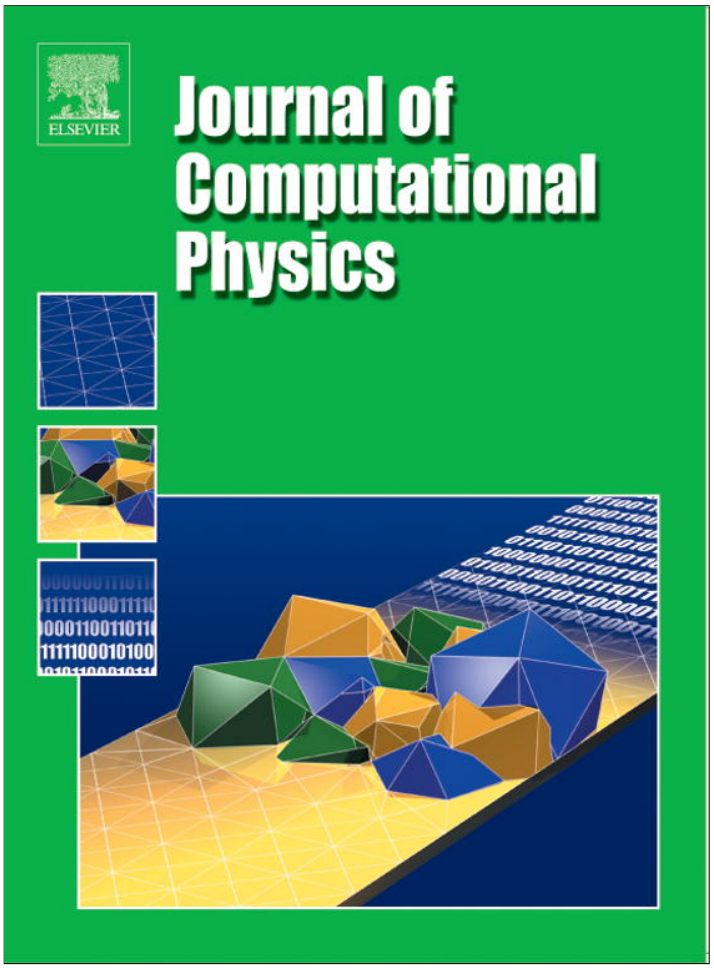

(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Journal of Computational Physics 236 (2013) 187–202

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A regularization method for the numerical solution of periodic Stokes flow

Karin Leiderman [a,*], Elizabeth L. Bouzarth [c], Ricardo Cortez [d], Anita T. Layton [b]

[a] University of California, Merced, School of Natural Sciences, Applied Mathematics Unit, Merced, CA 95343, United States
[b] Duke University, Department of Mathematics, Durham, NC 27708, United States
[c] Furman University, Department of Mathematics, Greenville, SC 29613, United States
[d] Tulane University, Department of Mathematics, New Orleans, LA 70118, United States

ARTICLE INFO

Article history:
Received 23 January 2012
Received in revised form 30 June 2012
Accepted 21 September 2012
Available online 13 October 2012

Keywords:
Stokes flow
Periodic Stokeslet
Regularization
Ewald summation

ABSTRACT

We introduce a regularization method that gives a smooth formulation for the fundamental solution to Stokes flow driven by an infinite, triply-periodic array of point forces. With this formulation, the velocity at any spatial location may be calculated, including at and very near the point forces; these locations typically lead to numerical difficulties due to the singularity within the Stokeslet when using other methods. For computational efficiency, we build upon previous methods in which the periodic Stokeslet is split into two rapidly decaying sums, one in physical space and one in reciprocal, or Fourier, space. We present two validation studies of our method. First, we compute the drag coefficient for periodic arrays of spheres with a variety of concentrations of sphere packings; and second, we prescribe a force density onto a periodic array of spheres, compute the resulting nearby velocity field, and compare these velocities to those computed using an immersed boundary method formulation. The drag coefficients computed with our method are within 0.63% of previously published values. The velocity field comparison shows a relative error of about 0.18% in the L_2-norm. We then apply our numerical method to a periodic arrangement of sinusoidal swimmers. By systematically varying their spacing in three directions, we are able to explore how their spacing affects their collective swimming speed.

1. Introduction

Substantial effort has been directed to developing analytical and computational techniques for simulating fluid flow driven by a periodic array of point forces. In the limit of zero Reynolds number, the inertial acceleration and advection terms in the Navier–Stokes equations can be neglected, and the resulting fluid motion can be described using the simpler Stokes equations. Examples of applications can be found in many physical and biological systems, including the creeping flow through model filters [1], porous media [2], and periodic suspensions of solid particles and liquid droplets [3–5].

Numerical methods based on boundary integrals are frequently used to compute Stokes flow. However, for flow driven by a periodic array of point forces, the development of efficient numerical methods involves a number of challenges. In particular, in three spatial dimensions, flow arising from a single point force, as given by the Stokeslet, decays as $1/r$, where r denotes the distance from the point force. Thus, a direct summation over a periodic array diverges. Hasimoto overcame this challenge by solving the periodically-forced Stokes equations with appropriate boundary conditions, instead of using a direct

* Corresponding author.
E-mail addresses: kleiderman@ucmerced.edu (K. Leiderman), liz.bouzarth@furman.edu (E.L. Bouzarth), rcortez@tulane.edu (R. Cortez), alayton@math.duke.edu (A.T. Layton).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.09.035

summation, and derived the pressure and velocity in the form of Fourier series [6]. This approach, while convergent, is somewhat problematic for practical computations due to the slow convergence of the Fourier series.

A method of circumventing this challenge stems from problems in electrostatics where the governing equation is Poisson rather than Stokes, but the fundamental solution decays similarly, as $1/r$. Ewald summation was first introduced in 1921 [7] as a technique to sum long-range Coulomb interactions efficiently. The idea is to recast a slowly and conditionally convergent series into a sum of two exponentially decaying series: one in real space (short-range interactions) and one in Fourier space (long-range interactions). Furthermore, the convergence of each series is controlled by one single parameter [7].

Ewald sums may be computed explicitly using a strict cutoff for each series or tuning the convergence parameter so that only the real space series is summed, but these methods are slow computationally; a sum over the N particles and all of their periodic images gives $O(N^2)$ complexity. Fourier-based particle-mesh methods have been introduced to decrease the complexity of the problem. In these methods, the real space sum is evaluated explicitly using a cutoff, but the reciprocal sum is computed over reciprocal vectors (via interpolation to a 3D grid) more efficiently by using FFT. This reduces the complexity to $O(N \log N)$. Excellent reviews of these methods for the Poisson problem can be found in [8,9].

Hasimoto incorporated Ewald's summation into the problem for periodic Stokes flow by replacing the Stokes velocity (Fourier series solution) with two complementary, rapidly converging series, one in real space and the other in Fourier space [6]. Slightly different forms of these series were later introduced by Beenakker [10] and then summarized and extended by Pozrikidis [11]. Techniques adapting the particle-mesh method have been developed for the triply periodic Stokes flow problem [12,13]. More recently, Lindbo and Tornberg [14] developed a spectrally accurate method (all errors decay at least as fast as $e^{-\alpha K^2}$ as K grows) in which they avoid interpolation and instead use convolutions with suitably scaled Gaussians.

While the methods discussed above deal with point forces, we are interested in simulating Stokes flow driven by forces singularly supported along a surface Γ. Although one may numerically integrate the Stokeslet over Γ, the Stokeslet, both in free space and for periodic flow, has a singularity. Owing to that singularity, if fluid pressure and velocity values are desired near the surface, where the integrals are nearly singular, the pressure and velocity values computed by means of a standard quadrature (e.g., the trapezoidal rule) may be highly inaccurate.

To reduce quadrature errors, the Stokeslet can be regularized via the introduction of a small cutoff parameter in the kernels. That is the basis of the method of regularized Stokeslets for free-space Stokes flow [15,16]. In this study, we first transform the periodic Stokeslet using an Ewald summation formulation from [11]. We do not focus on reducing the $O(N^2)$ complexity of the Ewald sum, but rather introduce a mesh-free regularization method that gives a smooth formulation for the Ewald sums which allows for pressure and velocity to be calculated on or very near a surface Γ.

2. Green's function for Stokes flow

At zero Reynolds number, the motion of an incompressible viscous fluid due to a concentrated point force is governed by the singularly forced Stokes equations

$$-\nabla p + \mu \Delta \mathbf{u} = -\mathbf{g}\delta(\hat{\mathbf{x}}), \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

where p and $\mathbf{u}$ are the hydrostatic pressure and velocity of the fluid, μ is the viscosity of the fluid, $\mathbf{g}$ is an arbitrary constant external force acting on the fluid, and $\hat{\mathbf{x}} = \mathbf{x} - \mathbf{x_0}$. The Green's function for the Stokes equations is the Stokeslet, which represents solutions to (1) and (2) at location $\mathbf{x}$. One can think of $\mathbf{x}$ as a point of observation and $\mathbf{x_0}$ as the location of a point source where the distance between them is defined by $r = ||\mathbf{x} - \mathbf{x_0}||$. In free space with three spatial dimensions, the Stokeslet is defined by

$$\mathcal{S}(\hat{\mathbf{x}}) = \frac{\mathbf{I}}{r} + \frac{\hat{\mathbf{x}}\hat{\mathbf{x}}}{r^3}. \tag{3}$$

The solution to Eqs. (1) and (2) is then given by

$$\mathbf{u}(\mathbf{x}) = \frac{1}{8\pi\mu}\mathcal{S}(\hat{\mathbf{x}})\mathbf{g}.$$

Another form of the Stokeslet can be written using the following derivation. Note that the Green's function of the Laplacian operator in three dimensions is $G(r) = -\frac{1}{4\pi r}$ and solves the equation $\Delta G(r) = \delta$. Similarly, the Green's function for the Biharmonic operator is $B(r) = -\frac{r}{8\pi}$ and solves the equation $\Delta^2 B(r) = \delta$. First replace δ in Eq. (1) with the equivalent form of G. Taking the divergence and enforcing incompressibility gives a Poisson equation for pressure. Using this solution, one finds

$$\nabla p = -\frac{1}{4\pi}\nabla\left(\mathbf{g}\cdot\nabla\left(\frac{1}{r}\right)\right).$$

Substituting the expressions for ∇p and δ into Eq. (1) gives

$$\mu\Delta\mathbf{u} = -\frac{1}{4\pi}(-\Delta + \nabla\nabla)\left(\frac{1}{r}\right)\mathbf{g}.$$

Inverting the Laplacian operator on the left hand side and commuting operators on the right hand side leads to

$$\mathbf{u} = \frac{1}{\mu}(-\Delta + \nabla\nabla)\Delta^{-2}\delta(\hat{\mathbf{x}})\mathbf{g}.$$

Recalling the definition of B from above, we see that in addition to Eq. (3), the Stokeslet may also be written as

$$\mathcal{S}(\hat{\mathbf{x}}) = 8\pi(-\Delta\mathbf{I} + \nabla\nabla)B(r).$$

The Stokeslet describes the velocity field induced by the point source anywhere except at the location of the source point. This restriction is due to the singular behavior that emerges as the observation point approaches the point source. Given a distribution of point sources on a surface in three dimensions, since the Stokes equations are linear, one may describe the induced velocity field by simply integrating the contributions of each point source over the surface. Computing the velocity on or very near the surface again leads to singularities or near singularities.

The method of regularized Stokeslets, developed by Cortez and collaborators [15,16], provides a way to circumvent these singularities by smoothing out the concentrated forces. Mathematically this is achieved by solving the modified equations:

$$\begin{aligned} -\nabla p + \mu\nabla^2\mathbf{u} &= -\mathbf{g}\phi_\epsilon(\hat{\mathbf{x}}),\\ \nabla\cdot\mathbf{u} &= 0. \end{aligned}$$

where the Dirac delta function in Eq. (1) is replaced with a radially symmetric blob function, $\phi_\epsilon(r)$, which has the property that $\int_0^\infty r^2\phi_\epsilon(r)dr = 1/4\pi$. For a given blob function, one can derive the corresponding Green's functions, $G_\epsilon(r)$ and $B_\epsilon(r)$, such that $\Delta^2 B_\epsilon(r) = \Delta G_\epsilon(r) = \phi_\epsilon(r)$. The smooth force leads to the regularized Stokeslet

$$\begin{aligned} \frac{1}{8\pi}\mathcal{S}_\epsilon(\hat{\mathbf{x}}) &= (-\Delta\mathbf{I} + \nabla\nabla)B_\epsilon(r),\\ &= H_1(r)\mathbf{I} + H_2(r)\hat{\mathbf{x}}\hat{\mathbf{x}}, \end{aligned}$$

where $H_1(r)$ and $H_2(r)$ are defined as

$$H_1(r) = -\frac{B_\epsilon'(r)}{r} - B_\epsilon''(r), \quad H_2(r) = \frac{rB_\epsilon''(r) - B_\epsilon'(r)}{r^3}. \tag{4}$$

Although $H_1(r)$ and $H_2(r)$ look singular, the form of $B_\epsilon(r)$ ensures that they are not and thus, the velocity may be defined everywhere. Eqs. (22) and (23) in Section 4.1 are examples of two blobs and their corresponding functions $H_1(r)$ and $H_2(r)$. The first example clearly shows $H_1(r)$ and $H_2(r)$ to be nonsingular. In the second example, after a Taylor series expansion about $r = 0$, one can see that $H_1(r)$ and $H_2(r)$ are also nonsingular.

3. Green's function for periodic Stokes flow

Our work is an extension of the formulation presented in [11], which is summarized in this section. Consider the steady motion of a Stokes fluid past a triply-periodic array of point forces located at the vertices of a three-dimensional lattice defined by the vectors

$$\mathbf{X_n} = i_1\mathbf{a}_1 + i_2\mathbf{a}_2 + i_3\mathbf{a}_3 \quad (i_1, i_2, i_3 = 0, \pm1, \pm2, \ldots), \tag{5}$$

where $\mathbf{a}_1, \mathbf{a}_2$, and $\mathbf{a}_3$ are the basis vectors that determine the shape of the lattice. The motion of the fluid is governed by the periodically forced Stokes equations:

$$\begin{aligned} -\nabla p + \mu\Delta\mathbf{u} &= -\sum_{\mathbf{n}}\mathbf{g}\delta(\hat{\mathbf{x}}_n),\\ \nabla\cdot\mathbf{u} &= 0. \end{aligned}$$

where $\hat{\mathbf{x}}_n = \mathbf{x} - \mathbf{x_0} - \mathbf{X_n}$ and a summation over $\mathbf{n}$ represents the summation over the infinite lattice extending in all periodic directions. We note that one point force is located at $\mathbf{x_0}$, the nth force is located at $\mathbf{x_n} = \mathbf{x_0} + \mathbf{X_n}$, so that $\hat{\mathbf{x}}_0 = \mathbf{x} - \mathbf{x_0}$. The velocity at any spatial location $\mathbf{x}$ is the sum of the contributions from every point force in the lattice:

$$\mathbf{u}(\mathbf{x}) = \frac{1}{8\pi\mu}\sum_{\mathbf{n}}\mathcal{S}(\hat{\mathbf{x}}_n)\mathbf{g}. \tag{6}$$

The fundamental difficulty with the summation in Eq. (6) is the $1/r$ decay so that the direct sum over the lattice is divergent. However, there are ways around a direct summation. Following the method of Pozrikidis [11], to find a rapidly convergent form of Eq. (6) the Stokeslet, $\mathcal{S}$, may be decomposed in the following way: $\mathcal{S}(\hat{\mathbf{x}}) = 8\pi(-\Delta\mathbf{I} + \nabla\nabla)B(r) = \Theta(\hat{\mathbf{x}}) + \Phi(\hat{\mathbf{x}})$ where

$$\begin{aligned} \Theta(\hat{\mathbf{x}}) &= 8\pi(-\mathbf{I}\Delta + \nabla\nabla)[B(r)\operatorname{erfc}(\xi r)],\\ \Phi(\hat{\mathbf{x}}) &= 8\pi(-\mathbf{I}\Delta + \nabla\nabla)[B(r)\operatorname{erf}(\xi r)], \end{aligned} \tag{7}$$

and ξ is an arbitrary positive splitting parameter whose value dictates the contribution from each sum. As ξ tends to zero, there is a larger contribution from $\Theta(\hat{\mathbf{x}})$ and as ξ grows larger than 1, there is a larger contribution from $\Phi(\hat{\mathbf{x}})$. The purpose of

this splitting is so that the sum over Θ exhibits Gaussian decay in real space and performing the sum over Φ in Fourier space results in exponential decay. This is seen by directly applying the operator in (7):

$$\Theta(\hat{\mathbf{x}}) = \frac{C(\xi r)}{r}\mathbf{I} + \frac{D(\xi r)}{r^3}\hat{\mathbf{x}}\hat{\mathbf{x}},$$

where

$$C(x) = \operatorname{erfc}(x) + \frac{2x}{\sqrt{\pi}}(2x-3)\exp(-x^2)$$

and

$$D(x) = \operatorname{erfc}(x) + \frac{2x}{\sqrt{\pi}}(1-2x^2)\exp(-x^2).$$

One can see the Gaussian decay of the contributions of Θ. The slow $1/r$ decay of the Stokeslet $\mathcal{S}$ is contained in the Φ term. However, since this term is smooth, an alternative is to sum the Fourier transform of Φ over the reciprocal lattice in Fourier space where $\mathbf{k}$, the transform variable, is defined as:

$$\mathbf{k} = j_1\mathbf{b}_1 + j_2\mathbf{b}_2 + j_3\mathbf{b}_3, \quad \mathbf{k}\cdot\mathbf{a}_l = j_l, \quad (l = 1,2,3),$$

and $\mathbf{a}_1, \mathbf{a}_2$, and $\mathbf{a}_3$ are the same as in (5). The vectors $\mathbf{b}_1, \mathbf{b}_2$, and $\mathbf{b}_3$ are the basis vectors for the reciprocal lattice defined by

$$\mathbf{b}_1 = 2\pi\frac{\mathbf{a}_2\times\mathbf{a}_3}{\tau}, \quad \mathbf{b}_2 = 2\pi\frac{\mathbf{a}_3\times\mathbf{a}_1}{\tau}, \quad \mathbf{b}_3 = 2\pi\frac{\mathbf{a}_1\times\mathbf{a}_2}{\tau}, \tag{8}$$

where τ is the volume of a lattice cell in physical space: $\tau = \mathbf{a}_1\cdot(\mathbf{a}_2\times\mathbf{a}_3)$. It is useful to now consider Poisson's formula for summation on an infinite lattice of points, which states that:

$$\sum_{\mathbf{n}} F(\mathbf{X}_{\mathbf{n}}) = \frac{1}{\tau}\sum_{\mathbf{m}}\widehat{F}(\mathbf{k}_{\mathbf{m}}). \tag{9}$$

The three-dimensional Fourier transform of Φ with respect to $\mathbf{X}$ is

$$\hat{\Phi}(\mathbf{k},\hat{\mathbf{x}}_0) = \frac{8\pi}{|\mathbf{k}|^2}\left(\frac{\mathbf{k}\mathbf{k}}{|\mathbf{k}|^2} - \mathbf{I}\right)\left[\left(1 + \frac{\omega^2}{4} + \frac{\omega^4}{8}\right)\exp\left(-\frac{\omega^2}{4}\right)\right]\exp(i\mathbf{k}\cdot\hat{\mathbf{x}}_{\mathbf{0}}),$$

where $\omega = |\mathbf{k}|/\xi$. The sum of this function over the reciprocal lattice, which leads to an expression for the sum in physical space as defined in (9), exhibits Gaussian decay. Therefore, the Green's function for the periodic problem is then a sum of Gaussian terms over all point forces:

$$\sum_{\mathbf{n}}\mathcal{S}(\hat{\mathbf{x}}_n) = \sum_{\mathbf{n}}\Theta(\hat{\mathbf{x}}_n) + \frac{1}{\tau}\sum_{|\mathbf{k}_{\mathbf{m}}|\neq 0}\hat{\Phi}(\mathbf{k}_{\mathbf{m}},\hat{\mathbf{x}}_0),$$

but with one sum in physical space and one sum in reciprocal space. Note that the sum of Φ in Fourier space contains no singularity and that Θ contains the entire $1/r$ singularity. In practice these sums are truncated and upper bounds are usually given in terms of the number of periodic layers or shells around a main lattice cell that are included in the sums. Truncation error estimates are given in [14] as functions of number of layers in physical and reciprocal space and the splitting parameter, ξ.

4. Regularization method

Consider the velocity due to a periodic collection of smoothed point forces placed on an infinite lattice in three dimensional Stokes flow:

$$-\nabla p + \mu\Delta\mathbf{u} = -\sum_{\mathbf{n}}\mathbf{g}\phi_\epsilon(\hat{\mathbf{x}}_n), \tag{10}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{11}$$

The summation over $\mathbf{n}$ again represents the summation over the infinite lattice extending in all periodic directions. At first glance, it may seem that this problem may be solved by simply choosing any blob function, deriving the regularized Stokeslet and decomposing it in the same way as in Section 3. We now consider the ramifications of this choice. The direct decomposition of the regularized Stokeslet would be $\mathcal{S}_\epsilon(\hat{\mathbf{x}}) = 8\pi(-\Delta\mathbf{I} + \nabla\nabla)B_\epsilon(r) = \Theta_\epsilon(\hat{\mathbf{x}}) + \Phi_\epsilon(\hat{\mathbf{x}})$ where

$$\Theta_\epsilon(\hat{\mathbf{x}}) = 8\pi(-\mathbf{I}\Delta + \nabla\nabla)[B_\epsilon(r)\operatorname{erfc}(\xi r)],$$
$$\Phi_\epsilon(\hat{\mathbf{x}}) = 8\pi(-\mathbf{I}\Delta + \nabla\nabla)[B_\epsilon(r)\operatorname{erf}(\xi r)].$$

For any choice of blob, the sum of Θ_ϵ converges with Gaussian decay due to the derivatives of $\mathrm{erfc}(\xi r)$. When calculating the direct sum of Φ_ϵ in physical space, we again have a $1/r$ term that decays very slowly. Moreover, to determine the convergence rate of the sum of Φ_ϵ in Fourier space requires the calculation of complicated integrals, even with a careful choice of blob. However, this can be avoided altogether.

Instead of this direct decomposition, we consider splitting Eqs. (10) and (11) into:

$$-\nabla p_1 + \mu\Delta \mathbf{u}_1 = -\sum_{\mathbf{n}} \mathbf{g}\delta(\hat{\mathbf{x}}_\mathbf{n}),$$
$$\nabla \cdot \mathbf{u}_1 = 0,$$

and

$$-\nabla p_2 + \mu\Delta \mathbf{u}_2 = -\sum_{\mathbf{n}} \mathbf{g}[\phi_\epsilon(\hat{\mathbf{x}}_\mathbf{n}) - \delta(\hat{\mathbf{x}}_\mathbf{n})],$$
$$\nabla \cdot \mathbf{u}_2 = 0.$$

By linearity of the Stokes equations, $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$ with a Green's function $\mathcal{S}_p$ and $p = p_1 + p_2$. Observe that $\mathbf{u}_1$ and p_1 are known quantities with expressions given in [11]. We note here again that in the expression for $\mathbf{u}_1$, the singularity is entirely contained within the physical space calculation (Θ), while there is no singularity within the Fourier space calculation since Φ is smooth. Our goal is to remove the singularity and replace it with a regularized term, *all within the physical space sum*, which allows for calculation of the velocity anywhere in the fluid including at the location of the point force, without any modification to the summation in Fourier space. The solutions for $\mathbf{u}$ and p are given by

$$\mathbf{u} = \mathbf{u}_1 + \frac{1}{8\pi\mu}\sum_{\mathbf{n}}[\mathcal{S}_\epsilon(\hat{\mathbf{x}}_n) - \mathcal{S}(\hat{\mathbf{x}}_n)]\mathbf{g}, \tag{12}$$
$$p = p_1 + \frac{1}{8\pi}\sum_{\mathbf{n}}[\nabla G_\epsilon(\hat{\mathbf{x}}_n) - \nabla G(\hat{\mathbf{x}}_n)]\cdot\mathbf{g}.$$

Now consider the periodic Green's function $\mathcal{S}_p$ for $\mathbf{u}$ in this framework:

$$\begin{aligned}\mathcal{S}_p(\hat{\mathbf{x}}) &= \sum_{\mathbf{n}}(\Theta(\hat{\mathbf{x}}_n) + \Phi(\hat{\mathbf{x}}_n) + \mathcal{S}_\epsilon(\hat{\mathbf{x}}_n) - \mathcal{S}(\hat{\mathbf{x}}_n)),\\ &= \sum_{\mathbf{n}}\underbrace{(\Theta(\hat{\mathbf{x}}_n) + \mathcal{S}_\epsilon(\hat{\mathbf{x}}_n) - \mathcal{S}(\hat{\mathbf{x}}_n))}_{\Theta_p} + \frac{1}{\tau}\sum_{|\mathbf{k}_\mathbf{m}|\neq 0}\hat{\Phi}(\mathbf{k}_\mathbf{m}, \hat{\mathbf{x}}_0).\end{aligned} \tag{13}$$

For a particular $\mathbf{x_0}$ within the main lattice cell (represented by $\mathbf{n} = \mathbf{0}$), the singularity is entirely contained within that cell, so to get an estimate on the difference in singularities between Θ and $\mathcal{S}$, we now split Θ_p into the $\mathbf{n} = \mathbf{0}$ cell and all other lattice cells:

$$\begin{aligned}\Theta_p &= \sum_{\mathbf{n}}\left(\frac{C(\xi r_\mathbf{n})}{r_\mathbf{n}}\mathbf{I} + \frac{D(\xi r_\mathbf{n})}{r_\mathbf{n}^3}\hat{\mathbf{x}}_\mathbf{n}\hat{\mathbf{x}}_\mathbf{n} + 8\pi H_1(r_\mathbf{n})\mathbf{I} + 8\pi H_2(r_\mathbf{n})\hat{\mathbf{x}}_\mathbf{n}\hat{\mathbf{x}}_\mathbf{n} - \frac{1}{r_\mathbf{n}}\mathbf{I} - \frac{\hat{\mathbf{x}}_\mathbf{n}\hat{\mathbf{x}}_\mathbf{n}}{r_\mathbf{n}^3}\right)\\ &= \sum_{\mathbf{n}}\left(\frac{C(\xi r_\mathbf{n})}{r_\mathbf{n}}\mathbf{I} + \frac{D(\xi r_\mathbf{n})}{r_\mathbf{n}^3}\hat{\mathbf{x}}_\mathbf{n}\hat{\mathbf{x}}_\mathbf{n} + E(r_\mathbf{n})\mathbf{I} + F(r_\mathbf{n})\hat{\mathbf{x}}_\mathbf{n}\hat{\mathbf{x}}_\mathbf{n}\right)\\ &= \sum_{\mathbf{n}\neq\mathbf{0}}\left(\frac{C(\xi r_\mathbf{n})}{r_\mathbf{n}}\mathbf{I} + \frac{D(\xi r_\mathbf{n})}{r_\mathbf{n}^3}\hat{\mathbf{x}}_\mathbf{n}\hat{\mathbf{x}}_\mathbf{n} + E(r_\mathbf{n})\mathbf{I} + F(r_\mathbf{n})\hat{\mathbf{x}}_\mathbf{n}\hat{\mathbf{x}}_\mathbf{n}\right)\\ &\quad + \left(\frac{C(\xi r_\mathbf{0}) - 1}{r_\mathbf{0}}\mathbf{I} + \frac{D(\xi r_\mathbf{0}) - 1}{r_\mathbf{0}^3}\hat{\mathbf{x}}_\mathbf{0}\hat{\mathbf{x}}_\mathbf{0} + 8\pi H_1(r_\mathbf{0})\mathbf{I} + 8\pi H_2(r_\mathbf{0})\hat{\mathbf{x}}_\mathbf{0}\hat{\mathbf{x}}_\mathbf{0}\right)\end{aligned} \tag{14}$$

where $E(r) = 8\pi H_1(r) - 1/r, F(r) = 8\pi H_2(r) - 1/r^3$ and H_1 and H_2 are given by Eq. (4). To restrict the addition of $\mathcal{S}_\epsilon - \mathcal{S}$ to physical space requires that the decay of $\mathcal{S}_\epsilon - \mathcal{S}$ be at least as fast as the decay of Θ, i.e., Gaussian. Note that $\mathcal{S}_\epsilon - \mathcal{S} = 8\pi(-\Delta\mathbf{I} + \nabla\nabla)(B_\epsilon(r) - B(r))$, so we can ensure this behavior with the right choice of blob function. We describe the choice of blob in the next section. First we prove that we indeed have a smooth solution.

Our claim is that the expression for Θ_p does not contain any singularities. The singularity is restricted to the main lattice box where $\mathbf{n} = \mathbf{0}$ and $H_1(r_\mathbf{0})\mathbf{I} + H_2(r_\mathbf{0})\hat{\mathbf{x}}_\mathbf{0}\hat{\mathbf{x}}_\mathbf{0}$ is nonsingular by construction, so the only term left to consider is $[(C(\xi r_\mathbf{0}) - 1)/r_\mathbf{0}]\mathbf{I} + [(D(\xi r_\mathbf{0}) - 1)/r_\mathbf{0}^3]\hat{\mathbf{x}}_\mathbf{0}\hat{\mathbf{x}}_\mathbf{0}$. These remaining expressions are also nonsingular. To explore this, consider the following:

$$\lim_{r\to 0}\frac{C(\xi r) - 1}{r} = \lim_{r\to 0}\frac{-\mathrm{erf}(\xi r) + \frac{2}{\sqrt{\pi}}(2\xi r - 3)\xi r\exp(-\xi^2 r^2)}{r} = \frac{-8\xi}{\sqrt{\pi}}.$$

Since $r_{\mathbf{0}} = |\hat{\mathbf{x}}_{\mathbf{0}}|$ and

$$\lim_{r\to 0}\frac{D(\xi r)-1}{r} = \lim_{r\to 0}\frac{-\mathrm{erf}(\xi r)+\frac{2}{\sqrt{\pi}}(1-2\xi^2 r^2)\xi r\exp(-\xi^2 r^2)}{r} = 0,$$

we have that

$$\lim_{r_{\mathbf{0}}\to 0}\left(\frac{C(\xi r_{\mathbf{0}})-1}{r_{\mathbf{0}}}\mathbf{I}+\frac{D(\xi r_{\mathbf{0}})-1}{r_{\mathbf{0}}^3}\hat{\mathbf{x}}_{\mathbf{0}}\hat{\mathbf{x}}_{\mathbf{0}}\right) = \frac{-8\xi}{\sqrt{\pi}}\mathbf{I},$$

and the term does not contain a singularity. Eq. (13) together with the expression in Eq. (14) can be used to directly compute the solution. The main benefits of this formulation are its smoothness and the fact that it can be used to calculate the velocity anywhere in the domain, including at and very near the point forces. Furthermore, we point out the expression for the $\hat{\Phi}$ needs no modification from that given in [11].

4.1. Choosing the blobs

To ensure that the only corrections necessary are those in the Θ_p function in Eq. (13), we must use a blob function such that $\mathcal{S}_\epsilon - \mathcal{S} = 8\pi(-\Delta\mathbf{I}+\nabla\nabla)(B_\epsilon(r)-B(r))$ decays with at least Gaussian behavior. This will be accomplished as long as the terms in Eq. (14) containing $E(r) = 8\pi H_1(r) - 1/r$ and $F(r) = 8\pi H_2(r) - 1/r^3$ exhibit Gaussian decay.

One possibility is to define a blob function $\phi_\epsilon(r)$ with compact support in $r \leqslant \epsilon$ with the property that $E(r) = F(r) \equiv 0$ for $r > \epsilon$. This would ensure that the only difference in the computation of the regularized velocity and the singular velocity is contained in the *local* term (within the main lattice box) contained in the last term of Eq. (14). A more general class of blob functions is one that includes non-compactly supported C^∞ blobs such that $E(r)$ and $F(r)$ exhibit Gaussian decay. This would allow the same truncation of the sum to compute Θ_p. We describe blob functions that guarantee these decay conditions for $E(r)$ and $F(r)$.

Second moment theorem. *Let $\phi_\epsilon(r)$ be a continuous bounded function that satisfies the following conditions:*

1. *$\phi_\epsilon(r)$ is radially symmetric*
2. *$\phi_\epsilon(r)$ has the decay property $|\phi_\epsilon(r)| \leqslant Cr^m\exp(-\xi^2 r^2)$ for some integer $m \geqslant 0$ and some constants $C > 0$ and $\xi > 0$.*
3. *$\int_0^\infty r^2\phi_\epsilon(r)dr = \frac{1}{4\pi}$*
4. *$\int_0^\infty r^4\phi_\epsilon(r)dr = 0$*

Then

$$|E(r)| = |8\pi H_1(r) - 1/r| \leqslant C_1 r^m\exp(-\xi^2 r^2)$$
$$|F(r)| = |8\pi H_2(r) - 1/r^3| \leqslant C_2 r^{m-2}\exp(-\xi^2 r^2)$$

for large enough r and for some constants $C_1 > 0, C_2 > 0$.

Proof. Define $M_n = \int_0^\infty r^{n+2}\phi_\epsilon(r)dr$ for $n = 0, 1, 2$. Consider the functions $G_\epsilon(r)$ and $B_\epsilon(r)$ defined as the solutions of $\Delta G_\epsilon = \phi_\epsilon$ and $\Delta B_\epsilon = G_\epsilon$ in $\mathbb{R}^3$. Since ϕ_ϵ is radially symmetric, the functions satisfy the following ordinary differential equations

$$G_\epsilon''(r) + \frac{2}{r}G_\epsilon'(r) = \frac{1}{r}(rG_\epsilon(r))'' = \phi_\epsilon(r), \tag{15}$$
$$B_\epsilon''(r) + \frac{2}{r}B_\epsilon'(r) = \frac{1}{r}(rB_\epsilon(r))'' = G_\epsilon(r), \tag{16}$$

with $G_\epsilon(0)$ and $B_\epsilon(0)$ bounded and satisfying the following conditions

$$\lim_{r\to\infty} G_\epsilon(r) + \frac{1}{4\pi r} = 0, \tag{17}$$
$$\lim_{r\to\infty} B_\epsilon(r) + \frac{r}{8\pi} = 0. \tag{18}$$

Eqs. (15) and (16) are solved by integrating $rG_\epsilon(r)$ and $rB_\epsilon(r)$ twice and using repeated integration by parts. The result is

$$\begin{aligned} G_\epsilon(r) &= -\frac{1}{r}M_0 - \int_r^\infty s\phi_\epsilon(s)ds + \frac{1}{r}\int_r^\infty s^2\phi_\epsilon(s)ds, \\ B_\epsilon(r) &= -\frac{r}{2}M_0 - \frac{r^2}{6}\int_r^\infty s\phi_\epsilon(s)ds + \frac{r}{2}\int_r^\infty s^2\phi_\epsilon(s)ds \\ &\quad -\frac{1}{2}\int_r^\infty s^3\phi_\epsilon(s)ds - \frac{1}{6r}\left(M_2 - \int_r^\infty s^4\phi_\epsilon(s)ds\right). \end{aligned} \tag{19}$$

For G_ϵ and B_ϵ to satisfy the conditions at infinity, Eqs. (17) and (18), we require $4\pi M_0 = 1$ (condition 3 of the theorem statement). All integrals in these expressions inherit a decay rate from the function ϕ_ϵ. However, $B_\epsilon(r)$ includes the slowly-decaying term $M_2/6r$, regardless of the decay properties of $\phi_\epsilon(r)$. Condition 4 eliminates this slow decay by requiring $M_2 = 0$.

We now establish the bound on the decay rate of $E(r)$ and $F(r)$. By definition, $E(r) = 8\pi H_1(r) - 1/r$ and $F(r) = 8\pi H_2(r) - 1/r^3$. From Eq. (4), these are

$$E(r) = -8\pi B_\epsilon''(r) - \frac{8\pi B_\epsilon'(r)}{r} - \frac{1}{r},$$
$$F(r) = \frac{8\pi B_\epsilon''(r)}{r^2} - \frac{8\pi B_\epsilon'(r)}{r^3} - \frac{1}{r^3}.$$

Using the expression for $B_\epsilon(r)$ in Eq. (19), we get

$$E(r) = \frac{16\pi}{3}\int_r^\infty s\phi_\epsilon(s)ds - \frac{4\pi}{r}\int_r^\infty s^2\phi_\epsilon(s)ds - \frac{4\pi}{3r^3}\int_r^\infty s^4\phi_\epsilon(s)ds, \tag{20}$$
$$F(r) = -\frac{4\pi}{r^3}\int_r^\infty s^2\phi_\epsilon(s)ds + \frac{4\pi}{r^5}\int_r^\infty s^4\phi_\epsilon(s)ds. \tag{21}$$

Note that for an integer $n \geqslant 1$, condition 2 implies that

$$\int_r^\infty s^n\phi_\epsilon(s)ds \leqslant C\int_r^\infty s^{n+m}\exp(-\xi^2 s^2)ds = C\xi^{-(n+m+1)}\int_{\xi r}^\infty z^{n+m}\exp(-z^2)dz \leqslant \tilde{C}\xi^{-2}r^{n+m-1}\exp(-\xi^2 r^2).$$

This inequality along with Eqs. (20) and (21) indicate that

$$|E(r)| \leqslant C_1\xi^{-2}r^m\exp(-\xi^2 r^2),$$
$$|F(r)| \leqslant C_2\xi^{-2}r^{m-2}\exp(-\xi^2 r^2).$$

Finally, we mention that for the case of blob functions ϕ_ϵ with compact support in $[0,\epsilon]$, all integrals in Eqs. (20) and (21) are zero when $r > \epsilon$ and we have that for $r > \epsilon, E(r) = F(r) = 0$. □

4.1.1. Examples of blobs

In the formulas below, we use the notation $z = r/\epsilon$. A blob with compact support in $[0,\epsilon]$ satisfying the conditions of the Second Moment Theorem, along with the associated functions $H_1(r)$ and $H_2(r)$, is given by

$$\phi_\epsilon(r) = \begin{cases} \frac{3465(5-13z^2)(1-z^2)^3}{1024\epsilon^3\pi}, & 0 \leqslant z < 1 \\ 0, & 1 \leqslant z \end{cases} \tag{22}$$
$$H_1(r) = \begin{cases} \frac{1}{8\pi\epsilon} + \frac{(1-z^2)(378z^8-1547z^6+2413z^4-1745z^2+565)}{1024\pi\epsilon}, & 0 \leqslant z < 1 \\ \frac{1}{8\pi r}, & 1 \leqslant z \end{cases}$$
$$H_2(r) = \begin{cases} \frac{1}{8\pi\epsilon^3} - \frac{(1-z^2)(315z^6-1225z^4+1745z^2-1027)}{1024\pi\epsilon^3}, & 0 \leqslant z < 1 \\ \frac{1}{8\pi r^3}, & 1 \leqslant z \end{cases}$$

Note that in Eq. (14), the terms containing the functions C and D decay as Gaussians in r. The terms containing $E(r)$ and $F(r)$ depend on the decay of the blob. For the blob in Eq. (22), these terms are identically zero for $\mathbf{n} \neq 0$. However, as long as the terms with $E(r)$ and $F(r)$ decay as Gaussians, their addition in Eq. (14) can be computed without affecting the truncation of the series. An example of a blob that leads to expressions for E and F that decay as Gaussians is:

$$\phi_\epsilon(r) = \frac{1}{2\pi^{3/2}\epsilon^3}(5-2z^2)\exp(-z^2) \tag{23}$$
$$H_1(r) = \frac{1}{8\pi r} + \frac{1}{4\pi^{3/2}\epsilon}\exp(-z^2) - \frac{\mathrm{erfc}(z)}{8\pi r}$$
$$H_2(r) = \frac{1}{8\pi r^3} - \frac{1}{4\pi^{3/2}\epsilon r^2}\exp(-z^2) - \frac{\mathrm{erfc}(z)}{8\pi r^3}$$

5. Validation studies and application

We present two validation studies of our numerical method as well as an application. The first problem is to compute the drag coefficient of a fixed bed of spheres with varying concentration and packings. Drag coefficients are compared to previously known results from the literature. For the first numerical test, we use two different blobs but for the second test and application, we limit the experiments to a single one. The second problem is to compute the velocity field near a bed of spheres whose motion is driven with a given force density. The velocities are compared to those computed using an

immersed boundary method for Stokes flow. Finally, we use the method presented in this study to investigate optimal spacing of a periodic arrangement of sinusoidal swimmers.

5.1. Standardizing blobs for performance comparison

The choice of regularization parameter ϵ for a given blob often seems ad hoc. A principal reason for this is that different (unscaled) blobs have different shapes. We propose a systematic way of comparing different blobs by first defining a *standardized* blob $\phi_\epsilon(r)$ as one that satisfies $\phi_1(0) = 1$; that is, the blob at $r = 0$ has unit height when $\epsilon = 1$. Following this definition, the standardized forms of the blobs in Eqs. (22) and (23) are

$$\tilde{\phi}_\epsilon(r) = \begin{cases} \frac{1}{\epsilon^3}\left(1 - \frac{13}{5}\delta_a^2 z^2\right)\left(1 - \delta_a^2 z^2\right)^3, & 0 \leqslant z < 1 \\ 0, & 1 \leqslant z \end{cases} \tag{24}$$

$$\tilde{\psi}_\epsilon(r) = \frac{1}{\epsilon^3}\left(1 - \frac{2}{5}\delta_b^2 z^2\right) e^{-\delta_b^2 z^2} \tag{25}$$

where $\delta_a = \sqrt[3]{\frac{17325}{1024\pi}}$, $\delta_b = \sqrt[3]{\frac{5}{2\pi^{3/2}}}$ and $z = r/\epsilon$. Note that $\tilde{\phi}_\epsilon(0) = \tilde{\psi}_\epsilon(0) = 1/\epsilon^3$. These are the blobs we use in this section.

5.2. Drag coefficients of cubic array of spheres

Our first test problem involves a periodic array of spheres that moves with a uniform velocity in the positive x-direction. We will compare the drag coefficient of the array using the total drag force on the spheres. We consider a sphere of radius a and discretize its surface using Spherical Centroidal Voronoi Tessellation (SCVT) and the package STRIPACK [17,18]. Using SCVT, a specified number of points are positioned on a unit sphere so that the points are well separated. Then, the Delaunay triangulation of these same points on the unit sphere is obtained using STRIPACK. We then use the centroid of each triangle as the location of the N point forces, $\mathbf{g}_i\phi_\epsilon$, where $i = 1, 2, \ldots, N$. The velocity of all points on the surface of the sphere is set to a constant vector $\mathbf{U} = (1, 0, 0)$ and we use Eq. (12) to solve for the forces. We choose the radius $a = 0.25, \epsilon$ based on the discretization, and use three different packings for the array: simple cubic, body-centered, and face-centered, which are displayed in Fig. 1. The sphere packings are defined in terms of the basis vectors for physical and reciprocal space. There is one central sphere, always in the same location for all three packings, and the location of the surrounding spheres is defined by the basis vectors. For a simple cubic packing, the basis vectors in physical space are $\mathbf{a}_1 = (d, 0, 0), \mathbf{a}_2 = (0, d, 0), \mathbf{a}_3 = (0, 0, d)$, where d is the center-to-center distance between the spheres, the reciprocal space vectors, $\mathbf{b}_1, \mathbf{b}_2$, and $\mathbf{b}_3$ are defined in Eq. (8), and the volume of one lattice cell is $\tau = d^3$. Body-centered packing is defined by the vectors

$$\begin{aligned} \mathbf{a}_1 &= d/\sqrt{3}(1, 1, -1), \quad \mathbf{b}_1 = \pi\sqrt{3}/d(1, 1, 0), \\ \mathbf{a}_2 &= d/\sqrt{3}(-1, 1, 1), \quad \mathbf{b}_1 = \pi\sqrt{3}/d(0, 1, 1), \\ \mathbf{a}_3 &= d/\sqrt{3}(1, -1, 1), \quad \mathbf{b}_1 = \pi\sqrt{3}/d(1, 0, 1), \end{aligned}$$

where the volume of one periodic box is $\tau = 4d^3/3\sqrt{3}$. Face-centered packing is defined by the vectors

$$\begin{aligned} \mathbf{a}_1 &= d/\sqrt{2}(1, 1, 0), \quad \mathbf{b}_1 = 2\pi/\sqrt{2}d(1, 1, -1), \\ \mathbf{a}_2 &= d/\sqrt{2}(0, 1, 1), \quad \mathbf{b}_1 = 2\pi/\sqrt{2}d(-1, 1, 1), \\ \mathbf{a}_3 &= d/\sqrt{2}(1, 0, 1), \quad \mathbf{b}_1 = 2\pi/\sqrt{2}d(1, -1, 1), \end{aligned}$$

where the volume of one periodic box is $\tau = d^3/\sqrt{2}$. The drag coefficient K is defined by the following equation for the drag force on each sphere in a cubic array [19,6,2]:

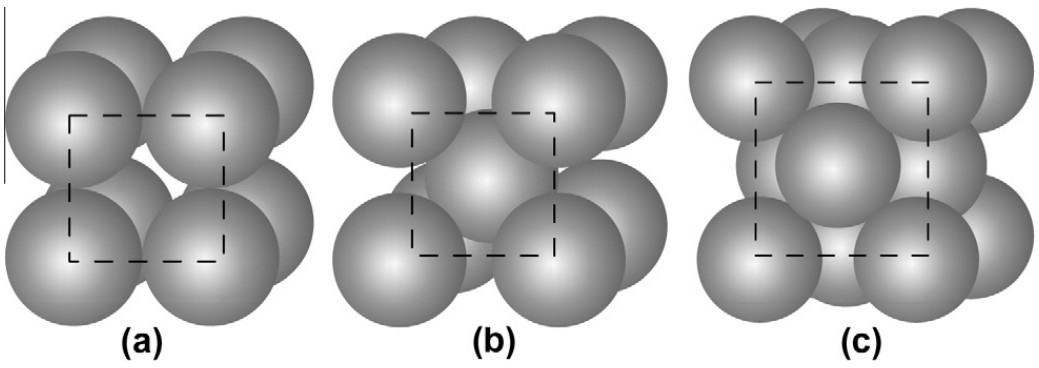

Fig. 1. Three different packings of the array of spheres: simple cubic (a), body-centered (b), and face-centered (c).

$$F = 6\pi\mu aKV,$$

where μ and V are the viscosity and average velocity of the fluid at the sphere surface, respectively. We calculate the drag coefficients, K, for arrays of spheres with simple cubic packing with five different concentrations and also for the other two packings with just one concentration. The concentration is defined by $c = 4\pi a^3/3\tau$. We use the two different blob functions from Eqs. (22) and (23) and compare these values to the values from Zick and Homsy [19], K_z. The results are shown in Tables 1 and 2 where we report the relative difference given by $|K - K_z|/\max(|K|, |K_z|)$.

By prescribing the velocity at the points on the sphere and solving for the forces at those same locations, we can find the velocity at any location in the domain. To provide insight into what the flow fields look like in each situation, in Fig. 2 we show the velocity field in the $z = 0$ plane, through the middle of the central sphere.

These comparisons show that the numerical method performs well for a variety of geometric arrays of objects with an ϵ properly chosen from the discretization and two different types of blob functions. The discretization that we use does not give identically-sized patches on the surface of the sphere that represent each force point. For the given sphere radius, we can find the average area of one of these patches. Assuming the shape of each patch is either square or circular, we can find the length of the diagonal and a radius, respectively, from the surface area of the sphere and the number of points used to discretize that surface. We find that if we choose an ϵ close to the radius of the circle ($\approx$0.0213) and half the diagonal of the square ($\approx$0.0267), we get good results. Tables 1 and 2 show the relative difference between our drag force and those from the literature. The difference tables correspond to the blobs in Eqs. (24) and (25). For all cases, we use $\epsilon = 0.0213, 0.0240$ and 0.0267. We find that for concentrations below 0.125, the choice of ϵ near the radius of the circle gives relative differences less than 1%. For all other concentrations and values of ϵ, we find the relative differences to be less than 3%.

To ensure that we are including enough terms in our sums, we systematically checked the relative difference in drag coefficients as we increased the number of layers summed in physical and reciprocal space (see Table 3). We found that summing beyond two layers in either space gives a relative difference in the value of the drag coefficient by at most 7.3×10^{-8}. Therefore, for all values reported in Table 1, we sum exactly 2 layers in each space.

5.3. Velocity comparison with immersed boundary method

In the next example, we validate our method by comparing its solution to the solution computed by the immersed boundary method [20], which is a popular numerical method for computing fluid motion driven by immersed structures. In this example, we compute the Stokes solution driven by forces singularly distributed over the surfaces of the spheres. Each sphere has radius $a = 0.25$ and is centered at $(0.5, 0.5, 0.5)$ of a unit box with periodic boundary conditions. This setup gives simple cubic packing with a concentration of 0.0654. The total surface force density $\mathbf{g}$ over the entire sphere is set to one in the positive z-direction. The sphere is discretized using the same method as in the previous sphere example except here we use 1600 points and each point force carries approximately (since the discretization is quasi-uniformly distributed) the same force density and in the same direction.

The immersed boundary method solves the Stokes Eqs. (1) and (2), with the Dirac delta function $\delta(\hat{\mathbf{x}})$ on the right-hand side of Eq. (1) replaced by a discrete (or, smoothed) delta function $\delta_h(\hat{\mathbf{x}})$ given by

$$\delta_h(\hat{\mathbf{x}}) = \frac{1}{h^3}\phi_{\mathrm{IB}}\left(\frac{\hat{x}_1}{h}\right)\phi_{\mathrm{IB}}\left(\frac{\hat{x}_2}{h}\right)\phi_{\mathrm{IB}}\left(\frac{\hat{x}_3}{h}\right)$$

$$\phi_{\mathrm{IB}}(r) = \begin{cases} \frac{1}{4}\left(1 + \cos\left(\frac{\pi r}{2}\right)\right), & |r| \leqslant 2 \\ 0, & \text{otherwise} \end{cases}$$

where $\hat{x}_1, \hat{x}_2$, and $\hat{x}_3$ are the x-, y- and z-components of $\hat{\mathbf{x}}$, respectively. The discretized form of Eq. (1), with smoothed Dirac delta function δ_h, and Eq. (2) is then reduced into a sequence of Poisson problems, one for each variable [21]

$$\Delta p = \nabla \cdot \mathbf{g}\delta_h(\hat{\mathbf{x}}), \quad \Delta \mathbf{u} = \nabla p.$$

The immersed boundary solution was computed using a $256 \times 256 \times 256$ spatial grid.

Table 1
Drag coefficients for periodic arrays with three different sphere packings and various concentrations, $\epsilon = 0.0214, 0.0228$ and 0.02425, with blob from Eq. (24). Here $N = 553$ and $\xi = \sqrt{\pi}/\tau$. P = packing, c = concentration, and RD = relative difference between K_ϵ and K_z.

P	c	K_z	$K_{\epsilon=0.0213}$	RD	$K_{\epsilon=0.0240}$	RD	$K_{\epsilon=0.0267}$	RD
SC	0.000125	1.0960	1.0944	0.0015	1.1012	0.0047	1.1064	0.0095
SC	0.008	1.5250	1.5204	0.0030	1.5250	0.0056	1.5437	0.0123
SC	0.027	2.0080	1.9987	0.0046	2.0080	0.0067	2.0391	0.0155
SC	0.064	2.8100	2.7852	0.0088	2.8100	0.0072	2.8653	0.0197
SC	0.125	4.2920	4.2019	0.0210	4.2920	0.0040	4.3954	0.0239
BC	0.125	4.4470	4.3564	0.0204	4.4671	0.0045	4.4470	0.0242
FC	0.125	4.4460	4.3557	0.0203	4.4665	0.0046	4.5542	0.0243

Table 2
Drag coefficients for periodic arrays with three different sphere packings and various concentrations, $\epsilon = 0.0214, 0.0228$ and 0.02425, with blob from Eq. (25). Here $N = 553$ and $\xi = \sqrt{\pi}/\tau$. P = packing, c = concentration, and RD = relative difference between K_ϵ and K_z.

P	c	K_z	$K_{\epsilon=0.0213}$	RD	$K_{\epsilon=0.0240}$	RD	$K_{\epsilon=0.0267}$	RD
SC	0.000125	1.0960	1.0969	0.0008	1.1030	0.0064	1.1074	0.0104
SC	0.008	1.5250	1.5253	0.0002	1.5371	0.0097	1.5455	0.0135
SC	0.027	2.0080	2.0071	0.0004	2.0276	0.0097	2.0422	0.0170
SC	0.064	2.8100	2.8019	0.0029	2.8423	0.0115	2.8713	0.0218
SC	0.125	4.2920	4.2533	0.0090	4.3502	0.0136	4.4201	0.0299

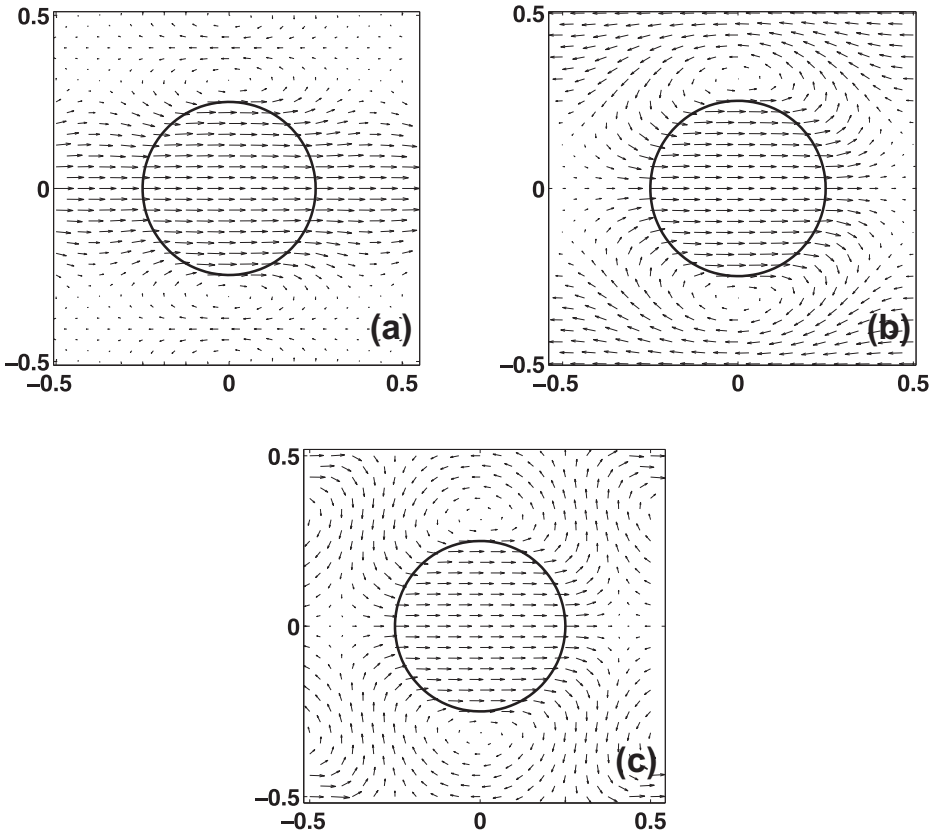

Fig. 2. Slice of the velocity field through the middle of the central sphere at $z = 0$ in the direction of the flow. The concentration is 0.125, sphere radius is 0.25, and ϵ is 0.04. The sphere consists of 553 point forces. (a) Simple cubic packing; (b) body-centered packing; (c) face-centered packing

Table 3
Number of shells to sum in physical (N) and reciprocal (M) space.

$\xi = 2.2$, Concentration = 0.125			$\xi = 0.88$, Concentration = 0.008		
N,M	N,M	Relative difference	N,M	N,M	Relative difference
1,1	1,2	4.1×10^{-5}	1,1	1,2	4.9×10^{-4}
1,1	2,1	8.9×10^{-3}	1,1	2,1	1.8×10^{-4}
1,2	2,2	8.9×10^{-3}	1,2	2,2	1.8×10^{-4}
2,1	2,2	4.1×10^{-5}	2,1	2,2	4.9×10^{-4}
2,2	2,3	1.5×10^{-12}	2,2	2,3	3.3×10^{-12}
2,2	3,2	7.3×10^{-8}	2,2	3,2	1.5×10^{-10}

The z-component of $\mathbf{u}$, computed by the regularized periodic Stokeslet method using $N = 1600$ quasi-uniformly distributed point forces, is evaluated in the $z = 0.5$ plane and exhibited in Fig. 3. Our solution agrees well with the immersed boundary solution; cross-sections of the computed velocities, evaluated at $y = 0, y = 0.25$, and $y = 0.5$ and at $z = 0.5$, appear indistinguishable in the left side of Fig. 3. Results at other locations and for other components of the velocities are similar (not shown). The relative differences of the z-component of $\mathbf{u}$, given by $|\mathbf{u}_z - \mathbf{u}_{z\mathrm{IB}}| / \max(|\mathbf{u}_z|, |\mathbf{u}_{z\mathrm{IB}}|)$, where $\mathbf{u}_{\mathrm{IB}}$ denotes the

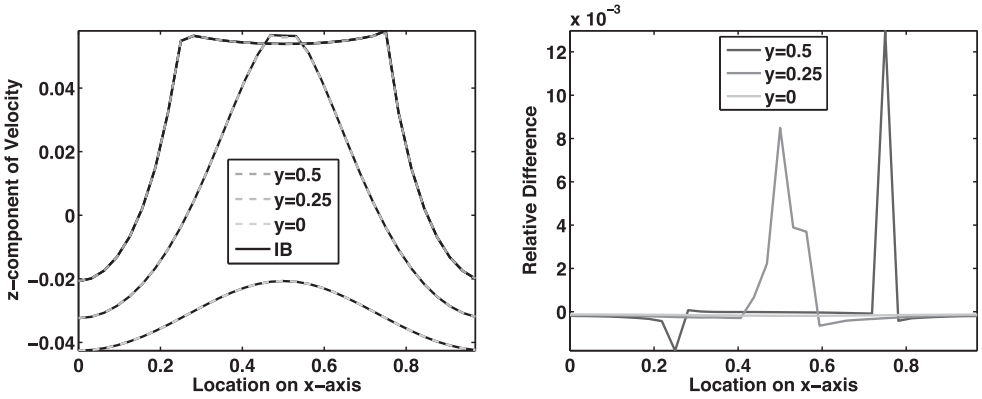

Fig. 3. Left: z-component of the velocity along the x-axis at the locations $y = 0$ (outside the sphere), $y = 0.25$ (partially through sphere), and $y = 0.5$ (directly through center of sphere), and in the plane $z = 0.5$. The dashed gray lines are the velocities computed with the method described in this paper, the black lines are velocities computed with the immersed boundary method. Right: Relative difference between the two methods used to calculate the velocities seen in the left plot.

immersed boundary velocity solution, are evaluated at selected cross-sections and shown in the right side of Fig. 3. As noted above, the discretization that we use here does not give identically-sized patches on the surface of the sphere that represent each force point. Using 1600 points on a sphere with radius 0.25, circular patches would have a radius of 0.0125 and square patches would have a length of 0.0222. With intuition from the last example and through some trial and error, we find that $\epsilon = 0.015$ gives the best agreement with the immersed boundary solution. Furthermore, we also sum two layers in physical and reciprocal space, as in the last example. The relative difference of the two velocity solutions, measured in L_2-norm, is 1.8×10^{-3}.

The velocity in the plane $x = 0.5$ was also computed, and is displayed in Fig. 4, to give insight into what the velocity field looks like around and through the sphere. In this example, the array of spheres has a concentration of 0.0654 with a simple cubic packing and so the velocity field has similar characteristics to the one depicted in Fig. 2(a).

5.4. Periodic array of sinusoidal swimmers

As an application, we consider the motion of an undulating flagellum, similar to the waveform often observed in sea-urchin spermatozoa, for example. The purpose of this example is to investigate the fluid flow around the flagellum and its swimming speed as the flagellum undulates in proximity to its periodic copies. In our simulations, we are able to change the periodic box dimensions independently so as to modify the proximity of the flagella in one or more coordinate directions.

The flagellum is represented by a single curve in space given parametrically by

$$x(s,t) = \alpha s, \quad y(s,t) = \beta s \sin(as - t), \quad z(s,t) = 0$$

for $0 \leqslant s \leqslant S_{max}$ and $t \geqslant 0$. In the simulations of this section we use the parameters $a = 2\pi/S_{max}, \beta = 0.4, \alpha = \sqrt{1-\beta^2}$ and $S_{max} = 0.75$. Given the flagellum shape at any value of time, t, its motion is given by the velocity $\mathbf{u}_f = (\partial x/\partial t, \partial y/\partial t, \partial z/\partial t) = (0, -\beta s \cos(as - t), 0)$. The following simulations are all performed at time $t = 0$. At this prescribed time and choice of β, the shape of the swimmer limits the size of the periodic box to a minimum of about 0.5 in

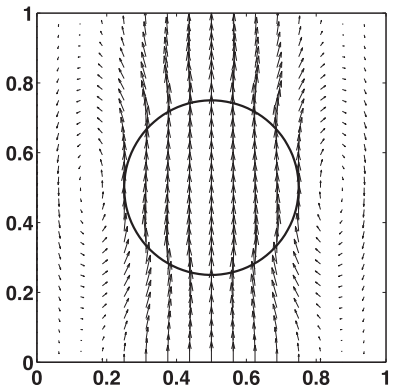

Fig. 4. Slice of the velocity field through the middle of the central sphere at $x = 0.5$ in the direction of the flow. The concentration is 0.0654 with sphere simply packed, the sphere radius is 0.25, and ϵ is 0.015. The sphere consists of 1600 point forces.

the y-direction and 0.75 in the x-direction. Because the flagellum oscillates in the $z = 0$ plane, the box length in z is unrestricted. The flagellum is discretized into 60 equidistant points and we use a value of ϵ equal to the distance between the points and the blob function in Eq. (22).

The flagellum must move in an environment where the net force and net torque are zero, so we set the body velocity to be the sinusoidal velocity plus a rigid body translation and rotation

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_f(\mathbf{x}) + \mathbf{U}_0 + (\Omega_0 \times \mathbf{x}) \tag{26}$$

for constant vectors $\mathbf{U}_0$ and Ω_0 (which are unknown). Then we obtain all the forces $\mathbf{g}_k, \mathbf{U}_0$ and Ω_0 simultaneously by solving

$$\mathbf{u}_f(\mathbf{x}_k) = \mathbf{g}_k \mathcal{S}_p(\mathbf{x}_k) - \mathbf{U}_0 - (\Omega_0 \times \mathbf{x}_k), \quad k = 1, 2, \ldots, N$$

$$0 = \sum_{k=1}^{N} \mathbf{g}_k \quad \text{zero net force}$$

$$0 = \sum_{k=1}^{N} \mathbf{x}_k \times \mathbf{g}_k \quad \text{zero net torque}$$

Note that for each k, the formula for $\mathbf{u}_f(\mathbf{x}_k)$ represents 3 equations for a total of $3N$ equations. The last two equations introduce a total of 6 more equations. The unknowns are $3N$ force components plus the 6 components of $\mathbf{U}_0$ and Ω_0. Each point moves with velocity given in Eq. (26).

Our first test is to determine the swimming velocity, $\mathbf{U}_0$, as we increase the length of periodic box in the x, y, and z directions simultaneously; these lengths are denoted L_x, L_y, and L_z, respectively. To gain intuition about what the velocity fields around the swimmers look like, the swimmer and the surrounding velocity field in the $z = 0, x = 0$, and $y = 0$ planes are shown in Figs. 5–7, respectively. The two plots in each of these figures depict the velocity field when the main lattice box has dimension $L_x = L_y = L_z = 0.75$ (left) and $L_x = L_y = L_z = 1.5$ (right). The velocity vectors are similarly scaled in each individual figure. We can see that the influence of the periodic copies of the swimmer decreases noticeably as the length of the periodic box increases from 0.75 up to 1.5.

We also report the resulting swimming velocities in the x-direction. As the swimmer goes through a periodic oscillation, it does not maintain a constant swimming speed; rather, the forward speed also oscillates between a minimum and a maximum value of the same sign (see Fig. 10). The dashed line in Fig. 8 shows the instantaneous swimming speed at $t = 0$ as a function of the side length of a cubic periodic box. These results indicate that as the box length is increased equally in all directions, the swimming velocity *decreases*, so that as the swimmers move closer to each other, they help each other swim faster. To begin to parse out which swimmers help to increase the collective speed, we ran a second test in which we fix the length in the x-direction, L_x, to be 0.75 and increase L_y and L_z in equal amounts. The solid line in Fig. 8 presents these velocities and shows that in this case as we increase L_y and L_z, the swimming velocity *increases*. The combined results of these first two tests indicate that as swimmers move closer to each other (side-to-side) and stay fixed (end-to-end) their speed will collectively slow. However, if they move closer to each other, staying an equal distance apart in all directions, until they are almost touching end-to-end, then their speed will collectively increase. We ran this same test using $L_x = 1$ and found similar behavior (not shown).

Our next tests are to independently change the length of the box in the y- and z-direction to find out which direction is more important in terms of increasing the collective speed of the swimmers. We discuss two cases: (i) $L_x = 1$, three values of L_z (0.5, 0.75 and 1), and we vary L_y in small increments, and (ii) $L_x = 1$, three values of L_y (0.5, 0.75 and 1) and we vary L_z in small increments. The resulting swimming velocities for these two cases are reported in the left and right plots of Fig. 9, respectively.

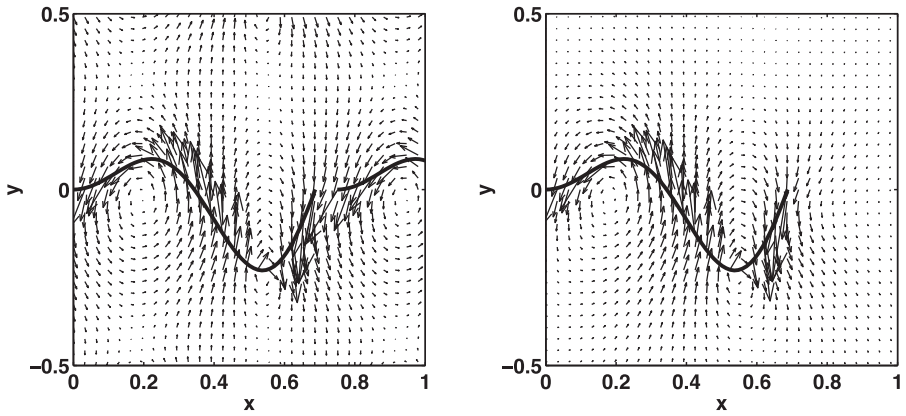

Fig. 5. Velocity field in the plane $z = 0$ with box lengths $L_x = L_y = L_z = 0.75$ (left) and $L_x = L_y = L_z = 1.5$ (right). Velocity vectors scaled by 0.03 in each plot.

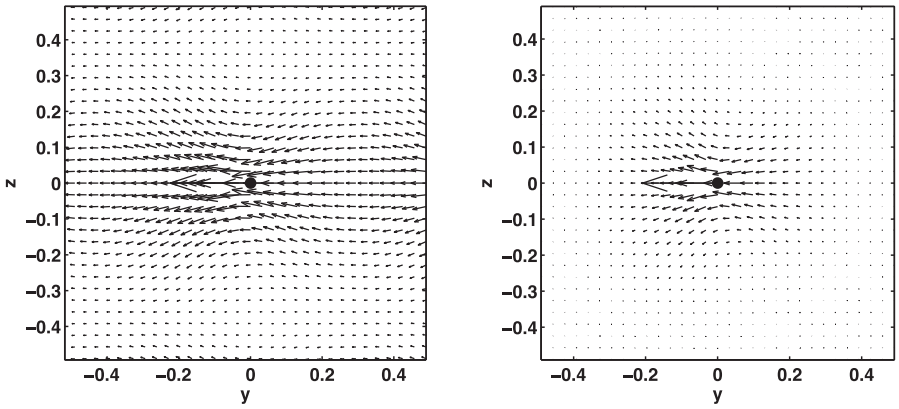

Fig. 6. Velocity field in the plane $x = 0$ with box lengths $L_x = L_y = L_z = 0.75$ (left) and $L_x = L_y = L_z = 1.5$ (right). Velocity vectors scaled by 0.015 in each plot.

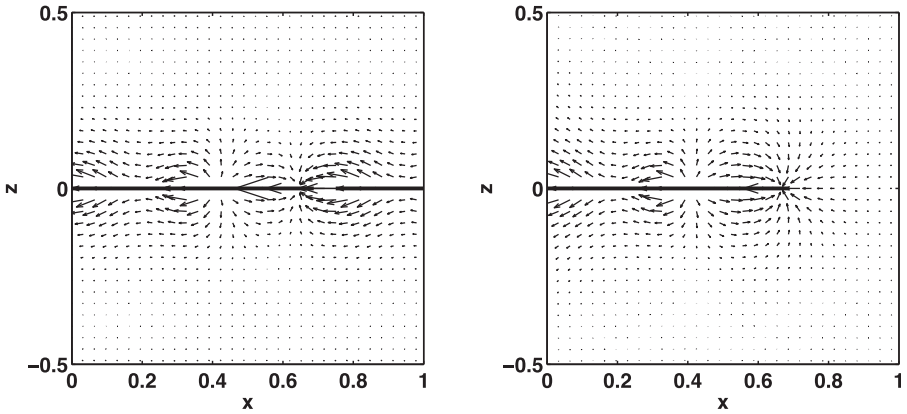

Fig. 7. Velocity field in the plane $y = 0$ with box lengths $L_x = L_y = L_z = 0.75$ (left) and $L_x = L_y = L_z = 1.5$ (right). Velocity vectors scaled by 0.01 in each plot.

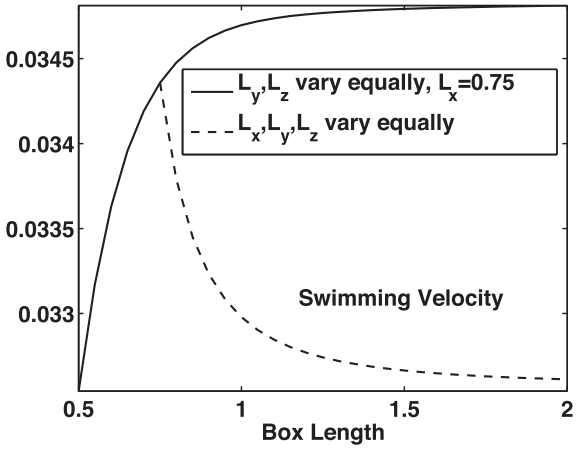

Fig. 8. The x-component of the swimming velocity, $\mathbf{U}_0$, as a function of box diameter. The dashed line represents the velocity with L_x, L_y, and L_z changing by equal amounts. The swimming velocity in this case decreases as L_x, L_y, and L_z increase. The solid line represents the velocity with L_y and L_z changing by equal amounts while L_x is a fixed value of 0.75. The swimming velocity in this case increases as L_y and L_z increase.

The left plot in Fig. 9 shows that in case (i), no matter what the fixed length of L_z, as swimmers get very close to one another in the y-direction, the general trend is that their collective speed slows. Moreover, when L_z is increased they collectively swim a little slower. In case (ii), we see that the swimming velocity is not as significantly increased or decreased by changing L_z as much as it was when changing L_y in case (i), however, we can still say something about these small changes. We see that as swimmers get closer to one another in the z-direction, they will only slow the collective speed if they are already relatively close to each other in the y-direction (the curve dips down at small values of L_z), otherwise, they slightly increase the collective speed. Furthermore, the length of L_y seems to play an important role in the speed, no matter

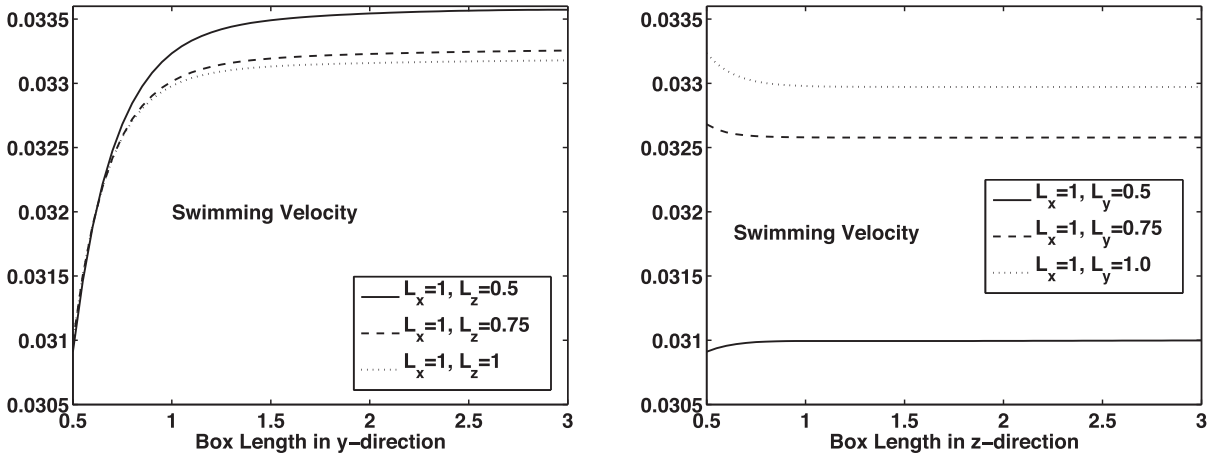

Fig. 9. The x-component of the swimming velocity, $\mathbf{U}_0$, as a function of box diameter. The lines represent different box lengths: $L_z = 0.5$ (solid), $L_z = 0.75$ (dashed), and $L_z = 1$ (dotted) with L_y varying from 0.5 up to 3.0 in smaller increments (left) and L_x is a fixed value of 1 and $L_y = 0.5$ (solid), $L_y = 0.75$ (dashed), and $L_y = 1$ (dotted) with L_z varying from 0.5 up to 3.0 in smaller increments (right).

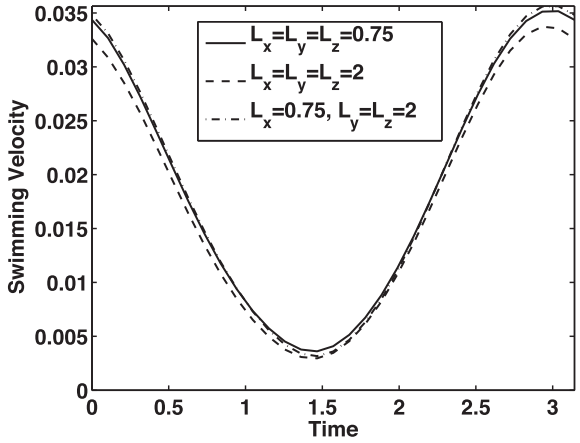

Fig. 10. Swimming velocities as a function of time. Box lengths $L_x = L_y = L_z = 0.75$ (solid), $L_x = L_y = L_z = 2$ (dashed), $L_x = 0.75, L_y = L_z = 2$ (dash-dot).

what the length of L_z. In case (i), when L_z is increased from 0.5 up to 0.75 and when the box dimensions are greater than 1 in the y-direction, the swimming speed decreases by about 1% (the relative difference between the solid and dashed lines in the left side of Fig. 9) whereas when L_y is increased from 0.5 up to 0.75, the swimming speed increases by about 5% (the relative difference between the solid and dashed lines in the right side of Fig. 9). Our data implies that the distance between swimmers in the y-direction, that is, in the plane in which they oscillate, has a larger impact on their collective swimming speed.

Finally, we kept the value of β fixed and computed the swimming speed $\mathbf{U}_0$ at various times to see if the implications of our previous test hold true for all waveforms during one oscillation. We found that during times when the swimming velocity is highest (near $t = 0$ and $t = \pi$) is when the distance between swimmers seems to matter the most. Fig. 10 shows the swimming velocity as a function of time (here we only plot $t = 0$ to $t = \pi$ since from $t = \pi$ to $t = 2\pi$ is identical) for three different periodic arrangements. The solid line depicts the swimming velocity in the case that $L_x = L_y = L_z = 0.75$. The swimming velocity is decreased by about 5% at all times if the swimmers move further apart as L_x, L_y, and L_z change equally (dashed line). Keeping L_x fixed and increasing L_y and L_z equally produces only minor changes in the swimming speed, increasing it slightly near $t = 0$ and $t = \pi$ while decreasing it slightly near $t = \pi/2$ (dash-dot line). Overall, this causes the average swimming speed over a time period to remain equal to the average speed depicted by the solid line (when $L_x = L_y = L_z = 0.75$).

The simulations show that the end-to-end distance between swimmers has a significant effect in the swimming speed of the periodic array of flagella. As the end-to-end distance decreases, the swimming speed increases. On the other hand, if the end-to-end distance L_x is kept fixed, decreasing the distance in the y-direction (the amplitude direction) has the effect of slowing down the swimmer. The distance L_z has negligible effect.

In these examples, since the periodic boxes were not always cubic, the number of layers needed to sum in reciprocal space had to be adjusted. For all of the results shown in this section, we summed 2 layers in physical space and 3 layers in reciprocal space. We did this because we found the relative difference to be on the order of 10^{-3} between 2 and 3 layers and 10^{-6} between 3 and 4 layers. Since the swimming speeds were on the order of 10^{-2} and changed in our tests on the order of 10^{-4} ,

we chose to increase the number of layers from 2 to 3 in reciprocal space. We did not find a significant difference when increasing the number in physical space.

6. Conclusions

In this paper, we have introduced a regularization method that gives a smooth formulation for the fundamental solution to Stokes flow driven by an infinite, triply-periodic array of point forces. With this formulation, the velocity at any spatial location may be calculated, including at and very near the point forces; these locations typically lead to numerical difficulties due to the singularity within the Stokeslet. We built upon methods in [11] in which the periodic Stokeslet is split into two rapidly decaying sums, one in physical space and one in reciprocal, or Fourier, space. The formulation within the reciprocal space sum is smooth while a singularity is contained within the physical space sum. In our method, we carefully remove this singularity and replace it with a regularized expression, all within the physical space sum. The benefit of this is that one need only modify the expression for this one sum while using the expression for the reciprocal sum exactly as given in [11]. The replacement of the singularity in ensured to be within the physical space with the right choice of blob function. The blob functions that we introduce in this paper are different from blob functions used for previous methods [15,16] in that they have the additional property that their second moment is identically zero. We give two examples of the functions that can be used in practice.

Another benefit to this method is the fact that it is mesh free. That is, one need not compute the velocity at all points on a regular grid of points as with the typical finite difference or finite volume methods. Here once the forces are known at the locations of the point sources (either calculated with some input velocity or prescribed), the velocity may be simply computed at any location in the domain.

We have validated our method using two numerical tests. In the first test we compute the drag coefficient for periodic arrays of spheres with a variety of concentrations of sphere packings. Here we prescribed the array of spheres with an average velocity and computed the resulting drag forces on each sphere. We show that we get very good agreement with drag coefficient values from the literature in all cases using intuitive choices for our blob parameters and with a relatively small number of periodic layers to sum in physical and reciprocal space. In our second test we prescribe a force density onto a periodic array of spheres, compute the resulting nearby velocity field, and compare these velocities to those computed using an immersed boundary method formulation. Again, we have very good agreement when we compare our solutions.

Finally, we have applied our numerical method to specific periodic arrangements of sinusoidal swimmers. By systematically varying their spacing in three directions, we were able to explore how their spacing affects their collective swimming speed. If a faster collective swimming speed is desired in this periodic arrangement, our data imply that swimmers should be closest in the end-to-end configuration and then spread out in other directions, specifically in the y-direction. Furthermore, we found that over the timespan of one oscillation, the spacing affects the swimming velocity the most when that velocity is the highest. On average, keeping the end-to-end distance fixed and spreading out in y and z will slightly increase the collective swimming velocity whereas spreading out equally in x, y, and z will slightly decrease the collective swimming speed. Our method also provides a framework for investigating the effect of different parameter values, such as β, to determine how smaller or larger flagellar oscillations affect swimming speed.

Acknowledgements

This research was partially supported by the National Science Foundation, through the Research and Training Grant DMS-0943760 to the Department of Mathematics at Duke University and through Grant DMS-07125021 to A.T. Layton.

References

[1] V. Kirsh, Stokes flow past periodic rows of porous cylinders, Theor. Found. Chem. Eng. 40 (2006) 465–471.
[2] A. Sangani, A. Acrivos, Slow flow past periodic arrays of cylinders with application to heat transfer, Int. J. Multiphase Flow 8 (1982) 193–206.
[3] Z. Li, H. Zhou, C. Pozrikidis, A numerical study of the shearing motion of emulsions and foams, J. Fluid Mech. 286 (1995) 379–404.
[4] H. Zhou, C. Pozrikidis, The flow of ordered and random suspensions of liquid drops in a channel, J. Fluid Mech. 255 (1993) 103–127.
[5] H. Zhou, C. Pozrikidis, The flow of suspensions in channels: single files of drops, Phys. Fluids A 5 (1993) 311–324.
[6] H. Hasimoto, On the periodic fundamental solutions of the stokes equations and their application to viscous flow past a cubic array of spheres, J. Fluid Mech. 5 (1959) 317–328.
[7] P.P. Ewald, Evaluations of optical and electrostatic lattice potentials, Ann. Phys. 64 (1921) 253–287.
[8] A. Toukmaji, J. Board Jr., Ewald summation techniques in perspective: a survey, Comp. Phys. Comm. 95 (1996) 73–92.
[9] M. Deserno, C. Holm, How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines, J. Chem. Phys. 109 (1998) 7678–7693.
[10] C. Beenakker, Ewald sums of the rotne-prager tensor, J. Chem. Phys. 85 (1986) 1581–1582.
[11] C. Pozrikidis, Computation of periodic green's functions of stokes flow, J. Eng. Math. 30 (1996) 79–96.
[12] A. Sierou, J.F. Brady, Accelerated stokesian dynamics simulations, J. Fluid Mech. 448 (2001) 115–146.
[13] D. Saintillan, E. Darve, E.S.G. Shaqfeh, A smooth particle-mesh Ewald algorithm for Stokes suspension simulations: The sedimentation of fibers, Phys. Fluids 17 (2005) 033301.
[14] D. Lindbo, A. Tornberg, Spectrally accurate fast summation for periodic stokes potentials, J. Comp. Phys. 229 (2010) 8994–9010.
[15] R. Cortez, The method of regularized stokeslets, SIAM J. Sci. Comput. 23 (4) (2001) 1204–1225.
[16] R. Cortez, L. Fauci, A. Medovikov, The method of regularized stokeslets in three dimensions: analysis, validation and application to helical swimming, Phys. Fluids 17 (031504) (2005) 15041–150411.

[17] Q. Du, M. Gunzburger, L. Ju, Constrained centroidal voronoi tessellations for surfaces, SIAM J. Sci. Comput. 24 (2003) 1488–1506.
[18] R. Renka, Algorithm 772, stripack: Delaunay triangulation and voronoi diagrams on the surface of a sphere, ACM Trans. Math. Soft. 23 (1997) 416–434.
[19] A. Zick, G. Homsy, Stokes flow through periodic arrays of spheres, J. Fluid Mech. 115 (1982) 13–26.
[20] C.S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.
[21] R.J. LeVeque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (3) (1997) 709–735.