Received: 9 June 2020 | Revised: 16 January 2021 | Accepted: 17 February 2021

DOI: 10.1002/fld.4970

RESEARCH ARTICLE

WILEY

Highly accurate special quadrature methods for Stokesian particle suspensions in confined geometries

Joar Bagge | **Anna-Karin Tornberg**

Department of Mathematics/Linné Flow Centre/Swedish e-Science Research Centre, KTH Royal Institute of Technology, Stockholm, Sweden

Correspondence
Joar Bagge, Department of Mathematics, KTH Royal Institute of Technology, Stockholm 100 44, Sweden.
Email: joarb@kth.se

Funding information
Göran Gustafssons Stiftelse för Naturvetenskaplig och Medicinsk Forskning; Vetenskapsrådet, Grant/Award Number: 2015-04998

Abstract

Boundary integral methods are highly suited for problems with complicated geometries, but require special quadrature methods to accurately compute the singular and nearly singular layer potentials that appear in them. This article presents a boundary integral method that can be used to study the motion of rigid particles in three-dimensional periodic Stokes flow with confining walls. A centerpiece of our method is the highly accurate special quadrature method, which is based on a combination of upsampled quadrature and quadrature by expansion, accelerated using a precomputation scheme. The method is demonstrated for rodlike and spheroidal particles, with the confining geometry given by a pipe or a pair of flat walls. A parameter selection strategy for the special quadrature method is presented and tested. Periodic interactions are computed using the spectral Ewald fast summation method, which allows our method to run in $O(n \log n)$ time for n grid points in the primary cell, assuming the number of geometrical objects grows while the grid point concentration is kept fixed.

KEYWORDS

boundary integral equations, fast Ewald summation, quadrature by expansion, rigid particle suspensions, Stokes flow, streamline computation

1 | INTRODUCTION

Microhydrodynamics is the study of fluid flow at low Reynolds numbers, also known as Stokes flow or creeping flow. Applications are found in biology, for example in the swimming of microorganisms[1] and in blood flow,[2] as well as in the field of microfluidics, which concerns the design and construction of miniaturized fluid devices.[3] Suspensions of rigid particles in Stokes flow are important both in various applications and in fundamental fluid mechanics.[4-7] In this article, we describe a boundary integral method that can be used to study the motion of rigid particles of different shapes in Stokes flow. The particle suspension may also be confined in a container geometry, such as a pipe or a pair of flat walls. The flow in the fluid domain (i.e., within the container but outside the particles) is governed by the *Stokes equations*, which for an incompressible Newtonian fluid take the form

$$\nabla p - \mu \nabla^2 \boldsymbol{u} = \boldsymbol{f}, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{2}$$

Here, p is the pressure, $\boldsymbol{u}$ is the flow velocity, $\boldsymbol{f}$ is the body force per unit volume, and μ is the viscosity of the fluid. The Stokes equations arise as a linearization of the Navier–Stokes equations in the case where fluid inertia can be neglected, that is, when the Reynolds number is much less than 1.

On the surfaces of the particles and walls, no-slip boundary conditions are prescribed. A problem of physical interest is the *resistance problem*: given the velocities of all particles, compute the forces and torques (caused by viscous resistance) acting on them by the fluid. The inverse problem is called the *mobility problem*: given the forces and torques acting on all particles by the fluid, compute the particle velocities. The mobility problem is useful in the case of noninertial particles, since then the net force on each particle must be zero, so any external forces (such as gravity) must be balanced by viscous forces from the fluid; given external forces and torques, one can then compute the motion of the particles.

Since the governing equations (1)–(2) are linear, boundary integral methods can be used to solve them. In these methods, the flow is expressed in terms of layer potentials, which are integrals over the boundary of the fluid domain (i.e., over the container walls and particle surfaces). This reduces the dimensionality of the problem from three to two, and leads to a smaller linear system compared with methods that must discretize the whole volume (such as the finite difference or finite element methods). If the meshed surfaces are rigid, as is the case here, it is also easy to move the particles without any need for remeshing. For a detailed discussion on the properties of boundary integral methods, we refer to the books by Pozrikidis,[8] Atkinson,[9] and Kress.[10] Of special importance are Fredholm integral equations of the second kind, which when discretized, for example using the Nyström method [9, ch. 4, 10, sec. 12.2], are known to remain well-conditioned as the system size increases [9, p. 113, 10, p. 282].

The linear system resulting from the discretization of a boundary integral equation is dense, and thus naive Gaussian elimination would require $O(N^3)$ operations to solve a system of N unknowns. Using an iterative solution method such as the generalized minimal residual method (GMRES),[11] the complexity is reduced to $O(N^2)$ since the condition number and thus the number of iterations are independent of the system size (but may depend on the geometry). For a large system, this complexity is still prohibitive. This can be overcome by using a fast summation method such as the fast multipole method (FMM)[12,13] or a fast Ewald summation method,[14-16] which reduce the complexity further to $O(N)$ or $O(N \log N)$, respectively.

One of the challenges of boundary integral methods is the need for accurate special quadrature methods for singular and nearly singular integrands. These are necessary when evaluating the layer potentials at a point on the boundary (where the integral kernel is singular) or close to the boundary (where the kernel is nearly singular, that is, hard to resolve using a quadrature rule designed for smooth integrands). Such special quadrature methods are the main focus of this article.

1.1 | Overview of related work

Volume discretization methods for particulate Stokes flow, based for example on finite elements or finite differences, can be divided into conforming and nonconforming methods. In conforming methods (e.g., the arbitrary Lagrangian–Eulerian method[17]), the mesh conforms to the particle geometry and moves with the particles. In nonconforming methods (e.g., the immersed boundary method[18] and the distributed Lagrange multiplier/fictitious domain method[19]), the mesh is fixed and cuts through the particles. The lattice-Boltzmann method[20,21] also has a fixed computational grid, and can be combined with the immersed boundary method.[22] For specific particle shapes, such as spherical or spheroidal particles, methods based on approximations such as Stokesian dynamics[23,24] or force-coupling[25] are available. For a more detailed overview, we refer to Liu et al.[25] and Wu and Shu,[22] and references therein. Many of these methods are limited by difficulties to enforce boundary conditions to high accuracy at the particle surface, especially for moving particles, which reduces the accuracy of the method as a whole. Boundary integral methods,[26,27] by contrast, have no volume grid but only a fixed grid on the surface of each particle, and allow boundary conditions to be set so that high accuracy can be achieved. Furthermore, boundary integral methods also allow unbounded domains to be treated naturally without truncation, for example when studying particle dynamics without confinement. As mentioned above, the accuracy of boundary integral methods hinges on the availability of an accurate special quadrature method.

In two dimensions, there are excellent special quadrature methods available, such as the one introduced by Helsing and Ojala,[28] which has been adapted to simulations of clean[29] and surfactant-covered[30] drops in Stokes flow, as well as vesicles.[31] However, this method is based on a complex variable formulation and not easy to generalize to three dimensions.

In three dimensions, the development of an accurate and efficient special quadrature method is still an active research problem, especially in the nearly singular case. For an overview of methods that have been used, we refer to Klöckner et al. [32, sec. 1] and Rahimian et al. [33, sec. 1]. One of the most promising methods which is still under development is *quadrature by expansion* (QBX), first introduced by Klöckner et al.[32] and Barnett[34] and applied to the Helmholtz equation in two dimensions. This method is based on the observation that the layer potentials are smooth all the way up to the boundary, and can therefore be locally expanded around a point away from the boundary. The expansions can be evaluated at a point closer to the boundary, or even on the boundary itself. The convergence theory of QBX was developed by Epstein et al.,[35] while af Klinteberg and Tornberg[36] analyzed the error from the underlying quadrature rule used to compute the expansion coefficients. A strength of QBX is that it separates source points and target points; the source points enter only in the computation of the expansion coefficients, which can then be used to evaluate the layer potential in all target points within a ball of convergence. QBX has been applied to spheroidal particles in three-dimensional Stokes flow by af Klinteberg and Tornberg,[37] using a geometry-specific precomputation scheme to accelerate the computation of the coefficients.

A different approach that has been taken to accelerate QBX is to couple it to a customized FMM, which has been done in two dimensions[38-40] and more recently in three dimensions.[41,42] This coupling is a natural step to take since the FMM uses expansions of the same kind as QBX, but it requires nontrivial modifications to the FMM. The resulting method has complexity $O(N)$ and works for any smooth geometry. The work published so far has been for the Laplace and Helmholtz equations, but it is likely to be extended to more kernels, including the ones needed for Stokes flow.

The QBX-FMM methods above all use *global* QBX, in which *all* source points are included in forming the local expansion. An alternative is *local* QBX, in which only source points that are close to the expansion center are included. Yet another variant is found in Reference 37, where all source points on a single particle is used when forming expansions close to that particle; we call this variant *particle-global*. Local QBX is typically combined with a patch-based discretization of the geometry. While it reduces the cost of the method, it also poses a challenge since the local layer potential from a single patch may not be as smooth as the global layer potential from the whole geometry (or a whole particle). Different versions of local QBX have been described in two dimensions[34,38] and three dimensions.[43] The latter article also uses target-specific expansions, which need only $O(p)$ terms to obtain the same accuracy as a QBX expansion based on spherical harmonics with $O(p^2)$ terms. However, they sacrifice the separation of source and target that is otherwise present in QBX. This separation is necessary in the QBX-FMM methods, although target-specific expansions can also be used in QBX-FMM methods to lower the computational cost.[42]

Some of the recent work have focused on automating the parameter selection based on a given error tolerance, resulting in the *adaptive* QBX method.[44] The results have so far not been generalized to three dimensions. There has also been work on a kernel-independent version of QBX, called *quadrature by kernel-independent expansion*[33] and meant to be combined with the kernel-independent FMM. The published work is in two dimensions, but a generalization to three dimensions is expected to follow.

Other methods, which are not based on QBX, have also been used successfully as special quadrature methods in three dimensions. One example is the "line interpolation method" introduced by Ying et al..[45] In this method, a line is constructed through the target point, which is close to the boundary, and its projection onto the boundary. The layer potential is evaluated at points further away from the boundary along this line, and also at the projection point where the line intersects the boundary if a separate singular integration method is available. The value at the target point is then computed using interpolation along the line (or extrapolation if no singular integration method is available). Like QBX, the success of this method hinges on the fact that the layer potential is smooth in the domain, so that it can be well interpolated (or extrapolated). It has been applied to surfactant-covered drops[46] and vesicles[47] in three-dimensional Stokes flow. The extrapolatory method used by Lu et al.[2] falls into the same category. Other types of methods are based on regularizing the kernel and adding corrections,[48-51] density interpolation techniques,[52] coordinate rotations and a subtraction method,[53] asymptotic approximations,[54] analytical expressions available only for spheres[55] or floating partitions of unity.[56-58] Many of these methods are target-specific, and their cost grows rapidly if there are many nearly singular target points.

1.2 | Scope of this article

In this article, we present a boundary integral method based on the Stokes double layer potential, which can be used to solve the mobility and resistance problems for a system of rigid particles in incompressible three-dimensional Stokes flow, possibly confined within a container geometry. Our formulation leads to a Fredholm integral equation of the second kind.

We use QBX for singular integration, and a combination of QBX and upsampled quadrature for nearly singular integration. Our QBX implementation is based on the work by af Klinteberg and Tornberg,[37] which we have extended to rodlike particles, plane walls and pipes (using particle-global QBX for the particles and local QBX for the two wall geometries). A precomputation scheme is used to greatly accelerate the QBX computations for all geometries. For this precomputation scheme to be feasible, we require that each particle or wall is rigid and has some degree of symmetry, such as axisymmetry or reflective symmetry. Nonetheless, we have chosen this route since the implementation is relatively simple compared with for example, a QBX-FMM method. When container walls are present, we restrict ourselves to periodicity in all three spatial directions and use a fast Ewald summation method called the spectral Ewald (SE) method[58-60] to accelerate computations.[*] In this situation our method scales as $O(N \log N)$ in the number of unknowns N, assuming fixed grid point concentration.

The main focus of this article is on the application of the previously developed QBX and SE methods to rigid particle suspensions with the confining geometry of a periodic straight pipe or a pair of periodic plane walls. Our contributions include:

- The combined special quadrature method based on QBX and upsampling, which we have implemented for spheroidal and rodlike particles, plane walls and pipes. (The QBX implementation for spheroids is reused from Reference 37. Our initial work on QBX for plane walls is published in the conference proceedings.[62])
- A strategy for experimentally selecting the parameters of the special quadrature method to meet a given error tolerance. We also demonstrate that the boundary integral method in full meets the given error tolerance and scales as $O(N \log N)$.
- Construction of fully smooth rodlike particles. We demonstrate the effect of smoothness on the convergence of the local expansions in this particular case.
- Derivation of a stresslet identity for an infinite pipe and a pair of infinite plane walls. This is used as an exact solution to test the special quadrature method.
- An outline of how streamlines can be efficiently computed for periodic problems using the SE method, by reusing data. (This idea was used, but not explicitly described, in Reference 37.)
- The so-called completion sources that appear in our formulation are distributed along the axis of symmetry of rodlike particles, and we have studied how the number of completion sources influences the accuracy.

1.3 | Organization of the article

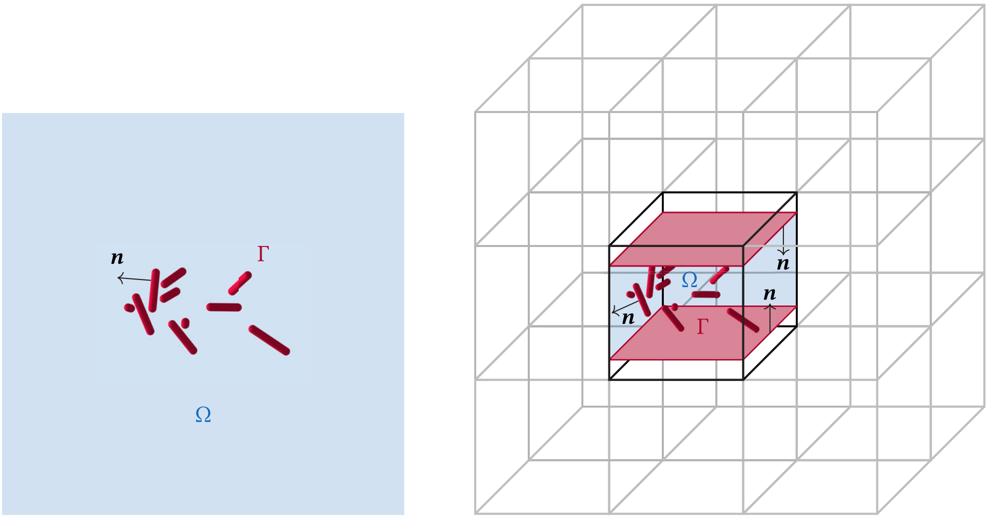
In Section 2, we introduce the mathematical formulation of the problem, including the boundary integral formulation and the boundary integral equations for the resistance and mobility problems. In Section 3, we describe the discretization of the geometry and the quadrature method, including the combined special quadrature. The details on our QBX method are then given in Section 4, including the precomputation scheme. In Section 5, we describe how periodicity is treated and how the special quadrature is combined with the SE method. Then, in Section 6, our parameter selection strategy for the special quadrature is described and demonstrated. Numerical results are given in Section 7, to demonstrate the accuracy and scaling of our method. Finally, in Section 8, we demonstrate the effect of nonsmooth geometries on the convergence. The appendices include a derivation of the stresslet identity for plane walls and pipes, details on the construction of the smooth rodlike particles, and a note on streamline computation.

2 | MATHEMATICAL FORMULATION

We consider two different kinds of problems: free-space problems and fully periodic problems. In a *free-space problem*, M particles (spheroids or rods) are located in a fluid extending to infinity. We denote the fluid domain by Ω and its boundary, that is, the union of all particle surfaces, by Γ. The Stokes equations (1)–(2) with $\boldsymbol{f} = \boldsymbol{0}$ hold in Ω, while no-slip boundary conditions are imposed on Γ. The unit normal vector $\boldsymbol{n}$ of Γ is defined to point into the fluid domain Ω, as shown in Figure 1(A).

A *fully periodic problem*, on the other hand, is periodic in all three spatial directions. The *primary cell* is a box with side lengths $\boldsymbol{B} = (B_1, B_2, B_3)$, which is considered to be replicated periodically in all three spatial directions, as shown in

[*] The implementation of the SE method that we use is publicly available at Reference 61.



(A) Free-space problem

(B) Fully periodic problem

FIGURE 1 The geometry for (A) a free-space problem, and (B) a fully periodic problem. In (B), the primary cell is marked with a darker outline than the other cells. The geometry is shown only in the primary cell. The lattice of periodic cells fills the whole space; only a small part is shown here [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 1(B). Let the number of particles in the primary cell be M. In this case we also allow a container consisting either of a pair of plane walls or a pipe. Only the geometry inside the primary cell is discretized, which means that Γ consists of the union of the M particle surfaces and the parts of the wall surfaces that lie in the primary cell.[†] The fluid domain Ω lies within the container but outside the particles; the flow is thus external to the particles, but internal to the surrounding walls. The unit normal vector $\boldsymbol{n}$ of Γ is always defined to point into Ω.

Below, we introduce our boundary integral formulation in the free-space setting, or for the primary cell without periodicity. Full treatment of the periodic problem is deferred to Section 5.

2.1 | Boundary integral formulation

Any flow field $\boldsymbol{u}$ that satisfies the Stokes equations (1)–(2) with $\boldsymbol{f} = \boldsymbol{0}$ can be expressed in terms of integrals over the boundary of the fluid domain Ω, as described for example by Pozrikidis [8, ch. 4] and Kim and Karrila [66, ch. 14–16]. The boundary integral formulation that we use is based on the *Stokes double layer potential* $\boldsymbol{\mathcal{D}}$, which in free space is given by[‡]

$$\mathcal{D}_i[\Gamma, \boldsymbol{q}](\boldsymbol{x}) = \int_\Gamma T_{ijk}(\boldsymbol{x} - \boldsymbol{y}) q_j(\boldsymbol{y}) n_k(\boldsymbol{y}) \,\mathrm{d}S(\boldsymbol{y}), \qquad T_{ijk}(\boldsymbol{r}) = -6\frac{r_i r_j r_k}{|\boldsymbol{r}|^5}. \tag{3}$$

Here, Γ and $\boldsymbol{n}$ are as in Figure 1, and the *double layer density* $\boldsymbol{q}$ is a continuous vector field defined on Γ. The tensor kernel $\boldsymbol{T}$ in (3) is known as the *stresslet*. The potential $\boldsymbol{\mathcal{D}}$ has a jump discontinuity as $\boldsymbol{x}$ passes over Γ. More specifically, for $\boldsymbol{x} \in \Gamma$ it holds that [8, p. 110]

$$\lim_{\varepsilon \to 0^+} \boldsymbol{\mathcal{D}}[\Gamma, \boldsymbol{q}](\boldsymbol{x} \pm \varepsilon \boldsymbol{n}) = \boldsymbol{\mathcal{D}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}) \mp 4\pi \boldsymbol{q}(\boldsymbol{x}). \tag{4}$$

For any closed Lyapunov surface $\tilde{\Gamma} \subseteq \Gamma$ and any constant vector $\tilde{\boldsymbol{q}}$, the *stresslet identity* [8, p. 28]

$$\boldsymbol{\mathcal{D}}[\tilde{\Gamma}, \tilde{\boldsymbol{q}}](\boldsymbol{x}) = \begin{cases} \boldsymbol{0}, & \text{if } \boldsymbol{x} \text{ is outside the domain enclosed by } \tilde{\Gamma}, \\ 4\pi \tilde{\boldsymbol{q}}, & \text{if } \boldsymbol{x} \in \tilde{\Gamma}, \\ 8\pi \tilde{\boldsymbol{q}}, & \text{if } \boldsymbol{x} \text{ is inside the domain enclosed by } \tilde{\Gamma}, \end{cases} \tag{5}$$

[†] For a single infinitely large plane wall, the method of images can be used,[63-65] which has the advantage that the wall itself does not need to be discretized. However, that method does not work when there are more than one wall, or when the wall is curved, which are the cases we consider here. Therefore, we must discretize the walls.

[‡] The Einstein summation convention is used in this article, meaning that indices appearing twice in the same term are to be summed over the set $\{1, 2, 3\}$. The remaining free indices take values in the same set.

holds. We will use this identity as a test case for the special quadrature method in Sections 6 and 7.1. In Appendix A we show that a variant of (5) holds also for the wall geometries that we consider, despite them not being closed surfaces.

The double layer potential $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{\mathcal{D}}[\Gamma, \boldsymbol{q}](\boldsymbol{x})$ is a solution to (1)–(2) in Ω for any continuous vector field $\boldsymbol{q}$. However, not every solution to (1)–(2) can be represented by a double layer potential alone; for instance, as noted in References [67, 8, p. 119], the force and torque exerted on any particle by the flow from a double layer potential will always be zero, whereas a Stokes flow in general can exert a nonzero force and torque on the particles (which is a central point of the resistance and mobility problems mentioned in Section 1). This is related to the presence of a nontrivial nullspace of the operator $\boldsymbol{q} \mapsto \boldsymbol{\mathcal{D}}[\Gamma, \boldsymbol{q}]$ for external flows, which can be immediately seen from the stresslet identity (5).

To remove the nontrivial nullspace and allow for nonzero forces and torques on the particles, we add a *completion flow* $\boldsymbol{\mathcal{V}}$, first introduced by Power and Miranda.[67] The completion flow is also a solution to (1)–(2) and can be identified as the flow from a point force $\boldsymbol{F}$ and a point torque $\boldsymbol{\tau}$ located at $\boldsymbol{y}$. It is given by

$$\mathcal{V}_i[\boldsymbol{F}, \boldsymbol{\tau}, \boldsymbol{y}](\boldsymbol{x}) = \frac{1}{8\pi\mu}\left(S_{ij}(\boldsymbol{x}-\boldsymbol{y})F_j + R_{ij}(\boldsymbol{x}-\boldsymbol{y})\tau_j\right), \qquad \boldsymbol{x} \in \Omega, \tag{6}$$

where the *stokeslet* $\boldsymbol{S}$ and the *rotlet* $\boldsymbol{R}$ are given by [§]

$$S_{ij}(\boldsymbol{r}) = \frac{\delta_{ij}}{|\boldsymbol{r}|} + \frac{r_i r_j}{|\boldsymbol{r}|^3} \quad \text{and} \quad R_{ij}(\boldsymbol{r}) = \epsilon_{ijk}\frac{r_k}{|\boldsymbol{r}|^3}, \tag{7}$$

respectively. We call a pair $(\boldsymbol{F}, \boldsymbol{\tau})$ a *completion source*. Such completion sources are placed in the interior of every particle. Mathematically, one completion source per particle is sufficient, but this may lead to numerical problems in some cases. In this article, we allow for multiple completion sources to be distributed along a line segment within the particle; as we show in Section 7.2.1, this is important for elongated particles. For the particle with index α, let $\boldsymbol{F}^{(\alpha)}$ and $\boldsymbol{\tau}^{(\alpha)}$ be the net force and torque, respectively, exerted on the fluid by the particle, and let $\boldsymbol{y}_{\mathrm{c}}^{(\alpha)}$ be the center of mass of the particle. (For a noninertial particle, $\boldsymbol{F}^{(\alpha)}$ and $\boldsymbol{\tau}^{(\alpha)}$ would be equal to the net external force and torque, respectively, acting on the particle.) The completion flow associated with particle α is then given by

$$\boldsymbol{\mathcal{V}}^{(\alpha)}[\boldsymbol{F}^{(\alpha)}, \boldsymbol{\tau}^{(\alpha)}](\boldsymbol{x}) = \frac{1}{N_{\mathrm{src}}}\sum_{s=1}^{N_{\mathrm{src}}} \boldsymbol{\mathcal{V}}[\boldsymbol{F}^{(\alpha)}, \boldsymbol{\tau}^{(\alpha)}, \boldsymbol{y}_{\mathrm{c}}^{(\alpha)} + C(s, N_{\mathrm{src}})\boldsymbol{a}^{(\alpha)}](\boldsymbol{x}), \tag{8}$$

where N_{src} is the number of completion sources per particle, $\boldsymbol{\mathcal{V}}$ is given by (6), and $\boldsymbol{a}^{(\alpha)}$ is a vector specifying the line segment along which completion sources are placed. The function C is given by

$$C(s, N_{\mathrm{src}}) = \begin{cases} 0, & \text{if } N_{\mathrm{src}} = 1, \\ -1 + 2\frac{s-1}{N_{\mathrm{src}}-1}, & \text{if } N_{\mathrm{src}} > 1. \end{cases} \tag{9}$$

Both the double layer potential $\boldsymbol{\mathcal{D}}$ and the completion flow $\boldsymbol{\mathcal{V}}$ have the property that they decay to zero as $\boldsymbol{x} \to \infty$. To be able to represent flows which do not decay, we add a *background flow* $\boldsymbol{u}_{\mathrm{bg}}$, which is a known solution to (1)–(2) in the whole physical space, ignoring all particles and walls. The total flow $\boldsymbol{u}$ in the presence of particles and walls is thus written as

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}_{\mathrm{bg}}(\boldsymbol{x}) + \boldsymbol{u}_{\mathrm{d}}(\boldsymbol{x}), \tag{10}$$

where $\boldsymbol{u}_{\mathrm{d}}$ is a *disturbance flow* which is responsible for enforcing the no-slip boundary conditions on the solid boundary Γ. As $\boldsymbol{x}$ moves away from Γ, the disturbance flow $\boldsymbol{u}_{\mathrm{d}}$ should decay to zero, and the total flow should therefore approach the background flow $\boldsymbol{u}_{\mathrm{bg}}$. The disturbance flow is written as

$$\boldsymbol{u}_{\mathrm{d}}(\boldsymbol{x}) = \boldsymbol{\mathcal{D}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}) + \sum_{\alpha=1}^{M} \boldsymbol{\mathcal{V}}^{(\alpha)}[\boldsymbol{F}^{(\alpha)}, \boldsymbol{\tau}^{(\alpha)}](\boldsymbol{x}), \tag{11}$$

[§] Here, δ_{ij} denotes the Kronecker delta, and ϵ_{ijk} is the Levi-Civita symbol.

where $\mathcal{V}^{(\alpha)}$ is as in (8), and the double layer density $\boldsymbol{q}$ must be determined through the boundary conditions. Note that $\boldsymbol{u}_{\mathrm{d}}$ as given by (11) decays as $\boldsymbol{x} \to \infty$, and by the superposition principle it satisfies (1)–(2). Also note that completion sources are placed inside the particles since the flow is external to the particles, but not inside the walls since the flow is internal to the walls (for details we refer to Pozrikidis [8, sec. 4.5]). On the other hand, the double layer density $\boldsymbol{q}$ is defined on the surfaces of both the particles and walls. The formulation (11) is complete, meaning that any flow which satisfies (1)–(2) and decays as $\boldsymbol{x} \to \infty$ can be represented in this way.

To derive the fundamental boundary integral equation, which is used to determine the double layer density $\boldsymbol{q}$ in (11), we insert (11) into (10) and then let $\boldsymbol{x} \in \Omega$ approach the solid boundary Γ. Enforcing no-slip boundary conditions on Γ yields, recalling the jump condition (4),

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}_{\mathrm{bg}}(\boldsymbol{x}) + \mathcal{D}[\Gamma, \boldsymbol{q}](\boldsymbol{x}) - 4\pi\boldsymbol{q}(\boldsymbol{x}) + \sum_{\alpha=1}^{M} \mathcal{V}^{(\alpha)}[\boldsymbol{F}^{(\alpha)}, \boldsymbol{\tau}^{(\alpha)}](\boldsymbol{x}) = \boldsymbol{U}_{\Gamma}(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Gamma. \tag{12}$$

The presence of the term $-4\pi\boldsymbol{q}(\boldsymbol{x})$, which is due to the jump condition, makes the boundary integral equation (12) a Fredholm integral equation of the second kind. The right-hand side $\boldsymbol{U}_{\Gamma}$ is the pointwise velocity of the boundary Γ. We assume the walls to be stationary and the particles to move as rigid bodies. This means that, if we let Γ_{w} be the union of all wall surfaces and $\Gamma_{\mathrm{p}}^{(\alpha)}$ the surface of particle α,

$$\boldsymbol{U}_{\Gamma}(\boldsymbol{x}) = \begin{cases} \boldsymbol{0}, & \boldsymbol{x} \in \Gamma_{\mathrm{w}}, \\ \boldsymbol{U}_{\mathrm{RBM}}^{(\alpha)} + \boldsymbol{\Omega}_{\mathrm{RBM}}^{(\alpha)} \times (\boldsymbol{x} - \boldsymbol{y}_{\mathrm{c}}^{(\alpha)}), & \boldsymbol{x} \in \Gamma_{\mathrm{p}}^{(\alpha)}, \end{cases} \tag{13}$$

where $\boldsymbol{U}_{\mathrm{RBM}}^{(\alpha)}$ and $\boldsymbol{\Omega}_{\mathrm{RBM}}^{(\alpha)}$ are the translational and angular velocity, respectively, of particle α (with RBM denoting rigid body motion).

As mentioned in Section 1, the viscous resistance that the particles experience from the fluid is related to their velocities. In the *resistance problem*, the velocities (i.e., $\boldsymbol{U}_{\mathrm{RBM}}^{(\alpha)}$ and $\boldsymbol{\Omega}_{\mathrm{RBM}}^{(\alpha)}$ for each particle) are specified in (12)–(13), while in the *mobility problem*, the viscous forces and torques (i.e., $\boldsymbol{F}^{(\alpha)}$ and $\boldsymbol{\tau}^{(\alpha)}$ for each particle) are specified [8, p. 129]. The boundary integral equations resulting from these two problems are described in more detail below. In both cases, the resulting integral equation is discretized using the Nyström method, as described in Section 3.

2.1.1 | The resistance problem

In this case, the velocities $\boldsymbol{U}_{\mathrm{RBM}}^{(\alpha)}$ and $\boldsymbol{\Omega}_{\mathrm{RBM}}^{(\alpha)}$ of all particles are known, while the corresponding forces $\boldsymbol{F}^{(\alpha)}$ and torques $\boldsymbol{\tau}^{(\alpha)}$ are to be computed. Following Pozrikidis [8, p. 130], the forces and torques are related to the unknown double layer density $\boldsymbol{q}$ by stipulating

$$\boldsymbol{F}^{(\alpha)}[\boldsymbol{q}] = \int_{\Gamma_{\mathrm{p}}^{(\alpha)}} \boldsymbol{q}(\boldsymbol{y}) \,\mathrm{d}S_{\boldsymbol{y}} \quad \text{and} \quad \boldsymbol{\tau}^{(\alpha)}[\boldsymbol{q}] = \int_{\Gamma_{\mathrm{p}}^{(\alpha)}} (\boldsymbol{y} - \boldsymbol{y}_{\mathrm{c}}^{(\alpha)}) \times \boldsymbol{q}(\boldsymbol{y}) \,\mathrm{d}S_{\boldsymbol{y}}. \tag{14}$$

These relations are inserted into (12), which can then be rearranged as

$$\mathcal{D}[\Gamma, \boldsymbol{q}](\boldsymbol{x}) - 4\pi\boldsymbol{q}(\boldsymbol{x}) + \sum_{\alpha=1}^{M} \mathcal{V}^{(\alpha)}[\boldsymbol{F}^{(\alpha)}[\boldsymbol{q}], \boldsymbol{\tau}^{(\alpha)}[\boldsymbol{q}]](\boldsymbol{x}) = \boldsymbol{U}_{\Gamma}(\boldsymbol{x}) - \boldsymbol{u}_{\mathrm{bg}}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma. \tag{15}$$

After solving this integral equation for $\boldsymbol{q}$, the forces and torques can be computed using (14), and the flow field can then be computed using (10)–(11).

2.1.2 | The mobility problem

In this case, the force $\boldsymbol{F}^{(\alpha)}$ and torque $\boldsymbol{\tau}^{(\alpha)}$ exerted on the fluid by each particle (which for a noninertial particle are equal to the net external force and torque acting on the particle) are known, but not the particle velocities $\boldsymbol{U}_{\mathrm{RBM}}^{(\alpha)}$ and $\boldsymbol{\Omega}_{\mathrm{RBM}}^{(\alpha)}$.

Following Pozrikidis [8, p. 135], the velocities are related to the double layer density $\boldsymbol{q}$ by

$$\boldsymbol{U}^{(\alpha)}_{\mathrm{RBM}}[\boldsymbol{q}] = -\frac{4\pi}{|\Gamma^{(\alpha)}_{\mathrm{p}}|}\int_{\Gamma^{(\alpha)}_{\mathrm{p}}} \boldsymbol{q}(\boldsymbol{y})\ \mathrm{d}S_{\boldsymbol{y}}, \tag{16}$$

$$\boldsymbol{\Omega}^{(\alpha)}_{\mathrm{RBM}}[\boldsymbol{q}] = -4\pi\sum_{n=1}^{3}\frac{\boldsymbol{\omega}^{(\alpha)}_n}{A^{(\alpha)}_n}\left(\boldsymbol{\omega}^{(\alpha)}_n \cdot \int_{\Gamma^{(\alpha)}_{\mathrm{p}}} (\boldsymbol{y}-\boldsymbol{y}^{(\alpha)}_{\mathrm{c}}) \times \boldsymbol{q}(\boldsymbol{y})\ \mathrm{d}S_{\boldsymbol{y}}\right). \tag{17}$$

Here, $|\Gamma^{(\alpha)}_{\mathrm{p}}|$ is the surface area of $\Gamma^{(\alpha)}_{\mathrm{p}}$, and

$$A^{(\alpha)}_n = \int_{\Gamma^{(\alpha)}_{\mathrm{p}}} \left|\boldsymbol{\omega}^{(\alpha)}_n \times (\boldsymbol{y}-\boldsymbol{y}^{(\alpha)}_{\mathrm{c}})\right|^2\ \mathrm{d}S_{\boldsymbol{y}}, \tag{18}$$

while $\boldsymbol{\omega}^{(\alpha)}_n$ are three linearly independent unit vectors which must satisfy

$$\frac{1}{\sqrt{A^{(\alpha)}_m A^{(\alpha)}_n}}\int_{\Gamma^{(\alpha)}_{\mathrm{p}}} \left[\boldsymbol{\omega}^{(\alpha)}_m \times (\boldsymbol{y}-\boldsymbol{y}^{(\alpha)}_{\mathrm{c}})\right] \cdot \left[\boldsymbol{\omega}^{(\alpha)}_n \times (\boldsymbol{y}-\boldsymbol{y}^{(\alpha)}_{\mathrm{c}})\right]\ \mathrm{d}S_{\boldsymbol{y}} = \delta_{mn}, \quad m,n = 1,2,3. \tag{19}$$

The boundary integral Equation (12) can then be rearranged as

$$\boldsymbol{\mathcal{D}}[\Gamma,\boldsymbol{q}](\boldsymbol{x}) - 4\pi\boldsymbol{q}(\boldsymbol{x}) - \boldsymbol{U}_{\Gamma}[\boldsymbol{q}](\boldsymbol{x}) = -\boldsymbol{u}_{\mathrm{bg}}(\boldsymbol{x}) - \sum_{\alpha=1}^{M}\boldsymbol{\mathcal{V}}^{(\alpha)}[\boldsymbol{F}^{(\alpha)},\boldsymbol{\tau}^{(\alpha)}](\boldsymbol{x}), \quad \boldsymbol{x}\in\Gamma, \tag{20}$$

where $\boldsymbol{U}_{\Gamma}[\boldsymbol{q}]$ is given by (13) but with $\boldsymbol{U}^{(\alpha)}_{\mathrm{RBM}}$ and $\boldsymbol{\Omega}^{(\alpha)}_{\mathrm{RBM}}$ replaced by the expressions in (16) and (17), respectively. After solving (20) for $\boldsymbol{q}$, the velocities can be computed using (16)–(17), and the flow field can be computed using (10)–(11).

3 | DISCRETIZATION AND QUADRATURE

In order to solve the boundary integral equation (15) associated with the resistance problem, or the boundary integral Equation (20) associated with the mobility problem, the integral operators in these equations must be discretized. This amounts to discretizing the double layer potential $\boldsymbol{\mathcal{D}}$, as well as the integrals occurring in relation (14) for the resistance problem, or relation (16)–(17) for the mobility problem. Following af Klinteberg and Tornberg,[37] we introduce the notation

$$\mathrm{I}[f] = \int_{\Gamma} f(\boldsymbol{y})\ \mathrm{d}S(\boldsymbol{y}) \tag{21}$$

for the integral of the arbitrary function f over the surface Γ. We introduce a quadrature rule Q_N called the *direct quadrature rule*, defined by a set of N nodes $\boldsymbol{x}_i \in \Gamma$ and weights $w_i \in \mathbb{R}$, $i = 1, \dots, N$. The details of this quadrature rule is specified in Sections 3.1 and 3.2. Using the direct quadrature rule Q_N, the integral in (21) can be approximated as

$$\mathrm{I}[f] \approx Q_N[f] = \sum_{i=1}^{N} f(\boldsymbol{x}_i) w_i. \tag{22}$$

Given a discretization of Γ, that is, a set of nodes $\{\boldsymbol{x}_i\}$, we define a grid spacing h (the exact definition of h is given for particles by equation (31) and for walls by Equation (33) below). An integral quantity approximated by Q_N on this discretization is then denoted by a superscript h; for example, the double layer potential

$$\mathcal{D}^h_i[\Gamma,\boldsymbol{q}](\boldsymbol{x}) = Q_N[T_{ijk}(\boldsymbol{x}-\cdot)q_j(\cdot)n_k(\cdot)]. \tag{23}$$

We then discretize (15) or (20) using the *Nyström method* [9, ch. 4, 10, sec. 12.2], in which the integral equation is enforced in the quadrature nodes. For the resistance problem, (15) then becomes

$$\boldsymbol{\mathcal{D}}^h[\Gamma, \boldsymbol{q}](\boldsymbol{x}_i) - 4\pi\boldsymbol{q}(\boldsymbol{x}_i) + \sum_{\alpha=1}^{M} \boldsymbol{\mathcal{V}}^{(\alpha),h}[\boldsymbol{F}^{(\alpha)}[\boldsymbol{q}], \boldsymbol{\tau}^{(\alpha)}[\boldsymbol{q}]](\boldsymbol{x}_i) = \boldsymbol{U}_\Gamma(\boldsymbol{x}_i) - \boldsymbol{u}_{\mathrm{bg}}(\boldsymbol{x}_i), \quad i = 1, \dots, N. \tag{24}$$

For the mobility problem, (20) becomes

$$\boldsymbol{\mathcal{D}}^h[\Gamma, \boldsymbol{q}](\boldsymbol{x}_i) - 4\pi\boldsymbol{q}(\boldsymbol{x}_i) - \boldsymbol{U}_\Gamma^h[\boldsymbol{q}](\boldsymbol{x}_i) = -\boldsymbol{u}_{\mathrm{bg}}(\boldsymbol{x}_i) - \sum_{\alpha=1}^{M} \boldsymbol{\mathcal{V}}^{(\alpha)}[\boldsymbol{F}^{(\alpha)}, \boldsymbol{\tau}^{(\alpha)}](\boldsymbol{x}_i), \quad i = 1, \dots, N. \tag{25}$$

The superscript h on $\boldsymbol{\mathcal{V}}^{(\alpha),h}$ in (24) and $\boldsymbol{U}_\Gamma^h$ in (25) signifies that these quantities, while not integrals themselves, contain integrals—namely, (14) or (16)–(17)— which are approximated using the direct quadrature rule Q_N. In both cases, the resulting linear system is solved iteratively using the GMRES.

In this article, we consider two distinct types of geometrical objects, namely, particles and walls, as indicated by Figure 2. Particles are mobile rigid bodies immersed in the fluid, while walls are stationary and surround the fluid domain. We consider two types of particles: spheroids, which are given by a surface

$$\frac{x_1^2 + x_2^2}{a^2} + \frac{x_3^2}{c^2} = 1 \tag{26}$$

in local coordinates; and rods, which consist of a cylinder with rounded caps, described in Appendix B. We also consider two types of walls, namely, plane walls and pipes with circular cross-section. Both wall geometries extend to infinity in the periodic setting, but we discretize only the part of each object that lies inside the primary cell.

The nature of the direct quadrature rule Q_N is different for particles and walls: for particles, it is a particle-global quadrature rule described in Section 3.1, while for walls it is a local patch-based quadrature rule described in Section 3.2. The special quadrature method for particles and walls is introduced in Section 3.3.

It should be noted that all geometrical objects shown in Figure 2 are smooth, that is, of class C^∞. The construction of the smooth rod particles is described in Appendix B. In Section 8, we consider the effect that a nonsmooth object would have on the convergence of the special quadrature method.

3.1 | Direct quadrature for particles

The discretization and direct quadrature rule of the spheroids are exactly the same as in Reference 37, while for the rods they are a slight variation of the former. Both kinds of particles are axisymmetric, and their parametrizations take this into

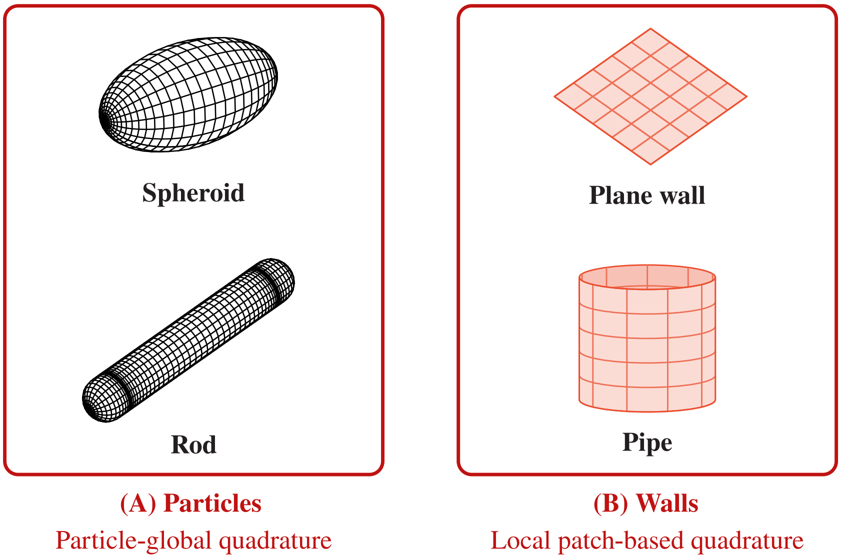


FIGURE 2 The geometrical objects considered in this article are (A) particles (spheroids and rods) and (B) walls (plane walls and pipes). The quadrature rule is different for particles and walls [Colour figure can be viewed at wileyonlinelibrary.com]

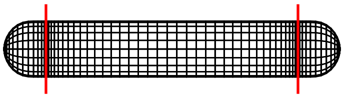


FIGURE 3 The grid on the rods consists of three parts: Two caps and a middle cylinder [Colour figure can be viewed at wileyonlinelibrary.com]

account, with one parameter $\varphi \in [0, 2\pi)$ varying in the azimuthal direction and the other parameter $\theta \in [0, \pi]$ varying in the polar direction. For instance, the spheroid (26) is parametrized using spherical coordinates as

$$\begin{cases} x_1 = a \sin\theta \cos\varphi, \\ x_2 = a \sin\theta \sin\varphi, \\ x_3 = c \cos\theta. \end{cases} \tag{27}$$

It is discretized using a tensorial grid with $n_\theta \times n_\varphi$ grid points. For the polar direction, let $(\theta_i, \lambda_i^\theta)$, $i = 1, \dots, n_\theta$, be the nodes and weights of an n_θ-point Gauss–Legendre quadrature rule [68, sec.3.5(v)] on the interval $[0, \pi]$. For the azimuthal direction, let $(\varphi_j, \lambda_j^\varphi)$, $j = 1, \dots, n_\varphi$, be the nodes and weights of the trapezoidal rule on the interval $[0, 2\pi)$. Since the integrand is periodic on this interval, the trapezoidal rule has spectral accuracy in this case.[69] The resulting tensorial quadrature rule, called the direct quadrature rule of the spheroid, is

$$Q_{n_\theta n_\varphi}[f] = \sum_{i=1}^{n_\theta} \sum_{j=1}^{n_\varphi} f(\boldsymbol{x}(\theta_i, \varphi_j)) W_{\text{sph}}(\theta_i, \varphi_j) \lambda_i^\theta \lambda_j^\varphi, \tag{28}$$

where $W_{\text{sph}}(\theta, \varphi)$ is the area element associated with the parametrization (27).

The rod consists of a cylinder with rounded caps. While the surface is smooth everywhere, the grid is divided into three parts as shown in Figure 3. The reason for this is to be able to set the resolution at the caps independently of the resolution at the middle of the rod; the resolution may need to be higher at the caps than at the middle due to the larger curvature.[¶] This also makes it possible to increase the resolution locally at a cap which is very close to another surface.

The rod is parametrized as

$$\begin{cases} x_1 = \varrho(\theta; L, R) \cos\varphi, \\ x_2 = \varrho(\theta; L, R) \sin\varphi, \\ x_3 = \beta(\theta; L, R), \end{cases} \tag{29}$$

where L is the length of the rod and R its radius. The shape functions $\varrho(\cdot\,; L, R) : [0, \pi] \to [0, R]$ and $\beta(\cdot\,; L, R) : [0, \pi] \to [-\frac{1}{2}L, \frac{1}{2}L]$ are described in Appendix B. They are chosen such that $\theta \in [0, \pi/3] = I_1$ and $\theta \in [2\pi/3, \pi] = I_3$ correspond to the two caps, while $\theta \in [\pi/3, 2\pi/3] = I_2$ corresponds to the middle part. Each cap is discretized using $n_1 \times n_\varphi$ grid points, and the middle cylinder is discretized using $n_2 \times n_\varphi$ grid points, so the total grid has $(2n_1 + n_2) \times n_\varphi$ grid points. The trapezoidal rule is again used in the azimuthal direction. In the polar direction, a separate Gauss–Legendre quadrature rule is used for each of the three parts. The tensorial direct quadrature rule of the rod is thus (with $n_3 = n_1$)

$$Q_{(2n_1+n_2)n_\varphi}[f] = \sum_{k=1}^{3} \sum_{i=1}^{n_k} \sum_{j=1}^{n_\varphi} f(\boldsymbol{x}(\theta_i^k, \varphi_j)) W_{\text{rod}}(\theta_i^k, \varphi_j) \lambda_i^k \lambda_j^\varphi, \tag{30}$$

where $(\theta_i^k, \lambda_i^k)$, $i = 1, \dots, n_k$, are the nodes and weights of an n_k-point Gauss–Legendre quadrature on the interval I_k, and $W_{\text{rod}}(\theta, \varphi)$ is the area element associated with (29).

For both spheroids and rods, we define the grid spacing h as the distance between adjacent grid points in the azimuthal direction at the equator of the particle, that is,

$$h = \frac{2\pi R}{n_\varphi} \text{ for rods}, \quad h = \frac{2\pi a}{n_\varphi} \text{ for spheroids}, \tag{31}$$

[¶] We also tested a discretization of the rod using a grid spanning the whole rod without dividing it into parts. We did not find any significant improvement in the quadrature error or computational cost from using such a grid rather than the one shown in Figure 3.

where n_φ is the number of grid points in the azimuthal direction, R is the radius of the rod and a is the equatorial semiaxis of the spheroid; c.f. (27) and (29), respectively.

The direct quadrature rules (28) and (30) are both particle-global in the sense that each particle is treated as a single unit, and the quadrature rule is applied to the particle as a whole. The quadrature rules has spectral accuracy for smooth integrands, that is, it converges exponentially as the number of grid points increases.

3.2 | Direct quadrature for walls

The wall geometries are present only in the periodic setting, and then only the part inside the primary cell needs to be discretized. For the plane wall, this part consists of a flat rectangle of size $L_1 \times L_2$, which is divided into $P_1 \times P_2$ flat subrectangles, called patches. Each patch is discretized using a tensorial grid with $n_1 \times n_2$ Gauss–Legendre grid points, as shown in Figure 4(A). In each direction of the patch, an n_d-point Gauss–Legendre quadrature rule is used with nodes and weights (s_i^d, λ_i^d), $i = 1, \ldots, n_d$, $d = 1, 2$. The resulting tensorial direct quadrature rule of the patch is

$$Q_{n_1 n_2}[f] = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f(\boldsymbol{x}(s_i^1, s_j^2)) W_{\text{wall}}(s_i^1, s_j^2) \lambda_i^1 \lambda_j^2, \tag{32}$$

where $W_{\text{wall}}(s^1, s^2)$ is the area element of the wall.

The part of the pipe in the primary cell consists of a cylinder with radius R_c and length L_c. Like the plane wall, it is divided into rectangular patches, but these are curved, as seen in Figure 4(B). Apart from that, the discretization and quadrature rule are the same as for the plane wall. The direct quadrature rule of the pipe is thus also given by (32), the only difference compared with the plane wall being the area element W_{wall} and the parametrization $(s^1, s^2) \mapsto \boldsymbol{x}$.

For both plane walls and pipes, we define the grid spacing h as

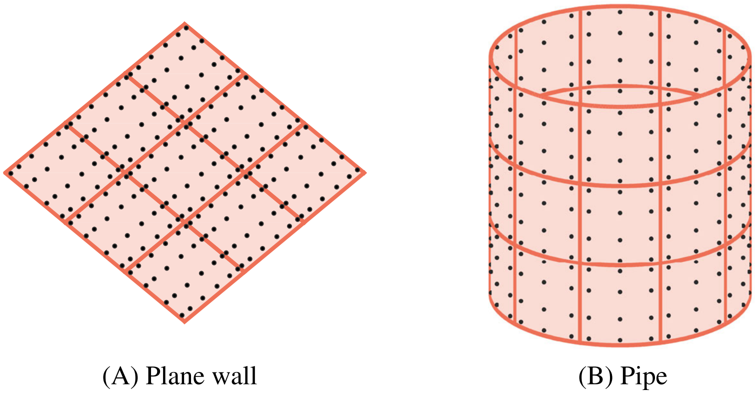
$$h = \max(h_1, h_2), \tag{33}$$

where h_1 and h_2 are the largest spacings between adjacent grid points in each of the two tensorial directions of the patches.

The direct quadrature rule (32) is local in the sense that the wall is subdivided into smaller patches, and the quadrature rule is applied to each patch separately. The grid can be refined in two different ways: by adding more grid points to each patch (which we call n-refinement), or by reducing the size of the patches and thus increasing their number (which we call P-refinement). Under n-refinement, the quadrature rule has spectral accuracy like the direct quadrature rule of the particles, while under P-refinement the quadrature rule is polynomially accurate with order determined by n_1 and n_2.

3.3 | Special quadrature: upsampled quadrature and QBX

The double layer potential $\boldsymbol{D}$ given by (3) is challenging to compute using direct quadrature in two different situations, in both cases due to its kernel $\boldsymbol{T}$. First, when the evaluation point $\boldsymbol{x}$ is on Γ itself, $\boldsymbol{T}$ becomes singular at the point $\boldsymbol{y} = \boldsymbol{x}$ (we


(A) Plane wall

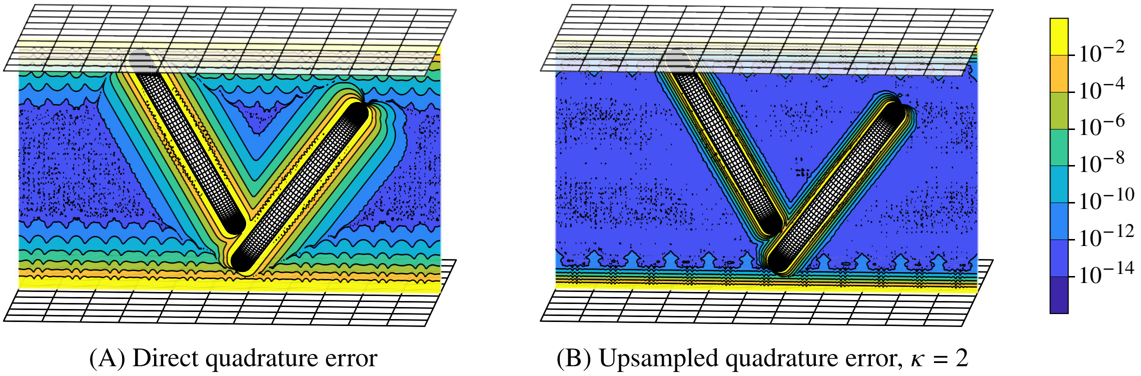
(B) Pipe

FIGURE 4 (A) A plane wall divided into 3 × 3 patches, each patch discretized using 6 × 4 grid points. (B) A pipe divided into 3 × 10 patches, each patch discretized using 4 × 3 grid points [Colour figure can be viewed at wileyonlinelibrary.com] refer to this as the *singular case* or the *onsurface evaluation case*). The integral exists as an improper integral as long as Γ is a Lyapunov surface [8, p. 37], but clearly a special quadrature method of some sort is needed to compute it. Second, when $\boldsymbol{x} \in \Omega$ is close to Γ, but not on Γ, $\boldsymbol{T}$ becomes very peaked and thus hard to resolve using the direct quadrature rule (we refer to this as the *nearly singular case* or the *offsurface evaluation case*).

The singular case is always present when solving the boundary integral Equation (12), while the nearly singular case occurs when particles are close to each other or close to a wall, and also if the flow field (10)–(11) is to be computed close to a particle or wall. The latter situation is shown in Figure 5(A), where the error grows exponentially as the evaluation point $\boldsymbol{x}$ approaches the boundary Γ. This behavior is well-known, and in two dimensions there are error estimates available for the Laplace and Helmholtz potentials[36] and for the Stokes potential.[30] To compute the double layer potential accurately close to a particle or wall, *special quadrature* is needed. Here, we consider two types of special quadrature: upsampled quadrature and QBX.

Assuming that the density $\boldsymbol{q}$ itself is well-resolved on the grid, *upsampled quadrature* provides a partial solution for the nearly singular case. In upsampled quadrature, the double layer density $\boldsymbol{q}$ is interpolated onto a grid refined by a factor κ in both directions, and the integral is then evaluated using direct quadrature on the finer grid. For the particle-global quadrature rules in Section 3.1, the grid of the whole particle is refined (increasing the number of grid points of each individual quadrature rule). The density is interpolated onto the finer grid using trigonometric interpolation in the azimuthal direction and barycentric Lagrange interpolation[70] in the polar direction. For the local patch-based quadrature rules in Section 3.2, only the N_{P} patches closest to the evaluation point $\boldsymbol{x}$ are refined, using n-refinement (thus increasing the number of grid points on them); other patches are sufficiently far away from the singularity that direct quadrature can be used. This is shown in Figure 6 for $N_{\mathrm{P}} = 9$. The refinement has spectral accuracy for both particles and walls. Since all geometrical objects are rigid, interpolation matrices can be precomputed.

As Figure 5(B) shows, upsampled quadrature pushes the region where the error is large closer to the boundary Γ. However, the error will always be large very close to Γ no matter how large the upsampling factor κ is. To be able to achieve small errors arbitrarily close to Γ, we use a special quadrature rule specifically designed for layer potentials with



(A) Direct quadrature error

(B) Upsampled quadrature error, $\kappa = 2$

FIGURE 5 Relative error in the center plane when evaluating the stresslet identity (5) for two rod particles between a pair of parallel horizontal plane walls in a periodic setting, using the direct quadrature rule in (A) and the upsampled quadrature rule with upsampling factor $\kappa = 2$ in (B). Note that the error is still large very close to the particles and walls in (B). The density $\boldsymbol{q}$ is constant in this example, so the error in (A) and (B) comes entirely from the nearly singular behavior of the stresslet $\boldsymbol{T}$ [Colour figure can be viewed at wileyonlinelibrary.com]

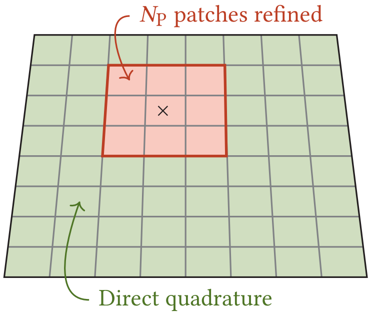


FIGURE 6 Upsampled quadrature for a plane wall: The N_{P} patches closest to the evaluation point (marked with ×) are refined, here for $N_{\mathrm{P}} = 9$. The other patches are treated using direct quadrature without refinement [Colour figure can be viewed at wileyonlinelibrary.com]

singular kernels, namely, QBX.[32,34] The idea behind QBX is to make a local series expansion of the potential $\boldsymbol{D}$ in the fluid domain, which converges rapidly since $\boldsymbol{D}$ is smooth all the way up to the boundary Γ. The expansion is made around a point $\boldsymbol{c}$, called the *expansion center*, which is inside the fluid domain (i.e., not on Γ), and it can be used to evaluate the potential inside a ball around $\boldsymbol{c}$ called the *ball of convergence*, as shown in Figure 7. The expansion is valid even at the point where the ball touches Γ, and can therefore be used in the singular case as well as the nearly singular case.[35] The application of QBX to the Stokes double layer potential $\boldsymbol{D}$ will be described in detail in Section 4.

In this article, we use a *combined quadrature* strategy, where direct quadrature is used far away from the boundary, upsampled quadrature is used in an intermediate region, and QBX is used in a small region closest to the boundary, as shown in Figure 8. For each particle and wall in the geometry, the evaluation point $\boldsymbol{x}$ is classified into one of these three regions, and the contribution to the double layer potential $\boldsymbol{D}$ from that particle or wall is computed as follows:

- If $\boldsymbol{x}$ is in the direct quadrature region, the double layer potential (3) is computed using direct quadrature (23) over the whole particle or wall, as described in Sections 3.1 and 3.2.
- If $\boldsymbol{x}$ is in the upsampled quadrature region, the behavior is different for particles and walls, as described above. For a particle, the density is upsampled globally on the whole particle surface and then integrated using direct quadrature on the fine grid. For a wall, the density is upsampled only on the N_P patches closest to the evaluation point, while direct quadrature without upsampling is used on the other patches (as in Figure 6).
- If $\boldsymbol{x}$ is in the QBX region, the behavior is similar to the upsampled quadrature region. For a particle, the density on the whole particle surface is used when computing the coefficients of the local expansion which is then used at the evaluation point (particle-global QBX). For a wall, only the density on the N_P patches closest to the expansion center $\boldsymbol{c}$ is used to compute the expansion (local QBX), while the contribution from other patches is computed using direct quadrature. In other words, the expansion is computed using a truncated wall, with N_P determining the number of patches in the truncated wall. The difference between particle-global and local QBX is described in more detail in Section 4.2.

The total double layer potential at $\boldsymbol{x}$ is then retrieved using superposition, that is, by summing the contributions from all particles and walls.

When using local QBX, the convergence rate of the local expansion will depend on the ratio between the distance from $\boldsymbol{c}$ to the wall and the distance from $\boldsymbol{c}$ to the edge of the truncated wall.[43] We have observed that $N_\text{P} = 1$ is too low for the wall QBX region in our case, since the expansion center may then be too close to the edge of the truncated wall. Setting

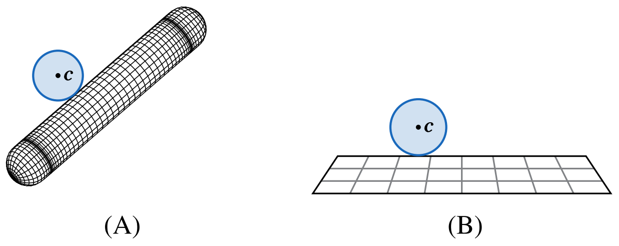


FIGURE 7 The idea behind quadrature by expansion is to make a series expansion of the potential close to a particle (A) or wall (B), valid inside a ball of convergence shown as a blue disc. The expansion is also valid at the point where the ball touches the boundary [Colour figure can be viewed at wileyonlinelibrary.com]

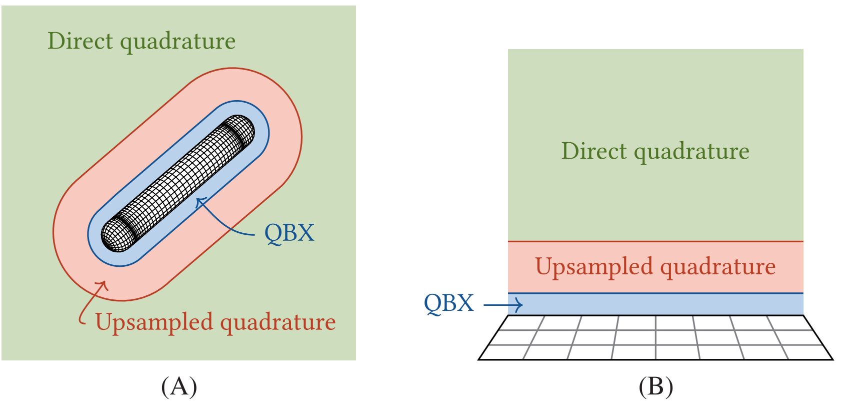


FIGURE 8 The regions of the combined quadrature strategy, shown here for a rod particle in (A) and a plane wall in (B). Depending on the location of the evaluation point $\boldsymbol{x}$, it is treated using direct quadrature, upsampled quadrature or quadrature by expansion [Colour figure can be viewed at wileyonlinelibrary.com]

$N_\mathrm{P} = 9$ seems to be sufficient to remedy this, and increasing N_P further has no effect. We therefore fix $N_\mathrm{P} = 9$ both for the QBX region and the upsampled quadrature region of plane walls and pipes, for the rest of this article.

The distances from the surface at which to switch from one quadrature region to the next (i.e., direct quadrature, upsampled quadrature, QBX) are parameters to be set, and these will be discussed in Section 6.

4 | QBX FOR THE STOKES DOUBLE LAYER POTENTIAL

In order to apply QBX to the Stokes double layer potential $\boldsymbol{\mathcal{D}}$ given by (3), we need to be able to write down a local expansion of the potential. We use the same approach as in Reference 37, which is summarized in Section 4.1. The differences between particles (for which particle-global QBX is used) and walls (for which local QBX is used) are summarized in Section 4.2. Finally, the precomputation scheme which is crucial for accelerating the method is described in Section 4.3.

4.1 | Local expansion of the double layer potential

Instead of expanding the double layer potential $\boldsymbol{\mathcal{D}}$ itself directly, we use the fact that $\boldsymbol{\mathcal{D}}$ can be expressed in terms of the so-called dipole potential $\mathcal{L}$ using the relation[37,71]

$$\mathcal{D}_i[\tilde{\Gamma}, \boldsymbol{q}](\boldsymbol{x}) = \left(x_j \frac{\partial}{\partial x_i} - \delta_{ij} \right) \mathcal{L}[\tilde{\Gamma}, q_j \boldsymbol{n} + n_j \boldsymbol{q}](\boldsymbol{x}) - \frac{\partial}{\partial x_i} \mathcal{L}[\tilde{\Gamma}, y_k q_k \boldsymbol{n} + y_k n_k \boldsymbol{q}](\boldsymbol{x}), \tag{34}$$

where $\tilde{\Gamma}$ is any subset of Γ. The dipole potential is the double layer potential of the Laplace equation and is defined as

$$\mathcal{L}[\tilde{\Gamma}, \boldsymbol{\rho}](\boldsymbol{x}) = \int_{\tilde{\Gamma}} \boldsymbol{\rho}(\boldsymbol{y}) \cdot \nabla_{\boldsymbol{y}} \frac{1}{|\boldsymbol{x} - \boldsymbol{y}|} \, \mathrm{d}S(\boldsymbol{y}). \tag{35}$$

The kernel of the dipole potential has a natural expansion based on the so-called Laplace expansion

$$\frac{1}{|\boldsymbol{x} - \boldsymbol{y}|} = \sum_{l=0}^{\infty} \frac{4\pi}{2l+1} \sum_{m=-l}^{l} r_x^l Y_l^{-m}(\theta_x, \varphi_x) \frac{1}{r_y^{l+1}} Y_l^m(\theta_y, \varphi_y), \tag{36}$$

where Y_l^m is the spherical harmonics function of degree l and order m (defined as in [35, eq. (3.5)]), while $(r_x, \theta_x, \varphi_x)$ and $(r_y, \theta_y, \varphi_y)$ are spherical coordinates of the points $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, with respect to a chosen expansion center $\boldsymbol{c}$, as shown in Figure 9. The expansion (36) is valid as long as $r_x < r_y$, that is, it can be used for all $\boldsymbol{x}$ within the ball of radius $r_\mathrm{QBX} = \min_{\boldsymbol{y} \in \tilde{\Gamma}} r_y$ centered at $\boldsymbol{c}$.

Inserting (36) into (35) leads to the expansion

$$\mathcal{L}[\tilde{\Gamma}, \boldsymbol{\rho}](\boldsymbol{x}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} r_x^l Y_l^{-m}(\theta_x, \varphi_x) z_{lm}[\boldsymbol{\rho}] \tag{37}$$

of the dipole potential, where the coefficients $z_{lm}[\boldsymbol{\rho}]$ are given by

$$z_{lm}[\boldsymbol{\rho}] = \frac{4\pi}{2l+1} \int_{\tilde{\Gamma}} \boldsymbol{\rho}(\boldsymbol{y}) \cdot \nabla_{\boldsymbol{y}} \frac{1}{r_y^{l+1}} Y_l^m(\theta_y, \varphi_y) \, \mathrm{d}S(\boldsymbol{y}). \tag{38}$$

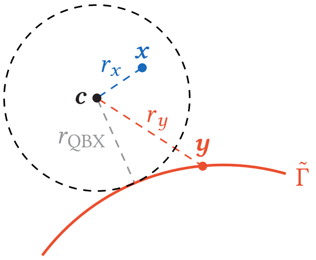


FIGURE 9 Illustration of the points $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{c}$, and the ball of convergence of (36) [Colour figure can be viewed at wileyonlinelibrary.com]

These coefficients are complex-valued due to Y_l^m, but the dipole potential $\mathcal{L}$ itself is real. Since the spherical harmonics functions satisfy $Y_l^{-m} = (Y_l^m)^*$, the coefficients also satisfy $z_{l,-m} = (z_{lm})^*$, where the asterisk denotes the complex conjugate. It is therefore enough to compute the coefficients for $m \geq 0$. The expansion (37) is in fact valid also at the point where the ball in Figure 9 touches $\tilde{\Gamma}$ (where $r_x = r_{\mathrm{QBX}}$), as established in Reference 35. This means that the expansion can be used both for offsurface evaluation (in the interior of the ball, where the double layer potential is nearly singular) as well as onsurface evaluation (at the point on $\tilde{\Gamma}$ closest to $\boldsymbol{c}$, where the double layer potential is singular).

Relation (34) allows us to express $\boldsymbol{D}$ using four dipole potentials with densities

$$\begin{aligned} \boldsymbol{\rho}^{(j)} &= q_j \boldsymbol{n} + n_j \boldsymbol{q}, \quad j = 1, 2, 3, \\ \boldsymbol{\rho}^{(4)} &= y_k q_k \boldsymbol{n} + y_k n_k \boldsymbol{q}. \end{aligned} \tag{39}$$

Each of the four dipole potentials is expanded using (37), with coefficients given by (38), which together with (34) provides a local expansion of the Stokes double layer potential.

In practice, the expansion (37) must be truncated, which is done at $l = l_{\max} = p_{\mathrm{QBX}}$. This results in the approximation

$$\mathcal{L}[\tilde{\Gamma}, \boldsymbol{\rho}^{(j)}](\boldsymbol{x}) \approx \mathcal{L}^{\mathrm{QBX}}[\tilde{\Gamma}, \boldsymbol{\rho}^{(j)}](\boldsymbol{x}) = \sum_{l=0}^{p_{\mathrm{QBX}}} \sum_{m=-l}^{l} r_x^l Y_l^{-m}(\theta_x, \varphi_x) z_{lm,j}^h, \quad j = 1, 2, 3, 4. \tag{40}$$

The coefficients $z_{lm,j}^h = z_{lm}^h[\boldsymbol{\rho}^{(j)}]$ are here computed using the upsampled quadrature rule introduced in Section 3.3, with upsampling factor $\kappa = \kappa_{\mathrm{QBX}}$. Upsampling is needed since the integrand in (38) becomes quite peaked for large l. However, the cost of upsampling can be entirely hidden in a precomputation step, as explained in Section 4.3. The number of coefficients that needs to be computed in (40) for each j is

$$N_{\mathrm{QBX}} = \frac{(p_{\mathrm{QBX}} + 1)(p_{\mathrm{QBX}} + 2)}{2}, \tag{41}$$

which takes into account that only coefficients with $m \geq 0$ need to be computed directly.

If the expansion (37) is absolutely convergent, the terms must decay in magnitude as $l \to \infty$. The size of the terms can be estimated using the bound

$$\left| \sum_{m=-l}^{l} r_x^l Y_l^{-m}(\theta_x, \varphi_x) z_{lm} \right| \leq r_{\mathrm{QBX}}^l \sqrt{\frac{2l+1}{4\pi}} \left(\sum_{m=-l}^{l} |z_{lm}|^2 \right)^{1/2}, \tag{42}$$

where we have used the fact that [35, eq. (3.36)]

$$\sum_{m=-l}^{l} |Y_l^m(\theta, \varphi)|^2 = \frac{2l+1}{4\pi}. \tag{43}$$

For a single dipole expansion such as (40) with a fixed j, the bound (42) with $z_{lm} = z_{lm,j}^h$ provides a way to estimate the decay of the terms and thus the truncation error of the truncated expansion. It is however not directly applicable to the Stokes double layer potential, which involves derivatives of dipole potentials as seen in (34). To estimate the truncation error for the Stokes double layer potential, we instead evaluate the error directly in the grid points, as explained in Section 6.1.2.

In summary, to compute $\boldsymbol{D}[\tilde{\Gamma}, \boldsymbol{q}](\boldsymbol{x})$ using QBX, the density $\boldsymbol{q}$ is first upsampled to a finer grid with upsampling factor κ_{QBX} and then converted into four dipole densities using (39). From these, four sets of dipole coefficients $z_{lm,j}^h$ are computed using the direct quadrature rule on the refined grid. The coefficients are used to evaluate the dipole potentials (40), from which the Stokes double layer potential $\boldsymbol{D}$ can be computed using (34). Note that the derivatives with respect to $\boldsymbol{x}$ in (34) can be computed analytically.

Since QBX can be used for both onsurface and offsurface evaluation, it is useful to introduce one expansion for each grid point. For each grid point $\boldsymbol{x}_i$ on the boundary, an expansion center $\boldsymbol{c}_i^+$ is thus placed at a distance r_{QBX} away from the boundary in the normal direction (i.e., in the fluid domain). This expansion center can be used to evaluate the double

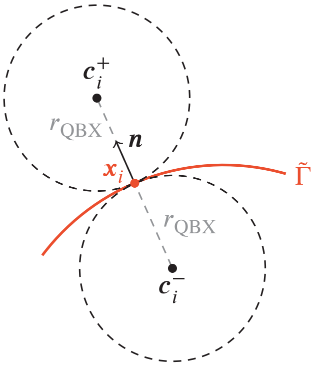


FIGURE 10 The two expansion centers $\boldsymbol{c}_i^+$ and $\boldsymbol{c}_i^-$ used for onsurface evaluation in a grid point $\boldsymbol{x}_i$ [Colour figure can be viewed at wileyonlinelibrary.com]

layer potential in a ball touching that grid point. In practice the balls of convergence of neighboring expansion centers will overlap, and for a given evaluation point the closest expansion center is used to evaluate the QBX potential.

For onsurface evaluation (but not offsurface evaluation), we also use a second expansion center $\boldsymbol{c}_i^-$ for each grid point, placed at a distance r_{QBX} away from the boundary in the *negative* normal direction (i.e., outside the fluid domain), as shown by Figure 10. The reason for this is that it significantly improves the convergence when solving the boundary integral equation using GMRES, since the spectrum of the discrete operator better matches that of the continuous operator, as was noted in References 32,33,37. Note that due to the jump condition (4), the correct value of the potential on $\tilde{\Gamma}$ is the average of the values from the two sides:

$$\boldsymbol{\mathcal{D}}[\tilde{\Gamma},\boldsymbol{q}](\boldsymbol{x}_i) = \frac{\boldsymbol{\mathcal{D}}^+[\tilde{\Gamma},\boldsymbol{q}](\boldsymbol{x}_i) + \boldsymbol{\mathcal{D}}^-[\tilde{\Gamma},\boldsymbol{q}](\boldsymbol{x}_i)}{2}, \tag{44}$$

where $\boldsymbol{\mathcal{D}}^+$ is the limit from the fluid domain and $\boldsymbol{\mathcal{D}}^-$ is the limit from the other side of $\tilde{\Gamma}$. While using two expansions may seem to double the computational cost, the extra cost appears only in the precomputation step, as described in Section 4.3, and thus does not affect the cost of evaluation itself.

There are two sources of error in the QBX approximation: the *truncation error* due to the fact that the expansion in (40) is truncated at $l = p_{\text{QBX}}$, and the *coefficient error* (called the "quadrature error" in References 35-37) due to the fact that the coefficients (38) are computed using a quadrature rule with finite precision. These two errors are controlled by the following three QBX parameters:

- The expansion radius r_{QBX}, which is the distance from the expansion center to $\tilde{\Gamma}$ and also the radius of the ball in which the expansion is valid. Increasing r_{QBX} makes the truncation error grow since the ball of convergence (and hence r_x) becomes larger, but the coefficient error decreases since the integrand in (38) becomes easier to resolve as r_y becomes larger.
- The expansion order p_{QBX}, which governs the number of terms to be included in the sum in (40). Increasing p_{QBX} makes the truncation error decrease since more terms are included, but the coefficient error grows since the integrand in (38) is harder to resolve for larger l.
- The upsampling factor κ_{QBX}, which governs the amount of grid refinement when computing the dipole coefficients (38). Increasing κ_{QBX} makes the coefficient error decrease since the resolution of the underlying quadrature rule increases.

A simple way to decrease both the truncation error and coefficient error is to increase p_{QBX} and κ_{QBX} simultaneously while keeping r_{QBX} fixed. We will continue to discuss how the QBX parameters should be selected to achieve a small overall error in Section 6. For a more in-depth analysis, we refer to Epstein et al.[35] for the truncation error, af Klinteberg and Tornberg[36] for the coefficient error, as well as the summary in [37, sec. 3.5].

4.2 | Global and local QBX

As was mentioned in Section 1.1, a QBX method can be either *(fully) global*, *particle-global* or *local*, the difference being which part of the boundary (i.e., which source points) to include when forming the local expansion. Here, we use particle-global QBX for particles and local QBX for walls. In essence, the difference between the three variants is what $\tilde{\Gamma}$ in Section 4.1 is taken to be:

- For a fully global QBX method, all grid points on the whole boundary are used to form the local expansion, that is, $\tilde{\Gamma} = \Gamma$.
- For a particle-global QBX method, all grid points on a single particle are used to form the expansion, that is, $\tilde{\Gamma} = \Gamma_{\mathrm{p}}^{(\alpha)}$, where $\Gamma_{\mathrm{p}}^{(\alpha)}$ is the surface of the particle with index α.
- For a local QBX method, only the grid points which are close to the expansion center are used to form the expansion. In our case, we choose $\tilde{\Gamma}$ to be the N_{P} patches of the wall which are closest to the expansion center, as shown in Figure 6. (The contribution from patches further away is not included in the expansion but computed using direct quadrature.)

Note that $\tilde{\Gamma}$ may depend on the location of the expansion center to be used, which in turn depends on the evaluation point. In principle, it is sufficient to let $\tilde{\Gamma}$ consist of the grid points close to the expansion center (i.e., local QBX), since that is where the integrand becomes nearly singular; for grid points further away, direct quadrature can be used. The reason to extend $\tilde{\Gamma}$ further is to improve the regularity of the layer potential that is being expanded, so that the expansion converges more rapidly. Indeed, in local QBX, the expanded layer potential consists of the contribution from a truncated part of the boundary, and may not be very smooth since $\tilde{\Gamma}$ ends abruptly. However, the larger $\tilde{\Gamma}$ is, the further away from the ball of convergence will the edge of $\tilde{\Gamma}$ be, and the less will it affect the convergence of the expansion. We have observed that $N_{\mathrm{P}} = 9$ is sufficient for $\tilde{\Gamma}$ for the walls.

In particle-global QBX, the expanded layer potential has the contribution from a whole particle, which consists of a closed and smooth surface, so the potential from it should be smooth. In a fully global QBX, the expanded layer potential is the global potential, which is smooth if Γ is regular enough. Unlike the particle-global QBX, the fully global QBX quickly becomes expensive unless a fast method (such as the FMM) is used to compute the far-field contribution. Therefore, the fully global QBX variant is not used in this article.

An advantage of the local and particle-global QBX variants over the fully global QBX is that, if the individual particles and walls are rigid, $\tilde{\Gamma}$ is the same (in local coordinates) for all particles or patches of the same shape, even if they have different orientations. This makes precomputation possible, which we shall return to in Section 4.3. Another advantage is that expansion centers can be placed without regards to other particles or walls, since each expansion contains only the contribution from a single particle or wall segment. In a fully global QBX method, each ball of convergence must be completely outside *all* particles and walls, which would complicate the placement of the expansion centers.

4.3 | Precomputation for QBX

The mapping given by (39) and the discrete version of (38), which takes the double layer density $\boldsymbol{q}$ on $\tilde{\Gamma}$ and returns the dipole coefficients $z_{lm,j}^{h}$ for a single expansion center $\boldsymbol{c}_i$, is a linear function of $\boldsymbol{q}$ and can therefore be represented by a matrix $\boldsymbol{M}_i$. This matrix is of size $4N_{\mathrm{QBX}} \times 3\tilde{N}$, where N_{QBX} is given by (41) and $\tilde{N}$ is the number of grid points on $\tilde{\Gamma}$ (before upsampling). There is one such matrix $\boldsymbol{M}_i$ for every expansion center, and it depends only on the geometry $\tilde{\Gamma}$, its discretization and the location of the expansion center in the local coordinates of $\tilde{\Gamma}$. For a rigid geometry $\tilde{\Gamma}$, such as in our case, the matrix $\boldsymbol{M}_i$ can therefore be precomputed and stored.

Note that the upsampling factor κ_{QBX} is effectively "hidden" in this precomputation step: upsampling influences the computation of $\boldsymbol{M}_i$ since $\boldsymbol{q}$ is upsampled before being inserted into (39), but it has no effect on the size of $\boldsymbol{M}_i$, which is set by the discretization of $\tilde{\Gamma}$ prior to upsampling. Therefore, upsampling does not affect the computational complexity of the method once $\boldsymbol{M}_i$ has been precomputed.

The matrix $\boldsymbol{M}_i$ which computes the coefficients $z_{lm,j}^{h}$ is used for *offsurface* evaluation, when the evaluation point is not known beforehand; the coefficients can then be used to evaluate the expansion at any evaluation point within the ball of convergence. For *onsurface* evaluation, that is, evaluation at one of the grid points of the boundary, the evaluation point itself is known beforehand and precomputation can be taken even further. In fact, the mapping that takes the expansion coefficients to the value of the potential $\boldsymbol{\mathcal{D}}[\tilde{\Gamma}, \boldsymbol{q}](\boldsymbol{x}_i)$, given by (40) and (34), is also linear and can therefore be represented by a matrix $\boldsymbol{S}_i$. This allows us to compute a matrix $\boldsymbol{R}_i = \boldsymbol{S}_i\boldsymbol{M}_i$ which maps the density $\boldsymbol{q}$ on $\tilde{\Gamma}$ directly to the value of the double layer potential $\boldsymbol{\mathcal{D}}$ at one of the grid points – effectively representing a set of target-specific quadrature weights for every grid point. The matrix $\boldsymbol{R}_i$ is of size $3 \times 3\tilde{N}$ and there is one such matrix for each grid point $\boldsymbol{x}_i$ on the boundary. Precomputing the $\boldsymbol{R}_i$ matrix hides not only κ_{QBX} but also p_{QBX}.

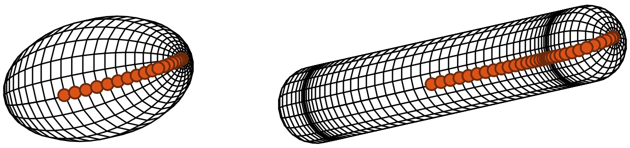


FIGURE 11 The matrices $\boldsymbol{M}_i$ and $\boldsymbol{R}_i$ need only be stored for the grid points along half a line of longitude, here indicated with red dots [Colour figure can be viewed at wileyonlinelibrary.com]

Since two expansion centers are used for onsurface evaluation, as the reader may recall from Figure 10, there are actually two $\boldsymbol{R}_i$ matrices for each grid point: $\boldsymbol{R}_i^+$ and $\boldsymbol{R}_i^-$, associated with $\boldsymbol{c}_i^+$ and $\boldsymbol{c}_i^-$, respectively. From (44), it is clear that these matrices can be combined as

$$\boldsymbol{R}_i = \frac{\boldsymbol{R}_i^+ + \boldsymbol{R}_i^-}{2} \tag{45}$$

to form a single matrix $\boldsymbol{R}_i$ for each grid point. This way, the extra cost of using two expansions is completely hidden in the precomputation step.

For the particles, the axisymmetry can be used to vastly reduce the amount of computations and storage needed to precompute the matrices $\boldsymbol{M}_i$ and $\boldsymbol{R}_i$. In fact, due to reflective symmetry, it suffices to compute $\boldsymbol{R}_i$ for the $n_\theta/2$ grid points ($n_1 + n_2/2$ grid points for rod particles) shown in Figure 11, and $\boldsymbol{M}_i$ for the corresponding expansion centers. The matrices for all other grid points and their expansion centers are then calculated using rotations and reflections, as in Reference 37. Note that if several particles of the same shape appear in a simulation, the precomputation only needs to be done for one such particle. This is true even if particles are scaled up or down by a constant factor, as the precomputed matrices can be scaled via homogeneity; if a particle (or wall) is scaled by a factor λ and r_{QBX} is scaled by the same factor, then the matrices $\boldsymbol{R}_i$ do not change while the matrices $\boldsymbol{M}_i$ which compute $z^h_{lm,j}$ are multiplied by λ^{-l} for $j = 1, 2, 3$ and λ^{-l+1} for $j = 4$, which can be seen from (38) and (39).

For a wall geometry with uniform patch size, as in our case, the geometry has a discrete translational symmetry for offsets equal to the patch size, due to periodicity. This means that the geometry "looks" exactly the same seen from any patch of the wall, and it is therefore enough to precompute the $\boldsymbol{M}_i$ and $\boldsymbol{R}_i$ matrices for the $n_1 n_2$ grid points and corresponding expansion centers of a single patch of the wall. In this case $\tilde{\Gamma}$ consists of that patch and its $N_{\text{P}} - 1$ closest neighbors, as indicated in Figure 6 for $N_{\text{P}} = 9$.

5 | PERIODICITY AND FAST METHODS

Up to this point we have not taken periodicity into account in the description of the mathematical formulation and its discretization; it is now time to remedy this. We will here give the details of the periodic formulation indicated in Figure 1(B), and in particular focus on how the special quadrature methods are combined with the fast summation method used for the periodic problem.

Consider a primary cell with side lengths $\boldsymbol{B} = (B_1, B_2, B_3)$ which is replicated periodically in all three spatial directions. The flow field is then periodic, that is, $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x} + \boldsymbol{k} \circ \boldsymbol{B})$ for any $\boldsymbol{k} \in \mathbb{Z}^3$, where $\circ$ denotes the Hadamard (element-wise) product. This changes the boundary integral formulation introduced in Section 2.1 in the following way: The layer potential $\boldsymbol{\mathcal{D}}$ and completion flow $\boldsymbol{\mathcal{V}}^{(\alpha)}$ which appear in the flow field (11) and in the fundamental boundary integral Equation (12) are replaced by their periodic counterparts $\boldsymbol{\mathcal{D}}^{3\text{P}}$ and $\boldsymbol{\mathcal{V}}^{(\alpha),3\text{P}}$. These are defined as infinite sums over the periodic lattice, that is,

$$\boldsymbol{\mathcal{D}}^{3\text{P}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \boldsymbol{\mathcal{D}}[\Gamma, \boldsymbol{q}](\boldsymbol{x} + \boldsymbol{k}\circ\boldsymbol{B}), \quad \boldsymbol{\mathcal{V}}^{(\alpha),3\text{P}}[\boldsymbol{F}, \boldsymbol{\tau}](\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \boldsymbol{\mathcal{V}}^{(\alpha)}[\boldsymbol{F}, \boldsymbol{\tau}](\boldsymbol{x} + \boldsymbol{k}\circ\boldsymbol{B}). \tag{46}$$

These sums converge slowly, and their value depends on the order of summation, so they cannot be computed using direct summation. We compute them using the SE method,[15,59] a fast Ewald summation method based on the fast Fourier transform (FFT). The SE method is described in detail for the stokeslet in Reference 59, for the stresslet in Reference 58 and for the rotlet in Reference 60, and has been combined with QBX previously in Reference 37. In the SE method, each of the periodic sums in (46) is split into two parts: the *real-space part*, which decays fast and can therefore be summed directly in real space; and the *Fourier-space part*, which is smooth and therefore decays fast in Fourier space.

No special treatment is needed for the completion flow $\boldsymbol{\mathcal{V}}^{(\alpha),3\text{P}}$ since the evaluation point is never close to the singular points (which are inside the particle), so the SE method as described in References 59,60 is used without modification.

For the double layer potential $\boldsymbol{\mathcal{D}}^{3\mathrm{P}}$, special quadrature is needed so SE must be combined with QBX and the upsampled quadrature rule. How this is done is described below.

The periodic sum for the double layer potential can be written explicitly as

$$\mathcal{D}_i^{3\mathrm{P}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \int_\Gamma T_{ijl}(\boldsymbol{x} + \boldsymbol{k}\circ\boldsymbol{B} - \boldsymbol{y}) q_j(\boldsymbol{y}) n_l(\boldsymbol{y}) \,\mathrm{d}S(\boldsymbol{y}). \tag{47}$$

The stresslet $\boldsymbol{T}$ is split into two parts

$$T_{ijl}(\boldsymbol{r}) = T_{ijl}^{\mathrm{R}}(\boldsymbol{r}; \xi) + T_{ijl}^{\mathrm{F}}(\boldsymbol{r}; \xi), \tag{48}$$

where $\boldsymbol{T}^{\mathrm{R}}$ is the real-space part and $\boldsymbol{T}^{\mathrm{F}}$ is the Fourier-space part. The Ewald parameter ξ is a positive number which is used to balance the decay of $\boldsymbol{T}^{\mathrm{R}}$ in real space and the decay of the Fourier coefficients of $\boldsymbol{T}^{\mathrm{F}}$ (a larger value of ξ makes the real-space part decay faster and the Fourier-space part decay slower, thus shifting computational work into Fourier space). In the split that we use, $\boldsymbol{T}^{\mathrm{R}}$ is given by References 37,58

$$\begin{aligned} T_{ijl}^{\mathrm{R}}(\boldsymbol{r}; \xi) = &-\frac{2}{r^4}\left(\frac{3}{r}\,\mathrm{erfc}(\xi r) + \frac{2\xi}{\sqrt{\pi}}(3 + 2\xi^2 r^2 - 4\xi^4 r^4)\mathrm{e}^{-\xi^2 r^2}\right) r_i r_j r_l \\ &+ \frac{8\xi^3}{\sqrt{\pi}}(2 - \xi^2 r^2)\mathrm{e}^{-\xi^2 r^2}(\delta_{ij} r_l + \delta_{jl} r_i + \delta_{li} r_j), \end{aligned} \tag{49}$$

where $r = |\boldsymbol{r}|$. The Fourier-space part is simply given by $\boldsymbol{T}^{\mathrm{F}} = \boldsymbol{T} - \boldsymbol{T}^{\mathrm{R}}$. Inserting (48) into (47) splits the periodic double layer potential into two parts $\boldsymbol{\mathcal{D}}^{3\mathrm{P}} = \boldsymbol{\mathcal{D}}^{3\mathrm{P,R}} + \boldsymbol{\mathcal{D}}^{3\mathrm{P,F}}$, where

$$\mathcal{D}_i^{3\mathrm{P,R}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}; \xi) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \int_\Gamma T_{ijl}^{\mathrm{R}}(\boldsymbol{x} + \boldsymbol{k}\circ\boldsymbol{B} - \boldsymbol{y}; \xi) q_j(\boldsymbol{y}) n_l(\boldsymbol{y}) \,\mathrm{d}S(\boldsymbol{y}), \tag{50}$$

$$\mathcal{D}_i^{3\mathrm{P,F}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}; \xi) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \int_\Gamma T_{ijl}^{\mathrm{F}}(\boldsymbol{x} + \boldsymbol{k}\circ\boldsymbol{B} - \boldsymbol{y}; \xi) q_j(\boldsymbol{y}) n_l(\boldsymbol{y}) \,\mathrm{d}S(\boldsymbol{y}). \tag{51}$$

The singularity of the stresslet is completely transferred to $\boldsymbol{T}^{\mathrm{R}}$, while $\boldsymbol{T}^{\mathrm{F}}$ is nonsingular.[37] The Fourier-space part (51) is computed using FFTs as described in Appendix D and Reference 58. The real-space potential (50) is evaluated in real space, and requires special quadrature due to the singularity of $\boldsymbol{T}^{\mathrm{R}}$, much as in the free-space setting. Note that since $\boldsymbol{T}^{\mathrm{R}}(\boldsymbol{r}; \xi)$ decays fast as $|\boldsymbol{r}| \to \infty$ it can be neglected for $|\boldsymbol{r}| > r_{\mathrm{c}}$, where r_{c} is called the *cutoff radius*. We can thus change the integration domain in (50) to $\Gamma^\star = \Gamma^\star(\boldsymbol{x}, \boldsymbol{k}; r_{\mathrm{c}}) = \{\boldsymbol{y} \in \Gamma : |\boldsymbol{x} + \boldsymbol{k}\circ\boldsymbol{B} - \boldsymbol{y}| \le r_{\mathrm{c}}\}$ and approximate

$$\mathcal{D}_i^{3\mathrm{P,R}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}; \xi) \approx \mathcal{D}_i^{3\mathrm{P,R}\star}[\Gamma, \boldsymbol{q}](\boldsymbol{x}; \xi) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \int_{\Gamma^\star} T_{ijl}^{\mathrm{R}}(\boldsymbol{x} + \boldsymbol{k}\circ\boldsymbol{B} - \boldsymbol{y}; \xi) q_j(\boldsymbol{y}) n_l(\boldsymbol{y}) \,\mathrm{d}S(\boldsymbol{y}). \tag{52}$$

The error of this approximation is determined by the product ξr_{c} as described in Reference 58. Rather than deriving a new QBX expansion from scratch for the real-space part $\boldsymbol{\mathcal{D}}^{3\mathrm{P,R}\star}$, we reuse the expansion of the total layer potential $\boldsymbol{\mathcal{D}}$ from Section 4. To be able to do this, we insert $\boldsymbol{T}^{\mathrm{R}} = \boldsymbol{T} - \boldsymbol{T}^{\mathrm{F}}$ into (52) to get

$$\mathcal{D}_i^{3\mathrm{P,R}\star}[\Gamma, \boldsymbol{q}](\boldsymbol{x}; \xi) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \int_{\Gamma^\star} T_{ijl}(\boldsymbol{x} + \boldsymbol{k}\circ\boldsymbol{B} - \boldsymbol{y}; \xi) q_j(\boldsymbol{y}) n_l(\boldsymbol{y}) \,\mathrm{d}S(\boldsymbol{y}) \tag{53}$$

$$- \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \int_{\Gamma^\star} T_{ijl}^{\mathrm{F}}(\boldsymbol{x} + \boldsymbol{k}\circ\boldsymbol{B} - \boldsymbol{y}; \xi) q_j(\boldsymbol{y}) n_l(\boldsymbol{y}) \,\mathrm{d}S(\boldsymbol{y}). \tag{54}$$

Note that the integration domain $\Gamma^\star$ ensures that both of these sums have few terms since r_{c} should be small – typically smaller than the size of the periodic cell. The integral in (53) represents the total layer potential from $\Gamma^\star$ and can thus be computed using the combined special quadrature method from Section 3.3, with truncation at r_c. The integral in (54) is computed using direct quadrature, which is possible since $\boldsymbol{T}^F$ is nonsingular.

As in the free-space setting in Section 3.3, the evaluation point $\boldsymbol{x}$ is classified into one of three regions (see Figure 8). The potential is evaluated using (52) in the direct quadrature region and (53)–(54) in the other two regions. The reader may wonder why we in the upsampled quadrature region do not simply evaluate (52) using upsampled quadrature. The reason is that (54) evaluated using the same quadrature method as in the Fourier-space part—that is, direct quadrature—is needed to cancel discretization errors in the latter.[#] These discretization errors may be larger than the SE error tolerance and are caused by the fact that $\boldsymbol{T}^F(\boldsymbol{r};\xi)$, while nonsingular, tends to become slightly peaked for small $|\boldsymbol{r}|$, that is, close to $\Gamma^\star$. Cancellation prevents these errors from influencing the error of the full method.

Another important point to note is that r_c must be chosen large enough so that no special quadrature is needed for the total potential when $|\boldsymbol{r}| > r_c$. This is because, for $|\boldsymbol{r}| > r_c$, the total potential is equal to the Fourier-space part, which is always computed using direct quadrature. Thus, r_c must be at least as large as the distance from Γ to the direct quadrature region.

6 | PARAMETER SELECTION

In this section, we develop our strategy for selecting the parameters of the combined special quadrature, that is, upsampled quadrature and QBX, when evaluating the Stokes double layer potential $\boldsymbol{D}$. We assume that a discretization of the geometry is given, with sufficient resolution for the density to be well-resolved and the direct quadrature to achieve a given error tolerance ε_{tol} at a given distance (sufficiently far away) from all surfaces. The goal is to select quadrature parameters for each particle and wall so that the error tolerance ε_{tol} is achieved also in the upsampled quadrature region and QBX region. Of course, there are many different ways to choose the parameters, some resulting in higher computational efficiency than others. Here, we do not aim to optimize the efficiency; instead, our focus is on achieving the given error tolerance at an acceptable (albeit not optimal) computational cost.

The parameters that must be selected are shown in Figure 12. Note that we allow for multiple upsampled quadrature regions with different upsampling factors $\kappa_{\text{U}i}$, in order to gradually increase the upsampling closer to the surface. Due to the precomputation scheme for QBX, using QBX may in fact be faster than using upsampled quadrature with the same upsampling factor. Therefore, the QBX region may extend further away from the surface than the expansion center (i.e., d_{QBX} may be larger than r_{QBX}, but of course not larger than $2r_{\text{QBX}}$).

The parameters to be selected are as follows:

- The threshold distances $d_{\text{U}i}$ for the upsampled quadrature regions, $i = 1, 2, \ldots, N_{\text{U}}$, and the threshold distance d_{QBX} for the QBX region. These determine at what distance from the surface each region starts. If $d_\Gamma(\boldsymbol{x})$ is the distance from the evaluation point $\boldsymbol{x}$ to the surface Γ, then $\boldsymbol{x}$ belongs to the ith upsampled quadrature region if

$$d_{\text{U}i} \geq d_\Gamma(\boldsymbol{x}) \geq \begin{cases} d_{\text{U}(i+1)} & \text{if } i < N_{\text{U}} \\ d_{\text{QBX}} & \text{if } i = N_{\text{U}}, \end{cases} \tag{55}$$

  and $\boldsymbol{x}$ belongs to the QBX region if $d_{\text{QBX}} \geq d_\Gamma(\boldsymbol{x}) \geq 0$. Each of these distances should be chosen so that the error does not exceed the tolerance in the region further away from the surface (e.g., $d_{\text{U}1}$ is selected based on the direct quadrature error).
- The upsampling factors $\kappa_{\text{U}i}$ for the upsampled quadrature regions, $i = 1, 2, \ldots, N_{\text{U}}$. These should be increasing, that is, $\kappa_{\text{U}1} < \kappa_{\text{U}2} < \ldots < \kappa_{\text{U}N_{\text{U}}}$. The upsampling factor $\kappa_{\text{U}i}$ determines the distance $d_{\text{U}(i+1)}$ at which the next region must begin, which we will come back to in Section 6.1.
- The QBX upsampling factor κ_{QBX}, which controls the amount of upsampling used when computing the coefficients in (38), and thus the QBX coefficient error as mentioned in Section 4.1. It should be chosen large enough so that the

[#] Due to the nonlocal nature of the Fourier transform, the Fourier-space part must be computed using the same quadrature method everywhere; here we use direct quadrature. Another possibility would be to use upsampled quadrature, but then upsampling would need to be done for *all* evaluation points, not only those in the upsampled quadrature region. In that case one might want to remove the direct quadrature region altogether and use only upsampled quadrature and QBX.

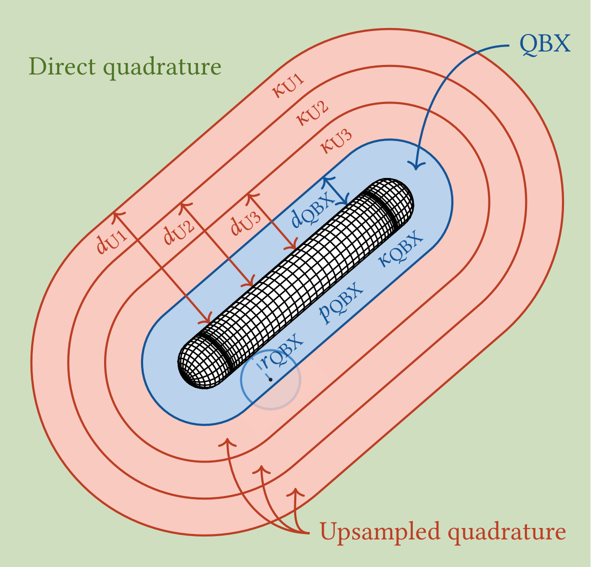


FIGURE 12 Parameters of the combined special quadrature, here for a rod with $N_U = 3$ upsampled quadrature regions (each geometrical object has its own set of corresponding parameters). Note that while the ball of convergence for QBX can be larger than the QBX region, the expansion is used only inside the QBX region. QBX, quadrature by expansion [Colour figure can be viewed at wileyonlinelibrary.com]

coefficient error and the truncation error are balanced. The upsampling factor influences the QBX precomputation time (which grows like $O(\kappa_{QBX}^2)$), but not the size of the precomputed $\boldsymbol{M}_i$ and $\boldsymbol{R}_i$ matrices, and thus not the evaluation time.

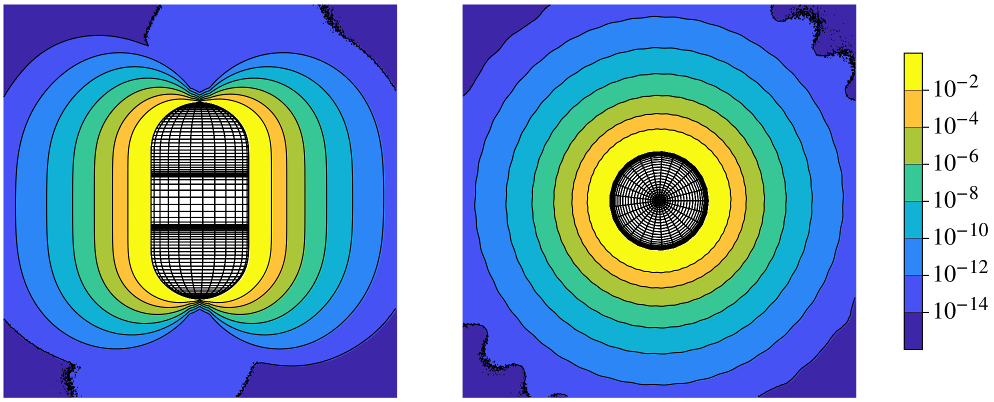
- The QBX expansion order p_{QBX}, which controls the number of terms included in the expansion in (40), and thus the QBX truncation error as mentioned in Section 4.1. It should be chosen so that the truncation error is below the error tolerance everywhere in the QBX region. The expansion order affects the size of the $\boldsymbol{M}_i$ matrix used for offsurface evaluation (which grows like $O(p_{QBX}^2)$), but not that of the $\boldsymbol{R}_i$ matrix used for onsurface evaluation. It should be noted that as p_{QBX} increases, the upsampling factor κ_{QBX} must also increase since higher-order coefficients are harder to resolve.
- The QBX expansion radius r_{QBX}, which affects both the coefficient error and the truncation error, but neither the precomputation time nor the evaluation time directly. It should typically be chosen as small as possible, since this speeds up the convergence of the expansion in (40) so that p_{QBX} can be chosen small. On the other hand, a very small r_{QBX} means that the upsampling factor κ_{QBX} must be large, since the expansion center moves closer to the surface.

  However, in our implementation the primary restriction on r_{QBX} is that it must be large enough for the balls of convergence to cover the QBX region sufficiently well. In general, r_{QBX} should not be smaller than the distance from one expansion center to the next, to ensure a good coverage. Since we have one expansion center per grid point, we require that r_{QBX} be not smaller than the grid spacing h (defined by (31) and (33) for particles and walls, respectively).[‖] Given h, it is useful to consider the ratio r_{QBX}/h when selecting parameters. As noted in Reference 37, this has the advantage that if r_{QBX}/h is kept fixed during refinement of the original grid, then the coefficient error is constant, assuming that the upsampling factor κ_{QBX} is also fixed. We will therefore consider r_{QBX}/h in the rest of this section.

Unfortunately, no general error estimates are available in three dimensions for the quadrature rules that we use here. The parameters must therefore be selected based on numerical experiments, and we present a strategy for doing so here. The idea is to start from the outermost upsampled quadrature region (U1) and then proceed inwards towards the surface of the particle or wall, determining the parameters in the following order:

1. Threshold distances and upsampling factors for the upsampled quadrature regions,
2. The QBX parameters r_{QBX}/h and d_{QBX},
3. The QBX parameters p_{QBX} and κ_{QBX}.

[‖] An alternative would be to introduce more expansion centers to maintain the coverage of the QBX region as r_{QBX} decreases below the grid spacing. Doing so would also increase the amount of work and storage needed in the precomputation step.



(A) Direct quadrature error, slice 1

(B) Direct quadrature error, slice 2

FIGURE 13 Error in two perpendicular slices (A) and (B) through the rod particle, when evaluating the stresslet identity (5) using direct quadrature in free space [Colour figure can be viewed at wileyonlinelibrary.com]

The process must be repeated for each type of particle and wall to be used. We develop the strategy in the context of a specific rod particle in Section 6.1; a summary of the parameter selection strategy in the general case follows in Section 6.2. In Sections 6.3 and 6.4, we apply the strategy to two more examples (a rod with a higher aspect ratio and a plane wall).

In order to estimate the error during the parameter selection process, we apply a constant density $\tilde{\boldsymbol{q}}$ such that $|\tilde{\boldsymbol{q}}| = 1$ to the surface and evaluate the stresslet identity (5).[**] This may seem like an overly simple test case since both the density and the solution are constant. However, the QBX expansions are not of the constant double layer potential $\boldsymbol{D}$ itself, but of the four dipole potentials $\mathcal{L}$ defined by (34), and these are *not* constant. In practice, the stresslet identity seems to provide a decent test case for both direct quadrature, upsampled quadrature and QBX, as shown by the results in Section 7.2 where the density is not constant.

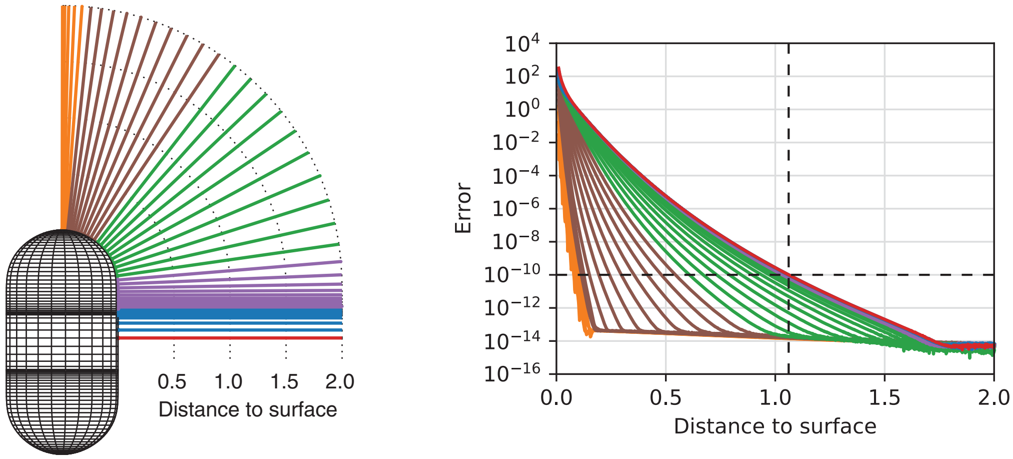
The SE parameters ξ and r_c will not be discussed at length here, but we note that the requirement that no special quadrature be needed for $|\boldsymbol{r}| > r_c$ implies that r_c must be at least as large as d_{U1}. The SE error is determined by the product ξr_c in real space and ξh_F in Fourier space, where h_F is the grid spacing of the uniform grid used for the Fourier-space part (see Appendix D). Given a tolerance ε_{tol}, the parameters ξ, r_c and h_F must satisfy the system $\xi r_c = A(\varepsilon_{tol})$, $\xi h_F = B(\varepsilon_{tol})$, where A and B are known functions. This leaves one degree of freedom which can be used to minimize the computational cost, albeit under the constraint $r_c \geq d_{U1}$. For a general discussion on the selection of SE parameters, including the functions A and B, we refer to af Klinteberg and Tornberg[58] for the stresslet, Lindbo and Tornberg[59] for the stokeslet, and af Klinteberg[60] for the rotlet.

6.1 | Introductory example: A rod particle with low aspect ratio

In this first example, we consider a rod particle of length $L = 2$ and radius $R = 0.5$ (i.e., aspect ratio 2), shown in Figure 13. The grid used for the direct quadrature has parameters $n_1 = 40$, $n_2 = 10$ and $n_\varphi = 25$ (introduced in Section 3.1), for a total of 2250 grid points. To give an idea of the error associated with the direct quadrature, we apply the constant density $\tilde{\boldsymbol{q}} = (1, 1, 1)/\sqrt{3}$ to the particle surface and compute the stresslet identity (5) using direct quadrature in two planes intersecting the particle. The absolute error in these planes is shown in Figure 13.

To determine how the error varies with the distance to the surface, we evaluate (5) along several normal lines centered on grid points of the particle; due to the symmetry of the error it is enough to consider the $n_1 + n_2/2 = 45$ lines shown in Figure 14(A). The error along these lines is shown in Figure 14(B). Given an error tolerance ε_{tol}, the smallest distance at which the error does not exceed ε_{tol} can be determined numerically. This distance is taken as d_{U1}. In this example, we will use the error tolerance $\varepsilon_{tol} = 10^{-10}$. As indicated in Figure 14(B), the error reaches 10^{-10} at $d_{U1} = 1.061$; special quadrature must be used within this distance to the surface. Having established the first threshold distance d_{U1}, we now proceed to determine the rest of the parameters for the upsampled quadrature regions, in Section 6.1.1.

[**] Since the computation of the layer potential is a linear function of $\boldsymbol{q}$, the error will scale with $Q = \max_{\boldsymbol{x}\in\Gamma}|\boldsymbol{q}(\boldsymbol{x})|$. In particular, if the maximum error is ε_{tol} when $Q = 1$, the maximum error will be $\alpha\varepsilon_{tol}$ when the density is multiplied by a constant α.



(A) Lines along which the error is plotted

(B) Direct quadrature error along the lines

FIGURE 14 Error of the direct quadrature along 45 normal lines centered on grid points of the particle surface. In (A), the normal lines are shown colored in groups of 10. In (B), the error along each line is shown with the same colors (here, the blue and purple curves are obscured by the red curve). The line with the greatest error is colored red; the error along this line reaches the value 10^{-10} at distance 1.061 from the surface (The smallest distance to surface included here is 0.01)

6.1.1 | Parameters for the upsampled quadrature regions

For the sake of simplicity we will always choose the upsampling factors to be $\kappa_{Ui} = i + 1$, meaning that the first upsampling factor will be $\kappa_{U1} = 2$, the next will be $\kappa_{U2} = 3$ and so forth (this may not be the optimal strategy with regard to computational cost, but recall that our goal is not to optimize for computational efficiency). In order to determine the threshold distance d_{Ui} of every upsampled quadrature region, we repeat the investigation from Figure 14 for different upsampling factors $\kappa = 1, 2, 3, \dots$, computing the stresslet identity error as a function of the distance to the surface for each upsampling factor. The maximal error at each distance is shown in Figure 15 ($\kappa = 1$ corresponds to Figure 14). The threshold distance $d_{U(i+1)}$ is now taken as the distance at which the error curve corresponding to $\kappa = \kappa_{Ui}$ intersects the error tolerance ε_{tol} ($i = 1, 2, \dots$). For instance, in this case the curve corresponding to $\kappa = \kappa_{U1} = 2$ intersects $\varepsilon_{\text{tol}} = 10^{-10}$ around $0.391 = d_{U2}$.

This procedure sets all of the parameters for the upsampled quadrature, as shown in Table 1. However, at some point we must switch from upsampled quadrature to QBX, which is determined by the QBX threshold distance d_{QBX}. Selecting

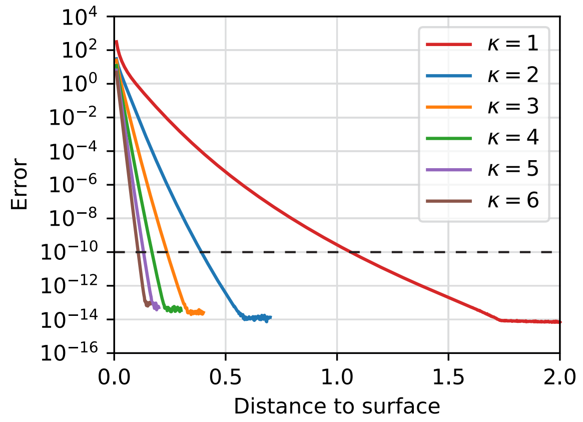
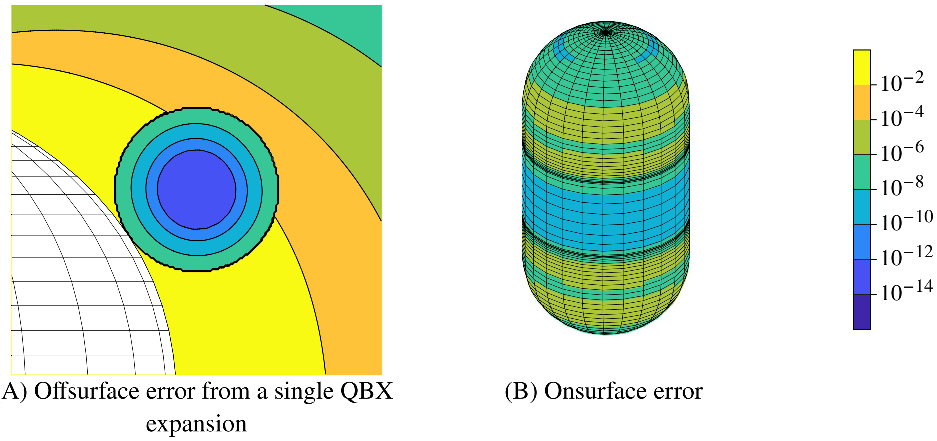


FIGURE 15 Maximal stresslet identity error along any of the lines shown in Figure 14(A), for upsampled quadrature with different upsampling factors κ (The smallest distance to surface included here is 0.01) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Parameters for the upsampled quadrature regions, tolerance 10^{-10}

i	1	2	3	4	5	6	…
κ_{Ui}	2	3	4	5	6	7	…
d_{Ui}	1.061	0.391	0.237	0.169	0.132	0.108	…



(A) Offsurface error from a single QBX expansion

(B) Onsurface error

FIGURE 16 Error when evaluating the stresslet identity (5) using QBX with the parameters $r_{\text{QBX}} = h = \pi/25$, $p_{\text{QBX}} = 18$ and $\kappa_{\text{QBX}} = 15$: (A) in a slice through the particle center (i.e., offsurface), using a single QBX expansion center and direct quadrature outside the ball of convergence; (B) in the grid points of the particle (i.e., onsurface). Note in (B) that the error seems to be related to the curvature of the boundary; in particular the error is larger in areas where the curvature changes, namely, in the smooth transition from cylinder to cap (Similar observations related to the convexity of the boundary have been previously reported[32,34]). QBX, quadrature by expansion [Colour figure can be viewed at wileyonlinelibrary.com]

d_{QBX} will also fix the number of upsampled quadrature regions N_{U}. We will determine d_{QBX} together with the other QBX parameters in Section 6.1.2.

6.1.2 | Parameters for the QBX region

To understand how the QBX error behaves, we plot the offsurface error from a single expansion in Figure 16(A). Note that the QBX parameters used in this figure are not yet selected to achieve the error tolerance, but meant only to demonstrate the general behavior of the error. Since the QBX error is the largest at the boundary of the ball of convergence (outside this ball the direct quadrature error is shown in Figure 16(A)), it is sufficient to measure the error at a point on this boundary, for example at the point where the ball touches the particle. Thus, we measure the QBX error at all the grid points of the rod—the onsurface error—shown in Figure 16(B) for these particular QBX parameters.

Recall that for the rod considered in this example, $R = 0.5$ and $n_{\varphi} = 25$; thus, by Equation (31), the grid spacing is $h = \pi/25 \approx 0.1257$. We now focus on selecting the parameters r_{QBX}/h, p_{QBX} and κ_{QBX} such that the error is bounded by ϵ_{tol} in the whole ball of convergence, for all QBX expansions of the particle. To do this, we consider the maximal onsurface error as we vary these three parameters, shown in Figure 17. As seen in Figure 17(A), r_{QBX}/h should be chosen as small as possible since this improves the decay of the truncation error as p_{QBX} grows; if r_{QBX}/h is small, p_{QBX} can also be chosen small, which is important since the offsurface evaluation time for QBX grows as $O(p_{\text{QBX}}^2)$. On the other hand, as Figure 18 shows, r_{QBX} must not be too small compared with h, since then the balls of convergence would not overlap properly, and large areas of the QBX region would not be covered by any ball of convergence.[††] For this reason we require that $r_{\text{QBX}} \geq h$. In fact, since r_{QBX} should be as small as possible, we will always set $r_{\text{QBX}} = h$, so that $r_{\text{QBX}}/h = 1$.

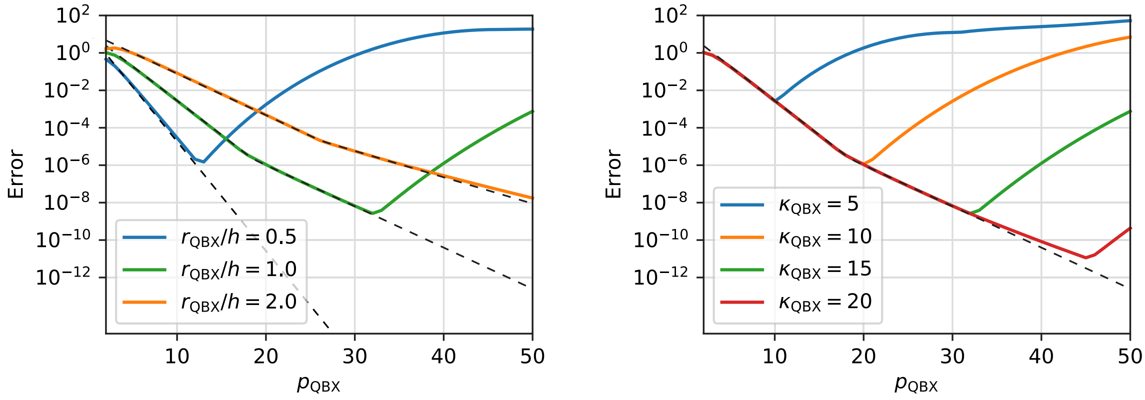
To select p_{QBX} and κ_{QBX}, we use the data shown in Figure 17(B), which is for $r_{\text{QBX}}/h = 1$. As can be seen there, the truncation error is independent of κ_{QBX} and depends only on p_{QBX}, so we simply select the smallest p_{QBX} such that the truncation error is below the tolerance.[‡‡] Then we select the smallest κ_{QBX} (restricted to multiples of five for convenience)

[††] Some areas of the QBX region will inevitably fall outside every ball of convergence no matter how large r_{QBX} is. However, these areas are mainly very close to the surface but not at the grid points, where it is typically not necessary to evaluate the layer potential.

[‡‡] The dashed curves in Figure 17 indicate the experimental truncation error estimate

$$e_{\text{trunc}} \approx \max\left(13(0.245\log(\rho) + 0.43)^{p_{\text{QBX}}}, 0.07(\rho - 0.63)(0.175\log(\rho) + 0.602)^{p_{\text{QBX}}}\right), \tag{56}$$

where $\rho = r_{\text{QBX}}/h$. This estimate was constructed for the rod particle in this particular example by applying curve fitting to data from a parameter study similar to that shown in Figure 17 itself. Unfortunately, this experimental estimate is of limited use in the parameter selection process since it would have to be reconstructed for every new geometrical object (such as a rod with a different aspect ratio), while the data used to construct it can just as well be used directly to select p_{QBX}.

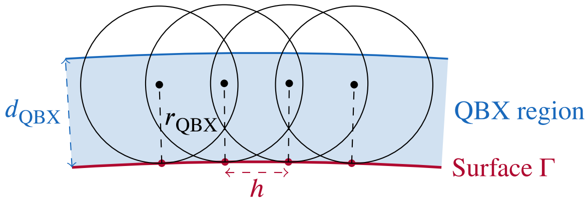


(A) Onsurface QBX error for different r_{QBX}/h

(B) Onsurface QBX error for different κ_{QBX}

FIGURE 17 Maximal onsurface error on the rod when evaluating the stresslet identity using QBX. In (A) for different r_{QBX}/h with $\kappa_{QBX} = 15$ fixed, and in (B) for different κ_{QBX} with $r_{QBX}/h = 1$ fixed. Note that each curve has a minimum, which is where the truncation error (which decreases as p_{QBX} grows) and the coefficient error (which increases as p_{QBX} grows) balance. The dashed lines indicate the experimental truncation error estimate (56). QBX, quadrature by expansion [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 18 Balls of convergence with the parameters r_{QBX} and d_{QBX} and the grid spacing h marked. QBX, quadrature by expansion [Colour figure can be viewed at wileyonlinelibrary.com]



such that the coefficient error is no larger than the truncation error (i.e., such that the minimum point of the error curve is to the right of the selected p_{QBX}). For example, for $\epsilon_{tol} = 10^{-10}$, we must choose $p_{QBX} = 40$ and $\kappa_{QBX} = 20$.

It remains to choose the threshold distance d_{QBX}, which determines the extent of the QBX region as shown in Figure 18. Clearly d_{QBX} cannot be larger than $2r_{QBX}$ since then the balls of convergence would not reach the edge of the QBX region. Even with $d_{QBX} = 2r_{QBX}$, there would be areas in the QBX region, close to its edge, that would not be inside any ball of convergence. To mitigate this problem, we introduce a safety factor γ, derived in Appendix C, and require that

$$d_{QBX} \leq 2\gamma r_{QBX}, \tag{57}$$

where $\gamma = 0.85$. As long as d_{QBX} satisfies (57), it can be chosen arbitrarily, in the sense that its value will not affect the conformance to the error tolerance, only the computational cost. We introduce the somewhat arbitrary additional constraint that $d_{QBX} \geq r_{QBX}$, and then select d_{QBX} as follows: If the interval $[r_{QBX}, 2\gamma r_{QBX}]$ contains any threshold distance d_{Ui} for the upsampled quadrature regions, set d_{QBX} equal to the largest d_{Ui} in the interval (i.e., the one with the smallest i). Otherwise, set $d_{QBX} = r_{QBX}$. In any case, this also sets the number of upsampled quadrature regions N_U since the last upsampled quadrature region ends where the QBX region begins. Our choices here are motivated by keeping N_U as low as possible since this reduces the computational cost, which we will return to in Section 6.1.4.

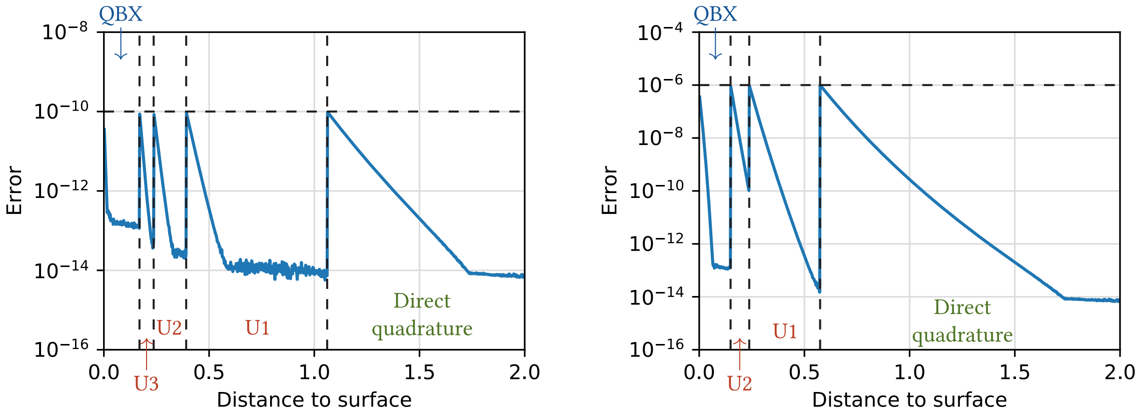
In our current example, d_{QBX} should be in the interval $[r_{QBX}, 2\gamma r_{QBX}] \approx [0.1257, 0.2136]$. As seen in Table 1, $d_{U4} = 0.169$ is the largest threshold distance in this interval, and thus we select $d_{QBX} = 0.169$ which means that $N_U = 3$ upsampled quadrature regions are used.

6.1.3 | Verification of selected parameters

To summarize, the parameters that were selected above for the rod in this example was, with error tolerance 10^{-10},

$$\begin{aligned}
r_{\mathrm{QBX}}/h &= 1, & N_{\mathrm{U}} &= 3, & & \\
d_{\mathrm{QBX}} &= 0.169, & d_{\mathrm{U1}} &= 1.061, & \kappa_{\mathrm{U1}} &= 2, \\
p_{\mathrm{QBX}} &= 40, & d_{\mathrm{U2}} &= 0.391, & \kappa_{\mathrm{U2}} &= 3, \\
\kappa_{\mathrm{QBX}} &= 20, & d_{\mathrm{U3}} &= 0.237, & \kappa_{\mathrm{U3}} &= 4.
\end{aligned} \tag{58}$$

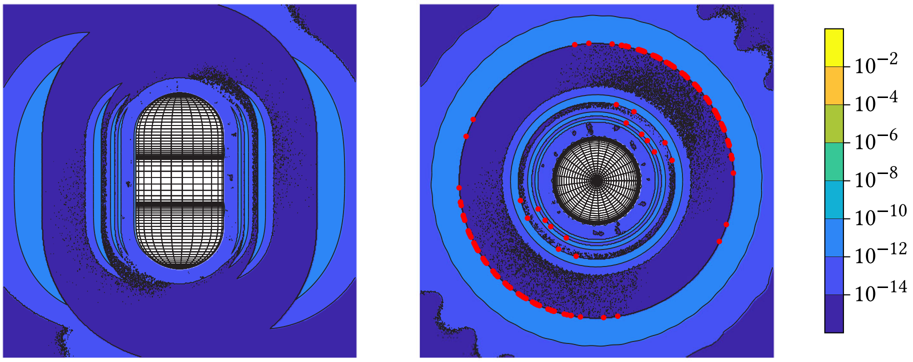
If the selected QBX upsampling factor κ_{QBX} seems large, recall that this parameter is completely hidden in the QBX precomputation step, as explained in Section 4.3. To verify that the selected parameters keep the error below the tolerance, we plot in Figure 19(A) the maximum error along the 45 lines that were used earlier (shown in Figure 14(A)). We also plot the error in two slices in Figure 20. The fact that the error slightly exceeds the tolerance at some points in (B) should come as no surprise, since we have used the error only along certain lines to select the parameters, not in the whole space. All the points where the tolerance is exceeded are close to the boundary between different quadrature regions and could thus be eliminated by adjusting the threshold distances slightly upwards (which we will however not do here).



(A) Special quadrature error, tolerance 10^{-10}

(B) Special quadrature error, tolerance 10^{-6}

FIGURE 19 Maximal stresslet identity error along any of the lines shown in Figure 14(A) as a function of the distance to the surface (for 1000 equispaced distances in $[0, 2]$), using the combined special quadrature with (A) tolerance 10^{-10} and (B) tolerance 10^{-6}. The different quadrature regions are marked. The largest error is 9.709×10^{-11} in (A), and 9.995×10^{-7} in (B) [Colour figure can be viewed at wileyonlinelibrary.com]



(A) Special quadrature error, tolerance 10^{-10}, slice 1

(B) Special quadrature error, tolerance 10^{-10}, slice 2

FIGURE 20 Error in two perpendicular slices (A) and (B) through the rod particle, when evaluating the stresslet identity (5) using combined special quadrature with tolerance 10^{-10} in free space. Each slice consists of 500×500 evaluation points. All points in (A) are below the tolerance, but 166 points in (B), marked red, are above the tolerance. The largest error is 9.768×10^{-11} in (A) and 1.074×10^{-10} in (B) [Colour figure can be viewed at wileyonlinelibrary.com]

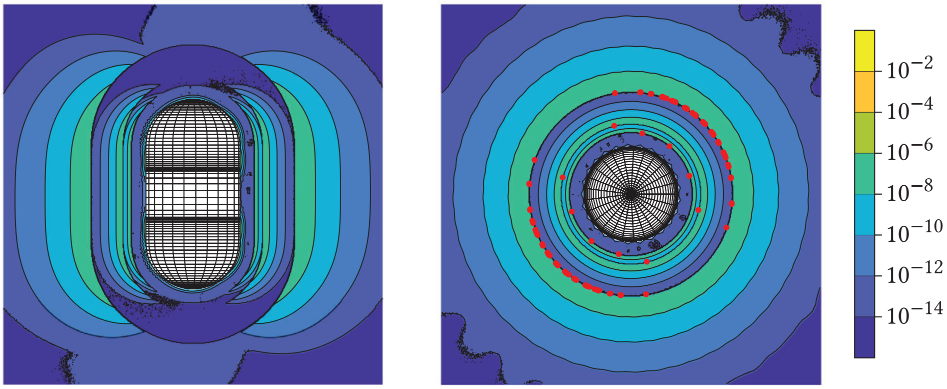
The parameter selection procedure is repeated for the same rod with the looser error tolerance 10^{-6}. The parameters for tolerance 10^{-6} are

$$\begin{aligned} &r_{\text{QBX}}/h = 1, && N_{\text{U}} = 2, && \\ &d_{\text{QBX}} = 0.149, && d_{\text{U1}} = 0.575, && \kappa_{\text{U1}} = 2, \\ &p_{\text{QBX}} = 21, && d_{\text{U2}} = 0.238, && \kappa_{\text{U2}} = 3, \\ &\kappa_{\text{QBX}} = 15, && && \end{aligned} \tag{59}$$

and the error is shown in Figures 19(B) and 21.

6.1.4 | A note on the computational cost

While our parameter selection strategy does not try to optimize the computational cost, we naturally strive for a reasonably low cost. We therefore comment on the computational cost for the different quadrature methods considered here. The



(A) Special quadrature error, tolerance 10^{-6}, slice 1

(B) Special quadrature error, tolerance 10^{-6}, slice 2

FIGURE 21 Error in two perpendicular slices (A) and (B) through the rod particle, when evaluating the stresslet identity (5) using combined special quadrature with tolerance 10^{-6} in free space. Each slice consists of 500×500 evaluation points. All points in (A) are below the tolerance, but 90 points in (B), marked red, are above the tolerance. The largest error is 9.214×10^{-7} in (A) and 1.079×10^{-6} in (B) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Computational complexities for the different quadrature methods

Direct quadrature	
Evaluate	$T_{\text{DE}} = O(N_{\text{grid}} N_{\text{eval,D}})$
Upsampled quadrature	
Interpolate density	$T_{\text{UI},i} = O(\kappa_{\text{U}i}^2 N_{\text{grid}}^2)$
Evaluate	$T_{\text{UE},i} = O(\kappa_{\text{U}i}^2 N_{\text{grid}} N_{\text{eval,U}i})$
QBX	
Compute coefficients	$T_{\text{QC}} = O(p_{\text{QBX}}^2 N_{\text{grid}} N_{\text{exp}})$
Evaluate expansion	$T_{\text{QE}} = O(p_{\text{QBX}}^2 N_{\text{eval,QBX}})$

Note: Time complexities for evaluating the double layer potential $\boldsymbol{\mathcal{D}}$, excluding precomputation time. Here, N_{grid} is the total number of grid points on the part of the surface included in the special quadrature method (i.e., $\tilde{\Gamma}$ as defined in Section 4.2); $N_{\text{eval,D}}$, $N_{\text{eval,U}i}$, and $N_{\text{eval,QBX}}$ are the number of evaluation points in the direct quadrature region, the ith upsampled quadrature region and the QBX region, respectively; and N_{exp} is the number of expansion centers that are to be used for the evaluation points in the QBX region.

computational complexity for evaluating the layer potential using each quadrature method is shown in Table 2. The precomputation time (for constructing the interpolation matrices and QBX matrices), which is naturally independent of the number of evaluation points, is not included (an example of how the precomputation time compares to the rest of the algorithm is found in Section 7.3.1). Note that the total evaluation time depends on the number of evaluation points in each quadrature region, and also on the number of expansions that are used for the evaluation points in the QBX region (recall that the closest expansion center is used for each evaluation point).

An example of evaluation times for a specific computer machine is given in Table 3, again excluding precomputation. The time required to find the closest expansion center for each evaluation point in the QBX region has been omitted from Tables 2 and 3 since it is negligible (around $10^{-8} \times N_{\mathrm{eval,QBX}}$ seconds).

For a particle, N_{grid} is the number of grid points on the whole particle, that is, $N_{\mathrm{grid}} = 2250$ for the rod that we have considered so far. Let us study the special case of a single evaluation point, relevant for example when computing a streamline. Based on Table 3, the evaluation time for this single point can be computed, depending on which quadrature region the point belongs to and the parameter $\kappa_{\mathrm{U}i}$ or p_{QBX}. This is shown in Table 4. From this, it can for example be seen that the evaluation takes roughly 1000 times longer for a point in the upsampled quadrature region with $\kappa_{\mathrm{U}i} = 2$ compared with the direct quadrature region. (The upsampled quadrature cost is in this case completely dominated by interpolating the density, that is, multiplying it by the precomputed interpolation matrix.)

TABLE 3 Example of actual evaluation times (seconds)

Direct quadrature	
Evaluate	$T_{\mathrm{DE}} = 5.6 \times 10^{-9} \times N_{\mathrm{grid}} N_{\mathrm{eval,D}}$
Upsampled quadrature	
Interpolate density	$T_{\mathrm{UI},i} = 8.0 \times 10^{-10} \times \kappa_{\mathrm{U}i}^2 N_{\mathrm{grid}}^2$
Evaluate	$T_{\mathrm{UE},i} = 5.7 \times 10^{-9} \times \kappa_{\mathrm{U}i}^2 N_{\mathrm{grid}} N_{\mathrm{eval,U}i}$
QBX	
Compute coefficients	$T_{\mathrm{QC}} = 5.3 \times 10^{-8} \times (0.071 p_{\mathrm{QBX}}^2 + 0.56 p_{\mathrm{QBX}} + 1) N_{\mathrm{grid}} N_{\mathrm{exp}}$
Evaluate expansion	$T_{\mathrm{QE}} = 2.2 \times 10^{-5} \times (0.0053 p_{\mathrm{QBX}}^2 - 0.0027 p_{\mathrm{QBX}} + 1) N_{\mathrm{eval,QBX}}$

Note: These times are for a modern workstation with a 6-core Intel Core i7-8700 CPU (4.6 GHz).
Abbreviation: QBX, quadrature by expansion.

TABLE 4 Evaluation times for $N_{\mathrm{grid}} = 2250$ and a single evaluation point (seconds)

Direct quadrature	
Time (s)	
1.3×10^{-5}	
Upsampled quadrature	
$\kappa_{\mathrm{U}i}$	**Time (s)**
2	1.6×10^{-2}
3	3.7×10^{-2}
4	6.5×10^{-2}
5	1.0×10^{-1}
6	1.5×10^{-1}
QBX (with $N_{\mathrm{exp}} = 1$)	
p_{QBX}	**Time (s)**
10	1.7×10^{-3}
20	4.9×10^{-3}
30	9.9×10^{-3}
40	1.7×10^{-2}
50	2.5×10^{-2}

It can also be seen in Table 4 that QBX is often faster than upsampled quadrature. For instance, QBX with $p_{\text{QBX}} = 40$ takes about as much time as upsampled quadrature with $\kappa_{\text{U}i} = 2$ and is faster than any $\kappa_{\text{U}i} \geq 3$. However, note that this conclusion may not hold when there are more than one evaluation point, since the evaluation time depends in an intricate way on both the number of evaluation points in each region and the number of expansions needed for QBX. In particular, if many expansions are needed (large N_{exp}) and there are few evaluation points per expansion, QBX will tend to be slower than upsampled quadrature due to the large cost of computing coefficients.

6.2 | Summary of the parameter selection strategy

The parameter selection strategy can, in the general case, be summarized as follows. In all steps, the stresslet identity (5) is used to estimate the error.

Input: Discretization of the geometry, error tolerance ε_{tol}.

Output: Parameters N_{U}, $(d_{\text{U}i}, \kappa_{\text{U}i})_{i=1}^{N_{\text{U}}}$, d_{QBX}, r_{QBX}, p_{QBX}, κ_{QBX} for the special quadrature.

For each distinct geometrical object:

1. Put $\kappa_{\text{U}i} = i + 1$. Numerically determine the threshold distances $d_{\text{U}i}$ to keep the error below ε_{tol}, as in Figure 15 and Table 1, up to the first i such that $d_{\text{U}i} \leq 2\gamma h$, where $\gamma = 0.85$ and h is the grid spacing (defined for particles in Equation (31) and for walls in Equation (33)).
2. Put $r_{\text{QBX}} = h$. If the last (smallest) $d_{\text{U}i}$ computed in step 1 lies in the interval $[h, 2\gamma h]$, put d_{QBX} equal to it. Otherwise, put $d_{\text{QBX}} = h$. This also sets N_{U}, the number of upsampled quadrature regions.
3. Choose p_{QBX} such that the truncation error is below ε_{tol}, based on a parameter study such as in Figure 17(B). Choose κ_{QBX} such that the coefficient error is no larger than the truncation error.

This strategy is designed to keep the error relative to $\max_{\Gamma}|\boldsymbol{q}|$ below ε_{tol} when evaluating the layer potential, provided that the density $\boldsymbol{q}$ is well-resolved by the discretization. While there is no guarantee that the error stays strictly below ε_{tol}, empirical evidence in Sections 6.1.3, 6.3, 6.4, 7.1, and 7.2 indicates that the error is typically close to the tolerance, and in any case of the correct order of magnitude. Note that the procedure, including the parameter studies, must be repeated every time a new geometrical object, such as a rod particle with a different aspect ratio, is used. We will now apply the procedure to two additional examples.

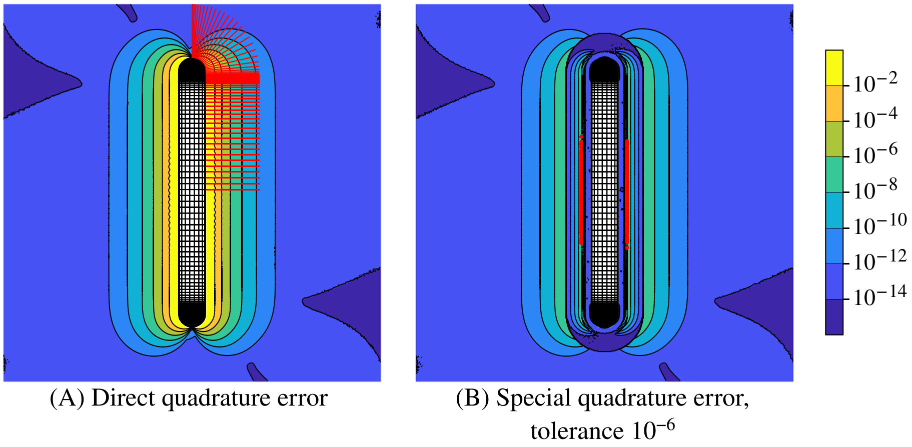
6.3 | Example II: A rod particle with higher aspect ratio

For the second example, we consider a more slender rod particle, namely, the rod of length $L = 10$ and radius $R = 0.5$ (aspect ratio 10) shown in Figure 22. The grid has parameters $n_1 = 35$, $n_2 = 60$, and $n_\varphi = 18$, in total 2340 grid points. The grid spacing as defined by (31) is $h = \pi/18 \approx 0.1745$. It should be noted that while h is based only on the grid resolution in the azimuthal direction, the resolution in the polar direction must not be much coarser. Otherwise, the distance between QBX expansion centers would be too large in the polar direction, and the balls of convergence would not cover the QBX region. (This limitation is due to having one expansion center per grid point.)

Applying the same constant density $\tilde{\boldsymbol{q}} = (1, 1, 1)/\sqrt{3}$, the direct quadrature error is shown in Figure 22(A). The special quadrature parameters are selected as described in Section 6.2, with the error along the 65 lines shown in red in Figure 22(A) used to select the threshold distances. The parameters for error tolerance 10^{-6} are

$$\begin{aligned}
&r_{\text{QBX}}/h = 1, && N_{\text{U}} = 2, && \\
&d_{\text{QBX}} = 0.234, && d_{\text{U}1} = 0.930, && \kappa_{\text{U}1} = 2, \\
&p_{\text{QBX}} = 28, && d_{\text{U}2} = 0.355, && \kappa_{\text{U}2} = 3. \\
&\kappa_{\text{QBX}} = 20, && &&
\end{aligned} \tag{60}$$

The error when using these parameters is shown in Figure 22(B). As before, the tolerance is not strictly enforced, but the error stays within a factor 2 of the tolerance.


(A) Direct quadrature error

(B) Special quadrature error, tolerance 10^{-6}

FIGURE 22 Error in a slice through the rod particle, when evaluating the stresslet identity (5) in free space using (A) direct quadrature, and (B) combined special quadrature with tolerance 10^{-6}. In (B), the tolerance is exceeded in 148 points, marked red; the evaluation grid consists of 500×500 points and the largest error is 1.455×10^{-6}. In (A), the 65 lines used to select parameters are shown in red [Colour figure can be viewed at wileyonlinelibrary.com]

The parameters (60) for the slender rod can be compared with the parameters (59) for the less slender rod with the same tolerance; the threshold distances are relative to the diameter of the cylindrical part of the rod in both cases. Note that the slender rod (60) has larger threshold distances than the other rod (59). This reflects the fact that the error of the underlying direct quadrature at a fixed distance from the rod is higher for the slender rod, since it has lower overall resolution (grid points per surface area). The slender rod also requires a higher p_{QBX} since $r_{\text{QBX}} = h$ is larger for the slender rod.

6.4 | Example III: A pair of plane walls

In this third example, we select parameters for a plane wall. Since we always consider walls in a periodic setting, we will do so here as well, and use the SE method described in Section 5. We will here select the SE parameters such that the error from SE is completely negligible compared with the quadrature errors which we strive to control here.[§§] Since the problem is periodic in all three spatial directions, we must have a pair of walls so that the fluid domain can be confined between them. The periodic cell is here of size $\boldsymbol{B} = (1, 1, 1)$ and the two walls are placed at a distance of 0.6 from each other. The walls are discretized using 11×11 patches each, with 8×8 grid points on each patch (as described in Section 3.2), in total 7744 grid points per wall. The grid spacing is $h \approx 0.01668$, as defined by (33). The constant density $\tilde{\boldsymbol{q}} = (0, 0, 1)$ is applied in the direction of the normal of the lower wall (pointing into the fluid domain). The direct quadrature error is shown in Figure 23(A).

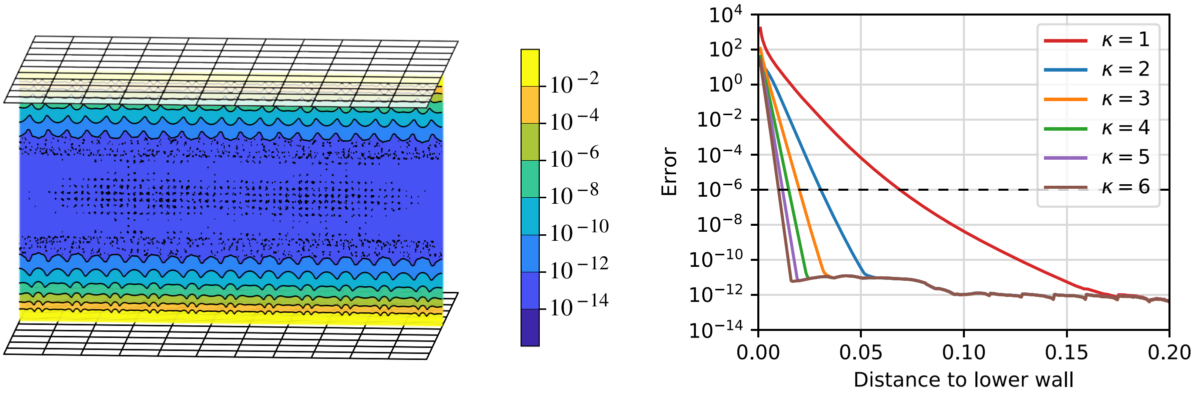
We follow the procedure in Section 6.2. The threshold distances of the upsampled quadrature regions are computed by evaluating the stresslet identity error along normal lines of the walls, with each line centered at a grid point. The error is plotted in Figure 23(B), and the resulting threshold distances for error tolerance 10^{-6} are $d_{\text{U1}} = 0.0687$, $d_{\text{U2}} = 0.0304$, and $d_{\text{QBX}} = 0.0198$.

To determine p_{QBX} and κ_{QBX}, we do a parameter study, shown in Figure 24(A). Note that the plane wall needs a significantly lower p_{QBX} than the rod particles to reach a given error. The error curves in Figure 24(A) level out at around 10^{-12} due to other errors not controlled by the QBX parameters. In order to reach the tolerance 10^{-6} it is sufficient to choose $p_{\text{QBX}} = 7$ and $\kappa_{\text{QBX}} = 10$. The selected parameters are thus

$$\begin{aligned} &r_{\text{QBX}}/h = 1, && N_{\text{U}} = 2, && \\ &d_{\text{QBX}} = 0.0198, && d_{\text{U1}} = 0.0687, && \kappa_{\text{U1}} = 2, \\ &p_{\text{QBX}} = 7, && d_{\text{U2}} = 0.0304, && \kappa_{\text{U2}} = 3. \\ &\kappa_{\text{QBX}} = 10, && && \end{aligned} \tag{61}$$

The error when using these parameters is shown in Figure 24(B).

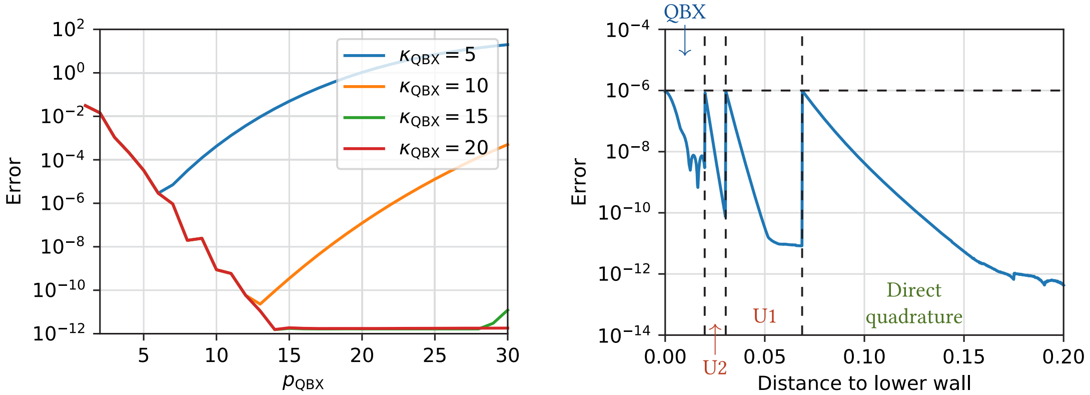
[§§] Specifically, the SE parameters used here are $\xi = 15.245$, $r_c = 0.4$, $P = 24$, and the uniform grid used for the Fourier-space part has $64 \times 64 \times 64$ grid points (see Section 5 and Appendix D for an explanation). These parameters should keep the SE error around 10^{-15} according to Reference 58.



(A) Direct quadrature error

(B) Upsampled quadrature error

FIGURE 23 Errors when evaluating the stresslet identity (5) in the periodic setting. (A) Error in the center plane for direct quadrature. (B) Largest error for upsampled quadrature with different upsampling factors κ, as a function of the distance to the lower wall [Colour figure can be viewed at wileyonlinelibrary.com]



(A) Onsurface QBX error for different κ_{QBX}

(B) Special quadrature error, tolerance 10^{-6}

FIGURE 24 (A) Maximal onsurface error when evaluating the stresslet identity using QBX, for different κ_{QBX} with $r_{\mathrm{QBX}}/h = 1$ fixed. (B) Maximal stresslet identity error as a function of the distance to the lower wall (for 1000 equispaced distances in $[0, 0.2]$), using the combined special quadrature with tolerance 10^{-6}. The largest error is 9.812×10^{-7}. QBX, quadrature by expansion [Colour figure can be viewed at wileyonlinelibrary.com]

7 | NUMERICAL RESULTS

Our numerical method can be summarized as follows:

1. The geometry is discretized as in Section 3. Parameters for the combined special quadrature method are selected as in Section 6.2. Parameter selection is done in free space for particles, but the same parameters can also be used in the periodic setting. For walls, parameter selection is done in the periodic setting.
2. The matrices $\boldsymbol{M}_i$ and $\boldsymbol{R}_i$ for offsurface QBX and onsurface QBX, respectively, are precomputed as in Section 4.3. Interpolation matrices for the upsampled quadrature regions are also precomputed. At this point, the special quadrature is ready to be used to evaluate the layer potential.
3. The boundary integral equation for either a resistance problem or a mobility problem is solved iteratively for $\boldsymbol{q}$ using GMRES. The SE method is used for periodic problems, as described in Section 5. A preconditioner is used in all cases, as described below.
   - For a resistance problem, velocities are given for all particles and the boundary integral equation is given by Equation (24).
   - For a mobility problem, forces and torques are applied to all particles and the boundary integral equation is given by Equation (25).

4. The flow field in the fluid domain may be computed in a postprocessing step using $\boldsymbol{q}$ from step 3. For a resistance problem, the forces and torques acting on all particles may also be computed here, while for a mobility problem, the particle velocities may be computed.

To improve the convergence of GMRES in step 3, we use a block-diagonal preconditioner similar to the one used by af Klinteberg and Tornberg.[37] The preconditioner is constructed by computing the explicit inverse of a single-particle system as well as a system consisting of a single wall patch (if walls are present in the simulation). These two types of blocks are then placed along the diagonal and rotated according to the geometry. This preconditioner has been seen to reduce the number of GMRES iterations by as much as a factor 17 for some systems with many particles, such as the ones in Section 7.3.

In this section, we test some aspects of our numerical method, with focus on the special quadrature. First, in Section 7.1, we test the quadrature on its own (i.e., steps 1–2 above) with geometries containing both particles and walls. This serves as a continuation of the tests in Section 6, where geometrical objects were considered only separately. In Section 7.2, we test the special quadrature in the context of the full numerical method (steps 1–4 above) and in particular how the quadrature tolerance influences the accuracy. Finally, in Section 7.3, we test the computational complexity of the method on a more complicated problem, and compute streamlines.

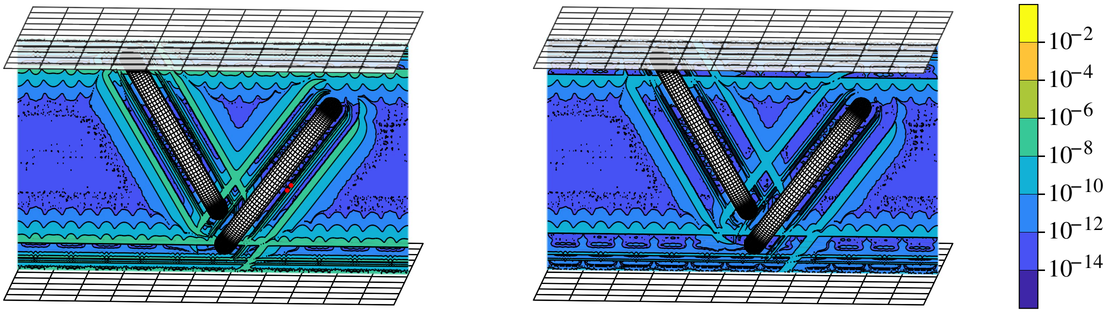
7.1 | Special quadrature with composite geometries

We consider two geometrical setups, shown in Figures 25 and 27. Both problems are periodic with a periodic cell of size $\boldsymbol{B} = (1, 1, 1)$, and the Spectral Ewald parameters are as in Section 6.4. As in Section 6, we use the stresslet identity (5) to estimate the error. This is the same test used to select the parameters, so it mainly serves as a consistency check (tests with nonconstant densities will follow in Section 7.2).

7.1.1 | Geometry 1: Two rods between a pair of plane walls

The first geometry consists of two plane walls discretized as in Section 6.4, at a distance 0.6 from each other. Between these walls are two rod particles of length $L = 0.5$ and radius $R = L/20$ (aspect ratio 10), discretized as in Section 6.3 (but scaled down a factor 20), oriented such that their axes lie in the center plane.

The stresslet identity error is shown for two different quadrature tolerances ε_{tol} in Figure 25. In (A), $\varepsilon_{\text{tol}} = 10^{-6}$, the quadrature parameters for the walls are as in Section 6.4, that is, given by (61); for the rods, the parameters are as in Section 6.3 but with all distances scaled by 1/20 to account for the difference in size. Thus, the parameters for $\varepsilon_{\text{tol}} = 10^{-6}$ for the rods are $d_{\text{QBX}} = 0.0117$, $p_{\text{QBX}} = 28$, $\kappa_{\text{QBX}} = 20$ and $N_{\text{U}} = 2$, $d_{\text{U1}} = 0.0465$, $d_{\text{U2}} = 0.0178$. In (B), for tolerance $\varepsilon_{\text{tol}} = 10^{-8}$, the parameters selected according to Section 6.2 are, for the walls $d_{\text{QBX}} = 0.0248$, $p_{\text{QBX}} = 10$, $\kappa_{\text{QBX}} = 10$ and $N_{\text{U}} = 2$, $d_{\text{U1}} = 0.0944$, $d_{\text{U2}} = 0.0386$; and for the rods $d_{\text{QBX}} = 0.0146$, $p_{\text{QBX}} = 40$, $\kappa_{\text{QBX}} = 25$ and $N_{\text{U}} = 2$, $d_{\text{U1}} = 0.0676$,



(A) Special quadrature error, tolerance 10^{-6} (B) Special quadrature error, tolerance 10^{-8}

FIGURE 25 Stresslet identity error in the center plane, for geometry 1, in (A) for tolerance 10^{-6} and in (B) for tolerance 10^{-8}. The largest error is 1.205×10^{-6} in (A) and 9.389×10^{-9} in (B). In (A), the error exceeds the tolerance in two points (the evaluation grid has 500×500 points), marked red [Colour figure can be viewed at wileyonlinelibrary.com]

$d_{U2} = 0.0237$. The maximal error in the center plane is plotted for varying quadrature tolerance in Figure 26. This shows that the error more or less follows the tolerance, as expected.

7.1.2 | Geometry 2: Two spheroids in a pipe

The second geometry consists of a pipe of radius 0.3, discretized using 5 × 10 patches with 6 × 6 grid points each. Inside the pipe are two spheroids with semiaxes $a = 0.05$ and $c = 0.1$, discretized with parameters $n_\theta = 36$ and $n_\varphi = 25$ (900 grid points per spheroid).

We select the error tolerance $\varepsilon_{\text{tol}} = 10^{-6}$. The parameters selected according to Section 6.2 are, for the pipe $d_{\text{QBX}} = 0.0614$, $p_{\text{QBX}} = 12$, $\kappa_{\text{QBX}} = 10$ and $N_U = 2$, $d_{U1} = 0.222$, $d_{U2} = 0.0888$; and for the spheroids $d_{\text{QBX}} = 0.0153$, $p_{\text{QBX}} = 27$, $\kappa_{\text{QBX}} = 15$ and $N_U = 2$, $d_{U1} = 0.0568$, $d_{U2} = 0.0235$. The error when using these parameters is shown in Figure 27(B), together with the direct quadrature error in Figure 27(A). Note that we have selected a much lower resolution for the pipe in comparison to the walls in geometry 1 (Section 7.1.1), which is reflected in the larger threshold distances compared with (61).

7.2 | Solving the boundary integral equation

Here, we investigate how the special quadrature tolerance influences the accuracy of the numerical method, that is, when solving the boundary integral equation. We will use the mobility problem as our model problem, and apply the force $\boldsymbol{F} = (0, 0, -1)$ to all particles, with zero torque and no background flow. In order to get the expected accuracy when solving the boundary integral equation, the double layer density must be well-resolved by the geometry discretization. It turns out that for elongated particles, the density and how easy it is to resolve depends heavily on the number of completion sources N_{src} (defined in Section 2.1). Therefore, we begin in Section 7.2.1 by investigating how large N_{src} must be to ensure

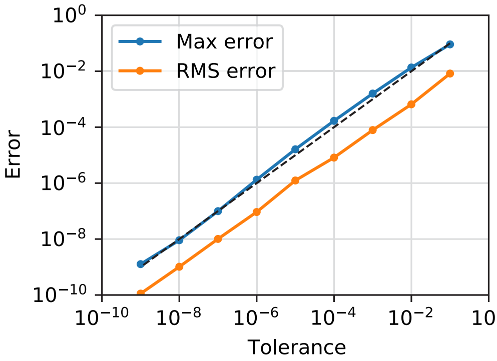
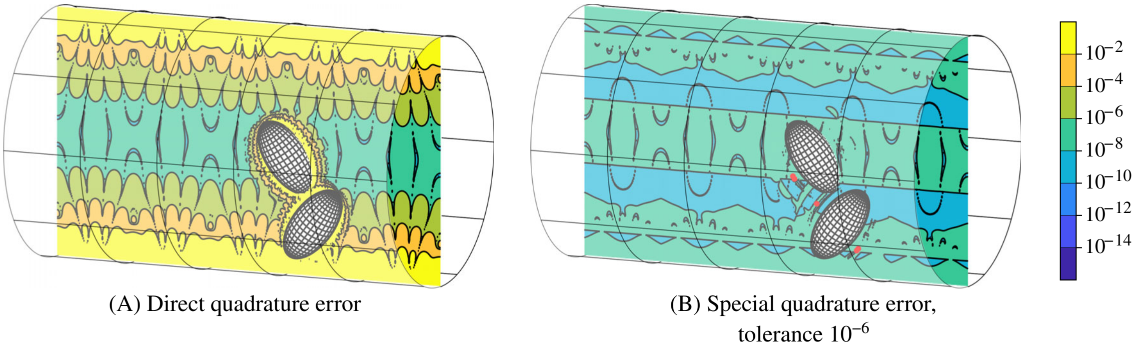


FIGURE 26 Maximal and root-mean-square (RMS) stresslet identity error in the center plane as a function of the special quadrature tolerance, for geometry 1. As observed already in Section 6, the tolerance is sometimes exceeded slightly at the threshold distances, which causes the max error curve to lie above the identity line Error = Tolerance (dashed) [Colour figure can be viewed at wileyonlinelibrary.com]



(A) Direct quadrature error

(B) Special quadrature error, tolerance 10^{-6}

FIGURE 27 Stresslet identity error in the center plane, for geometry 2, in (A) using direct quadrature and in (B) using combined special quadrature with tolerance 10^{-6}. In (B), the tolerance is exceeded in five points (the evaluation grid has 500 × 500 points), marked red; the largest error in the slice is 1.323×10^{-6} [Colour figure can be viewed at wileyonlinelibrary.com]

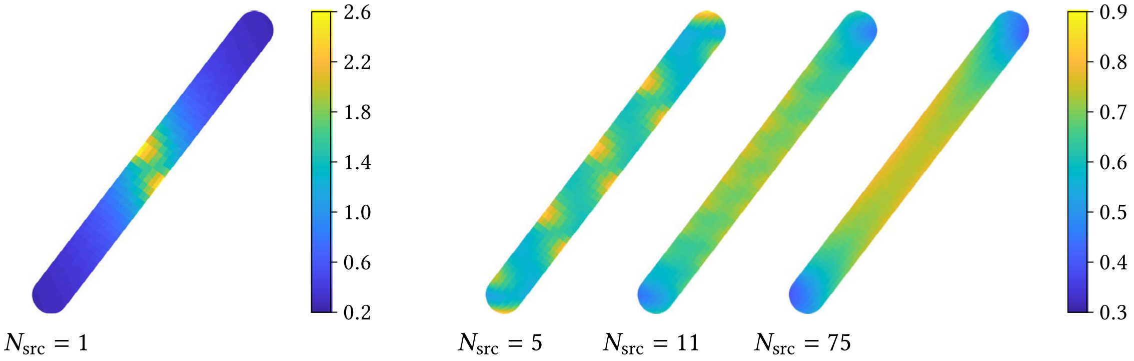


FIGURE 28 The magnitude of the completion flow $\mathcal{V}^{(\alpha=1)}[\boldsymbol{F}, \boldsymbol{0}](\boldsymbol{x})$ on the surface of the rod, for a few different values of N_{src}. Since the background flow is zero, this is exactly the right-hand side of the boundary integral Equation (20). Note that the color scale is different for $N_{\mathrm{src}} = 1$ compared with the other values [Colour figure can be viewed at wileyonlinelibrary.com]

that the density is well-resolved for a given discretization. Then, in Section 7.2.2, we study how the special quadrature tolerance influences the accuracy of the method.

7.2.1 | Selecting the number of completion sources

To study the influence of N_{src}, we consider a single rod particle with length $L = 0.5$ and radius $R = L/20$ (aspect ratio 10) in free space, shown in Figure 29(A) together with the flow field resulting from the force $\boldsymbol{F} = (0, 0, -1)$. We now solve this mobility problem for varying N_{src}, with the special quadrature error tolerance fixed to 10^{-9} here.[¶¶]

The completion flow $\mathcal{V}^{(\alpha)}$, which appears in the right-hand side of the boundary integral Equation (20), will change drastically as N_{src} grows from small values, as shown in Figure 28; the completion flow becomes increasingly smoother as N_{src} increases. Naturally, this means that the density $\boldsymbol{q}$ itself will change as N_{src} grows. However, the real physical quantities—the particle velocity and the flow field—should not change since the net force and torque on the particle does not change. Thus, these physical quantities can be used to gauge how N_{src} affects the accuracy of the solution. As Figure 29(B) shows, the effect is quite large, and most pronounced in the fluid flow velocity (the blue curve). (For this problem, the magnitude of the fluid flow velocity and particle velocity $\boldsymbol{U}_{\mathrm{RBM}}$ are both around 0.6, while the angular velocity $\boldsymbol{\Omega}_{\mathrm{RBM}}$ is zero.)

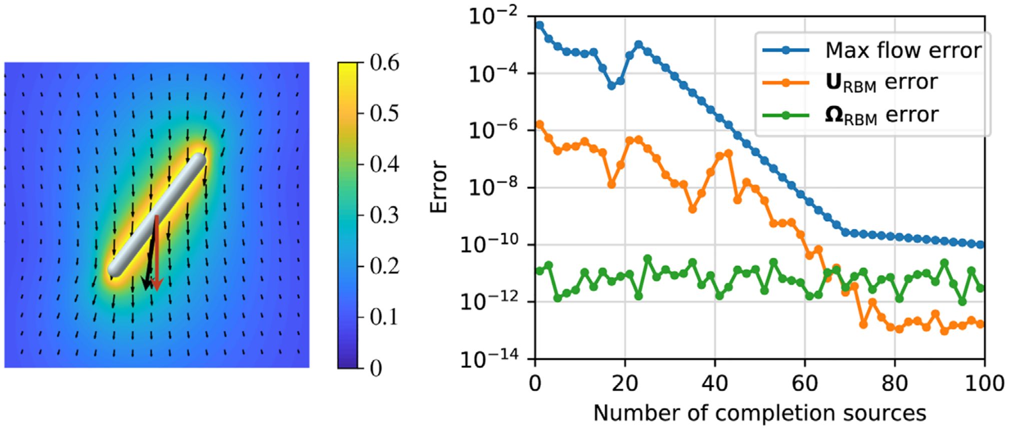
Thus, it is important to select N_{src} high enough for the error in Figure 29(B) to satisfy the error tolerance. The effect of N_{src} on the accuracy is stronger the more elongated the particle is; for particles with low aspect ratio, $N_{\mathrm{src}} = 1$ may be sufficient. The effect is very similar for the resistance problem, to the degree that the max flow error in Figure 29(B) can be used to determine N_{src} for both the mobility problem and resistance problem. Note that N_{src} does not affect the size of the linear system, that is, (24) or (25).

Our results here are consistent with work by Keaveny and Shelley,[27] where it was found that distributing the completion sources along the centerline of a spheroid leads to better accuracy and a more well-conditioned matrix for spheroids with high aspect ratios, compared with placing the completion source in a single point.

7.2.2 | Effect of the special quadrature on the accuracy

We continue to study the mobility problem, but now add another rod particle and a pair of plane walls, as shown in Figure 30. We fix $N_{\mathrm{src}} = 65$, which was enough to get the error below 10^{-9} in the previous problem. The walls are discretized using 22×22 patches with 8×8 grid points each (30,976 grid points per wall), and the rod particles are discretized as in Section 7.1. We set the special quadrature tolerance to different values between 10^{-1} and 10^{-8}, solve the mobility problem, and compute the flow field and particle velocities. The errors in the flow field, density and particle velocities are estimated

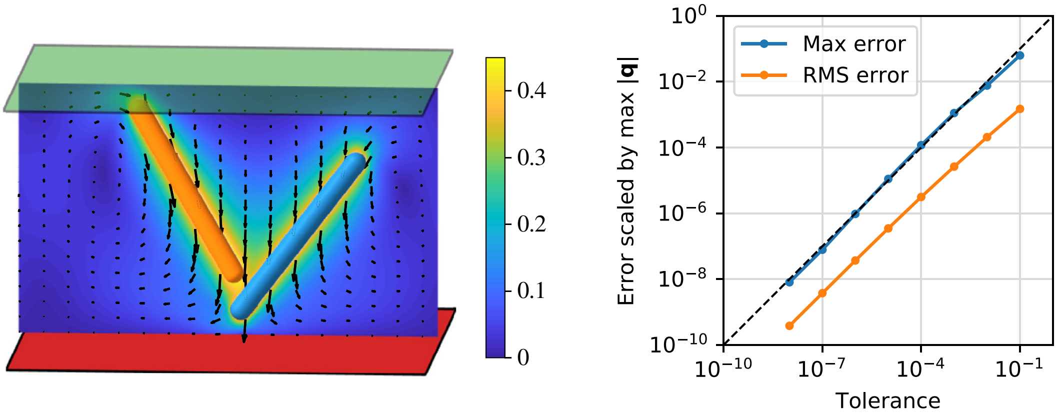
[¶¶] The rod particle is discretized as in Section 7.1, and the special quadrature parameters for $\epsilon_{\mathrm{tol}} = 10^{-9}$ are $r_{\mathrm{QBX}}/h = 1$, $d_{\mathrm{QBX}} = 0.0119$, $p_{\mathrm{QBX}} = 45$, $\kappa_{\mathrm{QBX}} = 25$, and $N_{\mathrm{U}} = 3$, $d_{\mathrm{U1}} = 0.0810$, $d_{\mathrm{U2}} = 0.0271$, $d_{\mathrm{U3}} = 0.0164$.



(A) Flow field

(B) Error contribution

FIGURE 29 (A) Flow field resulting from the mobility problem for a single rod in free space (color indicates velocity magnitude, small black arrows indicate velocity direction). The large red arrow indicates the applied force, and the large black arrow indicates the velocity of the rod (not to scale with the small arrows). (B) Contribution to the absolute error from the way the completion sources are distributed, as a function of N_{src} (for a rod particle of aspect ratio 10). The error is estimated as the difference to a reference solution with $N_{\mathrm{src}} = 135$. Note that the max flow error flattens out around 10^{-9}, the special quadrature error tolerance [Colour figure can be viewed at wileyonlinelibrary.com]



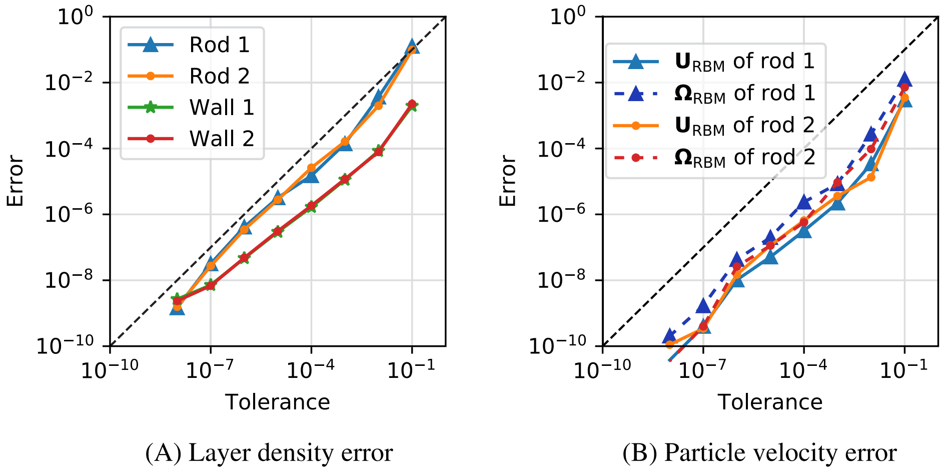
(A) Geometry and flow field

(B) Flow field error

FIGURE 30 (A) Flow field from the periodic mobility problem with force $\boldsymbol{F} = (0, 0, -1)$ applied to both particles (color indicates velocity magnitude, arrows indicate velocity direction). (B) Maximal and root-mean-square (RMS) flow field error in the center plane (estimated using a reference solution with tolerance 10^{-9}), scaled by the maximal density magnitude $\max\ |\boldsymbol{q}| \approx 5.4$. The dashed line indicates Scaled error = Tolerance [Colour figure can be viewed at wileyonlinelibrary.com]

using a reference solution with special quadrature tolerance 10^{-9}; these are shown in Figures 30(B) and 31(A,B). Note that the tolerance sets the flow field error relative to $\max\ |\boldsymbol{q}|$ quite accurately; the density and particle velocity errors are even smaller. We would like to point out that the value of the scale factor $\max\ |\boldsymbol{q}|$ is not known a priori, but it is of course known after having solved the boundary integral equation.

It should be noted that the error cannot be expected to follow the tolerance unless the density is well-resolved by the geometry discretization, since otherwise the interpolated density will be inaccurate. It has been observed that the density becomes hard to resolve, with either sharp peaks or high-frequency oscillations, when particles come very close to each other or the walls (where "very close" means that the distance from one surface to the other is of the same order of magnitude as the grid spacing, or smaller). Thus, one may be forced to increase the grid resolution in these cases. This may happen for example in suspensions with high volume fraction or in time-dependent simulations where particles may come arbitrarily close to each other.



(A) Layer density error

(B) Particle velocity error

FIGURE 31 (A) Maximal absolute error of the layer density on each geometrical object (colors as in Figure 30(A)), estimated using a reference solution with tolerance 10^{-9}. (B) Absolute error of the particle velocities [Colour figure can be viewed at wileyonlinelibrary.com]

For elongated particles, the set of matrices $\boldsymbol{M}_i$, $i = 1, \dots, n_\theta/2$, which are precomputed for offsurface QBX tends to become quite large for strict quadrature tolerances. The reason for this is that with particle-global QBX, the whole set of matrices consists of

$$3 n_\varphi n_\theta^2 (p_{\mathrm{QBX}} + 1)(p_{\mathrm{QBX}} + 2) \tag{62}$$

complex numbers, that is, it is quadratic in both p_{QBX} and n_θ (where n_θ is the number of grid points in the axial direction, which we define as $2n_1 + n_2$ for rod particles). For elongated particles, n_θ tends to be large; for example, for the rods considered in this section (aspect ratio 10), $n_\theta = 130$ and $n_\varphi = 18$. The set of matrices for tolerance 10^{-8} ($p_{\mathrm{QBX}} = 40$) then takes up around 25 gigabytes when stored in double precision, while for tolerance 10^{-6} ($p_{\mathrm{QBX}} = 28$) the matrices take up around 13 gigabytes. We emphasize that these storage sizes are determined by the discretization of the particle and p_{QBX} according to (62), but independent of the number of particles in the simulation (as long as all particles have the same shape). To reduce the size of the matrices for elongated particles, a local patch-based discretization could be used also for the particles, in the same way it is already used for the walls. This would reduce the number of grid points included in the special quadrature and thus the size of the matrices.

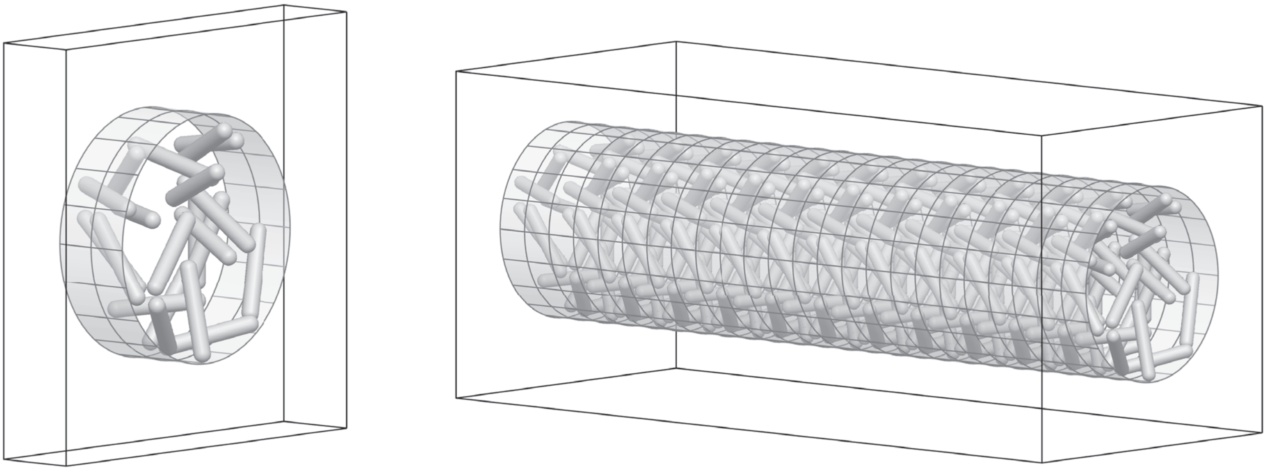
7.3 | Computational complexity and computation of streamlines

7.3.1 | Computational complexity of the method

The computational cost of our special quadrature method is quadratic in the number of grid points per particle (or patch), but linear in the number of particles (patches) if their discretization is kept fixed. For the SE method, the computational cost scales like $O(N \log N)$, where N is the number of unknowns in the system (i.e., three times the number of grid points in the primary cell), assuming that the number of grid points within a ball of radius r_c does not change. In other words, for fixed grid resolution and particle concentration, the time required per GMRES iteration when solving the boundary integral equation is expected to scale like $O(N \log N)$.

To test this scaling, we consider a problem with many rod particles confined in a pipe, shown in Figure 32: one segment (A) consists of a pipe segment of radius 0.3 and length 0.2 confined in a periodic cell of size $\boldsymbol{B} = (0.2, 1, 1)$, with 20 rods of length $L = 0.25$ and radius $R = L/12$ (aspect ratio 6) inside the pipe.[##] This segment is replicated to create a longer pipe, up to 12 times the original length (shown in (B)), with the same grid point concentration as the original segment. For $1, 2, 3, \dots, 12$ segments, we solve a resistance problem in which all particles are stationary and a quadratic background flow

[##] The discretization, special quadrature parameters and Spectral Ewald parameters are fixed as follows. Each pipe patch has 6×6 grid points, and each rod is discretized using $n_1 = 16$, $n_2 = 40$ and $n_\varphi = 18$ (1296 grid points per rod). The special quadrature parameters are selected for tolerance $\varepsilon_{\mathrm{tol}} = 10^{-4}$, and they are for the pipe $d_{\mathrm{QBX}} = 0.0319$, $p_{\mathrm{QBX}} = 8$, $\kappa_{\mathrm{QBX}} = 10$, and $N_{\mathrm{U}} = 1$, $d_{\mathrm{U1}} = 0.0720$; and for the rods $d_{\mathrm{QBX}} = 0.0110$, $p_{\mathrm{QBX}} = 19$, $\kappa_{\mathrm{QBX}} = 10$ and $N_{\mathrm{U}} = 1$, $d_{\mathrm{U1}} = 0.0254$. The SE parameters are $\xi = 52.954$, $r_c = 0.0897$, $P = 16$, and the uniform grid for the Fourier-space part has $32 n_s \times 160 \times 160$ grid points, where n_s is the number of segments. Furthermore, we use $N_{\mathrm{src}} = 45$ completion sources per rod.



(A) 1 segment

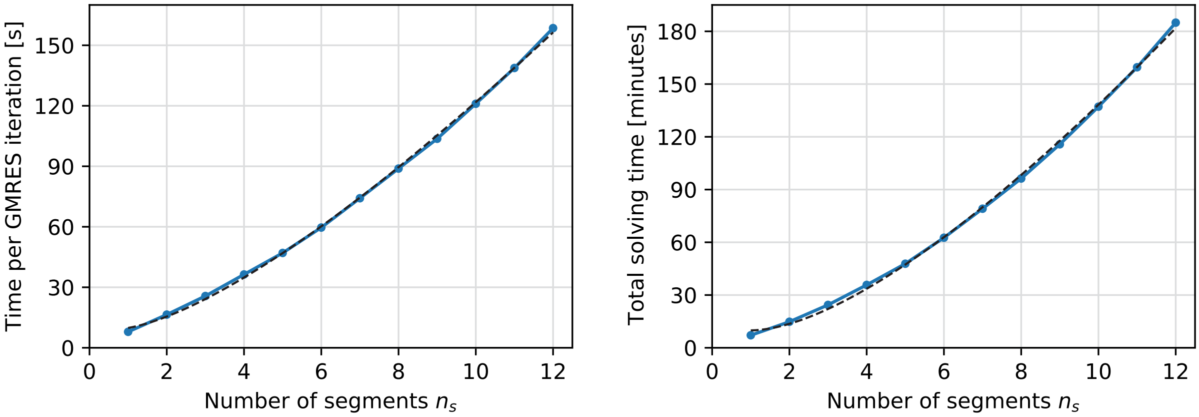
(B) 12 segments

FIGURE 32 The geometry in Section 7.3.1 is made up of stacked identical segments, where each segment contains 20 rods and 2×20 pipe patches. In total, there are 27,360 grid points and 82,080 unknowns in each segment

$$\boldsymbol{u}_{\mathrm{bg}}(\boldsymbol{x}) = \left(\frac{A^2 - x_2^2 - x_3^2}{A^2},\ 0,\ 0 \right) \tag{63}$$

is applied, where $A = 0.3$ is the radius of the pipe and $(x_2, x_3) = (0, 0)$ is its center line.

As seen in Figure 33(A), the time per GMRES iteration follows the expected scaling $O(N \log N)$. Since the structure of the linear system changes as the number of segments n_s grows, the number of GMRES iterations grows slightly with n_s. However, this growth is slow enough for the total solving time to also follow the scaling$O(N \log N)$, as shown in Figure 33(B). The runtimes in this figure are measured on a machine with a 6-core Intel Core i7-8700 CPU running at 4.6 GHz. The time required for precomputation (cf. Section 4.3) is not included in this figure; precomputation is independent of the number of segments and takes around 8 min for this problem.



(A) Time per GMRES iteration

(B) Total solving time

FIGURE 33 Time required to solve the resistance problem for the geometry in Figure 32 with GMRES tolerance 10^{-6} in MATLAB, (A) per GMRES iteration and (B) in total. With 82,080 unknowns per segment, the number of unknowns ranges from 82,080 to 984,960. The dashed curves are least-squares fits of $T = A\ n_s \log n_s + B\ n_s + C$ to the data, where n_s is the number of segments. In (A), $A = 5.8$, $B = -2.4$, $C = 12.2$ (s), and in (B), $A = 9.0$, $B = -8.7$, $C = 18.6$ (min). Thus, the time scales as $O(n_s \log n_s)$. Precomputation time (8 min) is not included. GMRES, generalized minimal residual method [Colour figure can be viewed at wileyonlinelibrary.com]

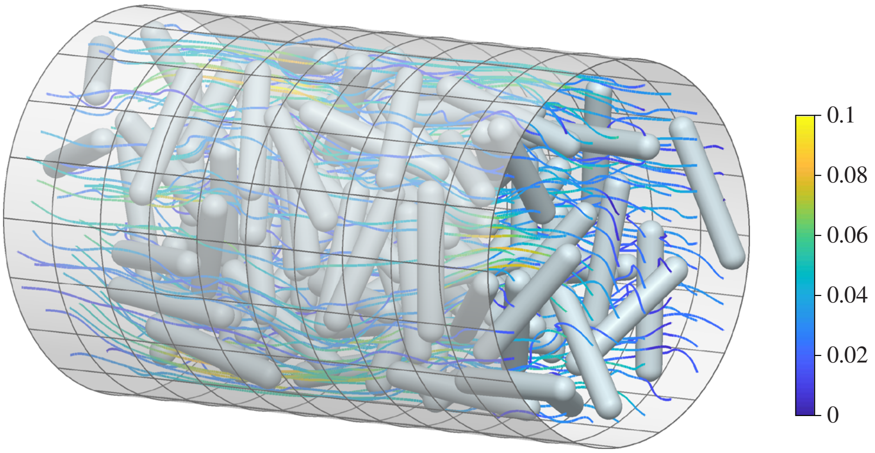


FIGURE 34 Streamlines for the resistance problem (color indicate velocity magnitude) [Colour figure can be viewed at wileyonlinelibrary.com]

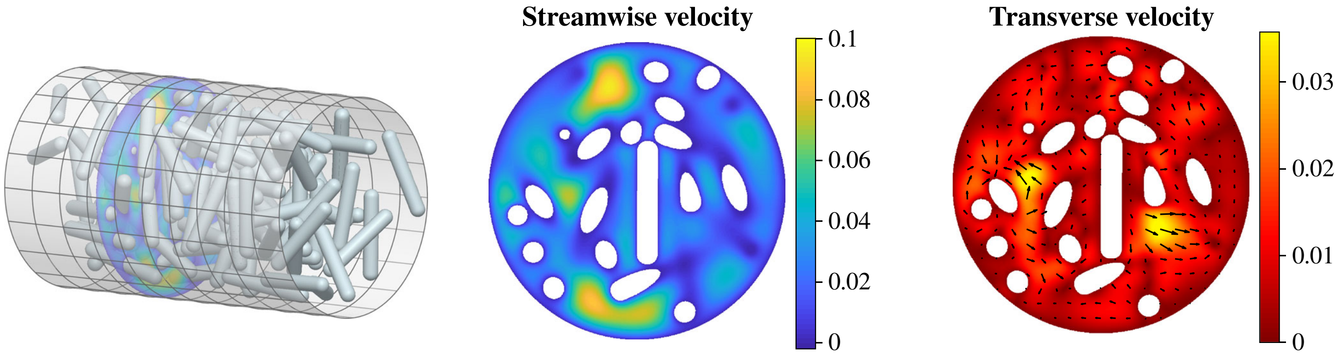


FIGURE 35 Flow field shown in a slice at streamwise position $x = 2/3$ [Colour figure can be viewed at wileyonlinelibrary.com]

7.3.2 | Streamline computation

In the postprocessing step (step 4 of the method summary), streamlines may be computed to visualize the flow field. When using the SE method, the Fourier-space part on the uniform grid can be reused to reduce the computation time, as described in Appendix D. Here, we compute streamlines for a periodic resistance problem with 100 rods in a pipe segment of length 1, otherwise identical to the problem in Section 7.3.1 (including all parameters). Figure 34 shows 95 streamlines; a typical streamline consists of around 3000 points and takes around 2 min to compute (i.e., around 0.04 s per time step). A slice of the same flow field is shown in Figure 35.

8 | EFFECTS OF NONSMOOTH GEOMETRIES

So far, all geometrical objects considered in this article have been smooth. In fact, special care has been taken to ensure that the rod particles, constructed in Appendix B, are everywhere smooth. The reason is that, as noted by Epstein et al.,[35] the convergence of the local expansions used in QBX depends on the smoothness of the boundary close to the expansion center. In this section, we demonstrate this using two different rod particles: one smooth and one nonsmooth, shown in Figure 36(A). The rods are both of length L and radius R, but the smooth rod is constructed as in Appendix B, while the nonsmooth rod consists of a cylinder of radius R and length $L - 2R$ joined to two half-spherical caps of radius R. The nonsmooth rod is thus of class C^1, since the curvature is discontinuous where the cylinder meets the spherical caps.

To illustrate the convergence of QBX, consider rods with $L/R = 20$ (aspect ratio 10), discretized using $n_1 = 35$, $n_2 = 60$, $n_\varphi = 18$ as described in Section 3.1. In Figure 36(B), the onsurface QBX stresslet identity error is plotted as a function of p_{QBX}, in the same way as in Section 6.3 (where this was done for the smooth rod). Clearly, the convergence with respect to p_{QBX} is much worse for the nonsmooth rod compared with the smooth one. The reason for this can be seen in Figure 37: the error decays extremely slowly close to the boundary between the cylinder and the caps, where the curvature is discontinuous. This is clearly a local effect, since the convergence is fine a little bit away from the discontinuity.

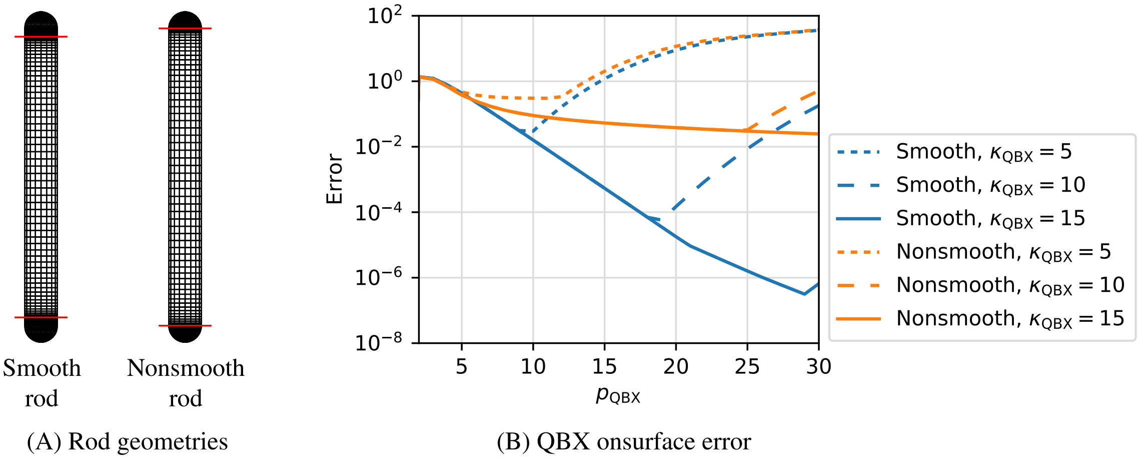


FIGURE 36 (A) A smooth and a nonsmooth rod, both of length L and radius R. Note that each cap of the smooth rod (marked with red lines) has length $1.5R$, while each cap of the nonsmooth rod has length R. (B) Maximal onsurface QBX stresslet identity error for the smooth and nonsmooth rod, with expansion radius $r_{\text{QBX}} = h = 2\pi R/n_{\varphi}$ fixed [Colour figure can be viewed at wileyonlinelibrary.com]

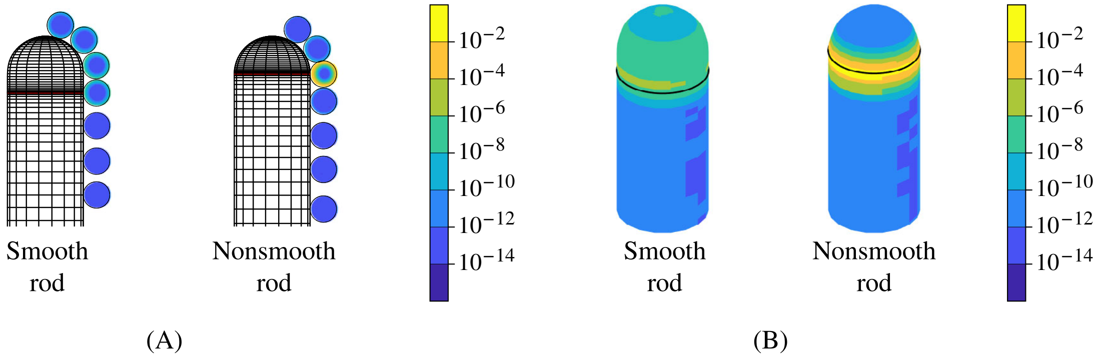


FIGURE 37 (A) The QBX stresslet identity error shown in a slice through selected balls of convergence, for $r_{\text{QBX}} = h$, $p_{\text{QBX}} = 25$, and $\kappa_{\text{QBX}} = 15$. (B) The onsurface QBX error on the rods, with the same parameters as in (A). QBX, quadrature by expansion [Colour figure can be viewed at wileyonlinelibrary.com]

It should be noted that it is entirely possible to use QBX on a nonsmooth geometry, but it requires special measures to be taken. Klöckner et al.[32] applied QBX to a geometry with a corner. In that example, the discretization was dyadically refined around the corner, to ensure that the layer potential appears locally smooth on the scale of the discretization. The same approach could likely be taken also for the nonsmooth rod particle, that is, refining the grid dyadically around the discontinuity. However, constructing the rod to be smooth in the first place has a clear advantage in this case, since no grid refinement is needed.

9 | CONCLUSIONS

We have presented a numerical method based on a boundary integral formulation that can be used to simulate rigid particles in Stokes flow with confining walls. A parameter selection strategy has also been presented for the combined special quadrature used in this method. We have demonstrated that the error of the method is controlled by the special quadrature tolerance as long as the layer density is well-resolved, and that the method scales as $O(N \log N)$ in the number of unknowns N (where N is three times the number of grid points in the primary cell) for fixed grid point concentration. This makes it possible to simulate systems with a large number of particles. The method can deal with particles and walls of different shapes; we have here considered spheroids, rod particles, pipes and plane walls, but it is straightforward to extend the method to any smooth geometry with sufficient symmetry.

To reach the desired accuracy, our method requires the layer density to be well-resolved on the underlying discretization grid, which may not be the case if the distance between particles is close to the grid spacing or smaller. Thus, in time-dependent simulations where particles may come arbitrarily close, the grid resolution may need to be significantly increased. How to avoid excessively high grid resolutions in cases like this is an open research question.

The method could be further improved for example by using local patch-based quadrature for elongated particles to reduce the size of the QBX matrices, and allowing the size of the wall patches to be set adaptively so that the resolution can be focused where particles are close to the wall. It could also be useful to allow parameters such as p_{QBX} to vary along the particle surface (in response to differences in the convergence rate of the local expansions, as seen for example in Figure 37), and to allow the expansion centers for QBX to be placed independently of the grid points of the discretization, so that the centers can be placed closer to the surface in order to decrease the expansion order p_{QBX}. Furthermore, if analytical quadrature error estimates were available, these could replace the numerical experiments used to select threshold distances and the QBX upsampling factor.

ACKNOWLEDGMENTS

This work has been supported by the Göran Gustafsson Foundation for Research in Natural Sciences and by the Swedish Research Council under grant no. 2015-04998. The authors gratefully acknowledge this support. The authors are grateful to Dr. Ludvig af Klinteberg for providing us with an implementation of QBX for spheroidal particles, parts of which were reused in this work.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Joar Bagge https://orcid.org/0000-0002-6953-8058

REFERENCES

1. Guasto JS, Rusconi R, Stocker R. Fluid mechanics of planktonic microorganisms. *Annu Rev Fluid Mech*. 2012;44:373-400. https://doi.org/10.1146/annurev-fluid-120710-101156.
2. Lu L., Morse M. J., Rahimian A., Stadler G., Zorin D.. Scalable simulation of realistic volume fraction red blood cell flows through vascular networks. Paper presented at: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '19); vol. 6, November 2019; Denver, CO. https://doi.org/10.1145/3295500.3356203.
3. Squires TM, Quake SR. Microfluidics: fluid physics at the nanoliter scale. *Rev Mod Phys*. 2005;77(3):977-1026. https://doi.org/10.1103/RevModPhys.77.977.
4. Reddig S, Stark H. Nonlinear dynamics of spherical particles in Poiseuille flow under creeping-flow condition. *J Chem Phys*. 2013;138(23):234902. https://doi.org/10.1063/1.4809989.
5. Gontijo RG, Cunha FR. Dynamic numerical simulations of magnetically interacting suspensions in creeping flow. *Powder Technol*. 2015;279:146-165. https://doi.org/10.1016/j.powtec.2015.03.033.
6. Mittal N, Ansari F, Gowda VK, et al. Multiscale control of nanocellulose assembly: transferring remarkable nanoscale fibril mechanics to macroscale fibers. *ACS Nano*. 2018;12(7):6378-6388. https://doi.org/10.1021/acsnano.8b01084.
7. Guasto JS, Ross AS, Gollub JP. Hydrodynamic irreversibility in particle suspensions with nonuniform strain. *Phys Rev E*. 2010;81(6):061401. https://doi.org/10.1103/PhysRevE.81.061401.
8. Pozrikidis C.. *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge, MA: Cambridge University Press; 1992. https://doi.org/10.1017/CBO9780511624124.
9. Atkinson K. E.. *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge, MA: Cambridge University Press; 1997. https://doi.org/10.1017/CBO9780511626340.
10. Kress R.. *Linear Integral Equations*. New York, NY: Springer; 2014. 3rd. https://doi.org/10.1007/978-1-4614-9593-2.
11. Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J Sci Stat Comput*. 1986;7(3):856-869. https://doi.org/10.1137/0907058.
12. Greengard L, Rokhlin V. A fast algorithm for particle simulations. *J Comput Phys*. 1987;73(2):325-348. https://doi.org/10.1016/0021-9991(87)90140-9.
13. Greengard L, Rokhlin V. A new version of the fast multipole method for the laplace equation in three dimensions. *Acta Numer*. 1997;6:229-269. https://doi.org/10.1017/S0962492900002725.
14. Deserno M, Holm C. How to mesh up Ewald sums. I. a theoretical and numerical comparison of various particle mesh routines. *J Chem Phys*. 1998;109(18):7678-7693. https://doi.org/10.1063/1.477414.
15. Lindbo D, Tornberg A-K. Spectral accuracy in fast Ewald-based methods for particle simulations. *J Comput Phys*. 2011;230(24):8744-8761. https://doi.org/10.1016/j.jcp.2011.08.022.

16. af Klinteberg L, Shamshirgar DS, Tornberg AK. Fast Ewald summation for free-space Stokes potentials. *Res Math Sci*. 2017;4(1):1-32. https://doi.org/10.1186/s40687-016-0092-7.
17. Swaminathan TN, Mukundakrishnan K, Hu HH. Sedimentation of an ellipsoid inside an infinitely long tube at low and intermediate Reynolds numbers. *J Fluid Mech*. 2006;551:357-385. https://doi.org/10.1017/S0022112005008402.
18. Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. *J Comput Phys*. 2005;209:448-476. https://doi.org/10.1016/j.jcp.2005.03.017.
19. Sharma N, Patankar NA. A fast computation technique for the direct numerical simulation of rigid particulate flows. *J Comput Phys*. 2005;205:439-457. https://doi.org/10.1016/j.jcp.2004.11.012.
20. Qi D, Luo L-S. Rotational and orientational behaviour of three-dimensional spheroidal particles in Couette flows. *J Fluid Mech*. 2003;477:201-213. https://doi.org/10.1017/S0022112002003191.
21. Schlauch E, Ernst M, Seto R, Briesen H, Sommerfeld M, Behr M. Comparison of three simulation methods for colloidal aggregates in Stokes flow: finite elements, lattice Boltzmann and Stokesian dynamics. *Comput Fluids*. 2013;86:199-209. https://doi.org/10.1016/j.compfluid.2013.07.005.
22. Wu J, Shu C. Particulate flow simulation via a boundary condition-enforced immersed boundary-lattice Boltzmann scheme. *Commun Comput Phys*. 2010;7(4):793-812. https://doi.org/10.4208/cicp.2009.09.054.
23. Sierou A, Brady JF. Accelerated stokesian dynamics simulations. *J Fluid Mech*. 2001;448:115-146. https://doi.org/10.1017/S0022112001005912.
24. Wang M, Brady JF. Spectral Ewald acceleration of stokesian dynamics for polydisperse suspensions. *J Comput Phys*. 2016;306:443-477. https://doi.org/10.1016/j.jcp.2015.11.042.
25. Liu D, Keaveny EE, Maxey MR, Karniadakis GE. Force-coupling method for flows with ellipsoidal particles. *J Comp Phys*. 2009;228:3559-3581. https://doi.org/10.1016/j.jcp.2009.01.020.
26. Youngren GK, Acrivos A. Stokes flow past a particle of arbitrary shape: a numerical method of solution. *J Fluid Mech*. 1975;69(2):377-403. https://doi.org/10.1017/S0022112075001486.
27. Keaveny EE, Shelley MJ. Applying a second-kind boundary integral equation for surface tractions in Stokes flow. *J Comput Phys*. 2011;230(5):2141-2159. https://doi.org/10.1016/j.jcp.2010.12.010.
28. Helsing J, Ojala R. On the evaluation of layer potentials close to their sources. *J Comput Phys*. 2008;227(5):2899-2921. https://doi.org/10.1016/j.jcp.2007.11.024.
29. Ojala R, Tornberg A-K. An accurate integral equation method for simulating multi-phase Stokes flow. *J Comput Phys*. 2015;298:145-160. https://doi.org/10.1016/j.jcp.2015.06.002.
30. Pälsson S, Siegel M, Tornberg A-K. Simulation and validation of surfactant-laden drops in two-dimensional Stokes flow. *J Comput Phys*. 2019;386:218-247. https://doi.org/10.1016/j.jcp.2018.12.044.
31. Barnett A, Wu B, Veerapaneni S. Spectrally accurate quadratures for evaluation of layer potentials close to the boundary for the 2D Stokes and Laplace equations. *SIAM J Sci Comput*. 2015;37(4):B519-B542. https://doi.org/10.1137/140990826.
32. Klöckner A, Barnett A, Greengard L, O'Neil M. Quadrature by expansion: a new method for the evaluation of layer potentials. *J Comput Phys*. 2013;252:332-349. https://doi.org/10.1016/j.jcp.2013.06.027.
33. Rahimian A, Barnett A, Zorin D. Ubiquitous evaluation of layer potentials using quadrature by Kernel-Independent expansion. *BIT Numer Math*. 2018;58:423-456. https://doi.org/10.1007/s10543-017-0689-2.
34. Barnett AH. Evaluation of layer potentials close to the boundary for laplace and Helmholtz problems on analytic planar domains. *SIAM J Sci Comput*. 2014;36(2):A427-A451. https://doi.org/10.1137/120900253.
35. Epstein CL, Greengard L, Klöckner A. On the convergence of local expansions of layer potentials. *SIAM J Numer Anal*. 2013;51(5):2660-2679. https://doi.org/10.1137/120902859.
36. af Klinteberg L, Tornberg AK. Error estimation for quadrature by expansion in layer potential evaluation. *Adv Comput Math*. 2017;43:195-234. https://doi.org/10.1007/s10444-016-9484-x.
37. af Klinteberg L, Tornberg AK. A fast integral equation method for solid particles in viscous flow using quadrature by expansion. *J Comp Phys*. 2016;326:420-445. https://doi.org/10.1016/j.jcp.2016.09.006.
38. Rachh M. Integral Equation Methods for Problems in Electrostatics, Elastostatics and Viscous Flow [Ph.D thesis]. New York, NY: New York University; 2015. https://search.proquest.com/docview/1710781501.
39. Rachh M, Klöckner A, O'Neil M. Fast algorithms for quadrature by expansion I: globally valid expansions. *J Comput Phys*. 2017;345:706-731. https://doi.org/10.1016/j.jcp.2017.04.062.
40. Wala M, Klöckner A. A fast algorithm with error bounds for quadrature by expansion. *J Comput Phys*. 2018;374:135-162. https://doi.org/10.1016/j.jcp.2018.05.006.
41. Wala M, Klöckner A. A fast algorithm for quadrature by expansion in three dimensions. *J Comput Phys*. 2019;388:655-689. https://doi.org/10.1016/j.jcp.2019.03.024.
42. Wala M, Klöckner A. Optimization of fast algorithms for global quadrature by expansion using target-specific expansions. *J Comput Phys*. 2020;403:108976. https://doi.org/10.1016/j.jcp.2019.108976.
43. Siegel M, Tornberg A-K. A local target specific quadrature by expansion method for evaluation of layer potentials in 3D. *J Comput Phys*. 2018;364:365-392. https://doi.org/10.1016/j.jcp.2018.03.006.
44. af Klinteberg L, Tornberg AK. Adaptive quadrature by expansion for layer potential evaluation in two dimensions. *SIAM J Sci Comput*. 2018;40(3):A1225-A1249. https://doi.org/10.1137/17M1121615.

45. Ying L, Biros G, Zorin D. A high-order 3D boundary integral equation solver for elliptic PDEs in smooth domains. *J Comput Phys*. 2006;219:247-275. https://doi.org/10.1016/j.jcp.2006.03.021.
46. Sorgentone C, Tornberg A-K. A highly accurate boundary integral equation method for surfactant-laden drops in 3D. *J Comput Phys*. 2018;360:167-191. https://doi.org/10.1016/j.jcp.2018.01.033.
47. Malhotra D., Rahimian A., Zorin D., Biros G. A parallel algorithm for long-timescale simulation of concentrated vesicle suspensions in three dimensions (Preprint); 2018. https://cims.nyu.edu/~malhotra/files/pubs/ves3d.pdf.
48. Beale JT, Lai M-C. A method for computing nearly singular integrals. *SIAM J Numer Anal*. 2001;38(6):1902-1925. https://doi.org/10.1137/S0036142999362845.
49. Tlupova S, Beale JT. Nearly singular integrals in 3D stokes flow. *Commun Comput Phys*. 2013;14(5):1207-1227. https://doi.org/10.4208/cicp.020812.080213a.
50. Beale JT, Ying W, Wilson JR. A simple method for computing singular or nearly singular integrals on closed surfaces. *Commun Comput Phys*. 2016;20(3):733-753. https://doi.org/10.4208/cicp.030815.240216a.
51. Tlupova S, Beale JT. Regularized single and double layer integrals in 3D stokes flow. *J Comput Phys*. 2019;386:568-584. https://doi.org/10.1016/j.jcp.2019.02.031.
52. Pérez-Arancibia C, Faria LM, Turc C. Harmonic density interpolation methods for high-order evaluation of Laplace layer potentials in 2D and 3D. *J Comput Phys*. 2019;376:411-434. https://doi.org/10.1016/j.jcp.2018.10.002.
53. Carvalho C, Khatri S, Kim AD. Close evaluation of layer potentials in three dimensions (Preprint); 2018. arXiv:1807.02474 [math.NA].
54. Carvalho C., Khatri S., Kim AD. Asymptotic approximations for the close evaluation of double-layer potentials (Preprint); 2018. arXiv:1810.02483 [math.NA].
55. Corona E, Veerapaneni S. Boundary integral equation analysis for suspension of spheres in Stokes flow. *J Comput Phys*. 2018;362:327-345. https://doi.org/10.1016/j.jcp.2018.02.017.
56. Bruno OP, Kunyansky LA. A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests, and applications. *J Comput Phys*. 2001;169(1):80-110. https://doi.org/10.1006/jcph.2001.6714.
57. Zhao H, Isfahani AHG, Olson LN, Freund JB. A spectral boundary integral method for flowing blood cells. *J Comput Phys*. 2010;229:3726-3744. https://doi.org/10.1016/j.jcp.2010.01.024.
58. af Klinteberg L, Tornberg AK. Fast Ewald summation for Stokesian particle suspensions. *Int J Numer Methods Fluids*. 2014;76(10):669-698. https://doi.org/10.1002/fld.3953.
59. Lindbo D, Tornberg A-K. Spectrally accurate fast summation for periodic Stokes potentials. *J Comput Phys*. 2010;229(23):8994-9010. https://doi.org/10.1016/j.jcp.2010.08.026.
60. af Klinteberg L.. Ewald summation for the rotlet singularity of Stokes flow (Preprint); 2016. arXiv:1603.07467 [physics.flu-dyn].
61. Lindbo D., af Klinteberg L, Saffar Shamshirgar D. The spectral Ewald unified package; 2018. http://github.com/ludvigak/SE_unified.
62. Bagge J, Tornberg A-K. Accurate quadrature methods with application to Stokes flow with particles in confined geometries. In: Chappell DJ, ed. *Proceedings of the Eleventh UK Conference on Boundary Integral Methods (UKBIM 11); July*. Nottingham; Nottingham Trent University; 2017:15-24. http://irep.ntu.ac.uk/id/eprint/31463.
63. Blake JR. A note on the image system for a stokeslet in a no-slip boundary. *Proc Camb Philol Soc*. 1971;70(2):303-310. https://doi.org/10.1017/S0305004100049902.
64. Gimbutas Z, Greengard L, Veerapaneni S. Simple and efficient representations for the fundamental solutions of Stokes flow in a half-space. *J Fluid Mech*. 2015;776:R1. https://doi.org/10.1017/jfm.2015.302.
65. Srinivasan S, Tornberg A-K. Fast Ewald summation for Green's functions of Stokes flow in a half-space. *Res Math Sci*. 2018;5(3):35. https://doi.org/10.1007/s40687-018-0153-1.
66. Kim S., Karrila S. J.. *Microhydrodynamics: Principles and Selected Applications*. Boston, MA: Butterworth–Heinemann; 1991. https://doi.org/10.1016/C2013-0-04644-0.
67. Power H, Miranda G. Second kind integral equation formulation of Stokes' flows past a particle of arbitrary shape. *SIAM J Appl Math*. 1987;47(4):689-698. https://doi.org/10.1137/0147047.
68. Olver FWJ, Olde Daalhuis AB, Lozier DW, et al. NIST digital library of mathematical functions; 2019. http://dlmf.nist.gov/Release 1.0.25 of 2019-12-15.
69. Trefethen LN, Weideman JAC. The exponentially convergent trapezoidal rule. *SIAM Rev*. 2014;56(3):385-458. https://doi.org/10.1137/130932132.
70. Berrut J-P, Trefethen LN. Barycentric lagrange interpolation. *SIAM Rev*. 2004;46(3):501-517. https://doi.org/10.1137/S0036144502417715.
71. Tornberg A-K, Greengard L. A fast multipole method for the three-dimensional Stokes equations. *J Comput Phys*. 2008;227(3):1613-1619. https://doi.org/10.1016/j.jcp.2007.06.029.
72. Saffar Shamshirgar D, Tornberg AK. Fast Ewald summation for electrostatic potentials with arbitrary periodicity (Preprint); 2017. arXiv:1712.04732 [math.NA].

How to cite this article: Bagge J, Tornberg A-K. Highly accurate special quadrature methods for Stokesian particle suspensions in confined geometries. *Int J Numer Meth Fluids*. 2021;93:2175–2224. https://doi.org/10.1002/fld.4970

APPENDIX A. THE STRESSLET IDENTITY FOR PLANE WALLS AND PIPES

Here, we show that a variant of the stresslet identity (5) holds for a pair of parallel infinite plane walls (in Section A.1) and an infinitely long pipe (in Section A.2).

A.1 A pair of parallel plane walls

Let Γ_1 and Γ_2 be two parallel infinite planes oriented as shown in Figure A1, one wall placed at $x_3 = a$ and the other at $x_3 = -a$ for some $a > 0$. Let the domain between the two walls (which we will think of as the fluid domain) be denoted by Ω.

Let $\tilde{\Gamma} = \Gamma_1 \cup \Gamma_2$, and let $\tilde{\boldsymbol{q}} \in \mathbb{R}^3$ be any constant vector. We shall show that

$$\boldsymbol{\mathcal{D}}[\tilde{\Gamma}, \tilde{\boldsymbol{q}}](\boldsymbol{x}) = \begin{cases} -8\pi\tilde{\boldsymbol{q}}, & \text{if } \boldsymbol{x} \in \Omega, \\ -4\pi\tilde{\boldsymbol{q}}, & \text{if } \boldsymbol{x} \in \tilde{\Gamma}, \\ \boldsymbol{0}, & \text{otherwise,} \end{cases} \tag{A1}$$

where the double layer potential $\boldsymbol{\mathcal{D}}$ is given by (3).

Since $\tilde{\boldsymbol{q}}$ is constant and the normals are given by $\boldsymbol{n}^1 = (0, 0, -1)$ and $\boldsymbol{n}^2 = (0, 0, 1)$, we can write

$$\mathcal{D}_i[\tilde{\Gamma}, \tilde{\boldsymbol{q}}](\boldsymbol{x}) = \tilde{q}_j \int_{\tilde{\Gamma}} T_{ijk}(\boldsymbol{x}-\boldsymbol{y}) n_k(\boldsymbol{y}) \, \mathrm{d}S(\boldsymbol{y}) \tag{A2}$$

$$= -\tilde{q}_j \underbrace{\int_{\Gamma_1} T_{ij3}(\boldsymbol{x}-\boldsymbol{y}) \, \mathrm{d}S(\boldsymbol{y})}_{J^1_{ij}(\boldsymbol{x})} + \tilde{q}_j \underbrace{\int_{\Gamma_2} T_{ij3}(\boldsymbol{x}-\boldsymbol{y}) \, \mathrm{d}S(\boldsymbol{y})}_{J^2_{ij}(\boldsymbol{x})}. \tag{A3}$$

The two integrals which we have called J^1_{ij} and J^2_{ij} can both be expressed in terms of the integral

$$J^0_{ij}(\boldsymbol{x}) = \int_{\Gamma_0} T_{ij3}(\boldsymbol{x}-\boldsymbol{y}) \, \mathrm{d}S(\boldsymbol{y}), \tag{A4}$$

where $\Gamma_0 = \{\boldsymbol{x} \in \mathbb{R}^3 : x_3 = 0\}$. The integrals are related through $J^0_{ij}(\boldsymbol{x}) = J^1_{ij}(\boldsymbol{x} + a\boldsymbol{e}_3) = J^2_{ij}(\boldsymbol{x} - a\boldsymbol{e}_3)$, with $\boldsymbol{e}_3 = (0, 0, 1)$. In fact, since Γ_0 is infinite, the integral $J^0_{ij}(\boldsymbol{x})$ as given by (A4) depends only on x_3, that is,

$$J^0_{ij}(\boldsymbol{x}) = J^0_{ij}(x_3) = \int_{\Gamma_0} T_{ij3}(x_3\boldsymbol{e}_3 - \boldsymbol{y}) \, \mathrm{d}S(\boldsymbol{y}). \tag{A5}$$

Inserting the expression for the stresslet $\boldsymbol{T}$ from (3) into (A5), we find

$$J^0_{ij}(x_3) = -6x_3 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{(x_3\delta_{i3} - y_i)(x_3\delta_{j3} - y_j)}{(y_1^2 + y_2^2 + x_3^2)^{5/2}} \mathrm{d}y_1 \mathrm{d}y_2, \tag{A6}$$

where $y_3 = 0$. This double integral can be computed analytically, and the result is

$$J^0_{ij}(x_3) = -4\pi \ \mathrm{sgn}(x_3)\delta_{ij}, \tag{A7}$$

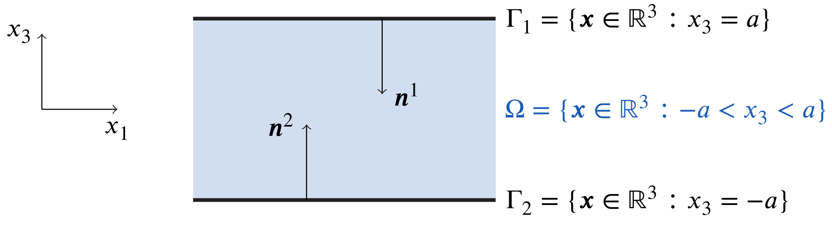

x_3
x_1
$\boldsymbol{n}^1$
$\boldsymbol{n}^2$
$\Gamma_1 = \{\boldsymbol{x} \in \mathbb{R}^3 : x_3 = a\}$
$\Omega = \{\boldsymbol{x} \in \mathbb{R}^3 : -a < x_3 < a\}$
$\Gamma_2 = \{\boldsymbol{x} \in \mathbb{R}^3 : x_3 = -a\}$

FIGURE A1 Two parallel infinite planes Γ_1 and Γ_2 [Colour figure can be viewed at wileyonlinelibrary.com]

where sgn(·) denotes the sign function. Using the relations $J^1_{ij}(\boldsymbol{x}) = J^0_{ij}(x_3 - a)$ and $J^2_{ij}(\boldsymbol{x}) = J^0_{ij}(x_3 + a)$ and inserting (A7) into (A3), we get

$$\mathcal{D}_i[\tilde{\Gamma}, \tilde{\boldsymbol{q}}](\boldsymbol{x}) = 4\pi\tilde{q}_i \operatorname{sgn}(x_3 - a) - 4\pi\tilde{q}_i \operatorname{sgn}(x_3 + a). \tag{A8}$$

From this the result (A1) follows.

A.2 A pipe

Let now $\tilde{\Gamma}$ be an infinitely long pipe given by the equation $x_2^2 + x_3^2 = a^2$ for some $a > 0$, as shown in Figure A2. Let the domain inside the pipe be denoted by Ω. We shall show that for any constant vector $\tilde{\boldsymbol{q}} \in \mathbb{R}^3$, the identity (A1) holds.

Let us introduce cylindrical coordinates and write $\boldsymbol{x} = x_1\boldsymbol{e}_1 + r\boldsymbol{e}_\varphi$ and $\boldsymbol{y} = y_1\boldsymbol{e}_1 + a\boldsymbol{e}_\theta$ for the evaluation point and integration variable, respectively. The unit vectors are given by

$$\boldsymbol{e}_1 = (1, 0, 0), \quad \boldsymbol{e}_\varphi = (0, \cos\varphi, \sin\varphi) \quad \text{and} \quad \boldsymbol{e}_\theta = (0, \cos\theta, \sin\theta), \tag{A9}$$

and $r \geq 0$. Using the fact that the normal vector is given by $\boldsymbol{n}(\boldsymbol{y}) = -\boldsymbol{e}_\theta$, we can write the double layer potential from (3) as

$$\mathcal{D}_i[\tilde{\Gamma}, \tilde{\boldsymbol{q}}](\boldsymbol{x}) = -\tilde{q}_j \int_0^{2\pi} \int_{-\infty}^{\infty} T_{ijk}(x_1\boldsymbol{e}_1 + r\boldsymbol{e}_\varphi - y_1\boldsymbol{e}_1 - a\boldsymbol{e}_\theta)(\boldsymbol{e}_\theta)_k \, a \, \mathrm{d}y_1\mathrm{d}\theta, \tag{A10}$$

where $(\boldsymbol{e}_\theta)_k$ denotes the kth component of $\boldsymbol{e}_\theta$. Using the variable substitution $y_1 - x_1 = u$, we can eliminate x_1. Writing out the stresslet $\boldsymbol{T}$ from (3), and using a few trigonometric identities, the integral in (A10) can be written as

$$\mathcal{D}_i[\tilde{\Gamma}, \tilde{\boldsymbol{q}}](\boldsymbol{x}) = 6\tilde{q}_j a \underbrace{\int_0^{2\pi} \int_{-\infty}^{\infty} \frac{(r\boldsymbol{e}_\varphi - u\boldsymbol{e}_1 - a\boldsymbol{e}_\theta)_i(r\boldsymbol{e}_\varphi - u\boldsymbol{e}_1 - a\boldsymbol{e}_\theta)_j(r\cos(\varphi - \theta) - a)}{\left(u^2 + r^2 + a^2 - 2ra\cos(\varphi - \theta)\right)^{5/2}} \, \mathrm{d}u \, \mathrm{d}\theta}_{I_{ij}(r,\varphi)}. \tag{A11}$$

At this point it is not immediately apparent that the integral which we have called $I_{ij}(r, \varphi)$ is independent of φ, but that does indeed turn out to be the case. We expect the offdiagonal elements of I_{ij} to be zero, which can be verified by first integrating in u and then in θ. It thus remains to compute the diagonal elements of I_{ij}.

To compute I_{11}, first integrate in u using the formula

$$\int_{-\infty}^{\infty} \frac{u^2}{(u^2 + C)^{5/2}} \, \mathrm{d}u = \frac{2}{3C}, \qquad C > 0. \tag{A12}$$

The outer integral becomes

$$I_{11}(r, \varphi) = \frac{2}{3} \int_0^{2\pi} \frac{r\cos(\varphi - \theta) - a}{r^2 + a^2 - 2ra\cos(\varphi - \theta)} \, \mathrm{d}\theta. \tag{A13}$$

The variable φ can now be eliminated using the substitution $\theta - \varphi = v$ (and the limits shifted back to $[0, 2\pi]$ due to periodicity). The value of the integral can then be calculated to be

$$I_{11}(r, \varphi) = \frac{2\pi}{3a} \left(\operatorname{sgn}(r - a) - 1\right), \qquad r \geq 0, \tag{A14}$$

where sgn(·) is the sign function.

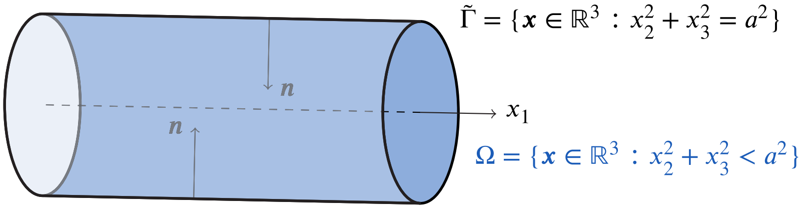

$\tilde{\Gamma} = \{\boldsymbol{x} \in \mathbb{R}^3 : x_2^2 + x_3^2 = a^2\}$
$\boldsymbol{n}$
$\boldsymbol{n}$
x_1
$\Omega = \{\boldsymbol{x} \in \mathbb{R}^3 : x_2^2 + x_3^2 < a^2\}$

FIGURE A2 An infinite cylindrical pipe $\tilde{\Gamma}$ [Colour figure can be viewed at wileyonlinelibrary.com]

To compute I_{22}, first integrate in u using the formula

$$\int_{-\infty}^{\infty} \frac{1}{(u^2+C)^{5/2}}\,\mathrm{d}u = \frac{4}{3C^2}, \qquad C>0, \tag{A15}$$

to get

$$I_{22}(r,\varphi) = \frac{4}{3}\int_0^{2\pi} \frac{(r\cos\varphi - a\cos\theta)^2\,(r\cos(\varphi-\theta)-a)}{(r^2+a^2-2ra\cos(\varphi-\theta))^2}\,\mathrm{d}\theta. \tag{A16}$$

Using the substitution $\theta - \varphi = v$ and shifting the limits back to $[0, 2\pi]$ yields the integral

$$I_{22}(r,\varphi) = \frac{4}{3}\int_0^{2\pi} \frac{(r\cos\varphi - a\cos v\cos\varphi + a\sin v\sin\varphi)^2\,(r\cos v - a)}{(r^2+a^2-2ra\cos v)^2}\,\mathrm{d}v, \tag{A17}$$

which we compute by expanding the square in the numerator, thus splitting the integral into six terms, after which each term can be integrated separately. The result is

$$I_{22}(r,\varphi) = \frac{2\pi}{3a}\left(\operatorname{sgn}(r-a)-1\right), \qquad r \geq 0. \tag{A18}$$

Note that the dependence on φ disappears when summing the six terms to get the above result.

Finally, to compute I_{33}, we again start by integrating in u using (A15), after which we use the substitution $\theta - \varphi = v$ to get

$$I_{33}(r,\varphi) = \frac{4}{3}\int_0^{2\pi} \frac{(r\sin\varphi - a\cos v\sin\varphi - a\sin v\cos\varphi)^2(r\cos v - a)}{(r^2+a^2-2ra\cos v)^2}\,\mathrm{d}v. \tag{A19}$$

Comparing (A17) and (A19), note that $I_{33}(r, \varphi + \pi/2) = I_{22}(r, \varphi)$. But as we saw in (A18), I_{22} does not depend on φ, so $I_{33} = I_{22}$.

To summarize, we have shown that

$$I_{ij}(r,\varphi) = \frac{2\pi}{3a}\left(\operatorname{sgn}(r-a)-1\right)\delta_{ij}. \tag{A20}$$

Inserting this into (A11), we find that

$$\mathcal{D}_i[\tilde{\Gamma}, \tilde{\boldsymbol{q}}](\boldsymbol{x}) = 4\pi\tilde{q}_i\left(\operatorname{sgn}(r-a)-1\right), \tag{A21}$$

from which the result (A1) follows for the pipe.

APPENDIX B. CONSTRUCTION OF SMOOTH ROD PARTICLES

In this section, we describe how the rod particles are constructed to ensure that they are smooth everywhere. Recall from Section 3.1 the parametrization

$$\begin{cases} x_1 = \varrho(\theta; L, R)\cos\varphi, \\ x_2 = \varrho(\theta; L, R)\sin\varphi, \\ x_3 = \beta(\theta; L, R), \end{cases} \tag{B1}$$

of the rod, where $\varphi \in [0, 2\pi)$ and $\theta \in [0, \pi]$ are parameters, L is the length of the rod and R the radius. The goal here is to derive the shape functions $\varrho(\cdot\,; L, R) : [0, \pi] \to [0, R]$ and $\beta(\cdot\,; L, R) : [0, \pi] \to [-\frac{1}{2}L, \frac{1}{2}L]$ so that the rod has the smooth shape shown in Figure B1. The rod consists of three smoothly joined parts: a top cap, corresponding to $\theta \in I_1 = [0, \pi/3]$;

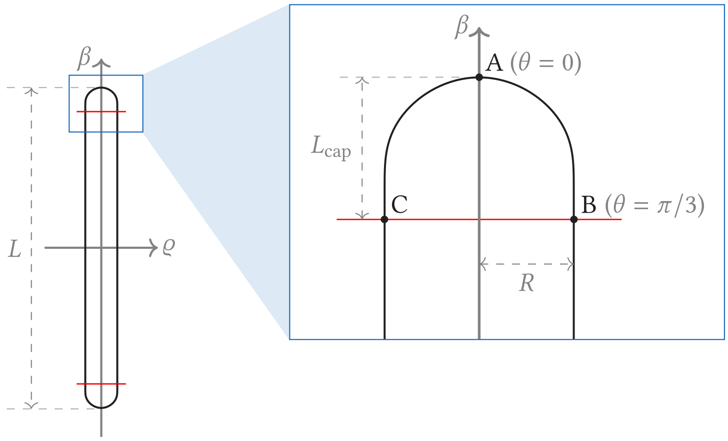


FIGURE B1 The shape of the smooth rod, here with $L=10$ and $R=0.5$ [Colour figure can be viewed at wileyonlinelibrary.com]

a middle cylinder, corresponding to $\theta \in I_2 = [\pi/3, 2\pi/3]$; and a bottom cap, corresponding to $\theta \in I_3 = [2\pi/3, \pi]$. Let the length of each cap be L_{cap}, as shown in Figure B1. The ratio L_{cap}/R determines the aspect ratio of the cap. Here, we fix this ratio by setting L_{cap} to

$$L_{\text{cap}} = 1.5R, \tag{B2}$$

which gives the cap a shape similar to a half-sphere. The length of the middle cylinder is then

$$L_{\text{mid}} = L - 3R. \tag{B3}$$

However, note that the derivation below is valid for any value of $L_{\text{cap}} \in (0, L/2)$, with $L_{\text{mid}} = L - 2L_{\text{cap}}$.

Let us for fixed L and R define $\mathbf{g}(\theta) = (g_1(\theta), g_2(\theta)) = (\varrho(\theta; L, R), \beta(\theta; L, R))$. For the middle cylinder, the parametrization is

$$g_1(\theta) = R, \quad g_2(\theta) = \left(1 - \frac{3}{\pi}\theta\right) L_{\text{mid}} + \frac{L_{\text{mid}}}{2}, \quad \theta \in I_2 = [\pi/3, 2\pi/3]. \tag{B4}$$

Note that g_2 is simply an affine function of θ. At the endpoints of the interval I_2 we have

$$\begin{aligned} \mathbf{g}(\pi/3) &= (R, L_{\text{mid}}/2), \\ \mathbf{g}'(\pi/3) &= (0, -(3/\pi)L_{\text{mid}}), \\ \mathbf{g}^{(n)}(\pi/3) &= (0,0), \quad n \geq 2, \end{aligned} \qquad \text{and} \qquad \begin{aligned} \mathbf{g}(2\pi/3) &= (R, -L_{\text{mid}}/2), \\ \mathbf{g}'(2\pi/3) &= (0, -(3/\pi)L_{\text{mid}}), \\ \mathbf{g}^{(n)}(2\pi/3) &= (0,0), \quad n \geq 2. \end{aligned} \tag{B5}$$

Our goal is now to extend the parametrization $\mathbf{g}(\theta)$ to I_1 and I_3 in a way such that the unit tangent vector $\mathbf{g}'(\theta)/|\mathbf{g}'(\theta)|$ and its higher derivatives are continuous everywhere. As an intermediate step we introduce an auxiliary function $\hat{\mathbf{g}}(t) = (\hat{g}_1(t), \hat{g}_2(t))$ with a different parameter $t \in [-1, 1]$. The function $\hat{\mathbf{g}}$ should trace the curve from C to B via A in Figure B1, with C corresponding to $t = -1$, A corresponding to $t = 0$ and B corresponding to $t = 1$. We will later relate $t \in [0, 1]$ to $\theta \in [0, \pi/3]$ to get the final parametrization. At this point, note that to match (B5) we must require

$$\begin{aligned} \hat{\mathbf{g}}(1) &= (R, L_{\text{mid}}/2), \\ \hat{\mathbf{g}}'(1) &= (0, -b), \\ \hat{\mathbf{g}}^{(n)}(1) &= (0,0), \quad n \geq 2, \end{aligned} \qquad \text{and} \qquad \begin{aligned} \hat{\mathbf{g}}(-1) &= (-R, L_{\text{mid}}/2), \\ \hat{\mathbf{g}}'(-1) &= (0, b), \\ \hat{\mathbf{g}}^{(n)}(-1) &= (0,0), \quad n \geq 2, \end{aligned} \tag{B6}$$

where b is some positive constant. In order to construct $\hat{\mathbf{g}}(t)$ we will use a bump function $\psi : \mathbb{R} \to \mathbb{R}$, which must satisfy the following:

- ψ must be infinitely differentiable on $\mathbb{R}$,
- ψ must have compact support in $[-1, 1]$, that is, $\psi(t) = 0$ if $t > 1$ or $t < -1$,
- $\psi(t)$ must be positive for $t \in (-1, 1)$,
- ψ must be even, that is, $\psi(t) = \psi(-t)$ for all $t \in \mathbb{R}$.

We also introduce its primitive function

$$\Psi(t) = \int_0^t \psi(\tau)\,\mathrm{d}\tau, \qquad t \in \mathbb{R}, \tag{B7}$$

which is an odd function since ψ is even. We choose a specific bump function, namely,[||||]

$$\psi(t) = \begin{cases} \frac{(t^2+1)\exp[4t/(t^2-1)]}{\left[(t^2-1)(1+\exp[4t/(t^2-1)])\right]^2}, & \text{if } t \in (-1,1), \\ 0, & \text{otherwise.} \end{cases} \tag{B8}$$

This function has the primitive function

$$\Psi(t) = \begin{cases} -\frac{1}{8}\tanh\left(-\frac{2t}{1-t^2}\right), & \text{if } t \in (-1,1), \\ -\frac{1}{8}, & \text{if } t \leq -1, \\ \frac{1}{8}, & \text{if } t \geq 1. \end{cases} \tag{B9}$$

We now construct $\hat{\mathbf{g}}(t)$ as

$$\hat{g}_1(t) = R\frac{\Psi(t)}{\Psi(1)}, \quad \hat{g}_2(t) = \frac{L_{\mathrm{mid}}}{2} - b\int_{-1}^{t} \frac{\Psi(\tau)}{\Psi(1)}\,\mathrm{d}\tau, \quad t \in [-1,1], \tag{B10}$$

which satisfies (B6). We can determine b by noting that we must have $\hat{g}_2(0) = L/2$ (at point A in Figure B1), which yields

$$b = L_{\mathrm{cap}}\frac{\Psi(1)}{\int_0^1 \Psi(\tau)\,\mathrm{d}\tau}. \tag{B11}$$

The integrals of Ψ in (B10) and (B11) are computed numerically using MATLAB's *integral* function.

Finally, we go from the parameter t to the parameter θ. We would like the discretization points to be distributed as Gauss–Legendre points in the arclength, and so we must choose θ so that it is proportional to the arclength on the caps. Consider the arclength

$$s(t) = \int_0^t |\hat{\mathbf{g}}'(\tau)|\,\mathrm{d}\tau, \quad t \in [0,1]. \tag{B12}$$

Let us then define

$$\theta = G(t) = \frac{\pi}{3}\frac{s(t)}{s(1)}, \quad t \in [0,1], \tag{B13}$$

and note that this defines $\theta \in I_1 = [0, \pi/3]$ as an invertible function of $t \in [0,1]$. We can now define $\mathbf{g}(\theta) = \mathbf{g}(G(t)) = \hat{\mathbf{g}}(t)$ for $t \in [0,1]$, and thus

$$\mathbf{g}(\theta) = \hat{\mathbf{g}}(G^{-1}(\theta)), \qquad \theta \in I_1 = [0, \pi/3]. \tag{B14}$$

The bottom cap should be the reflection of the top cap in the plane corresponding to $\beta = 0$, so

$$\mathbf{g}(\theta) = (g_1(\pi-\theta), -g_2(\pi-\theta)), \qquad \theta \in I_3 = [2\pi/3, \pi]. \tag{B15}$$

Now that we have defined $\mathbf{g}(\theta)$ for all $\theta \in [0,\pi]$, its two components g_1 and g_2 correspond to the shape factors $\varrho(\theta; L, R)$ and $\beta(\theta; L, R)$, respectively, which are to be used in (B1).

[||||] This function was found at https://math.stackexchange.com/a/101484.

APPENDIX C. DERIVATION OF THE SAFETY FACTOR γ

Recall from Section 6 that one may want to select d_{QBX} larger than r_{QBX} since QBX may be faster than the upsampled quadrature due to the precomputation scheme. Let us call the set of points of the QBX region with distance to Γ greater than r_{QBX} the *upper* QBX region, and the set of points with distance to Γ smaller than r_{QBX} the *lower* QBX region, as shown in Figure C1(A). As noted in Section 6.1.2, putting $d_{\text{QBX}} = 2r_{\text{QBX}}$ would lead to some areas of the upper QBX region not falling within any ball of convergence. To avoid this, we introduce a safety factor γ and require that

$$d_{\text{QBX}} \leq 2\gamma r_{\text{QBX}}. \tag{C1}$$

The goal here is to derive the value of the safety factor γ. We assume for simplicity that Γ is a flat surface.

The key is to choose d_{QBX} below the intersection of neighboring balls of convergence, marked by the point C in Figure C1(A). Since the grid on Γ is two-dimensional, the largest distance between neighboring grid points is not h but $\sqrt{2}h$, where h is as shown in Figure C1(B). The four balls of convergence of the expansion centers above the grid points D–G in this figure intersect at distance

$$d_\star(h) = r_{\text{QBX}} + \sqrt{r_{\text{QBX}}^2 - \left(\frac{\sqrt{2}h}{2}\right)^2} \tag{C2}$$

from Γ. Thus, choosing $d_{\text{QBX}} \leq d_\star(h)$ is sufficient to ensure that all points in the upper QBX region fall within a ball of convergence. This restriction on d_{QBX} can be simplified by minimizing $d_\star(h)$ with respect to h, subject to the constraint $0 < h \leq r_{\text{QBX}}$. The result is

$$d_{\star\star} = \min_{0<h\leq r_{\text{QBX}}} d_\star(h) = d_\star(r_{\text{QBX}}) = \left(1 + \frac{1}{\sqrt{2}}\right) r_{\text{QBX}}. \tag{C3}$$

It is thus sufficient to require that $d_{\text{QBX}} \leq d_{\star\star}$. Comparing (C3) and (C1), we see that the safety factor should be

$$\gamma = \frac{1}{2}\left(1 + \frac{1}{\sqrt{2}}\right) \approx 0.85. \tag{C4}$$

This derivation holds when Γ is a flat surface, in which case the requirement (C1) with $\gamma = 0.85$ guarantees that all points in the upper QBX region fall within a ball of convergence, as long as $h \leq r_{\text{QBX}}$. If Γ is curved, this guarantee holds on the concave side of Γ, but not necessarily on the convex side, where d_{QBX} may have to be even smaller for the guarantee to hold. Nonetheless, we use (C1) with $\gamma = 0.85$ also for convex surfaces such as rods and spheroids, and it seems to work

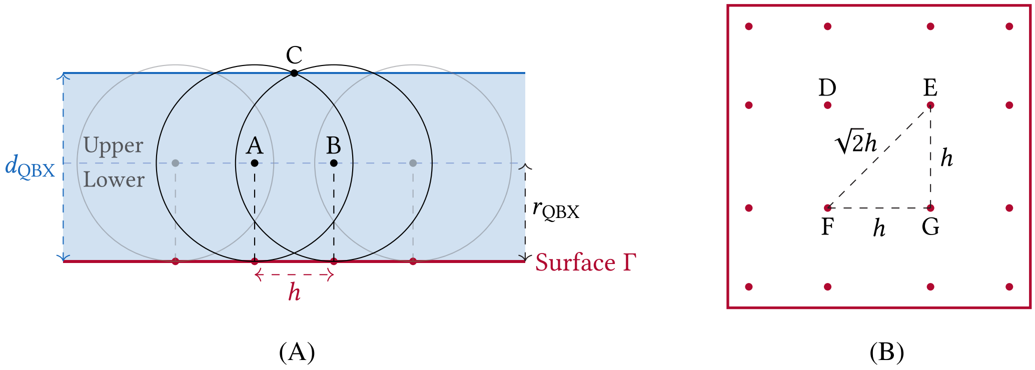


FIGURE C1 (A) Balls of convergence for a flat surface Γ (c.f. Figure 18), seen from the side. (B) Grid points of Γ, seen from above. The grid spacing h is defined as in (33), that is, the largest spacing between grid points in each tensorial direction [Colour figure can be viewed at wileyonlinelibrary.com] well in practice. Of course, the parameter selection strategy (Section 6.2, step 2) will in most cases choose d_{QBX} less than the upper bound $2\gamma r_{\mathrm{QBX}}$.

APPENDIX D. EFFICIENT COMPUTATION OF STREAMLINES IN PERIODIC FLOW

To compute streamlines in a periodic problem such as in Section 7.3, we must first solve the periodic boundary integral equation as described in Section 5 to get the density $\boldsymbol{q}$ on Γ. We can then compute the flow field

$$\boldsymbol{u}(\boldsymbol{x}_{\mathrm{e}}) = \boldsymbol{u}_{\mathrm{bg}}(\boldsymbol{x}_{\mathrm{e}}) + \boldsymbol{\mathcal{D}}^{\mathrm{3P}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}_{\mathrm{e}}) + \sum_{\alpha=1}^{M} \boldsymbol{\mathcal{V}}^{(\alpha),\mathrm{3P}}[\boldsymbol{F}^{(\alpha)}, \boldsymbol{\tau}^{(\alpha)}](\boldsymbol{x}_{\mathrm{e}}) \tag{D1}$$

at any evaluation point $\boldsymbol{x}_{\mathrm{e}}$ in the fluid domain. To compute a streamline we pick any point $\boldsymbol{x}_0$ in the fluid domain and then solve the differential equation

$$\frac{\mathrm{d}\boldsymbol{x}_{\mathrm{e}}}{\mathrm{d}t} = \boldsymbol{u}(\boldsymbol{x}_{\mathrm{e}}(t)), \qquad \boldsymbol{x}_{\mathrm{e}}(0) = \boldsymbol{x}_0. \tag{D2}$$

Of course, (D2) is discretized using some timestepping method, which must evaluate (D1) at every timestep. Recall that the periodic double layer potential $\boldsymbol{\mathcal{D}}^{\mathrm{3P}}$ is split into two parts

$$\boldsymbol{\mathcal{D}}^{\mathrm{3P}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}_{\mathrm{e}}) = \boldsymbol{\mathcal{D}}^{\mathrm{3P,R}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}_{\mathrm{e}}; \xi) + \boldsymbol{\mathcal{D}}^{\mathrm{3P,F}}[\Gamma, \boldsymbol{q}](\boldsymbol{x}_{\mathrm{e}}; \xi) \tag{D3}$$

and similarly for $\boldsymbol{\mathcal{V}}^{(\alpha),\mathrm{3P}}$. The first part $\boldsymbol{\mathcal{D}}^{\mathrm{3P,R}}$ decays fast and is treated according to Section 5. The second part $\boldsymbol{\mathcal{D}}^{\mathrm{3P,F}}$ decays slowly in real space, but since it is smooth its Fourier coefficients decay fast. In the Spectral Ewald method, $\boldsymbol{\mathcal{D}}^{\mathrm{3P,F}}$ as given by (51) is first discretized using the direct quadrature rule (22) to give

$$\mathcal{D}_i^{\mathrm{3P,F},h}[\Gamma, \boldsymbol{q}](\boldsymbol{x}_{\mathrm{e}}; \xi) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \sum_{s=1}^{N} T_{ijl}^{\mathrm{F}}(\boldsymbol{x}_{\mathrm{e}} + \boldsymbol{k}\circ\boldsymbol{B} - \boldsymbol{x}_s; \xi) q_j(\boldsymbol{x}_s) n_l(\boldsymbol{x}_s) w_s. \tag{D4}$$

This is a periodic sum of point sources with strengths $Z_{jl}(\boldsymbol{x}_s) = q_j(\boldsymbol{x}_s) n_l(\boldsymbol{x}_s) w_s$. The SE method[15,58,59] computes the periodic sum (D4) in five steps:

1. **Spreading point sources to a grid:** A three-dimensional uniform grid is constructed over the primary cell. A window function $W(\boldsymbol{r})$ is convolved with the point sources in the primary cell to give

$$H_{jl}(\boldsymbol{x}) = \sum_{s=1}^{N} Z_{jl}(\boldsymbol{x}_s) W([\boldsymbol{x} - \boldsymbol{x}_s]_*). \tag{D5}$$

Here, $[\cdot]_*$ denotes that the shortest periodic distance should be used, that is,

$$[\boldsymbol{r}]_* = \boldsymbol{r} + \boldsymbol{B} \circ \arg\min_{\boldsymbol{k}\in\mathbb{Z}^3} |\boldsymbol{r} + \boldsymbol{B} \circ \boldsymbol{k}|, \tag{D6}$$

where $\boldsymbol{B} = (B_1, B_2, B_3)$ is the size of the periodic cell. In this work the window function is a truncated Gaussian, given by $W(\boldsymbol{r}) = w(r_1)w(r_2)w(r_3)$, where

$$w(r) = \begin{cases} \mathrm{e}^{-A(r/r_{\mathrm{trunc}})^2}, & \text{if } |r| \le r_{\mathrm{trunc}} = h_{\mathrm{F}}P/2, \\ 0, & \text{otherwise.} \end{cases} \tag{D7}$$

Here, h_{F} is the grid spacing of the uniform grid, P is the number of grid points within the support of w, and $A = 0.9^2\pi P/2$. The parameter P is chosen as discussed by af Klinteberg and Tornberg.[58] It is also possible to use other window functions than the Gaussian, as discussed for example by Saffar Shamshirgar and Tornberg.[72]

The function $H_{jl}(\boldsymbol{x})$ as given by (D5) is evaluated on the uniform grid.

2. **FFT:** The three-dimensional Fourier transform $\widehat{H}_{jl}(\boldsymbol{k})$ is computed using the FFT. This is possible since $H_{jl}(\boldsymbol{x})$ is defined on a uniform grid.
3. **Scaling:** The result is multiplied by the Fourier transform of $\boldsymbol{T}^{\mathrm{F}}$, and divided by the Fourier transform of the window function W to undo the convolution in step 1. Since we will convolve again in step 5, this division is done twice. Thus, we here compute

$$\hat{\hat{H}}_i(\boldsymbol{k}) = \widehat{T}^{\mathrm{F}}_{ijl}(\boldsymbol{k};\xi)\frac{1}{[\widehat{W}(\boldsymbol{k})]^2}\widehat{H}_{jl}(\boldsymbol{k}), \tag{D8}$$

where

$$\widehat{T}^{\mathrm{F}}_{ijl}(\boldsymbol{k};\xi) = \sqrt{-1}\frac{\pi}{|\boldsymbol{k}|^2}\left[(\delta_{ij}k_l + \delta_{jl}k_i + \delta_{li}k_j) - 2\frac{k_ik_jk_l}{|\boldsymbol{k}|^2}\right]\left(8 + 2\frac{|\boldsymbol{k}|^2}{\xi^2} + \frac{|\boldsymbol{k}|^4}{\xi^4}\right)\mathrm{e}^{-|\boldsymbol{k}|^2/(4\xi^2)}, \tag{D9}$$

as given in Reference 58.
4. **IFFT:** An inverse FFT is applied to $\hat{\hat{H}}_i(\boldsymbol{k})$ to compute $\tilde{H}_i(\boldsymbol{x})$ on the uniform grid.
5. **Gathering:** In order to compute the final result at the evaluation point $\boldsymbol{x}_{\mathrm{e}}$ (which need not be on the uniform grid), another convolution with the window function is performed, that is,

$$\mathcal{D}_i^{3\mathrm{P},\mathrm{F},h}[\Gamma,\boldsymbol{q}](\boldsymbol{x}_{\mathrm{e}};\xi) = \int_B \tilde{H}_i(\boldsymbol{x})W([\boldsymbol{x}_{\mathrm{e}} - \boldsymbol{x}]_*)\,\mathrm{d}\boldsymbol{x}, \tag{D10}$$

where B denotes the primary cell. The integral in (D10) is evaluated using the trapezoidal rule on the uniform grid, which is spectrally accurate since the integrand is periodic.

Since the density $\boldsymbol{q}$ does not change during the computation of the streamlines, and the evaluation point $\boldsymbol{x}_{\mathrm{e}}$ enters only in step 5 above, it is possible to do step 1–4 once before starting to compute the streamlines, and save $\tilde{H}_i(\boldsymbol{x})$ on the uniform grid from step 4. When the Fourier-space part $\boldsymbol{\mathcal{D}}^{3\mathrm{P},\mathrm{F}}[\Gamma,\boldsymbol{q}](\boldsymbol{x}_{\mathrm{e}};\xi)$ is to be evaluated at $\boldsymbol{x}_{\mathrm{e}}(t)$ at every timestep of solving (D2), it is then enough to do only step 5. This speeds up the computation of the streamlines since evaluating (D10) is fast for a single evaluation point. The real-space part $\boldsymbol{\mathcal{D}}^{3\mathrm{P},\mathrm{R}}[\Gamma,\boldsymbol{q}](\boldsymbol{x}_{\mathrm{e}};\xi)$ must be computed from scratch at every timestep, but this is fast since it is a local sum due to its rapid decay.

The periodic completion flow $\boldsymbol{\mathcal{V}}^{(\alpha),3\mathrm{P}}$ which appears in (D1) is treated in a very similar way; for details, we refer to References 58-60. Note that steps 1–5 of the SE method are also what is used when solving the periodic boundary integral equation as described in Section 5, but in that situation all the evaluation points (i.e., the grid points of Γ) are known in advance so they can all be fed into step 5 at the same time.