University of California

Los Angeles

Rate Equations in Materials Science and Simulation of Multiphase Flows

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mathematics

by

Frédéric G. Gibou

2001

The dissertation of Frédéric G. Gibou is approved.

Russell E. Caflisch

Björn Engquist

Mark Goorsky

Mark Gyure

Stanley Osher, Committee Chair

University of California, Los Angeles

2001

To my wife Michelle,

and my family.

For their support and encouragement.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I wish to express my deepest gratitude to my advisors and mentors, Professors Russel Caflisch, Ronald Fedkiw and Stanley Osher, for their constant guidance, encouragement and endless support throughout my graduate years at UCLA.

I wish to further thank Professor Russel Caflisch who has advised me and actively participated on my research on Rate Equations.

I also wish to thank Professor Dimitri Vvedensky, Dr. Mark Gyure and especially Dr. Christian Ratsch for sharing their ideas and expertise in the field of Materials Science.

Part of this work was supported by the NSF and DARPA through cooperative agreement DMS-9615854 as part of the Virtual Integrated Prototyping Initiative.

Abstract of the Dissertation

Rate Equations in Materials Science and Simulation of Multiphase Flows

by

Frédéric G. Gibou

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2001

Professor Stanley Osher, Chair

The work of this thesis is divided into two main subjects: The first concerns rate equations arising in materials science, and the second the design of a numerical method to account for the thermal conductivity in flame propagation.

To compute the capture numbers σ_s used in mean-field rate equations that describe epitaxial growth, we use the *Island-Dynamics* model based on the level-set technique. In this approach, islands grow with a velocity that is computed from solving the diffusion equation for the adatom concentration. The capture number for each island is then calculated by integrating the growth velocity of an island around the island boundary. Thus, our method by construction includes all spatial correlations between islands. We found that the functional form of the σ_s is, to first approximation, affinely dependent on the island sizes. We then derive an analytical formula for the σ_s at steady state and find scaling laws for the adatom density and total number density.

The treatment of the second subject is twofold. First, we design a symmetric implicit method for solving the Stefan problem for the temperature; then we apply this algorithm to accurately account for the thermal conductivity in the

context of multiphase flame propagation.

CHAPTER 1

Rate Equations and Capture Numbers with Implicit Islands Correlations

1.1 Introduction

One of the challenges that material scientists currently face is to build the next generation of semiconductor materials. Applications include device structures in solid-state physics, electronics and optoelectronics that will be used, for example, in satellite communication. The development and refinement of molecular beam epitaxy (MBE) has been among the most important advances towards an engineering solution, because this technique obtains surfaces that are extremely clean. Broadly stated, MBE is simply crystallization by condensation or reaction of a vapor in ultra high vacuum, and under this process materials may be grown layer by layer (see Fig.(1.1)). Of all the different processes that occur during growth (see Fig.(1.2)), we only consider here deposition, nucleation and attachment to islands. That is, we place ourselves in the case of 'irreversible' aggregation. The thin-film growth can be monitored using a RHEED (Reflection High Energy Electron Diffraction) signal, whereas an STM (scanning tunneling microscope) allows us to look at its surface morphology after the growth (see Fig.(1.3)).

The ultimate goal is to be able to control the growth of the materials to obtain the desired surface morphology. We therefore need to understand the physics at the nanoscale level which is best done using a model-based approach rather than an empirical one. It is in part for these reasons that advances in modeling crystal growth under MBE is such an important topic. Modeling early stages of epitaxial growth is itself of great practical interest for material scientists since the surface morphology in the submonolayer regime greatly influences the later stages of the growth process and therefore the properties of a thin-film device. Models to study epitaxial growth that have successfully reproduced such quantities as the cluster size distribution (CSD) of islands on the surface are kinetic Monte Carlo

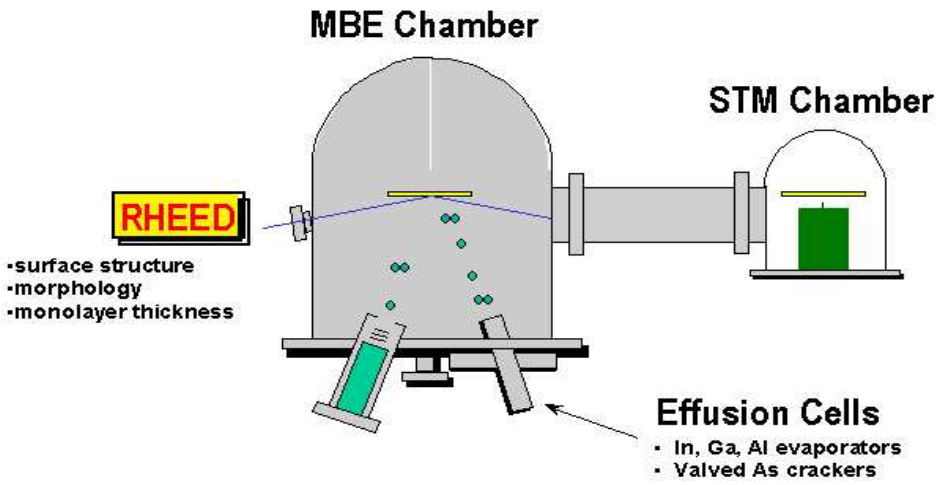

Figure 1.1: Atoms from the effusion cells are deposited on a flat substrate at the rate of F monolayers per seconds. There, they diffuse freely until they meet to form a dimer (two atoms) or attach to an existing group of atoms called islands.

(KMC) methods [1, 2, 3] or, more recently, the *Island-Dynamics* model based on the level-set method [4]. However, both of these methods include stochastic processes, so that many simulations need to be done (and averages need to be taken) in order to produce physical quantities of interest.

On the other hand, mean-field rate equations, introduced almost 30 years ago [5, 6], offer a completely deterministic description of epitaxial growth. Such equations for the submonolayer regime (without detachment or evaporation) typically read as

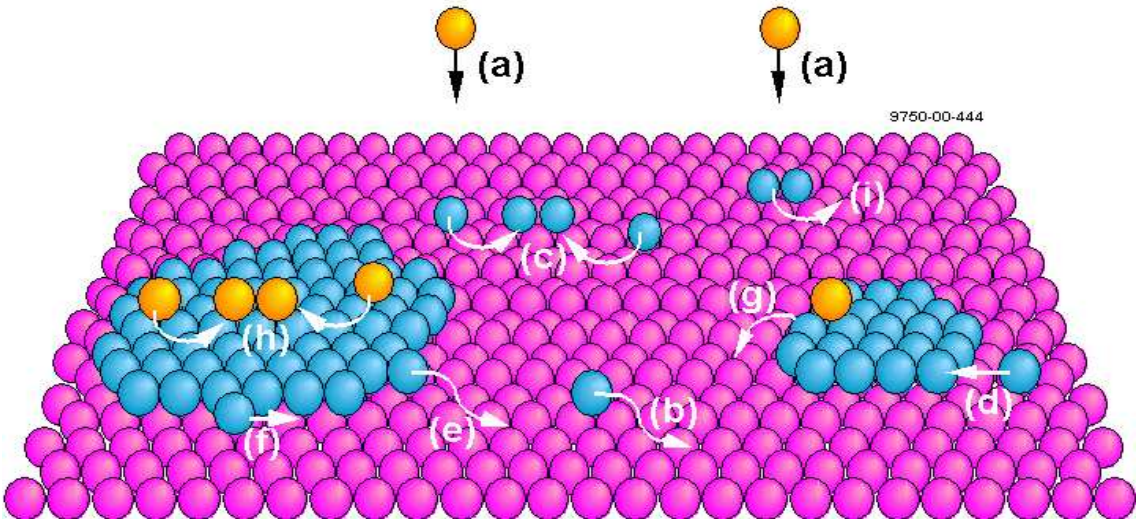

Figure 1.2: Different processes during Epitaxial growth. (a) deposition, (b) diffusion, (c) nucleation, (d) attachment, (e) detachment, (f) edge diffusion, (g) diffusion down step, (h) nucleation on top of islands, (i) dimer diffusion.

$$\frac{dn_1}{dt} = F - 2D\sigma_1 n_1^2 - Dn_1 \sum_{s>1} \sigma_s n_s, \tag{1.1}$$

$$\frac{dn_s}{dt} = Dn_1(\sigma_{s-1} n_{s-1} - \sigma_s n_s) \quad \text{for all } s > 1. \tag{1.2}$$

where n_s is the density of islands of size s, n_1 is the density of adatoms, D is the diffusion constant, F is the deposition flux, and σ_s are the capture numbers. Clearly, deterministic equations that accurately reproduce the relevant physical quantities would be of great practical value; they usually are easier to understand and analyze, and can yield theoretical insights that cannot be reached within a

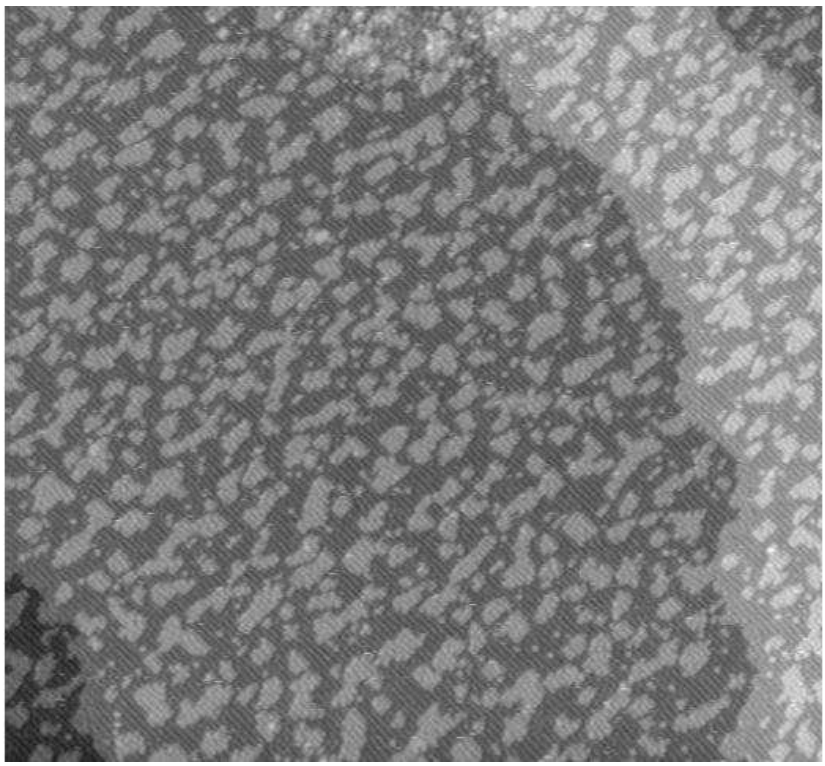

Figure 1.3: STM image of a thin-film obtained by Barvosa-Carter and Whitman at NRL. The different levels of grey represent different layers. The clear patches are islands forming on an existing layer.

stochastic framework. For example, nucleation theory predicts scaling of island densities as a function of temperature and deposition flux [6]. This result has been confirmed in simulations and experiment, and is, in fact, used to extract microscopic parameters such as diffusion constants from experimental measurements [7, 8]. Thus far, however, there has been no success in finding deterministic equations that, when integrated, produce the correct results for quantities that include spatial information. In particular, no deterministic approach has reproduced the CSD as observed in experiment and KMC simulations [9]. The main problem with using rate equations in the submonolayer regime is that the functional form for σ_s is not known. Bales and Chrzan [10] proposed an analytical

formula in terms of modified Bessel's functions. Their work is based on the *mean-field* assumption which states that at every point outside of an island, the local densities take on their average values, so that the distribution of surrounding islands is independent of its size. The integration of rate equations using this analytical form for the capture numbers gives excellent agreement with KMC simulations for the adatom density and also for the total number density. However, it fails to reproduce the correct cluster size distribution, the reason being that the mean-field assumption excludes correlations between islands. Bartelt and Evans addressed this issue and numerically computed capture numbers by monitoring the aggregation of diffusive adatoms to the islands using KMC simulations with a point-island model [11]. In the steady-state regime, the dependence of the capture numbers on the island size exhibits a plateau for islands smaller than the average size and an affine part for islands bigger than the average size. They then derived an asymptotic limit for the cluster size distribution using the resulting capture numbers and obtained excellent agreement with point-island KMC simulations results. However, the growth and subsequent correlations of islands are omitted in this approach, since a point-island model explicitly excludes this feature. More recent studies [12, 13] that include the spatial extent of islands still reveal a (less pronounced) plateau for the capture numbers. In these simulations, the capture numbers were measured for a fixed coverage and a geometry that was obtained from scanning tunneling microscopy images.

In this thesis we propose a new numerical method for computation of the capture numbers to remedy these issues. Our approach employs the *Island-Dynamics* model based on the level-set method [4, 15, 16, 17], which is a general technique for simulating the motion of moving boundaries. We find that the dependence of the capture numbers on the island size is, to first approximation, affine. In particular, there is no plateau as found in previous works [11, 12, 13]. We have con-

firmed this result by computing the capture numbers self-consistently and have explained the difference with the results obtained using a point-island model by linking the functional form of the capture numbers to the amount of nucleation. Further, we have derived scaling laws using the new form of the capture numbers.

1.2 Capture numbers and capture zones

In the *Island-Dynamics* model, the boundary of an island is represented as the zero level-set of a smooth function ϕ. The evolution of the boundary is then dictated by the evolution of ϕ which obeys the advection equation $\phi_t + v_n |\nabla\phi| = 0$, where v_n is the local normal velocity of the island boundary. In the case of irreversible aggregation $v_n = a^2[D\nabla\rho]$, where [.] refers to the jump across the boundary of the island, ρ is the adatom density and a is the lattice constant. The adatom density is considered continuous and satisfies the following diffusion equation:

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

In the case of irreversible aggregation, the boundary condition imposed is $\rho_{|\Gamma} = 0$ where Γ represents the boundary of islands. We note here that the seeding of new islands is performed in a probabilistic fashion in the *Island-Dynamics* model to ensure the correctness of the CSD as described in [4]. Therefore this description properly models the inherent stochastic nature of nucleation.

On the other hand mean-field rate equations offer a completely deterministic description of epitaxial growth. Such equations for the submonolayer regime (without detachment or evaporation) typically read as

$$\frac{dn_1}{dt} = F - 2R_{agg}(1) - \sum_{s>1} R_{agg}(s), \tag{1.3}$$

$$\frac{dn_s}{dt} = R_{agg}(s-1) - R_{agg}(s) \quad \text{for all } s > 1. \tag{1.4}$$

where $R_{agg}(s)$ is the rate at which islands of size s capture a diffusive adatom and become islands of size $s+1$.

What rate equations model is how islands form and grow under a deposition flux F of adatoms. More precisely, Eq.(1.4) says that the density n_s of islands

of size s is increased when islands of size $s-1$ capture a diffusive adatom, and decreases when islands of size s capture a diffusive adatom to become of size $s+1$. Eq.(1.3) says that the adatom density n_1 is increased by the deposition flux F, and decreased every time two adatoms come together to form a dimer (the factor of 2 in $2R_{agg}(1)$ accounts for the fact that when a dimer is formed, 2 adatoms are lost on the surface) or when an adatom attaches to an existing island ($\sum_{s>1} R_{agg}(s)$).

The rate of aggregation $R_{agg}(s)$ depends on the diffusive constant D and on the probability that an adatom meets an island of size s. The probability of finding an adatom is given by the adatom density n_1 and that of finding an island of size s is given by the density n_s. If these probabilities were independent, the rate of aggregation would be $R_{agg}(s) = Dn_1n_s$, but to account for some independence we write $R_{agg}(s) = D\sigma_s n_1 n_s$. The capture number σ_s measure how effectively islands of size s compete for the available adatoms. Therefore the rate equations are written as

$$\frac{dn_1}{dt} = F - 2D\sigma_1 n_1^2 - Dn_1 \sum_{s>1} \sigma_s n_s, \tag{1.5}$$

$$\frac{dn_s}{dt} = Dn_1(\sigma_{s-1}n_{s-1} - \sigma_s n_s) \quad \text{for all } s > 1. \tag{1.6}$$

We propose to find the functional form of the capture numbers using the *Island-Dynamics* model. The capture number of each island is computed by monitoring the rate of aggregation of adatoms to that island. Consider an island of size $\tilde{s}$ with boundary $\Gamma_{\tilde{s}}$. Growth of this island as described by velocity v_n is due to migration of adatoms toward this island (and subsequent capture), so the rate of aggregation of adatoms is equal to the rate of change in area (see Fig.(1.4)). This is expressed easily in terms of the level-set function as $\int_{\Gamma_{\tilde{s}}} v_n d\Gamma_{\tilde{s}}$,

so that the capture number of this island is given by

$$\sigma_{\tilde{s}} = \frac{\int_{\Gamma_{\tilde{s}}} v_n d\Gamma_{\tilde{s}}}{Dn_1}. \tag{1.7}$$

We emphasize that the main originality in this approach is that we allow each

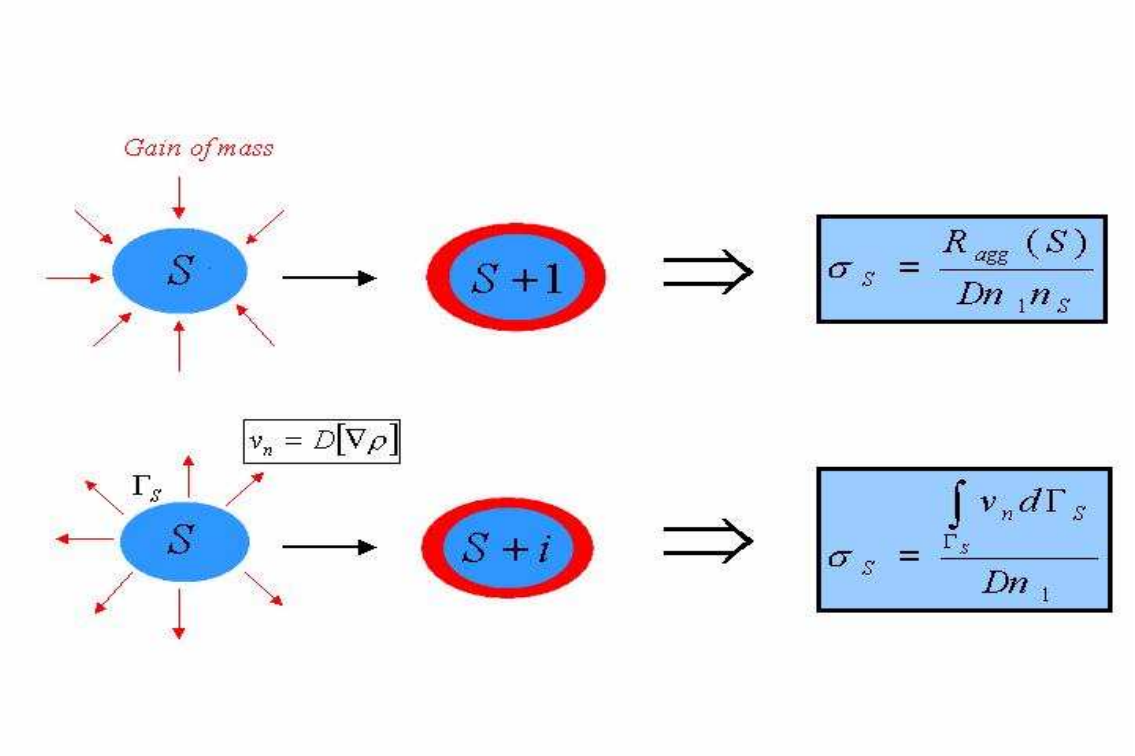

Figure 1.4: The rate of aggregation $R_{agg}(s)$ of an island of size s represents the gain of adatoms by that island per unit time. It is therefore given by the rate of change in area and can be expressed as the integral of the local normal velocity v_n around the island boundary.

island to grow in its own environment and do not use a simple model like the point-island model. So far, the size of any island can change continuously. In order to make comparisons with a discrete model, define a bin width $w = 1$ so that islands of size s are those islands with sizes $\tilde{s} \in [s \quad s + w)$. Then σ_s is the average of the $\sigma_{\tilde{s}}$. For the results shown below, we chose $w > 1$ to reduce

the noise in the data. All of our results represent averages over (at least) 50 simulations on a lattice with a lateral size of $L = 200$.

The results for the capture numbers are shown in Figures 1.5, 1.6 and 1.7, and suggest that the capture numbers have the functional form $\sigma_s = as + b$, that is, that they are affinely dependent on the island size. The slope $a(D/F, \theta)$ tends to a steady-state value $a(D/F)$ and attains this limit for a coverage $\theta \geq \theta_0(D/F)$, as shown in Fig. 1.9. The value of $\theta_0(D/F)$ is smaller for higher values of D/F, consistent with the scaling of the end of the nucleation phase and the beginning of the aggregation phase [18]. The value of the steady-state value $a(D/F)$ is smaller for higher D/F and behaves asymptotically like $a(D/F) = O\left[(D/F)^{-1/3}\right]$. The intercept b is, to first approximation, affinely dependent on the coverage as shown in Fig. 1.9, and is weakly dependent on D/F. We also observe scaling in coverage θ and in D/F for the capture numbers as a function of island size, scaled by their respective averages (see Fig. 1.8). If we write this scaling function as $\sigma_s/\sigma_{av} = C(s/s_{av})$, then we must have $C(1) = 1$. Indeed, we have

$$\begin{aligned}
C(s/s_{av}) &= \sigma_s/\sigma_{av} \\
\Leftrightarrow C(1) &= \sigma(s = s_{av})/\sigma_{av} \\
\Leftrightarrow C(1) &= \left(a\frac{\sum sn_s}{\sum n_s} + b\right) / \left(\frac{\sum (as+b)n_s}{\sum n_s}\right) \\
\Leftrightarrow C(1) &= \left(a\frac{\sum sn_s}{\sum n_s} + b\right) / \left(a\frac{\sum sn_s}{\sum n_s} + b\right) \\
\Leftrightarrow C(1) &= 1
\end{aligned}$$

which is consistent with our data (see Fig. 1.8).

The functional form $\sigma_s = as + b$ can be interpreted by considering the capture zones, whose boundaries are defined as the vertices of the diffusion field (See Fig. 1.10). On average, adatoms within a capture zone associated with an island will diffuse toward that island. Using this concept it was shown in [11] that at steady

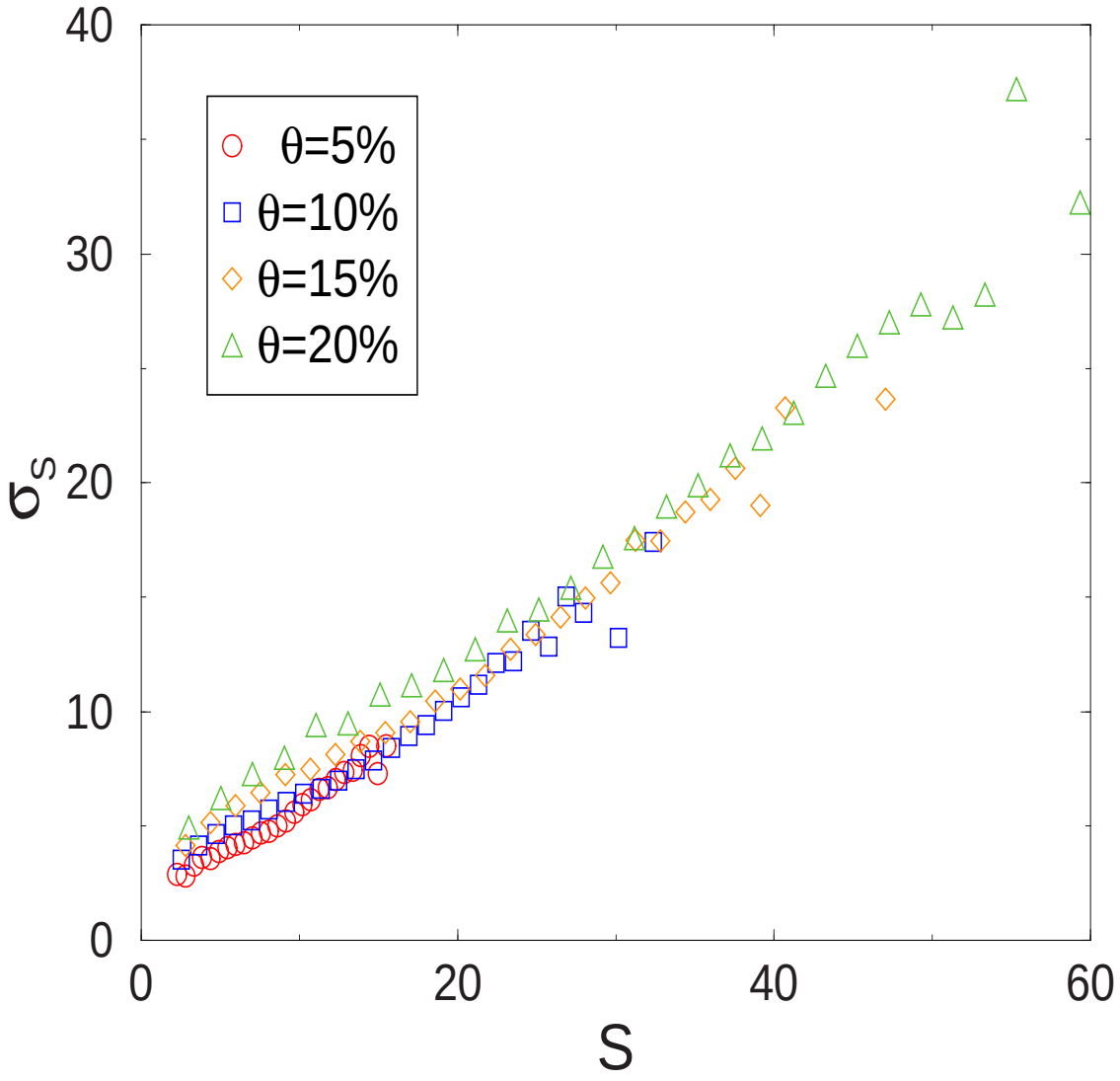

Figure 1.5: Capture numbers σ_s versus the islands sizes s for $D/F = 10^5$. Results shown, for four different values of the coverage θ.

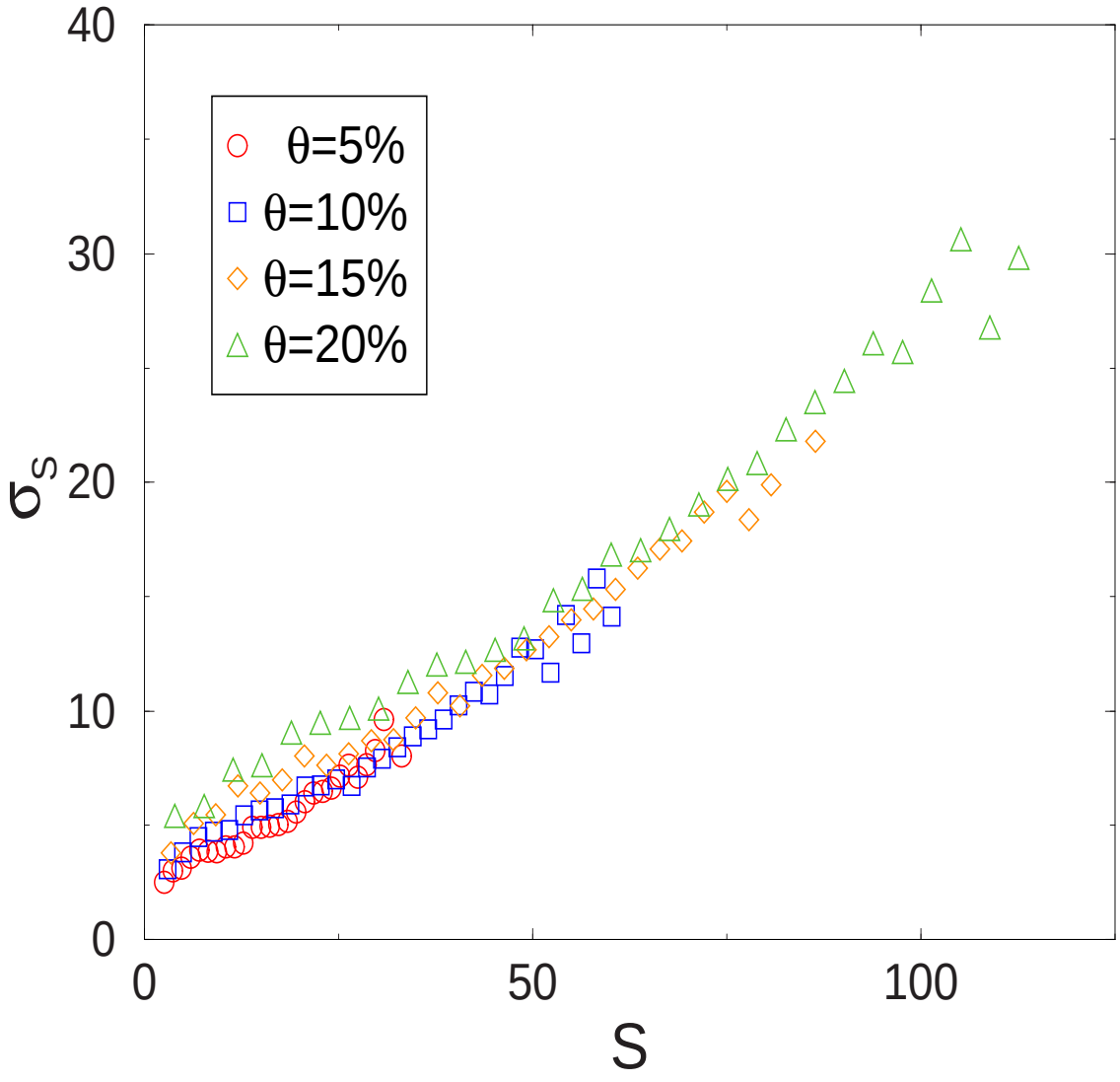

Figure 1.6: Capture numbers σ_s versus the islands sizes s for $D/F = 10^6$. Results shown, for four different values of the coverage θ.

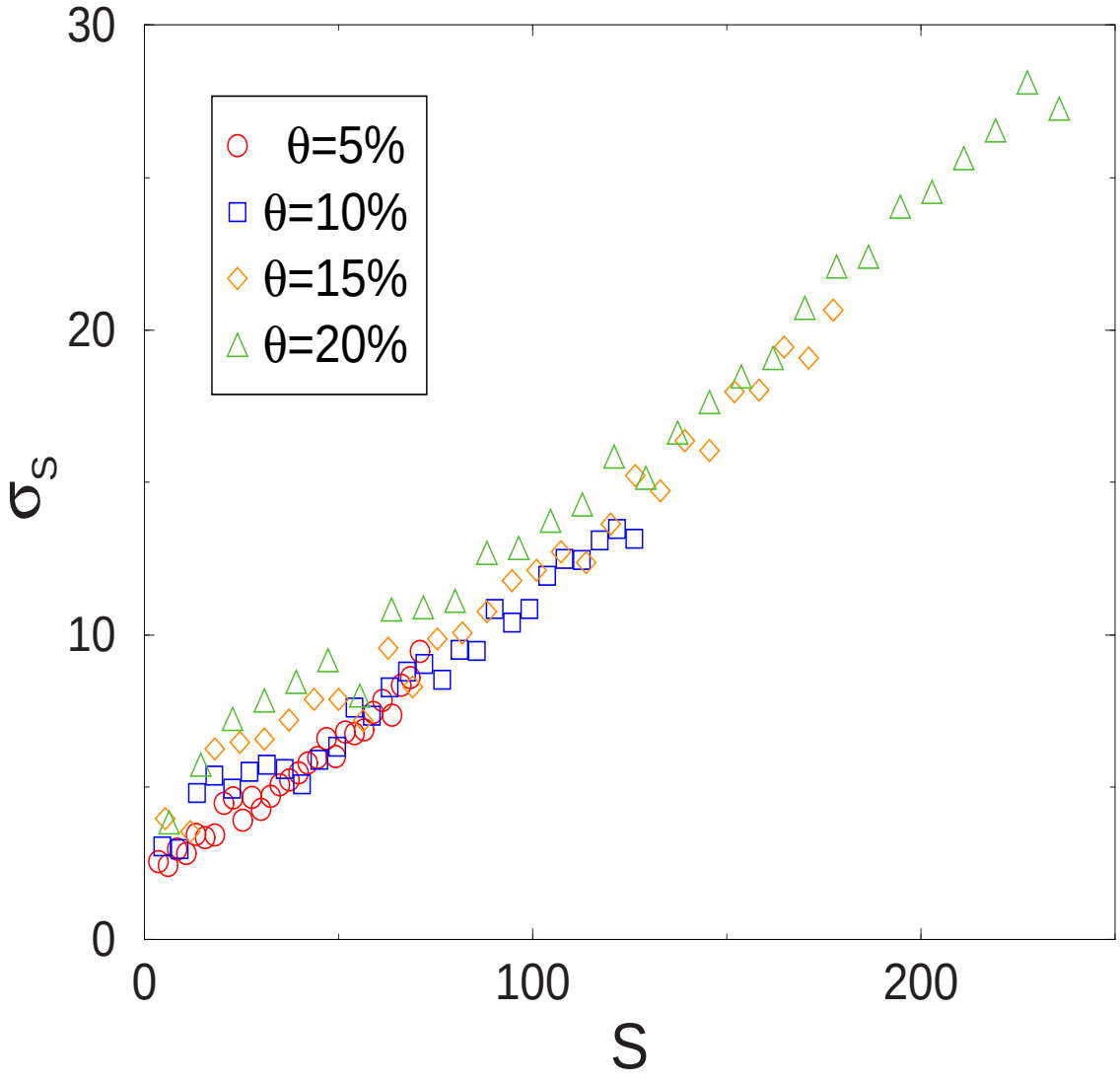

Figure 1.7: Capture numbers σ_s versus the islands sizes s for $D/F = 10^7$. Results shown, for four different values of the coverage θ.

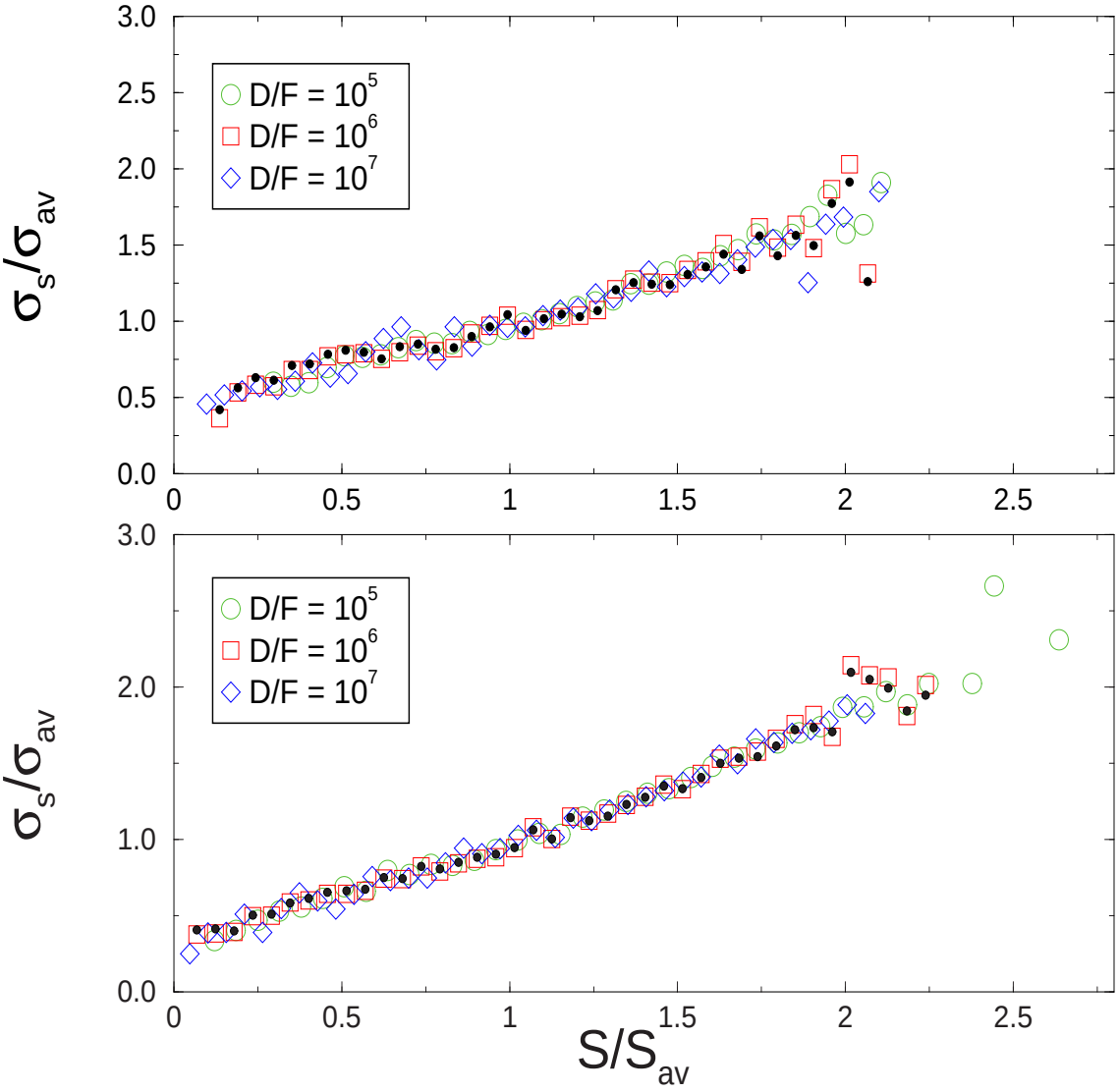

Figure 1.8: Capture numbers σ_s as computed by Eq. (1.7) (open symbols) and by Eq. (1.8) (solid symbols) versus the islands size, scaled by their respective average. Results shown are at 5% coverage (top) and 20% coverage (bottom), for three different values of D/F.

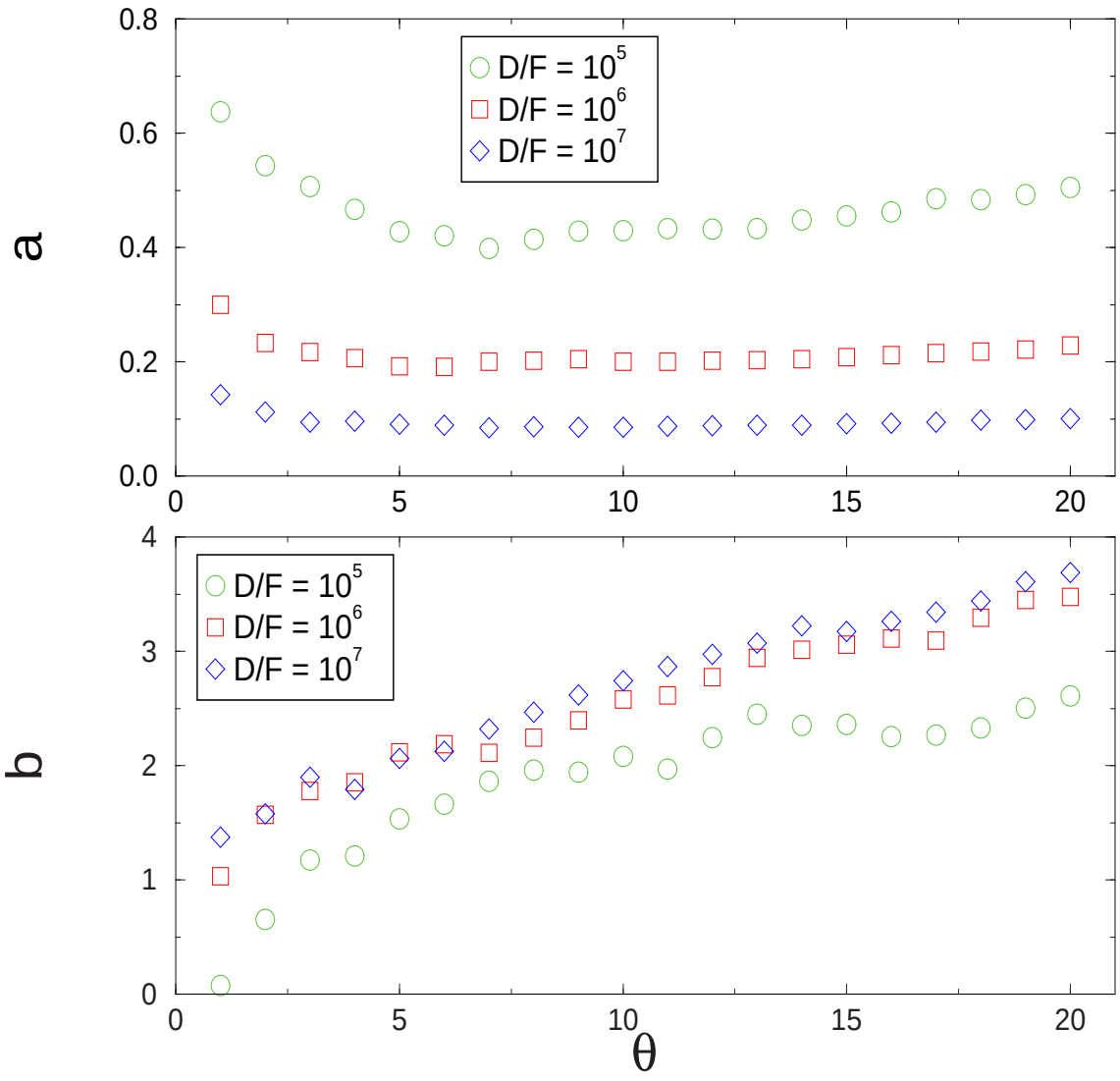

Figure 1.9: Coefficients a (top) and b (bottom) in $\sigma_S = as + b$, as a function of coverage θ, for different values of D/F.

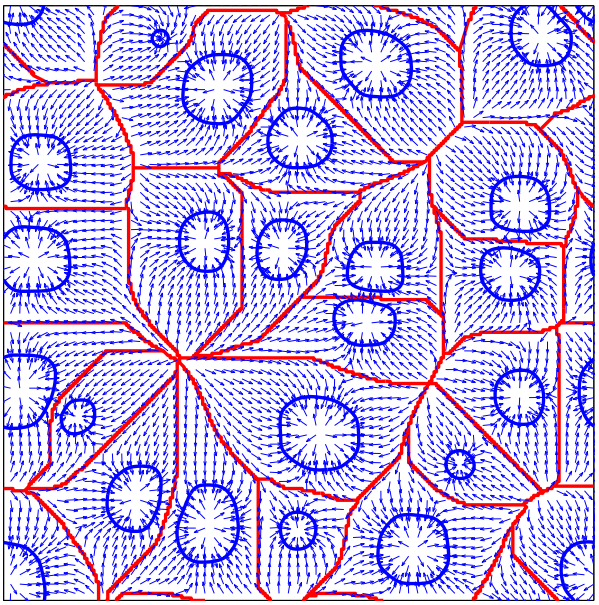

Figure 1.10: The blue arrows represent the gradient flow of the adatom density flowing towards the islands (in blue) associated with their capture zone (in red).

state

$$\sigma_s \approx \frac{F}{Dn_1} A_s, \tag{1.8}$$

where A_s is the average area of the capture zones of islands of size s. This implies that the capture number of an island is proportional to the area of its capture zone. Now, since the size of an island is itself proportional to the area of its capture zone, we would expect to have $\sigma_s = as + b$ (the intercept b corresponds to a point-island model). To check this, we also computed the areas of the capture zones and found excellent agreement between the capture numbers obtained using Eq. (1.7) and those obtained using Eq. (1.8), as shown in Fig. 1.8. We note that the relation between the Voronoi polygonal and the island size has also been studied by Mulheran and Blackman [19]. As an additional check to confirm our results, we also carried out a completely *self consistent* approach to calculating the capture numbers. Since the number of islands $N_s = L^2 n_s$ is increased every

time an island grows to the size s, and decreased every time an island of size s grows to a bigger size, we first rewrite the rate of change of n_s in the following conservative form:

$$\frac{dn_s}{dt} = \overline{J}^{\mathrm{IN}}(s) - \overline{J}^{\mathrm{OUT}}(s), \tag{1.9}$$

where $\overline{J}^{\mathrm{IN}}(s)$ is the average flux of islands entering the size interval $[s \quad s+1)$ and $\overline{J}^{\mathrm{OUT}}(s)$ the flux of islands leaving that interval. By introducing a counter $\triangle t L^2 \overline{J}(s)$ that is incremented by one every time an island grows to the size s or past that size, one can rewrite the rate equations as

$$\frac{dn_1}{dt} = F - 2\overline{J}(2) - \sum_{s>2} \overline{J}(s), \tag{1.10}$$

$$\frac{dn_s}{dt} = \overline{J}(s) - \overline{J}(s+1) \quad \text{for all } s > 1. \tag{1.11}$$

Comparison of Eqs. (1.5), (1.6) and (1.10), (1.11) gives the following expression for the *effective* capture numbers:

$$\sigma_s^{\mathrm{eff}} \stackrel{\mathrm{def}}{=} \frac{\overline{J}(s+1)}{D n_1 n_s}. \tag{1.12}$$

We have also used our *Island-Dynamics* model with probabilistic seeding style to compute the $\overline{J}(s)$. One has to be careful with how averaging is done. Indeed, the fluxes $J^{(i)}(s)$ can be computed from each simulation i and the densities $n_s^{(i)}$ can be calculated by solving

$$\frac{dn_1^{(i)}}{dt} = F - 2J^{(i)}(2) - \sum_{s>2} J^{(i)}(s), \tag{1.13}$$

$$\frac{dn_s^{(i)}}{dt} = J^{(i)}(s) - J^{(i)}(s+1) \quad \text{for all } s > 1. \tag{1.14}$$

However, it would be wrong to find $\sigma_s^{(i)}$ with a formula similar to (1.12) and define σ_s^{eff} as the average of the $\sigma_s^{(i)}$. Indeed, in this case, Eqs. (1.10) and (1.11)

wouldn't be consistent with Eqs. (1.5) and (1.6) as shown here:

$$\begin{aligned}\frac{dn_1}{dt} &= F - 2D\sigma_1^{eff} n_1 n_s - \sum_{s>2} D\sigma_s^{eff} n_1 n_s, \\ \frac{dn_s}{dt} &= D\sigma_{s-1}^{eff} n_1 n_s - D\sigma_s^{eff} n_1 n_s \quad \text{for all } s > 1.\end{aligned}$$

would be written

$$\begin{aligned}\frac{dn_1}{dt} &= F - 2D\left[\frac{\sum_i J^{(i)}(2)/Dn_1^{(i)}n_s^{(i)}}{\sum_i 1}\right] n_1 n_s - \sum_{s>2} D\left[\frac{\sum_i J^{(i)}(s+1)/Dn_1^{(i)}n_s^{(i)}}{\sum_i 1}\right] n_1 n_s, \\ \frac{dn_s}{dt} &= D\left[\frac{\sum_i J^{(i)}(s)/Dn_1^{(i)}n_s^{(i)}}{\sum_i 1}\right] n_1 n_s - D\left[\frac{\sum_i J^{(i)}(s+1)/Dn_1^{(i)}n_s^{(i)}}{\sum_i 1}\right] n_1 n_s \quad \text{for all } s > 1.\end{aligned}$$

which are not equivalent to (1.10) and (1.11).

Instead, one should first compute the average flux $\overline{J} = \sum_i J^{(i)} / \sum_i 1$ and then use Eq. (1.12) to define the (average) effective capture numbers used in mean-field rate equations. In this case Eqs. (1.10) and (1.11) are consistent with Eqs. (1.5) and (1.6) as shown here:

$$\begin{aligned}\frac{dn_1}{dt} &= F - 2\overline{J}(2) - \sum_{s>2} \overline{J}(s), \\ \frac{dn_s}{dt} &= \overline{J}(s) - \overline{J}(s+1) \quad \text{for all } s > 1.\end{aligned}$$

can be written as

$$\begin{aligned}\frac{dn_1}{dt} &= F - 2Dn_1 n_s \overline{J}(2)/Dn_1 n_s - \sum_{s>2} Dn_1 n_s \overline{J}(s+1)/Dn_1 n_s, \\ \frac{dn_s}{dt} &= Dn_1 n_s \overline{J}(s)/Dn_1 n_s - Dn_1 n_s \overline{J}(s+1)/Dn_1 n_s \quad \text{for all } s > 1.\end{aligned}$$

which, using Eq. (1.12), give

$$\begin{aligned}\frac{dn_1}{dt} &= F - 2D\sigma_1^{eff} n_1 n_s - \sum_{s>2} D\sigma_s^{eff} n_1 n_s, \\ \frac{dn_s}{dt} &= D\sigma_{s-1}^{eff} n_1 n_s - D\sigma_s^{eff} n_1 n_s \quad \text{for all } s > 1.\end{aligned}$$

In the simulations, we took a time step small enough to ensure that no island grows by more than one integer size. Using the measured values for $\overline{J}(s)$ we have integrated the set of rate equations described in Eqs. (1.10) and (1.11), which by construction is equivalent to Eqs. (1.5), (1.6) and (1.12), using a third-order explicit Runge-Kutta scheme with initial condition $n_s = 0$ for all s. The results for the total number density and the CSD are shown in Fig. 1.8 in comparison with level-set and KMC simulations. The agreement is excellent. Thus, we conclude that there exists a set of capture numbers that allows us to integrate mean-field rate equations to properly reproduce quantities such as the CSD that include spatial information.

Comparison of the extracted effective capture numbers σ_s^{eff} and the capture numbers σ_s previously described is shown in Fig. 1.14. The σ_s^{eff} are more noisy than the σ_s due to numerical difficulties [as defined the N_s are discrete and lead to jumps in the computed fluxes $\overline{J}(s)$, resulting in larger noise]. However, and more so for higher coverage ($\theta = 10\%, \theta = 15\%, \theta = 20\%$), the σ_s^{eff} exhibit the same functional form as the σ_s, that is, absence of a plateau for small islands. We have not yet been able to find an *analytic* form for the capture numbers as a function of coverage [$\sigma_s = \sigma_s(\theta)$] that could be used in the integration Eqs. (1.5)-(1.6). We speculate that the reason for this is that (i) small corrections to the affine dependence on s cannot be neglected; and (ii) the functional form for the capture numbers shown in Fig.1.8 might not be valid at a very early time since the nucleation process rearranges the capture zones (and therefore the capture numbers) at each seeding of an island. Moreover, it is meaningless to refer to a functional form when only two or three distinct sizes are present (see Fig. 1.11 where only islands of size between 2 and 3 are on the surface). (Binned, this would amount for islands of size 2 and a constant value for the associated capture number). Our results suggest that the time at which the affine dependence is a

fair approximation behaves asymptotically like $O\left[(D/F)^{-1/2}\right]$ as shown in Fig. 1.12 and we shall use this when deriving scaling laws for the adatom density and total number density.

The capture numbers presented here should be contrasted with the ones obtained in [11, 12, 13]. The main difference is the absence of the plateau for small islands in our results. We believe that this difference comes from the fact that we allow islands to grow in their environment when computing the capture numbers and therefore take into account all spatial correlations between islands. It is easy to see that a point-island model artificially increases the capture numbers for small islands, because it shifts the vertices of the capture zones in favor of small islands. Moreover, we show in the next section that the amount of nucleation in a point-island model is artificially big and accounts for a plateau. We cannot clearly identify the reason for the existence of a plateau for simulations with spatially extended islands. [12, 13] However, we speculate that the reason might be any of the following. (i) Annealing of small islands might be the source. During this process, small islands close to bigger islands (thus with small capture numbers) are absorbed by the bigger islands, increasing the average capture number for small islands and leading to the plateau. (ii) Experimental uncertainties, including processes that are outside of irreversible aggregation in the submonolayer regime could also offer a plausible explanation; or (iii) the effect of the finite time interval required for the approach in [11, 12, 13] is not completely clear.

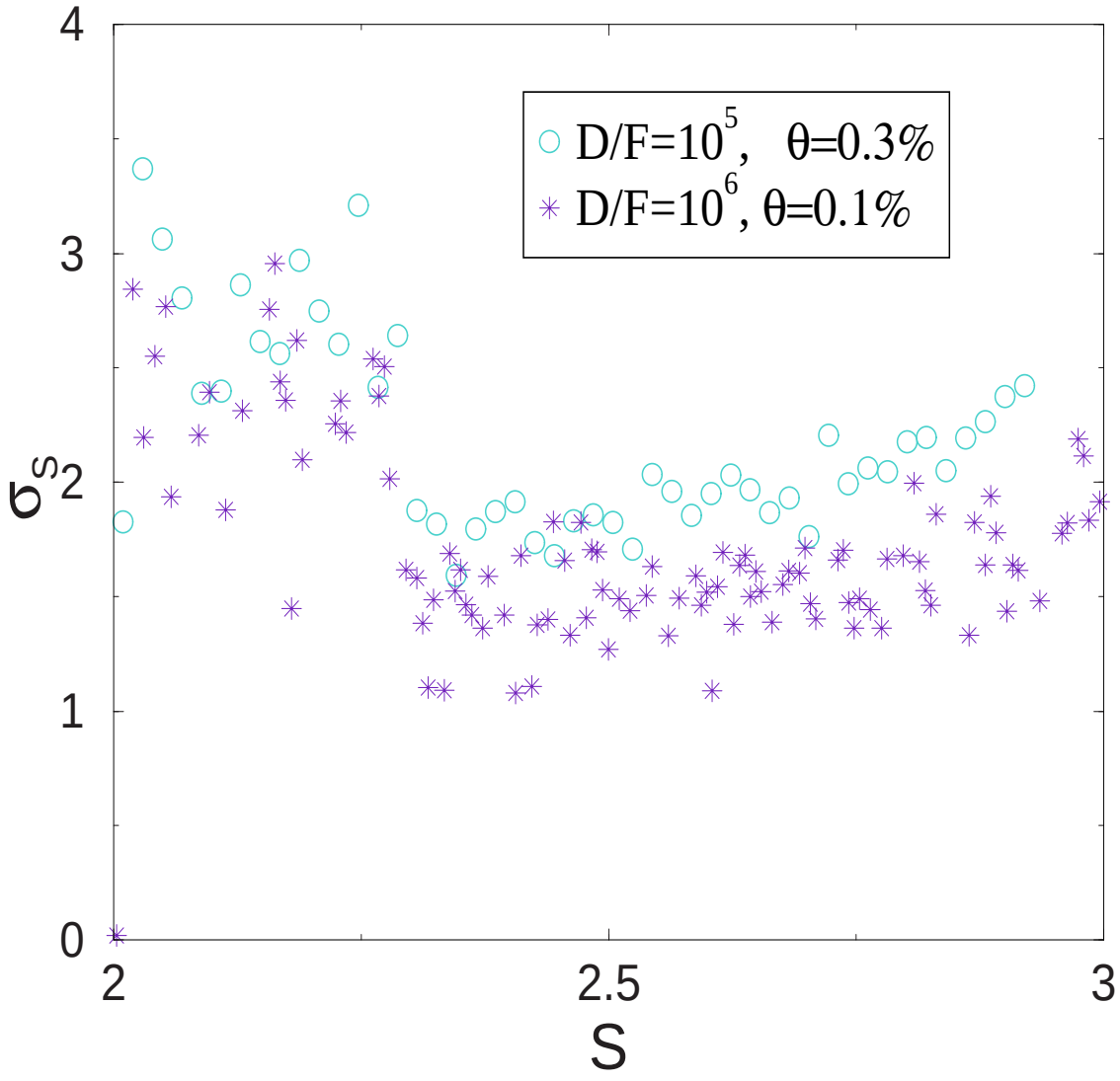

Figure 1.11: Capture numbers σ_s versus the islands sizes s for two different values of D/F. The corresponding coverage θ is chosen so that it corresponds to the nucleation phase. Binning this data would give an average capture number of magnitude slightly below 3, which is consistent with mean-field prediction.

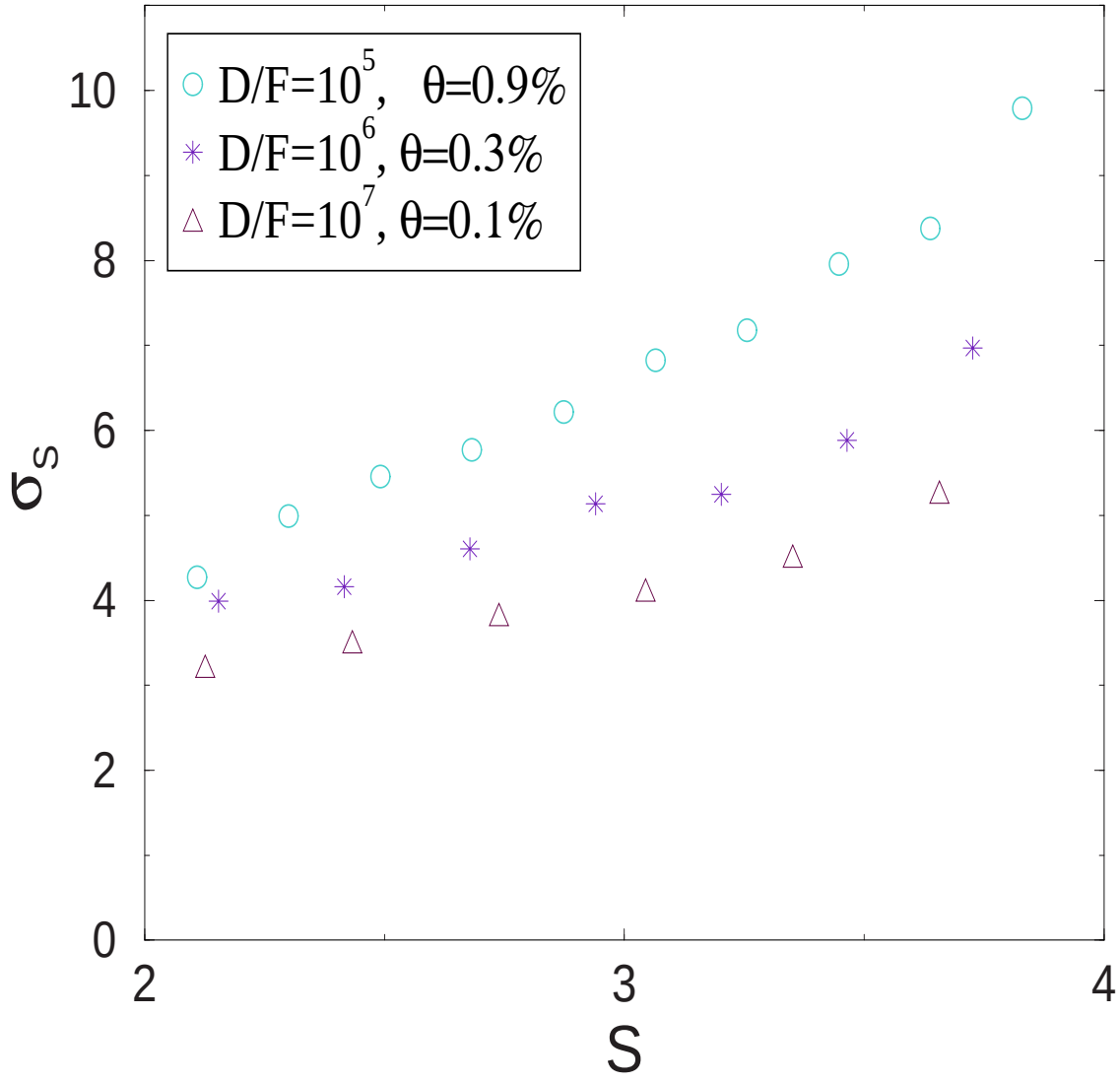

Figure 1.12: Capture numbers σ_s versus the islands sizes s for three different values of D/F. The corresponding coverage θ is chosen so that $\theta = O\left[(D/F)^{-1/2}\right]$.

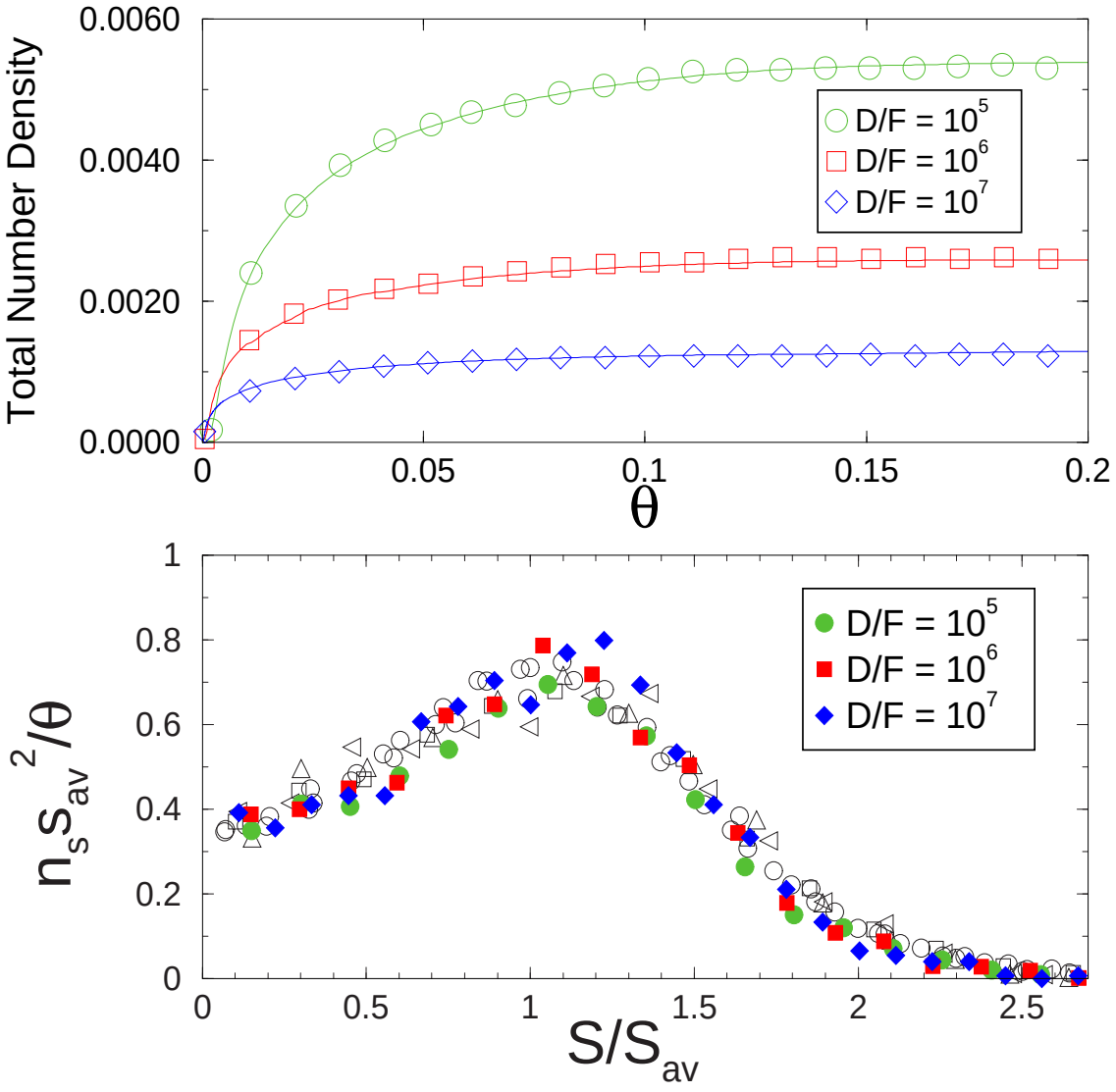

Figure 1.13: Top: total number density as a function of coverage for different values of D/F. Bottom: cluster size distribution at 20% coverage for different values of D/F. The solid continuous line (top) and the solid symbols (bottom) are the results of rate equations using the σ_s^{eff} and the open symbols are the results of level-set simulations.

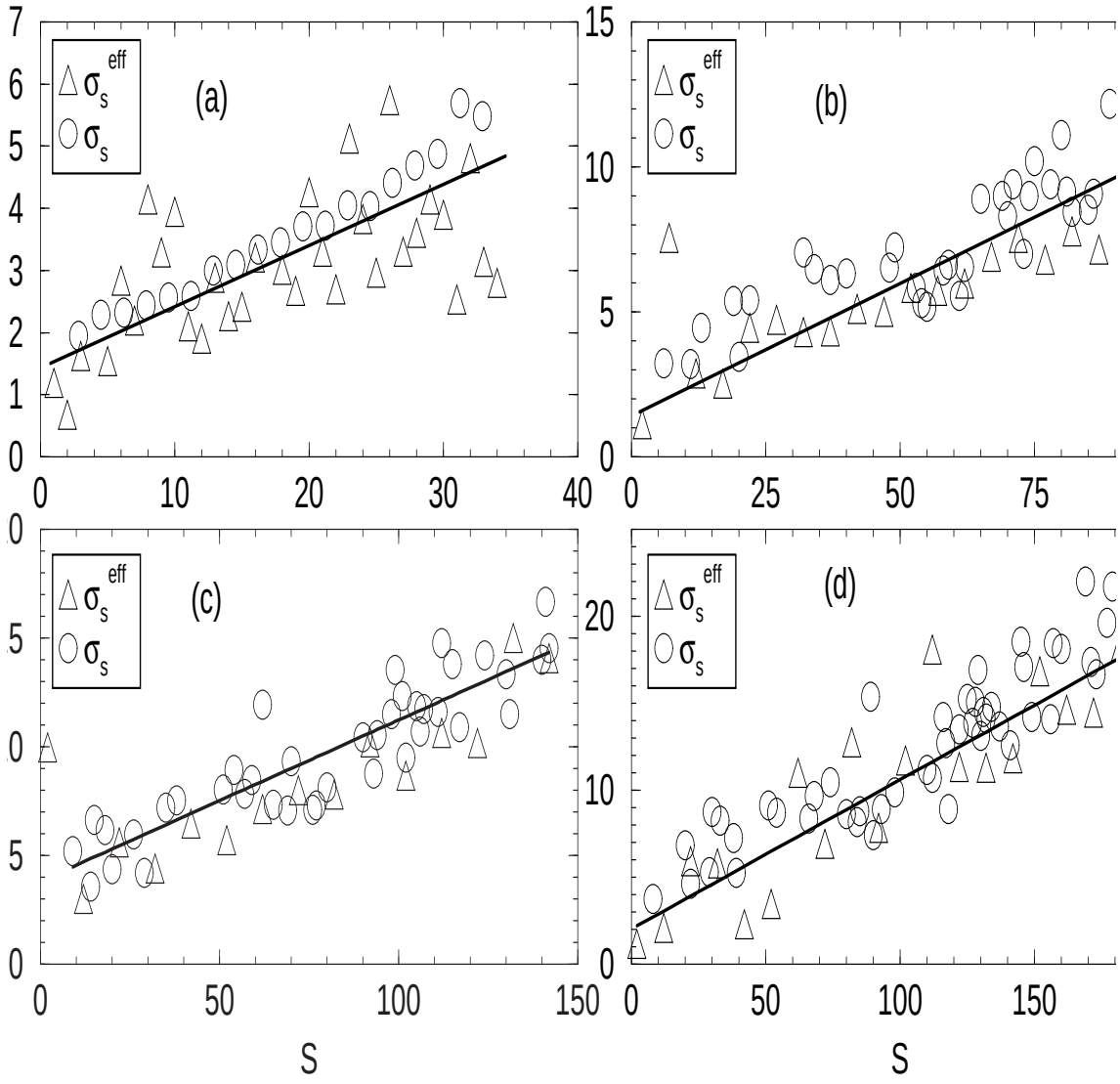

Figure 1.14: Capture numbers for $D/F = 10^7$ and different coverage $\theta = 2\%$ (a), $\theta = 10\%$ (b), $\theta = 15\%$ (c), $\theta = 20\%$ (d). The solid line is a guide to the eye for the σ_s^{eff}.

1.3 Analysis of Rate Equations

1.3.1 Analytical formula for the capture numbers at steady state

We can work out a derivation similar to that outlined in [11] to get a formula for the capture numbers given the cluster size distribution. The Rate Equations read:

$$\begin{aligned} \frac{dn_1}{dt} &= F - 2D\sigma_1 n_1^2 - Dn_1 \sum_{s>1} \sigma_s n_s \\ \frac{dn_s}{dt} &= R_{agg}(s-1) - R_{agg}(s) \quad \text{for all } s > 1 \quad , \end{aligned}$$

We want to exploit the scaling:

$$\begin{aligned} n_s &= \theta s_{av}^{-2} m(x) \\ \sigma_s &= \sigma_{av} C(x) \quad \text{, where } x = s/s_{av} \quad \text{(Quasi-hydrodynamics variable)} \end{aligned}$$

One can write the term dn_s/dt as

$$\begin{aligned} \frac{dn_s}{dt} &= \frac{d}{dt}(\theta s_{av}^{-2} m(x)) \\ &= \frac{d}{dt}(Fts_{av}^{-2} m(x)) \\ &= F\frac{d}{dt}(ts_{av}^{-2} m(x)) \\ &= F\left(s_{av}^{-2} m(x) + t\frac{d}{dt}(s_{av}^{-2} m(x))\right) \\ &= F\left(s_{av}^{-2} m(x) + t\frac{d}{dt}(s_{av}^{-2})m(x)) + ts_{av}^{-2}\frac{d}{dt}(m(x))\right) \end{aligned}$$

Now,

$$\begin{aligned} \frac{d}{dt}(s_{av}^{-2}) &= -2s_{av}^{-3}\frac{d}{dt}(s_{av}) \\ \frac{d}{dt}(m(x)) &= m'(x)\frac{dx}{dt} = sm'(x)\frac{d}{dt}(s_{av}^{-1}) = -sm'(x)s_{av}^{-2}\frac{d}{dt}(s_{av}) \end{aligned}$$

Thus,

$$\begin{aligned}
\frac{dn_s}{dt} &= F\left(s_{av}^{-2}m(x) - 2ts_{av}^{-3}\frac{d}{dt}(s_{av})m(x) - tss_{av}^{-2}s_{av}^{-2}m'(x)\frac{d}{dt}(s_{av})\right) \\
&= Fs_{av}^{-2}\left(m(x) - 2ts_{av}^{-1}\frac{d}{dt}(s_{av})m(x) - t\underbrace{ss_{av}^{-1}}_{x}s_{av}^{-1}\frac{d}{dt}(s_{av})m'(x)\right) \\
&= Fs_{av}^{-2}\left(m(x) - 2ts_{av}^{-1}\frac{d}{dt}(s_{av})m(x) - xts_{av}^{-1}\frac{d}{dt}(s_{av})m'(x)\right)
\end{aligned}$$

Now, if we let $\beta = ts_{av}^{-1}\frac{d}{dt}(s_{av})$ we have:

$$\frac{dn_s}{dt} = Fs_{av}^{-2}\left((1-2\beta)m(x) - x\beta m'(x)\right) \tag{1.15}$$

On the other hand one can write the rate of change of n_s as follows:

$$\begin{aligned}
\frac{dn_s}{dt} &= R_{agg}(s-1) - R_{agg}(s) \\
&= -\frac{R_{agg}(s-1) - R_{agg}(s)}{-1} \\
&\approx -\frac{d}{ds}(R_{agg}(s)) \\
&= -\frac{d}{ds}(D\rho\sigma_s n_s) \\
&= -D\rho\frac{d}{ds}(\sigma_{av}C(x)\theta s_{av}^{-2}m(x)) \\
&= -D\rho s_{av}^{-2}\theta\sigma_{av}\frac{d}{ds}(C(x)m(x)) \\
&= -D\rho s_{av}^{-2}\theta\sigma_{av}\frac{d}{dx}(C(x)m(x))\frac{dx}{ds} \\
&= -D\rho s_{av}^{-2}\theta\sigma_{av}\frac{d}{dx}(C(x)m(x))s_{av}^{-1} \\
&= -D\rho s_{av}^{-2}\sum sn_s\frac{\sum\sigma_s n_s}{\sum n_s}\frac{d}{dx}(C(x)m(x))(\frac{\sum sn_s}{\sum n_s})^{-1} \\
&= -D\rho s_{av}^{-2}\sum\sigma_s n_s\frac{d}{dx}(C(x)m(x))
\end{aligned}$$

Now, the adatom density satisfies:

$$\frac{dn_1}{dt} = F - 2D\sigma_1 n_1^2 - Dn_1\sum_{s>1}\sigma_s n_s$$

At steady state the average adatom density is almost constant in time and the nucleation term is negligible. Therefore one can take $dn_1/dt \approx 0$ and $\sigma_1 n_1^2 \approx 0$. It then follows that $D\rho \sum \sigma_s n_s = F$ and thus:

$$\frac{dn_s}{dt} = -Fs_{av}^{-2}\frac{d}{dx}(C(x)m(x)) \tag{1.16}$$

(1.15) and (1.16) give:

$$\begin{aligned} -Fs_{av}^{-2}\frac{d}{dx}(C(x)m(x)) &= Fs_{av}^{-2}\left((1-2\beta)m(x) - x\beta m'(x)\right) \\ \Leftrightarrow \frac{d}{dx}(C(x)m(x)) &= x\beta m'(x) - (1-2\beta)m(x) \end{aligned}$$

This implies that

$$\begin{aligned} \Leftrightarrow C(x) &= \frac{1}{m(x)}\left(\int_0^x \xi\beta m'(\xi) - (1-2\beta)m(\xi)d\xi + C(0)m(0)\right) \\ \Leftrightarrow C(x) &= \frac{1}{m(x)}\left(\int_0^x \beta\xi m'(\xi)d\xi - \int_0^x (1-2\beta)m(\xi)d\xi + C(0)m(0)\right) \\ \Leftrightarrow C(x) &= \frac{1}{m(x)}\left([\beta\xi m(\xi)]_0^x + (\beta-1)\int_0^x m(\xi)d\xi + C(0)m(0)\right) \quad \text{, by parts} \\ \Leftrightarrow C(x) &= \frac{1}{m(x)}\left(\beta x m(x) + (\beta-1)\int_0^x m(\xi)d\xi + C(0)m(0)\right) \end{aligned}$$

Hence a formula for the functional form of the capture numbers:

$$C(x) = \beta x + \frac{(\beta-1)\int_0^x m(\xi)d\xi + C(0)m(0)}{m(x)} \tag{1.17}$$

That is, the capture numbers are affine (with slope β and intercept $C(0)$) but with some correction terms.

Formula (1.17) shows that given the cluster size distribution and a value for β, there exists a unique functional form C for the capture numbers. However, one can also see that for a given cluster size distribution, there exists an infinite family (β, C_β) that can reproduce it via (1.17). It is instructive to comment on the value of β and the assumptions made in the derivation of (1.17). In that

derivation, we set $\beta = ts_{av}^{-1}\frac{d}{dt}(s_{av})$ and assumed that steady state is reached at 20% coverage and therefore that there is no nucleation of new islands. In our computations we obtained $\beta = 0.89$, whereas in the point-island model of [11], $\beta = 2/3$. If no new nucleation occurs at steady state and we let N be the total number of islands on the lattice (note that in this case N is independent of time), then one has

$$\begin{aligned}\frac{ds_{av}}{dt} &= \frac{d}{dt}\frac{\theta}{N} \\ &= \frac{d}{dt}\frac{Ft}{N} \\ &= \frac{F}{N}\end{aligned}$$

and therefore

$$\begin{aligned}\beta &\stackrel{\text{def}}{=} tS_{av}^{-1}\frac{ds_{av}}{dt} \\ &= t\frac{N}{Ft}\frac{F}{N} \\ &= 1\end{aligned}$$

Now, if nucleation occurs, then we have $N = N(t)$ and therefore

$$\begin{aligned}\frac{d}{dt}(s_{av}) &= F\frac{d}{dt}(t/N(t)) \\ &= F\frac{N(t) - t\frac{d}{dt}(N(t))}{N^2(t)} \\ &= \frac{F}{N(t)}\left(1 - t\frac{\frac{d}{dt}(N(t))}{N(t)}\right) \\ &< \frac{F}{N(t)} \quad \text{, since with nucleation } \tfrac{d}{dt}(N(t)) > 0\end{aligned}$$

so we have

$$\begin{aligned}\beta &\stackrel{\text{def}}{=} t\frac{N(t)}{Ft}\frac{ds_{av}}{dt} \\ &< t\frac{N(t)}{Ft}\frac{F}{N(t)} \\ &< 1\end{aligned}$$

Moreover, β gets smaller as $\frac{d}{dt}(N(t))$ gets larger, which is related to the amount of nucleation of new islands. It is well known that nucleation never reaches a steady state in a point-island model [14] and that explains the low value for β in [11]. Our value for β (0.93), shows that there is still some nucleation in our model at 20% coverage, but that this amount is small enough to keep the assumption valid. It is also interesting to use formula (1.17) with $\beta = 2/3$ to reproduce the results of [11] for the functional form of the capture numbers, as shown in Fig. 1.15.

1.3.2 Similarity solution of the CSD at steady state

We now comment on the consistency of our results

(1.15) and (1.16) give:

$$\begin{aligned}
-Fs_{av}^{-2}\frac{d}{dx}(C(x)m(x)) &= Fs_{av}^{-2}\left((1-2\beta)m(x) - x\beta m'(x)\right) \\
\Leftrightarrow \frac{d}{dx}(C(x)m(x)) &= x\beta m'(x) - (1-2\beta)m(x) \\
\Leftrightarrow C'(x)m(x) + C(x)m'(x) &= x\beta m'(x) - (1-2\beta)m(x) \\
\Leftrightarrow \frac{m'(x)}{m(x)} &= \frac{2\beta - 1 - C'(x)}{C(x) - x\beta} \\
\Leftrightarrow \int_0^x \frac{m'(\xi)}{m(\xi)}d\xi &= \int_0^x \frac{2\beta - 1 - C'(\xi)}{C(\xi) - \xi\beta}d\xi \\
\Leftrightarrow \ln\left(m(x)\right) &= \int_0^x \frac{2\beta - 1 - C'(\xi)}{C(\xi) - \xi\beta}d\xi + \ln\left(m(0)\right) \quad \text{, since } m(x) > 0
\end{aligned}$$

which gives:

$$m(x) = m(0)\exp\int_0^x \frac{2\beta - 1 - C'(\xi)}{C(\xi) - \xi\beta}d\xi \tag{1.18}$$

Using this formula (that first appeared in [11]) we obtain the CSD from interpolation of the capture numbers obtained from our model (see Fig. 1.16).

1.3.3 Scaling laws

Some understanding can be gained by deriving scaling laws for the θ-dependence and D/F-dependence of the adatom density ρ and the total number density of islands $N = \sum_{s>1} n_s$. Standard analysis assumes a point-island model for which $\sigma_s = \sigma_1$ for all s throughout the nucleation and growth phases. Our results suggest that this assumption is valid in the nucleation phase (see Fig. 1.11), but not in the growth phase, for which an affine dependence on the islands sizes is a better approximation (see Figs. 1.5, 1.6, 1.7 and 1.12). We present here the scaling laws that one obtains using a point-island model and the affine-dependence model and what the predominant processes are in each phase.

- The point-island model assumes that $\sigma_s = \sigma_1$ for all s. In that case the equations for the adatom density and the total number density of islands simplify to

$$\begin{aligned} \frac{d\rho}{dt} &= F - 2D\sigma_1\rho^2 - D\rho\sum_{s>1}\sigma_1 n_s \\ \frac{dN}{dt} &= \sum_{s>1} D\rho(\sigma_{s-1}n_{s-1} - \sigma_s n_s) \\ &= \sum_{s>1} D\sigma_1\rho(n_{s-1} - n_s) \\ &= D\sigma_1\rho^2 \end{aligned}$$

  For the purpose of deriving scaling laws one can take without loss of generality $\sigma_1 = 1$ and, since $\theta = Ft$, we have

$$\frac{d\rho}{d\theta} = 1 - 2R\rho^2 - R\rho N \quad \text{,where } R = D/F \tag{1.19}$$

$$\frac{dN}{d\theta} = R\rho^2 \tag{1.20}$$

For the R scaling we look for the constant α and β such that, for $R >> 1$

$$\begin{aligned} N &= R^{\alpha}\tilde{N}(\theta) \\ \rho &= R^{\beta}\tilde{\rho}(\theta) \end{aligned}$$

With these assumptions, equations (1.19) and (1.20) give

$$R^{\beta}\frac{d}{d\theta}\tilde{\rho} = R^{0} - 2R^{1+2\beta}\tilde{\rho}^{2} - R^{1+\alpha+\beta}\tilde{\rho}\tilde{N} \tag{1.21}$$

$$R^{\alpha}\frac{d}{d\theta}\tilde{N} = R^{1+2\beta}\tilde{\rho}^{2} \tag{1.22}$$

Matching the powers of R one must have $\alpha = 1 + 2\beta$ and the method of dominant balance gives $1 + \alpha + \beta = 0$. Together, these two conditions give the scaling laws in R:

$$\begin{aligned} \rho &= O\left(R^{-2/3}\right) \\ N &= O\left(R^{-1/3}\right) \end{aligned}$$

This scaling for $R >> 1$, along with (1.21) and (1.22) implies that

$$\begin{aligned} 1 - \tilde{\rho}\tilde{N} &= 0 \\ \frac{d\tilde{N}}{d\theta} &= \tilde{\rho}^{2} \end{aligned}$$

Thus,

$$\begin{aligned} \tilde{\rho} &= 1/\tilde{N} \\ \Leftrightarrow \frac{d\tilde{N}}{d\theta} &= 1/\tilde{N}^{2} \\ \Leftrightarrow \tilde{N} &= 3^{1/3}(\theta + \overline{\theta})^{1/3} \end{aligned}$$

and

$$\tilde{\rho} = 3^{-1/3}(\theta + \overline{\theta})^{-1/3}$$

Therefore one has

$$\begin{aligned} \rho &= 3^{-1/3} R^{-2/3} (\theta + \overline{\theta})^{-1/3} \\ N &= 3^{1/3} R^{-1/3} (\theta + \overline{\theta})^{1/3} \end{aligned}$$

These scaling laws are not valid for early time if one starts the growth with a clean surface ($\rho = 0$ and $N = 0$) since in this case the equation for the adatom density reduces to $\rho_t = F$ and therefore the adatom density must initially increase in coverage. Rescaling coverage as $\theta = R^{-\gamma}\tilde{\theta}$ which implies $\frac{\partial}{\partial\theta} = R^{\gamma}\frac{\partial}{\partial\tilde{\theta}}$ and writing

$$N = R^{\alpha}\tilde{N}(\tilde{\theta})$$
$$\rho = R^{\beta}\tilde{\rho}(\tilde{\theta})$$

equation (1.19) and (scaling2) give

$$R^{\beta+\gamma}\frac{d}{d\tilde{\theta}}\tilde{\rho} = R^{0} - 2R^{1+2\beta}\tilde{\rho}^{2} - R^{1+\alpha+\beta}\tilde{\rho}\tilde{N} \tag{1.23}$$

$$R^{\alpha+\gamma}\frac{d}{d\tilde{\theta}}\tilde{N} = R^{1+2\beta}\tilde{\rho}^{2} \tag{1.24}$$

Matching the powers of R one must have $\alpha + \gamma = 1 + 2\beta$ and the method of dominant balance gives $1 + \alpha + \beta = 1 + 2\beta = \beta + \gamma = 0$. Together, these two conditions give the scaling laws in R:

$$\begin{aligned} \tilde{\theta} &= R^{-1/2} \\ \rho &= R^{-1/2}\tilde{\rho}(\tilde{\theta}) \\ N &= R^{-1/2}\tilde{N}(\tilde{\theta}) \end{aligned}$$

Our results suggest that this scaling is valid in the nucleation phase, but not in the growth regime. Rather, we take the functional form for the capture numbers to be $\sigma_s = as + b$.

- The affine-dependence model assumes that $\sigma_s = as + b$. Our data suggest that $a = \tilde{a}R^{-1/3}$ and $b = \tilde{b}\theta$. In that case the equations for the adatom density and the total number density of islands simplify to

$$\begin{aligned}
\frac{d\rho}{dt} &= 1 - 2R\sigma_1\rho^2 - R\rho\sum_{s>1}(as+b)n_s \\
&= 1 - 2R\sigma_1\rho^2 - R\rho\left(\sum_{s>1} asn_s + \sum_{s>1} bn_s\right) \\
&= 1 - 2R\sigma_1\rho^2 - R\rho\left(a\theta + bN\right) \\
&= 1 - 2R\sigma_1\rho^2 - R\rho\left(\tilde{a}R^{-1/3}\theta + \tilde{b}\theta N\right) \\
\frac{dN}{dt} &= \sum_{s>1} R\rho(\sigma_{s-1}n_{s-1} - \sigma_s n_s) \\
&= R\sigma_1\rho^2
\end{aligned}$$

Now, $\sigma_1 = a + b = \tilde{a}R^{-1/3} + \tilde{b}\theta$, which reduces to $\sigma_1 = \tilde{b}\theta$ in the case where $R >> 1$. Therefore we have the following two equations

$$\frac{d\rho}{dt} = 1 - 2R\sigma_1\rho^2 - R\rho\left(\tilde{a}R^{-1/3}\theta + \tilde{b}\theta N\right) \tag{1.25}$$

$$\frac{dN}{dt} = R\tilde{b}\theta\rho^2 \tag{1.26}$$

For the R scaling we look for the constant α and β such that, for $R >> 1$

$$\begin{aligned}
N &= R^{\alpha}\tilde{N}(\theta) \\
\rho &= R^{\beta}\tilde{\rho}(\theta)
\end{aligned}$$

With these assumptions, equations (1.25) and (1.26) give

$$R^{\beta}\frac{d}{d\theta}\tilde{\rho} = R^0 - 2\sigma_1 R^{1+2\beta}\tilde{\rho}^2 - R^{1+\beta}\tilde{\rho}\left(\tilde{a}R^{-1/3}\theta + \tilde{b}\theta R^{\alpha}\tilde{N}\right) \tag{1.27}$$

$$R^{\alpha}\frac{d}{d\theta}\tilde{N} = R^{1+2\beta}\tilde{\rho}^2 \tag{1.28}$$

Matching the powers of R one must have $\alpha = 1 + 2\beta$ and the method of dominant balance gives $1 + \alpha + \beta = 0$. Together, these two conditions give

the scaling laws in R:

$$\begin{aligned} \rho &= O\left(R^{-2/3}\right) \\ N &= O\left(R^{-1/3}\right) \end{aligned}$$

This scaling for $R >> 1$, along with (1.27) and (1.28) implies that

$$\begin{aligned} 1 - \tilde{\rho}\theta\left(\tilde{a} + \tilde{b}\tilde{N}\right) &= 0 \\ \frac{d\tilde{N}}{d\theta} &= \tilde{b}\theta\tilde{\rho}^2 \end{aligned}$$

Thus,

$$\begin{aligned} \tilde{\rho} &= \frac{1}{\theta\left(\tilde{a} + \tilde{b}\tilde{N}\right)} \\ \Leftrightarrow \frac{d\tilde{N}}{d\theta} &= \tilde{b}\theta^{-1}\left(\tilde{a} + \tilde{b}\tilde{N}\right)^{-2} \\ \Leftrightarrow \tilde{N} &= \tilde{b}^{-1}\left(\left[3\tilde{b}^2 \ln\left(\frac{\theta}{\tilde{\theta}}\right)\right]^{1/3} - \tilde{a}\right) \quad \text{, where } \tilde{\theta} \in \mathbb{R} \end{aligned}$$

and

$$\tilde{\rho} = \theta^{-1}\left[3\tilde{b}^2 \ln\left(\frac{\theta}{\tilde{\theta}}\right)\right]^{-1/3}$$

Therefore one has

$$\begin{aligned} \rho &= \theta^{-1}\left[3\tilde{b}^2 \ln\left(\frac{\theta}{\tilde{\theta}}\right)\right]^{-1/3} R^{-2/3} \\ N &= \tilde{b}^{-1}\left(\left[3\tilde{b}^2 \ln\left(\frac{\theta}{\tilde{\theta}}\right)\right]^{1/3} - \tilde{a}\right) R^{-1/3} \quad \text{, where } \tilde{\theta} \in \mathbb{R} \end{aligned}$$

Therefore, N is increasing in time, ρ is decreasing in time while both ρ and N decrease as R increases. Moreover, equations (1.27) and (1.28) show that with this scaling the adatom density is balanced by the deposition flux and the attachment to islands (dominant terms), and the total number density is the product of nucleations. The R-dependence is the same as that obtained with a Point-Island model, but the evolution in time is different.

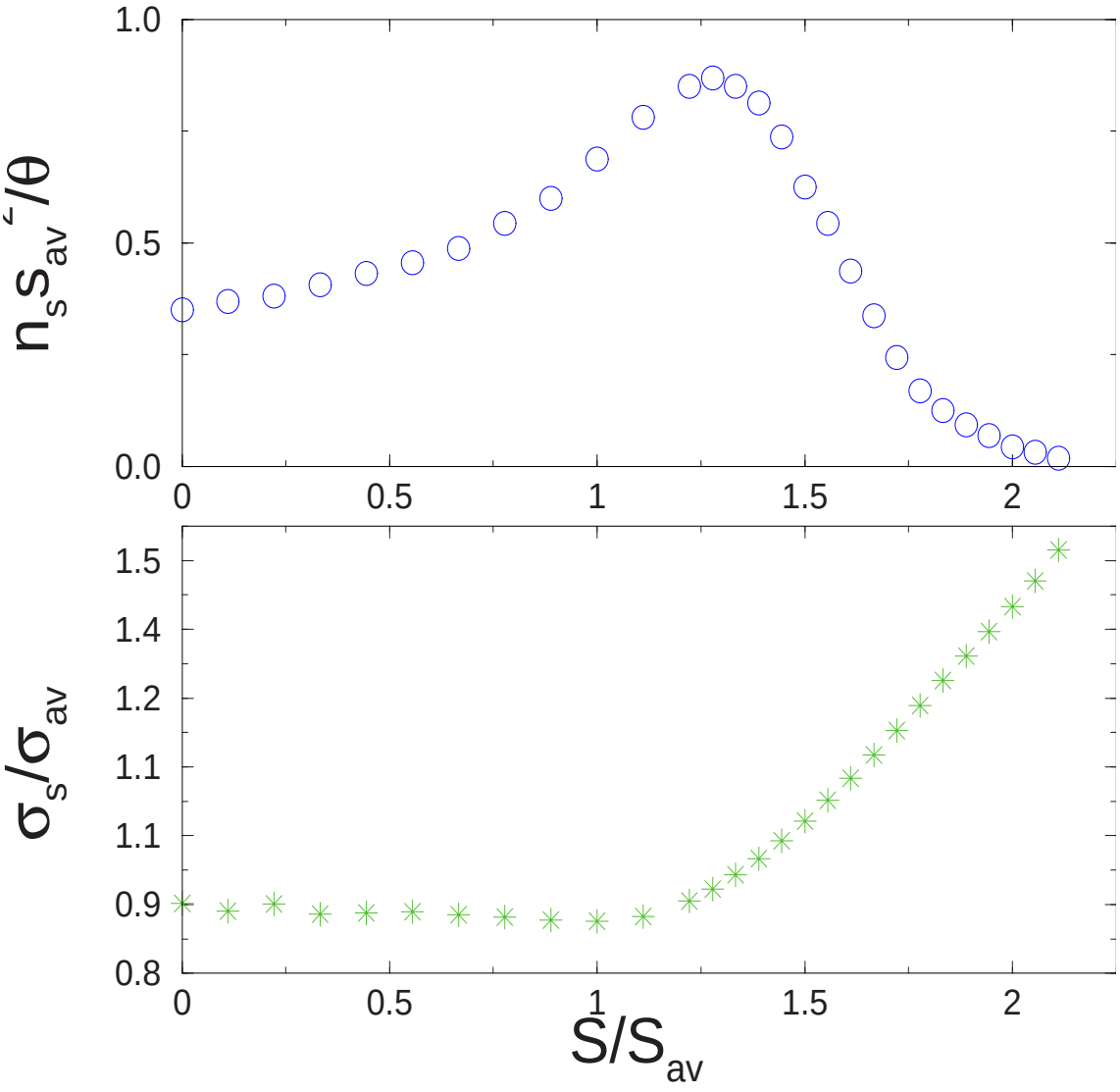

Figure 1.15: Data approximating the CSD obtained by Bartelt and Evans using their Point-Island model (top). Capture numbers obtained using formula (1.17) with $\beta = 2/3$ exhibiting the plateau found in [11] (bottom).

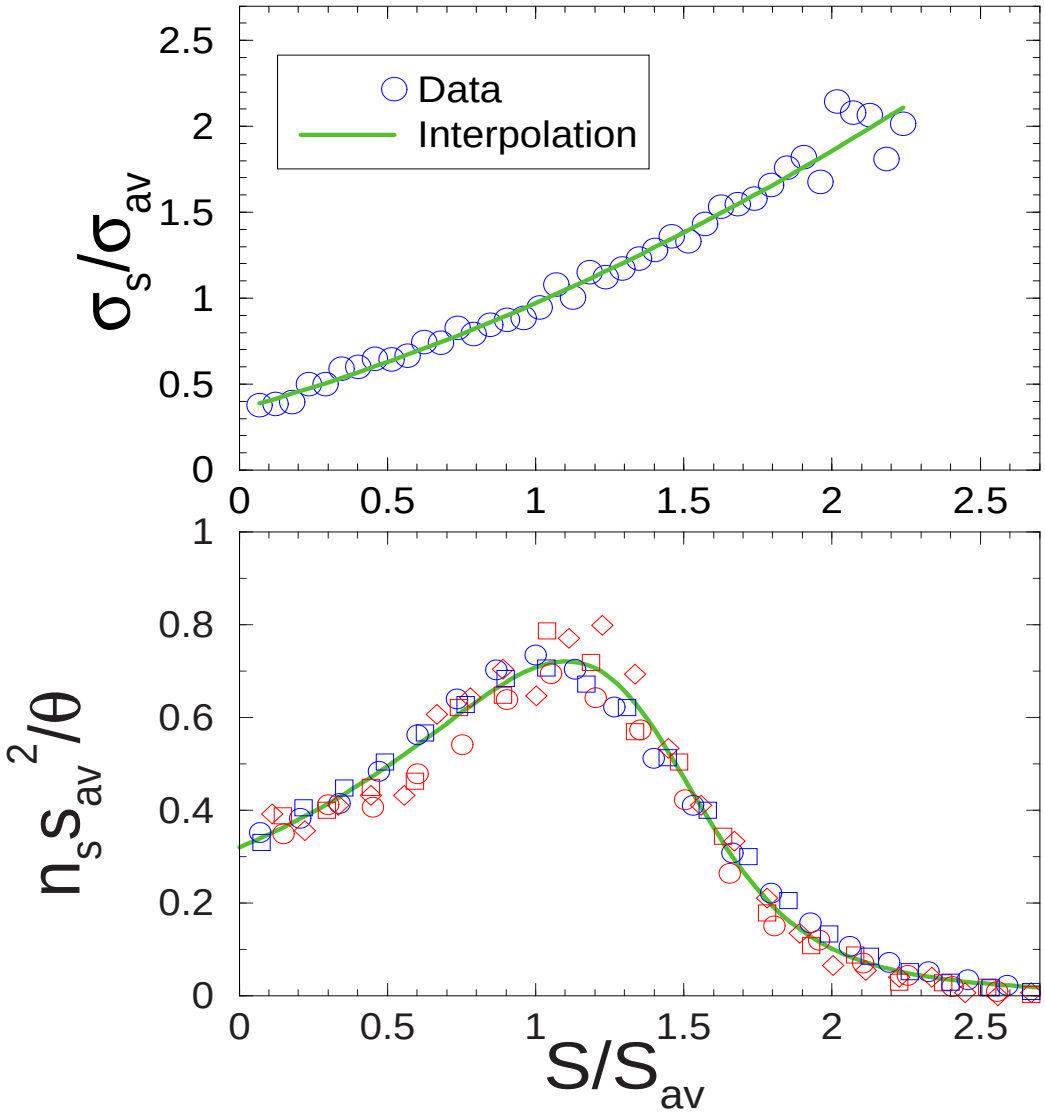

Figure 1.16: Top: Quadratic interpolation (solid green line) of the capture numbers versus the islands sizes scaled by their respective average (blue symbols). Bottom: The green solid line is the result of formula (1.18) applied with the quadratic interpolation of the capture numbers on top of level-set (red symbols) and KMC (blue symbols) simulations for different value of D/F.

1.4 Conclusion

In summary, we have shown that the capture numbers that include the effect of all spatial fluctuations are, to first approximation, affinely dependent on the island size, and that they are nearly time independent after the islands are seeded. In particular, there is no plateau as found in previous works. We have confirmed this result by computing the capture numbers self-consistently and have explained the difference with the results obtained using a point-island model by linking the functional form of the capture numbers to the amount of nucleation. Further, we have derived scaling laws for the adatom density and total number density using the new form of the σ_s.

CHAPTER 2

A Symmetric Method for Implicit Time Discretization of the Stefan Problem - Application to multiphase flows

2.1 Introduction

The text of this part of this manuscript is a preprint of what will appear in the Journal of Computational Physics and is co-authored by Cheng L. T., Fedkiw R. P. and Kang M.

In [20] ideas based on the Ghost Fluid Method [21] were used to develop a first order accurate discretization of the variable coefficient Poisson equation in the presence of an irregular interface across which the variable coefficients, the solution and the derivatives of the solution may have jumps. This new numerical method was applied to two phase incompressible flow in [22] and to incompressible flame discontinuities in [23]. In this chapter, we consider a similar Poisson equation where Dirichlet boundary conditions (instead of jump conditions) are imposed on the irregular interface. In this case, the solution is not coupled across the interface and we are able to design a second order accurate discretization as opposed to the first order accurate discretization proposed in [20] for the jump condition case. Both the discretization proposed here and that in [20] yield symmetric matrices which can readily be inverted with a number of fast methods (e.g. Preconditioned Conjugate Gradient (PCG) method (see [24])).

This new second order accurate discretization of the Poisson equation is applied to solving the Stefan problem. We use a level-set formulation [25] to evolve the interface location and a finite difference discretization of the heat equation on a Cartesian grid to solve for the temperature. In order to avoid the stringent $O(\triangle x^2)$, or even worse $O(\theta^2 \triangle x^2)$ with $0 < \theta \leq 1$ for cells cut by the interface, time step restriction imposed by explicit time discretization of the heat equation, we use implicit discretization in time. This implicit time discretization requires a matrix inversion that can be rather time consuming, especially if the embedded interface forces a nonsymmetric discretization of the spatial terms. This was the

case in [26] where the nonsymmetric matrix was inverted with the (very slow) Gauss-Seidel method (see [24]). In this chapter, we propose an alternative symmetric discretization to this problem that allows a (relatively fast) Preconditioned Conjugate Gradient method to be used for the matrix inversion. We note that if the interface location is not known exactly, our scheme will fail to be second order accurate.

The earliest level-set method for solidification type problems was presented in [27] where the authors recast the equations of motion into a boundary integral equation and used the level-set method to update the location of the interface. In [26] the boundary integral equations were avoided by using a finite difference method to solve the heat equation on a Cartesian grid with Dirichlet boundary conditions imposed on the interface. The jump in the first derivatives of the temperature was used to compute an interface velocity which was extended to a band about the interface and used to evolve a level-set function in time. The velocity calculation in [26] is rather awkward and both the standard grid and a 45^o rotated grid are used to aid in the removal of nonphysical grid anisotropy effects. This velocity computation was improved upon in [28] where the authors show good agreement between the level-set method and phase field methods for the case where the thermal conductivities are the same in both materials. In addition, [28] showed that the level-set method continues to perform well for the case where the thermal conductivities are different in the two materials. For more details on phase field methods for the Stefan problem, see [28] and the references therein.

In [29], the authors discretized the heat equation on a Cartesian grid in a manner very similar to that proposed in [26] resulting in a nonsymmetric matrix when applying an implicit time discretization. [29] used front tracking to update

the location of the interface improving upon the front tracking approach proposed in [30] which used the smeared out immersed boundary method from [31] and explicit time stepping.

In [32], the authors solved a variable coefficient Poisson equation in the presence of an irregular interface where Dirichlet boundary conditions were imposed. They used a finite volume method that results in a nonsymmetric discretization matrix. Both multigrid methods and adaptive mesh refinement were used in [32], and in [33] this nonsymmetric finite volume discretization was coupled to a volume of fluid front tracking method in order to solve the Stefan problem.

The interested reader is referred to [30], [26] and the references therein for an extensive summary of computational results for the Stefan problem. Most notably, [34] uses adaptive finite element methods for both the heat equation and for the interface evolution producing spectacular (and rare) three dimensional results.

2.2 Equations

2.2.1 Poisson Equation

Consider a Cartesian computational domain, Ω, with exterior boundary, $\partial\Omega$, and a lower dimensional interface, Γ, that divides the computational domain into disjoint pieces, Ω^- and Ω^+. The variable coefficient Poisson equation is given by

$$\nabla \cdot (\beta(\vec{x})\nabla u(\vec{x})) = f(\vec{x}), \qquad \vec{x} \in \Omega \tag{2.1}$$

where $\vec{x} = (x, y, z)$ are the spatial dimensions, $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ is the divergence operator, and $\beta(\vec{x})$ is assumed continuous on each disjoint sub-domain, Ω^- and Ω^+, but may be discontinuous across the interface Γ. Furthermore, $\beta(\vec{x})$ is assumed to be positive and bounded below by some $\epsilon > 0$. On $\partial\Omega$, either Dirichlet boundary conditions of $u(\vec{x}) = g(\vec{x})$ or Neumann boundary conditions of $u_n(\vec{x}) = h(\vec{x})$ are specified. Note that $u_n = \nabla u \cdot \vec{N}$ is the normal derivative of u with normal $\vec{N}$.

In [20], equation (2.1) was solved with a first order numerical method when the jump conditions, $[u]_\Gamma = a(\vec{x})$ and $[\beta u_n]_\Gamma = b(\vec{x})$ were specified across the interface. If instead, a Dirichlet boundary condition of $u_\Gamma = c(\vec{x})$ is specified on the interface, then equation (2.1) decouples into two separate equations, one on Ω^- and one on Ω^+. Therefore any jumps of u, βu_n or β across the interface can be ignored allowing equation (2.1) to be considered separately and independently on Ω^- and on Ω^+.

2.2.2 Heat Equation

Our primary interest is in coupling the Stefan problem to multiphase flow simulations, e.g. see [23] where the Stefan problem solution can be used to calculate

the flame speed. Therefore, we start with conservation of mass, momentum and energy,

$$\rho_t + \nabla \cdot \left(\rho \vec{V}\right) = 0 \tag{2.2}$$

$$(\rho v_1)_t + \nabla \cdot \left(\rho v_1 \vec{V}\right) + p_x = 0 \tag{2.3}$$

$$(\rho v_2)_t + \nabla \cdot \left(\rho v_2 \vec{V}\right) + p_y = 0 \tag{2.4}$$

$$(\rho v_3)_t + \nabla \cdot \left(\rho v_3 \vec{V}\right) + p_z = 0 \tag{2.5}$$

$$E_t + \nabla \cdot \left((E+p)\,\vec{V}\right) = \nabla \cdot (k \nabla T) \tag{2.6}$$

where ρ is the density, $\vec{V} =< v_1, v_2, v_3 >$ are the velocities, p is the pressure, $E = \rho e + \frac{\rho \vec{V} \cdot \vec{V}}{2}$ is the total energy per unit volume, e is the internal energy per unit mass, T is the temperature and k is the thermal conductivity. Note that viscosity is ignored.

Equations (2.2) through (2.6) can be manipulated to write

$$\rho e_t + \rho \vec{V} \cdot \nabla e + p \nabla \cdot \vec{V} = \nabla \cdot (k \nabla T) \tag{2.7}$$

which can be simplified to

$$\rho e_t + \rho \vec{V} \cdot \nabla e = \nabla \cdot (k \nabla T) \tag{2.8}$$

with the incompressibility assumption $\nabla \cdot \vec{V} = 0$. Assuming e depends on at most temperature, and that the specific heat at constant volume is a constant, c_v, independent of temperature leads to

$$e = e_o + c_v\,(T - T_o) \tag{2.9}$$

where e_o is internal energy per unit mass at some reference temperature T_o [36]. This allows equation (2.8) to be rewritten as

$$\rho c_v T_t + \rho c_v \vec{V} \cdot \nabla T = \nabla \cdot (k \nabla T) \tag{2.10}$$

which is a convection diffusion equation for T. Ignoring the effects of convection, i.e. setting $\vec{V} = 0$, leads to

$$\rho c_v T_t = \nabla \cdot (k \nabla T) \tag{2.11}$$

which is the standard heat equation.

2.2.3 Interface Velocity

Throughout this text, *unreacted* and *reacted* incompressible flows are separated by an interface across which the unreacted material is converted to the reacted material, and we use "u" and "r" subscripts to refer to the unreacted and reacted materials, respectively. The interface velocity is denoted by $\vec{W} = D\vec{N}$ where D is the normal component of the interface velocity and $\vec{N} =< n_1, n_2, n_3 >$ is the local unit normal to the interface. The normal component of the interface velocity is calculated by adding the unreacted materials normal velocity to the reaction speed, S. That is, $D = (V_N)_u + S$ where $V_N = \vec{V} \cdot \vec{N}$ is the normal velocity.

2.2.4 Jump Conditions

Conservation of mass, momentum and energy in equations (2.2) through (2.6) implies the standard Rankine-Hugoniot jump conditions across the interface

$$[\rho(V_N - D)] = 0 \tag{2.12}$$

$$\left[\rho(V_N - D)^2 + p\right] = 0 \tag{2.13}$$

$$\left[\left(\rho e + \frac{\rho(V_N - D)^2}{2} + p\right)(V_N - D)\right] = \left[k \nabla T \cdot \vec{N}\right] \tag{2.14}$$

where $[A] = A_r - A_u$ defines "$[\cdot]$" as the jump in a quantity across the interface. Note that equation (2.14) was derived assuming that the tangential velocities are

continuous across the interface, i.e. $[V_{T_1}] = [V_{T_2}] = 0$ where T_1 and T_2 are the unit tangent vectors. This is true as long as $D \neq V_N$, i.e. $S \neq 0$. For more details, see [35].

Equation (2.14) can be rewritten as

$$-\rho_u S \left(\left[e + \frac{p}{\rho} \right] + \frac{\rho_u^2 S^2}{2} \left[\frac{1}{\rho^2} \right] \right) = \left[k \nabla T \cdot \vec{N} \right] \tag{2.15}$$

using equation (2.12) and $D = (V_N)_u + S$. It is often easier to deal with the enthalpy per unit mass, $h = e + \frac{p}{\rho}$ instead of the internal energy. Assuming h depends on at most temperature, and that the specific heat at constant pressure is a constant, c_p, independent of temperature leads to

$$h = h_o + c_p \left(T - T_o \right) \tag{2.16}$$

where h_o is the enthalpy per unit mass at some reference temperature T_o [36]. Then equation (2.15) can be rewritten as

$$-\rho_u S \left([h_o] + [c_p] \left(T_I - T_o \right) + \frac{\rho_u^2 S^2}{2} \left[\frac{1}{\rho^2} \right] \right) = \left[k \nabla T \cdot \vec{N} \right] \tag{2.17}$$

where we have used the fact that the temperature is continuous across the interface, $[T] = 0$, and labeled the interface temperature T_I. It is convenient to choose the reference temperature T_o equal to the standard temperature at which the reaction takes place, e.g. in the case of freezing water $T_o = 273K$.

In the case of the Stefan problem, we assume that there is no expansion across the front, i.e. $[\rho] = 0$. This reduces equation (2.12) to $[V_N] = 0$, equation (2.13) to $[p] = 0$, and equation (2.17) to

$$-\rho S \left([h_o] + [c_p] \left(T_I - T_o \right) \right) = \left[k \nabla T \cdot \vec{N} \right] \tag{2.18}$$

where $\rho = \rho_u = \rho_r$. Furthermore, the standard interface boundary condition of $T_I = T_o$ reduces equation (2.18) to

$$-S \rho \left[h_o \right] = \left[k \nabla T \cdot \vec{N} \right] \tag{2.19}$$

where $[h_o]$ is calculated at the reaction temperature of $T_I = T_o$.

2.2.5 level-set Equation

The level-set equation

$$\phi_t + \vec{W} \cdot \nabla\phi = 0 \tag{2.20}$$

is used to keep track of the interface location as the set of points where $\phi = 0$. The unreacted and reacted materials are then designated by the points where $\phi > 0$ and $\phi \leq 0$ respectively. Using $\phi \leq 0$ instead of $\phi = 0$ for the reacted points removes the measure zero ambiguity of points that happen to lie on the interface. In this sense, the numerical interface lies in between $\phi = 0$ and the positive values of ϕ and can be located numerically by finding the zero level of ϕ. To keep the values of ϕ close to those of a signed distance function, i.e. $|\nabla\phi| = 1$, the reinitialization equation

$$\phi_\tau + S(\phi_o)\left(|\nabla\phi| - 1\right) = 0 \tag{2.21}$$

is iterated for a few steps in ficticious time, τ. The level-set function is used to compute the normal

$$\vec{N} = \frac{\nabla\phi}{|\nabla\phi|} \tag{2.22}$$

and the curvature

$$\kappa = -\nabla \cdot \vec{N} \tag{2.23}$$

in a standard fashion. For more details on the level-set function see [21, 22, 25, 37].

2.3 Numerical Method

2.3.1 Poisson Equation

Consider the variable coefficient Poisson equation in one spatial dimension

$$(\beta u_x)_x = f \tag{2.24}$$

with Dirichlet boundary conditions of $u = g$ on the interface where $\phi = 0$. One can consider each simply connected portion of the domain separately, i.e. equation (2.24) can be solved on the subdomain where $\phi \leq 0$ independent of the solution procedure for the subdomain where $\phi > 0$. Although in practice, it is usually simpler and more efficient to solve for both subdomains at the same time.

The computational domain is discretized into cells of size $\triangle x$ where the cell centers are referred to as grid points or grid nodes with the i-th grid node located at x_i. The cell edges are referred to as fluxes so that the two fluxes bounding the i-th computational cell are located at $x_{i\pm\frac{1}{2}}$. The solution to the Poisson equation is computed at the grid nodes and is written as $u_i = u(x_i)$. An analogous definition holds for f_i, β_i, and ϕ_i. Since β and ϕ are known only at the grid nodes x_i, their values at the fluxes is defined by the linear average of the nodal values, e.g. $\phi_{i+\frac{1}{2}} = \frac{\phi_i+\phi_{i+1}}{2}$ is a second order accurate approximation to ϕ at the flux located between the i-th and $(i+1)$-th cells.

In the absence of an irregular interface, the standard discretization for equation (2.24)

$$\frac{\beta_{i+\frac{1}{2}}\left(\frac{u_{i+1}-u_i}{\triangle x}\right) - \beta_{i-\frac{1}{2}}\left(\frac{u_i-u_{i-1}}{\triangle x}\right)}{\triangle x} = f_i \tag{2.25}$$

can be used to solve this problem with Dirichlet $u = g$ boundary conditions on $\partial\Omega$ enforced by setting $u_i = g_i$ when x_i is a boundary point. For each unknown,

u_i, equation (2.25) is used to fill in one row of a matrix creating a linear system of equations. Since the resulting matrix is symmetric, a wide number of fast linear solvers can be used. For linear solvers that require an initial guess, setting all u_i identically equal to zero is usually sufficient. In the examples section, the symmetric linear system is solved with a PCG method using an Incomplete Choleski preconditioner [24].

Next, suppose that an interface point, x_I, is located in between two grid points x_i and x_{i+1} with a Dirichlet $u = u_I$ boundary condition applied at x_I. Consider the solution to the left of x_I noting that it is independent of the solution to the right of x_I. Equation (2.25) is still valid for all the unknowns to the left and including u_{i-1}, but can no longer be applied at x_i to solve for u_i since the subdomain to the left of x_I does not contain a valid value of u_{i+1}. This can be remedied by defining a ghost value of u_{i+1}^G at x_{i+1} and rewriting equation (2.25) as

$$\frac{\beta_{i+\frac{1}{2}}\left(\frac{u_{i+1}^G - u_i}{\triangle x}\right) - \beta_{i-\frac{1}{2}}\left(\frac{u_i - u_{i-1}}{\triangle x}\right)}{\triangle x} = f_i \tag{2.26}$$

in order to solve for u_i. Possible candidates for u_{i+1}^G include

$$u_{i+1}^G = u_I \tag{2.27}$$

$$u_{i+1}^G = \frac{u_I + (\theta - 1)\, u_i}{\theta} \tag{2.28}$$

and

$$u_{i+1}^G = \frac{2u_I + (2\theta^2 - 2)\, u_i + (-\theta^2 + 1)\, u_{i-1}}{\theta^2 + \theta} \tag{2.29}$$

using constant, linear and quadratic extrapolation respectively. Here $\theta \in [0, 1]$ is defined by $\theta = \frac{x_I - x_i}{\triangle x}$, and can be calculated as $\theta = \frac{|\phi|}{\triangle x}$ since $\phi = 0$ at x_I and is signed distance away from x_I. Since equations (2.28) and (2.29) are poorly behaved for small θ, they are not used when $\theta \leq \triangle x$. Instead, u_i is set equal to

u_I which effectively moves the interface from x_I to x_i. This second order accurate perturbation of the interface location does not degrade the overall second order accuracy of the solution obtained using equation (2.25) to solve for the remaining unknowns. Furthermore, $u_i = u_I$ is second order accurate as long as the desired solution has bounded first derivatives.

Plugging equation (2.29) into equation (2.25) gives a nonsymmetric discretization of

$$\frac{\left(\frac{u_I - u_i}{\theta \triangle x}\right) - \left(\frac{u_i - u_{i-1}}{\triangle x}\right)}{.5\left(\theta \triangle x + \triangle x\right)} = f_i \tag{2.30}$$

in the case of $\beta = 1$. Equation (2.30) is the nonsymmetric discretization used in [26] and [29] to obtain second order accurate numerical methods. That is, both [26] and [29] use the quadratic extrapolation given in equation (2.29) to obtain second order accuracy. Alternatively, plugging (2.28) into equation (2.25) gives a symmetric discretization of

$$\frac{\beta_{i+\frac{1}{2}}\left(\frac{u_I - u_i}{\theta \triangle x}\right) - \beta_{i-\frac{1}{2}}\left(\frac{u_i - u_{i-1}}{\triangle x}\right)}{\triangle x} = f_i \tag{2.31}$$

based on linear extrapolation in the partial cell. Surprisingly, this symmetric discretization is second order accurate as well. This was first pointed out in [38] and is elaborated on here.

Assume that the standard second order accurate discretization in equation (2.25) is used to obtain the standard linear system of equations for u at every grid point except for x_i, and equation (2.26) is used to write a linear equation for u_i introducing a new unknown u_{i+1}^G. The system is closed with equation (2.28) for u_{i+1}^G. In practice, equations (2.28) and (2.26) are combined to obtain equation (2.31) and a symmetric linear system. Solving this linear system of equations leads to well determined values (up to some prescribed tolerance near

round-off error levels) of u at each grid node in the subdomain as well as a well determined value of u_{i+1}^G (from equation (2.28)). For the sake of reference, we designate $\vec{u}$ as the solution vector containing all these values of u. Below, $\vec{u}$ is shown to be a second order accurate solution to equation (2.24) on the subdomain under consideration by showing that it is the second order accurate solution to a modified problem where the interface location has been perturbed by $O(\Delta x^2)$.

Consider a modified problem where a Dirichlet boundary condition of $u_{i+1} = u_{i+1}^G$ is specified at x_{i+1}, and u_{i+1}^G is chosen to be the value of u_{i+1}^G from $\vec{u}$ defined above. This modified problem can be discretized to second order accuracy everywhere using the standard discretization in equation (2.25) at every node except at x_i where equation (2.26) is used instead. Note that equation (2.26) *is* the standard second order accurate discretization when a Dirichlet boundary condition of $u_{i+1} = u_{i+1}^G$ is applied at x_{i+1}. Thus, this new linear system of equations can be solved in standard fashion to obtain a second order accurate solution at each grid node. The realization that $\vec{u}$ is an exact solution to *this* new linear system implies that $\vec{u}$ is a valid second order accurate solution to this modified problem.

Next consider the interface location dictated by the modified problem. Since $\vec{u}$ is a second order accurate solution to the modified problem, $\vec{u}$ can be used to obtain the modified problem interface location to second order accuracy. The linear interpolant that uses u_i at x_i and u_{i+1}^G at x_{i+1} predicts an interface location of *exactly* x_I. Since higher order accurate interpolants (higher than linear) can contribute at most an $O(\Delta x^2)$ perturbation of the interface location, the interface location dictated by the modified problem is at most an $O(\Delta x^2)$ perturbation of the true interface location, x_I.

As a final note, plugging equation (2.27) into equation (2.25) effectively perturbs the interface by an $O(\triangle x)$ amount resulting in a first order accurate algo-

rithm.

In certain situations, β may only be known at the grid nodes and the interface in which case $\beta_{i+\frac{1}{2}}$ in equation (2.31) is determined from a ghost value, β^G_{i+1}, and the usual averaging

$$\beta_{i+\frac{1}{2}} = \frac{\beta_i + \beta^G_{i+1}}{2} \tag{2.32}$$

noting that the ghost value is easily defined using linear extrapolation

$$\beta^G_{i+1} = \frac{\beta_I + (\theta - 1)\,\beta_i}{\theta} \tag{2.33}$$

according to equation (2.28).

In multiple spatial dimensions, the equations are discretized in a dimension by dimension manner using the one dimensional discretization outlined above. That is, the $(\beta u_x)_x$, $(\beta u_y)_y$ and $(\beta u_z)_z$ terms in equation (2.1) are each discretized independently in the same manner that $(\beta u_x)_x$ was discretized in equation (2.24) above.

2.3.2 Heat Equation

Consider the heat equation (2.11) with an explicit Euler time discretization

$$\frac{T^{n+1} - T^n}{\triangle t} = \frac{1}{\rho c_v} \nabla \cdot (k \nabla T^n) \tag{2.34}$$

and Dirichlet boundary conditions of $T = g$ on the interface where $\phi = 0$. The $\nabla \cdot (k \nabla T^n)$ term is discretized in the same fashion as the variable coefficient Poisson equation (above) noting that each subdomain can be considered independently. Assuming that ρ and c_v are constant in the subdomain, allows equation (2.34) to be rewritten as

$$\frac{T^{n+1} - T^n}{\triangle t} = \nabla \cdot \left(\hat{k} \nabla T^n \right) \tag{2.35}$$

where $\hat{k} = \frac{k}{\rho c_v}$. For stability, a time step restriction of

$$\triangle t^H \hat{k} \left(\frac{2}{(\theta_1 \triangle x)^2} + \frac{2}{(\theta_2 \triangle y)^2} + \frac{2}{(\theta_3 \triangle z)^2} \right) \leq 1 \tag{2.36}$$

is needed where θ_1, θ_2 and θ_3 are the cell fractions in each spatial dimension for cells cut by the interface with $0 < \theta_1, \theta_2, \theta_3 \leq 1$.

Implicit Euler time discretization

$$\frac{T^{n+1} - T^n}{\triangle t} = \nabla \cdot \left(\hat{k} \nabla T^{n+1} \right) \tag{2.37}$$

avoids the time step stability restriction in equation (2.36). Equation (2.37) can be rewritten as

$$T^{n+1} - \triangle t \nabla \cdot \left(\hat{k} \nabla T^{n+1} \right) = T^n \tag{2.38}$$

where the $\nabla \cdot \left(\hat{k} \nabla T^{n+1} \right)$ term is discretized in the same fashion as the variable coefficient Poisson equation (above). For each unknown, T_i^{n+1}, equation (2.38) is used to fill in one row of a matrix creating a linear system of equations. Since the resulting matrix is symmetric, a wide number of fast linear solvers can be used. For linear solvers that require an initial guess, setting all T_i^{n+1} identically zero is usually sufficient. In the examples section, the symmetric linear system is solved with a PCG method using an Incomplete Choleski preconditioner [24]. Note that equation (2.37) is first order in time and second order in space, and one needs to choose $\triangle t$ proportional to $\triangle x^2$ in order to obtain an overall asymptotic accuracy of $O(\triangle x^2)$. In the numerical examples section, we chose the time step for the heat equation as either $\triangle t^H = .5 \triangle x$ or $\triangle t^H = .5 \triangle x^2$ depending on whether we are trying to obtain first or second order overall accuracy respectively.

The Crank-Nicolson scheme

$$\frac{T^{n+1} - T^n}{\triangle t} = \frac{1}{2} \nabla \cdot \left(\hat{k} \nabla T^{n+1} \right) + \frac{1}{2} \nabla \cdot \left(\hat{k} \nabla T^n \right) \tag{2.39}$$

can be used to achieve second order accuracy in both space and time with $\triangle t$ proportional to $\triangle x$. In the numerical examples section, we choose $\triangle t^H = .5\triangle x$. For the Crank-Nicholson scheme,

$$T^{n+1} - \frac{\triangle t}{2}\nabla \cdot \left(\hat{k}\nabla T^{n+1}\right) = T^n + \frac{\triangle t}{2}\nabla \cdot \left(\hat{k}\nabla T^n\right) \tag{2.40}$$

is used to create a symmetric linear system of equations for the unknowns T_i^{n+1}.

2.3.3 Stefan Problem

2.3.3.1 Interface Velocity

There are four cases to consider when computing $(T_x)_{i,j,k}$: Case 1 - If $T_{i,j,k}$, $T_{i-1,j,k}$ and $T_{i+1,j,k}$ all lie on the same side of the interface, then $(T_x)_{i,j,k}$ is calculated with $T_{i,j,k}$ and either $T_{i-1,j,k}$ or $T_{i+1,j,k}$ depending on which of these two are closer to the interface as determined by the local absolute value of ϕ. Case 2 - If $T_{i,j,k}$ and $T_{i-1,j,k}$ lie on one side of the interface, and $T_{i+1,j,k}$ lies on the other side of the interface, then $(T_x)_{i,j,k}$ is calculated using $T_{i,j,k}$ and the local interface boundary condition for T as long as the distance from $\vec{x}_{i,j,k}$ to the interface is greater than $\triangle x^2$. Otherwise, $T_{i-1,j,k}$ is used in place of $T_{i,j,k}$. Case 3 - If $T_{i,j,k}$ and $T_{i+1,j,k}$ lie on one side of the interface, and $T_{i-1,j,k}$ lies on the other side of the interface, then $(T_x)_{i,j,k}$ is calculated using $T_{i,j,k}$ and the local interface boundary condition for T as long as the distance from $\vec{x}_{i,j,k}$ to the interface is greater than $\triangle x^2$. Otherwise, $T_{i+1,j,k}$ is used in place of $T_{i,j,k}$. Case 4 - If $T_{i,j,k}$ lies on one side of the interface and both $T_{i-1,j,k}$ and $T_{i+1,j,k}$ lie on the opposite side of the interface, then the two local interface boundary conditions for T are used to calculate $(T_x)_{i,j,k}$ as long as the distance between the two interface locations is greater than $\triangle x^2$. Otherwise, the problem is under-resolved and we set $(T_x)_{i,j,k} = 0$. T_y and T_z are calculated in a similar fashion, and $\vec{N}$ is computed using equation (2.22) as described in

[22]. Finally, the normal derivative of the temperature is defined as $T_N = \nabla T \cdot \vec{N}$ at each grid node near the interface.

Once T_N is defined in a band about the interface, we extrapolate the values of $(T_N)_r$ from the reacted side of the interface to the unreacted side of the interface and extrapolate the values of $(T_N)_u$ from the unreacted side of the interface to the reacted side of the interface so that both $(T_N)_r$ and $(T_N)_u$ are defined at every grid point in a band about the interface. This is accomplished with constant extrapolation in the normal direction to the interface and implemented by solving

$$I_\tau \pm \vec{N} \cdot \vec{\nabla} I = 0 \tag{2.41}$$

to steady state where $I_\tau = 0$. This is done separately to advect $I = (T_N)_r$ in one direction and to advect $I = (T_N)_u$ in the other direction. Instead of time marching equation (2.41) in fictitious time τ, a first order accurate solution to the steady state of equation (2.41) is obtained using the fast (velocity) extension method in [39] (which is based on the Fast Marching Method, see e.g. [40]).

For grid nodes adjacent to the interface, equation (2.19) is used to find the reaction speed S where $\left[k\nabla T \cdot \vec{N}\right]$ is calculated in a node by node fashion using the values of $(T_N)_r$ and $(T_N)_u$ that have been defined at each grid node as outlined above.

The level set function is evolved in time from ϕ^n to ϕ^{n+1} using nodal velocities, $\vec{W} = S\vec{N}$, and a 3rd order accurate TVD Runge-Kutta (see [22] and [41]) time stepping method. Detailed discretizations for equation (2.20) and for equation (2.21) are given in [21]. Note that the 5th order WENO discretization from [21] is used to discretize the spatial terms in equations (2.20) and (2.21) for the

numerical examples in this chapter. For stability, a time step restriction of

$$\triangle t^L \left(\frac{w_1}{\triangle x} + \frac{w_2}{\triangle y} + \frac{w_3}{\triangle z} \right) \leq .5 \tag{2.42}$$

is used when solving equation (2.20) with $\vec{W} =< w_1, w_2, w_3 >$. The overall time step is chosen as $\min(\triangle t^H, \triangle t^L)$ where $\triangle t^H = .5\triangle x$ or $\triangle t^H = .5\triangle x^2$.

2.3.3.2 Ghost Cells

Equations (2.38) and (2.40) require a valid value of T^n at each grid point. As the interface moves, the grid points that cross the interface may no longer have valid values of T^n. For example, consider the solidification of water where a grid point in the water with $T^n > 273.15K$ crosses over the interface into the ice. Now that grid point is associated with the ice, but still has $T^n > 273.15K$ as opposed to a correct value of $T^n < 273.15K$. These errors seem to have been ignored by most authors and are probably negligible when the temperature is continuous across the interface. However, when the temperature (or more likely its equivalent in a related problem) is discontinuous across the interface, using this invalid value of T^n can cause significant errors.

In order to determine a ghost cell value of $T^n_{i,j,k}$ at a grid point adjacent to the interface, we use the interface boundary condition $T_I = g(\vec{x_I})$ at the closest interface location $\vec{x}_I = \vec{x}_{i,j,k} - \phi_{i,j,k}\vec{N}$. Then assuming the temperature profile is locally linear, the ghost value is defined as $T^n_{i,j,k} = T_I + \phi_{i,j,k}(T_N)_{i,j,k}$ where $(T_N)_{i,j,k}$ is the value of T_N that has already been extrapolated from the other side of the interface. That is, on the reacted side of the interface the extrapolated value of $(T_N)_u$ is used, and on the unreacted side of the interface the extrapolated value of $(T_N)_r$ is used.

Besides a valid value of T^n, equation (2.40) requires a valid value of $\frac{\triangle t}{2}\nabla \cdot$

$\left(\hat{k}\nabla T^n\right)$ at each grid point implying that ghost cell values of $\frac{\triangle t}{2}\nabla\cdot\left(\hat{k}\nabla T^n\right)$ need to be defined in grid cells adjacent to the interface in case they are uncovered as the interface moves across the grid. Since a second order accurate quadratic extension of a function does not change the values of its second derivative, ghost cell values of $\frac{\triangle t}{2}\nabla\cdot\left(\hat{k}\nabla T^n\right)$ are calculated by extrapolating this term across the interface according to equation (2.41) with $I = \frac{\triangle t}{2}\nabla\cdot\left(\hat{k}\nabla T^n\right)$. Once again the fast extension method from [39] is used. Here, in order to get smooth values of I, an Isobaric Fix technique (see [42] and [21]) is used to extrapolate the values of I across the interface that are at least one grid cell away from the interface, as opposed to the usual procedure of extrapolating the values that are adjacent to the interface.

2.4 Examples

In each example, we use the level-set function ϕ in order to decompose the domain into separate regions. The interior region Ω^- is defined by $\phi \leq 0$ while the exterior region Ω^+ is defined by $\phi > 0$.

2.4.1 Poisson Equation

Here we consider equation (2.1) for cases where β is either constant on each subdomain or spatially varying on each subdomain. When β is constant on each subdomain, it can be moved to the right hand side rewriting equation (2.1) as $\triangle u = \hat{f}$ where $\hat{f} = \frac{f}{\beta}$. Therefore, when β is constant on each subdomain, it can be ignored completely, i.e. absorbed into the right hand side.

Since Ω^- is completely decoupled from Ω^+, we only compute solutions for Ω^- here.

2.4.1.1 Example 1

Consider $u_{xx} = f$ on $\Omega = [-.5, .5]$ with an exact solution of $u = 4x^2 \sin(2\pi x)$ on Ω^- where $\phi = |x| - .313$ so that the interface does not lie on a grid point in any if the test cases. The right hand side is $f = 8\sin(2\pi x) + 32\pi x \cos(2\pi x) - 16\pi^2 x^2 \sin(2\pi x)$.

Figure 2.1 shows the numerical solution with 61 grid points plotted on top of the exact solution and Table 2.1 shows the results of the numerical accuracy tests.

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
41	4.422×10^{-4}	$--$	9.236×10^{-4}	$--$
81	1.132×10^{-4}	1.97	2.654×10^{-4}	1.79
161	2.736×10^{-5}	2.04	7.306×10^{-5}	1.86

Table 2.1: 1D Laplace Equation

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
41	1.939×10^{-2}	$--$	8.072×10^{-2}	$--$
81	5.015×10^{-3}	1.95	2.010×10^{-2}	2.01
161	1.198×10^{-3}	2.06	5.532×10^{-3}	1.86

Table 2.2: 1D Poisson Equation

2.4.1.2 Example 2

Consider $(\beta u_x)_x = f$ on $\Omega = [-.65, .25]$ with an exact solution of $u = e^{4x}\sin(2\pi x)$ and $\beta = \cos(x)$ on Ω^- where $\phi = |x| - .313$ so that the interface does not lie on a grid point in any if the test cases. The right hand side is $f = -2e^{4x}\sin(x)(2\sin(2\pi x) + \pi\cos(2\pi x)) + 4e^{4x}\cos(x)(4\sin(2\pi x) + \pi\cos(2\pi x) - \pi^2\sin(2\pi x))$. Figure 2.2 shows the numerical solution with 61 grid points plotted on top of the exact solution and Table 2.2 shows the results of the numerical accuracy tests.

2.4.1.3 Example 3

This example was taken from [44]. Consider $\triangle u = f$ on $\Omega = [-1, 1] \times [-1, 1]$ with an exact solution of $u = x^2 + y^2$ on Ω^-. The interface is parameterized by $(x(\theta), y(\theta))$ where $x(\theta) = .02\sqrt{5} + (.5 + .2\sin(5\theta))\cos(\theta)$ and $y(\theta) = .02\sqrt{5} + (.5 + .2\sin(5\theta))\sin(\theta)$ with $\theta \in [0, 2\pi]$. In order to compute ϕ, the interface was divided into 2000 equally spaced points. At each grid node, the magnitude of the

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
101×101	7.329×10^{-5}	$--$	9.777×10^{-5}	$--$
201×201	1.776×10^{-5}	2.04	2.427×10^{-5}	2.01
401×401	4.714×10^{-6}	1.92	6.178×10^{-6}	1.97

Table 2.3: 2D Laplace Equation

signed distance function ϕ was computed using the closest point, and the sign of ϕ was computed by using the cross-product between the normal and tangent vectors to the interface, so that ϕ is negative inside the closed contour. The right hand side is $f = 4$. Figure 2.3 shows the numerical solution with 61 grid points in each spatial direction plotted above a contour showing the interface location, and Table 2.3 shows the results of the numerical accuracy tests.

2.4.1.4 Example 4

This example was taken from [44]. Consider $\nabla \cdot (\beta \nabla u) = f$ on $\Omega = [-1, 1] \times [0, 3]$ with an exact solution of $u = e^x(x^2 \sin(y) + y^2)$ and $\beta = 2 + \sin(xy)$ on Ω^-. The interface is parameterized by $(x(\theta), y(\theta))$ where $x(\theta) = .6\cos(\theta) - .3\cos(3\theta)$ and $y(\theta) = 1.5 + .7\sin(\theta) - .07\sin(3\theta) + .2\sin(7\theta)$ with $\theta \in [0, 2\pi]$. In order to compute ϕ, the interface was divided into 2000 equally spaced points. At each grid node, the magnitude of the signed distance function ϕ was computed using the closest point, and the sign of ϕ was computed by using the cross-product between the normal and tangent vectors to the interface, so that ϕ is negative inside the closed contour. The right hand side is $f = y\cos(xy)(e^x(x^2\sin(y) + y^2) + 4xe^x\sin(y) + 2e^x\sin(y)) + \cos(xy)xe^x(x^2\cos(y) + 2y) + (2 + \sin(xy))e^x(-x^2\sin(y) + 2)$. Figure 2.4 shows the numerical solution with 61 grid points in the x-direction and 121 grid points in the y-direction plotted above a contour showing the interface location,

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
81×121	2.414×10^{-4}	$--$	1.129×10^{-3}	$--$
161×241	6.291×10^{-5}	1.93	3.043×10^{-4}	1.87
321×481	1.707×10^{-5}	1.88	7.804×10^{-5}	1.94

Table 2.4: 2D Poisson Equation

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
$26 \times 26 \times 26$	6.394×10^{-5}	$--$	2.272×10^{-4}	$--$
$51 \times 51 \times 51$	1.635×10^{-5}	1.96	5.198×10^{-5}	2.12
$101 \times 101 \times 101$	3.997×10^{-6}	2.03	1.306×10^{-5}	1.99

Table 2.5: 3D Laplace Equation

and Table 2.4 shows the results of the numerical accuracy tests.

2.4.1.5 Example 5

Consider $\triangle u = f$ on $\Omega = [0,1] \times [0,1] \times [0,1]$ with an exact solution of $u(x,y,z) = e^{-x^2-y^2-z^2}$ on Ω^- where $\phi = \sqrt{(x-.5)^2 + (y-.5)^2 + (z-.5)^2} - .3$. The right hand side is $f = 4(x^2+y^2+z^2-3/2)e^{-x^2-y^2-z^2}$. Figure 2.5 shows the $z = .5$ cross section of the numerical solution with 41 grid points in each spatial direction plotted above the $z = .5$ cross section of the interface, and Table 2.5 shows the results of the numerical accuracy tests.

2.4.1.6 Example 6

Consider $\nabla \cdot (\beta \nabla u) = f$ on $\Omega = [0,1] \times [0,1] \times [0,1]$ with an exact solution of $u = \sin(4\pi x)\sin(4\pi y)\sin(4\pi z)$, and $\beta = xyz$ on Ω^- where $\phi = \sqrt{(x-.5)^2 + (y-.5)^2 + (z-.5)^2} - .3$. The right hand side is $f = 4(x^2+y^2+z^2-3/2)e^{-x^2-y^2-z^2}$. Figure 2.6 shows

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
$21 \times 21 \times 21$	1.059×10^{-2}	$--$	3.690×10^{-2}	$--$
$41 \times 41 \times 41$	2.370×10^{-3}	2.16	8.989×10^{-3}	2.03
$81 \times 81 \times 81$	5.619×10^{-4}	2.03	2.170×10^{-3}	2.08

Table 2.6: 3D Poisson Equation

the $z = .5$ cross section of the numerical solution with 81 grid points in each spatial direction plotted above the $z = .5$ cross section of the interface, and Table 2.6 shows the results of the numerical accuracy tests.

2.4.2 Heat Equation

Here we consider equation (2.11) where k is a (possibly different) constant on each subdomain. In this case, equation (2.11) can be rewritten as $T_t = \hat{k}\triangle T$ where $\hat{k} = \frac{k}{\rho c_v}$. In the examples below, we take $\hat{k} = 1$.

2.4.2.1 Example 7

Consider $T_t = T_{xx}$ on $\Omega = [-1, 1]$ with an exact solution of $T = e^{-\pi^2 t}\cos(\pi x)$ on Ω^- where $\phi = |x| - .313$ so that the interface does not lie on a grid point in any if the test cases. Figure 2.7 shows the numerical solution computed with the Crank Nicholson scheme at $t = \frac{1}{\pi^2}$ with 121 grid points plotted on top of the exact solution. Tables 2.7, 2.8 and 2.9 show the results of the numerical accuracy tests.

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
41	1.443×10^{-2}	$--$	2.222×10^{-2}	$--$
81	7.240×10^{-3}	0.99	1.118×10^{-2}	1
161	3.634×10^{-3}	0.99	5.609×10^{-3}	0.97

Table 2.7: 1D Heat Equation - Backward Euler - $\triangle t \approx \triangle x$

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
41	6.198×10^{-4}	$--$	8.866×10^{-4}	$--$
81	1.540×10^{-4}	1.98	2.194×10^{-4}	2.00
161	3.839×10^{-5}	2.01	5.458×10^{-5}	2.00

Table 2.8: 1D Heat Equation - Backward Euler - $\triangle t \approx \triangle x^2$

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
41	4.084×10^{-4}	$--$	6.811×10^{-4}	$--$
81	9.907×10^{-5}	2.01	1.623×10^{-4}	2.08
161	2.424×10^{-5}	2.03	3.993×10^{-5}	2.00

Table 2.9: 1D Heat Equation - Crank Nicholson - $\triangle t \approx \triangle x$

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
81×81	1.282×10^{-5}	$--$	2.340×10^{-4}	$--$
161×161	5.618×10^{-6}	1.19	4.131×10^{-5}	2.50
321×321	2.539×10^{-6}	1.14	7.966×10^{-6}	2.37

Table 2.10: 2D Heat Equation - Backward Euler - $\triangle t \approx \triangle x$

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
81×81	4.886×10^{-6}	$--$	2.340×10^{-4}	$--$
161×161	9.307×10^{-7}	2.39	4.131×10^{-5}	2.50
321×321	1.687×10^{-7}	2.46	7.569×10^{-7}	5.77

Table 2.11: 2D Heat Equation - Backward Euler - $\triangle t \approx \triangle x^2$

2.4.2.2 Example 8

Consider $T_t = \triangle T$ on $\Omega = [-1, 1] \times [-1, 1]$ with an exact solution of $T = e^{-2t}\sin(x)\sin(y)$ on Ω^-. The interface is parameterized by $(x(\theta), y(\theta))$ where $x(\theta) = .02\sqrt{5} + (.5 + .2\sin(5\theta))\cos(\theta)$ and $y(\theta) = .02\sqrt{5} + (.5 + .2\sin(5\theta))\sin(\theta)$ with $\theta \in [0, 2\pi]$. In order to compute ϕ, the interface was divided into 2000 equally spaced points. At each grid node, the magnitude of the signed distance function ϕ was computed using the closest point, and the sign of ϕ was computed by using the cross-product between the normal and tangent vectors to the interface, so that ϕ is negative inside the closed contour. Figure 2.8 shows the numerical solution computed with the Crank Nicholson scheme at $t = .1$ with 81 grid points in each spatial direction plotted above a contour showing the interface location. Tables 2.10, 2.11 and 2.12 show the results of the numerical accuracy tests.

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
81×81	5.440×10^{-6}	$--$	2.340×10^{-4}	$--$
161×161	7.888×10^{-7}	2.78	4.131×10^{-5}	2.50
321×321	1.424×10^{-7}	2.46	6.207×10^{-7}	6.05

Table 2.12: 2D Heat Equation - Crank Nicholson - $\triangle t \approx \triangle x$

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
$26 \times 26 \times 26$	1.727×10^{-6}	$--$	4.129×10^{-6}	$--$
$51 \times 51 \times 51$	7.591×10^{-7}	1.19	1.937×10^{-6}	1.09
$101 \times 101 \times 101$	3.596×10^{-7}	1.08	9.524×10^{-7}	1.03

Table 2.13: 3D Heat Equation - Backward Euler - $\triangle t \approx \triangle x$

2.4.2.3 Example 9

Consider $T_t = \triangle T$ on $\Omega = [0, .5] \times [0, .5] \times [0, .5]$ with an exact solution of $T = e^{-3t}\sin(x)\sin(y)\sin(z)$ on Ω^- where $\phi = \sqrt{(x-.5)^2 + (y-.5)^2 + (z-.5)^2} - .15$. Figure 2.9 shows the $z = .25$ cross section of the numerical solution computed with the Crank Nicholson scheme at $t = .1$ with 41 grid points in each spatial location plotted above the $z = .25$ cross section of the interface. Tables 2.13, 2.14 and 2.15 show the results of the numerical accuracy tests.

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
$26 \times 26 \times 26$	4.139×10^{-7}	$--$	1.294×10^{-6}	$--$
$51 \times 51 \times 51$	1.049×10^{-7}	1.98	2.958×10^{-7}	2.12
$101 \times 101 \times 101$	2.559×10^{-8}	2.03	7.536×10^{-8}	1.97

Table 2.14: 3D Heat Equation - Backward Euler - $\triangle t \approx \triangle x^2$

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
$26 \times 26 \times 26$	5.607×10^{-7}	$--$	2.805×10^{-6}	$--$
$51 \times 51 \times 51$	7.620×10^{-8}	2.87	2.079×10^{-7}	3.75
$101 \times 101 \times 101$	2.094×10^{-8}	1.86	5.617×10^{-8}	1.88

Table 2.15: 3D Heat Equation - Crank Nicholson - $\triangle t \approx \triangle x$

2.4.3 Stefan Problem

Here we consider the Stefan problem with $k = \rho = c_v = 1$ in each subdomain. Then the temperature in each subdomain is governed by the heat equation $T_t = \triangle T$, and the interface velocity is given by

$$S = -\frac{1}{[h_o]} \left[\nabla T \cdot \vec{N} \right] \tag{2.43}$$

from equation (2.19). A Dirichlet $T = 0$ interface boundary condition is used at the interface separating the two subdomains.

2.4.3.1 Example 10

Let $\Omega = [0, 1]$ with an exact solution of $T = e^{t-x+.5} - 1$ on Ω^- and $T = 0$ on Ω^+ where $\phi = x - .5$ at $t = 0$. Here, $[h_o] = -1$ so that the interface velocity is $S = \left[\nabla T \cdot \vec{N} \right]$. Dirichlet boundary conditions are enforced on the $\partial\Omega$ using the exact solutions. Figure 2.10 shows the numerical solution computed with the Crank Nicholson scheme at $t = .25$ with 81 grid points plotted on top of the exact solution. Tables 2.16 and 2.17 show the results of the numerical accuracy tests. Note that the Crank Nicholson scheme is no longer second order accurate overall, due to the loss of accuracy in the computed interface velocity. In Table 2.18, we use the exact interface velocity and obtain the expected second order accurate results. In general, the exact interface velocity is not known, so the similarity

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
41	7.065×10^{-4}	$--$	8.949×10^{-4}	$--$
81	3.542×10^{-4}	0.99	4.527×10^{-4}	1.01
161	1.769×10^{-4}	1.01	2.272×10^{-4}	0.98

Table 2.16: 1D Stefan Problem - Backward Euler - $\triangle t \approx \triangle x$

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
41	2.372×10^{-4}	$--$	4.501×10^{-4}	$--$
81	1.129×10^{-4}	1.07	2.125×10^{-4}	1.08
161	5.388×10^{-5}	1.06	9.975×10^{-5}	1.09

Table 2.17: 1D Stefan Problem - Backward Euler - $\triangle t \approx \triangle x^2$

between the results in Tables 2.16 and 2.17 compel us to use the simpler backward Euler method for the two and three dimensional examples that follow. Even though the backward Euler method is only first order accurate overall, we still derive benefits from the symmetric second order accurate spatial discretization.

2.4.3.2 Example 11

Let $\Omega = [-5, 5] \times [-5, 5]$ and consider the Frank Sphere which is an exact solution of the Stefan problem, see for example [43]. In two spatial dimensions, the exact interface location is a disk of radius $R = s_o\sqrt{t}$ with an exact solution of $T = 0$

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
41	2.716×10^{-5}	$--$	6.621×10^{-5}	$--$
81	6.789×10^{-6}	2.00	1.479×10^{-5}	2.16
161	1.681×10^{-6}	2.01	4.055×10^{-6}	1.87

Table 2.18: 1D Stefan Problem - Crank Nicholson - $\triangle t \approx \triangle x$

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
41	1.912×10^{-3}	$--$	2.706×10^{-2}	$--$
81	9.587×10^{-4}	0.99	1.600×10^{-2}	0.75
161	4.195×10^{-4}	1.19	1.148×10^{-2}	0.47

Table 2.19: 2D Stefan Problem - Backward Euler - $\triangle t \approx \triangle x$

inside the disk and

$$T = T_\infty \left(1 - \frac{F(s)}{F(s_o)}\right) \tag{2.44}$$

outside the disk where $s = \frac{r}{\sqrt{t}}$, $r = \sqrt{x^2 + y^2}$, $F(s) = E_1(s^2/4)$,

$$E_1(z) = \int_z^\infty \frac{\exp(z)}{z} dz \tag{2.45}$$

and the value of s_o depends on the choice of T_∞, e.g. we take $T_\infty = -.5$ implying $s_o = 1.56$. Initially, $t = 1$ so that $R = 1.56$ and $\phi = \sqrt{x^2 + y^2} - 1.56$. In this example, $[h_o] = 1$ and the interface velocity is given by $S = -\left[\nabla T \cdot \vec{N}\right] = \frac{s_o}{2\sqrt{t}}$. Dirichlet boundary conditions are enforced on the $\partial\Omega$ using the exact solution. Figure 2.11 shows the numerical solution computed with the backward Euler scheme at $t = 1.5$ with 60 grid points in each spatial dimension plotted on top of the exact solution. Figure 2.12 shows the convergence of the Frank Sphere solution's radius as the grid size is refined. Table 2.19 and 2.20 shows the results of the numerical accuracy tests on the temperature field and the radius, respectively. Note that the numerical estimates for the radius were calculated using only the grid points adjacent to the interface.

2.4.3.3 Example 12

Let $\Omega = [-1.5, 1.5] \times [-1.5, 1.5]$ with an initially circular interface of radius .1 given by $\phi = \sqrt{x^2 + y^2} - .1$. Initially, $T = 0$ inside the circle and $T = -.5$ outside

Number of Points	$L^1 - \text{error}$	order	$L^\infty - \text{error}$	order
21	1.812×10^{0}	$--$	1.674×10^{-1}	$--$
41	3.105×10^{-1}	2.54	6.350×10^{-2}	1.39
81	6.013×10^{-2}	2.30	2.847×10^{-2}	1.15
161	1.396×10^{-2}	2.17	1.159×10^{-2}	1.29

Table 2.20: 2D Stefan Problem - Backward Euler - $\triangle t \approx \triangle x^2$

the circle. Here, $[h_o] = 1$ so the interface velocity is given by $S = -\left[\nabla T \cdot \vec{N}\right]$. Dirichlet boundary conditions of $T = -.5$ are enforced on $\partial\Omega$. The $T = -.5$ material is supercooled and the interface grows outward in an unstable fashion as shown in figures 2.13 and 2.14 after 5 iterations with respectively 500 and 1000 grid points in each spatial dimension.

2.4.3.4 Example 13

Let $\Omega = [-1.5, 1.5] \times [-1.5, 1.5] \times [-1.5, 1.5]$ with an initially spherical interface of radius .1 given by $\phi = \sqrt{x^2 + y^2 + z^2} - .1$. Here, $[h_o] = 1$ so the interface velocity is given by $S = -\left[\nabla T \cdot \vec{N}\right]$. Dirichlet boundary conditions of $T = -.5$ are enforced on $\partial\Omega$. The $T = -.5$ material is supercooled and the interface grows outward in an unstable fashion as shown in figure 2.15 at $t = .14$ with 100 grid points in each spatial dimension.

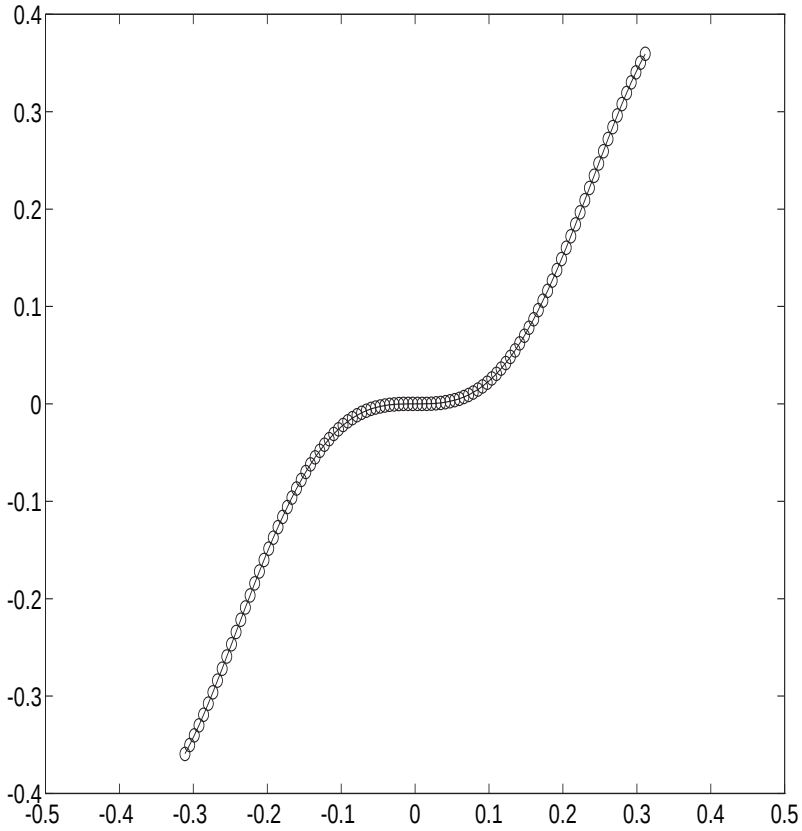

Figure 2.1: 1D Laplace Equation: Numerical solution (open symbol) with 61 grid points plotted on top of the exact solution (solid line).

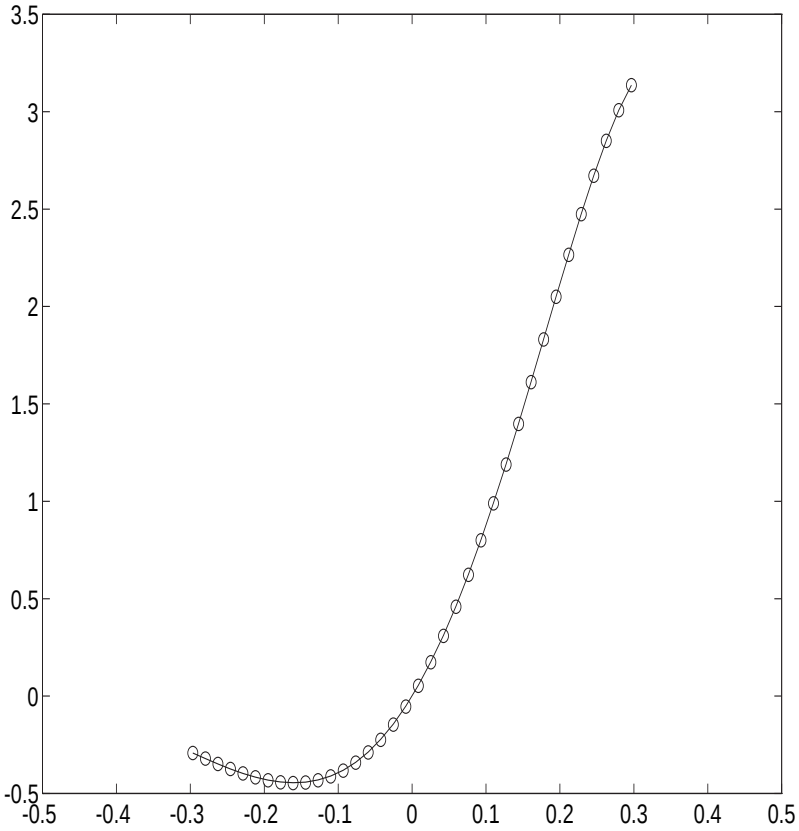

Figure 2.2: 1D Poisson Equation: Numerical solution (open symbol) with 61 grid points plotted on top of the exact solution (solid line).

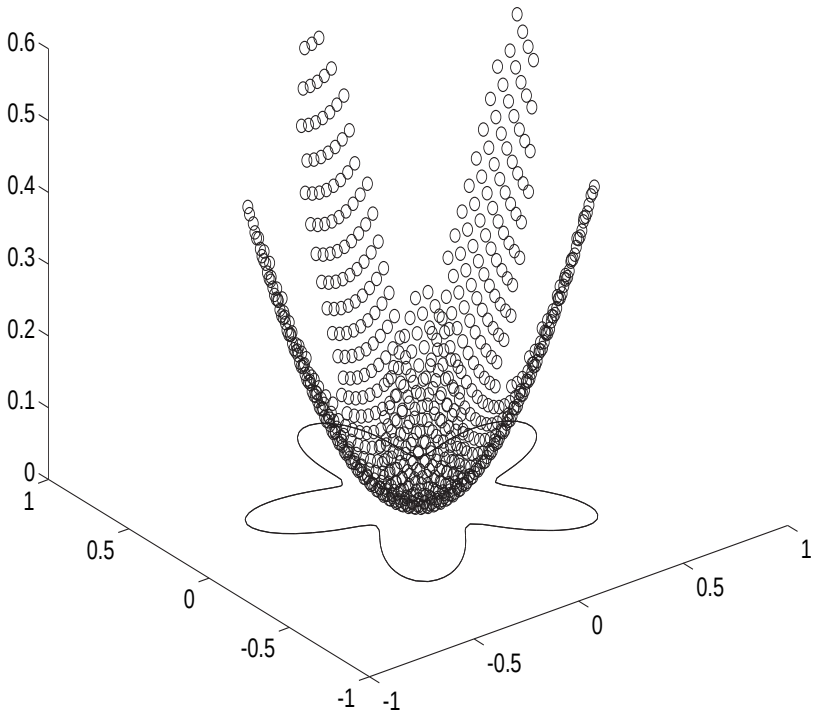

Figure 2.3: 2D Laplace Equation: Numerical solution (open symbol) with 61 grid points in each direction plotted above a contour showing the interface location (solid line).

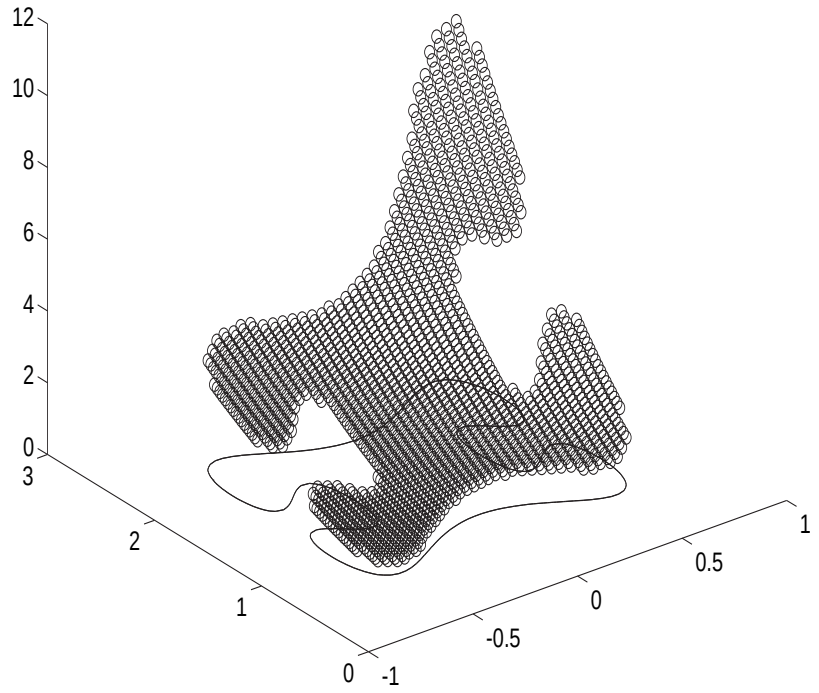

Figure 2.4: 2D Poisson Equation: Numerical solution (opened symbols) with 61 grid points in the x-direction and 121 grid points in the y-direction plotted above a contour showing the interface location (solid line)

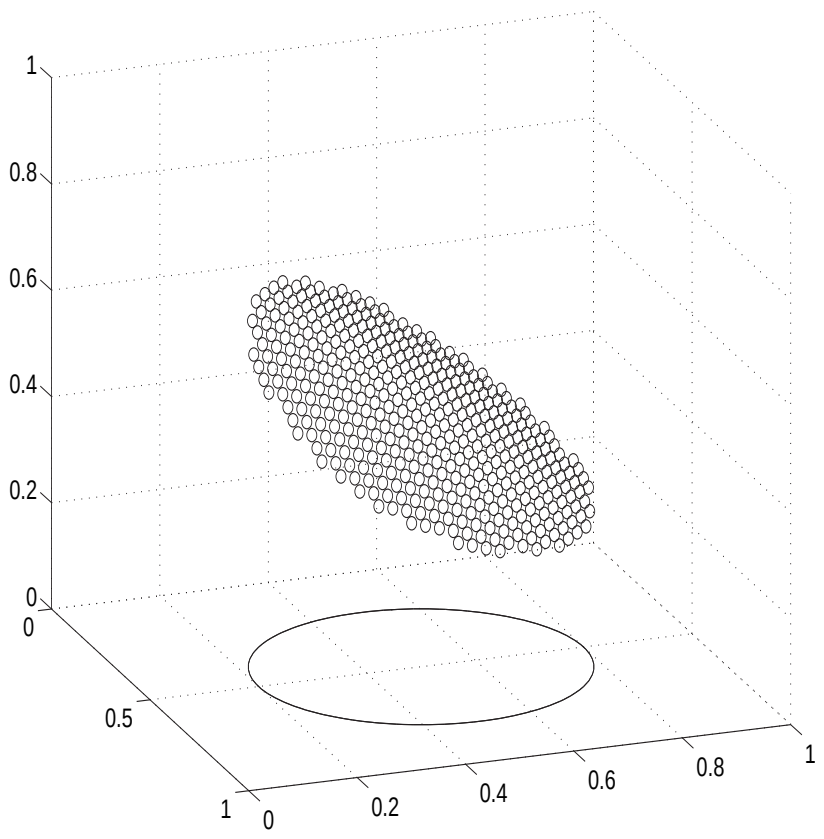

Figure 2.5: 3D Laplace Equation: $z = .5$ cross section of the numerical solution (open symbols) with 41 grid points in each spatial direction plotted above the $z = .5$ cross section of the interface (solid line).

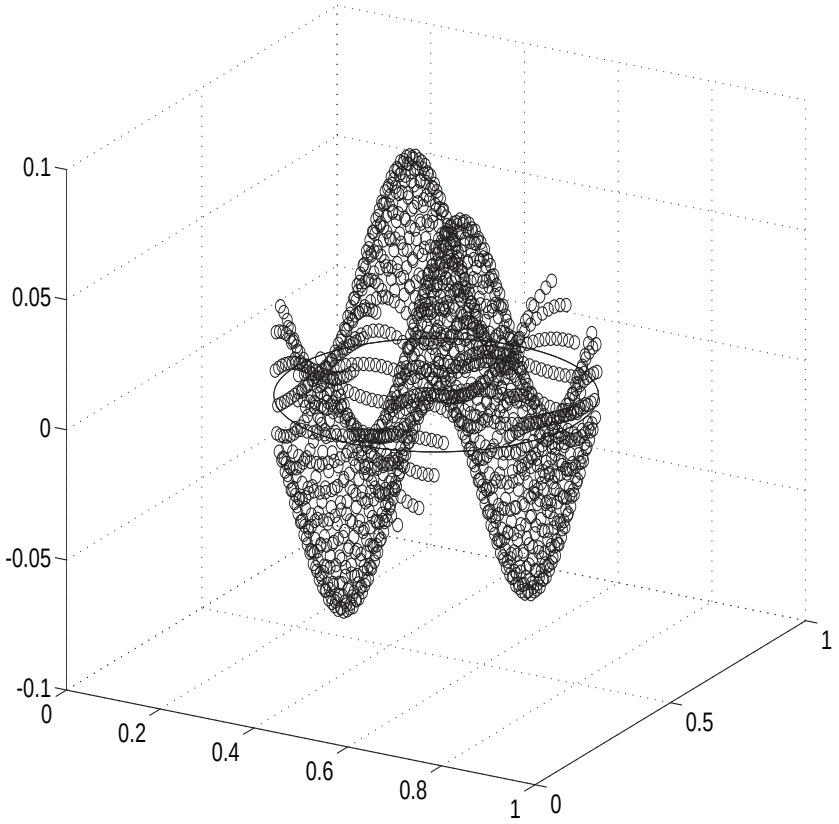

Figure 2.6: 3D Poisson Equation: $z = .5$ cross section of the numerical solution (open symbols) with 41 grid points in each spatial direction plotted above the $z = .5$ cross section of the interface (solid line).

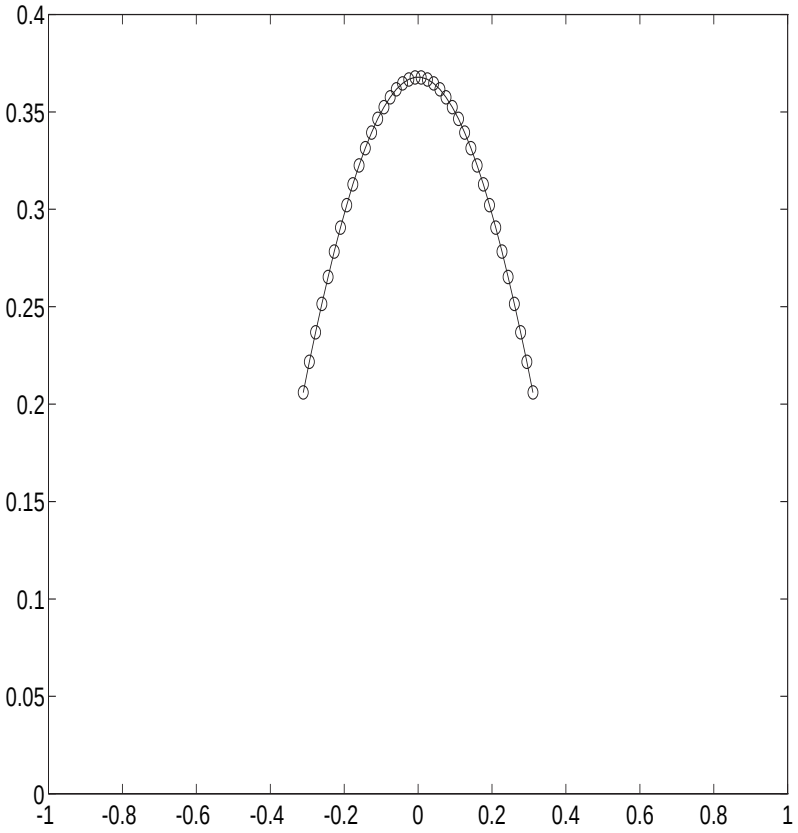

Figure 2.7: 1D Heat Equation: Numerical solution (open symbols) computed with the Crank Nicholson scheme at $t = \frac{1}{\pi^2}$ with 121 grid points plotted on top of the exact solution (solid line).

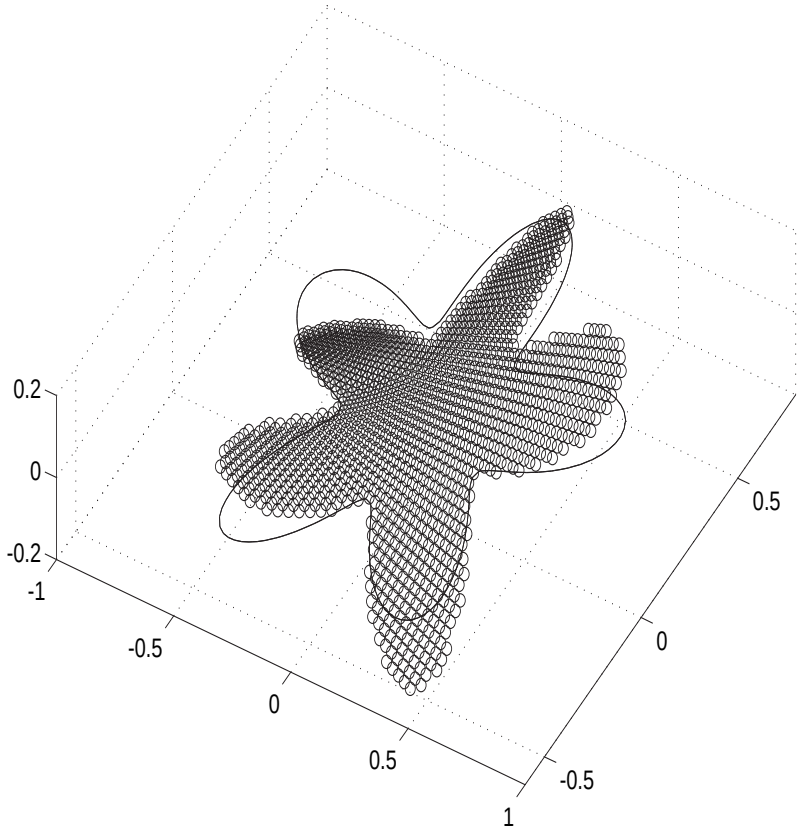

Figure 2.8: 2D Heat Equation. Numerical solution (open symbol) computed with the Crank Nicholson scheme at $t = .1$ with 81 grid points in each spatial direction plotted above a contour showing the interface location (solid line).

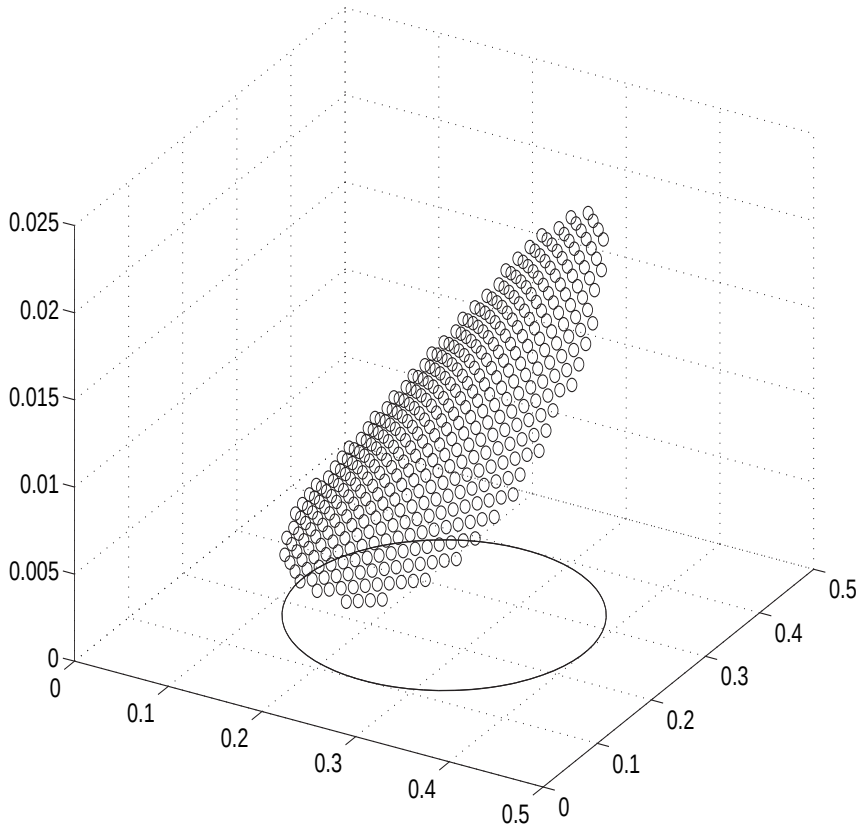

Figure 2.9: 3D Heat Equation: $z = .25$ cross section of the numerical solution (open symbols) computed with the Crank Nicholson scheme at $t = .1$ with 41 grid points in each spatial location plotted above the $z = .25$ cross section of the interface (solid line).

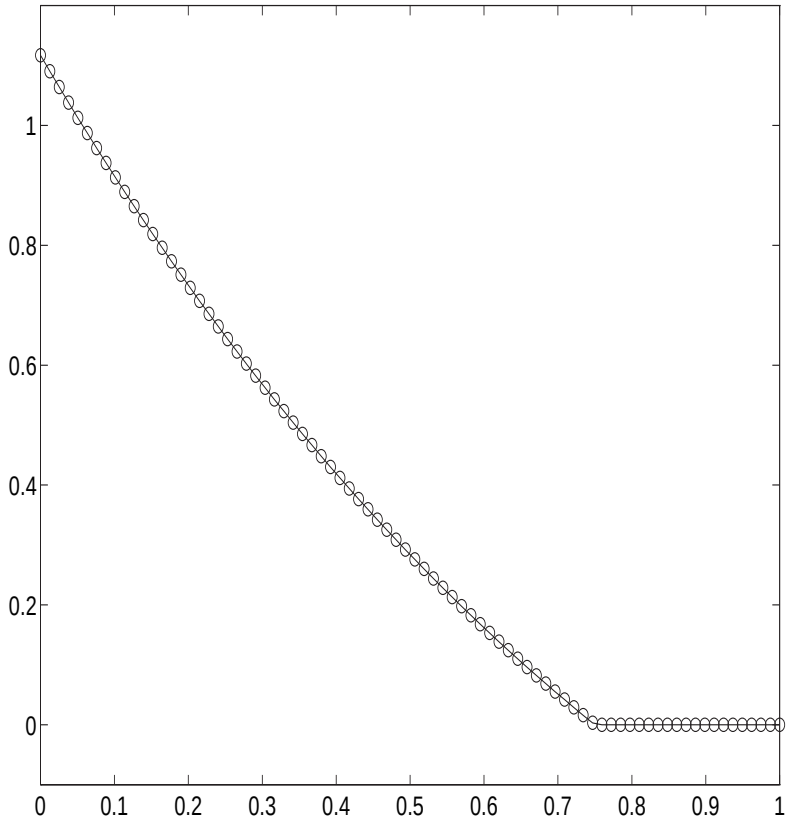

Figure 2.10: 1D Stefan Problem: Numerical solution (open symbol) computed with the Crank Nicholson scheme at $t = .25$ with 81 grid points plotted on top of the exact solution (solid line).

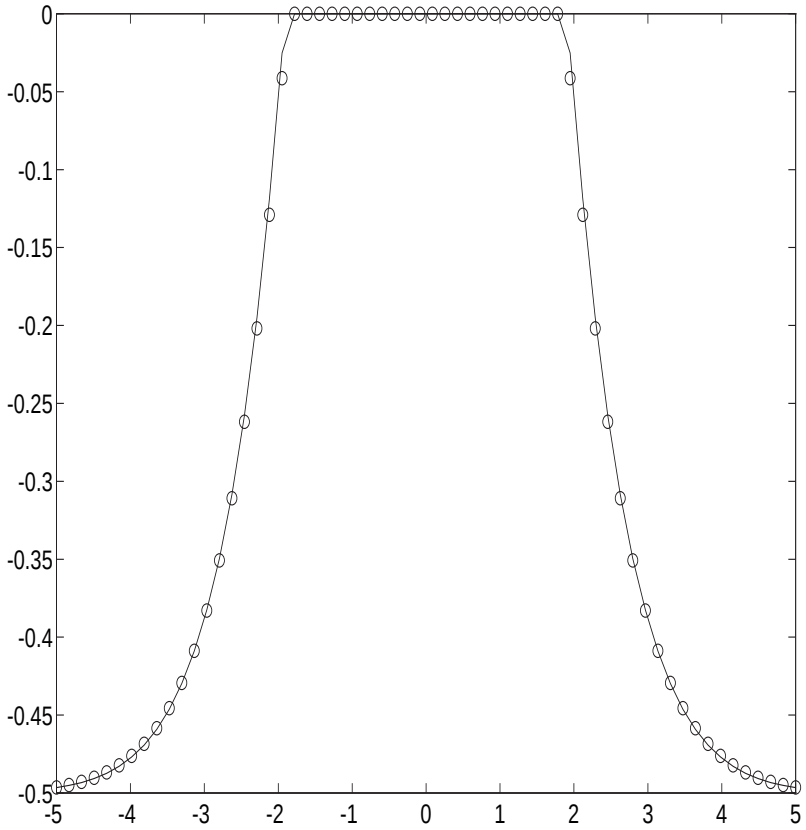

Figure 2.11: 2D Stefan Problem - Frank Sphere - Cross section of the temperature (open symbols) computed with the backward Euler scheme at $t = 1.5$ with 60 grid points in each spatial dimension plotted on top of the exact solution (solid line).

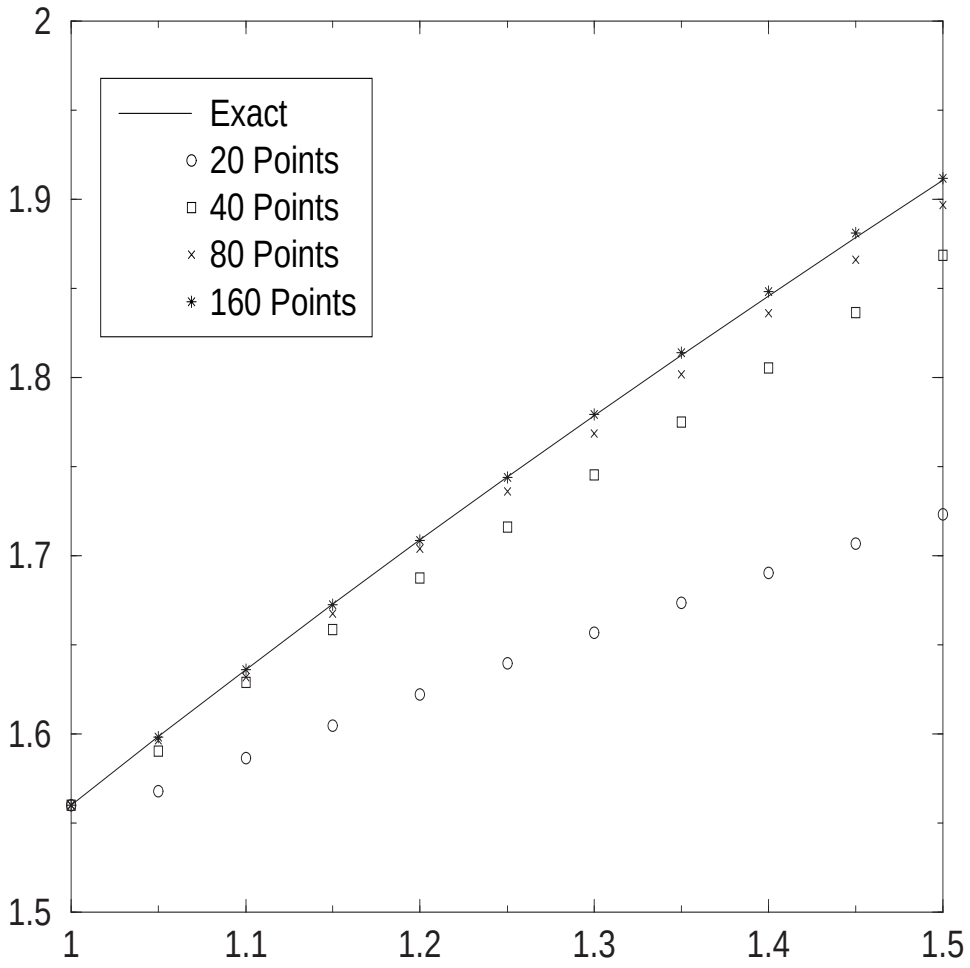

Figure 2.12: 2D Stefan Problem - Radius of the Frank's sphere computed (open symbols) with the backward Euler scheme at $t = 1.5$ with 20, 40, 80 and 160 grid points in each spatial dimension. The solid line shows the exact solution of the radius.

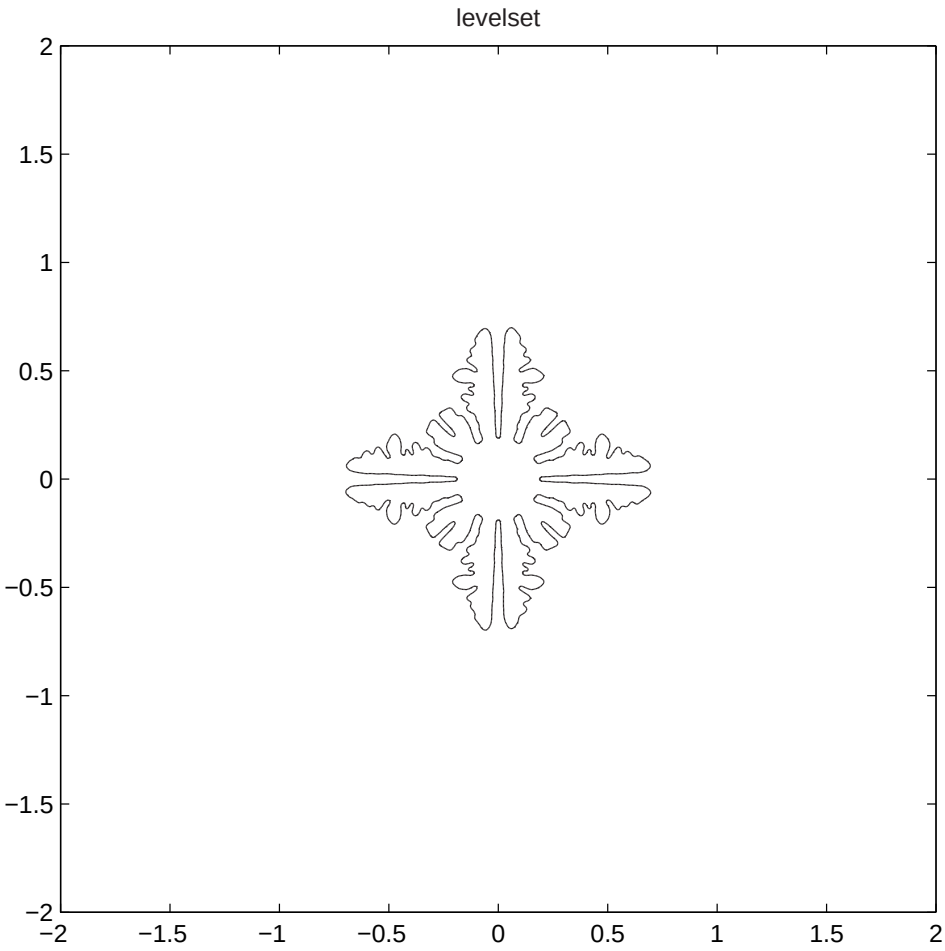

Figure 2.13: 2D Stefan Problem - Initially, $T = 0$ inside the circle of radius 0.1 and $T = -.5$ outside. Shown here is the interface that has grown outward in an unstable fashion after 5 iterations with 500 grid points in each spatial dimension.

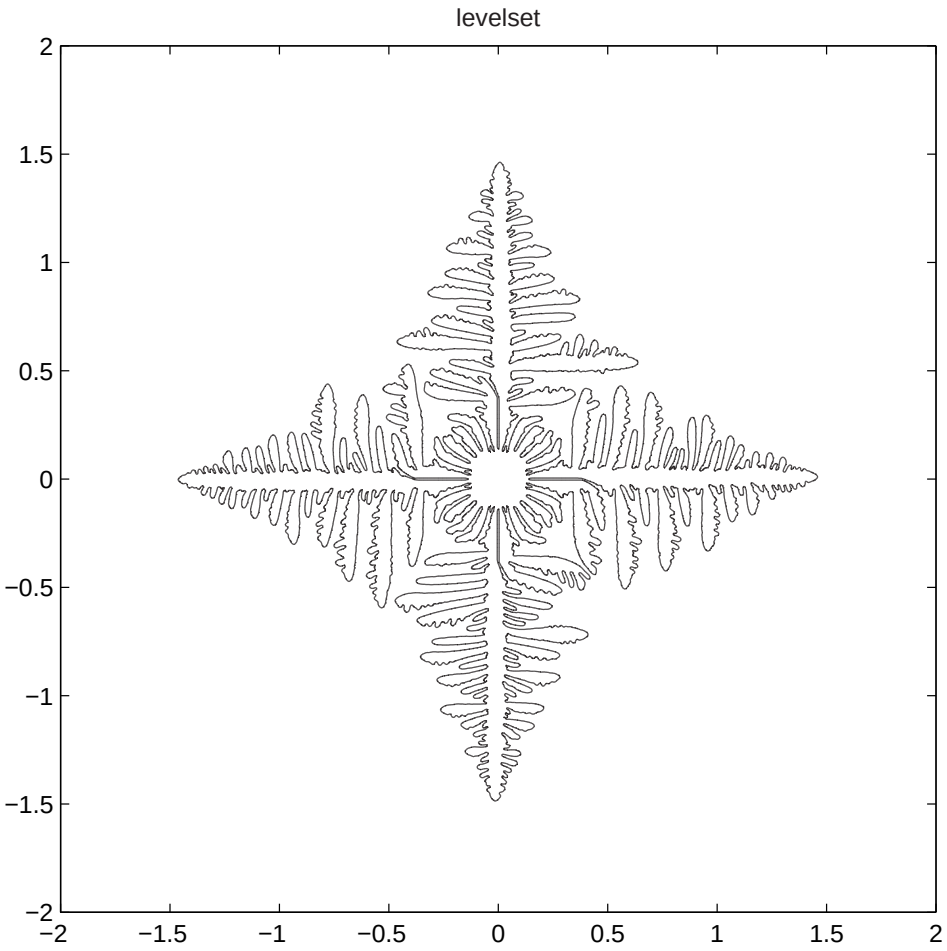

Figure 2.14: 2D Stefan Problem - Initially, $T = 0$ inside the circle of radius 0.1 and $T = -.5$ outside. Shown here is the interface that has grown outward in an unstable fashion after 5 iterations with 1000 grid points in each spatial dimension.

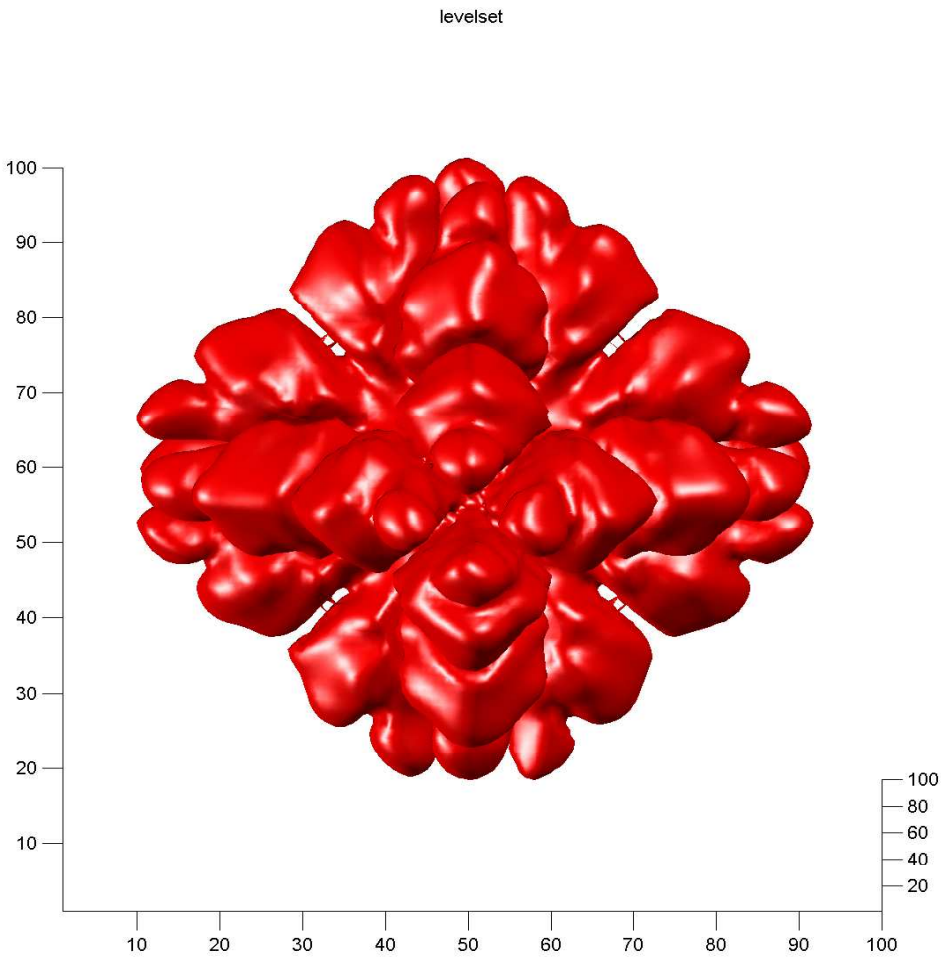

Figure 2.15: 3D Stefan Problem - Initially, $T = 0$ inside the sphere of radius 0.1 and $T = -.5$ outside. Shown here is the interface at $t = .14$ that has grown outward in an unstable fashion with 100 grid points in each spatial dimension.

2.5 Application to multiphase flows

We can apply the method described in the previous chapter along with the method developed in [23] to properly model flame propagation with inclusion of thermal conductivity. The local flame speed S is proportional to the jump in the temperature gradient at the flame interface as derived from the conservation of energy equations of the Euler equations, whereas the fluid velocity and other variables are obtained by solving the incompressible Euler equations for the conservation of mass and momentum.

As an example we compare our numerical results with the analytical Frank sphere solution in one spatial dimension, simulating the melting of ice. Consider the domain [0, 5]. The exact interface location is at distance $R(t) = S\sqrt{t}$ from the origin, with exact solution

$$T(x,t) = \begin{cases} T_o \left(1 - \frac{F(x)}{F(S)}\right) & \text{for } x \leq S \\ 0 & \text{for } x > S \end{cases}$$

, where $x = \frac{s}{\sqrt{t}}$, $F(x) = erfc(\frac{1}{2}x)$ and $erfc(z) = \frac{2}{\sqrt{\pi}} \int_o^z e^{-t^2} dt$. We choose $S = 2$, which implies $T_o = 4.06$.

We start our computation at $t = 1$ with the following values for the fluid variables:

- Density: $\rho_{Left} = 1/2$, $\rho_{Right} = 1$.
- Velocity: $\rho_{Left} = 0$, $\rho_{Right} = 0$.
- Pressure: $\rho_{Left} = 0$, $\rho_{Right} = 0$.

Figure 2.16 shows the temperature at different time. Figure 2.17 shows the space position of the interface compared with the analytic solution and Figure 2.18 shows the velocity initially ($t = 1$) and at the end of the computation ($t = 3$).

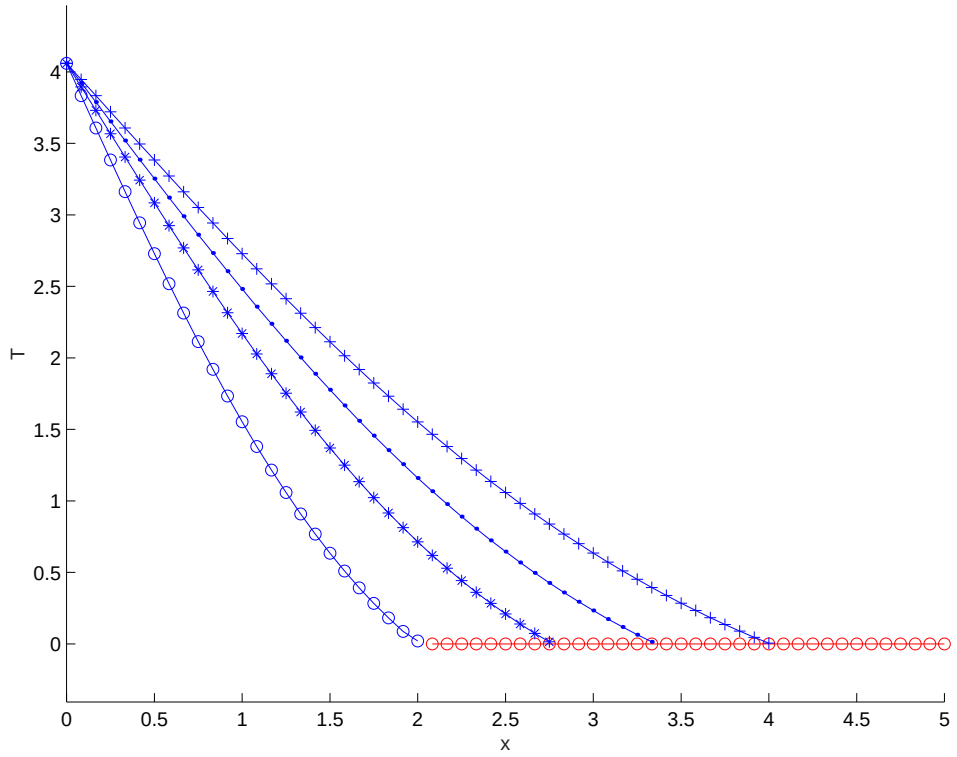

Figure 2.16: Exact solution for flame propagation problem - Numerical solution of the temperature profile's evolution (open symbol) on top of the exact solution (solid line).

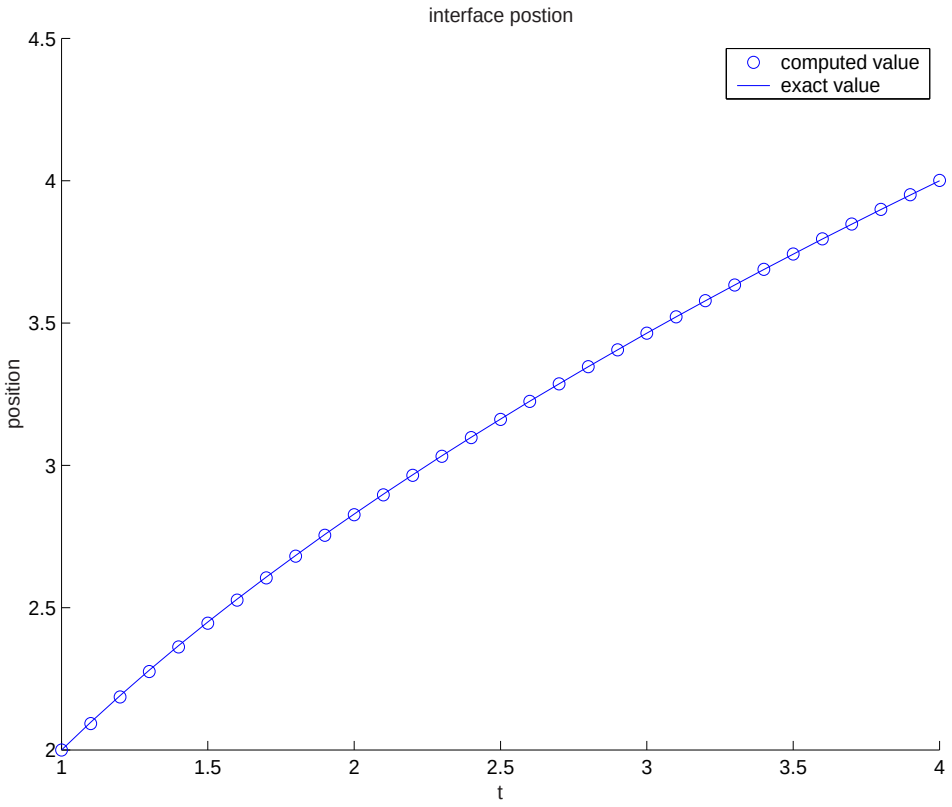

Figure 2.17: Exact solution for flame propagation problem - Numerical solution of the interface position (open symbol) on top of the exact solution (solid line).

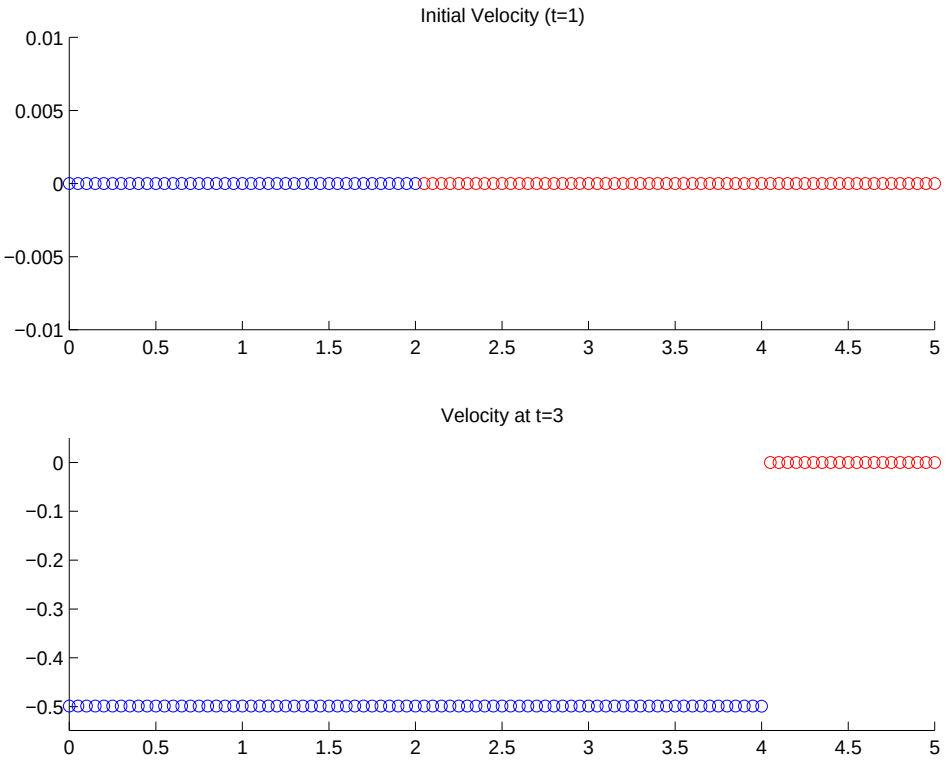

Figure 2.18: Initial velocity ($t = 1$ - top) and final velocity ($t = 3$ - bottom).

2.6 Conclusion

In summary, we have shown that our scheme is second order accurate for the Poisson equation, the heat equation and the Stefan problem in the case where the exact interface position is known. Our method yields symmetric matrix which is more efficient in terms of cpu time. Future work on this topic will explore the possibility of obtaining a second accurate scheme in the case where the exact interface position is not known.

References

[1] M.C. Bartelt and J.W. Evans, Phys. Rev. B **46**, 12 675 (1992).

[2] J.G. Amar and F. Family, Phys. Rev. Lett. **74**, 2066 (1995).

[3] C. Ratsch, A. Zangwill, P. Šmilauer, and D.D Vvedensky, Phys. Rev. Lett. **72**, 3194 (1994).

[4] C. Ratsch, M.F. Gyure, S. Chen, M. Kang, and D.D. Vvedensky, Phys. Rev. B **61**, R10 598 (2000).

[5] J.A. Venables, Philos. Mag. **27**, 693 (1973); J.A. Venables and D. J. Ball, Proc. R. Soc. London, Ser. A **322**, 331 (1971).

[6] S. Stoyanov and D. Kashchiev, Curr. Top. Mater. Sci. **7**, 69 (1981); J.A. Venables, G.D.T. Spiller, and M. Hanbücken, Rep Prog. Phys. **47**, 399 (1984).

[7] Y.W. Mo, J. Kleiner, M.B. Webb, and M.G. Lagally, Phys. Rev. Lett. **66**, 1998 (1991).

[8] J.A. Stroscio, D.T. Pierce, and R.A. Dragoset, Phys. Rev. Lett. **70**, 3615 (1993).

[9] C. Ratsch, P. Šmilauer, A. Zangwill, and D.D Vvedensky, Surf. Sci. **329**, L599 (1995).

[10] G.S. Bales and D.C Chrzan, Phys. Rev. B **50**, 6057 (1994).

[11] M.C. Bartelt and J.W. Evans, Phys. Rev. B **54**, R17 359 (1996).

[12] M.C. Bartelt, A.K. Schmid, J.W. Evans, and R.Q. Hwang Phys. Rev. Lett. **81**, 1901 (1998).

[13] M.C. Bartelt, C.R. Stoldt, C.J. Jenks, P.A. Thiel, and J.W. Evans, Phys. Rev. B **59**, 3125 (1999).

[14] Blackman and Wilding, Europhys. Lett. 16 (1), 115 (1991).

[15] S. Osher and J.A Sethian, J. Comput. Phys. **79**, 12 (1988).

[16] M.F Gyure, C. Ratsch, B. Merriman, R.E Caflisch, S. Osher, J.J. Zinck, and D.D. Vvedensky, Phys. Rev. E **58**, R6927 (1998).

[17] S. Chen textitet al., J. Comput. Phys., (to be published).

[18] J.G. Amar, F. Family, and P-M. Lam, Phys. Rev. B **50**, 8781 (1994).

[19] P.A. Mulheran and J.A. Blackman, Phys. Rev. B **53**, 10 261 (1996).

[20] Liu, X.-D., Fedkiw, R.P and Kang, M., *A Boundary Condition Capturing Method for Poisson's Equation on Irregular Domains*, J. Comput. Phys. 154, 151-178 (2000).

[21] Fedkiw, R., Aslam, T., Merriman, B., and Osher, S., *A Non-Oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (The Ghost Fluid Method)*, J. Comput. Phys. 152 (2), 457-492 (1999).

[22] Kang, M., Fedkiw, R., and Liu, X.-D., *A Boundary Condition Capturing Method for Multiphase Incompressible Flow*, J. Comput. Phys. (in review).

[23] Nguyen, D., Fedkiw, R. and Kang, M., *A Boundary Condition Capturing Method for Incompressible Flame Discontinuities*, J. Comput. Phys. (in review).

[24] Golub, G. and Van Loan, C., *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1989.

[25] Osher, S. and Sethian, J.A., *Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*, J. of Comput. Phys. 79, 12-49 (1988).

[26] Chen, S., Merriman, B., Osher, S. and Smereka, P., *A Simple Level Set Method for Solving Stefan Problems*, J. Comput. Phys. 135, 8-29 (1997).

[27] Sethian, J.A. and Strain, J., *Crystal Growth and Dendritic Solidification*, J. Comput. Phys. 98, 231-253 (1992).

[28] Kim, Y.-T., Goldenfeld, N. and Dantzig, J., *Computation of Dendritic Microstructures using a Level Set Method*, Physical Review E 62, 2471-2474 (2000).

[29] Udaykumar, H.S., Mittal, R. and Shyy, W., *Computation of Solid-Liquid Phase Fronts in the Sharp Interface Limit on Fixed Grids*, J. Comput. Phys. 153, 535-574 (1999).

[30] Juric, D. and Tryggvason, G., *A Front-Tracking Method for Dendritic Solidification*, J. Comput. Phys. 123, 127-148 (1996).

[31] Peskin, C., *Numerical Analysis of Blood Flow in the Heart*, J. Comput. Phys. 25, 220-252 (1977).

[32] Johansen, H. and Colella, P., *A Cartesian Grid Embedded Boundary Method for Poisson's Equation on Irregular Domains*, J. Comput. Phys. 147, 60-85 (1998).

[33] Johansen, H., *Cartesian Grid Embedded Boundary Finite Difference Methods for Elliptic and Parabolic Differential Equations on Irregular Domains*, Ph.D. Thesis, University of California, Berkeley, CA 1997.

[34] Schmidt, A., *Computation of Three Dimensional Dendrites with Finite Elements*, J. Comput. Phys. 125, 293-312 (1996).

[35] Fedkiw, R., Aslam, T., and Xu, S., *The Ghost Fluid Method for Deflagration and Detonation Discontinuities*, J. Comput. Phys. 154, 393-427 (1999).

[36] Atkins, P., *Physical Chemistry*, 5th edition, Freeman, 1994.

[37] Sussman, M., Smereka, P. and Osher, S., *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys. 114, 146-154 (1994).

[38] Fedkiw. R., *A Symmetric Spatial Discretization for Implicit Time Discretization of Stefan Type Problems*, (unpublished) June 1998.

[39] Adalsteinsson, D. and Sethian, J.A., *The Fast Construction of Extension Velocities in Level Set Methods*, J. Comput. Phys. 148, 2-22 (1999).

[40] Sethian, J.A., *Fast Marching Methods*, SIAM Review 41, 199-235 (1999).

[41] Shu, C.W. and Osher, S., *Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes*, J. Comput. Phys. 77, 439-471 (1988).

[42] Fedkiw, R., Marquina, A., and Merriman, B., *An Isobaric Fix for the Overheating Problem in Multimaterial Compressible Flows*, J. Comput. Phys. 148, 545-578 (1999).

[43] Almgren, R., *Variational Algorithms and Pattern Formation in Dendritic Solidification*, J. Comput. Phys. 106, 337-354 (1993).

[44] Li, Z., *A Fast Iterative Algorithm for Elliptic Interface Problems*, SIAM J. Numer. Anal. 35, 230-254 (1998).