Phase-Ordering Kinetics on Curved Surfaces

by

Oliver Lars Schoenborn

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Graduate Department of Physics
University of Toronto

© Copyright by Oliver Lars Schoenborn 1998
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.
A grand-papa Léo
Un doux zéphir
Un gazouillis d’hirondelle
Et une pomme et demie
Abstract

I investigate phase-ordering kinetics on static curved surfaces, starting from a well-known time-dependent Ginzburg-Landau equation, known as model A and valid in flat two-dimensional systems, and generalizing this to apply on curved surfaces. I develop and implement an interface formalism for model A, valid in both curved and flat surfaces. This is based on an interface velocity equation explicitly showing how interface motion couples to local surface geometry. I discuss extensively both theoretical and numerical aspects of this formalism. I derive a coupled set of curvature equations and use them to obtain an approximate expression for the curvature autocorrelation function (CAF) in the flat case. This is compared for the first time to numerical simulation results and shows that the CAF provides dynamical information not readily available from the traditional order-parameter structure-factor, yet is far easier to compute than the latter. A dominant length-scale is observed for the first time, in the domain interface undulations, even in Euclidean model A dynamics. I discuss how this affects the interpretation of what is needed for a system to exhibit dynamical scaling.

I look at the effect of surface Gauss curvature on the growth rate of domains and show that when the phase-ordering occurs on a corrugated surface, metastable long-range disorder may result. I show how these effects cause a break-down of dynamical scaling and power-law growth, how they bring about the elimination of the zero-temperature fixed point of Euclidean model A, and how phase-ordering in curved lipid-bilayer membranes should be affected. A new very-late stage regime appears for simulations of model A on sinusoid (i.e. egg-carton-like) surfaces. These features indicate that thermal noise should be included in future studies of phase-
ordering kinetics on curved surfaces. They also indicate that even *before* the order-parameter is explicitly coupled to surface quantities such as the local mean curvature of the surface (found in other work in this field), the dynamics already exhibits characteristics not found in Euclidean model A, such as logarithmically-slow growth reminiscent of random-field Ising model kinetics.
Acknowledgments

I thank my supervisor Rashmi C. Desai for his support during the course of this work. I am indebted to Madan Rao for his role in bringing to my attention the research which started this thesis work. I thank Chuck Yeung, Mohamed Laradji and François Léonard for helpful discussions, Mike Walker and Wasantha Wijesundera for giving me unlimited access to their computer cycles, and the National Science and Engineering Research Council (NSERC), the “Fond Canadien pour l’Aide à la Recherche” (FCAR), the Sumner Fellowship Foundation, and the University of Toronto School of Graduate Studies for their financial support in the first part of this work. I thank the developers of ARPACK, Celeste Sagui and Alex Lifshits for their help regarding various computational issues in the early stages of this work. Thanks to Wolfgang Koehler for his C++ “graphix” library, and to Paul Dierckx for his spline routines FITPACK.

Last but not least, I would like to thank my family and friends for having made this experience unforgettable, always being there in the ups as much as the downs, for all the lunches and dinners in Chinatown, for the philosophical discussions on Technology, for lemon pies with whipped-cream, for groovy bass-laden music, for humous and blueberry scones, “Hey good cooking” and wonderful etalks, for blues jamming and CD’s, for hikes in Allegheny National Forest, for the Spanish moss by the Rio Grande, for “theories of everything”, for amazing “Georgian” style dinners, for the Coyote’s howlings, musings and wisdom, for letting me say “it’s only at the next street light” so many times, for over 50 Monday and Wednesday afternoons with wonderful people, for “mium-mium” and “bonbons a patates”, for baked Camembert and good wine, for Serendipity, Little Bear and Humming Bird. God bless you all.
# Contents

Abstract iii

Acknowledgments v

List of Figures viii

1 Introduction 1

1.1 Time-dependent Ginzburg-Landau equation 5

1.2 Pattern-formation on curved geometries 13

1.3 Overview of thesis 15

2 Interface formalism in flat systems 17

2.1 Bulk formalism: limitations and alternatives 17

2.2 Interface formalism 21

2.3 Numerical method of integration 29

2.4 Results and discussion 38

2.5 Summary 47

3 Generalizations for curved spaces 49

3.1 TDGL equation 49

3.2 Interface equation 56

3.2.1 Geodesic interface curvature 58

3.2.2 Velocity vs curvature (NEIV) 61

3.2.3 Discussion 63
## CONTENTS

3.3 Numerical methods .................................................. 69  
3.3.1 TDGL on a curved surface ...................................... 70  
3.3.2 Interface formalism ............................................. 73  
3.4 Numerical test of the NEIV equation ............................ 76  
3.5 Measurement of order in curved space ............................ 81  
3.5.1 Lambda Structure Factor, or $\lambda$SF ........................ 81  
3.5.2 Curvature correlations .......................................... 87  
3.6 Summary .............................................................. 90  

4 Applications ............................................................ 92  
4.1 Domain growth vs surface curvature .............................. 93  
4.2 Long-range disorder and activated hopping ...................... 97  
4.3 Egg-carton and corrugated surfaces ............................... 104  
4.3.1 Numerical simulations .......................................... 105  
4.3.2 Geodesic curvature correlations ............................... 109  
4.3.3 Interface density .............................................. 112  
4.3.4 Total curvature correlations ................................... 116  
4.4 Summary .............................................................. 119  

5 Conclusions and extensions .......................................... 121  

A Discrete Laplace-Beltrami operator .................................. 130  
A.1 “Nearest-neighbor” discretization ................................ 130  
A.2 Isotropic discretization ........................................... 132  

B Implicit numerical integration ....................................... 136  

C Interface generation .................................................. 141  

Bibliography ............................................................. 146
# List of Figures

1.1 Typical path in phase-diagram following a quench .......................... 3
1.2 Snapshots of typical relaxational pattern-formation .......................... 4
1.3 Local free-energy density above and below $T_c$ .............................. 7

2.1 The constant and normal gauges ............................................... 23
2.2 An interface in the plane ....................................................... 24
2.3 Typical initial interface configuration (flat system) ........................ 30
2.4 All interfaces for 40 runs in flat system ...................................... 30
2.5 $R(t)$ for circular domain, theory and TDGL simulation ..................... 35
2.6 Typical interface evolving, in a flat system .................................. 36
2.7 $G^2_K(s,t)$ and $L(t)$ from bulk, flat system ................................ 39
2.8 $\Xi_K^2(x)$ for fig. 2.7 ......................................................... 40
2.9 Domain size vs dominant length ............................................... 42
2.10 Comparison of $a(t)$ with Euclidean numerical simulation .................. 46
2.11 Comparison of $G^1_K(s,t)$ from interface and bulk formalisms ........... 47

3.1 Special coordinate system for interface on surface ............................ 57
3.2 Torus surface and its parametrization ......................................... 71
3.3 Band domain on torus ......................................................... 76
3.4 Test of the NEIV equation on torus with TDGL ................................ 78
3.5 Test of the NEIV equation on torus with interface formalism ................ 79
3.6 $K_g R$ vs $R$ for circular domain on one-bump surface ....................... 80
3.7 Some eigenfunctions of the LB operator on the torus ......................... 85
3.8 Rescaled $\lambda$SF for torus runs; inset, Porod tail ............................ 86
# List of Figures

3.9  
$L(t)$ for flat and torus surfaces from $\lambda$SF .......................... 87

3.10  
$L(t)$ from $G_{K_s}^2(s, t)$ for torus runs .............................. 90

4.1  
Two ovoid domains on torus ............................................. 95

4.2  
$L(t)$ for ovoid domains on torus ....................................... 95

4.3  
$dL/dt$ vs $1/L$ in regions of different $K_{G}$ on torus .................. 96

4.4  
Surface consisting of two bumps, 3D view ............................. 98

4.5  
Side view of surface and projected interface .......................... 99

4.6  
Top projectional view for fig. 4.5 ..................................... 100

4.7  
Test of hopping condition: interface length vs time .................... 101

4.8  
Top projectional view for interface around four bumps ................. 102

4.9  
Unit cell of sinusoid surface ........................................... 106

4.10 Interfaces at earliest time on sinusoid surface, $\lambda = 20$ .......... 107

4.11 Initial interface configuration on sinusoid surface, $\lambda = 40$ .... 108

4.12 Rescaled $G_{K_s}(s, t)$ for sinusoid surface, $A = 4, \lambda = 20$ .... 110

4.13 Zoom on small-s region of fig. 4.12 ................................ 111

4.14 Variance of Gaussian part of $G_{K_s}(s, t)$ vs time .................. 112

4.15 Interface density vs time for sinusoid surface ....................... 114

4.16 Interface configuration at very late stage, sinusoid surface ......... 116

4.17 $G_{K}(s, t)$ for the sinusoid surface $A = 4, \lambda = 20$ .............. 117

4.18 $G_{K}(s, t)$ for the bimodal sinusoid surface .......................... 118

4.19 $G_{R}(s, t)$ for the sinusoid surface $A = 4, \lambda = 20$ .............. 117

A.1  
Interface when $\nabla_{LB}^2$ in TDGL is improperly discretized ............. 132

A.2  
Site-dependencies for discrete LB operator .......................... 134
Chapter 1

Introduction

A very important class of problems in Physics is that of pattern-formation, i.e. those involving the evolution of a disordered system toward a state displaying spatial or temporal structure, or both. Examples of such systems abound in Biology, where segmentation in arthropods, large-scale coherent motion in groups of birds, insects, or fish, electric waves in the heart and brain, and camouflage in mammals and reptiles are but a few of the better known ones. Non-biological examples are also numerous, such as Rayleigh-Bénard convection, Belousov-Zhabotinsky reactions, dislocation patterns and laser-induced melting[1], flame fronts, viscous fingering, dendritic growth, phase-separation in binary alloys and phase-ordering in ferromagnets. Not only do pattern-formation problems deal with some of the most fundamental questions in the Sciences (morphogenesis in living organisms), but many of them have important industrial implications, in non-linear optics, materials science[1] and pharmacology[2].

The ultimate goal of any pattern-formation study is to explain not only the final state (if any) of a system, but the processes that lead to that state, and to characterize those processes in such a way that unifying “principles" of pattern-formation may be inferred, leading to a common understanding of a large class of systems with a small number of principles. Pattern-formation is inherently non-equilibrium. As a consequence, many of the well-established thermodynamics concepts, such as free energy, entropy, temperature, must be used with a great deal of care and in some cases may not be used at all. Pattern-formation is also highly nonlinear so that the
equations used to model such systems can rarely be solved analytically. These are the greatest difficulties in modelling and understanding pattern-formation problems.

Despite the incredible diversity of patterns seen in nature and the even greater variety of systems which exhibit pattern-formation, some understanding has been achieved over the last century. Probably the most important realization has been that systems which are completely different from a microscopic point of view, in composition as much as forces involved and scale, can have qualitatively the same dynamics of pattern-formation. I.e., the details of the microscopic interactions between elementary particles of the system are not important. Rather, it is the cumulative interactions on a macroscopic scale which determine what characteristics the system dynamics have. Therefore a relatively small number of equations is sufficient to describe a stupendous variety of pattern-formation problems.

A useful way to quantify the presence or absence of large-scale structures in a system is to define an order-parameter $\phi$ which can vary in both space and time. It could be for instance the difference in local concentration of two types of molecules in a binary fluid, or the local magnetization in a ferrous alloy. The different variables characterizing a given system, such as concentrations, local pressure, momentum, energy, etc, can change in time at vastly differing rates. Variables that change much faster than the order-parameter act as thermal noise, while those which change much slower can in some cases be ignored or slaved to the order-parameter. Hence the equations used in this class of problem are often written in the form of stochastic partial differential equations relating spatial and temporal variations of the order-parameter and in some cases other slow variables. In this respect, field-theoretic models have been very popular. I describe them in a little more detail later.

Hohenberg and Halperin in the late 70's proposed a classification for several types of pattern-forming dynamics for which a field-theoretic description existed[3]. One of the classes is that of dissipative dynamics for one non-conserved order-parameter. Consider a system in a homogeneous, equilibrium state, and suddenly push it out of equilibrium into a region of its phase-diagram where at least two thermodynamic states coexist. This could be done by varying the temperature suddenly, as in a
Figure 1.1: A typical quench that takes a system initially in equilibrium at time $t = 0^-$ to an unstable region of the phase-diagram where two thermodynamic states ($\phi_+$ and $\phi_-$) are possible. Certain regions of the system relax to $\phi_+$, others to $\phi_-$. Long-range order appears in the form of large-scale domains of each phase.

quench, pictured in fig. 1.1. Due to the coexistence of two degenerate equilibrium phases different from the initial phase, and the presence of long-range fluctuations inherent to the fast modes, certain regions of the system relax to one state while other regions relax to the other, leading to the formation of a multitude of domains of each phase. This can be seen in fig. 1.2, where $\phi_+$ is white and $\phi_-$ is black. The domains are separated from one another by sharp interfaces, i.e. sharp on the length-scale of the domains but smooth and of finite-width on the length-scale of the molecules of the system. In the central part of the interfaces the system is out of equilibrium. The system therefore seeks to decrease the amount of interface through interface motion. The way it does this is usually the most challenging aspect of modelling
Figure 1.2: Typical relaxational phase-ordering system (also flat two-dimensional model A, eq. (1.4)). From left to right and top to bottom, times are $t = 0^+, 1, 6, 12, 25, 50, 100$ and 200. Color is order-parameter $\phi$: $\phi_+$ is white and $\phi_-$ is black. The first picture is a random configuration of very small values of $\phi$.

pattern-formation systems.

One type of system exhibiting non-conserved relaxational dynamics in the above conditions is the anisotropic Ising ferromagnet, i.e. an idealized ferromagnet in which
1.1. **TIME-DEPENDENT GINZBURG-LANDAU EQUATION**

The magnetization of any molecule can take only two values, either +1 (up) or -1 (down). Hohenberg and Halperin labelled the prototypical field-theoretic model used for this kind of problem as model A: one order-parameter, non-conserved, with relaxational dynamics. It is indeed the simplest of pattern-forming systems. In model A dynamics, there is no constraint on the average order-parameter per unit area at any time, it can increase or decrease as the patterns evolve. This is not so in the more complex model B, for instance, whose only difference from model A is a conservation law for the order-parameter, constraining the average to remain constant at all times. An example of such a system is a phase-separating binary alloy in a closed box. Since particles can be neither created nor destroyed, the average order-parameter cannot change. This imposes severe restrictions on interfacial motion, making the dynamics more complicated to model and understand. There is no such restriction in model A systems. More complex systems than model B exist but are not discussed in this thesis.

### 1.1 Time-dependent Ginzb urg-Landau equation

Field-theoretic models such as model A are based on a phenomenological extension of equilibrium concepts valid in thermodynamic systems. One starts by considering the *microscopic* state \{S_i\} of a set of N particles in a system of size L, and the system's Hamiltonian \(\mathcal{H}\{S_i\}\). For argument's sake let me assume \(S_i\) is the spin of a particle at a site labelled \(i\) in the system. The partition function of the system is then \(Z = \sum\{S_i\} \exp(-\mathcal{H}\{S_i\}/k_BT)\) and its Helmholtz free-energy \(\mathcal{F} \equiv -k_BT\ln Z\), where \(k_B\) is Boltzmann's constant and \(T\) the temperature. However \(\mathcal{F}\) is not well-defined for a system strongly out-of-equilibrium, such as is the case during a relaxational process. One way around this serious difficulty is to break the system up into cells of size \(\ell \ll L\). Denote the average spin on cell \(j\) by \(c_j \equiv \sum_{i=1,N} S_i/N\). The size \(\ell\) is taken much larger than the average distance \(a\) between particles of the system, so each cell contains a thermodynamically large number of particles. It is also taken much smaller than the thermal correlation length of the spin variable. Then I can
assume each cell is internally in thermal equilibrium but that \( c_i \) can vary slowly from one cell to the next, i.e. neighboring cells can be slightly out of equilibrium with one another. This allows me to use thermodynamic concepts like the Helmholtz free-energy, entropy, pressure *etc within* each cell, and the well known thermodynamic relationships between them. Now a continuum limit can be taken, so that particles become points and cell \( j \) becomes an infinitesimal area around point \( x \), and space becomes continuous rather than discrete. The system variables have become, in essence, continuous *fields* whose values vary from cell to cell (point to point). The order-parameter \( \phi \) is \( c_j \), and the Helmholtz free-energy of the spin system becomes a *functional* \( \mathcal{F}\{\phi(x)\} \) of the order-parameter field \( \phi(x) \).

Even in simple cases the Hamiltonian is too complicated for the partition function \( Z \) to be calculated analytically, and some approximation must be used to obtain \( \mathcal{F} \). I express \( \mathcal{F} \) as the sum of all local contributions \( V(\phi(x)) \), where \( V(\phi(x)) \) gives the free-energy of each cell based on the value of \( \phi \) for that cell, plus a sum of all contributions coming from gradients of the order-parameter between cells. In the continuum limit this is written

\[
\mathcal{F}\{\phi(x)\} = \int d\bar{x} \left( V(\phi(\bar{x})) + \frac{\xi^2}{2} |\nabla \phi|^2 \right)
\]  

(1.1)

where the constant \( \xi \) is the interface width length-scale. The local free-energy density of each cell is assumed to be expandable in a power-series of the order-parameter, *à la* Ginzburg-Landau, and only the lowest order terms are kept.[4]

The necessary and sufficient conditions for obtaining two coexisting equilibrium states is that \( V(\phi) \) be single-welled in the initial region of the phase-diagram \( t = 0^- \) in fig. 1.1) and double-welled in the post-quench region (i.e. \( t = 0^+ \)). Such a local free-energy density is sketched in fig. 1.3. It is customary to use what is called a \( \phi^4 \) form for \( V \):

\[
V(\phi) \equiv -\frac{a}{2} \phi^2 + \frac{b}{4} \phi^4
\]  

(1.2)

where \( a \) is negative for a one-state equilibrium and positive for degenerate two-state equilibrium (for fig. 1.3 it would be proportional to \( 1 - \frac{T}{T_c} \)), and \( b \) is strictly positive so as to insure the free energy remains finite. The dynamical equation for the order-
Figure 1.3: Local free-energy density $V(\phi)$ above and below $T_c$, model A. $T_c$ is the critical temperature appearing in fig. 1.1.

Parameter is finally phenomenologically defined as

$$\frac{\partial \phi}{\partial t} = -M \frac{\delta F}{\delta \phi}$$

(1.3)

where $\frac{\delta}{\delta \phi}$ denotes a functional derivative and $M$ is a positive constant which determines the time-scale of the dynamics. This equation can be compared to the equation of motion for a purely dissipative system in Newtonian mechanics. There should be a random thermal noise term in addition on the RHS of eq. (1.3) to account for the fast modes, but renormalization group calculations show the noise to be irrelevant in Euclidean model A[5]. But most importantly, in this thesis I deal only with deep quenches (to temperature zero), so I do not further comment on it at this point. After substituting eq. (1.2) and (1.1) into eq. (1.3) and carrying out the functional
derivative, one has, as the dynamical equation for model A dynamics[4],

\[
M^{-1} \frac{\partial \phi}{\partial t} = a\phi - b\phi^3 + \xi^2 \nabla^2 \phi
\]  

(1.4)

All 4 constants can be eliminated by appropriate rescaling of time, space, and order-parameter, so that it is customary to set them simply to 1. I use \( a = b = 1 \) throughout this thesis, hence \( \phi_{\pm} = \pm 1 \). I keep \( M \) and \( \xi \) in some derivations, for clarity, but they can be assumed equal to 1 when absent.

Phase-ordering kinetics, in this thesis, refers to the type of dynamics described by this model A equation, eq. (1.4), also referred to as a time-dependent Ginzburg-Landau (TDGL) equation. Because model A is such a basic equation in pattern-formation problems, and constitutes one of the simplest models to give non-trivial relaxational dynamics, a large body of work exists, of experimental, theoretical and numerical nature, starting from the early 1960's. Scalar model A dynamics is a rare occurrence experimentally, while model A for a two or three-component (i.e. vector) order-parameter is far more common in nature. However all have proven useful in furthering our understanding of pattern-formation, especially in leading the way to some of the more complex systems. Bray has recently published a review of work in this field[5]. Since I am going to be concerned with scalar model A only, I recall some of the results relevant to this kind of system only.

I always assume that the initial state of the system is one of complete disorder, where the order-parameter \( \phi \) fluctuates around the mean value of 0 and its variance \( \langle \phi^2 \rangle \ll 1 \), as pictured in the first snapshot of fig. 1.2 on page 4 by the dark and light gray dots (gray represents \( \phi = 0 \)). In this case Model A dynamics shows 3 different time-regimes. The first, at very early times, is the linear regime since \( \phi \) is everywhere very small. This regime is characterized by exponential growth of long-wavelength fluctuations in the order-parameter, and exponential decay of short-wavelength fluctuations. This can be seen by keeping only the linear terms in eq. (1.4), in which case a Fourier mode \( \phi_k(t) \) is found to have an exponential time dependence \( \phi_k(t) = \phi_k(0) \exp(\omega_k t) \), with \( \omega_k = 1 - k^2 \) the linear dispersion relation. Therefore,
\( \omega_k < 0 \) only for \( k > 1 \), and all wavelengths smaller than \( 2\pi \) decay.

The second time-regime starts when the order-parameter fluctuations become significant on the scale of \( \phi_{\pm}[6] \). In this case the non-linear term in eq. (1.4) becomes important. Its action is to slow down the growth of the unstable modes and make the order-parameter saturate at one of its two equilibrium values \( \phi_{\pm} \). In fig. 1.2 this regime ends shortly after the \( t = 12 \) picture. By the end of this regime, a complicated set of convoluted domains has formed, separated by sharp interfaces. Some domains are percolating. Every point inside a domain is in local thermodynamic equilibrium, internally and with its neighbors, since \( dV/d\phi = 0 \) everywhere there and \( \phi \) is constant in a given domain. All wavelengths smaller than \( 2\pi \) have decayed during the early regime. Hence the interface width, as I show in the next paragraph, is on the order of \( 2\pi \).

The third time-regime is therefore governed by the slow motion of interfaces as the system seeks to minimize the quantity of costly interfacial energy. Note that the width of interfaces remains approximately constant throughout this regime. An approximate value for the width can be found by solving eq. (1.4) for a straight interface in a two-dimensional system, since the interface is then stationary and the problem one-dimensional:

\[
0 = \phi - \phi^3 + \xi^2 \frac{\partial^2 \phi}{\partial x^2}
\]  

(1.5)

which has

\[
\phi(x) = \tanh \left( \frac{x}{\xi \sqrt{2}} \right)
\]  

(1.6)

as solution. The width, defined as the distance over which the order-parameter varies appreciably, can be checked to be roughly 3 times \( \xi \sqrt{2} \), i.e. \( \sim 5\xi \). Note that this regime is also characterized by the presence of two types of domains, diskoid domains and large convoluted ones. Both types evolve very differently. Indeed, diskoid domains can only decrease their perimeter by shrinking in size and disappearing altogether. However when large convoluted domains decrease their perimeter they become less convoluted, i.e. they simply “dewiggle” themselves. Once completely “dewiggled” they are diskoidal and start shrinking in size. This is discussed further in chapters 3
and 4, but you can get a flavor for it by looking at the late-time snapshots in fig. 1.2. In a finite system, there is a fourth regime where domains are comparable in size to the system itself. In model A, this regime ends when one of the two phases has completely taken over the whole system. In an infinite system however, this regime is never reached and the system never manages to equilibrate.

Simulation results discussed in this thesis only pertain to the third regime, also referred to as the late-stage regime, where the domains are reordering at a fairly slow rate. There are several interesting aspects to the dynamics during this regime, but I focus on two of them here. First, one observes self-similar dynamics. This has been observed experimentally as well. Self-similar dynamics refers to the fact that system configurations at a time $t_1$ look statistically identical to configurations at an earlier time $t_0$, if they are rescaled lengthwise by a factor that depends on $t_1/t_0$. The only interpretation of such a finding is that all dynamical lengths in the system have the same time-dependence, so that they can all be expressed in terms of one arbitrarily chosen reference length-scale $L(t)$. The numerical value of $L$ is not important, only its time-dependence, which is the second most important characteristic of model A dynamics: $L(t) \sim t^{1/2}$. It is common to refer to $L$ as a dominant or typical length-scale in the dynamics, however this is not necessary for dynamical scaling and should always be verified for model A. To my knowledge this is done for the first time in chapter 2, but in the interest of rigor the real meaning of $L(t)$ as a reference length rather than a dominant one should be kept in mind.

One way to measure the degree of order in pattern-forming systems is through the two-point equal-time order-parameter correlation function,

$$G_\phi(\bar{x}, t) \equiv \langle \phi(\bar{x}_0, t)\phi(\bar{x}_0 - \bar{x}, t) \rangle$$

(1.7)

The $\langle \rangle$ denotes an ensemble average, i.e an average over all possible initial conditions, and an average over all possible $\bar{x}_0$. Experimentally, the order-parameter
structure factor is more easily measurable and is the Fourier-transform of $G_\phi(\vec{x},t)$:

$$S_\phi(k,t) \equiv \frac{1}{A} \left\langle \phi(k,t)\phi(-k,t) \right\rangle$$

(1.8)

where $A$ is the system area and $\hat{\phi}(\vec{k},t)$ is the Fourier transform of the order-parameter $\phi(\vec{x},t)$. Homogeneity and isotropy of the dynamics in model A implies $G_\phi(\vec{x},t)$ depends neither on $\vec{x}_0$ nor on the direction of $\vec{x}$, and $S_\phi(k,t)$ does not depend on the direction of $\vec{k}$. Therefore, both functions are axisymmetric so that a circular averaging over all modes of same $|k|$ or all $\vec{x}$ of same length $|\vec{x}|$ can be done to obtain one-dimensional functions.

The most important consequence of self-similar dynamics is that all $S_\phi(k,t)$ taken at different times in the late-time regime can be rescaled to fall on one, universal and time-independent curve. The same goes for $G_\phi(\vec{x},t)$. As mentioned earlier, any time-dependent length in the system dynamics can be used as our reference length $L(t)$, in terms of which all other lengths can be expressed. $L(t)$ can be defined, for instance, from the first or second moment of $S_\phi(k,t)$ or $G_\phi(\vec{x},t)$. The rescaled correlation function is a time-independent curve $\Xi(y)$, with $y \equiv x/L(t)$. Since all physical constants in eq. (1.4) can be scaled away, the function $\Xi(y)$ is a universal curve valid for all pattern-formation systems which fall under the category of model A dynamics.

There have been many attempts to derive $\Xi(y)$ and the power-law behavior of $L(t)$ from first principles applied to eq. (1.4). Some examples are singular perturbation[7], interfacial dynamics [8], and renormalization group[9]. Even the well-known OJK method[8], the first to derive $\Xi(y)$ from the TDGL equation, uses some approximations based on isotropy and randomness which make it difficult to ascertain whether or not the method derives dynamical scaling or if it is unknowingly put in by hand. When dynamical scaling is assumed to hold a priori, hand-waving arguments can be used to obtain the $t^{1/2}$ power-law for $L(t)$. I give a simple example below. The main limitation of these methods is that analytical solutions rely strongly on the assumptions of isotropy and randomness, which are not valid in all systems.
In the late-stage regime of model A dynamics, the bulk part of the domains is in equilibrium. An interface equation is therefore extractable from the model A equation. In the late 1970's, Allen and Cahn[10] derived such an interface equation for model A systems. Interfaces are approximated by one-dimensional curves of 0 thickness. This is a good approximation when the interfaces are gently curved. The speed \( \nu = |\vec{v}| \) of the interface normal to itself is found to depend only on the local curvature \( K \) of the interface. Expressed in vector form,

\[ \vec{v} = M\xi^2 \vec{K} \]  

(1.9)

since \( \vec{K} \) is also perpendicular to the interface and points in the direction of motion. This result is known in the literature as the Allen-Cahn interface equation, and is of crucial importance in chapter 2. Interestingly, Allen and Cahn also find, in the same article[10], that this result is valid in some experimental model A systems, but not in others.

From eq. (1.9), dimensional analysis immediately yields a \( t^{1/2} \) power-law for \( L(t) \). Indeed, \( \nu \) is a speed. The interface width, proportional to \( \xi \), is a constant in time. Since all dynamical lengths in the system are proportional to \( L(t) \), \( \nu \) should be proportional to \( dL/dt \). On the RHS, \( K \) is a curvature and therefore should vary as \( 1/L \). Putting this together,

\[ \frac{dL}{dt} \sim \frac{1}{L} \]  

(1.10)

so that straightforward integration yields \( L \sim t^{1/2} \). This is not rigorous insofar as dynamical scaling must be assumed.

Research in dissipative pattern-forming problems has mostly moved on to the more complex models B, C, E, and so on. These extensions add, for example, a conservation law (B), a second, auxiliary conserved field such as a density of impurities (C), and so on. However, all this work has naturally focused on Euclidean systems, i.e. phase-ordering on a line, in the plane or in three-dimensional systems. One area that has been attracting increasing attention over the past five years is that of pattern-formation in lipid-bilayer membranes, which form continuous, curved geometries.
1.2 Pattern-formation on curved geometries

Lipid bilayer membranes form the prime example of curved, two-dimensional systems where shape and internal degrees of freedom are believed to play an important role in biological processes, such as intramembrane protein diffusion, shape-change, shape instabilities, vesiculation, ion adsorption, polarization of lipid molecules, etc. In principle, other curved, "soft" systems such as surfactant solutions, liquid-crystals, Langmuir monolayers, and polymerized surfaces could display similar properties. An understanding of the interaction of shape and internal degrees of freedom (such as described by an order-parameter) of a curvable surface is, however, still lacking. Experimentally, this is due mostly to the very challenging difficulties which must be circumvented in order to probe the pattern-formation dynamics without adversely affecting the membrane shape or state. Theoretically the time-dependent curved geometry of the membranes poses considerable mathematical challenge.

It is believed that phase-separation plays an important role in budding and vesiculation in lipid bilayers. These processes involve a lipid membrane in the form of a large vesicle floating in an aqueous solution and changing shape locally to form a bud, which in some cases separates itself from the mother vesicle to form a second, smaller vesicle. Some experiments suggest vesiculation is facilitated by a phase-separation occuring on the membrane, possibly acting to concentrate interfacial tension at the neck of the bud. Most probing techniques rely on "capturing" the membrane into a frozen state. Not only is the dynamics halted, but it is difficult to assess how much damage this does to the membrane ordering. Another example is that of ripple and egg-carton-shaped membranes. There are some indications that such patterns could be explained via a pattern forming process that couples to the shape of the bilayer. The article by Gebhardt et al. is particularly interesting because it discusses such pattern formation processes in the context of first-order transitions and non-conserved order-parameters, from an experimental point of view.

Most of the work in the area of membrane shape-change where spatial structures
are present within the membrane itself has been limited to equilibrium models which assume the domains are already formed. They make use of one form or another of coupling between an intra-membrane order-parameter and some geometric quantity such as the local mean curvature of the membrane[19]. One result has been the creation of shape phase-diagrams[20]. Recently, some researchers have used powerful computer techniques to simulate shape-change in surfaces made of two types of lipids[21, 22].

Any dynamics occurring in curved spaces, such as seen in General Relativity, invariably introduces new and non-trivial concepts and effects which require considerable care, thought and exploration. In this way, before understanding what effects a coupling between local mean curvature and a time-varying order-parameter may have on the membrane state (i.e. its shape and internal structure), it should first be assessed what happens in the absence of any such coupling. Namely, a systematic, grounds-up study has yet to be carried out, to gradually evolve towards a complete description of a dynamical membrane on which relaxational pattern-formation is taking place. Rather than approach the problem from the angle of shape-change when the membrane is inhomogeneous, I choose to approach it from the opposite angle, i.e. how is pattern-formation affected when it occurs on a curved surface. This allows for the gradual complexification of the problem from the point of view of well-established experimental, numerical and theoretical results in flat model A and more complicated flat systems. How are such characteristics as self-similarity and power-law growth, known to be valid in Euclidean systems, affected when the underlying space is curved? Are there new processes that may affect the dynamics and change its universality class? Or is the concept of universality class still valid? Is a different approach necessary to deal with some of the theoretical and computational difficulties that curved geometry may introduce? These are some of the issues that I wish to consider in this thesis.
## 1.3 Overview of thesis

In this thesis, I investigate the dynamics of relaxational pattern-formation, of the type described by model A in the classification of Hohenberg and Halperin, when it occurs on a static curved surface. I show how, even on a static surface with no explicit coupling between the order-parameter and any surface geometric quantity, some very interesting differences can be seen from the dynamics in flat systems. In so doing, I also introduce several new measurements which pertain to curved systems and become necessary in this case. My original contributions are in chapters 2, 3 and 4.

I first discuss some limitations of the discretized TDGL equation which become important in numerical simulations of the non-Euclidean model A. A new formalism, based on the interface velocity equation, is introduced. This naturally leads to a new measure of order in the system, through the definition of a curvature autocorrelation function. Since model A has not previously been discussed from this point of view, I give and discuss results of Euclidean model A simulations using this new formalism. This is necessary to compare both the new approach to the bulk approach, and in later chapters the non-Euclidean to the Euclidean dynamics. It also provides new insight into Euclidean model A dynamics. A set of curvature equations is derived which is used to obtain an approximate expression for the curvature autocorrelation function. This is compared for the first time to numerical simulation results.

In chapter 3, I first discuss phase-ordering kinetics on general Riemannian manifolds. I derive a non-Euclidean model A (NEMA) equation and a central analytical result, the non-Euclidean interface velocity (NEIV) equation, which explicitly relates the interface dynamics to the local intrinsic geometry of the surface. I prove that the interfacial dynamics in model A occurs via the minimization of the length of interfaces on the non-Euclidean manifolds, an otherwise difficult-to-prove statement. After describing some of the algorithmic issues relevant to both non-Euclidean formalisms, I put the NEIV result to test via simulations which compare the accuracy of both numerical formalisms and provide some guidelines for obtaining adequate accuracy.
in the simulations in more complex surfaces. Finally, I discuss a generalization of the order-parameter structure factor to measure bulk order, the lambda structure factor, as well as of the curvature autocorrelation function to measure interface order, and compare the two via simulations on the torus manifold. Some of the contents of chapter 3 have already been published[23].

In chapter 4, I use the various tools discussed or developed in the first three chapters to explore some of the consequences of non-Euclidean space on two-dimensional phase-ordering kinetics. First I look at the effect of surface Gauss curvature on the growth rate of domains. This is done by measuring the time rate-of-change of the interface length of a circular domain in two different regions of the torus manifold. Following this I explore the phenomenon of metastable long-range disorder when the phase-ordering occurs on a corrugated surface. I do a simple calculation to show that metastable states occur only above a certain threshold value of the ratio of amplitude of surface undulations to wavelength of undulations, test this numerically, and discuss some implications for self-affine surfaces such as those found in lipid bilayer systems as well as some implications when thermal noise is present. Finally, I discuss some simulation results of model A dynamics on a corrugated surface of the sinusoid type. Notably, I discuss the breakdown of dynamical scaling and of power-law growth, through the geodesic and total curvature autocorrelation functions, which differ on curved surfaces. Those two functions show several features not found in Euclidean dynamics. I discuss this as well as several measures of the dominant dynamical length-scale.

In chapter 5 I summarize the results and discuss some of the many possible extensions of this work. In the three appendices I discuss some techniques that may come in handy for future work.
Chapter 2

Interface formalism in flat systems

In this chapter, I first discuss some limitations of the discretized time-dependent Ginzburg-Landau (TDGL) equation which become important in numerical simulations of the non-Euclidean model A (NEMA). A new formalism, based on the interface velocity equation (1.9), is introduced. This naturally leads to a new measure of order in the system, through the definition of a curvature autocorrelation function. Since model A has not been discussed from this point of view before, I give and discuss results of Euclidean model A simulations using this new formalism. This is necessary to compare both the new approach to the bulk approach, and in later chapters the non-Euclidean to the Euclidean dynamics. It also provides new insight into Euclidean model A dynamics.

2.1 Bulk formalism: limitations and alternatives

Studies of pattern-formation problems must often make use of numerical simulations to solve the dynamical equation(s) due to the non-linear and time-dependent nature of the problems. Model A is no exception. But numerical simulations require that the dynamical equation be discretized, posing new difficulties. For instance, if an Euler integration scheme is used to discretize eq. (1.4)[24], the integration time-step is limited from above by the square of the space-mesh used in the discretization of the model A equation. For Euclidean model A, the spatial discretization can be taken
large enough for the maximum time-step to be within manageable limits, without unduly forfeiting numerical accuracy. Along with its great algorithmic simplicity, Euler integration schemes have proven sufficient and satisfactory for research in model A dynamics. For the NEMA however, the dependency of the maximum time-step on spatial discretization is a severe limitation. Indeed, when a surface is discretized it is bound to have regions where the grid is fine and regions where it is relatively coarse. The checkerboard instability, notorious with Euler methods, develops in the regions of fine mesh if the time-step is not taken small enough. The smallest surface mesh therefore controls the maximum allowable time-step for an Euler integration.

A work-around that eliminates the dependency between the time-step and spatial discretization is to use an implicit integration method. Explicit methods, such as Euler, apply a (possibly non-linear) map to the state of the system at time \( t \) to obtain the state at time \( t + \Delta t \). Implicit methods do the reverse, i.e. they apply a (possibly non-linear) map to the state of the system at time \( t + \Delta t \) to obtain the state at time \( t \). Since the direction of integration is towards increasing \( t \), the map must be inverted. This usually involves inverting a matrix if the map is linear. When it is not, other tricks must be used, specific to the problem at hand. Implicit methods are substantially more involved and more difficult to implement than explicit methods, so that virtually no effort has been spent in devising implicit integration methods for model A and related models. It may as well be so, for solutions to the model A equation are characterized, as was explained in chapter 1, by large-scale order in the form of domains separated by sharp interfaces. Mathematically, this implies that there are at least two vastly different length-scales present in the solutions, making the equation stiff: the domain length-scale and the interface length-scale. This disparity of length-scales severely constrains the allowable time-steps of several implicit methods I tried (alternating-direction, split-operator, iteration, and combinations of these), re-introducing, ironically, a dependency on the spatial discretization. This caused, for example, configurations to freeze when clearly they should be evolving to the final homogeneous equilibrium state characteristic of model A dynamics. Since this thesis work was started, Kawakatsu et al.[25] have, apparently,
found a workable implicit method for a similar dynamical (Euclidean) system.

Another possible work-around is to use a finite element integration scheme. This involves defining a non-uniform surface grid which optimizes certain desired properties of the system surface. For the NEMA one such property is that the mesh be as uniform as possible, so that the distance between any two points is roughly constant. However such methods are sophisticated and even more difficult to implement than implicit methods, not only at the integration stage, but at the surface-discretization stage itself. Some of the pre-existing canned routines available in numerical methods archives were found to be in principle adaptable to the problem of NEMA, but this is clearly not the ideal solution.

The second drawback of a bulk formalism for model A arises from the presence of long-range order in the late stages of the dynamics. In this stage the inside of the domains is in local thermodynamic equilibrium and no longer changes in time. All the non-equilibrium dynamics is concentrated at the interfaces, which are one-dimensional on the scale of the domains. Therefore most of the computation time required to evolve the domains is wasted on evolving time-independent regions. Finally, bulk-measured quantities such as the order-parameter structure factor, discussed in chapter 3, require considerable amounts of computer resources to manipulate large amounts of data which contain no internal structure whatsoever: all the structure of the dynamics is once more at the interfaces, i.e., the structure is the interfaces, with the bulk being homogeneous and predictable.

In pattern-formation problems, the focus of research is on the macroscopic, long-range order, rather than on microscopic forces and structures. The reason should be clear from chapter 1. This is also why field theories are useful for this kind of problem. Similarly, in the TDGL equation the fine structure of domain interfaces is not of physical relevance. For instance, different forms of local free-energy density \( V(\phi) \) in eq. (3.1b) induce the same model A dynamics on macroscopic scales of interest, as long as \( V(\phi) \) satisfies the minimal qualitative constraints of symmetry and number of extrema. Differences appear only on the scale of the interface width, as in the interface profile, etc.
An interface formalism suffers none of these drawbacks. Indeed, interfaces are smoothly curved lines with no disparate length-scales. This eliminates the stiffness problem, allowing for more freedom in the choice of numerical integration methods. They are also one-dimensional. Hence far less information is needed to describe the order-parameter configuration. In model A, the quantity of interface decreases rapidly in time as $t^{-1/2}$. As the interfaces shrink in length, the number of data points needed decreases and points can be discarded. This contrasts with the bulk formalism, where the quantity of data remains constant even when there are no interfaces left! After the model A interfaces are formed, they are decoupled from the bulk and no longer interact among themselves, so that interfaces can be evolved individually if necessary. Also, interface points need not coincide with lattice points. i.e., an interface is discretized independently of the system discretization. This eliminates the dependency between integration time-step and system grid size even in explicit numerical integration methods. Moreover, an interface formalism dispenses with the microscopic and uninteresting interfacial structure. Finally, a measure of order in the system can be obtained through interfacial quantities only, therefore through one-dimensional measurements, further decreasing the necessary computation time. Eq. (1.9) is an obvious candidate for the basis of an interface formalism. It relates the interface velocity, itself the time-variation of the interface position, with the curvature of the interface, which is a local and easily computed quantity. Since such an interface formalism has not previously been implemented for model A, I do this in the rest of the chapter.

An unexpected advantage of an interface formalism for model A is that it gives some insight into the nature of "dominant length-scale", discussed briefly in chapter 1. By definition, a dominant length-scale should show up in measurements of the order of the system, such as the order-parameter structure factor. This is the case, for instance, in model B[26]. There, the peak in the structure factor translates into oscillations in the order-parameter autocorrelation function. However, none of these two characteristics are observed in model A, since the peak in the structure factor is at the origin, i.e. at infinite wavelengths. This hardly qualifies as a typical length-scale. I also discussed briefly in chapter 1 how a dominant length-scale was not necessary to
observe dynamical scaling. Therefore, talking of one dominant length-scale in model A is unjustified without more concrete evidence. An interface formalism allows for the easy computation of a different measure of order in the system, the *curvature* autocorrelation function. This function is extremely costly to obtain from *bulk* configurations, but trivial to compute from the interfaces themselves. The curvature autocorrelation function provides a means to test the existence of a dominant length-scale in model A dynamics.

One inevitable drawback of interface formalisms is that they cannot describe how interfaces form from the small order-parameter fluctuations present just after the quench. Therefore, the initial interface configurations must be obtained independently. The easiest way to do this is of course to use the bulk formalism to evolve the order-parameter for only the relatively short time necessary for interfaces to become established, and then extract the interfaces from the resulting configurations. Even for short integration times, this suffers from the drawbacks mentioned at the beginning of this section, drawbacks which become severe for the NEMA. I give one alternative in appendix C, which has not been tested yet. Therefore, in this thesis initial interface configurations are always obtained by integrating the appropriate bulk equation (model A or NEMA) from \( t = 0^+ \) to \( t = 17 \) (unless otherwise specified), when the domains have fully formed and interfaces can be extracted. A full-length run takes the configuration to \( t = 500 \) or more.

### 2.2 Interface formalism

Interface formalisms have been implemented for somewhat different systems before. Notable are works by Brower *et al.*[27], Ben-Jacob *et al.*[28] and Weeks and van Saarloos[29] on the problem of dendritic growth in crystals and solidification fronts in general. These systems are the opposite of model A systems, in that interfaces are unstable and grow in time rather than shrink. Some of the concepts developed there were further adapted to model B dynamics, by Tim Rogers[30] in his Doctoral thesis. The main emphasis in these papers and Rogers' thesis was on deriving a *curvature*
2.2. INTERFACE FORMALISM

equation for the interfaces, rather than using the Allen-Cahn (AC) or some other interface velocity equation itself. This was useful because the shape of an interface is translation- and rotation-invariant. I.e., the curvature as a function of arclength is sufficient to uniquely define the shape of a two-dimensional curve. For reasons that become clear later, I use instead the AC equation as the basis for an interface formalism.

In differential geometry, the concept of gauge is very important. A gauge is a particular choice of parametrization of a manifold, whether that manifold is a line, a surface or a higher dimensional “surface”. Differential geometry allows for the characterization of manifolds in a gauge-invariant way, i.e. measurements such as curvature, orientation correlation functions, gradients of a field on the manifold, and so on, give identical results independent of the chosen parametrization. However, expressions for these measurements can be greatly simplified through the choice of an appropriate gauge, which usually depends on the particularities of the problem at hand.

When a manifold changes in time, such as a curved line representing an interface, a time-dependent gauge is desirable. For a deforming curved line, two natural gauges present themselves: the constant gauge and the normal gauge. The constant gauge is, not surprisingly, constant in time. It is usually chosen such that one unit of distance of the parameter of the curve corresponds to one unit of arclength, independent of time. The normal gauge on the other hand changes so as to make points of the interface move only normal to the interface. A circular interface in model A shrinks in time. The motion of interface points in both gauges is depicted in fig. 2.1. From this picture, it is clear why the velocity of interface points has a non-zero tangential component in the constant gauge. Only the normal gauge has the tangential component 0 at all times. The parameter of the gauge is $\alpha$ for the normal gauge, and arclength $s$ for the constant gauge. Assume that at time $t = 0$, the real interface length $L$ is equal to the parametric length $L_0$ (i.e. the parameter $\alpha$ varies from $\alpha = 0$ to $\alpha = L_0$). In the constant gauge the density of points remains constant and $L_0(t) = L(t)$ at all times. In the normal gauge, $L_0$ is constant, so the density of points $L_0/L$ changes in time.
The number of points decreases in time in the constant gauge, while it is constant in the normal gauge. Analytically, the normal gauge is advantageous for model A since the Allen-Cahn equation is implicitly written in this gauge. Numerically, its greatest advantage is that the number of points remains constant. Therefore I use the normal gauge at all times in this thesis. However, results are written in terms of arclength rather than $\alpha$, whenever possible, since this is the physical gauge.

I use $u$ and $v$ instead of the customary $x$ and $y$ to denote the flat cartesian coordinate system. $v$ must not be confused with the norm of the velocity vector, which has the symbol $\nu$. A point $P$ of an interface $I$ is given by a vector $\vec{r}(\alpha) \equiv (u(\alpha), v(\alpha))$ relative to some arbitrary origin $O$, as shown in fig. 2.2. Derivatives of geometric quantities can be expressed either in terms of $\alpha$ or in terms of arclength. Since arclength is a measure of true distance along the interface, a derivative expressed in terms of arclength is independent of parametrization. I denote an arclength derivative by a prime ($'$). Any other derivative is explicitly written out, as for example $d/d\alpha$. The two are easily related by use of

$$ds = \sqrt{g} \, d\alpha. \quad (2.1)$$
2.2. INTERFACE FORMALISM

Figure 2.2: An interface $I$ in the plane. The plane coordinates are denoted $u$ and $v$. A point $P$ of the interface has parameter $\alpha$ and is given by a vector $\vec{R}$ (bold $R$ in figure). $\hat{\tau}$ and $\hat{n}$ are the tangent and normal vectors to $I$ at $P$.

This defines $g$, the jacobian of the transformation between the gauges $s$ and $\alpha$. It is also equal to the determinant of the metric tensor on the interface in the normal gauge, equal to the metric itself since a line is one-dimensional, i.e. the tensor is a scalar.

The unit tangent $\hat{\tau}$ and the unit normal $\hat{n}$ satisfy

\begin{align}
\hat{\tau} & \equiv \frac{d\vec{R}}{ds} = (u', v') \\
\hat{n} & = (-v', u')
\end{align} \tag{2.2a, 2.2b}

where eq. (2.2b) is obtained from the requirement that $\hat{\tau} \cdot \hat{n} = 0$. Eq. (2.2a) also
2.2. INTERFACE FORMALISM

implies

\[ u'^2 + v'^2 = 1. \]  \hfill (2.3)

I recall the Frenet equations which define \( \hat{\tau}, \hat{n} \) and the curvature \( \vec{K} = K\hat{n} \) of a line in the plane:

\[ \frac{d\hat{\tau}}{ds} = K\hat{n} \]  \hfill (2.4a)

\[ \frac{d\hat{n}}{ds} = -K\hat{\tau} \]  \hfill (2.4b)

so that the Allen-Cahn interface equation (1.9) is written, in the normal gauge,

\[ \left( \frac{\partial \vec{R}}{\partial t} \right)_{\alpha} = K\hat{n} \]  \hfill (2.5a)

\[ = \frac{\partial^2 \vec{R}}{\partial s^2} \]  \hfill (2.5b)

Eq. (2.5b) follows from eq. (2.4a), (2.2a) and (2.3). Though \( u \) and \( v \) appear to be decoupled, they are in fact coupled through the arclength, \( (\delta s)^2 = (\delta u)^2 + (\delta v)^2 \).

I now derive an alternate equation for the interface, in terms of curvature rather than position. It is a gauge-invariant derivation directly in the normal gauge, retaining the vector nature of the interface curvature whenever possible. The shape of a curve in the plane requires only two quantities to be completely defined, the curvature and a gauge. Since the normal gauge is time-dependent, an equation for the gauge, i.e. for \( g \), must also be derived. I first derive the equation for the metric of the normal gauge.

The metric is written in terms of \( \vec{R} \) as

\[ g(\alpha, t) \equiv \left| \frac{\partial \vec{R}}{\partial \alpha} \right|^2. \]  \hfill (2.6)

Taking the time derivative at constant \( \alpha \),

\[ \left( \frac{\partial g}{\partial t} \right)_{\alpha} = 2 \frac{\partial \vec{R}}{\partial \alpha} \cdot \left( \frac{\partial}{\partial t} \left( \frac{\partial \vec{R}}{\partial \alpha} \right) \right)_{\alpha} \]
\[2.2. \text{ INTERFACE FORMALISM}\]

\[= 2 \frac{\partial \vec{R}}{\partial \alpha} \cdot \frac{\partial}{\partial \alpha} \left( \frac{\partial \vec{R}}{\partial t} \right)_{\alpha}\]

\[= -2 \frac{\partial^2 \vec{R}}{\partial \alpha^2} \cdot \vec{v} \tag{2.7}\]

where the subscript \(\alpha\) denotes constant \(\alpha\). The last step uses

\[0 = \frac{\partial}{\partial \alpha} \left( \frac{\partial \vec{R}}{\partial \alpha} \cdot \vec{v} \right) = \frac{\partial^2 \vec{R}}{\partial \alpha^2} \cdot \vec{v} + \frac{\partial \vec{R}}{\partial \alpha} \cdot \frac{\partial \vec{v}}{\partial \alpha} \tag{2.8}\]

which is 0 at all times because \(\frac{\partial \vec{R}}{\partial \alpha} = \sqrt{g} \dot{\tau}\) is orthogonal to \(\vec{v}\) in the normal gauge. Exploiting the orthogonality of \(\dot{\tau}\) and \(\vec{v}\) comes in handy several times during the following derivation. By making use of

\[\alpha' \equiv d\alpha/ds = 1/\sqrt{g} \tag{2.9a}\]

\[\alpha'' \equiv d^2 \alpha/ds^2 = -\frac{1}{2g^2} \frac{\partial g}{\partial \alpha} \tag{2.9b}\]

the Laplacian in arclength \(s\) is written

\[\frac{\partial^2}{\partial s^2} = \frac{1}{g} \frac{\partial^2}{\partial \alpha^2} + \alpha'' \frac{\partial}{\partial \alpha} \tag{2.10}\]

so that from the definition \(\vec{K} \equiv d^2 \vec{R}/ds^2\)

\[\frac{\partial^2 \vec{R}}{\partial \alpha^2} = g \left( \vec{K} - \dot{\tau} \alpha'' \sqrt{g} \right). \tag{2.11}\]

Substituting eq. (2.11) in eq. (2.7) and rewriting the latter in terms of \(\sqrt{g}\) rather than \(g\) I finally obtain

\[\left( \frac{\partial \sqrt{g}}{\partial t} \right)_{\alpha} = -\sqrt{g} \vec{K} \cdot \vec{v} \tag{2.12}\]

Since the vector \(\vec{K}\) always points in the same direction as \(\vec{v}\) in model A (opposite to \(\vec{v}\) when interfaces are unstable such as for dendritic growth), \(\sqrt{g}\) always decreases in time, i.e. the interface always shrinks in length.
A similar derivation, but far lengthier, can be done for \((\partial K/\partial t)_{\alpha}\). The dot product of \(\hat{n}\) with eq. (2.11) yields

\[
\frac{\hat{n}}{g} \cdot \frac{\partial^2 \vec{R}}{\partial \alpha^2} = K. \tag{2.13}
\]

Applying the time derivative and using the chain rule,

\[
\left( \frac{\partial K}{\partial t} \right)_{\alpha} = \frac{\hat{n}}{g} \cdot \frac{\partial^2 \vec{R}}{\partial \alpha^2} - \frac{\hat{n}}{g^2} \cdot \frac{\partial^2 \vec{R}}{\partial \alpha^2} \left( \frac{\partial g}{\partial t} \right)_{\alpha} + \frac{1}{g} \left( \frac{\partial \hat{n}}{\partial t} \right)_{\alpha} \cdot \frac{\partial^2 \vec{R}}{\partial \alpha^2}. \tag{2.14}
\]

The second term is simplified via eq. (2.13) and eq. (2.12). The third one is simplified by using eq. (2.11) and noting that \(\hat{n}\) can only change in direction, hence \((\frac{\hat{n}}{\partial t}_{\alpha})\) is oriented along \(\hat{r}\). With those simplifications I get

\[
\left( \frac{\partial K}{\partial t} \right)_{\alpha} = \frac{\hat{n}}{g} \cdot \frac{\partial^2 \vec{v}}{\partial \alpha^2} + 2K \vec{K} \cdot \vec{v} - \sqrt{g} \alpha'' \hat{r} \cdot \left( \frac{\partial \hat{n}}{\partial t} \right)_{\alpha}. \tag{2.15}
\]

The second derivative of \(\vec{v}\) is

\[
\frac{\partial^2 \vec{v}}{\partial \alpha^2} = \frac{\partial^2}{\partial \alpha^2}(\nu \hat{n}) = \frac{\partial^2 \hat{n}}{\partial \alpha^2} \nu + 2 \frac{\partial \hat{n}}{\partial \alpha} \frac{\partial \nu}{\partial \alpha} + \hat{n} \frac{\partial^2 \nu}{\partial \alpha^2}. \tag{2.16}
\]

The second term of eq. (2.16) vanishes when dot-multiplied with \(\hat{n}\) since \(\partial \hat{n}/\partial \alpha = -\sqrt{g} K \hat{r}\). The first term is simplified by writing the second derivative of \(\hat{n}\) in terms of arclength via eq. (2.1), applying the chain rule and using both Frenet equations. This yields

\[
\frac{\partial^2 \hat{n}}{\partial \alpha^2} = -\sqrt{g} \hat{r} \frac{\partial}{\partial s}(K \sqrt{g}) - g K \vec{K}, \tag{2.17}
\]

so that I have

\[
\left( \frac{\partial K}{\partial t} \right)_{\alpha} = \frac{1}{g} \frac{\partial^2 \nu}{\partial \alpha^2} + K \vec{K} \cdot \vec{v} - \sqrt{g} \alpha'' \hat{r} \cdot \left( \frac{\partial \hat{n}}{\partial t} \right)_{\alpha}. \tag{2.18}
\]

Finally I show that \(\hat{r} \cdot (\frac{\partial \hat{n}}{\partial t}_{\alpha})\) in the last term of eq. (2.18) satisfies

\[
\hat{r} \cdot \left( \frac{\partial \hat{n}}{\partial t} \right)_{\alpha} = -\frac{\partial \nu}{\partial s}. \tag{2.19}
\]
2.2. INTERFACE FORMALISM

It is easiest to show this by deriving $\frac{\partial \nu}{\partial s}$. The norm of the velocity is

$$\nu = \hat{n} \cdot \left( \frac{\partial \tilde{R}}{\partial t} \right)_\alpha$$

Taking the arclength derivative and using the chain rule, I have

$$\frac{\partial \nu}{\partial s} = \frac{\hat{n}}{\sqrt{g}} \cdot \frac{\partial}{\partial \alpha} \left( \frac{\partial \tilde{R}}{\partial t} \right)_\alpha + \left( \frac{\partial \tilde{R}}{\partial t} \right)_\alpha \cdot \frac{\partial \hat{n}}{\partial s}$$

$$= \frac{\hat{n}}{\sqrt{g}} \cdot \left( \frac{\partial}{\partial t} \left( \frac{\partial \tilde{R}}{\partial \alpha} \right) \right)_\alpha$$

(2.20)

(2.21)

since $\tilde{v}$ is orthogonal to $\frac{\partial \alpha}{\partial s}$. Eq. (2.2a) and the chain rule yield

$$\frac{\partial \nu}{\partial s} = \hat{n} \cdot \left( \frac{\partial \tilde{t}}{\partial t} \right)_\alpha$$

(2.22)

(2.23)

The second step was achieved by using $d(\tilde{t} - \hat{n})/ds = 0$, completing the proof.

Thus, substituting eq. (2.19) into eq. (2.18) with eq. (2.10), I have the following set of curvature equations describing the evolution of interfaces in flat model A systems:

$$\left( \frac{\partial K}{\partial t} \right)_\alpha = \frac{\partial^2 \nu}{\partial s^2} + K \tilde{K} \cdot \tilde{v}$$

(2.24a)

$$\left( \frac{\partial \sqrt{g}}{\partial t} \right)_\alpha = -\sqrt{g} \tilde{K} \cdot \tilde{v}$$

(2.24b)

$$\tilde{v} = M \xi^2 \tilde{K}$$

(2.24c)

where $M$ and $\xi^2$ are the coefficients found in eq. (3.1) and $\nu \equiv |\tilde{v}|$. Eq. (2.24) will be of use in section 2.4 when I introduce and discuss the curvature autocorrelation function.

Note that eq. (2.24a) and eq. (2.24b) are purely geometric, the physics of model A being solely contained in the AC equation, eq. (2.24c). The first term on the RHS of eq. (2.24a) is diffusive and causes all modes of $K$ to dissipate, except the zeroth
mode which is of constant amplitude. The diffusive term thus seeks to shrink the interface into a straight line or a perfect circle. Due to the AC equation, the second term on the RHS of eq. (2.24a) is cubic in \( K \). It seeks to increase the curvature and dominates when the diffusive term is negligible, i.e. for circular domains. Hence it causes circular domains to shrink in radius, as is indeed the case in model A. Broadly speaking, the diffusive term of eq. (2.24a) dominates during the early stages of interface motion, when interfaces have just formed and are very convoluted, while the cubic term dominates only in interfaces which have become circular. However, this was found to be only approximately true.

2.3 Numerical method of integration

I now present the numerical simulation method used to evolve model A interfaces. Three computer programs written in C++ are referred to in the following sections. The first, genus1, is the program that integrates the bulk TDGL equation. The second program, geninterf, is used to extract interfaces from a bulk order-parameter configuration produced by genus1. The third program, IFinteg, takes an interface as extracted by geninterf (or created by hand) and evolves it in time according to eq. (2.5a), using a simple Euler method. An implicit method is described in appendix B. The initial interfaces are created by genus1 by evolving a random order-parameter configuration from \( t = 0^+ \) to \( t = 17 \). Each interface is a set of \((u, v)\) pairs one unit of arclength apart. Most interfaces form closed loops but due to the system's periodic boundary conditions, many are large enough to be percolating from one side of the system to the next. I refer to these as open. A typical set of interfaces extracted from a bulk configuration is shown in fig. 2.3. In this picture each point is separated from its two neighbors by a distance of a unit of arclength.

Translational invariance of interfaces in flat systems implies that interfaces are equally likely to be found at any point of the system. This can be seen in fig. 2.4 where the interfaces of all 40 runs are plotted on the same graph. A typical set of 40 runs generates between 300 and 800 interfaces, for a total interface length
Figure 2.3: A typical initial configuration of interfaces in one system.

Figure 2.4: All interfaces for 40 runs for the same system as fig. 2.3 (though at a later time, \( t = 67 \) instead of 17).
2.3. **NUMERICAL METHOD OF INTEGRATION**

between 25,000 and 100,000 units, depending on the system size used. System sizes of 100 × 100, 200 × 200 and 40π × 80π were used, as discussed later. As mentioned in the opening section of this chapter, model A interfaces no longer interact once separated by distances larger than the interface width, because the interface length can only decrease. Therefore an interface can be evolved with eq. (2.5a) independently from all others.

A discretized interface in the normal gauge is a line of particles which can only move perpendicularly to the interface, therefore neighboring points can never merge. If an interface has N points indexed 1 to N, the index of a point can be used as the α parameter. Neighboring points are then separated by a parametric distance Δα = 1. The metric at site i is approximated by the square of the (arc-)distance between sites i and i + 1. In the present section, any quantity Q(α) written as Q(i) implicitly indicates Q at site i corresponding to parameter value α. The metric is required for computing the partial derivatives with respect to arclength that enter the calculation of curvature K and ñ. From eq. (2.4a) and (2.2a) the curvature K is computed as

\[ K = u'u'' - v'u'' \]  

(2.25)

As usual the prime superscript denotes differentiation with respect to *arclength*, and u and v are coordinates of the interface. The u' at site i of the interface is discretized, following the Euler method, as

\[ u'(i) = \frac{1}{\sqrt{g(i)}} \frac{\partial}{\partial \alpha} u(i) = \frac{1}{\sqrt{g(i-1)} + \sqrt{g(i)}} (u(i+1) - u(i-1)) \]  

(2.26)

The second derivative of u and v are discretized as

\[ u''(i) = \frac{1}{\sqrt{g(i)}} \frac{\partial}{\partial \alpha} \left( \frac{1}{\sqrt{g(i)}} \frac{\partial}{\partial \alpha} u(i) \right) \]  

(2.27a)

\[ = \frac{2}{\sqrt{g(i-1)} + \sqrt{g(i)}} \left( \frac{u(i+1) - u(i)}{\sqrt{g(i)}} - \frac{u(i) - u(i-1)}{\sqrt{g(i-1)}} \right) \]  

(2.27b)

Note that, for clarity, time is not explicitly mentioned but should obviously be as-
sumed. Expressions for $v'$ and $v''$ follow the same discretization.

The timestep is always taken so as to avoid the well-known checkerboard instability common to Euler methods. With the normal gauge, the distance between neighboring points is not constant along the interface after a few integration time-steps have been taken, as evidenced by eq. (2.24b). With inhomogeneous meshes, the checkerboard instability appears first in the region which has the smallest mesh and propagates from there. Therefore, I take

$$\Delta t_{\text{max}} \equiv \frac{1}{2}(\sqrt{g}_{\text{min}})^2. \quad (2.28)$$

The time step used for a given interface is 0.4 of its $\Delta t_{\text{max}}$.

Once the curvature $K$ is obtained, a forward Euler step consists of updating all points of an interface via

$$u(i, t + \Delta t) = u(i, t) - \Delta t K(i, t)v'(i, t) \quad (2.29a)$$
$$v(i, t + \Delta t) = v(i, t) + \Delta t K(i, t)u'(i, t) \quad (2.29b)$$

After the Euler step, a new $\Delta t$ is calculated based on the new smallest meshsize, to be used at the next Euler step. Therefore, each interface has its own $\Delta t = 0.4\Delta t_{\text{max}}$. In the simulations done, $\Delta t$ was found to vary between 0.2 and 0.001.

An important aspect of this Euler integration is the decreasing time step as points along the interface move closer together. At the beginning of the simulation, the interface is represented by a homogeneous mesh, i.e. it has $\sqrt{g}(i) = 1$ for all $i$. After some time, there can be considerable disparity in the distances between points. This constrains the time step to unnecessarily small values. Therefore whenever the shortest meshsize of an interface crosses some threshold, the interface is remeshed via a 4-point polynomial interpolation found in [24], optimized for the current application. A threshold of 0.5 was found to work well: it keeps low the number of interpolations (which inevitably introduce roundoff errors) needed during a run, but is sufficiently large that distance between points are not overly spread-out and the timestep can
remain substantial. The interpolation algorithm is a very efficient routine that walks along the interface at regular intervals of 1 and interpolates the value of $\hat{R}$ at those points. Remeshing therefore resets all $\sqrt{g}(i)$ to 1. One complete interpolation of an interface requires approximately as many operations as a few Euler integration steps.

The periodic boundary conditions of the system imply interfaces can exit one side of the system and enter through the other, as can be seen in fig. 2.3. This implies a discontinuous jump in $\hat{R}$, an undesirable condition when computing $K$. Rather than do this, geninterf assumes the system is replicated *ad infinitum* along $u$ and $v$. This allows closed interfaces to be void of jumps. However, curves of open topology invariably have their end point some integer number of system sizes away from their starting point. Both topologies are therefore essentially different: closed interfaces can shrink to a point and disappear, whereas open interfaces have a discontinuity that only allows them to straighten but never disappear.

The "origin" of an interface is site $i = 1$. For a closed interface, the integration algorithm handles the boundary sites $i = 1$ and $i = N$ naturally since $\hat{R}(1), \hat{R}(2), \hat{R}(N - 1)$ and $\hat{R}(N)$ are neighbors in real space. For open interfaces, the integration algorithm must be tricked into thinking the interface is a line of infinite length. Denote by $\hat{R}_{\text{jump}}$ the vector $aL_x\hat{x} + bL_y\hat{y}$ where $a$ and $b$ are non-zero integers specifying how many multiples of the system are between the origin and end of an open interface. If site $i - 1$ ($i + 1$) is called left (right) neighbor of site $i$, the trick is to use $\hat{R}(N) - \hat{R}_{\text{jump}}$ as the left neighbor of site 1, and $\hat{R}(1) + \hat{R}_{\text{jump}}$ as the right neighbor of site $N$. Note that $\hat{R}_{\text{jump}}$ is never equal to $\Delta \hat{R} = \hat{R}(N) - \hat{R}(1)$ but rather always a little larger than that.

The interface curvature equations (2.24), on the other hand, are not bothered by system boundaries, since only $K(s)$ is needed. The topological difference between open and closed interfaces is only visible through the value of

$$\int_C K \sqrt{g} \, d\alpha = n2\pi$$

(2.30)

where the integral is over one interface, and $n = 0, \pm 1$. Open interfaces have $n = 0,$
closed interfaces have $n = \pm 1$ (true in flat systems only).

The main disadvantage of eq. (2.24) over eq. (2.5a) or (2.5b) is that $K(s)$ is not sufficient to completely describe the shape of an interface when it is on a curved surface. I.e., the positions of points are needed as well, as I show in chapter 3. It is far better to compute $\tilde{K}$ from $\tilde{R}$ rather than *vice versa*, since $\tilde{R}$ can be obtained from $\tilde{K}(s)$ only via a piecewise walk along the interface, implying a cumulative error in $\tilde{R}$. Even in a flat system, the cumulative error leads to mismatch of endpoints in $\tilde{R}$. Another difficulty with eq. (2.24) is that a separate equation is required for the orientation of the interface, not given by $K(s)$. In flat model A systems orientation is not needed, but this is not so for interfaces on curved surfaces.

The numerical simulation of a circular domain of radius $R$ can be compared with the analytical solution to the Allen-Cahn (AC) equation. For a circular domain, $\hat{n}(\alpha, t)$ does not change in time. The AC interface equation therefore simplifies to

$$\frac{dR}{dt} = \frac{M \xi^2}{R}$$

which has the solution

$$R_{th}(t) = \sqrt{R_0^2 - 2M\xi^2 t}$$

where $R_0 \equiv R(t = 0)$ and the subscript *th* indicates theoretical. The average radius $R_n(t) \equiv 1/\langle K(t) \rangle$ as obtained from a numerical simulation with IFinteg, for $M = \xi = 1$ and $R_0 = 60$, is plotted in fig. 2.5 and can be compared with $R_{th}(t)$. The error, obtained from the standard deviation of $K$ along the circular interface, varies around 0.04%, too small to be seen. Linear regression gives $R_n(t)^2 = (3600.42 \pm 0.08)-(1.99987 \pm 0.00007)t$, in excellent agreement with the theoretical result (0.01% error), even at very small $R$.

Two snapshots of the interface shape, the curvature $K(i)$ and the distance between neighboring points $\sqrt{g(i)}$, are shown in fig. 2.6 where the largest of the 400 or so interfaces of the 40 runs is displayed using IFinteg, at two different times. The top window shows the interface at the earliest time, with $\sqrt{g} = 1$ everywhere. The bottom window is at much later time, when curvature has everywhere decreased, the shape
2.3. NUMERICAL METHOD OF INTEGRATION

Figure 2.5: $R_n(t) = 1/(K(t))$ and $R_{th}(t)$ for circular domain. Dashed line is linear regression. Error bars too small to be seen.

roughened, and $\sqrt{g}$ is no longer homogeneous. Several remeshings were automatically performed by IFInteg between the two snapshots.

When only the interfaces are used to describe the state of the system, rather than the order-parameter itself, order in the system is most easily measured via a curvature autocorrelation function instead of the usual order-parameter structure factor. The generic definition is

$$G_K(s,t) \equiv \langle \bar{K}(0,t) \cdot \bar{K}(s,t) \rangle$$

(2.33)

where the angle brackets denote ensemble average over interfaces. To my knowledge, prior to the present work $G_K(s,t)$ has only been discussed in Tim Rogers' thesis and never measured in numerical simulations.

Numerically there are two ways of computing $G_K(s,t)$. The first uses all interface
2.3. NUMERICAL METHOD OF INTEGRATION

Figure 2.6: One interface at 2 different times (top window and bottom window), following integration of the AC equation by \texttt{Integ} (note however that window name is "Integ"). Top left sub-window shows curvature, bottom left is meshlength, right plot is $R$.

points of all interfaces independently. I denote it by $G_k^l(s, t)$,

$$G_k^l(s, t) \equiv \frac{1}{N(s, t)} \sum_{i=1}^{N_f(t)} \sum_{j,k} K_i(j, t) K_i(k, t)$$  \hspace{1cm} (2.34)

where $N_f(t)$ is the number of interfaces that are at least of length $2s$ (indicated by the prime superscript) while $N(s, t)$ is the total number of points of all those interfaces. $n_i$ is the number of points on interface $i$ (hence $N(s, t) = \sum_{i=1}^{N_f(t)} n_i(t)$). $K_i(j, t)$ is the
curvature on interface $i$ at some point $j$ at time $t$. The sum over $j, k$ is over all pairs of sites separated by a distance $s = |s(j) - s(k)|$ and belonging to the same interface. Of course this requires $\sqrt{g}$ to have the same value for all points of all interfaces. Eq. (2.34) is more suitable for analytical calculations.

The second definition gives smaller statistical error in numerical simulations. I denote it by $G^2_K(s, t)$,

$$G^2_K(s, t) \equiv \frac{1}{N(s, t)} \sum_{i=1}^{N_i} n_i \left( \frac{\sum_{j,k} K_i(s_j, t) K_i(s_k, t)}{\sum_{j=1}^{n_i} K_i(s_j)^2} \right)$$

(2.35)

All symbols are as in $G^1_K(s, t)$. The only difference is that $G^2_K(s, t)$ is a weighted average of the normalized $G^1_K(s, t)$ for each individual interface, giving more importance to longer interfaces. Also, $G^2_K(s, t) < 1$ for all $s$, except for $s = 0$ where it always equals 1. $G^2_K(s, t)$ can be interpreted as the average relative value of curvature a distance $s$ on either side of a point of curvature $K$.

I must remark on the $K(i)$ appearing in the definitions of $G^1_K(s, t)$ and $G^2_K(s, t)$. Eq. (2.33) uses the scalar product of the curvature vectors. But the $K(i)$ in eq. (2.34) and (2.35) are signed scalars. This is because eq. (2.34) and (2.35) are easily used on curved manifolds, while the forms involving scalar products, despite yielding smaller error bars in the Euclidean case, are not (at present). Note that the definition which uses a scalar product should give the correct dominant length-scale (if any) even when the domains are circular, while the scalar definition does not. The sign of $K$ need not be consistent for different interfaces as only products of $K$ within the same interface are sensible. Therefore IFinteg chooses the sign of the curvature for the first point of an interface as positive. This automatically set the sign of $K$ everywhere else on the interface.

Simulations were done for systems of size $100 \times 100$, $200 \times 200$ and $40\pi \times 80\pi$. The latter size is relevant to simulations in chapters 3 and 4. Of course, as long as the finite-size regime is avoided, results are independent of system size. Forty random order-parameter configurations were generated and evolved to $t = 17$ with genus1. The interfaces were then evolved with IFinteg after having been extracted.
2.4 RESULTS AND DISCUSSION

with geninterf, from \( t = 17 \) to \( t = 1000 \). The configurations were also evolved to \( t = 1000 \) with genus1, so that the interface formalism could be compared to the bulk formalism. The mesh size used was 1, and the time step for the bulk integrations 0.03 with an Euler method as described in chapter 3. Typical computation time required to evolve over 400 interfaces (total length over 80000) from \( t = 17 \) to \( t = 1000 \) was 1.3 hours on an HP735, as opposed to 8 hours for the bulk equations. This factor of 6 gain increases dramatically for curved surfaces in chapters 3 and 4.

2.4 Results and discussion

The top graph of fig. 2.7 shows a plot of \( G_K^2(s, t) \) at 6 different times during the late-stage regime, as obtained after integration of eq. (1.4) with genus1. I use the first root of \( G_K^2(s, t) \) to define the reference length-scale \( L(t) \) relevant to late-stage dynamics in Euclidean model A. The error bars, not shown for clarity, are 0 at the origin and linearly increase to 0.02 in the vicinity of the minimum, then further increase linearly afterwards as \( s \) increases though at a much slower rate. The error was defined by making an analogy between the curvature \( K \) and magnetization \( m \) of one-dimensional Ising magnets of different lengths[31]. The error for distance \( s \) is then the weighted average of the deviation of each magnet's value of \( \langle m(0)m(s) \rangle \) from the value of \( \langle m(0)m(s) \rangle \) for the ensemble of magnets. \( G_K^2(s, t) \) corresponds to \( \langle m(0)m(s) \rangle \). This was deemed the most reasonable method of error calculation, given the values of \( K \) along an interface, and therefore the statistical error in \( K \), are correlated. The bottom graph of fig. 2.7 is a log-log plot of \( L \) as a function of time. Fig. 2.8 (top) shows the six \( G_K^2(s, t) \) rescaled with \( L(t) \), i.e. \( \Xi_K^2(x), x \equiv s/L(t) \). The bottom graph is \( \ln \Xi_K^2(x) \) vs \( x^2 \), for \( x < 0.5 \).

The salient features of these four plots are the negative autocorrelation for \( s \approx 1.5L(t) \), the power law for \( L(t) \), the perfect scaling of the \( G_K^2(s, t) \) on length-scales up to twice \( L(t) \), the Gaussian form of \( G_K^2(s, t) \) for distances much smaller than \( L(t) \) (up to about \( L(t)/2 \)), and finally the constancy in time (within error) of the minimum value of \( G_K^2(s, t) \). Note that using a vector curvature with a scalar product in \( G_K^2(s, t) \),
Figure 2.7: Top: $G_k^2(s, t)$ from bulk configurations at 6 different times. Bottom: log plot of $L(t)$ as obtained from first root of $G_k^2(s, t)$. Linear regression gives $L \sim t^z$, $z = 0.48 \pm 0.01$. 
Figure 2.8: $\Xi_k^2(x)$ for top graph of fig. 2.7, $x \equiv s/L(t)$. Bottom shows how the curvature autocorrelation is Gaussian up to length-scales of roughly $L(t)/2$. 
as the original eq. (2.33), results in smaller error bars (by factors of 3 to 4), slightly stronger correlation near the dip, as well as a flatter tail.

The perfect dynamical scaling indicates that the number of runs and system size used give an accurate and reliable measure of $G_K^2(s,t)$, and that the curvature autocorrelation function correctly captures this very important characteristic of model A dynamics. The power law in $L(t)$ is $0.48\pm0.01$, very close to the theoretical prediction of $1/2$ discussed in chapter 1. The discrete bulk integration on the other hand gives a power-law of $0.45\pm0.02$. $G_K^2(s,t)$ therefore not only captures to a high accuracy the power-law behavior of model A dynamics, but performs better than the discrete bulk integration.

The most interesting feature of the curvature autocorrelation function is undoubtedly the relatively large negative autocorrelation apparent at distances $s \approx 1.5L(t)$. The dip indicates that if the curvature is known to have some value at point $s_1$ on the interface, a distance $L$ away positive and negative values of $K$ are equally likely, while at distances $1.5L$ away the curvature has on average a norm roughly 0.15 of the value at $s_1$ and an opposite sign to it. This also indicates that there is a dominant wavelength for undulations of the interfaces as a function of arclength. Indeed a Fourier transform (not shown) of $G_K^2(s,t)$ exhibits a well-defined peak at a non-zero value of $q \equiv 2\pi/s$. One could have expected the curvature structure factor to have a peak at 0 wavenumber, as is the case for the order-parameter structure factor. However, it appears that curvature measurements are sensitive to information to which the order-parameter itself is not.

The dominant length-scale for model A dynamics thus seems to have a different nature than, for instance, that of model B dynamics. In the literature on model A and B dynamics one loosely speaks of this dominant length-scale as an average domain size. However the concept of domain size is well defined only when the domains are morphologically disconnected or, if not, if they have a well-defined width. Model B domains satisfy the latter, but model A dynamics following a quench through the critical point satisfies neither. Only sufficiently off-critical quenches (see fig. 1.1) create domains whose bubble morphology lends itself to the definition of a “typical”
domain size. However, the curvature autocorrelation function suggests a dominant length-scale is present in the undulations of the interface rather than the size of the domains. The difference is schematized in fig. 2.9. For sufficiently off-critical quenches, the dominant length-scale and the dominant curvature radius should be roughly equal. The presence of a dominant undulation wavelength implies that referring to interfaces as "random" may be incorrect. One expects that truly random interfaces have a Gaussian curvature autocorrelation function rather than the one found here. Several analytical methods developed to derive the scaling function for the order-parameter structure factor make use of Gaussian assumptions about the order-parameter field as well as the randomness of the interfaces[32]. The distribution of curvatures is, however, perfectly Gaussian.
The Gaussian form of $G_{\kappa}(s, t)$ at small $s/L$ (i.e. up to roughly $1/2$) merits further investigation. In his Ph.D. thesis, Tim Rogers derives an approximate expression for $G_{\kappa}(s, t)$ for (flat) model B dynamics via eq. (2.24a). The result is a curvature autocorrelation which is a perfect Gaussian. This was never tested numerically. A similar derivation can be done for the model A curvature autocorrelation and compared with the numerical results of this section.

Consider the model A curvature equations:

\[
\left( \frac{\partial K}{\partial t} \right) = \frac{\partial^2 K}{\partial s^2} + K^3 \tag{2.36a}
\]
\[
\left( \frac{\partial \sqrt{g}}{\partial t} \right) = -\sqrt{g} K^2 \tag{2.36b}
\]

Now let me make the following mean-field approximation:

\[
\left( \frac{\partial \sqrt{g}}{\partial t} \right) \approx -\sqrt{g} h(t) \tag{2.37a}
\]
\[
h(t) \equiv \frac{1}{L} \int K^2 \, ds \tag{2.37b}
\]

where $L$ is the length of the interface. This approximation becomes exact for circular domains. Furthermore, neglect the cubic term of eq. (2.36a). This term is dominant for circular domains. For convoluted domains, numerical testing indicates $K^3$ is comparable to the diffusion term for as many as half the interface points. Therefore, the two approximations work in opposite directions, one becoming exact for circular domains, the other getting better for convoluted domains. On short length-scales, however, convoluted domains are locally circular, but the $K^3$ term should be negligible since convoluted domains see their curvature decrease rather than increase. Eq. (2.36a) then becomes

\[
\left( \frac{\partial K}{\partial t} \right) = \frac{1}{g(t)} \frac{\partial^2 K}{\partial \alpha^2} \tag{2.38}
\]

Going to Fourier space and making use of a change of variable for time, $g(t)dt' = dt$,
2.4. RESULTS AND DISCUSSION

the integration can now be performed, and the curvature structure factor \( \chi_q \) obtained:

\[
\chi_q(t') \equiv \frac{1}{N} K_q(t') K_{-q}(t') = \frac{K_q(0)^2}{N} e^{-2q^2t'}
\]  

(2.39)

where \( N \) is the number of points on the interface and \( q \) is the wavenumber in the reciprocal space of \( \alpha \). Assuming all \( K_q \) have equal amplitude at \( t = 0 \), a backwards Fourier transform yields

\[
G^1_K(s, t)(\alpha, t') = \sqrt{\frac{\pi}{8t'}} e^{-\alpha^2/8t'}
\]  

(2.40)

Now \( t' \) must be found as a function of \( t \). This can be done by noting that \( h(t) \) is equal to \( G^1_K(s, t)(0, t'(t)) \), i.e. \( h(t) = \sqrt{\frac{\pi}{8t'}} \). Also,

\[
t'(t) = \int_0^t \frac{dt}{g(t)}
\]

The equation for the metric is therefore

\[
\left( \frac{\partial g}{\partial t} \right)_\alpha = -2g \sqrt{\frac{\pi}{8}} \left( \int_0^t \frac{dt}{g(t)} \right)^{-1/2}
\]  

(2.41)

This has \( g(t) = t^{-1}/\pi \) as solution, so that \( t'(t) = \pi t^2/2 \). From eq. (2.1), \( \alpha = s/\sqrt{g} = s\sqrt{\pi t} \). Substituting in eq. (2.40),

\[
G^1_K(s, t) = \frac{1}{2t} e^{-s^2/4t}.
\]  

(2.42)

This is a Gaussian whose width increases as \( t^{1/2} \). Its amplitude, which has units of \( 1/\text{length}^2 \), decreases as \( 1/t \). These time-dependencies are consistent with power-law growth and dynamical scaling in model A dynamics. At small distances, the Gaussian form of \( G_K^2(s, t) \) can be approximated by a \( 1 - As^2 \), \( A > 0 \). Porod has shown that such negative \( s^2 \) dependence is characteristic of fluctuating systems without sharp variations[33]. A plot of \( K(s) \), as shown in fig. 2.6, indeed looks very much like a snapshot of a one-dimensional fluctuating membrane, smooth and without any sharp
variations. In appendix C, this interpretation becomes useful in implementing an algorithm to generate "random" interfaces with the correct curvature autocorrelation function.

Of course, the two approximations that allowed for the calculation of $G_K(s, t)$ are reasonable only on short length-scales, so it is not surprising that eq. (2.42) does not capture the dominant wavelength of undulations of the interfaces. Also, all $K_q(0)$ were assumed equal. This is obviously wrong, since even at the earliest times the curvature structure factor (the Fourier transform of the $G_K(s, t)$) shows a well-defined peak at a non-zero $q$ mode. If the correct $\chi_q(0)$ is used, then the analytical $G_K(s, t)$ will have a dip at least at early times. Therefore the strongest approximation may yet lie in the $\chi_q(0)$ rather than the mean-field and linearization approximations, though this seems unlikely. Note that the linearization approximation can be eliminated by combining eq. (2.36a) and (2.36b) into

$$
\left( \frac{\partial (\sqrt{g}K)}{\partial t} \right) = \sqrt{g} \frac{\partial^2 K}{\partial s^2}
$$

(2.43)

This is obtained by multiplying eq. (2.36a) by $\sqrt{g}$ and eq. (2.36b) by $K$ and adding the resulting equations. With the mean-field approximation for $g$, this equation is identical in form to eq. (2.38), but with $K$ replaced by $K\sqrt{g}$. The problem arises when solving the PDE for $g$, eq. (2.41), as a power-law form for $g$ no longer works. It therefore seems that, if the cubic $K$ term is kept, an approximation of the form (2.37a) cannot be used.

I compare the form of eq. (2.42),

$$
a(t)e^{-\frac{t^2}{\alpha_0^2}},
$$

(2.44)

with that measured in numerical simulations. The predicted values from eq. (2.42) are therefore $a(t) = 1/2t$, $b(t) = 4t$. $G_K^1(s, t)$ and $G_K^2(s, t)$ were obtained by evolving the 400 or so interfaces with genus1 and 1Finteg, for both the 200 x 200 and 40π x 80π systems. Fig. 2.10 shows the value of $1/2a(t)$ as a function of time, for simulations with the bulk and interface formalisms. The value of $a(t)$ is obtained from $G_K^1(s = 0, t)$.
2.4. RESULTS AND DISCUSSION

Linear regression gives \( 1/2a(t) \simeq (1.1 \pm 0.1)t \). Computing \( b(t) \) can be done by linear regression of \( -\ln(G_k^i(s, t)/G_k^i(s = 0, t)) \) vs \( s^2 \) (with \( i = 1, 2 \)), in which case \( b(t) \) is \( 1/\text{slope} \), or by linear regression of \( \log(-\ln(G_k^i(s, t)/G_k^i(s = 0, t))) \) vs \( \log(s) \), in which case \( b(t) \) is \( 10^{-y_0} \) where \( y_0 \) is the intercept obtained from linear regression. Both methods, with both \( G_k^1(s, t) \) and \( G_k^2(s, t) \) for bulk and interface formalisms, give altogether \( b(t) \simeq (1.7 \pm 0.2)t \), rather than \( 4t \). However, the value of \( L(t) \) plotted in fig. 2.7(b), where \( L \) is the value at which \( G_k^1(s, t) \) or \( G_k^2(s, t) \) first cross the \( x \)-axis, is found to satisfy altogether \( L^2 \simeq (4.0 \pm 0.1)t \). The prediction eq. (2.42) therefore seems to be more closely related to the envelope of \( G_k(s, t) \) rather than to its shape as such.

Before summarizing this chapter, I briefly compare \( G_k(s, t) \) for the bulk and interface simulations. \( \Xi_k^2(x) \) from IFinteg is identical to that in fig. 2.8 for distances \( s \lesssim 1.5L \), and within error bar otherwise. \( G_k^1(s, t) \) from IFinteg shows some differences with that obtained from genus1 and geninterf. Fig. 2.11 shows \( G_k^1(s, t) \) measured
Figure 2.11: Comparison of $G_k^1(s,t)$ as obtained from bulk integration (genus1) and interface formalism (IFinteg), at $t = 33$ and $t = 133$. at 2 different times, for both formalisms. Note that the value of $G_k^1(s,t)$ at $s = 0$ is equal to the variance of the curvature distribution for all interfaces present, $\langle K^2 \rangle$. The region around $s = 0$ and the width vary more slowly with the bulk simulations, in agreement with the different power-law behavior of $L(t)$ found earlier in the bulk and interface simulations. However, I have found that $G_k^1(s,t)$ is typically not as smooth as $G_k^2(s,t)$.

2.5 Summary

In this chapter I have explained several important drawbacks of bulk equations such as the model A equation, and discussed several alternatives. I have shown how evolving only the interfaces of a model A system is more efficient, less time-consuming, as reliable and in certain cases even better than evolving the order-parameter via a discretized TDGL equation. I have also shown how interface dynamics exhibits the same characteristics of dynamical scaling and power-law growth as the bulk dynamics, and allowed for the easy measurement of the curvature autocorrelation function. I also discussed several features of the model A interfacial dynamics via this correlation function, such as the Gaussian behavior at small distances along interfaces and the
clear signature of a dominant length-scale in the dynamics. It was found to have a different nature than the dominant length-scale of model B dynamics. An analytical derivation of the curvature autocorrelation function was obtained and compared with the numerical results. They agree well in terms of envelope behavior, but the analytical prediction does not exhibit the dominant interface undulation mode observed in the simulations.
Chapter 3

Generalizations for curved spaces

In chapter 2 I described and implemented a formalism that allowed me to evolve an order-parameter field by focusing only on regions not in equilibrium, namely the interfaces between domains. The formalism was based on the Allen-Cahn interface equation. In this chapter I first discuss phase-ordering kinetics on general Riemannian manifolds. I derive a non-Euclidean model A (NEMA) equation and a central analytical result, the non-Euclidean interface velocity (NEIV) equation, which explicitly relates the interface dynamics to the local intrinsic geometry of the surface. After describing some of the algorithmic issues relevant to both non-Euclidean formalisms, I test the NEIV equation via simulations with both formalisms. Finally, I discuss generalizations of the order-parameter structure factor and curvature autocorrelation function introduced in chapter 2.

For ease of reference, I recall Euclidean model A from chapter 1:

\[
\frac{\partial \phi}{\partial t} = -M \frac{\delta \mathcal{F}(\phi)}{\delta \phi} \tag{3.1a}
\]

\[
\mathcal{F}(\phi) = \int dS \left( V(\phi) + \frac{\xi^2}{2} (\nabla \phi)^2 \right) \tag{3.1b}
\]

3.1 TDGL equation

In order to use a dynamical equation in a curved space, I have to make use of some of the powerful tools of differential geometry and tensor calculus. A concise introduction
can be found in François David’s article [34], in particular the first 2 chapters. I introduce the notation and concepts as required. There is one basic requirement to fulfill: the physics must be independent of the coordinate system used on the manifold. This automatically suggests some of the necessary transformations.

I first define a curvilinear coordinate system $\mathcal{C}$ on the manifold, such that with each point on the manifold there is associated a unique coordinate pair $\vec{R} = (u^1, u^2)$. Note that a point is defined per se, without the need for a coordinate system. However a coordinate pair necessarily requires such a coordinate system and must invariably be associated with a point. It must be kept in mind that $\vec{R}$ is not strictly speaking a vector. A coordinate system is necessary for differential operations on a manifold to be defined, but Differential Geometry makes the quantitative result of differential operations independent of the choice of $\mathcal{C}$.

Consider the surface integral in eq. (3.1b). The numerical value of the integrand is defined for each point on the manifold and does not depend on the choice of $\mathcal{C}$. You can merely point to a point on the manifold and get the value of the integrand at that point. I.e., if the manifold is cut into $N$ little boxes forming a connected and complete covering of the surface, changing the coordinate system does not affect the $N$ little boxes, only the order in which they are summed by the integral. Hence, for the result of the integral to be independent of $\mathcal{C}$ it is sufficient that $dS$ at a point be independent of $\mathcal{C}$. This is achieved by writing $dS$ as $\sqrt{g_{\mathcal{m}}} du^1 du^2$, where $g_{\mathcal{m}}$ is the determinant of the metric tensor $g_{ij}$ defined on the manifold. The metric tensor is like a jacobian matrix, relating lengths and angles in $\mathcal{C}$ to those in a hypothetical Euclidean imbedding space, i.e. the space in which I do length and angle measurements. The $g_{ij}$ are the covariant components of the tensor, while the $g^{ij}$ are the contravariant components, defined by the relationship $g_{ik}g^{kj} = \delta^i_j$, with $\delta^i_j = 1$ when $i = j$ and 0 otherwise. As is customary in tensor calculus, I use implicit summation over alternate repeated indices, i.e. $g_{ik}g^{kj} \equiv \sum_k g_{ik}g^{kj}$. For a surface, $g_{ij}$ is a $2 \times 2$ matrix and $g^{ij}$ its inverse. The covariant components of the metric tensor for a surface are given by

$$g_{ij} \equiv \frac{\partial \vec{X}}{\partial u^i} \cdot \frac{\partial \vec{X}}{\partial u^j}$$  \hspace{1cm} (3.2)
where \( \bar{X}(u^1, u^2) \) is a vector pointing to the surface, relative to the origin in Euclidean space. Therefore, \( \sqrt{g_{ij}} \) represents the ratio of length in Euclidean space to length along \( u^i \), while \( g_{12} \) is proportional to the cosine of the angle between the coordinate lines \( u^1 \) and \( u^2 \). Note that a scalar is a zero-rank tensor.

In the same way that a tensor can be given either via its covariant or contravariant components, so can a vector, with the same notational conventions. The contravariant components of a vector \( \bar{a} \) are \( a^i \), while the covariant components are \( a_i \). It is common to refer to \( a^i \) simply as a vector, and to \( a_i \) as a one-form. The numerical values of \( a^i \) depend on the parametrization used, as do those of any non-zero-rank tensor. Therefore, a parametrization-invariant scalar product \( \bar{a} \cdot \bar{b} \) is defined as \( g_{ij} a^i b^j \), or in terms of the corresponding one-forms as \( g^{ij} a_i b_j \). The square-gradient term in eq. (3.1b) involves the scalar product of two one-forms denoted \( \partial_i \phi \): \( (\bar{\nabla} \phi)^2 \) is really \( g^{ij} \partial_i \phi \partial_j \phi \) (why the gradient operator produces a one-form rather than a vector is discussed in [35]). It is easily checked that redefining the scalar product in this square-gradient term is sufficient to make it parametrization-invariant, even though \( \partial_i \phi \) is not itself independent of coordinate system.

The remaining step is to define a functional derivative in non-Euclidean space. Consider a functional \( \mathcal{H}\{\phi; \phi, i\} \) depending on \( \phi \) and the partial derivatives of \( \phi \) with respect to coordinate \( u^i \), denoted by the subscript \( \cdot, i \), \( i = 1, 2 \). A functional derivative \( \delta / \delta \phi \) of \( \mathcal{H} \) is written for a flat, cartesian coordinate system

\[
\frac{\delta \mathcal{H}\{\phi; \phi, i\}}{\delta \phi} = \frac{\partial \mathcal{H}}{\partial \phi} - \frac{\partial}{\partial u^i} \left( \frac{\partial \mathcal{H}}{\partial \phi, i} \right)
\]  

(3.3)

The first term on the RHS is a scalar, therefore independent of \( \mathcal{C} \). However, in curved space the second term is the divergence of a one-form, therefore it involves both a scalar product and the differentiation of a one-form. Differentiation of vectors, one-forms and higher-rank tensors (which are generalizations of vectors) is done via the covariant derivative \( D_i \) along the direction \( u^i \). This derivative has the same properties as the Euclidean derivative, with the exception that derivatives do not commute when applied to anything other than a scalar function: \( D_i D_j T \neq D_j D_i T \) if
3.1. TDGL EQUATION

$T$ is a non-zero-rank tensor. $D_i$ takes into account, amongst other things, that in a curvilinear coordinate system the direction of the basis vectors changes as one moves. When applied to a one-form $a_j$, 

$$D_i a_j \equiv \frac{\partial a_j}{\partial u^i} - \Gamma^k_{ij} a_k$$  \hspace{1cm} (3.4)

where

$$\Gamma^k_{ij} \equiv \frac{1}{2} g^{kl} \left( \partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij} \right)$$  \hspace{1cm} (3.5)

is the affine connection on the manifold. Since $\Gamma^k_{ij} = \Gamma^k_{ji}$, the affine connection has 6 different components for a 2-dimensional manifold. The divergence is by definition the scalar product $g^{ij} D_i a_j$. Therefore the second term on the RHS of eq. (3.3) must be written

$$\frac{\partial}{\partial u^i} \left( \frac{\partial \mathcal{H}}{\partial \phi_i} \right) \rightarrow g^{ij} \frac{\partial}{\partial u^i} \left( \frac{\partial \mathcal{H}}{\partial \phi_j} \right) - g^{ij} \Gamma^k_{ij} \frac{\partial \mathcal{H}}{\partial \phi_k}$$  \hspace{1cm} (3.6)

This can be rewritten by using the following identity for the Laplacian on a curved surface, called the Laplace-Beltrami operator $\nabla_{LB}^2$:

$$\nabla_{LB}^2 \equiv g^{ij} \frac{\partial}{\partial u^i} \frac{\partial}{\partial u^j} - g^{ij} \Gamma^k_{ij} \frac{\partial}{\partial u^k}$$  \hspace{1cm} (3.7a)

$$= \frac{1}{\sqrt{g_m}} \frac{\partial}{\partial u^i} \left( g^{ij} \sqrt{g_m} \frac{\partial}{\partial u^j} \right)$$  \hspace{1cm} (3.7b)

where $g_m$ is the determinant of the metric tensor $g_{ij}$ of the manifold. The Laplace-Beltrami operator gives the same numerical value for the Laplacian of a scalar function, independent of $\mathcal{C}$. Therefore I write the non-Euclidean functional derivative as

$$\frac{\delta \mathcal{H}\{\phi, \phi, i\}}{\delta \phi}_{\text{NE}} = \frac{\partial \mathcal{H}}{\partial \phi} - \frac{1}{\sqrt{g_m}} \frac{\partial}{\partial u^i} \left( g^{ij} \sqrt{g_m} \frac{\partial \mathcal{H}}{\partial \phi_j} \right).$$  \hspace{1cm} (3.8)

where the subscript NE indicates the operation is done in a non-Euclidean space. In model A, $\mathcal{H}$ is given by eq. (3.1b). Therefore $\partial \mathcal{H}/\partial \phi = \partial \mathcal{V}/\partial \phi$, while

$$\frac{\partial \mathcal{H}}{\partial \phi_j}_{\text{NE}} = \frac{\delta}{\delta \phi_j} \left( \frac{\xi^2}{2} \int \left( \phi, i \right)^2 dS \right) = \xi^2 \phi, j$$  \hspace{1cm} (3.9)
Using $\phi_{,j} \equiv \frac{\partial \phi}{\partial u^j}$ and putting everything together, model A in a non-Euclidean manifold is written

$$\frac{\partial \phi}{\partial t} = -\frac{dV(\phi)}{d\phi} + \frac{1}{\sqrt{g_m}} \frac{\partial}{\partial u^j} \left( g^{ij} \sqrt{g_m} \frac{\partial \phi}{\partial u^i} \right). \quad (3.10)$$

The second term is of course the Laplacian of $\phi$ in curved manifolds. This result is not surprising since in going from Euclidean to non-Euclidean space the correspondence $\nabla^2 \rightarrow \nabla_{LB}^2$ often holds, but must nonetheless be checked rigorously. Eq. (3.10) naturally goes to the Euclidean form for a Euclidean manifold since $\nabla_{LB}^2 \rightarrow \nabla^2$ in this limit.

Several things can be said about eq. (3.10). The most important is that the dynamics of the order-parameter field $\phi(u^1, u^2, t)$ is coupled only to the intrinsic geometry of the surface since the only geometry-dependent term in eq. (3.10) is the Laplace-Beltrami (LB) operator. This operator depends only on the metric and the affine connection, which are not sufficient to uniquely define the complete geometry of a surface. The intrinsic geometry of a manifold is characterized only by quantities which can be measured from within the manifold, without regards for any particular imbedding of the surface in a higher-dimensional space. Two surfaces can have the same intrinsic geometry but not look the same from the imbedding three-dimensional space, i.e. not have the same extrinsic geometry. Therefore, the dynamics of the order-parameter as given by eq. (3.10) is not only valid on any surface, but for a specific surface it also describes the dynamics on all surfaces with the same intrinsic geometry. Note that two surfaces which do not have the same intrinsic geometry cannot be made to have the same metric everywhere except in limited regions.

It is worth pointing out that rescaling of time, length, order-parameter and coefficients valid in the Euclidean model A is also valid without modification in eq. (3.10). This is inevitable since the dimensionality of each member of the LB operator is still two, as in the Euclidean case. From a purely differential geometric point of view, there is no difference in the algebraic form of eq. (3.7b) for a curved and a non-curved space. A flat surface with a cylindrical or otherwise curvilinear coordinate system
3.1. TDGL EQUATION

uses eq. (3.7b) as well. The only difference is in the numerical values of the metric components and its derivatives.

The Laplacian expression in eq. (3.10) must be dealt with carefully. There are an infinity of ways to parametrize a given surface, and therefore an infinity of metrics that can be defined for that surface. I could in principle choose a parametrization, write out the different terms in the Laplacian for that parametrization, and order them so that I get one part of the TDGL which is essentially Euclidean, plus some extra terms which account for the surface being warped:

\[
\frac{\partial \phi}{\partial t} = \left\{ -\frac{dV(\phi)}{d\phi} + g^{11} \frac{\partial}{\partial u^1} \frac{\partial \phi}{\partial u^1} + g^{22} \frac{\partial}{\partial u^2} \frac{\partial \phi}{\partial u^2} \right\} \\
+ 2g^{12} \frac{\partial}{\partial u^1} \frac{\partial \phi}{\partial u^2} + \frac{1}{\sqrt{g}} \frac{\partial \phi}{\partial u^i} (g^{ij} \sqrt{g}).
\] (3.11)

This equation can be interpreted as describing a relaxational, non-conserved dynamics in Euclidean \((u^1, u^2)\) space, with anisotropic, position-dependent diffusion given by coefficients \(g^{11}\) and \(g^{22}\), with extra terms whose effect I seek to understand. However different parametrizations give different diffusion coefficients and different extra terms. It is hard to imagine how one could extract any physical features of eq. (3.10) in this way. An alternate route is to make use of a peculiarity of two-dimensional Riemannian manifolds, i.e. that one physical quantity is sufficient to completely describe its local intrinsic geometry: the Gauss curvature \(K_G\). Any parametrization gives the same \(K_G\) at a given point of the surface, and all surfaces with the same intrinsic geometry have the same \(K_G\) for every point of the surface. A transformation of eq. (3.10) leading to an expression involving \(K_G\) explicitly seems like a more promising approach. Or, perhaps something could be gained by interpreting the metric as a static inhomogeneous surfactant field[36]. However, I use yet a different route, which I discuss in the following section.

The first question that springs to mind with the NEMA dynamics given by eq. (3.10) is how the LB operator affects the linearly unstable modes found in the Euclidean case. If the eigenvalues of the LB operator are denoted by \(-\lambda\), then the linear dispersion relation \(\omega(\lambda)\) for eq. (3.10) is simply \(\omega(\lambda) = 1 - \lambda\). \(\phi = \text{const}\) is
3.1. **TDGL EQUATION**

still an eigenfunction of the LB operator with eigenvalue \( \lambda = 0 \). It is also known that the eigenfunctions of \( \nabla_{\text{LB}}^2 \) form a complete orthonormal set for any Riemannian manifold[37]. From this it is easily shown that all \( \lambda \) are real and non-negative. Therefore all modes with \( 0 < \lambda < 1 \) are linearly unstable modes. Of course, the eigenfunctions and eigenvalues depend on the manifold, but not on \( \mathcal{C} \). Furthermore, it is known that the number of roots, and therefore the number of oscillations, of the eigenfunctions increases with \( \lambda[37] \). The interface width is therefore still closely related to the eigenfunction associated with the largest unstable \( \lambda \), i.e. \( \lambda = 1 \), though it now seems impossible to estimate what the related interface width might be. However, the physical nature of the LB operator is the same as that of the Laplacian in Euclidean model A. Interface width is a short-scale feature of the dynamics, and on sufficiently short length-scales any curved space appears locally Euclidean. Therefore, if on the scale of the interface width both the manifold and the interfaces are gently curved, I can assume the interface width is not affected at all. Therefore, the \( \lambda = 1 \) mode should still have oscillations on the scale of \( 2\pi \). This is indeed observed in simulations, though no quantitative investigation was carried out as my focus is on long-range order rather than short-lengthscale effects of geometry.

I discuss in further detail the eigenfunctions of the LB operator in section 3.5.1. The short analysis above first and foremost allows for the conclusion that a band of linearly unstable modes still exists and that interface width is still on the order of \( 5\xi \) (see section 1.1), and therefore that exponential growth of fluctuations, followed by formation of domains separated by sharp interfaces, and finally subsequent slow interface motion, still define 3 different dynamical regimes, as discussed in chapter 1. In order to gain further insight into the late stage regime described by the NEMA, I now extract an interface equation from eq. (3.10) that allows me to implement the ideas already put forward in chapter 2, but for interfaces on curved surfaces.
3.2 Interface equation

The influence of the curved geometry on the motion of NEMA interfaces is best seen by generalizing the Allen-Cahn interface equation (1.9) discussed in chapter 1. The aim is therefore to extract an interface equation from eq. (3.10), of a similar form as eq. (1.9), valid in any two-dimensional Riemannian manifold. The total curvature $K$ of a point belonging to a curve in the two-dimensional manifold is the curvature as measured in the imbedding three-dimensional space. Intuitively, I expect the total curvature $K$ to be replaced with the geodesic curvature $K_g$. The latter is the component of the former which lies in the plane tangent to the manifold at the point of interest. In other words it is that part of the total curvature which is intrinsic to the manifold, i.e. the part measurable from within the surface. Because it is purely intrinsic to the surface, geodesic curvature can be expressed in terms of the intrinsic geometric characteristics of the surface such as metric tensor $g_{ij}$ and affine connection $\Gamma^k_{ij}$. The derivation I present here is lengthier than that which appears in one of my publications[23], but also geometrically far more transparent. Indeed, the tools of modern differential geometry are powerful and compact, hiding many geometrical concepts that are useful in understanding interfaces on curved surfaces. I therefore use classical differential geometry with some Riemannian concepts, in three steps. First I use an appropriately chosen coordinate system on a generic manifold to simplify the non-Euclidean TDGL equation (3.10). Then I find an expression for the geodesic curvature of a line in a Riemannian manifold. Finally I use this expression to further simplify the TDGL equation and obtain the interface velocity.

For a general coordinate system on the curved surface, the LB operator of eq. (3.10) expands to a nasty sum of inhomogeneous terms. These can be greatly simplified by selecting an appropriate coordinate system both inside the surface and in three-dimensional space in the vicinity of the interface, and making use of a very practical feature of Riemannian manifolds, namely that any Riemannian manifold is locally Euclidean. I therefore define $C$ in the vicinity of a point $P$ of an interface, shown in fig. 3.1, such that lines of constant $u^2$ are parallel to the interface while lines
Figure 3.1: Coordinate system used to generalize Allen-Cahn interface equation. See text for description.

of constant $u^1$ are perpendicular to it (i.e. $u^1$ is parallel to the interface). This figure shows a patch of the surface labelled $S$, a part of interface is labelled $I$, while $P$ is the point whose velocity I seek, $G$ is a geodesic line tangent to the interface at $P$, and the surface patch has been translated and rotated (if necessary) so that $P$ lies at the origin of the imbedding three-dimensional Euclidean coordinate system. The $\hat{x}\hat{y}$ plane is chosen tangent to the surface at $P$, i.e. the $\hat{x}\hat{y}$ plane coincides with what is referred to as the tangent manifold at $P$. Furthermore the $\hat{x}$ axis is chosen to be tangent to $I$ at $P$. The patch of surface can be represented by a function $z = f(x, y)$ in the vicinity of $P$, with $P$ at the extremum of that function on the domain of the patch. Geodesics are by definition curves of extremal length in a manifold. In the Euclidean plane, only straight lines are geodesics. On the surface of a sphere, only circles (or parts thereof) whose centers coincide with that of the sphere are geodesics. Therefore, the chosen curvilinear coordinate system and three-dimensional coordinate system guarantee that $G$ lies in the $\hat{x}\hat{z}$ plane which contains the normal to $f(x, y)$ at $P$. 
3.2. INTERFACE EQUATION

This choice of coordinate systems also implies $g_{12} = g_{21} = 0$, and the partial
derivatives of the order-parameter along $u^1$ are 0 ($\partial_1 \phi = (\partial_1)^2 \phi = 0$). The non-
Euclidean TDGL equation therefore simplifies to (note eq. (3.7a) and (3.10))

$$\frac{\partial \phi}{\partial t} = -\frac{dV}{d\phi} + (g^{22} \partial_{22} - g^{11} \Gamma_{11}^2 \partial_2 - g^{22} \Gamma_{22}^2 \partial_2) \phi. \quad (3.12)$$

I now need to find the curvature of a line in following $u^1$ in the chosen coordinate systems.

3.2.1 Geodesic interface curvature

The curvature of a line in a Euclidean space of dimension $d$ can be expressed as a
positive scalar $K$ times a $d$-dimensional unit vector pointing to the center of curvature.
Thus curvature can be split into $d$ mutually orthogonal components. A line interface
on a two-dimensional surface embedded in three-dimensional space therefore has 3
independent components to the total curvature $\vec{K}$ at any of its points. This also
means that $\vec{K}$ can be expressed as the sum of two orthogonal vectors. The first,
denoted $\vec{K}_n$, is perpendicular to the surface at the point of interest ($P$), and the
other, denoted $\vec{K}_g$, is contained in the tangent plane to the surface at the same point
($P$). Using the coordinate system defined in the previous section, this is expressed

$$\vec{K} = \vec{K}_n + \vec{K}_g \equiv -K_n \hat{z} + K_g \hat{n} \quad (3.13)$$

where $\hat{n}$ is a three-dimensional unit vector (not shown in fig. 3.1) tangent to the
surface at $P$, and perpendicular to $I$. $K_n$ is called the normal curvature, while $K_g$
is by construction the geodesic curvature. Hence

$$K_g(P) = \sqrt{\vec{K}^2 - \vec{K}_n^2} \quad (3.14)$$

Note that $\vec{K}_n$ is also the total curvature vector of the geodesic $G$ at $P$. I can express $K$
and $K_n$ in terms of surface geometric quantities by using a parametric representation
$\vec{X}_M$ for the surface, $\vec{X}_G$ for the geodesic line $G$ with arclength parameter $s_G$, and $\vec{X}_I$
for the interface $I$ with arclength parameter $s_I$:

$$
\vec{X}_M = [F(u^1, u^2), G(u^1, u^2), H(u^1, u^2)] \\
\vec{X}_G = [F(u^1(s_G), u^2(s_G)), G(u^1(s_G), u^2(s_G)), H(u^1(s_G), u^2(s_G))] \\
\vec{X}_I = [F(u^1(s_I), 0), G(u^1(s_I), 0), H(u^1(s_I), 0)]
$$  \tag{3.15a} \tag{3.15b} \tag{3.15c}

where I used square brackets to indicate three-dimensional vectors and set, without loss of generality, the interface to be on the $u^2 = 0$ line. $I$ and $G$ share the same unit tangent $\hat{\tau}$ (not shown) at $P$ as well as the same $\vec{K}_n$. Therefore $\dot{\tau} \equiv d\vec{X}_G/ds_G = d\vec{X}_I/ds_I$ (of unit length only because I used arclength parametrizations for both curves). I also define an "acceleration" vector for each curve,

$$
\vec{a}_I \equiv \frac{d\dot{\tau}}{ds_I} \equiv \frac{d^2\vec{X}_I}{ds_I^2} \\
\vec{a}_G \equiv \frac{d\dot{\tau}}{ds_G} \equiv \frac{d^2\vec{X}_G}{ds_G^2}
$$  \tag{3.16a} \tag{3.16b}

In terms of $a_{I,G}$ and $\dot{\tau}$, $K$ and $K_n$ are written[38]

$$
K = \frac{|\dot{\tau} \times \vec{a}_I|}{|\dot{\tau}|^3} \tag{3.17a} \\
K_n = \frac{|\dot{\tau} \times \vec{a}_G|}{|\dot{\tau}|^3} \tag{3.17b}
$$

Once more, the coordinate system of the previous section causes several terms to be null at $P$ in eq. (3.16). Namely,

$$
\partial_2 F = \partial_1 G = \partial_1 H = \partial_2 H = 0 \\
\frac{d^n u^2}{ds_I^n} = \frac{du^2}{ds_G} = 0
$$  \tag{3.18a} \tag{3.18b}

With these and the chain rule

$$
\frac{\partial \vec{X}_{\{I,G\}}}{\partial s_{\{I,G\}}} = \frac{\partial \vec{X}_{\{I,G\}}}{\partial u^1} \frac{du^1}{ds_{\{I,G\}}} + \frac{\partial \vec{X}_{\{I,G\}}}{\partial u^2} \frac{du^2}{ds_{\{I,G\}}}, \tag{3.19}
$$
3.2. INTERFACE EQUATION

eq. (3.17) becomes, after lengthy but otherwise straightforward algebra,

$$K = \sqrt{\left(\frac{\partial^2 F}{\partial_1} \right)^2 + \left(\frac{\partial^2 H}{\partial_1} \right)^2} \quad (3.20a)$$

$$K_n = \sqrt{\left[ \frac{\partial^2 G + \frac{u^2}{(u')^2} \partial_2 G}{(\partial_1 F)^2} \right]^2 + \left(\frac{\partial_2 H}{\partial_1} \right)^2} \quad (3.20b)$$

where prime denotes differentiation with respect to the relevant arclength.

The final step is to link the derivatives of $F, G, H$ and $u'$ to the metric tensor and affine connection at $P$. In the three-dimensional coordinate-system defined in fig. 3.1, the metric tensor on the surface is simply

$$g_{ij}(P) = \begin{pmatrix} (\partial_1 F)^2 & 0 \\ 0 & (\partial_2 G)^2 \end{pmatrix} \quad (3.21)$$

while the two equations defining geodesics such as $G$ are

$$\frac{d^2 u^k}{ds_G^2} + \Gamma^k_{ij} \frac{du^i}{ds_G} \frac{du^j}{ds_G} = 0. \quad (3.22)$$

Together with eq. (3.18b), you can find

$$\frac{u^2}{(u')^2} = -\Gamma^2_{11}. \quad (3.22a)$$

Differentiating the tensor components in eq. (3.21) I also get

$$\partial^2 G \partial_2 G = g_{22} \Gamma^2_{11}. \quad (3.22b)$$

Putting all this together in eq. (3.20) and substituting in eq. (3.14), I have

$$K_g = g^{11} \sqrt{g_{22} \Gamma^2_{11}} \quad (3.23)$$
so that eq. (3.12) becomes

\[
\frac{\partial \phi}{\partial t} = -\frac{dV}{d\phi} + \left( g^{22} \partial_{22}^2 \phi - \frac{K_g}{\sqrt{g_{22}}} \partial_2 \phi - g^{22} \Gamma_{22}^{22} \partial_2 \phi \right) \phi. \tag{3.24}
\]

### 3.2.2 Velocity vs curvature (NEIV)

Model A dynamics is governed by the minimization of the system free energy. When the domains have fully formed, the energy dissipation can only occur via the decrease in the quantity of interfaces present, i.e. by the decrease of the length of interfaces. When an interface can no longer shrink in length on the patch \( S \), the free energy is at a minimum in this region, and the interface should become stationary on \( S \). For the interface \( I \) to be of minimal length in the patch region \( S \) of fig. 3.1, it must be a geodesic in this region. Since a geodesic has \( K_g = 0 \) everywhere on \( S \), the equation for a locally stationary interface, in the chosen coordinate system, must be

\[
0 = g^{22} \partial_{22}^2 \phi - g^{22} \Gamma_{22}^{22} \partial_2 \phi - \frac{dV(\phi)}{d\phi}. \tag{3.25}
\]

This is also equivalent to a one-dimensional equation for a flat interface in flat space but in a coordinate system which is cartesian along \( u^1 \) and non-cartesian along \( u^2 \). I.e., the solution to eq. (3.25) is the one-dimensional profile discussed in the introduction on page 9. If I suppose the interface is gently curved on the manifold in the vicinity of \( P \), so that \( K_g \ll 1 \), and that the geodesic \( G \) is similarly gently curved in three-dimensional space, so that \( K_n \ll 1 \), the interface profile of \( I \) along \( u^2 \) should be given to very close approximation by eq. (3.25) in the vicinity of \( P \). This allows me to separate eq. (3.24) into two parts, one equal to the RHS of eq. (3.25) and the other containing the remaining terms:

\[
\frac{\partial \phi}{\partial t} = \left\{ -\frac{dV}{d\phi} + g^{22} \partial_{22}^2 \phi - g^{22} \Gamma_{22}^{22} \partial_2 \phi \right\} - \frac{K_g}{\sqrt{g_{22}}} \partial_2 \phi \tag{3.26}
\]
The remaining terms should therefore give a first-order approximation of the driving force controlling interface motion:

\[
\frac{\partial \phi}{\partial t} = -K_g \frac{\partial \phi}{\partial s_2}
\]  (3.27)

where I have made use of the equality \( g^{22} = 1/g_{22} \) at \( P \) in the coordinate system chosen, as well as \( ds_2 = \sqrt{g_{22}}du^2 \) to define an arclength coordinate along \( u^2 \). An arclength measures distance in the Euclidean imbedding space and is therefore independent of parametrization. Arclength differentiation of a scalar function along a line is the same as covariant differentiation if the line is taken as a one-dimensional manifold, which is what eq. (3.27) does. Since \( K_g \) is small, it changes very slowly as the interface is moving, so eq. (3.27) admits moving-front solutions of the type \( \phi(s_2 - \nu t) \) where the speed \( \nu \) is \( K_g \). When written with the \( M \) and \( \xi \) coefficients,

\[
\nu = M\xi^2 K_g
\]  (3.28)

I refer to eq. (3.28) as the non-Euclidean interface velocity equation, or NEIV equation for short. For a Euclidean system, \( K_g = K \), recovering the Allen-Cahn result. Apart from the three assumptions of \( K_g \ll 1/\xi, K_n \ll 1/\xi \) and one-dimensional profile normal to the interface, the NEIV equation is completely general with regards to Riemannian surfaces, as well as being completely independent of the choice of coordinate system. Eq. (3.28) makes explicit the effect of non-Euclidean space on interface motion, which the non-Euclidean TDGL equation does not do.

The apparent simplicity of the NEIV equation is misleading, however. The geodesic curvature of a line in a curved manifold depends both on the position of that line and its orientation on the surface. Therefore, \( K_g \) introduces into the dynamics a new, position- and orientation-dependent length-scale. This strongly suggests that NEMA dynamics cannot be self-similar, and therefore that dynamical scaling will not be observed on curved surfaces. The only exception to this conclusion might be self-similar surfaces, such as found over a certain range of length-scales in lipid bilayer membranes. Different surfaces should therefore show different phase-ordering
3.2.3 Discussion

In chapter 2, I introduced the concepts of normal and constant gauge. As for the Allen-Cahn equation, the NEIV equation is also best implemented numerically using the normal gauge. The presence of a curved surface does not qualitatively change anything to the normal gauge. The gauge is still, by definition, that in which interface points have a velocity perpendicular to the interface. Eq. (3.28) is written

$$\left(\frac{\partial u_i}{\partial t}\right)_\alpha = M\xi^2 k_g n^i$$

where $n^i$ is the $i^{th}$ component of $\hat{n}$, and $K_g$ is given by, in compact notation,

$$K_g = \epsilon_{ij} u'^i (u'^j + \Gamma^j_{ki} u'^k u'^l)$$

where implicit summation over repeated indices $\{i, j = 1, 2\}$ is assumed, as usual. Note that $u^1$ and $u^2$ now form a curvilinear coordinate system, but the prime still denotes differentiation with respect to arclength. Arclength is always the physical length of the line as measured in the Euclidean imbedding space. $\epsilon_{ij}$ is $(j - i)\sqrt{g_m}$. Recall that the subscript $m$ in $\sqrt{g_m}$ refers to the determinant of the metric of the manifold rather than that of the interface, which is $\sqrt{g}$. Eq. (3.30) may look more familiar if the first part is written out:

$$K_g = \sqrt{g_m} (u'^1 u''^2 - u'^2 u''^1) + \epsilon_{ij} u'^i \Gamma^j_{kl} u'^k u'^l$$

The terms in parenthesis are none other than eq. (2.25). In a flat cartesian coordinate system, $\sqrt{g_m} = 1$ and $\Gamma^j_{kl} = 0$ so eq. (3.31) and eq. (2.25) are equal, as they should be. The normal to the interface, $\hat{n}$, is known via the tangent $\hat{r}$, either by the Frenet equation eq. (2.4a) generalized for curved manifolds, or by using the property that $\hat{n} \cdot \hat{r} = 0$ independently of the chosen parametrization of the surface. The dot-product must of course be defined via the surface metric tensor, i.e. $\hat{n} \cdot \hat{r} = g_{ij} n^i r^j$, where...
3.2. INTERFACE EQUATION

\( \tau^i \equiv u'^i \) and \( n^i \) is the \( i^{th} \) component of the unit normal vector. Using \( \hat{n} \cdot \hat{r} = 0 \) to define \( \hat{n} \), and writing \( u^1 \equiv u \) and \( u^2 \equiv v \) as in chapter 2, \( \hat{r} \) and \( \hat{n} \) are written in the curved case

\[
\hat{r} \equiv \frac{\partial \tilde{R}}{\partial s} = (u', v') \tag{3.32a}
\]

\[
\hat{n} = \frac{1}{\sqrt{g_m}}(-u'g_{22} - u'g_{12}, u'g_{11} + v'g_{12}) \tag{3.32b}
\]

where eq. (3.32a) is identical to eq. (2.2a). It can be checked that eq. (3.32b) goes to eq. (2.2b) in the Euclidean limit. Note that \( K_g \), as given by eq. (3.30), is a signed scalar, so that the proper definition of \( \hat{n} \) is eq. (3.32b). If \( |K_g| \) is used instead, then \( \hat{n} \) must be computed via the Frenet equation (3.45a). Otherwise, \( |K_g| \hat{n} \) will not point in the direction of motion of the interface. Also, to see that \( \hat{r} \) is indeed a unit vector, note that the three-dimensional expression of \( \hat{r} \) is by definition \( d\tilde{X}/ds \) where \( \tilde{X}(u(s), v(s)) \) is a three-dimensional Euclidean vector pointing to the interface on the surface. \( d\tilde{X}/ds \) is a unit vector[38]. With help of the chain rule,

\[
1 = \left| \frac{\partial \tilde{X}}{\partial s} \right|^2 = \left| \frac{\partial \tilde{X}}{\partial u} \frac{\partial u}{\partial s} + \frac{\partial \tilde{X}}{\partial v} \frac{\partial v}{\partial s} \right|^2 \tag{3.33a}
\]

\[
= \left| \frac{\partial \tilde{X}}{\partial u} \right|^2 u'^2 + \left| \frac{\partial \tilde{X}}{\partial v} \right|^2 v'^2 + 2 \frac{\partial \tilde{X}}{\partial u} \cdot \frac{\partial \tilde{X}}{\partial v} u'v' \tag{3.33b}
\]

\[
\equiv g_{ij} u'^i u'^j \tag{3.33c}
\]

This shows that \( g_{ij} u'^i u'^j \), the square norm of \( \hat{r} \) as expressed from within the two-dimensional manifold, is 1.

Great care must be taken when differentiating vectors when the space is not Euclidean. Consider a vector field on the manifold. A vector order-parameter field on the manifold is an example, but also, more relevant here, is the series of tangent vectors of an interface and the bundle of normal vectors of an interface. Differentiating a vector field involves subtracting two vectors of this field separated by an element of arc distance \( ds \) along this interface. This simple operation implicitly assumes parallelism is well defined because one vector must be displaced by a distance \( ds \).
before it can be subtracted from the other. In curved space, parallelism depends on
the path taken. This can be seen by displacing a vector along a spherical surface
along various simple closed paths: its direction when it has returned to its departure
point, depends on the path taken.

The problem is partly solved by adopting a definition of parallelism which is
independent of any parametrization of the manifold: geodesics are used as the path
between points. As in flat space, transport along a geodesic means keeping a constant
angle between the vector and the tangent to the geodesic. The differential operator
that does this was introduced in section 3.1 as the covariant derivative. It must be kept
in mind that this operator is defined on tensors of any rank (i.e. scalars, contravariant
vectors, covariant vectors, and tensors of higher rank) but that its form differs in every
case. A vector in the familiar Euclidean sense becomes a contravariant vector in the
non-Euclidean case and its components are written \( w^i \). Covariant differentiation of a
contravariant vector is written

\[
D_i w^j = \frac{\partial w^j}{\partial u^i} + \Gamma^j_{ik} w^k
\]  

(3.34)

while, when applied to a scalar \( A \), it is equivalent to the Euclidean \( dA/du^i \). When
differentiating along an arbitrary direction, say along an interface, \( D_i \) is instead de-
noted simply by \( D_s \) as it uses arclength as its unit of distance. \( D_s \) applied to a vector
is written

\[
D_s w^i = \frac{\partial w^j}{\partial s} + \Gamma^j_{ik} w^i \frac{\partial u^k}{\partial s}.
\]  

(3.35)

It is instructive to consider why this is so. A vector \( w^j \) parallel-transported along a
small geodesic element between two points will have new components \( w'^j \) given by[34]

\[
w'^j = w^j - \Gamma^j_{ik} w^i \delta u^k
\]  

(3.36)

where the \( \delta u^k \) are the displacements along the geodesic. Now, if a vector field \( w^j \)
exists along a curve \( I \), such as the tangents to an interface, in the Euclidean plane
\( D_s w^j \) is easy to calculate. First, translate \( w^j(s) \) along a geodesic, i.e. a straight line,
3.2. INTERFACE EQUATION

to point \( s + \delta s \) of \( I \) (note that \( \delta s^2 = g_{ij} \delta u^i \delta u^j \)), and call the result \( w^i( s + \delta s ) \). In the Euclidean plane, \( w^i( s + \delta s ) = w^i( s ) \). Secondly, subtract \( w^i( s + \delta s ) \) from \( w^i( s + \delta s ) \), and divide by \( \delta s \). The same thing is done on curved surfaces, the only difference being that the geodesic is itself a curve and \( w^i( s + \delta s ) \) is given by eq. (3.36). Therefore,

\[
\delta s \, D_s w^i(s) = w^i(s + \delta s) - \left[ w^i(s) - \Gamma^i_{jk} w^j(s) \delta u^k \right]
\]  

(3.37)

from which one immediately obtains eq. (3.35). Interestingly, \( D_i g_{jk} = D_i g^{jk} = D_i g^{jk} = 0 \). Note also that I denote the covariant analog of the partial derivative by a calligraphic \( D \), as in \( D_i \).

I want now to find the non-Euclidean form for eq. (2.5b), as when eq. (3.29) is written out with eq. (3.30) and (3.32b), it looks rather monstrous and difficult to relate to eq. (2.5b). It is often sufficient when going from Euclidean to non-Euclidean geometry to make a few transformations:

\[
\begin{align*}
\text{vector } \vec{w} & \rightarrow w^i \\
\partial_i^n & \rightarrow D_i^n \\
\vec{a} \cdot \vec{b} & \rightarrow a^i g_{ij} b^j
\end{align*}
\]  

(3.38a, 3.38b, 3.38c)

where \( \partial_i^n \) is the \( n \)th order partial derivative along \( u^i \). However the result of applying eq. (3.38) should always be checked rigorously. Eq. (3.38) suggests the interface velocity should be written, on curved surfaces, as \( \vec{v} = D_s^2 \vec{R} \).

I now show this. Recall first that \( \vec{R} \) is not a vector in the usual sense. It is a short-hand notation for the \((u,v)\) pairs pointing to the interface from the origin of the curvilinear coordinate system, as in chapter 2. I write eq. (3.30) using

\[
D_s^2 u^i \equiv \frac{\partial^2 u^i}{\partial s^2} + \Gamma^i_{jk} \frac{\partial u^j}{\partial s} \frac{\partial u^k}{\partial s}
\]  

(3.39)

to condense the notation: \( K_g = \epsilon_{ij} u^i D_s^2 u^j \). Substituting this and eq. (3.32b) in
3.2. INTERFACE EQUATION

eq. (3.29) gives, for the u-part,

\[
\left( \frac{\partial u}{\partial t} \right)_\alpha = -u'(v'g_{22} + u'g_{12})D_s^2v + (v'^2g_{22} + u'v'g_{12})D_s^2u.
\] (3.40)

The \( v'^2g_{22} \) is related to \( u' \) by

\[
g_{22}v'^2 = 1 - g_{11}u'^2 - 2g_{12}u'u'
\] (3.41)

while the two components of \( \hat{n} \) are related to one another by applying \( D_s \) on both sides of \( g_{ij}u^{i'}u^{j'} = 1 \) and using eq. (3.35):

\[
(v'g_{22} + u'g_{12})D_s^2v = -(u'g_{11} + v'g_{12})D_s^2u.
\] (3.42)

Upon substitution of eq. (3.41) and (3.42) in eq. (3.40), everything cancels out except \( D_s^2u \). Using the two same rules but reversed for \( v \) rather than \( u \) one shows the same for \( \left( \frac{\partial v}{\partial t} \right)_\alpha \). Therefore the domain interfaces have a velocity given by

\[
\left( \frac{\partial u^i}{\partial t} \right)_\alpha = D_s^2u^i.
\] (3.43)

It is interesting to note that

\[
D_s^2 \hat{\mathbf{r}} = 0
\] (3.44)

is satisfied only when the interface is a geodesic on the surface.

The derivation (eq. (2.6) to eq. (2.24)) of the curvature equations (2.24) in chapter 2 is independent of the dimensionality of the imbedding space, as long as \( \hat{\mathbf{r}}, \hat{\mathbf{t}} \) and \( \hat{\mathbf{n}} \) are defined as in the previous section, but using their three-dimensional vector representations. The Frenet equations for lines in curved two-dimensional spaces are[35],

\[
D_s \hat{\mathbf{r}} = K_s \hat{\mathbf{n}}
\] (3.45a)

\[
D_s \hat{\mathbf{n}} = -K_s \hat{\mathbf{r}}
\] (3.45b)
3.2. INTERFACE EQUATION

and \( \hat{\tau} \) and \( \hat{n} \) can be written either as two- or three-dimensional vectors. Working in three-dimensional Euclidean space avoids scalar products involving surface metric quantities until the final equations are written out. The \( \vec{R} \) of section 2.2 is now like eq. (3.15c), a three-dimensional vector \( \vec{R} = [x(\alpha), y(\alpha), z(\alpha)] \) pointing from the origin of the Euclidean space to the interface on the surface. Of course, \( \vec{K} \) of eq. (2.11) is now the total curvature of the interface in three-dimensional space. For instance, eq. (2.12) is unchanged, but since \( \vec{K} \cdot \vec{v} = \vec{K} \cdot \hat{n} \nu = K_g \nu \), it is simplified to eq. (3.46b) below. The non-Euclidean curvature equations for a one-dimensional interface evolving in a two-dimensional manifold imbedded in a space of any dimensionality are easily verified to be

\[
\left( \frac{\partial K_g}{\partial t} \right)_\alpha = \frac{\partial^2 \nu}{\partial s^2} + K_g \vec{K}_g \cdot \vec{v} \tag{3.46a}
\]

\[
\left( \frac{\partial \sqrt{g}}{\partial t} \right)_\alpha = -\sqrt{g} \vec{K}_g \cdot \vec{v} \tag{3.46b}
\]

\[
\vec{v} = M \xi^2 \vec{K}_g \tag{3.46c}
\]

Note again that vectors appearing in eq. (3.46) can be written in their two- or three-dimensional form. Scalar products are Euclidean if three-dimensional vectors are used, but non-Euclidean when the vectors \( \hat{\tau}, \hat{n} \) and \( \vec{v} \) are represented from within the surface.

What cannot be assessed directly from eq. (3.29) is whether an interface evolving by that equation is always decreasing its length, as it should given the nature of interface motion in model A dynamics. Eq. (3.29) says that if an interface has \( K_g = 0 \) everywhere, it does not move. But since an interface is not an equilibrium system, there is no guarantee that a state where \( K_g = 0 \) everywhere is ever reached in the first place. Eq. (3.46b) shows that \( \sqrt{g} \) is a monotonically decreasing function in time, everywhere along the interface, at all times, until \( K_g = 0 \) everywhere. Since elements of arc along the interface and elements of the \( \alpha \) parameter are related by \( ds = \sqrt{g} \, d\alpha \), and \( d\alpha \) is constant between any pair of points of the interface, \( ds \) decreases monotonically in time; therefore so does the interface length \( L = \int ds \). A further
consequence of the monotonically decreasing interface length is that interfaces will inevitably reach points of local minimum length in configuration space, i.e. they can become stationary before disappearing, thus preventing the system from minimizing its Ginzburg-Landau free energy. Such local configurational minima of length define geodesics. Closed geodesics can exist only on curved surfaces.

One can get a better feel for geodesics by considering a stretchable rubber loop on the surface of interest. If the loop cannot decrease or increase in length without partly moving out of the surface, it follows a geodesic. An example is a circular interface around the neck of a peanut-shaped surface. In curved spaces, geodesics are uniquely defined only over sufficiently small distances. The combination of eq. (3.29), which is a physical approximation, and the exact geometrical equation (3.46b), firmly establishes that non-Euclidean model A interfaces, as given by eq. (3.29), monotonically decrease in length in time and can therefore become trapped in metastable states if closed geodesics exist on the surface.

In flat systems the shape of a curve is invariant under operations of translation and rotation. Hence $\tilde{R}(s)$ is sufficient to completely describe the shape of the curve. But in curved surfaces, knowledge of the shape $\tilde{R}(s)$ of the interface requires knowledge of the surface metric tensor and affine connection at $s$ if only $K_g$ is known. If the starting point of the interface is translated and the starting angle rotated, the surface quantities in eq. (3.30) change and so does the shape of the curve. It is much more accurate to obtain $K_g(s)$ from $\tilde{R}(s)$ rather than vice versa, so eq. (3.46) is unlikely to be used on its own in numerical simulations on curved surfaces. However, it may yet be numerically relevant in an intermediate step in a more robust integration routine, described in appendix B.

### 3.3 Numerical methods

Most simulations in chapter 4 are only possible with the interface formalism as the bulk simulations become unmanageably slow or the required measurements are simply too time-consuming to obtain from bulk configurations. However, bulk integration is
necessary to create interfaces. In the present section, I therefore explain how both
the bulk and interface simulations are done. For the bulk simulation methods, I give
numerical details pertaining only to the torus manifold but they are valid for any
surface of genus 1 (see below). More complex surfaces, such as the sinusoid surfaces
of chapter 4, are dealt with in the same way. Why the torus was chosen for some of
the simulations is explained below.

3.3.1 TDGL on a curved surface

Two simple surfaces which have non-trivial geometry are the sphere and the torus.
The sphere is a homogeneous and isotropic manifold with a constant Gauss curvature.
These characteristics are nice in that spatial and orientational averaging of measure-
ments is possible, while the torus measurements can only be averaged azimuthally.
Also, the eigenfunctions of the LB operator, needed in section 3.5.1, are known an-
alytically for the sphere, but not for the torus. However, domains on a sphere are
not dramatically affected by the spherical geometry until they span almost a third of
the sphere area, which means geometry effects are important only after the late-stage
regime gives way to the finite-size regime. The torus, on the other hand, has a region
of rather strong negative Gauss curvature, so that geometry effects are not only likely
to be visible before the finite-size regime, but also the effect of the sign of Gauss cur-
vature on the dynamics can be investigated. This is done in chapter 4. The sphere
is impractical both analytically and numerically because any parametrization of the
sphere involves at least 2 poles (this is a topological characteristic) in the vicinity of
which the metric components are ill-behaved, and does not easily incorporate periodic
boundary conditions. Simulations on the sphere hence require far more sophisticated
algorithms. The torus, however, has no poles and periodic boundary conditions nat-
urally fit in with the topology of the manifold. Another nice feature of the torus is
that its area is the same as that of the flat rectangle with sides equal in length to the
small and large radii of the torus, \( r_0 \) and \( R \) (see fig. 3.2).

Fig. 3.2 shows how the torus is parametrized: the \( \theta \) coordinate goes from 0 to
\( 2\pi r_0 \), with \( r_0 \) the small radius of the torus, while the \( \varphi \) coordinate is azimuthal and
3.3. **NUMERICAL METHODS**

![Diagram of a torus manifold](image)

Figure 3.2: Torus manifold of long radius $R$ and small radius $r_0$. Point $P$ on manifold has coordinates $(u^1, u^2) \equiv (\theta, \varphi)$. $\theta = 0$ is at inner equator (labelled “ie”).

goest from 0 to $2\pi R$, with $R$ the large radius. The “poles” of the torus are the circles of constant $\theta$ at ‘np’ and ‘sp’, while the inner equator is at ‘ie’ and the outer equator at ‘oe’.

The TDGL simulations use the usual FTCS (Forward in Time, Centered in Space) Euler integration method[24], with the time-step controlled by the largest eigenvalue of the discretized torus manifold in the chosen parametrization. In the $(\theta, \varphi)$ parametrization shown above, the torus surface is written in parametric form

$$
\tilde{X}(\theta, \varphi) = R \left\{ \left[ 1 - \rho \cos \left( \frac{\theta}{r_0} \right) \right] \cos \left( \frac{\varphi}{R} \right), \left[ 1 - \rho \cos \left( \frac{\theta}{r_0} \right) \right] \sin \left( \frac{\varphi}{R} \right), \rho \sin \left( \frac{\theta}{r_0} \right) \right\}
$$

(3.47)

In this representation the induced covariant metric tensor is

$$
g_{ij} = \begin{pmatrix}
1 & 0 \\
0 & \left( 1 - \rho \cos \left( \frac{\theta}{r_0} \right) \right)^2
\end{pmatrix},
$$

(3.48)

where $\rho \equiv r_0/R$ must be smaller than 1 if the surface is to be non-self-intersecting and orientable. The Euclidean limit is given by $\rho \to 0$, when the torus becomes a
cylinder. Note that a cylinder has Euclidean geometry since the Gauss curvature is everywhere 0. The dynamical equation for $\phi$ (eq. (3.10)) becomes

$$\frac{\partial \phi}{\partial t} = -\frac{dV(\phi)}{d\phi} + \frac{\partial^2 \phi}{\partial \theta^2} + \frac{1}{(1 - \rho \cos(\frac{\theta}{\rho}))^2} \frac{\partial^2 \phi}{\partial \varphi^2} - \frac{\rho \sin(\frac{\theta}{\rho})}{1 - \rho \cos(\frac{\theta}{\rho})} \frac{\partial \phi}{\partial \theta}$$

(3.49)

The derivatives can be discretized in the usual way[39], for instance

$$\frac{\partial \phi}{\partial t} \approx \frac{\phi_i(n+1) - \phi_i(n)}{\Delta t}$$

(3.50a)

$$\frac{\partial \phi}{\partial \theta} \approx \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta \theta}$$

(3.50b)

$$\frac{\partial^2 \phi}{\partial \theta^2} \approx \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta \theta}$$

(3.50c)

where the subscript $i$ indicates the $i^{th}$ lattice point along $\theta$ while $n$ is the number of timesteps taken since the beginning of the simulation. Euler methods are numerically unstable if $\Delta t$ is too large[24]. In flat space, the largest wavenumber controls the stability of the Euler integration. This wavenumber is proportional to the inverse of the smallest distance between points, i.e. to the lattice spacing. Similarly, the largest eigenvalue of the LB operator on the discretized torus corresponds to the smallest physical distance between points on the torus lattice. With the parametrization used, this is easily checked to be given by the tether along $\varphi$ located at $\theta = 0$, i.e. on the inner equator circle. Its length is $(1 - \rho)\Delta \varphi$ where $\Delta \varphi$ is the mesh in $(\theta, \varphi)$ space. A rough estimate of the largest eigenvalue is $(1 - \rho)^2 \Delta \theta^2 / 4$, so that I use

$$\Delta t_{\text{max}} = (1 - \rho)^2 \frac{\Delta \theta^2}{4}$$

(3.51)

and $\Delta t = \Delta t_{\text{max}} / 3$. Smaller values of $\Delta t$ were not found to be necessary. It was checked that when $\Delta t$ gets close to $\Delta t_{\text{max}}$, a numerical checkerboard instability develops on the inner equator circle of the torus ($\theta = 0$ line) and rapidly invades the rest of the system. The program that performs the bulk integration is called genus1, introduced in chapter 2 (it can evolve the NEMA on any genus-1 manifold, which includes any simply-connected, non-intersecting surface with periodic boundary conditions).
The LB operator in eq. (3.49) couples a lattice site only to its 4 nearest neighbors. This can be slightly improved upon by using an *isotropic* discrete LB operator, discussed in appendix A. The isotropic Laplacian has been useful in certain Euclidean model A and B simulations[30, 40]. For the torus, it gives smoother interface motion.

3.3.2 Interface formalism

Numerical integration of eq. (3.29) is in all ways as described in chapter 2, except that computation of $K_g$ and the element of arclength $\Delta s$ require knowledge of 10 new geometric quantities at every interface point: the determinant of the manifold metric $\sqrt{g_m}$, the 3 metric tensor components $g_{ij}$ of the manifold (since $g_{12} = g_{21}$), and the 6 components of the affine connection on the manifold (since $\Gamma^k_{12} = \Gamma^k_{21}$).

The first issue is then whether all these quantities, which are in general different at every lattice point of the surface, should be calculated only as needed for each interface at every time step as the interfaces move, or should be calculated only once at the beginning of the simulation (valid since the surface does not change in time). When only a few interfaces are being evolved in time, such that very little of the system surface is explored by the interfaces, it is more efficient to compute the geometric quantities at every time step at every interface point. When a very large number of interfaces is involved, covering most of the surface as I showed in fig. 2.4, it becomes more efficient to store the 10 arrays in memory at the beginning of the simulation. This is what I have done.

It is better, according to Thompson *et al.*[39], to use the discrete surface vectors $\hat{X}(u, v)$ to obtain the metric tensor numerically even if it is known analytically. Therefore the 10 geometric quantities were computed from their definitions, at each lattice point of the surface and stored for use during the integration of the NEIV equation. I recall those definitions here for convenience:

\begin{align}
  g_{ij} &\equiv \frac{\partial \hat{X}}{\partial u^i} \cdot \frac{\partial \hat{X}}{\partial u^j} \\
  \sqrt{g_m} &\equiv g_{11}g_{22} - g_{12}g_{12}
\end{align}

(3.52a) (3.52b)
3.3. NUMERICAL METHODS

\[ \Gamma_{ij}^k = \frac{1}{2} g^{kl} \left( \frac{\partial g_{jl}}{\partial u^i} + \frac{\partial g_{li}}{\partial u^j} - \frac{\partial g_{ij}}{\partial u^l} \right) \]  

(3.52c)

while the inverse metric components are obtained from the matrix inverse of \( g_{ij} \). Eq. (3.52) must be discretized. Denoting the mesh along \( u^i \) by \( \Delta^i \), the derivative along \( u^i \) of a field \( f \) at site \( u^i = n\Delta^i, n \) integer, is always approximated by a centered difference:

\[ \frac{df}{du^i} \to \frac{f(n + 1) - f(n - 1)}{2\Delta^i}. \]  

(3.53)

The 10 geometric quantities are rigorously known only at the lattice points of the manifold. The second issue is therefore how to best compute those 10 geometric quantities at the interface points, which almost never coincide with lattice points. In principle they should be calculated straight from the surface vectors \( \tilde{X}(u, v) \) interpolated at the interface points. But the computation of the 10 geometric quantities from \( \tilde{X}(u, v) \) requires many steps and slows down the integration of the NEIV equation by as much as an order of magnitude. This is why I implemented instead a 3-point two-dimensional linear interpolation of the pre-stored geometric quantities at each lattice point.

The interpolation proceeds as follows. First find in which surface lattice cell an interface point falls. Since the four vertices of a cell are rarely coplanar, each cell is divided into two triangles. The coordinates of the triangle to which the interface point belongs is then passed on to the interpolation routine. From these 3 points the routine creates a plane \( ax + by + cz = d \) and finds the \( z_0 \) for the given \( P(x_0, y_0) \) of the interface point. This method works well as long as the surface is smooth on short length-scales, and was found to give very satisfactory results in such cases (see for instance, fig. 3.6). More sophisticated methods are necessary only if extremely high accuracy is required when the interfaces are geodesically flat, but this is not the case in this thesis work.

The calculation of arclength requires some attention. Since interface points are described as two-dimensional \((u(\alpha), v(\alpha))\) pairs in the \( \alpha \) gauge rather than the arclength gauge, the metric tensor of the surface is required to compute the metric of
the normal gauge on the interface. The arc-distance $\Delta s$ between neighboring points at $\alpha$ and $\alpha + \Delta \alpha$ along the interface, as long as the distance is on the order of the surface lattice spacing, is given to a good approximation by

$$
(\Delta s)^2 = g_{ij} \left[ u'(\alpha + \Delta \alpha) - u'(\alpha) \right] \left[ u'(\alpha + \Delta \alpha) - u'(\alpha) \right].
$$

Since $\Delta \alpha \equiv 1$ between neighboring points, $\Delta s = \sqrt{g}$, and I define the metric of the normal gauge $\alpha$ as

$$
g = g_{ij} \left[ u'(\alpha + 1) - u'(\alpha) \right] \left[ u'(\alpha + 1) - u'(\alpha) \right].
$$

The metric $g_{ij}$ at $(u(\alpha + 1), v(\alpha + 1))$ must not be very different from that at $(u(\alpha), v(\alpha))$, so that an appropriately fine grid must be used for each surface.

Once $\sqrt{g}$ is obtained for all points of an interface, arclength derivatives of $u(\alpha)$ and of $v(\alpha)$ use the same expressions as in chapter 2 except for the first derivative. The expression for $u'$ at site $i$ is a weighted average of the left and right first derivatives,

$$
u'(i) = \frac{1}{\sqrt{g(i-1)} + \sqrt{g(i)}} \left( \frac{u(i+1) - u(i)}{\sqrt{g(i)} g(i-1)} + \frac{u(i) - u(i-1)}{\sqrt{g(i-1)} g(i)} \right)
$$

which makes use of eq. (2.1). This was found to have slightly smoother behavior than eq. (2.26) when the surface is not Euclidean.

In flat systems, an implicit method was tested on eq. (2.5b) and found to be extremely stable. The same implicit method also worked extremely well on the Euclidean curvature equations, eq. (2.24). However, a similar implicit method failed for eq. (3.43), due apparently to the non-linear terms coupled via the affine connection, so the implicit method was not discussed in chapter 2. A fully implicit method does not seem possible with eq. (3.29). The curvature equations (3.46) could come in handy in this respect, as they could be used to form a semi-implicit method together with eq. (3.29). This is discussed in appendix B.
3.4 **Numerical test of the NEIV equation**

To my knowledge, the Allen-Cahn interface equation has never been tested numerically for validity (though it was compared with experiment in Allen and Cahn’s article, [10]). The simplest configuration which allows this is a circular domain, but tracking interface points of a circular domain on a square grid is no easy task. Alternatively, a polar grid would require implicit methods. Ironically, the NEIV equation is far simpler to test.

Consider a torus manifold and a narrow, azimuthally oriented, band domain as initial condition. This is shown in fig. 3.3. The band domain has two interfaces, one on each side of the ‘oe’ line (see fig. 3.2). In a flat system without thermal noise, such a stripe domain does not move since the interfaces are flat. On the torus manifold, the interfaces have a constant, non-zero geodesic curvature and a non-minimal length. The band domain hence grows until it completely covers the torus. Denote the position of the top interface as $\theta_t$ with the parametrization depicted in fig. 3.2. The interface has a total curvature radius $R - r_0 \cos(\theta_t/r_0)$. Eq. (3.14) implies $K$ and $K_g$ are related by $K_g = K \sin(\gamma)$, with $\gamma$ the angle between the total curvature vector $\vec{K}$ and the normal to the surface. $\gamma$ is easily checked to be equal to
3.4. **NUMERICAL TEST OF THE NEIV EQUATION**

Let $\theta_I/r_0$. Therefore

$$K_g^{\text{torus}}(\theta_I) = \frac{\sin \left( \frac{\theta_I}{r_0} \right)}{R - r_0 \cos \left( \frac{\theta_I}{r_0} \right)}.$$  \hspace{1cm} (3.57)

Using $M = \xi = 1$, the NEIV equation predicts the interface speed $\nu$ varies with $\theta_I$ as $\nu(\theta_I(t)) = K_g^{\text{torus}}(\theta_I(t))$. A numerical simulation of the NEMA can be done starting with the band domain as initial condition, with the torus parametrized as in fig. 3.2. In $(\theta, \varphi)$ space, the configuration then evolves as a straight band expanding in width. This makes tracking the center of the interfaces trivial since both interfaces are straight, with constant $\theta_I$, and an azimuthal averaging can even be used to decrease the numerical error of the measurements. At each time step, the average position and velocity of the interface is therefore obtained. A parametric plot of the average speed for the top interface as a function of position $\theta_I$ appears in fig. 3.4, with time as the parameter, as measured in the numerical simulation. The NEIV theoretical prediction is also plotted. The simulation was done for two different grid sizes of the torus. Clearly, the velocity of the interfaces in the numerical simulation tends to the NEIV theoretical result as the surface grid is refined. This implies that the interface velocity equation is an excellent description of the continuum interface dynamics.

The same interfaces can be simulated with the interface formalism itself. Each interface of the band domain is created by hand and evolved with the program IFinteg. In all simulations, $M = \xi = 1$. The interface velocity is therefore given exactly by the geodesic curvature. At each time step the average position and geodesic curvature of each interface is taken. The parametric plot of geodesic curvature vs position $\theta_I$ is plotted in fig. 3.5, with time as parameter. The theoretical prediction eq. (3.57) is also plotted. The agreement with theory is excellent, even near the center of the torus where the geometry is most non-Euclidean and interface velocity is very small. In other words, the discretized NEIV equation is more accurate (i.e. is closer to the continuum TDGL solution) than is the discretized TDGL equation.

A second, useful, special case is that of a circular domain centered on a surface
3.4. NUMERICAL TEST OF THE NEIV EQUATION

![Figure 3.4: Interface velocity of stripe domain as a function of $\theta_I$ on the torus; $R = 40, r_0 = 20$. Numerical simulation with $\Delta \theta = 0.5$ (circles) and 1 (squares), and analytical prediction (line) are shown. At $t = 0$ the top interface is near $\theta = 60$ and the bottom one near $\theta = 66$.]

consisting of one axisymmetric bump,

$$
\bar{X} = \left[ u, v, Ae^{-(u^2 + v^2)/2\sigma^2} \right]
$$

(3.58)

I used $A = 20$ and $\sigma = 10$, with two different grids, one of $148 \times 148$ and one of $300 \times 300$, for a system size of $L_x = L_y = 120$. A circular domain centered in $(0, 0)$ on the surface has a radius $R = \sqrt{u^2 + v^2}$ as projected in the $xy$ plane, and a total curvature $K = 1/R$ directed towards $(0, 0)$. In this case $K_g = K \sin \gamma$ can be written
3.4. NUMERICAL TEST OF THE NEIV EQUATION

Figure 3.5: Geodesic curvature $K_g(t)$ of the two interfaces (top and bottom) of the band domain, as a function of position $\theta(t)$. Same domain configuration as in fig. 3.4, but evolved with $\mid$Fin$\mid$eg. Error bars are too small to be seen.

$K_g R = \sin \gamma$, with $\sin \gamma$ easily expressed as a function of $R$, i.e.

$$K_g R = \frac{1}{\sqrt{1 + \left(\frac{A}{\sigma}\right)^2 \left(\frac{R}{\sigma}\right)^2 e^{R^2/\sigma^2}}}$$

(3.59)

Fig. 3.6 shows $K_g R$ vs $R$ at several times during the numerical integration of eq. (3.29) by $\mid$Fin$\mid$eg. This is compared with eq. (3.59). It can be seen that the coarseness of the surface lattice, which inevitably affects the accuracy of the linear interpolation of the metric outlined in the previous section, affects the accuracy of the geodesic curvature computed by $\mid$Fin$\mid$eg. The most non-Euclidean part of the surface is around $R \simeq \sigma$, where the mesh size has the greatest effect. The largest tether length in the $148 \times 148$ run is 1.27 while it is half that in the $300 \times 300$ run. It is
3.4. NUMERICAL TEST OF THE NEIV EQUATION

Figure 3.6: $K_g R$ of a circular domain as a function of its radius $R$, as given by numerical integration on 2 different grid sizes; also shown is analytical prediction eq. (3.59). Error bars too small to be seen.
therefore advisable to mesh surfaces such that the longest tether has a length around 0.7. Finer grids should be unnecessary. Note that the time required to integrate the interface equations is independent of the surface grid, while it strongly depends on the grid for bulk integration. Since the interfaces must presently be obtained by first integrating the bulk equation for a short period of time, the grid cannot be made too fine. If the bulk step can be circumvented and interfaces created directly through some stochastic process, the lower bound on the mesh size may disappear. A possible technique to do this is discussed in appendix C.

3.5 Measurement of order in curved space

In chapter 1 I mentioned that the order-parameter structure-factor is commonly used as a measure of the domain order in Euclidean systems, because it is directly accessible experimentally. For curved surfaces, the order-parameter structure-factor cannot be calculated via a two-dimensional Fourier-transform as in Euclidean systems. Here, I first discuss why and give two alternatives. One of the two, the lambda structure factor, is measured for simulations on the torus. The curvature autocorrelation function introduced in chapter 2 must also be generalized for non-Euclidean systems. I discuss this after the lambda structure factor. Curvature correlations turn out to be far easier to measure and appear more sensitive to the surface's intrinsic geometry.

3.5.1 Lambda Structure Factor, or $\lambda$SF

In Euclidean systems, the order-parameter structure-factor can be obtained with the interface formalism if the order-parameter field is reconstructed from the interface shapes $\bar{R}$, which is an easy task. This reconstruction can be done in non-Euclidean systems as well. However, for a phase-ordering process constrained to a curved surface, how to measure the order-parameter structure-factor (OPSF) is a priori ill-defined. On the one hand, a two-dimensional structure-factor measures only the order within the two-dimensional manifold, implying it is insensitive to the extrinsic geometry of the surface. For the numerical experiments of NEMA, this is ideal
since the extrinsic geometry does not influence the dynamics, and such a structure-factor is valid for all surfaces with the same intrinsic geometry. On the other hand, experiments work in three-dimensional space, suggesting the extrinsic geometry of the surface would have to be eliminated from experimental measurements. But most importantly, the order-parameter is not defined outside the surface. How a three-dimensional structure-factor for a two-dimensional order-parameter is measured is not clear at this point.

A two-dimensional OPSF is more readily accessible, but cannot be computed as in Euclidean systems. Here I propose a method of measuring the OPSF within the two-dimensional manifold, which may possibly be related to some theoretical treatments of X-ray and neutron scattering from rough surfaces[41]. In Euclidean space, the OPSF is obtained by Fourier-transforming the order-parameter configuration to get $\phi_k(t)$, and computing $\langle \phi_k \phi_{-k} \rangle$. The Fourier transform, by definition, is a generalization of a change of basis from vectors to functions. It projects the order-parameter function onto a different basis, that of plane waves. The result is the "amplitude" of the order-parameter "vector" along each of the plane-wave "vectors".

However, plane-waves do not constitute a parametrization-independent basis in curved manifolds, for they are intimately linked to the translational and rotational invariance of Euclidean space. If the order-parameter field on a curved manifold is Fourier-transformed in the usual way by

$$\hat{\phi}(k_u, k_v) = \int \int e^{i\vec{k} \cdot \vec{R}} \, du \, dv$$

(3.60)

with $\vec{k} \cdot \vec{R} \equiv k_u u + k_v v$, it is clear that the result depends on the parametrization $(u, v)$ of the surface. I therefore make the following conjecture: eigenfunctions of the Laplace-Beltrami operator should be used instead of plane-waves. Indeed, in the Euclidean limit the LB operator is equal to the Laplacian, whose eigenfunctions are the plane-waves. The conjecture is sensible only because the eigenfunctions of the LB operator are known to form a complete orthonormal basis[37], and because its eigenfunctions and eigenvalues are independent of parametrization (simply because
the Laplacian of a scalar function does not depend on parametrization if given by the LB operator acting on the function).

Eigenfunctions and eigenvalues (called eigenpairs) depend only on the intrinsic geometry of a manifold. The Fourier transform becomes an "eigentransform", projecting the order-parameter "vector" onto the eigenfunctions $\psi_{in}(u^1, u^2)$ of the LB operator:

$$\hat{\phi}_{in}(t) \equiv \int \phi(u^1, u^2, t)\psi_{in}(u^1, u^2) \, dS$$  \hspace{1cm} (3.61)

where the $\psi_{in}(u^1, u^2)$ satisfy $\nabla^2_{LB}\psi_{in}(u^1, u^2) = \lambda_{in}\psi_{in}(u^1, u^2)$ and are assumed normalized. The subscript $i$ refers to an eigenspace of degeneracy $N_i$, the subscript $n$ to the $n^{th}$ eigenmode of that eigenspace. Therefore I denote the common value of the degenerate $\lambda_{in}, n = 1 \ldots N_i$, by $\Lambda_i$. I call the structure factor calculated with $\hat{\phi}_{in}(t)$ the $\lambda$-structure factor, or $\lambda$SF for short.

In Euclidean systems, eigenvalues correspond to the square-norm of wavevectors, so they are highly degenerate. This allows for a circular averaging of the structure-factor, equivalent to averaging the structure factor over all eigenvalues belonging to the same eigenspace. This seems like the reasonable thing to do in the non-Euclidean case as well. This yields a function of one variable:

$$S(\sqrt{\Lambda}_i, t) = \left\langle \frac{1}{A} \frac{1}{N_i} \sum_{n=1}^{N_i} |\hat{\phi}_{in}(t)|^2 \right\rangle$$  \hspace{1cm} (3.62)

where $A$ is the area of the surface. For an infinite continuous two-dimensional manifold, $\sqrt{\Lambda}_i$ is a continuous variable of dimension inverse length, equivalent to $|\vec{k}|$ in Euclidean space. Note that eq. (3.62) is completely independent of any choice of parametrization of the surface. It is therefore in principle an observable of the system.

I now discuss measurements of the $\lambda$SF on the torus manifold. The torus is particularly convenient since in the $(\theta, \varphi)$ parametrization (fig. 3.2), the metric tensor is diagonal (see eq. (3.48)), has one element constant, and the LB operator is separable. This implies the eigenfunctions can be separated into their $\theta$ and $\varphi$ components, with
the $\varphi$ part equal to one-dimensional plane-waves. I.e., an eigenfunction $\psi_{in}(\theta, \varphi)$ can be written as a product

$$\psi_{in}(\theta, \varphi) = \Phi_{k_\varphi}(\varphi)\Theta_\lambda(\theta).$$

(3.63)

with $\Phi_{k_\varphi}(\varphi) \equiv e^{ik_\varphi \varphi}$. Only the eigenfunctions $\Theta_{in}(\theta)$ along the $\theta$ direction need to be obtained and stored. However, to each $k_\varphi$ corresponds a different set of $\lambda$'s. The degeneracy of $\lambda$ therefore appears through the many different $k_\varphi$ associated with a given lambda. I used a sparse matrix eigenvalue solver called ARPACK[42]. Fig. 3.7 shows some typical $\Theta_{\lambda}(\theta)$ for $k_\varphi = 0.05$. The fourth plot shows three eigenfunctions, two of which have the same $\lambda$.

I used $\rho = 0.5$ and $r_0 = 20$ as these values allowed for a sufficiently long late-stage regime and discernible geometric effects while keeping $\Delta t_{\text{max}}$ reasonably large so that computation time for the simulation was within manageable limits. Other simulations with somewhat smaller $\rho$ were done but nothing qualitatively different was observed. The system size in $(\theta, \varphi)$ space is $2\pi r_0 \times 2\pi r_0 / \rho$, discretized with $N_\theta = 126$ and $N_\varphi = 252$ for $r_0 = 20$. This gives $\Delta \varphi = \Delta \theta = 0.99733$. Forty independent runs, each with different random initial conditions, were done with genus1 and the average $\lambda_{SF}$ computed. The total number of eigenvalues of the LB operator is equal to the number of lattice points used to discretize the surface, which was $126 \times 252 = 31752$. However, only the lowest 2000 were used. They were found by selecting the first 27 wavevectors $k_\varphi$ (which vary from 0 to roughly 0.7), and for each of these, computing all 126 eigenfunctions. This covers the range of interest of $0 < \sqrt{\Lambda} < 0.5$.

In fig. 3.8 the $\lambda_{SF}$ is plotted vs $\sqrt{\Lambda}$. The axes have been rescaled using the second moment $k(t)$ of the $\lambda_{SF}$ as a function of $\sqrt{\Lambda}$, i.e.,

$$k(t) \equiv \left( \int (\sqrt{\Lambda})^2 S(\sqrt{\Lambda}, t) \, d\sqrt{\Lambda} \right)^{1/2}$$

(3.64)

The overlap of the curves is excellent up to fairly late times ($t \sim 250$). Scaling definitely does not hold after $t = 500$, when finite-size effects set in. Inset to fig. 3.8 is a log-log plot of the tail portion of the rescaled $\lambda_{SF}$'s, clearly satisfying Porod's law $S(\sqrt{\Lambda}, t) \sim \sqrt{\Lambda}^{-3}$. This is a welcome result given that Porod's law follows from
3.5. MEASUREMENT OF ORDER IN CURVED SPACE

Figure 3.7: The $\theta$ part $\Theta_\lambda(\theta)$ of the eigenfunctions of the Laplace-Beltrami operator on the torus with $r_0 = 20, R = 40$, for $k_\varphi = 0.05$.

The existence of sharp interfaces[33], and therefore should not be influenced by non-Euclidean geometry as long as the surface is almost flat on the scale of the interface width.

Fig. 3.9 shows the reference length $L(t) = 2\pi / k(t)$ vs $t^{1/2}$ (labelled "torus ET"), for the $\lambda$SF’s of fig. 3.8. For comparison, $L(t)$ was also calculated for the flat system with same area (labelled "flat ET"). Clearly, $L \sim t^{1/2}$ during the whole late-stage regime. This agrees with a result discussed in a different context in chapter 4. To see around what time the geometry becomes important, I obtained $L(t)$ from a structure factor.
3.5. MEASUREMENT OF ORDER IN CURVED SPACE

Figure 3.8: Rescaled λSF on torus with \( r_0 = 20, R = 40 \), for times 33, 66, 133 and 266. Inset is log-log plot (base 10) of tail part, verifying Porod's law.

using a Fourier transform ("flat FFT" and "torus FFT" on the plot legend). Different parametrizations would give slightly different curves for "torus FFT", but the time at which the curves of the flat and torus systems deviate must be parametrization-invariant. At early times the FFT method gives \( L \sim t^{1/2} \) for both geometries, as it should since domains are not large enough to feel the non-Euclidean geometry. It is around \( t \sim 100 \) that domains are large enough so that measurements of their structure are affected by geometry. Finite-size effects only enter around \( t \simeq 500 \). This suggests the imperfect overlap of the rescaled λSF curves in fig. 3.8 could be due to geometry. Below, I discuss an alternate measure of order in the system, via interface curvature correlations, which I also compare with the λSF measurements.
3.5.2 Curvature correlations

The computation of curvature correlations is far less demanding than the $\lambda$SF. No eigenfunctions are needed, no projection of the order-parameter, and no eigenmode averaging is necessary. The measure of order of the convoluted interfaces on curved surfaces can take at least two forms. The first is the geodesic analog of the curvature autocorrelation introduced in chapter 2, namely the geodesic curvature autocorrela-
where as usual the angle brackets denote statistical averaging over many initial conditions. Vectors at two different points of a flat system (such as the curvature vector at two different points of an interface) can be scalar-multiplied only after implicitly parallel-transporting one of the two vectors to the position of the other. As discussed earlier, in flat space vectors are translationally invariant, so the path used in this transport is irrelevant to the result of the scalar product. This is not so in curved systems, where parallelism depends on the path taken. The scalar product of vectors coming from different points of a curved surface also depends on the path taken, even if the path is a geodesic, for uniqueness of geodesics is only guaranteed over very small distances. Therefore no scalar product can be used in \( G_{Ks} (s, t) \) and the sign of \( K_g \) must be kept. As discussed in chapter 2, the sign convention is unimportant as long as it is self-consistent for a given interface.

The second form of curvature autocorrelation involves the total curvature \( \tilde{K} \) of the interface, i.e. the curvature of the interface measured in three-dimensional space rather than in the two-dimensional space of the manifold. In three dimensions, the sign of the curvature of a line is obviously not well defined. The vector of curvature is the only way to properly represent the curvature of the interface, so a dot-product must be used in this case:

\[
G_{KR}(s, t) \equiv \left< \tilde{K}(0, t) \cdot \tilde{K}(s, t) \right>
\]  

(3.66)

Computationally the definitions have the same form as eq. (2.35).

Note that for flat systems, where the scalar product of \( \tilde{K}_g \) is straightforward, both definitions of curvature autocorrelation, the total and the geodesic one, give the same function. It is only in curved surfaces that they differ in very important ways. In the curved case, when an interface is stationary, it is so only because \( K_g \) is 0 everywhere and the interface is perfectly autocorrelated, from a two-dimensional point of view. This perfect autocorrelation stems from the fact that if a two-dimensional observer
within the manifold knew the position and orientation of the interface at one point of the interface, they would know the complete shape of the interface by solving the geodesic eq. (3.44) with the appropriate initial values. The only disorder remaining in the system is that arising from the relative position of the interfaces and from the extrinsic geometry of the geodesic lines on the surface. This is not measurable via a geodesic curvature autocorrelation function.

On the other hand, the total curvature of a convoluted interface at early times, when the domains are much smaller than any geometric length-scale of the surface, is equal to the geodesic curvature of the interface, in modulus. Under this condition the total and geodesic curvature autocorrelation function are nearly identical at early times (analytically they are rigorously equal). At late times, when an interface has become stationary, the total curvature at a point of the interface is the curvature of the surface along the direction of that interface (see eq. (3.14)). This total curvature autocorrelation function will therefore give information about the curvature autocorrelations of the surface along geodesic lines. This information is part of the extrinsic geometry of the surface and does not enter the geodesic curvature autocorrelation function. The two autocorrelation functions are therefore complementary for interface dynamics on curved surfaces. I make use of the total curvature autocorrelation function only in chapter 4.

The 40 torus runs of section 3.5.1 were redone with the interface formalism and used to measure $G_{K_s}^2(s, t)$. $G_{K_s}^2(s, t)$ for the torus is found to be identical to $G_{K}^2(s, t)$ for flat systems, chapter 2. But $G_{K_s}^2(s, t)$ seems more sensitive to non-Euclidean geometry effects than the $\lambda$SF, or perhaps sensitive to different aspects of it. The first intercept of $G_{K_s}^2(s, t)$ is used as a definition of the dominant interface length-scale $L(t)$ and is plotted in fig. 3.10. Also shown there is the same measurement for a flat system of size $2\pi r_0 \times 2\pi R$ (same area and size along $\theta$ as torus). In the time range 100 - 250 it is clear that $L(t)$ for the torus runs is slower than a $t^{1/2}$ power law. At later times, $L(t)$ increases faster: the domains are only slightly convoluted, the dip below 0 in $G_{K_s}^2(s, t)$ gradually moves above 0 and the first root of the correlation function moves towards positive infinity. To verify that the slowing down for $100 < t < 250$
is not an artifact of the NEIV, $G^2_{K_\tau}(s,t)$ was measured from the bulk configurations evolved with genus1. The slow down is indeed real, but is less important than with the interface formalism. This may explain why the A\SF missed detection of the slow down. However, NEMA dynamics is obviously richer than Euclidean model A, so it is also possible that both measurements are sensitive to different aspects of it.

3.6 Summary

In summary, I have generalized the model A equation to apply in any Riemannian manifold of any dimension. The only difference is in the diffusion term which is given by the Laplace-Beltrami operator rather than the usual Laplacian. An interface equation similar to the Allen-Cahn interface equation for flat model A systems
was derived for general two-dimensional Riemannian manifolds. The NEIV equation \( \ddot{\gamma} = M \xi^2 K \gamma \dot{\gamma} \) shows how interface motion in model A dynamics is coupled to the local geometry of the surface. The coupled curvature equations of chapter 2 were further generalized, and can be used to show that interfaces monotonically decrease in length with time, implying that metastable, stationary states are possible on curved surfaces. The numerical methods used to simulate phase-ordering kinetics with both bulk and interface formalisms were discussed and used on the torus surface, though the methods themselves are general. The discrete interface formalism was found to be closer to the continuum dynamics than the discrete bulk formalism. Two non-Euclidean measurements of order in the system were discussed: one uses the order-parameter field itself, via the \( \lambda \)SF, while the other uses the interfaces, via either the geodesic or total curvature autocorrelation functions. The \( \lambda \)SF was measured for the torus runs with the bulk formalism. It has the correct tail behavior and gives the correct power law for \( L(t) \) at early and intermediate times, but \( L(t) \) obtained from the geodesic curvature autocorrelation function indicate a slight slow-down not picked up by the \( \lambda \)SF.
Chapter 4

Applications

In this chapter, I use the various tools discussed and developed in the first three chapters to explore some of the consequences of non-Euclidean space on two-dimensional phase-ordering kinetics. First I look at the effect of surface Gauss curvature on the growth rate of domains. This is done by measuring the time rate-of-change of the interface of a circular domain in two different regions of the torus manifold. Following this I explore the phenomenon of metastable long-range disorder when the phase-ordering occurs on a corrugated surface. I do a simple calculation to show that metastable states occur only above a certain threshold ratio of amplitude of surface undulations to wavelength of undulations, test this numerically, and discuss some implications for self-affine surfaces such as those found in lipid bilayer systems as well as some implications when thermal noise is present. Finally, I discuss some simulation results of convoluted domains on a corrugated surface of the sinusoid type. Notably, I discuss the breakdown of dynamical scaling and of power-law growth, through the geodesic and total curvature autocorrelation functions, which differ on curved surfaces. Those two functions show several features not found in Euclidean dynamics. I discuss this as well as several measures of the dominant dynamical length-scale.
4.1 Domain growth vs surface curvature

A naive interpretation of the NEIV equation might suggest interfaces always evolve more slowly in non-Euclidean surfaces than in flat surfaces, because the geodesic curvature of a line is always smaller than its total curvature (see for instance eq. (3.14)). Closer inspection shows this to be wrong. This could have important consequences in the context of biological membranes, where the speed of reaction fronts could be influenced by the geometry of the membrane even when there is no extraneous coupling of the order-parameter (representing for example some lipid phase) to extrinsic geometry such as local mean curvature.

A surface has two orthogonal radii of curvature at every point. They can be of the same or opposite sign. The product of the two corresponding curvatures is the Gauss curvature, which is zero for Euclidean surfaces. The sign of the Gauss curvature is a characteristic of local intrinsic geometry: a region of surface which has a positive Gauss curvature looks like part of a sphere, while a region of surface with a negative Gauss curvature looks like a saddle.

Consider for the moment three different surfaces of constant Gauss curvature: one flat, one spherical and one hyperbolic. $K_G = 0$ everywhere for the flat surface, $K_G = \text{const} > 0$ everywhere for the spherical surface, and $K_G = \text{const} < 0$ everywhere for the hyperbolic surface. The three surfaces are homogeneous and isotropic[43]. Now consider the bulk equations for model A dynamics on Riemannian manifolds, eq. (3.10). The energetic cost of interfaces is represented by a square-gradient term, in eq. (3.1b), which becomes a diffusion term after the functional derivative is taken, as seen in eq. (3.10). This diffusion term is represented, for general Riemannian manifolds, by the Laplace-Beltrami operator. The physical process of diffusion seeks to eliminate gradients by randomly moving particles around. When particles are concentrated in one region of space, the result at time infinity is a uniform density (perhaps extremely small) of particles over all available space.

Now further consider a circular region of same geodesic radius on each of these three spaces. Such a circular region is constructed by picking a point and moving a
pen attached to that point by an unstretchable string of length $R$, keeping the string maximally extended. The area of the circle is least on the surface of positive Gauss curvature, and most on the surface of negative Gauss curvature. Diffusion therefore occurs faster on the hyperbolic surface than on the sphere, since particles have more space to spread out over in the former case and less in the latter case, relative to flat systems. This result has been known for some time[34]. Since interface motion in model A dynamics arises from the competition between the local free-energy term (which maintains bulk terms in one of two equilibrium phases and is not concerned with gradients) and a diffusion process, one can see how model A dynamics should be accelerated, at least for ovoid domains, in regions where the Gauss curvature is negative, and decelerated in regions of positive Gauss curvature.

To verify this, I ran simulations of ovoid domain interfaces in two distinct regions of the torus, as pictured in fig. 4.1, as well as in flat space, which I refer to as the third region. The simulation was done with Ifinteg, as tracking the domain boundaries with the bulk formalism is extremely difficult and not as accurate (the results were, however, qualitatively compared with runs by genus1). The inner region and the outer region of the torus do not have constant Gauss curvature but it is strictly negative in the inner region and strictly positive on the outer one. Recall that the torus, of small radius $r_0$ and large radius $R$, is parametrized as shown in fig. 3.2. For simplicity the domains have a radius of $2\pi r_0/4$ as measured in $(\theta, \varphi)$ space. They are therefore ovoid on the torus rather than circular and their perimeter differs in all regions. I denote the interface length by $L(t)$. Fig. 4.2 shows $L(t)$ in the two regions as well as in flat space. Inset are the corresponding circular domains as seen in the $(\theta, \varphi)$ space, for $K_G < 0$ and $K_G > 0$. It can be seen that at equal $L$ the slopes are different. This is clearer when looking at the decay rate of each interface, given by $dL/dt$. Since the interfaces have different $L$ at a given time in the three regions, $dL/dt$ for the three regions must be compared at equal $L$ rather than at equal time. Also, since it is known that in the flat case ($K_G = 0$) $dL/dt \sim L^{-1}$, $-dL/dt$ is plotted as a function of $1/L$ in fig. 4.3, on a logarithmic scale. Any difference due to geometry must show up as a vertical offset between the curves at equal $1/L$, since any circular domain of
Figure 4.1: Two ovoid domain interfaces on the real torus (mesh is transparent). The left domain is in the region $K_G > 0$, the right one in the region $K_G < 0$.

Figure 4.2: Typical $L(t)$ for circular domains in regions of negative, 0 and positive Gauss curvature. Real torus was used for the simulation. Inset are the corresponding circular domains in $(\theta, \varphi)$ space of the real torus, $r_0 = 20, R = 40$. 
any size in a flat system falls on the same curve on such a graph. Small values of $1/L$ correspond to early times, large values to late times. Clearly, $dL/dt$ is smallest in modulus in the region of $K_G > 0$, largest in the region of $K_G < 0$, and intermediate in flat systems ($K_G = 0$). As expected, the 3 curves merge as $1/L$ increases.

It is interesting to note that given the physical origin of the influence of Gauss curvature on domain growth in model A, much the same can be expected in model B as well as in other systems where diffusion and interfaces are present. Simulations of model B on a spherical surface were done recently by Taniguchi[21], but not over sufficiently long time-scales for any conclusive geometrical effects to be seen. Gauss curvature effects were however observed in simulations of crystal growth on toroidal geometries[44]. In the context of phase-ordering kinetics and phase-separation in lipid bilayers, where diffusion is known to play a very important role biologically, this Gauss curvature dependence could be of use in controlling the rate of such processes as protein confinement[45], diffusion, and shape change[46].
4.2 Long-range disorder and activated hopping

In section 3.2.3, I discussed how the NEIV equation implies interfaces could attain metastable states as their length could reach a local minimum in configuration space. The state is metastable in that the system free energy is not at an absolute minimum. This differs from flat surfaces where the dynamics stops only when the system free energy has reached its absolute minimum: interfaces invariably disappear if they are of closed (circular) topology, or straighten out if they are of open (percolating) topology. Surfaces with shorter-scale structure than the torus allow metastable interface configurations. The simplest example of such a surface is of the form \( f(x, y) = A \sin(2\pi x/\lambda) \sin(2\pi y/\lambda) \), but self-affine surface, random surfaces, and surfaces with many modes are also examples. They all belong to a class of surfaces which I refer to as corrugated. At early times, when the interface undulations are short compared to \( \lambda \), the ordering kinetics is essentially Euclidean. In this section, I show how at later times, when the domains are large in comparison to \( \lambda \), the bumps can act to slow down interfaces and eventually halt the evolution of interfaces. It is sufficient to consider only a typical "unit cell" of such a surface on which one domain is evolving.

Consider a surface which I describe by the set of vectors

\[
\vec{X} = [x, y, A \sin(kx) \sin^2(ky)]
\]  

where \( A \) is the amplitude of the surface and \( k \) the wavenumber. The \( \sin^2 \) along \( y \) is merely for continuity of the first derivative of geometric quantities, but only the qualitative shape of the surface matters here so no generality is lost. The system size is \( L_x = \lambda = 2\pi/k \) along \( x \) and \( L_y = \lambda/2 = \pi/k \) along \( y \), with \( \lambda = 40 \). A domain is assumed to have evolved to an ideal ovoid shape, whose interface circumnavigates the two bumps. This surface and the ovoid interface are shown in fig. 4.4. A top view with contours is also shown in fig. 4.6.

A simple calculation can be done to derive the minimal condition for the ovoid interface shown in fig. 4.4 to attain a stationary state, in the absence of thermal noise.
Consider the interface as it shrinks in length. It is forced to tilt itself, slowly moving up the outside of each bump. The total curvature at a point of the interface is the curvature as measured in three-dimensional imbedding space. Recall that the total curvature vector \( \vec{K} \) of a geodesic line on the surface is normal to the surface since \( K_g = 0 \) for a geodesic. Therefore a necessary condition for the interface to become stationary is for a configuration to exist where \( \vec{K} \), near both ends of the ovoid domain, is normal to the surface. A side view of the bumps with the interface is pictured in fig. 4.5 and a top view in fig. 4.6. This is not a sufficient condition however since all points along the interface must satisfy this criterion, but it does provide a minimal condition.

I now denote the vector going from \( O \) to \( P \) in fig. 4.5 as \( \vec{R} = [x, A \sin(kx)] \). The tangent to the surface in \( P \) is therefore \( \vec{t} = [1, Ak \cos(kx)] \), and the condition is \( \vec{R} \cdot \vec{t} = 0 \), leading to the transcendental equation \( 2x = -A^2k \sin(2kx) \). With the substitution \( x' = 2kx \) and \( \gamma = Ak \) this is written

\[
\frac{1}{\gamma^2} = \frac{\sin x'}{x'}. \tag{4.2}
\]
This equation has a solution only when $\gamma$ is larger than a certain value, for when $\gamma \to 0$ the LHS tends to $-\infty$ while the RHS is bounded between 1 and roughly $-0.22$. The condition is therefore $-1/\gamma^2 \gtrsim -0.22$, or

$$\frac{A}{\lambda} \gtrsim 0.34. \quad (4.3)$$

For $\lambda = 40$, eq. (4.3) implies interfaces can only attain a stationary configuration for $A \gtrsim 13.6$. Simulations for this surface and interface configuration find the threshold to be very close to 17.2, as shown in fig. 4.7. This is an error of roughly 20%, but is not surprising considering the interface can change its length via motion not represented in the side view of fig. 4.5. For instance, once point $P$ has $K_g = 0$, neighboring points might not yet satisfy the same condition, causing the interface to tilt further. However if $A/\lambda$ is large enough there comes a point past which the interface has no choice but to stretch if it is to hop over the bumps. It is not unusual for the presence of a second dimension to give a system more freedom, thereby softening constraints such as eq. (4.3).

The relationship between the amplitude $A$ of a bump and its wavelength $\lambda$ can be generalized to more complex surfaces. For instance, lipid bilayer membranes are
4.2. LONG-RANGE DISORDER AND ACTIVATED HOPPING

Figure 4.6: Top projectional view for fig. 4.5. Interface is thick line. $K$ is curvature vector, other symbols correspond to those in fig. 4.5. Line $AB$ is "pivot" line for interface. Circles are contours for the surface.

usually characterized by their bending rigidity $\kappa$. Such membranes form random self-affine surfaces whose average square width $W^2$ is given by [47]

$$W^2 = \frac{\ell^2 k_B T}{4\pi^3 \kappa}$$  \hspace{1cm} (4.4)

where $\ell$ is the size of the membrane as projected on the $xy$ plane. Alternately, $\ell$ can also be smaller than the membrane size, in which case only a subset of the complete membrane is being considered. The amplitude $A$ of eq. (4.1) can be approximated by the width $W$ for a given size, while the wavelength $\lambda$ can be approximated by $\ell$. As a consequence of eq. (4.3), phase-ordering may not occur at all if $\kappa$ is sufficiently small, as eq. (4.3) is then satisfied on all length-scales $\ell$. The condition for this is

$$\kappa \lesssim 0.07 k_B T.$$  \hspace{1cm} (4.5)

The model A dynamics on self-affine surfaces should therefore become slower as $\kappa$
is decreased from infinity. When $\kappa < 0.07k_B T$, domains might still form but freeze in their early disordered state. For real systems the hopping condition on $\kappa$ is not likely to be as simple, as for such small values of $\kappa$ the surface tension will usually be non-negligible. This changes the expression for $W^2$ and therefore the criterion for bump-hopping, but doesn’t otherwise introduce any new difficulty. Numerical simulations on self-affine surfaces require a very fine mesh so that the surface and interface metrics vary very slowly along the interface. Apart from large memory requirements, this is no impediment to integration via the interface formalism. However obtaining the interfaces from early-time integration of the bulk TDGL equation becomes prohibitively slow and must await the development of an alternate method to construct interfaces. One is proposed in appendix C.

It is important to note that eq. (4.3) is a necessary (minimal) condition for inter-
faces to get blocked around two bumps, and that blockage around a larger number of aligned bumps occurs at similar or larger values of $A/\lambda$. This can be seen by considering three aligned bumps instead of two. The middle bump constrains the interface to belong to the $xy$ plane in its vicinity, so that point $O$ of fig. 4.5 still lies between bumps rather than, say, at the center of the middle bump. The condition eq. (4.3) therefore still holds for any number of aligned bumps larger than or equal to two. This is sensible since what matters is $\lambda$, not the number of bumps.

However the situation is the reverse for bumps that are not aligned. Consider 4 bumps in a $2 \times 2$ array, for example as in fig. 4.9, and an interface that circumnavigates the four bumps. The top projectional view is shown in fig. 4.8. Point $O$, one of the
interface pivot points on segment $AB$, is still halfway between bumps, near the system boundary. The maximum-curvature director however no longer lies along $x$ but along the line $x = y$. Therefore, the pivot point normal to the interface at $P$ is at $O'$ rather than $O$, decreasing the apparent wavelength by a factor of $\sqrt{2}$, so that smaller amplitudes (by the same factor) are sufficient to trap the interface. The same kind of argument can be extended to, say, 6 bumps in a hexagonal configuration, bringing $O'$ yet closer to $P$. For a large surface consisting of a great number of bumps, gradually increasing $A$ from zero should cause interfaces to become stationary first around very large conglomerates of bumps. However the larger the conglomerate must be, the rarer the occurrence. As $A$ is increased further, smaller conglomerates of bumps trap closed interfaces, until an $A$ is reached where the domains stop growing when the dominant interface undulation length becomes comparable to the surface $\lambda$, leading to long-range disorder.

Since the interface is in a local minimum of length when it is a geodesic, changing the tilt of the interface in either direction causes it to relax back to its previous configuration only as long as the push is smaller than some critical amplitude. Point $C$ in fig. 4.5 is the point at which the length of the interface is at a local maximum, after which the length once more decreases, and the interface manages to hop over the bump and eventually disappear. Thermal noise allows for fluctuations in the length of the interface. Hence it is statistically possible for any bump amplitude to be overcome as long as sufficient time is given for the process to occur.

Once domains are formed, the bulk is in equilibrium and only the interfaces contribute to the system free energy $\mathcal{F}$. From eq. (3.10), this is

$$\frac{\mathcal{F}}{k_B T} = \frac{\xi^2}{2} \int g^{ij} \frac{\partial \phi}{\partial u^i} \frac{\partial \phi}{\partial u^j} dS$$\hspace{1cm} (4.6)

Once the interfaces are fully formed the interface width remains constant in time. This allows for the integral to be simplified, since the square-gradient term is non-
zero only over the width of the interface. It can therefore be written as

\[ \int \left( \frac{\partial \phi}{\partial n} \right)^2 \, dn \int dL \]  

(4.7)

where \( dL \) is an element of interface length while \( n \) is a coordinate perpendicular to the interface everywhere along it. Then along \( n \), the profile is approximately given by eq. (1.6) if the interface is gently curved. Using eq. (1.6), the integral of the profile yields \( \sqrt{8}/(3 \xi) \), and writing \( L \equiv \int dL \) as the total interface length,

\[ \frac{\mathcal{F}}{k_B T} = \frac{\sqrt{2} \xi}{3} L. \]  

(4.8)

If moving the interface from its stationary geodesic configuration to point \( C \) causes the interface to stretch by an amount \( \Delta L \), then the transition probability per unit time of hopping the bump is

\[ P_{\text{hop}} = e^{-\frac{\sqrt{2} k \Delta L}{4}} \]  

(4.9)

The value of \( \Delta L \) depends not only on \( A/\lambda \) but also on the exact geometry of the surface. When the surface contains a large number of such bumps, there should therefore be a new regime after the late-time regime, governed by thermal-noise activated hopping of the interfaces. This dynamics will undoubtedly be extremely slow unless \( A/\lambda \) is near its threshold value.

### 4.3 Egg-carton and corrugated surfaces

The kind of surface I am now going to use involves one or more sinusoid components, as in

\[ \bar{X} = [x, y, A \sin(2\pi kx) \sin(2\pi ky)]. \]  

(4.10)

Any surface of the form \( z = f(x, y) \) can be decomposed into a linear superposition of a finite number of such surfaces. Sinusoid surfaces form the basis for two important classes of surfaces. One of them is the class of surfaces which consist of a periodic array of bumps. Though not necessarily mathematically identical to a sinusoid surface, the
qualitative features of the latter, such as a dominant amplitude and a low number of modes, are indeed the same for all surfaces of that class and should provide some insight into model A dynamics on such surfaces. Experimentally, fluid as well as crystalline lipid bilayers are known to take this shape, also labelled "egg-carton", under certain conditions [17].

The second class of surfaces consists of corrugated surfaces, such as the random self-affine surfaces found in lipid membrane systems [47], discussed already in section 4.2. The chief difference from the first class is the larger number of underlying modes making up the surface, and the disordered undulations of the surface. However, the main features are the same, namely more or less ordered undulations of the surface on length-scales comparable to the domains in the late-stage regime. Therefore, model A dynamics on sinusoid surfaces should give some important clues for NEMA on both egg-carton-like and corrugated surfaces in general.

4.3.1 Numerical simulations

Many values of $A$ and $\lambda$ were used but here I discuss simulations done with $A = 4$ and $\lambda = 20$. The surface fits in a box of size $A \times 200 \times 200$. A unit cell of the surface can be seen in fig. 4.9, where the $z$ axis is stretched for clarity. The unit cell fits in a box of size $A \times \lambda \times \lambda$. A handy feature of surfaces consisting of a periodic array of smaller surfaces is that it is only necessary to access the unit cell for the geometrical quantities. This saves tremendously on memory and memory swapping.

Forty random numbers were generated and used as initial random seeds for the initial random order-parameter configurations, which were evolved using the non-Euclidean TDGL equation via genus1 from $t = 0$ to $t = 17$, as described in section 3.3.1. There are 227 points along $x$ and $y$, so $\Delta x = \Delta y = 0.44$. The surface lattice tethers therefore vary in length between roughly 0.44 and 0.71 and the time step in the Euler integration of the TDGL equation to obtain the initial random interfaces was a little over 0.015. The computing time required to generate one configuration to $t = 17$ is roughly 5 minutes on an HP735 with 120 meg RAM (physical+virtual). From the configurations, geninterf is used to extract the interfaces, which are then evolved with
4.3. EGG-CARTON AND CORRUGATED SURFACES

The interface formalism by Ifinteg. The program geninterf requires roughly three minutes per configuration for this system size.

A typical simulation takes the interfaces from \( t = 17 \) to 1000, but often \( t \) up to 10000 is desirable due to the interfaces slowing down. Integrating the bulk equations for the 40 configurations would require 8 days for a \( t = 1000 \) run, and 60 days for a \( t = 10000 \) run. The latter requires 8 hours when the interface formalism is used, half of which is used in evolving the interfaces from \( t = 1000 \) to \( t = 10000 \). Since the smallest mesh length varies as \( A/\lambda \), the required computation time grows as \((t_f - t_i)L_x L_y (A/\lambda)^4 \) for a bulk Euler integration on a sinusoid surface, where \( t_f - t_i \) is the total integration time and \( L_x, L_y \) the system size along \( x \) and \( y \). Small variations in surface parameters can lead to huge differences in required computation time for the bulk TDGL equations. For the interface equations, computation time only grows as \((t_f - t_i)^\gamma L_x L_y (A/\lambda)^2 \), with \( \gamma \) most likely close to 1/2. For corrugated surfaces, the mesh must be very fine and the advantage of an interface formalism becomes phenomenal.

The grid size requirements for the interface and bulk integrations are slightly different. For the interface formalism, the \( 227 \times 227 \) lattice used is appropriate given
the results of section 3.4 where it was found that the maximum tether length should be around 0.65. But this is smaller than necessary for the bulk integration since the maximum tether length can easily be 1 or slightly more. The bulk runs to obtain the initial interface configuration used the same mesh as for the interface formalism. Hence using a grid only 1.4 times coarser to obtain the initial configuration could lead to a four-fold gain in computation time for this stage. Using two grids may be feasible when the surface is analytically known such as is the case here, but for corrugated surfaces, which are usually generated from a stochastic PDE, this may not be possible or even desirable.

In fig. 4.10 are shown all the interfaces of a typical simulation of eq. (4.10) with \( A = 4 \) and \( \lambda = 20 \), at \( t = 17 \). Only 1/4 of the surface is shown. Already at such early times, the surface geometry has influenced domain formation and caused interfaces to form predominantly between bumps. This is not surprising given the initial dominant length-scale in model A dynamics in flat systems is around 10 at \( t = 17 \) (see fig. 2.7 of chapter 2), equal to half the wavelength of the surface bumps. The average value
Figure 4.11: Initial interface configuration on the sinusoid surface given by eq. (4.10) with \( A = 4 \) and \( \lambda = 40 \).

Of \( K_g \) of the interfaces at this early time is found to be around 0.1, showing that the initial dominant undulation length is also around 10, as for the flat system. Only 58 interfaces remain by \( t = 1000 \), all others having shrunk and disappeared. Fig. 4.16, taken from a larger system, shows what the interface configuration is like at such late times. A little under half the interfaces are closed (non-percolating). For the same sinusoid surface but with \( \lambda = 40 \), i.e. \( A/\lambda \) is half the previous value, the initial interfaces completely sample the whole surface. A contour plot of the first four cells of that surface are shown with the interfaces, in fig. 4.11. One can conclude that if larger ratios of \( A/\lambda \) are needed, the system size, \( A \) and \( \lambda \) should all be increased as well (keeping \( A/\lambda \) fixed to its larger value), so as to minimize the influence of geometry on the formation of domains.
4.3. **EGG-CARTON AND CORRUGATED SURFACES**

4.3.2 **Geodesic curvature correlations**

I recall the main features of the curvature autocorrelation function for Euclidean model A, discussed in chapter 2. The function is Gaussian at small arc-distances $s$ along interfaces. As long as finite size effects are not important, the $s$-intercept of $G_k^2(s, t)$ can be used as a measure of $L(t)$, the dominant length-scale in the system. This $L(t)$ follows a $t^{1/2}$ power law. The $G_k^2(s, t)$ taken at different times can be rescaled along $s$ with its respective $L(t)$. All rescaled $G_k^2(s, t)$ fall on one curve, showing that the curvature autocorrelation function obeys dynamical scaling. Finally, the superposition of the curves, especially near the minimum, shows that the relative degree of order of the interfaces remains constant in time.

The geodesic curvature autocorrelation was measured at various times for the interfaces on the sinusoid surface. The first arc-distance $s$ to satisfy $G_k^2(s, t) = 0$ was used as a definition of $L(t)$ to rescale $G_k^2(s, t)$. The rescaled $G_k^2(s, t)$ at several times between $t = 35$ and $t = 400$ are plotted in fig. 4.12. The most important feature of fig. 4.12 is the lack of superposition of the different curves, made especially clear by the decreasing amplitude of the minimum. Such a weakening of the dip amplitude would seem to indicate the envelope of the correlation function decays faster in rescaled space on the sinusoid surface than in a flat system. This should also affect the shape of the curve for $s < L$. A closer inspection of the curves for $0 < s < 1$ reveals a systematic drift towards the left as time increases, a phenomenon not seen in Euclidean systems, but consistent with the faster spatial decay (in rescaled space) of correlations as time increases.

On sufficiently short length-scales, the geometry appears Euclidean, so that $G_k^2(s, t)$ should also be Gaussian for sufficiently small $s$. Fig. 4.13 shows a log plot of $|\ln(G_k^2(s, t))|$ vs $s/L(t)$ for $s/L(t) < 1$. At early times, $G_k^2(s, t)$ is Gaussian for distances up to $s \lesssim 0.7L$. However by $t = 300$ $G_k^2(s, t)$ is Gaussian only over distances smaller than $0.2L$. The distance at which $G_k^2(s, t)$ becomes non-Gaussian can be understood in the following way. At early times the interface undulations are much smaller than the bumps in the surface. Therefore one must walk along the interface
Figure 4.12: Rescaled $G_{\kappa}(s,t)$ at different times between $t = 35$ and $t = 400$, for the sinusoid surface given by eq. (4.10) with $A = 4$ and $\lambda = 20$. Long dashed line is $t = 35$, short dashed is $t = 400$, intermediate times gradually move from one to the other.

over many undulations before seeing that the geometry has warped the interface. If this distance is larger than or on the same scale as $L$, $G_{\kappa}(s,t)$ does not differ much from Euclidean systems since the interface correlations decay faster than the surface variations. As the interfaces deconvolve, undulations span ever greater distances. A walk along the interface leads us to see the geometry is non-Euclidean before the undulation has changed. i.e., correlations in the interface decay over distances larger than that over which surface variations are significant. An unexplained result is the time-dependence of the variance of the Gaussian part of $G_{\kappa}(s,t)$. The variance was extracted by the same method as for the Euclidean case in chapter 2. In Euclidean systems, the variance was found to increase linearly in time (section 2.4). On the sinusoid surface with $A = 4, \lambda = 20$ however, the variance $\sigma^2$ is found to have a logarithmic dependence on time, as shown in fig. 4.14 where $\sigma^2$ is plotted vs $\log_{10} t$ for
4.3. EGG-CARTON AND CORRUGATED SURFACES

Figure 4.13: Log plot zooming on small-s region of fig. 4.12. Vertical is $\ln(G_{K}(s, t))$, horizontal is $s$, with same conventions as in fig. 4.12 for times. Two extra times are shown, $t = 500$ and $t = 700$.

times between $t = 17$ and 700. Linear regression gives $\sigma^2 = (63 \pm 3) \log_{10} t - (50 \pm 5)$, which means $\sigma^2 = (63 \pm 3) \log_{10}(t/t_0)$, with $t_0 = 6 \pm 2$. Without a theoretical reason to back this up, it is difficult to give too much importance to the logarithmic behavior, though a slowing-down fits in well with the discussion about activated hopping of section 4.2 and results of section 4.3.3. This measurement appears very reliable, because the error bars are small for small $s$ (equal to 0 at $s = 0$), and because it is independent of an arbitrary choice of intercept. For $A/\lambda$ half as small ($\lambda = 40$), the variance has a sublinear time-dependence, though not logarithmic.

The self-similarity of the dynamics in Euclidean model A implies that on any scale the domains always appear to involve the same amount of order. The breakdown of dynamical scaling as well as the decreased correlations imply that as one looks on
ever-increasing scales, the domains appear *more and more* disordered. This is an inevitable consequence of the inhomogeneity of the surface geometry: different parts of interfaces in different regions of the surface change in time at rates that depend on the local surface geometry, thus increasing the disorder in the system.

### 4.3.3 Interface density

I just showed that using $G_{K_s}(s,t) = 0$ to obtain $L(t)$ and rescale the $G_{K_s}(s,t)$ did not yield a universal geodesic curvature autocorrelation function. In flat systems, since all dynamical lengths follow the same time-dependence, $L(t)$ can be defined in a multitude of ways, for instance by $G_{K_s}(s,t) =$constant for any constant. All would give $L \sim t^{1/2}$, but with different coefficients, and all would give the same rescaled curvature autocorrelation function. The breakdown of scaling seen in the last section

---

**Figure 4.14:** Variance of Gaussian part of $G_{K_s}(s,t)$ vs time. Horizontal axis is $\log_{10} t$. Times used were $t = 17$ to $t = 700$. **
signifies that using different intercepts to obtain $L(t)$ does not give the same time
dependence for the resulting $L(t)$. This was verified numerically. In the previous
section I showed how the variance of $G_{Ko}^2(s,t)$ for small $s$ showed the dynamics was
slowing down in time, dramatically in the case of the $A = 4, \lambda = 20$ surface. This
therefore gives information about the dynamics on short length-scales. There exists
another measurement which does not depend on the arbitrary choice of an intercept
for $G_{Ko}(s,t)$, and should give information about the large-scale dynamics. It is the
interface density $\ell$ introduced earlier,

$$\ell(t) \equiv L_I(t)/A_s$$

(4.11)

where $L_I(t)$ is the total length of interfaces at time $t$ and $A_s$ is the area (constant
in time) over which the interfaces are dispersed. In order to compare $d\ell/dt$ for the
sinusoid surface and a flat system, I plot it as the logarithmic time derivative of the
interface density, or equivalently, the logarithmic time derivative of the total interface
length, as a function of interface density.

In fig. 4.15 I show a log-log plot of

$$-d(\ln \ell)/dt = f(1/\ell^2)$$

(4.12)

for the sinusoid surface $A = 4, \lambda = 20$. As reference, the points obtained from different
and completely independent series of runs in two other flat systems are also plotted
with a linear regression, dotted. The squares were obtained from a flat system of size
$200 \times 200$, while the triangles were obtained from the flat runs associated to the torus
simulations of section 3.5, where the area was $4\pi^2 R r_0 \simeq 31583$. As confirmed by the
overlap of the two flat system curves, the relationship (4.12) is independent of area
and quantity of interface. Any straight line on this plot indicates a power law in time
for $\ell(t)$. If the slope of the line is denoted $m$, the intercept $b$ and $\ell_0 \equiv \ell(0)$, then
straightforward integration yields

$$\ell(t) = (\ell_0^{2m} - 2m10^b t)^{1/m}$$

(4.13)
indicating $\ell \sim t^{1/2m}$ at late times. More negative $m$ thus corresponds to slower interface dynamics. The interface density has units of one over length, while the dominant length-scale in Euclidean model $A$ grows as $t^{1/2}$. Hence the scaling assumption valid for flat systems predicts $\ell \sim t^{-1/2}$, i.e. $m$ should be equal to $-1$ in flat systems. The linear regressions for the flat system curve gives $m = -1.07 \pm 0.01$, which means a power law of $-0.46$ instead of $-1/2$. This agrees with results of section 3.5 and chapter 2.

At early times (small values of $1/\ell^2$), model $A$ dynamics on the sinusoid surface also follows the same power law, as shown by the long-dashed linear-regression line. At later times however, one can distinguish a fairly long time regime, extending from $t \approx 250$ to $t \approx 1600$, almost a full decade of time, during which $m$ is more
negative. The dynamics has therefore substantially slowed down. Linear regression gives \( m = -1.70 \pm 0.04 \), corresponding to a power law behavior for \( \ell \) of \( t^{-0.3} \). No dependency on system size was found, insofar as a smaller system of size 100 \( \times \) 100 gives \( m = -2.0 \pm 0.1 \), suggesting the statistical error on the slope could be as large as 20 to 30%.

Though the slowing down is undeniable, the power law itself should be taken with caution, as power laws are notoriously difficult to extract reliably without a theoretical prediction as backup. It is also difficult to see how the complex interaction of interfaces with the surface as described by the NEIV equation should give a power law for \( \ell(t) \) rather than any other time-dependence, though it is also difficult to believe a sinusoid surface has just the right geometry to allow this, pointing to something more universal. In nature, power laws usually indicate an invariance with space or time scale. Therefore a power law behavior might be expected for self-affine surfaces, which are inherently scale-invariant, but hardly for a sinusoid surface. It is certainly a result worth further theoretical and numerical investigation.

At times larger than 2000, the dynamics becomes extremely slow. To check that this effect was real rather than due to finite-size effects, a larger system of size 300 \( \times \) 300 was simulated, and the exact same time-dependence found. In this regime, the interfaces have almost 0 geodesic curvature almost everywhere, but continue to hop the surface bumps, one by one, at a very slow rate. The interfaces waver between the regular array of bumps, so that only the two- and four-bump-like configurations of section 4.2 are relevant. This is shown in fig. 4.16. The hopping criterion for the four-bump configurations being \( A/\lambda \approx 0.25 \), it is not surprising that \( A/\lambda = 0.2 \), as is the case here, leads to this very slow but “determined” hopping in the very late-stage regime. After \( t = 2000 \), the \( K_g \) is very small and very short-length-scale fluctuations, due to numerical roundoff error, start becoming important. For this reason, the \( G_{ks}^2(s,t) \) becomes almost linear at small \( s \), a characteristic of sharply (though minutely) fluctuating systems. Therefore, the interface density may be the only reliable measurement for a time-dependent length in this late stage. Note that with the bulk formalism, 40 runs of 300 \( \times \) 300, \( A = 4, \lambda = 20 \) system to \( t = 10000 \) (as
was required here) would take over 60 days on an HP9000/735. It required 22 hours with IFinteg on the HP9000/735.

4.3.4 Total curvature correlations

When the interface correlations decay faster in space than the wavelength of the surface, $G_{\kappa}(s,t)$ is, as discussed earlier, qualitatively very close to $G_{\kappa,s}(s,t)$. Both correlation functions are not exactly equal as the scalar-product in $G_{\kappa}(s,t)$ produces a slightly stronger dip amplitude by about 20%, and the error bars are substantially smaller in $G_{\kappa}(s,t)$. Near the minimum they are as much as 3 times smaller. How-


4.3. EGG-CARTON AND CORRUGATED SURFACES

Figure 4.17: $G_{K}(s, t)$ for the sinusoid surface $A = 4, \lambda = 20$

However, while the average $K$ decreases in time, the $\bar{K}$ eventually starts increasing until it reaches values compatible with the surface $\lambda$. Therefore, $G_{K}(s, t)$ becomes very different from $G_{K_{e}}(s, t)$ at late times.

Fig. 4.17 shows $G_{K}(s, t)$ for the sinusoid surface, eq. (4.10) with $A = 4$ and $\lambda = 20$, at times between 17 and 10000 (17, 35, 50, 100, 150... 400, 500, 700, 1000, 1200, 1450, 1750, 2100, 2500, 3000, 4000... 10000). The curves for $t = 17$ and 10000 are labelled, with curves at intermediate times moving gradually from the former to the latter. For this surface, the wavelength of the bumps is not much larger than the initial dominant length-scale of the interfaces in Euclidean systems, so that a cross-over regime is not seen. The correlations simply increase in time, as more and more interfaces coarsen while being pushed between the bumps and adopting the surface's wavelength. At late times, the interfaces can be approximated by a random sequence of straight lines of length $\lambda/2$ and arcs of radius $\lambda/4$ and arclength $(\pi/4)\lambda/2$, as seen in fig. 4.16. This
Figure 4.18: $G^x(s, t)$ for the bimodal sinusoid surface eq. (4.14), $\lambda_1 = 40$ and $\lambda_2 = 50$. Times are $t = 17, 24, 35, 60, 100$ with the remaining times at same values as in fig. 4.17.

Explains the peak positions in $G^s(s, t)$ being roughly at integer multiples of $(\pi/4)\lambda/2$. At early times, only two peaks can be seen, with a third peak starting to appear. The emergence of the peaks at larger distances seems to coincide with the Euclidean regime in fig. 4.15. During the slow regime ($250 \lesssim t \lesssim 1600$), no more peaks appear. Those that have formed grow slowly in amplitude and saturate.

Fig. 4.18 shows $G^s(s, t)$ for a sinusoid surface consisting of two wavelengths rather than one:

$$\tilde{X} = [x, y, 4 \sin(2\pi k_1 x) \sin(2\pi k_1 y) + 6.25 \sin(2\pi k_2 x) \sin(2\pi k_2 y)]$$  

where $\lambda_1 = 2\pi/k_1 = 40$ and $\lambda_2 = 2\pi/k_2 = 50$. Hence both have $A/\lambda \simeq 0.1$ and the undulations of this surface are of longer wavelength than the previous one. In this case, the earliest curve looks exactly like $G^2(s, t)$ in flat systems and even has the dip.
positioned at roughly the same value, namely $s \simeq 10$. Already at $t = 24$ the interfaces are being pulled into the valleys between the bumps. This is the crossover regime. The curvature correlations settle into their definitive value around $t = 100$, close to $(\pi/4)\lambda/2$ with lambda around 40. Much longer wavelengths would be necessary to distinctly show the three regimes: a scaling regime at early times during which the amplitude of the minimum would not change, then the cross-over during which the dominant length-scale saturates to a dominant mode of the surface, and finally the late-time regime where the interfaces move more slowly and hop the bumps. Note that contrary to the $A = 4$, $\lambda = 20$ case, the very-late stage regime of fig. 4.15 never occurs because $A/\lambda$ is not large enough.

4.4 Summary

In this chapter, I discussed several new features of phase-ordering kinetics on curved surfaces through the phenomenological equations for the bulk order-parameter and the interfaces and through numerical simulations. I first showed how the growth rate of domains is affected by the Gaussian curvature of the surface in the vicinity of the domain interfaces, and explained what the physical reason is. Relative to Euclidean systems ($K_G = 0$), the rate decreases when $K_G$ becomes larger than 0, while it increases otherwise. Similar (though not identical) effects were found in another work[44] related to crystallization in curved geometries.

In the following section I investigated some of the consequences of the existence of metastable interface configurations on the phase-ordering kinetics, by considering closed interfaces on a surface consisting of two or more bumps. Based on simple geometrical considerations, I obtained an expression relating the amplitude of the bumps to their wavelength, eq. (4.3), giving the minimal condition for metastable states to exist. Simulations were done to test this and found to be in very good agreement. The consequences of having more bumps were discussed, leading to the conclusion that smaller ratios of amplitude to wavelength are sufficient to trap interfaces in metastable states. When applied to lipid bilayer membranes, eq. (4.3) suggests that if the bend-
4.4. SUMMARY

ing rigidity of a membrane is below the threshold of \(0.07 k_B T\), phase-ordering may not occur at all. Finally, the transition probability per unit time out of the metastable state was derived for the case where thermal noise is present.

In the last section, full advantage of the interface formalism was taken to study phase-ordering kinetics on corrugated surfaces, especially those made of a periodic array of bumps. Several of the simulation results would not have been possible with the traditional bulk formalism. I showed how all but one of the characteristics of Euclidean model A interface dynamics break down: dynamical scaling, power-law growth, \(t^{1/2}\) time-dependence for \(L(t)\), and constant degree of order on all scales. This was done by studying the geodesic curvature autocorrelation function, \(G_{\kappa s}(s, t)\), introduced in chapter 3. Its behavior at small distances was also investigated through the measurement of its variance since it still follows a Gaussian on short scales. The logarithmic time-dependence is much slower than the linear time-dependence found in Euclidean systems. I discussed how the interface density can give information complementary to the variance of \(G_{\kappa s}(s, t)\), and how it allowed for the break-up of the non-Euclidean dynamics into four time regimes rather than the three seen in Euclidean systems (not including the finite-size regime). Finally, I discussed measurements of the total curvature autocorrelation function \(G_{\kappa}(s, t)\). I showed how it is similar to \(G_{\kappa s}(s, t)\) at early times but gives different and complementary information about the dynamics at late times, and provides information about the surface geometry.

These "non-Euclidean" measurements are simple to use and versatile when an interface formalism is available. All three types of measurements show how a relatively simple extension of Euclidean model A dynamics to apply on curved surfaces such as lipid bilayers can lead to rich and interesting dynamics even in the absence of explicit coupling to geometric quantities like the local mean curvature, or other effects such as surface deformability.
Chapter 5

Conclusions and extensions

In this thesis, I investigated phase-ordering kinetics on static curved surfaces. I started from a well-known time-dependent Ginzburg-Landau equation known as model A in the classification of Hohenberg and Halperin and valid in flat two-dimensional systems, and generalized it for model A kinetics on curved surfaces. This approach is opposite to the mainstream work in the broader problem of pattern-formation in lipid membranes, where it has been found that the shape of the surfaces and the internal order of the membrane have an important influence on each other. It is also now recognized that such systems are inherently non-equilibrium and dynamical. While other work has focused on the study of how intra-membrane domain structures can explain observed membrane shapes, and more recent work has looked at the influence of a dynamical self-ordering field on membrane shape, my thesis starts from the opposite end, namely, how is the dynamics of pattern-formation affected when it occurs on a curved surface? My thesis was that important differences from the Euclidean model A dynamics would be found even when the surface was static and no external couplings were included, such as local mean curvature common to other work in this field. I believe I have successfully shown this. In so doing, I have also had to develop and implement an interface formalism for model A and introduce several new measurements which pertain to curved systems and become necessary in this case. Some new insight into Euclidean model A dynamics was gained as well. I now summarize in more detail the work and results of chapters 2 to 4, then discuss some
possible extensions to this work.

In chapter 1, I gave a brief overview of what pattern-formation is and what is known about the class of pattern-formation problems described by the so-called model A equation in flat systems. I also introduced the Allen-Cahn equation, and presented some of the extant work on the topic of pattern-formation and shape in lipid-membranes, which provided the motivation for the present thesis.

In chapter 2, I explained several important drawbacks of bulk equations such as the model A equation, and discussed several alternatives. I showed how evolving only the interfaces of a Euclidean model A system is more efficient and less time-consuming. For instance, in Euclidean systems the gain in computation time is at least five-fold and the memory requirements ten-fold less. An important verification was that interface dynamics exhibits the same characteristics of dynamical scaling and power-law growth as the bulk dynamics, as seen via measurements of the interface curvature autocorrelation function. The latter is trivially measurable from the interfaces. The form of this correlation function indicates that there is, unexpectedly, a dominant length-scale in the interface dynamics, not seen before in the usual order-parameter structure-factor (OPSF). In model A, domains of all sizes exist; however, a closer look indicates there is a dominant undulation mode along the interfaces. The nature of this dominant length-scale is therefore very different from that in the well-known model B, where domains exhibit clearly a typical time-dependent length-scale, also seen in the OPSF.

In this chapter, I also gave a fully covariant (i.e. parametrization-independent) derivation of the coupled curvature equations. I discussed several issues relevant to simulations of interfaces, such as periodic boundary conditions, interface generation with genus1 and geninterf, and how a position equation is better suited to simulations on curved surfaces than are curvature equations. The latter were used to obtain an approximate analytical expression of the curvature autocorrelation function. The prediction has the correct amplitude and power-law growth, but does not account for the observed oscillations of $G_\kappa(s,t)$ measured from simulations. I gave possible explanations for this.
In chapter 3 I generalized the model A equation to apply in any Riemannian manifold of any dimension. The physical difference is in the diffusion term which is given by the Laplace-Beltrami operator rather than the usual Laplacian. An interface equation similar to the Allen-Cahn interface equation for flat model A systems was derived for general two-dimensional Riemannian manifolds. The NEIV equation \( \dot{\nu} = M\xi^2 K g \hat{n} \), simple in appearance, shows how interface motion in model A dynamics is coupled to the local geometry of the surface in a non-trivial way, suggesting a rather wide variety of dynamics that depends on the surface. The nature of the NEIV equation makes it difficult to determine whether interface motion is always accompanied by a decrease in interface length. The coupled curvature equations, which were generalized for interfaces on curved surfaces, could be used to show this to be the case.

I discussed some of the numerical methods peculiar to non-Euclidean simulations with both bulk and interface formalisms, for instance, the computation of the metric tensor and affine connection. For the bulk formalism, an important addendum is appendix A, where I show how to properly discretize the Laplace-Beltrami operator with 8 or 12 neighbors. Simulations of the bulk equation on the torus manifold indicate that the discrete interface formalism is closer to the continuum dynamics than the discretized TDGL equation, yet it is much faster by two orders of magnitude. A simulation of a circular interface on a one-bump surface was done to assess how fine a grid is necessary to obtain good accuracy with the interface formalism. It was found that the maximum tether length should be around 0.7.

Three non-Euclidean measurements of order in the system were discussed, one for the order-parameter field itself, the \( \lambda SF \), and the other two for the interfaces, i.e., the geodesic and total curvature autocorrelation functions. All are parametrization-invariant generalizations of their Euclidean counterparts. The \( \lambda SF \) is extremely computer intensive, requiring the computation of many eigenfunctions of the Laplace-Beltrami operator, followed by a projection of the order-parameter on the eigenfunctions and finally an averaging over equal eigenvalues. The \( \lambda SF \) was measured for the torus runs with the bulk formalism. It gives the correct \( t^{1/2} \) power law for \( L(t) \) at early
and intermediate times and satisfies Porod's law, but $L(t)$ obtained from the geodesic curvature autocorrelation function indicates a slight slow-down not picked-up by the SF.

Finally, in chapter 4, I explored what I set out to do at the beginning of this thesis, namely some of the features of non-Euclidean phase-ordering kinetics not found in flat systems. I first showed how the growth rate of domains is affected by the Gauss curvature $K_G$ of the surface in the vicinity of the domain interfaces, and explained what the physical reason is. Relative to Euclidean systems ($K_G = 0$), ovoid domains evolve more slowly when $K_G$ is greater than 0, while they evolve faster when $K_G < 0$, an unexpected finding from the point of view of the NEIV equation. Similar (though not identical) effects were found in crystallization problems[44].

I then investigated some of the consequences of the existence of metastable interface configurations, discussed in chapter 3, on the phase-ordering kinetics by considering closed interfaces on a surface consisting of two or more bumps. Based on simple geometrical considerations together with the physics given by the two forms of the NEIV equation, I obtained an expression relating the amplitude of the bumps to their wavelength, eq. (4.3), giving the minimal condition for metastable states to exist. Simulations were done to test this and were found to be in very good agreement with the prediction. The consequences of having more bumps were discussed, leading to the conclusion that smaller ratios of amplitude to wavelength are sufficient to trap interfaces in metastable states. When applied to lipid bilayer membranes, eq. (4.3) suggests that as the bending rigidity is decreased, the dynamics slows down, and may not occur at all once $\kappa \lesssim 0.07k_BT$. Finally, the transition probability out of the metastable state was derived for the case where thermal noise is present, and one may expect a very late stage regime where interfaces move via thermally-activated hopping.

In the last section of chapter 4, interface dynamics was used to its fullest to study phase-ordering kinetics on corrugated surfaces, especially those made of a periodic array of bumps. Many of the simulation results would not have been possible with the non-Euclidean bulk formalism: typical simulations on sinusoid surfaces with the inter-
face formalism required one day, while 60 to 100 days would have been necessary with the bulk equation. I showed how all but one of the characteristics of Euclidean model A interface dynamics break down: dynamical scaling, power-law growth, $t^{1/2}$ time-dependence for $L(t)$, and constant degree of order on all scales. This was achieved by studying the geodesic curvature autocorrelation function, $G_{\kappa_s}(s,t)$, introduced in chapter 3. Its behavior at small distances was also investigated through the measurement of its variance since it still follows a Gaussian on short scales. The variance is a good measure of $L(t)$, and has, interestingly, a sublinear time-dependence in some cases and even logarithmic time-dependence for bumps of higher amplitudes. I discussed how the interface density can give information complementary to the variance of $G_{\kappa_s}(s,t)$, and how it allows for the clear distinction of a fourth time-regime characterized by extremely slow interface motion. The latter was explained in terms of bump hopping discussed in the previous section on long-range disorder and activated hopping. Finally, I discussed measurements of the total curvature autocorrelation function $G_R(s,t)$. I showed how it is similar to $G_{\kappa_s}(s,t)$ at early times but gives different and complementary information about the dynamics at late times, as it clearly shows how the interfaces follow the geometrical features of the surface.

Conclusions

The curvature autocorrelation function (CACF) has proven to be a valuable tool in my exploration of phase-ordering kinetics on curved surfaces. It provides information on the dynamics which may be buried in or not even be obtainable from the order-parameter structure factors (OPSF), yet it is far easier to compute than the two- or three-dimensional OPSF needed for domains on a curved surface. It showed that a dominant length-scale exists, not in the size of the domains (since the OPSF has its peak at 0 wave-vector), but in the undulations of the domain interfaces. This is a surprising and new result even in Euclidean model A, not seen before since order has been traditionally measured or computed via the OPSF.

In the introduction I argued that a dominant length-scale is not sufficient and
not even necessary for the dynamics to exhibit spatio-temporal self-similarity and dynamical scaling. Rather, all dynamical length-scales must have the same power-law growth, though these growth laws may have different coefficients. The CACF shows that a dominant length-scale is present in both Euclidean and non-Euclidean dynamics, yet dynamical scaling breaks down in the latter, at least in the case of corrugated, non-self-affine, surfaces. This is apparent not only in the non-Euclidean interface velocity equation, but in the dependency of domain growth on surface Gauss curvature $K_G$ and in the results of simulations on sinusoid surfaces. Part of the introductory statement is hence proven: the presence of a dominant length-scale to the dynamics is not sufficient to guarantee dynamical scaling. Moreover, the nature of the dominant length-scale seems to be different from that in the well-known model B, where a peak in the OPSF is clearly seen. It is not known whether the CACF for model B would show a dominant mode to the interface shape, which would imply there are two dominant length-scales in model B. Yet this would still be consistent with dynamical scaling, as long as both have the same power-law growth.

In flat systems, it was shown[9] that model A exhibits a zero-temperature strong-coupling fixed point towards which any model A system will "flow" if quenched to a temperature below its critical point. Therefore, the effect of thermal noise can be neglected in model A, its main effect being to increase $\xi$ in the TDGL equation, i.e. to slightly increase the width of interfaces. But since domains grow forever, interfaces invariably become one-dimensional on the scale of the domains. Thermal noise just retards the process. Results of sections 4.2 and 4.3 show that metastable states exists when a surface is "sufficiently" bumpy, meaning the ratio of amplitude of the bumps to their breadth is large enough. This has an important consequence for model A on curved surfaces: it indicates that the zero-temperature strong-coupling fixed point no longer exists. Indeed, the metastable states can trap interfaces and lead to long-range disorder. In a flat system, when the domains are ordering, the structure of the domains appears statistically the same on all length-scales. In other words, one can observe the system from any distance, it appears equally ordered. On curved surfaces, when interfaces become trapped, looking at the system from further
and further away shows ever smaller domains. In the infinite distance limit, the system appears to be completely disordered and homogeneous, with no long-range correlations. Therefore, thermal noise changes the quality of the dynamics once the domain interface undulations are on the same scale as the surface corrugations. In the very late-stage (and extremely slow) regime observed in simulations on sinusoid surfaces, interfaces will move via thermally-activated hopping, and it is only very slowly that the system will reach equilibrium. Similar conclusions are valid for model B. Two papers of relevance might be those by Delker et al. and by Parmar et al.[48].

Research in the area of shape and internal-structure coupling in lipid-bilayer membranes commonly couples the order-parameter field to the local mean curvature $C_m$ of the surface. For kinetics of phase-ordering, this produces a bilinear coupling term between $\phi$ and $C_m$, identical in form to the coupling term found in some Random-Field Ising Models (RFIM)[49], except that $C_m$ has medium- to long-range correlations for the case of surfaces while it is uncorrelated in RFIM systems. Short-range correlations in the RFIM have been investigated as well[50]. RFIM systems exhibit decelerated growth in the form of logarithmic growth laws of various kinds. Simulations on sinusoid surfaces show that as the amplitude of the bumps increases, domain growth at later stages is similarly decelerated, and can even show logarithmic time-dependencies. Yet, there is no explicit coupling term between the order-parameter and the surface curvature, only between the gradients of the order-parameter and the surface geometry. The non-Euclidean TDGL for model A on curved surfaces forms the basis of the more complex, RFIM-like models used in some lipid-bilayer membrane shape problems[19], yet already exhibits much of the richness found in RFIM systems.

Much can still be done to gain better understanding of the broader problem of pattern-formation on curved-surfaces. Hopefully the concepts, tools, methods and results discussed in this thesis will facilitate further work and progress in this important class of problems. I close this thesis by discussing some possible extensions of this work, first on a per-chapter basis, and finally to this research as a whole.
Extensions

In chapter 2, an important problem is that of obtaining an analytical expression for the curvature autocorrelation function, which would include the dip below 0 in $G_K(s, t)$. The presence of this dip is important as it indicates a dominant length-scale in the interface shape rather than simply a decay length for the curvature correlations.

In chapter 3, several issues still need attention. Theoretically, is there a better way to compute a covariant order-parameter structure-factor (OPSF) than by using eigenfunctions of the Laplace-Beltrami operators? Other projection methods exist which might require less modes to be kept, for instance Karhunen-Loeve decomposition or perhaps wavelet decomposition. A theoretical relationship between $G_K(s, t)$ and a covariant OPSF would be useful, as well as theoretical predictions for $G_K(s, t)$ and $G_K(s, t)$. The latter are likely to depend on the particular surface used. Another interesting question is what happens to Tomita's sum rule, which in Euclidean systems is a consequence of the randomness of interfaces. Since the surface decreases the randomness of the interfaces geodesically, but does the reverse in three-dimensional space, Tomita's sum rule will likely be changed in non-Euclidean systems. A very important issue both theoretically and numerically is whether the geodesic curvature autocorrelation function can be computed keeping the vector nature of $K_g$. This measurement is less sensitive to finite-size effects and produces useful information even for circular domains. It seems that parallel-transporting the curvature vectors along the interface would be the way to go to achieve this. Numerically, implicit integration methods for the interface formalism could speed up the simulations by making the time-step independent of the interface mesh-size, and a fast interface-generation algorithm, perhaps based on appendix C, would be handy in avoiding the sluggish bulk-integration step necessary to generate interfaces. Experimentally, methods for measuring the OPSF and both curvature autocorrelation functions (or their related structure factors) is of obvious interest if any experiments are to suggest particular future directions for this class of problems.

In chapter 4, theoretical predictions, for instance for the power-law or logarithmic
time-dependence of the variance of $G^2_{k,s}(s,t)$, would be useful in gaining a better understanding of those results. Mostly, simulations to test some of the predictions, for instance the dynamics on self-affine surfaces, and experiments would be interesting extensions. The question of whether dynamics on a self-affine surface re-instates dynamical scaling is important both theoretically and experimentally. Experimental verification of the dependency of domain growth on surface Gauss curvature would be welcome. Thermally activated hopping could be studied near the critical amplitude of $A/\lambda$.

In the broader context of pattern-formation on curved surfaces such as lipid-bilayers and surfactants, numerous directions are possible. For instance, what happens in model B? This could combine work from this thesis with that of Tim Roger's thesis[30]. What happens when the surface can change in time? What happens when the order-parameter is coupled to surface tension (surfactants) or bending rigidity (lipid bilayers), which act predominantly as a space-dependent chemical potential, somewhat related to the random-field Ising model? When the surface can change, domains will tend to warp the surface if coupling to local mean curvature is included. Yet activated hopping would tend to straighten the surface, leading to an interesting competition between the two effects. Model C is also interesting, since it couples the model A order-parameter to a conserved field, such as impurities, of obvious relevance to lipid membrane problems. A variation on this theme is surfactants which accumulate along the one-dimensional interfaces on the surface[25].
Appendix A

Discrete Laplace-Beltrami operator

There are two points I want to discuss in this appendix. The first is how to properly discretize the Laplace-Beltrami (LB) operator defined in eq. (3.7) and needed to integrate eq. (3.10), and the second is how to obtain an expression for the discrete isotropic LB operator. For the first point, I strongly recommend a reading of the book by Thompson et al. on numerical grid generation[39], where several aspects of discrete differential geometry are discussed in a very approachable manner.

A.1 “Nearest-neighbor” discretization

In discretizing the LB operator, one is confronted with a choice of discrete representations. Consider the one-dimensional LB operator in some arbitrary parametrization \( u \). The LB operator is then the second arclength derivative,

\[
\frac{\partial^2}{\partial s^2} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial u} \left( \frac{1}{\sqrt{g}} \frac{\partial}{\partial u} \right)
\]

where \( \sqrt{g} \), in this one-dimensional example, is defined by \( ds = \sqrt{g} \, du \). A centered-difference discretization begets

\[
\frac{\partial^2}{\partial s^2} f(i) \rightarrow \beta(i) \left[ (f(i + 1) - f(i))\beta(i + \frac{1}{2}) - (f(i) - f(i - 1))\beta(i - \frac{1}{2}) \right]
\]  

(A.1)
A.1. "NEAREST-NEIGHBOR" DISCRETIZATION

where \((i)\) stands for the function value at lattice site \(i\), and \(\beta(i) \equiv 1/\sqrt{g_m(i)}\). Thompson shows that \(f(i+\frac{1}{2})\) can be obtained by averaging \(f(i + 1)\) and \(f(i)\). The normal thing to do would then be to apply the same rule to \(\beta(i+\frac{1}{2})\). But he strongly cautions that numerical results by several authors show this to be dangerous and can introduce "spurious instabilities". His suggestion is therefore that field values at intermediate lattice sites can be obtained by averaging the field values at the two nearest neighbor sites if necessary, but to "never apply this rule to geometric quantities" such as the metric tensor. He also suggests that even when an analytic expression is known for the desired geometric quantities, they should be obtained numerically from the discretized manifold, rather than analytically.

A test of this was done, by using \(\beta(i+\frac{1}{2}) = (\beta(i+1) + \beta(i))/2\). Spurious instabilities were indeed observed in simulations of the non-Euclidean model A, eq. (3.10), on several non-flat surfaces. In fig. A.1, I show how an ovoid interface on the two-bumps surface discussed in section 4.2 has evolved, following a bulk integration of eq. (3.10) with this averaging scheme. The ovoid domain, after evolving to the awkward and obviously wrong configuration pictured there, freezes in that state. This effect was observed only when the geometry was strongly non-Euclidean, as for the two-bumps surface, as well as for corrugated surfaces discussed in chapter 4. It was not observed on the torus geometry.

A correct way of obtaining \(\beta(i+\frac{1}{2})\) for the discrete lattice is directly from the surface vectors \(\vec{X}\). For instance, if the metric component \(g_{kl}\) is required at lattice site \(i+\frac{1}{2}\), it should be computed directly from its discrete definition,

\[
g_{kl}(i+\frac{1}{2}) = \left| \frac{\vec{X}(i+1) - \vec{X}(i)}{\Delta u} \right|^2
\]

With this, the ovoid domain of fig. A.1 evolved properly.
A.2. ISOTROPIC DISCRETIZATION

I introduce the short-hand notation \( \phi(,) \) to indicate \( \phi \) at \((i, j)\), and \( \phi(\pm n, \pm m) \) to indicate \( \phi \) at \((i \pm n, j \pm m)\), with \( n, m \) integer and half-integer values. In flat systems, the Laplacian of a function \( \phi \) at \((i, j)\) depends only on the values of \( \phi \) at site \((i, j)\) and the 4 nearest-neighbor sites,

\[
\nabla^2 \phi(,) = \frac{\phi(+1,) - 2\phi(,) + \phi(-1,)}{\Delta x^2} + \frac{\phi(,+1) - 2\phi(,) + \phi(,-1)}{\Delta y^2}.
\]

(A.3)
A.2. ISOTROPIC DISCRETIZATION

In some cases, it is useful to include more than these (four) nearest neighbors and include the four next-nearest neighbors. Such a discretized Laplacian is referred to as isotropic. It is best done by writing the Laplacian as though the coordinate system were rotated by 45 degrees and ignoring the nearest-neighbor lattice sites. Denote the rotated length-preserving coordinate system by a prime, i.e.

\begin{align}
x' &= \frac{x + y}{\sqrt{2}} \quad \text{(A.4a)} \\
y' &= \frac{-x + y}{\sqrt{2}} \quad \text{(A.4b)} \\
\nabla'^2 \phi(i,j) &= \frac{\phi(+1,+1) - 2\phi(i,j) + \phi(-1,-1)}{2\Delta x^2} \\
&\quad + \frac{\phi(-1,+1) - 2\phi(i,j) + \phi(+1,-1)}{2\Delta y^2} \quad \text{(A.4c)}
\end{align}

One then uses the simple average of eq. (A.3) and eq. (A.4c) as the expression for the isotropic Laplacian of \( \phi \) at site \((i,j)\).

In flat systems, the improved approximation does not make much difference for model A dynamics, other than nudging the numerical value of the \( z \) in \( L \sim \xi^{1/z} \) closer to the continuum value of 2. The same procedure as for flat systems can be applied to the Laplace-Beltrami operator, except that more neighbors must be kept. I recall the definition of the Laplace-Beltrami operator, eq. (3.7b),

\[ \nabla^2_{LB} = \frac{1}{\sqrt{g_m}} \frac{\partial}{\partial u^i} \left( g^{ij} \sqrt{g_m} \frac{\partial}{\partial u^j} \right). \quad \text{(A.5)} \]

Consider the first term, which I denote \( L_{11} \),

\[ L_{11} = \frac{1}{\sqrt{g_m}} \frac{\partial}{\partial u^1} \left( g^{11} \sqrt{g_m} \frac{\partial}{\partial u^1} \right). \quad \text{(A.6)} \]

One way of discretizing \( L_{11} \) acting on a field \( \phi \) at lattice site \((i,j)\) when the mesh size along \( u_1 \) is \( \Delta u_1 \) is
If you now write out the three other terms of eq. (A.5), it is clear that the discrete nearest-neighbor $\nabla_{\text{LB}}^2 \phi(i,j)$ actually depends on the eight nearest neighbors of site $(i,j)$ for values of $\phi$ and on the eight half-integer neighbors of site $(i,j)$ for the values of the metric tensor components. This is schematized in fig. A.2. Once more, the rotated Laplacian is identical in form to the nearest-neighbor expression, except that
the metric in the rotated coordinate system must be used, and the site indices for $\phi$ change as in going from eq. (A.3) to (A.4c). For instance,

$$u' = \frac{u + v}{\sqrt{2}}$$

$$v' = \frac{-u + v}{\sqrt{2}}$$

$$\mathcal{L}'_{11}(\phi) \simeq \frac{1}{\sqrt{g''_{m}(\cdot)2\Delta u_{i}^{2}}} \times \left\{ g^{11'}(+\frac{1}{2}, +\frac{1}{2})\sqrt{g'_{m}(-\frac{1}{2}, +\frac{1}{2})} \left[ \phi(+1, +1) - \phi(,) \right] 
+ g^{11'}(-\frac{1}{2}, -\frac{1}{2})\sqrt{g'_{m}(-\frac{1}{2}, -\frac{1}{2})} \left[ \phi(-1, -1) - \phi(,) \right] \right\}. \quad (A.8c)$$

The metric tensor in the primed coordinate system is related to that in the non-primed via the usual tensor transformation rules,

$$g'_{mn} = g_{ab} \frac{\partial u^{a}}{\partial u'^{m}} \frac{\partial u^{b}}{\partial u'^{n}}$$

so that

$$g'_{11} = \frac{1}{2}g_{11} + \frac{1}{2}g_{22} + g_{12}$$

$$g'_{22} = \frac{1}{2}g_{11} + \frac{1}{2}g_{22} - g_{12}$$

$$g'_{12} = -\frac{1}{2}g_{11} + \frac{1}{2}g_{22}$$

$$g'_{m} \equiv g'_{11}g'_{22} - g'_{12}g'_{12}$$

while the contravariant components are obtained as usual by inverting the matrix for $g'_{mn}$.

The average of the two discretizations (the nearest-neighbor and isotropic) can now be used as an improved Laplace-Beltrami operator, coupling site $(i, j)$ to its twelve nearest neighbors. In the torus runs for instance, it was found that using this operator brought the scaling exponent $z$ slightly closer to 2 for $L(t)$ taken from the $\lambda$SF. It had a more drastic impact on measurements of the interface density as it smoothened the time-variations of the latter to a very noticeable degree.
Appendix B

Implicit numerical integration

Euler integration schemes, as used in the numerical simulations of this thesis in both the bulk and interface formalisms, do not have very robust stability properties. For flat systems, this is not much of an issue since the system lattice can be made fairly coarse, allowing the time-step to be of decent size, yet with satisfactory accuracy in the integration itself. In systems where the lattice is not uniform, such as on curved surfaces for the bulk formalism, or for interfaces in both flat and curved surfaces for the interface formalism, tiny time-steps have to be used in certain cases. I discussed in chapter 2 how the curvature equations (2.24) could be used to evolve interfaces in flat systems, due to translational and rotational invariance of the shape of curves. Though an explicit method was found to work well in both flat and curved surfaces when eq. (3.43) was used, it was found, though not discussed in chapter 2, that an Euler integration scheme failed miserably in regions of the interface where curvature was substantial, if eq. (2.24) was used. The distance between neighboring interface-points is not constant along an interface and becomes more uneven in those regions as time increases. The smallest mesh size limits the time-step in an explicit scheme like Euler. One work-around is to remesh the interface when the smallest mesh crosses some prescribed threshold, so as to keep the time-step from becoming small, thereby hopefully improving the stability of the Euler scheme. But at early times the interfaces evolve quickly so that an impractical number of remeshings is necessary. The only method found to work for eq. (2.24) was an implicit integration method.
applied to the diffusive part of eq. (2.24a).

The same implicit method was successfully applied to the velocity eq. (2.5b), but was not found to work in the non-Euclidean case, eq. (3.43), apparently due to the nonlinear coupling between the u and v via the affine connection. If this is the real reason, an implicit integration could still work on eq. (3.46) since the latter has exactly the same form as its Euclidean counterpart. As discussed in chapter 3, the position of the interface on the surface is needed when integrating eq. (3.46), due to the loss of translational and rotational invariance of the shape of the interface. This suggests a semi or fully implicit method for the position equation of the interface could be devised by combining an integration of eq. (3.43) with one of eq. (3.29), so I give a brief overview of how this could be done by first describing the implicit method used for eq. (2.24).

The implicit method for eq. (2.24) works as follows. First note that I work with eq. (2.24a) and eq. (3.46b) after substitution of eq. (3.46c) in them and setting \( M = \xi = 1 \), so that eq. (2.24) becomes

\[
\left( \frac{\partial K}{\partial t} \right)_a = \frac{\partial^2 K}{\partial s^2} + K^3 \quad (B.1a)
\]

\[
\left( \frac{\partial \sqrt{g}}{\partial t} \right)_a = -\sqrt{g}K^2 \quad (B.1b)
\]

Also note that the integration time-step may change as time increases. This is discussed below. The \( n^{th} \) integration step evolves an interface from \( t \) to \( t + \Delta t_n \), yielding \( K_{n+1}(s) \) and \( g_{n+1}(s) \). Now assume \( K_n(s) \) and \( g_n(s) \) are known. First obtain an approximation to \( K_{n+1}(s) \) by implicitly integrating only the \( K^3 \) term of eq. (B.1a). This is known as operator splitting[24]. I denote this intermediate integration step with the subscript \( n + \frac{1}{2} \)

\[
K_{n+\frac{1}{2}} = K_n + \Delta t_n K^3_{n+\frac{1}{2}} \quad (B.2)
\]

for each interface point. This is an equation of the form \( y^3\Delta t - y + b = 0 \) and is easy to solve. Good accuracy of the solution requires \( b^2\Delta t \ll 4/27 \). From this intermediate
result, use eq. (B.1b) to obtain

\[ g_{n+\frac{1}{2}} = g_n - 2\Delta t_n g_{n+\frac{1}{2}} K_n^2 \]  

(B.3)

for each interface point. Finally, use \( g_{n+\frac{1}{2}} \) to implicitly integrate the diffusive term at interface point \( i \):

\[
K_{n+1}(i) = K_n(i) + \Delta t_n K_n^3(i) + \frac{2\Delta t_n}{\sqrt{g_{n+\frac{1}{2}}(i-1)} + \sqrt{g_{n+\frac{1}{2}}(i)}} \times \\
\left[ \frac{K_{n+1}(i+1) - K_{n+1}(i)}{\sqrt{g_{n+\frac{1}{2}}(i)}} - \frac{K_{n+1}(i) - K_{n+1}(i-1)}{\sqrt{g_{n+\frac{1}{2}}(i-1)}} \right]
\]

(B.4)

This is the one-dimensional equivalent of eq. (A.7), but written for an implicit scheme. As can be seen, solving this nearest-neighbor discrete diffusion equation for \( K_{n+1} \) requires inverting a tridiagonal matrix. This is the typical signature of an implicit method. It also means that nonlinear terms are always problematic since matrices are linear operators. This is why I split the integration into two parts, one for \( K^3 \), and the other for the diffusive term alone, though other methods could be used as well[24].

Thankfully, powerful techniques exist for solving a matrix system of the form \( A\vec{x} = \vec{b} \) where \( A \) is a tridiagonal matrix and \( \vec{x} \) the vector to solve for. An interface is equivalent to a one-dimensional system with periodic boundary conditions. In such a case, \( A \) is cyclic tridiagonal, i.e., it is tridiagonal with 2 extra non-zero elements, \( A_{1N} \) and \( A_{N1} \) (assuming \( A \) is an \( N \times N \) matrix). Powerful techniques also exist for this case. I therefore solve eq. (B.4) in two steps. First, recast the cyclic tridiagonal problem in terms of a pure tridiagonal problem. This is done by using the Sherman-Morrison formula for cyclic tridiagonal systems (see e.g., section 2.7 of [24]). Secondly, solve the pure tridiagonal system, by using any of the multitude of pre-existing tridiagonal solvers in books or on the Internet. I adapted tridag from [24] for extra efficiency. Finally, use the solution to the pure problem to get the solution to the cyclic problem. These steps are well described in section 2.7 of [24] so I do not discuss this in any
further details. The bottom line is that eq. (B.4) can be solved rigorously even with periodic boundary conditions.

Obviously, an implicit method for a periodic boundary problem where there are nonlinear terms is more involved than the simple and straightforward Euler algorithm. In some cases however, it is well worth the effort. As a parenthesis, the method just described can also be extended to higher-dimensional problems, assuming proper operator-splitting is used for the diffusion term. For instance, a simple Euclidean two-dimensional diffusion problem has its LHS written as, for an implicit integration,

\[ \left[ I - \Delta t_n (L_{xx} + L_{yy}) \right] \phi_{n+1} \]  

where \( I \) is the identity matrix and \( L_{xx} \) is the discrete representation of \( \partial_{xx} \). The "vector" to solve for is the two-dimensional field \( \phi_{n+1} \) and the matrix \( A \) is the discretization of the term in parenthesis to the left of it. Note that \( \phi_{n+1} \) is the "rolled" out representation of the matrix, e.g. all matrix columns are sequential in computer memory. Expression (B.5) could be approximated by

\[ (I - \Delta t_n L_{xx})(I - \Delta t_n L_{yy})\phi_{n+1}. \]  

In this representation, \( A = A_x A_y \) can be decomposed into a block subset of \( n \times m + 2 \) independent, cyclic, tridiagonal matrices, for a system lattice of size \( n \times m \). The one-dimensional algorithm can then be applied to each matrix successively.

The implicit scheme outlined in the above paragraphs could be applied to the non-Euclidean curvature equations, eq. (B.1), to obtain \( K_g(s, n + 1) \). Once this is done, the velocity equation in terms of \( K_g \), eq. (3.29), could be integrated either explicitly or implicitly. If a discretization of the form given by eq. (3.56) is used for \( u' \) and \( v' \), then an implicit discretization of eq. (3.29) could take the form

\[ (I - \Delta t_n B)^{-1} \bar{R}_{n+1} \equiv A^{-1} \bar{R}_{n+1} = \bar{R}_n \]  

which is a tridiagonal matrix system to invert, but where each element of \( A \) is itself
a $2 \times 2$ matrix and each one of $\tilde{R}_n$, a doublet $(u,v)$, one for each for interface point. I.e., $B$ has the form

$$B = \begin{pmatrix}
    b_1^{2\times2} & c_2^{2\times2} & 0 & 0 & 0 & \cdots & 0 & a_N^{2\times2} \\
    a_1^{2\times2} & b_2^{2\times2} & c_3^{2\times2} & 0 & 0 & \cdots & 0 & 0 \\
    0 & a_2^{2\times2} & b_3^{2\times2} & c_4^{2\times2} & 0 & \cdots & 0 & 0 \\
    \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
    c_1^{2\times2} & 0 & \cdots & 0 & 0 & 0 & a_{N-1}^{2\times2} & b_N^{2\times2}
\end{pmatrix} \tag{B.8}$$

where the three matrices $a_i^{2\times2}$, $b_i^{2\times2}$ and $c_i^{2\times2}$ have the form

$$\frac{K_g \gamma}{\sqrt{g_m} \Delta s} \begin{pmatrix}
    -g_{11} & -g_{12} \\
    g_{12} & g_{22}
\end{pmatrix} \tag{B.9}$$

where $\gamma$ is a term different for $a_i^{2\times2}$, $b_i^{2\times2}$ and $c_i^{2\times2}$ and depends on the exact discretization used for $u'$ and $v'$, and where all quantities, i.e. $\gamma, K_g, \sqrt{g_m}, \Delta s$ and the metric tensor components $g_{kl}$ are evaluated at site $i$ and integration step $n + \frac{1}{2}$.

A final word must be said on the time-step used. For maximum efficiency, both explicit and implicit methods should seek to maximize the integration time-step at any time. For an explicit method such as Euler, this can be done by simply calculating a new $\Delta t$ at each time step based on the smallest mesh of the interface. For an implicit method, one method is to increase $\Delta t$ if the largest variation in the integrated variable is smaller than some prescribed threshold, and decrease it otherwise. This was found to work well for both eq. (2.5b) and eq. (B.1).
Appendix C

Interface generation

In this appendix I want to outline a possible method for generating a "random" set of interfaces which does not require integrating the bulk TDGL equation. The method exploits a simple relationship which exists between the curvature structure-factor (Fourier transform of the curvature autocorrelation function) and the dispersion relation of a linear stochastic partial differential equation.

Consider a scalar field $\phi(x,t)$. The following does not depend on the dimensionality of space so for simplicity I use $d = 1$. Now assume it obeys the stochastic PDE

$$\frac{\partial \phi}{\partial t} = L\phi(x,t) + \eta(x,t)$$  \hspace{1cm} (C.1)

where $L$ is a linear operator whose analytical form I need not know except that it must be linearly stable, and $\eta(x,t)$ is a Gaussian white noise. A physical example of such a system is a fluctuating membrane in two- or three-dimensional space, with dominating energy contributions coming from surface tension $\sigma$ and bending rigidity $\kappa$. In this case, $L$ is often written as $\sigma \nabla^2 - \kappa \nabla^4$.

Eq. (C.1) can be solved analytically by going to Fourier space. Denote the Fourier transform of a function $f(x)$ by $f(q)$ and that of $L$ by $-\omega(q)$. In the example above, $\omega(q)$ would be $\sigma q^2 + \kappa q^4$. In Fourier space, eq. (C.1) is written

$$\frac{\partial \phi(q)}{\partial t} = -\omega(q)\phi(q,t) + \eta(q,t)$$  \hspace{1cm} (C.2)
which has a formal solution

\[ \phi(q, t) = \phi(q, 0) \exp(-\omega(q)t) + \int_0^t e^{-\omega(q)(t-t')} \eta(q, t') dt' \]  

(C.3)

Several things need to be said about this solution. First, the dependence of mode \( q \) on its amplitude at \( t = 0 \) decays exponentially fast, with a decay rate \( 1/\omega(q) \). In this sense, only modes for which \( \omega(q) = 0 \) are sensitive to the initial condition. Secondly, I must impose the constraint that the noise is uncorrelated in time. This is usually a reasonable assumption. In this sense, \( \langle \eta(q, t)\eta(q', t') \rangle = 2\delta(q + q')\delta(t - t') \), where \( \langle \ldots \rangle \) is, as usual, an ensemble average. Finally, \( \omega(q) \) need not be known analytically, as long as it is known numerically. I assume for simplicity that \( \omega(-q) = \omega(q) \).

The interest of eq. (C.3) comes from the very simple relationship between the structure factor for \( \phi \), \( S(q, t) = \langle |\phi(q, t)|^2 \rangle \), and the dispersion relation \( \omega(q) \). Note that \( \phi(q, t) \) is in general complex, but because \( \phi(x, t) \) is real \( \phi(q, t)^* = \phi(-q, t) \). \( S(q, t) \) can be obtained by multiplying the solution for \( \phi(q, t) \) from eq. (C.3) by the solution for \( \phi(-q, t) \). This yields

\[
\langle \phi(q, t)\phi(-q, t) \rangle = e^{-2\omega(q)t} \langle \phi(q, 0)\phi(-q, 0) \rangle \\
+ \int_0^t \int_0^t e^{-\omega(q)(t-t')-\omega(q)(t-t'')} \langle \eta(q, t')\eta(-q, t'') \rangle dt' dt'' \\
+ \int_0^t e^{-\omega(q)(2t-t')} \langle \phi(-q, 0)\eta(q, t') \rangle dt' \\
+ \int_0^t e^{-\omega(q)(2t-t')} \langle \phi(q, 0)\eta(-q, t') \rangle dt' 
\]  

(C.4)

I assume that enough time has passed for any dependence on the initial condition \( \{\phi(q, 0)\} \) to have disappeared, since I am only interested in equilibrium solutions to eq. (C.1). Hence the first term on the RHS vanishes. The last two terms as well as the noise can safely be assumed uncorrelated to the initial condition. Using \( \langle \eta(q, t)\eta(-q, t') \rangle = 2\delta(t - t') \) allows me to carry out the two time integrations in the
remaining term, giving finally

\[ S(q,t) \equiv \langle \phi(q,t)\phi(-q,t) \rangle = \frac{1}{\omega(q)} \]  \hspace{1cm} (C.5)

Therefore, if the structure factor of a field \( \phi(x,t) \) is known at least numerically, so is \( \omega(q) \). Once \( \omega(q) \) is known, and if Gaussian and uncorrelated noise can be generated, then "random" configurations of \( \phi(x,t) \) can be generated which statistically follow the structure factor used as input. The initial PDE (C.1), as well as the linear operator \( \mathcal{L} \), are in fact not needed at all.

Now, replace \( \phi(x,t) \) by the physical quantity \( K(s) \), the curvature as a function of arclength. The relevance of this method to interface generation is incumbent upon several conditions which happen to characterize model A dynamics. First of all, the curvature autocorrelation for flat model A is known numerically at all times (during which interfaces exist) because of dynamical scaling. In fact, this autocorrelation function need only be known at one time, whereupon appropriate scaling of the arclength and, possibly, the amplitude of \( G_k(s,t) \) (if using \( G_k^1(s,t) \)), provides it at any other time. Secondly, a simple one-dimensional Fourier transform of the autocorrelation is all that is necessary to obtain the curvature structure factor \( S_K(q,t) \). Thirdly, it was seen in chapter 4 that interfaces, at early times, have a structure factor very close to that of Euclidean systems, as long as the geometry length-scale(s) are large enough compared to the initial domain length-scale. The latter is on the order of 10-20 in the dimensionless units I have been using throughout the thesis. Therefore, obtaining a set of initial interfaces on any surface requires integrating \( N \) independent equations like eq. (C.1) (one for each mode) and one inverse-Fourier transform to get the solution \( K(s) \) in real space. The whole operation should in principle take orders of magnitude less time than integrating the two-dimensional bulk equations for the order-parameter, does not require storage of any bulk configurations and circumvents the otherwise necessary step of extracting interfaces from bulk configurations with geninterf.

Needless to say, there are certain important difficulties which must still be worked
around before this scheme can be implemented for the purposes of studying interface
dynamics on curved surfaces at late times. The most important is the self-avoiding
nature of the interfaces. There is no guaranty that solutions generated from eq. (C.1)
are self-avoiding. If anything it is extremely unlikely. There are several possible venues
here. It could be that self-avoidance, though a necessary physical condition, may not
influence the evolution of an interface according to the interface equation eq. (3.29).
This is unlikely to be true on curved surfaces as interface "knots" could form around
corrugations and form knotted geodesics, which are not of physical relevance. This
problem is still interesting in its own right, however.

Another possibility is that the typical length-scale over which self-intersection
do not occurs is much larger than the typical domain length-scale. In this case, one
could simply generate (self-intersecting) interfaces and cut them into separate pieces
wherever they meet, without affecting in any drastic manner the structure factor for
the resulting set of interfaces. This could easily be checked by computing the resultant
curvature structure factor and comparing with the input one. More problematic
configurations could simply be discarded.

The second difficulty arises when the surface geometry length-scales are compa-
rable to the early-times domain length-scales. It was seen in chapter 4 that such a
setup affected the interface formation process in non-trivial ways. It is difficult to see
how this could be included in the present scheme. It may however be possible to use
a very fine interface mesh (by using a large maximum-$q$ mode), rescale $S_K(q,t)$ to a
time when the domain length-scale is much smaller than any geometry length-scale,
and generate interfaces for that time to evolve them with the interface formalism
(IfInteg) till the end of the run. The length-scale of the domains when the inter-
faces emerge from the early-stages of pattern-formation depends on the constant $\xi^2$
in the TDGL equation. Therefore, this corresponds to starting a bulk simulation
with a much smaller $\xi^2$ but a correspondingly larger $M$ (such that $M\xi^2 = 1$), and
then switching back to $M = \xi = 1$ when the dominant length-scale of the domains
approaches that of the surface geometry. It is unclear whether interfaces evolved in
this way would have the same statistical properties as those generated directly from
the non-Euclidean TDGL equation.
Bibliography


[9] A. J. Bray, Exact renormalization-group results for domain growth scaling in

motion and its application to antiphase domain coarsening, Acta Metall. 27,
1085 (1979).

mation of Amphiphilic Membranes, New York, September 1991, International
Workshop on Amphiphilic Membranes, Springer.

and Fission of Vesicles, Biophysical Journal 65, 1396-1403 (October 1993).
A. J. Verkleij, C. J. A. V. Echteld, W. J. Gerritsen, P. R. Cullis, and D. Kruijff,
The lipidic particle as an intermediate structure in membrane fusion processes
and bilayer to hexagonal HII transitions, Biochim. Biophy. Acta 600(3), 620-4
(August 1980).
H. Gaub, R. Büschl, H. Ringsdorf, and E. Sackmann, Chemical Physics of Lipids
37 (1985).

[13] E. Sackmann, Membrane bending energy concept of vesicle- and cell-shapes and
shape-transitions, FEBS 346, 3-16 (1994).


[15] J. A. N. Zasadzinski and M. B. Schneider, Ripple wavelength, amplitude, and
configuration in lyotropic liquid crystals as a function of effective headgroup size,

[16] C. Gebhardt, H. Gruler, and E. Sackmann, On domain structure and local
curvature in lipid bilayers and biological membranes, Z. Naturforsch. C 32(7-8),
581-96 (July/Aug 1977).


