

This is the accepted version of the following article:

af Klinteberg L, Tornberg A-K. (2014). Fast Ewald summation for Stokesian particle suspensions. *International Journal for Numerical Methods in Fluids*, 76(10), 669–698. doi:10.1002/flid.3953,

which has been published in final form at <http://dx.doi.org/10.1002/flid.3953> .

Fast Ewald summation for Stokesian particle suspensions

Ludvig af Klinteberg^{1,*} and Anna-Karin Tornberg¹

¹*Numerical Analysis, Department of Mathematics, KTH Royal Institute of Technology, 100 44 Stockholm*

Abstract

We present a numerical method for suspensions of spheroids of arbitrary aspect ratio which sediment under gravity. The method is based on a periodized boundary integral formulation using the Stokes double layer potential. The resulting discrete system is solved iteratively using GMRES accelerated by the spectral Ewald (SE) method, which reduces the computational complexity to $\mathcal{O}(N \log N)$, where N is the number of points used to discretize the particle surfaces. We develop predictive error estimates, which can be used to optimize the choice of parameters in the Ewald summation. Numerical tests show that the method is well conditioned and provides good accuracy when validated against reference solutions.

1 Introduction

The behavior of particles as they sediment in a suspension is a problem which is simple to formulate, yet surprisingly hard to understand and accurately describe [20]. Particle suspensions are however widely occurring both in natural and industrial processes, so the interest in them is more than just academic. A large body of work has been directed towards understanding the properties of suspensions of spherical particles, where both analytical and numerical results are available [2, 6, 9, 42, 44, 50]. Suspensions of non-spherical particles display more complex behavior and are harder to model, since the flow depends on both the position and orientation of the particles, as compared to only the position in the case of spherical particles. Using a slender-body approximation, suspensions of elongated particles (rigid fibers) have been successfully studied in the last decade, see for example [8, 22, 41]. For a recent review on the dynamics of sedimentation, we refer to Guazzelli et al. [21]. They identify that one of the current challenges lie in studying systems that are more complex than just spheres and fibers, such as suspensions of platelets and deformable particles.

Periodic boundary conditions are often used in simulations of suspensions to approximate the behavior of an infinite suspension, where there are no wall effects present. The simulation is then carried out in a box called the primary cell, which in the computations is replicated infinitely in all spatial directions.

The flow in a suspension can often be modeled as Stokes flow, due to the low Reynolds number of the fluid surrounding the sedimenting particles. This allows the flow in the fluid between the particles to be described using boundary integrals on the particles' surfaces.

*To whom correspondence should be addressed. Email: ludvigak@kth.se

Compared to a grid-based method, such as the finite element method (FEM) or the finite difference method (FDM), the dimensionality of the domain which has to be discretized (using a set of N points) is then reduced from three to two, which is attractive from a computational point of view. An additional benefit of using a boundary integral method is that movement of the particles is easily handled by translations and rotations of the surfaces, whereas a grid-based method has to either continuously fit the volume grid to the particles, or apply an alternative scheme to enforce the boundary conditions, see [24] and the references therein.

However, the benefits of reduced dimensionality come at an expense, since the long-range hydrodynamic interactions between particles then requires the contributions from all particles to be computed at every evaluation point, an operation which is expensive and has $\mathcal{O}(N^2)$ complexity. These long-range interactions can however be efficiently computed using fast methods with reduced complexity, and for the free-space problem the fast multipole method (FMM) is a well-established method which gives $\mathcal{O}(N)$ complexity [19, 45]. For the periodic problem there is an additional level of difficulty, since the long-range interactions are not only from all particles, but from all periodic images of all particles. These long-range periodic interactions can however be efficiently computed using Ewald summation methods, which are among the standard tools when computing long-range electrostatic interactions in molecular simulations (see the survey by Deserno & Holm [11]), though they have also been used to some extent in hydrodynamics [41, 44, 49]. The key element in these methods is that the long-range interactions are evaluated on a grid in reciprocal space (\mathbf{k} -space) using a fast Fourier transform (FFT), dramatically accelerating the computations and reducing the complexity to $\mathcal{O}(N \log N)$. A recent contribution to the class of fast Ewald methods is the spectral Ewald (SE) method by Lindbo & Tornberg [32, 33], developed both for the electrostatic potential and for the Stokeslet potential. In the spectral Ewald method the grid operations are carried out with spectral accuracy, allowing a minimal grid to be used in the FFT and reducing the memory and computation requirements of the method.

In this work we develop a mobility-based boundary integral method for periodic particle suspensions, using the double layer potential for Stokes flow and the spectral Ewald method to accelerate the computations. The boundary integral formulation is based on the completed double layer formulation by Power & Miranda [37], and is similar to the traction-based formulation for periodic suspensions by Fan et al. [13], which uses direct Ewald summation. The double layer formulation has the benefit that the condition number of the system is bounded [16], and furthermore that the singularity of the integral operator can be treated at a negligible cost using the method of singularity subtraction. Here we give a formulation for the periodic double layer potential which gives a zero mean flow in the domain, and also develop accurate error estimates for the Ewald summation. We apply the method to suspensions of spheroids of varying aspect ratio, and validate our results against available reference solutions.

2 Periodic Stokes flow

We are in this work concerned with three-periodic Stokes flow, where we have a primary cell of dimensions $L_1 \times L_2 \times L_3$ and volume $V = L_1 L_2 L_3$, which is replicated infinitely in all directions. Inside the primary cell we have N_p rigid bodies (particles) with surface Γ_α , volume $\Theta^{(\alpha)}$, center of mass $\mathbf{x}_c^{(\alpha)}$ and density $\rho^{(\alpha)}$, $\alpha = 1, \dots, N_p$. The surface of each particle has an outward pointing unit normal $\hat{\mathbf{n}}$ and is assumed Lyapunov smooth. In the domain D_e

exterior to the particles we have a liquid of viscosity μ and density ρ_e which is governed by the Stokes equation,

$$-\nabla P + \mu \Delta \mathbf{u} + \mathbf{f} = 0,$$

and the continuity equation,

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

where \mathbf{u} is velocity, P pressure and \mathbf{f} body force. We require the velocity field \mathbf{u} to be periodic with respect to the primary cell, which we express as $\mathbf{u}(\mathbf{x}) = \mathbf{u}(\mathbf{x} + \boldsymbol{\tau}(\mathbf{p}))$, where $\boldsymbol{\tau}(\mathbf{p}) = (p_1 L_1, p_2 L_2, p_3 L_3)$, $\mathbf{p} \in \mathbb{Z}^3$, is the periodic shift.

2.1 Boundary integral formulation

We are interested in systems of sedimenting particles, so we therefore consider a case where each particle is subject to a gravitational force $\mathbf{f}^{(\alpha)} = (\rho^{(\alpha)} - \rho_e) \Theta^{(\alpha)} \mathbf{g}$, where \mathbf{g} is the gravity vector, but no external torque. The flow in the domain D_e can then be represented by the completed double layer formulation [37, 16], which we sum over all periodic images of all bodies in the primary cell,

$$u_j(\mathbf{x}) = \sum_{\alpha=1}^{N_p} \left(\mathcal{W}_j[\Gamma_\alpha, \mathbf{q}](\mathbf{x}) + \mathcal{V}_j[\Gamma_\alpha, \mathbf{f}^{(\alpha)}](\mathbf{x}) \right), \quad (1)$$

where \mathcal{W} is the Stokes double layer potential with density \mathbf{q} ,

$$\mathcal{W}_j[\Gamma, \mathbf{q}](\mathbf{x}) = \sum_{\mathbf{p} \in \mathbb{Z}^3} \int_{\Gamma} q_l(\mathbf{y}) T_{jlm}(\mathbf{x}, \mathbf{y} + \boldsymbol{\tau}(\mathbf{p})) \hat{n}_m(\mathbf{y}) dS(\mathbf{y}), \quad (2)$$

and \mathcal{V} is the completion flow required to ensure that the representation is complete. The completion flow must be chosen such that the net torque and force on the body is correct, since the double layer potential is unable to represent either. Here we have chosen the completion flow equal to that from a point force \mathbf{f} acting at the body center of mass \mathbf{x}_c ,

$$\mathcal{V}_j[\Gamma, \mathbf{f}](\mathbf{x}) = \sum_{\mathbf{p} \in \mathbb{Z}^3} \frac{1}{8\pi\mu} G_{jm}(\mathbf{x}, \mathbf{x}_c + \boldsymbol{\tau}(\mathbf{p})) f_m. \quad (3)$$

There is a slight abuse of notation in (2) and (3) since the periodic sums over \mathbf{p} are only conditionally convergent, but we shall defer the discussion of periodic properties to section 3. The tensors T and G are the stresslet and stokeslet fundamental singularities,

$$T_{ijk}(\mathbf{x}, \mathbf{y}) = -6 \frac{r_i r_j r_k}{r^5},$$

$$G_{ij}(\mathbf{x}, \mathbf{y}) = \frac{\delta_{ij}}{r} + \frac{r_i r_j}{r^3},$$

where $\mathbf{r} = \mathbf{x} - \mathbf{y}$ and $r = |\mathbf{r}|$. They can equivalently be formulated as operators acting on r [39, 13],

$$G_{ij} = (\delta_{ij} \nabla^2 - \nabla_i \nabla_j) r,$$

$$T_{ijk} = \left[(\delta_{ij} \nabla_k + \delta_{jk} \nabla_i + \delta_{ki} \nabla_j) \nabla^2 - 2 \nabla_i \nabla_j \nabla_k \right] r =: \mathbf{K}_{ijk} r. \quad (4)$$

Note that the sums over all periodic images in (2) and (3) are only conditionally convergent; we will discuss their evaluation in sections 3 and 4.

On the surface Γ_α of each particle we set a no-slip boundary condition. For a rigid body with center of mass $\mathbf{x}_c^{(\alpha)}$ and translational and rotational velocities $\mathbf{V}^{(\alpha)}$ and $\boldsymbol{\Omega}^{(\alpha)}$, this implies that

$$\mathbf{u}(\mathbf{x}) = \mathbf{U}^{(\alpha)}(\mathbf{x}) := \mathbf{V}^{(\alpha)} + \boldsymbol{\Omega}^{(\alpha)} \times (\mathbf{x} - \mathbf{x}_c^{(\alpha)}), \quad \mathbf{x} \in \Gamma_\alpha. \quad (5)$$

The double layer potential contains a jump as $\mathbf{x} \rightarrow \Gamma$ from the external domain,

$$\lim_{\substack{\mathbf{x} \rightarrow \Gamma \\ \mathbf{x} \in D_e}} \mathcal{W}_j[\Gamma, \mathbf{q}](\mathbf{x}) = -4\pi q_j(\mathbf{x}) + \mathcal{W}_j[\Gamma, \mathbf{q}](\mathbf{x}),$$

such that (1) gives us a second-kind Fredholm integral equation for the unknown density \mathbf{q} when we let $\mathbf{x} \in D_e$ go to the surface of one of the bodies,

$$-4\pi q_j(\mathbf{x}) + \sum_{\alpha=1}^{N_p} \left(\mathcal{W}_j[\Gamma_\alpha, \mathbf{q}](\mathbf{x}) + \mathcal{V}_j[\Gamma_\alpha, \mathbf{f}^{(\alpha)}](\mathbf{x}) \right) = U_j^{(\beta)}(\mathbf{x}), \quad \mathbf{x} \in \Gamma_\beta, \quad \beta = 1, \dots, N_p. \quad (6)$$

The system is closed by adding the constitutive equations relating \mathbf{q} to $\mathbf{V}^{(\alpha)}$ and $\boldsymbol{\Omega}^{(\alpha)}$ [38],

$$\mathbf{V}^{(\alpha)} = -\frac{4\pi}{S_{\Gamma_\alpha}} \int_{\Gamma_\alpha} \mathbf{q}(\mathbf{y}) dS(\mathbf{y}), \quad (7)$$

$$\boldsymbol{\Omega}^{(\alpha)} = -4\pi \sum_{n=1}^3 \frac{\boldsymbol{\omega}^{(\alpha,n)}}{A_n^{(\alpha)}} \left(\boldsymbol{\omega}^{(\alpha,n)} \cdot \int_{\Gamma_\alpha} (\mathbf{y} - \mathbf{x}_c^{(\alpha)}) \times \mathbf{q}(\mathbf{y}) dS(\mathbf{y}) \right), \quad (8)$$

where S_{Γ_α} is the surface area of Γ_α and

$$A_n^{(\alpha)} = \int_{\Gamma_\alpha} \left[\boldsymbol{\omega}^{(\alpha,n)} \times (\mathbf{y} - \mathbf{x}_c^{(\alpha)}) \right] \cdot \left[\boldsymbol{\omega}^{(\alpha,n)} \times (\mathbf{y} - \mathbf{x}_c^{(\alpha)}) \right] dS(\mathbf{y}).$$

The vectors $\boldsymbol{\omega}^{(\alpha,n)}$, $n = 1, 2, 3$, are independent unit vectors, which must satisfy

$$\frac{1}{\sqrt{A_n^{(\alpha)} A_m^{(\alpha)}}} \int_{\Gamma_\alpha} \left[\boldsymbol{\omega}^{(\alpha,m)} \times (\mathbf{y} - \mathbf{x}_c^{(\alpha)}) \right] \cdot \left[\boldsymbol{\omega}^{(\alpha,n)} \times (\mathbf{y} - \mathbf{x}_c^{(\alpha)}) \right] dS(\mathbf{y}) = \delta_{mn}.$$

They can be quickly computed using the modified Gram-Schmidt (MGS) algorithm [10, p. 107].

With (6) we now have a complete boundary integral formulation, though there are two issues that require immediate attention before we can move on. The first issue is that we cannot compute the single and double layer potentials without first defining how the periodic sums in \mathbf{p} are computed, which we will discuss further in section 3. The second issue is that the integrand in the double layer potential (2) contains a singularity at $(\mathbf{y} = \mathbf{x}, \mathbf{p} = 0)$ if $\mathbf{x} \in \Gamma$. The Lyapunov smoothness of the surface ensures that the integral still exists as an improper integral, since $r_k \hat{n}_k$ then goes to zero as $\mathbf{y} \rightarrow \mathbf{x}$ (see Kim & Karrila [27, p. 24]), but the singularity must still be addressed before any numerical quadrature can be applied. We will discuss how this can be done by the method of singularity subtraction in section 2.3.

It is here worth noting that the viscosity μ of the liquid enters the formulation in one place only: as a scaling factor to the gravitational body force \mathbf{f} in the completion flow (3). Scaling the magnitude of the body force in Stokes flow will only affect the time scale of the flow, not the dynamics. For this reason we have used $\mu = 1$ for all the computations throughout this paper.

2.2 Useful identities

We here state a few integral identities which we shall find useful when working with boundary integrals for Stokes flow. The first is the stokeslet integral identity

$$\int_{\Gamma} G_{ij}(\mathbf{x}, \mathbf{y}) \hat{n}_j(\mathbf{y}) dS(\mathbf{y}) = 0, \quad (9)$$

independently of \mathbf{x} . The second identity is the stresslet integral identity

$$\int_{\Gamma} T_{jlm}(\mathbf{x}, \mathbf{y}) \hat{n}_m(\mathbf{y}) dS(\mathbf{y}) = \delta_{jl} \begin{cases} 0, & \mathbf{x} \text{ outside the domain enclosed by } \Gamma, \\ 4\pi, & \mathbf{x} \in \Gamma, \\ 8\pi, & \mathbf{x} \text{ inside the domain enclosed by } \Gamma. \end{cases} \quad (10)$$

It is worth noting that the sign of the right-hand side of (10) changes if the arguments \mathbf{x} and \mathbf{y} are swapped, as $T_{jlm}(\mathbf{y}, \mathbf{x}) = -T_{jlm}(\mathbf{x}, \mathbf{y})$. The third identity is that for a rigid body with a double layer density \mathbf{q} that satisfies (6), we have (see [38, ch.4])

$$\int_{\Gamma} q_i(\mathbf{y}) \hat{n}_i(\mathbf{y}) dS(\mathbf{y}) = 0. \quad (11)$$

This last identity can be derived from (6), (9) and (10) together with the assumption of rigid body motion.

2.3 Singularity subtraction

We address the singularity in the double layer potential (2) by the method of singularity subtraction, which for this case is based on the stresslet identity (10). We add and subtract $\mathbf{q}(\mathbf{x})$ to the integrand in (2) and apply (10), such that for $\mathbf{x} \in \Gamma$ we get a contribution from the case $\mathbf{p} = 0$,

$$\begin{aligned} \mathcal{W}_j[\Gamma, \mathbf{q}](\mathbf{x}) &= \sum_{\mathbf{p} \in \mathbb{Z}^3} \int_{\Gamma} (q_l(\mathbf{y}) - q_l(\mathbf{x}) + q_l(\mathbf{x})) T_{jlm}(\mathbf{x}, \mathbf{y} + \boldsymbol{\tau}(\mathbf{p})) \hat{n}_m(\mathbf{y}) dS(\mathbf{y}) \\ &= \sum_{\mathbf{p} \in \mathbb{Z}^3} \int_{\Gamma} (q_l(\mathbf{y}) - q_l(\mathbf{x})) T_{jlm}(\mathbf{x}, \mathbf{y} + \boldsymbol{\tau}(\mathbf{p})) \hat{n}_m(\mathbf{y}) dS(\mathbf{y}) + 4\pi q_j(\mathbf{x}) \\ &= \mathcal{W}_j[\Gamma, \mathbf{q} - \mathbf{q}(\mathbf{x})](\mathbf{x}) + 4\pi q_j(\mathbf{x}). \end{aligned} \quad (12)$$

This improves the regularity of the integrand, which is now bounded, hence reducing the error if omitting the point $\mathbf{y} = \mathbf{x}$ when applying a numerical quadrature. It does however only help us when evaluating for points $\mathbf{x} \in \Gamma$. For points \mathbf{x} which are very close to (but not on) Γ , the distance $r = |\mathbf{x} - \mathbf{y}|$ is very small (but never zero) and the double layer potential exhibits a nearly singular behavior that can cause numerical errors. This typically occurs when particles are very close to each other, and will be discussed further in section 9.2.

2.4 Discrete system

We here introduce the discrete form of our boundary integral formulation (6). Consider an approximation of the surface integrals using a quadrature rule with weights w_j (see section

9.1 for the specific quadrature used in this work), such that for a function $g(\mathbf{x})$

$$\sum_{\alpha=1}^{N_p} \int_{\Gamma_\alpha} g(\mathbf{x}) dS(\mathbf{x}) \approx \sum_{\alpha,j} w_j^{(\alpha)} g(\mathbf{x}_j^{(\alpha)}) = \sum_{s=1}^N w_s g(\mathbf{x}_s), \quad (13)$$

where the sum $s = 1, \dots, N$ goes over all discretization points on all bodies in the primary cell. We can then express the total contribution from the quadrature of the double layer potential as

$$\sum_{\alpha=1}^{N_p} \mathcal{W}_j^h[\Gamma_\alpha, \mathbf{q}](\mathbf{x}) := \sum_{s=1}^N \sum_{\mathbf{p} \in \mathbb{Z}^3} T_{jlm}(\mathbf{x} - \mathbf{x}_s + \tau(\mathbf{p})) q_l(\mathbf{x}_s) n_m(\mathbf{x}_s) \quad (14)$$

where $\mathbf{n}(\mathbf{x}_s) = w_s \hat{\mathbf{n}}(\mathbf{x}_s)$ is the discrete form of the vector surface element $d\mathbf{S}$, and \mathcal{W}^h is the discrete form of \mathcal{W} . Using the quadrature (14) together with singularity subtraction (12), we can formulate our discrete system for $\mathbf{q}(\mathbf{x}_t)$, $t = 1, \dots, N$, by applying the Nyström method,

$$\sum_{\alpha=1}^{N_p} \left(\mathcal{W}_j^h[\Gamma_\alpha, \mathbf{q} - \mathbf{q}(\mathbf{x}_t)](\mathbf{x}_t) + \mathcal{V}_j[\Gamma_\alpha, \mathbf{f}^{(\alpha)}](\mathbf{x}_t) \right) = U_j^h(\mathbf{x}_t), \quad t = 1, \dots, N, \quad (15)$$

where $\mathbf{U}^h(\mathbf{x}_t)$ is computed from \mathbf{q} using the same quadrature rule as \mathcal{W}^h . This system is non symmetric and well-conditioned, and therefore suitable for iterative solution using GMRES [40]. The computation of the double layer potential and surface velocities is then viewed as a matrix-vector product (\mathbf{q} being the vector), which is computed in every GMRES iteration.

3 Periodic properties of singularities

We have formulated our boundary integral equation in (6), and introduced singularity subtraction (12) to make it available for numerical solution (15). However, the sums over all periodic images (\mathbf{p}) are only conditionally convergent and not suitable for evaluation by direct summation. To compute the sums we will apply the technique of Ewald summation, which makes the sums converge rapidly. Before we go into that, however, we shall investigate the properties of the stokeslet and stresslet singularities in a periodic setting. Specifically, we shall see that the stresslet, unlike the stokeslet, produces a mean flow when summed periodically, unless balanced by a constant term.

3.1 Periodic stresslet

We begin by considering the flow $\varphi(\mathbf{x})$ from a single stresslet singularity with strength S_{lm} , located at the point \mathbf{y} in the primary cell,

$$\varphi_j(\mathbf{x}) = \sum_{\mathbf{p} \in \mathbb{Z}^3} T_{jlm}(\mathbf{x} - \mathbf{y} + \tau(\mathbf{p})) S_{lm}. \quad (16)$$

This sum is only conditionally convergent, so the summation order affects the result. In section 4 we will introduce Ewald summation, which accelerates convergence and makes the sum absolutely convergent by computing it in a spherical order. Here we will look at a Fourier representation of the sum to determine its mean value, which we later will need when deriving

the Ewald sum. Using Poisson's summation formula, we take the summation over periodic images to \mathbf{k} -space,

$$\varphi_j(\mathbf{x}) = \frac{S_{lm}}{V} \sum_{\mathbf{k} \neq 0} \widehat{T}_{jlm}(\mathbf{k}) e^{i\mathbf{k} \cdot \mathbf{r}} + \frac{S_{lm}}{V} \widehat{T}_{jlm}^{(0)}(\mathbf{y}), \quad (17)$$

where \widehat{T} is the Fourier transform of T , $k_i \in \{2\pi n/L_i : n \in \mathbb{Z}\}$ and $\mathbf{r} = \mathbf{x} - \mathbf{y}$. The second term is the $\mathbf{k} = 0$ component of the \mathbf{k} -space sum, which we shall discuss shortly.

Beginning with $\mathbf{k} \neq 0$, we first take the definition of the operator \widehat{K} from (4) and introduce

$$\widehat{K}_{jlm}(\mathbf{k}) = -i \left[(\delta_{jl}k_m + \delta_{lm}k_j + \delta_{mj}k_l)k^2 - 2k_jk_lk_m \right], \quad k = |\mathbf{k}|, \quad (18)$$

which represents the result of applying \widehat{K} to a Fourier series. Together with the result

$$\widehat{r}(\mathbf{k}) = -8\pi/k^4,$$

this allows us to write the periodic form of the stresslet as

$$\widehat{T}_{jlm}(\mathbf{k}) = \widehat{K}_{jlm}(\mathbf{k})\widehat{r}(\mathbf{k}) = i \frac{8\pi}{k^4} \left[(\delta_{jl}k_m + \delta_{lm}k_j + \delta_{mj}k_l)k^2 - 2k_jk_lk_m \right], \quad \mathbf{k} \neq 0.$$

Returning to $\mathbf{k} = 0$, the second term of (17) contains the tensor $\widehat{T}_{jlm}^{(0)}(\mathbf{y})$, which is not defined so far in this derivation. For our current application we shall derive an expression for this tensor based on two criteria: (i) we want the mean flow through the primary cell to be zero, and (ii) we want the identity (10) to be valid also in the periodic setting. If we omit the $\mathbf{k} = 0$ term, i.e. we set $\widehat{T}^{(0)} = 0$, then our resulting solutions will have non zero mean flow¹ and the stresslet identity (10) will not be satisfied. However, by satisfying the first criterion, the second criterion will also be satisfied, as we shall see.

3.2 Stresslet mean flow

In our application we are interested in a periodic setup of rigid particles sedimenting through a quiescent liquid, meaning that we want to fix our frame of reference such that we have zero net flux through each side of the primary cell. To quantify the flux from the periodic double layer potential, we begin with the flow $\boldsymbol{\varphi}$ from the single stresslet singularity (16). Without loss of generality, we now consider the plane surface D_3 , which is normal to the unit vector \mathbf{e}_3 . It has dimensions $L_1 \times L_2$ and surface area $A_3 = L_1L_2$. The flux F_3 through D_3 can by (17) be written

$$\begin{aligned} F_3 &= \int_{D_3} \varphi_3(\mathbf{x}) dS(\mathbf{x}) \\ &= \frac{8\pi}{V} S_{lm} \sum_{\mathbf{k} \neq 0} \frac{i}{k^4} \left[(\delta_{3l}k_m + \delta_{lm}k_3 + \delta_{m3}k_l)k^2 - 2k_3k_lk_m \right] \int_{D_3} e^{i\mathbf{k} \cdot \mathbf{r}} dS(\mathbf{x}) + A_3 \frac{S_{lm}}{V} \widehat{T}_{3lm}^{(0)}(\mathbf{y}). \end{aligned}$$

¹At a glance, one might think that $\widehat{T}^{(0)} = 0$ gives a zero mean flow, since the integral of (17) over the entire primary cell vanishes. In our case the flow equations are only valid in the domain outside the particles, so the integral cannot be taken over the entire cell.

Since D_3 covers exactly one periodic length in the x_1 and x_2 directions, it follows that the integral over D_3 can be non zero only if $k_1 = k_2 = 0$, and then

$$\int_{D_3} e^{i\mathbf{k}\cdot\mathbf{r}} dS(\mathbf{x}) = e^{ik_3 r_3} A_3$$

and $k = |k_3|$. This allows us to write

$$F_3 = -\frac{8\pi}{V} S_{mm} A_3 \sum_{k_3 \neq 0} \frac{\sin(k_3 r_3)}{k_3} + A_3 \frac{S_{lm}}{V} \widehat{T}_{3lm}^{(0)}(\mathbf{y}).$$

The sum over k_3 can be evaluated using a standard result from Fourier series [17, p. 46]²,

$$\begin{aligned} \sum_{k_3 \neq 0} \frac{\sin(k_3 r_3)}{k_3} &= \frac{L_3}{\pi} \sum_{m=1}^{\infty} \frac{\sin(2\pi m r_3 / L_3)}{m} = \frac{L_3}{2} - r_3, & 0 < r_3 < L_3, \\ &= \frac{L_3}{2} - \text{mod}(r_3, L_3), & r_3 \in \mathbb{R}. \end{aligned}$$

We are thus able to write the final expression for the flux through D_3 from an arbitrary periodic stresslet singularity,

$$F_3 = -\frac{8\pi}{V} S_{mm} A_3 \left(\frac{L_3}{2} - \text{mod}(r_3, L_3) \right) + A_3 \frac{S_{lm}}{V} \widehat{T}_{3lm}^{(0)}(\mathbf{y}).$$

The first term of this expression has the form of a sawtooth wave in r_3 , with the jump appearing when $r_3 = 0$, i.e. \mathbf{y} lies in the plane D_3 .

In our application we are not considering arbitrary stresslet singularities, but rather a double layer potential distributed on a surface, $S_{lm}(\mathbf{y}) = q_l(\mathbf{y}) \hat{n}_m(\mathbf{y})$. We denote the flow resulting from only the double layer potential $\mathbf{u}^{(T)}(\mathbf{x})$, and write

$$u_j^{(T)}(\mathbf{x}) = \frac{1}{V} \sum_{\alpha} \sum_{\mathbf{k} \neq 0} \int_{\Gamma_{\alpha}} q_l(\mathbf{y}) \hat{n}_m(\mathbf{y}) \widehat{T}_{jlm}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}} dS(\mathbf{y}) + u_j^{(T,0)},$$

where

$$u_j^{(T,0)} = \frac{1}{V} \sum_{\alpha=1}^{N_p} \int_{\Gamma_{\alpha}} q_l(\mathbf{y}) \hat{n}_m(\mathbf{y}) \widehat{T}_{jlm}^{(0)}(\mathbf{y}) dS(\mathbf{y}).$$

Now, for periodically replicated bodies we are interested in the mean flow generated by the double layer potential, defined as the flux through the side D_j of the primary cell divided by the side area,

$$\begin{aligned} \langle u_j^{(T)} \rangle &= \frac{1}{A_j} \int_{D_j} u_j^{(T)}(\mathbf{x}) dS(\mathbf{x}) \\ &= -\frac{8\pi}{V} \sum_{\alpha=1}^{N_p} \int_{\Gamma_{\alpha}} q_m(\mathbf{y}) \hat{n}_m(\mathbf{y}) \left(\frac{L_j}{2} - \text{mod}(x_j - y_j, L_j) \right) dS(\mathbf{y}) + u_j^{(T,0)}. \end{aligned}$$

² $\text{mod}(a, b)$ refers to the operation *a modulo b*, which returns the remainder of the division of a by b , s.t. $0 \leq \text{mod}(a, b) < b$.

Here j has a fixed value, and there is no implicit summation over j . We assume that no side of the primary cell cuts the surface Γ_α , as this is only a matter of shifting point of reference, and apply the identity (11), that for the double layer density on the surface of a rigid body it holds that

$$\int_{\Gamma_\alpha} q_m(\mathbf{y}) \hat{n}_m(\mathbf{y}) dS(\mathbf{y}) = 0.$$

This allows us to reduce our expression for the mean flow through the primary cell to

$$\langle u_j^{(T)} \rangle = -\frac{8\pi}{V} \sum_{\alpha=1}^{N_p} \int_{\Gamma_\alpha} q_m(\mathbf{y}) \hat{n}_m(\mathbf{y}) y_j dS(\mathbf{y}) + u_j^{(T,0)}.$$

We now see that we can obtain zero mean flow, $\langle u_j^{(T)} \rangle = 0$, by defining the $\mathbf{k} = 0$ component of the stresslet as

$$\hat{T}_{jlm}^{(0)}(\mathbf{y}) = 8\pi \delta_{lm} y_j, \quad (19)$$

such that

$$u_j^{(T,0)} = \frac{8\pi}{V} \sum_{\alpha=1}^{N_p} \int_{\Gamma_\alpha} q_m(\mathbf{y}) \hat{n}_m(\mathbf{y}) y_j dS(\mathbf{y}).$$

We also find, by numerical experiments, that this choice for $\hat{T}^{(0)}$ yields that the periodic stresslet satisfies the identity (10), such that both our requirements are satisfied.

3.3 Periodic stokeslet

For a derivation of the periodic properties of the stokeslet, we refer to Pozrikidis [39]. Here we just summarize that

$$\hat{G}_{jm}(\mathbf{k}) = \frac{8\pi}{k^4} (\delta_{jm} k^2 - k_j k_m) \quad \text{and} \quad \hat{G}_{jm}(\mathbf{k} = 0) = 0,$$

such that the flow due to periodic point force \mathbf{f} can be expressed as

$$\psi_j(\mathbf{x}) = \frac{8\pi}{V} f_m \sum_{\mathbf{k} \neq 0} \hat{G}_{jm}(\mathbf{k}) e^{i\mathbf{k} \cdot \mathbf{r}}.$$

The derivation by Pozrikidis shows that the $\mathbf{k} = 0$ term is zero for the periodic stokeslet, and that force balance in the domain is maintained by a constant pressure gradient. Hence, the mean flow is zero,

$$\langle \psi \rangle = 0.$$

4 Ewald summation

The periodic sums (2) and (3) over the stresslet and stokeslet singularities in our formulation are only conditionally convergent. To resolve this issue and compute them efficiently, we will

use the Ewald summation technique, invented by P.P. Ewald [12] in 1921 for the electrostatic potential. In Ewald summation, a slowly converging sum over infinite periodic images is decomposed into two sums; one over near neighbors which converges rapidly in real space, and one containing the far-field interactions which converges rapidly in \mathbf{k} -space.

To apply the technique, one requires an Ewald decomposition of the potential being summed. An Ewald decomposition for the stokeslet was first derived by Hasimoto [23], and later Pozrikidis [39] gave an alternative decomposition following a decomposition method introduced by Beenakker [3] for the Rotne-Prager tensor. The two decompositions are very similar, though the one by Pozrikidis appears to have slightly slower convergence, as reported by Lindbo and Tornberg [31]. For the stresslet an Ewald decomposition was derived by Fan et al. [13] using Beenakker's decomposition method. Recently, a second decomposition for the stresslet was derived by Marin [34, Paper IV], based on the Hasimoto decomposition of the stokeslet.

4.1 Stresslet decomposition

In the context of Ewald summation it is convenient to consider the discrete sum from the beginning, so we define $\mathbf{u}^{(T)}$ to be the contribution from the discrete double layer potential (14),

$$u_j^{(T)}(\mathbf{x}) := \sum_{\alpha=1}^{N_p} \mathcal{W}_j^h[\Gamma_\alpha, \mathbf{q}](\mathbf{x}) = \sum_{s=1}^N \sum_{\mathbf{p} \in \mathbb{Z}^3} T_{jlm}(\mathbf{x} - \mathbf{x}_s + \tau(\mathbf{p})) q_l(\mathbf{x}_s) n_m(\mathbf{x}_s),$$

and use this expression as the basis for our derivation. To obtain the Ewald decomposition by Fan et al., we now apply Beenakker's method of decomposing r into $r \operatorname{erfc}(\xi r) + r \operatorname{erf}(\xi r)$ in the operator formulation of the stresslet (4). This gives us the decomposition

$$T_{jlm}(\mathbf{r}) = T_{jlm}^{(R)}(\mathbf{r}) + T_{jlm}^{(F)}(\mathbf{r}),$$

where

$$\begin{aligned} T_{jlm}^{(R)}(\mathbf{r}) &= K_{jlm}(r \operatorname{erfc}(\xi r)), \\ T_{jlm}^{(F)}(\mathbf{r}) &= K_{jlm}(r \operatorname{erf}(\xi r)), \end{aligned}$$

and a corresponding decomposition $u_j^{(T)} = u_j^{(R)} + u_j^{(F)}$. This has the property that $T^{(R)}$ is short-range and converges rapidly in real space, while $T^{(F)}$ is smooth and converges rapidly when summed in \mathbf{k} -space. The parameter ξ controls the speed with which the two sums converge.

4.1.1 Real space sum

The real space component is obtained by differentiation,

$$\begin{aligned} T_{jlm}^{(R)}(\mathbf{r}) &= K_{jlm}(r \operatorname{erfc}(\xi r)) = C(\xi, \mathbf{r}) \hat{r}_j \hat{r}_l \hat{r}_m + D(\xi, \mathbf{r}) (\delta_{jl} \hat{r}_m + \delta_{lm} \hat{r}_j + \delta_{mj} \hat{r}_l) \\ &=: A_{jlm}(\xi, \mathbf{r}), \end{aligned} \tag{20}$$

where $\hat{\mathbf{r}} = \mathbf{r}/r$ and

$$\begin{aligned} C(\xi, \mathbf{r}) &= -\frac{2}{r} \left(\frac{3}{r} \operatorname{erfc}(\xi r) + \frac{2\xi}{\sqrt{\pi}} (3 + 2\xi^2 r^2 - 4\xi^4 r^4) e^{-\xi^2 r^2} \right), \\ D(\xi, \mathbf{r}) &= \frac{8\xi^3 r}{\sqrt{\pi}} (2 - \xi^2 r^2) e^{-\xi^2 r^2}. \end{aligned} \quad (21)$$

Using $S_{lm}(\mathbf{x}_s) = q_l(\mathbf{x}_s)n_m(\mathbf{x}_s)$ for simplicity, we can write the real space sum as

$$u_j^{(R)}(\mathbf{x}) = \sum_{s=1}^N \sum_{\mathbf{p} \in \mathbb{Z}^3} A_{jlm}(\xi, \mathbf{x} - \mathbf{x}_s + \tau(\mathbf{p})) S_{lm}(\mathbf{x}_s). \quad (22)$$

4.1.2 k-space sum

We write the contribution to \mathbf{u} from \mathbf{k} -space as

$$u_j^{(F)}(\mathbf{x}) = \sum_{s=1}^N \sum_{\mathbf{p} \in \mathbb{Z}^3} T_{jlm}^{(F)}(\mathbf{x} - \mathbf{x}_s + \tau(\mathbf{p})) S_{lm}(\mathbf{x}_s).$$

Again, using Poisson summation we take the sum over all periodic boxes to \mathbf{k} -space,

$$\sum_{\mathbf{p} \in \mathbb{Z}^3} T_{jlm}^{(F)}(\mathbf{x} - \mathbf{x}_s + \tau(\mathbf{p})) = \frac{1}{V} \sum_{\mathbf{k} \neq 0} \hat{T}_{jlm}^{(F)}(\mathbf{k}) e^{i\mathbf{k} \cdot (\mathbf{x} - \mathbf{x}_s)} + \frac{1}{V} \hat{T}_{jlm}^{(F,0)}(\mathbf{x}_s). \quad (23)$$

The transform of the \mathbf{k} -space part of the stresslet can be expressed as $\hat{T}_{jlm}^{(F)} = \hat{\mathbf{K}}_{jlm} \hat{\mathbf{Q}}$, where

$$\hat{\mathbf{K}}_{jlm} = -i \left[(\delta_{jl} k_m + \delta_{lm} k_j + \delta_{mj} k_l) k^2 - 2k_j k_l k_m \right]$$

is introduced in (18) and $\hat{\mathbf{Q}}$ is the transform of the decomposition factor, derived in plain terms by Pozrikidis [39] as

$$\hat{\mathbf{Q}} = \mathcal{F}[r \operatorname{erf}(\xi r)] = -\frac{8\pi}{k^4} \left(1 + \frac{1}{4} \frac{k^2}{\xi^2} + \frac{1}{8} \frac{k^4}{\xi^4} \right) e^{-k^2/4\xi^2}.$$

This gives us the transform of $T^{(F)}$,

$$\hat{T}_{jlm}^{(F)} = i \frac{\pi}{k} \left[(\delta_{jl} \hat{k}_m + \delta_{lm} \hat{k}_j + \delta_{mj} \hat{k}_l) - 2\hat{k}_j \hat{k}_l \hat{k}_m \right] \left(8 + 2 \frac{k^2}{\xi^2} + \frac{k^4}{\xi^4} \right) e^{-k^2/4\xi^2}.$$

For $\mathbf{k} = 0$, we note that the properties of mean flow discussed in section 3.2 are independent of the Ewald decomposition, and so $\hat{T}^{(F,0)} = \hat{T}^{(0)}$. For $\mathbf{k} \neq 0$, we are interested in the real part of the \mathbf{k} -space sum in (23), so we define

$$B_{jlm}(\xi, \mathbf{k}) := -\hat{T}_{jlm}^{(F)}(\xi, \mathbf{k}) e^{k^2/4\xi^2}$$

and write it as

$$\frac{1}{V} \sum_{\mathbf{k} \neq 0} \hat{T}_{jlm}^{(F)}(\mathbf{k}) e^{i\mathbf{k} \cdot (\mathbf{x} - \mathbf{x}_s)} = \frac{1}{V} \sum_{\mathbf{k} \neq 0} B_{jlm}(\xi, \mathbf{k}) e^{-k^2/4\xi^2} e^{-i\mathbf{k} \cdot (\mathbf{x} - \mathbf{x}_s)}.$$

We can now write the \mathbf{k} -space part of the Ewald sum as

$$\begin{aligned} u_j^{(F)}(\mathbf{x}) &= u_j^{(F,k)}(\mathbf{x}) + u_j^{(F,0)} \\ &:= \frac{1}{V} \sum_{\mathbf{k} \neq 0} B_{jlm}(\xi, \mathbf{k}) e^{-k^2/4\xi^2} \sum_{s=1}^N S_{lm}(\mathbf{x}_s) e^{-i\mathbf{k} \cdot (\mathbf{x} - \mathbf{x}_s)} + u_j^{(F,0)}, \end{aligned} \quad (24)$$

with

$$B_{jlm}(\xi, \mathbf{k}) = -i \frac{\pi}{k} \left[(\delta_{jl} \hat{k}_m + \delta_{lm} \hat{k}_j + \delta_{mj} \hat{k}_l) - 2 \hat{k}_j \hat{k}_l \hat{k}_m \right] \left(8 + 2 \frac{k^2}{\xi^2} + \frac{k^4}{\xi^4} \right)$$

and, using (19),

$$u_j^{(F,0)} = \frac{1}{V} \sum_{s=1}^N \hat{T}_{jlm}^{(0)}(\mathbf{x}_s) S_{lm}(\mathbf{x}_s).$$

4.1.3 Self interaction

If the target point \mathbf{x} in the original sum (14) is one of the evaluation points, $\mathbf{x} = \mathbf{x}_t$, $t \in \{1, \dots, N\}$, then the term corresponding to $\mathbf{p} = 0$ and $s = t$ is excluded, since the value is singular there (or zero when singularity subtraction is applied). This is in the context of Ewald summation referred to as removing the self interaction, and the same term is excluded from the real space sum (22). Part of this unwanted self interaction can however be incorporated into the \mathbf{k} -space sum (24) through the decomposition, and in that case a correction term must be added to remove this contribution. This is the case when doing Ewald summation of the stokeslet. In this decomposition however, taking the limit $r \rightarrow 0$ reveals that $\lim_{r \rightarrow 0} A_{jlm}(\xi, \mathbf{r}) - T_{jlm}(\mathbf{r}) = 0$. This means that no unwanted self-interaction has been included in the \mathbf{k} -space sum, and no correction has to be added.

4.1.4 Complete stresslet decomposition

The final Ewald sum for the stresslet becomes

$$\begin{aligned} u_j^{(T)}(\mathbf{x}) &= \sum_{s=1}^N \sum_{\mathbf{p} \in \mathbb{Z}^3} A_{jlm}(\xi, \mathbf{x} - \mathbf{x}_s + \tau(\mathbf{p})) S_{lm}(\mathbf{x}_s) \\ &+ \frac{1}{V} \sum_{\mathbf{k} \neq 0} B_{jlm}(\xi, \mathbf{k}) e^{-k^2/4\xi^2} \sum_{s=1}^N S_{lm}(\mathbf{x}_s) e^{-i\mathbf{k} \cdot (\mathbf{x} - \mathbf{x}_s)} \\ &+ \frac{1}{V} \sum_{s=1}^N \hat{T}_{jlm}^{(0)}(\mathbf{x}_s) S_{lm}(\mathbf{x}_s), \end{aligned} \quad (25)$$

where A , B and $\hat{T}^{(0)}$ are defined in equations (20), (24) and (19). The first sum converges rapidly in real space, and can therefore be truncated outside a cutoff radius r_c . Likewise, the second sum converges rapidly in \mathbf{k} -space, and can be truncated outside a maximum wave number K .

4.2 Stokeslet decomposition

For the periodic stokeslet sum (3), we use the Ewald decomposition by Hasimoto [23, 31],

$$\begin{aligned}
u_j^{(G)}(\mathbf{x}) &:= \sum_{\alpha=1}^{N_p} \mathcal{V}_j[\Gamma_\alpha, \mathbf{f}^{(\alpha)}](\mathbf{x}) \\
&= \frac{1}{8\pi\mu} \left(\sum_{\mathbf{p} \in \mathbb{Z}^3} \sum_{\alpha} A_{jm}^{(G)}(\xi, \mathbf{x} - \mathbf{x}_c^{(\alpha)} + \boldsymbol{\tau}(\mathbf{p})) f_m^{(\alpha)} \right. \\
&\quad \left. + \frac{1}{V} \sum_{\mathbf{k} \neq 0} B_{jm}^{(G)}(\xi, \mathbf{k}) e^{-k^2/4\xi^2} \sum_{\alpha=1}^{N_p} f_m^{(\alpha)} e^{-i\mathbf{k} \cdot (\mathbf{x} - \mathbf{x}_c^{(\alpha)})} \right)
\end{aligned} \tag{26}$$

where

$$A_{jm}^{(G)}(\xi, \mathbf{r}) = \left(\frac{2\xi e^{-\xi^2 r^2}}{\sqrt{\pi}} + \frac{\text{erfc}(\xi r)}{r} \right) (\delta_{jm} + \hat{r}_j \hat{r}_m) - \frac{4\xi}{\sqrt{\pi}} e^{-\xi^2 r^2} \delta_{jm},$$

and

$$B_{jm}^{(G)}(\xi, \mathbf{k}) = \frac{\pi}{k^2} \left(\delta_{jm} - \hat{k}_j \hat{k}_m \right) \left(8 + 2 \frac{k^2}{\xi^2} \right).$$

The self interaction term is given by

$$\mathbf{u}_{\text{self}}^{(G)}(\mathbf{x}_c^{(\alpha)}) = \lim_{\mathbf{x} \rightarrow \mathbf{x}_c^{(\alpha)}} \left(A^{(G)}(\xi, \mathbf{x} - \mathbf{x}_c^{(\alpha)}) - G(\mathbf{x} - \mathbf{x}_c^{(\alpha)}) \right) \mathbf{f}^{(\alpha)} = -\frac{4\xi}{\sqrt{\pi}} \mathbf{f}^{(\alpha)}.$$

Here we will never evaluate the stokeslet at the points $\mathbf{x}_c^{(\alpha)}$, so we need not consider the self interaction term.

5 Truncation error estimates for Ewald sum

When we compute the Ewald sums for the stresslet (25) and stokeslet (26), we need to truncate them at some point to avoid adding an infinite number of terms. To make our computations efficient, we would furthermore like to truncate the sums where the truncation error falls below a certain tolerance. In the case of Ewald summation of the electrostatic potential, Kolafa & Perram [28] developed an RMS error estimate for the case of charges randomly distributed throughout the primary cell. Their estimate, though only a statistical estimate, is very efficient and highly used in the field of molecular dynamics. For the Hasimoto decomposition of the stokeslet potential, Lindbo & Tornberg [32] developed a similar estimate, by following the same procedure as Kolafa & Perram. Here we develop the same kind of estimate for the stresslet decomposition detailed in section 4.1.

5.1 Real space sum

We want to estimate the pointwise error committed in the real space sum (22) when it is truncated with a cutoff radius r_c , such that the sum only contains interactions between

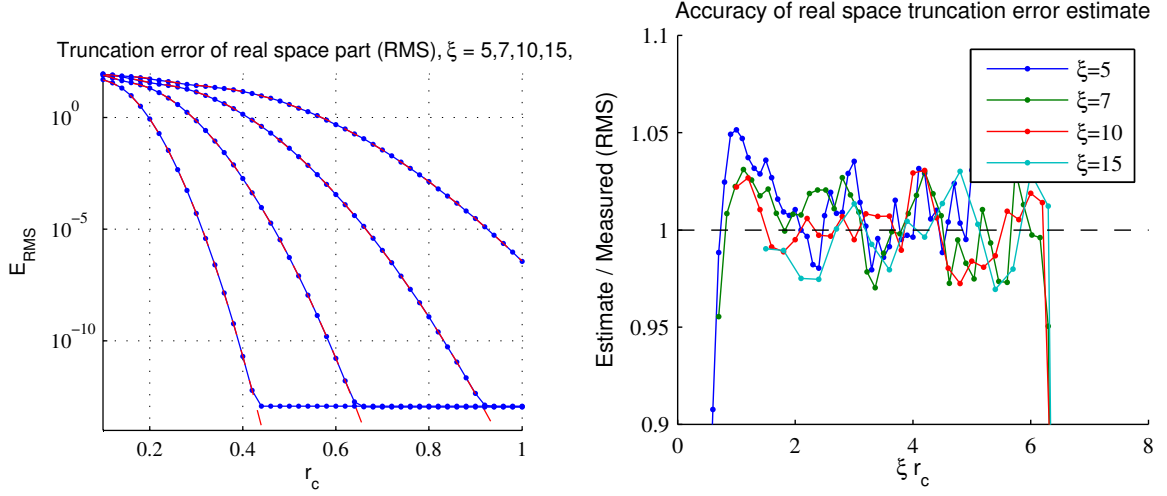


Figure 1: Left: Real space truncation error (RMS) vs radius r_c for 5000 random points in a 2^3 box. Blue dotted line measured against converged reference solution, red dashed is estimate (32). Right: Ratio between estimated and measured errors vs ξr_c , stays within a few percent until error hits machine precision.

points within a distance r_c from each other. Here we will estimate and measure the error in RMS (root mean squares),

$$e_{rms,j}^{(R)} := \sqrt{\frac{1}{N} \sum_{s=1}^N \left(u_j^{(R)}(\mathbf{x}_s) - \tilde{u}_j^{(R)}(\mathbf{x}_s) \right)^2}, \quad (27)$$

where $\mathbf{u}^{(R)}$ and $\tilde{\mathbf{u}}^{(R)}$ are the truncated and exact solutions. Our goal is to produce a predictive RMS error estimate like the one by Kolafa & Perram [28] for the original Ewald sum.

The truncation error in the real space sum at a point \mathbf{x}_t can be expressed as

$$e_j^{(R)}(\mathbf{x}_t) = \sum_{s \in \text{FL}_t} A_{jlm}(\xi, \mathbf{x} - \mathbf{x}_s + \tau(\mathbf{p})) q_l(\mathbf{x}_s) n_m(\mathbf{x}_s),$$

where the far list $\text{FL}_t = \{(\mathbf{x}_s, \mathbf{p}) : |\mathbf{x}_t - \mathbf{x}_s + \tau(\mathbf{p})| > r_c\}$ is the set of set of truncated points. Following the steps taken by Kolafa & Perram, we make the assumption that all the points are uncorrelated, allowing us to write

$$(e_{rms,j}^{(R)})^2 \approx \frac{1}{V} \sum_{s=1}^N q_l^2(\mathbf{x}_s) n_m^2(\mathbf{x}_s) \int_{|\mathbf{r}| > r_c} A_{jlm}^2(\xi, \mathbf{r}) dV.$$

For each component (fixed j) this expression is a sum of 9 components as the implicit summations over l and m are carried out. The RMS measure should be independent of how the coordinate system is oriented, so instead of evaluating all 9 components of the integral we take an average of the integrand first. We thus define

$$\overline{A_j^2}(\xi, \mathbf{r}) := \frac{1}{9} \sum_{l=1}^3 \sum_{m=1}^3 A_{jlm}^2(\xi, \mathbf{r}) \quad (28)$$

and

$$Q := \sum_{s=1}^N \sum_{l=1}^3 \sum_{m=1}^3 q_l^2(\mathbf{x}_s) n_m^2(\mathbf{x}_s) \quad (29)$$

such that we can write

$$(e_{rms,j}^{(R)})^2 \approx \frac{Q}{V} \int_{|\mathbf{r}|>r_c} \overline{A_j^2}(\xi, \mathbf{r}) dV. \quad (30)$$

Introducing spherical coordinates, (28) sums to

$$\overline{A_j^2}(\xi, \mathbf{r}) = \frac{1}{36} (C^2 + 6CD + 17D^2 - (C + 3D)^2 (\cos(2\theta) - 2 \cos(2\varphi) \sin^2(\theta))).$$

where $C = C(\xi, r)$ and $D = D(\xi, r)$, as defined in (21). Integrating this over one shell (fixed r), we get

$$\int_0^{2\pi} \int_0^\pi \overline{A_j^2}(\xi, \mathbf{r}) \sin(\theta) d\theta d\varphi = \frac{4\pi}{27} (C^2 + 6CD + 15D^2).$$

We now evaluate the integral over r ,

$$\begin{aligned} \frac{4\pi}{27} \int_{r_c}^\infty (C^2 + 6CD + 15D^2) r^2 dr &= \frac{1}{108} \left(768\sqrt{\pi}\xi e^{-\xi^2 r_c^2} (\xi^2 r_c^2 - 4) \operatorname{erfc}(\xi r_c) \right. \\ &\quad + 2247\sqrt{2\pi}\xi \operatorname{erfc}(\sqrt{2}\xi r_c) + \frac{576\pi \operatorname{erfc}(\xi r_c)^2}{r_c} \\ &\quad \left. + 4\xi^2 r_c e^{-2\xi^2 r_c^2} (448\xi^6 r_c^6 - 1392\xi^4 r_c^4 + 1588\xi^2 r_c^2 + 135) \right). \end{aligned} \quad (31)$$

Combining (28), (30) and (31) gives us an estimate for the RMS error (27), albeit a rather complex one. To obtain a more manageable expression, we series expand (31) for large ξr_c , finally arriving at

$$e_{rms,j}^{(R)} \approx e^{-\xi^2 r_c^2} \sqrt{\frac{1}{27} \frac{Q}{V} \xi^2 r_c (327 + 1588\xi^2 r_c^2 - 1392\xi^4 r_c^4 + 448\xi^6 r_c^6)}. \quad (32)$$

To evaluate the validity of (32), we apply it on a test case with $N = 5000$ points randomly distributed in a 2^3 box and \mathbf{q} and \mathbf{n} randomly drawn from a uniform distribution, $\mathbf{q}, \mathbf{n} \in \mathcal{U}(-1, 1)$. The results, shown in Figure 1, indicate that the estimate is accurate to within a few percent. It is tempting to simplify (32) even further by keeping only the highest order term, but doing so throws the accuracy of the estimate off by as much as a factor 5.

5.2 k-space sum

Just as the real space sum (22) is truncated beyond a cutoff radius r_c , the \mathbf{k} -space sum (24) needs to be truncated at some point. Since \mathbf{k} -space is discrete, it is natural to truncate the

summation outside a box with half-sides K in reciprocal space, such that the error committed at a point is

$$e_j^{(F)}(\mathbf{x}_t) = \frac{1}{V} \sum_{|k_i| > K} B_{jlm}(\xi, \mathbf{k}) e^{-k^2/4\xi^2} \sum_{s=1}^N q_l(\mathbf{x}_s) n_m(\mathbf{x}_s) e^{-i\mathbf{k} \cdot (\mathbf{x} - \mathbf{x}_s)}.$$

Similarly to the real space case, we are interested in obtaining an RMS estimate for the pointwise \mathbf{k} -space truncation error as a function of our numerical parameters, which in this case are K and ξ . Deriving such an estimate by following Kolafa & Perram is however much more difficult than it was in real space, so we here content ourselves with a heuristic estimate.

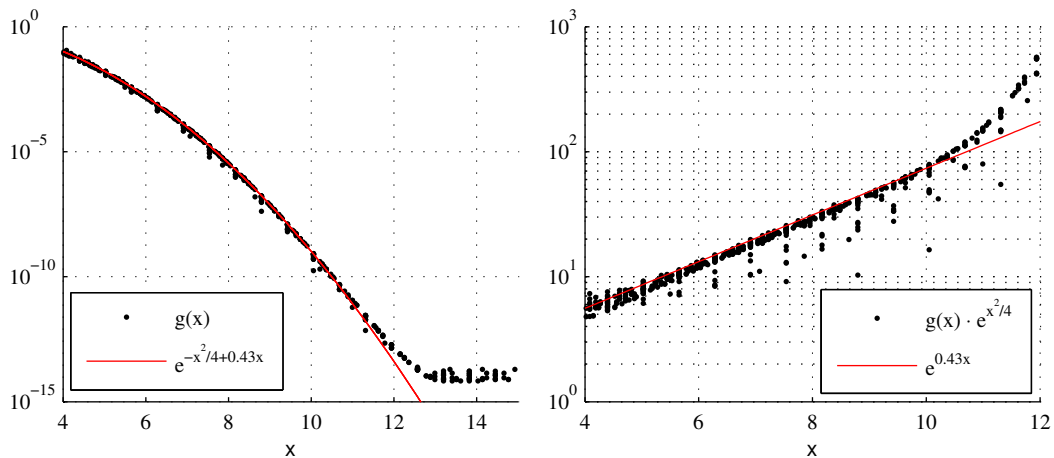


Figure 2: Left: Computed RMS errors versus $x = K/\xi$ after scaling (dots) and final estimate (solid line). Right: Computed RMS errors after removing exponential decay (dots) and exponential that matches the remaining behaviour (solid line).

First, we compute the RMS \mathbf{k} -space truncation errors (compared to well converged reference solutions) for a variety of N , $\{L_i\}$, ξ and K , with the points randomly distributed in the boxes (both cubic and non-cubic) and $\mathbf{q}, \mathbf{n} \in \mathcal{U}(-1, 1)$. Second, after some experimentation we observe that the errors collapse nicely as they decay if we scale them as $g(x) = e_{rms}^{(F)} / \sqrt{\xi^2 L Q V^{-1}}$, where $x = K/\xi$ and $L := \min_i L_i$, see Figure 2 (left). Observing that the decay should be dominated by the factor $e^{-K^2/4\xi^2} = e^{-x^2}$ for large K/ξ , we plot $g(x)e^{x^2/4}$ in a semi-log graph (Figure 2, right). This reveals that all points in the area of exponential decay ($x \gtrsim 4$) seem to fall on or below the straight line which is drawn by e^{cx} , $c \approx 0.43$. This line appears to provide a good limiting value in the region, before round-off errors start to accumulate at $x \approx 10$. With that we have finally reached our heuristic error estimate,

$$e_{rms}^{(F)} \approx e^{-K^2/4\xi^2 + 0.43K/\xi} \sqrt{\xi^2 L \frac{Q}{V}}. \quad (33)$$

This estimate shows good accuracy in all our test cases for $K/\xi \geq 4$, usually within 10–20%, and almost always overestimates the error. Figure 3 shows a test case with 7000 random points in a 2^3 box, where the error is well estimated to within 20% except for the case $\xi = 5$, in which the error is overestimated by a factor 2. This estimate is judged accurate enough to

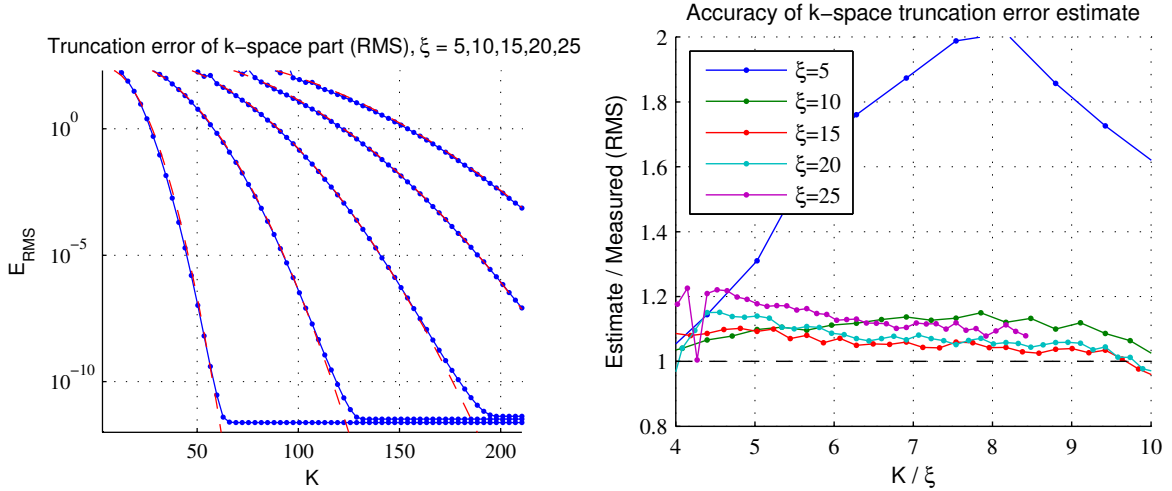


Figure 3: Left: \mathbf{k} -space truncation error (RMS) vs K for 7000 random points in a 2^3 box, ξ increasing from left to right. Blue dotted line measured against converged reference solution, red dashed is estimate (33). Right: Ratio between estimated and measured errors, estimate is consistently conservative and within 20% for $\xi \geq 20$.

be of practical use in applications where runtime parameters need to be determined based on an error tolerance.

5.3 Estimates for quadrature points on surfaces

The truncation error estimates for the \mathbf{k} -space and real space sums that we have developed give, as we have seen, very satisfactory results for randomly distributed data. However, for points derived from quadrature over surfaces (see section 9 for details on quadrature), we find that our definition (29) gives $Q \sim 1/N$ when we consider the case where a given geometry is discretized with an increasing number of points as N increases. At the same time we observe that in practice the truncation errors appear to be independent of N . Using the quadrature definition from section 2.4, where the pointwise data is $\mathbf{q}(\mathbf{x}_s)$ and $\mathbf{n}(\mathbf{x}_s) = w_s \hat{\mathbf{n}}(\mathbf{x}_s)$, we thus redefine Q for the quadrature case as

$$Q := \sum_{l=1}^3 \left(\sum_{s=1}^N w_s q_l^2(\mathbf{x}_s) \right) \sum_{m=1}^3 \left(\sum_{s=1}^N w_s \hat{n}_m^2(\mathbf{x}_s) \right) \approx \sum_{l=1}^3 \|q_l\|_2^2 \sum_{\alpha=1}^{N_p} S_{\Gamma_\alpha}, \quad (34)$$

where we have used the definition (13) and the identity $\sum_m \hat{n}_m^2 = 1$. We have also added the relationship to the 2-norm of the components of \mathbf{q} , with the 2-norm defined as

$$\|f\|_2^2 := \sum_{\alpha=1}^{N_p} \int_{\Gamma_\alpha} f^2(\mathbf{y}) dS(\mathbf{y}).$$

The interpretation of this is that the truncation errors related to the Ewald summation are independent of the discretization used, but rather depend only on the geometry and the distribution of \mathbf{q} on the surfaces.

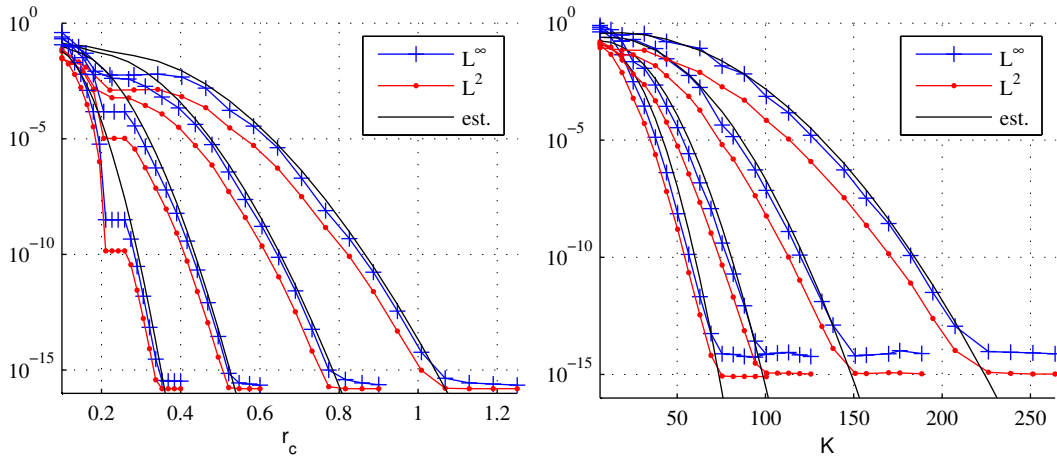


Figure 4: Real- and \mathbf{k} -space errors, measured in 2-norm (\cdot) and max norm ($+$), compared to error estimates ($-$) on an actual particle setup for $\xi = 6, 8, 12, 18$. The jump at $r_c = 0.2$ in the real space sum is due to the maximum particle diameter being $2R = 0.2$. These plots are for $N = 4000$, but the errors follow the estimates equally well for $N = 2000$ and $N = 1000$.

With this alternative definition of Q we still use the estimates (32) and (33). To illustrate that this works in practice, we set up a system of 4000 quadrature points distributed on 5 particles randomly distributed in a 1^3 box, with maximum radius $R = 0.1$ and aspect ratios between 3:4 and 4:3. We apply a unit downward force to the particles and solve for \mathbf{q} . We then compute the Ewald truncation errors (compared to a fully converged reference result) when evaluating the double layer potential $\mathcal{W}_j[\Gamma_\alpha, \mathbf{q}]$ at the quadrature points for a range of ξ, r_c, K . The results, shown in Figure 4, indicate that the estimates (32) and (33) together with the new definition of Q (34) actually seem to give a rather close upper bound on the error in max norm, while they overestimate the error in 2-norm by roughly one order of magnitude. This indicates that our estimates, though derived as approximations of a statistical measure, are of large practical importance.

6 Fast Ewald summation

Now we have the Ewald sums (25) and (26) for the stresslet and stokeslet, together with useful truncation error estimates related to the cutoffs r_c and K and the Ewald parameter ξ . The sums converge rapidly in terms of r_c and K , but not in terms of computational time; the stresslet sum (25) has $\mathcal{O}(N^2)$ complexity and the stokeslet sum (26) has $\mathcal{O}(N_p N)$ complexity, both with constants that grow rapidly as we increase the cutoffs. Evidently, a fast method of computing these sums is required.

6.1 Fast k-space summation: The spectral Ewald method

The spectral Ewald (SE) method was developed for periodic stokeslet potentials by Lindbo & Tornberg [31], and we compute the stokeslet sum (26) as described by them. We will here detail how the method can be applied to the stresslet sum (25), essentially by following their derivation step by step and introducing the higher-rank tensors of the stresslet. We begin by

introducing an arbitrary parameter η to split the Gaussian in the \mathbf{k} -space sum (24),

$$u_j^{(F,k)}(\mathbf{x}) = \frac{1}{V} \sum_{\mathbf{k} \neq 0} B_{jlm}(\xi, \mathbf{k}) e^{-(1-\eta)k^2/4\xi^2} \sum_{s=1}^N S_{lm}(\mathbf{x}_s) e^{-\eta k^2/4\xi^2} e^{-i\mathbf{k} \cdot (\mathbf{x} - \mathbf{x}_s)}.$$

We rewrite this by defining

$$\widehat{H}_{lm}(\mathbf{k}) := \sum_{s=1}^N S_{lm}(\mathbf{x}_s) e^{-\eta k^2/8\xi^2} e^{-i\mathbf{k} \cdot \mathbf{x}_s},$$

such that

$$u_j^{(F,k)}(\mathbf{x}) = \frac{1}{V} \sum_{\mathbf{k} \neq 0} B_{jlm}(\xi, \mathbf{k}) e^{-k^2/4\xi^2} e^{-i\mathbf{k} \cdot \mathbf{x}} e^{-\eta k^2/8\xi^2} \widehat{H}_{lm}(-\mathbf{k}).$$

The \mathbf{k} -space function \widehat{H}_{lm} is the result of a convolution in real space, with $H_{lm}(\mathbf{x})$ available as

$$H_{lm}(\mathbf{x}) = \sum_{s=1}^N S_{lm} \left(\frac{2\xi^2}{\pi\eta} \right)^{3/2} e^{-2\xi^2 |\mathbf{x} - \mathbf{x}_s|_*^2 / \eta}, \quad (35)$$

where $|\cdot|_*$ is the shortest distance in the periodic lattice. The function $H(\mathbf{x})$ is now a smooth function in the domain, created by a superposition of N Gaussians centered at the points \mathbf{x}_s , $s = 1, \dots, N$.

We now define

$$\widehat{\widetilde{H}}_j(\mathbf{k}) := B_{jlm}(\xi, -\mathbf{k}) e^{-(1-\eta)k^2/4\xi^2} \widehat{H}_{lm}(\mathbf{k}), \quad (36)$$

which is a tensor product in \mathbf{k} -space, such that

$$u_j^{(F,k)}(\mathbf{x}) = \frac{1}{V} \sum_{\mathbf{k} \neq 0} \widehat{\widetilde{H}}_j(-\mathbf{k}) e^{-\eta k^2/8\xi^2} e^{-i\mathbf{k} \cdot \mathbf{x}}.$$

We have for periodic functions that

$$\sum_{\mathbf{k}} \widehat{f}(-\mathbf{k}) \widehat{g}(\mathbf{k}) = \int_{\Omega} f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x},$$

and we identify $\widehat{f} = \widehat{\widetilde{H}}_{lm}$ and $\widehat{g} = e^{-\eta k^2/8\xi^2} e^{-i\mathbf{k} \cdot \mathbf{x}}$. Just as above, \widehat{g} is the result of a convolution, so we finally get

$$u_j^{(F,k)}(\mathbf{x}) = \int_{\Omega} \widetilde{H}_j(\mathbf{y}) \left(\frac{2\xi^2}{\pi\eta} \right)^{3/2} e^{-2\xi^2 |\mathbf{x} - \mathbf{y}|_*^2 / \eta} d\mathbf{y}. \quad (37)$$

The integrand of (37) is smooth and periodic, so a natural choice of quadrature is the trapezoidal rule, which has spectral accuracy in such a case [29].

The steps for calculating the \mathbf{k} -space sum can be summarized as

- I. **Spreading:** Discretize the primary cell using a uniform grid \mathbf{M} (defined below) and evaluate $H_{lm}(\mathbf{x})$ on it using (35), essentially spreading Gaussians on a grid.

II. **FFT**: Compute \widehat{H}_{lm} using the three-dimensional FFT.

III. **Scaling**: Compute \widetilde{H}_j via (36).

IV. **IFFT**: Compute \widetilde{H}_j using the inverse FFT.

V. **Quadrature**: Evaluate (37) with trapezoidal rule to get $u_j^{(F,k)}$.

The steps outlined here need to be performed for every element of H_{lm} and \widetilde{H}_j , so we need nine identical grids for steps I–II, and three grids for steps IV–V, with step III providing the mapping in between. The grid \mathbf{M} is uniform with $M_i = ML_i$ points in each direction and grid spacing $h = 1/M$. This implies that the number of modes represented in each direction is $M_i/2$ (assuming M_i even), and that the maximum wave number represented (as discussed in section 5.2) is $K = \frac{2\pi}{L_i} \frac{M_i}{2} = \pi M$ for $i = 1, 2, 3$.

Steps I and V involve evaluating $\mathcal{O}(N)$ Gaussians on a regular grid with $\mathcal{O}(M^3)$ points. In a naive implementation this would amount to $\mathcal{O}(NM^3)$ exponentials, which is very expensive. It is however natural to truncate the support of the Gaussians to a cube of P^3 points around the center point. This support P is tightly related to the free parameter η , which controls the width of the Gaussians. As detailed by Lindbo & Tornberg [33], writing

$$\eta = \left(\frac{2w\xi}{m} \right)^2,$$

lets us view the Gaussians as having half-width w and a shape parameter m controlling their decay rate. Furthermore, the approximation errors of the method arise from the spreading and quadrature steps, and for $m \sim \sqrt{P}$ these errors balance, such that spectral accuracy in P is obtained, meaning that the approximation error is bounded by $Ce^{-\alpha P}$ (for some C, α). We have in this paper used $m(P) = 0.9\sqrt{\pi P}$ throughout, with good results. Our observations (illustrated in Figure 5) are that $P = 16$ gives a relative precision around 10^{-10} , while $P = 24$ is enough for full precision³, though $P = 32$ reaches full precision faster when increasing the grid size. The latter is because the approximation errors in that case are completely decoupled from the spectrum truncation errors.

After truncation of the Gaussians, the complexity of evaluating them is $\mathcal{O}(NP^3)$, though the cost is still high since the exponential function has to be evaluated P^3 times per point. This cost can be dramatically reduced by applying the fast Gaussian gridding (FGG) method by Greengard & Lee [18]. The regularity of the grid is then exploited in a way that minimizes the number of exponential function evaluations. We refer to [33] for a complete discussion of how FGG is implemented in the spectral Ewald method.

Finally, the complexity of the \mathbf{k} -space summation using the spectral Ewald method can be written

$$\underbrace{\mathcal{O}(NP^3)}_{\text{Steps I and V.}} + \underbrace{\mathcal{O}(M^3 \log M^3)}_{\text{Steps II and IV.}} + \underbrace{\mathcal{O}(M^3)}_{\text{Step III.}}. \quad (38)$$

³The implementation we use is optimized for P being multiples of 8, which is why we compare these specific supports.

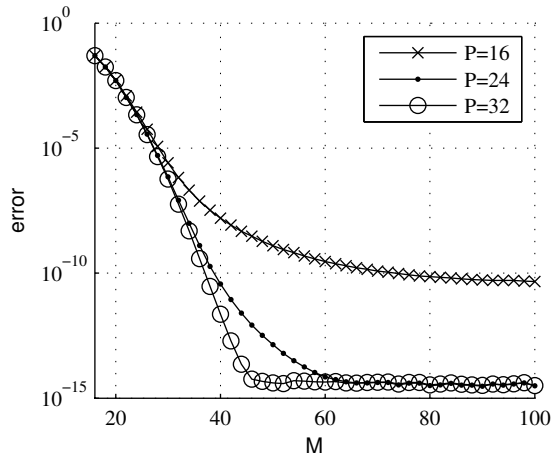


Figure 5: Relative \mathbf{k} -space truncation error (RMS) vs grid size for Gaussian support $P = 16, 24, 32$, $\xi = 11$, $L = 1^3$. The system is 5000 random points in a unit cube.

6.2 Fast real space summation

When we use GMRES for solving our discrete boundary integral formulation (15), the real space sum (22) will be computed in every iteration using the same geometry $(\mathbf{x}_s, \mathbf{n}(\mathbf{x}_s))$ and different $\mathbf{q}(\mathbf{x}_s)$. It is then useful to view it as a matrix-vector product, whose coefficients are calculated once for every geometry and stored for subsequent iterations,

$$\mathcal{A}_{jl}(\mathbf{x}_t, \mathbf{x}_s) = \sum_{\mathbf{p} \in \mathbb{Z}^3} A_{jlm}(\xi, \mathbf{x}_t - \mathbf{x}_s + \tau(\mathbf{p})) n_m(\mathbf{x}_s),$$

such that

$$u_j^{(R)}(\mathbf{x}_t) = \sum_{s=1}^N \mathcal{A}_{jl}(\mathbf{x}_t, \mathbf{x}_s) q_l(\mathbf{x}_s), \quad t = 1, \dots, N. \quad (39)$$

Evaluating (39) clearly has $\mathcal{O}(N^2)$ computational complexity. However, for a given tolerance ϵ and Ewald parameter ξ there will be a truncation radius r_c , such that the interactions are only calculated for the near neighbors NL_t of each point, defined as $\text{NL}_t = \{(\mathbf{x}_s, \mathbf{p}) : |\mathbf{x}_t - \mathbf{x}_s + \tau(\mathbf{p})| \leq r_c\}$. If the number of near neighbors of each point, $|\text{NL}_t|$, is kept constant as the system grows, then the complexity of computing the interactions is $\mathcal{O}(N)$. Computing the neighbor lists NL_t can also be accomplished at an $\mathcal{O}(N)$ cost by first creating a cell list, as is standard in molecular dynamics (see [14, Appendix F] for details). The total cost for evaluating the real space sum is therefore $\mathcal{O}(N)$, under the assumptions of constant number of near neighbors. The implication of this scaling on the overall complexity will be discussed in section 8.2.

The coefficients matrices \mathcal{A}_{jl} will be sparse for sufficiently small r_c , and can therefore be efficiently stored in memory. The sparsity structure will be identical for all matrices, so memory usage can be reduced by storing it only once. Additional memory can be saved by noting that $\mathcal{A}_{jl} = \mathcal{A}_{lj}$, so only 6 out of 9 entries actually need to be stored.

7 Complete numerical formulation

With the framework for computing fast Ewald sums in place, we are now able to fully formulate our numerical method. We reorder our discrete boundary integral equation (15) to make the RHS independent of \mathbf{q} , such that our final discrete equation to be solved becomes

$$-\sum_{\alpha=1}^{N_p} \mathcal{W}_j^h[\Gamma_\alpha, \mathbf{q} - \mathbf{q}(\mathbf{x}_t)](\mathbf{x}_t) + U_j(\mathbf{x}_t) = \sum_{\alpha=1}^{N_p} \mathcal{V}_j[\Gamma_\alpha, \mathbf{f}^{(\alpha)}](\mathbf{x}_t), \quad t = 1, \dots, N. \quad (40)$$

The surface velocity $\mathbf{U}(\mathbf{x}_t)$ is computed as in (5), using (7), (8) and the definition

$$\mathbf{U}(\mathbf{x}_t) := \mathbf{U}^{(\alpha)}(\mathbf{x}_t), \quad \mathbf{x}_t \in \Gamma_\alpha,$$

such that the entire LHS depends on $\mathbf{q}(\mathbf{x}_t)$ only. We solve this $3N \times 3N$ system for $\mathbf{q}(\mathbf{x}_t)$ using the implementation of GMRES available in MATLAB [35], letting the LHS be the matrix-vector product computed in each iteration. The Ewald sums are computed as described in section 6, using an optimized C implementation called through MATLAB's MEX interface, with the \mathbf{k} -space sum computed once every iteration and the sparse real space interaction matrix computed and stored in the first iteration and applied in subsequent iterations. The fast Fourier transforms are computed using the FFTW library [15], also available in MATLAB.

7.1 Singularity subtraction

The singularity subtraction (12) is efficiently implemented by noting that all of the singular behaviour is contained in the real space part, such that we can write

$$\begin{aligned} W_j^h[\Gamma_\alpha, \mathbf{q} - \mathbf{q}(\mathbf{x}_t)] &= \sum_{s=1}^N \sum_{\mathbf{p} \in \mathbb{Z}^3} T_{jlm}^{(R)}(\mathbf{r}) (q_l(\mathbf{x}_s) - q_l(\mathbf{x}_t)) n_m(\mathbf{x}_s) \\ &+ \sum_{s=1}^N \sum_{\mathbf{p} \in \mathbb{Z}^3} T_{jlm}^{(F)}(\mathbf{r}) q_l(\mathbf{x}_s) n_m(\mathbf{x}_s) - q_l(\mathbf{x}_t) \sum_{s=1}^N \sum_{\mathbf{p} \in \mathbb{Z}^3} T_{jlm}^{(F)}(\mathbf{r}) n_m(\mathbf{x}_s). \end{aligned} \quad (41)$$

The computational cost from adding the singularity subtraction to the kernel of the real space sum is negligible, so the only extra cost is from the evaluation of the last term in (41). This leads to one extra \mathbf{k} -space Ewald summation, but since the sum is only dependent on the geometry ($\mathbf{x}_s, \mathbf{n}(\mathbf{x}_s)$) it is enough to compute it in the first GMRES iteration and reuse it in subsequent iterations.

8 Parameter choice and complexity

We have developed truncation error estimates and complexity statements for the Ewald summation and detailed how the sums are computed, so we are now able to discuss the issue of complexity and parameter choice when solving our discrete system (40).

8.1 Parameter choice

When solving the system with GMRES to a tolerance ϵ around 10^{-10} , convergence is typically obtained in 10-50 iterations for the geometries studied in this work. The condition number

of the system is mesh independent, and the number of iterations required stays constant as the discretization is refined. The errors in the fast Ewald summation are composed of the approximation error and the truncation errors. The approximation error depends on the Gaussian support P , as discussed in section 6.1. For the truncation errors, we have the estimates (32) and (33) with Q as defined in (34). Using these estimates, the cutoff radius r_c and grid size \mathbf{M} for the Ewald summation can be determined for a given ξ and error tolerance ϵ . The work balance between real space and \mathbf{k} -space is determined by the decomposition parameter ξ , which can be selected to minimize the total runtime. On a given architecture, the wall clock times t_{rs} and t_{se} for the real space and \mathbf{k} -space sums can be tabulated for a parameter space, and later used for approximating the runtime required for a problem. The cost of the real space sum with average number of near neighbors $\langle |\text{NL}| \rangle$ is $\mathcal{O}(\langle |\text{NL}| \rangle N)$, and can be tabulated for different $\langle |\text{NL}| \rangle$. The cost of the \mathbf{k} -space sum using the spectral Ewald method is given in (38), where the cost of each term can be tabulated separately. If the matrix form of the fast real space summation described in section 6.2 is used, then the cost of that is incurred only the first time, since subsequent real space computations require only the computation of a matrix-vector product (at a negligible cost in this context). The \mathbf{k} -space sum, on the other hand, has to be computed once every iteration. The total cost for a system that converges in N_{gm} iterations is then

$$t_{\text{wall}} = t_{rs} + (N_{gm} + 1) t_{se}, \quad (42)$$

including the extra SE call required for singularity subtraction, as discussed in section 7.1. The process for selecting runtime parameters would then be as follows:

1. For a given tolerance ϵ , determine the corresponding $r_c(\xi)$ and $\mathbf{M}(\xi)$ for a range of ξ . Also set the Gaussian support P to match ϵ .
2. Estimate $t_{rs}(\xi)$ and $t_{se}(\xi)$ from r_c and \mathbf{M} using tabulated values.
3. Estimate N_{gm} either by solving a coarsened problem or by experience from similar problems.
4. Minimize $t_{\text{wall}}(\xi)$ using (42) and N_{gm} .

In practice, when running larger test cases on a desktop workstation with 8 GB working memory, the benefit of having to evaluate the real space sum only once per time step is so large that it is often useful to set r_c to the maximum value such that the matrices \mathcal{A}_{jl} in (39) fit in memory. Then this r_c determines ξ for a given tolerance ϵ .

8.2 Complexity

When considering complexity in N , we must first establish what we mean when we say that N grows. We here limit ourselves to cases where the physical properties of the system remain unchanged, which can happen either by the domain growing and the particle and discretization density staying constant, or by the discretization density increasing for a given system. In both cases the factor Q/V , central to the Ewald summation truncation errors, will stay constant.

In the first case, with the domain growing at a constant particle density, N will scale as L^3 (for a cubic domain). The cutoff radius r_c can then be kept unchanged, and the real

space sum will have $\mathcal{O}(N)$ complexity, since the number of near neighbors is constant. The size M of the FFT grid will have to grow, to maintain a constant $K \sim M/L$, so we will have $M \sim L \sim N^{1/3}$. The cost of the FFT in the \mathbf{k} -space sum will then have complexity $\mathcal{O}(N \log N)$, which will also be the total asymptotic complexity.

In the second case, where the number of discretization points on a given geometry grows with N , the cutoff radius r_c will have to shrink to maintain a constant number of near neighbors and an $\mathcal{O}(N)$ complexity for the real space sum. We have $|\text{NL}_t| \sim Nr_c^2$ when considering discretization points on surfaces which are cut by the r_c sphere, so r_c will be required to scale as $N^{-1/2}$. Considering only the exponential term of the real space error estimate (32), we find that ξr_c must stay constant to maintain the same truncation error, so ξ must scale as $N^{1/2}$. Looking at the \mathbf{k} -space truncation error, and again only considering the exponential part, we see that we require $K/\xi \sim M/\xi$ constant, which implies that M must scale as $N^{1/2}$. Again the cost of the FFT is what dominates the total asymptotic complexity, which in this case becomes $\mathcal{O}(N^{3/2} \log N)$. This is higher than the complexity of the first case we considered. In practice, however, one would select a per-body discretization which gives satisfactory accuracy, and scale up the system size to simulate a system with as many particles as possible, to avoid artifacts of the periodicity assumption. This allows us to consider the method having an $\mathcal{O}(N \log N)$ complexity in practice.

9 Spatial and temporal discretization

9.1 Particle geometry and quadrature

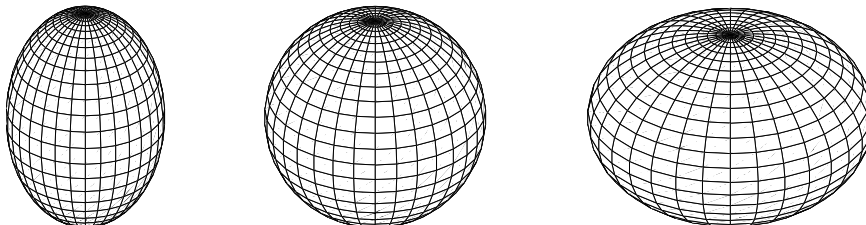


Figure 6: Examples of prolate, spherical and oblate spheroids, with respective aspect ratio ($a:c$) 2:3, 1:1 and 3:2. The mesh shows the quadrature points, which in this example are 32×32 using Gauss-Legendre and trapezoidal quadrature.

We have in this work restricted ourselves to spheroidal particles, which are defined by the surface

$$\frac{x^2 + y^2}{a^2} + \frac{z^2}{c^2} = 1,$$

when viewed in a coordinate system with the origin at $\mathbf{x}_c^{(\alpha)}$ and the z -axis parallel to the axis of symmetry. They can be classified as being either prolate ($a < c$), spherical ($a = c$) or oblate ($a > c$), as illustrated in Figure 6. It is natural to parameterize the surface of each spheroid using spherical coordinates $\varphi \in [0, 2\pi]$ and $\theta \in [0, \pi]$, and also to perform the numerical integration over the surface in those coordinates using a set of $m \times n$ quadrature points. Integration along φ is over one whole period of a periodic function, making the trapezoidal rule a good choice of quadrature in that direction, due to its spectral accuracy in such a case

[29]. In the θ direction we have applied n -point Gauss-Legendre quadrature [1, ch. 25], so the overall accuracy should be spectral in m and n for a smooth and well-resolved integrand.

9.2 Nearly singular quadrature

The above quadrature scheme performs well when the integrand is smooth, but loses accuracy rapidly when the integral is nearly singular. This is the case for the real space part of the double layer potential when two bodies are close to each other, such that some \mathbf{x}_ℓ in (40) and (41) are close to the surface of a neighboring body. One possible way of dealing with this is to use a floating partition of unity [7, 48], which evaluates the nearly singular part of the problematic integral using a local polar patch for each \mathbf{x}_ℓ . Here we have implemented a very similar approach, using a complete rotated grid instead of a local patch. Using a rotated grid gives a higher accuracy than a local patch, since there is no partition of unity function that must be resolved on the grids, but is more expensive, since there is more work involved for each \mathbf{x}_ℓ . In this work the rotated grid approach is feasible since there are a limited number of quadrature points on each particle.

When evaluating the double layer potential from a spheroidal particle with surface Γ at a point \mathbf{x} where the integral is nearly singular, we first find the projection $\mathbf{x}_p \in \Gamma$ of \mathbf{x} on Γ , such that $(\mathbf{x} - \mathbf{x}_p)/\|\mathbf{x} - \mathbf{x}_p\| = \hat{\mathbf{n}}(\mathbf{x}_p)$ using Newton’s method. We then create a rotated polar $m_p \times n_p$ grid with the north pole at \mathbf{x}_p on the spheroid, using the same quadrature rule as on the original grid, but with the θ direction quadrature points clustered closer to the pole by a sinh transformation [25]. This is to resolve the rapidly varying integrand in the nearly singular region⁴. The double layer density \mathbf{q} is interpolated from the base grid to the rotated grid using a global interpolation scheme, with barycentric Lagrange interpolation [4] in the θ direction and trigonometric interpolation in the φ direction. The real space part of the double layer potential at \mathbf{x} is then evaluated using the points on the rotated grid and the corresponding interpolated density values. Interpolating the density to the rotated grid points and then evaluating the potential at \mathbf{x} using those points is quite costly, since it has to be done for every point \mathbf{x} which is close to Γ . The whole operation can however be precomputed and stored in the real space interaction matrix \mathcal{A} described in section 6.2, so the extra work from the nearly singular points only incurs an extra cost at the first GMRES iteration.

Our implementation of this scheme allows for a hierarchy of thresholds d_i , such that the potential at \mathbf{x} is evaluated using an $m_p^{(i)} \times n_p^{(i)}$ rotated grid if the distance $d(\mathbf{x})$ between Γ and \mathbf{x} satisfies $d_{i-1} \leq d(\mathbf{x}) < d_i$. The thresholds and grid sizes must be tuned to the desired accuracy. As an example, we consider a setup in a unit cube where the Ewald and quadrature parameters have been tuned to achieve an error of around 10^{-4} , and the particle in question is an oblate spheroid with $(a, c) = (0.05, 0.1)$ and 16^2 quadrature points. The thresholds used are $(d_0, d_1, d_2) = (1, 0.5, 10^{-5}) \cdot c$, and the corresponding grids are $(16^2, 32^2, 64^2)$. We measure the error by evaluating the stresslet identity (10) at a large number of random points distributed in layers around the particle, and graph the largest error observed in each layer versus the distance between the layer and the surface. The results, shown in Figure 7, show that the accuracy of the baseline quadrature scheme deteriorates rapidly as one approaches the particle, but that the errors can be controlled quite well by using rotated grids.

⁴A similar approach, but using a floating partition of unity instead of a complete rotated grid, was used by Zhao et al. [49] to resolve close interactions when simulating flowing blood cells.

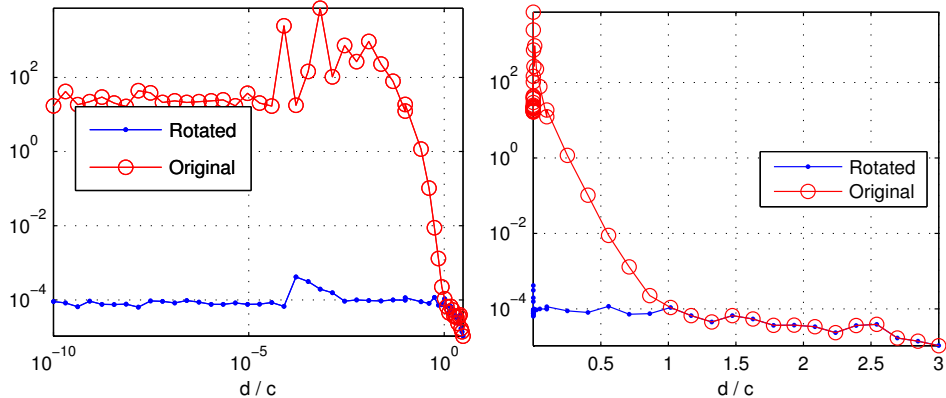


Figure 7: Largest observed error when evaluating the stresslet identity (10), which should evaluate to zero, at a distance d from the particle surface, using both the original grid and the rotated grid. The particle is an oblate spheroid, $(a, c) = (0.05, 0.1)$, with 16^2 quadrature points. The left and right plots contain the same data, but with different scaling on the x -axis.

9.3 Time stepping

We have so far only discussed how to compute the double layer density \mathbf{q} and the rigid body motion vectors $\mathbf{U}^{(\alpha)}$ and $\mathbf{\Omega}^{(\alpha)}$ for a given geometry of particles. However, to simulate the sedimentation of particles, we also need to move them forward in time. This is naturally done using a Lagrangian representation of the rigid body motion of the particles. For each particle it is then sufficient to keep track of the centroid position $\mathbf{x}_c^{(\alpha)}(t)$ and the coordinate frame vectors $\mathbf{D}^{(\alpha,i)}(t)$, $i = 1, 2, 3$, describing the angular position of the particle. Their evolution is described by an ODE,

$$\begin{aligned} \frac{d\mathbf{x}_c^{(\alpha)}}{dt} &= \mathbf{U}^{(\alpha)}(t), \\ \frac{d\mathbf{D}^{(\alpha,i)}}{dt} &= \mathbf{\Omega}^{(\alpha)}(t) \times \mathbf{D}^{(\alpha,i)}(t), \quad i = 1, 2, 3, \end{aligned}$$

where $\mathbf{U}^{(\alpha)}(t)$ and $\mathbf{\Omega}^{(\alpha)}(t)$ are a function of $\mathbf{x}_c^{(\alpha)}(t)$ and $\mathbf{D}^{(\alpha,i)}(t)$. This system can readily be solved using a numerical time stepping method. However, to make the rigid body rotations preserve the orthonormality of the coordinate frame vectors over time, we choose to formulate them using unit quaternion operations (described in Appendix A). We then let the unit quaternion $\vec{\zeta}^{(\alpha)}(t)$ represent the rotation of a particle since the initial state, such that

$$\mathbf{D}^{(\alpha,i)}(t) = Z\left(\vec{\zeta}^{(\alpha)}(t)\right) \mathbf{D}^{(\alpha,i)}(0),$$

where $Z(\vec{\zeta})$ is the rotation matrix corresponding to $\vec{\zeta}$. The ODE which we need to integrate in time is then

$$\begin{aligned} \frac{d\mathbf{x}_c^{(\alpha)}}{dt} &= \mathbf{U}^{(\alpha)}(t), \\ \frac{d\vec{\zeta}^{(\alpha)}}{dt} &= \frac{1}{2}[0, \mathbf{\Omega}^{(\alpha)}(t)] * \vec{\zeta}^{(\alpha)}(t), \end{aligned}$$

where $\vec{\zeta}^{(\alpha)}(0) = [1, \mathbf{0}]$ and $*$ is the quaternion product. In this work we solve this using the Bogacki-Shampine 3(2) method [5], which is a third order Runge-Kutta method with an embedded second order formula used for estimating the local time stepping error. The time step Δt is adaptively adjusted to keep the error estimate below a given tolerance.

10 Numerical results

10.1 Validation

Periodic array of spheres

To validate our method, we begin by computing the sedimentation velocity on a simple cubic (SC) array of spheres (see Figure 8a). The primary cell then contains a single sphere of radius a , subject to a gravitational force F . The sedimentation velocity U is computed by solving (40), and the drag coefficient K is then computed as $K = F/6\pi\mu aU$. This is a well-studied problem, and reference values of K for different concentrations $\rho = 4\pi a^3/3V$ are available from Zick & Homsy [50] and Sangani & Acrivos [42]. The same problem was used by Leiderman et al. [30] to validate their regularization method for the periodic single layer potential, and by Sierou & Brady [44] for their accelerated Stokesian dynamics (ASD).

The system is solved to high precision ($\epsilon = 10^{-12}$ for GMRES and Ewald summation), to isolate quadrature errors. Furthermore, the directions of the gravitational force and axis of symmetry of the sphere are randomized (and not parallel), in an attempt to avoid any errors being hidden by symmetries of the problem. The results are presented in Table 1 together with the reference values K_{ref} , which are presented in the precision tabulated in the original sources [42, 50]. We see that we get very close to the reference values when using 64×64 points on the sphere, which we take as evidence that our computations are correct. We also note that the errors increase with the concentration for both quadrature methods, meaning that the problem gets harder to solve the more tightly packed the spheres are. This was also observed in [30].

| ρ | 8×8 | 16×16 | 32×32 | 64×64 | K_{ref} |
|----------|--------------|----------------|----------------|----------------|------------------|
| 0.000125 | 1.0963 | 1.0963 | 1.0963 | 1.0963 | 1.096 |
| 0.001000 | 1.2121 | 1.2121 | 1.2121 | 1.2121 | 1.212 |
| 0.014137 | 1.6998 | 1.6999 | 1.6999 | 1.6999 | 1.7000 |
| 0.065450 | 2.8416 | 2.8418 | 2.8420 | 2.8420 | 2.8420 |
| 0.113097 | 3.9736 | 3.9731 | 3.9737 | 3.9738 | 3.9738 |
| 0.179594 | 5.9988 | 6.0014 | 6.0037 | 6.0040 | 6.004 |

Table 1: Computed drag coefficients K for periodic array of spheres at different concentrations ρ , compared to reference values K_{ref} , for $m \times n$ quadrature points.

Spheres in a line

As a second validation test, we consider a test case which was studied in free space by Tran-Cong & Phan-Thien [46] using the boundary element method and, more recently, by Wilson [47] using a modification to the method of reflections. Three spheres of radius a are positioned along a straight line in the xy plane, with center-to-center distances as . They are subject to

a gravitational force $\mathbf{f} = -6\pi a\mu\hat{\mathbf{z}}$, which would produce a unit Stokes velocity $U_s = 1$ on an isolated sphere. The outer spheres sediment with a vertical velocity U_1 and rotate with an angular velocity Ω , while the center sphere sediments with a vertical velocity U_0 and does not rotate.

The results available in the literature for this test problem are for three particles in free space, while our method computes the flow in periodic particle systems. However, the periodic results should converge to the free space results as the ratio between the radius a and the periodic box size L goes to zero. To test this, we set up the case in a periodic box and compute the velocities for different a and for $s = 2.05$ and $s = 2.10$, using 32×32 quadrature points on each sphere. In Table 2 we present our results together with the numerical values reported by both Tran-Cong and Wilson, as tabulated by Wilson. The values presented are the dimensionless drag $D_i = U_s/U_i$ and the dimensionless angular velocity $\Omega = |\boldsymbol{\Omega}|/(U_s/a)$. For the smallest ratio of radius to periodic length all values are equal (in the presented precision) to those reported by Wilson in the free space case, except for the fourth decimal in the angular velocity at $s = 2.05$, which is also the most difficult of the two cases, since the separation is smaller (only $1/20$ of a radius). It is worth noting that the angular velocity appears to be much less dependent on a/L ratio than the drag coefficients.

| | $s = 2.05$ | | | $s = 2.10$ | | |
|-----------------|------------|--------|----------|------------|--------|----------|
| | D_0 | D_1 | Ω | D_0 | D_1 | Ω |
| $a/L = 10^{-2}$ | 0.5837 | 0.6631 | 0.1932 | 0.5891 | 0.6698 | 0.1918 |
| $a/L = 10^{-4}$ | 0.5564 | 0.6280 | 0.1932 | 0.5613 | 0.6340 | 0.1918 |
| $a/L = 10^{-6}$ | 0.5561 | 0.6277 | 0.1932 | 0.5610 | 0.6337 | 0.1918 |
| $a/L = 10^{-8}$ | 0.5561 | 0.6277 | 0.1933 | 0.5610 | 0.6337 | 0.1919 |
| Wilson | 0.5561 | 0.6277 | 0.1934 | 0.5610 | 0.6337 | 0.1919 |
| Tran-Cong | 0.557 | 0.629 | 0.197 | 0.562 | 0.634 | 0.192 |

Table 2: Dimensionless drag D_i and angular velocities Ω for three spheres of radius a in a line with center separation as , computed in a periodic box of size L^3 and compared to free space results by Wilson [47] and Tran-Cong [46].

10.2 Convergence

Periodic array of spheres

To quantify our observations from the first validation test, we continue to work with the SC array of spheres, and measure the order of convergence in the drag coefficient K as the number of grid points in both directions on the surface is successively doubled. We do this for $m = n = 8, 16, 32, 64$ and concentration $\rho = 10^{-p}$, $p = 1, \dots, 6$, and measure the relative error as $e_r(n) = |K(n) - K(2n)|/K(2n)$. The results, graphed in Figure 8b, show a clear third order convergence in $h = 1/n$, with an error constant that increases with the concentration ρ . To explain this one can view the singularity subtraction (12) as only removing the first term of a series expansion of the source density around the singularity. The remaining leading order term that will contribute to the quadrature error is then $\mathcal{O}(h)$. The average quadrature weight at a point is $\mathcal{O}(h^2)$, so the final error term will be $\mathcal{O}(h^3)$. To illustrate the interplay between the polynomial convergence from the singularity and the spectral convergence from the quadrature of the smooth parts, we redo the computations for $m = n = 3, \dots, 16$ and com-

pute the error compared to a highly resolved reference value, $e_r(n) = |K(n) - K_{ref}|/K_{ref}$. In Figure 8c one can clearly observe the spectral convergence before the polynomial error starts to dominate, again with an error that increases with concentration ρ .

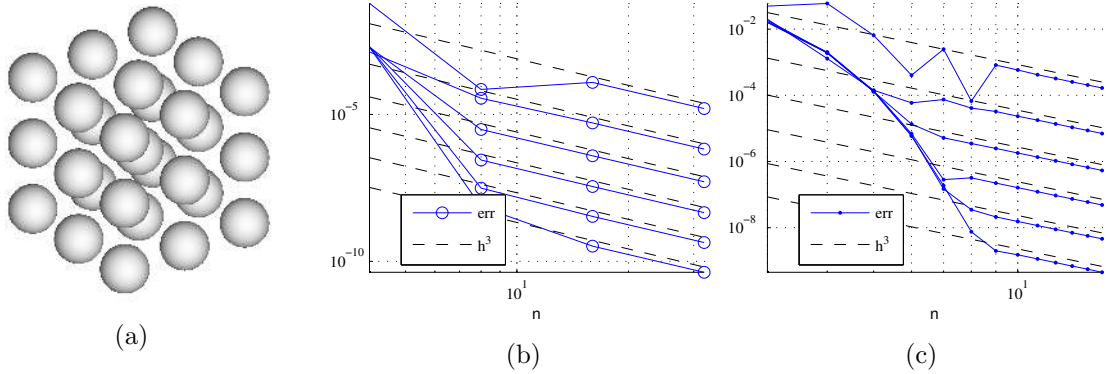


Figure 8: **(a)** Simple cubic array of spheres, concentration $\rho = 0.18$. **(b)** Error in drag coefficient measured as difference between refinements, compared to h^3 reference line. Concentration ρ increases by factors of 10 from 10^{-6} (bottom) to 10^{-1} (top). **(c)** Error measured against reference solution for small n , showing how the polynomial error starts to dominate over the spectral error at a some point.

Velocity components

As a final convergence check, we study the components of $\mathbf{U}^{(\alpha)}$ and $\mathbf{\Omega}^{(\alpha)}$, computed by (7) and (8), for two prolate bodies (aspect ratio 3:4) sedimenting in the primary cell, shown in Figure 9a. Figure 9b displays the relative error, measured between successive refinements, in all components as $m = n = 8, 16, 32, 64$. Results are satisfactory, showing h^3 convergence in all 12 components for $n \geq 16$.

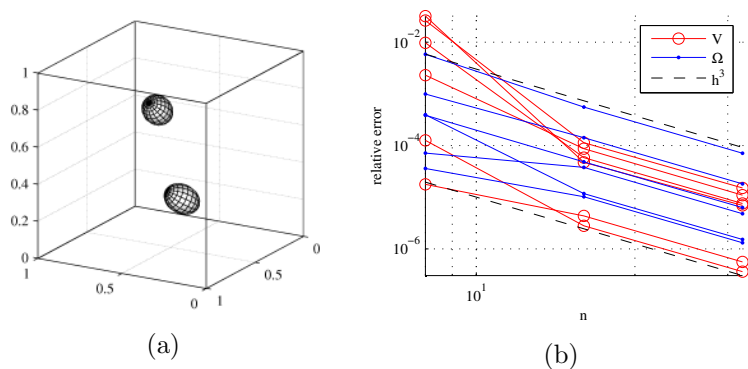


Figure 9: **(a)** Configuration of sedimenting prolate bodies. **(b)** Relative error, measured between refinements, in all six translational (o) and rotational (\cdot) components of sedimenting prolate bodies, compared to h^3 reference (- -).

10.3 Sedimentation

Symmetric motion

As a numerical experiment for illustrating sedimentation dynamics, we consider a setup of eight oblate bodies with axes $a = 1/10$ and $c = 1/20$ in a cubic primary cell, $L = 1$, all subject to a downward gravitational force $\mathbf{f} = -6\pi a\mu\hat{\mathbf{z}}$. At time $t = 0$ the bodies are horizontally aligned in two layers, separated by a vertical distance of 0.5 and symmetrically oriented around the vertical line $x = y = 0.5$, at a distance $r = 0.2\sqrt{2}$ and $r = 0.3\sqrt{2}$ from it (see Figure 10b). The bodies are discretized using 16×16 points. The resulting system has a total of 6144 unknowns and is solved to a tolerance $\epsilon = 10^{-4}$, which takes around 20 iterations with GMRES. The Runge-Kutta scheme, also using a tolerance of $\epsilon = 10^{-4}$, requires three solves per time step, which takes around 5–10 seconds on a quad-core desktop computer, depending on how many points are close enough to require evaluation using rotated grids.

This highly symmetrical setup results in a periodic sedimentation behavior, shown in Figures 10a and 10c. We observe that the symmetries are well preserved by the numerical method over several periods, as the vertical velocities $w(t)$ and orientations $\theta(t)$ of all 8 particles collapse onto the same lines in Figure 10a, and return to their initial values after one period of length $T \approx 10.24$. The motion of the particles is very similar to the tumbling orbits observed by Jung et al. [26] for sedimenting spheroids in free space, both in experiments and simulations.

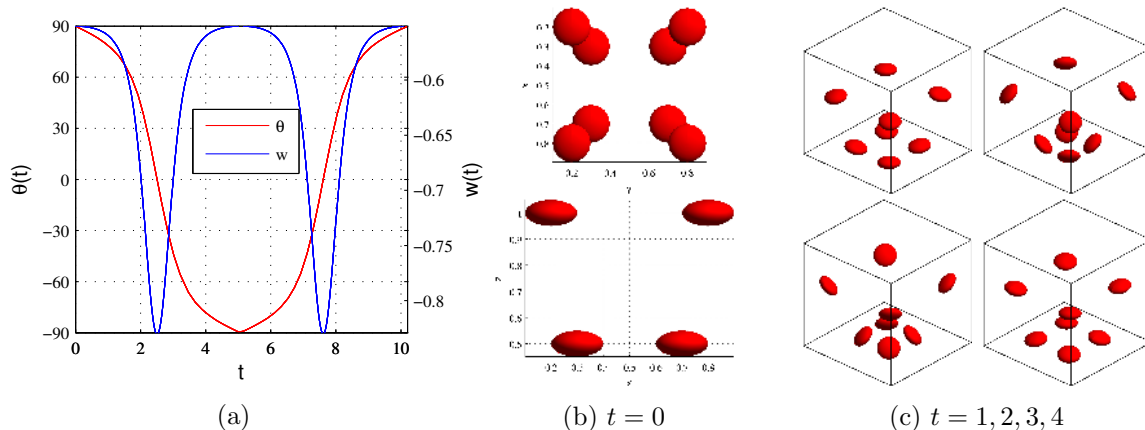


Figure 10: **(a)** Vertical velocity $w(t)$ and angle θ between symmetry axis and $\hat{\mathbf{z}}(t)$ during periodic tumbling. **(b)** Symmetric initial configuration of 8 particles. **(c)** Snapshots of particles' tumbling motion.

We next break the initial symmetries of the above setup of particles by moving one particle horizontally to a distance $r = 0.25\sqrt{2}$ from the center line, see Figure 11b. The resulting motion, shown in Figures 11a and 11c, quickly deviates from the one observed for the completely symmetric setup and becomes seemingly chaotic. However, the diagonal symmetry which is still present in the horizontal plane is preserved under the motion, which can be seen in Figure 11a, where the motions of 8 particles are described by 6 unique velocity and orientation curves. As the chaotic sedimentation motion develops, the particles start to form a cluster which sediments faster than the average velocity during the periodic motion. This behavior is also observed in simulations of sedimenting fibers, see for example Butler & Shaqfeh [8].

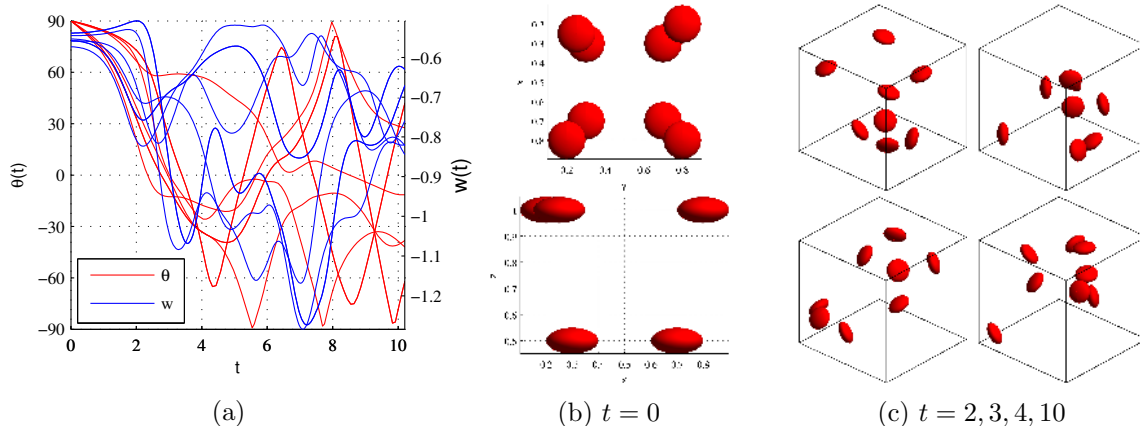


Figure 11: **(a)** Vertical velocity $w(t)$ and angle θ between symmetry axis and $\hat{\mathbf{z}}(t)$ during chaotic sedimentation. **(b)** Initial configuration of 8 particles with broken symmetry. **(c)** Snapshots of particles' motion, which exhibits chaotic behavior.

Chaotic motion

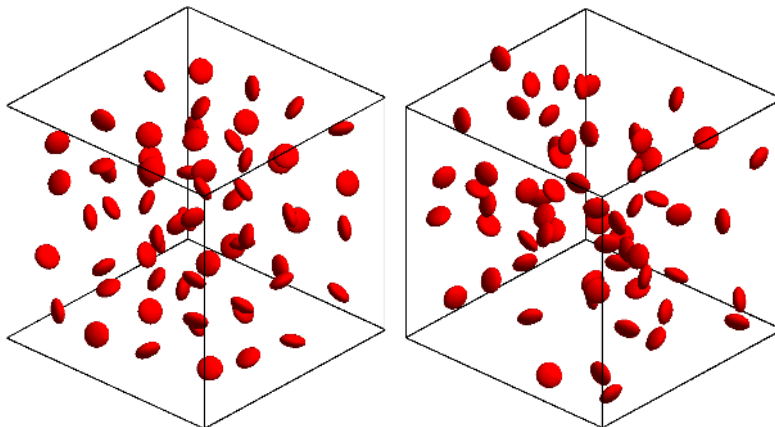


Figure 12: Completely unstructured motion at $t = 3$ (right) for 64 oblate particles, starting from perturbed initial positions and random orientations (left).

We extend the above case with sedimenting oblate bodies to a box of dimension $(2, 2, 2)$, containing 64 bodies with 16×16 quadrature points, random initial orientations and initial positions which are randomly perturbed from an equispaced setup. Using the same computer and tolerance as above, the solution now takes around 40 seconds per time step and the number of GMRES iterations per equation solve is the same as for 16 particles, so the scaling and conditioning are good. In the solution we note that the particles quickly develop a chaotic, tumbling motion. The tendency of the oblate particles is to form aligned pairs, which sediment rapidly before breaking up. We also observe a tendency of the particles to form groups or clusters containing several particles, sedimenting more rapidly than isolated particles. However, the particles tend to get very close to each other during the process, which presents a problem to our numerical scheme. As the shortest distance between two colliding particles becomes very small ($\mathcal{O}(10^{-5})$), two things tend to happen: First, the interactions

get very hard to resolve, even using the rotated grid scheme, causing large variations in the particle velocities even for small movements. Second, the time stepping algorithm tends to overstep, causing the particle surfaces to overlap before the step is rejected. These two factors make the time stepping algorithm adjust the time step Δt to a very short ($\mathcal{O}(10^{-5})$) length, which in practice causes the computation to stall. This could be improved by developing a quadrature method that is able to accurately handle the interactions of very close particles, minimizing the numerical errors that cause the particle velocities to fluctuate as the particles make only small movements. The numerical time stepping scheme would then be able to better select a suitable time step. If this remedy is not sufficient, then implicit or semi-implicit time-stepping strategies must be investigated. This would however incur a substantial extra cost in each time step.

11 Conclusions

In this paper we have presented a method for simulating sedimentation in periodic particle suspensions where the long-range particle interactions are computed using a fast $\mathcal{O}(N \log N)$ algorithm, given an underlying quadrature scheme. The method is based on a boundary integral formulation using the double layer (stresslet) potential, which results in a system that is well-conditioned and converges rapidly under iterative solution using GMRES. We have discussed the periodic properties of the stresslet, and derived an expression for the $\mathbf{k} = 0$ component to eliminate the mean flow which arises from the periodic formulation. The computations of the periodic potential are performed using the spectral Ewald method, which we have adapted for use with the stresslet. We have developed predictive truncation error estimates for the Ewald decomposition of the stresslet, and described how these estimates are used to optimize the efficiency of the computations.

Using a basic quadrature scheme and applying the method to a number of test problems, we find that the method performs well both in terms of efficiency and accuracy; the method is well-conditioned and scales well when introducing more particles. The results are also in accordance with available reference data for a number of test cases. The dominating error of the method is $\mathcal{O}(h^3)$ and comes from the singularity in the double layer potential, which is treated by singularity subtraction. The singularity subtraction only works for target points on the surface of the same particle, and not for nearby points on neighboring particles or in the domain. For this nearly singular case we have implemented a method of rotated grids, which is able to resolve rather close interactions well, but comes at a high computational cost. Still, the combination of well-known quadrature and time-stepping algorithms implemented in this work struggles as particles get very close to each other.

A natural direction of continued work will be to develop efficient quadrature techniques that can very accurately handle both the singular and nearly singular behavior of the double layer potential, also for very small separation distances. Together with an adaptive time stepping procedure, this should enable the method to be used for flows where the particles cluster close together and nearly collide. The framework presented here is both efficient and has favorable scaling properties, and with the above mentioned improvements, large scale time dependent simulations of particle suspensions should be made possible. This will allow the study of suspension dynamics and its dependence both on concentration and particle shape.

Acknowledgments

We are grateful to Dr. Dag Lindbo for providing us with the thoroughly optimized fast Gaussian gridding code used in the spectral Ewald method for the stresslet, as well as the entire spectral Ewald implementation for the stokeslet.

This work has been supported by the Swedish Research Council under grant no. 2011-3178. The authors gratefully acknowledge this support.

Appendix A Quaternion operations

A unit quaternion \vec{q} describing a 3D rotation can be written as [43]

$$\vec{q} = [s, \mathbf{v}],$$

where $s \in \mathbb{R}$, $\mathbf{v} \in \mathbb{R}^3$ and

$$|\vec{q}| = \sqrt{s^2 + v_1^2 + v_2^2 + v_3^2} = 1.$$

The corresponding rotation matrix is

$$Z(\vec{q}) = \begin{pmatrix} 1 - 2v_2^2 - 2v_3^2 & 2v_1v_2 - 2sv_3 & 2v_1v_3 + 2sv_2 \\ 2v_1v_2 + 2sv_3 & 1 - 2v_1^2 - 2v_3^2 & 2v_2v_3 - 2sv_1 \\ 2v_1v_3 - 2sv_2 & 2v_2v_3 - sv_1 & 1 - 2v_1^2 - 2v_2^2 \end{pmatrix}.$$

When simulating rigid body motion, one can let $\vec{q}(t)$ represent the angular position of a body with angular velocity $\boldsymbol{\omega}(t)$. The time evolution of $\vec{q}(t)$ is then governed by an ODE [36],

$$\begin{aligned} \frac{d\vec{q}}{dt} &= \frac{1}{2}[0, \boldsymbol{\omega}(t)] * \vec{q}(t), \\ \vec{q}(0) &= [1, \mathbf{0}], \end{aligned}$$

which can be integrated in time using an explicit method. The quaternion product ($*$) is defined as

$$\vec{q}_3 = \vec{q}_1 * \vec{q}_2 = [s_1s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1\mathbf{v}_2 + s_2\mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2],$$

and has the properties that $|\vec{q}_3| = 1$ and that \vec{q}_3 represents the sequence of rotations,

$$Z(\vec{q}_3) = Z(\vec{q}_1)Z(\vec{q}_2).$$

Applying a sequence of orthonormal rotation matrices will in limited precision result in a matrix that is not orthonormal, and there is no well-defined way of re-orthonormalizing a rotation matrix. When using quaternions one applies the sequence of rotations as quaternion products, and computes the rotation matrix using the final result. The rotation matrix is then guaranteed to be orthonormal. The quaternion product can still drift away from unit in finite precision, but the re-normalization is then well defined as $\vec{q} \rightarrow \vec{q}/|\vec{q}|$.

References

- [1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. U.S. Govt. Print. Off, Washington, 10 edition, 1972.
- [2] G. K. Batchelor. Sedimentation in a dilute dispersion of spheres. *J. Fluid Mech.*, 52(02):245, 1972, doi:10.1017/S0022112072001399.
- [3] C. W. J. Beenakker. Ewald sum of the RotnePrager tensor. *The J. Chem. Phys.*, 85(3):1581, 1986, doi:10.1063/1.451199.
- [4] J.-P. Berrut and L. N. Trefethen. Barycentric Lagrange Interpolation. *SIAM Rev.*, 46(3):501–517, 2004, doi:10.1137/S0036144502417715.
- [5] P. Bogacki and L. Shampine. A 3(2) pair of Runge - Kutta formulas. *Appl. Math. Lett.*, 2(4):321–325, 1989, doi:10.1016/0893-9659(89)90079-7.
- [6] J. F. Brady and G. Bossis. Stokesian Dynamics. *Annu. Rev. Fluid Mech.*, 20(1):111–157, 1988, doi:10.1146/annurev.fl.20.010188.000551.
- [7] O. P. Bruno and L. a. Kunyansky. A Fast, High-Order Algorithm for the Solution of Surface Scattering Problems: Basic Implementation, Tests, and Applications. *J. Comput. Phys.*, 169(1):80–110, 2001, doi:10.1006/jcph.2001.6714.
- [8] J. E. Butler and E. S. G. Shaqfeh. Dynamic simulations of the inhomogeneous sedimentation of rigid fibres. *J. Fluid Mech.*, 468:205–237, 2002, doi:10.1017/S0022112002001544.
- [9] R. H. Davis and A. Acrivos. Sedimentation of Noncolloidal Particles at Low Reynolds Numbers. *Annu. Rev. Fluid Mech.*, 17(1):91–118, 1985, doi:10.1146/annurev.fl.17.010185.000515.
- [10] J. W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 1997, doi:10.1137/1.9781611971446.
- [11] M. Deserno and C. Holm. How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines. *The J. Chem. Phys.*, 109(18):7678, 1998, doi:10.1063/1.477414.
- [12] P. P. Ewald. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Annalen der Physik*, 369(3):253–287, 1921, doi:10.1002/andp.19213690304.
- [13] X. Fan, N. Phan-Thien, and R. Zheng. Completed double layer boundary element method for periodic suspensions. *Zeitschrift für angewandte Math. und Physik*, 49(2):167–193, 1998, doi:10.1007/s000330050214.
- [14] D. Frenkel and B. Smit. *Understanding Molecular Simulation*. Academic Press, London, 2nd edition, 2001.
- [15] M. Frigo and S. Johnson. The Design and Implementation of FFTW3. *Proc. IEEE*, 93(2):216–231, 2005, doi:10.1109/JPROC.2004.840301.

- [16] O. Gonzalez. On Stable, Complete, and Singularity-Free Boundary Integral Formulations of Exterior Stokes Flow. *SIAM J. on Appl. Math.*, 69(4):933–958, 2009, doi:10.1137/070698154.
- [17] I. Gradshteyn and I. Ryzhik. *Table of integrals, series, and products*. Academic Press, Oxford, 7th edition, 2007.
- [18] L. Greengard and J.-Y. Lee. Accelerating the Nonuniform Fast Fourier Transform. *SIAM Rev.*, 46(3):443–454, 2004, doi:10.1137/S003614450343200X.
- [19] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987, doi:10.1016/0021-9991(87)90140-9.
- [20] E. Guazzelli. Sedimentation of small particles: how can such a simple problem be so difficult? *C. R. Mecanique*, 334(8-9):539–544, 2006, doi:10.1016/j.crme.2006.07.009.
- [21] E. Guazzelli and J. Hinch. Fluctuations and Instability in Sedimentation. *Annu. Rev. Fluid Mech.*, 43(1):97–116, 2011, doi:10.1146/annurev-fluid-122109-160736.
- [22] K. Gustavsson and A.-K. Tornberg. Gravity induced sedimentation of slender fibers. *Phys. Fluids*, 21(12):123301, 2009, doi:10.1063/1.3273091.
- [23] 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(02):317–328, 2006, doi:10.1017/S0022112059000222.
- [24] G. Hou, J. Wang, and A. Layton. Numerical Methods for Fluid-Structure Interaction - A Review. *Commun. Comput. Phys.*, 12(2):337–377, 2012, doi:10.4208/cicp.291210.290411s.
- [25] P. R. Johnston and D. Elliott. A sinh transformation for evaluating nearly singular boundary element integrals. *Int. J. for Numer. Methods Eng.*, 62(4):564–578, 2005, doi:10.1002/nme.1208.
- [26] S. Jung, S. Spagnolie, K. Parikh, M. Shelley, and A.-K. Tornberg. Periodic sedimentation in a Stokesian fluid. *Phys. Rev. E*, 74(3):035302, 2006, doi:10.1103/PhysRevE.74.035302.
- [27] S. Kim and S. J. Karrila. *Microhydrodynamics: Principles and Selected Applications*. Dover Publications, New York, 2005.
- [28] J. Kolafa and J. W. Perram. Cutoff Errors in the Ewald Summation Formulae for Point Charge Systems. *Mol. Simul.*, 9(5):351–368, 1992, doi:10.1080/08927029208049126.
- [29] A. Kurganov and J. Rauch. The Order of Accuracy of Quadrature Formulae for Periodic Functions. In A. Bove, D. Del Santo, and M. V. Murthy, editors, *Advances in Phase Space Analysis of Partial Differential Equations*, pages 155–159. Birkhäuser Boston, Boston, 2009.
- [30] K. Leiderman, E. L. Bouzarth, R. Cortez, and A. T. Layton. A regularization method for the numerical solution of periodic Stokes flow. *J. Comput. Phys.*, 236:187–202, 2013, doi:10.1016/j.jcp.2012.09.035.

- [31] D. Lindbo and A.-K. Tornberg. Spectrally accurate fast summation for periodic Stokes potentials. *J. Comput. Phys.*, 229(23):8994–9010, 2010, doi:10.1016/j.jcp.2010.08.026.
- [32] D. Lindbo and A.-K. Tornberg. Fast and spectrally accurate summation of 2-periodic Stokes potentials. arXiv:1111.1815v1 [physics.flu-dyn], 2011.
- [33] D. Lindbo and A.-K. Tornberg. Spectral accuracy in fast Ewald-based methods for particle simulations. *J. Comput. Phys.*, 230(24):8744–8761, 2011, doi:10.1016/j.jcp.2011.08.022.
- [34] O. Marin. *Boundary integral methods for Stokes flow : Quadrature techniques and fast Ewald methods*. PhD thesis, KTH Royal Institute of Technology, 2012, <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-105540>, .
- [35] MATLAB. *version 7.12.0.635 (R2011a)*. The MathWorks Inc., Natick, Massachusetts, 2011.
- [36] I. Omelyan. Algorithm for numerical integration of the rigid-body equations of motion. *Phys. Rev. E*, 58(1):1169–1172, 1998, doi:10.1103/PhysRevE.58.1169.
- [37] H. Power and G. Miranda. Second Kind Integral Equation Formulation of Stokes Flows Past a Particle of Arbitrary Shape. *SIAM J. on Appl. Math.*, 47(4):689–698, 1987, doi:10.1137/0147047.
- [38] C. Pozrikidis. *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge University Press, Cambridge, 1992, doi:10.1017/CBO9780511624124.
- [39] C. Pozrikidis. Computation of periodic Green’s functions of Stokes flow. *J. Eng. Math.*, 30(1-2):79–96, 1996, doi:10.1007/BF00118824.
- [40] Y. Saad and M. H. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. on Sci. Stat. Comput.*, 7(3):856–869, 1986, doi:10.1137/0907058.
- [41] D. Saintillan, E. Darve, and E. S. G. Shaqfeh. A smooth particle-mesh Ewald algorithm for Stokes suspension simulations: The sedimentation of fibers. *Phys. Fluids*, 17(3):033301, 2005, doi:10.1063/1.1862262.
- [42] A. Sangani and A. Acrivos. Slow flow through a periodic array of spheres. *Int. J. Multiph. Flow*, 8(4):343–360, 1982, doi:10.1016/0301-9322(82)90047-7.
- [43] K. Shoemake. Animating rotation with quaternion curves. *ACM SIGGRAPH Comput. Graph.*, 19(3):245–254, 1985, doi:10.1145/325165.325242.
- [44] A. Sierou and J. F. Brady. Accelerated Stokesian Dynamics simulations. *J. Fluid Mech.*, 448:115–146, 2001, doi:10.1017/S0022112001005912.
- [45] A.-K. Tornberg and L. Greengard. A fast multipole method for the three-dimensional Stokes equations. *J. Comput. Phys.*, 227(3):1613–1619, 2008, doi:10.1016/j.jcp.2007.06.029.

- [46] T. Tran-Cong and N. Phan-Thien. Stokes problems of multiparticle systems: A numerical method for arbitrary flows. *Phys. Fluids A: Fluid Dyn.*, 1(3):453–461, 1989, doi:10.1063/1.857414.
- [47] H. J. Wilson. Stokes flow past three spheres. *J. Comput. Phys.*, 245:302–316, 2013, doi:10.1016/j.jcp.2013.03.020.
- [48] L. Ying, G. Biros, and D. Zorin. A high-order 3D boundary integral equation solver for elliptic PDEs in smooth domains. *J. Comput. Phys.*, 219(1):247–275, 2006, doi:10.1016/j.jcp.2006.03.021.
- [49] H. Zhao, A. H. Isfahani, L. N. Olson, and J. B. Freund. A spectral boundary integral method for flowing blood cells. *J. Comput. Phys.*, 229(10):3726–3744, 2010, doi:10.1016/j.jcp.2010.01.024.
- [50] A. A. Zick and G. M. Homsy. Stokes flow through periodic arrays of spheres. *J. Fluid Mech.*, 115:13–26, 1982, doi:10.1017/S0022112082000627.