Journal of Computational Physics 228 (2009) 5803–5818

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A numerical scheme for the Stefan problem on adaptive Cartesian grids with supralinear convergence rate

Han Chen [a], Chohong Min [b], Frédéric Gibou [c,*]

[a] Department of Mechanics, Huazhong University of Science and Technology, Wuhan, China
[b] Department of Mathematics, KyungHee University, 130-701 Seoul, Republic of Korea
[c] Mechanical Engineering Department and Computer Science Department, University of California, Santa Barbara, CA 93106, United States

ARTICLE INFO

Article history:
Received 26 March 2007
Received in revised form 20 January 2009
Accepted 30 April 2009
Available online 12 May 2009

Keywords:
Level set
Stefan problem
Non-graded adaptive grid
Quadtree

ABSTRACT

We present a level set approach to the numerical simulation of the Stefan problem on non-graded adaptive Cartesian grids, i.e. grids for which the size ratio between adjacent cells is not constrained. We use the quadtree data structure to discretize the computational domain and a simple recursive algorithm to automatically generate the adaptive grids. We use the level set method on quadtree of Min and Gibou [C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, J. Comput. Phys. 225 (2007) 300–321] to keep track of the moving front between the two phases, and the finite difference scheme of Chen et al. [H. Chen, C. Min, F. Gibou, A supra-convergent finite difference scheme for the poisson and heat equations on irregular domains and non-graded adaptive Cartesian grids, J. Sci. Comput. 31 (2007) 19–60] to solve the heat equations in each of the phases, with Dirichlet boundary conditions imposed on the interface. This scheme produces solutions that converge supralinearly (~ 1.5) in both the L^1 and the L^∞ norms, which we demonstrate numerically for both the temperature field and the interface location. Numerical results also indicate that our method can simulate physical effects such as surface tension and crystalline anisotropy. We also present numerical data to quantify the saving in computational resources.

1. Introduction

The Stefan problem is a moving interface model where the main physical process is diffusion, and is at the center of the study of crystal growth. Moreover, it has important applications in the growing field of semi-conductors, in combustion, in bio-nanotechnology, in tissue engineering and many others. The difficulty in solving the Stefan problem stems from the fact that the interface position must be computed as part of the solution. In addition, boundary conditions, some of which are depending on the interface shape, must be satisfied at the evolving front.

Successful numerical methods for this type of problems need to be able to efficiently solve the heat equation on irregular domains and keep track of a moving interface that may undergo complex topological changes such as the merging and the pinching of two fronts. The interface that separates the two phases can be either explicitly tracked or implicitly captured. The main advantage of explicit approaches, e.g. front tracking [18], is their accuracy. The main disadvantage is that special care is needed for topological changes. In turn, the explicit treatment of connectivity makes the method challenging to extend to three spatial dimensions. Implicit representations such as the level set method [29,30,35] and the phase-field method

[*] Corresponding author.
E-mail address: fgibou@engineering.ucsb.edu (F. Gibou).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2009.04.044

[20] represent the front as an iso-contour of a continuous function. Topological changes are consequently handled in a natural fashion, and thus these methods are readily implemented in both two and three spatial dimensions.

In this paper we use the level set method, first introduced by Osher and Sethian [30,29,35], which is a sharp interface model that can be used to exactly locate the interface in order to apply discretizations that depend on the exact interface location. This method can handle discontinuous material properties easily. The earliest level set method for solidification-type problems was presented by Sethian and Strain [34], where the authors recast the equations of motion into a boundary integral formulation and used the level set method to update the location of the interface. Chen et al. [11] introduced a level set approach where the diffusion equation was solved directly with finite difference schemes. A similar approach was proposed by Udaykumar et al. [39]. In both cases, the solution is second-order accurate and the resulting linear system is nonsymmetric. Kim et al. [19] demonstrated that this method produces results in agreement with the solvability theory [23]. Gibou et al. proposed a second-order accurate symmetric discretization [15,14] and third-order nonsymmetric discretization [13]. The methods presented in all these works are on uniform grids.

Physical phenomena modelled by the Stefan problem present differences in length scales and numerical approximations on uniform grids are in such cases extremely inefficient in terms of C.P.U. and memory requirement since only a small fraction of the domain needs high grid resolution, while other parts of the domain can produce accurate solutions on much coarser grids. In fact, since the Stefan problem is described by a parabolic partial differential equation, the solution is smooth far from the interface where coarse grids are sufficient to adequately capture the variations of the solution. On the other hand, the solution presents a kink near the interface, due to Dirichlet boundary conditions. It is therefore desirable to design a scheme that refines automatically near the interface in order to capture the sharp features in the solution's gradients, while leaving coarser cells far away.

Adaptive mesh refinement techniques are advantageous in those cases: AMR (adaptive mesh refinement) on Cartesian grids has been pioneered by the work of Berger and Oliger [9] and Berger and Colella [8] for compressible flows. Within this framework, the domain is first discretized as a coarse grid and then rectangular blocks of uniform grids with finer resolution are added where the solution requires higher accuracy. Several applications for elliptic, parabolic and hyperbolic systems have been proposed [4–6,17,22]. This line of work was also applied to two-phase flows in Sussman et al. [37].

Implementations based on recursive data structures have become more popular in recent years. These approaches do not impose patches of uniform grids but rather allows the grid cells to continuously change in size. Quadtree (in 2D) and octree (in 3D) data structures [32,33] are convenient in representing such grids, and have been proven to be optimal [1]. Recently, Min et al. [27] solved the variable coefficient Poisson equation on a rectangular domain with non-graded adaptive Cartesian grids, i.e. grids for which the size ratio between adjacent cells is not constrained, and obtained second-order accurate solutions with second-order accurate gradients in the L^1 and L^∞ norms. This work was then extended in Chen et al. [10] to irregular domains with non-graded Cartesian grids to achieve second-order accuracy for both the solution and its gradients. The advantage of non-graded grids is that mesh generation is efficient and straightforward. It also save computational resources.

In this paper, we build on the work of [10,26] to present a level set approach to the numerical simulation of the Stefan problem on adaptive Cartesian grids. We note that the extremely simple refinement scheme we are using produces grids that are mostly graded, but not necessarily. We emphasize that the numerical method we present is applicable to non-graded Cartesian grids and that other refinement criteria, potentially more efficient or more adapted to particular problems, can be used. We demonstrate the supra-linear convergence of our method in the L^1 and the L^∞ norms and demonstrate that it can simulate physical effects such as thermal conductivity, crystalline anisotropy, surface tension, molecular kinetics and undercoolings. Finally, we illustrate the efficiency and accuracy by comparing with computations on uniform grids.

2. Equations

2.1. Equations for the Stefan problem

Consider a domain $\Omega = \Omega^- \cup \Omega^+$, where the two subdomains Ω^- and Ω^+ are separated by an interface Γ. The boundary of Ω is denoted by $\partial\Omega$ (see Fig. 1). The Stefan problem describes the evolution of a scalar T, equals to T_s in Ω^- and T_l in Ω^+, such that:

$$\frac{\partial T_s}{\partial t} = \nabla \cdot (D_s \nabla T_s) \quad \text{in } \Omega^-, \tag{1}$$

$$\frac{\partial T_l}{\partial t} = \nabla \cdot (D_l \nabla T_l) \quad \text{in } \Omega^+, \tag{2}$$

where the subscripts s and l denote the solid and liquid phases, respectively. The diffusion constants D_s and D_l can be discontinuous across Γ. The temperature on the solid–liquid interface satisfies the following conditions

$$T_s = T_l = T_\Gamma \quad \text{on } \Gamma, \tag{3}$$

where T_Γ denotes the local interface temperature, which is given in terms of the Gibbs–Tompson boundary condition (see e.g. [2,3]):

$$T_\Gamma = -\epsilon_c \kappa - \epsilon_v V \cdot n, \tag{4}$$

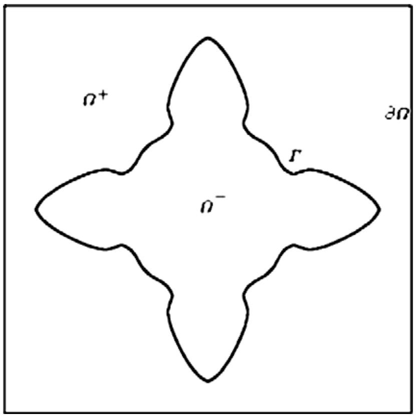


Fig. 1. Schematic of the computational domain.

where V denotes the interface velocity, n denotes the normal vector to the interface, κ denotes the interface curvature and ϵ_c and ϵ_v two parameters controlling the amount of surface tension forces and molecular kinetics. Finally, the interface normal velocity is given by the jump in flux to the interface:

$$V \cdot n = -(D_l \nabla T_l - D_s \nabla T_s) \quad \text{on } \Gamma. \tag{5}$$

2.2. Level set equations

We use the level set method, introduced by Osher and Sethian [30], to implicitly represent the interface between the two phases. In particular, we describe Ω^- by the set of points (x, y) such that $\phi(x, y) < 0$. Likewise, we describe Ω^+ by the set of points (x, y) such that $\phi(x, y) > 0$. The interface Γ is defined by $\phi(x, y) = 0$. The evolution of the interface is then given by the evolution of ϕ:

$$\phi_t + V_n |\nabla \phi| = 0, \tag{6}$$

where $V_n = V \cdot n$ is the normal component of the velocity. In the case of the Stefan problem, this equation simplifies to

$$\phi_t - [\nabla T] \cdot \nabla \phi = 0, \tag{7}$$

where the jump of the temperature gradient $[\nabla T]$ is taken from the liquid phase to the solid phase. The normal to the interface and the interface curvature are defined by

$$n = \frac{\nabla \phi}{|\nabla \phi|}, \quad \kappa = \nabla \cdot n, \tag{8}$$

respectively and are numerically approximated by central differencing. In order to keep the level set function close to a signed distance function, we use the reinitialization scheme of Sussman et al. [38]:

$$\phi_\tau + \text{Sign}(\phi_0)(|\nabla \phi| - 1) = 0 \tag{9}$$

for a few iterations τ. ϕ_0 is a level set function that is not necessarily a signed distance function but describes the same contour as ϕ. The interested reader is referred to [29,35] for general details in the level set method.

3. Automatic grid generation

We represent the grid in two spatial dimensions using the quadtree data structure [32,33]: Referring to Fig. 2, the root of the tree corresponds to the entire domain and we add four children cells as we split the domain in four equal parts. This process is repeated recursively in order to refine parts of the domain where interesting features develop.

By definition, a quadtree is graded if the difference between two adjacent cell levels is at most one, i.e. a cell can be at most twice as big as its neighbor. In our approach, we do not constrain the ratio between adjacent cells, i.e. even though for most of our computations our grids turn out to be graded, the method we present can handle non-graded meshes. This has the benefit of providing a straightforward algorithm for mesh generation as described next. Due to the parabolic nature of the Stefan problem, the solution will be smooth far away from the interface so that coarse grids are sufficient to capture the variations of the solution. Near the interface, the solution presents a kink due to the Dirichlet boundary condition. Therefore, the grid should be refined near the interface and coarsen elsewhere. A simple algorithm for generating the grid is that proposed in [26,28]: Simply split a cell C if the following expression is satisfied:

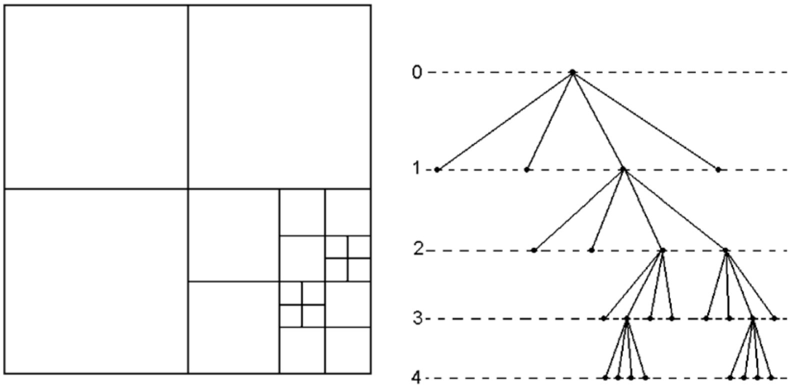


Fig. 2. Discretization of a two dimensional domain (left) and its quadtree representation (right). In this example, the tree is non-graded since the difference of level between neighboring cells exceeds one.

$$\min_{v \in \text{Vertices}} |\phi(v)| < \text{Lip} \times \text{Diag}(C), \tag{10}$$

where v is any vertex of cell C, Vertices is the set of all vertices of C, Lip is the Lipschitz constant taken to be close to one (since the level set is close to a signed distance function – we take Lip = 1.1), and $\text{Diag}(C)$ is the diagonal length of cell C. This formula simply expresses the fact that the level set function is Lipschitz continuous. Therefore, if the formula is satisfied, we know that the interface within the current cell and that we need to refine accordingly. We note that this simple algorithm does not necessarily produce graded meshes.

In addition, we can control three parameters pertaining to the quality of the adaptive grid. The largest cell size in the grid is determined by the parameter MinRes, such that the cell size $\triangle x = L/\text{MinRes}$ for the coarsest cell (L is the length of the root cell Ω). The smallest cell size is controlled by the parameter MaxRes, such that the cell size $\triangle x = L/\text{MaxRes}$ for the finest cell. Finally, it is straightforward to impose a band of uniform cells along the interface with the finest resolution if needed. The parameter BandWidth controls the approximate number of cells in this band across the interface. The implementation of these three parameters are done by splitting a cell if it is larger than the size determined by MinRes, stopping splitting a cell if it is already no larger than the size determined by MaxRes and splitting a cell if it is within the uniform band around the interface until MaxRes is reached, in addition to the splitting criterion 10.

4. Numerical methods

4.1. Treating T-junction nodes

The main difficulty in designing numerical schemes on non-graded adaptive Cartesian grids comes from T-junctions nodes, i.e. nodes for which adjacent nodes are not necessarily aligned in a Cartesian direction. In Min et al. [27,25,26], we introduced a new approach to define the values at these ghost nodes that automatically produces second-order accurate schemes. In particular, referring to Fig. 3, we define a ghost node T_4 as:

$$T_4 = \frac{s_5 T_6 + s_6 T_5}{s_5 + s_6} - \frac{s_5 s_6}{s_2 + s_3}\left(\frac{T_2 - T_0}{s_2} + \frac{T_3 - T_0}{s_3}\right). \tag{11}$$

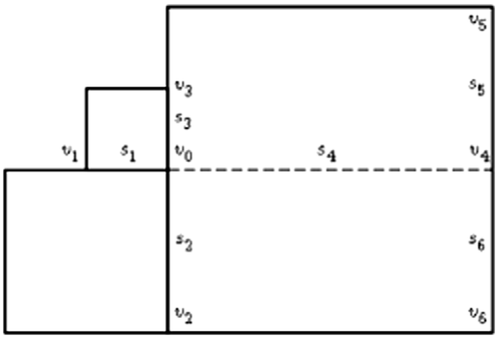


Fig. 3. A general configuration in 2D illustrating a T-junction, where v denotes a cell vertex and s denotes the length of a cell edge. In this grid, v_0 does not have a direct neighbor to the right. The broken line between v_0 and the ghost node v_4 is imaginary.

The last term in Eq. (11) compensates for the dominant error term in the linear interpolation which involves the second-order derivative in the y direction. Once every node has its regular neighboring nodes or well-defined ghost nodes or simply given by the Dirichlet boundary condition at the interface in all the spatial directions, the first and second-order derivatives can be discretized naturally, as if the grid was uniform. In particular, we define the first and second-order derivatives as:

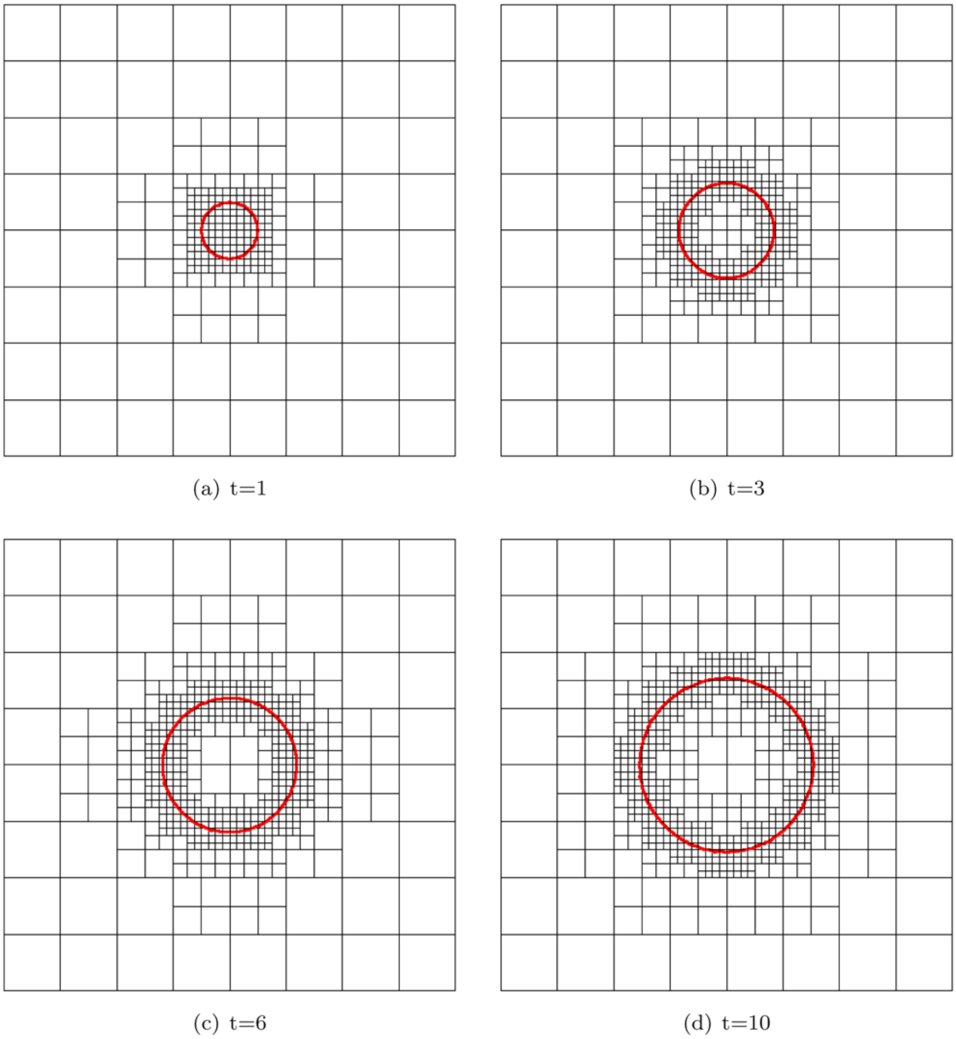
$$T_x(v_0) = \frac{T_4 - T_0}{s_4} \cdot \frac{s_1}{s_1 + s_4} + \frac{T_0 - T_1}{s_1} \cdot \frac{s_4}{s_1 + s_4}, \tag{12}$$

$$T_{xx}(v_0) = \frac{T_4 - T_0}{s_4} \cdot \frac{2}{s_1 + s_4} + \frac{T_1 - T_0}{s_1} \cdot \frac{2}{s_1 + s_4}. \tag{13}$$

Chen et al. [10] used such discretizations and a Crank–Nicolson scheme to solve the heat equation on irregular domains and showed that such discretizations produce second-order accurate solutions with second-order accurate gradients. We refer the reader to [10] for details.

4.2. Level set evolution

The level set equation is solved using the semi-Lagrangian method described in [26]. We update the level set function as: $\phi^{n+1}(\mathbf{x}^{n+1}) = \phi^n(\mathbf{x}_d)$, where $\mathbf{x}^{n+1}$ is any grid node and $\mathbf{x}_d$ is the corresponding departure point from which the characteristic curve originates. In our simulation, the second-order trapezoidal method is used to find the departure point backward in time,


(a) t=1

(b) t=3

(c) t=6

(d) t=10

Fig. 4. Frank-sphere solution. Evolution of the interface and corresponding adaptive grid. Note that the grid is not necessarily graded.

$$\hat{\mathbf{x}} = \mathbf{x}^{n+1} - \Delta t \cdot \mathbf{V}^{n+1}(\mathbf{x}^{n+1}),$$

$$\mathbf{x}_d = \mathbf{x}^{n+1} - \frac{\Delta t}{2} \cdot \mathbf{V}^{n}(\hat{\mathbf{x}}) - \frac{\Delta t}{2} \cdot \mathbf{V}^{n+1}(\mathbf{x}^{n+1}). \qquad (14)$$

Non-oscillatory interpolation formulas are then used to recover the value of the solution at such points.

4.3. Reinitialization equation

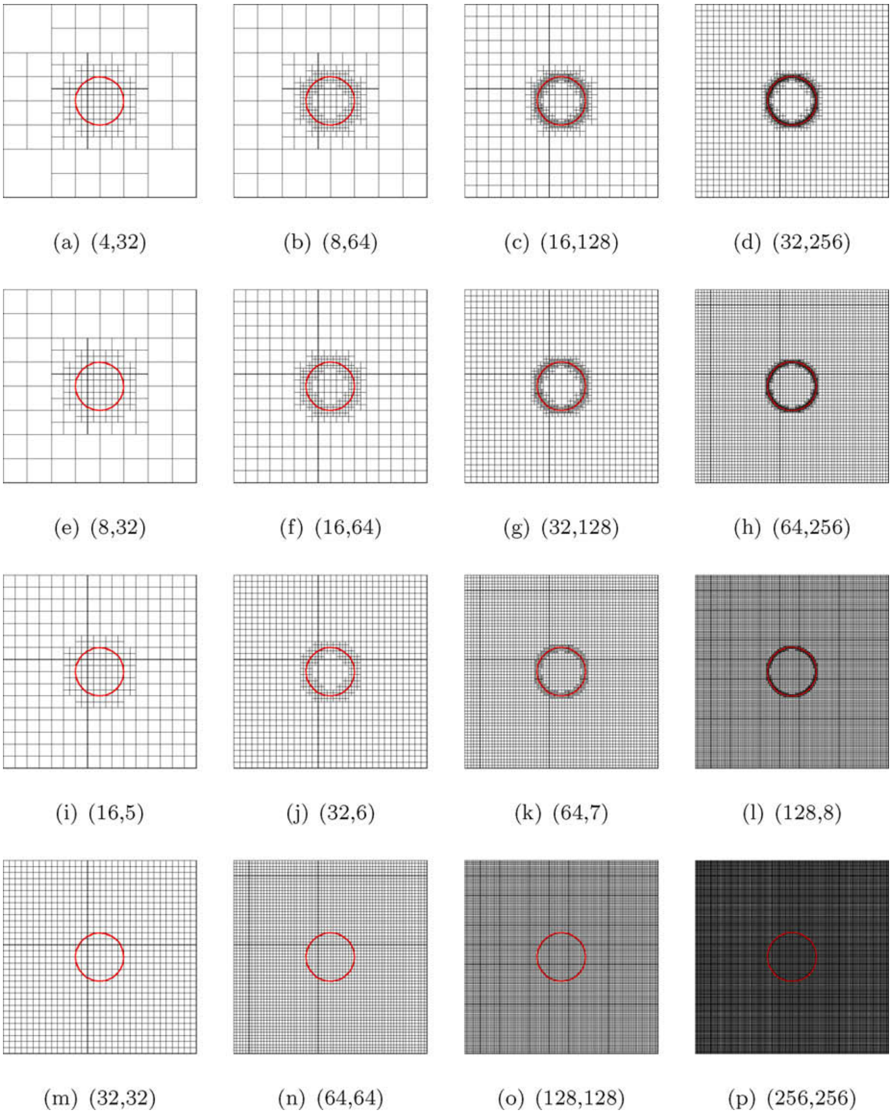
The reinitialization equation is solved using a standard Godunov scheme

$$\frac{\phi^{n+1} - \phi^{n}}{\Delta\tau} + \mathrm{sign}(\phi^0)[H_G(D_x^+\phi, D_x^-\phi, D_y^+\phi, D_y^-\phi) - 1] = 0, \qquad (15)$$

where the Godunov Hamiltonian H_G is given by

$$H_G(a,b,c,d) = \begin{cases} \sqrt{\max(|a^+|^2, |b^-|^2) + \max(|c^+|^2, |d^-|^2)} & \text{if } \mathrm{sign}(\phi^0) \geqslant 0 \\ \sqrt{\max(|a^-|^2, |b^+|^2) + \max(|c^-|^2, |d^+|^2)} & \text{if } \mathrm{sign}(\phi^0) < 0 \end{cases} \qquad (16)$$

and $a^+ = \max(a,0), a^- = \min(a,0)$. The derivatives are discretize using the following second order finite differences with the minmod slope limiters (see [21,36] for more details):


(a) (4,32) (b) (8,64) (c) (16,128) (d) (32,256)

(e) (8,32) (f) (16,64) (g) (32,128) (h) (64,256)

(i) (16,5) (j) (32,6) (k) (64,7) (l) (128,8)

(m) (32,32) (n) (64,64) (o) (128,128) (p) (256,256)

Fig. 5. Illustration of the grids in the convergence test for the Frank sphere problem. The values for (MinRes, MaxRes) are given for each case.

$$D_x^+\phi = \frac{\phi_2 - \phi_0}{s_2} - \frac{s_2}{2}\text{minmod}(\phi_{xx}(v_0), \phi_{xx}(v_2)), \tag{17}$$

$$D_x^-\phi = \frac{\phi_0 - \phi_1}{s_1} - \frac{s_1}{2}\text{minmod}(\phi_{xx}(v_0), \phi_{xx}(v_1)), \tag{18}$$

where $\phi_{xx}(v_0), \phi_{xx}(v_1)$ and $\phi_{xx}(v_2)$ are calculated as in Eq. (13). The discretization in the other directions can be obtained similarly.

4.4. Algorithm for solving the Stefan problem

Our method for solving the Stefan problem uses the level set approach on non-graded adaptive grids of [26] to effectively capture the interface at each time step, and the finite difference schemes in [10] to solve the heat equations in both the liquid and the solid phases. The solution of the heat equation and its gradients are second-order accurate. Two copies of the temperature, T_s^n and T_l^n, are defined on every grid node in the whole computational domain Ω. They represent the temperature at time step n in the solid region Ω^- and the liquid region Ω^+, respectively.

After initializing T_s^0 and T_l^0 in Ω^- and Ω^+, we quadratically extend those quantities to a small band around the interface, i.e., we extend T_s^0 from Ω^- to Ω^+ and extend T_l^0 from Ω^+ to Ω^-. The extrapolation procedures used are detailed in [26] and are an extension to quadtrees of the technique introduced by Aslam [7] on uniform grids.

The interface is evolved by solving Eq. (7) using the semi-Lagrangian method in Section 4.2. We note that Eq. (5) is only valid exactly on the interface so that the velocity field $\mathbf{V}$ needs to be extended to nodes on each side of the interface by constant extrapolation in the direction normal to the interface using extrapolation procedures given in [26].

Then the heat equations are solve in both Ω^- and Ω^+ with the new interface given by the zero contours of ϕ at time t^{n+1}. Dirichlet boundary conditions are imposed on the interface using the Gibbs–Thomson relation in Eq. (4). On the boundary of the computational domain, $\partial\Omega$, either Dirichlet or Neumann boundary conditions can be imposed. When computing the Gibbs–Thomson relation in Eq. (4) we use the value of the normal velocity V at time n. As detailed in section [14,13], the interface may sweep some grid nodes from time t^n to t^{n+1} so that the temperature at these nodes needs to be extrapolated as well.

To summarize, our algorithm to solve the Stefan problems is outlined in Algorithm 2.

Algorithm 2. Procedure to solve the Stefan problem

```
1.  Initialize $\phi$ as a signed distance function
2.  Initialize $T_s$ in $\Omega^-$, and $T_l$ in $\Omega^+$
3.  Quadratically extrapolate $T_s^n$ from $\Omega^-$ to $\Omega^+$ and $T_l^n$ from $\Omega^+$ to $\Omega^-$
4.  Calculate the velocity $\mathbf{V}$ at grid nodes and constantly extrapolate it
     away from the interface
5.  Evolve the interface by solving Eq. (7) for $\phi^{n+1}$, and reinitialize
     it as a signed distance function, using (9)
6.  Solve the heat equations in $\Omega^-$ and $\Omega^+$ for $T_s^{n+1}$ and $T_l^{n+1}$, using the
     Gibbs–Thomson relation (4) as the Dirichlet boundary condition on
     the interface
7.  Regenerate the adaptive grid using $\phi^{n+1}$, and quadratically interpolate
     $T_s^{n+1}, T_l^{n+1}$ and $\phi^{n+1}$ onto the new grid
8.  if (the final time is not reached)
       go to 3 with $n = n + 1$
     end if
```

5. Numerical results for the Stefan problem

5.1. Accuracy and efficiency

We discuss the efficiency and accuracy of our numerical method on the known Frank-sphere exact solution [12]: In two spatial dimensions, the region described by a disk with zero temperature, is growing into the supercooled liquid. The radius of the growing disk is given by $R(t) = s_0\sqrt{t}$, parameterized by s_0. The temperature field is given by

$$T(r,t) = T(s) = \begin{cases} 0 & \text{if } s \leqslant s_0 \\ T_\infty\left(1 - \frac{F(s)}{F(s_0)}\right) & \text{if } s > s_0, \end{cases} \tag{19}$$

where r is the distance to the center of the disk, $s = r/\sqrt{t}, F(s) = E_1(s^2/4)$, with $E_1(z) = \int_z^\infty \frac{e^{-t}}{t} dt$. Here, $F(s)$ is the similarity solution of the heat equation, the coefficient s_0 and T_∞ represent the original radius and the temperature infinitely far from the disk's boundary, respectively. They are related by the jump condition at the freezing front,

$$V_n = -\left[\frac{\partial T}{\partial \mathbf{n}}\right], \tag{20}$$

where V_n is the normal velocity at the solid–liquid interface and equal to

$$V_n = \frac{dR}{dt} = \frac{s_0}{2\sqrt{t}}. \tag{21}$$

Combining Eqs. (19)–(21) yields the relation between T_∞ and s_0:

$$T_\infty = \frac{1}{2} s_0 F(s_0)/F'(s_0). \tag{22}$$

Fig. 4 illustrates the evolution of the interface, as well as that of the adaptive grid, at different times. The final time is $t = 10$, which demonstrates the robustness of our algorithm for large time computations. The computation domain is $\Omega = [-2,2] \times [-2,2]$ and the other parameters used in the computation are: $s_0 = 0.25, T_\infty = -0.05709187113307$, (MinRes, MaxRes) = (8,64). The time step is $\triangle t = \triangle x$ where $\triangle x$ is the size of the finest cell.

To demonstrate the order of accuracy of our numerical method, we simulate the growth of a two dimensional Frank sphere up to $t = 2$ in a domain $\Omega = [-2,2] \times [-2,2]$, with different choices of (MinRes, MaxRes). The other parameters used are: $s_0 = 0.25, T_\infty = -0.05709187113307$ and $\triangle t = 2 \times \triangle x$ where $\triangle x$ is the size of the finest cell. The adaptive grids at $t = 1$ with different (MinRes, MaxRes) are illustrated in Fig. 5. Tables 1–8 give the error for the interface location (error in ϕ) and

Table 1
Accuracy results on uniform grids for ϕ in the Frank sphere simulation.

Resolution	L^∞ error	Order	L^1 error	Order
32×32	5.197×10^{-2}	–	3.253×10^{-2}	–
64×64	1.489×10^{-2}	1.804	1.100×10^{-2}	1.564
128×128	5.395×10^{-3}	1.464	3.535×10^{-3}	1.638
256×256	1.737×10^{-3}	1.635	1.100×10^{-3}	1.684

Table 2
Accuracy results on uniform grids for T in the Frank sphere simulation.

Resolution	L^∞ error	Order	L^1 error	Order
32×32	3.016×10^{-3}	–	5.395×10^{-4}	–
64×64	1.094×10^{-3}	1.463	1.739×10^{-4}	1.633
128×128	4.476×10^{-4}	1.290	5.523×10^{-5}	1.655
256×256	1.498×10^{-4}	1.579	1.442×10^{-5}	1.938

Table 3
Accuracy results on adaptive grids for ϕ in the Frank Sphere simulation with MaxRes/MinRes = 2.

(*MinRes*, *MaxRes*)	L^∞ error	Order	L^1 error	Order
(16,32)	5.503×10^{-2}	–	3.504×10^{-2}	–
(32,64)	1.492×10^{-2}	1.883	1.102×10^{-2}	1.669
(64,128)	5.410×10^{-3}	1.463	3.541×10^{-3}	1.638
(128,256)	1.750×10^{-3}	1.629	1.112×10^{-3}	1.672

Table 4
Accuracy results on adaptive grids for T in the Frank sphere simulation with MaxRes/MinRes = 2.

(*MinRes*, *MaxRes*)	L^∞ error	Order	L^1 error	Order
(16,32)	3.223×10^{-3}	–	8.818×10^{-4}	–
(32,64)	1.100×10^{-3}	1.552	2.314×10^{-4}	1.930
(64,128)	4.490×10^{-4}	1.292	7.122×10^{-5}	1.700
(128,256)	1.509×10^{-4}	1.573	1.860×10^{-5}	1.937

Table 5
Accuracy results on adaptive grids for ϕ in the Frank sphere simulation with MaxRes/MinRes = 4.

$(MinRes, MaxRes)$	L^∞ error	Order	L^1 error	Order
(8,32)	5.522×10^{-2}	–	3.519×10^{-2}	–
(16,64)	1.494×10^{-2}	1.886	1.102×10^{-2}	1.676
(32,128)	5.501×10^{-3}	1.441	3.661×10^{-3}	1.589
(64,256)	1.801×10^{-3}	1.611	1.134×10^{-3}	1.691

Table 6
Accuracy results on adaptive grids for T in the Frank sphere simulation with MaxRes/MinRes = 4.

$(MinRes, MaxRes)$	L^∞ error	Order	L^1 error	Order
(8,32)	3.231×10^{-3}	–	1.397×10^{-3}	–
(16,64)	1.093×10^{-3}	1.564	3.756×10^{-4}	1.895
(32,128)	4.567×10^{-4}	1.259	1.172×10^{-4}	1.680
(64,256)	1.552×10^{-4}	1.557	3.059×10^{-5}	1.938

Table 7
Accuracy results on adaptive grids for ϕ in the Frank sphere simulation with MaxRes/MinRes = 8.

$(MinRes, MaxRes)$	L^∞ error	Order	L^1 error	Order
(4,32)	5.521×10^{-2}	–	3.519×10^{-2}	–
(8,64)	1.490×10^{-2}	1.890	1.098×10^{-2}	1.680
(16,128)	5.559×10^{-3}	1.423	3.656×10^{-3}	1.586
(32,256)	1.924×10^{-3}	1.530	1.209×10^{-3}	1.597

Table 8
Accuracy results on adaptive grids for T in the Frank sphere simulation with MaxRes/MinRes = 8.

$(MinRes, MaxRes)$	L^∞ error	Order	L^1 error	Order
(4,32)	3.230×10^{-3}	–	1.484×10^{-3}	–
(8,64)	1.095×10^{-3}	1.560	5.614×10^{-4}	1.403
(16,128)	4.580×10^{-4}	1.258	1.661×10^{-4}	1.757
(32,256)	1.659×10^{-4}	1.465	5.338×10^{-5}	1.638

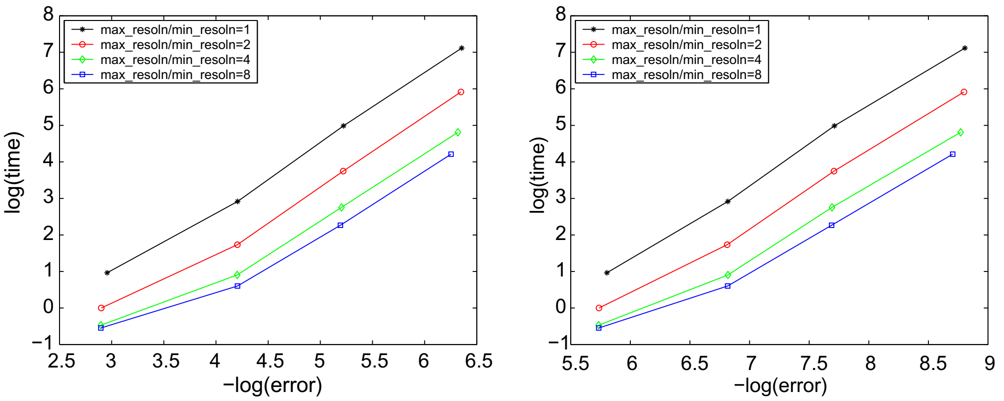
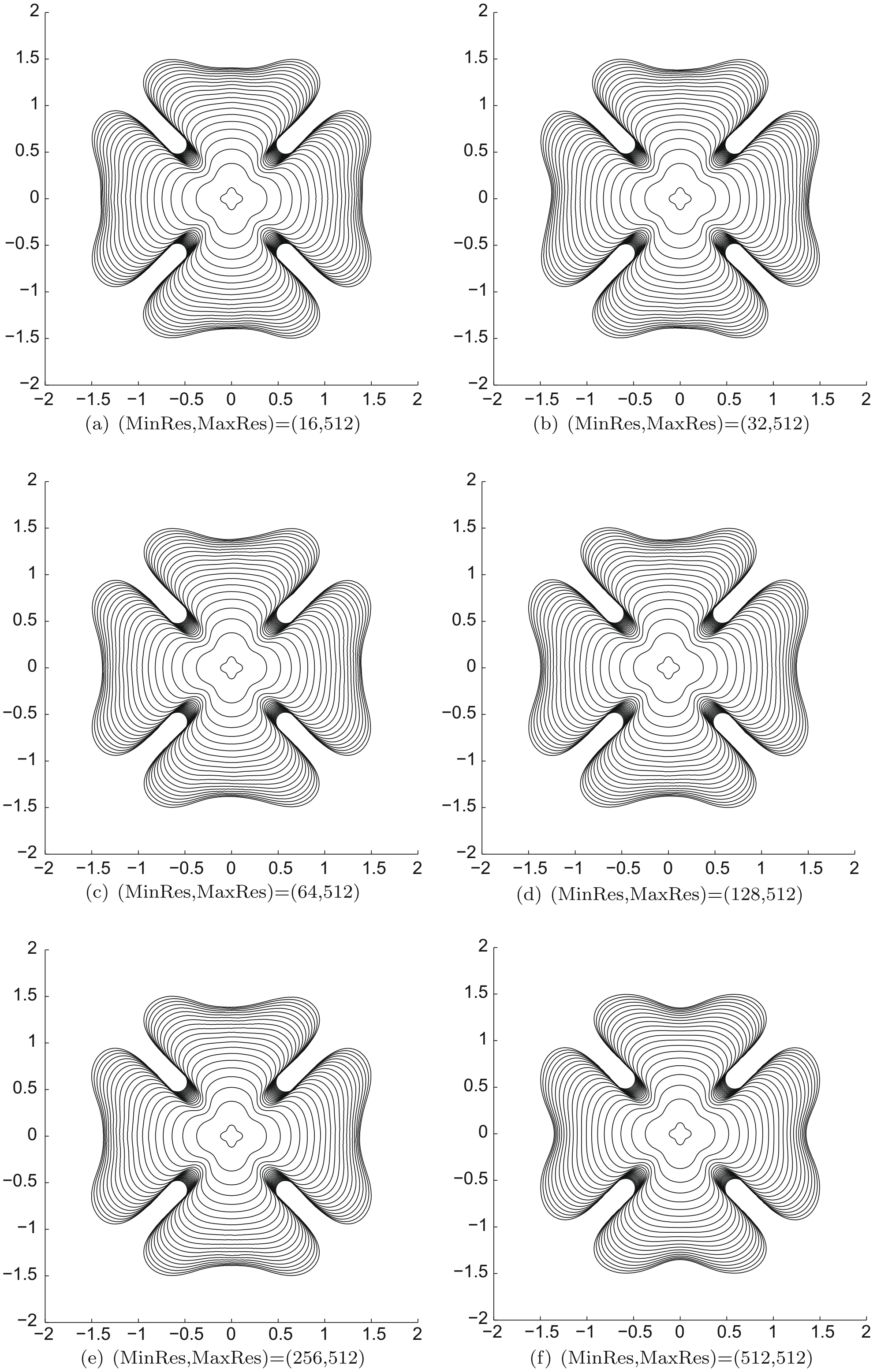


Fig. 6. Log–log plot of the computational time as a function of maximum error in ϕ (left) and T (right). For these tested grid resolutions, the error is mainly determined by MaxRes, confirming that a coarser grid resolution can be used at locations far away from the interface (parabolicity of the Stefan problem).

the error for the temperature field T in both the L^1 and L^∞ norms. We note that all the computation are carried out to second-order accuracy but that the resulting overall solution has a lower convergence rate (≈ 1.5). This can be explained by the different inaccuracy in extrapolation and reinitialization procedures that are not iterated to steady state.



(a) (MinRes,MaxRes)=(16,512)

(b) (MinRes,MaxRes)=(32,512)

(c) (MinRes,MaxRes)=(64,512)

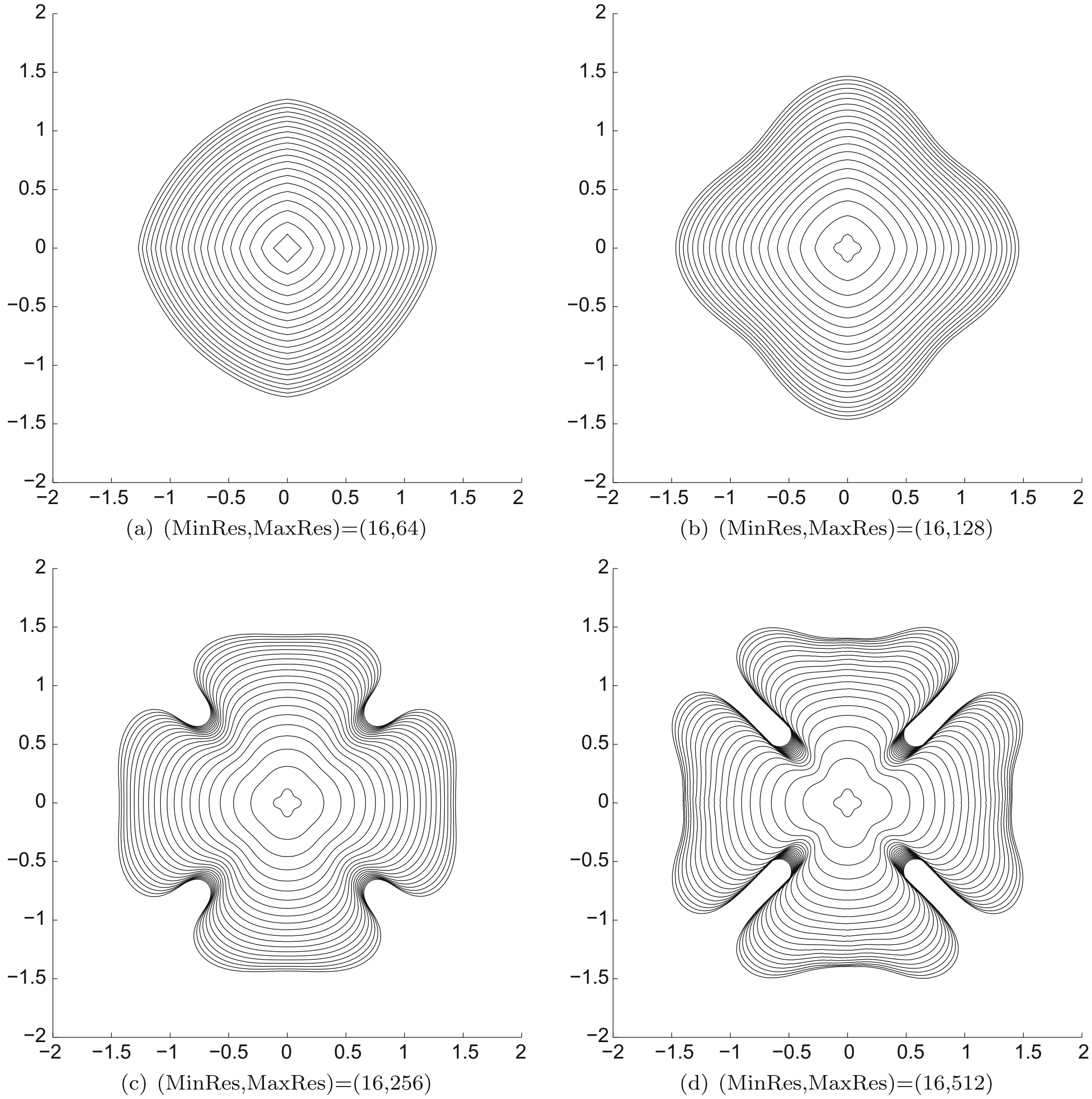
(d) (MinRes,MaxRes)=(128,512)

(e) (MinRes,MaxRes)=(256,512)

(f) (MinRes,MaxRes)=(512,512)

Fig. 7. Growth histories of the interface in example 5.3 with MaxRes = 512 and MinRes increasing from 16 to 512. Time levels shown are in uniform increments from $t = 0$ to $t = 0.8$.

To demonstrate the saving of computational efforts using adaptive grids, we plot the computational time on a 1.6 GHz laptop as a function of the maximum error in ϕ and T, in Fig. 6. If we define the degree of adaptivity as the ratio of MaxRes to MinRes, a uniform grid will have a degree of adaptivity equal to one. From these figures, we can see that decreasing MinRes does not deteriorate the accuracy of the solutions. This is consistent the parabolic nature of the partial differential equation, which can produces smooth solution far away from the interface. This implies that the maximum errors in both ϕ and T are mainly determined by MaxRes, which occurs near the interface (kink in the solution). The computational time on adaptive grids can be several orders of magnitude less than that on uniform grids with the same MaxRes, while equivalent accuracy can be achieved. These results clearly justify our choice of adaptivity criterion. The saving in computational time decreases when the adaptivity is increased, and this is due to the fact that decreasing MinRes when the adaptivity is high does not decrease the total number of grid nodes as much as when the grid is uniform or the adaptivity is low. We also note that the maximum error in ϕ or T is increasing when we fix MaxRes and decrease MinRes, meaning that the errors on the coarsest grid cells will gradually influence the overall accuracy on the whole grid and the coarsest grid resolution will finally put a cap on the maximum accuracy that can be obtained, regardless of the choice of the finest grid resolution. The results in Tables 2 and 7 suggest that a ratio Max/Min = 8 gives similar results as that of uniform grids, with a significant gain in efficiency (significantly smaller linear system). We note that we always use the same BiCGSTAB solver, even for symmetric linear systems (which are obtained with uniform grids), for which faster solvers such as PCG could be used [16,31]. More



(a) (MinRes,MaxRes)=(16,64)

(b) (MinRes,MaxRes)=(16,128)

(c) (MinRes,MaxRes)=(16,256)

(d) (MinRes,MaxRes)=(16,512)

Fig. 8. Grid refinement study: growth histories of the interface in example 5.3 with various values of MaxRes. MinRes = 16 and is fixed. Time levels shown are in uniform increments from $t = 0$ to $t = 0.8$.

efficient solvers such as multigrid methods should be used so that the difference between solving symmetric vs. nonsymmetric system does not differ by a large factor.

5.2. Numerical results for unstable solidification

We now present some numerical simulation results of the growth of a solid seed into an undercooled liquid. In all the following examples, The temperature field is initialized uniformly as the Stefan number *St*, i.e., $T = St < 0$ in the liquid region, and $T = 0$ in the solid region. Unless otherwise stated, adiabatic boundary conditions are imposed on the four sides of the computational domain Ω, and we assume that the diffusion constant is the same in both phases.

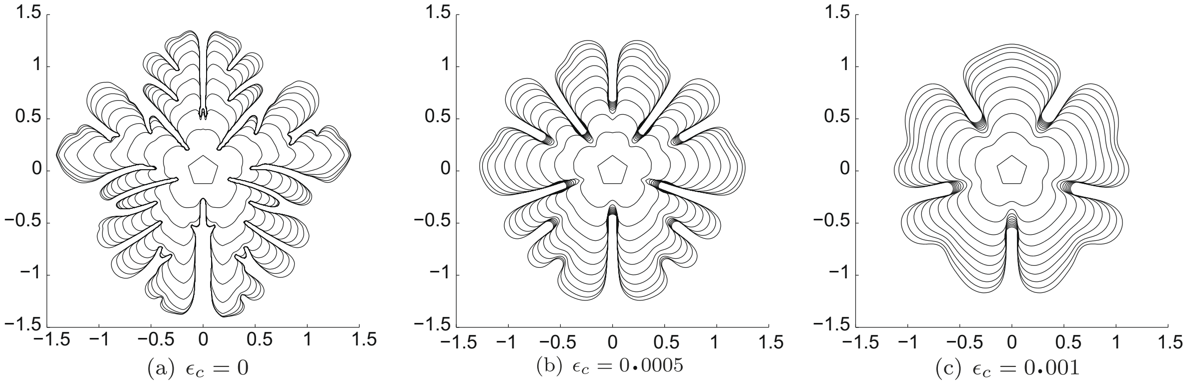
5.3. Role of grid resolutions on developing features

This example tests the role of (MinRes, MaxRes) in the computational results of dendritic solidification. Consider the growth of a small solid seed placed in a surrounding region of undercooled liquid with the Stefan number $St = -0.5$. Our computational domain is $\Omega = [-2, 2] \times [-2, 2]$ and the initial shape of the solid seed is given by the following parametric equations,

$$\begin{aligned} x(s) &= (R + P\cos(8\pi s))\cos(2\pi s), \\ y(s) &= (R + P\cos(8\pi s))\sin(2\pi s), \end{aligned} \tag{23}$$

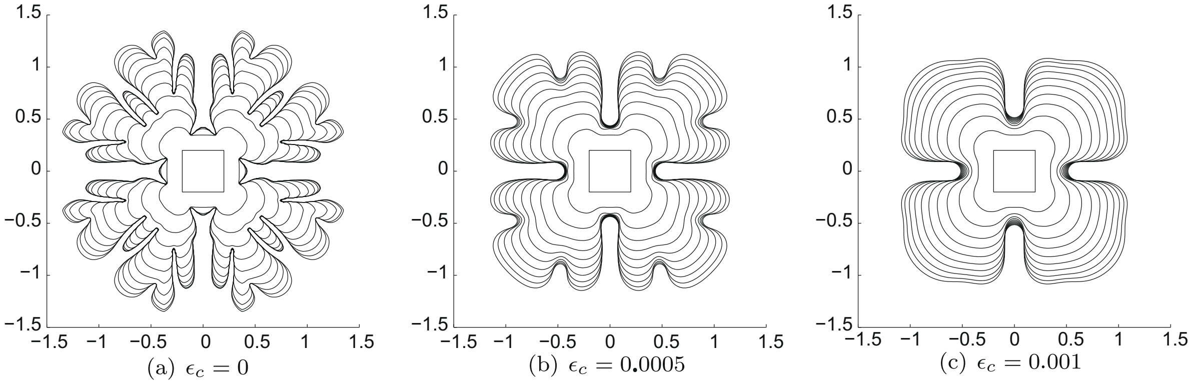
where $R = 0.1, P = 0.02$ and $s \in [0, 2\pi]$. The Gibbs–Thomson relation, Eq. (4), with $\epsilon_c = 0.002$ and $\epsilon_v = 0.002$ is applied on the solid–liquid interface. Adiabatic boundary conditions are imposed on the four sides of Ω.

Fig. 7 compares the growth histories of the interface with *max_resoln* = 512 and MinRes increasing from 16 to 512. No difference is observed between the results obtained on these different grids. This demonstrates that a minimum resolution of MinRes = 16 is enough to capture the smooth features of the solution far away from the interface and grid refinement can



(a) $\epsilon_c = 0$ (b) $\epsilon_c = 0.0005$ (c) $\epsilon_c = 0.001$

Fig. 9. Effect of varying isotropic surface tension. The initial solid seed is a regular pentagon. We impose the Gibbs–Thomson relation 4 at the interface with $\epsilon_v = 0$ and different ϵ_c values. Time levels shown are in uniform increments from $t = 0$ to $t = 0.4$.



(a) $\epsilon_c = 0$ (b) $\epsilon_c = 0.0005$ (c) $\epsilon_c = 0.001$

Fig. 10. Effect of varying isotropic surface tension. The initial solid seed is a square. We impose the Gibbs–Thomson relation 4 at the interface with $\epsilon_v = 0$ and different ϵ_c values. Time levels shown are in uniform increments from $t = 0$ to $t = 0.4$.

be done by only increasing MaxRes. Increasing MaxRes does not increase much the total number of computational nodes since the region near the interface is proportional to a one dimensional problem.

Fig. 8 illustrates the growth histories of the interface from $t = 0$ to $t = 0.8$, with various values of MaxRes. Time levels shown are in uniform increments from $t = 0$ up to a final time of $t = 0.8$. In these computations, the time step is $\triangle t = 0.001$. The convergence of these plots under grid refinement is similar to that in [11]. But compared with those in [18,39,40,14], Fig. 8 has flatter boundaries and doesn't exhibit pronounced secondary branches.

5.3.1. *Effect of varying isotropic surface tension*

Figs. 9 and 10 demonstrate the effect of isotropic surface tension by In this example, we impose the Gibbs–Thomson relation $T = -\epsilon_c\kappa$ on the interface with varying values of ϵ_c. The computational domain is $\Omega = [-1.5, 1.5] \times [-1.5, 1.5]$, the undercooled liquid has a Stefan number of $St = -0.5$ and the time step is $\triangle t = 0.004$. Moving adaptive grids with (MinRes, MaxRes) = (32, 256) are used in our simulation. Fig. 9 depicts the growth histories of a regular pentagon solid seed into

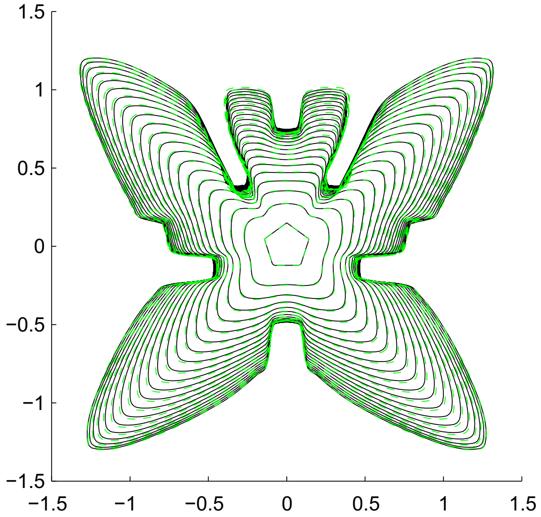
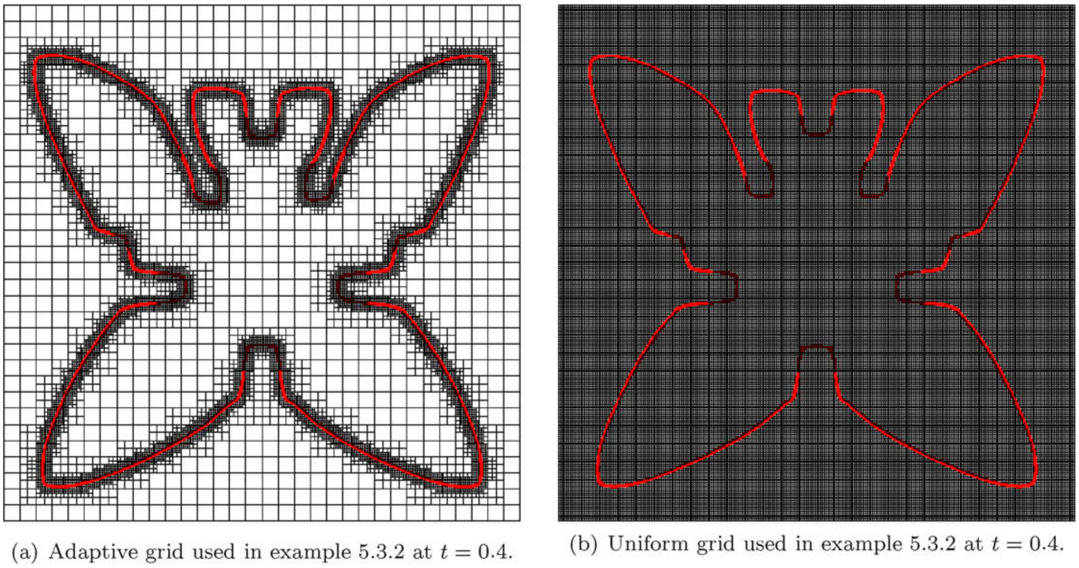


Fig. 11. Effect of anisotropic surface tension. The Gibbs–Thomson relation $T = -0.001(8/3\sin^4(2\alpha - \pi/2))$ is imposed on the interface. The solid lines depict the interface growth histories on a uniform 256 × 256 grid, and the dashed lines depict the interface growth histories on adaptive moving grids with (MinRes, MaxRes) = (32, 256). Time levels shown are in uniform increments from $t = 0$ to $t = 0.4$.


(a) Adaptive grid used in example 5.3.2 at $t = 0.4$.

(b) Uniform grid used in example 5.3.2 at $t = 0.4$.

Fig. 12. Comparison of the grids used in example 5.3.2.

the supercooled liquid. Fig. 10 depicts the growth histories of a square solid seed into the supercooled liquid. With increasing values of ϵ_c, the interface becomes smoother. These results demonstrate the capability of our method to capture the smoothing effect of surface tension on the interface growth.

5.3.2. *Effect of anisotropic surface tension*

Anisotropic surface tension forces the solid seed to grow along preferred directions. In this example, the initial seed is a regular pentagon, placed in an undercooled liquid with the Stefan number $St = -0.5$. The Gibbs–Thomson relation we impose on the interface is $T = -0.001(8/3\sin^4(2\alpha - \pi/2))\kappa$, where α is the angle between the normal to the interface and the x-axis. Fig. 11 illustrates the evolution of the interface on both a uniform 256×256 grid and adaptive moving grids with (MinRes,MaxRes) = (32,256). Our computational domain is $\Omega = [-1.5, 1.5] \times [-1.5, 1.5]$ and the time step in the simulation is $\triangle t = 0.001$. We can see that the fourfold anisotropy of the Gibbs–Thomson relation leads to fastest growth along the four

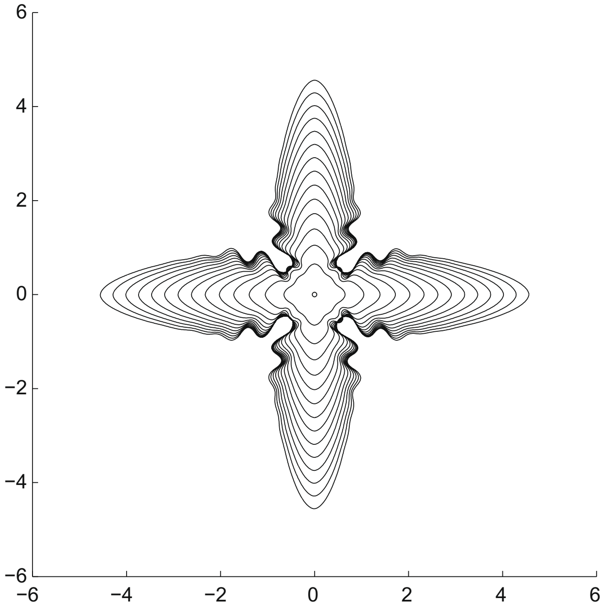


Fig. 13. Growth histories of a circular seed under standard four-fold anisotropic surface tension.

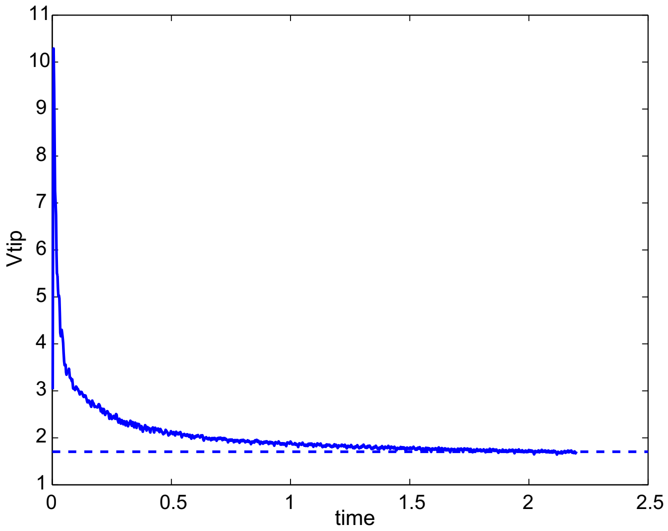


Fig. 14. Tip velocity of a circular seed growing under standard four-fold anisotropic surface tension as a function of time. The tip velocity reaches a steady state value of 1.7.

diagonal directions. The branch in the positive y direction, as well as its secondary branches, is due to the upward sharp corner of the initial seed. In Fig. 11, the results obtained on the uniform grid and adaptive moving grids are almost identical. In Fig. 12, we plot the adaptive and uniform grids at the final time $t = 0.4$, in which the number of nodes in the case of the adaptive grid is only 23% of that of the uniform grid. To obtain the results in Fig. 11, the computational time when (MinRes, MaxRes) = (32, 256) on adaptive grids is only about 15% of that the uniform 256×256 grid.

5.3.3. Comparison with the microscopic solvability predictions

In this example, we assess the accuracy of our method in predicting the tip growth rate of a dendrite. Numerical simulation is performed for a circular seed of radius 0.05 at the center of the computational domain $\Omega = [-6, 6] \times [-6, 6]$. The undercooled liquid has a Stefan number of $St = -0.45$. We impose the Gibbs–Thomson relation (4) on the solid–liquid interface, where $\epsilon_v = 0$ and $\epsilon_c = -0.001[1 + 0.4(1 - \cos 4\alpha)]$, where α is the angle between the normal to the interface and the x-axis. Fig. 13 depicts the evolution of the interface from $t = 0$ to $t = 2.2$ on a (MinRes, MaxRes) = (64, 1024) moving grids. In this simulation, $\triangle t = 0.002$. Fig. 14 plots the tip velocity as a function of time. The tip velocity reaches a steady state value of 1.7, in agreement with solvability theory [24].

6. Conclusions

We have presented a level set approach to the numerical simulation of the Stefan problem on non-graded adaptive Cartesian grids, i.e. grids for which the size ratio between adjacent cells is not constrained. Compared with uniform grids, adaptive grids relocate computational resources efficiently by refining only where needed, thus reducing both the amount of computational time and the memory usage significantly. Compared with traditional graded adaptive grids, the non-graded property of our adaptive grids allows faster and more flexible grid generation. We discretize our computational domain using the optimal quadtree data structure, and a simple recursive algorithm is used to generate the adaptive grids automatically. Since in the Stefan problem large errors are usually observed near the interface and dendritic structures and other physical/geometrical properties of the interface need to be captured accurately at the interface, we construct our adaptive grid such that finest cells are placed around the interface to form a uniform band, and coarser cells are placed farther away form the interface. Numerical examples are presented to confirm this choice and validate the efficiency of such grids for the Stefan problem.

We demonstrate the supra-convergence for both the temperature field and the evolution of the interface by comparing with exact Frank-sphere solutions. On our adaptive grids, the computational time can be saved by several orders of magnitude while the same accuracy in the solution is achieve; if the same amount of computational time is used, our solution can be several orders of magnitude more accurate compared with that on uniform grids. We also present numerical examples to demonstrate that our method's capability to capture the effects of many physical and numerical parameters in the Stefan problem. Specifically, we study the effect of grid resolutions, the smoothing effect of isotropic surface tension, and the effect of anisotropic surface tension. Comparison with the microscopic solvability prediction is performed for a standard four-fold anisotropic problem to confirm the accuracy of our method.

Acknowledgments

The research of H. Chen and F. Gibou was supported in part by a Sloan Research Fellowship in Mathematics, by the National Science Foundation Under Grant Agreement DMS 0713858 and by the Department of Energy Under Grant Agreement DE-FG02-08ER15991. The research of C. Min was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2008-331-C00045).

References

[1] M.J. Aftosmis, M.J. Berger, J.E. Melton, Adaptive Cartesian mesh generation, in: CRC Handbook of Mesh Generation (Contributed Chapter), 1998.

[2] V. Alexiades, A.D. Solomon, Mathematical Modeling of Melting and Freezing Processes, Hemisphere, Washingotn, DC, 1993.

[3] V. Alexiades, A.D. Solomon, D.G. Wilson, The formation of a solid nucleus in supercooled liquid. i, J. Non-Equil. Thermody. 13 (1988) 281–300.

[4] A. Almgren, A Fast Adaptive Vortex Method Using Local Corrections, Ph.D. Thesis, University of California, Berkeley, 1991.

[5] A. Almgren, J. Bell, P. Colella, L. Howell, M. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, J. Comput. Phys. 142 (1998) 1–46.

[6] A. Almgren, R. Buttke, P. Colella, A fast adaptive vortex method in three dimensions, J. Comput. Phys. 113 (1994) 177–200.

[7] T. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2004) 349–355.

[8] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64–84.

[9] M. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484–512.

[10] H. Chen, C. Min, F. Gibou, A supra-convergent finite difference scheme for the poisson and heat equations on irregular domains and non-graded adaptive Cartesian grids, J. Sci. Comput. 31 (2007) 19–60.

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

[12] F. Frank, Proc. Royal Soc. A 201 (1950) 586.

[13] F. Gibou, R. Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. 202 (2005) 577–601.

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

[15] F. Gibou, R. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (2002) 205–227.

[16] G. Golub, C. Loan, Matrix Computations, The John Hopkins University Press, 1989.

[17] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, J. Comput. Phys. 147 (1998) 60–85.

[18] D. Juric, G. Tryggvason, A front tracking method for dendritic solidification, J. Comput. Phys. 123 (1996) 127–148.

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

[20] J. Langer, in: J. Souletie, J. Vannimenus, R. Stora (Eds.), Lectures in the Theory of Pattern Formation, Chance and Matter, North Holland, Amsterdam, 1987.

[21] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 126 (1996) 202–212.

[22] P. McCorquodale, P. Colella, D. Grote, J.-L. Vay, A node-centered local refinement algorithm for Poisson's equation in complex geometries, J. Comput. Phys. 201 (2004) 34–60.

[23] D. Meiron, Selection of steady-states in the two-dimensional symmetric model of dendritic growth, Phys. Rev. A 33 (1986) 2704.

[24] D. Meiron, Selection of steady states in the two-dimensional symmetric model of dendritic growth, Phys. Rev. A 33 (1986) 2704–2715.

[25] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equation on non-graded adaptive grids, J. Comput. Phys. 219 (2006) 912–929.

[26] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, J. Comput. Phys. 225 (2007) 300–321.

[27] C. Min, F. Gibou, H. Ceniceros, A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids, J. Comput. Phys. 218 (2006) 123–140.

[28] C.-H. Min, Local level set method in high dimension and codimension, J. Comput. Phys. 200 (2004) 368–382.

[29] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, New York, NY, 2002.

[30] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.

[31] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing, New York, NY, 1996.

[32] H. Samet, The Design and Analysis of Spatial Data Structures, Addison-Wesley, New York, 1989.

[33] H. Samet, Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS, Addison-Wesley, New York, 1990.

[34] J. Sethian, J. Strain, Crystal growth and dendritic solidification, J. Comput. Phys. 98 (1992) 231–253.

[35] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, Cambridge, 1999.

[36] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, J. Comput. Phys. 77 (1988) 439–471.

[37] M. Sussman, A.S. Algrem, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, An adaptive level set approach for incompressible two-phase flow, J. Comput. Phys. 148 (1999) 81–124.

[38] 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.

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

[40] P. Zhao, J. Heinrich, Front tracking finite element method for dendritic solidification, J. Comput. Phys 173 (2001) 765–796.