A hybrid level-set / embedded boundary method applied to solidification-melt problems

Alexandre Limare, Stéphane Popinet, Christophe Josserand, Zhonghan Xue, Arthur R. Ghigo

A hybrid level-set / embedded boundary method applied to solidification-melt problems

A. Limare[a], S. Popinet[b], C. Josserand[c], Z. Xue[c,d], A. Ghigo[e]

[a] *Fluid mechanics department, ArianeGroup, Les Mureaux, 78130, Les Mureaux, France*
[b] *LadHyX, Centre National de la Recherche Scientifique, Ecole polytechnique, Institut polytechnique de Paris, 91120, Palaiseau, France*
[c] *Institut Jean le Rond ∂'Alembert, Sorbonne Université, Centre National de la Recherche Scientifique, UMR 7190, 75005, Paris, France*
[d] *State Key Laboratory for Strength and Vibration of Mechanical Structures, School of Aerospace, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China*
[e] *Department of Mathematics, British Columbia University, 1984 Mathematics Road, Vancouver, BC V6T 1Z4, Canada*

Abstract

In this paper, we introduce a novel way to represent the interface for two-phase flows with phase change. We combine a level-set method with a Cartesian embedded boundary method and take advantage of both. This is part of an effort to obtain a numerical strategy relying on Cartesian grids allowing the simulation of complex boundaries with possible change of topology while retaining a high-order representation of the gradients on the interface and the capability of properly applying boundary conditions on the interface. This leads to a two-fluid conservative second-order numerical method. The ability of the method to correctly solve Stefan problems, onset dendrite growth with and without anisotropy is demonstrated through a variety of test cases. Finally, we take advantage of the two-fluid representation to model a Rayleigh–Bénard instability with a melting boundary.

1. Introduction

Liquid–solid phase change (solidification or melting) is present in many industrial processes, particularly in metallurgy [1] and 3-D printing [2]. Controlling ice formation and accretion is also crucial in aeronautics with a recent increasing interest due to the evolution of safety policies [3]. More generally, icing dynamics control a large number of important environmental processes [4], such as sea-ice formation [5, 6] or permafrost thawing [7]. From an industrial point of view, reproducible solidification processes which create complex geometries for solid materials with isotropic properties at a low cost have been a goal pursued for decades. Complex shape generation generally involves putting the matter in a liquid state as an intermediary step before solidifying it, hence the need to have a good knowledge of the process of solidification. This process is difficult to study experimentally and often requires the use of intrusive or sometimes destructive methods. Similarly, experimental studies on icing often provide partial measurements only (surface temperature for instance) even when they are made in controlled conditions [8, 9, 10, 11, 12, 13].

Numerical methods able to accurately simulate the process of solidification and/or melting are thus of particular interest. Developing these methods is especially challenging however, since melting and solidification processes combine multiple difficulties. The first difficulty is classical and common to all free-boundary problems: how to accurately describe and follow the evolution of a complex boundary? This can be seen essentially as a geometric and kinematic problem and a broad range of methods have been proposed to solve it. The second difficulty concerns the dynamics of this motion (i.e. the relation between accelerations and forces) and requires the development of methods able to accurately couple the geometry of the boundaries with the underlying equations of motion. This coupling is clearly "higher-order" (in the sense of space/time derivatives) than the kinematic problem and thus more difficult to solve. A representative example is the approximation of surface tension terms which has been particularly challenging (see [14] for a review).

In fact, this coupling is especially difficult in the case of solidification/melting since the dynamics are driven almost entirely by singular terms on the boundary, such as temperature gradient jumps [15]. In the case of dendritic crystallisation the boundary topology can also become extremely complex and boundary-discontinuity difficulties can be compounded by the appearance of metastable states, for example in supercooled liquids.

A classical and accurate way to deal with partial differential equations with jumps is to use *boundary conforming* discretisation techniques combined with a Finite Volume Method (FVM) which ensures discrete and global mass conservation. In a boundary conforming framework, the mesh is constructed so that the edges of discretisation elements (for example triangles in 2D or tetrahedra in 3D) always coincide with the boundaries. Accurate jumps in the solutions can then be obtained by imposing the discrete boundary conditions directly on the edge of boundary elements. This allows in principle to design numerical schemes of arbitrary order of accuracy. The main limitation of these techniques is that they are inherently *Lagrangian* i.e. they are

most easily formulated in a Lagrangian frame of reference and are thus in principle limited to small material deformations, such as occur for example in solid mechanics. While techniques exist to overcome this limitation, such as Lagrangian-remapping [16], they are usually complex and costly and still have difficulties dealing with complex topology changes such as merging and splitting.

This limitation of boundary-conforming techniques has led to the development of a broad range of methods able to couple general boundaries with the Eulerian framework more suitable to the discretisation of the equations of fluid motion. The issue then becomes: how to represent jumps/boundary conditions now that discrete boundaries do not coincide with real boundaries? The solution adopted by almost all methods to date is to approximate these (surface) jumps with localised *volumetric* terms which naturally fit within an Eulerian framework. This can be seen as replacing true Heaviside/Dirac functions with continuous/differentiable approximations and has a long history, dating back at least to the pioneering papers of Peskin [17, 18].

A direct consequence of this approximation of discontinuous functions by differentiable approximations is that the resulting schemes can be at most first-order accurate spatially (by Godunov's theorem), in contrast with the boundary-conforming schemes mentioned earlier. This slow convergence is particularly problematic for applications which are mostly driven by interfacial terms, such as solidification and melting.

The goal of the present article is thus to lift this severe limitation and to present a Finite-Volume method able to deal with arbitrary boundary deformations, while conserving mass and preserving at least second-order spatial accuracy for the discretisation of boundary conditions and the overall solution.

2. A brief review of existing schemes

Non-boundary-conforming methods can can be classified in two families: front-tracking methods where one stores explicitly the position of the interface and front-capturing methods where the interface position is defined indirectly. In both cases, the numerical methods often consider no density variation between the liquid and solid phases: in fact taking into account such a density variation can be done by imposing a specific boundary condition for the velocity at the interface. However, this does not affect the numerical challenge of the phase change and it can be modeled separately (see for instance [19, 20]).

Juric and Tryggvason [21] for instance combined an explicit tracking of massless Lagrangian particles and an immersed boundary method. Another type of method based on the cellular automaton can also be used [22, 23, 24] often to study grain growth at the meso-scale. Reuther and Rettenmayr [25] simulated the dendritic solidification using an anisotropy-free meshless front-tracking method. However, the main drawbacks of these tracking methods are their difficulty to cope with change of topology and their complex extension to 3D.

In the second category, the interface is expressed implicitly using some auxiliary variables defined on every cell, for which values are ranging usually between zero and unity. Among others, one can cite the enthalpy method of Voller [26] where the phase change occurs over a restricted temperature range and the solid-liquid interface is described as a mushy zone. Another family is the Volume of Fluid (VOF) method which ensures mass conservation [27, 28]. Yet another widespread method is the phase field method [29, 30, 31, 32, 33] which explicitly relies on a smooth, differentiable field representing phase transition. But, as discussed in the introduction, a large number of grid points in this transition zone is required for convergence. Therefore, although phase-field methods are classical tools for modeling liquid/solid phase transition, it is crucial to develop alternative methods where the solidification/melting front is sharp and for which the heat flux jump at the interface is well defined.

The level-set method [34, 35] is thus a natural way to represent the interface which is simply a level set (usually the zero value) of a function defined in the calculation domain. Levet-set methods are well suited for modeling time-dependent, moving-boundary problems but also have their own specific drawbacks; they do not preserve mass/volume well in their original formulation, they introduce a smearing of the interface and reduce to low order accuracy regions where characteristics of the flow merge (*i.e.* caustic singularities). Furthermore, additional difficulties arise also for the imposition of a flux jump condition on an interface and the associated construction of extension velocities. However, level-set methods are quite straightforward to implement, versatile enough to be combined with another method and their advantages and drawbacks, linked to the mathematical properties of the equations at play, have been studied quite thoroughly. Solutions have been found for applying an immersed boundary condition using a finite-difference treatment for the variables, for instance the LS-STAG method [36], the Immersed Boundary Smooth Extension [37] or the Ghost Fluid method [38]. Note that all these methods can be shown to still rely on smooth approximations of Dirac/Heaviside functions [14], and are thus only first-order accurate spatially.

On the other hand, cartesian embedded-boundary or *cut-cell* methods have been extensively used for a large range of flows [39, 40, 41]. They rely on a finite-volume discretization where cells are arbitrarily intersected by an embedded boundary. These methods show a second-order accuracy when applying immersed boundary condition and are conservative [42]. From an engineering point of view, this also greatly eases the mesh generation process. The main drawbacks of such methods are linked to grid irregularities in the cut regions which introduce local

variations in truncation errors. This is all the more critical when the motion of the boundary is controlled by quantities calculated on the interface such as skin friction [43] or temperature gradients for phase change.

An important trend of the last two decades for numerical phase change models has been to create hybrid methods to compensate some of their shortcomings. For instance, phase change with VOF is especially hard since it has no built-in way of imposing Dirichlet conditions exactly on the interface, therefore it is often combined with other non-conservative methods which are able to impose a boundary condition on the interface. Sussman and Puckett [44] combined the VOF and level set methods; VOF ensures conservative properties whereas the level set method provides accurate geometric informations such as normals and curvature. An extension of this method called CLSMOF was introduced in [45] with application to the freezing of supercooled droplets in [46]. Recently, a hybrid VOF-IBM (Immersed Boundary Method) method has been developed for the simulation of freezing films and drops [19]. Finally, we point out that an approach similar to the one we will introduce has been developed by Udaykumar et al. [47, 48, 49, 50]. We will show hereafter how our method differs, in particular with respect to discrete conservation properties and the associated representation of coupling boundary conditions at the phase change interface.

In the present article, we introduce a novel hybrid method by combining a level-set representation of the interface with a cut-cell method for the immersed boundary condition. Although it combines well known numerical schemes, this new method presents the advantage of considering a sharp interface for the melting/solidification front, at which gradients (for instance heat fluxes) can be computed accurately for each domain. This method is therefore conservative by construction and expected to have a second-order accuracy. We also discuss the implementation of the method in 3D and on adaptive grids.

3. Principle of the method

Most numerical methods take a "one-fluid" approach for multiphase flows, meaning that the computational domain on which the numerical solver is applied contains both phases with a more-or-less smooth change on the physical properties. Here, we develop a two-fluid method, where each phase is described using its own set of equations and variables. These two domains ($\Omega_{\phi+}$ and $\Omega_{\phi-}$ in Figure 1) are coupled through the motion of the boundary Γ and the associated boundary/jump conditions. This approach has two main advantages which are directly related to the similarities with boundary-conforming Lagrangian methods: 1) The set of equations solved in one phase can be different from those in the other phase (*e.g.* a diffusion equation in the solid and a Navier–Stokes equation coupled with advection–diffusion in the fluid), 2) accurate boundary/jump conditions can be imposed on the boundary. Specifically, the same Dirichlet boundary condition is applied on the interface for both phases and the (discontinuous) heat fluxes on the interface are calculated independently for each phase with at least second-order accuracy using finite-volume conservative numerical operators.

The boundary/interface Γ is described using a levelset function ϕ. The domain outside of the interface is defined by $\Omega_{\phi+} = \{\forall \boldsymbol{x}/\phi(\boldsymbol{x}) > 0\}$ and the inside of the interface is defined in a similar manner $\Omega_{\phi-} = \{\forall \boldsymbol{x}/\phi(\boldsymbol{x}) < 0\}$, both are subdomains of the calculation domain Ω. We depicted a possible situation on Fig. 1: in that case the blue domain represents Ω_- and is made of 3 disconnected subdomains. Let T_S and T_L be temperature fields defined respectively in $\Omega_{\phi-}$ and $\Omega_{\phi+}$.

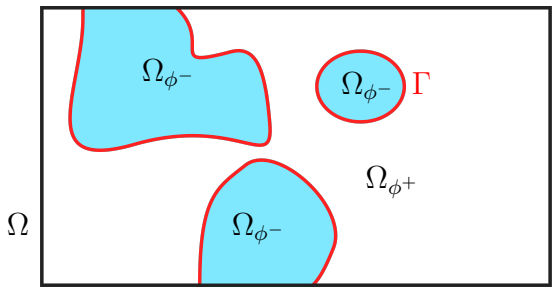


Figure 1: Schematic view of domains used for calculation: in blue $\Omega_{\phi-}$, in white, $\Omega_{\phi+}$, in red the interface Γ

The temperature gradient jump then gives the velocity of the interface, which is the starting point for the construction of a continuous extension velocity which we will refer to as the phase change velocity $\boldsymbol{v}_{pc}$. The assumption here is that the auxiliary field $\boldsymbol{v}_{pc}$ has a meaning in both domains, not only on the interface Γ, thus allowing the transport of the levelset function to solve the kinematic problem.

In the next section we present the physical model and the equations to be solved, which belong to the family of Stefan problems coupled with a velocity field [51]. The details of the coupling between the level-set method for the kinematic problem and the cut-cell technique [52, 42] for the dynamic problem are given in section 5. Finally in Section 6 we present several semi-analytical test cases and a more complex case by Favier *et al.* [53] where the equations solved are simple diffusion in the solid and the Navier–Stokes equations in the liquid.

4. Physical model

The liquid–solid interface denoted Γ separates two phases of a pure material. Its position is determined by a prescribed temperature field at the interface, that is not *a priori* constant and can depend on the interface curvature and velocity, following for instance the Gibbs–Thomson relation. We consider that the solid domain cannot deform and that the liquid one obeys the incompressible Navier–Stokes equations. Within this framework, the energy equation simplifies into a diffusion equation in the solid domain and an advection–diffusion equation in the liquid. The interface dynamics are determined by the difference between the heat fluxes at the interface, following the well-known Stefan equation. The solid and liquid parameters (diffusion coefficients, viscosity and density in the fluid domain in particular) usually depend on the temperature, but we will consider here constant values since this paper focusses on the phase change dynamics, and the generalization to smooth, temperature-dependent parameters does not bring additional numerical challenges. We will only use the Boussinesq approximation for the Navier–Stokes equation to model the Rayleigh–Bénard thermal convective instability during solidification. Our model includes the crucial physical effects for solidification that are undercooling, crystalline anisotropy, surface tension, and molecular kinetics. This allows us to treat all types of solidification/melting problems in particular when the fluid is supercooled and thus study solidification fronts where instabilities occur giving birth to dendrites and fingering [54, 55, 56]. Indeed, dendrites are formed when an ice crystal grows in a supercooled liquid, representing a crucial test for numerical methods dealing with liquid/solid phase transition [21].

Our model thus reduces to the following set of differential equations:

- for the temperature field $T_L(\mathbf{x},t)$ in the fluid domain:

$$\rho_L C_L \left(\frac{\partial T_L}{\partial t} + \mathbf{u} \cdot \nabla T_L \right) = \nabla \cdot (\lambda_L \nabla T_L), \tag{1}$$

- for the temperature field $T_S(\mathbf{x},t)$ in the solid domain:

$$\rho_S C_S \frac{\partial T_S}{\partial t} = \nabla \cdot (\lambda_S \nabla T_S). \tag{2}$$

where $\mathbf{u}(\mathbf{x},t)$ is the velocity field (we consider that the velocity vanishes in the solid domain). We denote with subscript L and S the coefficients related to the liquid and solid respectively and will use i to denote either. ρ_L, C_L and λ_L (ρ_S, C_S and λ_S) are the liquid (solid) density, thermal capacity and thermal conductivity respectively. The velocity field $\mathbf{u}(\mathbf{x},t)$ in the fluid domain obeys the incompressible Navier–Stokes equation that reads in its usual form:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla(\mathbf{u}) \right) = -\nabla p + \nabla \cdot (2\mu \mathbf{D}) + \rho \mathbf{g}, \tag{3}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{4}$$

where $\mathbf{D} = \frac{1}{2}(\nabla(\mathbf{u}) + {}^t\nabla(\mathbf{u}))$ is the deformation tensor and $\mathbf{g}$ the acceleration of gravity. ρ and μ are the density and viscosity that can eventually depend on space through for instance the temperature field. In this paper, we consider that the solver for the velocity field in the liquid domain already exists and we will simply couple it with the temperature field and solidification front dynamics. This coupling will be used in the validation section 6, using the Boussinesq approximation for the density variation with the temperature. This set of differential equations needs to be complemented by the boundary conditions at the interface Γ describing the solidification front. First, the temperature at the solidification front depends on the local interface curvature and velocity, through the so-called Gibbs–Thomson relation [4, 15, 57]

$$\forall \boldsymbol{x} \in \Gamma, \qquad T(\boldsymbol{x},t) = T_m - \epsilon_\kappa \kappa - \epsilon_v v_{pc} \tag{5}$$

where T_m is the melting temperature, κ the local curvature of the interface, v_{pc} the local speed of the interface, ϵ_v, the molecular kinetic coefficient and ϵ_κ the surface tension coefficient. Unless otherwise stated, these two coefficients will be taken as constant in the present study.

Finally, the last equation couples the thermal equations between the two domains (solid and liquid) stating that the solidification front evolves through the balance between the heat flux at the front, the so-called Stefan equation:

$$\rho_S L_H \boldsymbol{v_{pc}} \cdot \boldsymbol{n} = -\left[\lambda \nabla T|_\Gamma\right] \cdot \boldsymbol{n} = \left(\lambda_S \nabla T_S|_\Gamma - \lambda_L \nabla T_L|_\Gamma\right) \cdot \boldsymbol{n} \tag{6}$$

where L_H is the latent heat and $\boldsymbol{n}$ the normal to the interface from solid to liquid. The velocity of the interface $\boldsymbol{v_{pc}}$ is thus related through the latent heat to the jump in the heat flux (and therefore in general to the temperature gradient) across the interface.

Finally, a dimensionless version of this set of equations will be used, introducing reduced temperature, geometrical length and time scales. They usually lead to a dimensionless Stefan number, that compares thermal diffusion and latent heat. Its definition might depend on the specificity of the problem (geometry, boundary conditions). For the sake of simplicity, we will consider later on that the density of the liquid and the solid are the same ($\rho_L = \rho_S = \rho$): although it is not true in general (for the ice/water phase change, we have $\rho_S/\rho_L \sim 0.9$ for instance), it is not a crucial ingredient for the numerics [19]. In fact, taking into account the density variation between the liquid and solid phases would lead to a specific boundary condition for the velocity at the interface, corresponding to the flux induced by the front dynamics [20]. Considering a domain of size L_0 and defining thus a time scale $\tau = L_0^2/D_S$) using the thermal diffusion coefficient in the solid ($D_S = \lambda_S/(\rho_S C_S)$), we obtain the following set of dimensionless equation (defining also $D_L = \lambda_L/(\rho_L C_L)$):

$$\frac{\partial \theta_L}{\partial t} + \mathbf{u} \cdot \nabla \theta_L = \frac{D_L}{D_S} \Delta \theta_L, \tag{7}$$

$$\frac{\partial \theta_S}{\partial t} = \Delta \theta_S, \text{ and} \tag{8}$$

$$\boldsymbol{v}_{pc} \cdot \boldsymbol{n} = \mathrm{St} \left(\frac{\lambda_L}{\lambda_S} \nabla \theta_L|_\Gamma - \nabla \theta_S|_\Gamma \right) \cdot \boldsymbol{n}. \tag{9}$$

We have introduced a reduced temperature, defined using a temperature T_1, coming in general from the boundary conditions and thus depending on the specific problem to investigate, leading typically:

$$\theta_{L,S} = \frac{T - T_m}{T_1 - T_m} = \frac{T - T_m}{\Delta T}.$$

Here $\Delta T = T_1 - T_m$ is supposed to be positive, leading to the following definition of the Stefan number:

$$\mathrm{St} = \frac{C_S \Delta T}{L_H}.$$

The Stefan number thus quantifies the ratio between the available heat in the system with the latent heat. In the following we will in general use this set of dimensionless equations, noting the dimensionless temperature T instead of θ by simplicity.

5. Numerical method

The goal of this paper is to present a numerical method able to solve accurately the thermal equations (1, 2, 5 and 6), that will be coupled with an existing solver for the fluid equation (4). The method will be implemented in the free software *Basilisk* [58]. We use a novel approach for the numerical modelling of the interface by combining a level-set function with an embedded boundary (cut-cell) treatment for the fluxes. This means that for interfacial cells we store two different values for the temperature fields in order to correctly compute the temperature gradient in each phase. We will first describe the choices made for the level-set function and the Cartesian embedded-boundary and then explain how we combined both of these approaches to obtain a consistent numerical description of the physical situation.

5.1. Global algorithm

Our method can be summarized as:

1. Calculate the phase change velocity on the interface $\boldsymbol{v_{pc}}|_\Gamma$
2. Extend or reconstruct a continous phase change velocity field $\boldsymbol{v}_{pc}$ in the vicinity of the interface from the $\boldsymbol{v_{pc}}|_\Gamma$ value
3. Advect the level-set function and recalculate the volume and face fractions
4. Redistance the level-set function
5. Initialize fields of newly emerged cells
6. Apply the appropriate solver for each independent phase
7. Perform mesh adaptation

Key points that will be further detailed are Steps 1 and 2 which combine the level-set representation of the interface for the reconstruction of a continuous field with the calculation of the gradients on the interface relying on the embedded boundary representation of the interface, Step 5 that is critical for the global accuracy of the method and Step 7 which allows efficient calculations.

5.2. The level-set method

The level-set is a method initially designed to study the motion by a velocity field $\boldsymbol{v}$ of an interface Γ of codimension 1 that bounds several open regions Ω (possibly connected)[59]. The main idea is to use a function ϕ sufficiently smooth (Lipschitz continuous for instance) and define the interface as the 0-level-set of ϕ:

$$\forall \boldsymbol{x} \in \Gamma\ ,\quad \phi(\boldsymbol{x}, t) = 0 \tag{10}$$

and the equation of motion of the level-set function is:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{v}\nabla\phi = 0 \tag{11}$$

where $\boldsymbol{v}$ is the desired velocity on the interface. The level-set method has multiple advantages, the main one for our calculations being built-in topological regularization that deals easily with merging and pinching off, and allows robust calculation of geometric properties.

Even though the 0-level-set will be advected with the correct velocity, ϕ will no longer be a distance function and can become irregular after several timesteps. Because the values of the level-set function in the vicinity of the 0-level-set are used to reconstruct a velocity field (see Section 5.4), it hinders this reconstruction process, hence the need to correct the values of the level-set to get $|\nabla\phi| = 1$. One way is to iterate on the following Hamilton-Jacobi equation [60]:

$$\begin{cases} \phi_\tau + sign(\phi^0)\,(|\nabla\phi| - 1) = 0 \\ \phi(x, 0) = \phi^0(x) \end{cases} \tag{12}$$

where τ is a fictitious time and ϕ^0 is the value of ϕ at the beginning of the redistancing process. Numerous methods for reinitialization exist, see [61] for a comparative study or the recent work of Chiodi and Desjardins [62]. We took the method of Min & Gibou [63] with corrections by Min [64], derived from the method of Russo & Smereka [65]. In order to preserve the mass and the position of 0-level-set ϕ_0, the idea is to include the initial interface location in the stencils of the discretized spatial derivatives. We have recalled the procedure which can be naturally extended to 3D in Appendix A.

5.3. Embedded Boundary (cut-cells)

The 0-level-set of the distance function defines the interface Γ between the two phases. This level-set function is used as an input to modify control volumes in a finite-volume manner. In this section, we draw the main lines of the embedded boundary method as defined in [52, 42] and introduce the notations used hereafter. The main idea is to consider a domain Ω with a general boundary Γ and embed this domain in a regular Cartesian grid with Δx the grid spacing and $\boldsymbol{e}_d$ the unit vector in the d direction. The intersection of each cell with Ω gives a collection of irregular cells as shown on Fig. 2. The vertex-centered levelset field can then be used to obtain the volume fractions $\mathcal{V}_i$ defined as

$$|V_i| = \mathcal{V}_i \Delta x^D \tag{13}$$

with $|V_i|$ the volume of a cell and D the dimension of the problem (2 or 3). Similarly, the face fractions $\alpha_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e_d}}$ are defined as

$$|A_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}_d}| = \alpha_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e_d}} \Delta x^{D-1} \tag{14}$$

where $|A_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}_d}|$ is the surface of the face. The details of the calculation of the volume fractions and face fractions can be found on Basilisk's website (http://basilisk.fr/src/embed.h) and is a direct adaptation of the Johansen & Collela's [52]. These fractions give access to collection of piecewise linear segments, whose centroids x_i^Γ and normals $\boldsymbol{n}_i^\Gamma$ are defined as

$$\boldsymbol{x}_i^\Gamma = \frac{1}{|\mathcal{A}_i^\Gamma|} \int_{\mathcal{A}_i^\Gamma} \boldsymbol{x} d\mathcal{A}. \tag{15}$$

$$\boldsymbol{n}_i^\Gamma = \frac{1}{|\mathcal{A}_i^\Gamma|} \int_{\mathcal{A}_i^\Gamma} \boldsymbol{n}_i^\Gamma d\mathcal{A}. \tag{16}$$

which, in turn form the basis for the construction of conservative, high-order discretization operators, especially the divergence operator ∇:

$$\begin{aligned} \nabla \cdot \overrightarrow{F} &\approx \frac{1}{|V_i|} \int_{V_i} \nabla \cdot \overrightarrow{F} dV = \frac{1}{|V_i|} \int_{\partial V_i} \overrightarrow{F} \cdot \boldsymbol{n} dA \\ &\approx \left[\left(\sum_{\pm=+,-} \sum_{d=1}^{D} \pm \alpha_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e_d}} F^d(\boldsymbol{x}_{i\pm\frac{1}{2}\boldsymbol{e_d}}) \right) + \alpha_i^\Gamma \boldsymbol{n}_i^\Gamma \cdot \overrightarrow{F}(\boldsymbol{x}_i^\Gamma) \right] . \end{aligned} \tag{17}$$

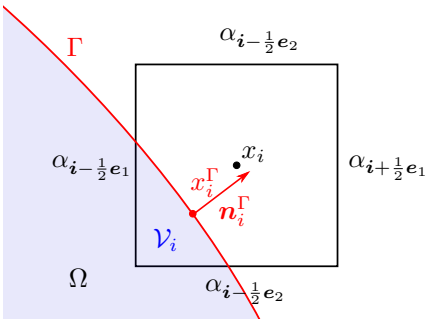


Figure 2: Cut-cell with its initial center of gravity x_i outside the calculation domain Ω

The key point here is that the equations solved on the separate subdomains $\Omega_{\phi^\pm}$ bounded by Γ are sufficiently smooth (since they do not include the interfacial discontinuities) and can therefore be extended to the domain made of the cells that have a non-zero volume fraction, even when the cells have their original center outside of the calculation domain. We depict on Fig. 2 a possible configuration where a cell has its center x_i outside of the domain Ω bounded by Γ. This leads to a conservative, finite-volume methodology which is at least second-order accurate. The main advantage of Cartesian grids embedded boundary over structured or unstructured grid methods is simpler grid generation. The underlying regular grid also allows the use of simpler data structures and numerical methods over a majority of the domain. Accuracy is maintained at the boundaries using an algorithm detailed in [52]. For each partially covered cell or interfacial cell (cells for which $0 < \mathcal{V}_i < 1$), the flux through the boundary, which is the crucial ingredient of Eq. 6, is calculated using only values from other cells. The gradient of a variable a_1 defined only in one phase of the calculation domain Ω_1, phase 1 on Fig. 3, on the embedded boundary $\nabla a_1|_\Gamma$ in the direction of $\boldsymbol{n_\Gamma}$ is calculated as

$$\nabla a_1|_\Gamma = \frac{1}{d_2 - d_1}\left[\frac{d_2}{d_1}(a_\Gamma - a_1^{I_1}) - \frac{d_1}{d_2}(a_\Gamma - a_1^{I_2})\right] \tag{18}$$

where $a_1^{I_{1,2}}$ are quadratically interpolated values of a_1 on each segment $\mathcal{S}_{1,2}$ and a_Γ is the imposed Dirichlet boundary condition on the interface (here the Gibbs–Thomson relation) which is the same for both phases. The stencil used for the interpolation follows the procedure described in [42], depending on the normal of the interface $\boldsymbol{n} = \{n_l, l = 1, \ldots, d\}$, here the dimension $d = 2$, the segments $\mathcal{S}_1$ and $\mathcal{S}_2$ are chosen to be normal to $\boldsymbol{e_k}$ where $\{k : n_k \geq n_l, l = 1, 2\}$.
The calculation of the gradient of $\nabla a_2|_\Gamma$ in Ω_2, is done by using a second variable a_2 defined only in the second domain, volume fractions and face fractions of the second calculation domain can be deduced from the ones previously calculated, they are just the complementary to 1.

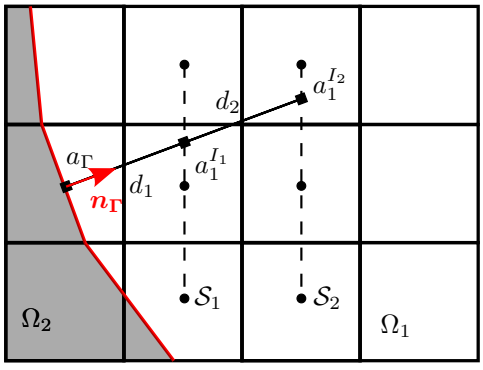


Figure 3: Gradients calculation

The two scalar variables controlling the motion of the interface are the temperature variables denoted T_L, T_S, the temperature fields for the liquid and the solid respectively. For each temperature field, we apply the embedded boundary method and build the associated discretization operators independently using the appropriate volume and face metrics. The interface temperature T_Γ given by Eq. 5 is required to calculate the temperature gradients. We thus need two quantities:

- the curvature κ, calculated using the height function[1] as in Popinet [67],
- the phase change velocity v_{pc}, we assume that we have previously calculated the velocity of the interface and that this velocity remains constant during a timestep,

the details of these calculations will be discussed in Section 5.5. This yields the temperature gradients $\nabla T_L|_\Gamma$ and $\nabla T_S|_\Gamma$ with second-order accuracy. Recalling that the velocity of the interface is defined as the jump in the normal direction of the interface of the gradient of the temperature fields $L_H \boldsymbol{v_{pc}}|_{\boldsymbol{\Gamma}} = [\lambda \nabla T|_\Gamma] \cdot \boldsymbol{n}$, we also expect second-order accuracy on the velocity of the interface.

We would like to highlight here the main difference between our method and the method introduced in [50]. In this article, the authors clearly drop the finite-volume approach taken in [48] in favour of a finite difference approximation for the interface velocity. This approach also requires the local modification of the temperature field to populate values in the vicinity of the interface [68] whereas our method clearly defines two temperature fields and does not necessitate such a modification of the temperature fields. As a result our method guarantees a discretely conservative treatment of heat/mass fluxes.

5.4. Speed reconstruction off the interface

To use the phase change velocity as the velocity in the level-set advection equation, Eq. 11, we have to build a continuous velocity in the vicinity of the interface. In this section we now describe how we rely on our level-set function for this process, starting from the discrete velocity defined only on the interface $v_{pc}|_\Gamma$ that we have previously calculated. We follow the approach of Peng *et al.* [69], and solve an additional PDE so that $\boldsymbol{v_{pc}}$ is constant along a curve normal to $\Gamma(\tau)$

$$\frac{\partial \boldsymbol{v_{pc}}}{\partial \tau} + \delta S(\phi) \boldsymbol{n_\phi} . \nabla \boldsymbol{v_{pc}} = 0 \tag{20}$$

where δ is equal to 0 in interfacial cells and 1 elsewhere, $S(\phi)$ is the sign function and the vector $\boldsymbol{n_\phi} = \frac{\nabla \phi}{|\nabla \phi|}$ is normal to the isovalues of the level-set function. The velocity reconstruction process is initialized by setting $\boldsymbol{v_{pc}}(x, \tau = 0) = \boldsymbol{v_{pc}}|_\Gamma$ in the interfacial cells (blue cells on Fig. 4 with the value of interfacial centroids in red) and 0 elsewhere. We want to highlight here that this field reconstruction step couples the embedded boundary representation of the interface using the interface centroids and the level-set method which gives the normal to the interface $\boldsymbol{n_\phi}$.
The reconstruction is divided in two steps on which we iterate until we reach convergence:

1. A few iterations of Eq. 20 are performed, typically $2D$ with D the dimension of the problem, such that the velocity converges in the vicinity of the interface. Note that the definition of δ ensures that only non-interfacial cell values are updated, the initial value in interfacial cells acts as a source term.
2. The value of the velocity in interfacial cells is modified as

$$\widetilde{\boldsymbol{v_{pc}}} = \boldsymbol{v_{pc}} + \frac{\epsilon}{\alpha_{ij}} \tag{21}$$

where ϵ is the error between the interpolation of $\boldsymbol{v_{pc}}$ on the boundary face centroids:

$$\forall \boldsymbol{x}_{i,\Gamma} \in \text{facets}, \qquad \mathcal{L}(\boldsymbol{v_{pc}}(\boldsymbol{x}_{i,\Gamma})) = \boldsymbol{v_{pc}}|_\Gamma + \epsilon \tag{22}$$

with $\mathcal{L}$ a biquadratic interpolation operator using Lagrange polynomials on a standard 3×3 stencil such that

$$\mathcal{L}(q(\boldsymbol{x})) = \sum_{i,j=-1\ldots1} \alpha_{ij} q_{ij} \tag{23}$$

is the interpolation of a cell-centered field at $\boldsymbol{x}$.

The aim of the second step is to correct the approximation error done at the initialization of this reconstruction process, by setting the value in the cell centers equal to the value in the centroids.

We show on Fig. 5 an application of this reconstruction method, the initial calculation domain is $[-0.5 : 0.5] \times [-0.5 : 0.5]$. The initial interface is a circle of diameter $9/10^{\text{th}}$. The initial grid maximal resolution is

[1] With a level-set function, it would be tempting to use the classical relation for the calculation of the curvature

$$\kappa = \frac{\phi_y^2 \phi_{xx} - 2\phi_x \phi_y \phi_{xy} + \phi_x^2 \phi_{yy}}{\left|\phi_x^2 + \phi_y^2\right|^{3/2}}, \tag{19}$$

preliminary tests showed no difference between the 2 methods. For a more detailed comparison of the influence of curvature on dendritic growth, one may refer to [66].

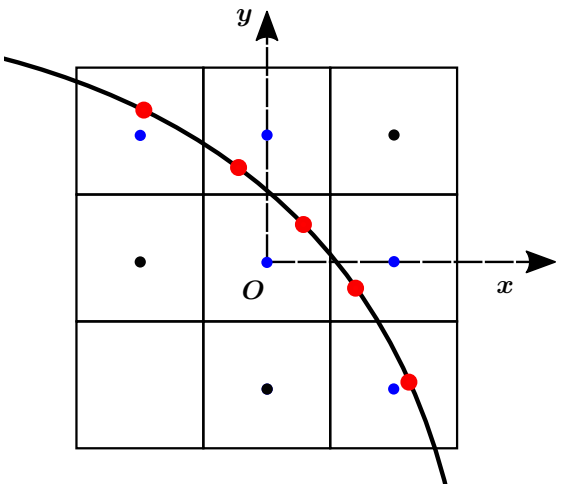


Figure 4: Cell categories for the reconstruction of v_{pc} off the interface. In blue: interfacial cells, in red: interface centroids, in black: non-interfacial cells

128×128. We set $\boldsymbol{v_{pc}}|_\Gamma = -\boldsymbol{n}\kappa$ in interfacial cells, with κ the local curvature. The same test case has been run using the curvature calculated with the height function and the level-set function. The extension method for the velocity is applied with a CFL number of 0.3. The level-set function is then advected with the continuous velocity. At the end of each iteration the level-set function is reinitialized. The interface is output every 60 iterations and remains circular for both methods, as shown on Fig. 5. This demonstrates both the robustness and the accuracy of the method without any additional regularization. We obtain similar results in 3D, as shown in Fig. 6.

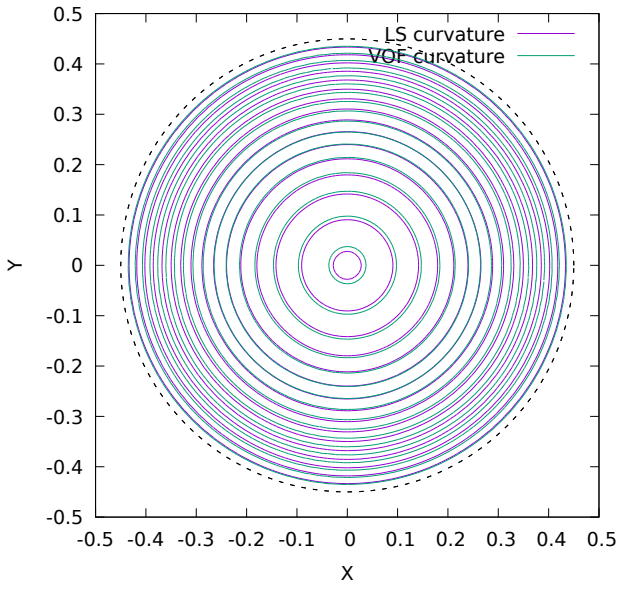
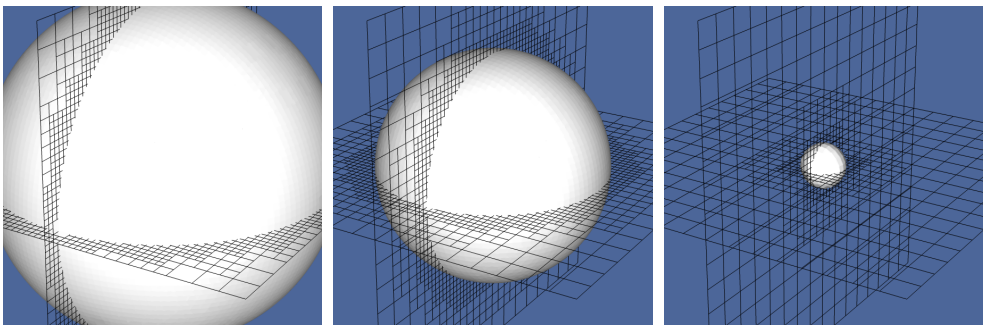


Figure 5: Shrinking circle, dashed line: initial interface

5.5. Embedded boundary motion: timestep constraint, emerging cells scalar field initialization and truncation error variations

In the case of a moving interface Γ, the two considered domains are functions of time $\Omega_+ = \Omega_+(t)$ and $\Omega_- = \Omega_-(t)$. Cells that were non-interfacial can become interfacial. In these cells an initialization technique for the undefined fields is required. We show a typical case of a moving boundary on Fig. 7 where the interface Γ at instant t^{n-1} and instant t^n are displayed with a solid line and a dashed line respectively. The blue cell for which the solid temperature was undefined at instant t^{n-1} becomes an interfacial cell after displacement and the solid temperature field needs to be initialized in this cell.



(a) $t = 0$ (b) $t = 120\Delta t$ (c) $t = 240\Delta t$

Figure 6: Shrinking sphere, $\boldsymbol{v} = \kappa \boldsymbol{n}$ imposed on the interface centroids

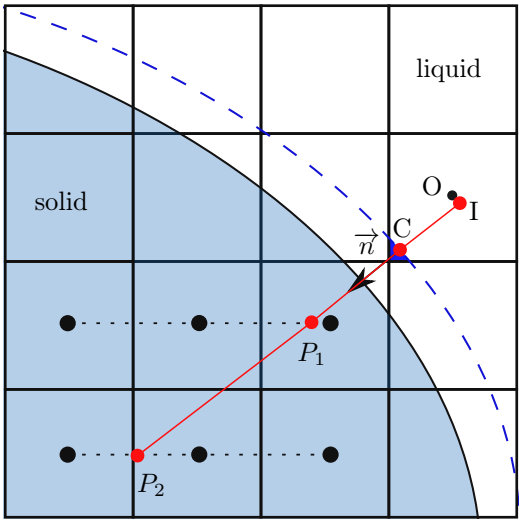


Figure 7: Case of a new interfacial cell, in dark blue. Solid line: t^{n-1} and dashed line: t^n

More generally, we tag new interfacial cells if they verify 2 conditions:

$$\begin{cases} \mathcal{V}_{ij}^{n-1}(1-\mathcal{V}_{ij}^{n-1}) = & 0 \\ \mathcal{V}_{ij}^{n}(1-\mathcal{V}_{ij}^{n}) \neq & 0 \end{cases}.$$

where we denote $\mathcal{V}^{n-1}$ the volume fraction at the previous timestep and $\mathcal{V}^n$ at the current one, this assertion is tested at Step 3 of the global algorithm. In order to initialize the fields of emerging cells, we took an approach similar to what is already done for gradient calculations. We detail here the initialization procedure only for the temperature. The Dirichlet boundary condition on the interface T_Γ^n is obtained from the geometric properties of the interface after advection and an interpolation on the interface of the previously reconstructed phase change velocity field $\mathcal{L}(\boldsymbol{v_{pc}}^{(n)}(\boldsymbol{x}_{i,\Gamma}))$:

$$T_{L,\Gamma}^n = T_{S,\Gamma}^n = T_\Gamma^n = T_m - \epsilon_\kappa \kappa^{(n)} - \epsilon_v \mathcal{L}(\boldsymbol{v_{pc}}^{(n-1)}(\boldsymbol{x}_{i,\Gamma})). \tag{24}$$

The temperature field is interpolated, if possible, at two points along the direction of the normal to the embedded boundary at P_1 and P_2. The temperature at the centroid C is given by the Gibbs–Thomson relation. A quadratic interpolation of the values P_1, P_2, C gives the value at I, the orthogonal projection of O onto the line $(C, \overrightarrow{n})$ where $\overrightarrow{n}$ is the normal to the interface at C. We neglect tangential variations of the temperature. Similar procedure can be devised for other variables (velocity, pressure) according to their boundary conditions on the interface.

Without any timestep constraint, some cells that were completely uncovered might become completely covered. This means that a cell could undergo a complete phase change during one timestep. Therefore, the following constraint is applied:

$$\Delta t < \frac{\Delta x}{|\boldsymbol{v_{pc}}|}. \tag{25}$$

The volume fractions $\mathcal{V}_i$ and face fractions $\alpha_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}_d}$ are considered constant during one timestep. We solve a fixed-boundary problem at each timestep, see Eqs. (18-19) in Schwartz *et al.* [42]. Thus, our numerical scheme

for the displacement is only first-order accurate in time. This is a strong approximation because in diffusion-driven cases, the motion timescale of the interface is comparable to the diffusion timescale. Future work should focus on using a better approximation of the position of the interface during a timestep.

Another issue related to Cartesian cut-cell methods with moving embedded boundaries is the oscillation of fluxes calculated on the boundary due to varying truncation errors. In cut cells, the discretization stencils are offset and can vary abruptly as the interface moves, simply because the interface becomes cut/uncut. Another situation is shown on Fig. 8, the red solid line shows the interface Γ^{n-1} at instant t^{n-1}, and the solid line is Γ^n. As one can see the stencil used for the calculation of the fluxes varies and can introduce spurious oscillations. Therefore, obtaining smoothly varying values of fluxes on the interface is an active field of research [43, 70]. Because these variations of the truncation error interact with the motion of the embedded boundary they can quickly deteriorate the quality of the solution.

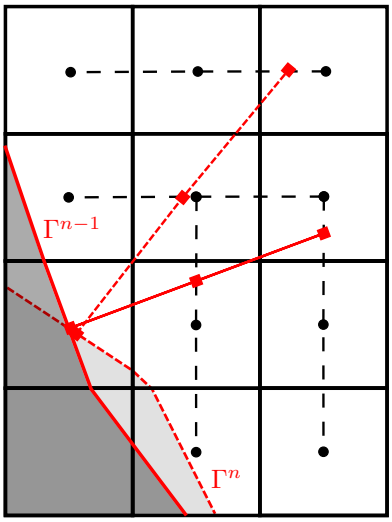


Figure 8: Stencil jump with interface motion, solid red line: Γ^{n-1}, dashed: Γ^n

5.6. Mesh adaptation

The *Basilisk* library has mesh adaptation capabilities. It uses quad/octrees with a 2-1 balancing rule, see [71, 72]. When one cell is refined, projected values onto new cells are typically calculated via a bi/trilinear interpolation of the field on the coarser level. In the presence of an embedded boundary, specific refinement and coarsening functions have been written to take the face and volume fractions into account. We present on Fig. 9 a typical 3×3 stencil where the central cell is cut by the interface Γ. We detail here the case of 3 of the children cells for a scalar field a_1 defined only in the domain Ω_1, we will refer to the sub-cells by the color of their cell center:

- red cell: a standard bilinear interpolation can be applied. The associated stencil for interpolation, denoted by a red rectangle, only contains cells that are either partially covered or fully uncovered,
- green cell: only three cells on the coarser level are accessible, therefore a triangular interpolation is used,
- blue cell: this cell is completely covered, therefore it does not need to be initialized.

The same characterization is done in phase Ω_2 simultaneously for the prolongation of a_2, which gives

- red cell: this cell is partially covered, on the coarser grid, the diagonal cell is completely covered, therefore it is initialized with the value of its parent,
- green cell: triangular interpolation,
- blue cell: bilinear interpolation.

The phase change velocity is used as an adaptation criterion in our simulations in combination with the other "standard" adaptation criteria, namely the temperature, the velocity in the liquid phase and the volume fraction.

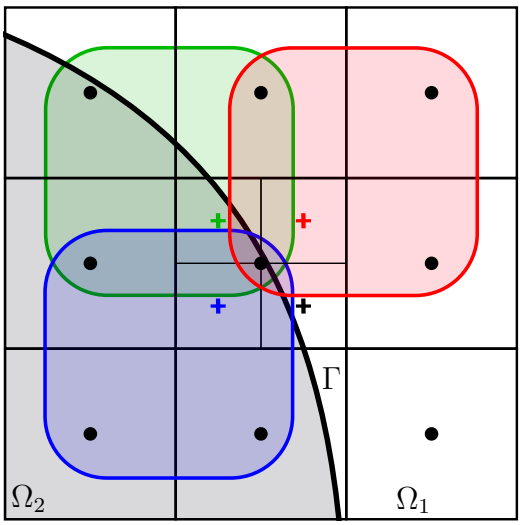


Figure 9: Mesh adaptation with an embedded boundary. Possible configurations for the prolongation operator.

6. Test & validation cases

In this section we present eight different numerical configurations to demonstrate and characterize the ability of our method to obtain accurate solutions, all of code associated for running those simulations is available in A. Limare's sandbox. The first case involves a planar interface and validates the accuracy of the method without the initialization procedure for emerging cells. The second validates the scalar field initialization procedure. The third one tests the stability of the method in 2D and in particular the speed reconstruction method with a standard case known as the Franks's spheres. The fourth one illustrates the ability of our method to capture instabilities and in particular the formation of dendrites; it also checks and quantifies the accuracy of our method. The fifth shows the compatibility of our method with anisotropy in the Gibbs–Thomson condition (cube_sixfold.c). The sixth is exactly the same simulation one but in 3D and shows the formation of dendrites in 3D (crystal_growth3D.c). The seventh one makes a comparison of the tip velocity calculated using our method with a linear solvability theory. The last simulation is taken from [53] and combines two different solvers: in the liquid and solid phase we solve the coupled diffusion equations for the temperature as for the previous cases, but in the liquid, we now solve in addition the Navier-Stokes equations allowing for fluid motion (Favier_Ra-Be.c). In particular, we recover some of the main results of their study, which are the existence of a critical Rayleigh number for the instability and the variation with time of the associated wavelength because of the melting of the solid boundary. Except when it is explicitly stated, we will consider for the example the dimensionless set of equations (Eqs. 7 to 9) considering the fluid at rest ($\mathbf{u} = 0$ everywhere) and taking the thermal ratios unity, $D_L/D_S = \lambda_L/\lambda_S = 1$ (recall that we have already taken $\rho_L = \rho_S = \rho$).

6.1. Solidifying domain

This test case is borrowed from [35]. It is a simple Stefan problem where an initially planar interface translates at constant velocity. The interface is located at $t = 0$ at the position $x = 0$ with the initial temperature field:

$$T_0(x) = \begin{cases} -1 + e^{-V \cdot x}, x > 0 \\ 0, x \leq 0 \end{cases} \tag{26}$$

It is easy to show that the solution of the diffusion equation for the temperature field and the Stefan condition for the interface leads to the translation of the planar interface at constant velocity $V = 1$. Indeed, the temperature field evolves as

$$T(x,t) = \begin{cases} -1 + e^{-V(x-Vt)}, x > Vt \\ 0, x \leq Vt \end{cases} \tag{27}$$

which gives the steadily moving planar interface, whose equation is

$$\Gamma(t) = \{x = Vt, y = s\} \ , \ s \in \Re. \tag{28}$$

We perform an error analysis of our method by studying the error on the initial phase change velocity, see Table 1. It shows a second-order accuracy on the initial temperature gradient jump. A classical error analysis on the final temperature field shows also a second-order accuracy, see Table 2. We show a comparison on Table 3 of the same calculation as run in [35]: our results show the same slope when refining the grid but an error that

is 2 orders of magnitude higher. However, these results have to be taken carefully. Indeed, in their original article [35] have shown very low level of errors for a purely level-set based method. In contrast, Udaykumar *et al.* [47] use a sharp interface method show error levels (Fig. 17 in [47]) that are on par with our results.
We have a slight drop in the order of accuracy for low resolution. Note that we do not really validate the accuracy of the initialization procedure of the temperature in the solid here, because the newly solid cells only need to be initialized with $T = 0$. Finally we did an error analysis with a fixed timestep $\Delta t = 1 \times 10^{-6}$ and a fixed number of 400 iterations: the results are presented in Table 4 and also show second-order asymptotic accuracy.

Grid	L_1-error	order
32^2	3.18e-04	–
64^2	8.04e-05	1.98
128^2	2.02e-05	1.99
256^2	5.07e-06	2

Table 1: Convergence of the initial temperature gradient when refining grid size

Grid	Timestep	L_1-error	order	L_∞-error	order
32^2	1.6×10^{-3}	1.59e-4	–	5.31e-4	–
64^2	4×10^{-4}	6.52e-05	1.28	0.000252	1.07
128^2	1×10^{-4}	1.55e-05	2.07	6.46e-05	1.96
256^2	2.5×10^{-5}	4.06e-06	1.93	1.63e-05	1.99

Table 2: Convergence of the temperature field when refining grid size and time step

Grid	Timestep	L_1-error	order
80^2	$1. \times 10^{-4}$	5.36e-7	–
160^2	2.5×10^{-5}	1.34e-7	1.996
320^2	6.25×10^{-6}	3.36e-8	1.999

Table 3: Convergence of the initial temperature gradient when refining grid size and time step, taken from [35]

Grid	L_1-error	order	L_∞-error	order
32^2	1.51e-05	–	1.67e-4	–
64^2	5.52e-06	1.45	8.78e-05	0.92
128^2	1.4e-06	1.97	2.28e-05	1.94
256^2	3.32e-07	2.07	4.73e-06	2.26

Table 4: Convergence of the temperature field with grid refinement, fixed timestep = 10^{-6}

Grid	L_1-error	order
80^2	1.32×10^{-3}	–
160^2	2.63×10^{-4}	2.33
320^2	3.67×10^{-5}	2.84
640^2	3.75×10^{-6}	3.29

Table 5: Convergence of the temperature field with grid refinement, fixed timestep = 10^{-6}, results from [35]

6.2. Planar interface with an expanding liquid domain

This case tests the diffusion of two tracers separated by an embedded boundary (taken from [73, 47]). It corresponds to the melting of an ice layer by imposing a warm temperature condition T_1 at the top boundary and the melting one T_m at the bottom. The Stefan number $St = \frac{C(T_1 - T_m}{L_H}$ for our simulation is 2.85 and the dimensionless temperature ($(T - T_m)/(T_1 - T_m)$ still denoted T in the dimensionless equation) is 1 at the top ($y = 1$) and 0 at the bottom ($y = 0$) as shown in Fig. 10. The initial temperature in the liquid is

$$T_L(x, y, t_0) = 1 - \frac{\mathrm{erf}(\frac{1-y}{2\sqrt{t_0}})}{\mathrm{erf}(\lambda)}$$

where $\lambda = 0.9$. The temperature in the solid is $T_S = 0$. With these initial conditions, the interface position as a function of time is given by

$$y(t) = 1 - 2\lambda\sqrt(t). \tag{29}$$

We start the simulation at $t = t_0 = 0.03$ such that there are at least two full cells above the interface in order to have a correct approximation of the gradients for Eq. 18, in the liquid phase. Notice that the initialization method of the temperature field in newly liquid cells is thus tested for this set up. Error plots in Table 6 shows convergence of the $L_1 - norm$ with an order of accuracy slightly above 2. The results on the L_∞-error also show the expected order of accuracy for low resolution and a drop at 256×256 which requires further investigation. We compared our results on the same simulation in [47]. We note that in [47] Figs. 14-16, there is a slight difference with our calculations. Indeed, in [47] simulations are run while either imposing directly the position of the interface or the temperature field. Notice that even though those theoretical values are imposed in their simulations and not in ours, we obtain similar error levels.

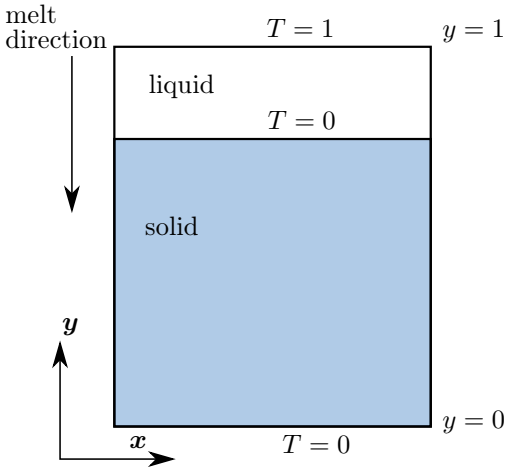


Figure 10: Scheme of the configuration studied

Grid	Timestep	L_1-error	order	L_∞-error	order
32^2	1.e-2	1.97e-03	–	4.86e-03	–
64^2	2.5e-3	3.80e-04	2.37	6.40e-04	2.93
128^2	6.25e-4	8.31e-05	2.19	1.41e-04	2.17
256^2	1.56e-4	2.00e-05	2.05	6.06e-05	1.23

Table 6: Convergence of the temperature field, melting solid

6.3. Frank's Spheres

Frank's spheres correspond to the growth of an ice sphere in an undercooled liquid. The theory of this test case has been studied originally by Frank [74], and it is a crucial test of the numerical stability of the scheme. Indeed, in the absence of anisotropy, a growing sphere (whatever the space dimension $D = 2$ or 3 in practice) is an exact solution of the dynamics, although it is unstable due to the well known Mullins-Sekerka instability. The next cases in this paper focus on the simulation of dendritic growth. In the simulations, we want to stress that because of the numerical noise, the sphere destabilizes and forms dendrites: the numerical robustness of the scheme can be thus tested by investigating how the ice domain diverges from the sphere. Therefore, starting with a spherical initial interface (circle in 2D or sphere in 3D) containing a solid seed surrounded by an undercooled liquid, we will test the stability of the numerical scheme by inspecting the sphere growth. In fact, a class of self-similar solutions has been developed by Frank, in one, two and three dimensions, for which the sphere growth follows a square-root-of-time evolution characterized by the number S

$$R(t) = St^{1/2} \tag{30}$$

The solutions of the problem can be parametrized using the self similar variable $s = r/R(t)$. and the corresponding dimensionless temperature field is 0 for $s < S$ (corresponding to $r < R(t)$ and using for simplicity the same notation T for the temperature field and its self-similar function):

$$T(r,t) = T(s) = T_\infty \left(1 - \frac{F_D(s)}{F_D(S)}\right) \tag{31}$$

if $s > S$. The functions F_D are solutions of the equations and for $D = 2$ we have:

$$F_2(s) = E_1\left(\frac{s^2}{4}\right) \tag{32}$$

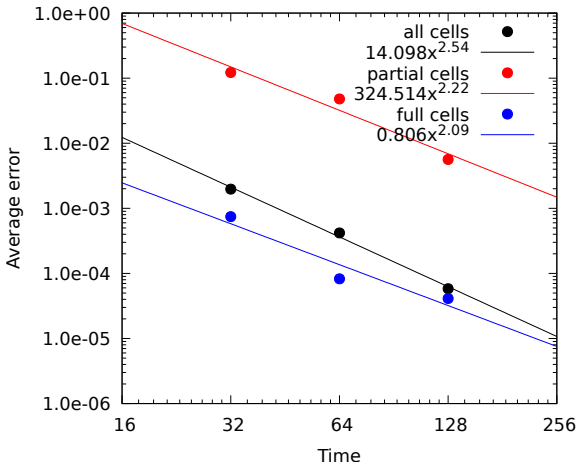
where

$$E_1(x) = \int_1^\infty \frac{e^{-xt}}{t}dt = \int_x^\infty \frac{e^{-t}}{t}dt \tag{33}$$

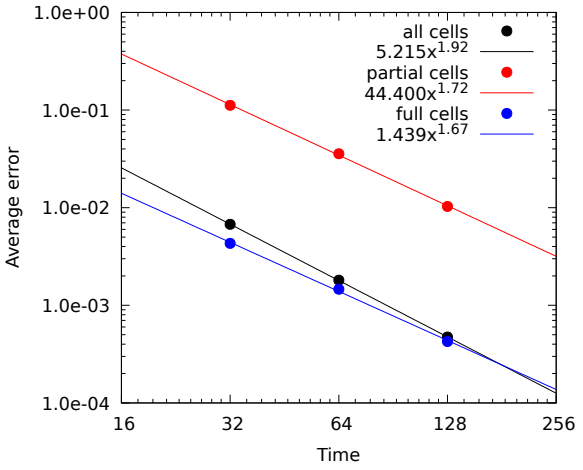
Calculations are performed with the parameters of Almgren [75]:

$$T_\infty = -0.5 = \frac{1}{2}S\frac{F_2(S)}{F_2'(S)} \tag{34}$$

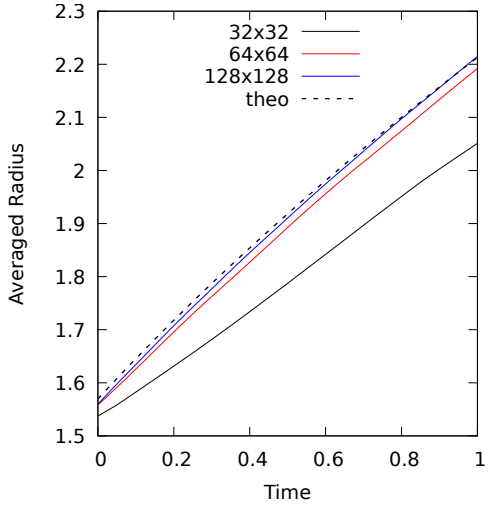
which gives a value of $S = 1.56$. Results of the calculation are shown on Fig. 11 the initial time is $t_0 = 1$. Fig. 11a shows results for calculation after 100 iterations with fixed timestep $\Delta t = 10^{-4}$, we recover a second-order convergence for the final temperature field. Calculations have also been performed for varying mesh size and timestep where $\Delta t = 0.2(\Delta x)^2$. The order of accuracy is between 1.5 and 2.



(a) Frank's Spheres convergence results, fixed timestep $\Delta t = 10^{-4}$.



(b) Frank's Spheres convergence results varying timestep and mesh size.



(c) Comparison of simulated radii with different grid size and theoretical prediction.

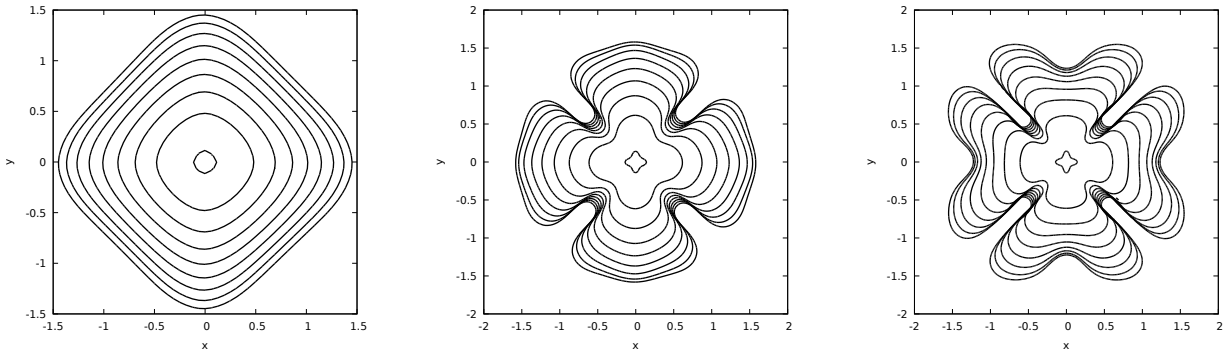
Figure 11: Results for Frank's spheres.

6.4. Crystal growth

As discussed above, crystal formation is physically unstable and leads to dendritic growth [56]. In order to study the formation of dendrites we consider an ice crystal growing in an undercooled liquid as in Chen *et al.* [35]. In the absence of liquid flow, this configuration consists in the diffusion of two tracers separated by a complex embedded boundary. The dimensionless computational domain is $\Omega = [-2 : 2] \times [-2 : 2]$, the initial

interface (0-level-set) is defined by: $\Gamma(r,\theta) = (r,\theta)/r^2(1-0.3\cos(4\theta)) - \dfrac{1}{15}$ where θ is the angle of the outward normal with the x axis. The interface moves according to the Stefan relation with St $= 0.5$. The boundaries are thermally isolated, therefore we should reach a steady state around the time 0.8 with about half the domain that is solid.

The ice particle is initially at $T_S = 0$ and the temperature in the liquid is $T_L = -0.5$. The temperature on the interface follows the Gibbs–Thomson relation, taking $\epsilon_\kappa = \epsilon_v = 2 \times 10^{-3}$. The interface is plotted every 0.1 unit time unit on Fig. 12 for three different grid sizes, showing an instability which depends on the grid size: in fact the instability generates high-curvature unstable regions that are eventually stabilized by the Gibbs–Thomson contribution in the melting temperature. Therefore, the smaller the mesh size, the faster and the more complex the instability grows. The length of the dendrites can be directly calculated and is fixed by the value of ϵ_κ. Our results are in fact quite comparable with [35, 76], but the onset of instabilities can be seen on the 256^2 case far earlier than in their simulations, indicating a very low level of built-in regularization in our method.



(a) 64^2 (b) 128^2 (c) 256^2

Figure 12: Influence of the spatial resolution on the initial dendritic growth. As the resolution increases, shorter wavelengths are resolved leading to an earlier development of the instability.

6.5. Crystal growth with sixfold anisotropy

It is well known that to describe accurately dendritic growth in crystals, anisotropy of the Gibbs–Thomson condition on the curvature needs to be implemented, taking for instance the form [76], in 2D:

$$\epsilon_\kappa = 0.001\left(1 + \epsilon\left[\frac{8}{3}\sin^4(3(\theta - \frac{\pi}{2})) - 1\right]\right). \tag{35}$$

We will use $\epsilon = 0.4$, $\theta = (Ox, \boldsymbol{n})$ in our simulations. We expect to have 6 primary dendrites growing at the same speed, details of the calculation are given in Table 7.

Undercooling	Domain size	ϵ_v
0.8	$[-2:2] \times [-2:2]$	0.001

Table 7: Details of the calculation

We show results of the calculations on Fig. 13, where the interface is plotted every $\Delta t = 3 \times 10^{-3}$ and the final time of the calculation is $t = 3.6 \times 10^{-2}$. Since we use adaptive mesh refinement, the maximum equivalent resolution is 512^2. We plot a circle of radius 1.27 with a dashed line, which is a simple fit to compare the size of the dendrites. At the final time, the size of the six main dendrites is very similar which is a strong indication of the correct treatment, even in non-grid-aligned directions, of the anisotropic Gibbs–Thomson condition. We notice also the presence of secondary dendrites, as expected from crystal growth instability.

6.6. Crystal growth in 3D

The details of the calculation are similar to those of the previous calculation, using now a 3D Gibbs–Thomson condition, following:

$$\epsilon_\kappa = \overline{\epsilon_\kappa}\left[1 - 3\epsilon_4 + 4\epsilon_4^2\left(\sum_{i=1}^{3} n_i^4\right)\right] \tag{36}$$

with $\epsilon_4 = 0.4$, $\overline{\epsilon_\kappa} = 0.001$, the initial undercooling is 0.8 as in the previous calculation. Here the results are shown for a grid with an equivalent resolution of $256^3 \approx 16 \times 10^6$, the initial number of cells is 4×10^4, the final

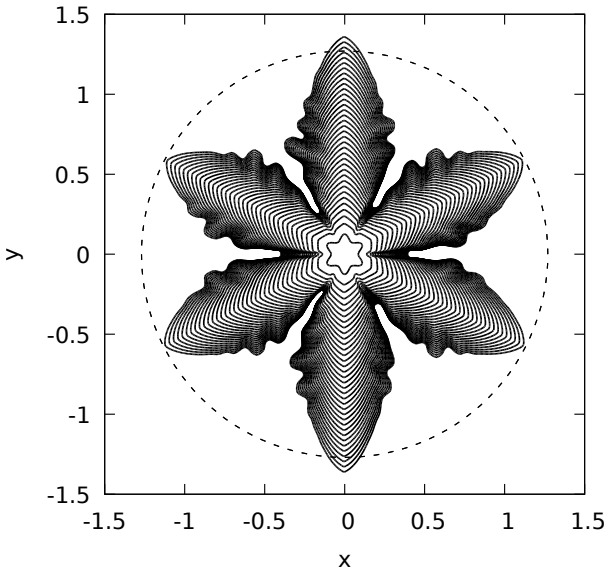


Figure 13: Anisotropy effect with a sixfold symmetry

number of cells is about 2×10^6 and the final time is 3.6×10^{-2}. This variation of the number of cells with time gives a speed-up thanks to the AMR of about 10. The value $\epsilon_4 = 0.4$ introduces a strong anisotropy on the interface temperature. Fig. 14 represents the interface at the end of the calculation and a slice of the mesh in a medial plane. Results are quite similar to the simulations by Lin *et al.* [77]. One can see the expected fourfold periodicity of the main dendrites. The secondary dendrites are also quite well captured by our method demonstrating both its robustness regarding anisotropy and its accuracy. We point out the strength of the mesh adaptation method allowing very local mesh refinement. Future simulations will focus on having a locally converged state with regards to the mesh adaptation criteria. The main driving adaptation criterion is linked to the thermal boundary layer formed near the interface. This is a case where the AMR capabilities of *basilisk* are put to good use, because the complexity of this case is linked to the dimensionality of the solid/liquid interface which is between 2 and 3. Therefore if we compare a simulation using adaptive grids and standard Cartesian one, because the dimensionality of the problem at hand is lover than 3, the bigger the simulation, the better the speed-up thanks to AMR[78].

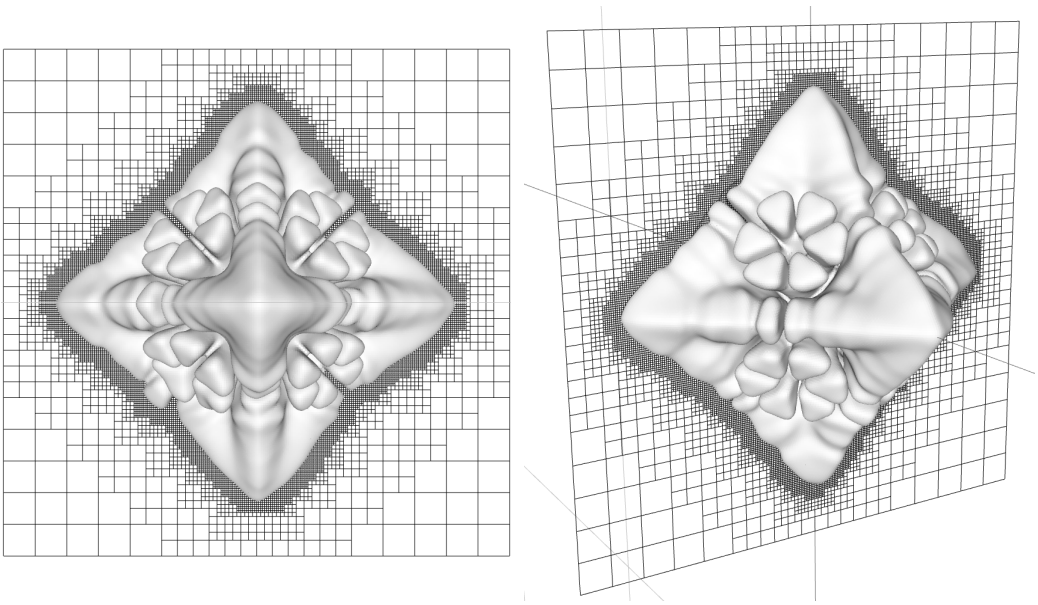


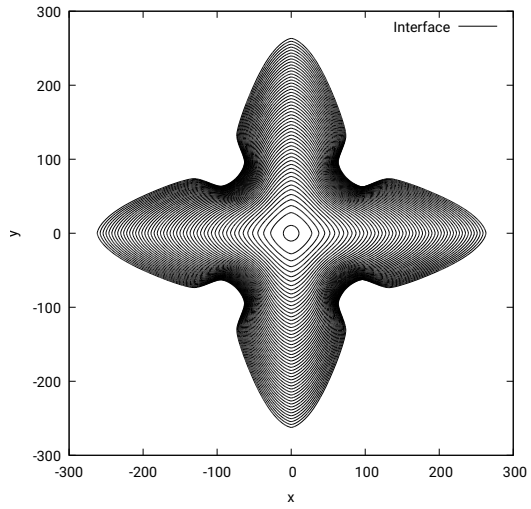
Figure 14: Crystal growth in 3D

6.7. Comparison of the tip velocity with linear solvability theory

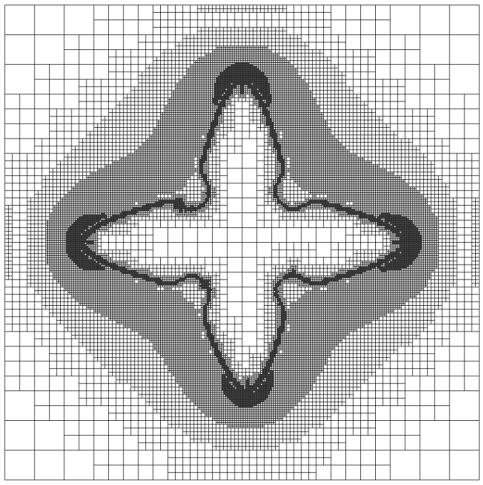
This case models anisotropy effects and can be compared with predictions of microscopic solvability theory [76, 79]. Here the Gibbs–Thomson condition is:

$$T_\Gamma = -\overline{\epsilon_\kappa}(1 - 15\epsilon\cos 4\theta)\kappa \tag{37}$$

with $\overline{\epsilon_\kappa} = 0.5$, $\epsilon = 0.05$, the initial undercooling is 0.55 and every other parameter is unity. The mesh has an equivalent resolution of 512^2 and the computational domain is $[-400 : 400] \times [-400 : 400]$. The results are plotted for the adimensionalized field $\tilde{V}$, $\tilde{x}$, and $\tilde{t}$ see Fig. 16. A difficulty associated with this test case is the initialization of the temperature in the newly solid cells which depends on the temperature of the interface and therefore its velocity with the Gibbs–Thomson condition. As expected, small oscillations in the temperature gradients influence the motion of the interface and create oscillations in the velocity of the interface, Fig. 16b. The expected tip velocity is 1.7×10^{-3}, in our simulation, the velocity reaches the value 1.9×10^{-3} then drifts slowly, as shown by the regression coefficients in the Fig. 16b. The global drift is probably due to the influence of the boundary. The local oscillations are linked to i)the jumps in discretization stencil and the associated truncation errors and ii)the fact that we solve fixed-mesh problems at each timestep, something that could be fixed by reformulating our method using a Arbitrary Lagrangian Eulerian framework.



(a) Interface output at different instants during the simulation



(b) Mesh at final time

Figure 15: Linear solvability test case, interface during calculation

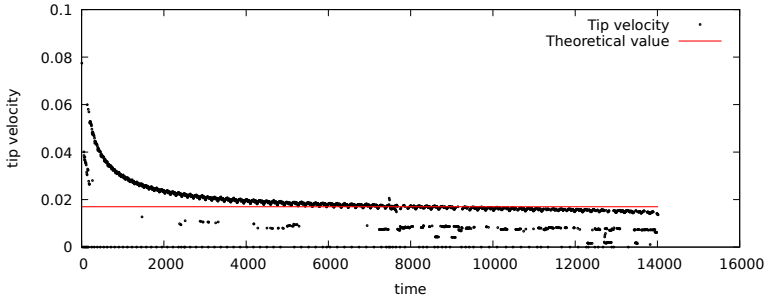
6.8. Rayleigh–Bénard instability with a moving melting boundary

Finally, in order to validate the coupling of our new method for solving solidification fronts with the Navier-Stokes equation, we study the threshold of the Rayleigh–Bénard instability of a melting ice layer, following the recent study of Favier *et al.* [53]. The configuration studied is depicted on Fig. 17. A pure and incompressible material under the influence of gravity $\boldsymbol{g} = -g\boldsymbol{e_Z}$ is comprised between two walls such that it is heated from below (by imposing a temperature T_1 at $z = 0$) and cooled from above (T_0 at $z = H$). The melting temperature varies between these two imposed temperatures $T_0 < T_m < T_1$, so that, taking $\lambda_L = \lambda_S$, the equilibrium position of the ice layer is simply determined by the balance of the thermal fluxes, giving:

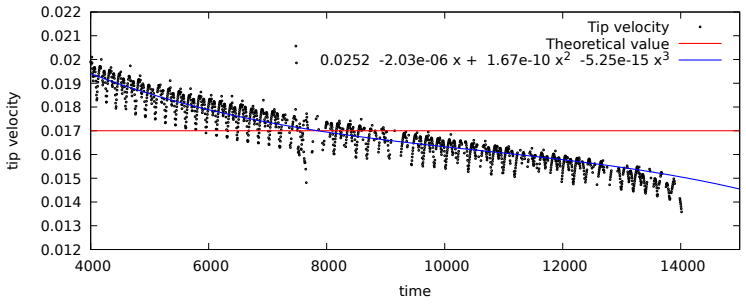
$$h_m = H\frac{T_m - T_0}{T_1 - T_0}.$$

This configuration is similar to the classical one for Rayleigh–Benard (R-B)instabilities, but this time the upper boundary of the liquid can move through melting or freezing and thus initiate the R-B instability during the dynamics. Depending on the physical parameters and the initial conditions, different stationary regimes have been observed in numerical simulations using phase-field modeling for phase change [80].

We thus consider a two-dimensional system bounded by two horizontal walls as shown in Fig. 17 separated by a distance H with periodic boundary conditions on the horizontal direction with an aspect ratio β between the horizontal and vertical dimensions. We apply no slip boundary conditions on the upper and lower boundaries and also on the interface between the two phases Γ. We perform the simulation using the Boussinesq approximation, where the variation of the liquid density with the temperature is taken into account in the buoyancy force only. More precisely, we assume that the density $\rho_S = \rho_L = \rho$ and the thermal diffusivity $D_S(T) = D_L(T) = D_0$



(a) Whole simiulation



(b) Zoom. Oscillations in the interface velocity

Figure 16: Tip velocity as a function of time

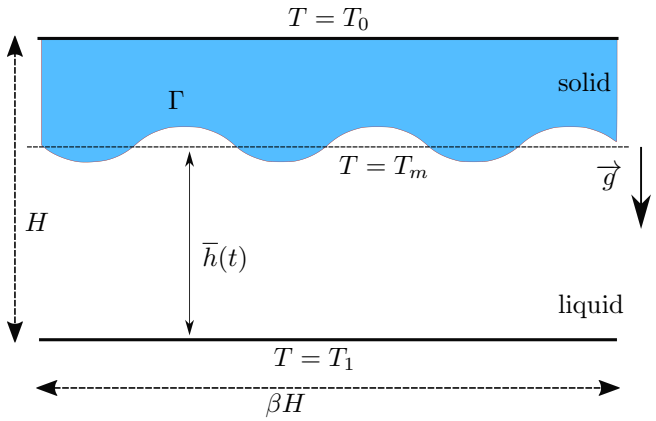


Figure 17: Scheme of the Rayleigh-Benard instability with a melting boundary

are constant and equal in both domains, as well as the fluid viscosity. The dimensionless set of equations thus reads in the fluid domain:

$$\frac{1}{\sigma}\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = -\nabla P + Ra\,\theta\,\boldsymbol{e_y} + \nabla^2\boldsymbol{u} \tag{38}$$

$$\frac{\partial \theta}{\partial t} + \boldsymbol{u}\cdot\nabla\theta = \nabla^2\theta \tag{39}$$

where $\theta = \dfrac{T - T_0}{T_1 - T_0}$ is the dimensionless reduced temperature, P is the dimensionless pressure, Ra and σ are the Rayleigh and Prandtl numbers respectively:

$$Ra = \frac{g\alpha(T_1 - T_0)H^3}{\nu D_0} \text{ and } \sigma = \frac{\nu}{D_0}\ . \tag{40}$$

where $\alpha = 1$ is the thermal expansion coefficient. As in [53], we impose $\sigma = 1$ and only the Rayleigh number is varied for all of our simulations. In the solid phase we have:

$$\frac{\partial \theta}{\partial t} = \nabla^2\theta\ . \tag{41}$$

The Stefan condition is applied on the boundary, and the temperature on the interface is supposed to be constant $\theta_\Gamma = \theta_m$. We apply our method with two different solvers, one for the Navier–Stokes equations in the liquid and a simple diffusion solver in the solid.

We re-perform the calculations of Favier to study the onset of the R-B instability and the formation of convection cells. We define similarly the effective Rayleigh number defined using the fluid layer thickness following:

$$Ra_e = Ra(1-\theta_m)\overline{h}^3 \tag{42}$$

where the averaged fluid height $h(t)$ is defined as

$$\overline{h}(t) = \frac{1}{\beta}\int_0^\beta h(x,t)dx. \tag{43}$$

Convection cells are expected to appear once the simulations reach a critical effective Rayleigh number $Ra_c = 1707.76$. Details of the initial grids are given in Table 8: the initial effective Rayleigh number Ra_e is much lower than the critical Rayleigh number such that the calculations always start with a diffusion-driven dynamic. Results of the average height during the calculations are given in Fig. 18. For each curve, we added a triangle sign, to position when the simulation reaches the critical Rayleigh number. This collection of triangles clearly separates two regimes, the diffusion-driven phase from the convection-influenced one. Once the apparent critical Rayleigh number 1707.76 is reached, the thermal exchange between the bottom boundary and the interface is greatly enhanced and the interface melts much faster.

N_x	N_y	Ra	St	θ_m	β	h_0
512	64	$[10^3 : 10^6]$	10	0.3	8	0.05

Table 8: Numerical set up for the study of the critical Rayleigh number Ra_c

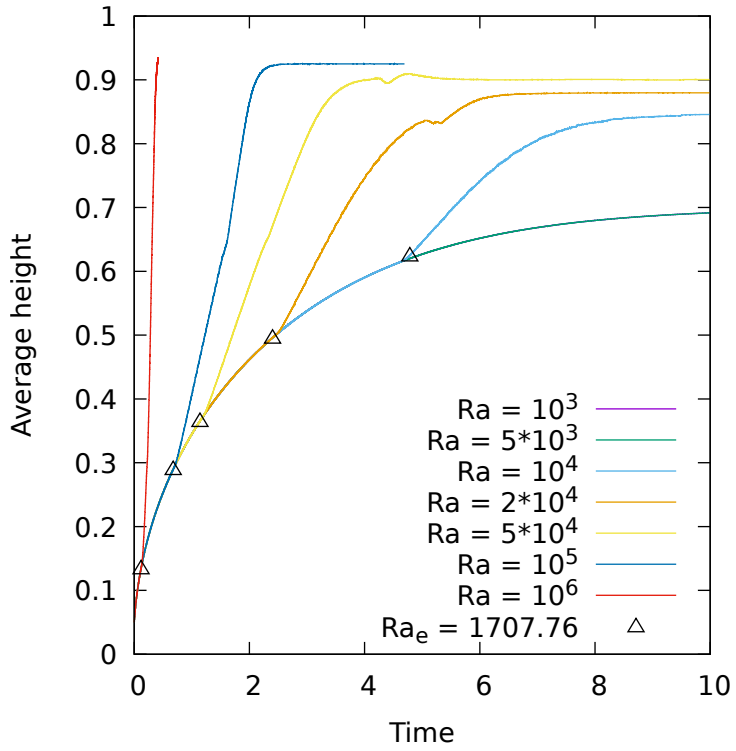


Figure 18: Rayleigh–Bénard instability with a melting boundary. Average height evolution for different Rayleigh numbers

For sufficiently high Rayleigh numbers, the characteristic size of the convection cells of this flow will vary with a secondary bifurcation mechanism. Secondary bifurcations also occured for $Ra = 10^5$ and 10^6 in the previous calculations. These bifurcations occur once the averaged height $\overline{h}$ is equal to the characteristic wavelength of the convection rolls. In the cases where the Rayleigh number is about 10^5–10^6 once the secondary bifurcation is reached, these convection cells have a sufficient time to merge and re-stabilize because the motion of the melting boundary is sufficiently slow. We also performed a simulation with a higher Rayleigh number $\mathrm{Ra} = 10^8$, details of the calculation are given in Table 9. In the $\mathrm{Ra} = 10^8$ case, convection cells never fully stabilize giving birth to many unsteady thermal plumes as shown in Fig. 19.

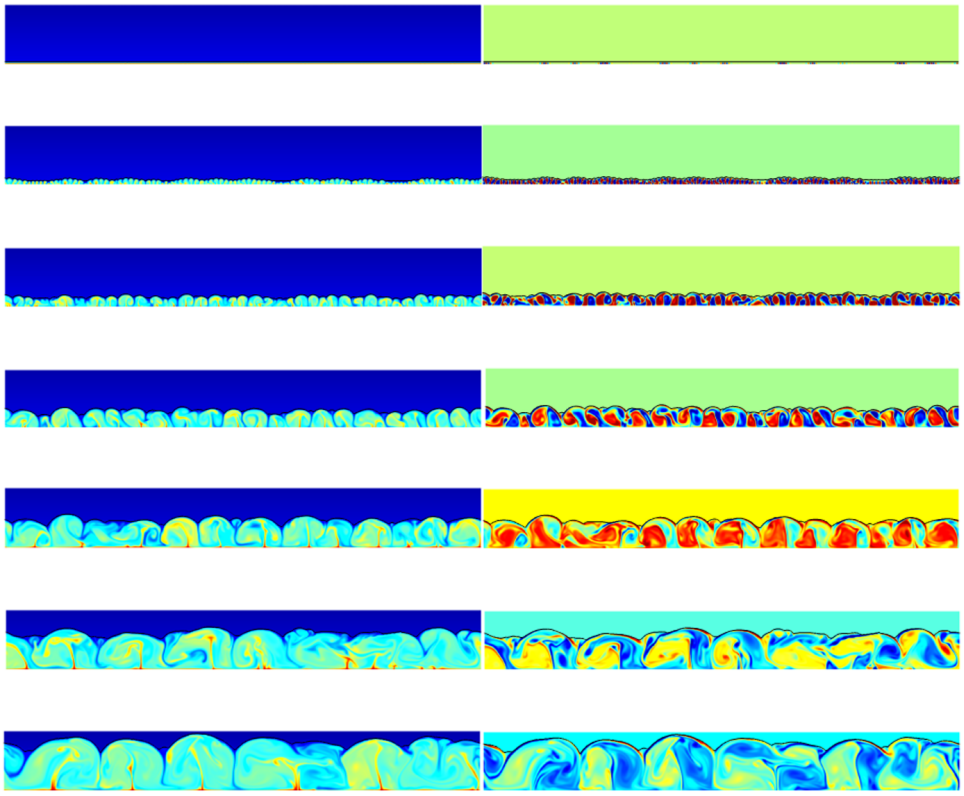


Figure 19: $Ra = 1 \times 10^8$ case, at instant: $t = 0.0005$, $t = 0.0011$, $t = 0.0016$, $t = 0.0023$, $t = 0.0033$, $t = 0.0044$ and $t = 0.0059$. Colored by left: temperature, right: vorticity.

N_x	Ra	St	θ_M	α	h_0
2048	10^8	1	0.05	8	0.02

Table 9: Parameters for the high Rayleigh configuration

7. Conclusion

An original level-set embedded boundary hybrid method has been developed for the simulation of liquid-solid phase change. Its key features are (a) the use of finite-volume conservative operators for embedding the interface which is then seen as a boundary condition for independent sub-problems, (b) the associated second-order accuracy on the gradients on the boundary, (c) a simple velocity extension method, (d) an initialization method derived from the embedded boundary method. The method has been validated on numerous classical melting/solidification problems, from planar melting to dendritic growth and an extension of Rayleigh-Bénard problem in the presence of phase change. The method exhibits spatial and time convergence orders ranging between 1.5 and 2. It has been validated in two and three space dimensions and the use of adaptive mesh refinement allows a precise description of dendritic growth for instance. Future work will develop the coupling of this method with a Volume of Fluid description of fluid/fluid interfaces to allow three phases (gas–liquid–solid) simulations, with a special attention to contact line dynamics. These developments will allow the combination of the accurate, sharp interface dynamics algorithms already implemented in Basilisk with our new conservative, second-order phase change model, which will open a broad range of potential applications.

Appendix A. Redistancing method of Min & Gibou

The main idea behind this method derived from Russo & Smereka [65] is to have a subcell-accurate method in interfacial cells and a simple spatial discretization operator elsewhere, for instance an Essentially Non-Oscillatory (ENO) scheme

$$D_x^+\phi_{ijk} = \frac{\phi_{i+1jk} - \phi_{ijk}}{\Delta x} - \frac{\Delta}{2}\,\mathrm{minmod}(D_{xx}\phi_{ijk}, D_{xx}\phi_{i+1jk}) \tag{A.1}$$

$$D_x^-\phi_{ijk} = \frac{\phi_{ijk} - \phi_{i-1jk}}{\Delta x} - \frac{\Delta}{2}\,\mathrm{minmod}(D_{xx}\phi_{ijk}, D_{xx}\phi_{i-1jk}) \tag{A.2}$$

where

$$D_{xx}\phi_{ijk} = \frac{\phi_{i-1jk} - 2\phi_{ijk} + \phi_{i+1jk}}{\Delta x^2}$$

and

$$\text{minmod}(\alpha,\beta) = \begin{array}{l}\text{if } (\alpha\beta>0)\begin{cases}\text{if } |\alpha|<|\beta| \ , \ \alpha\\ \text{else } \beta\end{cases}\\ \text{else } 0\end{array}$$

we then define a Hamiltonian H_G such that:

$$H_G(\boldsymbol{a},\boldsymbol{b}) = \begin{cases}\sqrt{\sum_{i=1}^{d}\max((a_i^-)^2,(b_i^+)^2)} & \text{if } sgn(\phi^0)\geq 0\\ \sqrt{\sum_{i=1}^{d}\max((a_i^+)^2,(b_i^-)^2)} & \text{if } sgn(\phi^0)<0\end{cases} \tag{A.3}$$

where d is the number of dimensions of the problem considered, $a_i^+ = \max(a_i,0)$, $a_i^- = \min(a_i,0)$ and $\boldsymbol{a},\boldsymbol{b}$ are vectors such that

$$\boldsymbol{a} = (a_i) = (D_i^+\phi) \ , \ i=\{x,y,z\} \tag{A.4}$$
$$\boldsymbol{b} = (b_i) = (D_i^-\phi) \ , \ i=\{x,y,z\}. \tag{A.5}$$

Thus, Eq. 12 becomes:

$$\phi_\tau + sign(\phi^0)[H_G(a,b)] = 0 \tag{A.6}$$

the ENO scheme Eqs. A.1 and A.2 is modified in cells where the interface is located to limit the displacement of the 0-level-set. A quadratic ENO polynomial interpolation gives:

$$D_x^+\phi_{ijk} = \frac{0-\phi_{ijk}}{\Delta x^+} - \frac{\Delta x^+}{2}\,\text{minmod}(D_{xx}\phi_{ijk}, D_{xx}\phi_{i+1jk}) \tag{A.7}$$

and

$$\Delta x^+ = \begin{cases}\Delta x\ \left(\dfrac{\phi^0_{i,j}-\phi^0_{i+1jk}-sgn(\phi^0_{ijk}-\phi^0_{i+1jk})\sqrt{D}}{\phi^0_{xx}}\right) & \text{if } \left|\phi^0_{xx}\right|>\epsilon\\ \Delta x\ \dfrac{\phi^0_{ijk}}{\phi^0_{ijk}-\phi^0_{i+1jk}} & \text{else.}\end{cases} \tag{A.8}$$

with

$$\begin{aligned}\phi^0_{xx} &= \text{minmod}(\phi^0_{i-1jk}-2\phi^0_{ijk}+\phi^0_{i+1jk} \ , \ \phi^0_{ijk}-2\phi^0_{i+1jk}+\phi^0_{i+2jk})\\ D &= \left(\phi^0_{xx}/2-\phi^0_{ijk}-\phi_{i+1jk}\right)^2 - 4\phi^0_{ijk}\phi^0_{i+1jk}\end{aligned}$$

$D_x^-\phi_{ijk}$ is modified in a similar fashion, see [64] for details. One should note that for a smooth interface, without kinks, this method is third-order accurate, whereas in the interfacial cells the order of accuracy is reduced and falls between 1 and 2. We validated our method with a 3D case adapted from [65] of a perturbed distance field to an ellispoid:

$$\phi(x,y,z,t=0) = f(x,y,z)\times g(x,y,z) \tag{A.9}$$

with g real distance, and f a perturbation function such that:

$$f(x,y) = \epsilon + (x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2 \tag{A.10}$$

with $x_0=3.5$, $y_0=2.$, $z_0=1.$, $\epsilon=0.1$. Results are shown on Figs. A.20 to A.23 where we plot the isosurface a certain level-set with a slice view of the distance function before and after the reinitialization. We show here that we have extended Min's method to 3D calculations[2] and obtain an order of accuracy of 2 on Fig. A.24.

For the discretization in time we use the TVD RK3 of [81]. In Min [64], the author demonstrated that the fastest and most accurate method is to use a Gauss-Seidel iteration with a fast-sweeping method [82]. The raster-scan visiting algorithm associated (loops going from N_x to 1) would require specific cache construction to work with our *foreach()* iterators on adaptive grids which would probably compensate the gains associated with this method.

[2]The associated code for redistanciation is available at: LS_reinit.h and the associated ellipsoid redistanciation can be found at: distanceToEllispoid.c.

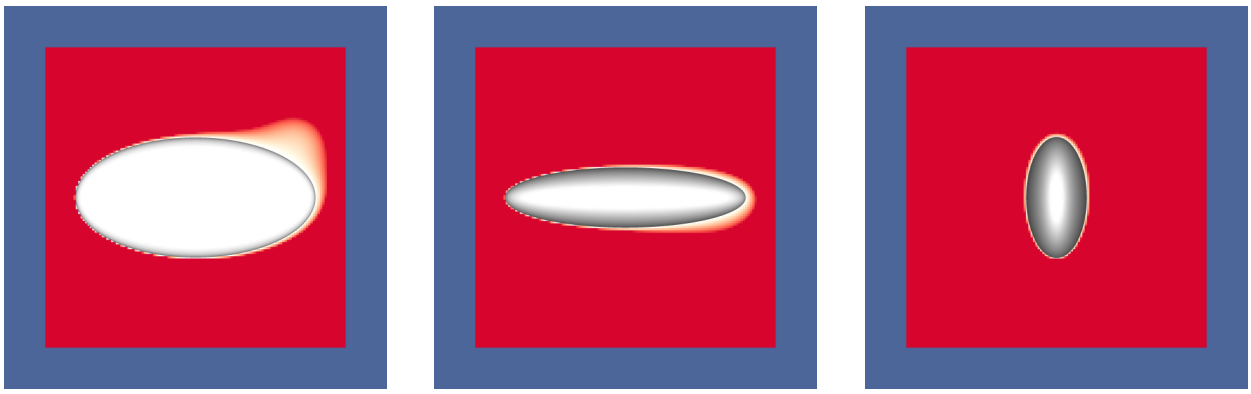


Figure A.20: Initial value of the 0-level-set

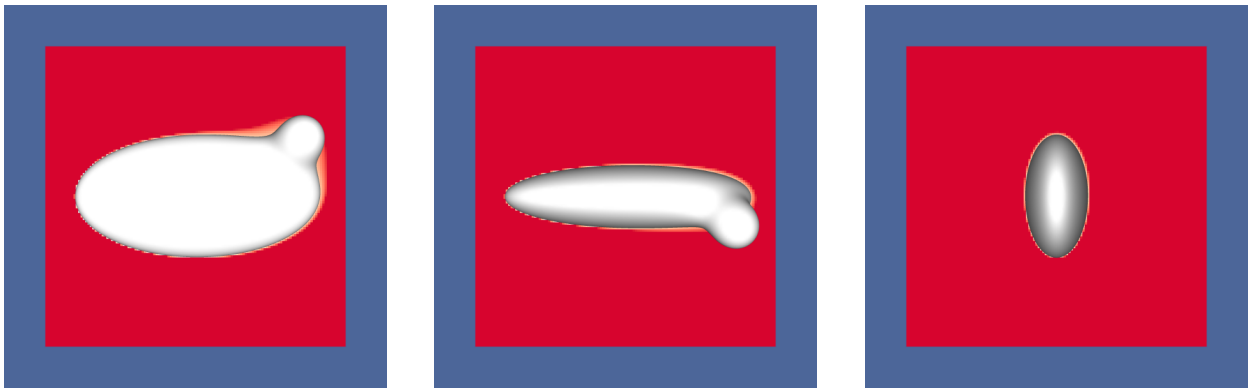


Figure A.21: Initial value of the 0.8-level-set

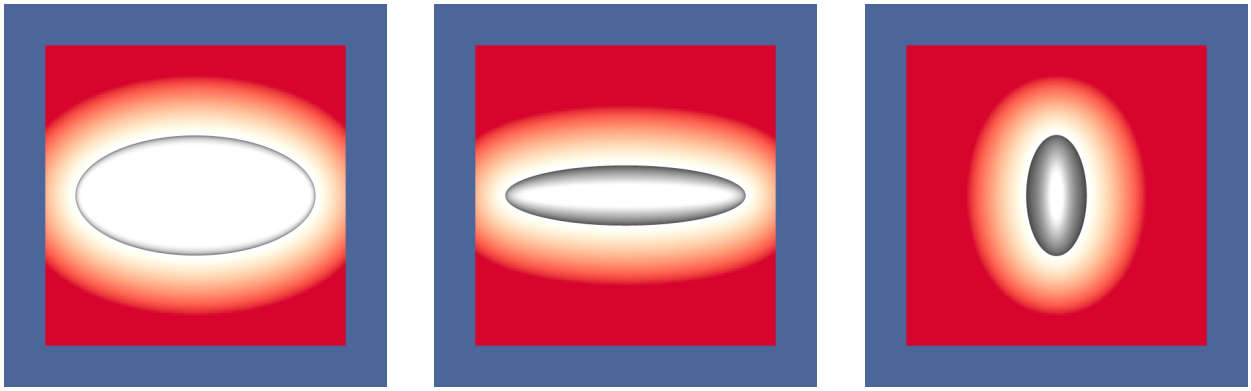


Figure A.22: Final value of the 0-level-set

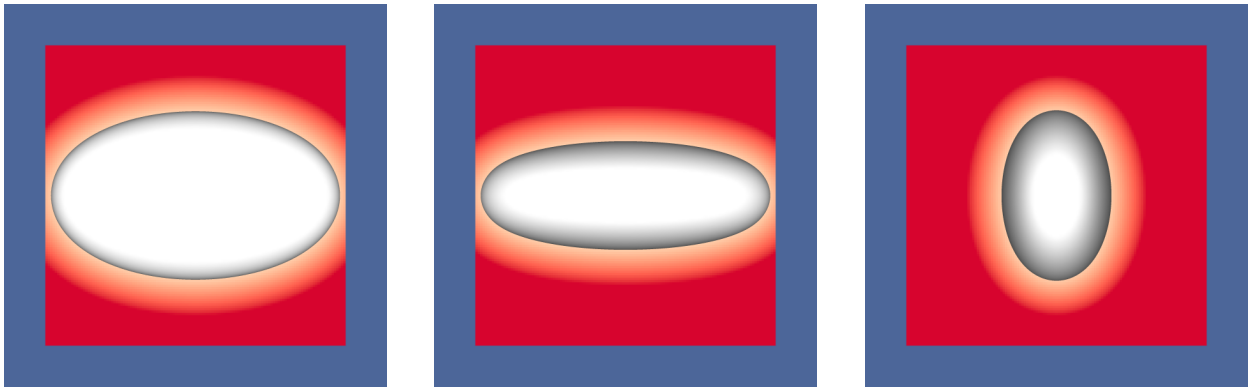
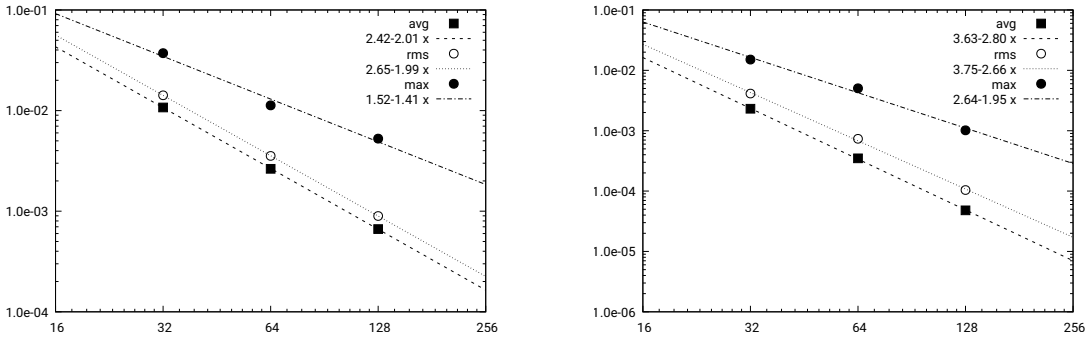


Figure A.23: Final value of the 0.8-level-set

References

[1] B. Chalmers, Principles of Solidification, Wiley, 1964.



(a) Error on the 0-level-set

(b) All cells

Figure A.24: Convergence results

[2] J. Lewandowski, M. Seifi, Metal additive manufacturing: a review of mechanical properties, Ann. Rev. Mat. Research 46 (2016) 151–186.

[3] A. Baumert, S. Bansmer, P. Trontin, P. Villedieu, Experimental and numerical investigations on aircraft icing at mixed phase conditions, International Journal of Heat and Mass Transfer 123 (2018) 957–978.

[4] M. Worster, Perspectives in Fluid Dynamics — a Collective Introduction to Current Research, Cambridge Univ. Press, 2000, Ch. Solidification of Fluids.

[5] J. Wettlaufer, M. Worster, H. H.E., The phase evolution of young sea ice, Geophys. Res. Lett. 24 (10) (1997) 1251–1254.

[6] D. Feltham, N. Untersteiner, J. Wettlaufer, W. M.G., Sea ice is a mushy layer, Geophys. Res. Lett. 33 (2006) L14501.

[7] M. Walvoord, B. Kurylyk, Hydrologic impacts of thawing permafrost-a review, Vadose Zone Journal 15 (6) (2016). doi:10.2136/vzj2016.01.0010.

[8] E. Ghabache, C. Josserand, T. Séon, Frozen Impacted Drop: From Fragmentation to Hierarchical Crack Patterns, Physical Review Letters 117 (7) (2016) 074501. doi:10.1103/PhysRevLett.117.074501.

[9] M. Schremb, C. Tropea, Solidification of supercooled water in the vicinity of a solid wall, Phys. Rev. E 94 (2016) 052804.

[10] V. Thiévenaz, T. Séon, C. Josserand, Solidification dynamics of an impacted drop, Journal of Fluid Mechanics 874 (2019). doi:10.1017/jfm.2019.459.

[11] V. Thiévenaz, C. Josserand, T. Séon, Retraction and freezing of a water film on ice, Physical Review Fluids 5 (2020) 041601(R).

[12] V. Thiévenaz, T. Séon, C. Josserand, Freezing-damped impact of a water drop, Euro. Phys. Lett. 132 (2020) 24002.

[13] A. Monier, A. Huerre, C. Josserand, T. Séon, Freezing a rivulet, Physical Review Fluids 5 (6) (2020) 062301.

[14] S. Popinet, Numerical models of surface tension, Annual Review of Fluid Mechanics 50 (2018) 49–75.

[15] S. H. Davis, Theory of Solidification, Cambridge University Press, 2006.

[16] R. Loubère, M. J. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods, Journal of Computational Physics 209 (1) (2005) 105–138.

[17] C. S. Peskin, Flow patterns around heart valves: a numerical method, Journal of computational physics 10 (1972) 252–271.

[18] C. S. Peskin, Numerical analysis of blood flow in the heart, Journal of computational physics 25 (3) (1977) 220–252.

[19] S. Lyu, K. Wang, Z. Zhang, A. Pedrono, C. Sun, D. Legendre, A hybrid vof-ibm method for the simulation of freezing liquid films and freezing drops, J. Comp. Phys. 432 (2021) 110160.

[20] M. Tembely, C. Dolatabadi, A comprehensive model for predicting droplet freezing features on a cold substratel, J.Fluid Mech. 859 (2018) 566–585.

[21] D. Juric, G. Tryggvason, A front-tracking method for dendritic solidification, Journal of computational physics 123 (1) (1996) 127–148.

[22] C.-A. Gandin, M. Rappaz, A coupled finite element-cellular automaton model for the prediction of dendritic grain structures in solidification processes, Acta metallurgica et materialia 42 (7) (1994) 2233–2246.

[23] Z. MF, A modified cellular automaton model for the simulation of dendritic growth in solidification of alloys, Isij International 41 (5) (2001) 436–445.

[24] M. F. Zhu, C. P. Hong, A three dimensional modified cellular automaton model for the prediction of solidification microstructures., ISIJ international 42 (2002) 520–526. doi:10.2355/isijinternational.42.520.

[25] K. Reuther, M. Rettenmayr, Simulating dendritic solidification using an anisotropy-free meshless front-tracking method, Journal of Computational Physics 279 (C) (2014) 63–66.

[26] V. R. Voller, An enthalpy method for modeling dendritic growth in a binary alloy, International Journal of Heat and Mass Transfer 51 (3-4) (2008) 823–834.

[27] C. W. Hirt, A. A. Amsden, J. L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, Journal of Computational Physics 14 (3) (1974) 227–253.

[28] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annual review of fluid mechanics 31 (1) (1999) 567–603.

[29] G. Caginalp, An analysis of a phase field model of a free boundary, Archive for Rational Mechanics and Analysis 92 (3) (1986) 205–245.

[30] A. Karma, W.-J. Rappel, Quantitative phase-field modeling of dendritic growth in two and three dimensions, Physical review E 57 (4) (1998) 4323.

[31] W. J. Boettinger, J. A. Warren, C. Beckermann, A. Karma, Phase-Field Simulation of Solidification, Annual Review of Materials Research 32 (1) (2002) 163–194. doi:10.1146/annurev.matsci.32.101901.155803.

[32] M. Plapp, Remarks on some open problems in phase-field modelling of solidification, Philosophical Magazine 91 (2011) 25–44.

[33] E. W. Hester, L.-A. Couston, B. Favier, K. J. Burns, G. M. Vasil, Improved phase-field models of melting and dissolution in multi-component flows, arXiv preprint arXiv:2006.16004 (2020).

[34] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, Journal of computational physics 79 (1) (1988) 12–49.

[35] S. Chen, B. Merriman, S. Osher, P. Smereka, A simple level set method for solving Stefan problems, Journal of Computational Physics 135 (1) (1997) 8–29.

[36] Y. Cheny, O. Botella, The ls-stag method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties, Journal of Computational Physics 229 (4) (2010) 1043–1076.

[37] J. Mac Huang, M. J. Shelley, D. B. Stein, A stable and accurate scheme for solving the stefan problem coupled with natural convection using the immersed boundary smooth extension method, arXiv preprint arXiv:2006.04736 (2020).

[38] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, et al., A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method), Journal of computational physics 152 (2) (1999) 457–492.

[39] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries, Journal of Computational Physics 190 (2) (2003) 572–600.

[40] D. Hartmann, M. Meinke, W. Schröder, A strictly conservative cartesian cut-cell method for compressible viscous flows on adaptive grids, Computer Methods in Applied Mechanics and Engineering 200 (9-12) (2011) 1038–1052.

[41] M. Berger, M. Aftosmis, Progress towards a cartesian cut-cell method for viscous compressible flow, in: 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2012, p. 1301.

[42] P. Schwartz, M. Barad, P. Colella, T. Ligocki, A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions, Journal of Computational Physics 211 (2) (2006) 531–550.

[43] L. Schneiders, D. Hartmann, M. Meinke, W. Schröder, An accurate moving boundary formulation in cut-cell methods, Journal of Computational Physics 235 (2013) 786–809.

[44] M. Sussman, E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, Journal of Computational Physics 162 (2) (2000) 301–337. doi:10.1006/jcph.2000.6537.

[45] G. Li, Y. Lian, Y. Guo, M. Jemison, M. Sussman, T. Helms, M. Arienti, Incompressible multiphase flow and encapsulation simulations using the moment-of-fluid method, International Journal for Numerical Methods in Fluids 79 (9) (2015) 456–490.

[46] M. Vahab, C. Pei, M. Y. Hussaini, M. Sussman, Y. Lian, An adaptive coupled level set and moment-of-fluid method for simulating droplet impact and solidification on solid surfaces with application to aircraft icing (2016). doi:10.2514/6.2016-1340.

[47] H. S. Udaykumar, R. Mittal, W. Shyy, Computation of solid–liquid phase fronts in the sharp interface limit on fixed grids, Journal of computational physics 153 (2) (1999) 535–574.

[48] T. Ye, R. Mittal, H. S. Udaykumar, W. Shyy, An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries, Journal of computational physics 156 (2) (1999) 209–240.

[49] H. Udaykumar, S. Marella, S. Krishnan, Sharp-interface simulation of dendritic growth with convection: benchmarks, International journal of heat and mass transfer 46 (14) (2003) 2615–2627.

[50] Y. Yang, H. Udaykumar, Sharp interface cartesian grid method iii: solidification of pure materials and binary solutions, Journal of Computational Physics 210 (1) (2005) 55–74.

[51] S. Gupta, The Classical Stefan Problem - Basic Concepts, Modelling and Analysis, Elsevier Science B.V, Amsterdam, 2003.

[52] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, Journal of Computational Physics 147 (1) (1998) 60–85.

[53] B. Favier, J. Purseed, L. Duchemin, Rayleigh–bénard convection with a melting boundary, Journal of Fluid Mechanics 858 (2019) 437–473.

[54] G. Ivantsov, Temperature field around a spherical, cylindrical, and needle-shaped crystal, growing in a pre-cooled melt, Doklady, Akademiya Nauk SSR, v. 58, 1947 p 567-569 (1947).

[55] W. W. Mullins, R. Sekerka, Stability of a planar interface during solidification of a dilute binary alloy, Journal of applied physics 35 (2) (1964) 444–451.

[56] J. S. Langer, Instabilities and pattern formation in crystal growth, Reviews of Modern Physics 52 (1) (1980) 1–28. doi:10.1103/revmodphys.52.1.

[57] J. Dantzig, M. Rappaz, Solidification, EPFL Press, 2011.

[58] S. Popinet, http://basilisk.frhttp://basilisk.fr (2021).

[59] F. Gibou, R. Fedkiw, S. Osher, A review of level-set methods and some recent applications, Journal of Computational Physics 353 (2018) 82 – 109. doi:https://doi.org/10.1016/j.jcp.2017.10.006.
URL http://www.sciencedirect.com/science/article/pii/S0021999117307441

[60] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational Physics 114 (1) (1994) 146 – 159. doi:https://doi.org/10.1006/jcph.1994.1155.
URL http://www.sciencedirect.com/science/article/pii/S0021999184711557

[61] Z. Solomenko, P. D. Spelt, L. Ó. Náraigh, P. Alix, Mass conservation and reduction of parasitic interfacial waves in level-set methods for the numerical simulation of two-phase flows: a comparative study, International Journal of Multiphase Flow 95 (2017) 235–256.

[62] R. Chiodi, O. Desjardins, A reformulation of the conservative level set reinitialization equation for accurate and robust simulation of complex multiphase flows, Journal of Computational Physics 343 (2017) 186–200.

[63] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive cartesian grids, Journal of Computational Physics 225 (1) (2007) 300–321.

[64] C. Min, On reinitializing level set functions, Journal of computational physics 229 (8) (2010) 2764–2772.

[65] G. Russo, P. Smereka, A remark on computing distance functions, Journal of Computational Physics 163 (1) (2000) 51–67.

[66] J. López, P. Gómez, J. Hernández, F. Faura, A two-grid adaptive volume of fluid approach for dendritic solidification, Computers & Fluids 86 (2013) 326–342.

[67] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, Journal of Computational Physics 228 (16) (2009) 5838–5866.

[68] J. A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annual review of fluid mechanics 35 (1) (2003) 341–372.

[69] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-Based Fast Local Level Set Method, Journal of Computational Physics 155 (2) (1999) 410–438. doi:10.1006/jcph.1999.6345.
URL http://www.sciencedirect.com/science/article/pii/S0021999199963453

[70] M. J. Berger, M. J. Aftosmis, An ODE-based Wall Model for Turbulent Flow Simulations, in: 55th AIAA Aerospace Sciences Meeting, 2017, p. 0528.

[71] S. Popinet, A quadtree-adaptive multigrid solver for the serre–green–naghdi equations, Journal of Computational Physics 302 (2015) 336–358.

[72] J. A. Van Hooft, S. Popinet, C. C. Van Heerwaarden, S. J. Van der Linden, S. R. de Roode, B. J. Van de Wiel, Towards adaptive grids for atmospheric boundary-layer simulations, Boundary-layer meteorology 167 (3) (2018) 421–443.

[73] J. Crank, Free and moving boundary problems, Oxford University Press, 1987.

[74] F. C. Frank, Radially symmetric phase growth controlled by diffusion, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 201 (1067) (1950) 586–599.

[75] R. Almgren, Variational algorithms and pattern formation in dendritic solidification, Journal of computational physics 106 (2) (1993) 337–354.

[76] L. Tan, N. Zabaras, A level set simulation of dendritic solidification with combined features of front-tracking and fixed-domain methods, Journal of Computational Physics 211 (1) (2006) 36–63.

[77] H. Lin, C. Chen, C. Lan, Adaptive three-dimensional phase-field modeling of dendritic crystal growth with high anisotropy, Journal of Crystal Growth 318 (1) (2011) 51–54.

[78] S. Popinet, Quadtree-adaptive tsunami modelling, Ocean Dynamics 61 (9) (2011) 1261–1285.

[79] Y.-T. Kim, N. Goldenfeld, J. Dantzig, Computation of dendritic microstructures using a level set method, Physical Review E 62 (2) (2000) 2471.

[80] J. Purseed, B. Favier, L. Duchemin, E. W. Hester, Bistability in rayleigh-bénard convection with a melting boundary, Physical Review Fluids 5 (2) (2020) 023501.

[81] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of computational physics 77 (2) (1988) 439–471.

[82] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, H.-K. Zhao, Fast sweeping algorithms for a class of hamilton-jacobi equations, SIAM journal on numerical analysis 41 (2) (2003) 673–694.