Journal of Computational Physics 419 (2020) 109521

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An integral equation method for the Cahn-Hilliard equation in the wetting problem

Xiaoyu Wei [b,*], Shidong Jiang [a,2], Andreas Klöckner [b,3], Xiao-Ping Wang [c,1]

[a] Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ 07102, United States of America
[b] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States of America
[c] Department of Mathematics, the Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China

ARTICLE INFO

Article history:
Received 16 April 2019
Received in revised form 3 March 2020
Accepted 30 April 2020
Available online 8 May 2020

Keywords:
Integral equation method
Cahn-Hilliard equation
Young's angle
Convex splitting
Volume potential
Second-kind integral equation

ABSTRACT

We present an integral equation approach to solving the Cahn-Hilliard equation equipped with boundary conditions that model solid surfaces with prescribed Young's angles. The discretization of the system in time using convex splitting leads to a modified biharmonic equation at each time step. To solve it, we split the solution into a volume potential computed with free space kernels, plus the solution to a second kind integral equation (SKIE). The volume potential is evaluated with the help of a box-based volume-FMM method. For non-box domains, the source density is extended by solving a biharmonic Dirichlet problem. The near-singular boundary integrals are computed using quadrature by expansion (QBX) with FMM acceleration. Our method has linear complexity in the number of surface/volume degrees of freedom and can achieve high order convergence in space with adaptive refinement to manage error from function extension.

1. Introduction

The Cahn-Hilliard equation is frequently used in the phase field model in modeling processes involving the evolution of interfaces ([1–3]). There has been a large body of work on numerical methods for the Cahn-Hilliard equations (cf. [4–7] and references therein). Most of the existing methods are finite element/finite difference methods, or spectral methods for simple geometries. Being a fourth order partial differential equation, standard finite difference or finite element methods lead to very ill-conditioned matrices. This in turn causes low accuracy and may require preconditioning for efficient solution. Spectral methods are superior in terms of convergence order and number of unknowns in the discretized system at the cost of only being able to handle simple geometries.

In this paper, we present a high-order accurate numerical method along with an efficient solution algorithm for solving the two-phase Cahn-Hilliard equation using integral equation methods in complex geometry. At each time step, our method requires the evaluation of volume potentials on an adaptive volume grid, followed by the solution of a well-conditioned system of second kind integral equations (SKIEs) with unknown boundary densities. The volume potentials are evaluated in

* Corresponding author.
E-mail addresses: xywei@illinois.edu (X. Wei), shidong.jiang@njit.edu (S. Jiang), andreask@illinois.edu (A. Klöckner), mawang@ust.hk (X.-P. Wang).

[1] This research was supported in part by the Hong Kong RGC-GRF grants 605513 and 16302715, RGC-CRF grant C6004-14G, and NSFC-RGC joint research grant N-HKUST620/15.
[2] This research was supported by the NSF under grant DMS-1418918.
[3] This research was supported by the NSF under grant DMS-1654756.

https://doi.org/10.1016/j.jcp.2020.109521

linear complexity with respect to the number of quadrature nodes using the volume fast multipole method (FMM) based on a "box FMM" similar to, for example, [8–12]. The boundary integral equations involve weakly singular integrals, which are discretized using the quadrature-by-expansion (QBX) method [13]. The idea of QBX is to exploit the smoothness of the (smooth) layer potential away from the surface by forming locally-valid expansions which are then evaluated to compute the near or on-surface value of the potential. Finally, the resulting linear system is solved by iterative solvers such as GMRES [14] with matrix-vector products accelerated by a version of the FMM [15–18], yielding linear complexity with respect to the number of boundary nodes. When an adaptive volume mesh is used, the adaptive box structure is adjusted after each step to capture the moving interface by refining in the neighborhood of the interface and coarsening away from the interface based on solution gradient; in addition, local refinement near the domain boundary is used to achieve higher order accuracy during volume potential evaluation. The overall scheme is well conditioned, potentially high-order in space, and has asymptotically optimal complexity in terms of problem size.

The outline of the remainder of the paper is as follows. In Section 2, we present the model equations and the temporal discretization scheme. In Section 3, we analyze the static problem at each time step and provide a mathematical basis for the SKIE formulation. In Section 4, we present the SKIE formulation for the static problem. In Section 5, we present the numerical algorithms to accompany the method formulation so that the final solver is robust and has linear complexity, and in Section 6 we present some numerical results. In Section 7, we summarize the paper and discuss some future perspectives.

2. The Cahn-Hilliard equation: original problem specification

We consider a two-phase Cahn-Hilliard equation derived from a simple phase-field model given by the following Ginzburg-Landau energy functional [19]:

$$\mathcal{E}_{GL}[\phi] = \int_{\Omega} \frac{\epsilon}{2}\|\nabla\phi(\mathbf{x})\|^2 + \frac{[\phi(\mathbf{x})^2-1]^2}{4\epsilon} d\mathbf{x}, \tag{1}$$

where $\phi(\mathbf{x})$ is the composition field and ϵ represents the interface thickness. When $\Omega = \mathbb{R}^2$, taking the gradient flow of (1) in $H^{-1}(\mathbb{R}^2)$ yields the Cahn-Hilliard equation [20]:

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} = \Delta\mu(\mathbf{x},t) \qquad \forall(\mathbf{x},t)\in\Omega\times\mathbb{R}^+, \tag{2}$$

$$\mu(\mathbf{x},t) = -\epsilon\Delta\phi(\mathbf{x},t) + \frac{\phi(\mathbf{x},t)^3-\phi(\mathbf{x},t)}{\epsilon} \qquad \forall(\mathbf{x},t)\in\Omega\times\mathbb{R}^+, \tag{3}$$

where μ is the Fréchet derivative of (1), also known as the chemical potential.

In practice, we are interested in solving the initial-boundary value problem on bounded domains. The initial conditions are

$$\phi(\mathbf{x},0) = \phi_0(\mathbf{x}) \qquad \forall\mathbf{x}\in\Omega, \tag{4}$$

$$\mu(\mathbf{x},0) = -\epsilon\Delta\phi_0(\mathbf{x}) + \frac{\phi_0^3(\mathbf{x})-\phi_0(\mathbf{x})}{\epsilon} \qquad \forall\mathbf{x}\in\Omega. \tag{5}$$

Being a fourth order PDE, the Cahn-Hilliard equation requires two sets of boundary conditions. In the wetting problem, the boundary models solid surfaces, and a physically relevant way to impose boundary conditions is by adding a surface energy term to the free energy to account for the interaction with the solid wall: [21]

$$\mathcal{E}_S = \int_{\partial\Omega} \gamma(\phi(\mathbf{x}))ds_{\mathbf{x}}, \tag{6}$$

where we choose $\gamma(\phi) = (\sqrt{2}/3)\cos\theta_Y \sin((\pi/2)\phi)$, which gives the equilibrium contact angle (Young's angle) of θ_Y [22]. Re-take the gradient flow but now with the total free energy $\mathcal{F} = \mathcal{E}_{GL} + \mathcal{E}_S$, and the extra boundary energy term yields the relaxation boundary condition [23]

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} = -\epsilon\partial_n\phi(\mathbf{x},t) + \frac{\partial\gamma(\phi)}{\partial\phi}(\mathbf{x},t) \qquad \forall(\mathbf{x},t)\in\partial\Omega\times\mathbb{R}^+, \tag{7}$$

as well as the zero-flux boundary condition

$$\partial_n\mu(\mathbf{x},t) = 0 \qquad \forall(\mathbf{x},t)\in\partial\Omega\times\mathbb{R}^+, \tag{8}$$

where $\partial_n = \mathbf{n}\cdot\nabla$ with $\mathbf{n}$ being the unit outward normal vector of $\partial\Omega$.

Using Rothe's method, an energy decaying time stepping scheme for the system (2)–(5) and (7)–(8) is introduced in [24], which is based on the idea of convex splitting [25]: at the n-th time step, given $\phi^n(\mathbf{x})$, find $\phi^{n+1}(\mathbf{x})$ such that for all $\mathbf{x}$ inside the domain,

$$\frac{\phi^{n+1}(\mathbf{x})-\phi^n(\mathbf{x})}{\delta t}=\Delta\mu^{n+1}(\mathbf{x}), \tag{9}$$

$$\mu^{n+1}(\mathbf{x})=-\epsilon\,\Delta\phi^{n+1}(\mathbf{x})+\frac{s\phi^{n+1}(\mathbf{x})-(1+s)\phi^n(\mathbf{x})+(\phi^n)^3(\mathbf{x})}{\epsilon}, \tag{10}$$

and that for all $\mathbf{x}$ on the boundary,

$$\frac{\phi^{n+1}(\mathbf{x})-\phi^n(\mathbf{x})}{\delta t}=-\epsilon\,\partial_n\phi^{n+1}(\mathbf{x})+\frac{\partial\gamma}{\partial\phi}(\phi^n(\mathbf{x})), \tag{11}$$

$$\partial_n\mu^{n+1}(\mathbf{x})=0. \tag{12}$$

The splitting parameter s is chosen to be large enough such that the scheme is energy-stable (under the assumption that $\|\phi\|_\infty<\infty$, see [24] for details). If a bound M of the solution is known $\|\phi\|_{\infty,\Omega}\le M$, for all $t>0$, then letting $s\ge(3M^2-1)/2$ guarantees energy stability of the scheme; however, even though the numerical solutions mostly stay around the $[-1,1]$ range, the mathematical problem of solution boundedness for the Cahn-Hilliard equation is still open. In this paper we choose to fix $s=1.5$, which is sufficient to allow for arbitrary δt in all our tests.

It is worth noting that time discretization for the Cahn-Hilliard equation is still an active research area. In this paper, we choose to adopt the first order linearly-implicit convex-splitting method and direct our main efforts to solving the static forced modified biharmonic subproblems using an integral equation approach. Other time discretization methods include: higher-order convex splitting methods ([26,27]), stabilization methods ([7,28]), the method of invariant energy quadratization (IEQ) ([29,30]), and the recent scalar auxiliary variable (SAV) method [31]. Some of those advanced time integration schemes (like the SAV method) produce fourth-order static boundary value problems with constant coefficients, similar to the one treated in this paper. The integral equation formulation and fast algorithms in this paper can thus be straightforwardly generalized to adopt such time discretization techniques.

For convenience of deriving integral representations of the solution, we rewrite the problem (9)–(12) by collecting unknowns onto one side of the equation:

$$(\Delta^2-b\Delta+c)\phi^{n+1}(\mathbf{x})=f_1(\mathbf{x}) \qquad (\mathbf{x}\in\Omega), \tag{13}$$

$$(\Delta-b)\phi^{n+1}(\mathbf{x})+\frac{1}{\epsilon}\mu^{n+1}(\mathbf{x})=f_2(\mathbf{x}) \qquad (\mathbf{x}\in\Omega), \tag{14}$$

$$(\partial_n+c)\phi^{n+1}(\mathbf{x})=h(\mathbf{x}) \qquad (\mathbf{x}\in\partial\Omega), \tag{15}$$

$$\frac{1}{\epsilon}\partial_n\mu^{n+1}(\mathbf{x})=0 \qquad (\mathbf{x}\in\partial\Omega), \tag{16}$$

where $b=s/\epsilon^2$, $c=1/(\epsilon\delta t)$, and the inhomogeneous terms f_1, f_2 and the boundary data h are given by

$$f_1(\mathbf{x})=f_1[\phi^n(\mathbf{x})]=c\phi^n(\mathbf{x})+\Delta f_2(\mathbf{x}), \tag{17}$$

$$f_2(\mathbf{x})=f_2[\phi^n(\mathbf{x})]=\frac{(\phi^n(\mathbf{x}))^3-(1+s)\phi^n(\mathbf{x})}{\epsilon^2}, \tag{18}$$

$$h(\mathbf{x})=h[\phi^n(\mathbf{x})]=c\phi^n(\mathbf{x})+\frac{1}{\epsilon}\frac{\partial\gamma}{\partial\phi}(\phi^n(\mathbf{x})). \tag{19}$$

3. Fundamental solutions and their properties

Denote the roots of the quadratic equation $x^2-bx+c=0$ by λ_1^2 and λ_2^2. Then the fourth order operator $\Delta^2-b\Delta+c$ in (13) can be factored as $(\Delta-\lambda_1^2)(\Delta-\lambda_2^2)$. To employ an IE method for the solution of (13), we seek a fundamental solution G_0 that should solve, in a weak sense,

$$(\Delta-\lambda_1^2)(\Delta-\lambda_2^2)G_0(\mathbf{x},\mathbf{y})=\delta(\mathbf{x}-\mathbf{y}). \tag{20}$$

Motivated by the factorization structure, we recall the Green's function of the Yukawa operator that satisfies (weakly)

$$(\Delta-\lambda_i^2)G_i(\mathbf{x},\mathbf{y})=\delta(\mathbf{x}-\mathbf{y}) \tag{21}$$

and is given by the expression

$$G_i(\mathbf{x},\mathbf{y})=-\frac{1}{2\pi}K_0(\lambda_i r), \quad i=1,2, \tag{22}$$

where $r = \|x - y\|_2$ and K_0 is the modified Bessel function of the second kind of order zero (see, for example, [32]). For (20), we formulate the fundamental solution as follows:

Definition 1. The fundamental solution G_0 is given by the formula

$$G_0(\mathbf{x}, \mathbf{y}) = -\frac{1}{2\pi}\frac{1}{\lambda_1^2 - \lambda_2^2}[K_0(\lambda_1 r) - K_0(\lambda_2 r)]. \tag{23}$$

In the special case of $\lambda_1 = \lambda_2 = \lambda$, the fundamental solution G_0 is defined by taking the limit $\lambda_1 \to \lambda_2$ of (23), and is given by the formula

$$G_0(x, y) = -\frac{1}{4\pi\lambda} r K_1(\lambda r), \tag{24}$$

where K_1 is the modified Bessel function of the second kind of order one.

For simplicity of discussion, we assume that $\lambda_1^2 \neq \lambda_2^2$ for the rest for this paper. In practice, if $\lambda_1^2 = \lambda_2^2$, the equality can always be broken by slightly changing the values of the time discretization parameters s and δt.

Proposition 2. *G_0 as defined by (23) satisfies (20).*

Proof. By definition (23),

$$\begin{aligned} G_0(\mathbf{x}, \mathbf{y}) &= -\frac{1}{2\pi}\frac{1}{\lambda_1^2 - \lambda_2^2}[K_0(\lambda_1 r) - K_0(\lambda_2 r)] \\ &= \frac{1}{\lambda_1^2 - \lambda_2^2}\left[G_1(\mathbf{x}, \mathbf{y}) - G_2(\mathbf{x}, \mathbf{y})\right], \end{aligned}$$

therefore,

$$\Delta G_0(\mathbf{x}, \mathbf{y}) = \frac{1}{\lambda_1^2 - \lambda_2^2}\left[\Delta G_1(\mathbf{x}, \mathbf{y}) - \Delta G_2(\mathbf{x}, \mathbf{y})\right].$$

Using (21), we have

$$\begin{aligned} \Delta G_0(\mathbf{x}, \mathbf{y}) &= \frac{1}{\lambda_1^2 - \lambda_2^2}\left\{\left[\delta(\mathbf{x} - \mathbf{y}) + \lambda_1^2 G_1(\mathbf{x}, \mathbf{y})\right] - \left[\delta(\mathbf{x} - \mathbf{y}) + \lambda_2^2 G_2(\mathbf{x}, \mathbf{y})\right]\right\} \\ &= \frac{\lambda_1^2}{\lambda_1^2 - \lambda_2^2} G_1(\mathbf{x}, \mathbf{y}) - \frac{\lambda_2^2}{\lambda_1^2 - \lambda_2^2} G_2(\mathbf{x}, \mathbf{y}) \\ &= \lambda_1^2 G_0(\mathbf{x}, \mathbf{y}) + G_2(\mathbf{x}, \mathbf{y}) \\ &= \lambda_2^2 G_0(\mathbf{x}, \mathbf{y}) + G_1(\mathbf{x}, \mathbf{y}), \end{aligned}$$

that is,

$$(\Delta - \lambda_1^2) G_0(\mathbf{x}, \mathbf{y}) = G_2(\mathbf{x}, \mathbf{y}), \quad \text{and } (\Delta - \lambda_2^2) G_0(\mathbf{x}, \mathbf{y}) = G_1(\mathbf{x}, \mathbf{y}). \tag{25}$$

Combining (25) and (21), we have

$$(\Delta - \lambda_2^2)(\Delta - \lambda_1^2) G_0(\mathbf{x}, \mathbf{y}) = (\Delta - \lambda_1^2)(\Delta - \lambda_2^2) G_0(\mathbf{x}, \mathbf{y}) = \delta(\mathbf{x} - \mathbf{y}). \quad \square$$

Now we are equipped to define layer and volume potential operators.

Definition 3 *(Single layer potentials).* Given a density function $\sigma(\mathbf{x}) \in C(\partial\Omega)$, the single layer potential operators are

$$S_i[\sigma](\mathbf{x}) = \int_{\partial\Omega} G_i(\mathbf{x}, \mathbf{y})\sigma(\mathbf{y}) ds_{\mathbf{y}} \qquad (\mathbf{x} \in \mathbb{R}^2, i = 0, 1, 2). \tag{26}$$

Definition 4 *(Volume potentials).* Given a density function $f(\mathbf{x}) \in C(\Omega)$, the volume potential operators are

$$V_i[f](\mathbf{x}) = \int_{\Omega} G_i(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y} \qquad (\mathbf{x} \in \mathbb{R}^2, i = 0, 1, 2). \tag{27}$$

Note that here we only require the density functions to be continuous. In practice, higher regularity is desired for use of high order discretization, and our solver in this paper builds on C^1 density functions.

Regarding the jump conditions of layer potentials across the domain boundary, we have the following result:

Theorem 5 *(Jump relations). The normal derivative of $S_0[\sigma]$ has no jump across the boundary $\partial\Omega$. However, when $\mathbf{x}$ approaches a point $\mathbf{p} \in \partial\Omega$ nontangentially, the normal derivative of the single layer potential operators $S_i[\sigma]$ $(i = 1, 2)$ on $\partial\Omega$ satisfies the following jump relation:*

$$\lim_{\mathbf{x}\to\mathbf{p}^{\pm}} \frac{\partial S_i[\sigma](\mathbf{x})}{\partial \mathbf{n}_{\mathbf{p}}} = \left(\pm\frac{1}{2}I + \partial_n S_i\right)[\sigma](\mathbf{p}) \qquad (i = 1, 2), \tag{28}$$

where $\mathbf{n}_{\mathbf{p}}$ is the unit outward normal of $\partial\Omega$ at $\mathbf{p}$, and

$$\partial_n S_i[\sigma](\mathbf{p}) = \mathrm{PV}\int_{\partial\Omega} \frac{\partial G_i(\mathbf{p},\mathbf{y})}{\partial \mathbf{n}_{\mathbf{p}}}\sigma(\mathbf{y})ds_{\mathbf{y}} \qquad (i = 1, 2), \tag{29}$$

and I is the identity operator. Here $\mathbf{x} \to \mathbf{p}^{\pm}$ means that $\mathbf{x}$ approaches $\mathbf{p}$ from the exterior $(+)$ or the interior $(-)$ of the domain, respectively, and $\mathrm{PV}\cdot$ denotes the Cauchy principal value.

Proof. We expand K_0 at the origin using asymptotic formulae [32, (9.6.10)–(9.6.13)]. When $z \to 0$,

$$\begin{aligned} K_0(z) &= -\ln\frac{z}{2}\sum_{k=0}^{\infty}\frac{z^{2k}}{4^k(k!)^2} + \sum_{k=0}^{\infty}\frac{\psi(k+1)z^{2k}}{4^k(k!)^2} \\ &= -\left(\ln\frac{z}{2}+\gamma\right)\left(1+\frac{z^2}{4}+\frac{z^4}{64}+\mathcal{O}(z^6)\right)+\frac{z^2}{4}+\frac{3z^4}{128}+\mathcal{O}(z^6), \end{aligned} \tag{30}$$

where $\psi(x)$ is the *psi* (*digamma*) function defined as

$$\psi(n) = -\gamma + \sum_{k=1}^{n-1}\frac{1}{k} \qquad (n = 1, 2, 3, \dots), \tag{31}$$

and $\gamma = 0.5772156649\dots$ is Euler's constant [32].

From (30) it is clear that the leading-order non-smooth behavior results from the logarithmic term. Then (28) follows directly from the well-known jump condition for the normal derivative of the single layer potential with logarithmic kernel ([33, Lemma 3.30]). □

Using the compactness of integral operators with weakly singular kernels [34, Theorem 1.11], we also have the following corollary regarding the compactness of those single layer potential operators and their derivatives

Corollary 6 *(Compactness). For a Lipschitz domain Ω, all three single layer potential operators are compact in $C(\Omega)$. Also, the principal value parts of $\partial_n S_i[\sigma]$ $(i = 0, 1, 2)$ are all compact.*

4. Integral equation formulation

In this section, we present a second-kind integral equation formulation for the system (13)–(19). First we introduce a construction of the volume potential that reduces the problem to a pure (volume-homogeneous) boundary value problem (PBVP). Then we present an integral equation formulation for the PBVP that only requires solving a Fredholm integral equation of the second kind.

4.1. Reduction to pure boundary value problem

We choose to represent the solutions ϕ^n, μ^n in terms of sums of volume and layer potentials. We first introduce the volume potential in the terms of $\tilde{u}$ and $\tilde{v}$:

$$\phi^{n+1}(\mathbf{x}) = \tilde{u}(\mathbf{x}) + u(\mathbf{x}) \quad \text{with } \tilde{u}(\mathbf{x}) = V[\phi^n] \qquad (\mathbf{x} \in \Omega), \tag{32}$$

$$\frac{1}{\epsilon}\mu^{n+1}(\mathbf{x}) = \tilde{v}(\mathbf{x}) + v(\mathbf{x}) \quad \text{with } \tilde{v}(\mathbf{x}) = f_2(\mathbf{x}) - (\Delta - b)\tilde{u}(\mathbf{x}) \qquad (\mathbf{x} \in \Omega) \tag{33}$$

with u and v defined below, and where we require that $V[\phi^n]$ satisfies

$$(\Delta^2 - b\Delta + c)V[\phi^n](\mathbf{x}) = f_1(\mathbf{x}), \quad (\mathbf{x} \in \Omega), \tag{34}$$

and $V[\cdot]$ is a volume potential operator to be defined.

For the sole purpose of removing inhomogeneities in (13) and (14), one could simply use the volume potential

$$V_0[f_1] := \int_\Omega G_0(\mathbf{x}, \mathbf{y}) f_1(\mathbf{y}) d\mathbf{y}. \tag{35}$$

In numerical computation, however, evaluating f_1 (cf. (17)) directly is not desirable since it involves second order differentiation and thus requires $\phi^n \in C^2$, which can be restrictive. Applying Green's second identity to (35) yields

$$\begin{aligned} V_0[f_1] &= V_0[c\phi^n] - V_0[\Delta f_2] \\ &= V_0[c\phi^n] - \int_\Omega \Delta_{\mathbf{y}} G_0(\mathbf{x}, \mathbf{y}) f_2(\mathbf{y}) d\mathbf{y} \\ &\quad + \int_{\partial\Omega} \partial_{\mathbf{n_y}} G_0(\mathbf{x}, \mathbf{y}) f_2(\mathbf{y}) d\mathbf{y} - \int_{\partial\Omega} G_0(\mathbf{x}, \mathbf{y}) \partial_{\mathbf{n_y}} f_2(\mathbf{y}) d\mathbf{y} \\ &= V_0[c\phi^n] - \int_\Omega \Delta_{\mathbf{x}} G_0(\mathbf{x}, \mathbf{y}) f_2(\mathbf{y}) d\mathbf{y} \\ &\quad + \int_{\partial\Omega} \partial_{\mathbf{n_y}} G_0(\mathbf{x}, \mathbf{y}) f_2(\mathbf{y}) d\mathbf{y} - \int_{\partial\Omega} G_0(\mathbf{x}, \mathbf{y}) \partial_{\mathbf{n_y}} f_2(\mathbf{y}) d\mathbf{y}. \end{aligned}$$

Motivated by this derivation and in view of (25), we define $V[\phi^n]$ for use in (32) as

$$V[\phi^n] = cV_0[\phi^n] + V_1[f_2[\phi^n]] + \lambda_2^2 V_0[f_2[\phi^n]]. \tag{36}$$

The boundary terms have been dropped in this definition, however inserting (36) into (34) shows that the volume condition remains satisfied nonetheless. By using our construction (36) numerical differentiation of ϕ^n is completely avoided in the volume potential evaluation stage.

Clearly, one does not need to solve for but only *evaluate* $\tilde{u}$ and $\tilde{v}$. Specifically, given ϕ^n, we first evaluate $\tilde{u}$ by computing the volume integral $V[\phi^n]$ directly with the help of the box FMM. In the same FMM pass, we also obtain $\Delta\tilde{u}$ by taking derivatives of the local expansions. The loss of accuracy incurred due to numerical differentiation can be recovered by increasing the FMM order. Using $\Delta\tilde{u}$, we can then compute $\tilde{v}$ directly from its defining formula (33).

Then u and v can be found as the solutions of the following pure boundary value problem

$$(\Delta^2 - b\Delta + c)u(\mathbf{x}) = 0 \qquad (\mathbf{x} \in \Omega), \tag{37}$$

$$v + (\Delta - b)u(\mathbf{x}) = 0 \qquad (\mathbf{x} \in \Omega), \tag{38}$$

$$(\partial_n + c)u(\mathbf{x}) = g_1(\mathbf{x}) \qquad (\mathbf{x} \in \partial\Omega), \tag{39}$$

$$\partial_n v(\mathbf{x}) = g_2(\mathbf{x}) \qquad (\mathbf{x} \in \partial\Omega), \tag{40}$$

where the boundary data g_1 and g_2 are given by

$$g_1(\mathbf{x}) = h(\mathbf{x}) - \tilde{u}_n(\mathbf{x}) - c\tilde{u}(\mathbf{x}), \tag{41}$$

$$g_2(\mathbf{x}) = -\tilde{v}_n(\mathbf{x}). \tag{42}$$

4.2. Second kind integral equation formulation for the pure boundary value problem

We now derive a second kind integral equation (SKIE) formulation for the boundary value problem (37) to (42). We first represent u by the formula

$$\begin{aligned} u(\mathbf{x}) &= S_1[\sigma_1](\mathbf{x}) + S_0[\sigma_2](\mathbf{x}) \\ &= \int_{\partial\Omega} [G_1(\mathbf{x}, \mathbf{y})\sigma_1(\mathbf{y}) + G_0(\mathbf{x}, \mathbf{y})\sigma_2(\mathbf{y})] ds_{\mathbf{y}}, \end{aligned} \tag{43}$$

where σ_i $(i = 1, 2)$ are unknown densities on $\partial\Omega$. Obviously, this representation satisfies (37). Substituting (43) into (38) yields

$$\begin{aligned} v(\mathbf{x}) &= (-\Delta + b)u(\mathbf{x}) \\ &= \int_{\partial\Omega} [(-\Delta + b)G_1(\mathbf{x},\mathbf{y})\sigma_1(\mathbf{y}) + (-\Delta + b)G_0(\mathbf{x},\mathbf{y})\sigma_2(\mathbf{y})]\, ds_{\mathbf{y}}. \end{aligned} \tag{44}$$

Note that since $b = \lambda_1^2 + \lambda_2^2$, the following corollary follows naturally from the definition of the Yukawa kernels (21):

Corollary 7. *When $\mathbf{y} \neq \mathbf{x}$, $G_i(\mathbf{x},\mathbf{y})$ $(i = 1, 2)$ are both locally eigenfunctions of the Yukawa operator $\Delta - b$. Specifically,*

$$(\Delta - b)G_1(\mathbf{x},\mathbf{y}) = -\lambda_2^2 G_1(\mathbf{x},\mathbf{y}), \quad (\Delta - b)G_2(\mathbf{x},\mathbf{y}) = -\lambda_1^2 G_2(\mathbf{x},\mathbf{y}). \tag{45}$$

Proof. We only prove the first identity. The other one follows analogously. Substituting $b = \lambda_1^2 + \lambda_2^2$ into the left hand side of the identity,

$$(\Delta - b)G_1(\mathbf{x},\mathbf{y}) = [\Delta - (\lambda_1^2 + \lambda_2^2)]G_1(\mathbf{x},\mathbf{y}) = (\Delta - \lambda_1^2)G_1(\mathbf{x},\mathbf{y}) - \lambda_2^2 G_1(\mathbf{x},\mathbf{y}).$$

By definition, $(\Delta - \lambda_1^2)G_1(\mathbf{x},\mathbf{y}) = \delta(\mathbf{x},\mathbf{y}) = 0$. Thus

$$(\Delta - b)G_1(\mathbf{x},\mathbf{y}) = -\lambda_2^2 G_1(\mathbf{x},\mathbf{y}). \quad \square$$

On the other hand, (25) yields

$$\begin{aligned} (-\Delta + b)G_0(\mathbf{x},\mathbf{y}) &= [-\Delta + (\lambda_1^2 + \lambda_2^2)]G_0(\mathbf{x},\mathbf{y}) \\ &= -G_1(\mathbf{x},\mathbf{y}) + \lambda_1^2 G_0(\mathbf{x},\mathbf{y}) \\ &= -G_2(\mathbf{x},\mathbf{y}) + \lambda_2^2 G_0(\mathbf{x},\mathbf{y}). \end{aligned} \tag{46}$$

Thus

$$v(\mathbf{x}) = \int_{\partial\Omega} \left\{ \lambda_2^2 G_1(\mathbf{x},\mathbf{y})\sigma_1(\mathbf{y}) + [-G_1(\mathbf{x},\mathbf{y}) + \lambda_1^2 G_0(\mathbf{x},\mathbf{y})]\sigma_2(\mathbf{y}) \right\} ds_{\mathbf{y}}. \tag{47}$$

Combining the jump relations (28), the boundary conditions (39) and (40), and the representations (43) and (47), we obtain the following system of boundary integral equations:

$$(D + A)[\sigma](\mathbf{x}) = g(\mathbf{x}) \qquad (\mathbf{x} \in \partial\Omega), \tag{48}$$

where

$$D = -\frac{1}{2}\begin{bmatrix} 1 & 0 \\ \lambda_2^2 & -1 \end{bmatrix}, \quad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \sigma = \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix}, \quad g = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, \tag{49}$$

and the entries of the operator matrix A are given by the formulae

$$\begin{aligned} A_{11} &= \partial_n S_1 + cS_1, & A_{12} &= \partial_n S_0 + cS_0, \\ A_{21} &= \lambda_2^2 \partial_n S_1, & A_{22} &= -\partial_n S_1 + \lambda_1^2 \partial_n S_0. \end{aligned} \tag{50}$$

Clearly, D has nonzero determinant, and all entries of A are compact operators (Corollary 6). Therefore, the system (48) is a Fredholm integral equation of the second kind. When discretizing a second kind integral equation, the condition number of the resulting linear system remains bounded when the number of unknowns increases as the discretization gets refined. For an iterative solver achieving a fixed residual norm, the numerical solution thus remains accurate as the mesh is refined. Besides, with bounded condition number, the number of GMRES iterations is less likely to grow when refining the mesh.

5. Numerical realization: discretization and algorithms

For the implementation of our method, the volume potentials in (36) are evaluated using a version of the volume FMM ([8–12]). We further discretize the SKIE system (48) using the Nyström method using QBX quadrature and solve the linear system with the help of GMRES. The required matrix-vector products are carried out using the GIGAQBX quadrature-enabled fast algorithm [13,35,36]. To handle complex input geometries, we use Gmsh [37] to generate a triangulation along the domain boundary with each panel roughly of the size h_b. Over each boundary panel, Legendre-Gauss quadrature nodes of order q_b are used to form the discretization.

If $\Omega = [x_l, x_r]^2$, the volume FMM connects seamlessly with the boundary discretization; however, additional treatment is needed when dealing with general geometries, as the volume FMM requires box-shaped domains to operate. In this section, we present those extra steps necessary to achieve flexibility in handling complex geometries.

5.1. Regularity and non-square domains

For non-square domains, the volume potentials are evaluated over a box B that encloses the physical domain such that $\Omega \subset B$. Use of the box FMM requires that volume source densities be extended to $B \setminus \Omega$. Noting that f_1, f_2 are all algebraic functions of ϕ^n, we only need to extend ϕ to $B \setminus \Omega$. There are several factors that need to be considered when deciding how to extend ϕ:

- The smoother the extension is, the higher accuracy in the order of approximation for those densities can be achieved using (piecewise) polynomials, yielding better overall accuracy.
- Higher order extensions are more costly to evaluate. Typically for C^k extension one needs to solve a PDE where $k+1$ boundary conditions can be imposed, e.g., $\Delta^{k+1} u = 0$. It is natural to ask that the extension process does not cost much more than solving the original problem, i.e., $k \leq 1$.
- Our SKIE formulation poses certain minimum regularity requirements. Specifically, evaluation of g_2 in (40) requires at minimum $\tilde{v} \in C^1(B)$, which then requires $\tilde{u}$ to be in $C^3(B)$.

To satisfy the condition $\tilde{u} \in C^3(B)$, the density ϕ should satisfy certain regularity requirements. First we note the standard regularity result for the Poisson equation [38]:

Theorem 8. *Let $\Omega \subset \mathbb{R}^d$ be open and bounded,*

$$u(x) := \int_{\Omega} \Phi(\mathbf{x} - \mathbf{y}) f(\mathbf{y}) d\mathbf{y}, \tag{51}$$

where Φ is the fundamental solution. Then if $f \in C_0^{\alpha}(\overline{\Omega})$, $0 < \alpha < 1$, then $u \in C^{2,\alpha}(\overline{\Omega})$, and

$$\|u(x)\|_{C^{2,\alpha}(\overline{\Omega})} \leq c \|f\|_{C^{\alpha}(\overline{\Omega})}. \tag{52}$$

The leading order singular (least regular) term in our construction of $V[\phi^n]$ is $V_1[f_2]$, for which we have the following result:

Proposition 9. *If $f_2 \in C^1(B)$, the volume potential $V_1[f_2] \in C^3(B)$.*

Proof. Since, like the Poisson kernel, the kernel G_1 also has a leading-order logarithmic singularity, Theorem 8 also applies to $V_1[f_2]$. Combined with the assumption that the source density is in $C^1(B)$, the proposition follows. □

Based on these considerations, we choose to use C^1 extensions. To obtain such an extension, we assume that Ω is simply connected and solve the following Stokes problem:

$$-\Delta \mathbf{u} + \nabla p = 0 \qquad \text{in } \mathbb{R}^2 \setminus \Omega, \tag{53}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \mathbb{R}^2 \setminus \Omega, \tag{54}$$

$$\mathbf{u} = \nabla^{\perp} \phi \qquad \text{on } \partial\Omega, \tag{55}$$

where $\mathbf{u}$ is the velocity field of the fluid and p is the pressure field. The Stokes problem (53), (54), (55) can also be solved via an SKIE formulation [39]. We evaluate the stream function ω up to a constant by evaluating complex layer potential representations in [40], such that

$$\mathbf{u} = \nabla^{\perp} \omega = \begin{bmatrix} \frac{\partial \omega}{\partial x_2} \\ -\frac{\partial \omega}{\partial x_1} \end{bmatrix}, \tag{56}$$

then ω is a biharmonic function that solves the boundary value problem

$$\Delta^2 \omega = 0, \qquad \text{in } \mathbb{R}^2 \setminus \Omega, \tag{57}$$

$$\nabla \omega = \nabla \phi, \qquad \text{on } \partial\Omega. \tag{58}$$

Since $\partial\Omega$ is simply connected, we only need to add a constant to ω to obtain a C^1 extension of ϕ:

$$\phi(\mathbf{x}) = \omega(\mathbf{x}) + \left(\int_{\partial\Omega} \phi(\mathbf{y}) ds_{\mathbf{y}} - \int_{\partial\Omega} \omega(\mathbf{y}) ds_{\mathbf{y}} \right) \qquad (\mathbf{x} \in \mathbb{R}^2 \setminus \Omega). \tag{59}$$

Remark 10. The assumption of Ω being simply connected is for simplicity of the formulation and not critical. If Ω is topologically more complicated, methods in [41,40] can still be used for performing C^1 extension.

5.2. Boundary layers

From matched asymptotic analysis of the Cahn-Hilliard equation (2)–(8) [23], it is apparent that for very small ϵ, the solution can develop boundary layers.

Boundary layers cause stability concerns for our scheme using C^1 extension. Since the exterior biharmonic problem does not admit a Laplace-type maximum principle, the presence of the boundary layer causes the extended density to be very large in magnitude compared to the density inside the domain, resulting in lost accuracy after each time step. As a result, the numerical solution may not remain bounded after several time steps and lose energy stability.

Dealing with such boundary layers numerically is the subject of ongoing research. Using an adaptive volume mesh that refines towards the boundary can be helpful. Instead of evaluating the volume potentials over one single bounding box B, we can choose B to be union of a set of boxes that covers Ω. By reducing the Hausdorff distance between ∂B and $\partial\Omega$, the extended density is less likely to blow up. However, such refinement is very expensive in practice, and there is no guarantee that the boundary layer is resolved a-priori since a reliable refinement criterion is not known.

For long-time simulations, we provide an alternative formulation which is a less costly approximation based on the total energy of the system. When solving the SKIE (48), we use a homogeneous right hand side for the second equation $g_2 = 0$, which effectively makes the layer potential u satisfy the contact angle dynamics, while ignoring the second boundary condition. After solving for σ_1 and evaluating u, we add a constant to u such that the integral of ϕ over the whole domain is conserved. This approach conserves the total mass and keeps the numerical solution bounded at all time.

Formally, the approximation replaces the representation (43) with the following stabilized representation:

$$u(\mathbf{x}) = S_1[\sigma_1](\mathbf{x}) + C \tag{60}$$

where C is the added constant. The net effect of the approximation spreads the influence of boundary layers across the whole domain. Long-time simulations using the stabilized representation are performed and the results are in line with those obtained from finite element method. Although the modification seems crude, it maintains the most relevant solution characteristics, including:

- H^{-1} gradient flow structure in the bulk region, since the volume potential is unchanged.
- Contact angle dynamics from the first boundary condition.
- Integral conservation over the whole domain.

6. Numerical results

6.1. Spatial convergence

Based on error analysis of the volume FMM ([9]), QBX ([42]), the GIGAQBX fast algorithm ([13], [36]) and standard approximation theory ([43]), we expect the following error estimate for the overall numerical accuracy of our scheme:

Heuristic 1 *(Numerical accuracy). Assuming density extension has regularity $\phi^0 \in C^k(B)$, denote the numerical solution at $t = \delta t$ by ϕ_h^1 and its relative error by $E_0 := \|\phi_h^1 - \phi(\delta t)\|_\infty / \|\phi(\delta t)\|_\infty$. Asymptotically, for a single time step, we have*

$$E_0 \le C_0 \delta t + (E_v + E_b)\,,$$

where C_0 is a constant independent of δ_t, E_v includes the errors from volume potential evaluation and E_b includes the errors from the BIE solution.

1. *For E_v, we have*

$$E_v \le \underbrace{C_1 h_v^{\min(\sqrt{q_v},k+1)}}_{\text{discretization error}} + \underbrace{C_2 2^{-m_v}}_{\text{volume FMM error}} + \xi,$$

   where h_v is the volume mesh size, q_v is the number of quadrature points per volume cell, m_v is the volume FMM order, ξ is the error of near-field volume potentials (controlled by the tolerance of adaptive quadrature over nearfield boxes), and C_1, C_2 are constants independent of h_v and ξ.
2. *For E_b, assuming that the solution density σ is smooth and that its Hölder norms $\|\sigma\|_{C^{s,\beta}}$ for $s \in \mathbb{N}$, $\beta \in [0, 1)$ can be controlled with $\|\phi(\delta t)\|_\infty$, we have*

$$E_b \le \underbrace{C_3 h_b^{q_b}}_{\text{discretization error}} + \underbrace{C_4 2^{-2q_b}}_{\text{QBX quadrature error}} + \underbrace{C_5 h_b^{p+1} 2^{-p}}_{\text{QBX truncation error}} + \underbrace{C_6 2^{-m_b}}_{\text{QBX FMM error}} + C_7 \eta,$$

Table 1
Results using first-order approximation (q_b: number of quadrature points per boundary cell, q_v: number of quadrature points per volume cell, e_0: relative error at $t = \delta t$).

q_b	q_v	δx	e_0	EOC
1	1	1.56×10^{-2}	9.91×10^{-1}	–
1	1	7.81×10^{-3}	6.72×10^{-1}	0.56
1	1	3.91×10^{-3}	2.91×10^{-1}	1.21
1	1	1.95×10^{-3}	1.34×10^{-1}	1.19
1	1	9.77×10^{-4}	9.02×10^{-2}	0.57
1	1	4.88×10^{-4}	5.24×10^{-2}	0.78

Table 2
Results using second-order approximation (q_b: number of quadrature points per boundary cell, q_v: number of quadrature points per volume cell, e_0: relative error at $t = \delta t$).

q_b	q_v	δx	e_0	EOC
2	4	1.56×10^{-2}	4.01×10^{-1}	–
2	4	7.81×10^{-3}	2.41×10^{-1}	0.73
2	4	3.91×10^{-3}	6.99×10^{-2}	1.79
2	4	1.95×10^{-3}	1.76×10^{-2}	1.99
2	4	9.77×10^{-4}	4.84×10^{-3}	1.86

where h_b is the boundary mesh size, q_b is the number of quadrature points per boundary cell, p is the QBX order, m_b is the FMM order used in GIGAQBX, η is the error of the linear solver (controlled by the tolerance of GMRES), and C_3, C_4, C_5, C_6, C_7 are constants independent of h_b and η.

Among all the error sources listed in Heuristic 1, we are most interested in the discretization errors that decay algebraically with mesh refinement. In this section, we describe experiments to verify this expected error decay, by choosing other parameters of the scheme so as to minimize the contributions of their corresponding errors. Specifically, we fix the tolerance for GMRES to be 10^{-14}, and the tolerance for (precomputed) near-field adaptive quadrature in the volume FMM to be 10^{-13}. Also, we set the FMM orders and the QBX order to be sufficiently high. Specifically, we let the QBX order match the boundary quadrature order $q_b = p + 1$, and fix the volume FMM order and the GIGAQBX FMM order $m_v = m_b = 8$. We use two-dimensional Taylor expansions for the FMM, thus achieving FMM order 8 in 2D requires $\frac{1}{2} \times 8 \times 9 = 36$ terms. To back our choices, neither increasing the QBX order p up to 10 nor the FMM order m_v, m_b up to 10 yields obvious effects on the results, indicating that the error in the computation of the volume potential is dominated by discretization errors. The same characteristic mesh size $h_b = h_v = \delta x$ is used for volume and boundary meshes. For simplicity, we also keep the orders of volume and boundary discretizations the same.

To gauge convergence, we fix $\delta t = 10^{-2}$ and perform self-convergence tests by refining both spatial grids successively and calculating the relative error from the result of one time step $e_0 := \|\phi_h^1 - \phi_0^1\|_\infty / \|\phi_h^1\|_\infty$, where ϕ_0^1 is the reference solution. The tests are performed on a circular domain with radius 0.247 centered at the origin. The bounding box is chosen to be $B = [-0.25, 0.25]^2$. The initial condition is

$$\phi^0(\mathbf{x}) = \tanh\left[\frac{10}{\sqrt{2\epsilon}}(|\mathbf{x} \cdot \mathbf{e}_1| - 0.1)\right], \tag{61}$$

where $\epsilon = 10^{-2}$, and $\mathbf{e}_1 = (1, 0)$.

In this work, since we are using C^1 extension of ϕ, based on the estimates above, when the discretization error dominates, the total error of the scheme should scale as $\mathcal{O}(h_b^{q_b}) + \mathcal{O}(h_v^{\min(\sqrt{q_v}, 2)})$ when $h_b, h_v \to 0$. In particular, when $h_b = h_v = \delta x$, and $q_b = \sqrt{q_v} = r$, the total discretization error is expected to be $\mathcal{O}(\delta x^r)$ as $\delta x \to 0$.

When using piecewise constant approximations for both boundary and volume discretizations, the results are shown in Table 1, where q_b, q_v stand for the number of quadrature points per cell on the boundary mesh and the volume mesh, respectively. The convergence order is calculated using values of the error measure e_0 from the current row and the row above. The same statistics for the second order discretizations (i.e. piecewise linear approximations on the boundary mesh and piecewise bi-linear approximations on the volume mesh) are shown in Table 2.

In both cases, the numerical results align well with expectations. When $r = 1$, $q_b = q_v = 1$, discretizations for both boundary and volume densities are piecewise constant (first order), and Table 1 demonstrates empirical convergence order close to 1. When $r = 2$, $q_b = 2, q_v = 4$, boundary densities are discretized with piecewise linear functions, and volume densities are discretized with piecewise bilinear functions, in which case Table 2 shows empirical convergence order close to 2.

When higher-order volume approximations are used with a uniform volume mesh, the convergence order will be limited to 2 due to the volume densities being limited to only C^1 at the boundary. For example, as shown in Table 3.

Table 3
Results using third-order approximation, without extra boundary refinement (q_b: number of quadrature points per boundary cell, q_v: number of quadrature points per volume cell, e_0: relative error at $t = \delta t$).

q_b	q_v	δx	e_0	EOC
3	9	1.56×10^{-2}	2.11×10^{-1}	–
3	9	7.81×10^{-3}	3.73×10^{-2}	2.50
3	9	3.91×10^{-3}	1.04×10^{-2}	1.84
3	9	1.95×10^{-3}	2.36×10^{-3}	2.13
3	9	9.77×10^{-4}	5.58×10^{-4}	2.08

Table 4
Results using third-order approximation, with extra boundary refinement (q_b: number of quadrature points per boundary cell, q_v: number of quadrature points per volume cell, e_0: relative error at $t = \delta t$).

q_b	q_v	δx	e_0	EOC
3	9	1.56×10^{-2}	2.11×10^{-1}	–
3	9	7.81×10^{-3}	3.45×10^{-2}	2.61
3	9	3.91×10^{-3}	4.42×10^{-3}	2.96
3	9	1.95×10^{-3}	6.99×10^{-4}	2.66
3	9	9.77×10^{-4}	1.04×10^{-4}	2.75

Table 5
Temporal convergence tests (δt: time step size, e_f: relative error at final time).

δt	e_f	EOC
0.1	2.92×10^{-1}	–
0.05	1.42×10^{-1}	1.04
0.025	7.53×10^{-2}	0.92

However, higher order approximations can still be used by adding extra levels of refinement to volume cells that intersect $\partial\Omega$. To guide the mesh refinement process, we form a two-dimensional tensor-product Legendre series of degree q_v (using the discrete tensor-product Legendre transform) over each cell, and compute the sum of the absolute values of the $q_v + 1$ coefficients of the terms of total degree q_v as the error indicator for that cell. To make the adaptive mesh for the box FMM, we start from a uniform mesh with prescribed δx, and iteratively perform refinement cycles as follows:

1. Compute the average values of the error indicators for the interior cells as M_i, and that for the boundary-intersecting cells as M_b.
2. Refine each boundary-intersecting cell by one level (into four cells).
3. Refine some additional cells as necessary to maintain a radius ratio of at most 2 between adjacent boxes.

We repeat the refinement cycles until $M_b \leq M_i$. As shown by the results in Table 4, since the source density is smooth away from the boundary, by performing such refinements, the solver is able to take advantage of higher order of convergence in the bulk region where the source density is smooth.

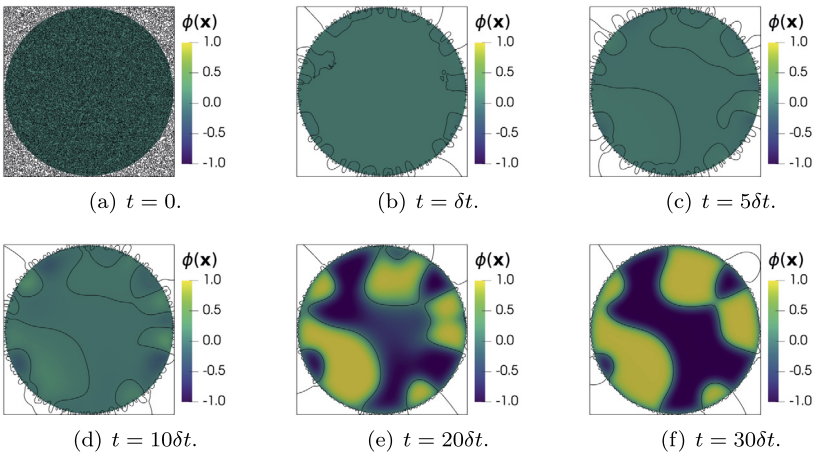
6.2. Temporal convergence

In terms of temporal convergence, we expect to have first order in time due to use of the convex splitting scheme of [24]. We once again use the initial condition (61) over a disk domain to perform this test. To verify the expectation, we fix $\epsilon = 0.5$ and the final time $t_f = 0.1$, and perform self-convergence tests by taking different δt. The numerical error is measured at $t = t_f$ using the formula $e_f := \|\phi_h^f - \phi_0^f\|_\infty / \|\phi_h^f\|_\infty$, against the reference solution obtained by taking $\delta t = 0.0125$. The results are presented in Table 5, confirming that the scheme is of first order in time.

6.3. Short-time dynamics

In this example, we test for the ability of our scheme to resolve small structures using high order approximation in combination with adaptive mesh refinement. We mimic, using the initial condition, a near-uniform state with small random perturbations. Those perturbations will lead to a phase separation process. We expect that our scheme is able to resolve the dynamics of structures at all scales during the phase separation.

The setup tries to capture short-time dynamics of the Cahn-Hilliard equation. We choose $\epsilon = 10^{-2}$, $\delta t = 10^{-4}$, and use fourth-order quadrature rules for volume and boundary discretizations. For volume densities,



(a) $t = 0$. (b) $t = \delta t$. (c) $t = 5\delta t$.

(d) $t = 10\delta t$. (e) $t = 20\delta t$. (f) $t = 30\delta t$.

Fig. 1. Short-time dynamics of the Cahn-Hilliard equation ($\delta t = 10^{-4}$).

$$\phi(x, y) \approx \sum_{0 \le k,l \le 3} a_{k,l} P_k(x) P_l(y), \tag{62}$$

(where P_k is the Legendre polynomial of degree k defined on a cell). Like 6.1, we use

$$E := \sum_{0 \le k \le 3} |a_{k,3-k}| \tag{63}$$

as the error indicator for the cell for mesh adaptation purposes.

The simulation starts with an initial profile generated using the following procedure:

1. A uniform 200×200 Cartesian grid is generated on B. Then for each grid point a real number is drawn from a uniform random distribution on $[-10^{-3}, 10^{-3}]$.
2. A cubic B-spline representation of the initial ϕ is formed using the Cartesian grid points as knots and the random numbers as interpolating values.
3. The values and derivatives of the initial ϕ are then evaluated using the cubic B-spline representation on the quadrature points of the box mesh as well as the boundary mesh.

Numerical solutions are shown in Fig. 1. The first 3 time steps are performed solving the full SKIE for the boundary integral equation to capture the fast dynamics near the boundary, while the later steps are performed using the approximation (60). In our experience, the numerical solution tends to form boundary layers in regions away from contact lines (where its tangential derivative is close to 0). The color maps are shown for each sub-figure individually, while the contours are all showing the zero level set of ϕ over the whole bounding box. The short-time dynamics can be understood as interplay of three mechanisms: surface-energy driven phase transition, bulk-energy driven spinodal decomposition, and solid boundary induced nucleation. In later time steps, the smoothing term in the free energy starts to play a more important role and leads to coarsening of the patterns. As a benefit of the high-order accuracy of our scheme, our method is able to resolve the interfaces as well as the small patterns formed on the boundary very well.

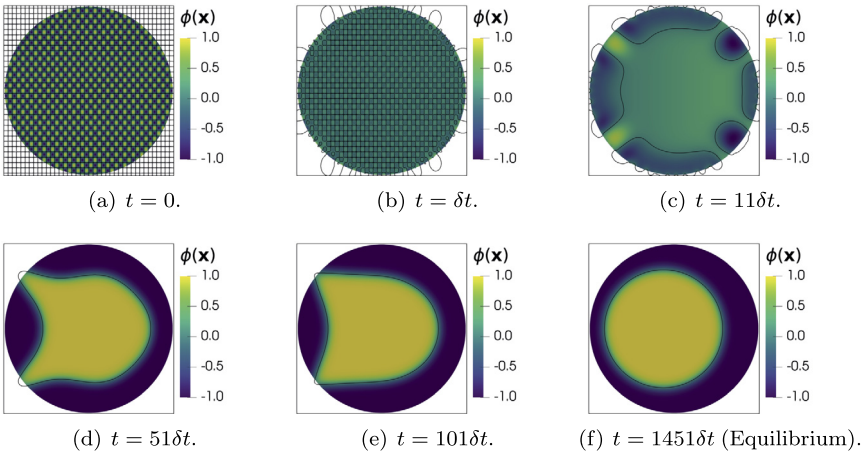
6.4. Long-time dynamics

Since our solver is only first-order in time, choosing larger time step sizes reduces solution accuracy significantly; however, through this example we demonstrate that even with large time steps, higher spatial order is still useful for preserving certain qualitative properties of the solution. In this example, we verify that the solver can preserve solution symmetry across long time scales.

We still use $\epsilon = 10^{-2}$, but take larger time step size $\delta t = 0.5$. The initial condition is

$$\phi^0(x, y) = \sin\left(\frac{40\pi}{L}x\right)\cos\left(\frac{32\pi}{L}y\right), \tag{64}$$

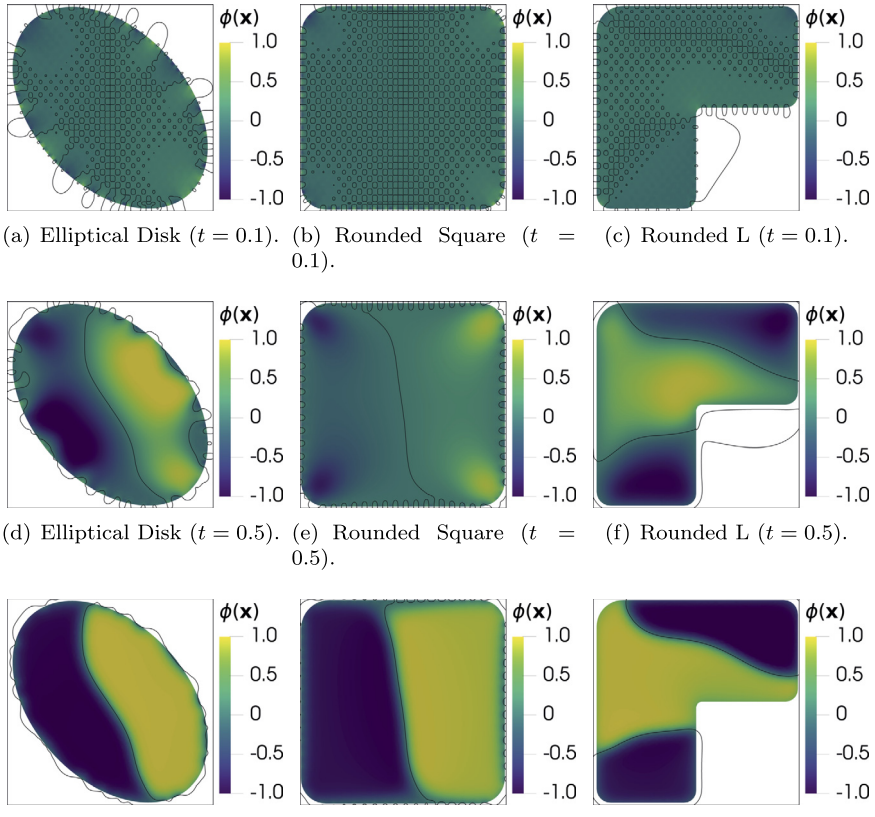
where $L = 0.5$ is the size of the bounding box. Note that $\phi^0(x, -y) = \phi^0(x, y)$. For this example we use the approximation (60) from the start. Since δt is rather large compared to δx, the temporal truncation error of the convex splitting scheme dominates; however, we expect that the scheme is still stable and reaches an equilibrium state when $t \to \infty$, and that the solution ϕ at any time stays symmetric with respect to the x axis.



(a) $t = 0$. (b) $t = \delta t$. (c) $t = 11\delta t$.

(d) $t = 51\delta t$. (e) $t = 101\delta t$. (f) $t = 1451\delta t$ (Equilibrium).

Fig. 2. Long-time dynamics of the Cahn-Hilliard equation ($\delta t = 0.5$).



(a) Elliptical Disk ($t = 0.1$). (b) Rounded Square ($t = 0.1$). (c) Rounded L ($t = 0.1$).

(d) Elliptical Disk ($t = 0.5$). (e) Rounded Square ($t = 0.5$). (f) Rounded L ($t = 0.5$).

(g) Elliptical Disk ($t = 1$). (h) Rounded Square ($t = 1$). (i) Rounded L ($t = 1$).

Fig. 3. Numerical solutions for complex geometries.

Numerical results are shown in Fig. 2. The simulation stops at $t = 780$ when $\|\phi^{n+1} - \phi^n\|_\infty < 10^{-14}$, which confirms that the numerical solution reaches an equilibrium state. Besides being able to resolve the interface and the contact lines, our method is also shown to be able to properly resolve the interactions between the contact lines and the geometry with no alignment requirements between the two sets of meshes: in the case circular domain, as long as both the boundary mesh and the box mesh are symmetric, the solution symmetry will be well-preserved from the beginning to the equilibrium state.

Then, we demonstrate the ability of our scheme to handle moderately complex smooth geometries. We report the numerical solution at $t = 1$ for the same problem setup as the experiment underlying Fig. 2, but with different physical domains. The results are shown in Fig. 3.

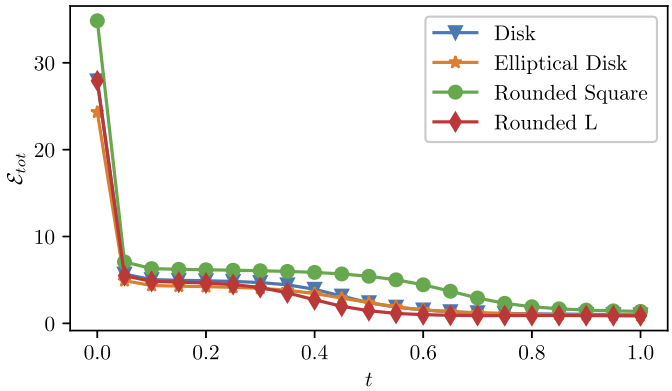


Fig. 4. Total energy of the numerical solutions.

Table 6
Condition numbers of FEM matrices (2nd order).

δx	$N_{DOFs}(FEM)$	$\kappa(FEM)$
1.11×10^{-1}	16	9.79×10^{3}
6.26×10^{-2}	50	1.24×10^{4}
3.32×10^{-2}	178	2.15×10^{4}
1.71×10^{-2}	674	6.82×10^{4}
8.64×10^{-3}	2626	1.12×10^{5}

Table 7
Condition numbers of IEM matrices (4th order).

δx	$N_{DOFs}(IEM)$	$\kappa(IEM)$
1.11×10^{-1}	56	9.86×10^{2}
6.26×10^{-2}	100	9.87×10^{2}
3.32×10^{-2}	200	9.88×10^{2}
1.71×10^{-2}	364	9.88×10^{2}
8.64×10^{-3}	720	9.89×10^{2}

The total energy of the numerical solution for different geometries is shown in Fig. 4. According to the theory, the energy decaying property is preserved as long as the numerical solution remains bounded. With the stabilization applied, we observe that the numerical solution remains bounded for various geometries.

6.5. Comparisons with FEM

It is a natural question to ask how the integral equation approach that we take in this article compares to more popular finite element methods. In this example we aim to provide some perspective on this by presenting a comparison between our solver and a simple finite element solver.

The finite element solver we consider is based on a mixed formulation for the time-discretized system (9)–(12), using Q^1 elements for both ϕ and μ. We report the condition number and number of degrees of freedom with their dependence on δx for FEM in Table 6, where δx is calculated as the square root of average cell measure, and the same statistics for IEM in Table 7. The tests are performed over a circular domain with radius 0.247. In all tests with both the FEM and the IEM, we fix $\epsilon = 0.5$, $\theta_Y = \pi/3$ and $\delta t = 1$. The rest of the parameters for the IEM are the same as in 6.1. The data confirms that, being a second-kind integral equation, the linear system has bounded condition number that does not scale with mesh size, unlike the FEM whose linear system's condition number grows in an unbounded manner when refining the mesh. In addition, when $\delta x \to 0$, the number of unknowns in the linear solve scales as $\mathcal{O}(1/\delta x)$ for our method, while it scales as $\mathcal{O}(1/\delta x^2)$ for the FEM.

7. Conclusions

We have introduced an integral equation method for the Cahn-Hilliard equation in bounded 2D domains with solid boundary conditions. For each time step, the method consists of two stages: (1) evaluate volume potentials and (2) solve a second-kind integral equation to match the boundary conditions. Fast multipole based fast algorithms are used to attain linear complexity for both stages. To handle complex geometry, a C^1 extension of ϕ onto a larger domain is performed by solving an additional SKIE. Through numerical experiments, the method is shown to require asymptotically fewer degrees of

freedom in the solution of the relevant linear systems than a finite element method while maintaining a condition number bounded independent of mesh resolution. To remedy some of the numerical stability concerns for long-time simulations caused by boundary layers, we have proposed a stabilized representation approximating the original problem, and we have validated the approximation with numerical results.

The present work opens some exciting future perspectives. We will seek to apply the same techniques in this paper to second-order time discretization schemes, for example the SAV method [31] offering higher accuracy order in time as well as superior energy stability. In addition, the challenges posed by boundary layers are by no means unique to Cahn-Hilliard equations, and it is of great interest to develop fast volume potential evaluators that remain stable and accurate when the source density develops boundary layers. Once the boundary layer issue can be handled, our SKIE formulation will be immediately applicable to the full long-time simulation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported in part by the National Science Foundation under grants DMS-1418961 and DMS-1654756, and by the Hong Kong Research Grant Council (RGC-GRF grants 605513 and 16302715, RGC-CRF grant C6004-14G, and NSFC-RGC joint research grant N-HKUST620/15). Any opinions, findings, and conclusions, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Hong Kong Research Grant Council; Neither NSF nor HKRGC has approved or endorsed its content. Part of the work was performed while the authors were participating in the HKUST-ICERM workshop 'Integral Equation Methods, Fast Algorithms and Their Applications to Fluid Dynamics and Materials Science' held in 2017.

References

[1] D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modeling, J. Comput. Phys. 155 (1) (1999) 96–127, https://doi.org/10.1006/jcph.1999.6332.
[2] C. Liu, J. Shen, A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method, Physica D 179 (3) (2003) 211–228, https://doi.org/10.1016/S0167-2789(03)00030-7.
[3] L.-Q. Chen, Phase-field models for microstructure evolution, Annu. Rev. Mater. Res. 32 (1) (2002) 113–140, https://doi.org/10.1146/annurev.matsci.32.112001.132041.
[4] J. Barrett, J. Blowey, Finite element approximation of the Cahn-Hilliard equation with concentration dependent mobility, Math. Comp. Am. Math. Soc. 68 (226) (1999) 487–517, https://doi.org/10.1090/S0025-5718-99-01015-7.
[5] E.V.L. de Mello, O. Teixeira da Silveira Filho, Numerical study of the Cahn–Hilliard equation in one, two and three dimensions, Physica A 347 (2005) 429–443, https://doi.org/10.1016/j.physa.2004.08.076.
[6] W.M. Feng, P. Yu, S.Y. Hu, Z.K. Liu, Q. Du, L.Q. Chen, A Fourier spectral moving mesh method for the Cahn-Hilliard equation with elasticity, Commun. Comput. Phys. 5 (2–4) (2009) 582–599.
[7] J. Shen, X. Yang, Numerical approximations of Allen-Cahn and Cahn-Hilliard equations, Discrete Contin. Dyn. Syst. 28 (A2010) (2010) 1669–1691, https://doi.org/10.3934/dcds.2010.28.1669.
[8] H. Cheng, J. Huang, T.J. Leiterman, An adaptive fast solver for the modified Helmholtz equation in two dimensions, J. Comput. Phys. 211 (2) (2006) 616–637, https://doi.org/10.1016/j.jcp.2005.06.006.
[9] F. Ethridge, L. Greengard, A new fast-multipole accelerated Poisson solver in two dimensions, SIAM J. Sci. Comput. 23 (3) (2001) 741–760, https://doi.org/10.1137/S1064827500369967.
[10] H. Langston, L. Greengard, D. Zorin, A free-space adaptive FMM-based PDE solver in three dimensions, Commun. Appl. Math. Comput. Sci. 6 (1) (2011) 79–122, https://doi.org/10.2140/camcos.2011.6.79.
[11] D. Malhotra, G. Biros, PVFMM: a parallel kernel independent FMM for particle and volume potentials, Commun. Comput. Phys. 18 (3) (2015) 808–830, https://doi.org/10.4208/cicp.020215.150515sw.
[12] A. Gholami, D. Malhotra, H. Sundar, G. Biros FFT FMM, Or multigrid? A comparative study of state-of-the-art Poisson solvers for uniform and nonuniform grids in the unit cube, SIAM J. Sci. Comput. 38 (3) (2016) C280–C306, https://doi.org/10.1137/15M1010798.
[13] A. Klöckner, A. Barnett, L. Greengard, M. O'Neil, Quadrature by expansion: a new method for the evaluation of layer potentials, J. Comput. Phys. 252 (2013) 332–349, https://doi.org/10.1016/j.jcp.2013.06.027.
[14] Y. Saad, M. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869, https://doi.org/10.1137/0907058.
[15] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (2) (1987) 325–348, https://doi.org/10.1016/0021-9991(87)90140-9.
[16] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, J. Comput. Phys. 155 (2) (1999) 468–498, https://doi.org/10.1006/jcph.1999.6355.
[17] Z. Gimbutas, V. Rokhlin, A generalized fast multipole method for nonoscillatory kernels, SIAM J. Sci. Comput. 24 (3) (2003) 796–817, https://doi.org/10.1137/S1064827500381148.
[18] L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole algorithm in two and three dimensions, J. Comput. Phys. 196 (2) (2004) 591–626, https://doi.org/10.1016/j.jcp.2003.11.021.
[19] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, J. Chem. Phys. 28 (2) (1958) 258–267, https://doi.org/10.1063/1.1744102.
[20] J.W. Cahn, J.E. Hilliard, Spinodal decomposition: a reprise, Acta Metall. 19 (2) (1971) 151–161, https://doi.org/10.1016/0001-6160(71)90127-1.
[21] L. Modica, Gradient theory of phase transitions with boundary contact energy, Ann. Inst. Henri Poincare (C) Non Linear Anal. 4 (5) (1987) 487–512, https://doi.org/10.1016/S0294-1449(16)30360-2.

[22] X. Xu, X. Wang, Derivation of the Wenzel and Cassie equations from a phase field model for two phase flow on rough surface, SIAM J. Appl. Math. 70 (8) (2010) 2929–2941, https://doi.org/10.1137/090775828.
[23] X. Chen, X. Wang, X. Xu, Analysis of the Cahn–Hilliard equation with a relaxation boundary condition modeling the contact angle dynamics, Arch. Ration. Mech. Anal. 213 (1) (2014) 1–24, https://doi.org/10.1007/s00205-013-0713-x.
[24] M. Gao, X.-P. Wang, A gradient stable scheme for a phase field model for the moving contact line problem, J. Comput. Phys. 231 (4) (2012) 1372–1386, https://doi.org/10.1016/j.jcp.2011.10.015.
[25] D.J. Eyre, An unconditionally stable one-step scheme for gradient systems, 1997, pp. 1–15, unpublished article.
[26] A. Baskaran, J.S. Lowengrub, C. Wang, S.M. Wise, Convergence analysis of a second order convex splitting scheme for the modified phase field crystal equation, SIAM J. Numer. Anal. 51 (5) (2013) 2851–2873, https://doi.org/10.1137/120880677.
[27] J. Shen, C. Wang, X. Wang, S. Wise, Second-order convex splitting schemes for gradient flows with Ehrlich–Schwoebel type energy: application to thin film epitaxy, SIAM J. Numer. Anal. 50 (1) (2012) 105–125, https://doi.org/10.1137/110822839.
[28] J. Zhu, L.-Q. Chen, J. Shen, V. Tikare, Coarsening kinetics from a variable-mobility Cahn-Hilliard equation: application of a semi-implicit Fourier spectral method, Phys. Rev. E 60 (4) (1999) 3564–3572, https://doi.org/10.1103/PhysRevE.60.3564.
[29] F. GuilléN-GonzáLez, G. Tierra, On linear schemes for a Cahn-Hilliard diffuse interface model, J. Comput. Phys. 234 (2013) 140–171, https://doi.org/10.1016/j.jcp.2012.09.020.
[30] X. Yang, Linear, first and second-order, unconditionally energy stable numerical schemes for the phase field model of homopolymer blends, J. Comput. Phys. 327 (2016) 294–316, https://doi.org/10.1016/j.jcp.2016.09.029.
[31] J. Shen, J. Xu, J. Yang, The scalar auxiliary variable (SAV) approach for gradient flows, J. Comput. Phys. 353 (2018) 407–416, https://doi.org/10.1016/j.jcp.2017.10.021.
[32] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables, Courier Corporation, 1965.
[33] G.B. Folland, Introduction to Partial Differential Equations, Princeton University Press, 1976.
[34] D. Colton, R. Kress, Integral Equation Methods in Scattering Theory, Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, 2013.
[35] M. Rachh, A. Klöckner, M. O'Neil, Fast algorithms for quadrature by expansion I: globally valid expansions, J. Comput. Phys. 345 (2017) 706–731, https://doi.org/10.1016/j.jcp.2017.04.062.
[36] M. Wala, A. Klöckner, A fast algorithm with error bounds for quadrature by expansion, J. Comput. Phys. 374 (2018) 135–162, https://doi.org/10.1016/j.jcp.2018.05.006.
[37] C. Geuzaine, J.-F. Remacle Gmsh, A 3-D finite element mesh generator with built-in pre- and post-processing facilities, Int. J. Numer. Methods Eng. 79 (11) (2009) 1309–1331, https://doi.org/10.1002/nme.2579.
[38] D. Gilbarg, N.S. Trudinger, Elliptic Partial Differential Equations of Second Order, 2nd edition, Classics in Mathematics, Springer-Verlag, Berlin, Heidelberg, 2001.
[39] L. Greengard, M.C. Kropinski, A. Mayo, Integral equation methods for Stokes flow and isotropic elasticity in the plane, J. Comput. Phys. 125 (2) (1996) 403–414, https://doi.org/10.1006/jcph.1996.0102.
[40] M. Rachh, T. Askham, Integral equation formulation of the biharmonic Dirichlet problem, J. Sci. Comput. 75 (2) (2018) 762–781, https://doi.org/10.1007/s10915-017-0559-8, arXiv:1705.09715.
[41] P. Farkas, Mathematical Foundations for Fast Algorithms for the Biharmonic Equation, University of Chicago, Department of Mathematics, 1989.
[42] C.L. Epstein, L. Greengard, A. Klöckner, On the convergence of local expansions of layer potentials, SIAM J. Numer. Anal. 51 (5) (2013) 2660–2679, https://doi.org/10.1137/120902859.
[43] L.N. Trefethen, Approximation Theory and Approximation Practice, SIAM, 2013.