An Adaptive Level Set Method for Stefan Problems

Shingyu Leung [*], John Lowengrub [†]& Stanley Osher [‡]

April 12, 2004

Abstract

An adaptive level set method is implemented to solve the quasi-steady state approximation of the Stefan problem. The grid adaptivity is based on a minimization of the Winslow functional with a monitor function coupled with the level set function. Temperature field is computed by applying techniques from Ghost-Fluid Method [17, 5] to the Laplace equation in the transformed coordinate system. By extending this temperature using fast-sweeping method [21, 8, 9], the normal velocity of the interface is obtained and is used to update the level set function. Several numerical examples will also be included to demonstrate the computational efficiency of the resulting algorithm.

1 Introduction

Crystal growth is a classical problem for phase transition from liquid to solid through heat transfer. The process of this growth starts with a small solid seed put into a bath of undercooled liquid. The heat is then transferred to the surrounding and this mechanism expands the solid and the instability in the growth of the interface would drive the crystal into a dendritic shape. The resulting shape of this growth depends on the initial profile of the seed, the surrounding temperature, the normal velocity of the interface propagation, and more importantly, the surface tension of the interface between solid and liquid phases. Understanding the relationship between these parameters and the resulting dendritic shape is important. One reason is that the properties of the final alloy, like the conductivity or the strength, strongly depend on how the crystal evolves.

Simulation of the dendritic growth is difficult numerically due to the following reasons. The numerical method should be able to

[*]Department of Mathematics, UCLA, Los Angeles, CA 90095. Email: **syleung@math.ucla.edu**
[†]Department of Mathematics, UCI, Irvine, CA 92697. Email: **lowengrb@math.uci.edu**
[‡]Department of Mathematics, UCLA, Los Angeles, CA 90095. Email: **sjo@math.ucla.edu**

1. solve the heat (or Laplace) equation on an arbitrary domain with irregular boundaries;
2. evolve the interface according to some physical laws;
3. handle the topological changes of the interface easily;
4. solve the problem in a reasonable time.

Different numerical schemes have been introduced and applied to this problem. They can be categorized into the following three groups.

1. Front-tracking method [7]. This is a Lagrangian-type method in which particles are put on the interface and are advected by the interface velocity. The location of the interface is then represented explicitly, and therefore, the heat (or Laplace) equation can be approximated by finite difference. Discontinuities of the material properties between solid and liquid can be handled easily. However, new particles have to be added to keep the interface resolutions sharp. Merging of the interface can also be done, but special care is needed. The situation with no interface kinetics is also difficult to deal with.
2. Phase-field method [10, 11, 12]. The interface is smeared out. A phase-variable ϕ is introduced and $\phi \to 0$ for the location in the solid phase away from the interface and $\phi \to 1$ for the location in the liquid phase away from the interface. The motion laws for the interface and the heat equation are modified to accommodate with this phase variable. This results in an Eulerian method. The interface is represented implicitly and can be calculated from the phase variable ϕ when necessary. Therefore, topological changes in the interface can be easily taken care of. The problem of this approach is that the phase-field method only reproduces the dynamics of the interface for the sharp interface model in the limit when Δx goes to zero. An extra level of grid adaptivity is needed to resolve this induced transition layer [16]. Moreover, when the physical properties between solid and liquid are different, it is not straight forward to apply the method [13].
3. Level Set Method [18, 3, 13, 5, 4]. The interface is represented as the zero level set of a level set function ϕ. This level set function is chosen initially to be a signed distance function from the interface. Unlike the phase-field method, the level set function will then be modified with its own motion law. As a result, there is no arbitrary interface width introduced in the model and the sharp interface model can be solved directly. Discontinuous properties of materials can also be handled easily.

However, one draw back of using Level Set Method is the numerical dissipation inherited from finite difference methods: to increase the accuracy of the solution, one need to increase the number of grid points by, say, doubling the grids in all directions. Some of the test cases in the references need almost 400 grids in each direction to get a reasonable solution of the crystal structure. This is not efficient in the sense that many grid points are far away from the interface and the solutions there are not directly used at all in obtaining the normal velocity for updating the zero level set.

Resolution of solutions using Level Set Method would also be vary limited by the grid size. It should be noticed that the zero level set cannot represent regions where the magnitude of the curvature is great than $1/\Delta x$. Therefore, to get an accurate interface in complex regions, one cannot only simply increases the order of the numerical method in the evolution stage, but needs to increase the total number of grid points. This means, the local resolution of the interface would result in a global uniform increase in the number of grids. This can lower the efficiency of the Level Set Method.

The aim of this paper is to improve the computational efficiency of the level set approach as in [3, 5] by applying an adaptive moving grid approach. Since this adaptive approach is based on structured quadrilateral grids [14, 6, 20, 19], simple finite difference method can be used. With this novel, efficient adaptive method in place, more physical quantities can be incorporated in the model.

The rest of the paper is organized as follow. The quasi-steady Stefan Problem will be given in Section 2. The numerical procedures will be explained in details in Section 3. Section 4 contains several numerical examples to test the convergence rates and the efficiency of the adaptive approach.

2 The Quasi-steady Stefan Problem

The classical Stefan problem is given by

$$\begin{aligned} \frac{\partial T}{\partial t} &= k_s \nabla^2 T \quad \text{in} \quad \Omega_s \\ \frac{\partial T}{\partial t} &= k_l \nabla^2 T \quad \text{in} \quad \Omega_l \,, \end{aligned} \tag{1}$$

with T denoting both the temperature of the crystal in Ω_s and that of the supercooled liquid Ω_l, separated by the interface Σ. A level set function ϕ is introduced such that the zero level set of this function gives Σ. k_s and k_l are related to volumetric heat capacities and thermal diffusivities of the materials. The Gibbs-Thomson relation is imposed on the interface,

$$T = -\sigma\kappa - \mu v_n \text{ on } \Sigma \,, \tag{2}$$

where σ is the surface tension coefficient, κ is the curvature of the interface, and μ is the molecular kinetic coefficient. The normal velocity of the interface is computed from

$$v_n = [\mathbf{n} \cdot \nabla T] \text{ on } \Sigma \,, \tag{3}$$

where [.] is defined as the jump across the interface, i.e. $[f] = f_s|_\Sigma - f_l|_\Sigma$. This normal velocity will then be extended throughout the whole domain and be used to move the level set function according to the level set equation

$$\phi_t + v_n |\nabla \phi| = 0 \text{ in } \Omega \,. \tag{4}$$

If we assume that the diffusion coefficients k_s and k_l are large, then we obtain the following quasi-steady Stefan problem

$$\begin{aligned} \nabla^2 T = 0 \quad &\text{in} \quad \Omega \\ T = -\sigma\kappa - \mu v_n \quad &\text{on} \quad \Sigma \\ v_n = [\mathbf{n} \cdot \nabla T] \quad &\text{on} \quad \Sigma \\ \phi_t + v_n |\nabla \phi| = 0 \quad &\text{in} \quad \Omega\,. \end{aligned} \tag{5}$$

3 Numerical Method

We will focus on the two-dimensional case. However, this idea can be applied to higher dimensional space in a straight forward manner.

The numerical procedures will be mainly composed of the following steps:

1. Adapt the grid according to the level set function ϕ [14, 19].
2. Solve the Laplace equation with the boundary condition imposed on the interface [17, 5].
3. Update the location of the interface by solving the level set equation.

3.1 Grid Adaptation

The adaptivity of the grid can be seen as determining a time dependent map between the physical domain (x, y) and the computational domain (ξ, η). The computational domain is fixed so we can think of it as the indexing space of the discretization in the physical space. This means $(\xi, \eta) \in \{1, 2, \ldots, n_\xi\} \times \{1, 2, \ldots, n_\eta\}$ with $\Delta\xi = \Delta\eta = 1$. We write this desired mapping as $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$.

Given a level set function ϕ^n defined on the grids $(x^n_{i,j}, y^n_{i,j})$ at time t^n, we would like to find a new set of structured grids $(x^{n+1}_{i,j}, y^{n+1}_{i,j})$ with grid points concentrated around the zero level set. A simple idea to find such a map is to minimize Winslow's functional defined in the computational domain

$$\min_{(x,y)} E(x(\xi,\eta), y(\xi,\eta)) = \frac{1}{2} \int \omega \left(|\nabla_{\xi,\eta} x|^2 + |\nabla_{\xi,\eta} y|^2 \right) d\xi d\eta\,, \tag{6}$$

where ω in the equation is called the monitor function and controls how the grids will be adapted. From the computational examples below, it is found that

$$\omega(\phi) = 1 + \alpha \exp(-\phi^2/\sigma^2)\,, \tag{7}$$

with $\alpha = 10$ and $\sigma = 0.1$, is a reasonable choice. This monitor function is symmetric in $\phi = 0$, meaning that the grid points should be evenly distributed around the zero level set independent of the sign of the function. Another property is that this function has its largest value when $\phi = 0$, therefore the concentration of the grid points decreases

away from the zero level set. In [19], the authors used a Heaviside level set function, instead of a signed distance level set function. They proposed

$$\omega(\psi) = \sqrt{1 + |\nabla_{x,y}\psi|^2}\,, \tag{8}$$

with $\psi = \text{sign}(\phi)$. In our case, the monitor function is chosen in a more natural way such that the signed-distance property of the level set function is incorporated in the grid adaptation.

The Euler-Lagrange equation for the above minimization problem can be derived

$$\begin{aligned} \nabla_{\xi,\eta} \cdot (\omega \nabla_{\xi,\eta} x) &= 0 \\ \nabla_{\xi,\eta} \cdot (\omega \nabla_{\xi,\eta} y) &= 0\,. \end{aligned} \tag{9}$$

Notice that the equations above are elliptic and uncoupled. Therefore, they can be solved easily using standard iterative schemes like SOR or PCG.

3.2 Coordinate Transformation

The next step is to solve the Laplace equation in this new set of grid points (x^{n+1}, y^{n+1}). This can be done by using the following coordinate transformation

$$\begin{aligned} dx &= A d\xi + B d\eta \\ dy &= M d\xi + N d\eta\,, \end{aligned} \tag{10}$$

where

$$A = \frac{\partial x}{\partial \xi}\,,\; B = \frac{\partial x}{\partial \eta}\,,\; M = \frac{\partial y}{\partial \xi} \text{ and } N = \frac{\partial y}{\partial \eta} \tag{11}$$

are determined from the local grid information. Using this transformation and defining Δ to be the Jacobian of this transformation, we can rewrite the gradient operator in the Cartesian coordinates as

$$\nabla_{x,y} = \frac{1}{\Delta}\left(-M\frac{\partial}{\partial \eta} + N\frac{\partial}{\partial \xi}, -B\frac{\partial}{\partial \xi} + A\frac{\partial}{\partial \eta}\right)\,. \tag{12}$$

The Laplacian is then replaced by the Beltrami operator given by

$$\begin{aligned} \nabla^2_{x,y} T &= \frac{1}{\Delta}\left[\frac{\partial}{\partial \xi}\left(\frac{(B^2+N^2)T_\xi - (AB+MN)T_\eta}{\Delta}\right)\right. \\ &\quad \left. + \frac{\partial}{\partial \eta}\left(\frac{(A^2+M^2)T_\eta - (AB+MN)T_\xi}{\Delta}\right)\right]\,. \end{aligned} \tag{13}$$

Notice that our approximation of the Beltrami operator at $T(\xi_i, \eta_j)$ uses a linear combination of the 9 grid points centered at (ξ_i, η_j), i.e. $\{T_{i,j},\, T_{i-1,j-1},\, T_{i,j-1},\, T_{i+1,j-1},\, T_{i-1,j},\, T_{i+1,j},\, T_{i-1,j+1},\, T_{i,j+1},\, T_{i+1,j+1}\}$.

However, special care is needed when the grid point is located near the interface. Since a boundary condition is imposed on the interface, the Ghost Fluid Method is used

[17, 5]. For example, consider the one-dimensional case where we assume $\phi_{i-1}\phi_i > 0$ and $\phi_i\phi_{i+1} < 0$, i.e. the interface is located between the grid point at x_i and x_{i+1} but not between x_{i-1} and x_i. By the Ghost Fluid Method, the Laplacian at x_i can be approximated by

$$\nabla_h^2 T_i = \frac{T_{i-1} - 2T_i + T_{i+1}^*}{\Delta x^2}\,, \tag{14}$$

where T_{i+1}^* is the value extrapolated from $T(\phi^{-1}(0))$, T_i, and/or T_{i-1}, depending on the desired accuracy.

In the two dimensional case with the Cartesian coordinates, we only need to check the above condition on $\phi_{i,j}$ with the 4 neighboring grid points $(x_{i\pm1}, y_j)$ and $(x_i, y_{j\pm1})$. However, with the adaptive grids, a total of 8 points $(\xi_{i\pm1}, \eta_j)$, $(\xi_i, \eta_{j\pm1})$ and $(\xi_{i\pm1}, \eta_{j\pm1})$ need to be considered in general.

3.3 Temperature Extension

After the temperature is computed in both domain Ω_s and Ω_l, we need to compute the jump of its normal derivative across the interface. This can be done using the following extension algorithm [2, 4].

1. Define $T^1 = \chi_{\Omega_1} T$ and $T^2 = \chi_{\Omega_2} T$, where $\chi_{\{\}}$ is the characteristic function and is defined to be 1 if (x, y) is in the subscripted domain and is 0 otherwise.
2. Define $T_n^1 = \chi_{\Omega_1}(\vec{n} \cdot \nabla T)$, $T_n^2 = \chi_{\Omega_2}(\vec{n} \cdot \nabla T)$.
3. Define $T_{nn}^1 = \chi_{\Omega_1}(\vec{n} \cdot \nabla T_n)$, $T_{nn}^2 = \chi_{\Omega_2}(\vec{n} \cdot \nabla T_n)$.
4. Extend T_{nn}^1 and T_{nn}^2 to Ω_2 and Ω_1, respectively, by solving

$$\begin{aligned} (T_{nn}^1)_\tau + H(\phi)\vec{n} \cdot \nabla T_{nn}^1 &= 0 \\ (T_{nn}^2)_\tau - H(-\phi)\vec{n} \cdot \nabla T_{nn}^2 &= 0\,, \end{aligned} \tag{15}$$

   where $H(.)$ is the Heaviside function.

5. Extend T_n^1 and T_n^2 to Ω_2 and Ω_1, respectively, by solving

$$\begin{aligned} (T_n^1)_\tau + H(\phi)(\vec{n} \cdot \nabla T_n^1 - T_{nn}^1) &= 0 \\ (T_n^2)_\tau - H(-\phi)(\vec{n} \cdot \nabla T_n^2 - T_{nn}^2) &= 0\,. \end{aligned} \tag{16}$$

6. Compute $v_n = T_n^1 - T_n^2$.

In [2, 4], the above PDE's are solved in the τ-direction for few iteration steps. However, because of the restriction in the step size coming from the CFL condition, the resulting speed of information extension would be relatively slow. To resolve this problem, we determine the steady state solution of the above PDE's directly by considering the time-independent extension equations. Equations (15) and (16) are replaced by

$$\begin{aligned} H(\phi)\vec{n} \cdot \nabla T_{nn}^1 &= 0 \\ -H(-\phi)\vec{n} \cdot \nabla T_{nn}^2 &= 0\,, \end{aligned} \tag{17}$$

and

$$\begin{aligned} H(\phi)(\vec{n}\cdot\nabla T_n^1 - T_{nn}^1) &= 0 \\ -H(-\phi)(\vec{n}\cdot\nabla T_n^2 - T_{nn}^2) &= 0\,, \end{aligned} \tag{18}$$

respectively. Then, these PDE's are solved using the idea of the fast sweeping method [21, 8, 9]. Note that all equations are in the form of

$$uT_\xi + vT_\eta = R(\xi,\eta)\,. \tag{19}$$

With upwind differencing, we get the following iterative scheme for $T_{i,j}$ with $1 \le i \le n_\xi$ and $1 \le j \le n_\eta$,

$$T_{i,j}^{k+1} = \frac{\left[\begin{array}{l} \Delta\eta\{(u-|u|)T_{i+1,j}^{k'} - (u+|u|)T_{i-1,j}^{k'}\} + \\ \Delta\xi\{(v-|v|)T_{i,j+1}^{k'} - (v+|v|)T_{i,j-1}^{k'}\} - 2\Delta\xi\Delta\eta R_{i,j} \end{array}\right]}{-2(\Delta\eta|u| + \Delta\xi|v|)}\,, \tag{20}$$

for $k = 1, 2, \cdots$ and $k' = k$ or $k+1$, depending on the direction of the sweeping. Computationally, the determination of k' is automatic and this sweeping procedure can be done, for example, by the following piece of code in FORTRAN,

```
DO I=1,NX
DO J=1,NY
   T(I,J)=UPDATE_T(T,R,I,J)
ENDDO
ENDDO

DO I=NX,-1,1
DO J=1,NY
   T(I,J)=UPDATE_T(T,R,I,J)
ENDDO
ENDDO

DO I=1,NX
DO J=NY,-1,1
   T(I,J)=UPDATE_T(T,R,I,J)
ENDDO
ENDDO

DO I=NX,-1,1
DO J=NY,-1,1
   T(I,J)=UPDATE_T(T,R,I,J)
ENDDO
ENDDO

DO J=1,NY
DO I=1,NX
   T(I,J)=UPDATE_T(T,R,I,J)
ENDDO
ENDDO

DO J=NY,-1,1
DO I=1,NX
   T(I,J)=UPDATE_T(T,R,I,J)
ENDDO
ENDDO

DO J=1,NY
DO I=NX,-1,1
   T(I,J)=UPDATE_T(T,R,I,J)
ENDDO
ENDDO

DO J=NY,-1,1
DO I=NX,-1,1
   T(I,J)=UPDATE_T(T,R,I,J)
ENDDO
ENDDO
```

with the function `UPDATE_T` given by the expression (20).

Although the normal velocity v_n obtained from the above algorithm is defined at all grid points, a normal direction extension algorithm is still implemented to reduce the numerical error [4, 15]. At each point (x_i, y_j),

1. determine the closest point on the interface to that point, given by

$$(x^*, y^*) = (x_i, y_j) - \phi(x_i, y_j)\vec{n}\,, \tag{21}$$

   where $\vec{n}$ is the normal vector at the point (x_i, y_j).

2. use cubic interpolation to determine $v_n(x^*, y^*)$ and assign

$$v_n(x_i, y_j) = v_n(x^*, y^*)\,. \tag{22}$$

3.4 Updating the Level Set Function

The level set equation in the transformed coordinates is given by

$$\phi_t + \frac{v_n}{\Delta}\sqrt{a\phi_\xi^2 + 2b\phi_\xi\phi_\eta + c\phi_\eta^2} = 0\,, \tag{23}$$

where $a = B^2 + N^2$, $b = -(AB + MN)$, $c = A^2 + M^2$ and $\Delta = AN - BM$. This Hamilton-Jacobi equation is solved using a fifth order WENO-LF scheme in space and a third order TVD-RK scheme in time.

3.5 Reinitialization

To reduce the error in determining the zero level set by interpolation, it is desirable to preserve the signed-distance property of the level set function. This can be done by solving

$$\phi_\tau + S(\phi_0)(|\nabla\phi| - 1) = 0\,, \tag{24}$$

with $S(.)$ approximating the signum function. In the transformed coordinates, the above equation reads

$$\phi_\tau + S(\phi_0)\left(\frac{1}{\Delta}\sqrt{a\phi_\xi^2 + 2b\phi_\xi\phi_\eta + c\phi_\eta^2} - 1\right) = 0\,. \tag{25}$$

Again, this Hamilton-Jacobi equation is solved using a fifth order WENO-LF scheme in space and a third order TVD-RK scheme in time. Few iterations in the τ-direction are sufficient to get a regular level set function near the zero level set.

3.6 A Predictor-Corrector Approach

Traditionally, the moving grid technique updates the objective function and the location of the grids in the following way:

1. Given the grids (x^n, y^n) and the function $\phi^n(x^n, y^n)$ defined at t^n, update ϕ^n and get $\phi^{n+1/2}(x^n, y^n)$. This step is denoted by the operator E_{x^n}.
2. With the function $\phi^{n+1/2}(x^n, y^n)$ fixed, adapt the grid to obtain (x^{n+1}, y^{n+1}). The output of this step is denoted by $A(\phi^{n+1/2})$.
3. Interpolate and determine $\phi^{n+1}(x^{n+1}, y^{n+1})$ at (x^{n+1}, y^{n+1}). The interpolation procedure is denoted by $I_{x^{n+1}}$.

The problem with the above procedure is the excessive numerical dissipation in the operator $I_{x^{n+1}}$. To reduce this error, different types of interpolation scheme were introduced [19].

In this work, a predictor-corrector approach will be used.

1. The procedures E_{x^n} and $A(\phi^{n+1/2})$ are considered as a predictor step. The function $\phi^{n+1/2}(x^n, y^n)$ is considered as a prediction of the function values at the next time step t^{n+1}, and it is **only** used to give the new location of the grids .
2. A new operator $E_{x^n \to x^{n+1}}$ is introduced. This operator couples the level set equation with the grids motion driven by the velocities

$$F = \frac{dx}{dt} \text{ and } G = \frac{dy}{dt}\,, \tag{26}$$

in terms of the following transformation

$$\begin{aligned} dt &= d\tau \\ dx &= F d\tau + A d\xi + B d\eta \\ dy &= G d\tau + M d\xi + N d\eta\,, \end{aligned} \tag{27}$$

where the local grid information is given by

$$A = \frac{\partial x}{\partial \xi}\,,\ B = \frac{\partial x}{\partial \eta}\,,\ M = \frac{\partial y}{\partial \xi}, N = \frac{\partial y}{\partial \eta} \text{ and } \Delta = AN - BM\,. \tag{28}$$

Now the Level Set equation in this moving grid coordinates is given by

$$\phi_\tau - \frac{FN - GB}{\Delta}\frac{\partial \phi}{\partial \xi} - \frac{GA - FM}{\Delta}\frac{\partial \phi}{\partial \eta} + \frac{v_n}{\Delta}\sqrt{a\phi_\xi^2 + 2b\phi_\xi\phi_\eta + c\phi_\eta^2} = 0\,, \tag{29}$$

4 Numerical Examples

4.1 Example 1

This example is used to test the convergence of different quantities, including the temperature, the normal velocity and the location of the interface at time $t = 0.1$. There are two disjoint interfaces fixed in the domain, i.e. two level set functions ϕ and ψ will be used to define the interfaces at $t = 0$.

$$\begin{aligned} \phi &= r - 0.3 \\ \psi &= r - 0.8\,. \end{aligned} \tag{30}$$

The temperatures on the interfaces are fixed to be

$$\begin{aligned} T &= -0.0005/0.3 \text{ when } \phi = 0 \\ T &= -0.5 \text{ when } \psi = 0 \end{aligned} \tag{31}$$

initially. Figure 1 shows the contour lines of the temperature field computed using 50-by-50 and 100-by-100 rectangular grids. Solution, with the computational grids in the physical space using the proposed adaptive grids, are plotted in Figure 2 and 3.

Several quantities will be measured. The first one is the temperature on the interface $\phi = 0$. Cubic interpolation will be used to interpolate the computed solutions on $r = 0.3$ at each degree. L_1 and L_∞ errors are then computed. Because the exact temperature can be computed easily, we can also compute the L_1-error of the numerical solution using

$$\int |T - T_{\text{exact}}| dx dy = \int |T - T_{\text{exact}}| \Delta d\xi d\eta \,, \tag{32}$$

where Δ is the Jacobian of the transformation in the adaptive grid mapping. As seen on the first row in Figure 6, by adaptive grids toward $\phi = 0$, we can lower the error near the interface by approximating a factor of 3. The x-axis represents $\text{Log}(\min \Delta)$, with Δ is the grid area, and the y-axis is the logarithm of the corresponding errors.

The normal velocity of the interface at $t = 0$ can also be computed and is given by

$$v_n|_{\phi=0} = \frac{299}{180 \log(8/3)} \,. \tag{33}$$

The L_1 and L_∞ of the errors on the interface $\phi = 0$ using different number of grid points are plotted on the third row in Figure 4.

The last row in Figure 4 shows the errors in the location of the crystal interface at $t = 0.1$. The shape of the crystal is always a circle with expanding radius. The L_1 error is measured by

$$\int_0^{2\pi} |\phi(r = r_{\text{exact}}, \theta)| d\theta \,, \tag{34}$$

with $\phi(r = r_{\text{exact}})$ is computed using cubic interpolation.

Least square fitting is then used to approximate the slope of the graphs. Because $\Delta = O(h^2)$ where h is the length of the grid cell, we multiply the slope of the graphs by 2 and the rates are given in Table 1. In the cases of adaptive grids, we notice that the grid distortion is so serious that it affects the accuracy when the number of grids is small. We also show the convergence rates by ignoring the data using 50-by-50 grids in Table 1.

4.2 Example 2

The initial condition is a square of length 0.2. The computational domain $\Omega = [-1, 1]^2$. Temperature on the interface is given by $T = -0.0005\kappa$ and that on the outer boundary $\{x = \pm 1\} \cup \{y = \pm 1\}$ is given as -1. The interfaces from $t = 0$ to 0.2 in increment of 0.025 are plotted in Figures 5-7. Grid anisotropy can be eliminated by increasing

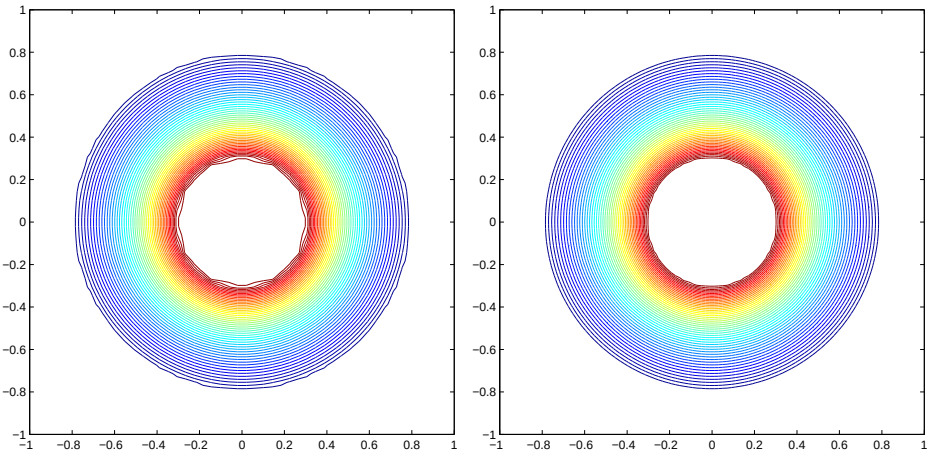

Figure 1: Temperature contours using 50-by-50 and 100-by-100 uniform rectangular grids.

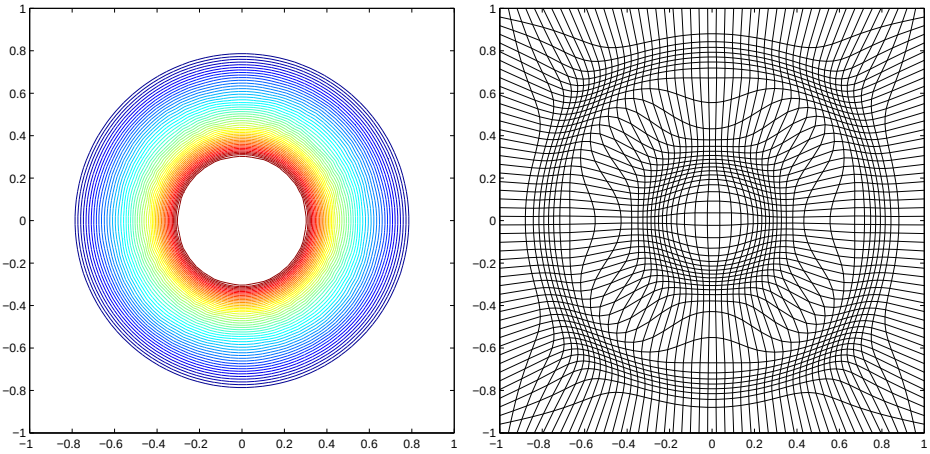

Figure 2: Temperature contours computed by 50-by-50 grids adapted to both the zero level sets $\phi = 0$ and $\psi = 0$.

	Uniform		(a)		(b)	
	L_1	L_∞	L_1	L_∞	L_1	L_∞
$T\|_{\phi=0}$ at $t = 0$	1.05	1.00	1.07	1.06	-	-
$T\|_\Omega$ at $t = 0$	2.02	1.88	0.87	0.69	1.19	0.98
$v_n\|_{\phi=0}$ at $t = 0$	1.44	1.39	0.59	0.80	1.22	1.20
$\phi = 0$ at $t = 0.1$	1.39	1.32	0.52	0.92	1.21	1.40

Table 1: Convergence rates for difference quantities using both uniform and adaptive grids. (a) and (b) are obtained using grids adapted to both $\phi = 0$ and $\psi = 0$. Data from 50-by-50 grid points are dropped on column (b).

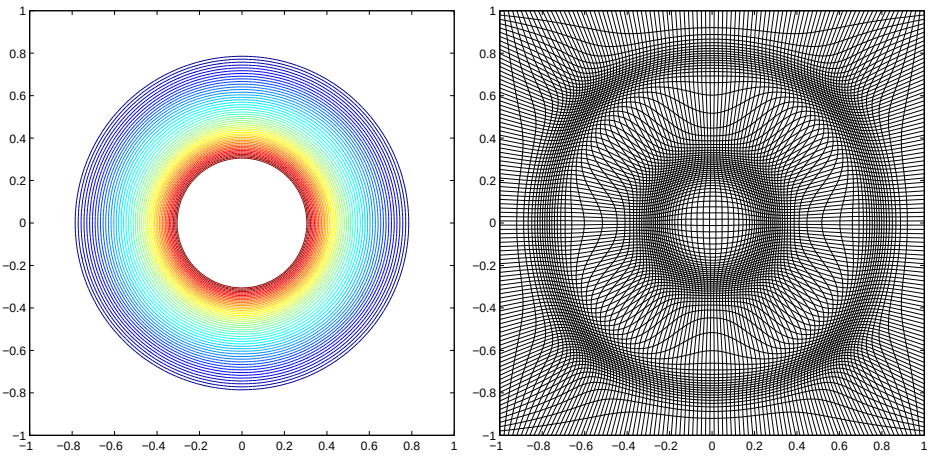

Figure 3: Temperature contours computed by 100-by-100 grids adapted to both the zero level sets $\phi = 0$ and $\psi = 0$.

the total number of grids. Comparing the solutions in the first row of Figure 5 with Figure 6, only around one-forth of grid points are needed in the adaptive grid to get similar results from uniform rectangular grids.

4.3 Example 3

This example simulates crystal anisotropy in the sense that the crystal grows in some preferred directions. The Gibbs-Thomson relation on the interface is given by [1]

$$T = -\sigma(\vec{n})\kappa - \mu(\vec{n})v_n\,, \tag{35}$$

with

$$\begin{aligned} \sigma(\theta) &= \sigma\left\{1 + A_s\left[\frac{8}{3}\sin^4\left(\frac{1}{2}m_s(\theta-\phi_s)\right) - 1\right]\right\} \\ \mu(\theta) &= \mu\left\{1 + A_k\left[\frac{8}{3}\sin^4\left(\frac{1}{2}m_l(\theta-\phi_k)\right) - 1\right]\right\}\,. \end{aligned} \tag{36}$$

Here m_s and m_k determine the mode of symmetry of the crystal, ϕ_s and ϕ_k determine the angles of the symmetry axis made with the x-axis, A_s and A_k are the strength of the anisotropy and θ is the angle made between the normal vector $\vec{n}$ and the x-axis.

Three different initial conditions are imposed:

(a) Square with sides 1/6.

(b) Square with sides 1/6 and rotated by $\pi/4$.

(c) Four-fold initial condition given by

$$r = r_0 + p_0\cos 4\theta\,, \tag{37}$$

with $r_0 = 0.05$ and $p_0 = 0.01$.

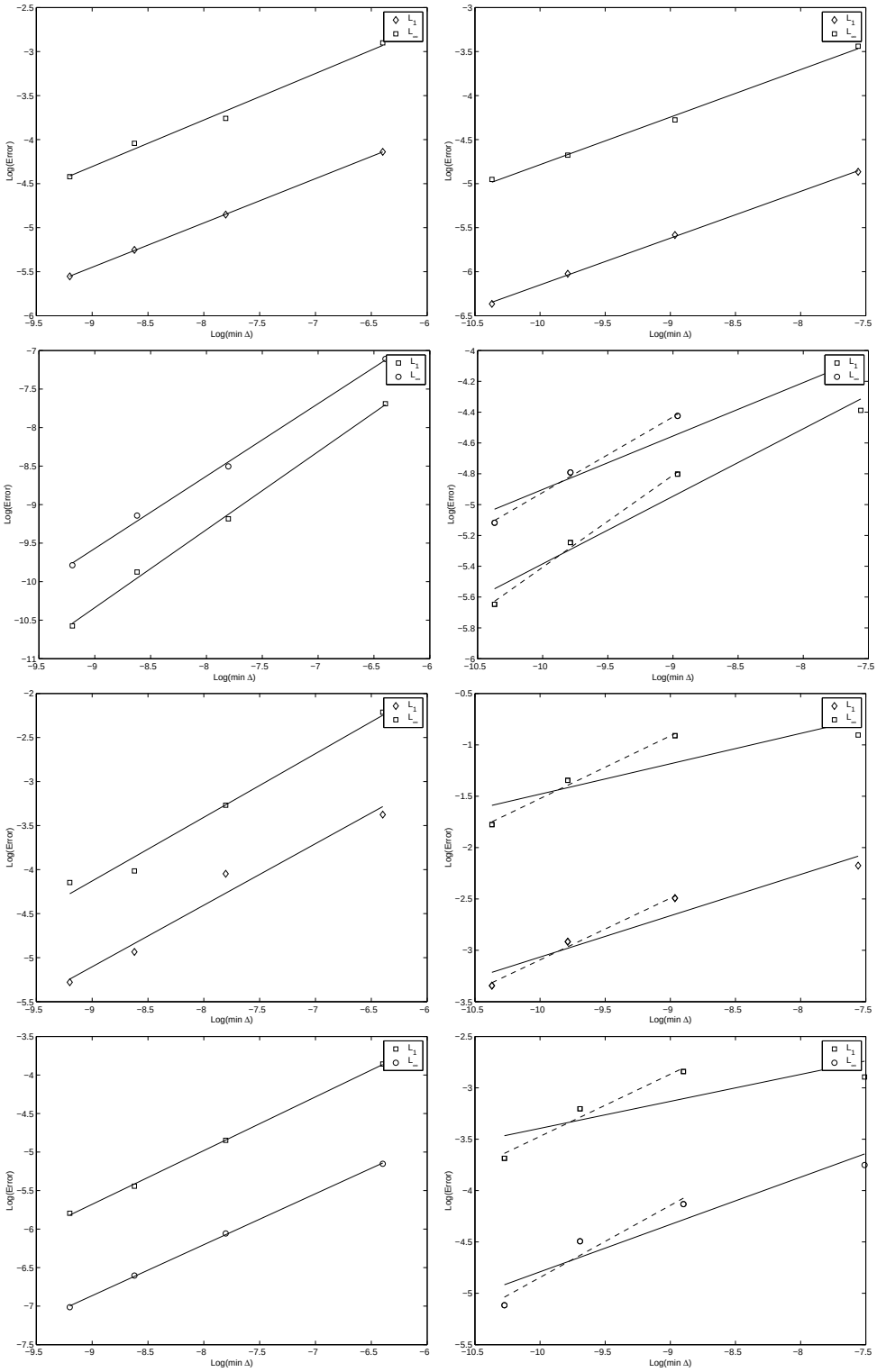

Figure 4: Convergence tests. The first and second column correspond to the results using uniform rectangular grids and grids adapted to both $\phi = 0$ and $\psi = 0$, respectively. The first row shows the convergence of temperatures on $\phi = 0$. Convergence of temperatures in the whole domain are shown on the second row. Third row shows the convergence of the normal velocity on $\phi = 0$ at $t = 0$. The convergence of the location of $\phi = 0$ at $t = 0.1$ using different type of grids are shown on the last row.

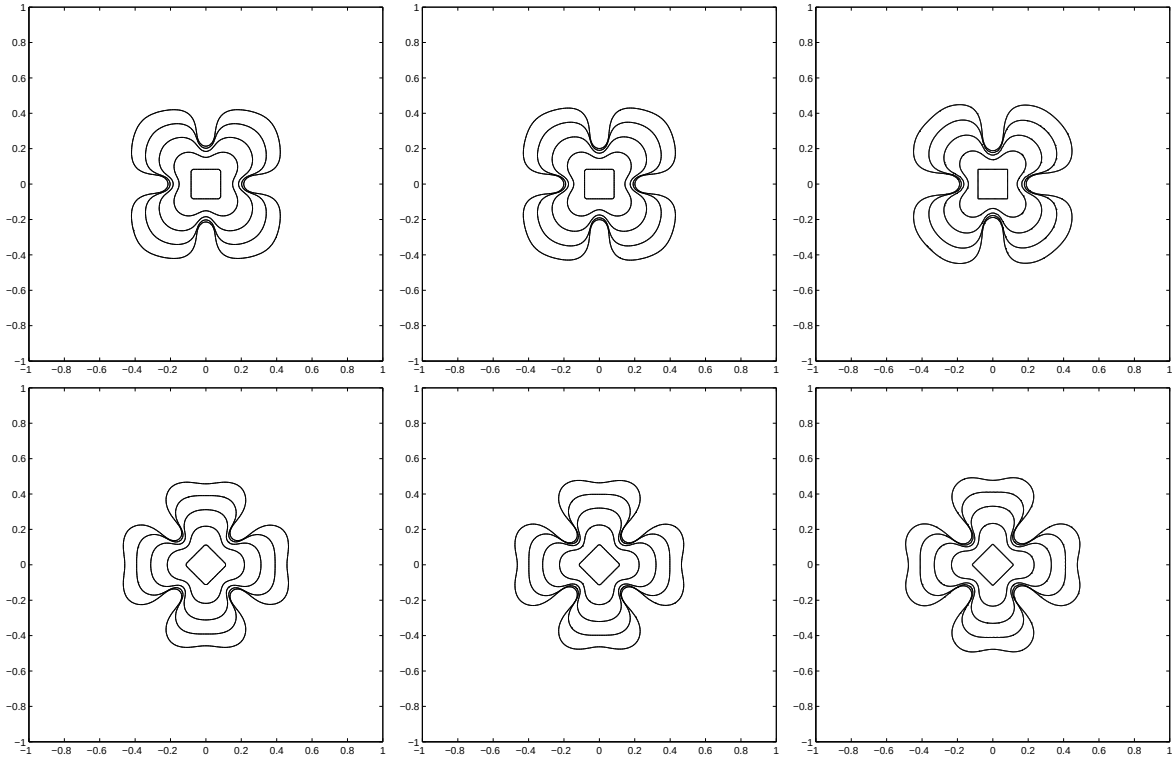

Figure 5: (Example 2) Evolution of the interface at increments of 0.025 from $t = 0$ to 0.2 using 150-by-150, 200-by-200 and 400-by-400 uniform rectangular grids with two different orientations of the initial profile.

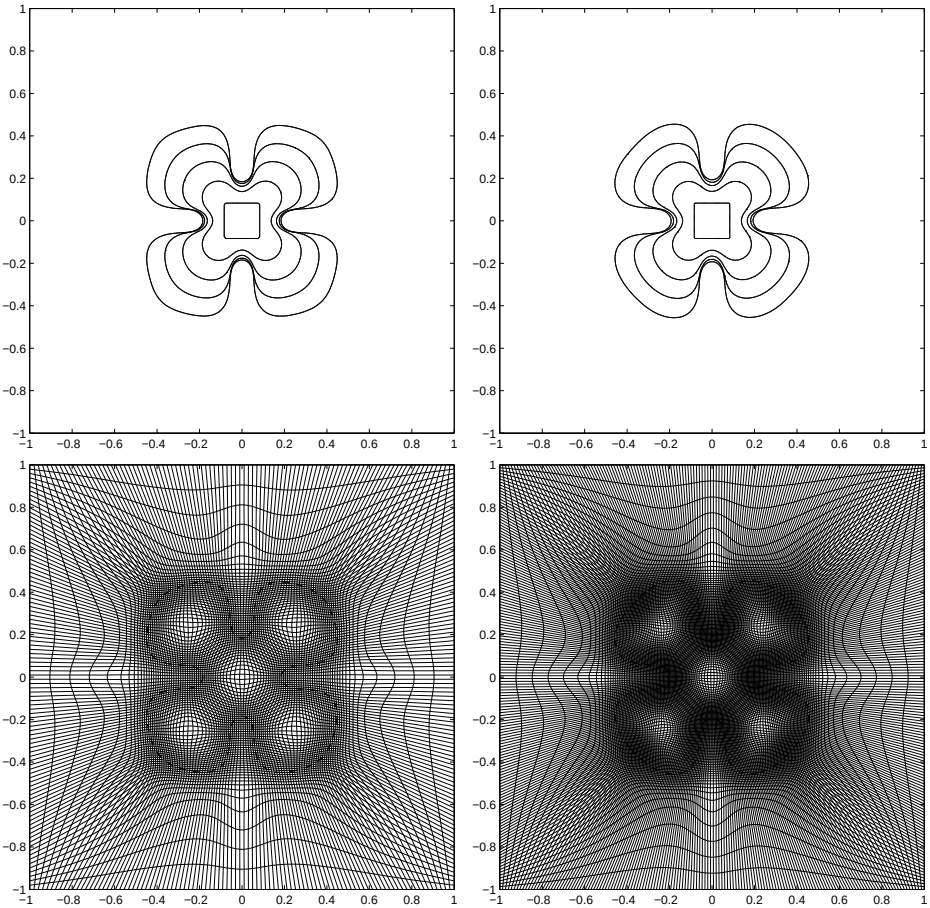

Figure 6: (Example 2) Evolution of the interface at increments of 0.025 from $t = 0$ to 0.2 using 100-by-100 and 150-by-150 adaptive grids. The second row shows the grid points at the last time step.

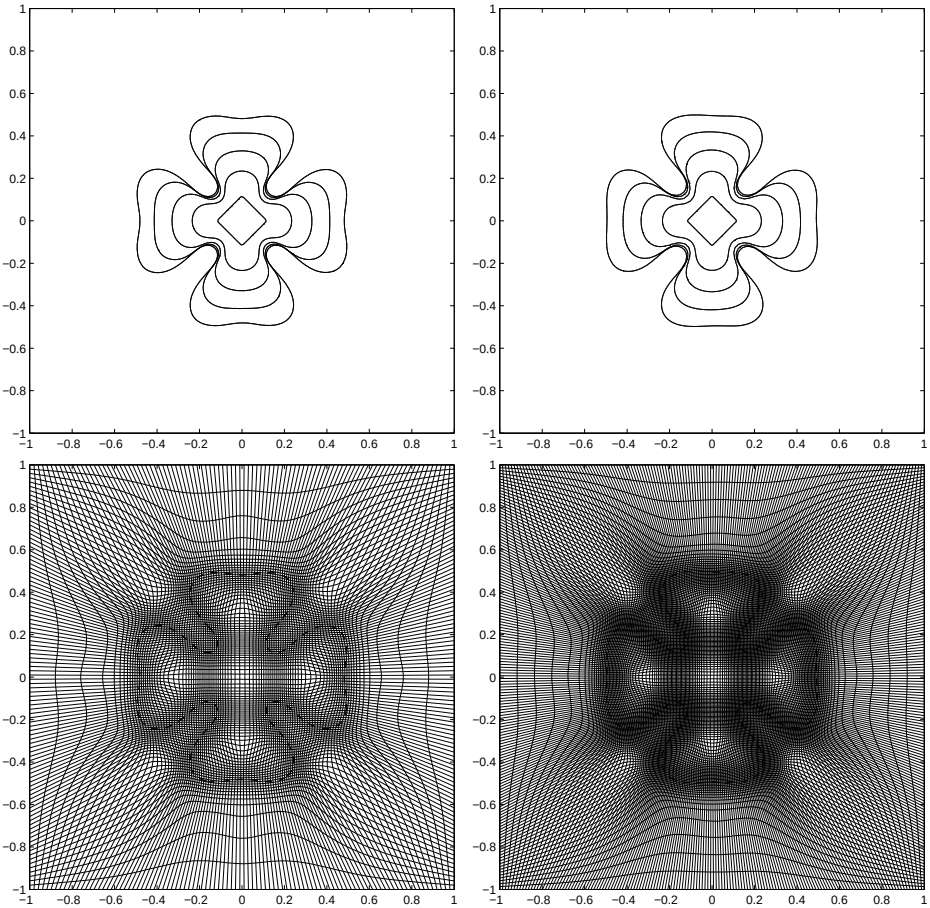

Figure 7: (Example 2) Evolution of the interface at increments of 0.025 from $t = 0$ to 0.2 using 100-by-100 and 150-by-150 adaptive grids with another orientation for the initial seed. The second row shows the grid points at the last time step.

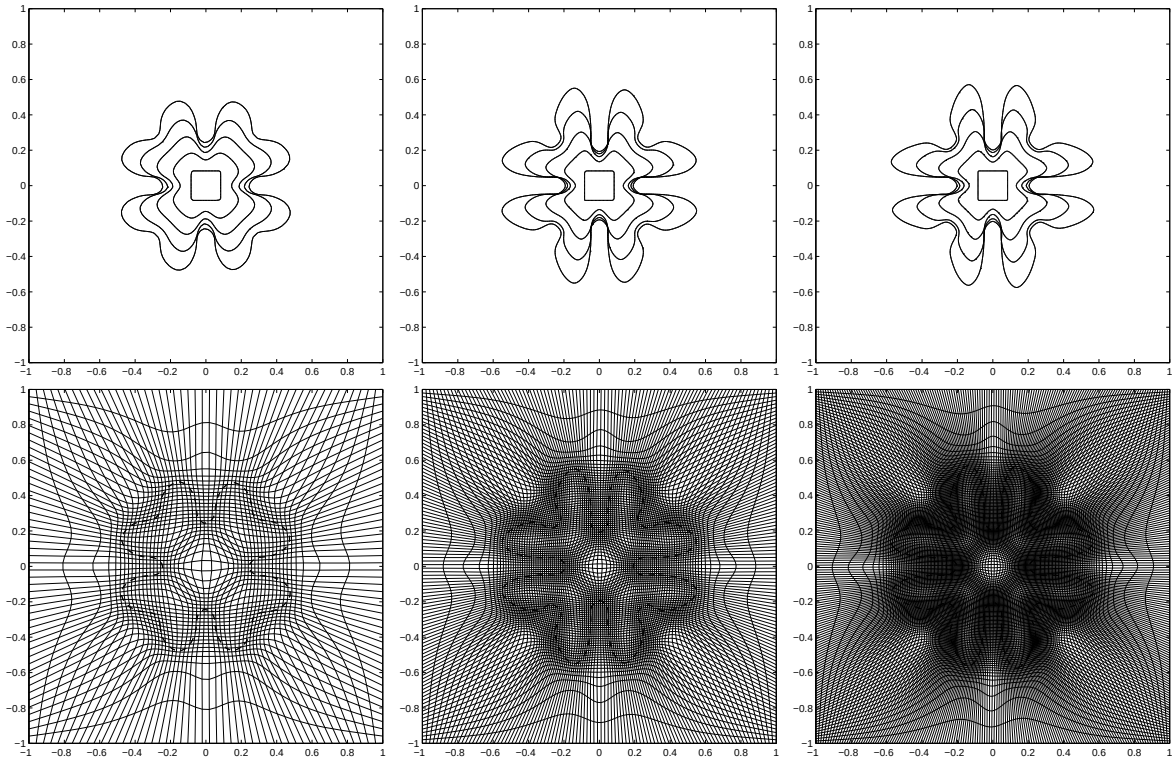

Figure 8: (Example 3a) 50-by-50, 100-by-100 and 150-by-150 adaptive grids, $A_s = A_k = 0.5$, $\sigma = \mu = 0.0005$ and $m_s = m_k = 4$. Second row shows the adaptive grids at the last time step.

All solutions are obtained at increments of 0.025 from $t = 0$ to 0.2. The outer boundary condition is imposed on $r = 0.95$ given by $T = -1$.

Figure 8 shows the solutions using adaptive moving grids with initial condition (a). The mode of symmetry in the Gibbs-Thomson relation is 4. Comparing the solutions, the resolution of the solutions using 100-by-100 adaptive grids is similar to that of 200-by-200 uniform adaptive grids. The numerical dissipation is reduced by having grid points concentrated around the zero level set of ϕ, comparing with Figure 9.

Solutions with the initial configuration (b) are plotted in Figures 10-11. Similar to case (a), we get similar resolution of solutions using only around one-forth of total grid points in the adaptive method than that from the uniform rectangular grids.

Figure 12-13 show the solutions with the initial shape given by (c). The surface tension coefficient and the kinetic coefficient are half of the cases (a) and (b). The growth is now becoming more sensitive to the initial profile and the resulting shape of the crystal will be more complex. The mode of symmetry of the crystal is increased to 6. Again, we obtain similar shape of the crystal using only one-forth of grid points in the adaptive method than that from the uniform grids.

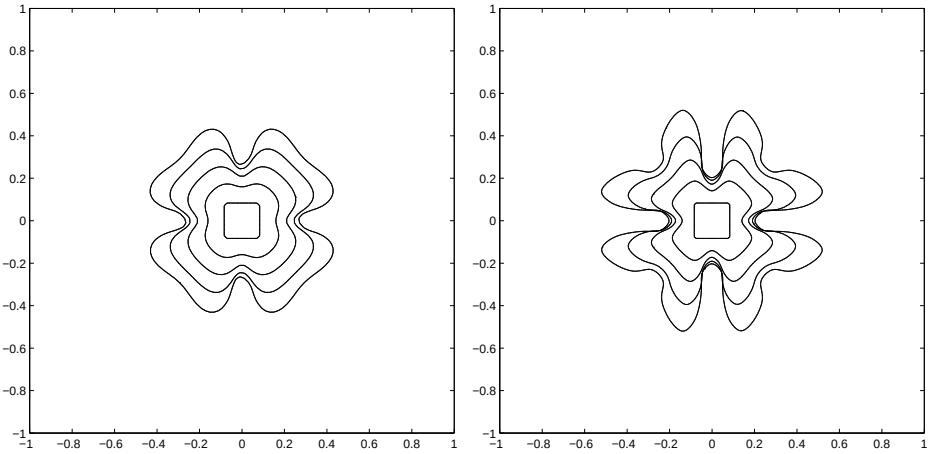

Figure 9: (Example 3a) 100-by-100 and 200-by-200 uniform rectangular grids, $A_s = A_k = 0.5$, $\sigma = \mu = 0.0005$ and $m_s = m_k = 4$.

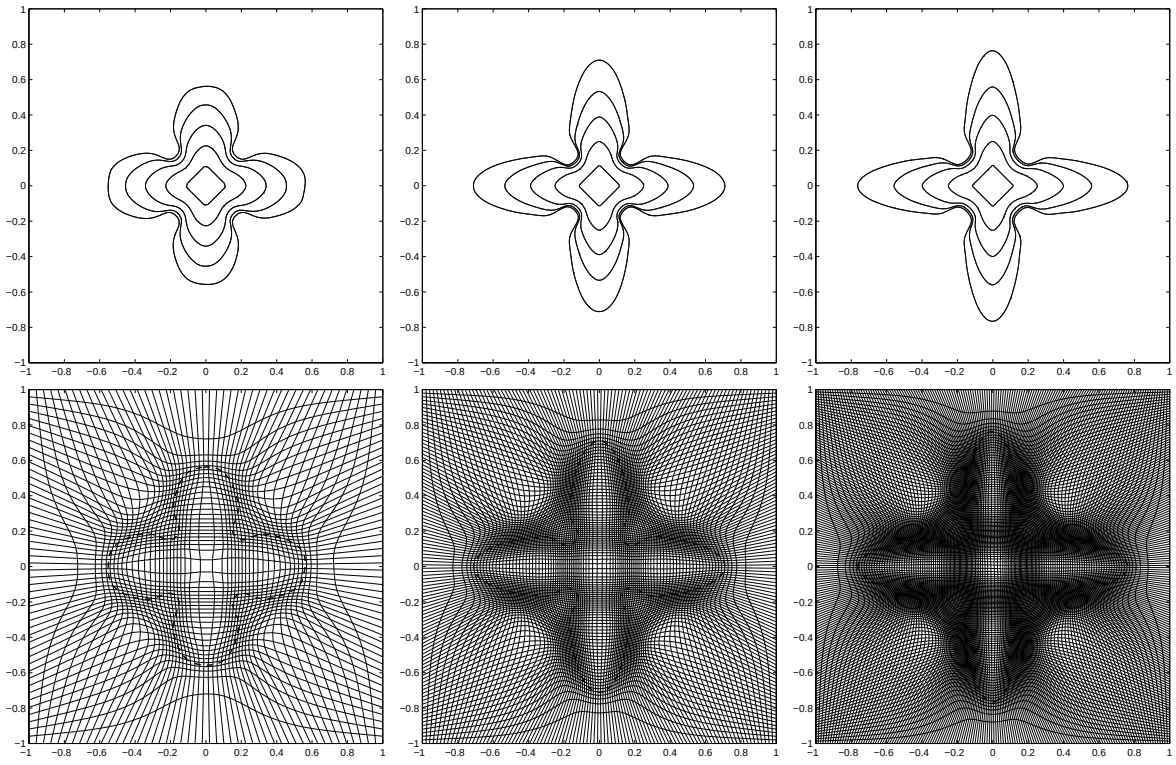

Figure 10: (Example 3b) 50-by-50, 100-by-100 and 150-by-150 adaptive grids, $A_s = A_k = 0.5$, $\sigma = \mu = 0.0005$ and $m_s = m_k = 4$. Second row shows the adaptive grids at the last time step.

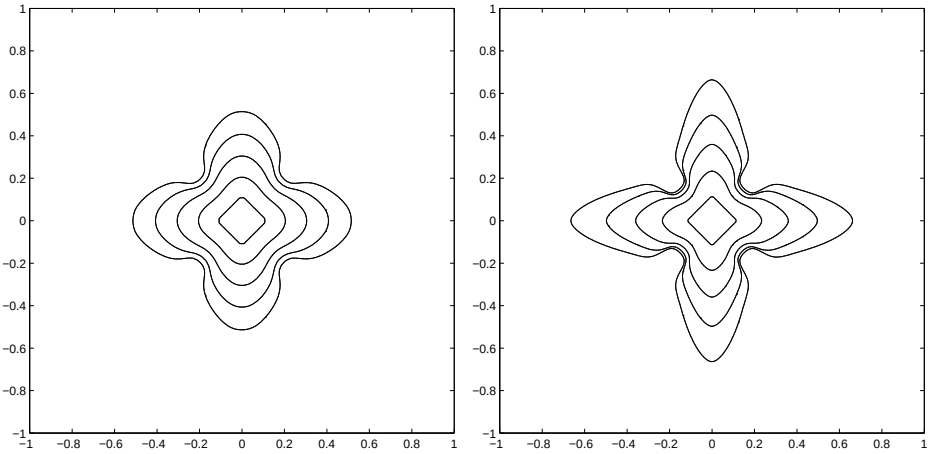

Figure 11: (Example 3b) 100-by-100 and 200-by-200 uniform rectangular grids, $A_s = A_k = 0.5$, $\sigma = \mu = 0.0005$ and $m_s = m_k = 4$.

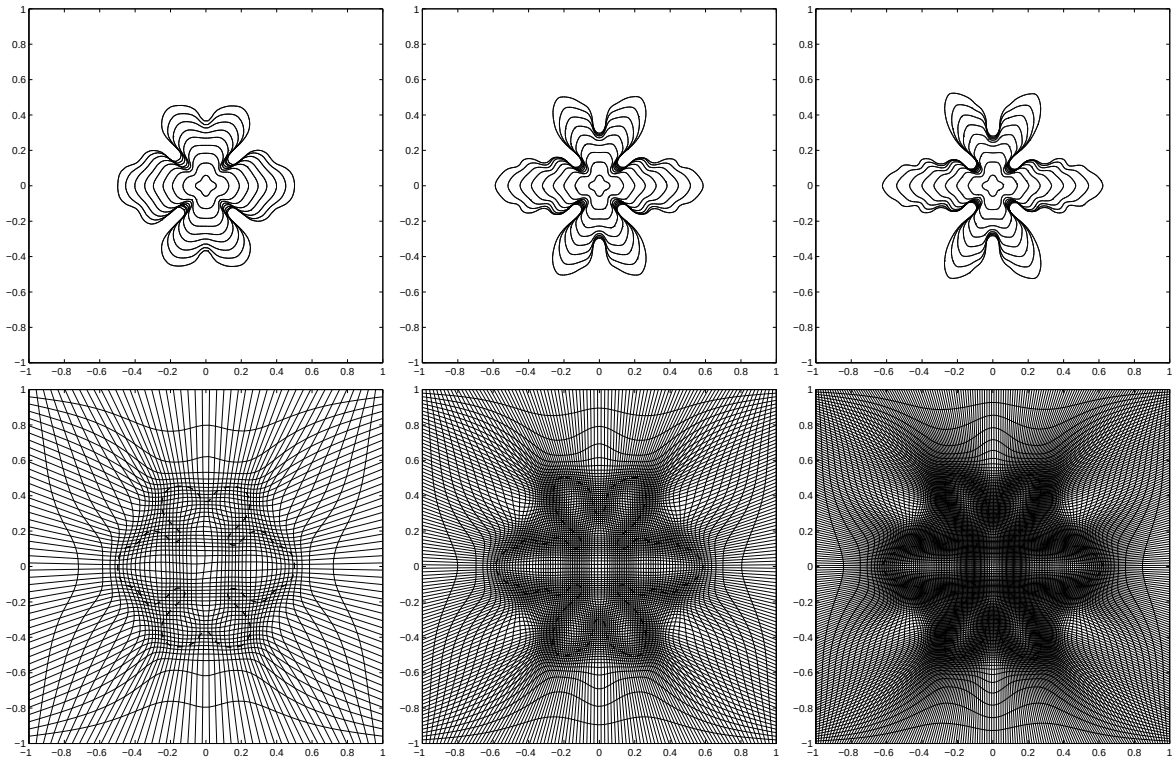

Figure 12: (Example 3c) 50-by-50, 100-by-100 and 150-by-150 adaptive grids, $A_s = A_k = 0.5$, $\sigma = \mu = 0.00025$ and $m_s = m_k = 6$. Second row shows the adaptive grids at the last time step.

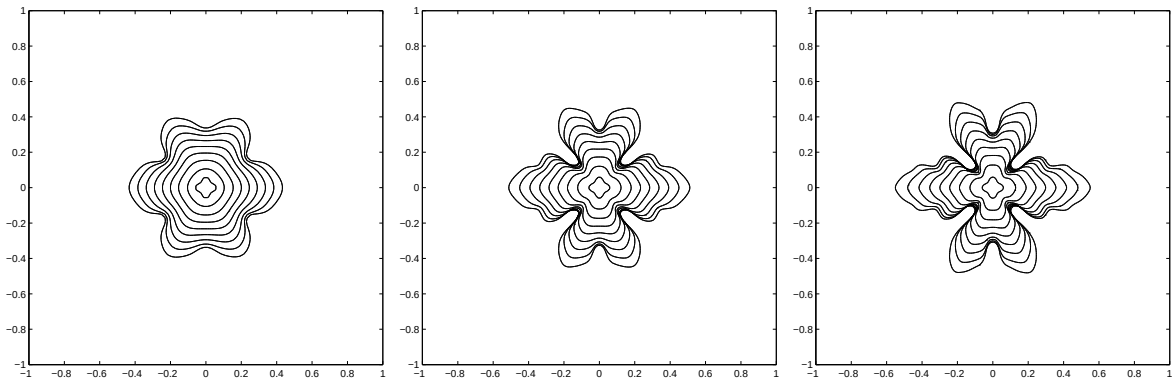

Figure 13: (Example 3c) 100-by-100, 150-by-150 and 200-by-200 uniform rectangular grids, $A_s = A_k = 0.5$, $\sigma = \mu = 0.00025$ and $m_s = m_k = 6$.

4.4 Example 4

The last example demonstrates the situation where there are more than one initial seed. Two four-folded seeds, with shapes same as in (37), are dropped into the super-cooled bath with centers at

(a) $(\pm 0.10, \pm 0.10)$,

(b) $(\pm 0.05, \pm 0.05)$.

The Gibbs-Thomson relation on the interfaces are given by $T = -0.0005\kappa$. Temperature on the outer boundary $r = 0.95$ is defined as -1.0. Solutions at $t = 0$ to 0.1 using 150-by-150 grid points are shown in Figure 14-15.

5 Conclusion

An adaptive Level Set Method has been implemented to solve the quasi-steady Stefan problem. Numerical examples show that this adaptive algorithm can give high resolution solution using few number of computational grids. Comparing with the solutions using uniform rectangular grids, approximately only one-forth of grid points are needed in this adaptive method to give similar resolutions of the solutions. This results in an efficient numerical method to deal with problems containing a complex interface.

Of course, the adaptive approach presented in this paper can also be applied to other applications of Level Set Method.

6 Acknowledgment

S. Leung was supported by (...).

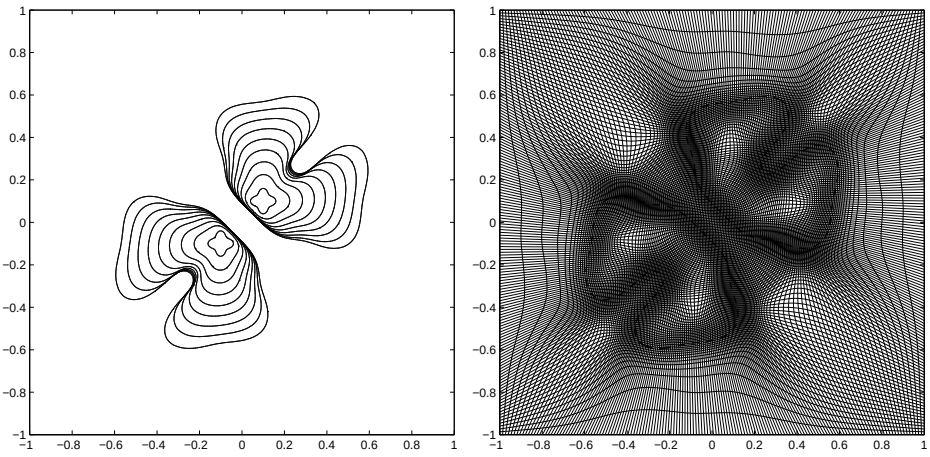

Figure 14: (Example 4a) Two four-fold initial seeds are put at $(\pm 0.10, \pm 0.10)$. Interfaces at $t = 0$ to 0.1 at increment of 0.0125 are shown on the left hand side. The grid points at the last time step on the right hand side.

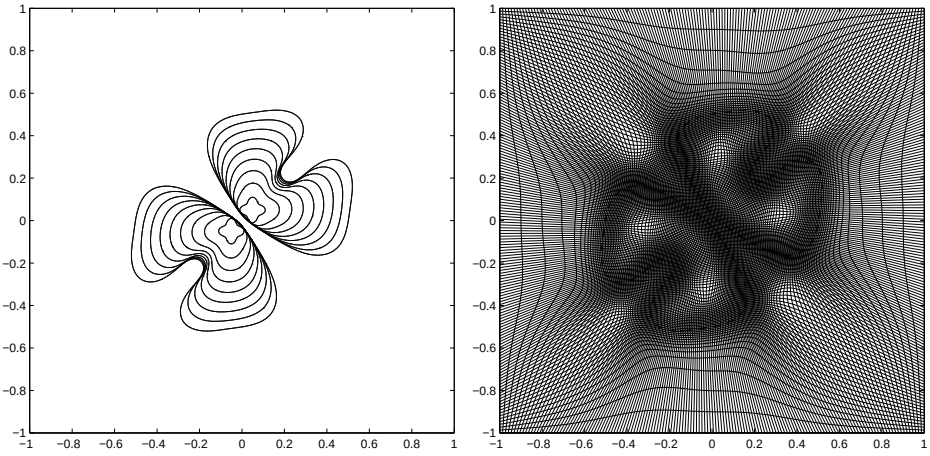

Figure 15: (Example 4b) Two four-fold initial seeds are put at $(\pm 0.05, \pm 0.05)$. Interfaces at $t = 0$ to 0.1 at increment of 0.0125 are shown on the left hand side. The grid points at the last time step are plotted on the right hand side.

References

[1] R. Almgren. Variational algorithms and pattern formation in dendritic solidification. *J. Comput. Phys.*, 106:337–354, 1993.

[2] T. Aslam. A partial differential equation approach to multidimensional extrapolation. *J. Comput. Phys.*, 193:349–355.

[3] S. Chen, B. Merriman, S. Osher, and P. Smereka. A simple level set method for solving stefan problems. *J. Comput. Phys.*, 135, 1997.

[4] F. Gibou and R. Fedkiw. A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem. In preparation, 2003.

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

[6] W. Huang and R.D. Russell. Moving mesh stragegy based on a gradient flow equation for two-dimensonal problems. *SIAM J. Sci. Comput.*, 20(3):998–1015, 1999.

[7] D. Juric and G. Tryggvason. A front-tracking method for dendritic solidification. *J. Comput. Phys.*, 123:127–148, 1996.

[8] C. Y. Kao, S. J. Osher, and Y.-H. Tsai. Fast sweeping method for static Hamilton-Jacobi equations. Preprint, 2002.

[9] C.Y. Kao, S.J. Osher, and J. Qian. Lax-Friedrichs sweeping schemes for static Hamilton-Jacobi equations. *J. Comp. Phys.*, page in press, 2003.

[10] A. Karma and W. Rappel. Numerical simulation of three-dimensional dendritic growth. *Physical Review Letters*, 77:4050–4053, 1996.

[11] A. Karma and W. Rappel. Phase-field method for computationally efficient modelling of solidification with arbitrary interface kinetics. *Physical Review E*, 53:R3017–R3020, 1996.

[12] A. Karma and W. Rappel. Quantitative phase-field modeling of dendritic growth in two and three dimensions. *Physical Review E*, 57:4323–4349, 1998.

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

[14] P. Kumpp and S. Steinberg. *Fundamentals of Grid Generation.* CRC Press, 1993.

[15] P. Macklin and J. Lowengrub. A second order accurate algorithm for evolving interfaces via gradients of geometry-dependent interior poisson problems: Application to tumor growth. In preparation, 2004.

[16] B. Merriman, J. K. Bence, and S. J. Osher. Motion of multiple junctions: a level set approach. *J. Comput. Phys.*, 112:334–363, 1994.

[17] S. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer-Verlag, New York, 2003.

[18] J. Sethian and J. Strain. Crystal growth and dendritic solidification. *J. Comput. Phys.*, 98:231–253, 1992.

[19] H.K. Tang, T. Tang, and Zhang P. An adaptive mesh redistribution method for nonlinear hamilton-jacobi equations in two- and three-dimensions. *J. Comput. Phys.*, 188, 2003.

[20] J.F. Thompson, B.K. Soni, and N.P. Weatherill. *Handbook of Grid Generation.* CRC Press, 1999.

[21] Y.-H. Tsai, L.-T. Cheng, S. J. Osher, and H. K. Zhao. Fast sweeping method for a class of Hamilton-Jacobi equations. Preprint, 2001.