A Level Set Method for Thin Film Epitaxial Growth *

Susan Chen †,
Barry Merriman ‡,
Myungjoo Kang §,
Russel E. Caflisch ¶,
Christian Ratsch ‖,
Li-Tien Cheng **,
Mark Gyure, ††,
Ronald P. Fedkiw ‡‡,
Stanley Osher *

November 3, 2000

---

*Research supported in part by NSF and DARPA through grant NSF-DMS9615854 as part of the Virtual Integrated Prototyping (VIP) Initiative

†Mathematics Department, UCLA. Email: suchen@math.ucla.edu.
‡Mathematics Department, UCLA. Email: barry@math.ucla.edu.
§Mathematics Department, UCLA. Email: mkang@math.ucla.edu.
¶Mathematics Department, UCLA. Email: caflisch@math.ucla.edu.
‖HRL Labs and Mathematics Dept., UCLA. Email: cratsch@math.ucla.edu.
**Mathematics Department, UCLA. Email: lcheng@math.ucla.edu.
††HRL Laboratories. Email: gyure@hrl.com
‡‡Research supported in part by ONR N00014-97-1-0027. Mathematics Department, UCLA. Email: rfedkiw@math.ucla.edu.
*Mathematics Department, UCLA. Email: sjo@math.ucla.edu.

Abstract

We present a level set based numerical algorithm for simulating a model of epitaxial growth. The island dynamics model is a continuum model for the growth of thin films (see also [4, 12, 21, 22, 31]). In this paper, we emphasize the details of the numerical method used to simulate the island dynamics model.

1 Introduction

Modeling epitaxial growth presents an enormous challenge to theoretical physicists and materials scientists. The range of length and time scales represented by problems of practical interest (e.g., the growth of device layers) spans many orders of magnitude [13]. Atomistic processes can significantly affect quantities such as surface morphology at the largest length and time scales [3]. A model for epitaxial growth with great potential for use in engineering applications would describe lateral scales of several microns or more, be appropriate for a variety of homoepitaxial and heteroepitaxial systems, and be capable of describing different growth techniques.

Continuum equations of motion that take the form of partial differential equations [41] for the surface height profile do yield information on morphology at large length scales. As they are typically formulated [41, 42], however, continuum equations are appropriate only in a regime where the surface is already assumed to be macroscopically rough. These continuum methods are therefore unsuitable for describing atomic scale roughness which is a concern in many device applications. Kinetic Monte Carlo (KMC) simulations [9, 19, 23, 44] offer an alternative to continuum equations. They allow easy implementation of a wide range of atomistic kinetic processes, which can, in principle, be identified and their rates determined from first principles calculations [32]. However, simulations are usually based on the length and time scales of single atoms and adatom hopping rates, so modeling systems of practical interest is not always feasible.

Despite the practical limitations of the methods described above, they have been used with great effect to provide a comprehensive conceptual and computational framework for describing homoepitaxial growth [39], especially by molecular-beam epitaxy. More fundamental problems arise, however, when attempts are made to extend these techniques to heteroepitaxial systems, where the effects of lattice mismatch must be incorporated, or to other growth methods, such as vapor-phase epitaxy (VPE), which requires coupling the atomistic kinetics on the substrate to the hydrodynamic delivery of new material. Some aspects of these issues have been addressed for particular systems, but no general methodology has emerged to provide a unifying framework in the spirit of the work described above.

In this paper, a new model and closely related numerical technique are presented that address these issues. In this model, growth is described by the creation and subsequent motion of island boundaries; hence, this model is referred to as "island dynamics" [4, 12]. The model is discrete in the growth direction, but continuous in the lateral directions and therefore, in principle, can describe growth on arbitrarily large lateral length scales. Moreover, since the lateral directions are treated continuously, continuum equations representing any field variable can be coupled to the growth by solving the appropriate boundary value problem for the field and using local values of this field to determine the local velocity of the island boundaries. For example, the strain fields that occur in the presence of lattice mismatch or the hydrodynamic fields in a VPE reactor can be readily accommodated by this method.

Although island dynamics is a natural way of describing many aspects of epitaxial growth, its implementation requires tracking a large number of individual interfaces that coalesce or are created by nucleation. The development of the level set method for simulating the motion of free boundaries [7, 25], now makes numerical implementation of such a model practical. The island dynamics model and some preliminary results from it have been introduced in previous work [4, 12, 21, 31]; the emphasis in this work is on the numerical issues associated with using the level set method in the context of a model for irreversible growth.

This paper proposes a new numerical technique leading to an improved treatment of the internal $\rho = 0$ boundary condition over the smeared out delta function method proposed in [21] which uses a "slushy" interface formulation that does not achieve the $\rho = 0$ internal boundary condition unless the penalty term K is infinite which is, of course, not numerically possible. This "slushy" interface formulation was proposed (in [21]) as an alternative to the higher order accurate method in [7], since the method in [7] is much too slow (numerically) for the types of problems considered in this paper (and e.g. [4, 12, 21, 31]). Our new numerical technique gives an accurate representation of the $\rho = 0$ internal boundary condition (unlike [21]). Furthermore when this technique is implemented in conjunction with implicit time stepping, one obtains a symmetric matrix which is faster to invert than the nonsymmetric matrix produced in [7] making this new technique fast enough for the problems considered herein. We remark that our new formulation has been 0 implemented in previous work, see e.g. [22].

2 Island Dynamics and the Level Set Formalism

In this section, a brief introduction to the level set method is given, followed by a description of the island dynamics model for irreversible growth. In the island dynamics methodology, the physics is almost entirely contained in the specification of the normal velocity of island boundaries. The evolution of the boundaries can, in principle, be done by any numerical method. In practice, however, the level set method is the preferred approach since this method handles topological changes such as mergers in a completely natural way. Since simulating epitaxial growth in the so-called layer-by-layer growth regime may require handling the nucleation and subsequent merger of hundreds or even thousands of islands, this advantage has considerable practical significance.

Since the level set method was first introduced by Osher and Sethian [25], level set algorithms have been successfully applied to a wide variety of problems [6, 7, 10, 14, 20, 38, 45]. One can find an extensive review of level set terminology and accomplishments in [21]. In brief, the basis behind the level set method is that any given curve or interface Γ in R^n, bounding an open region Ω, can be represented as the zero level set of a function $\phi(\vec{x},t)$, i.e.

$$\Gamma = \{\vec{x} : \phi(\vec{x},t) = 0\} \tag{1}$$

Given a velocity field $\vec{v}$, one can analyze the motion of the curve Γ by relating it to the motion of the zero level set of ϕ. The partial differential equation that will move the level sets of ϕ by $\vec{v}$ is:

$$\frac{\partial\phi}{\partial t} + \vec{v}\cdot\nabla\phi = 0 \tag{2}$$

The normal vector $\vec{n}$ can be written in terms of ϕ as $\vec{n} = \frac{\nabla\phi}{|\nabla\phi|}$, and since $\vec{v} = v_n\,\vec{n}$, equation (2) is equivalent to

$$\frac{\partial\phi}{\partial t} + v_n\,|\nabla\phi| = 0 \tag{3}$$

which can be referred to as the level set equation.

One of the many advantages of using a level set approach is that the resulting numerical scheme is Eulerian, i.e. only a fixed number of gridpoints are needed. In contrast, when using a front-tracking method [16], one has to account for a potentially large number of gridpoints, depending upon the number of islands. Such methods are computationally expensive and topological changes such as merging may be difficult to handle. Phase field methods [5, 17, 18, 43] are currently popular in modeling solidification problems because boundaries are not explicitly tracked. But phase field methods depend upon a small parameter for interface thickness. Without proper numerical resolution of this parameter, there is no guarantee that the computational results from phase field methods will be accurate or fully converged. A further discussion of this is contained in [20].

The earliest level set approach for solidification type problems coupled level set ideas to boundary integral methods [33]. Later, the boundary integral dependencies were removed, producing a simpler algorithm [7]. For the island dynamics model, the level set method is the best approach due in part to its flexibility and relatively low computational cost. In addition to being able to resolve sharp interfaces (island boundaries in this case), the level set method can handle such topological changes as mergers and breakups, as already mentioned above. When there are several monolayers, the level set function is useful for determining which monolayer a given gridpoint is on. A more detailed discussion of how ϕ is used to compute such statistics as the number of islands, island sizes and island boundary lengths is included in Section 3.7. These statistics are important when comparing the island dynamics model to KMC methods, as well as to experimental data. Good agreement with these statistics provides significant validation of the island dynamics model and the level set based numerical algorithm.

For a continuous adatom density $\rho(x, y, t)$, the diffusion equation is

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (D\,\nabla \rho) \,+\, F - 2\,D\sigma_1\rho^2 \tag{4}$$

where D stands for the diffusion coefficient and F for the flux of atoms to the surface. Realistic parameter values for D are of the order $O(10^5)$ – $O(10^8)$ and F is $O(1)$, making D/F $O(10^5) - O(10^8)$. Note that the flux F can be spatially varying but is usually constant on the spatial scale under consideration here. The length scale is of the order of the lattice spacing and the time scale is of the order of coverage of the substrate. The last term in

(4), $-2\,D\sigma_1\rho^2$, accounts for the loss of adatoms due to nucleation. The factor 2 comes from the assumption that an island of size 2 is stable, i.e. the critical island size is 1. The numerical boundary condition for ρ on a square grid is periodic. For each of the islands, the choice of boundary condition for ρ will depend upon the physics of the model. Irreversible aggregation or growth is a term for the process wherein any adatom hitting an island boundary will attach irreversibly to that island. The results presented in Section 4 are based upon an irreversible aggregation model, so the corresponding choice for the boundary condition imposed on the island boundaries is:

$$\rho = 0 \tag{5}$$

The term $-2\,D\sigma_1\rho^2$ in (4) is an approximation to the loss of adatoms due to nucleation, in that it spreads the loss over the surface. Since $N(t)$ is the total number of islands nucleated up to a time t, this term comes from the assumption that nucleation occurs at a continuous rate given by

$$\frac{dN}{dt} = D\sigma_1 < \rho^2 > \tag{6}$$

where $< \cdot >$ denotes the spatial integral. The coefficient σ_1 is the so-called capture number [2] for nucleation. Note that the spatially varying ρ^2 term in (4) is the same term used to determine $< \rho^2 >$ in (6).

Island boundaries move with a normal velocity v_n. This velocity is determined by the physics of adatoms attaching to the island boundaries and is proportional to the net particle flux to the boundary. In the case of irreversible aggregation, this flux is simply the surface diffusive flux of adatoms, given by $-D\,\nabla\rho$. Let a denote the lattice constant, and a^2 the area per atom. The outward normal velocity is given by

$$v_n = a^2\,[-D\,\nabla\rho\cdot\vec{n}] = -a^2\,D\left[\frac{\partial\rho}{\partial\vec{n}}\right] \tag{7}$$

where $[\cdot]$ denotes the jump across island boundaries in the normal direction (i.e., $[f] = f_+ - f_-$ in which the subscripts $\pm$ denote the two sides of the interface with the normal $\vec{n}$ pointing from "-" to "+") and it is assumed that $[D] = 0$. This is a valid assumption since the interface is not a phase boundary, only a terrace boundary, and energetically the different heights are the same. In other cases, there might be a change in D, e.g. if the boundary was a phase transition in the reconstruction. Note that the method in [7], (upon which this work is based), allows for jumps in the diffusion coefficient for Stefan type problems.

3 Numerical Method

In this work, the model equations (4, 5, 7) are similar to the model equations for the Stefan problem. In [7], a level set method was presented for solving the Stefan problem and for simulating dendritic solidification. The numerical algorithm presented here is based upon the work done in [7]; however, there are important differences between the two problems which have necessitated the development of a numerical method that is flexible and fast enough to be of practical interest. In the following subsections, the numerical algorithm is first outlined. Details are then presented of how this algorithm has been improved and modified (over the method in [7]) in order to solve the island dynamics problem.

3.1 Outline of the Method

After initialization of ρ, ϕ and N, the general outline of the numerical method is as follows:

1. Compute an approximation to the normal velocity field.
2. Update ϕ by solving equation (3).
3. Solve the diffusion equation for ρ, with the internal boundary condition of ρ equal to a constant (e.g. 0) incorporated into the numerical scheme.
4. Update $N(t)$ from equation (6). If $N(t)$ has increased to the next integer value, then a new island is seeded. This event is reflected by the appropriate modification of ϕ at gridpoints near the nucleation site. Return to step 1.

3.2 Normal Velocity

The first step of the numerical algorithm is to compute an approximation to the velocity field v_n. Equation (7) is valid only at the island boundaries, but numerically, it is best to extend v_n off the interface to every gridpoint in order to obtain a smooth velocity field. This minimizes the development of kinks in ϕ.

At the start of the velocity computation, a first order approximation of v_n is computed only at gridpoints bordering or on the fronts that represent the island boundaries on each level. This approximation is obtained by first computing approximations to $\frac{\partial\rho}{\partial x}$ and $\frac{\partial\rho}{\partial y}$ at every gridpoint (x_i, y_j). The first order scheme used is either backward or forward differencing. Special care is taken so that $\frac{\partial\rho}{\partial x}$ and $\frac{\partial\rho}{\partial y}$ are not differenced across island boundaries.

In addition to $\frac{\partial\rho}{\partial x}$ and $\frac{\partial\rho}{\partial y}$, it is also necessary to compute values of the outward normal vector $\vec{n}$ at every gridpoint. The formulas for this are

$$\vec{n} = -\frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|}, \text{ or } (n_x, n_y) = -\frac{(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y})}{\sqrt{\frac{\partial\phi}{\partial x}^2 + \frac{\partial\phi}{\partial y}^2}}$$

In the formulas above, the expressions $\frac{\partial\phi}{\partial x}$ and $\frac{\partial\phi}{\partial y}$ are computed by first order approximations, that are either forward or backward differences in space. The choice of which formula to use depends upon the interface. The formula that involves nodes that are closest to the interface is the one that is chosen.

Once $\frac{\partial\rho}{\partial x}, \frac{\partial\rho}{\partial y}, n_x$ and n_y have been computed at all gridpoints, v_n can be approximated. At every gridpoint, (x_i, y_j), a check is done of its 9 neighbors to determine whether an interface is separating them. If there is at least one neighboring gridpoint, e.g. (x^*, y^*), that is separated from (x_i, y_j) by an island boundary, then v_n is computed from the jump condition given in equation (7). In other words,

$$v_n = \pm a^2 D \left[\frac{\partial\rho}{\partial x} n_x + \frac{\partial\rho}{\partial y} n_y\right]_{(x^*,y^*)}^{(x_i,y_j)} \tag{8}$$

The sign value above is adjusted to the appropriate value, depending on whether (x_i, y_j) is within or outside an island.

Right before islands have merged, their boundaries are close, but not connected to one another. The velocity computation is robust enough to handle the case of gridpoints that are caught in between these islands. This is illustrated in Figure 1, where 3 gridpoints (A,B,C) are separated by 2 island boundaries. At gridpoints A and C, $\frac{\partial\rho}{\partial x}$ is computed by backward and forward difference schemes, respectively. At B, $\frac{\partial\rho}{\partial x}$ is set to 0, since there are no other neighboring points on its same level. Since v_n is computed from jumps in $\frac{\partial\rho}{\partial x}$, the value of v_n at B will be computed either as a jump between A and B or a jump between B and C. Either way, v_n at B will be nonzero. Furthermore, numerical computations have shown that v_n behaves in a relatively smooth manner as islands merge.

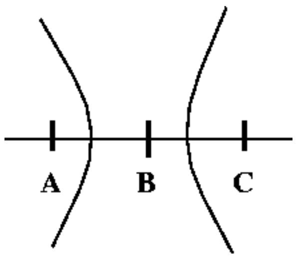

Figure 1: Merger between 2 islands

Grid bias effects may occur since the jump condition in (8) is computed using the first gridpoint found, (x^*, y^*), that is not on the same monolayer as (x_i, y_j). To uniformize the values of v_n in the normal direction, the initial approximation to v_n is refined by solving the equation

$$\frac{\partial v_n}{\partial \tau} + \text{sgn}(\phi)(\vec{n} \cdot \nabla v_n) = 0 \tag{9}$$

for a few iterations in a fictitious time τ, again only at gridpoints bordering the front. When computing ∇v_n, the spatial differences are computed using only values of v_n that have been previously defined (i.e. adjacent to the interface). In the case where the spatial derivative depends on a grid node where v_n is not defined, the derivative is set identically to 0.

Once v_n has been defined adjacent to the interface, those values must be extended to a narrow band (of about 5 gridcells) in order to update equation (3). One could use (9) to extend v_n off the interface to all the gridpoints, as was first suggested in [45] and carefully implemented in [28]. This extension process can be computationally expensive, so instead a simple extension routine for v_n is applied. First, all gridpoints are categorized as being either marked (close to or on the interface) or unmarked (away from the interface). At every marked node, a search is done of its neighbors for unmarked nodes. When an unmarked node is found, the value of v_n at this gridpoint is set equal to an average of v_n values from its marked neighbors, and this unmarked node is changed to a marked node. This process continues until all the nodes have been marked. There are better methods than described above but that they do not change the results here. See also [1] for a related fast method for extending quantities off an interface which depends more directly on the characteristic nature of equation (9) than the method described above.

In Section 4.1, it is shown that for the case of irreversible aggregation, the problem is unstable. Under grid refinement, this physical instability can be seen numerically through a fingering effect on the boundary. Since practical interest lies in simulating the growth and development of many islands, the effect of this instability is minimal; for the coarser grid sizes used in our numerical experiments, islands tend to merge (coalesce) before any sort of fingering takes place. However, when the number of islands is restricted, this effect can be seen as shown in Figure 2(a). One way of preventing dendritic growth from occurring is to compute an average normal velocity for each island, using the formula:

$$\bar{v}_n = \frac{\oint_{\Gamma_i} v_n ds}{\oint_{\Gamma_i} ds}, i = 1, ...N \tag{10}$$

where Γ_i denote the island boundaries. The original expression for v_n in equation (7) corresponds physically to the case of diffusion limited aggregation, whereas equation (10) does not. By using (10), islands are forced to grow isotropically before merging. This is comparable to (but not the same as) adding edge diffusion in KMC methods; both approaches have the effect of producing compact island shapes. The main benefit of using $\bar{v}_n$ is that unstable growth modes are damped out and the fingering effect seen in Figure 2(a) vanishes, as seen in Figure 2(b). $\bar{v}_n$ is computed from v_n by solving (10) before v_n has been extended. For each island, an approximation to the line integral of v_n is computed and divided by the island boundary length. A discussion of how island boundary lengths are computed follows in Section 3.7. After $\bar{v}_n$ is computed at gridpoints on or near the interface, it is extended in a manner similar to v_n.

Another feature that is desirable at times is to restrict the shape of the islands before merger to the same anisotropic shape, e.g. squares. This can be done by making the velocity dependent upon the angle, ν, made between the $x-$axis and the normal vector $\vec{n}$. For example, square shaped islands can be obtained using the velocity $\tilde{v}_n$ given by

$$\tilde{v}_n = \bar{v}_n(|\cos(\nu)| + |\sin(\nu)|)$$

The idea of using the velocity to obtain kinetic crystal shapes is discussed in more detail in [29] and is based on a result obtained in [24] and [36]. Results using either v_n, $\bar{v}_n$ or $\tilde{v}_n$ are shown in Figure 3. In Figure 3(a), the island shapes correspond to the case of diffusion limited aggregation.

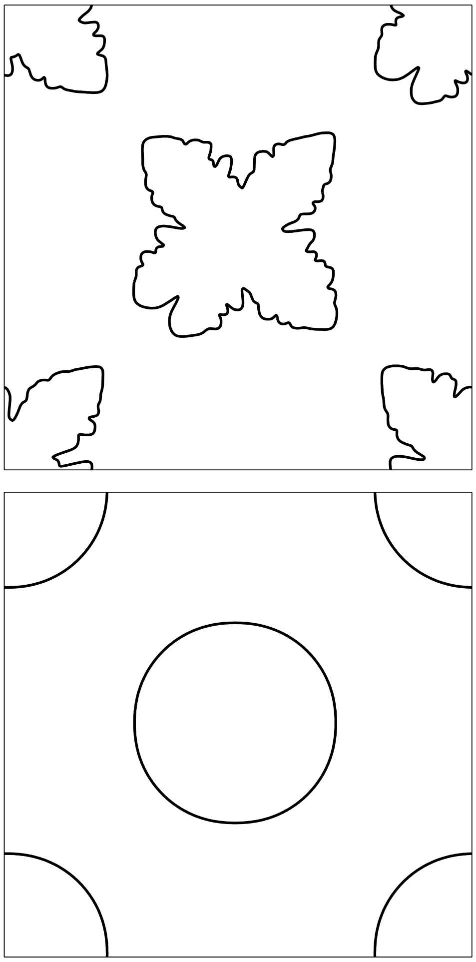

Figure 2: Effect of velocity on instability: (a) v_n from eq. 7 (top) (b) $\bar{v}_n$ from eq. 10 (bottom).

Though the equations for this case are weakly ill-posed, there is no noticeable dendritic fingering in Figure 3(a) due to numerical dissipation on the coarse grid. Figure 3(b) is generated using $\bar{v}_n$; here the growth can be described as isotropic with non-curvature dependent incorporation of adatoms. In the last panel, Figure 3(c), using $\tilde{v}_n$, the island shapes are anisotropic and the absorption of adatoms is dependent upon $\vec{n}$. Note that equations (3) and (9) are formulated using the expression v_n. If, however, a decision is made to use either $\tilde{v}_n$ or $\bar{v}_n$, then (3) and (9) are solved with v_n replaced by either $\tilde{v}_n$ or $\bar{v}_n$.

3.3 Level Set Computational Issues

The level set equation, (3), is solved using a method of lines approach, that employs a 3^{rd} order Runge-Kutta method along with a 3^{rd} order Hamilton-Jacobi ENO scheme. This approach is fairly standard, and the interested reader is referred to [26, 35] for more details. There are some unique aspects of this level set approach that are worth discussing in more detail because they show how useful the level set approach is for problems such as the island dynamics equations. The issues addressed in this section are how ϕ is used to track islands on different monolayers and the issue of reinitialization. Section 3.5 will show how nucleation events are represented using ϕ and Section 3.7 will cover how ϕ can be used to compute various island statistics.

A unique aspect of this numerical method is that only one level set function is needed to keep track of islands on different layers. By using just one level set function rather than one function per layer, the algorithm is kept simple and memory costs are kept down. From one function, one can determine where the island boundaries are, as well as what layer an island is on. This is a useful feature since there is observable roughening and coarsening of thin film surfaces in experimental and KMC data.

The manner in which ϕ is used is in the identification of contour levels $\phi = m$, m being a non-negative integer, with island boundaries on the $(m+1)^{st}$ layer. Thus at gridpoints near nucleation sites, ϕ must be raised to at least the next highest integer value in order to properly represent islands on a new layer. Figure 4 represents a typical profile of ϕ illustrating how ϕ can be used to represent island growth on 3 different layers.

In contrast to other level set applications [28, 38], ϕ is not reinitialized as a distance function after equation (3) has been solved. The reason for this is that other level set applications are only concerned with the $\phi = 0$ contour

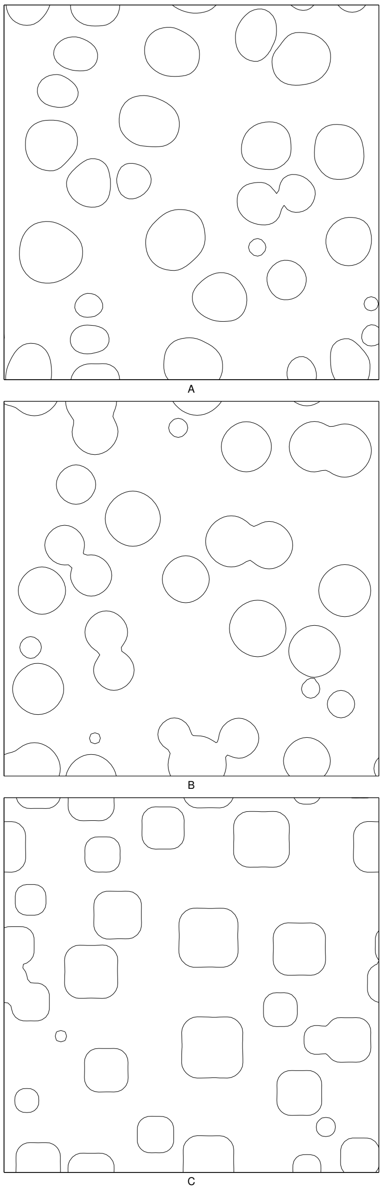

Figure 3: Effect of velocity on island shapes: (A) v_n, (B) $\bar{v}_n$, (C) $\tilde{v}_n$

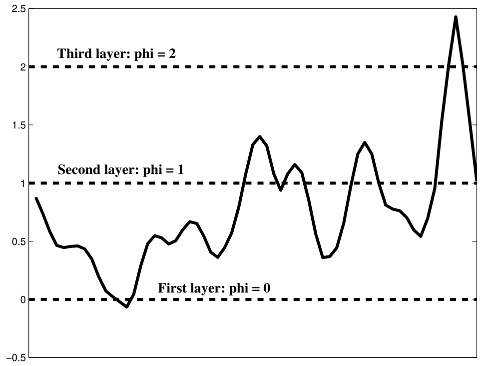

Figure 4: 2 dimensional profile of phi for y held fixed

level; hence, constant reinitialization to a distance function is desirable in order to obtain a very well-behaved function ϕ. However, in the island dynamics model, one has to keep track of many contour levels. Reinitialization of ϕ to an exact distance function from any one specific contour level could introduce spurious peaks in ϕ, thereby creating erroneous islands.

Although reinitialization to a distance function is not possible, due to nucleation, there is still a need to check that ϕ stays well-behaved. Nucleation introduces local regions where ϕ is nonsmooth and may cause oscillations. In order to minimize the impact from new extrema of ϕ formed from nucleation, a search is done for any spurious oscillations in ϕ. At a nucleation site, ϕ is raised to the next highest integer value plus a constant peak height (see equation (21)). Since equation (3) should not introduce any new extrema, any extrema in ϕ should be equal to an integer plus this constant peak height. Thus a criteria is set for determining whether local extrema in ϕ are spurious. First, a search is done for any local maximum values of ϕ. If the fractional parts of these values do not fall within a certain range of the peak height, these values of ϕ are considered spurious oscillations and are reset so that their fractional parts are equal to the peak height.

3.4 Computation of the Adatom Density

The diffusion equation for ρ, (4), is solved using a finite difference scheme. In general, the stability condition for explicit finite difference schemes for parabolic equations leads to a timestep restriction. In particular, for large values of the diffusion coefficient D, this timestep restriction is severe and of the form

$$\Delta t \leq \frac{C\Delta x^2}{D} \quad (C = \text{ constant})$$

Since realistic values of D are of the order $O(10^5) - O(10^8)$, this timestep restriction makes explicit finite difference schemes impractical. In contrast, an implicit finite difference scheme is unconditionally stable. By switching to an implicit method, there is the added benefit of being able to compute on a much longer timescale than is possible for KMC methods. (Of course, the fact that this method is stable using relatively large timesteps is no guarantee that the results are more accurate than KMC methods, which use smaller timesteps.)

The implicit scheme that is applied to (4) assumes that

$$\frac{\partial \rho}{\partial t} \approx \frac{\rho^{k+1} - \rho^k}{\Delta t}$$

and that the spatial derivatives of ρ are approximated using values of ρ^{k+1}, where k is the index for time. One may want to use the Strang splitting method because of the nonlinear term ρ^2 in the last term of (4). In fact, since the nonlinear ordinary differential equation can be solved analytically, the stability concerns introduced by this factor are non-existent. Moreover, the Strang splitting method is not as accurate with large timesteps; i.e., when we use large timesteps to generate a qualitative solution, the results from Strang splitting are worse than the method we now use. The expression ρ^2 in (4) is linearized, so that

$$\begin{aligned} (\rho^2)^{k+1} &\approx (\rho^2)^k + 2\rho^k(\rho^{k+1} - \rho^k) \\ &\approx 2\rho^k\rho^{k+1} - (\rho^2)^k. \end{aligned}$$

Therefore, the implicit scheme produces a time dependent linear system of the form

$$A\rho^{k+1} = b \tag{11}$$

A and b are dependent upon surface morphology (e.g. ϕ^{k+1} which is updated first), and their structure is determined by the spatial discretizations of the derivatives $\frac{\partial^2 \rho}{\partial x^2}$ and $\frac{\partial^2 \rho}{\partial y^2}$ in (4). These discretizations are described in some detail below.

It is sufficient to explain how the spatial derivatives are derived with respect to one variable, since there are no mixed partial derivative terms in

(4). The extension of the following discretization formulas to two dimensions is straightforward and simple. In one dimension, let the spatial gridpoints be denoted by x_i, and let $\rho_i \approx \rho(x_i, \cdot)$ and $\phi_i \approx \phi(x_i, \cdot)$. The standard central finite difference scheme for $\rho_{xx} = \frac{\partial^2 \rho}{\partial x^2}$ (at some fixed time) is

$$(\rho_{xx})_i \approx \frac{\left(\frac{\rho_{i+1}-\rho_i}{\Delta x}\right) - \left(\frac{\rho_i - \rho_{i-1}}{\Delta x}\right)}{\frac{1}{2}(\Delta x + \Delta x)} = \frac{\rho_{i+1} - 2\rho_i + \rho_{i-1}}{\Delta x^2} \tag{12}$$

where Δx is the spatial grid spacing.

For gridpoints near the island boundaries, (12) cannot be used because the first derivatives of ρ are discontinuous across the interface. Furthermore, the internal boundary condition needs to be incorporated into the finite difference stencil. One way of doing this is to use one-sided subcell discretizations. Suppose the boundary point, x_f, falls in between two gridpoints x_i and x_{i+1}. From ϕ, the distances between x_i, x_{i+1} and x_f can be estimated by

$$x_f - x_i \;\approx\; \frac{-(\phi_i - int(\phi_i))\ \Delta x}{(\phi_{i+1} - \phi_i)} \;=\; \theta_1 \Delta x \tag{13}$$

$$x_{i+1} - x_f \;\approx\; \frac{(\phi_{i+1} - int(\phi_{i+1}))\ \Delta x}{(\phi_{i+1} - \phi_i)} \;=\; \theta_2 \Delta x \tag{14}$$

Note that $\theta_1, \theta_2 \in [0, 1]$, $\theta_1 + \theta_2 = 1$ and $\text{int}(\phi_i)$ is an integer part of ϕ_i. To avoid numerical errors caused by division by 0, θ_1 or θ_2 are not used if either is less than Δx^2. If $\theta_1 < \Delta x^2$, then x_f is assumed equal to x_i. If $\theta_2 < \Delta x^2$, then x_f is assumed equal to x_{i+1}. Either assumption is effectively a second order perturbation of the interface location. As will be shown later in the discussion of the Ghost Fluid Method, a second order perturbation of the interface location will not affect the overall first order accuracy of the spatial discretization.

Using the formulas above, one can construct numerical stencils for ρ_{xx} that avoid differencing across the front. By formal truncation analysis, these formulas are only first order accurate ($O(\Delta x)$) and are given by:

$$(\rho_{xx})_i \;\approx\; \frac{\left(\frac{\rho_f - \rho_i}{\theta_1 \Delta x}\right) - \left(\frac{\rho_i - \rho_{i-1}}{\Delta x}\right)}{\frac{1}{2}(\theta_1 \Delta x + \Delta x)} \tag{15}$$

$$(\rho_{xx})_{i+1} \;\approx\; \frac{\left(\frac{\rho_{i+2} - \rho_{i+1}}{\Delta x}\right) - \left(\frac{\rho_{i+1} - \rho_f}{\theta_2 \Delta x}\right)}{\frac{1}{2}(\Delta x + \theta_2 \Delta x)} \tag{16}$$

where ρ_f denotes the value of ρ at x_f and is determined from the boundary condition. Thus, the boundary condition is incorporated into the stencil through the specification of ρ_f. In the case of irreversible aggregation, $\rho_f = 0$.

When equations (12), (15) and (16) are generalized to two dimensions and coupled to a standard implicit scheme for (4), they form a nonsymmetric linear system. In [7], a similar system was solved using the Gauss-Seidel method. The scope of the island dynamics problem is different from the one in [7] in that typical simulations involve much larger system sizes and longer computational times. Also, given that values of D are much larger than in [7], the Gauss-Seidel method would be too slow for all practical purposes. In terms of speed, it is preferable to solve a symmetric linear system because a fast iterative method such as the preconditioned conjugate gradient (PCG) method can be applied. Consequently, what is needed is a discretization of the spatial derivatives that will lead to a symmetric system of equations, yet still include information about the internal boundary condition.

In [21], an alternative strategy for discretizing the internal boundary condition was proposed. This strategy was based on the delta function formulation of [30] which was adapted for level set methods by [38] and [6]. The method proposed in [21] used a penalty method to keep ρ near zero in the hope of approximating the $\rho = 0$ internal boundary condition in equation (5). This is done with a strong spatial sink term that is added to the right hand side of equation (4) in the form $-K\rho\delta$ where δ is a smoothed out delta function that acts in a band near the interface, causing the sink term to have an effect on ρ in a finite band of cells. Usually, this delta function formulation works because the thickness of the band exposed to the delta function shrinks to zero as Δx goes to zero producing a vanishing contribution to the true physics of the problem. Unfortunately, since equation (7) is always discretized using the grid points immediately adjacent to the interface, the contribution of the delta function source term is not diminished as Δx goes to zero. On the contrary, this band of cells where the delta function is applied always makes a large contribution to the velocity of the front. Furthermore, the $\rho = 0$ boundary condition is only obtained as K goes to infinity and $K\rho$ approaches a finite limit (notably $K\rho$ approaches v_n). Neither of these conditions can be obtained numerically and results using this method were unsatisfactory.

In order to alleviate the difficulties associated with the implementation of the internal boundary condition in [7] and [21], equations (15) and (16) are replaced with the following discretizations for ρ_{xx} at gridpoints near the boundary.

$$(\rho_{xx})_i \approx \frac{\left(\frac{\rho_f-\rho_i}{\theta_1\Delta x}\right)-\left(\frac{\rho_i-\rho_{i-1}}{\Delta x}\right)}{\Delta x} \tag{17}$$

$$(\rho_{xx})_{i+1} \approx \frac{\left(\frac{\rho_{i+2}-\rho_{i+1}}{\Delta x}\right)-\left(\frac{\rho_{i+1}-\rho_f}{\theta_2\Delta x}\right)}{\Delta x} \tag{18}$$

These equations were derived using ideas generated by the Ghost Fluid Method [10]. That is, equation (17) is derived using linear extrapolation of ρ from one side of the interface to the other, obtaining

$$\rho_G = \rho_f + (1-\theta_1)\left(\frac{\rho_f-\rho_i}{\theta_1}\right) \tag{19}$$

as a ghost cell value for ρ at x_{i+1}. The standard second order discretization of $\frac{\partial^2\rho}{\partial x^2}$ at x_i using ρ_G at x_{i+1} is

$$(\rho_{xx})_i \approx \frac{\left(\frac{\rho_G-\rho_i}{\Delta x}\right)-\left(\frac{\rho_i-\rho_{i-1}}{\Delta x}\right)}{\Delta x} \tag{20}$$

and the substitution of equation (19) into equation (20) leads directly to (17). Equation (18) is derived similarly. Note that similar ideas were used in [15], but their final matrix was nonsymmetric, making their method more similar to [7] and to related work in [40]. It is interesting to note that a formula similar to (17) appears in [27] for a different problem where the formula was used to alleviate CFL restrictions by assuming that the interface undergoes an $O(\Delta x)$ perturbation in location.

As formulas for the second derivatives, (17) and (18) have $O(1)$ errors. As 0 for the boundary values ρ_f, however, these formulas have $O(\Delta x^2)$ errors. By the maximum principle for parabolic systems, the resulting consistency error is then $O(\Delta x^2)$ everywhere. Computational experiments confirm this fact and show that the resulting scheme is stable.

Assuming that $|\nabla\rho| \neq 0$ and that $\nabla\rho$ is not parallel to the boundary (which is true for the $\rho_f = 0$ problem considered here), a change in boundary value with fixed domain is equivalent to a change in domain boundary with fixed boundary value. This shows that an alternative explanation for the boundary condition is that it corresponds to an $O(\Delta x^2)$ change in the interface location with no change in the boundary value ρ_f. This change in the domain also results in an $O(\Delta x^2)$ consistency error. An advantage of this formulation is that it preserves the property that $\rho > 0$ inside the domain.

While the above argument holds for the one-dimensional case, it is not obvious that it applies or can be extended to multiple spatial dimensions. However, extensive numerical testing of this method was carried out in [8] in one, two and three spatial dimensions for the Poisson equation $\nabla\cdot(k\nabla\rho) = f$ with Dirichlet boundary conditions on irregular domains considering both spatially varying and spatially constant k. In [8], the algorithm showed numerical evidence of second order accuracy in both the L^1 and L^∞ norms as compared to exact solutions for a wide variety of problems. Furthermore, [8] tested this method on an implicit time stepping discretization of the heat equation $\rho_t = \Delta\rho$ with Dirichlet boundary conditions on an irregular domain. For the heat equation, similar second order accuracy in both the L^1 and L^∞ norms was observed in the numerical calculations for one, two and three dimensional numerical examples.

The largest advantage of using (17) and (18) is that they lead to a symmetric linear system. This is best illustrated by considering the local matrix structure corresponding to the two discretizations of ρ_{xx}. Suppose x_f falls between gridpoints x_i and x_{i+1}. Also, assume that $\rho_f = 0$ and θ_1 and θ_2 are defined as in formulas (13) and (14). If one uses the standard discretization (12) for ρ_{xx} at gridpoints x_{i-1} and x_{i+2}, and formulas (15), (16) for ρ_{xx} at x_i and x_{i+1}, then the corresponding local matrix structure for the numerical discretization of ρ_{xx} would look like:

$$\frac{1}{\Delta x^2}\begin{pmatrix} -2 & 1 & 0 & 0 \\ \frac{2}{\theta_1+1} & \frac{-2}{\theta_1} & 0 & 0 \\ 0 & 0 & \frac{-2}{\theta_2} & \frac{2}{\theta_2+1} \\ 0 & 0 & 1 & -2 \end{pmatrix}\begin{pmatrix} \rho_{i-1} \\ \rho_i \\ \rho_{i+1} \\ \rho_{i+2} \end{pmatrix}$$

which is clearly nonsymmetric.

Now if (15) and (16) are replaced by formulas (17) and (18), the analogous local matrix structure is

$$\frac{1}{\Delta x^2}\begin{pmatrix} -2 & 1 & 0 & 0 \\ 1 & -(1+\frac{1}{\theta_1}) & 0 & 0 \\ 0 & 0 & -(1+\frac{1}{\theta_2}) & 1 \\ 0 & 0 & 1 & -2 \end{pmatrix}\begin{pmatrix} \rho_{i-1} \\ \rho_i \\ \rho_{i+1} \\ \rho_{i+2} \end{pmatrix}$$

Thus, (17) and (18) produce the desired symmetric matrix structure. By using these formulas, a tradeoff is made in the accuracy of the interface location in order to produce a symmetric linear system.

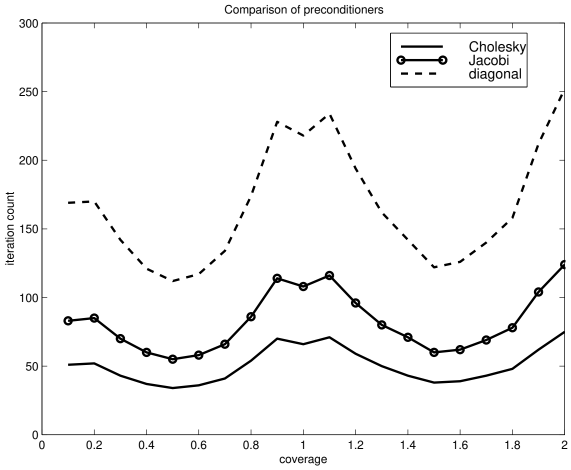

Figure 5: Iteration counts for different preconditioners

The resulting symmetric matrix system, (11), is solved using PCG. Even though PCG is a standard fast iterative solver (see [11] for a survey of iterative solvers), $70-80\%$ of CPU time for typical runs is still spent solving the diffusion equation. Presently, the Cholesky preconditioner is used at every timestep. This preconditioner performs well in comparison to other preconditioners (see Figure 5), but the hope in future work is to improve code speed by applying different optimal preconditioners at different coverages.

3.5 Nucleation

An important feature of this numerical algorithm is the ability to model nucleation events, i.e. the seeding of new islands. The time to nucleate a new island is determined by $N(t)$, which increases by the nucleation rate given by equation (6). Whenever $N(t)$ has increased to the next integer value, that signals the time for a nucleation event. Numerically, every nucleation event will affect values of ρ and ϕ near nucleation sites, which are simply gridpoints at which a new island is centered. These sites are chosen probabilistically, which means that the location of a new island is chosen with a probability that is weighted by the local value of ρ^2. A justification for this choice is discussed in [31].

New islands are represented on the grid as square-shaped. Their ideal area, $2a^2$, is predetermined by the number of atoms in a new island, 2, and the atomic length a. In order to ensure that new islands will not disappear due to inadequate numerical representation, the smallest numerical island size is $4\Delta x^2$, i.e. the area of the square formed by 4 grid cells. If $\Delta x \leq \frac{a}{\sqrt{2}}$, then the grid size is fine enough to represent the ideal area for a new island. On coarser grids ($\Delta x > \frac{a}{\sqrt{2}}$), new island areas are set equal to a value, $4\Delta x^2$, larger than the ideal value, since Δx is the smallest length representable on the grid.

The algorithm for representing a new island is as follows:

1. Choose the nucleation site (x_i, y_j).
2. Set $I =$ lowest integer value $> \phi(x_i, y_j)$.
3. Reset ϕ values at gridpoints near (x_i, y_j) so that locally ϕ is pyramid shaped.

In step 3, local values of ϕ are reset to new values ϕ^{new}. At the nucleation site (x_i, y_j), ϕ^{new} is a maximum value, based on a predetermined peak height. This peak height is between 0 and 1 and is typically set to 0.5, so that

$$\phi^{new}(x_i, y_j) = I + 0.5 \tag{21}$$

The area of the base of the pyramid formed by ϕ^{new} is either $2a^2$ or $4\Delta x^2$, based upon the grid size. Values of ϕ are reset to ϕ^{new} at gridpoints within and neighboring this base area, centered at (x_i, y_j). These local values are determined such that (21) is satisfied and $\phi^{new} = I$ at the base of the pyramid. Within the base area, values of ϕ^{new} range between I and $I + 0.5$. At neighboring gridpoints outside of the base area, ϕ^{new} is extended smoothly to values below I. See Figure 6 for a profile of ϕ^{new} near a nucleation site.

Numerically, nucleation causes local values of ϕ to change. No corresponding changes are made explicitly in ρ. Instead, after a nucleation event has taken place, the normal velocity is computed at all gridpoints. Care is taken so that the velocity is kept equal to 0 at gridpoints where ϕ has been reset to ϕ^{new}. This is necessary so that new islands will not move until ρ has been updated. Implicitly stored in ϕ, the new island will be "felt" by ρ once equation (4) has been solved.

3.6 Timestep Restrictions

In essence, three differential equations are solved at each timestep: (3), (4) and (6). Though the equations are updated by different timestepping

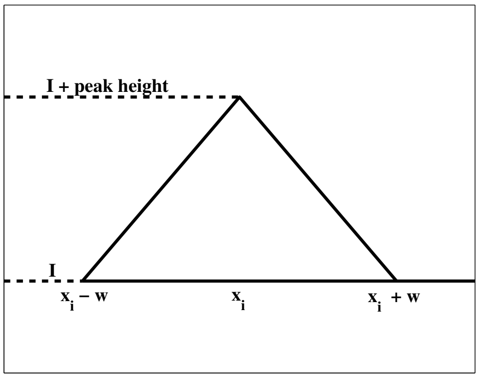

Figure 6: Local profile of ϕ^{new} for y held fixed near a nucleation site x_i. $w = \frac{a}{\sqrt{2}}$

schemes, e.g. Runge-Kutta for (3) and implicit Euler for (4), the actual value of Δt must be the same for all 3 updates in order to avoid synchronization errors. Instead of a constant timestep, Δt is determined adaptively by a number of factors. From the flux term F in equation (4), a restriction is placed on Δt such that

$$F\Delta t < 0.01.$$

This is to guarantee that there is sufficient accuracy in integrating the F term, and that there is no more than 1% coverage per timestep. In order to ensure the stability of the explicit timestepping scheme for (3), the required CFL condition is given by

$$\Delta t < \frac{\Delta x}{\max(v_n)}.$$

Furthermore, since ρ represents a density, it should have nonnegative values at all gridpoints. If, after solving (4) using a timestep Δt, a negative value of ρ is detected, the calculation of the three equations is redone using a reduced timestep (e.g. in our code, we use $\Delta t/4$).

In addition to the timestep restrictions generated by equations (3) and (4), another factor contributing to the adaptive timestep is nucleation. New islands are seeded one at each timestep in order to avoid islands being seeded too closely to one another. Whenever $N(t)$ has increased by more than 1 within a single timestep, Δt is reduced (once again using $\Delta t/4$). This has the effect of slowing down the code during stages of heavy nucleation. In terms of speed, nucleation adds a time scale factor of $O(L^2)$ where L^2 represents the physical system size. The $O(L^2)$ term is due to the fact that the larger L is, the more nucleation events can occur. Future work will involve the implementation of a multiple seeding algorithm that will reduce the $O(L^2)$ factor in the code scaling.

There are continued reductions in Δt until either all the timestep restrictions are met or a minimum value of $\Delta t = 10^{-16}$ is reached. (So far, we have never hit the minimum in any of our *large* number of computations. Conceivably, this could occur and the code would have to be terminated. In that case, we would need to propose an alternate strategy for choosing Δt.)

3.7 Computation of Island Dynamics Statistics

Lastly, ϕ is used for computing a variety of island statistics. These statistics are useful as a barometer of how well the level set method is working and of how accurate the island dynamics model is at describing features of thin film growth. In Section 4, results garnered from these statistics will be presented. How these statistics are obtained using the level set function is described below.

The numerical quantities that are most needed for obtaining quantitative results are the number, areas and perimeters of islands. As discussed in [21], these statistics can be computed by treating islands as the connected components of the contour levels of ϕ. An algorithm has been developed for labeling every connected component, hence every island, and for associating every gridpoint as being within or outside an island. In this algorithm, nodes are categorized as labeled or unlabeled. From an arbitrary starting node, neighbors of labeled nodes are checked to see if they are unlabeled and if there is no boundary (contour level) separating them from their labeled neighbor. If both criteria are met, unlabeled nodes receive the same label as their labeled neighbor. This process continues until either all the nodes or islands have been labeled. Afterwards, the number of islands is computed as the number of labels used.

It is relatively easy to compute island areas and boundary lengths. The approach used here is to triangulate the grid and use ϕ to interpolate the places where the interface cuts through the triangles. From these interpolated values, one can easily compute subcell areas and perimeters. This is illustrated in Figure 7. Within the triangle formed by the gridpoints (x_i, y_j), (x_{i+1}, y_j) and (x_i, y_{j+1}), the front intersects at two interpolated points, $I_A =$

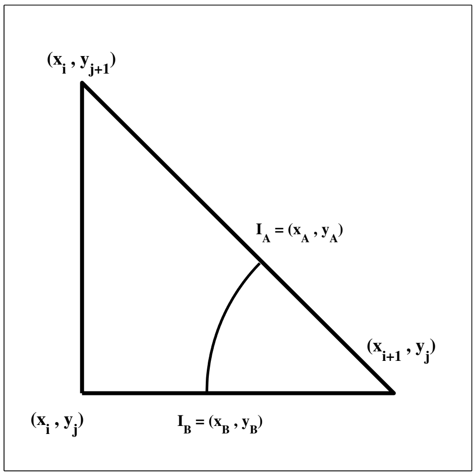

Figure 7: Interpolated values within triangle formed from gridpoints

(x_A, y_A) and $I_B = (x_B, y_B)$. Assuming that (x_{i+1}, y_j) is within an island and (x_i, y_j) and (x_i, y_{j+1}) are outside an island, the contribution to the island boundary length from within the island is

$$\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

and the contribution to the island area is approximated by

$$\frac{1}{2}(y_A - y_B)(x_{i+1} - x_B)$$

using a linear interpolant between I_A and I_B.

4 Computational Results

In this section, some of the results obtained from this numerical method are presented. All of these results pertain to the model of irreversible aggregation.

4.1 Step Trains

Growth might either occur by nucleation and growth on a singular, or perfectly flat surface, or via step-flow on a vicinal, or stepped surface. These steps originate because a crystal is typically cut at a (small) angle with respect to one of the low-index crystal planes. During step flow, all atoms diffuse toward the next step edge before they meet another atom, and nucleate a dimer. Thus, before considering the full island dynamics model with nucleation, it is instructive to consider the case of step trains in the model of irreversible aggregation. Although the terms steps and islands both describe features of crystal growth, there are differences between the two. Islands are isolated regions that are one layer higher than their surrounding regions. Steps are boundaries on a surface substrate, along which the surface changes height by one or more layers. A step train is a series of steps. It is possible to obtain theoretical solutions for the island dynamics equations in the special case of step trains without nucleation. One can then compare these solutions with computational results from the level set method.

In the case of a periodic step train, let the step boundary be 0 by the function $X(y,t)$. The evolution of the step can be modeled by the island dynamics equations: (4), (5), and (7). In the absence of nucleation, these equations reduce to:

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (D\, \nabla \rho) \;+\; F, \qquad X - S < x < X + S \tag{22}$$

$$\rho = 0, \qquad x = X - S, X + S \tag{23}$$

$$v_n = -a^2 D \left(\left.\frac{\partial \rho}{\partial \vec{n}}\right|_{X+S} - \left.\frac{\partial \rho}{\partial \vec{n}}\right|_{X-S} \right) \tag{24}$$

where

$$X_t = v_n \tag{25}$$

and

$$\vec{n} = \frac{(1, -X_y)}{\sqrt{1 + X_y^2}}$$

denotes the normal to the step. The period of the step in the $x-$direction is $2S$. Note that the term $\cdot|_{X+S}$ denotes to the limit from right and the term $\cdot|_{X-S}$ denotes to the limit from the left.

After performing a shift, $x = v_0 t + x'$, to center the step, the evolution equations become

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (D\, \nabla \rho) \;+\; F + v_0 \frac{\partial \rho}{\partial x'}, \qquad X' - S < x' < X' + S \tag{26}$$

$$\rho = 0, \qquad x' = X' - S, X' + S \tag{27}$$

$$v_n = -a^2 D \left(\frac{\partial \rho}{\partial \vec{n}}\bigg|_{X'+S} - \frac{\partial \rho}{\partial \vec{n}}\bigg|_{X'-S} \right) \tag{28}$$

$$X'_t = v_n - v_0 \tag{29}$$

By perturbation analysis of equations (26), (27), (28) and (29), one can obtain leading order approximations to the analytic solutions of the form

$$\begin{aligned} \rho(x', y, t) &= \rho_0(x') + \epsilon\rho_1(x', y, t) + ... \\ X'(y, t) &= \epsilon X'_1(y, t) + ... \end{aligned}$$

where ρ_0, ρ_1 and X'_1 are of the form

$$\begin{aligned} \rho_0 &= b_0 + b_1 x' + b_2 e^{-\lambda x'} \\ \rho_1 &= e^{iky+\omega t}(\hat{\rho}_+ e^{\alpha_+ x} + \hat{\rho}_- e^{\alpha_- x}) \\ X'_1 &= \hat{X}_1 e^{iky+\omega t} \end{aligned}$$

in which

$$\begin{aligned} v_0 &= 2a^2 FL \\ \lambda &= 2a^2 D^{-1} FL \\ b_0 &= (2a^2)^{-1} coth(\lambda L) \\ b_1 &= -(2a^2 L)^{-1} \\ b_2 &= -(2a^2 sinh(\lambda L))^{-1} \end{aligned}$$

and for the second order approximation, one will find that in leading order,

$$\omega = a^2 D(\rho'_0(S) + \rho'_0(-S))|k|.$$

Further analysis show that $(\rho'_0(S) + \rho'_0(-S)) > 0$, hence $\omega > 0$. Thus there exists a class of unstable solutions.

Computational results can be obtained by applying the level set method to the steptrain model, with no nucleation. Consider the case of initial steps $X(x, t = 0) = \epsilon \cos(kx)$ with initial density $\rho(x, y, t = 0) = \rho_0(x) + \epsilon\rho_1(x, y, t = 0)$. These two equations are compatible for ϵ small. (If ϵ is too large, then the step velocities derived from X and ρ will not match up and the two solutions will not agree.) In the simple case of a straight step, (i.e. where $\epsilon = 0$), the theoretical solution matches up with the computed solution from the level set method. The step travels at the correct velocity v_0 and one can

Grid Size	Max Error	Order
50 X 50	1.366×10^{-2}	
100 X 100	6.506×10^{-3}	1.070
200 X 200	3.206×10^{-3}	1.021
400 X 400	1.591×10^{-3}	1.011

Table 1: Step trains: error measured between the analytic solution with $F = 1$ and level set results for the case of a straight step, i.e. $\epsilon = 0$.

measure the error between the analytic and computed solutions. In Table 1, the error is recorded for increasingly finer grid sizes. The corresponding order of accuracy is one, meaning the numerical method is $O(\Delta x)$.

For ϵ small, a more interesting result occurs when the level set method produces the instability predicted from the theoretical solution. This instability can be seen in the development of dendrites over time, as shown in Figures 8 and 9. The solutions obtained using the level set method are accurate up to the time at which spurious oscillations occur. In Table 2, the error is measured between the analytic and computed solutions for the case corresponding to Figure 8 (top), third curve from the left. Similar to the case for $\epsilon = 0$, the measured order of accuracy in Table 2 shows that the level set method is first order accurate for $\epsilon > 0$ up to small times. For later times, the dendritic fingering from level set results (as seen in Figures 8 and 9) shows the inherent instability of the problem. Oscillations due to the physically correct unstable growth have the effect of amplifying roundoff and discretization errors. These oscillations are quickly magnified, causing a numerical instability at later times.

4.2 Island Dynamics Computations

For the full island dynamics model, one of the earliest checks performed was related to mass conservation. Under a constant flux term F, the coverage on the surface should be $\Theta = Ft$, where t is the computational time. However, it is known that level set methods have a (small) mass loss. In Figure 10, we plot the actual coverage on the surface as a function of time. The actual coverage is measured by simply adding up the area of all islands on the surface (this neglects the adatoms, but for typical values of D/F the number of single adatoms is several orders of magnitude smaller than the

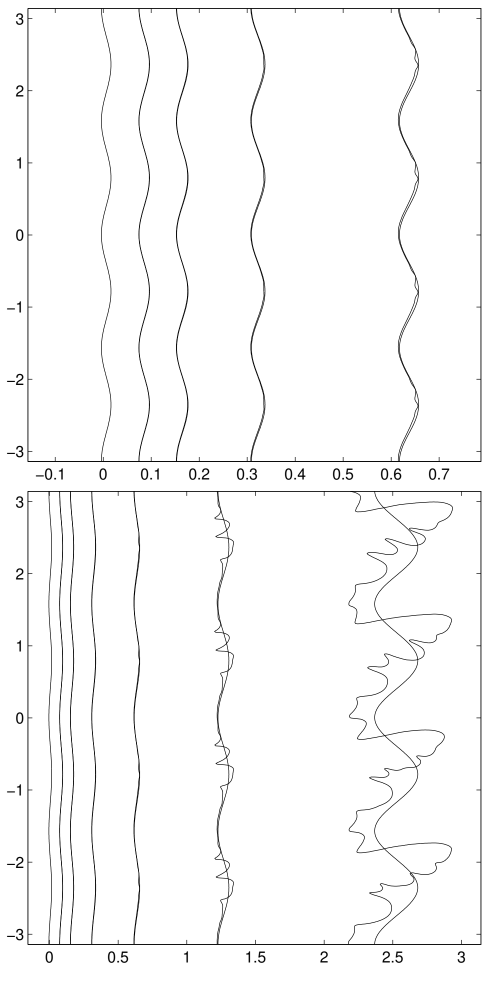

Figure 8: Step trains: analytic (smooth) and computed (dendritic) solutions. $D = 20$, $F = 1$, $\epsilon \hat{X}_1 = -0.01, k = 4, \omega = 6.8789, t = 0.0, 0.0125, 0.025, 0.05, 0.1, 0.2, 0.4$ (bottom). First five curves zoomed in (top).

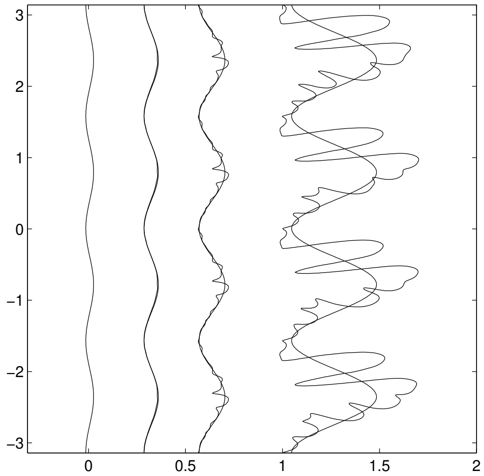

Figure 9: Step trains: analytic (smooth) and computed (dendritic) solutions. $D = 10, F = 1, \epsilon\hat{X}_1 = -0.02, k = 4, \omega = 11.946, t = 0.0, 0.05, 0.1, 0.2.$

Grid Size	Max Error	Order
32 X 32	1.749×10^{-2}	
64 X 64	1.030×10^{-2}	0.764
128 X 128	4.555×10^{-3}	1.177
256 X 256	2.302×10^{-3}	0.984
512 X 512	1.166×10^{-3}	0.982

Table 2: Step trains: error measured between the analytic solution and level set results for the case corresponding to $D = 20, F = 1, \epsilon\hat{X}_1 = -0.01, k = 4, \omega = 6.8789, t = 0.025.$

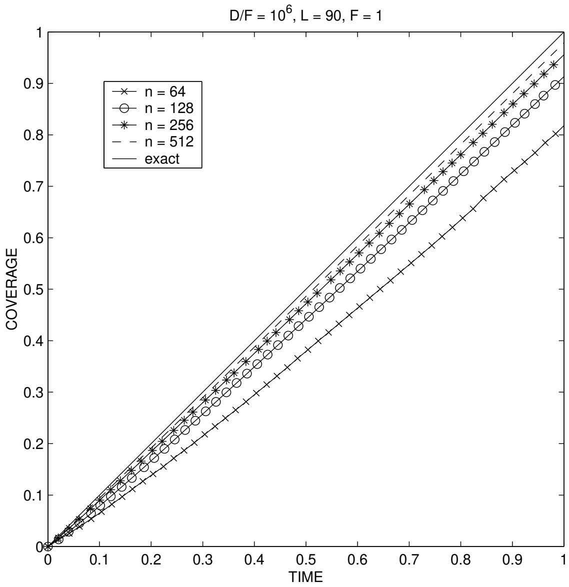

Figure 10: Conservation of mass under grid refinement

number of adatoms that have been incorporated into the islands). The solid (straight) line represents the ideal case of perfect mass conservation. We see that there is a small mass loss in our method due to numerical dissipation, but, in general, mass is conserved very well. In particular, the mass loss can be controlled under grid refinement and the order of accuracy for mass conservation is $O(\Delta x)$.

To validate the island dynamics model and its numerical implementation, results have been compared to those obtained from KMC simulations. The KMC simulations were carried out on a square lattice and included a process for fast edge diffusion so that island shapes are compact. The focus here is on the submonolayer regime for the case of irreversible aggregation, where the nucleation rate is given by equation (6).

With the approach discussed here, one can obtain the entire island size distribution (including *spatial* information). Scaling of the size distribution for different values of the coverage Θ and D/F in comparison with the KMC simulations is shown in Figure 11. The filled symbols in Figure 11 correspond to scaled numbers from the level set method (LS), while the open symbols correspond to data obtained from KMC methods. The agreement between the two methods is very good for two values of D/F and two values of

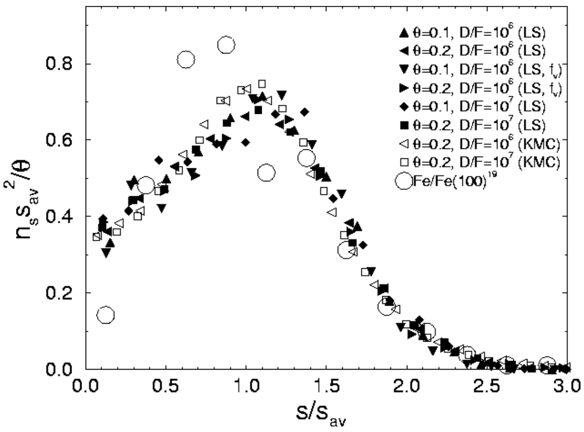

Figure 11: Comparison of scaled island-size distributions, where n_s is the number of islands of size s, $s_{\rm av}$ is the average island size, and Θ is the coverage. The experimental data for Fe/Fe(100) comes from [37].

Θ. Experimental data is represented on the graph by the large circular symbols. For larger values of s/s_{av}, the agreement between the experimental and simulated data is also very good. The discrepancy between the two types of data for smaller values of s/s_{av} may be due to noise. Since the correct island size distribution is obtained, this model captures many of the relevant processes in the submonolayer aggregation regime.

The island dynamics model has no inherent limitations that restrict its validity to submonolayer growth. In fact, one of the advantages of the level set method is that it can describe the merger (coalescence) of islands within its framework, without specification of any extra parameters. For many technological applications, it is of interest whether a thin film grows in a layer-by-layer fashion, or whether it becomes rough (many layers exposed at the same time). A quantity that experimentally determines this is the RHEED (reflective high energy electron diffraction) signal. During layer-by-layer growth, the RHEED signal oscillates with a periodicity that corresponds to the completion time for each layer. There is evidence that the RHEED signal is due to variations in the step edge density [34]: at layer completion, there are very few step (island) edges, while at a partially grown layer, there are many exposed step (island) edges.

In Figure 12 results are shown for the step edge density oscillations for different values of D/F. The step edge density oscillates with an amplitude

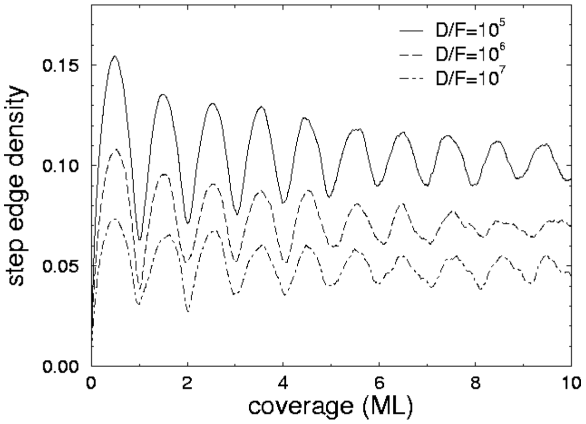

Figure 12: Step-edge density for different values of D/F obtained with the level-set method. The data represents the average over 5 independent realizations, with $L/a = 300$ and 400 gridpoints (laterally).

that is damped. The latter is due to progressive roughening of the surface. For higher values of D/F, the magnitude is lower, because there are fewer, bigger islands on the surface. Thus, this model qualitatively reproduces the correct physics in the multilayer regime as well. A more quantitative study with a comparison to the corresponding KMC data is currently being pursued, and will be published elsewhere.

5 Conclusions

In this article, we have presented further developments and results from a level set based method that simulates the island dynamics model for the growth of epitaxial thin films. Many parts of this numerical method were originally proposed in [21], although this paper makes some notable improvements. In particular, a new treatment was proposed for the internal boundary condition that is mush faster than the method in [7] and does not needlessly smear out the interface as originally proposed in [21]. Results using our numerical formulation have been previously published in [4, 12]. In this paper, we have focused on explaining the numerical algorithm, as well as the numerical challenges that needed to be overcome in order to accurately solve and evolve the equations of motion. We have also detailed how the numerical method was used to address some of the unique issues arising from the island dynamics model, such as nucleation and multilayer growth.

The computational results presented here are from the model of irreversible aggregation. In the case of step trains, the level set method is first order accurate when compared to an analytic solution. For the full island dynamics model with nucleation, good agreement with corresponding results obtained from KMC simulations shows that the method is accurate and captures the correct physics. In the future, we plan to extend the model to incorporate a more general boundary condition, corresponding to reversible aggregation. We also plan to improve the overall speed and accuracy of the code through further development of the method. In conclusion, we feel that the level set method applied to the island dynamics model is a useful and important addition to computational methods for simulating thin film growth.

References

[1] D. Adalsteinsson and J. .A. Sethian, The fast construction of extension velocities in level set methods, *J. Comput. Phys.* **148**, 2 (1999).

[2] G. S. Bales and D. C. Chrzan, Dynamics of irreversible island growth during submonolayer epitaxy, *Phys. Rev. B* **50**, 6057 (1994).

[3] A.-L. Barabasi and H. E. Stanley, *Fractal Concepts in Surface Growth* (Cambridge University Press, Cambridge, England, 1995).

[4] R. E. Caflisch, M. F. Gyure, B. Merriman, S. Osher, C. Ratsch, D. D. Vvedensky and J. J. Zinck, Island dynamics and the level set method for epitaxial growth, *Appl. Math. Lett.* **12**, 13 (1999).

[5] G. Caginalp and E. A. Socolovsky, Computation of sharp phase boundaries by spreading: the planar and spherically symmetric cases, *J. Comput. Phys.* **95**, 85 (1991).

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

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

[8] L. T. Cheng, R. Fedkiw, F. Gibou and M. Kang, A symmetric method for implicit time discretization of the Stefan problem, preprint, [submitted to *J. Comput. Phys.*].

[9] S. Clarke and D. D. Vvedensky, Origin of reflection high-energy electron-diffraction intensity oscillations during molecular-beam epitaxy: A computational modeling approach, *Phys. Rev. Lett.* **58**, 2235 (1987).

[10] R. Fedkiw, T. Aslam, B. Merriman and S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *J. Comput. Phys.* **152**, 457 (1999).

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations, 3rd ed.*, (Johns Hopkins University Press, Baltimore, 1996).

[12] M. F. Gyure, C. Ratsch, B. Merriman, R. E. Caflisch, S. Osher, J. J. Zinck and D. D. Vvedensky, Level set methods for the simulation of epitaxial phenomena, *Phys. Rev. E* **58**, R6931 (1998).

[13] *Proceedings of the Workshop on Virtual Molecular Beam Epitaxy*, edited by M. F. Gyure and J. J. Zinck, *Comp. Mat. Sci.* **6**, pp. 113-196 (1996).

[14] T. Hou, Z. Li, S. Osher and H.-K. Zhao, A hybrid method for moving interface problems with application to the Hele-Shaw flow, *J. Comput. Phys.* **134**, No. 2, 236 (1997).

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

[16] D. Juric and G. Tryggvason, A front tracking method for dendritic solidification, *J. Comput. Phys.* **123**, 127 (1996).

[17] A. Karma and W. J. Rappel, Phase-field method for computationally efficient modeling of solidification with arbitrary interface kinetics, *Phys. Rev. E* **53**, 3017 (1996).

[18] R. Kobayashi, Modeling and numerical simulations of dendritic crystal growth, *Physica D* **63**, 410 (1993).

[19] A. Madhukar and S. V. Ghaisas, The nature of molecular beam epitaxial growth examined via computer simulations, *CRC Crit. Rev. Sol. State and Mater. Sci.* **14**, 1 (1988).

[20] B. Merriman, J. Bence and S. Osher, Motion of multiple junctions: A level set approach, *J. Comput. Phys.* **112**, 334 (1994).

[21] B. Merriman, R. Caflisch, and S. Osher, Level set methods, with an application to modeling the growth of thin films, *Free Boundary Value Problems, Theory and Applications*, edited by I. Athanasopoulos, G. Makrakis and J. F. Rodrigues, (CRC Press, Boca Raton, 1999).

[22] B. Merriman, R. Caflisch, S. Osher, C. Ratsch, S. Chen, and M. Kang, Island dynamics and level set methods for continuum modeling of epitaxial growth, to appear in conference proceedings.

[23] H. Metiu, Y.-T. Lu, and Z. Y. Zhang, Epitaxial growth and the art of computer simulations, *Science* **255**, 1088 (1992).

[24] S. Osher and B. Merriman, The Wulff shape as the asymptotic limit of a growing crystalline interface, *Asian J. Math* **1** (3), 560 (1997).

[25] S. Osher and J. A. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.* **79**, 12 (1988).

[26] S. Osher and C.-W. Shu, High-order essentially non-oscillatory schemes for Hamilton-Jacobi equations, *SIAM J. Numer. Analys.* **28**, 907 (1991).

[27] R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield and M. L. Welcome, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, *J. Comput. Phys.* **120**, 278 (1995).

[28] D. Peng, B. Merriman, H.-K. Zhao, S. Osher, and M. Kang, A PDE-based fast local level set method, *J. Comput. Phys.* **155**, 410 (1999).

[29] D. Peng, S. Osher, B. Merriman, and H.-K. Zhao, The geometry of Wulff crystal shapes and its relations with Riemann problems, *Proc. Int. Conf. on Nonlinear PDEs and Applications*, edited by G.-Q. Chen and E. D. Benedetto, *Contemporary Math.* **238**, pp. 251-303 (1999).

[30] C. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25**, 220 (1977).

[31] C. Ratsch, M. F. Gyure, S. Chen, M. Kang, and D.D. Vvedensky, Fluctuations and scaling in aggregation phenomena, preprint, [submitted to *Phys. Rev. B*].

[32] C. Ratsch, P. Ruggerone, and M. Scheffler, in *Surface Diffusion: Atomistic and Collective Processes*, edited by M. C. Tringides (Plenum, New York, 1997), pp. 83-101.

[33] J. Sethian and J. Strain, Crystal growth and dendritic solidification, *J. Comput. Phys.* **98**, 231 (1992).

[34] T. Shitara, D. D. Vvedensky, M. R. Wilby, J. Zhang, J. H. Neave, and B. A. Joyce, Step-density variations and reflection high-energy

electron-diffraction intensity oscillations during epitaxial growth on vicinal GaAs(001), *Phys. Rev. B* **46**, 6815 (1992).

[35] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, II , *J. Comput. Phys.* **77**, 439 (1994).

[36] P. Soravia, Generalized motion of a front propagating along its normal direction - a differential games approach, *Nonlinear Analysis TMA* **22**, 1247 (1994).

[37] J.A. Stroscio and D.T. Pierce, Scaling of diffusion-mediated island growth in iron-on-iron homoepitaxy, *Phys. Rev. B* **49** 8522 (1994).

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

[39] J. Y. Tsao, *Materials Fundamentals of Molecular Beam Epitaxy*, (Academic Press, Boston, 1993).

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

[41] J. Villain, Continuum models of crystal growth from atomic beams with and without desorption, *J. Phys. I* **1**, 19 (1991).

[42] D. D. Vvedensky, A. Zangwill, C. N. Luse, and M. R. Wilby, Stochastic equations of motion for epitaxial growth, *Phys. Rev. E* **48**, 852 (1993).

[43] S-L. Wang and R. F. Sekerka, Computation of the dendritic operating state at large supercoolings by the phase field model, *Phys. Rev. E* **53**, 3760 (1996).

[44] J. D. Weeks and G. H. Gilmer, Dynamics of crystal growth, *Adv. Chem. Phys.* **40**, 157 (1979).

[45] H.-K. Zhao, T. Chan, B. Merriman and S. Osher, A Variational Level Set Approach to Multiphase Motion, *J. Comput. Phys.* **127**, 179 (1996).