

# A NEW FORMULATION FOR CERTAIN INTERFACE PROBLEMS IN POLAR COORDINATES AND APPLICATIONS\*

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

**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 non-smooth or discontinuous solution to 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 Poisson solver has been developed for the interface problems with piecewise constant but discontinuous coefficient using the new formulation in polar coordinates system.

**Key words.** interface problems, singular source, delta function, level set function, discontinuous coefficients, polar coordinates, immersed interface method, smooth extension, fast Poisson solver.

**AMS subject classifications.** 65N06, 65N50

**1. Introduction.** Our original *motivation* of the paper is 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.** The *Poisson equation* with discontinuities and singularities:

$$\begin{aligned} \Delta u(\mathbf{x}) &= f(\mathbf{x}), \quad \mathbf{x} \in \mathcal{R} - \Gamma, \\ u(\mathbf{x}) &= u_0(\mathbf{x}), \quad \text{on } \partial\mathcal{R}, \end{aligned} \tag{1.1}$$

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

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

are prescribed, where  $s$  is the arc-length parameter of the interface  $\Gamma$ ,  $u_n = \frac{\partial u}{\partial \mathbf{n}} = \nabla u \cdot \mathbf{n}$  is the normal derivative of  $u$ , and  $\mathbf{n}$  is the unit normal direction of  $\Gamma$  pointing outward,

---

\*The first author was supported in part by an ARO grant, 39676-MA, and an NSF grant, DMS-00-73403. The second author is supported in part by a NSC grant 89-2115-M-007-041 from Taiwan

<sup>†</sup>Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695. (zhilin@math.ncsu.edu.)

<sup>‡</sup>Department of Mathematics, National TsingHua University, Hsinchu, Taiwan, (wangwc@math.nthu.edu.tw)

<sup>§</sup>Department of Mathematics, National Taiwan University, Taipei, Taiwan, (chern@math.ntu.edu.tw)

<sup>¶</sup>Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan, (mclai@math.nctu.edu.tw)

see Fig. 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

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

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,

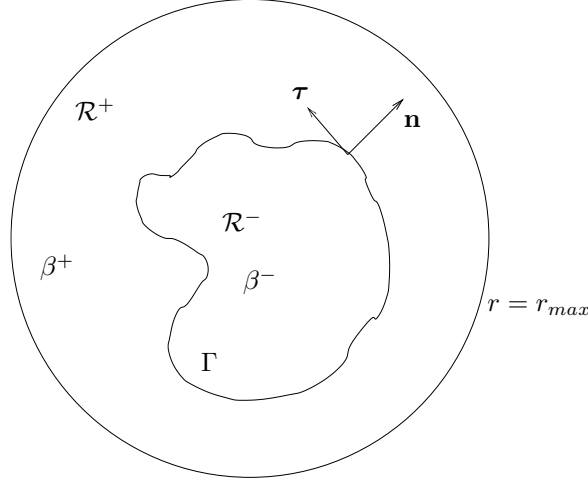


FIG. 1.1. (a). A diagram of a circular domain  $\mathcal{R} = \mathcal{R}^+ \cup \mathcal{R}^-$  with an immersed interface  $\Gamma(s)$ . The coefficients  $\beta(\mathbf{x})$  may have a jump across the interface.

we use bold face of lower case letters such as  $\mathbf{x} = (x, y)$ ,  $\mathbf{y}, \dots$  to express points in the domain  $\mathcal{R}$ , and bold face of upper case letters such as  $\mathbf{X}, \mathbf{Y}, \dots$  for points on the interface  $\Gamma$  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.** The *elliptic equation* with discontinuities and singularities:

$$\begin{aligned} \nabla \cdot (\beta(\mathbf{x}) \nabla u(\mathbf{x})) &= f(\mathbf{x}), \quad \mathbf{x} \in \mathcal{R} - \Gamma, \\ u(\mathbf{x}) &= u_0(\mathbf{x}), \quad \text{on } \partial\mathcal{R}, \end{aligned} \tag{1.3}$$

where  $\beta(\mathbf{x})$  has a constant value in each sub-domain,

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

The jump conditions

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

are prescribed along the interface  $\Gamma$ .

Note that the 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  $\Gamma$  in the domain in the sense of distribution. For example, if  $w \equiv 0$ , then the problem (1.3)-(1.5) is well known to be equivalent to

$$\begin{aligned} \nabla \cdot (\beta(\mathbf{x})\nabla u(\mathbf{x})) &= f(\mathbf{x}) + \int_{\Gamma} v(s) \delta_2(\mathbf{x} - \mathbf{X}(s)) ds, \\ u(\mathbf{x}) &= u_0(\mathbf{x}), \quad \text{on } \partial\mathcal{R}, \end{aligned} \tag{1.6}$$

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

**1.3. Some applications of the interface problems and a brief review of numerical methods.** In the computation of micro-magnetics for ferromagnetic materials, or electrostatics for macromolecules, the potential function can be model as the solution of Interface problem A or B, see [5]. For a potential problems defined on an infinite domain, usually a big circle or rectangle is used to divide the infinite domain into two parts. 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 the interface problems A and B including the IB model (1.6), in polar coordinates.

The interface problems A and B probably can be solved with some efforts using other methods as well, for example, finite element methods with a body fitted grid; the first order (in the infinite norm) ghost fluid method (GFM) [15], which is simple to implement with a symmetric linear system of equations; 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, there are few discussions 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 Poisson equation in *polar coordinates* because the equation (1.3) has non-constant coefficients 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.

---

<sup>1</sup>In some of the literature,  $\Gamma$  is called as an internal boundary, and (1.5) is called as internal boundary conditions.

**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 the interface problem A to a problem with a smooth solution. Theoretical analysis is given for the new formulation.

In Section 3, a new numerical method that uses the new formulation introduced in Section 2 is proposed. The method is second order accurate in the maximum norm. The cost of the new method is about 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], the ghost fluid method [15] and maybe some others. The question is how to do it correctly and accurately to obtain efficient and accurate algorithm. Our proposed extension is particularly simple when the interface  $\Gamma$  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 the normal derivative.

In Section 4, we generalize the fast immersed interface method developed in [12] for the 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 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 the interface problem A.** Let us repeat the interface problem A below:

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

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

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

We assume that  $\varphi(\mathbf{x}) \in C^3(\mathcal{R})^2$  in the neighborhood of the interface  $\Gamma$ , 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  $\Gamma$ , we define the extensions of  $w(\mathbf{X}(s))$  and  $v(\mathbf{X}(s))$  along the normal line (both directions) as

$$w_e(\mathbf{x}) = w_e(\mathbf{X}(s) + \alpha \mathbf{n}) = w(\mathbf{X}(s)), \tag{2.3}$$

---

<sup>2</sup>In the algorithms implementation,  $\varphi(\mathbf{x})$  just needs to be in  $C^2(\mathcal{R})$ , see Section 3-5. Second order accurate results are obtained when  $\varphi(\mathbf{x}) \in C^2$ .

and

$$v_e(\mathbf{x}) = v_e(\mathbf{X}(s) + \alpha \mathbf{n}) = v(\mathbf{X}(s)), \quad (2.4)$$

for all  $\alpha \in \mathcal{R}$  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

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

Note that  $\tilde{u}(\mathbf{x}) \in C^2$  in the neighborhood of the interface  $\Gamma$  since we assume that  $w(s)$ ,  $v(s)$ , and  $\Gamma$  are all in  $C^2$ . Define also

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

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),  $\hat{u}(\mathbf{x})$  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:*

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

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

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

**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

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

What are left to prove are 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  $w_e(\mathbf{X}(s))$ . Differentiate 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

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

here again 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  $\Gamma$ , and  $\nabla w_e(\mathbf{x}) \cdot \mathbf{n} = 0$ , we conclude that

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

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

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

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

**3. A numerical algorithm for the interface problems A.** 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 partial differential equation from a particular side at the grid points near the interface.

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

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

$$r_i = \left(i - \frac{1}{2}\right) \Delta r, \quad i = 1, 2, \dots, M \quad \text{with} \quad r_M = r_{max}, \quad \Delta r = \frac{r_{max}}{M - \frac{1}{2}}; \quad (3.1)$$

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

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

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

for later use. We choose such a grid that the fast Poisson solver developed in [8] can be used. The level set function is defined at the grid points according to  $\varphi_{ij} = \varphi(r_i, \theta_j)$ . 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 part is 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 upper case letters such as  $U$ ,  $U_{ij}$ ,  $U_n$ , etc., for discrete approximations, and lower case letters such as  $u$ ,  $q$ ,  $\varphi$  etc., for the exact solutions or exact quantities unless stated otherwise.

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

$$\begin{aligned}\varphi_{i,j}^{max} &= \max\{\varphi_{i-1,j}, \varphi_{i,j}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\}, \\ \varphi_{i,j}^{min} &= \min\{\varphi_{i-1,j}, \varphi_{i,j}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\}.\end{aligned}\quad (3.3)$$

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

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

We define  $\mathbf{x}_{ij} = (r_i, \theta_j)$  as an *sub-irregular 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 sub-irregular grid point, then we call it a *regular grid point*.

Note that all the irregular and sub-irregular points are located within 2 grid size from the interface. *In our numerical scheme, we only need to extend the jumps to all the irregular and sub-irregular grid points.* Given any such a 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 part is to find the projections effectively and accurately in the polar coordinates. Denote the projection of  $\mathbf{x} = (r, \theta)$  on the interface as  $\mathbf{X}^*$ . Assuming that  $\mathbf{x}$  is near the interface, we can write

$$\mathbf{X}^* = \mathbf{x} + \alpha \mathbf{p}, \quad \text{where} \quad \mathbf{p} = \begin{bmatrix} \varphi_r(\mathbf{x}) \\ \frac{\varphi_\theta(\mathbf{x})}{r^2} \end{bmatrix}. \quad (3.5)$$

Using the Taylor expansion at  $\mathbf{x}$ , we get the following quadratic equation for the unknown scalar  $\alpha$ :

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

where

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

In our algorithm,  $\mathbf{x}$  is an irregular or sub-irregular grid point. The partial derivatives  $\nabla \varphi(\mathbf{x})$ ,  $\varphi_{rr}$ ,  $\varphi_{r\theta}$ , 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:** If (3.6) has more than one solution, which can happen only if the curvature is very large ( $O(\frac{1}{h})$ ) relative to the grid, we just need to take one solution. 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 exist, see [2] for the reasoning of 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 sub-irregular.
- Find the projections for irregular and sub-irregular grid points using (3.5) and (3.6).
- Extend the jumps to irregular and sub-irregular grid points from the projections to get  $\tilde{u}_{ij}$  and  $\hat{u}_{ij}$  defined in (2.5) and (2.6).
- Form the discrete Laplacian

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

where  $\Delta_h$  is the standard central finite difference operator. In polar coordinates, for example, it is

$$\begin{aligned} \Delta_h U_{ij} = & \frac{U_{i-1,j} - 2U_{ij} + U_{i+1,j}}{(\Delta r)^2} + \frac{1}{r_i} \frac{U_{i+1,j} - U_{i-1,j}}{2\Delta r} \\ & + \frac{1}{r_i^2} \frac{U_{i,j-1} - 2U_{ij} + U_{i,j+1}}{(\Delta \theta)^2}. \end{aligned}$$

- 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 sub-irregular 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

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

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.  $C_{ij} = 0$ ), then the finite difference scheme (3.8) is first order accurate because the second order derivatives  $q(\mathbf{x})$  are discontinuous. The jumps in  $\Delta q$  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 from the grid point  $\mathbf{x}_{ij}$ . The correction term can be written as

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

where  $(i_k, j_k) = \{(-1, 0), (1, 0), (0, -1), (0, 1)\}$ ,  $H(\cdot)$  once again is the Heaviside function<sup>3</sup>, and  $\gamma_{i+i_k, j+j_k}$  are the coefficients of the discrete Laplacian  $\Delta_h$  corresponding

<sup>3</sup>In the discrete case, we define  $H(\varphi) = 1$  if  $\varphi \geq 0$ .



to  $U_{i+i_k, j+j_k}$ . In polar coordinates, these coefficients are

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

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,  $F(\mathbf{x}) = f(\mathbf{x}) - H(\varphi)\Delta\tilde{u}(\mathbf{x})$ . Then we have*

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

for any point on the interface  $\Gamma$ .

**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

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

The following lemma is the basis for the correction terms.

LEMMA 3.2. *Let  $\varphi(\mathbf{x}) \in C^3$ ,  $\mathbf{x}_{ij}$  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*

$$\begin{aligned} \Delta_h^r q(\mathbf{x}_{ij}) &\stackrel{\text{define}}{=} \frac{q(\mathbf{x}_{i-1, j}) - 2q(\mathbf{x}_{ij}) + q(\mathbf{x}_{i+1, j})}{\Delta r^2} + \frac{1}{r_i} \frac{q(\mathbf{x}_{i+1, j}) - q(\mathbf{x}_{i-1, j})}{2\Delta r} \\ &= \frac{\partial^2 q}{\partial r^2}(\mathbf{x}_{ij}) + \frac{1}{r_i} \frac{\partial q}{\partial r}(\mathbf{x}_{ij}) + \gamma_{i+1, j} \left( \frac{\varphi_{i+1, j}}{|\nabla \varphi_{i+1, j}|} \right)^2 \frac{F_{i+1, j} - F_{ij}}{2} \end{aligned} \quad (3.13)$$

**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

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

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

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

which is known to be an approximate solution of the 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

$$\begin{aligned} q(\mathbf{x}_{i+1,j}) &= q^+(\mathbf{X}^*) + \nabla q^+(\mathbf{X}^*) \cdot (\mathbf{x}_{i+1,j} - \mathbf{X}^*) \\ &\quad + \frac{1}{2} (\mathbf{x}_{i+1,j} - \mathbf{X}^*)^T \text{He}(q^+(\mathbf{X}^*)) (\mathbf{x}_{i+1,j} - \mathbf{X}^*) + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3), \end{aligned}$$

where  $\text{He}(q^+(\mathbf{X}^*))$  is the Hessian matrix of  $q(\mathbf{x})$  at  $\mathbf{X}^*$ . Since

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

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

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

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

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

We have

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

Therefore

$$\begin{aligned} \Delta_h^r q(\mathbf{x}_{ij}) &= \frac{q(\mathbf{x}_{i-1,j}) - 2q(\mathbf{x}_{ij}) + q(\mathbf{x}_{i+1,j})}{\Delta r^2} + \frac{1}{r_i} \frac{q(\mathbf{x}_{i+1,j}) - q(\mathbf{x}_{i-1,j})}{2\Delta r} \\ &= \frac{q(\mathbf{x}_{i-1,j}) - 2q(\mathbf{x}_{ij}) + q^-(\mathbf{x}_{i+1,j})}{\Delta r^2} + \frac{1}{r_i} \frac{q^-(\mathbf{x}_{i+1,j}) - q(\mathbf{x}_{i-1,j})}{2\Delta r} \\ &\quad + \gamma_{i+1,j} \left( \frac{1}{2} |\mathbf{x}_{i+1,j} - \mathbf{X}^*|^2 \left[ \frac{\partial^2 q(\mathbf{x}^*)}{\partial n^2} \right] + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3) \right). \end{aligned}$$

Since  $q^-(\mathbf{x}_{i+1,j})$  is the extension of  $q^-(\mathbf{x})$  from  $\mathcal{R}^-$  into  $\mathcal{R}^+$ , the expression above can be written as

$$\begin{aligned} \Delta_h^r q(\mathbf{x}_{ij}) &= \frac{\partial^2 q}{\partial r^2}(\mathbf{x}_{ij}) + \frac{1}{r_i} \frac{\partial q}{\partial r}(\mathbf{x}_{ij}) \\ &\quad + \gamma_{i+1,j} \left( \frac{1}{2} |\mathbf{x}_{i+1,j} - \mathbf{X}^*|^2 \left[ \frac{\partial^2 q(\mathbf{X}^*)}{\partial n^2} \right] + O(|\mathbf{x}_{i+1,j} - \mathbf{X}^*|^3) \right). \end{aligned} \tag{3.15}$$

Finally from (3.14) and Lemma 3.1, we have

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

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.  $\square$

Lemma 3.2 tells us the contribution to the correction term from  $\mathbf{x}_{i+1,j}$  if it is in the different side of the interface from 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 different side of the interface from  $\mathbf{x}_{ij}$ .

With the correction term  $C_{ij}$ , we have matched the second order derivatives in the finite difference scheme. 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 exact solution when the solution is a piecewise quadratic function.

Since the left hand side 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 to dealing with the interface is  $O(N_1)$  where  $N_1$  is the total number of irregular and sub-irregular 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  $|\varphi(\mathbf{x})| \leq \delta$ , where  $\delta$  is a given width, that contains the interface, we can naturally define the ‘normal derivative’ at all grid points in the tube as

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

Therefore we can get second order accurate ‘normal derivative’ at regular grid point 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 applications of the level set method in which the velocity is

calculated at the grid points. The scheme discussed here will also be used in sub-section 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 be in  $C^2(\mathcal{R})$ , we can compute  $\nabla\varphi/|\nabla\varphi|$  to second order accuracy using the centered differencing at all grid points. It remains to evaluate  $(u_r, u_\theta)$  to second order accuracy. Again, we take advantage of the smooth property of  $q(\mathbf{x})$  with the following formula

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

$$\frac{\partial U_{ij}}{\partial r} \approx \begin{cases} \frac{U_{i+1,j} - U_{i-1,j}}{2\Delta r}, & \text{otherwise,} \end{cases}$$

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

$$\frac{\partial U_{ij}}{\partial \theta} \approx \begin{cases} \frac{U_{i,j+1} - U_{i,j-1}}{2\Delta\theta}, & \text{otherwise,} \end{cases}$$

where the correction term is

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

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

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

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

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 sub-section 3.2.

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

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

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

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

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

excluding the interface  $\Gamma$ . This is a Poisson equation which 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 can not divide  $\beta$  from the flux jump condition because  $\beta$  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 following system

$$\begin{aligned} \Delta u &= \frac{f}{\beta}, \quad \text{if } \mathbf{x} \in \mathcal{R}^+ \cup \mathcal{R}^- - \Gamma, \\ [u] &= w(s), \quad [\beta u_n] = v(s), \quad [u_n] = g(s). \end{aligned} \quad (4.2)$$

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  $\mathbf{X}_k^*$  ( $k = 1, 2, \dots, N_\Gamma$ ) of the irregular grid points from the  $\varphi \geq 0$  side, where  $N_\Gamma$  is the number of such projections. We call these points as the *control points*. The reason to choose 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  $G = [G_1, G_2, \dots, G_{N_\Gamma}]^T$  in the discretization. The dimension of  $G$  is much smaller than that of the solution of  $U$  that is defined at all the grid points. Therefore we use the GMRES method [19] to find the discrete intermediate unknown  $G = [U_n]$ . 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  $[U_n]$  and computing the residual.** There are two coupled equations for the unknown  $u(\mathbf{x})$  and  $g(s)$ . The first equation is the PDE  $\Delta u = f(\mathbf{x})/\beta$  excluding the interface  $\Gamma$  with  $[u] = w(s)$  and  $[u_n] = g(s)$  are given. The solution depends on  $g(s)$  and can be written as  $u_g(\mathbf{x})$ . The second equation is the flux jump condition  $[\beta \frac{\partial u_g}{\partial n}] = v(s)$ . So there are two steps discretizations:

- 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)

$$AU + BG = F + F_w = F_1, \quad (4.3)$$

where  $U$  is 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 five point 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  $[u_n] = g$ .

- 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

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

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

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

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

$$\begin{aligned} (D - EA^{-1}B)G &= F_2 - EA^{-1}F_1, \\ \text{or } SG &= b, \end{aligned} \quad (4.6)$$

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

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

of the Schur complement system. In practice, the matrices  $A, B, E$ , and  $D$ , are not explicitly formed since the GMRES method only requires to compute the residuals.

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  $\frac{\partial u^\pm}{\partial n}(\mathbf{X}_k^*)$ ,  $k = 1, 2, \dots, N_\Gamma$ , at the control points.

#### 4.2. An outline of the algorithm for solving the interface problem B.

- Set-up the grid (3.1).
- Label the grid points as regular, irregular, and sub-irregular.
- Find the projections of irregular and sub-irregular 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 the equation (4.6).
- Set  $G_k^0 = v(\mathbf{X}_k^*)$ ,  $k = 1, 2, \dots, N_\Gamma$ , at the control points as an initial guess of  $G$ .
- Use the GMRES method to solve the system of (4.6) with some pre-conditioning techniques.

There are several important implementation details that are addressed at 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) and (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 sub-irregular 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

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

to get an approximation of  $[u_n](\mathbf{X}_k^*)$  from those  $[U_n](\mathbf{X}_p^*)$  at the control points. The summation include 3 ~ 6 projections from the  $\varphi \geq 0$  side within a circle:

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

In our experiment, we choose  $\epsilon$  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 of them that are closer to  $\mathbf{X}_k^*$  than the others in the circle. The coefficients  $\gamma_p$  are obtained from the un-determined coefficient method by expanding  $g(\mathbf{X}_p^*)$  along the interface, see Section 3.2 in [14] for the detailed description of the least square interpolation scheme to determine  $\gamma_p$ . Note that we have omitted the dependency of  $p$  on the index  $k$  for simplification of notations.

**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 Poisson equation (4.1a)-(4.1b) to get  $U(G)$ . The second step is to compute the residual vector  $R(G)$  in (4.7) which is the same as  $\beta^+ U_n^+(G) - \beta^- U_n^-(G) - V$ . Therefore we need to compute  $U_n^+(G_k)$  and  $U_n^-(G_k)$  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  $\frac{\partial q^\pm}{\partial n}(\mathbf{X}_k^*)$ ,  $k = 1, 2, \dots, N_\Gamma$ , 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  $(Q_n^\pm)_{ij} \approx q_n(\mathbf{x}_{ij})$  at grid points in sub-section 3.3. Therefore we can use an interpolation scheme to get  $(Q_n^\pm)_k$ , an approximation to  $q_n^\pm(\mathbf{X}_k^*)$ , from the values at nearby grid points. Since  $q_n^+(\mathbf{X}_k^*) = q_n^-(\mathbf{X}_k^*)$ , we can either use

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

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

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

and set  $(Q_n^+)_k = (Q_n^-)_k$ . Due to the jumps in the second order derivatives  $q(\mathbf{x})$ , the results obtained from (4.9) and (4.10) are slightly different. We will explain in the

next sub-section for 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  $\varphi_{ij} < 0$ , then  $(Q_n^+)_{ij}$  is regarded as the smooth extension (up to second order derivatives) of  $Q_n^+$  from the + side and can be computed using

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

Similarly, we use

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

to extend the  $(Q_n^-)_{ij}$  from the - side to an irregular grid point where  $\varphi_{ij} > 0$  if necessary. The discrete average jump  $[F]_{ij}$  is defined as

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

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 can not be zero. Once we have computed  $Q_n^\pm(\mathbf{X}_k^*)$ , the normal derivative of the solution at  $\mathbf{X}_k^*$  is then determined from

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

We use a second order interpolation scheme. The coefficients  $\gamma_{ij}^k$  are determined from the weighted least squares interpolation, see Section 4 in [12]. The linear system of equations for  $\gamma_{ij}^k$  is under-determined and is solved by a singular value decomposition (SVD) approach. One of the advantages of such interpolation is that the magnitude of the coefficients can not be very large regardless of the position of  $\mathbf{X}_k^*$ . The total 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 the 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 pre-conditioning techniques are crucial to reduce the number of iterations. Note that we do not form the iteration matrix for the Schur complement system. Our strategy of the pre-conditioning is to enforce the flux jump condition during the iteration which seems to work well. The pre-conditioning



techniques that we have implemented are the following:

$$\text{If } \beta^+ < \beta^- : \quad \begin{cases} (Q_n^+)_k = \sum_{ij} \gamma_{ij}^k (Q_n^-)_{ij}, & (U_n^+)_k = (Q_n^+)_k + \hat{u}_n(\mathbf{X}_k^*), \\ (U_n^-)_k = (Q_n^-)_k = \frac{v(\mathbf{X}_k^*) - \beta^+ G_k}{\beta^+ - \beta^-}, \end{cases}$$

$$\text{If } \beta^+ > \beta^- : \quad \begin{cases} (U_n^-)_k = (Q_n^-)_k = \sum_{ij} \gamma_{ij}^k (Q_n^+)_{ij}, \\ (U_n^+)_k = \frac{v(\mathbf{X}_k^*) - \beta^- G_k}{\beta^+ - \beta^-}, \end{cases}$$

The second part of the pre-conditioning strategy is the same as in [12] to enforce the flux jump condition, but the first part, which is discussed in the previous sub-section, 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  $O(N_o.(MN \log(MN) + N_1))$ , where  $N_o.$  is the number of iterations of the GMRES method and  $N_1$  is the total number of irregular and sub-irregular 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 are done on Sun Ultra workstations. Our computer codes<sup>4</sup> 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 efforts 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

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

with different parameters. Note that  $\varphi$  depends on both  $r$  and  $\theta$ .

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

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

To check the order of accuracy, we use two non-linear functions:

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

---

<sup>4</sup>The code is available to public upon request.

as the exact solution. Note that the solution depends on both  $r$  and  $\theta$ . The source term excluding the interface  $\Gamma$  is

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

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

$$E_p = \max_k |r_e(\theta_k^*) - r_k^*| \quad (5.5)$$

where  $(r_k^*, \theta_k^*)$ 's, are computed projections of all irregular grid points,  $r_e(\theta^*)$  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

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

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

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

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

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

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

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

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

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

**5.2. Numerical examples in which  $\beta$  is a piecewise constant.** Now we consider the case where  $\beta$  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  $\beta$ . Now we use a more complicated interface

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

TABLE 5.1

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

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

see Fig. 5.1 (a). The source term now is

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

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

TABLE 5.2

*Numerical results and convergence analysis for discontinuous coefficient and singular sources with  $N = 2M$ . The solution is independent of the coefficient  $\beta$ .*

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

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

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

$M$	$r_p$	$E_{u_n,g}$	$r_{u_n,g}$	$E_{u_n,\Gamma}$	$\mathbf{r}_{\mathbf{u}_n,\Gamma}$	$E_u$	$\mathbf{r}_u$	$No.$
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	$1.066 \cdot 10^{-2}$	7.89	$3.628 \cdot 10^{-2}$	6.23	$2.701 \cdot 10^{-3}$	7.86	7
320	3.1	$3.217 \cdot 10^{-3}$	1.73	$4.198 \cdot 10^{-3}$	3.11	$8.041 \cdot 10^{-4}$	1.75	7
640	3.1	$1.013 \cdot 10^{-3}$	1.67	$1.075 \cdot 10^{-3}$	1.64	$2.043 \cdot 10^{-4}$	1.98	6

In Table 5.2, we present the grid refinement analysis with large jump in the coefficient  $\beta$ . 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 the  $\beta$  from different side of the interface is  $\rho = \beta^-/\beta^+ = 10^{-3}$ . It is  $\rho = 10^3$  in Table 5.2 (b). In either of

the case, we can see the average convergence rates for the rest of the quantities are quadratic. The number of iterations, or the number of calls to fast Poisson solver is small (less than 10). It is actually decreasing as the mesh size increases. It is almost the same for different ratio of  $\beta^-$  and  $\beta^+$ .

In Table 5.3, we present the grid refinement analysis with the solution dependent of the coefficient  $\beta$

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

The source term is also adjusted accordingly. In Table 5.3, we show the grid refinement analysis with two extreme ratios of  $\beta$ . The problem is harder because one of the solution that is divided by the smaller  $\beta$  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 5.3

*Numerical results and convergence analysis for discontinuous coefficient and singular sources with  $N = 2M$  for the solution that depends on  $\beta$ .*

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

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

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

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

In the last column of Table 5.3, we show the number of iterations of the iterative method without the pre-conditioning strategy described in Section 4.5. We see the number of iterations is almost doubled and is growing. The accuracy is nearly the same and there is not need to list it.

We also tested the convergence behavior when  $\beta^+$  is close to  $\beta^-$ . The number of iterations is small. For example, when  $\beta^+ = 1$  and  $\beta^- = 1.1$  or vice versa, only 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 the average convergence rates for all quantities are quadratic. Usually, the more complex of the interface, the more oscillatory of 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) that depends on  $\beta$ . This test case is close to the tougher example in Table 5.3 (b). We see the grid refinement analysis behaviors more like a regular non-interface problem with the order of convergence approaches to number two more uniformly. Again the number of GMRES iterations is small and decreases a little as the size of the mesh increases.

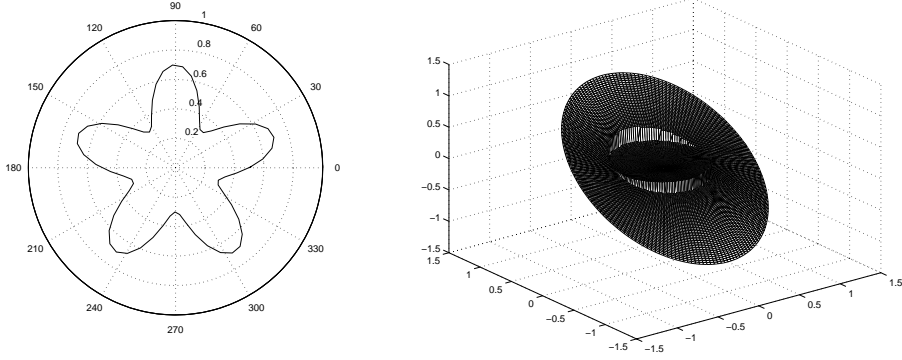


FIG. 5.1. (a) The domain  $r = 1$  and the interface  $r = 0.5 + 0.2\sin(5\theta)$  used in Table 5.2 and Table 5.3. (b) The solution used for Table 5.1 and Table 5.2.

TABLE 5.4

*Numerical results and convergence analysis for 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  $\beta$ ,  $\rho = \beta^-/\beta^+ = 100$  is also smaller than that used in Table 5.2 and Table 5.3. As a result, the error behaviors more line a non-interface problem with a more uniform convergence rate.*

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

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 needs to be  $\varphi(\mathbf{x}) \in C^2$  and  $|\nabla\varphi(\mathbf{x})| \neq 0$  near the interface. In experiment, we have found out 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 seem to have super-linear 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

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

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 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 super-linear 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. Still, the number of iterations is smaller when  $l = 3$  than that when  $l = 2$ .

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	$4.4608 \cdot 10^{-3}$		10	$5.4403 \cdot 10^{-3}$		10
80	$1.0608 \cdot 10^{-3}$	2.0721	9	$1.7050 \cdot 10^{-3}$	1.6739	10
160	$2.5915 \cdot 10^{-4}$	2.0333	8	$5.5425 \cdot 10^{-4}$	1.6212	9
320	$6.4062 \cdot 10^{-5}$	2.0162	6	$1.9660 \cdot 10^{-4}$	1.4952	9
640	$1.5920 \cdot 10^{-5}$	2.0087	5	$7.7421 \cdot 10^{-5}$	1.3445	8

**6. Conclusion.** In this paper, we have proposed a new formulation that transforms the certain interface problems with discontinuous/non-smooth 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 interface problems in polar coordinates. The method is easier to implement for 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 Poisson equations with piecewise constant but discontinuous coefficient. The number of iterations is almost independent of the mesh sizes and the jumps in the coefficient.

**7. Acknowledgment.** The first author would like to thank the National Center for Theoretical Sciences (NCTS) of Taiwan for their support during his visit there (May 8, 2001–June 8, 2001). We would also like to thank Dr. K. Ito of North Carolina State University for beneficial discussions.

#### 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.*, 29:332–364, 1992.
- [3] D. Calhoun. *A Cartesian grid method for solving the streamfunction-vorticity equations in irregular geometries*. PhD thesis, University of Washington, 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.*, 58:122–141, 1998.
- [5] Weinan E. Selected problems in material science. World Mathematics 2000, Springer, in press.
- [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.*, 134:236–252, 1997.
- [7] K. Ito, K. Kunisch, and Z. Li. Level-set function approach to an inverse interface problem. *Inverse Problems*, 17:1225–1242, 2001.
- [8] M. Lai and W-C Wang. Fast direct solvers for poisson equation on 2d polar and spherical geometries. *Numerical Methods for Partial Differential Equations*, in press, 2000.
- [9] 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.
- [10] 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.
- [11] Z. Li. *The Immersed Interface Method — A Numerical Approach for Partial Differential Equations with Interfaces*. PhD thesis, University of Washington, 1994.
- [12] Z. Li. A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, 35:230–254, 1998.
- [13] Z. Li and K. Ito. Maximum principle preserving schemes for interface problems with discontinuous coefficients. *SIAM J. Sci. Comput.*, 23:1225–1242, 2001.
- [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.*, 152:281–304, 1999.
- [15] X. Liu, R. Fedkiw, and M. Kang. A boundary condition capturing method for Poisson’s equation on irregular domain. *J. Comput. Phys.*, 160:151–178, 2000.
- [16] A. Mayo and A. Greenbaum. Fast parallel iterative solution of Poisson’s and the biharmonic equations on irregular regions. *SIAM J. Sci. Stat. Comput.*, 13:101–118, 1992.
- [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 Numerica*, pages 1–39, 2002.
- [19] Y. Saad. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.
- [20] J. A. Sethian. *Level Set Methods and Fast Marching methods*. Cambridge University Press, 2nd edition, 1999.
- [21] A. Wiegmann and K. Bube. The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 35:177–200, 1998.