A two-grid adaptive volume of fluid approach for dendritic solidification

J. López a, P. Gómez b, J. Hernández b,*, F. Faura a

a Dept. de Ingeniería de Materiales y Fabricación, ETSII, Universidad Politécnica de Cartagena, E-30202 Cartagena, Spain
b Dept. de Mecánica, ETSII, Universidad Nacional de Educación a Distancia (UNED), E-28040 Madrid, Spain

A R T I C L E   I N F O

Article history:
Received 15 November 2012
Received in revised form 30 April 2013
Accepted 19 July 2013
Available online 27 July 2013

Keywords:
Volume of fluid method
Two-grid approach
Distance function
Solidification
Dendritic growth simulation

A B S T R A C T

A two-grid approach to simulating dendritic growth in the solidification of pure metals is presented. The approach is based on using a uniform Cartesian grid to solve the energy equation and an adaptive Cartesian grid of higher-resolution to solve the interface evolution, providing a more accurate representation of the interface. The energy equation is solved using a diffuse-interface method, and the advection equation of a discretized solid fraction function is solved using an unsplit VOF (volume of fluid) method, with PLIC (piecewise linear interface calculation) reconstruction of the interface. The thermal gradients at both sides of the interface, which are needed to obtain the front velocity, are calculated with the aid of the distance function to the reconstructed interface, which is obtained using an efficient method described in detail in this work. The influence of the grid resolution used to solve the advection equation on the accuracy of the method is analyzed. For the dendritic growth cases considered in this work, particularly, it was found that using a grid resolution for the advection equation two times higher than that used for the energy equation considerably improves the results, while keeping the CPU time consumed at an acceptable level. To underline the importance of an accurate estimation of the interface curvature, the results obtained using three techniques commonly used in VOF methods of different degrees of accuracy (a convolved VOF technique, a distance function technique and an improved height function technique, which provides second-order accuracy) are compared. The proposed methodology is assessed by comparing the numerical results with analytical solutions and with results obtained by different authors for the formation of complex dendritic structures in two and three dimensions.

© 2013 The Authors. Published by Elsevier Ltd. Open access under CC BY-NC-ND license.

1. Introduction

An important requirement of any method for solving interfacial-dominated problems, such as those considered in this work, is the ability to accurately locate the position of the interface. In the front-tracking method [1], the interface is directly tracked using an auxiliary grid formed by a set of marker points, leading to precise localization of the interface and an accurate estimation of its orientation and curvature. The front-tracking method has been applied successfully by several authors to the simulation of solid-liquid phase fronts in two (see, for example [2–5]) and three [6,7] dimensions. In this approach, which could be considered as a two-grid approach, the energy equation is solved using a fixed grid different from that used for the interface representation. The explicit interface representation of the front-tracking method also allows thermal boundary conditions to be accurately applied at the solid front. However, handling the complex topological changes of the interface that typically occur in dendritic growth is not a simple task for this method, especially in 3D. This difficulty can be avoided by using methods based on fixed grids to implicitly capture the interface, among them the phase-field, level-set, enthalpy and volume of fluid (VOF) methods, each of which has its own advantages and disadvantages.

The phase-field method has been extensively used to simulate dendritic growth problems and has considerably contributed to our knowledge of the phenomena involved in microstructure evolution in solidification processes (see, for example [8–11], and the review in [12]). This method uses an auxiliary variable that varies smoothly from zero to unity over the diffuse-interface region between the two phases, and avoids the need to explicitly compute interfacial geometric quantities and to apply thermal boundary conditions at the solid front. The relatively straightforward way to implement this method has contributed to its prominence for solving interface evolution in solidification problems during the last two decades. On the other hand, the main drawbacks of the...
phase-field method are related with the high computational effort required (the phase-field method only reproduces the dynamics of the interface for the sharp interface model at the limit when the cell size approaches zero) and the large number of parameters involved in solving the evolution equation for the phase-field variable, which are difficult to determine. An additional difficulty arises when the diffusivity presents a discontinuity across the interface. The level-set method [see, for example (13–16)] is an alternative that significantly reduces the mentioned drawbacks of the phase-field methods. As in the phase-field method, the interface position is not tracked explicitly (it is embedded as the zero-level set of a higher-dimensional function, usually a signed distance function), and complex topological changes of the interface are easily handled. Unlike in the phase-field method, the level-set function is modified with its own motion law. However, although the level-set method provide acceptable accuracy at fine resolution, their rather poor mass conservation properties and the difficulty in maintaining accuracy during topological changes limit their use, in practice, in the simulation of solidification processes. Jacot and Rappaz [17] proposed a method in two-dimensions, referred to by these authors as a pseudo front-tracking method, which calculates a distance function to a PLIC reconstructed interface, from which the interface curvature is calculated. Another approach which has certain similarities with the VOF method is the enthalpy method, although the interface evolution is not directly derived from the solution of the solid fraction advection equation. Instead, a single energy conservation equation for solid and liquid phases is solved for the enthalpy, from which a solid volume fraction (ranging from 0 to 1) is obtained to define the interface. Successful implementation of the enthalpy method for modeling dendritic growth can be found in [18,19]. A drawback of this approach is that the results are very sensitive to grid orientation, although recent improvements [20] have been proposed to reduce grid anisotropy. It is well known that the VOF method has been widely used for simulating interfacial flows with fixed grids [see, for example (21)]. However, its application to phase change processes in solidification problems is not yet common due to the difficulties involved in accurately estimating quantities related with the interface geometry. A recent successful application of a VOF method with PLIC reconstruction of the interface for simulating the crystal growth of pure metals was presented by López et al. [22], in which the thermal gradients at both sides of the interface are calculated with aid of the distance function to the reconstructed interface and the interface curvature is computed using the height function technique [23–26].

For large deformations problems, such as those considered in this work, higher resolution grids are generally required by the above mentioned implicit front-capturing methods to obtain results with an accuracy comparable to that of the front-tracking method. Additionally, the physical phenomena involved in dendritic growth problems may exhibit significant differences in length scales, making the numerical methods based on fixed grids relatively inefficient. Generally, only a small fraction of the domain, which usually coincides with a narrow region around the interface, requires high grid resolution for an accurate solution to be obtained. Adaptive grid strategies that use a higher resolution grid either side of the interface have been extensively used to overcome this difficulty; see, for example, the works of Provatas et al. [9,27], using phase-field methods, or the work of Chen et al. [28], using a level-set method.

The aim of this work was to improve the accuracy and computational efficiency of the VOF method proposed in [22] to simulate the dendritic growth of pure metals by using a two-grid approach, which involves a uniform Cartesian grid (main grid) to solve the energy equation and an adaptive Cartesian grid of higher resolution (fine grid) for the interface representation. This approach presents certain analogy with the front-tracking approach described at the beginning of this section. It should be mentioned that, in this type of problem, the thermal field generally presents smoother variations than those of the implicit function representing the interface, which, to a certain extent, permits a coarser grid to be used to solve the energy equation. This could be especially relevant in the case of complex interface structures, for which, although the spatial scales of the heat diffusion problem might be relatively well resolved, the interface would still require higher resolution grids.

In particular, we investigate, for a given main grid, the level of refinement of the adaptive grid needed to efficiently attain the maximum performance of the proposed approach. Other authors have used similar strategies to solve interfacial flow problems. For example, Rudman [29] and, more recently, Cervone et al. [30] and Caboussat et al. [31] used a grid finer than the velocity-pressure grid to solve the VOF equation. Similar approaches were used by Gómez et al. [32] and Herrmann [33] using a level-set scheme to solve the interface advection, and by Raad et al. [34], Raad and Bidoade [35] and Vincent et al., [36], among others, using Lagrangian markers to track the free surface. In the field of dendritic growth simulation, Di and Li [37] proposed a two-grid approach in which the energy and level-set evolution equations were solved using different adaptive grids. It should be mentioned that, in certain situations, only refining the grid used to solve the interface evolution may not be sufficient to capture high-order dendritic structures, in which case a full adaptive refinement of the thermal field would also become necessary to get efficient computation. Other situations where the thermal field would require adaptive refinement are, for example, interaction of dendrites with particles or with each other [38–40].

In this work, the energy equation is solved using a uniform Cartesian grid and a diffuse-interface method that avoids the need to apply the thermal boundary conditions directly at the liquid/solid front (detailed descriptions of this method can be found, for example, in [22,1,41]). The advection equation of a discretized solute fraction function, which is equal to one in solid cells, zero in liquid cells and between zero and one in interfacial cells, is solved using an adaptive Cartesian grid and the unsplit VOF advection method proposed by López et al. [42] (extended to three dimensions by Hernández et al. [26]). The thermal gradients at both sides of the interface, which are needed to accurately obtain the front velocity, are calculated with the aid of a distance function. The distance function is obtained at every time step by computing the exact distances from the computational cells surrounding the interface to the reconstructed interface segments (polygons in 3D) using an efficient method described in detail in Section 3.5. The interface curvature, whose estimation is a crucial issue in the simulation, is computed using three techniques of different degrees of accuracy and which are commonly used in VOF methods. The proposed methodology, which takes advantage of the strengths of the different combined techniques, obviously leads to an implementation complexity higher than that of a single method. Similar methodologies have already been successfully applied to two-phase flow problems (see, for example, References [43–48], just to mention a few). The accuracy and efficiency of the proposed methodology are assessed by comparing the numerical results with analytical solutions obtained in several tests and with results obtained by different authors for the formation of complex dendritic structures in two and three dimensions.

2. Governing equations for the solidification of a pure metal

In the analysis carried out in this work, the thermo-physical and transport properties of both phases are assumed to be constant. Following a volume-averaging approach [41,1] to solve the energy
equation, and assuming that the solidification process occurs in a zone of finite thickness close to either side of the reconstructed interface, the energy equation can be expressed as

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) - \mathcal{Q} \frac{\partial \phi}{\partial t}. \quad (1)$$

with

$$\Phi = \begin{cases} 0, & \text{if } \phi < -\epsilon, \\ 1, & \text{if } \phi > \epsilon, \\ 0.5 + \phi/2\epsilon, & \text{if } -\epsilon \leq \phi \leq \epsilon, \end{cases}$$

(see Reference [49]) where $\mathcal{Q} = \rho_1 \left[L + (c_v - c_s)(T_1 - T_m)\right]$ and $\rho$, $c$, and $k$ are, respectively, the density, heat capacity and thermal conductivity; $\epsilon$ is the half-thickness of the diffuse liquid/solid interface; $\phi$ is a distance function to the reconstructed interface, which is computed as described in Section 3.5; $L$ is the latent heat of solidification; $T_m$ is the pure material melting temperature; $T_i$ is the interface temperature, and subscripts $-$ and $+$ denote liquid and solid phases, respectively. The evolution of the solid volume fraction $f$ is governed by the simple advection equation

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f = 0, \quad (2)$$

where the velocity vector field used will be defined in Section 3.3. Eq. (1) is solved using the diffuse-interface method presented in Section 3.2 and Eq. (2) is solved using the PLIC-VOF method presented in Section 3.4. Note that Eq. (1) is coupled to Eq. (2) through the distance to the reconstructed interface function and the thermo-physical and transport properties of both phases, which obviously depend on the value of the solid volume fraction.

3. Numerical procedure

The method proposed in this work is based on a two-grid approach in which Eq. (1) is solved using a uniform Cartesian grid with a cell size equal to $h$ and Eq. (2) is solved using a higher-resolution adaptive Cartesian grid with a cell size $h_{\text{max}}$ equal to $h/2^{\ell_{\text{max}}}$, with $\ell_{\text{max}}$ being the level of refinement chosen (note that, for $\ell_{\text{max}} = 1$, both grids have the same size). The interface is advanced in $2^{\ell_{\text{max}}-1}$ equal time substeps ($\Delta t^{\text{adv}}$) per time step $\Delta t$ used to solve the energy equation. In this work, the fine grid is rebuilt at each substep in a local region either side of the interface using an octree-based algorithm [50,51], although a more simple uniform cell-splitting operation could have been employed. Fig. 1 shows the resulting fine grid for $\ell_{\text{max}} = 3$ which conforms to a seed with a circular interface.

The numerical procedure used in this work is summarized in Algorithm 1 and the numerical implementation details are given below.

**Algorithm 1. Summary of the two-grid approach.**

1: Initialize the front velocity $V_f^0$ at all the fine cells
2: for $n = 0$ to $n^\text{end}$ do
3:  Set the initial values of the discretized solid volume fraction $F$
4:  Construct the initial interface in the fine grid and obtain $\phi$
5:  if $n = 1$ then
6:   Initialize the front velocity $V_f^1$ at all the fine cells
7:  end if
8:  Calculate the interface curvature
9:  for $n = 1$ to $n^\text{adv}$ do
10:   Obtain the front velocity $V_f$ at the interfacial nodes
11:   Obtain $V_f^n$ by extending $V_f^0$ to all the fine cells
12: end if
13: Calculate $V_f^\phi$ and make $\mathbf{v}^\phi = \frac{V_f^\phi}{V_f^\text{ext}} V_f^\text{ext}$
14: Interpolate the components of $\mathbf{v}^\phi$ to the face centers of the fine cells
15: for $n^\text{adv} = 1$ to $2^{\ell_{\text{max}} - 1}$ do
16:  Solve Eq. (2) with $\Delta t^{\text{adv}} = \Delta t/2^{\ell_{\text{max}} - 1}$ to obtain $F^n_{\text{adv}}$
17:  Rebuild the octree grid using Algorithm 3
18:  Extend the velocity $\mathbf{v}$ to the new fine cell faces
19:  Reconstruct the interface
20: end for
21: Obtain $\phi^{n+1}$
22: Solve Eq. (1) to obtain $T^{n+1}$
23: Calculate the new time step $\Delta t^{n+1}$
24: end for

### 3.1. Grid construction

The finest grid, which only covers a local region either side of the interface, is embedded into the uniform Cartesian main grid (refinement level $\ell = 1$) using the recursive octree subdivision described below until the required grid resolution (refinement level $\ell = \ell_{\text{max}}$) is achieved. A cell with a refinement level of $\ell = 1$ to $\ell = \ell_{\text{max}} - 1$ must be refined if it is an interfacial cell or if at least one of its 8 neighboring cells (26 in 3D) is an interfacial cell. Beginning from the refinement level $\ell = 1$, any cell that satisfies this condition is divided into 4 (8 in 3D) uniform cells. This local refinement procedure is recursively applied until the maximum refinement level $\ell_{\text{max}}$ is achieved. Algorithm 2 summarizes the procedure used for the initial octree grid generation.

Due to the movement of the interface, the octree grid structure must be rebuilt for each time substep used to solve Eq. (2) (see example in Fig. 2). Cell subdivision is suppressed in any cell of refinement level $\ell_{\text{max}} - 1$ that does not satisfy the above mentioned refinement condition (this occurs in the cell depicted with a dashed black line in the right hand picture of Fig. 2). This coarsening procedure is repeated for the next grid levels until $\ell = 1$ is
reached. Then, a procedure similar to that of Algorithm 2 is carried out to refine any cell satisfying the refinement condition that has still not been refined (cells depicted with red dashed lines in the right hand picture of Fig. 2). The procedure used to rebuild the octree grid structure for each time substep is shown in Algorithm 3.

Algorithm 2. Initial octree grid generation.

1: for $\ell = 1$ to $\ell = \ell_{\text{max}} - 1$ do
2: for every cell $i$ of level $\ell$ do
3: if $i$ is an interfacial cell then
4: refine cell $i$ and any of the 8 neighboring cells to $i$ (26 in 3D) that have not yet been refined
5: end if
6: end for
7: end for

Algorithm 3. Octree grid rebuilding.

1: for $\ell = \ell_{\text{max}} - 1$ to 1 do
2: for all refined cell $i$ of the level $\ell$ do
3: if cell $i$ and its neighboring cells are not interfacial cells then
4: coarsen cell $i$
5: end if
6: end for
7: end for
8: for $\ell = 1$ to $\ell_{\text{max}} - 1$ do
9: for all cell $i$ of the level $\ell$ which is not yet refined do
10: if cell $i$ or any of its neighboring cells is an interfacial cell then
11: refine cell $i$
12: end if
13: end for
14: end for

Note that, with the above procedure, neighboring cells do not differ from each other by more than one refinement level.

3.2. Energy equation

As already mentioned, Eq. (1) is solved on a uniform Cartesian computational grid, in which the discrete values of the temperature, distance function and material properties are stored at cell centers. Eq. (1) is discretized using central differences for the spatial derivatives and a fully implicit time integration scheme. The density, thermal conductivity and heat capacity in each computational cell are calculated from

$$\rho = \rho_- + (\rho_+ - \rho_-)F, \quad k = k_+ + (k_- - k_+)F, \quad c = c_- + (c_+ - c_-)F,$$

where $F$ is the discretized version of the solid volume fraction, $f$. The last term of Eq. (1) is a source term which accounts for the latent heat released (absorbed) during solidification (melting) at the diffuse liquid/solid interface. As in [22], $\epsilon$ is taken to be equal to the cell size of the computational grid, $h$. The system of linear equations that arises from the discretization of Eq. (1) is solved using a pre-conditioned bi-conjugate gradient stabilized (Bi-CGSTAB) method with an incomplete LU factorization preconditioning matrix [52].

3.3. Interface velocity computation

The velocity of the solidification front, $V$, is determined from the energy balance at the interface, given by

$$V = \frac{\mathbf{q}}{Q},$$

where $\mathbf{q} = q_s - q_l$ is the jump in heat flux through the interface, which is calculated as indicated below. The velocity $V$ is computed in what we will call interfacial nodes (in) of the fine grid, which include the centers of the interfacial cells (ic), where $0 < F < 1$, and the centers of cells (adj) adjacent to the former (with one face in common), which satisfy the condition $\phi_c \phi_{\text{adj}} \leq 0$ (see Fig. 1). The interface velocity is subsequently extended away from the interface over all the cells in the fine grid, using the method proposed by Tan and Zabaras [53]. The extended velocity, $V_{\text{ext}}$, provides the velocity vector field used in Section 3.4 to advance the interface to the next time step,

$$\mathbf{v} = -\frac{\mathbf{V}_\phi}{V_{\text{ext}}},$$

where $\mathbf{V}_\phi$ is computed using central differences. Note that, for a cell sharing a face with a cell of a lower refinement level (see the example in Fig. 3), computing $\mathbf{V}_\phi$ requires the value of $\phi$ at a node that lies outside the fine grid (node 4 in the example of Fig. 3(a)) to be known. This value is obtained from a linear interpolation from the available $\phi$ – values at the grid with refinement level $\ell_{\text{max}} - 1$ (see Fig. 3(b)), using the procedure described in Section 3.5.

The heat fluxes at the liquid and solid sides of the interface, $q_s$ and $q_l$, respectively, are calculated at the point located in the zero-level set (point $l$ in the 2D example of Fig. 1) closest to the
considered interfacial node (in), following a procedure similar to that used in [2]. The location of point \( I \) is obtained as

\[
x_I = x_{\text{in}} - n_{\text{in}} \phi_{\text{in}},
\]

where \( n_{\text{in}} = \frac{\nabla \phi}{|\nabla \phi|} \) is computed using central differences, and two probe points at each side of the interface (see Fig. 1) are defined as

\[
x_{\text{ip}_1} = x_I + h n_{\text{in}},
\]

\[
x_{\text{ip}_2} = x_I + 2h n_{\text{in}}.
\]

The heat fluxes are then calculated as

\[
q_{\text{ip}_1} = \pm k \frac{1.5T_I - 2T_{\text{ip}_1} + 0.5T_{\text{ip}_2}}{h},
\]

where temperatures \( T_{\text{ip}_1} \) and \( T_{\text{ip}_2} \) are obtained using trilinear interpolations, and the interface temperature, \( T_I \), may in general be a function of the interface curvature and velocity.

The interface curvature, which may be involved in the calculation of the liquid/solid interface temperature, as in the tests presented in Section 4, is obtained at \( x_I \), as indicated in Section 3.6.

### 3.3.1. Assessment of the interface location calculation

To assess the accuracy of the interface location calculated through Eq. (4), Fig. 4 shows the maximum and average errors, calculated as \( |x_I - x_{I\text{exact}}| \), over all the interfacial nodes ‘in’ corresponding to a spherical shape of radius \( r = 1 \) which, in order to avoid artificial regularity of the results, is centered at \( x_c = (2.1, 1.856, 2.073) \) in a domain of \( 4 \times 4 \times 4 \). The results are obtained with different grid sizes using Youngs’ method and the second-order CLC-CBIR method proposed by López et al. [54] to reconstruct the interface. The exact interface location \( x_{I\text{exact}} \) is calculated as \( x_c + n_{I\text{exact}} r \), where \( n_{I\text{exact}} \) is the unit-length vector calculated as \( (x_{\text{in}} - x_I) / |x_{\text{in}} - x_I| \). As expected, the accuracy in the interface location increases as the grid refinement increases, with a convergence order that depends on the interface reconstruction method used.

### 3.4. Interface tracking

The interface is evolved from \( t^{\text{adv}} \) to \( t^{\text{adv}+1} \) using a PLIC-VOF approach. The interface is first reconstructed in each interfacial cell using Youngs’ PLIC scheme [55] (implemented as indicated in [54]), and then advected to the next time substep using the evolu-
tion equation for the solid volume fraction (Eq. (2)), \( f \), which can be rewritten as
\[
\frac{\partial f}{\partial t} + \mathbf{V} \cdot \nabla f - \mathbf{f} \cdot \mathbf{v} = 0.
\] (8)

Integrating over a given cell, \( \Omega \), of volume \( V_{\Omega} \), and the time interval \( \Delta t^{n+1} = t^{n+1} - t^n \), the following expression is obtained at each time step for the discrete volume fraction of the solid phase in the cell, \( F \):
\[
F^{n+1} = F^n + \int_{t_n}^{t_{n+1}} \left( \mathbf{V} \cdot \nabla f - \mathbf{f} \cdot \mathbf{v} \right) \, d\Omega \, dt + \frac{1}{V_{\Omega}} \int_{t_n}^{t_{n+1}} \mathbf{f} \cdot \, d\Omega \, dt.
\] (9)

The first integral in Eq. (9), which represents the net volume fraction advected out of the cell, \( V_{\Omega} \), is solved geometrically using the unsplit advection method proposed by López et al. [42] (extended to 3D by Hernández et al. [28]). The second integral in Eq. (9) represents the variation in the volume of the solid phase in the cell due to phase change. Using an explicit computation of the second integral in Eq. (9) yields the following expression:
\[
F^{n+1} = F^n + \int_{t_n}^{t_{n+1}} \left( \mathbf{V} \cdot \nabla f - \mathbf{f} \cdot \mathbf{v} \right) \, d\Omega \, dt - \frac{1}{V_{\Omega}} \int_{t_n}^{t_{n+1}} \mathbf{f} \cdot \, d\Omega \, dt.
\] (10)

where the term \( \mathbf{V} \cdot \nabla f \) is discretized at the cell center using a standard second-order finite difference approximation. The final volume fraction distribution at time \( t^{n+1} \) is obtained imposing the following limiting conditions to avoid the formation of ‘undershoots’ (values of \( F \) lower than 0.0) or ‘overshoots’ (values of \( F \) higher than 1.0) [22]:
\[
F^{n+1, \text{update}} = \begin{cases} 
\min[1.0, \max(F^n, F^{n+1})] & \text{if } V^n \geqslant 0, \\
\max[0.0, \min(F^n, F^{n+1})] & \text{if } V^n < 0.
\end{cases}
\] (11)

where \( V^n \) is the interface velocity at \( n \), which was calculated as indicated in Section 3.3. As in [22], the artificial generation of ‘wips’ (small volumes of liquid in solid regions or of solid in liquid regions), which generally represents a very small percentage of the total solid volume, are simply eliminated.

After each advection step, the volume fraction value of each refined cell is updated by simple averaging of the volume fraction values of its child cells. On the other hand, the front velocity components defined at the fine cell faces are extended to the faces of the new fine cells resulting from grid rebuilding (cells depicted with dashed red lines in Fig. 2), which are needed for the next advection step, using the method proposed by Tan and Zabaras [53].

3.5. Distance function construction

At the end of the \( 2^{\alpha-1} \) advection steps, and once the interface is reconstructed in every interfacial cell of the fine grid as a plane defined by \( \mathbf{n} \cdot \mathbf{x} + C = 0 \) using a PLIC scheme, where \( \mathbf{n} \) points to the solid, the distance between the center of any computational cell and the interface (defined by all the polygons resulting from the intersection between the interfacial cells and the corresponding interfacial planes) is computed in a local region either side of the interface using the algorithm presented in this section. Sussman and Puckett [43], Bourlioux [44], Menard et al. [45] and Yang et al. [46] proposed similar algorithms to compute the distance from a point to an interface polygon, although they did not provide implementation details.

1. In every cell \( (l,m,n) \) of the domain, the signed distance function is initialized as follows:
\[
\phi_{l,m,n} = \begin{cases} 
\phi_0 & \text{if } F_{l,m,n} \geqslant 0.5, \\
-\phi_0 & \text{if } F_{l,m,n} < 0.5.
\end{cases}
\] (12)

where \( F_{l,m,n} \) is the solid volume fraction at the cell. The initial value \( \phi_0 \) is made equal to \( 4h/2^\nu \), with \( \nu \) being the refinement level of the considered cell.

2. The distance \( d \) to the interface polygon (Fig. 5) constructed at every interfacial cell \( (l,n,m') \) \( (0 < F_{l,m,n'} < 1) \) of the fine grid is computed from every cell \( (l,m,n) \) in a local region either side of the interface within a distance \( 3h/2^{\nu-1} \), with \( \nu \) being the refinement level of cell \( (l,m,n) \), following the procedure described below in Section 3.5.1. The distance function \( \phi_{l,m,n} \) is then updated as
\[
\phi_{l,m,n} = \text{sign}(\phi_{l,m,n}) \min(|d|, |\phi_{l,m,n}|).
\] (13)

3.5.1. Distance from a point to an interface polygon

The procedure proposed to compute the distance, \( d \), from a cell center \( P \) to an interface polygon \( \mathcal{P} \) of \( N \) vertices is applicable to convex polygons of an arbitrary number of faces, although it can easily be extended to the general case. Other procedures were proposed in [47,48], although these are limited to orthogonal hexahedral grids, in which the number of interface polygon vertices varies from 3 to 6.

In the present procedure it will be assumed that the \( N \) vertices of the reconstructed interface polygon are ordered counterclockwise as viewed from inside the solid. All the geometrical operations needed to obtain the ordered vertices of the polygon resulting from the intersection between an interfacial cell and the corresponding interface plane, \( \mathbf{n} \cdot \mathbf{x} + C = 0 \), are made using the geometrical tools described in [56] and the routines supplied in [57]. The algorithm basically consists of the following steps:

1. At every edge \( \tau_i \) of the interface polygon, defined by vertices \( \mathbf{x}_i \) and \( \mathbf{x}_{i+1} \), a plane \( \Gamma_i \) perpendicular to the interface polygon and containing the edge, is defined as \( \mathbf{n} \cdot \mathbf{x} + C_i = 0 \) (the constant \( C_i \) can be obtained as \( -\mathbf{n} \cdot \mathbf{x}_i \), where \( \mathbf{n} = \mathbf{n} \times \mathbf{r}_i \), with \( \mathbf{r}_i = (\mathbf{x}_{i+1} - \mathbf{x}_i)/|\mathbf{x}_{i+1} - \mathbf{x}_i| \) (see Fig. 5(a) and (b)). Then, the signed distance, \( \phi_i \), from the cell center, \( P \), to every plane \( \Gamma_i \) of the interface polygon \( (i = 1, \ldots, N) \) is computed as
\[
\phi_i = \mathbf{n} \cdot \mathbf{x}_P + C_i.
\] (14)

2. If \( P \) is inside the region delimited by the \( N \) planes \( \Gamma_i \) (Fig. 5(b)), which means that all the values \( \phi_i \) have a positive sign, the distance, \( d \), from \( P \) to the interface polygon is calculated as
\[
d = |\mathbf{n} \cdot \mathbf{x}_P + C|.
\] (15)

and the algorithm finishes.

3. Find the first vertex \( i \) with \( \phi_i < 0 \), and construct two new planes, \( \Gamma_{i-1} \) and \( \Gamma_{i+2} \), perpendicular to edge \( \tau_i \) and passing through points \( \mathbf{x}_i \) and \( \mathbf{x}_{i+1} \), respectively, defined as \( \tau_i \cdot \mathbf{x} + C_{i+2} = 0 \) (the constant \( C_{i+2} \) is calculated as \( -\mathbf{r}_i \cdot \mathbf{x}_i \) and \( -\mathbf{r}_i \cdot \mathbf{x} + C_{i-1} = 0 \) (the constant \( C_{i+2} \) is calculated as \( -\mathbf{r}_i \cdot \mathbf{x}_{i-1} \)). Compute the signed distances from point \( P \) to \( \Gamma_{i-1} \) and \( \Gamma_{i+2} \), respectively, as
\[
\phi_{i-1} = -\tau_i \cdot \mathbf{x}_P + C_{i+1},
\] (16)
\[
\phi_{i+2} = -\tau_i \cdot \mathbf{x}_P + C_{i+2}.
\] (17)
4. If $\phi_{i+1} \leq 0$ and $\phi_{i} \leq 0$, which means that $P$ is inside the region delimited by the new planes $\Gamma_{i+1}$ and $\Gamma_{i}$ (Fig. 5(c)), compute $d$ as the distance from $P$ to edge $L_i$ as
\[ d = |x_i - x_i + 1|, \]
where
\[ x_Q = x_i + [n_i \cdot (x_P - x_i)]n_i, \]
and the algorithm finishes.

5. If $\phi_{i+1} \leq 0$ (Fig. 5(d)), the distance $d$ is computed as
\[ d = |x_i \cdot n_i - x_P|, \]
where
\[ x_Q = x_i \cdot n_i + (x_P - x_i)\cdot n_i, \]
and the algorithm finishes.

6. If $\phi_{i+1} \geq 0$ and $\phi_{i+1} > 0$ (note that these conditions may only hold simultaneously if the interior angle of the polygon at vertex $i + 1$ is lower than 90), the distance $d$ is computed as
\[ d = |x_{i+1} - x_P|, \]
and the algorithm finishes; otherwise, the algorithm continues with Step 3 for the next vertex $i$ where $\phi_i < 0$.

The source codes and pseudo-codes for the above algorithm (2D and 3D versions) are available for download at [57].

3.5.2. Assessment of the distance function calculation procedure

Obviously, the accuracy of the distance function will depend on the accuracy of the PLIC scheme used to reconstruct the interface. Fig. 7(a) shows the distance function errors $E_{Lx}$ and $E_{Lx}$, defined as
\[ E_{Lx} = \frac{\sum_{n=0}^{n_{df}} (\phi - \phi_{\text{exact}})^2}{n_{df}}, \]
\[ E_{Lx} = \max(|\phi - \phi_{\text{exact}}|), \]
for a sphere of radius 0.25, centered in a unit domain, obtained over a zone close to either side of the interface of thickness equal to $6h$. 

which contains \( n_{el} \) cells, using two reconstruction schemes: Youngs’ method and the second-order CLC-CBIR method [54]. It can be observed from the figure that the proposed method for computing the distance function, in combination with a second-order reconstruction method, produces results with second-order convergence. However, since the main objective of this work is to assess the capability of a conventional PLIC-VOF method for simulating solidification problems, all the results presented in this work were obtained using the simpler reconstruction method of Youngs, which, additionally, will facilitate future comparison of the performance of other PLIC-VOF methods in the analysis of solidification problems.

Cummins et al. [23] proposed a different approach based on computing the distance function from the centroids of the reconstructed interface segments. A variant of this method in three-dimensions was proposed by Lovic and Lakehal [58]. These authors do not take into account that the point of the interface closest to a given cell center could be any point other than the centroid of the reconstructed interface, and, to reduce the errors in the distance function computation, they apply an averaging procedure. The errors \( E_{L_1} \) and \( E_{L_2} \) obtained using their method for distance calculation and the CVTNA reconstruction method [59] are compared in Fig. 7(b) with those obtained using the distance calculation method proposed in this work and the CLC-CBIR reconstruction method, in a zone close to either side of the interface of fixed thickness equal to 0.1. Note that, for the finest grid, the \( L_\infty \) error norm obtained with the proposed method is one-order of magnitude lower than that obtained by Lovic and Lakehal [58]. As both the CVTNA and CLC-CBIR methods provide similar accuracy, the differences between the results presented in Fig. 7(b) can be mainly attributed to the distance calculation method used.

### 3.6. Interface curvature computation

One aspect that may significantly affect the accuracy of the results obtained for dendritic growth simulation is the interface curvature. To demonstrate the importance of accurately estimating
the interface curvature in this type of problem, the results presented in this work were obtained using three interface curvature techniques of different degrees of accuracy, and which are commonly used in VOF approaches: (1) convolution (CV), (2) distance function (DF) and (3) height function (HF) techniques. A brief description of each technique follows.

3.6.1. CV technique

To obtain the interface curvature using this technique, the interfacial unit-length normal vector is first calculated at cell \((i,j,k)\) as

\[
n_{ICV} = \frac{\nabla F}{|\nabla F|},
\]

(21)

where, for example,

\[
\nabla F = - \sum_{\text{int}(d_0)} \text{max} \left[ 0, \left( d_0^2 - d^2 \right) \right]^2 6 r F_{i,j,k,t},
\]

(22)

with \(d_0\) being the smoothing length divided by cell size and \(d = (r^2 + s^2 + t^2)^{1/2}\). Then, the curvature \(\kappa\) is obtained as

\[
\kappa = - \nabla \cdot n_{ICV},
\]

(23)

using standard centered finite-difference operators. In this work, a smoothing length of \(d_0 = 3\) is used. Finally, the interface curvature \(\kappa_I\) at the location \(\mathbf{x}_I\) obtained from Eq. (4) (see Fig. 8(a)) is bilinearly interpolated (trilinearly interpolated in 3D) from the curvatures calculated for surrounding cells from Eq. (23).

3.6.2. DF technique

This approach is based on the distance function \(\phi\) reconstructed using the algorithm described in Section 3.5. First, the interfacial unit-length normal vector is calculated as

\[
n_{DF} = \frac{\nabla \phi}{|\nabla \phi|},
\]

(24)

where, for example, \(\nabla \phi\) is calculated using the expression equivalent to Eq. (22) for the distance function \(\phi\) (a similar smoothing approach was proposed by Cummins et al. [23] to alleviate the high frequency errors in the reconstructed DF). Then, the curvature \(\kappa\) is obtained using the equivalent expression to Eq. (23). In this case, we use a smoothing length \(d_0 = 2\) (a more detailed analysis of the influence of the smoothing length on the curvature accuracy will be discussed elsewhere). Finally, the interface curvature \(\kappa_I\) at \(\mathbf{x}_I\) (see Fig. 8(a)) is interpolated as in the above section.

3.6.3. HF technique

The interface curvature in each interfacial cell, \(\kappa_{IC}\), is computed using the improved technique with correction of the HF discretization error proposed by López et al. [60], which also uses the adaptive HF stencil proposed by Hernández et al. [26]. The partial derivatives of the HF function are obtained using the finite difference formula proposed by López et al. [61]. Then, the interface curvature \(\kappa_I\) at \(\mathbf{x}_I\) is obtained from the inverse distance weighted interpolation (see Fig. 8(b))

\[
\kappa_I = \sum_{i} \frac{\kappa_i}{d_{IC}}^2,
\]

where \(d_{IC}\) is the distance from point \(i\) to the center of the interfacial cell ‘ic’ and the summation extends over the interfacial cells within a \(3 \times 3\) stencil \((3 \times 3 \times 3\) in 3D\) centered at the cell containing point \(i\).

3.6.4. Assessment of the interface curvature computation

Table 1 shows the maximum and average errors of the curvature calculated at the interface points \(I\) for different grid sizes, using the three techniques described above for a circular interface with radius equal to 1 and a spherical interface with radius equal to 2. Note that the HF technique produces the most accurate results, with a second-order convergence. For the finest grid considered, the HF technique provides an accuracy two or three order of magnitude higher than that obtained with the other techniques. Also, note that the other two techniques do not produce results convergent with grid refinement, and that, in general, the DF technique produces more accurate results than those obtained using the CV technique.

4. Results and discussion

To assess the accuracy and computational efficiency of the proposed two-grid adaptive VOF approach, different tests (three in 2D and one in 3D) were performed. First, a comparison with the exact solution of the Frank sphere test is carried out in Section 4.1. Then, dendritic growth simulations in 2D and 3D, which must be able to reproduce complex interface shapes, are conducted in Sections 4.2 and 4.3, respectively. Finally, our numerical solutions are compared in Section 4.4 with solvability theory and with the results obtained by other authors using level set and phase field methods. All the simulations presented in this section were carried out using the same thermo-physical and transport properties for the solid and liquid regions \((c_s = c_l = 1, k_s = k_l = 1, \rho_s = \rho_l = 1)\) and \(L = 1\).

4.1. Frank sphere test

In this test, taken from the work of Chen et al. [14], the solid region is a circle of radius \(R\) centered in the domain, and the temperature field in parametric form is given by

\[
T(r, t) = T(s) = \left\{ \begin{array}{ll} T_\infty \left(1 - \frac{1}{s/S} \right) & \text{for } s > S \\ 0 & \text{for } s \leq S \end{array} \right.
\]

(25)

where \(T_\infty\) is a given undercooling temperature, \(s\) is the distance to the center of the domain, \(s = r t^{1/2}\), \(S = R t^{1/2}\), and \(F(s) = \int_{-\infty}^{s} e^{-z^2/2} dz\). An exact solution in two dimensions can be found in [62]. The results for the mean radius error obtained for \(T_\infty = -0.5\), a domain size of \(16 \times 16\), homogeneous Neumann boundary conditions (insulated domain), a time step \(\Delta t = 0.1h\) for the solution of Eq. (1) and three grid sizes with several refinement levels for the solution of Eq. (2) are presented in Fig. 9. It can be observed that for the coarsest grid, the maximum performance of the proposed method is achieved using three refinement levels for the
Table 1
Maximum, $E_{\text{max}}$, and average, $E_{\text{avg}}$, interface curvature errors for a circular shape with radius equal to 1 and a spherical shape with radius equal to 2, obtained with different grid sizes using the CV, DF and HF techniques (the convergence orders are included in parentheses).

<table>
<thead>
<tr>
<th>Grid size</th>
<th>CV $E_{\text{max}}$</th>
<th>CV $E_{\text{avg}}$</th>
<th>DF $E_{\text{max}}$</th>
<th>DF $E_{\text{avg}}$</th>
<th>HF $E_{\text{max}}$</th>
<th>HF $E_{\text{avg}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circular interface with radius equal to 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>$5.70 \times 10^{-2}$</td>
<td>$2.54 \times 10^{-2}$</td>
<td>$3.61 \times 10^{-2}$</td>
<td>$2.03 \times 10^{-2}$</td>
<td>$5.74 \times 10^{-2}$</td>
<td>$1.72 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>(-0.9)</td>
<td>(0.4)</td>
<td>(1.1)</td>
<td>(4.5)</td>
<td>(3.14)</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>$1.08 \times 10^{-1}$</td>
<td>$6.66 \times 10^{-2}$</td>
<td>$2.71 \times 10^{-2}$</td>
<td>$9.36 \times 10^{-3}$</td>
<td>$2.51 \times 10^{-3}$</td>
<td>$1.92 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>(-1.2)</td>
<td>(-0.6)</td>
<td>(-0.9)</td>
<td>(1.8)</td>
<td>(1.9)</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>$2.54 \times 10^{-1}$</td>
<td>$9.38 \times 10^{-2}$</td>
<td>$4.13 \times 10^{-2}$</td>
<td>$1.71 \times 10^{-2}$</td>
<td>$7.07 \times 10^{-4}$</td>
<td>$5.31 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td>(-1.3)</td>
<td>(-0.6)</td>
<td>(0.4)</td>
<td>(1.2)</td>
<td>(1.9)</td>
<td></td>
</tr>
<tr>
<td>0.025</td>
<td>$6.45 \times 10^{-1}$</td>
<td>$1.67 \times 10^{-1}$</td>
<td>$6.10 \times 10^{-2}$</td>
<td>$1.32 \times 10^{-2}$</td>
<td>$3.04 \times 10^{-4}$</td>
<td>$1.39 \times 10^{-4}$</td>
</tr>
<tr>
<td>Spherical interface with radius equal to 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>$6.24 \times 10^{-2}$</td>
<td>$1.91 \times 10^{-2}$</td>
<td>$1.51 \times 10^{-2}$</td>
<td>$5.40 \times 10^{-3}$</td>
<td>$6.19 \times 10^{-3}$</td>
<td>$2.10 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>(-1.3)</td>
<td>(-0.7)</td>
<td>(-1.5)</td>
<td>(-0.7)</td>
<td>(1.6)</td>
<td>(2.3)</td>
</tr>
<tr>
<td>0.1</td>
<td>$1.53 \times 10^{-1}$</td>
<td>$3.12 \times 10^{-2}$</td>
<td>$4.13 \times 10^{-2}$</td>
<td>$9.04 \times 10^{-3}$</td>
<td>$1.98 \times 10^{-3}$</td>
<td>$4.31 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td>(-1.3)</td>
<td>(-1.2)</td>
<td>(-0.7)</td>
<td>(2.1)</td>
<td>(1.9)</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>$3.69 \times 10^{-1}$</td>
<td>$5.66 \times 10^{-2}$</td>
<td>$9.73 \times 10^{-2}$</td>
<td>$1.50 \times 10^{-2}$</td>
<td>$4.47 \times 10^{-4}$</td>
<td>$1.13 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

$R_{\text{exact}} - R$

$40 \times 4C$

$\ell_{\text{max}} = 1$ 2 3 4 5

$60 \times 6C$

$\ell_{\text{max}} = 1$ 2 3 4 5

$80 \times 8C$

$\ell_{\text{max}} = 1$ 2 3 4 5

$\tau = 0.1h$ and three grid sizes for the solution of Eq. (1) with different refinement levels for the solution of Eq. (2).

The test presented in this section is that proposed by Juric and Tryggvason [1], in which the following Gibbs-Thomson relation is considered for the interface temperature:

$$T_I = \epsilon_\kappa \kappa - \epsilon_V V,$$

where $\epsilon_\kappa = \epsilon_V = 0.002$ (in the present work, $\kappa$ is negative when the center of curvature lies in the solid phase). An initial solid seed, given in parametric form by

$$x(s) = [0.1 + 0.02 \cos(8\pi s)] \cos(2\pi s),$$
$$y(s) = [0.1 + 0.02 \cos(8\pi s)] \sin(2\pi s),$$

is considered in the center of a domain of size $4 \times 4$ with all the boundaries insulated. The initial temperatures of the solid seed and of the undercooled region are made equal to 0.0 and -0.5, respectively. The time step for the solution of Eq. (1) was chosen as $\Delta t = 0.025h/V_{\text{max}}$, where $V_{\text{max}}$ is the maximum Cartesian component of $v$. A simple iterative procedure is used to solve the coupling between Eqs. (26), (3) and (7) until the difference between the interface temperatures of two consecutive iteration steps at any cell of the domain is lower than $10^{-8}$.

Fig. 10 shows the results of the test for the evolution of the interface shape, in time increments of 0.04 up to $t = 1.0$, obtained with different refinement levels for the solution of the advection equation and the three techniques considered in this work to compute the interface curvature. In this test, we use a relatively coarse main grid of size $100 \times 100$ to solve Eq. (1), which underlines the differences between the results obtained with the three curvature computation techniques considered in this work. The results presented in this figure clearly illustrate the importance of an accurate estimation of the interface curvature. Note the relatively high independence of the results obtained with the HF technique from the refinement level used to solve Eq. (2). Also note that the maximum performance of the method for the main grid used is reached for $\ell_{\text{max}} = 2$ when the HF or DF technique is used, while for the CV technique, an additional refinement level is required for the results of the advection equation. For the finest grid, the accuracy of the results can be further increased by using an additional refinement level, although, as will be discussed below, at the cost of a considerable increase in CPU time. The error between the exact and computed mean radius at $t = 1.5$ is also presented in Table 2 as a function of grid size. The results show that the convergence is better than second order for fine grids.
to be independent of \( \ell_{\text{max}} \), which would obviously affect the CPU time consumed.

The influence on the interface shape results of the refinement level used for the advection equation can be more clearly seen in Fig. 11(a), which shows the superimposed results for the interface shapes obtained using the three refinement levels and curvature computation techniques considered in Fig. 10 at several instants, in time increments of 0.2. A detail of the superimposed interfaces at \( t = 0.2 \) is also shown in Fig. 11(b). It can be observed that the results obtained with \( \ell_{\text{max}} = 2 \) and 3 are almost the same whether the HF and DF technique is used (for the CV technique, the differences are greater). To get more accurate results, the main grid where the energy equation is solved should also be refined. The results obtained using the HF technique, \( \ell_{\text{max}} = 2 \) and different main grid sizes are also shown in Fig. 12. Note that almost grid-independent results are obtained for the intermediate main grid, which compares favorably with results presented by other authors (see, for example, the results in References [1, 53, 14, 28]) and shows a very good degree of agreement with those obtained by Udaykumar et al. [2] and Zhao and Heinrich [5] using a front-tracking method. Several tests were conducted to assess the effect of grid anisotropy on the results. For example, we carried out simulations with the same initial seed of Fig. 10 rotated 19° and 45° with respect

### Table 2

Mean radius error in the Frank sphere test for different main grid sizes and refinement levels at \( t = 1.5 \) (convergence orders are in parentheses).

<table>
<thead>
<tr>
<th>Grid size, ( h )</th>
<th>Refinement level, ( \ell_{\text{max}} )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>16/40</td>
<td>( 2.08 \times 10^{-2} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 1.61 \times 10^{-2} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 1.40 \times 10^{-2} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 1.40 \times 10^{-2} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 1.40 \times 10^{-2} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16/60</td>
<td>( 7.40 \times 10^{-3} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 5.69 \times 10^{-3} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 4.25 \times 10^{-3} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 3.73 \times 10^{-3} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 3.70 \times 10^{-3} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 3.70 \times 10^{-3} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16/80</td>
<td>( 4.73 \times 10^{-3} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 3.73 \times 10^{-3} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 2.15 \times 10^{-3} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 9.87 \times 10^{-4} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 7.76 \times 10^{-4} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 10. Results of the dendritic growth simulation test in 2D obtained with a main grid size of \( 100 \times 100 \), \( \epsilon = \epsilon_{V} = 0.002 \), three different techniques for interface curvature computation, and three refinement levels: (a) \( \ell_{\text{max}} = 1 \), (b) 2 and (c) 3.
to grid lines, which were stopped at \( t = 0.14 \) in order to avoid boundary effects. The results were rotated to 0° and superimposed, and the interface shapes were found to be almost coincident.

### 4.3. Dendritic growth simulation in 3D

In order to assess the accuracy and efficiency of the proposed model for simulating dendritic growth in 3D, a test taken from [7,6] was carried out (see also Reference [22]). The interface temperature is assumed to be given by

\[
T_i = \left\{ 1 - 0.3 \left[ 4 \left( n_x^4 + n_y^4 + n_z^4 \right) - 3 \right] \right\} 10^{-4} \kappa, \tag{28}
\]

where \((n_x, n_y, n_z)\) are the components of the interface normal vector \( \mathbf{n} \), calculated as \( \nabla \phi / |\nabla \phi| \) at the corresponding interfacial node using central differences. The initial seed is a sphere of radius \( R_0 = 0.035 \) perturbed as

\[
R = R_0 \left\{ 1 + 0.1 \cos \left[ \pi \arccos \left( \frac{1}{\sqrt{3}} \right) \right] \right\},
\]

where \( \pi = \arccos(\max(|n_x|, |n_y|, |n_z|)) \). An octant, of size 0.5³, of the physical domain is considered, with appropriate boundary conditions applied at the symmetry planes. The initial temperatures of the solid seed and of the undercooled region are made equal to 0.0 and 0.15, respectively. The time step is chosen as \( \Delta t = 0.05 h / V_{\text{max}} \).

Fig. 13 shows the computed results for the interface shape at \( t = 0.03 \) using the different curvature computation techniques and a main grid of 80³ cells to solve Eq. (1) with different refinement levels for the calculation of the interface evolution. As in the 2D dendritic growth simulations, it can be observed that the re-
Fig. 12. Dendritic growth simulation test in 2D. Results obtained for the same conditions as in Fig. 10, using the HF technique to compute the interface curvature, $\ell_{\text{max}} = 2$ and main grid sizes of (a) 100$^2$, (b) 200$^2$ and (c) 400$^2$ cells.

Fig. 13. Dendritic growth simulation test in 3D. Computed results for the interface shape at $t = 0.03$ using different curvature computation techniques and a main grid of 80$^3$ cells. The isocontours of $F = 0.5$, triangulated using the method described in [54], are represented for (a) $\ell_{\text{max}} = 1$, (b) $\ell_{\text{max}} = 2$ and (c) $\ell_{\text{max}} = 3$. (d) Comparison of the isocontours of $F = 0.5$ at the plane $z = 0$ for different refinement levels.
results obtained using the HF technique exhibit the highest degree of grid independence. A comparison between our results using the HF technique and the results of a front-tracking method presented by Zhao et al. [7] with the same main grid size reveals a good degree of agreement.

In order to assess the computational efficiency of the proposed method, Fig. 14 shows the percentages of cumulative CPU-time consumed by the different algorithms used in the simulation at \( t = 0.03 \) (the test was run on an Intel T9550 processor) for different refinement levels. Note that the most time-consuming task for \( \ell_{\text{max}} = 1 \) and 2 is the resolution of the energy equation, even though the Bi-CGSTAB algorithm converges very fast in this type of problem (only around five iterations are usually needed). As expected, the CPU time consumed by this task relative to that consumed by the algorithms solved in the fine grid (VOF, distance function, and interface curvature), diminishes as \( \ell_{\text{max}} \) increases. For \( \ell_{\text{max}} = 3 \), the time consumed by the advection and reconstruction of the interface becomes considerably higher than that for \( \ell_{\text{max}} = 2 \), although the improvement in the results is negligible whether the DF or HF technique is used, as can be observed from Fig. 13. Also note that, when the CV technique is used, a minimum value of \( \ell_{\text{max}} = 3 \) would be necessary to achieve an accuracy comparable to that obtained with the HF technique, which would considerably increase the computation time. Note the relatively low CPU time consumed by the proposed algorithm for distance function computation; indeed, it is considerably lower than that consumed by the reconstruction and advection algorithms (for \( \ell_{\text{max}} = 1 \), even lower than that required by the velocity front calculation).

4.4. Comparison with solvability theory

In this section we compare the steady-state results obtained for dendritic growth using the proposed approach with the predictions of microscopic solvability theory, applying a test taken from [15], in which it is assumed that, in Eq. (26), \( \epsilon_w = 0.5(1 - 0.75 \cos 4\theta) \), where \( \theta \) is the angle between the outward interface normal vector and the \( x \)-direction, and \( \epsilon_v = 0 \). The initial seed is a circle of radius 15. A quarter (600 × 600) of the physical domain is considered (using a domain double this size produced a variation in the steady-tip velocity of less than 0.06%), with appropriate boundary conditions applied at the symmetry planes and assuming that the rest of the boundaries are insulated. For all the simulations, the initial temperature of the solid seed is taken to be equal to 0.0, \( \ell_{\text{max}} = 2 \), and a time step of \( \Delta t = 0.025 / h \) is chosen. Two different cases are considered for values of the initial temperature of the undercooled region: \( T_{\infty} = -0.55 \) and \( -0.65 \). To determine an appropriate size for the main grid used to solve Eq. (1) in each case and to ensure the grid-independence of the results, an analysis of sensitivity to the grid was carried out using the HF technique to compute the interface curvature. It can be observed from Fig. 15 that a main grid size of 200\(^2\) cells for \( T_{\infty} = -0.55 \) and 600\(^2\) cells for \( T_{\infty} = -0.65 \) (marked with vertical arrows in the top pictures) is an appropriate choice. Note that the steady-state tip velocity varies by less than 2.8 \% for \( T_{\infty} = -0.55 \) and less than 0.6 \% for \( T_{\infty} = -0.65 \) when the number of main grid cells is further increased by 200 along each coordinate direction. Also note from the bottom pictures in Fig. 15 that nearly grid-independent results are obtained for the interface shape, especially for \( T_{\infty} = -0.65 \). In the rest of this section, the grid sizes marked with vertical arrows in Fig. 15 will be used.

Fig. 16 shows, for the same simulation settings of Fig. 15, the results obtained with the different interface curvature computation techniques considered in this work at various instants, in time increments of 500 up to 9000 for \( T_{\infty} = -0.55 \) and of 100 up to 1500 for \( T_{\infty} = -0.65 \). Fig. 17(a) shows a comparison with the interface shape at \( t = 9400 \) obtained by Kim et al. [15] using a level-set method with a grid size of \( h = 0.4 \) for \( T_{\infty} = -0.55 \) and \( h = 0.2 \) for \( T_{\infty} = -0.65 \), and with that obtained by Provatas et al. [9] using a phase field method with an adaptive grid. Note that the interface shape obtained with the proposed approach and the HF technique to compute the interface curvature is almost identical to that obtained by Kim et al. [15] and very close to that obtained by Provatas et al. [9]. The results obtained for the interface velocity at the dendrite tips as a function of time are also compared in Fig. 17(b) with those obtained by Kim et al. [15] and Provatas et al. [9]. The comparison shows a good degree of agreement (similar agreement is also found with the more recent results presented in [20] using the enthalpy method). The steady-state values of \( V_{\text{tip}} = 0.034 \) for \( T_{\infty} = -0.55 \) and 0.0938 for \( T_{\infty} = -0.65 \) predicted by microscopic solvability theory are also included in the figure. Table 3 shows the steady-state tip velocities calculated using the
Fig. 15. Grid sensitivity of results (steady state tip velocities on the top pictures and interface shapes on the bottom pictures) for the solvability test using the HF technique, different main grid sizes and a refinement level $\eta_{\text{max}} = 2$. Results obtained for (a) $T_1 = -0.55$ at $t = 9000$ and (b) $T_1 = -0.65$ at $t = 1500$.

Fig. 16. Solvability theory test. Same conditions as in Fig. 15, but using different interface curvature computation techniques. Interface shapes obtained for (a) $T_1 = -0.55$ in time increments of 500 up to 9000 and (b) $T_1 = -0.65$ in time increments of 100 up to 1500.
different interface curvature techniques at $t = 9000$ for $T_\infty = -0.55$ and at $t = 1500$ for $T_\infty = -0.65$. A very good degree of agreement with solvability theory can be observed, especially when the HF technique is used to compute the interface curvature.

5. Conclusions

A two-grid approach based on an adaptive PLIC-VOF method to calculate the interface and a diffuse-interface method to solve the energy equation has been proposed for the simulation of dendritic growth of pure metals. The diffuse interface was defined by using a distance function distribution obtained using an efficient method which has been described in detail. A detailed analysis of the influence on the accuracy of the results of the grid resolution used to solve the advection equation and of the method used to calculate the interface curvature was carried out. We have demonstrated that using a locally refined grid to solve Eq. (2), while keeping the grid used for the energy equation unrefined, can substantially improve the accuracy of the results. Also, the importance of using an accurate technique, such as the HF technique, to calculate the interface curvature for dendritic growth simulation has been demonstrated. When the numerical results obtained with the proposed approach were compared with analytical solutions and with numerical results obtained by other authors for the formation of complex dendritic structures in two and three dimensions, a very good degree of agreement was found, demonstrating that the capabilities of VOF methods are comparable to other methods such as the phase-field, front-tracking and level-set methods, for the simulation of solidification problems.

Table 3

<table>
<thead>
<tr>
<th>Dendritic tip velocity $V_{tip}$</th>
<th>$T_\infty = -0.55$</th>
<th>$T_\infty = -0.65$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present results, CV</td>
<td>0.0313</td>
<td>0.0912</td>
</tr>
<tr>
<td>Present results, DF</td>
<td>0.0310</td>
<td>0.0906</td>
</tr>
<tr>
<td>Present results, HF</td>
<td>0.0339</td>
<td>0.0933</td>
</tr>
<tr>
<td>Solvability theory</td>
<td>0.0340</td>
<td>0.0938</td>
</tr>
</tbody>
</table>

Acknowledgements

The authors gratefully acknowledge the support of the Ministerio de Economía y Competitividad and FEDER through projects DPI2009-11788 and DPI2010-21696-C02.

References