Open Archive TOULOUSE Archive Ouverte (OATAO)

Solving elliptic problems with discontinuities on irregular domains – the Voronoi Interface Method

Arthur Guittet [a,*], Mathieu Lepilliez [c,d,e], Sebastien Tanguy [c], Frédéric Gibou [a,b]

[a] *Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070, United States*
[b] *Department of Computer Science, University of California, Santa Barbara, CA 93106-5110, United States*
[c] *Institut de Mécanique des Fluides de Toulouse, 2bis allée du Professeur Camille Soula, 31400 Toulouse, France*
[d] *Centre National d'Etudes Spatiales, 18 Avenue Edouard Belin, 31401 Toulouse Cedex 9, France*
[e] *Airbus Defence & Space, 31 Avenue des Cosmonautes, 31402 Toulouse Cedex 4, France*

ABSTRACT

We introduce a simple method, dubbed the Voronoi Interface Method, to solve Elliptic problems with discontinuities across the interface of irregular domains. This method produces a linear system that is symmetric positive definite with only its right-hand-side affected by the jump conditions. The solution and the solution's gradients are second-order accurate and first-order accurate, respectively, in the L^∞ norm, even in the case of large ratios in the diffusion coefficient. This approach is also applicable to arbitrary meshes. Additional degrees of freedom are placed close to the interface and a Voronoi partition centered at each of these points is used to discretize the equations in a finite volume approach. Both the locations of the additional degrees of freedom and their Voronoi discretizations are straightforward in two and three spatial dimensions.

1. Introduction

We focus on the class of Elliptic problems that can be written as:

$$\underline{\nabla} \cdot (\beta \underline{\nabla} u) + ku = f \qquad \text{in } \Omega^- \cup \Omega^+,$$
$$[u] = g \qquad \text{on } \Gamma,$$
$$[\beta \underline{\nabla} u \cdot \underline{n}_\Gamma] = h \qquad \text{on } \Gamma, \tag{1}$$

where the computational domain Ω is composed of two subdomains, Ω^- and Ω^+, separated by a co-dimension one interface Γ (see Fig. 1), with $\underline{n}_\Gamma$ the outward normal. Here, $\beta = \beta(\underline{x})$, with $\underline{x} \in \mathbb{R}^n$ ($n \in \mathbb{N}$), is bounded from below by a positive constant and $[q] = q_\Gamma^+ - q_\Gamma^-$ indicates a discontinuity in the quantity q across Γ, f is in L^2, g, h and k are given. Note that this general formulation includes possible discontinuities in the diffusion coefficient β and in the gradient of the solution $\underline{\nabla} u$. Dirichlet or Neumann boundary conditions are applied on the boundary of Ω, denoted by $\partial\Omega$. This class of equations, where some or all of the jump conditions are non-zero, is a corner stone in the modeling of the dynamics of

---

* Corresponding author.
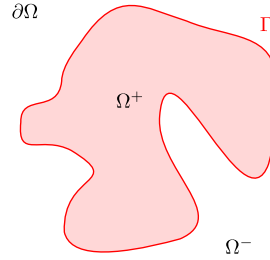 *E-mail address:* arthur.guittet@gmail.com (A. Guittet).

Fig. 1. Geometry of the problem.

important physical and biological phenomena as diverse as multiphase flows with and without phase change, biomolecules' electrostatics, electrokinetics (Poisson–Nerntz–Planck) models with source term or electroporation models.

Given the importance of this class of equations, several approaches have been pursued to computationally approximate their solutions, each with their own pros and cons. The finite element method (FEM) is one of the earliest approaches to solve this problem [4,10,12,19,32,35] and has the advantage of providing a simple discretization formalism that guarantees the symmetry and definite positiveness of the corresponding linear system, even in the case of unstructured grids. It also provides a framework where *a priori* error estimates can be used to best adapt the mesh in order to capture small scale details. However, the FEM is based on the generation of meshes that must conform to the irregular domain's boundary and must satisfy some restrictive quality criteria, a task that is difficult, especially in three spatial dimensions. The difficulty is exacerbated when the domain's boundary evolves during the course of a computation, as it is the case for most of the applications modeled by these equations. Mesh generation is the focus of intense research [58], as the creation of unwanted sliver elements can deteriorate the accuracy of the solution.

Methods based on *capturing* the jump conditions do not depend on the generation of a mesh that conforms to the domain's boundary, hence avoiding the mesh generation difficulty altogether. However, they must impose the boundary conditions implicitly, which is a non-trivial task. A popular approach is the Immersed Interface Method (IIM) of Leveque and Li [36], and the more recent development of Immersed Finite Element Method (IFEM) and Immersed Finite Volume Method (IFVM) [40,27,23]. The basis of the IIM is to use Taylor expansions of the solution on each side of the interface and modify the stencils local to the interface in order to impose the jump conditions. As such, solutions can be obtained on simple Cartesian grids and the solution is second-order accurate in the L^∞ norm. The corresponding linear system, however, is asymmetric unless the coefficient β has no jump across the interface. Another difficulty is the need to approximate surface derivatives along Γ as well as the evaluation of high-order jump conditions. These difficulties have been addressed in the Piecewise-polynomial Interface Method of Chen and Strain [11] and several other approaches have improved the efficacy of the IMM [38,39,64,9,1–3]. We note also that the earliest approach on Cartesian grid is that of Mayo [43], who derived an integral equation to solve the Poisson and the bi-harmonic problems with piecewise constant coefficients on irregular domains; the solution is second-order accurate in the L^∞ norm. We also refer the interested researcher to the matched interface and boundary (MIB) method [66,65].

The finite element community has also proposed embedded interface approaches, including discontinuous Galerkin and the eXtended Finite Element Method (XFEM) [37,29,48,17,8,47,33,22,28,62]. The basic idea is to introduce additional degrees of freedom[1] near the interface and augment the standard basis functions on these elements with basis functions that are combined with a Heaviside function in order to help capture the jump conditions.

In [16], the authors introduce a second-order accurate discretization in the case of, possibly adaptive, Cartesian meshes using a cut-cell approach. The jump condition is imposed by determining the fluxes on both side of the interface, which are constructed from a combination of least squares and quadratic approximations. In [51], the authors also use a cut cell approach but impose the jump with the help of a compact 27-point stencil.

The Ghost Fluid Method (GFM), originally introduced to approximate two-phase compressible flows [21], has been applied to the system problem (1) in [41]. The basic idea is to consider fictitious domains and ghost values that capture the jump conditions in the discretization at grid nodes near the interface. An advantage of this approach is that only the right-hand-side of the linear system is affected by the jump conditions. However, in order to propose a dimension-by-dimension approach, the projection of the normal jump conditions must be projected onto the Cartesian directions. As a consequence, the tangential component of the jump is ignored. Nonetheless, the method has been shown to be convergent with first-order accuracy [42]. The GFM was also shown to produce symmetric positive definite second-order accuracy [25] and even fourth-order accuracy [24], but for a different class of problem, namely for solving Elliptic problems on irregular domains with Dirichlet boundary conditions. In fact, symmetric positive definite second-order accurate solutions can also be obtained in the case where Neumann or Robin boundary conditions are imposed on irregular domains [55,50,53,54]. These methods can be trivially extended to the case of adaptive Cartesian grids and we refer the interested readers to the review of Gibou, Min and Fedkiw [26] for more details. In the case of jump conditions, Coco and Russo [14] have also used a fictitious domain

---

[1] We understand by degrees of freedom the set of locations at which the solution is sampled.

approach, where a relaxation scheme is used to impose the boundary condition; the solution is second-order accurate. The same authors have also introduced a method to consider Dirichlet, Neumann and Robin boundary conditions on an irregular interface [15]. Latige et al. [46] have also presented a method based on fictitious domains using a piecewise polynomial representation of the solution on a dual grid, also obtaining second-order accurate solutions. Finally, [31] have applied the ghost fluid idea in a variational framework.

Related ideas are used in methods combining fictitious domains and variational formulations [49,7], dubbed virtual nodes approaches. Some of these approaches can be considered similar to XFEM methods [30,60,18], while others are different and offer advantages when considering under-resolved, possibly non-smooth, interfaces [49,59,56]. This philosophy has been used in [34], which introduces a virtual node algorithm for solving Elliptic problems on irregular geometries with jump conditions as well as Dirichlet or Neumann boundary conditions imposed on Γ. The solutions are second-order accurate in the L^∞ norm and the approach provides a unifying treatment for Dirichlet, Neumann and jump boundary conditions; however sacrificing simplicity.

Rather recently, Cisternino and Weynans [13] introduced a second-order accurate method that uses additional degrees of freedom on the domain's boundary and use them to discretize the Poisson operator with jump conditions in a dimension-by-dimension framework. The authors also present how to carefully approximate the gradients. The method produces second-order accurate solutions and a nonsymmetric linear system in part because of the need to change the size of the stencil for nodes adjacent to the domain's boundary.

We introduce a capturing computational approach, the Voronoi Interface Method, that produces second-order accurate solutions in the L^∞ norm. This approach is based on building a Voronoi diagram local to the interface, which enables the direct discretization of the jump conditions in the normal direction. The linear system is symmetric positive definite and the jump conditions only influence the right-hand-side. The construction of the local Voronoi mesh is a straightforward and parallelizable process and can be built with existing libraries that are freely available. In the present work, we use the excellent `Voro++` library in three spatial dimensions [57]. This method is different from body-fitted methods in that it relies on the post-processing of an existing background mesh, thus avoiding the standard difficulties associated with body-fitted approaches. We note that previous works have developed solvers for the Poisson equation on Voronoi diagrams (see [63,61] and the references therein); however discontinuities across an irregular interface were not considered.

2. The geometrical tools

2.1. The level-set method

The level-set method [52] is a powerful way of representing irregular interfaces as the zero contour of a continuous function. This representation is convenient in that it can be applied to the case of moving boundaries that can change their topology. It also provides a framework that lends itself to design *sharp* discretizations.

We use the level-set set framework to capture the irregular interface on which the discontinuities are enforced. We define a level-set function ϕ on the domain Ω such that the irregular interface Γ is described by $\Gamma = \{\underline{x} \in \mathbb{R}^n \in \Omega \mid \phi(\underline{x}) = 0\}$ and ϕ is negative on one side of the interface and positive on the other side, as pictured in Fig. 1. Even though infinitely many functions satisfy this criteria, it is convenient to work with a signed distance function to the irregular interface, i.e. a function that is negative on one side of the interface, positive on the other side and such that its magnitude at every point is the distance from the point to the interface. Constructing a signed distance function from an arbitrary function can be done for example by following the procedure explained in [44]. The normal to the interface is then obtained as

$$\underline{n} = \frac{\underline{\nabla}\phi}{\|\underline{\nabla}\phi\|},$$

and the curvature as

$$\kappa = \underline{\nabla} \cdot \underline{n}.$$

Note that if the level-set function is a signed distance function, $\|\underline{\nabla}\phi\| = 1$, and the projection onto Γ of any given point $\underline{x}$ is easily computed as:

$$\underline{x}_\Gamma = \underline{x} - \phi(\underline{x})\underline{\nabla}\phi(\underline{x}). \tag{2}$$

2.2. Voronoi diagrams

The solver we present is based on Voronoi diagrams, which can be generated locally with existing procedures and freely available libraries. In this work, we use the excellent `Voro++` library [57]. For the sake of clarity, we introduce the Voronoi diagram: given a set of points, which we call seeds, the Voronoi cell of a given seed consists of all the points of the domain that are closer to that seed than to any other seed. Hence, the collection of all the Voronoi cells of a set of seeds is a tessellation of the domain, i.e. a tiling that fills the domain and does not contain any overlaps.

Given a computational mesh, which we consider to be uniform in this section for clarity, we propose to modify the mesh so that the irregular interface coincides with the edges of the new mesh and the degrees of freedom close to the interface are all located at the same distance from the interface.
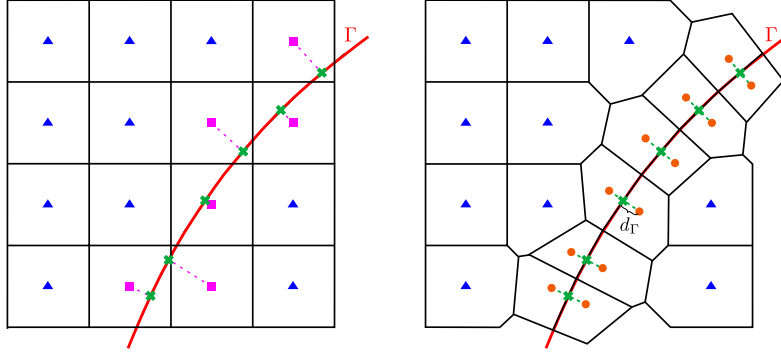
Fig. 2. Illustration of the procedure for generating a Voronoi diagram based computational mesh. The left figure presents the original uniform mesh and the right figure shows the final computational mesh. The purple square degrees of freedom have been removed and the orange dots degrees of freedom have been added close to the interface Γ.
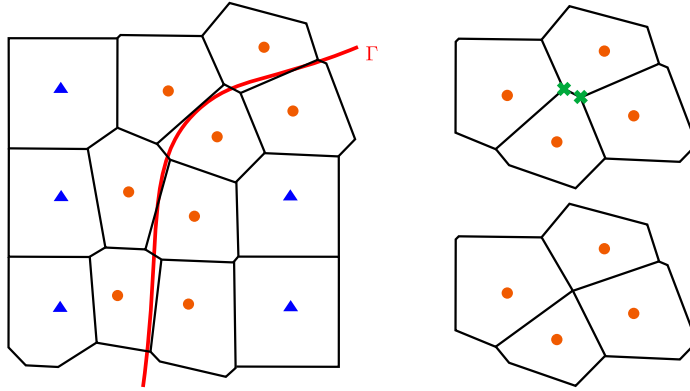

Fig. 3. Example of a Voronoi mesh where a degree of freedom is connected to more than one other degree of freedom located on the other side of the irregular interface. The smoothing procedure is illustrated on the right.

The procedure is illustrated in Fig. 2. Starting from a uniform grid, we find the projection of the degrees of freedom whose control volume is crossed by the irregular interface onto the interface using (2), and we remove those degrees of freedom from the original list of unknowns. If a projected point is within `diag`/5 of a previously computed projected point, where `diag` is the length of the diagonal of the smallest grid cell, we skip this point. Otherwise, we add two new degrees of freedom located at a distance d_Γ of the interface in the normal direction on either side of the interface. We repeat this procedure for all projected points. The new set of degrees of freedom is therefore made up of the original degrees of freedom whose control volume is not crossed by the interface and of the new degrees of freedom added next to the interface. This constitutes the set of seeds for the Voronoi diagram computational mesh on which we perform the computations. Each Voronoi cell can then be generated independently based on the local neighborhood of each degree of freedom, making the generation of the Voronoi mesh embarrassingly parallel.

Note that all the new degrees of freedom are placed at the same distance d_Γ from the interface. This is a free parameter of our method, and experimenting with various reasonable values shows little impact on the numerical results. We choose $d_\Gamma = $ `diag`/5. This simple procedure will be shown in Sections 4.1 and 4.2 to be sufficient to construct second-order accurate solutions in the L^∞ norm.

2.3. Smoothing the mesh

The algorithm described in the previous section can lead to undesirable geometrical configurations in the case when the interface is not sufficiently resolved. Fig. 3 presents one such configuration. The control volumes of some of the degrees of freedom are connected by a face that is not capturing properly the interface.

It is possible to remediate this issue by modifying the Voronoi partition in a post-processing step. The control volume of any degree of freedom that has been added next to the interface should be connected to exactly one control volume associated to a degree of freedom on the other side of the irregular interface. Consequently, if more than one neighbor is found across the interface, we disconnect the undesired ones by removing the connecting edge as shown in Fig. 3. The edge and its two associated vertices are then replaced by a single vertex located in the middle of the removed edge. The control volume of all the degrees of freedom of the resulting mesh are connected to at most one control volume on the other side of the interface and the interface is captured properly.
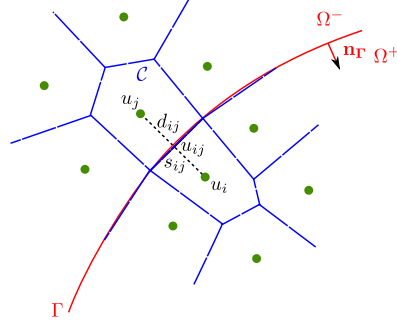
Fig. 4. Nomenclature for the finite volume discretization for the degree of freedom i. For each neighboring degree of freedom j, we call d_{ij} the distance between i and j, s_{ij} the length of the edge (or surface of the polygon in three spatial dimensions) connecting i and j, and u_{ij} the value of u at the middle of the segment $[i, j]$. Note that by construction u_{ij} can be considered to be exactly on the irregular interface Γ, in which case we define u_{ij}^+ and u_{ij}^-.

However, this procedure alters the mesh which is no longer a Voronoi diagram, and the edge between two degrees of freedom is not guaranteed to be orthogonal to the line connecting the two degrees of freedom. The impact of this post-processing algorithm is analyzed in Sections 4.1 and 4.2 and does not seem to improve the method, we therefore recommend not using it.

2.4. Interpolating back to the original mesh

In general, if solving a diffusion equation with discontinuities is part of a larger solver, it is necessary to interpolate the solution from the Voronoi mesh back to the original mesh. This is an easy task given some basic bookkeeping information linking the original degrees of freedom to the ones generated for the Voronoi mesh. With this information, the solution on the Voronoi mesh can be accessed for the same cost than accessing the data on the original mesh. The algorithm to interpolate the solution at a given point (x, y) from the Voronoi mesh is then as follows:

1. locate the cell of the original mesh containing (x, y),
2. using the bookkeeping information, identify the Voronoi degree of freedom $v(x, y)$ closest to (x, y),
3. find the two neighbors of $v(x, y)$ closest to (x, y) and on the same side of the interface,
4. compute the multilinear interpolation of the solution using those three degrees of freedom and evaluate it at (x, y).

This simple procedure produces a second-order interpolation at any given point (x, y).

3. Solving a Poisson equation on Voronoi diagrams

We discretize equation (1) with a finite volume approach on the Voronoi diagram introduced in Section 2.2. We use the notations from Fig. 4. We consider the degree of freedom i with the set of Voronoi neighbors $\{j\}$. Applying a finite volume approach to the problem at i, we can write

$$\int_{\mathcal{C}} \underline{\nabla} \cdot (\beta \underline{\nabla} u)\, dV = \int_{\partial \mathcal{C}} (\beta \underline{\nabla} u) \cdot \underline{\mathbf{n}}_{\mathcal{C}}\, dl \approx \sum_j s_{ij} \beta_i \frac{u_{ij} - u_i}{d_{ij}/2},$$

where $\underline{\mathbf{n}}_{\mathcal{C}}$ is the outer normal to the face of $\mathcal{C}$ connecting i and j, s_{ij} is the length of that face (or area of the surface in three spatial dimensions) and d_{ij} is the distance between the degrees of freedom i and j. For the case when i and j are on either side of the interface with $\phi_i > 0$, where ϕ_i is the value of the level-set function at the degree of freedom i, we can match the flux at the irregular interface as follows,

$$s_{ij} \beta_i \frac{u_{ij}^+ - u_i}{d_{ij}/2} = s_{ij} \beta_j \frac{u_j - u_{ij}^-}{d_{ij}/2} - s_{ij} \left[\beta \underline{\nabla} u \cdot \underline{\mathbf{n}}_\Gamma \right].$$

We also know that $u_{ij}^+ = u_{ij}^- + [u]$. Injecting this into the previous expression gives

$$\begin{aligned}
s_{ij} \beta_i \frac{u_{ij}^+ - u_i}{d_{ij}/2} &= s_{ij} \beta_j \frac{u_j - u_{ij}^+ + [u]}{d_{ij}/2} - s_{ij} \left[\beta \underline{\nabla} u \cdot \underline{\mathbf{n}}_\Gamma \right] \\
\Leftrightarrow \quad u_{ij}^+ (\beta_i + \beta_j) &= \beta_j u_j + \beta_i u_i + \beta_j [u] - \frac{d_{ij}}{2} \left[\beta \underline{\nabla} u \cdot \underline{\mathbf{n}}_\Gamma \right] \\
\Leftrightarrow \quad u_{ij}^+ &= \frac{1}{\beta_i + \beta_j} \left(\beta_j u_j + \beta_i u_i + \beta_j [u] - \frac{d_{ij}}{2} \left[\beta \underline{\nabla} u \cdot \underline{\mathbf{n}}_\Gamma \right] \right).
\end{aligned}$$

In the case when i and j are on the same side of the interface, the derivation is the same but the contributions from the discontinuities vanish. The contribution from the interaction between the degrees of freedom i and j, in the case when $\phi_i > 0$, to the finite volume discretization of (1) can then be written

$$\beta_i s_{ij} \frac{u_{ij}^+ - u_i}{d_{ij}/2} = \frac{2\beta_i}{\beta_i + \beta_j} \frac{s_{ij}}{d_{ij}} \left(\beta_j u_j + \beta_i u_i - (\beta_i + \beta_j) u_i + \beta_j [u] - \frac{d_{ij}}{2} \left[\beta \underline{\nabla} u \cdot \underline{\mathbf{n}}_\Gamma \right] \right)$$

$$= \frac{2\beta_i \beta_j}{\beta_i + \beta_j} s_{ij} \frac{u_j - u_i}{d_{ij}} - \frac{2\beta_i \beta_j}{\beta_i + \beta_j} \frac{s_{ij}}{d_{ij}} \left(-[u] + \frac{d_{ij}}{2\beta_j} \left[\beta \underline{\nabla} u \cdot \underline{\mathbf{n}}_\Gamma \right] \right)$$

$$= \tilde{\beta}_{ij} s_{ij} \frac{u_j - u_i}{d_{ij}} - \tilde{\beta}_{ij} \frac{s_{ij}}{d_{ij}} \left(-[u] + \frac{d_{ij}}{2\beta_j} \left[\beta \underline{\nabla} u \cdot \underline{\mathbf{n}}_\Gamma \right] \right),$$

where $\tilde{\beta}_{ij}$ is the harmonic mean between β_i and β_j, i.e.

$$\tilde{\beta}_{ij} = \frac{|\phi_i| + |\phi_j|}{|\phi_i|/\beta_i + |\phi_j|/\beta_j} = \frac{2\beta_i \beta_j}{\beta_i + \beta_j}.$$

The contribution of the interaction between the degrees of freedom i and j to the linear system is therefore

$$\tilde{\beta}_{ij} s_{ij} \frac{u_j - u_i}{d_{ij}},$$

while the contribution to the right-hand side is

$$\tilde{\beta}_{ij} \frac{s_{ij}}{d_{ij}} \left(-[u] + \frac{d_{ij}}{2\beta_j} \left[\beta \underline{\nabla} u \cdot \underline{\mathbf{n}}_\Gamma \right] \right).$$

Note that we made use of the fact that ϕ is a distance function and u_{ij} is midway between i and j to simplify the expression. Similarly, we can derive the contribution of the interaction between i and j for the case when $\phi_i < 0$ and obtain the general expression for the interaction between any degrees of freedom i and j

$$\tilde{\beta}_{ij} s_{ij} \frac{u_j - u_i}{d_{ij}} + \texttt{Vol}(\mathcal{C}) \cdot k_i \cdot u_i = \tilde{\beta}_{ij} \frac{s_{ij}}{d_{ij}} \left(-\text{sign}(\phi_i)[u] + \frac{d_{ij}}{2\beta_j} \left[\beta \underline{\nabla} u \cdot \underline{\mathbf{n}}_\Gamma \right] \right) + \texttt{Vol}(\mathcal{C}) \cdot f_i,$$

where $\texttt{Vol}(\mathcal{C})$ is the volume of the Voronoi cell associated to the degree of freedom i and k_i and f_i are the respective values of k and f at the degree of freedom i. This discretization is entirely implicit and leads to a symmetric positive definite matrix. The discontinuities contribute only to the right-hand side of the linear system. Note that this formulation is identical to the Ghost Fluid Method of [41] in the case where the irregular interface is orthogonal to the flux between the two degrees of freedom and located midway. In fact, the Voronoi Interface Method can be interpreted as a Ghost Fluid Method where the flux between two degrees of freedom is guaranteed to be orthogonal to the face connecting their respective control volumes. We solve the linear system with the Conjugate Gradient iterative solver provided by the Petsc libraries [5,6] and preconditioned with the Hypre multigrid [20]. We enforce Dirichlet boundary conditions on $\partial\Omega$.

4. Numerical validation on uniform meshes

In this section, we validate the addition of the degrees of freedom along the irregular interface and analyze the convergence of our method on various examples. In order to demonstrate that our solver captures the discontinuities properly, all the results in this section are presented on meshes that are uniform away from the interface. Doing so, we make sure that the error on the interface dominates the overall error. Since the degrees of freedom are the seeds of the Voronoi cells, it is convenient to compute the gradient of the solution at every point located in the middle of two degrees of freedom, i.e. $\underline{\nabla} u_{ij} \cdot \underline{\mathbf{n}}_{ij} = (u_j - u_i)/d_{ij}$ where $\underline{\mathbf{n}}_{ij}$ is the normal to the edge of the Voronoi cell connecting the degrees of freedom i and j. The errors presented are normalized.

4.1. Validation of the construction of the Voronoi diagrams close to the interface

In this first example, we are interested in the influence of the quality of the mesh close to the interface. We consider three different possibilities, represented in Fig. 5 for a circular irregular interface described by $\phi(x, y) = -\sqrt{x^2 + y^2} + r_0$, with $r_0 = 0.5$, in a domain $\Omega = [-1, 1]^2$.

For the first case, we place the new degrees of freedom at regular intervals on the irregular interface, making use of the explicit parametric expression available for a circle. We choose to place $N = 1.5 \lfloor \frac{2\pi r_0}{\min(x_{min}, y_{min})} \rfloor$ new degrees of freedom on either side of the interface, at a distance $\texttt{diag}/5$ from the interface with $\texttt{diag} = \sqrt{x_{min}^2 + y_{min}^2}$. For the second case, we place the new degrees of freedom according to the procedure described in Section 2.2, at a distance $\frac{\texttt{diag}}{5}$ from the
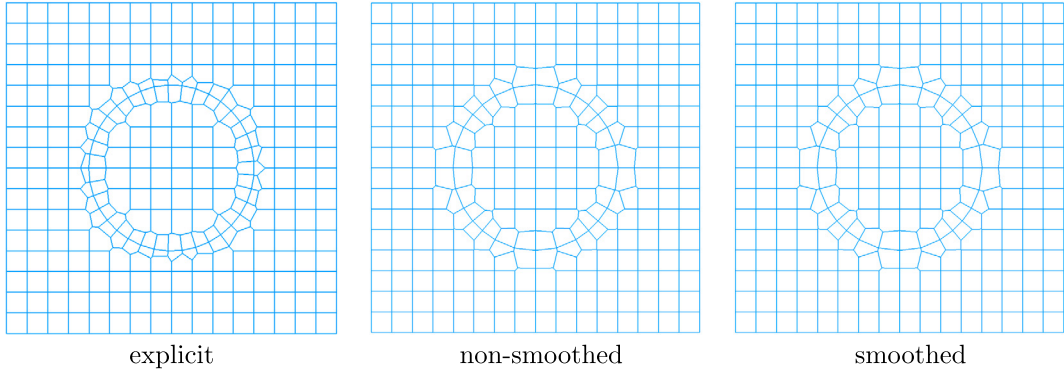
Fig. 5. Visualization of the three different meshes on a resolution $2^4 \times 2^4$ for Section 4.1.
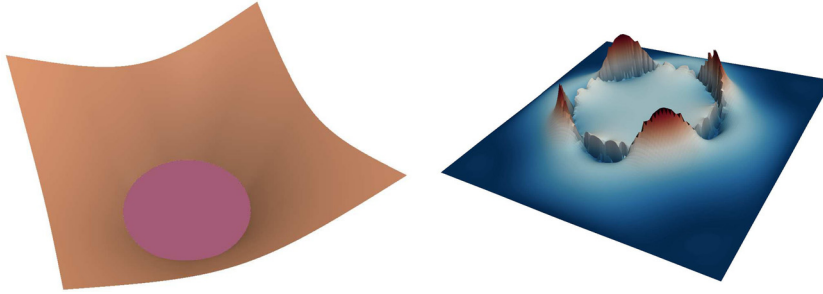

Fig. 6. Left: representation of the solution for Example 4.1. Right: visualization of the localization of the error on a non-smoothed mesh (case 2) of resolution $2^7 \times 2^7$.

Table 1

Convergence of the error on the solution in the L^∞ norm for Example 4.1. The first case corresponds to the degrees of freedom placed along the interface using the explicit parametrization, the second case corresponds to the mesh obtained following the method described in Section 2.2, and the third case is the smoothed version of case 2.

Resolution	Explicit		Non-smoothed		Smoothed	
	Solution	Order	Solution	Order	Solution	Order
2^3	$3.66 \cdot 10^{-3}$	–	$1.27 \cdot 10^{-2}$	–	$1.20 \cdot 10^{-2}$	–
2^4	$1.79 \cdot 10^{-3}$	1.03	$2.34 \cdot 10^{-3}$	2.44	$2.20 \cdot 10^{-3}$	2.45
2^5	$5.77 \cdot 10^{-4}$	1.63	$6.17 \cdot 10^{-4}$	1.92	$6.02 \cdot 10^{-4}$	1.87
2^6	$1.56 \cdot 10^{-4}$	1.89	$1.62 \cdot 10^{-4}$	1.93	$1.61 \cdot 10^{-4}$	1.91
2^7	$4.32 \cdot 10^{-5}$	1.86	$4.45 \cdot 10^{-5}$	1.87	$4.23 \cdot 10^{-5}$	1.86
2^8	$1.13 \cdot 10^{-5}$	1.94	$1.14 \cdot 10^{-5}$	1.97	$1.14 \cdot 10^{-5}$	1.96
2^9	$2.83 \cdot 10^{-6}$	1.99	$2.96 \cdot 10^{-6}$	1.95	$2.95 \cdot 10^{-6}$	1.94
2^{10}	$7.19 \cdot 10^{-7}$	1.98	$7.46 \cdot 10^{-7}$	1.99	$7.45 \cdot 10^{-7}$	1.98

interface. Finally, for the third case, we start from the partition obtained with the second case and modify it according to the procedure described in Section 2.3 to obtain a smoothed mesh.

We monitor the convergence of our method on these three meshes for the following solution taken from [64],

$$u(x, y) = \begin{cases} 1 + \log(2\sqrt{x^2 + y^2}) & \text{if } \phi(x, y) < 0, \\ 1 & \text{if } \phi(x, y) > 0, \end{cases}$$

and $\beta(x, y) = 1$. Note that for this case we have $[u] = 0$ and $[\underline{\nabla} u \cdot \underline{n}] = 2$, with continuous β and a discontinuity in the flux across the interface. A representation of the solution is given in Fig. 6 together with a visualization of the localization of the error. We report the convergence of the solver on this example for the three different meshes in Tables 1 and 2. We observe second-order convergence for the solution and first-order convergence for the gradient of the solution, and very similar errors for all three meshes. We conclude that smoothing the mesh obtained with the procedure explained in Section 2.3 does not seem to improve the accuracy of the solver.

4.2. Influence of the smoothing of the mesh

We further consider the influence of the smoothing procedure described in Section 2.3. This time, we consider an interface described by $\phi(x, y) = -\sqrt{x^2 + y^2} + r_0 + r_1 \cos(5\theta)$, with $r_0 = 0.5$, $r_1 = 0.15$ and θ the angle between (x, y) and $(1, 0)$,

Table 2
Convergence of the error on the gradient of the solution in the L^∞ norm for Example 4.1. The first case corresponds to the degrees of freedom placed along the interface using the explicit parametrization, the second case corresponds to the mesh obtained following the method described in Section 2.2, and the third case corresponds to its smoothed version.

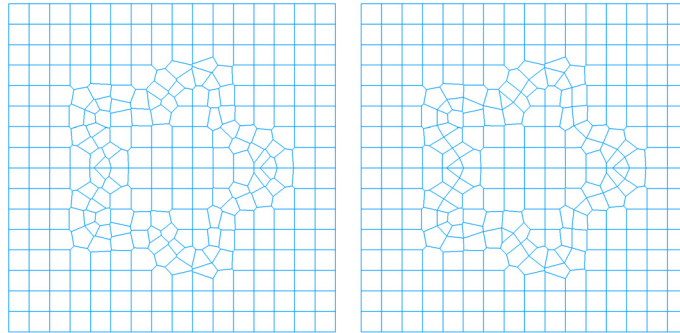| Resolution	Explicit		Non-smoothed		Smoothed	
	Gradient	Order	Gradient	Order	Gradient	Order
2^3	$4.60 \cdot 10^{-2}$	–	$4.71 \cdot 10^{-2}$	–	$4.24 \cdot 10^{-2}$	–
2^4	$3.28 \cdot 10^{-2}$	0.49	$2.40 \cdot 10^{-2}$	0.98	$2.36 \cdot 10^{-2}$	0.84
2^5	$1.79 \cdot 10^{-2}$	0.87	$1.23 \cdot 10^{-2}$	0.96	$1.23 \cdot 10^{-2}$	0.94
2^6	$9.56 \cdot 10^{-3}$	0.91	$8.34 \cdot 10^{-3}$	0.56	$8.35 \cdot 10^{-3}$	0.56
2^7	$5.15 \cdot 10^{-3}$	0.89	$3.94 \cdot 10^{-3}$	1.08	$3.94 \cdot 10^{-3}$	1.08
2^8	$2.56 \cdot 10^{-3}$	1.01	$2.12 \cdot 10^{-3}$	0.90	$2.12 \cdot 10^{-3}$	0.90
2^9	$1.30 \cdot 10^{-3}$	0.98	$1.12 \cdot 10^{-3}$	0.92	$1.12 \cdot 10^{-3}$	0.92
2^{10}	$6.61 \cdot 10^{-4}$	0.98	$5.61 \cdot 10^{-4}$	1.00	$5.61 \cdot 10^{-4}$	1.00

Fig. 7. Visualization of the non-smoothed (left) and smoothed (right) meshes on a resolution $2^4 \times 2^4$ for Section 4.2.

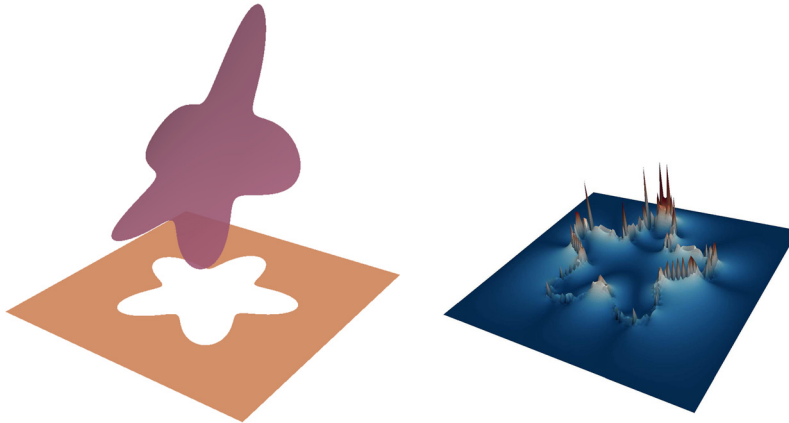**Fig. 8.** Left: representation of the solution for Example 4.2. Right: visualization of the error on a non-smoothed mesh and for a resolution $2^7 \times 2^7$.

in a domain $\Omega = [-1, 1]^2$. Since we do not have an explicit parametrization of the interface that would enable to place the degrees of freedom at regular intervals, we only consider the mesh generated form the procedure described in Section 2.2 and its smoothed version obtained by applying the procedure described in Section 2.3. Fig. 7 gives a visualization of the meshes obtained.

For this section, we choose to work with the exact solution taken from [9]

$$u(x, y) = \begin{cases} 0 & \text{if } \phi(x, y) < 0, \\ e^x \cos(y) & \text{if } \phi(x, y) > 0, \end{cases}$$

with $\beta^- = \beta^+ = 1$. The solution is represented in Fig. 8. We monitor the convergence of the solver in Table 3 and observe second-order convergence for the solution and first-order convergence for the gradient of the solution in both cases. Given that the smoothing algorithm requires additional processing and does not seem to improve the accuracy (in fact, we notice for this particular example that the non-smoothed results are more accurate), we choose to work with the non-smoothed mesh constructed as described in Section 2.2 for the remaining of this article.

Table 3
Convergence of the error on the solution and its gradient in the L^∞ norm for Example 4.2.

Resolution	Non-smoothed				Smoothed			
	Solution	Order	Gradient	Order	Solution	Order	Gradient	Order
2^3	$2.39 \cdot 10^{-3}$	–	$3.01 \cdot 10^{-2}$	–	$2.00 \cdot 10^{-2}$	–	$1.06 \cdot 10^{-1}$	–
2^4	$1.06 \cdot 10^{-3}$	1.17	$5.08 \cdot 10^{-1}$	−4.07	$1.44 \cdot 10^{-2}$	0.47	$5.27 \cdot 10^{-1}$	−2.32
2^5	$3.43 \cdot 10^{-4}$	1.63	$8.42 \cdot 10^{-3}$	5.91	$4.21 \cdot 10^{-3}$	1.78	$2.84 \cdot 10^{-2}$	4.21
2^6	$6.82 \cdot 10^{-5}$	2.33	$3.95 \cdot 10^{-3}$	1.09	$1.62 \cdot 10^{-3}$	1.38	$1.66 \cdot 10^{-2}$	0.78
2^7	$2.79 \cdot 10^{-5}$	1.29	$2.93 \cdot 10^{-3}$	0.43	$4.55 \cdot 10^{-4}$	1.83	$8.03 \cdot 10^{-3}$	1.04
2^8	$7.35 \cdot 10^{-6}$	1.92	$1.40 \cdot 10^{-3}$	1.06	$1.11 \cdot 10^{-4}$	2.03	$2.66 \cdot 10^{-3}$	1.59
2^9	$1.89 \cdot 10^{-6}$	1.96	$6.31 \cdot 10^{-4}$	1.15	$2.76 \cdot 10^{-5}$	2.01	$9.62 \cdot 10^{-4}$	1.47
2^{10}	$4.75 \cdot 10^{-7}$	1.99	$3.28 \cdot 10^{-4}$	0.94	$6.86 \cdot 10^{-6}$	2.01	$3.34 \cdot 10^{-4}$	1.53
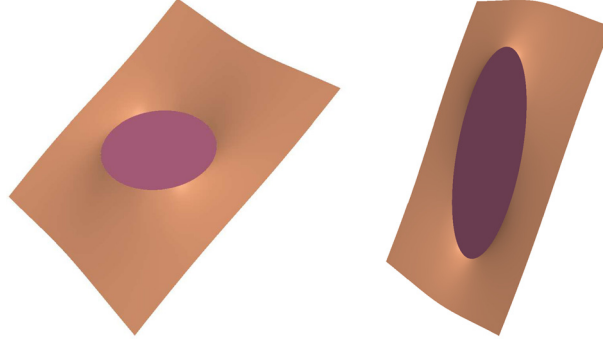
Fig. 9. Two examples of solutions for Example 4.3. Left: $\beta^- = 1$ and $\beta^+ = 10$. Right: $\beta^- = 10$ and $\beta^+ = 1$.

4.3. Example with a discontinuity in the diffusion coefficient

We now consider the exact solution

$$u(x, y) = \begin{cases} \frac{x(\rho+1) - x(\rho-1)r_0^2/r^2}{\rho+1+r_0^2(\rho-1)} & \text{if } \phi(x, y) < 0, \\ \frac{2x}{\rho+1+r_0^2(\rho-1)} & \text{if } \phi(x, y) > 0, \end{cases}$$

with $r_0 = .5$, $r = \sqrt{x^2 + y^2}$, $\phi(x, y) = -r^2 + r_0^2$ and $\rho = \beta^+/\beta^-$ in the domain $\Omega = [-1, 1]^2$. This corresponds to Example 7.3 from [64]. In this case, u is continuous, but the diffusion coefficient β experiences a large jump across the irregular interface Γ. We also have $[\underline{\nabla}u \cdot \underline{\textbf{n}}] \neq 0$. A visualization of the solution is given in Fig. 9. The gradient of the solution is given by

$$\underline{\nabla}u(x, y) = \begin{cases} \frac{1}{\rho+1+r_0^2(\rho-1)} \begin{pmatrix} \rho + 1 - r_0^2(\rho - 1)\frac{y^2-x^2}{(x^2+y^2)^2} \\ r_0^2(\rho - 1)\frac{2xy}{(x^2+y^2)^2} \end{pmatrix} & \text{if } \phi(x, y) < 0, \\ \frac{1}{\rho+1+r_0^2(\rho-1)} \begin{pmatrix} 2 \\ 0 \end{pmatrix} & \text{if } \phi(x, y) > 0. \end{cases}$$

The errors on the solution and its gradient are monitored in Table 4 which shows second-order convergence on the solution and first-order convergence on its gradient. Fig. 10 provides a visualization of the localization of the error. We also monitor the evolution of the 1-norm condition number of the matrix of the linear system as the mesh is refined in and present the results in Table 5. The condition number depends on the mesh resolution and on the diffusion coefficient. When the diffusion coefficient is large, the condition number gets large rapidly. However, the discontinuities at the interface are entirely captured by the right hand side and therefore do not affect the conditioning of the matrix.

4.4. A complete example

This example is meant to test our method to its full capacity, with discontinuities in all four quantities (the solution, its gradient, the diffusion coefficient and the flux across the interface), and with a complex irregular interface. We choose the exact solution

$$u(x, y) = \begin{cases} e^x & \text{if } \phi(x, y) < 0, \\ \cos(x)\sin(y) & \text{if } \phi(x, y) > 0, \end{cases}$$

Table 4
Convergence on the solution and its gradient for Example 4.3, for two different combinations of diffusion coefficients.

Resolution	$\beta^- = 1,\ \beta^+ = 10^5$				$\beta^- = 10^5,\ \beta^+ = 1$			
	Solution	Order	Gradient	Order	Solution	Order	Gradient	Order
2^3	$1.10 \cdot 10^{-2}$	–	$3.84 \cdot 10^{-2}$	–	$1.74 \cdot 10^{-2}$	–	$1.14 \cdot 10^{-1}$	–
2^4	$3.42 \cdot 10^{-3}$	1.68	$1.93 \cdot 10^{-2}$	1.00	$4.70 \cdot 10^{-3}$	1.89	$5.65 \cdot 10^{-2}$	1.01
2^5	$1.39 \cdot 10^{-3}$	1.30	$1.17 \cdot 10^{-2}$	0.72	$1.26 \cdot 10^{-3}$	1.90	$2.54 \cdot 10^{-2}$	1.15
2^6	$3.82 \cdot 10^{-4}$	1.86	$6.19 \cdot 10^{-3}$	0.92	$3.39 \cdot 10^{-4}$	1.90	$1.77 \cdot 10^{-2}$	0.53
2^7	$1.34 \cdot 10^{-4}$	1.51	$3.59 \cdot 10^{-3}$	0.79	$1.05 \cdot 10^{-4}$	1.69	$8.06 \cdot 10^{-3}$	1.13
2^8	$3.43 \cdot 10^{-5}$	1.97	$1.84 \cdot 10^{-3}$	0.96	$2.67 \cdot 10^{-5}$	1.98	$4.87 \cdot 10^{-3}$	0.73
2^9	$9.76 \cdot 10^{-6}$	1.81	$1.05 \cdot 10^{-3}$	0.82	$7.55 \cdot 10^{-6}$	1.82	$2.40 \cdot 10^{-3}$	1.02
2^{10}	$2.58 \cdot 10^{-6}$	1.92	$5.41 \cdot 10^{-4}$	0.95	$1.96 \cdot 10^{-6}$	1.95	$1.25 \cdot 10^{-3}$	0.94

Table 5
Evolution of the condition number as the mesh is refined for Example 4.3. The condition number depends solely on the diffusion coefficient β and on the resolution on the mesh since the discontinuities are captured by the right hand side of the linear system.

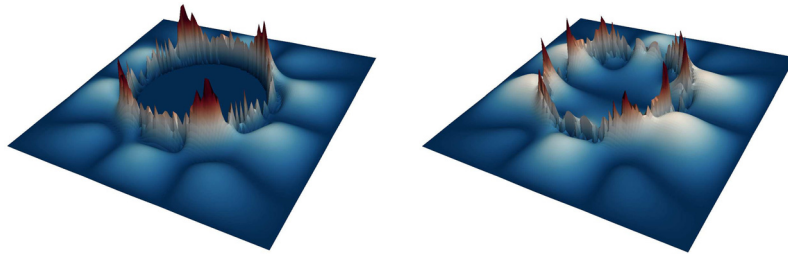| Resolution	$\beta^- = 1,\ \beta^+ = 10^5$	$\beta^- = 10^5,\ \beta^+ = 1$	$\beta^- = \beta^+ = 10^5$	$\beta^- = \beta^+ = 1$
2^3	$2.38 \cdot 10^6$	$2.02 \cdot 10^6$	$5.59 \cdot 10^6$	$1.72 \cdot 10^2$
2^4	$1.17 \cdot 10^7$	$5.09 \cdot 10^6$	$1.23 \cdot 10^7$	$8.11 \cdot 10^2$
2^5	$4.86 \cdot 10^7$	$1.57 \cdot 10^7$	$2.69 \cdot 10^7$	$3.66 \cdot 10^3$
2^6	$2.09 \cdot 10^8$	$7.18 \cdot 10^7$	$5.72 \cdot 10^7$	$1.58 \cdot 10^4$
2^7	$8.52 \cdot 10^8$	$3.14 \cdot 10^8$	$1.19 \cdot 10^8$	$6.59 \cdot 10^4$
2^8	$3.86 \cdot 10^9$	$1.32 \cdot 10^9$	$2.42 \cdot 10^8$	$2.70 \cdot 10^5$
2^9	$1.48 \cdot 10^{10}$	$5.43 \cdot 10^9$	$4.89 \cdot 10^8$	$1.09 \cdot 10^6$
2^{10}	$6.59 \cdot 10^{10}$	$2.20 \cdot 10^{10}$	$9.84 \cdot 10^8$	$4.39 \cdot 10^6$

Fig. 10. Visualization of the localization of the error in the L^∞ norm on a grid of resolution $2^7 \times 2^7$ for Example 4.3. Left: $\beta^- = 1$ and $\beta^+ = 10^5$. Right: $\beta^- = 10^5$ and $\beta^+ = 1$.

in a domain $\Omega = [-1, 1]^2$ with $\phi(x, y) = -\sqrt{x^2 + y^2} + r_0 + r_1 \cos(n\theta)$, where $r_0 = 0.5$, $r_1 = 0.15$ and $n = 5$, and we define the diffusion coefficient as

$$\beta(x, y) = \begin{cases} y^2 \ln(x + 2) + 4 & \text{if } \phi(x, y) < 0, \\ e^{-y} & \text{if } \phi(x, y) > 0. \end{cases}$$

Note that with our method, each degree of freedom has a control volume that is entirely on one side of the irregular interface, and therefore we can easily define a forcing term for any analytical solution. The exact solution and the diffusion coefficient are represented in Fig. 11. The convergence is summarized in Table 6 and once again indicates second-order convergence for the solution and first-order convergence for the gradient of the solution. A visualization of the localization of the error on the solution is given in Fig. 12.

Fig. 13 presents the percentage of the runtime consumed by the four principal components of the algorithm, i.e constructing the Voronoi mesh, assembling the matrix, computing the right hand side and solving the linear system. For coarse grids, the bottleneck of the computation is the construction of the Voronoi mesh, but for high resolution we observe that inverting the linear system is the costliest. These results correspond to our implementation in the absence of parallelization. The finest resolution of 1024×1024 takes 18 s on a single core of an Intel i7-2600 3.40 GHz cpu.

4.5. Adding subdomains

We now propose an example with multiple subdomains. Note that the method we propose leads naturally to a linear system with N rows, the number of degrees of freedom. This number increases slightly as the number of subdomains increases and additional degrees of freedom are added next to the irregular interfaces.

Fig. 11. Left: visualization of the solution u for Example 4.4. Right: visualization of the diffusion coefficient β.
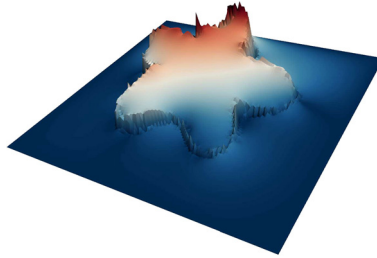

Fig. 12. Visualization of the localization of the error for Example 4.4 on a resolution of $2^7 \times 2^7$.

Table 6
Convergence on the solution and its gradient for Example 4.4.

Resolution	Solution	Order	Gradient	Order
2^3	$3.97 \cdot 10^{-3}$	–	$4.37 \cdot 10^{-1}$	–
2^4	$9.98 \cdot 10^{-4}$	1.99	$5.01 \cdot 10^{-1}$	−0.20
2^5	$2.90 \cdot 10^{-4}$	1.78	$3.67 \cdot 10^{-3}$	7.09
2^6	$8.84 \cdot 10^{-5}$	1.71	$1.73 \cdot 10^{-3}$	1.09
2^7	$2.06 \cdot 10^{-5}$	2.10	$9.84 \cdot 10^{-4}$	0.81
2^8	$5.22 \cdot 10^{-6}$	1.98	$4.77 \cdot 10^{-4}$	1.05
2^9	$1.33 \cdot 10^{-6}$	1.97	$2.45 \cdot 10^{-4}$	0.96
2^{10}	$3.39 \cdot 10^{-7}$	1.97	$1.23 \cdot 10^{-4}$	0.99

For simplicity, we choose to work with non-intersecting irregular interfaces, but our method is suited for any general configuration. We divide the computational domain $\Omega = [-1, 1]^2$ in 4 subdomains represented in Fig. 14 and separated by the three contours defined by

$$\Gamma^0 = \left\{ (x, y), \ \phi^0(x, y) = \sqrt{x^2 + y^2} - 0.2 \right\},$$

$$\Gamma^1 = \left\{ (x, y), \ \phi^1(x, y) = \sqrt{x^2 + y^2} - 0.5 + 0.1 \cos(5\theta) \right\},$$

$$\Gamma^2 = \left\{ (x, y), \ \phi^2(x, y) = \sqrt{x^2 + y^2} - 0.8 \right\},$$

where θ is the angle between (x, y) and the x-axis. We choose the exact solution

$$u(x, y) = \begin{cases} e^x + 1.3 & \text{if } (x, y) \in \Omega^0, \\ \cos(y) + 1.8 & \text{if } (x, y) \in \Omega^1, \\ \sin(x) + 0.5 & \text{if } (x, y) \in \Omega^2, \\ -x + \ln(y + 2) & \text{if } (x, y) \in \Omega^3, \end{cases}$$
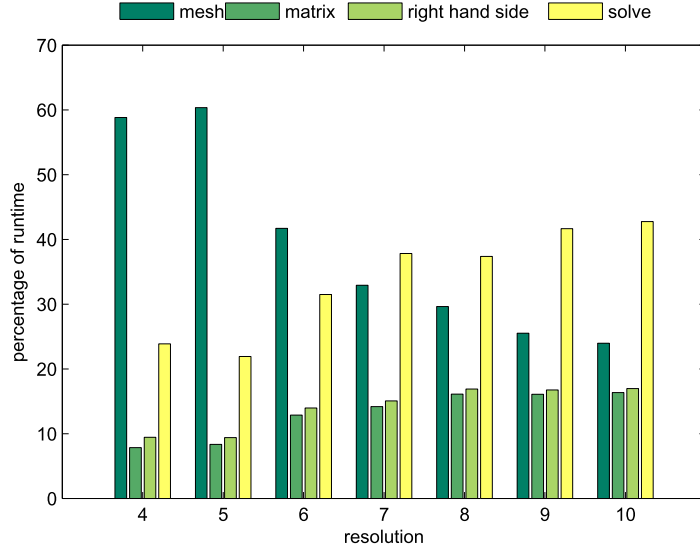
Fig. 13. Representation of the computation time consumed by the four main sections of our implementation, constructing the mesh, assembling the matrix, computing the right hand side and solving the linear system for Example 4.4. For coarse grids, building the Voronoi partition takes the most time, but as the resolution of the grid increases, the inversion of the linear system becomes the costliest.
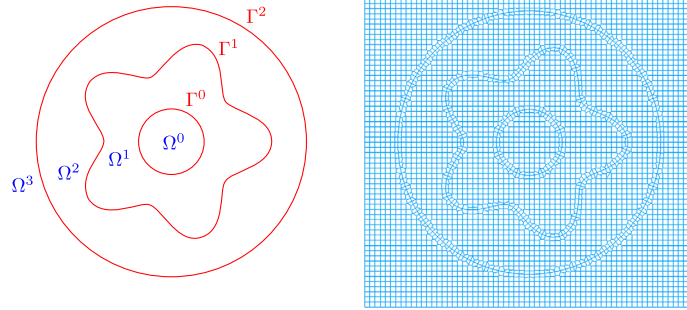

Fig. 14. Illustration of the division of the computational domain into four subdomains, together with the Voronoi mesh generated, for Example 4.5.

and the diffusion coefficient

$$\beta(x, y) = \begin{cases} y^2 + 1 & \text{if } (x, y) \in \Omega^0, \\ e^x & \text{if } (x, y) \in \Omega^1, \\ y + 1 & \text{if } (x, y) \in \Omega^2, \\ x^2 + 1 & \text{if } (x, y) \in \Omega^3. \end{cases}$$

The solution and the diffusion coefficient are represented in Fig. 15.

The convergence of the solver is presented in Table 7. We observe second order convergence for the solution and first order convergence for its gradient.

4.6. Application to three spatial dimensions

We now present an example in three spatial dimensions and with a spherical interface of radius 0.5 in a domain $\Omega = [-1, 1]^3$. We work with the exact solution

$$u(x, y, z) = \begin{cases} e^z & \text{if } \phi(x, y, z) < 0, \\ \cos(x) \sin(y) & \text{if } \phi(x, y, z) > 0, \end{cases}$$

and the diffusion coefficient

$$\beta(x, y, z) = \begin{cases} y^2 \ln(x + 2) + 4 & \text{if } \phi(x, y, z) < 0, \\ e^{-z} & \text{if } \phi(x, y, z) > 0. \end{cases}$$

The geometry, together with a slice of the solution and of the diffusion coefficient, is represented in Fig. 16. Table 8 presents the numerical results and indicates second-order convergence for the solution and first-order convergence for its gradient.

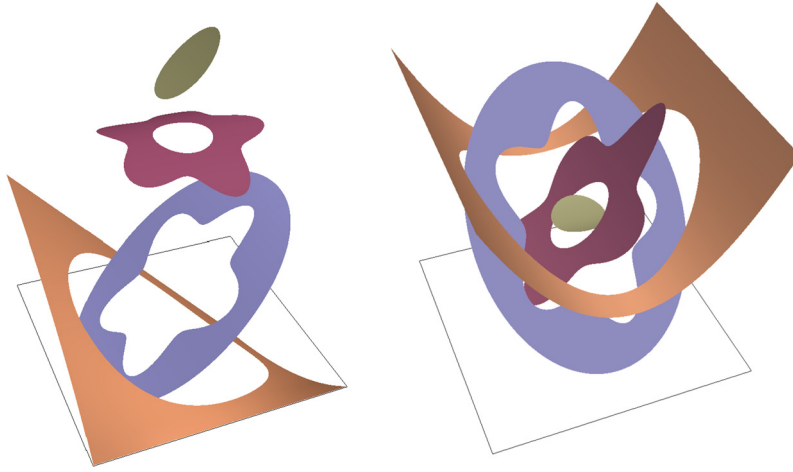**Fig. 15.** Visualization of the solution (left) and the diffusion coefficient (right) for Example 4.5.

Table 7
Convergence on the solution and its gradient in the L^∞ norm for Example 4.5.

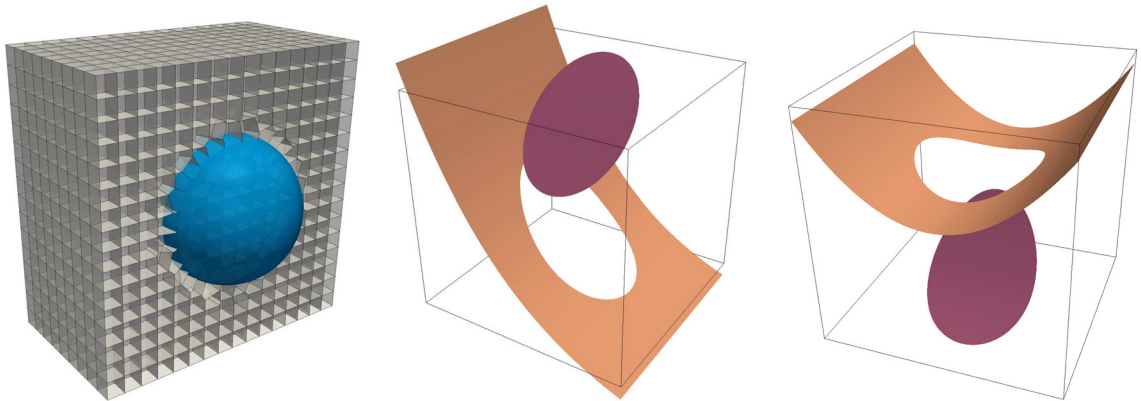| Resolution	Solution	Order	Gradient	Order
2^4	$1.00 \cdot 10^{-3}$	–	$3.33 \cdot 10^{-3}$	–
2^5	$2.33 \cdot 10^{-4}$	2.11	$1.01 \cdot 10^{-3}$	1.72
2^6	$6.23 \cdot 10^{-5}$	1.90	$3.46 \cdot 10^{-4}$	1.54
2^7	$1.56 \cdot 10^{-5}$	2.00	$1.59 \cdot 10^{-4}$	1.12
2^8	$4.00 \cdot 10^{-6}$	1.96	$6.82 \cdot 10^{-5}$	1.22
2^9	$1.01 \cdot 10^{-6}$	1.99	$4.00 \cdot 10^{-5}$	0.77
2^{10}	$2.55 \cdot 10^{-7}$	1.99	$2.24 \cdot 10^{-5}$	0.84

Fig. 16. Left: representation of the irregular interface and the associated Voronoi mesh on a resolution $2^4 \times 2^4 \times 2^4$ for Example 4.6. Center: visualization of the solution on the slice $x = 0$. Right: visualization of the diffusion coefficient on the slice $x = 0$. Note that the surfaces have been translated to facilitate the visualization.

Table 8
Convergence on the solution and its gradient in the L^∞ norm on a sphere (Example 4.6).

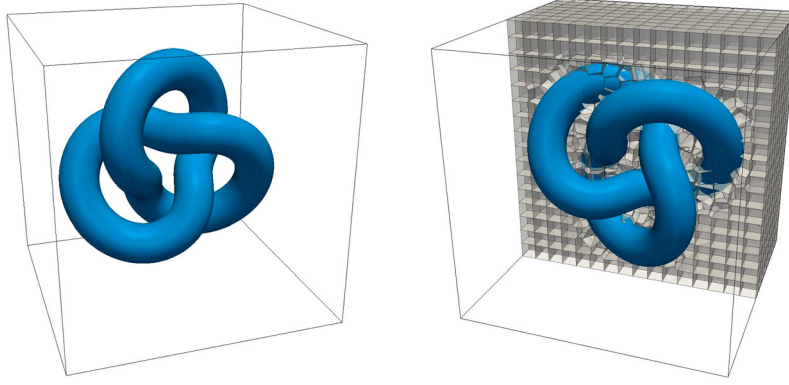| Resolution	Solution	Order	Gradient	Order
2^3	$3.61 \cdot 10^{-3}$	–	$1.13 \cdot 10^{-2}$	–
2^4	$1.21 \cdot 10^{-3}$	1.58	$7.69 \cdot 10^{-3}$	0.56
2^5	$3.04 \cdot 10^{-4}$	1.99	$3.83 \cdot 10^{-3}$	1.01
2^6	$7.74 \cdot 10^{-5}$	1.98	$2.43 \cdot 10^{-3}$	0.66
2^7	$1.97 \cdot 10^{-5}$	1.98	$1.24 \cdot 10^{-3}$	0.98

Fig. 17. Representation of the irregular interface and the associated Voronoi mesh on a resolution $2^4 \times 2^4 \times 2^4$ for Example 4.7.

4.7. A complex geometry in three spatial dimensions

For a more complicated geometry, we select the intricate contour borrowed from [34] and parametrized by

$$\Gamma_{\text{trefoil}} = \left\{ \frac{R}{3} \begin{pmatrix} (2 + \cos(3t)) \cos(2t) \\ (2 + \cos(3t)) \sin(2t) \\ \sin(3t) \end{pmatrix}, t \in [0, 2\pi] \right\},$$

where $R = 0.7$ is the major radius of the trefoil. We then define

$$\Omega^+ = \left\{ \underline{\boldsymbol{x}} \in \mathbb{R}, \min_{\underline{\boldsymbol{y}} \in \Gamma_{\text{trefoil}}} \|\underline{\boldsymbol{x}} - \underline{\boldsymbol{y}}\|_2 < r \right\},$$

with $r = 0.15$ the minor radius of the trefoil. The contour is represented in Fig. 17.

For this example, we use the exact solution

$$u(x, y, z) = \begin{cases} yz \sin(x) & \text{if } \phi(x, y, z) < 0, \\ xy^2 + z^3 & \text{if } \phi(x, y, z) > 0, \end{cases}$$

and the diffusion coefficient

$$\beta(x, y, z) = \begin{cases} y^2 + 1 & \text{if } \phi(x, y, z) < 0, \\ e^{x+z} & \text{if } \phi(x, y, z) > 0. \end{cases}$$

Slices of the solution and of the diffusion coefficient are displayed in Fig. 18. The convergence of our method on this complex irregular interface is reported in Table 9 and once more indicates second order convergence for the solution and first order convergence for its gradient.

4.8. The screened Poisson equation

This example and the following one add a non-zero k to the previous Example 4.7. In this example, we choose $k < 0$ as

$$k(x, y, z) = \begin{cases} -e^x & \text{if } \phi(x, y, z) < 0, \\ -\cos(y) \sin(z) - 2 & \text{if } \phi(x, y, z) > 0. \end{cases}$$

The convergence results are presented in Table 10 and show second order convergence with errors very similar to the ones obtained when $k = 0$.

4.9. The Helmholtz equation case

Our last example on a uniform base mesh is exactly the same than the one from the previous section but for the Helmholtz equation case, i.e. $k(x, y, z) > 0$. We set

$$k(x, y, z) = \begin{cases} e^y & \text{if } \phi(x, y, z) < 0, \\ \cos(x) \sin(z) + 2 & \text{if } \phi(x, y, z) > 0. \end{cases}$$

In this case, the linear system obtained is more complicated to solve because the matrix is no longer diagonally dominant, meaning that the problem is not convex and iterative solvers such as the Conjugate Gradient used so far are not guaranteed to converge. Instead, we solve the linear directly with an LU decomposition. The numerical results are presented in Table 11 and are almost identical to the results from the previous section, illustrating the second order convergence of our method for the Helmholtz equation.
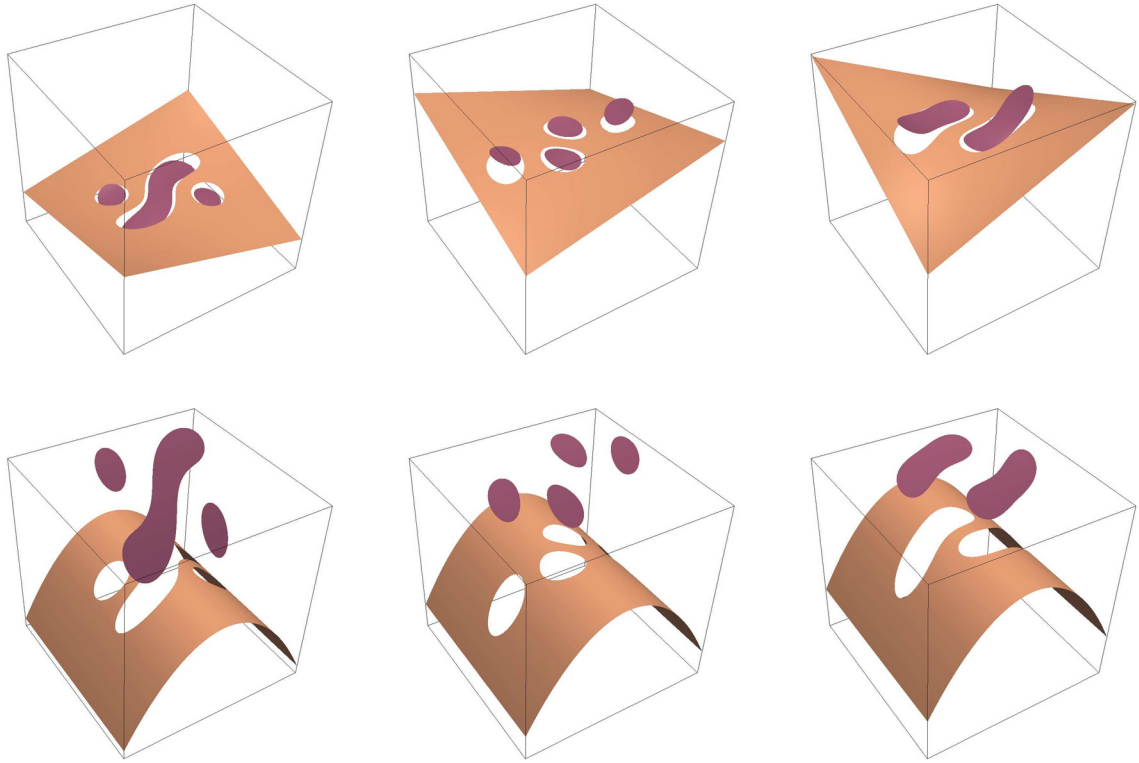
Fig. 18. Visualization of the solution (top row) and the diffusion coefficient (bottom row) on three slices for Example 4.7. The slices are taken, from left to right, at $x = 0.3$, $x = -0.3$ and $x = -0.5$. Note that the surfaces of the diffusion coefficient have been translated to facilitate the visualization.

Table 9
Convergence on the solution and its gradient in the L^∞ norm on a complex three-dimensional contour (Example 4.7).

Resolution	Solution	Order	Gradient	Order
2^4	$3.89 \cdot 10^{-3}$	–	$1.44 \cdot 10^{-1}$	–
2^5	$1.34 \cdot 10^{-3}$	1.54	$2.56 \cdot 10^{-2}$	2.67
2^6	$3.45 \cdot 10^{-4}$	1.96	$9.75 \cdot 10^{-3}$	1.21
2^7	$8.25 \cdot 10^{-5}$	2.07	$5.04 \cdot 10^{-3}$	0.95

Table 10
Convergence on the solution and its gradient in the L^∞ norm on a complex three-dimensional contour for the screened Poisson equation (Example 4.8).

Resolution	Solution	Order	Gradient	Order
2^4	$3.87 \cdot 10^{-3}$	–	$1.44 \cdot 10^{-1}$	–
2^5	$1.34 \cdot 10^{-3}$	1.53	$2.25 \cdot 10^{-2}$	2.67
2^6	$3.44 \cdot 10^{-4}$	1.96	$9.74 \cdot 10^{-3}$	1.21
2^7	$8.22 \cdot 10^{-5}$	2.06	$5.04 \cdot 10^{-3}$	0.95

Table 11
Convergence on the solution and its gradient in the L^∞ norm on a complex three-dimensional contour for the Helmholtz equation (Example 4.9).

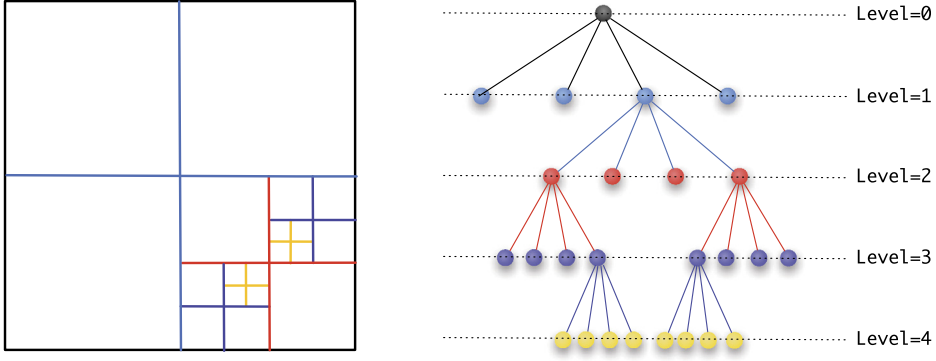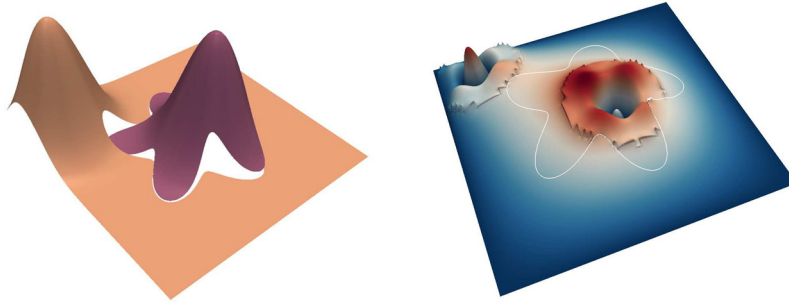| Resolution	Solution	Order	Gradient	Order
2^4	$3.91 \cdot 10^{-3}$	–	$1.44 \cdot 10^{-1}$	–
2^5	$1.35 \cdot 10^{-3}$	1.54	$2.26 \cdot 10^{-2}$	2.67
2^6	$3.46 \cdot 10^{-4}$	1.96	$9.75 \cdot 10^{-3}$	1.21
2^7	$8.27 \cdot 10^{-5}$	2.06	$5.05 \cdot 10^{-3}$	0.95

Fig. 19. Example of a Quadtree grid.

Fig. 20. Left: visualization of the solution for Example 5.2 with $\alpha = 10$. Higher values of α narrow the peaks. Right: representation of the error interpolated on the base Quadtree mesh of level 8/10 for $\alpha = 50$.

5. Extension to adaptive meshes

In the previous section we demonstrated the efficiency of our method based on uniform meshes in both two and three spatial dimensions. However, it can be applied straightforwardly to any mesh and in this section we propose an implementation on Quad/Oc-trees.

5.1. Introduction to the Quad/Oc-tree data structure

A Quad/Oc-tree grid refers to a Cartesian grid that uses the Quad/Oc-tree data structure for its storage in two/three spatial dimensions. Starting from a root cell corresponding to the entire domain, four (eight in three spatial dimensions) children are created if the cell satisfies a given splitting criterion. The process is iterated recursively until the maximum allowed level is reached. The root cell has level 0 and the finest cells have the maximum level allowed. We denote a tree with coarsest level n and finest level m by level n/m. The process is illustrated in Fig. 19. This data structure provides an $O(\ln(n))$ access to the data stored at the leaves. We refer the reader to [45] for further details on the Quad/Oc-tree data structure and the associated discretization techniques.

Since this article is a proof of concept and we know the exact solution in all the numerical example we propose, we make use of this knowledge in the construction of the tree. We will use solutions of the form

$$u(\underline{\pmb{x}}) = e^{-\alpha \|\underline{\pmb{x}} - \underline{\pmb{x}}_0\|_2^2},$$

and any given leaf $\mathcal{L}$ of the tree with center coordinates $\underline{\pmb{x}}_c$ is split if $\|\underline{\pmb{x}}_0 - \underline{\pmb{x}}_c\|_2 < \lambda \cdot \mathtt{diag}$, where $\mathtt{diag}$ is the length of the diagonal of $\mathcal{L}$ and λ controls the spread of the mesh around the peaks of the solution.

5.2. Solution on a Quadtree mesh

For this example, we consider the exact solution, represented in Fig. 20,

$$u(x, y) = \begin{cases} e^{-50((x+0.7)^2 + (y-0.7)^2)} & \text{if } \phi(x, y) < 0, \\ e^{-50((x-0.1)^2 + (y+0.1)^2)} & \text{if } \phi(x, y) > 0, \end{cases}$$

with $\beta^- = \beta^+ = 1$ and on the domain $\Omega = [-1, 1]^2$. We use the same geometry as in Section 4.4, i.e. $\phi(x, y) = -\sqrt{x^2 + y^2} + 0.5 + 0.15 \cos(5\theta)$.
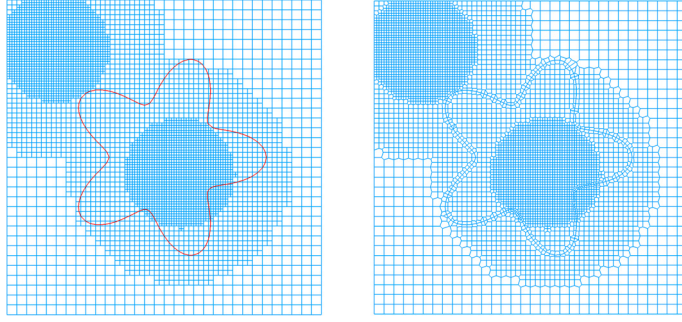
Fig. 21. Left: the initial level 5/7 Quadtree mesh generated for Example 5.2. The irregular interface Γ is represented in red. Right: the corresponding Voronoi mesh. Note that the mesh is not uniform along Γ. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 12
Convergence on the solution and its gradient in the L^∞ norm on a Quadtree-based Voronoi mesh (Example 5.2).

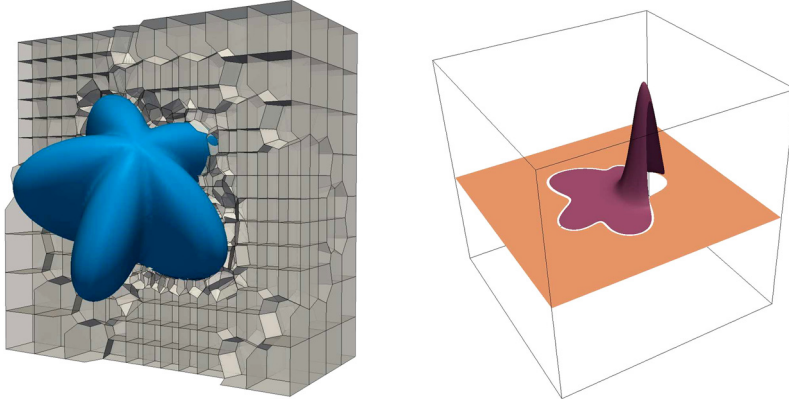| Resolution	Solution	Order	Gradient	Order
5/7	$2.56 \cdot 10^{-3}$	–	$8.89 \cdot 10^{-3}$	–
6/8	$6.48 \cdot 10^{-4}$	1.98	$5.14 \cdot 10^{-3}$	0.79
7/9	$1.69 \cdot 10^{-4}$	1.94	$2.52 \cdot 10^{-3}$	1.03
8/10	$4.30 \cdot 10^{-5}$	1.97	$1.33 \cdot 10^{-3}$	0.92
9/11	$1.09 \cdot 10^{-5}$	1.99	$6.84 \cdot 10^{-4}$	0.96

Fig. 22. Left: representation of the geometry together with the Voronoi mesh on the initial mesh of resolution 3/5 for Example 5.3. Note that the mesh is not uniform along Γ. Right: visualization of the solution on the $y = -0.1$ slice and after applying three successive refinement operations to the original mesh.

We start by constructing a mesh of level 5/7 with the criteria described in the previous section and with $\lambda = 8$. A visualization of this initial mesh is given in Fig. 21. Note that the mesh is not uniform close to the interface. We then split every cell of the mesh to monitor the convergence of the solver. The results are presented in Table 12 and show second-order convergence for the solution and first-order convergence for its gradient.

5.3. Solution on an Octree mesh

For this last example, we work with the exact solution

$$u(x, y, z) = \begin{cases} e^{-50((x+0.7)^2+(y-0.7)^2+(z-.5)^2)} & \text{if } \phi(x, y, z) < 0, \\ e^{-50((x-0.1)^2+(y+0.1)^2+(z+.3)^2)} & \text{if } \phi(x, y, z) > 0, \end{cases}$$

and on the irregular interface described by

$$\phi(x, y, z) = -\sqrt{x^2 + y^2 + z^2} + r_0 + r_1 \cos(5\theta) \cos(\frac{2\pi}{r_0} z),$$

with $r_0 = 0.25$ and $r_1 = r_0/3$, and rotated around the z-axis, y-axis and x-axis by respectively 0.6 radians, 0.2 radians and 0.9 radians. We set $\beta^- = \beta^+ = 1$. The geometry and the Voronoi mesh generated for the initial level 3/5 are represented

Table 13

Convergence on the solution and its gradient in the L^∞ norm on star shaped irregular interface and on an Octree base mesh (Example 5.3).

Resolution	Solution	Order	Gradient	Order
3/5	$6.76 \cdot 10^{-2}$	–	$6.54 \cdot 10^{-2}$	–
4/6	$1.47 \cdot 10^{-2}$	2.20	$1.97 \cdot 10^{-2}$	1.73
5/7	$3.58 \cdot 10^{-3}$	2.04	$7.04 \cdot 10^{-3}$	1.49
6/8	$8.96 \cdot 10^{-4}$	2.00	$4.24 \cdot 10^{-3}$	0.73

in Fig. 22. The convergence is presented in Table 13 and shows second-order convergence for the solution and first-order convergence for its gradient.

6. Summary

We have presented a novel fully implicit approach based on Voronoi diagrams for solving an Elliptic equation with discontinuities in the solution, its gradient, the diffusion coefficient and the flux across an irregular interface. The interface was captured through a level-set framework, and the equation was discretized with a finite volumes approach on the local Voronoi mesh. The contributions from the discontinuities were included naturally in the right-hand-side of the linear system, preserving its positive symmetric definiteness. We demonstrated second-order convergence of the solution and first-order convergence of its gradient, in the L^∞ norm, in both two and three spatial dimensions and on both uniform and adaptive Quad/Oc-tree base meshes. Large ratios in the diffusion coefficients are readily considered. We believe this approach could be utilized for numerous physical applications including those mentioned in the introduction.

Acknowledgement

References

[1] L. Adams, T. Chartier, New geometric immersed interface multigrid solvers, SIAM J. Sci. Comput. 25 (2004) 1516–1533.
[2] L. Adams, T. Chartier, A comparison of algebraic multigrid and geometric immersed interface multigrid methods for interface problems, SIAM J. Sci. Comput. 26 (2005) 762–784.
[3] L. Adams, Z. Li, The immersed interface/multigrid methods for interface problems, SIAM J. Sci. Comput. 24 (2002) 463–479.
[4] I. Babuška, The finite element method for elliptic equations with discontinuous coefficients, Computing 5 (1970) 207–213.
[5] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Users Manual, Argonne National Laboratory, 2012.
[6] S. Balay, J. Brown, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc web page, 2012.
[7] Z. Bao, J.-M. Hong, J. Teran, R. Fedkiw, Fracturing rigid materials, IEEE Trans. Vis. Comput. Graph. 13 (2007) 370–378.
[8] T. Belytschko, N. Moës, S. Usui, C. Parimi, Arbitrary discontinuities in finite elements, Int. J. Numer. Methods Eng. 50 (2001) 993–1013.
[9] P.A. Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, J. Comput. Phys. 197 (2004) 364–386.
[10] J. Bramble, J. King, A finite element method for interface problems in domains with smooth boundaries and interfaces, Adv. Comput. Math. 6 (1996) 109–138.
[11] T. Chen, J. Strain, Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems, J. Comput. Phys. 227 (16) (2008) 7503–7542.
[12] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math. 79 (1996) 175–202.
[13] M. Cisternino, L. Weynans, A parallel second order Cartesian method for elliptic interface problems, Commun. Comput. Phys. 12 (2012) 1562–1587.
[14] A. Coco, G. Russo, Second order multigrid methods for elliptic problems with discontinuous coefficients on an arbitrary interface, I: one dimensional problems, Numer. Math., Theory Methods Appl. 5 (19) (2012).
[15] A. Coco, G. Russo, Finite-difference ghost-point multigrid methods on cartesian grids for elliptic problems in arbitrary domains, J. Comput. Phys. 241 (0) (2013) 464–501.
[16] R. Crockett, P. Colella, D. Graves, A cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions, J. Comput. Phys. 230 (7) (2011) 2451–2469.
[17] C. Daux, N. Moës, J. Dolbow, N. Sukumar, T. Belytschko, Arbitrary branched and intersecting cracks with the extended finite element method, Int. J. Numer. Methods Eng. 48 (2000) 1741–1760.
[18] J. Dolbow, I. Harari, An efficient finite element method for embedded interface problems, Int. J. Numer. Methods Eng. 78 (2009) 229–252.
[19] M. Dryja, A Neumann–Neumann algorithm for mortar discretization of elliptic problems with discontinuous coefficients, Numer. Math. 99 (2005) 645–656.
[20] R.D. Falgout, U.M. Yang, Hypre: a library of high performance preconditioners, in: P.M. Sloot, A.G. Hoekstra, C.K. Tan, J.J. Dongarra (Eds.), Computational Science – ICCS 2002, in: Lecture Notes in Computer Science, vol. 2331, Springer, Berlin, Heidelberg, 2002, pp. 632–641.
[21] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[22] T. Fries, T. Belytschko, The intrinsic XFEM: a method for arbitrary discontinuities without additional unknowns, Int. J. Numer. Methods Eng. 68 (2006) 1358–1385.
[23] E. Ewing, Z. Li, T. Lin, Y. Lin, The immersed finite volume element methods for the elliptic interface problems, Math. Comput. Simul. 50 (1999) 43–61.

[24] F. Gibou, R. Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. 202 (2005) 577–601.
[25] F. Gibou, R. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (2002) 205–227.
[26] F. Gibou, C. Min, R. Fedkiw, High resolution sharp computational methods for elliptic and parabolic problems in complex geometries, J. Sci. Comput. 54 (2013) 369–413.
[27] Y. Gong, B. Li, Z. Li, Immersed-interface finite-element methods for elliptic interface problems with non-homogeneous jump conditions, SIAM J. Numer. Anal. 46 (2008) 472–495.
[28] S. Groß, A. Reusken, An extended pressure finite element space for two-phase incompressible flows with surface tension, J. Comput. Phys. 224 (2007) 40–58.
[29] G. Guyomarch, C.-O. Lee, K. Jeon, A discontinuous Galerkin method for elliptic interface problems with application to electroporation, Commun. Numer. Methods Eng. 25 (2009) 991–1008.
[30] A. Hansbo, P. Hansbo, A finite element method for the simulation of strong and weak discontinuities in solid mechanics, Comput. Methods Appl. Mech. Eng. 1993 (2004) 3523–3540.
[31] S. Hou, W. Wang, L. Wang, Numerical method for solving matrix coefficient elliptic equation with sharp-edged interfaces, J. Comput. Phys. 229 (19) (2010) 7162–7179.
[32] J. Huang, J. Zou, A mortar element method for elliptic problems with discontinuous coefficients, IMA J. Numer. Anal. 22 (2001) 549–576.
[33] H. Ji, J. Dolbow, On strategies for enforcing interfacial constraints and evaluating jump conditions with extended finite element method, Int. J. Numer. Methods Eng. 61 (2004) 2508–2535.
[34] J.L.H. Jr., L. Wang, E. Sifakis, J.M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions, J. Comput. Phys. 231 (4) (2012) 2015–2048.
[35] B. Lamichhane, B. Wohlmuth, Mortar finite elements for interface problems, Computing 72 (2004) 333–348.
[36] R. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.
[37] A.J. Lew, G.C. Buscaglia, A discontinuous-Galerkin-based immersed boundary method, Int. J. Numer. Methods Eng. 76 (2008) 427–454.
[38] Z. Li, A fast iterative algorithm for elliptic interface problems, SIAM J. Numer. Anal. 35 (1998) 230–254.
[39] Z. Li, K. Ito, The Immersed Interface Method – Numerical Solutions of PDEs Involving Interfaces and Irregular Domains, SIAM Frontiers in Applied Mathematics, vol. 33, 2006.
[40] Z. Li, T. Lin, X. Wu, New cartesian grid methods for interface problems using the finite element formulation, Numer. Math. 96 (2003) 61–98.
[41] X.D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 154 (2000) 151.
[42] X.-D. Liu, T. Sideris, Convergence of the ghost–fluid method for elliptic equations with interfaces, Math. Comput. 72 (2003) 1731–1746.
[43] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numer. Anal. 21 (1984) 285–299.
[44] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids, J. Comput. Phys. 219 (2) (Dec. 2006) 912–929.
[45] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, J. Comput. Phys. 225 (1) (2007) 300–321.
[46] M. Latige, T. Colin, G. Gallice, A second order cartesian finite volume method for elliptic interface and embedded Dirichlet problems, Comput. Fluids 83 (2013) 70–76.
[47] N. Moës, M. Cloirec, P. Cartraud, J. Remacle, A computational approach to handle complex microstructure geometries, Comput. Methods Appl. Mech. Eng. 192 (2003) 3162–3177.
[48] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, Int. J. Numer. Methods Eng. 46 (1999) 131–150.
[49] N. Molino, J. Bao, R. Fedkiw, A virtual node algorithm for changing mesh topology during simulation, ACM Trans. Graph.: SIGGRAPH Proc. 23 (2004) 385–392.
[50] Y.T. Ng, C. Min, F. Gibou, An efficient fluid–solid coupling algorithm for single-phase flows, J. Comput. Phys. 228 (23) (Dec. 2009) 8807–8829.
[51] M. Oevermann, C. Scharfenberg, R. Klein, A sharp interface finite volume method for elliptic equations on Cartesian grids, J. Comput. Phys. 228 (2009) 5184–5206.
[52] S. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49.
[53] J. Papac, F. Gibou, C. Ratsch, Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions, J. Comput. Phys. 229 (3) (Feb. 2010) 875–889.
[54] J. Papac, A. Helgadottir, C. Ratsch, F. Gibou, A level set approach for diffusion and Stefan-type problems with Robin boundary conditions on Quadtree/Octree adaptive cartesian grids, J. Comput. Phys. (2012).
[55] J.W. Purvis, J.E. Burkhalter, Prediction of critical Mach number for store configurations, AIAA J. 17 (1979) 1170–1177.
[56] C. Richardson, J. Hegemann, E. Sifakis, J. Hellrung, J. Teran, An XFEM method for modeling geometrically elaborate crack propagation in brittle materials, Int. J. Numer. Methods Eng. 88 (2011) 1042–1065.
[57] C.H. Rycroft, Voro++: a three-dimensional Voronoi cell library in C++, Chaos 19 (2009).
[58] Sandia National Laboratory, International Meshing Roundtable, Thistle Marble Arch, London, United Kingdom, 2014.
[59] E. Sifakis, K. Der, R. Fedkiw, Arbitrary cutting of deformable tetrahedralized objects, in: Proceedings of SIGGRAPH 2007, 2007, pp. 73–80.
[60] J.-H. Song, P. Areias, T. Belytschko, A method for dynamic crack and shear band propagation with phantom nodes, Int. J. Numer. Methods Eng. 67 (2006) 868–893.
[61] N. Sukumar, Voronoi cell finite difference method for the diffusion operator on arbitrary unstructured grids, Int. J. Numer. Methods Eng. 57 (2003) 1–34.
[62] F. van der Bos, V. Gravemeier, Numerical simulation of premixed combustion using an enriched finite element method, J. Comput. Phys. 228 (2009) 3605–3624.
[63] R. Vanselow, Relations between FEM and FVM applied to the Poisson equation, Computing 57 (1996) 93–104.
[64] A. Wiegmann, K. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (2000) 827–862.
[65] S. Yu, Y. Zhou, G. Wei, Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces, J. Comput. Phys. 224 (2) (2007) 729–756.
[66] S. Zhao, G. Wei, High-order {FDTD} methods via derivative matching for Maxwell's equations with material interfaces, J. Comput. Phys. 200 (1) (2004) 60–103.