Modelling dendritic solidification with melt convection using the extended finite element method

Nicholas Zabaras[1], Baskar Ganapathysubramanian and Lijian Tan

Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, 188 Frank H.T. Rhodes Hall, Cornell University, Ithaca, NY 14853-3801, USA

Abstract

Dendritic solidification of pure materials from an undercooled melt is studied using the extended finite element method/level set method for modelling the thermal problem and a volume-averaged stabilized finite element formulation for modelling fluid flow. The extended finite element method using evolving enrichment functions allows accurate modelling of the discontinuous thermal conditions at the moving sharp freezing front thus capturing its motion precisely. The solution of the velocity field in the melt is obtained assuming that the sharp-interface is diffused over the length of two finite elements. The methodology presented is shown to be an effective tool for capturing the interface phenomena and freezing interface growth using a single uniform finite element grid. The whole formulation is packaged into a flexible, modular and parallel library with the ability to incorporate new physics. Comparisons with other numerical methods as well as analytical results emphasize the fidelity of the method in modelling the underlying physical phenomena and growth mechanisms. Various examples of dendritic growth in two- and three-dimensions are presented.

Key words: Extended finite element method, Crystal growth, Dendritic solidification, Convection, Level set method

---

[1] Corresponding author: N. Zabaras, Fax: 607-255-1222, Email: zabaras@cornell.edu, URL: http://mpdc.mae.cornell.edu/

Preprint submitted to Elsevier Preprint *6 March 2006*

1 Introduction

Modelling of dendritic growth in the solidification of pure metals and alloys remains a significant challenge in materials science and applied physics. Successful modelling of dendritic solidification requires both the solution of a complex free-boundary problem and an accurate account of the surface tension and kinetic anisotropy. The first task is rendered extremely difficult by the several orders of magnitude disparity of length scale between the thickness of the diffusion boundary layer of heat/solute that surrounds the dendritic tip and the microscopic capillary scale, while the second task is complicated by the need to accurately compute the curvature of the phase interface.

The various techniques for the numerical simulation of dendrites can in principle be divided into two basic groups - explicit interface tracking and the diffused-interface approach. The explicit interface tracking method includes the (conceptually) older adaptive grid techniques as well as the more recent level set and extended finite element methods. The phase field method along with its variants constitute the diffused-interface approach to modelling dendritic growth.

The phase field method belongs to a group of methods that rely on treating a microscopically sharp-interface as a diffused-region immersed in the calculation domain. The phase field variable Φ is introduced that varies smoothly from zero to unity between the two phases over the diffused-interface region, which has a small but numerically resolvable thickness. This variable serves to distribute the interfacial forces and other source terms over the diffused-interface region. The phase field method derives its attractiveness from the fact that explicit tracking of the interface and satisfaction of the interfacial boundary conditions are avoided. Further, the computation of interface normals and curvature is also avoided by solving a certain evolution equation for the phase field variable. This evolution equation can be derived from the thermodynamics of phase transition. An additional feature of the evolution equation for the phase field is that it contains an explicit anti-diffusivity that maintains the thin and well defined interface region without introducing spurious oscillations or violating conservation of mass. The phase field method can be shown to reduce to the standard sharp-interface formulation in the limit of vanishing interface thickness [1]. The computational complexity of the problem reduces by increasing the interface thickness but the solution of the phase field method deteriorates with increasing interface thickness. This necessitates the grid spacing to be the order of or smaller than the interface thickness. Wheeler et al. [2] and Wang et al. [3] have shown that the interface thickness must be smaller than the capillary length for the solution to converge to the sharp-interface limit. Karma and Rappel [4] derived improved asymptotic coefficients for the thin-interface limit of the phase field equations, which improves the convergence of the method for a coarser grid density. The restriction of grid size is generally not a problem for the symmetric model of solidification (where the diffusivity of the solid and liquid phases are assumed to be equal), however phase-field asymptotics for unequal diffusivities lead to computationally complex forms which require extra grid resolution and hence slower computational performance [5]. One other potential drawback of the phase field method is the significant computational effort required, especially when investigating dendritic growth in the presence of convection and multiple array dendritic growth. However, the techniques of adaptive gridding and parallel computing have resulted in assuaging this drawback to some extent [6]. Another potential drawback of the phase field methods is that there is a large number of parameters that are involved in the solution of the evolution equations. Their determination for accurate physical simulation of real world materials is rather difficult.

The growth of dendrites usually leads to a change in topology, the formation of sharp corners and singularities as well as merging and breaking of crystal structures. It is in the context of modelling such behavior that the level set methods were developed. These techniques work by embedding the propagating interface as the zero level set of a time dependent, implicit function, and then solving the resulting equations of motion in a fixed grid (Eulerian setting) [7,8]. This approach allows one to track the motion of very complex interfaces [9–15]. One of the main difficulties associated with the use of level set methods is solving the front propagation equation. The solution to this partial differential equation need not be differentiable, even with smooth boundary data. In addition, the method requires the accurate computation of the interface normals and curvature.

Another approach for representing localized behavior using finite elements is via using the partition of unity method [16]. The main idea of this method is to extend the classical finite element approximation by augmenting the set of nodal shape functions with products of a subset of these same shape functions and local enrichment functions. The extended finite element method (X-FEM) models arbitrarily evolving geometric features with evolving enrichment functions. This results in good approximations of functions with arbitrary discontinuities and discontinuous derivatives which are independent of the finite element mesh [17]. This is particularly useful for evolution problems with moving discontinuities such as crack propagation and solidification [18–21]. The major appeal of these methods for incorporating discontinuities in finite elements is that they do not require the mesh to conform to the discontinuities in the approximating function or its derivatives. They also avoid remeshing for moving discontinuities. The recent coupling of the extended finite element method with the level set method to track the interface has combined the advantages of explicitly tracking the interface with a high accuracy and maintaining a single uniform finite element grid. The extended-FEM/level set combined methodology has already been applied to simple phase change and solidification problems in [22–24].

Dendritic microstructures generally form because of the constitutional undercooling in alloys and pure melts. There is enough experimental evidence to suggest that fluid flow can have a significant effect on the morphology of the microstructure [25]. Melt convection adds new length and time scales to the problem and results in morphologies that are potentially very different from those generated purely by diffusive heat and solute transport. Further, the evolving microstructure can also result in new and possibly complicated flow phenomena. Phase field methods have been used extensively to simulate the growth of two-dimensional dendrites with melt flow [26,27]. Other dendritic growth finite element based simulations with melt flow including front-tracking techniques are presented in [28–32].

In the present work, a hybrid X-FEM/level set method is used to simulate the growth of dendrites for pure materials in the presence of melt convection. A dimension-independent formulation is described that extends the conventional X-FEM approximation to include the effects of melt convection. The issue of numerical stabilization of the resulting convection-diffusion thermal problem is addressed. Techniques implemented for efficient computation and optimization at various stages of the numerical scheme are discussed. Domain decomposition of the computational domain for parallel computing is described. A highly modular, efficient and flexible software library has been developed that ensures the ease of incorporating new physics and investigating other related solidification phenomena. A single regular grid is used for the entire simulation. To the best of our knowledge, this is the first reported effort in applying the extended finite element method to dendritic solidification that includes melt flow effects for both two- and three-dimensional solidification growth.

The plan of the paper is the following. Section 2 briefly introduces the problem of interest, while Section 3 summarizes the relevant fundamentals of the X-FEM method. Section 4 provides the algorithmic developments for modelling the transport phenomena and interface growth conditions and summarizes the implementation of the fast marching scheme for re-initialization and velocity extension. Section 5 considers a number of benchmark examples in order to allow us to evaluate the accuracy and efficiency of the X-FEM approach. Some conclusions are finally drawn in Section 6.

2 Governing equations

Consider the domain Ω as shown in Fig. 1 filled with a pure material. At every time step and in each location in the domain Ω, the material is either solid or liquid. The region where the material is solid is represented as Ω_s while the liquid region is represented as Ω_l ($\Omega = \Omega_s \cup \Omega_l$). The interface between the solid and liquid phases is denoted by Γ_{int}. The outward normal to the interface from Ω_s is denoted by $\boldsymbol{n}_s$. The temperature field is represented as $T(\boldsymbol{x}, t)$. To describe the interface, we construct a field ϕ such that at any time t, the interface is equal to the zero level set of ϕ, i.e.

$$\Gamma_{int}(t) = \{\boldsymbol{x} \in \Omega \ : \ \phi(\boldsymbol{x}, t) = 0\}\,. \tag{1}$$

The field ϕ is here taken as a signed distance function $\phi(\boldsymbol{x}, t)$ defined as follows:

$$\phi(\boldsymbol{x}, t) = \min_{\boldsymbol{y} \in \Gamma_{int}} \parallel \boldsymbol{x} - \boldsymbol{y} \parallel sign(\boldsymbol{n}_s \cdot (\boldsymbol{x} - \boldsymbol{y})), \tag{2}$$

where $\phi(\boldsymbol{x}, t)$ is positive in the liquid region, negative in the solid region and zero at the interface.

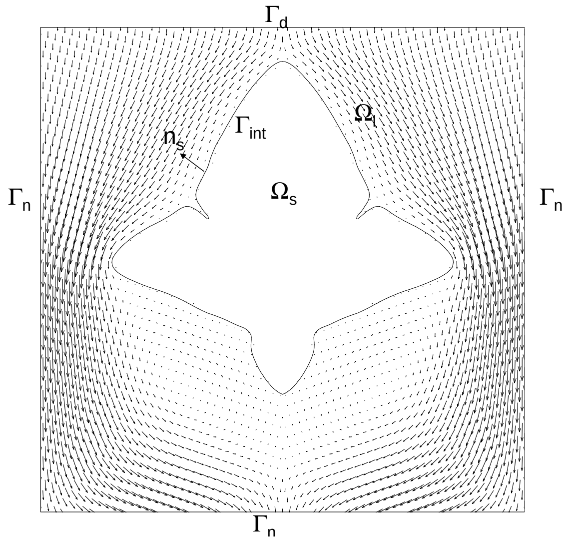

Fig. 1. Problem schematic and nomenclature for dendritic solidification growth in an undercooled melt.

In the cases where fluid flow is considered, the fluid enters the domain from the top with a velocity V_{in} and leaves the domain at the bottom (see Fig. 1). On the bottom boundary, where the fluid leaves the domain, a zero pressure condition is applied. No slip boundary conditions are imposed on the remaining sides of the domain.

The governing equations for the temperature evolution are:

$$\frac{\partial T}{\partial t} + \boldsymbol{v}.\nabla T = \alpha_l \nabla^2 T, \qquad \boldsymbol{x} \in \Omega_l, \tag{3}$$

$$\frac{\partial T}{\partial t} = \alpha_s \nabla^2 T, \qquad \boldsymbol{x} \in \Omega_s, \tag{4}$$

where α_s and α_l denote the thermal diffusivities of the solid and liquid phases, respectively. The conditions on the boundary Γ $(= \Gamma_d \cup \Gamma_n)$ of Ω are

$$T = T_d, \qquad \boldsymbol{x} \in \Gamma_d, \tag{5}$$

$$k_l \nabla T \cdot \boldsymbol{n} = q, \qquad \boldsymbol{x} \in \Gamma_n. \tag{6}$$

In most of the cases examined, the heat flux q is set to zero. The initial temperature of the melt is taken to be at an undercooling (with respect to the melting temperature T_m) denoted by ΔT. The jump condition (Stefan condition) on the solid-liquid interface is given by:

$$LV = k_s \frac{\partial T_s}{\partial n} - k_l \frac{\partial T_l}{\partial n}, \tag{7}$$

where V is the interface velocity (normal component to the interface), L is the latent heat of fusion and k_s, k_l represent the thermal conductivities of the solid and liquid phases, respectively. The temperature at the interface also satisfies the Gibbs-Thompson relation

$$T = T_m - \varepsilon_c \kappa - \varepsilon_v V, \tag{8}$$

where κ is the curvature of the propagating interface, ε_c is the surface tension coefficient and ε_v is the kinetic mobility coefficient. The parameters ε_c and ε_v are usually anisotropic.

A diffused-interface model is used in the solution of the momentum equations. The interested reader can refer to [27] for a discussion regarding the accuracy of the solution obtained with such models for various values of the interfacial thickness. Instead of considering the sharp-interface explicitly tracked by $\phi(\boldsymbol{x})$, the interface is smeared out over the region $\phi(\boldsymbol{x}) \in [-w, w]$, where in this work, w is usually taken as the size of a typical finite element used in the discretization of the problem. This diffused-interface is characterized by the liquid volume fraction Φ. In the liquid region Φ is 1, while in the solid region

Φ is set to zero. The function $\Phi(\boldsymbol{x})$ is defined as follows:

$$\Phi(\boldsymbol{x},t) = \begin{cases} 1, & \phi(\boldsymbol{x},t) \geq w, \\ 0, & \phi(\boldsymbol{x},t) \leq -w, \\ \phi(\boldsymbol{x},t)/2w + 0.5, & \phi(\boldsymbol{x},t) \in [-w,w]. \end{cases} \tag{9}$$

Following the approach summarized in Zabaras and Samanta [33], the volume-averaged momentum and continuity equations (applied to the whole domain Ω) are given as follows:

$$\frac{\partial \boldsymbol{v}}{\partial t} + \nabla \cdot \left(\frac{\boldsymbol{v}\boldsymbol{v}}{\Phi}\right) = -\nabla p + \frac{p}{\Phi}\nabla\Phi + \nabla \cdot [Pr(\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T)] - \frac{(1-\Phi)^2}{\Phi^2}\frac{Pr}{Da}\boldsymbol{v}, \tag{10}$$

$$\nabla \cdot \boldsymbol{v} = 0, \tag{11}$$

where $\boldsymbol{v}$ from now on is the volume-averaged velocity equal to $\Phi < \boldsymbol{v}_l >^l$, with $< \boldsymbol{v}_l >^l$ the intrinsic volume-averaged melt velocity. Also, Pr is the Prandtl number ($Pr = \frac{\nu_l}{\alpha_l}$) and Da is the Darcy number given by $Da = K_o/{L_s}^2$ where K_o is the permeability constant in the Kozeny-Carman equation and L_s is the characteristic length of the system. For simplicity of the model, the densities of the two phases are taken the same and buoyancy effects are considered negligible. For a complete list of publications related to the derivation of the above equations, the interested reader can refer to [33].

Remark 1. The Kozeny-Carman model used in Eq. (10) was introduced as a matter of convenience due to its widespread use in macroscopic solidification simulators. As our main emphasis is in modeling sharp-front solidification problems, other models of diffused mushy zones can be used as well without significant changes in the computed solutions (see Section 5.5). Such models may include the variable viscocity approach advocated in [26].

3 Extended finite element method

In the standard finite element formulation for Eqs. (3)-(4), the basis functions are at least C^1 continuous within elements. The need for enrichment comes from representing the temperature gradient discontinuity given in Eq. (7) inside elements. Such discontinuities are arbitrary with respect to the fixed grid. The standard FEM is modified by locally extending the basis functions and approximating the solution as follows [16]:

$$T(\boldsymbol{x}) = \sum_{i=1}^{n} N_i(\boldsymbol{x}) T_i + \sum_{j=1}^{n^E} \psi_j(\boldsymbol{x}) a_j, \tag{12}$$

where $N_i(\boldsymbol{x})$ are the standard finite element shape functions for the node i, T_i are the nodal degrees of freedom, $\psi_j(\boldsymbol{x})$ are the enrichment functions and a_j are the nodal enrichment degrees of freedom. Here, n and n^E are the number of nodal standard degrees of freedom and nodal enriched degrees of freedom, respectively.

We will briefly review this formulation with more details provided in [16–20]. The enriched elements are defined as the elements that are cut by the interface as determined by the values of the level set function. A node is enriched if at least one of the edges containing it is intersected by the interface. The set of enriched nodes is just the union of the enriched nodes of all enriched elements. Since the interface is evolving with time, the set of enriched elements and nodes is also changing with time. The schematic of enrichment is shown in Fig. 2. To each of the enriched nodes we associate the additional degrees of freedom $\{a_j\}$. The mapping between the enriched degree of freedom j and the standard degree of freedom i is denoted here as $i(j)$ and defined as $i = i(j) = \{$the standard numbering of the node (dof) whose enriched numbering is $j\}$. The enrichment shape function ψ_j is defined for each enriched node j. It is usually represented as $N_{i(j)}(\boldsymbol{x}) g_{i(j)}(\boldsymbol{x})$, where $g_{i(j)}(\boldsymbol{x})$ is the enrichment kernel [17].

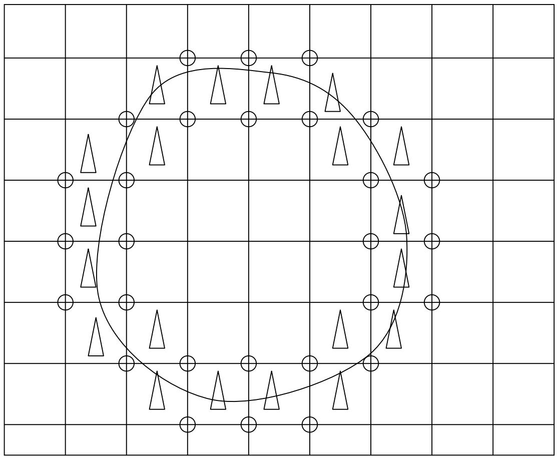

Fig. 2. The definition of enriched nodes (○) and enriched finite elements (△).

The functions ψ_j are required to be linearly independent with each other and with the standard finite element basis. The corresponding derivatives of T are given by [23],

$$\nabla T(\boldsymbol{x}) = \sum_{i=1}^{n} \nabla N_i(\boldsymbol{x}) T_i + \sum_{j=1}^{n^E} \nabla \psi_j(\boldsymbol{x}) a_j, \tag{13}$$

where,

$$\nabla \psi_j(\boldsymbol{x}) = \nabla N_{i(j)}(\boldsymbol{x}) g_{i(j)}(\boldsymbol{x}) + N_{i(j)}(\boldsymbol{x}) \nabla g_{i(j)}(\boldsymbol{x}). \tag{14}$$

A discontinuity can be represented by constructing $g_i(\boldsymbol{x})$ such that $\nabla g_i(\boldsymbol{x})$ is discontinuous inside elements. A popular way to construct $g_i(\boldsymbol{x})$ such that the derivative of $g_i(\boldsymbol{x})$ is discontinuous is [24]

$$g_i(\boldsymbol{x}) = |\phi(\boldsymbol{x})| - |\phi_i|, \tag{15}$$

$$\nabla g_i(\boldsymbol{x}) = \mathrm{sign}(\phi(\boldsymbol{x})) \nabla \phi(\boldsymbol{x}), \tag{16}$$

where $\phi(\boldsymbol{x})$ is the level set field and ϕ_i is the nodal value of the level set field. An obvious advantage of this construction is the zero contribution of the enrichment to the nodal temperature values. As a result, the nodal value of the temperature field after enrichment is still the standard nodal degree of freedom. For the derivation of the finite element equations for the thermal problem, it is convenient to write the enriched approximations as

$$T(\boldsymbol{x}, t) = \sum_{i}^{n} \hat{N}_i(\boldsymbol{x}) \hat{T}_i(t), \tag{17}$$

or in matrix form

$$T(\boldsymbol{x}, t) = \hat{\mathbf{N}}^T(\boldsymbol{x}) \hat{\mathbf{T}}(t), \tag{18}$$

where

$$\hat{\mathbf{N}}^T(\boldsymbol{x}) = [N_1, N_2, \ldots, N_n, \psi_1, \psi_2, \ldots, \psi_{n^E}], \tag{19}$$

$$\hat{\mathbf{T}}^T(t) = [T_1, T_2, \ldots, T_n, a_1, a_2, \ldots, a_{n^E}]. \tag{20}$$

The level set ϕ and the velocity $\boldsymbol{v}$ are discretized by the standard finite element basis so that

$$\phi(\boldsymbol{x}, t) = \sum_{i=1}^{n} N_i(\boldsymbol{x}) \phi_i(t), \tag{21}$$

$$\boldsymbol{v}(\boldsymbol{x},t) = \sum_{i=1}^{n} N_i(\boldsymbol{x})\boldsymbol{v}_i(t). \tag{22}$$

Note that in this work, enrichment is only used in the solution of the thermal problem to allow us to model the interface discontinuity in heat flux and thus accurately compute the interface growth conditions. While enrichment can be used in a front-tracking approach to the solution of the momentum equations that enforces the no slip conditions on Γ_{int}, we have selected to use the single-domain volume-averaged approach (Eq. (10)) to avoid an unnecessary increase in the cost of the simulation without sacrificing the accuracy of the solution.

Remark 2. In the work presented in this paper, we only use structured finite element grids for the solution of the temperature and flow fields. However, note that in solving the level set sub-problem (see Section 4.1) we do have a choice of using the fast marching method (using an overlapped triangular mesh) or the ODE-based re-initialization scheme (using a structured grid).

4 Numerical implementation

4.1 Computing the level set function ϕ

The level set variable ϕ is used to explicitly track the interface Γ_{int} as well as to compute the enrichment kernel $g_i(\boldsymbol{x})$. Initially, ϕ is set equal to the signed distance function from the interface Γ_{int}. The idea behind the level set method is to move ϕ with the correct speed V at the interface, which is extracted from Eq. (7). The equation of motion governing ϕ is given as

$$\phi_t + F|\nabla\phi| = 0, \tag{23}$$

where F is the extension of the interface velocity V (see Section 4.2). This equation moves ϕ with the correct speed at the interface so that Γ_{int} will always be equal to the zero level set of ϕ. To solve Eq. (23), a Galerkin/least-squares (GLS) finite element scheme is used [24]. Using standard notation, it takes the form:

$$\begin{aligned} &\mathop{\mathbf{A}}_{e=1}^{N_{el}} \int_{\Omega_e} N^e \left(\frac{\partial\phi}{\partial t} + F|\nabla\phi| \right) d\Omega_e \\ &+ \mathop{\mathbf{A}}_{e=1}^{N_{el}} \int_{\Omega_e} \left(\frac{F}{|\nabla\phi|} \nabla N^e \cdot \nabla\phi \right) \tau_\phi^e \left(\frac{\partial\phi}{\partial t} + F|\nabla\phi| \right) d\Omega_e = 0, \end{aligned} \tag{24}$$

where N^e is the shape function for element e and the stabilization parameter τ_ϕ^e is defined as

$$\tau_\phi^e = \frac{\epsilon h_e}{\mid F \mid},$$

where h_e is the length of the element e and ϵ is a parameter that is set to a value of 10^{-3} in all the calculations presented here. Finally, N_{el} is the number of elements in the discretized domain.

Equation (24) leads to the following linear system:

$$\mathbf{A}\dot{\phi} = \mathbf{b}. \tag{25}$$

The elemental contributions of $\mathbf{A}$ and $\mathbf{b}$ are given as:

$$A_{ij}^e = \int\limits_{\Omega_e} \left(N_i^e + GLS_i^e\right) N_j^e \, d\Omega_e, \tag{26}$$

$$b_i^e = -\int\limits_{\Omega_e} \left(N_i^e + GLS_i^e\right) F|\nabla\phi| \, d\Omega_e, \tag{27}$$

where N_i^e is the local elemental shape function at node i and GLS is the Galerkin least squares weighting function defined as

$$GLS_i^e = \tau_\phi^e \nabla N_i^e \cdot \frac{\nabla\phi}{|\nabla\phi|} F. \tag{28}$$

We finally note that in the implementation of the level set algorithm discussed here (Eq. 24), calculations were performed in only a small portion of the domain Ω (narrow band around the interface). Section 4.2 elaborates further on this topic.

Due to the complexity of dendritic solidification problems, the computed level set function will eventually deviate from the signed distance function. So after computing the temperature and level set fields for several time steps, a re-initialization step is required to restore the level set function to signed distance while keeping its zero level set contour undisturbed. These techniques are reviewed next.

4.2 Re-initialization and velocity extension

Re-initialization techniques based on the finite difference method are described in detail in [7,8]. In an effort to make our developments suitable for coupling with FEM-based solvers for heat transfer and fluid flow, we provide a simple extension of the fast marching method to finite element grids and use it to implement the re-initialization procedure. For convenience, we will only emphasize re-initialization to the positive region $\phi > 0$. Re-initialization to the negative region $\phi < 0$ follows a similar methodology.

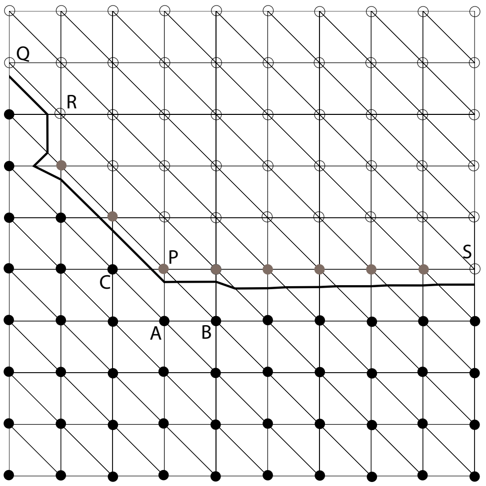

Fig. 3. Fast marching scheme for an unstructured grid in two dimensions.

In the fast marching method, each node is given a tag. There are three types of tags: 'alive', 'band' and 'far-away'. These tags are given based on the distance of the node from the interface. This is shown schematically in Fig. 3 where the 'alive', 'band' and 'far-away' nodes are shown as black, gray and empty circles, respectively. For a detailed discussion on the nomenclature and methodology used in the fast marching technique, the interested reader is referred to [7]. Briefly, an 'alive' node implies that ϕ has already been re-initialized on the node and can be used to re-initialize other nodes. Nodes adjacent to 'alive' nodes, which in addition have all their neighbors within an element marked 'alive', are ready to have their values computed and are marked as 'band' nodes. Other nodes are simply marked as 'far-away'.

The essential idea of the method is " .. to sweep the front ahead in an upwind fashion by considering a set of points in a narrow band around the existing front, and to march this narrow band forwards, freezing the values of existing points and bringing new ones into the narrow band. The key is in the selection of *which* grid point in the narrow band to update" (pp. 88, [7]).

In each iteration, the 'band' node with the smallest value of ϕ (since ϕ determines the distance to the interface) will have its value reinitialized. The fast marching method uses a balanced heap data structure to store all the 'band' nodes with the node that has the smallest ϕ always at the root. Details of implementation of heap data structures are given in [7,8].

When we reinitialize the value at a 'band' node P, there could be more than one element, which has all nodes 'alive' except P (e.g. elements (A, B, P) and (A, C, P) shown in Fig. 3). We define as *upwind element* of a 'band' node, the element with the smallest average ϕ over all its 'alive' nodes. This upwind element is used to reinitialize the value to the 'band' node. For example, if $\phi_A + \phi_B < \phi_A + \phi_C$, then element (A, B, P) in Fig. 3 will be selected as upwind element to extend value to node P.

Application of the fast marching method involves estimating the level set variable for a node from its 'alive' neighbors. Previous work [8] describes how to apply this extrapolation for structured grids. For finite element grids, the problem boils down to estimating ϕ_P on node P from values at 'alive' nodes of its upwind element (e.g. ϕ_A on node A and ϕ_B on node B for the 2D case shown in Fig. 3). In the following derivations, we discuss the formulation of extending values for the 2D case with similar calculations applied to 3D problems. Consider a node P marked as 'band', with two nodes upwind to it (A and B), marked 'alive'. Using the gradient of ϕ (computed at node P), $\nabla\phi = (\phi_{,x}, \phi_{,y})^T$, in this upwind element, one can write the following:

$$\begin{pmatrix} \Delta x_A & \Delta y_A \\ \Delta x_B & \Delta y_B \end{pmatrix} \begin{pmatrix} \phi_{,x} \\ \phi_{,y} \end{pmatrix} = \begin{pmatrix} \Delta\phi_A \\ \Delta\phi_B \end{pmatrix},$$

where $\Delta\phi_A \equiv \phi_A - \phi_P$, $\Delta x_A \equiv x_A - x_P$ and $\Delta y_A \equiv y_A - y_P$. For convenience, we re-define ϕ_P, ϕ_A, ϕ_B as ϕ, ϕ_1, ϕ_2 and introduce the matrix M as

$$M \equiv \begin{pmatrix} \Delta x_A & \Delta y_A \\ \Delta x_B & \Delta y_B \end{pmatrix}^{-1}.$$

In order to calculate ϕ (that is, ϕ_P), we take

$$\begin{pmatrix} \phi_{,x} \\ \phi_{,y} \end{pmatrix} = M \begin{pmatrix} \phi_1 - \phi \\ \phi_2 - \phi \end{pmatrix}.$$

According to the signed distance property of the level set function, the relation

$||\nabla\phi|| = 1$ holds, leading to the following:

$$\sum_{i=1}^{2}\left(\sum_{j=1}^{2} M_{ij}(\phi - \phi_j)\right)^2 = 1. \tag{29}$$

With the grid-related matrix M and ϕ_j known, Eq. (29) becomes a quadratic equation for computing ϕ. Generally, Eq. (29) will have two roots. In the positive region ($\phi > 0$), the larger root is used. Otherwise, the smaller root is used. Numerically, if the ϕ value near the interface deviates significantly from signed distance, the above quadratic equation may not attain any real root. Although this is not the case in any of the simulations reported in this paper, for such pathological cases we assume ϕ to be $\frac{\phi_1+\phi_2}{2}$ thus leading to an algorithm that is robust under any circumstances.

Finally, we note that the fast marching scheme requires a triangle mesh in 2D, or a tetrahedral mesh in 3D. For convenience of generating the computation mesh and using rectangular (2D) or regular hexahedral (3D) elements for X-FEM, overlapped meshes are used in our calculation. The level set related calculations (solving level set equation, re-initialization and normal/curvature calculation) are performed on a smaller triangular/tetrahedral grid generated by dividing each rectangular(2D)/ regular hexahedral (3D) element into triangular/tetrahedral elements. Since the level set evolution and update is specifically used to capture the dynamics of the interface growth, the computational domain for the level set related calculation includes only a narrow band around the interface [34]. This smaller domain includes elements within a specified distance from the interface. For example, Fig. 4 shows the narrow band used at time 15000 for the solvability case with under-cooling 0.55 discussed later in this paper (see Section 5.3). The thicker lines mark the boundaries of the narrow band. In all calculations reported in this work, the boundaries of the narrow band are defined by the $|\phi| = 15 \times \Delta x$ contour, where Δx is the element size of the rectangular grid.

4.3 Computing the temperature field T

The weak form of the temperature evolution applicable to the whole domain Ω is given as follows:

$$\frac{1}{\Delta t}\int_{\Omega} \delta T(T^{n+1} - T^n)d\Omega + \int_{\Omega} \delta T\boldsymbol{v}\cdot\nabla T d\Omega + \int_{\Omega} \nabla\delta T\cdot(\alpha\nabla T)d\Omega = \int_{\Gamma_n} \delta T q d\Gamma, \tag{30}$$

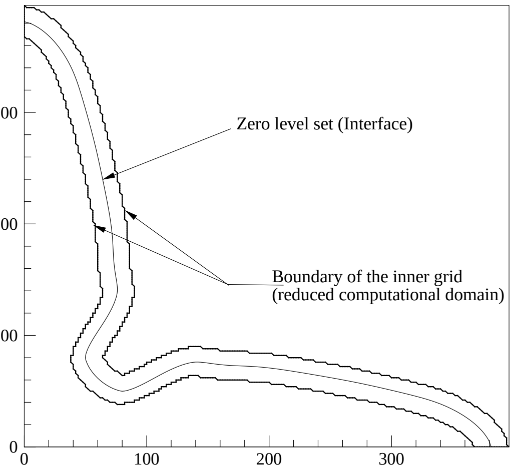

Fig. 4. Narrow band for level set calculation at time 15000 (solvability case with under-cooling 0.55).

where the diffusivity α takes the corresponding values of the solid/liquid phases depending on the location of the integration point with respect to the interface, the velocity $\boldsymbol{v}$ is zero in the solid phase and q is the imposed heat flux on the boundary Γ_n of the domain.

To account for convection-dominated thermal transport, an SUPG (streamline-upwind Petrov Galerkin) stabilization is introduced by taking δT_α as follows:

$$\delta T_\alpha = N_\alpha + \tau \nabla N_\alpha \cdot \boldsymbol{v}, \tag{31}$$

where the subscript α refers to the node number and τ is the stabilization parameter [33]. In the case of an enriched element, the enriched basis functions (for the enriched dof) are $N_\alpha g_\alpha$, where g_α is the enrichment kernel function. The test function for the enriched degrees of freedom takes the following form:

$$\delta T_\alpha = N_\alpha g_\alpha + g_\alpha \tau \nabla N_\alpha \cdot \boldsymbol{v} + \tau [N_\alpha \nabla g_\alpha] \cdot \boldsymbol{v}. \tag{32}$$

By substituting Eq. (18) in Eq. (30) and using Eqs. (31) and (32), the following discrete equations are obtained:

$$\frac{1}{\Delta t}\mathbf{M}\hat{\mathbf{T}}^{n+1} + \mathbf{K}\hat{\mathbf{T}}^{n+1} + \mathbf{C}\hat{\mathbf{T}}^{n+1} = \frac{1}{\Delta t}\mathbf{M}^{*}\hat{\mathbf{T}}^{n} + \mathbf{f}_q^{n+1}, \tag{33}$$

where

$$\mathbf{M} = \mathop{\mathbf{A}}_{e=1}^{N_{el}} \int_{\Omega_e} (\hat{\mathbf{N}} + \hat{\mathbf{N}}_{SUPG}^{n+1})^T \hat{\mathbf{N}}^{n+1} d\Omega_e, \tag{34}$$

$$\mathbf{M}^* = \mathop{\mathbf{A}}_{e=1}^{N_{el}} \int_{\Omega_e} (\hat{\mathbf{N}} + \hat{\mathbf{N}}_{SUPG}^{n+1})^T \hat{\mathbf{N}}^{n} d\Omega_e, \tag{35}$$

$$\mathbf{K} = \mathop{\mathbf{A}}_{e=1}^{N_{el}} \int_{\Omega_e} \alpha \nabla (\hat{\mathbf{N}}^{n+1})^T \nabla \hat{\mathbf{N}}^{n+1} d\Omega_e, \tag{36}$$

$$\mathbf{C} = \mathop{\mathbf{A}}_{e=1}^{N_{el}} \int_{\Omega_e} (\hat{\mathbf{N}} + \hat{\mathbf{N}}_{SUPG}^{n+1})^T \boldsymbol{v} \nabla (\hat{\mathbf{N}}^{n+1}) d\Omega_e, \tag{37}$$

$$\mathbf{f}_q = \mathop{\mathbf{A}}_{e=1}^{N_{el}} \int_{\Gamma_{n,e}} (\hat{\mathbf{N}}^{n+1})^T q^{n+1} d\Gamma_e, \tag{38}$$

where $\hat{\mathbf{N}}_{SUPG}$ is the SUPG part of the test function, defined along the lines of $\hat{\mathbf{N}}$ (cf Eq. (18)). The system to be solved is a linear system

$$\mathbf{A}\hat{\mathbf{T}}^{n+1} = \mathbf{b}, \tag{39}$$

where

$$\mathbf{A} = \frac{1}{\Delta t}\mathbf{M} + \mathbf{K} + \mathbf{C}, \tag{40}$$

$$\mathbf{b} = \frac{1}{\Delta t}\mathbf{M}^* \hat{\mathbf{T}}^n + \mathbf{f}_q^{n+1}. \tag{41}$$

Since the geometry of the interface is independent of the finite element mesh, it becomes necessary to modify the quadrature routines for the volume integrals given in Eqs. (34)-(37). For elements intersected by the interface, the classical quadrature rules may not be accurate enough to capture the discontinuities and the change in material properties across the interface [17,35]. An approach similar to that used in [23] is followed in this work by using a subdivision of the enriched elements into quadrilaterals/hexahedrals. In n dimensions ($n = 2, 3$), we divide the element that is cut by the interface into r^n smaller quadrilaterals/hexahedrals. Extensive numerical investigations revealed that a value of $r = 10$ is more than sufficient to accurately model the discontinuities in 2 dimensions while a value of $r = 6$ was found sufficient for 3 dimensional problems. This amounts to 400 Gauss quadrature points per enriched element in 2D and 1728 Gauss quadrature points per enriched element in 3D.

To enforce the interface temperature condition given in Eq. (8), we exploit the fact that for bi/tri-linear finite elements, the temperature is linearly distributed along any interface segment. The constraint condition is enforced at the points where the interface intersects the element boundaries (see Fig. 5 for the 2D case). For each intersection point I of the interface with an element boundary, we can write the interface temperature condition as:

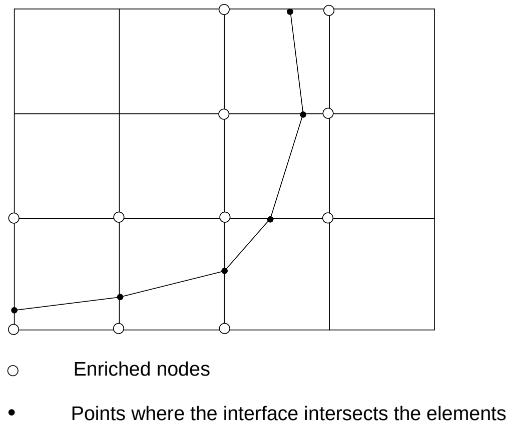

Fig. 5. Enforcing the interface temperature constraint.

$$\mathbf{G}_I^T \hat{\mathbf{T}} - T_I^m = 0 \tag{42}$$

where T_I^m is the melting temperature at point I and $\mathbf{G}_I$ is the column vector consisting of the basis functions evaluated at I. In fact, $\mathbf{G}_I^T \hat{\mathbf{T}}$ is the interpolation of T at the intersection point I using the nodal values (T_i, a_i). Then the constraint for the whole interface can be formulated as

$$\mathbf{G}\hat{\mathbf{T}} - \mathbf{T^m} = \mathbf{0} \tag{43}$$

where $\mathbf{G}$ is the matrix with ith row equal to $\mathbf{G}_i^T$ and $\mathbf{T^m}$ is the column vector with ith row equal to T_i^m. Incorporating this constraint with a penalty formulation in Eq. (39) results in the following:

$$(\mathbf{A} + \lambda \mathbf{G}^T \mathbf{G})\hat{\mathbf{T}} = \mathbf{b} + \lambda \mathbf{G}^T \mathbf{T^m} \tag{44}$$

where λ is the penalty parameter. The penalty enforcement is implemented once one determines the points where the interface intersects the element edges. At these points, the enriched shape functions are evaluated and then used to compute $\mathbf{G}$. In two dimensions, the interface intersects a quadrilateral grid at two and only two points (assuming it is not coincident with an edge). In three-dimensions, the two-dimensional subdomain of intersection of a regular hexahedral element with the interface could have 3, 4 or 5 points of intersection with the element edges. This calculation is implemented by looping over pairs of nodes and comparing the nodal level set values for a change in sign. A change in sign would represent a cutting point in between these nodes.

After the temperature update has been performed for a time step, the evolution of the interface requires computing the interface heat fluxes. The speed at which the front propagates is given by Eq. (7). One method used to evaluate V involves a simple jump calculation [12]. Consider a point $\boldsymbol{x}_d$ on the interface. Two points $\boldsymbol{x}_s$ and $\boldsymbol{x}_l$ are chosen on either side of the point x_d such that x_s lies on the solid side and x_l on the liquid side:

$$\boldsymbol{x}_{s,\ l} = \boldsymbol{x}_d \pm \frac{\nabla\phi}{\parallel \nabla\phi \parallel}\delta x_n, \tag{45}$$

This calculation makes sure that the two points on either side of the interface lie on the normal (at $\boldsymbol{x}_d$) to the interface. The distance of these points to $\boldsymbol{x}_d$ is determined by the parameter δx_n. The temperatures T_s, T_l, T_d at the three points are evaluated using the enriched interpolation functions. As these temperatures are obtained using the enriched basis functions, the jump across the interval $2\delta x_n$ is implicitly calculated. Then the normal front velocity is given by

$$V = \frac{1}{L}[k_s \frac{T_s - T_d}{\delta x_n} - k_l \frac{T_d - T_l}{\delta x_n}]. \tag{46}$$

Extensive numerical investigations showed that taking δx_n to be in the range of $10\% - 20\%$ of the element size resulted in very good results for the front velocity. We have also implemented the domain integral method used in [36], however the implementation given above was shown to be more efficient and accurate within the developed framework. An alternative approach to derivative jump treatment can be found in [37].

4.4 Computing the melt velocity $\boldsymbol{v}$

As noted earlier, for the velocity evolution, the sharp-interface is considered to be diffused over two elements. We treat the diffused-interface as a narrow mushy-zone. Volume-averaging is then applied to the whole region. The velocity in the solid region is set to zero, so that no-slip condition is applied at the solid/liquid interface. The formulation is briefly summarized below with more details provided in [33]. The function spaces $S_{\boldsymbol{v}}$ and S_p as first defined as follows:

$$S_{\boldsymbol{v}} \equiv \{\boldsymbol{v} | \boldsymbol{v} \in L_2^{nsd}, \mathrm{div}\boldsymbol{v} \in L_2, \boldsymbol{v} = 0 \text{ on } \Gamma_{wall}\},$$
$$S_p \equiv \{p | p \in L_2, \int_\Omega p d\Omega = 0\}.$$

The classical Galerkin formulation for the flow problem can be then stated as: Find $\mathbf{V} \equiv \{\boldsymbol{v}, p_l\} \in S_{\boldsymbol{v}} \times S_p$ such that for all $\mathbf{W} = \{\boldsymbol{w}, q\} \in S_{\boldsymbol{v}} \times S_p$, $B(\mathbf{W}, \mathbf{V}) = L(\mathbf{W})$ holds, where

$$\begin{aligned} B(W,V) &= \int\limits_\Omega \boldsymbol{w} \cdot \Big(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \left(\frac{\boldsymbol{v}}{\Phi}\right) + \frac{(1-\Phi)^2}{\Phi^2}\frac{Pr}{Da}\boldsymbol{v}\Big) d\Omega - \int\limits_\Omega p \nabla \Phi \cdot \boldsymbol{w} d\Omega \\ &\quad - \int\limits_\Omega \Phi p \nabla \cdot \boldsymbol{w} d\Omega + \int\limits_\Omega Pr \nabla \boldsymbol{w} \cdot (\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T) d\Omega + \int\limits_\Omega q \nabla \cdot \boldsymbol{v} d\Omega, \\ L(W) &= -\int\limits_\Omega \boldsymbol{w} \cdot \Phi Pr Ra_T \theta \boldsymbol{e_g} d\Omega. \end{aligned}$$

where Ra_T is the thermal Rayleigh number used to denote buoyant convection effects. In our examples we do not consider buoyant effects, but include it in the formulation for generality. In the finite element implementation of the Navier-Stokes equations, stabilizing techniques are needed to accommodate equal order interpolation velocity-pressure elements. A stabilized FEM technique for porous media flows is presented in [33] and is briefly discussed below. After introducing a modified pressure space S'_p as follows:

$$S'_p \overset{\text{def}}{=} \{p | p \in H^1(\Omega) \int\limits_\Omega q d\Omega = 0\}, \tag{47}$$

the stabilized weak form is the following: Find $\mathbf{V} = \{\boldsymbol{v}, p\} \in S_{\boldsymbol{v}} \times S'_p$ such that for all $\mathbf{W} = \{\boldsymbol{w}, q\} \in S_{\boldsymbol{v}} \times S'_p$ the following holds:

$$B_{\text{stab}}(\mathbf{W}, \mathbf{V}) = L_{\text{stab}}(\mathbf{W}), \tag{48}$$

where:

$$\begin{aligned} B_{\text{stab}}(\mathbf{W}, \mathbf{V}) &= B(\mathbf{W}, \mathbf{V}) + \int\limits_\Omega \mathcal{F}(\boldsymbol{v}, p) \cdot \mathcal{G}(\boldsymbol{w}, q) d\Omega + \int\limits_\Omega \tau_5 \nabla \cdot \boldsymbol{v} \nabla \cdot \boldsymbol{w} d\Omega, \\ L_{\text{stab}}(\mathbf{W}) &= L(\mathbf{W}) \ + \ \int\limits_\Omega \{\frac{p}{\Phi} \nabla \Phi - \Phi \ Pr (Ra_T \ \theta) \mathbf{g}\} \cdot \mathcal{G}(\boldsymbol{w}, q) d\Omega, \end{aligned}$$

where we defined:

$$\mathcal{F}(\boldsymbol{v}, p) = \frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v}_* \cdot \nabla \left(\frac{\boldsymbol{v}}{\Phi}\right) + \nabla p + \frac{(1-\Phi)^2}{\Phi^2}\frac{Pr}{Da}\boldsymbol{v} - Pr \nabla^2 \boldsymbol{v}, \tag{49}$$

$$\mathcal{G}(\boldsymbol{w}, q) = \tau_1 \boldsymbol{v}_* \cdot \nabla \left(\frac{\boldsymbol{w}}{\Phi}\right) - \tau_2 \frac{(1-\Phi)^2}{\Phi^2}\frac{Pr}{Da}\boldsymbol{w} - \tau_3 Pr \nabla^2 \boldsymbol{w} + \tau_4 \nabla q, \tag{50}$$

with $\boldsymbol{v}_*$ a divergence-free velocity that in the implementation of Eq. (48) at a given time is usually taken as the known velocity at the previous time step. The particular values of the parameters $\tau_1, \ldots, \tau_5$ used in this work are given in [33].

4.5 Summary of the algorithm

The computational efficiency of the methodology is improved by taking into account the following:

1. Global linear system - Using local enrichment strategy, the total degrees of freedom (dof) include the standard dof and the local enriched dof. We give all grid points two degrees of freedom – the standard and the enriched. For the non-enriched nodes, the number of enriched degrees of freedom is set zero as essential boundary condition in the global system. The actual size during evaluation of the temperature is just the total number of DOF's (standard + enriched).
2. Parallel computing - With the aid of the PETSC library, the matrix system can be easily parallelized. The parallelized KSPGMRES solver is used for solving the assembled linear systems.
3. Predetermination of terms - On integration points, we need to evaluate the weights w, shape function $N(\boldsymbol{x})$, and its derivative $\frac{\partial N}{\partial x}$. These terms are only evaluated on all the Gauss points of the first element. Their values on other element's Gauss points are the same as the evaluated values on the corresponding Gauss points of the first element. For the enriched elements, the neighboring information (neighboring nodes and neighboring elements) is necessary to reconstruct the interface. This information can also be predetermined for all the elements so that no search is required in later calculations.

Once the initial conditions are set, the velocity, temperature and the level set variable are consecutively updated. A summary of the algorithm is given below:

1. Update temperature
    - Update the enrichment functions ψ
    - Enforce essential boundary conditions
    - Loop over all elements:
        - If element is not enriched: compute the sub-matrices
        - If element is enriched: Subdivide the element in r^{nsd} (see Section 4.3) parts and compute the sub matrices for each sub element
    - Enforce the penalty constraint
    - Assemble the global matrix and solve the linear system

- Calculate the normal front velocity and extend over the narrow-band

(2) Update level set
  - Enforce essential boundary conditions
  - Loop over all elements:
    - · compute the GLS factor
    - · compute the sub-matices
  - Assemble the global matrix and solve the linear system
  - Re-initialize the level set variable to the signed distance

(3) Update the melt velocity
  - Compute the volume fraction Φ
  - Enforce essential boundary conditions
  - Loop over all elements:
    - · compute the stabilizing terms
    - · compute the sub matrices
  - Assemble the global matrix and solve the linear system

5 Numerical examples

This section provides validation of the extended finite element methodology used here through comparisons with various analytically solvable solidification problems and other bench marked dendritic solidification problems. This is followed by examples investigating the effects of fluid flow and undercooling on the shape of the growing crystal.

5.1 Solidification in two-dimensional corner

Solidification in an infinite corner was simulated by solving the Stefan problem in one-quarter of a square region $[0,1] \times [0,1]$. The temperature boundary condition applied at the corner surface $x = 0,\ y = 0$ is $T_c = -1^oC$. The initial uniform liquid temperature is $T_{in} = 0.3^oC$ and the melting temperature is $T_m = 0^oC$. This problem was solved analytically by Rathjen and Jiji [38], for the case of equal thermal diffusivities in both phases. The analytical solution is given in the following form:

$$y^* = \left(\lambda^u + \frac{C}{x^{*u} - \lambda^u}\right)^{1/u}, \tag{51}$$

where $C = 0.159$, $u = 5.02$, $\lambda = 0.70766$, $y^* = y/\sqrt{4at}$ and $x^* = x/\sqrt{4at}$. The thermal diffusivity is here taken as $a = 1.0\ m^2/s$. The domain was discretized using a relatively coarse 50×50 grid. The computed non-dimensionalized front location (at $t = 0.5$) is compared with the analytical solution (dark circles) in Fig. 6. The extended finite element captures the phase transition accurately even with a coarse grid.

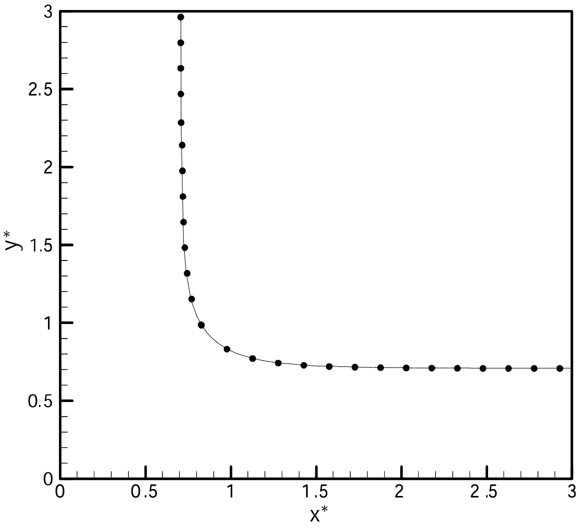

Fig. 6. Comparison of the non-dimensional front for solidification in a 2D corner.

5.2 *Rapid solidification of a circle: long-time behavior*

The temporal evolution of an interface separating two phases has been studied for its large time behavior by Caginalp [39]. Their study involved adaptation of the renormalization group and scaling theory to the phase transition problem. The characteristic length, R(t), governing the morphology of late stage growth was found to vary as $t^{1/2}$ in 3D and as t in 2D. This problem allows us to test the validity and accuracy of the formulation for large times. In the present example, the growth of a circular disc (initial radius 1, initial temperature 0^oC, melting temperature $T_m = 0^oC$) is considered in an undercooled melt. Since the theoretical trends are obtained for rapid solidification, the non-dimensional undercooling has a correspondingly large value of -0.5. Both surface tension and kinetic undercooling effects are included ($\varepsilon_c = -0.008$, $\varepsilon_v = -0.008$). The boundary temperature is maintained at temperature $T = -0.5$. Figure 7 plots the variation in the diameter of the growing disk with time. Notice that there is an initial period during which the growth is super-linear. In accordance to the theoretical predictions, after the initial growth period, the growth dynamics is linear with time.

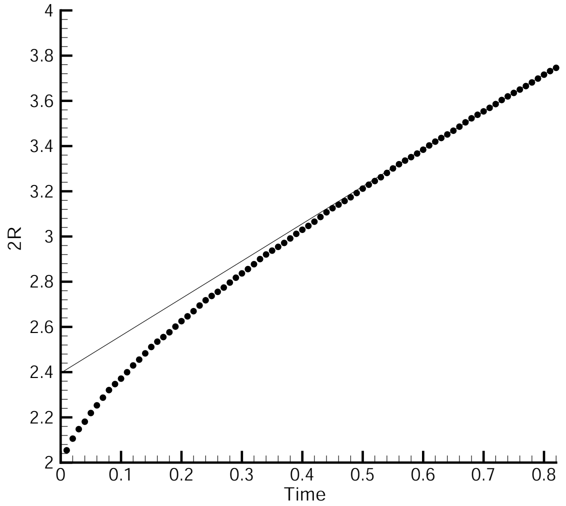

Fig. 7. Variation with time of the diameter of a rapidly solidifying cylinder.

5.3 Comparison with solvability theory

The sharp-interface Stefan problem has been studied extensively using microscopic solvability theory in determining the steady-state features of dendritic growth. According to solvability theory, the Stefan problem admits a family of discrete solutions, with the only stable solution corresponding to the fastest growing dendrite. This unique solution is also characterized by a unique tip shape and tip velocity. This problem has been solved using a level set method by Kim et al. [13] and has been extensively investigated using the phase field approach by Karma and Rappel [40] and by Provatas et al. [41]. The growth of a circular disc, with a four-fold growth axis of symmetry is simulated for comparison with the prediction of solvability theory. The grid considered is a 200×200 quadrilateral element grid. The computational domain is a square region of side length 800. The computational domain is taken to be one quarter of the physical domain because of symmetry considerations. The grid length for this simulation is $\Delta x = 2$ while the time step is $\Delta t = 50$. The initial radius of the seed is $R = 30$ and its temperature is set to 0. The undercooling is -0.55. The interface temperature is evaluated through the following formula:

$$T = -d_o \left(1 - 15\epsilon cos(4\theta)\right) \kappa, \tag{52}$$

where $\epsilon = 0.05$ is the anisotropic strength, θ is the angle made by the interface normal with the x axis and $d_o = 1.0$ is the capillary length scale. Time is nondimensionalized as $\tau = \frac{t\alpha}{d_o^2}$ while the tip velocity is nondimensionalized as $\tilde{V}_{tip} = \frac{V_{tip} d_o}{\alpha_l}$ (this non-dimensionalization is used everywhere else in this

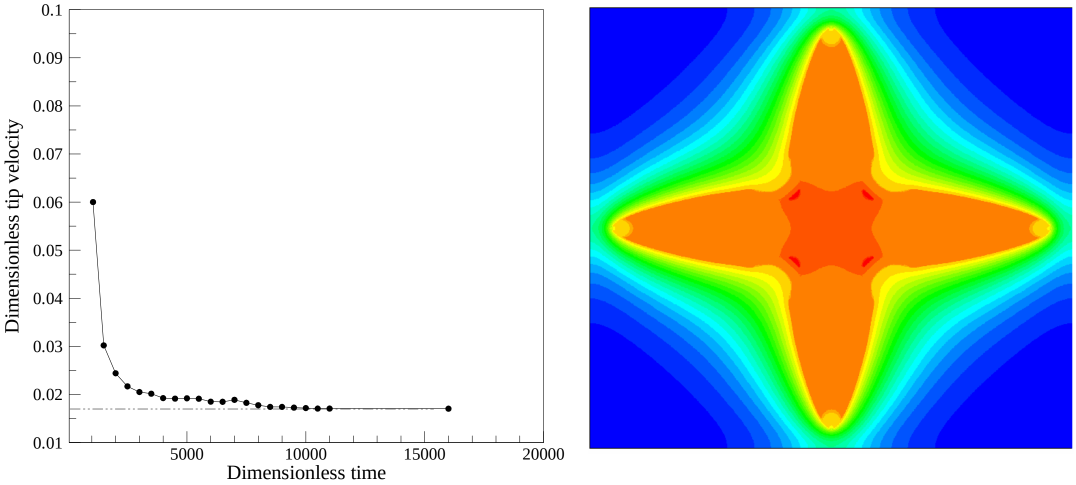

Fig. 8. On the left, variation of dimensionless tip velocity with time. On the right, temperature contours.

work, unless otherwise specified). The boundaries of the domain are all kept insulated. The growth velocity of the initially circular seed is around 0.06 (Fig. 8, left). It monotonically drops down and tapers off asymptotically after $\tau \approx 11000$. The tip velocity reaches a steady-state value of 0.017 as predicted by solvability theory. The steady-state temperature contours are shown in Fig. 8 (right). Notice that the tips of the dendrite have a lower temperature than the rest of the dendrites due to the curvature induced undercooling. Fig. 9 shows the final shape of the dendrite and the growth velocities. As predicted by solvability theory, the velocities of the dendrite tips are much larger than those on the rest of the dendrite body. Notice that the plot of the dimensionless tip velocity with respect to dimensionless time is not very smooth during the transition to the asymptotic limit. This is due in part to the formation of secondary dendritic tips along the main arm of the crystal that could lead to temperature fluctuations resulting in mild oscillations in the tip velocity. This issue is addressed next.

Provatas et al. [42] have simulated this solvability example using an adaptive grid based phase field model. In their results, the primary dendrite appears to grow symmetric secondary dendrites. Under the same growth conditions (but smaller time) Kim et al. [13] using a purely level set formulation failed to capture these secondary dendrites. The enriched finite element simulator was used to run an enlarged problem (three times the original domain, under the same conditions) to see if the methodology presented here could capture the secondary dendrites. The grid density for this problem is the same as the solvability case solved above. Figure 10 shows the steady state shape of the crystal in the extended domain. Our simulation proceeds to give symmetric secondary dendrites on all branches.

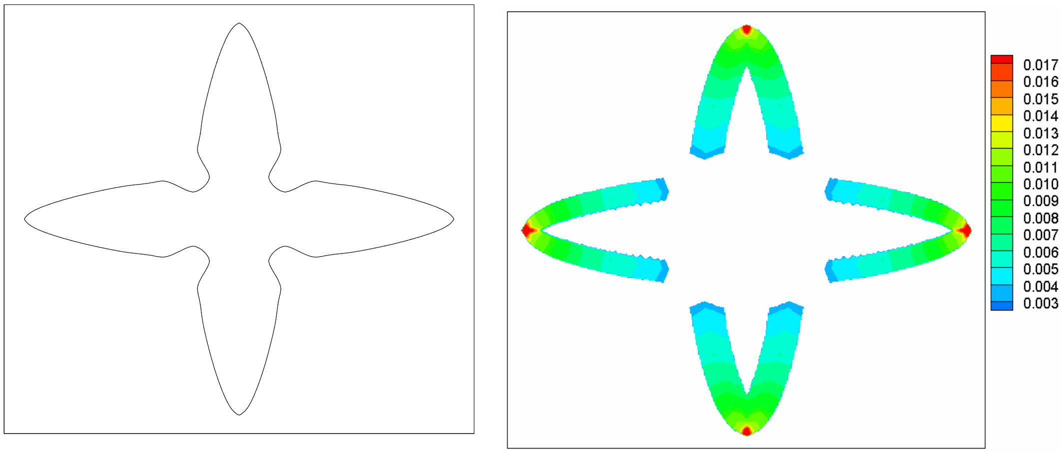

Fig. 9. On the left, the final shape. On the right, the tip velocities.

The reason for the suppression of the secondary dendrites could be due to the fact that extensive use of the re-initialization procedure (based on solving an ODE to steady state) used in [13] for the level set tends to smooth out any curvature irregularities. Changes in the curvature coupled with local temperature changes lead to the formation of secondary dendrites. The extensive application of re-initialization damps out the formation of secondary dendrites. In this sense, the fast marching technique does not diffuse out any curvature.

We also implemented the reinitialization procedure used in Kim et al. [13]. Comparing the reinitialization procedure (with the fast marching method) by running the same case gave interesting results. For fewer number of iterations (of the reinitializing procedure), it is possible to capture the secondary dendrites, but the shape loses some of its inherent symmetry. There is also some amount of instability in the simulator due to the lesser number of reinitializations. Increasing the number of iterations results in the dramatic suppression of any secondary dendrite formation.

5.3.1 Grid-convergence studies

As stated earlier, the solvability problem has an analytical result for the steady state tip velocity (0.017). The problem is also complex enough to serve as a benchmark study to investigate the effects of grid density. In this subsection, the variation in the shape of the growing solid and the steady state tip velocity is calculated for grid sizes 50, 100, 200, 300 and 350. Figure 11(a) shows the crystal shape. Notice that the shapes computed using the three finest meshes are indistinguishable. Figure 11(b) shows the steady state tip velocity variation with the grid size. The tip velocity obtained with a 200×200 grid is 0.017132, that obtained with a 300×300 grid is 0.017012, while the tip velocity is 0.017001 when computed using the 350×350 grid. Figure 12 plots the error

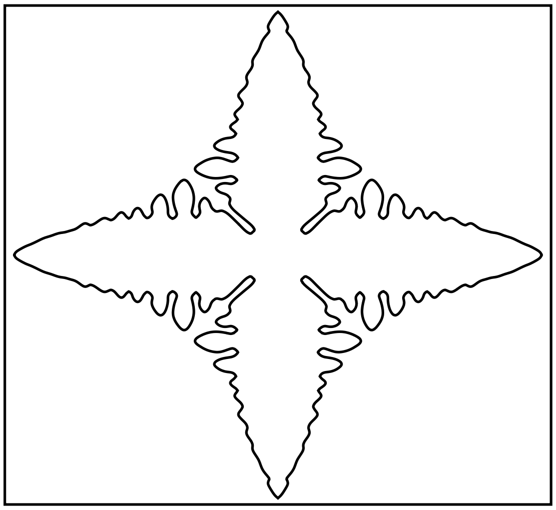

Fig. 10. Steady-state crystal shape obtained at time $\tau \approx 17500$ for a simulation in an extended domain.

(difference of the tip velocity from the analytical tip velocity) as a function of the grid size. The error plot (Fig. 12) reveals that the convergence of the interface tip velocity is super-linear with respect to the mesh size.

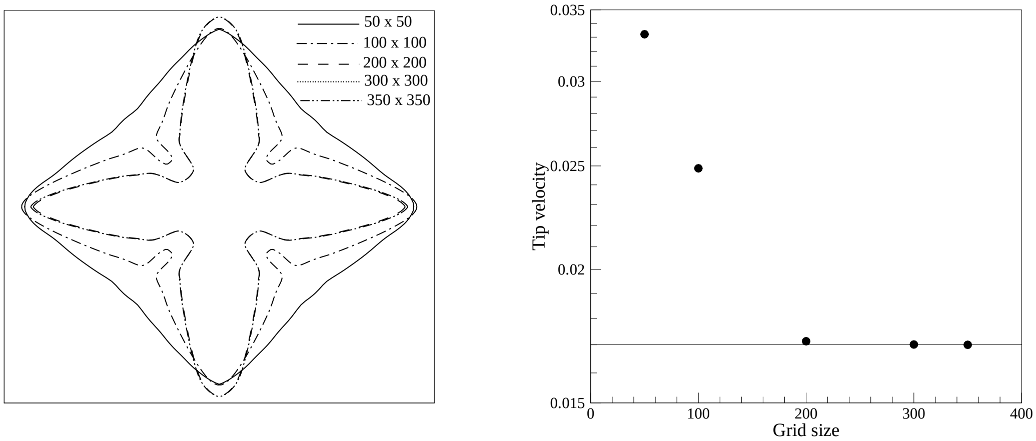

Fig. 11. (a) Crystal shape versus grid density, (b) Tip velocity versus grid density.

5.3.2 Grid-orientation effects

A simulation was conducted to demonstrate that artificial grid anisotropy is negligible in the present methodology. The solvability example is solved with an undercooling of -0.65 on a computational grid of 400×400 quadrilateral elements for two different sets of preferred directions given by the relation

$$T = -d_o\,(1 - 15\epsilon cos(4(\theta + \theta_s)))\,\kappa. \tag{53}$$

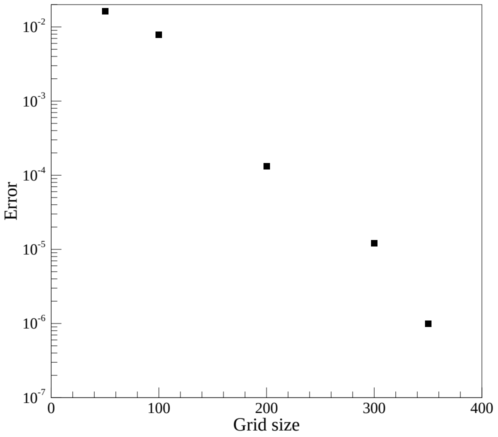

Fig. 12. Tip velocity error versus grid density.

Figure 13 shows the temperature contours for two values of θ_s, 0 and $\frac{\pi}{4}$ which correspond to rotating the coordinate axes by 0 and 45 degrees, respectively.

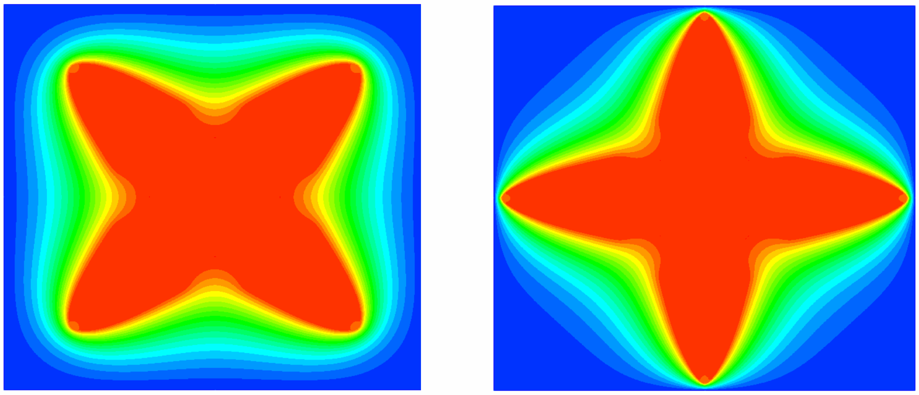

Fig. 13. Effect of grid anisotropy: rotation of the grid by 45^o ($\tau = 15000$).

5.3.3 Low-undercooling effects

Plapp and Karma [43] developed a novel methodology coupling a Monte-Carlo sampling technique with a conventional finite differences approach to simulate growth of a crystal under small under-cooling conditions. The same solvability problem as above but with undercooling of -0.30, in a region of quarter symmetry $[0, 300] \times [0, 300]$, is simulated to verify if the extended finite element methodology is able to capture the effects of small undercooling dynamics. The computational domain was divided into 200×200 quadrilateral elements. The temperature contours and interface position at dimensionless time $\tau = 5000$ are shown in Fig. 14. These results compare well with those presented in Plapp and Karma [43]. The next simulation considered, involved reducing the undercooling further to a value of -0.1. The simulation domain is taken to be a quarter square of dimensions $[0, 160] \times [0, 160]$. For this problem, the preferential anisotropy is removed (see [43] for more details). This leads to growth preferentially along the higher-harmonics. This can be seen in the temperature contours and interface position in Fig. 15.

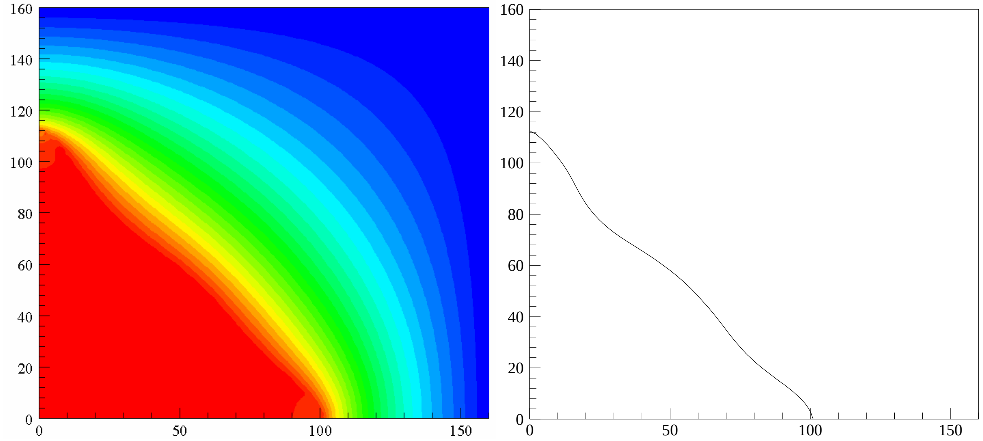

Fig. 14. Temperature contours and interface position for growth under low undercooling, $\Delta T = 0.3$.

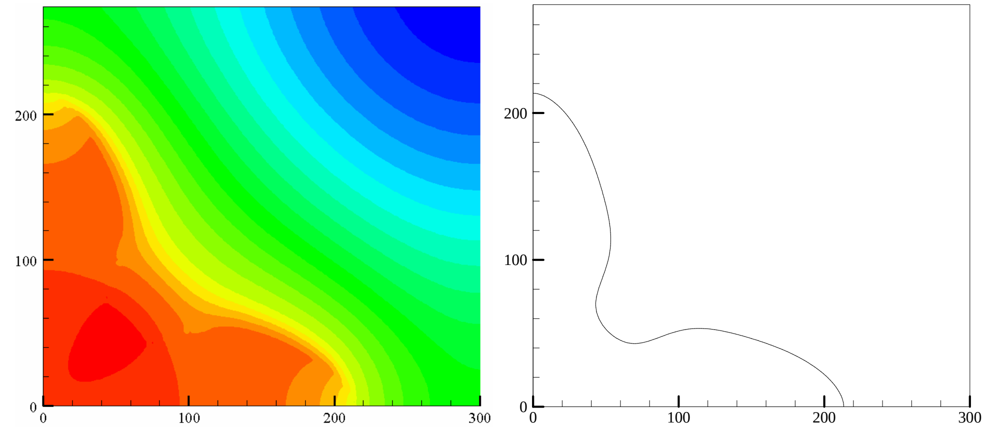

Fig. 15. Temperature contours and interface position for growth under low undercooling, $\Delta T = 0.1$.

5.4 Three-dimensional dendritic growth

Karma and Rappel [40] simulated the growth of dendrites in an undercooled melt using a phase field approach. Plapp and Karma [43] used a mixed finite differences (FD) - MC method for simulating three-dimensional (3D) dendritic solidification at low-undercooling. Schmidt [44] and Bansch and Schmidt [28] used a deforming mesh approach to simulate 3D solidification. Jeong et al. [45]

used an adaptive phase field method to simulate the growth of a 3D seed in the presence and absence of fluid flow. In this example, the growth of a spherical seed (of an initial radius $R = 30$) placed in an under-cooled melt is simulated. This simulation was previously performed by Jeong et al. [45]. Initially, the temperature in the melt is maintained at an undercooling of $\Delta T = -0.45$. All the sides of the domain are insulated. Anisotropy is introduced through the direction-dependent coefficient in the Gibbs-Thompson equation:

$$T = -d_o\left((1-3\epsilon)(1+\frac{4\epsilon}{1-3\epsilon}({n_1}^4+{n_2}^4+{n_3}^4))\right)\kappa, \tag{54}$$

where $d_o = 0.5$, $\epsilon = 0.05$ and n_i are the components of the normal unit vector along the i^{th} coordinate axes. The physical domain considered was a cube of dimension $[-400, 400]^3$ but only an octant of the cube was used for the computation. This domain was discretized by a $70 \times 70 \times 70$ grid. The steady-state temperature distribution and the crystal shape are given in Fig. 16. The anisotropy leads to preferential growth of arms in six normal directions.

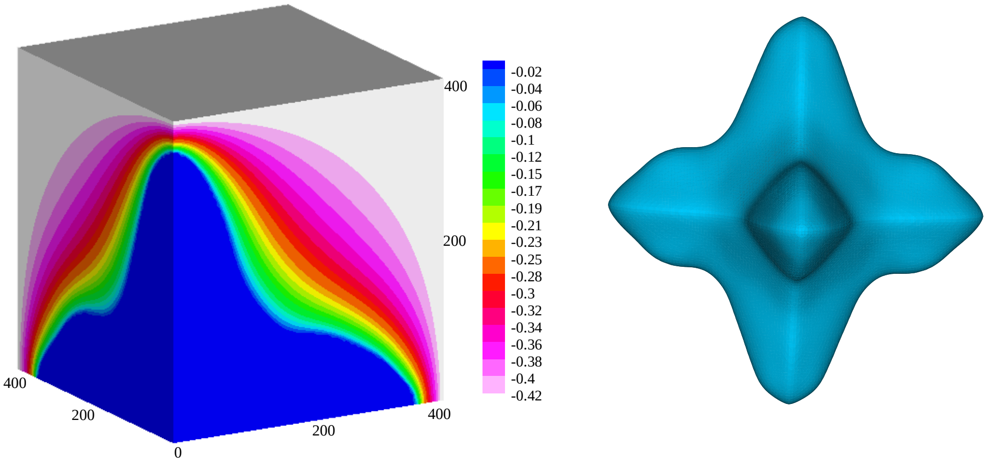

Fig. 16. On the left, temperature contours. On the right, steady-state crystal shape ($\tau = 80$).

The tip velocity reaches a steady-state value around 0.029 as shown in Fig. 17. This is in good comparison with the value of the tip velocity of 0.031 reported in [45]. It should be noted that the steady-state tip velocity in the 3D case is about 1.65 times the steady-state tip velocity obtained in the corresponding 2D case. The problem was re-run with a finer grid ($74 \times 74 \times 74$). This resulted in a slight increase in the steady state tip velocity to about 0.0305. This compares very well with the results for the tip velocity of 0.030 published in [45]. The computation using the finer mesh took about 12 hours to reach steady-state on a Pentium IV processor.

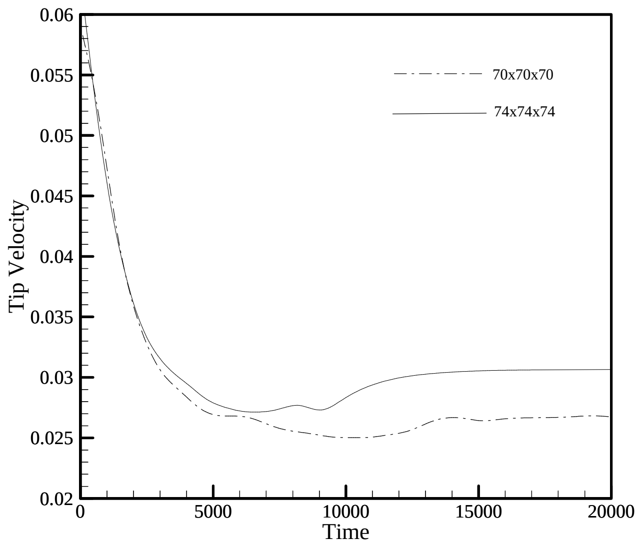

Fig. 17. Evolution of the tip velocity for a 3D diffusion-based growth using two different grids.

5.5 Dendritic growth in the presence of convection

We consider the modelling of equiaxed dendritic growth of a pure substance into a uniformly supercooled melt that is flowing around the crystal. This case was extensively investigated by Beckermann et al. [27] using a phase field approach while Al-Rawahi and Tryggvasson [31] studied the same problem following an explicit front-tracking approach. The same problem is investigated here to compare our methodology with those given above and furthermore to emphasize the computational advantages of the extended finite element method. The physical domain extends from -400 to 400 in both x and y directions. Since the problem is symmetric about the y axis, a computational domain $[0, 400] \times [-400, 400]$ is considered. This computational domain is discretized using a 200×400 grid. The mesh elements used are uniform quadrilateral elements. The initial temperature in the melt is -0.55, and the fluid flowing through the inlet has a fixed inlet temperature of -0.55. The other sides of the domain are insulated. The inlet velocity is set at $V_{in} = 1.0$. No slip boundary conditions are imposed on the vertical walls and a pressure boundary condition is imposed at the outlet. The material has a Prandtl number of 23.1.

Figure 18 shows the temperature and streamline contours at three different times. As reported in [27], there is significant distortion of the temperature with higher temperature gradients upstream. The upstream tip grows the fastest while the downstream tip has the smallest growth velocity. The time variation of the tip velocities is given in Fig. 19. The downstream tip velocity has not reached a steady-state even at the end of the simulation. The upstream tip velocity reaches a steady value while the tip normal to the flow direction shows a slightly oscillating behavior. The scaled steady-state upstream tip velocity is 0.0246 as compared to the value of 0.0244 obtained in [27,31]. The upstream tip radius for this case turns out to be 6.1 which is in agreement with the results in [31]. The upstream tip radius appears to reduce due to the introduction of flow. The 'tilting' of the dendritic arms normal to the flow in the direction opposite the flow direction is observed. This angle of growth is around 0.74^o to the x axis.

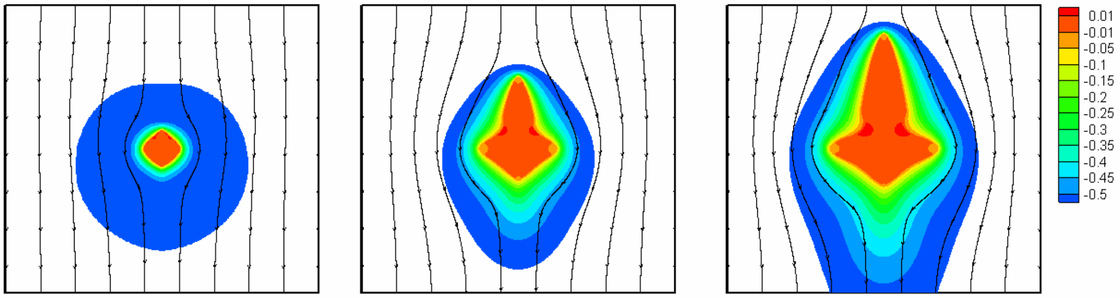

Fig. 18. Two-dimensional solidification with convection: Temperature and streamline contours at scaled times 10, 60 and 100.

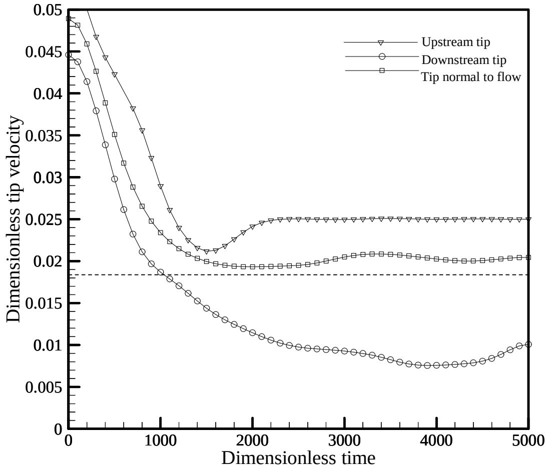

Fig. 19. Two-dimensional solidification with convection: Tip velocities as a function of time.

The present simulation took around 30 minutes to complete on a 1 GHz Pentium IV desktop. This is about 10 times faster than the solution in [27] obtained with a similar discretization (200 × 400 versus 288 × 576) (though the authors in [27] state that their code was not fully optimized).

The inlet velocity V_{in} and the undercooling ΔT both play a role in the evolution of the crystal shape. We simulated growth of the crystal for two sets of growth conditions. In the first set of simulations, the inlet velocity was varied from 0.01 to 0.1 while keeping the undercooling constant. This was done to understand the effect of flow at constant undercooling. The shapes of the crystals for two sets ($\Delta T = -0.55, -0.35$) of undercoolings are given in Fig. 20. At the lower undercooling, increasing the inlet velocity caused considerable variation in the heat transfer between the upstream and downstream regions. The growth of the lower tip is stunted even at moderate inlet velocities and is highly suppressed at higher velocities. Similarly the upstream tip velocities are very sensitive to the inlet velocity. Interestingly, a higher undercooling results in smoothing the variation between the upstream tip and the downstream tip. Notice that at higher undercoolings, the sensitivity of the crystal shape to the inlet speed decreases, though the average tip velocity increases with the undercooling. The time for the tip to reach the end of the computational domain is a measure of the average rate of growth. The growth rate increases (nearly) linearly with the undercooling (5000 dimensionless steps for -0.55, 8500 dimensionless steps for -0.45, 13000 dimensionless steps for -0.35).

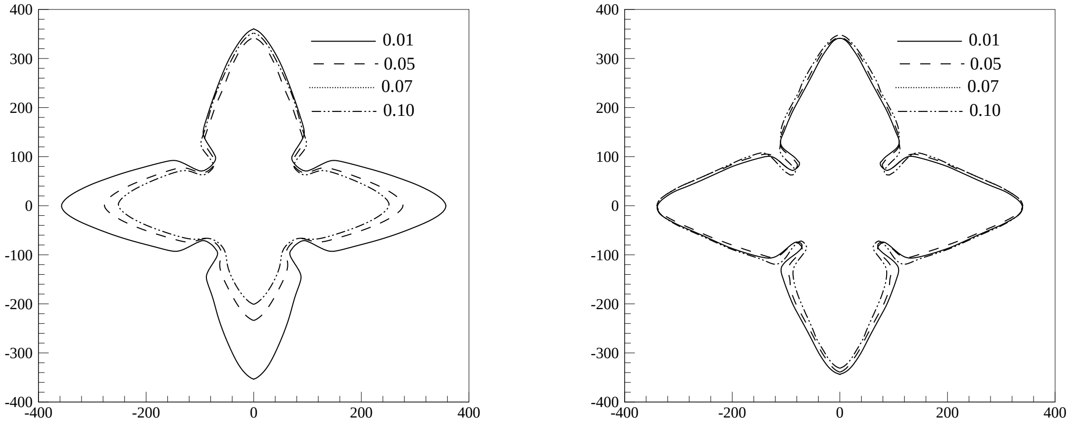

Fig. 20. Two-dimensional solidification with convection: Crystal shapes at various inlet velocities with undercoolings of -0.35 (left) and -0.55 (right).

Figure 21 shows the final crystal shapes for growth with an inlet velocity 0.05 and 0.10 and different undercoolings. As the undercooling increases the rate of growth increases appreciably in the downstream direction, as well as in the direction normal to the flow. Further more, with an increase in the undercooling, the curvature of the crystal, near the cusp between two arms, increases. Notice that there is a reduction in the 'tilting' of the side branches at the higher undercooling. We feel that this is mainly due to the fact that the tip has reached the end of the computational domain and the confinement effects of the flow push the tip downward. It is found that as the inlet velocity increases, the upstream tip grows faster and the downstream tip grows slower.

However for the perpendicular tip, its evolution does not change significantly. For a given inlet flow velocity, the velocity of all the tips increases significantly as the undercooling increases. The following observations are made from this study:

- Increased fluid flow as well as increased undercooling both increase the growth rates
- The effect of increased fluid flow is more prominent at lower undercoolings
- Increasing undercooling while keeping V_{in} constant increases the net growth rate nearly linearly
- The side branches are relatively unaffected by changes in flow but are affected by changes in undercooling
- Increased undercooling for the same flow speed results in more uniform growth between the upstream and downstream tips

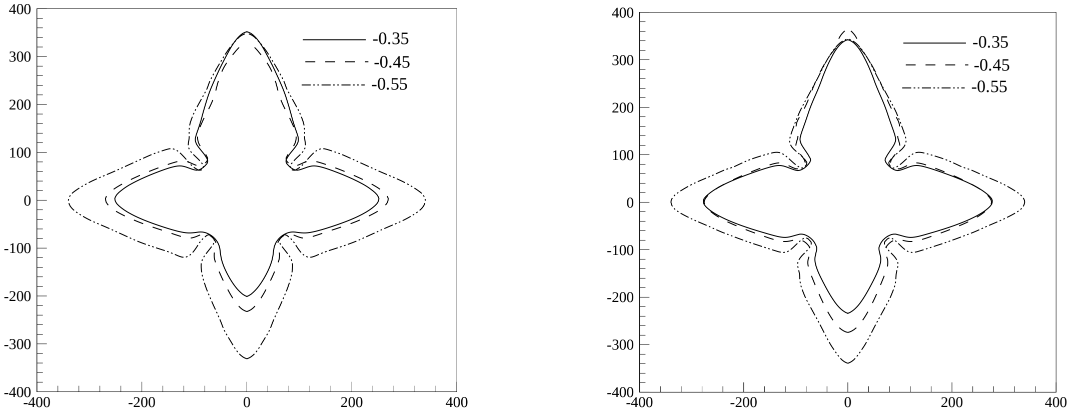

Fig. 21. Two-dimensional solidification with convection: Crystal shapes at various undercoolings with inlet velocity of 0.05 and 0.10.

Remark 3: The modelling of the fluid flow is performed using the volume-averaging technique after assuming that the sharp-interface is smeared out across two elements. This mushy zone is characterized by the liquid fraction that is defined in terms of the level set function. The momentum equation is then solved in the whole computational domain instead of just the liquid region. The assumption of the Kozeny-Carman permeability model is not based on the physics of the problem (which is that of sharp-solidification). It is rather a convenient tool for an accurate implementation of the fluid flow problem that satisfies the no-slip boundary condition near the interface. Figure 22 shows the velocity vectors near the interface during the growth of the crystal. This figure corresponds to an inlet velocity of 0.1 and an undercooling of -0.55 and dimensionless time 1000. The physical domain extends from -400 to 400 in both x and y directions. But since the problem is symmetric about the y axis, a computational domain $[0, 400] \times [-400, 400]$ is considered. This computational domain is discretized using a 200×400 uniform quadrilateral elements. The fluid flowing through the inlet has a fixed inlet temperature of -0.55. The material has a Prandtl number of 23.1. The rest of the simulation parameters are as defined in the examples above. Note that the fluid velocity near the interface drops down to negligible values. The velocity of the fluid was extracted at discrete points on the interface. The largest velocity magnitude is of the order of 3×10^{-5}. To emphasize the fact that the momentum evolution solution technique is not particularly important (as long as the no-slip velocity condition is satisfied), we simulated the same problem using a different methodology for the momentum equation. Tonhardt and Amberg [26] used a variable viscosity model to enforce the no-slip boundary condition. In this method the viscosity of the material in the solid is much larger than the viscosity in the melt, with the viscosity varying according to a pre-determined function (dependent on the liquid fraction) across the interface. In our simulation, we assumed the viscosity of the solid to be 1000 times the viscosity of the melt, varying smoothly (linearly) across two elements near the interface. Figure 22 (right) shows the velocity vectors near the interface during the growth of the crystal. In this case too, the maximum velocity magnitude on the interface is negligibly small, in the order of 7×10^{-6}.

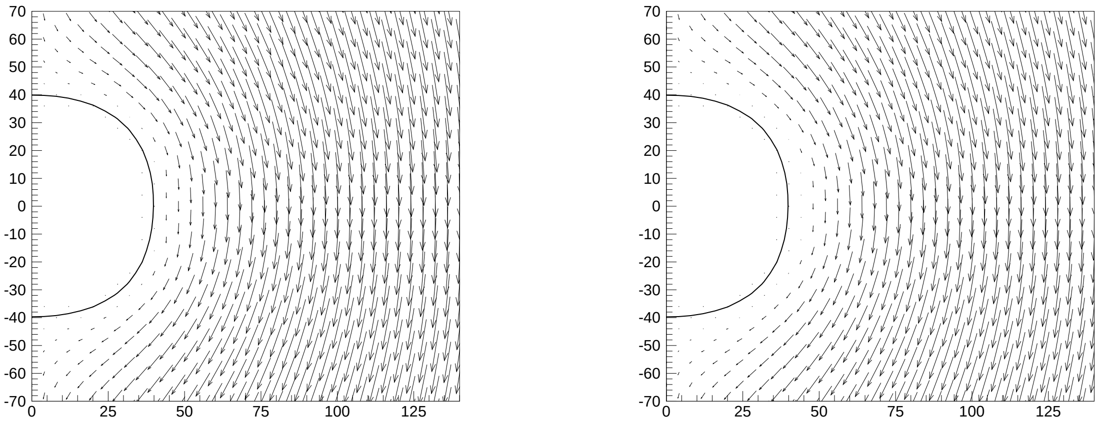

Fig. 22. Velocity vectors near the interface. On the left: volume averaging model, On the right: variable viscosity model.

5.6 Three-dimensional dendritic growth in the presence of convection

In this section, we address the ability of the extended finite element formulation to deal with three-dimensional dendritic growth in the presence of convection. Two examples are considered in this section.

5.6.1 *Growth under low-undercooling*

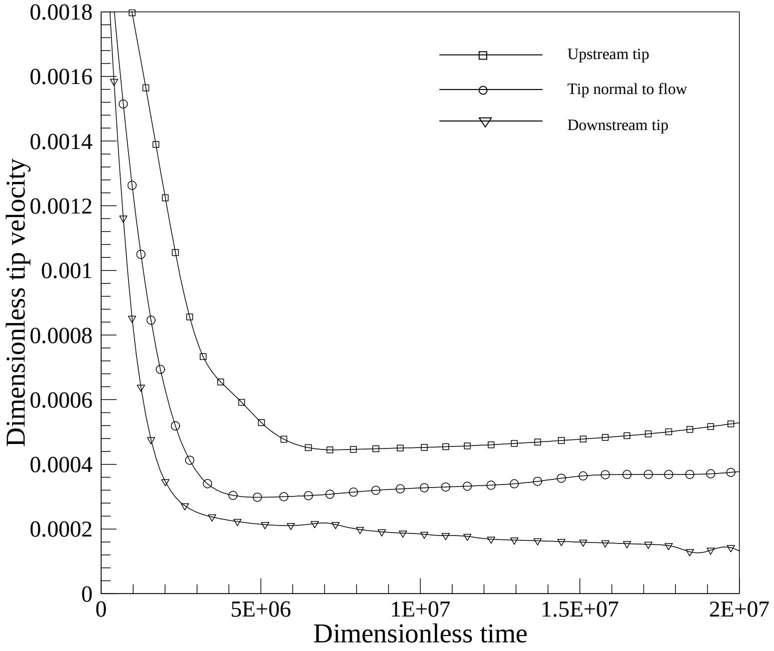

Fig. 23. Evolution of the tip velocities.

Al-Rawahi and Tryggvasson [32] investigated the three-dimensional growth of a crystal in the presence of convection. The seed grew in a melt which had a low-undercooling. The dimensions of the physical domain considered are $[-5100, 5100]^3$. A spherical seed of initial size $2r = 300$ is placed in the center of the domain. Fluid enters the domain from the top. The fluid is initially at an undercooling of $\Delta T = -0.15$. The melting temperature is 0.0. The fluid entering the domain has a temperature $T = -0.15$. All the other boundaries are insulated. Equal material properties for the two phases are considered. The fluid has unit Prandtl number. The inlet velocity is $V_{in} = 0.002$. The material has unit heat capacity (C_p), latent heat(L) and conductivity (k). These values lead to $St = \frac{C_p \Delta T}{L} = 0.15$, $Pr = 1.0$, $Pe = \frac{k V_i n d_o}{C_p} = 0.002$ (see [32] for more details of the problem). No slip boundary conditions are imposed on the vertical boundaries. The Gibbs-Thompson relation has an anisotropic surface tension effect given by

$$\epsilon = d_o(1 - A_s(4(n_1^3 + n_2^3 + n_3^3) - 3)) \tag{55}$$

where where $d_o = 1.0$ and $A_s = 0.3$. Because of symmetry, the computational domain is $[0, 5100] \times [0, 5100] \times [-5100, 5100]$. The domain is descretized into uniform regular hexahedrals. The number of elements are $46 \times 46 \times 92$ respectively in each coordinate direction. Al-Rawahi and Trygvasson used 256^3 elements for the same problem. The time evolution of the tip velocities is plotted in Fig. 23. The upstream and downstream tip velocities and the velocity of the perpendicular tip at the end of the simulation are $0.00058, 0.00022, 0.00044$, respectively. The velocity results shown here compare well with the velocities of upstream tip $= 0.00052$, downstream tip $= 0.00024$ and perpendicular tip $= 0.00039$ given in [32]. The shape of the crystal is shown in Fig. 24.

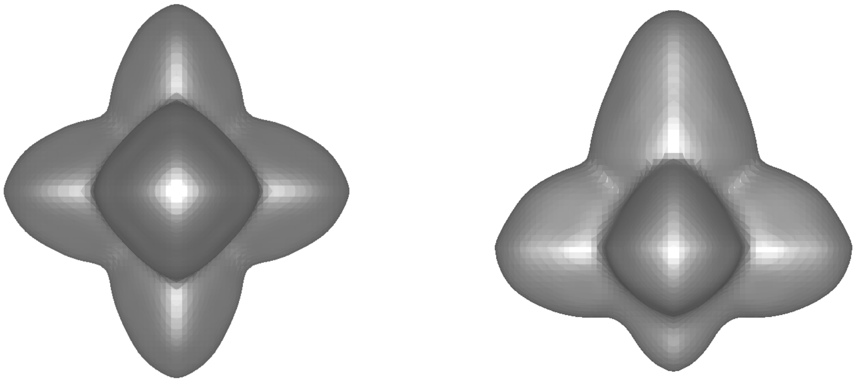

Fig. 24. Three-dimensional crystal growth with convection under low-undercooling conditions: On the left, top view and on the right, side view of the crystal at $\tau = 1.5E07$.

5.6.2 Growth under high-undercooling

A spherical seed of radius 20 is placed at the origin in an initially quiescent undercooled melt. The degree of undercooling of the melt is $\Delta T = -0.45$. The thermal diffusivity is $\alpha = 4$ and the Prandtl number is $Pr = 23.1$. At $t = 0$, fluid flows into the system from the top boundary. On the vertical walls, no slip boundary conditions are imposed. The physical domain is $[-200, 200] \times [-200, 200] \times [-200, 200]$. The computational domain is taken to be $[0, 200] \times [0, 200] \times [-200, 200]$. The domain is discretized using a $100 \times 100 \times 200$ grid. The temperature of the fluid flowing in at the inlet is maintained at ΔT. Insulated boundary conditions are maintained on the vertical sides. Other conditions include anisotropy given by:

$$T = -d_o(1 - \epsilon(4({n_1}^4 + {n_2}^4 + {n_3}^4)) - 3)\kappa, \tag{56}$$

with $d_o = 0.5$ and $\epsilon = 0.05$.

This case has been investigated by Jeong et al. [45] using an adaptive mesh, phase field simulation. The present computation took about 20 hrs on 32 nodes of the VII node cluster at the Cornell Theory Center. Though the simulation was moderately compute intensive, the visualization of the data was the hardest part of the analysis. Each time step yielded about 300 MB of data that had to be analyzed on a local machine.

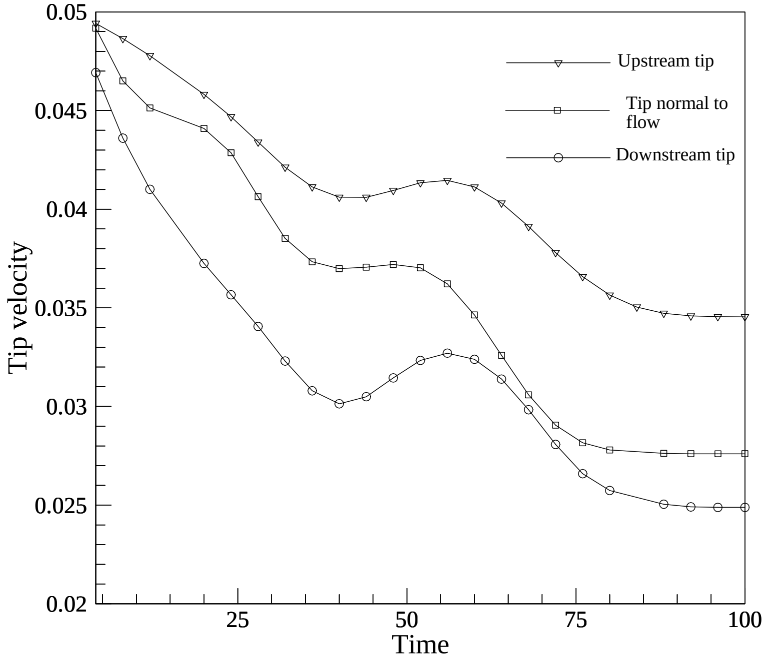

Fig. 25. Three-dimensional growth with melt convection: Evolution of the tip velocities.

The large value of the inlet velocity causes the thermal boundary layer to be suppressed upstream. Fig. 25 plots the evolution of the upstream, downstream and the tip velocity normal to the flow direction. It is seen that the upstream tip grows at an increased rate (versus the case without melt flow) as expected. The upstream tip velocity (non-dimensionalized) reaches a value of about 0.0345 which is 15% more than the case with out fluid flow. The downstream tip has a tip speed of about 0.0248 which is about 17% lower than the steady state diffusion tip speed. Notice that the downstream tip velocity decreases, then sharply increases and then drops again. This is explained by looking at the shape of the developing crystal in the presence of convection. The top view and elevation of the crystal are shown in Fig. 26. The sphere is constrained to grow in the six preferred directions (along the co-ordinate axes). The fluid flow leads to an increase in the tip velocity in the upstream direction and a consequent decrease in the tip velocity in the downstream direction. With further growth of the crystal, the flow passes between the growing arms of the crystal. The non-interacting colder fluid that flows between the arms of the dendrites leads to the formation of the small globules near the lower part of the crystal as seen in Fig. 26. This causes the flow to re-attach and leads to the increase in the downstream tip velocity. However, further growth of the globules cuts off the thermal gradient near the lower tip. This leads to a gradual reduction in the downstream tip velocity. Though none of the other investigations of this problem have revealed this phenomena, we assert that this is because the present simulation is resolving a much smaller length scale using the increased number of grid points (2×10^6) as well as resolving the interface better using the enriched finite element approximation.

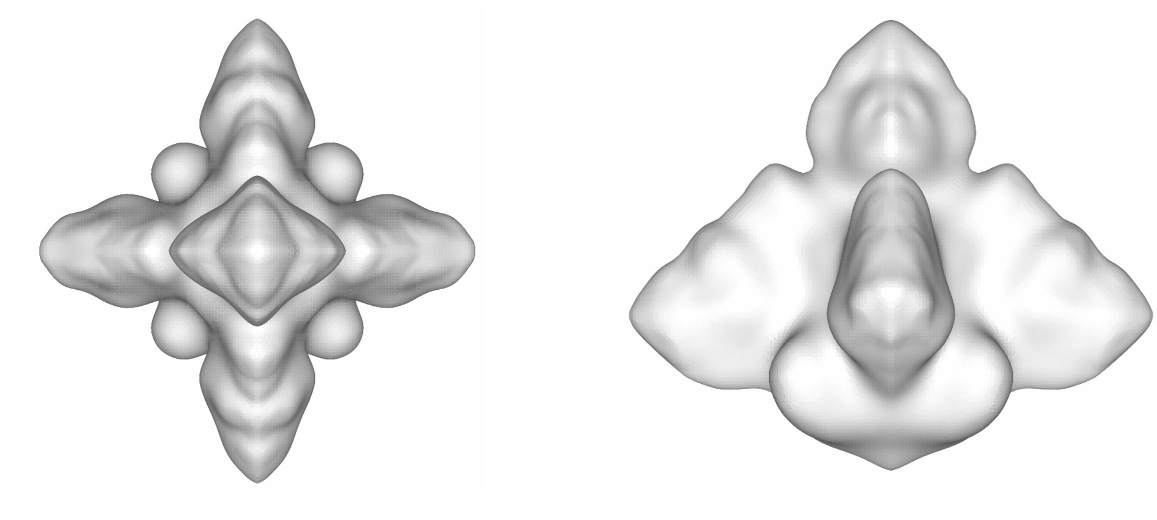

Fig. 26. Three-dimensional growth with melt convection: On the left, top view and on the right, side view of the crystal at $\tau = 80$.

6 Conclusions

We have presented a dimension-independent methodology to simulate the growth of dendrites in the presence of convection. The formulation involved using the extended finite element/level set method to simulate the temperature evolution and a volume-averaged stabilized finite element formulation for the velocity evolution. The solution of the velocity field involved the assumption that the sharp-interface is smeared out across two elements.

The simulations showed that the extended finite element is an attractive tool for capturing various phase transition phenomena in a computationally effective way. The moving interface can be accurately captured in 2D and 3D by using a single uniform grid. The object-oriented nature of the implementation drastically improved the ease of introducing melt convection. Further efforts to include the effects of concentration effects on the growth morphology are being undertaken.

Acknowledgements

The work presented here was funded in part by the NASA Microgravity Materials Science Program (grant NAG8-1671) and the Office of Industrial Technologies of the U.S. Department of Energy (DE-FC07-02ID14396). This research was conducted using the resources of the Cornell Theory Center, which receives funding from Cornell University, New York State, federal agencies, and corporate partners.

References

[1] W.W. Mullins, R.F. Sekerka, Stability of a planar interface during solidification of a dilute binary alloy, J. Appl. Phys. 35 (1964) 444–451.

[2] A. Wheeler, W. Boettinger, G. Mcfadden, Phase-field model for isothermal phase transitions in binary alloys, Physical Review A 45 (1992) 7424–7440.

[3] S. L. Wang, R.F. Sekerka, A.A. Wheeler, B.T. Murray, S.R. Coriell, R.J. Braun, G.B. McFadden, Thermodynamically consistant phase field models for solidification, Physica D 69 (1993) 189–200.

[4] A. Karma, W.J. Rappel, Phase field model for computationally efficient modelling of solidification with arbitrary interface kinetics, Phys. Rev. E 53 (1996) 3017–3020.

[5] R. Almgren, Second-order phase field asymptotics for unequal conductivities, SIAM J. Appl. Math. 59 (1999) 2086–2107.

[6] J.A. Warren, W.L. George, A parallel 3D dendritic growth simulator using the phase-field method, J. Comput. Phys. 177 (2002) 264–283.

[7] J.A. Sethian, Level set Methods (Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Science, Cambridge University Press, 1996.

[8] S. Osher, R. Fedkiw, Level set Methods and Dynamic Implicit Surfaces, Springer-Verlag, New York, 2003.

[9] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.

[10] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-phase flows, Computers and Fluids 27 (1998) 663–680.

[11] Y.C. Chang, T.Y. Hou, B. Merriman, S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flow, J. Comput. Phys. 124 (1996) 449–464.

[12] S. Chen, B. Merriman, S. Osher, P. Smereka, A simple level set method for solving stefan problems J. Comput. Phys. 135 (1997) 8–29.

[13] Y.T. Kim, N. Goldenfeld, J. Dantzig, Computation of dendritic microstructures using a level set method, Physical Review E 62 (2000) 2471–2474.

[14] F. Gibou, R. Fedkiw, R. Caflisch, S. Osher, A level set approach for the numerical simulation of dendritic growth, J. Scientific Computing 19 (2003) 183-199.

[15] L. Tan, N. Zabaras, A level set simulation of dendritic solidification with combined features of front-tracking and fixed domain methods, J. Comput. Phys. 211 (2006) 36-63.

[16] T. Belytschko, N. Moës, S. Usui, C. Parimi, Arbitrary discontinuity in finite elements, Int. J. Num. Meth. Engg. 50 (2001) 993–1013.

[17] J. Dolbow, N. Moës, T. Belytschko, An extended finite element method for modelling crack growth with frictional contact, Comp. Meth. Appl. Mech. Engg. 190 (2001) 6825–6846.

[18] N. Moës, A. Gravouil, T. Belytschko, Non-planar 3D crack growth by the extended finite element and level set–Part I: mechanical model, Int. J. Num. Meth. Engg. 53 (2002) 2549–2568.

[19] N. Moës, A. Gravouil, T. Belytschko, Non-planar 3D crack growth by the extended finite element and level set–Part II: level set update, Int. J. Num. Meth. Engg. 53 (2002) 2569–2586.

[20] G.J. Wagner, N. Moës, W.K. Liu, T. Belytschko, The extended finite element method for rigid particles in Stokes flow, Int. J. Num. Meth. Engg., 51(2001) 293–313.

[21] J. Chessa, T. Belytschko, An extended finite element method for two-phase flow, J. Appl. Mech. 70 (2003) 10–17.

[22] R. Merle, J. Dolbow, Solving thermal and phase change problems with the extended finite element method, Computational Mechanics 28 (2002) 339–350.

[23] H. Ji, D. Chopp, J.E. Dolbow, A hybrid extended finite element/level set method for modelling phase transformations, Int. J. Num. Meth. Engg. 54 (2002) 1209–1233.

[24] J. Chessa, P. Smolinski, T. Belytschko, The extended Finite Element Method for solidification problems, Int. J. Num. Meth. Engg. 53 (2002) 1959–1977.

[25] M.E. Glicksman, S.C. Huang, Convective heat transfer during dendritic growth. In *Convective transport and instability phenomena*, J. Zierep, H. Ortel (ed.), 1982.

[26] R. Tonhardt, G. Amberg, Phase field simulation of dendritic growth in a shear flow, J. Crystal Growth 194 (1998) 406–425.

[27] C. Beckermann, H.-J. Diepers, I. Steinbach, A. Karma, X. Tong, Modelling melt convection in phase-field simulations of solidification, J. Comput. Phys. 154 (1999) 468-496.

[28] E. Bansch, A. Schmidt, Simulation of dendritic crystal growth with thermal convection, Interf. Free Boundaries 2 (2000) 95-115.

[29] H.S. Udaykumar, R. Mittal, W. Shyy, Computation of solid liquid phase fronts in the sharp-interface limit on fixed grids, J. Comput. Physics 153 (1999) 535–574.

[30] D. Juric, Direct numerical simulation of solidification microstructures affected by fluid flow, Proceedings of the Eighth International Conference on Modelling of Casting, Welding, and Advanced Solidification Processes, San Diego, CA (MCWASP VIII, July 1998).

[31] N. Al-Rawahi, G. Tryggvason, Numerical simulation of dendritic solidification with convection: two-dimension geometry, J. Comput. Phys. 180 (2002) 471-496.

[32] N. Al-Rawahi, G. Tryggvason, Numerical simulation of dendritic solidification with convection: three-dimension flow, J. Comput. Phys. 194 (2002) 677-696.

[33] N. Zabaras, D. Samanta, A stabilized volume-averaging finite element method for flow in porous media and binary alloy solidification processes, Int. J. Num. Method Engg., 60 (2004) 1103-1138.

[34] J. A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annual Review of Fluid Mechanics 35 (2003) 341-372.

[35] N. Sukumar, D.L. Chopp, N. Moës, T. Belytschko, Modelling holes and inclusions by level sets in the extended finite element method, Comp. Meth. Appl. Mech. Engg. 190 (2001) 6183–6200.

[36] H. Ji, J.E. Dolbow, On strategies for enforcing interfacial constraints and evaluating jump conditions with the extended finite element method, Int. J. Num. Meth. Engg., 61(14) (2004) 2508 - 2535.

[37] N. Moës, M. Cloirec, P. Cartraud, J.-F. Remacle, A computational approach to handle complex microstructure geometries, Comp. Meth. Appl. Mech. Engrg. 192 (2003) 3163-3177.

[38] L.M. Jiji, K.A. Rathjen, T. Drzewiecki, Two-dimensional solidification in a corner, Int. J. Heat and Mass Transfer 13 (1970) 215–218.

[39] G. Caginalp, Dynamical renormalization group calculation of a two-phase sharp-interface model, Phys. Rev. E 60 (1999) 6267–6270.

[40] A. Karma, W.J. Rappel, Quantitative phase-field modelling of dendritic growth in two and three-dimensions, Phys. Rev. Letters 80 (1998) 4323–4349.

[41] N. Provatas, N. Goldenfeld, J. Dantzig, Adaptive mesh refinement computation of solidification microstructures using dynamic data structures, J. Comput. Phys. 148 (1999) 265–290.

[42] N. Provatas, N. Goldenfeld, J. A. Dantzig, Adaptive grid methods in solifification microstructure modelling, Phys. Rev. Letters 80 (1998) 3308–3311.

[43] M. Plapp, A. Karma, Multiscale finite-difference-diffusion-monte-carlo method for simulating dendritic solidification, J. Comput. Phys. 165 (2000) 592–619.

[44] A. Schmidt, Computation of three-dimensional dendrites with finite elements, J. Comput. Phys. 25 (1996) 293–312.

[45] J. Jeong, N. Goldenfeld, J. Dantzig, Phase field model for three-dimensional dendritic growth with fluid flow, Physical Review E 64 (2001) 041602.