Interpolated compressed inverse preconditioning

Fast and accurate simulation of close-to-touching discs in Stokes flow

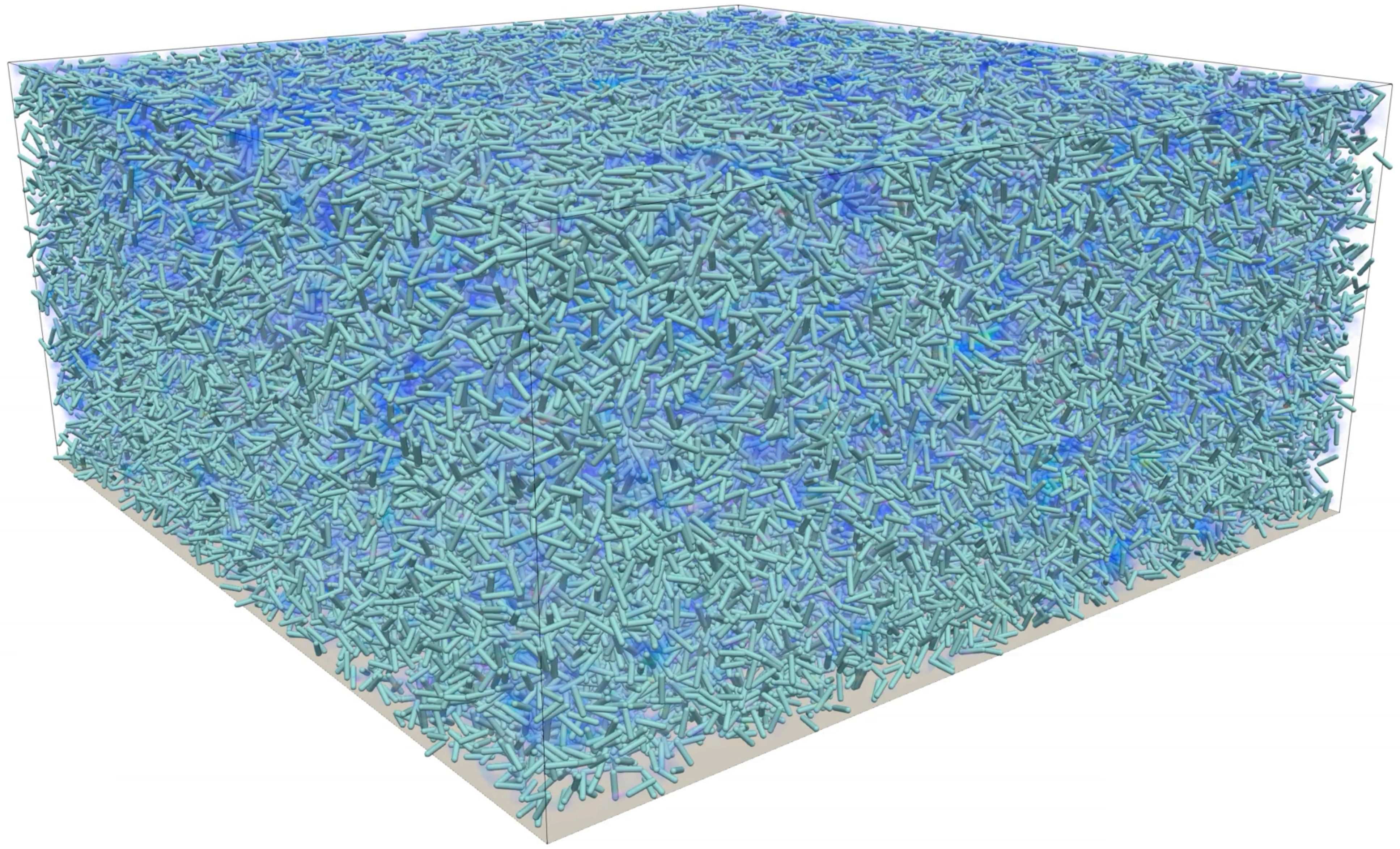
Mariana Martínez Aguilar
EPFL

Dan Fortunato
Flatiron Institute

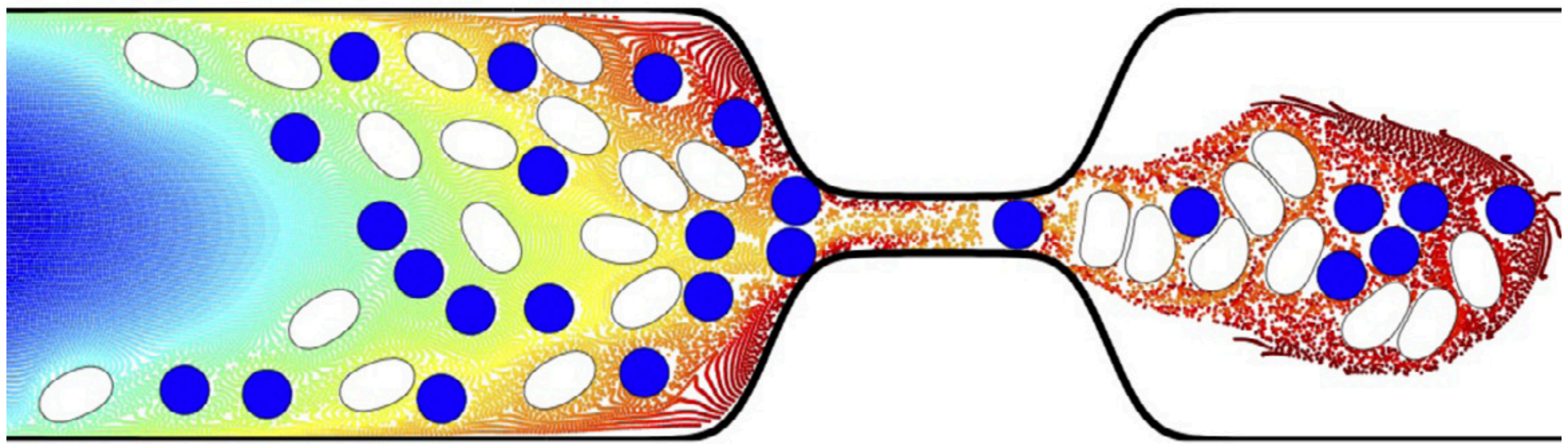
Dhairya Malhotra
Flatiron Institute

Introduction
Particulate flows

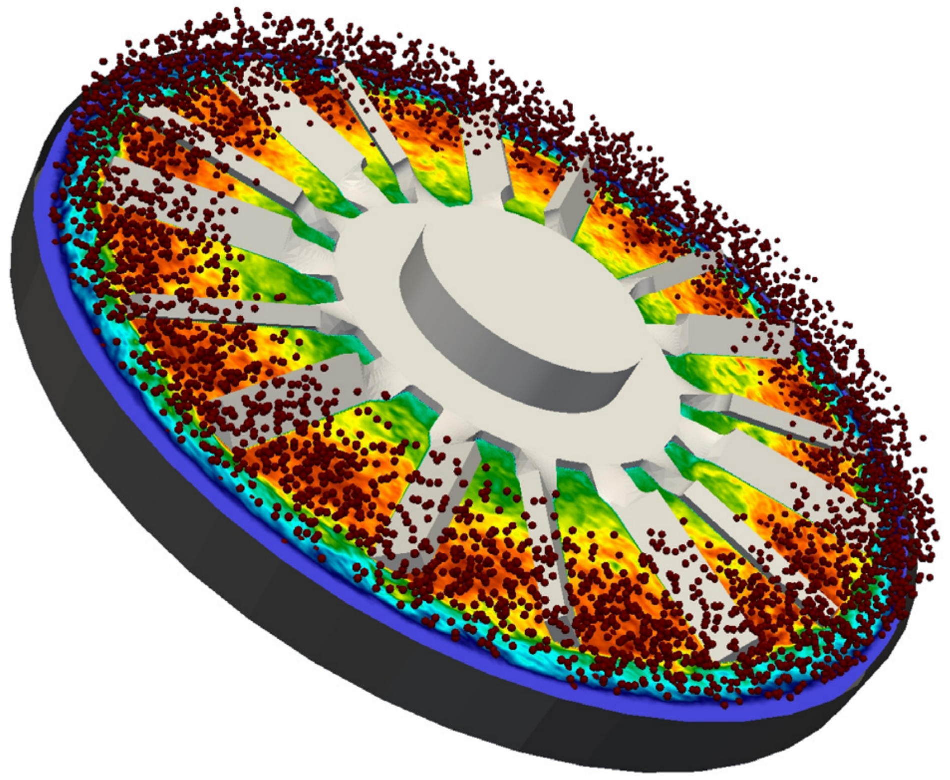
Active swimmers [Yan et al, 2020]



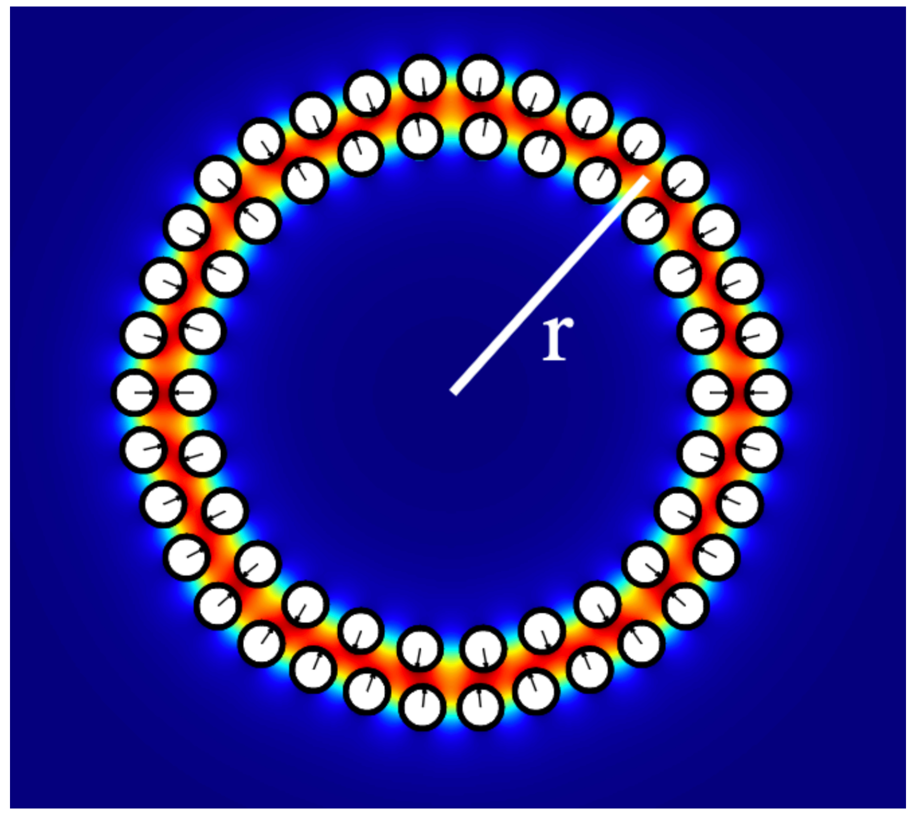
Vesicle flows [Lu et al, 2017]



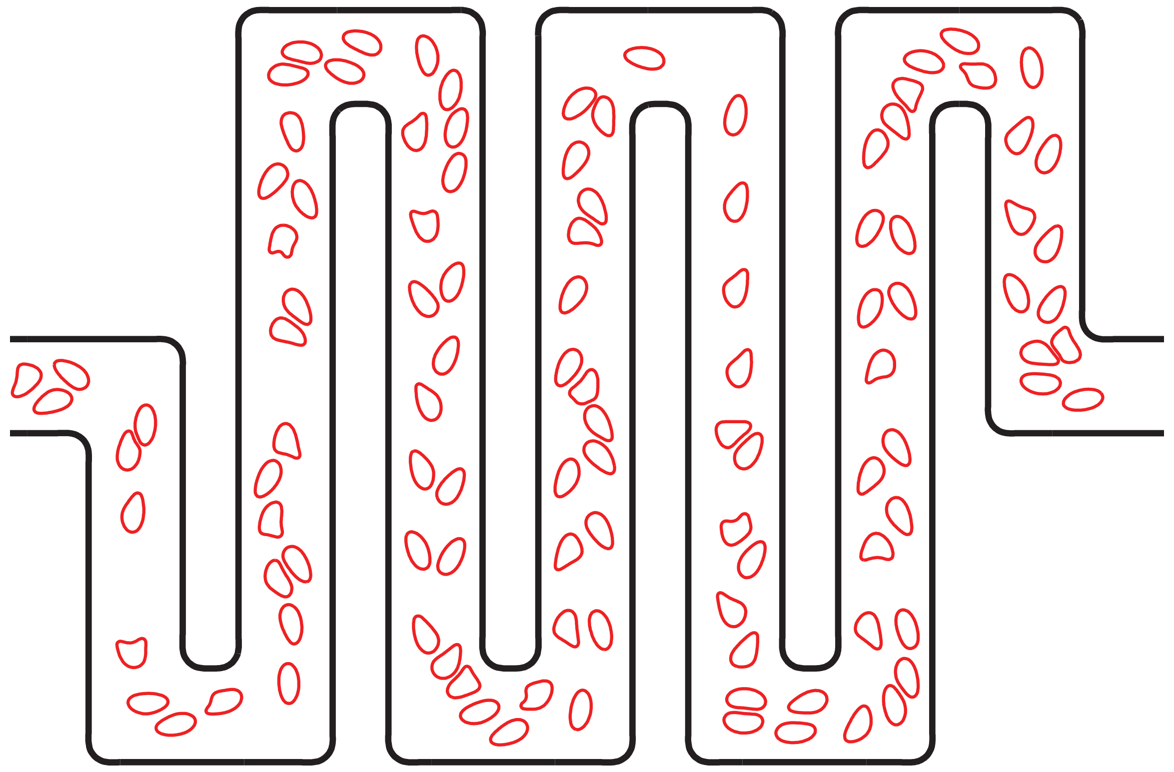
Industrial processes [Krause et al, 2021]



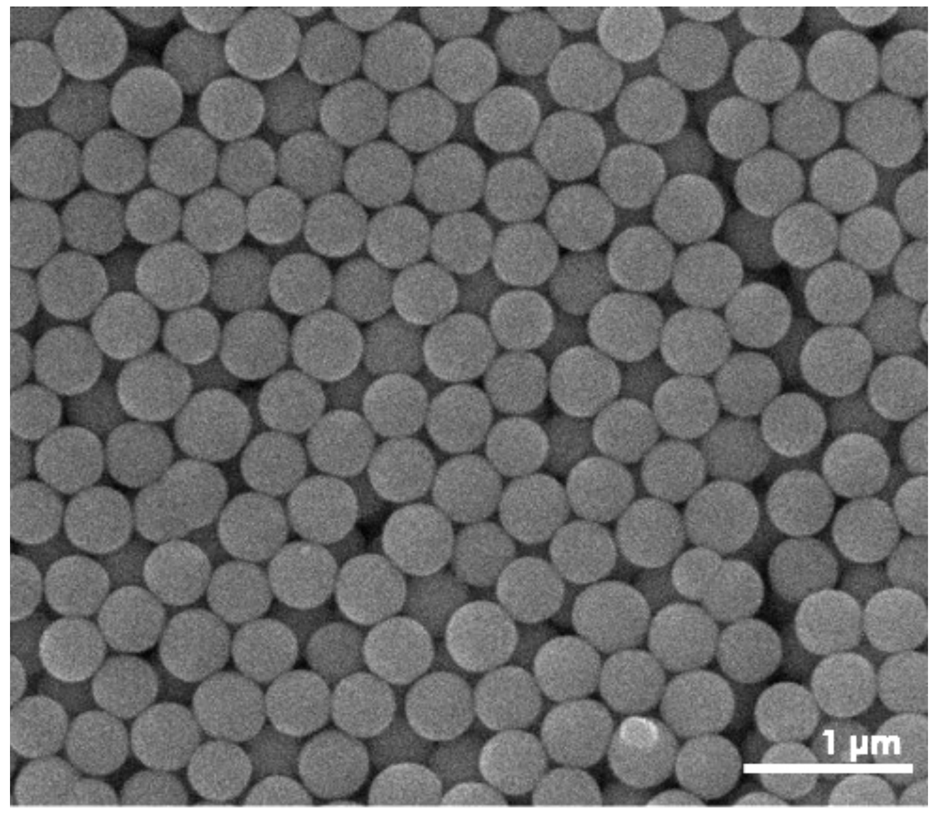
Janus particles [Fu et al, 2021]



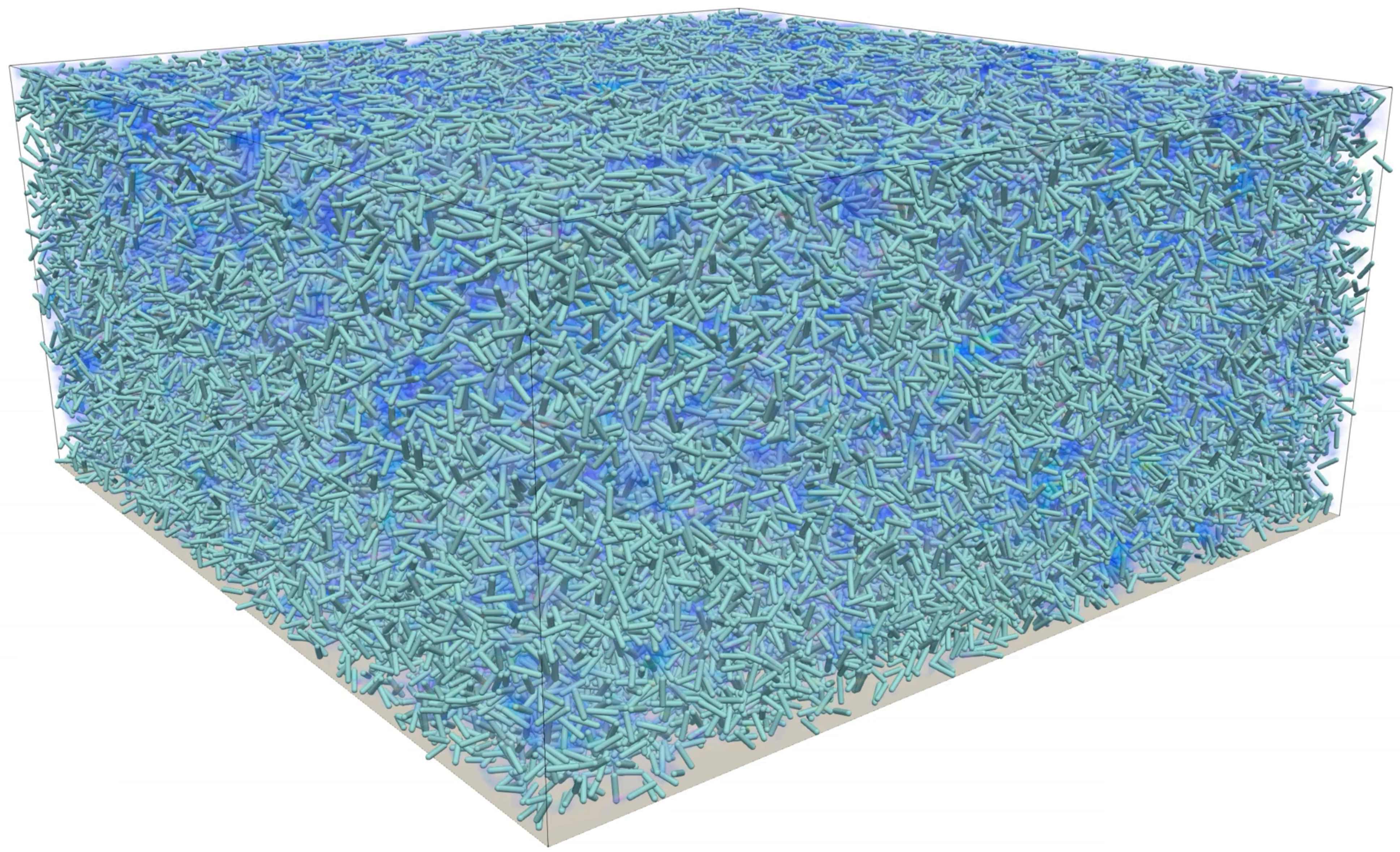
Microfluidics [Marple et al, 2015]



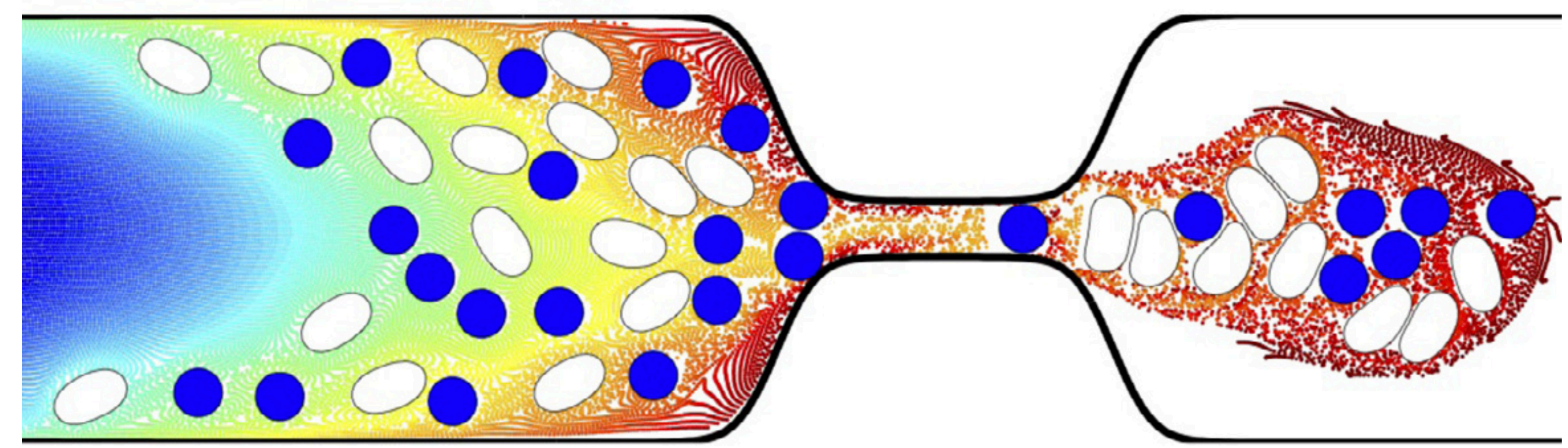
Colloidal suspensions [Wikipedia]



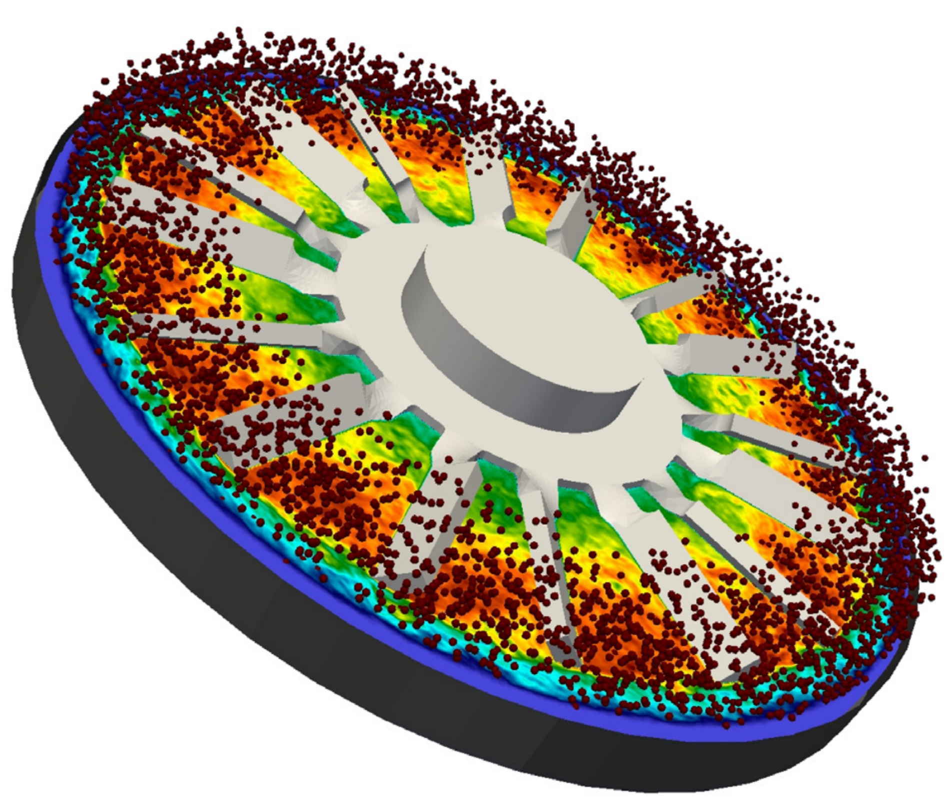
Active swimmers [Yan et al, 2020]



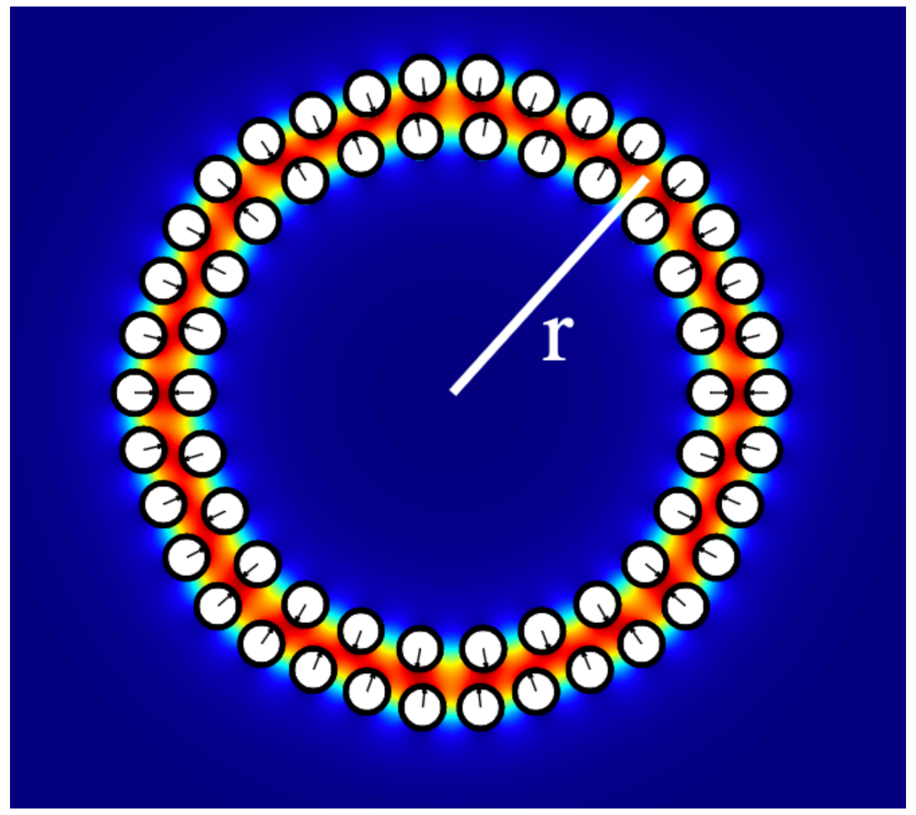
Vesicle flows [Lu et al, 2017]



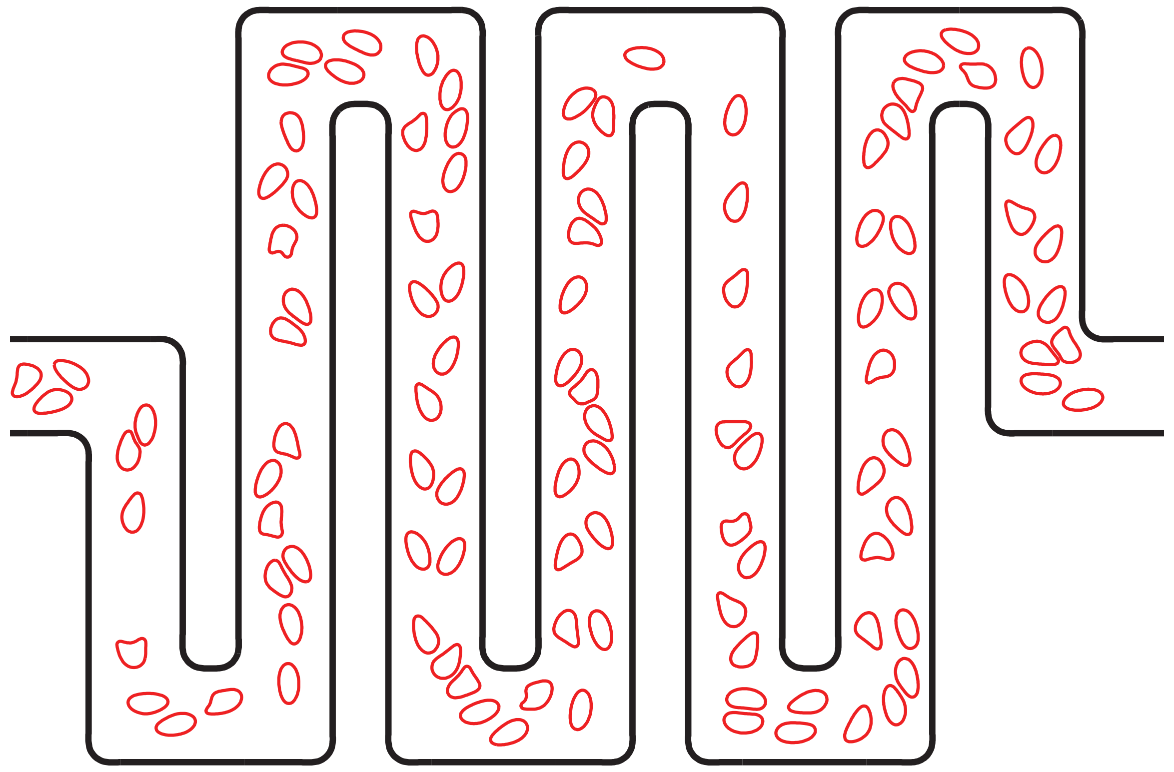
Industrial processes [Krause et al, 2021]



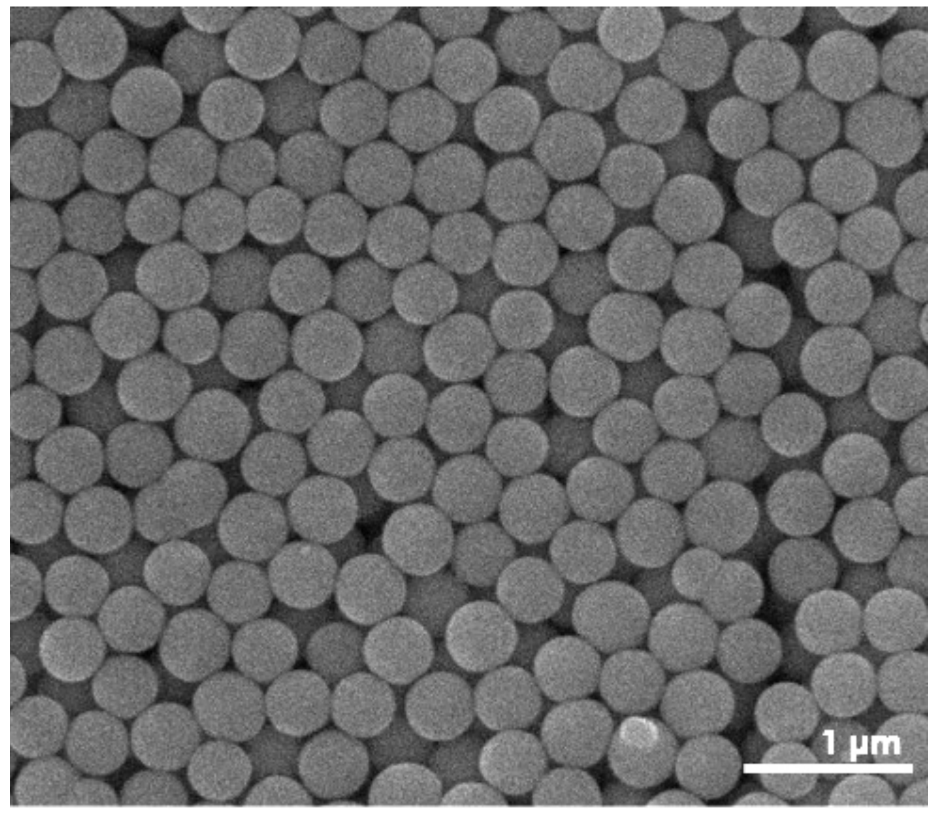
Janus particles [Fu et al, 2021]



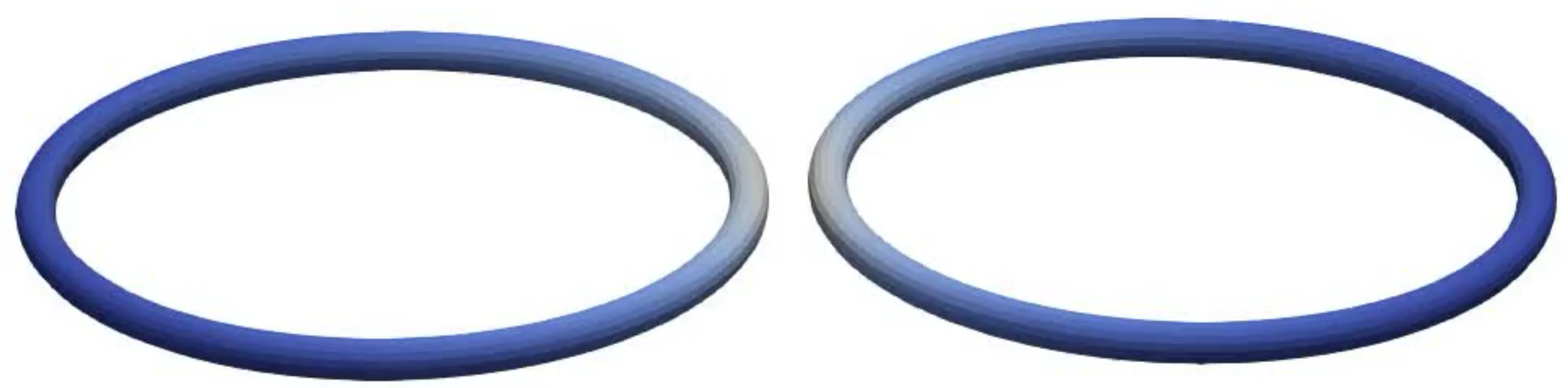
Microfluidics [Marple et al, 2015]

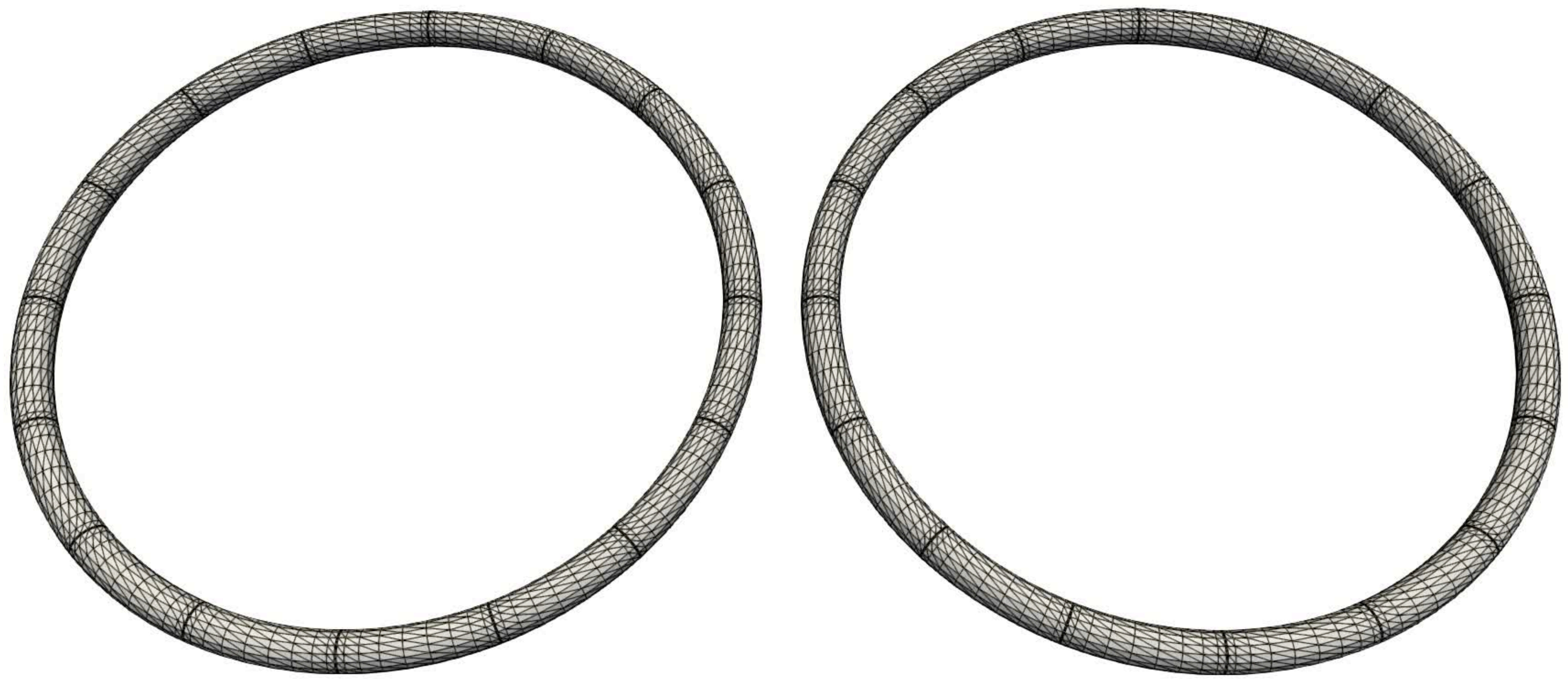


Colloidal suspensions [Wikipedia]



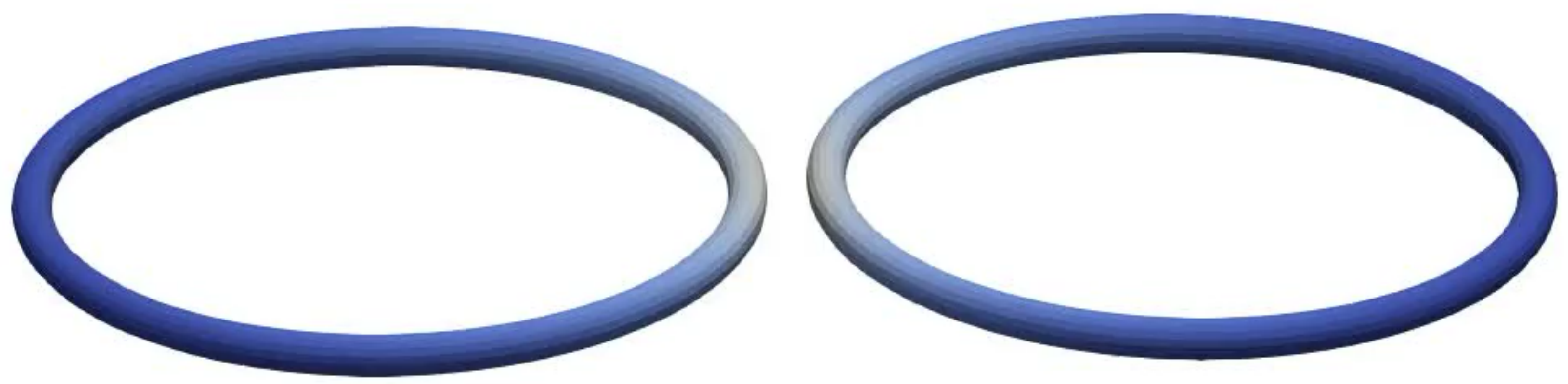
Example: Sedimenting rings in Stokes flow [Malhotra & Barnett, 2024]

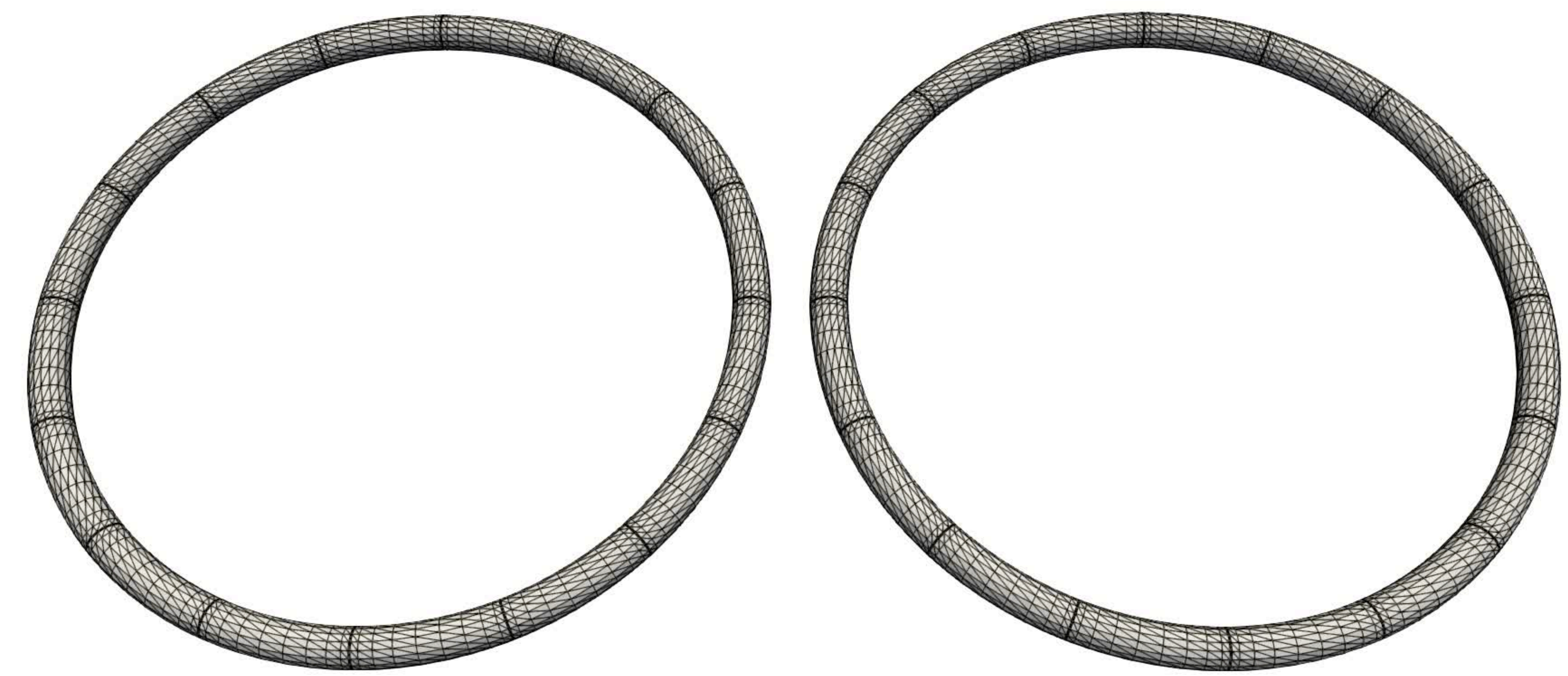




Figures from [Malhotra & Barnett, 2024]

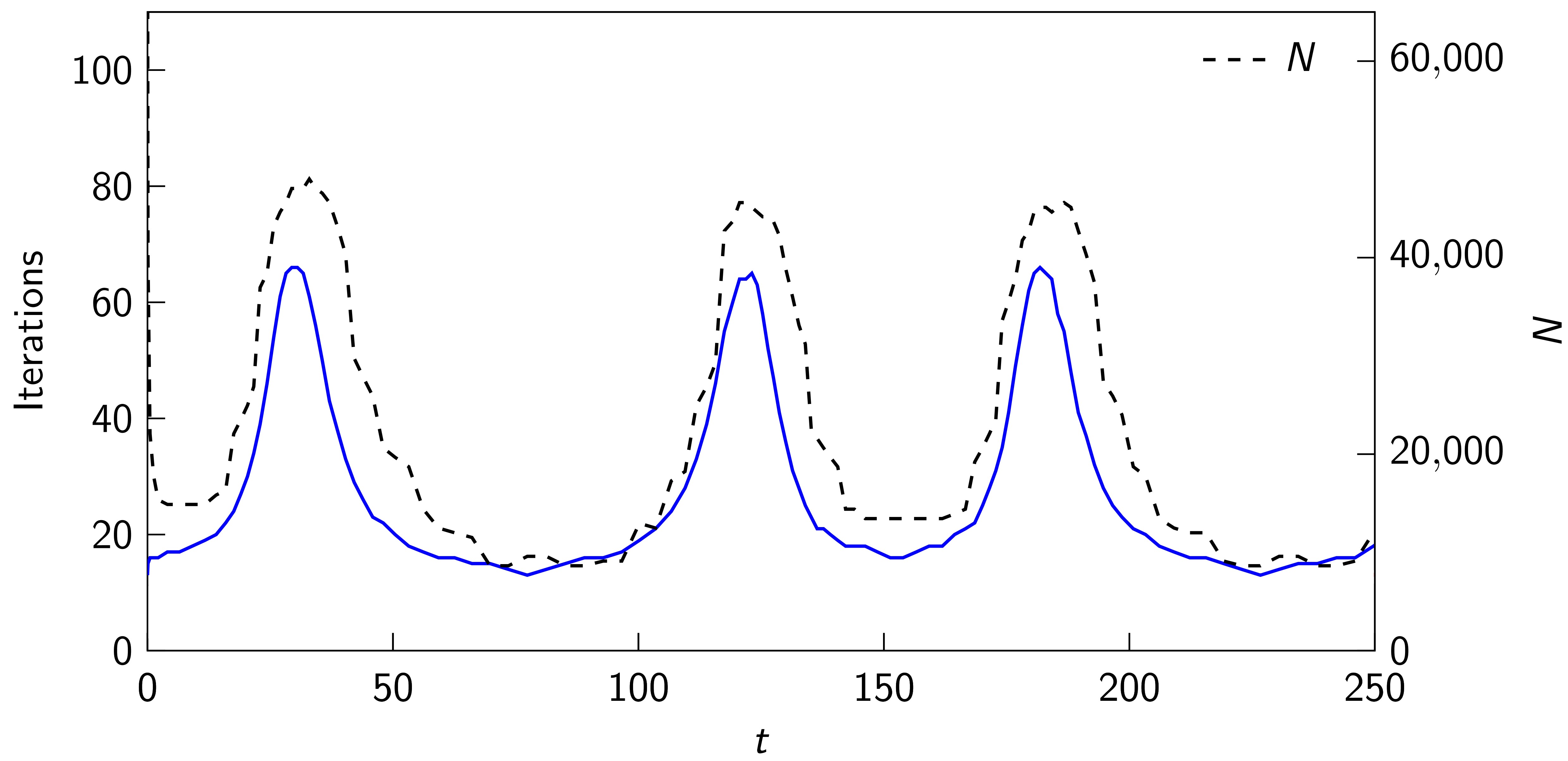
Example: Sedimenting rings in Stokes flow [Malhotra & Barnett, 2024]

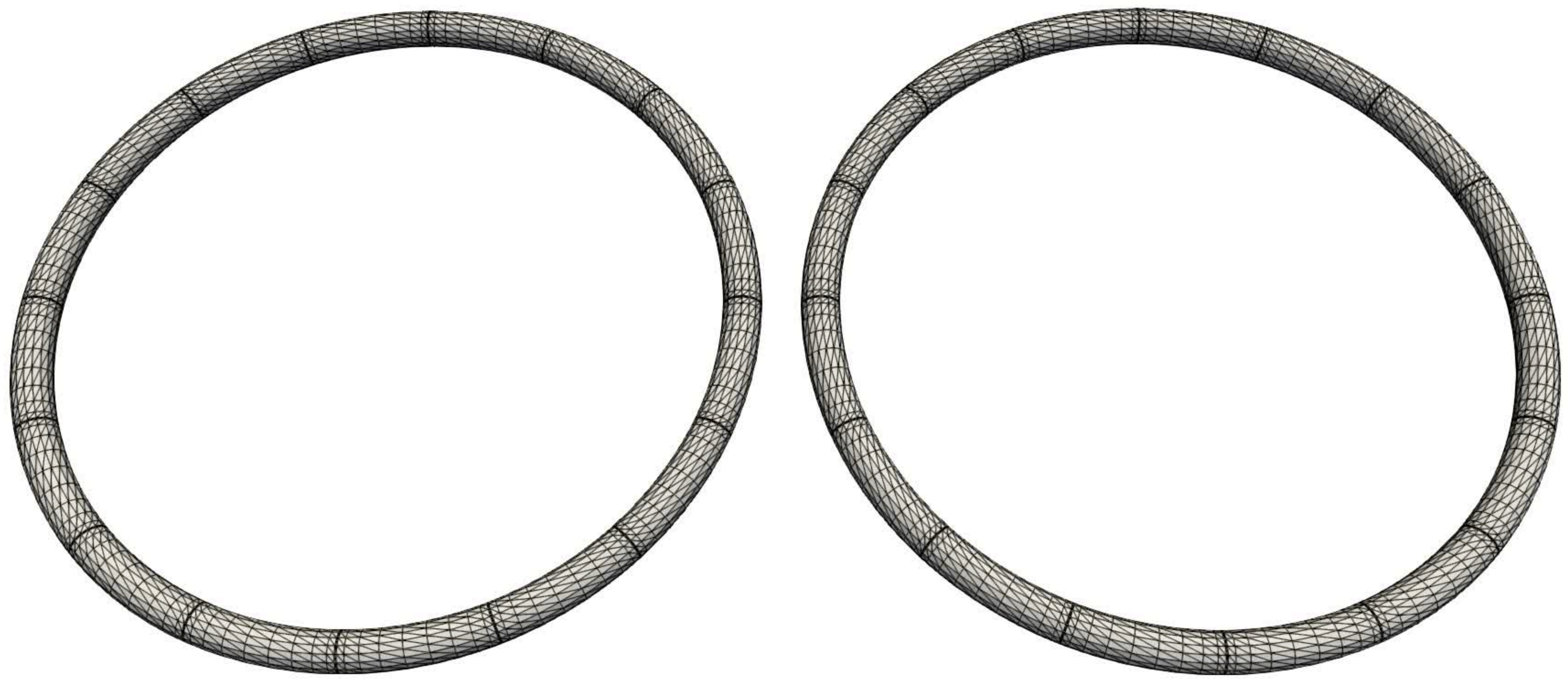




Figures from [Malhotra & Barnett, 2024]

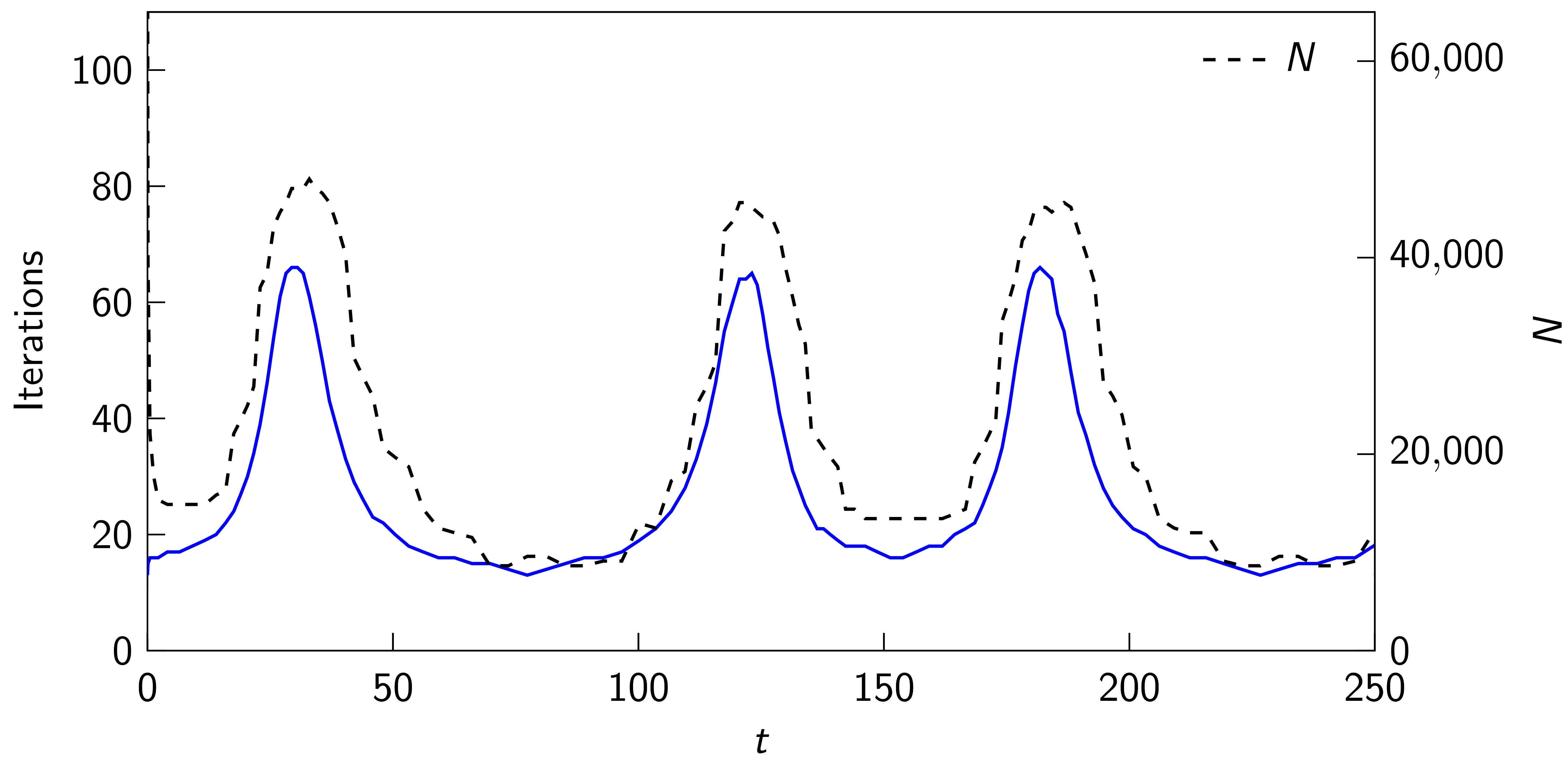
Example: Sedimenting rings in Stokes flow [Malhotra & Barnett, 2024]

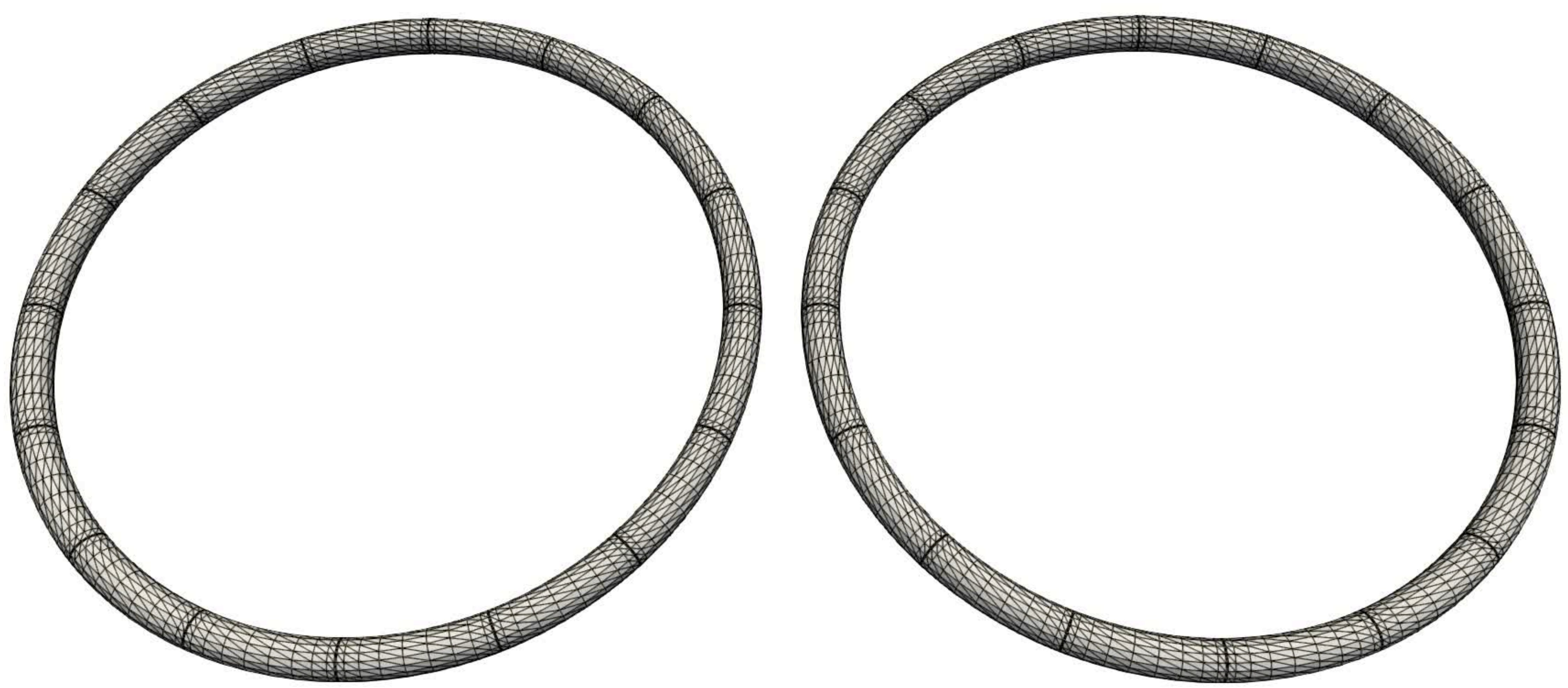




Figures from [Malhotra & Barnett, 2024]

Example: Sedimenting rings in Stokes flow [Malhotra & Barnett, 2024]





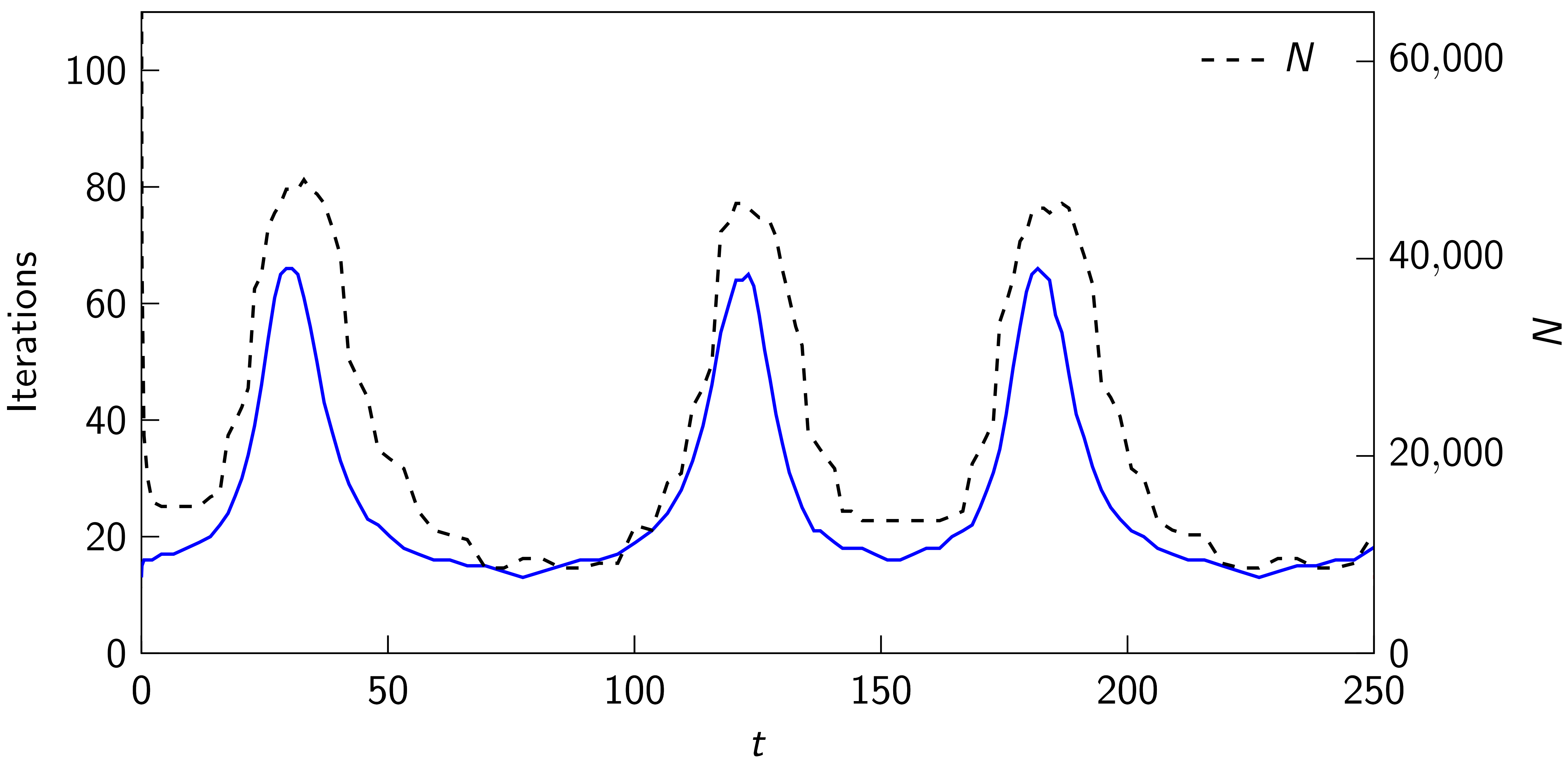
Close interactions:

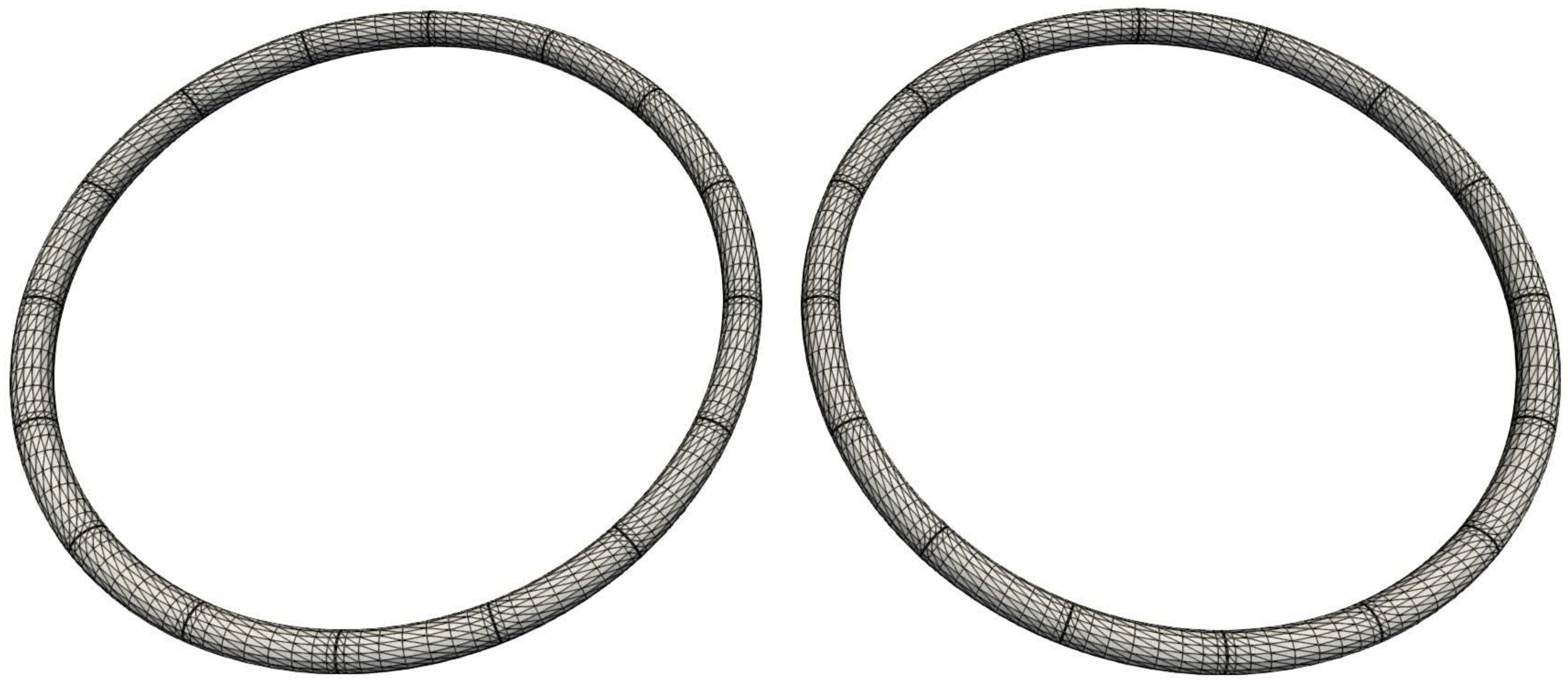
~5× more GMRES iterations
~5× more unknowns
× ~5× smaller Δt

>125× **slower**!

Figures from [Malhotra & Barnett, 2024]

<u>Example</u>: Sedimenting rings in Stokes flow [Malhotra & Barnett, 2024]





Close interactions:

~5× more GMRES iterations
~5× more unknowns
× ~5× smaller Δt

>125× **slower**!

Figures from [Malhotra & Barnett, 2024]

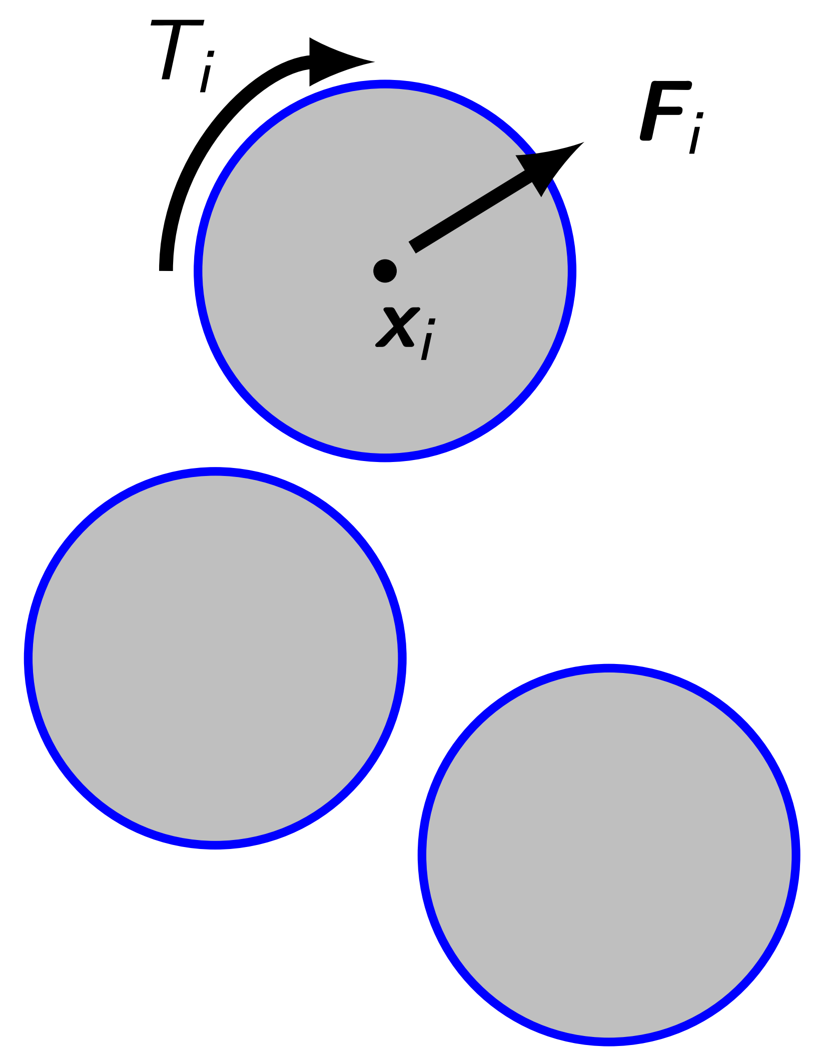
Consider a collection of identical rigid discs $\Omega = \bigcup_{i=1}^{N_\Omega} \Omega_i$ immersed in a Stokesian fluid:

$$\begin{aligned} \Delta \boldsymbol{u} - \nabla p &= 0 \quad \text{in } \mathbb{R}^2 \setminus \Omega \\ \nabla \cdot \boldsymbol{u} &= 0 \quad \text{in } \mathbb{R}^2 \setminus \Omega \\ \boldsymbol{u} &\to 0 \quad \text{as } \boldsymbol{x} \to \infty \end{aligned}$$

$$\int_{\partial\Omega_i} \boldsymbol{f}_i = -\boldsymbol{F}_i$$

$$\int_{\partial\Omega_i} \boldsymbol{f}_i \cdot (\boldsymbol{x} - \boldsymbol{x}_i)^\perp = -T_i$$

fluid traction



along with velocity boundary conditions

$$\boldsymbol{u} = \boldsymbol{V} + \boldsymbol{u}_s \quad \text{on } \partial\Omega$$

where $\boldsymbol{V}(\boldsymbol{x}) = \boldsymbol{v}_i + \boldsymbol{\omega}_i \times (\boldsymbol{x} - \boldsymbol{x}_i)$ is the rigid body velocity of disc i and $\boldsymbol{u}_s$ is a given slip velocity.

Stokes mobility problem
Setup

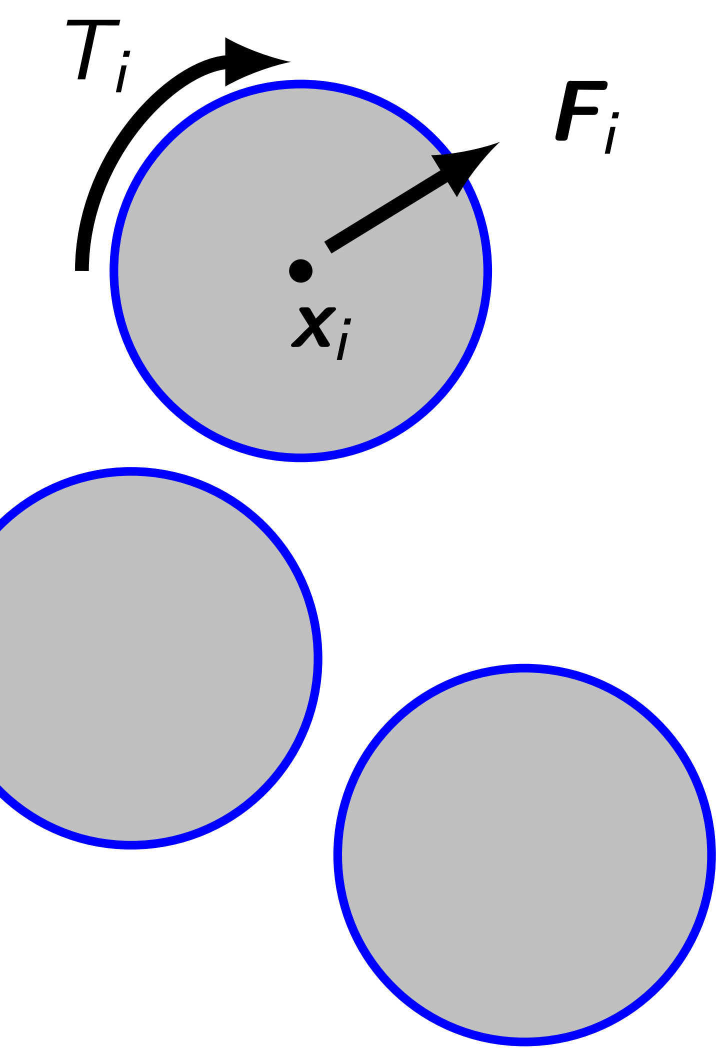
Consider a collection of identical rigid discs $\Omega = \bigcup_{i=1}^{N_\Omega} \Omega_i$ immersed in a Stokesian fluid:

$$\begin{aligned} \Delta \boldsymbol{u} - \nabla p &= 0 && \text{in } \mathbb{R}^2 \setminus \Omega \\ \nabla \cdot \boldsymbol{u} &= 0 && \text{in } \mathbb{R}^2 \setminus \Omega \\ \boldsymbol{u} &\to 0 && \text{as } \boldsymbol{x} \to \infty \end{aligned}$$

$$\int_{\partial\Omega_i} \boldsymbol{f}_i = -\boldsymbol{F}_i$$

$$\int_{\partial\Omega_i} \boldsymbol{f}_i \cdot (\boldsymbol{x} - \boldsymbol{x}_i)^\perp = -T_i$$

fluid traction



along with velocity boundary conditions

$$\boldsymbol{u} = \boldsymbol{V} + \boldsymbol{u}_s \quad \text{on } \partial\Omega$$

where $\boldsymbol{V}(\boldsymbol{x}) = \boldsymbol{v}_i + \omega_i \times (\boldsymbol{x} - \boldsymbol{x}_i)$ is the rigid body velocity of disc i and $\boldsymbol{u}_s$ is a given slip velocity.

Stokes mobility problem: Given prescribed forces $\boldsymbol{F}_i$ and torques T_i on each disc, find the unknown rigid body velocities $(\boldsymbol{v}_i, \omega_i)$.

Stokes mobility problem
Boundary integral formulation

Represent fluid velocity using Stokes single- and double-layer potentials:

$$S(\boldsymbol{r}) = \frac{1}{4\pi}\left(\log\frac{1}{|\boldsymbol{r}|}\boldsymbol{I} + \frac{\boldsymbol{r}\otimes\boldsymbol{r}}{|\boldsymbol{r}|^2}\right)$$

$$D(\boldsymbol{r}) = \frac{1}{\pi}\frac{\boldsymbol{r}\otimes\boldsymbol{r}}{|\boldsymbol{r}|^4}(\boldsymbol{r}\cdot\boldsymbol{n})$$

$$\boldsymbol{u}(\boldsymbol{x}) = \int_{\partial\Omega} S(\boldsymbol{x}-\boldsymbol{y})\underbrace{\boldsymbol{\nu}}_{\text{known (“completion flow”)}}(\boldsymbol{y})d\boldsymbol{y} + \int_{\partial\Omega} D(\boldsymbol{x}-\boldsymbol{y}){\color{red}\boldsymbol{\sigma}}(\boldsymbol{y})d\boldsymbol{y}$$

known (“completion flow”)

[Pozrikidis, 1992]

[Rachh & Greengard, 2016]

Stokes mobility problem
Boundary integral formulation

Represent fluid velocity using Stokes single- and double-layer potentials:

$$S(\boldsymbol{r}) = \frac{1}{4\pi}\left(\log\frac{1}{|\boldsymbol{r}|}\boldsymbol{I} + \frac{\boldsymbol{r}\otimes\boldsymbol{r}}{|\boldsymbol{r}|^2}\right)$$

$$D(\boldsymbol{r}) = \frac{1}{\pi}\frac{\boldsymbol{r}\otimes\boldsymbol{r}}{|\boldsymbol{r}|^4}(\boldsymbol{r}\cdot\boldsymbol{n})$$

$$\boldsymbol{u} = \mathcal{S}[\boldsymbol{\nu}(\boldsymbol{F}_i, T_i)] + \mathcal{D}[\boldsymbol{\sigma}]$$

[Pozrikidis, 1992]

[Rachh & Greengard, 2016]

Stokes mobility problem
Boundary integral formulation

Represent fluid velocity using Stokes single- and double-layer potentials:

$$S(\boldsymbol{r}) = \frac{1}{4\pi}\left(\log\frac{1}{|\boldsymbol{r}|}\boldsymbol{I} + \frac{\boldsymbol{r}\otimes\boldsymbol{r}}{|\boldsymbol{r}|^2}\right)$$

$$D(\boldsymbol{r}) = \frac{1}{\pi}\frac{\boldsymbol{r}\otimes\boldsymbol{r}}{|\boldsymbol{r}|^4}(\boldsymbol{r}\cdot\boldsymbol{n})$$

$$\boldsymbol{u} = \mathcal{S}[\nu(\boldsymbol{F}_i, T_i)] + \mathcal{D}[\boldsymbol{\sigma}]$$

Represent rigid body velocity:

$$\boldsymbol{V} = -\sum_{i=1}^{3N_\Omega}\mathfrak{v}_i\mathfrak{v}_i^T\boldsymbol{\sigma}$$

orthonormal basis for rigid body motion on disc i

[Pozrikidis, 1992]

[Rachh & Greengard, 2016]

Stokes mobility problem
Boundary integral formulation

Represent fluid velocity using Stokes single- and double-layer potentials:

$$S(\boldsymbol{r}) = \frac{1}{4\pi}\left(\log\frac{1}{|\boldsymbol{r}|}\boldsymbol{I} + \frac{\boldsymbol{r}\otimes\boldsymbol{r}}{|\boldsymbol{r}|^2}\right)$$

$$D(\boldsymbol{r}) = \frac{1}{\pi}\frac{\boldsymbol{r}\otimes\boldsymbol{r}}{|\boldsymbol{r}|^4}(\boldsymbol{r}\cdot\boldsymbol{n})$$

$$\boldsymbol{u} = \mathcal{S}[\boldsymbol{\nu}(\boldsymbol{F}_i, T_i)] + \mathcal{D}[\boldsymbol{\sigma}]$$

Represent rigid body velocity:

$$\boldsymbol{V} = -\sum_{i=1}^{3N_\Omega} \mathfrak{v}_i \mathfrak{v}_i^T \boldsymbol{\sigma}$$

orthonormal basis for rigid body motion on disc i

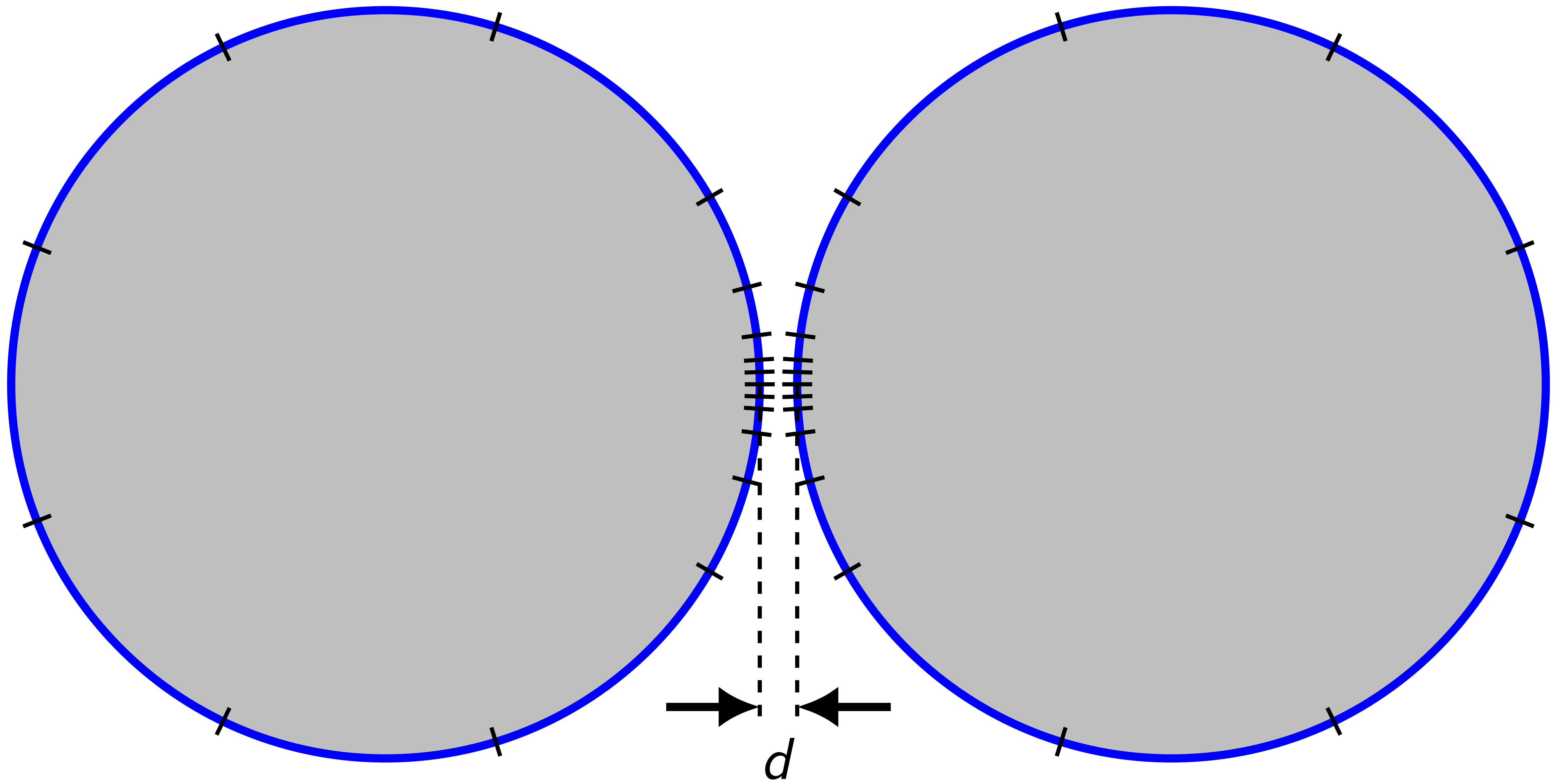
Apply boundary conditions:

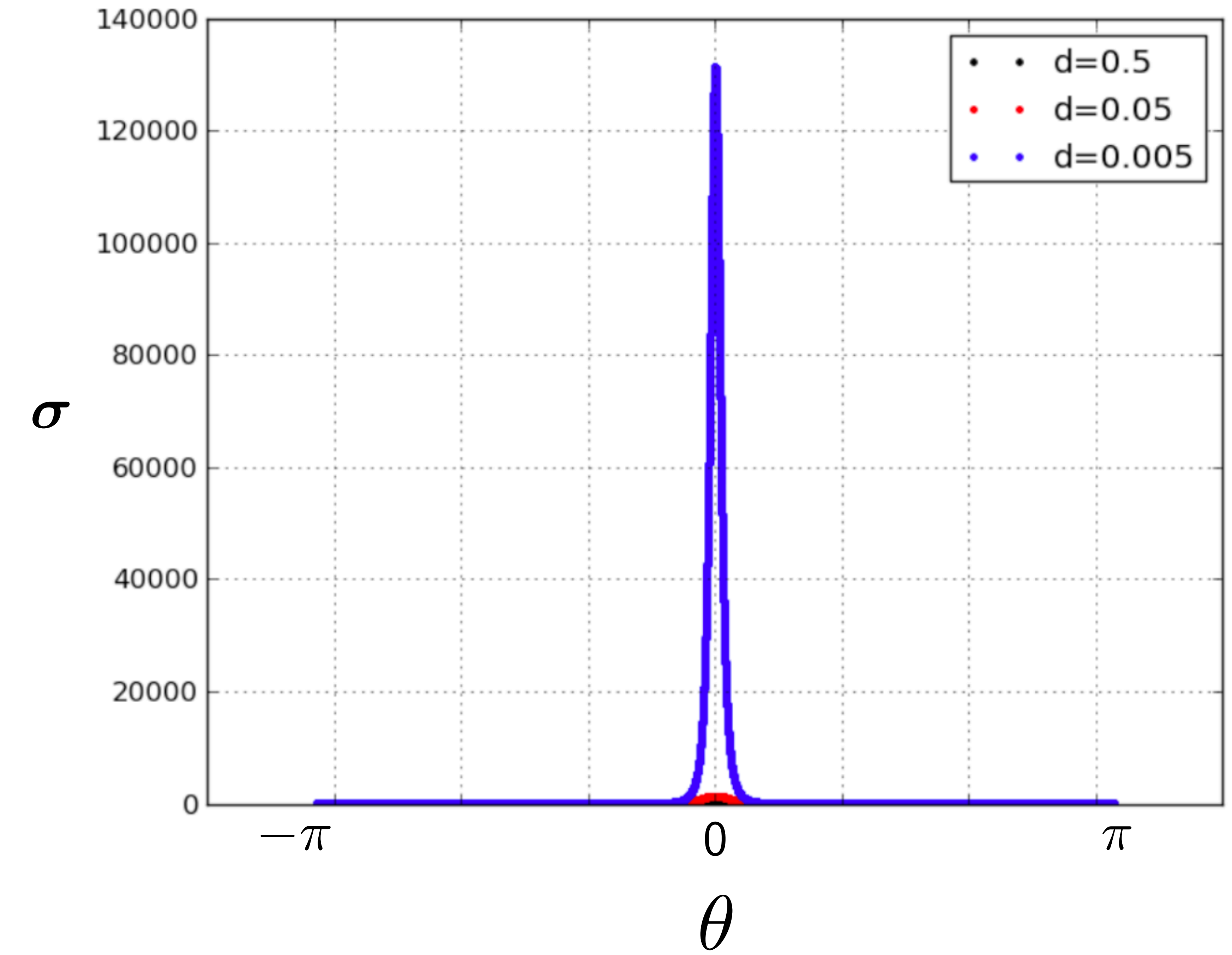
$$\left(I/2 + D + \sum_{i=1}^{3N_\Omega} \mathfrak{v}_i \mathfrak{v}_i^T\right)\boldsymbol{\sigma} = \boldsymbol{u}_s - S\boldsymbol{\nu}$$

(second kind BIE)

[Pozrikidis, 1992]
[Rachh & Greengard, 2016]

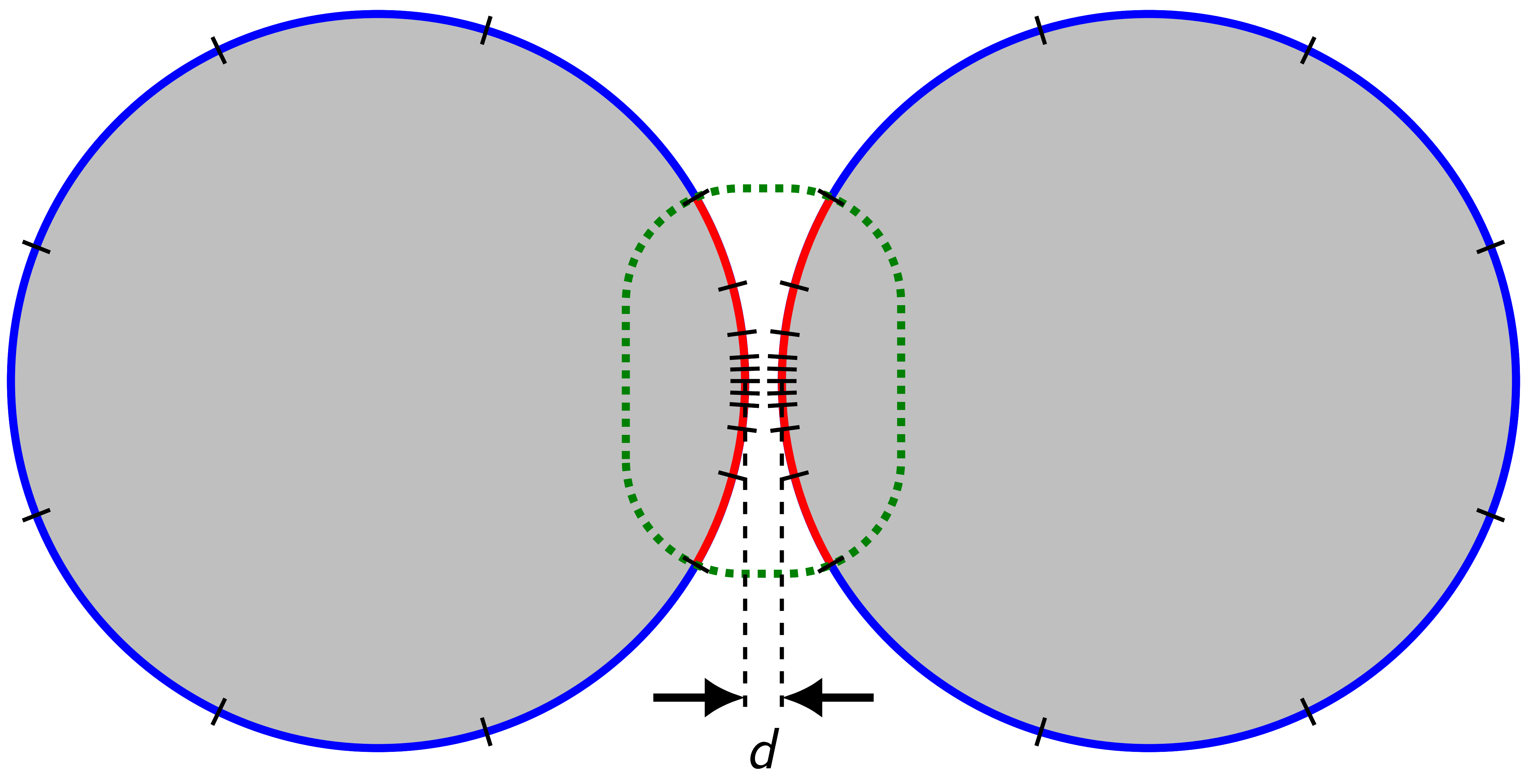
Nyström discretization
Close interactions

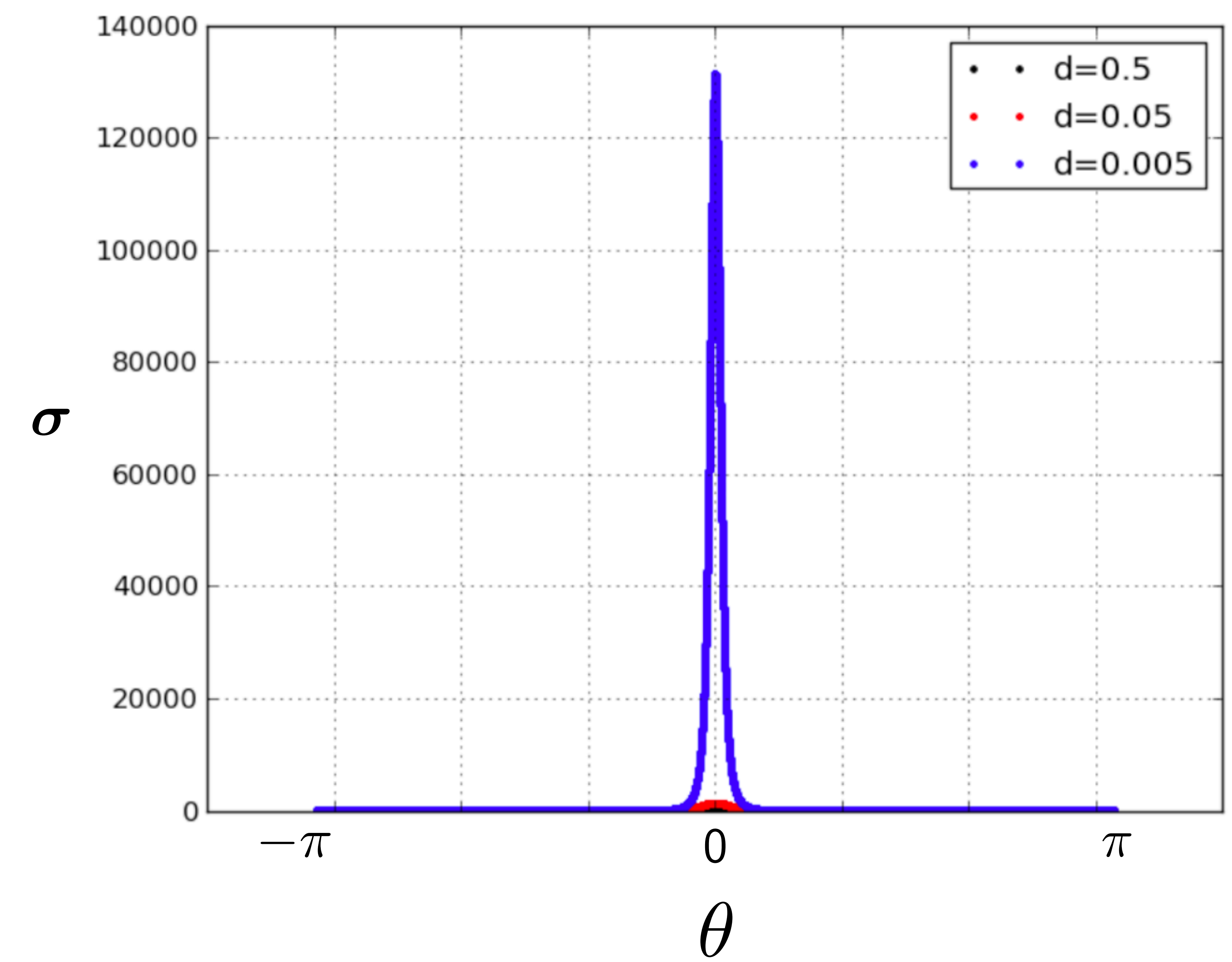




- Discretize $\partial\Omega$ into high-order panels
- Construct layer potential operators
  - Adaptive quadrature for near integrals
  - Special quadrature for singular integrals
- Solve BIE: $K\boldsymbol{\sigma} = g$

Nyström discretization
Close interactions

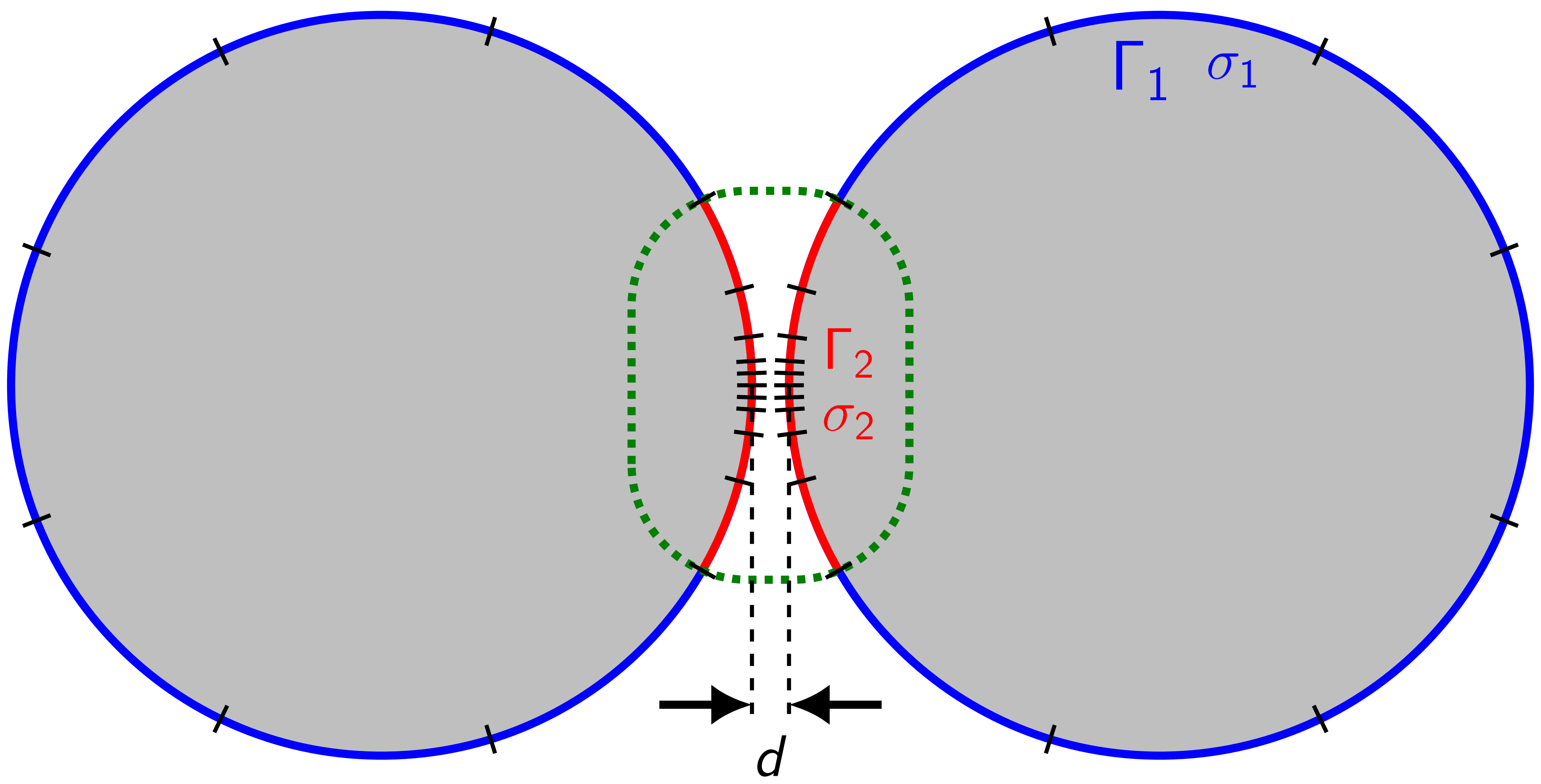




- Discretize $\partial\Omega$ into high-order panels
- Construct layer potential operators
  - Adaptive quadrature for near integrals
  - Special quadrature for singular integrals
- Solve BIE: $K\boldsymbol{\sigma} = g$

Compress close interactions

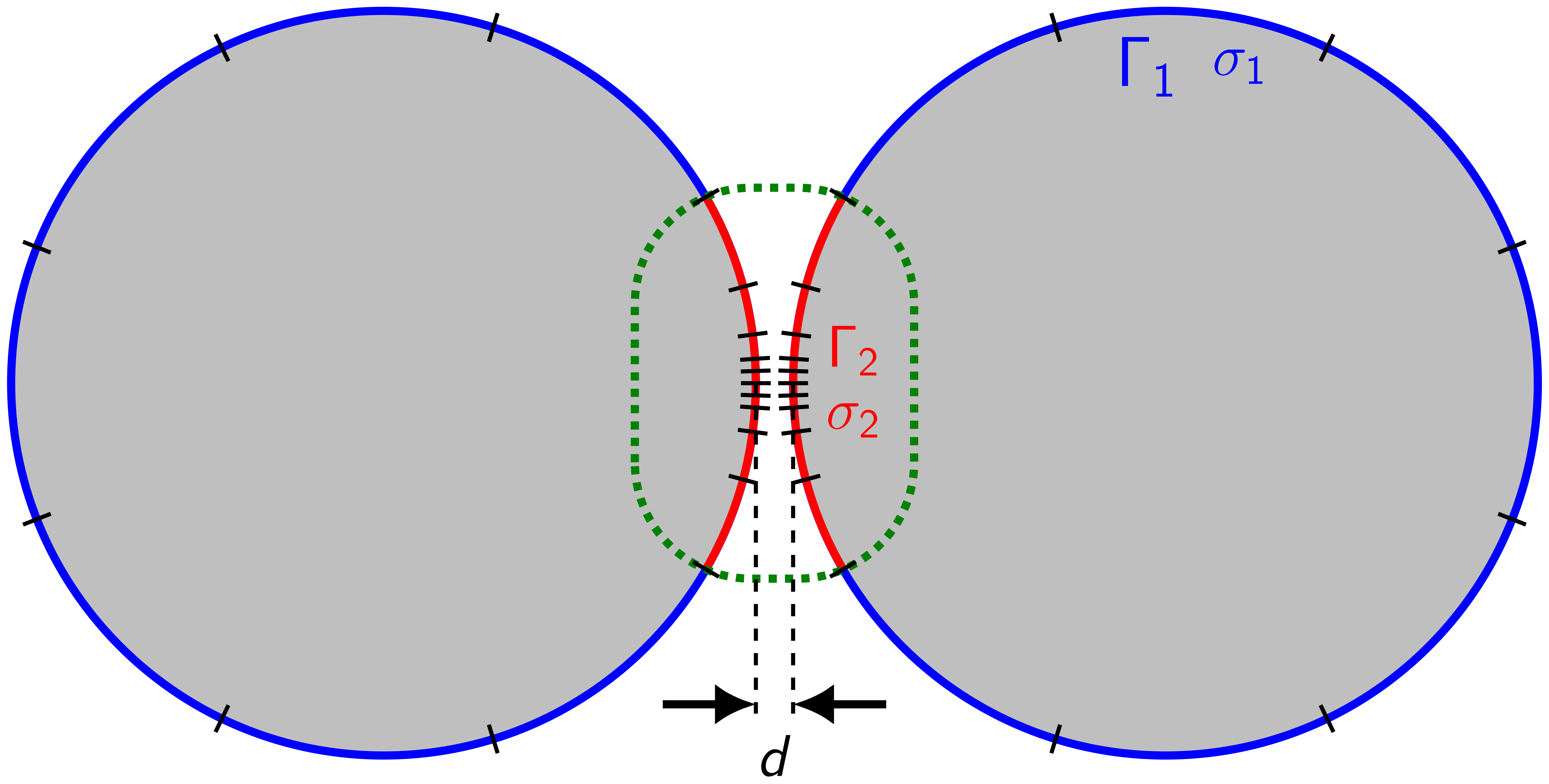
Compressing close interactions
Recursively compressed inverse preconditioning (RCIP)



$$\begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$$

Compressing close interactions

Recursively compressed inverse preconditioning (RCIP)

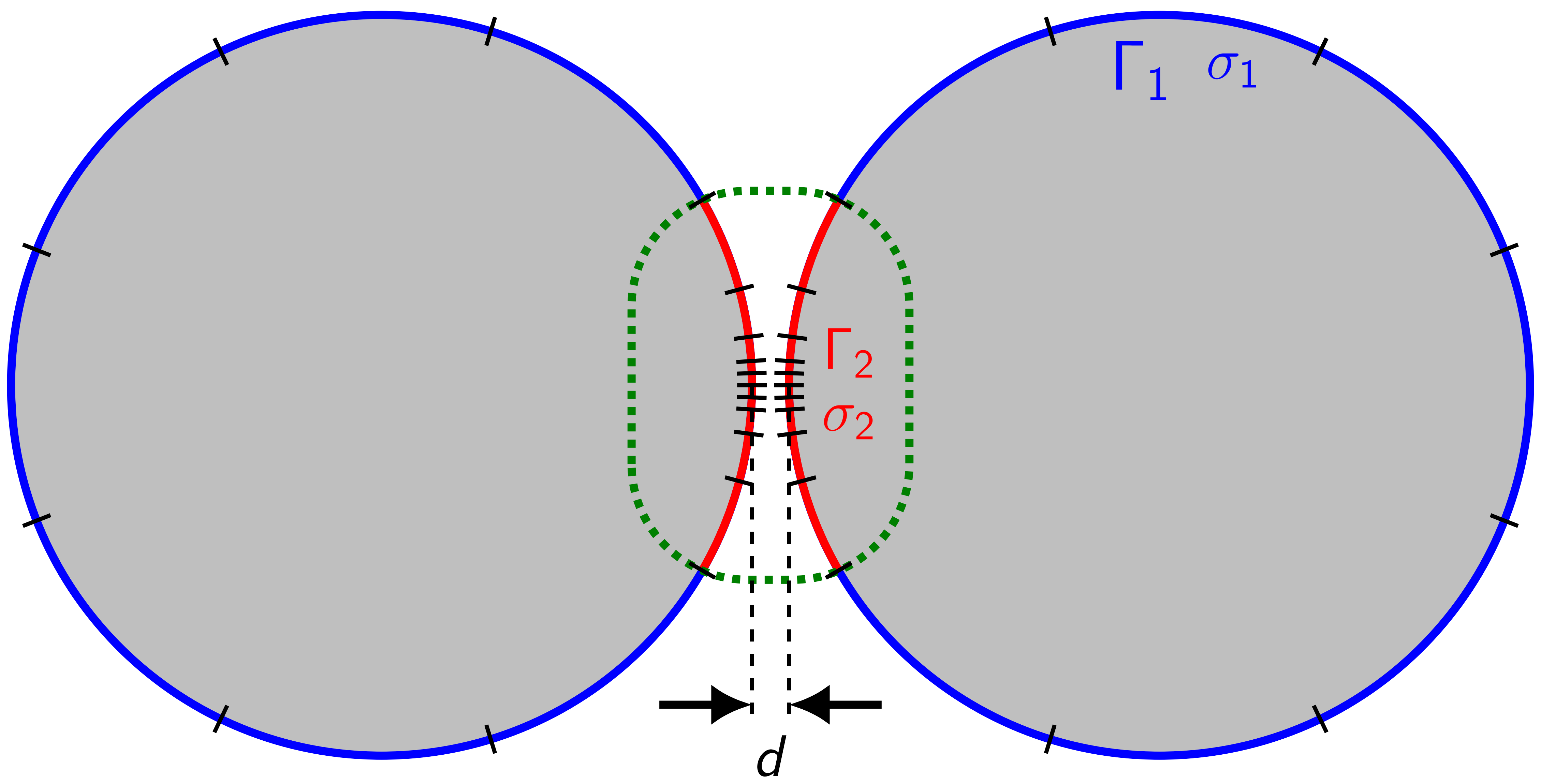


$$\begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$$

Build compressed operator using RCIP [Helsing, 2012]

Compressing close interactions

Recursively compressed inverse preconditioning (RCIP)



$$\begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$$

Build compressed operator using RCIP [Helsing, 2012]

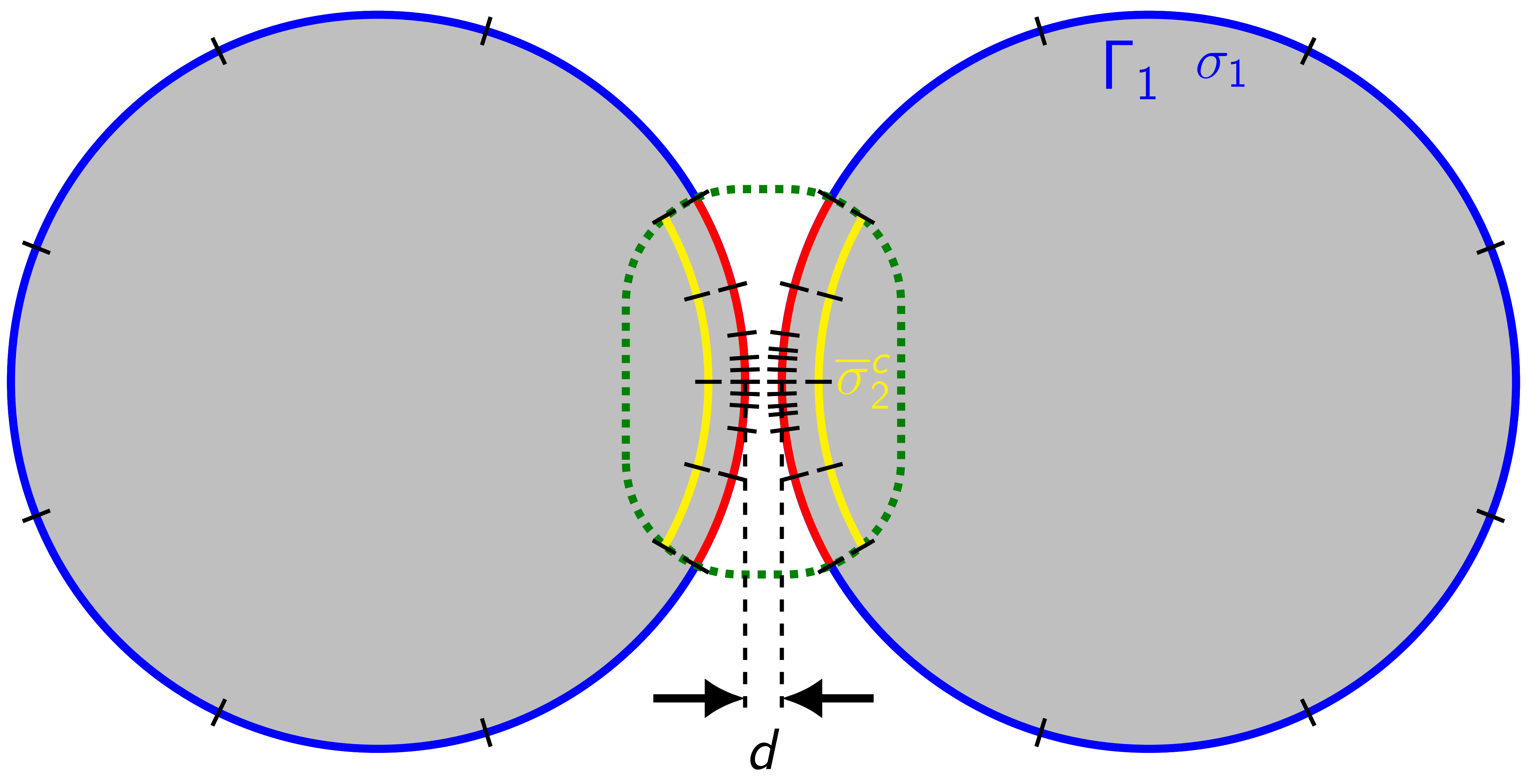
Right precondition with $\begin{pmatrix} I & 0 \\ 0 & K_{22}^{-1} \end{pmatrix}$:

$$\begin{pmatrix} K_{11} & K_{12}K_{22}^{-1} \\ K_{21} & I \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \overline{\sigma}_2 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$$

Then $\overline{\sigma}_2 = K_{22}\sigma_2$ is smooth and $K_{12}K_{22}^{-1}$ is low rank.

Compressing close interactions

Recursively compressed inverse preconditioning (RCIP)



$$\begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$$

Build compressed operator using RCIP [Helsing, 2012]

Right precondition with $\begin{pmatrix} I & 0 \\ 0 & K_{22}^{-1} \end{pmatrix}$:

$$\begin{pmatrix} K_{11} & K_{12}K_{22}^{-1} \\ K_{21} & I \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \overline{\sigma}_2 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$$

Then $\overline{\sigma}_2 = K_{22}\sigma_2$ is smooth and $K_{12}K_{22}^{-1}$ is low rank.

Project onto a coarse mesh:

$$\begin{pmatrix} K_{11} & K_{12}^c R \\ K_{21}^c & I \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \overline{\sigma}_2^c \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2^c \end{pmatrix}$$

where $R = \underbrace{\left(W_c^{-1} P^T W_f\right)}_{\substack{L^2 \text{ projection} \\ \text{(fine to coarse)}}} K_{22}^{-1} \underbrace{P}_{\substack{\text{interpolation} \\ \text{(coarse to fine)}}}.$

Compressing close interactions

Interpolated compressed inverse preconditioning (ICIP)

What is the cost of constructing $R(d)$ for each new d?

Compressing close interactions

Interpolated compressed inverse preconditioning (ICIP)

What is the cost of constructing $R(d)$ for each new d?

	2D	3D
Direct	$p^3(\log d)^3$	$p^6(\log d)^6$

(for $\log d$ panels of order p)

What is the cost of constructing $R(d)$ for each new d?

	2D	3D
Direct	$p^3(\log d)^3$	$p^6(\log d)^6$
RCIP	$p^3 \log d$	$p^6 \log d$

(for $\log d$ panels of order p)

Compressing close interactions
Interpolated compressed inverse preconditioning (ICIP)

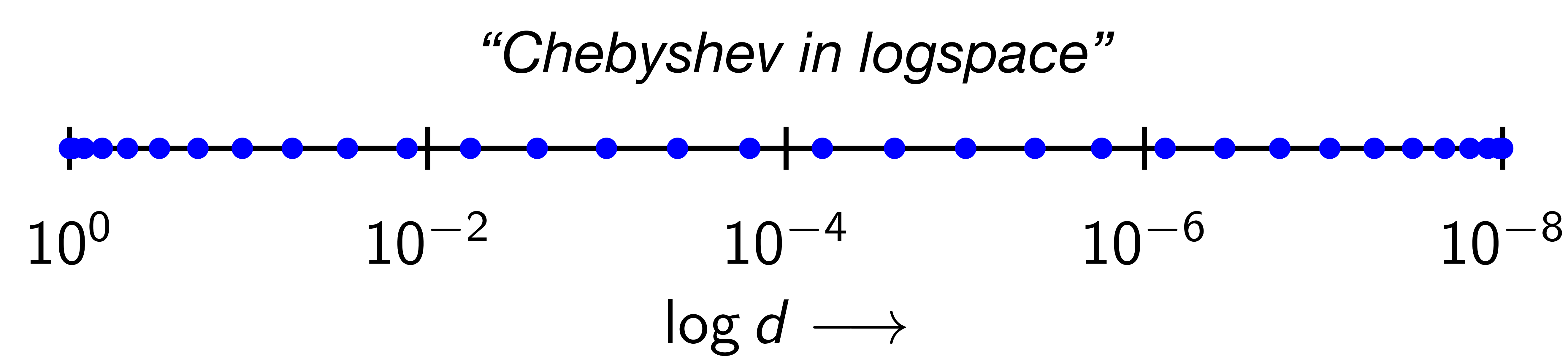
What is the cost of constructing $R(d)$ for each new d?

	2D	3D
Direct	$p^3(\log d)^3$	$p^6(\log d)^6$
RCIP	$p^3 \log d$	$p^6 \log d$

(for $\log d$ panels of order p)

Idea: Precompute a library of R's and interpolate to a new d on the fly.

$$[R(d)]_{ij} \approx \sum_{k=0}^{q} [R_k]_{ij} T_k(\log d)$$



Compressing close interactions
Interpolated compressed inverse preconditioning (ICIP)

What is the cost of constructing $R(d)$ for each new d?

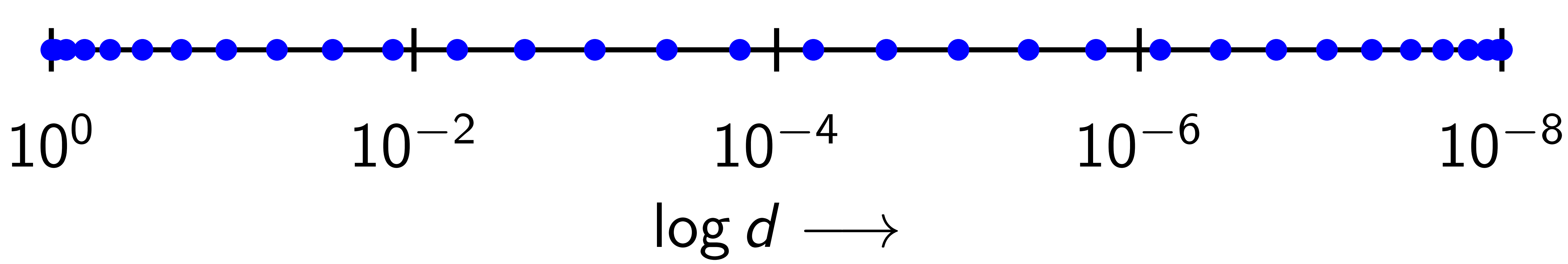
	2D	3D
Direct	$p^3(\log d)^3$	$p^6(\log d)^6$
RCIP	$p^3 \log d$	$p^6 \log d$
ICIP	$p^2 q$	$p^4 q$

(for $\log d$ panels of order p)

Idea: Precompute a library of R's and interpolate to a new d on the fly.

$$[R(d)]_{ij} \approx \sum_{k=0}^{q} [R_k]_{ij} T_k(\log d)$$

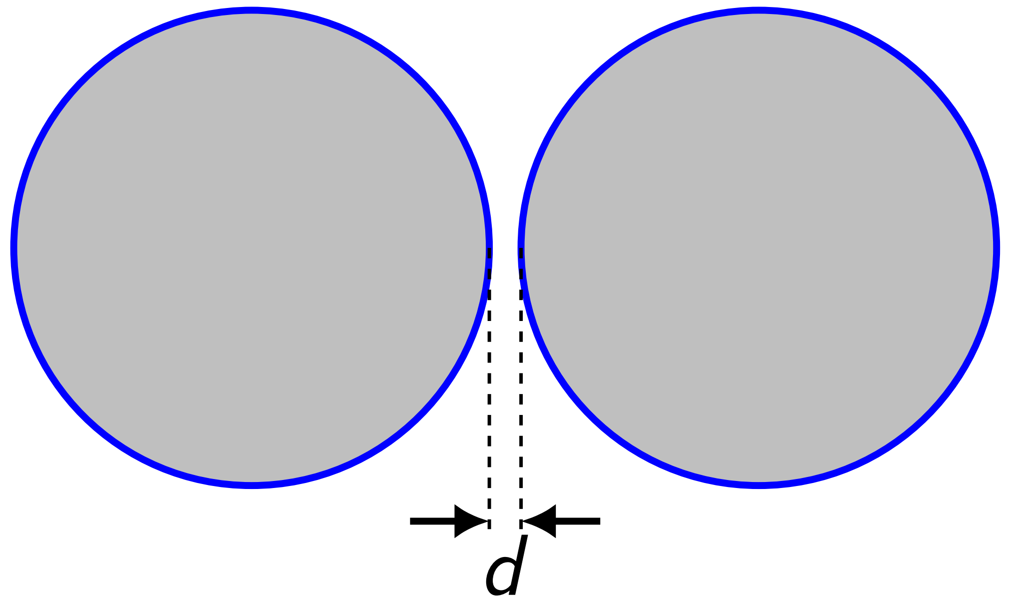
"Chebyshev in logspace"



Results
Convergence of ICIP

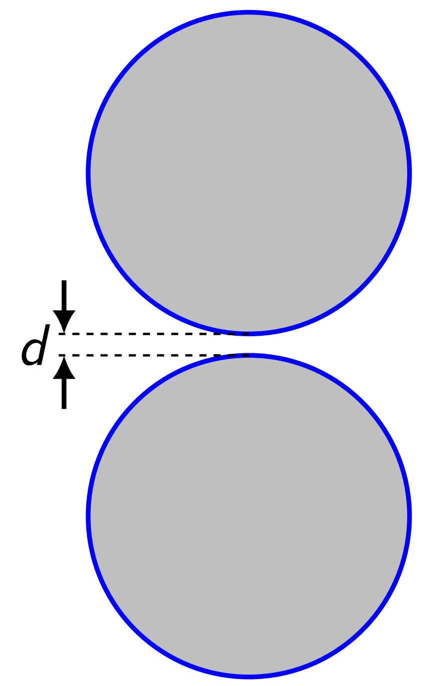
Errors for Stokes mobility with two discs

d	Adaptive discretization	Interpolating $R(d)$				
		$q=8$	$q=16$	$q=24$	$q=32$	$q=40$
10^{-1}	7.6×10^{-15}	1.0×10^{-4}	2.9×10^{-7}	2.1×10^{-9}	9.1×10^{-12}	9.0×10^{-15}
10^{-2}	1.8×10^{-13}	2.8×10^{-3}	3.6×10^{-6}	3.6×10^{-8}	6.5×10^{-10}	1.6×10^{-14}
10^{-3}	4.4×10^{-13}	3.4×10^{-5}	5.6×10^{-10}	4.8×10^{-14}		
10^{-4}	3.8×10^{-11}	1.5×10^{-3}	1.4×10^{-9}	1.4×10^{-13}		
10^{-5}	9.0×10^{-9}	1.5×10^{-5}	2.1×10^{-12}			
10^{-6}	2.0×10^{-7}	6.0×10^{-4}	1.4×10^{-11}			
10^{-7}	4.3×10^{-7}	1.7×10^{-5}	4.1×10^{-11}			
10^{-8}	5.3×10^{-8}	6.3×10^{-4}	3.9×10^{-9}			



Results
GMRES iterations for ICIP

GMRES iteration counts to solve Stokes mobility with $\epsilon = 10^{-8}$



Adaptive discretization

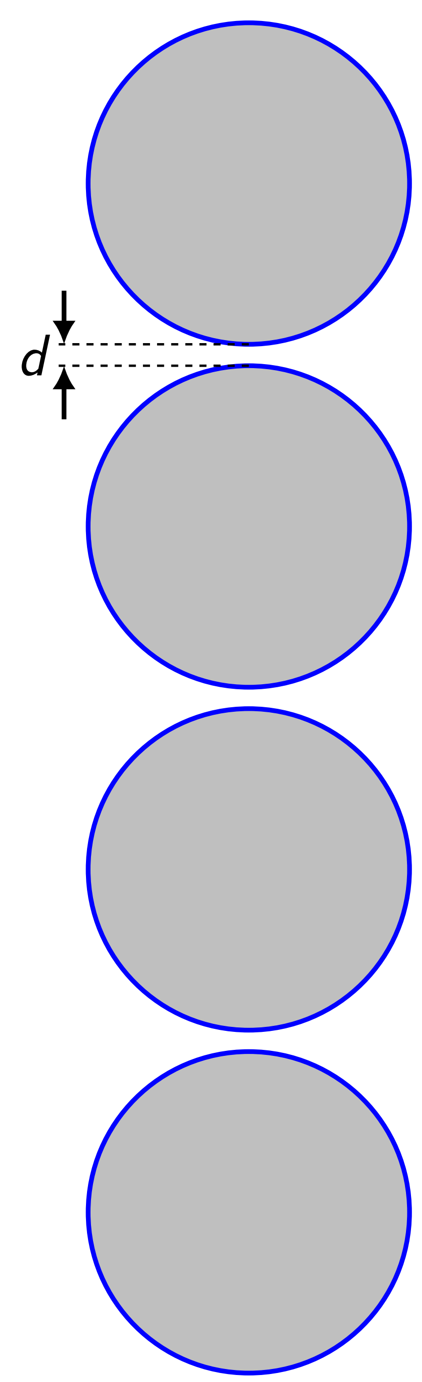
N_{disc}	$d = 10^{-1}$	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
2	15	37	104	337	1283	1848	2344

Interpolated compressed inverse preconditioning (ICIP)

N_{disc}	$d = 10^{-1}$	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
2	18	20	21	21	21	21	21

Results
GMRES iterations for ICIP

GMRES iteration counts to solve Stokes mobility with $\epsilon = 10^{-8}$



Adaptive discretization

N_{disc}	$d = 10^{-1}$	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
2	15	37	104	337	1283	1848	2344
4	25	75	271	1134	3770	5301	6620

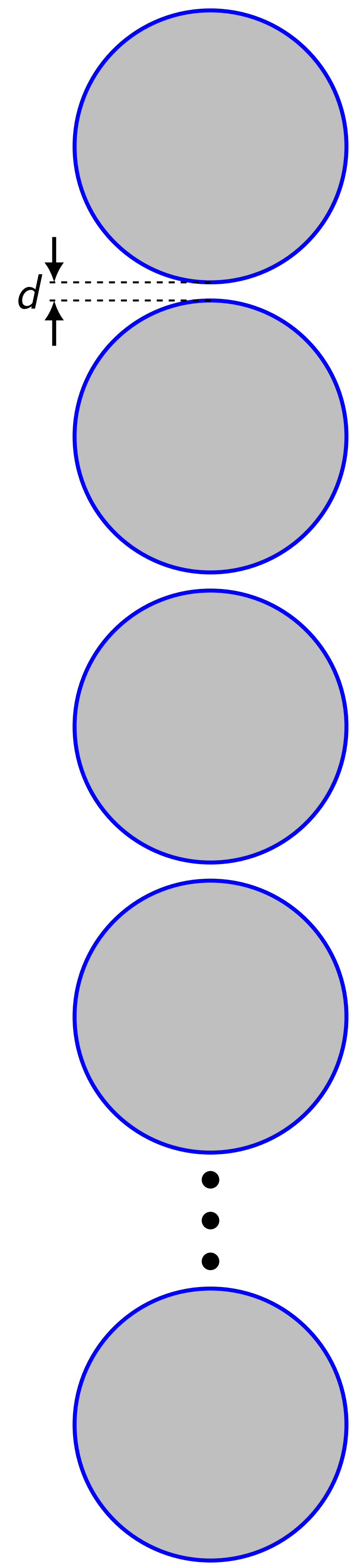
Interpolated compressed inverse preconditioning (ICIP)

N_{disc}	$d = 10^{-1}$	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
2	18	20	21	21	21	21	21
4	28	34	36	37	37	37	37

Results

GMRES iterations for ICIP

GMRES iteration counts to solve Stokes mobility with $\epsilon = 10^{-8}$



Adaptive discretization

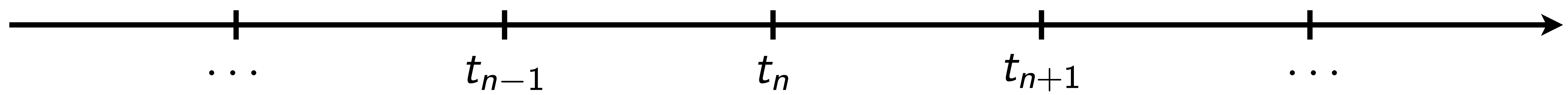
N_{disc}	$d = 10^{-1}$	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
2	15	37	104	337	1283	1848	2344
4	25	75	271	1134	3770	5301	6620
16	35	147	629	2754	8000+	–	–
64	36	148	683	3094	–	–	–
256	37	149	683	3094	–	–	–

Interpolated compressed inverse preconditioning (ICIP)

N_{disc}	$d = 10^{-1}$	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
2	18	20	21	21	21	21	21
4	28	34	36	37	37	37	37
16	46	71	74	80	86	87	88
64	49	96	108	131	186	237	251
256	49	98	110	134	220	371	608

Accelerating GMRES

Krylov subspace recycling

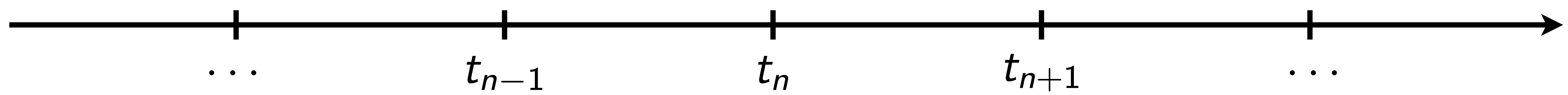


Forward Euler at timestep n:

- Solve mobility problem using GMRES: $K_n \sigma_n = g_n$
- Advance to t_{n+1}: $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta t \boldsymbol{V}(\sigma_n)$

Accelerating GMRES

Krylov subspace recycling

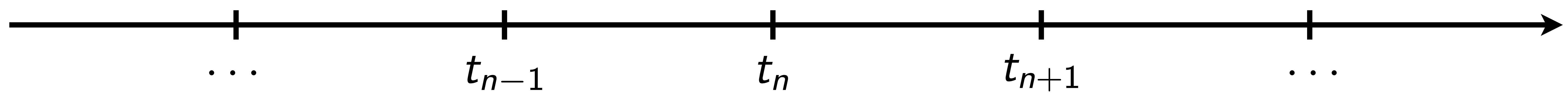


Forward Euler at timestep n:

- Solve mobility problem using GMRES: $K_n \sigma_n = g_n$
- Advance to t_{n+1}: $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta t \, \boldsymbol{V}(\sigma_n)$

Question: Can we use information from t_{n-1} to accelerate GMRES at t_n?

Accelerating GMRES
Krylov subspace recycling



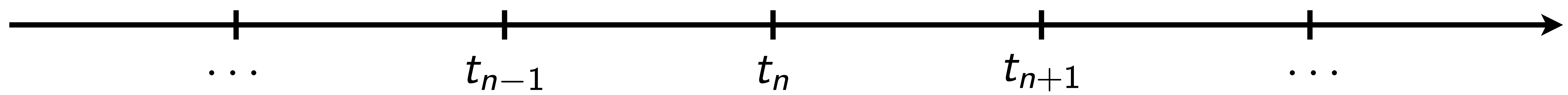
Forward Euler at timestep n:

- Solve mobility problem using GMRES: $K_n \sigma_n = g_n$
- Advance to t_{n+1}: $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta t\, \boldsymbol{V}(\sigma_n)$

Idea: Use σ_{n-1} as an initial guess to GMRES… doesn't help at all.

Accelerating GMRES

Krylov subspace recycling



Forward Euler at timestep *n*:

- Solve mobility problem using GMRES: $K_n \sigma_n = g_n$
- Advance to t_{n+1}: $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta t \, \boldsymbol{V}(\sigma_n)$

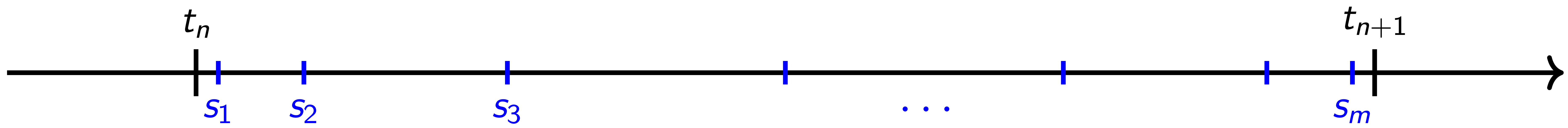
Idea: Reuse Krylov subspace from previous time step… very effective!

- Krylov subspace after *k* iterations: $X = [b \quad Ab \quad \cdots \quad A^{k-1} b]$
- Compute QR decomposition: $QR = AX$
- Define preconditioner: $P = I - QQ^T + XR^{-1}Q^T$

$$\boxed{\begin{aligned} PAx &= x \ \text{ for all } \ x \in \operatorname{span}(X) \\ Py &= y \ \text{ for all } \ y \perp \operatorname{span}(X) \end{aligned}}$$

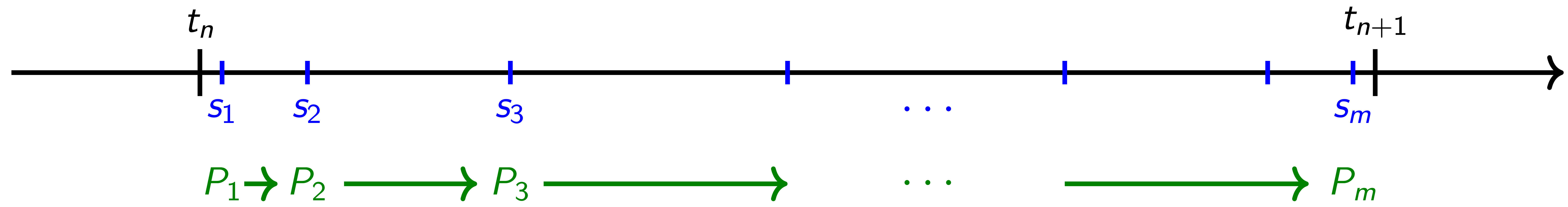
Accelerating GMRES

Krylov recycling with spectral deferred corrections



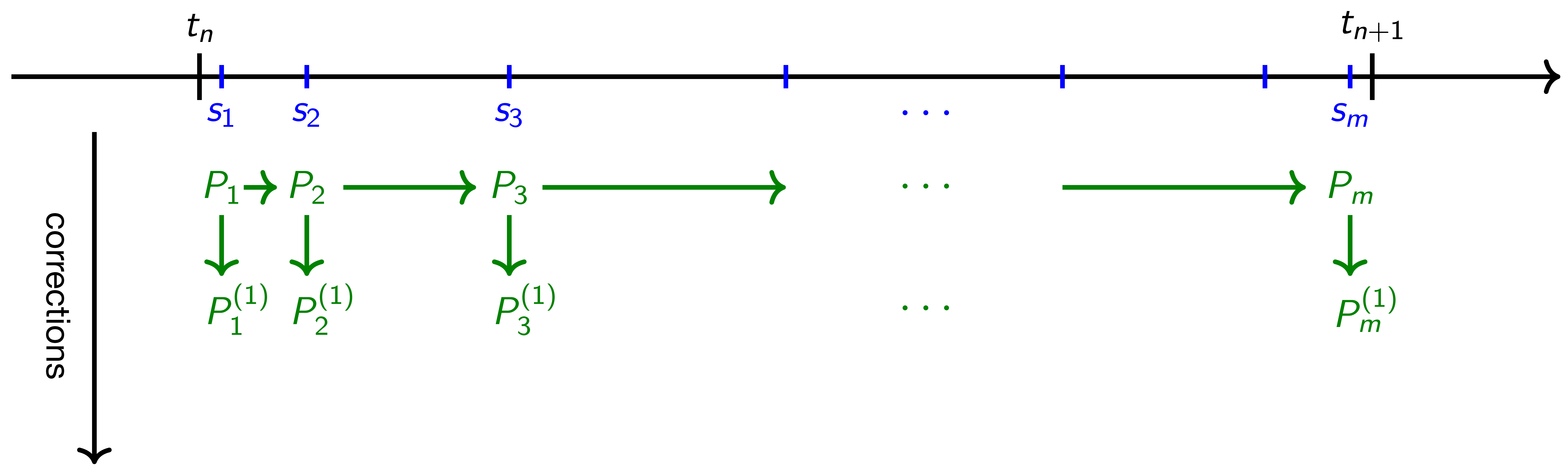
Accelerating GMRES

Krylov recycling with spectral deferred corrections



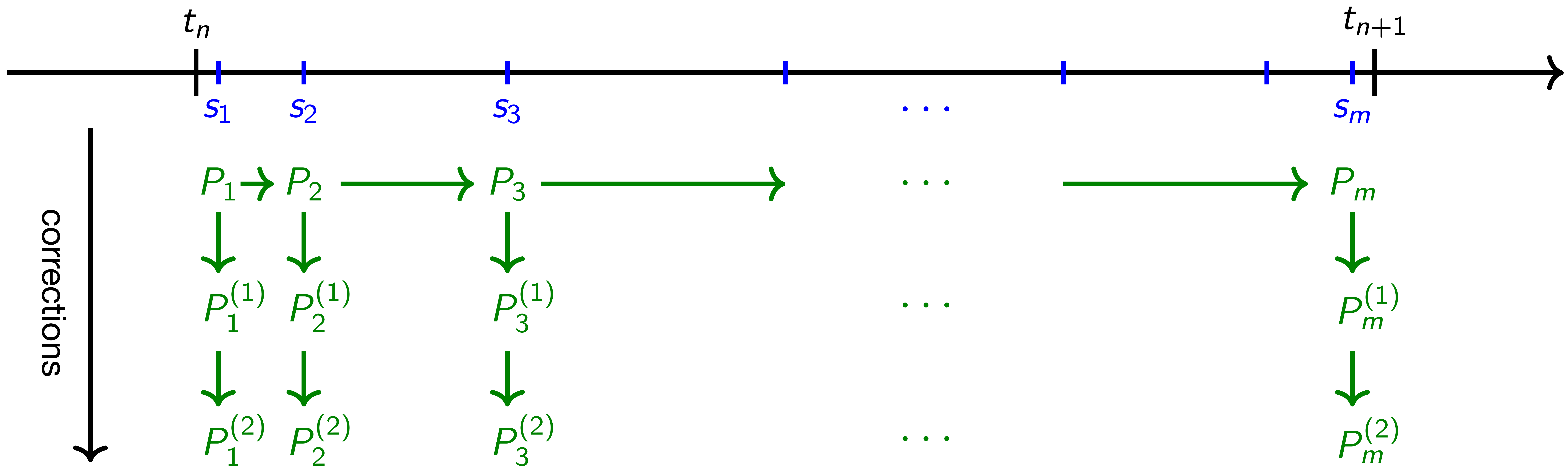
Accelerating GMRES

Krylov recycling with spectral deferred corrections



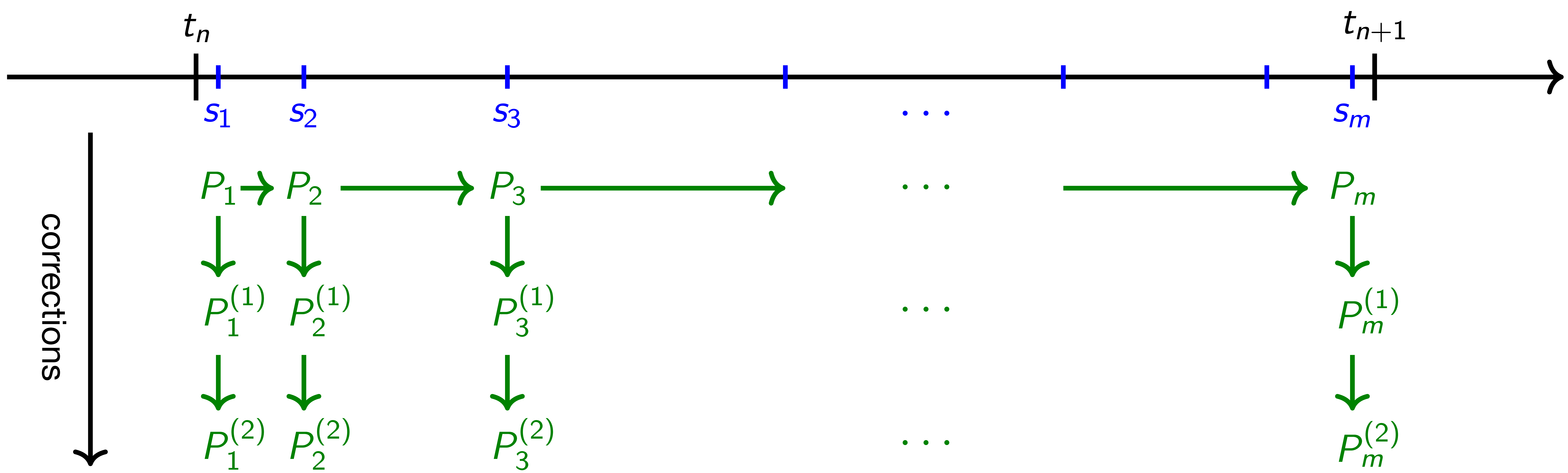
Accelerating GMRES

Krylov recycling with spectral deferred corrections



Accelerating GMRES

Krylov recycling with spectral deferred corrections

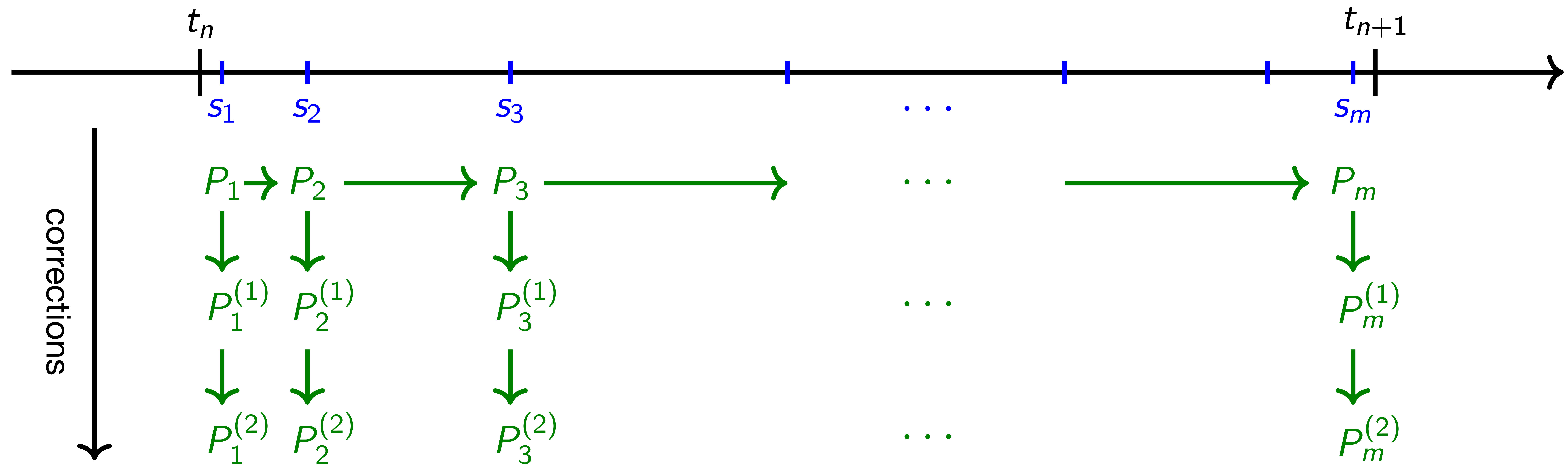


GMRES iterations **without** preconditioner

correction ↓ \ substep →				
66	66	66	66	66
66	66	66	66	66
66	66	66	66	66
66	66	66	66	66

Accelerating GMRES

Krylov recycling with spectral deferred corrections



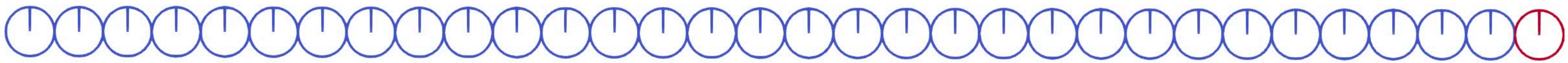
GMRES iterations **without** preconditioner

correction ↓ \ substep ⟶					
	66	66	66	66	66
	66	66	66	66	66
	66	66	66	66	66
	66	66	66	66	66

GMRES iterations **with** preconditioner

correction ↓ \ substep ⟶					
	66	30	22	45	30
	35	17	33	28	24
	8	4	14	5	12
	1	1	2	2	4

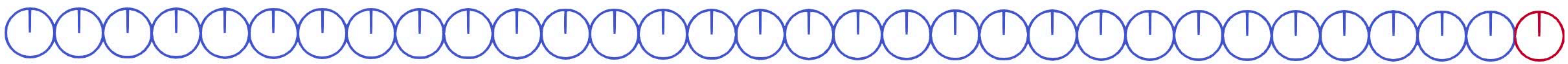
Results
A chain of discs



Average GMRES iterations per timestep **without** Krylov preconditioning = 130
with Krylov preconditioning = 30

32 discs (36k unknowns)
10^{-4} minimum distance
10th-order adaptive SDC
7-digit accuracy
~2000 timesteps
64 cores, 48 hours
84 seconds/timestep
1.6 seconds/mobility solve
100k mobility solves
3.2 million iterations

Results
A chain of discs

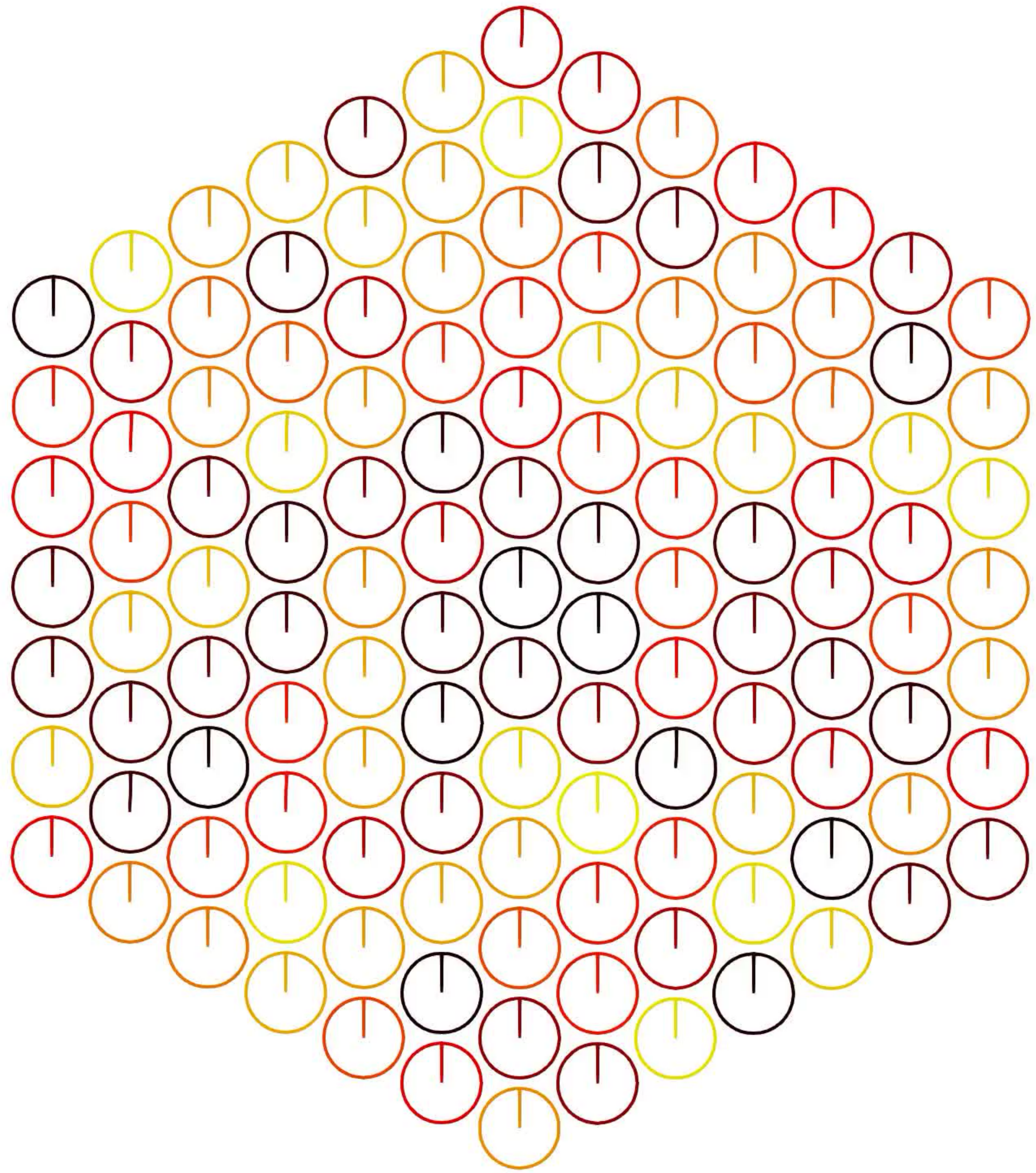


Average GMRES iterations per timestep **without** Krylov preconditioning = 130
with Krylov preconditioning = 30

32 discs (36k unknowns)
10^{-4} minimum distance
10th-order adaptive SDC
7-digit accuracy
~2000 timesteps
64 cores, 48 hours
84 seconds/timestep
1.6 seconds/mobility solve
100k mobility solves
3.2 million iterations

Results

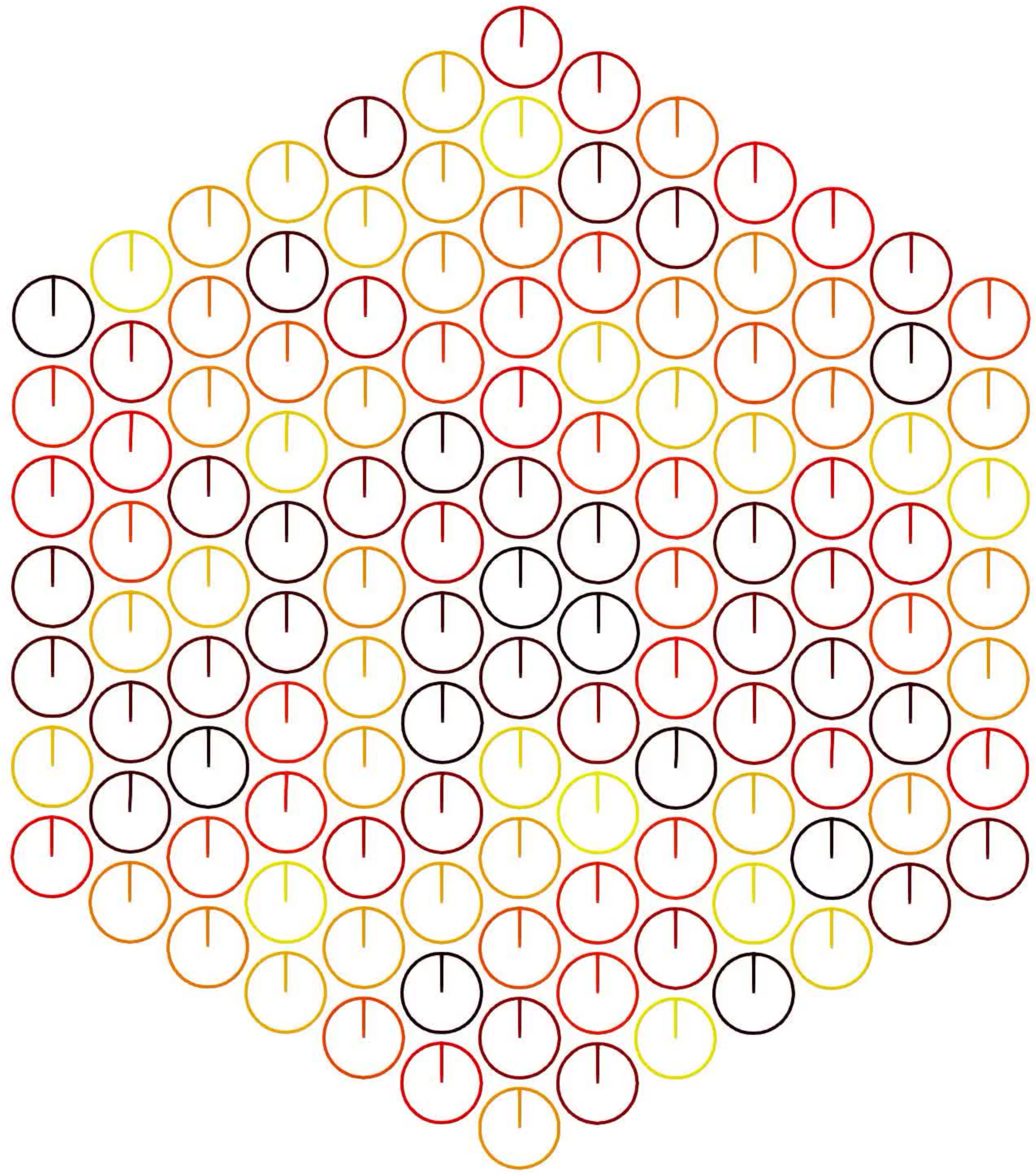
Sedimentation flow



127 discs (200k unknowns)
10^{-4} minimum distance
10th-order adaptive SDC
5-digit accuracy
~200 timesteps
64 cores, 1 week
6,480 mobility solves
1.3m iterations

Results

Sedimentation flow



127 discs (200k unknowns)
10^{-4} minimum distance
10th-order adaptive SDC
5-digit accuracy
~200 timesteps
64 cores, 1 week
6,480 mobility solves
1.3m iterations

Conclusions

- **Interpolated compressed inverse preconditioning (ICIP)**: a fast pairwise preconditioner using interpolation of precomputed compressed interaction operators.
- **Krylov subspace recycling** can drastically accelerate high-order timestepping.
- Ongoing work: flows in confined geometries, extension to spheres in 3D.
- Limited to simple geometries for now (discs, flat planes, etc.)

Conclusions

- **Interpolated compressed inverse preconditioning (ICIP)**: a fast pairwise preconditioner using interpolation of precomputed compressed interaction operators.
- **Krylov subspace recycling** can drastically accelerate high-order timestepping.
- Ongoing work: flows in confined geometries, extension to spheres in 3D.
- Limited to simple geometries for now (discs, flat planes, etc.)

Mariana Martínez Aguilar
EPFL

Thanks!

Dhairya Malhotra
Flatiron Institute