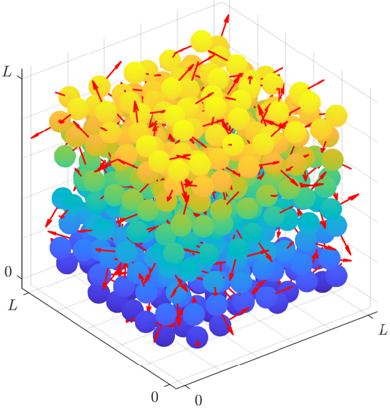
Closely interacting rigid particles in Stokes flow

Anna Broms
Numerical analysis, KTH, Stockholm



Setting: The Stokes mobility problem

Given, for each particle:
Position and orientation,
External force and torque,

Compute:
Angular and translational velocity of all particles.

Setting: The Stokes mobility problem

Given, for each particle:
Position and orientation,
External force and torque,

Compute:
Angular and translational velocity of all particles.

Motivation: Study systems of many Brownian 3D particles dynamically (particles assumed to be rigid, non-deformable)

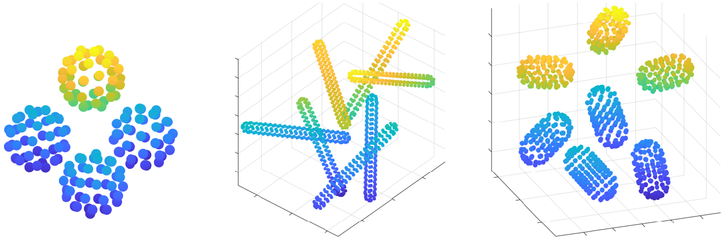
Goal: Coarsely resolved particles with controllable accuracy and SPD mobility matrices

How?

- A BIE double layer formulation with QBX – what if we only aim for a few digits of accuracy?

How?

- A rigid multiblob approach (as developed by A. Donev et al.): spherical blobs on a particle surface interact via the RPY-tensor and are constrained to move as a rigid body.



How?

- A rigid multiblob approach (as developed by A. Donev et al.): spherical blobs on a particle surface interact via the RPY-tensor and are constrained to move as a rigid body.

$$\sum_j \boldsymbol{N}_{ij}\boldsymbol{\lambda}_j = \boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{x}_i - \boldsymbol{q}_p) \quad \text{for all } \boldsymbol{x}_i \in \Gamma_p$$

$$\sum_{\boldsymbol{x}_i \in \Gamma_p} \boldsymbol{\lambda}_i = \boldsymbol{f}_p$$

$$\sum_{\boldsymbol{x}_i \in \Gamma_p} (\boldsymbol{x}_i - \boldsymbol{q}_p) \times \boldsymbol{\lambda}_i = \boldsymbol{t}_p$$

- The RPY-tensor $\boldsymbol{N}_{ij}$, a regularised Stokeslet, depends on the blob radius a_h.

RPY-quadrature

$$\int_{\Gamma_p} \boldsymbol{N}_{ij}(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{\lambda}_j(\boldsymbol{y})\,\mathrm{d}S_y = \boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{x}-\boldsymbol{q}_p) \quad \boldsymbol{x} \in \Gamma_p$$

$$\int_{\Gamma_p} \boldsymbol{\lambda}(\boldsymbol{y})\,\mathrm{d}S_y = \boldsymbol{f}_p$$

$$\int_{\Gamma_p} (\boldsymbol{y}-\boldsymbol{q}_p) \times \boldsymbol{\lambda}(\boldsymbol{y})\,\mathrm{d}S_y = \boldsymbol{t}_p,$$

RPY-quadrature

$$\int_{\Gamma_p} \boldsymbol{N}_{ij}(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{\lambda}_j(\boldsymbol{y})\,\mathrm{d}S_y = \boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{x}-\boldsymbol{q}_p) \quad \boldsymbol{x} \in \Gamma_p$$

$$\int_{\Gamma_p} \boldsymbol{\lambda}(\boldsymbol{y})\,\mathrm{d}S_y = \boldsymbol{f}_p$$

$$\int_{\Gamma_p} (\boldsymbol{y}-\boldsymbol{q}_p) \times \boldsymbol{\lambda}(\boldsymbol{y})\,\mathrm{d}S_y = \boldsymbol{t}_p,$$

with

$$\int_{\Gamma_p} \boldsymbol{N}_{ij}\boldsymbol{\lambda}_j\,\mathrm{d}S_y = \int_{D(\boldsymbol{x})} \left(\boldsymbol{S}_{ij}(\boldsymbol{x},\boldsymbol{y}) + \frac{2a_h^2}{3}\boldsymbol{D}_{ij}(\boldsymbol{x},\boldsymbol{y}) \right) \boldsymbol{\lambda}_j(\boldsymbol{y})\,\mathrm{d}S_y + \\ + \frac{1}{8\pi\mu} \int_{D^c(\boldsymbol{x})} \left(\left(\frac{4}{3a_h} - \frac{3r}{8a_h^2} \right) \boldsymbol{I} + \frac{\boldsymbol{r}\boldsymbol{r}^T}{8a_h^2} \right) \boldsymbol{\lambda}(\boldsymbol{y})\,\mathrm{d}S_y$$

Three difficulties

- **Coarse discretisations - badly resolved layer densities.**
  Idea:
  - Split density into $\lambda_{\text{peaked}} + \lambda_{\text{smooth}}$
  - Expand each in some basis. Solve for expansion coefficients instead of point values on the particle surfaces.
  - Literature?

Three difficulties

- **Coarse discretisations - badly resolved layer densities.**
  Idea:
  - Split density into $\lambda_{\text{peaked}} + \lambda_{\text{smooth}}$
  - Expand each in some basis. Solve for expansion coefficients instead of point values on the particle surfaces.
  - Literature?
- **Nearly singular integrands.**
  Idea: QFS inspired
  - Move source surface to the interior of the particle and impose bc on particle surface.
  - Want accurate RBM, not primarily interested in the layer density.

Three difficulties

- **Coarse discretisations - badly resolved layer densities.**
  Idea:
  - Split density into $\lambda_{\text{peaked}} + \lambda_{\text{smooth}}$
  - Expand each in some basis. Solve for expansion coefficients instead of point values on the particle surfaces.
  - Literature?
- **Nearly singular integrands.**
  Idea: QFS inspired
  - Move source surface to the interior of the particle and impose bc on particle surface.
  - Want accurate RBM, not primarily interested in the layer density.
- **Quadrature that preserves positive-definiteness of the mobility matrix.**
  Idea: Inspiration from Galerkin methods such as "inner product preserving Nyström discretisation" by Bremer (2012)?

The RPY-tensor

The block $\boldsymbol{N}_{ij}$ describes the motion on blob i resulting from a given force on blob j, governed by the far field approximation

$$\boldsymbol{N}_{ij} \approx \eta^{-1} \left(\boldsymbol{I} + \frac{1}{6} a_h^2 \nabla_{\boldsymbol{x}}^2 \right) \left(\boldsymbol{I} + \frac{1}{6} a_h^2 \nabla_{\boldsymbol{y}}^2 \right) \mathbb{G}(\boldsymbol{r}_{ij}), \tag{1}$$

with η the viscosity and

$$\mathbb{G}(\boldsymbol{r}) = \frac{1}{8\pi r} \left(\boldsymbol{I} + \frac{\boldsymbol{r} \otimes \boldsymbol{r}}{r^2} \right) \tag{2}$$

the Stokeslet.

The RPY-tensor

The RPY-tensor, corrected for overlapping blobs so that the resulting mobility $\boldsymbol{N}_{ij}$ is positive-definite, takes the form

$$\boldsymbol{N}_{ij} = \frac{1}{6\pi\eta a_h} \begin{cases} C_1(r_{ij})\boldsymbol{I} + C_2(r_{ij})(\boldsymbol{r}_{ij} \otimes \boldsymbol{r}_{ij})/r_{ij}^2, & r_{ij} > 2a_h, \\ C_3(r_{ij})\boldsymbol{I} + C_4(r_{ij})(\boldsymbol{r}_{ij} \otimes \boldsymbol{r}_{ij})/r_{ij}^2, & r_{ij} \leq 2a_h, \end{cases} \tag{3}$$

with

$$\begin{aligned} C_1(r) &= \frac{3a_h}{4r} + \frac{a_h^3}{2r^3}, \qquad & C_2(r) &= \frac{3a_h}{4r} - \frac{3a_h^3}{2r^3}, \\ C_3(r) &= 1 - \frac{9r}{32a_h}, & C_4(r) &= \frac{3r}{32a_h}, \end{aligned} \tag{4}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$ is the center-center vector for two blobs i and j. The diagonal blocks simply reduce to $(6\pi\eta a_h)^{-1}\boldsymbol{I}$, i.e. the well-known translational part of the mobility matrix for a single sphere.