Journal of Computational Physics 229 (2010) 8994–9010

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Spectrally accurate fast summation for periodic Stokes potentials

Dag Lindbo *, Anna-Karin Tornberg

Numerical Analysis/Linné Flow Centre, Royal Institute of Technology (KTH), 10044 Stockholm, Sweden

ARTICLE INFO

Article history:
Received 19 January 2010
Received in revised form 12 August 2010
Accepted 20 August 2010
Available online 27 August 2010

Keywords:
Viscous flow
Stokes equations
Potential theory
Ewald summation
FFT
Spectral accuracy

ABSTRACT

A spectrally accurate method for the fast evaluation of N-particle sums of the periodic Stokeslet is presented. Two different decomposition methods, leading to one sum in real space and one in reciprocal space, are considered. An FFT based method is applied to the reciprocal part of the sum, invoking the equivalence of multiplications in reciprocal space to convolutions in real space, thus using convolutions with a Gaussian function to place the point sources on a grid. Due to the spectral accuracy of the method, the grid size needed is low and also in practice, for a fixed domain size, independent of N. The leading cost, which is linear in N, arises from the to-grid and from-grid operations. Combining this FFT based method for the reciprocal sum with the direct evaluation of the real space sum, a spectrally accurate algorithm with a total complexity of $\mathcal{O}(N \log N)$ is obtained. This has been shown numerically as the system is scaled up at constant density.

1. Introduction

Boundary integral and potential methods have successfully been used to solve Stokes equations for half a century. The elementary solution (Green's function) for the velocity field for the free-space problem is given by the Stokeslet, or Oseen-Burgers tensor, by

$$S(\mathbf{x}) = \frac{\mathbf{I}}{|\mathbf{x}|} + \frac{\mathbf{x}\mathbf{x}}{|\mathbf{x}|^3}. \tag{1}$$

See [1] for an introduction to viscous flow and boundary integral methods. Here, we consider a system of N point sources at location $\mathbf{x}_n$ with strength $\mathbf{f}_n$, $n = 1, 2, \ldots, N$. In the periodic setting, the velocity field is given as a sum over these point sources and all periodic images,

$$\mathbf{u}(\mathbf{x}) = \sum_{n=1}^{N} \sum_{\mathbf{p}} S(\mathbf{x} - \mathbf{x}_n + \mathbf{p})\mathbf{f}_n, \tag{2}$$

where $\mathbf{p}$ form the discrete set $\{[iL_x\ jL_y\ kL_z] : (i,j,k) \in \mathbb{Z}^3\}$ and L_x, L_y and L_z are the periodic lengths in the x, y, and z directions respectively. When discretizing boundary integral equations one ends up with a sum equivalent to (2). The present work is concerned with the efficient evaluation of this conditionally convergent sum.

The fundamental difficulty with summing (2) is the $\sim 1/r$ decay, similar to the decay in the Coulomb potential. The summation method introduced by Ewald [2] to treat electrostatic problems is commonly used, and work on particle mesh (PME) methods has followed. See Darden et. al. [3] and Essmann et. al. [4] for pioneering work in speeding up Ewald summation using fast Fourier transforms (FFT). Deserno and Holm [5] present a comparison of three gridded Ewald methods

* Corresponding author. Tel.: +46 8 790 6927.
E-mail addresses: dag@kth.se (D. Lindbo), annak@nada.kth.se (A.-K. Tornberg).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.08.026

for the Coulomb potential in the periodically closed case and state the methods in a unified framework in terms of convolutions. Shan et. al. [6] also give an overview of gridded Ewald methods in a convolution framework, and introduce the Gaussian split Ewald (GSE) method. We shall follow a similar line of reasoning when developing our method for the Stokes problem. It is well known that these methods reduce the computational cost of direct Ewald summation from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$.

Other related work include fast multi-pole methods (FMM) introduced by Greengard and Rokhlin [7], recently extended to the Stokes problem by Tornberg and Greengard [8] and Wang et. al. [9]. These methods are known to be $\mathcal{O}(N)$ efficient, but have not been developed for periodicity in 3D. For 2D Stokes flow an FMM-accelerated boundary integral method was introduced by Greengard and Kropinski [10], and by Ying, Biros and Zorin [11] for a Dirichlet problem in 3D.

Early work on applying Ewald's ideas to the infinite sum of Stokeslets in (2) was done by Hasimoto [12], and a more recent approach follows work by Beenakker [13] on a related problem. Grid-based (PME) methods have been developed for Stokes, e.g. by Saintillan et. al. [14], and by Sierou and Brady [15]. The latter work is an extension of the *Stokesian Dynamics* method for many-body hydrodynamic interactions (cf. Brady and Bossis [16]), which introduces a particle-mesh method to get a more efficient means of evaluating far-field hydrodynamic interactions. The authors evaluate the accuracy of their method extensively from the point of view of fluid mechanics, but leave the impression that the numerical accuracy of their method is fairly low. Saintillan et. al. present a method for sedimentation of rigid fibers, where, again, a particle-mesh method is introduced to improve the computational complexity of the method. Here, the smoothness of the underlying interpolation method is identified as an important factor in the accuracy of the PME method.

Our work should be seen as an alternative to the PME methods from the aforementioned work involving Stokes. We shall follow the presentation by Pozrikidis [17], where the Beenakker approach is applied to Stokes, and focus on analyzing the proposed PME method analytically and evaluate it only in terms of numerical accuracy and computational complexity. Our position is that the PME method should preserve the spectral reciprocal space convergence properties of the underlying Ewald summation method. This is accomplished in our method by avoiding interpolation completely, which is used in some form by the works cited. Instead, our method is based on convolutions with suitably scaled Gaussians. We shall see that the only approximation needed is an integral of a periodic function. Spectral accuracy, which here means that all errors decay at least as fast as $e^{-\alpha K^2}$ as K grows, is significant because it facilitates high-accuracy simulations where the accumulated effects of interpolation error would otherwise affect accuracy as the number of source points grows. Hence, we are able to scale our method to large systems without going to expensive computational grids. Furthermore, our method appears to provide a substantial simplification over previous work – it can be implemented with no more than 50 lines of Matlab code. Even with a basic interpreted implementation of our method, competing with a C implementation of the direct Ewald summation, the break-even point is very low. We believe that this work presents a practical and useful contribution that will help improve the efficiency of viscous flow calculations in various settings.

Recent work by Beale and Strain [18] provides an alternative view of Ewald summation for Stokes, in the context of an elastic interface in 2D periodic Stokes flow. The approach in this paper follows other recent, more general, work by Strain on spectral methods for elliptic systems [19], as well as several error estimates and other useful results from [20]. Their approach appears tractable for problems in 3D, as does obtaining an FFT-based (gridded) summation method, but the authors refrain from pursuing these extensions. None the less, several comparisons will be given throughout this paper, as there are interesting parallels and differences. For instance, as pointed out by e.g. Beale and Strain in [18], gridded Ewald methods have striking similarities to the *non-uniform FFT* (see Dutt and Rokhlin [21], Greengard and Lee [22] and Sammis and Strain [23]) that we shall draw on in this work.

The disposition of this paper is as follows: in Section 2 we present the Ewald-like summation methods for (2) due to Hasimoto, Pozrikidis and Beenakker in a plain manner and discuss the parameters present. Then, in Section 3, we derive an efficient method for evaluating the frequency domain part of these expression, and discuss error, computational complexity, and parameter choices associated with this method. In Section 4 we demonstrate spectral accuracy and $\mathcal{O}(N)$ complexity with a series of numerical experiments. We conclude with considering the full problem, i.e. including the real space sum, and show that we get an $\mathcal{O}(N \log N)$ method when a simple method for treating the real space sum is introduced.

2. Ewald summation for Stokes

In the literature there exists at least three results that apply Ewald's decomposition idea to the Stokeslet and obtain absolutely convergent means of summing (2). In this section we present two of them in a plain and unified framework:

$$\mathbf{u}(\mathbf{x}_m) = \sum_{\mathbf{p}} \sum_{n=1}^{N} A(\xi, \mathbf{x}_m - \mathbf{x}_n + \mathbf{p}) \mathbf{f}_n + \frac{1}{V} \sum_{\mathbf{k} \neq 0} B(\xi, \mathbf{k}) e^{-k^2/4\xi^2} \sum_{n=1}^{N} \mathbf{f}_n e^{-i\mathbf{k}\cdot(\mathbf{x}_m - \mathbf{x}_n)} - \mathbf{u}_{\text{self}}, \tag{3}$$

where the term $(n = m, \mathbf{p} = 0)$ is excluded from the real space sum, $\mathbf{k} \in \{2\pi\kappa_i / L_i : \kappa_i \in \mathbb{Z}, i = 1, 2, 3\}$ (no sum), $k = |\mathbf{k}|$, and $V = L_x L_y L_z$. The positive constant ξ, is known as the Ewald parameter (see Section 2.5). Eq. (3) exposes the Ewald decomposition idea – that the sum over replicated Stokeslets (2) is split into one sum in real space, one sum in frequency domain (henceforth called **k**-space), and a self-contribution.

2.1. Hasimoto decomposition

From the original formulation by Hasimoto [12] we get

$$A(\xi,\mathbf{x}) = 2\left(\frac{\xi e^{-\xi^2 r^2}}{\sqrt{\pi} r^2} + \frac{\operatorname{erfc}(\xi r)}{2r^3}\right)(r^2\mathbf{I} + \mathbf{xx}) - \frac{4\xi}{\sqrt{\pi}} e^{-\xi^2 r^2}\mathbf{I}, \tag{4}$$

with $r = \|x\|$, and

$$B(\xi,\mathbf{k}) = 8\pi\left(1 + \frac{k^2}{4\xi^2}\right)\frac{1}{k^4}(k^2\mathbf{I} - \mathbf{kk}), \tag{5}$$

and the self interaction term is given by $\mathbf{u}_{\text{self}}(\mathbf{x}_m) = \frac{4\xi}{\sqrt{\pi}}\mathbf{f}_m$, where $\operatorname{erfc}(x) := \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}\mathrm{d}t$ is the complementary error function. Note that the Hasimoto sums are usually given on a slightly different form. We have, in order to avoid notational clutter in the rest of this paper, adopted the parametrization of Beenakker (below). The substitution introduced is simply $\xi_* \to \frac{\pi}{\xi^2}$.

In anticipation of later analysis, we now derive radial bounds for A and B. That is, we intend to bound $|A(\xi,\mathbf{x})\mathbf{f}|$ and $|B(\xi,\mathbf{k})\mathbf{f}|$ in terms of r and k respectively, for some $\mathbf{f} \in \mathbb{R}^3$. The following inequality will be used throughout:

$$\operatorname{erfc}(x) \leqslant e^{-x^2}, \quad x \geqslant 0.$$

For the real space sum we find that

$$|A(\xi,\mathbf{x})\mathbf{f}| \leqslant 2\left(\frac{\xi e^{-\xi^2 r^2}}{\sqrt{\pi} r^2} + \frac{\operatorname{erfc}(\xi r)}{2r^3}\right)2r^2\|\mathbf{f}\|_2 + \frac{4\xi}{\sqrt{\pi}} e^{-\xi^2 r^2}\|\mathbf{f}\|_2 \leqslant \left(\frac{8\xi}{\sqrt{\pi}} + \frac{2}{r}\right)e^{-\xi^2 r^2}\|\mathbf{f}\|_2, \tag{6}$$

and for the reciprocal space sum we get

$$|B(\xi,\mathbf{k})\mathbf{f}| \leqslant 8\pi\left(1 + \frac{k^2}{4\xi^2}\right)\frac{1}{k^2}\|\mathbf{f}\|_2. \tag{7}$$

2.2. Beenakker/Pozrikidis decomposition

Beenakker introduced a different method of derivation in [13] which was applied to the Stokeslet by Pozrikidis in [17]. In the setting (3) we state it as

$$A(\xi,\mathbf{x}) = \frac{C_1(\xi r)}{r}\mathbf{I} + \frac{C_2(\xi r)}{r^3}\mathbf{xx} \tag{8}$$

$$C_1(r) = \operatorname{erfc}(r) + \frac{2}{\sqrt{\pi}}(2r^2 - 3)re^{-r^2}$$

$$C_2(r) = \operatorname{erfc}(r) + \frac{2}{\sqrt{\pi}}(1 - 2r^2)re^{-r^2}$$

and

$$B(\xi,\mathbf{k}) = \frac{8\pi}{\xi^4}\left(\frac{1}{\omega^4} + \frac{1}{4\omega^2} + \frac{1}{8}\right)(k^2\mathbf{I} - \mathbf{kk}), \tag{9}$$

where we have introduced $\omega = |\mathbf{k}|/\xi$. Here, the self interaction term is $\mathbf{u}_{\text{self}}(\mathbf{x}_m) = \frac{8\xi}{\sqrt{\pi}}\mathbf{f}_m$.

Again, we determine

$$|A(\xi,\mathbf{x})\mathbf{f}| \leqslant \frac{\|\mathbf{f}\|_2}{r}(|C_1(\xi r)| + |C_2(\xi r)|) \leqslant \left(\frac{2}{r} + \frac{8\xi}{\sqrt{\pi}} + \frac{8r^2\xi^3}{\sqrt{\pi}}\right)e^{-\xi^2 r^2}\|\mathbf{f}\|_2, \tag{10}$$

and

$$|B(\xi,\mathbf{x})\mathbf{f}| \leqslant 8\pi\left(\frac{k^2}{8\xi^4} + \frac{1}{4\xi^2} + \frac{1}{k^2}\right)\|\mathbf{f}\|_2. \tag{11}$$

2.3. Other decompositions

The Beenakker decomposition is based on a decomposition of the Stokeslet as $(I\Delta - \nabla\nabla)[r\operatorname{erfc}(\xi r) + r\operatorname{erf}(\xi r)]$, cf. [17, Section 3]. Pozrikidis [17] notes that erfc is costly to compute, so he suggests using $\exp(-r)$ instead. The resulting decomposition has algebraic decay in reciprocal space. It may be possible to derive an analogous method to the one we give in Section 3, but, due to the lack of Gaussian decay, the calculations may be more involved.

In 2D more options exist, including the use of complex variables and the decomposition derived by Beale and Strain [18]. We refrain from surveying this field.

2.4. Truncation of Ewald sums

The expressions (4)–(9) decay like $\exp(-t^2)$ with respect to the number of image boxes and Fourier modes respectively. Hence, for a given accuracy requirement, the sums in (3) can be truncated. We derive error bounds for these sums below for later use, see Fig. 1. First though, a remark on complexity related to the truncation of Ewald sums in electrostatics.

Cutoff error estimates have been shown for Ewald summation applied to electrostatics by Kolafa and Perram [24] and Strain [20]. In the former work it is shown that, under certain uniformity assumptions on the source distributions, Ewald summation attains $\mathcal{O}(N^{3/2})$ complexity [24, Appendix B].

However, there are many situations where this assumption does not hold – e.g. when sources are clustered, as may be the case in particle simulations and boundary integral methods, especially in complex geometries. Furthermore, we are not aware of any generalization of the Kolafa and Perram result (asymptotic complexity $\mathcal{O}(N^{3/2})$) to Stokes. Hence, we shall maintain the view that evaluating either the real or reciprocal sum in (3) for all $\mathbf{x}_m$ is $\mathcal{O}(N^2)$ hard.

2.4.1. Truncation error bound for $\mathbf{k}$-space sum

The reciprocal space sum is composed of a plane wave $\sum \mathbf{f}_n e^{-i\mathbf{k}\cdot(\mathbf{x}_n-\mathbf{x}_m)}$, multiplied by an amplification factor $B(\xi,\mathbf{k})e^{-k^2/4\xi^2}$. The latter factor captures how the sequence decays as $\mathbf{k}\to\infty$. Take some $\mathbf{f}$ and consider

$$\sum_{\mathbf{k}\neq 0} B(\xi,\mathbf{k})\mathbf{f}e^{-k^2/4\xi^2} = \sum_{|k_i|<k_\infty,\mathbf{k}\neq 0} B(\xi,\mathbf{k})\mathbf{f}e^{-k^2/4\xi^2} + E^F(k_\infty). \tag{12}$$

The objective here is to bound $E^F(k_\infty)$ for each B.

Let $\mathbf{k}'$ denote what remains of $\mathbf{k}$-space after the box $|k_i|<k_\infty$ has been subtracted, i.e. $E^F(k_\infty) := \sum_{\mathbf{k}'} B(\xi,\mathbf{k})\mathbf{f}e^{-k^2/4\xi^2}$. We can now bound E^F by instead summing outside of the sphere inscribed in the box $|k_i|<k_\infty$,

$$|E^F(k_\infty)| \leqslant \sum_{|\mathbf{k}|>k_\infty} |B(\xi,\mathbf{k})\mathbf{f}|e^{-k^2/4\xi^2}. \tag{13}$$

To proceed further we use the bounds on $|B(\xi,\mathbf{k})\mathbf{f}|$ given by (7) and (11), denoted $|B(\xi,\mathbf{k})\mathbf{f}|\leqslant B'(k)$. Knowing that the terms in (13) decay with k we may bound (13) by the corresponding integral, after the substitution $\mathbf{k}=2\pi\bar{\kappa}/L, \kappa_i\in\mathbb{Z}$,

$$\begin{aligned}
|E^F(\kappa_\infty)|_i &\leqslant \int_{|\bar{\kappa}|\geqslant\kappa_\infty} B'(\kappa)e^{-\kappa^2(\pi/\xi L)^2}\,\mathrm{d}\bar{\kappa}\\
&= 4\pi\int_{\kappa_\infty}^{\infty}\kappa^2 B'(\kappa)e^{-\kappa^2(\pi/\xi L)^2}\,\mathrm{d}\kappa\\
&\leqslant \begin{cases} \int_{\kappa_\infty}^{\infty}\frac{8\|\mathbf{f}\|_2}{\xi^2}(\pi^2\kappa^2+\xi^2L^2)e^{-\kappa^2(\pi/\xi L)^2}\,\mathrm{d}\kappa \\ \int_{\kappa_\infty}^{\infty}\frac{8\|\mathbf{f}\|_2}{\xi^4L^2}(2\pi^2\kappa^4+\pi^2\kappa^2\xi^2L^2+\xi^4L^4)e^{-\kappa^2(\pi/\xi L)^2}\,\mathrm{d}\kappa, \end{cases}\\
&= \begin{cases} \frac{6\xi L^3\|\mathbf{f}\|_2}{\sqrt{\pi}}\operatorname{erfc}(\frac{\pi\kappa_\infty}{\xi L}) + 4L^2\|\mathbf{f}\|_2\kappa_\infty e^{-\kappa_\infty^2(\pi/\xi L)^2} \\ \frac{\xi(1+8L^3)\|\mathbf{f}\|_2}{\sqrt{\pi}}\operatorname{erfc}(\frac{\pi\kappa_\infty}{\xi L}) + \|\mathbf{f}\|_2(16L^2\kappa_\infty+\frac{8\pi^2\kappa_\infty^3}{\xi^2})e^{-\kappa_\infty^2(\pi/\xi L)^2}, \end{cases}
\end{aligned}$$

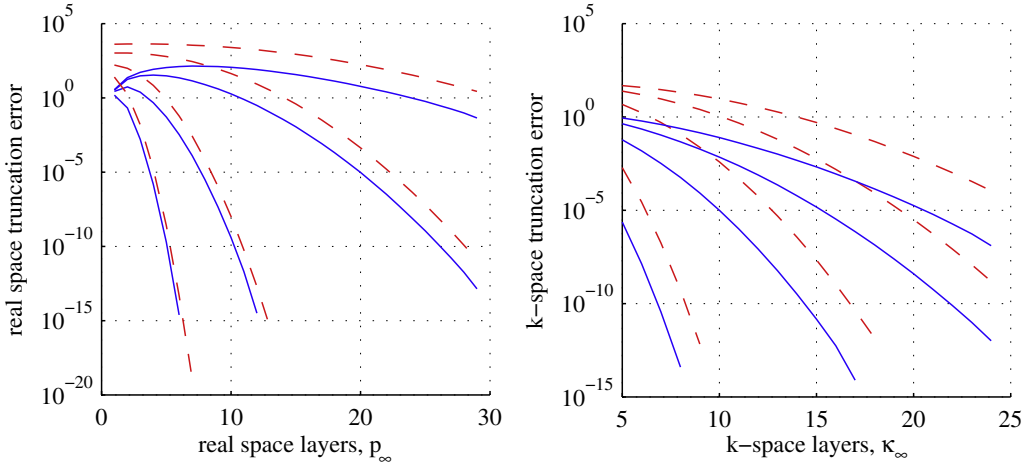

Fig. 1. Left: computed real space truncation error (15), and bound (16) for a sequence $\xi = 1/10, 1/5, 1/2, 1$. Right: computed $\mathbf{k}$-space truncation error (12), and bound (14) for a sequence $\xi = 5, 10, 15, 20$. In both figures: Hasimoto decomposition, and truncation bound as dashed line. The ξ-values are from the challenging regime for each sum respectively, and we have taken $\|\mathbf{f}\| = 1$ and $N = 1$.

$$\leqslant \begin{cases} \frac{2L^2\|\mathbf{f}\|_2}{\sqrt{\pi}}(2\sqrt{\pi}\kappa_\infty + 3\xi L)e^{-\kappa_\infty^2(\pi/\xi L)^2}, & \text{Hasimoto} \\ \frac{\|\mathbf{f}\|_2}{\xi^2\sqrt{\pi}}(8\xi^3L^3 + \xi^3 + 16\sqrt{\pi}\xi^2L^2\kappa_\infty + 8\pi^{5/2}\kappa_\infty^3)e^{-\kappa_\infty^2(\pi/\xi L)^2}. & \text{Beenakker} \end{cases} \tag{14}$$

2.4.2. Truncation error bound for real space sum

To bound the real space sum we proceed in a similar manner. Define $E^R(p_\infty)$ via

$$\sum_{\mathbf{p}} A(\xi,\mathbf{p})\mathbf{f} = \sum_{|p_i|<p_\infty} A(\xi,\mathbf{p})\mathbf{f} + E^R(p_\infty), \tag{15}$$

and sum outside of the sphere $|\mathbf{p}| > p_\infty$,

$$|E^R(p_\infty)| \leqslant \sum_{|\mathbf{p}|>p_\infty} |A(\xi,\mathbf{p})\mathbf{f}|.$$

Noting that the upper bounds on $|A(\xi,\mathbf{p})\mathbf{f}|$, from (6) and (10), and denoted $|A(\xi,\mathbf{p})\mathbf{f}| \leqslant A'(r)$, are decaying in r, we integrate as before,

$$\begin{aligned} |E^R(p_\infty)| &\leqslant \int_{|\mathbf{p}|>p_\infty} A'(r)\mathrm{d}\mathbf{p} = 4\pi\int_{p_\infty}^{\infty} r^2A'(r)\mathrm{d}r \\ &= \begin{cases} 4\pi\|\mathbf{f}\|_2\int_{p_\infty}^{\infty} r^2\left(\frac{8\xi}{\sqrt{\pi}} + \frac{2}{r}\right)e^{-\xi^2r^2}\mathrm{d}r \\ 4\pi\|\mathbf{f}\|_2\int_{p_\infty}^{\infty} r^2\left(\frac{8\xi}{\sqrt{\pi}} + \frac{2}{r} + \frac{8\xi^3r^2}{\sqrt{\pi}}\right)e^{-\xi^2r^2}\mathrm{d}r, \end{cases} \\ &= \begin{cases} \frac{4\|\mathbf{f}\|_2}{\xi^2}\left(2\pi\mathrm{erfc}(\xi p_\infty) + (\pi + 4\sqrt{\pi}p_\infty\xi)e^{-p_\infty^2\xi^2}\right) \\ \frac{20\pi\|\mathbf{f}\|_2}{\xi^2}\mathrm{erfc}(\xi p_\infty) + \|\mathbf{f}\|_2(\frac{4\pi+4\xi\sqrt{\pi}p_\infty}{\xi^2} + 16\xi\sqrt{\pi}p_\infty^3)e^{-p_\infty^2\xi^2}, \end{cases} \\ &\leqslant \begin{cases} \frac{\|\mathbf{f}\|_2}{\xi^2}(12\pi + 16\sqrt{\pi}p_\infty\xi)e^{-p_\infty^2\xi^2}, & \text{Hasimoto} \\ \frac{8\sqrt{\pi}\|\mathbf{f}\|_2}{\xi^2}(2\xi^3p_\infty^3 + 5\xi p_\infty + 3\sqrt{\pi})e^{-p_\infty^2\xi^2}. & \text{Beenakker} \end{cases} \end{aligned} \tag{16}$$

These bounds will be used in Section 5.

2.5. Ewald parameter

The parameter ξ plays the role of partitioning the sum (3) between real and reciprocal space. Numerically, the choice is significant because it lets us balance the computational work between the two sums. That is, *whilst the numerical result is independent of ξ, the total computational time to obtain this value has a minimum for some ξ*. In the table below we illustrate this by showing how many layers (i.e. the truncation in real and reciprocal space, respectively) that are necessary to get both Ewald sums to converge to within a tolerance of 10^{-15}, in a unit cube (see Table A.1 for a more comprehensive comparison). Here we use a small number of particles at randomly distributed positions, $N = 10$, and evaluate the sum at each one. The timing results were obtained on a Intel (Core2 Duo) laptop computer with a code that was written in C.

The main point is that we can make the real space sum cheap at the expense of the **k**-space sum. We will present an efficient method that changes the computational complexity of the **k**-space sum, thus favoring putting less work into the real space sum.

With the two decompositions available, we can comment on their relative merit. We give these computational results as Appendix A, see Fig. A.10 and Table A.1. As for accuracy as a function of computational time, the Hasimoto decomposition has an edge over the Beenakker/Pozrikidis decomposition – most markedly in the real space sum.

Saintillan et. al. [14] suggest that we may view the real space sum as being of $\mathcal{O}(N)$ complexity: For a sufficiently large ξ, the real space interactions are, to some desired accuracy, only between a small number (independent of N) of neighboring particles. However, this argument only holds if the density of sources is kept constant – i.e. by making the box bigger as N grows. In Section 5 we elaborate on this.

3. Fast k-space summation method

As seen in the previous section, we may make the real-space sum cheap to compute by a suitable choice of ξ. This motivates us to search for an efficient method for computing the **k**-space sum. We wish to leverage the well known efficiency of the FFT. For this we need to find a representation of the arbitrarily placed source points as a discrete function on a uniform grid. This is achieved by constructing a convolution between the point sources (represented as δ-functions) and suitably scaled Gaussians.

The frequency domain part of both summation methods for (3) from the previous section reads

$$\mathbf{u}^F(\mathbf{x}_m) = \frac{1}{V}\sum_{\mathbf{k}\neq 0} B(\xi,\mathbf{k}) e^{-k^2/4\xi^2} \sum_n \mathbf{f}_n e^{-i\mathbf{k}\cdot(\mathbf{x}_m-\mathbf{x}_n)}. \tag{17}$$

Below, we derive a gridded summation method for (17) that can be seen as a member of the particle mesh Ewald family. We start by introducing a parameter η to split the Gaussian term in (17),

$$\mathbf{u}^F(\mathbf{x}_m) = \frac{1}{V}\sum_{\mathbf{k}\neq 0} B(\xi,\mathbf{k}) e^{-k^2/4\xi^2} e^{-i\mathbf{k}\cdot\mathbf{x}_m} \sum_n \mathbf{f}_n e^{i\mathbf{k}\cdot\mathbf{x}_n} = \frac{1}{V}\sum_{\mathbf{k}\neq 0} B(\xi,\mathbf{k}) e^{-(1-\eta)k^2/4\xi^2} e^{-i\mathbf{k}\cdot\mathbf{x}_m} \sum_n \mathbf{f}_n e^{-\eta k^2/4\xi^2} e^{i\mathbf{k}\cdot\mathbf{x}_n}. \tag{18}$$

Define

$$\widehat{H}_k := \sum_n \mathbf{f}_n e^{-\eta k^2/8\xi^2} e^{-i\mathbf{k}\cdot\mathbf{x}_n}, \quad \text{s.t.}$$

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

and note that half of the term $e^{-\eta k^2/4\xi^2}$ from (18) went into $\widehat{H}_k$, leaving half the Gaussian for later use.

The expression for $\widehat{H}_k$ is in the form of a product in Fourier space, $\widehat{H}_k = \sum_n \mathbf{f}_n \widehat{v}_k \widehat{w}_k$, with

$$\widehat{v}_k = e^{-\eta k^2/8\xi^2}$$

$$\widehat{w}_k = e^{-i\mathbf{k}\cdot\mathbf{x}_n}.$$

A product in reciprocal space is equivalent to a convolution in real space. This fact was utilized also by Deserno and Holm [5] when analyzing gridded Ewald methods for the Coulomb potential. Here, the inverse transforms are available:

$$v = \left(\frac{2\xi^2}{\pi\eta}\right)^{3/2} e^{-2\xi^2 r^2/\eta}$$

$$w = \delta(\mathbf{x}-\mathbf{x}_n).$$

This implies that $H(\mathbf{x})$ is given by the following convolution:

$$H(\mathbf{x}) = \sum_n \mathbf{f}_n \int_\Omega \delta(\mathbf{y}-\mathbf{x}_n)\left(\frac{2\xi^2}{\pi\eta}\right)^{3/2} e^{-2\xi^2|\mathbf{y}-\mathbf{x}|_*^2/\eta} d\mathbf{y} = \sum_n \mathbf{f}_n \left(\frac{2\xi^2}{\pi\eta}\right)^{3/2} e^{-2\xi^2|\mathbf{x}-\mathbf{x}_n|_*^2/\eta}, \tag{19}$$

where $|\cdot|^*$ denotes distance to closest periodic image, and $\Omega \subseteq \mathbb{R}^3$ denotes the domain. This expression provides a representation of the source particles as a smooth function – in essence a superposition of appropriately scaled Gaussians. Further, define

$$\widehat{\widetilde{H}}_k := B(\xi,\mathbf{k}) e^{-(1-\eta)k^2/4\xi^2} \widehat{H}_k, \tag{20}$$

so that

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

For periodic functions we have that $\sum_{\mathbf{k}} \hat{f}_k \hat{g}_{-k} = V\int_\Omega f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}$. Identifying $g = \widehat{\widetilde{H}}$ and $f = e^{-\eta k^2/8\xi^2} e^{-i\mathbf{k}\cdot x_m}$ above gives, via the convolution/multiplication argument again, that

$$\mathbf{u}^F(\mathbf{x}_m) = \int_\Omega \widetilde{H}(\mathbf{x}) \left[\int_\Omega \delta(\mathbf{y}-\mathbf{x}_m)\left(\frac{2\xi^2}{\pi\eta}\right)^{3/2} e^{-2\xi^2|\mathbf{y}-\mathbf{x}|_*^2/\eta} d\mathbf{y}\right] d\mathbf{x} = \int_\Omega \widetilde{H}(\mathbf{x})\left(\frac{2\xi^2}{\pi\eta}\right)^{3/2} e^{-2\xi^2|\mathbf{x}-\mathbf{x}_m|_*^2/\eta} d\mathbf{x}, \tag{21}$$

where $\widetilde{H}(\mathbf{x})$ is the inverse transform of $\widehat{\widetilde{H}}$.

Expressions (18)–(20) provide the foundation for a fast method to compute the frequency domain part of the Ewald sum (3). The steps are as follows:

1. Pick ξ such that the real space sum converges rapidly. Cf. Section 5.
2. Construct a uniform grid over the rectangular domain with sides $[L_x, L_y, L_z]$, and evaluate $H(\mathbf{x})$ on this grid using (19). A suitable choice for η is discussed in Sections 3.1 and 3.2.
3. Since the grid is equidistant, $\widehat{H}$ can be efficiently computed via a FFT in three dimensions.
4. Apply the scaling (20) to get $\widehat{\widetilde{H}}$.
5. An inverse FFT yields $\widetilde{H}(\mathbf{x})$.
6. Evaluate the integral (21) with trapezoidal quadrature. This is spectrally accurate since the integrand is periodic (see Section 3.2).

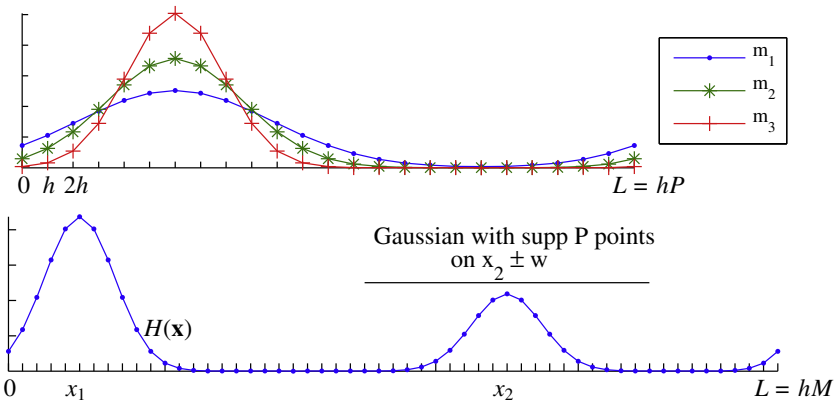

Fig. 2. Top: parametrization of Gaussian with support on entire domain, P points, for a sequence $m_1 < m_2 < m_3$. Bottom: Gaussians with support on $x_i \pm w$, and P points on this interval.

Remark 1. The fundamental benefit with this procedure is that once $H(\mathbf{x})$ has been computed, the FFT simultaneously yields $\widehat{H}_k$ *for all* $\mathbf{k}$. The same mechanism works to our advantage in the inversion step (IFFT + Eq. (21)).

The method presented by Saintillan et. al. [14] is similar in construction. They use Cardinal B-splines to construct a grid function, whereas our convolution approach avoids approximation entirely. Accuracy aside, this seems to provide a fairly large simplification for us, in terms of derivation, presentation and, most notably, implementation. Sierou et. al. [15] uses a Taylor expansion in the corresponding step, with obvious concerns for high accuracy. We hope that the numerical results presented later in this paper lend additional merit to our approach.

3.1. Discretization and scaling

We now discuss the discretization of our proposed method from the previous section. There is a free parameter, η, that can be used to control the width, denoted w, of the Gaussian used for the convolutions in (19) and (21), see Fig. 2. It is natural to think in terms of the bell curve from basic statistics, $e^{-t^2/2\sigma^2}$, where σ (the standard deviation) controls the decay. If we compare to the exponential in (19) we find that in terms of our parameters we can express σ as $\sigma^2 = \eta/4\xi^2$. Now we let w correspond to m standard deviations, i.e. $m\sigma = w$. Eliminate σ and we get

$$\eta = \left(\frac{2w\xi}{m}\right)^2. \tag{22}$$

The purpose of this was simply to frame the choice of the parameter η in an intuitive setting. Aside from the half-width, w, we can control how quickly the Gaussians decay in the interval $\pm w$ with the parameter m, see Fig. 2 (top).

We shall let the domain $\Omega = [0, L)^3$ be discretized with M points in each direction and let $h = L/M$ denote the grid size. It is implied throughout that L and M can be different in each direction, and subscripts will be used when necessary, i.e. L_i and M_i. Finally, we let $P \leqslant M$ denote the number of grid points within the support of each Gaussian, as seen in Fig. 2 (bottom).

With the choice of η given by (22), and with $w = Ph/2$ the Gaussians present in (19) and (21) take the form

$$e^{-2\xi^2 r^2/\eta} = e^{-2r^2(m/Ph)^2} \tag{23}$$

which evaluates to $e^{-m^2/2}$ at the truncation point $r = w$.

3.2. Error analysis

We claim that our method for computing (17) is accurate to exponential-squared order. More specifically, that all errors incurred from truncation and discretization decay as $\mathcal{O}(e^{-\alpha M^2})$, for some $\alpha > 0$, as the grid is refined (i.e. M grows). There are two components of the error present: First, the uniform discretization of the box implies that there will be a maximal frequency that is resolved, i.e. that the frequency domain is truncated. Secondly, there will be a numerical error in evaluating the integral (21) that is closely tied to how well the Gaussian basis functions are resolved on the grid.

3.2.1. ***k*-Space truncation error**

First we consider the immediate error due to the **k**-space truncation, which is carried over from the sum (17). Any direct calculation will have to truncate this sum at some radius k_∞. If we in our method use a M^3 grid, this corresponds to a **k**-space truncation at $k_\infty = M\pi/L$, so we can view the truncation error as $\approx Ce^{-M^2\pi^2/4L\xi^2}$.

The bounds (14) given in Section 2.4.1 support the conclusion of e^{-k^2} convergence for both decompositions. However, from a practical point of view (see Section 5), the constant is not as tight as desired. Furthermore, for some prescribed tol-

erance ε, one would like to solve $|R(k_\infty)| < \varepsilon$ for k_∞. This is not immediately forthcoming from (14). Instead, when we treat the full problem in Section 5, we will instead construct an estimate $|R(k_\infty)| \approx Ce^{-k_\infty^2/4\xi^2}$.

3.2.2. Quadrature error

Consider now the second error, stemming from the discretization error in (21), which is the contribution to the total error from our method. We give a 1D error bound for $\widehat{H}(\mathbf{x}) \equiv 1$, which is sensible due to the regularity of $\widehat{H}$. Further, a 1D bound will arrive at the correct asymptotic error estimate for the 3D integral.

Introduce $C = (2\xi^2/\pi\eta)^{1/2}$ and $\alpha = 2\xi^2/\eta$, so that with (21) in mind we consider $1 = \int_0^L Ce^{-\alpha(x-L/2)^2}\mathrm{d}x$. The objective is to bound the error due to the quadrature $T_P := \sum_{n=-P/2}^{P/2} Ce^{-\alpha(hn)^2}$, $h := 2w/P$,

$$E^q := |T_P - 1| = |h\sum_{n=-\infty}^{\infty} Ce^{-\alpha(hn)^2} - 1 - h\sum_{|n|>P/2} Ce^{-\alpha(hn)^2}| \leqslant |h\sum_{n=-\infty}^{\infty} Ce^{-\alpha(hn)^2} - 1| + |h\sum_{|n|>P/2} Ce^{-\alpha(hn)^2}|. \tag{24}$$

To bound the first term we will use the Poisson summation formula, $h\sum_{n=-\infty}^{\infty} f(hn) = \sum_{\kappa=-\infty}^{\infty} \hat{f}(\kappa/h)$, with $f(x) = Ce^{-\alpha x^2}$ and, hence, $\hat{f}(\kappa) = e^{-\pi^2\eta\kappa^2/(2\xi^2L^2)}$. This gives

$$h\sum_{n=-\infty}^{\infty} Ce^{-\alpha(hn)^2} - 1 = \sum_{\kappa=-\infty}^{\infty} e^{-\pi^2\eta\kappa^2/(2h^2\xi^2L^2)} - 1 = \sum_{\kappa\neq 0} e^{-\pi^2\eta\kappa^2/(2h^2\xi^2L^2)} \leqslant 4e^{-\pi^2\eta/(2h^2\xi^2L^2)} = 4e^{-\pi^2P^2/(2m^2L^2)},$$

where we have inserted (22).

The second term in (24) is bounded by the complementary error function in a straight-forward fashion,

$$h\sum_{|n|>P/2} Ce^{-\alpha(hn)^2} \leqslant 2hC\int_{P/2}^{\infty} e^{-\alpha(ht)^2}\mathrm{d}t = \mathrm{erfc}\left(\xi L\sqrt{1/2\eta}\right) = \mathrm{erfc}(m/\sqrt{2}).$$

Putting these results together we have the quadrature error bound

$$E^q \leqslant 4e^{-\pi^2P^2/(2m^2L^2)} + \mathrm{erfc}(m/\sqrt{2}). \tag{25}$$

The number of grid points within the support of each Gaussian, P, is present, but the shape parameter, m, also plays a role. By making m small the Gaussian is well resolved, and the first term is small. But then this implies that a tail of the Gaussian has been truncated, as evident in the second term.

3.2.3. Remarks on accuracy

There is an important mathematical consequence of the choice of η given by (22), namely that *we decouple the two error contributions from the previous sections*. With the truncation error bounds (14) in mind, we may think of the **k**-space truncation error associated with some grid size M as $\approx Ce^{-M^2\pi^2/4L\xi^2}$. This term grows with ξ, demanding a larger M to reach a desired tolerance. On the other hand, we have already seen that the quadrature error estimate is independent of ξ.

If ξ is sufficiently small the quadrature error will dominate, meaning that we benefit from taking $P = M$ (i.e. $w = L/2$). In Section 5, where we include the real space sum, ξ will be large enough that the truncation error will demand a larger FFT-grid, M, than the resolution of the Gaussians, P. With the parametrization (22) we have the flexibility to adjust the width of the Gaussians in the to-grid and from-grid operations such that the desired accuracy is met while still not over-resolving them which would lead to an increased computational cost.

In related work we have not found an error estimate of this order. The GSE method by Shan et. al. [6] for the electrostatic case ought to be spectrally accurate as well. However, they provide only a numerical evaluation, for the full problem (i.e. including the real space sum) in a parameter setting specialized for MD simulations which unfortunately only presents a narrow view of the error from the GSE method itself. In other meshed Ewald methods for Coulombic potentials we find that two types of interpolation schemes are used: Lagrangian (in e.g. [3]), and pth order splines (e.g. [4]). The spline methods are C^{p-2} smooth and the Lagrange methods are less so. Our approach is fundamentally different. Instead of interpolation we use the fact that a multiplication in Fourier space is equivalent to a convolution in real space. The Gaussians used for the convolution to define our grid functions are C^∞ smooth and have analytical continuation. As a direct consequence of this we obtain the exponential-squared convergence, which must be regarded as very fast. In the work by Saintillan et. al. [14], which also treats the Stokes potential, an interpolation scheme based on Cardinal B-splines is used. They report convergence of algebraic kind, K^p, where they use interpolation of order from $p = 1$ to 9 in demonstrations. It is unclear what interpolation order they use in practice. In their framework, known as *Stokesian Dynamics*, Sierou and Brady [15] use Lagrangian polynomial interpolation, with the aforementioned limited smoothness.

3.3. Complexity

The contributions to the computational cost of the proposed algorithm are

$$\underbrace{\mathcal{O}(NP^3)}_{\text{Eq.(19)}} + \underbrace{\mathcal{O}(M^3\log M^3)}_{\text{FFT+IFFT}} + \underbrace{\mathcal{O}(M^3)}_{\text{Eq.(20)}} + \underbrace{\mathcal{O}(NP^3)}_{\text{Eq.(21) }\forall m}.$$

However, we regard P as an accuracy parameter, independent of the mesh and number of source points. So the complexity of our proposed algorithm should be stated as

$$\mathcal{O}(N) + \mathcal{O}(M^3 \log M^3) + \mathcal{O}(M^3) + \mathcal{O}(N)$$

This is in sharp contrast to the $\mathcal{O}(N^2\kappa_\infty^3)$ amount of computation to evaluate (17) for all m, where k_∞ denotes the cutoff in frequency domain.

The tasks of evaluating (19) and (21) are equivalent to the so called Gaussian gridding step in non-uniform FFTs [21], which it is known to dominate the computational cost of that algorithm. To alleviate this cost Greengard and Lee [22] proposed a fast Gaussian gridding method, which reduces the arithmetic involved by a significant factor. This technique should be of great benefit to our method, see details given in Appendix B.

Our method, as is natural for all gridded summation methods, requires the full grid to be stored, so the complexity in memory could at first glance be a concern. It is therefore natural to ask what the relationship between accuracy and the size of grid required is for a range of ξ. In Section 4.1 we demonstrate that full accuracy is attained at inexpensive discretizations, and in Section 4.2 we investigate the break-even point in terms of the number of particles, N.

In related work, as previously discussed, a variety of interpolation schemes are present. These methods all have some footprint on the grid which may be compared to the P^3 points that our Gaussians are defined on. In particular the pth order Cardinal B-splines, presented in [5, Appendix E] have support on p^3 grid points. These are designed to be as compact as possible, and it is clear that our method is less compact – how much will be discussed in the next section.

4. Numerical results

Now we substantiate our analysis from the preceding sections about spectral accuracy and computational complexity. When we investigate the errors we compute ∞-norm as a relative measure and not, as appropriate. That is,

$$e_\infty := \max_{m\in[1\ldots N]} \|\mathbf{u}^F(\mathbf{x}_m) - \mathbf{u}^*(\mathbf{x}_m)\|_2, \quad e_\infty^{\text{rel}} := \max_{m\in[1\ldots N]} \frac{\|\mathbf{u}^F(\mathbf{x}_m) - \mathbf{u}^*(\mathbf{x}_m)\|_2}{\|\mathbf{u}^*(\mathbf{x}_m)\|_2}, \tag{26}$$

where $\mathbf{u}^*$ is a well converged value using the direct summation method, as discussed in Section 2.5. These measures are very costly to compute, limiting us to somewhat small N.

For clarity, we shall in this section omit the real space sum – focusing instead on our proposed method versus the Ewald sum in $\mathbf{k}$-space, (17). See Section 5 for the full problem.

4.1. Accuracy in $\mathbf{k}$-space

We have seen in the preceding error analysis that approximation errors arising from our method (25) do not depend on ξ, but the $\mathbf{k}$-space truncation error (cf. Section 3.2.1) does. From that point of view the $\mathbf{k}$-space truncation error becomes the primary concern for a given ξ. Numerical truncation results for the Ewald sums were given in Fig. 1, and in Section 5 we illustrate how the truncation error estimates can, for a desired accuracy ε, imply a suitable ξ which then implies an FFT-grid, M.

Here we shall focus on the approximation errors, i.e. how to choose the number of points within the support of each Gaussian, P, and the shape parameter, m. We take positions $\mathbf{x}_j$ from a uniform random distribution in $(0\ \ 1)^3$, and force values, $\mathbf{f}_j$, randomized but adjusted such that $\sum \|\mathbf{f}_n\|_2 = 1$.

For small values of ξ, the $\mathbf{k}$-space truncation error is very small, and the approximation errors dominate. In Fig. 3 we see that the errors in ∞-norm (26) follow the quadrature error estimate, with $\xi \in [1,4]$, $P = M$, $w = 1/2$, and $m = 8$. Both decompositions work equally well. Here the error was measured in a relative sense (since the magnitude of $\mathbf{u}^F$ varies with ξ), whereas the error estimate $E^q = 4e^{-M^2\pi^2/(2m^2)}$, from (25), is designed as an absolute measure.

Taking $\xi = 6$ we find that a grid of size $M = 31$ is needed to make the $\mathbf{k}$-space truncation errors machine precision small. Then, with $w = hP/2$, we obtain the errors given in Fig. 4 (left), as P and m vary. For a particular m, the same Gaussian convergence as in Fig. 3 is seen, until the error due to the truncated Gaussian is reached. We repeat this with $\xi = 10$ and $M = 61$, also in Fig. 4 (right). The error estimate $E^q = 4e^{-P^2\pi^2/(2m^2)} + \operatorname{erfc}(m/\sqrt{2})$ (which is a guiding estimate, not an upper bound) is actually seen to predict the error very well, breaking down only in the region where round-off errors enter the computation.

For non-unit domains equivalent results are easily attained, as long as all directions are equally well refined. That is, with a domain $\Omega = [0,L) \times [0,\beta L) \times [0,\gamma L)$ and a grid $M \times \beta M \times \gamma M$, the expected spectral accuracy is observed as P grows.

Roughly speaking, our method is useful in the range $P = 7$ (accuracy $\sim 10^{-4}$) to $P = 25$ (accuracy $\sim 10^{-15}$). That is, the Gaussians have support on between 7^3 and 25^3 grid points. This is in line with guidelines for the non-uniform FFT: 12 points per Gaussian for single precision and 24 for double. The size of the FFT-grid, M, is most often determined by the Ewald parameter, ξ (unless ξ is small).

As a concluding remark, the parameter η may in principle be larger than one. A loss of accuracy may then occur in (20) due to a positive exponent producing a numerical overflow. However, with the parametrization given by (22), and given the accuracy considerations discussed above, this does not happen for practical choices of parameters.

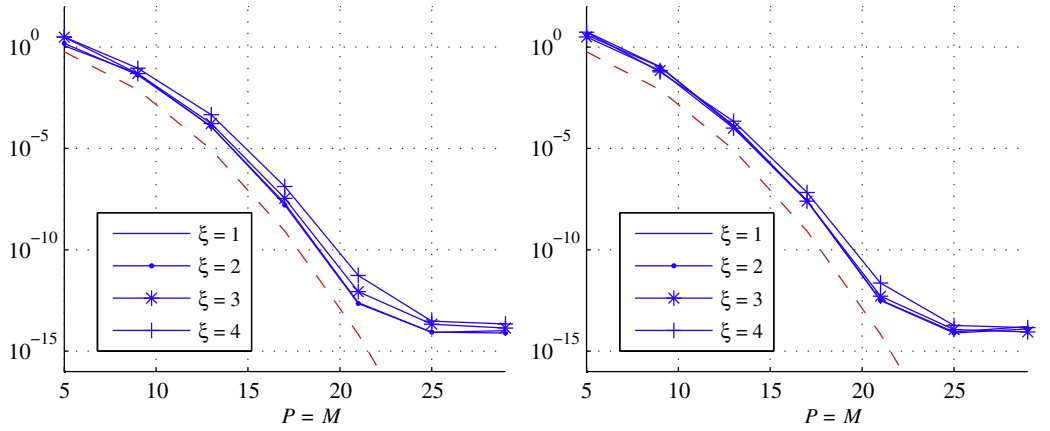

Fig. 3. Relative error, e_∞^{rel}, of FFT-based algorithm for (17) as a function of box resolution, for unit cube box with $N = 100$ particles, $P = M$ and $m = 8$. Ewald parameter, $\xi = 1, 2, 3, 4$. Left: Beenakker decomposition. Right: Hasimoto decomposition. The dashed line in each plot is the error estimate (25).

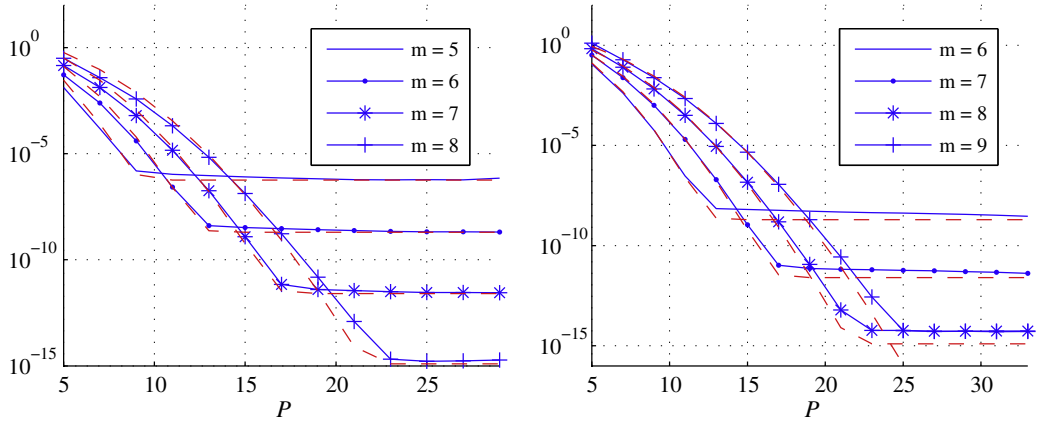

Fig. 4. Error of FFT-based algorithm (26), for (17) as a function of Gaussian resolution, P, for unit cube box with $N = 100$ particles and $\sum \|\mathbf{f}_n\|_2 = 1$. The dashed line in each plot is the error estimate (25) in terms of P. Hasimoto decomposition used. Left: Ewald parameter $\xi = 6$, and grid $M = 31$. Right: $\xi = 10$, $M = 61$.

4.2. Computational complexity

In the preceding section we saw that our method gives very accurate results with very small computational grids. This is a desirable trait of our method, because the transforms will be dominated in run-time by the operations (19) and (21), that scale linearly with N. In effect, we have reduced the cost of computing (17) from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$, where the constant in the former case depends on the **k**-space truncation of Eq. (17), and the constant in the latter case depends on the width of the support of the Gaussians (denoted P).

As previously noted, in related work compact interpolation schemes are used for constructing a grid function that represents the particle configuration. This corresponds to Eq. (19) in our method, and it is evident that in our method each source point will have support on a larger part of the grid. To illustrate this we show timing results in Fig. 5, where the number of source points, N, grows. Here we use a fixed grid of size $M = 31$, and successively truncate the Gaussian, as discussed in Sections 3.1 and 3.3, at smaller P. So whereas other methods have narrow interpolation schemes, e.g. the SPME method of Saintillan et. al. [14] needs 9^3 points for their Cardinal B-splines, our method is useful in the range 7^3 to 25^3 points.

We must note, however, that there is no expensive arithmetic involved in evaluating (19) and (21) – computing $\exp(x)$ is done in less than 15 clock cycles on current commodity hardware – whereas computing and evaluating e.g. high order B-splines may require more involved arithmetic per point where it has support. Additionally, the fast Gaussian gridding method, due to Greengard and Lee [22], would immediately speed up these calculations by a significant factor (see Appendix B).

4.2.1. Break-even

There are several reasons why it is hard to obtain meaningful break-even results when proposing a new algorithm. Implementations can be vastly different in efficiency; parameters need to be chosen fairly; algorithms may have subtly different

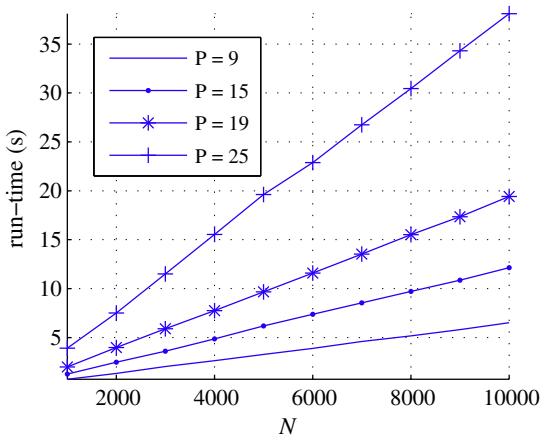

Fig. 5. Run-time of FFT-based method versus system size, N, with grid size $M = 31$, where we vary the number of points within the support of each Gaussian.

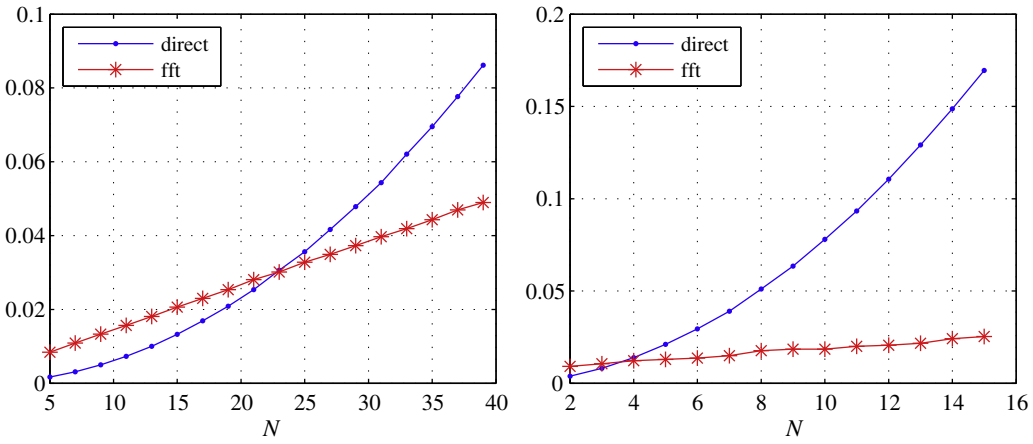

Fig. 6. Run time (seconds) vs. number of points. Left: $\xi = 5$ and the **k**-space truncation of (17) at $\kappa_\infty = 4$. Left: $\xi = 10$, $\kappa_\infty = 10$.

intended use (low or high accuracy for example). The results given in this section need to be seen in this light. For instance, the direct evaluation of the **k**-space Ewald sum (17) was implemented in C, whereas the FFT-based method is implemented mostly in Matlab.

When comparing these methods we seek equal accuracy. This implies that we will take the number of layers in **k**-space to be half the FFT-grid size, $\kappa_\infty = M/2$, assuming truncation errors dominate. From the truncation results in Fig. 1 we find that for $\xi = 5{,}10$ one would take $\kappa_\infty = 4$, 10 layers to obtain a accuracy of roughly 10^{-10}. From the approximation error given in Fig. 4, $P = 15$, $m = 7$ is appropriate for equal accuracy, and, hence, $M = 15$, 20. Timing results with these parameters, for a range of N, are given in Fig. 6. These suggest a very low break-even point – below $N = 25$.

5. Full problem

Up to this point we have mainly been concerned with analyzing our proposed method as a drop-in replacement to the **k**-space sum in (3). We shall now add the contribution from the real space sum and discuss what this implies for the parameters and the overall complexity of the periodic summation problem with our method applied to the **k**-space sum.

5.1. Linear complexity for real space sum

As previously noted, it has been suggested that the real space sum,

$$\mathbf{u}^R(\mathbf{x}_m) = \sum_{\mathbf{p}} \sum_{n=1}^{N} A(\xi, \mathbf{x}_m - \mathbf{x}_n + \mathbf{p})\mathbf{f}_n, \tag{27}$$

can be viewed as of $\mathcal{O}(N)$ complexity if a fixed interaction radius r_c is maintained as N, the number of source points, grows. It is, however, crucial to this argument that the density of particles is kept constant, i.e. that the box grows with N, and that these sources are evenly distributed. The complexity of the real space sum will then be $\mathcal{O}(N)$ since each particle will interact with a fixed number of it's neighbors. Before we proceed, it should be noted that these may not be suitable assumptions in all settings where our method is applicable.

In order to benefit from a fixed interaction radius an interaction list of near neighbors must be constructed and maintained. How to construct and maintain such a neighbor list is well studied, e.g. in the field of computational molecular dynamics (MD). We refer the reader to the textbook by Allen and Tildesley [25] for an overview. Here we present a numerical experiment where we can use a simple strategy to get the real space sum to scale linearly with N.

Start with N_0 particles in a domain of size L_0^3, and split this into $q_0 \times q_0 \times q_0$, boxes of equal size, with $q_0 \geqslant 3$. The Ewald parameter, ξ, is chosen such that the sum is converged, to within a prescribed tolerance, at a radius $\frac{L_0}{q_0}$. For each particle we sum over all the particles in its own box and in all adjacent boxes (with periodicity), i.e. $\sim 27p$ interactions where $p = N_0/q^3$ is the average number of points in each box. Now we grow the domain by adding boxes, i.e. $q_1 = q_0 + 1$, $L_1 = L_0 + L_0/q_0$, and $N_1 = L_1^3 N_0/L_0^3$. This approach to the real space sum is also used by Sierou and Brady [15].

5.2. Truncation error analysis – determining parameters

There remains to determine a value for ξ, and an FFT-grid M, in terms of a prescribed tolerance ε. We impose that the truncation error arising from the real space (27) and $\mathbf{k}$-space (17) sums be approximately equal, i.e. that $E^R \approx E^F < \varepsilon$.

Neither of the truncation error estimates, (16) and (14), provide an elementary expression for the desired parameters. To simplify the situation we shall use error estimates

$$E^R \approx C_R e^{-r^2\xi^2}$$
$$E^F \approx C_F e^{-k^2/4\xi^2}$$

and numerically find suitable constants C_R and C_F. To make a fair comparison as the size of the system is scaled up we let $\sum_{n=1}^{N} \|\mathbf{f}_n\|_2 \equiv 1$. This makes the estimates above hold independently of N. With the requirement that the real space sum be converged at a radius $r = L/q$, we obtain

$$\xi > \frac{q}{L}\sqrt{\log(1/\varepsilon) + \log(C_R)}, \tag{28}$$

from the real space truncation estimate. The FFT-grid, M, is related to the $\mathbf{k}$-vector via $M = 2\kappa + 1$, where $k =: 2\pi\kappa/L$. Hence,

$$k > 2\xi\sqrt{\log(1/\varepsilon) + \log(C_F)},$$

and

$$M > \frac{2\xi L}{\pi}\sqrt{\log(1/\varepsilon) + \log(C_F)} + 1. \tag{29}$$

In Fig. 7 we give computational results that suggest sensible values, $C_R = 1/10$ and $C_F = 1$. For a range of tolerances we then get the following parameter values

$\varepsilon =$	10^{-06}	10^{-08}	10^{-10}	10^{-12}
$\xi =$	10.1	12.0	13.6	15.0
$M =$	27	35	43	53

where we have taken $L = 1$ and $q = 3$.

To verify the error estimate we give a brief numerical example. Take $\mathbf{x}_i$ from a uniform random distribution in $[0,1]^3$ and compute the error between the real sum computed as outlined in the previous section and a well converged direct computation of (27), denoted $\mathbf{u}^{*R}$,

$$e_2 := \frac{1}{N}\sum_n \|\mathbf{u}_n^R - \mathbf{u}_n^{*R}\|_2.$$

This error is given in the following table for various $\xi(\varepsilon)$ and N (which the error is evidently independent of):

$\xi =$	10.1	12.0	13.6	15.0
$N = 50$	6.9×10^{-10}	6.3×10^{-13}	7.8×10^{-16}	6.6×10^{-18}
$N = 100$	1.2×10^{-08}	4.5×10^{-11}	2.3×10^{-13}	1.4×10^{-15}
$N = 200$	1.0×10^{-08}	4.1×10^{-11}	2.1×10^{-13}	1.3×10^{-15}
$N = 1000$	1.1×10^{-08}	4.7×10^{-11}	2.8×10^{-13}	2.0×10^{-15}

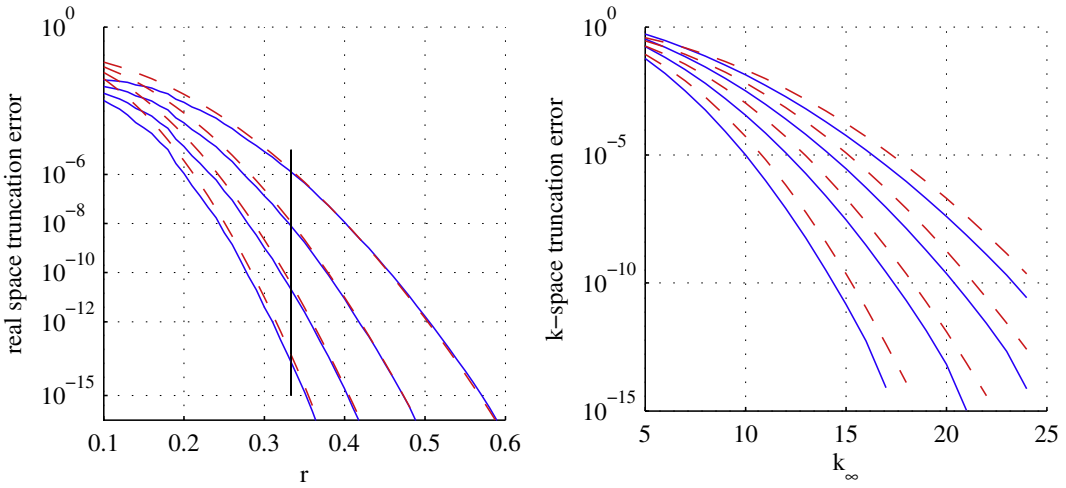

Fig. 7. Left: Computed real space truncation error and estimate $E^R \approx \frac{1}{10} e^{-\xi^2 r^2}$. Error computed by implementing a variable truncation radius in (15), since already $p = 1$ would have converged (15) in this range of ξ (and we have e.g. $N = 1000$, but E^R indep. N). Vertical line at $r = 1/3$ instructive for the present application. Right: Computed **k**-space truncation error (12), and approximation $E^F \approx e^{-k^2/4\xi^2}$. In both Figs.: Hasimoto decomposition, the truncation estimate as dashed line, $\xi = 10, 12, 14, 16$, $\sum \|\mathbf{f}\| = 1$.

Proceeding similarly with the **k**-space sum, i.e. computing the error when using the proposed fast method in a similar norm, with the same sequence of ξ as above, gives.

$M =$	27	35	43	53
$N = 2$	4.2×10^{-07}	6.0×10^{-09}	4.4×10^{-10}	9.6×10^{-12}
$N = 6$	6.6×10^{-08}	1.8×10^{-09}	1.1×10^{-10}	2.3×10^{-12}
$N = 10$	4.9×10^{-08}	8.3×10^{-10}	8.8×10^{-11}	1.9×10^{-12}
$N = 14$	3.2×10^{-08}	9.9×10^{-10}	6.4×10^{-11}	1.5×10^{-12}
$N = 18$	3.0×10^{-08}	7.4×10^{-10}	4.8×10^{-11}	7.8×10^{-13}

The small values for N here are due to the expense of converging the sum (17) at such large values of ξ to compute the error. None the less, the heuristic error estimates proposed here seem vindicated and we proceed to the scaling towards large N.

5.3. Overall complexity

As the domain is scaled up, the width of the Gaussians must be kept fixed, so that they always have support on P^3 grid points. The to-grid (19) and from-grid (21) steps in the FFT method will scale linearly. Since the domain needs to grow with N (to keep the real space sum $\mathcal{O}(N)$), the size of the FFT-grid will grow with N as well, according to (29). Hence, we need to think of the FFT-cost as $\mathcal{O}(N \log(N))$ rather than $\mathcal{O}(M \log(M))$. In this sense our method is $\mathcal{O}(N)$ but coupled to the real space sum we can get no better than $\mathcal{O}(N \log(N))$.

5.4. Numerical results for the full problem

To illustrate the computational complexity of the method described here we take a tolerance $\varepsilon = 10^{-8}$ and the domain to be initially $L_0 = 1$. We split the domain into $3 \times 3 \times 3$ boxes, i.e. $q_0 = 3$. This implies $\xi \approx 12$. Keeping with the estimates from the previous section and taking the source count to initially be $N_0 = 20$, $N_0 = 300$ and $N_0 = 1500$ points, we get the following parameters as the problem is scaled up:

$L =$	1	4/3	5/3	2	7/3	8/3	3
$N =$	20	47	93	160	254	379	540
$N =$	300	711	1389	2400	3811	5689	8100
$N =$	1500	3556	6944	12000	19056	28444	40500
$M =$	35	45	57	67	79	89	101

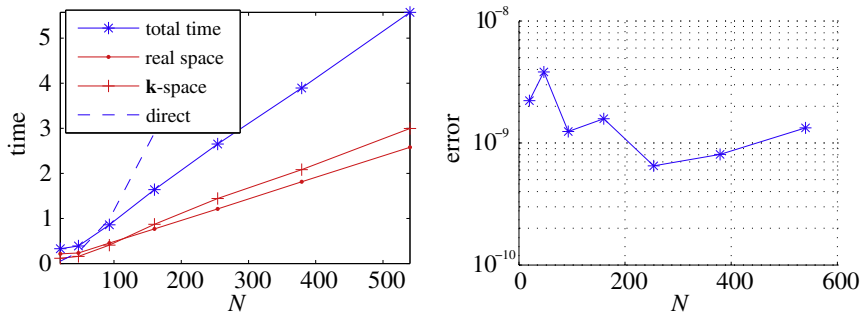

Fig. 8. Left: Run time (seconds) vs. number of points, with FFT-method for **k**-space sum and truncated real space sum from Section 5.1, with $N_0 = 20$. As dashed, the direct Ewald summation (3), C implementation. Note break even at $N < 100$. Right: Error with respect to well converged direct summation satisfies desired accuracy $\varepsilon = 10^{-8}$.

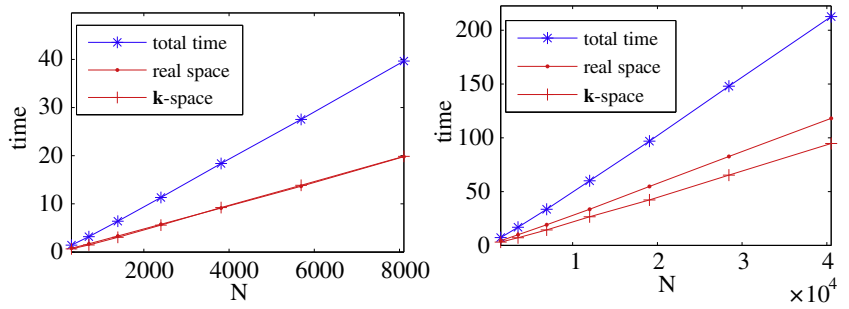

Fig. 9. Run time (seconds) vs. number of points, with FFT-method for **k**-space sum and truncated real space sum from Section 5.1, for different concentrations. Left: $N_0 = 300$. Right: $N_0 = 1500$.

Note that $h := L/M$ is constant. We let $w = hP/2$, with $P = 19$, and $m = 7$.

We use the Hasimoto decomposition here. The timing results corresponding to this numerical experiment are presented in Fig. 8 and 9. We note the expected scaling. The actual magnitude of the measured run-times do not say much, since we have not invested time towards a high-performance implementation of either the FFT-method for the **k**-space sum or the truncated real space summation method. None the less, we note with some satisfaction that the run-time of the real and **k**-space sums are well balanced.

In Fig. 8 we also include a comparison with direct Ewald summation (3) to equal accuracy to illustrate that one can get to much larger systems with our method, and, crucially, that the desired accuracy is met. Following conventional wisdom, we have taken $\xi = \sqrt{\pi/L}$, and found that three layers in both real and **k**-space was appropriate.

6. Conclusions

We have presented an $\mathcal{O}(N)$ and spectrally accurate method for computing the reciprocal space sum in Ewald summation for the Stokes potential, where N is the number of source points. A convolution was computed to obtain a grid representation of the source distribution directly. This avoids the need for polynomial interpolation, which would otherwise affect the reciprocal space convergence. Notably, the parameters of our method can easily be adjusted to control the approximation errors such that the dominating error comes from the **k**-space truncation.

Given a domain size, a tolerance ε and a decomposition parameter ξ, this determines the needed grid size to bound the error arising from the truncation of the **k**-space sum. Then the parameters for the Gaussians can be chosen accordingly, such that the approximation errors are sufficiently small. The cost of the FFTs are proportional to the grid size and independent of N. The footprint of each Gaussian is P^3 points, and the cost of the to-grid and from-grid steps is linear in N, which gives the asymptotic cost of computing the reciprocal sum.

The break-even point over direct Ewald summation for the **k**-space sum has been suggested in numerical experiments to be very low – around or below $N = 50$ – depending on the Ewald parameter, even when a very basic implementation of our method is used.

As was explained above, for a fixed domain size and a fixed choice of the decomposition parameter ξ, our method for evaluating the reciprocal sum is $\mathcal{O}(N)$. Within this setting, the real space sum is however $\mathcal{O}(N^2)$. To balance the cost of evaluating the two sums, the value of ξ must be adjusted with respect to the increasing concentration of point sources. Or, in the case of

constant concentration, the domain size will grow as N is increased. This case was discussed in Section 5, yielding an $\mathcal{O}(N \log N)$ method for the evaluation of the full sum.

We have given error bounds and estimates for our proposed method and the underlying Ewald summation methods. These provide practical parameter choices such that a desired accuracy is attained. We put this into use in Section 5 and support our error analysis with numerical results.

It is also interesting to emphasize that the proposed algorithm is elementary, in the sense that one can implement it in e.g. Matlab with no more than 50 lines of code.

Acknowledgments

The feedback from Professor J. Strain and the anonymous reviewers has been very valuable. Their added perspectives and clarification requests have significantly altered and improved the exposition of this work. We gratefully acknowledge their contributions.

A.K.T. is a Royal Swedish Academy of Sciences Research Fellow supported by a grant from the Knut and Alice Wallenberg Foundation and thankfully acknowledges this support. We also thank Doghonay Arjmand for helpful and enthusiastic contributions to mathematical details.

Appendix A. Comparison of decompositions

Here we present a brief comparison of the Hasimoto and Beenakker/Pozrikidis decompositions, as defined in Section 2.

In Fig. A.10 we present the required computational time as a function of attained accuracy, with each decomposition. Here we see that the Hasimoto decomposition has a clear edge over the Beenakker decomposition, especially for the real space sum.

In Table A.1 we give an expanded version of the table from Section 2, to include a comparison between the two decompositions, and non-unit box domains.

It should be noted that in both these comparisons there is a dependence on the particle distribution. In a severely clustered situation, these conclusions here may not reliably hold.

Table A.1
Convergence of Beenakker and Hasimoto decompositions, with respect to number of image boxes and Fourier modes, i.e. the number of layers required to obtain 16 digits of accuracy. Both for unit cube and 1-by-1-by-4 size domains, with $N = 10$ particles at randomly distributed positions. The term "layers" refers to the number of image boxes and Fourier modes, respectively, included when summing (3).

ξ	Real space		**k**-space		
	Layers	Run-time	Layers	Run time	Total time (s)
Beenakker decomposition, box size (1,1,1)					
0.5	13	1.32	1	0.00	1.32
1.0	7	0.23	2	0.00	0.23
2.0	4	0.05	4	0.01	0.06
4.0	2	0.01	8	0.04	**0.04**
8.0	1	0.00	15	0.22	0.22
10.0	1	0.00	19	0.46	0.46
Beenakker decomposition, box size (1,1,4)					
0.50	13	1.50	4	0.01	1.50
1.00	7	0.25	8	0.04	0.29
2.00	3	0.03	15	0.25	**0.27**
4.00	2	0.01	31	2.05	2.06
8.00	1	0.00	61	15.15	15.15
Hasimoto decomposition, box size (1,1,1)					
0.5	12	0.67	1	0.00	0.67
1.0	6	0.09	1	0.00	0.09
2.0	3	0.01	3	0.00	**0.02**
4.0	2	0.01	7	0.02	0.03
8.0	1	0.00	14	0.17	0.17
10.0	1	0.00	18	0.35	0.35
Hasimoto decomposition, box size (1,1,4)					
0.50	12	0.67	3	0.00	0.67
1.00	6	0.09	7	0.03	**0.12**
2.00	3	0.02	14	0.18	0.20
4.00	2	0.01	29	1.52	1.53
8.00	1	0.00	58	11.86	11.86

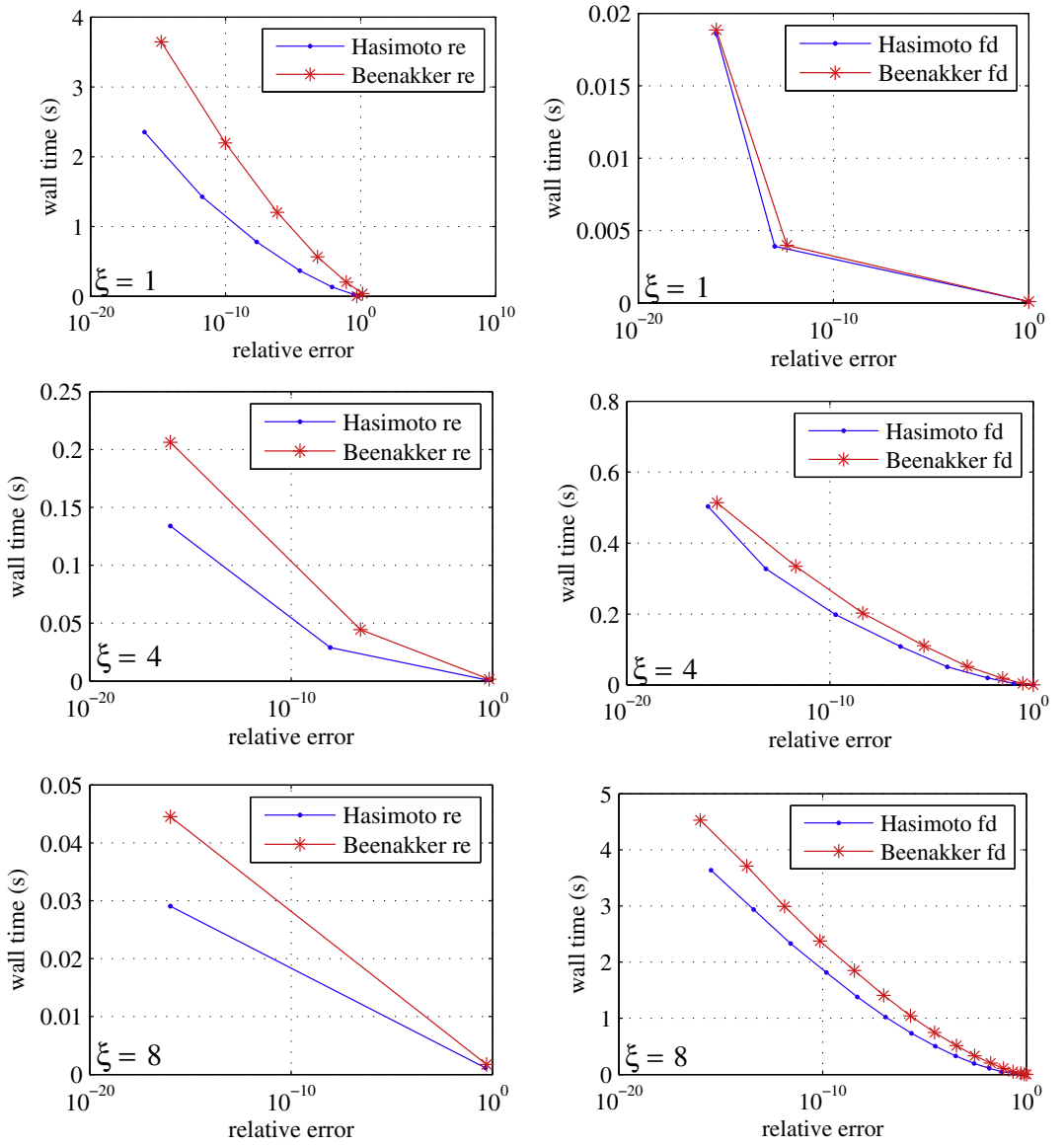

Fig. A.10. Comparison of decompositions. Computational time as a function of error, as the number of layers in real and **k**-space is increased. $N = 50$. Top row: $\xi = 1$. Middle row: $\xi = 4$. Bottom row: $\xi = 8$. Left column: Real space sum. Right column: **k**-space sum. The Hasimoto decomposition provides faster converging sums, notably so in real space.

Appendix B. Fast Gaussian gridding

Here we adapt the ideas from Greengard and Lee [22] to the task of efficiently computing the grid-representation of our source distribution (19), i.e. a sum on the form

$$H(\mathbf{x}) = C\sum_n \mathbf{f}_n e^{-\alpha|\mathbf{x}-\mathbf{x}_n|_*^2}$$

The key observation is that we wish to evaluate $H(\mathbf{x})$ on an equidistant grid, i.e. $\mathbf{x} = [ih, jh, kh] + \mathbf{x}'$, where $\mathbf{x}'$ is some offset and (i,j,k) are integer index triplets in the range $0, 1, \ldots, M-1$. Recall that the Gaussians in (19) have support on a cube of M points in each dimension. Naively evaluating the sum above would need NM^3 evaluations of $\exp(x)$. An elementary rearrangement of terms will reduce this count to $4N$ evaluations of $\exp(x)$ plus roughly $N(M^3 + 3M)$ multiplications (cf. [22, p. 448]). To see this, take $\mathbf{x}' = 0$ and consider one of the factors in the sum above,

$$e^{-\alpha(x-x_n)^2} = e^{-\alpha(ih-x_n)^2} = e^{-\alpha((ih)^2-2ihx_n+x_n^2)} = \underbrace{e^{-\alpha(ih)^2}}_{(a)}\left(\underbrace{e^{-2\alpha hx_n}}_{(b)}\right)^i \underbrace{e^{-\alpha x_n^2}}_{(c)}.$$

Note that the term (a) is independent of x_n, so those M evaluations of $\exp(x)$ are done once, stored and reused for each of the N sources x_n. The terms (b) and (c) each incur one $\exp(x)$ for each x_n. The same procedure is then applied for $e^{-\alpha(y-y_n)^2}$ and ditto for z, so that we obtain

$$e^{-\alpha(x-x_n)^2}e^{-\alpha(y-y_n)^2}e^{-\alpha(z-z_n)^2} = e^{-\alpha(ih)^2}e^{-\alpha(jh)^2}e^{-\alpha(kh)^2}\left(e^{2\alpha hx_n}\right)^i\left(e^{2\alpha hy_n}\right)^j\left(e^{2\alpha hz_n}\right)^k e^{-\alpha(x_n^2+y_n^2+z_n^2)},$$

which exposes the same structure as above.

The actual speedup of using this procedure depends on how faster the CPU can retire multiplication instructions (best case being two per clock cycle) in relation to computing $\exp(x)$ (typically 15–50 cycles depending on CPU). Greengard and Lee report a factor of 5–10 improvement in run-time in 2D, and the operation counts are more favorable for 3D.

References

[1] C. Pozrikidis, Boundary Integral and Singularity Methods for Linearized Viscous Flow, Cambridge University Press, 1992.
[2] P. Ewald, Die berechnung optischer und elektrostatischer gitterpotentiale, Ann. Phys. 64 (1921) 253–287.
[3] T. Darden, D. York, L. Pedersen, Particle mesh Ewald - an N.log(N) method for Ewald sums in large systems, J. Chem. Phys. 98 (1993) 10089–10092.
[4] U. Essmann, L. Perera, M. Berkowitz, T. Darden, H. Lee, L. Pedersen, A smooth particle mesh Ewald method, J. Chem. Phys. 103 (1995) 8577–8593.
[5] M. Deserno, C. Holm, How to mesh up Ewald sums. I.A theoretical and numerical comparison of various particle mesh routines, J. Chem. Phys. 109 (1998) 7678–7693.
[6] Y. Shan, J. Klepeis, M. Eastwood, R. Dror, D. Shaw, Gaussian split Ewald: a fast Ewald mesh method for molecular simulation, J. Chem. Phys. 122 (2005).
[7] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys 73 (1987) 325–348.
[8] A.-K. Tornberg, L. Greengard, A fast multipole method for the three-dimensional Stokes equations, J. Comput. Phys 227 (2008) 1613–1619.
[9] H. Wang, T. Lei, J. Li, J. Huang, Z. Yao, A parallel fast multipole accelerated integral equation scheme for 3D Stokes equations, Int. J. Numer. Meth. Eng. 70 (2007) 812–839.
[10] L. Greengard, M. Kropinski, Integral equation methods for Stokes flow in doubly-periodic domains, J. Eng. Math. 48 (2004) 157–170.
[11] L.X. Ying, G. Biros, D. Zorin, A high-order 3d boundary integral equation solver for elliptic pdes in smooth domains, J. Comput. Phys 219 (2006) 247–275.
[12] H. Hasimoto, On the periodic fundamental solutions of the Stokes equations and their application to viscous flow past a cubic array of spheres, J. Fluid Mech. 5 (1959) 317–328.
[13] C.W.J. Beenakker, Ewald sum of the Rotne–Prager Tensor, J. Chem. Phys. 85 (1986) 1581–1582.
[14] D. Saintillan, E. Darve, E. Shaqfeh, A smooth particle-mesh Ewald algorithm for Stokes suspension simulations: the sedimentation of fibers, Phys. Fluids 17 (2005).
[15] A. Sierou, J. Brady, Accelerated Stokesian dynamics simulations, J. Fluid Mech. 448 (2001) 115–146.
[16] J. Brady, G. Bossis, Stokesian dynamics, Annu. Rev. Fluid Mech. 20 (1988) 111–157.
[17] C. Pozrikidis, Computation of periodic Green's functions of Stokes flow, J. Eng. Math. 30 (1996) 79–96.
[18] J.T. Beale, J. Strain, Locally corrected semi-lagrangian methods for stokes flow with moving elastic interfaces, J. Comput. Phys 227 (2008) 3896–3920.
[19] J. Strain, Locally-corrected spectral methods and overdetermined elliptic systems, J. Comput. Phys 224 (2007) 1243–1254.
[20] J. Strain, Fast potential-theory.2. layer potentials and discrete sums, J. Comput. Phys 99 (1992) 251–270.
[21] A. Dutt, V. Rokhlin, Fast fourier-transforms for nonequispaced data, SIAM J. Sci. Comput. 14 (1993) 1368–1393.
[22] L. Greengard, J. Lee, Accelerating the nonuniform fast fourier transform, SIAM Rev. 46 (2004) 443–454.
[23] I. Sammis, J. Strain, A geometric nonuniform fast Fourier transform, J. Comput. Phys 228 (2009) 7086–7108.
[24] J. Kolafa, J. Perram, Cutoff errors in the Ewald summation formulas for point-charge systems, Mol. Simulat. 9 (1992) 351–368.
[25] M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, Oxford University Press, 1989.