Supplemental Material: Discrete Particle Simulation Method

Adrien Lefauve & David Saintillan

1 Governing equations

In this section, we briefly introduce the discrete model for a two-dimensional suspension of a large number N of rigidly confined self-propelled asymmetric particles coupled via hydrodynamic interactions. The individual dynamics of a swimmer with center of mass located at $\mathbf{R}(t)$ and orientation $\mathbf{p}(t)$ embedded in a two-dimensional flow field with velocity $\mathbf{u}(\mathbf{R})$ follows [1]:

$$\begin{aligned} \dot{\mathbf{R}} &= v_s\,\mathbf{p} + \mu_\perp\,(\mathbf{I}-\mathbf{pp})\cdot\mathbf{u} + \mu_\parallel\,\mathbf{pp}\cdot\mathbf{u}, && \text{(1a)} \\ \dot{\mathbf{p}} &= \nu\,(\mathbf{I}-\mathbf{pp})\cdot\mathbf{u} + \nu'\,(\mathbf{I}-\mathbf{pp})\cdot\nabla\mathbf{u}\cdot\mathbf{p}, && \text{(1b)} \end{aligned}$$

where we shall set $\mu_\parallel = \mu_\perp = 1$ and $\nu' = 0$ to focus on the influence of the novel polar orientational coefficient ν, which is specific to fore-aft asymmetric particles and is the dominant orientation mechanism for long-wave dynamics. The velocity $\mathbf{u}(\mathbf{R})$ results from the perturbation flows driven by the particles. More specifically, the equations of motion for each particle i ($i = 1, ..., N$) are written

$$\begin{aligned} \dot{\mathbf{R}}_i &= v_s\,\mathbf{p}_i + \mathbf{u}(\mathbf{R}_i), && \text{(2a)} \\ \dot{\mathbf{p}}_i &= \nu\,(\mathbf{I}-\mathbf{p}_i\mathbf{p}_i)\cdot\mathbf{u}(\mathbf{R}_i), && \text{(2b)} \end{aligned}$$

and are coupled via $\mathbf{u}(\mathbf{R}_i)$, which is obtained as the linear superposition of the dipolar velocity fields induced by all other $j \neq i$ particles [1]:

$$\mathbf{u}(\mathbf{R}_i) = \sum_{\substack{j=1 \\ j\neq i}}^{N} \mathbf{u}^d(\mathbf{R}_i|\mathbf{R}_j,\boldsymbol{\sigma}_j) = \sum_{\substack{j=1 \\ j\neq i}}^{N} \frac{\sigma}{2\pi|\mathbf{R}_{ij}|^2}(2\hat{\mathbf{R}}_{ij}\hat{\mathbf{R}}_{ij} - \mathbf{I})\cdot\Big(\dot{\mathbf{R}}_j - \mathbf{u}(\mathbf{R}_j)\Big). \tag{3}$$

Here, $\mathbf{R}_{ij} = \mathbf{R}_i - \mathbf{R}_j$, $\hat{\mathbf{R}}_{ij} = \mathbf{R}_{ij}/|\mathbf{R}_{ij}|$, and we note that obtaining $\mathbf{u}(\mathbf{R}_i)$ requires the knowledge of $\dot{\mathbf{R}}_j$ and $\mathbf{u}(\mathbf{R}_j)$ that remain unknown, motivating the need for an approximation that we shall later address.

2 Simulation method

Our algorithm is designed to perform the time evolution of the system of $4n$ ODEs (2a)-(2b) coupled with the $2n$ equations (3). For this purpose, the program handles the position $\mathbf{R}_i^n$, orientation $\mathbf{p}_i^n$ and dipole

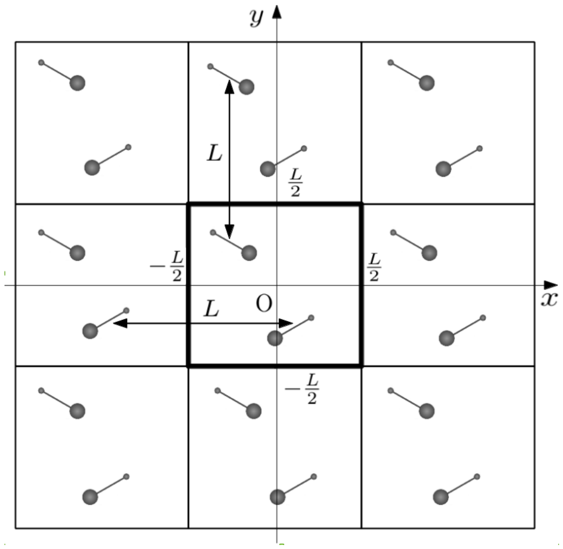

FIG. 1: Image system for periodic boundary conditions in a square box of size L. $N = 2$ particles in the computational domain (bold line) are represented with $N_{img} = 1$ image in each direction, for a total of $(N_{img}+1)^2-1 = 8$ images per particle.

moment $\boldsymbol{\sigma}_i^n$ of N point particles (subscript $i = 1, ..., N$ refers to particle number, whereas superscript n denotes the time step). To model an initially uniform and isotropic phase, positions are randomly initialized in the computational domain of linear size L following a uniform law: $\mathbf{R}_i^0 \sim \mathcal{U}([-L/2, L/2]^2)$ while orientations are randomized using $\theta_i^0 \sim \mathcal{U}([0, 2\pi])$ and $\mathbf{p}_i^0 = (\cos\theta_i^0, \sin\theta_i^0)$. The initial dipole moments are taken to be $\boldsymbol{\sigma}_i^0 = \sigma v_s \mathbf{p}_i^0$. The domain assumes periodic boundary conditions, naturally requiring the introduction of a set of image particles (hereafter referred to as *periodic images*). As illustrated in Fig. 1, N_{img} images in each direction are simply deduced from the N actual particles by $\pm L$ translations along x and y.

At each time step n, the fluid velocity $\mathbf{u}(\mathbf{R}_i)$ is computed by summing over the interactions due to

- the $N-1$ particles $j \neq i$ inside the computational domain,
- their $(N-1)[(2N_{img}+1)^2-1]$ periodic images,
- the $(2N_{img}+1)^2-1$ own periodic images of particle i,

representing a total of $N(2N_{img}+1)^2-1$ terms:

$$\mathbf{u}(\mathbf{R}_i^n) = \Bigg[\underbrace{\sum_{\substack{j=1\\ j\neq i}}^{N} \mathbf{G}(\mathbf{R}_{ij}^n)}_{N-1 \text{ real particles}} + \underbrace{\sum_{j=1}^{N} \Bigg(\sum_{\substack{p=-N_{img}\\ p\neq 0}}^{N_{img}} \sum_{\substack{q=-N_{img}\\ q\neq 0}}^{N_{img}} \mathbf{G}(\mathbf{R}_{ij}^n + pL\mathbf{e}_x + qL\mathbf{e}_y) \Bigg)}_{N_{img} \text{ periodic images in } x \text{ and } y.} \Bigg] \cdot \boldsymbol{\sigma}_j^n, \tag{4}$$

where $\mathbf{e}_x, \mathbf{e}_y$ are the Cartesian unit vectors and the Green tensor $\mathbf{G}$ takes the following form:

$$\mathbf{G}(\mathbf{R}) = \frac{1}{2\pi|\mathbf{R}|^2}(2\hat{\mathbf{R}}\hat{\mathbf{R}} - \mathbf{I}). \tag{5}$$

Making the necessary approximation $\boldsymbol{\sigma}_j^n \approx \boldsymbol{\sigma}_j^{n-1} = \dot{\mathbf{R}}_j^{n-1} - \mathbf{u}(\mathbf{R}_j^{n-1})$, $\mathbf{u}(\mathbf{R}_i^n)$ may be obtained and particle i advanced in time using a fourth-order Runge-Kutta (RK4) time marching scheme. The time increment δt is small enough such that a particle swimming at speed v_s during δt travels less that 10% of the average interparticle distance $\sqrt{L^2/N}$: $\delta t < L/(10 v_s \sqrt{N})$. Rotational diffusion is finally included by adding to the angular velocity $\dot{\mathbf{p}}_i$ the stochastic vector $\sqrt{2d/\delta t}\,(\mathbf{I} - \mathbf{p}_i\mathbf{p}_i)\cdot\mathbf{n}$ where δt is the time increment and $n_x, n_y \sim \mathcal{N}(0,1)$ are two Gaussian-distributed independent components.

We note that direct interactions captured by the first term in equation (4) are singular, which could lead to unphysical blow-up in simulations. To prevent blow-up, we regularize direct interactions by approximating the Green tensor as

$$\mathbf{G}(\mathbf{R}) = \frac{1}{2\pi(|\mathbf{R}|+\varepsilon)^2}(2\hat{\mathbf{R}}\hat{\mathbf{R}} - \mathbf{I}), \tag{6}$$

where ε is a small user-defined parameter, which should scale with the particle size $\sqrt{\sigma}/2$. Note that this regularization introduces a weak artificial fluid compressibility in the vicinity of particles, which can lead to unphysical dynamics if ε is chosen too large. To ensure that the dynamics are not affected by the regularization, we performed a series of simulations for values of ε ranging from $0.01\sqrt{\sigma}$ to $1.0\sqrt{\sigma}$, and saw no significant change in dynamics when $\varepsilon \lesssim 0.5\sqrt{\sigma}$, which is the value we used in all simulations shown in the paper.

Most simulations were typically run with $N_{img} = 200$ images in each directions (x and y) for greater accuracy, though convergence has been observed for lower values. In theory, this would require the computation of $4N[(2N_{img}+1)^2 - 1] \approx 6 \times 10^5 N$ interactions to get each $\mathbf{u}(\mathbf{R}_i)$ (hence $6 \times 10^5 N^2$ at each time step). It follows that simulating systems with thousands of swimmers on a single-processor workstation is only possible provided that we use an accelerated algorithm that we shall briefly introduce.

3 Accelerated algorithm for the interactions of periodic images

The general philosophy behind most accelerated algorithms is to evaluate directly the dominant interactions and approximate the remaining ones. We apply this idea to compute the sum of interactions due to one particle j and all its images with a single particle i, using the following algorithm:

- Search the closest particle to i among j and its closest 8 images inside all adjacent boxes. In other words, find $\mathbf{R}_{ij}^{min}$, the vector $\mathbf{R}_{ij} + pL\mathbf{e}_x + qL\mathbf{e}_y$ of minimal length with $(p,q) \in \{-1,0,1\}^2$ (see Fig. 2a). Compute directly the Green's tensor $\mathbf{G}(\mathbf{R}_{ij}^{min})$ using Eq. (5).

- All other $(2N_{img}+1)^2 - 1$ interactions may be deduced from $\mathbf{R}_{ij}^{min}$ by simple translations. Therefore, if we denote by $\mathbf{G}^{far}$ the following sum of Green's tensors:

$$\mathbf{G}^{far}(\mathbf{R}) = \sum_{\substack{p=-N_{img}\\ p\neq 0}}^{N_{img}} \sum_{\substack{q=-N_{img}\\ q\neq 0}}^{N_{img}} \mathbf{G}(\mathbf{R} + pL\mathbf{e}_x + qL\mathbf{e}_y), \tag{7}$$

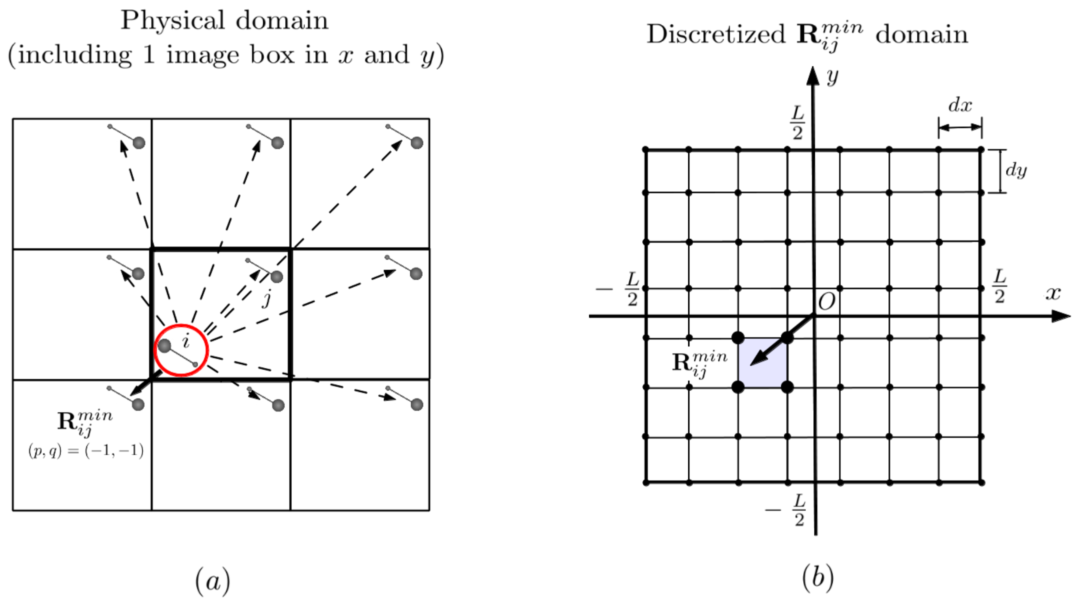

FIG. 2: (a) Compute directly $\mathbf{G}(\mathbf{R}_{ij}^{min})$ giving the interaction with the closest particle from i (in this case, the bottom-left image $(p,q)=(-1,-1)$). (b) Approximate the far-field Green's tensor $\mathbf{G}^{far}$ due to other particles $(p,q)\neq(-1,-1)$: interpolate the value at $\mathbf{R}_{ij}^{min}$ using nearest grid values of the discrete $\mathbf{G}^{far}$.

the problem is reduced to the computation of $\mathbf{G}^{far}(\mathbf{R}_{ij}^{min})$. We note that $\mathbf{G}^{far}(\mathbf{R})$ is a smooth function since the singular behavior of the Green's function is captured by the first term on the right-hand side of equation (4). Our method therefore consists in approximating the value of this far-field Green's tensor at $\mathbf{R}_{ij}^{min}$ using a discrete tensor precomputed on a grid. The domain of $\mathbf{R}_{ij}^{min}$, namely $[-L/2, L/2]^2$, is first discretized into N_{grid}^2 grid points and the values of $\mathbf{G}^{far}$ computed at each grid point before the start of the simulation. Finally, $\mathbf{G}(\mathbf{R}_{ij}^{min})$ is approximated when needed during the simulation using a bilinear interpolation between the nearest grid values of the generic discrete $\mathbf{G}^{far}$ tensor (see Fig. 2b). A typical number of $N_{grid}=250$ was used in our simulations for accuracy.

- Once this is done for all $j\neq i$, the velocity induced at $\mathbf{R}_i$ is then simply obtained by :

$$\mathbf{u}(\mathbf{R}_i)=\sum_{\substack{j=1\\ j\neq i}}^{N}\left(\mathbf{G}(\mathbf{R}_{ij}^{min})+\mathbf{G}^{far}(\mathbf{R}_{ij}^{min})\right)\cdot\boldsymbol{\sigma}_j^n. \tag{8}$$

The obvious benefit of this algorithm is that the discrete far-field Green's tensor *need only be evaluated once* before the actual simulation. At each RK4 stage and for each physical particle pair (i,j), the cost of summing over $(2N_{img}+1)^2-1$ interactions is reduced to a simple bilinear interpolation, significantly reducing computation time by a factor of $4[(2N_{img}+1)^2-1]$, i.e. typically 6×10^5. Note that despite their much greater efficiency, more complex strategies to evaluate Green's tensors with slower decay (e.g. $\sim 1/r$ in 3D) such as the smooth particle-mesh Ewald algorithm [2] are not judged necessary given

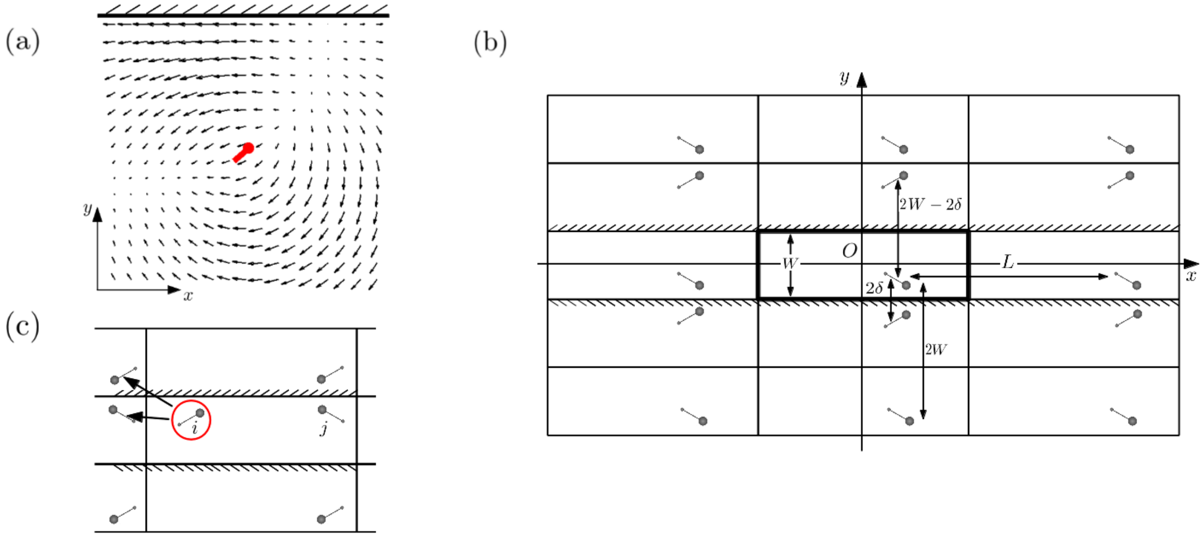

FIG. 3: (a) Dipolar flow generated by a swimmer is modified near a wall to guarantee no-penetration (field computed from simulations). (b) Image system in a narrow channel for a particle located at a distance δ of the lower wall, consisting of a series of periodic images and a series of symmetric images with different orientation in the y direction. In x, periodic images for all of these must be added. (c) Interactions at i due to the two closest images of j (periodic and symmetric) are computed directly.

the relatively fast decay of the interactions in our geometry ($\sim 1/r^2$ in 2D).

4 Image system for simulations in a narrow channel with rigid walls

This section explains how our simulations in domains periodic in the x direction (length L) and rigidly confined in the y direction (width W) have been conducted. Since the 2D flow in the xy plane is potential (Hele-Shaw flow), the no-slip condition for viscous flow cannot be enforced at the walls in y ($y = \pm W/2$), explaining why we focus on the implementation of a no-penetration condition typical of potential flows. As evidenced in Fig. 3a, no-penetration modifies the flow generated by a swimmer in the vicinity of the wall. Considering a particle with orientation $\mathbf{p}$ located at $y = -W/2 + \delta$ (at a distance δ from a confining wall, as sketched in Fig. 3b), the normal component of the dipolar flow it drives at the walls can be cancelled by introducing an image particle at the same distance on the other side of the wall ($y - 2\delta$), with orientation $\mathbf{p}^* = (p_x, -p_y)$, i.e. with angle $-\theta$ (see Fig. 3b). By doing that, we however induced a non-zero velocity through the virtual wall lying at $y = -W/2 - W$, which we cancel using the exact same strategy: we add an image of orientation $(\mathbf{p}^*)^* = \mathbf{p}$, located at $y - 2W$. Carrying out this method of reflection, we find that two distinct series of images are needed in the y direction the cancel the normal component of the flow exactly on both walls:

- a set of *symmetric images* with orientation $\mathbf{p}^*$ at locations $y \pm 2nW - 2\delta$ (for physical particles located at $y < 0$, as in Fig. 3a) or $y \pm 2nW + 2\delta$ (for $y > 0$),

- a set of *periodic images* with orientation $\mathbf{p}$ at locations $y \pm 2nW$.

In the x direction, the usual periodic boundary conditions appropriate for an infinitely long canal require the usual set of periodic images at locations $x \pm mL$ for each image introduced in y. The accelerated algorithm presented in the previous section has been modified to implement this new type of boundary conditions. To compute the interaction due to particle j on particle i, we now not only compute directly the interaction due to the closest periodic image of j, but also the one due its closest symmetric image, which can potentially be very close to i if both i and j are lying near the same wall (see Fig. 3c). The far-field contributions due to all other particles is then approximated following the same strategy based on the use of a discrete Green's tensor, whose computation requires special care. Due to the combination of both types of images (periodic and symmetric), we note that the pre-computation of two discrete Green's tensors corresponding to the two sets of images is needed in this case.

References

[1] T. Brotto, J.-B. Caussin, E. Lauga, and D. Bartolo, Phys. Rev. Lett. **110**, 038101 (2013).

[2] D. Saintillan, E. Darve, and E. S. G. Shaqfeh, Phys. Fluids **17**, 033301 (2005).