NEW FORMULATIONS FOR INTERFACE PROBLEMS IN POLAR COORDINATES*

ZHILIN LI†, WEI-CHENG WANG‡, I-LIANG CHERN§, AND MING-CHIH LAI¶

Abstract. In this paper, numerical methods are proposed for some interface problems in polar or Cartesian coordinates. The new methods are based on a formulation that transforms the interface problem with a nonsmooth or discontinuous solution into a problem with a smooth solution. The new formulation leads to a simple second order finite difference scheme for the partial differential equation and a new interpolation scheme for the normal derivative of the solution. In conjunction with the fast immersed interface method, a fast solver has been developed for the interface problems with a piecewise constant but a discontinuous coefficient using the new formulation in a polar coordinate system.

1. Introduction. The original motivation for this paper was our desire to solve certain elliptic interface problems defined in a disk using polar coordinates. To this purpose, we consider two types of interface problems.

1.1. Interface problem A. This problem involves the *Poisson equation* with discontinuities and singularities

$$(1.1) \qquad \Delta u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \mathcal{R} - \Gamma,$$

$$u(\mathbf{x}) = u_0(\mathbf{x}) \quad \text{on} \quad \partial\mathcal{R},$$

where Δ is the Laplacian operator, $\mathcal{R}$ is a circular domain in two space dimensions, $\partial\mathcal{R}$ is the boundary of $\mathcal{R}$, and $\Gamma \in C^2$ is a closed interface that does not pass the origin and does not intersect with itself within the domain $\mathcal{R}$. Across the interface Γ, the jump conditions in the solution and in the flux,

$$(1.2) \quad [u]_{|\mathbf{X}\in\Gamma} = w(s), \qquad [u_n]_{|\mathbf{X}\in\Gamma} = v(s), \quad w(s) \in C^2(\Gamma), \quad v(s) \in C^2(\Gamma),$$

are prescribed, where s is the arc-length parameter of the interface Γ, $u_n = \frac{\partial u}{\partial \mathbf{n}} = \nabla u \cdot \mathbf{n}$ is the normal derivative of u, and $\mathbf{n}$ is the unit normal direction of Γ pointing outward;
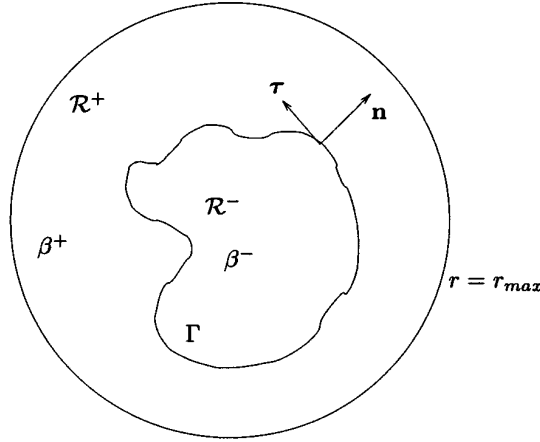
FIG. 1.1. *A diagram of a circular domain $\mathcal{R} = \mathcal{R}^+ \cup \mathcal{R}^-$ with an immersed interface $\Gamma(s)$. The coefficients $\beta(\mathbf{x})$ may have a jump across the interface.*

see Figure 1.1 for a geometric illustration. The jumps are defined as the difference of the limiting values from two different sides of the interface, for example,

$$[u]_{|\mathbf{X} \in \Gamma} = \lim_{\mathbf{x} \to \mathbf{X}, \mathbf{x} \in \mathcal{R}^+} u(\mathbf{x}) - \lim_{\mathbf{x} \to \mathbf{X}, \mathbf{x} \in \mathcal{R}^-} u(\mathbf{x}) \stackrel{\text{define}}{=} u^+(\mathbf{X}) - u^-(\mathbf{X}).$$

In this paper, since the jump is always defined at the interface, we can use either $[u]_{|\mathbf{X}}$ or $[u]$ with the subscript dropped for the jump if there is no confusion. Across the interface, the source term $f(\mathbf{x})$ can also have a discontinuity. We refer the readers to [9, 10, 11] for more explanations on the jump conditions. In this paper, we use bold-face, lowercase letters such as $\mathbf{x} = (x, y)$, $\mathbf{y}, \ldots$ to express points in the domain $\mathcal{R}$ and use boldface, uppercase letters such as $\mathbf{X}$, $\mathbf{Y}, \ldots$ to express points on the interface Γ, unless stated otherwise. We use a Dirichlet boundary condition on $\partial\mathcal{R}$ for simplicity. Other boundary conditions can be treated using standard techniques.

1.2. Interface problem B. This problem involves the *elliptic equation* with discontinuities and singularities

$$(1.3) \qquad \nabla \cdot (\beta(\mathbf{x})\nabla u(\mathbf{x})) = f(\mathbf{x}), \quad \mathbf{x} \in \mathcal{R} - \Gamma,$$

$$u(\mathbf{x}) = u_0(\mathbf{x}) \quad \text{on} \quad \partial\mathcal{R},$$

where $\cdot$ is the Euclidean inner product and $\beta(\mathbf{x})$ has a constant value in each sub-domain,

$$(1.4) \qquad \beta(\mathbf{x}) = \begin{cases} \beta^+ & \text{if } \mathbf{x} \in \mathcal{R}^+, \\ \beta^- & \text{if } \mathbf{x} \in \mathcal{R}^-. \end{cases}$$

The jump conditions

$$(1.5) \qquad [u]_{|\mathbf{X} \in \Gamma} = w(s), \qquad [\beta u_n]_{|\mathbf{X} \in \Gamma} = v(s)$$

are prescribed along the interface Γ.

Note that problem A is a special case of problem B with $\beta^- = \beta^+ = 1$. Problem B can be written as a single partial differential equation (PDE) to include the interface Γ in the domain in the sense of distributions. For example, if $w \equiv 0$, then the problem (1.3)–(1.5) is well known to be equivalent to

$$(1.6) \qquad \nabla \cdot (\beta(\mathbf{x})\nabla u(\mathbf{x})) = f(\mathbf{x}) + \int_\Gamma v(s)\, \delta_2(\mathbf{x} - \mathbf{X}(\mathbf{s}))\, ds,$$

$$u(\mathbf{x}) = u_0(\mathbf{x}) \quad \text{on} \quad \partial\mathcal{R},$$

where the equation is defined for all $\mathbf{x} \in \mathcal{R}$ including Γ, and $\delta_2(\mathbf{x})$ is the two-dimensional Dirac delta function (see [10, 11] for the equivalence). The nonhomogeneous jump in the solution also can be incorporated as a double layer source distribution along the interface Γ in the potential theory.

1.3. Some applications of the interface problems and a brief review of numerical methods. In the computation of micromagnetics for ferromagnetic materials, or electrostatics for macromolecules, the potential can be modeled as the solution of interface problem A or B; see [5]. For a potential problem defined on an infinite domain, usually a big circle or rectangle is used to divide the infinite domain into two parts. The problem then can be solved on either one or both domains. If a circle is used in the separation, then the discussions in this paper can be applied.

Equation (1.6), in which the interface conditions (1.5) with $w = 0$[1] is expressed as a distribution using the Dirac delta function, is the core of Peskin's *immersed boundary* (IB) method; see [18] for an overview. Numerically, the IB method uses a *discrete delta function* to distribute the integral in (1.6) to the nearby grid points. The IB method have been used for many applications, particularly in fluid mechanics and mathematical biology. However, the IB method is known to be only first order accurate. In this paper, we will propose second order accurate algorithms for interface problems A and B, including the IB model (1.6), in polar coordinates.

Interface problems A and B probably can be solved with some effort using other methods as well, for example, finite element methods with a body fitted grid; the ghost fluid method (GFM) [15] which is first order accurate, the second order finite difference methods based on integral equations [4, 16], the explicit jump immersed interface [21], and a few others. We refer the readers to [13] for a brief discussion of different methods. However, few discussions exist on how to apply these techniques to problems in polar coordinates.

The immersed interface method (IIM) [9, 11, 13] has a number of advantages in solving interface problems. However, the IIM method has not been directly applied to the interface problems in *polar coordinates* because (1.3) has a nonconstant coefficient in polar coordinates. The jump relations of the solution and its derivatives are not as useful and informative in polar coordinates as the ones derived in [9, 11] for Cartesian coordinates.

1.4. The contributions and the organization of the paper. In section 2, a new formulation that is based on the extension of the jump conditions (1.2) along the normal lines is proposed. The new formulation transforms interface problem A to a problem with a smooth solution. Theoretical analysis is given for the new formulation there.

---

[1]In some of the literature, Γ is called an internal boundary and the conditions in (1.5) are called internal boundary conditions.

In section 3, a numerical method that uses the new formulation is proposed. The method is second order accurate in the maximum norm. The cost of the new method is approximately one call to a fast Poisson solver.

We should point out that the idea of extending the solution smoothly across the interface is not new; see, for example, the description of the IIM method in terms of extensions [3] and the GFM [15]. The problem is how to do it correctly and accurately to obtain efficient and accurate algorithms. Our proposed extension is particularly simple when the interface Γ is implicitly defined by a level set function. The strategy of the extension developed in this paper is unique and distinct from other approaches. The numerical method developed from the jump extension is simple and accurate for both the solution and its normal derivative.

In section 4, we generalize the fast IIM developed in [12] to interface problem B in polar coordinates. The iterative method utilizes the new algorithm described in section 3 as a fast Poisson solver. The iterative method for a piecewise constant coefficient is second order accurate and the number of iterations is almost independent of the mesh size and the jump in the coefficient.

Numerical experiments and analysis are reported in section 5.

2. A new formulation and extensions for interface problem A. Let us repeat interface problem A below:

$$(2.1) \qquad \Delta u = f, \quad \mathbf{x} \in \mathcal{R} - \Gamma, \qquad [u]_\Gamma = w(s), \qquad [u_n]_\Gamma = v(s),$$
$$u(\mathbf{x}) = u_0(\mathbf{x}).$$

We assume that $w \in C^2(\Gamma)$ and $v \in C^2(\Gamma)$.

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

$$(2.2) \qquad \varphi(\mathbf{x}) \begin{cases} < 0 & \text{if } \mathbf{x} \in \mathcal{R}^-, \\ = 0 & \text{if } \mathbf{x} \in \Gamma, \\ > 0 & \text{if } \mathbf{x} \in \mathcal{R}^+. \end{cases}$$

We assume that $\varphi(\mathbf{x}) \in C^3(\mathcal{R})^2$ in the neighborhood of the interface Γ, which is the zero level set $\varphi(\mathbf{x}) = 0$. Usually, the level set function is chosen as the signed distance function from the interface; see [17, 20] and the references therein. 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 (in both directions) as

$$(2.3) \qquad w_e(\mathbf{x}) = w_e \left(\mathbf{X}(s) + \alpha \mathbf{n} \right) = w(\mathbf{X}(s))$$

and

$$(2.4) \qquad v_e(\mathbf{x}) = v_e \left(\mathbf{X}(s) + \alpha \mathbf{n} \right) = v(\mathbf{X}(s))$$

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

$$(2.5) \qquad \tilde{u}(\mathbf{x}) = w_e(\mathbf{x}) + v_e(\mathbf{x}) \frac{\varphi(\mathbf{x})}{|\nabla \varphi(\mathbf{x})|},$$

---

[2]In the algorithm's implementation, $\varphi(\mathbf{x})$ just needs to be in $C^2(\mathcal{R})$; see sections 3–5. Second order accurate results are obtained when $\varphi(\mathbf{x}) \in C^2(\mathcal{R})$.

where $|\cdot|$ is used to denote the Euclidean norm. We assume that $|\nabla\varphi(\mathbf{x})|$ is nonzero in a neighborhood of the interface. For example, $|\nabla\varphi(\mathbf{x})| = 1$ if $\varphi(\mathbf{x})$ is chosen as the signed distance function. 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 $C^3(\mathcal{R})$ in the neighborhood of the interface Γ. Define also

$$(2.6) \qquad \hat{u}(\mathbf{x}) = H(\varphi(\mathbf{x}))\tilde{u}(\mathbf{x}) = \begin{cases} 0 & \text{if } \varphi(\mathbf{x}) < 0, \\ \frac{1}{2}\tilde{u}(\mathbf{x}) & \text{if } \varphi(\mathbf{x}) = 0, \\ \tilde{u}(\mathbf{x}) & \text{if } \varphi(\mathbf{x}) > 0 \end{cases}$$

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* (2.1) *and $\hat{u}(\mathbf{x})$ be defined in* (2.6). *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:*

$$(2.7) \qquad \Delta q(\mathbf{x}) = f(\mathbf{x}) - H(\varphi(\mathbf{x}))\,\Delta\hat{u}(\mathbf{x}), \qquad \mathbf{x} \in \mathcal{R} - \Gamma,$$

$$(2.8) \qquad [q]_\Gamma = 0, \qquad [q_\tau]_\Gamma = 0, \qquad [q_n]_\Gamma = 0,$$

where τ is the unit tangent direction and $q_\tau = \nabla q \cdot \tau = \frac{\partial q}{\partial \tau}$. In other words, the new function $q(\mathbf{x})$ is a smooth function across the interface Γ.

Proof. If $\mathbf{x} \in \mathcal{R}^-$, then we have $\hat{u}(\mathbf{x}) = 0$, so $\Delta\hat{u}(\mathbf{x}) = 0$ and $H(\varphi(\mathbf{x})) = 0$. Therefore $\Delta q(\mathbf{x}) = \Delta u(\mathbf{x}) - 0 = f(\mathbf{x})$ and (2.7) is true. If $\mathbf{x} \in \mathcal{R}^+$, then $H(\varphi(\mathbf{x})) = 1$; we have

$$\Delta q(\mathbf{x}) = \Delta u(\mathbf{x}) - \Delta\hat{u}(\mathbf{x}) = f(\mathbf{x}) - H(\varphi(\mathbf{x}))\Delta\hat{u}(\mathbf{x}).$$

It remains to prove the jump conditions. Note that for any $\mathbf{X}(s) \in \Gamma$, we have

$$[q]_{\mathbf{X}(s)} = [u]_{\mathbf{X}(s)} - [\hat{u}]_{\mathbf{X}(s)} = w(s) - \tilde{u}^+(\mathbf{X}(s)) = w(s) - w_e(\mathbf{X}(s)) = 0.$$

Since $w(s)$ is differentiable along the interface, so is $w_e(\mathbf{X}(s))$. Differentiating the expression above with respect to s, we get

$$\frac{d}{ds}[q]_{\mathbf{X}(s)} = w'(s) - \frac{d}{ds}w_e(\mathbf{X}(s)) = [q_\tau]_{\mathbf{X}(s)} = 0.$$

To prove the last jump condition, we proceed with the following derivation:

$$\nabla\left(v_e(\mathbf{x})\,\frac{\varphi(\mathbf{x})}{|\nabla\varphi(\mathbf{x})|}\right)\cdot\mathbf{n} = v_e(\mathbf{x})\,\nabla\frac{\varphi(\mathbf{x})}{|\nabla\varphi(\mathbf{x})|}\cdot\mathbf{n}$$
$$= v_e(\mathbf{x})\left(\frac{\nabla\varphi(\mathbf{x})}{|\nabla\varphi(\mathbf{x})|} + \varphi(\mathbf{x})\,\nabla\frac{1}{|\nabla\varphi(\mathbf{x})|}\right)\cdot\mathbf{n}.$$

Here we have used the fact that $v_e(\mathbf{x})$ is a constant along the normal line. Using the facts that $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$, $\varphi = 0$ on Γ, and $\nabla w_e(\mathbf{x})\cdot\mathbf{n} = 0$, we conclude that

$$\hat{u}_n^+(\mathbf{X}(s)) = \nabla w_e(\mathbf{x})\cdot\mathbf{n} + v_e(\mathbf{X}(s))\,\frac{\nabla\varphi(\mathbf{X}(s))}{|\nabla\varphi(\mathbf{X}(s))|}\cdot\mathbf{n}$$
$$= v(\mathbf{X}(s))\,\frac{\nabla\varphi(\mathbf{X}(s))}{|\nabla\varphi(\mathbf{X}(s))|}\cdot\frac{\varphi(\mathbf{X}(s))}{|\nabla\varphi(\mathbf{X}(s))|} = v(\mathbf{X}(s))$$

for any point $\mathbf{X}(s)$ on the interface. Therefore we get

$$[q_n](\mathbf{X}(s)) = [u_n](\mathbf{X}(s)) - \hat{u}_n^+(\mathbf{X}(s)) = 0. \qquad \square$$

Remark. While $q(\mathbf{x})$ is smooth across the interface Γ, its second derivatives usually have finite jumps across the interface. This can also be observed from (2.7), where the right-hand side is discontinuous across the interface.

3. A numerical algorithm for interface problem A. In this section, we assume that $\beta = 1$ for simplicity. We describe our algorithm for solving (2.1). While the discussion is for polar coordinates, it can and has been applied to Cartesian grids with minor modifications.

We assume the domain is $0 \le r \le r_{max}$, $0 \le \theta < 2\pi$. We construct the following grid in polar coordinates:

$$r_i = \left(i - \frac{1}{2}\right)\Delta r, \quad i = 1, 2, \ldots, M \quad \text{with} \quad r_M = r_{max}, \quad \Delta r = \frac{r_{max}}{M - \frac{1}{2}};$$

(3.1)

$$\theta_j = j\Delta\theta, \quad j = 0, 1, \ldots, N - 1 \quad \text{with} \quad \theta_0 = 0, \quad \Delta\theta = \frac{2\pi}{N}.$$

The j index ends at $N - 1$ because $u(r, 0) = u(r, 2\pi)$. We choose $\Delta\theta \sim \Delta r$ and introduce

$$(3.2) \qquad\qquad h = \max\{\,\Delta r,\ \Delta\theta\,\}$$

for later use. We choose such a grid so that the fast Poisson solver developed in [8] can be used. The level set function is defined at grid points according to $\varphi_{ij} = \varphi(r_i, \theta_j)$. For a discontinuous quantity such as $u(\mathbf{x})$, $f(\mathbf{x})$, etc., if $\mathbf{x}$ happens to be a grid point, then we define its value at the interface as the limiting value from a particular side of the interface, say, the $\mathcal{R}^+$ side instead of the average of the limiting values from the two sides. This is because our algorithms actually approximate the PDE from a particular side of the interface.

Because $q(\mathbf{x})$ is smooth across the interface, we should be able to get reasonably good accuracy with the standard central five-point finite difference scheme. Clearly, the crucial issues are how to extend $w(s)$ and $v(s)$ along the normal line accurately and how far we should extend. These issues will be discussed in the following subsections.

We will use uppercase letters such as U_{ij}, the approximate solution to $u(r, \theta)$ at (r_i, θ_j), and U, the vector formed by all U_{ij}, etc., for discrete approximations and lowercase letters such as u, q, φ etc., for the exact solutions or exact quantities, unless stated otherwise.

3.1. The extension of the jumps along the normal line. Let

$$(3.3) \qquad\qquad \varphi_{i,j}^{max} = \max\{\varphi_{i-1,j}, \varphi_{i,j}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\},$$

$$\varphi_{i,j}^{min} = \min\{\varphi_{i-1,j}, \varphi_{i,j}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\}.$$

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

$$(3.4) \qquad\qquad \varphi_{i,j}^{max}\varphi_{i,j}^{min} \le 0.$$

We define $\mathbf{x}_{ij} = (r_i, \theta_j)$ as a *subirregular grid point* if it is not an irregular grid point but one of its four neighbors is an irregular grid point.

If $\mathbf{x}_{ij}$ is neither an irregular nor a subirregular grid point, then we call it a *regular grid point.*

Note that all the irregular and subirregular points are located within a two-grid size from the interface. In our numerical scheme, we only need to extend the jumps to all the irregular and subirregular grid points. Given any such grid point $\mathbf{x} = \mathbf{x}_{ij}$, the normal extension of the jumps is simply the jumps at the orthogonal projection of $\mathbf{x}_{ij}$ on the interface. So the crucial issue is to find the projections effectively and accurately in polar coordinates. Assume that $\mathbf{x} = (r, \theta)$ is a point near the interface. Let $\mathbf{X}^*$ be the orthogonal projection of $\mathbf{x}$ on the interface. We can write

$$(3.5) \qquad \mathbf{X}^* = \mathbf{x} + \alpha \mathbf{p}, \quad \text{where} \quad \mathbf{p} = \left[\begin{array}{c} \varphi_r(\mathbf{x}) \\ \dfrac{\varphi_\theta(\mathbf{x})}{r^2} \end{array} \right].$$

Since $\mathbf{x}$ is close to the interface, α is small. Using the Taylor expansion at $\mathbf{x}$, we get the following quadratic equation for the unknown scalar α:

$$(3.6) \qquad \varphi(\mathbf{x}) + (\nabla\varphi(\mathbf{x}) \cdot \mathbf{p})\, \alpha + \frac{1}{2} \left(\mathbf{p}^T He(\varphi(\mathbf{x}))\, \mathbf{p} \right) \alpha^2 = 0,$$

where

$$(3.7) \qquad \mathbf{p}^T He(\varphi)\, \mathbf{p} = \varphi_r^2\, \varphi_{rr} + \frac{2}{r^2}\, \varphi_r \varphi_\theta \varphi_{r\theta} + \frac{1}{r^4}\, \varphi_{r\theta}^2\, \varphi_{\theta\theta}.$$

In our algorithm, $\mathbf{x}$ is an irregular or subirregular grid point. The partial derivatives $\nabla\varphi(\mathbf{x})$, φ_{rr}, $\varphi_{r\theta}$, and $\varphi_{\theta\theta}$ are computed at the grid point using the standard centered five-point difference formula. For Cartesian coordinates, we refer the readers to [7, 14] for the details.

Remark. Usually (3.6) has two solutions. We pick one that minimizes $\|\mathbf{x} - \mathbf{X}^*\|_2$. If there are more than one point on the interface that have equal distance to $\mathbf{x}$, which can happen only if the curvature is very large ($\sim 1/h$) relative to the grid, we just take one of those points as the projection. The extension then is carried out with the chosen projection. This actually has little effect on the global accuracy because such points are few, if they exist; see [2] for the reasoning for a similar problem.

3.2. An outline of the algorithm. Our algorithm for solving (2.1) is outlined below:

- Set-up a grid (3.1).
- Label all the grid points as regular, irregular, and subirregular.
- Find the projections for irregular and subirregular grid points using (3.5) and (3.6).
- Extend the jumps to irregular and subirregular grid points from the projections to get $\tilde{u}_{ij}$ and $\hat{u}_{ij}$ defined in (2.5) and (2.6).
- Form the discrete equations

$$(3.8) \quad \Delta_h U_{ij} = \begin{cases} f_{ij} - H(\varphi_{ij})\Delta_h \tilde{u}_{ij} + \Delta_h \hat{u}_{ij} + C_{ij} & \text{if } \mathbf{x}_{ij} \text{ is irregular,} \\ f_{ij} & \text{otherwise,} \end{cases}$$

where Δ_h is the standard central finite difference operator. In polar coordi-

nates, for example, it is[3]

$$\Delta_h U_{ij} = \frac{U_{i-1,j} - 2U_{ij} + U_{i+1,j}}{(\Delta r)^2} + \frac{1}{r_i}\frac{U_{i+1,j} - U_{i-1,j}}{2\Delta r}$$

$$+ \frac{1}{r_i^2}\frac{U_{i,j-1} - 2U_{ij} + U_{i,j+1}}{(\Delta\theta)^2}.$$

The correction term C_{ij} is defined in (3.10) and it is important for second order accuracy of our method.

• Apply the fast Poisson solver for polar coordinates [8] to solve the discrete system of equations (3.8).

Now we can see why we just need to extend the jumps to the irregular and subirregular points. At a regular grid point where $\varphi_{ij} > 0$, we have

$$-H(\varphi_{ij})\Delta_h\tilde{u}_{ij} + \Delta_h\hat{u}_{ij} = \Delta_h\hat{u}_{ij} - \Delta_h\tilde{u}_{ij} = \Delta_h\tilde{u}_{ij} - \Delta_h\tilde{u}_{ij} = 0.$$

If $\varphi_{ij} < 0$, we have $\Delta_h\hat{u}_{ij} = 0$ and $\Delta_h\tilde{u}_{ij} = 0$ from the definition. Therefore the term $-H(\varphi_{ij})\Delta_h\tilde{u}_{ij} + \Delta_h\hat{u}_{ij}$ has no effect on the right-hand side at regular grid points.

Below we discuss how to determine C_{ij} in (3.8) at irregular grid points. Define

$$(3.9) \qquad\qquad F_{ij} = f_{ij} - H(\varphi_{ij})\,\Delta_h\tilde{u}_{ij}.$$

We expect $F_{ij} + C_{ij}$ to be a good approximation to $\Delta_h q_{ij} = \Delta_h(u_{ij} - \hat{u}_{ij})$. Without the correction term (i.e., $C_{ij} = 0$), the finite difference scheme (3.8) is first order accurate because the second order derivatives of $q(\mathbf{x})$ are discontinuous. The jumps in Δq is reflected in the jump of F_{ij}.

Let $\mathbf{x}_{ij}$ be an irregular grid point. The contribution to the correction term comes from those of the four neighboring grid points that are in the other side of the interface opposite the grid point $\mathbf{x}_{ij}$. The correction term can be written as

$$(3.10) \quad C_{ij} = \sum_{i_k, j_k} H(-\varphi_{i+i_k, j+j_k}\,\varphi_{ij})\,\gamma_{i+i_k, j+j_k}\,\left(\frac{\varphi_{i+i_k, j+j_k}}{|\nabla\varphi_{i+i_k, j+j_k}|}\right)^2\frac{F_{i+i_k, j+j_k} - F_{ij}}{2},$$

where $(i_k, j_k) = \{(-1,0), (1,0), (0,-1), (0,1)\}$, $H(\cdot)$ once again is the Heaviside function,[4] and $\gamma_{i+i_k, j+j_k}$ is the coefficient of the discrete Laplacian Δ_h corresponding to $U_{i+i_k, j+j_k}$. In polar coordinates, these coefficients are

$$(3.11) \qquad\qquad \gamma_{i\pm1,j} = \frac{1}{(\Delta r)^2} \pm \frac{1}{2r_i\Delta r}, \qquad \gamma_{i,j\pm1} = \frac{1}{(\Delta\theta)^2}.$$

To show why we have the correction term (3.10), we introduce the following lemmas.

LEMMA 3.1. _Let_ $q(\mathbf{x}) = u(\mathbf{x}) - \hat{u}(\mathbf{x})$, $\frac{\partial^2}{\partial n^2}$ _be the second order derivatives of_ $q(\mathbf{x})$ _along the outer normal direction, and_ $F(\mathbf{x}) = f(\mathbf{x}) - H(\varphi)\Delta\tilde{u}(\mathbf{x})$. _Then we have_

$$(3.12) \qquad\qquad \left[\frac{\partial^2 q}{\partial n^2}\right] = [\Delta q] = [f - H(\varphi)\Delta\tilde{u}] = [F]$$

for any point on the interface Γ.

---

[3]Note that $\tilde{u}$ and $\hat{u}$ are defined using the known jump conditions and are not unknowns, so we still use lowercase letters to represent them at the grid points.

[4]In the discrete case, we define $H(\varphi) = 1$ if $\varphi \geq 0$.

Proof. Since the Laplacian operator is invariant under orthogonal coordinates, we have

$$\Delta q = \frac{\partial^2 q}{\partial n^2} + \frac{\partial^2 q}{\partial \tau^2} = F.$$

From $[q] = 0$, $[q_n] = 0$, and $[q_\tau] = 0$ proved in Theorem 2.1, we can conclude that $[\frac{\partial^2 q}{\partial \tau^2}] = 0$, which is derived in [9, 11] for the interface relations in the local coordinates. Therefore we conclude

$$[\Delta q] = [F] = \left[\frac{\partial^2 q}{\partial n^2}\right]. \qquad \square$$

The following lemma is the basis for the correction terms.

LEMMA 3.2. *Let $\varphi(\mathbf{x}) \in C^3$ in the neighborhood of the zero level set $\varphi(\mathbf{x}) = 0$, $\mathbf{x}_{ij}$ be an irregular point with $\varphi_{ij} < 0$ and $\varphi_{i+1,j} > 0$, and $\tilde{u}(\mathbf{x})$ be well defined within two grid points from $\mathbf{x}_{ij}$. Then we have*

$$(3.13) \quad \Delta_h^r q(\mathbf{x}_{ij}) \stackrel{define}{=} \frac{q(\mathbf{x}_{i-1,j}) - 2q(\mathbf{x}_{ij}) + q(\mathbf{x}_{i+1,j})}{(\Delta r)^2} + \frac{1}{r_i} \frac{q(\mathbf{x}_{i+1,j}) - q(\mathbf{x}_{i-1,j})}{2\Delta r}$$

$$= \frac{\partial^2 q}{\partial r^2}(\mathbf{x}_{ij}) + \frac{1}{r_i} \frac{\partial q}{\partial r}(\mathbf{x}_{ij}) + \gamma_{i+1,j} \left(\frac{\varphi_{i+1,j}}{|\nabla \varphi_{i+1,j}|}\right)^2 \frac{F_{i+1,j} - F_{ij}}{2}.$$

Proof. Let $\mathbf{X}^*$ be the projection of $\mathbf{x}_{i+1,j}$ on the interface. Expanding $\varphi(\mathbf{x})$ in Taylor expansion at $\mathbf{X}^*$, we get

$$0 = \varphi(\mathbf{X}^*) = \varphi(\mathbf{x}_{i+1,j}) + \nabla\varphi(\mathbf{x}_{i+1,j}) \cdot (\mathbf{X}^* - \mathbf{x}_{i+1,j}) + O\left(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^2\right).$$

Since $\mathbf{X}^* - \mathbf{x}_{i+1,j}$ is parallel to the normal direction $\mathbf{n}$ according to the definition of the projection, and $\varphi_{i+1,j} > 0$, we get

$$(3.14) \quad |\mathbf{x}_{i+1,j} - \mathbf{X}^*| = \frac{\varphi(\mathbf{x}_{i+1,j})}{|\nabla\varphi(\mathbf{x}_{i+1,j})|} + O\left(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^2\right),$$

which is known to be an approximate distance between $\mathbf{X}^*$ and $\mathbf{x}_{i+1,j}$. Now we expand $q(\mathbf{x}_{i+1,j})$ in Taylor expansion at $\mathbf{X}^*$ to get

$$q(\mathbf{x}_{i+1,j}) = q^+(\mathbf{X}^*) + \nabla q^+(\mathbf{X}^*) \cdot (\mathbf{x}_{i+1,j} - \mathbf{X}^*)$$
$$+ \frac{1}{2}(\mathbf{x}_{i+1,j} - \mathbf{X}^*)^T He(q^+(\mathbf{X}^*))(\mathbf{x}_{i+1,j} - \mathbf{X}^*) + O\left(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3\right),$$

where $q^+(\mathbf{X}^*)$ is the limiting value of $q(\mathbf{x})$ from the outer side of $\mathcal{R}$ and so forth, and $He(q^+(\mathbf{X}^*))$ is the Hessian matrix of $q(\mathbf{x})$ at $\mathbf{X}^*$. Since

$$\frac{\mathbf{X}^* - \mathbf{x}_{i+1,j}}{|\mathbf{X}^* - \mathbf{x}_{i+1,j}|} = -\mathbf{n} + O(h^2),$$

from the continuity condition of $q(\mathbf{x})$ and its derivative at $\mathbf{X}^*$, we get

$$q(\mathbf{x}_{i+1,j}) = q^-(\mathbf{X}^*) - \frac{\partial q^-(\mathbf{X}^*)}{\partial n}|\mathbf{x}_{i+1,j} - \mathbf{X}^*| + \frac{1}{2}|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^2 \frac{\partial^2 q^-(\mathbf{X}^*)}{\partial n^2}$$
$$+ \frac{1}{2}|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^2 \left[\frac{\partial^2 q(\mathbf{X}^*)}{\partial n^2}\right] + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3).$$

Define the smooth extension of $q^-(\mathbf{x})$ from $\mathcal{R}^-$ into $\mathcal{R}^+$ in the neighborhood of $\mathbf{X}^*$ as

$$q^-(\mathbf{x}) = q^-(\mathbf{X}^*) - \frac{\partial q^-(\mathbf{X}^*)}{\partial n}|\mathbf{x} - \mathbf{X}^*| + \frac{1}{2}|\mathbf{x} - \mathbf{X}^*|^2 \frac{\partial^2 q^-(\mathbf{X}^*)}{\partial n^2}.$$

We have

$$q(\mathbf{x}_{i+1,j}) = q^-(x_{i+1,j}) + \frac{1}{2}|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^2 \left[\frac{\partial^2 q(\mathbf{X}^*)}{\partial n^2}\right] + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3).$$

Therefore

$$\Delta_h^r q(\mathbf{x}_{ij}) = \frac{q(\mathbf{x}_{i-1,j}) - 2q(\mathbf{x}_{ij}) + q(\mathbf{x}_{i+1,j})}{\Delta r^2} + \frac{1}{r_i}\frac{q(\mathbf{x}_{i+1,j}) - q(\mathbf{x}_{i-1,j})}{2\Delta r}$$

$$= \frac{q(\mathbf{x}_{i-1,j}) - 2q(\mathbf{x}_{ij}) + q^-(\mathbf{x}_{i+1,j})}{\Delta r^2} + \frac{1}{r_i}\frac{q^-(\mathbf{x}_{i+1,j}) - q(\mathbf{x}_{i-1,j})}{2\Delta r}$$

$$+ \gamma_{i+1,j}\left(\frac{1}{2}|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^2 \left[\frac{\partial^2 q(\mathbf{x}^*)}{\partial n^2}\right] + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3)\right).$$

Since $q^-(\mathbf{x}_{i+1,j})$ is the extension of $q^-(\mathbf{x})$ from $\mathcal{R}^-$ into $\mathcal{R}^+$, the first two terms in the expression above are second order approximations to $\frac{\partial^2 q}{\partial r^2}(\mathbf{x}_{ij})$ and $\frac{1}{r_i}\frac{\partial q}{\partial r}(\mathbf{x}_{ij})$, respectively. Therefore the expression above can also be written as

$$(3.15) \quad \Delta_h^r q(\mathbf{x}_{ij}) = \frac{\partial^2 q}{\partial r^2}(\mathbf{x}_{ij}) + \frac{1}{r_i}\frac{\partial q}{\partial r}(\mathbf{x}_{ij})$$

$$+ \gamma_{i+1,j}\left(\frac{1}{2}|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^2 \left[\frac{\partial^2 q(\mathbf{X}^*)}{\partial n^2}\right] + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3)\right).$$

Finally, from (3.14) and Lemma 3.1, we have

$$\frac{1}{2}|\mathbf{x} - \mathbf{X}^*|^2 \left[\frac{\partial^2 q(\mathbf{X}^*)}{\partial n^2}\right] = \frac{1}{2}\left(\frac{\varphi_{i+1,j}}{|\nabla\varphi_{i+1,j}|}\right)^2 \left[\frac{\partial^2 q(\mathbf{X}^*)}{\partial n^2}\right] + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3)$$

$$= \frac{1}{2}\left(\frac{\varphi_{i+1,j}}{|\nabla\varphi_{i+1,j}|}\right)^2 [F(\mathbf{X}^*)] + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3)$$

$$= \frac{1}{2}\left(\frac{\varphi_{i+1,j}}{|\nabla\varphi_{i+1,j}|}\right)^2 (F_{i+1,j} - F_{ij}) + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3),$$

where we have used (3.12), the fact that $\varphi_{i+1,j} = O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|)$, and $[F(\mathbf{X}^*)] = F_{i+1,j} - F_{ij} + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|)$. Plugging the expression above into (3.15), we get the result of the lemma. □

The last term in (3.13) in Lemma 3.2 is the contribution to the correction term C_{ij} from $\mathbf{x}_{i+1,j}$ if it is in the side of the interface opposite the grid point $\mathbf{x}_{ij}$. Similarly, we can show the contributions to the correction term from the other neighboring points if they are in the side of the interface opposite $\mathbf{x}_{ij}$.

With the correction term C_{ij}, the finite difference scheme matches the differential equation up to second order derivatives after the Taylor expansion at $\mathbf{X}^*$. As a result, the computed solution of (3.8) has global second order accuracy in the infinite norm;

see [13] for the analysis and section 5 for the numerical results. If the extension is exact, the method described in this section gives the exact solution if it is a piecewise quadratic function.

Since the left-hand side of the system of the finite difference equations (3.8) is the standard discrete Laplacian in polar coordinates, the FFT-based fast Poisson solver [8] or the one from Fishpack [1] can be applied directly. The main cost is the Poisson solver, which typically requires $MN \log(MN)$ operations. The cost in dealing with the interface is $O(N_1)$, where N_1 is the total number of irregular and subirregular grid points. Usually we have $N_1 \sim \max\{M, N\}$, which is one dimension lower than the total number of grid points.

3.3. Computing the normal derivative. Quite often, we need to compute not only the solution of the Poisson equation (2.1) but also the normal derivative of the solution at the interface. With the new formulation, it can be done easily since $q(\mathbf{x})$ is smooth across the interface.

Because the level set function is defined everywhere, or at least in a tube $|\varphi(\mathbf{x})| \leq \delta$ that contains the interface, where δ is a given width, we can naturally define the "normal derivative" at all grid points in the tube as

$$(3.16) \qquad \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) = \nabla u(\mathbf{x}) \cdot \frac{\nabla \varphi(\mathbf{x})}{|\nabla \varphi(\mathbf{x})|}.$$

Therefore we can get a second order accurate "normal derivative" at regular grid points in the tube using the central differencing. In this section, we discuss how to compute the normal derivative of the solution (2.1) at irregular grid points. This is one of the crucial steps in the application of the level set method in which the velocity is calculated at the grid points. The scheme discussed here will also be used in subsection 4.4 to compute the normal derivative of the solution at the projections of irregular grid points on the interface.

Since we assume that the level set function is in C^3 in theory (the C^2 condition is enough for the implementation) in the neighborhood of the interface, we can compute $\nabla \varphi / |\nabla \varphi|$ to second order accuracy using the centered differencing at all grid points in the tube. It remains to evaluate (u_r, u_θ) to second order accuracy. Again, we take advantage of the smooth property of $q(\mathbf{x})$ with the following formula:

$$(3.17) \quad \frac{\partial U_{ij}}{\partial r} \approx \begin{cases} \dfrac{U_{i+1,j} - U_{i-1,j}}{2\Delta r} - \dfrac{\hat{u}_{i+1,j} - \hat{u}_{i-1,j}}{2\Delta r} \\ \qquad\qquad + H(\varphi_{ij})\dfrac{\tilde{u}_{i+1,j} - \tilde{u}_{i-1,j}}{2\Delta r} + C_{ij}^r & \text{if } \mathbf{x}_{ij} \text{ is irregular,} \\ \dfrac{U_{i+1,j} - U_{i-1,j}}{2\Delta r} & \text{otherwise,} \end{cases}$$

$$(3.18) \quad \frac{\partial U_{ij}}{\partial \theta} \approx \begin{cases} \dfrac{U_{i,j+1} - U_{i,j-1}}{2\Delta \theta} - \dfrac{\hat{u}_{i,j+1} - \hat{u}_{i,j-1}}{2\Delta \theta} \\ \qquad\qquad + H(\varphi_{ij})\dfrac{\tilde{u}_{i,j+1} - \tilde{u}_{i,j-1}}{2\Delta \theta} + C_{ij}^\theta & \text{if } \mathbf{x}_{ij} \text{ is irregular,} \\ \dfrac{U_{i,j+1} - U_{i,j-1}}{2\Delta \theta} & \text{otherwise,} \end{cases}$$

where the correction term is

$$(3.19)\quad C_{ij}^r = \frac{1}{2\Delta r} H(-\varphi_{i+1,j}\,\varphi_{ij}) \left(\frac{\varphi_{i+1,j}}{|\nabla\varphi_{i+1,j}|}\right)^2 \frac{F_{i+1,j}-F_{ij}}{2}$$

$$-\frac{1}{2\Delta r} H(-\varphi_{i-1,j}\,\varphi_{ij}) \left(\frac{\varphi_{i-1,j}}{|\nabla\varphi_{i-1,j}|}\right)^2 \frac{F_{i-1,j}-F_{ij}}{2},$$

$$C_{ij}^\theta = \frac{1}{2\Delta\theta} H(-\varphi_{i,j+1}\,\varphi_{ij}) \left(\frac{\varphi_{i,j+1}}{|\nabla\varphi_{i,j+1}|}\right)^2 \frac{F_{i,j+1}-F_{ij}}{2}$$

$$-\frac{1}{2\Delta\theta} H(-\varphi_{i,j-1}\,\varphi_{ij}) \left(\frac{\varphi_{i,j-1}}{|\nabla\varphi_{i,j-1}|}\right)^2 \frac{F_{i,j-1}-F_{ij}}{2}.$$

The correction terms are needed to offset the discontinuity in the second order derivatives of $q(\mathbf{x})$. The reasoning is the same as discussed in subsection 3.2.

4. The GMRES iteration for solving interface problem B. In this section, we consider the algorithm for solving the elliptic interface problem (1.3)–(1.5) with β given by (1.4) ($\beta^+ \neq \beta^-$). Divided by the coefficient in each subdomain of $\mathcal{R}$, the original problem can be written as

$$\Delta u = \frac{f}{\beta^+} \quad \text{if } \mathbf{x} \in \mathcal{R}^+,$$

(4.1a)

$$\Delta u = \frac{f}{\beta^-} \quad \text{if } \mathbf{x} \in \mathcal{R}^-,$$

(4.1b)
$$u(x,y) = u_0(x,y) \ \text{ on } \ \partial\mathcal{R},$$

excluding the interface Γ. This is a Poisson equation that can be solved easily using the algorithm discussed in the previous section if we know the jump in the solution $[u] = w(s)$ and the jump in the normal derivative $[u_n]$. However, the usual jump condition is in the flux $[\beta u_n] = v$ instead of $[u_n]$. We cannot divide β from the flux jump condition because β is discontinuous. In [12], we proposed a fast iterative method for Cartesian grids. In this section, we describe a similar iterative method in polar coordinates for (4.1a) with jump condition (1.5) using the new fast Poisson solver and the new interpolation scheme described in section 3.

As described in [12], the idea is to augment the unknown $[u_n] = g(s)$ to the original problem to have the system

(4.2)
$$\Delta u = \frac{f}{\beta} \quad \text{if } \mathbf{x} \in \mathcal{R}^+ \cup \mathcal{R}^- - \Gamma,$$

$$[u] = w(s), \quad [\beta u_n] = v(s), \quad [u_n] = g(s).$$

Note that $g(s)$ is also an unknown. The system is still closed because of an additional equation $[u_n] = g(s)$.

In the discretization, we represent the unknown jump $g(s) = [u_n]$ only at the projections $\mathbf{X}_k^*$ ($k = 1, 2, \ldots, N_\Gamma$) of the irregular grid points from the $\varphi \geq 0$ side, where N_Γ is the number of such projections. We call these points the *control points*. The reason for choosing the projections on one particular side is to avoid clustered control points. The intermediate unknown jump $g = [u_n]$ is defined at those control points as $G = [G_1, G_2, \ldots, G_{N_\Gamma}]^T$ in the discretization. The dimension of G is much smaller than that of the solution of U defined at all the grid points. Therefore we

use the GMRES method [19] to find the discrete intermediate unknown G. Once we know $g(s)$, we can solve (4.1a) with $[u] = w(s)$ and $[u_n] = g(s)$ using the new method described in section 3.

4.1. Setting up the system of equations for G and computing the residual. There are two coupled equations for the unknown $u(\mathbf{x})$ and $g(s)$. The first is the PDE $\Delta u = f(\mathbf{x})/\beta$ excluding the interface Γ with $[u] = w(s)$ and $[u_n] = g(s)$ given. The solution depends on $g(s)$ and can be written as $u_g(\mathbf{x})$. The second equation is the flux jump condition $[\beta \frac{\partial u_g}{\partial n}] = v(s)$. There are two discretization steps as follows:

- The system of the finite difference equations, which is obtained from the algorithm discussed in section 3 with given jumps $[u] = w$ and $[u_n] = g$, can be written as (in the matrix-vector form)

$$(4.3) \qquad AU + BG = F + F_w = F_1.$$

In (4.3), U is the vector formed from the approximation to $u(\mathbf{x})$ at all grid points, G is a discrete form of $g(s)$ at the control points on the interface, A is the matrix obtained from the standard discrete Laplacian in polar coordinates, F is the vector formed from the source term, F_w is the part of $-H(\varphi_{ij})\Delta_h \tilde{u}_{ij} + \Delta_h \hat{u}_{ij} + C_{ij}$ in (3.8) corresponding to the jump $[u] = w$, and $-BG$ is the part of the term corresponding to the jump $[u_n] = g$. We will not discuss the structures and properties of those matrices above and below because they are never explicitly formed in our implementation. However, they are useful in the theoretical discussions in this section.

- The discretization of the flux jump condition $[\beta u_n] = v$ in terms of u, $[u] = w$, and $[u_n] = g$ using an interpolation scheme, can be written as

$$(4.4) \qquad EU + DG = F_2,$$

where E, D, are two matrices. This is discussed in section 4.4.

If we put the two systems (4.3) and (4.4) together, we get

$$(4.5) \qquad \begin{bmatrix} A & B \\ E & D \end{bmatrix} \begin{bmatrix} U \\ G \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}.$$

Since the dimension of G, which is defined at the control points on the interface, is much smaller than the dimension of U, which is defined at all grid points, it is advantageous to focus on the Schur complement

$$(4.6) \qquad (D - EA^{-1}B)G = F_2 - EA^{-1}F_1,$$

$$\text{or} \quad SG = b, \quad \text{where} \quad S = D - EA^{-1}B, \quad b = F_2 - EA^{-1}F_1,$$

for the unknown G. The Schur complement system can be solved using the GMRES method [19]. Each iteration involves a call to the Poisson solver described in section 3 to get $U = A^{-1}(F_1 - BG)$, and an interpolation scheme of (4.4) for the flux jump condition $[\beta u_n] = v$ to get the residual vector

$$(4.7) \qquad R(G) = (D - EA^{-1}B)G - (F_2 - EA^{-1}F_1) = DG + EU - F_2$$

of the Schur complement system.

Note that if we take $G = 0$ in (4.7), then $-R(0) = b$ is the right-hand side of the Schur complement system. Also the residual vector (4.7) is the same as $R(G) =$

$\beta^+ U_n^+(G) - \beta^- U_n^-(G) - V$, where $U_n^\pm(G)$ is the vector whose components are the approximation of the normal derivative $\frac{\partial u^\pm}{\partial n}(\mathbf{X}_k^*)$, computed with the given normal jump G, and V is the vector whose components are $v(\mathbf{X}_k^*)$, $k = 1, 2, \ldots, N_\Gamma$, at the control points.

4.2. Outline of the algorithm for solving interface problem B.
- Set up the grid (3.1).
- Label the grid points as regular, irregular, and subirregular.
- Find the projections of irregular and subirregular grid points.
- Set $[u] = w$, $[u_n] = 0$, and solve the Poisson equation (4.1a)–(4.1b). Evaluate the residual (4.7), which gives the right-hand side b for (4.6).
- Set $G_k^0 = v(\mathbf{X}_k^*)$, $k = 1, 2, \ldots, N_\Gamma$, at the control points as an initial guess of G.
- Use the GMRES method to solve system (4.6) with some preconditioning techniques.

Several important implementation details are addressed in the following subsections.

4.3. Interpolating G at projections that are not control points.
In the GMRES iteration, the unknown G is the intermediate jump in the normal derivative defined only at the control points, that is, the projections of the irregular grid points where $\varphi \geq 0$. However, when we solve the Poisson problem (4.1a)–(4.1b) using the algorithm described in section 3, we need to know the jumps at all the projections before extending them to all irregular and subirregular grid points. This is done through the weighted least square interpolation [12]. At a projection $\mathbf{X}_k^*$, which is not a control point, we use

$$(4.8) \qquad G_k = \sum_p \gamma_p \, G_p$$

to get an approximation of G_k from those G_p at the control points. The summation includes $3 \sim 6$ projections from the $\varphi \geq 0$ side within a circle as follows:

$$\|\mathbf{X}_k^* - \mathbf{X}_p^*\|_2 \leq \epsilon.$$

In our experiment, we choose ϵ to be $2h \sim 3h$. The circle should enclose at least three control points to ensure second order accuracy. If there are more than six control points in the circle, then we choose the six that are closer to $\mathbf{X}_k^*$ than the others in the circle. The coefficients γ_p are obtained from the undetermined coefficient method by expanding $g(\mathbf{X}_p^*)$ along the interface up to second order derivatives. The least square solution of the underdetermined system of equations for γ_p is solved with the singular value decomposition (SVD); see section 3.2 in [14] for the detailed description of the interpolation scheme. Note that we have omitted the dependency of p on the index k to simplify notation.

4.4. Computing the residual of the Schur complement.
In the GMRES iteration, given a guess G, the matrix-vector multiplication involves two steps. The first step is to solve the Laplace equation (4.1a)–(4.1b) to get U, the vector whose components are the solution at all grid points. The second step is to compute the residual vector $R(G)$ in (4.7), which is the same as $\beta^+ U_n^+(G) - \beta^- U_n^-(G) - V$, where again $U_n^\pm(G)$ is the vector whose components are the approximation of the normal derivative $\frac{\partial u^\pm}{\partial n}(\mathbf{X}_k^*)$, computed with the given normal jump G. Therefore we need to compute each component of $U_n^+(G)$ and $U_n^-(G)$ at those control points $\mathbf{X}_k^*$.

Once again, it is easier to evaluate $Q_n^\pm(G)$ because $q(\mathbf{x})$ is smooth, where $Q_n^\pm(G)$ is the vector whose components are the approximation of the normal derivative $\frac{\partial q^\pm}{\partial n}(\mathbf{X}_k^*)$, $k = 1, 2, \ldots, N_\Gamma$, at the control points. Once we have $Q_n^\pm(G)$, we can get $U_n^\pm(G)$ using the relation $u(\mathbf{x}) = q(\mathbf{x}) + \hat{u}(\mathbf{x})$. We have already described how to evaluate $(U_n^\pm)_{ij}$ at grid points in subsection 3.3, which gives $(Q_n^\pm)_{ij} = (U_n^\pm)_{ij} - \hat{u}_{ij}$. Therefore we can use an interpolation scheme to get $(Q_n^\pm)_k$, the kth component of vector $(Q_n^\pm)$, from the values at nearby grid points. Since $q_n^+(\mathbf{X}_k^*) = q_n^-(\mathbf{X}_k^*)$, we can either use

$$(4.9) \qquad (Q_n^+)_k = \sum_{ij} \gamma_{ij}^k \, (Q_n^+)_{ij}$$

and set $(Q_n^-)_k = (Q_n^+)_k$, or use

$$(4.10) \qquad (Q_n^-)_k = \sum_{ij} \gamma_{ij}^k \, (Q_n^-)_{ij}$$

and set $(Q_n^+)_k = (Q_n^-)_k$. Due to the jumps in the second order derivatives of $q(\mathbf{x})$, the results obtained from (4.9) and (4.10) are slightly different. We will explain in the next subsection the choice of the scheme. The summation is done through the grid points in a neighborhood of $\mathbf{X}_k^*$. We use the *nine-point stencil* centered at the irregular grid point whose projection is $\mathbf{X}_k^*$ for the interpolation. If $\varphi_{ij} < 0$ and $\mathbf{x}_{ij}$ is a regular grid point, then $(Q_n^+)_{ij} = (Q_n^-)_{ij}$ and it is evaluated using the central finite difference scheme. If $\varphi_{ij} < 0$ and $\mathbf{x}_{ij}$ is an irregular grid point, then $(Q_n^+)_{ij}$ is the extension of Q_n^+ from the outside of the interface as follows:

$$(4.11) \qquad (Q_n^+)_{ij} = (Q_n^-)_{ij} + [F]_{ij} \frac{\varphi_{ij}}{|\nabla \varphi_{ij}|}.$$

Similarly, the extension from the inside to the outside grid point ($\varphi_{ij} > 0$) is

$$(4.12) \qquad (Q_n^-)_{ij} = (Q_n^+)_{ij} + [F]_{ij} \frac{\varphi_{ij}}{|\nabla \varphi_{ij}|}.$$

The discrete average jump $[F]_{ij}$ at an irregular grid point is defined as

$$(4.13) \qquad [F]_{ij} = \frac{\displaystyle\sum_{i+i_k, j+j_k} H(-\varphi_{i+i_k, j+j_k} \, \varphi_{ij}) \, (F_{i+i_k, j+j_k} - F_{ij})}{\displaystyle\sum_{i_k, j_k} H(-\varphi_{i+i_k, j+j_k} \, \varphi_{ij})},$$

where $(i_k, j_k) = \{\, (-1, 0), (1, 0), (0, -1), (0, 1) \,\}$, as we used earlier in (3.10). Note that at an irregular grid point, the denominator cannot be zero. Once we have computed $Q_n^\pm(\mathbf{X}_k^*)$, the normal derivatives of the solution at $\mathbf{X}_k^*$ are then determined from

$$(4.14) \qquad (U_n^-)_k = (Q_n^-)_k, \qquad (U_n^+)_k = (Q_n^+)_k + \hat{u}_n(\mathbf{X}_k^*).$$

To determine the coefficients γ_{ij}^k, the undetermined coefficients method is once again used. By expanding $q(\mathbf{x}_{ij})$ at $\mathbf{X}_k^*$ up to second order derivatives for all nine points that are involved, we get the linear system of six equations for the nine unknowns γ_{ij}^k. The underdetermined linear system is solved by the SVD approach; see section 4 in [12] for this least squares interpolation. One of the advantages of such an interpolation is that the magnitude of the coefficients cannot be very large regardless of the position of $\mathbf{X}_k^*$. The CPU time spent in dealing with the interfaces usually is less than 5% of the total CPU time. The main cost is the time used for the Poisson solver in polar coordinates.

4.5. The preconditioning strategy. The iterative method described above works for interface problem B. But the number of iterations seems to grow linearly with the mesh size; see Table 5.3. We believe this is because the flux jump condition involves the normal derivative. Some preconditioning techniques are crucial to reducing the number of iterations. Note that we do not form the iteration matrix of the Schur complement system and that many conventional preconditioning strategies based on the structure of the coefficient matrix cannot be applied. Our preconditioning strategy is to enforce the flux jump condition during the iteration, which seems to work well. The preconditioning techniques that we have implemented are the following:

$$
\text{If} \quad \beta^+ \le \beta^-, \quad \text{use} \quad
\begin{cases}
(Q_n^+)_k = \sum_{ij} \gamma_{ij}^k \, (Q_n^-)_{ij}, \quad (U_n^+)_k = (Q_n^+)_k + \hat{u}_n(\mathbf{X}_k^*), \\[2mm]
(U_n^-)_k = (Q_n^-)_k = \dfrac{v(\mathbf{X}_k^*) - \beta^+ G_k}{\beta^+ - \beta^-}.
\end{cases}
$$

$$
\text{If} \quad \beta^+ > \beta^-, \quad \text{use} \quad
\begin{cases}
(U_n^-)_k = (Q_n^-)_k = \sum_{ij} \gamma_{ij}^k \, (Q_n^+)_{ij}, \\[2mm]
(U_n^+)_k = \dfrac{v(\mathbf{X}_k^*) - \beta^- G_k}{\beta^+ - \beta^-}.
\end{cases}
$$

The second part of the preconditioning strategy is the same as in [12] to enforce the flux jump condition, but the first part, which is discussed in the previous subsection, is new in this paper. The reasoning is simple; for example, if $\beta^- > \beta^+$, then it is likely that u^- is flatter than u^+ from the jump relation in the flux. Therefore u^- and q_n^- are likely to be more accurate than u^+ and q_n^+.

The computational cost of the algorithm described in this section is $O\left(k(MN \log(MN) + N_1)\right)$, where k is the number of iterations of the GMRES method and N_1 is the total number of irregular and subirregular grid points. Generally, the extra time spent in dealing with the interface is less than 5% of the total CPU time. The main cost is the Poisson solver at each iteration.

5. Numerical examples. We have done intensive tests on the methods discussed in this paper. Most of experiments were done on Sun Ultra workstations. Our computer codes[5] have not been optimized and parallelized. However, all the numerical results confirm second order accuracy for the solution and the normal derivative from both sides of the interface in the infinity norm. The extra effort spent on the interface is only a small portion of the total machine time if we compute the interpolation coefficients outside of the GMRES iteration.

In this section, we present our results for the interface

$$(5.1) \qquad \varphi(r,\theta) = r - (r_0 + \lambda \sin(\omega\theta)), \quad 0 \le r \le r_{max}, \quad 0 \le \theta < 2\pi,$$

with different parameters. Note that φ depends on both r and θ.

5.1. An example in which β is a constant. Since any constant β can be absorbed in the source term, we simply take $\beta = 1$. We present our results with the interface

$$(5.2) \qquad \varphi(r,\theta) = r - (0.5 + 0.1 \sin(4\theta + \pi)), \quad 0 \le r \le 1, \quad 0 \le \theta < 2\pi.$$

---

[5]The codes are available from the authors upon request.

To check the order of accuracy, we use two nonlinear functions,

$$
(5.3) \qquad u(r,\theta) = \begin{cases} r^4 & \text{if } \varphi(r,\theta) < 0, \\ r^2 \sin\theta & \text{if } \varphi(r,\theta) \geq 0 \end{cases}
$$

as the exact solution. Note that the solution depends on both r and θ. The source term excluding the interface Γ is

$$
(5.4) \qquad f(r,\theta) = \begin{cases} 16\,r^2 & \text{if } \varphi(r,\theta) < 0, \\ 3\,\sin\theta & \text{if } \varphi(r,\theta) > 0. \end{cases}
$$

The Dirichlet boundary condition at $r_{max} = 1$, the jump in the solution, and the normal derivative are computed from the exact solution at the projections, where $\varphi \geq 0$.

Table 5.1 lists the grid refinement analysis of the computed results. The error of the computed projections is defined as

$$
(5.5) \qquad E_p = \max_k \left| r_e(\theta_k^*) - r_k^* \right|,
$$

where the (r_k^*, θ_k^*)'s are computed projections of all irregular grid points and $r_e(\theta^*)$ is the exact interface relation given by $r_e(\theta) = 0.5 + 0.1\sin(4\theta + \pi)$. The error in the solution E_u is defined as

$$
(5.6) \qquad E_u = \max_{\substack{0 \leq i \leq M \\ 0 \leq j \leq N}} \left| u(r_i, \theta_j) - U_{ij} \right|,
$$

where U_{ij} is the computed solution at the grid points.

The errors of the normal derivatives are measured at two levels. $E_{u_n,g}$ is the error measured at all irregular grid points

$$
(5.7) \qquad E_{u_n,g} = \max_{(r_i,\theta_j) \text{ is irregular}} \left| u_n(r_i, \theta_j) - (U_n)_{ij} \right|,
$$

while $E_{u_n,\Gamma}$ is the error measured at all the projections of irregular grid points

$$
(5.8) \qquad E_{u_n,\Gamma} = \max_{(r_k^*,\theta_k^*)} \left| u_n(r_k^*, \theta_k^*) - (U_n)_{(r_k^*,\theta_k^*)} \right|,
$$

where $(U_n)_{ij}$, $(U_n)_{(r_k^*,\theta_k^*)}$ are computed normal derivative at grid points and at the projections, respectively. At a projection, we compare both $(U_n^+)_{(r_k^*,\theta_k^*)}$ and $(U_n^-)_{(r_k^*,\theta_k^*)}$ with the exact normal derivatives from each side of the interface. The convergence order $r_E(M)$ is defined as

$$
(5.9) \qquad r_E(M) = \frac{\log(E(M)/E(2M))}{\log 2}.
$$

We see that the computed projections are *third* order accurate and that the computed solution, the normal derivative at the grid points, and the normal derivative at the projections are all second order accurate.

TABLE 5.1
Numerical results and convergence analysis for singular sources with $N = 2M$, $\beta^- = \beta^+ = 1$.

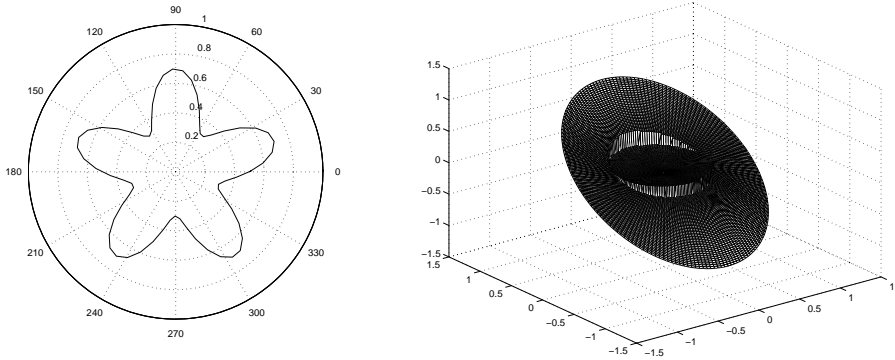| M	E_p	r_p	$E_{u_n,g}$	$r_{u_n,g}$	$E_{u_n,\Gamma}$	$r_{u_n,\Gamma}$	E_u	r_u
40	$1.915 \ 10^{-4}$		$1.294 \ 10^{-2}$		$9.044 \ 10^{-3}$		$1.877 \ 10^{-3}$	
80	$2.452 \ 10^{-5}$	2.97	$3.053 \ 10^{-3}$	2.08	$2.071 \ 10^{-3}$	2.13	$3.384 \ 10^{-4}$	2.47
160	$3.079 \ 10^{-6}$	2.99	$6.147 \ 10^{-4}$	2.31	$5.262 \ 10^{-4}$	1.98	$7.427 \ 10^{-5}$	2.19
320	$3.866 \ 10^{-7}$	2.99	$1.595 \ 10^{-4}$	1.95	$1.476 \ 10^{-4}$	1.83	$1.786 \ 10^{-5}$	2.06
640	$4.834 \ 10^{-8}$	3.00	$3.768 \ 10^{-5}$	2.08	$3.759 \ 10^{-5}$	1.97	$4.245 \ 10^{-6}$	2.07

FIG. 5.1. *Left: the domain $r = 1$ and the interface $r = 0.5 + 0.2 \sin(5\theta)$ used in Table 5.2 and Table 5.3. Right: the solution used for Table 5.1.*

5.2. Numerical examples in which β is a piecewise constant. Now we consider the case where β is discontinuous. First we present the grid refinement analysis for the same exact solution as in the previous example. Therefore the solution is independent of β. Now we use a more complicated interface

$$(5.10) \qquad \varphi(r,\theta) = r - \left(\frac{1}{2} + \frac{1}{5} \sin(5\theta) \right), \quad 0 \le r \le 1, \quad 0 \le \theta < 2\pi;$$

see the left side of Figure 5.1. The source term now is

$$(5.11) \qquad f(r,\theta) = \begin{cases} 16\beta^- \, r^2 & \text{if } \varphi(r,\theta) < 0, \\ 3\,\beta^+ \, \sin\theta & \text{if } \varphi(r,\theta) > 0. \end{cases}$$

We normalize the PDE and the jump condition (1.3)–(1.5) so that $\max\{\beta^-, \ \beta^+\} = 1$ by dividing the larger β from the PDE (1.3) and the second jump condition in (1.5).

In Table 5.2, we present the grid refinement analysis with a large jump in the coefficient β. The projection error is listed in the second column in Table 5.2(a), while the convergence rate is listed in the second column in Table 5.2(b). Again we see third order convergence. In Table 5.2(a), the ratio of β opposite of the interface is $\rho = \beta^-/\beta^+ = 10^{-3}$. It is $\rho = 10^3$ in Table 5.2(b). In either the case, we can see that the average convergence rates for the rest of the quantities are quadratic. The number of iterations, or the number of calls to the fast Poisson solver, is small (less than 10). It actually decreases as the mesh size increases. It is almost the same for the different ratios of β^- and β^+.

TABLE 5.2
Numerical results and convergence analysis for a discontinuous coefficient and singular sources with $N = 2M$. The solution is independent of the coefficient β.

(a) $\beta^- = 10^{-3}$, $\beta^+ = 1$.

M	E_p	$E_{u_n,g}$	$r_{u_n,g}$	$E_{u_n,\Gamma}$	$r_{u_n,\Gamma}$	E_u	r_u	No.
40	$3.828\ 10^{-3}$	0.1204		0.1746		$4.099\ 10^{-2}$		10
80	$4.565\ 10^{-4}$	$3.147\ 10^{-2}$	1.94	$4.857\ 10^{-2}$	1.85	$9.972\ 10^{-3}$	2.04	8
160	$8.954\ 10^{-5}$	$4.292\ 10^{-3}$	2.87	$1.815\ 10^{-2}$	1.42	$1.551\ 10^{-3}$	2.69	8
320	$1.035\ 10^{-5}$	$1.448\ 10^{-3}$	1.58	$2.503\ 10^{-3}$	2.86	$2.628\ 10^{-4}$	2.56	5
640	$1.247\ 10^{-6}$	$1.716\ 10^{-4}$	3.08	$5.608; 10^{-4}$	2.16	$7.534\ 10^{-5}$	1.80	5

(b) $\beta^- = 1$, $\beta^+ = 10^{-3}$.

M	r_p	$E_{u_n,g}$	$r_{u_n,g}$	$E_{u_n,\Gamma}$	$r_{u_n,\Gamma}$	E_u	r_u	No.
40		10.77		12.59		3.150		22
80	3.1	2.5425	2.08	2.5425	2.21	0.6301	2.21	16
160	2.4	$1.066\ 10^{-2}$	7.89	$3.628\ 10^{-2}$	6.23	$2.701\ 10^{-3}$	7.86	7
320	3.1	$3.217\ 10^{-3}$	1.73	$4.198\ 10^{-3}$	3.11	$8.041\ 10^{-4}$	1.75	7
640	3.1	$1.013\ 10^{-3}$	1.67	$1.075\ 10^{-3}$	1.97	$2.043\ 10^{-4}$	1.98	6

TABLE 5.3
Numerical results and convergence analysis for a discontinuous coefficient and singular sources with $N = 2M$ for the solution dependent on β.

(a) $\beta^- = 10^{-3}$, $\beta^+ = 1$. *The last column is the number of iterations without the preconditioning strategy described in section 4.5.*

M	$E_{u_n,g}$	$r_{u_n,g}$	$E_{u_n,\Gamma}$	$r_{u_n,\Gamma}$	E_u	r_u	No.	$No.^{np}$
40	$5.118\ 10^{-2}$		$3.118\ 10^{-2}$		$6.444; 10^{-2}$		12	21
80	$1.261\ 10^{-2}$	2.02	$1.126\ 10^{-2}$	1.50	$1.546\ 10^{-2}$	2.06	12	22
160	$1.615\ 10^{-3}$	2.97	$3.335\ 10^{-3}$	1.75	$2.853\ 10^{-3}$	2.44	13	21
320	$3.798\ 10^{-4}$	2.09	$7.267\ 10^{-4}$	2.20	$5.720\ 10^{-4}$	2.32	14	26
640	$8.295\ 10^{-5}$	2.20	$1.428\ 10^{-4}$	2.35	$1.541\ 10^{-4}$	1.89	9	33

(b) $\beta^- = 1$, $\beta^+ = 10^{-3}$.

M	$E_{u_n,g}$	$r_{u_n,g}$	$E_{u_n,\Gamma}$	$r_{u_n,\Gamma}$	E_u	r_u	No.
40	$1.934\ 10^{-2}$		$1.373\ 10^{-1}$		$1.772\ 10^{-3}$		27
80	$8.445\ 10^{-3}$	1.20	$4.316\ 10^{-2}$	1.67	$2.635\ 10^{-4}$	2.75	18
160	$2.172\ 10^{-3}$	1.96	$1.853\ 10^{-2}$	1.22	$7.565\ 10^{-5}$	1.80	22
320	$7.661\ 10^{-4}$	1.50	$1.895\ 10^{-3}$	3.29	$2.374\ 10^{-5}$	1.67	19
640	$1.612\ 10^{-4}$	2.25	$5.740\ 10^{-4}$	1.72	$4.654\ 10^{-6}$	2.35	22

In Table 5.3, we present the grid refinement analysis with the solution dependent on the coefficient β:

$$(5.12) \qquad u(r,\theta) = \begin{cases} \dfrac{r^4}{\beta^-} & \text{if } \varphi(r,\theta) < 0, \\[2mm] \dfrac{r^2 \sin\theta}{\beta^+} & \text{if } \varphi(r,\theta) \geq 0. \end{cases}$$

The source term is also adjusted accordingly. In Table 5.3, we show the grid refinement analysis with two extreme ratios of β. The problem is more difficult because one of the solutions that is divided by the smaller β has large magnitude. We still observe second order convergence for all the quantities. The number of GMRES iterations remains small compared to the size of the problems and the large jump in the coefficient.

TABLE 5.4

Numerical results and convergence analysis for a discontinuous coefficient and singular sources with $N = 2M$, $\beta^- = 1$, and $\beta^+ = 10^{-2}$. The interface is defined in (5.2), which has smaller curvature than that of (5.10). The ratio in β, $\rho = \beta^-/\beta^+ = 100$, is also smaller than that used in Tables 5.2 and 5.3. As a result, the error behaves more like a noninterface problem with a more uniform convergence rate.

M	$E_{u_n,g}$	$r_{u_n,\Gamma}$	$E_{u_n,\Gamma}$	$r_{u_n,\Gamma}$	E_u	r_u	No.
40	$5.952\ 10^{-4}$		$1.064\ 10^{-3}$		$3.104\ 10^{-2}$		12
80	$1.227\ 10^{-4}$	2.28	$3.238\ 10^{-4}$	1.72	$8.499\ 10^{-3}$	1.87	13
160	$2.441\ 10^{-5}$	2.33	$9.488\ 10^{-5}$	1.77	$2.417\ 10^{-3}$	1.81	11
320	$4.746\ 10^{-6}$	2.36	$2.434\ 10^{-5}$	1.96	$5.882\ 10^{-4}$	2.04	9
640	$7.853\ 10^{-7}$	2.60	$6.083\ 10^{-6}$	2.00	$1.461\ 10^{-4}$	2.01	9

In the last column of Table 5.3(a), we show the number of iterations of the iterative method without the preconditioning strategy described in section 4.5. We see that the number of iterations is almost doubled and is increasing. The accuracy is nearly the same, so there is no need to list it.

We also tested the convergence behavior when β^+ is close to β^-. The number of iterations is small in this case. For example, when $\beta^+ = 1$ and $\beta^- = 1.1$ or vice versa, only *three or four* iterations are needed. This is true for all the tested mesh sizes $M = 40, 80, 160, 320,$ and 640.

For interface problems, the errors may not decrease monotonically; see [12] for the analysis. Still, we see that the average convergence rates for all quantities are quadratic. Usually, the more complex the interface is, the more oscillatory are the errors. In Table 5.4, we show the grid refinement analysis with a simpler interface (5.2). The other parameters are $\beta^- = 1$, $\beta^+ = 10^{-2}$, and the exact solution is defined in (5.12), which depends on β. This test case is close to the tougher example in Table 5.3(b). We see the grid refinement analysis behaves more like a regular noninterface problem with the order of convergence approaching number two more uniformly. Again the number of GMRES iterations is small and decreases slightly as the size of the mesh increases.

As a final test, we check the effect of the regularity of the level set function $\varphi(\mathbf{x})$ on the accuracy of the computed solution. Theoretically, the second order accuracy is guaranteed if $\varphi(\mathbf{x}) \in C^3$ in the neighborhood of the zero level set $\varphi(\mathbf{x}) = 0$, which is the interface. In implementation, the level set function just needs to be $\varphi(\mathbf{x}) \in C^2$ and $|\nabla\varphi(\mathbf{x})| \neq 0$ near the interface. In experiments, we have found that $\varphi(\mathbf{x}) \in C^2$ is enough for second order accuracy. If $\varphi(\mathbf{x}) \in C^1$, the algorithm still works and seems to have superlinear convergence; see Table 5.5. Ideally, the level set function should be chosen as the signed distance function, which satisfies $|\nabla\varphi(\mathbf{x})| = 1$. If a level set function is not the signed distance function, then a re-initialization technique can be applied to get the signed distance function while the interface is unchanged; see [6, 17, 20].

In the test, the level set function is chosen as

$$(5.13) \qquad \varphi(r,\theta) = \begin{cases} r - 0.5 & \text{if } r < 0.5, \\ r - 0.5 + (r - 0.5)^l & \text{if } r \geq 0.5, \end{cases}$$

where $l \geq 1$ is an integer. We see that $\varphi(\mathbf{x}) \in C^{l-1}$ but not C^l. The interface is the circle $r = 0.5$.

The exact solution is given in (5.12). The other parameters are $\beta^- = 1$, $\beta^+ = 10^{-2}$. This test case is close to the tougher example in Table 5.3(b). We see that

TABLE 5.5

Effect of the regularity of $\varphi(\mathbf{x})$ on the fast iterative method for the interface problem B. The parameters are $N = 2M$, $\beta^- = 1$, and $\beta^+ = 10^{-2}$. The interface is defined in (5.13). In the first three columns, $l = 3$ and $\varphi(\mathbf{x}) \in C^2$, we see clearly second order accuracy. In the last three columns, $l = 2$ and $\varphi(\mathbf{x}) \in C^1$ but not in C^2, the accuracy is affected.

M	$E_u, l=3$	$r_u, l=3$	No. $l=3$	$E_u, l=2$	$r_u, l=2$	No. $l=2$
40	$4.4608 \ 10^{-3}$		10	$5.4403 \ 10^{-3}$		10
80	$1.0608 \ 10^{-3}$	2.0721	9	$1.7050 \ 10^{-3}$	1.6739	10
160	$2.5915 \ 10^{-4}$	2.0333	8	$5.5425 \ 10^{-4}$	1.6212	9
320	$6.4062 \ 10^{-5}$	2.0162	6	$1.9660 \ 10^{-4}$	1.4952	9
640	$1.5920 \ 10^{-5}$	2.0087	5	$7.7421 \ 10^{-5}$	1.3445	8

when $l = 3$, $\varphi(\mathbf{x}) \in C^2$, we have second order convergence. When $l = 2$, $\varphi(\mathbf{x}) \in C^1$, the algorithm still works with a superlinear convergence. The accuracy is affected by the computation of second order derivatives in $\varphi(\mathbf{x})$. In all of cases, the number of iterations is small because the interface is simple. Furthermore, the number of iterations when $l = 3$ is slightly smaller than that when $l = 2$.

6. Conclusion. In this paper, we have proposed a new formulation that transforms certain interface problems with a discontinuous/nonsmooth solution to a problem with a smooth solution. A second order scheme for the PDE and an interpolation scheme for the normal derivative are developed and tested for the interface problems in polar coordinates. The method is easy to implement in both polar and Cartesian coordinates. There is no need to differentiate the jumps along the interface and use the local coordinates as the original IIM does. Coupled with the fast IIM idea, we have also developed a second order fast iterative method for elliptic interface problems with a piecewise constant but a discontinuous coefficient in polar coordinates. The number of iterations of the method is almost independent of the mesh sizes and the jumps in the coefficient.

REFERENCES

[1] J. ADAMS, P. SWARZTRAUBER, AND R. SWEET, *Fishpack*, http://www.netlib.org/fishpack/.

[2] R. P. BEYER AND R. J. LEVEQUE, *Analysis of a one-dimensional model for the immersed boundary method,* SIAM J. Numer. Anal., 29 (1992), pp. 332–364.

[3] D. CALHOUN, *A Cartesian Grid Method for Solving the Streamfunction-Vorticity Equations in Irregular Geometries*, Ph.D. thesis, University of Washington, Seattle, WA, 1999.

[4] H. W. CHEN AND L. GREENGARD, *A method of images for the evaluation of electrostatic fields in systems of closely spaced conducting cylinders,* SIAM J. Appl. Math., 58 (1998), pp. 122–141.

[5] WEINAN E, *Selected Problems in Material Science,* in Mathematics Unlimited—2001 and Beyond, Springer, Berlin, 2001, pp. 407–432.

[6] T. HOU, Z. LI, S. OSHER, AND H. ZHAO, *A hybrid method for moving interface problems with application to the Hele-Shaw flow,* J. Comput. Phys., 134 (1997), pp. 236–252.

[7] K. ITO, K. KUNISCH, AND Z. LI, *Level-set function approach to an inverse interface problem,* Inverse Problems, 17 (2001), pp. 1225–1242.

[8] M. LAI AND W.-C. WANG, *Fast direct solvers for Poisson equation on 2d polar and spherical geometries,* Numer. Methods for Partial Differential Equations, 18 (2002), pp. 56–58.

 [9] R. J. LeVeque and Z. Li, *The immersed interface methods for elliptic equations with discontinuous coefficients and singular sources,* SIAM J. Numer. Anal., 31 (1994), pp. 1019–1044.

[10] R. J. LeVeque and Z. Li, *Immersed interface methods for Stokes flow with elastic boundaries or surface tension,* SIAM J. Sci. Comput., 18 (1997), pp. 709–735.

[11] Z. Li, *The Immersed Interface Method—A Numerical Approach for Partial Differential Equations with Interfaces,* Ph.D. thesis, University of Washington, Seattle, WA, 1994.

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

[13] Z. Li and K. Ito, *Maximum principle preserving schemes for interface problems with discontinuous coefficients,* SIAM J. Sci. Comput., 23 (2001), pp. 339–361.

[14] Z. Li, H. Zhao, and H. Gao, *A numerical study of electro-migration voiding by evolving level set functions on a fixed cartesian grid,* J. Comput. Phys., 152 (1999), pp. 281–304.

[15] X.-D. Liu, R. Fedkiw, and M. Kang, *A boundary condition capturing method for Poisson's equation on irregular domains,* J. Comput. Phys., 160 (2000), pp. 151–178.

[16] A. Mayo and A. Greenbaum, *Fast parallel iterative solution of Poisson's and the biharmonic equations on irregular regions,* SIAM J. Sci. Statist. Comput., 13 (1992), pp. 101–118.

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

[18] C. S. Peskin, *The immersed boundary method,* Acta Numer., 11 (2002), pp. 1–39.

[19] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,* SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[20] J. A. Sethian, *Level Set Methods and Fast Marching Methods,* 2nd ed., Cambridge University Press, Cambridge, UK, 1999.

[21] A. Wiegmann and K. P. Bube, *The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources,* SIAM J. Numer. Anal., 35 (1998), pp. 177–200.