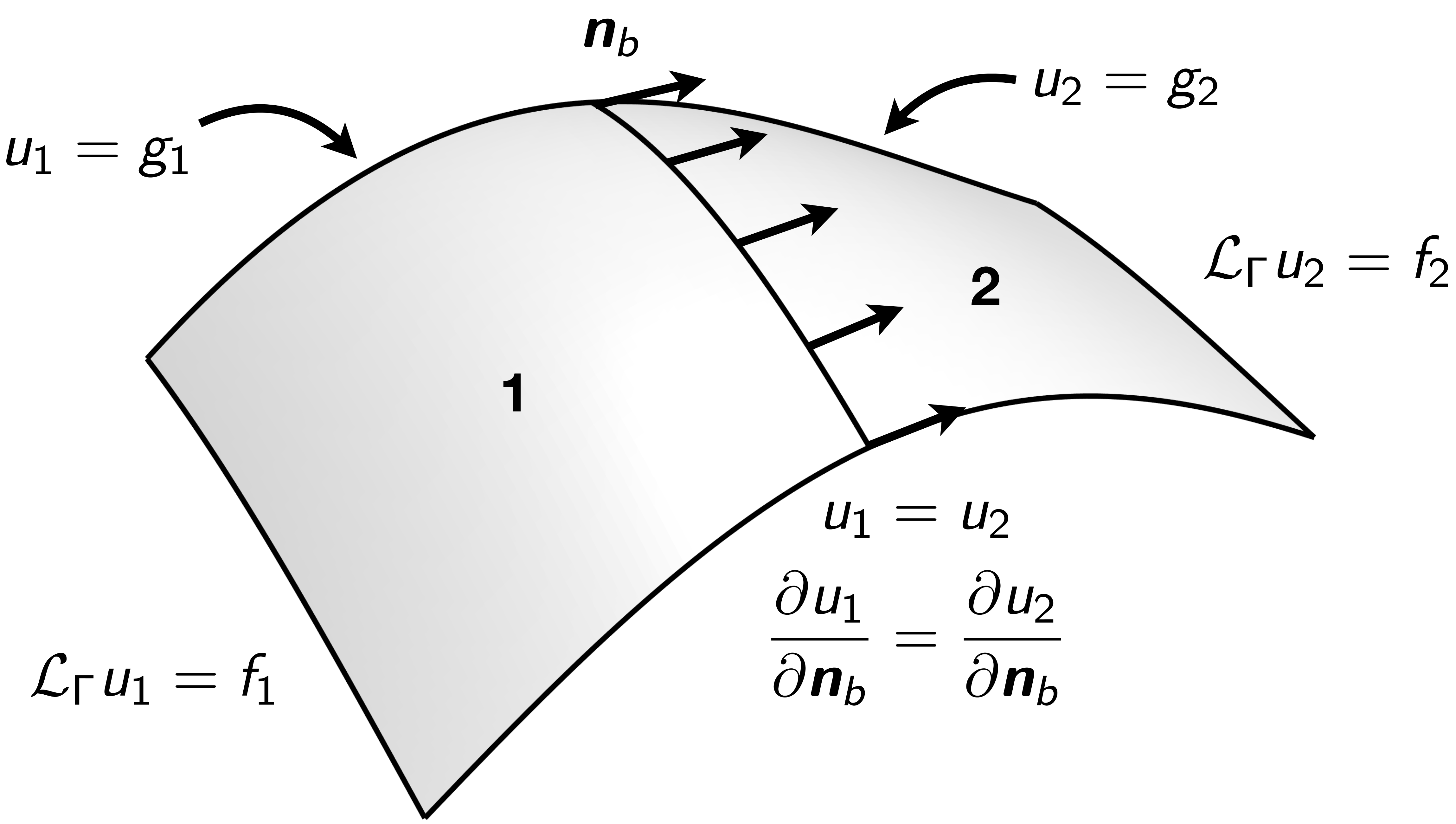
- For example, the x-component of the discrete surface gradient is:

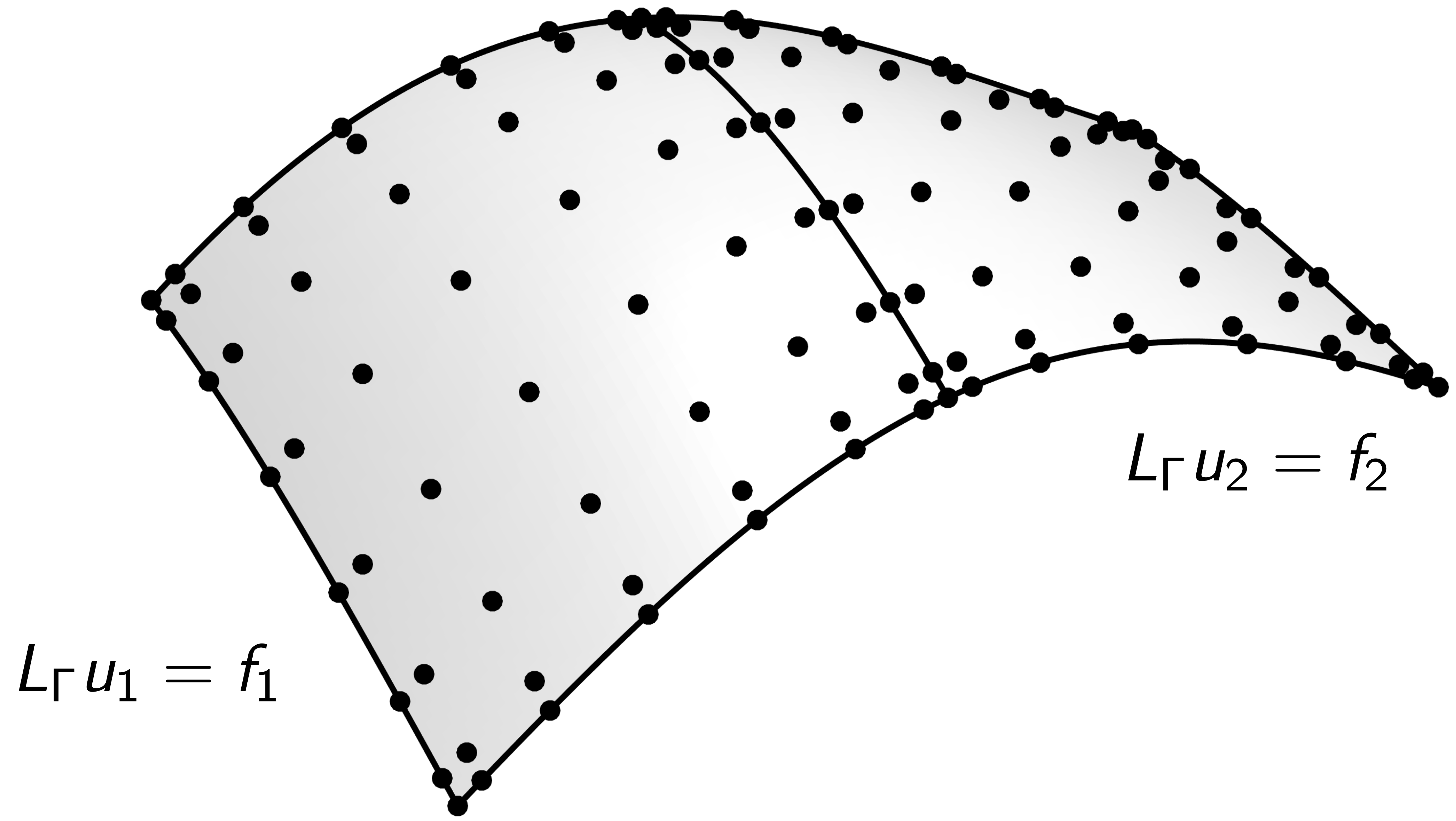
$$D_x = \begin{bmatrix} \ddots & & \\ & (\xi_x)_{ij} & \\ & & \ddots \end{bmatrix} (D \otimes I) \;+\; \begin{bmatrix} \ddots & & \\ & (\eta_x)_{ij} & \\ & & \ddots \end{bmatrix} (I \otimes D)$$

- In general, the PDE results in a $(p+1)^2 \times (p+1)^2$ linear system, $L_\Gamma u = f$, which we can invert directly.

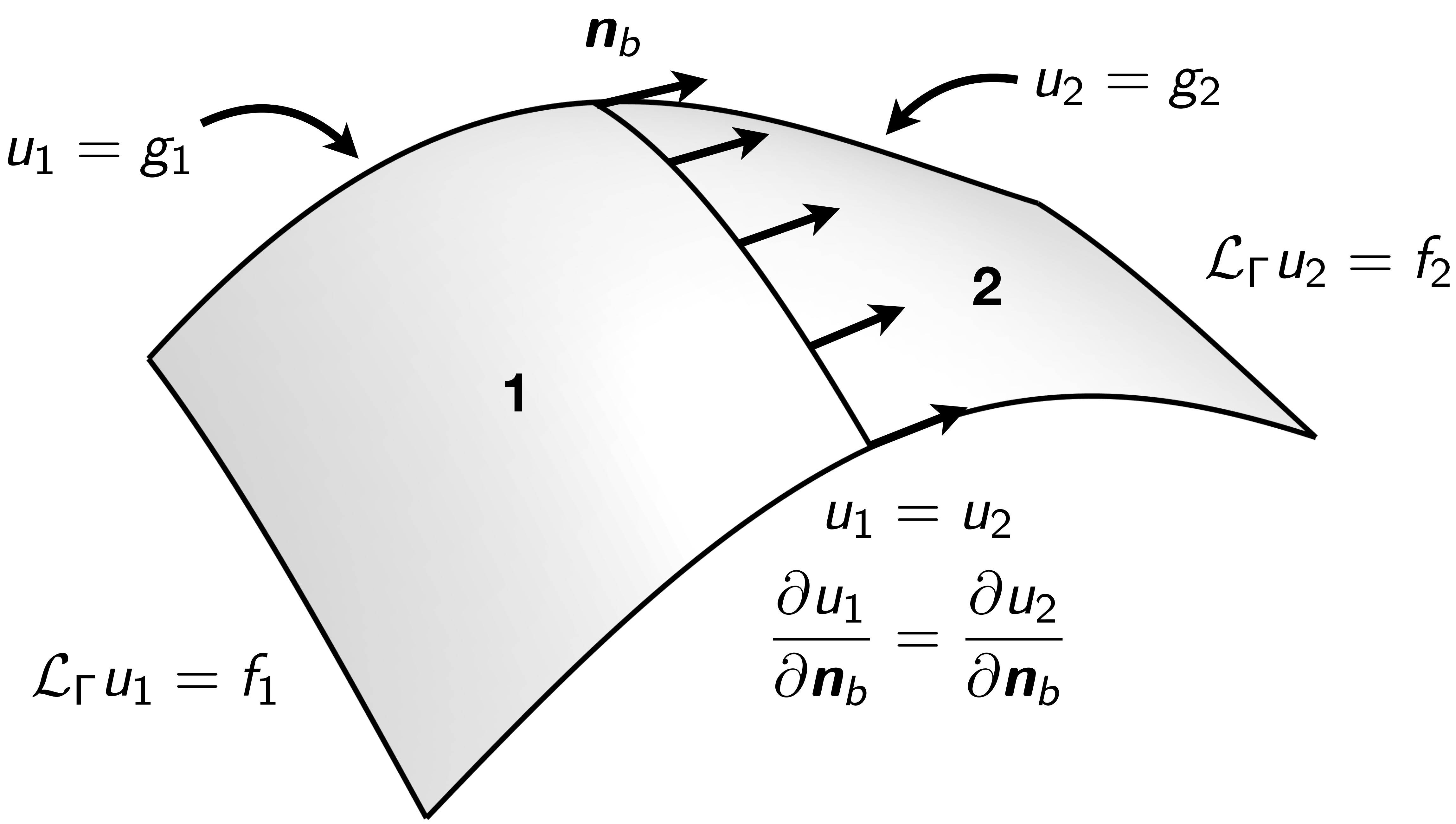
High-order discretization

Two glued patches

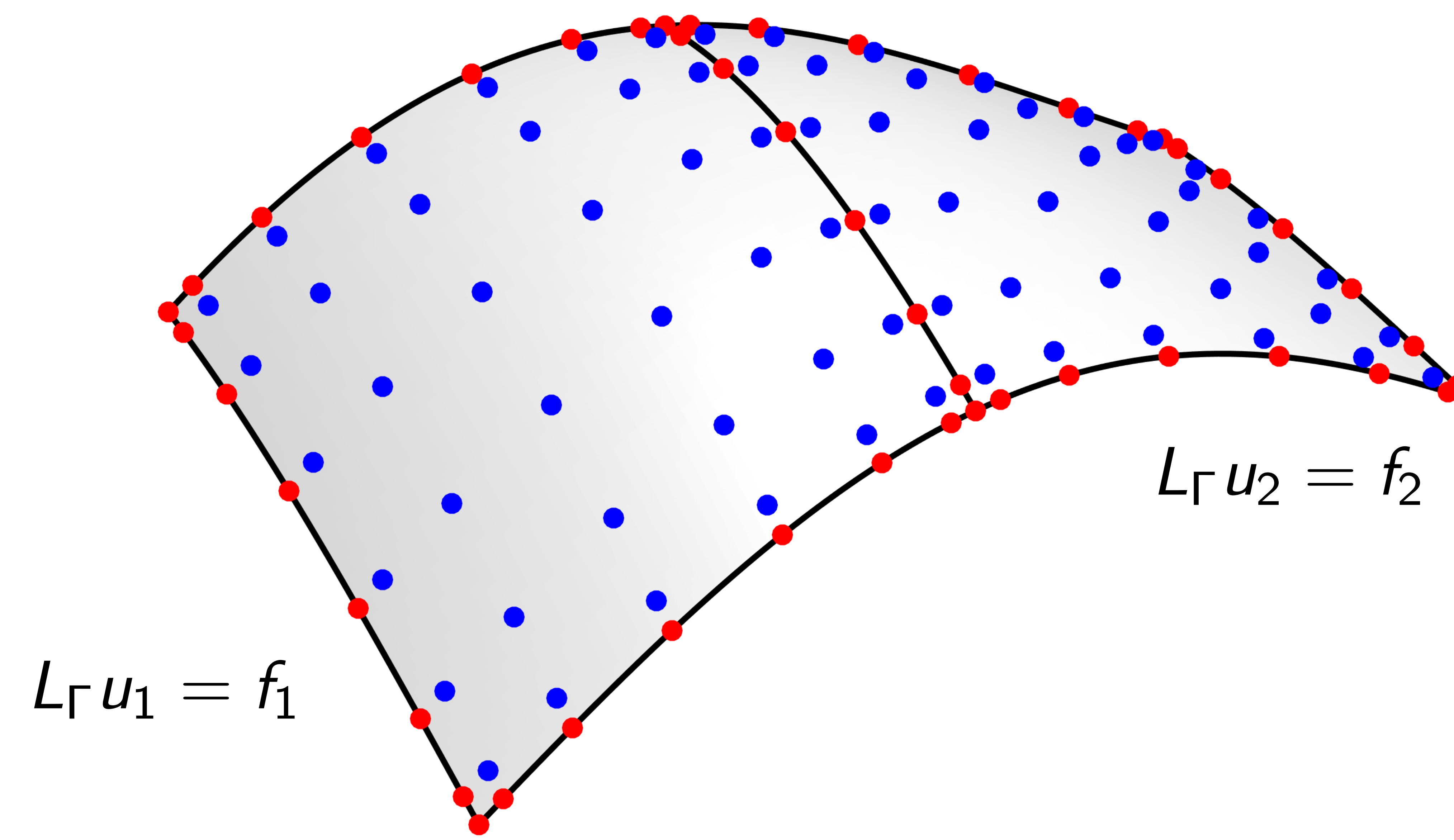




[F., 2022]

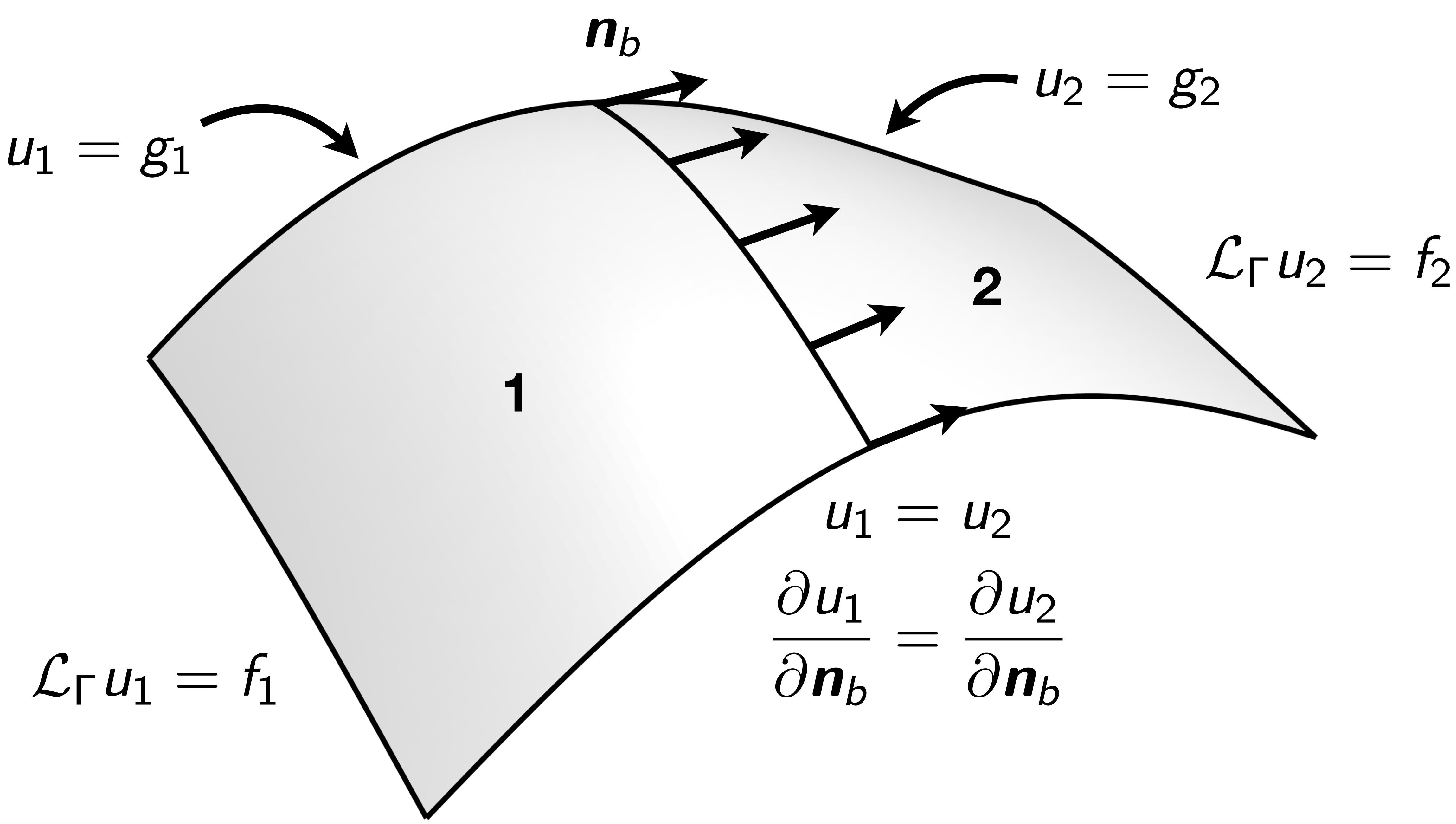


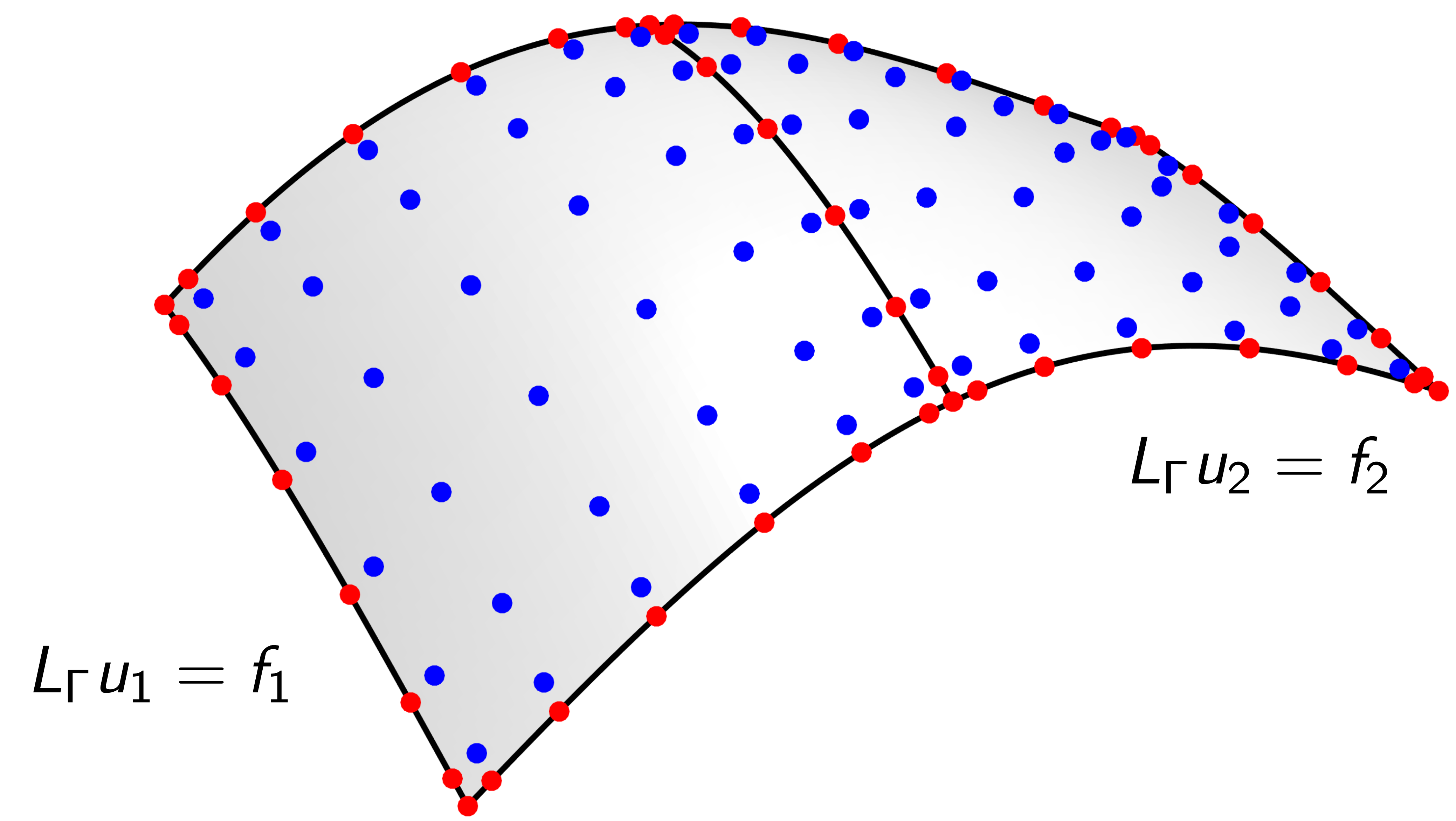
$\boldsymbol{n}_b$
$u_2 = g_2$
$u_1 = g_1$
$\mathcal{L}_\Gamma u_2 = f_2$
2
1
$u_1 = u_2$
$\dfrac{\partial u_1}{\partial \boldsymbol{n}_b} = \dfrac{\partial u_2}{\partial \boldsymbol{n}_b}$
$\mathcal{L}_\Gamma u_1 = f_1$



$L_\Gamma u_2 = f_2$
$L_\Gamma u_1 = f_1$

[F., 2022]



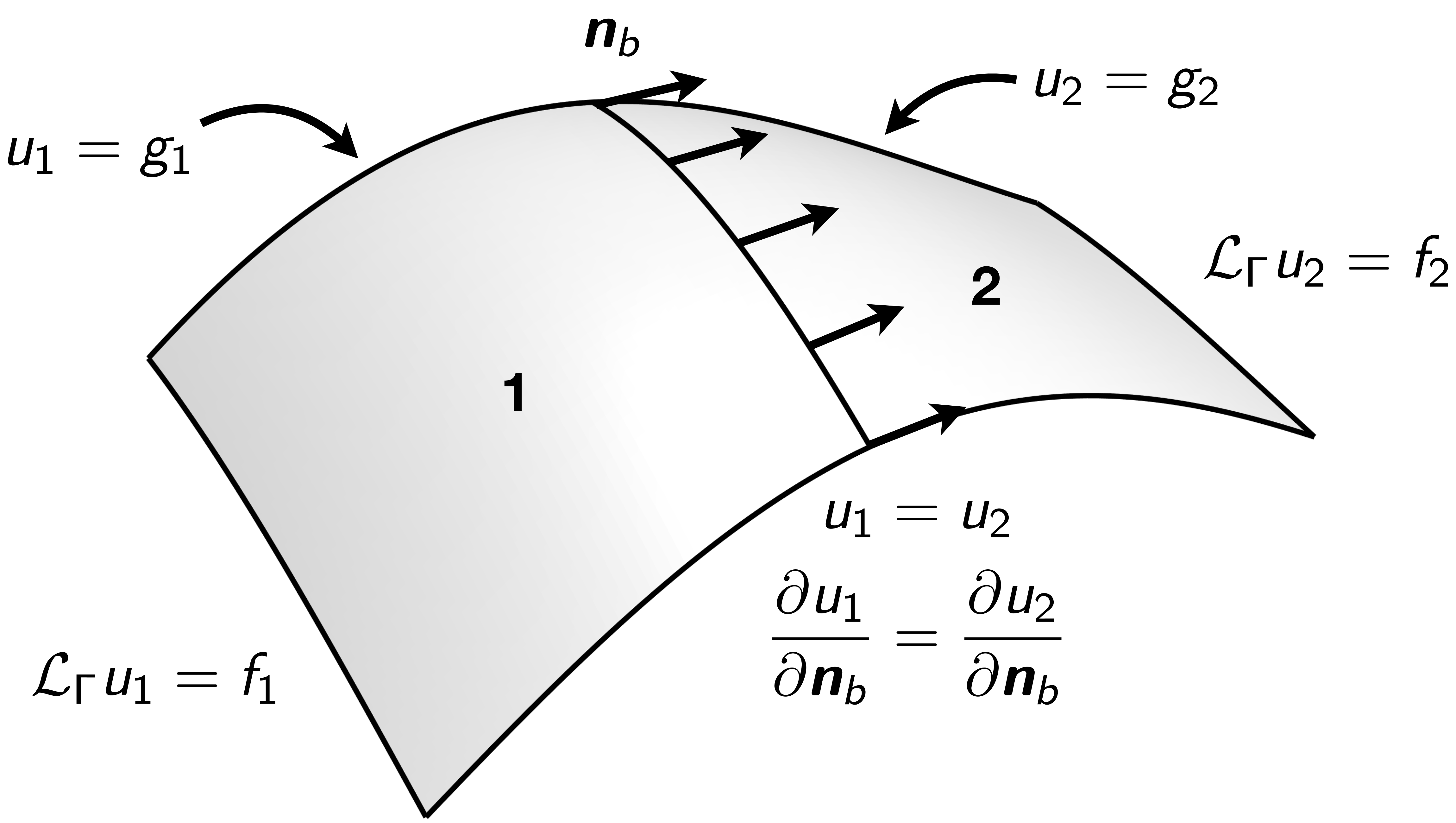


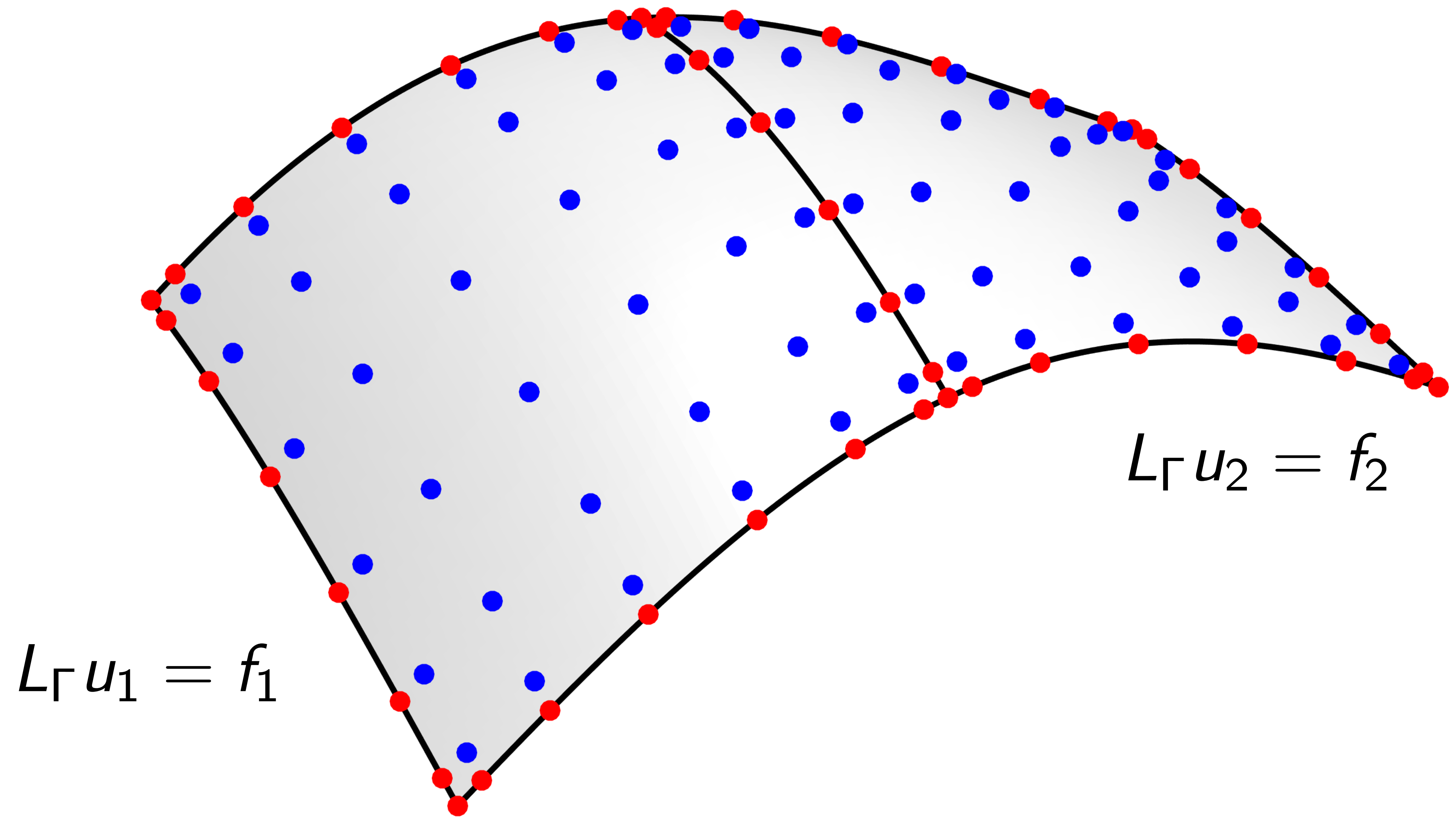
- Know how to do local solves on each element: “Solution operator” $S_1 \begin{bmatrix} g_1 \\ u_{\text{glue}} \end{bmatrix} \mapsto u_1 \qquad S_2 \begin{bmatrix} g_2 \\ u_{\text{glue}} \end{bmatrix} \mapsto u_2$

[F., 2022]

High-order discretization

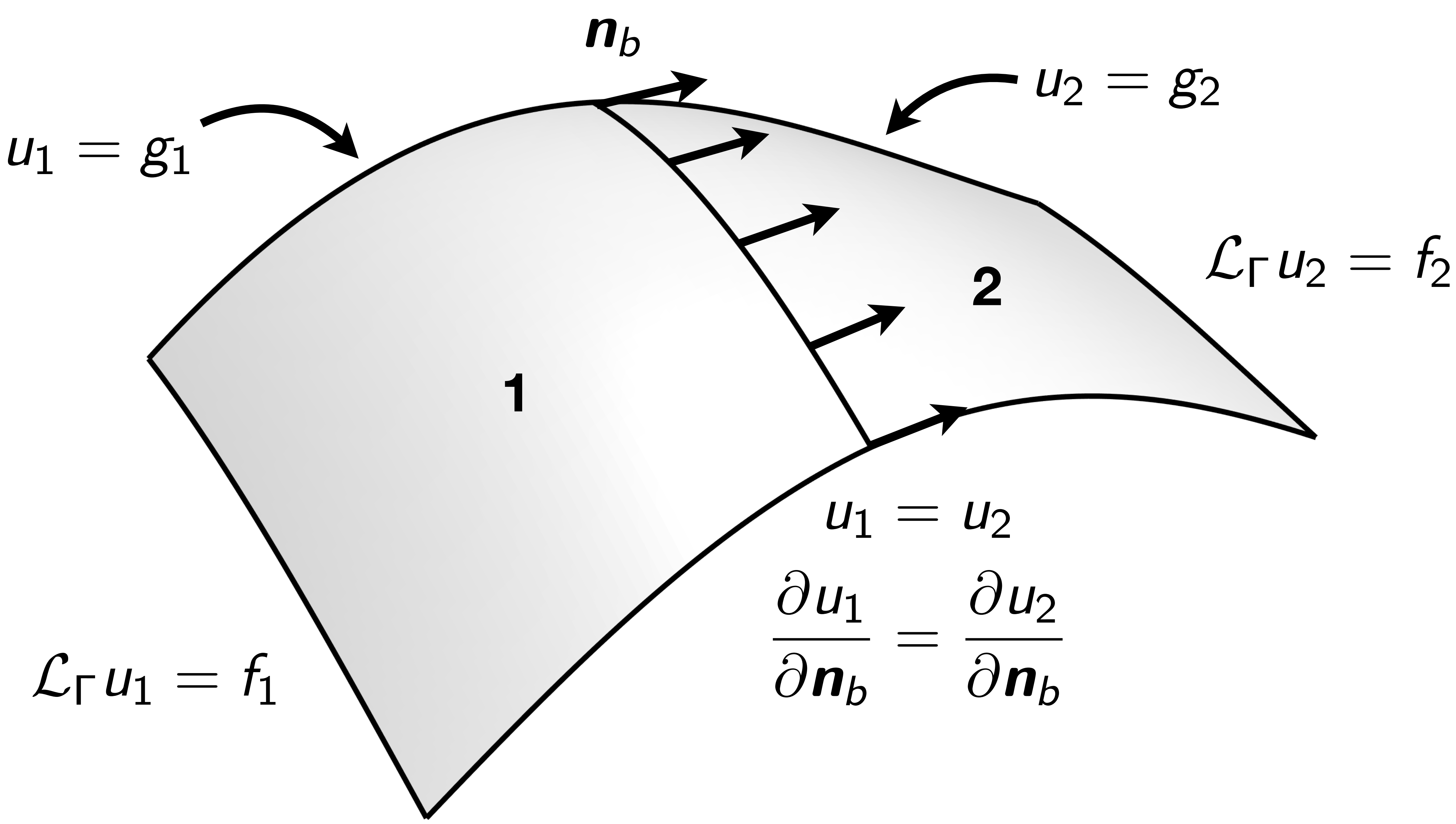
Two glued patches

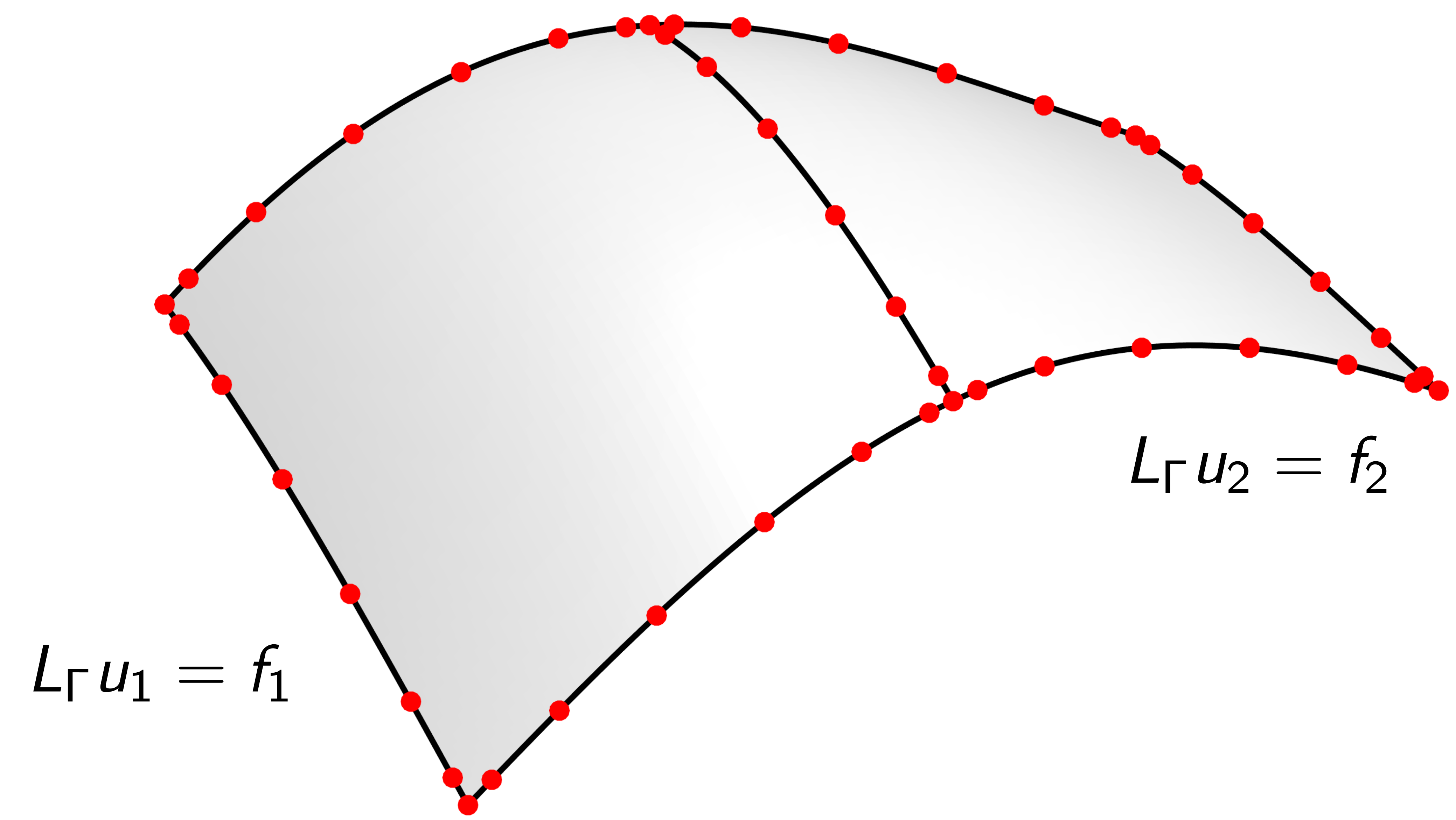




- Know how to do local solves on each element: "Solution operator" $S_1 \begin{bmatrix} g_1 \\ u_{\text{glue}} \end{bmatrix} \mapsto u_1 \qquad S_2 \begin{bmatrix} g_2 \\ u_{\text{glue}} \end{bmatrix} \mapsto u_2$
- Know how information flows out of each element: "Dirichlet-to-Neumann map" $DtN_1 \begin{bmatrix} g_1 \\ u_{\text{glue}} \end{bmatrix} \mapsto \frac{\partial u_1}{\partial \boldsymbol{n}_b} \qquad DtN_2 \begin{bmatrix} g_2 \\ u_{\text{glue}} \end{bmatrix} \mapsto \frac{\partial u_2}{\partial \boldsymbol{n}_b}$

[F., 2022]





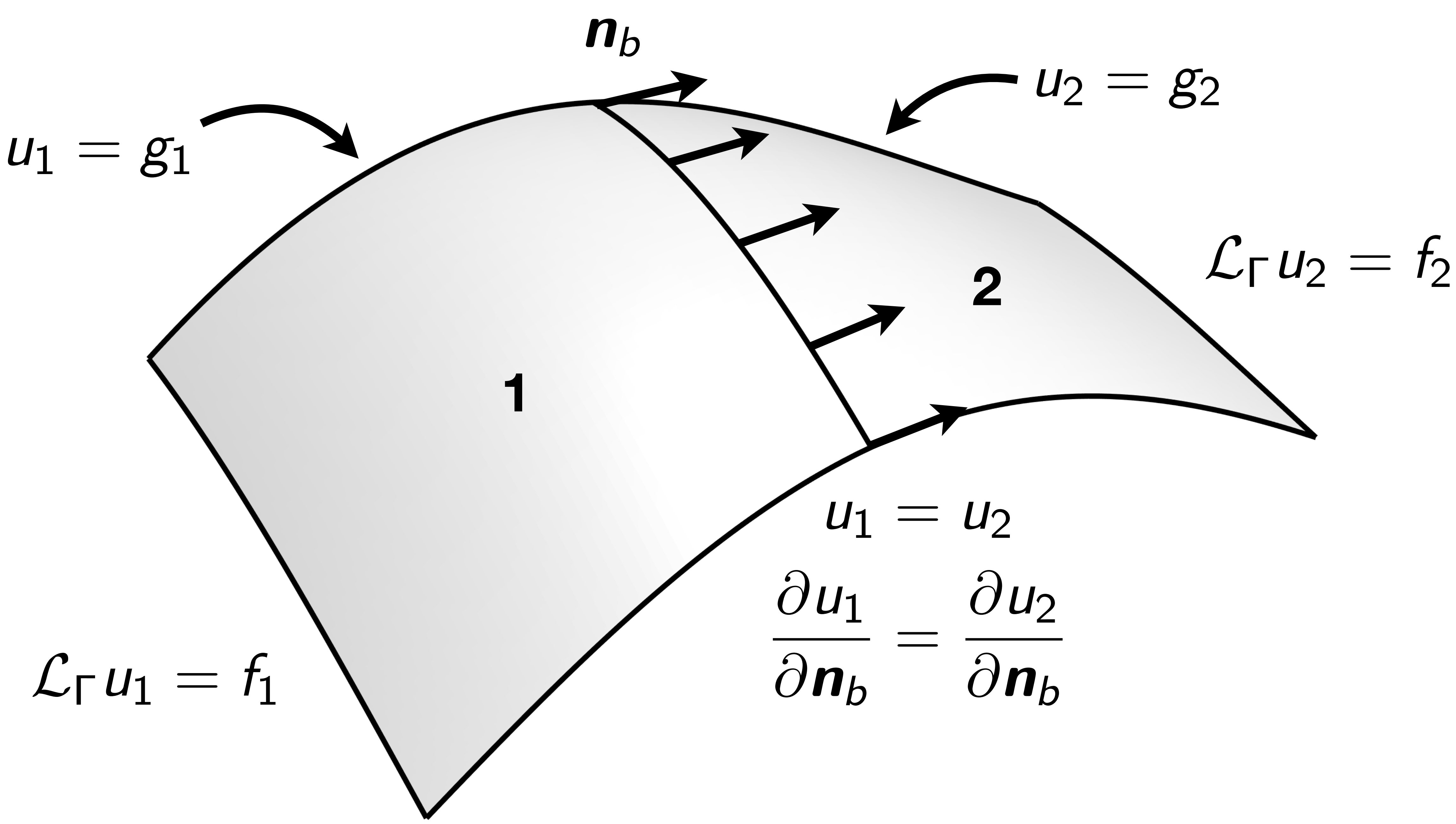
- Know how to do local solves on each element: “Solution operator”
- Know how information flows out of each element: “Dirichlet-to-Neumann map”

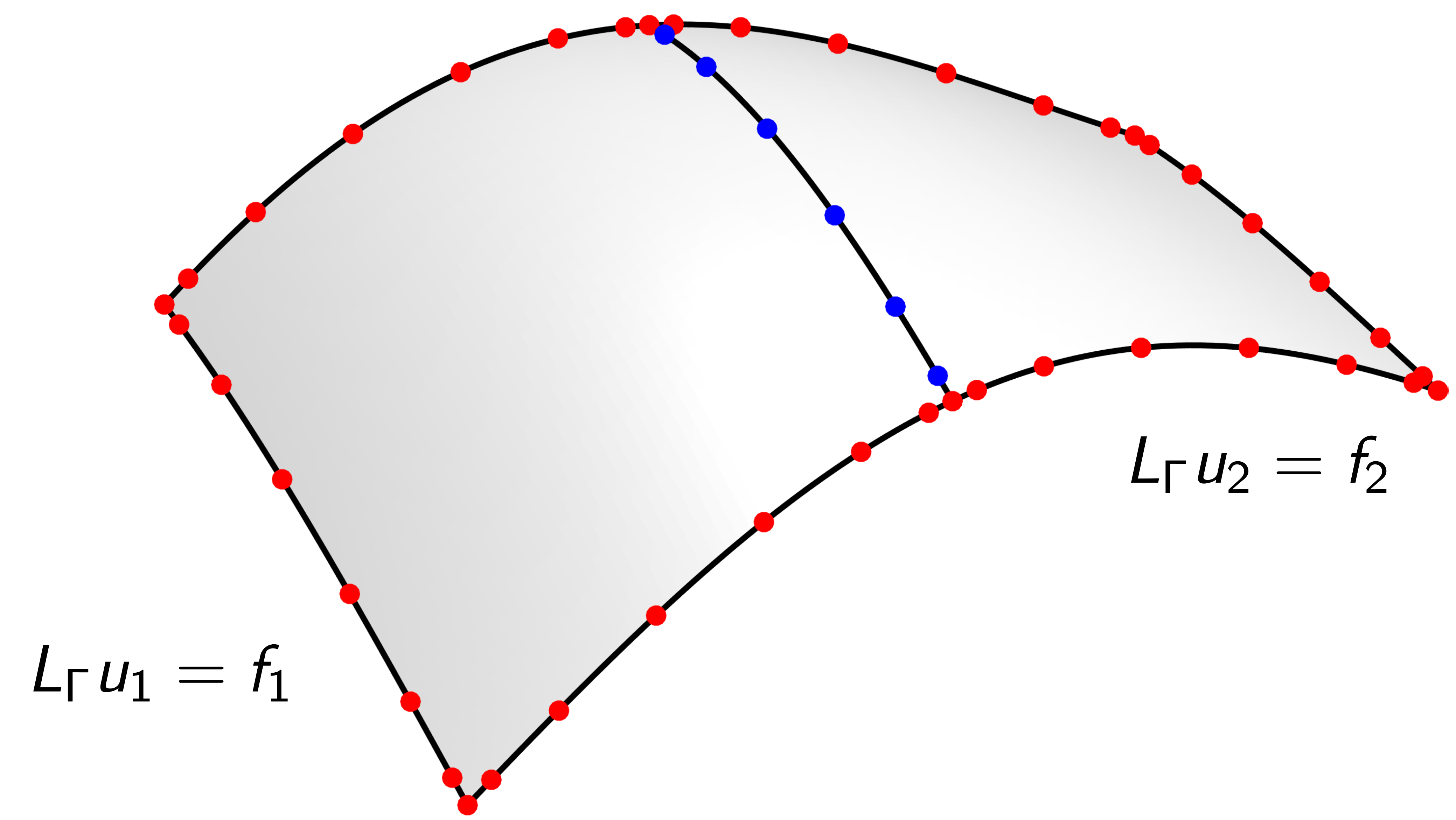
$$S_1 \begin{bmatrix} g_1 \\ u_{\text{glue}} \end{bmatrix} \mapsto u_1 \qquad S_2 \begin{bmatrix} g_2 \\ u_{\text{glue}} \end{bmatrix} \mapsto u_2$$

$$DtN_1 \begin{bmatrix} g_1 \\ u_{\text{glue}} \end{bmatrix} \mapsto \frac{\partial u_1}{\partial \boldsymbol{n}_b} \qquad DtN_2 \begin{bmatrix} g_2 \\ u_{\text{glue}} \end{bmatrix} \mapsto \frac{\partial u_2}{\partial \boldsymbol{n}_b}$$

[F., 2022]

High-order discretization
Two glued patches





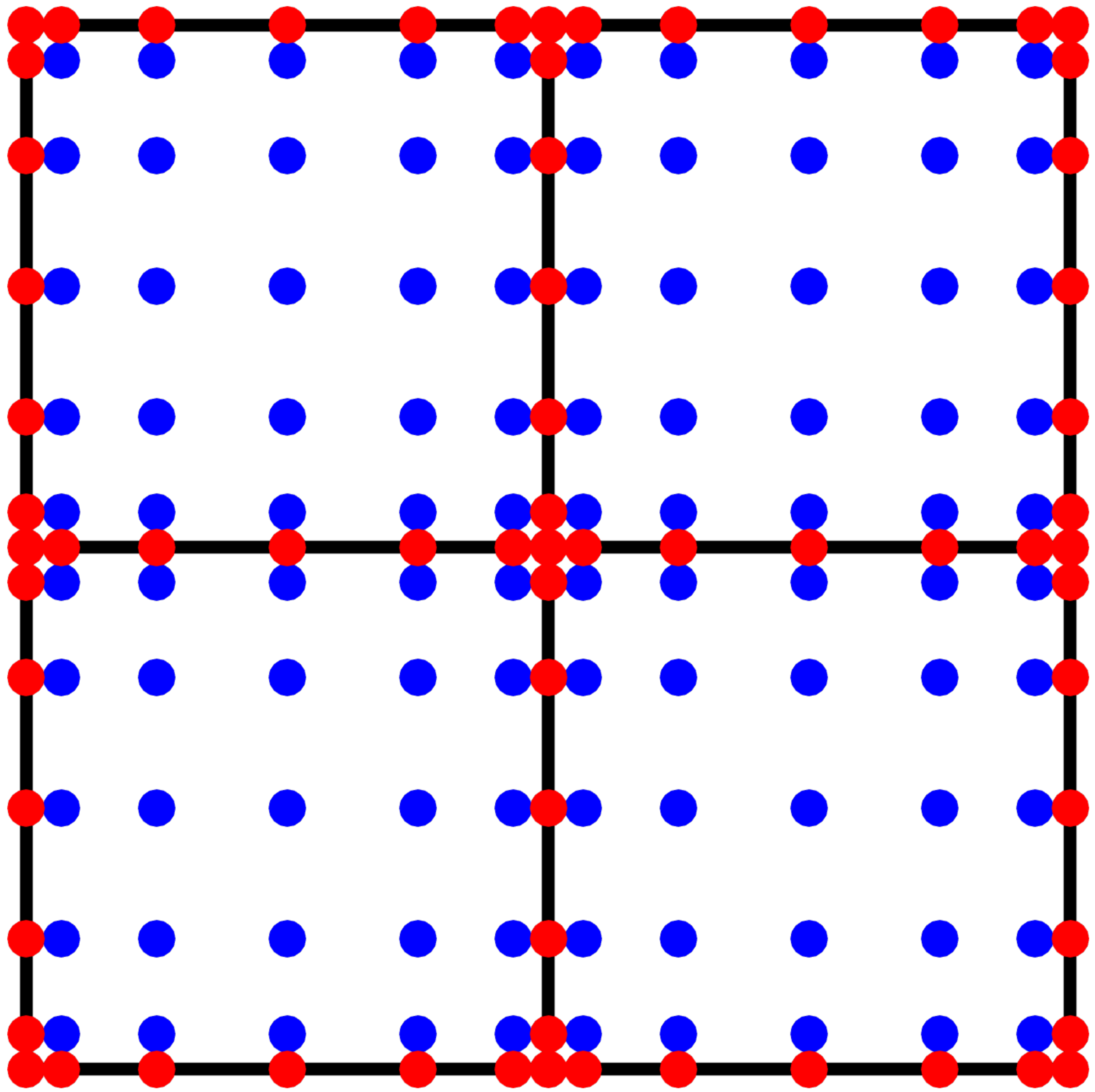
- Know how to do local solves on each element: “Solution operator” $S_1 \begin{bmatrix} g_1 \\ u_{\text{glue}} \end{bmatrix} \mapsto u_1$ $\quad S_2 \begin{bmatrix} g_2 \\ u_{\text{glue}} \end{bmatrix} \mapsto u_2$
- Know how information flows out of each element: “Dirichlet-to-Neumann map” $DtN_1 \begin{bmatrix} g_1 \\ u_{\text{glue}} \end{bmatrix} \mapsto \frac{\partial u_1}{\partial \boldsymbol{n}_b}$ $\quad DtN_2 \begin{bmatrix} g_2 \\ u_{\text{glue}} \end{bmatrix} \mapsto \frac{\partial u_2}{\partial \boldsymbol{n}_b}$
- Take Schur complement to eliminate interior degrees of freedom:

$$S_{\text{glue}} = -\left(DtN_1^{\text{glue}} + DtN_2^{\text{glue}}\right)^{-1} \left[DtN_1^{\text{glue},1} \quad DtN_2^{\text{glue},2}\right] \qquad S_{\text{glue}} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = u_{\text{glue}}$$

[F., 2022]

A fast direct solver on surfaces
Hierarchical Poincaré–Steklov method

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)



[Martinsson, 2013]

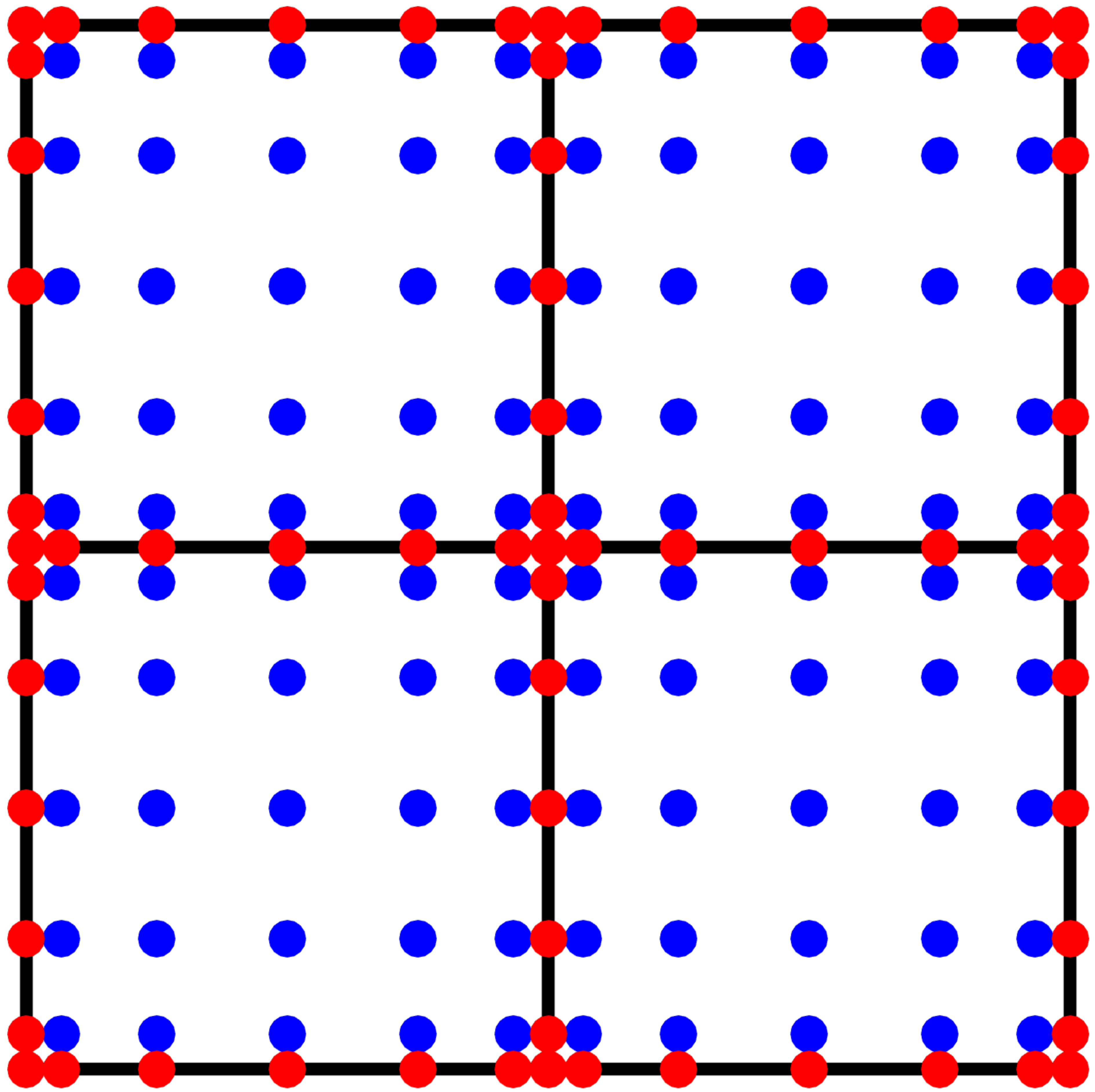
[Gillman & Martinsson, 2014]

[F., 2022]

A fast direct solver on surfaces
Hierarchical Poincaré–Steklov method

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

Build element operators $\mathcal{O}(Np^6)$

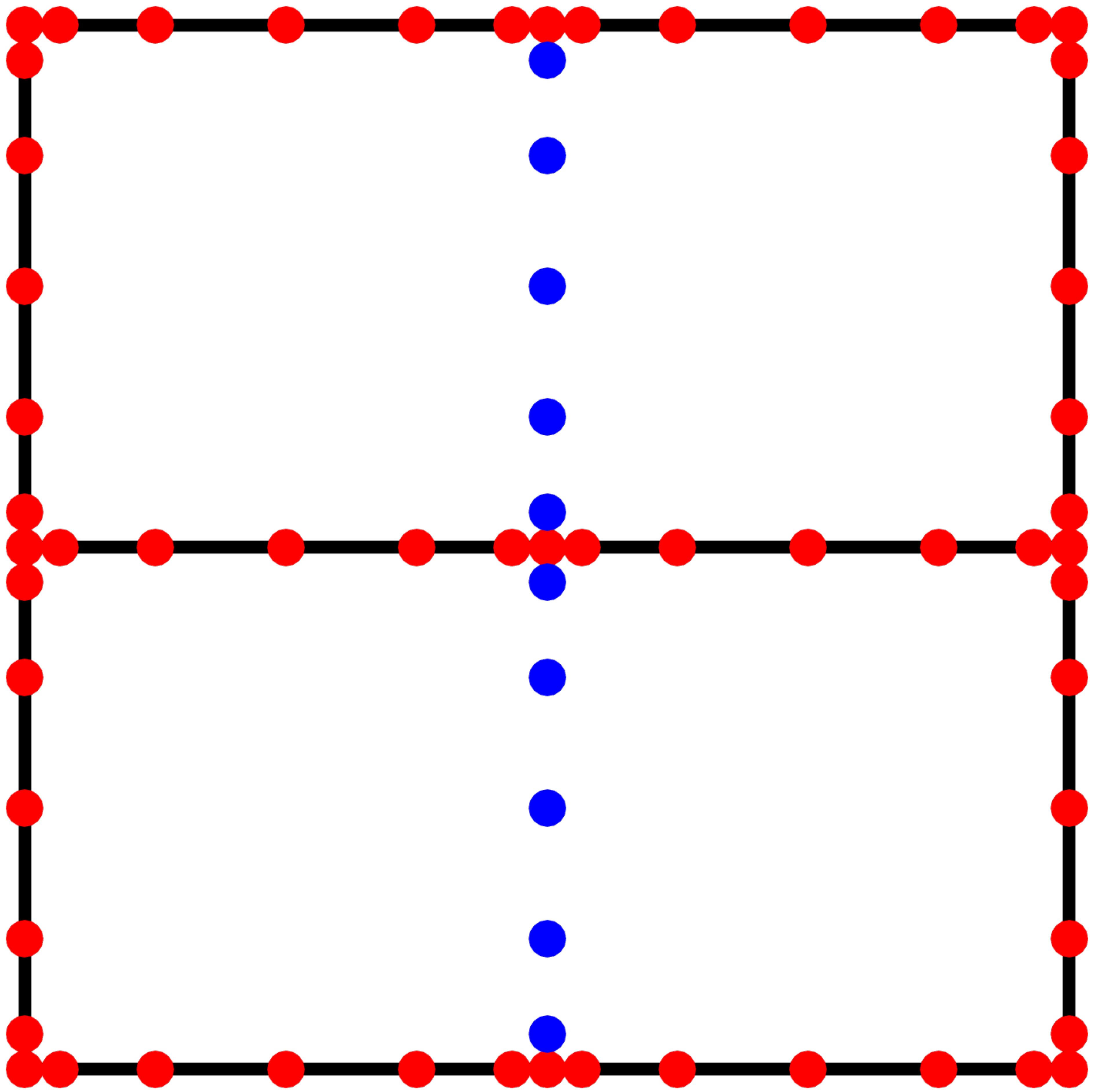


[Martinsson, 2013]
[Gillman & Martinsson, 2014]
[F., 2022]

A fast direct solver on surfaces
Hierarchical Poincaré–Steklov method

<u>Key idea</u>: Recursively glue elements together in a hierarchy (cf. nested dissection)

Merge operators $\mathcal{O}(N^{3/2})$



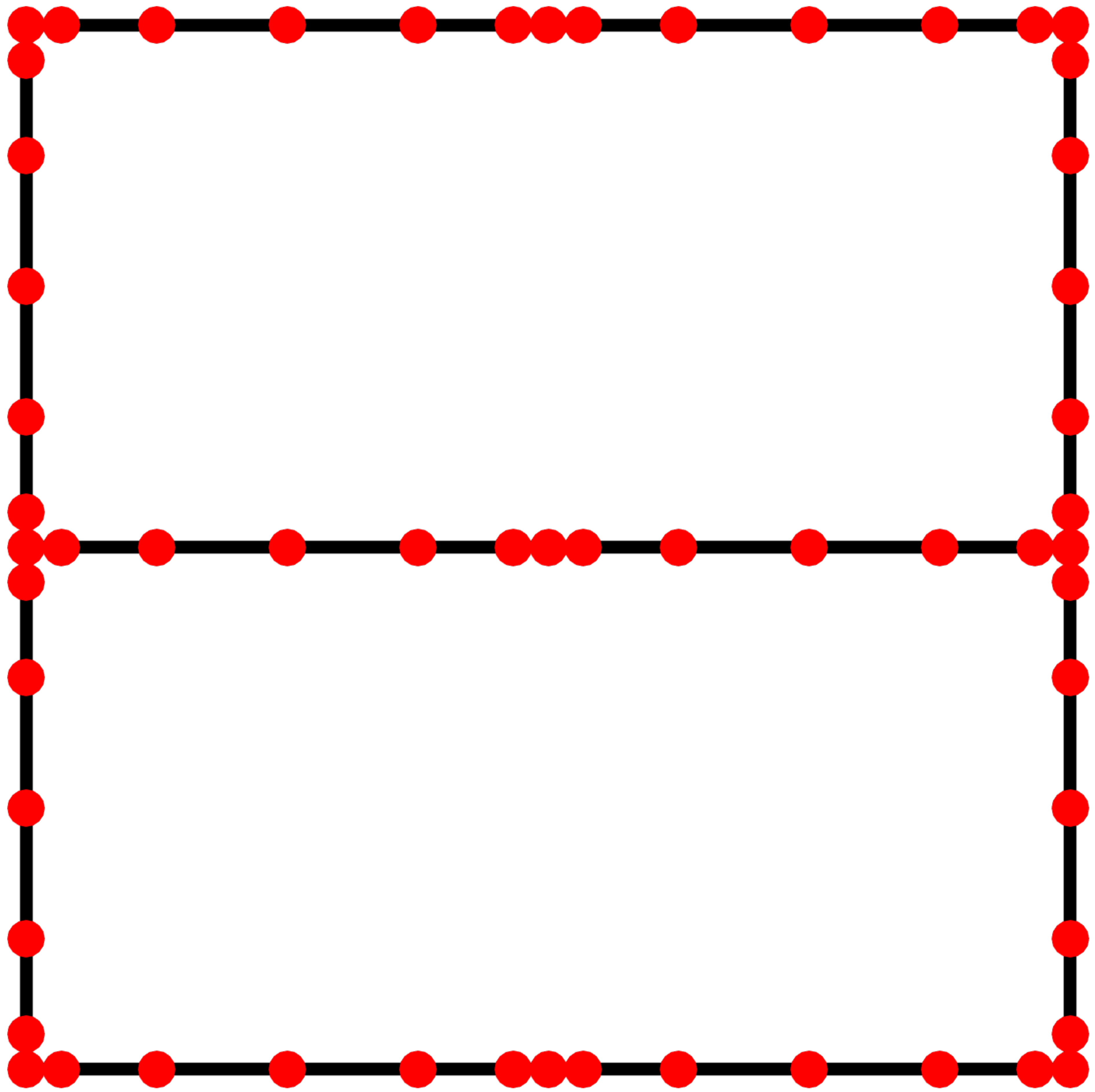
[Martinsson, 2013]

[Gillman & Martinsson, 2014]

[F., 2022]

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

Merge operators $\mathcal{O}(N^{3/2})$



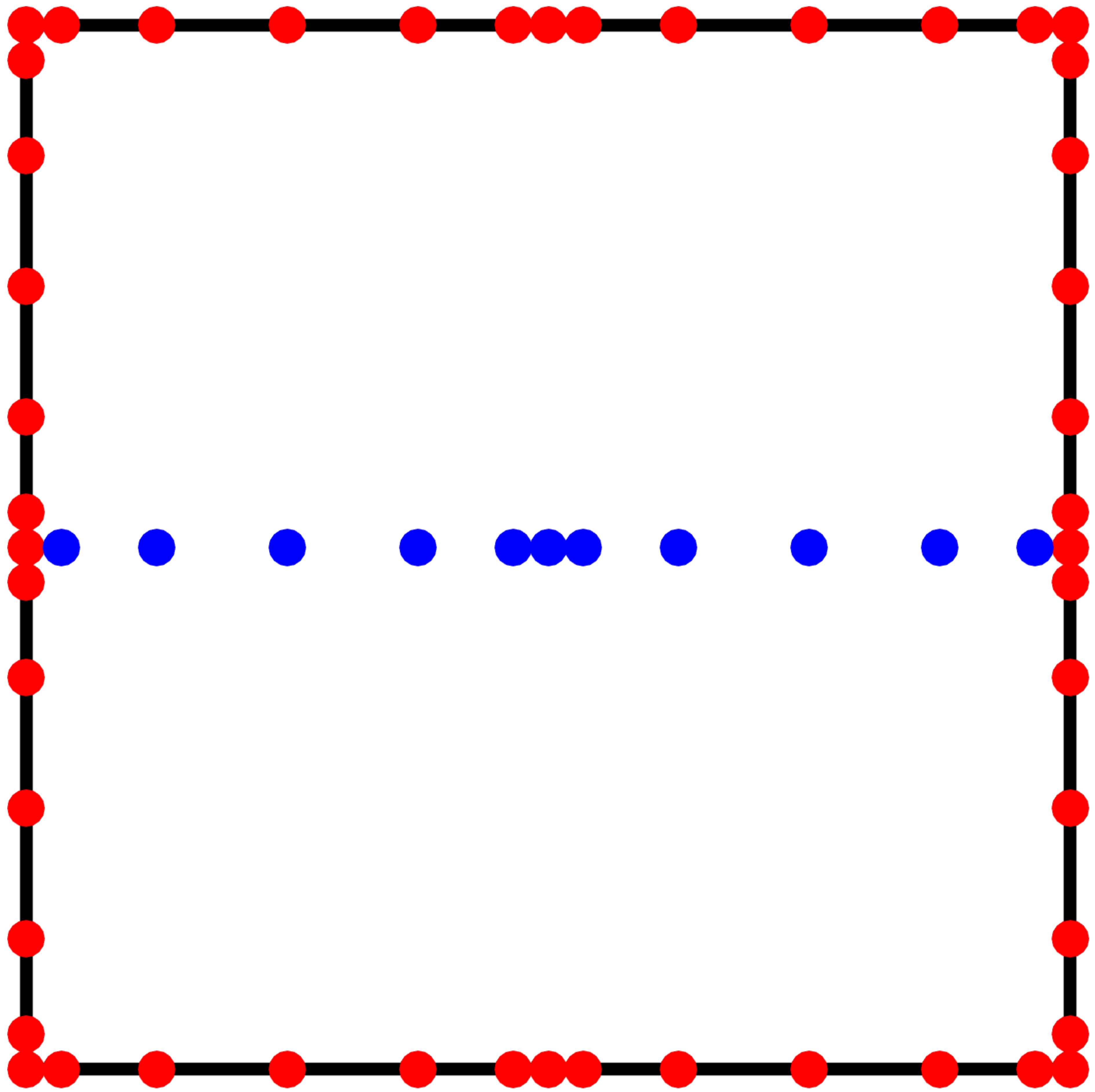
[Martinsson, 2013]

[Gillman & Martinsson, 2014]

[F., 2022]

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

Merge operators $\mathcal{O}(N^{3/2})$



[Martinsson, 2013]

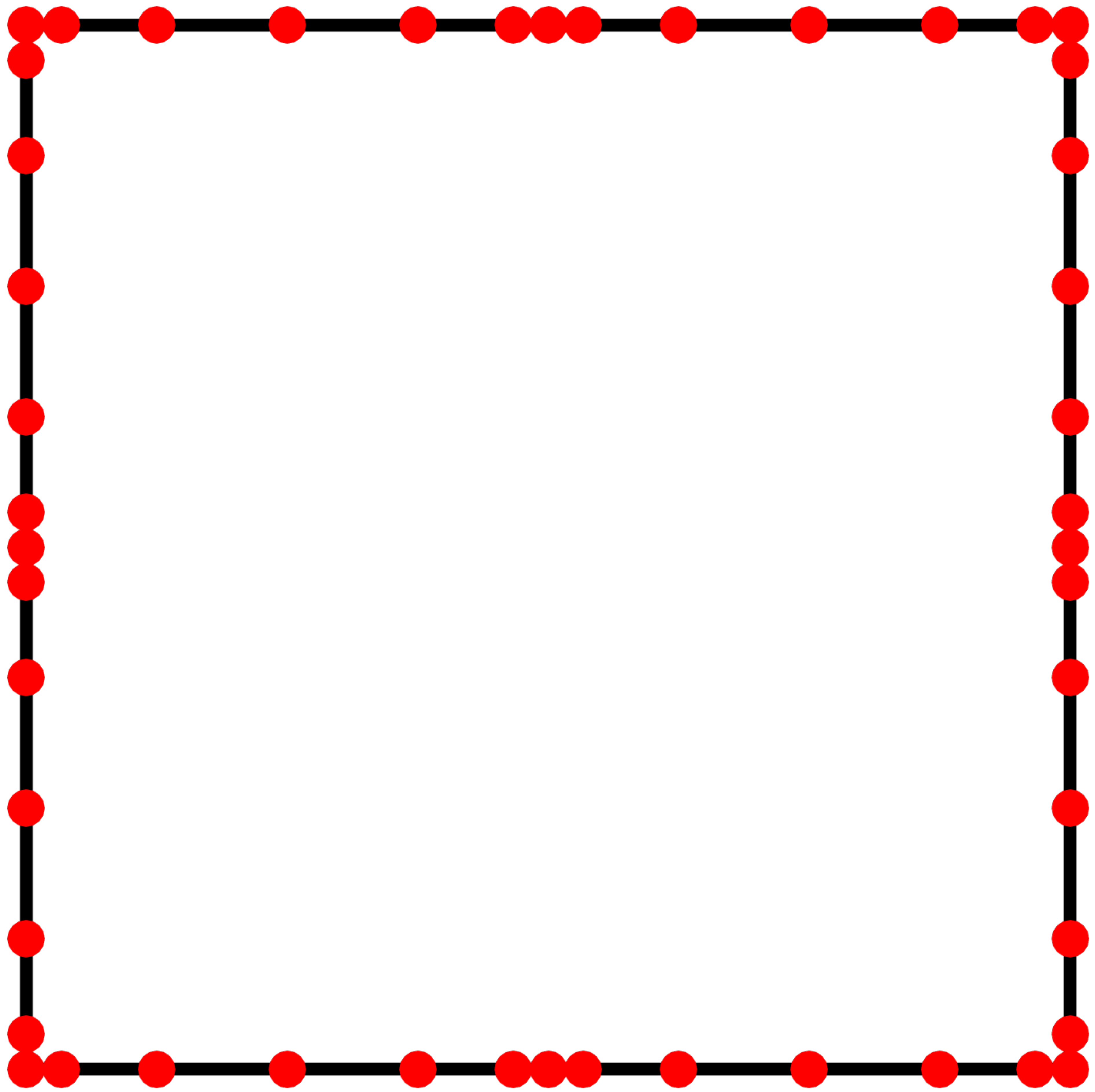
[Gillman & Martinsson, 2014]

[F., 2022]

A fast direct solver on surfaces
Hierarchical Poincaré–Steklov method

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

Merge operators $\mathcal{O}(N^{3/2})$



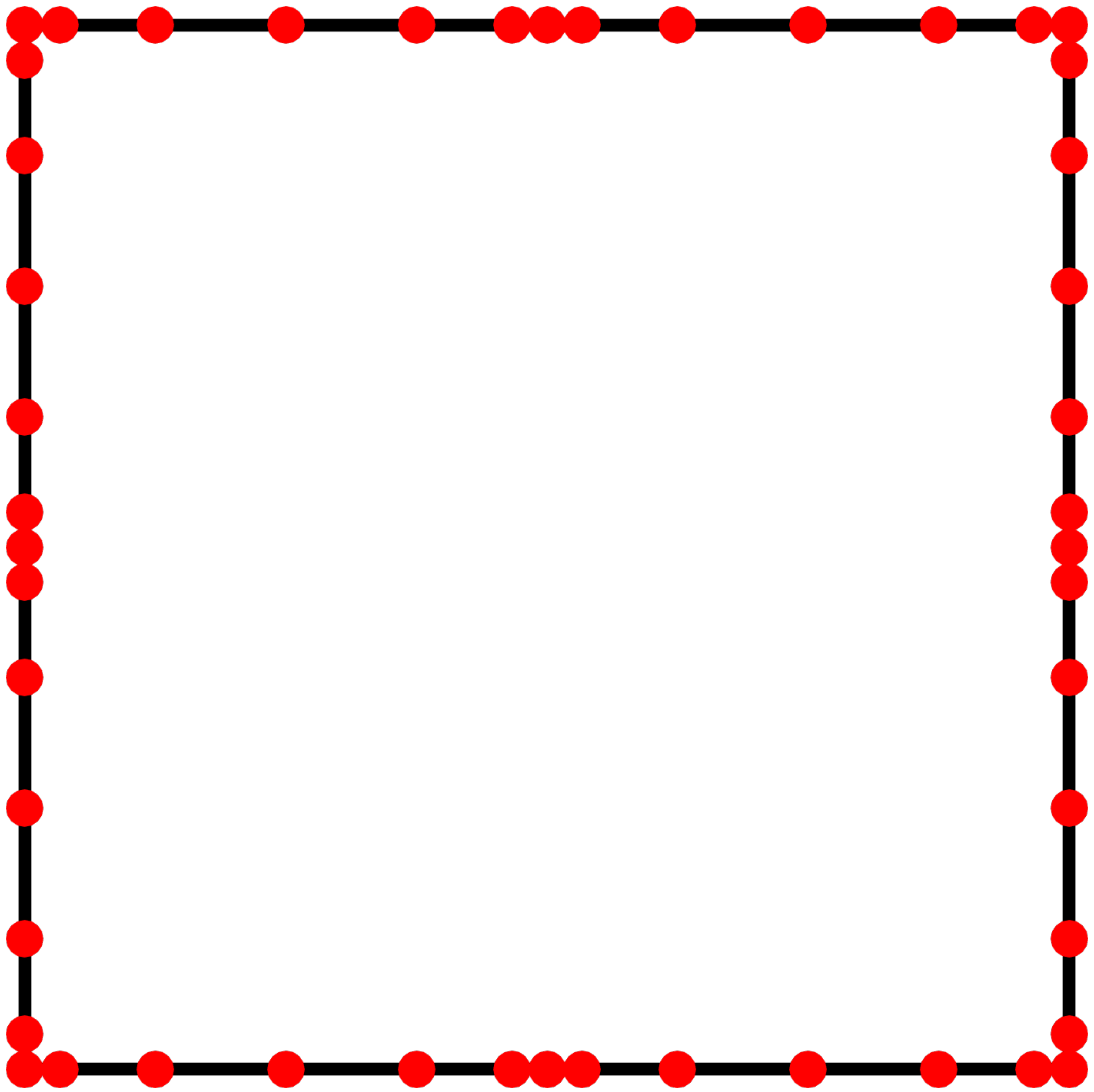
[Martinsson, 2013]

[Gillman & Martinsson, 2014]

[F., 2022]

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

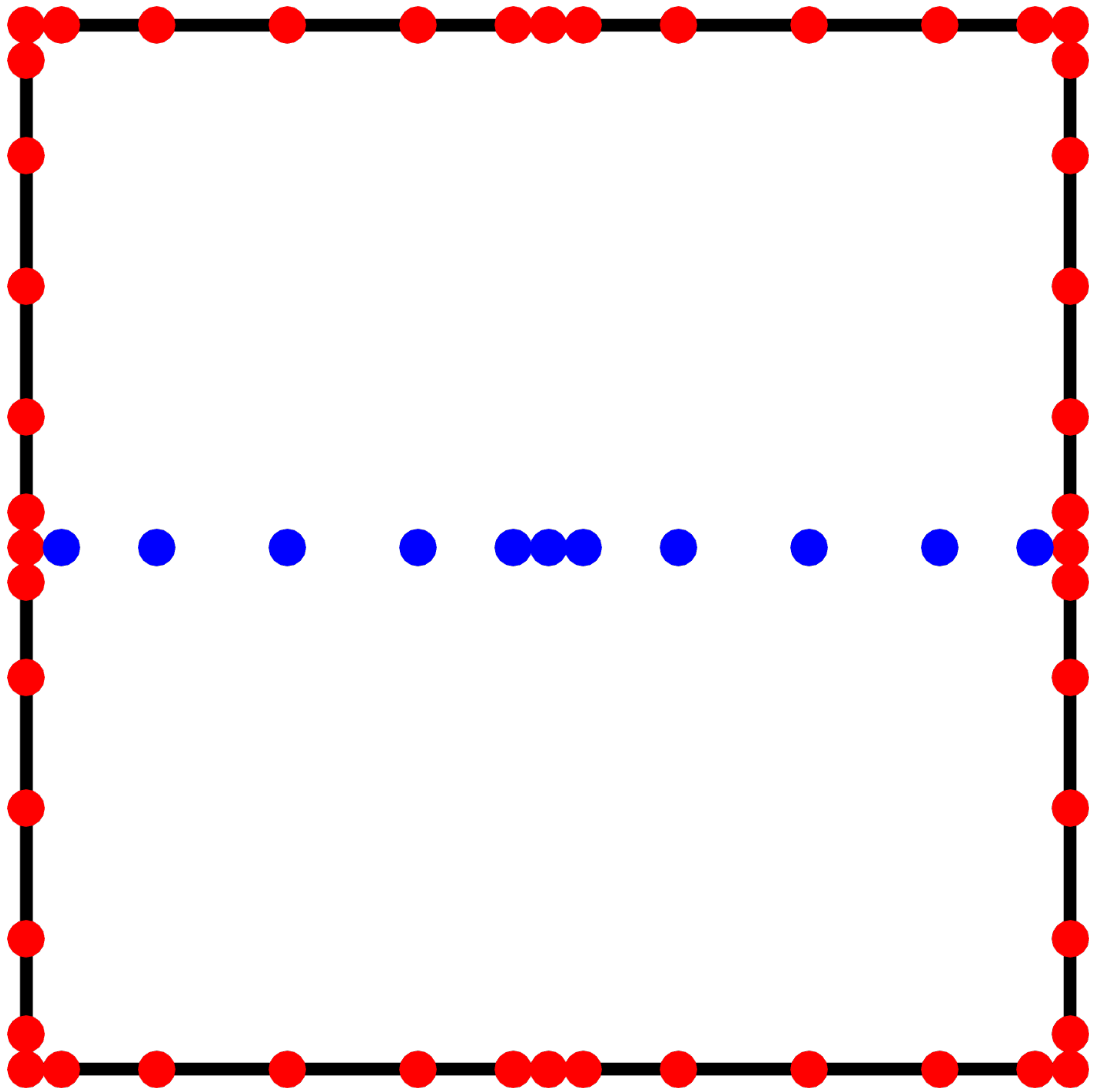
Apply solution operators $\mathcal{O}(N \log N)$



[Martinsson, 2013]
[Gillman & Martinsson, 2014]
[F., 2022]

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

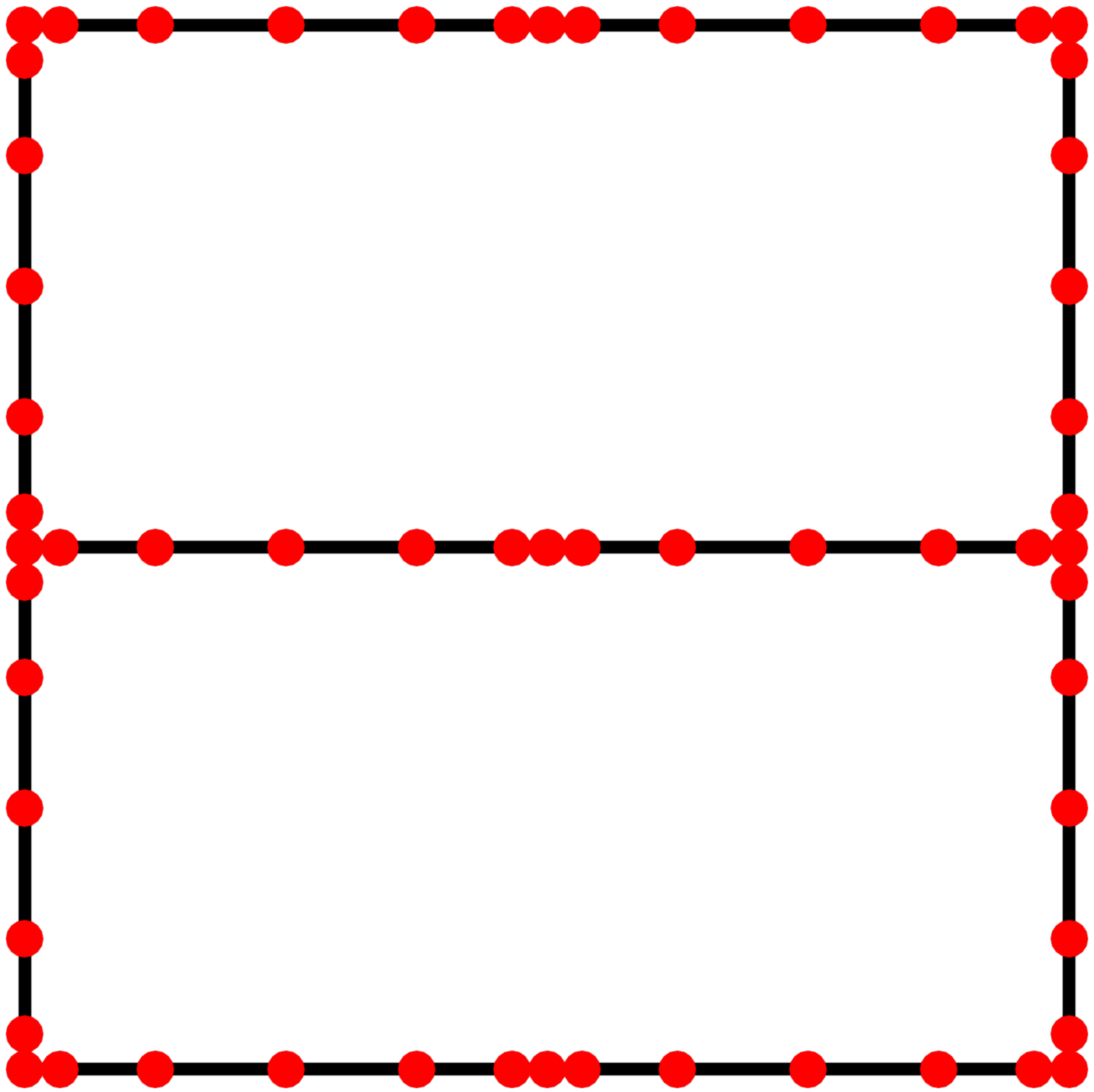
Apply solution operators $\mathcal{O}(N \log N)$



[Martinsson, 2013]
[Gillman & Martinsson, 2014]
[F., 2022]

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

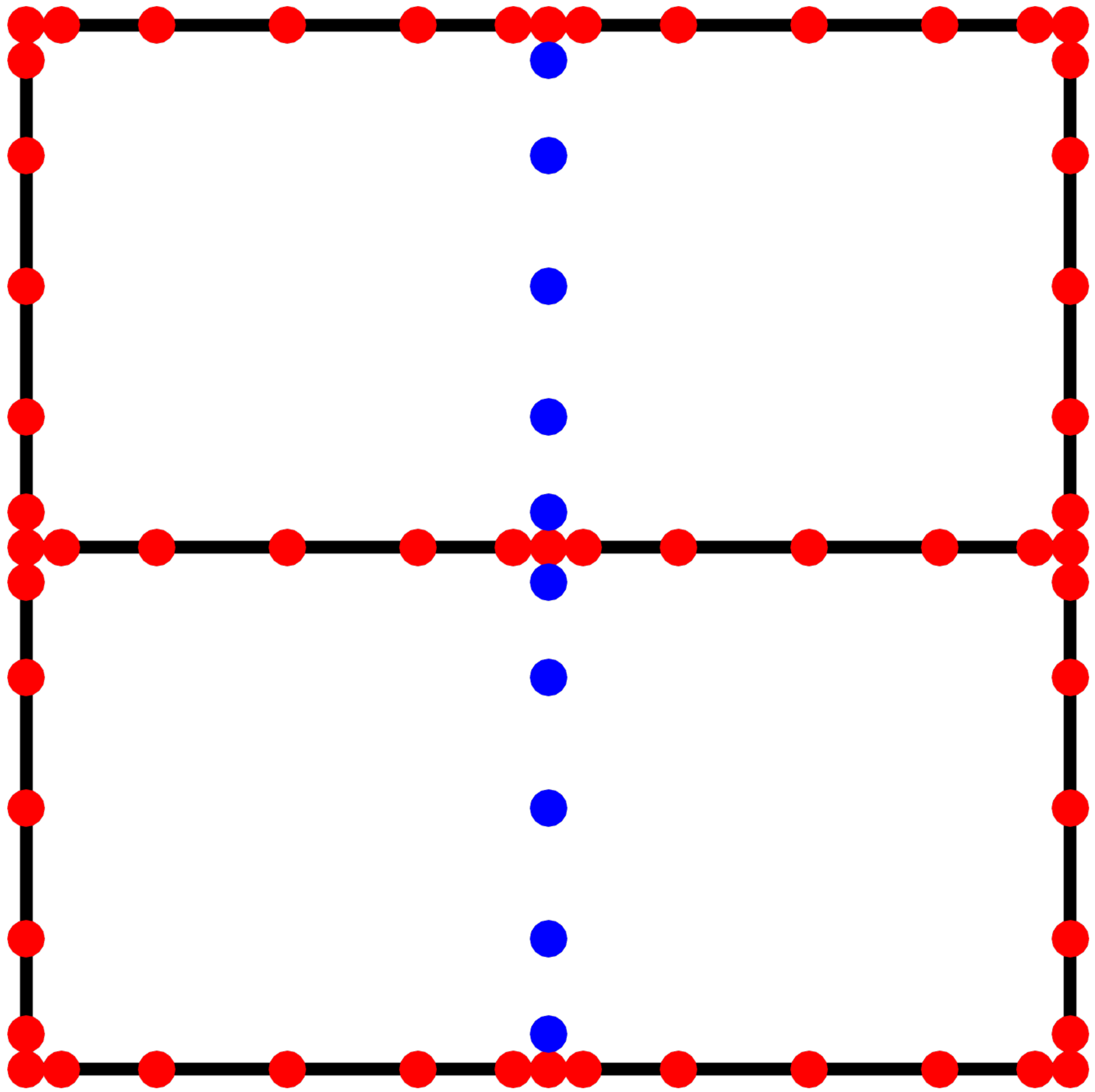
Apply solution operators $\mathcal{O}(N \log N)$



[Martinsson, 2013]
[Gillman & Martinsson, 2014]
[F., 2022]

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

Apply solution operators $\mathcal{O}(N \log N)$



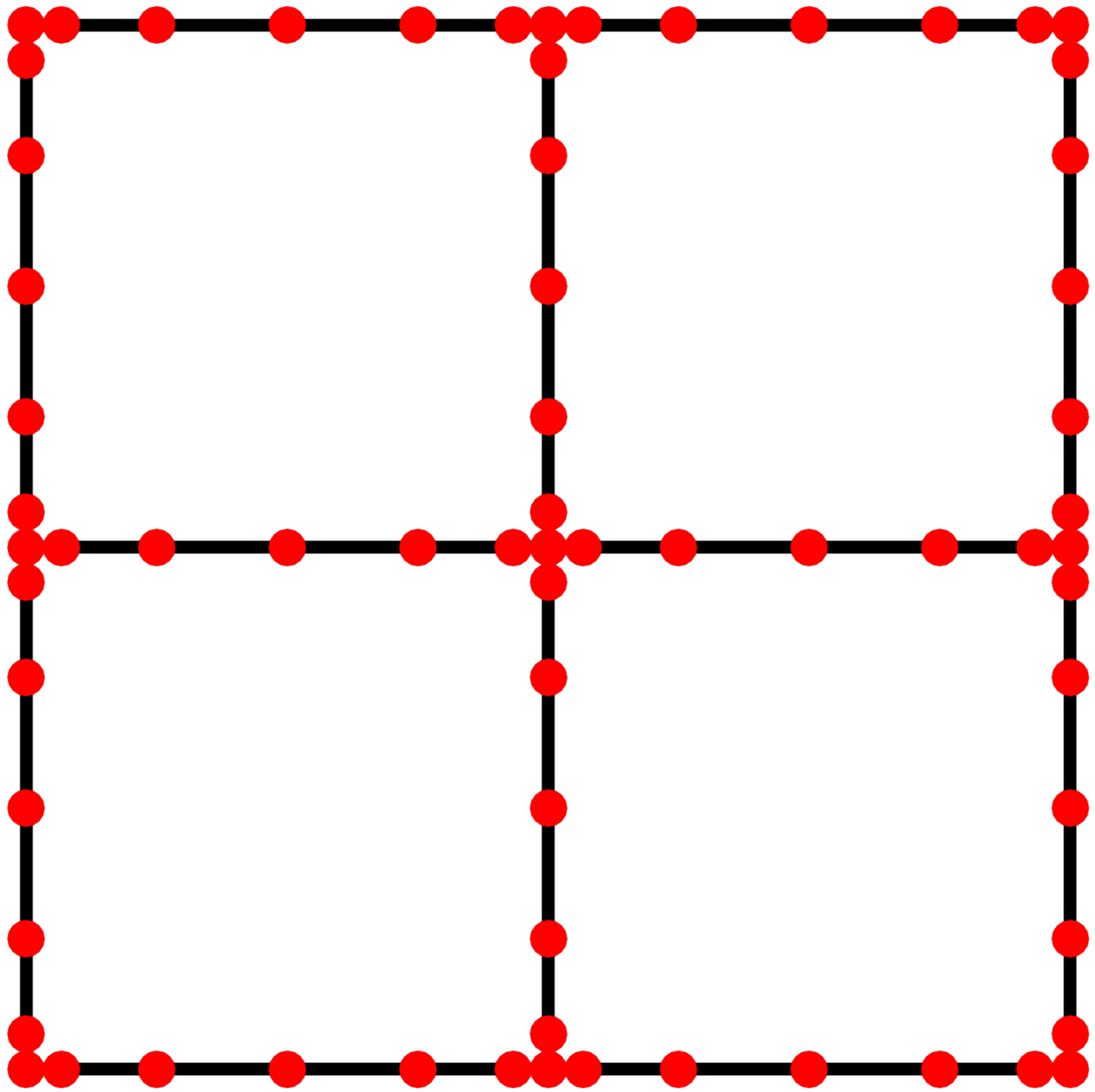
[Martinsson, 2013]

[Gillman & Martinsson, 2014]

[F., 2022]

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

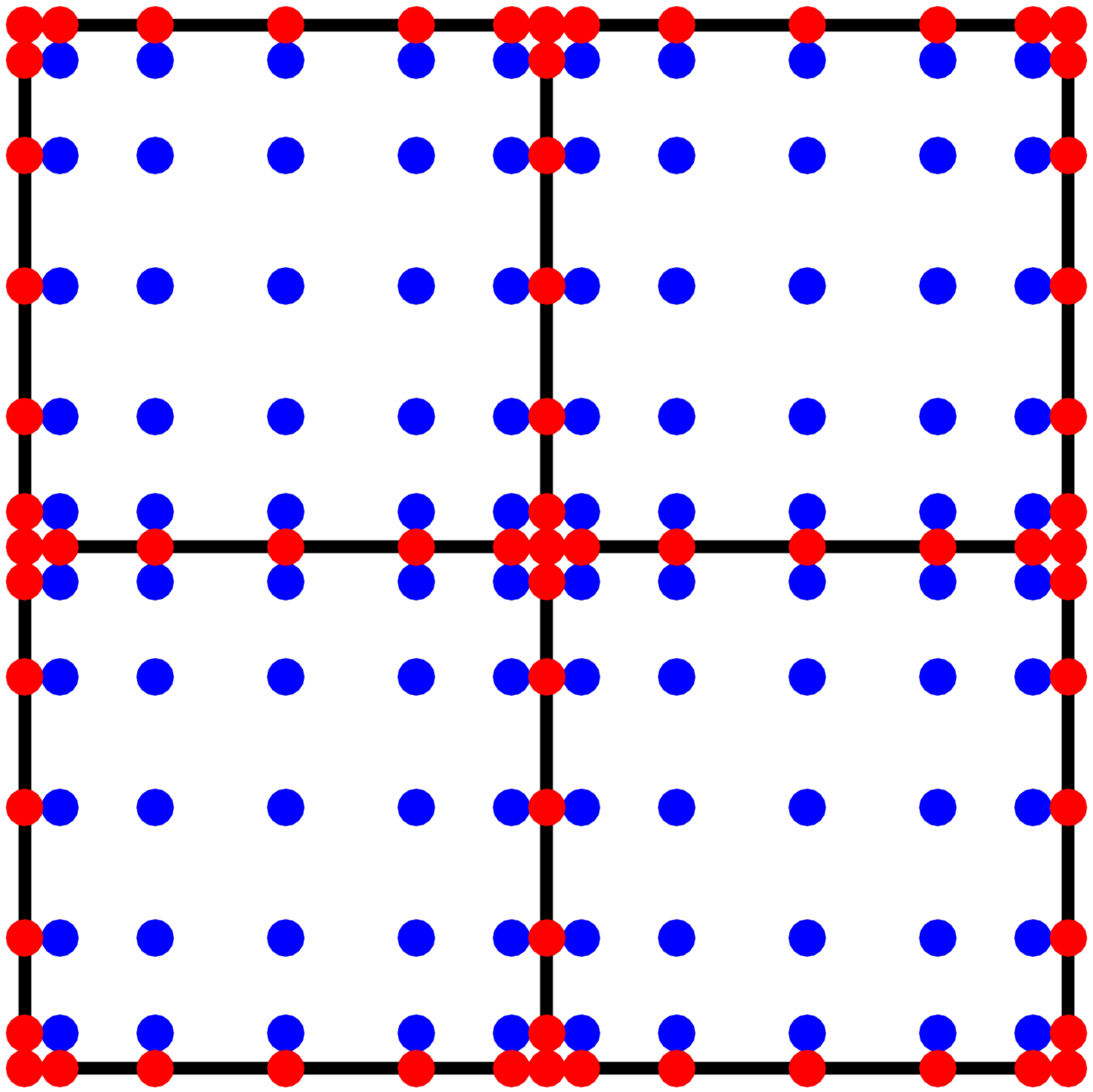
Apply solution operators $\mathcal{O}(N \log N)$



[Martinsson, 2013]
[Gillman & Martinsson, 2014]
[F., 2022]

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

Apply solution operators $\mathcal{O}(N \log N)$



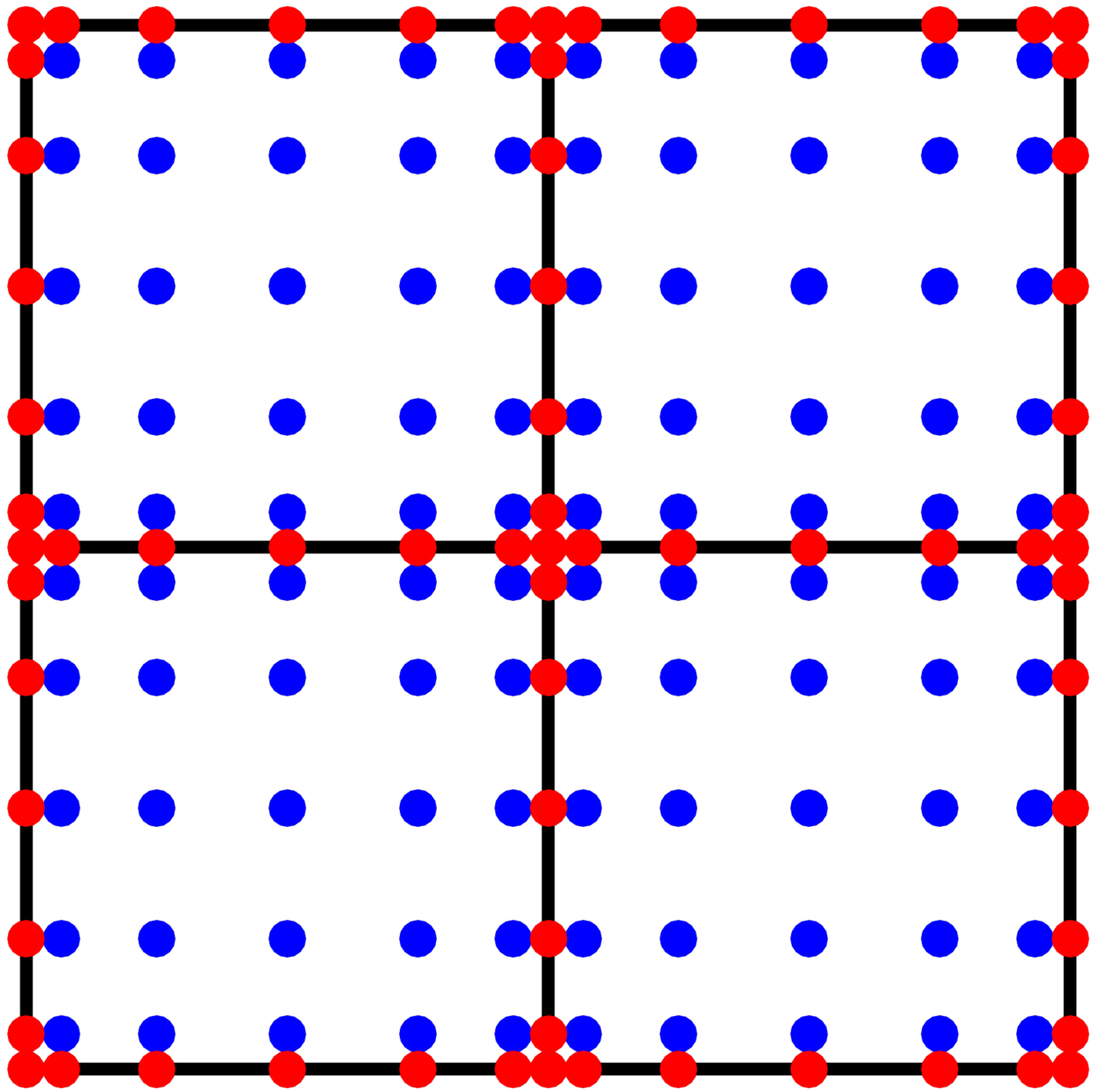
[Martinsson, 2013]

[Gillman & Martinsson, 2014]

[F., 2022]

Key idea: Recursively glue elements together in a hierarchy (cf. nested dissection)

Apply solution operators $\mathcal{O}(N \log N)$



$$\underbrace{N^{3/2}}_{\text{Factorization}} + \underbrace{N \log N}_{\text{Solve}}$$

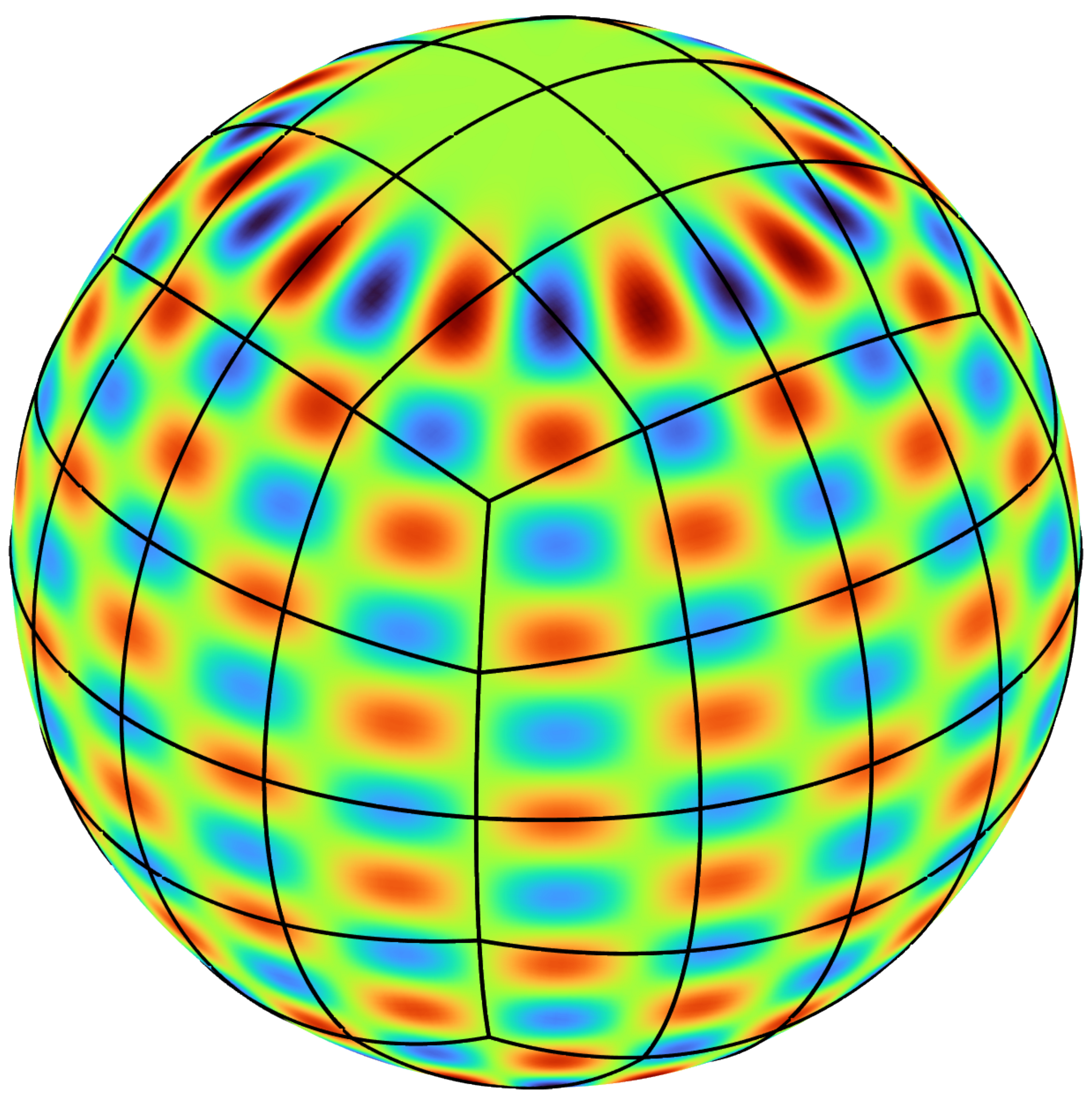
Factorization results in a hierarchy of solution operators stored in memory, so repeated solves are fast.

[Martinsson, 2013]
[Gillman & Martinsson, 2014]
[F., 2022]

Examples

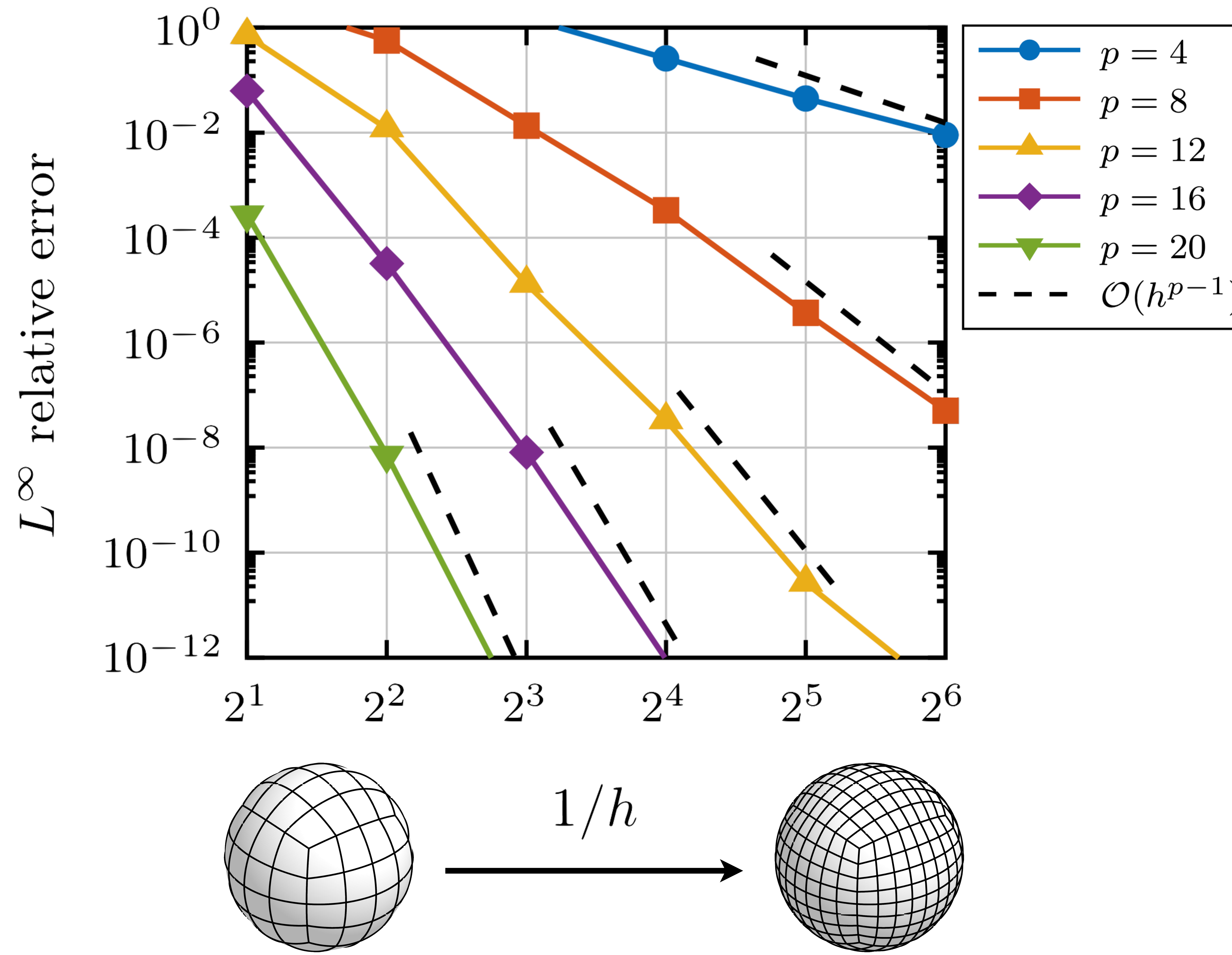
Laplace–Beltrami and convergence

$$\Delta_\Gamma u = f, \quad \Gamma = \text{sphere}$$

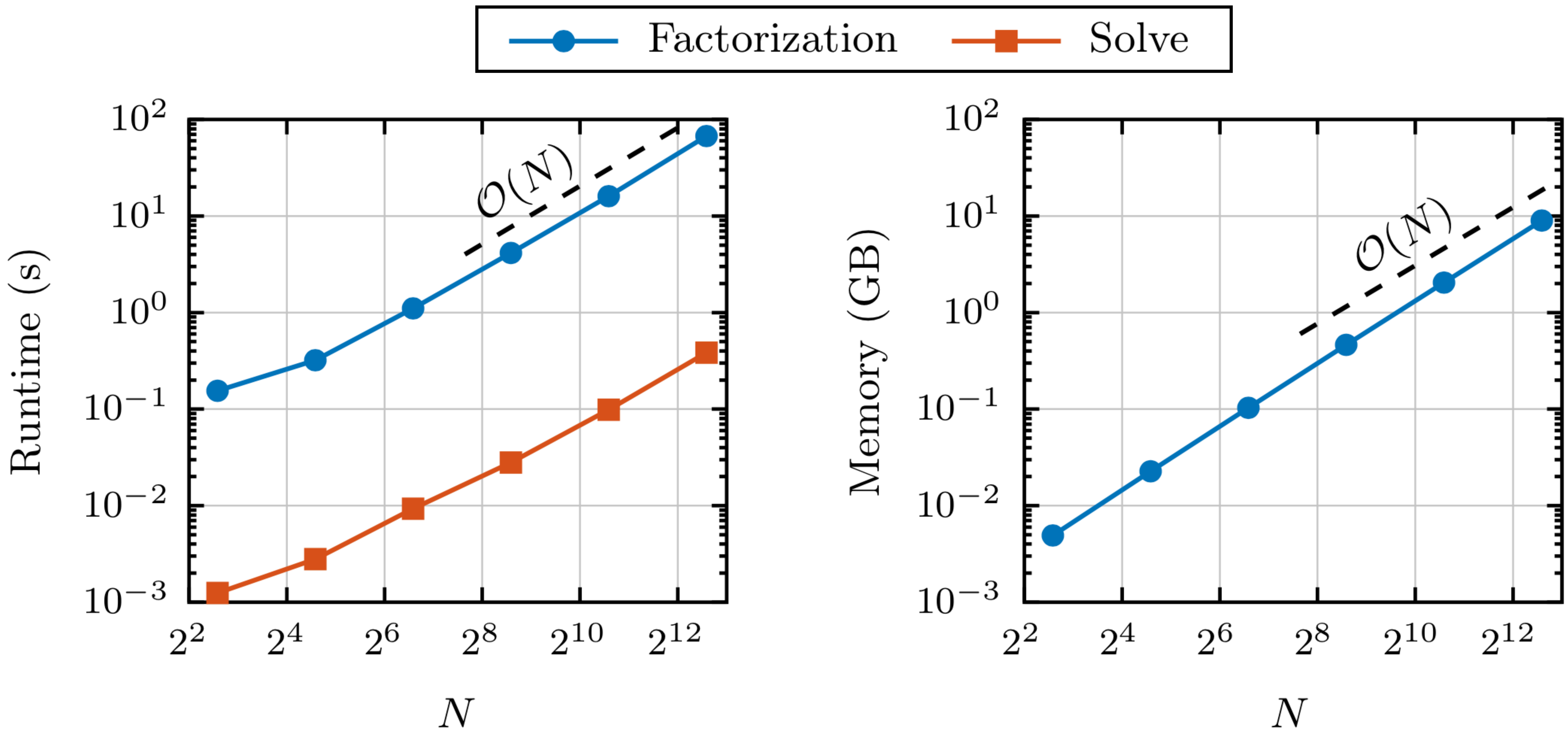


$$u(\boldsymbol{x}) = \text{spherical harmonic, } Y_\ell^m(\boldsymbol{x})$$

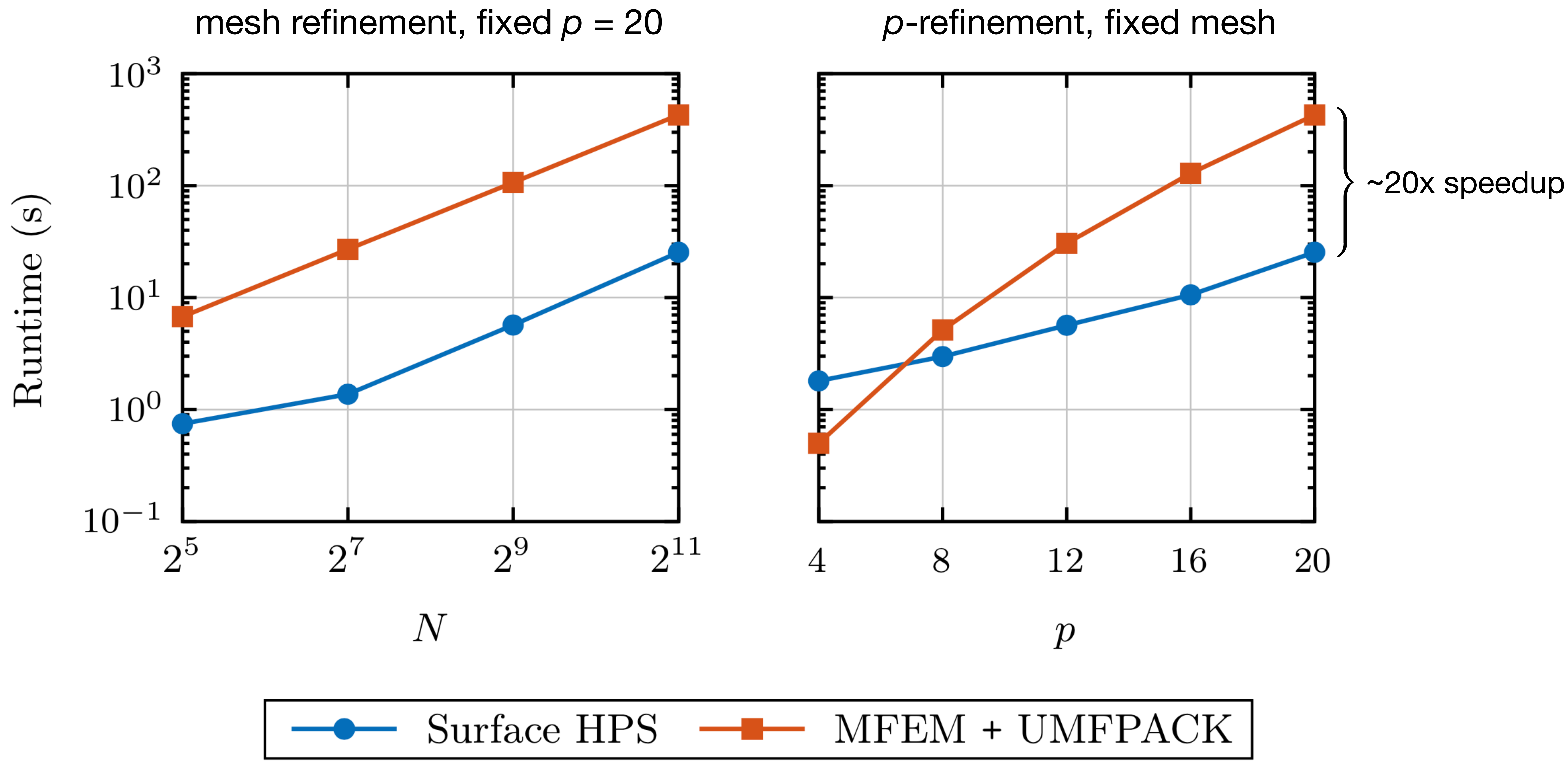
$$f(\boldsymbol{x}) = -\ell(\ell+1)Y_\ell^m(\boldsymbol{x})$$



<u>Computational complexity:</u> $\mathcal{O}(N^{3/2})$ factorization, but $\mathcal{O}(N)$ is observed pre-asymptotically.



Significant speedups at high order when compared to FEM with a sparse direct solver.



Any smooth vector field $\boldsymbol{f}$ tangent to a surface can be written as:

$$\boldsymbol{f} = \underbrace{\nabla_\Gamma u}_{\textit{curl-free}} + \underbrace{\boldsymbol{n} \times \nabla_\Gamma v}_{\textit{div-free}} + \underbrace{\boldsymbol{w}}_{\textit{harmonic}}$$

where $\boldsymbol{w}$ satisfies $\nabla_\Gamma \cdot \boldsymbol{w} = 0$ and $\nabla_\Gamma \cdot (\boldsymbol{n} \times \boldsymbol{w}) = 0$.

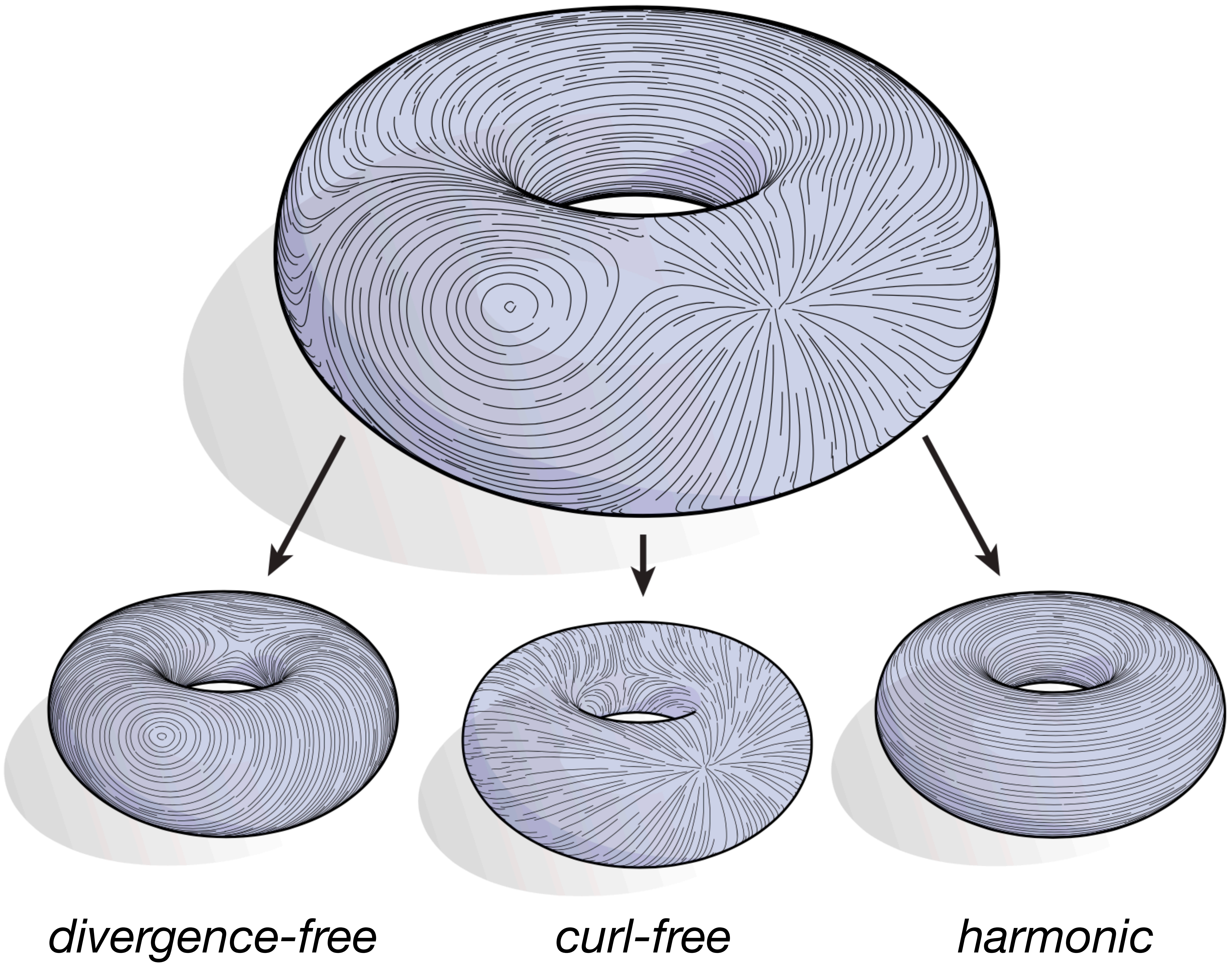


Illustration by Keenan Crane

Any smooth vector field $\boldsymbol{f}$ tangent to a surface can be written as:

$$\boldsymbol{f} = \underbrace{\nabla_\Gamma u}_{\textit{curl-free}} + \underbrace{\boldsymbol{n} \times \nabla_\Gamma v}_{\textit{div-free}} + \underbrace{\boldsymbol{w}}_{\textit{harmonic}}$$

where $\boldsymbol{w}$ satisfies $\nabla_\Gamma \cdot \boldsymbol{w} = 0$ and $\nabla_\Gamma \cdot (\boldsymbol{n} \times \boldsymbol{w}) = 0$.

One may compute this decomposition by solving

$$\Delta_\Gamma u = \nabla_\Gamma \cdot \boldsymbol{f}$$

$$\Delta_\Gamma v = -\nabla_\Gamma \cdot (\boldsymbol{n} \times \boldsymbol{f})$$

and then setting $\boldsymbol{w} = \boldsymbol{f} - \nabla_\Gamma u - \boldsymbol{n} \times \nabla_\Gamma v$.

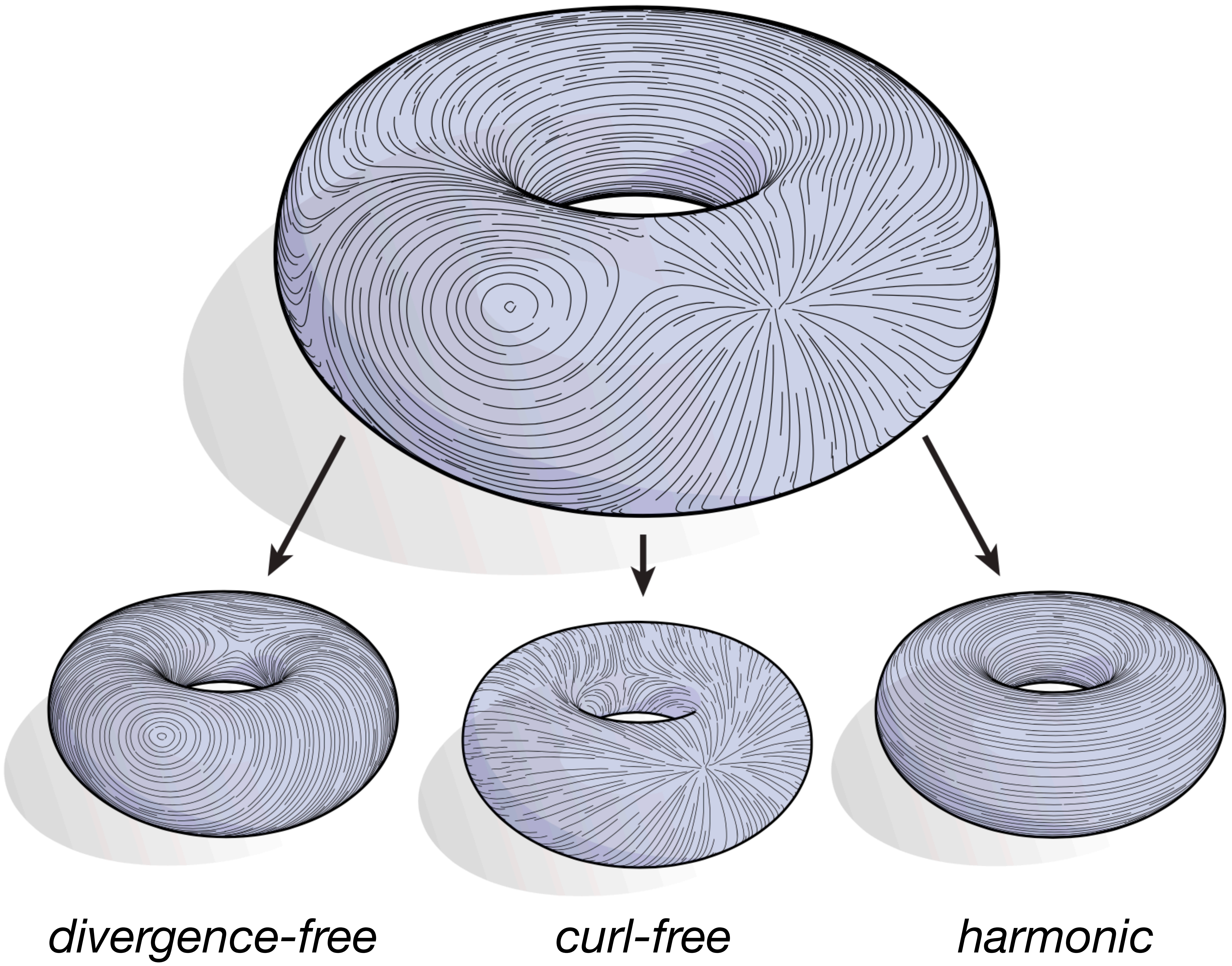
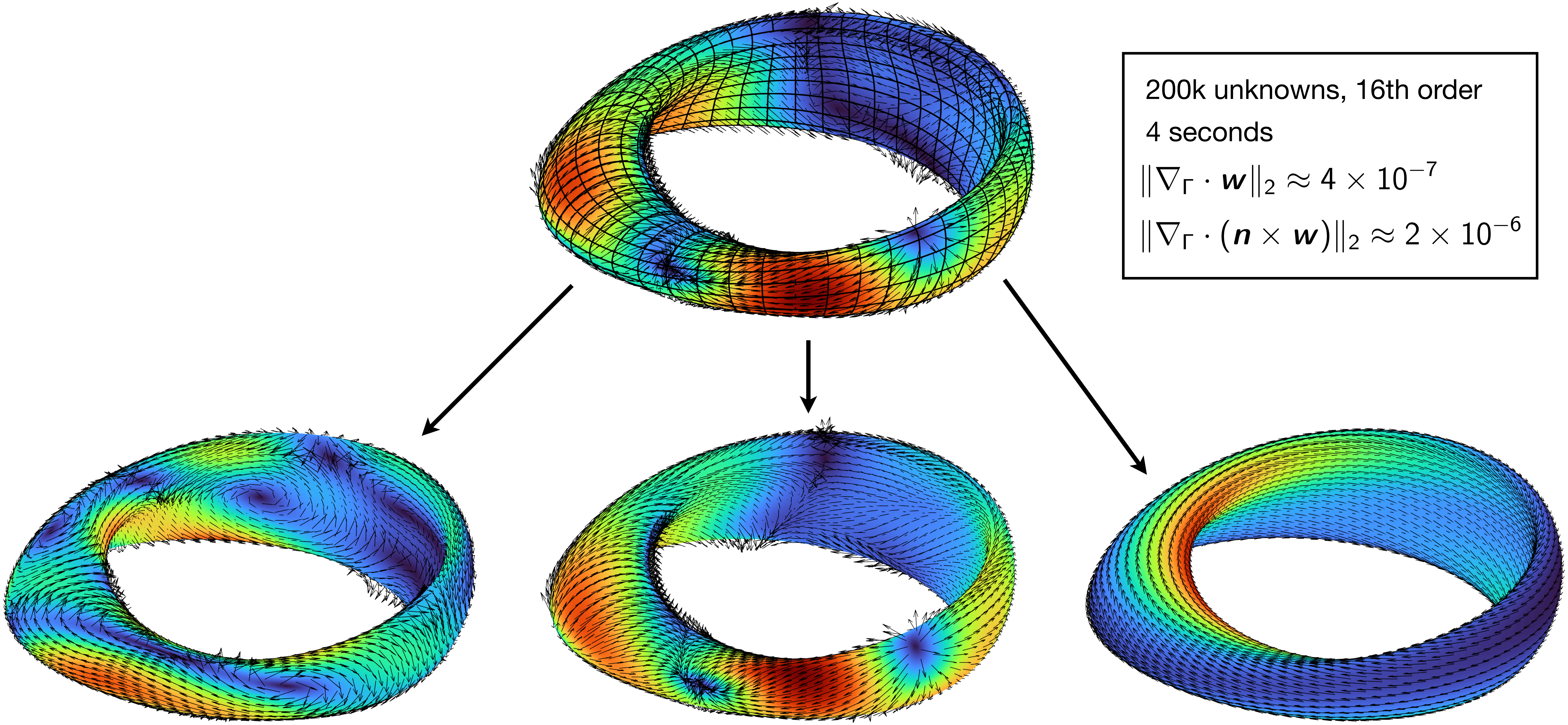


Illustration by Keenan Crane



200k unknowns, 16th order

4 seconds

$\|\nabla_\Gamma \cdot \boldsymbol{w}\|_2 \approx 4 \times 10^{-7}$

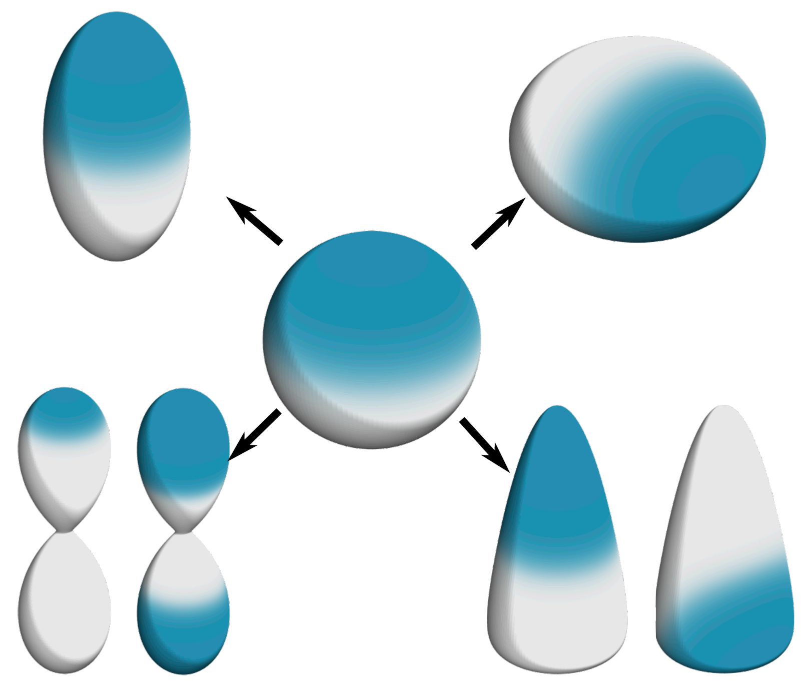
$\|\nabla_\Gamma \cdot (\boldsymbol{n} \times \boldsymbol{w})\|_2 \approx 2 \times 10^{-6}$

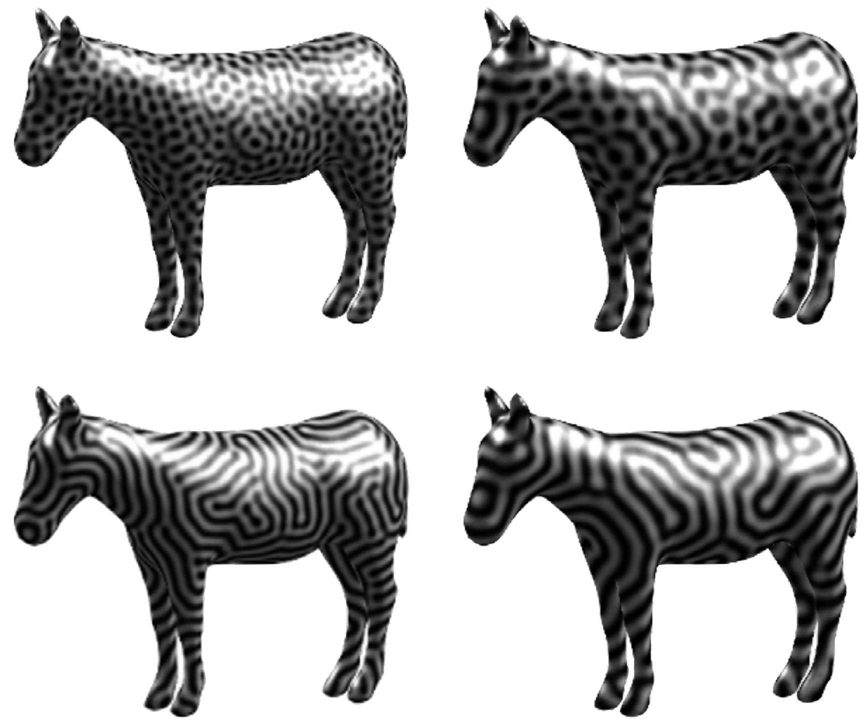
divergence-free *curl-free* *harmonic*

Examples
Reaction–diffusion systems

- Reaction–diffusion processes are ubiquitous in biology.

$$\frac{\partial u}{\partial t} = \underbrace{\mathcal{L}_\Gamma u}_{\text{Diffusion}} + \underbrace{\mathcal{N}(u)}_{\text{Reaction}} \quad \text{on } \Gamma$$



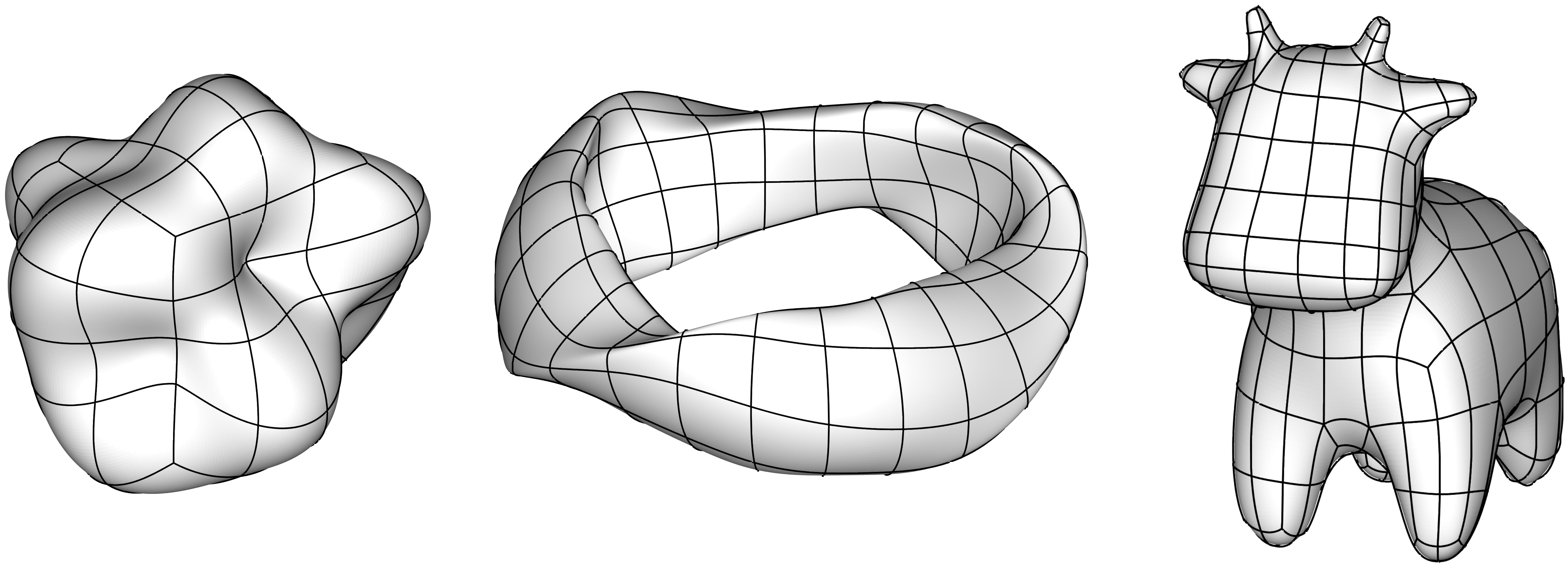


- Reaction and diffusion timescales are often orders of magnitude different.
- Implicit time-stepping can alleviate stability issues (e.g., backward Euler or IMEX-BDF4)

$$\frac{\partial u}{\partial t} = \mathcal{L}_\Gamma u + \mathcal{N}(u) \xrightarrow{\text{(e.g. backward Euler)}} u^{k+1} = \underbrace{(I - \Delta t \mathcal{L}_\Gamma)}_{\substack{\text{Stored in RAM,} \\ \text{very fast apply}}}^{-1} \left(u^k + \Delta t \mathcal{N}(u^k)\right)$$

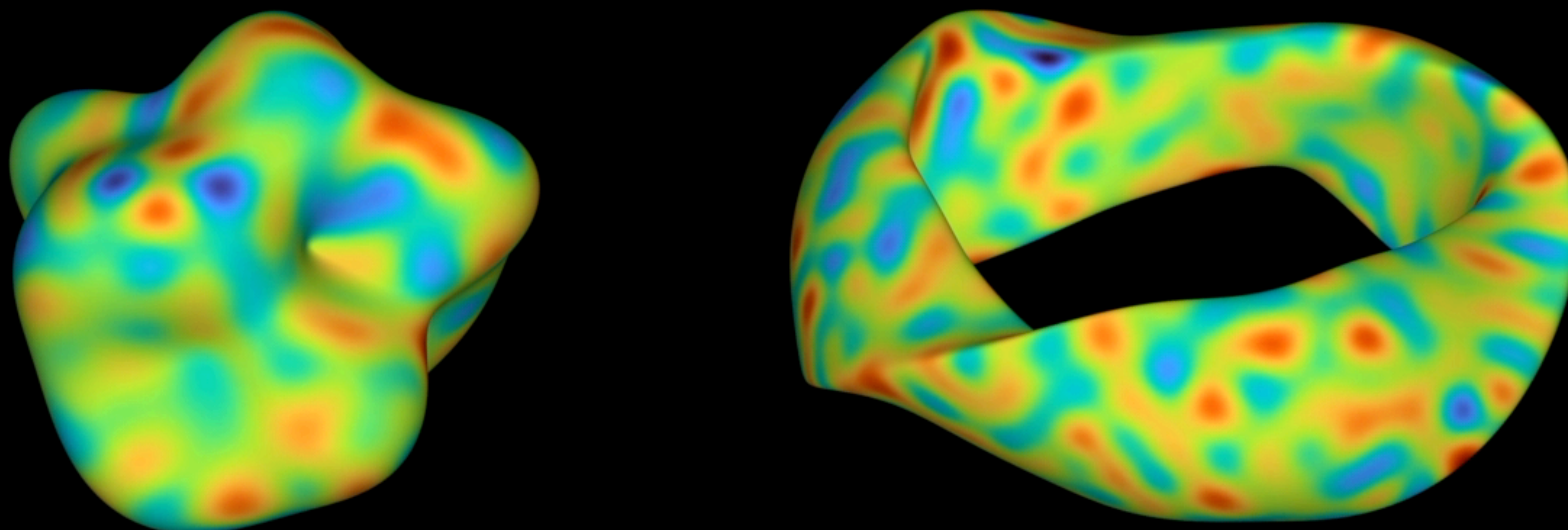
- If geometry, time step, and parameters do not change with time, we can precompute a solver once and reuse it at every step.

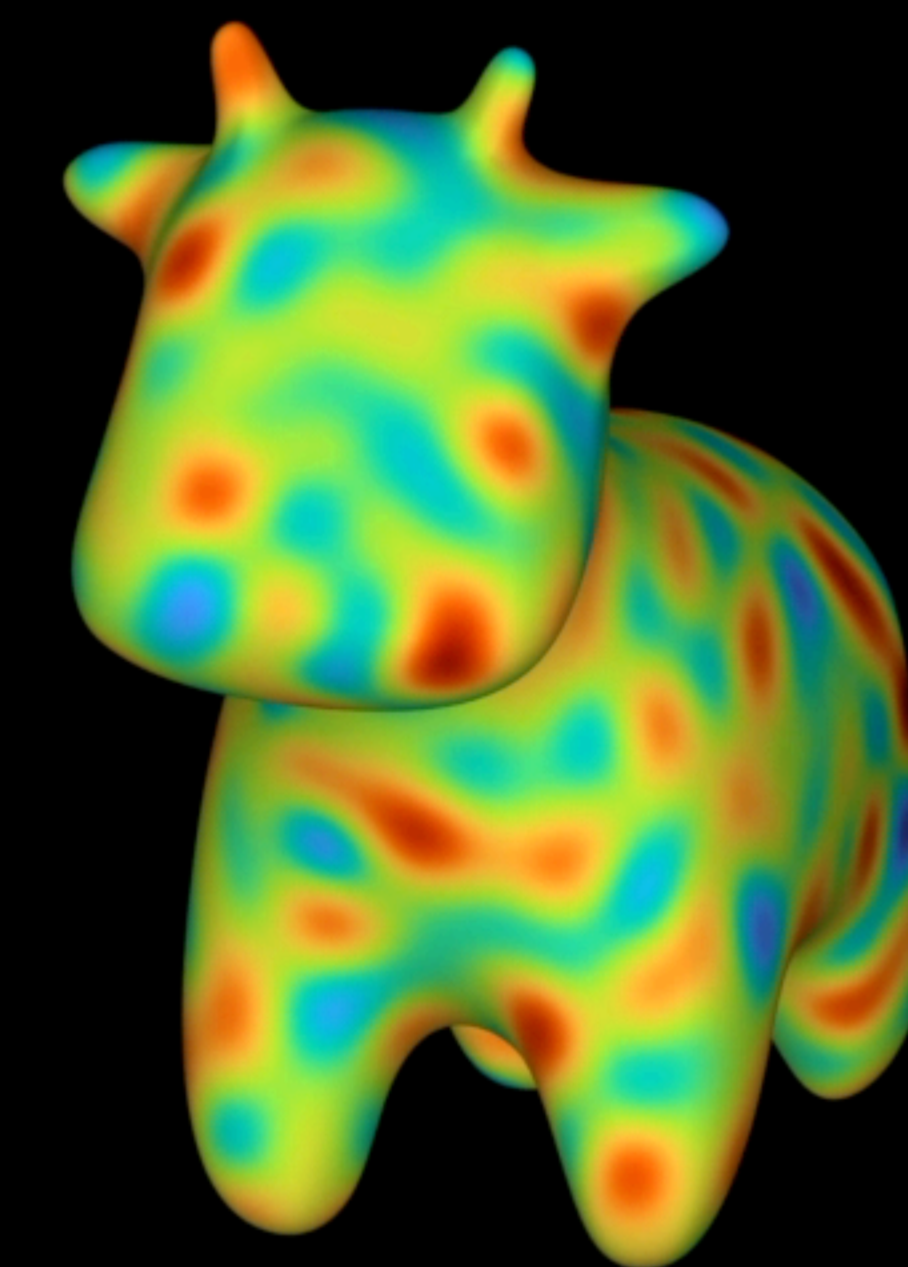
Examples
Reaction–diffusion systems

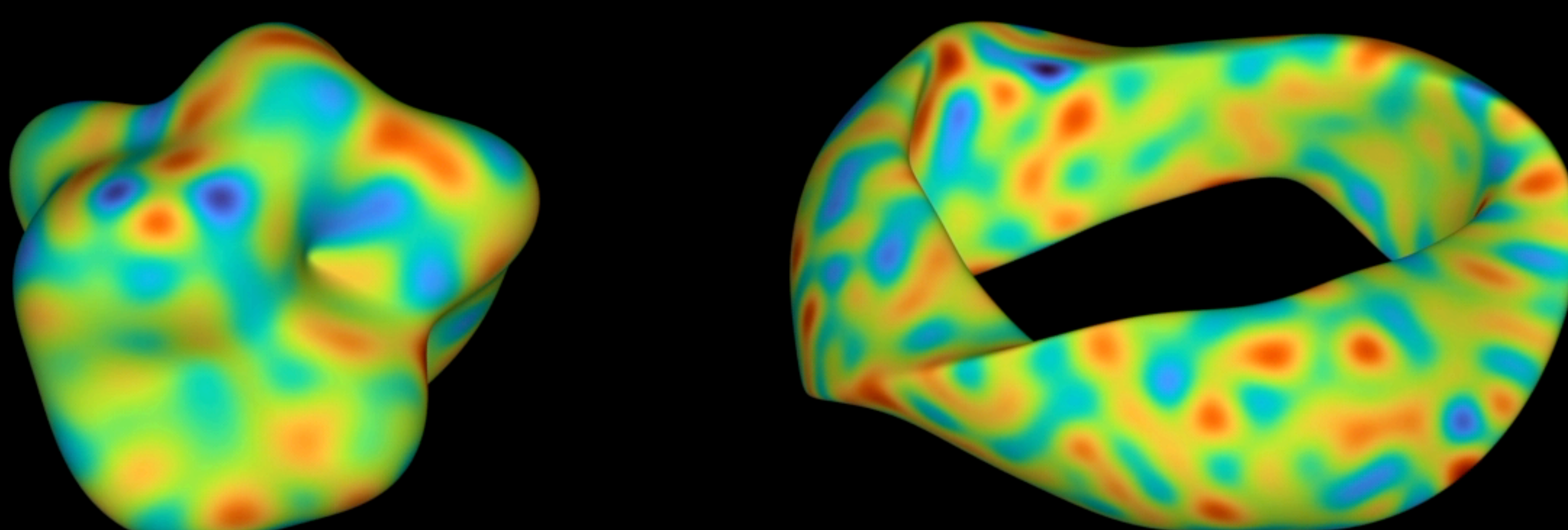


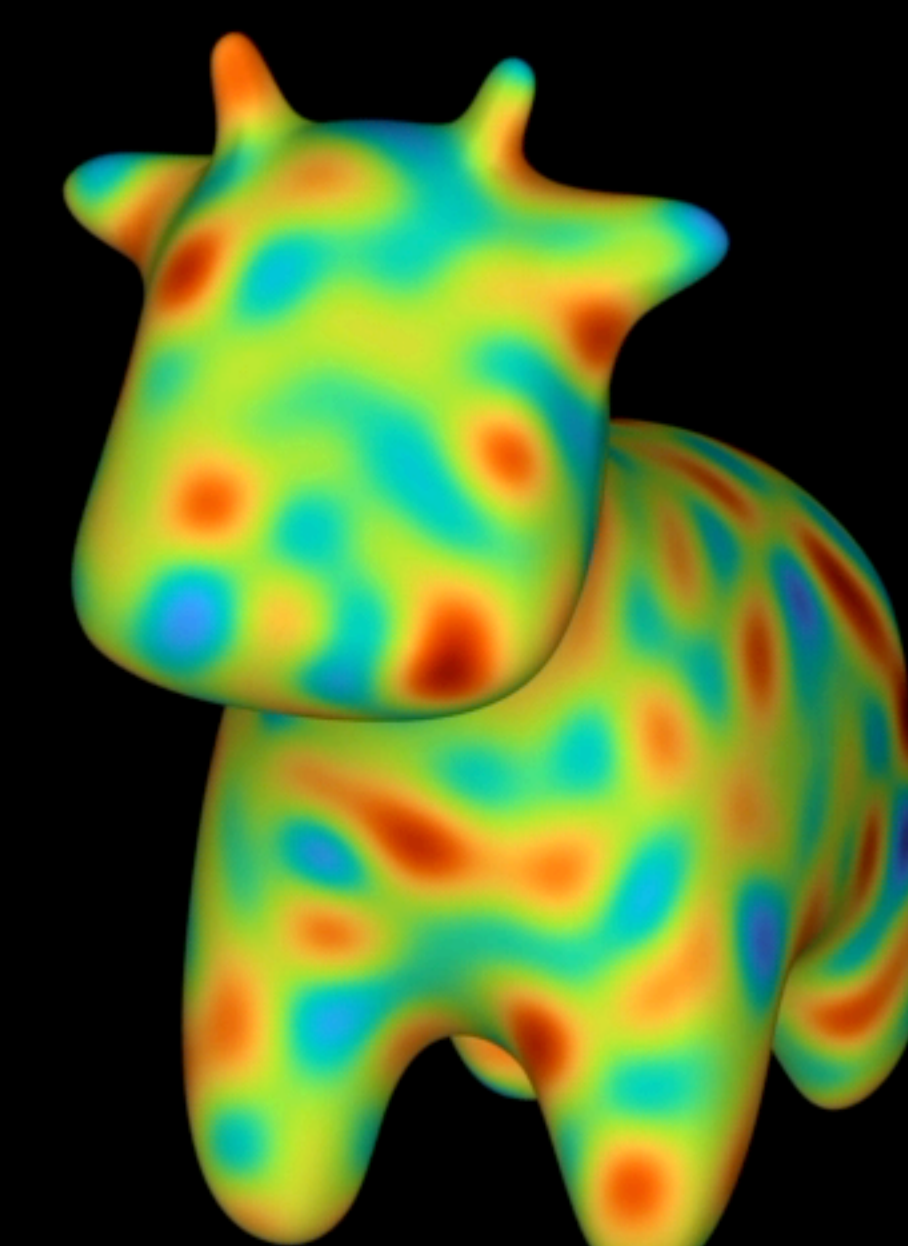
16th order
8-core Intel i9 Macbook Pro
64GB RAM

$$\frac{\partial u}{\partial t} = \delta_u^2 \Delta_\Gamma u + \alpha u(1 - \tau_1 v^2) + v(1 - \tau_2 u)$$
$$\frac{\partial v}{\partial t} = \delta_v^2 \Delta_\Gamma v + \beta v(1 + \alpha \tau_1 uv/\beta) + u(\gamma + \tau_2 v)$$



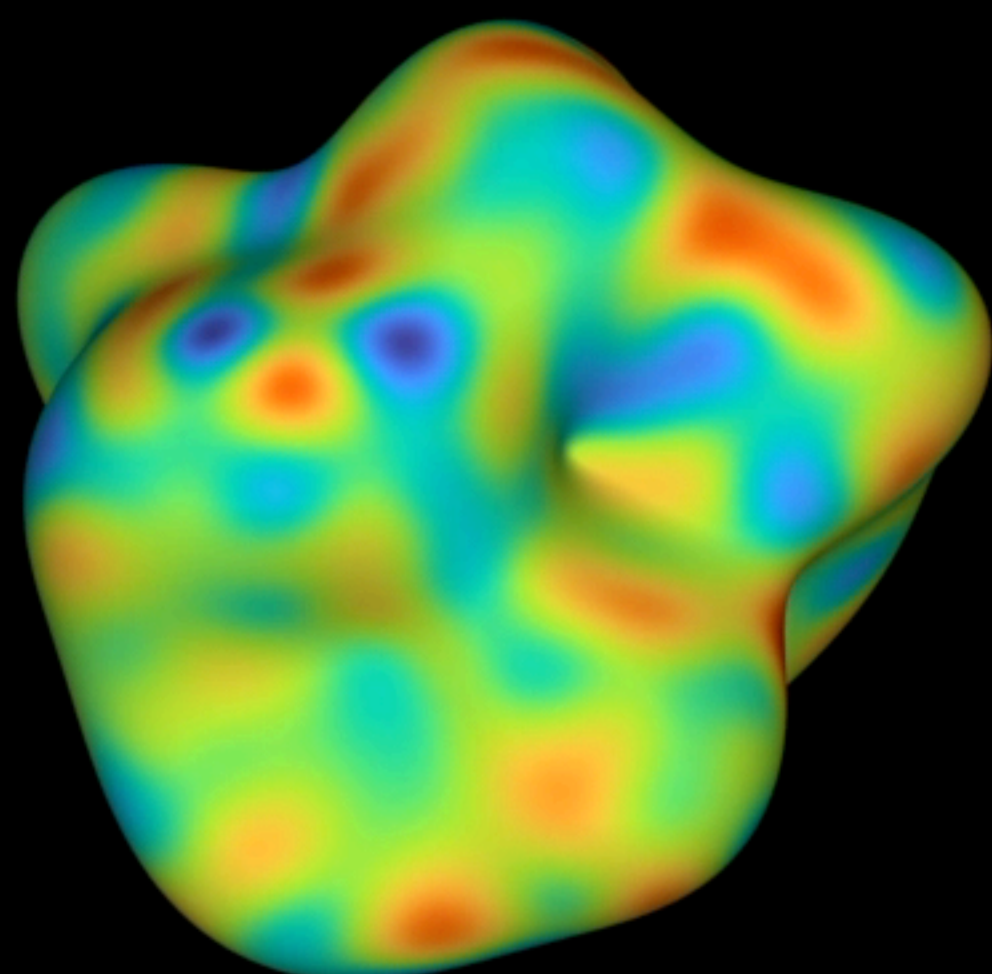
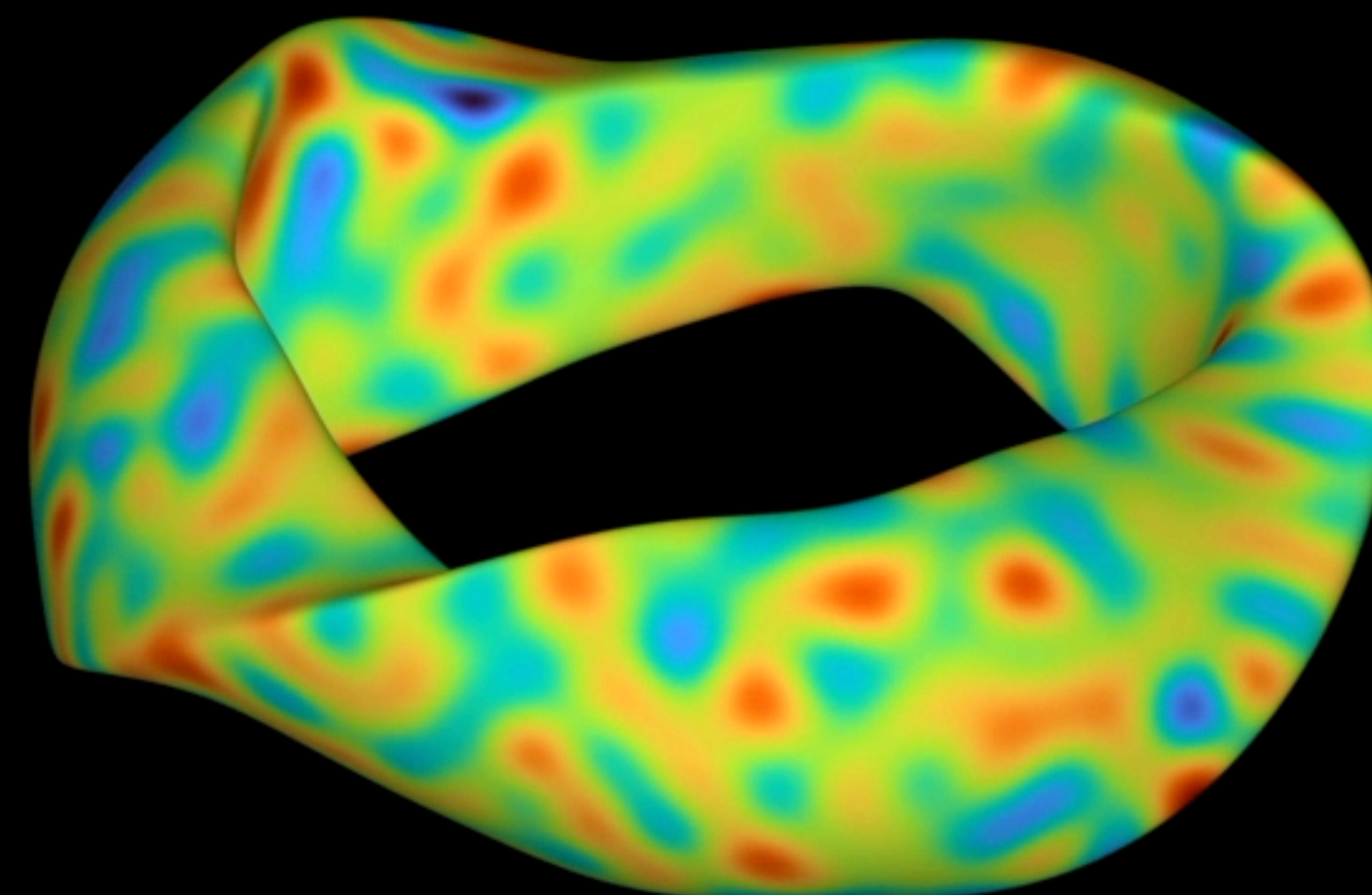
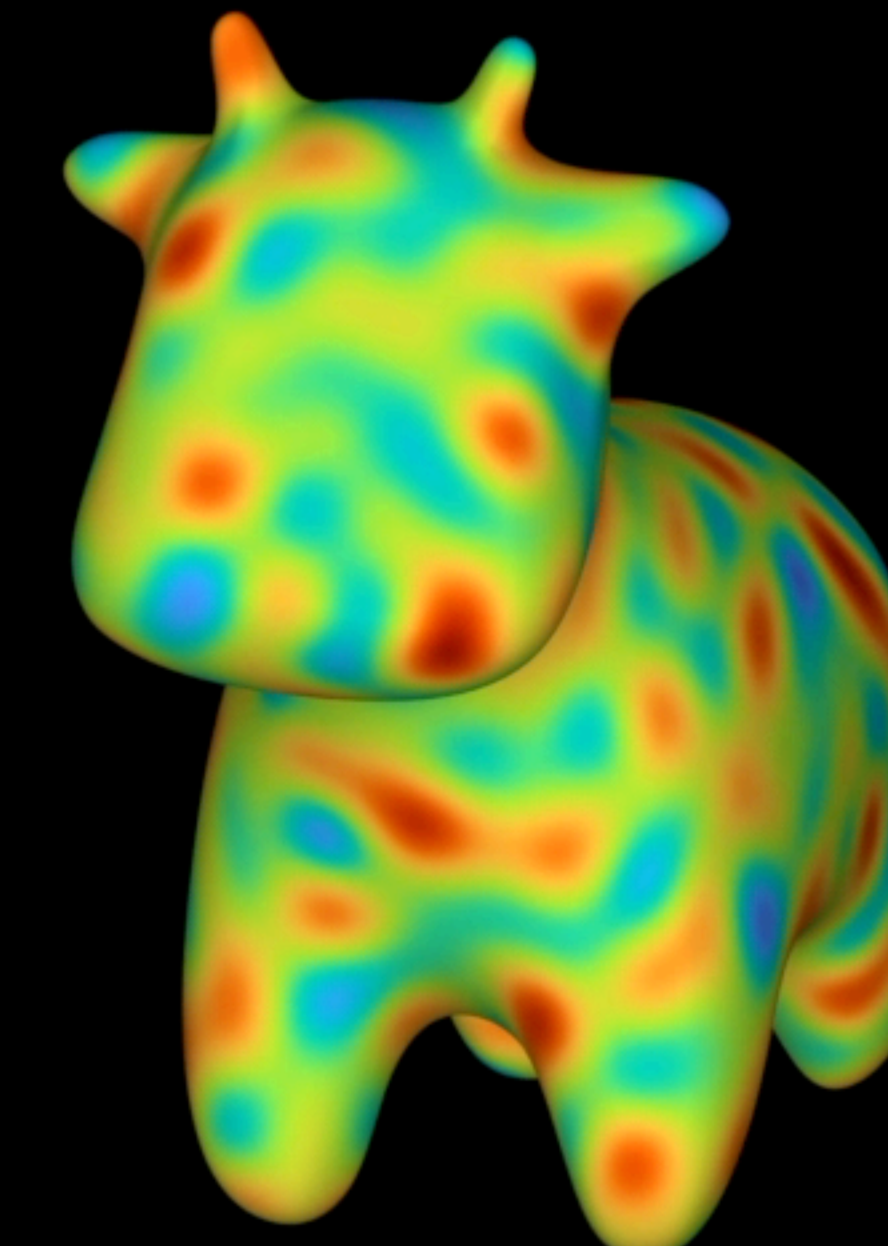






$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u + u - (1 + ci)u|u|^2$$

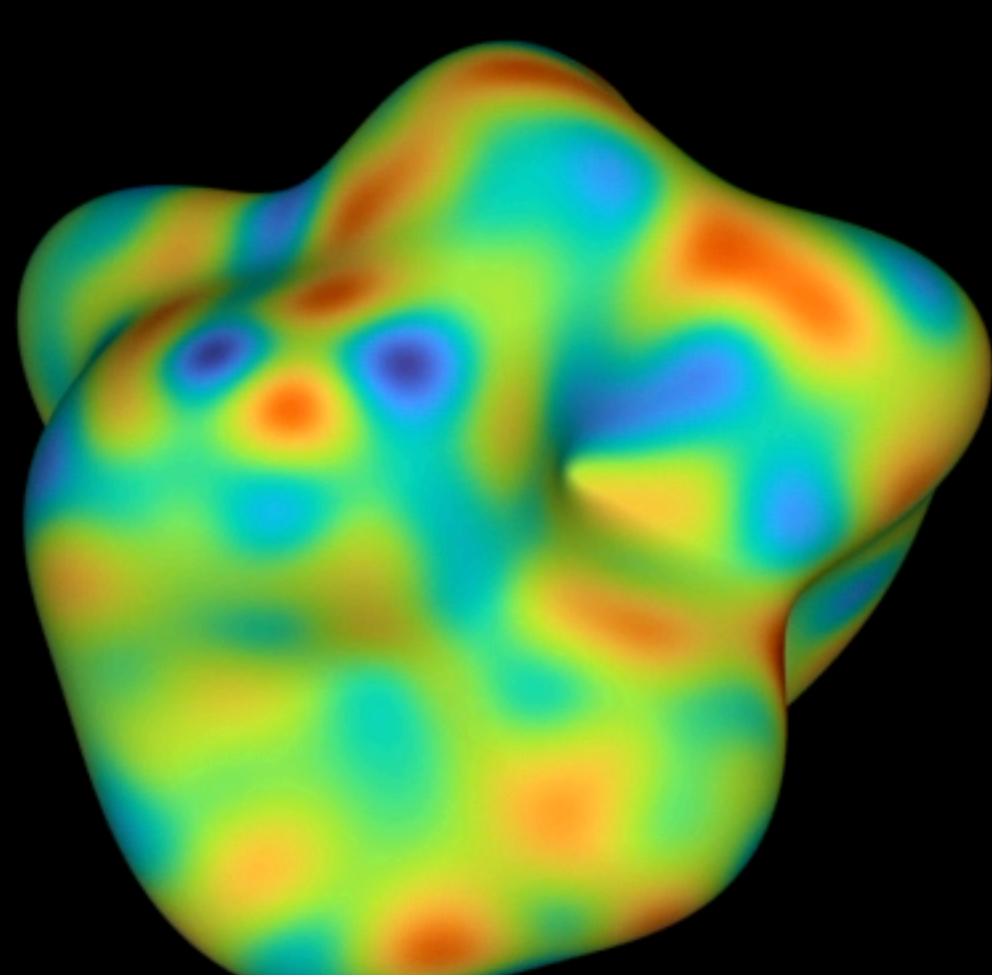
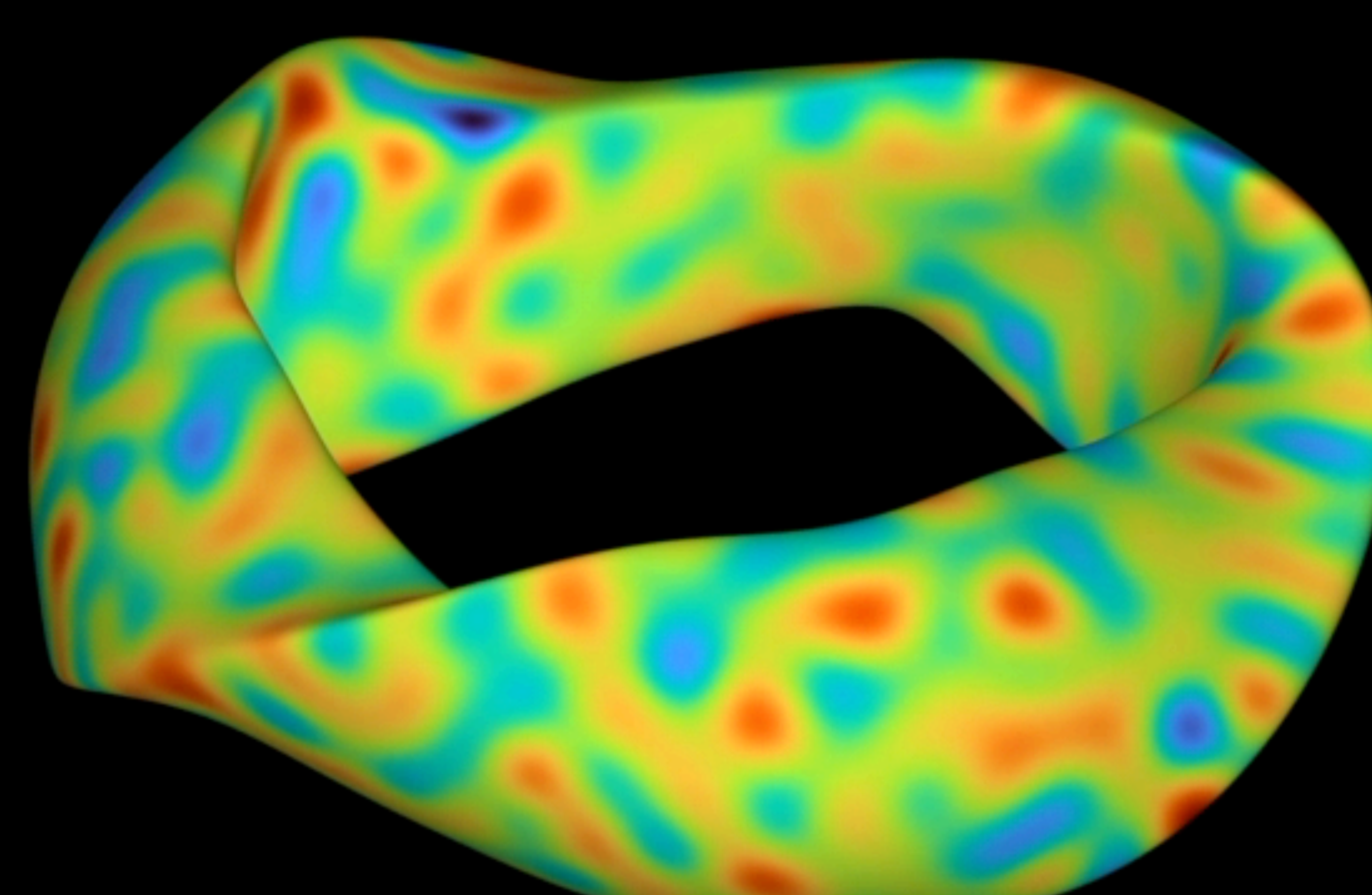
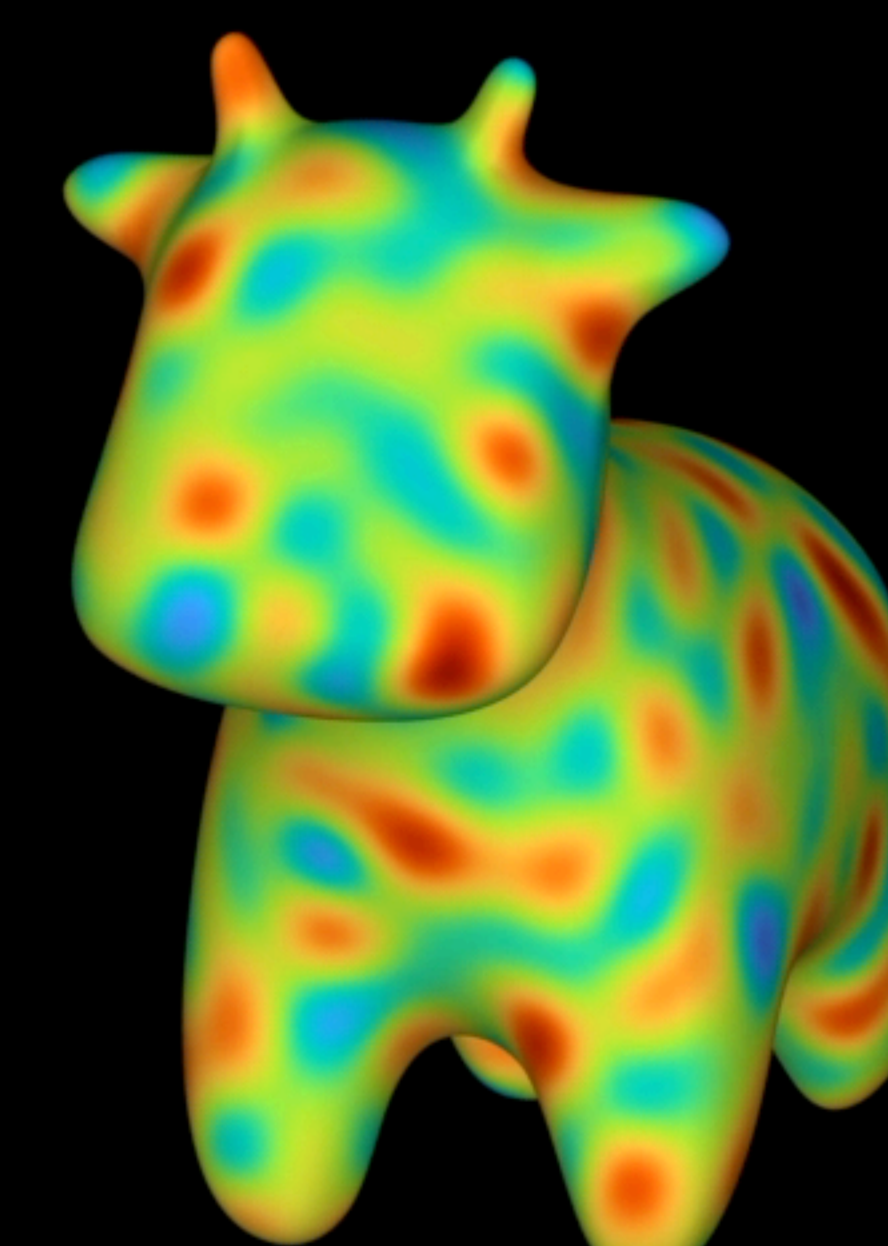
$$\frac{\partial u}{\partial t} = \delta_u^2 \Delta_\Gamma u + \alpha u(1 - \tau_1 v^2) + v(1 - \tau_2 u)$$
$$\frac{\partial v}{\partial t} = \delta_v^2 \Delta_\Gamma v + \beta v(1 + \alpha \tau_1 uv/\beta) + u(\gamma + \tau_2 v)$$

25k unknowns
~20 fps

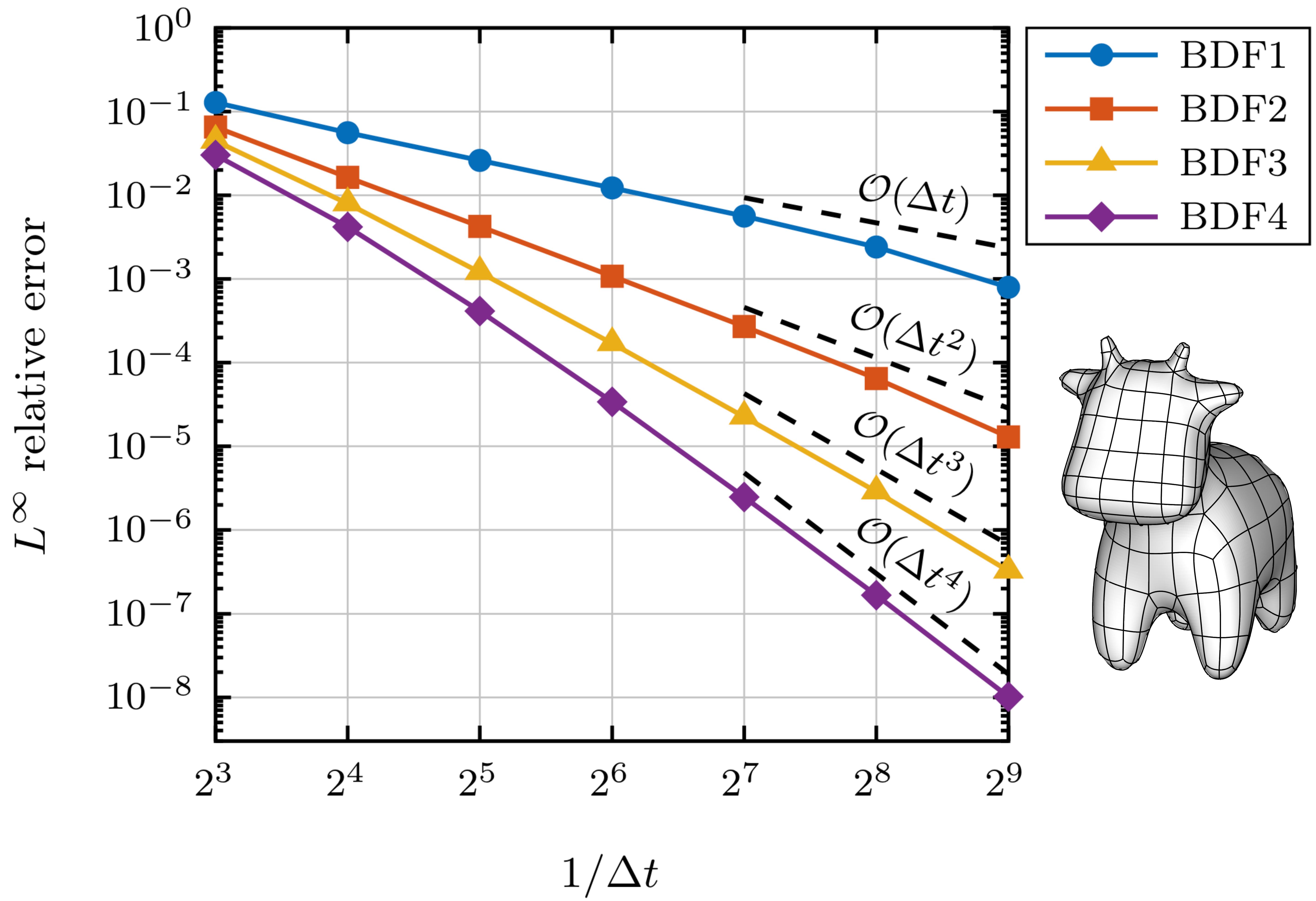
50k unknowns
~13 fps

135k unknowns
~4 fps

$$\frac{\partial u}{\partial t} = \delta^2 \Delta_\Gamma u + u - (1 + ci)u|u|^2$$

IMEX-BDF methods can achieve high-order accuracy in time.



Thanks!

- Miller*, F.*, Muratov, Greengard, & Shvartsman, *Forced and spontaneous symmetry breaking in cell polarization*, Nat. Comput. Phys. (2022)
- Miller, F., Novaga, Shvartsman, & Muratov, *Generation and motion of interfaces in a mass-conserving reaction-diffusion system*, SIAM J. Appl. Dyn. Syst. (2023)
- F., *A high-order fast direct solver for surface PDEs,* to appear in SIAM J. Sci. Comput. (2023)

https://surfacefun.readthedocs.io/

A fast direct solver on surfaces

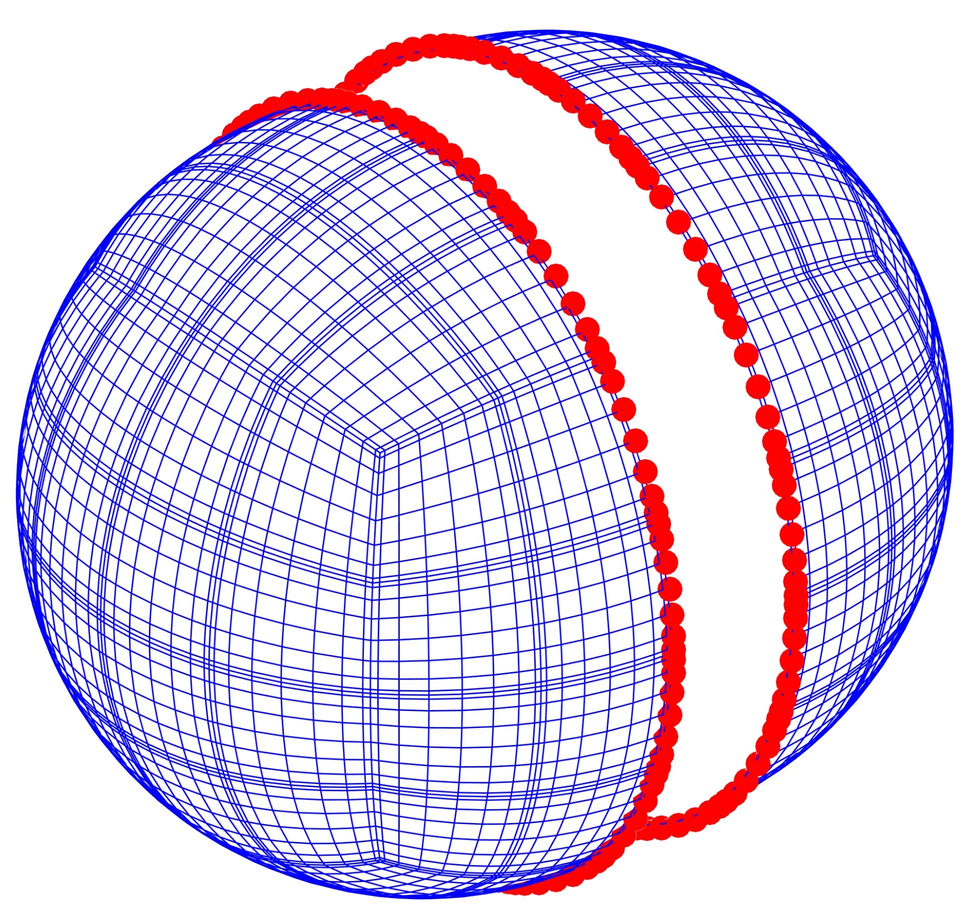
Laplace–Beltrami and rank deficiency

$$\Delta_\Gamma u = f$$

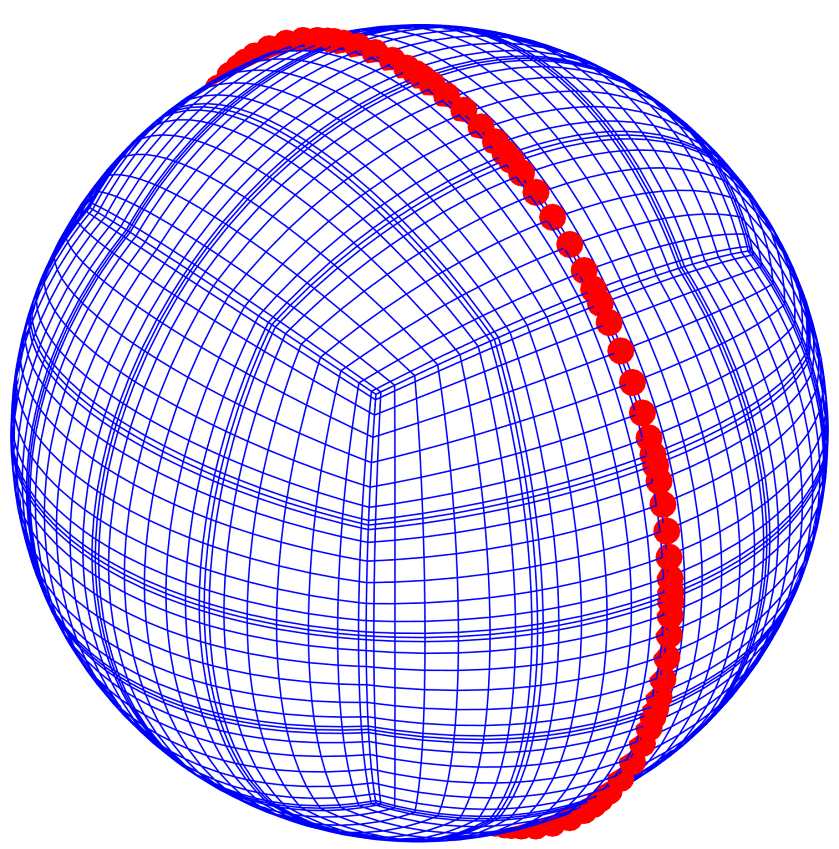
- The Laplace–Beltrami problem on a closed surface is rank-one deficient, but is uniquely solvable under the mean-zero conditions:

$$\int_\Gamma u = 0 \qquad \text{and} \qquad \int_\Gamma f = 0$$

- This rank deficiency is only seen in the final gluing:



$$\xrightarrow{\dim\left(\operatorname{null}\left(DtN_1^{\text{glue}} + DtN_2^{\text{glue}}\right)\right) = 1}$$



- We add the mean-zero constraint to fix the rank deficiency at the top level:

$$\dim\left(\operatorname{null}\left(DtN_1^{\text{glue}} + DtN_2^{\text{glue}} + \mathbf{q}\mathbf{q}^T\right)\right) = 0$$