A Moving Mesh Finite Element Approach for the Cahn-Hilliard Equation

Peter Westwood MSc MoSaIC

August 22, 2010

Acknowledgments

Firstly, I would like to thank my supervisor Professor Mike Baines for his greatly appreciated support along with his endless enthusiasm and ideas. It has been a pleasure to work with him throughout the year. I would also like to thank my fellow Maths students for their support in what has been a great and enjoyable year. Also, I would like to thank my family for encouraging me to further my studies and supporting me throughout. It is greatly appreciated.

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Signed.....Date.....

Abstract

This dissertation is a feasibility study on the use of a velocity-based moving mesh finite element method, based on a conservation principle constant in time, to approximate the dynamical behaviour of the Cahn-Hilliard equation. The method is implemented in both 1-D and 2-D. In the 1-D case, both a mass monitor and an arclength monitor are assessed and compared, with only a mass monitor assessed in the 2-D model. The method and dynamics are investigated in detail and compared to alternative methods used to approximate the Cahn-Hilliard equation.

Contents

List of Figures

Chapter 1

Introduction

In this section, I introduce the Cahn-Hilliard equation, its general form and the methods applied in previous papers. I will also refer to a reduced version of the equation which I use in the project, along with the relevant imposed boundary conditions. The equation is investigated in both 1-D and 2-D form, generating the equation and imposing the relavent boundary conditions for both cases.

1.1 The Cahn-Hilliard Equation

Phase field models have become a large area of research in mathematical modelling. The Cahn-Hilliard equation (7)(8) in particular is a well known phase field model. In phase field based models, sharp interfaces are represented by very thin transition layers in order that the phase field varies continuously over these tansition layers, and yet remains uniform over the bulk phases .

This type of model is ideal when approaching equations involving several physical parameters which also generates steep fronts over the transition layers. Since the interfaces are replaced by thin transition layers or diffuse interfaces, in order to capture these dynamics correctly, high resolution of very thin layers is required. The Cahn-Hilliard equation describes spinodal decomposition of a continuous homogeneous fluid mixture quenched below its critical temperature, where it then exhibits instability and develops multi-phase behaviour. Spinodal decomposition is defined as the process by which a solution of two or more components of a mixture separate into distinct regions or phases, ultimately displaying a discontinuous inhomogenous mixture. This process occurs uniformly throughout the mixture and the orientation of the different phases are found not to be composed in a random fashion. Instead, the dynamics appear to adhere very tightly to those predicted by the Cahn-Hilliard equation. The study of multi-phase multi-component fluids is a vast field of research with a variety of applications in engineering, most notably polymer science where interfacial phenomena, coextrusion of polymers and phase separation under shear are key processes. Hence, exhaustive research is being carried out in mathematical modelling to understand and capture the dynamics of the Cahn-Hilliard equation efficiently with high resolution and accuracy.

The basis of the problem that John W. Cahn and J. E. Hilliard were reviewing in (7)(8), was that of minimizing the free energy functional of a fluid mixture. Letting $\phi(\mathbf{x})$ represent the relative

concentration of two components subject to conservation of mass,

$$\int_{\Omega} \phi(\mathbf{x}) d\mathbf{x} = \phi_m |\Omega|, \tag{1.1}$$

where ϕ_m is a constant corresponding to the spatially uniform mixture and Ω is the region of space occupied by the system. Let $f(\phi)$ represent the bulk energy density, taken as a symmetric double well potential given by,

$$f(\phi) = \frac{1}{4}(1-\phi)^2, \tag{1.2}$$

containing two minima which correspond to the two stable phases of the fluid. With the thickness of the interfacial layers denoted by ϵ, then the free energy functional is given as,

$$\mathbf{H}[\phi] = \int_{\Omega} \{f(\phi(\mathbf{x})) + \frac{1}{2}\epsilon^2 |\nabla\phi(\mathbf{x})|^2\} d\mathbf{x}, \tag{1.3}$$

where $\nabla\phi(\mathbf{x})$ represents the surface energy of the fluid. The chemical potential, μ, is then defined as the first variation of the free energy functional, $\mathbf{H}[\phi]$, given as

$$\mu(\phi) = \frac{\partial \mathbf{H}[\phi]}{\partial \phi(\mathbf{x})} = f'(\phi(\mathbf{x})) - \epsilon^2 \nabla^2 \phi(\mathbf{x}), \tag{1.4}$$

where $\mu = constant$ gives, ultimately, the equilibrium state.

The Cahn-Hilliard equation was developed by Cahn and Hilliard to generalize the problem of minimizing the free energy functional into a time-dependent situation by approximating interfacial diffusive fluxes as being proportional to chemical potential gradients and enforcing conservation of the field. This is shown by the equation

$$\frac{\partial \phi(t,\mathbf{x})}{\partial t} = -\nabla.\mathbf{J}, \tag{1.5}$$

where $\mathbf{J} = -\lambda(\phi)\nabla\mu$, with $\lambda(\phi) > 0$ the mobility or Onsager coefficient and $\mathbf{J}$ the interfacial diffusive flux. The mobility can be adapted to model phase separation dynamics controlled by either bulk diffusion ($\lambda \to 1$) or interface diffusion ($\lambda \to (1-\phi^2)$). In our case, we are using the basic case with a mobility of 1. This then gives the Cahn-Hilliard equation,

$$\frac{\partial \phi(t,\mathbf{x})}{\partial t} = \nabla.[\lambda(\phi)\nabla\mu(\phi)], \tag{1.6}$$

with the chemical potential $\mu(\phi)$ given by equation 1.4.

There are many industrial applications of the Cahn-Hilliard equation. These mostly arise in the study of binary fluids (mostly alloys), interfacial fluid flow and in mixtures in polymer science. Therefore, due to applications of the equation being fundamentally linked with fluids and fluid flow, a particular area of interest in the field of research currently is the coupling of phase separation of the Cahn-Hilliard equations to the Navier-Stokes equations in fluid dynamics. This is a growing area of theoretical and practical interest since many numerical methods have been tried and tested for both the one- and two-dimensional cases. However, the three-dimensional case has been researched very little despite being of great interest to industry for obvious practical implications. A variety of difficulties are associated with the Cahn-Hilliard equation.

1.2 Difficulties

There are multiple problems associated with the Cahn-Hilliard equation. Initial treatment of the interfaces is difficult since the positions of these interfaces are unknown prior to the initial state. It is also quickly apparent that no analytic solution of the equation exists with a complete well-mixed mixture, as the phase field showing the relative concentration is an entirely random function ranging from -1 to 1. From direct inspection of the equation, one can see that it contains both a fourth derivative and a Laplacian acting on a nonlinear term, $f(\phi)$. The use of finite elements to solve this equation generates C^0 solutions, smooth within the elements themselves and continuous across their boundaries, which are unable to represent fourth order spatial derivatives (24). Furthermore, Laplacians acting on nonlinear terms are difficult to model and can involve expensive computational techniques to solve. In order to bypass this problem, *Ceniceros, et al.* (1) decomposed the fourth order Cahn-Hilliard equation into a coupled system of second order equations and removed the leading linear terms to enable a semi-implicit time stepping technique. This reduced coupled conservative system, introduced in (10), is the system to which I will be applying my moving mesh method. I will introduce this in detail in the following section.

A further problem encountered when attempting to model the equation, is to provide an adaptive mesh method which captures the fast dynamics involved in the formation of the steep gradients present at phase-change boundaries (or interfacial layers) generated by the equation. This is discussed in greater detail in section 2.

1.3 Previous Approaches

Many different approaches have been taken to model the Cahn-Hilliard equation. A range of fully and semi-implicit (10) finite difference methods and both continuous (12) and discontinuous (11) Galerkin-based finite element methods has been used in previous studies to try and capture the fast initial dynamics of the equation. Explicit approaches have often been discarded due to the prohibitively small timestep required to capture the initial dynamics. These methods all involved discretizing the time intervals, either implicitly or explicitly, and applying the solution to an adaptive mesh.

Of greatest interest was the paper by *Ceniceros, et al.* (10) using a semi-implicit finite difference method. The area of interest was not so much the method, but more the coupled equations used, derived in a previous paper by *Ceniceros, et al.* (1). By reducing the equation to a coupled system of second order ODEs, high computational costs were averted. This coupled system along with a unity mobility is much simpler and more adaptable to alternative methods.

All previous methods have encountered problems. A basic but rather important requirement is that the method needs high resolution in both space and time. This requires the thickness of the interfacial layers, ϵ, to be be very small relative to the size of the domain. A common conclusion from previous papers employing fixed grid adaptive methods, was that in order to accurately resolve the phase field the mesh size of the finest level needs to be $O(\epsilon)$, which is the smallest scale associated

with the width of the transition layer. Most of the previous methods used (certainly in the case of fully implicit methods) involved the use of expensive computational techniques, which were required to maintain at least 2nd order accuracy in both time and space, when taking account of the nonlinear terms and high order derivatives such as the biharmonic operator.

For finite time discretization methods on fixed meshes the scheme used was also required to be A-stable in the discretizations, otherwise the high frequency components of the equation were not smoothed. This was investigated in greater detail in conjunction with a continuous Galerkin finite element method by *Garcke, et al.* (12), and it was shown that Θ-splitting schemes, originally proposed by *Strang* (23) and applied to Navier-Stokes equations by *Bristeau, Glowinski & Periaux* (5), were suitable for the time discretizations with $\theta = 1 - \frac{\sqrt{2}}{2}$ for 2nd order accuracy.

The use of high cost iterative solvers to solve the nonlinear term and biharmonic operator is difficult to avoid. However, in the reduced system presented by *Ceniceros, et al.* (10), iterative solvers were no longer required to solve the nonlinear term and biharmonic operator. Taking a constant mobility, $\lambda = 1$, gives the following reduced system,

$$\frac{\partial \phi(t,\mathbf{x})}{\partial t} = \nabla^2(\tau\phi - \epsilon^2\nabla^2\phi) + g(\phi), \tag{1.7}$$

for $\mathbf{x} \in \Omega$, where $g(\phi) = \nabla^2(f'(\phi) - \tau\phi)$ and τ is a constant given by, $\tau = \max f''(\phi)$. Introducing the auxilliary variables,

$$\varphi_1 = \phi \text{ and } \varphi_2 = \tau\phi - \epsilon^2\nabla^2\phi, \tag{1.8}$$

the modified equation 1.7 gives rise to the system,

$$\frac{\partial \varphi_1(t,\mathbf{x})}{\partial t} = \nabla^2\varphi_2 + g(\varphi_1) \tag{1.9}$$

$$\varphi_2(t,\mathbf{x}) = \tau\varphi_1 - \epsilon^2\nabla^2\varphi_1, \tag{1.10}$$

with $g(\varphi_1) = \nabla^2(f'(\varphi_1) - \tau\varphi_1)$. This coupled, conservative Cahn-Hilliard equation is the system I will be using, applying a finite element-based moving mesh method to assess the effectiveness of an r-refinement adaptive mesh method at capturing the fast initial dynamics of the equation. For the purposes of this dissertation, I will be taking constant values of $\tau = 2$ and $\epsilon^2 = 0.05$, where $\tau = \max f''(\phi)$ and ϵ represents the interfacial thickness.

One can easily prove a conservation property of the Cahn-Hilliard equation when zero total flux boundary conditions are employed, by integrating equation 1.9 over the domain Ω,

$$\int_\Omega \frac{\partial \varphi_1}{\partial t} d\Omega = \int_\Omega \nabla^2 \left(\varphi_2 + f'(\varphi_1) - \tau\varphi_1\right) d\Omega. \tag{1.11}$$

Taking the time derivative outside of the integral on the left hand side, and applying the divergence theorem to the right hand side,

$$\frac{d}{dt}\int_\Omega \varphi_1 d\Omega = \oint \nabla \left(\varphi_2 + f'(\varphi_1) - \tau\varphi_1\right) . \underline{\hat{n}} dS \tag{1.12}$$

$$= \oint \frac{\partial}{\partial n} \left(\varphi_2 + f'(\varphi_1) - \tau\varphi_1\right) dS. \tag{1.13}$$

Introducing zero flux boundary conditions as a restriction on the outward flux at the boundaries,

$$\oint \frac{\partial \phi}{\partial n} ds = 0, \tag{1.14}$$

ensuring that the left hand side of equation 1.12 equates to zero. We then arrive at the following statement displaying conservation of the phase field integral,

$$\int_{\Omega} \varphi_1 d\Omega = constant\ in\ time. \tag{1.15}$$

In the following Chapters I introduce the methods that I will be using to solve the Cahn-Hilliard equation numerically. Firstly, I will broadly discuss moving mesh methods, highlighting the velocity-based method, upon which this feasibility study is based, which I intend to use in conjunction with a finite element approach to model the Cahn-Hilliard equation. The middle Chapters (3, 4, 5 & 6) then guide the reader through the moving mesh finite element formulation to generate a linear system, which when solved will provide the solution, along with supplementary material discussing potential problems and methods used to overcome these issues. The final Chapters (7 & 8) then analyse the results of the model and take a critical view of the velocity-based moving mesh method as a valid approach to tackling the tricky dynamics of the Cahn-Hilliard equation.

Chapter 2

Adaptive Mesh Method

One of the main issues associated with modelling the Cahn-Hilliard equation is in being able to follow the fast dynamics generated from the equation as the phases form. This behaviour requires an adaptive method to ensure points are close enough to maintain high resolution for the steeper gradients, and more sparse for shallow gradients. This is particularly apparent in the case of the Cahn-Hilliard equation, since the distinct regions display extreme opposites in gradients at both boundaries and areas inside the newly-formed regions. For this reason adaptive mesh refinement is the highest priority in the modelling of the equation.

2.1 Static Mesh Methods

The adaptive mesh refinement methods used previously, when taking an Eulerian approach, mainly fall into two categories which both involve using a static mesh as a basis. The two methods are called h-refinement and p-refinement.

h-refinement

h-refinement is the process where nodes are added to points of interest, in this case where the dynamics of the solution occur, and may involve removing nodes of little interest where no dynamics in the solution are apparent. This requires an effective flagging strategy based on either gradients between or at nodes, or differences in nodal values. There are three main forms of h-refinement which have been used throughout research. The most intricate of the three is adaption using both coarsening and refinement of a grid, flagging both cells where little and significant change occur. The alternative methods, which are much more widely used, begin with either a highly refined initial grid which is then coarsened, or a coarse grid which is refined. In the case of *Ceniceros, et al.* (10) a coarsening, h-refinement method was applied, using a static grid and employing ghost cells and interpolation.

p-refinement

p-refinement uses a static grid with a higher order polynomial to accurately represent the solution.

This method is more accurate between the nodal values in comparison to h-refinement. However, interpolation in the cells is limited by the degree of the polynomial chosen and hence cannot accurately model dynamics occuring in between nodes.

2.2 Velocity Based Methods

An alternative Lagrangian-based approach is to use a specified number of nodes, moving them to the areas of interest where the fast dynamics of the solution occur. This method is commonly referred to as **r-refinement** and is divided into *Mapping*-based and *Velocity*-based techniques. These methods have been chronicled in a number of papers, most notably by *Huang* (14) and *Budd* (6). Both velocity and mapping based methods are dependent on a suitable monitor function. The most common types of r-refinement are mapping-based methods. These mapping-based r-refinement techniques use a time-dependent mapping from the original static grid to a moving grid. A widely used technique to solve Moving Mesh Partial Differential Equations or MMPDEs, which is an equidistributive version of the mapping-based method based on equidistribution (6). In this case, a monitor function is chosen which, when integrated over each element, is constant in space. This then implies that x satisfies

$$0 = \frac{\partial}{\partial \xi}\left(m\frac{\partial x}{\partial \xi}\right), \tag{2.1}$$

where ξ is the reference coordinate in a fixed grid, $x(\xi, t)$ is the mapping at a time t and m is the chosen monitor function. The velocity of the mapped coordinate is then found via the relaxed form of equation 2.1,

$$\frac{\partial x}{\partial t} = \frac{1}{\tau}\frac{\partial}{\partial \xi}\left(m\frac{\partial x}{\partial \xi}\right). \tag{2.2}$$

or some variant of this equation.

Velocity-based techniques create a velocity to move the original grid. There are many velocity-based approaches one can use, falling into three main categories: moving finite elements (20)(19), deformation map method (18) and geometric conservation law (9). A form of the geometric conservation law velocity-based technique, previously introduced to handle a two-phase Stefan problem by *Baines, et al.* (3), is the type of method I intend to use to try to solve the Cahn-Hilliard equation.

An advantage of r-adaptivity over the other types of adaptive mesh refinement, is that the dynamics of the solution can still be tracked well and the method can be less computationally expensive. Problems with this method are a potential lack of resolution in the solution in regions away from singularities as well as a susceptibility to node tangling. It is therefore important to resolve the initial singularities quickly. A further possible problem with this method is that further dynamics leading to singularities in regions previously considered relatively featureless, away from the initial singularities, can be poorly tracked due to a lack of points/large elements in the region.

In review, most Cahn-Hilliard equation-based research undertaken so far applies a form of h-refinement (10)(1)(11)(12) for their adaptive mesh method. Currently, there are few papers taking a moving adaptive mesh method to model the solution of the Cahn-Hilliard equation. One such

paper (13) took an interface tracking, Lagrangian approach to model the process. However, in this dissertation, I intend to introduce an alternative r-refinement based adaptive mesh method, previously used to solve high order PDEs (2)(4)(6), to help model the dynamics of the Cahn-Hilliard equation. In theory, the velocity of the solution dictates the position of the nodes in the subsequent timestep. This should allow for the solution to require the minimum number of nodes for all time, instead of decreasing or increasing the number of nodes. Therefore, this could be much less computationally expensive in comparison to previous h-refinement-based methods.

Chapter 3

Application to the Cahn-Hilliard Equation

We now proceed with our feasibility study using mesh movement.

3.1 A Moving Mesh Finite Element Formulation

The most important aspect of our approach to a moving mesh finite element method for a problem is to define which property of the solution we wish to conserve with respect to time. The conserved property is usually expressed in terms of a monitor function. There are two main monitor functions which we will consider to model our dynamic time-dependent solution.

3.1.1 Monitor Functions

Conservation of Mass

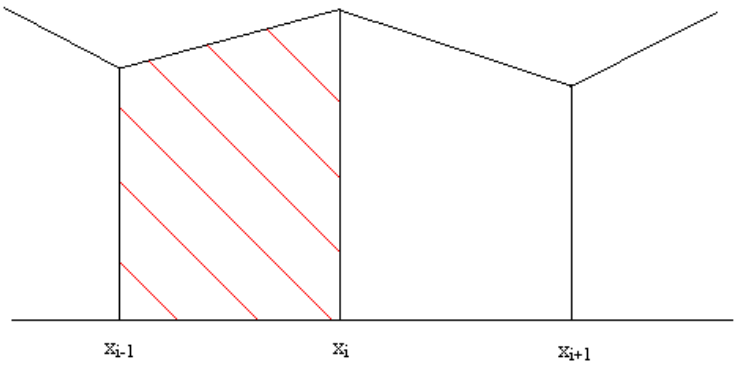


Figure 3.1: Piecewise area of solution is conserved between nodes.

Since, for the Cahn-Hilliard system, we know that the integral, $\int_\Omega \varphi_1 d\Omega$, is conserved, it is consistent to use a monitor function which conserves the partitional area under the solution, often called a *Mass Conservation* monitor function. In the case of the Cahn-Hilliard equation, we may impose conservation of the phasefield $\phi = \varphi_1$ over the local interval, i.e.

$$\int_{i-1}^{i} \varphi_1 d\Omega = \delta_i = constant, \ \forall i. \tag{3.1}$$

This can be written as,

$$\int_\Omega W_i \varphi_1 d\Omega = \int_{i-1}^{i} \varphi_1 d\Omega = constant, \tag{3.2}$$

where W is the characteristic function in the interval and the W_i's form a block-based partition of unity.

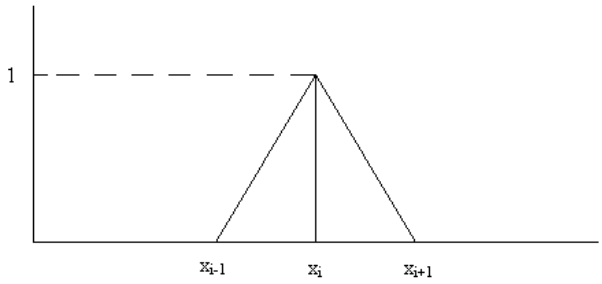


Figure 3.2: Representation of a piecewise linear hat function, w_i

Introducing a more general test function w_i, which is a piecewise linear hat function, also a member of a partition of unity, we can replace 3.2 by

$$\int_\Omega w_i \varphi_1 d\Omega = \delta_i, \ \forall i, \tag{3.3}$$

to generate node movement.

As one can see in 1-D, if linear hat functions are used as the test functions and φ_1 is expanded using the linear hat functions to give, $\varphi_1 = \sum_j W_j \Phi_j$, we arrive at

$$\int_{i-1}^{i} W_i \sum_j W_j \Phi_j dx = \delta_i, \ \forall i, \tag{3.4}$$

which is a mass matrix formulation of the connection between the Φ_j and the δ_i represented in matrix form as,

$$M\underline{\Phi} = \underline{\delta}, \tag{3.5}$$

where M is the standard mass matrix given by $\int W_i W_j dx$ and $\underline{\Phi}$, $\underline{\delta}$ the vectors of components Φ_j and δ_i. This shows the finite element formulation of the phase field φ_1, now represented by $\underline{\Phi}$, as the matrix system 3.5, where M represents the standard mass matrix, the elemental form of which (between nodes $i-1$ and i) is,

$$M_i^e = (x_i - x_{i-1}) \begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{pmatrix}. \tag{3.6}$$

Remark

Care should be taken when introducing either Dirichlet or periodic boundary conditions on φ_1 (or Φ) (15). In order to ensure the hat functions remain a partition of unity, the second test function can be augmented by the first and the penultimate test function by the last. For instance, if there exists a Dirichlet condition on the first element boundary, then element two must now have an adapted linear hat function $W_2 = W_1 + W_2$ taking into account that element one is now known.

Conservation of Arclength

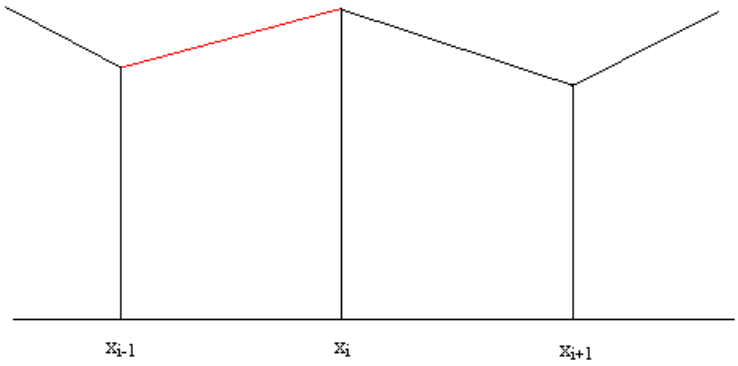


Figure 3.3: Piecewise arclength of solution is conserved between nodes.

An alternative monitor to φ_1 is a normalised arclength monitor function. In this case, rather than conserving the area under the solution between nodes, we conserve the normalised arclength of the solution between nodes. In some instances this may be a more preferential monitor function since extreme singularities may not form properly when using a conservation of mass approach. Using an arclength monitor should also help to maintain a good balance between areas with many nodes and those with few nodes. Considering the arclength defined to be

$$\int_{i-1}^{i} \sqrt{1 + \left(\frac{\partial \varphi_1}{\partial x}\right)^2}\, dx, \tag{3.7}$$

then conservation of normalised arclength requires that

$$\frac{1}{\theta} \int_{i-1}^{i} \sqrt{1 + \left(\frac{\partial \varphi_1}{\partial x}\right)^2}\, dx = a_i = constant,\ \forall i, \tag{3.8}$$

θ is defined as $\theta = \int w\varphi_1 dx$. Analogously to 3.3 we use,

$$\frac{1}{\theta}\int_0^1 w_i\sqrt{1+\left(\frac{\partial\varphi_1}{\partial x}\right)^2}dx = a_i = constant,\ \forall i, \tag{3.9}$$

to generate node movement.

Chapter 4

1-D Cahn-Hilliard Equation using a Conservation of Mass Monitor

Taking the Cahn-Hilliard system of equations derived by *Ceniceros, et al.* (10) we have the following 1-D system,

$$\frac{\partial \varphi_1}{\partial t} = \frac{\partial^2 \varphi_2}{\partial x^2} + g(\varphi_1) \tag{4.1}$$

$$\varphi_2 = \tau\varphi_1 - \epsilon^2 \frac{\partial^2 \varphi_1}{\partial x^2} \tag{4.2}$$

where

$$g(\varphi_1) = \frac{\partial^2}{\partial x^2}(f'(\varphi_1) - \tau\varphi_1). \tag{4.3}$$

We apply periodic boundary conditions, as used in (10) over the interval from 0 to 1. Applying Liebnitz Integral Rule to the left hand side of equation 4.1, we can generate an expression for a velocity v, on which I can base my r-refinement strategy,

$$\frac{d}{dt}\int_0^1 w_i\varphi_1 dx = \int_0^1 w_i \frac{\partial \varphi_1}{\partial t} dx + \int_0^1 w_i \frac{\partial(\varphi_1 v)}{\partial x} dx \tag{4.4}$$

$$= \int_0^1 w_i \left[\frac{\partial^2 \varphi_2}{\partial x^2} + g(\varphi_1)\right] dx + \int_0^1 w_i \frac{\partial(\varphi_1 v)}{\partial x} dx \tag{4.5}$$

where v is the velocity, which we now write as $\dot{x}$.

4.1 Calculating the $\frac{\partial \varphi_1}{\partial t}$ Term

Applying the distributed conservation of mass principle (equation 3.3) ensures that the left hand side of equation 4.4 is zero, giving the following weak form for equation 4.1,

$$\frac{d}{dt}\int_0^1 w\varphi_1 dx = 0 = \int_0^1 \frac{\partial}{\partial t}(w\varphi_1) dx + \int_0^1 \frac{\partial}{\partial x}(w\varphi_1 \dot{x}) dx \tag{4.6}$$

$$= \int_0^1 \left[w\frac{\partial \varphi_1}{\partial t} + \varphi_1 \frac{\partial w}{\partial t} + w\frac{\partial}{\partial x}(\varphi_1 \dot{x}) + \varphi_1 \dot{x}\frac{\partial w}{\partial x}\right] dx. \tag{4.7}$$

Since the test function is advected with the velocity $\dot{x}$, the test function w satisfies

$$\frac{\partial w}{\partial t} + \dot{x}\frac{\partial w}{\partial x} = 0. \tag{4.8}$$

This allows us to reduce equation 4.7 to

$$\begin{aligned} 0 &= \int_0^1 \left[w\frac{\partial \varphi_1}{\partial t} + w\frac{\partial}{\partial x}(\varphi_1 v) \right] dx && (4.9)\\ &= \int_0^1 \left[w\left(\frac{\partial^2 \varphi_2}{\partial x^2} + g(\varphi_1) \right) + w\frac{\partial}{\partial x}(\varphi_1 v) \right] dx && (4.10)\\ &= w\frac{\partial \varphi_2}{\partial x}\Big|_0^1 - \int_0^1 \frac{\partial w}{\partial x}\frac{\partial \varphi_2}{\partial x}dx + \int_0^1 wg(\varphi_1)dx + \int_0^1 w\frac{\partial}{\partial x}(\varphi_1 v)dx. && (4.11) \end{aligned}$$

Taking $w = W_i$ to be a member of the set of linear hat functions, and $v = \sum_i W_i v_i$ to represent the velocity, we obtain

$$0 = W_i\frac{\partial \varphi_2}{\partial x}\Big|_0^1 - \int_0^1 \frac{\partial W_i}{\partial x}\frac{\partial \varphi_2}{\partial x}dx + \int_0^1 W_i g(\varphi_1)dx + \int_0^1 \sum_j W_i\frac{\partial}{\partial x}(\varphi_1 W_j v_j)dx \ \forall i, \tag{4.12}$$

We now consider the terms of equation 4.12 separately. The first term will be referred to as the Boundary Term, the second is referred to as the φ_2 Term, the third is called the g Term, and finally the Velocity Term present at the far right.

Velocity Term

First we consider the velocity term and integrate by parts, giving

$$\int_0^1 \sum_j W_i\frac{\partial}{\partial x}(\varphi_1 W_j v_j)dx = \left[\varphi_1 \sum_j W_i W_j v_j \right]_0^1 - \int_0^1 \varphi_1 \sum_j \frac{\partial W_i}{\partial x} W_j v_j dx \ \forall i \tag{4.13}$$

On evaluating the left hand term on the right hand side of equation 4.13 one can see that, through the properties of the hat functions, only the boundary terms survive. However, by keeping the boundary points stationary, this implies that the velocities of the boundary terms are both equal to zero. Therefore the boundary term is equal to zero, leaving the much simpler version,

$$\int_0^1 \sum_j W_i\frac{\partial}{\partial x}(\varphi_1 W_j v_j)dx = \int_0^1 \varphi_1 \sum_j \frac{\partial W_i}{\partial x} W_j v_j dx \ \forall i. \tag{4.14}$$

In matrix form, this can be written

$$B_{(\varphi_1)}\underline{v}, \tag{4.15}$$

where $B_{(\varphi_1)}$ represents the matrix with entries $\int_0^1 \varphi_1 \frac{\partial W_i}{\partial x} W_j dx$, with the following general form for the i^{th} row,

$$\begin{aligned} B_{(\varphi_1)}i &= \int_{i-1}^{i+1} \varphi_1 W_i \frac{\partial W_j}{\partial x}dx \ \forall i && (4.16)\\ &= \int_{i-1}^{i} \varphi_1 W_i \frac{\partial W_j}{\partial x}dx + \int_{i}^{i+1} \varphi_1 W_i \frac{\partial W_j}{\partial x}dx && (4.17)\\ &= \frac{1}{x_i - x_{i-1}}\int_{i-1}^{i} \varphi_1 W_i dx + \frac{-1}{x_{i+1} - x_i}\int_{i}^{i+1} \varphi_1 W_i dx. && (4.18) \end{aligned}$$

Taking the left integral from $i-1$ to i on its own and evaluating the integral numerically using Simpson's rule,

$$\begin{aligned} \frac{1}{x_i - x_{i-1}}\int_{i-1}^{i} \Phi W_i dx &= \frac{1}{x_i - x_{i-1}}(x_i - x_{i-1})\left(\frac{W_i\Phi_i + W_{i-1}\Phi_{i-1}}{2} \right) && (4.19)\\ &= \frac{1}{2}(\Phi_i + \Phi_{i-1}). && (4.20) \end{aligned}$$

Applying this to the remaining half of the integral, we finally obtain an expression for the i^{th} row of the B matrix weighted by φ_1 to be $\frac{1}{2}(\Phi_{i-1} - \Phi_{i+1})$, which in matrix form is represented as,

$$B_{(\varphi_1)} = \begin{pmatrix} 0 & -\frac{\Phi_1}{2} & \dots & \dots \\ \frac{\Phi_0}{2} & 0 & -\frac{\Phi_2}{2} & \dots \\ \vdots & \ddots & \ddots & -\frac{\Phi_{N+1}}{2} \\ \vdots & & \frac{\Phi_N}{2} & 0 \end{pmatrix} \tag{4.21}$$

This unsymmetric matrix, however, can be problematic to invert, and so we decided to take an alternative approach by introducing a velocity potential, Ψ, where $v = \frac{\partial \Psi}{\partial x}$. Introducing this into the equation 4.14, and expanding $\Psi = \sum_j \Psi_j W_j$, we get the following representation for the Velocity Term,

$$\int_0^1 \varphi_1 \frac{\partial W_i}{\partial x} \sum_j W_j v_j dx = \int_0^1 \varphi_1 \frac{\partial W_i}{\partial x} \sum_j \frac{\partial W_j}{\partial x} \Psi_j dx \ \forall i, \tag{4.22}$$

in matrix form,

$$K_{(\varphi_1)} \underline{\Psi}, \tag{4.23}$$

where $K_{(\varphi_1)}$ is the symmetric stiffness matrix weighted by φ_1. We obtain v from Ψ by minimising the error between the velocity and the gradient of Ψ via

$$\int_0^1 W_i \left(v - \frac{\partial \Psi}{\partial x} \right) dx = 0, \ \forall i. \tag{4.24}$$

By expanding the velocity and velocity potential using a series of linear hat functions

$$\int_0^1 W_i v dx = \int_0^1 W_i \frac{\partial \Psi}{\partial x} dx \tag{4.25}$$

$$\int_0^1 W_i \sum_j W_j v_j dx = \int_0^1 W_i \sum_j \frac{\partial W_j}{\partial x} \Psi_j dx \tag{4.26}$$

$$\left[\int_0^1 W_i \sum_j W_j dx \right] \underline{v} = \left[\int_0^1 W_i \sum_j \frac{\partial W_j}{\partial x} dx \right] \underline{\Psi}, \tag{4.27}$$

or, in matrix form,

$$M \underline{v} = B \underline{\Psi}. \tag{4.28}$$

where the matrix B is the unweighted version of $B_{(\varphi_1)}$ in equation 4.21.

φ_2 Term

Coming back to equation 4.12, we now evaluate the φ_2 term, expanding φ_2 as $\sum_i W_i \varphi_j$, giving

$$\int_\Omega \frac{\partial W_i}{\partial x} \sum_j \frac{\partial W_j}{\partial x} \varphi_j dx = \left[\int_\Omega \frac{\partial W_i}{\partial x} \sum_j \frac{\partial W_j}{\partial x} dx \right] \underline{\varphi} \ \forall i, \tag{4.29}$$

which in matrix form can be expressed as

$$K \underline{\varphi}, \tag{4.30}$$

where K is the standard stiffness matrix, the elemental form of which (between nodes $i-1$ and i) is given as

$$K_i^e = \frac{1}{(x_i - x_{i-1})} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}. \tag{4.31}$$

Remark

Again, great care should be taken when introducing the periodic boundary conditions into the system. This requires a reduced system with the test function adjacent to either boundary having an additional weight (15).

g Term

The $g(\varphi_1)$ term in equation 4.12 refers to the $g(\varphi_1)$ term in equation 4.33. Here, we start by looking at the function $f(\varphi_1)$, given by the bulk energy density function, $\frac{1}{4}(1-\varphi_1^2)^2$. For $g(\varphi_1)$ we require the first derivative of f with respect to φ_1, giving $f'(\varphi_1) = \varphi_1^3 - \varphi_1$. Therefore, by considering the i^{th} row of the integral equation in the weak form and integrating by parts,

$$\begin{aligned} \int_{i-1}^{i+1} W_i g(\varphi_1) dx &= \int_{i-1}^{i+1} W_i(\varphi_1^3 - (\tau+1)\varphi_1)_{xx} dx & (4.32) \\ &= W_i(\varphi_1^3 - (\tau+1)\varphi_1)_x\big|_{i-1}^{i+1} + \int_{i-1}^{i+1} \frac{\partial W_i}{\partial x}(\varphi_1^3 - (\tau+1)\varphi_1)_x dx. & (4.33) \end{aligned}$$

Expanding φ_1 as $\varphi_1 = \sum_j W_j \Phi_j$, one can easily see that the left hand term on the right hand side of equation 4.33 vanishes, since W_i vanishes at these values. The remaining term can be approximated by the trapezium rule to give

$$\int_{i-1}^{i+1} \frac{\partial W_i}{\partial x}(\varphi_1^3 - (\tau+1)\varphi_1)_x dx = \frac{-1}{x_i - x_{i-1}} \left(\varphi_1^3 - (\tau+1)\varphi_1\right)\Big|_{i-1}^{i} + \frac{1}{x_{i+1} - x_i} \left(\varphi_1^3 - (\tau+1)\varphi_1\right)\Big|_{i}^{i+1} \quad \forall i \tag{4.34}$$

which leads to the following representation in matrix form,

$$-K\underline{g}. \tag{4.35}$$

where $\underline{g} = (\underline{\Phi_1^3} - (\tau+1)\underline{\Phi_1})$. Higher order quadrature can also be used in equation 4.34.

Boundary Term

The boundary term relies on conditions imposed on the equation. We decided to have stationary endpoints and to consider both φ_1 and φ_2 to be periodic at the boundaries of the domain, effectively those suggested in (10). With this being the case, periodicity is weakly imposed by ensuring the first order spatial derivatives are also periodic making the boundary term in equation 4.12 vanish.

This finally gives the finite element form of the system proposed by *Ceniceros, et al.* (equation 4.1) to be

$$\begin{aligned} K_{(\varphi_1)}\underline{\Psi} &= -K\underline{\varphi} - K\underline{g} & (4.36) \\ M\underline{v} &= B\underline{\Psi}. & (4.37) \end{aligned}$$

Once the velocities v have been found from equation 4.37, the new positions of the nodes are calculated by integrating the mesh forward in time with a simple Forward Euler method,

$$\frac{\underline{x}^{n+1} - \underline{x}^{n}}{\Delta t} = \underline{v}. \tag{4.38}$$

With the new nodal positions known, the new φ_1 solution is obtained by solving the mass conservation equation 3.5, with the elements of the mass matrix recalculated using the new nodal positions.

4.2 Calculating φ_2

So far, the generation of φ_2 has not been discussed. This requires a weak form of equation 4.2. Applying a test function to equation 4.2 and integrating, one gets

$$\int_\Omega w\varphi_2 dx = \tau \int_\Omega w\varphi_1 dx - \epsilon^2 \int_\Omega w \frac{\partial^2 \varphi_1}{\partial x^2} dx \tag{4.39}$$

$$= \tau \int_\Omega w\varphi_1 dx - \epsilon^2 \left[w \frac{\partial \varphi_1}{\partial x}\bigg|_\Omega - \int_\Omega \frac{\partial w}{\partial x}\frac{\partial \varphi_1}{\partial x} dx \right] \tag{4.40}$$

after integration by parts.

Then by introducing the hat functions as the test functions and expanding $\varphi_1 = \sum_i W_i \Phi_i$ and $\varphi_2 = \sum_i W_i \varphi_i$, one then gets the following version of equation 4.40,

$$\int_\Omega W_i \sum_j W_j dx \underline{\varphi} = \tau \int_\Omega W_i \sum_j W_j dx \underline{\Phi} - \epsilon^2 \left[W_i \sum_j \frac{\partial W_j}{\partial x} \underline{\Phi} \right]_\Omega + \epsilon^2 \int_\Omega \frac{\partial W_i}{\partial x} \sum_j \frac{\partial W_j}{\partial x} dx \underline{\Phi} \ \forall i. \tag{4.41}$$

One can easily see that the boundary term vanishes from weakly imposing periodic boundary conditions which we imposed when reviewing the previous **Boundary Term** of the φ_1 equation. This then allows the previous equation to be expressed in matrix form,

$$M\underline{\varphi} = \tau M \underline{\Phi} + \epsilon^2 K \underline{\Phi}. \tag{4.42}$$

4.3 Adaptive Timestepping

A characteristic of the Cahn-Hilliard equations discussed in the paper by *Ceniceros, et al.* (10) is the requirement of a small initial timestep to determine the fast forming phase boundaries, and a larger timestep once the phasefields are formed. We also found in the runs of the program that 'tangling' of the coordinates initially was often likely. By introducing an adaptive timestepping method described below, the equation itself chooses the suitable relevant timestep. Ideally we could then employ conditions to prevent any potential tangling of the coordinates, and ensure the coordinates were monotonically increasing. We implemented two adaptive timestepping strategies.

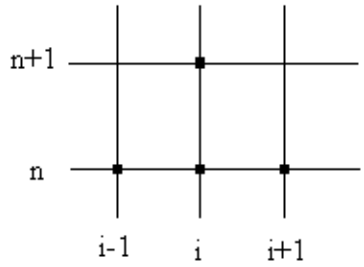


Figure 4.1: 2-D stencil representing nodes at n and $n+1$ time levels, displaying an explicit timestepping approach.

4.3.1 Explicit Adaptive Timestep Method

We introduce an explicit adaptive timestep method based on an explicit Euler approach to the relationship between velocity and the subsequent velocity potential Ψ, in the form

$$v = \dot{x} = \Psi_x. \tag{4.43}$$

Fig.4.1 gives a graphical representation of the stencil for the explicit method.

By applying the index n to represent the timestep, we then get an explicit form of this equation,

$$\frac{x_i^{n+1} - x_i^n}{\Delta t} = \left[\frac{\Psi_{i+1}^n - \Psi_{i-1}^n}{x_{i+1}^n - x_{i-1}^n}\right] \tag{4.44}$$

$$= \left[\frac{\Psi_{i+1}^n - \Psi_{i-1}^n}{\Delta x_i^n}\right], \tag{4.45}$$

where $x_{i+1}^n - x_{i-1}^n$ is rewritten as Δx_i^n. We can write the system as

$$\frac{x_i^{n+1} - x_i^n}{\Delta t} = \left[\frac{\Psi_{i+1}^n}{\Delta x_i^n}\frac{(x_{i+1}^n - x_i^n)}{(x_{i+1}^n - x_i^n)}\right] - \left[\frac{\Psi_{i-1}^n}{\Delta x_i^n}\frac{(x_i^n - x_{i-1}^n)}{(x_i^n - x_{i-1}^n)}\right]. \tag{4.46}$$

which can be rearranged to give

$$x_i^{n+1} = \frac{\Psi_{i+1}^n \Delta t}{\Delta x_i^n \Delta^+ x_i^n} x_{i+1}^n + \left[1 - \frac{\Psi_{i+1}^n \Delta t}{\Delta x_i^n \Delta^+ x_i^n} - \frac{\Psi_{i-1}^n \Delta t}{\Delta x_i^n \Delta^- x_i^n}\right] x_i^n + \frac{\Psi_{i-1}^n \Delta t}{\Delta x_i^n \Delta^- x_i^n} x_{i-1}^n \tag{4.47}$$

$$= a x_{i+1}^n + b x_i^n + c x_{i-1}^n, \tag{4.48}$$

say.

Here Δx_i^n is partitioned into two other differences, $\Delta^+ x_i^n$ or $\Delta^- x_i^n$, representing $(x_{i+1}^n - x_i^n)$ and $(x_i^n - x_{i-1}^n)$ respectively. Since the equation 4.48 shows the representation of the i^{th} row, this can be represented in matrix form as

$$\underline{x}^{n+1} = T\underline{x}^n, \tag{4.49}$$

where T is the matrix represented by the coefficients of x^n on the right hand side of 4.48. A sufficient condition for the stability of x_i is that, $a > 0$, $b > 0$, $c > 0$, and that $a + b + c = 1$. Upon inspection of the second term b, this requires the following restriction on the timestep Δt,

$$\Delta t < \Delta x^n(i) \left[\frac{\Delta^+ x^n(i)}{\Psi_{i+1}^n} + \frac{\Delta^- x^n(i)}{\Psi_{i-1}^n}\right]. \tag{4.50}$$

This restriction on the timestep ensures that no x_i^{n+1}-values move beyond x_{i-1}^n or x_{i+1}^n. Therefore, in order for the system to have monotonically increasing x-values, it is sufficient to reduce the Δt to $\frac{\Delta t}{2}$. By using this as a non-tangling strategy in tandem with a suitably small initial timestep we can get a good idea of the initial dynamics.

4.3.2 Implicit Adaptive Timestep Method

With the explicit adaptive timestepping method, Δt often tends to be greatly restrictive. In order to eradicate this, we considered an implicit adaptive timestepping method to model the properties of solutions generated by the Cahn-Hilliard equation. Fig. 4.2 gives a graphical representation of the stencil for the implicit method similar to that used by *Ceniceros, et al.* in (10),

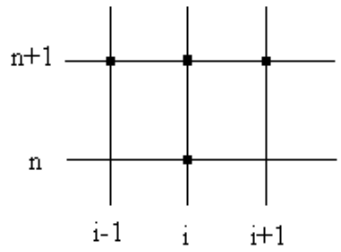


Figure 4.2: 2-D stencil representing nodes at n and $n+1$ time levels, displaying an implicit timestepping approach.

The implicit method has similar methodology to the explicit method. However, rather than multiply the terms in equation 4.45 by unit fractions (expressed as differences in the x-values of the current timestep over the current timestep), as in equation 4.46, we multiply by fractions expressed as a ratio of differences in the x-values of the new timestep over those of the current timestep, (which does not affect the first-order-in-time property of the algorithm).

This gives the following row formulation for Ψ_{i+1} and Ψ_{i-1},

$$\frac{x_i^{n+1} - x_i^n}{\Delta t} = \left[\frac{\Psi_{i+1}^n}{\Delta x_i^n}\frac{(x_{i+1}^{n+1} - x_i^{n+1})}{(x_{i+1}^n - x_i^n)}\right] - \left[\frac{\Psi_{i-1}^n}{\Delta x_i^n}\frac{(x_i^{n+1} - x_{i-1}^{n+1})}{(x_i^n - x_{i-1}^n)}\right]. \tag{4.51}$$

Without loss of generality we can assume that both Ψ_{i+1} and Ψ_{i-1} are positive, since an arbitrary constant can be added to them to ensure that this is so without affecting the velocity. This implicit method allows us to ensure that the new x_i^{n+1}-values would lie within x_{i+1}^{n+1} and x_{i-1}^{n+1} values, which therefore ensures a monatonically increasing set of x-values. To show this, suppose that (x_i, t) is the first point at which a maximum value occurs in the domain, then the left hand side of equation 4.51 is positive. Along with this, the left hand term on the right hand side is negative since $x_{i+1}^{n+1} < x_i^{n+1}$, and the right hand term on the right hand side is positive since $x_i^n > x_{i-1}^n$, resulting in a negative right hand side. This gives a contradiction and implies that the maximum must lie on the boundary. A similar argument implies that the minimum must also lie on the boundary. Therefore, since our domain is $[0, 1] \times [0, t]$, the $(i + 1)^{th}$ term is always the maximum term in the stencil, giving a

monotonically increasing set of x-values for any Δt.

Rearranging 4.51 gives the following expression for the i^{th} row with $\Psi > 0$,

$$\begin{aligned} x_i^n &= \frac{\Psi_{i+1}^n \Delta t}{\Delta x^n(i)\Delta^+ x^n(i)} x_{i+1}^{n+1} + \left[1 - \frac{\Psi_{i+1}^n \Delta t}{\Delta x^n(i)\Delta^+ x^n(i)} - \frac{\Psi_{i-1}^n \Delta t}{\Delta x^n(i)\Delta^- x^n(i)}\right] x_i^{n+1} \\ &\quad + \frac{\Psi_{i-1}^n \Delta t}{\Delta x^n(i)\Delta^- x^n(i)} x_{i-1}^{n+1} \qquad (4.52) \\ &= a x_{i+1}^{n+1} + b x_i^{n+1} + c x_{i-1}^{n+1}, \qquad (4.53) \end{aligned}$$

say. This system can be expressed in matrix form as,

$$T \underline{x}^{n+1} = \underline{x}^n, \qquad (4.54)$$

where T is a tridiagonal matrix with right, centre and left entries represented by a, b and c as in equation 4.53.

The adaptive implicit timestepping method ensures that the coordinates remain in a monotonically increasing set for any Δt. This is an advantageous method, since it allows us to choose any timestep and still maintain a monotonically increasing set of x-coordinates. Despite this advantage, this method is best applied only when larger timesteps are appropriate, since it is less and less accurate for large timesteps.

4.4 Merging Points and Steep Fronts

Through further analysis of the initial equation for $\frac{\partial \varphi_1}{\partial t}$, it is possible to extract a hyperbolic term, present when isolating the cubic term in g in the second derivative, writing it as,

$$\begin{aligned} \frac{\partial^2(\varphi_1^3)}{\partial x^2} &= \frac{\partial\left(3\varphi_1^2 \frac{\partial \varphi_1}{\partial x}\right)}{\partial x} \qquad (4.55) \\ &= 3\varphi_1^2 \frac{\partial^2 \varphi_1}{\partial x^2} + 6\varphi_1 \left(\frac{\partial \varphi_1}{\partial x}\right)^2, \qquad (4.56) \end{aligned}$$

where the second term on the right hand side is the first order hyperbolic term. Such equations are known to lead to the presence of steep fronts in a time-dependent solution. One can see that this term on its own may generate discontinuities, since this is a viscous form of a Hamilton-Jacobi equation, which can be manipulated into a form of the viscous Burgers' equation present in conservation laws. However, one can deduce that only steep fronts rather than discontinuities are apparent in this time-dependent solution, due to the presence of the first diffusion term on the right hand side as well as the remaining part of the g term. These diffusive terms enforce smooth solutions, diffusing the discontinuities, but retaining the steep gradients.

When applying the explicit adaptive timestep method to the mass monitor-based model of the Cahn-Hilliard equation, it was clear that the velocities generated were simply too large to ensure that the x-coordinates remained in sequence. Evidently, this was undesirable, therefore in order to

maintain the coordinate sequence, we decided to further restrict the timestep to an interval which culminated in the first of two consecutive points in the sequence coinciding. At this stage in time we merged the two points, giving the resultant point new φ_1, φ_2 and Ψ values, taken to be the average of the values from the colliding points. These new values ensured that we reverted back to the piecewise continuous solutions for φ_1, φ_2 and Ψ that we had had prior to the discontinuity present just before merging.

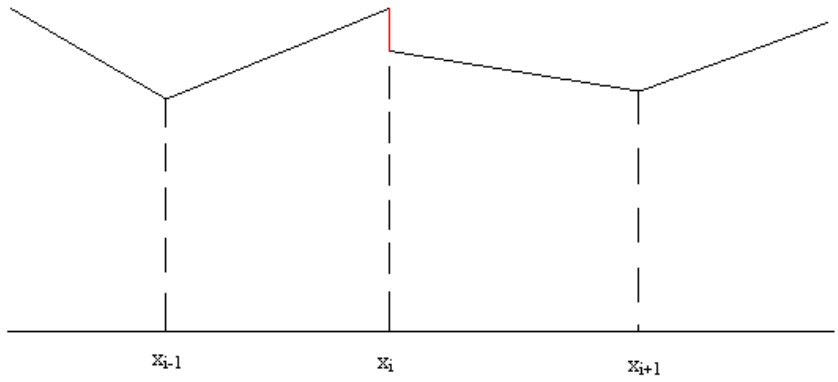


Figure 4.3: Graph representing a discontinuity present in the piecewise solution.

The process of merging itself is of course undesirable, since the process requires the removal of nodes from generated singularities, which contradicts the thoughts behind the velocity-based adaptive mesh method, reducing the resolution.

Remark

A further point can be made on the presence of the hyperbolic term within the coupled Cahn-Hilliard system. For specific cases, where one models the thickness of the interfacial layers with $\epsilon \to 0$ along with a certain bulk energy density $f(\phi)$, a Hamilton-Jacobi system can be a direct outcome of the Cahn-Hilliard system we are using. In more general cases, once phase separation occurs, one might suppress the diffusion terms and proceed with a discontinuous approach.

4.4.1 Additional Resolution

A potential method to counteract the reduction in resolution that merging causes, is to increase the resolution of the initial data by introducing an initial refinement of the grid. The idea being to resolve parts of the initial data where steep gradients occur before starting the time discretization. This is of particular concern when attempting to use a random function between 0 and 1 to determine the initial data. By introducing a certain tolerance level, if consecutive values in the initial data

were beyond this value, then additional nodes were added, evenly spaced between the initial nodes in question. These additional values were then given values of initial data via linear interpolation, ensuring the original piecewise data remained the same.

A further solution to this problem could be from introducing smoothing into the model, to ensure the elements are not too close together. However, this is again an undesirable process as it attempts to remove or reduce the presence of steep fronts which are a characteristic of the solution. Unfortunately, the presence of additional nodes does not remove the issue of node-tangling present in the 1-D mass monitor model. So we seek an alternative approach, deciding to introduce an arclength monitor to gain a better distribution of the nodes and restrict overlapping.

Chapter 5

1-D Cahn-Hilliard Equation using an Arclength Monitor

In review of the dynamics displayed by the solution of the Cahn-Hilliard equation, it was envisaged that an alternative monitor function may be more effective to model the solution more successfully and, potentially, more efficiently. The alternative monitor function is an *Arclength Conservation* based approach. In this case, rather than conserve the area under the solution between nodes, we conserve the normalised arclength of the solution between nodes. In order to do this we introduce a normalized conservation principle,

$$\frac{1}{\theta}\int_0^1 W_i\sqrt{1+\left(\frac{\partial\varphi_1}{\partial x}\right)^2}\,dx = a_i = constant,\ \forall i, \tag{5.1}$$

where the normalising factor θ is defined by the total arclength over the whole region,

$$\int_0^1 \sqrt{1+\left(\frac{\partial\varphi_1}{\partial x}\right)^2}\,dx = \theta = constant. \tag{5.2}$$

This is a consistent conservation principle if $\sum_i a_i = 1$, since the W_i form a partition of unity.

We can now differentiate the equation 5.1 with respect to time and apply the Liebnitz Integral Rule, as with the mass conserving approach to find the nodal velocities. Assuming the points in the interval $[0,1]$, along with the W_i, move with an unknown velocity v, we get the following series of equations,

$$\begin{aligned}
a_i\dot{\theta} &= \frac{d}{dt}\int_0^1 W_i\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}\,dx,\ \forall i, && (5.3)\\
&= \int_0^1 W_i\frac{\partial}{\partial t}\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}\,dx + \int_0^1 W_i\frac{\partial}{\partial x}\left(\dot{x}\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}\right)dx && (5.4)\\
&= \int_0^1 W_i\frac{\frac{\partial\varphi_1^n}{\partial x}\frac{\partial^2\varphi_1^n}{\partial x\partial t}}{\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}}dx + \int_0^1 W_i\frac{\partial}{\partial x}\left(v\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}\right)dx, && (5.5)
\end{aligned}$$

where both $\dot{\theta}$ and v are unknown, and $\dot{\theta} = \frac{d\theta}{dt}$. One can then find the value for $\dot{\theta}$ by summation over all i,

$$\dot{\theta} = \int_0^1 \frac{\frac{\partial\varphi_1^n}{\partial x}\frac{\partial^2\varphi_1^n}{\partial x \partial t}}{\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}} dx + \left(v\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2} \right)\Bigg|_0^1 . \tag{5.6}$$

Since the velocities are zero at the boundaries, the boundary term involving the velocity becomes zero, giving

$$\dot{\theta} = \int_0^1 \frac{\frac{\partial\varphi_1^n}{\partial x}\frac{\partial^2\varphi_1^n}{\partial x \partial t}}{\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}} dx. \tag{5.7}$$

We can now substitute in for $\frac{\partial\varphi_1}{\partial t}$ using the previously defined equation 4.1, i.e.

$$\frac{\partial\varphi_1}{\partial t} = \frac{\partial^2\varphi_2}{\partial x^2} + \frac{\partial^2}{\partial x^2}(\varphi_1^3 - (\tau+1)\varphi_1), \tag{5.8}$$

to give

$$\dot{\theta} = \int_0^1 \frac{\frac{\partial\varphi_1^n}{\partial x}\frac{\partial}{\partial x}\left(\frac{\partial^2\varphi_2^n}{\partial x^2} + \frac{\partial^2}{\partial x^2}[(\varphi_1^n)^3 - (\tau+1)\varphi_1^n]\right)}{\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}} dx. \tag{5.9}$$

Upon integration element by element

$$\dot{\theta} = \sum_{elements_k} \left[\frac{\frac{\partial\varphi_1^n}{\partial x}\Big|_{k-\frac{1}{2}} \left(\frac{\partial^2\varphi_2^n}{\partial x^2} + \frac{\partial^2}{\partial x^2}[(\varphi_1^n)^3 - (\tau+1)\varphi_1^n]\right)\Big|_{k-1}^{k}}{\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}\Big|_{k-\frac{1}{2}}} \right]. \tag{5.10}$$

Coming back to equation 5.5, we now have only one remaining unknown, v. In order to include the newly determined $\dot{\theta}$ into the system of equations in 5.5 and ensure that the equations remain linearly independent, we impose the zero boundary conditions on v. Having set the velocities at either end of the domain to be zero, only a reduced set of equations is used, based on internal points. By using a test function belonging to a set of linear hat functions, and integrating the terms in equation 5.5, we get

$$a_i\dot{\theta} = \int_0^1 W_i \frac{\frac{\partial\varphi_1^n}{\partial x}\frac{\partial^2\varphi_1^n}{\partial x\partial t}}{\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}} dx + \int_0^1 W_i \frac{\partial}{\partial x}\left(v\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2} \right) dx, \ \forall i, \tag{5.11}$$

$$\begin{aligned} = \ & W_i \left[\frac{\frac{\partial\varphi_1^n}{\partial x}\frac{\partial\varphi_1^n}{\partial t}}{\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}} \right]_0^1 - \int_0^1 \frac{\partial W_i}{\partial x} \frac{\frac{\partial\varphi_1^n}{\partial x}\frac{\partial\varphi_1^n}{\partial t}}{\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}} dx \\ & + W_i \left[v\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2} \right]_0^1 - \int_0^1 \frac{\partial W_i}{\partial x}\left(v\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2} \right) dx. \end{aligned} \tag{5.12}$$

Since we impose periodic boundary conditions along with $v = 0$ at the boundaries, we can remove the boundary terms to reduce equation 5.12 to,

$$a_i\dot{\theta} = -\int_0^1 \frac{\partial W_i}{\partial x} \frac{\frac{\partial\varphi_1^n}{\partial x}\frac{\partial\varphi_1^n}{\partial t}}{\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2}} dx - \int_0^1 \frac{\partial W_i}{\partial x}\left(v\sqrt{1+\left(\frac{\partial\varphi_1^n}{\partial x}\right)^2} \right) dx. \tag{5.13}$$

We again substitute in for $\frac{\partial \varphi_1}{\partial t}$ using the previously defined equation, 4.1, to give

$$a_i\dot{\theta} = -\int_0^1 \frac{\partial W_i}{\partial x}\frac{\frac{\partial \varphi_1^n}{\partial x}\left(\frac{\partial^2 \varphi_2^n}{\partial x^2} + \frac{\partial^2}{\partial x^2}[(\varphi_1^n)^3 - (\tau+1)\varphi_1^n]\right)}{\sqrt{1+\left(\frac{\partial \varphi_1^n}{\partial x}\right)^2}}dx - \int_0^1 \frac{\partial W_i}{\partial x}\left(v\sqrt{1+\left(\frac{\partial \varphi_1^n}{\partial x}\right)^2}\right)dx. \tag{5.14}$$

The right hand side can be evaluated element by element since the approximate solution is piecewise linear, hence $\frac{\partial \varphi_1}{\partial x}$ is constant,

$$\begin{aligned} a_i\dot{\theta} =& \frac{\frac{\partial \varphi_1^n}{\partial x}}{\sqrt{1+\frac{\partial \varphi_1^n}{\partial x}}}\int_{i-1}^{i}\frac{\partial W_i}{\partial x}\left(\frac{\partial \varphi_2^n}{\partial x} + \frac{\partial}{\partial x}[(\varphi_1^n)^3 - (\tau+1)\varphi_1^n]\right)dx \\ &+ \frac{\frac{\partial \varphi_1^n}{\partial x}}{\sqrt{1+\frac{\partial \varphi_1^n}{\partial x}}}\int_{i}^{i+1}\frac{\partial W_i}{\partial x}\left(\frac{\partial \varphi_2^n}{\partial x} + \frac{\partial}{\partial x}[(\varphi_1^n)^3 - (\tau+1)\varphi_1^n]\right)dx \\ &+ \sqrt{1+\frac{\partial \varphi_1^n}{\partial x}}\int_{i-1}^{i}\frac{\partial W_i}{\partial x}v dx + \sqrt{1+\frac{\partial \varphi_1^n}{\partial x}}\int_{i}^{i+1}\frac{\partial W_i}{\partial x}v dx, \ \forall i. \end{aligned} \tag{5.15}$$

This results in the following matrix representation for the internal points,

$$a\dot{\theta} = -K_1\underline{\varphi} - K_1(\underline{\Phi}^3 - (\tau+1)\underline{\Phi}) - B_1\underline{v}, \tag{5.16}$$

where K_1 is the stiffness matrix weighted by the first differential of the arclength, $\left(\frac{\frac{\partial \varphi_1^n}{\partial x}}{\sqrt{1+\frac{\partial \varphi_1^n}{\partial x}}}\right)$, and B_1 is the B matrix weighted by the arclength, $\left(\sqrt{1+\frac{\partial \varphi_1^n}{\partial x}}\right)$. Once the nodal velocities are found, we then use them to project the x-coordinates using the standard Forward Euler method,

$$x_i^{n+1} = x_i^n + v\Delta t. \tag{5.17}$$

5.1 Evolution of φ_1 values

In using an arclength monitor, it is no longer feasible to use the conservation principle directly. Instead, the evolution of the φ_1 values with time are generated by an Arbitrary Legrange-Eulerian method (ALE Method). Using the area of the equation, we can introduce a new evolving variable σ generated in the following way,

$$\sigma_i = \int_0^1 W_i\varphi_1 dx, \ \forall i, \tag{5.18}$$

Differentiating this with respect to time, we then arrive at the same system of equations generated by the mass monitor in order to calculate the evolution of the new variable σ,

$$\sigma_i = \int_0^1 W_i\frac{\partial \varphi_1}{\partial t}dx + \int_0^1 W_i\frac{\partial(\varphi_1 v)}{\partial x}dx, \ \forall i. \tag{5.19}$$

Substituting in for $\frac{\partial \varphi_1}{\partial t}$ from equation 5.8 above, and integrating term by term, as previously shown in the mass monitor method by equation 4.36, we arrive at the following matrix system,

$$\dot{\underline{\sigma}} = -K\underline{\varphi} - K\underline{g} - B_{(\varphi_1)}\underline{v}, \tag{5.20}$$

where K is the standard stiffness matrix, $\underline{g} = (\underline{\Phi}^3 - (\tau+1)\underline{\Phi})$ and $B_{(\varphi_1)}$ is the weighted B matrix, both previously referenced when formulating the Cahn-Hilliard system using a mass monitor by equations 4.31 and 4.21 respectively. Having found the change in σ values with respect to time, expressed as $\underline{\dot{\sigma}}$, we can then find the new $\underline{\sigma}$ values using the standard Eulerian formula,

$$\sigma_i^{n+1} = \sigma_i^n + \dot{\sigma_i}^n \Delta t, \ \forall i. \tag{5.21}$$

Finally, in order to calulate the new φ_1 values, we can use the following relation for the internal values,

$$M\underline{\Phi} = \underline{\sigma}, \tag{5.22}$$

where M is the standard mass matrix and $\underline{\Phi}$ and $\underline{\sigma}$ are the vectors representing φ_1 and σ values respectively, as in equation 3.5.

5.2 Calculating φ_2 values

The formulation for φ_2 remains much more simple. This requires expanding the weak form of equation 4.2. Applying a test function to equation 4.2 and integrating,

$$\begin{aligned} \int_\Omega w\varphi_2 dx &= \tau \int_\Omega w\varphi_1 dx - \epsilon^2 \int_\Omega w \frac{\partial^2 \varphi_1}{\partial x^2} dx & (5.23) \\ &= \tau \int_\Omega w\varphi_1 dx - \epsilon^2 \left[w \frac{\partial \varphi_1}{\partial x}\bigg|_\Omega - \int_\Omega \frac{\partial w}{\partial x}\frac{\partial \varphi_1}{\partial x} dx \right] & (5.24) \end{aligned}$$

Then by introducing the hat function as the test function and expanding $\varphi_1 = \sum_i W_i \Phi_i$ and $\varphi_2 = \sum_i W_i \varphi_i$, equation 5.24 becomes

$$\int_\Omega W_i \sum_j W_j dx \underline{\varphi} = \tau \int_\Omega W_i \sum_j W_j dx \underline{\Phi} - \epsilon^2 \left[W_i \sum_j \frac{\partial W_j}{\partial x}\underline{\Phi} \right]_\Omega + \epsilon^2 \int_\Omega \frac{\partial W_i}{\partial x} \sum_j \frac{\partial W_j}{\partial x}\underline{\Phi} \ \forall i. \tag{5.25}$$

One can easily see that the left hand ϵ^2 term vanishes with periodic boundary conditions. This then allows equation 5.25 to be expressed in the matrix form,

$$M\underline{\varphi} = \tau M\underline{\Phi} + \epsilon^2 K\underline{\Phi}. \tag{5.26}$$

In addition to the arclength monitor method, several attempts were made to ensure node-tangling was less likely to occur. Along similar steps to the mass monitor model in 1-D, an explicit adaptive timestepping approach was used, as well as including additional nodes to further resolve the initial data. A second approach to ensure nodes were distributed evenly in terms of arclength, to be well suited to the monitor, was used. This was however, difficult to program with the presence of random initial data between 0 and 1.

Chapter 6

The 2-D Cahn-Hilliard Equation

6.1 Grid Structure

Since we are using finite elements to solve the equation, in order to produce a simple formulation of the problem in 2-D, a triangular grid is used with piecewise linear approximation (16). The most effective method to produce a triangulation of the region is a Delaunay triangulation, due to the definitions of the criteria used to create the triangles. Using this method, and considering a set of nodes, no nodes lie inside the circumcircle of each triangle generated, and there are no mesh points in the interior of any element circumcircles. When creating the triangular elements this property ensures that the smallest angle inside each triangle is maximised. This is extremely important since small angles are known to cause ill-conditioning in stiffness matrices, as I will verify later on when generating the stiffness and mass matrices for a generic triangle. With this in mind, triangular elements were produced using a Delaunay triangulation of the region $[0,1]\times[0,1]$ with $(N+1)\times(N+1)$ nodes in an initially uniform grid, and repeated once new nodal coordinates were generated in each timestep.

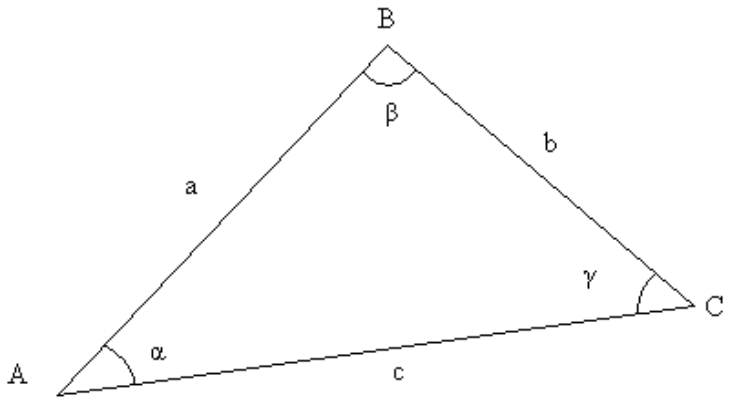


Figure 6.1: A 2-D triangular element.

By splitting the grid into a series of small elemental triangles, the formations of the stiffness matrix and the mass matrix are comparatively simple in 2-D. In order to understand the structure of an elemental stiffness matrix, consider an arbitrary triangle in space, represented by the nodes

A, B & C (Fig.6.1), with corresponding angles, α, β & γ, and lengths, $AB = a$, $BC = b$ & $CA = c$. The angles can be found using the cosine rule, i.e.

$$cos\alpha = \frac{a^2 + c^2 - b^2}{2ac}. \tag{6.1}$$

Each diagonal element on the elemental stiffness matrix is calculated in a standard way by evaluating

$$\int_\Delta (\nabla w_A).(\nabla w_A) d\Omega = -\int_\Delta (\nabla w_A).(\nabla w_B + \nabla w_C) d\Omega \tag{6.2}$$

$$= \frac{1}{2}(cot\beta + cot\gamma). \tag{6.3}$$

Similarly, non-diagonal elements are calculated as

$$\int_\Delta (\nabla w_B).(\nabla w_C) d\Omega = -\frac{1}{2} cot\alpha. \tag{6.4}$$

This therefore gives the following expression for an element stiffness matrix for the triangle ABC,

$$K^e_{(ABC)} = \frac{1}{2} \begin{pmatrix} cot\beta + cot\gamma & -cot\gamma & -cot\beta \\ -cot\gamma & cot\gamma + cot\alpha & -cot\alpha \\ -cot\beta & -cot\alpha & cot\beta + cot\alpha \end{pmatrix}. \tag{6.5}$$

The Mass Matrix has the simple form,

$$M^e_{(ABC)} = area_\Delta \begin{pmatrix} \frac{1}{6} & \frac{1}{12} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \end{pmatrix}. \tag{6.6}$$

where $area_\Delta$ is given by the formula,

$$area_\Delta = \frac{a^2}{\frac{1}{2}(cot\alpha + cot\beta)}. \tag{6.7}$$

6.2 Finite Element Formulation

6.2.1 Generating φ_2 values

In order to generate the φ_2 values from the known initial data of the φ_1 values, we need to assess the equation for φ_2 given by the Cahn-Hilliard system generated by *Ceniceros, et al.* in 2-D,

$$\varphi_2 = \tau\varphi_1 - \epsilon^2 \nabla^2 \varphi_1. \tag{6.8}$$

By integrating this over the region, $\Omega = [0,1] \times [0,1]$,

$$\int_\Omega w_i \varphi_2 d\Omega = \tau \int_\Omega w_i \varphi_1 d\Omega - \epsilon^2 \int_\Omega w_i \nabla^2 \varphi_1 d\Omega. \tag{6.9}$$

Integrating the far right hand term by parts,

$$\int_\Omega w_i \varphi_2 d\Omega = \tau \int_\Omega w_i \varphi_1 d\Omega - \epsilon^2 \oint_{d\Omega} w_i \underline{\nabla}\varphi_1 . \underline{\hat{n}} d\Gamma + \epsilon^2 \int_\Omega \underline{\nabla} w_i . \underline{\nabla} \varphi_1 d\Omega. \tag{6.10}$$

Imposing periodic boundary conditions on φ_1 ensures that the boundary integral equates to zero. The remaining integrals are of basic mass matrix and stiffness matrix construction, where periodicity boundary conditions are imposed on both φ_1 and φ_2, resulting in the matrix system

$$M\underline{\varphi} = \tau M \underline{\Phi} + \epsilon^2 K \underline{\Phi}, \tag{6.11}$$

where $\underline{\Phi}$ and $\underline{\varphi}$ are the vectors representing the values of φ_1 and φ_2 respectively, and M & K are assembled versions of the mass and stiffness matrices defined by equation 6.6 and equation 6.5.

6.2.2 Generating Nodal Velocities

In the 2-D case, only the conservation of mass monitor was implemented, due to time constraints. This version was similar in construction to the 1-D case. In 2-D, the Cahn-Hilliard system developed by *Ceniceros, et al.* is now written,

$$\frac{\partial}{\partial t}\varphi_1 = \nabla^2\varphi_2 + g(\varphi_1) \tag{6.12}$$

$$\varphi_2 = \tau\varphi_1 - \epsilon^2\nabla^2\varphi_1, \tag{6.13}$$

where $g(\varphi_1) = \nabla^2(\varphi_1^3 - (\tau+1)\varphi_1)$. In this 2-D case, using a conservation of mass monitor, we assume that

$$\int_\Omega w_i\varphi_1 d\Omega = \delta_i, \ \forall i, \tag{6.14}$$

where δ_i is a constant and w_i a hat function, both located at node i. From equation 6.14,

$$\frac{d}{dt}\int_\Omega w_i\varphi_1 d\Omega = 0, \ \forall i. \tag{6.15}$$

Since this is now a 2-D system, the Reynold's Transport Theorem is applied in lieu of Liebnitz Rule to the above equation, in the form

$$\frac{d}{dt}\int_\Omega w_i\varphi_1 d\Omega = 0 \tag{6.16}$$

$$= \int_\Omega \frac{d}{dt} w_i\varphi_1 d\Omega + \oint_{\partial\Omega} w_i(\varphi_1\underline{v}).\underline{n}d\Gamma, \ \forall i,$$

where $\underline{v}$ is the vector representing the nodal velocities. The boundary integral can be evaluated as an area integral using Green's Theorem, making equation 6.17 composed entirely of area integrals,

$$\int_\Omega w_i\left[\frac{\partial}{\partial t}\varphi_1 + \underline{\nabla}.(\varphi_1\underline{v})\right]d\Omega = 0, \ \forall i. \tag{6.18}$$

Substituting in for $\frac{\partial}{\partial t}\varphi_1$ from equation 6.12

$$\int_\Omega w_i\left[\nabla^2\varphi_2 + g\right]d\Omega + \int_\Omega w_i\underline{\nabla}.(\varphi_1\underline{v})d\Omega = 0, \ \forall i. \tag{6.19}$$

Applying Green's Theorem to the velocity integral, the equation can then be rearranged as

$$\int_\Omega w_i\left[\nabla^2\varphi_2 + g\right]d\Omega - \int_\Omega \varphi_i\underline{\nabla}w_i.\underline{v}d\Omega = 0, \ \forall i. \tag{6.20}$$

Re-introducing the notion of velocity potential (vital for uniquesness in 2-D), represented as,

$$\underline{v} = \underline{\nabla}\Psi, \tag{6.21}$$

we can now rewrite equation 6.20, as

$$\int_\Omega \varphi_i\underline{\nabla}w_i.\underline{\nabla}\Psi d\Omega = \int_\Omega w_i\left[\nabla^2\varphi_2 + g\right]d\Omega, \ \forall i, \tag{6.22}$$

which can be expressed in matrix form as,

$$K_{(\varphi_1)}\underline{\Psi} = \underline{f}, \tag{6.23}$$

where $K_{(\varphi_1)}$ is a weighted stiffness matrix relating the nodes as an assembly of elemental matrices representing the relations between the three nodal points of each triangle. The elemental matrix, $K^e_{(\varphi_1)}$, is displayed in the following way by denoting φ_1^A as the value of φ_1 at the node A,

$$K^e_{(ABC)(\varphi_1)} = \frac{(\varphi_1^A + \varphi_1^B + \varphi_1^C)}{3} \begin{pmatrix} cot\beta + cot\gamma & -cot\gamma & -cot\beta \\ -cot\gamma & cot\gamma + cot\alpha & -cot\alpha \\ -cot\beta & -cot\alpha & cot\beta + cot\alpha \end{pmatrix}. \tag{6.24}$$

$\underline{f}$ Vector

Having assessed the structure of the left hand side containing the velocity potential term, we now turn our attention to the right hand side f vector, containing the terms associated with $\frac{\partial \varphi_1}{\partial t}$,

$$f_i = \int_\Omega w_i \left[\nabla^2 \varphi_2 + g(\varphi_1)\right] d\Omega, \ \forall i, \tag{6.25}$$

where $g(\varphi_1) = \nabla^2(\varphi_1^3 - (\tau+1)\varphi_1)$. Through integration by parts in two dimensions, we get the following expression for components of $\underline{f}$,

$$f_i = \oint_{\partial\Omega} w_i \underline{\nabla}\varphi_2 . \underline{\hat{n}} d\Gamma - \int_\Omega \underline{\nabla} w_i . \underline{\nabla}\varphi_2 d\Omega + \int_\Omega w_i g(\varphi_1) d\Omega, \ \forall i. \tag{6.26}$$

Using periodic boundary conditions on both φ_1 and φ_2, the boundary integral equates to zero. We can then assess the g term separately, as

$$\int_\Omega w_i g(\varphi_1) d\Omega = \int_\Omega w_i \nabla^2(\varphi_1^3 - (\tau+1)\varphi_1) d\Omega \tag{6.27}$$

$$= \oint_{\partial\Omega} w_i \underline{\nabla}(\varphi_1^3 - (\tau+1)\varphi_1) . \underline{\hat{n}} d\Gamma - \int_\Omega \underline{\nabla} w_i . \underline{\nabla}(\varphi_1^3 - (\tau+1)\varphi_1) d\Omega, \ \forall i. \tag{6.28}$$

Imposing periodic boundary conditions, the boundary integral equates to zero. This gives the following components of the $\underline{f}$ vector,

$$f_i = -\int_\Omega \underline{\nabla} w_i . \underline{\nabla}\varphi_2 d\Omega - \int_\Omega \underline{\nabla} w_i . \underline{\nabla}(\varphi_1^3 - (\tau+1)\varphi_1) d\Omega, \ \forall i. \tag{6.29}$$

Re-introducing the velocity potential term, this system can be represented in matrix terms as

$$K_{(\varphi_1)}\underline{\Psi} = \underline{f} = -K\underline{\varphi} - K(\underline{\Phi}^3 - (\tau+1)\underline{\Phi}). \tag{6.30}$$

6.2.3 Evolution of Nodes

Once the velocity potential for each node has been found, we can calculate the velocity by using the two dimensional gradient operator on the equation,

$$\underline{v} = \underline{\nabla}\Psi. \tag{6.31}$$

In order to approximate the velocity, we use the standard finite element matrix relationship between the velocity and its potential

$$M\underline{v} = B\underline{\Psi} \tag{6.32}$$

where the B matrix in 2-D has the following formulation

$$B_{ij} = \int_{\Omega} W_i.\underline{\nabla}W_j d\Omega = \sum_{elements\ around\ i} \int_{\Delta} W_i.\underline{\nabla}W_j d\Omega = \sum_{elements\ around\ i} \underline{\nabla}W_j \int_{\Delta} W_i d\Omega. \tag{6.33}$$

This method is only used on internal points, since one of the boundary conditions imposed on the problem is that boundary nodes have zero velocity, and remain static nodes. However, this method was not implemented due to time constraints, and a much coarser 2-D central difference approach was used. The positions of the new internal nodes are then found using the explicit Forward Euler discretization,

$$\frac{\underline{x}^{n+1} - \underline{x}^{n}}{\Delta t} = \underline{v}. \tag{6.34}$$

In the 1-D case, timestepping was created and restricted via an adaptive timestepping method, ensuring a stable timestep was self-generated by the program. This approach generated a varying timestep dependent on the nodal velocities and positions and the previous timestep. In the 2-D case, the timestepping was determined via trial and error, and remained constant throughout the course of the program. This was not further adapted due to time constraints as well as being difficult to implement in 2-D.

Having found the new nodal positions, we generate a new Delaunay triangulation using the new coordinates of the nodes. This then allows us to generate the new φ_1 values from equation 6.14 in the form

$$M\underline{\Phi} = \underline{\delta}, \tag{6.35}$$

where the mass matrix, M, is recalculated with the new nodes and areas of the triangles which have arisen from the new Delaunay triangulation.

Chapter 7

Numerical Results

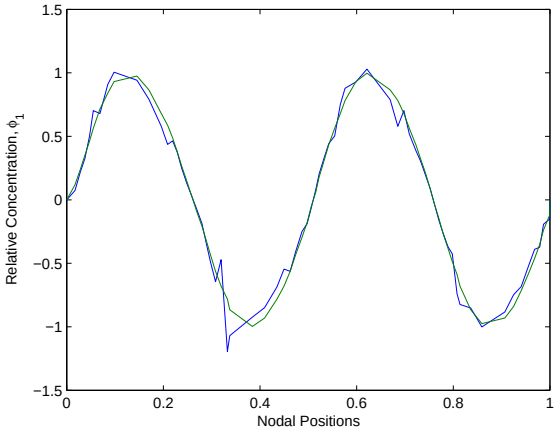
All of the 1-D and 2-D models were computed using MATLAB. Throughout the course of the dissertation, the programs were particularly sensitive and difficult to debug. Unfortunately, all of the programs failed to handle the extreme dynamics of phase boundary formation. Therefore, no full dynamic solution to the Cahn-Hilliard equation was obtained via the programs. However, the programs did present solutions prior to phase boundary formation. Despite the difficulties encountered, I discovered interesting properties of the moving mesh method used to solve the Cahn-Hilliard equation. Initial runs of the programs used a random function, but were then replaced with a smooth periodic function more inclined to the boundary conditions to test the accuracy of the method for the problem. To investigate the solutions developed by the 1-D programs, we used smooth periodic initial data in the form

$$\varphi_1 = sin(4\pi x). \tag{7.1}$$

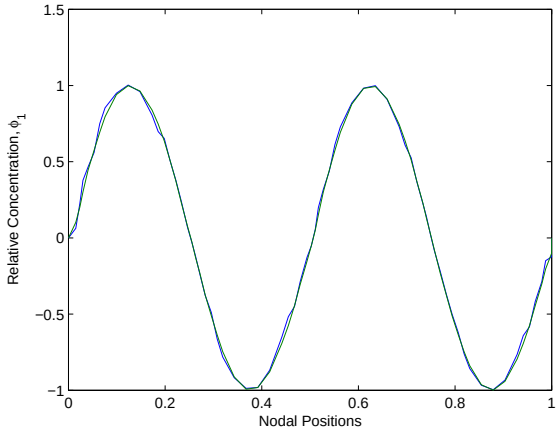
In order to assess the accuracy of the solution found by the numerical methods, the initial data was also substituted into the Cahn-Hilliard equation and computed analytically to find a solution exact in space and approximated in time. Furthermore, the nodal positions were evolved using a similar Forward Euler numerical method along with the same Δt used in the corresponding numerical method.

7.1 1-D Mass Monitor

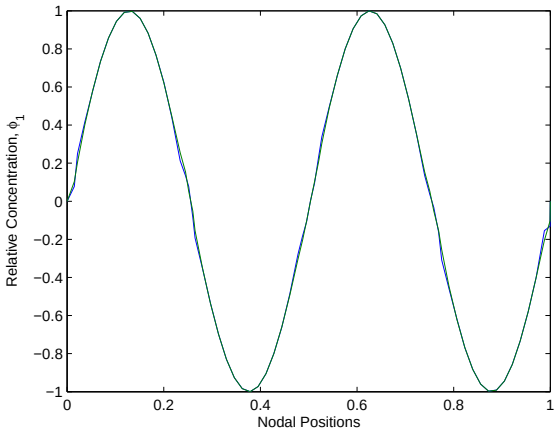
In the 1-D mass monitor approach, node tangling was particularly prominent. The presence of negligible elements, culminating from the velocities generated by the equation, developed near to singular matrices ensuring the problem quickly became ill-posed. For this reason, the adaptive timestepping method, initial addition of nodes and merging were implemented as a preventative technique. Unfortunately, this resulted in a limited number of timesteps that the program could run before merging was no longer an issue, and the program encountered similar element-based problems and the solution collapsed. Due to this, in order to show convergence of the method, only 5 timesteps were taken. Due to issues regarding matrix calculations, only even numbers of nodes were used to set up the initial grid.



(a)
n=22,
m=70,
L_2-
norm=0.7288



(b)
n=42,
m=70,
L_2-
norm=0.2095



(c)
n=62,
m=71,
L_2-
norm=0.1375

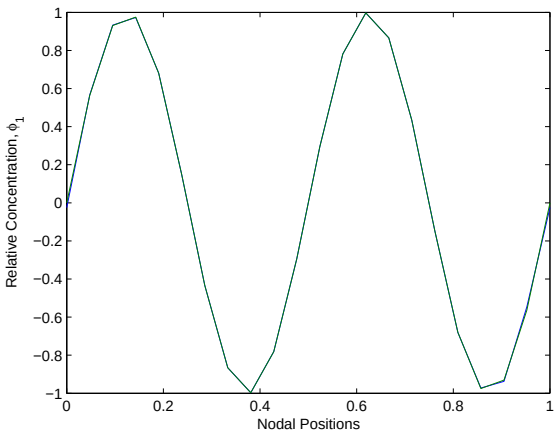
Figure 7.1: Results for the mass monitor program with n initial nodes, m total nodes after 5 timesteps and a variable timestep along with the true solution (in green)

Since this method contained the use of an adaptive timestepping method, where Δt alters for each timestep, each run was simulated to a different point in time. The factors affecting Δt was dependent on both the initial data and the initial number of nodes used to formulate the piecewise initial data. Due to the process of adding further nodes to resolve the steep gradients, the total number of nodes differed from the initial starting amount. Since we are only displaying the results using a piecewise smooth representation of the initial data, the number of initial nodes was the single factor affecting Δt. By increasing the number of initial nodes from 21 to 41, Δt decreased by a factor of 10. Therefore, error analysis of this method remains unclear due to the disparity in time evolved by each solution. Further analysis of the finite element method used is also difficult to calculate due to the moving mesh.

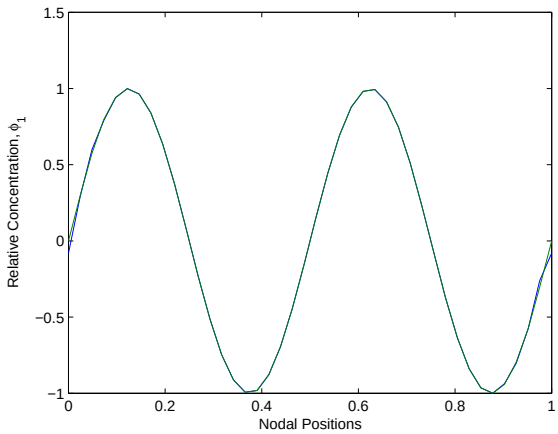
7.2 1-D Arclength Monitor

The program designed to model the 1-D arclength monitor involved a similar set-up to the mass monitor program. An adaptive timestepping method used along with the initial data was of the same nature, as well as the initial number of nodes used to represent it.

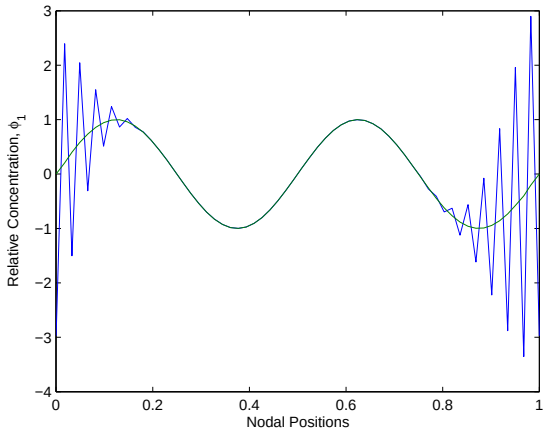
From running the program, several problematic areas akin to those encountered in the mass monitor program arose. The most significant of these was that although the process of merging nodes/elements was not required when using an arclength monitor, the solution still collapsed after a finite number of timesteps. However, a distinguishing feature of the arclength monitor program was the generation of spurious oscillations at the boundaries in the case where $n = 62$, which were possibly due to a bug in the program or from the boundary conditions imposed. These oscillations suggest that the timestepping method method may be unstable, and a more A-stable scheme may be more appropriate.



(a)
n=22,
L_2-
norm=0.0398



(b)
n=42,
L_2-
norm=0.1232



(c)
n=62,
L_2-
norm=8.08

Figure 7.2: Results for the arclength monitor program with n initial nodes after 5 timesteps and a variable timestep along with the true solution

7.3 2-D Case

In the 2-D case, the smooth periodic initial data used was of the form,

$$\varphi_1 = sin(4\pi x)sin(4\pi y). \tag{7.2}$$

Unfortunately, due to time constraints an analytic solution along with Forward Euler timestepping of the initial data, with which one could compare the solution from the numerical method, was not computed. The initial grid used was a unit square of equally spaced nodes with the Delaunay triangulation in Fig.7.3,

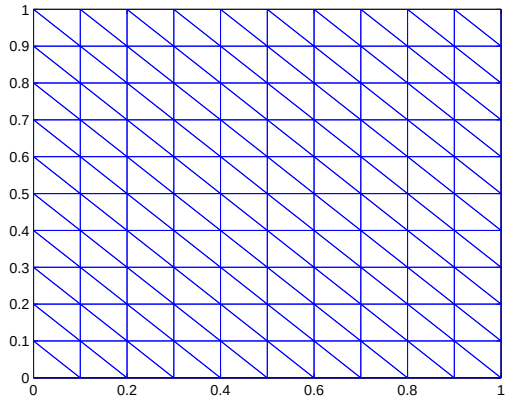


Figure 7.3: Delaunay triangulation of the unit square grid

From observations of results from the mass monitor program in 1-D, one may expect the 2-D mass monitor program to have similar problems of node-tangling as well as the generation of singular matrices. However, these issues were not so prevalent and the method appeared to be more stable than the 1-D programs, allowing for a larger timestep to be used. In this case a timestep of several orders in magnitude larger than those generated by the adaptive timestepping method in 1-D were used. This gave the result shown in Fig.7.4.

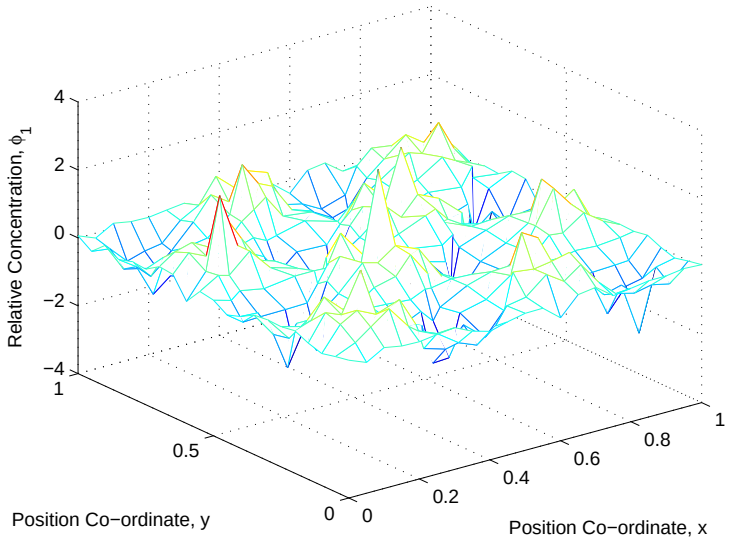


Figure 7.4: Solution of the Cahn-Hilliard equation in 2-D with $n = 20$ using 10 timesteps and a fixed $\Delta t = 5 \times 10^{-6}$

Clearly, problems are still present in the method, displayed through the anomally in the solution in the middle of the region (where $\varphi_1 \approx 3$). However, no node tangling occurs despite the use of a large timestep. The profile did also show some signs of generating steep fronts, which is more visible in the Delaunay triangulation of the region shown in Fig.7.5.

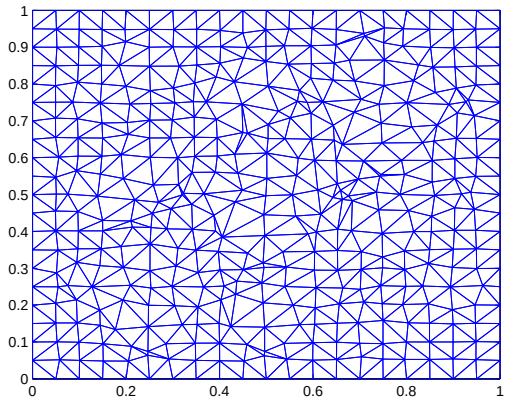


Figure 7.5: Delaunay triangulation of the unit square grid after 20 timesteps

Unfortunately, error analysis of the method in 2-D has not been evaluated qualitatively, owing to time constraints, and as such the accuracy of this solutions can not be guaranteed. Also, further analysis of the finite element method used is difficult to evaluate due to the moving mesh method with which it is applied.

Chapter 8

Conclusions and Further Work

8.1 Summary

In this chapter I will be evaluating and discussing the findings and methods used, suggest possible improvements and further avenues to pursue. In this feasibility study, a velocity-based moving mesh method based on a monitor function was applied in tandem with a finite element method to model the fast dynamics of the fourth order Cahn-Hilliard equation.

In Chapter 1 the system was introduced and its properties were discussed. We also discussed the applications of the Cahn-Hilliard equation and introduced some previous approaches taken to model it. In Chapter 2, we reviewed the various approaches available for grid adaptation, focussing on the velocity-based approach on which the feasibility study was centred as well as discuss the properties and advantages over the more widely used Eulerian (static-grid) approach. In Chapter 3 we discussed the most suitable monitor functions upon which to base the finite element moving mesh method upon. Initially, we started by discussing a mass conservation approach to model the Cahn-Hilliard equation, proven to be conservative over the region, and discussed the possible advantages of an alternative arclength monitor. Chapter 4 detailed the equations generated by the velocity-based moving mesh method, and solved them in detail using a finite element method subject to the conservation of mass monitor function. The systems were derived, and the boundary conditions then stated and implemented. In Chapter 5 we constructed the system of equations generated by the velocity-based moving mesh method, which were solved using a finite element method with an arclength monitor.

Chapter 6 detailed the equations developed by the moving mesh method in 2-D, which were again solved in conjunction with a conservation of mass monitor function. The numerical method was then used to compute solutions of the systems generated throughout Chapters 4, 5 and 6 using MATLAB. These were then reviewed in Chapter 7 where problematic areas were highlighted in greater detail. Of particular concern was the short length of time the programs simulated up to before exhibiting unphysical phenomena and catastrophic node tangling.

8.2 Remarks and Further Work

8.2.1 Timestepping

Throughout the dissertation, it was apparent that significantly small timesteps were required in order to display a solution to the Cahn-Hilliard equation and avoid node tangling. Unfortunately, node tangling and mesh racing (6) became a very important issue to plague all of the models. This was overcome briefly in the 1-D models by the introduction of an explicit adaptive timestep method to generate a maximum timestep size, used to ensure no overtaking occurred in the grid. Applying this method, one could soon find the severe limitations of the current set-up, with timesteps of the order 10^{-14} being presented as maximum timesteps in the 1-D conservation of mass approach before node tangling occurred. Further adaptation would be advisable, applying an explicit adaptive timestepping method until the steep fronts have been formed, subsequently applying an implicit adaptive timestepping strategy to view the steady state solutions. Despite curtailing the issue, it was clear the problem lay in the limitations of the timestepping method itself, rather than the adaptivity employed.

For the purposes of the dissertation a standard first order Forward Euler method was used. This method has clear limited stability properties and is a restrictive way of evolving nodes, although it was particularly easy to implement and model. The implementation of more stable, higher order methods would be an improved way of evolving nodal positions, and may remove some of the restraining issues regarding the magnitude of the timesteps. Evidence in the behaviour of the solution and the negligible timesteps generated by the explicit adaptive timestep method, suggest a stiff system. Since the system was based around a series of second order PDEs and the timestepping method used was only of first order accuracy, it is reasonable to suggest a more stable solution would arise from implementing a method of at least second order accuracy. With this in mind, and via review of *Garcke, et al.* (12), a suitable approach would be implementing a strongly A-stable second order Θ-splitting scheme. The A-stability property of this scheme lends itself to oppose the high frequency oscillations present in the Cahn-Hilliard model, as well as Θ-splitting schemes displaying strong stability properties in general. Alternatively, second order methods which may be easier to implement include a range of implicit Runge-Kutta or Backward Difference schemes.

In order to assess the viability of this moving mesh approach with an alternative timestepping method, we applied an ODE15s stiff system solver in MATLAB to the 1-D mass monitor model. This scheme is a significantly more stable adaptive timestepping method, and could help to show how stiff the system is along with the properties inherent with the moving mesh finite element method used. Inspection of Fig.8.1 shows that merging is still apparent during the formation of the interfacial layers, despite the use of a more stable timestepping scheme. Furthermore, the scheme was unable to produce results after a certain amount of time, due to the timestep being below the minimum tolerance of MATLAB. This suggests that the system is particularly stiff, and that when using a mass monitor function-based moving mesh finite element approach in 1-D, merging is inevitable. Although Fig.8.1 shows merging, it also indicates the time and positions where phase separation occurs. This was also found to be the case with other alternative initial data ($\varphi_1 = tanh(x - \frac{1}{3}) - tanh(x - \frac{1}{3})$

as well as a random function), with results consistently showing merging to occur at $\approx 1.6 \times 10^{-6} s$. Unfortunately, due to time constraints we were unable to implement the ODE15s solver in either the 1-D arclength monitor nor the 2-D models.

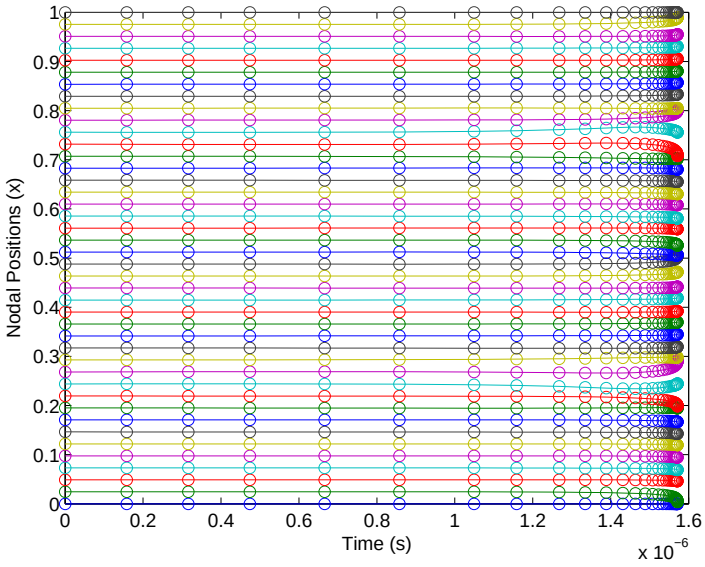


Figure 8.1: Timesteps produced by the ODE15s solver with initial $\varphi_1 = sin(4\pi x)$ in the 1-D mass monitor model

8.2.2 Initial Grid Distribution

For the purposes of this dissertation, a uniform grid of equal node spacing was most frequently used as a base mesh for the initial data in 2-D. Due to time constraints a method to adapt the initial grid in 2-D was not possible. However, a uniform nodal distribution allowed for a simple basis and subsequent formulation of the problem. The systems in 1-D were, however, particularly sensitive and susceptible to node tangling or overtaking. Two simple alternative approaches to the initial mesh were attempted in order to avoid node tangling or overtaking, which soon became a major obstacle in the modelling of the dynamics of the Cahn-Hilliard equation, along with the resolution of steep fronts in initial data. Since the initial data was a piecewise distribution, discontinuous at nodes but continuous in the elements, there was a possibility of steep gradients forming in the initial data, with few points able to resolve and model the evolution of these sharp initial fronts suitably. The approach employed to overcome this problem involved placing additional nodes between nodes with relative intial data differing beyond a certain tolerance level. The number of additional nodes was then determined by the magnitude over the tolerance of the difference in initial data. These additional nodes were then distributed evenly element-wise between the two initial nodes, with corresponding initial data given by linear interpolation. Despite this approach ensuring that the steep fronts were resolved more accurately from the initial data, node tangling was even more apparent, and hence required a heavily reduced timestep.

A suitable approach for avoiding potential node tangling was much tougher to formulate with piecewise linear initial data. A method involving equidistribution of arclength was employed with

smooth initial data, with a view to adapting it to incorporate random piecewise initial data. An iterative solver was employed in order to achieve an equidistribution of the arclength of the initial data, which although helping to prolong the time stepsize, added additional computational expense. Unfortunately, due to the time constraint, further exploration of adaptative methods concerning the initial mesh could not be pursued in greater detail. However, the issue quickly became apparent as a way of introducing a more resolved initial grid against the cost of iterative solvers.

8.2.3 Extensions to Further Work

A velocity-based moving mesh approach has potential for further modelling of the Cahn-Hilliard equation. This is mainly due to the fact there exist many more methods with which it can be applied and still remain a novel approach. Previous work (3) suggests that the method can model thin diffusive interfaces successfully, however, an alternative approach may curtail the large velocities present and remove the likelihood of node tangling and mesh racing. Extensions to the moving mesh approach one may initially like to consider, is to implement a finite difference method (FDM) in 1-D. Finite difference methods have been used almost exhaustively in previous papers on modelling the Cahn-Hilliard equation, although few have attempted employing a Lagrangian-based adaptive mesh method. In particular, velocity-based adaptive mesh methods may have a more significant role to play in developing current models. In using finite differences to approach the problem, one can more easily find the errors apparent in the method. A further point one may pursue includes the weighting or scaling of the monitor function in any future finite element approach. One suggestion for future work may involve a comparison study on the eifficiency of alternative adaptive mesh techniques such as geometric conservation law (9)(14) and moving finite element (20)(19) approaches. However, these methods are also highly susceptible to node tangling, and an alternative approach may need to be adapted significantly in order to address this issue.

Further alternative approaches to tackling the Cahn-Hilliard equation may also be succesful. Certainly, by assessing the individual components of the second derivative of the g term in equation 4.56, and the subsequent discovery of a diffusive Hamilton-Jacobi equation in conservation form, one can see that the presence of a hyperbolic term may also lend itself to a level set method due to its links to conservation laws (22). In fact, level set methods have been developed in recent times to be able to handle multiple interfaces and simulate multiphase fluids (21)(17). However, previous approaches using this method have seemingly only used an Eulerian view of grid adaptation. Hence, a mapping-based moving mesh approach (18), in tandem with a level set method may be something to pursue in greater detail. Alternatively, a joint approach may be applied to model the interfacial layers. In this case, once phase separation occurs one may wish to suppress the diffusion terms present in the system and pursue a conservation law approach, treating the interfacial layers as discontinuities. Hybrid level set methods have already been produced to model the Cahn-Hilliard equation as a joint approach (17), however there exist many alternative methods with which one can model evolving Hamilton-Jacobi discontinuities and which may be used for a different approach.

Bibliography

[1] V. Badalassi, H. Ceniceros, and S. Banerjee, *Computation of multiphase systems with phase field models*, J. Comput. Phys., 190 (2003), pp. 371–397.

[2] M. Baines, M. Hubbard, and P. Jimack, *A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries*, Appl. Numer. Math., 54 (2005), pp. 450–469.

[3] M. Baines, M. Hubbard, P. Jimack, and R. Mahmood, *A moving-mesh finite element method and its application to the numerical solution of phase-change*, Comm. in Comput. Phys., 6 (2009), pp. 595–624.

[4] B. Bhattacharya, *A moving mesh finite element method for high order non-linear diffusion problems*, master's thesis, University of Reading, 2006.

[5] M. Bristeau, R. Glowinski, and J. Periaux, *Numerical methods for the navier-stokes equations: Applications to the simulation of compressible and incompressible viscous flows*, Research Report UH/MD-4, University of Houston, 1987. Computer Physics Report.

[6] C. Budd, W. Huang, and R. Russell, *Adaptivity with moving grids*, Acta Numerica, (2009), pp. 111–241.

[7] J. Cahn and J. Hilliard, *Free energy of a nonuniform system i*, J. Chem. Phys., 28 (1958).

[8] ——, *Free energy of a nonuniform system iii*, J. Chem. Phys., 31 (1959).

[9] W. Cao, W. Huang, and R. Russell, *A moving mesh method based on the geometric conservation law*, SIAM J. Sci. Comput., 24 (2002), pp. 118–142.

[10] H. Ceniceros and A. Roma, *A nonstiff, adaptive mesh refinement-based method for the cahn-hilliard equation*, J. Comput. Phys., 225 (2007), pp. 1849–1862.

[11] X. Feng and O. Karakashian, *Fully discrete dynamic mesh discontinuous galerkin methods for the cahn-hilliard equation of phase transition*, Math. Comp., 76 (2007), pp. 1093–1117.

[12] H. Garcke, M. Rumpf, and U. Weikard, *The cahn-hilliard equation with elasticity: finite element approximation and qualitative studies*, Interfaces and Free Boundaries, 3 (2001), pp. 101–118.

[13] H. Hu, N. Patankar, and M. Zhu, *Direct numerical simulations of fluid-solid systems using the arbitrary lagrangian-eulerian technique*, J. Comput. Phys., 169 (2001), pp. 427–462.

[14] W. Huang and X. Zhan, *Adaptive moving mesh modeling for two dimensional groundwater flow and transport*, Contempory Mathematics, 383 (2004), pp. 283–296.

[15] M. Hubbard, M. Baines, and P. Jimack, *Consistent dirichlet boundary conditions for numerical solution of moving boundary problems*, Appl. Numer. Math., 59 (2009), pp. 1337–1353.

[16] A. Iserles, *A first course in the numerical alaysis of differential equations*, Cambridge texts in applied mathematics, second ed., 2009.

[17] M. Kronbichler and G. Kreiss, *A hybrid level-set-cahn-hilliard model for two-phase flow*, in 1st European Conference on Microfluidics, Bologna, Italy, December 2008.

[18] G. Liao and D. Anderson, *A new approach to grid generation*, Appl. Anal., 44 (1992), pp. 285–298.

[19] K. Miller, *Moving finite elements ii*, SIAM J. Num. Anal., 18 (1981), pp. 1033–1057.

[20] K. Miller and R. Miller, *Moving finite elements i*, SIAM J. Num. Anal., 18 (1981), pp. 1019–1032.

[21] S. Osher and R. Fedkiw, *Level set methods: An overview and some recent results*, J. Comput. Phys., 169 (2001), pp. 463–502.

[22] J. Sethian and P. Smereka, *Level set methods for fluid interfaces*, Ann. Rev. Fluid Mech., 35 (2003), pp. 341–372.

[23] G. Strang, *On the construction and comparison of difference schemes*, SIAM J. Numer. Anal., 5 (1968), pp. 506–517.

[24] P. Yue, C. Zhou, J. Feng, C. Ollivier-Gooch, and H. Hu, *Phase-field simulations of interfacial dynamics in viscoelastic fluids using finite elements with adaptive meshing*, J. Comput. Phys., 219 (2006), pp. 47–67.