A comparative study of penalization and phase field methods for the solution of the diffusion equation in complex geometries

Gerardo Tauriello, Petros Koumoutsakos *

Professorship of Computational Science, ETH Zürich, Clausiusstrasse 33, CH-8092 Zürich, Switzerland

A R T I C L E   I N F O

Article history:
Received 11 July 2014
Received in revised form 19 October 2014
Accepted 25 November 2014
Available online 10 December 2014

Keywords:
Fictitious domain methods
Phase field methods
Volume penalization
Level set methods
Parabolic problems

A B S T R A C T

We present a comparative study of penalization and phase field methods for the solution of the diffusion equation in complex geometries embedded using simple Cartesian meshes. The two methods have been widely employed to solve partial differential equations in complex and moving geometries for applications ranging from solid and fluid mechanics to biology and geophysics. Their popularity is largely due to their discretization on Cartesian meshes thus avoiding the need to create body-fitted grids. At the same time, there are questions regarding their accuracy and it appears that the use of each one is confined by disciplinary boundaries. Here, we compare penalization and phase field methods to handle problems with Neumann and Robin boundary conditions. We discuss extensions for Dirichlet boundary conditions and in turn compare with methods that have been explicitly designed to handle Dirichlet boundary conditions. The accuracy of all methods is analyzed using one and two dimensional benchmark problems such as the flow induced by an oscillating wall and by a cylinder performing rotary oscillations. This comparative study provides information to decide which methods to consider for a given application and their incorporation in broader computational frameworks. We demonstrate that phase field methods are more accurate than penalization methods on problems with Neumann boundary conditions and we present an error analysis explaining this result.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

The solution of partial differential equations (PDE) in complex and moving geometries is at the core of numerous application domains. The discretization of the governing equations can be broadly distinguished in Eulerian and Lagrangian approaches. We believe that a key distinction between these two approaches is the way that boundary conditions are enforced. For example, in solid mechanics, we compute the numerical solution of the equations of motion within a given physical domain while boundary conditions (BC) are defined either on the displacement of the interface or on the traction applied to it. Lagrangian or Arbitrary Lagrangian Eulerian methods track physical interfaces explicitly to simplify the handling of BC. While doing so, Lagrangian methods assume a fixed connectivity within the physical domain. This can lead to numerical errors in problems with extreme loadings and large deformations [1,2]. Eulerian methods, on the other hand, solve the equations on a fixed grid while the physical domain is deforming. Since such methods do not assume a fixed...
connectivity, they can readily handle highly deforming solids. Furthermore, Eulerian methods with deforming and moving obstacles are commonly employed in fluid mechanics and they have been used for challenging simulations of single and multiple swimmers [3,4] and mixing devices [5]. Such methods can readily be coupled with Eulerian solid mechanics methods to enable the study of flow–structure interaction problems [6,7]. Similar problems with large deformations occur in numerical simulations of biological growth processes. There, one commonly requires the solution of PDEs to compute mechanical properties and the diffusion of chemical species within growing and deforming objects [8–11].

PDEs can efficiently be solved on regularly shaped domains by employing simple Cartesian grids. These grids facilitate the development of the numerical methods and their implementation on High Performance Computing architectures [12,13]. PDE solvers for problems with complex geometries usually employ carefully generated body-fitted meshes. The equations can then be discretized with a finite element method (FEM) [14] or a finite difference method suited for unstructured meshes [15]. Those methods are highly accurate but have two major drawbacks: First, the resulting computations are more irregular and expensive than for a regular mesh. Second, the mesh generation can be computationally expensive and this cost may become prohibitive for applications with moving and deforming domains even when Arbitrary Lagrangian Eulerian methods [16,17] are used.

Numerous works have addressed the solution of PDEs within complex geometries without a body-fitted mesh. We distinguish grid based and particle methods and present a non-exhaustive list of related works. For grid based methods, the list starts with the extensions of solutions of Poisson’s equation from rectangular to irregular regions [18]. Hunt introduced a two dimensional (2D) finite difference method to approximate Robin BC without a body-fitted mesh by defining special discretizations of the spatial derivatives close to the boundary [19]. The classic immersed boundary method embeds the irregular boundaries in regular Cartesian meshes and adds a forcing term to the PDE close to the boundary to impose the BC [20,21]. The immersed interface method, proposed as a more accurate alternative to the immersed boundary method, translates the forcing term into jump conditions for the solution variable of the PDE [22]. Numerically, this again introduces special discretizations of the spatial derivatives close to the boundary which can be defined with finite difference, finite volume or finite element methods [23,24].

Level set methods were introduced as a powerful technique to accurately capture smooth interfaces and their deformations [25–28]. Based on the implicit level set representation of the interface one can define methods to extrapolate values from one side of the interface to the other one [29]. This can be used to solve PDEs within an irregular domain by embedding it into a regular domain and using “ghost” grid points near boundaries [30]. Such extrapolation methods can be computationally expensive as they require the solution of several hyperbolic PDEs at each time step to impose the BC. Alternatively, one can use the level set function to accurately identify the location of the interface and define special discretizations of the spatial derivatives close to the boundary for Dirichlet BC [31].

Phase field methods are an alternative to level set methods to implicitly define interfaces and have been used in various fields such as crystal growth, multi-component fluid flows and material sciences [32–34]. Commonly, the phase field is used to distinguish between phases in a material. Kockelkoren et al. introduced a variant of the phase field method to solve the diffusion equation within an irregular domain embedded in a larger regular domain [35]. Levine and Rappel later extended this method to impose Robin BC by adding a forcing term to the PDE [36]. This method was recently coupled with a remeshed particle level set method to solve reaction–diffusion systems on the surface and the interior of deforming geometries [11]. The “diffuse domain” approach extends the phase field method to prescribe Dirichlet, Neumann and Robin BC with a variety of forcing terms [37].

Penalization methods follow a similar approach by embedding the irregular domain into a larger regular domain called “fictitious domain” and adding a forcing term to the PDE. In contrast to immersed boundary methods, this forcing term may affect all of the “external” part of the larger domain. This is for instance used to impose Dirichlet BC for the Navier–Stokes equations by using Brinkman penalization [38–40]. Ramiere et al. proposed a “spread interface” penalization method to impose Dirichlet, Neumann and Robin BC for the solution of elliptic problems [41]. Kadoch et al. introduced a volume penalization method to impose homogeneous Neumann BC for scalar advection–diffusion with moving obstacles as in a chemical mixer [5]. Theoretical error estimates for fictitious domain and penalization methods were computed for solutions to Poisson’s equation with flux BC [42].

Particle methods [43–45] have been proposed for the solution of PDEs such as diffusion equations, starting with the method of Particle Strength Exchange [46]. The incorporation of BC in such methods remains a topic of active research [11,47–49] with significant success in the formalism of the Reproducing Kernel Particle methods [50]. In recent years, it has become evident that particle methods simulating mechanical systems with large deformations need to be coupled with remeshing procedures [51,52] that employ a Cartesian grid. Hence, the enforcement of BC in remeshed particle methods can be mapped back to the problem of enforcing them on a Cartesian grid.

In this work, we consider the diffusion equation as the model PDE and use it to compare solution methods for irregular domains that employ Cartesian grids for low computational cost [5,31,35,36,39,41]. To the best of our knowledge, those methods were never carefully compared on identical time dependent test problems and we believe that this comparison can guide decisions on which methods to consider for a given problem.

We exclude immersed boundary and immersed interface methods from our analysis as they cannot readily be used to solve the diffusion equation within a given domain. They have been designed to solve flow problems with an interface within the flow as opposed to solving a general PDE within a domain. We furthermore exclude methods which require ghost values near boundaries as they require computationally expensive extrapolations to compute those ghost values. The
remaining methods can roughly be classified in two classes: methods which extend the PDE with forcing terms and methods which define special discretizations of the spatial derivatives close to the boundary. Within the former class, we consider phase field \cite{35,36} (see Section 3.2) and penalization \cite{5,41} (see Section 3.3) methods which can impose Neumann and Robin BC. We prefer methods which can impose Neumann and Robin BC as those methods might be generalized to any flux BC. Those methods can also be extended to impose Dirichlet BC. We also consider Brinkman \cite{39} and spread interface \cite{41} penalization methods which were specifically designed for Dirichlet BC (see Section 3.4). Within the class of methods with special discretizations of the spatial derivatives, we consider a level set method for Dirichlet BC \cite{31} (see Section 3.6) and an extension of the method for Neumann BC in one dimension (1D) (see Section 3.7). We note that those methods have two disadvantages: the special discretizations of the spatial derivatives introduces branching operations in the implementation and often requires access to additional neighboring grid points. Given an arbitrarily shaped domain, it is not always the case that a certain number of neighboring grid points lie within the domain. The methods thus fail depending on how the domain boundary cuts through the mesh. Branching operations, on the other hand, limit the efficiency of the implementation on modern computer architectures, which rely on vectorization to achieve their best performance.

This article is structured as follows: In Section 2 we introduce the diffusion equation and its BC which we use as a benchmark problem for our comparative study. Section 3 describes the methods considered and the numerical discretization of the diffusion equation. In Section 4 we show results for 1D and 2D benchmark problems as well as for the flow induced by an oscillating wall and by an oscillating spinning cylinder. In Section 5 we propose a theoretical error analysis for the phase field method and extend the method for moving interfaces, generic fluxes and reaction terms. Section 6 presents a similar analysis for the penalization methods. Section 7 summarizes the results and proposes future extensions of the method.

2. Governing equations

We consider the diffusion equation within a closed domain \( \Omega \) with a smooth boundary \( \Gamma = \partial \Omega \) (Fig. 1):
\[
\frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u) \quad \text{in } \Omega,
\]
\[
(D \nabla u) \cdot n = B_R - \alpha_R u \quad \text{on } \Gamma = \partial \Omega,
\]
where \( u \) is a diffusing scalar, \( D \) is the diffusion constant, \( n \) is the outward pointing surface normal on \( \Gamma \) and \( B_R \) and \( \alpha_R \) define the Robin boundary condition (BC). The values of \( B_R \) may vary in space while the values of \( \alpha_R \) are constant. The Neumann BC can be defined by setting \( \alpha_R = 0 \).

We also consider Dirichlet BC with a known value \( B_D \) which may vary in space:
\[
\frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u) \quad \text{in } \Omega,
\]
\[
u = B_D \quad \text{on } \Gamma.
\]
We can consider this as a limiting case of Eq. (1) for \( \alpha_R \to \infty \) and \( B_R = \alpha_R B_D \).

3. Methods

We wish to solve Eq. (1) or Eq. (2) in complex, time dependent domains \( \Omega \) (see Section 5.1). Our goal is to use a Cartesian grid to discretize the computational domain. For that purpose, we embed the domain \( \Omega \) in a larger domain \( \tilde{\Omega} \) (Fig. 1) where we define the scalar \( \tilde{u} \) which approximates the exact \( u \) in \( \Omega \). We then solve the following equation that provides a unifying formulation for the phase field and penalization techniques as described below:
\[
\frac{\partial \tilde{u}}{\partial t} = \tilde{\psi} \nabla \cdot (\tilde{D} \nabla \tilde{u}) + f - b \tilde{u} \quad \text{in } \tilde{\Omega},
\]
periodic BC for \( \tilde{u} \) 
\[
\text{on } \partial \tilde{\Omega}.
\]

The parameters \( \tilde{\psi}, \tilde{D}, f \) and \( b \) depend on the specific method as explained in the next subsections. \( f - b \tilde{u} \) represents a forcing term that is used to impose the BC in penalization and phase field methods. The \( b \tilde{u} \) term is only required to impose Robin and Dirichlet BC. The phase field method additionally introduces the factor \( \tilde{\psi} \) for the diffusion term (otherwise \( \tilde{\psi} = 1 \)). Dirichlet BC may also be imposed by setting \( \tilde{\psi} = 1 \) and \( b = f = 0 \) and introducing a special discretization for the diffusion term close to the boundary \( \Gamma \). In 1D one can also easily follow the same approach for Neumann and Robin BC.
3.1. Level set representation of the boundary

The boundary \( \Gamma \) is embedded in \( \Omega \) and is implicitly represented as the zero isocontour of a level set function [25]:

\[
\Gamma = \{ x \mid \phi(x) = 0 \}.
\] (4)

We initialize \( \phi \) as a signed distance function such that \( |\nabla \phi| = 1 \) and we choose to have \( \phi < 0 \) in \( \Omega \) and \( \phi > 0 \) on the outside. The outward pointing surface normal is computed as \( \mathbf{n} = \nabla \phi / |\nabla \phi| \).

3.2. Phase field method for Neumann and Robin boundary conditions

The phase field method [35,36] can be used to solve the diffusion equation with moving and static boundaries. The phase field \( \psi \) (denoted as \( \phi \) in the referred articles) is a mollified Heaviside function which is \( \approx 1 \) inside \( \Omega \) and \( \approx 0 \) otherwise. We define it as a function of the level set function \( \phi \) (Eq. (4)), with a width \( w (\psi_w(\phi) = \psi(\phi/w)) \) and we assume antisymmetry \( \psi(x) = 1 - \psi(-x) \). Here, we choose the function

\[
\psi(x) = \begin{cases} 
1, & x < -1, \\
\frac{1}{2}(1 - x - \sin(\pi x)/\pi), & -1 \leq x \leq 1, \\
0, & x > 1,
\end{cases}
\] (5)

which has the properties \( \psi(1) = 0 \), \( \psi(-1) = 1 \) and \( \psi'(1) = \psi'(1) = 0 \).

We can then approximate a solution to Eq. (1) by solving

\[
\frac{\partial \tilde{u}}{\partial t} = \nabla \cdot (\psi_w D \nabla \tilde{u}) + (B_R - \alpha_R \tilde{u}) \frac{(\psi'_w)^2}{K} \quad \text{in } \tilde{\Omega},
\]

periodic BC for \( \tilde{u} \) on \( \partial \tilde{\Omega} \).

where \( K = f(\tilde{\Omega}) (\psi'_w)^2 / L \) is a normalization factor and \( L \) is the length of the interface \( \Gamma \) (\( L \) must be replaced with the area \( A \) of the interface in three dimensional (3D) setups). In the present work, we use the value \( K = 3/(4w) \) which is accurate if the maximal curvature is \( \ll 1/w \). We can rewrite Eq. (6) as Eq. (3) with

\[
\tilde{\psi} = \frac{1}{\psi_w}, \quad \tilde{D} = \psi_w D, \quad f = B_R \frac{(\psi'_w)^2}{K \psi_w}, \quad b = \alpha_R \frac{(\psi'_w)^2}{K \psi_w} \quad \text{in } \tilde{\Omega}.
\] (7)

Following [35], we set \( \partial \tilde{u}/\partial t = 0 \) for \( \psi_w < 10^{-8} \) in Eq. (3) which essentially ignores all values of \( \tilde{u} \) far outside of \( \Omega \) (\( \phi \geq w \)). We note that the BC on \( \Gamma \) are now enforced using the \( (\psi'_w)^2/(K \psi_w) \) term which is only non-zero close to \( \Gamma \) (\( |\phi| < w \)). We furthermore note that by setting \( B_R = \alpha_R B_D \) with \( \alpha_R \gg 1 \) we can also handle Dirichlet BC as in Eq. (2).

3.3. Penalization for Neumann and Robin boundary conditions

The “spread interface” penalization method [41] distinguishes three kinds of cells for the Cartesian grid used for the discretization of the domain \( \tilde{\Omega} \). We denote the grid cells fully inside \( \Omega \) as \( \Omega_h \), the ones intersecting \( \Gamma \) as \( \Omega_{h,R} \) and the ones fully outside of \( \Omega \) as \( \Omega_{h,e} \). In summary, \( \Omega_h \cup \Omega_{h,R} \cup \Omega_{h,e} \) denote all our grid cells.

To approximate Eq. (1) we can then solve Eq. (3) with

\[
\tilde{\psi} = 1, \quad \tilde{D} = D, \quad f = 0, \quad b = 0 \quad \text{in } \Omega_h,
\]

\[
\tilde{\psi} = 1, \quad \tilde{D} = D, \quad f = \frac{B_R}{\epsilon}, \quad b = \frac{\alpha_R}{\epsilon} \quad \text{in } \Omega_{h,R},
\]

\[
\tilde{\psi} = 1, \quad \tilde{D} = \eta, \quad f = 0, \quad b = 0 \quad \text{in } \Omega_{h,e},
\] (8)

where \( 0 \leq \eta \ll 1 \) is the penalization parameter. In contrast to similar methods for Dirichlet BC, we can even set \( \eta = 0 \) if our numerical method allows this as is the case in the present work. The parameter \( \epsilon \) corrects for approximating \( \Gamma \) with \( \Omega_{h,R} \). \( \epsilon \) is of the order of grid spacings and should enforce

\[
\int_{\Gamma} (\alpha_R u - B_R) ds = \int_{\Omega_{h,R}} \frac{\alpha_R u - B_R}{\epsilon} ds.
\] (9)

We choose a simple approximation to Eq. (9) by computing \( \epsilon = \int_{\Omega_{h,R}} dx / f_R ds = \int_{\Omega_{h,R}} dx / L \), where \( L \) is the length of the interface \( \Gamma \). In 2D, this can lead to resolution dependent errors introduced by \( \omega_{h,R} \) which may have varying widths ranging from \( h \) to \( 2h \). For 1D test cases this is not an issue. Better approximations to Eq. (9) exist for 2D problems [41] and we refer the reader to the cited article for a more detailed discussion. In this work, we show that even in 1D test cases, this penalization method has resolution dependent errors which can be removed with other penalization methods or the phase field method. We therefore neglect better approximations to Eq. (9).
Table 1
Summary of parameters for Eq. (3) considered in this study.

<table>
<thead>
<tr>
<th>Phase field (RBC, Eq. (7))</th>
<th>Penalty Spread (RBC, Eq. (8))</th>
<th>Penalty (DBC, Eq. (10))</th>
<th>Penalty Spread (DBC, Eq. (12))</th>
<th>Penalty Exterior (DBC, Eq. (13))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \psi )</td>
<td>( \psi_w )</td>
<td>( \tilde{D} )</td>
<td>( f )</td>
<td>( b )</td>
</tr>
<tr>
<td>( \frac{1}{\sqrt{\omega}} )</td>
<td>( \frac{1}{\sqrt{\omega}} )</td>
<td>( D ) in ( \Omega_b \cup \omega_h, r )</td>
<td>( B_R/e ) in ( \omega_h, r )</td>
<td>( \alpha_R ) in ( \omega_h, r )</td>
</tr>
<tr>
<td>( \tilde{D} )</td>
<td>( \psi_w D )</td>
<td>( \delta_w B_R )</td>
<td>( \delta_w \alpha_R )</td>
<td>( \delta_w \alpha_R )</td>
</tr>
</tbody>
</table>

The discrete distinction of grid cells can be replaced with a Heaviside function to distinguish the region inside of \( \Omega \) and the one outside of it [5]. Similarly, we use the phase field \( \psi_w \) defined in Eq. (5) as a mollified Heaviside function. To enforce the BC, we introduce the mollified Dirac delta function \( \delta_w(\phi) = -\psi_w(\phi) = -\psi'(\phi)/\psi \). We then approximate a solution to Eq. (1) by solving Eq. (3) with

\[
\tilde{\psi} = 1, \quad \tilde{D} = \psi_w D + (1 - \psi_w)\eta, \quad f = \delta_w B_R, \quad b = \delta_w \alpha_R \quad \text{in } \tilde{\Omega},
\]

where \( 0 \leq \eta \ll 1 \) is the penalization parameter defined above, which we can again set to zero if our numerical method allows this. This method is very similar to the phase field method in Eq. (7). The main difference is the \( \psi_w \) term in the left-hand side of Eq. (6). We note that in the referred article [5], only zero Neumann BC \( B_R = \alpha_R = 0 \) were considered and a sharp interface \( \psi = 0 \) was used. Nonetheless, we show that the version proposed here has a similar accuracy in space (less than first order as shown for instance in Fig. 6 of [5]). We again note that by setting \( B_R = \alpha_R B_D \) with \( \alpha_R \gg 1 \) we can also handle Dirichlet BC as in Eq. (2).

3.4. Penalization for enforcing the Dirichlet boundary conditions

The Brinkman penalization method [39] can readily be expressed in terms of the phase field \( \psi_w \) defined in Eq. (5). We can then approximate a solution to Eq. (2) by solving Eq. (3) with

\[
\tilde{\psi} = 1, \quad \tilde{D} = D, \quad f = \frac{B_D}{\eta} (1 - \psi_w), \quad b = \frac{1}{\eta} (1 - \psi_w) \quad \text{in } \tilde{\Omega},
\]

where \( 0 < \eta \ll 1 \) is the penalization parameter.

Two variants of the “spread interface” penalization method [41] were also proposed to enforce Dirichlet BC using the distinction of grid cells into \( \Omega_b, \omega_h, r \text{ and } \omega_h, e \) as described in Section 3.3. We include the methods which they denoted as “spread interface penalization” and “exterior penalization” (variants with “L2 penalty” in [41]) in our study. For the spread interface penalization we approximate Eq. (2) by solving Eq. (3) with

\[
\tilde{\psi} = 1, \quad \tilde{D} = D, \quad f = 0, \quad b = 0 \quad \text{in } \omega_b, \quad \tilde{\psi} = 1, \quad \tilde{D} = D, \quad f = \frac{B_D}{\eta}, \quad b = \frac{1}{\eta} \quad \text{in } \omega_h, r, \quad \tilde{\psi} = 1, \quad \tilde{D} = D, \quad f = 0, \quad b = 0 \quad \text{in } \omega_h, e.
\]

For the exterior penalization on the other hand we use the following parameters in Eq. (3):

\[
\tilde{\psi} = 1, \quad \tilde{D} = D, \quad f = 0, \quad b = 0 \quad \text{in } \Omega_b, \quad \tilde{\psi} = 1, \quad \tilde{D} = D, \quad f = 0, \quad b = 0 \quad \text{in } \omega_h, r, \quad \tilde{\psi} = 1, \quad \tilde{D} = D, \quad f = \frac{B_D}{\eta}, \quad b = \frac{1}{\eta} \quad \text{in } \omega_h, e.
\]

3.5. Numerical discretization

To sum up, we solve Eq. (3) with parameters \( \tilde{\psi}, \tilde{D}, f \) and \( b \) as defined in Table 1. We note that we set \( \eta = 0 \) in Eq. (8) and Eq. (10) since our numerical method allows this. The computational domain \( \Omega \) is discretized on a regular Cartesian mesh with grid spacing \( h \). The inhomogeneous diffusion \( \nabla \cdot (D \nabla u) \) at grid point \( i \) is discretized by the following second order accurate stencil:

\[
\left[ \nabla \cdot (\tilde{D} \nabla u) \right]_i \approx \frac{1}{h^2} \sum_{j \in N(i)} \tilde{D}_{i,j} (u_j - u_i),
\]
where \( N(i) \) is the set of nearest neighbors of \( i \). \( \hat{D}_{i,j} \) is computed using a harmonic mean [53]:

\[
\hat{D}_{i,j} = 2 \left( \frac{1}{D_i} + \frac{1}{D_j} \right)^{-1} = \frac{2\hat{D}_i\hat{D}_j}{\hat{D}_i + \hat{D}_j}.
\]  

(15)

We note that for constant values of \( \hat{D}_i = D \) this reduces to the well known central difference stencil for the Laplacian \( (\hat{D}_{i,j} = D) \). We would also like to point out that in all the considered methods, the domain \( \Omega \) could be discretized in any preferred way, since the boundary \( \Gamma \) is not required to be aligned with the mesh.

We discretize Eq. (3) in time using a forward Euler scheme with a constant time step \( \delta t \). The explicit discretization in time limits the time step. For the diffusion term, the time step is limited by \( \delta t < h^2 / (2dD) \), where \( d \) is the dimensionality of the problem. The BC term, on the other hand, limits the time step as \( \delta t < 1 / \max(b) \). The value of \( \max(b) \) depends on the method used and is \( \max(b) \approx 3.9\alpha \rho / w \) for the phase field method using Eq. (7), \( \max(b) = \alpha \rho / h \) for the spread interface penalization method using Eq. (8), \( \max(b) = \alpha \rho / w \) for the penalization method using Eq. (10) and \( \max(b) = 1 / \eta \) for any of the penalization methods for DBC described in Section 3.4.

The limitation of \( \delta t \) due to the BC term can easily become prohibitive when trying to solve problems with Dirichlet BC (due to the large \( \alpha \rho \) or \( 1 / \eta \) respectively). In these cases we use a time splitting scheme where we first consider the \( \delta u \) and \( f \) terms in Eq. (3) using backward Euler and then perform the diffusion step with forward Euler. Given the solution \( u^n \) at time \( t^n \) we first compute

\[
u^* = u^n + \delta t f^n \frac{1}{1 + \delta t b}.
\]  

(16)

and then do a forward Euler step for the diffusion part using \( u^* \) to compute \( u^{n+1} \) at time \( t^{n+1} = t^n + \delta t \). We note that we could also use more stable and more accurate time integration methods such as alternating direction implicit (ADI) but this was not considered for the present work.

The expected accuracy of the discretization in time and in space is of \( O(h^2) \). Our spatial discretization of the inhomogeneous diffusion in Eq. (14) has an error of \( O(h^2) \) and we use first order time integration schemes. Due to the stability constraints of forward Euler, we choose \( \delta t = O(h^2) \) which turns the time integration error \( O(\delta t) \) into \( O(h^2) \).

The initial condition for \( \tilde{u} \) must also be set for the values in \( \Omega \) which are outside of the domain \( \Omega \). Fig. 2 shows the three considered options visualized for a 1D function \( u(y) \) defined for \( y \geq 0 \) (this \( u(y) \) will be used in Section 4.4):

1. ICE0: Using the exact \( \tilde{u}(y) \) as if the function would be defined outside of \( \Omega \) too.
2. ICE1: Mirroring the values from within \( \Omega \) (i.e. locally symmetric values of \( u \) around \( \Gamma \)).
3. ICE2: Setting \( \tilde{u}(y) = 0 \) outside of \( \Omega \).

Unless otherwise noted, we use the mirroring method (ICE1) for all our tests.

3.6. Level set method for Dirichlet boundary conditions

We compare the penalization and phase field methods described above with a level set method which was originally proposed to solve Poisson’s equation on irregular domains [31]. We implement the method by solving Eq. (2) for all grid points within \( \Omega \) \( (\phi < 0) \) using the same Cartesian mesh as we used for the penalization and phase field methods. The grid points outside of \( \Omega \) are not used. We use Eq. (14) to discretize the diffusion term \( \nabla \cdot (D \nabla u) \) at grid point \( i \), where we must define ghost values if the neighboring value \( u_j \) is outside of \( \Omega \). We use the discretized value \( \phi_i \) of the level set function at grid point \( i \) and the value \( \phi_j \) at its neighboring point \( j \) to determine if we need ghost values. We then define the distance to the boundary \( \Gamma \) as \( \theta \) with

\[
\theta = \frac{\phi_i}{\phi_i - \phi_j}.
\]  

(17)
If we draw a line from point \( i \) to point \( j \), \( \theta h \) is the distance from point \( i \) to the boundary (see Fig. 3). In 1D this simplifies as we have \( \phi_j = \phi_i + h \) and therefore \( \theta = -\phi_i / h = |\phi_i| / h \). We can define the ghost value for \( u_j \) using \( \theta \) as

\[
    u_j = \frac{B_D + (\theta - 1)u_i}{\theta}.
\]

(18)

This discretization can lead to numerical issues if \( \theta \) is close to zero. We do not use Eq. (18) for \( \theta \approx h \) and instead set \( u_i = B_D \) [31]. This only introduces an \( O(h^2) \) error \( (B_D = u_i + O(\theta h) = u_i + O(h^2)) \) and is thus in line with the desired accuracy of the method. Nonetheless, the discretization leads to numerical instabilities for our forward Euler time integration scheme unless we choose \( \delta t = O(h^3) \). This can be fixed with implicit time integration schemes but for the purpose of the study in this work we have just chosen a very small \( \delta t \) when using this stencil.

We can define an alternative way to compute the ghost value \( u_j \) which doesn’t suffer from the stability constraints explained above. For this we consider the next grid point \( u_{i+1} \) away from the interface \( \Gamma \) inside of \( \Omega \) (see Fig. 3). We then compute the ghost value for \( u_j \) as

\[
    u_j = \frac{2B_D + (\theta - 1)u_{i+1}}{\theta + 1}.
\]

(19)

This discretization removes the numerical issues if \( \theta \) is close to zero but it can fail if \( u_{i+1} \) is not inside \( \Omega \).

### 3.7. Finite difference method for Neumann boundary condition in 1D

To the best of our knowledge, there is no work which extends the method outlined in Section 3.6 or immersed interface methods to Neumann and Robin BC. The 2D finite difference stencils proposed by Hunt [19] for instance can easily fail for general interfaces \( \Gamma \) as it requires up to three grid points in each direction inside of \( \Omega \) when discretizing the Robin BC. In Fig. 3 this implies that we would need \( u_i, u_{i+1}, u_{i+2} \) to be within \( \Omega \). For general interfaces \( \Gamma \) this is a stringent assumption. For the purpose of comparison, we can nonetheless consider a simple 1D case of Eq. (1) with \( \alpha_B = 0 \) (Neumann BC). If we consider a boundary as depicted in Fig. 3, we can define \( u' \) on \( \Gamma \) as \( u'_\Gamma = -B_R / D \) and approximate the second derivative \( u''_\Gamma \) at grid point \( i \) to first order as

\[
    u''_\Gamma = \frac{2(u_{i+1} - u_i - h u'_\Gamma)}{h^2 (1 + 2\theta)}.
\]

(20)

A second order accurate version of the above is given by

\[
    u''_\Gamma = \frac{(3\theta^2 - 1)u_{i+2} + (8 - 6\theta^2)u_{i+1} + (3\theta^2 - 7)u_i - 6hu'_\Gamma}{h^2 (2 + 6\theta + 3\theta^2)}.
\]

(21)

### 4. Results

The methods outlined in the previous section were implemented using the CUBISM framework [12]. The diffusion equation is solved with Neumann (NBC), Robin (RBC) and Dirichlet (DBC) boundary conditions discretized on the original and the extended domain (through penalization and phase field methods as summarized in Table 1). We distinguish the solution of the following equations for the NBC and RBC cases:

1. Phase field (RBC): Eq. (3) with parameters as in Eq. (7) with \( w = 2h \).
2. Penalty Spread (RBC): Eq. (3) with parameters as in Eq. (8).
3. Penalty (RBC): Eq. (3) with parameters as in Eq. (10) with \( w = 2h \).
4. FDM 1 (NBC): Eq. (1) with the first order stencil defined in Eq. (20) (1D NBC only).
5. FDM 2 (NBC): Eq. (1) with the second order stencil defined in Eq. (21) (1D NBC only).

In turn, we solve for the diffusion equation (2) with DBC by using:

1. Phase field (DBC limit): Eq. (3) with parameters as in Eq. (7) with \( w = h, B_R = \alpha_R B_D \) and \( \alpha_R = 10^6 \).
2. Penalty Spread (DBC limit): Eq. (3) with parameters as in Eq. (8) with $B_R = \alpha_R B_D$ and $\alpha_R = 10^6$.
3. Penalty (DBC): Eq. (3) with parameters as in Eq. (11) with $w = h$ and $\eta = 10^{-6}$.
4. Penalty Spread (DBC): Eq. (3) with parameters as in Eq. (12) with $\eta = 10^{-6}$.
5. Penalty Exterior (DBC): Eq. (3) with parameters as in Eq. (13) with $\eta = 10^{-6}$.
6. Level set (DBC): Eq. (2) with ghost values as defined in Eq. (18).
7. Level set shifted (DBC): Eq. (2) with ghost values as defined in Eq. (19).

We focus on phase field and penalization methods for NBC and RBC for most of this section as those methods might be generalized to any flux BC. We provide a comparison of the methods for DBC in Section 4.3 and a comparison with the “FDM” methods for a 1D problem in Section 4.4. We again point out that methods with special discretizations of the spatial derivatives close to the interface (“Level set” for DBC and “FDM” for NBC) have disadvantages in implementation (branching operations needed) and they may fail depending on how the domain boundary cuts through the mesh.

In the context of the restrictions detailed in Section 3.5, we choose the time step $\delta t$ for the time integration scheme according to:

$$\delta t = \min\left(\frac{h^2}{8D}, \frac{1}{l_b}\right), \quad l_b = \begin{cases} \frac{1.5\alpha_R/w}{\omega}, & \text{for Phase field (RBC),} \\ \frac{0.5\alpha_R/h}{\omega}, & \text{for Penalty Spread (RBC),} \\ \frac{0.5\alpha_R/w}{\omega}, & \text{for Penalty (RBC),} \\ 0, & \text{else.} \end{cases}$$

(22)

For all convergence results, we report the convergence in space and time where we vary both the grid spacing $h$ and the time step $\delta t$ according to the equation above. The time integration error is then in $O(h^2)$ and will not hide the spatial convergence which will at most be in $O(h^2)$ given the used discretization of the diffusion operator.

Given the computed discrete solutions $u_i$ in $\Omega_h$ (grid cells fully enclosed by $\Omega$) and the exact analytical result $u_e$ at $t = 10$, we compute the $L_\infty = \max_i(|u_i - u_e|)$ as well as the $L_1$ and $L_2$ norms defined as

$$L_p = \left(\sum_i |u_i - u_e|^p vol_i\right)^{1/p},$$

(23)

where $vol_i$ is the volume of the $i$-th grid point.

4.1. Benchmark problems in one and two dimensions

We define a 1D test problem with constant $D$ and $B_R = 0$. We manufacture an exact solution of Eq. (1) by using a 1D sine function in a domain $\Omega$ of width 2R centered around $x = x_c$ (see two left plots of Fig. 4):

$$u(x, t) = \sin(\omega x + \varphi) \exp(-\omega^2 D t),$$

$$0 = \omega R + \arctan(D\omega/\alpha_R) - \pi,$$

$$\varphi = \arctan(D\omega/\alpha_R) - \pi - \omega(x_c - R).$$

(24) (25) (26)

We solve this in $\Omega = [0, 1]$ with $x_c = 0.5$. For $\alpha_R = 0$, we choose $D = 0.005$ while for $\alpha_R = 0.02$, we set $D = 0.002$. We use the Newton–Raphson method to numerically obtain $\omega$ from Eq. (25). In the limit of $\alpha_R \rightarrow 0$ (i.e. a no-flux BC), we obtain $\omega = \pi / (2R)$.

For this test case, our regular lattice has points at $x_k = ih + h/2$. We consider grid spacings $h = 2^k$ with $k \in [-9, -5]$ which implies that for $R = 0.25$ we have a body fitted mesh. This is a considerable advantage for the spread interface penalization technique. Fig. 5 shows the results of the convergence study for the 1D sine test case with a homogeneous NBC ($\alpha_R = 0$). We observe that the spread interface penalization performs very well for $R = 0.25$ but performs considerably worse at $R = 0.24$. We note that for the 1D test problems our simplistic choice of $\epsilon$ in Eq. (8) has no influence. In Fig. 5, we also observe that the penalization technique with parameters as in Eq. (10) perform poorly compared to the other variants even though it was originally meant to handle no-flux NBC as in this benchmark problem. The phase field variant on the other hand shows a convergence close to second order independently of the choice of $R$.  

![Fig. 4. Analytical solutions to benchmark problems (1D sine and 2D exponential) with $R = 0.25$ shown at $t = 0, 1, \ldots, 10$.](image-url)
Fig. 5. Convergence for 1D sine test case with zero NBC ($B_R = 0, \alpha_R = 0$) for different sizes of $\Omega$ (left: $R = 0.25$, right: $R = 0.24$).

Fig. 6. Convergence for 1D sine test case with zero RBC ($B_R = 0, \alpha_R = 0.02$) for different sizes of $\Omega$ (left: $R = 0.25$, right: $R = 0.24$).

In Fig. 6 we report the errors for the 1D sine test case with a homogeneous RBC using $\alpha_R = 0.02$. Comparing with Fig. 5, we observe that the convergence of the phase field method drops to first order. Our error analysis in Section 5 explains the $O(h^2)$ error for $\alpha_R = 0$ and gives a possible source of the $O(h)$ error for $\alpha_R \neq 0$. The convergence for the spread interface penalization method also drops to first order for $R = 0.25$ and shows similar results for $R = 0.24$ as for $\alpha_R = 0$. The other penalization method performs as in the $\alpha_R = 0$ case which shows that our extension of the method to RBC does not introduce additional errors.

As a second benchmark problem, we consider a radially symmetric problem in 2D with a circular domain $\Omega$ with radius $R$ centered at [0.5, 0.5]. This problem will be solved again on a Cartesian grid. An exact solution of the governing equation (1) can be obtained as

\[
\begin{align*}
\frac{1}{\pi} \left( \frac{4aDt}{4aDt + 1} \right)^{\frac{1}{2}} \exp \left( -\frac{aR^2}{4aDt + 1} \right), \\
B_R(t) = \frac{\alpha_R(4aDt + 1 - 2aDR)}{\pi(4aDt + 1)^2} \exp \left( -\frac{aR^2}{4aDt + 1} \right). 
\end{align*}
\]

We solve for this problem in $\hat{\Omega} = [0, 1]^2$ with a diffusion constant $D = 0.005$, a radius of $R = 0.25$ and $a = 10$ (see Fig. 4).

In Fig. 7 we show the $L_2$ errors with NBC ($\alpha_R = 0$) and RBC ($\alpha_R = 0.02$). We chose $\alpha_R = 0.02$ to have comparable magnitudes for the $(D \nabla u)$ and $\alpha_R \nabla u$ terms which gives them a comparable influence on the solution. Similarly to the 1D sine test case, we observe a convergence close to second order for the phase field method with $\alpha_R = 0$ which drops to first order with $\alpha_R \neq 0$. Both penalization methods show a convergence close (or slightly below) first order for both NBC and RBC. We also observe that the penalization technique with parameters as in Eq. (10) performs similarly for its intended use (homogeneous NBC as in Fig. 5) as for our extension to inhomogeneous NBC and RBC.

Fig. 8 shows the convergence results for different initializations of $u$ outside of $\Omega$ according to the variants detailed in Section 3.5 and shown in Fig. 2. We observe that extending the exact initial condition (ICE0) and mirroring the values (ICE1) perform very similarly. Setting the values to 0 outside of $\Omega$ (ICE2), on the other hand, introduces an $O(h)$ error for
the phase field method. According to Eq. (42) in our error analysis in Section 5, a continuous extension of \( u \) outside of \( \Omega \) will only introduce an \( O(h^2) \) error and mirroring (ICE1) will minimize this error. For discontinuous extensions as in ICE2 on the other hand, an \( O(h) \) error is added to Eq. (39) and Eq. (42).

4.2. Mass conservation in 2D

To test how well mass is conserved with the different methods, we consider a test case where a given initial condition diffuses with no-flux boundary conditions (\( B_R = \alpha_R = 0 \)). We consider a circular domain \( \Omega \) with radius \( R \) centered at \([0.5, 0.5]\) and define the following initial conditions in polar coordinates:

\[
  u(r, t = 0) = \begin{cases} 
    \cos(\frac{\pi}{2a} r^2), & |r| < a, \\
    0, & |r| \geq a.
  \end{cases}
\]  

(28)

The exact solution at \( t \to \infty \) can be computed due to the conservation of \( \int_{\Omega} u \):

\[
  u(r, t = \infty) = \frac{A}{\pi R^2},
\]

\[
  A = \int_{\Omega} u(t = 0) = \begin{cases} 
    \frac{\pi a^2}{2} - 2 R^2/\pi, & a < R, \\
    \frac{\pi R^2/2 + a^2(\cos(\pi R/a) - 1)/\pi + aR \sin(\pi R/a)}{\pi}, & a \geq R.
  \end{cases}
\]  

(29)

We choose \( D = 0.005 \) and run up to \( t = 20 \) which is long enough to reach the steady-state.

Fig. 9 shows the \( L_\infty \) errors for \( a = 0.25 \) (left, with \( u(R, 0) = 0 \)) and \( a = 0.5 \) (right, with \( u(R, 0) \neq 0 \)). In both cases, the phase field method shows a convergence close to second order. In the \( a = 0.5 \) case, we also observe larger differences in absolute error between the different initializations of \( u \) outside of \( \Omega \) (ICE0 and the default ICE1), but both show similar convergence rates. The penalization methods on the other hand show a convergence close (or slightly below) first order for all cases.
4.3. 2D flow inside a cylinder performing rotary oscillations

We consider the flow inside a rotating cylinder of radius $R$ centered at [0.5, 0.5] with the angular velocity of the cylinder given by $v(R, t) = U(t) = aR \cos(at)$. The exact solution to Eq. (1) with an NBC ($\alpha_R = 0$) for the $y$-component of the velocity $v_y$ (see Fig. 10) is

$$u(r, \theta, t) = v_y(r, \theta, t) = v(r, t) \cos(\theta),$$
$$v(r, t) = R \left( v_C(r, t) \right),$$
$$v_C(r, t) = \frac{aR \exp(iat) I_1(\sqrt{icr})}{I_1(\sqrt{icR})},$$
$$B_R(\theta, t) = v^{(R)}(t) \cos(\theta),$$
$$v^{(R)}(t) = D \frac{\partial v}{\partial r} \bigg|_{r = R} = DR \left( \frac{aR \exp(iat) \sqrt{i}c(I_0(\sqrt{icR}) + I_2(\sqrt{icR}))}{2I_1(\sqrt{icR})} \right).$$

(30)

where $c = \sqrt{a/D}$ and $I_n$ are the modified Bessel functions of first kind. We would like to point out that this test case requires a boundary condition term $B_R$ which depends on space ($\theta$) and time ($t$).

In Eq. (30) we took the real component of the complex solution $v_C$ but one could also take the imaginary component $v_I(r, t) = \Im(v_C(r, t))$ and get $v_I(R, t) = aR \sin(at)$. Also, we chose to look at the $y$-component of the velocity vector in Cartesian coordinates $v = (v_x, v_y)$ with $v_x = -v_y/r = -v \sin(\theta)$ and $v_y = v x/r = v \cos(\theta)$. One could also choose the $x$-component and have $u = v_x$ with a boundary condition given by $B_R = -v^{(R)}(t) \sin(\theta)$ in Eq. (30).

A solution to Eq. (1) can also be obtained using the corresponding vorticity field $\omega = \partial v_y/\partial x - \partial v_x/\partial y$ (see Fig. 10):

$$u(r, \theta, t) = \omega(r, t) = R \left( \omega_C(r, t) \right),$$
$$\omega_C(r, t) = \frac{aR \exp(iat) \sqrt{i}c I_1(\sqrt{icr})}{I_1(\sqrt{icR})},$$
$$B_R(t) = -a^2 R \sin(at).$$

(31)
The computational domain is chosen as \( \hat{\Omega} = [0, 1]^2 \) and we set a constant \( D = 0.001 \) and \( a = 0.628 \). We chose \( \alpha \) such that approximately one period of the oscillatory angular velocity is completed if we run up to \( t = 10 \).

In Fig. 11, we show the errors when solving for the \( y \)-component of the velocity \( v_y \) in Eq. (30) and when solving for the vorticity \( \omega \) in Eq. (31). We consider cylinder radii of \( R = 0.25 \) (top) and \( R = 0.4 \) (bottom) where the latter reduces the derivative of \( v \) and the absolute value of \( \omega \) at \( r = 0 \). In all cases, the phase field method shows a convergence close to second order while the spread interface penalization method does not converge and the other penalization method displays a convergence below first order. In the plot inlets of Fig. 11, we observe that the \( L_\infty \) error converges at first order for the phase field method.

Fig. 12 shows a more detailed view of \( v_y \) computed with the phase field method with a cylinder radius of \( R = 0.4 \) and a grid spacing of \( h = 1/64 \). On the right, we observe that the spatial distribution of the error closely follows the magnitude of the computed \( v_y \). The plot on the left of Fig. 12 also shows the values of the computed \( v_y \) outside of \( \Omega \) (i.e. “inside” the wall). We note that the values at \( r > R + 2h \) are ignored and kept as they were set initially (mirroring values as in ICE1).

We can rewrite Eq. (30) as a solution to Eq. (2) with a DBC \( B_D(\theta, t) = v_y(R, \theta, t) = a R \cos(\theta) \cos(\theta) \) and use this as a benchmark problem to compare the methods for DBC. Fig. 13 shows the \( L_2 \) errors for the \( y \)-component of the velocity \( v_y \) with a cylinder radius of \( R = 0.4 \). The errors can be compared with the bottom left plot in Fig. 11, where the same problem was solved with an RBC. The convergence rates of all methods for DBC with a forcing term (i.e. phase field and penalization) are close to first order. The phase field method in the limit for DBC \( (B_R = \alpha_R B_D) \) with \( \alpha_R \gg 1 \) is not visible in the plot as it performs exactly the same as the Brinkman penalization method (“Penalty (DBC)”). The same applies to the spread interface penalization method in the limit for DBC which is hidden by the “Penalty Spread (DBC)” which has identical errors. The main difference between those methods is that the methods “in the limit for DBC” do not require a solution to Eq. (3) for values of \( \phi > h \) outside of \( \Omega \). They might therefore allow for the same accuracy with less computational effort. The “Penalty Exterior (DBC)” method performs worse and the other penalty methods should be preferred. The level set methods, described in Section 3.6, outperform the other methods in this test case and show a convergence close to second order. We note that the errors for the level set methods are lower than what we obtained with the “Phase field (RBC)” method in Fig. 11. For this test case, we observe that the level set methods have the best accuracy due to their special discretizations.
of the spatial derivatives close to the interface. However, the special discretizations are more complex to implement as they require the identification of stencils which cross the boundary and branching operations to get a separate control flow for those points. Those branching operations limit the efficiency of the implementation on modern computer architectures. Furthermore, the methods cannot easily be extended to flux boundary conditions in 2D or 3D and the special stencils can lead to stability issues or might fail depending on how the surface cuts through the mesh (as explained in Section 3.6).

4.4. 1D flow over an oscillating wall

A simplified variant of the test in Section 4.3 is the flow in the half-space \( y \geq y_0 \) with the wall moving in \( x \)-direction as \( v_x(y_0, t) = U(t) = \cos(at) \). A solution to Eq. (1) in \( \Omega = \{ y \mid y \geq y_0 \} \) with \( \alpha_R = 0 \) is then given by

\[
\begin{align*}
  u(y, t) &= v_x(y, t) = \exp(-\lambda(y-y_0)) \cos(at - \lambda(y-y_0)), \\
  B_R(t) &= \frac{a}{2\lambda} (\cos(at) - \sin(at)), \\
  \lambda &= \sqrt{a/(2D)}.
\end{align*}
\]

We will consider this in vorticity formulation where we have

\[
\begin{align*}
  u(y, t) &= \omega(y, t) = \lambda \exp(-\lambda(y-y_0)) (\cos(at - \lambda(y-y_0)) - \sin(at - \lambda(y-y_0))), \\
  B_R(t) &= -a \sin(at) = \frac{du}{dt}.
\end{align*}
\]

We solve this in \( \tilde{\Omega} = [0, 1] \) and choose \( D = 0.001 \) and \( a = 0.628 \) (see Fig. 14).
As errors that the solution of expansions the solution as the surface of the wall is also the expression for the phase field method but both methods show a convergence close to second order. We also observe a convergence of second order for the “FDM 1 (NBC)” variant even though it uses a first order stencil to discretize the Laplacian next to the boundary.

5. Error analysis for phase field method

In this section, we estimate the errors made in the phase field method described in Section 3.2. We approximate a solution to Eq. (1) by solving Eq. (6) in an extended domain with \( \tilde{u}_\Omega = u + O(h^p) \), where \( u \) is the exact solution and \( p \) is the order of the method. The results suggest that the phase field method has second order accuracy (\( p = 2 \)) for \( \alpha_R = 0 \). In the following, we will analyze the error by computing \( p \) when taking the time-derivative of the integral of \( u \) over \( \Omega \) as in

\[
\frac{\partial}{\partial t} \int_{\Omega} \tilde{u} \, dx = \frac{\partial}{\partial t} \int_{\Omega} u \, dx + O(h^p).
\]

We therefore analyze the errors in an integral formulation, which is in contrast to the method of matched asymptotic expansions which was used to prove convergence for the similar “diffuse domain” approach [37].

First, we will have to expand the left-hand side to the extended domain \( \tilde{\Omega} \). We will use the phase field function to show that \( \int_{\tilde{\Omega}} \tilde{u} \, dx = \int_{\tilde{\Omega}} \psi \tilde{u} \, dx + O(w^2) \), where \( w = O(h) \) defines the width of the phase field function.

To simplify integrals within the band where \( 0 < \psi_w < 1 \), we introduce a local coordinate system \((s, \phi)\) in direction of the surface and the normal given by the level set function \( \phi \) with \( |\phi| < w \). We can compute the integral in that band around \( \Gamma \) as

\[
\int_{\phi=-w}^{w} dx = \int_{\Gamma} ds \int_{\phi=-w}^{w} d\phi.
\]

This introduces curvature-dependent errors which we will ignore in the following analysis. One can reason about those errors by considering Eq. (35) with \( \Gamma \) being the surface of a sphere of radius \( R \). Locally, this should be similar to integrating \( \int_{\phi=-w}^{w} dx \) for an infinitesimal surface element \( ds \) with curvature \( \kappa = 1/R \). In spherical coordinates \((r, \theta, \phi)\) we have

\[
\int_{\phi=-w}^{w} dx = \int_{0}^{2\pi} \int_{0}^{\pi} \int_{R-w}^{R+w} r^2 \sin(\theta) \, dr \, d\theta \, d\phi = \int_{0}^{2\pi} \int_{0}^{\pi} \int_{R-w}^{R+w} R^2 \sin(\theta) \, dr \, d\theta \, d\phi = \int_{0}^{2\pi} \int_{0}^{\pi} \int_{R-w}^{R+w} (r/R)^2 \, dr,
\]

which gives the local coordinate system \( \kappa = 1/R \) becomes

\[
\int_{\phi=-w}^{w} dx = \int_{\Gamma} ds \int_{\phi=-w}^{w} \left( \frac{R + \phi}{R} \right)^2 \, d\phi = \int_{\Gamma} ds \int_{\Gamma} \left( 1 + \kappa \phi \right)^2 \, d\phi = \int_{\Gamma} ds \int_{\Gamma} d\phi + O(w^3)O(\kappa^2).
\]

As long as the maximal curvature \( \kappa \) is \( \ll 1/w \) we can safely ignore this error. Also, we note that if we only consider the part of the band within \( \Omega \), we get

\[
\int_{\phi=0}^{\phi=w} dx = \int_{\Gamma} ds \int_{\phi=0}^{\phi=w} d\phi + O(w^2)O(\kappa),
\]

which is also negligible as long as the maximal curvature is \( \ll 1/w \).
Given the properties of a phase field function \( \psi_w = \psi(\phi/w) \) like the one in Eq. (5) and a sufficiently smooth function \( F = F(s, \phi) \), we can derive the following:

\[
\begin{align*}
\int_{\phi=-w}^{w} \psi_w F \, dx &= w \int_{\Gamma} F \, ds + O(w^2) \int_{\Gamma} \frac{\partial F}{\partial \phi} \, ds + O(w^3) = \int_{\phi=-w}^{0} F \, dx + O(w^2), \\
\int_{\phi=-w}^{w} \psi_w' F \, dx &= -w \int_{\Gamma} F \, ds + O(w^2) \int_{\Gamma} \frac{\partial^2 F}{\partial \phi^2} \, ds + O(w^4), \\
\int_{\phi=-w}^{w} \frac{\left(\psi_w\right)^2}{K} F \, dx &= \int_{\Gamma} F \, ds + O(w^2) \int_{\Gamma} \frac{\partial^2 F}{\partial \phi^2} \, ds + O(w^4).
\end{align*}
\]

Eq. (39) requires \( \psi(x) = 1 - \psi(-x) \), Eq. (40) requires \( \int \psi_w'(x) \, dx = -1 \) and Eq. (41) requires the definition of \( K = \int_{\Omega} \left(\psi_w^2\right)^2/A \, dx \). Also note that the higher order terms drop if we have \( \frac{\partial^i F}{\partial \phi^i} = 0 \) for \( i \geq 1 \). In the following, we will use the phase field function \( \psi_w \) as defined in Eq. (5), which is 1 for \( \phi \leq -w \) and 0 for \( \phi \geq w \).

We can now use Eq. (39) to expand \( \int_{\Omega} u \, dx \) to the extended domain \( \Omega \):

\[
\int_{\Omega} u \, dx = \int_{\phi < -w} u \, dx + \int_{\phi = -w}^0 u \, dx
\]

\[
= \int_{\phi < -w} u \, dx + \int_{\phi = -w}^{w} \psi_w u \, dx + O(w^2)
\]

\[
= \int_{\phi < -w} \psi_w u \, dx + \int_{\phi = -w}^{w} \psi_w u \, dx + \int_{\phi > w} \psi_w u \, dx + O(w^2)
\]

\[
= \int_{\phi < -w} \psi_w u \, dx + O(w^2).
\]

If we initialize \( u \) by mirroring the values outside of \( \Omega \) (ICE1 in Section 3.5 and Fig. 2), we minimize the error in the initial approximation \( \int_{\Omega} \psi_w u \, dx = \int_{\Omega} u \, dx \). We now assess the approximation in Eq. (34). We apply Eq. (42) and Eq. (5) to the left-hand side of Eq. (34) and simplify using Eq. (41):

\[
\frac{\partial}{\partial t} \int_{\Omega} u \, dx = \frac{\partial}{\partial t} \int_{\Omega} \psi_w u \, dx + O(w^2) = \int_{\Omega} \frac{\partial}{\partial t} (\psi_w u) \, dx + O(w^2) = \int_{\Omega} \psi_w \frac{\partial u}{\partial t} \, dx + O(w^2)
\]

\[
= \int_{\phi < w} \nabla \cdot (\psi_w D \nabla u) + (B_R - \alpha R \tilde{u}) \left(\frac{\psi_w^2}{K}\right) \, dx + O(w^2)
\]

\[
= \int_{\phi = w} \left(\psi_w D \nabla u\right) \cdot n \, ds + \int_{\phi = -w}^{w} (B_R - \alpha R \tilde{u}) \left(\frac{\psi_w^2}{K}\right) \, dx + O(w^2)
\]

\[
= \int_{\Gamma} (B_R - \alpha R \tilde{u}) \, ds + O(w^2).
\]

We then apply Eq. (1) to the right-hand side of Eq. (34):

\[
\frac{\partial}{\partial t} \int_{\Omega} u \, dx = \frac{\partial u}{\partial t} \int_{\Gamma} \nabla \cdot (D \nabla u) \, ds = \int_{\Gamma} (D \nabla u) \cdot n \, ds
\]

\[
= \int_{\Gamma} (B_R - \alpha R u) \, ds.
\]
We finally compare Eq. (43) and Eq. (44) using \( \tilde{u} | \Omega = u + O(h^p) \) to obtain
\[
\frac{\partial}{\partial t} \int_\Omega \tilde{u} \, dx = \frac{\partial}{\partial t} \int_\Omega u \, dx + O(w^2) + O(h^p). \tag{45}
\]

Given that we choose \( w = O(h) \), we can conclude that \( p = 2 \) in Eq. (34) and we therefore have a second order accurate method. To explain the observed \( O(h) \) error when \( \alpha_K \neq 0 \), we have to reassess Eq. (41) which we used with \( F = B - \alpha_K \tilde{u} \) in Eq. (43). Eq. (41) requires \( F \) to be sufficiently smooth for \(-w \leq \phi \leq w\), but our approximate solution \( \tilde{u} \) might not be smooth enough. Concretely, if \( \tilde{u} = u + O(h^2) \) with a smooth \( u \), we might have a piecewise linear function \( \tilde{u} \). In Eq. (41) this discontinuity in the first derivative of \( \tilde{u} \) (and thus \( F \)) leads to an additional \( O(w) \) error. This error translates into an \( O(w) \) error in Eq. (45) and can explain the observed convergence for \( \alpha_K \neq 0 \).

We note that this analysis does not include any errors from the numerical quadrature of the integrals.

The derivation above considers integrals over the domain \( \Omega \). Similarly, we can look at small volume elements spanning an infinitesimal surface element \( ds \) and \(-w \leq \phi \leq w\) and get the same error estimates locally while we get 0 errors in the remainder of \( \Omega \).

5.1. Extension for moving interfaces

So far, we considered the diffusion equation within a static domain \( \Omega \). In this section, we show that Eq. (6) can also be used to solve Eq. (1) in a moving and deforming domain. We introduce \( v_f \) as the velocity of the interface \( \Gamma(t) = \partial \Omega(t) \).

We assume that the level set function \( \psi \) is kept as a signed distance to the moving interface \( \Gamma(t) \). This can be realized by extending \( v_f \) in normal direction into a narrow band around \( \Gamma(t) \) within \( \Omega \) and then solving the level set equation [25]:
\[
\frac{\partial \psi}{\partial t} = -v_f \cdot \nabla \psi. \tag{46}
\]

The level set function needs to be periodically reinitialized to keep its signed distance property [28]. We can do this by solving
\[
\frac{\partial \psi}{\partial \tau} + \text{sign}(\psi_0)(|\nabla \psi| - 1) = 0, \tag{47}
\]

to steady state, where \( \psi_0 = \phi(\tau = 0) \). Here, we will assume that \( \phi \) is a signed distance function at all times.

We have chosen the phase field function \( \psi' \) as a function of the level set (i.e. \( \psi' = \psi \)). We can therefore derive
\[
\frac{\partial \psi'}{\partial t} = \psi' \frac{\partial \phi}{\partial t} = -v_f \cdot \nabla \phi \psi' = -v_f \cdot \nabla \psi'. \tag{48}
\]

where we used \( \nabla \psi' = \nabla \phi \psi' \). This leads to
\[
\frac{\partial}{\partial t} (\psi' \tilde{u}) = \psi' \frac{\partial \tilde{u}}{\partial t} - \tilde{u} v_f \cdot \nabla \psi'. \tag{49}
\]

We can insert Eq. (49) into Eq. (43), where we add
\[
\frac{\partial}{\partial t} \int_\Omega \tilde{u} \, dx = (43) + \int_\Omega -\tilde{u} v_f \cdot \nabla \psi' \, dx
\]
\[
= (43) + \int_{\phi=-w}^{w} -\tilde{u} v_f \cdot \nabla \psi' \, dx
\]
\[
= (43) + \int_{\Gamma} \tilde{u} v_f \cdot \mathbf{n} \, ds + O(w^2), \tag{50}
\]

where we used Eq. (40) and \( \mathbf{n} = \nabla \phi \). In Eq. (44) on the other hand we have to consider the moving domain \( \Omega \) and add
\[
\frac{\partial}{\partial t} \int_\Omega u \, dx = (44) + \int_{\Gamma} u v_f \cdot \mathbf{n} \, ds. \tag{51}
\]

Therefore, the moving interface only adds an additional \( O(w^2) \) error to Eq. (45) and does not worsen the accuracy of the method. We would like to point out though that given our observations with Robin boundary conditions, we might be introducing an \( O(w) \) error in Eq. (50) due to the possible lack of smoothness of \( \tilde{u} \). Alternatively, we can write a variant to Eq. (6), where we solve for \( \psi' \tilde{u} \) as follows:
\[ \frac{\partial}{\partial t} \left( \psi_w \tilde{u} \right) = \nabla \cdot \left( \psi_w D \nabla \tilde{u} \right) + (B_R - \alpha_R \tilde{u} + \tilde{u} \mathbf{v}_f \cdot \mathbf{n}) \frac{(\psi'_w)^2}{K} \text{ in } \tilde{\Omega}, \]

suitable (e.g. periodic) BC for \( \tilde{u} \) on \( \partial \tilde{\Omega} \). \hspace{1cm} (52)

We can easily see that Eq. (52) inserted in Eq. (43) leads to the same result shown in Eq. (50) and hence both variants have the same accuracy. As we show in the next section, this alternative formulation might remove the \( O(w) \) error mentioned above, if our governing equations include a convection term which moves \( u \) together with the interface.

5.2. Extension for generic flux and reaction terms

Given the results in the previous section, we consider the solution to equations of the form

\[ \frac{\partial \tilde{u}}{\partial t} = \nabla \cdot \mathbf{F}(u) + R(u) \text{ in } \Omega(t), \]

\[ \mathbf{F}(u) \cdot \mathbf{n} = B_F(u) \text{ on } \Gamma(t) \]

where \( \mathbf{F}(u) \) is the flux, \( R(u) \) is the reaction term and \( B_F(u) \) is the given flux boundary condition. In Eq. (1) for instance, we set \( \mathbf{F}(u) = D \nabla u \), \( R(u) = 0 \) and \( B_F(u) = B_R - \alpha_R u \). We then introduce the phase field function \( \psi_w \) and solve a generalized version of Eq. (6):

\[ \psi_w \frac{\partial \tilde{u}}{\partial t} = \nabla \cdot \left( \psi_w \mathbf{F}(... \right) + \psi_w R(\tilde{u}) + B_F(\tilde{u}) \frac{(\psi'_w)^2}{K} \text{ in } \tilde{\Omega}, \]

suitable (e.g. periodic) BC for \( \tilde{u} \) on \( \partial \tilde{\Omega} \). \hspace{1cm} (54)

Alternatively, we can generalize Eq. (52) and solve for \( \psi_w u \) as in

\[ \frac{\partial}{\partial t} \left( \psi_w \tilde{u} \right) = \nabla \cdot \left( \psi_w \mathbf{F}(\tilde{u}) \right) + \psi_w R(\tilde{u}) + (B_F(\tilde{u}) + \tilde{u} \mathbf{v}_f \cdot \mathbf{n}) \frac{(\psi'_w)^2}{K} \text{ in } \tilde{\Omega}, \]

suitable (e.g. periodic) BC for \( \tilde{u} \) on \( \partial \tilde{\Omega} \). \hspace{1cm} (55)

where \( \mathbf{v}_f \) is the velocity of the interface \( \Gamma(t) = \partial \Omega(t) \) as in the previous section. The boundary condition term \( B_F(\tilde{u}) + \tilde{u} \mathbf{v}_f \cdot \mathbf{n} \) can then be simpler than \( B_F(\tilde{u}) \) in Eq. (54), if \( \mathbf{F}(\tilde{u}) \) contains a convection term. Consider a convection–diffusion equation with a Neumann boundary condition \( D \nabla u \cdot \mathbf{n} = B_N \). There, we have \( \mathbf{F}(u) = D \nabla u - u \mathbf{v} \) and \( B_F(u) = B_N - u \mathbf{v} \cdot \mathbf{n} \) in Eq. (53). We furthermore assume that the velocity field for \( u \) is consistent with the motion of the interface (i.e. \( \mathbf{v} = \mathbf{v}_f \)). We then notice that the boundary condition term \( B_F(\tilde{u}) + \tilde{u} \mathbf{v}_f \cdot \mathbf{n} \) in Eq. (55) simplifies to \( B_N \). This removes the dependency on \( \tilde{u} \) for the boundary condition term and removes the potential \( O(w) \) error which we observed for Robin boundary conditions and which we discussed in the previous section.

We can derive the same \( O(w^2) \) error for the general flux term \( \mathbf{F}(u) \) in Eq. (53) as we had for \( D \nabla u \) in Eqs. (43)–(44). The addition of the reaction term \( R(u) \), on the other hand, requires some attention. In Eq. (43), we add

\[ \frac{\partial}{\partial t} \int_{\tilde{\Omega}} \tilde{u} \, d\mathbf{x} = \text{(43)} + \int_{\tilde{\Omega}} \psi_w R(\tilde{u}) \, d\mathbf{x} \]

\[ = \text{(43)} + \int_{\phi = -w} \int_{\phi = -w} R(\tilde{u}) \, d\mathbf{x} + \int_{\phi = w} \psi_w R(\tilde{u}) \, d\mathbf{x} \]

\[ = \text{(43)} + \int_{\phi = -w} \int_{\phi = -w} R(\tilde{u}) \, d\mathbf{x} + \int_{\phi = w} R(\tilde{u}) \, d\mathbf{x} + O(w^2) \]

\[ = \text{(43)} + \int_{\Omega} R(\tilde{u}) \, d\mathbf{x} + O(w^2), \]

where we used Eq. (39). In Eq. (44), we add

\[ \frac{\partial}{\partial t} \int_{\tilde{\Omega}} u \, d\mathbf{x} = \text{(44)} + \int_{\Omega} R(u) \, d\mathbf{x} \]

Therefore, the reaction term only adds an additional \( O(w^2) \) error to Eq. (45) and keeps the second order accuracy of the method. For this error estimate to be correct, we need to obey some requirements for the values and terms which are evaluated outside of the domain \( \Omega (\phi > 0) \). \( \tilde{u} \) needs to be continuous (i.e. in \( C^0 \)) for \( -w \leq \phi \leq w \), \( \psi_w \mathbf{F}(\tilde{u}) \) needs to be 0 for \( \phi \geq w \), \( R(\tilde{u}) \) needs to be continuous (i.e. in \( C^0 \)) for \( -w \leq \phi \leq w \) and the boundary condition term \( B_F(\tilde{u}) \) in Eq. (54) and \( B_F(\tilde{u}) + \tilde{u} \mathbf{v}_f \cdot \mathbf{n} \) in Eq. (55) respectively) needs to be differentiable (i.e. in \( C^1 \)) for \( -w \leq \phi \leq w \).
6. Error analysis for penalization methods

Following the approach used in Section 5, we analyze why the penalty methods described in Section 3.3 perform worse than the phase field method. We approximate a solution to Eq. (1) using Eq. (3) with the parameters defined in Eq. (8) ("Penalty Spread") or Eq. (10) ("Penalty") and \( \eta = 0 \). We again assume that our computed solution \( \tilde{u} \) approximates the exact solution \( u = u + O(h^p) \), where \( p \) is the order of the method.

We observe that for the "Penalty Spread" method, we can compare \( \int_{\Omega} \tilde{u} \, d\mathbf{x} \) with \( \int_{\Omega} u \, d\mathbf{x} \) as follows:

\[
\frac{\partial}{\partial t} \int_{\Omega} \tilde{u} \, d\mathbf{x} = \int_{\Omega_h} \frac{\partial \tilde{u}}{\partial t} \, d\mathbf{x} + \int_{\Omega,\Gamma} \frac{\partial u}{\partial t} \, d\mathbf{x} + \int_{\Omega} \frac{\partial \tilde{u}}{\partial t} \, d\mathbf{x} \\
= \int_{\Omega_h} \nabla \cdot (D \nabla \tilde{u}) \, d\mathbf{x} + \int_{\Omega,\Gamma} \nabla \cdot (D \nabla \tilde{u}) \, d\mathbf{x} + \int_{\Omega} \frac{B_R - \alpha_R \tilde{u}}{\epsilon} \, d\mathbf{x} \\
= \int_{\Omega_h} \frac{B_R - \alpha_R \tilde{u}}{\epsilon} \, d\mathbf{x} = \int_{\Gamma} (B_R - \alpha_R \tilde{u}) \, d\mathbf{s} + O(h^p) = \frac{\partial}{\partial t} \int_{\Omega} u \, d\mathbf{x} + O(h^p),
\]

where we have chosen a perfect \( \epsilon \) according to Eq. (9). We also notice that the only relevant quantities are in \( \Omega_S = \Omega_h \cup \Omega_h,\Gamma \). If we have \( \Omega_S = \Omega \), we have \( \int_{\Omega} \tilde{u} \, d\mathbf{x} = \int_{\Omega} u \, d\mathbf{x} \) and we capture the boundary condition perfectly according to Eq. (34). In general though, we have a staircase approximation of the domain and we instead get

\[
\int_{\Omega} u \, d\mathbf{x} = \int_{\Omega} u \, d\mathbf{x} + O(h).
\]

Unless we perfectly capture the domain as was the case for 1D tests with \( R = 0.25 \), we will always introduce an \( O(h) \) error with this method.

For the "Penalty" method, we similarly observe that we can compare \( \int_{\Omega} u \, d\mathbf{x} \) with \( \int_{\Omega} \tilde{u} \, d\mathbf{x} \) using Eq. (40) as follows:

\[
\frac{\partial}{\partial t} \int_{\Omega} \tilde{u} \, d\mathbf{x} = \int_{\Omega} \frac{\partial \tilde{u}}{\partial t} \, d\mathbf{x} \\
= \int_{\phi < w} \nabla \cdot (\psi_w D \nabla \tilde{u}) - (B_R - \alpha_R \tilde{u}) \psi'_w \, d\mathbf{x} \\
= \int_{\phi = w} (\psi_w D \nabla \tilde{u}) \cdot \mathbf{n} \, d\mathbf{s} - \int_{\phi = -w} (B_R - \alpha_R \tilde{u}) \psi'_w \, d\mathbf{x} \\
= \int_{\Gamma} (B_R - \alpha_R \tilde{u}) \, d\mathbf{s} + O(w^2) = \frac{\partial}{\partial t} \int_{\Omega} u \, d\mathbf{x} + O(h^2) + O(h^p).
\]

Similarly to the "Penalty Spread" method, we notice that the only relevant quantities of \( \tilde{u} \) are in \( \Omega_P \) with \( \phi < w \). The problem is that for the method to only have the \( O(w^2) \) error, we would need to compare \( \int_{\Omega_P} \tilde{u} \, d\mathbf{x} \) with \( \int_{\Omega} \tilde{u} \, d\mathbf{x} \) such that we can use the result of Eq. (60) in Eq. (34). Yet, this introduces an \( O(w) = O(h) \) error in

\[
\int_{\Omega_P} \tilde{u} \, d\mathbf{x} = \int_{\Omega} \tilde{u} \, d\mathbf{x} + w \int_{\Gamma} \tilde{u} \, d\mathbf{s} + O(w^2) = \int_{\Omega} \tilde{u} \, d\mathbf{x} + O(w).
\]

7. Conclusions

We have presented a comparative study of the accuracy of penalization and phase field methods to solve the diffusion equation with Robin (RBC), Neumann (NBC) and Dirichlet (DBC) boundary conditions. We consider 1D and 2D benchmark problems as well as the flow induced by an oscillating spinning cylinder and an oscillating wall. Our study elucidates the trade-off between accuracy and computational efficiency in different numerical approximations.

We observe that methods with special discretizations of the spatial derivatives close to the interface ("Level set" for DBC and "FDI" for NBC) have a convergence of second order. However, those methods are not easy to implement and cannot easily be extended to flux boundary conditions in 2D or 3D. Also, the construction of the special stencils can be tedious, lead to stability issues and might fail depending on how the surface cuts through the mesh.

On the other side, among the methods without special discretizations of the spatial derivatives, we find that the phase field method can also reach a convergence of second order for problems with NBC while it avoids the challenges listed
above. For NBC, the phase field method consistently outperforms the penalty methods which show a first order convergence or worse. In Eulerian solid mechanics for finite-strain elasticity [6], traction BC are analogous to NBC for the diffusion equation. We therefore expect the phase field method to be suited for such problems.

Both the penalty and phase field methods show a first order convergence for RBC and DBC. The error of the phase field method was usually lower for RBC and performed identically to the penalty method for DBC. We believe that the lessons learned through the diffusion equation can be extended to solutions of the Navier–Stokes equations. For example in flows past active swimmers [3], the penalty method for DBC is preferable to impose velocity boundary conditions as it has a simpler implementation and exhibits the same accuracy as the phase field method.

We distinguish methods with special stencils (for the “Level set” and “FDM” methods) and boundary aware grid cells (for the “Penalty Spread” and “Penalty Exterior” methods). In these methods, we do not perform the same computation for all grid points but instead need a control flow in the code to choose the desired discretization of the Laplacian or the parameters to be used. The extended (or fictitious) domain methods (“Phase field” and “Penalty”), on the other hand, can work without such a control flow. One might still choose to have a control flow to remove parts of the computation like the forcing term when it is known to be zero, but the equations are defined everywhere and the used parameters vary smoothly in space. This has two advantages: First, the resulting operations are regular and this can lead to efficient implementations on modern computer architectures. Second, the equations can be mathematically transformed to derive PDEs for other variables of interest. In the field of vortex methods, for instance, one can take the curl of the Navier–Stokes equations in velocity-formulation to derive equations for the vorticity. This was used to derive vortex methods with Brinkman penalization to solve fluid–solid interaction problems [3,40]. Such a derivation may not have been possible with a method that requires discrete elements.

The phase field method can be extended to PDEs in moving domains with different fluxes and with reaction terms (see Section 5.1 and Section 5.2). Future work will focus on the analysis of the method for convection–diffusion problems as well as largely deforming solids [6,49]. The level set methods for Dirichlet boundary conditions on the other hand might be an alternative to penalty methods for the solution of flow problems with a given velocity at the interface. The applicability of those methods to vortex methods and the impact on the computational efficiency and the stability of flow solvers is a subject of ongoing investigations.

Acknowledgements

The authors thank Anthony Leonard and Georges-Henri Cottet for fruitful discussions on all the methods used in this study. This work was supported by the Swiss National Supercomputing Centre (CSCS) (project number s436), the Swiss National Science Foundation (SNF) (200021-134999) and the European Research Council (ERC) (2-73985-14).

References
