A small-scale decomposition for 3D boundary integral computations with surface tension

D. M. Ambrose[b], M. Siegel[a,*], S. Tlupova[c]

[a] *Department of Mathematical Sciences and Center for Applied Mathematics and Statistics, New Jersey Institute of Technology, Newark, NJ 07102, USA*
[b] *Department of Mathematics, Drexel University, Philadelphia, PA 19104*
[c] *Department of Mathematics, University of Michigan, Ann Arbor, MI*

Abstract

An efficient, non-stiff boundary integral method for the initial value problem for interfacial Darcy flow (which is a model of porous media flow) in three space dimensions is presented. We consider a 'doubly-periodic' interface separating two fluids, with surface tension present at the boundary. Surface tension introduces high order (i.e., high derivative) terms in the governing equation, and this imposes a severe stability constraint on explicit time-integration methods. Furthermore, the high order terms appear in a nonlocal operator, which makes it difficult to design an efficient implicit method. The stiffness is removed by developing a small-scale decomposition in the spirit of prior work in the two-dimensional problem by Hou, Lowengrub, and Shelley. In order to develop this small-scale decomposition, we formulate the problem using a generalized isothermal parameterization of the free surface. An additional difficulty is the efficient calculation of the Birkhoff-Rott integral for the velocity of the interface. We present a new algorithm, based on Ewald summation, to compute this in $O(N \log N)$ operations, where N is the number of interface grid points. Our non-stiff method is expected to apply widely to problems for doubly-periodic interfacial flow with surface tension which have a boundary integral formulation.

Keywords: Interfacial flow, boundary integral method, surface tension, Ewald summation
2000 MSC: 76D27, 65M38

1. Introduction

In this paper, we develop an efficient boundary integral method for interfacial flow in three-dimensional porous media, with velocity given by Darcy's law and accounting for surface tension on the interface. The two-dimensional version of this flow, or Hele-Shaw flow, has been widely studied [1] and has served as a model for the development of numerical methods (see, e.g., [2], [3], [4]). When the fluids are taken to be incompressible and immiscible

*Corresponding author
Email addresses: ambrose@math.drexel.edu (D. M. Ambrose), misieg@njit.edu (M. Siegel), stlupova@umich.edu (S. Tlupova)

there is a boundary integral representation of the governing equations. Numerical methods based on this representation, or boundary integral methods, have the advantage that they can be made arbitrarily high order accurate. Thus, they are useful for investigating phenomena that require high accuracy to resolve, such as singularity formation on an interface [5], [6]. We study the 'doubly-periodic' Darcy problem, in which the fluids are each of infinite vertical extent but are periodic in the two horizontal directions. This problem typically has strong stiffness constraints (owing to the presence of surface tension), but we develop a small-scale decomposition to remove the stiffness.

The method we develop is related to the method of Hou, Lowengrub, and Shelley (HLS) for removing the stiffness from two-dimensional interfacial flows with surface tension [3], [7]. In those papers, the authors considered a boundary integral formulation of different interfacial flow problems with surface tension: the Kelvin-Helmholtz and Rayleigh-Taylor problems for inviscid interfacial flow, and Darcy's flow. Of these, the Darcy problem is more stiff, with a third order constraint for stability of an explicit method. The stiffness was removed by reformulating the problem in natural geometric variables and by using a convenient choice of parameterization. In particular, the free surface (which is a one-dimensional curve) was described not by its Cartesian coordinates but instead by its tangent angle and arclength. Furthermore, a non-Lagrangian tangential velocity was used in order to keep the curve parameterized by arclength. With these choices, the problem becomes semilinear, and lends itself to an efficient implicit discretization of the leading order or highest derivative terms when time stepping.

There are a number of difficulties for the present problem in three space dimensions which are not present in the two-dimensional setting. The first is that there is no perfect analogue of the arclength-angle formulation used by HLS. In analytical studies of the three-dimensional vortex sheet with surface tension [8], Darcy flow [9], and water waves [10], the first author (together with Nader Masmoudi) studied the evolution of the second fundamental form of the interface, and used an isothermal parameterization to facilitate the analysis. We incorporate the isothermal parameterization in our boundary integral method, which gives it some of the same advantages in three-dimensional flow as the arclength-angle formulation in two dimensions. Isothermal coordinates are always known to exist locally for a surface, but sometimes do not exist globally. In the present setting, we formulate the problem using a generalized isothermal parameterization that holds for a wider class of surfaces. This parameterization will be discussed in detail in §3 below. As in the work by HLS, we enforce the parameterization with an appropriate choice of the tangential velocities. Numerical examples that are presented in §7 are for the special case of a purely isothermal parameterization.

Perhaps the greatest difficulty is in making an efficient calculation of the Birkhoff-Rott integral, which must be computed for the physical normal velocity of the free surface. The Birkhoff-Rott integral is derived from the Biot-Savart law for recovering the fluid velocity from the vorticity. In the present setting, the fluids are irrotational in the bulk but there is a jump in velocity across the interface; there is therefore measure-valued vorticity supported on the interface. The Birkhoff-Rott integral is an integral over the free surface; it is, in

fact, an integral over all of the periodic images of the free surface. In the two-dimensional setting, it is possible to sum over all the periodic images in closed form, and this requires only computing an integral over one period. By contrast, we have no such closed form available, and the sum over all the periodic images needs to be computed. An efficient method for computing this sum, by Ewald summation, is one of the primary contributions of the present work.

The Ewald sum is a rapidly convergent series that represents the interactions of N 'particles' and their periodic images (in our case, from discretization of the Birkhoff-Rott integral). It has two parts: a Fourier or 'reciprocal' sum that accounts for far-field interactions, and a 'real space' sum that captures local interactions. A naive or direct calculation of these sums takes $O(N^2)$ operations. Efforts to develop efficient methods for Ewald summation have been focussed on the triply-periodic geometry, and include the widely-used particle-mesh-Ewald method [11], [12] which has $O(N \log N)$ complexity for distributed or non-clustered particles. Alternatively, Duan and Krasny [13] exploited separability of the reciprocal sum and used a treecode to accelerate computation of the real space sum, to obtain an overall $O(N \log N)$ method for arbitrary particle distributions.

There is much less work in the doubly-periodic geometry of interest here. In an important recent development, Lindbo and Tornberg [14] extended the particle-mesh-Ewald method in [12] to the doubly-periodic geometry. The resulting method is $O(N \log N)$ for distributed particles, but has suboptimal complexity (i.e., greater that $O(N \log N)$) when the particles are clustered, for example, on an interface. We instead follow the approach of Duan and Krasny, but note that fast computation of the reciprocal sum in the doubly-periodic case is hampered by its lack of separability. Therefore, in §6, we first develop a fast $O(N)$ method to compute the reciprocal sum. This can be combined with scaling of Ewald parameters to obtain a simple but suboptimal $O(N^{3/2})$ algorithm (methods based on optimizing Ewald parameters are well-known for the triply-periodic geometry). The addition of a treecode to speed up the computation of the real space sum (see §6.3.3) then gives an overall $O(N \log N)$ algorithm for arbitrary particle distributions. Our method uses the simple quadrature of Haroldsen and Meiron [15] for the spatial discretization of the Birkhoff-Rott integral, which can be made to have arbitrary or $O(h^p)$ accuracy, where h is a measure of the grid size.

Other difficulties for the present problem are the need to find the generalized isothermal parameterization for a given initial surface, and dealing with the fact that our problem only becomes quasilinear rather than semilinear (as in the HLS case) after the reformulation. These problems are resolved in §5 and in our previous work [16], in which we treated a model problem for 3D porous media flow which did not require evaluation of the Birkhoff-Rott integral. In that work the first and second fundamental coefficients are used as dynamical variables, which creates additional technical difficulties. This includes the need to satisfy geometric compatibility conditions among the fundamental coefficients during time evolution, and an accurate and efficient method to recover the free surface from its fundamental coefficients. These problems are largely sidestepped in the current work by using the interface function itself as the dynamical variable. The resulting method removes the high order time-step constraint for stability, while needing only the same number of operations per time

step as an explicit method.

There are a number of other methods for the computation of moving curves and surfaces in three-dimensional flow that have similarities to the current work. For the 3D vortex sheet without surface tension, Hou, Hu, and Zhang studied singularity formation in a model problem [17] using a parameterization in isothermal coordinates, but not for the purpose of removing stiffness. By forming an approximate system, they did not need to compute the Birkhoff-Rott integral; this is similar to our prior work [16]. Hou and Zhang [18] propose and analyze a method for 3D water waves without surface tension that uses isothermal coordinates. The small-scale decomposition has been adapted to treat the motion of one-dimensional filaments in 3D flows [19], [20] and surfaces in axisymmetric flow [21], [22]. Veerapaneni et al. [23], [24], [25] present a non-stiff boundary integral algorithm for a model of vesicle evolution in two-dimensional, axisymmetric, and three-dimensional Stokes flow. Although they do not use isothermal coordinates, their algorithm has some similarities with that presented here. In both cases, the semi-implicit discretization leads to a linear system, and an advantage of isothermal coordinates is that this system can be inverted somewhat more efficiently. A disadvantage is that it is more difficult to make the current method adaptive in space.

The rest of this paper is organized as follows. The governing equations for interfacial porous media flow with surface tension are presented in §2. In §3 we discuss the first and second fundamental forms of a surface and give details of our choice of parameterization of the interface. The small-scale decomposition for our problem is given in §4, and the semi-implicit method is presented in §5. The surface integration method, including the fast Ewald summation, is discussed in §6, and numerical results are presented in §7. Concluding remarks are given in §8, and a derivation of the Ewald sum for our problem is given in Appendix A.

2. Governing Equations

In this section we present the evolution equations for 3D interfacial porous media flow with surface tension. The reader is referred to [16] for details of the derivation.

We consider the flow of two immiscible, incompressible fluids in a three-dimensional porous medium. The sharp interface separating the two fluids is taken to be doubly periodic, that is, with period 2π in the horizontal x and y directions but 'free' in the z direction, in which the fluids are taken to have infinite extent. We denote quantities related to the upper fluid with a subscript 1 and those related to the lower fluid with a subscript 2. The fluid domains are denoted by $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively, and we let S be the interface between them.

Motion of the interface and fluids is driven by gravity, surface tension, and/or a prescribed far-field pressure gradient. For fluid $i \in \{1, 2\}$, we introduce the fluid velocity $\mathbf{V}_i$, pressure p_i, fluid mobility K_i, and density ρ_i, with the latter two quantities constant in fluid i. The constant acceleration due to gravity is g and we let (x, y, z) be a point in space. The fluid velocities are governed by the incompressible Darcy's law, which is written as

$$\mathbf{V}_i = -K_i \nabla (p_i + \rho_i g z) \quad \text{in } \mathcal{D}_i, \tag{1}$$

$$\nabla \cdot \mathbf{V}_i = 0. \tag{2}$$

The free surface S is parameterized as $\mathbf{X}(\vec{\alpha}, t) = (x(\vec{\alpha}, t), y(\vec{\alpha}, t), z(\vec{\alpha}, t))$, where $\vec{\alpha} = (\alpha, \beta)$ and the principal or 'root' interval for α and β is taken to be $[0, 1]$. The unit tangent and normal vectors are given by

$$\hat{\mathbf{t}}^1 = \frac{\mathbf{X}_\alpha}{|\mathbf{X}_\alpha|}, \qquad \hat{\mathbf{t}}^{\mathbf{2}} = \frac{\mathbf{X}_\beta}{|\mathbf{X}_\beta|}, \qquad \hat{\mathbf{n}} = \hat{\mathbf{t}}^{\mathbf{1}} \times \hat{\mathbf{t}}^{\mathbf{2}}. \tag{3}$$

The boundary conditions include the kinematic condition that the surface S moves with the normal velocity of the fluid, which is continuous across S, so that

$$\mathbf{V}_1 \cdot \hat{\mathbf{n}} = \mathbf{V}_2 \cdot \hat{\mathbf{n}} = U \quad \text{on} \quad S. \tag{4}$$

The boundary condition for the pressure at S is given by the Laplace-Young condition

$$p_1 - p_2 = \sigma(\kappa_1 + \kappa_2) \quad \text{on} \quad S,$$

where σ is the positive, constant coefficient of surface tension, and κ_1 and κ_2 are the principal curvatures of S, which are taken to be positive when the intersection of the free-surface S with the corresponding principal plane is concave upward. Furthermore, letting $\hat{\mathbf{k}}$ be the unit normal in the z direction, the boundary condition as $z \to \pm\infty$ is

$$\mathbf{V} \to V_\infty \hat{\mathbf{k}},$$

for a given constant V_∞.

The fluids are irrotational in the bulk with vorticity only present on the free surface, which is to say the free surface is a vortex sheet. Therefore, the fluid velocity on the free surface S is represented by the Birkhoff-Rott integral [26] as

$$\mathbf{V}(\mathbf{X}) = \frac{1}{4\pi} \mathrm{PV} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{j}' \times \frac{(\mathbf{X} - \mathbf{X}')}{|\mathbf{X} - \mathbf{X}'|^3} \, d\alpha' d\beta' + \mathbf{V}_\infty \hat{\mathbf{k}}, \tag{5}$$

where PV denotes principal value integral and

$$\mathbf{j} = \mu_\alpha \mathbf{X}_\beta - \mu_\beta \mathbf{X}_\alpha \tag{6}$$

is the (unnormalized) vortex sheet density. A prime denotes function evaluation at (α', β').

The evolution of the free surface is given by

$$\mathbf{X}_t = U\hat{\mathbf{n}} + V_1 \hat{\mathbf{t}}^{\mathbf{1}} + V_2 \hat{\mathbf{t}}^{\mathbf{2}}, \tag{7}$$

where the normal velocity, U, is determined by the kinematic condition (4) and is therefore given by

$$U = \mathbf{V}(\mathbf{X}) \cdot \hat{\mathbf{n}}.$$

The tangential velocities are chosen to enforce a preferred parameterization of the free surface, and are not determined by the physics. We discuss our choice of V_1 and V_2 in §3 below.

Given the free surface $\mathbf{X}$, the quantities μ_α and μ_β are determined by solving a system of integral equations, which is derived in dimensionless form in [16] and is given by

$$\mu_\alpha = -B\kappa_\alpha - Wz_\alpha + A_T|\mathbf{X}_\alpha|\mathbf{V}\cdot\hat{\mathbf{t}}^{\mathbf{1}}, \tag{8}$$
$$\mu_\beta = -B\kappa_\beta - Wz_\beta + A_T|\mathbf{X}_\beta|\mathbf{V}\cdot\hat{\mathbf{t}}^{\mathbf{2}}. \tag{9}$$

where $\kappa = (\kappa_1+\kappa_2)/2$ is the mean curvature. The above equations have three dimensionless constants related to the physical parameters, which are

$$B = \frac{4\sigma}{(K_1^{-1}+K_2^{-1})U_c l^2}, \qquad W = \frac{2(\rho_1-\rho_2)g}{(K_1^{-1}+K_2^{-1})U_c}, \qquad A_T = \frac{2(K_1-K_2)}{K_1+K_2}.$$

Here, l is the length of the periodic box and U_c is a characteristic velocity. For simplicity, we henceforth take $A_T = V_\infty = 0$ and consider the interface motion to be driven by gravity and surface tension. If $A_T \neq 0$ there is an additional step of solving (8), (9), using, for example, GMRES. With this simplification, the above equations for μ then become a single equation

$$\mu = -B\kappa - Wz. \tag{10}$$

For nonzero surface tension we can set $U_c = 4\sigma(K_1^{-1}+K_2^{-1})^{-1}l^{-2}$ so that $B = 1$. This leaves a single physical parameter, the Bond number $W = (\rho_1-\rho_2)gl^2/(2\sigma)$, which gives the ratio of buoyancy to surface tension forces. However, for convenience we will leave both B and W as parameters in the subsequent analysis.

3. Fundamental forms and isothermal parameterization

We summarize the surface differential geometry required for our method, and refer the reader to [27] for more details. Given a parameterized surface $\mathbf{X}(\alpha,\beta)$, we define the components of the first fundamental form to be

$$E = \mathbf{X}_\alpha\cdot\mathbf{X}_\alpha, \qquad F = \mathbf{X}_\alpha\cdot\mathbf{X}_\beta, \qquad G = \mathbf{X}_\beta\cdot\mathbf{X}_\beta, \tag{11}$$

where subscripts denote differentiation. The components of the second fundamental form are L, M, and N, where

$$L = \mathbf{X}_{\alpha\alpha}\cdot\hat{\mathbf{n}} = -\mathbf{X}_\alpha\cdot\hat{\mathbf{n}}_\alpha, \tag{12}$$
$$M = \mathbf{X}_{\alpha\beta}\cdot\hat{\mathbf{n}} = -\mathbf{X}_\alpha\cdot\hat{\mathbf{n}}_\beta = -\mathbf{X}_\beta\cdot\hat{\mathbf{n}}_\alpha, \tag{13}$$
$$N = \mathbf{X}_{\beta\beta}\cdot\hat{\mathbf{n}} = -\mathbf{X}_\beta\cdot\hat{\mathbf{n}}_\beta. \tag{14}$$

We will study surfaces which have a generalized isothermal parameterization in which

$$G(\alpha,\beta,t) = \lambda(t)E(\alpha,\beta,t) \quad \text{and} \quad F(\alpha,\beta,t) = 0. \tag{15}$$

Our approach to maintaining such a parameterization is to start with a surface which has a generalized isothermal parameterization, and to maintain it by appropriate choice of artificial tangential velocities.

In [8], [9], [10], Ambrose and Masmoudi used an isothermal parameterization (in which $\lambda(t) = 1$) to study the well-posedness of vortex sheets with surface tension, Darcy flow, and water waves. Furthermore, Ambrose and Siegel [16] used an isothermal parameterization to remove the stiffness from a model problem for Darcy flow in which the Birkhoff-Rott integral is replaced by an approximate (nonlocal) term that is easier to evaluate. These applications have required that the surface is parameterizable using a global isothermal parameterization. While this is sometimes the case, the above generalized isothermal parameterization allows for a larger number of surfaces to be treated. In particular it allows for doubly-periodic interfaces in which the average arclength in the α and β directions differ, that is, $\int_0^1 \int_0^1 |\mathbf{X}_\alpha| \, d\alpha d\beta \neq \int_0^1 \int_0^1 |\mathbf{X}_\beta| \, d\alpha d\beta$.

The mean curvature of the surface is an important geometric quantity. In general, the mean curvature is

$$\kappa = \frac{EN + GL - 2FM}{2(EG - F^2)}. \tag{16}$$

With our parameterization, this reduces to

$$\kappa = \frac{\lambda L + N}{2\lambda E}. \tag{17}$$

3.1. Geometric identities

We use the definitions of the interface tangents and normal, the second fundamental form, and the isothermal parameterization to derive the following identities:

$$\hat{\mathbf{t}}^1_\alpha \cdot \hat{\mathbf{n}} = -\hat{\mathbf{n}}_\alpha \cdot \hat{\mathbf{t}}^1 = \frac{L}{\sqrt{E}},$$

$$\hat{\mathbf{t}}^1_\beta \cdot \hat{\mathbf{n}} = -\hat{\mathbf{n}}_\beta \cdot \hat{\mathbf{t}}^1 = \frac{M}{\sqrt{E}}, \qquad \hat{\mathbf{t}}^2_\alpha \cdot \hat{\mathbf{n}} = -\hat{\mathbf{n}}_\alpha \cdot \hat{\mathbf{t}}^2 = \frac{M}{\sqrt{\lambda E}},$$

$$\hat{\mathbf{t}}^2_\beta \cdot \hat{\mathbf{n}} = -\hat{\mathbf{n}}_\beta \cdot \hat{\mathbf{t}}^2 = \frac{N}{\sqrt{\lambda E}}.$$

Similarly,

$$\hat{\mathbf{t}}^1_\alpha \cdot \hat{\mathbf{t}}^2 = -\hat{\mathbf{t}}^2_\alpha \cdot \hat{\mathbf{t}}^1 = -\frac{E_\beta}{2\sqrt{\lambda} E},$$

$$\hat{\mathbf{t}}^1_\beta \cdot \hat{\mathbf{t}}^2 = -\hat{\mathbf{t}}^2_\beta \cdot \hat{\mathbf{t}}^1 = \frac{\sqrt{\lambda} E_\alpha}{2E},$$

where in the first identity we used $F_\alpha = 0$ to write $\mathbf{X}_{\alpha\alpha} \cdot \mathbf{X}_\beta = -\mathbf{X}_\alpha \cdot \mathbf{X}_{\beta\alpha} = -E_\beta/2$, and in the second identity we used $\mathbf{X}_{\alpha\beta} \cdot \mathbf{X}_\beta = G_\alpha/2$.

These identities are used to represent the components for the first derivatives of the unit tangent and normal vectors as,

$$\hat{\mathbf{t}}^1_\alpha = \frac{\mathbf{X}_{\alpha\alpha}}{\sqrt{E}} - \frac{E_\alpha}{2E}\hat{\mathbf{t}}^1 = -\frac{E_\beta}{2\sqrt{\lambda} E}\hat{\mathbf{t}}^2 + \frac{L}{\sqrt{E}}\hat{\mathbf{n}},$$

$$\hat{\mathbf{t}}^1_\beta = \frac{\mathbf{X}_{\alpha\beta}}{\sqrt{E}} - \frac{E_\beta}{2E}\hat{\mathbf{t}}^1 = \frac{\sqrt{\lambda}E_\alpha}{2E}\hat{\mathbf{t}}^2 + \frac{M}{\sqrt{E}}\hat{\mathbf{n}},$$
$$\hat{\mathbf{t}}^2_\alpha = \frac{\mathbf{X}_{\alpha\beta}}{\sqrt{\lambda E}} - \frac{E_\alpha}{2E}\hat{\mathbf{t}}^2 = \frac{E_\beta}{2\sqrt{\lambda}E}\hat{\mathbf{t}}^1 + \frac{M}{\sqrt{\lambda E}}\hat{\mathbf{n}},$$
$$\hat{\mathbf{t}}^2_\beta = \frac{\mathbf{X}_{\beta\beta}}{\sqrt{\lambda E}} - \frac{E_\beta}{2E}\hat{\mathbf{t}}^2 = -\frac{\sqrt{\lambda}E_\alpha}{2E}\hat{\mathbf{t}}^1 + \frac{N}{\sqrt{\lambda E}}\hat{\mathbf{n}},$$
$$\hat{\mathbf{n}}_\alpha = -\frac{L}{\sqrt{E}}\hat{\mathbf{t}}^1 - \frac{M}{\sqrt{\lambda E}}\hat{\mathbf{t}}^2,$$
$$\hat{\mathbf{n}}_\beta = -\frac{M}{\sqrt{E}}\hat{\mathbf{t}}^1 - \frac{N}{\sqrt{\lambda E}}\hat{\mathbf{t}}^2,$$

which will be used in the derivation of equations for the tangential interface velocity in the next section.

3.2. Tangential velocities

The tangential velocities V_1 and V_2 are chosen to enforce the generalized isothermal parameterization. In this section an elliptic system of equations for the tangential velocities is derived from evolution equations for the first fundamental forms.

We start by differentiating the evolution equation (7) and using the geometric identities to derive the relations

$$\mathbf{X}_{\alpha t} = \left(V_{1\alpha} + \frac{V_2 E_\beta}{2\sqrt{\lambda}E} - \frac{UL}{\sqrt{E}}\right)\hat{\mathbf{t}}^1 + \left(V_{2\alpha} - \frac{V_1 E_\beta}{2\sqrt{\lambda}E} - \frac{UM}{\sqrt{\lambda E}}\right)\hat{\mathbf{t}}^2 + \left(U_\alpha + \frac{V_1 L}{\sqrt{E}} + \frac{V_2 M}{\sqrt{\lambda E}}\right)\hat{\mathbf{n}}, \quad (18)$$

$$\mathbf{X}_{\beta t} = \left(V_{1\beta} - \frac{V_2 E_\alpha}{2E} - \frac{UM}{\sqrt{E}}\right)\hat{\mathbf{t}}^1 + \left(V_{2\beta} + \frac{V_1 E_\alpha}{2E} - \frac{UN}{\sqrt{\lambda E}}\right)\hat{\mathbf{t}}^2 + \left(U_\beta + \frac{V_1 M}{\sqrt{E}} + \frac{V_2 N}{\sqrt{\lambda E}}\right)\hat{\mathbf{n}}. \quad (19)$$

We also require a formula for $\hat{\mathbf{n}}_t = \hat{\mathbf{t}}^1_t \times \hat{\mathbf{t}}^2 + \hat{\mathbf{t}}^1 \times \hat{\mathbf{t}}^2_t$. Time differentiation of the tangent vectors is obtained using the above equations for $\mathbf{X}_{\alpha t}$ and $\mathbf{X}_{\beta t}$ and the geometric relations, with the result

$$\hat{\mathbf{n}}_t = -\left(\frac{U_\alpha}{\sqrt{E}} + \frac{V_1 L}{E} + \frac{V_2 M}{\sqrt{\lambda}E}\right)\hat{\mathbf{t}}^1 - \left(\frac{U_\beta}{\sqrt{\lambda E}} + \frac{V_1 M}{\sqrt{\lambda}E} + \frac{V_2 N}{\lambda E}\right)\hat{\mathbf{t}}^2. \quad (20)$$

We note that since $\hat{\mathbf{n}}$ is a unit vector, $\hat{\mathbf{n}}_t \cdot \hat{\mathbf{n}} = 0$ so there is no normal component to $\hat{\mathbf{n}}_t$, which simplifies the algebra.

The normal interface velocity U is not arbitrary, but is instead determined by the kinematic condition. The tangential velocities, however, can be chosen however we like, as they

relate only to the parameterization of the free surface. We choose these tangential velocities to maintain the generalized isothermal parameterization.

Evolution equations for E, F, and G are derived using the relations (18) and (19), with the result

$$E_t = 2\sqrt{E}(\mathbf{X}_{\alpha t} \cdot \hat{\mathbf{t}}^1) = 2\sqrt{E}V_{1\alpha} + \frac{V_2 E_\beta}{\sqrt{\lambda E}} - 2UL,$$

$$G_t = 2\sqrt{G}(\mathbf{X}_{\beta t} \cdot \hat{\mathbf{t}}^2) = 2\sqrt{\lambda E}V_{2\beta} + \frac{V_1 \lambda E_\alpha}{\sqrt{E}} - 2UN,$$

$$F_t = \sqrt{E}(\sqrt{\lambda}V_{2\alpha} + V_{1\beta}) - \frac{V_1 E_\beta + \sqrt{\lambda}V_2 E_\alpha}{2\sqrt{E}} - 2UM. \tag{21}$$

Given these evolution equations for the first fundamental coefficients, it is possible to find both the evolution equation for λ and the elliptic system which V_1 and V_2 must satisfy. From the definition of λ, we have

$$\lambda(t) = \frac{\int_0^1 \int_0^1 G(\alpha, \beta, t)\ d\alpha d\beta}{\int_0^1 \int_0^1 E(\alpha, \beta, t)\ d\alpha d\beta},$$

and we could differentiate this to find λ_t, although a simpler derivation is given below.

We now find a pair of elliptic equations for the tangential velocities. If we start with a surface with a generalized isothermal parameterization, and choose the tangential velocities so that they satisfy these elliptic equations, then the parameterization will be maintained at positive times. The elliptic equations can be derived from the requirement $(\lambda E - G)_t = 0$ and $F_t = 0$. In practice, discretization error causes the parameterization to drift slightly away from isothermal, and we find that adding a relaxation term $-\theta(\lambda E - G)$ to the right hand side of the first equation and $-\theta F$ to the second (with $\theta \geq 0$) prevents or minimizes such drift. Using the formulas for E_t, F_t, and G_t, we then have

$$\left(\frac{V_1}{\sqrt{E}}\right)_\alpha - \left(\frac{V_2}{\sqrt{\lambda E}}\right)_\beta = \frac{-\lambda_t E + 2U(\lambda L - N) - \theta(\lambda E - G)}{2\lambda E} \equiv g_1 \tag{22}$$

$$\left(\frac{V_2}{\sqrt{E}}\right)_\alpha + \left(\frac{V_1}{\sqrt{\lambda E}}\right)_\beta = \frac{2UM - \theta F}{\sqrt{\lambda}E} \equiv g_2 \tag{23}$$

Notice that the only unknowns in this system are V_1 and V_2, as the rest of the quantities are determined from $\mathbf{X}(\vec{\alpha}, t)$, which is being evolved. Thus, at each time step, the solution of this system yields the tangential velocities which maintain the chosen parameterization. We remark again that this method of maintaining an isothermal parameterization has been implemented in [16], in the case in which $\lambda = 1$ and $\theta = 0$ and thus $\lambda_t = 0$.

An equation for λ_t is derived by integrating (22) over $[0, 1] \times [0, 1]$. The integral of the left hand side is zero (due to periodicity) and that of the right hand side gives, after some algebra,

$$\lambda_t = 2\int_0^1 \int_0^1 \frac{U(\lambda L - N)}{E}\ d\alpha d\beta, \tag{24}$$

where we have assumed $\lambda E - G = 0$.

4. The small-scale decomposition

In this section the normal component U of the Birkhoff-Rott integral is decomposed as a sum of nonlocal operators

$$U(\mathbf{X}) = U_s(\mathbf{X}) + J_1(\mathbf{X}), \tag{25}$$

in which $U_s(\mathbf{X})$ contains the high order terms that are dominant at small scales, and the remainder term $J_1(\mathbf{X})$ is lower order. The tangential velocities $V_1,\ V_2$ are similarly decomposed. We choose $U_s(\mathbf{X})$ so that its symbol or Fourier transform is known analytically and has a simple form, which is a key to our design of an efficient implicit method.

Assume $\mathbf{X}(\vec{\alpha})$ is sufficiently regular and that $\mathbf{X}(\vec{\alpha})-\mathbf{X}(\vec{\alpha'}) \neq 0$ for $\vec{\alpha} \neq \vec{\alpha'}$. Taylor expand the Green's function in (5) and simplify using the generalized isothermal coordinates (15) to give

$$\frac{\mathbf{X}(\alpha,\beta)-\mathbf{X}(\alpha',\beta')}{|\mathbf{X}(\alpha,\beta)-\mathbf{X}(\alpha',\beta')|^3} \approx \frac{\mathbf{X}_\alpha(\alpha',\beta')(\alpha-\alpha')+\mathbf{X}_\beta(\alpha',\beta')(\beta-\beta')}{[E(\alpha',\beta')]^{\frac{3}{2}}\left[(\alpha-\alpha')^2+\lambda(t)\,(\beta-\beta')^2\right]^{\frac{3}{2}}} \tag{26}$$

at leading order for $|\vec{\alpha}-\vec{\alpha'}| << 1$. Add and subtract (26) to the Birkhoff-Rott integral (5) to obtain

$$\begin{aligned}\mathbf{V}(\mathbf{X}) = \frac{1}{4\pi}\mathrm{PV}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \quad \mathbf{j}' \quad \times \Bigg\{ & \frac{\mathbf{X}'_\alpha(\alpha-\alpha')+\mathbf{X}'_\beta(\beta-\beta')}{E'^{\frac{3}{2}}\left[(\alpha-\alpha')^2+\lambda(t)\,(\beta-\beta')^2\right]^{\frac{3}{2}}} + \frac{\mathbf{X}-\mathbf{X}'}{|\mathbf{X}-\mathbf{X}'|^3} \\ - \quad & \frac{\mathbf{X}'_\alpha(\alpha-\alpha')+\mathbf{X}'_\beta(\beta-\beta')}{E'^{\frac{3}{2}}\left[(\alpha-\alpha')^2+\lambda(t)\,(\beta-\beta')^2\right]^{\frac{3}{2}}} \Bigg\}\, d\alpha' d\beta',\end{aligned} \tag{27}$$

where we recall that primed functions such as $\mathbf{X}'$ are evaluated at (α',β') and unprimed functions are evaluated at (α,β).

We now introduce the generalized Riesz transforms

$$H_1 f(\alpha,\beta) = \frac{1}{2\pi}\mathrm{PV}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{f(\alpha',\beta')(\alpha-\alpha')}{\left[(\alpha-\alpha')^2+\lambda(t)\,(\beta-\beta')^2\right]^{\frac{3}{2}}}\, d\alpha' d\beta', \tag{28}$$

$$H_2 f(\alpha,\beta) = \frac{1}{2\pi}\mathrm{PV}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{f(\alpha',\beta')(\beta-\beta')}{\left[(\alpha-\alpha')^2+\lambda(t)\,(\beta-\beta')^2\right]^{\frac{3}{2}}}\, d\alpha' d\beta'. \tag{29}$$

and represent equation (27) in terms of these transforms as

$$\mathbf{V}(\mathbf{X}) = H_1\left(\frac{\mathbf{j}\times\mathbf{X}_\alpha}{2E^{\frac{3}{2}}}\right) + H_2\left(\frac{\mathbf{j}\times\mathbf{X}_\beta}{2E^{\frac{3}{2}}}\right) + \mathbf{J}(\alpha,\beta,t). \tag{30}$$

Here $\mathbf{J}$ denotes the principal value integral

$$\mathbf{J}(\alpha,\beta,t) = \frac{1}{4\pi}\mathrm{PV}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{j}' \times \left\{ \frac{\mathbf{X}-\mathbf{X}'}{|\mathbf{X}-\mathbf{X}'|^3} - \frac{\mathbf{X}'_\alpha(\alpha-\alpha')+\mathbf{X}'_\beta(\beta-\beta')}{E'^{\frac{3}{2}}\left[(\alpha-\alpha')^2+\lambda(t)\,(\beta-\beta')^2\right]^{\frac{3}{2}}} \right\}\, d\alpha' d\beta'. \tag{31}$$

The Fourier transforms or symbols of the generalized Riesz transforms are [18]

$$\widehat{H}_1 f(\mathbf{k}) = -i\frac{k_1}{k_\lambda}\hat{f}_{\mathbf{k}}, \quad \widehat{H}_2 f(\mathbf{k}) = -i\frac{k_2}{\lambda k_\lambda}\hat{f}_{\mathbf{k}}, \tag{32}$$

for vector wavenumber $(k_1, k_2) \in \mathbb{Z}^2$, where $k_\lambda = \sqrt{\lambda k_1^2 + k_2^2}$ and we set $\widehat{H}_1 f(\mathbf{0}) = \widehat{H}_2 f(\mathbf{0}) = 0$. The Fourier coefficients $\hat{f}(\mathbf{k})$ are defined by

$$\hat{f}(\mathbf{k}) = \int_0^1 \int_0^1 e^{-2\pi i(k_1\alpha + k_2\beta)} f(\alpha, \beta)\, d\alpha d\beta. \tag{33}$$

The Riesz transform of a vector $\mathbf{f} = (f_1, f_2)$ is the vector obtained by transforming each component of $\mathbf{f}$, that is, $H_i(\mathbf{f}) = (H_i(f_1), H_i(f_2))$, for $i = 1,\ 2$.

The dominant terms at high wavenumber or small scales in equation (30) are the two Riesz transforms, which act like three derivatives of $\mathbf{X}$. This is given a precise meaning in the case of isothermal coordinates (i.e., $\lambda(t) = 1$) in [8], where it is shown that under certain conditions $\mathbf{J}$ is smoother or lower order than the Riesz transform terms by degree -1. Roughly speaking, this means that $\mathbf{J}$ has one derivative less than the H_1 and H_2 terms. The theory in [8] can be easily extended to the case where α and β are generalized isothermal coordinates.

We continue to simplify the Riesz transforms in (30). Introduce the notation $f \cong g$ to indicate that f and g have the same order or smoothness, that is, the difference between f and g is smoother than either of these functions. We also need that, at small scales, smooth functions can be passed through Riesz transforms,

$$H_i[fg] = gH_i[f] + \mathcal{S}[f] \tag{34}$$

where the commutator $\mathcal{S}$ is a smoothing operator on f. This is proven in [8] for isothermal coordinates, but the result is easily extended to generalized isothermal coordinates. Calculate the cross product in (30) using the relation $\mathbf{X}_\alpha \times \mathbf{X}_\beta = \mathbf{n}|\mathbf{X}_\alpha \times \mathbf{X}_\beta|$, pass $\mathbf{n}$ through the Riesz tranforms (incurring a commutator) and take the inner product with $\mathbf{n}$ to obtain

$$U \cong -\frac{\sqrt{\lambda}}{2}\left\{H_1\left(\frac{\mu_\alpha}{E^{\frac{1}{2}}}\right) + H_2\left(\frac{\mu_\beta}{E^{\frac{1}{2}}}\right)\right\} \tag{35}$$

$$\equiv U_s, \tag{36}$$

where we have also used $|\mathbf{X}_\alpha \times \mathbf{X}_\beta| = \sqrt{\lambda}E$. Putting back the lower order terms we have

$$U = U_s + J_1, \tag{37}$$

where

$$J_1 = \mathbf{V}\cdot\mathbf{n} + \frac{\sqrt{\lambda}}{2}\left\{H_1\left(\frac{\mu_\alpha}{E^{\frac{1}{2}}}\right) + H_2\left(\frac{\mu_\beta}{E^{\frac{1}{2}}}\right)\right\} \tag{38}$$

is lower order than the Riesz transform terms in (35).

We also need a small-scale decomposition of the tangential velocities. Define V_{1s}, V_{2s} to be the solution of (22),(23) for $U = U_s$ and $\theta = 0$, that is,

$$\left(\frac{V_{1s}}{\sqrt{E}}\right)_\alpha - \left(\frac{V_{2s}}{\sqrt{\lambda E}}\right)_\beta = \frac{-\lambda_t E + 2U_s\left(\lambda L - N\right)}{2\lambda E} \tag{39}$$

$$\left(\frac{V_{2s}}{\sqrt{E}}\right)_\alpha + \left(\frac{V_{1s}}{\sqrt{\lambda E}}\right)_\beta = \frac{2U_s M}{\sqrt{\lambda} E}. \tag{40}$$

If we write

$$V_1 = V_{1s} + (V_1 - V_{1s}) \tag{41}$$

$$V_2 = V_{2s} + (V_2 - V_{2s}), \tag{42}$$

then V_{1s}, V_{2s} contain the high order terms of V_1, V_2 that are dominant at small scales, and $V_1 - V_{1s}$, $V_2 - V_{2s}$ are lower order. This follows from $U \cong U_s$ being the highest order term of the right hand side of (22), (23). We evaluate the leading order or subscript-s terms using the isothermal curvature (17) and use the full curvature (16) elsewhere (e.g., in $U = \mathbf{V} \cdot \mathbf{n}$). This means that the exact equations are solved, even if the parameterization is not precisely isothermal due to discretization and time-stepping errors.

We now rewrite the evolution equation (7) as

$$\mathbf{X}_t = U_s\hat{\mathbf{n}} + V_{1s}\hat{\mathbf{t}}^{\mathbf{1}} + V_{2s}\hat{\mathbf{t}}^{\mathbf{2}} + (U - U_s)\hat{\mathbf{n}} + (V_1 - V_{1s})\hat{\mathbf{t}}^{\mathbf{1}} + (V_2 - V_{2s})\hat{\mathbf{t}}^{\mathbf{2}}. \tag{43}$$

The first three terms give the leading order behavior at small scales, and the last three terms are smoother or lower order. Equation (43) is a quasilinear system for the highest-derivative terms, rather than semilinear as in the HLS case, but this is dealt with in §5. There, we propose a semi-implicit scheme that discretizes high order terms implicitly and lower order terms explicitly. This gives a linear system for $\mathbf{X}$ at time $n+1$ with nonlocal implicit terms that can be evaluated fast using the FFT.

5. Semi-implicit method

We describe the method with a first order time discretization but with the equations exact in space. The improvement to second order in time is discussed in §5.3. Assume that the quantities $\mathbf{X}^n$, U_s^n, and U^n are known at time n as functions of the isothermal parameterization. Tangential velocities V_1^n, V_2^n are determined from (22), (23) with the right-hand side evaluated at time n. Other quantities such as the first and second fundamental forms, unit tangents and normal, and λ can be computed at time n from the current interface position $\mathbf{X}^n$.

We use (43) and write a scheme that is implicit in the leading order quantities U_s, V_{1s} and V_{2s} and explicit in the remainder terms $U - U_s$, etc. Define from (10), (17), (36)

$$U_s^{n+1} = -\frac{\sqrt{\lambda^n}}{2}\left\{H_1\left(\frac{\mu_\alpha^{n+1}}{E^{n\frac{1}{2}}}\right) + H_2\left(\frac{\mu_\beta^{n+1}}{E^{n\frac{1}{2}}}\right)\right\}, \tag{44}$$

where

$$\mu^{n+1} = -B\left(\frac{\lambda^n L^{n+1} + N^{n+1}}{2\lambda^n E^n}\right) - Wz^{n+1}, \tag{45}$$

with

$$L^{n+1} = \mathbf{X}^{n+1}_{\alpha\alpha} \cdot \hat{\mathbf{n}}^n \qquad N^{n+1} = \mathbf{X}^{n+1}_{\beta\beta} \cdot \hat{\mathbf{n}}^n. \tag{46}$$

This is just the update of the small-scale velocity U_s to U_s^{n+1} with the high order components evaluated at time $n+1$ and the lower order ones at time n. We find a slight performance advantage treating Wz implicitly, even though it is lower order. Similarly define from (39), (40)

$$\left(\frac{V_{1s}^{n+1}}{\sqrt{E^n}}\right)_\alpha - \left(\frac{V_{2s}^{n+1}}{\sqrt{\lambda^n E^n}}\right)_\beta = \frac{-\lambda_t^n E^n + 2U_s^{n+1}\left(\lambda L^n - N^n\right)}{2\lambda E^n} \tag{47}$$

$$\left(\frac{V_{2s}^{n+1}}{\sqrt{E^n}}\right)_\alpha + \left(\frac{V_{1s}^{n+1}}{\sqrt{\lambda^n E^n}}\right)_\beta = \frac{2U_s^{n+1} M^n}{\sqrt{\lambda^n} E^n}. \tag{48}$$

We then discretize (43) as

$$\begin{aligned}\mathbf{X}^{n+1} - \Delta t(U_s^{n+1}\hat{\mathbf{n}}^n + V_{1s}^{n+1}\hat{\mathbf{t}}^{\mathbf{1}n} + V_{2s}^{n+1}\hat{\mathbf{t}}^{\mathbf{2}n}) = \mathbf{X}^n + \Delta t\,[(U^n - U_s^n)\hat{\mathbf{n}}^n \\ + (V_1^n - V_{1s}^n)\hat{\mathbf{t}}^{\mathbf{1}n} + (V_2^n - V_{2s}^n)\hat{\mathbf{t}}^{\mathbf{2}n}]\,.\end{aligned} \tag{49}$$

This equation is of the form

$$\mathcal{L}[\mathbf{X}^n](\mathbf{X}^{n+1}) = \mathbf{f}(\mathbf{X}^n) \tag{50}$$

where $\mathcal{L}$ is a linear operator on $\mathbf{X}^{n+1}$ with coefficients that depend on $\mathbf{X}^n$. This linear system is inverted with preconditioned GMRES.

An advantage of the small-scale decomposition is that there is a fast $O(N \log N)$ computation of the implicit (nonlocal) velocities in (49) by the FFT alone. In contrast, a semi-implicit discretization of the original Birkhoff-Rott integral in (5) in which $(\mathbf{X} - \mathbf{X}')/|\mathbf{X} - \mathbf{X}'|^3$ is discretized at time n and the vortex sheet density $\mathbf{j}$ at time $n+1$ requires a fast multipole or related method (see §6) for calculation of the velocity at each iteration. These methods are associated with large constant prefactors and are considerably slower than computations based on the FFT alone for currently feasible values of N, as considered in this paper.

We note that the GMRES inversion of (49) can be entirely avoided by 'freezing' lower order coefficients such as E^n, which gives a constant coefficient linear system that diagonalizes in Fourier space. This approach was used in our previous paper [16], but is avoided here since we believe it can introduce additional smoothing.

5.1. Space discretization

We use a Fourier basis to represent the interface as

$$\mathbf{X}(\vec{\alpha}, t) = 2\pi(\alpha, \sqrt{\lambda}\beta, 0)^T + \sum_{k_1=-n_p/2}^{n_p/2-1} \sum_{k_2=-n_p/2}^{n_p/2-1} \widehat{\mathbf{X}}_{\mathbf{k}} e^{i\mathbf{k}\cdot\vec{\alpha}}, \tag{51}$$

where the first term denotes a planar interface and the sum represents periodic deviations from the planar surface. Numerical solution of the initial value problem is based on a uniform discretization in the (α, β) plane, e.g., $\alpha_j = (j-1)/n_p$ for $j = 1, \ldots, n_p$ with a similar discretization of β. Our method is spectrally accurate with the exception of the quadrature of the Birkhoff-Rott integral. Spatial derivatives are calculated via wavenumber multiplication of Fourier coefficients, and the Riesz transforms $H_1 f$, $H_2 f$ in (35) are computed from the Fourier coefficients of f using (32) (an exception is discussed in §6.4). Geometric quantities such as the interface tangents, normal, and the first and second fundamental forms are computed by Fourier-based differentiation of $\mathbf{X}$. Unless otherwise noted, a spectral filter [6] is applied to prevent amplification of round-off error when taking derivatives. The filter sets Fourier amplitudes that are below a given tolerance (here 10^{-14}) to zero.

Tangential velocities are computed at time n using the Fourier transform. For example, define $\overline{V}_1^n = V_1^n/\sqrt{E^n}$ and $\overline{V}_2^n = V_2^n/\sqrt{E^n}$ and take the Fourier transform of (22), (23) to obtain

$$\begin{pmatrix} \widehat{\overline{V}}_{1\mathbf{k}}^n \\ \widehat{\overline{V}}_{2\mathbf{k}}^n \end{pmatrix} = \frac{\lambda^n}{2\pi i k_\lambda^2} \begin{pmatrix} k_1 & \frac{k_2}{\sqrt{\lambda^n}} \\ -\frac{k_2}{\sqrt{\lambda^n}} & k_1 \end{pmatrix} \begin{pmatrix} \widehat{g}_{1\mathbf{k}}^n \\ \widehat{g}_{2\mathbf{k}}^n \end{pmatrix} \tag{52}$$

from which it is easy to obtain V_1^n and V_2^n at time n.

5.2. Preconditioner

A simple preconditioner for the operator $\mathcal{L}$ in (50) is its inverse when $\mathbf{X}^n$ is chosen to be a flat interface. This inverse operator can be constructed analytically. A related preconditioner is used in [25].

In the preconditioning step we let $\mathbf{X}^n = (2\pi\alpha, 2\pi\sqrt{\lambda}\beta, 0)$ be a plane and solve $\mathcal{L}_p(\mathbf{Z}) \equiv \mathcal{L}[\mathbf{X}^n](\mathbf{Z}) = \mathbf{f}$. The operator $\mathcal{L}_p(\mathbf{Z})$ is computed from the expressions (44)-(48) for U_s^{n+1}, V_{1s}^{n+1} and V_{2s}^{n+1} but with interface quantities at time n replaced by flat interface quantities. We write $\mathbf{Z}(\vec{\alpha}, t) = (x(\vec{\alpha}, t), y(\vec{\alpha}, t), z(\vec{\alpha}, t))$, and note that for planar $\mathbf{X}^n$ and $\mathbf{X}^{n+1} = \mathbf{Z}$,

$$\begin{aligned} &\mathbf{n}^n = (0, 0, 1), \quad E^n = 4\pi^2, \quad L^n = M^n = N^n = 0, \\ &V_{1s}^{n+1} = V_{2s}^{n+1} = 0, \quad L^{n+1} = z_{\alpha\alpha}, \quad N^{n+1} = z_{\beta\beta}, \end{aligned} \tag{53}$$

and λ is a constant flat interface quantity. It follows that

$$\begin{aligned} U_s^{n+1} = \frac{\sqrt{\lambda}}{4\pi^2} \Bigg\{ & H_1 \left(\frac{B(\lambda z_{\alpha\alpha\alpha} + z_{\beta\beta\alpha})}{8\pi^2\lambda} + W z_\alpha \right) \\ & + H_2 \left(\frac{B(\lambda z_{\alpha\alpha\beta} + z_{\beta\beta\beta})}{8\pi^2\lambda} + W z_\beta \right) \Bigg\}. \end{aligned} \tag{54}$$

Equation (50) is inverted by taking its Fourier transform

$$\widehat{\mathcal{L}_p(\mathbf{Z})}_{\mathbf{k}} = \widehat{\mathbf{Z}}_{\mathbf{k}} - \Delta t \widehat{U}_{s\mathbf{k}}^{n+1} \mathbf{n} \equiv \widehat{\mathbf{f}}_{\mathbf{k}}^n, \tag{55}$$

where

$$\widehat{U}_{s\mathbf{k}}^{n+1} = \left[-\tilde{B} k_\lambda^3 + \tilde{W} k_\lambda \right] \widehat{z}_{\mathbf{k}}, \tag{56}$$

with $\tilde{B} = B/(4\lambda^{3/2})$, $\tilde{W} = W/(2\lambda^{1/2})$, and we recall that $k_\lambda = \sqrt{\lambda k_1^2 + k_2^2}$. It follows that

$$\widehat{\mathbf{Z}}_{\mathbf{k}} = \left(\widehat{f}^n_{1\mathbf{k}}, \widehat{f}^n_{2\mathbf{k}}, \frac{\widehat{f}^n_{3\mathbf{k}}}{1 + \Delta t(\tilde{B}k_\lambda^3 - \tilde{W}k_\lambda)} \right)^T \tag{57}$$

is the solution of (55) in terms of the components of $\widehat{\mathbf{f}}^n_{\mathbf{k}}$. The cost of solving for $\mathbf{Z}$ is $O(N \log N)$ per GMRES iteration, where $N = n_p^2$ is the total number of surface grid points.

We note that the above spectral analysis shows that the eigenvalues $\mu_{\mathbf{k}}$ of the operator $\mathcal{L}$ for the plane are $\mu_{\mathbf{k}} = 1 + \Delta t(\tilde{B}k_\lambda^3 - \tilde{W}k_\lambda)$, which implies that the condition number is $O(\Delta t n_p^3)$. The condition number of the full operator in (50) is expected to have the same order.

5.3. Second order method

To gain second or higher order time accuracy we use the semi-implicit backward difference formulas developed in [28] and used in [20], [23]. The second order scheme for (43) is

$$\nu\mathbf{X}^{n+1} - \mathbf{X}^0 = \Delta t \left[2f(\mathbf{X}^n) - f(\mathbf{X}^{n-1}) + U_s^{n+1}\hat{\mathbf{n}}(\mathbf{X}^c) + V_{1s}^{n+1}\hat{\mathbf{t}}_1(\mathbf{X}^c) + V_{2s}^{n+1}\hat{\mathbf{t}}_2(\mathbf{X}^c) \right] \tag{58}$$

where $\nu = 3/2$ and

$$\mathbf{f}(\mathbf{X}^n) = (U^n - U_s^n)\hat{\mathbf{n}}^n + (V_1^n - V_{1s}^n)\hat{\mathbf{t}}^1(\mathbf{X}^{\mathbf{c}}) + (V_2^n - V_{2s}^n)\hat{\mathbf{t}}^2(\mathbf{X}^{\mathbf{c}}), \tag{59}$$

$$\mathbf{X}^0 = 2\mathbf{X}^n - \frac{1}{2}\mathbf{X}^{n-1}, \tag{60}$$

$$\mathbf{X}^c = 2\mathbf{X}^n - \mathbf{X}^{n-1}. \tag{61}$$

Here $\mathbf{X}^c$ is an interfacial position obtained by interpolation from previous time steps. Different definitions of ν, $\mathbf{X}^0$ and $\mathbf{X}^c$ can be made to achieve higher order accuracy (see [23]).

5.4. Initial data

In our method we need to construct initial data with an isothermal parameterization. The data is first specified in terms of an arbitrary nonisothermal parameterization, and then evolved to be isothermal by numerical integration of (7) with $U = 0$ and V_1, V_2 determined from (22), (23) with relaxation parameter $\theta > 0$. This evolution leaves the interface shape fixed, but moves points tangentially along the surface so that $|\lambda E - G|$ and $|F|$ decrease in time to a preset small tolerance.

When $U = 0$ the high order terms responsible for stiffness are absent, and we use an explicit time integration method to evolve (7) with second order temporal accuracy and spectral accuracy in space. The relative difference between our parameteriation and an exact isothermal one is measured by the quantities $D_1(t) = \max_{\alpha,\beta} \frac{|\lambda E - G|}{E}$ and $D_2(t) = \max_{\alpha,\beta} \frac{|F|}{|\mathbf{X}_\alpha||\mathbf{X}_\beta|}$. For example, when the interface shape

$$x(\alpha,\beta) = 2\pi\alpha, \quad y(\alpha,\beta) = 2\pi\beta, \quad z(\alpha,\beta) = A\cos(2\pi\alpha)\cos(2\pi\beta), \tag{62}$$

for which $\lambda = 1$ is evolved with $A = 0.5$, $N = 100^2$, $\Delta t = 10^{-3}$, and $\theta = 50$, the parameterization at $t = 1.0$ has D_1 and D_2 less than 10^{-12}. Thus, the parameterization is effectively isothermal. A different method for generating isothermal initial data is described in [16].

6. Surface integration

6.1. Ewald summation for a doubly-periodic interface

In a doubly-periodic domain the Birkhoff-Rott integral (5) can be written

$$\mathbf{V}(\mathbf{X}) = \frac{1}{4\pi}\mathrm{PV}\int_{B_c} (\mu_\alpha \mathbf{X}_\beta - \mu_\beta \mathbf{X}_\alpha)' \times \nabla_X G\left(\mathbf{X}, \mathbf{X}'\right)\ d\alpha' d\beta' + \mathbf{V}_\infty \hat{\mathbf{k}}, \tag{63}$$

where B_c is the fundamental or root cell, and G is a lattice sum of periodic extensions of the free space Green's function $\tilde{G}$,

$$G(\mathbf{X}, \mathbf{X}') = \sum_{\mathbf{m}\in\mathbb{Z}^2} \tilde{G}(\mathbf{X} - \mathbf{X}' - 2\tilde{\mathbf{m}}\pi), \tag{64}$$

with

$$\tilde{G}(\mathbf{X} - \mathbf{X}' - 2\tilde{\mathbf{m}}\pi) = \frac{1}{[(x - x' - 2m\pi)^2 + (y - y' - 2n\pi)^2 + (z - z')^2]^{1/2}}. \tag{65}$$

Here the index $\mathbf{m} = (m, n)$ sums over the doubly periodic images of the root cell B_c, and $\tilde{\mathbf{m}} = (m, n, 0)$. When the sum is written as in (64) it is strictly divergent. Beale [29] notes that a conditionally convergent sum for $\mathbf{X} \neq \mathbf{X}'$ is obtained when a reflection and constant are added to each term,

$$G(\mathbf{X}, \mathbf{X}') = {\sum_{\mathbf{m}\in\mathbb{Z}^2}}' \left(\frac{1}{2}\tilde{G}(\mathbf{X} - \mathbf{X}' - 2\tilde{\mathbf{m}}\pi) + \frac{1}{2}\tilde{G}(\mathbf{X} - \mathbf{X}' + 2\tilde{\mathbf{m}}\pi) - \frac{1}{2\pi|\mathbf{m}|} \right), \tag{66}$$

and that (66) gives the doubly-periodic fundamental solution to Laplace's equation. The prime on the sum indicates that the $1/|\mathbf{m}|$ term is omitted at $\mathbf{m} = 0$. There is some arbitrariness in the fundamental solution (66), since a linear function $G_L = az + b$ for constant $a,\ b$ can be added to G so that $G + G_L$ remains a doubly periodic fundamental solution to Laplace's equation. However, the additional term $\nabla_X G_L = a\widehat{\mathbf{k}}$ makes no contribution to the velocity in (63) , since the density $\mu_\alpha \mathbf{X}_\beta - \mu_\beta \mathbf{X}_\alpha$ has zero mean.

The Ewald summation technique converts the slowly convergent sum of algebraic functions (66) into a rapidly convergent sum of transcendental functions. A derivation is provided in Appendix A, with the result

$$\begin{aligned} G = {} & \frac{1}{2}{\sum_{\mathbf{m}\in\mathbb{Z}^2}}' \left(\frac{\mathrm{erfc}(\sqrt{s}\xi)}{\sqrt{s}} - \frac{\mathrm{erfc}(\pi\xi|\mathbf{m}|)}{\pi|\mathbf{m}|} \right) \\ & + \frac{1}{4\pi}\sum_{\mathbf{m}\in\mathbb{Z}^2} \left(R_{mn}(z - z')\cos m(x - x')\cos n(y - y') - \frac{2\mathrm{erfc}(\frac{|\mathbf{m}|}{\xi})}{|\mathbf{m}|} \right) + \frac{\xi}{\sqrt{\pi}}, \end{aligned} \tag{67}$$

where $s = [(x - x')/2 - m\pi]^2 + [(y - y')/2 - n\pi]^2 + [(z - z')/2]^2$ and

$$\begin{aligned} R_{mn}(z - z') = \frac{1}{\sqrt{m^2 + n^2}} \Bigg[& \quad e^{\sqrt{m^2+n^2}(z-z')}\mathrm{erfc}\left(\frac{\sqrt{m^2 + n^2}}{\xi} + \frac{z - z'}{2}\xi \right) \\ & + \quad e^{-\sqrt{m^2+n^2}(z-z')}\mathrm{erfc}\left(\frac{\sqrt{m^2 + n^2}}{\xi} - \frac{z - z'}{2}\xi \right) \Bigg]. \end{aligned} \tag{68}$$

Here ξ is a parameter that controls the 'decay rate' of terms in the above sum.

The sum (67) is pointwise convergent away from the singularity at $\mathbf{X} = \mathbf{X}'$. In particular we note that

$$R_{mn}(z - z' = 0) = \frac{2\text{erfc}\left(\frac{|\mathbf{m}|}{\xi}\right)}{|\mathbf{m}|} \tag{69}$$

and the infinite component at $\mathbf{m} = 0$ cancels out. The Ewald sum (67) is presented without the constant terms in [30]. An equivalent sum representing the electrostatic potential of a doubly-periodic array of point charges that are charge neutral is derived in [31]. The first sum in (67) represents local contributions to the velocity at $\mathbf{X}$; it is called the real space sum and we denote it by G^a (see Appendix A). The second sum represents far-field contributions to the velocity at $\mathbf{X}$ and is called the reciprocal sum, which we denote by G^b. Integrals over these sums will be called real space and reciprocal space integrals.

6.2. Velocity kernels

The kernels for the velocity integral in (5) are

$$\nabla G^a(\mathbf{X}, \mathbf{X}') = -\frac{1}{8}\sum_{\mathbf{m}\in\mathbb{Z}^2}\left[\frac{2\xi}{\sqrt{\pi}}\frac{e^{-\xi^2 s}}{s} + \frac{\text{erfc}(\sqrt{s}\xi)}{s^{3/2}}\right](x - x' - 2m\pi, y - y' - 2n\pi, z - z')^T \tag{70}$$

for the real space sum and

$$\frac{\partial G^b}{\partial x}(\mathbf{X}, \mathbf{X}') = -\frac{1}{4\pi}\sum_{\mathbf{m}\in\mathbb{Z}^2} m\ R_{mn}(z - z')\sin m(x - x')\cos n(y - y'), \tag{71}$$

$$\frac{\partial G^b}{\partial y}(\mathbf{X}, \mathbf{X}') = -\frac{1}{4\pi}\sum_{\mathbf{m}\in\mathbb{Z}^2} n\ R_{mn}(z - z')\cos m(x - x')\sin n(y - y'), \tag{72}$$

$$\frac{\partial G^b}{\partial z}(\mathbf{X}, \mathbf{X}') = \frac{1}{4\pi}\sum_{\mathbf{m}\in\mathbb{Z}^2} \frac{\partial R_{mn}}{\partial z}(z - z')\cos m(x - x')\cos n(y - y'), \tag{73}$$

for the reciprocal space sum, where

$$\frac{\partial R_{nm}}{\partial z}(z - z') = e^{k(z-z')}\text{erfc}\left(\frac{k}{\xi} + \frac{z - z'}{2}\xi\right) - e^{-k(z-z')}\text{erfc}\left(\frac{k}{\xi} - \frac{z - z'}{2}\xi\right),$$

and $k = \sqrt{m^2 + n^2}$. The superscript T in (70) denotes transpose. These kernels are computed by introducing a cutoff wavenumber k_{max} and summing for $k \le k_{max}$.

6.3. Discretization and fast computation of integrals

6.3.1. Reciprocal space integrals

Computation of the interface velocity requires evaluation of an integral over the reciprocal space sum of the form

$$\mathbf{I}_1(\mathbf{X}) = \int_{B_c} f(\alpha', \beta')\nabla G^b(\mathbf{X}, \mathbf{X}')d\alpha' d\beta'. \tag{74}$$

Fast computation of the integral is hampered by the lack of separability of z and z' in the $R_{mn}(z-z')$ terms. We derive a simple, spectrally accurate and fast $O(N)$ method to evaluate this integral. Another fast method for computing the reciprocal space sum in a related problem is given in [14].

Our method is described for the specific case of the Green's function $\partial G^b/\partial z$, with other integrals treated similarly. First expand the trigonometric functions in (73) to get

$$I_1^3(\mathbf{X}) = \frac{1}{4\pi}\sum_{\mathbf{m}\in\mathbb{Z}^2}\big[\cos(mx)\cos(ny)I_{cc}(z) + \cos(mx)\sin(ny)I_{cs}(z) + \sin(mx)\cos(ny)I_{sc}(z) + \sin(mx)\sin(ny)I_{ss}(z)\big], \tag{75}$$

where the superscript 3 denotes the third component of (74) and

$$\begin{aligned}
I_{cc}(z) &= \int_{B_c} f(\alpha',\beta')R'_{mn}(z-z')\cos(mx')\cos(ny')d\alpha'd\beta',\\
I_{cs}(z) &= \int_{B_c} f(\alpha',\beta')R'_{mn}(z-z')\cos(mx')\sin(ny')d\alpha'd\beta',\\
I_{sc}(z) &= \int_{B_c} f(\alpha',\beta')R'_{mn}(z-z')\sin(mx')\cos(ny')d\alpha'd\beta',\\
I_{ss}(z) &= \int_{B_c} f(\alpha',\beta')R'_{mn}(z-z')\sin(mx')\sin(ny')d\alpha'd\beta',
\end{aligned}$$

with $R'_{mn} = \partial R_{mn}/\partial z$. We discretize the root cell B_c into $N = n_p^2$ points so that the grid spacing is $h = \Delta\alpha = \Delta\beta = 1/n_p$. The integrands are C^∞ and periodic in α' and β' and are discretized using trapezoid rule quadrature, which is spectrally accurate for periodic functions. We thus obtain, for example,

$$\begin{aligned}
I_{cc}(z) &= \int_{B_c} f(\alpha',\beta')R'_{mn}(z-z')\cos(mx')\cos(ny')d\alpha'd\beta'\\
&\approx \sum_{i,j} f_{i,j}R'_{mn}(z-z_{i,j})\cos(mx_{i,j})\cos(ny_{i,j})h^2\\
&= e^{kz}a_{mn}(z) - e^{-kz}b_{mn}(z),
\end{aligned} \tag{76}$$

where for a function $g(\alpha,\beta)$ we define $g_{i,j} = g(\alpha_i,\beta_j)$, $\sum_{i,j} g_{i,j} = \sum_{i=1}^{n_p}\sum_{j=1}^{n_p} g_{i,j}$, and

$$\begin{aligned}
a_{mn}(z) &= \sum_{i,j} f_{i,j}e^{-kz_{i,j}}\operatorname{erfc}\left(\frac{k}{\xi} + \frac{z-z_{i,j}}{2}\xi\right)\cos(mx_{i,j})\cos(ny_{i,j})h^2,\\
b_{mn}(z) &= \sum_{i,j} f_{i,j}e^{kz_{i,j}}\operatorname{erfc}\left(\frac{k}{\xi} - \frac{z-z_{i,j}}{2}\xi\right)\cos(mx_{i,j})\cos(ny_{i,j})h^2.
\end{aligned}$$

For simplicity, we write these sums as

$$a_{mn}(z) = \sum_{i,j} \alpha_{mn}^{i,j} \text{erfc}\left(\frac{k}{\xi} + \frac{z - z_{i,j}}{2}\xi\right), \tag{77}$$

$$b_{mn}(z) = \sum_{i,j} \xi_{mn}^{i,j} \text{erfc}\left(\frac{k}{\xi} - \frac{z - z_{i,j}}{2}\xi\right), \tag{78}$$

where

$$\alpha_{mn}^{i,j} = f_{i,j} e^{-kz_{i,j}} \cos(mx_{i,j}) \cos(ny_{i,j}) h^2,$$
$$\beta_{mn}^{i,j} = f_{i,j} e^{kz_{i,j}} \cos(mx_{i,j}) \cos(ny_{i,j}) h^2.$$

Evaluation of (77)-(78) (for example) at all N points in the root cell B_c requires $O(k_{max}^2 N^2)$ operations, where k_{max} is the cutoff for $k = \sqrt{m^2 + n^2}$. The dependence on N^2 is due to these sums not being separable in z and z_{ij}. To compute the sums in $O(k_{max}^2 N)$, we apply the method of [32] for fast computation of weighted sums of erfc functions.

6.3.2. Fast summation of erfc functions

Evaluation of the coefficients (77) and (78) in the reciprocal sum involves computing weighted sums of erfc functions. Direct evaluation of this sum at M target points is $O(MN)$. In [32], an ϵ-exact approximation was derived that evaluates the sum in $O(M+N)$ operations. The method relies on using a separable Fourier series representation of $\text{erfc}(x-y)$ for $|x-y| \leq r$, and an asymptotic representation for $|x-y| > r$. By an appropriate choice of the cutoff r and the number of terms in the Fourier series the method can be made arbitrarily precise or '$\epsilon-$ exact', as discussed below. We summarize the method, omitting details of the derivation and instead refer the reader to [32].

Following the notation of [32], we write (77), for example, as

$$E(y) = \sum_{i=1}^{N} q_i \text{erfc}(y - x_i), \tag{79}$$

where $x_i = \xi z_i/2$ with z_i an ordering of $z_{i,j}$, and $y = k/\xi + \xi z/2$ is a target point. The complementary error function is computed using the series

$$\text{erfc}(x) = 1 - \frac{4}{\pi} \sum_{\substack{n=1 \\ n \text{ odd}}}^{2p-1} \frac{e^{-n^2\tau^2}}{n} \sin(2n\tau x) + \text{error}(x),$$

which converges rapidly for $|x| < r$ (where r depends on a prespecified accuracy ϵ). For $x \leq -r$ set $\text{erfc}(x) = 2$ and for $x > r$ approximate $\text{erfc}(x) = 0$. The sum in (79) is computed as

$$\hat{E}(y) = \sum_{(y-c_k)<-r} 2Q_k + \sum_{|y-c_k|\leq r} \left\{ Q_k - \frac{4}{\pi} \sum_{\substack{n=1 \\ n \text{ odd}}}^{2p-1} \left[A_n^k \sin(2n\tau(y - c_k)) - B_n^k \cos(2n\tau(y - c_k)) \right] \right\}, \tag{80}$$

where the interval $[-r, r]$ has been divided into k subintervals of length $2r_0$, c_k is the center of each subinterval, τ is a real number related to the sampling interval, and p is the truncation number. These parameters are determined so that

$$\max_{y_j} \frac{|\hat{E}(y_j) - E(y_j)|}{\sum_{i=1}^{N} |q_i|} \leq \epsilon \tag{81}$$

for a chosen accuracy ϵ. The quantities Q_k, A_n^k, B_n^k are defined as

$$\begin{aligned} Q_k &= \sum_{i=1}^{N_k} q_i, \\ A_n^k &= \frac{e^{-n^2\tau^2}}{n} \sum_{i=1}^{N_k} q_i \cos(2n\tau(x_i - c_k)), \\ B_n^k &= \frac{e^{-n^2\tau^2}}{n} \sum_{i=1}^{N_k} q_i \sin(2n\tau(x_i - c_k)), \end{aligned}$$

where N_k is the number of points in the cluster or interval with center c_k. These coefficients are not dependent on the evaluation point y, but rather on the cluster points and center, therefore, they are computed and stored to be re-used for each y. It is then $O((2p+1)N)$ work to compute Q_k, A_n^k, and B_n^k. In addition, evaluating $\hat{E}(y)$ at M points is $O((2pl + m)M)$, where l is the number of clusters for which $|y - c_k| \leq r$, and m is the number of clusters for which $(y - c_k) < -r$. This makes the total computational time $O((2p+1)N + (2pl+m)M)$.

6.3.3. Real space integrals

The real space integrals are of the form

$$I_2(\mathbf{X}) = \int_B f(\alpha', \beta') \nabla_i G^a(\mathbf{X}, \mathbf{X}') dx' dy',$$

where ∇G^a is defined in (70). Unlike the reciprocal space integrals, the real space integrals are singular. These are discretized using the simple and accurate method of Haroldsen and Meiron [15]. This method uses a trapezoid rule integration that omits the singular point to obtain a low order or $O(h)$ accurate approximation. The discretization error can be shown to be a power series in odd orders of h, and the approximation is improved to $O(h^{2p_d+1})$ by doing p_d Richardson extrapolations. These extrapolations are implemented using a coarser mesh, and are equivalent to incorporating weights $w_{i,j}$ in the original trapezoid rule integration

$$\mathbf{I}_2(\mathbf{X}) = {\sum_{i,j}}' w_{i,j} f_{i,j} \nabla G^a(\mathbf{X}, \mathbf{X}_{i,j}) h^2, \tag{82}$$

where the prime indicates that the singular point at $\mathbf{X} = \mathbf{X}_{i,j}$ is skipped. The method has the same operation count as simple trapezoid rule integration. For example, one Richardson

extrapolation gives $w_{i,j} = -2$ if $i,\ j$ are both even and $w_{i,j} = 2$ otherwise, and gives a third order or $O(h^3)$ accurate method. The results presented below use this third order method, although in principle it is straightforward to achieve higher order accuracy.

The Green's function G^a decays rapidly as $\mathbf{X}'$ moves away from $\mathbf{X}$, with the rate of decay depending on the parameter ξ. We impose a fixed cutoff r_c on the real space sum (82) that depends on a prespecified error tolerance, and is independent of N. For a given target point $\mathbf{X}$ in the root box, the sum (82) is computed only for those points $\mathbf{X}_{i,j}$ in the root and nearest neighbor boxes satisfying

$$\xi\sqrt{s} \le r_c. \tag{83}$$

A naive or straightforward evaluation of the real space sum (82) by direct summation has $O(N^2)$ complexity. We proceed to develop two methods for fast computation of this sum. The first method uses direct summation and a judicious scaling of the parameters with N. This has the benefit of simplicity but exhibits suboptimal $O(N^{3/2})$ complexity. The second method uses a treecode to speed up the computation of the real space sum. This is technically more difficult but has nearly optimal complexity $O(N \log N)$.

Method 1: *Direct summation.* For fixed ξ the number of interface grid points that satisfy the cutoff (83) is $O(N)$ for each target point $\mathbf{X} \in B_c$, and thus direct calculation of the real space sum at N targets is $O(N^2)$. This operation count can be reduced by taking $\xi = \xi(N)$ to be an increasing function of N, which by (83) is equivalent to decreasing the set of points $\mathbf{X}_{i,j}$ over which (82) is summed for each target $\mathbf{X}$. The complexity of direct summation is then minimized by equating the amount of work between the real and reciprocal space sums. Let $\xi \sim N^\gamma$, with γ to be determined. At a fixed target point $\mathbf{X}$, the number of grid points $\mathbf{X}_{ij}$ satisfying the cutoff (83) scales as $N^{1-2\gamma}$. Since there are $O(N)$ target points in the root box, the total operation count for the real space sum is $O(N^{2-2\gamma})$.

With the fast summation of erfc functions, the reciprocal sum (75) is evaluated in $O(k_{max}^2 N)$ operations, where k_{max} is the wavenumber cutoff. An error analysis [14] shows that for a fixed, prespecified accuracy, $k_{max} \sim \xi \sim N^\gamma$. (This is due to the factor of k/ξ in the argument of the erfc functions, which controls their rate of decay.) Therefore, the total operation count for the reciprocal space sum is $O(N^{2\gamma+1})$. Equating the amount of work between the two sums gives $\gamma = 1/4$, so that the total operation count of Method 1 is $O(N^{3/2})$.

Method 2: *Treecode-accelerated real space sum.* Duan and Krasny [13] used a treecode to speed up the computation of the real space part of the Ewald sum (with fixed ξ) from $O(N^2)$ to $O(N \log N)$ in triply-periodic electrostatic problems. When applied to compute (82), the treecode algorithm divides interface points into a nested set of clusters and approximates velocity interactions between a point and a distant cluster using Taylor's expansion of (70). Interaction of nearby points is computed using direct summation. The real space sums in the doubly- and triply-periodic problems are similar, and our implementation of the treecode closely follows [13] (see also [33]). The treecode algorithm is combined with the fast $O(k_{max}^2 N)$ reciprocal space computation for fixed ξ and k_{max} to give an overall $O(N \log N)$ method.

The treecode algorithm introduces two additional parameters, the order of the Taylor's

series expansion q_T, and a separation parameter ϵ_T. For the particle-cluster interaction, the Taylor's approximation is chosen over the direct sum when $\epsilon_T > R/D$, where R is the radius of the cluster and D is the distance from the target point to the cluster center. A decrease in ϵ_T improves the accuracy of the treecode, but slows the computations.

All erfc function evaluations in the real space sum are computed using the fast method of Mori [34], with parameters set to give relative error less than 10^{-15}.

6.4. Singularity reduction

Numerical tests show that the order of the singularity in (63) needs to be reduced to obtain a stable numerical method when using the Richardson-extrapolation-based quadrature of Haroldsen and Meiron. This is implemented by subtracting the leading order singular term before discretizing the Birkhoff-Rott integral, or equivalently by discretizing all integrals on the right-hand side of (49), including the U_s term, by the same (Haroldsen-Meiron) quadrature. Introduce the notation $\mathbf{V}_h^n$ to denote the space discretization of (63) using the trapezoidal rule with quadrature weights $w_{i,j}$, as in (82). The order of the singularity is reduced by discretizing the right-hand side $\mathcal{R}$ of (49) as

$$\mathcal{R}_h^n = \mathbf{X}^n + U_h^n - U_{sh}^n + \tilde{V}_1^n \hat{\mathbf{t}}^{\mathbf{1}n} + \tilde{V}_2^n \hat{\mathbf{t}}^{\mathbf{2}n}. \tag{84}$$

Here $U_h^n = \mathbf{V}_h^n \cdot \mathbf{n}^n$, U_{sh}^n are the trapezoidal-rule discretization of the Riesz transforms in (36) with the same quadrature weights, and $\tilde{V}_1^n$, $\tilde{V}_2^n$ are solutions to

$$\left(\frac{\tilde{V}_1^n}{\sqrt{E^n}}\right)_\alpha - \left(\frac{\tilde{V}_2^n}{\sqrt{\lambda^n E^n}}\right)_\beta = \frac{2\left(U_h^n - U_{sh}^n\right)\left(\lambda^n L^n - N^n\right) - \theta(\lambda^n E^n - G^n)}{2\lambda^n E^n} \tag{85}$$

$$\left(\frac{\tilde{V}_2^n}{\sqrt{E^n}}\right)_\alpha + \left(\frac{\tilde{V}_1^n}{\sqrt{\lambda^n E^n}}\right)_\beta = \frac{2\left(U_h^n - U_{sh}^n\right) M^n - \theta F^n}{\sqrt{\lambda^n} E^n}, \tag{86}$$

which is the difference in tangential velocities $V_1^n - V_{1s}^n$ and $V_2^n - V_{2s}^n$ computed using the reduced order integral $U_h^n - U_{sh}^n$. A similar singularity subtraction or reduction of order is shown to stabilize boundary integral methods for 3D water waves in [35], [18], and for the Rayleigh-Taylor instability in [30]. We find that the singularity reduction is not necessary for stability of the first order method without Richardson Extrapolation, i.e., with quadrature weights $w_{i,j} = 1$.

7. Numerical results

7.1. Quadrature

The third order surface integration method is validated by computing the Birkhoff-Rott integral (63) for a flat interface $\mathbf{X}(\alpha, \beta) = (2\pi\alpha, 2\pi\beta, 0)$ with $\lambda = 1$ and periodic density $\mathbf{j}(\alpha, \beta)$. The integral then reduces to a sum of Riesz transforms (28), (29), and we compare the computed value with an exact analytical expression obtained from the Fourier symbols (32). Tests (not shown) indicate that the quadrature converges at the expected $O(h^3)$ rate,

where $h = 1/n_p$ and $N = n_p^2$ is the number of surface discretization points. It is also verified that a direct discretization of (5) over a large number of periods is in good agreement with the Ewald summation method.

There is no exact solution for a nonflat interface, and we empirically compute the truncation error as $E_\infty(h) = ||\mathbf{V}_h - \mathbf{V}_{h/2}||_\infty / ||\mathbf{V}_{h/2}||_\infty$, the maximum relative velocity difference between uniform grid spacings h and $h/2$. Figures 1 through 3 give an example of the quadrature error for interface shape $\mathbf{X}(\vec{\alpha}) = (2\pi\alpha, 2\pi\beta, (1/2)\sin(2\pi\alpha))$ and density $\mathbf{j}(\vec{\alpha}) = (\sin(2\pi\alpha)\sin(2\pi\beta), \cos(2\pi\alpha)\cos(2\pi\beta), \sin(2\pi\alpha)\cos(2\pi\beta))$. The fast method of §6.3.1 is used to compute the reciprocal space integral, and the real space integral is calculated by the two methods presented in §6.3.3. The first is direct summation with parameters chosen as

$$\xi = N^{1/4}/4, \quad r_c = 2\pi, \quad \text{and} \quad k_c = 3\xi \tag{87}$$

to obtain $O(N^{3/2})$ complexity in the computation of the Birkhoff-Rott integral. The specific choice of constants in (87) (i.e., 1/4, 2π and 3) is guided by error estimates [14], and adjusted by trial and error so that the CPU times for real and reciprocal space sums are approximately equal and minimal for a given accuracy. The second method uses the treecode with fixed or N-independent Ewald parameters

$$\xi = 1.5, \quad r_c = 2\pi \quad \text{and} \quad k_c = 3\xi, \tag{88}$$

with order of the Taylor's series expansion $q_T = 10$ and separation parameter $\epsilon_T = 0.27$.

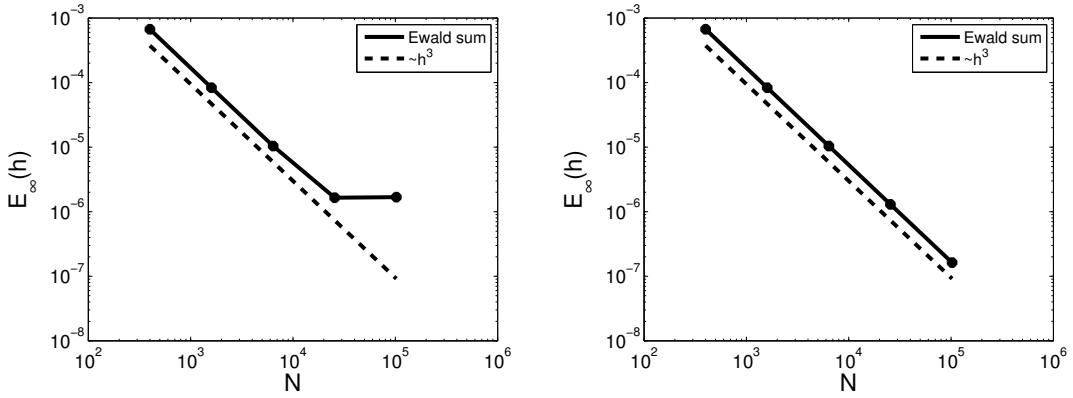

Figure 1: Log-log plot of truncation error $E_\infty(h = 1/\sqrt{N})$ versus N, for $\epsilon = 10^{-6}$ (left) and $\epsilon = 10^{-12}$ (right). Other parameter values are $\xi = N^{1/4}/4$, $r_c = 2\pi$, and $k_{max} = 3\xi$. The dashed line shows $E_\infty \sim h^3$, for comparison.

Figure 1 shows the truncation error $E_\infty(h)$ versus N in the case of the first method for two values of the parameter ϵ in (81) which controls the accuracy of the fast erfc summation. In the left panel we set $\epsilon = 10^{-6}$; the error is $O(h^3)$ but does not decrease below $E_\infty \approx 10^{-6}$ as N is increased. Reducing ϵ to 10^{-12} (Figure 1, right) improves the accuracy of the reciprocal sum computation and gives $O(h^3)$ convergence for a wider range of N. CPU times for the more accurate computation in the right panel of Figure 1 are given in Figure 2. This shows

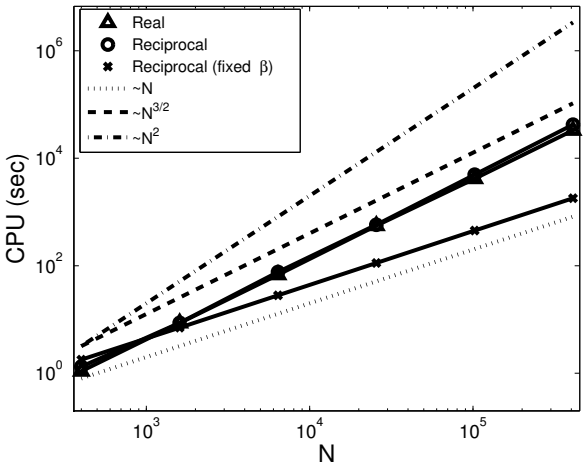

Figure 2: CPU times for direct calculation of the real space sum (method 1) and fast computation of the reciprocal sum when parameters are scaled with N as in Figure 1, right. Also shown is the CPU time for the reciprocal sum with fixed parameters $\xi = 1.5$, $r_c = 2\pi$, $k_c = 3\xi$ and $\epsilon = 10^{-12}$ ('x' markers). The dashed lines illustrate N, $N^{3/2}$ and N^2 scaling.

that if the parameters are scaled with N as in (87) and the real space integral is computed by direct summation, we observe the expected $O(N^{3/2})$ complexity of the Birkhoff-Rott quadrature without the treecode. This method is of suboptimal complexity, but is simple to implement in that it does not involve the treecode. The figure also shows that when ξ, r_c, and k_c are instead fixed and the fast integration method of §6.3.1 is used, the reciprocal sum is computed in an optimal $O(N)$ amount of work. This is exploited in the treecode-accelerated Ewald algorithm discussed below.

Quadrature error and CPU times for the treecode-accelerated Ewald method are shown in Figure 3. The reciprocal space integral is computed with fixed parameters (i.e., without dependence on N) and exhibits $O(N)$ complexity like that shown by the 'x' markers in Figure 2. The real space integral is computed by the treecode in $O(N \log N)$ operations, and the figure shows that the order of accuracy of the combined method is $O(h^3)$ and the total CPU time is consistent with an $O(N \log N)$ complexity.

7.2. Parameters

In the following numerical examples, we use the initial interface shape (62) for which $\lambda(t) = 1$. We fix the surface tension at $B = 1$ (unless noted otherwise) and set the relaxation parameter to $\theta = 50$. We use Ewald parameters (87) for quadrature method 1 and (88) for method 2, with treecode specific parameters $q_T = 9$ and $\epsilon_T = 0.35$. In all examples $\epsilon = 10^{-12}$.

7.3. Convergence

We test the order of accuracy of the full method, which is designed to be second order in time and third order in space. The temporal accuracy is validated by fixing the number of space points at $N = 32^2$ and performing runs to $t = 0.4$ with $\Delta t_i = 0.01/2^i$ for $i = 0, \ldots, 3$.

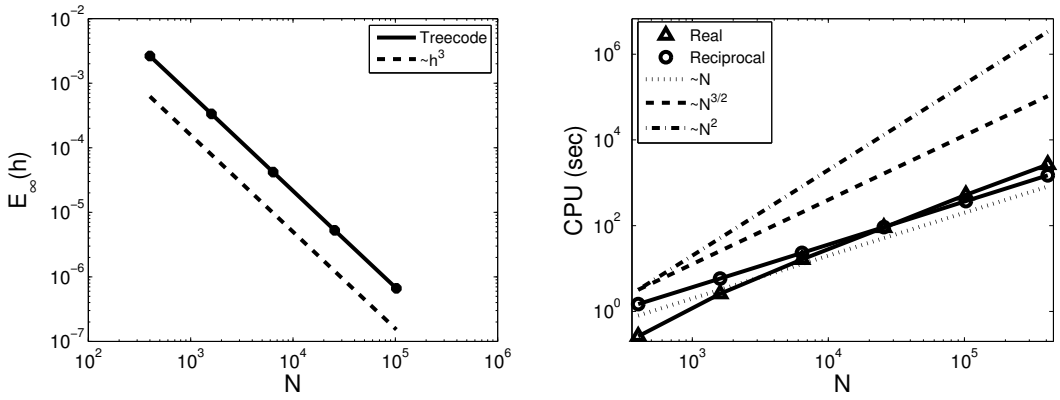

Figure 3: Order of accuracy and CPU times for the treecode-accelerated Ewald sum with $\xi = 1.5$, $r_c = 2\pi$ and $k_c = 3\xi$. Treecode specific parameters are $q_T = 10$ and $\epsilon_T = 0.27$. Left: Log-log plot of truncation error E_∞ versus N. The dashed line shows $E_\infty \sim h^3$. Right: CPU times for the real and reciprocal space integrals. The dashed lines are as in Figure 2.

We set $W = 0$ and the initial interface shape is given by (62) with $A = 0.1$. The deformed initial interface relaxes under surface tension toward a flat surface. The GMRES tolerance is set to 10^{-12}. In this example we use method 1 with direct real space summation, but similar results are obtained with the treecode-accelerated method. We empirically calculate the order of accuracy p using the formula

$$p = \frac{1}{\ln 2} \ln \frac{||\mathbf{X}_{\Delta t} - \mathbf{X}_{\Delta t/2}||_\infty}{||\mathbf{X}_{\Delta t/2} - \mathbf{X}_{\Delta t/4}||_\infty}, \tag{89}$$

where $||\cdot||_\infty$ is the max norm on the coarsest grid. The results are shown in Table 1, and verify the second order time accuracy of the method.

Δt	p
0.01	–
0.005	–
0.0025	2.0013
0.00125	2.0007

Table 1: Temporal order of accuracy p calculated from (89).

Next we test the order of spatial accuracy. The time step is fixed at $\Delta t = 0.002$ and we perform runs to $t = 0.2$ for $N = 16^2, 32^2, 64^2$ and 128^2. The other parameters are the same as for the temporal validation. The order of spatial accuracy is calculated from (89) with grid spacing Δx in place of Δt. The results, which are shown in Table 2, validate the third order spatial accuracy.

Similar results are obtained for both temporal and spatial accuracy when the Bond number W is nonzero.

Δx	p
16^2	–
32^2	–
64^2	2.9877
128^2	2.9967

Table 2: Spatial order of accuracy p.

There is no known exact solution to the 3D initial value problem for (7), so the full time-dependent numerical method is further validated using an analytical solution to the governing equations linearized about a flat interface. The linearized solution for $\lambda = 1$ is

$$x_L(\vec{\alpha}, t) = 2\pi\alpha, \quad y_L(\vec{\alpha}, t) = 2\pi\beta, \quad z_L(\vec{\alpha}, t) = \sum_{\mathbf{k}} \widehat{z}_{\mathbf{k}}(0) \exp[\sigma(k)t + i\mathbf{k} \cdot \vec{\alpha}] \tag{90}$$

where $\widehat{z}_{\mathbf{k}}(0)$ are the Fourier coefficients of the initial data and

$$\sigma(k) = \frac{k}{2}\left(W - \frac{Bk^2}{2}\right) \tag{91}$$

with $k = |\mathbf{k}|$. Figure 4 compares the maximum amplitude of the full numerical and linearized solutions for initial interface shape (62) with $A = 0.1$ and $A = 0.5$. When $A = 0.1$, the time-dependent amplitudes nearly overlap in the given time interval, with the difference between the computed and linearized solutions being less than 10^{-2} in the maximum norm. At the larger amplitude $A = 0.5$, nonlinear effects are observed and the two solutions diverge after a short time. The close agreement between the linearized and numerical solution at a sufficiently small amplitude further validates the method.

7.4. Stability

The time-step constraint for stability of the semi-implicit method is compared with an explicit discretization of the evolution equation (7). The explicit method does not require isothermal coordinates and for it we set $V_1 = V_2 = 0$. We consider the relaxation under surface tension ($W = 0$) of the initial interface shape (62), with $A = 0.1$. For an isothermal parameterization with $n_p = \sqrt{N}$ equally spaced node points in the α and β variables, a frozen coefficient analysis [16] shows the time step constraint for stability of an explicit method to be

$$\Delta t \leq C E_m^{3/2}/(BN^{3/2}), \tag{92}$$

where C is a constant independent of N and $E_m = \min_{\alpha,\beta} E(\alpha, \beta)$. The dependence of Δt on $1/N^{3/2}$ is expected to hold even for more general interface parameterizations.

Figure 5 shows the Fourier coefficients $\log_{10}|\hat{z}_{\mathbf{k}}|$ versus $\mathbf{k} = (k, l)$ calculated by the explicit method for the two resolutions $N = 32^2$ (top) and $N = 64^2$ (bottom). Fourier coefficients are shown for $0 \leq k \leq \sqrt{N}/2$ and $0 \leq l \leq \sqrt{N} - 1$; modes with $l = \sqrt{N}/2 + 1$ to $l = \sqrt{N} - 1$ actually correspond to negative-l wavenumber modes. Negative-k modes have

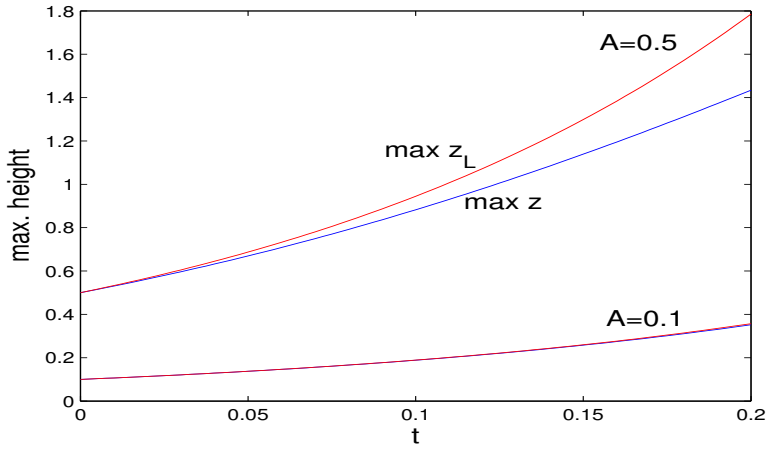

Figure 4: Maximum interface height for full numerical solution z and linearized solution z_L. Initial interface shape is (62) with $A = 0.5$ (upper curves) and $A = 0.1$ (lower curves). The two curves at $A = 0.1$ nearly overlap.

the same amplitude as the positive-k modes, due to the real-valuedness of z, and are not shown. There are no spectral or Fourier filters [6] used in the calculation, so that we may assess the stability of high-wavenumber modes generated by round-off error.

Instability in a spectral method is typically observed as a rapid and unphysical growth of the high-wavenumber modes. High-wavenumber growth and instability are clearly exhibited in the calculation at the top left in Figure 5. The time step here, $\Delta t = 10^{-3}$, is above but close to the stability threshold, and the explicit method is found to be stable at $t = 0.04$ when the time step is decreased to 4.0×10^{-4}, as shown in the plot at top right. Increasing the number of node points to $N = 64^2$ and reducing the time steps by a factor of eight gives instability at $\Delta t = 1.25 \times 10^{-4}$ (bottom, left) and stability at $\Delta t = 5.0 \times 10^{-5}$ (bottom, right). The factor of eight reduction in time step required to maintain stability is consistent with the constraint (92).

Results from the semi-implicit method using $N = 128^2$ and a larger time step $\Delta t = 2.5 \times 10^{-2}$ are shown in Figure 6. It is clear from the spectrum that the method is stable, even at this higher value of N and larger time step compared with the explicit calculations in Figure 5. (We note that the Fourier spectrum exhibits noise in the high-wavenumber modes, at the level of round-off). The computation can be continued stably until the interface completely relaxes to a flat sheet. The method shows similar stability for larger initial perturbations, and for interface motion driven by gravity with $W > 0$. For example, Figure 7 shows the surface-tension-driven evolution with a larger initial perturbation $A = 0.5$ and $\Delta t = 2.5 \times 10^{-2}$. The solution at time $t = 1.0$ (Figure 7, left) shows that the interface smoothly relaxes toward a flat sheet, with no sign of numerical instability. The calculation can be continued until the sheet relaxes to $z = 0$. Figure 7 (right) plots the logarithm of $D_1(t) = \max_{\alpha,\beta} \frac{|E-G|}{E}$ and $D_2(t) = \max_{\alpha,\beta} \frac{|F|}{|\mathbf{X}_\alpha||\mathbf{X}_\beta|}$ versus t, which measure how close the parameterization is to being isothermal. For the evolution here, with decaying interface

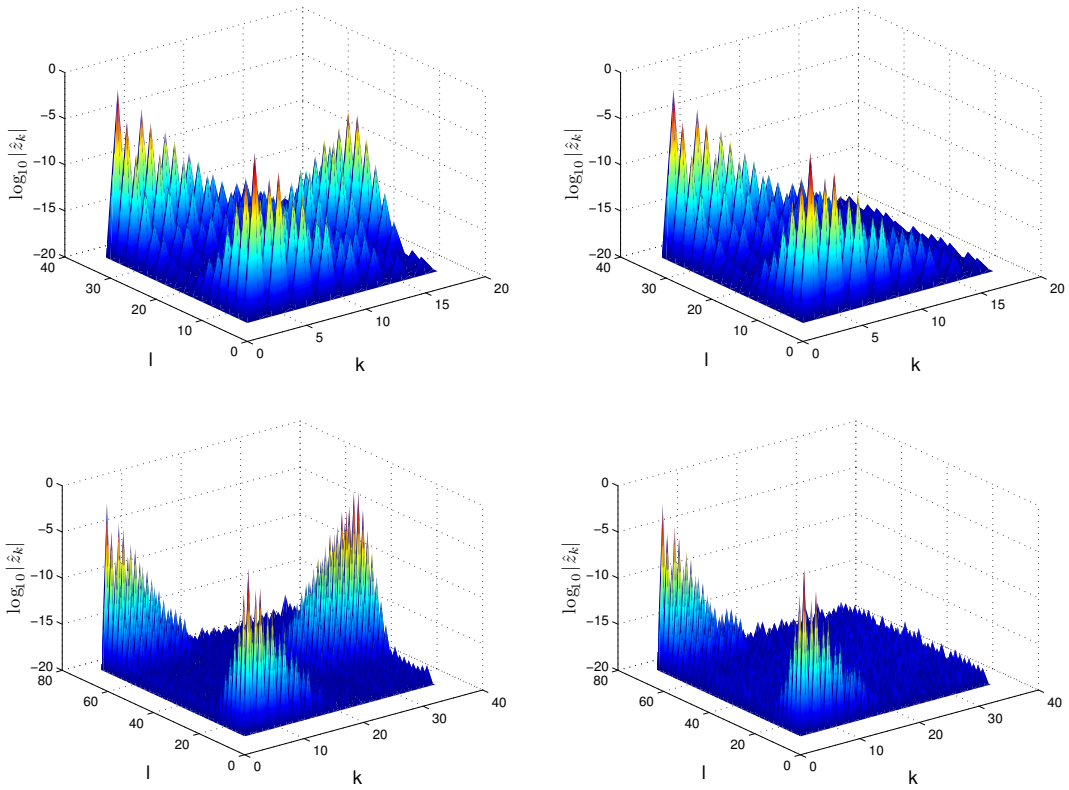

Figure 5: Plot of $\log_{10}|\hat{z}_{\mathbf{k}}|$ versus $\mathbf{k}$ at $t = 0.04$ for the explicit method. Top left: $N = 32^2$, $\Delta t = 10^{-3}$. Top right: $N = 32^2$, $\Delta t = 4.0 \times 10^{-4}$. Bottom left: $N = 64^2$, $\Delta t = 1.25 \times 10^{-4}$. Bottom right: $N = 64^2$, $\Delta t = 5.0 \times 10^{-5}$. Modes with amplitude less than 10^{-18} are not shown.

amplitude, these quantities eventually decrease in time. This indicates the effectiveness of the relaxation terms in (22), (23) at maintaining the isothermal parameterization.

Table 3 compares the largest stable time step for the explicit and semi-implicit methods. This time step is identified as one which gives stable results in the given time interval, but leads to numerical instability or non-convergence of the GMRES method when doubled. The 'explicit' data reflects the time-step constraint (92), in that the largest stable time step scales with $1/B$ (compare 'explicit' columns one and two) and with $1/N^{3/2}$ (compare 'explicit' rows one through three). In contrast, the semi-implicit method is stable at much larger time steps and is not subject to a high order time-step constraint. For the 'implicit' data, we report the largest time step for which the GMRES iteration converges over the given time interval at a tolerance of 10^{-8} (this tolerance is approximately the maximum amplitude of the highest wavenumber modes for the $N = 32^2$ computation in column 6). GMRES fails to converge when Δt is doubled from the values listed in the table. The table provides evidence that the semi-implicit method developed here is effective at reducing the time-step constraint for stability from the presence of surface tension.

Hou, Lowengrub and Shelley [3] note that there is generally a first order CFL constraint

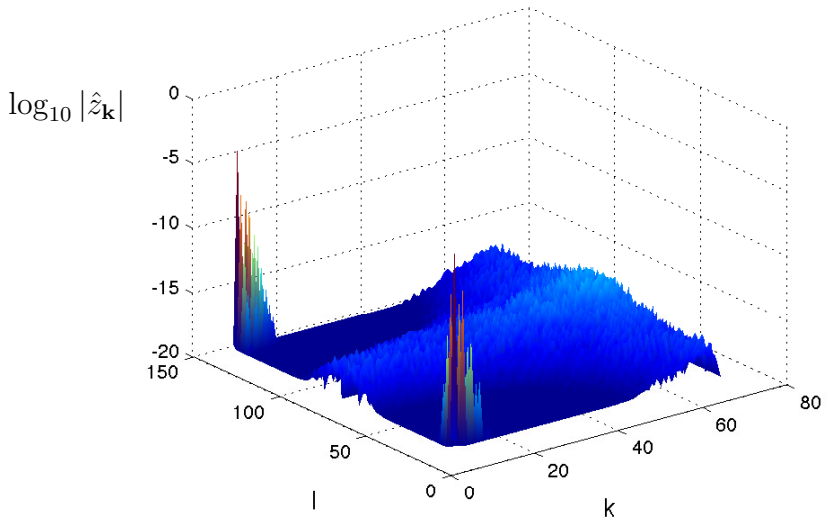

Figure 6: Plot of $\log_{10}|\hat{z}_{\mathbf{k}}|$ versus $\mathbf{k}$ at $t = 1.0$ for the implicit method with $\Delta t = 2.5 \times 10^{-2}$ and $N = 128^2$.

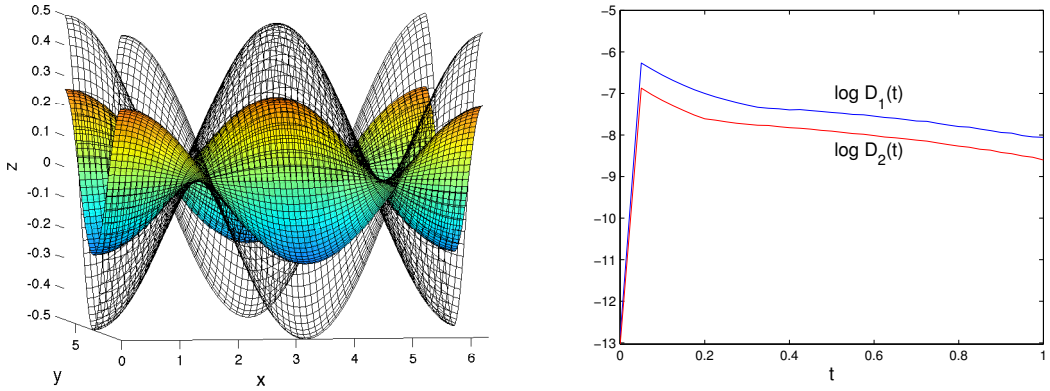

Figure 7: Surface-tension-driven evolution of the initial interface shape (62) with $A = 0.5$, $W = 0$, $N = 128^2$, and $\Delta t = 2.5 \times 10^{-2}$. The left panel shows the initial interface and the reduced amplitude solution at $t = 1$ (the colors are for visibility only). The right panel shows the logarithm of $D_1(t) = \max_{\alpha,\beta} \frac{|E-G|}{E}$ and $D_2(t) = \max_{\alpha,\beta} \frac{|F|}{|\mathbf{X}_\alpha||\mathbf{X}_\beta|}$ versus t, which measure the relative error in the isothermal parameterization.

coming from a transport term in the governing equations. They observe this CFL constraint in the 2D problem only at a relatively large number of interface grid points. Nevertheless, HLS recommend that long time simulations be performed with a CFL condition in mind. Further comments on the CFL condition are given in §7.5.

7.5. Complexity

The GMRES solution of the discrete linear system (49) (or the second order version (58)) has $O(N \log N)$ complexity per iteration. The slowest component of the numerical solution

	explicit			implicit		
N	W=0, B=1.0	W=0, B=5.0	W=10, B=1.0	W=0, B=1.0	W=0, B=5.0	W=10, B=1.0
32^2	5.0×10^{-4}	1.0×10^{-4}	5.0×10^{-4}	1.0×10^{-1}	2.0×10^{-2}	6.0×10^{-2}
64^2	6.25×10^{-5}	1.25×10^{-5}	6.25×10^{-5}	1.0×10^{-1}	2.0×10^{-2}	8.0×10^{-2}
128^2	7.8×10^{-6}	1.56×10^{-6}	7.8×10^{-6}	1.0×10^{-1}	2.0×10^{-2}	4.0×10^{-2}
256^2	1.0×10^{-6}	2.0×10^{-7}	1.0×10^{-6}	1.0×10^{-1}	2.0×10^{-2}	4.0×10^{-2}

Table 3: Largest stable time step for the explicit and semi-implicit methods. The GMRES tolerance is 10^{-8}. Initial data is (62) with $A = 0.5$ for $W = 0$ and $A = 0.1$ for $W = 10$. The $W = 0$ runs are continued until the interface has nearly relaxed to a flat surface, and $W = 10$ runs are taken to $t = 0.4$.

is the calculation of the Birkhoff-Rott integral. However, this enters only in the right hand side of the linear system and is computed once per time step, for either the first or second order in time method. Other terms such as U_s^{n+1} are computed 'fast' using the 2D FFT, which has the same asymptotic complexity as the Ewald calculation but a much smaller constant prefactor. Therefore, the number of GMRES iterations does not significantly affect the overall CPU time, unless it becomes too large. We show that the preconditioner is effective at reducing the number of GMRES iterations.

	Preconditioner				
	W=0		W=10		
N	None	$\mathcal{L}_p^{-1}(f)$	None	$\mathcal{L}_p^{-1}(f)$	$\mathcal{L}_p^{-1}(f)$, varying Δt
32^2	3	2	3	2	3 (0.002)
64^2	15	6	6	5	5 (0.001)
128^2	46	19	22	14	7 (0.0005)
256^2	166	55	53	38	10 (0.00025)

Table 4: Number of GMRES iterations in the solution of (58) using the treecode-accelerated Ewald sum (method 2). In columns 1 through 4, $\Delta t = 0.005, 0.001$ for $W = 0, 10$, respectively. The time step in column 5 is given in parentheses, and represents a first order CFL constraint. The GMRES tolerance is 10^{-8}. The number of iterations are tabulated at $t = 0.1$ for $W = 0$ and $t = 0.4$ for $W = 10$.

The performance of the GMRES method is summarized in Table 4, which shows the number of iterations required to invert the linear system (50) for the second order in time algortihm with the treecode-accelerated Ewald sum (method 2). We consider $W = 0$ and 10 and an initial interface shape given by (62), with $A = 0.5$ when $W = 0$ and $A = 0.1$ for $W = 10$. The preconditioner reduces the number of GMRES iterations in all cases, although this number is still observed to be weakly mesh dependent. The number of iterations also depends on the time step. If the first order CFL constraint is taken into account (as recommended by HLS) and the time step is reduced by a factor of 2 when $n_p = \sqrt{N}$ is doubled, then the number of GMRES iterations increases rather slowly (Table 4, last column). We emphasize that each iteration involves only a few calls to a 2D FFT, and unless

the number of iterations becomes very large the CPU time spent in the GMRES algorithm is only a small fraction of the total computational effort. For example, with $N = 128^2$ the total CPU time for the 14 GMRES iterations with the preconditioner is about 10^{-1} s, while a single calculation of the Birkhoff-Rott integral using method 2 takes 10^2 s.

7.6. Long-time evolution

The efficiency of the non-stiff method enables long-time computations which would otherwise be prohibitively slow using an explicit method. Figure 8 (top) presents an example of the long-time evolution for gravity-driven flow with $W = 10$ and a slightly perturbed initial interface with $A = 0.1$. The interface develops a fingering instability which is a 3D analogue of a well-known phenomenon in 2D porous media or Hele-Shaw flow. The periodic box in (x, y) acts as impermeable walls for the flow and the finger aligns with the 'walls', similar to the alignment of fingers in 2D wall-bounded or channel flow. There are upward and downward propagating fingers, separated by saddle points on the interface. The finger width is approximately one-half of the periodic box or 'channel.'

The lower left panel of Figure 8 shows the Fourier coefficients $\log_{10} |\hat{z}_{\mathbf{k}}|$ at $t = 0.65$, projected onto the k-axis. The Fourier spectrum oscillates since many modes have zero amplitude due to symmetry, but the envelope of the spectrum for $0 < k \lesssim 50$ shows a nice exponential decay that is characteristic of stable computations of smooth solutions (the decay appears linear on the semi-log plot). High-wavenumber modes with $50 \lesssim k$ have amplitudes well below the GMRES tolerance at $t = 0.65$. There is no sign of numerical instability in the high-wavenumber modes.

The lower right panel of Figure 8 shows the relative error in the isothermal parameterization as measured by the logarithm of $D_1(t) = \max_{\alpha,\beta} \frac{|E-G|}{E}$ and $D_2(t) = \max_{\alpha,\beta} \frac{|F|}{|\mathbf{X}_\alpha||\mathbf{X}_\beta|}$. The isothermal parameterization is effectively enforced during the computation, although D_1 and D_2 increase slowly over time due to interface and grid stretching. The computation is stopped when D_1 is near 10^{-3}. We recall that the drift of the parameterization away from isothermal is not a source of error, since the exact curvature is used in the computation. However, if D_1 and D_2 become too large the method of stiffness reduction may become less effective. The magnitude of these quantities at the end of the computation is a sign that localized grid stretching is leading to underresolution of the interface. This is most evident at the tip of the finger, which is becoming underresolved at $t = 0.9$. In the current work, the requirement of an isothermal parameterization precludes the redistribution of grid points to control their spacing on the interface. This topic will be addressed in future work.

8. Conclusions

We have presented a non-stiff boundary integral method for three-dimensional porous media flow with surface tension. The stiffness is removed by a three-dimensional analogue of the small-scale decomposition developed by Hou, Lowengrub and Shelley for two-dimensional interfacial flow. Our method employs a special isothermal parameterization of the interface, which is enforced by an appropriate choice of tangential velocities. The isothermal parameterization allows the problem to be reformulated so that high order terms that are dominant

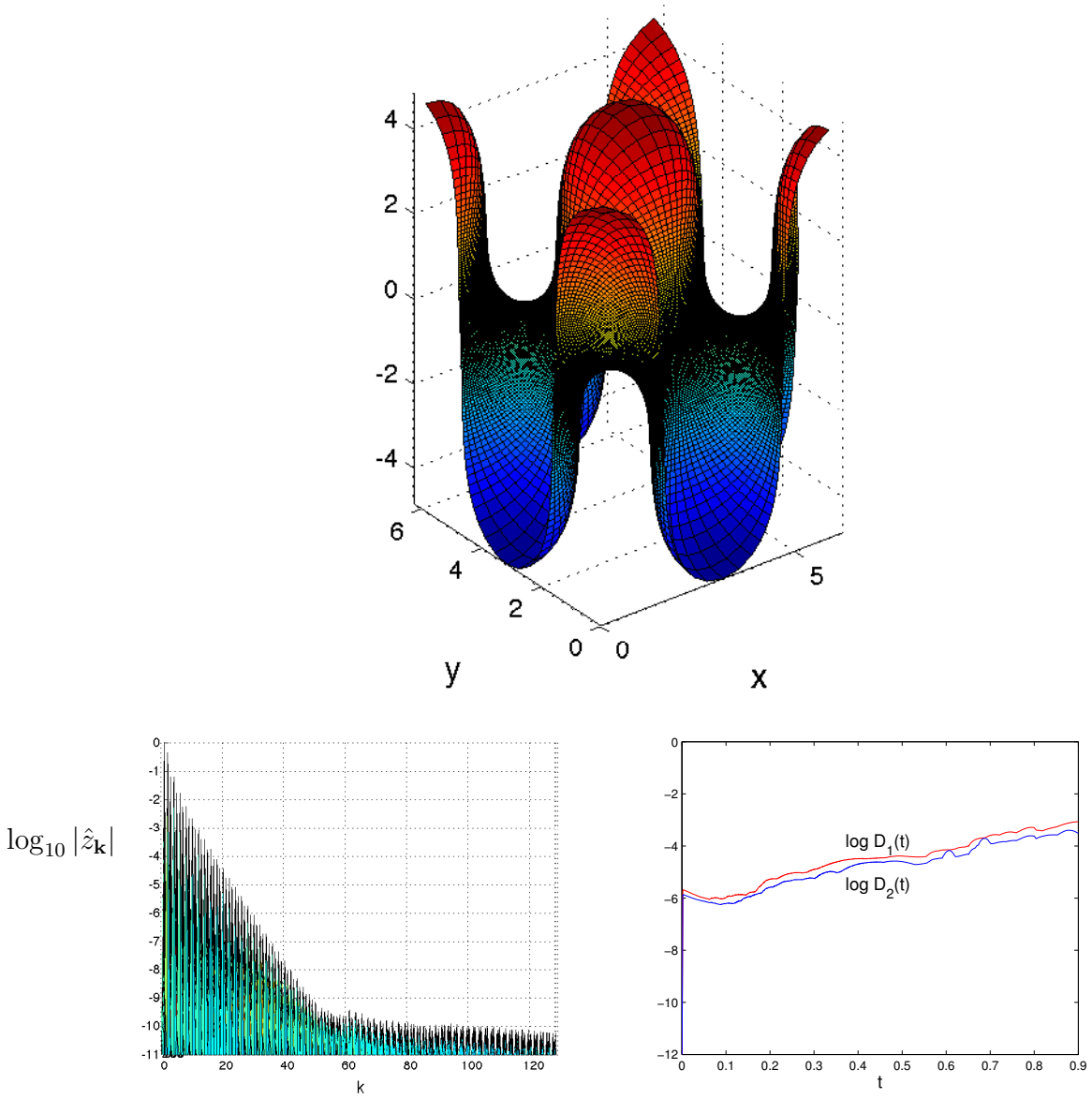

Figure 8: Top: Interface shape for $W = 10$ at $t = 0.9$. Parameters are $N = 256^2$, $\Delta t = 10^{-3}$ for $0 < t \leq 0.65$ and $\Delta t = 5 \times 10^{-4}$ for $0.65 < t \leq 0.9$. The GMRES tolerance is 10^{-7}. Bottom left: Fourier coefficients $\log_{10}|\hat{z}_{\mathbf{k}}|$ at $t = 0.65$, projected onto the k-axis. Bottom right: $\log_{10} D_1(t)$ and $\log_{10} D_2(t)$ plotted versus t.

at small scales appear quasilinearly, and simplify under the Fourier transform. This enables the efficient application of implicit time-integration methods. An important component of our method is an efficient algorithm for calculating the Birkhoff-Rott integral for a doubly-periodic surface. This algorithm, based on Ewald summation, computes the integral in $O(N \log N)$ operations per time step.

The numerical method is tested by computing the relaxation of a perturbed interface

to a flat surface under the action of surface tension, and the evolution of a gravity-driven fingering instability. The algorithm is found to be effective at eliminating the third order time-step constraint that plagues explicit time-integration methods. Furthermore, the method is efficient, requiring approximately the same amount of work per time step as an explicit method. The algorithm developed here is second order accurate in time and third order in space, although it is straightforward to achieve arbitrary $O(h^p)$ spatial accuracy (for odd p).

The method developed in this paper applies broadly, in problems beyond that considered here. It is generally useful for computing the motion of doubly-periodic fluid interfaces with surface tension in three-dimensional flow, when a boundary integral formulation of the problem exists. The main drawback of the method is that the restriction to isothermal coordinates makes it difficult to implement adaptive mesh refinement. In addition, the current algorithm does not apply to a closed surface (e.g., a drop or bubble) since this does not have a global isothermal parameterization. Such an interface can be treated by decomposing it into a collection of overlapping patches, each of which is separately parameterized by isothermal coordinates. This 'domain decomposition' approach can also be useful for adaptive mesh refinement, and is a topic of current research.

Acknowledgements
This work was supported by the NSF under Grant Nos. DMS-1016406 and DMS-1009105 (MS) and DMS-1016267 and DMS-1008387 (DMA).

Appendix A. The Ewald sum

We derive the Ewald sum (67) following Baker et al. [30], with modifications to account for the reflection and constant terms added for conditional convergence. Start with the doubly-periodic Green's function (66), use the identity

$$\frac{2}{\sqrt{\pi}}\int_0^\infty e^{-aq^2}dq = \frac{1}{\sqrt{a}},$$

and let $s_\pm = [(x-x')/2 \pm m\pi]^2 + [(y-y')/2 \pm n\pi]^2 + [(z-z')/2]^2$ to obtain

$$G(\mathbf{X},\mathbf{X}') = \frac{1}{2\sqrt{\pi}}{\sum_{\mathbf{m}\in\mathbb{Z}^2}}' \int_0^\infty \left(e^{-q^2 s_-} + e^{-q^2 s_+} - \frac{2}{\pi} e^{-q^2|\mathbf{m}|^2} \right) dq,$$

where the prime on the sum indicates that the third term in parentheses is omitted when $|\mathbf{m}| = 0$. Although the terms in parentheses must be kept together to obtain a convergent sum, for convenience we consider each sum separately. Thus, let G_k for $k = 1, \ldots, 3$ be the contribution to G from the kth term in parentheses. G is written as a rapidly convergent sum of error functions by splitting the integrals, e.g., $G_1 = G_1^a + G_1^b$, where

$$G_1^a(\mathbf{X},\mathbf{X}') = \frac{1}{2\sqrt{\pi}}\sum_{\mathbf{m}\in\mathbb{Z}^2}\int_0^\xi e^{-q^2 s_-}dq, \tag{A.1}$$

$$G_1^b(\mathbf{X},\mathbf{X}') = \frac{1}{2\sqrt{\pi}}\sum_{\mathbf{m}\in\mathbb{Z}^2}\int_\xi^\infty e^{-q^2 s_-}dq, \tag{A.2}$$

and similarly for G_2 and G_3. The sum in G_1^b converges rapidly, and we perform a change of variables to get

$$G_1^b(\mathbf{X},\mathbf{X}') = \frac{1}{2\sqrt{\pi}} \sum_{\mathbf{m}\in\mathbb{Z}^2} \frac{1}{\sqrt{s_-}} \int_{\sqrt{s_-}\xi}^{\infty} e^{-\tilde{q}^2} d\tilde{q} = \frac{1}{4} \sum_{\mathbf{m}\in\mathbb{Z}^2} \frac{\mathrm{erfc}(\sqrt{s_-}\xi)}{\sqrt{s_-}}. \tag{A.3}$$

To achieve rapid convergence in G_1^a (when q is small), apply the Jacobi identity

$$\frac{1}{\sqrt{\pi}} \frac{1}{q} \sum_{n=-\infty}^{\infty} \exp\left(-\frac{n^2}{q^2} + 2inb\right) = \sum_{n=-\infty}^{\infty} \exp\left(-(b-n\pi)^2 q^2\right)$$

to (A.1), which then gives

$$G_1^a(\mathbf{X},\mathbf{X}') = \frac{1}{2\pi^{3/2}} \sum_{\mathbf{m}\in\mathbb{Z}^2} \int_0^{\xi} \frac{1}{q^2} \exp\left(-\frac{m^2}{q^2} + 2im\frac{x-x'}{2}\right) \times \exp\left(-\frac{n^2}{q^2} + 2in\frac{y-y'}{2}\right) \exp\left(-q^2\left(\frac{z-z'}{2}\right)^2\right) dq.$$

Let $\tilde{q} = \dfrac{1}{q}$ and rewrite the sum to obtain

$$\begin{aligned} G_1^a(\mathbf{X},\mathbf{X}') &= \frac{1}{2\pi^{3/2}} \sum_{\mathbf{m}\in\mathbb{Z}^2} \int_{1/\xi}^{\infty} e^{-(m^2+n^2)\tilde{q}^2} e^{-\frac{1}{\tilde{q}^2}\left(\frac{z-z'}{2}\right)^2} d\tilde{q} \cdot e^{im(x-x')} e^{in(y-y')} \\ &= \frac{1}{8\pi^{3/2}} \sum_{\mathbf{m}\in\mathbb{Z}^2} \int_{1/\xi}^{\infty} e^{-(m^2+n^2)\tilde{q}^2} e^{-\frac{1}{\tilde{q}^2}\left(\frac{z-z'}{2}\right)^2} d\tilde{q} \cdot \left(e^{im(x-x')} + e^{-im(x-x')}\right)\left(e^{in(y-y')} + e^{-in(y-y')}\right) \\ &= \frac{1}{2\pi^{3/2}} \sum_{\mathbf{m}\in\mathbb{Z}^2} \int_{1/\xi}^{\infty} e^{-(m^2+n^2)\tilde{q}^2} e^{-\frac{1}{\tilde{q}^2}\left(\frac{z-z'}{2}\right)^2} d\tilde{q} \cdot \cos m(x-x') \cos n(y-y'). \end{aligned}$$

Use

$$\int_x^{\infty} e^{-k^2\tilde{x}^2 - b^2/\tilde{x}^2} d\tilde{x} = \frac{\sqrt{\pi}}{4k}\left[e^{2kb}\mathrm{erfc}\left(kx + \frac{b}{x}\right) + e^{-2kb}\mathrm{erfc}\left(kx - \frac{b}{x}\right)\right]$$

with

$$k = \sqrt{m^2+n^2}, \qquad b = \frac{z-z'}{2},$$

to finally obtain

$$G_1^a(\mathbf{X},\mathbf{X}') = \frac{1}{8\pi} \sum_{\mathbf{m}\in\mathbb{Z}^2} R_{mn}(z-z') \cos m(x-x') \cos n(y-y'), \tag{A.4}$$

where

$$\begin{aligned} R_{mn}(z-z') = \frac{1}{\sqrt{m^2+n^2}}\Bigg[& e^{\sqrt{m^2+n^2}(z-z')}\mathrm{erfc}\left(\frac{\sqrt{m^2+n^2}}{\xi} + \frac{z-z'}{2}\xi\right) \\ + & e^{-\sqrt{m^2+n^2}(z-z')}\mathrm{erfc}\left(\frac{\sqrt{m^2+n^2}}{\xi} - \frac{z-z'}{2}\xi\right)\Bigg]. \end{aligned} \tag{A.5}$$

The same analysis shows that G_2^a and G_2^b can be written as identical rapidly convergent sums as those for G_1^a and G_1^b. We also apply the above transformations to G_3 to obtain

$$G_3^b = -\frac{1}{2\pi} {\sum_{\mathbf{m}\in\mathbb{Z}^2}}' \frac{\mathrm{erfc}(\pi\xi|\mathbf{m}|)}{|\mathbf{m}|}, \tag{A.6}$$

and

$$\begin{aligned} G_3^a &= -\frac{1}{\sqrt{\pi}} {\sum_{\mathbf{m}\in\mathbb{Z}^2}}' \int_0^\xi e^{-\pi^2|\mathbf{m}|^2 q^2}\, dq \\ &= -\frac{1}{\sqrt{\pi}} \sum_{\mathbf{m}\in\mathbb{Z}^2} \int_0^\xi e^{-\pi^2|\mathbf{m}|^2 q^2}\, dq + \frac{\xi}{\sqrt{\pi}} \\ &= -\frac{1}{2\pi} \sum_{\mathbf{m}\in\mathbb{Z}^2} \frac{\mathrm{erfc}(\frac{|\mathbf{m}|}{\xi})}{|\mathbf{m}|} + \frac{\xi}{\sqrt{\pi}}. \end{aligned} \tag{A.7}$$

The Ewald sum (67) follows from combining (A.3)-(A.7) and the equivalent sums for G_2, and writing in terms of $s = s_-$.

[1] S. Tanveer, Surprises in viscous fingering, J. Fluid Mech. 409 (2000) 273–308.

[2] T. Y. Hou, Z. Li, S. Osher, H. Zhou, A hybrid method for moving interface problems with application to Hele-Shaw flow, J. Comp. Phys. 134 (1997) 236–252.

[3] T. Y. Hou, J. S. Lowengrub, M. J. Shelley, Removing the stiffness from interfacial flows with surface tension, J. Comp. Phys. 114 (1994) 312–338.

[4] G. Tryggvason, H. Aref, Numerical experiments on Hele-Shaw flow with a sharp interface, J. Fluid Mech. 136 (1983) 20.

[5] G. Baker, R. E. Caflisch, M. Siegel, Singularity formation during Rayleigh-Taylor instability, J. Fluid Mech. 252 (1993) 51–78.

[6] R. Krasny, A study of singularity formation in a vortex sheet by the point vortex approximation, J. Fluid Mech. 167 (1986) 65–93.

[7] T. Hou, J. Lowengrub, M. Shelley, The long-time motion of vortex sheets with surface tension, Phys. Fluids 9 (1997) 1933.

[8] D. M. Ambrose, N. Masmoudi, Well-posedness of 3D vortex sheets with surface tension, Comm. Math. Sci. 5 (2007) 391–430.

[9] D. M. Ambrose, Well-posedness of two-phase Darcy flow in 3-D, Quart. Appl. Math. 65 (2007) 189–203.

[10] D. M. Ambrose, N. Masmoudi, The zero surface tension limit of three-dimensional water waves, Indiana Univ. Math. J. 50 (2009) 479–521.

[11] T. Darden, D. York, L. Pedersen, Particle mesh Ewald an Nlog(N) method for Ewald sums in large systems, J. Chem. Phys. 98 (1993) 10089–10092.

[12] D. Lindbo, A.-K. Tornberg, Spectral accuracy in fast Ewald-based methods for particle simulations, J. Comp. Phys. 230 (2011) 8744–8761.

[13] Z.-H. Duan, R. Krasny, An Ewald summation based multipole method, J. Chem. Phys. 113 (2000) 3492–3495.

[14] D. Lindbo, A.-K. Tornberg, Fast and spectrally accurate Ewald summation for 2-periodic electrostatic systems, J. Chem. Phys. 136 (2012) 164111.

[15] D. J. Haroldsen, D. I. Meiron, Numerical calculation of three-dimensional interfacial potential flows using the point vortex method, SIAM J. Sci. Comp. 20(2) (1998) 648–683.

[16] D. M. Ambrose, M. Siegel, A non-stiff boundary integral method for 3D porous media flow with surface tension, Math. Comp. Sim. 82 (6) (2012) 968–983.

[17] T. Y. Hou, G. Hu, P. Zhang, Singularity formation in 3-D vortex sheets, Phys. Fluids 15(1) (2003) 147–172.

[18] T. Y. Hou, P. Zhang, Convergence of a boundary integral method for 3-D water waves, Discrete Cont. Dyn. Sys., Series B 2(1) (2002) 1–24.

[19] T. Y. Hou, I. Klapper, H. Si, Removing the stiffness of curvature in computing 3-D filaments, J. Comp. Phys. 143 (1998) 628–664.

[20] A.-K. Tornberg, M. Shelley, Simulating the dynamics and interactions of flexible fibers in Stokes flows, J. Comp. Phys. 196 (2004) 8.

[21] H. D. Ceniceros, H. Si, Computation of axisymmetric suction flow through porous media in the presence of surface tension, J. Comp. Phys. 165 (2000) 237–260.

[22] Q. Nie, The nonlinear evolution of vortex sheets with surface tension in axisymmetric flows, J. Comp. Phys. 174 (2001) 438–459.

[23] S. K. Veerapaneni, D. Gueyner, D. Zorin, G. Biros, A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D, J. Comp. Phys. 228(7) (2009) 2334–2353.

[24] S. K. Veerapaneni, D. Gueyner, D. Zorin, G. Biros, A numerical method for simulating the dynamics of 3D axisymmetric vesicles suspended in viscous flow, J. Comp. Phys. 228(19) (2009) 7233–7249.

[25] S. K. Veerapaneni, A. Rahimian, G. Biros, D. Zorin, A fast algorithm for simulating vesicle flows in three dimensions, J. Comp. Phys. 230(14) (2011) 5610–5634.

[26] R. E. Caflisch, X.-F. Li, Lagrangian theory for 3D vortex sheets with axial or helical symmetry, Trans. Th. Stat. Phys. 21 (1992) 559–578.

[27] M. Lipschutz, Differential Geometry, McGraw-Hill, New York, 1969.

[28] U. M. Ascher, S. J. Ruuth, B. T. R. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM J. Num. Anal. 32(3) (1995) 797–823.

[29] J. T. Beale, A convergent boundary integral method for three-dimensional water waves, Math. Comp. 70 (2004) 977–1029.

[30] G. R. Baker, D. I. Meiron, S. A. Orszag, Boundary integral methods for axisymmetric and three-dimensional Rayleigh-Taylor instability problems, Physica D 12 (1984) 19–31.

[31] A. Grzybowski, E. Gwozdz, A. Brodka, Ewald summation of electrostatic interactions in molecular dynamics of a three-dimensional system with periodicity in two directions, Phys. Rev. B 61 (2000) 6706–6712.

[32] T. C. Raykar, R. Duraiswami, B. Krishnapuram, Fast weighted summation of erfc functions, Tech. Rep. CS-TR-4848, Department of Computer Science, University of Maryland (2007).

[33] K. Lindsay, R. Krasny, A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow, J. Comp. Phys. 172 (2001) 879–907.

[34] M. Mori, A method for evaluation of the error function of real and complex variable with high relative accuracy, Publ. RIMS, Kyoto Univ. 19 (1983) 1081–1094.

[35] J. T. Beale, A grid-based boundary integral method for elliptic problems in three dimensions, SIAM J. Numer. Anal. 42 (2004) 599–620.