2.29 Project: Modeling Stokes Flow Through Randomly Oriented Cylinders

Xinyu Mao

5/20/2021

Motivation

- **Stokes flow (creeping flow)**: a type of fluid flow where inertial forces are small compared with viscous forces ($\mathrm{Re} \ll 1$)

- Governing equations (Newtonian fluids)

$$\nabla \cdot \boldsymbol{u} = 0$$
$$0 = \nabla p + \mu \Delta \boldsymbol{u} + \boldsymbol{f}$$

  - Linear, elliptic PDE
  - Methods: Finite difference/volume/elements, boundary integral methods, lattice Boltzmann methods etc.

- Stokes flow through randomly oriented cylinders
  - Micro-scale organisms
  - Microfluidic engineering applications

- Challenges for random cylinder system
  - Complex geometry when cylinder number is large
  - Sharp corners for overlapping cylinders

Algorithm: Regularized Stokeslet Boundary integral method

- **Boundary integral method**: formulate boundary value problem (BVP) as integral equations on the boundary of domain

- **Stokeslet**: analytical solution of Stokes flow generated by a singular point force concentrated at $\boldsymbol{x}_0$

$$-\nabla p + \mu \Delta \boldsymbol{u} = 0, \; \nabla \cdot \boldsymbol{u} = 0 \quad \Longrightarrow \quad -\nabla p + \mu \Delta \boldsymbol{u} + \boldsymbol{g}\, \delta(\boldsymbol{x} - \boldsymbol{x_0}) = 0, \; \nabla \cdot \boldsymbol{u} = 0$$

$\boldsymbol{g}$	Singular point force
$\delta(\cdot)$	Dirac delta function
x_0	Point of action of force

- **Stokeslet (velocity field)**

$$u_i(\boldsymbol{x}) = \frac{1}{8\pi\mu} S_{ij}(\boldsymbol{x}, \boldsymbol{x_0})\, g_j \qquad S_{ij}(\boldsymbol{x}, \boldsymbol{x_0}) = \frac{\delta_{ij}}{|\boldsymbol{x} - \boldsymbol{x_0}|} + \frac{(x_i - x_{0i})(x_j - x_{0j})}{|\boldsymbol{x} - \boldsymbol{x_0}|^3}$$

S_{ij} Component of the Oseen tensor

BVP Example: sphere moving in fluids (= flow past sphere)

- **Thought:** one could integrate the Stokeslet over the boundary to obtain the global solution

Algorithm: Regularized Stokeslet Boundary integral method

- **Issues of direct integration:** special numerical techniques (e.g., handling of delta function)

Alternative: regularized Stokeslet boundary integral

- Cortez (2001, 2005)
- **Regularization:**

$$-\nabla p + \mu \Delta \boldsymbol{u} + \boldsymbol{g}\, \delta(\boldsymbol{x} - \boldsymbol{x_0}) = 0\,,\ \nabla \cdot \boldsymbol{u} = 0$$

$$\Rightarrow \quad -\nabla p + \mu \Delta \boldsymbol{u} + \boldsymbol{g}\, \phi_\epsilon(\boldsymbol{x} - \boldsymbol{x_0}) = 0\,,\ \nabla \cdot \boldsymbol{u} = 0$$

ϕ_ϵ Cutoff function (approximation of delta function in 3D). Example:

$$\phi_\epsilon(\boldsymbol{x} - \boldsymbol{x_0}) = \frac{15\epsilon^4}{8\pi(r^2+\epsilon^2)^{7/2}} \quad \text{with} \quad r = |\boldsymbol{x} - \boldsymbol{x_0}|$$

ϵ measures degree of approximation of $\phi_\epsilon(\cdot)$ to $\delta(\cdot)$

- **Regularized Stokeslet (velocity field):**

$$u_i(\boldsymbol{x}) = \frac{1}{8\pi\mu} S^\epsilon_{ij}(\boldsymbol{x}, \boldsymbol{x_0})\, g_j$$

$$S^\epsilon_{ij}(\boldsymbol{x}, \boldsymbol{x_0}) = \delta_{ij} \frac{|\boldsymbol{x} - \boldsymbol{x_0}|^2 + 2\epsilon^2}{(|\boldsymbol{x} - \boldsymbol{x_0}|^2 + \epsilon^2)^{3/2}} + \frac{(x_i - x_{0i})(x_j - x_{0j})}{(|\boldsymbol{x} - \boldsymbol{x_0}|^2 + \epsilon^2)^{3/2}}$$

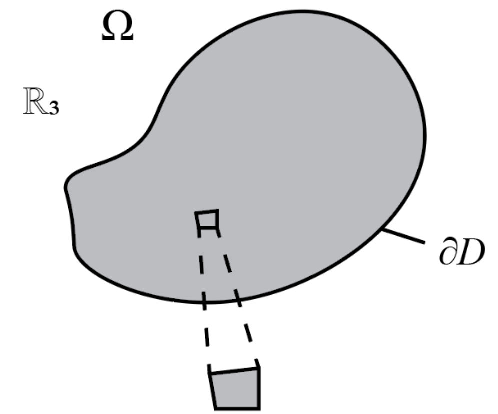
$$\epsilon \to 0\,,\ S^\epsilon_{ij} \to S_{ij}$$

Algorithm: Regularized Stokeslet Boundary integral method

- **Regularized Stokeslet boundary integral**

$$u_i(\boldsymbol{x}) = \frac{1}{8\pi\mu} S_{ij}^{\epsilon}(\boldsymbol{x}, \boldsymbol{x_0}) g_j \quad \cdots$$

$$u_i(\boldsymbol{x}) = \frac{1}{8\pi\mu} \int_{\partial D} S_{ij}^{\epsilon}(\boldsymbol{x}, \boldsymbol{y}) g_j(\boldsymbol{y}) dS(\boldsymbol{y}) \quad \text{for all } \boldsymbol{y} \in \partial D$$



dS: area per element
Δs: length scale per element

- **Discretization**

$$\dot{x}_i[m] = \frac{1}{8\pi\mu} \sum_{n=1}^{N} S_{ij}^{\epsilon}(\boldsymbol{x}[m], \boldsymbol{x}[n]) g_j[n] S[n] \qquad \text{for} \qquad m = 1, \ldots, N$$

⇒ This gives a linear system: $\boldsymbol{u} = \mathbf{A}\boldsymbol{f}$. Since $\boldsymbol{u}$ is known at ∂D due to no slip condition, we can solve for $\boldsymbol{f}$.

⇒ The entire flow field for grid in Ω (relevant quantities denoted by ~): $\tilde{\boldsymbol{u}} = \tilde{\mathbf{A}}\boldsymbol{f}$

- **Error analysis**: regularization error + discretization error

$$\mathcal{O}\left(\frac{\Delta s^2}{\epsilon^3}\right) + \mathcal{O}(\epsilon^q) \qquad q = 1 \text{ near } \partial D,\ q = 2 \text{ away from } \partial D$$

Algorithm: Regularized Stokeslet Boundary integral method

Remaining issue: High computational cost

$$\mathcal{O}\left(\frac{\Delta s^2}{\epsilon^3}\right) + \mathcal{O}(\epsilon^q)$$

1. Both ϵ and Δs should be sufficiently small
2. Approximating delta function requires refined nodes, but this is redundant for traction force

Nearest-neighbor Regularized Stokeslet

Smith (2018)

- Two sets of discretization $Q \gg N$
  - Force discretization: $\boldsymbol{x}[n],\ n=1,\dots,N$ for $g_j[n]S[n]$
  - Quadrature discretization: $\boldsymbol{X}[q],\ q=1,\dots,Q$ for $S^{\epsilon}_{ij}(\cdot,\cdot)$
- For each quadrature node, find its nearest force node $\nu[q,n]$
- Mapping the force from the nearest force node to the quadrature node
- Linear system size for $\boldsymbol{u}=\mathbf{A}\boldsymbol{f}$: $\mathcal{O}(Q\times Q) \to \mathcal{O}(N\times N)$

$$\dot{x}_i[m] = \frac{1}{8\pi\mu}\sum_{n=1}^{N} S^{\epsilon}_{ij}(\boldsymbol{x}[m],\boldsymbol{x}[n])\,g_j[n]S[n]$$

Regularized Stokeslet

$$\dot{x}_i[m] = \frac{1}{8\pi\mu}\sum_{q=1}^{Q} S^{\epsilon}_{ij}(\boldsymbol{x}[m],\boldsymbol{X}[q])\sum_{n=1}^{N}\nu[q,n]\,g_j[n]S[n]$$

"Mapping"

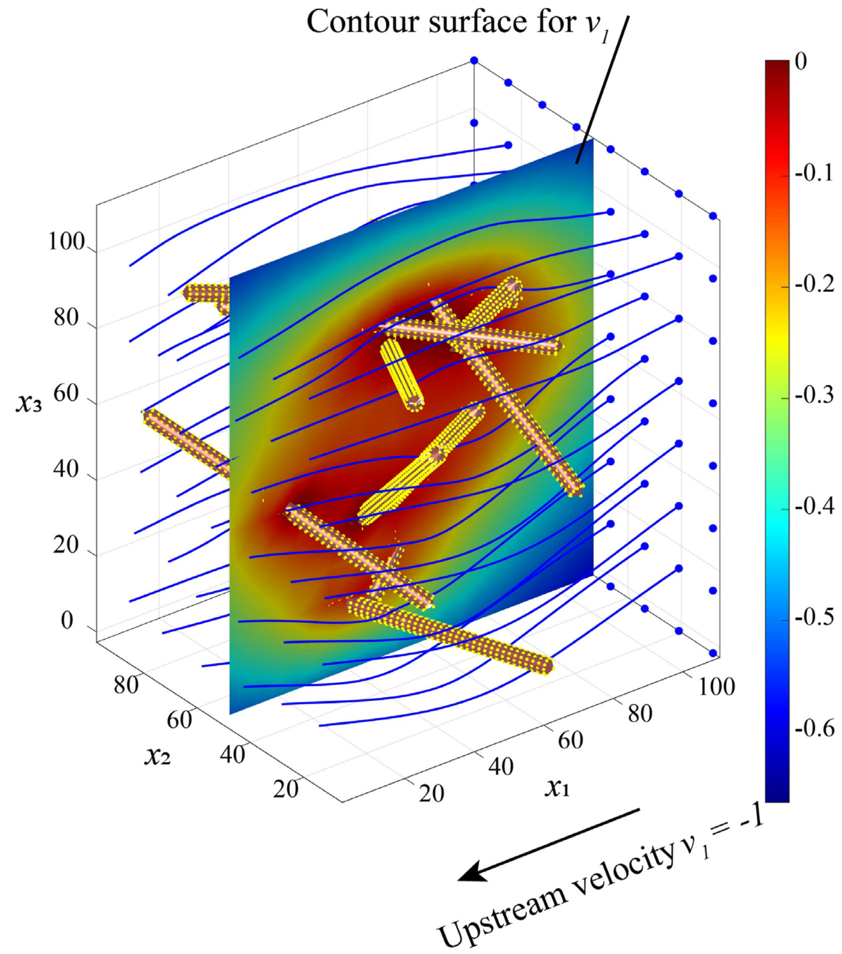
Nearest-neighbor Regularized Stokeslet

Implementation

Nearest-neighbor Regularized Stokeslet (MATLAB)

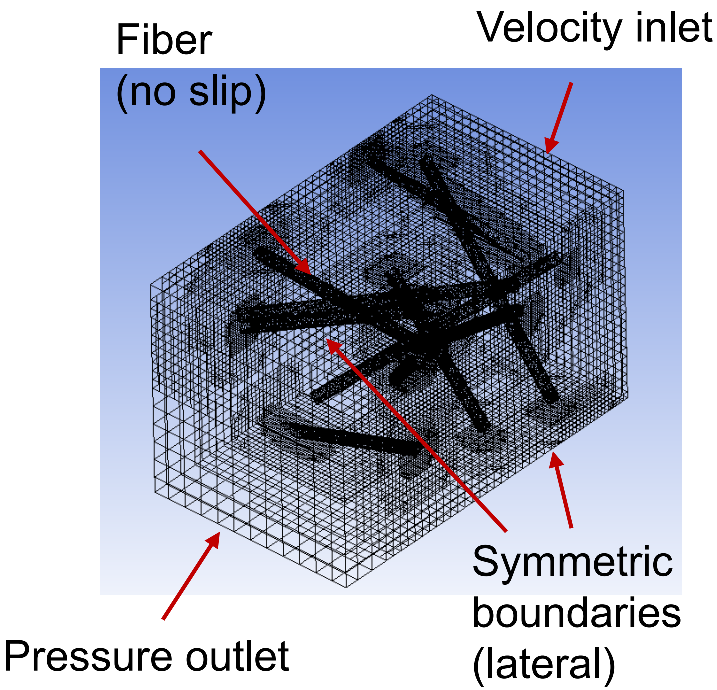
Cylinder number: 10
Number of quadrature nodes: 10,900
Number of force nodes: 820

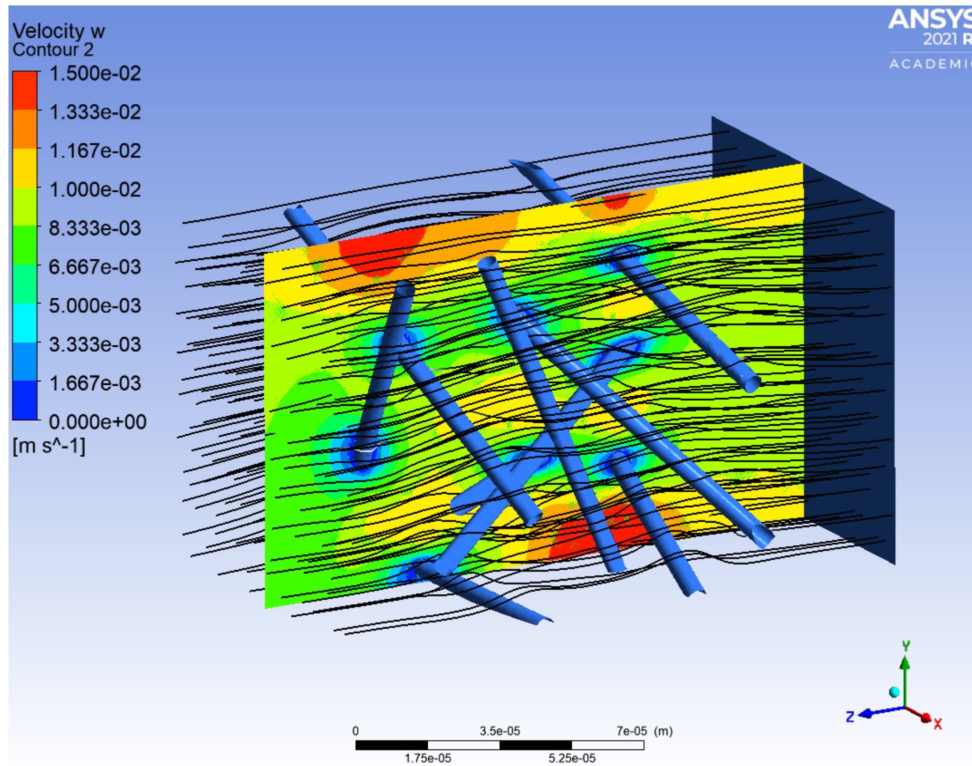
Easy to implement open boundary



Implementation

Finite-volume method (ANSYS Fluent)
(Geometry is a little different from last slide)





Cylinder number: 10
Number of nodes: 300,000

Easier to implement but it takes time to generate a satisfying mesh

Project summary

- Studied and reviewed regularized Stokeslet boundary integral methods
- Implemented Stokes flow through randomly oriented cylinders by using boundary integral and finite volume methods with MATLAB and ANSYS Fluent, respectively

References

[1] Cortez R. The method of regularized Stokeslets. SIAM Journal on Scientific Computing. 2001;23(4):1204-25.

[2] Cortez R, Fauci L, Medovikov A. The method of regularized Stokeslets in three dimensions: analysis, validation, and application to helical swimming. Physics of Fluids. 2005 Mar 23;17(3):031504.

[3] Smith DJ. A boundary element regularized Stokeslet method applied to cilia-and flagella-driven flow. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2009 Dec 8;465(2112):3605-26.

[4] Smith DJ. A nearest-neighbour discretisation of the regularized stokeslet boundary integral equation. Journal of Computational Physics. 2018 Apr 1;358:88-102.

Thank you!