A Cartesian Grid-based Boundary Integral Method for an Elliptic Interface Problem on Closely Packed Cells

Wenjun Ying *

Department of Mathematics, MOE-LSC and Institute of Natural Sciences, Shanghai Jiao Tong University, Minhang, Shanghai 200240, P. R. China.

Abstract

In this work, we propose a second-order version and a fourth-order version of a Cartesian grid-based boundary integral method for an interface problem of the Laplace equation on closely packed cells. When the cells are closely packed, the boundary integrals involved in the boundary integral formulation for the interface problem become nearly singular. Direct evaluation of the boundary integrals has accuracy issues. The grid-based method evaluates a boundary integral by first solving an equivalent, simple interface problem on a Cartesian grid with a fast Fourier transform based Poisson solver, then interpolating the grid solution to get values of the boundary integral at discretization points of the interface. The grid-based method presents itself as an alternative but accurate numerical method for evaluating nearly singular, singular and hyper-singular boundary integrals. This work can be regarded as a further development of the kernel-free boundary integral method [W.-J. Ying and C. S. Henriquez, A kernel-free boundary integral method for elliptic boundary value problems, Journal of Computational Physics, Vol. 227 (2007), pp. 1046-1074] for problems in unbounded domains. Numerical examples with both second-order and fourth-order versions of the grid-based method are presented to demonstrate the accuracy of the method.

Key words: Laplace equation, cell suspension, inclusion of grains, boundary integral method, Cartesian grid method, finite difference method, fast Fourier transform, kernel-free

* Corresponding author.
Email address: wying@sjtu.edu.cn (Wenjun Ying).

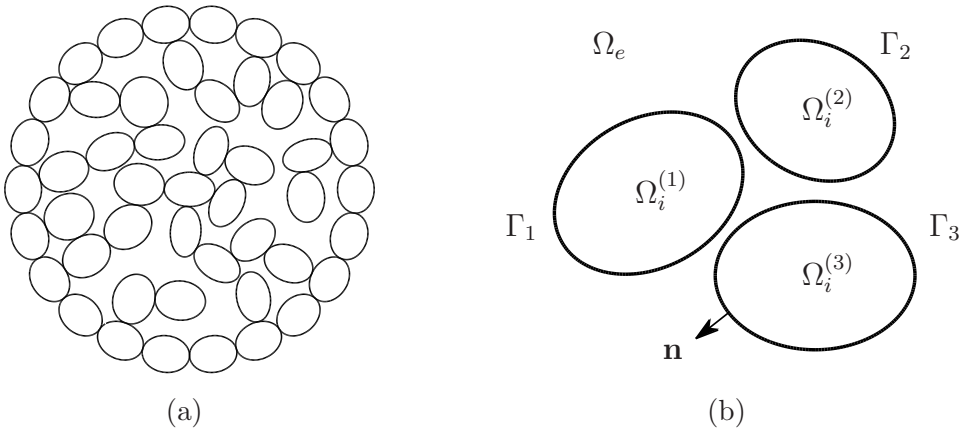


Fig. 1. Multi-component interfaces: (a) closely packed cells, (b) illustration of symbols for the interface and subdomains (cells)

1 Introduction

Let $\Omega_i \subset \mathbb{R}^2$ be a bounded open set with smooth boundary, which may have multiple disconnected components, $\Omega_e = \mathbb{R}^2 \setminus \bar{\Omega}_i$ be the unbounded, complementary domain and Γ be the interface, the common boundary of Ω_i and Ω_e. When the interface Γ has multiple components, we write $\Gamma = \bigcup_{k=1}^{K} \Gamma_k$, and assume each component Γ_k is a simple closed curve. We call the subdomain enclosed by each interface component Γ_k a cell, denoted by $\Omega_i^{(k)}$.

Let $\mathbf{p} = (x, y)^{\mathrm{T}} \in \mathbb{R}^2$ be a point in space. Suppose $\Phi_i(\mathbf{p})$ and $\Phi_e(\mathbf{p})$ are two unknown potential functions on Ω_i and Ω_e, respectively. They satisfy the Laplace equation

$$\triangle \Phi_i(\mathbf{p}) = 0 \qquad \text{in } \Omega_i \tag{1}$$

and

$$\triangle \Phi_e(\mathbf{p}) = 0 \qquad \text{in } \Omega_e. \tag{2}$$

Let

$$\Phi(\mathbf{p}) = \begin{cases} \Phi_i(\mathbf{p}) & \mathbf{p} \in \Omega_i \\ \Phi_e(\mathbf{p}) & \mathbf{p} \in \Omega_e \end{cases}.$$

In general, the function $\Phi(\mathbf{p})$ is discontinuous across the interface Γ. Let

$$\Phi_i(\mathbf{p}) - \Phi_e(\mathbf{p}) = V_m(\mathbf{p}) \qquad \text{on } \Gamma, \tag{3}$$

where $V_m(\mathbf{p})$ will be known. Assume the conductivities σ_i and σ_e on Ω_i and Ω_e are constant but distinct ($\sigma_i \neq \sigma_e$). Let

$$\sigma_i \frac{\partial \Phi_i(\mathbf{p})}{\partial \mathbf{n}_\mathbf{p}} - \sigma_e \frac{\partial \Phi_e(\mathbf{p})}{\partial \mathbf{n}_\mathbf{p}} = J_m(\mathbf{p}) \qquad \text{on } \Gamma. \tag{4}$$

Here, $\mathbf{n_p}$ is the unit normal vector pointing from the bounded domain Ω_i to the unbounded domain Ω_e at point $\mathbf{p} \in \Gamma$; $J_m(\mathbf{p})$ will be known, too. We assume the potential function $\Phi_e(\mathbf{p})$ satisfies the far field condition

$$\Phi_e(\mathbf{p}) \to 0 \qquad \text{as } |\mathbf{p}| = \sqrt{x^2 + y^2} \to \infty. \tag{5}$$

The interface problem (1)-(5) may describe the electric potential of biological cells or cardiac myocytes [24,49] in the applications of gene transfection [33,34], electrochemotherapy of tumors [39] and cardiac defibrillation [4], where the far-field condition (5) may need to be appropriately modified to model the stimulation of biological cells by an external electric field.

We focus here on the interface problem of the Laplace equation. But similar interface problems and computational issues occur in other applications or for other elliptic partial differential equations. For example, the motion of many drops of one viscous fluid in another, or the fluid motion of vesicles, such as blood cells, is often modeled by Stokes flow, leading to a similar interface problem with many components embedded in a surrounding medium [35, 45, 46, 53, 54].

For the interface problem on closely packed cells, when it is solved by a boundary integral method, evaluation of the involved boundary integrals by the standard method such as the composite trapezoidal rule has accuracy issues [5] since the boundary integrals become nearly singular. Special treatment is usually required for the nearly singular boundary integrals [6,8,9,15–17,23,47,48].

The method that we will present is different from Ying-Beale [48], where the nearly singularities of boundary integrals on closely packed cells are handled by kernel regularization and asymptotic analysis. In this work, we follow the kernel-free boundary integral method [50–52], which is originally motivated by A. Mayo's work [29–31]. To evaluate a boundary integral, first we solve an equivalent interface problem on a larger rectangle/box, which embeds the interface Γ, with the finite difference method on a Cartesian grid, which covers the rectangle/box. Then we interpolate the discrete solution on the Cartesian grid to get values of the boundary integral at discretization points of the interface. The grid-based method may lead to second-order or fourth-order accurate numerical solutions, depending on the finite difference scheme and the interpolation stencil used. The method provides an alternative approach to accurately evaluating a nearly singular, singular or hyper-singular boundary integral on closely packed cells.

The grid-based boundary integral method of this work can be regarded as a further development of the kernel-free boundary integral method. The previous work on the kernel-free boundary integral method are all for boundary value or interface problems in bounded domains while this one is proposed for the

interface problem in the free space, an unbounded domain.

The remainder of the paper is organized as follows. Section 2 describes the boundary integral equation for the interface problem introduced in Section 1. Section 3 presents details of different components of the grid-based method for evaluating boundary integrals. Section 4 summarizes the grid-based boundary integral method. Section 5 has numerical results by both second-order and fourth-order versions of the grid-based method. Finally, in Section 6, we discuss on the advantages of and possible further improvements for the grid-based boundary integral method.

2 Boundary Integral Equation

We will express the potential function $\Phi(\mathbf{p})$ in terms of single and double layer potentials of the form

$$v(\mathbf{p}) = -\int_\Gamma G(\mathbf{p}-\mathbf{q})\psi(\mathbf{q})\,ds_{\mathbf{q}}\,, \qquad w(\mathbf{p}) = \int_\Gamma \frac{\partial G(\mathbf{q}-\mathbf{p})}{\partial \mathbf{n}_{\mathbf{q}}}\varphi(\mathbf{q})\,ds_{\mathbf{q}} \tag{6}$$

with some density functions ψ and φ. Here, $G(\mathbf{p}) = (2\pi)^{-1}\log|\mathbf{p}|$ is the fundamental solution of the Laplace equation in $\mathbb{R}^2$ and $s_{\mathbf{q}}$ is the arc length parameter of the interface Γ. Let

$$v(\mathbf{p}) = \begin{cases} v_i(\mathbf{p}) & \mathbf{p}\in\Omega_i \\ v_e(\mathbf{p}) & \mathbf{p}\in\Omega_e \end{cases} \quad \text{and} \quad w(\mathbf{p}) = \begin{cases} w_i(\mathbf{p}) & \mathbf{p}\in\Omega_i \\ w_e(\mathbf{p}) & \mathbf{p}\in\Omega_e \end{cases}.$$

We recall that the single layer potential v is continuous at Γ but $\partial v/\partial\mathbf{n}$ has a jump discontinuity at Γ (e.g., refer to Hsiao-Wendland [21]),

$$\begin{cases} \dfrac{\partial v_i(\mathbf{p})}{\partial \mathbf{n}_{\mathbf{p}}} = \dfrac{1}{2}\psi(\mathbf{p}) - \displaystyle\int_\Gamma \frac{\partial G(\mathbf{p}-\mathbf{q})}{\partial \mathbf{n}_{\mathbf{p}}}\psi(\mathbf{q})\,ds_{\mathbf{q}} \\ \dfrac{\partial v_e(\mathbf{p})}{\partial \mathbf{n}_{\mathbf{p}}} = -\dfrac{1}{2}\psi(\mathbf{p}) - \displaystyle\int_\Gamma \frac{\partial G(\mathbf{p}-\mathbf{q})}{\partial \mathbf{n}_{\mathbf{p}}}\psi(\mathbf{q})\,ds_{\mathbf{q}} \end{cases}; \tag{7}$$

the double layer potential w has a jump discontinuity at Γ,

$$\begin{cases} w_i(\mathbf{p}) = \dfrac{1}{2}\varphi(\mathbf{p}) + \displaystyle\int_\Gamma \frac{\partial G(\mathbf{q}-\mathbf{p})}{\partial \mathbf{n}_{\mathbf{q}}}\varphi(\mathbf{q})\,ds_{\mathbf{q}} \\ w_e(\mathbf{p}) = -\dfrac{1}{2}\varphi(\mathbf{p}) + \displaystyle\int_\Gamma \frac{\partial G(\mathbf{q}-\mathbf{p})}{\partial \mathbf{n}_{\mathbf{q}}}\varphi(\mathbf{q})\,ds_{\mathbf{q}} \end{cases}, \tag{8}$$

while $\partial w/\partial\mathbf{n}$ is continuous across Γ (e.g., refer to Hsiao-Wendland [21]). In this work, we denote by $[v] = v_i - v_e$, $[w] = w_i - w_e$, $[\partial_{\mathbf{n}} v] = \partial_{\mathbf{n}} v_i - \partial_{\mathbf{n}} v_e$ and

$[\partial_{\mathbf{n}} w] = \partial_{\mathbf{n}} w_i - \partial_{\mathbf{n}} w_e$ the jumps of the one-side limits of the single, double layer potentials and their normal derivatives across the interface Γ. In general, by the square bracket of a piecewise smooth function, we mean its jump across the interface, the one-side limit of the function on Γ from the interior domain Ω_i subtracted by the one-side limit of the function on Γ from the exterior domain Ω_e.

Now, assuming the solution Φ to the interface problem (1)-(5) exists, let

$$\psi(\mathbf{p}) = \frac{\partial \Phi_i(\mathbf{p})}{\partial \mathbf{n}_\mathbf{p}} - \frac{\partial \Phi_e(\mathbf{p})}{\partial \mathbf{n}_\mathbf{p}} \qquad \text{on } \Gamma. \tag{9}$$

Then the potential function $\Phi(\mathbf{p})$ can be represented as

$$\Phi(\mathbf{p}) = \int_\Gamma \frac{\partial G(\mathbf{q}-\mathbf{p})}{\partial \mathbf{n}_\mathbf{q}} V_m(\mathbf{q})\, ds_\mathbf{q} - \int_\Gamma G(\mathbf{p}-\mathbf{q})\, \psi(\mathbf{q})\, ds_\mathbf{q}. \tag{10}$$

According to the properties above, this expression for Φ will have the jumps prescribed in (3) and (9). The unknown density $\psi(\mathbf{p})$ in (10) is determined by the interface condition (4).

Let $\mathbf{t}_\mathbf{p} = (x'(s), y'(s))^T$ be the unit tangent along the interface, so that the unit outward normal $\mathbf{n}_\mathbf{p} = (y'(s), -x'(s))^T$. From the continuity properties of the single and double layer potentials and the interface condition (4), we get the boundary integral equation (refer to Ying-Beale [48])

$$\frac{1}{2}\psi(\mathbf{p}) + \mu \int_\Gamma \frac{\partial G(\mathbf{p}-\mathbf{q})}{\partial \mathbf{n}_\mathbf{p}}\, \psi(\mathbf{q})\, ds_\mathbf{q} = \mu\, \frac{\partial}{\partial \mathbf{n}_\mathbf{p}} \int_\Gamma \frac{\partial G(\mathbf{q}-\mathbf{p})}{\partial \mathbf{n}_\mathbf{q}} V_m(\mathbf{q})\, ds_\mathbf{q} + \jmath_m(\mathbf{p})$$

with $\mu = (\sigma_e - \sigma_i)/(\sigma_e + \sigma_i) \in (-1, 1)$ and $\jmath_m(\mathbf{p}) = J_m(\mathbf{p})/(\sigma_i + \sigma_e)$. The integral equation above can be re-written concisely as

$$\frac{1}{2}\,\psi + \mu\, \mathcal{M}^* \psi = \mu\, \mathcal{N} V_m + \jmath_m \qquad \text{on } \Gamma, \tag{11}$$

where $\mathcal{M}^*$ and $\mathcal{N}$ are the integral operators given by

$$(\mathcal{M}^*\psi)(\mathbf{p}) = \int_\Gamma \frac{\partial G(\mathbf{p}-\mathbf{q})}{\partial \mathbf{n}_\mathbf{p}} \psi(\mathbf{q})\, ds_\mathbf{q} \qquad \text{on } \Gamma,$$

$$(\mathcal{N}\varphi)(\mathbf{p}) = \frac{\partial}{\partial \mathbf{n}_\mathbf{p}} \int_\Gamma \frac{\partial G(\mathbf{q}-\mathbf{p})}{\partial \mathbf{n}_\mathbf{q}} \varphi(\mathbf{q})\, ds_\mathbf{q} \qquad \text{on } \Gamma$$

for density functions ψ and φ defined on the interface.

Integral equations such as (11) are often solved by the biconjugate gradient method [44, 54] or the generalized minimal residual (GMRES) method [36, 38, 45]. In this work, we solve the integral equation (11) by the Richardson

iteration. The spectrum of the operator $\mathcal{M}^*$ is contained in the interval $-\frac{1}{2} < \lambda \leq \frac{1}{2}$ (e.g., refer to Kress [25]), and consequently the iteration

$$\psi_{\nu+1} = (1-\beta)\,\psi_\nu + 2\beta(\mu\,\mathcal{N}V_m + \jmath_m - \mu\,\mathcal{M}^*\psi_n) \qquad \text{for} \quad \nu = 0, 1, 2, \cdots \tag{12}$$

converges to the exact solution for $0 < \beta < 2/(1+\mu)$.

Let $\mathcal{L}$ and $\mathcal{M}$ be the single layer and double layer boundary integral operators given by

$$(\mathcal{L}\psi)(\mathbf{p}) = \int_\Gamma G(\mathbf{p}-\mathbf{q})\psi(\mathbf{q})\,ds_{\mathbf{q}},$$
$$(\mathcal{M}\varphi)(\mathbf{p}) = \int_\Gamma \frac{\partial G(\mathbf{q}-\mathbf{p})}{\partial \mathbf{n}_{\mathbf{q}}}\varphi(\mathbf{q})\,ds_{\mathbf{q}}$$

for density functions ψ and φ defined on the interface. The solution (10) to the interface problem can also be concisely re-written as

$$\Phi(\mathbf{p}) = \mathcal{M}V_m - \mathcal{L}\psi \qquad \text{for} \quad \mathbf{p} \in \mathbb{R}^2. \tag{13}$$

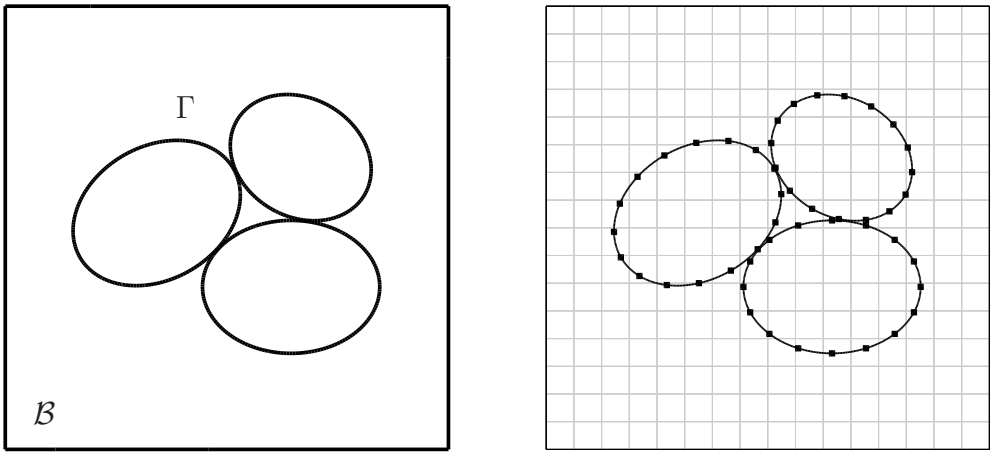
3 Evaluation of Boundary Integrals

As discussed in Section 1, for the interface problem (1)-(5) around multiple closely packed cells, direct evaluation of the boundary integrals $\mathcal{L}\psi$, $\mathcal{M}\varphi$, $\mathcal{M}^*\psi$ and $\mathcal{N}\varphi$ will be inaccurate due to the nearly singularity or hyper-singularity of the integrals. To accurately evaluate the boundary integrals while avoiding special treatment for the (nearly) singularity of integrals, we follow the kernel-free boundary integral method [50–52].

We choose a larger rectangle/box $\mathcal{B}$ to properly embed the cells and the interface Γ, assuming the boundary of the box $\mathcal{B}$ is sufficiently far away from the interface Γ, and cover the box $\mathcal{B}$ with a uniform Cartesian grid; refer to Fig. 2 for illustration.

To evaluate the single or double layer potential boundary integral, $\mathcal{L}\psi$ or $\mathcal{M}\varphi$, 1) first we solve an equivalent interface problem on the Cartesian grid with a finite difference method; 2) then we interpolate the solution on the Cartesian grid to get values of the boundary integral at discretization points of the interface (see the marked rectangles in Fig. 2 (b)).

To evaluate the adjoint double layer potential $\mathcal{M}^*\psi$ or the hyper-singular boundary integral $\mathcal{N}\varphi$, we do not interpolate values of the single layer potential or the double layer potential in step 2 above. Instead, we first compute the first-order partial derivatives of the interpolant polynomial with the grid solution



(a) (b)

Fig. 2. Three closely packed cells: a) embedded into a larger rectangle/box $\mathcal{B}$; b) on a Cartesian grid with marked discretization points of the interface Γ

and then get values of the adjoint double layer potential or the hyper-singular boundary integral by linear combinations of the first-order partial derivatives.

Due to the existence of the complex interface, the discrete interface equations on the Cartesian grid have to be modified. We correct each linear system by modifying its right hand side. The correction needs jumps of partial derivatives of the boundary integral. For the same reason, the polynomial interpolation encountered also needs jumps of the partial derivatives.

In this section, we will give details of the Cartesian grid-based evaluation method for boundary integrals on closely packed cells.

3.1 Equivalent interface problems

It is well-known that the single layer potential $v(\mathbf{p}) = -\mathcal{L}\psi(\mathbf{p})$ satisfies the interface problem (e.g., refer to Hsiao-Wendland [21])

$$\begin{cases} \triangle v = 0 & \text{in } \mathcal{B} \setminus \Gamma, \\ [v] = 0 & \text{on } \Gamma, \\ [\partial_{\mathbf{n}} v] = \psi & \text{on } \Gamma, \\ v = -\mathcal{L}\psi & \text{on } \partial\mathcal{B}. \end{cases}$$

The double layer potential $w(\mathbf{p}) = \mathcal{M}\varphi(\mathbf{p})$ satisfies the interface problem

$$\begin{cases} \triangle w = 0 & \text{in } \mathcal{B} \setminus \Gamma, \\ [w] = \varphi & \text{on } \Gamma, \\ [\partial_{\mathbf{n}} w] = 0 & \text{on } \Gamma, \\ w = \mathcal{M}\varphi & \text{on } \partial\mathcal{B}. \end{cases}$$

These two interface problems are much simpler to solve than the original interface problem (1)-(5).

Based on our assumptions on the interface Γ and the larger box $\mathcal{B}$, each interface problem above has a unique solution for sufficiently smooth density functions ψ and φ. Since the boundary of $\mathcal{B}$ is assumed to be sufficiently far away from Γ, the integrand functions of $v(\mathbf{p}) = -(\mathcal{L}\psi)(\mathbf{p})$ and $w(\mathbf{p}) = (\mathcal{M}\varphi)(\mathbf{p})$ for $\mathbf{p}$ on $\partial\mathcal{B}$ are regular, smooth functions. We may compute the Dirichlet boundary conditions for the interface problems by directly evaluating the boundary integrals. The computation will have no accuracy issues.

In the rest of this section, we consider the unified interface problem below

$$\begin{cases} \triangle u = 0 & \text{in } \mathcal{B} \setminus \Gamma, \\ [u] = \varphi & \text{on } \Gamma, \\ [\partial_{\mathbf{n}} u] = \psi & \text{on } \Gamma, \\ u = \mathcal{M}\varphi - \mathcal{L}\psi & \text{on } \partial\mathcal{B}. \end{cases} \tag{14}$$

Its solution is $u(\mathbf{p}) = (\mathcal{M}\varphi)(\mathbf{p}) - (\mathcal{L}\psi)(\mathbf{p})$.

3.2 Discretization of the PDE on a Cartesian grid

Assume the larger rectangle/box $\mathcal{B} = (a, b) \times (c, d)$ is chosen so that it can be partitioned into a uniform $I \times J$ Cartesian grid with $h = (b - a)/I = (d - c)/J > 0$. Here, I and J are two positive integers. For $i = 0, 1, \cdots, I$ and $j = 0, 1, \cdots, J$, let $x_i = a + ih$ and $y_j = c + jh$ be the coordinates of the vertical and horizontal grid lines, respectively. Denote by $\mathbf{p}_{i,j} = (x_i, y_j)^{\mathrm{T}}$ the $(i, j)^{th}$ node of the Cartesian grid.

We may discretize the Laplace equation of the interface problem (14) by either the standard five-point finite difference scheme or the standard nine-point compact finite difference scheme [27, 40].

Let $u_{i,j}$ be a finite difference approximation of $u(\mathbf{p}_{i,j})$ or $u(x_i, y_j)$ at the $(i, j)^{th}$

grid node $\mathbf{p}_{i,j}$. The five-point finite difference equation reads

$$\triangle_h^{(5)} u_{i,j} \equiv \frac{1}{h^2}\Big\{-4u_{i,\,j} + (u_{i+1,\,j} + u_{i,\,j+1} + u_{i-1,\,j} + u_{i,\,j-1})\Big\} = 0. \tag{15}$$

The nine-point finite difference equation reads

$$\begin{aligned}\triangle_h^{(9)} u_{i,j} \equiv \frac{1}{h^2}\Big\{&-\frac{10}{3}u_{i,j} + \frac{2}{3}\left(u_{i+1,\,j} + u_{i,\,j+1} + u_{i-1,\,j} + u_{i,\,j-1}\right) \\ &+ \frac{1}{6}\left(u_{i+1,\,j+1} + u_{i-1,\,j+1} + u_{i-1,\,j-1} + u_{i+1,\,j-1}\right)\Big\} = 0. \end{aligned}\tag{16}$$

Note that $u_{i,j}$ is known for $i = 0, I$ or $j = 0, J$ by the Dirichlet boundary condition $u = \mathcal{M}\varphi - \mathcal{L}\psi$ on $\partial\mathcal{B}$. Either finite difference discretization yields a linear system for $(I-1)\times(J-1)$ unknowns, whose coefficient matrix can be inverted by a fast Poisson solver [10, 11, 19, 20, 41].

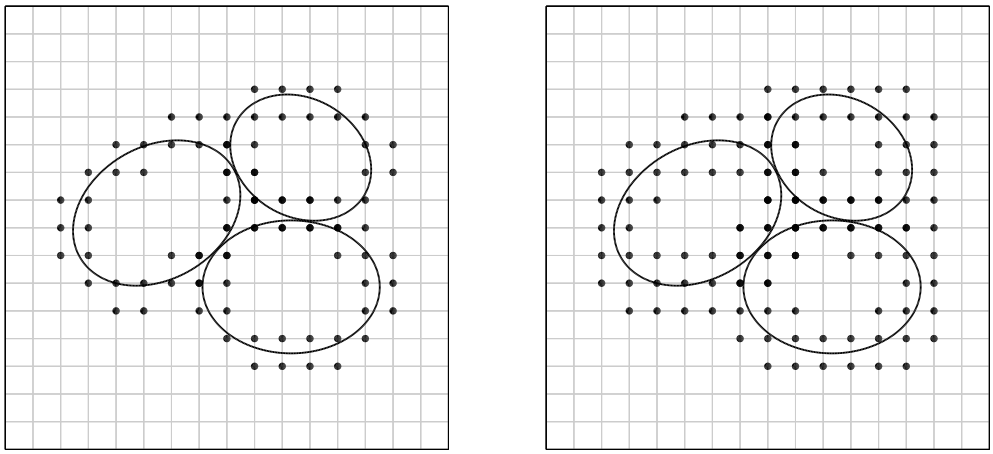
3.3 Correction of the discrete linear system

We know that, if there is no interface Γ for the Laplace equation, given sufficiently smooth Dirichlet boundary data, the five-point finite difference discretization (15) produces second-order accurate numerical solution and the nine-point finite difference discretization (16) yields fourth-order accurate numerical solution [27,40]. However, due to the existence of the complex interface Γ, straightforward finite difference discretization of the Laplace equation usually has very large local truncation errors at some nodes of the Cartesian grid and produces inaccurate numerical solution. To get a numerical solution with the formal order of accuracy, appropriate correction is needed at those grid nodes.

One point that we shall keep in mind during the correction is that, in order that we can still be able to invert the modified linear system with a fast Poisson solver, we should avoid modifying the coefficient matrix and instead only change the right hand side of the linear system.

We classify the grid nodes $\{\mathbf{p}_{i,j}\}$ as interior, exterior, regular and irregular nodes. We call a node $\mathbf{p}_{i,j}$ an interior grid node if it lies inside one of the cells $\{\Omega_i^{(k)}\}_{k=1}^K$; otherwise, we call a node $\mathbf{p}_{i,j}$ an exterior grid node.

We identify a grid node $\mathbf{p}_{i,j}$ as a regular grid node if the nodes involved with the finite difference stencil at $\mathbf{p}_{i,j}$ are all in the exterior domain Ω_e or all in the same cell $\Omega_i^{(k)}$ for some $k \in \{1, 2, \cdots, K\}$; otherwise, we call a node $\mathbf{p}_{i,j}$ an irregular grid node. It is obvious that the classification of regular and irregular grid nodes depends on the finite difference scheme; refer to Fig. 3 for three closely



(a) (b)

Fig. 3. Three closely packed cells on a Cartesian grid with irregular nodes marked: a) classification of irregular nodes by the five-point finite difference discretization; b) classification of irregular nodes by the nine-point compact finite difference discretization

packed cells on a 16×16 Cartesian grid with irregular nodes classified by the five-point and nine-point different finite difference discretizations. A regular grid node with the five-point finite difference discretization is not necessarily also a regular grid node with the nine-point compact finite difference method. An irregular grid node with the nine-point compact finite difference scheme is not necessarily also an irregular grid node with the five-point finite difference method.

In this work, we assume that the Cartesian grid is fine enough so that, for the finite difference stencil at an irregular grid node $\mathbf{p}_{i,j}$, any line segment that connects $\mathbf{p}_{i,j}$ with another point in the stencil intersects the interface at most twice.

At a regular grid node, the five-point finite difference discretization has a second-order local truncation error while the nine-point finite difference discretization has a fourth-order local truncation error. At an irregular grid node, both the five-point and the nine-point finite difference discretizations may introduce large local truncation errors on the order of $O(h^{-2})$. We will estimate the local truncation errors at irregular grid nodes.

For the solution $u(\mathbf{p}) = u(x, y)$ to the simple interface problem (14), denote by $u^{(k)}(x, y)$ its restriction on $\Omega_i^{(k)}$ for $k = 1, 2, \cdots, K$, and by $u^{(0)}(x, y)$ its restriction on the exterior domain Ω_e. That is, we have

$$u(x, y) = \begin{cases} u^{(k)}(x, y) & \text{if } (x, y)^{\mathrm{T}} \in \Omega_i^{(k)}, \\ u^{(0)}(x, y) & \text{if } (x, y)^{\mathrm{T}} \in \Omega_e. \end{cases}$$

In particular, in the finite difference equations (15) and (16), the discrete value $u_{i,j}$ is an approximation of $u^{(k)}(x_i, y_j)$ if $(x_i, y_j)^{\mathrm{T}} \in \Omega_i^{(k)}$ or an approximation of $u^{(0)}(x_i, y_j)$ if $(x_i, y_j)^{\mathrm{T}} \in \Omega_e$.

To help compute local truncation errors at irregular grid nodes, we assume the functions $u^{(k)}(x, y)$, for $k = 0, 1, \cdots, K$, have smooth extensions in the whole box $\mathcal{B}$. As a matter of fact, it suffices to assume that each function $u^{(k)}(x, y)$ has a smooth extension to the other side in a neighborhood of the interface Γ_k. Here, $\Gamma_0 \equiv \Gamma$. For simplicity, in the next, we do not use different symbols for a function $u^{(k)}(x, y)$ and its extension. In other words, we assume $u^{(k)}(x, y)$ is a smooth function defined on $\mathcal{B}$ and coincides with $u(x, y)$ in the k^{th} cell/ellipse $\Omega_i^{(k)}$ for $k > 0$ or the exterior domain Ω_e for $k = 0$.

We write the five-point finite difference equation (15) and the nine-point compact finite difference equation (16) in a unified formulation

$$\sum_{r,s} a^{(r,s)} u_{i+r,j+s} = 0. \tag{17}$$

Here, the subscripts $r, s = -1$, 0 or 1. Assume $a^{(0,0)} = -4/h^2$ for the five-point finite difference scheme and $a^{(0,0)} = -10/(3h^2)$ for the nine-point compact finite difference scheme. Note that the non-zero coefficients $\{a^{(r,s)}\}$ are all on the order of $O(h^{-2})$.

At a grid node $\mathbf{p}_{i,j} = (x_i, y_j)^{\mathrm{T}} \in \Omega_i^{(k)}$ for $k \in \{ 1, 2, \cdots, K \}$ or $\mathbf{p}_{i,j} = (x_i, y_j)^{\mathrm{T}} \in \Omega_e$ for $k = 0$, by our assumption on the function $u^{(k)}(x, y)$ and its extension, we have

$$\sum_{r,s} a^{(r,s)} u^{(k)}(x_{i+r}, y_{j+s}) = \begin{cases} O(h^2) & \text{for the five-point scheme,} \\ O(h^4) & \text{for the nine-point scheme.} \end{cases} \tag{18}$$

Now we replace $u_{i+r,j+s}$ with $u(x_{i+r}, y_{j+s})$ in the summation on the left hand side of (17). If $\mathbf{p}_{i,j}$ is a regular grid node, the new summation after the replacement is on the order of h^2 or h^4 as (18). If $\mathbf{p}_{i,j}$ is an irregular grid node, the new summation is $O(h^{-2})$ and not negligible. We denote the summation by

$$E_{i,j} = \sum_{r,s} a^{(r,s)} u(x_{i+r}, y_{j+s}). \tag{19}$$

Next we subtract the right hand side of (19) by the summation on the left hand side of (18). This yields

$$E_{i,j} = \sum_{|r|+|s|\neq 0} a^{(r,s)} \left\{ u(x_{i+r}, y_{j+s}) - u^{(k)}(x_{i+r}, y_{j+s}) \right\} + O(h^p), \tag{20}$$

where $p = 2$ for the five-point finite difference scheme and $p = 4$ for the nine-point finite difference scheme. In the summation above, we exclude the case $r = s = 0$ since $u(x_i, y_j) = u^{(k)}(x_i, y_j)$ then.

Let

$$D_{i,j}^{(r,s)} = u(x_{i+r}, y_{j+s}) - u^{(k)}(x_{i+r}, y_{j+s}) = u^{(\ell)}(x_{i+r}, y_{j+s}) - u^{(k)}(x_{i+r}, y_{j+s}) \quad (21)$$

with $1 \le \ell \le K$ if $(x_{i+r}, y_{j+s})^T \in \Omega_i^{(\ell)}$ and $\ell = 0$ if $(x_{i+r}, y_{j+s})^T \in \Omega_e$.

In the case that the line segment that connects the points $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i+r,j+s}$ does not intersect the interface Γ, we have $\ell = k$ and $D_{i,j}^{(r,s)} = 0$. In the case that the line segment that connects the points $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i+r,j+s}$ intersects the interface Γ at one and exactly one point, one of the points must be in the exterior domain Ω_e and we have $\ell = 0 \neq k$ or $k = 0 \neq \ell$. In the case that the line segment that connects the points $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i+r,j+s}$ intersects the interface at two points (see Fig. 4), which are on two different components Γ_k and Γ_ℓ, we have $\ell > 0$ and $k > 0$. For the third case, we may decompose the difference $D_{i,j}^{(r,s)}$ as a sum of two parts

$$\begin{aligned} u^{(\ell)}(x_{i+r}, y_{j+s}) - u^{(k)}(x_{i+r}, y_{j+s}) = &\left\{u^{(\ell)}(x_{i+r}, y_{j+s}) - u^{(0)}(x_{i+r}, y_{j+s})\right\} \\ &+ \left\{u^{(0)}(x_{i+r}, y_{j+s}) - u^{(k)}(x_{i+r}, y_{j+s})\right\}. \end{aligned}$$

Here, we remark that we do not consider the case that the line segment which connects the points $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i+r,j+s}$ intersects a single interface component Γ_k at two different points (For the interface problem (1)-(5) of our interest, we may assume any such two intersected points on Γ_k can be connected by a short path that completely lies in $\Omega_i^{(k)}$. This assumption implies the difference (21) in this case is negligible by the smoothness of $u^{(k)}(x, y)$). Now we show that $E_{i,j}$ can be approximated by a weighted sum of the differences of the form

$$\left\{u^{(k)}(x_{i+r}, y_{j+s}) - u^{(0)}(x_{i+r}, y_{j+s})\right\} \quad (22)$$

with $k > 0$. Assume the line segment that connects the points $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i+r,j+s}$ intersects the interface segment Γ_k at point $\mathbf{q}_k$. Let $(\xi, \eta)^{\mathrm{T}} \equiv \mathbf{p}_{i+r,j+s} - \mathbf{q}_k$.

We make local Taylor series expansions around the intersection point $\mathbf{q}_k$ for both functions $u^{(k)}(x, y)$ and $u^{(0)}(x, y)$, which gives us

$$\begin{aligned} u^{(k)}(x_{i+r}, y_{j+s}) = u^+ &+ \left\{u_x^+\xi + u_y^+\eta\right\} + \frac{1}{2}\left\{u_{xx}^+\xi^2 + 2u_{xy}^+\xi\eta + u_{yy}^+\eta^2\right\} \\ &+ \frac{1}{6}\left\{u_{xxx}^+\xi^3 + 3u_{xxy}^+\xi^2\eta + 3u_{xyy}^+\xi\eta^2 + \eta^3 u_{yyy}^+\right\} \\ &+ \frac{1}{24}\left\{u_{xxxx}^+\xi^4 + 4u_{xxxy}^+\xi^3\eta + 6u_{xxyy}^+\xi^2\eta^2\right. \\ &\left. + 4u_{xyyy}^+\xi\eta^3 + \eta^4 u_{yyyy}^+\right\} + O(h^5) \end{aligned} \quad (23)$$

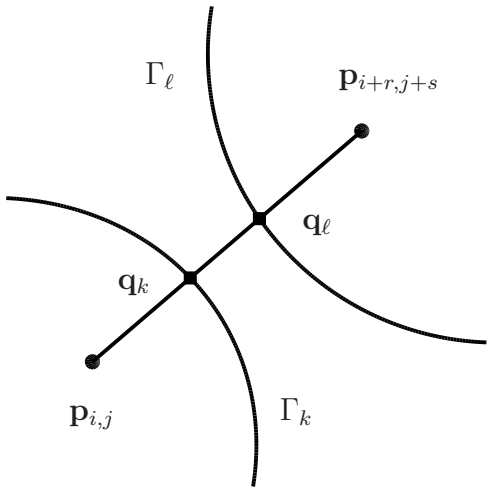


Fig. 4. A line segment that intersects two cells $\Omega_i^{(k)}$ and $\Omega_i^{(\ell)}$

and

$$\begin{aligned} u^{(0)}(x_{i+r}, y_{j+s}) = u^- &+ \left\{ u_x^- \xi + u_y^- \eta \right\} + \frac{1}{2}\left\{ u_{xx}^- \xi^2 + 2u_{xy}^- \xi\eta + u_{yy}^- \eta^2 \right\} \\ &+ \frac{1}{6}\left\{ u_{xxx}^- \xi^3 + 3u_{xxy}^- \xi^2\eta + 3u_{xyy}^- \xi\eta^2 + \eta^3 u_{yyy}^- \right\} \\ &+ \frac{1}{24}\left\{ u_{xxxx}^- \xi^4 + 4u_{xxxy}^- \xi^3\eta + 6u_{xxyy}^- \xi^2\eta^2 \right. \\ &\left. + 4u_{xyyy}^- \xi\eta^3 + \eta^4 u_{yyyy}^- \right\} + O(h^5). \end{aligned} \tag{24}$$

Here, the values of the functions and their partial derivatives on the right hand sides of the Taylor expansions above are all evaluated at the intersection point $\mathbf{q}_k$ and coincide with those of the solution function $u(x, y)$ and its corresponding partial derivatives; the superscripts, "+" and "-", mean the values are one-side limits of the functions, respectively, from the interior and exterior domains. Moreover, subtracting (23) by (24) yields

$$\begin{aligned} & u^{(k)}(x_{i+r}, y_{j+s}) - u^{(0)}(x_{i+r}, y_{j+s}) \\ = [u] &+ \left\{ [u_x]\xi + [u_y]\eta \right\} + \frac{1}{2}\left\{ [u_{xx}]\xi^2 + 2[u_{xy}]\xi\eta + [u_{yy}]\eta^2 \right\} \\ &+ \frac{1}{6}\left\{ [u_{xxx}]\xi^3 + 3[u_{xxy}]\xi^2\eta + 3[u_{xyy}]\xi\eta^2 + \eta^3[u_{yyy}] \right\} \\ &+ \frac{1}{24}\left\{ [u_{xxxx}]\xi^4 + 4[u_{xxxy}]\xi^3\eta + 6[u_{xxyy}]\xi^2\eta^2 \right. \\ &\left. + 4[u_{xyyy}]\xi\eta^3 + \eta^4[u_{yyyy}] \right\} + O(h^5). \end{aligned} \tag{25}$$

The quantities with the square brackets on the right hand side of the expan-

sion above are jumps of the piecewise smooth function $u(x, y)$ and its partial derivatives across the interface Γ at the point $\mathbf{q}_k$.

After replacing the difference (22) in $E_{i,j}$ by the fourth-order local Taylor series expansion (25) and truncating high order terms, we denote the resulting sum by $C_{i,j}^{(9)}$, which is a third-order $O(h^3)$ approximation to $E_{i,j}$, i.e.,

$$E_{i,j} = C_{i,j}^{(9)} + O(h^3).$$

Here, the fact that the coefficients $\{a^{(r,s)}\}$ in (19) are on the order of h^{-2} is used. The quantity $C_{i,j}^{(9)}$ is readily computable once the jumps of $u(x, y)$ and its partial derivatives (up to the fourth-order) are known. As a matter of fact, the jumps of the partial derivatives can be calculated from the interface conditions and the Laplace equation in (14). Details of the calculation are presented in Subsection 3.5.

We know that the computable quantity $C_{i,j}^{(9)}$ is for the fourth-order nine-point compact finite difference scheme. For the five-point finite difference scheme, it is enough to replace the difference (22) in $E_{i,j}$ by the second-order local Taylor series expansion

$$\begin{aligned} & u^{(k)}(x_{i+r}, y_{j+s}) - u^{(0)}(x_{i+r}, y_{j+s}) \\ = & [u] + ([u_x]\xi + [u_y]\eta) + \left\{ \frac{1}{2}[u_{xx}]\xi^2 + [u_{xy}]\xi\eta + \frac{1}{2}[u_{yy}]\eta^2 \right\} + O(h^3). \end{aligned} \quad (26)$$

We denote the resulting sum by $C_{i,j}^{(5)}$ after replacing the differences in the form of (22) in $E_{i,j}$ by the second-order local Taylor series expansion (26) and truncating high order terms. The computable quantity $C_{i,j}^{(5)}$ is a first-order $O(h)$ approximation to $E_{i,j}$, i.e.,

$$E_{i,j} = C_{i,j}^{(5)} + O(h).$$

With the linear combinations $C_{i,j}^{(5)}$ and $C_{i,j}^{(9)}$ of truncated Taylor expansions, we modify the finite difference equation by the five-point scheme at an irregular grid node to be

$$\sum_{r,s} a^{(r,s)} u_{i+r,j+s} = C_{i,j}^{(5)}; \quad (27)$$

and modify the finite difference equation by the nine-point compact scheme at an irregular grid node to be

$$\sum_{r,s} a^{(r,s)} u_{i+r,j+s} = C_{i,j}^{(9)}. \quad (28)$$

The modified five-point finite difference equation (27) at an irregular grid node has a local truncation error on the order of h while the modified nine-point

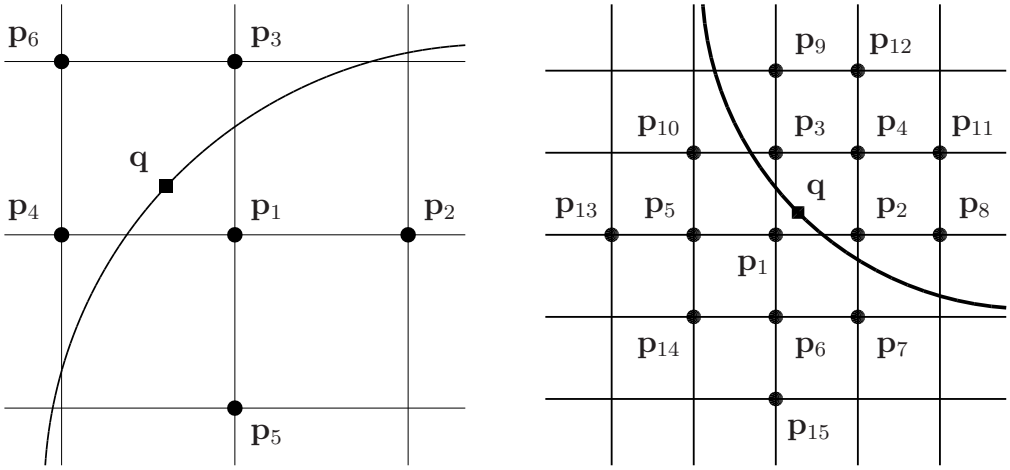


Fig. 5. Interpolation stencils: a) for the quadratic interpolation; b) for the quartic interpolation

finite difference equation (28) at an irregular grid node has a local truncation error on the order of h^3.

For both the five-point and the nine-point schemes, since the modification for the finite difference equations at irregular grid nodes does not change the coefficient matrix from the standard one, the corrected linear system can still be efficiently solved with a fast Poisson solver [20, 41].

3.4 Interpolation for integral values on the interface

By the kernel-free boundary integral method [50–52], after the grid solution $u_{i,j}$ is obtained by solving the modified finite difference system, we need to compute approximations of the corresponding boundary integral (for the single and double layer potentials) or its first-order partial derivatives (for the adjoint double layer potential and the hyper-singular boundary integrals) at discretization points of the interface by a two-variable Lagrange polynomial interpolation (refer to [12, 13] for a survey on and the history of multivariate polynomial interpolation).

With the grid solution $u_{i,j}$ computed by the five-point finite difference scheme, we make a quadratic interpolation, assuming the interpolant has the form

$$f_2(x, y) = c_1 + (c_2 x + c_3 y) + (c_4 x^2 + c_5 xy + c_6 y^2). \tag{29}$$

To determine the coefficients $\{c_\nu\}_{\nu=1}^6$, we need a six-point interpolation stencil as illustrated by Fig. 5 (a).

With the grid solution $u_{i,j}$ computed by the nine-point finite difference scheme,

we make a quartic interpolation, assuming the interpolant has the form

$$f_4(x,y) = c_1 + (c_2x + c_3y) + (c_4x^2 + c_5xy + c_6y^2) + (c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3) + (c_{11}x^4 + c_{12}x^3y + c_{13}x^2y^2 + c_{14}xy^3 + c_{15}y^4), \quad (30)$$

To determine the coefficients $\{c_\nu\}_{\nu=1}^{15}$, we need a fifteen-point interpolation stencil as illustrated by Fig. 5 (b).

Fig. 6 shows the first few Lagrange interpolation stencils (of polynomial degree up to six) for computing values in the shaded region, which is at the lower-left corner of a grid element/cell, with discrete data at the grid nodes.

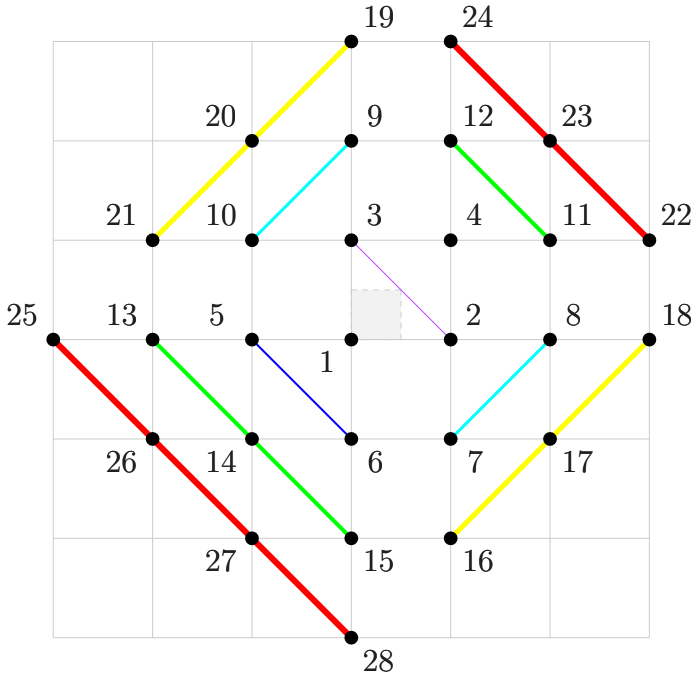


Fig. 6. Stencils for two-variable Lagrange interpolation in the shaded region, which is at the lower-left corner of a grid cell

In Fig. 6, the grid nodes in different interpolation stencils from low to high are labeled orderly by integer numbers. The stencil of linear interpolation consists of the first three points (labeled from 1 to 3) and has the "2-1" pattern. The stencil of quadratic interpolation consists of the first six points (labeled from 1 to 6) and has the "1-3-2" pattern. The stencil of cubic interpolation consists of the first ten points (labeled from 1 to 10) and has the "2-4-3-1" pattern. The stencil of quartic interpolation consists of the first fifteen points (labeled from 1 to 15) and has the "1-3-5-4-2" pattern. The stencil of quintic interpolation consists of the first twenty-one (labeled from 1 to 21) points and has the "2-4-6-5-3-1" pattern. Other interpolation stencil has either the "1-3-5-$\cdots$-6-4-2" pattern or the "2-4-6-$\cdots$-5-3-1" pattern. By the "1-3-5-$\cdots$-6-4-2" pattern, we mean the first column from left to right in the interpolation stencil has 1 grid

node, the second column has 3 grid nodes, the third column has 5 grid nodes and the last three columns have 6, 4, 2 grid nodes, respectively.

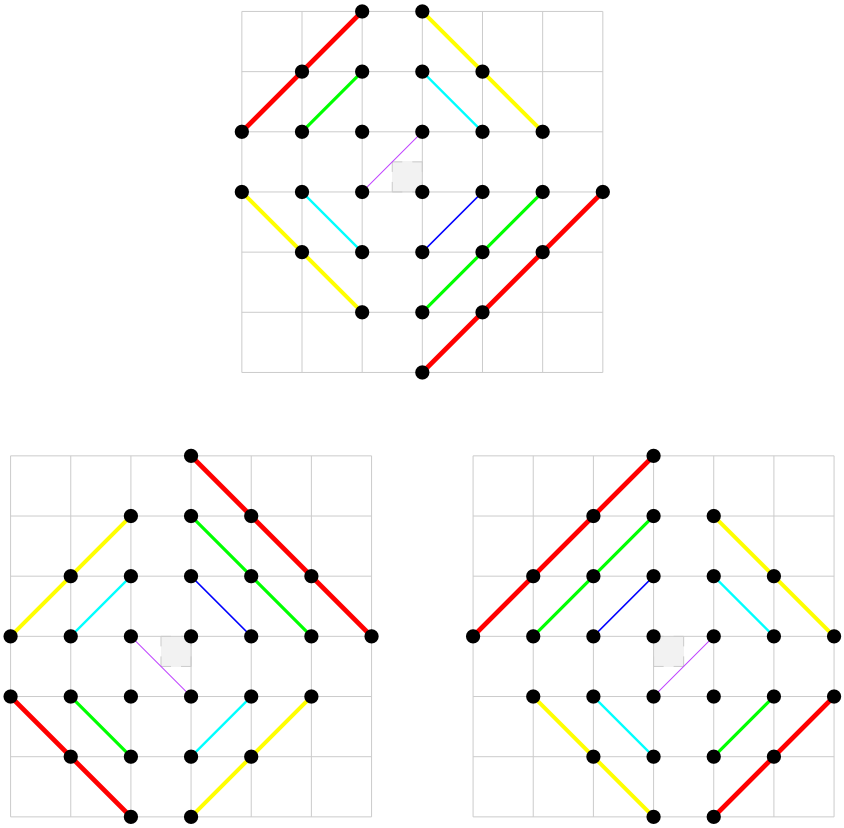


Fig. 7. Stencils for the two-variable Lagrange interpolation at points on the other three shaded regions in a grid cell

The interpolation stencils for points at the other three regions in a grid cell can be obtained by rotation or reflection from those shown in Fig. 6. They are illustrated in Fig. 7.

For the grid solution $u_{i,j}$ to the simple interface problem (14), due to the discontinuity of the solution or/and its partial derivatives across the interface Γ, the two-variable polynomial interpolation need correction/modification, too. Similarly, we also do not want to modify the interpolation stencil or change the coefficient matrix of the resulting linear system and instead only adjust the discrete data at the grid nodes.

In the next, we only discuss on the correction for the quartic interpolation with (30). The correction for the quadratic interpolation with (29) is completely analogous.

Let $\mathbf{q}_k$ be a point on the boundary Γ_k of the k^{th} cell. Suppose we want to approximate the limit value at $\mathbf{q}_k$ of the extended function $u^{(k)}(x, y)$, which

coincides with the solution $u(x,y)$ on $\Omega_i^{(k)}$. We expect the interpolation polynomial $f_4(x,y)$ satisfies

$$f_4(x_{i+r_\nu}, y_{j+s_\nu}) \approx u^{(k)}(x_{i+r_\nu}, y_{j+s_\nu}) \qquad \text{for} \quad \nu = 1,\, 2, \cdots, 15, \tag{31}$$

where $\{\,(x_{i+r_\nu}, y_{j+s_\nu})\,\}_{\nu=1}^{15}$ are the points in the interpolation stencil used.

At an interpolation point $(x_{i+r_\nu}, y_{j+s_\nu})$ that is also inside $\Omega_i^{(k)}$, we simply replace $u^{(k)}(x_{i+r_\nu}, y_{j+s_\nu})$ in (31) by $u_{i+r_\nu, j+s_\nu}$, which gives us

$$f_4(x_{i+r_\nu}, y_{j+s_\nu}) = u_{i+r_\nu, j+s_\nu}. \tag{32}$$

At an interpolation point $(x_{i+r_\nu}, y_{j+s_\nu})$ that is inside $\Omega_i^{(\ell)}$ with $0 < \ell \neq k$, we replace $u^{(k)}(x_{i+r_\nu}, y_{j+s_\nu})$ in (31) by $u_{i+r_\nu, j+s_\nu} + \Big\{u^{(k)}(x_{i+r_\nu}, y_{j+s_\nu}) - u^{(\ell)}(x_{i+r_\nu}, y_{j+s_\nu})\Big\}$, which gives us

$$f_4(x_{i+r_\nu}, y_{j+s_\nu}) = u_{i+r_\nu, j+s_\nu} + \Big\{u^{(k)}(x_{i+r_\nu}, y_{j+s_\nu}) - u^{(\ell)}(x_{i+r_\nu}, y_{j+s_\nu})\Big\}. \tag{33}$$

At an interpolation point $(x_{i+r_\nu}, y_{j+s_\nu})$ that is in the exterior domain Ω_e, we replace $u^{(k)}(x_{i+r_\nu}, y_{j+s_\nu})$ by $u_{i+r_\nu, j+s_\nu} + \Big\{u^{(k)}(x_{i+r_\nu}, y_{j+s_\nu}) - u^{(0)}(x_{i+r_\nu}, y_{j+s_\nu})\Big\}$, which gives us

$$f_4(x_{i+r_\nu}, y_{j+s_\nu}) = u_{i+r_\nu, j+s_\nu} + \Big\{u^{(k)}(x_{i+r_\nu}, y_{j+s_\nu}) - u^{(0)}(x_{i+r_\nu}, y_{j+s_\nu})\Big\}. \tag{34}$$

As before, we replace the difference on the right hand side of (33) by

$$\begin{aligned}\Big\{u^{(k)}(x_{i+r_\nu}, y_{j+s_\nu}) - u^{(\ell)}(x_{i+r_\nu}, y_{j+s_\nu})\Big\} = &\Big\{u^{(k)}(x_{i+r_\nu}, y_{j+s_\nu}) - u^{(0)}(x_{i+r_\nu}, y_{j+s_\nu})\Big\} \\ &+ \Big\{u^{(0)}(x_{i+r_\nu}, y_{j+s_\nu}) - u^{(\ell)}(x_{i+r_\nu}, y_{j+s_\nu})\Big\}.\end{aligned}$$

Now we also see that the right hand sides of (33)-(34) can be computed in terms of the differences of the form (22).

Moreover, we may further replace the differences by the corresponding fifth-order accurate local Taylor series expansions, and get the coefficients $\{c_\nu\}_{\nu=1}^{15}$ of the interpolation polynomial $f_4(x,y)$ by solving the resulting linear system.

Finally, we remark that the quartic interpolation and the fifth-order local Taylor series expansions are for the nine-point finite difference scheme. As we did before in [50, 52], for the five-point finite difference scheme, it suffices that we work with the quadratic polynomial interpolation and the third-order local Taylor series expansions for computing the corresponding differences of the form (22).

In the computation for the local Taylor series expansions, we need jumps of the solution and its partial derivatives, too. Details of the calculation for the jumps are presented in the next subsection.

3.5 Calculation for jumps of partial derivatives

In this section, we will derive equations for calculating jumps of the solution $u(x, y)$ to the simple interface problem (14) and its partial derivatives up to the fourth-order.

Let s be an arc length parameter of the interface Γ. Denote a point on the interface by $\mathbf{p} = (x(s), y(s))^T$. Differentiating the first interface condition in (14) with respect to the arc length parameter s, together with the second interface condition in (14), we get

$$\begin{cases} x'(s)[u_x] + y'(s)[u_y] = \varphi_s, \\ y'(s)[u_x] - x'(s)[u_y] = \psi. \end{cases} \tag{35}$$

Solving this two by two linear system gives us jumps of the first-order partial derivatives of u on Γ.

Differentiating the identities in (35) with respect to the arc length parameter s, respectively, together with the Laplace equation that the solution $u(x, y)$ satisfies, we get

$$\begin{cases} (x')^2[u_{xx}] + 2x'y'[u_{xy}] + (y')^2[u_{yy}] = \varphi_{ss} - \Big\{x''(s)[u_x] + y''(s)[u_y]\Big\}, \\ x'y'[u_{xx}] + \Big\{(y')^2 - (x')^2\Big\}[u_{xy}] - x'y'[u_{yy}] = \psi_s - y''[u_x] + x''[u_y], \\ [u_{xx}] + [u_{yy}] = 0. \end{cases} \tag{36}$$

Solving this three by three linear system gives us jumps of the second-order partial derivatives of u on Γ.

Differentiating the first equation in (36) with respect to the arc length parameter s gives us

$$(x')^3[u_{xxx}] + 3(x')^2y'[u_{xxy}] + 3x'(y')^2[u_{xyy}] + (y')^3[u_{yyy}] = r_{3,1} \tag{37}$$

with

$$r_{3,1} = \varphi_{sss} - (x'''[u_x] + y'''[u_y]) - 3\Big\{x''x'[u_{xx}] + (x''y' + x'y'')[u_{xy}] + y''y'[u_{yy}]\Big\}.$$

Differentiating the second equation in (36) with respect to the arc length parameter s gives us

$$(x')^2y'[u_{xxx}] + \Big\{2x'(y')^2 - (x')^3\Big\}[u_{xxy}] + \Big\{(y')^3 - 2(x')^2y'\Big\}[u_{xyy}] - x'(y')^2[u_{yyy}] = r_{3,2} \tag{38}$$

with

$$r_{3,2} = \psi_{ss} - y'''[u_x] + x'''[u_y] - (x''y' + 2x'y'')[u_{xx}] + 3(x'x'' - y'y'')[u_{xy}] + (2x''y' + x'y'')[u_{yy}].$$

We may get two more equations for jumps of the third-order partial derivatives by first differentiating the Laplace equation and then taking jumps of the resulting equations across Γ. They read

$$[u_{xxx}] + [u_{xyy}] = 0, \tag{39}$$

$$[u_{xxy}] + [u_{yyy}] = 0. \tag{40}$$

We may get jumps of the third-order partial derivatives, $[u_{xxx}]$, $[u_{xxy}]$, $[u_{xyy}]$ and $[u_{yyy}]$, by solving the system consisting of Eqns. (37)-(40).

Differentiating (37) with respect to the arc length parameter s yields

$$(x')^4[u_{xxxx}] + 4(x')^3y'[u_{xxxy}] + 6(x')^2(y')^2[u_{xxyy}] + 4x'(y')^3[u_{xyyy} + (y')^4[u_{yyyy}] = r_{4,1} \tag{41}$$

with

$$\begin{aligned} r_{4,1} = {} & \varphi_{ssss} - (x''''[u_x] + y''''[u_y]) \\ & - \Big\{(4x'''x' + 3x''x'')[u_{xx}] + (4x'''y' + 6x''y'' + 4x'y''')[u_{xy}] + (4y'''y' + 3(y'')^2)[u_{yy}]\Big\} \\ & - 6\Big\{x''(x')^2[u_{xxx}] + (2x'x''y' + (x')^2y'')[u_{xxy}] + (x''(y')^2 + 2x'y'y'')[u_{xyy}] + y''(y')^2[u_{yyy}]\Big\}. \end{aligned}$$

Differentiating (38) with respect to the arc length parameter s leads to

$$\begin{aligned} (x')^3y'[u_{xxxx}] + (x')^2\Big\{3(y')^2 - (x')^2\Big\}[u_{xxxy}] + 3x'y'\Big\{(y')^2 - (x')^2\Big\}[u_{xxyy}] \\ + (y')^2\Big\{(y')^2 - 3(x')^2\Big\}[u_{xyyy}] - x'(y')^3[u_{yyyy}] = r_{4,2} \end{aligned} \tag{42}$$

with

$$\begin{aligned} r_{4,2} = {} & \psi_{sss} - y''''[u_x] + x''''[u_y] - (x'''y' + 3x''y'' + 3x'y''')[u_{xx}] \\ & + (3x''x'' + 4x'x''' - 3y''y'' - 4y'y''')[u_{xy}] + (3x'''y' + 3x''y'' + x'y''')[u_{yy}] \\ & -3\Big\{x'x''y' + (x')^2y''\Big\}[u_{xxx}] + 3\Big\{2(x')^2x'' - x''(y')^2 - 3x'y'y''\Big\}[u_{xxy}] \\ & -3\Big\{2(y')^2y'' - 3x'x''y' - (x')^2y''\Big\}[u_{xyy}] + 3\Big\{x''(y')^2 + x'y'y''\Big\}[u_{yyy}]. \end{aligned}$$

Other three equations for jumps of the fourth-order partial derivatives read

$$[u_{xxxx}] + [u_{xxyy}] = 0, \tag{43}$$

$$[u_{xxxy}] + [u_{xyyy}] = 0, \tag{44}$$

$$[u_{xxyy}] + [u_{yyyy}] = 0, \tag{45}$$

which are similarly obtained by first differentiating the Laplace equation and then taking jumps of the resulting equations across Γ.

We may get jumps of the fourth-order partial derivatives, $[u_{xxxx}]$, $[u_{xxxy}]$, $[u_{xxyy}]$, $[u_{xyyy}]$ and $[u_{yyyy}]$, by solving the system consisting of Eqns. (41)-(45).

4 Algorithm Summary

Setup: 1) discretize the interface Γ by a set of quasi-uniformly spaced points; 2) partition the rectangle/box $\mathcal{B}$ into a uniform Cartesian grid and identity interior and irregular grid nodes; 3) for each five-point or nine-point finite difference stencil at the irregular grid nodes, find all intersection points of the interface with the line segments that connect the corresponding irregular grid node with other points in the stencil.

Iteration of the boundary integral equation: 1) choose an initial guess for the discrete unknown density of the boundary integral equation; 2) evaluate the boundary integral by the Cartesian grid-based method; 3) update the discrete unknown density by the Richardson iteration, which may be replaced with a Krylov subspace method [37]; 4) repeat the previous steps until the discrete residual of the boundary integral equation is sufficiently small in some norm.

Evaluation of a boundary integral: 1) first we correct the right hand side of the discrete interface equations at irregular grid nodes; 2) compute the Dirichlet data on the boundary of the bounding box by directly evaluating the boundary integrals on the interface with the composite trapezoidal rule at discretization points of the interface; 3) next we solve the modified linear system by a fast Fourier transform (precisely fast sine transform) based solver; 4) then we get approximate values of the boundary integral at discretization points of the interface by the two-variable Lagrange interpolation with the data on the Cartesian grid.

Steps 1) and 4) in the evaluation of boundary integrals above both involve calculation for jumps of partial derivatives. In the calculation for jumps of partial derivatives, the tangential differentiation of the densities of the single layer and double layer boundary integrals is done numerically by the Lagrange interpolation with the discrete data at discretization points of the interface.

Classification of irregular grid nodes and each step in the grid-based evaluation for boundary integrals depend on the finite difference scheme used.

5 Numerical Results

In this section, we present numerical examples for the interface problem (1)-(5) with the boundary integral method proposed in this work.

In all examples, the interface Γ consists of multiple ellipses, some of which are closely packed. For $k = 1, 2, \cdots, K$, we represent the k^{th} ellipse by

$$\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} = \begin{pmatrix} c_{k,x} \\ c_{k,y} \end{pmatrix} + \begin{pmatrix} \cos\alpha_k & -\sin\alpha_k \\ \sin\alpha_k & \cos\alpha_k \end{pmatrix} \begin{pmatrix} a_k \cos\theta \\ b_k \sin\theta \end{pmatrix} \qquad \text{for} \quad \theta \in [0,\, 2\pi),$$

where $(c_{k,x}, c_{k,y})^T$ is the center of the ellipse, a_k and b_k are the major and minor axis radii, α_k is the rotation angle of the ellipse from the standard one whose axes are aligned with the coordinate lines. We choose conductivity parameters $\sigma_i = 1$ and $\sigma_e = 3$ for the interface problem. The interface data $V_m(\mathbf{p})$ and $J_m(\mathbf{p})$ are selected so that the exact solution to the interface problem (1)-(5) reads

$$\Phi_i(\mathbf{p}) = \Phi_i(x, y) = x \qquad \text{in} \quad \Omega_i$$

$$\Phi_e(\mathbf{p}) = \Phi_e(x, y) = \sum_{k=1}^{K} \frac{c_{k,x} - x}{(c_{k,x} - x)^2 + (c_{k,y} - y)^2} \qquad \text{in} \quad \Omega_e.$$

The bounding box $\mathcal{B}$ of the simple interface problem (14), which is formulated for evaluating boundary integrals, is set to be $\mathcal{B} = [-1.5, 1.5] \times [-1.5, 1.5]$. In addition to partitioning the bounding box $\mathcal{B}$ into a uniform $I \times J$ Cartesian grid, we also divide each interface component Γ_k into M pieces with the discretization points $\{(x_m^{(k)}, y_m^{(k)})\}_{m=1}^M$ given by

$$\begin{pmatrix} x_m^{(k)} \\ y_m^{(k)} \end{pmatrix} = \begin{pmatrix} c_{k,x} \\ c_{k,y} \end{pmatrix} + \begin{pmatrix} \cos\alpha_k & -\sin\alpha_k \\ \sin\alpha_k & \cos\alpha_k \end{pmatrix} \begin{pmatrix} a_k \cos\theta_m \\ b_k \sin\theta_m \end{pmatrix}$$

for $\theta_m = 2\pi m/M$, $m = 1, 2, \cdots, M$. The Dirichlet boundary condition $u(\mathbf{p}) = (\mathcal{M}\varphi)(\mathbf{p}) - (\mathcal{L}\psi)(\mathbf{p})$ on $\partial\mathcal{B}$ are computed by directly evaluating the boundary integrals with the composite trapezoidal rule on the discretization points of the interface, $\{(x_m^{(k)}, y_m^{(k)})\}_{m=1}^M$ for $k = 1, 2, \cdots, K$. The finite difference equations on the Cartesian grid for the simple interface problem (14), whose right hand sides are modified/corrected at irregular grid nodes as described in Subsection 3.3, are all inverted with fast Fourier transform based Poisson solvers [20].

In the calculation for jumps of partial derivatives, the numerical differentiation of densities φ and ψ, whose values are known only at the discretization points $\{(x_m^{(k)}, y_m^{(k)})\}_{m=1}^M$ for $k = 1, 2, \cdots, K$, is done by a local cubic or quartic Lagrange interpolation when the simple interface problem (14) is respectively discretized with the five-point or nine-point finite difference scheme. With the

five-point finite difference scheme, the local cubic Lagrange interpolation for the numerical differentiation can be replaced with a quadratic Lagrange interpolation, which will not degrade the global second-order accuracy of the numerical solution to the interface problem (1)-(5).

The parameter for the Richardson iteration is fixed to be $\beta = 1.4/(1+\mu)$. In each run, the unknown density for the Richardson iteration is initialized with zero and the iteration is terminated when the discrete maximum norm of the residual is less than the tolerance $\epsilon = 10^{-10}$.

The Cartesian grid-based boundary integral method proposed in this work was implemented in custom codes written in the C/C++ computer language. The numerical experiments were performed in double precision on a MacBook Pro laptop computer with a 2.5 GHz Intel Core i7 CPU for Examples 1-3 and on a desktop computer with Intel(R) Xeon(R) 2.8 GHz CPU for Examples 4-6.

Finally, we remark that, in our implementation, the correction for the right hand side of the discrete interface equations, the interpolation for integral values on the interface with the grid data, and the calculation for the jumps of partial derivatives are slightly different from what is described in this paper. For example, the local Taylor series expansions (25) and (26) for differences of the form (21) are not necessarily made around the intersection points (sometimes it is more convenient to make the local Taylor series expansions around the interpolation points or other points nearby the intersection points); the interpolant polynomials for integral values on the interface are in practice functions of a pair of two scaled and translated variables rather than the original independent variables x and y. But these deviations are not essential.

In each table of numerical results in this section, the first row shows the sizes of the uniform Cartesian grids for the bounding box $\mathcal{B}$, the second row shows the numbers of discretization points on each interface component (ellipse), the third row has the numbers of Richardson iterations and the fourth and fifth rows, respectively, show the discrete maximum interior and exterior errors of the numerical solution at the grid nodes. The last row shows the CPU times (in seconds) used by the computer.

Example 1. The interface in this example consists of three closely packed ellipses, $K = 3$. The centers, radii and rotation angles (in degrees) of the ellipses are listed in Table 1. The distances between the ellipses are 4.79×10^{-4}, 4.79×10^{-4} and 3.22×10^{-4}, respectively. Fig. 8 shows isolines of the numerical potential $\Phi(\mathbf{p})$ on a uniform 512×512 Cartesian grid. Numerical results are summarized in Tables 2-3.

Example 2. The interface in this example is the union of a circle centered at the origin and a circular layer of five ellipses, which are identical up to rotation;

Table 1
Three ellipses of Example 1: center $(c_{k,x}, c_{k,y})$, major axis a_k, minor axis b_k and rotation angle α_k (in degrees)

k	$c_{k,x}$	$c_{k,y}$	a_k	b_k	α_k
1	0	0.35	0.65	0.38	90°
2	−0.6	−0.4	0.653	0.4	−30°
3	0.6	−0.4	0.653	0.4	30°

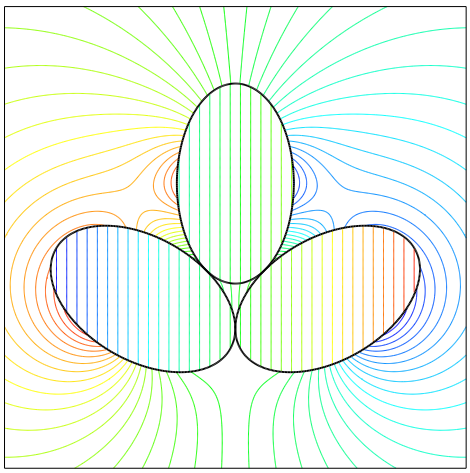


Fig. 8. Isolines plot of the potential (Example 1)

Table 2
Numerical results of Example 1 by the second-order method

grid size	64 × 64	128 × 128	256 × 256	512 × 512	1024 × 1024
M	32	64	128	256	512
#Richardson	30	30	32	32	32
$\|\mathbf{e}_h^{int}\|_\infty$	2.02E-2	3.54E-3	2.47E-4	6.46E-5	6.65E-6
$\|\mathbf{e}_h^{ext}\|_\infty$	1.34E-2	3.00E-3	2.65E-4	7.80E-5	1.09E-5
CPU (secs)	5.47E-2	1.76E-1	7.05E-1	2.74E+0	1.14E+1

refer to Fig. 9. The radii of the cells are chosen such that any two neighboring ellipses are almost in touch with a distance less than 10^{-4}. We generate the cells in the following way. First we put five identical ellipses (up to rotation) around the circle centered at the origin with radius equal to 0.4. We assume the centers of the five surrounding cells are evenly spaced and at a distance of 0.7 away from the origin. We choose the radii of the ellipses so that all of the cells are initially in touch with each other. To get the final configuration for the computation, we further multiply the radii of the cells by a number very close to one, which is $\rho = 0.9999$. Fig. 9 shows isolines of the numerical

Table 3
Numerical results of Example 1 by the fourth-order method

grid size	64×64	128×128	256×256	512×512	1024×1024
M	32	64	128	256	512
#Richardson	31	31	32	32	32
$\|\mathbf{e}_h^{int}\|_\infty$	1.92E-3	4.31E-4	9.71E-6	2.58E-7	1.27E-8
$\|\mathbf{e}_h^{ext}\|_\infty$	2.79E-3	2.13E-4	8.05E-6	2.33E-7	1.34E-8
CPU (secs)	1.32E-1	3.64E-1	1.21E+0	4.37E+0	1.73E+1

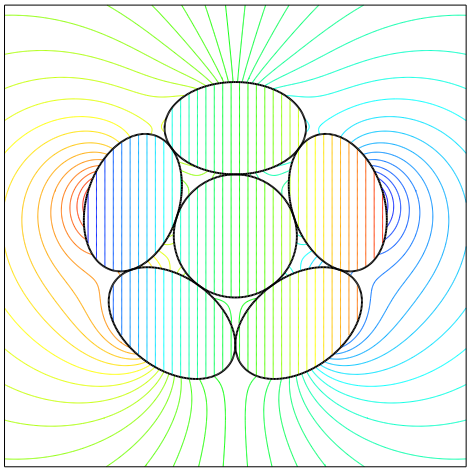


Fig. 9. Isolines plot of the potential (Example 2)

potential $\Phi(\mathbf{p})$ on a uniform 512×512 Cartesian grid. Numerical results are summarized in Tables 4-5. Table 4 has the results by the second-order version of the grid-based boundary integral method. Table 5 has the results by the fourth-order version of the grid-based boundary integral method.

Table 4
Numerical results of Example 2 by the second-order method

grid size	64×64	128×128	256×256	512×512	1024×1024
M	32	64	128	256	512
#Richardson	30	30	32	32	33
$\|\mathbf{e}_h^{int}\|_\infty$	3.56E-2	8.13E-3	1.04E-3	1.61E-4	2.37E-5
$\|\mathbf{e}_h^{ext}\|_\infty$	4.83E-2	8.28E-3	1.05E-3	1.62E-4	2.38E-5
CPU (secs)	9.26E-2	2.82E-1	1.05E+0	3.96E+0	1.65E+1

Example 3. The interface in this example is the union of a circle centered at the origin and a circular layer of nine ellipses, which are identical up to

Table 5
Numerical results of Example 2 by the fourth-order method

grid size	64×64	128×128	256×256	512×512	1024×1024
M	32	64	128	256	512
#Richardson	36	33	31	33	33
$\|\mathbf{e}_h^{int}\|_\infty$	7.91E-3	9.50E-4	4.36E-5	1.63E-6	3.24E-8
$\|\mathbf{e}_h^{ext}\|_\infty$	8.41E-3	3.75E-4	3.77E-5	1.41E-6	2.30E-8
CPU (secs)	2.73E-1	6.16E-1	1.71E+0	6.17E+0	2.34E+1

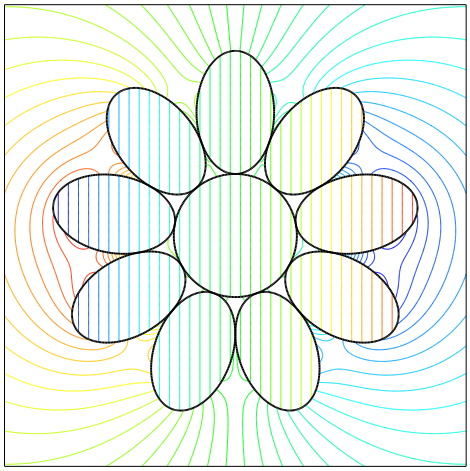


Fig. 10. Isolines plot of the potential (Example 3)

rotation; refer to Fig. 10. The cells are generated in the same way as the previous example. The circle in the middle is also centered at the origin with radius equal to 0.4. The centers of the surrounding ellipses are also evenly spaced and at a distance of 0.8 away from the origin. The distance of two neighboring cells is also less than 10^{-4}. Fig. 10 shows isolines of the numerical potential $\Phi(\mathbf{p})$ on a uniform 512×512 Cartesian grid. Numerical results are summarized in Tables 6-7. Table 6 has the results by the second-order version of the grid-based boundary integral method. Table 7 has the results by the fourth-order version of the grid-based boundary integral method.

Example 4. The interface in this example is the union of two circular layers of cells (see Fig. 11), each of which is a circle and has the same radius. The radii of the cells are chosen such that all cells are located inside the unit circle centered at the origin. The cells in the outer layer are almost in touch with the unit circle. Two adjacent cells in the same circular layer have a distance less than 10^{-4}. Fig. 11 shows isolines of a numerical potential $\Phi(\mathbf{p})$ for the interface problem. Numerical results are summarized in Tables 8-9. Table 8 has the results by the second-order version of the grid-based boundary integral

Table 6
Numerical results of Example 3 by the second-order method

grid size	64×64	128×128	256×256	512×512	1024×1024
M	32	64	128	256	512
#Richardson	31	32	32	33	34
$\|\mathbf{e}_h^{int}\|_\infty$	8.25E-2	1.14E-2	1.61E-3	1.60E-4	6.16E-5
$\|\mathbf{e}_h^{ext}\|_\infty$	7.60E-2	1.20E-2	1.80E-3	1.88E-4	6.31E-5
CPU (secs)	1.46E-1	4.44E-1	1.53E+0	5.88E+0	2.40E+1

Table 7
Numerical results of Example 3 by the fourth-order method

grid size	64×64	128×128	256×256	512×512	1024×1024
M	32	64	128	256	512
#Richardson	34	34	34	34	34
$\|\mathbf{e}_h^{int}\|_\infty$	8.74E-3	5.26E-4	4.25E-5	1.11E-6	6.26E-8
$\|\mathbf{e}_h^{ext}\|_\infty$	8.78E-3	4.91E-4	4.64E-5	1.24E-6	6.80E-8
CPU (secs)	4.04E-1	9.46E-1	2.62E+0	8.62E+0	3.17E+1

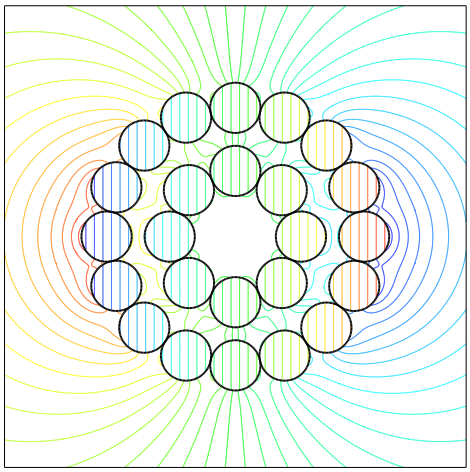


Fig. 11. Isolines plot of the potential (Example 4)

method. Table 9 has the results by the fourth-order version of the grid-based boundary integral method.

Example 5. The interface in this example is the union of a few circular layers of cells (see Fig. 12), each of which is an ellipse. The cells in each layer are identical up to rotation. The radii of ellipses in the direction pointing

Table 8
Numerical results of Example 4 by the second-order method

grid size	128×128	256×256	512×512	1024×1024	2048×2048
M	32	64	128	256	512
#Richardson	32	33	33	34	35
$\|\mathbf{e}_h^{int}\|_\infty$	5.81E-2	1.03E-2	3.31E-3	2.95E-4	7.28E-5
$\|\mathbf{e}_h^{ext}\|_\infty$	6.09E-2	1.07E-2	3.41E-3	3.65E-4	9.25E-5
CPU (secs)	1.02E+0	3.90E+0	1.52E+1	6.22E+1	2.59E+2

Table 9
Numerical results of Example 4 by the fourth-order method

grid size	128×128	256×256	512×512	1024×1024	2048×2048
M	32	64	128	256	512
#Richardson	32	32	34	34	35
$\|\mathbf{e}_h^{int}\|_\infty$	1.16E-2	7.79E-4	3.05E-5	1.50E-6	7.89E-8
$\|\mathbf{e}_h^{ext}\|_\infty$	9.91E-3	7.51E-4	2.85E-5	1.51E-6	8.22E-8
CPU (secs)	1.55E+0	4.96E+0	1.86E+1	7.06E+1	2.87E+2

to the center are all the same as the radius of the circle in the middle of the configuration. The radii of the cells are chosen such that all cells are located inside the unit circle centered at the origin and any two adjacent cells in a circular layer are almost in touch with a distance less than 10^{-4}. Fig. 12 shows isolines of a numerical potential $\Phi(\mathbf{p})$ for the interface problem. Numerical results are summarized in Tables 10-11. Table 10 has the results by the second-order version of the grid-based boundary integral method. Table 11 has the results by the fourth-order version of the grid-based boundary integral method.

Table 10
Numerical results of Example 5 (37 cells) by the second-order method

grid size	512×512	1024×1024	2048×2048	4096×4096
M	32	64	128	256
#Richardson	32	33	33	33
$\|\mathbf{e}_h^{int}\|_\infty$	3.74E-2	2.81E-3	4.04E-4	7.33E-5
$\|\mathbf{e}_h^{ext}\|_\infty$	4.26E-2	3.00E-3	4.09E-4	7.69E-5
CPU (secs)	7.53E+0	3.07E+1	1.24E+2	5.01E+2

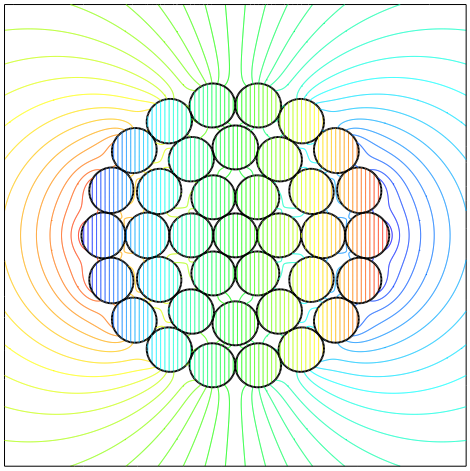


Fig. 12. Isolines plot of the potential (Example 5)

Table 11
Numerical results of Example 5 (37 cells) by the fourth-order method

grid size	512×512	1024×1024	2048×2048	4096×4096
M	32	64	128	256
#Richardson	35	34	34	34
$\|\mathbf{e}_h^{int}\|_\infty$	7.13E-3	5.89E-4	1.71E-5	9.04E-7
$\|\mathbf{e}_h^{ext}\|_\infty$	4.26E-3	5.66E-4	1.79E-5	7.77E-7
CPU (secs)	1.11E+1	3.99E+1	1.56E+2	6.48E+2

Example 6. The interface in this example is the union of a circular layer of uniform cells and a set of randomly generated cells, which are enclosed by the circular layer of cells; refer to Fig. 13. All of the cells are contained inside the unit circle centered at the origin. They are generated in the same way as those in the previous examples. We first generate cells some of which are right in touch with each other and then multiply the radii of the cells by 0.9999. The cells on the outer circular layers are almost in touch with the unit circle by a distance less than 10^{-4}. As before, the distances between those cells which are very close to each other are less than 10^{-4}, too. Fig. 13 shows isolines of a numerical potential $\Phi(\mathbf{p})$ for the interface problem. Numerical results are summarized in Tables 12-13. Table 12 has the results by the second-order version of the grid-based boundary integral method. Table 13 has the results by the fourth-order version of the grid-based boundary integral method.

The simulation corresponding to the plot in Fig. 13 has 48 cells/ellipses. The minimum and maximum radii of the cells are about 0.0828 and 0.1369, respectively. The minimum and maximum aspect ratios of the ellipses are about 0.6048 and 0.9478, respectively.

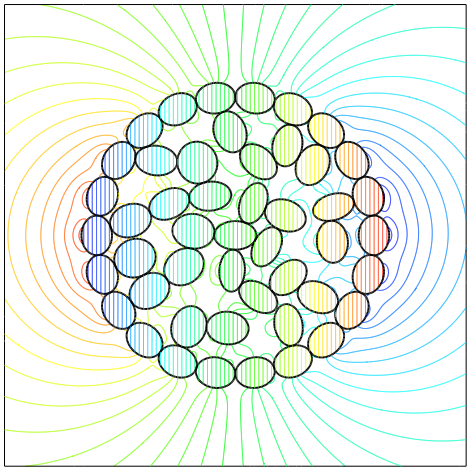


Fig. 13. Isolines plot of the potential (Example 6)

Table 12
Numerical results of Example 6 (48 cells) by the second-order method

grid size	512×512	1024×1024	2048×2048	4096×4096
M	32	64	128	256
#Richardson	36	35	35	36
$\|\mathbf{e}_h^{int}\|_\infty$	2.31E-1	2.47E-2	3.60E-3	2.51E-4
$\|\mathbf{e}_h^{ext}\|_\infty$	1.97E-1	2.13E-2	3.46E-3	2.69E-4
CPU (secs)	1.01E+1	3.91E+1	1.58E+2	6.54E+2

Table 13
Numerical results of Example 6 (48 cells) by the fourth-order method

grid size	512×512	1024×1024	2048×2048	4096×4096
M	32	64	128	256
#Richardson	38	35	36	36
$\|\mathbf{e}_h^{int}\|_\infty$	8.73E-2	3.54E-3	2.81E-4	9.72E-6
$\|\mathbf{e}_h^{ext}\|_\infty$	5.52E-2	2.80E-3	2.70E-4	9.42E-6
CPU (secs)	1.40E+1	4.82E+1	1.93E+2	7.89E+2

All numerical results for Examples 1-6 show that both second-order and fourth-order versions of the grid-based boundary integral method for the interface problem (1)-(5) generate solutions with the desired order of accuracy. The numerical results also show the grid-based boundary integral method is stable and efficient in the sense that the number of the Richardson iterations used is essentially independent of the grid sizes for each example. The computational work involved with the method is essentially linearly proportional to the num-

ber of nodes in the underlying Cartesian grid as indicated by the CPU times that the computer takes, which scale linearly with the number of grid nodes.

6 Discussion

In this work, we have presented a second-order version and a fourth-order version of a Cartesian grid-based boundary integral method for the interface problem (1)-(5) of the Laplace equation on closely packed cells. When the cells are closely packed, the boundary integrals involved with the boundary integral formulation for the interface problem become nearly singular. It is hard to get accurate values of the boundary integrals by a standard evaluation method such as the composite trapezoidal rule, which otherwise has super-algebraic convergence in accuracy [25]. The grid-based boundary integral method proposed in this work avoids direct evaluation of the nearly singular, singular and hyper-singular boundary integrals.

The second-order version of the method works with the five-point finite difference discretization as well as a quadratic two-variable Lagrange interpolation. The fourth-order version of the method works with the nine-point compact finite difference discretization as well as a quartic two-variable Lagrange interpolation. With both versions of the method, the discrete finite difference equations, whose right hand sides at irregular grid nodes are corrected, are solved with a fast Fourier transform based Poisson solver.

The computational work involved with the Cartesian grid-based boundary integral method for the interface problem (1)-(5) is essentially linearly proportional to the number of nodes on the Cartesian grid that is used to solve the equivalent interface problem (14). Nevertheless, the efficiency of the method for the interface problem (1)-(5) may be further improved after it is combined with a local and adaptive mesh refinement algorithm as well as a simple, fast summation technique for computing the Dirichlet boundary condition for the simple interface problem (14). This improvement will be studied and reported separately.

The method presented in this work is not kernel-free since the computation of the Dirichlet boundary condition on the boundary of the bounding box for the simple interface problem (14) needs to know and work with the fundamental solution of the Laplace equation in the free space. However, this method can be regarded as an extension of the kernel-free boundary integral method [50–52]. The kernel-free boundary integral method was proposed for solving boundary value and interface problems in bounded domains [50–52] while the current one is proposed for the interface problem (1)-(5) in the free space. The methodology should be applicable for solving exterior boundary

value problems of the Laplace equation or other elliptic partial differential equation as long as the fundamental solution of the PDE in the free space is known.

The grid-based boundary integral method can be readily extended for the interface problem or other exterior boundary value problems on closely packed cells in three space dimensions. It may have advantages for modeling multi-component flows, multiphase materials and evolution of microstructures [1–3, 7, 14, 18, 22, 26, 28, 32, 42, 43].

Acknowledgments

The first author is grateful for the inspiring discussion with John Strain about a higher-order version of the kernel-free boundary integral method. Research of the first author was supported in part by the National Science Foundation of the USA under Grant DMS–0915023 and is supported by the National Natural Science Foundation of China under Grants DMS–11101278 and DMS–91130012. Research of the first author is also supported by the Young Thousand Talents Program of China.

References

[1] N. Akaiwa, D. I. Meiron, Numerical-simulation of two-dimensional late-stage coarsening for nucleation and growth, Physical Review E 51 (1995) 5408–5421.

[2] N. Akaiwa, D. I. Meiron, Two-dimensional late-stage coarsening for nucleation and growth at high-area fractions, Physical Revew E 54 (1996) R13–R16.

[3] N. Akaiwa, K. Thornton, P. W. Voorhees, Large-scale simulations of microstructural evolution in elastically stressed solids, Journal of Computational Physics 173 (2) (2001) 61–86.

[4] T. Ashihara, T. Yao, T. Namba, M. Ito, T. Ikeda, A. Kawase, S. Toda, T. Suzuki, M. Inagaki, M. Sugimachi, M. Kinoshita, K. Nakazawa, Electroporation in a model of cardiac defibrillation, J. Cardiovasc. Electrophysiol. 12 (12) (2001) 1393–1403.

[5] K. E. Atkinson, The Numerical Solution of Integral Equations of the Second Kind, Cambridge University Press, Cambridge, UK, 1997.

[6] A. Barnett, Evaluation of layer potentials close to the boundary for Laplace and Helmholtz problems on analytic planar domains, to appear (2014).

[7] A. Barua, S. Li, H. Feng, X. Li, J. Lowengrub, An efficient rescaling algorithm for simulating the evolution of multiple elastic precipitates, Commun. Comput. Phys. 14 (4) (2013) 940–959.

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

[9] J. T. Beale, M. C. Lai, A method for computing nearly singular integrals, SIAM J. Numer. Anal. 38 (6) (2001) 1902–1925.

[10] B. L. Buzbee, G. H. Golub, C. W. Nielson, On direct methods for solving Poisson's equations, SIAM J. Numer. Anal. 7 (4) (1970) 627–656.

[11] F. W. Dorr, The direct solution of the discrete Poisson equation on a rectangle, SIAM Review 12 (2) (1970) 248–263.

[12] M. Gasca, T. Sauer, On the history of multivariate polynomial interpolation, Journal of Computational and Applied Mathematics 122 (2000) 23–35.

[13] M. Gasca, T. Sauer, Polynomial interpolation in several variables, Advances in Computational Mathematics 12 (2000) 377–410.

[14] A. Greenbaum, L. Greengard, G. B. McFadden, Laplace's equation and the Dirichlet-Neumann map in multiply connected domains, J. Comput. Phys. 105 (2) (1993) 267–278.

[15] S. Hao, A. H. Barnett, P. G. Martinsson, P. Young, High-order accurate methods for Nyström discretization of integral equations on smooth curves in the plane, Adv. Comput. Math. 40 (2014) 245–272.

[16] J. Helsing, R. Ojala, Corner singularities for elliptic problems: Integral equations, graded meshes, quadrature, and compressed inverse preconditioning, Journal of Computational Physics 227 (2008) 8820–8840.

[17] J. Helsing, R. Ojala, On the evaluation of layer potentials close to their sources, Journal of Computational Physics 227 (5) (2008) 2899–2921.

[18] T. Y. Hou, J. S. Lowengrub, M. J. Shelley, Boundary integral methods for multicomponent fluids and multiphase materials, Journal of Computational Physics 169 (2) (2001) 302–362.

[19] E. N. Houstis, T. S. Papatheodorou, Algorithm 543: FFT9, fast solution of Helmholtz-type partial differential equations D3, ACM Trans. Math. Softw. 5 (4) (1979) 490–493.

[20] E. N. Houstis, T. S. Papatheodorou, High-order fast elliptic equation solver, ACM Trans. Math. Software 5 (4) (1979) 431–441.

[21] G. C. Hsiao, W. L. Wendland, Boundary Integral Equations, Springer-Verlag Berlin Heidelberg, 2008.

[22] H.-J. Jou, P. H. Leo, J. S. Lowengrub, Microstructural evolution in inhomogeneous elastic media, Journal of Computational Physics 131 (1) (2003) 109–148.

[23] A. Klöckner, A. Barnett, L. Greengard, M. O'Neil, Quadrature by expansion: A new method for the evaluation of layer potentials, Journal of Computational Physics 252 (2013) 332–349.

[24] W. Krassowska, J. C. Neu, Response of a single cell to an external electric field, Biophysical Journal 66 (1994) 1768–1776.

[25] R. Kress, Linear Integral Equations, 2nd ed., Springer, New York, 1999.

[26] P. H. Leo, J. S. Lowengrub, H. J. Jou, A diffuse interface model for microstructural evolution in elastically stressed solids, Acta Materialia 46 (6) (1998) 2113–2130.

[27] R. J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations: steady-state and time-dependent problems, Society for Industrial and Applied Mathematics, 2007.

[28] S. Li, J. Lowengrub, P. H. Leo, A rescaling scheme with application to the long-time simulation of viscous fingering in a Hele-Shaw cell, Journal of Computational Physics 225 (1) (2007) 554–567.

[29] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numer. Anal. 21 (1984) 285–299.

[30] A. Mayo, Fast high order accurate solution of Laplace's equation on irregular regions, SIAM J. Sci. Statist. Comput. 6 (1985) 144–157.

[31] A. Mayo, The rapid evaluation of volume integrals of potential theory on general regions, J. Comput. Phys. 100 (1992) 236–245.

[32] G. B. McFadden, P. W. Voorhees, R. F. Boisvert, D. I. Meiron, A boundary integral method for the simulation of two-dimensional particle coarsening, Journal of Scientific Computing 1 (2) (1986) 117–144.

[33] L. M. Mir, M. F. Bureau, J. Gehl, R. Rangara, D. Rouy, J. M. Caillaud, P. Delaere, D. Branellec, B. Schwartz, D. Scherman, High-efficiency gene transfer into skeletal muscle mediated by electric pulses, Proc. Natl. Acad. Sci. USA 96 (8) (1999) 4262–4267.

[34] E. Neumann, S. Kakorin, K. Toensing, Fundamentals of electroporative delivery of drugs and genes, Bioelectrochem. Bioenerg. 48 (1) (1999) 3–16.

[35] C. Pozrikidis, Interfacial dynamics for Stokes flow, J. Comput. Phys. 169 (2001) 250–301.

[36] Y. Saad, Iterative methods for sparse linear systems, PWS Publishing Company, Boston, 1996.

[37] Y. Saad, Iterative Methods for Sparse Linear Systems, second edition, Society for Industrial and Applied Mathematics, 2003.

[38] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual method for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.

[39] G. Sersa, T. Cufer, M. Cemazar, M. Rebersek, R. Zvonimir, Electrochemotherapy with bleomycin in the treatment of hypernephroma metastasis: case report and literature review, Tumori 86 (2) (2000) 163–165.

[40] J. Strikwerda, Finite Difference Schemes and Partial Differential Equations, Society for Industrial and Applied Mathematics, 2007.

[41] P. N. Swarztrauber, The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle, SIAM Review 19 (3) (1977) 490–501.

[42] K. Thornton, N. Akaiwa, P. W. Voorhees, Large-scale simulations of Ostwald ripening in elastically stressed solids: I. development of microstructure, Physical Review E 52 (2004) 1353.

[43] K. Thornton, J. Argen, P. W. Voorhees, Modeling the evolution of phase boundaries in solids at the meso- and nano-scales, Acta Materilia 51 (19) (2003) 5675–5710.

[44] H. A. van der Vorst, BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution nonsymmetric linear systems, SIAM J. Sci. Stat. 13 (1992) 631–644.

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

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

[47] L. Ying, G. Biros, D. Zorin, A high-order 3d boundary integral equation solver for elliptic PDEs in smooth domains, Journal of Computational Physics 219 (1) (2006) 247–275.

[48] W.-J. Ying, J. T. Beale, A fast accurate boundary integral method for potentials on closely packed cells, Communications in Computational Physics 14 (4) (2013) 1073–1093.

[49] W.-J. Ying, C. S. Henriquez, Hybrid finite element method for describing the electrical response of biological cells to applied fields, IEEE Trans. Biomed. Engrg. 54 (4) (2007) 611–620.

[50] W.-J. Ying, C. S. Henriquez, A kernel-free boundary integral method for elliptic boundary value problems, J. Comp. Phy. 227 (2) (2007) 1046–1074.

[51] W.-J. Ying, W.-C. Wang, A kernel-free boundary integral method for implicitly defined surfaces, Journal of Computational Physics 252 (2013) 606–624.

[52] W.-J. Ying, W.-C. Wang, A kernel-free boundary integral method for variable coefficients elliptic PDEs, Communications in Computational Physics 15 (4) (2014) 1108–1140.

[53] A. Z. Zinchenko, R. H. Davis, An efficient algorithm for hydrodynamical interaction of many deformable drops, J. Comput. Phys. 157 (2000) 539–587.

[54] A. Z. Zinchenko, M. A. Rother, R. H. Davis, A novel boundary-integral algorithm for viscous interaction of deformable drops, Phys. Fluids 9 (1997) 1493–1511.