A Mo ing Mesh Finite E e ent Appr o ch for the C hn i i rd Eq tion

Peter estwood MSc MoSaIC

August

Acknowledgments

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

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 math

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 inter with the width of the transition layer. Most of the previous methods used (certainly in the case of

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

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

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,

$$\phi \; d\Omega = \text{constant in time.} \tag{1.15}$$

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

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 grinci

Chapter 3

Application to the Cahn-Hilliard Equation

Since, for the Cahn-Hilliard system, we know that the integral, ϕ d

Remark

Care should be taken when introducing either Dirichlet or periodic boundary conditions on ϕ (or Φ) (15). In order to ensure the hat functions remain a part

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

$$\frac{1}{\theta}\int w_i \sqrt{1 + \left(\frac{\partial \phi}{\partial x}\right)^2}\, dx = a_i = \text{constant}, \ \forall$$

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,

$$\frac{\partial \phi}{\partial t} = \frac{\partial\ \phi}{\partial x} + g(\phi\ ) \tag{4.1}$$

$$\phi = \tau\phi - \square\,\frac{\partial\ \phi}{\partial x} \tag{4.2}$$

where

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

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 refinement strategy,

$$\frac{d}{dt}\int w_i \phi\, dx = \int w_i \frac{\partial \phi}{\partial t} dx + \int w_i \frac{\partial(\phi\, v)}{\partial x} dx \tag{4.4}$$

$$= \int w_i \left[\frac{\partial\ \phi}{\partial x} + g(\phi\ )\right] dx + \int w_i \frac{\partial(\phi\ v)}{\partial x} dx \tag{4.5}$$

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

4.1 Calculating the $\frac{\partial \phi}{\partial t}$ 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,

$$\frac{d}{dt}\int w\phi\, dx = 0 = \int \frac{\partial}{\partial t}(w\phi\ )dx + \int \frac{\partial}{\partial x}(w\phi\ \dot{x})dx \tag{4.6}$$

$$= \int \left[w\frac{\partial \phi}{\partial t} + \phi\ \frac{\partial w}{\partial t} + w\frac{\partial}{\partial x}(\phi\ \dot{x}) + \phi\ \dot{x}\frac{\partial w}{}\right.$$

This allows us to reduce equation 4.7 to

$$0 = \int \left[w\frac{\partial \phi}{\partial t} + w\frac{\partial}{\partial x}(\phi\ v) \right] dx \tag{4.9}$$

$$= \int \left[w\frac{\partial^2 \phi}{\partial x^2} + g(\phi\ ) + w\frac{\partial}{\partial x}(\phi\ v) \right] dx \tag{4.10}$$

$$= \left. w\frac{\partial \phi}{\partial x} \right| - \int \frac{\partial w}{\partial x}\frac{\partial \phi}{\partial x} dx + \int w g(\phi\ ) dx + \int w\frac{\partial}{\partial x}(\phi\ v) dx. \tag{4.11}$$

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

$$0 = \left. W_i \frac{\partial \phi}{\partial x} \right| - \int \frac{\partial W_i}{\partial x}\frac{\partial \phi}{\partial x} dx + \int W_i g(\phi\ ) dx + \sum_j \int W_i \frac{\partial}{\partial x}(\phi\ W_j v_j) dx \ \forall i, \tag{4.12}$$

We now consider the terms of equation 4.12 separately. The fir

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 to be $-(\Phi_{i-} - \Phi_i)$, which in matrix form is represented as,

$$B_{\phi_1} = \begin{pmatrix} 0 & -\,{}^{1} & \dots & \dots \\ {}^{0} & 0 & -\,{}^{2} & \dots \\ \vdots & \ddots & \ddots & -\,{}^{N+1} \\ \vdots & & {}^{N} & 0 \end{pmatrix} \tag{4.21}$$

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 $v = \frac{\partial}{\partial x}$. 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,

$$\int \phi \frac{\partial W_i}{\partial x} \sum_j W_j v_j dx = \int \phi \frac{\partial W_i}{\partial x} \sum_j \frac{\partial W_j}{\partial x} \Psi_j dx \ \forall i, \tag{4.22}$$

in matrix form,

$$K_{\phi_1} \underline{\Psi}, \tag{4.23}$$

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

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

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

$$\int W_i v dx = \int W_i \frac{\partial \Psi}{\partial x} dx \tag{4.25}$$

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

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

or, in matrix form,

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

where the matrix B is the unweighted version of B_{ϕ_1} in equation 4.21.

ϕ Term

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

$$\int \frac{\partial W_i}{\partial x} \sum_j \frac{\partial W_j}{\partial x} \phi_j dx = \left[\int \frac{\partial W_i}{\partial x} \sum_j \frac{\partial W_j}{\partial x} dx \right] \underline{\phi} \ \forall i, \tag{4.29}$$

which in matrix form can be expressed as

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

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

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

Remark

Again, great care should be taken when introducing the periodic boundary conditions into the system. This requires a reduced system with the test functio

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,

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

With the new nodal positions known, the new ϕ 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

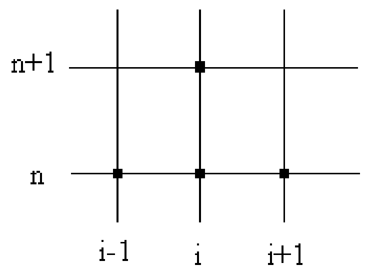


Figure 4.1: 2-D stencil representing nodes at n and n + 1 time levels, displaying an explicit timestepping approach.

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 equationo

This restriction on the timestep ensures that no x_i^n -values move beyond x_{i-}^n or x_i^n . Therefore, in order for the system to have monotonically increasing x-values, it is sufficient to reduce the Δt to $\frac{\Delta t}{}$. 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), monotonically increasing set of x-values for any Δt.

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, ϕ 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, ϕ and Ψ that we had had prior to the discontinuity present just before merging.

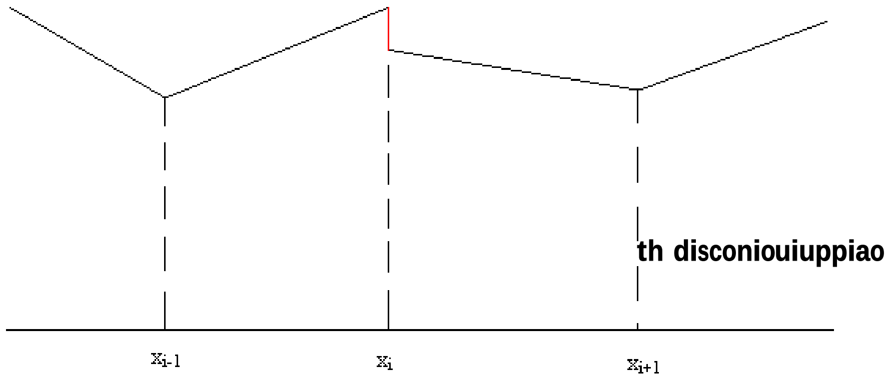


Figure 4.3: Graph representing a discontinuity present in the piecewise solution.

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 $f(\varphi)$, a Hamilton-Jacobi system can be a direct outcome of the Cahn-Hilliard system we are using. In more general cases, once phase separatie31218399.547(

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 e ective to 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,

$$\dot{\theta} = \frac{\frac{\partial \phi_1^n}{\partial x}\frac{\partial^2 \phi_1^n}{\partial x \partial t}}{\sqrt{1 + \left(\frac{\partial \phi_1^n}{\partial x}\right.}} dx + \left(v\sqrt{1 + \frac{\partial \phi^n}{\partial x}} \right)\Bigg| \quad . \tag{5.6}$$

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

$$\dot{\theta} = \frac{\frac{\partial \phi_1^n}{\partial x}\frac{\partial^2 \phi_1^n}{\partial x \partial t}}{\sqrt{1 + \left(\frac{\partial \phi_1^n}{\partial x}\right.}} dx. \tag{5.7}$$

We can now substitute in for $\frac{\partial \phi_1}{\partial t}$ using the previously defined equation 4.1, i.e.

$$\frac{\partial \phi}{\partial t} = \frac{\partial \ \phi}{\partial x} + \frac{\partial}{\partial x}(\phi \ - (\tau + 1)\phi \ ), \tag{5.8}$$

to give

$$\dot{\theta} = \frac{\frac{\partial \phi_1^n}{\partial x}\frac{\partial}{\partial x}\left(\frac{\partial^2 \phi_2^n}{\partial x^2} + \frac{\partial^2}{\partial x^2}[(\phi^n) \ - (\tau + 1)\phi^n]\right.}{\sqrt{1 + \left(\frac{\partial \phi_1^n}{\partial x}\right.}} dx. \tag{5.9}$$

Upon integration element by element

$$\dot{\theta} = \sum_{\text{elements}\,k} \left[\frac{\frac{\partial \phi_1^n}{\partial x}\Big|_{k-\frac{1}{2}} \left(\frac{\partial^2 \phi_2^n}{\partial x^2} + \frac{\partial^2}{\partial x^2}[(\phi^n) \ - (\tau + 1)\phi^n] \ \Big|_{k-}^{k}\right.}{\sqrt{1 + \left(\frac{\partial \phi_1^n}{\partial x}\right.}\Big|_{k-\frac{1}{2}}} \right]. \tag{5.10}$$

Coming back to equation 5.5, we now have only one remaining unknown, v. In order to include the newly determined $\dot{\theta}$

where K is the standard sti□ness matrix, $\underline{g} = (\underline{\Phi} - (\tau + 1)\underline{\Phi})$ and B_{ϕ_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 $\dot{\underline{\sigma}}$, we can then find the new $\underline{\sigma}$ values using the standard Eulerian formula,

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

Finally, in order to calulate the new ϕ 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 and σ values respectively, as in equation 3.5.

5.2 Calculating φ_2 values

The formulation for ϕ remains much more simple. This requires expanding the weak form of equation wht

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.

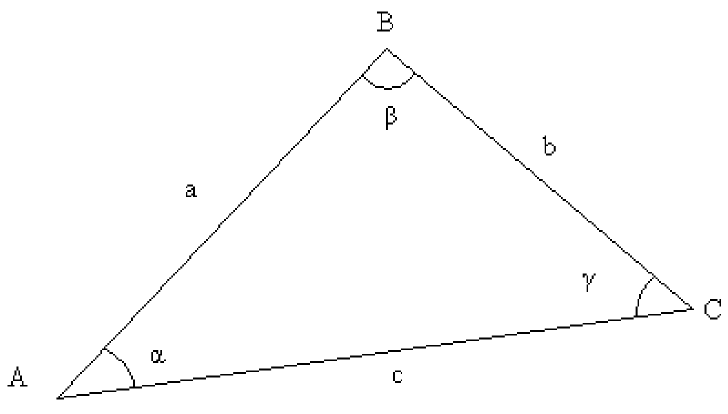


Figure 6.1: A 2-D triangular element.

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.

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

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 where

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

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

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

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

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,

$$\underline{x}^n \quad - \underline{x}^n$$

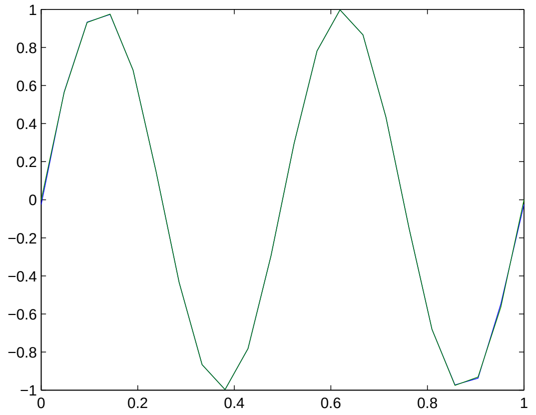
Chapter 7

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 an

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

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



(a)

n=22,

L_2-

norm=0.0398

7.3 2-D Case

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

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

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,

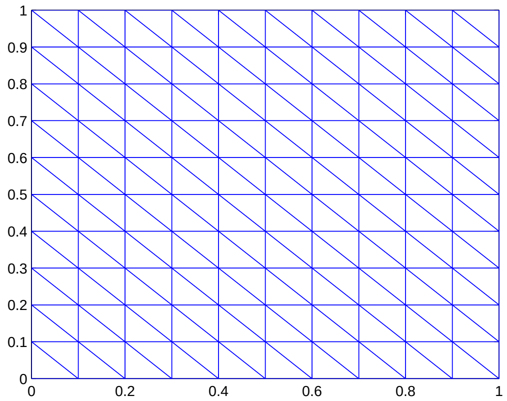
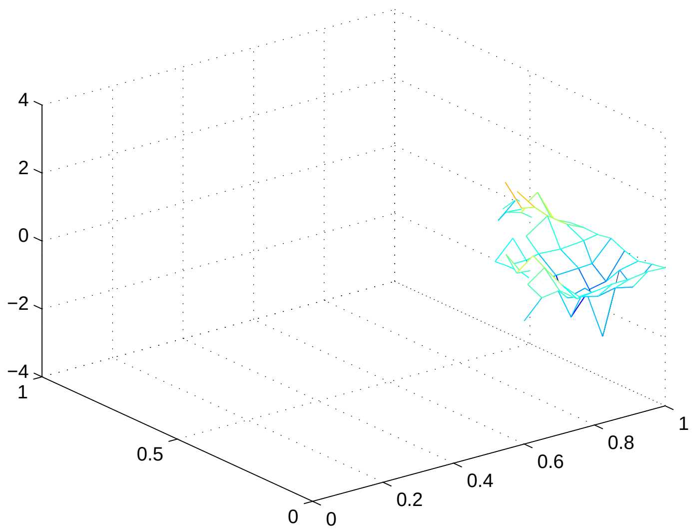


Figure 7.3: Delaunay triangulation of the unit square grid

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


4
2
0
−2
−4
1
0.5
0
0
0.2
0.4
0.6
0.8
1

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

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^{-} being presented as maximum timesteps in the 1-D conservation

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] ——,

[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