Klinteberg *et al. Res Math Sci(2017)4:1*
DOI 10.1186/s40687-016-0092-7

Research in the Mathematical Sciences

RESEARCH Open Access

Fast Ewald summation for free-space Stokes potentials

Ludvig af Klinteberg, Davoud Saffar Shamshirgar and Anna-Karin Tornberg*

*Correspondence:
akto@kth.se
KTH Mathematics, Linné Flow Centre/Swedish e-Science Research Centre, 100 44 Stockholm, Sweden

Abstract

We present a spectrally accurate method for the rapid evaluation of free-space Stokes potentials, i.e., sums involving a large number of free space Green's functions. We consider sums involving stokeslets, stresslets and rotlets that appear in boundary integral methods and potential methods for solving Stokes equations. The method combines the framework of the Spectral Ewald method for periodic problems (Lindbo and Tornberg in J Comput Phys 229(23):8994–9010, 2010. doi:10.1016/j.jcp.2010.08.026), with a very recent approach to solving the free-space harmonic and biharmonic equations using fast Fourier transforms (FFTs) on a uniform grid (Vico et al. in J Comput Phys 323:191–203, 2016. doi:10.1016/j.jcp.2016.07.028). Convolution with a truncated Gaussian function is used to place point sources on a grid. With precomputation of a scalar grid quantity that does not depend on these sources, the amount of oversampling of the grids with Gaussians can be kept at a factor of two, the minimum for aperiodic convolutions by FFTs. The resulting algorithm has a computational complexity of $O(N \log N)$ for problems with N sources and targets. Comparison is made with a fast multipole method to show that the performance of the new method is competitive.

1 Background

In this paper, we consider the evaluation of free-space potentials of Stokes flow, i.e., vector fields defined by sums involving a large number of free space Green's functions such as the so-called stokeslet, stresslet or rotlet. The stokeslet is the free space Green's function for velocity and is given by

$$S(\mathbf{r}) = \frac{1}{r}\mathbf{I} + \frac{1}{r^3}\mathbf{r}\mathbf{r}, \quad \text{or} \quad S_{jl}(\mathbf{r}) = \frac{\delta_{jl}}{r} + \frac{r_j r_l}{r^3}, \quad j, l = 1, 2, 3,$$

with $r = |\mathbf{r}|$ and where δ_{jl} is the Kronecker delta. The stresslet and rotlet will be introduced in the following. The discrete sums are on the form

$$\mathbf{u}(\mathbf{x}_\mathrm{m}) = \sum_{\substack{\mathrm{n}=1 \\ \mathrm{n}\neq\mathrm{m}}}^{N} S(\mathbf{x}_\mathrm{m} - \mathbf{x}_\mathrm{n})\mathbf{f}(\mathbf{x}_\mathrm{n}), \quad \mathrm{m} = 1, \ldots, N. \tag{1}$$

and appear in boundary integral methods and potential methods for solving Stokes equations.

Springer Open

These sums have the same structure as the classical Coulombic or gravitational N-body problems that involve the harmonic kernel, and the direct evaluation of such a sum for $\mathtt{m} = 1, \ldots, N$ requires $O(N^2)$ work. The Fast Multipole Method (FMM) can reduce that cost to $O(N)$ work, where the constant multiplying N will depend on the required accuracy. FMM was first introduced by Greengard and Rokhlin for the harmonic kernel in 2D and later in 3D [5,15] and has since been extended to other kernels, including the fundamental solutions of Stokes flow considered here [12,16,27,29,32]. Related is also the development of a so-called pre-corrected FFT method based on fast Fourier transforms. This method has been applied to the rapid evaluation of stokeslet sums for panel-based discretizations of surfaces [31].

For periodic problems, FFT-based fast methods built on the foundation of so-called Ewald summation have been successful. Also here, development started for the harmonic potential, specifically for evaluation of the electrostatic potential and force in connection to molecular dynamic simulations, see, e.g., the survey by Deserno and Holm [7]. One early method was the *Particle Mesh Ewald* (PME) method by Darden et al. [6], later refined to the *Smooth Particle Mesh Ewald* (SPME) method by Essman et al. [8]. The SPME method was extended to the fast evaluation of the stokeslet sum by Saintillan et al. [26]. To recover the exponentially fast convergence of the Ewald sums that is lost when such a traditional PME approach is used, the present authors have developed a spectrally accurate PME-type method, the Spectral Ewald (SE) method both for the sum of stokeslets [21], and stresslets [3]. It has also been implemented for the sum of rotlets [1], and the source code is available online [24]. The Spectral Ewald method was recently used to accelerate the Stokesian Dynamics simulations in [30].

The present work deals with the efficient and fast summation of free space Green's functions for Stokes flow (stokeslets, stresslets and rotlets), as exemplified by the sum of stokeslets in (1). The problem has no periodicity, but the approach will still be based on Ewald summation and fast Fourier transforms (FFTs), using ideas from [28] to extend the Fourier treatment to the free-space case. Before we explain this further, we will introduce the idea behind Ewald summation.

1.1 Triply periodic Ewald summation

Consider the Stokes equations in $\mathbb{R}^3$, singularly forced at arbitrary locations $\mathbf{x}_\mathtt{n}$, $\mathtt{n} = 1, \ldots, N$, with strengths $8\pi\mu\mathbf{f}(\mathbf{x}_\mathtt{n}) \in \mathbb{R}^3$ (with the $8\pi\mu$ scaling for convenience). Introduce the three-dimensional delta function $\delta(\mathbf{x} - \mathbf{x}_0)$, and write

$$-\nabla p + \mu\nabla^2\mathbf{u} + \mathbf{g}(\mathbf{x}) = 0, \quad \mathbf{g}(\mathbf{x}) = 8\pi\mu\sum_{\mathtt{n}=1}^{N}\mathbf{f}(\mathbf{x}_\mathtt{n})\,\delta(\mathbf{x} - \mathbf{x}_\mathtt{n}),$$
$$\nabla\cdot\mathbf{u} = 0,$$

where $\mathbf{u}$ is the velocity, p is the pressure and μ is the viscosity. The free-space problem is given by adding the boundary condition that the fluid is at rest at infinity,

$$\lim_{|\mathbf{x}|\to\infty}\mathbf{u} = 0.$$

The solution to this problem, evaluated at the source locations, is given by (1).

The classical Ewald summation formulas were derived for the triply periodic problem for the electrostatic potential by Ewald [9] and for the stokeslet by Hasimoto [17]. Here, assume that all the point forces are located within a box $\mathcal{D} = [-L_1/2, L_1/2] \times [-L_2/2, L_2/2] \times [-L_3/2, L_3/2]$ and that we impose periodic boundary conditions. The solution to this problem is a sum not only over all the point forces, but also over all their periodic replicas,

$$\mathbf{u}^{3P}(\mathbf{x}_{\mathtt{m}}) = \sum_{\mathbf{p}\in P_3} \sum_{\mathtt{n}=1}^{N*} S(\mathbf{x}_{\mathtt{m}} - \mathbf{x}_{\mathtt{n}} + \mathbf{p})\mathbf{f}(\mathbf{x}_{\mathtt{n}}), \quad \mathtt{m} = 1, \ldots, N.$$

Here, the sum over $\mathbf{p}$ formalizes the periodic replication of the point forces with

$$P_3 = \left\{(j_1L_1, j_2L_2, j_3L_3\} : \mathbf{j} \in \mathbb{Z}^3\right\}.$$

The $N*$ indicates that the term ($\mathtt{n} = \mathtt{m}$, $\mathbf{p} = \mathbf{0}$) is excluded from the sum. The slow decay of the stokeslet, however, makes this infinite sum divergent. To make sense of this summation, one usually assumes that the point forces are balanced by a mean pressure gradient, such that the velocity integrates to zero over the periodic box. Under these assumptions, Hasimoto [17] derived the following Ewald summation formula

$$\begin{aligned}\mathbf{u}^{3P}(\mathbf{x}_{\mathtt{m}}) = &\sum_{\mathbf{p}\in P_3} \sum_{\mathtt{n}=1}^{N*} S^R(\mathbf{x}_{\mathtt{m}} - \mathbf{x}_{\mathtt{n}} + \mathbf{p}, \xi)\mathbf{f}(\mathbf{x}_{\mathtt{n}}) \\ &+ \frac{1}{V} \sum_{|\mathbf{k}|\neq 0} \hat{S}^F(\mathbf{k}, \xi) \sum_{\mathtt{n}=1}^{N} \mathbf{f}(\mathbf{x}_{\mathtt{n}}) e^{-i\mathbf{k}\cdot(\mathbf{x}_{\mathtt{m}} - \mathbf{x}_{\mathtt{n}})} \\ &+ \lim_{|\mathbf{r}|\to 0} \left(S^R(\mathbf{r}, \xi) - S(\mathbf{r})\right) \mathbf{f}(\mathbf{x}_{\mathtt{m}}),\end{aligned} \tag{2}$$

where the $\mathtt{n} = \mathtt{m}$, $\mathbf{p} = 0$ term is excluded from the real space sum, $V = L_1L_2L_3$, and

$$\begin{aligned}S^R(\mathbf{r}, \xi) &= 2\left(\frac{\xi e^{-\xi^2 r^2}}{\sqrt{\pi} r^2} + \frac{\operatorname{erfc}(\xi r)}{2r^3}\right)\left(r^2\mathbf{I} + \mathbf{r}\mathbf{r}\right) - \frac{4\xi}{\sqrt{\pi}} e^{-\xi^2 r^2}\mathbf{I}, \\ \hat{S}^F(\mathbf{k}, \xi) &= 8\pi\left(1 + \frac{k^2}{4\xi^2}\right)\frac{1}{k^4}\left(\mathbf{I}k^2 - \mathbf{k}\mathbf{k}\right) e^{-k^2/4\xi^2},\end{aligned} \tag{3}$$

with $r = |\mathbf{r}|$, $k = |\mathbf{k}|$,

$$\mathbf{k} \in \mathbb{K} = \left\{2\pi(j_1/L_1, j_2/L_2, j_3/L_3) : \mathbf{j} \in \mathbb{Z}^3\right\},$$

and

$$\lim_{|\mathbf{r}|\to 0} \left(S^R(\mathbf{r}, \xi) - S(\mathbf{r})\right) = -\frac{4\xi}{\sqrt{\pi}}\mathbf{I}. \tag{4}$$

The last term in (2) is commonly referred to as the self-interaction term. When evaluating the potential at $\mathbf{x}_{\mathtt{m}}$, we should exclude the contribution from the point force at that same location. For the real space part, we can directly skip the term in the summation when $\mathbf{p} = 0$ and $\mathtt{n} = \mathtt{m}$. We, however, need to subtract the contribution from this point that has

been included in the Fourier sum. We can use that $S^F = S - S^R$, and subtract the limit as $|\mathbf{r}| \to 0$ (2). Both S and S^R are singular, but the limit of the difference is finite (4).

Both sums now decay exponentially, one in real space and one in Fourier space. The parameter $\xi > 0$ is a decomposition parameter that controls the decay of the terms in the two sums. The sum in real space can naturally be truncated to exclude interactions that are now negligible. The sum in k-space, however, is still a sum of complexity $O(N^2)$, now with a very large constant introduced by the sum over $\mathbf{k}$.

Methods in the PME family make use of FFTs to evaluate the k-space sum, accelerating the evaluation such that ξ can be chosen larger to push more work into the k-space sum, allowing for tighter truncation of the real space sum, and in total an $O(N \log N)$ method. This procedure introduces approximations since a grid must be used and, as with the FMM, the constant multiplying $N \log N$ will depend on the accuracy requirements.

1.2 The free-space problem and this contribution

Considering the free space problem, we can introduce the same kind of decomposition as in (2). The real space sum stays the same, with the minor change that the sum over $\mathbf{p}$ is removed, and the self interaction term does not change. However, the discrete sum in Fourier space is replaced by the inverse Fourier transform,

$$\mathbf{u}^F(\mathbf{x}, \xi) = \frac{1}{(2\pi)^3} \int_{\mathbb{R}^3} \hat{S}^F(\mathbf{k}, \xi) \cdot \sum_{\mathrm{n}=1}^{N} \mathbf{f}(\mathbf{x}_\mathrm{n}) e^{i\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}_\mathrm{n})} \, \mathrm{d}\mathbf{k}. \tag{5}$$

Here, note the $1/k^2$ singularity in $\hat{S}^F(\mathbf{k}, \xi)$ as defined in (3). The integral is well defined, and integration can be performed, e.g., in spherical coordinates. A numerical quadrature method in spherical coordinates would, however, require non-uniform FFTs for non-rectangular grids in k-space. Instead, we will use a very recent idea introduced by Vico et al. [28] to solve free space problems by FFTs on uniform grids.

The method by Vico et al. [28] is based on the idea to use a modified Green's function. With a right-hand side of compact support, and a given domain inside which the solution is to be found, a truncated Green's function can be defined that coincides with the original one for a large enough domain (and is zero elsewhere), such that the analytical solution defined through a convolution of the Green's function with the right-hand side remains unchanged. The gain is that the Fourier transform of this truncated Green's function will have a finite limit at $\mathbf{k} = 0$. A length scale related to the truncation will, however, be introduced, introducing oscillations in Fourier space which will require some upsampling to resolve.

The authors of [28] present this approach for radial Green's functions, e.g., the harmonic and biharmonic kernels. In the present work, we are considering kernels that are not radial. We will, however, use this idea in a substep of our method, defining the Fourier transform of the biharmonic (for stokeslet and stresslet) or harmonic (for rotlet) kernels, and define our non-radial kernels from these. The need of upsampling that the truncation brings can be taken care of in a precomputation step, and hence for a scalar quantity only. What remains is an aperiodic discrete convolution that requires an upsampling of a factor of two.

The key ingredients in our method for the rapid summation of kernels of Stokes flow (stokeslet, stresslet and rotlet) in free space will hence be the following. We make use

of the framework of Ewald summation, to split the sums in two parts—one that decays rapidly in real space, and one in Fourier space. The Fourier space treatment is based on the Spectral Ewald method for triply and doubly periodic problems that has been developed previously [3,21–23]. This means that point forces will be interpolated to a uniform grid using truncated Gaussian functions that are scaled to allow for best possible accuracy given the size of the support. The implementation of the gridding is made efficient by the means of Fast Gaussian Gridding (FGG) [14,22].

In the periodic problem, an FFT of each component of the grid function is computed, a scaling is done in Fourier space, and after inverse FFTs, truncated Gaussians are again used to evaluate the result at any evaluation point. The new development in this paper is to extend this treatment to the free space case, when periodic sums are replaced by discretized Fourier integrals. As mentioned above, a precomputation will be made to compute a modified free-space harmonic or biharmonic kernel that will be used to define the scaling in Fourier space.

The details are yet to be explained, but as we hope to convey in the following, the method that we develop here for potentials of Stokes flow can easily be extended to other kernels. For any kernel that can be expressed as a differentiation of the harmonic and/or biharmonic kernel, the Ewald summation formulas can easily be derived and only minor changes in the implementation of the method will be needed.

Any method based on Ewald summation and acceleration by FFTs will be most efficient in the triply periodic case. As soon as there is one or more directions that are not periodic, there will be a need of some oversampling of FFTs, which will increase the computational cost. For the FMM, the opposite is true. The free space problem is the fastest to compute, and any periodicity will invoke an additional cost, which will become substantial or even overwhelming if the base periodic box has a large aspect ratio. Hence, implementing the FFT-based Spectral Ewald method for a free-space problem and comparing it to an FMM method will be the worst possible case for the SE method. Still, as we will show in the results section, using an open source implementation of the FMM [13], our new method is competitive and often performs better than that implementation of the FMM for uniform point distributions (one can, however, expect this adaptive FMM to perform better for highly non-uniform distributions).

There is an additional value in having a method that can be used for different periodicities, thereby keeping the structure intact and easing the integration with the rest of the simulation code, concerning, e.g., modifications of quadrature methods in a boundary integral method to handle near interactions. A three-dimensional adaptive FMM is also much more intricate to implement than the SE method. Open source software for the Stokes FMM does exist for the free space problem (as the one used here), but we are not aware of any software for the periodic problem.

1.3 Outline of paper

The outline of the paper is as follows. In Sect. 2, we start by introducing the stokeslet, stresslet and rotlet, and write them on the operator form that we will later use. In Sect. 3, we introduce the ideas behind Ewald decomposition and establish a framework for straightforward derivation of decompositions of different kernels. The new approach to solving free-space problems by FFTs introduced by Vico et al. [28] is presented in the following section, together with a detailed discussion on oversampling needs and precomputation.

The new method for evaluating the Fourier space component is described in Sect. 5, while the evaluation of the real space sum is briefly commented on in Sect. 6. New truncation error estimates are derived in Sect. 7, and in Sect. 8 we summarize the full method. Numerical results are presented in Sect. 9, where the performance of the method is discussed and comparison to an open source implementation of the FMM [13] is made.

2 Green's functions of free-space Stokes flow

We will consider three different Green's functions of free-space Stokes flow, the stokeslet S, the stresslet T and the rotlet Ω. They are defined as

$$S_{jl}(\mathbf{r}) = \frac{\delta_{jl}}{r} + \frac{r_j r_l}{r^3}, \tag{6}$$

$$T_{jlm}(\mathbf{r}) = -6\frac{r_j r_l r_m}{r^5}, \tag{7}$$

$$\Omega_{jl}(\mathbf{r}) = \epsilon_{jlm}\frac{r_m}{r^3}, \tag{8}$$

where $r = |\mathbf{r}|$. They can equivalently be formulated as operators acting on the fundamental solutions of the biharmonic and—in the case of the rotlet—harmonic equations,

$$B(\mathbf{r}) = r,$$
$$H(\mathbf{r}) = 1/r.$$

We then write [10,25]

$$S_{jl}(\mathbf{r}) = \left(\delta_{jl}\nabla^2 - \nabla_j\nabla_l\right) r, \tag{9}$$

$$T_{jlm}(\mathbf{r}) = \left[\left(\delta_{jl}\nabla_m + \delta_{lm}\nabla_j + \delta_{mj}\nabla_l\right)\nabla^2 - 2\nabla_j\nabla_l\nabla_m\right] r, \tag{10}$$

$$\Omega_{jl}(\mathbf{r}) = \left(-\epsilon_{jlm}\nabla_m\nabla^2\right) r = \left(-2\epsilon_{jlm}\nabla_m\right)\frac{1}{r}. \tag{11}$$

Here ϵ_{jlm} is the Levi-Civita symbol, and repeated indices are summated according to the Einstein summation convention.

For a single forcing term $8\pi\mu\mathbf{f}$ at a source location $\mathbf{x}_0$, the velocity field of the solution is given by

$$\mathbf{u}(\mathbf{x}) = S(\mathbf{x}-\mathbf{x}_0)\mathbf{f}, \text{ or } u_j(\mathbf{x}) = S_{jl}(\mathbf{x}-\mathbf{x}_0)f_l, \quad j = 1, 2, 3.$$

Similarly, the stress field and vorticity associated with this solution can be written,

$$\sigma_{jl}(\mathbf{x}) = T_{jlm}(\mathbf{x}-\mathbf{x}_0)f_m, \quad \omega_j(\mathbf{x}) = \Omega_{jl}(\mathbf{x}-\mathbf{x}_0)f_l.$$

In integral equations, the stresslet often appears instead multiplying sources with two indices, also producing a velocity,

$$\mathbf{u}_j(\mathbf{x}) = T_{jlm}(\mathbf{x}-\mathbf{x}_0)f_{lm},$$

and this is the case that we will consider here. The typical form is then

$$f_{lm} = n_l q_m, \tag{12}$$

where $\mathbf{n}$ is a vector normal to a surface and $\mathbf{q}$ is a double-layer density.

We want to rapidly evaluate discrete-sum potentials of the type given in (1), either at the source locations as indicated in that sum, or at any other arbitrary points, and we want to do so for the three different Green's functions. To allow for a generic notation in the following despite the differences, we introduce the unconventional notation

$$\mathbf{u}(\mathbf{x}) = \sum_{\mathrm{n}=1}^{N} G(\mathbf{x} - \mathbf{x}_{\mathrm{n}}) \cdot \mathbf{f}(\mathbf{x}_{\mathrm{n}}), \tag{13}$$

where G can denote either the stokeslet S, the stresslet T and the rotlet Ω, and the dot-notation $\mathbf{u} = G(\mathbf{r}) \cdot \mathbf{f}$ will be understood to mean

$$u_j(\mathbf{x}) = S_{jl}(\mathbf{r})f_l, \quad u_j(\mathbf{x}) = T_{jlm}(\mathbf{r})f_{lm}, \quad u_j(\mathbf{x}) = \Omega_{jl}(\mathbf{r})f_l, \quad j = 1, 2, 3,$$

in the three different cases.

3 Ewald summation

3.1 Decomposing the Green's function

In Ewald summation, we take a non-smooth and long-range Green's function G, such as (6–8), and decompose it into two parts,

$$G(\mathbf{r}) = G^R(\mathbf{r}) + G^F(\mathbf{r}).$$

This is done such that G^R, called the real space component, decays exponentially in $r = |\mathbf{r}|$. At the same time, G^F, called Fourier space component, decays exponentially in Fourier space. The original example of this, derived by Ewald [9], decomposes the Laplace Green's function as

$$\frac{1}{r} = \frac{\mathrm{erfc}(\xi r)}{r} + \frac{\mathrm{erf}(\xi r)}{r}, \tag{14}$$

where ξ is a parameter that controls the decay rates in the real and Fourier spaces. Here, the real space component decays like $e^{-\xi^2 r^2}$, while the Fourier space component decays like $k^{-2}e^{-k^2/4\xi^2}$. The rapid decay rates allow truncation of the components; the real space component is reduced to local interactions between near neighbors, while the Fourier space component is truncated at some maximum wave number k_∞.

There are two different ways of deriving an Ewald decomposition, which we shall refer to as screening and splitting. In screening, one introduces a screening function $\gamma(\mathbf{r}, \xi)$, $\int_{\mathbb{R}^3} \gamma(\mathbf{r}, \xi)\, \mathrm{d}\mathbf{r} = 1$, that decays smoothly away from zero. The Green's function is then decomposed using its convolution with γ,

$$G(\mathbf{r}) = G(\mathbf{r}) - (G * \gamma)(\mathbf{r}, \xi) + (G * \gamma)(\mathbf{r}, \xi),$$

such that

$$G^R(\mathbf{r}, \xi) = G(\mathbf{r}) - (G * \gamma)(\mathbf{r}, \xi),$$
$$G^F(\mathbf{r}, \xi) = (G * \gamma)(\mathbf{r}, \xi),$$

and (by the convolution theorem)

$$\widehat{G}^F(\mathbf{k},\xi) = \widehat{G}(\mathbf{k})\widehat{\gamma}(\mathbf{k},\xi), \tag{15}$$

where $\widehat{f}$ denotes the Fourier transform of f,

$$\widehat{f}(\mathbf{k}) = \mathcal{F}[f](\mathbf{k}) = \int_{\mathbb{R}^3} f(\mathbf{x})e^{-i\mathbf{k}\cdot\mathbf{x}}\,\mathrm{d}\mathbf{x}.$$

The original Ewald decomposition (14) can be derived in this fashion, using the screening function

$$\gamma_E(\mathbf{r},\xi) = \xi^3\pi^{-3/2}e^{-\xi^2r^2} \rightleftharpoons \widehat{\gamma}_E(\mathbf{k},\xi) = e^{-k^2/4\xi^2},$$

where $r = |\mathbf{r}|$, $k = |\mathbf{k}|$. For the stokeslet (6), an Ewald decomposition was derived by Hasimoto [17], which was later shown [18] to be equivalent to using the screening function

$$\gamma_H(\mathbf{r},\xi) = \xi^3\pi^{-3/2}e^{-\xi^2r^2}\left(\frac{5}{2} - \xi^2r^2\right) \rightleftharpoons \widehat{\gamma}_H(\mathbf{k},\xi) = e^{-k^2/4\xi^2}\left(1 + \frac{1}{4}\frac{k^2}{\xi^2}\right).$$

In splitting, one starts with the operator form of the Green's function (9,10,11), $G(\mathbf{r}) = \mathrm{K}\,r$, where K is an operator acting on r. Knowing K, one then splits the Green's function using a splitting function Φ,

$$G(\mathbf{r}) = \mathrm{K}[r - \Phi(r,\xi)] + \mathrm{K}\,\Phi(r,\xi),$$

such that

$$\begin{aligned} G^R(\mathbf{r},\xi) &= \mathrm{K}[r - \Phi(r,\xi)], \\ \widehat{G}^F(\mathbf{k},\xi) &= \widehat{\mathrm{K}}(\mathbf{k})\widehat{\Phi}(k,\xi), \end{aligned} \tag{16}$$

where $\widehat{\mathrm{K}}(\mathbf{k})$ denotes the prefactor that is produced when K is applied to $e^{i\mathbf{k}\cdot\mathbf{x}}$ (e.g., if $\mathrm{K} = \Delta$ then $\widehat{\mathrm{K}} = -|\mathbf{k}|^2 = -k^2$). The splitting method was invented by Beenakker [4], who used

$$\Phi_B(r,\xi) = r\,\mathrm{erf}(\xi r) \rightleftharpoons \widehat{\Phi}_B(k,\xi) = -\frac{8\pi}{k^4}\left(1 + \frac{1}{4}\frac{k^2}{\xi^2} + \frac{1}{8}\frac{k^4}{\xi^4}\right)e^{-k^2/4\xi^2}.$$

We have now defined $\widehat{G}^F(\mathbf{k},\xi)$ in two different ways in (15) and (16) and can equate the two. We have that $G(\mathbf{r}) = Kr$, where $r = B(\mathbf{r})$ is the fundamental solution of the biharmonic equation, i.e.,

$$\nabla^4 B(\mathbf{r}) = -8\pi\,\delta(\mathbf{x}).$$

From this, we get

$$\hat{G}(\mathbf{k}) = \widehat{B}(\mathbf{k})\hat{K}(\mathbf{k}) = -\frac{8\pi}{k^4}\hat{K}(\mathbf{k}). \tag{17}$$

Hence, the screening and splitting methods can be shown [2] to be related to each other as

$$\widehat{\Phi} = -\frac{8\pi}{k^4}\widehat{\gamma} \rightleftharpoons \gamma = -\frac{1}{8\pi}\nabla^4\Phi.$$

Using this, one can derive the screening and splitting functions related to the Ewald, Hasimoto and Beenakker decompositions, shown in Table 1.

The relations listed in the table are very useful in derivation of Ewald summation formulas. Finding the real space part $G^R(\mathbf{r}, \xi)$ is easiest using the splitting approach, since this only involves differentiation. The k-space term is, however, simpler to derive with the screening approach. Combining (15) and (17), it directly follows

$$\widehat{G}^F(\mathbf{k}, \xi) = \hat{K}(\mathbf{k})\widehat{B}(\mathbf{k})\widehat{\gamma}(\mathbf{k}, \xi). \tag{18}$$

Considering the information in the table, we can see that the Ewald decomposition yields the fastest decay in Fourier space. However, this screening function can only be used if the Green's function can be written as an operator acting on $1/r$, like the rotlet (11). In this case, we can think about the splitting approach as if splitting $1/r$ such that Φ is $\operatorname{erf}(\xi r)/r$ as in (14).

If we attempt to use the Ewald screening function for the stokeslet or stresslet, this will not produce a useful decomposition since this screening function does not "screen" the point forces. The field produced by a point force convolved with the screening function does not converge rapidly (with distance from the source location) to the field produced by that point force. If we were to do the calculation, this manifests itself in slowly decaying terms in the real space sum.

Both the Hasimoto and Beenacker screening functions work for the stokeslet and stresslet. The Hasimoto decomposition will yield somewhat faster decaying terms in both real and Fourier space and will henceforth be the one that we will use.

3.2 Ewald free-space formulas

In the triply periodic setting, the Ewald summation formula as derived by Hasimoto was given in (2). As given in (5), for the free-space problem the discrete sum in Fourier space is replaced by the inverse Fourier transform. With our generic notation, we can evaluate the discrete-sum potential (13) as

$$\mathbf{u}(\mathbf{x}) = \sum_{\mathrm{n}=1}^{N} G^R(\mathbf{x} - \mathbf{x}_\mathrm{n}, \xi) \cdot \mathbf{f}(\mathbf{x}_\mathrm{n}) + \frac{1}{(2\pi)^3}\int_{\mathbb{R}^3} \widehat{G}^F(\mathbf{k}, \xi) \cdot \sum_{n=1}^{N} \mathbf{f}(\mathbf{x}_\mathrm{n}) e^{i\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}_\mathrm{n})}\, \mathrm{d}\mathbf{k}. \tag{19}$$

Table 1 Summary of the screening and splitting functions related to the Ewald, Hasimoto and Beenakker decompositions

	γ	$\widehat{\gamma}$	Φ
Ewald	$\alpha e^{-\xi^2 r^2}$	$e^{-k^2/4\xi^2}$	$r\operatorname{erf}(\xi r) + \frac{r\operatorname{erf}(\xi r)}{2\xi^2 r^2} + \frac{e^{-\xi^2 r^2}}{\sqrt{\pi}\xi}$
Hasimoto	$\alpha e^{-\xi^2 r^2}\left(\frac{5}{2} - \xi^2 r^2\right)$	$e^{-k^2/4\xi^2}\left(1 + \frac{1}{4}\frac{k^2}{\xi^2}\right)$	$r\operatorname{erf}(\xi r) + \frac{e^{-\xi^2 r^2}}{\sqrt{\pi}\xi}$
Beenakker	$\alpha e^{-\xi^2 r^2}\left(10 - 11\xi^2 r^2 + 2\xi^4 r^4\right)$	$e^{-k^2/4\xi^2}\left(1 + \frac{1}{4}\frac{k^2}{\xi^2} + \frac{1}{8}\frac{k^4}{\xi^4}\right)$	$r\operatorname{erf}(\xi r)$

The constant $\alpha = \xi^3\pi^{-3/2}$, and ξ is the decomposition parameter

We now apply the splitting approach to derive the real space formulas, using the Hasimoto splitting for the stokeslet and stresslet, and the Ewald splitting for the rotlet,

$$
\begin{aligned}
S^R_{jl}(\mathbf{r},\xi) &= \left(\delta_{jl}\nabla^2 - \nabla_j\nabla_l\right)\left[r - \Phi^H(r,\xi)\right],\\
T^R_{jlm}(\mathbf{r}) &= \left[\left(\delta_{jl}\nabla_m + \delta_{lm}\nabla_j + \delta_{mj}\nabla_l\right)\nabla^2 - 2\nabla_j\nabla_l\nabla_m\right]\left[r - \Phi^H(r,\xi)\right],\\
\Omega^R_{jl}(\mathbf{r}) &= -\epsilon_{jlm}\nabla_m\nabla^2\left[r - \Phi^E(r,\xi)\right] = -2\epsilon_{jlm}\nabla_m\frac{\operatorname{erfc}(\xi\mathbf{r})}{r},
\end{aligned}
$$

where the splitting functions Φ^H and Φ^E are found in the first and second lines of Table 1. This gives us

$$
\begin{aligned}
S^R_{jl}(\mathbf{r},\xi) &= 2\left(\frac{\xi e^{-\xi^2r^2}}{\sqrt{\pi}} + \frac{\operatorname{erfc}(\xi r)}{2r}\right)(\delta_{jl} + \hat{r}_j\hat{r}_l) - \frac{4\xi}{\sqrt{\pi}}e^{-\xi^2r^2}\delta_{jl},\\
T^R_{jlm}(\mathbf{r}) &= -\frac{2}{r}\left[\frac{3\operatorname{erfc}(\xi r)}{r} + \frac{2\xi}{\sqrt{\pi}}\left(3 + 2\xi^2r^2\right)e^{-\xi^2r^2}\right]\hat{r}_j\hat{r}_l\hat{r}_m\\
&\quad + \frac{4\xi^3}{\sqrt{\pi}}e^{-\xi^2r^2}(\delta_{jl}\hat{r}_m + \delta_{lm}\hat{r}_j + \delta_{mj}\hat{r}_l),\\
\Omega^R_{jl}(\mathbf{r}) &= 2\varepsilon_{jlm}\hat{r}_m\left(\frac{\operatorname{erfc}(\xi r)}{r^2} + \frac{2\xi}{\sqrt{\pi}}\frac{1}{r}e^{-\xi^2r^2}\right),
\end{aligned}
$$

where $\hat{\mathbf{r}} = \mathbf{r}/|\mathbf{r}|$. Only the stokeslet has a nonzero limit as given in (4), which must be included to remove the self-interaction when evaluating at a source point location.

Let us now turn to the Fourier space terms, using the screening approach with the Hasimoto (stokeslet, stresslet) and Ewald (rotlet) screening functions. From (18), for each Green's function using the specific form of the differential operator in (9,10,11) to define $\widehat{\mathrm{K}}(\mathbf{k})$, with $\widehat{\gamma}_E(\mathbf{k},\xi)$ and $\widehat{\gamma}_H(\mathbf{k},\xi)$ as given in the first and second line of Table 1, we obtain

$$
\begin{aligned}
\widehat{S}^F(\mathbf{k},\xi) &= A^S(\mathbf{k},\xi)e^{-k^2/4\xi^2},\\
\widehat{T}^F(\mathbf{k},\xi) &= A^T(\mathbf{k},\xi)e^{-k^2/4\xi^2},\\
\widehat{\Omega}^F(\mathbf{k},\xi) &= A^\Omega(\mathbf{k},\xi)e^{-k^2/4\xi^2},
\end{aligned}
$$

where

$$
A^S_{jl}(\mathbf{k},\xi) = -\left(k^2\delta_{jl} - k_jk_l\right)\left(1 + k^2/(4\xi^2)\right)\widehat{B}(|\mathbf{k}|), \tag{20}
$$

$$
A^T_{jlm}(\mathbf{k},\xi) = -i\left[(k_m\delta_{jl} + k_j\delta_{lm} + k_l\delta_{mj})k^2 - 2k_jk_lk_m\right]\left(1 + k^2/(4\xi^2)\right)\widehat{B}(|\mathbf{k}|), \tag{21}
$$

$$
A^\Omega_{jl}(\mathbf{k},\xi) = i\varepsilon_{jlm}k_mk^2\widehat{B}(|\mathbf{k}|) = 2i\varepsilon_{jlm}k_m\widehat{H}(|\mathbf{k}|). \tag{22}
$$

Here, $\widehat{H}(k)$ and $\widehat{B}(k)$ are the Fourier transforms of $H(r) = 1/r$ and $B(r) = r$,

$$
\widehat{H}(k) = \frac{4\pi}{k^2}, \quad \widehat{B}(k) = -\frac{8\pi}{k^4}. \tag{23}
$$

For a smooth compactly supported function $\widehat{G}^F(\mathbf{k},\xi)$, the Fourier integral in (19) can be approximated to spectral accuracy with a trapezoidal rule in each coordinate direction,

allowing for the use of FFTs for the evaluation. Inserting the definitions in (23) into (20–22), we can, however, note that the Fourier component has a singularity at $k = 0$ for all three Green's functions.

We will introduce modified Green's functions for the harmonic and biharmonic equations, which will still yield the exact same result as the original ones in the solution domain, and where the Fourier transforms of these functions have no singularity for $k = 0$. The necessary ideas will be introduced in the next section, following the recent work by Vico et al. [28].

4 Free-space solution of the harmonic and biharmonic equations

Consider the Poisson equation

$$-\Delta\varphi(\mathbf{x}) = 4\pi f(\mathbf{x})$$

with free-space boundary conditions ($\varphi \to 0$ as $|\mathbf{x}| \to \infty$). The solution is given by

$$\varphi(\mathbf{x}) = \int_{\mathbb{R}^3} H(|\mathbf{x}-\mathbf{y}|)f(\mathbf{y})\,d\mathbf{y} = \frac{1}{(2\pi)^3}\int_{\mathbb{R}^3} \widehat{H}(|\mathbf{k}|)\hat{f}(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{x}}\,\mathrm{d}\mathbf{k}, \tag{24}$$

where $H(r) = 1/r$ is the harmonic Green's function and $\widehat{H}(k) = 4\pi/k^2$ its Fourier transform. Note that they are both radial.

Assume now that f is compactly supported within a domain $\tilde{\mathcal{D}}$; a box with sides $\tilde{\mathbf{L}}$,

$$\tilde{\mathcal{D}} = \{\mathbf{x} \mid x_i \in [0, \tilde{L}_i]\},$$

and that we seek the solution $\varphi(\mathbf{x})$ for $\mathbf{x} \in \tilde{\mathcal{D}}$. The largest point-to-point distance in the domain is $|\tilde{\mathbf{L}}|$. Let $\mathcal{R} \geq |\tilde{\mathbf{L}}|$. Without changing the solution, we can then replace H with a truncated version,

$$H^{\mathcal{R}}(r) = H(r)\,\mathrm{rect}\left(\frac{r}{2\mathcal{R}}\right),$$

where

$$\mathrm{rect}(x) = \begin{cases} 1 & \text{for } |x| \leq 1/2, \\ 0 & \text{for } |x| > 1/2. \end{cases}$$

The Fourier transform of this truncated Green's function is [28]

$$\widehat{H}^{\mathcal{R}}(k) = 8\pi\left(\frac{\sin(\mathcal{R}k/2)}{k}\right)^2. \tag{25}$$

This function has a well-defined limit at $k = 0$,

$$\widehat{H}^{\mathcal{R}}(0) = \lim_{k\to 0}\widehat{H}^{\mathcal{R}}(k) = 2\pi\mathcal{R}^2.$$

Similarly, to solve the biharmonic equation on the same size domain, we can define $B^{\mathcal{R}}(r) = B(r)\,\mathrm{rect}\left(\frac{r}{2\mathcal{R}}\right)$, which has the Fourier transform [28]

$$\widehat{B}^{\mathcal{R}}(k) = 4\pi\frac{(2-\mathcal{R}^2k^2)\cos(\mathcal{R}k) + 2\mathcal{R}k\sin(\mathcal{R}k) - 2}{k^4}, \tag{26}$$

with the limit value

$$\widehat{B}^{\mathcal{R}}(0) = \lim_{k\to 0} \widehat{B}^{\mathcal{R}}(k) = \pi\mathcal{R}^4.$$

4.1 Solving the harmonic and biharmonic equations using FFTs

We will now describe how to solve the harmonic and biharmonic equations using FFTs. In doing so, we will for simplicity of notation assume that the domain is a cube, i.e., $\tilde{L}_1 = \tilde{L}_2 = \tilde{L}_3 = \tilde{L}$. The steps are as follows:

1. Introduce a grid of size $\tilde{M}^3$ with grid size $h = \tilde{L}/\tilde{M}$ and evaluate $f(\mathbf{x})$ on that grid.
2. Define an oversampling factor s_f, and zero-pad (described in the subsequent section) to do a 3D FFT of size $(s_f\tilde{M})^3$, defining $\hat{f}(\mathbf{k})$ for

$$\mathbf{k} = \frac{2\pi}{\tilde{L}}\frac{1}{s_f}(k_1, k_2, k_3), \qquad k_i \in \left\{-\frac{s_f\tilde{M}}{2}, \ldots, \frac{s_f\tilde{M}}{2} - 1\right\}.$$

3. Set $\mathcal{R} = \sqrt{3}\tilde{L}$ and evaluate $\widehat{H}^{\mathcal{R}}(k)$ (25), with $k = |\mathbf{k}|$ for the set of $\mathbf{k}$-vectors defined above.
4. Multiply $\hat{f}(\mathbf{k})$ and $\widehat{H}^{\mathcal{R}}(k)$ for each $\mathbf{k}$. Do a 3D IFFT and truncate the result to keep the $\tilde{M}^3$ values defining the approximation of the solution $\varphi(\mathbf{x})$ on the grid.

To solve the biharmonic equation instead, replace $\widehat{H}^{\mathcal{R}}(k)$ by $\widehat{B}^{\mathcal{R}}(k)$ as given in (26).

With this, we have for all $\mathbf{x}_j$ in the grid computed the approximation

$$\varphi(\mathbf{x}_j) \approx \frac{(\Delta k)^3}{(2\pi)^3} \sum_{k_1,k_2,k_3=-s_f\frac{\tilde{M}}{2}}^{s_f\frac{\tilde{M}}{2}-1} \widehat{H}^{\mathcal{R}}(|\mathbf{k}|)\hat{f}(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{x}_j},$$

where $\Delta k = \frac{2\pi}{\tilde{L}}\frac{1}{s_f}$.

Note that we at no occasion explicitly multiply with a prefactor, assuming there is a built-in scaling of $1/(s_f\tilde{M})^3$ in the 3D inverse FFT. There should be a multiplication with h^3 in step 2, and with $(\Delta k/2\pi)^3$ above, but that cancels such that only the built-in scaling remains.

Since the convolution is aperiodic, we need to oversample by at least a factor of two. In Vico et al. [28], they advise that we need an additional factor of two to resolve the oscillatory behavior of the Fourier transform of the truncated kernel, which would yield $s_f = 4$. It does, however, turn out that the need of oversampling is less than this, as we will discuss in the next section. If we oversample sufficiently, the error will decay spectrally with $\tilde{M}$ given that the right-hand side f is smooth.

4.2 Zero-padding/oversampling

Consider the first integral in (24). With f compactly supported on a cube with size $\tilde{L}^3$, H must be defined on a cube with size $(2\tilde{L})^3$ to be able to compute the convolution. $H^{\mathcal{R}}$, with $\mathcal{R} = |\tilde{\mathbf{L}}| = \sqrt{3}\tilde{L}$ coincides with H inside the sphere of radius $\mathcal{R}$, the smallest sphere with the cube inscribed. When we use the FFT, we "periodize" the computations. We hence need to zero-pad the data so that this periodization interval is large enough to make sure that $H^{\mathcal{R}}$ is not polluted within the cube of size $(2\tilde{L})^3$.

Assume that we zero-pad the data up to a domain size of $2\tilde{L} + \delta$. If the sphere of radius $\mathcal{R}$ is to fit within this domain, we would have $\delta = 2(\mathcal{R} - \tilde{L})$. However, as is illustrated in Fig. 1, since it is enough that $H^{\mathcal{R}}$ is not polluted within the cube, it is sufficient with $\delta = \mathcal{R} - \tilde{L}$. In terms of an oversampling factor s_f, this corresponds to

$$s_f \tilde{L} \geq 2\tilde{L} + \delta = \tilde{L} + \mathcal{R},$$

and the necessary condition becomes

$$s_f \geq \frac{\tilde{L} + \mathcal{R}}{\tilde{L}},$$

such that with $\mathcal{R} = \sqrt{3}\tilde{L}$, we get $s_f \geq 1 + \sqrt{3} \approx 2.8$. Note that an argument based instead on a large enough sampling ratio in the Fourier domain to resolve the oscillatory truncated Green's function would yield the smallest oversampling rate to be that with $\delta = 2(\mathcal{R} - \tilde{L})$, where the Green's function is without pollution in the full sphere, and hence $s_f \geq 2\sqrt{3} \approx 3.5$.

For non-cubic domains, we will have a larger oversampling requirement,

$$s_f \geq 1 + \frac{\mathcal{R}}{\min_i \tilde{L}_i}, \tag{27}$$

which is $s_f \geq 1 + |\widetilde{\mathbf{L}}|/(\min_i \tilde{L}_i)$ with the smallest possible $\mathcal{R}$. This additional cost can, however, be limited to a precomputation step, through the scheme suggested in [28] as discussed in the next section.

4.3 Precomputation

We will now further discuss step 4 in the algorithm introduced in Sect. 4.1. For ease of notation, we do so in one dimension. Each dimension will be treated the same way, so the extension is simple to make. Let $M_g = s_f \tilde{M}$ be the number of grid points, such that $h = \tilde{L}/\tilde{M} = (s_f \tilde{L})/M_g$, and let $k = (2\pi/(s_f \tilde{L}))\tilde{k}$, where $\tilde{k}$ is an integer. By the means of an IFFT, we can compute

$$\varphi_j = \frac{1}{M_g} \sum_{\tilde{k}=-M_g/2}^{M_g/2-1} \widehat{G}(k) \hat{f}_{\tilde{k}} e^{i\frac{2\pi}{M_g}\tilde{k}j}, \qquad j = 0, \ldots, M_g - 1, \tag{28}$$

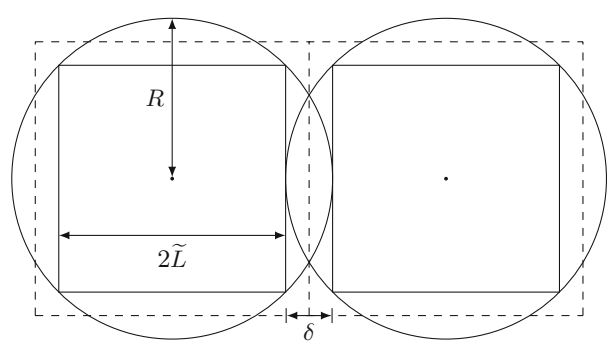

Fig. 1 Illustration of the minimum zero-padding δ required to accurately represent the $\mathcal{R}$-truncated Green's function inside the domain of dimensions $2\widetilde{\mathbf{L}}$, when using a periodic Fourier transform. The condition $\delta \geq \mathcal{R} - \tilde{L}$ must be satisfied to avoid pollution from neighboring Green's functions inside the domain of interest

where $\widehat{G}(k)$ could be either $\widehat{H}^{\mathcal{R}}(k)$ or $\widehat{B}^{\mathcal{R}}(k)$, and the Fourier coefficients $\hat{f}_{\tilde{k}}$ have been computed by an FFT,

$$\hat{f}_{\tilde{k}} = \sum_{l=0}^{M_g-1} f(lh) e^{-i\frac{2\pi}{M_g}\tilde{k}l}. \tag{29}$$

Inserting (29) into (28), and rearranging the order of the sums, we get

$$\varphi_j = \sum_{l=0}^{M_g-1} \left[\frac{1}{M_g} \sum_{\tilde{k}=-M_g/2}^{M_g/2-1} \widehat{G}(k) e^{i\frac{2\pi}{M_g}\tilde{k}(j-l)} \right] f(lh) = \sum_{l=0}^{M_g-1} G_{j-l} f(lh),$$

where G_{j-l}, $j = 0, \ldots, M_g - 1$, will define the effective Green's function on the grid, centered at grid point l. Note here that f has compact support and $f(lh) = 0$ for $l > \tilde{M} - 1$, and even though φ_j is computed on the large grid, we will truncate and keep only the $\tilde{M}$ first values. Hence, for each l, only $\tilde{M}$ values of G_{j-l} are actually needed to produce our result, and since $G_{(j+1)-(l+1)} = G_{j-l}$ a total of $2\tilde{M}$ grid values of the Green's function are used in the calculation. Hence, one can without knowing f precompute an effective Green's function on a grid using the oversampling rate $s_f \geq 1 + \mathcal{R}/\tilde{L}$ derived in the previous section, and truncate it to the $2\tilde{M}$ values centered around $r = 0$. Let us denote by $\tilde{G}$ the *mollified Green's function* that is the result of this procedure. Since we carry out the aperiodic convolution using FFTs, what we actually need to precompute is $\widehat{\tilde{G}}$, the Fourier transform of the mollified Green's function. In 3D, the steps for precomputing this are as follows:

1. Evaluate $\hat{G}$ on a grid of size $(s_f\tilde{M})^3$ and do a 3D IFFT to get $\tilde{G}$.
2. Truncate $\tilde{G}$ to the $(2\tilde{M})^3$ points around the center.
3. Do a 3D FFT to get $\widehat{\tilde{G}}$.

An example of the mollified harmonic Green's function computed in this way is in Fig. 2 shown for $G = H^{\mathcal{R}}$.

Once f is given, we can now compute φ using an aperiodic convolution, which in practice is evaluated through an FFT with an oversampling factor of 2. This requires the following steps:

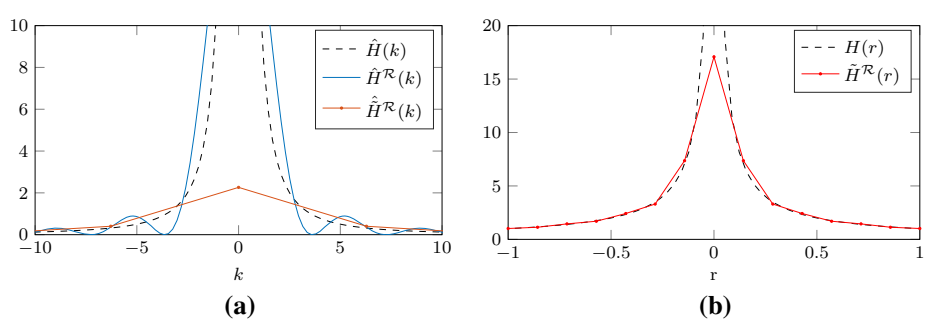

Fig. 2 Example of the mollified harmonic Green's function generated by inverse transform of $\hat{H}^{\mathcal{R}}$ using a finite-size IFFT. **a** Fourier space representations of the original (23), truncated (25) and mollified harmonic Green's functions. The latter is computed using an FFT. **b** The harmonic Green's function and its mollified counterpart

1. Zero-pad f to size $(2\tilde{M})^3$ and do a 3D FFT to get $\hat{f}$.
2. Do a 3D IFFT of $\widehat{\tilde{G}}\hat{f}$ and truncate to the $(\tilde{M})^3$ values that correspond to the original domain.

This is beneficial when we want to solve the equation using several right-hand sides f, since we only have to evaluate FFTs using the larger oversampling rate (27) in the precomputation step, while subsequent FFTs only require an oversampling rate of 2.

5 Evaluating the Fourier space component

Let us now go back to the Fourier space component in the Ewald decomposition (19). We will use the notation

$$\widehat{G}^{F,\mathcal{R}}(\mathbf{k},\xi) = A^{G,\mathcal{R}}(\mathbf{k},\xi)e^{-k^2/4\xi^2}, \tag{30}$$

where $G = S$, T, and Ω, and where the superindex $\mathcal{R}$ indicates that $\widehat{H}(k)$ and $\widehat{B}(k)$ are replaced by $\widehat{H}^{\mathcal{R}}(k)$ and $\widehat{B}^{\mathcal{R}}(k)$ in the definitions (20), (21) and (22). This means that the modified Green's functions $\widehat{S}^{F,\mathcal{R}}$, $\widehat{T}^{F,\mathcal{R}}$ and $\widehat{\Omega}^{F,\mathcal{R}}$ will have no singularity at $k = 0$.

The task is now to compute

$$\mathbf{u}^F(\mathbf{x},\xi) = \frac{1}{(2\pi)^3}\int_{\mathbb{R}^3} e^{i\mathbf{k}\cdot\mathbf{x}}e^{-k^2/4\xi^2}A^{G,\mathcal{R}}(\mathbf{k},\xi)\cdot\sum_{n=1}^{N}\mathbf{f}(\mathbf{x}_\mathrm{n})e^{-i\mathbf{k}\cdot\mathbf{x}_\mathrm{n}}\,\mathrm{d}\mathbf{k} \tag{31}$$

for a given set of target points. The integrand of the inverse transform is now smooth and can after truncation be easily evaluated using the trapezoidal rule, but evaluation is still costly—$\mathcal{O}(N^2)$ if evaluating at N target points. We will now outline the spectral Ewald method, which uses the fast Fourier transform to reduce the cost of this evaluation, yielding a method with a total cost (including the real space sum) of $\mathcal{O}(N\log N)$. Before we discuss the actual discretization and implementation details, we start by describing the mathematical foundation of the method.

5.1 Foundations

First, we introduce a scalar parameter $\eta > 0$ and define

$$\widehat{\mathbf{g}}(\mathbf{k},\xi,\eta) = \sum_{n=1}^{N}\mathbf{f}(\mathbf{x}_\mathrm{n})e^{-i\mathbf{k}\cdot\mathbf{x}_\mathrm{n}}e^{-\eta k^2/8\xi^2}, \tag{32}$$

which is the Fourier transform of the smooth function

$$\mathbf{g}(\mathbf{x},\xi,\eta) = \sum_{n=1}^{N}\mathbf{f}(\mathbf{x}_\mathrm{n})\left(\frac{2\xi^2}{\pi\eta}\right)^{3/2}e^{-2\xi^2|\mathbf{x}-\mathbf{x}_\mathrm{n}|^2/\eta}. \tag{33}$$

Hence, instead of using (32) to directly evaluate $\widehat{\mathbf{g}}(\mathbf{k},\xi,\eta)$, we can evaluate $\mathbf{g}(\mathbf{x},\xi,\eta)$ and use an actual computation of the Fourier transform,

$$\widehat{\mathbf{g}}(\mathbf{k},\xi,\eta) = \int_{\mathbb{R}^3}\mathbf{g}(\mathbf{x},\xi,\eta)e^{-i\mathbf{k}\cdot\mathbf{x}}\,\mathrm{d}\mathbf{x}.$$

We furthermore define

$$\widehat{\mathbf{w}}(\mathbf{k},\xi,\eta) = e^{-(1-\eta)k^2/4\xi^2} A^{G,\mathcal{R}}(\mathbf{k},\xi)\cdot\widehat{\mathbf{g}}(\mathbf{k},\xi,\eta), \tag{34}$$

such that (31) can be written

$$\mathbf{u}^F(\mathbf{x},\xi) = \frac{1}{(2\pi)^3}\int_{\mathbb{R}^3} \widehat{\mathbf{w}}(\mathbf{k},\xi,\eta) e^{-\eta k^2/8\xi^2} e^{i\mathbf{k}\cdot\mathbf{x}}\,\mathrm{d}\mathbf{k}.$$

Using the convolution theorem, we can write this as

$$\mathbf{u}^F(\mathbf{x},\xi) = \int_{\mathbb{R}^3} \mathbf{w}(\mathbf{y},\xi,\eta)\left(\frac{2\xi^2}{\pi\eta}\right)^{3/2} e^{-2\xi^2|\mathbf{x}-\mathbf{y}|^2/\eta}\,\mathrm{d}\mathbf{y}, \tag{35}$$

where

$$\mathbf{w}(\mathbf{x},\xi,\eta) = \frac{1}{(2\pi)^3}\int_{\mathbb{R}^3} \widehat{\mathbf{w}}(\mathbf{k},\xi,\eta) e^{i\mathbf{k}\cdot\mathbf{x}}\,\mathrm{d}\mathbf{k}.$$

5.2 Discretization

Assume that we are to evaluate (31) for $\mathbf{x} = \mathbf{x}_{\mathtt{m}}$, $\mathtt{m} = 1,\ldots,N$, and for simplicity of notation that all points are contained in a cube with equal sides L,

$$\mathbf{x}_{\mathtt{n}} \in \mathcal{D} = [0,L]^3, \quad \mathtt{n} = 1,\ldots,N.$$

The choice of η will be discussed shortly, in Sect. 5.3. At this point, assume that the Gaussians $e^{-2\xi^2|\cdot|^2/\eta}$ in (33) and (35) decay rapidly and will be truncated outside a diameter d. Then, $\mathbf{g}$ becomes compactly supported, such that we can compute $\widehat{\mathbf{g}}$ using an FFT, and the integral in (35) becomes a local operation around each target point $\mathbf{x}_{\mathtt{m}}$. To accommodate the support of the truncated Gaussians, we must extend the domain by some length δ_L. We will discuss the choice of this length in the discussion on η. For now, we consider the extended domain with sides $\tilde{L} = L + \delta_L$,

$$\tilde{\mathcal{D}} = [-\delta_L/2, L+\delta_L/2]^3.$$

This domain is discretized using a uniform grid with $\tilde{M}^3$ points and grid spacing $h = \tilde{L}/\tilde{M}$.

To initialize our calculations, we precompute $\widehat{H}^{\mathcal{R}}(k)$ in case of the rotlet, and $\widehat{B}^{\mathcal{R}}(k)$ in case of the stokeslet or stresslet, as described in Sect. 4.3. They need to be precomputed on a domain of size $2\tilde{L}$, with $\mathcal{R} = \sqrt{3}\tilde{L}$.

The first step of our computations is to evaluate $\mathbf{g}$ on the grid as in (33). After that we zero-pad the FFT by a factor of 2, to have an oversampled representation of $\widehat{\mathbf{g}}$, before we scale it to define $\widehat{\mathbf{w}}$ as in (34). We will then multiply by the precomputed fundamental solution ($\widehat{H}^{\mathcal{R}}(k)$ or $\widehat{B}^{\mathcal{R}}(k)$) and the additional scaling factors as given in (20), (21) and (22), and apply an inverse FFT to perform a discrete convolution.

The computation of $\mathbf{u}^F(\mathbf{x}_{\mathtt{m}},\xi)$, $\mathtt{m} = 1,\ldots,N$, can hence be broken down into the following steps:

1. *Spreading* Compute $\mathbf{g}$ on the grid using (33) and truncated Gaussians.
2. *FFT* Compute $\widehat{\mathbf{g}}$ using the three-dimensional FFT, zero-padded to the double size.
3. *Scaling* Compute $\widehat{\mathbf{w}}$ using (34) and precomputed $\widehat{H}^{\mathcal{R}}(k)$ or $\widehat{B}^{\mathcal{R}}(k)$.

4. *IFFT* Apply the inverse three-dimensional FFT to $\widehat{\mathbf{w}}$. Truncate the result to have $\mathbf{w}$ defined on the original grid.
5. *Quadrature* For each $\mathbf{x}_{\mathtt{m}}$, $\mathtt{m} = 1, \ldots, N$, evaluate $\mathbf{u}^F$ using (35) and the trapezoidal rule, with the Gaussian truncated outside the sphere of diameter d centered at $\mathbf{x}_{\mathtt{m}}$.

This is the spectral Ewald method. A major cost of the method is the large number of exponential function evaluations in steps 1 and 5. This can be accelerated through the method of fast Gaussian gridding (FGG) [14,22]. It is then natural to truncate the Gaussians outside a cube of P^3 grid points, in which case

$$d = hP.$$

The computational cost of the FGG in steps 1 and 5 is then $\mathcal{O}(NP^3)$, while the cost of the FFTs in steps 2 and 4 is $\mathcal{O}(\tilde{M}^3 \log \tilde{M})$. The cost of the scaling in step 3 is $\mathcal{O}(\tilde{M}^3)$, and negligible in this context.

5.3 Errors in the spectral Ewald method

The use of the spectral Ewald method for computing the Fourier space component introduces approximation errors in the solution, which are separate from the Fourier integral truncation error (further discussed in Sect. 7.1). The approximation errors stem from the use of a discrete quadrature rule in the quadrature step and from the truncation and discretization of the Gaussians $e^{-2\xi^2|\cdot|^2/\eta}$ in the spreading and quadrature steps. The Ewald parameter ξ should be regarded as free, since it is used for work and error balancing between the real and Fourier space sums (more on this in Sect. 8.2). This leaves two variables for controlling the approximation errors: the scalar parameter η and the Gaussian truncation width $d = hP$. Following [22], we write η as

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

where m is a shape parameter controlling how fast the Gaussian decays within the support d. It can be shown [22] that the approximation errors decay exponentially in P with the choice $m(P) = C\sqrt{\pi P}$ and that the constant C should be taken slightly below unity for optimal results (we use the value $C = 0.976$ suggested in [22]). With these choices, the approximation errors of the method are controlled through a single parameter P, and they furthermore decay exponentially in that parameter.

It is evident from the algorithm that δ_L must be chosen such that the support of the truncated Gaussians in (33) and (35) is included, i.e., $\delta_L \geq d$. However, it turns out that this is not always enough. In the spectral Ewald method, we have taken the Gaussian $e^{-\xi^2 r^2}$ of the screening function (Table 1) and separated it into a series of convolutions of Gaussians, through the factorization

$$e^{-k^2/4\xi^2} = e^{-\eta k^2/8\xi^2} \cdot e^{-(1-\eta)k^2/4\xi^2} \cdot e^{-\eta k^2/8\xi^2}.$$

The first and last factors correspond to the Gaussian $e^{-2\xi^2 r^2/\eta}$ in the gridding and quadrature steps and are already properly resolved and truncated by our choices of η and d. For $\eta \geq 1$, the entire Gaussian $e^{-\xi^2 r^2}$ is contained in these two factors, and the middle factor can be viewed as a deconvolution of the type used in the non-uniform FFT [20]. However,

for $\eta < 1$ the middle factor represents the Gaussian $e^{-\xi^2 r^2/(1-\eta)}$, and (34) corresponds to a convolution with that Gaussian, carried out in Fourier space. For the convolution to be properly represented, we must make sure that the domain $\tilde{L}$ includes the support of $e^{-\xi^2 r^2/(1-\eta)}$ to the desired truncation level. The original Gaussians are truncated at the level $e^{-2\xi^2(d/2)^2/\eta} = e^{-m^2/2}$. For the remainder Gaussians to be truncated at the same level, we need that

$$e^{-\xi^2(\delta_L/2)^2/(1-\eta)} \leq e^{-m^2/2},$$

i.e.,

$$\delta_L \geq \sqrt{2(1-\eta)m^2/\xi^2}.$$

To guarantee that both Gaussians have proper support, we thus need

$$\delta_L \geq \begin{cases} d & \text{if } \eta \geq 1, \\ \max\left(d, \sqrt{2(1-\eta)m^2/\xi^2}\right) & \text{if } \eta < 1. \end{cases} \tag{36}$$

With this extra support for $\eta < 1$, the approximation errors are decoupled from the Fourier space truncation errors, which are further discussed in Sect. 7.1. An example of this decoupling is shown in Fig. 3, where it can be seen that the larger choice of δ_L is actually only needed if the grid size $\tilde{M}$ is picked larger than necessary for a given error tolerance.

6 Evaluating the real space component

The real space part of the free-space Ewald sum (19) has the general form

$$\mathbf{u}^R(\mathbf{x}) = \sum_{\mathrm{n}=1}^{N} G^R(\mathbf{x} - \mathbf{x}_\mathrm{n}) \cdot \mathbf{f}(\mathbf{x}_\mathrm{n}).$$

Since $G^R(r)$ decays rapidly (roughly as $e^{-\xi^2 r^2}$), the sum can be truncated outside some truncation radius r_c. Assume that we wish to evaluate the potential at points $\mathbf{x}_\mathrm{m}$, $\mathrm{m} = 1, \ldots, N$. The expression that we need to evaluate is then

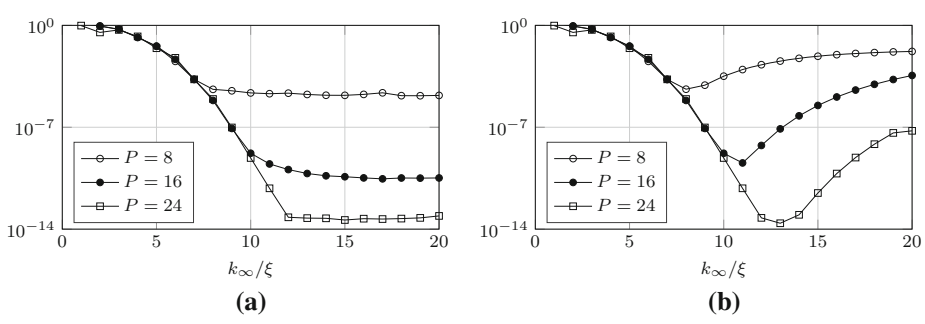

Fig. 3 Error in the stokeslet Fourier space component for various values of the discrete Gaussian support P. The system is a unit cube with 1000 sources, $\tilde{M} \in [2, 40]$, $\xi = 2\pi$ and $k_\infty = \pi\tilde{M}/L = \pi\tilde{M}$. To the *left* ,the domain is extended to include the support of the remainder Gaussians, while to the *right* the domain is only extended to cover the support of the gridding and quadrature Gaussians. Evidently, the extra support is only needed if $\tilde{M}$ is picked larger than necessary for a given error tolerance. **a** δ_L set through (36). **b** $\delta_L = d$

$$\mathbf{u}^R(\mathbf{x}_\mathrm{m}) = \sum_{\substack{\mathrm{n}=1 \\ |\mathbf{x}_\mathrm{m}-\mathbf{x}_\mathrm{n}|\le r_c}}^{N} G^R(\mathbf{x}_\mathrm{m}-\mathbf{x}_\mathrm{n})\cdot \mathbf{f}(\mathbf{x}_\mathrm{n}), \quad \mathrm{m}=1,\ldots,N.$$

Naively implemented, this has an $\mathcal{O}(N^2)$ computational cost. It is, however, straightforward to find the interaction list of each target point $\mathbf{x}_\mathrm{m}$ by first creating a cell list [11]. This reduces the real space cost to $\mathcal{O}(N)$, under the assumption that the average number of interactions of each target point stays constant when N changes.

7 Truncation errors

Truncation errors are introduced when we cut off the real space interactions outside a radius r_c, and when we truncate the Fourier space integral outside a maximum wave number k_∞. The magnitudes of these errors can be accurately estimated through the analysis methodology introduced by Kolafa and Perram [19] for periodic electrostatic force computations. Denoting by $\tilde{\mathbf{u}}(\mathbf{x})$ the truncated solution, one can then derive statistical error estimates for the root mean square (RMS) truncation error, defined as

$$\delta\mathbf{u} = \sqrt{\frac{1}{N}\sum_{\mathrm{n}=1}^{N}\left|\mathbf{u}(\mathbf{x}_\mathrm{n})-\tilde{\mathbf{u}}(\mathbf{x}_\mathrm{n})\right|^2}.$$

The analyses for both the real and Fourier space components rely on the following property:

Lemma 1 (Kolafa and Perram [19, appx.A]) *Let* $(\mathbf{x}_n, q_n)$ *be a configuration of point sources, and let*

$$E(\mathbf{x}) = \sum_{n=1}^{N} q_n\Big(f(\mathbf{x}-\mathbf{x}_n) - \tilde{f}(\mathbf{x}-\mathbf{x}_n)\Big),$$

be an error measure due to a set of pointwise errors. Assuming that the points are randomly distributed, and that E has a Gaussian distribution, the root mean square (RMS) error

$$\delta E = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\big(E(\mathbf{x}_n)\big)^2}$$

can be approximated as

$$\delta E^2 \approx \frac{1}{|V|}\sum_i q_i^2 \int_V \Big(f(\mathbf{r}) - \tilde{f}(\mathbf{r})\Big)^2 \,\mathrm{d}\mathbf{r},$$

where V is the volume enclosing all point-to-point vectors $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$.

7.1 Fourier space truncation error

The Fourier space error comes from truncating the integral of the Fourier transform outside a maximum wave number k_∞,

$$\mathbf{u}^F(\mathbf{x}) - \tilde{\mathbf{u}}^F(\mathbf{x}) = \frac{1}{(2\pi)^3}\int_{k>k_\infty} \widehat{G}^F(\mathbf{k},\xi)\cdot\sum_{\mathrm{n}=1}^{N}\mathbf{f}(\mathbf{x}_\mathrm{n})e^{i\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}_\mathrm{n})}\,\mathrm{d}\mathbf{k},$$

where all points $\mathbf{x}_\mathrm{n}$ are contained in a cube of size L. In our case, the integral is approximated using an FFT over an M^3 grid covering an L^3 domain, such that

$$k_\infty = \frac{2\pi}{\tilde{L}}\frac{\tilde{M}}{2}.$$

The RMS of the truncation error is given by

$$\delta\mathbf{u}^F = \sqrt{\frac{1}{N}\sum_{\mathrm{n}=1}^{N}\left|\mathbf{u}^F(\mathbf{x}_\mathrm{n}) - \tilde{\mathbf{u}}^F(\mathbf{x}_\mathrm{n})\right|^2},$$

and can be estimated using the method of Kolafa &Perram. Such estimates already exist for the periodic stokeslet [21] and rotlet [1] potentials, as well as for the Beenakker decomposition of the stresslet [3]. However, it turns out that periodic estimates fail for free-space potentials that are based on the truncated biharmonic potential $B^\mathcal{R}$. This is because the dominating term of $\widehat{B}^\mathcal{R}$ for large k is a factor $\mathcal{R}^2k^2/2$ larger than $\widehat{B}$. For potentials based on the truncated harmonic potential $H^\mathcal{R}$, one can use the periodic estimates, as the difference in magnitude between $\widehat{H}^\mathcal{R}$ and $\widehat{H}$ is negligible. We can thus use the existing estimates for the rotlet available in [1], while we need to derive new ones for the stokeslet and stresslet. The final set of estimates is shown in Table 2.

7.1.1 Stokeslet

Beginning with the stokeslet potential, we consider the truncation error contribution from a single source located at the origin,

$$u_j^F(\mathbf{x}) - \tilde{u}_j^F(\mathbf{x}) = e_{jl}(\mathbf{x})f_l,$$

where

$$e_{jl}(\mathbf{r}) = \frac{1}{(2\pi)^3}\int_{k>k_\infty}\left(1+\frac{k^2}{4\xi^2}\right)\widehat{B}^\mathcal{R}(\mathbf{k})e^{-(k/2\xi)^2}e^{i\mathbf{k}\cdot\mathbf{r}}k^2\left(\delta_{jl}-\hat{k}_j\hat{k}_l\right)\mathrm{d}\mathbf{k},$$

and

$$\widehat{B}^\mathcal{R}(\mathbf{k}) = 4\pi\frac{(2-\mathcal{R}^2k^2)\cos(\mathcal{R}k)+2\mathcal{R}k\sin(Rk)-2}{k^4}.$$

We now keep only the highest order term in k, which dominates the error for large k_∞,

$$e_{jl} \approx -\frac{4\pi\mathcal{R}^2}{4\xi^2(2\pi)^3}\int_{k>k_\infty}k^2\cos(\mathcal{R}k)e^{-(k/2\xi)^2}e^{i\mathbf{k}\cdot\mathbf{r}}\left(\delta_{jl}-\hat{k}_j\hat{k}_l\right)\mathrm{d}\mathbf{k}.$$

Table 2 Fourier space truncation error estimates for the stokeslet, stresslet and rotlet [1]

	Stokeslet, S^F	*Stresslet, T^F*	*Rotlet, Ω^F*
$\delta\mathbf{u}^F$	$\sqrt{Q}\frac{\mathcal{R}k_\infty^3}{\xi^2\pi L}e^{-k_\infty^2/4\xi^2}$	$\sqrt{\frac{7Q}{6}}\frac{\mathcal{R}k_\infty^4}{\xi^2\pi L}e^{-k_\infty^2/4\xi^2}$	$\sqrt{\frac{8\xi^2Q}{3\pi L^3k_\infty}}e^{-k_\infty^2/4\xi^2}$

The quantity Q is defined as in (37) for the stokeslet and rotlet, and as in (38) for the stresslet

The error should be independent of the coordinate system orientation, and since we are deriving a statistical error measure, we approximate the directional component by its root mean square (computed using spherical coordinates),

$$\left(\delta_{jl} - \hat{k}_j\hat{k}_l\right) \approx \sqrt{\frac{1}{9}\sum_{j,l=1}^{3}\left(\delta_{jl} - \hat{k}_j\hat{k}_l\right)^2} = \frac{\sqrt{2}}{3}.$$

We now integrate in spherical coordinates, choosing a system such that $\hat{k}_3$ is parallel to $\hat{z}$. Integration in $\theta \in [0, \pi]$ then gives us

$$e_{jl} \approx -\frac{\sqrt{2}}{3}\frac{\mathcal{R}^2}{2\xi^2\pi r}\int_{k>k_\infty} k^3\cos(\mathcal{R}k)\sin(rk)e^{-(k/2\xi)^2}\,\mathrm{d}k.$$

To get a good approximation of e_{jl}, we need to approximate the remaining integral. First, we have from the exponential decay that the dominating contribution will come from the beginning of the interval, where $k \approx k_\infty$. This allows us to approximate

$$I = \int_{k>k_\infty} k^3\cos(\mathcal{R}k)\sin(rk)e^{-(k/2\xi)^2}\,\mathrm{d}k \approx k_\infty^3\int_{k>k_\infty}\cos(\mathcal{R}k)\sin(rk)e^{-(k/2\xi)^2}\,\mathrm{d}k.$$

Next, we have that $\mathcal{R} \gg r$, so $\mathcal{R}$ is the dominating frequency in the integrand, such that we can write $\cos(\mathcal{R}k)\sin(rk) \approx \cos(\mathcal{R}k)\sin(rk_\infty)$, and

$$I \approx k_\infty^3\sin(k_\infty r)\int_{k>k_\infty} e^{i\mathcal{R}k-(k/2\xi)^2}\,\mathrm{d}k,$$

where we implicitly assume that the real part of the complex exponential is our quantity of interest. A final approximation (again assuming $k \approx k_\infty$) makes this integrable, and we get

$$\begin{aligned}|I| &\approx \left|\frac{k_\infty^3\sin(k_\infty r)}{i\mathcal{R} - k_\infty/2\xi^2}\int_{k>k_\infty}(i\mathcal{R} - k/2\xi^2)e^{iRk-(k/2\xi)^2}\,\mathrm{d}k\right| \\ &= \left|\frac{k_\infty^3\sin(k_\infty r)}{i\mathcal{R} - K/2\xi^2}e^{i\mathcal{R}k_\infty-(k_\infty/2\xi)^2}\right| \le \frac{k_\infty^3|\sin(k_\infty r)|}{|i\mathcal{R} - k_\infty/2\xi^2|}e^{-(k_\infty/2\xi)^2}.\end{aligned}$$

We have (for a cube) that $\mathcal{R} \ge \sqrt{3}\tilde{M}h$, while $k_\infty = \pi\tilde{M}/\tilde{L} = \pi/h$. Typical parameter values are around $k_\infty/\xi = \mathcal{O}(10)$ and $\tilde{M} = \mathcal{O}(50)$, so $k_\infty/2\xi^2 = \mathcal{O}(50h/\pi) \ll \mathcal{R}$ and $|i\mathcal{R} - k_\infty/2\xi^2| \approx \mathcal{R}$. We can therefore write

$$|I| \approx \frac{k_\infty^3|\sin(k_\infty r)|}{\mathcal{R}}e^{-(k_\infty/2\xi)^2},$$

and

$$|e_{jl}| \approx \frac{\sqrt{2}}{6}\frac{\mathcal{R}k_\infty^3|\sin(k_\infty r)|}{\xi^2\pi r}e^{-(k_\infty/2\xi)^2}.$$

We can now use Lemma 1 to estimate the statistical error by integrating over a sphere of radius $L/2$ (which then contains all point sources),

$$\left(\delta \mathbf{u}^F\right)^2 \approx \sum_{\mathrm{n}=1}^{N} \sum_{j=1}^{3} f_l^2(\mathbf{x}_\mathrm{n}) \frac{1}{|V|} \int_V e_{jl}^2(r)\, \mathrm{d}\mathbf{r}$$
$$\approx Q \frac{6}{\pi L^3} 3 \left(\frac{\sqrt{2}}{6} \frac{\mathcal{R} k_\infty^3}{\xi^2 \pi} e^{-(k_\infty/2\xi)^2} \right)^2 \int_0^{L/2} \sin^2(k_\infty r) 4\pi \, \mathrm{d}r,$$

where

$$Q = \sum_{\mathrm{n}=1}^{N} |\mathbf{f}(\mathbf{x}_\mathrm{n})|^2. \tag{37}$$

Assuming that $\sin^2(k_\infty r)$ has many oscillations in the interval $[0, L/2]$, we replace it by its average value $1/2$, such that $\int_0^{L/2} \sin^2(k_\infty r) 4\pi \, \mathrm{d}r \approx \pi L$. Finally, we can write the stokeslet truncation error estimate as

$$\delta \mathbf{u}^F \approx \sqrt{Q} \frac{\mathcal{R} k_\infty^3}{\xi^2 \pi L} e^{-(k_\infty/2\xi)^2}.$$

7.1.2 Stresslet

For the stresslet, the derivation for the error estimate is completely analogous to the one for the stokeslet. The difference is that the leading order term is k^3 instead of k^2 and that the RMS of the directional component is

$$\sqrt{\frac{1}{27} \sum_{j,l,m=1}^{3} \left(\left(\delta_{jl} \hat{k}_m + \delta_{lm} \hat{k}_j + \delta_{mj} \hat{k}_l \right) - 2 \hat{k}_j \hat{k}_l \hat{k}_m \right)^2} = \frac{7}{27}.$$

This allows us to directly write the stresslet truncation error estimate as

$$\delta \mathbf{u}^F \approx \sqrt{\frac{7Q}{6}} \frac{\mathcal{R} k_\infty^4}{\xi^2 \pi L} e^{-(k_\infty/2\xi)^2},$$

where

$$Q = \sum_{\mathrm{n}=1}^{N} \sum_{l,m=1}^{3} q_l^2(\mathbf{x}_\mathrm{n}) n_m^2(\mathbf{x}_\mathrm{n}), \tag{38}$$

and $\mathbf{q}$ and $\mathbf{n}$ are as in (12).

The Fourier space truncation error estimates for the stokeslet, stresslet and rotlet are summarized in Table 2. The close match between the estimates and the actual measured error is shown in Fig. 4.

7.2 Real space truncation error

The real space truncation error is due to neglecting interactions in the real space sum for which $r > r_c$,

$$\mathbf{u}^R(\mathbf{x}) - \tilde{\mathbf{u}}^R(\mathbf{x}) = \sum_{|\mathbf{x}-\mathbf{x}_\mathrm{n}|>r_c} G^R(\mathbf{x}-\mathbf{x}_\mathrm{n}) \cdot \mathbf{f}(\mathbf{x}_\mathrm{n}).$$

The RMS of this error is given by

$$\delta\mathbf{u}^R = \sqrt{\frac{1}{N}\sum_{\mathrm{n}=1}^{N}\left|\mathbf{u}^R(\mathbf{x}_\mathrm{n}) - \tilde{\mathbf{u}}^R(\mathbf{x}_\mathrm{n})\right|^2}.$$

Following the analysis by Kolafa and Perram, we can use Lemma 1 to estimate $\delta\mathbf{u}^R$ as

$$\left(\delta\mathbf{u}^R\right)^2 \approx \frac{1}{L^3}\sum_{\mathrm{n}=1}^{N}\left(\mathbf{f}(\mathbf{x}_\mathrm{n})\right)^2 \cdot \int_{r>r_c}\left(G^R(\mathbf{r})\right)^2 \mathrm{d}\mathbf{r}.$$

Estimates based on this approximation are already available in the literature for the stokeslet [23] and rotlet [1] decompositions used in this paper and are shown in the summary in Table 3. We will here derive a similar estimate for the Hasimoto decomposition of the stresslet, essentially by repeating the derivation of [3] for the Beenakker decomposition.

The real space component of the stresslet has the form

$$T^R_{jlm}(\xi,\mathbf{r}) = A_1(\xi,r)\hat{r}_j\hat{r}_l\hat{r}_m + A_1(\xi,r)\left(\delta_{jl}\hat{r}_m + \delta_{lm}\hat{r}_j + \delta_{mj}\hat{r}_l\right),$$

and the RMS error is approximated as

$$\left(\delta\mathbf{u}^R\right)^2 \approx \frac{1}{L^3}\sum_{\mathrm{n}=1}^{N}\sum_{j=1}^{3} q_l^2(\mathbf{x}_\mathrm{n})q_m^2(\mathbf{x}_\mathrm{n})\int_{r>r_c}\left(T^R_{jlm}(\mathbf{r})\right)^2 \mathrm{d}\mathbf{r}. \tag{39}$$

Arguing that the error should be independent of the coordinate system orientation, we replace $(T^R_{jlm})^2$ by its average value over the tensor components, computed using spherical coordinates

$$\sum_{j=1}^{3}\left(T^R_{jlm}(\mathbf{r})\right)^2 \approx 3\overline{\left(T^R\right)^2} = \frac{3}{27}\sum_{j,l,m=1}^{3}\left(T^R_{jlm}\right)^2 = \frac{1}{9}\left(A_1^2 + 6A_1A_2 + 15A_2^2\right). \tag{40}$$

This quantity has only radial dependence and is integrable,

$$\begin{aligned}\int_{r>r_c} 3\overline{\left(T^R\right)^2}4\pi r^2\,\mathrm{d}r = &-\frac{32}{3}\sqrt{\pi}\xi e^{-\xi^2 r_c^2}\operatorname{erfc}(\xi r_c) + 21\sqrt{2\pi}\xi\operatorname{erfc}\left(\sqrt{2}\xi r_c\right)\\ &+\frac{16\pi\operatorname{erfc}(\xi r_c)^2}{r_c} + \frac{4}{9}\xi^2 r_c e^{-2\xi^2 r_c^2}\left(28\xi^2 r_c^2 - 3\right)\\ \approx &\frac{112}{9}\xi^4 r_c^3 e^{-2\xi^2 r_c^2},\end{aligned} \tag{41}$$

where we have kept only the dominating term (for large ξr_c) in the last step. Combining (39), (40) and (41) gives us the estimate for the stresslet shown in Table 3. In Fig. 5, the estimates of Table 3 are shown togehter with actual measured errors.

Table 3 Real space truncation error estimates for the stokeslet [23], stresslet and rotlet [1]

	Stokeslet, S^R	*Stresslet, T^R*	*Rotlet, Ω^R*
$\delta\mathbf{u}^R$	$\sqrt{\frac{4Qr_c}{L^3}}e^{-\xi^2 r_c^2}$	$\sqrt{\frac{112Q\xi^4 r_c^3}{9L^3}}e^{-\xi^2 r_c^2}$	$\sqrt{\frac{8Q}{3L^3 r_c}}e^{-\xi^2 r_c^2}$

The quantity Q is defined in the same way as for the Fourier component

8 Summary of method

We now summarize the free-space fast Ewald method for Stokes potentials. Our goal is to compute a discrete-sum potential of the type (13),

$$\mathbf{u}(\mathbf{x}) = \sum_{\mathrm{n}=1}^{N} G(\mathbf{x}-\mathbf{x}_\mathrm{n})\cdot\mathbf{f}(\mathbf{x}_\mathrm{n}),$$

for a set of N target points $\mathbf{x}$, with G being the stokeslet (6), stresslet (7) or rotlet (8). We assume that all target and source points are contained in the cubic domain $\mathcal{D} = [0, L]^3$.

Using an Ewald decomposition (Sect. 3.2) and an Ewald parameter $\xi > 0$, we split the potential into a short-range part $\mathbf{u}^R$ acting locally, and a long-range part $\mathbf{u}^F$ computed in Fourier space,

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}^R(\mathbf{x},\xi) + \mathbf{u}^F(\mathbf{x},\xi) + \mathbf{u}^{\mathrm{self}}(\mathbf{x},\xi),$$

where $\mathbf{u}^{\mathrm{self}}$ refers to the self-interaction term (4), which has to be taken into account only for the stokeslet potential.

The real space component is truncated outside an interaction radius r_c,

$$\mathbf{u}^R(\mathbf{x}) \approx \sum_{\substack{\mathrm{n}=1\\ |\mathbf{x}-\mathbf{x}_\mathrm{n}|\le r_c}}^{N} G^R(\mathbf{x}-\mathbf{x}_\mathrm{n})\cdot\mathbf{f}(\mathbf{x}_\mathrm{n}).$$

This is a local operation in the neighborhood of each target point $\mathbf{x}$ and can be efficiently evaluated using, e.g., a cell list (Sect. 6).

The Fourier space component is evaluated through a Fourier integral, truncated at a maximum wave number k_∞,

$$\mathbf{u}^F(\mathbf{x},\xi) \approx \frac{1}{(2\pi)^3}\int_{|\mathbf{k}|\le k_\infty} e^{i\mathbf{k}\cdot\mathbf{x}}\widehat{G}^{F,\mathcal{R}}(\mathbf{k},\xi)\cdot\sum_{\mathrm{n}=1}^{N}\mathbf{f}(\mathbf{x}_\mathrm{n})e^{-i\mathbf{k}\cdot\mathbf{x}_\mathrm{n}}\,\mathrm{d}\mathbf{k}.$$

The superscript $\mathcal{R}$ denotes that we have removed the singularity in the integrand (at $k = 0$) by truncating the original Green's function outside a maximum interaction radius $\mathcal{R}$ [Eqs. (20–22), (30), Sect. 4]. The integral is evaluated using the spectral Ewald method (Sect. 5.2), which uses FFTs on an $\tilde{M}^3$ grid to efficiently compute the long-range interactions. The method requires the domain $\mathcal{D}$ to be extended by a length δ_L (36) for accurate function support and then zero-padded by a factor 2 for the convolution to be aperiodic when using FFTs. In fact, an oversampling factor $s_f \ge 1+\sqrt{3} \approx 2.8$ is required to accurately resolve the truncated Green's function (Sect. 4.2), but the cost of that can be reduced to a precomputation step involving the fundamental solution to the harmonic or biharmonic equation (Sect. 4.3).

8.1 Computational complexity

We wish to express how the computational cost of the method scales with an increased number of sources and targets N, which we assume to be evenly distributed in the domain $\mathcal{D}$. The system can be scaled up in two different ways: by increasing the point density in a fixed domain or by increasing the domain size L with a fixed point density. Either way, the scaling arguments have as their starting point that the real space sum be $\mathcal{O}(N)$. This is achieved by keeping a constant number of near neighbors (within r_c) for each target under scaling. Additionally, we want the level of the truncation errors to be constant, which is achieved by keeping ξr_c and $\tilde{M}\xi^{-1}L^{-1}$ constant.

If N increases with $\mathcal{D}$ fixed, then $r_c \propto N^{-1/3}$ is required for an $\mathcal{O}(N)$ real space sum. If the accuracy is to remain constant, then $\xi \propto r_c^{-1} \propto N^{1/3}$ and the grid size is scaled as $\tilde{M} \propto \xi \propto N^{1/3}$. This puts the Fourier space cost at $\mathcal{O}(\tilde{M}^3 \log \tilde{M}) \propto \mathcal{O}(N \log N)$.

If the domain size L increases with a fixed point density, then $N \propto L^3$ and the real space sum is $\mathcal{O}(N)$ if we keep r_c and ξ constant. Then $\tilde{M} \propto L \propto N^{1/3}$, such that the Fourier space cost is $\mathcal{O}(\tilde{M}^3 \log \tilde{M}) \propto \mathcal{O}(N \log N)$.

8.2 Parameter selection

For a given system (N charges in a domain of size L), the required parameters for our free-space Ewald method are the Ewald parameter ξ, the real space truncation radius r_c, the number of grid points M covering the original domain, and the Gaussian support width P. Based on these parameters, one can then set δ_L using (36), which then gives $\tilde{L} = L + \delta_L$. This in turn gives $\tilde{M}$, by satisfying $h = L/M = \tilde{L}/\tilde{M}$. We will here draft a strategy for optimizing ξ, r_c, M and P in a large-scale numerical computation.

For a given value of ξ and absolute error tolerance ϵ, close-to optimal values for M and r_c can be computed using the estimates in Tables 2 and 3. The support width P affects the error in the Fourier space component, and we have in practice observed that for the relative error, $P = 16$ gives at least 8 digits of accuracy, while $P = 24$ gives at least 12 digits (see Fig. 3). Our experience is also that $P = 32$ is enough to guarantee that the approximation errors are at roundoff. A look at Fig. 4, however, suggests that full machine precision cannot be achieved even with $P = 32$ and high Fourier space resolution, at least not for the stokeslet and the stresslet. In fact, it turns out that between one and two digits of accuracy are lost for kernels whose Ewald split is based on B (it happens also for the

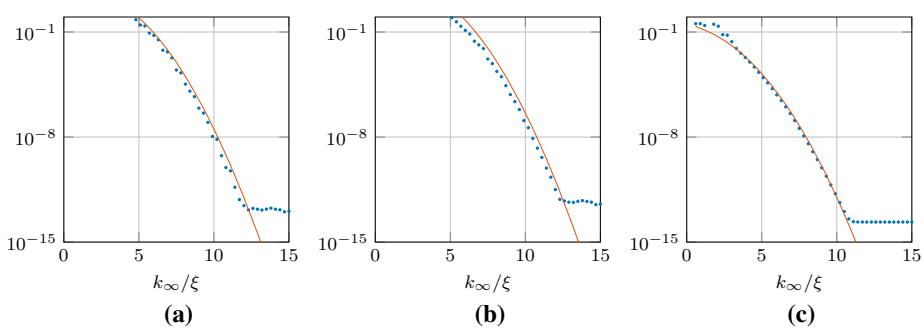

Fig. 4 RMS of relative Fourier space truncation errors for the stokeslet, stresslet and rotlet. Dots are measured value, and solid lines are computed using the estimates of Table 2. The system is $N = 10^4$ randomly distributed point sources in a cube with sides $L = 3$, with $k_\infty = \pi\tilde{M}/\tilde{L}$, $\xi = 3.49$, $M = 1, \ldots, 50$, and $P = 32$. **a** Stokeslet, **b** stresslet, **c** rotlet

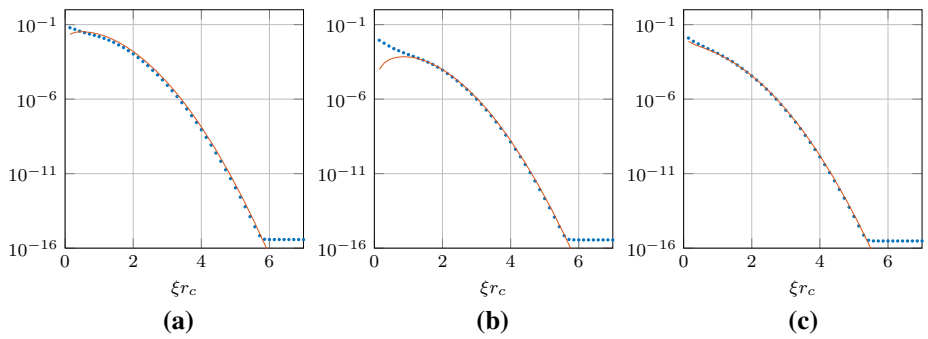

Fig. 5 RMS of relative real space truncation errors for the stokeslet, stresslet and rotlet. *Dots* are measured value, and *solid lines* are computed using the estimates of Table 3. The system is $N = 2000$ randomly distributed point sources in a cube with sides $L = 3$, with $\xi = 4.67$, and $r_c \in [0, L/2]$. **a** Stokeslet, **b** stresslet, **c** rotlet

rotlet if it is based on B rather than H), and we believe it to be due to cancelation errors in the evaluation of $B^{\mathcal{R}}$.

Which value of ξ to choose is highly implementation dependent, as the variable is used to shift the workload between the real and Fourier space components. A straightforward strategy for finding an optimal value is to start with a small but representative subset of the original system and compute a reference solution for that subset. Picking a starting value for ξ, one then sets $P = 32$, and adjusts M and r_c until the error tolerance is strictly met. Then P can be decreased in steps of two[1] until the tolerance is reached again. Using this starting point for (ξ, r_c, M, P), one then does a parameter sweep in ξ for finding the configuration with the smallest runtime, while keeping ξr_c and M/ξ constant during the sweep. Once an optimal setup is found, the original (large) system can be computed using the same set of parameters, except M which is scaled such that L/M remains constant for both systems.

9 Results

We consider systems of N random point sources drawn from a uniform distribution in a box of size L^3. We evaluate the sum (13) with *stokeslets* (6), *stresslets* (7) and *rotlets* (8) using our free space Spectral Ewald (FSE) method, at the same N target locations. All components of the force/source strengths are random numbers from a uniform distribution on $[-1, 1]$. All computationally intensive routines are written in C and are called from Matlab using MEX interfaces. The results are obtained on a desktop workstation with an Intel Core i7-3770 Processor (3.40 GHz) and 8 GB of memory, running all four cores unless stated so. To measure the actual errors, we compare to the result from evaluating the sum by direct summation.

9.1 Computational cost

First, we measure the computational cost of our implementation of the method. In the left plot of Fig. 6, the computing time for evaluation of the sums is plotted versus N, for all three kernels and for both the Spectral Ewald (FSE) method and direct summation. The parameters in the Spectral Ewald method have been set to keep the relative RMS error

[1] P a multiple of two is favorable for code optimization using vector instructions.

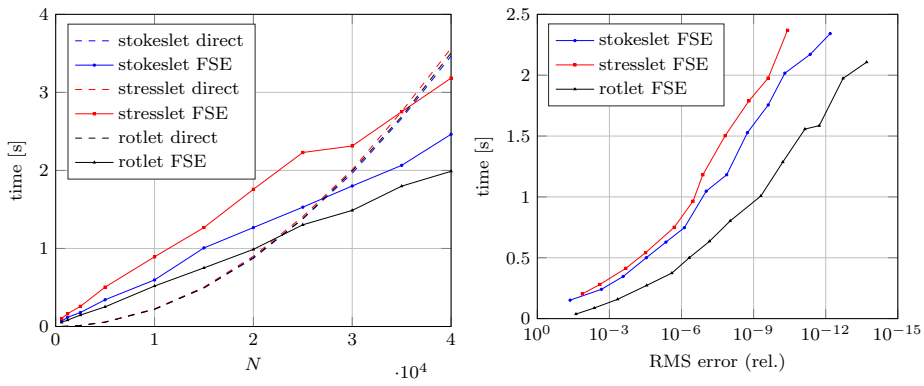

Fig. 6 *Left* Comparison of direct and fast evaluation of the sum in (13) for the stokeslet, stresslet and rotlet including the precomputation step. The system grows at constant density (N/L^3 is constant), with $\xi = 7$ and r_c constant for all values of N. The relative RMS error is less than 0.5×10^{-8}. *Right* Runtime of computing (13) using FSE for all kernels as a function of the relative RMS error. $N = 20{,}000$, $L = 2$ and $\xi = 7$

below 0.5×10^{-8}. The optimal value of ξ cannot be determined theoretically, since it is implementation and hardware dependent. When we vary N in Fig. 6, we change the size of the box, to keep a constant number density $N/L^3 = 2500$. If an optimal value of ξ is determined for one system (see discussion in Sect. 8.2), the same value can be kept as the system is scaled up or down in this manner. The parameters r_c, P and the grid resolution L/M are kept constant as N and hence L is increased, yielding an increase in the grid size. We have used $\xi = 7$ for all three kernels, $r_c = 0.63$, 0.63 and 0.58 for the stokeslet, stresslet and rotlet, respectively, and $P = 16$ for all kernels. For $L = 2$, M is set to 48, 50 and 38 for the three kernels and is then scaled with L.

The precomputation step does not depend on the location of the sources and can be performed once the size of the domain is set. The precomputation cost can therefore usually be amortized over many calls to the method, as a simulation code is run for many time steps and possibly iterations within time steps. Despite this, we have chosen to plot the runtimes including the precomputation cost, and later discuss it in more detail.

From these data (Fig. 6 left), we can find the approximate breakeven points, i.e., the values of N for which any larger system will benefit from using the fast method. We find it to be approximately $N = 27{,}000$ for the stokeslet, 35,000 for the stresslet and 23,000 for the rotlet with precomputation, which is reduced to 22,000, 29,000 and 18,000 without the precomputation step (not shown). If the precomputation step is to be done only once, the decomposition parameter ξ should, however, be chosen differently for optimal performance, which would bring down the break even point further. Note that this is a strict error tolerance. For lower accuracy requirements, the crossover occurs at lower values of N. These are higher values than have previously been reported in the literature, e.g., in [27], where $N = 5000$ was reported as the breakeven for the stokeslet. There are two factors affecting these numbers, one is that these results are run on multiple cores for which the direct sum parallelizes better than the FFTs involved in the fast method. The other factor is that the direct sums relatively speaking have become faster to evaluate also on a single core, where compilers can speed up the code significantly using vector instructions, while the more complicated algorithms cannot benefit from this as extensively.

9.2 Comparison with the FMM

To make sure that our method is competitive, we have compared to a fast multipole implementation available as free software [13], running both codes on a single core and comparing timings for the stokeslet. Note that these timings differ from those in Fig. 6, which are computed using multiple cores. We set the accuracy level to six digits in the FMM. For $N = 20{,}000$, this yields a relative RMS error of about 5.6×10^{-9}, and we set the parameters for the FSE method to obtain a similar error level (for this case we get 4.3×10^{-9}). For $N = 20{,}000$, the FSE code (including precomputation) and the FMM code both use about 3 s. The direct evaluation takes 3.6 s with our code and 6.7 s with the code provided with the FMM package. It should, however, be noted that the FMM as well as the direct code from that package returns not only the three vector components produced by the stokeslet, but also the associated (scalar) pressure, which increases the cost somewhat. The breakeven point for both FSE and FMM is about $N = 17{,}000$ when comparing to our direct code. If we instead compare to the direct code in the FMM package, the break even point for the FMM decreases to $N = 10{,}000$. Most fair would be to compare the FMM to a direct sum written as the faster one, but including also the pressure component, which should place the break even point between the two numbers above. For the FSE code, assuming that the precomputation will be done only once, and choosing ξ instead to optimize the runtime without precomputation, the break even point drops from $N = 17{,}000$ to 11,000.

Let us consider also a larger system with $N = 400{,}000$, with $N/L^3 = 2500$ (i.e., $L \approx 5.43$). For the stokeslet summation by FSE, we pick the parameters $\xi = 8$, $r_c = 0.5651$, $M = 144$ and $P = 16$ to obtain a relative RMS error of 5×10^{-8}. This means that the FFTs are computed for grids with $\tilde{M} = 2(M + P)$. The time for evaluation is about 64 s (including the precomputation), and the speed-up compared to our direct evaluation of the sum is a factor of about 23. Excluding the precomputation cost, the computing time is reduced by 15 s, and this factor increases to 29. For the FMM, the evaluation time is about 180 s, yielding a speed-up of a factor of about 8 compared to our direct sum or a factor of 15 as compared to the one provided with the FMM code [13]. Checking the relative RMS error from both the FSE and FMM computations, they are similar, around 0.5×10^{-8} for FSE and 10^{-8} for the FMM. Hence, for this example on a single core, the FSE method including the precomputation is almost three times as fast as the FMM method, but the difference would be reduced somewhat if the time for computing the extra pressure component was excluded.

In the adaptive FMM code, a box is split into 8 children boxes if the number of sources is larger than a set value. If any of the children boxes still have too many source points, it is split again. With a uniform distribution of points, most leaf boxes are on the same level of refinement, which in this case will be four divisions. The curve for computational cost versus N will not be smooth, since this is a discrete process (either you keep the box as one or you split into eight), which changes the cost balance between different parts of the algorithm. This is why the larger computational cost for the FMM method in this case could not be predicted considering the timing for $N = 20{,}000$, where the timing of the FMM and FSE methods were similar.

We did not set out to make a thorough comparison of the two methods. All results are for uniform distributions of source points. Typically, the FSE performs better compared

to the FMM for higher accuracies. Moving toward an increasingly non-uniform point distribution, the adaptivity of the FMM will at some point pay off. With this, we have, however, showed that the FSE method is competitive with the FMM.

9.3 Cost versus accuracy

To show how the computational cost depends on the accuracy requirements, we now consider a fixed system with $N = 20{,}000$ sources in a box with $L = 2$ and vary the error tolerance. In the right plot of Fig. 6, we plot the runtime for summing the stokeslet, stresslet and rotlet kernels as a function of the relative RMS error in the result. Computing the k-space contribution for the stokeslet and rotlet involves gridding of three vector components, three FFTs, a scaling in Fourier space, three inverse FFTs and the quadrature step for the three components of the solution, see the algorithm in Sect. 5.2. The stresslet instead requires the gridding of 9 components and hence 9 FFTs. After the scaling step, there are three resulting vector components, as for the other kernels. All three kernels require the same amount of precomputing. Hence, it is not surprising to see that the stresslet is the most expensive kernel to compute. We expect a higher cost of the stokeslet as compared to the rotlet due to the slower decay of the Fourier space part, as given in Table 2. This means that larger FFT grids are needed to obtain the same accuracy. See, e.g., the discussion in connection to the left plot in Fig. 6 where the choice of M for the box $L = 2$ is 48 for the stokeslet and 38 for the rotlet.

9.4 Cost breakdown

For the same system as in Fig. 6 (left), we now study the computational cost for the different parts of the calculations for the stokeslet. In the left plot of Fig. 7, we show the total evaluation runtime for the stokeslet sum together with the three parts that makes up this total cost: the real space and Fourier space evaluations plus the precomputation in Fourier space. We use the choice of $\delta_L = d$ in (36), such that $\tilde{L} = L + d$. With this, $\tilde{M} = M + P$, and the FFTs in the Fourier space evaluations will be of size $(2\tilde{M})^3$. For the precomputation, the size of the FFT grids in each dimension will be taken as the smallest even number that is greater than $(1+\sqrt{3})\tilde{M}$. The plot shows that the computational cost is very similar for the real space evaluation and the total Fourier space work (precomputation plus evaluation).

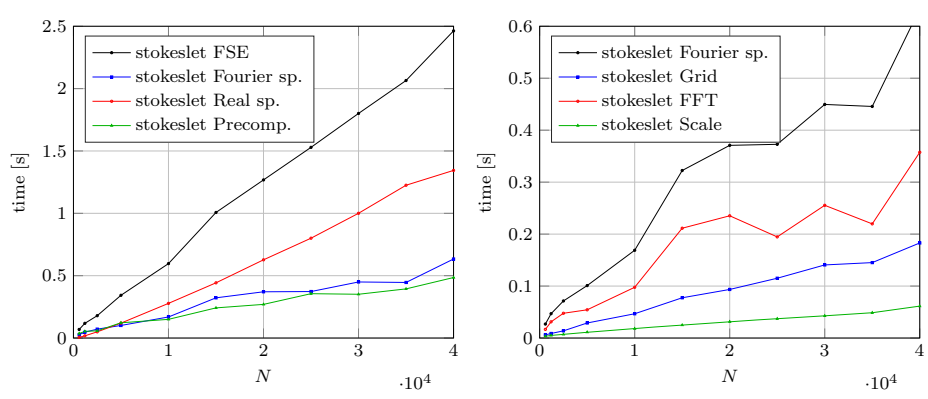

Fig. 7 Breakdown of runtimes (*left*) and Fourier space runtime (*right*) for evaluating the stokeslet as a function of number of particles. The full runtime was also shown in the left plot of Fig. 6

While implementation dependent, we often see that optimizing ξ for performance puts the costs at comparable magnitudes. As discussed above, the precomputation does not depend on the sources and can be done only once as long as the domain size does not change. Excluding the precomputation cost from the timing of the stokeslet, the runtime is reduced somewhere between a quarter and one third. Readjustment of ξ to instead balance computational costs excluding the precomputation would yield a further reduced runtime.

In the right plot of Fig. 7, we further break down the cost of evaluating the Fourier space sum into three parts: *Grid* (the to and from grid operations with Gaussians), *FFT* (the total of 6 FFTs) and *Scale*, the multiplication in step 3 of the algorithm in Sect. 5.2. Note here that the oscillations in the FFT curve are due to the fact that the FFT is more efficient for some grid sizes. The scaling step is clearly the cheapest of the three parts. The cost of the gridding step is $O(P^3N)$, where P^3 are the number of grid points in the support of a Gaussian, and the cost of each FFT of size $(2\tilde{M})^3$ is $O(\tilde{M}^3 \log \tilde{M})$. Due to the connection to the real space sum, the choice of $\tilde{M}$ will be such that this cost scales as $O(N \log N)$, as discussed in Sect. 8.1.

10 Conclusions

We have presented a new fast summation method for free space Green's functions of Stokes flow. The method is based on an Ewald decomposition to split the sum in two parts, one in real space and one in Fourier space. The real space sum can simply be truncated outside of some radius of interaction that depends on the choice of decomposition parameter and the required accuracy. The focus of this paper is on the Fourier space sum, the treatment of which is set in the framework of the Spectral Ewald method, previously developed for periodic problems [3,21]. The adaptation to the free space problem involves a very recent approach to solving the free-space harmonic and biharmonic equations using FFTs on a uniform grid [28]. The Ewald Fourier space kernels for the stokeslet, stresslet and rotlet are defined from the precomputed Fourier representation of mollified harmonic (rotlet) and biharmonic (stokeslet and stresslet) kernels, and the method can easily be extended to any kernel that can be expressed as a differentiation of the harmonic and/or biharmonic kernel. New truncation error estimates have been derived for the free space kernels.

The extension of the FFT-based Spectral Ewald method to the free space problem incurs an additional computational cost compared to the periodic problem. This is essentially due to the computation of larger FFTs, as computational grids are zero-padded to the double size before the FFTs are computed. There is also an additional cost of two oversampled FFTs for precomputing the Fourier representation of the mollified harmonic or biharmonic kernel. This precomputation does not depend on the sources, and the cost can often be amortized over many sum evaluations.

Truncation error estimates have been derived for the kernels for which they did not already exist, such that precise estimates of the errors introduced by truncating the real and Fourier space sums are available for all three kernels, the stokeslet, stresslet and rotlet. Errors decay exponentially in the physical distance and wave mode number used for cutoff. Approximation errors in the evaluation of the Fourier sum decays exponentially with the support of the Gaussians. An intricate detail needed to preserve the decoupling between

truncation and approximation errors that is not relevant for the periodic Spectral Ewald method is discussed in Sect. 5.3.

Numerical results are presented for the evaluation of the stokeslet, stresslet and rotlet sums. They show the expected $O(N \log N)$ computational cost of the method. We have compared to an open source implementation of the FMM method [13] and have shown that our method is competitive, as it performs better for the uniform source distributions and high accuracies considered here.

With this, we have developed a new FFT-based method for the fast evaluation of free space Green's functions for Stokes flow (stokeslets, stresslets and rotlets) in a free space setting. This free space Spectral Ewald method allows the use of the same framework as the periodic one, which makes it easy to swap methods depending on the problem under consideration. The source code for the triply periodic SE method is available online [24], and we plan to shortly release also the code for this free space implementation.

Acknowledgements
This work has been supported by the Göran Gustafsson Foundation for Research in Natural Sciences and Medicine, by the Swedish Research Council under Grant No. 2011-3178, and by the Swedish e-Science Research Centre (SeRC). The authors gratefully acknowledge this support.

Received: 15 July 2016 Accepted: 10 December 2016
Published online: 01 February 2017

References

1. af Klinteberg, L.: Ewald summation for the rotlet singularity of Stokes flow (2016a). arXiv:1603.07467 [physics.flu-dyn]
2. af Klinteberg, L.: Fast and accurate integral equation methods with applications in microfluidics. PhD thesis, KTH Royal Institute of Technology (2016b). http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-185758
3. af Klinteberg, L., Tornberg, A.-K.: Fast Ewald summation for Stokesian particle suspensions. Int. J. Numer. Methods Fluids **76**(10), 669–698 (2014). doi:10.1002/fld.3953
4. Beenakker, C.W.J.: Ewald sum of the Rotne–Prager tensor. J. Chem. Phys. **85**(3), 1581 (1986). doi:10.1063/1.451199
5. Cheng, H., Greengard, L., Rokhlin, V.: A fast adaptive multipole algorithm in three dimensions. J. Comput. Phys. **155**(2), 468–498 (1999). doi:10.1006/jcph.1999.6355
6. Darden, T., York, D., Pedersen, L.: Particle mesh Ewald—an N·log(N) method for Ewald sums in large systems. J. Chem. Phys. **98**(12), 10089–10092 (1993)
7. Deserno, M., Holm, C.: How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines. J. Chem. Phys. **109**(18), 7678 (1998). doi:10.1063/1.477414
8. Essmann, U., Perera, L., Berkowitz, M.L., Darden, T., Lee, H., Pedersen, L.G.: A smooth particle mesh Ewald method. J. Chem. Phys. **103**(19), 8577–8593 (1995)
9. Ewald, P.P.: Die Berechnung optischer und elektrostatischer Gitterpotentiale. Ann. Phys. **369**(3), 253–287 (1921). doi:10.1002/andp.19213690304
10. Fan, X., Phan-Thien, N., Zheng, R.: Completed double layer boundary element method for periodic suspensions. Z. Angew. Math. Phys. **49**(2), 167–193 (1998). doi:10.1007/s000330050214
11. Frenkel, D., Smit, B.: Understanding Molecular Simulation, 2nd edn. Academic Press, London (2001)
12. Fu, Y., Rodin, G.J.: Fast solution methods for three-dimensional Stokesian many-particle problems. Commun. Numer. Methods. Eng. **16**, 145–149 (2000)
13. Greengard, L.: Fast multipole methods for the Laplace, Helmholtz and Stokes equations in three dimensions (2012). http://www.cims.nyu.edu/cmcl/fmm3dlib/fmm3dlib.html
14. Greengard, L., Lee, J.-Y.: Accelerating the nonuniform fast Fourier transform. SIAM Rev. **46**(3), 443–454 (2004). doi:10.1137/S003614450343200X
15. Greengard, L., Rokhlin, V.: A fast algorithm for particle simulations. J. Comput. Phys. **73**, 325–348 (1987)
16. Gumerov, N.A., Duraiswami, R.: Fast multipole method for the biharmonic equation in three dimensions. J. Comput. Phys. **215**, 363–383 (2006)
17. Hasimoto, H.: 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 (1959). doi:10.1017/S0022112059000222
18. Hernández-Ortiz, J., de Pablo, J., Graham, M.: Fast computation of many-particle hydrodynamic and electrostatic interactions in a confined geometry. Phys. Rev. Lett. **98**(14), 140602 (2007). doi:10.1103/PhysRevLett.98.140602
19. Kolafa, J., Perram, J.W.: Cutoff errors in the Ewald summation formulae for point charge systems. Mol. Simul. **9**(5), 351–368 (1992). doi:10.1080/08927029208049126
20. Lee, J.-Y., Greengard, L.: The type 3 nonuniform FFT and its applications. J. Comput. Phys. **206**(1), 1–5 (2005). doi:10.1016/j.jcp.2004.12.004
21. Lindbo, D., Tornberg, A.-K.: Spectrally accurate fast summation for periodic Stokes potentials. J. Comput. Phys. **229**(23), 8994–9010 (2010). doi:10.1016/j.jcp.2010.08.026
22. Lindbo, D., Tornberg, A.-K.: 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

23. Lindbo, D., Tornberg, A.-K.: Fast and spectrally accurate summation of 2-periodic Stokes potentials (2011b). arXiv:1111.1815v1 [physics.flu-dyn]
24. Lindbo, D., af Klinteberg, L., Shamshirgar, D. Saffar.: The spectral Ewald unified package (2016). http://github.com/ludvigak/SE_unified
25. Pozrikidis, C.: Computation of periodic Green's functions of Stokes flow. J. Eng. Math. **30**(1–2), 79–96 (1996). doi:10.1007/BF00118824
26. Saintillan, D., Darve, E., Shaqfeh, E.: A smooth particle-mesh Ewald algorithm for Stokes suspension simulations: the sedimentation of fibers. Phys. Fluids **17**, 03301 (2005)
27. Tornberg, A.-K., Greengard, L.: 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
28. Vico, F., Greengard, L., Ferrando, M.: Fast convolution with free-space Green's functions. J. Comput. Phys. **323**, 191–203 (2016). doi:10.1016/j.jcp.2016.07.028
29. Wang, H., Lei, T., Li, J., Huang, J., Yao, Z.: A parallel fast multipole accelerated integral equation scheme for 3D Stokes equations. Int. J. Numer. Methods Eng. **70**(7), 812–839 (2007)
30. Wang, M., Brady, J.F.: Spectral Ewald acceleration of Stokesian dynamics for polydisperse suspensions. J. Comput. Phys. **306**, 443–477 (2016)
31. Wang, X., Kanpka, J., Ye, W., Aluru, N.R., White, J.: Algorithms in FastStokes and its application to micromachined device simulation. IEEE Trans. Comput. Des. Integr. Circuits Syst. **25**, 248–257 (2006)
32. Ying, L., Biros, G., Zorin, D.: A kernel-independent adaptive fast multipole algorithm in two and three dimensions. J. Comput. Phys. **196**(2), 591–626 (2004). doi:10.1016/j.jcp.2003.11.021