FINITE DIFFERENCE METHODS FOR DISCRETIZING SINGULAR SOURCE TERMS IN A POISSON INTERFACE PROBLEM

JOHN D. TOWERS[C]

Abstract. We propose two new finite difference algorithms for a Poisson problem where jumps in the solution and its normal derivative are prescribed along an interface. Our algorithms assume a Cartesian grid that is not fitted to the interface in any way.

We employ the level set framework, making it straightforward to reformulate the problem as a Poisson problem with singular source terms that encode the jump requirements. We then discretize the singular source terms using finite difference methods for delta functions and Heaviside functions that are similar to those appearing in [34, 35, 36]. Since the jumps are captured via source terms, the resulting system of finite difference equations can be solved using any fast Poisson solver. We present the two-dimensional versions of our algorithms, but the formulas for the three-dimensional (or higher) case can be written in essentially identical forms. Thus our algorithms generalize directly to any number of spatial dimensions.

We use a recent result of Beale and Layton [2] to prove that one of our algorithms captures the solution with second order accuracy, and additionally that the discrete gradient is approximated with nearly second order accuracy. We measure accuracy in the discrete L^∞ norm.

Numerical experiments confirm the rates of convergence predicted by the Beale-Layton result for the algorithm to which that theorem applies. For the other algorithm, our numerical experiments indicate second order accuracy for the undifferentiated variable, and accuracy for the gradient that depends on the problem, varying from first order (or slightly less) to second order.

1. Introduction.

We are interested in finite difference algorithms for the following Poisson interface problem:

$$\begin{cases} \Delta u = S^+ \text{ in } \Omega_+, \quad \Delta u = S^- \text{ in } \Omega_-, \\ u = f \text{ on } \partial\Omega, \\ \left[\frac{\partial u}{\partial n}\right] = a \text{ on } \Gamma, \\ [u] = b \text{ on } \Gamma. \end{cases} \tag{1.1}$$

Here Ω_+ and Ω_- are bounded open regions in $\mathbb{R}^2$ such that $\bar{\Omega}_+ \cup \bar{\Omega}_- = \bar{\Omega}$, where Ω is an open rectangle. Γ is a one-dimensional manifold defined by the zero level set

Date: May 24, 2009.

This paper was written as part of the international research program on Nonlinear Partial Differential Equations at the Centre for Advanced Study at the Norwegian Academy of Science and Letters in Oslo during the academic year 2008-09.

[C]MiraCosta College, 3333 Manchester Avenue, Cardiff-by-the-Sea, CA 92007-1516, USA. E-mail:john.towers@cox.net.

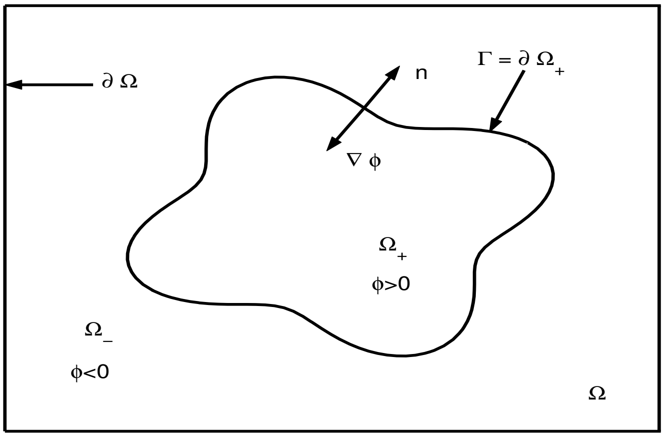

FIGURE 1. The geometry of the Poisson interface problem (1.1).

$\phi(x, y) = 0$, where the level set function $\phi : \bar{\Omega} \mapsto 0$ is assumed to be smooth and have a nonvanishing gradient near Γ. We assume that $\Gamma \cap \partial\Omega = \emptyset$. With respect to the level set function, the open sets $\Omega_\pm$ are defined by

$$\phi(x, y) > 0 \text{ for } (x, y) \in \Omega_+ \ , \quad \phi(x, y) < 0 \text{ for } (x, y) \in \Omega_-, \tag{1.2}$$

see Figure 1. The sign of ϕ is chosen so that $\partial\Omega_+ = \Gamma$. We use $\partial/\partial n$ to denote differentiation in the direction of the normal vector that points outward from Ω_+. With our sign convention, $-(\nabla\phi/ |\nabla\phi|) \cdot \nabla = \partial/\partial n$. For a piecewise continuous function $p(x, y)$ that is defined in a neighborhood of Γ, we use the notation $[p] : \Gamma \mapsto \mathbb{R}$ to denote the jump in p along Γ:

$$[p](x, y) = \lim_{\substack{(\xi,\eta)\to(x,y) \\ (\xi,\eta)\in\Omega_+}} p(\xi, \eta) - \lim_{\substack{(\xi,\eta)\to(x,y) \\ (\xi,\eta)\in\Omega_-}} p(\xi, \eta). \tag{1.3}$$

To distinguish between the sets Ω_- and Ω_+, we assume that the level set function ϕ is defined throughout $\bar{\Omega}$. However, we only require that ϕ is smooth near Γ. Specifically, we assume that for some $\alpha > 0$, ϕ is smooth on the set $B_\alpha := \{(x, y) : |\phi(x, y)| < \alpha\}$. We also assume that for some $\mu > 0$,

$$|\nabla\phi(x, y)| \geq \mu \text{ for } (x, y) \in B_\alpha. \tag{1.4}$$

An additional technical assumption concerning B_α is that $B_\alpha \subset \Omega$. We assume that $S^\pm$ is smooth on $\Omega_\pm \cup B_\alpha$. We also assume that the jump data a and b are smooth in B_α, and that the boundary data f is smooth in a neighborhood of $\partial\Omega$.

We will focus on the case of two space dimensions, but emphasize that this is only for the sake of concreteness, and to keep the notation simple. It will become clear that our algorithms and analysis generalize directly to the setting of 3 space dimensions.

Steady incompressible Stokes flow [2, 11] provides an example where a problem like (1.1) arises in fluid mechanics. In this case the interface Γ corresponds to an elastic boundary immersed in a fluid. This boundary is modeled by a forcing term F, which is concentrated on Γ. Assuming that F is known, the problem has the

form

$$\begin{cases} \Delta p = 0 \text{ in } \Omega_\pm, \\ [p] \text{ and } [\partial p/\partial n] \text{ specified}, \end{cases} \tag{1.5}$$

$$\begin{cases} \Delta u = \frac{1}{\mu} p_x \text{ in } \Omega_\pm, \\ [u] = 0 \text{ and } [\partial u/\partial n] \text{ specified}, \end{cases} \quad \begin{cases} \Delta v = \frac{1}{\mu} p_y \text{ in } \Omega_\pm, \\ [v] = 0 \text{ and } [\partial v/\partial n] \text{ specified}, \end{cases} \tag{1.6}$$

with suitable boundary conditions prescribed on $\partial\Omega$. Here p is the pressure, (u, v) is the velocity, and μ is the viscosity. The jump conditions are determined from F. The methods of this paper can be used directly to solve the problem (1.5) for the pressure. If only one fluid is present, i.e., μ is constant, the methods of this paper could also be used to solve (1.6) for u and v. One of our methods gives nearly second order accuracy for both p and its partial derivatives. After applying this method to (1.5), we could use approximations of p_x and p_y in (1.6). One technical detail here is that one would have to extend the discontinuous p_x and p_y smoothly across the elastic boundary Γ. Fortunately, there are straightforward techniques for accomplishing this extension within a level set framework, see [1, 8, 25]. For the two-fluid case where μ jumps from one constant state to another across the interface Γ, see [2, 11].

A more general interface problem than (1.1) is the variable coefficient problem

$$\begin{cases} \beta^+ \Delta u = S^+ \text{ in } \Omega_+, \quad \beta^- \Delta u = S^- \text{ in } \Omega_-, \\ u = f \text{ on } \partial\Omega, \\ \left[\beta \frac{\partial u}{\partial n}\right] = a \text{ on } \Gamma, \\ [u] = b \text{ on } \Gamma, \end{cases} \tag{1.7}$$

which includes the discontinuous coefficient

$$\beta(x, y) = H(\phi)\beta^+(x, y) + (1 - H(\phi))\beta^-(x, y).$$

Here H denotes the Heaviside function:

$$H(z) = \begin{cases} 0, & z \le 0, \\ 1, & z > 0, \end{cases} \tag{1.8}$$

and β^- and β^+ are smooth except for a jump along Γ. For the case where β is piecewise constant, the variable coefficient problem (1.7) can be reduced to (1.1) by solving a certain integral equation for $[\partial u/\partial n]$, see [2, 19].

In this paper we are assuming that jumps across the interface are specified; we then derive the singular source terms that are implied by the jumps (see Section 2). However, there are also situations where singular source terms are specified directly. A simple physical example (borrowed from [10]) is steady state heat conduction in a rectangular region Ω with a heat source concentrated on Γ. This would lead to a Poisson problem with a delta function source term.

Another example is the Immersed Boundary Method of Peskin [28, 29], which we now outline, following the description in [4]. The Immersed Boundary Method applies to the incompressible Navier-Stokes equations. It was originally devised to model the flow of blood in the heart. The computational domain is a box, generally with periodic boundary conditions. The model features a moving boundary, located entirely within the box. This boundary moves with the fluid, but also exerts a force on the fluid. What is relevant from our point of view is that this force is concentrated

on the boundary, and is captured by a delta function source term. The equations are of the form

$$\begin{cases} u_t + u \cdot \Big(\nabla u\Big) = \mu u_{xx} + f, \\ \nabla \cdot u = 0, \end{cases}$$

with $u = (u_1, u_2)$ representing the velocity, μ the viscosity, and f the force. The force f is discretized by a product of one-dimensional delta functions

$$f_{j,k} = \sum_{i=1}^{N} F_i(t)\delta^h(x_j - X_i)\delta^h(y_k - Y_i)\,\Delta s. \tag{1.9}$$

Here, the control points $(X_i, Y_i) = (X(s_i), Y(s_i))$ locate the boundary curve Γ, which is defined parametrically, $(X, Y) = (X(s), Y(s))$. Peskin proposed and analyzed various one-dimensional delta functions that can be used for the approximate delta function δ^h appearing in (1.9). Engquist, Tornberg, and Tsai [7] successfully adapted the "product of delta functions" approach to the case where Γ is represented implicitly by a level set instead of parametrically.

Beyer and Leveque [4] studied the one-dimensional time-dependent problem

$$\begin{cases} u_t = u_{xx} + c(t)\delta(x - \alpha), \quad x \in (0,1), \quad \alpha \in (0,1), \\ u(0,t) = u(1,t) = 0, \end{cases} \tag{1.10}$$

and also the steady state version where the PDE becomes $u_{xx} = -c\delta(x - \alpha)$. Their purpose was to analyze the Immersed Boundary Method in the simplest possible nontrivial case. As a part of this analysis, they derived certain moment conditions that a one dimensional discrete delta function must satisfy in order to achieve second order accuracy.

Leveque and Li [10] developed the Immersed Interface Method, a second order method for the problem (1.7), their goal being to develop a second order version of the Immersed Boundary Method. This method has been applied successfully by Leveque, Li, their coworkers, and others to a wide variety of problems including (1.7), but also more general problems [11, 12, 13, 14, 15, 16, 38].

In [10] the authors observed that the Immersed Boundary Method could be applied to problems like (1.1), but noted two limitations of the method. First, it is not possible to incorporate discontinuities in the solution, only discontinuities in the normal derivative. Second, it did not seem possible in two dimensions to achieve second order accuracy at all grid points, using a forcing function of the form (1.9), i.e., a two-dimensional discrete delta function. The results of the present paper indicate that if the interface is represented by a level set, rather than the Lagrangian representation of the Immersed Boundary Method, it is possible to overcome these limitations. Specifically, it is possible to construct a discrete delta function and a discrete dipole term which force the desired jumps in both the normal derivative and the solution itself. Moreover, this can be done in two or three dimensions, and the approximations have second order accuracy at all grid points.

Beale and Layton [2] have pointed out that at least a few of the known finite difference methods for solving (1.1) on a regular grid have the form

$$\Delta^h U_{j,k} = S(x_j, y_k) + T^h_{j,k}, \tag{1.11}$$

where Δ^h is the discrete Laplacian, (x_j, y_k) is the typical grid point, and $T^h_{j,k}$ is a term that is zero except for grid points near Γ. Clearly, $T^h_{j,k}$ comprises the nontrivial

part of the algorithm. It is responsible for the correct jumps in the approximate solution and its normal derivative. In [2] it is proven that if the finite difference approximation is second order accurate, except possibly near the interface, where the accuracy may decrease to first order, the resulting approximation will be second order accurate, measured in the discrete L^∞ norm. Additionally, the accuracy of the gradient approximation will have close to second order accuracy. For the reader's convenience, we provide a statement of their result in Section 6. Moreover, they show that the Immersed Interface Method, the related method of Weigman and Bube [38], and also the method of Mayo (described in the following paragraph), satisfy the hypotheses of their theorem, and hence verify the second order accuracy of these methods. In Section 6 we show that the theorem of [2] also applies to one of the algorithms proposed in this paper.

In order to solve a Poisson problem on an irregular domain, Mayo [19, 20, 21] has also developed a practical method for problems of the type discussed here. First, the Poisson problem on an irregular domain is reduced to a problem of the form (1.1). The main idea of the approach is to extend the solution of the original problem to a rectangular region that contains the irregular domain. This extended solution has jumps at the boundary of the irregular domain (this irregular domain corresponds to Ω_+ in our setup), but these jumps can be determined by solving a Fredholm integral equation. With the problem reduced in this way, Mayo devised a formula for the term $T^h_{j,k}$ that results in second order accuracy for both the reduced problem (1.1), and the original Poisson problem.

Liu, Fedkiw, and Kang [17] have also developed a finite difference method for (1.7). Their method is based on the Ghost Fluid Method [8]. It is first order accurate, but is simple to implement in both two and three dimensions, captures interfaces sharply, and can be used with a standard Poisson solver. Liu and Sideris [18] have analyzed this scheme, including a proof of convergence.

Two additional recent papers that address the interface problem (1.1), and also the more general problem (1.7), using a uniform Cartesian mesh are [9] and [37].

Another example where singular source terms appear in PDEs is the level set method of Osher and Sethian [22, 25, 26, 27, 30]. Level set algorithms also use delta functions and Heaviside functions in quadrature formulas such as

$$\int_{\Omega_+} g\, dA = \int_\Omega gH(\phi)\, dA, \tag{1.12}$$

$$\int_\Gamma g\, ds = \int_\Omega g\delta(\phi)\, |\nabla\phi|\, dA, \tag{1.13}$$

which are valid if $\mathrm{supp}(g) \subset \Omega$; see [5] for a proof of (1.13). In [32] and [33] Tornberg and Engquist carried out a systematic study of discretized delta functions and Heaviside functions, both as source terms in PDEs and as used in quadrature problems. Among other things, they showed that seemingly reasonable methods of discretizing delta functions may result in approximations to the integral (1.13) that are inconsistent, meaning that they generate approximations that do not converge to the correct solution. Since then a number of methods for constructing consistent delta function and Heaviside function approximations have been proposed and analyzed [3, 7, 23, 24, 31, 34, 35, 36, 39, 40]. Several of these papers have devised discrete delta functions that result in second order accuracy for the quadrature problem (1.13). However, in dimensions higher than one, there does not seem to be

a direct connection between second order accuracy for the quadrature problem and second order accuracy in the L^∞ norm for the Poisson interface problem. Both [7] and [24] give an example where a discrete delta function gives second order accuracy for the quadrature problem, but yields only first order accuracy for the Poisson interface problem. (It should be noted that if grid points outside of a narrow band surrounding the interface are excluded from the error calculation, or if the discrete L^1 norm is used, the methods of [7] and [24] give second order accuracy.)

The rest of the paper is organized as follows. In Section 2 we derive formulations of the interface problem (1.1) that capture the jump requirements via singular source terms. In Section 3 we discretize the computational domain and fix notation for various finite difference operators. In Section 4 we describe our finite difference method for discretizing Heaviside functions and delta functions. These discretizations are similar to ones that we proposed and analyzed in [34, 35, 36]. In Section 5, we discretize the singular source term formulations that we derived in Section 2, yielding new algorithms for the interface problem (1.1). In Section 6 we use the result of [2] mentioned previously to prove that one of our discretizations is second order accurate in the discrete L^∞ norm. In Section 7 we provide numerical examples. These confirm the rates of convergence predicted by the Beale-Layton result for the algorithm to which that theorem applies. For the other algorithm, our numerical experiments indicate second order accuracy for the undifferentiated variable, and accuracy for the gradient that varies, depending on the problem, from less than first order to second order. Finally, Section 8 contains some concluding remarks.

2. **Correspondence between jump conditions and singular source terms.**

In this section we derive several formulations of (1.1) that capture the jumps via singular source terms. In these calculations we operate formally with distributions like $\delta(\phi)$ and the even more singular $\Delta H(\phi)$. We emphasize that this is only as an aid in constructing algorithms. Ultimately, we check our numerical approximations against the solutions of (1.1).

Let ψ be a smooth test function supported in the interior of Ω. Starting from

$$\int_{\Omega_\pm} \psi \Delta u \, dA = \int_{\Omega_\pm} \psi S^\pm \, dA, \tag{2.1}$$

we apply Green's identity separately over Ω_- and Ω_+ to get

$$\int_{\Omega_\pm} u \Delta \psi \, dA = \int_{\partial\Omega_\pm} u \frac{\partial \psi}{\partial n_\pm} \, ds - \int_{\partial\Omega_\pm} \psi \frac{\partial u}{\partial n_\pm} \, ds + \int_{\Omega_\pm} \psi S^\pm \, dA. \tag{2.2}$$

Here $n_\pm$ is the unit normal vector outward from $\Omega_\pm$. Define

$$S(x,y) = H(\phi(x,y)) S^+(x,y) + (1 - H(\phi(x,y))) S^-(x,y). \tag{2.3}$$

Adding the integrals in (2.2) over Ω_+ and Ω_-, using the jump conditions $[u] = b$, $[\partial u/\partial n] = a$, along with (1.13), we obtain

$$\begin{aligned} \int_\Omega u \Delta \psi \, dA &= \int_\Gamma [u] \frac{\partial \psi}{\partial n} \, ds - \int_\Gamma \psi \left[\frac{\partial u}{\partial n} \right] ds + \int_\Omega \psi S \, dA \\ &= \int_\Omega b \frac{\partial \psi}{\partial n} \delta(\phi) \, |\nabla \phi| \, dA - \int_\Omega \psi a \delta(\phi) \, |\nabla \phi| \, dA + \int_\Omega \psi S \, dA. \end{aligned} \tag{2.4}$$

Next, we wish to replace the integral involving $\partial\psi/\partial n$ by one or more integrals that instead involve the undifferentiated test function ψ. Using the formal relationship

$$\nabla H(\phi) = H'(\phi)\nabla(\phi) = \delta(\phi)\nabla\phi, \tag{2.5}$$

we compute

$$\begin{aligned}
\int_\Omega \psi b\Delta H(\phi)\, dA &= \int_\Omega \nabla\cdot\Big(\psi b\nabla H(\phi)\Big)\, dA - \int_\Omega \nabla H(\phi)\cdot\nabla\Big(\psi b\Big)\, dA \\
&= -\int_\Omega \delta(\phi)\nabla\phi\cdot\nabla\Big(\psi b\Big)\, dA \\
&= \int_\Omega \delta(\phi)\left(b\frac{\partial\psi}{\partial n} + \psi\frac{\partial b}{\partial n}\right)|\nabla\phi|\, dA \\
&= \int_\Omega b\frac{\partial\psi}{\partial n}\delta(\phi)\,|\nabla\phi|\, dA + \int_\Omega \psi\frac{\partial b}{\partial n}\delta(\phi)\,|\nabla\phi|\, dA.
\end{aligned} \tag{2.6}$$

Solving for the integral involving $\partial\psi/\partial n$, we find that

$$\int_\Omega b\frac{\partial\psi}{\partial n}\delta(\phi)\,|\nabla\phi|\, dA = \int_\Omega \psi b\Delta H(\phi)\, dA - \int_\Omega \psi\frac{\partial b}{\partial n}\delta(\phi)\,|\nabla\phi|\, dA. \tag{2.7}$$

Substituting (2.7) into (2.4) yields

$$\begin{aligned}
\int_\Omega u\Delta\psi\, dA = \int_\Omega \psi b\Delta H(\phi)\, dA - \int_\Omega \psi a\delta(\phi)\,|\nabla\phi|\, dA - \int_\Omega \psi\frac{\partial b}{\partial n}\delta(\phi)\,|\nabla\phi|\, dA \\
+ \int_\Omega \psi S\, dA.
\end{aligned} \tag{2.8}$$

Recalling that a weak form of the Poisson equation $\Delta w = P$ is $\int_\Omega w\Delta\psi\, dA = \int_\Omega P\psi\, dA$, equation (2.8) indicates that, at least formally, the interface problem (1.1) can be expressed as

$$\begin{cases} \Delta u = b\Delta H(\phi) - \left(a + \frac{\partial b}{\partial n}\right)\delta(\phi)\,|\nabla\phi| + S \text{ in } \Omega, \\ u = f \text{ on } \partial\Omega. \end{cases} \tag{2.9}$$

Remark 2.1. In (2.9), the portion of the source term responsible for the jump $[\partial u/\partial n] = a$ is $-a\delta(\phi)\,|\nabla\phi|$. The portion that causes the jump $[u] = b$ is

$$b\Delta H(\phi) - \frac{\partial b}{\partial n}\delta(\phi)\,|\nabla\phi|\,.$$

The distribution $\Delta H(\phi)$ is a dipole type of singularity since

$$\Delta H(\phi) = \nabla\cdot(\delta(\phi)\nabla\phi) = \delta'(\phi)\,|\nabla\phi|^2 + \delta(\phi)\Delta\phi.$$

The formulation (2.9) is not quite in a form that we wish to discretize. To achieve the desired formulation, we use the identity

$$\Delta(bH) = b\Delta H + H\Delta b + 2\nabla b\cdot\nabla H \tag{2.10}$$

to write $b\Delta H$ in a slightly different way:

$$\begin{aligned}
b\Delta H &= \Delta\Big(bH\Big) - H\Delta b - 2\nabla b\cdot\nabla H \\
&= \Delta\Big(bH\Big) - H\Delta b - 2\nabla b\cdot\nabla\phi\delta(\phi) \\
&= \Delta\Big(bH\Big) - H\Delta b + 2\frac{\partial b}{\partial n}\delta(\phi)\,|\nabla\phi|\,.
\end{aligned} \tag{2.11}$$

Substituting this relationship into (2.9), we get the following formulation of the interface problem (1.1):

$$\begin{cases} \Delta u = \Delta\Big(bH\Big) - H\Delta b - \left(a - \frac{\partial b}{\partial n}\right)\delta(\phi)\,|\nabla\phi| + S \text{ in } \Omega, \\ u = f \text{ on } \partial\Omega. \end{cases} \tag{2.12}$$

This is the first of two versions of the problem (1.1) that we will discretize.

In (2.12), we can replace b by any any convenient alternative $\hat{b}$, as long as $\hat{b}$ is smooth and $\hat{b} = b$ on Γ. With this in mind, the second version of (1.1) that we will discretize is

$$\begin{cases} \Delta u = \Delta\Big(\hat{b}H\Big) - H\Delta\hat{b} + S \text{ in } \Omega, \\ \hat{b} = b + \left(\frac{\partial b}{\partial n} - a\right)\phi/\,|\nabla\phi|\,, \\ u = f \text{ on } \partial\Omega. \end{cases} \tag{2.13}$$

This version results from (2.12) via the following readily verified relationships:

$$\hat{b} = b + O(\phi), \quad \frac{\partial\hat{b}}{\partial n} = a + O(\phi). \tag{2.14}$$

With this choice of $\hat{b}$, we have not only $\hat{b} = b$ on Γ, which is required. We also have $\frac{\partial\hat{b}}{\partial n} = a$ on Γ. This second relationship results in a simplified algorithm, since no discrete delta function is needed. It turns out that this discretization also results in a higher convergence rate for the discrete gradient.

3. **Discretization of Ω and spatial difference operators.**

We take $\Omega = (-L, L)\times(-L, L)$ to be a square centered on the origin. We choose the mesh size $h = \Delta x = \Delta y$ so that grid points are located on $\partial\Omega$. The grid points are $(x_j, y_k) = (jh, kh)$, $-\mathcal{J} \le j, k \le \mathcal{J}$, where $h = \mathcal{J}/L$. For a function $Z(x_j, y_k)$ defined at the grid points (x_j, y_k), we use the abbreviation $Z_{j,k}$. To fix notation, the standard forward and backward spatial divided difference operators are denoted by

$$D^x_{\pm} Z_{j,k} = \pm\left(Z_{j\pm1,k} - Z_{j,k}\right)/h, \quad D^y_{\pm} Z_{j,k} = \pm\left(Z_{j,k\pm1} - Z_{j,k}\right)/h. \tag{3.1}$$

We will also use the centered divided difference operators:

$$D^x_0 Z_{j,k} = \frac{1}{2}\left(D^x_+ Z_{j,k} + D^x_- Z_{j,k}\right), \quad D^y_0 Z_{j,k} = \frac{1}{2}\left(D^y_+ Z_{j,k} + D^y_- Z_{j,k}\right). \tag{3.2}$$

The discrete gradient operator is defined by:

$$\nabla^h Z_{j,k} = \left(D^x_0 Z_{j,k}, D^y_0 Z_{j,k}\right), \tag{3.3}$$

and the standard five-point discrete Laplacian is

$$\Delta^h Z_{j,k} = D^x_+ D^x_- Z_{j,k} + D^y_+ D^y_- Z_{j,k}. \tag{3.4}$$

We will need the following finite difference identity :

$$\Delta^h\left(P_{j,k}Q_{j,k}\right) = P_{j,k}\Delta^h Q_{j,k} + Q_{j,k}\Delta^h P_{j,k} + 2\langle P, Q\rangle_{j,k} \tag{3.5}$$

$$\langle P, Q\rangle_{j,k} := \frac{1}{2}\left(D^x_+ P_{j,k} D^x_+ Q_{j,k} + D^x_- P_{j,k} D^x_- Q_{j,k}\right) \tag{3.6}$$

$$+\frac{1}{2}\left(D^y_+ P_{j,k} D^y_+ Q_{j,k} + D^y_- P_{j,k} D^y_- Q_{j,k}\right) \tag{3.7}$$

The relationship (3.5) is just the discrete version of the identity (2.10), with b and H replaced by P and Q.

4. Singular quantities via finite differences.

In this section we derive discrete versions of $\delta(\phi)$ and $H(\phi)$. The level set framework is convenient for discretizing singularities that are concentrated on Γ. To illustrate this point, consider the following approach to deriving a first order accurate discrete delta function that is concentrated on Γ [34]. We start with the formal calculation

$$\nabla H(\phi) = H'(\phi)\nabla\phi = \delta(\phi)\nabla\phi. \tag{4.1}$$

Taking the dot product of $\nabla\phi$ on both sides, and then solving for $\delta(\phi)$, we obtain

$$\delta(\phi) = \nabla H(\phi) \cdot \nabla\phi / \left|\nabla\phi\right|^2. \tag{4.2}$$

This formula is now readily discretized by replacing ∇ by the discrete gradient operator ∇^h. This approach works for any manifold Γ of codimension one in $\mathbb{R}^n$ that is defined by a level set. Below we will use a version of this approach that is slightly more complicated, since we wish to achieve second order accuracy.

In what follows, we will use the first several primitives of the Heaviside function:

$$\begin{aligned} &I(z) = \int_0^z H(\zeta)\,d\zeta = \max(0,z), \quad J(z) = \int_0^z I(\zeta)\,d\zeta = \frac{1}{2}\max(0,z)^2, \\ &K(z) = \int_0^z J(\zeta)\,d\zeta = \frac{1}{6}\max(0,z)^3. \end{aligned} \tag{4.3}$$

To simplify the formulas, we will usually write $H(\phi(x_j, y_k))$ in the abbreviated form $H_{j,k}$, and similarly for $I(\phi(x_j, y_k))$, $J(\phi(x_j, y_k))$, $K(\phi(x_j, y_k))$.

To derive the discrete delta function that we will use in discretizing (1.1), we start from the following (formal) calculation:

$$\begin{aligned} \Delta I(\phi) &= \nabla\cdot(\nabla I(\phi)) = \nabla\cdot(I'(\phi)\nabla\phi) \\ &= \nabla\cdot(H(\phi)\nabla\phi) = \nabla H(\phi)\cdot\nabla\phi + H(\phi)\Delta\phi \\ &= H'(\phi)\nabla\phi\cdot\nabla\phi + H(\phi)\Delta\phi = \delta(\phi)\left|\nabla\phi\right|^2 + H(\phi)\Delta\phi. \end{aligned} \tag{4.4}$$

Solving for $\delta(\phi)$, we obtain

$$\delta(\phi) = \Delta I(\phi)/\left|\nabla\phi\right|^2 - H(\phi)\Delta\phi/\left|\nabla\phi\right|^2. \tag{4.5}$$

If we repeat this calculation, starting instead with $\Delta J(\phi)$, the result is

$$H(\phi) = \Delta J(\phi)/\left|\nabla\phi\right|^2 - I(\phi)\Delta\phi/\left|\nabla\phi\right|^2. \tag{4.6}$$

Finally, we plug (4.6) into (4.5). This results in the formula for $\delta(\phi)$ that we will discretize:

$$\delta(\phi) = \Delta I(\phi)/\left|\nabla\phi\right|^2 - \left(\Delta J(\phi) - I(\phi)\Delta\phi\right)\Delta\phi/\left|\nabla\phi\right|^4. \tag{4.7}$$

Let $\mathcal{N}_1$ denote the set of grid points (x_j, y_k) which are separated from the interface $\partial\Omega$ by less than one mesh width. Specifically $(x_j, y_k) \in \mathcal{N}_1$ means that

$$\max\{\phi_{j,k}, \phi_{j\pm1,k}, \phi_{j,k\pm1}\} > 0, \quad \min\{\phi_{j,k}, \phi_{j\pm1,k}, \phi_{j,k\pm1}\} < 0. \tag{4.8}$$

By discretizing (4.7), we obtain the following discrete version of $\delta(\phi)$:

FDM2c:

$$\delta^{\mathrm{F2c}}(\phi)_{j,k} = \begin{cases} \frac{\Delta^h I_{j,k}}{|\nabla^h \phi_{j,k}|^2} - \frac{\left(\Delta^h J_{j,k} - I_{j,k}\Delta^h \phi_{j,k}\right)\Delta^h \phi_{j,k}}{|\nabla^h \phi_{j,k}|^4}, & (x_j, y_k) \in \mathcal{N}_1, \\ 0, & (x_j, y_k) \notin \mathcal{N}_1. \end{cases} \tag{4.9}$$

FDM2c is a slightly modified version of what we called Method 2 in [34]. We have replaced $\left(\nabla^h I_{j,k} \cdot \nabla^h \phi_{j,k}\right)$ in Method 2 of [34] by $\left(\Delta^h J_{j,k} - I_{j,k}\Delta^h \phi_{j,k}\right)$, which can be understood in terms of the identity $\nabla J \cdot \nabla \phi = \Delta J - I \Delta \phi$. This modified version is better suited to the problem of this paper. Let us briefly explain this. Our difference scheme will be (very roughly) of the form

$$\Delta^h U_{j,k} = \text{Singular Terms}. \tag{4.10}$$

We find that from an accuracy perspective, it is desirable to be able to write (4.10) in the equivalent form

$$\Delta^h U_{j,k} = \Delta^h \text{Something} + \text{Less Singular Terms}. \tag{4.11}$$

Both of the difference operators in (4.9) that result in singular terms are of the form Δ^h; specifically they are $\Delta^h I_{j,k}$ and $\Delta^h J_{j,k}$. In our schemes they will occur in the form $P_{j,k}\Delta^h I_{j,k}$ and $Q_{j,k}\Delta^h J_{j,k}$, where P and Q are smooth. By applying the identity (3.5) to these terms, our difference schemes can be put in the equivalent form (4.11).

The Heaviside version of (4.7) is

$$H(\phi) = \Delta J(\phi) / |\nabla \phi|^2 - (\Delta K(\phi) - J(\phi)\Delta\phi)\, \Delta\phi / |\nabla \phi|^4 . \tag{4.12}$$

After discretizing (4.12), the result is

FDMH2c:

$$H^{\mathrm{F2c}}(\phi)_{j,k} = \begin{cases} \frac{\Delta^h J_{j,k}}{|\nabla^h \phi_{j,k}|^2} - \frac{\left(\Delta^h K_{j,k} - J_{j,k}\Delta^h \phi_{j,k}\right)\Delta^h \phi_{j,k}}{|\nabla^h \phi_{j,k}|^4}, & (x_j, y_k) \in \mathcal{N}_1, \\ H(\phi_{j,k}), & (x_j, y_k) \notin \mathcal{N}_1. \end{cases} \tag{4.13}$$

To simplify notation, we will usually drop the dependence on ϕ and simply write $\delta^{\mathrm{F2c}}_{j,k}$ or $H^{\mathrm{F2c}}_{j,k}$.

Our discrete delta function and Heaviside function only require that ϕ be smooth in a narrow band of width $O(h)$ surrounding Γ. Away from this narrow band, ϕ is only required to have the correct sign.

5. Discrete versions of the Poisson interface problem.

In this section we describe discretizations of the source term problems (2.12) and (2.13). For (2.12) we discretize the delta function portion via

$$\left(a - \frac{\partial b}{\partial n}\right) \delta(\phi)\, |\nabla \phi| \Big|_{(x_j, y_k)} \approx (a_{j,k} - c_{j,k})\, \delta^{\mathrm{F2c}}_{j,k} \left|\nabla^h \phi_{j,k}\right| . \tag{5.1}$$

Here

$$c_{j,k} = \frac{\partial b}{\partial n}(x_j, y_k). \tag{5.2}$$

For the terms involving H, we use

$$\Delta\Big(bH\Big)\Big|_{(x_j, y_k)} \approx \Delta^h \left(b_{j,k} H_{j,k}\right), \quad H\Delta b\Big|_{(x_j, y_k)} \approx H^{\mathrm{F2c}}_{j,k} \Delta^h b_{j,k}. \tag{5.3}$$

In the first part of (5.3), the Heaviside function is not regularized in any way; any smoothing here would cause the solution to be smeared. On the other hand, in the second occurrence we use H^{F2c} in order to achieve second order accuracy. Finally, the term $S(x, y)$ is discretized using

$$S(x_j, y_k) \approx H^{\mathrm{F2c}}_{j,k} S^{+}_{j,k} + (1 - H^{\mathrm{F2c}}_{j,k}) S^{-}_{j,k}. \tag{5.4}$$

Let $U_{j,k}$ denote the approximate solution produced by our algorithms, i.e., $U_{j,k} \approx u(x_j, y_k)$, where u is the exact solution of (1.1). Combining (5.1), (5.3), and (5.4), the full discretization for the formulation (2.12) is then

$$\begin{cases} \Delta^h U_{j,k} = \Delta^h \Big(b_{j,k} H_{j,k} \Big) - H^{\mathrm{F2c}}_{j,k} \Delta^h b_{j,k} - (a_{j,k} - c_{j,k})\, \delta^{\mathrm{F2c}}_{j,k} \left| \nabla^h \phi_{j,k} \right| \\ \qquad\qquad + H^{\mathrm{F2c}}_{j,k} S^{+}_{j,k} + (1 - H^{\mathrm{F2c}}_{j,k}) S^{-}_{j,k}, \qquad (x_j, y_k) \in \Omega, \\ U_{j,k} = f_{j,k}, \qquad\qquad (x_j, y_k) \in \partial\Omega. \end{cases} \tag{5.5}$$

Numerical tests indicate that this discretization gives a second order approximation for u. It produces approximations of the discrete gradient $\nabla^h u(x_j, y_k)$ that vary between first order (or slightly less) and second order.

Remark 5.1. From the following calculation

$$\begin{aligned} &\Delta^h \Big(b_{j,k} H_{j,k} \Big) - H^{\mathrm{F2c}}_{j,k} \Delta^h b_{j,k} \\ &= H_{j,k} \Delta^h b_{j,k} - H^{\mathrm{F2c}}_{j,k} \Delta^h b_{j,k} + b_{j,k} \Delta^h H_{j,k} + 2 \langle b, H \rangle_{j,k}, \end{aligned} \tag{5.6}$$

it is clear that this dipole term vanishes for $(x_j, y_k) \notin \mathcal{N}_1$. The jump data $b_{j,k}$ must be defined in a slightly wider band surrounding Γ, specifically in

$$\mathcal{N}_2 := \mathcal{N}_1 \cup \{ (x_j, y_k) : (x_{j\pm1}, y_k) \in \mathcal{N}_1 \text{ or } (x_j, y_{k\pm1}) \in \mathcal{N}_1 \}.$$

Since $\delta^{\mathrm{F2c}}_{j,k}$ vanishes for $(x_j, y_k) \notin \mathcal{N}_1$, the quantities $a_{j,k}$ and $c_{j,k}$ only need to be defined for $(x_j, y_k) \in \mathcal{N}_1$. Similarly, we only need the level set function ϕ to be smooth in a narrow band surrounding Γ. Outside of that band, we only need ϕ to have the proper sign.

It turns out that the discrete version of (2.13) generally yields a better approximation to the discrete gradient $\nabla^h u(x_j, y_k)$ than (5.5). We discretize the formulation (2.13) via

$$\begin{cases} \Delta^h U_{j,k} = \Delta^h \left(\hat{b}_{j,k} H_{j,k} \right) - H^{\mathrm{F2c}}_{j,k} \Delta^h \hat{b}_{j,k}, + H^{\mathrm{F2c}}_{j,k} S^{+}_{j,k} + (1 - H^{\mathrm{F2c}}_{j,k}) S^{-}_{j,k}, \\ \qquad\qquad\qquad\qquad (x_j, y_k) \in \Omega, \\ \hat{b}_{j,k} = b_{j,k} + \left(\frac{\partial b}{\partial n}(x_j, y_k) - a_{j,k} \right) \phi_{j,k} / \left| \nabla \phi_{j,k} \right|, \\ U_{j,k} = f_{j,k}, \qquad\qquad (x_j, y_k) \in \partial\Omega. \end{cases} \tag{5.7}$$

6. **Convergence theorem for the discretization** (5.7).

In this section we prove that approximate solutions generated by the discretization (5.7) converge to the actual solution of the Poisson interface problem (1.1) at a rate of $O(h^2)$ in the discrete L^∞ norm. We will use the following result:

Theorem 6.1 (Beale, Layton [2])**.** *Let u be the exact solution of the interface Poisson problem* (1.1)*. Assume that Γ is C^4, and that $u \in C^4(\bar{\Omega}_+)$, $u \in C^4(\bar{\Omega}_-)$. Assume that the solution is approximated using*

$$\begin{cases} \Delta^h U_{j,k} = S_{j,k} + T^h_{j,k}, & (x_j, y_k) \in \Omega, \\ U_{j,k} = f(x_j, y_k), & (x_j, y_k) \in \partial\Omega, \end{cases} \tag{6.1}$$

where $T^h_{j,k} = 0$ if $(x_j, y_k) \notin \mathcal{N}_1$. Suppose that the following truncation error estimates hold:

$$\begin{aligned} &\Delta^h u(x_j, y_k) - S(x_j, y_k) = O(h^2) \text{ for all points } (x_j, y_k) \notin \mathcal{N}_1 \ , \\ &\Delta^h u(x_j, y_k) - S(x_j, y_k) - T^h_{j,k} = O(h) \text{ for all points } (x_j, y_k) \in \mathcal{N}_1 \ . \end{aligned} \tag{6.2}$$

Then

$$\begin{aligned} &U_{j,k} - u(x_j, y_k) = O(h^2) \text{ for all } (x_j, y_k) \in \bar{\Omega} \ , \\ &\nabla^h U_{j,k} - \nabla^h u(x_j, y_k) = O(h^2 \log(1/h)) \text{ for all } (x_j, y_k) \in \Omega \ . \end{aligned} \tag{6.3}$$

Remark 6.1. The theorem above is also valid in the three-dimensional setting.

Remark 6.2. Reference [2] gives specific regularity conditions for the data f, $S^\pm$, a, b that are sufficient to guarantee that the solution u satisfies the regularity conditions of Theorem 6.1.

Our convergence proof will require a number of auxiliary quantities, which we now define. Let μ be a C^∞ function $\mu : \mathbb{R} \mapsto [0,1]$ such that $\mu(z) = 1$ for $|z| < \alpha/4$ and $\mu(r) = 0$ for $|z| \geq \alpha/2$. Let $\rho(x,y) = \mu(\phi(x,y))$. Recall that we are assuming that $S^\pm$ is defined and smooth on $\Omega_\pm \cup B_\alpha$. We extend $S^\pm$ to be smooth and defined on all of $\bar{\Omega}$:

$$\tilde{S}^+ = \begin{cases} S^- + \rho\,(S^+ - S^-)\,, & \phi \leq 0, \\ S^+, & \phi \geq 0, \end{cases} \quad \tilde{S}^- = \begin{cases} S^-, & \phi \leq 0, \\ S^+ + \rho\,(S^- - S^+)\,, & \phi \geq 0. \end{cases} \tag{6.4}$$

Let $\sigma = \tilde{S}^+ - \tilde{S}^-$. Clearly σ vanishes outside of the narrow band $B_{\alpha/2}$. Finally, with

$$\tilde{b} = \rho\hat{b}, \quad \gamma = \Delta\tilde{b} - \sigma, \tag{6.5}$$

we define the following three quantities:

$$A(x,y) = \frac{\gamma}{|\nabla\phi|^2}, \quad B(x,y) = \frac{\gamma\Delta\phi}{|\nabla\phi|^4}, \quad C(x,y) = \frac{\gamma\,(\Delta\phi)^2}{|\nabla\phi|^4}. \tag{6.6}$$

Each of A, B, and C is smooth in $\bar{\Omega}$ and vanishes outside of $B_{\alpha/2}$.

For $\nu \in (0,1)$, let $C^\nu(\Omega)$ denote the class of Hölder continuous functions on Ω with exponent ν, and for a positive integer m, let $C^{m+\nu}(\Omega)$ denote the class of functions in $C^m(\Omega)$ all of whose mth partial derivatives are in $C^\nu(\Omega)$. In the proof of the lemma below we will use the following fact [6]: If w is a solution of the Poisson equation $\Delta w = R$ in Ω, and $R \in C^\nu(\Omega)$, then $w \in C^{2+\nu}(\bar{\Omega}_1)$ for any closed subdomain $\bar{\Omega}_1 \subset \Omega$.

Lemma 6.1. *In addition to the regularity assumptions stated in Theorem 6.1, assume that for some $\nu \in (0,1)$,*

$$\phi \in C^{4+\nu}(\bar{B}_\alpha), \quad S^\pm \in C^{2+\nu}(\bar{\Omega}_\pm \cup \bar{B}_\alpha), \quad a \in C^{4+\nu}(\bar{B}_\alpha), \quad b \in C^{5+\nu}(\bar{B}_\alpha). \tag{6.7}$$

Let

$$\begin{aligned} w &= u - \tilde{b}H + AJ, \\ R &= J\Delta A + 2\nabla J \cdot \nabla A + B\Delta K - CJ + \tilde{S}^-. \end{aligned} \tag{6.8}$$

Then $w \in C^2(\bar{\Omega})$ *and*

$$\begin{cases} \Delta w = R, & (x,y) \in \Omega, \\ w = f, & (x,y) \in \partial\Omega. \end{cases} \tag{6.9}$$

Thus, $\Delta w = R$ *in the classical sense for all points in* Ω*, including points lying on* Γ.

Proof. First, it is clear that $w = f$ on $\partial\Omega$ since

$$\tilde{b}H\Big|_{\partial\Omega} = AJ\Big|_{\partial\Omega} = 0. \tag{6.10}$$

Next we verify that the PDE $\Delta w = R$ is satisified in each of Ω_-, Ω_+. For $(x,y) \in \Omega_-$, $R = S^-$, and $w = u$, so clearly $\Delta w = R$ for $(x,y) \in \Omega_-$. For $(x,y) \in \Omega_+$, $w = u - \tilde{b} + AJ$, so

$$\begin{aligned} \Delta w &= \Delta u - \Delta\tilde{b} + \Delta(AJ) \\ &= S^+ - \Delta\tilde{b} + A\Delta J + J\Delta A + 2\nabla J \cdot \nabla A \\ &= J\Delta A + 2\nabla J \cdot \nabla A + \tilde{S}^+ - \Delta\tilde{b} + A\Delta J. \end{aligned} \tag{6.11}$$

We use the relationship

$$\tilde{S}^+ - \Delta\tilde{b} = \sigma - \Delta\tilde{b} + \tilde{S}^- = -\gamma + \tilde{S}^-, \tag{6.12}$$

and the definition of A to write

$$\begin{aligned} \tilde{S}^+ - \Delta\tilde{b} + A\Delta J &= -\gamma + \tilde{S}^- + \frac{\gamma\Delta J}{|\nabla\phi|^2} \\ &= -\gamma\left(1 - \frac{\Delta J}{|\nabla\phi|^2}\right) + \tilde{S}^- \\ &= -\gamma\left(-\frac{\Delta\phi\Delta K}{|\nabla\phi|^4} + \frac{(\Delta\phi)^2 J}{|\nabla\phi|^4}\right) + \tilde{S}^- \\ &= B\Delta K - CJ + \tilde{S}^-. \end{aligned} \tag{6.13}$$

Here we have used the identity

$$\frac{\Delta J}{|\nabla\phi|^2} - \frac{\Delta\phi\Delta K}{|\nabla\phi|^4} + \frac{(\Delta\phi)^2 J}{|\nabla\phi|^4} = 1 \text{ for } (x,y) \in \Omega_+. \tag{6.14}$$

Plugging (6.13) into (6.11), we have verified that $\Delta w = R$ for $(x,y) = \Omega_+$.

We claim that $[w] = 0$ and $[\partial w/\partial n] = 0$. In fact,

$$\begin{aligned} [w] &= [u] - [\tilde{b}] + [AJ] = b - b + 0 = 0, \\ [\partial w/\partial n] &= [\partial u/\partial n] - [\partial\tilde{b}/\partial n] + [\partial(AJ)/\partial n] = a - a + 0 = 0. \end{aligned} \tag{6.15}$$

Using $[w] = 0$, along with the fact that w is differentiable in $\Omega_\pm$, we find that w is weakly differentiable in Ω, and in fact $w \in H^1(\Omega)$. By a calculation similar to the derivation of (2.4) (with w replacing u), and using the fact that $[\partial w/\partial n] = 0$, we get

$$\int_\Omega \nabla w \cdot \nabla\psi \, dA = -\int_\Omega \psi R \, dA. \tag{6.16}$$

Thus, w is a weak solution of $\Delta w = R$ in Ω, and in fact must be the unique H^1 solution of (6.9).

As a result of the regularity assumptions (6.7), $R \in C^\nu(\bar{\Omega})$. Recall that $\Omega = (-L, L) \times (-L, L)$. For $\epsilon > 0$, let $\Omega^\epsilon = (-L+\epsilon, L-\epsilon) \times (-L+\epsilon, L-\epsilon)$, and assume that ϵ is small enough that $\Omega_+ \subset \Omega^{2\epsilon}$. According to the fact stated before the lemma, $w \in C^{2+\nu}(\bar{\Omega}^\epsilon)$. Since $u \in C^4(\bar{\Omega}_-)$, and $\tilde{b}$ and AJ are smooth in Ω_-, we have $w \in C^2(\bar{\Omega}_-)$. From $w \in C^{2+\nu}(\bar{\Omega}^\epsilon) \subset C^2(\bar{\Omega}^\epsilon)$ and $w \in C^2(\bar{\Omega}_-)$, it follows that $w \in C^2(\bar{\Omega})$.

$\square$

Remark 6.3. The cutoff function ρ used in the proof above is only an analytical tool. It is not used in our algorithms.

Remark 6.4. Lemma 6.1 indicates that as an alternative to the methods (5.5) and (5.7), one could simply use a standard (no singular source terms) finite difference scheme for the problem (6.9), which has a continuous right hand side. A drawback to this approach is that one must then construct the cutoff function ρ, making the scheme more complicated.

Li et. al. [15, 16], in connection with the Immersed Interface Method, have already observed that problems like (1.1) and (1.7) can be reduced to ones with a continuous source term. Moreover, they propose actually solving the reduced problem that they derive. With their setup this is a practical approach; their formulation does not include any cutoff functions.

Theorem 6.2. *Assume that the data and solution satisfy the regularity assumptions of Lemma 6.1. For the approximations generated by the discretization* (5.7)*, we have second order accuracy for the approximate solution, and nearly second order accuracy for the approximate gradient:*

$$\begin{aligned} &U_{j,k} - u(x_j, y_k) = O(h^2) \text{ for all } (x_j, y_k) \in \bar{\Omega} \ , \\ &\nabla^h U_{j,k} - \nabla^h u(x_j, y_k) = O(h^2 \log(1/h)) \text{ for all } (x_j, y_k) \in \Omega \ . \end{aligned} \tag{6.17}$$

Proof. Our goal is to apply Theorem 6.1. The quantity in (5.7) corresponding to $T^h_{j,k}$ is

$$T^h_{j,k} = \Delta^h \left(\hat{b}^h_{j,k} H_{j,k} \right) - H^{\mathrm{F2c}}_{j,k} \Delta^h \hat{b}^h_{j,k} + H^{\mathrm{F2c}}_{j,k} S^+_{j,k} + (1 - H^{\mathrm{F2c}}_{j,k}) S^-_{j,k} - S_{j,k}. \tag{6.18}$$

Recalling Remark 5.1, the quantity $\Delta^h \left(\hat{b}^h_{j,k} H_{j,k} \right) - H^{\mathrm{F2c}}_{j,k} \Delta^h \hat{b}^h_{j,k} = 0$ for $(x_j, y_k) \notin \mathcal{N}_1$. Also, since $H^{\mathrm{F2c}}_{j,k} = H_{j,k}$ for $(x_j, y_k) \notin \mathcal{N}_1$, it is clear that $H^{\mathrm{F2c}}_{j,k} S^+_{j,k} + (1 - H^{\mathrm{F2c}}_{j,k}) S^-_{j,k} - S_{j,k} = 0$ for $(x_j, y_k) \notin \mathcal{N}_1$. Thus $T^h_{j,k} = 0$ for $(x_j, y_k) \notin \mathcal{N}_1$, as required by Theorem 6.1.

Next we verify the truncation error requirements (6.2) of Theorem 6.1. For $(x_j, y_k) \notin \mathcal{N}_1$,

$$\Delta^h u(x_j, y_k) - S(x_j, y_k) = \Delta u(x_j, y_k) + O(h^2) - S(x_j, y_k) = O(h^2). \tag{6.19}$$

This follows from the fact that Δ^h is a second order approximation to Δ, that u satisfies $\Delta u = S$ in $\Omega_\pm$, and that $u \in C^4(\bar{\Omega}_\pm)$. To deal with the case where $(x_j, y_k) \in \mathcal{N}_1$ we start by defining

$$A^h_{j,k} = \frac{\gamma_{j,k}}{|\nabla^h \phi_{j,k}|^2}, \quad B^h_{j,k} = \frac{\gamma_{j,k} \Delta^h \phi_{j,k}}{|\nabla^h \phi_{j,k}|^4}, \quad C^h_{j,k} = \frac{\gamma_{j,k} \left(\Delta^h \phi_{j,k}\right)^2}{|\nabla^h \phi_{j,k}|^4}. \tag{6.20}$$

Next, we use the definition (6.18) of $T^h_{j,k}$ and $w(x_j, y_k) = w_{j,k}$ of Lemma 6.1 to compute:

$$\begin{aligned}\Delta^h u_{j,k} - T^h_{j,k} - S_{j,k} &= \Delta^h \left(w_{j,k} + \tilde{b}_{j,k} H_{j,k} - A_{j,k} J_{j,k} \right) \\ - \Delta^h \left(\hat{b}_{j,k} H_{j,k} \right) &+ H^{\mathrm{F2c}}_{j,k} \Delta^h \hat{b}_{j,k} - H^{\mathrm{F2c}}_{j,k} S^+_{j,k} - (1 - H^{\mathrm{F2c}}_{j,k}) S^-_{j,k}.\end{aligned} \tag{6.21}$$

For h sufficiently small, $\mathcal{N}_1 \subset B_{\alpha/4}$, implying that $\rho(x_j, y_k) = 1$ for $(x_j, y_k) \in \mathcal{N}_1$. As a further consequence, $\hat{b}_{j,k} = \tilde{b}_{j,k}$ and $S^\pm_{j,k} = \tilde{S}^\pm_{j,k}$ for $(x_j, y_k) \in \mathcal{N}_1$. Taking this into account, we continue our calculation of $\Delta^h u_{j,k} - T^h_{j,k} - S_{j,k}$:

$$\begin{aligned}&\Delta^h u_{j,k} - T^h_{j,k} - S_{j,k} \\ &= \Delta^h w_{j,k} + \Delta^h \left(\tilde{b}_{j,k} H_{j,k} \right) - \Delta^h \left(A_{j,k} J_{j,k} \right) \\ &- \Delta^h \left(\tilde{b}^h_{j,k} H_{j,k} \right) + H^{\mathrm{F2c}}_{j,k} \Delta^h \tilde{b}^h_{j,k} - H^{\mathrm{F2c}}_{j,k} \tilde{S}^+_{j,k} - (1 - H^{\mathrm{F2c}}_{j,k}) \tilde{S}^-_{j,k} \\ &= \Delta^h w_{j,k} - \Delta^h \left(A_{j,k} J_{j,k} \right) + H^{\mathrm{F2c}}_{j,k} \Delta^h \tilde{b}^h_{j,k} - H^{\mathrm{F2c}}_{j,k} \sigma_{j,k} - \tilde{S}^-_{j,k}\end{aligned} \tag{6.22}$$

Recalling that $\gamma = \Delta \tilde{b} - \sigma$, we obtain

$$\begin{aligned}&\Delta^h u_{j,k} - T^h_{j,k} - S_{j,k} \\ &= \Delta^h w_{j,k} - \Delta^h \left(A_{j,k} J_{j,k} \right) + \gamma_{j,k} H^{\mathrm{F2c}}_{j,k} - \tilde{S}^-_{j,k}.\end{aligned} \tag{6.23}$$

Next, we substitute the identity

$$\gamma_{j,k} H^{\mathrm{F2c}}_{j,k} = A^h_{j,k} \Delta^h J_{j,k} - B^h_{j,k} \Delta^h K_{j,k} + C^h_{j,k} J_{j,k}, \tag{6.24}$$

obtaining

$$\begin{aligned}&\Delta^h u_{j,k} - T^h_{j,k} - S_{j,k} \\ &= \Delta^h w_{j,k} - \Delta^h \left(A_{j,k} J_{j,k} \right) + A^h_{j,k} \Delta^h J_{j,k} - B^h_{j,k} \Delta^h K_{j,k} + C^h_{j,k} J_{j,k} - \tilde{S}^-_{j,k} \\ &= \Delta^h w_{j,k} - J_{j,k} \Delta^h A^h_{j,k} - 2 \langle A^h, J \rangle_{j,k} - B^h_{j,k} \Delta^h K_{j,k} + C^h_{j,k} J_{j,k} - \tilde{S}^-_{j,k} \\ &- \Delta^h \left(A_{j,k} J_{j,k} \right) + \Delta^h \left(A^h_{j,k} J_{j,k} \right) \\ &= \underbrace{\Delta w_{j,k} - J_{j,k} \Delta A_{j,k} - 2 \nabla J_{j,k} \cdot \nabla A_{j,k} - B_{j,k} \Delta K_{j,k} + C_{j,k} J_{j,k} - \tilde{S}^-_{j,k}}_{=0 \text{ by Lemma 6.1}} \\ &+ \underbrace{\Delta^h w_{j,k} - \Delta w_{j,k}}_{\mathcal{E}_1} - \underbrace{\left(J_{j,k} \Delta^h A^h_{j,k} - J_{j,k} \Delta A_{j,k} \right)}_{\mathcal{E}_2} - 2 \underbrace{\left(\langle A^h, J \rangle_{j,k} - \nabla J_{j,k} \cdot \nabla A_{j,k} \right)}_{\mathcal{E}_3} \\ &- \underbrace{\left(B^h_{j,k} \Delta^h K_{j,k} - B_{j,k} \Delta K_{j,k} \right)}_{\mathcal{E}_4} + \underbrace{C^h_{j,k} J_{j,k} - C_{j,k} J_{j,k}}_{\mathcal{E}_5} \\ &- \underbrace{\left(\Delta^h \left(A_{j,k} J_{j,k} \right) - \Delta^h \left(A^h_{j,k} J_{j,k} \right) \right)}_{\mathcal{E}_6}.\end{aligned} \tag{6.25}$$

We must verify that each of $\mathcal{E}_1$ through $\mathcal{E}_6$ is $O(h)$, or smaller, for $(x_j, y_k) \in \mathcal{N}_1$. First, for $\mathcal{E}_1$, using the fact (from Lemma 6.1) that $w \in C^2(\bar{\Omega})$, $\Delta^h w_{j,k} = \Delta w_{j,k} + O(h)$, and so $\mathcal{E}_1 = O(h)$. For $\mathcal{E}_2$ and $\mathcal{E}_5$,

$$\mathcal{E}_2 = \left(\Delta^h A^h_{j,k} - \Delta A_{j,k} \right) J_{j,k}, \quad \mathcal{E}_5 = \left(C^h_{j,k} - C_{j,k} \right) J_{j,k}. \tag{6.26}$$

Since $J_{j,k} = O(h^2)$ for $(x_j, y_k) \in \mathcal{N}_1$, and the quantities in the parentheses are bounded (or smaller), we have $\mathcal{E}_2 = O(h^2)$ and $\mathcal{E}_5 = O(h^2)$. For $\mathcal{E}_3$, note that

$$\nabla J_{j,k} \cdot \nabla A_{j,k} = I_{j,k} \nabla \phi_{j,k} \cdot \nabla A_{j,k}, \tag{6.27}$$

and this quantity is $O(h)$ for $(x_j, y_k) \in \mathcal{N}_1$ due to the factor of $I_{j,k}$. Similarly, using (3.6) and the mean value theorem, one finds that the quantity $\langle A^h, J\rangle_{j,k} = O(h)$ for $(x_j, y_k) \in \mathcal{N}_1$, and thus $\mathcal{E}_3 = O(h)$. Concerning $\mathcal{E}_4$,

$$\Delta K_{j,k} = I_{j,k} \left|\nabla \phi\right|^2 + \frac{1}{2} J_{j,k} \Delta \phi_{j,k} = O(h) \text{ for } (x_j, y_k) \in \mathcal{N}_1. \tag{6.28}$$

Similarly, an application of the mean value theorem gives $\Delta^h K_{j,k} = O(h)$ for $(x_j, y_k) \in \mathcal{N}_1$, and since each of the quantities $B^h_{j,k}$ and $B_{j,k}$ is bounded, we have $\mathcal{E}_4 = O(h)$. Finally, to deal with $\mathcal{E}_6$, we write

$$\mathcal{E}_6 = \left(A_{j,k} - A^h_{j,k}\right) \Delta^h J_{j,k} + J_{j,k} \left(\Delta^h A_{j,k} - \Delta^h A^h_{j,k}\right) + 2\langle A - A^h, J\rangle_{j,k}. \tag{6.29}$$

The first term is $O(h^2)$ since $A_{j,k} - A^h_{j,k} = O(h^2)$ and $\Delta^h J_{j,k}$ is bounded. The second term is $O(h^4)$ since $J_{j,k}$ is $O(h^2)$ for $(x_j, y_k) \in \mathcal{N}_1$ and $\Delta^h A_{j,k} - \Delta^h A^h_{j,k} = O(h^2)$. The third term can be dealt with in much the same way as $\mathcal{E}_3$, and is $O(h^2)$. $\square$

7. **Numerical examples.**

Below we present the results of numerical experiments. Except for Example 6, the data of all of the examples satisfy the smoothness hypotheses of Theorem 6.2. In each case the computational domain is the square $\Omega = [-1, 1] \times [-1, 1]$. We use the notation

$$\begin{aligned} \left|E_u\right|_\infty &= \max_{(x_j, y_k) \in \bar{\Omega}} \left|u(x_j, y_k) - U_{j,k}\right|, \\ \left|E_{\nabla^h u}\right|_\infty &= \max_{(x_j, y_k) \in \Omega} \left|\nabla^h u(x_j, y_k) - \nabla^h U_{j,k}\right|. \end{aligned} \tag{7.1}$$

When using the discretization (5.7), we approximated the quantity $\hat{b}_{j,k}$ using

$$\hat{b}^h_{j,k} = b_{j,k} + \left(\frac{\partial b}{\partial n}(x_j, y_k) - a_{j,k}\right) \phi_{j,k} / \left|\nabla^{h,4} \phi_{j,k}\right|. \tag{7.2}$$

Here $\nabla^{h,4}$ denotes the fourth order discrete gradient operator:

$$\nabla^{h,4} \phi_{j,k} = (4/3) \nabla^h \phi_{j,k} - (1/3) \nabla^{2h} \phi_{j,k}. \tag{7.3}$$

In our numerical experiments we measure the observed rate of convergence by repeatedly halving the mesh size. With $E(h)$ denoting the error using mesh size h, the empirical rate of convergence is then

$$\text{Rate} = \frac{\log(E(h)/E(\frac{1}{2}h))}{\log(2)}. \tag{7.4}$$

Concerning the rate of convergence predicted for the discrete gradient produced by the discretization (5.7), note that if $E(h) = h^2 \log(1/h)$, then

$$\lim_{h \to 0} \frac{\log(E(h)/E(\frac{1}{2}h))}{\log(2)} = 2, \tag{7.5}$$

so that in our examples, this rate of convergence may be indistinguishable from a second order rate of convergence.

TABLE 1. Example 1.

h	Jump Data a_1, using (5.5) $\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate	Jump Data a_1, using (5.7) $\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate
.05	5.504e−4		3.077e−3		1.105e−3		2.878e−3	
.05/2	1.420e−4	1.95	1.003e−3	1.62	2.832e−4	1.96	8.940e−4	1.69
.05/4	3.566e−5	1.99	2.898e−4	1.79	7.170e−5	1.98	2.514e−4	1.83
.05/8	9.258e−6	1.94	7.908e−5	1.87	1.804e−5	1.99	6.687e−5	1.91
	Jump Data a_2, using (5.5)				Jump Data a_2, using (5.7)			
h	$\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate	$\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate
.05	7.947e−4		8.470e−3		1.098e−3		2.990e−3	
.05/2	2.168e−4	1.87	4.633e−3	0.87	2.508e−4	2.13	7.271e−4	2.04
.05/4	5.710e−5	1.93	2.457e−3	0.91	5.988e−5	2.07	1.845e−4	1.98
.05/8	1.507e−5	1.92	1.308e−3	0.91	1.461e−5	2.04	4.719e−5	1.97

Example 1. This example is borrowed from [10]; see also [7, 17, 24, 32, 37]. The problem is (1.1) with $S = 0$. Ω_+ is the disk $r = 1/2$ and $r = \sqrt{x^2 + y^2}$. The data is chosen so that the solution of the problem is

$$u(x, y) = \begin{cases} 1, & r < 1/2, \\ 1 + \log(2r), & r \geq 1/2. \end{cases} \tag{7.6}$$

From (7.6), one finds that $[u] = 0$, $[\partial u/\partial n] = -2$. For a level set function, we use $\phi(x, y) = 1/2 - r$, which is a signed distance function. The boundary data f is defined by (7.6), and for the jump data a and b, we tested with two versions; denoted a_1, b_1 and a_2, b_2:

$$\begin{aligned} &a_1(x, y) = -2, \quad b_1(x, y) = 0, \\ &a_2(x, y) = -2\exp(r - 1/2), \quad b_2(x, y) = 0. \end{aligned} \tag{7.7}$$

Example 2. For this example $S = 0$ again. This time the level set function is $\phi(x, y) = \left(x^2 - y^2\right) - (.4 + \log(r))$, which is not a signed distance function. The data is chosen so that the solution of the problem is

$$u(x, y) = \begin{cases} .4 + \log(2r) =: u^-(x, y), & \phi(x, y) \leq 0, \\ x^2 - y^2 =: u^+(x, y), & \phi(x, y) > 0. \end{cases} \tag{7.8}$$

From (7.8), one finds that $[u] = 0$, and since $\phi(x, y) = u^+(x, y) - u^-(x, y)$, $[\partial u/\partial n] = -\left|\nabla\phi\right|$. We tested with two versions of the derivative jump data.

$$\begin{aligned} &a_1(x, y) = -\left|\nabla\phi\right|, \quad b_1(x, y) = 0, \\ &a_2(x, y) = (1 - xy\phi(x, y))a_1(x, y), \quad b_2(x, y) = 0. \end{aligned} \tag{7.9}$$

Example 3. This example is borrowed from [10]; see also [17, 37]. In this example, $S = 0$, and there is a jump in both u and $\partial u/\partial n$. With $r = \sqrt{x^2 + y^2}$, Ω_+ is the open disk $r < 1/2$. The solution of the problem is

$$u(x, y) = \begin{cases} e^x\cos(y), & r < 1/2, \\ 0, & r \geq 1/2, \end{cases} \tag{7.10}$$

which implies that $[u] = e^x\cos(y)$, $[\partial u/\partial n] = e^x(x\cos(y) - y\sin(y))$. We ran two tests. In the first one, we used the signed distance function $\phi_1 = 1/2 - r$ for the

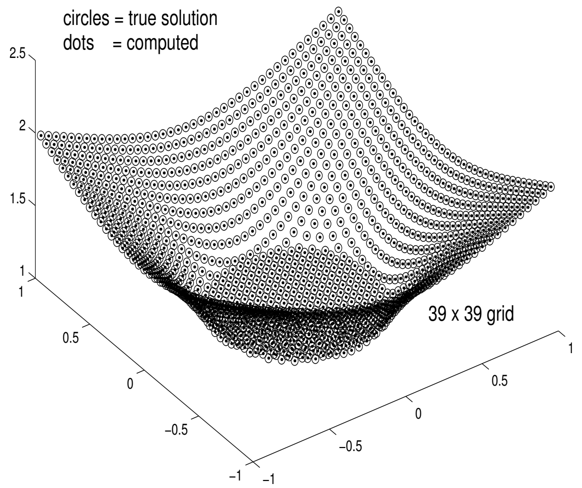

FIGURE 2. Example 1. Jump in $\partial u/\partial n$.

TABLE 2. Example 2.

h	Jump Data a_1, using (5.5) $\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate	Jump Data a_1, using (5.7) $\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate
.05	6.787e−4		9.076e−3		8.030e−4		7.559e−3	
.05/2	1.532e−4	2.15	3.103e−3	1.55	1.856e−4	2.11	1.958e−3	1.95
.05/4	3.730e−5	2.04	9.368e−4	1.73	4.876e−5	2.01	4.876e−4	2.01
.05/8	9.520e−6	1.97	2.973e−4	1.66	1.154e−5	2.00	1.219e−4	2.00
	Jump Data a_2, using (5.5)				Jump Data a_2, using (5.7)			
h	$\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate	$\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate
.05	6.787e−4		9.102e−3		1.111e−3		9.204e−3	
.05/2	1.532e−4	2.15	3.163e−3	1.53	2.530e−4	2.13	2.367e−3	1.96
.05/4	3.744e−5	2.03	1.061e−3	1.58	6.230e−5	2.02	6.019e−4	1.98
.05/8	9.640e−6	1.96	3.963e−4	1.42	1.550e−5	2.01	1.521e−4	1.98

level set function. The jump data for this test is

$$\begin{aligned} &a_1(x,y) = 2e^{\xi}\left(\xi\cos(\eta) - \eta\sin(\eta)\right), \quad b_1(x,y) = e^{\xi}\cos(\eta), \quad \frac{\partial}{\partial n}b_1(x,y) = 0, \\ &\xi = \frac{1}{2}\cos(\theta), \quad \eta = \frac{1}{2}\sin(\theta). \end{aligned} \tag{7.11}$$

Here θ is the polar angle defined by $x = r\cos(\theta)$, $y = r\sin(\theta)$.

In the second test, we used $\phi_2 = 1/4 - r^2$, and

$$\begin{aligned} &a_2(x,y) = 2e^{x}(x\cos(y) - y\sin(y)), \quad b_2(x,y) = e^{x}\cos(y), \\ &\frac{\partial}{\partial n}b_2(x,y) = \left(xe^{x}\cos(y) - ye^{x}\sin(y)\right)/r. \end{aligned} \tag{7.12}$$

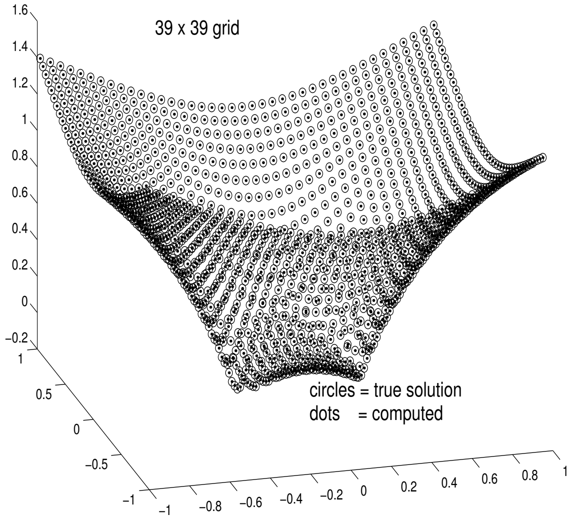

FIGURE 3. Example 2. Jump in $\partial u/\partial n$.

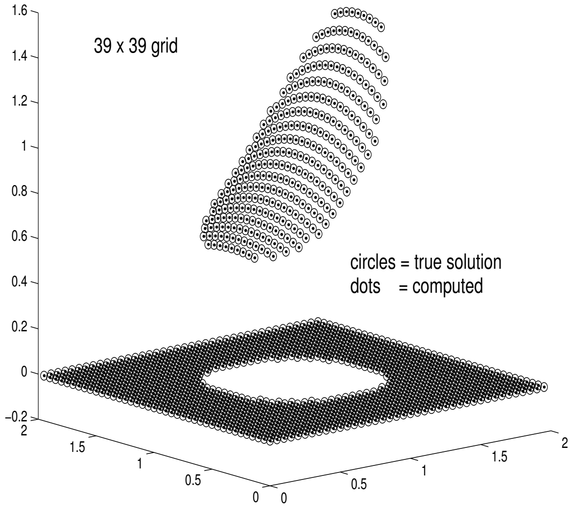

FIGURE 4. Example 3. Jump in u and $\partial u/\partial n$.

Example 4. This example is borrowed from [10]; see also [17]. We used $\phi(x, y) = 1/2 - r$ for the level set function, which is a signed distance function. Again $S = 0$,

TABLE 3. Example 3.

h	Jump data ϕ_1, a_1, b_1, using (5.5). $\|E_u\|_\infty$	Rate	$\|E_{\nabla^h u}\|_\infty$	Rate	Jump data ϕ_1, a_1, b_1, using (5.7). $\|E_u\|_\infty$	Rate	$\|E_{\nabla^h u}\|_\infty$	Rate
.05	1.127e−3		7.385e−3		2.928e−4		1.129e−3	
.05/2	3.345e−4	1.76	4.247e−3	0.80	7.168e−5	2.03	2.993e−4	1.92
.05/4	9.043e−5	1.88	2.337e−3	0.86	1.767e−5	2.02	7.742e−5	1.95
.05/8	2.279e−5	1.99	1.274e−3	0.88	3.378e−6	2.01	1.987e−5	1.96
	Jump data ϕ_2, a_2, b_2, using (5.5).				Jump data ϕ_2, a_2, b_2, using (5.7).			
h	$\|E_u\|_\infty$	Rate	$\|E_{\nabla^h u}\|_\infty$	Rate	$\|E_u\|_\infty$	Rate	$\|E_{\nabla^h u}\|_\infty$	Rate
.05	6.174e−4		1.040e−2		1.516e−3		4.342e−3	
.05/2	1.950e−4	1.66	6.329e−3	0.72	3.657e−4	2.05	1.146e−3	1.92
.05/4	5.669e−5	1.78	3.707e−3	0.77	8.950e−5	2.03	2.970e−4	1.95
.05/8	1.568e−5	1.85	2.059e−3	0.85	2.213e−5	2.02	7.584e−5	1.97

TABLE 4. Example 4.

h	Jump data ϕ_1, a_1, b_1, using (5.5). $\|E_u\|_\infty$	Rate	$\|E_{\nabla^h u}\|_\infty$	Rate	Jump data ϕ_1, a_1, b_1, using (5.7). $\|E_u\|_\infty$	Rate	$\|E_{\nabla^h u}\|_\infty$	Rate
.05	9.565e−4		7.169e−3		2.646e−4		1.251e−3	
.05/2	2.844e−4	1.75	4.082e−3	0.81	6.393e−5	2.05	3.238e−4	1.95
.05/4	7.725e−5	1.88	2.243e−3	0.86	1.566e−5	2.02	8.518e−5	1.93
.05/8	1.961e−5	1.98	1.192e−3	0.91	3.878e−6	2.02	2.139e−5	1.99
	Jump data ϕ_2, a_2, b_2, using (5.5).				Jump data ϕ_2, a_2, b_2, using (5.7).			
h	$\|E_u\|_\infty$	Rate	$\|E_{\nabla^h u}\|_\infty$	Rate	$\|E_u\|_\infty$	Rate	$\|E_{\nabla^h u}\|_\infty$	Rate
.05	4.2e−16		3.9e−15		4.2e−16		3.9e−15	
.05/2	2.6e−15	N/A	3.1e−14	N/A	2.6e−15	N/A	3.1e−14	N/A
.05/4	2.8e−14	N/A	1.5e−13	N/A	2.8e−14	N/A	1.5e−13	N/A
.05/8	7.7e−14	N/A	5.6e−13	N/A	7.7e−14	N/A	5.6e−13	N/A

and this time the solution is

$$u(x,y) = \begin{cases} x^2 - y^2, & r < 1/2, \\ 0, & r \geq 1/2, \end{cases} \tag{7.13}$$

which yields $[u] = x^2 - y^2$, $[\partial u/\partial n] = 4(x^2 - y^2)$. As in the previous examples, we tested two versions. For the first version, we express the jump data using

$$\begin{aligned} &a_1(x,y) = 4(\xi^2 - \eta^2), \quad b_1(x,y) = \xi^2 - \eta^2, \quad \frac{\partial}{\partial n} b_1(x,y) = 0, \\ &\xi = \frac{1}{2}\cos(\theta), \quad \eta = \frac{1}{2}\sin(\theta). \end{aligned} \tag{7.14}$$

For the second version, we used $\phi_2(x,y) = 1/4 - r^2$, and

$$a_2(x,y) = 4(x^2 - y^2), \quad b_2(x,y) = x^2 - y^2, \quad \frac{\partial}{\partial n} b_2(x,y) = 4(x^2 - y^2). \tag{7.15}$$

With this data, the computed solution is exact, modulo rounding error. Also, although we do not display the results, we also get essentially the exact solution if we replace ϕ_2 by the signed distance function ϕ_1.

Example 5. This example comes from [16]. In polar coordinates (r, θ), the level set function is

$$\phi(r,\theta) = 0.5 + 0.1\sin(4\theta + \pi) - r, \quad \theta \in [0, 2\pi]. \tag{7.16}$$

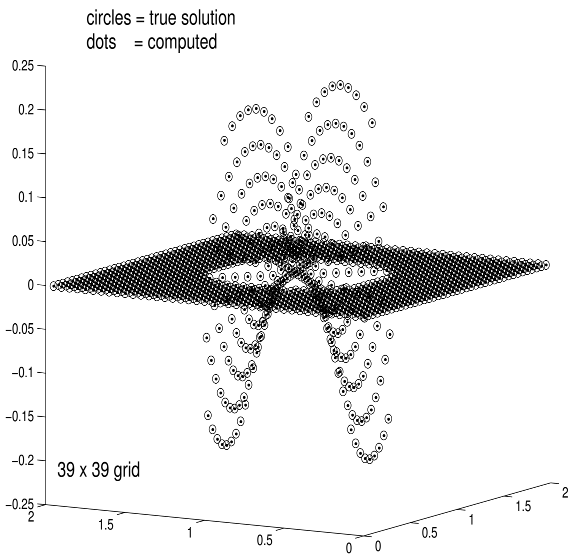

FIGURE 5. Example 4. Jump in u and $\partial u/\partial n$.

The solution is

$$u(r,\theta) = \begin{cases} r^4 =: u^-(r,\theta), & \phi \le 0, \\ r^2\sin(\theta) =: u^+(r,\theta), & \phi > 0. \end{cases} \tag{7.17}$$

This problem has a nonzero source term, given by

$$S(r,\theta) = \begin{cases} 3\sin(\theta), & \phi \le 0, \\ 16r^2, & \phi > 0. \end{cases} \tag{7.18}$$

We emphasize that we are still using a Cartesian mesh (a polar mesh is used in [16]). As before, we tested with two sets of jump data. First, we used

$$b_1 = u^+ - u^-, \quad a_1 = \frac{\partial b_1}{\partial n}. \tag{7.19}$$

For our second set of data, we used

$$b_2 = (1 - x\phi)b_1, \quad a_2 = (1 - xy\sin(\phi))a_1. \tag{7.20}$$

Note that for this data, $a_2 \ne \partial b_2/\partial n$.

Example 6. This example is similar to Example 3, the difference being that the region Ω_+ is now a semicircle. The purpose is to test our algorithms in the case where Γ is Lipschitz continuous, but only piecewise smooth. Theorem 6.2 does not apply to this example. We represent the semicircle via the level set function

$$\phi(x,y) = \min\left(\phi_1(x,y), \phi_2(x,y)\right).$$

where $\phi_1(x,y) = 1/2 - \sqrt{x^2+y^2}$ and $\phi_2(x,y) = (y-x)/\sqrt{2}$. The solution of the problem is

$$u(x,y) = \begin{cases} 0, & \phi \le 0, \\ e^x\cos(y), & \phi > 0. \end{cases} \tag{7.21}$$

TABLE 5. Example 5.

h	Jump data a_1, b_1, using (5.5). $\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate	Jump data a_1, b_1, using (5.7). $\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate
.05	1.583e−3		2.488e−3		1.583e−3		2.488e−3	
.05/2	3.957e−4	2.00	6.368e−4	1.97	3.957e−4	2.00	6.368e−4	1.97
.05/4	9.893e−5	2.00	1.610e−4	1.98	9.893e−5	2.00	1.610e−4	1.98
.05/8	2.473e−5	2.00	4.050e−5	1.99	2.473e−5	2.00	4.050e−5	1.99
h	Jump data a_2, b_2, using (5.5). $\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate	Jump data a_2, b_2, using (5.7). $\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate
.05	1.675e−3		9.491e−3		2.581e−3		7.022e−3	
.05/2	4.130e−4	2.02	5.171e−3	0.88	6.471e−4	2.00	1.985e−3	1.83
.05/4	1.030e−4	2.00	2.178e−3	1.25	1.620e−4	2.00	5.411e−4	1.88
.05/8	2.589e−5	1.99	1.168e−3	0.90	4.088e−5	1.99	1.430e−4	1.92

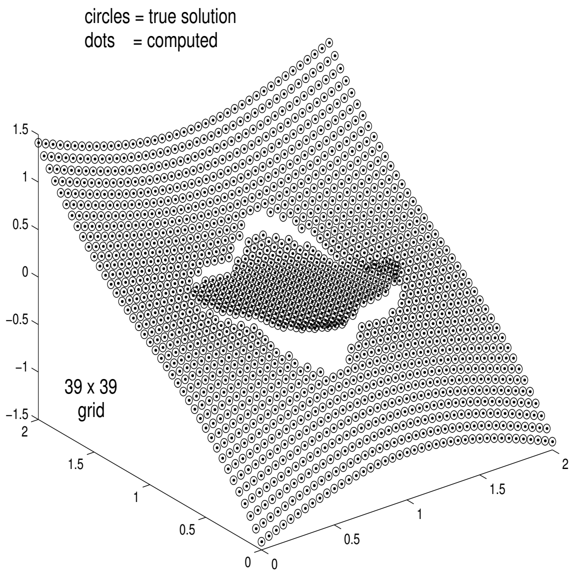

FIGURE 6. Example 5. Jump in u and $\partial u/\partial n$.

For jump data, we used

$$
\begin{aligned}
a(x,y) &= (1-\sin(\phi))\begin{cases}\nabla(e^x\cos(y))\cdot(-\nabla\phi_1), & |\phi_1|\le|\phi_2|,\\ \nabla(e^x\cos(y))\cdot(-\nabla\phi_2), & |\phi_2|>|\phi_2|,\end{cases}\\
b(x,y) &= (1-\phi)^q e^x\cos(y),\ q=2,\quad \frac{\partial}{\partial n}b(x,y)=\begin{cases}\nabla b\cdot(-\nabla\phi_1), & |\phi_1|\le|\phi_2|,\\ \nabla b\cdot(-\nabla\phi_2), & |\phi_2|>|\phi_2|.\end{cases}
\end{aligned}
\tag{7.22}
$$

We used $q=2$ in order to make the jump data $b\in C^1$. With $q=1$, b has jumps in its first partial derivatives, and the approximations seem not to converge.

TABLE 6. Example 6.

h	Using (5.5). $\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate	Using (5.7). $\lvert E_u\rvert_\infty$	Rate	$\lvert E_{\nabla^h u}\rvert_\infty$	Rate
.05	2.984e−3		1.471e−2		3.759e−3		2.419e−2	
.05/2	7.233e−4	2.04	6.702e−3	1.07	9.224e−4	2.03	9.605e−3	1.33
.05/4	1.542e−4	2.23	3.433e−3	0.97	1.866e−4	2.31	3.210e−3	1.56
.05/8	3.523e−5	2.13	1.330e−3	1.36	4.194e−5	2.15	1.262e−3	1.35

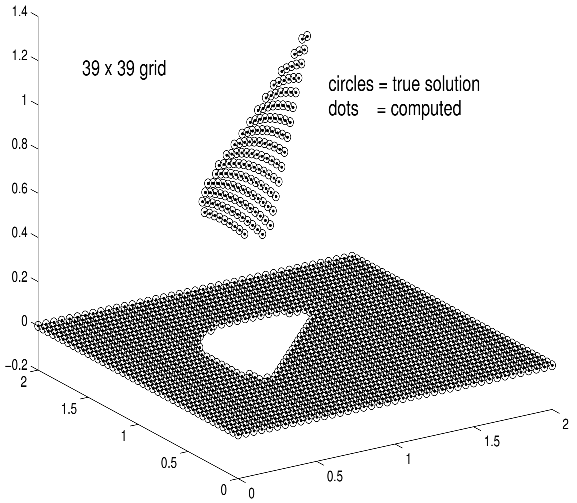

FIGURE 7. Example 6. Jump in u and $\partial u/\partial n$.

8. **Conclusion.**

We have devised two finite difference schemes, (5.5) and (5.7), for the Poisson interface problem (1.1). These schemes discretize certain level set formulations of the problem that contain singular source terms. We have discretized the singular source terms using finite difference methods. For one of our algorithms, we have used the result of [2] to prove second order accuracy for the approximate solution, and nearly second order accuracy for the discrete gradient.

For algorithm (5.7), Examples 1 through 5 in the previous section confirm the accuracy predicted by the theorem mentioned above. For algorithm (5.5), our numerical tests indicate second order accuracy for the approximate solution. For the discrete gradient produced by (5.5), the empirical convergence rate could be as low as first order (or slightly less), and as high as second order, depending on the problem. In some examples, algorithm (5.5) gives a somewhat more accurate approximation of the undifferentiated variable u than algorithm (5.7). This explains our interest in algorithm (5.5), despite that fact that we do not have a convergence proof, and that the accuracy of the discrete gradient is not optimal.

Although we did not display the results, we also computed the truncation error $U_{j,k} - S_{j,k} - T^h_{j,k}$ as a part of our numerical experiments. For (5.7), the truncation error for Examples 1 through 5 was $O(h)$, in agreement with Theorem 6.2. For (5.5) the truncation error was variable, and seemed to roughly correlate with the empirical rate of convergence of the discrete gradient. The relationship seems to be (roughly) that an $O(h^{p-1})$ truncation error implies an $O(h^p)$ (or possibly $O(h^p \log(1/h))$) error in the discrete gradient.

Finally, we used closed form expressions for $\partial b/\partial n$ to obtain the results shown in Tables 1 through 6, but we also tested our algorithms with a second order difference approximation to $\partial b/\partial n$, specifically

$$\left.\frac{\partial b}{\partial n}\right|_{(x_j,y_k)} \approx -\nabla^h b_{j,k} \cdot \frac{\nabla^h \phi_{j,k}}{|\nabla^h \phi_{j,k}|}. \tag{8.1}$$

This generally resulted in larger errors, but the observed rates of convergence indicated in Tables 1 through 5 (where Γ is smooth) were unchanged. For Example 6 (where Γ is only Lipschitz), we observed a decrease in the rates of convergence. Algorithm (5.5) gave better than first order convergence for u and slightly less than first order convergence for the discrete gradient. Algorithm (5.7) resulted in less than first order convergence for u, and the discrete gradient seemed not to converge in the L^∞ norm.

Acknowledgements.

Most of this work was done while I was on a sabbatical from my normal teaching duties at MiraCosta College during the spring semester of 2009.

I thank the Centre for Advanced Study in Norway for providing a stimulating atmosphere during my visit in January, 2009. This work resulted from a seminar that I presented there.

I thank Tom Beale for helping with my questions.

References

[1] T.D. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2003) 349–355.

[2] J.T. Beale and A. Layton, On the accuracy of finite difference methods for elliptic problems with interfaces, Commun. Appl. Math. Comput. Sci. 1 (2006) 91-119.

[3] J.T. Beale, A proof that a discrete delta function is second order accurate, J. Comput. Phys. 227 (2008) 2195–2197.

[4] R.P. Beyer, R.J. Leveque, Analysis of a one-dimensional model for the immersed boundary method, SIAM J. Numer. Anal. 29 (1992) 332–364.

[5] Y.C. Chang, T.Y. Hou, B. Merriman, S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, J. Comput. Phys. 124 (1996) 449–464.

[6] R. Courant, D. Hilbert, Methods of mathematical physics, Volume 2, Interscience Publishers, New York, 1962.

[7] B. Engquist, A.K. Tornberg, R. Tsai, Discretization of Dirac Delta Functions in Level Set Methods, J. Comput. Phys. 207 (2005) 28–51.

[8] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in flows (the Ghost Fluid Method), J. Comput. Phys. 152 (1999) 457–492.

[9] S. Hou, X.D. Liu, A numerical method for solving variable coefficient elliptic equations with interfaces, J. Comput. Phys., 202 (2005) 411–445.

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

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

[12] Z. Li, A note on immersed interface methods for three dimensional elliptic equations, Computers Math. Appl. 31 (1996) 9–17.
[13] Z. Li, Immersed interface methods for moving interface problems, Numerical algorithms 14 (1997) 269-293.
[14] Z. Li and M.C. Lai, The immersed interface method for the Navier Stokes equations with singular forces, J. Comput. Phys. 171 (2001) 822–842.
[15] Z. Li, W.C. Wang, I.L., Chern, M.C. Lai, New formulations for interface problems in polar coordinates, SIAM J. Sci. Comput. 25 (2003) 224–245.
[16] Z. Li, K. Ito, The immersed interface method, SIAM Frontiers in Applied Matheatics, Philadelphia, 2006.
[17] X.-D. Liu, R. Fedkiw and M. Kang, A boundary condition capturing method for Poissons equation on irregular domains, J. Comput. Phys. 160 (2000) 151-178.
[18] X.D. Liu, T.D. Sideris, Convergence of the ghost fluid method for elliptic equations with interfaces, Mathematics of Computation 72 (2003) 1731–1746.
[19] A. Mayo, The rapid evaluation of volume integrals of potential theory on general regions, J. Comput. Phys. 100 (1992) 236–245.
[20] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numeral Anal. 21 (1984) 285–299.
[21] 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, 101–118.
[22] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive cartesian grids, J. Comput. Phys. 225 (2007) 300–320.
[23] C. Min, F. Gibou, Geometric integration over irregular domains with application to level-set methods, J. Comput. Phys. 226 (2007) 1432–1443.
[24] C. Min, F. Gibou, Robust second-order accurate discretizations of the multi-dimensional Heaviside and Dirac delta functions, J. Comput. Phys. 227 (2008) 9686–9695.
[25] S. Osher, R. Fedkiw, Level set methods and dynamic implicit surfaces, Springer-Verlag, New York, 2003.
[26] S. Osher, J. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[27] D. Peng, B. Merriman, S. Osher, H. Zhao and M. Kang. A PDE-based fast local level set method, J. Comput. Phys. 155 (1999) 410–438.
[28] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[29] C.S. Peskin. The immersed boundary method. Acta Numerica 11 (2002) 479–511.
[30] J.A. Sethian, Level set methods and fast marching methods, Cambridge University Press, Cambridge, 1999.
[31] P. Smereka, The numerical approximation of a delta function with application to level set methods, J. Comput. Phys. 211 (2006) 77–90.
[32] A.K. Tornberg, B. Engquist. Numerical approximations of singular source terms in differential equations, J. Comput. Phys. 200 (2004) 462–488.
[33] A.K. Tornberg, B. Engquist, Regularization Techniques for Numerical Approximation of PDEs with Singularities, Journal of Scientific Computing 19 (2003) 527–552.
[34] J.D. Towers. Two methods for discretizing a delta function supported on a level set, J. Comput. Phys. 220 (2007) 915–931.
[35] J.D. Towers. A convergence rate theorem for finite difference approximations to delta functions, J. Comput. Phys. 227 (2008) 6591–6597.
[36] J.D. Towers. Finite difference methods for approximating Heaviside functions, J. Comput. Phys. 228 (2009) 3478–3489.
[37] B.L. Vaughan, B.G. Smith, D.L. Chopp, A comparison of the extended finite element method with the immersed interface method for elliptic equations with discontinuous coefficients and singular sources, Commun. Appl. Math. Comput. Sci. 1 (2006) 207-228.
[38] A. Wiegmann and K.P. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (2000) 827-862.
[39] X. Wen, High order numerical methods to a type of delta function integrals, J. Comput. Phys. 226 (2007) 1952–1967.
[40] X. Wen, High order numerical quadratures to one dimensional delta function integrals, SIAM J. Sci. Comput. 30 (2008) 1825–1846.