

Research Article

On Solving the Poisson Equation with Discontinuities on Irregular Interfaces: GFM and VIM

Ásdís Helgadóttir ¹, Arthur Guittet,² and Frédéric Gibou^{2,3}

¹Faculty of Industrial Engineering, Mechanical Engineering and Computer Science Department, University of Iceland, Reykjavik 107, Iceland

²Mechanical Engineering Department, University of California, Santa Barbara, CA 93106, USA

³Computer Science Department, University of California, Santa Barbara, CA 93106, USA

Correspondence should be addressed to Ásdís Helgadóttir; asdishe@hi.is

Received 15 May 2018; Accepted 3 October 2018; Published 17 October 2018

Guest Editor: Manoj Kumar

Copyright © 2018 Ásdís Helgadóttir et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We analyze the accuracy of two numerical methods for the variable coefficient Poisson equation with discontinuities at an irregular interface. Solving the Poisson equation with discontinuities at an irregular interface is an essential part of solving many physical phenomena such as multiphase flows with and without phase change, in heat transfer, in electrokinetics, and in the modeling of biomolecules' electrostatics. The first method, considered for the problem, is the widely known Ghost-Fluid Method (GFM) and the second method is the recently introduced Voronoi Interface Method (VIM). The VIM method uses Voronoi partitions near the interface to construct local configurations that enable the use of the Ghost-Fluid philosophy in one dimension. Both methods lead to symmetric positive definite linear systems. The Ghost-Fluid Method is generally first-order accurate, except in the case of both a constant discontinuity in the solution and a constant diffusion coefficient, while the Voronoi Interface Method is second-order accurate in the L^∞ -norm. Therefore, the Voronoi Interface Method generally outweighs the Ghost-Fluid Method except in special case of both a constant discontinuity in the solution and a constant diffusion coefficient, where the Ghost-Fluid Method performs better than the Voronoi Interface Method. The paper includes numerical examples displaying this fact clearly and its findings can be used to determine which approach to choose based on the properties of the real life problem in hand.

1. Introduction

The Poisson equations with discontinuities across irregular interfaces emerge in applications such as multiphase flows with and without phase change, in heat transfer, in electrokinetics, or in the modeling of biomolecules' electrostatics. Several numerical methods have been proposed to solve this system, each with their own advantages and disadvantages. One approach is in the context of Discontinuous Galerkin methods, an extension of the finite element method (FEM); e.g., see [1–12] and the references therein. Finite element methods lead to symmetric positive definite linear systems that can be efficiently solved with fast iterative solvers [13]. In addition, FEM-type approaches can derive and use *a priori* error estimates to refine the mesh where higher resolution is needed. However, FEM-type methods rely on the quality of

the underlying mesh, which is often difficult to obtain in cases where the irregular domain undergoes large deformations. In this case, it is challenging to generate a mesh with elements that pass a quality measure needed to ensure accurate solutions.

Differential quadrature method is worth mentioning as presented in [14, 15]. There Lagrange interpolation and/or modified cubic B-splines are used, depending on boundary conditions, to approximate the solution to two-dimensional nonlinear hyperbolic partial differential equations. Irregular interfaces and discontinuities were, however, not studied there.

Another approach is within the context of finite difference methods (FDM); e.g., see [16–25] and the references therein. For finite difference methods the grid is Cartesian (uniform or adaptive), which leads to a straightforward grid generation

process. However, interfaces must be represented by other means and the treatment of boundary conditions requires additional considerations.

In order to capture the interface and enforce the correct boundary conditions, several approaches have been explored. The immersed boundary method (for example, [26–30]) uses the δ -formulation that smears out the solution profile across the interface. This produces algorithms that are straightforward to implement since they are similar to solving the same equations on a regular domain. However, the smearing of the solution introduces $O(1)$ errors near the interface. The immerse interface method (IIM) (see, for example, [31–37]) is a sharp interface method that leads to second-order accurate solution, albeit it is not a robust second-order method since it reaches its order by minimizing the truncation error. The method leads to neither symmetric nor positive definite linear systems and the application of the immerse interface method may be difficult in three spatial dimensions.

In [20], Liu *et al.* introduced a Ghost-Fluid Methodology (a finite difference approach) that treats the jump condition in a dimension-by-dimension framework. This leads to a linear system that is symmetric positive definite and the jump conditions only affect the right-hand side of the linear system, leading to an easy-to-implement method. However, the GFM suffers a loss in accuracy due to smearing of the tangential derivative of the discontinuity at the interface caused by this dimension-by-dimension framework, as indicated in [20].

The Voronoi Interface Method (VIM) [38] avoids the loss of accuracy of the GFM of [20] by constructing local Voronoi partitioning near the interface. The cells adjacent to the interface therefore have their faces orthogonal to the fluxes of the solution, hence providing a configuration that can leverage the Ghost-Fluid Methodology to its fullest. The advantage is therefore that the solution is second-order accurate. A drawback is that the solution is computed at the cells' center of the Voronoi partition and an additional interpolation step is required, if the solution is needed on the original Cartesian mesh. As is the case of GFM of [20], the linear system is symmetric positive definite and only its right-hand side is modified by the jump conditions. Finally, we note that even though the construction of a global Voronoi partition may be difficult and costly, the construction of a local partition, i.e., only for cells that are adjacent to the interface, is straightforward. In [38], the authors used the Voro++ library of [39].

In this paper we highlight the performance of both GFM and VIM. It is worth noting that both methods are unconditionally stable since they are implicit. Both methods have also shown that they can handle high ratios of discontinuity in their coefficients.

2. Governing Equations and Numerical Methods

We consider the Poisson equation with variable coefficient and discontinuities (jumps) across an irregular interface, Γ ,

which splits the computational domain Ω into two domains Ω^+ and Ω^- , both in \mathbb{R}^n , $n \in \mathbb{N}$. The governing equation is

$$\nabla \cdot (\beta \nabla u) = f \quad \text{for } \underline{x} \in \Omega^+ \cup \Omega^-, \quad (1)$$

where $f = f(\underline{x})$ and $\beta = \beta(\underline{x})$ are given. Here, β is bounded from below by a positive constant and f is in $L^2(\Omega)$. This equation is supplemented by jump conditions on the irregular interface,

$$\begin{aligned} [u] &= a, \\ [\beta \nabla u \cdot \underline{n}] &= h, \end{aligned} \quad (2)$$

where $[u] = u_{\Gamma^+} - u_{\Gamma^-}$ denotes a jump in u across Γ . The functions $a = a(\underline{x})$ and $h = h(\underline{x})$ are given. Either Dirichlet, Neumann, or Robin boundary conditions can be imposed at the boundaries of the computational domain, $\partial\Omega$. In order to represent the irregular interface, we use the signed distance level-set function, ϕ , which is positive inside Ω^+ , negative inside Ω^- , and zero on Γ . The outward normal to the irregular interface can be computed from the level-set function by

$$\underline{n} = (n_1, n_2, n_3) = \frac{\nabla \phi}{|\nabla \phi|}. \quad (3)$$

2.1. The Ghost-Fluid Method. A detailed description of the Ghost-Fluid Method is given in [20], we will explore here the main aspects in two spatial dimensions. The discontinuity in βu_n , where u_n refers to the derivative in the normal direction to the interface, is $[\beta u_n]_{\Gamma} = [\beta u_x]_{\Gamma} n_1 + [\beta u_y]_{\Gamma} n_2$. The discontinuity in βu_t , where u_t refers to the derivative in the tangential direction to the interface, is $[\beta u_t]_{\Gamma} = [\beta u_x]_{\Gamma} n_2 - [\beta u_y]_{\Gamma} n_1$. These equations lead to $[\beta u_x]_{\Gamma} = [\beta u_n]_{\Gamma} n_1 + [\beta u_t]_{\Gamma} n_2$ and $[\beta u_y]_{\Gamma} = [\beta u_n]_{\Gamma} n_2 - [\beta u_t]_{\Gamma} n_1$. However, in order to devise a method in a dimension-by-dimension framework, the Ghost-Fluid Method smears out the discontinuity in the tangential derivative leading to the simplification $[\beta u_x]_{\Gamma} = [\beta u_n]_{\Gamma} n_1$ and $[\beta u_y]_{\Gamma} = [\beta u_n]_{\Gamma} n_2$. The discretization at each grid point i, j is then given by

$$\begin{aligned} & \frac{\beta_{i+1/2,j} ((u_{i+1,j} - u_{i,j}) / \Delta x) - \beta_{i-1/2,j} ((u_{i,j} - u_{i-1,j}) / \Delta x)}{\Delta x} \\ & + \frac{\beta_{i,j+1/2} ((u_{i,j+1} - u_{i,j}) / \Delta y) - \beta_{i,j-1/2} ((u_{i,j} - u_{i,j-1}) / \Delta y)}{\Delta y} \quad (4) \\ & = f_{i,j} + F^x + F^y, \end{aligned}$$

where $\beta_{i+1/2,j} = (\beta_i + \beta_{i+1})/2$ if $x_{i,j}$ and $x_{i+1,j}$ are on the same side of the interface or by $\beta_{i+1/2,j} = \beta^+ \beta^- (|\phi^-| + |\phi^+|) / (\beta^+ |\phi^-| + \beta^- |\phi^+|)$ otherwise, where ϕ^+ is the value of ϕ of the node in Ω^+ and ϕ^- is the value of ϕ of the node in Ω^- . Here β^{\pm} refers to the value of β adjacent to the interface in the Ω^{\pm} domain and Δx and Δy are the cells' sizes in the x - and y -directions, respectively. The left-hand side thus gives the same symmetric positive definite matrix as the one generated by the standard five-point stencil discretization of the Poisson equation on regular domains and only the right-hand side is altered when a discontinuity occurs. Furthermore,

$F^x = F^L + F^R$ and $F^y = F^B + F^T$ are only activated if there is a discontinuity present in the local five point stencil. Here, F^L and F^R and the contribution from the left and right grid points to the current grid point and F^B and F^T are the contributions from the bottom and top grid points.

We give the details for F^L and F^R in the x -direction, referring the reader to the original paper [20] for the description for F^B and F^T in the y -direction, since they follow the same procedure. If $\phi_{i-1,j}$ and $\phi_{i,j}$ have opposite signs, define $\theta = |\phi_{i-1,j}|/(|\phi_{i,j}| + |\phi_{i-1,j}|)$, $a_\Gamma = (a_{i,j}|\phi_{i-1,j}| + a_{i-1,j}|\phi_{i,j}|)/(|\phi_{i,j}| + |\phi_{i-1,j}|)$, where $a_{i,j}$ is the jump at the grid node and $\beta_\Gamma = (b_{i,j}n_{i,j}^1|\phi_{i-1,j}| + b_{i-1,j}n_{i-1,j}^1|\phi_{i,j}|)/(|\phi_{i,j}| + |\phi_{i-1,j}|)$. F^L is then defined as

$$F^L = \begin{cases} \frac{\beta_{i-1/2,j}a_\Gamma}{(\Delta x)^2} - \frac{\beta_{i-1/2,j}b_\Gamma\theta}{\beta^+\Delta x} & \text{if } \phi_{i,j} \leq 0 \text{ and } \phi_{i-1,j} > 0, \\ -\frac{\beta_{i-1/2,j}a_\Gamma}{(\Delta x)^2} + \frac{\beta_{i-1/2,j}b_\Gamma\theta}{\beta^-\Delta x} & \text{if } \phi_{i,j} > 0 \text{ and } \phi_{i-1,j} \leq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Similarly, if $\phi_{i+1,j}$ and $\phi_{i,j}$ have opposite signs, define $\theta = |\phi_{i+1,j}|/(|\phi_{i,j}| + |\phi_{i+1,j}|)$, $a_\Gamma = (a_{i,j}|\phi_{i+1,j}| + a_{i+1,j}|\phi_{i,j}|)/(|\phi_{i,j}| + |\phi_{i+1,j}|)$, and $\beta_\Gamma = (b_{i,j}n_{i,j}^1|\phi_{i+1,j}| + b_{i+1,j}n_{i+1,j}^1|\phi_{i,j}|)/(|\phi_{i,j}| + |\phi_{i+1,j}|)$. F^R is then defined as

$$F^R = \begin{cases} \frac{\beta_{i+1/2,j}a_\Gamma}{(\Delta x)^2} + \frac{\beta_{i+1/2,j}b_\Gamma\theta}{\beta^+\Delta x} & \text{if } \phi_{i,j} \leq 0 \text{ and } \phi_{i+1,j} > 0, \\ -\frac{\beta_{i+1/2,j}a_\Gamma}{(\Delta x)^2} - \frac{\beta_{i+1/2,j}b_\Gamma\theta}{\beta^-\Delta x} & \text{if } \phi_{i,j} > 0 \text{ and } \phi_{i+1,j} \leq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The Ghost-Fluid Method leads to a symmetric positive definite linear systems that captures the discontinuity in the normal derivative while smearing out the discontinuity in the tangential direction to the interface.

2.2. The Voronoi Interface Method. We present a summary of the method and refer the reader to [38] for a detailed description. For a given set of seeds, the Voronoi cell of a seed is defined as the points of space that are closer to that seed than any other. The union of the Voronoi cells is a tessellation of space, and the first step for the Voronoi Interface Method is to generate the Voronoi mesh associated with the background mesh chosen, which in our case is a uniform Cartesian grid. The seeds are of two types,

- (i) if a cell of the background mesh is not crossed by the irregular interface Γ , its center is a seed of the Voronoi mesh,
- (ii) if a cell of the background mesh is crossed by the irregular interface Γ , we locate the projection of its center onto the interface and generate two seeds located on either side of the projected point at a distance $\delta = \min(\Delta x, \Delta y)/5$.

The Voronoi partition associated with those seeds is constructed using a simple geometric algorithm. We refer the reader to the Voronoi++ library [39] for an efficient tool to compute Voronoi partitions in both two and three spatial dimensions.

Equation (1) is then discretized on the Voronoi mesh using a finite volume approach. The complete derivation is presented in [38] and leads to the discretization of the interaction between point i and point j

$$\begin{aligned} & \tilde{\beta}_{ij} s_{ij} \frac{u_j - u_i}{d_{ij}} \\ & = \tilde{\beta}_{ij} \frac{s_{ij}}{d_{ij}} \left(-\text{sign}(\phi_i) [u] + \frac{d_{ij}}{2\beta_j} [\beta \nabla u \cdot \mathbf{n}_\Gamma] \right) \quad (7) \\ & \quad + \text{Vol}(\mathcal{C}_i) f_i, \end{aligned}$$

where any variable of the form γ_k would be the quantity γ at point k , d_{ij} is the distance between i and j , s_{ij} is the length of the face between i and j , $\tilde{\beta}_{ij} = 2\beta_i\beta_j/(\beta_i + \beta_j)$, and $\text{Vol}(\mathcal{C}_i)$ is the volume of the Voronoi cell associated with point i . The discontinuities at the interface only affect the right-hand side of the linear system where the irregular interface is located and the system is symmetric positive definite.

We remark that the solution is provided at the center of the Voronoi cells. If the solution is needed on the original background mesh, then an interpolation step is required. This can be done, for example, with least square interpolations and it does not impact the order of accuracy of the solution as long as the order of the polynomial interpolants is high enough. In this article, we work with second-order polynomial interpolants.

3. Numerical Experiments: Results

We present a pair of two-dimensional numerical examples where the two methods are compared. Both examples have a star shaped irregular interface. The first example has a coefficient β that is constant in the whole domain. This example has a constant discontinuity in the solution across the irregular interface but neither a discontinuity in the normal nor tangential derivatives at the interface. The second example has a β coefficient that is not constant and has a discontinuity across the irregular interface. This example has a nonconstant discontinuity in the solution and nonconstant discontinuities in the normal and tangential components of the gradient of the solution at the irregular interface.

3.1. Constant β -Coefficient. Let us consider $\nabla \cdot (\beta \nabla u) = f(x, y)$ in two spatial dimensions in $\Omega = [-1, 1]^2$ with the level set function $\phi = -\sqrt{x^2 + y^2} + 0.5 + 0.15 \cos(5 \arctan(y/x))$. We take $\beta = 1$ when $\phi \leq 0$ and an exact solution of $u = \cos(x)\cos(y)$. For the region $\phi > 0$, we take $\beta = 1$ and the exact solution to be $u = \cos(x)\cos(y) + 1$. A representation of the solution and the Voronoi mesh are given in Figure 1. The comparison of the Ghost-Fluid Method and the Voronoi Interface Method is shown in Table 1 for the maximum

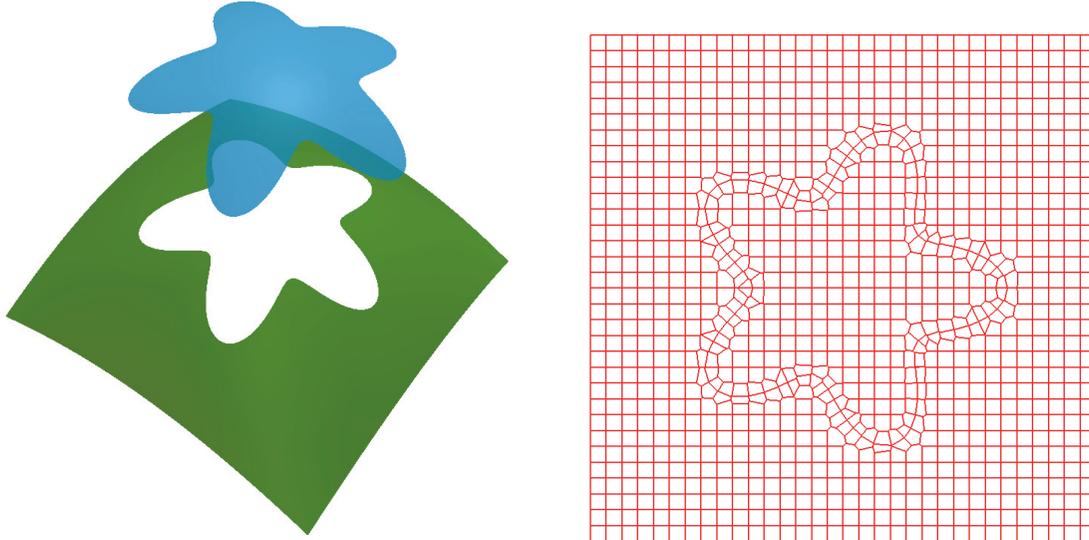


FIGURE 1: Left: visualization of the exact solution, for example, shown in Section 3.1. Right: example of a Voronoi mesh generated.

TABLE 1: Comparison of the L^∞ error in the solution for the Ghost-Fluid Method and the Voronoi Interface Method, for example, shown in Section 3.1.

dx	Ghost-Fluid Method		Voronoi Interface Method		VIM Interpolated	
	$\ u - u_h\ _\infty$	Order	$\ u - u_h\ _\infty$	Order	$\ u - u_h\ _\infty$	Order
0.25	2.627×10^{-3}	—	3.709×10^{-3}	—	4.098×10^{-1}	—
0.125	6.587×10^{-4}	2.00	6.172×10^{-4}	2.59	5.901×10^{-2}	2.80
0.0625	1.648×10^{-4}	2.00	1.565×10^{-4}	1.98	1.788×10^{-4}	8.37
0.03125	4.121×10^{-5}	2.00	4.016×10^{-5}	1.96	4.030×10^{-5}	2.15
0.015625	1.030×10^{-5}	2.00	1.020×10^{-5}	1.98	1.023×10^{-5}	1.98
0.0078125	2.576×10^{-6}	2.00	2.598×10^{-6}	1.97	2.656×10^{-6}	1.95
0.00390625	6.439×10^{-7}	2.00	6.591×10^{-7}	1.98	6.655×10^{-7}	2.00

TABLE 2: Comparison of the L^1 error in the solution for the Ghost-Fluid Method and the Voronoi Interface Method, for example, shown in Section 3.1.

dx	Ghost-Fluid Method		Voronoi Interface Method		VIM Interpolated	
	$\ u - u_h\ _1$	Order	$\ u - u_h\ _1$	Order	$\ u - u_h\ _1$	Order
0.25	9.010×10^{-4}	—	1.790×10^{-2}	—	4.694×10^{-2}	—
0.125	2.597×10^{-4}	1.79	2.918×10^{-3}	5.04	1.417×10^{-3}	5.04
0.0625	6.940×10^{-5}	1.90	7.365×10^{-5}	4.31	7.130×10^{-5}	4.31
0.03125	1.792×10^{-5}	1.95	1.848×10^{-5}	2.00	1.780×10^{-5}	2.00
0.015625	4.552×10^{-6}	1.98	4.625×10^{-6}	1.98	4.518×10^{-6}	1.98
0.0078125	1.147×10^{-6}	1.99	1.156×10^{-6}	1.98	1.142×10^{-6}	1.98
0.00390625	2.879×10^{-7}	1.99	2.890×10^{-7}	1.99	2.871×10^{-7}	1.99

error in the solution, in Table 2 for the average error in the solution, in Table 3 for the maximum error of the gradient, and in Table 4 for the average error of the gradient. For the Voronoi Interface Method, we present the errors on both the Voronoi mesh and the Cartesian mesh. For the Voronoi mesh, the gradients are computed on the faces of the Voronoi cells. For example, the solution only experiences a constant discontinuity in its solution and no discontinuities in its normal or tangential derivatives nor is there a discontinuity in

the β coefficient, which is constant in the entire domain. Both methods give second-order accuracy in the solution in the L^∞ - and L^1 -norms and the solution's gradient in the L^1 -norm. However, the Voronoi method gives first-order accuracy for the gradient of the solution in the L^∞ -norm while the GFM gives second-order accuracy for such simplified problems. A likely explanation for this difference is that the fluxes in the volume of fluid derivation for the Voronoi Interface Method are not necessarily computed at the center of the faces

TABLE 3: Comparison of the L^∞ error in the gradient of the solution for the Ghost-Fluid Method and the Voronoi Interface Method, for example, shown in Section 3.1.

dx	Ghost-Fluid Method		Voronoi Interface Method		VIM Interpolated	
	$\ \nabla u - \nabla u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	Order
0.25	4.829×10^{-3}	—	8.160×10^{-3}	—	1.872×10^{-0}	—
0.125	1.398×10^{-3}	1.79	1.550×10^{-3}	2.40	3.666×10^{-1}	2.35
0.0625	3.776×10^{-4}	1.89	5.192×10^{-4}	1.58	4.180×10^{-2}	3.13
0.03125	9.866×10^{-5}	1.94	4.367×10^{-4}	0.25	2.082×10^{-2}	1.01
0.015625	2.530×10^{-5}	1.96	2.551×10^{-4}	0.78	1.037×10^{-2}	1.00
0.0078125	6.419×10^{-6}	1.98	1.376×10^{-4}	0.89	5.195×10^{-3}	0.99
0.00390625	1.618×10^{-6}	1.99	8.580×10^{-5}	0.68	2.597×10^{-3}	1.00

TABLE 4: Comparison of the L^1 error in the gradient of the solution for the Ghost-Fluid method and the Voronoi Interface Method, for example, shown in Section 3.1.

dx	Ghost-Fluid Method		Voronoi Interface Method		VIM Interpolated	
	$\ \nabla u - \nabla u_h\ _1$	Order	$\ \nabla u - \nabla u_h\ _1$	Order	$\ \nabla u - \nabla u_h\ _1$	Order
0.25	3.636×10^{-3}	—	2.575×10^{-3}	—	3.797×10^{-1}	—
0.125	8.704×10^{-4}	2.06	3.490×10^{-4}	2.88	2.908×10^{-2}	3.71
0.0625	2.155×10^{-4}	2.01	7.553×10^{-5}	2.21	5.617×10^{-3}	2.37
0.03125	5.343×10^{-5}	2.01	1.711×10^{-5}	2.14	1.474×10^{-3}	1.93
0.015625	1.331×10^{-5}	2.01	4.126×10^{-6}	2.05	3.707×10^{-4}	1.99
0.0078125	3.322×10^{-6}	2.00	9.930×10^{-7}	2.05	9.345×10^{-5}	1.99
0.00390625	8.298×10^{-7}	2.00	2.465×10^{-7}	2.01	2.350×10^{-5}	1.99

TABLE 5: Comparisons of the L^∞ error in the solution for the Ghost-Fluid Method and the Voronoi Interface Method, for example, shown in Section 3.2.

dx	Ghost-Fluid Method		Voronoi Interface Method		VIM Interpolated	
	$\ u - u_h\ _\infty$	Order	$\ u - u_h\ _\infty$	Order	$\ u - u_h\ _\infty$	Order
0.25	6.254×10^{-1}	—	9.802×10^{-3}	—	6.044×10^{-1}	—
0.125	4.063×10^{-1}	0.62	2.560×10^{-3}	1.94	9.178×10^{-2}	2.72
0.0625	2.493×10^{-1}	0.70	7.709×10^{-4}	1.73	7.041×10^{-4}	7.03
0.03125	1.474×10^{-1}	0.76	2.370×10^{-4}	1.70	2.160×10^{-4}	1.70
0.015625	8.488×10^{-2}	0.80	5.560×10^{-5}	2.09	4.929×10^{-5}	2.13
0.0078125	4.802×10^{-2}	0.82	1.413×10^{-5}	1.98	1.293×10^{-5}	1.93
0.00390625	1.481×10^{-2}	0.84	3.618×10^{-6}	1.97	3.377×10^{-6}	1.94

between two points. We also observe that the interpolation from the Voronoi mesh to the Cartesian mesh does not impact the solution itself but affects its gradient. However, the order of accuracy is conserved. We conclude that for such simple problems the GFM is preferable.

3.2. *Nonconstant β -Coefficient.* Let us consider $\nabla \cdot (\beta \nabla u) = f(x, y)$ in two spatial dimensions in $\Omega = [-1, 1]^2$ with the level set function $\phi = -\sqrt{x^2 + y^2} + 0.5 + 0.15 \cos(5 \arctan(y/x))$. We set $\beta = y^2 \ln(x + 2) + 4$ and the exact solution to $u = e^x$ in the region where $\phi \leq 0$. In the region where $\phi > 0$, we set $\beta = e^{-y}$ and the exact solution to $u = \cos(x)\sin(y)$. Figure 2 provides a representation of the solution and of the diffusion coefficient. The comparison of the Ghost-Fluid Method and the Voronoi Interface Method is shown in Table 5

for the maximum error in the solution, in Table 6 for the average error in the solution, in Table 7 for the maximum error of the gradient, and in Table 8 for the average error of the gradient. Again, for the Voronoi Interface Method we present the results on both the Voronoi and the Cartesian meshes. For example, there is a nonconstant discontinuity in the solution, a nonconstant discontinuity in the normal derivative of the solution to the interface, a nonconstant discontinuity in the tangential derivative of the solution to the interface, a discontinuity in the β coefficient across the interface, and a β coefficient that is not constant in each domain as in the previous example. The Ghost-Fluid Method is only first-order accurate in both the L^∞ - and the L^1 -norm, as well as in the L^1 -norm for the gradient of the solution. However, it is not consistent in the L^∞ -norm for the gradient of the solution. This example is a case where the lack of



FIGURE 2: Visualization of the exact solution u (left) and of the diffusion coefficient β (right), for example, shown in Section 3.2.

TABLE 6: Comparison of the L^1 error in the solution for the Ghost-Fluid Method and the Voronoi Interface Method, for example, shown in Section 3.2.

dx	Ghost-Fluid Method		Voronoi Interface Method		VIM Interpolated	
	$\ u - u_h\ _1$	Order	$\ u - u_h\ _1$	Order	$\ u - u_h\ _1$	Order
0.25	6.446×10^{-2}	—	2.291×10^{-3}	—	4.557×10^{-2}	—
0.125	3.393×10^{-2}	0.93	4.022×10^{-4}	2.51	2.041×10^{-3}	4.48
0.0625	1.692×10^{-2}	1.00	1.038×10^{-4}	1.95	9.891×10^{-5}	4.37
0.03125	7.908×10^{-3}	1.10	2.985×10^{-5}	1.80	2.718×10^{-5}	1.86
0.015625	3.982×10^{-3}	0.99	7.243×10^{-6}	2.04	6.729×10^{-6}	2.01
0.0078125	2.155×10^{-3}	0.89	1.790×10^{-6}	2.02	1.717×10^{-6}	1.97
0.00390625	1.035×10^{-3}	1.06	4.509×10^{-7}	1.99	4.401×10^{-7}	1.96

TABLE 7: Comparison of the L^∞ error in the gradient of the solution for the Ghost-Fluid Method and the Voronoi Interface Method, for example, shown in Section 3.2.

dx	Ghost-Fluid Method		Voronoi Interface Method		VIM Interpolated	
	$\ \nabla u - \nabla u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	Order
0.25	1.259	—	1.309×10^{-2}	—	2.658×10^{-0}	—
0.125	1.227	0.04	9.931×10^{-3}	0.55	6.334×10^{-1}	2.07
0.0625	1.139	0.11	2.954×10^{-3}	1.60	8.950×10^{-2}	2.82
0.03125	1.156	-0.02	1.589×10^{-3}	0.89	4.373×10^{-2}	1.03
0.015625	1.188	-0.04	1.038×10^{-3}	0.62	2.155×10^{-2}	1.02
0.0078125	1.205	-0.02	5.177×10^{-4}	1.00	1.070×10^{-2}	1.01
0.00390625	1.214	-0.01	3.194×10^{-4}	0.70	5.329×10^{-3}	1.01

orthogonality between the cells' faces and the solution's fluxes lowers the accuracy of the GFM's dimension-by-dimension approach. The Voronoi Interface Method provides a solution to that drawback and, therefore, produces a second-order accurate solution in the L^∞ - and L^1 -norms, and first-order accurate (resp., second-order accurate) gradients in the L^∞ -

(resp., L^1 -) norm. Similarly to the previous example, we observe a decrease in accuracy for the gradient computed on the Cartesian mesh after the interpolation step, but the order of accuracy is conserved. The solution itself does not suffer from the interpolation step. For this type of complex problems, the Voronoi Interface Method is able to provide a

TABLE 8: Comparison of the L^1 error in the gradient of the solution for the Ghost-Fluid Method and the Voronoi Interface Method, for example, shown in Section 3.2.

dx	Ghost-Fluid Method		Voronoi Interface Method		VIM Interpolated	
	$\ \nabla u - \nabla u_h\ _1$	Order	$\ \nabla u - \nabla u_h\ _1$	Order	$\ \nabla u - \nabla u_h\ _1$	Order
0.25	2.611×10^{-1}	—	2.126×10^{-3}	—	3.895×10^{-1}	—
0.125	1.438×10^{-1}	0.86	6.395×10^{-4}	1.73	3.400×10^{-2}	3.52
0.0625	7.136×10^{-2}	1.01	1.957×10^{-4}	1.71	5.631×10^{-3}	2.59
0.03125	3.922×10^{-2}	0.86	6.409×10^{-5}	1.61	1.464×10^{-3}	1.94
0.015625	2.019×10^{-2}	0.96	1.593×10^{-5}	2.01	3.713×10^{-4}	1.98
0.0078125	1.033×10^{-2}	0.97	4.132×10^{-6}	1.95	9.359×10^{-5}	1.99
0.00390625	5.222×10^{-3}	0.98	1.099×10^{-6}	1.91	2.358×10^{-5}	1.99

second-order accurate solution and consistent gradient that the Ghost-Fluid Method cannot produce. VIM is therefore the recommended approach.

4. Conclusion

This paper has considered the numerical solution of the Poisson equation with jump conditions across an irregular interface. In particular, we have compared the results obtained with the Ghost-Fluid Method and the Voronoi Interface Method. The Ghost-Fluid Method imposes the jump conditions in a dimension-by-dimension framework, leading to a linear system that is symmetric positive definite in which the jump conditions only affect the right-hand side. However, the dimension-by-dimension approach forces a smearing of the tangential quantities in the jump, leading to a loss of accuracy unless both the discontinuity in the solution and the variable coefficient β are constant. The Voronoi Interface method solves that problem by constructing a Voronoi partition for cells adjacent to the interface. A finite volume discretization over those cells produces discretized fluxes that are orthogonal to the cells' faces themselves and aligned with the normal direction to the interface. The Ghost-Fluid philosophy can therefore be readily applied, resulting in a linear system that is also symmetric positive definite with only its right-hand side affected by the jump conditions. The resulting solution is second-order accurate (versus first-order accurate in the general case of the Ghost-Fluid Method) and the solution's gradient is first-order accurate (versus zeroth-order accurate in the case of the Ghost-Fluid Method). The Voronoi Interface Method can therefore be considered superior in general. In the particular case where both the discontinuity across the interface is constant and the variable coefficient β is constant over each subdomains and across the interface, the Ghost-Fluid Method gives a second-order accurate solution and also second-order accurate gradient, giving this approach an advantage over the Voronoi Interface Method. The likely reason is that the discrete fluxes for the Voronoi Interface Method are not located at the center of the faces between two points. Finally, the Voronoi Interface Method provides the solution at the center of the Voronoi cells. An interpolation step is required if the solution on the original Cartesian mesh is needed which can, for example, be carried out with least square interpolations. The order of

accuracy is then preserved though the quality of the gradient of the solution is impacted.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The present address of Arthur Guittet is Google LLC, 1600 Amphitheatre Parkway Mountain View, CA 94043.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The research of Á. Helgadóttir was supported by the University of Iceland Research Fund 2015 under HII4090070. The researches of A. Guittet and F. Gibou were supported in part by the NSF under DMS-1412695 and DMREF-1534264.

References

- [1] I. Babuska, "The finite element method for elliptic equations with discontinuous coefficients," *Computing*, vol. 5, pp. 207–213, 1970.
- [2] J. H. Bramble and J. T. King, "A finite element method for interface problems in domains with smooth boundaries and interfaces," *Advances in Computational Mathematics*, vol. 6, no. 2, pp. 109–138, 1996.
- [3] Z. Chen and J. Zou, "Finite element methods and their convergence for elliptic and parabolic interface problems," *Numerische Mathematik*, vol. 79, no. 2, pp. 175–202, 1998.
- [4] M. Dryja, "A Neumann-Neumann algorithm for a mortar discretization of elliptic problems with discontinuous coefficients," *Numerische Mathematik*, vol. 99, no. 4, pp. 645–656, 2005.
- [5] J. Huang and J. Zou, "A mortar element method for elliptic problems with discontinuous coefficients," *IMA Journal of Numerical Analysis (IMAJNA)*, vol. 22, no. 4, pp. 549–576, 2002.

- [6] B. P. Lamichhane and B. I. Wohlmuth, "Mortar finite elements for interface problems," *Computing*, vol. 72, no. 3-4, pp. 333–348, 2004.
- [7] A. Hansbo and P. Hansbo, "A finite element method for the simulation of strong and weak discontinuities in solid mechanics," *Computer Methods Applied Mechanics and Engineering*, vol. 193, no. 33–35, pp. 3523–3540, 2004.
- [8] A. Hansbo and P. Hansbo, "An unfitted finite element method, based on Nitsche's method, for elliptic interface problems," *Computer Methods Applied Mechanics and Engineering*, vol. 191, no. 47-48, pp. 5537–5552, 2002.
- [9] J.-H. Song, P. M. A. Areias, and T. Belytschko, "A method for dynamic crack and shear band propagation with phantom nodes," *International Journal for Numerical Methods in Engineering*, vol. 67, no. 6, pp. 868–893, 2006.
- [10] J. Dolbow and I. Harari, "An efficient finite element method for embedded interface problems," *International Journal for Numerical Methods in Engineering*, vol. 78, no. 2, pp. 229–252, 2009.
- [11] S. Gross and A. Reusken, "An extended pressure finite element space for two-phase incompressible flows with surface tension," *Journal of Computational Physics*, vol. 224, no. 1, pp. 40–58, 2007.
- [12] L. Parussini and V. Pediroda, "Fictitious domain approach with hp-finite element approximation for incompressible fluid flow," *Journal of Computational Physics*, vol. 228, no. 10, pp. 3891–3910, 2009.
- [13] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, NY, USA, 2nd edition, 2003.
- [14] R. Jiwari, "Lagrange interpolation and modified cubic B-spline differential quadrature methods for solving hyperbolic partial differential equations with Dirichlet and Neumann boundary conditions," *Computer Physics Communications*, vol. 193, pp. 55–65, 2015.
- [15] S. Pandit, R. Jiwari, K. Bedi, and M. E. Koksai, "Haar wavelets operational matrix based algorithm for computational modelling of hyperbolic type wave equations," *Engineering Computations (Swansea, Wales)*, vol. 34, no. 8, pp. 2793–2814, 2017.
- [16] R. J. LeVeque and Z. Li, "Immersed interface methods for Stokes flow with elastic boundaries or surface tension," *SIAM Journal on Scientific Computing*, vol. 18, no. 3, pp. 709–735, 1997.
- [17] Z. Tan, D. V. Le, Z. Li, K. M. Lim, and B. C. Khoo, "An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane," *Journal of Computational Physics*, vol. 227, no. 23, pp. 9955–9983, 2008.
- [18] J. T. Beale and A. T. Layton, "On the accuracy of finite difference methods for elliptic problems with interfaces," *Communications in Applied Mathematics and Computational Science*, vol. 1, pp. 91–119, 2006.
- [19] A. Wiegmann and K. P. Bube, "The explicit-jump immersed interface method: finite difference method for pdes with piecewise smooth solutions," *SIAM Journal on Numerical Analysis*, vol. 37, no. 3, pp. 827–862, 2000.
- [20] X.-D. Liu, R. P. Fedkiw, and M. Kang, "A boundary condition capturing method for Poisson's equation on irregular domains," *Journal of Computational Physics*, vol. 160, no. 1, pp. 151–178, 2000.
- [21] F. Gibou, R. P. Fedkiw, L.-T. Cheng, and M. Kang, "A second-order-accurate symmetric discretization of the Poisson equation on irregular domains," *Journal of Computational Physics*, vol. 176, no. 1, pp. 205–227, 2002.
- [22] F. Gibou and R. Fedkiw, "A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem," *Journal of Computational Physics*, vol. 202, no. 2, pp. 577–601, 2005.
- [23] Z. Jomaa and C. Macaskill, "The Shortley-Weller embedded finite-difference method for the 3D Poisson equation with mixed boundary conditions," *Journal of Computational Physics*, vol. 229, no. 10, pp. 3675–3690, 2010.
- [24] I.-L. Chern and Y.-C. Shu, "A coupling interface method for elliptic interface problems," *Journal of Computational Physics*, vol. 225, no. 2, pp. 2138–2174, 2007.
- [25] Y. C. Zhou, S. Zhao, M. Feig, and G. W. Wei, "High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources," *Journal of Computational Physics*, vol. 213, no. 1, pp. 1–30, 2006.
- [26] C. S. Peskin, "Flow patterns around heart valves: a numerical method," *Journal of Computational Physics*, vol. 10, no. 2, pp. 252–271, 1972.
- [27] C. S. Peskin, "Numerical analysis of blood flow in the heart," *Journal of Computational Physics*, vol. 25, no. 3, pp. 220–252, 1977.
- [28] C. S. Peskin, "The immersed boundary method," *Acta Numerica*, vol. 11, pp. 479–517, 2002.
- [29] C. S. Peskin and B. F. Printz, "Improved volume conservation in the computation of flows with immersed elastic boundaries," *Journal of Computational Physics*, vol. 105, no. 1, pp. 33–46, 1993.
- [30] M. Sussman, P. Smereka, and S. Osher, "A level set approach for computing solutions to incompressible two-phase flow," *Journal of Computational Physics*, vol. 114, no. 1, pp. 146–159, 1994.
- [31] R. J. LeVeque and Z. L. Li, "The immersed interface method for elliptic equations with discontinuous coefficients and singular sources," *SIAM Journal on Numerical Analysis*, vol. 31, no. 4, pp. 1019–1044, 1994.
- [32] L. Adams and T. P. Chartier, "New geometric immersed interface multigrid solvers," *SIAM Journal on Scientific Computing*, vol. 25, no. 5, pp. 1516–1533, 2004.
- [33] L. Adams and T. P. Chartier, "A comparison of algebraic multigrid and geometric immersed interface multigrid methods for interface problems," *SIAM Journal on Scientific Computing*, vol. 26, no. 3, pp. 762–784, 2005.
- [34] L. Adams and Z. Li, "The immersed interface/multigrid methods for interface problems," *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 463–479, 2002.
- [35] L. Adams and T. Chartier, "A decomposed Immersed Interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions," *Journal of Computational Physics*, vol. 197, no. 1, pp. 364–386, 2004.
- [36] Z. Li and K. Ito, *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, vol. 33, SIAM, Philadelphia, NC, USA, 2006.
- [37] T. Y. Hou, Z. Li, S. Osher, and H. Zhao, "A hybrid method for moving interface problems with application to the Hele-Shaw flow," *Journal of Computational Physics*, vol. 134, no. 2, pp. 236–252, 1997.
- [38] A. Guittet, M. Lepilliez, S. Tanguy, and F. Gibou, "Solving elliptic problems with discontinuities on irregular domains—the Voronoi interface method," *Journal of Computational Physics*, vol. 298, pp. 747–765, 2015.
- [39] C. H. Rycroft, "VORO++: a three-dimensional voronoi cell library in C++," *Chaos*, vol. 19, 2009.

