Superconvergent second order Cartesian method for solving free boundary problem for invadopodia formation

Olivier Gallinato, Clair Poignard

HAL Id: hal-01483484
https://hal.inria.fr/hal-01483484

Submitted on 6 Mar 2017

Superconvergent Second Order Cartesian Method for Solving Free Boundary Problem for Invadopodia Formation

Olivier Gallinato[a], Clair Poignard[a,1]

[a]*Team MONC, INRIA Bordeaux-Sud-Ouest, Institut de Mathématiques de Bordeaux, CNRS UMR 5251 &
Université de Bordeaux, 351 cours de la Libération, 33405 Talence Cedex, France.*

Abstract

In this paper, we present a superconvergent second order Cartesian method to solve a free boundary problem with two harmonic phases coupled through the moving interface. The model recently proposed by the authors and colleagues describes the formation of cell protrusions. The moving interface is described by a level set function and is advected at the velocity given by the gradient of the inner phase. The finite differences method proposed in this paper consists of a new stabilized ghost fluid method and second order discretizations for the Laplace operator with the boundary conditions (Dirichlet, Neumann or Robin conditions). Interestingly, the method to solve the harmonic subproblems is superconvergent on two levels, in the sense that the first and second order derivatives of the numerical solutions are obtained with the second order of accuracy, similarly to the solution itself. We exhibit numerical criteria on the data accuracy to get such properties and numerical simulations corroborate these criteria. In addition to these properties, we propose an appropriate extension of the velocity of the level-set to avoid any loss of consistency, and to obtain the second order of accuracy of the complete free boundary problem. Interestingly, we highlight the transmission of the superconvergent properties for the static subproblems and their preservation by the dynamical scheme. Our method is also well suited for quasistatic Hele-Shaw-like or Muskat-like problems.

Keywords: Finite differences on Cartesian grids, Superconvergence, Interface conditions, Free boundary problem
2000 MSC: 65M06, 65M12, 92C37

1. Introduction

Early-stage carcinoma are mostly confined to the epithelium, which is separated from the underlying tissue by a basement membrane composed of dense fibers of extracellular matrix (ECM).

Email addresses: `olivier.gallinato@inria.fr` (Olivier Gallinato), `clair.poignard@inria.fr` (Clair Poignard)

[1]Corresponding author

In order to cross this tight barrier, metastatic cells use a complex internal machinery, named *invadopodia*, which lies on the actin polymerization and that leads to the formation of proteolytic, protrusive and very localized subcellular structures. *Invadopodia* are elongated shapes, which are formed during cell invasion and mesenchymal migration. This phenomenon is the crucial and initiating point in the metastasic process, which is the major cause of death from cancer. The authors and colleagues proposed recently a free boundary problem to model invadopodia and more generally for cell protrusion formation [16]. In this paper, we present a superconvergent second order method on Cartesian grid to solve this quite complex free boundary value problem. Our finite difference method is based on the derivation of appropriate superconvergent schemes: wide-stencils are proposed to reach specific superconvergence properties for the solutions of the Poisson problem. As a result, the global method leads to the second-order accuracy of the moving interface, its normal vector and even its curvature. The accuracy on the curvature will be of major interest in further study for the use of interface regularization techniques in order to model subsequent phenomena involved in cell migration, as myosin-dependant protrusion retraction, for instance.

The purpose of the study is to simulate accurately cell protrusion formation, however the domain of applications can be extended to free-boundary problems arising from physics or biology such as Hele-Shaw, Muskat or quasistatic two-phase Stefan like problems.

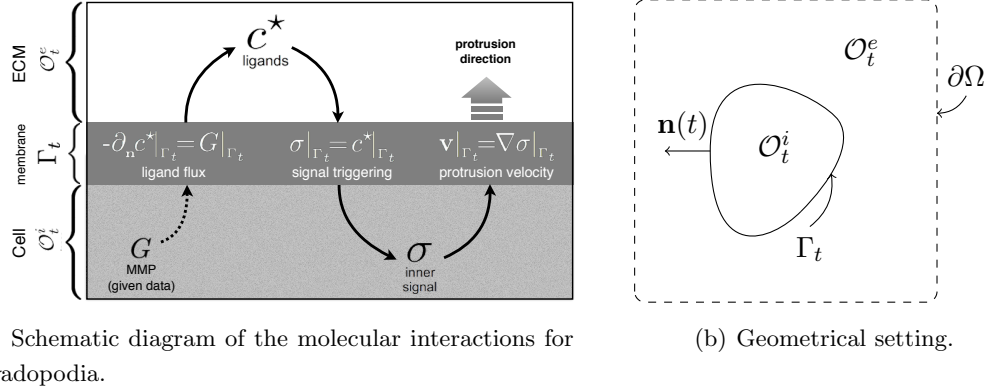### 1.1. Free-boundary problem for invadopodium formation

The invadopodium process relies on a coupled dynamics between the outer and the inner of the cell. Let us present briefly the new model of the phenomenon detailed in [16]. Specific enzymes (MT1-MMPs) produced by the cell membrane degrade the extracellular matrix (ECM), producing ligands that diffuse and bind to membrane receptors. In response, cell generates a signal which diffuses inside the cell and, which leads to actin polymerization: rigid filaments are polymerized, oriented towards the location of the detected ligand. The force exerted by the filaments on the membrane generates a protrusion which grows at the velocity of the filament polymerization. The scheme of the process is given by Figure 1(a), while the geometrical framework is detailed in Figure 1(b). At any time t, the cell membrane is parameterized by the map $\gamma(t,.)$ defined on the torus $\mathbb{T} = \mathbb{R}/2\pi\mathbb{R}$:

$$\Gamma_t = \{\gamma(t,\theta), \quad \theta \in \mathbb{T}\}.$$

The cell cytoplasm $\mathcal{O}_t^i$ is the domain enclosed by Γ_t and the ECM is the outer domain

$$\mathcal{O}_t^e = \Omega \setminus \mathcal{O}_t^i.$$

Assume the flux of MT1-MMP enzymes $g(t,\cdot)$ be given at any time on the cell membrane. It generates a flux of the degraded matrix (called ligands and denoted by $c^\star$) on the cell boundary, and these ligands diffuse in the extracellular medium as described by equations (1a)–(1b). When bound to the cell membrane, the ligands generate a signal σ, which diffuses inside the cell, as

2

(a) Schematic diagram of the molecular interactions for invadopodia.

(b) Geometrical setting.

Figure 1: Schematic diagram of the molecular interactions involved in our model and geometrical settings. The cell $\mathcal{O}_t^i$ is imbedded in the bath $\mathcal{O}_t^e$. The whole domain Ω does not depend on the time variable. It is defined by $\Omega = \mathcal{O}_t^e \cup \overline{\mathcal{O}_t^i}$

accounted for in equations (1c)–(1d). The cell membrane motion is described by equation (1e).

Degradation of the ECM:

$$\Delta c^\star = 0, \quad x \in \mathcal{O}_t^e, \tag{1a}$$

$$c^\star|_{\partial\Omega} = 0, \quad -\partial_{\mathbf{n}} c^\star|_{\Gamma_t} = g|_{\Gamma_t}. \tag{1b}$$

Generation of the inner signal for actin polymerization:

$$\Delta \sigma = 0, \quad t \in [0, T], \quad x \in \mathcal{O}_t^i, \tag{1c}$$

$$\sigma|_{\Gamma_t} = c^\star|_{\Gamma_t}. \tag{1d}$$

Motion of the cell membrane:

$$\partial_t \gamma(t, \theta) = \nabla \sigma(\gamma(t, \theta)), \quad \theta \in \mathbb{T}, \quad \text{and} \quad \Gamma_t = \{\gamma(t, \theta), \theta \in \mathbb{T}\}. \tag{1e}$$

The interested reader will refer to [16] for further details about biological phenomena, and modeling hypotheses. The theoretical analysis of the free-boundary problem is also performed in this article. In particular, the well-posedness of the free-boundary problem in Sobolev spaces is precisely proven, provided strictly positive boundary data $g|_{\Gamma_t}$. The proof is based on the explicit characterization of Dirichlet-to-Neumann maps thanks to complex analysis tools. Then, appropriate change of unknowns and the quasilinearization of the problem leads to a parabolic-type equation, which prevents the loss of regularity and thus provides the well-posedness.

1.2. Eulerian approach for the numerical formulation

In order to avoid any remeshing of the domain, specific finite difference methods on Cartesian grids regarding the Eulerian formulation of problem (1) is proposed. Such a formulation consists in detecting the cell membrane by a level-set function $\psi(t, x)$ defined in the entire domain Ω, and

3

such that

$$\Gamma_t = \{x : \boldsymbol{\psi}(t,x) = 0\}, \quad \mathcal{O}_t^i = \{x : \boldsymbol{\psi}(t,x) < 0\}, \quad \mathcal{O}_t^e = \{x : \boldsymbol{\psi}(t,x) > 0\}. \tag{2}$$

Equation (1e) is then changed into a transport equation for the level-set at the velocity $\mathbf{v}$, which derives from the gradient of σ as stated in (1e). Here arises a first numerical difficulty, since the velocity of the Lagrangian formalism is only defined on the interface, *i.e.* $\mathbf{v}|_{\Gamma_t} = \nabla\sigma|_{\Gamma_t}$. An extension of the velocity to the whole domain is thus required in order to solve the transport equation (3a). We thus replace equation (1e) by

$$\partial_t \boldsymbol{\psi} + \mathbf{v} \cdot \nabla \boldsymbol{\psi} = 0, \quad t \geq 0, \quad x \in \Omega, \tag{3a}$$

where the velocity $\mathbf{v}$ is extended to Ω from the data $\nabla\sigma|_{\Gamma_t}$ as follows:

$$(\nabla\boldsymbol{\psi} \cdot \nabla)\mathbf{v} = 0, \quad \forall x \in \Omega, \quad \text{with} \quad \mathbf{v}|_{\{\boldsymbol{\psi}(t,x)=0\}} = \nabla\sigma|_{\{\boldsymbol{\psi}(t,x)=0\}}. \tag{3b}$$

Remark 1 (Comment on the velocity extension). *The solution to problem* (1a)–(1b)–(1c)–(1d)–(3) *depends on the choice of the extension, however only the level-set function would change, but the location of Γ_t and the quantities σ, and $c^\star$ would be the same. Therefore the choice of the extension operator is only guided to provide appropriate numerical properties.*

As explained in the following, the extension (3b) *will provide the good properties to reach second order of accuracy on the interface, its normal and even its curvature. However it is obvious that at any time step, shocks are generated by this extension, and merge with ridges of the level set function, initialized as the signed distance function. If close to the interface, shocks may dramatically decrease the accuracy of the numerical solution. To avoid such drawbacks it is necessary to refine the mesh, for instance thanks to local adaptive meshing such as quadtree or octree. This is far from the scope of the present paper. Therefore, throughout the paper we consider that at any time of simulation the interface Γ_t is far from a shock, and the neighborhood of the interface, in which the velocity is continuously extended can therefore be defined as any area without such ridge.*

1.3. Motivation

Solving accurately the above partial differential equations (PDEs) is crucial in order to provide relevant biological interpretations of the model, however it is very challenging. One of the main difficulty lies in the fact that the velocity of the cell membrane is given as the gradient of the inner signal, which provides *a priori* a loss of accuracy[2]: usual second order schemes could lead to a first order accuracy of the gradient, and thus the level-set would be not enough accurate.

---

[2]Such a possible loss of accuracy is similar to the *a priori* loss of regularity of the continuous solution, as described in [16].

The loss of consistency could then be drastically increased by the coupling and the transmission of errors, leading to inconsistant scheme, and thus irrelevant simulations. In what follows, to overcome these difficulties we propose a numerical method based on the use of superconvergence, as defined in Definition 2, and on its transmission from one sub-problem to another.

Definition 2 (Superconvergent solution). *A numerical approximation f on a grid of a function* f *of the continuous domain is said superconvergent if, in addition to approximating* f *with a given order of accuracy, the numerical gradient $\nabla^h f$ of f approximates the gradient ∇f of* f *with the same order of accuracy.*

If in addition to the gradient, the numerical second order derivatives are also obtained with the same accuracy, then the numerical approximation f is said superconvergent on two levels (towards f *).*

In this paper, we numerically highlight that our method makes it possible to compute the solution of the Poisson problem with two levels of superconvergence, under specific conditions. Among them, the accuracy on the boundary data is crucial. Interestingly, our superconvergent method does not require exact data. This is one of the main novelty of the study: specific requirements on the data (the boundary datum (g), possible volumic source terms (f) in the static problems (1c)–(1a) –for invadopodium model f vanishes– and the level set (ψ)) are exhibited in Section 3.2 to obtain superconvergent solution to each static subproblem, and we show that these requirements are compatible with the overall dynamical coupling. Indeed, as data do not need to be exact, the transmission of the superconvergence properties from one sub-problem to the other is ensured.

The superconvergent method is based on a stabilized version of the Ghost Fluid Method, introduced by Fedkiw *et al.* [12]. Depending on the extrapolation level of the ghost values (linear, quadratic or cubic extrapolations can be considered), we obtained different results of superconvergence for the numerical solutions of the Poisson problem and of the overall problem of protrusion formation. In [16], linear extrapolations led to a specific first order method for the Neumann problem (1a)–(1b), and a second order method for the Dirichlet problem (1c)–(1d). The authors numerically showed that the resulting quantities σ, $c^\star$, ψ and even the normal to the interface are first order accurate and suggested the involvement of underlying superconvergence properties related to the methods of resolution of the Poisson equations. Actually, the consistency of the solutions partially results from the first order superconvergence of the solution of the Neumann problem, wich is transmitted to the solution of the inner problem thanks to the Dirichlet condition.

Hence, this result encouraged the authors to continue the study of the superconvergence properties and their transmission, to obtain second order numerical approximations of $c^\star$, σ and ψ. The main idea is to use higher extrapolations for the computation of the ghost values. It is worth noting that second order accurate methods to solve separately each problem (1a)–(1b), (1c)–(1d) (for a given location of the level set function ψ), and (3) (for a given velocity at the interface) do not necessarily lead to an overall second order accuracy of the coupled problem (1a)–(1b)–(1c)–(1d)–

(3). Actually, the second order accuracy of $c^\star$ requires a second order accuracy on the Neumann boundary condition (see [10] for instance), and the second order accuracy of σ requires a second order accuracy on the Dirichlet boundary condition. Thus, the interface and its normal should be second order accurate.

Another specificity of our method lies on the velocity extension (3b), which leads to the super-convergence of the level set function on one level. This surprising behavior makes it possible to obtain the second order accuracy of the level set function and the normal vector only from a second order accurate velocity, without any additional superconvergent property. This only requires *super-convergence property* on one level for σ. This result can be obtained from quadratic extrapolations of the ghost values for each static sub-problem, which give numerical schemes of Shortley-Weller type [32], for the discretization of the Poisson problems.

Then, the spontaneous following question relies upon the cubicly extrapolated ghost values. Obviously, the stencils are wider. However, in addition to σ, $c^\star$, $\boldsymbol{\psi}$ and their first derivatives, their second order derivatives will also be second order accurate in maximum norm, exhibiting a superconvergence property on two levels. This is particularly the case for the interface curvature. This is of major interest for further study about cell migration. Indeed, the protrusion retraction could be accounted for by using a specific retraction velocity $\mathbf{v_{ret}}$ and techniques of interface regularization, which depend on surface tension. For instance, we could consider a well-known interface advection problem with curvature-dependent speed, introduced by Sethian and Osher in [29], and given by

$$\partial_t \boldsymbol{\psi} + \mathbf{v_{ret}} \cdot \nabla \boldsymbol{\psi} = F(\kappa), \quad t \in [T, T'], \quad x \in \Omega,$$

where F is a given nonlinear function of the curvature κ. Nonlinearities of F require the second order accuracy on κ in order to keep the appropriate properties on ψ. This aspect is far beyond the topic of this paper and will be addressed in further studies. However, this justifies the presentation of cubic type schemes in what follows.

The main motivation of the paper is therefore to propose a specific finite difference method on Cartesian grid to compute every unknowns and first and second derivatives of the problem of protrusion formation with a second order accuracy, thanks to a stabilized Ghost Fluid Method and cubic extrapolations of the ghost values for the Poisson problems. It is understood that the reader, if uninterested in the second order accuracy of the curvature for further regularization approach, can easily replace the cubic extrapolations by quadratic extrapolations and keep second order accurate solutions of the problem, except the curvature.

Hence, the use of superconvergence properties can be adjusted to other problems. Actually, superconvergence may be a powerfull property to solve many free boundary problems, for which the interface velocity is implicitly given by one component of the solution. This may be particularly interesting to solve static 2-phase Stefan-type problems [17, 7, 31] or Hele-Shaw-like problems [19].

1.4. Superconvergence: state of the art

To our best knowledge, superconvergence on two levels and superconvergence from perturbed data have never been mentioned in literature. Several studies have addressed the one order of superconvergence, especially for finite element [40, 23, 2] and finite volume type methods [8, 3, 38]. Regarding finite difference methods on polygonal domains, Ferreira and Grigorieff have shown in [13] one level of superconvergence at the second order of accuracy for general elliptic operators while Li *et al.* in [26, 24, 25] specifically studied the Shortley-Weller scheme. In [27], Matsunaga and Yamamoto proved that the Shortley-Weller scheme provides a third order accuracy near the interface, which can be seen as an essential aspect of the superconvergence property. The Shortley-Weller approximation, introduced in [32] in 1938, has been widely studied [4, 27, 35, 26, 36] and is well-known for its superconvergence property, without complete proof until very recently. More precisely, in [37], Yoon and Min proved the superconvergence in L^2-norm at the second order of accuracy thanks to a discrete version of the divergence theorem, and Weynans, in [34], proposed a different proof for the maximum norm, involving Green's functions and analysis tools strongly inspired from [4, 9]. From a numerical point of view, Gibou and colleagues observed in [28] and [6] the superconvergence of the solution to Poisson equation with variable coefficients on non-graded adaptive Cartesian grids.

It is worth noting that all these studies consider only exact data. Moreover, they are limited to one level of superconvergence and to Dirichlet boundary conditions. In addition to these works, Jomaa *et al.* achieved in [20] the second order accuracy of the solution to the Robin problem using the Shortley-Weller method with a quadratic treatment of the boundary condition. For the Neumann condition, Kreiss and more recently Khatri [22, 21] also pointed out the second order accuracy of the solution thanks to quadratic or cubic extrapolations for the Neumann condition. However, none of these articles refer to superconvergence properties of their schemes.

Superconvergent approaches are well-suited for solving models in which one unknown derives from the solution of a Poisson problem. In these approaches, the specific superconvergent properties of the numerical solution of the Poisson problem, which can be seen as a *discrete version of the elliptic regularity properties* like in [15], prevent the loss of consistency at each derivation. However, other approaches should be considered. In particular, Bruno and Hoch propose in [5] a Chebyshev polynomial technique to postprocess the derivative computation and get a limited order-of-accuracy deterioration compared to the solution. This could be an interesting alternative to explore for solving the invadopodia model.

Finally, we should mention that in some cases, after a long time, the interface can lose in regularity or develop pathological configurations. In each situation, the method should be adjusted in order to avoid instabilities, as this is addressed in [15]. In other cases, the advection governed by (3a), could lead to instabilities in our numerical approach if the interface develop a kink for instance. Then, alternative approaches should be considered, as proposed by Friedman in [14] for Stefan problem.

1.5. Outline of the paper

As stated above, the main goal of this paper is to solve Problem (1) with the second order accuracy, using a superconvergent second order accurate Cartesian method for Poisson-type equations.

The schemes used to reach the superconvergent properties are presented in Section 2. There are based on centered schemes and the use of the ghost fluid method [12]. The keypoint lies in the appropriate extrapolations (cubic for the static subproblems, quadratic for the transport of the level set) for the ghost values to obtain the superconvergence. In particular we propose new stencils, partially introduced in [16], which stabilize the standard stencils such as these of Gibou *et al.* [18] or Cisternino and Weynans [10] for the grid points close to the interface and for the points on the interface.

Section 3 is devoted to a step-by-step study of the overall method: we first highlight the superconvergent properties of the schemes. In particular we show that the use of cubic extrapolations in the static subproblems leads to a solution with *two levels of superconvergence*. Then, in subsection 3.2.3, we show that superconvergent properties are preserved even with perturbed data. More precisely, we highlight that our method ensures the *preservation of the superconvergence properties* to the static coupled problem (1a)–(1b)–(1c)–(1d). We finally show that the transport (3), solved thanks to a RK2-LUD scheme [33], do not degrade the two levels of superconvergence provided that the extended velocity is superconvergent on one level.

Again, we emphasize that linear and quadratic extrapolations can similarly be used instead of cubic extrapolations in the discretization of the static part of the model (Poisson problems), resulting in first and second order accuracies of the respective overall methods. As said above, the quadratic approach does not result in the second order accuracy of the curvature. Note that the corresponding solutions of the so-discretized Poisson problem have specific superconvergence properties, which are summarized in the final Table 9 without details. The interested reader will refer to [15] for more informations about those alternative finite-difference superconvergent methods to solve the Poisson problem on Cartesian grid.

1.6. Illustration of the main results

The numerical method proposed in this paper makes it possible to solve the invadopodia model given by equations (1a)–(1b)–(1c)–(1d)–(3) with a second order accuracy for the unknowns σ, $c^\star$, ψ and in addition for the normal vector field $\mathbf{n}$ and the curvature κ defined from the level-set ψ by

$$\mathbf{n} = \frac{\nabla \psi}{|\nabla \psi|}, \quad \kappa = \nabla \cdot (\mathbf{n}).$$

This is summarized in the following numerical simulation, where g is given by

$$\mathrm{g} = 0.1 \left[2 + \cos\left(3\,\pi\,(x+y)\right)\cos\left(\pi\,(x+0.3)\right) \right],$$

and $\Gamma|_{t=0}$ is the circle centered in $\mathbf{O}$ of radius 0.25.Figure 2 provides the changes of the cell morphology given by the model, that is the shape of the cell membrane Γ_t, at different time points.

(a) $t = 0$ (b) $t = 0.25$ (c) $t = 0.5$
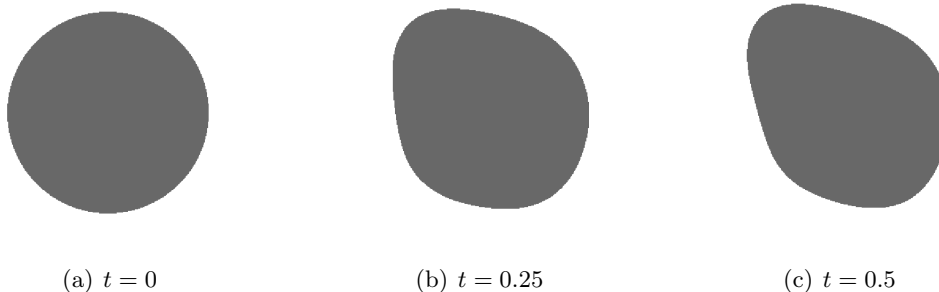
Figure 2: Numerical simulation of a change in cell morphology by solving the problem of invadopodia given by (1a)–(1b)–(1c)–(1d)–(3).

The interface velocity, before extension, is computed as the gradient of a harmonic function. As expected, Figure 3 shows a satisfying conservation in the volume.
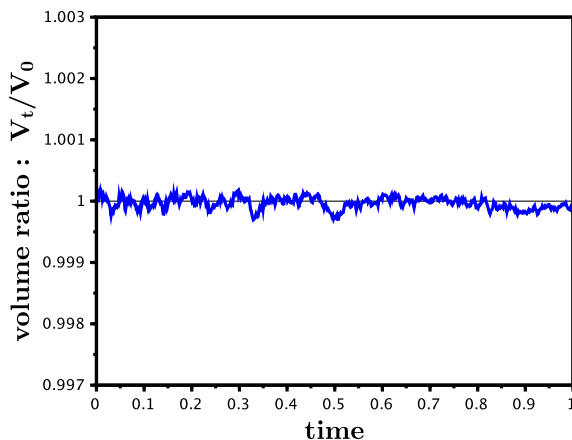
Figure 3: Volume conservation.

Table 1 illustrates the accuracy of the numerical method. The quantities σ, $c^\star$, ψ, $\mathbf{n}$, κ are second order accurate in L^∞ norm, by comparison with a numerical reference solution, computed on a fine mesh (500×500 points), at $T = 1$ (after 659 time steps).

2. Discretizations of the differential operators for the dynamical coupled problem

This section is devoted to the discretization of the differential operators. Throughout the paper, we consider a Cartesian grid, denoted by $\mathcal{G}^h$, with the same grid spacing in x and y, denoted by h. The set of the grid points is denoted by $\mathcal{P}^h$. The continuous domain $\boldsymbol{\omega}$ generically stands either for $\mathcal{O}_t^e$, $\mathcal{O}_t^i$ or Ω. Let first set the notations in order to present clearly the discretizations.

9

Number of points	Number of time steps	$c^{\star}$		σ		$\mathbf{v}$	
		L^{∞}-error	order	L^{∞}-error	order	L^{∞}-error	order
64	86	4.830×10^{-5}	-	3.733×10^{-5}	-	7.769×10^{-4}	-
96	128	1.606×10^{-5}	2.72	7.129×10^{-6}	4.08	2.297×10^{-4}	3.01
144	190	4.883×10^{-6}	2.83	3.237×10^{-6}	3.02	1.167×10^{-4}	2.34
216	285	1.793×10^{-6}	2.71	2.384×10^{-6}	2.26	6.106×10^{-5}	2.09
324	427	1.270×10^{-6}	2.24	3.032×10^{-7}	2.97	2.478×10^{-5}	2.12
		ψ		$\mathbf{n}$		κ	
64	86	3.620×10^{-4}	-	8.617×10^{-3}	-	5.089×10^{-1}	-
96	128	1.627×10^{-4}	1.97	5.049×10^{-3}	1.32	2.365×10^{-1}	1.89
144	190	8.066×10^{-5}	1.85	2.383×10^{-3}	1.59	1.381×10^{-1}	1.61
216	285	3.368×10^{-5}	1.95	9.596×10^{-4}	1.80	6.337×10^{-2}	1.71
324	427	1.069×10^{-5}	2.17	3.270×10^{-4}	2.02	2.050×10^{-2}	1.98

Table 1: Numerical results of accuracy for a simulation of protrusion formation at the final time $T = 1$.

2.1. Notations and definitions of the discrete quantities

We define the numerical level-set function ψ as a perturbation of the restriction of the exact level-set $\boldsymbol{\psi}$ to the grid points $\mathcal{P}^h$. Then, we denote by Γ_t^h, $\mathcal{O}_t^{i,h}$ and $\mathcal{O}_t^{e,h}$ the numerical interface and the inner and outer numerical domains, respectively:

$$\Gamma_t^h = \left\{ x \in \mathcal{G}^h : \psi(t,x) = 0 \right\}, \quad \mathcal{O}_t^{i,h} = \left\{ x \in \mathcal{P}^h : \psi(t,x) < 0 \right\}, \quad \mathcal{O}_t^{e,h} = \left\{ x \in \mathcal{P}^h : \psi(t,x) > 0 \right\}.$$

The discrete domain ω^h generically stands either for $\mathcal{O}_t^{e,h}$, $\mathcal{O}_t^{i,h}$ or $\mathcal{P}^h$. Note that these definitions are the discrete versions of (2).

2.1.1. Definitions of the grid, the neighboring and the interface points

Grid points of ω^h are simply denoted $x_{i,j}$. A *neighboring* point is a grid point whose one of its closest neighborhood (at least) does not belong to ω^h. Other grid points are called *inner* points.

For $\theta \in [0, 1]$, the point on the grid located at the distance $\pm h\theta$ in the x–direction (resp. y–direction) of the grid point $x_{i,j}$ is denoted by $x_{i\pm\theta,j}$ (resp. $x_{i,j\pm\theta}$). If the point $x_{i\pm\theta,j}$ (resp. $x_{i,j\pm\theta}$) belongs to the intersection between the grid and the interface Γ^h, it is named interface point and may be denoted by $x_{i\pm\theta,j}^{\Gamma}$ (resp. $x_{i,j\pm\theta}^{\Gamma}$) if necessary.

For any function u defined in the continuous domain $\boldsymbol{\omega}$, we denote by $u_{i\pm\theta_x,j\pm\theta_y}$ its value at the point $x_{i\pm\theta_x,j\pm\theta_y}$. The value of u at the interface points $x_{i\pm\theta_x,j\pm\theta_y}^{\Gamma}$ is denoted by $u_{i\pm\theta_x,j\pm\theta_y}^{\Gamma}$.

Since the discretization is performed direction by direction, only the x–direction is presented in the following, and the subscript j is sometimes omitted to simplify notations. Hence u_i stands for $u_{i,j}$, where j is fixed at the same value.

2.1.2. Quadratic and cubic extrapolation operators

As said above, the numerical schemes are based on appropriate extrapolations. More precisely, the quadratic and cubic extrapolations, respectively denoted by Q and C are defined as

$$\mathsf{Q}(\mathbf{Z};\boldsymbol{\theta}) = \sum_{i=1}^{3} a_i^{\mathsf{Q}}(\boldsymbol{\theta})\mathbf{Z}_i, \quad \forall(\mathbf{Z};\boldsymbol{\theta}) \in \mathbb{R}^3 \times [-3,3]^3, \tag{4a}$$

$$\mathsf{C}(\mathbf{Z};\boldsymbol{\theta}) = \sum_{i=1}^{4} a_i^{\mathsf{C}}(\boldsymbol{\theta})\mathbf{Z}_i, \quad \forall(\mathbf{Z};\boldsymbol{\theta}) \in \mathbb{R}^4 \times [-4,4]^4, \tag{4b}$$

where the vector fields $a^{\mathbb{L}} : [-n_{\mathbb{L}}, n_{\mathbb{L}}]^{n_{\mathbb{L}}} \to \mathbb{R}^{n_{\mathbb{L}}}$ are given by

$$a_i^{\mathbb{L}}(\boldsymbol{\theta}) = \frac{\displaystyle\prod_{\substack{k=1,\cdots,n_{\mathbb{L}} \\ k \neq i}} \theta_k}{\displaystyle\prod_{\substack{k=1,\cdots,n_{\mathbb{L}} \\ k \neq i}} (\theta_k - \theta_i)}, \quad \text{for } \mathbb{L} = \mathsf{Q}, \mathsf{C}, \tag{4c}$$

the integer $n_{\mathbb{L}}$ being equal to 3 (resp. 4) for $\mathbb{L} = \mathsf{Q}$ (resp. $\mathbb{L} = \mathsf{C}$).

Remark 3 (About quadratic and cubic extrapolations). *One can wonder why cubic and quadratic extrapolations have to be introduced. First of all, one can notice that in order to get a truncation error on the Laplace operator with a second order of accuracy, it is necessary to compute the ghost values with cubic extrapolation. Such cubic extrapolation adds an error of order h^3 to the gradient of the solution, making it possible, if the numerical scheme is superconvergent on two levels, to derive numerically two times the numerical solution, keeping the second order of accuracy of the first and second order derivatives. Quadratic extrapolation is then sufficient for the second numerical derivation to keep this second order of accuracy. Moreover, as noted below, quadratic extrapolation is necessary to prevent numerical instabilities (which appear with cubic extrapolations) for the velocity extension and the transport of the level set.*

2.1.3. Localization of the level set

The distance $h\theta_x$ of the interface point $x_{i-\theta_x}^{\Gamma}$ to the neighboring point $x_{i,j}$, if not exactly known, has to be computed. In order to add a second order error at worst on the second order of derivatives of the level-set, we use a fourth order approximation. For such purpose, the level set function ψ is approximated with a third order interpolation polynomial P in the Lagrange form. Denoting by α the normalized distance between x_i and any point $x_{i-\alpha}$ of the x-axis, the polynomial P can be written as a function of α:

$$P(\alpha) = \frac{1}{6}\left(\psi_{i-2} - 3\psi_{i-1} + 3\psi_i - \psi_{i+1}\right)\alpha^3 + \frac{1}{2}\left(\psi_{i-1} - 2\psi_i + \psi_{i+1}\right)\alpha^2 \tag{5}$$
$$+ \frac{1}{6}\left(-\psi_{i-2} + 6\psi_{i-1} - 3\psi_i - 2\psi_{i+1}\right)\alpha + \psi_i.$$

Solving $P(\alpha) = 0$ thanks to Newton's method provides the desired approximation of θ_x.

2.2. Discrete operators to solve the static subproblems (1a)–(1b) and (1c)–(1d)

2.2.1. Discrete Laplacian

The second order derivative $\partial_{xx}^h u_i$ of a function u at the grid point x_i is defined at the point thanks to the standard five-point stencil:

$$\partial_{xx}^h u_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}. \tag{6}$$

At any inner grid point, all the neighbors are still in ω^h and thus the above formula is well defined. However if x_i is a neighboring point, then for instance $x_{i-1} \notin \omega^h$ and thus u_{i-1} is not defined: it is called a ghost value –denoted by u_{i-1}^G for clarity–and it has to be extrapolated from the values at the interface and at the inner points (see Figure 4).

(a) Standard cubic Ghost fluid approach [12] (b) Stabilization of the method
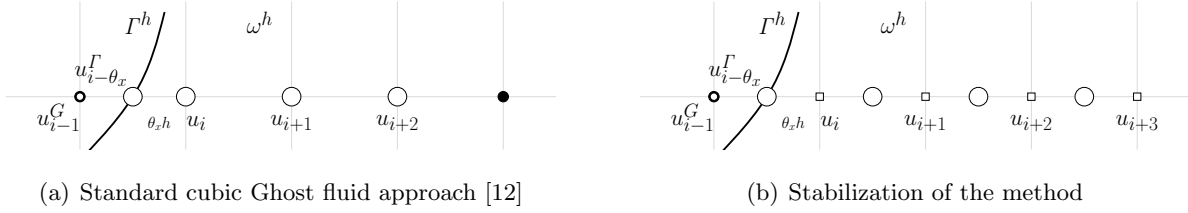
Figure 4: Standard and the stabilized methods for the ghost value computation.

The standard cubic Ghost fluid approach [12] consists in using $u_{i-\theta}^\Gamma$ and the inner points (u_{i+k}) for $k = 0, 1, 2$ (see Figure 4(a))

$$u_{i-1}^G = \mathsf{C}(u_{i-\theta_x}^\Gamma, u_i, u_{i+1}, u_{i+2}; \boldsymbol{\theta}), \quad \text{with} \quad \boldsymbol{\theta} = [1 - \theta_x, 1, 2, 3]^T.$$

The drawback is that the coefficients of the extrapolation involve terms[3] in $1/\theta_x$, which blows up for $\theta_x \to 0$, leading to instabilities as mentioned in [18]. This may affect the second order of accuracy of the solution and also prevent from any superconvergent properties. We thus propose the following extrapolation of u_{i-1}^G, in order to stabilize the discrete Laplacian operator:

$$u_{i-1}^G = \mathsf{C}(u_{i-\theta_x}^\Gamma, u_{i+1-\theta_x}, u_{i+2-\theta_x}, u_{i+3-\theta_x}; (\boldsymbol{\theta}_k)_{k=1,\cdots,4}), \quad \text{with} \quad \boldsymbol{\theta}_k = (k - \theta_x). \tag{7}$$

The new second order derivative at the neighboring point x_i reads

$$
\begin{aligned}
\partial_{xx}^{\mathsf{C}} u_i = \frac{1}{h^2} \Bigg[& u_{i+1} - 2\,u_i + \frac{(2-\theta_x)(3-\theta_x)(4-\theta_x)}{6} u_{i-\theta_x}^\Gamma - \frac{(1-\theta_x)(3-\theta_x)(4-\theta_x)}{2} u_{i+1-\theta_x} \\
& + \frac{(1-\theta_x)(2-\theta_x)(4-\theta_x)}{2} u_{i+2-\theta_x} - \frac{(1-\theta_x)(2-\theta_x)(3-\theta_x)}{6} u_{i+3-\theta_x} \Bigg],
\end{aligned}
\tag{8}
$$

---

[3]Simple calculations left to the reader lead to the formula for the standard Ghost fluid method with cubic extrapolation:

$$\partial_{xx}^{\mathsf{C}} u_i = \frac{6}{\theta_x(1+\theta_x)(2+\theta_x)\,h^2} u_{i-\theta_x}^\Gamma - \frac{3-\theta_x}{\theta_x h^2} u_i + \frac{2\,(2-\theta_x)}{(1+\theta_x)\,h^2} u_{i+1} - \frac{1-\theta_x}{(2+\theta_x)\,h^2} u_{i+2}.$$

12

Then, as the values $u_{i+k-\theta_x}$ are not at the grid points, they are extrapolated at the order 4 from the values at the grid points as follows:

$$u_{i+k-\theta_x,j} = \mathsf{C}\left(u_{i,j}, u_{i+1,j}, u_{i+2,j}, u_{i+3,j}; (\theta_x - k + \ell)_{\ell=0,\cdots,3}\right), \quad k \in \{1, \cdots, 3\}, \tag{9}$$

and the discretization is finally of the form:

$$\partial_{xx}^{\mathsf{C}} u_i = \frac{1}{h^2}\left[\alpha(\theta_x)\, u_i + \beta(\theta_x)\, u_{i+1} + \gamma(\theta_x)\, u_{i+2} + \delta(\theta_x)\, u_{i+3} + \eta(\theta_x)\, u_{i-\theta_x}^\Gamma\right],$$

Note that even though one more point is necessary compared to the standard method, the new discretization is thus stabilized, since the coefficients do not blow up for $\theta_x \to 0$.

Remark 4 (Invertibility of the resulting discretization matrix in the 1D Dirichlet case). *In contrast to the non-stabilized method, the discretization matrix is not diagonally dominant for a range of values of θ_x. Nevertheless, it is invertible. The main arguments are as follows. Consider the matrices*

$$A = (a_{ij})_{1 \le i,j \le n}, \quad a_{ij} = \begin{cases} -2 & \text{if } j = i, \\ 1 & \text{if } j = i \pm 1, \\ 0 & \text{otherwise,} \end{cases}$$

and $B = (b_{ij})_{1 \le i,j \le n}$, with $[b_{11}, b_{12}, b_{13}, b_{14}] = [\alpha(\theta_x) + 2, \beta(\theta_x) - 1, \gamma(\theta_x), \delta(\theta_x)]$, and the other coefficients b_{ij} are 0.

The discretization matrix is therefore of the form $M = A + B = A(I_n + A^{-1}B)$. The matrix A is monotone and it can be easily checked by induction that its inverse matrix is given by

$$A^{-1} = (\bar{a}_{ij})_{1 \le i,j \le n}, \quad \bar{a}_{ij} = \begin{cases} -\dfrac{i(n-j+1)}{n+1} & \text{if } j \ge i, \\ -\dfrac{j(n-i+1)}{n+1} & \text{otherwise.} \end{cases}$$

The matrix $I_n + A^{-1}B$ is then a block matrix of the form $\left(\begin{array}{c|c} U & O \\ \hline V & I_n \end{array}\right)$. It follows that

$$det(I_n + A^{-1}B) = det(U) = \frac{1}{6}\left[\theta_x^4 + (n-9)\,\theta_x^3 + (-9n+26)\,\theta_x^2 + (26n-24)\,\theta_x - 24n\right].$$

Deriving with respect to n, it is direct that $det(U)$ is nonzero for any $n \ge 1$ and $\theta_x \in [0,1]$.

We are confident on the fact that further study may lead to the same conclusion for 2D and Neumann problems.

2.2.2. Normal vector to the isolines

The normal vector $\mathbf{n}$ is given thanks to the level-set $\boldsymbol{\psi}$ as

$$\mathbf{n} = \frac{\nabla\boldsymbol{\psi}}{|\nabla\boldsymbol{\psi}|},$$

it is thus important to define the first order derivative at the grid points. They are computed thanks to the standard centered scheme:

$$\partial_x^h u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \quad \partial_y^h u_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2h}, \quad \nabla^h = \begin{pmatrix} \partial_x^h \\ \partial_y^h \end{pmatrix}.$$

Since the level-set ψ is defined in the whole domain Ω^h, the above formula makes it possible to compute the normal vector to the isoline at the grid point $x_{i,j}$:

$$\boldsymbol{n}_{i,j} = \frac{\nabla^h \psi_{i,j}}{|\nabla^h \psi_{i,j}|}. \tag{10}$$

Normal at the interface points. The normal vector at the interface point is of great interest, especially to solve (1a)–(1b). Since ψ is defined in the whole discrete domain, the normal vector at the interface is extrapolated from the neighboring values. For instance at the interface point $x_{i,j+\theta_y}^\Gamma$, the normal vector $\boldsymbol{n}_{i,j+\theta_y}$ is extrapolated[4] by the values at the grid points $(x_{i,j+k})_{k=-1,\cdots,2}$:

$$\boldsymbol{n}_{i,j+\theta_y} = \mathsf{C}(\boldsymbol{n}_{i,j+2}, \boldsymbol{n}_{i,j+1}, \boldsymbol{n}_{i,j}, \boldsymbol{n}_{i,j-1}; [\theta_y - 2, \theta_y - 1, \theta_y, \theta_y + 1]), \tag{11}$$

where $\boldsymbol{n}_{i,j+k}$ are given by (10).

2.2.3. Direct and indirect first order derivatives at the interface

It is important to define appropriately the direct and indirect first order interface derivatives –denoted respectively by ∂_x^Γ and $\partial_x^{\Gamma,I}$– especially for Neumann problem. We introduce second order accurate interface derivatives, which are extensions of the first order definitions presented in [16], and which stabilize the stencils of [10, 20]. Note that this stable approach ensures the *stencil continuity.*

In the example of Figure 5(a), the point $x_{i,j+\theta_y}^\Gamma$ is on the y-axis and the *indirect* x-derivative of a function u at this point computed with the stable ghost fluid method and cubic extrapolation of the ghost value:
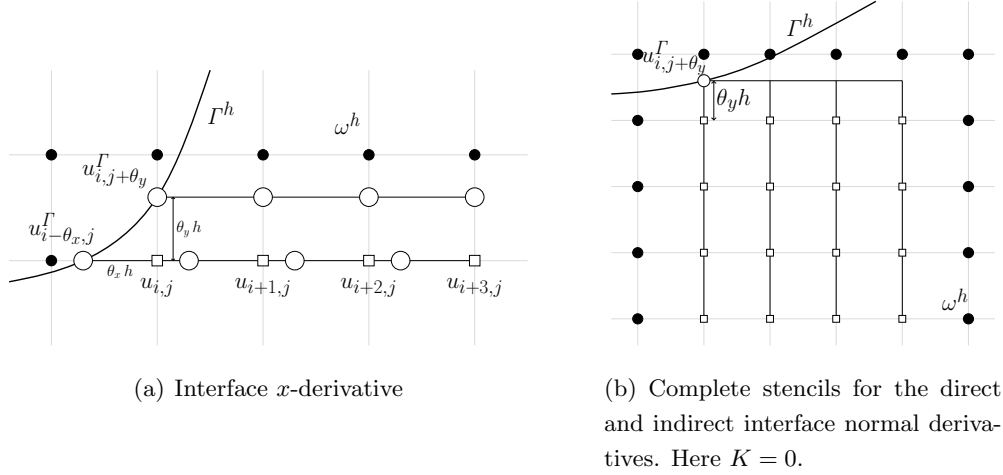
$$\partial_x^{\Gamma,I} u_{i,j+\theta_y} = \frac{u_{i+1,j+\theta_y} - u_{i-1,j+\theta_y}^G}{2h}, \tag{12}$$

where

$$u_{i-1,j+\theta_y}^G = \mathsf{C}\left((u_{i+k,j+\theta_y})_{k=0,\cdots,3}; [1, 2, 3, 4]\right).$$

---

[4]The above extrapolation has to be understood component by component, by abuse of notation.

(a) Interface x-derivative

(b) Complete stencils for the direct and indirect interface normal derivatives. Here $K = 0$.

Figure 5: Examples of the stencils for the interface normal derivative.

Then the indirect interface derivative reads (see Figure 5):

$$\partial_x^{\Gamma,I} u_{i,j+\theta_y}^{\Gamma} = \frac{-4\,u_{i,j+\theta_y}^{\Gamma} + 7\,u_{i+1,j+\theta_y} - 4\,u_{i+2,j+\theta_y} + u_{i+3,j+\theta_y}}{2\,h}, \tag{13}$$

The above values $u_{i+k,j+\theta_y}$, $k = 0, \cdots, 3$, have also to be extrapolated. Depending on the orientation of the normal vector at the point $x_{i,j+\theta_y}$, we choose $K \in \{0, 1, 2, 3, 4\}$ such that all the points $(x_{i+k,j+K-\ell})_{(k,\ell)=0,\cdots,3}$ are grid points[5] of ω^h. Then the values $u_{i+k,j+\theta_y}$ for $k = 0, \cdots, 3$ are approached in the y–direction as

$$u_{i+k,j+\theta_y} = \mathsf{C}\left((u_{i+k,j+K-\ell})_{\ell=0,\cdots,3}; (\theta_y - K + \ell))_{\ell=0,\cdots,3}\right).$$

The direct interface derivative is given by

$$\partial_x^{\Gamma} u_{i-\theta_x,j} = \frac{u_{i+1-\theta_x,j} - u_{i-1-\theta_x,j}^{G}}{2h}, \tag{14}$$

where

$$u_{i-1-\theta_x,j}^{G} = \mathsf{C}\left((u_{i+k-\theta_x,j})_{k=0,\cdots,3}; [1, 2, 3, 4]\right).$$

Using the definition of the cubic extrapolation operator (4b), the direct derivative reads

$$\partial_x^{\Gamma} u_{i-\theta_x,j}^{\Gamma} = \frac{-4\,u_{i-\theta_x,j}^{\Gamma} + 7\,u_{i+1-\theta_x,j} - 4\,u_{i+2-\theta_x,j} + u_{i+3-\theta_x,j}}{2\,h}. \tag{15}$$

Here again the values $u_{i+k-\theta_x,j}$, $k = 1, \cdots, 4$, are extrapolated from the neighboring grid point values, thanks to (9).

---

[5]In some very specific cases, it is impossible to have 4 aligned grid points of ω: this means that the grid has to be refined and less accurate stencils have to be used. We do not consider such cases.

Remark 5 (Continuity of the stencils). *The interesting novelty of stencils (13)–(15) lies in their continuity, in the sense that if the points $x^\Gamma_{i,j+\theta_y}$ and $x^\Gamma_{i-\theta_x,j}$ converge towards each other, the stencils of $\partial^{\Gamma,I}_x u^\Gamma_{i,j+\theta_y}$ and $\partial^\Gamma_x u^\Gamma_{i-\theta_x,j}$ are identical. Note that our method seems to have similarities with methods presented in previous works on wave equations [22] or on the evolution of surfactant concentration on a moving interface [21]. In particular, in these previous works the stencils satisfy such a continuity, but the authors do not mention any superconvergence property.*

Remark 6 (Treatment of the Neumann condition at the interface point). *If Neumann condition is imposed on the interface the normal derivative is given thanks to the vector at the interface (11) and the above direct and indirect derivatives (13)–(15). For instance, at the interface point $x^\Gamma_{i,j+\theta_y}$, the normal derivative reads:*

$$\partial^\Gamma_{\mathbf{n}} u_{i,j+\theta_y} = \begin{pmatrix} \partial^{\Gamma,I}_x u_{i,j+\theta_y} \\ \partial^\Gamma_y u_{i,j+\theta_y} \end{pmatrix} \cdot \boldsymbol{n}_{i,j+\theta_y}. \tag{16}$$

To simplify notation, let transiently denote by n^Γ_x and n^Γ_y the x and y components of $\boldsymbol{n}_{i,j+\theta_y}$. The keypoint to treat the Neumann boundary condition (1b) is that $\partial^\Gamma_{\mathbf{n}} u_{i,j+\theta_y}$ has the following form:

$$\partial^\Gamma_{\mathbf{n}} u_{i,j+\theta_y} = \frac{1}{2h} \left[\left(\alpha\, u^\Gamma_{i,j+\theta_y} + \sum_{i,j} a_{i,j} u_{i,j} \right) n^\Gamma_x + \left(\beta\, u^\Gamma_{i,j+\theta_y} + \sum_{i,j} b_{i,j} u_{i,j} \right) n^\Gamma_y \right] := -g^\Gamma_{i,j+\theta_y},$$

where α, β, $(a_{i,j})$ and $(b_{i,j})$ are the known extrapolation parameters, which depend on θ_x or θ_y. Then, the interface value of the solution is given by

$$u^\Gamma_{i,j+\theta_y} = -\frac{2h\, g^\Gamma_{i,j+\theta_y} + n^\Gamma_x \sum_{i,j} a_{i,j} u_{i,j} + n^\Gamma_y \sum_{i,j} b_{i,j} u_{i,j}}{\alpha\, n^\Gamma_x + \beta\, n^\Gamma_y}. \tag{17}$$

Using the interface derivative discretizations as in [10] is sufficient to achieve the numerical gradient superconvergence. However, our approach stabilizes the method. Interestingly, it avoids the addition of interface unknowns in the discretization matrix as for Cisternino-Weynans' method.

Most of all, the continuous approach is absolutely required to achieve two levels of superconvergence at the order 2 for the solution u. Indeed, without stencil continuity, jumps of the values of the numerical second derivative appear, preventing the superconvergence on the second level in maximum norm.

Note that Robin condition can be handled by this method by adding a term to the denominator in relation (17).

In this subsection, we have presented all the operators needed to solve the static subproblems (1a)–(1b) and (1c)–(1d), provided that the level-set function ψ is known at the grid points of the whole domain. In order to solve the coupled invadopodia model, it thus remains to define the extension and the transport operators.

16

2.3. Operators for the motion of the level set

2.3.1. Extended velocity, transport and volume conservation

In the Eulerian description, the transport of the level set is given by equation (3a). However, as the velocity is biologically given as the gradient of σ in the inner area $\mathcal{O}_t^i$, it does not exist in the outer area. In order to avoid the appearance of too high or discontinuous gradients near the interface, a continuous extension $\mathbf{v}$ of this velocity is required from Γ^h to its neighborhood. Similarly to in [16], the choice of the extension operator is given by (3b), meaning that the velocity is extended constantly along the normal to the isolines of ψ. Equation (3b) for velocity extension was originally introduced in [39, 1], and were solved thanks to the fast marching method.

Authors already noticed that the signed distance function was preserved if advected with this kind of extended velocity. This result follows from the derivation of Equation (3a):

$$\partial_t \nabla \psi + \nabla (\mathbf{v} \cdot \nabla \psi) = \partial_t \nabla \psi + (\mathbf{v} \cdot \nabla) \nabla \psi + (\nabla \psi \cdot \nabla) \mathbf{v} + \mathbf{v} \times (\nabla \times \nabla \psi) + \nabla \psi \times (\nabla \times \mathbf{v}) = 0$$

In the right hand side, the third and fourth terms are zero by construction of the extended velocity and by curl operator property, respectively. The fifth term is colinear to the tangential vector. Multiplying by $\nabla \psi$, we therefore obtain

$$\partial_t \left(|\nabla \psi|^2 \right) + \mathbf{v} \cdot \nabla \left(|\nabla \psi|^2 \right) = 0$$

Then, if the level set function is initialized as the signed distance function, we have $|\nabla \psi| = 1$ at the initial time, and the property is preserved at each time. Thus, the expensive procedure of redistanciation can be avoided, since the level set function is automatically reinitialized during advection.

The operators are discretized with a second order accurate truncature error, as described below. Assuming that the velocity is second order accurate, it directly follows that at each time

$$|\nabla^h \psi| = 1 + \mathcal{O}(h^2).$$

Surprisingly, this conservation of the signed distance function with a second order accuracy seems to result in a superconvergent behavior of the level set function. More precisely, if advected with any second order extended velocity defined by (3b), the level set and its normal vector are both second order accurate. If in addition σ has two levels of superconvergence, the extension preserves the superconvergence properties and the velocity therefore has one level of superconvergence. The level set funcion, its normal vector and its curvature are then second order accurate. These results are reported in Table 8 of Section 3.

Finally, it is worth noting that the extended velocity is not divergence-free. However, the volume is still well-conserved since only the interface velocity, which is extrapolated from the divergence-free velocity $\nabla \sigma$, is implied in volume conservation. A quick proof reads

$$\partial_t \int_{\mathcal{O}_t^i} d\mathbf{x} = \int_{\Gamma_t} \mathbf{v} \cdot \mathbf{n} \, ds = \int_{\Gamma_t} \partial_\mathbf{n} \sigma \, ds = \int_{\mathcal{O}_t^i} \nabla \cdot \nabla \sigma d\mathbf{x} = 0.$$

17

From the numerical point of view, $\nabla \cdot \nabla \sigma$ is approximated with a second order accuracy in maximum norm, which results in a second order accurate volume conservation and explains the good result of Figure 3.

2.3.2. Discretization

The advection equation (3a) is approximated thanks to the standard second order Runge-Kutta method (RK2) for the time derivative and with the linear upwind differencing scheme (LUD), introduced by Warming and Beam [33], for the gradient of ψ. Since RK2 scheme in time does not involve any difficulty, we do not present it.

Even if the LUD scheme is well-known, details of a modified LUD scheme is presented below, for the discretization of velocity extension (3b). The modification is introduced for the points close to the interface, in order to take the interface value into account.

First, it is worth noting that the standard LUD discretization can be built from the usual centered scheme for the first derivatives and quadratic extrapolations of the downwind values[6]. In the above subsection 2.2 cubic extrapolations have been used. The choice of cubic extrapolations was justified by Remark 3 in order to have a chance to get two levels of superconvergence of the solutions to the static problems, as it will be investigated in the next section. However, it is known that for propagation equations, such cubic extrapolations of the downwind values generate numerical instabilities, and thus quadratic extrapolations have to be preferred. Let us mention that such quadratic extrapolations will not impact the superconvergence properties, since they are used for the velocity, which is given as the gradient of the solution to (1c)–(1d), and thus it is sufficient to focus on one level of superconvergence on the velocity.

Before giving the numerical schemes of upwind derivatives, let us first define the downwind value.

Definition 7 (Downwind value). *Let $x_{i,j}$ and $x_{i\pm1,j}$ be close grid points . The point $x_{i\pm,j}$ is said to be downwind in the x–direction compared to $x_{i,j}$, if*

$$\pm n_{xi,j} < 0.$$

Downwind points in the y–direction are similarly defined.

2.3.3. Upwind derivative based on LUD scheme

The upwind discrete derivative operators ∂_x^{UP} and ∂_y^{UP}, applied to each of the velocity components, has to be upwind, with respect to each of the normal component (with the right sign). Here again we focus on the x-direction. At any point grid point x_i

$$\partial_x^{\mathrm{UP}} v_i = \frac{v_{i+1} - v_{i-1}}{2h}.$$

---

[6]Downwind values that appear in transport and velocity extension correspond to the ghost value of the static subproblems.

Far from the interface (at the inner grid points), the downwind value is quadratically extrapolated with the 3 closest upwind value. For instance if x_{i+1} is downwind (see Figure 6(a)), then

$$v_{i+1} = \mathsf{Q}(v_i, v_{i-1}, v_{i-2}; [1, 2, 3]),$$

and thus $\partial_x^{\mathrm{UP}} v_i$ reads

$$\partial_x^{\mathrm{UP}} v_i = \frac{3\, v_i - 4\, v_{i-1} + v_{i-2}}{2\, h}. \tag{18}$$

(a) Far from the interface   (b) Close to the interface
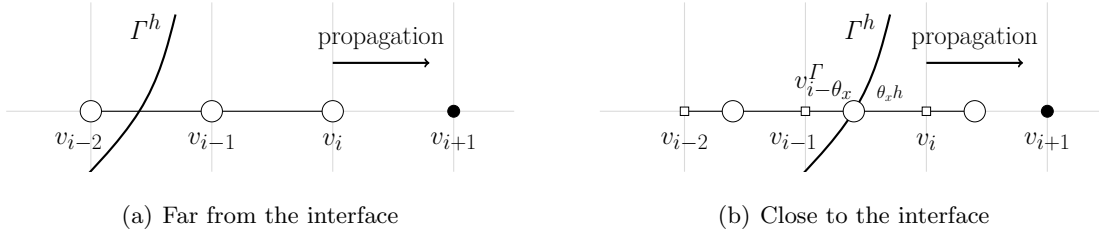
Figure 6: Second order upwind schemes (based on LUD scheme) for velocity extension.

At the point neighboring to the interface (see Figure 6(b)), it is necessary to include the interface value of the velocity, to account for the condition (3b). In order to keep a continuity of the stencil (see Remark 8), as in the previous subsection v_{i+1} is extrapolated as

$$v_{i+1,j} = \mathsf{Q}((v_{i+k-\theta_x})_{k=-1,0,1}; (\theta_x - k + 1)_{k=-1,0,1}]),$$

and the derivative reads

$$\partial_x^{\mathrm{UP}} v_i = \frac{\left[(1 + \theta_x)(2 + \theta_x)\, v_{i+\theta_x} - 2\theta_x\,(2 + \theta_x)\, v_{i-\theta_x}^{\Gamma} + \theta_x\,(1 + \theta_x)\, v_{i-1-\theta_x} - 2v_{i-1}\right]}{4h}. \tag{19}$$

Then the intermediate points $v_{i\pm1-\theta_x}$ are quadratically interpolated:

$$v_{i\pm1-\theta_x} = \mathsf{Q}((v_{i+k})_{k=-2,-1,0}; (\pm(1-k) - \theta_x)_{k=-2,-1,0}]). \tag{20}$$

Remark 8 (Stencil continuity). *Note that this approach implies that in both cases $\theta_x \to 0$ and $\theta_x \to 1$, the scheme tends to be the same as (18). In particular, in the case $\theta_x \sim 1$, it gives the continuity between the stencil close to the interface ($\theta_x < 1$) and the stencil far from the interface ($\theta_x > 1$), which seems to be essential to ensure the preservation of the superconvergence property on the extended velocity. On the other side of the interface, the discretizations are similar, in the opposite direction since the normal component has the opposite sign. The results in subsection 3.3 confirm that the method of extension is second order accurate and preserves the velocity superconvergence.*

19

2.4. Algorithm for the numerical computation

We are now ready to provide the algorithm to solve the coupled problem (1a)–(1b)–(1c)–(1d)–(3).

Note that if linear extrapolations of the ghost values give a symmetric matrix for the Dirichlet problem [18], it is not the case for the Neumann problem in general, or for the Dirichlet problem with our stencils. The non-symmetry of the matrix is somehow the cost to get superconvergence properties. In what follows, the non-symmetric matrices are inverted thanks to the BiCGStab method. For each simulation, the computational domain is $\mathcal{D} = [-0.5, 0.5] \times [-0.5, 0.5]$.

The computation holds as follows: starting from ψ^{t_n}, the solution to the static subproblems $c^{\star t_n}$ and σ^{t_n} are computed thanks to the operators defined in subsection (2.2). Then the velocity at the interface, derived from $\nabla^\Gamma \sigma^{t_n}|_{\Gamma^{h,t_n}}$ is extrapolated constantly along the normals and the membrane is transported by solving (3) on one time step.

The above algorithm makes it possible to solve accurately the dynamical coupled problem (See Figure 2). In particular, the numerical convergence is shown to be at the order 2 for $c^\star$, σ, ψ and even $\boldsymbol{n}$, as provided by table 1, showing the two levels of superconvergence of the unknowns: σ, $c^\star$, and even ψ. In the next section superconvergence properties are investigated in more details.

3. Numerical study of the superconvergence properties

The aim of this section is to verify that the above schemes provide effectively superconvergent solutions. For such a purpose, it is necessary to compare first and second order derivatives of σ and $c^\star$ as well as first order derivatives of $\mathbf{v}$ in the appropriate domains. The first order derivatives of the velocity have been already defined in the whole domain in subsection 2.3.3, however the first order derivatives of $c^\star$ or σ have been defined only at the interface. We thus introduce the first order derivatives at the grid points in subsection 3.1. Then we study the superconvergence properties of the static problems in subsection3.2. We first study the superconvergence of each problem, solved separately, and we then exhibit the preservation of the superconvergence for the static coupled problem (1a)–(1b)–(1c)–(1d). We conclude this subsection by proposing the minimal requirements on the approximation of the data to obtain superconvergence on two levels and transmission of this property. Subsection 3.3 is devoted to the preservation of the superconvergent properties by the dynamics (3).

3.1. Numerical derivatives at the grid points of the inner or outer domains

If x_i is an inner points of ω^h, the standard centered stencil is used for the first order derivative $\partial_x^h u_i$ of a function u.

If x_i is a neighboring point, then for instance $x_{i-1} \notin \omega$ and thus u_{i-1} is a ghost value. Depending whether u is the solution –namely $c^\star$, or σ– or a first order derivative of the solution, the ghost

value is approached by cubic or quadratic extrapolations respectively. We thus define for $\mathbb{L} = \mathsf{Q}, \mathsf{C}$ the ghost value $u_{i-1}^{\mathbb{L}}$ as[7]

$$u_{i-1}^{\mathbb{L}} = \mathbb{L}((u_{i+k-1-\theta_x})_{k=1,\cdots,n_{\mathbb{L}}}; \boldsymbol{\theta}_{\mathbb{L}}), \quad \text{with} \quad \boldsymbol{\theta}_{\mathbb{L}} = (k - \theta_x)_{k=1,\cdots,n_{\mathbb{L}}}. \tag{21}$$

The values $u_{i+k-\theta_x}$ are not at the grid points, and thus they are extrapolated at the order 2 or 3 (as noticed in Remark 3) from the values at the grid points:

$$u_{i+k-\theta_x} = \mathbb{L}((u_{i+\ell})_{\ell=0,\cdots,n_{\mathbb{L}}-1}; \boldsymbol{\theta}_{\mathbb{L}}), \quad \text{with} \quad \boldsymbol{\theta}_{\mathbb{L}} = (k - \ell - \theta_x))_{\ell=0,\cdots,n_{\mathbb{L}}-1}, \quad \text{for } k = 0, ..., n_{\mathbb{L}} - 1,$$

where $\mathbb{L}$ stands for Q, or C. We then define the derivative operator $\partial_x^{\mathbb{L}}$ at the neighboring point x_i as

$$\partial_x^{\mathbb{L}} u_i = \frac{u_{i+1} - u_{i-1}^{\mathbb{L}}}{2h}, \quad \mathbb{L} = \mathsf{Q}, \text{ or } \mathsf{C}.$$

We abusively denote by $\nabla_x^{\mathbb{L}}$ the first order derivative operators, corresponding to ∂_x^h on the inner grid points and $\partial_x^{\mathbb{L}}$ on the grid points neighboring the interface. In addition $\nabla^{\mathbb{L}}$ stands for

$$\nabla^{\mathbb{L}} = \begin{pmatrix} \partial_x^{\mathbb{L}} \\ \partial_y^{\mathbb{L}} \end{pmatrix}, \quad \mathbb{L} = \mathsf{Q}, \text{ or } \mathsf{C}.$$

3.2. Superconvergence properties for the static problems

In this subsection, we one of the static subproblems (1a)-(1b) or (1c)-(1d). The set $\boldsymbol{\omega}$ generically denotes either the outer area $\mathcal{O}_t^e$ or the inner area $\mathcal{O}_t^i$ of problem (1) at a given time t. Γ denotes the location of Γ_t at the time t. We consider the generic problem

$$\Delta u = f, \quad \text{in } \boldsymbol{\omega}, \quad u|_{\partial\Omega \cap \partial\boldsymbol{\omega}} = 0, \tag{22a}$$

$$u|_{\Gamma \cap \partial\boldsymbol{\omega}} = g, \text{ (Dirichlet)}, \quad \text{or} \quad \partial_{\mathbf{n}} u|_{\Gamma \cap \partial\boldsymbol{\omega}} = g, \text{ (Neumann)}. \tag{22b}$$

3.2.1. Superconvergence for the Dirichlet problem (numerical test 1)

Consider the Dirichlet problem (22) whose exact solution given by

$$u(x, y) = x^2 \cos(\pi(4x + y)), \quad \text{in } \boldsymbol{\omega},$$

where $\boldsymbol{\omega} = \{(x, y) : \boldsymbol{\psi}(x, y) < 0\}$, where $\boldsymbol{\psi}$ is given in polar coordinates:

$$\boldsymbol{\psi}(r \cos\theta, r \sin\theta) = r + 0.4 \cos^2(\theta + \frac{\pi}{4}) \sin^2(\theta + \frac{\pi}{4}) - 0.3, \quad \forall (r, \theta) \in \mathbb{R}^+ \times \mathbb{R}/2\pi\mathbb{Z}. \tag{23}$$

The functions g and f are explicitly inferred from u. The tests of numerical accuracy are given in Table 2, in discrete L^1-norm and maximum norm.

Note that Table 3 shows that the second-order partial derivatives and the second-order mixed derivatives (in the sense of two successive derivations) are also superconvergent.

---

[7]Remind that $n_{\mathsf{Q}} = 3$ and $n_{\mathsf{C}} = 4$.

Number of points	$\nabla^{\mathsf{C}} u$				$\nabla^{\mathsf{Q}} \cdot \nabla^{\mathsf{C}} u$			
	L^1-error	order	L^∞-error	order	L^1-error	order	L^∞-error	order
96	1.980×10^{-4}	-	3.034×10^{-3}	-	9.181×10^{-3}	-	1.096×10^{-1}	-
144	8.667×10^{-5}	2.04	1.349×10^{-3}	2.00	4.069×10^{-3}	2.01	5.618×10^{-2}	1.65
216	3.842×10^{-5}	2.02	5.987×10^{-4}	2.00	1.805×10^{-3}	2.01	2.437×10^{-2}	1.85
324	1.720×10^{-5}	2.01	2.670×10^{-4}	2.00	8.088×10^{-4}	2.00	1.085×10^{-2}	1.90
486	7.656×10^{-6}	2.01	1.190×10^{-4}	2.00	3.601×10^{-4}	2.00	4.957×10^{-3}	1.91

Table 2: Superconvergence of the solution for the Dirichlet problem (numerical test 1).

Number of points	$\partial_x^{\mathsf{Q}} \partial_x^{\mathsf{C}} u$				$\partial_y^{\mathsf{Q}} \partial_y^{\mathsf{C}} u$			
	L^1-error	order	L^∞-error	order	L^1-error	order	L^∞-error	order
96	1.005×10^{-2}	-	1.984×10^{-1}	-	1.054×10^{-3}	-	1.277×10^{-1}	-
144	4.415×10^{-3}	2.03	9.466×10^{-2}	1.82	4.278×10^{-4}	2.22	6.670×10^{-2}	1.60
216	1.950×10^{-3}	2.02	3.647×10^{-2}	2.09	1.779×10^{-4}	2.19	2.725×10^{-2}	1.90
324	8.721×10^{-4}	2.01	1.843×10^{-2}	1.95	7.812×10^{-5}	2.14	1.317×10^{-2}	1.87
486	3.880×10^{-4}	2.01	8.546×10^{-3}	1.94	3.449×10^{-5}	2.11	6.207×10^{-3}	1.86
	$\partial_x^{\mathsf{Q}} \partial_y^{\mathsf{C}} u$				$\partial_y^{\mathsf{Q}} \partial_x^{\mathsf{C}} u$			
96	2.700×10^{-3}	-	3.417×10^{-1}	-	1.874×10^{-3}	-	1.794×10^{-1}	-
144	9.307×10^{-4}	2.63	1.655×10^{-1}	1.79	6.805×10^{-4}	2.50	8.647×10^{-2}	1.80
216	3.419×10^{-4}	2.55	7.624×10^{-2}	1.85	2.610×10^{-4}	2.43	3.609×10^{-2}	1.98
324	1.293×10^{-4}	2.50	3.299×10^{-2}	1.92	1.089×10^{-4}	2.34	1.787×10^{-2}	1.90
486	5.116×10^{-5}	2.45	1.447×10^{-2}	1.95	4.572×10^{-5}	2.29	7.916×10^{-3}	1.92

Table 3: Accuracy of the second order partial and mixed derivatives of the solution for the Dirichlet problem (numerical test 1).

3.2.2. Superconvergence for the Neumann problem (numerical test 2)

Consider Γ implicitly defined as the level 0 of $\boldsymbol{\psi}$ in polar coordinates:

$$\boldsymbol{\psi}(r \cos \theta, r \sin \theta) = -r - 0.25 \cos^2\left(\theta + \frac{\pi}{4}\right) + 0.4, \quad \forall (r, \theta) \in \mathbb{R}^+ \times \mathbb{R}/2\pi\mathbb{Z}.$$
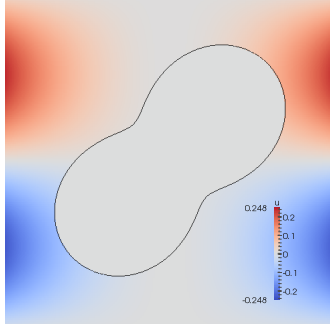
The exact solution of the Neumann problem (22) is given by

$$\mathrm{u}(x, y) = x^2 \sin\left(2\,\pi\,y\right),$$

in the outer area $\boldsymbol{\omega}$, where $\boldsymbol{\psi} < 0$. Table 4 shows the expected superconvergent results of the numerical accuracy tests in maximum norm.

Number of points	u		$\nabla^h u$		$\nabla \cdot \nabla^h u$	
	L^∞-error	order	L^∞-error	order	L^∞-error	order
96	4.488×10^{-5}	-	1.096×10^{-3}	-	1.035×10^{-2}	-
144	1.992×10^{-5}	2.00	4.918×10^{-4}	1.98	4.633×10^{-3}	1.98
216	8.944×10^{-6}	1.99	2.201×10^{-4}	1.98	2.069×10^{-3}	1.99
324	3.970×10^{-6}	1.99	9.825×10^{-5}	1.98	9.222×10^{-4}	1.99
486	1.766×10^{-6}	1.99	4.380×10^{-5}	1.99	4.107×10^{-4}	1.99

Table 4: Superconvergence results for the Neumann problem (numerical test 2).

Figure 7: Plot of the numerical solution of the Neumann problem (Test-case 2).

3.2.3. Minimal requirements on data to ensure superconvergence on two levels and transmission of superconvergence properties

We have numerically highlighted superconvergence properties on two levels for the solution of the Poisson's equation with an exact boundary condition. As the distance to the interface is computed with (5), the interface location is actually known with a fourth order accuracy. This small perturbation does not change significantly the results compared to an exact interface location. However, stronger perturbations would have lead to the loss of superconvergence. For instance, in the case of the Dirichlet problem, any third order perturbation only gives one level of superconvergence, while any second order perturbation does not give any superconvergence property. It is consistent with the fact that, as noted in several studies, the cornerstone of superconvergence lies on the Laplace operator consistancy. Perturbing the data is equivalent to use a pertubed Laplace operator, with exact data. At first glance, a fourth order perturbation on the level set function or on the Dirichlet boundary data implies a perturbed Laplace operator which keeps a second order truncation error, leading to the preservation of the superconvergence properties. A third order perturbation gives a first order truncation error near the interface and logically leads to the same superconvergent behavior on one level, as for the Shortley-Weller discretization. Finally, a second order perturbation implies the inconsistency of the operator near the interface, leading to second order accurate solution without superconvergence, as Gibou *et al.* For the Neumann problem, the reasoning is similar with one less order of accuracy on the numerical boundary data g, as suggested by (17).

In the case of the coupled problem, we cannot expect better than a second order accuracy on the data, especially g and ψ, since each subproblem is solved with a second order method. Indeed, the second order solution of one subproblem gives the boundary data of the second problem, that cannot be more accurate than order 2.

In this subsection we exhibit specific second order perturbation of data that lead to superconvergence on two levels. More precisely, we highlight that the data can be only second order accurate, provided that they have themselves some superconvergent properties: this is the transmission of superconvergence properties.

23

Perturbation of data. Let $\boldsymbol{\varphi}$ be a function of class $C^2(\boldsymbol{\omega})$, and suppose that its direct approximation $\varphi_{i,j}$ on the Cartesian grid has the following form

$$\varphi_{ij} = \boldsymbol{\varphi}(\mathbf{x}_{ij}) + \boldsymbol{\pi}(\mathbf{x}_{ij})\, h^2 + r_{ij}\, h^{2+p}, \qquad \forall\, \mathbf{x}_{ij} \in \omega^h, \tag{24}$$

where $\boldsymbol{\pi}$ is a function of class $C^\infty(\boldsymbol{\omega})$ and $(r_{ij})_{i,j}$ are random numbers that ensure that derivatives of order higher than p are not second order accurate. Hence φ is superconvergent (towards $\boldsymbol{\pi}$) on p levels, for given $p \geq 1$.

Remark 9 (Superconvergence on Γ^h). *For the seek of simplicity, and by abuse of language, a function g is said superconvergent on Γ^h if it is the restriction of a superconvergent scalar field G on ω^h.*

In what follows, expression (24) is used to perturb data of the problem and show the transmission of their superconvergent properties to the solution.

Perturbed Poisson's problems (numerical tests 3 and 4). Consider the same problems as subsections 3.2.1–3.2.2. Both Dirichlet and Neumann problems are still solved separately and the data are perturbed by

$$f_{i,j} = \mathrm{f}(x_{i,j}) + \mathcal{R}_f\, h^2, \quad \text{on } \omega^h, \tag{25}$$

$$\psi_{i,j} = \boldsymbol{\psi}(x_{i,j}) + \sin(\pi\, x\, y)\, h^2 + \mathcal{R}_\psi\, h^p, \quad \text{on } \mathcal{P}^h, \tag{26}$$

$$g^\Gamma = \mathrm{g}(x^\Gamma) + \cos(\frac{\pi}{2}\, x^\Gamma)\, h^2 + \mathcal{R}_g\, h^q, \quad \text{on } \Gamma^h, \tag{27}$$

where $\mathcal{R}_f$, $\mathcal{R}_\psi$ and $\mathcal{R}_g$ are fields of random numbers of $[-5, 5]$. For both problems, we set $p = 4$, which means that the interface location, its normal vector and the curvature are second order accurate. The numerical function f is also second order accurate in order to preserve the second order truncation error of the Laplace operator. The Dirichlet problem is solved by setting $q = 4$ so that the boundary data g is superconvergent on two levels. For the Neumann problem, we just consider one level of superconvergence for the boundary datum by setting $q = 3$. Table 5 shows the results of the numerical accuracy tests. Surprisingly, the solution is superconvergent on two levels for both problems.

The same tests are successively performed with $p = 3$ and then using $q - 1$ instead of q for each problem (see results in Table 6). In each of these numerical tests, one level of superconvergence is lost, which confirms the minimal requirements to achieve two levels of superconvergence.

Effect of the outer-inner coupling (numerical test 5). The consequence of the previous results is the preservation of the superconvergence by coupled problems. Solve now the static coupled problem (1a)-(1b)–(1c)–(1d) with the following continuous data:

$$\boldsymbol{\psi}(r\cos\theta, r\sin\theta) = r + 0.5\, \cos^2\theta\, \sin^2\theta - 0.4, \quad \text{in } \mathcal{P}^h, \tag{28}$$

$$\mathrm{g} = \sin^2(\pi\, (x + y)), \quad \text{on } \Gamma^h, \tag{29}$$

$$\mathrm{f} = 0, \quad \text{in } \omega^h. \tag{30}$$

	Number of points	u L^∞-error	order	$\nabla^C u$ L^∞-error	order	$\nabla^Q \cdot \nabla^C u$ L^∞-error	order
Dirichlet problem	96	1.621×10^{-4}	-	3.161×10^{-3}	-	1.174×10^{-1}	-
	144	7.287×10^{-5}	1.97	1.335×10^{-3}	2.13	6.367×10^{-2}	1.51
	216	3.248×10^{-5}	1.98	5.773×10^{-4}	2.10	2.467×10^{-2}	1.92
	324	1.444×10^{-5}	1.99	2.580×10^{-4}	2.06	1.091×10^{-2}	1.95
	486	6.423×10^{-6}	1.99	1.151×10^{-4}	2.04	5.011×10^{-3}	1.94
Neumann problem	96	5.734×10^{-5}	-	1.105×10^{-3}	-	1.092×10^{-2}	-
	144	2.594×10^{-5}	1.96	4.963×10^{-4}	1.97	5.021×10^{-3}	1.92
	216	1.165×10^{-5}	1.97	2.219×10^{-4}	1.98	2.249×10^{-3}	1.95
	324	5.235×10^{-6}	1.97	9.916×10^{-5}	1.98	1.008×10^{-3}	1.96
	486	2.324×10^{-6}	1.98	4.419×10^{-5}	1.98	4.478×10^{-4}	1.97

Table 5: Transmission of superconvergent properties for the Dirichlet problem (numerical test 3, $p = q = 4$) and the Neumann problem (numerical test 4, $p = 4$, $q = 3$).

Number of points	$\nabla^Q \cdot \nabla^C u$ in Dirichlet problem				$\nabla^Q \cdot \nabla^C u$ in Neumann problem			
	$p = 3$, $q = 4$ L^∞-error	order	$p = 4$, $q = 3$ L^∞-error	order	$p = 3$, $q = 3$ L^∞-error	order	$p = 4$, $q = 2$ L^∞-error	order
96	1.411×10^{-1}	-	5.760×10^{-1}	-	1.121×10^{-2}	-	1.457×10^{-1}	-
144	5.980×10^{-2}	2.12	4.744×10^{-1}	0.48	5.080×10^{-3}	1.95	1.546×10^{-1}	-0.15
216	3.796×10^{-2}	1.62	2.873×10^{-1}	0.86	3.857×10^{-3}	1.32	6.511×10^{-2}	0.99
324	2.341×10^{-2}	1.48	2.326×10^{-1}	0.75	3.661×10^{-3}	0.92	5.471×10^{-2}	0.81
486	1.450×10^{-2}	1.40	1.849×10^{-1}	0.70	2.561×10^{-3}	0.91	4.192×10^{-2}	0.77

Table 6: Loss of one level of superconvergence if the minimal requirements on data are not satisfied.

The functions ψ and g are perturbed on the grid with (26) and (27), by setting $p = 4$ and $q = 3$, which are the minimal requirements to get superconvergence on two levels for the solution to the Neumann problem.
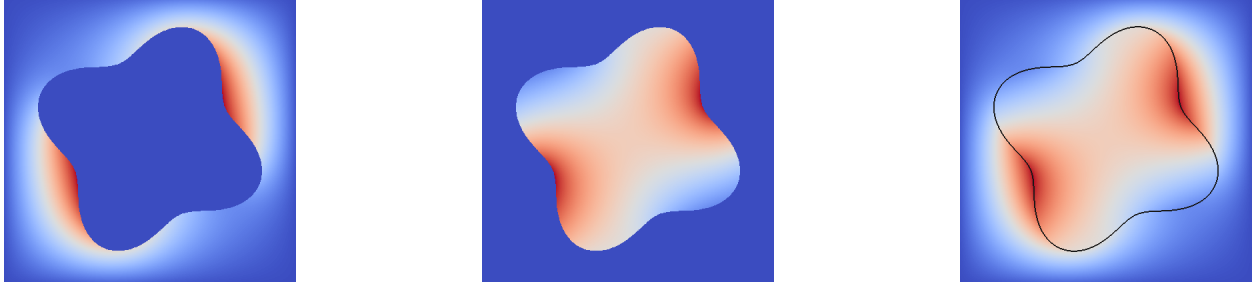
The only specificity of the Neumann-Dirichlet coupling lies in the interface transmission between the outer solution $c^\star$ and the inner solution σ, which is performed by reconstructing the interface value of $c^\star$ thanks to the formula (17). Figure 8 shows a plot of the numerical solutions computed on a 1000×1000 grid and which is used as a reference solution, since we do not know the analytic expression of the exact solutions $c^\star$ and σ. As shown in Table 7, two levels of superconvergence are obtained for both outer and inner solutions, exhibiting thus a preservation of the superconvergence properties by the coupling.

3.3. Superconvergence preservation by the dynamics (numerical test 6)

The interface Γ at the initial time is the circle of center $(0,0)$ and radius 0.25. Let consider an exact velocity $\mathbf{v}$, which explicitly depends on the time variable t, and whose components are given in the inner area $\mathcal{O}_t^i$ by

$$\mathrm{v}_x = 0.8 \cos\left(2\,\pi\,(y + 0.35)\right) \sin\left(\pi\,x\right) \sin\left(\pi\,(0.6 - 2t)\right), \tag{31}$$

$$\mathrm{v}_y = -0.5 \cos\left(\pi\,x\right) \sin\left(\pi\,(y^2 - 0.35^2)\right) \sin\left(\pi\,(0.5 - 2t)\right). \tag{32}$$

Figure 8: Numerical reference solution of the coupled problem (numerical test 5). Left: solution $c^\star$ of the Neumann problem. Middle: solution σ of the Dirichlet problem. Right: merged solutions.

	Number of points	u L^∞-error	order	$\nabla^{\mathsf{C}} u$ L^∞-error	order	$\nabla^{\mathsf{Q}} \cdot \nabla^{\mathsf{C}} u$ L^∞-error	order
	96	7.899×10^{-5}	-	2.148×10^{-3}	-	1.949×10^{-1}	-
	144	3.277×10^{-5}	2.17	9.569×10^{-4}	1.99	8.776×10^{-2}	1.97
$c^\star$	216	1.716×10^{-5}	1.60	4.296×10^{-4}	1.98	4.874×10^{-2}	1.45
	324	7.347×10^{-6}	2.09	2.029×10^{-4}	1.85	2.046×10^{-2}	2.14
	486	2.857×10^{-6}	2.33	8.033×10^{-5}	2.28	8.871×10^{-3}	2.06
	96	7.685×10^{-5}	-	6.497×10^{-3}	-	1.291×10^{0}	-
	144	3.254×10^{-5}	2.12	3.429×10^{-3}	1.58	6.496×10^{-1}	1.69
σ	216	1.731×10^{-5}	1.56	1.681×10^{-3}	1.76	3.311×10^{-1}	1.66
	324	7.634×10^{-6}	2.02	7.277×10^{-4}	2.06	1.538×10^{-1}	1.89
	486	2.825×10^{-6}	2.45	2.820×10^{-4}	2.34	6.656×10^{-2}	2.06

Table 7: Accuracy of the numerical solutions $c^\star$ and σ and their derivatives for the static part of Problem 1 (numerical test 5).

The initial level set function is perturbed with (26) and $p = 4$, while at each time step the velocity components in $\mathcal{O}_t^{i,h}$ are similarly perturbed thanks to

$$v_x = \mathrm{v}_x + (1 - x^2)\, h^2 + \mathcal{R}_{v_x}\, h^3, \tag{33}$$

$$v_y = \mathrm{v}_y + 2y\, \sin(\pi\, xy)\, h^2 + \mathcal{R}_{v_y}\, h^3. \tag{34}$$

The velocity is therefore superconvergent on one level at each time step. The level set function is advected until the final time $T = 0.5$. The errors on ψ, $\boldsymbol{v}$ and their derivatives are computed at the final time by comparison with a reference simulation on a 1000×1000 grid, and in a tubular area around the level 0 of the level set function so as to avoid the ridges, which are generated by the velocity extension.

Table 8 shows the superconvergent results of the numerical accuracy tests in maximum norm, for the level set function ψ, the extended velocity $\boldsymbol{v}$ and their derivatives.

Figure 9 shows plots of the interface location and the x-component of the extended velocity at the initial and final times. The point $(0, 0)$ in Figure 9(a) and the ridges in Figure 9(b) are singularities, which can disturb the advection or the computation of errors if they are too close to the interface. Note that it does not happen in the simulations presented in this study. Nonetheless,

Number of points	ψ L^∞-error	order	$\nabla^{\mathsf{C}}\psi$ L^∞-error	order	$\nabla^{\mathsf{Q}}\cdot\nabla^{\mathsf{C}}\psi$ L^∞-error	order	$\boldsymbol{v}$ L^∞-error	order	$\nabla^{\mathsf{Q}}\boldsymbol{v}$ L^∞-error	order
96	2.430×10^{-4}	-	1.009×10^{-2}	-	3.560×10^{-1}	-	5.934×10^{-4}	-	2.858×10^{-2}	-
144	9.738×10^{-5}	2.26	4.272×10^{-3}	2.12	1.626×10^{-1}	1.93	2.810×10^{-4}	1.84	1.404×10^{-2}	1.75
216	3.976×10^{-5}	2.23	1.841×10^{-3}	2.10	7.563×10^{-2}	1.91	1.240×10^{-4}	1.93	6.135×10^{-3}	1.90
324	1.466×10^{-5}	2.31	7.027×10^{-4}	2.19	2.822×10^{-2}	2.08	4.612×10^{-5}	2.10	2.494×10^{-3}	2.01
486	5.729×10^{-6}	2.31	2.693×10^{-4}	2.23	1.000×10^{-2}	2.20	1.762×10^{-5}	2.17	1.001×10^{-3}	2.07

Table 8: Preservation of superconvergence after velocity extension and advetion (numerical test 6).

(a) initial time   (b) final time
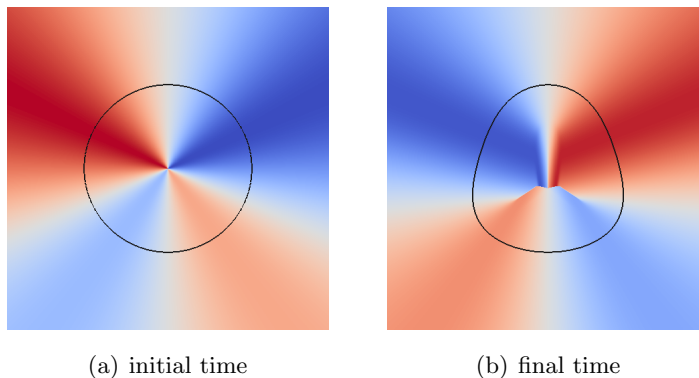
Figure 9: Interface location (black line) and x-component of the extended velocity (w_x) at the initial and final times.

this problem can be overcome by occasionally reinitializing the level set function to a distance function [30], especially when the interface moves closer to an area where a ridge is previously appeared. In this case, the *redistanciation* should be performed at the order 4 in order not to degrade the superconvergence properties.

4. Conclusion

We have proposed a second order Cartesian method to solve the moving boundary problem of invadopodia. Our method is based upon the understanding and the full use of superconvergent properties on two levels and their transmission. These properties result from a specific method to solve Poisson equation, built thanks to a stable ghost fluid method and cubic extrapolations of the ghost values. This method can handle both Dirichlet and Neumann conditions on the boundary of the domain. Throughout numerical tests on 2D-domains, we have highlighted and classified the superconvergence properties, with respect to the boundary condition. We have also identified the required criteria of accuracy on the boundary data and on the interface location to achieve the superconvergence: the data do not have to be high-order accurate but just only superconvergent themselves, which is particularly suitable for coupling. If data are second order accurate and superconvergent, and despite the nonconsistency of the equivalent perturbed Laplace operator, the

solution, its gradient and the divergence of its gradient (and all the second order derivatives) are also second order accurate. Moreover, these properties are stable through dynamics thanks to an appropriate velocity extension and a second order scheme for advection.

The transmission of superconvergence properties is one of the main results of the paper. While superconvergence has been studied and proved in other works for the case of exact data, the transmission of superconvergence properties from second order superconvergent data to the solution is rarely addressed and is still an open issue. We are confident that this study is a step towards the theoretical understanding of this transmission phenomenon.

We emphasize that the method may be useful to many coupled problems similar to invadopodia. More precisely, this is well suited to solve free boundary problems of Hele-Shaw type, in which the interface velocity is the gradient of an harmonic function.

Obviously, our method use wide stencils. The parallel implementation of the code –for a three-dimensional use for instance– is complex to carry out, requiring four communication layers. This is the cost to get superconvergence properties on two levels, which result in the second order accuracy of the solutions, including the curvature. This point is of great interest for further modeling of cell migration processes as protrusion and rear retraction, using interface regularization techniques.

However, if second order accuracy and/or two levels of superconvergence are not required, our study can be generalized to similar methods, with less accurate extrapolations. Based on a previous study in [15], results of superconvergence and minimal requirements on data for each case (linear, quadratic and cubic extrapolations) are summarized, without details, in Table 9. Thus, each problem that requires superconvergence can be solved with the most suitable method (and the least computationally expensive as possible), with respect to the desired accuracy and the coupling degree. In particular, the first order accuracy observed in [16] can be explained by those results.

Finally, we must mention many remaining issues, which are a crucial point for concrete applications. First, the behavior of the time scheme is not yet well understood, possibly because the stability condition depends on the spatial discretization. In the cubic case, the discretization is not only second order accurate but also superconvergent on two levels. This could require a nonlinear CFL, in the same spirit as in [11]. Using this kind of condition, some instabilities encountered on very refined meshes disappear. The CFL analysis is therefore essential.

Another issue is related to the appearance of many pathological configurations, with too few points, which may lead to a local loss of superconvergence. Similarly, the emergence of strong curvatures and shocks near the interface during advection of the level set can also impact the results of convergence and the accuracy of solutions. These situations generally arise because of a too coarse grid, that is not always possible to refine, and because of the fixed nature of Cartesian grid. For all these sensitive cases, patches are possible but often cumbersome to implement. For this reason, a real challenge would be to extend this study to adaptive meshes, such as quadtree and octree meshes.

	Dirichlet			Neumann		
	L-scheme	Q-scheme	C-scheme	L-scheme	Q-scheme	C-scheme
Order of accuracy required on data						
G	2	2	2	1	2	2
$\nabla^{\mathbb{L}} G$		2	2			2
$\nabla^{\tilde{\mathbb{L}}} \cdot \nabla^{\mathbb{L}} G$			2			
ψ	2	2	2	1	2	2
$\nabla^{\mathbb{L}} \psi$		2	2	1	2	2
$\nabla^{\tilde{\mathbb{L}}} \cdot \nabla^{\mathbb{L}} \psi$			2			2
Order of accuracy of the solution						
u	2	2	2	1	2	2
$\nabla^{\mathbb{L}} u$	1	2	2	1	2	2
$\nabla^{\tilde{\mathbb{L}}} \cdot \nabla^{\mathbb{L}} u$		1	2		1	2

Table 9: Zoology of the Cartesian numerical methods to solve the Poisson equation, and their superconvergence properties, with minimal requirements on data accuracy, based on [15]. L-scheme, Q-scheme and C-scheme are built with linear, quadratic and cubic extrapolations, respectively. The interface is implicitely defined by the level set function ψ and G denotes any lifting of the boundary datum g. $\mathbb{L}$ stands for L in the case of the L-scheme and $(\tilde{\mathbb{L}}, \mathbb{L})$ stands for either (L, Q) or (Q, C) respectively for either the Q-scheme or the C-scheme.

References

[1] D Adalsteinsson and J.A Sethian. The fast construction of extension velocities in level set methods. *J. Comput. Phys.*, 148(1):2–22, January 1999.

[2] Ivo Babuška, Uday Banerjee, and John E. Osborn. Superconvergence in the generalized finite element method. *Numerische Mathematik*, 107(3):353–395, 2007.

[3] Chunjia Bi. Superconvergence of finite volume element method for a nonlinear elliptic problem. *Numerical Methods for Partial Differential Equations*, 23(1):220–233, 2007.

[4] J. H. Bramble and B. E. Hubbard. On the formulation of finite difference analogues of the Dirichlet problem for Poisson's equation. *Numerische Mathematik*, 4(1):313–327, 1962.

[5] O. Bruno and D. Hoch. Numerical differentiation of approximated functions with limited order-of-accuracy deterioration. *SIAM Journal on Numerical Analysis*, 50(3):1581–1603, 2012.

[6] Han Chen, Chohong Min, and Frédéric Gibou. A Supra-Convergent Finite Difference Scheme for the Poisson and Heat Equations on Irregular Domains and Non-Graded Adaptive Cartesian Grids. *Journal of Scientific Computing*, 31(1–2):19–60, 2007.

[7] Han Chen, Chohong Min, and Frédéric Gibou. A numerical scheme for the Stefan problem on adaptive Cartesian grids with supralinear convergence rate. *Journal of Computational Physics*, 228(16):5803–5818, 2009.

[8] So-Hsiang Chou and Xiu Ye. Superconvergence of finite volume methods for the second order elliptic problem. *Computer Methods in Applied Mechanics and Engineering*, 196(37–40):3706–3712, 2007. Special Issue Honoring the 80th Birthday of Professor Ivo Babuška.

[9] Philippe G. Ciarlet. Discrete maximum principle for finite-difference operators. *aequationes mathematicae*, 4(3):338–352, 1970.

[10] Marco Cisternino and Lisl Weynans. A parallel second order Cartesian method for elliptic interface problems. *Communications in Computational Physics*, 12:1562–1587, June 2012.

[11] Erwan Deriaz. Stability conditions for the numerical solution of convection-dominated problems with skew-symmetric discretizations. *SIAM J. Numer. Anal.*, 50(3):1058–1085, 2012.

[12] Ronald P. Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152(2):457–492, 1999.

[13] José A. Ferreira and Rolf D. Grigorieff. On the supraconvergence of elliptic finite difference schemes. *Applied Numerical Mathematics*, 28(24):275 – 292, 1998.

[14] A. Friedman. *Variational principles and free-boundary problems.* Pure and applied mathematics. Wiley, 1982.

[15] Olivier Gallinato. *Modeling of cancer phenomena and superconvergent methods for the resolution of interface problems on Cartesian grid.* Theses, Université de Bordeaux, November 2016.

[16] Olivier Gallinato, Masahito Ohta, Clair Poignard, and Takashi Suzuki. Free boundary problem for cell protrusion formations: theoretical and numerical aspects. *Journal of Mathematical Biology*, pages 1–45, 2016.

[17] Frédéric Gibou and Ronald 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(2):577–601, January 2005.

[18] Frédéric Gibou, Ronald P. Fedkiw, Li-Tien Cheng, and Myungjoo Kang. A Second-Order-Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains. *Journal of Computational Physics*, 176:205–227, 2002.

[19] Thomas Y. Hou, Zhilin Li, Stanley Osher, and Hongkai Zhao. A hybrid method for moving interface problems with application to the heleshaw flow. *Journal of Computational Physics*, 134(2):236 – 252, 1997.

[20] Z. Jomaa and C. Macaskill. The ShortleyWeller embedded finite-difference method for the 3D Poisson equation with mixed boundary conditions. *Journal of Computational Physics*, 229(10):3675–3690, 2010.

[21] Shilpa Khatri and Anna-Karin Tornberg. An embedded boundary method for soluble surfactants with interface tracking for two-phase flows. *Journal of Computational Physics*, 256:768 – 790, 2014.

[22] Heinz-Otto Kreiss, N. Anders Petersson, and Jacob Ystrm. Difference approximations of the neumann problem for the second order wave equation. *SIAM Journal on Numerical Analysis*, 42(3):1292–1323, 2004.

[23] Nick Levine. Superconvergent Recovery of the Gradient from Piecewise Linear Finite-element Approximations. *IMA Journal of Numerical Analysis*, 5(4):407–427, 1985.

[24] Zi-Cai Li, Hsin-Yun Hu, Qing Fang, and Tetsuro Yamamoto. Superconvergence of Solution Derivatives for the ShortleyWeller Difference Approximation of Poisson's Equation. II. Singularity Problems. *Numerical Functional Analysis and Optimization*, 24(3-4):195–221, 2003.

[25] Zi-Cai Li, Hsin-Yun Hu, Song Wang, and Qing Fang. Superconvergence of solution derivatives of the ShortleyWeller difference approximation to Poisson's equation with singularities on polygonal domains. *Applied Numerical Mathematics*, 58(5):689–704, 2008.

[26] Zi-Cai Li, Tetsuro Yamamoto, and Qing Fang. Superconvergence of solution derivatives for the ShortleyWeller difference approximation of Poisson's equation. part I: smoothness problems. *Journal of Computational and Applied Mathematics*, 151(2):307–333, 2003.

[27] Nami Matsunaga and Tetsuro Yamamoto. Superconvergence of the Shortley–Weller approximation for Dirichlet problems. *Journal of Computational and Applied Mathematics*, 116(2):263–273, 2000.

[28] Chohong Min, Frédéric Gibou, and Hector D. Ceniceros. A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids. *Journal of Computational Physics*, 218(1):123–140, 2006.

[29] S. Osher and J.A. Sethian. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[30] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2003.

[31] Joseph Papac, Frédéric Gibou, and Christian Ratsch. Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions. *Journal of Computational Physics*, 229(3):875–889, 2010.

[32] George H. Shortley and Royal Weller. The Numerical Solution of Laplace's Equation. *Journal of Applied Physics*, 9(5):334, 1938.

[33] R. F. Warming and Richard M. Beam. Upwind second-order difference schemes and applications in aerodynamic flows. *AIAA Journal*, 14(9):1241–1249, 2016/07/06 1976.

[34] Lisl Weynans. A proof in the finite-difference spirit of the superconvergence of the gradient for the Shortley-Weller method. Inria Research Report RR–8757. https://hal.inria.fr/hal-01176994.

[35] Tetsuro Yamamoto. *Symbolic Algebraic Methods and Verification Methods*, chapter A New Insight of the Shortley-Weller Approximation for Dirichlet Problems, pages 245–253. Springer Vienna, Vienna, 2001.

[36] Gangjoon Yoon and Chohong Min. A review of the supra-convergences of Shortley-Weller method for Poisson equation. *J. KSIAM*, 18(1):51–60, 2014.

[37] Gangjoon Yoon and Chohong Min. Convergence Analysis of the Standard Central Finite Difference Method for Poisson Equation. *Journal of Scientific Computing*, pages 1–16, 2015.

[38] Tie Zhang and Shuhua Zhang. The gradient superconvergence of the finite volume method for a nonlinear elliptic problem of nonmonotone type. *Applications of Mathematics*, 60(5):573–596, 2015.

[39] Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127(1):179 – 195, 1996.

[40] Milŏs Zlámal. Superconvergence and reduced integration in the finite element method. *Mathematics of Computation*, 32(143):663–685, 1978.