Fast Solvers for 3D Poisson Equations Involving Interfaces in a Finite or the Infinite Domain

Ming-Chih Lai [*] Zhilin Li[†] Xiaobiao Lin [‡]

Abstract

In this paper, numerical methods are proposed for Poisson equations defined in a finite or in the infinite domain in three dimensions. In the domain, there can exists an interface across which the flux and the solution are discontinuous. To deal with the discontinuity in the source term and in the flux, the original problem is transformed to a new one with a smooth solution. Such a transformation can be carried out easily through an extension of the jumps along the normal direction if the interface is expressed as the zero level set of a three dimensional function. An auxiliary sphere is used to separate the infinite region into an interior and exterior domain. The Kelvin's inversion is used to map the exterior domain into an interior domain. The two Poisson equations defined in the interior and the exterior written in spherical coordinates are solved simultaneously. By choosing the mesh size carefully and exploiting the fast Fourier transform, the resulting finite difference equations can be solved efficiently. The approach in dealing with the interface has also been used with the artificial boundary condition technique which truncates the infinite domain. Numerical results demonstrate second order accuracy of our algorithms.

Key words. arbitrary interface, fast 3D Poisson solver, immersed interface method, infinite domain, extension of jumps, spherical coordinates, level set function, artificial boundary condition.

AMS subject classifications. 65N06, 65N50

1 Introduction

In this paper, we consider the three-dimensional Poisson equation of the form

$$\Delta u(\mathbf{x}) = \begin{cases} \nabla\cdot\mathbf{M}_1(\mathbf{x}), & \text{if } (\mathbf{x})\in\Omega^-, \\ \nabla\cdot\mathbf{M}_2(\mathbf{x}), & \text{if } (\mathbf{x})\in\Omega^+ = \mathbb{R}^3-\Omega^-, \end{cases} \qquad \lim_{|\mathbf{x}|\to\infty} u(\mathbf{x}) = 0. \tag{1.1}$$

We assume that $\mathbf{M}_i(\mathbf{x}), 1\le i\le 2$, is continuous and bounded in each domain of its definition. In many applications, we simply have $\mathbf{M}_2(\mathbf{x}) = \mathbf{0}$. Across the interface between Ω^- and Ω^+, there is a finite jump in the flux of the solution

$$[u_n] = \left[\frac{\partial u}{\partial \mathbf{n}}\right] = [\,\nabla u\cdot\mathbf{n}\,] = [\,(\mathbf{M}_2-\mathbf{M}_1)\cdot\mathbf{n}\,], \tag{1.2}$$

[*]Department of Applied Mathematics, National Chiao Tung University, Taiwan, mclai@math.nctu.edu.tw

[†]Center for Research in Scientific Computation & Department of Mathematics North Carolina State University, Raleigh, NC 27695, e-mail: zhilin@math.ncsu.edu

[‡]Department of Mathematics, North Carolina State University, Raleigh, NC 27695, e-mail: xblin@math.ncsu.edu

where $\mathbf{n}$ is the unit normal direction pointing outward along the interface $\Gamma = \partial\Omega^-$. The jump is defined as the difference of the limiting values of the quantity between the Ω^+ side and the Ω^- side. Figure 1 illustrates the description of the problem in the case of $\mathbf{M}_2 = \mathbf{0}$ and $\nabla \cdot \mathbf{M}_1 = f(\mathbf{x})$. The jump conditions can be derived either through physical reasoning or mathematical derivations, see [5, 11, 12].

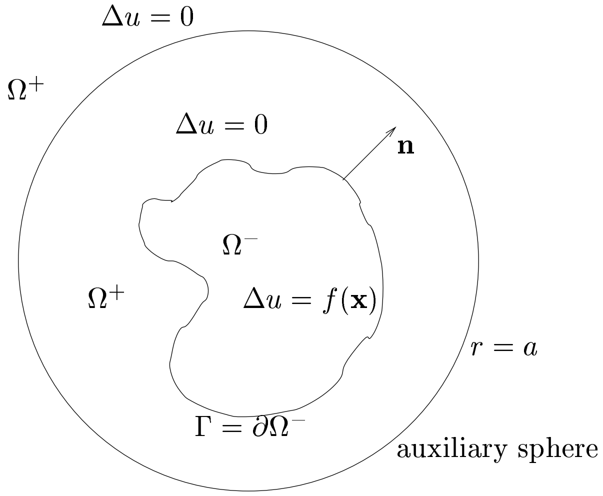

Figure 1: A diagram of the problem with $\mathbf{M}_2 = \mathbf{0}$ and $\nabla \cdot \mathbf{M}_1 = f(\mathbf{x})$ in a bounded domain Ω^-. The problem is defined in the entire space. An auxiliary sphere $r = a$ is used to divide the infinite domain into two parts.

Modeling and simulation of ferromagnetic materials, see [5, 3], is one of many applications of the problem stated above. Ferromagnetic materials are widely used as recording media. They are also currently being explored as an alternative to semi-conductors as memory devices. Let $\mathcal{M}$ be the magnetization vector field, then the magnetic field is given by $-\nabla u$, where the potential u satisfies the Poisson equation

$$\Delta u = \nabla \cdot (\mathcal{M}\chi_{\Omega^-}), \tag{1.3}$$

where χ_{Ω^-} is the characteristic function of Ω^-. Here, Ω^- is the domain occupied by the ferromagnetic material, see [5] for a detailed description. Therefore, the source term has a delta function singularity which causes a discontinuity in the flux of the solution across the boundary of Ω^-. The domain Ω^- is typically a closed surface (not necessarily a sphere).

It is important to provide efficient and reliable numerical solutions for the equation above. As quoted from [5], *"Even though there is an extensive literature on numerical simulations using micro-magnetics, see [1] for a review, with even commercially available software, the accuracy of these numerical computations is still a serious issue."*

In this paper, we propose second order accurate fast algorithms based on the fast Fourier transform (FFT) and the immersed interface method (IIM) [10, 12, 15, 13, 16] to solve the three dimensional Poisson equation with discontinuities in the flux and the source term in the entire space.

To solve (1.1)-(1.2) numerically, we need to overcome two difficulties associated with the problem. The first one is how to solve the Poisson equation in the infinite domain. The second one is how to handle the discontinuities in the source term and in the flux of the solution.

Finite difference approaches are often preferred for a number of reasons. They are simple to implement and fast when coupled with FFT. For non-linear problems in which a Green function may not be available, or for free boundary/moving interface problems, finite difference methods with fixed underlying grids are more attractive. The challenge is how to maintain the accuracy near the boundary or the interface.

For non-interface problems defined in an infinite domain, we refer the readers to [20] for a review on the regularity of the solution and a variety of numerical methods, particularly, the artificial boundary condition (ABC) techniques. Using ABC techniques, one needs to truncate the infinite domain and then to impose some boundary conditions at the truncated boundary. In [2], the authors derived a sequence of different accurate boundary conditions for Helmholtz equations along the artificial sphere. Some other artificial boundary conditions can be found in [20].

In this paper, we propose an algorithm that can solve the Poisson equation on the entire three dimensional space. The main idea is to use an auxiliary sphere to separate the infinite domain and use the Kelvin's inversion to transform the exterior domain to a finite interior domain. The mesh points corresponding to the domain outside of the sphere is carefully chosen so that the finite difference discretizations from both sides can match together. The resulting finite difference equations can be further simplified by using the discrete Fourier transform in the θ coordinate. Then the generalized cyclic reduction method can be used to solve the remaining block tri-diagonal linear system. Comparing with the ABC approach that solves the solution only inside a sphere, our new method can get an approximate solution in the entire space, but the computation cost is not necessarily doubled. More importantly, we do not have to worry about how big the artificial sphere is or what kind of boundary conditions should be, which makes the algorithm potentially more efficient.

To deal with the discontinuity across the interface $\Gamma = \partial\Omega^-$, we generalize the method developed in [16] for two dimensional problems to spherical coordinates. The main idea is to extend the jumps in the solution and the flux along the normal line of the interface in a neighborhood of Γ. Then, we can construct a new function that has the same jump conditions as the solution of (1.1)-(1.2). Using a transformation, we get a new interface problem whose solution is smooth across the interface. Therefore, a second order accurate scheme can be obtained using the standard central finite difference scheme in spherical coordinates with minor modifications. The key to the success of this approach is how to get the orthogonal projections of the grid points (in the neighborhood of the interface) on the interface which is not trivial when the geometry is described in spherical coordinates.

Other possible approaches include the use of an integral equation like the fast multipole method [6], since the solution to the Poisson equation can be expressed as a convolution of the fundamental solution with the source term f. For the ferromagnetic materials simulation, a boundary correction method based on the volume integral has been used in [3].

The paper is organized as follows. In Sec. 2, we describe the transformation based on an orthogonal extension of known jumps. The solution to the new Poisson problem is smooth. An efficient fast Poisson solver in the whole space and the artificial boundary condition technique are explained in Sec. 3. The entire algorithm for the interface problem defined in a finite or the infinite domain is outlined in Sec. 4. Numerical examples and grid refinement analysis are provided in

Sec. 5.

2 A transformation based on the orthogonal extension of the jumps

Without loss of generality, we consider the following more general problem

$$
\begin{aligned}
&\Delta u(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}), & \text{if } \mathbf{x} \in \Omega^-, \\ f_2(\mathbf{x}), & \text{if } \mathbf{x} \in \Omega^+ = \mathbb{R}^3 - \Omega^-, \end{cases} \\
&[u] = w, \quad [u_n] = v, \qquad \lim_{|\mathbf{x}| \to \infty} u(\mathbf{x}) = 0
\end{aligned}
\tag{2.4}
$$

where $w \in C^2(\Gamma)$ and $v \in C^2(\Gamma)$ are two functions defined only on the interface $\Gamma = \partial\Omega^-$. First, we briefly review how to transformation the above interface problem to a problem with a smooth solution. This was first introduced in [16] for two dimensional problems.

Let $\varphi(\mathbf{x})$ be a real-valued function such that

$$
\varphi(\mathbf{x}) \begin{cases} < 0, & \text{if } \mathbf{x} \in \Omega^-, \\ = 0, & \text{if } \mathbf{x} \in \Gamma = \partial\Omega^-, \\ > 0, & \text{if } \mathbf{x} \in \Omega^+. \end{cases} \tag{2.5}
$$

We assume that $\varphi(\mathbf{x}) \in C^3(\mathbb{R}^3)$ in a neighborhood of the interface Γ[1], so that the zero level set of φ, $\varphi(\mathbf{x}) = 0$, represents the interface. Usually, the level set function is chosen as the signed distance function ($|\nabla\varphi| = 1$) from the interface, see [17, 18] and the references therein. Therefore, it is reasonable to assume that $|\nabla\varphi| \neq 0$ in the neighborhood of the interface Γ. We define the extensions of $w(\mathbf{X}(s))$ and $v(\mathbf{X}(s))$ along the normal line (both directions) as

$$
w_e(\mathbf{x}) = w_e\left(\mathbf{X}(s) + \alpha\mathbf{n}\right) = w(\mathbf{X}(s)), \tag{2.6}
$$

and

$$
v_e(\mathbf{x}) = v_e\left(\mathbf{X}(s) + \alpha\mathbf{n}\right) = v(\mathbf{X}(s)), \tag{2.7}
$$

for all $\alpha \in R$ such that the normal lines do not intersect[2], where $\mathbf{n}$ is the unit normal direction pointing outward. We construct the following function based on the extension

$$
\tilde{u}(\mathbf{x}) = w_e(\mathbf{x}) + v_e(\mathbf{x}) \frac{\varphi(\mathbf{x})}{|\nabla\varphi(\mathbf{x})|}. \tag{2.8}
$$

Note that $\tilde{u}(\mathbf{x}) \in C^2$ in the neighborhood of the interface Γ since we assume that $w(s)$, $v(s)$ are in C^2 in the domain of the definition, and φ is in $C^3(\mathbb{R}^3)$ in the neighborhood of the interface Γ. Let us also define

$$
\hat{u}(\mathbf{x}) = H(\varphi(\mathbf{x}))\tilde{u}(\mathbf{x}) = \begin{cases} 0, & \text{if } \varphi(\mathbf{x}) < 0, \\ \frac{1}{2}\tilde{u}(\mathbf{x}), & \text{if } \varphi(\mathbf{x}) = 0, \\ \tilde{u}(\mathbf{x}), & \text{if } \varphi(\mathbf{x}) > 0, \end{cases} \tag{2.9}
$$

[1] In implementation, $\varphi(\mathbf{x}) \in C^2$ seems to be enough for second order accuracy.

[2] Theoretically, there is always a neighborhood that the normal line do not intersect. Numerically, in case the normal lines intersect, we can simply pick up an extension and the algorithm still works.

in the same neighborhood in which $\tilde{u}(\mathbf{x})$ is well defined, where $H(\cdot)$ is the Heaviside function. We have the following theorem.

Theorem 2.1 *Let $u(\mathbf{x})$ be the solution of (2.4), $\hat{u}(\mathbf{x})$ be defined in (2.9). Define $q(\mathbf{x}) = u(\mathbf{x}) - \hat{u}(\mathbf{x})$. Then in the neighborhood of the interface where $w_e(\mathbf{x})$ and $v_e(\mathbf{x})$ are well defined, the following are true:*

$$\Delta q(\mathbf{x}) = f(\mathbf{x}) - H(\varphi(\mathbf{x}))\, \Delta\hat{u}(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^3 - \Gamma, \tag{2.10}$$

$$[q] = 0, \qquad [\nabla q \cdot \boldsymbol{\tau}] = 0, \qquad [\nabla q \cdot \mathbf{n}] = 0, \tag{2.11}$$

where $\boldsymbol{\tau}$ is any unit tangent direction of the interface Γ. In other words, the new function $q(\mathbf{x})$ is a smooth (C^1) function across the interface Γ.

The proof can be found in [16].

2.1 The orthogonal projection on the interface in spherical coordinates

Using Theorem 2.1, we can transform the original interface problem to a new one with a smooth solution. In order to take advantage of the transformation, we need to find the extensions (2.6)-(2.7) along the normal line of the surface Γ to get $\tilde{u}$ and $\hat{u}$. Since our fast Poisson solver is in spherical coordinates, it is natural to find the orthogonal projection of a point near the interface using the information (level set function and the underlying grid) in spherical coordinates.

The spherical coordinates system is

$$\begin{aligned} x &= r\, \sin\phi\, \cos\theta, \\ y &= r\, \sin\phi\, \sin\theta, \qquad r \geq 0, \quad 0 \leq \theta < 2\pi, \quad 0 \leq \phi < \pi. \\ z &= r\, \cos\phi. \end{aligned}$$

Let $\mathbf{x}$ be a point near the interface Γ, and $\mathbf{x}^*$ be the corresponding orthogonal projection of $\mathbf{x}$ on the interface, both in spherical coordinates. We can write

$$\mathbf{x}^* = \mathbf{x} + \alpha\mathbf{p}, \tag{2.12}$$

where α is a scalar, and $\mathbf{p}$ is a direction to be determined. If the direction $\mathbf{p}$ is known, we can use the following quadratic equation

$$\varphi(\mathbf{x}) + (\nabla\varphi(\mathbf{x}) \cdot \mathbf{p})\, \alpha + \frac{1}{2}\, \left(\mathbf{p}^T He(\varphi(\mathbf{x}))\, \mathbf{p}\right)\, \alpha^2 = 0, \tag{2.13}$$

to approximate the scalar α, where $\nabla\varphi$, and the Hessian matrix $He(\varphi)$ are evaluated at $\mathbf{x}$. The above equation is a third order accurate approximation to $\varphi(\mathbf{x}^*) = 0$. The key here is how to decide the right direction $\mathbf{p}$ in order to get an approximation of the orthogonal projection $\mathbf{x}^*$.

Let $[\Delta x, \Delta y, \Delta z]^T$ and $[\Delta r, \Delta\phi, \Delta\theta]^T$ be the first order approximation of the vector $\mathbf{x} - \mathbf{x}^*$ in Cartesian and spherical coordinates, respectively. Then we have

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \approx \begin{bmatrix} \sin\phi\cos\theta & r\cos\phi\cos\theta & -r\sin\phi\sin\theta \\ \sin\phi\sin\theta & r\cos\phi\sin\theta & r\sin\phi\cos\theta \\ \cos\phi & -r\sin\phi & 0 \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta\phi \\ \Delta\theta \end{bmatrix}.$$

The columns of the Jacobian matrix above are orthogonal and the Euclidean norms are 1, r, and $|r\sin\phi|$, respectively. Therefore, we get

$$d^2 = \|(x,y,z)-(x^*,y^*,z^*)\|^2 \approx (\Delta x)^2+(\Delta y)^2+(\Delta z)^2 = (\Delta r)^2+(r\,\Delta\phi)^2+(r\sin\phi\Delta\theta)^2.$$

Since $\varphi(r^*,\phi^*,\theta^*)=0$, we have $\varphi(r,\phi,\theta)=\varphi(r,\phi,\theta)-\varphi(r^*,\phi^*,\theta^*)$ and

$$\begin{aligned} \varphi(r,\phi,\theta) &= \varphi_r\,\Delta r+\varphi_\phi\,\Delta\phi+\varphi_\theta\,\Delta\theta \\ &= \varphi_r\,\Delta r+\frac{\varphi_\phi}{r}\,(r\,\Delta\phi)+\frac{\varphi_\theta}{r\,\sin\phi}\,(r\,\sin\phi\,\Delta\theta), \end{aligned} \tag{2.14}$$

where we have dropped the higher order terms. Using the Cauchy-Schwartz's inequality, we have

$$|\varphi| \le \sqrt{(\varphi_r)^2+\left(\frac{\varphi_\phi}{r}\right)^2+\left(\frac{\varphi_\theta}{r\,\sin\phi}\right)^2}\cdot\sqrt{(\Delta r)^2+(r\,\Delta\phi)^2+(r\,\sin\phi\,\Delta\theta)^2}.$$

or in the limit case, we have

$$\frac{|\varphi|}{|\nabla\varphi|} \le \sqrt{(\Delta r)^2+(r\,\Delta\phi)^2+(r\,\sin\phi\,\Delta\theta)^2}.$$

The minimum of d can be reached if the equal sign can be reached since $|\varphi|/\nabla\varphi|$ is an approximation to the distance between (r,ϕ,θ) and its orthogonal projection (r^*,ϕ^*,θ^*). The equal sign can be reached if and only if the two vectors in the inner product (2.14) are collinear, i.e., there is a scalar α such that

$$\begin{aligned} r-r^* &= \alpha\,\varphi_r, \\ \phi-\phi^* &= \alpha\,\frac{\varphi_\phi}{r^2}, \\ \theta-\theta^* &= \alpha\,\frac{\varphi_\theta}{(r\,\sin\phi)^2}. \end{aligned}$$

Thus we concluded that the projection is along the direction

$$\mathbf{p} = \left[\varphi_r,\ \frac{\varphi_\phi}{r^2},\ \frac{\varphi_\theta}{(r\,\sin\phi)^2}\right]^T, \tag{2.15}$$

with an undetermined scalar factor α which can find an approximation using (2.13).

3 Fast solvers for 3D Poisson equation in the infinite domain

The Poisson equation in the three-dimensional infinite domain R^3 can be written in spherical coordinates as

$$\frac{\partial^2 u}{\partial r^2}+\frac{2}{r}\frac{\partial u}{\partial r}+\frac{1}{r^2}\frac{\partial^2 u}{\partial\phi^2}+\frac{\cot\phi}{r^2}\frac{\partial u}{\partial\phi}+\frac{1}{r^2\sin^2\phi}\frac{\partial^2 u}{\partial\theta^2}=f(r,\phi,\theta). \tag{3.16}$$

A vanishing condition at infinity

$$u\to 0 \qquad \text{as} \quad r\to\infty, \tag{3.17}$$

is imposed in order to ensure the uniqueness of the solution.

In this section, we will introduce two different fast direct methods for Eqs. (3.16)-(3.17). Both methods involve second-order finite difference discretizations to the Poisson equation (3.16) which can be solved efficiently when the FFT-based fast solver [9] is implemented. The only difference of the two methods is whether the problem is solved in the whole space or just within a truncated domain.

3.1 A new approach without truncation

First, we present a new approach which does not involve any truncation of the infinite domain. Thus, a numerical solution can be obtained in the whole three-dimensional space. Another important advantage of this approach is that there is no need to impose any artificial boundary conditions since the vanishing condition can be treated naturally in our finite difference setting. Also, the second-order accuracy is preserved in the whole domain.

In [9], Lai et. al. have developed a 3D FFT-based fast direct solver for the equation (3.16) on a bounded spherical domain $\{r \le a\}$. This solver has three major features; namely, there are no pole conditions, a fast linear algebraic solver can be applied, and the scheme is able to handle different boundary conditions. In this subsection, we will develop the Poisson solver on the entire three-dimensional space based on the method on a finite domain described in [9] and the Kevin's transformation.

Before we proceed, let us decompose the infinite domain R^3 into two regions using a sphere which is centered at the origin with radius a. Let the interior domain be $\Omega_a = \{r \le a\}$ and the exterior domain be $\Omega_a^c = \{r > a\}$, respectively, so that $R^3 = \Omega_a \cup \Omega_a^c$. Then the boundary $\partial\Omega_a$ will be $\{r = a\}$. The present idea is to transform the part of Poisson equation in the exterior unbounded domain Ω_a^c to a Poisson equation in a bounded domain. To accomplish that, we introduce an inversion mapping $\bar{r} = a^2/\xi$, or $(\bar{r}, \phi, \theta) \to (a^2/\xi, \phi, \theta)$ that maps the exterior domain $\{\xi \ge a\}$ into the interior domain $\{\bar{r} \le a\}$. Then, we define the functions $\bar{u}$ and $\bar{f}$ in Ω_a whose values are related to the functions u and f in Ω_a^c by

$$\bar{u}(\bar{r}, \phi, \theta) = \frac{a^2}{\bar{r}}\, u\left(\frac{a^2}{\bar{r}}, \phi, \theta\right), \qquad \bar{f}(\bar{r}, \phi, \theta) = \frac{a^2}{\bar{r}}\, f\left(\frac{a^2}{\bar{r}}, \phi, \theta\right), \quad \bar{r} \le a. \tag{3.18}$$

The above transform (3.18) is called the Kelvin's inversion. The functions $\bar{u}$ and $\bar{f}$ have the same regularities in the domain Ω_a as the functions u and f in Ω_a^c. Also, after applying the Kelvin's transform, the Laplace operator is preserved as

$$\begin{aligned} & u_{\xi\xi} + \frac{2}{\xi}u_\xi + \frac{1}{\xi^2}u_{\phi\phi} + \frac{\cot\phi}{\xi^2}u_\phi + \frac{1}{\xi^2\sin^2\phi}u_{\theta\theta} \\ = \; & \frac{\bar{r}^5}{a^6}\left(\bar{u}_{\bar{r}\bar{r}} + \frac{2}{\bar{r}}\bar{u}_{\bar{r}} + \frac{1}{\bar{r}^2}\bar{u}_{\phi\phi} + \frac{\cot\phi}{\bar{r}^2}\bar{u}_\phi + \frac{1}{\bar{r}^2\sin^2\phi}\bar{u}_{\theta\theta}\right). \end{aligned}$$

One can directly conclude that if u is harmonic in Ω_a^c, then $\bar{u}$ is harmonic for every points in Ω_a except the origin. Therefore, instead of solving one Poisson equation in the infinite domain, we solve the following two Poisson equations in the finite domain Ω_a,

$$u_{rr} + \frac{2}{r}u_r + \frac{1}{r^2}u_{\phi\phi} + \frac{\cot\phi}{r^2}u_\phi + \frac{1}{r^2\sin^2\phi}u_{\theta\theta} = f, \quad \text{in} \quad \Omega_a, \tag{3.19}$$

$$\bar{u}_{\bar{r}\bar{r}} + \frac{2}{\bar{r}}\bar{u}_{\bar{r}} + \frac{1}{\bar{r}^2}\bar{u}_{\phi\phi} + \frac{\cot\phi}{\bar{r}^2}\bar{u}_\phi + \frac{1}{\bar{r}^2\sin^2\phi}\bar{u}_{\theta\theta} = \frac{a^4}{\bar{r}^4}\bar{f}, \quad \text{in} \quad \Omega_a. \tag{3.20}$$

The above two equations are coupled at the boundary $r = a$ by the condition $\bar{u}(a, \phi, \theta) = a\,u(a, \phi, \theta)$. The vanishing boundary condition of (3.17) is transformed to some condition of $\bar{u}$ at the origin, which is not specified explicitly. We shall see this does not cause any trouble in our finite difference scheme later. Notice that the values of u and $\bar{u}$ and their derivatives should agree with each other at the boundary $\partial\Omega_a$ if $\bar{u}$ is transferred back to the original equation and the geometry.

We now discuss how to solve Eqs. (3.19)-(3.20) based on the fast direct solver developed in [9]. First, we choose a grid

$$(r_i, \phi_j, \theta_k) = \Big(i\Delta r,\ (j-1/2)\Delta\phi, k\Delta\theta\Big), \qquad 1 \le i \le M,\ 1 \le j \le L,\ 1 \le k \le N, \tag{3.21}$$

where the mesh spacings for each direction are

$$\Delta r = a/M, \quad \Delta\phi = \pi/L, \quad \Delta\theta = 2\pi/N. \tag{3.22}$$

In order to match the position of the ghost grid point in radial direction, the mesh width $\Delta\bar{r}$ is chosen to satisfy

$$a - \Delta\bar{r} = \frac{a^2}{a + \Delta r}. \tag{3.23}$$

This requirement has the meaning that the ghost grid point of u outside the boundary $\{r = a\}$ will be the grid point of $\bar{u}$ which is the closest one to the boundary. By a simple calculation, we obtain $\Delta\bar{r} = a/(M+1)$. So the grid points used for $\bar{u}$ can be chosen similarly by

$$(\bar{r}_i, \phi_j, \theta_k) = \Big(i\Delta\bar{r},\ (j-1/2)\Delta\phi, k\Delta\theta\Big), \qquad 1 \le i \le M+1,\ 1 \le j \le L,\ 1 \le k \le N. \tag{3.24}$$

Notice that, by the choice of Δr and $\Delta\bar{r}$, we have $r_M = \bar{r}_{M+1} = a$ and thus the coupling boundary condition is $\bar{u}_{M+1,j,k} = a\,u_{M,j,k}$.

Once we have set up the grid, we can apply the standard second-order centered finite difference approximation to the equations (3.19) and (3.20). This leads to the following finite difference equations

$$\begin{aligned} &\frac{u_{i+1,j,k} - 2u_{ijk} + u_{i-1,j,k}}{(\Delta r)^2} + \frac{2}{r_i}\,\frac{u_{i+1,j,k} - u_{i-1,j,k}}{2\Delta r} + \frac{1}{r_i^2}\,\frac{u_{i,j+1,k} - 2u_{ijk} + u_{i,j-1,k}}{(\Delta\phi)^2} \\ &+ \frac{\cot\phi_j}{r_i^2}\,\frac{u_{i,j+1,k} - u_{i,j-1,k}}{2\Delta\phi} + \frac{1}{r_i^2\sin^2\phi_j}\,\frac{u_{i,j,k+1} - 2u_{ijk} + u_{i,j,k-1}}{(\Delta\theta)^2} = f_{ijk}, \end{aligned} \tag{3.25}$$

$$\begin{aligned} &\frac{\bar{u}_{i+1,j,k} - 2\bar{u}_{ijk} + \bar{u}_{i-1,j,k}}{(\Delta\bar{r})^2} + \frac{2}{\bar{r}_i}\,\frac{\bar{u}_{i+1,j,k} - \bar{u}_{i-1,j,k}}{2\Delta\bar{r}} + \frac{1}{\bar{r}_i^2}\,\frac{\bar{u}_{i,j+1,k} - 2\bar{u}_{ijk} + \bar{u}_{i,j-1,k}}{(\Delta\phi)^2} \\ &+ \frac{\cot\phi_j}{\bar{r}_i^2}\,\frac{\bar{u}_{i,j+1,k} - \bar{u}_{i,j-1,k}}{2\Delta\phi} + \frac{1}{\bar{r}_i^2\sin^2\phi_j}\,\frac{\bar{u}_{i,j,k+1} - 2\bar{u}_{ijk} + \bar{u}_{i,j,k-1}}{(\Delta\theta)^2} = \frac{a^4}{\bar{r}_i^4}\bar{f}_{ijk}. \end{aligned} \tag{3.26}$$

Since the the solution is periodic in θ direction, we can approximate u by the truncated Fourier series

$$u(r_i, \phi_j, \theta_k) = \sum_{n=-N/2}^{N/2-1} u_n(r_i, \phi_j)\, e^{in\theta_k}, \tag{3.27}$$

where $u_n(r_i, \phi_j)$ is the discrete Fourier coefficient given by

$$u_n(r_i, \phi_j) = \frac{1}{N} \sum_{k=0}^{N-1} u(r_i, \phi_j, \theta_k)\, e^{-in\theta_k}. \tag{3.28}$$

The above transformation between the physical space and Fourier space can be efficiently performed using the FFT with $O(N \log_2 N)$ arithmetic operations. The expansions for the functions $f, \bar{u}$, and $\bar{f}$ are similar.

Substituting those expansions into the standard centered finite difference scheme (3.25)-(3.26) and equating the corresponding Fourier coefficients, the finite difference equations for the nth Fourier coefficients u_n and $\bar{u}_n$ are as follows

$$\begin{aligned} &\frac{U_{i+1,j} - 2\,U_{ij} + U_{i-1,j}}{(\Delta r)^2} + \frac{2}{r_i}\frac{U_{i+1,j} - U_{i-1,j}}{2\,\Delta r} + \frac{1}{r_i^2}\frac{U_{i,j+1} - 2\,U_{ij} + U_{i,j-1}}{(\Delta\phi)^2} \\ &\quad + \frac{\cot\phi_j}{r_i^2}\frac{U_{i,j+1} - U_{i,j-1}}{2\Delta\phi} + \frac{1}{r_i^2 \sin^2\phi_j}\frac{2\cos(n\Delta\theta) - 2}{(\Delta\theta)^2}\,U_{ij} = F_{ij}, \end{aligned} \tag{3.29}$$

$$\begin{aligned} &\frac{\bar{U}_{i+1,j} - 2\,\bar{U}_{ij} + \bar{U}_{i-1,j}}{(\Delta\bar{r})^2} + \frac{2}{\bar{r}_i}\frac{\bar{U}_{i+1,j} - \bar{U}_{i-1,j}}{2\,\Delta\bar{r}} + \frac{1}{\bar{r}_i^2}\frac{\bar{U}_{i,j+1} - 2\,\bar{U}_{ij} + \bar{U}_{i,j-1}}{(\Delta\phi)^2} \\ &\quad + \frac{\cot\phi_j}{\bar{r}_i^2}\frac{\bar{U}_{i,j+1} - \bar{U}_{i,j-1}}{2\Delta\phi} + \frac{1}{\bar{r}_i^2 \sin^2\phi_j}\frac{2\cos(n\Delta\theta) - 2}{(\Delta\theta)^2}\,\bar{U}_{ij} = \frac{a^4}{\bar{r}_i^4}\bar{F}_{ij}. \end{aligned} \tag{3.30}$$

Here, we denote the discrete values by $U_{ij} = u_n(r_i, \phi_j)$, $\bar{U}_{ij} = \bar{u}_n(\bar{r}_i, \phi_j)$, $F_{ij} = f_n(r_i, \phi_j)$ and $\bar{F}_{ij} = \bar{f}_n(\bar{r}_i, \phi_j)$.

We now discuss how to handle the numerical boundary conditions for $\phi = 0$ and $\phi = \pi$. Using a staggered grid in the ϕ direction, we do not put the grid points on the north ($\phi = 0$) and south ($\phi = \pi$) poles directly. Thus, the numerical boundary values for U_{i0} and $U_{i,L+1}$ can be easily obtained by the symmetry constraint of Fourier coefficients [9]; that is, $U_{i0} = (-1)^n U_{i1}$ and $U_{i,L+1} = (-1)^n U_{iL}$. Similarly, we have $\bar{U}_{i0} = (-1)^n \bar{U}_{i1}$ and $\bar{U}_{i,L+1} = (-1)^n \bar{U}_{iL}$. Therefore, no complicated pole treatments are needed in our setting. More surprisingly, the numerical boundary values U_{0j} and $\bar{U}_{0j}$ will not be needed either since the coefficients of those values in the finite difference equations (3.29)-(3.30) are zero. One can easily check this from the above finite difference equations in the case of $i = 1$. The numerical boundary conditions for U_{Mj} and $\bar{U}_{M+1,j}$ are provided by the coupling boundary condition $\bar{U}_{M+1,j} = a\,U_{Mj}$. By ordering the unknowns appropriately, the resulting linear system of equations for U_{ij} and $\bar{U}_{ij}$ can be solved altogether by the generalized cyclic reduction algorithm [19] with $O(LM \log_2 M)$ operations.

3.2 An artificial boundary approach

The traditional numerical computation for (3.16)-(3.17) is to truncate the unbounded region by a sphere, say $r = a$; then an artificial boundary condition is imposed at $r = a$. The resulting problem is then discretized and solved in the bounded region $\{r \leq a\}$. However, the main difficulty is how to choose the artificial boundary condition at $r = a$ so that the vanishing condition at infinity can be translated appropriately to the boundary. In [2], the authors developed a sequence of local boundary conditions for some elliptic equations. Those artificial boundary conditions can be considered as applying the differential operators to the solution at the boundary. The order of accuracy of those boundary conditions depends on the order of the highest derivative in the differential operators. In this paper, we use the following second-order accurate boundary condition

$$\frac{\partial^2 u}{\partial r^2} + \frac{4}{r}\frac{\partial u}{\partial r} + \frac{2u}{r^2} = 0, \qquad \text{at } r = a. \tag{3.31}$$

The detail of the derivation of the above condition can be found in [2]. In short, we now solve the equation (3.16) in a finite domain $\{r \leq a\}$ with the boundary condition (3.31).

The finite difference discretization for the resulting problem is almost the same as the one described before. That is, we use the same grid as described in (3.21)-(3.22) and follow the same solution procedures to solve one Poisson equation in the domain $\{r \leq a\}$. The treatments for numerical boundary values are also the same except at the boundary $r = a$ which uses the condition (3.31) instead. Therefore, the numerical boundary value of $U_{M+1,j}$ can be found by the second-order approximation of (3.31)

$$\frac{U_{M+1,j} - 2\,U_{Mj} + U_{M-1,j}}{(\Delta r)^2} + \frac{4}{r_M}\frac{U_{M+1,j} - U_{M-1,j}}{2\,\Delta r} + \frac{2\,U_{Mj}}{r_M^2} = 0. \tag{3.32}$$

Once we incorporate the numerical boundary values into the difference scheme, the resulting linear equations are again solved by the generalized cyclic reduction method.

Remark 3.1 *If the domain is a finite $r \leq a$ instead of the infinite domain, then we can apply the exact boundary condition at $r = a$ instead of the artificial boundary condition.*

Let us close this section by summarizing the algorithms and the operation counts in the following three steps:

1. Compute the Fourier coefficients for the right-hand side function as in (3.28) by FFT, which requires $O(MLN \log_2 N)$ operations.

2. Solve the block tridiagonal linear systems for N Fourier coefficients by the generalized cyclic reduction algorithm, which requires $O(NLM \log_2 M)$ operations.

3. Invert the Fourier coefficients as in (3.27) by FFT to obtain the solution, which requires $O(MLN \log_2 N)$ operations.

The overall operation count is roughly $O(NLM \log_2(M + N))$ for $M \times L \times N$ grid points. One should notice that the artificial boundary condition approach saves about the half of the work comparing with the approach without truncation since the former does not compute the solution outside the truncating boundary.

4 Dealing with the interface and an outline of the algorithm.

In this section, we explain how to handle the interface and give an outline of our algorithm. For the interface problem (2.4), using the transformation described in Section 2 and a level set function to express the interface, we can get a discrete Poisson equation with a modified right hand side. The discrete Poisson equation can be solved either in the entire space or in a sphere $r \leq a$ using an artificial boundary condition technique. Below we explain our algorithm step by step.

Step 1: Indexing grid points. First we generate a grid (3.21)-(3.22) and choose a sphere $r = a$ that encloses the interface $\Gamma = \partial\Omega^-$. A level set function whose zero level set ($\varphi(r,\varphi,\theta) = 0$) is the interface Γ, is defined at grid points as φ_{ijk}. If we only want to obtain the solution in a bounded domain, then we have to choose the artificial sphere $r = a$ carefully in order to use an artificial boundary condition technique while maintaining second order accuracy.

Let

$$\begin{aligned} \varphi_{ijk}^{max} &= \max\{\varphi_{i-1,j,k}, \varphi_{i,j,k}, \varphi_{i+1,j,k}, \varphi_{i,j-1,k}, \varphi_{i,j+1,k}, \varphi_{i,j,k-1}, \varphi_{i,j,k+1}\}, \\ \varphi_{ijk}^{min} &= \min\{\varphi_{i-1,j,k}, \varphi_{i,j,k}, \varphi_{i+1,j,k}, \varphi_{i,j-1,k}, \varphi_{i,j+1,k}, \varphi_{i,j,k-1}, \varphi_{i,j,k+1}\}. \end{aligned} \tag{4.33}$$

We define $\mathbf{x}_{ijk}$ as an ***irregular grid point*** if

$$\varphi_{ijk}^{max} \cdot \varphi_{ijk}^{min} \leq 0. \tag{4.34}$$

We define $\mathbf{x}_{ijk} = (r_i, \phi_j, \theta_k)$ as an ***sub-irregular grid point*** if it is not an irregular grid point, but one of its six neighbors is an irregular grid point. If $\mathbf{x}_{ijk}$ is neither an irregular nor a sub-irregular grid point, then we call it a ***regular grid point.***

Step 2: Finding the orthogonal projections. If $\mathbf{x}_{ijk}$ is an irregular or sub-irregular grid point, we use the second order central finite difference scheme to find the approximate orthogonal direction $\mathbf{p}$ defined in (2.15) at $\mathbf{x}_{ijk}$, and the Hessian matrix $He(\varphi(\mathbf{x}_{ijk}))$ in spherical coordinates. Then we solve the quadratic equation (2.13) to get α and an approximate orthogonal projection $\mathbf{x}_{ijk}^*$ of $\mathbf{x}_{ijk}$ on the interface Γ.

Step 3: Extending the jumps. We need to extend the jumps to all irregular and sub-irregular grid points $\mathbf{x}_{ijk}$ from their orthogonal projection $\mathbf{x}_{ijk}^*$ on the interface according to the following

$$(w_e)_{ijk} = w(\mathbf{x}_{ijk}^*(s)), \qquad (v_e)_{ijk} = v(\mathbf{x}_{ijk}^*(s)). \tag{4.35}$$

Now we compute $\tilde{u}_{ijk}$ and $\hat{u}_{ijk}$ defined in (2.8) and (2.9) at irregular and sub-irregular grid points $\mathbf{x}_{ijk}$ according to the following

$$\begin{aligned} \tilde{u}_{ijk} &= w_e(\mathbf{x}_{ijk}) + v_e(\mathbf{x}_{ijk}) \frac{\varphi(\mathbf{x}_{ijk})}{|\nabla\varphi(\mathbf{x}_{ijk})|}, \\ \hat{u}_{ijk} &= = \begin{cases} 0, & \text{if } \varphi(\mathbf{x}_{ijk}) \leq 0, \\ \tilde{u}_{ijk}, & \text{if } \varphi(\mathbf{x}_{ijk}) > 0, \end{cases} \end{aligned} \tag{4.36}$$

Step 4: Adding the correction terms. Since the solution of the Poisson equation $\Delta(u(\mathbf{x}) - \hat{u}(\mathbf{x}) = f(\mathbf{x}) - H(\varphi(\mathbf{x}))\, \Delta\hat{u}(\mathbf{x})$ has a smooth solution, an approximate solution can be obtained by solving the following finite difference equations

$$\Delta_h U_{ijk} = \begin{cases} f_{ijk} - H_h(\varphi_{ijk})\Delta_h \tilde{u}_{ijk} + \Delta_h \hat{u}_{ijk} + C_{ijk}, & \text{if } \mathbf{x}_{ijk} \text{ is irregular,} \\ f_{ijk}, & \text{otherwise.} \end{cases} \tag{4.37}$$

Here, C_{ijk} is a correction term needed to offset the discontinuities in second order derivatives, and Δ_h is the standard discrete Laplacian in spherical coordinates described by

$$\begin{aligned} \Delta_h U_{ijk} = \; & \frac{U_{i+1,j,k} - 2U_{ijk} + U_{i-1,j,k}}{(\Delta r)^2} + \frac{2}{r_i}\frac{U_{i+1,j,k} - U_{i-1,j,k}}{2\Delta r} + \frac{1}{r_i^2}\frac{U_{i,j+1,k} - 2U_{ijk} + U_{i,j-1,k}}{(\Delta\phi)^2} \\ & + \frac{\cot\phi_j}{r_i^2}\frac{U_{i,j+1,k} - U_{i,j-1,k}}{2\Delta\phi} + \frac{1}{r_i^2 \sin^2\phi_j}\frac{U_{i,j,k+1} - 2U_{ijk} + U_{i,j,k-1}}{(\Delta\theta)^2}, \end{aligned}$$

and $H_h(\varphi_{ijk})$ is the discrete Heaviside function

$$H_h(\varphi_{ijk}) = \begin{cases} 1 & \text{if } \varphi_{ijk} > 0, \\ 0 & \text{if } \varphi_{ijk} \le 0. \end{cases} \tag{4.38}$$

The correction term C_{ijk} is defined in (4.40) and it is important for second order accuracy of our method. Note that when $i = 1$, the coefficient of U_{0jk} in the expression above happens to be zero. Therefore we have avoided the singularity at $r = 0$. The extension of the jumps to sub-irregular grid points is used to compute $\Delta_h \tilde{u}_{ijk}$ and $\Delta_h \hat{u}_{ijk}$ at irregular grid points.

Define

$$F_{ijk} = \begin{cases} f_{ijk} - H_h(\varphi_{ijk})\Delta_h \tilde{u}_{ijk}, & \text{if } \mathbf{x}_{ijk} \text{ is irregular,} \\ f_{ijk}, & \text{otherwise.} \end{cases} \tag{4.39}$$

The correction term then is

$$C_{ijk} = \sum_{i_l,j_l,k_l} H_h(-\varphi_{i+i_l,j+j_l,k+k_l}\,\varphi_{ijk})\gamma_{i+i_l,j+j_l,k+k_l}\left(\frac{\varphi_{i+i_l,j+j_l,k+k_l}}{|\nabla\varphi_{i+i_l,j+j_l,k+k_l}|}\right)^2 \frac{F_{i+i_l,j+j_l,k+k_l} - F_{ijk}}{2} \tag{4.40}$$

where $(i_l, j_l, k_l) = \{(-1,0,0),(1,0,0),(0,-1,0),(0,1,0),(0,0,-1),(0,0,1)\}$, $\gamma_{i+i_l,j+j_l,k+k_l}$ are the coefficients of the discrete Laplacian in spherical coordinates:

$$\gamma_{i\pm1,j,k} = \frac{1}{(\Delta r)^2} \pm \frac{1}{r_i \Delta r}, \quad \gamma_{i,j\pm1,k} = \frac{1}{(r_i\,\Delta\phi)^2} \pm \frac{\cot\phi_j}{2r_i^2\,\Delta\phi}, \quad \gamma_{i,j,k\pm1} = \frac{1}{(r_i \sin\phi_j \Delta\theta)^2}. \tag{4.41}$$

The gradient vector

$$\nabla\varphi = \left(\varphi_r,\; \frac{1}{r}\varphi_\phi,\; \frac{1}{r\sin\phi}\,\varphi_\theta\right) \tag{4.42}$$

is computed using the standard second order central finite difference scheme at involved grid points.

It is clear now that the discontinuity in the solution and the flux only affect the right hand side of the finite difference equations.

Step 5: Solving the discrete system. We solve the discrete system of equations with modified right hand side using the fast method described in Section 3 to get an approximate solution to the original problem either in the entire space or inside the truncated domain $r \le a$.

5 Performance study

We have done intensive tests on the methods discussed in this paper using Sun Ultra workstations or PCs. Our computer codes[3] have not been optimized and parallelized. However, all the numerical results confirm second order accuracy of the present methods in the infinity norm.

5.1 Example 1.

In this example, the level set function is

$$\varphi(r, \phi, \theta) = r^2 - (1 + Br \sin\phi \cos\theta) . \tag{5.43}$$

The interface in Cartesian grid, therefore, is $x^2 + y^2 + z^2 - Bx = 1$. The exact solution is chosen as

$$u(r, \phi, \theta) = \begin{cases} r^3 \sin\phi\cos\theta & \text{if } \varphi(r, \phi, \theta) \le 0, \\ \dfrac{\sin\phi\cos\theta}{r^2} & \text{if } \varphi(r, \phi, \theta) > 0. \end{cases} \tag{5.44}$$

The source term is

$$f(r, \phi, \theta) = \begin{cases} 10\, r \sin\phi\cos\theta & \text{if } \varphi(r, \phi, \theta) \le 0, \\ 0 & \text{if } \varphi(r, \phi, \theta) > 0. \end{cases} \tag{5.45}$$

The jump condition is determined from the exact solution and the interface. In the special case when $B = 0$, the interface is the unit sphere $r = 1$, the jump conditions are

$$[u] = 0, \qquad [u_n] = -5 \sin\phi \cos\theta. \tag{5.46}$$

If $B \neq 0$, then both the solution and the flux have a non-zero jump.

In Table 1, we show the result of the grid refimenet analysis for the solution both inside and outside of the auxiliary sphere $r = a = 2$. The error is defined as

$$E_\infty = \max_{ijk} \left\{ |u(r_i, \phi_j, \theta_k) - U_{ijk}| \right\} , \tag{5.47}$$

where $u(r_i, \phi_j, \theta_k)$ and U_{ijk} are the exact solution and computed solutions at the grid point (r_i, ϕ_j, θ_k). We use E^1_∞ to denote the maximum error inside the sphere, and E^2_∞ to denote the maximum error outside the sphere. The order of accuracy is defined as

$$\text{order} = \frac{\log(E(M)/E(2M))}{\log 2}. \tag{5.48}$$

In Table 1 (a), $B = 0$ and the interface is the unit sphere which is aligned with the grid line in r direction. The solution is continuous, but both the flux and the source term have a finite jump. The grid refinement analysis show clearly second order accuracy as the mesh gets finer. In Table 1 (b), the interface depends on r, ϕ, and θ. It is not aligned with any grid lines. Now the solution, the flux, and the source term all have a finite jump across the interface. Generally, for

Table 1: The grid refinement analysis for Example 1. The auxiliary sphere separating the infinite domain is $r = a = 2$. The CPU time unit is in seconds.

(a) Sphere interface: $B = 0$. The solution is continuous but the flux is not.

M	N	L	E^1_∞	order	E^2_∞	order	CPU(s)
16	32	16	7.0822×10^{-3}		4.4105×10^{-3}		0.0900
32	64	32	1.7932×10^{-3}	1.9816	1.1444×10^{-3}	1.9464	0.6700
64	128	64	4.5196×10^{-4}	1.9883	2.9192×10^{-4}	1.9710	5.5200
128	256	128	1.1349×10^{-4}	1.9936	7.3742×10^{-5}	1.9850	49.870

(b) An arbitrary interface: $B = 0.25$. Both the solution and the flux are discontinuous.

M	N	L	E^1_∞	order	E^2_∞	order	CPU(s)
16	32	16	5.0432×10^{-2}		9.0128×10^{-3}		0.0900
32	64	32	1.3027×10^{-2}	1.9529	2.4347×10^{-3}	1.8882	0.6600
64	128	64	1.8168×10^{-3}	2.8420	6.3863×10^{-4}	1.9307	5.5500
128	256	128	5.2148×10^{-4}	1.8008	1.7160×10^{-4}	1.8960	49.870

Table 2: The grid refinement analysis for Example 1 using the artifical boundary condition (3.31) and $B = 0.25$. The CPU time unit is in seconds.

M	N	L	$E^1_\infty(a = 2)$	order $(a = 2)$	$E^1_\infty(a = 5)$	order $(a = 5)$	CPU(s)
16	32	16	1.8356×10^{-2}		2.9658×10^{-2}		0.0700
32	64	32	3.4625×10^{-3}	2.4063	6.8574×10^{-3}	2.1127	0.4800
64	128	64	4.9865×10^{-4}	2.7957	2.3333×10^{-3}	1.5553	3.8200
128	256	128	1.2012×10^{-4}	2.0536	6.2092×10^{-4}	1.9099	32.070

interface problems, the maximum errors usually do not decrease monotonically (see [14, 4]). But the average convergence rate in Table 1 (b) shows second order convergence again.

In Table 2, we show the result of the grid refinement analysis using the artificial boundary condition (3.31). The error is measured at grid points inside the artificial sphere $r = a$. The CPU time was cut by less than half. In this table, we list the numerical results for $a = 2$ and $a = 5$. It is hard to determine the optimal value of a since the choice is problem-dependent. In this case, the artificial sphere $a = 2$ seems to be a good choice for this particular artificial boundary condition technique. When we increase the radius of the sphere to $a = 5$, the maximum error actually gets a little bigger. The part of reason is that the mesh resolution when $a = 5$ is worse than the case when $a = 2$. Nevertheless, both cases show second order accuracy.

5.2 Example 2.

In this example, we show the result for a problem where the solution has a stronger discontinuity. The level set function is

$$\varphi(r, \phi, \theta) = r - (1 + B \sin\phi \cos\theta) . \tag{5.49}$$

The exact solution is chosen as

$$u(r, \phi, \theta) = \begin{cases} r \sin\phi \cos\theta \left(\dfrac{5}{2} - \dfrac{3}{2} r^2 \right) & \text{if } \varphi(r, \phi, \theta) \le 0, \\ \dfrac{2 \sin\phi \cos\theta}{r^2} & \text{if } \varphi(r, \phi, \theta) > 0. \end{cases} \tag{5.50}$$

The source term is

$$f(r, \phi, \theta) = \begin{cases} -15\, r \sin\phi \cos\theta & \text{if } \varphi(r, \phi, \theta) \le 0, \\ 0 & \text{if } \varphi(r, \phi, \theta) > 0. \end{cases} \tag{5.51}$$

The jumps in $[u]$ and $[u_n]$ are determined from the exact solution and the interface. Note that even in the special case when $B = 0$, the solution is discontinuous. In the following tests, we choose $B = 0.1$.

In Table 3, we show the grid refinement analysis with an auxiliary sphere $r = a = 2$ for the solution both inside and outside of the sphere. Average second order accuracy is again verified.

In Table 4, we show the result of the ABC approach. The results agree with our analysis for Example 1 and Table 2.

We should point out that our method does require the interface to be smooth enough in Euclidean space because the method is based on the extension of the jumps along the orthogonal directions. Some interfaces that are arbitrarily differentiable in spherical coordinates may have a singularity in Euclidean space. For example, $\varphi(r, \theta, \phi) = r - (1 - B\cos\theta)$ has a singularity at the pole $\phi = 0$ and $\phi = \pi$ when it is expressed in Cartesian coordinates.

[3] The code is available to public upon request.

Table 3: The grid refinement analysis for Example 2. The auxiliary sphere separating the unbounded domain is $r = a = 2$ and $B = 0.1$. The CPU time unit is in seconds.

M	N	L	E^1_∞	order	E^2_∞	order	CPU(s)
16	32	16	1.5243×10^{-2}		1.4778×10^{-2}		0.080
32	64	32	8.2877×10^{-3}	0.8791	1.8384×10^{-3}	3.0069	0.6600
64	128	64	1.1670×10^{-3}	2.8282	4.9265×10^{-4}	1.8998	5.5500
128	256	128	3.6756×10^{-4}	1.6667	1.2586×10^{-4}	1.9687	50.090

Table 4: The grid refinement analysis for Example 2 using the artifical boundary condition (3.31). The CPU time unit is in seconds.

M	N	L	$E^1_\infty(a = 2)$	order $(a = 2)$	$E^1_\infty(a = 5)$	order $(a = 5)$	CPU(s)
16	32	16	3.8444×10^{-2}		3.4777×10^{-1}		0.0700
32	64	32	8.0087×10^{-3}	2.2631	3.2515×10^{-2}	3.4189	0.4800
64	128	64	1.1029×10^{-3}	2.8602	1.4151×10^{-2}	1.2003	3.7800
128	256	128	3.5020×10^{-4}	1.6551	3.4269×10^{-3}	2.0459	31.9500

Acknowledgment

The authors would like to thank Drs. K. Ito and S. Tsynkov of North Carolina State University for beneficial discussions. Mr. Yi Chen wrote the computer code to find the orthogonal projection of a point on the interface using spherical coordinates as his Master degree project. The first author is supported in part by the National Science Council of Taiwan under research grant NSC-91-2115-M-009-016. The second author is partially supported by USA-ARO grants 39676-MA and 43751-MA and USA- NFS grants DMS-0073403 and DMS-0201094.

References

[1] A. Aharoni. *Introduction to the theory of ferromagnetism.* Oxford University Press, 1996.

[2] A. Bayliss, M. Gunzburger and E. Turkel, Boundary conditions for the numerical solution of elliptic equations in exterior regions, *SIAM J. Appl. Math.* vol 42, No 2, 430–451, (1982).

[3] C. G. Cervera, Z. Gimbutas, and W. E. Accurate evaluation of the stray field in micromagnetics simulations. 2002.

[4] M. Dumett and J. Keener. A numerical method for solving anisotropic elliptic boundary value problems on an irregular domain in 2D. *J. Comput. Phys.*, in press, 2002.

[5] Weinan E. Selected problems in material science. *Mathematics Unlimited, 2001 and beyond*, editors: B. Engquist and others, 2002.

[6] L. Greengard and V. Rokhlin. A fast algorithm for particle summations. *J. Comput. Phys.*, 73:325–348, 1987.

[7] J. B. Keller and D. Givoli, Exact non-reflecting boundary conditions, *J. Comput. Phys.*, 82, 172–192, (1989).

[8] M.-C. Lai and W.-C. Wang, Fast direct solvers for Poisson equation on 2D polar and spherical geometries, *Numer. Methods Partial Differential Eq.*, vol 18, 56-68, (2002).

[9] M.-C. Lai, W.-W. Lin and W. Wang, A fast spectral/difference method without pole conditions for Poisson-type equations in cylindrical and spherical geometries, *IMA J. Numer. Anal.*, vol 22, 537-548, (2002).

[10] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 31:1019–1044, 1994.

[11] R. J. LeVeque and Z. Li. Immersed interface method for Stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.*, 18:709–735, 1997.

[12] Z. Li. *The Immersed Interface Method — A Numerical Approach for Partial Differential Equations with Interfaces.* PhD thesis, University of Washington, 1994.

[13] Z. Li. A note on immersed interface methods for three dimensional elliptic equations. *Computers Math. Appl.*, 31:9–17, 1996.

[14] Z. Li. A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, 35:230–254, 1998.

[15] Z. Li and K. Ito. Maximum principle preserving schemes for interface problems with discontinuous coefficients. *SIAM J. Sci. Comput.*, 23:1225–1242, 2001.

[16] Z. Li, W-C. Wang, I-L. Chern, and M-C. Lai. New formulations for interface problems in polar coordinates. *SIAM J. Sci. Comput.*, in press, 2002.

[17] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer , New York, 2002.

[18] J. A. Sethian. *Level Set Methods and Fast Marching methods.* Cambridge University Press, 2nd edition,1999.

[19] P. N. Swarztrauber, A direct method for the discrere solution of separable elliptic equations, *SIAM J. Numer. Anal.* vol 11, 1136–1149, (1974).

[20] S. Tsynkov. Numerical solution of problems on unbounded domains. a review. *Appl. Numer. Math.*, 27:465–532, 1998.