University of Sussex

Mmath Project

Finite Elements approximation of motion by mean curvature in $\Re^2$

Author:
Joe Eyles

Supervisor:
Dr. Vanessa Styles

13th May 2013

Contents

1 Introduction

The purpose of this project is to model the evolution of closed curves $\Gamma(t)$, fig. 1, in $\Re^2$. It is intended that the velocity of the curves should be proportional to the curvature of the curve, and also that a forcing term could be added to the equation. Thus our velocity is given by

$$\underline{V} = -\kappa\underline{\hat{n}} + f\underline{\hat{n}}, \tag{1}$$

where κ is the curvature, f is the forcing term, and $\underline{\hat{n}}$ is the unit normal to the curve.

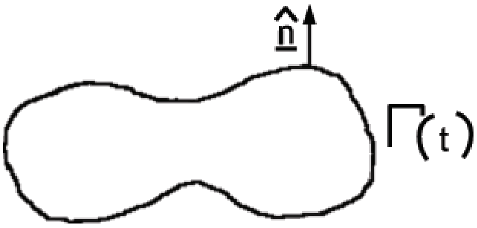

Figure 1: An example of the curve $\Gamma(t)$ with unit normal $\underline{\hat{n}}$ and curvature $-\kappa$.

Two approaches to numerically approximate this equation are taken, the first is a parametric approach. Here the curve is parametrised by $\theta \in [0, 2\pi]$, and then the resulting parametrised equation is approximated using a finite element method. This method works in simpler cases, but cannot deal with topology changes.

The other approach is to use the Allen-Cahn equation, which models mean curvature flow with smooth transitions between phase spaces. The equation will be derived by calculus of variations, and then approximated with a finite element approach. This is the more powerful method, as it does not break down in topology changes. Within the Allen-Cahn equation, there is the opportunity to choose different functions for the free energy. Two possible functions will be explored.

The resulting system of algebraic equations for both of these approaches are programmed in MATLAB, there are a number of variants of each approach, for example explicit and semi-implicit schemes. Some of the properties that these programs would ideally have are accuracy, efficiency (or speed) and an ability to handle a topology change.

Mean curvature flow has a number of applications, including image processing, the motion of grain boundaries in alloys, and phase transitions. In image processing, variants of this equation are used to find the boundaries of objects, which is an important problem in the realm of computer vision.

A grain boundary is the boundary between two crystallites, which are microscopic crystals. Here the surface tension is equal to the curvature, often with a forcing term.

Phase transition is the change of a material from one state to another, for example gas to liquid. This will tend to happen on the boundaries, which will shrink - similar to our model.

2 The Parametric Approach

The first approach we will consider is a parametric solution of equation (1). We parametrise our curve by a variable $\theta \in [0, 2\pi]$, to give us $\underline{X}(\theta) = \begin{pmatrix} x(\theta) \\ y(\theta) \end{pmatrix}$. As we have a closed curve, we need the periodic boundary condition that $\underline{X}(0, t) = \underline{X}(2\pi, t)$ $\forall t > 0$.

2.1 The Equation of Motion in Parametric form

Noting that in parametric form,

$$\underline{\hat{n}} = \frac{1}{|\underline{X}_\theta|} \begin{pmatrix} -y_\theta \\ x_\theta \end{pmatrix} = \frac{\underline{n}}{|X_\theta|},$$

and that:

$$\kappa = \frac{1}{|\underline{X}_\theta|} \langle \left(\frac{\underline{X}_\theta}{|\underline{X}_\theta|} \right)_\theta, \underline{n} \rangle,$$

we can rewrite (1) as:

$$\underline{V} = -\frac{1}{|\underline{X}_\theta|} < \left(\frac{\underline{X}_\theta}{|\underline{X}_\theta|} \right)_\theta, \underline{n} > \underline{n} + f \frac{\underline{n}}{|\underline{X}_\theta|},$$

which, as $\underline{V} = \underline{X}_t$, can be rearranged to:

$$\underline{X}_t - \frac{1}{|\underline{X}_\theta|} \left(\frac{\underline{X}_\theta}{|\underline{X}_\theta|} \right)_\theta - \frac{f(\theta)\underline{n}}{|\underline{X}_\theta|} = 0, \forall t > 0, \theta \in [0, 2\pi]. \tag{2}$$

2.2 Deriving the Finite Element Method

We wish to derive a finite element approximation of (2). For $\theta \in I$, where $I = [0, 2\pi]$ and $\underline{X} = \underline{X}(\theta, t) \in \Re^2$. With periodic boundary condition $\underline{X}(\theta, t) = \underline{X}(\theta + 2\pi, t)$, $t > 0$, and initial condition $\underline{X}(\theta, 0) = \underline{X}^0(\theta)$.

First we multiply through by $|\underline{X}_\theta|$ and a test function $\rho \in V := \{\rho \in H^1(I)$ s.t. $\rho(0) = \rho(2\pi)\}$, and then integrate over I:

$$\int_I \underline{X}_t |\underline{X}_\theta| \rho d\theta - \int_I \left(\frac{\underline{X}_\theta}{|\underline{X}_\theta|} \right)_\theta \rho d\theta - \int_I f(\theta) \underline{n} \rho d\theta = 0, \forall \rho \in V.$$

We then integrate the second term by parts:

$$\int_I \underline{X}_t |\underline{X}_\theta| \rho d\theta + \int_I \frac{\underline{X}_\theta}{|\underline{X}_\theta|} \rho_\theta d\theta - \left[\frac{\underline{X}_\theta}{|\underline{X}_\theta|} \rho \right]_{\theta=0}^{\theta=2\pi} - \int_I f(\theta) \underline{n} \rho d\theta = 0.$$

Note that:

$$\left[\frac{\underline{X}_\theta}{|\underline{X}_\theta|}\rho\right]_{\theta=0}^{\theta=2\pi} = \frac{\underline{X}_\theta(2\pi)}{|\underline{X}_\theta(2\pi)|}\rho(2\pi) - \frac{\underline{X}_\theta(0)}{|\underline{X}_\theta(0)|}\rho(0) = 0,$$

as we have the periodic boundary conditions. This means that we are left with:

$$\int_I \underline{X}_t|\underline{X}_\theta|\rho d\theta + \int_I \frac{\underline{X}_\theta}{|\underline{X}_\theta|}\rho_\theta d\theta - \int_I f(\theta)\underline{n}\rho d\theta = 0. \tag{3}$$

We now derive a semi discrete finite element approximation of (3). Following the techniques introduced in [3], we discretize the interval I into $N-1$ uniform intervals with N nodes, $0 = x_1, x_2, \ldots, x_N = 2\pi$, with the distance between each node equal to h. Consider the space

$$V_h = \{\rho \text{ s.t. } \rho(0) = \rho(2\pi) \text{ and } \rho \text{ is piecewise affine on } I_j\}$$

where $I_j = [x_j, x_j + 1], j = 1, \ldots, N-1$. Endow the space V_h with the usual nodal basis, $\{\phi_j(\theta)\}_{j=1}^N$. Then we have the discrete problem:

$$\int_I (\underline{X}_h)_t|(\underline{X}_h)_\theta|\rho_h d\theta + \int_I \frac{(\underline{X}_h)_\theta}{|(\underline{X}_h)_\theta|}\rho_{h_\theta} d\theta - \int_I f_h(\theta)\underline{n}\rho_h d\theta = 0, \forall \rho_h \in V_h, \forall t > 0,$$

where $\underline{X}_h(\theta, t) = \sum_{i=1}^N \underline{X}_i(t)\phi_i(\theta)$, $f_h(\theta) = \sum_{i=1}^N f_i(\theta)\phi_i(\theta)$ and the test function $\rho_h(\theta) = \phi_j(\theta)e^k$ for $j = 1, \ldots, N$, $k = 1$ or 2, and e^k is the standard basis in $\Re^2$. Finally let

$$q_j = |\underline{X}_j - \underline{X}_{j-1}|. \tag{4}$$

Thus we have:

$$\begin{aligned} 0 &= \int_I \sum_{i=1}^N (\underline{X}_i)_t \phi_i\phi_j|(\underline{X}_h)_\theta|e^k d\theta + \int_I \frac{\sum_{i=1}^N \underline{X}_i(\phi_i)_\theta}{|(\underline{X}_h)_\theta|}(\phi_j)_\theta e^k d\theta \\ &\quad - \int_I \sum_{i=1}^N f_i\phi_i\underline{n}\phi_j e^k d\theta = 0, \forall j \in [1, \ldots, N], \forall t > 0. \end{aligned}$$

Considering each integral individually, we first have:

$$\begin{aligned} \int_I \sum_{i=1}^N (\underline{X}_i)_t\phi_i\phi_j|(\underline{X}_h)_\theta|e^k d\theta &= \int_{(j-1)h}^{jh} \left((\underline{X}_{j-1})_t\phi_{j-1}\phi_j + (\underline{X}_j)_t\phi_j\phi_j\right)|(\underline{X}_h)_\theta|e^k d\theta \\ &\quad + \int_{jh}^{(j+1)h} \left((\underline{X}_j)_t\phi_j\phi_j + (\underline{X}_{j+1})_t\phi_{j+1}\phi_j\right)|(\underline{X}_h)_\theta|e^k d\theta, \\ &\quad \forall j \in [1, \ldots, N]. \end{aligned}$$

Here $(N+1)h = x_{N+1} = x_2$, due to the periodic boundary conditions. We will use mass lumping, which is defined as:

$$\int_I \phi_i \phi_j d\theta = \left\{ \begin{array}{ll} \int_I \phi_i d\theta & \text{if } i = j \\ 0 & \text{if } i \neq j \end{array} \right. . \tag{5}$$

Thus we have:

$$\begin{aligned} \int_I \sum_{i=1}^{N} (\underline{X}_i)_t \phi_i \phi_j |(\underline{X}_h)_\theta| e^k d\theta &= \int_{(j-1)h}^{jh} (\underline{X}_j)_t \phi_j |(\underline{X}_h)_\theta| e^k d\theta \\ &\quad + \int_{jh}^{(j+1)h} (\underline{X}_j)_t \phi_j |(\underline{X}_h)_\theta| e^k d\theta, \forall j \in [1, \ldots, N]. \end{aligned}$$

Since we have that, on the interval $[(j-1)h, jh]$, $(\underline{X}_h)_\theta = \frac{\underline{X}_j - \underline{X}_{j-1}}{h}$. Likewise, on the interval $[jh, (j+1)h]$, we have $(\underline{X}_h)_\theta = \frac{\underline{X}_{j+1} - \underline{X}_j}{h}$. This means, upon noting (4), we are left with:

$$\int_I \sum_{i=1}^{N} (\underline{X}_i)_t \phi_i \phi_j |(\underline{X}_h)_\theta| e^k d\theta = \frac{(\underline{X}_j)_t}{2} \left(q_j + q_{j+1} \right) e^k, \forall j \in [1, \ldots, N].$$

The next integral we consider is:

$$\begin{aligned} & \int_I \frac{\sum_{i=1}^{N} \underline{X}_i}{|(\underline{X}_h)_\theta|} \phi_{i_\theta} \phi_{j_\theta} e^k d\theta \\ = & \int_{(j-1)h}^{jh} \frac{(\underline{X}_h)_\theta}{|(\underline{X}_h)_\theta|} (\phi_j)_\theta e^k d\theta + \int_{jh}^{(j+1)h} \frac{(\underline{X}_h)_\theta}{|(\underline{X}_h)_\theta|} (\phi_j)_\theta e^k d\theta \\ = & \int_{(j-1)h}^{jh} \frac{\underline{X}_j - \underline{X}_{j-1}}{h} \frac{h}{q_j} (\phi_j)_\theta e^k d\theta + \int_{jh}^{(j+1)h} \frac{\underline{X}_{j+1} - \underline{X}_j}{h} \frac{h}{q_{j+1}} (\phi_j)_\theta e^k d\theta. \end{aligned}$$

So we have:

$$\int_I \frac{\sum_{i=1}^{N} \underline{X}_i (\phi_j)_\theta}{|(\underline{X}_h)_\theta|} (\phi_j)_\theta e^k d\theta = \left(\frac{\underline{X}_{j+1} - \underline{X}_j}{q_{j+1}} - \frac{\underline{X}_j - \underline{X}_{j-1}}{q_j} \right) e^k.$$

The final integral to consider is:

$$\int_I \sum_{i=1}^{N} f_i \phi_i \underline{n} \phi_j e^k d\theta.$$

Since we know that

$$\underline{n} = \begin{pmatrix} -y_\theta \\ x_\theta \end{pmatrix} = \sum_{i=1}^{N} \begin{pmatrix} -y_i \\ x_i \end{pmatrix} (\phi_j)_\theta,$$

we have:

$$\begin{aligned}\int_I \sum_{i=1}^N f_i \phi_i \underline{n} \phi_j e^k d\theta &= \begin{pmatrix} -y_{j-1} \\ x_{j-1} \end{pmatrix} \int_{(j-1)h}^{jh} \sum_{i=1}^N f_i \phi_i (\phi_{j-1})_\theta \phi_j e^k d\theta \\ &+ \begin{pmatrix} -y_j \\ x_j \end{pmatrix} \int_{(j-1)h}^{(j+1)h} \sum_{i=1}^N f_i \phi_i) (\phi_j)_\theta \phi_j e^k d\theta \\ &+ \begin{pmatrix} -y_{j+1} \\ x_{j+1} \end{pmatrix} \int_{jh}^{(j+1)h} \sum_{i=1}^N f_i \phi_i (\phi_{j+1})_\theta \phi_j e^k d\theta.\end{aligned}$$

Note that $(\phi_j)_\theta = \frac{1}{q_j}$ on $[(j-1)h, jh]$, and so is constant. It is similarly constant on $[jh, (j+1)h]$. Also using mass lumping (5), calculating the integrals yield:

$$\int_I \sum_{i=1}^N f_i \phi_i \underline{n} \phi_j e^k d\theta = \begin{pmatrix} -y_{j-1} \\ x_{j-1} \end{pmatrix} \frac{-f_j}{2} e^k + \begin{pmatrix} -y_{j+1} \\ x_{j+1} \end{pmatrix} \frac{f_j}{2} e^k = \frac{-f_j}{2} e^k \begin{pmatrix} y_{j-1} - y_{j+1} \\ -x_{j-1} + x_{j+1} \end{pmatrix}.$$

Now, if we set $(\underline{X}_j)_t = \frac{\underline{X}_j^{n+1} - \underline{X}_j^n}{\Delta t}$, for time step Δt, then we have the explicit scheme:

$$\underline{X}_j^{n+1} = \frac{2\Delta t \left(\frac{\underline{X}_{j+1}^n - \underline{X}_j^n}{q_{j+1}^n} - \frac{\underline{X}_j^n - \underline{X}_{j-1}^n}{q_j^n} + \frac{\underline{X}_j^n}{2\Delta t} \left(q_j^n + q_{j+1}^n \right) + \begin{pmatrix} -y_{j-1}^n \\ x_{j-1}^n \end{pmatrix} \frac{-f_j}{2} + \begin{pmatrix} -y_{j+1}^n \\ x_{j+1}^n \end{pmatrix} \frac{f_j}{2} \right)}{q_j^n + q_{j+1}^n}. \tag{6}$$

And the semi-implicit scheme:

$$\begin{aligned} & 2\underline{X}_j^{n+1} \left(\frac{q_j^n}{2} + \frac{q_{j+1}^n}{2} + \frac{\Delta t}{q_{j+1}^n} + \frac{\Delta t}{q_j^n} \right) - \underline{X}_{j+1}^{n+1} \frac{2\Delta t}{q_{j+1}^n} - \underline{X}_{j-1}^{n+1} \frac{2\Delta t}{q_j^n} \\ = \quad & \underline{X}_j^n \left(q_j^n + q_{j+1}^n \right) + 2\Delta t \left(\begin{pmatrix} -y_{j-1}^n \\ x_{j-1}^n \end{pmatrix} \frac{-f_j}{2} + \begin{pmatrix} -y_{j+1}^n \\ x_{j+1}^n \end{pmatrix} \frac{f_j}{2} \right). \end{aligned} \tag{7}$$

Note that the matrix that we need to invert is of tridiagonal form, with the $i = 1, j = N$ and $i = N, j = 1$ elements non zero. This is called a tridiagonal matrix with cyclic boundary conditions.

2.3 Solving a Tridiagonal System with Cyclic Boundary Conditions

2.3.1 Solving a Tridiagonal System

A tridiagonal matrix is one such that the leading diagonal and two adjacent diagonals are possibly non zero, but all other entries are zero. In short, A is tridiagonal if:

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & \dots & & & 0 \\ a_{21} & a_{22} & a_{23} & & & & \vdots \\ 0 & \ddots & \ddots & \ddots & & & \\ \vdots & & & & & & 0 \\ & & & & a_{N-1,N-2} & a_{N-1,N-1} & a_{N-1,N} \\ 0 & \dots & & & 0 & a_{N,N-1} & a_{N,N} \end{pmatrix}.$$

These matrices can be inverted using the Crout factorization, see Algorithm. 1.

Algorithm 1 Crout factorization for a tridiagonal System

```
Let $A$ be an $N \times N$ tridiagonal matrix, and consider the system $Ax = b$. We will also let $U$ and $L$ be $N \times N$ matrices, although we will only use the leading diagonal and one adjacent diagonal. Finally let $z$ be a vector of length $N$. Algorithm taken from [1].
STEP 1
    $l_{11} = a_{11}$
    $u_{12} = \frac{a_{12}}{l_{11}}$
    $z_1 = \frac{b_1}{l_{11}}$
STEP 2
for $i = 2 \dots n-1$ do
    $l_{i,i-1} = a_{i,i-1}$
    $l_{ii} = a_{ii} - l_{i,i-1}u_{i-1,i}$
    $u_{i,i+1} = \frac{b_i}{l_{ii}}$
    $z_i = \frac{b_i - l_{i,i-1}z_{i-1}}{l_{ii}}$
end for
STEP 3
    $l_{n,n-1} = a_{n,n-1}$
    $l_{nn} = a_{nn} - l_{n,n-1}u_{n-1,n}$
    $z_n = \frac{b_n - l_{n,n-1}z_{n-1}}{l_{nn}}$
STEP 4
    $x_n = z_n$
STEP 5
for $i = n-1 \dots 1$ do
    $x_i = z_i - u_{i,i+1}x_{i+1}$
end for
    Return $x$
```

2.3.2 Sherman-Morrison Formula

Given a matrix A, suppose we know A^{-1}. If we change an element in A, to give $\overline{A}$, we would like to be able to compute $\overline{A}^{-1}$ quickly and efficiently. We have $\overline{A} = A + \underline{u} \otimes \underline{v}$, where $\underline{u}$ is a column vector and $\underline{v}$ is a row vector, and $\otimes$ is defined by $\underline{u} \otimes \underline{v} = B$ where the ij^{th} element of B is given by $b_{ij} = u_i v_j$. From [2] we have the following Theorem.

Theorem: Sherman-Morrison Formula. *Given an $N \times N$ matrix A, its inverse A^{-1}, and two vectors of length N, denoted $\underline{u}$ and $\underline{v}$, we have that the inverse of $\overline{A} = A + \underline{u} \otimes \underline{v}$ is:*

$$\overline{A}^{-1} = A^{-1} - \frac{\left(A^{-1}\underline{u}\right) \otimes \left(\underline{v}A^{-1}\right)}{1 + \underline{v}A^{-1}\underline{u}}.$$

Proof.

$$\left(A + \underline{u} \otimes \underline{v}\right)^{-1} = A^{-1}\left(1 + A^{-1}\underline{u} \otimes \underline{v}\right)^{-1}.$$

By the series expansion of $(1+x)^{-1}$,

$$\begin{aligned}
&= \left(1 - A^{-1}\underline{u} \otimes \underline{v} + A^{-1}\underline{u} \otimes \underline{v}A^{-1}\underline{u} \otimes \underline{v} \ldots\right) A^{-1} \\
&= A^{-1} - A^{-1}\underline{u} \otimes \underline{v}A^{-1} + A^{-1}\underline{u} \otimes \underline{v}A^{-1}\underline{u} \otimes \underline{v}A^{-1} \ldots \\
&= A^{-1} - \left(A^{-1}\underline{u}\right) \otimes \left(\underline{v}A^{-1}\right) + \left(A^{-1}\underline{u}\right) \otimes \left(\underline{v}A^{-1}\right) \underline{v}A^{-1}\underline{u} \ldots \\
&= A^{-1} - \left(A^{-1}\underline{u}\right) \otimes \left(\underline{v}A^{-1}\right) \left(1 - \left(\underline{v}A^{-1}\underline{u}\right) + \left(\underline{v}A^{-1}\underline{u}\right)^2 - \left(\underline{v}A^{-1}\underline{u}\right)^3 \ldots\right) \\
&= A^{-1} - \frac{\left(A^{-1}\underline{u}\right) \otimes \left(\underline{v}A^{-1}\right)}{1 + \underline{v}A^{-1}\underline{u}}.
\end{aligned}$$

$\square$

2.3.3 How the Solution to a Tridiagonal System and the Sherman-Morrison Formula are Applied

In the Parametric approach, we have a system with a tridiagonal matrix with cyclic boundary conditions, $A\underline{x} = \underline{b}$. We want to calculate this efficiently, without storing any matrix explicitly. To do this we must first consider the Sherman-Morrison formula. So let B consist of the tridiagonal elements of A, with zeros everywhere else, and let $\underline{u} := \left(a_{1n}, 0, \ldots 0, a_{n1}\right)^t$ and $\underline{v} := (1, 0, \ldots 01)$. Thus we have $A = B + \underline{u} \otimes \underline{v}$.

We want to calculate $\underline{x} = A^{-1}\underline{b}$, which is equivalent to:

$$\underline{x} = \left(B + \underline{u} \otimes \underline{v}\right)^{-1} \underline{b}.$$

Which, by the Sherman-Morrison formula, is equivalent to:

$$\underline{x} = \left(B^{-1} - \frac{\left(B^{-1}\underline{u}\right) \otimes \left(\underline{v}B^{-1}\right)}{1 + \underline{v}B^{-1}\underline{u}}\right) \underline{b},$$

$$\underline{x} = B^{-1}\underline{b} - \frac{\left(B^{-1}\underline{u}\right) \otimes \left(\underline{v}\left(B^{-1}\underline{b}\right)\right)}{1 + \underline{v}B^{-1}\underline{u}}).$$

Now we have $B^{-1}b$, $B^{-1}\underline{u}$ and $B^{-1}\underline{b}$, all of which can be quickly and efficiently calculated by the algorithm given for a tridiagonal system, Algorithm 1.

3 Numerical Results for the Parametric Approach

In this section we present some numerical results. The programs parameters are the time step, Δt, and the number of nodes, N. The forcing term and the parametrisation of the curve are also declared at the start of the program.
A drawback of the parametric approach is simply that the curve must be parametrised. Although for some shapes this can be relatively easy (for example a circle or an ellipse), for other shapes this can be difficult. This is, however, somewhat true for any scheme: calculating and programming initial data for a complicated shape can be difficult.

3.1 Explicit

Fig. 2 shows a curve evolving in the explicit scheme, with no forcing term. A small value of Δt has purposefully been chosen to highlight the difficulties. The curve shrinks somewhat as expected, although when it becomes very small the shape looses all coherence. Here the nodes are too close together, and the approximated solution goes awry. As can be seen, the value of Δt is not small enough, and as a result causes the curve to taken on a jagged appearance. An example of the same curve, but with a smaller value of Δt can be seen in fig. 3. The smaller value of Δt leads to a more stable approximation. This effect is later explained by the maximum principle, which tells us that for a small enough Δt the numerical approximation gives rise to curves that evolve as expected.

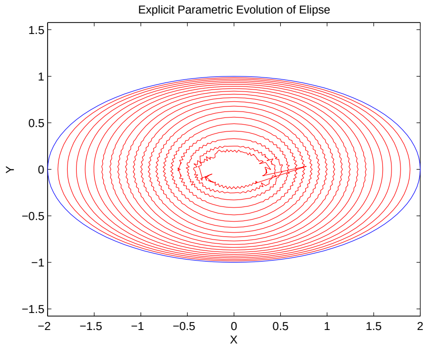

Figure 2: The blue ellipse is the initial curve, the co-centric red lines show the curve evolving over time, with $f = 0$. $\Delta t = 0.0005$, which is too large.

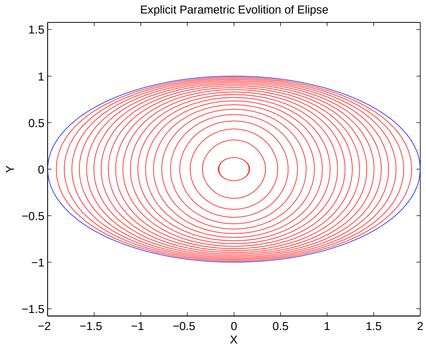

Figure 3: The blue ellipse is the initial curve, the co-centric red lines show the curve evolving over time, with $f = 0$. $\Delta t = 0.00005$, which is small enough.

3.2 Semi-Implicit

The semi-implicit scheme can be seen evolving in fig. 4. This scheme has less of the short comings of the explicit scheme. Here an ellipse is used, to illustrate that for greater curvature, the velocity of the curve is greater. Note that the solution is much smoother, due to there being no bounds on the time step.

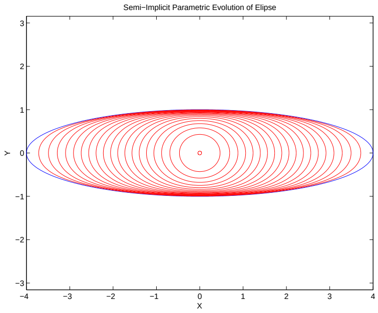

Figure 4: The blue ellipse is the initial curve, the co-centric red lines show the curve evolving over time.

3.3 The Effect of the Forcing Term

The forcing term is a function of θ, so $f = f(\theta)$, and pushes the curve at each point in the normal direction. This can be seen in fig. 5. Here the forcing term is given by:

$$f(\theta_i) = \begin{cases} 0.0003\theta_i^2 & \text{if } \theta_i \leq \frac{Nh}{2} \\ 0.0003(Nh - \theta_i)^2 & \text{if } \theta_i > \frac{Nh}{2} \end{cases} .$$

This caused the left hand side of the curve to bend inwards at a much greater speed than the right hand side.

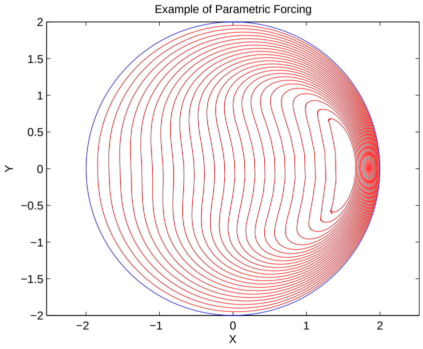

Figure 5: The circle ellipse is the initial curve, the co-centric red lines show the curve evolving over time.

Given a circle, we know that the curvature is $\frac{1}{r}$, and thus the equation of motion is $\underline{V} = -\frac{1}{r}\underline{\hat{n}} + f\underline{\hat{n}}$. So if we take $f = \frac{1}{r}$, then we have $\underline{V} = 0$, and so expect the curve to remain stationary, as in fig. 6. Here $r = 1$, $\Delta t = 0.0001$ and $N = 100$.

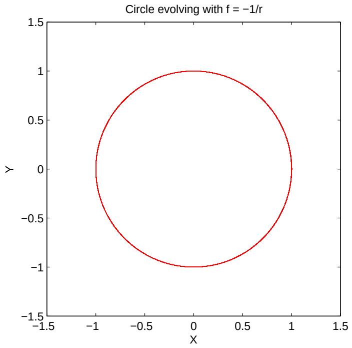

Figure 6: A circle with $f = \frac{1}{r}$, note how it does not move.

3.4 The Speed Test

As we do not prove the convergence of our finite element approximation, we will do some computations to show that it seems to correctly approximate our model. To this end we will investigate the speed at which the curve evolves. To do this we will use a circle, so we must first solve the true solution for a circle. We know that the velocity is given by the curvature, so, with r as the radius, we must solve the differential equation (using the fact that the curvature of a

Table 1: The average error in the speed test for the parametric approach, showing different values for N and Δt. The error was measured with a radius ranging between $0.25 \leq r \leq 1$.

		N		
		10	10^2	10^3
Δt	10^{-1}	0.1542	0.1183	0.1191
	10^{-3}	0.1354	1.5×10^{-3}	1.3×10^{-3}
	10^{-6}	0.1811	1.9×10^{-3}	1.8630×10^{-5}

circle is $\frac{1}{r}$):

$$\begin{cases} \frac{dr}{dt} = -\frac{1}{r} \\ r(0) = r_0 \end{cases} .$$

Using separation of variables:

$$\int r dr = \int -1 dt$$

$$\Rightarrow \frac{r^2}{2} = -t + c$$

$$\Rightarrow r = \sqrt{-2t + c}.$$

Considering the initial condition:

$$r_0 = \sqrt{c},$$

our solution is:

$$r = \sqrt{-2t + r_0^2}.$$

Comparing the exact value of r with the value approximated by the semi-implicit scheme gave the average errors displayed in Table 1. A number of different values of Δt and N were used. We can see that for smaller values of Δt and larger values of N the error is smaller, and thus the approximation is better. For large enough N and small enough Δt the error is adequately small, as we had $r_0 = 1$.

3.5 Topology Change

One of the major issues with the parametric approach is that of self intersection. Namely the algorithm has no way of knowing when the lines are self intersecting, and thus will carry on regardless. This can be seen in fig. 7. Here the forcing term pushes the top and bottom of the curve together. When the curves meet, they pass through each other and keep going. The forcing taken is given by:

$$f(\theta_i) = \begin{cases} 0.001\theta_i^2 & \text{if } \theta_i = 1, \dots, \frac{Nh}{4} \\ 0.001\left(\frac{Nh}{2} - \theta_i\right)^2 & \text{if } \theta_i = \frac{Nh}{4}, \dots, \frac{Nh}{2} \\ 0.001\left(\theta_i - \frac{Nh}{2}\right)^2 & \text{if } \theta_i = \frac{Nh}{2}, \dots, \frac{3Nh}{4} \\ 0.001\left(Nh - \theta_i\right)^2 & \text{if } \theta_i = \frac{3Nh}{4}, \dots, Nh \end{cases} .$$

This is not what we want to happen - we would prefer it if the curve "pinched off" into two separate curves. It is for this reason that we later consider the phase field approach.

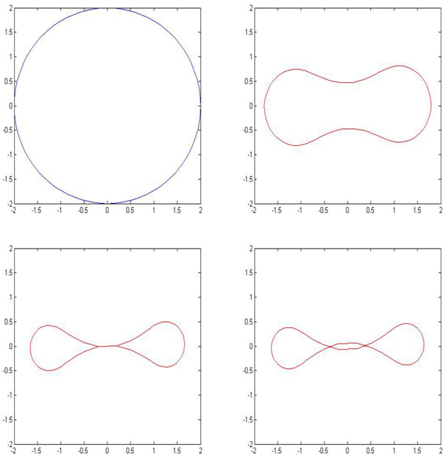

Figure 7: Top left: The initial starting shape (a circle). Top right: The forcing term begins to push the top and bottom of the shape together. Bottom left: The curve intersects with itself. Bottom right: The curve passes through itself.

3.6 The Maximum Principle

It was previously mentioned that that the value of Δt would effect the stability of the solution. Here we will prove the maximum principle for the explicit and semi-implicit case, deriving a bound on Δt only for the explicit case.

3.6.1 Explicit Scheme with $f = 0$

Theorem: Maximum Principle for the Parametric Explicit Scheme with $f = 0$. *We have, for boundary conditions $\underline{X}_b$, that:*

$$\max_{j\in[1,N]} |\underline{X}_j^n| \leq \max_{j\in[1,N]} \left(|\underline{X}_j^0|, |\underline{X}_b|\right),$$

if and only if $\frac{q_j^n q_{j+1}^n}{2} > \Delta t$.

Proof. We will prove this by induction. The base case is:

$$\max_{j\in[1,N]} |\underline{X}_j^0| \leq \max_{j\in[1,N]} \left(|\underline{X}_j^0|, |\underline{X}_b|\right),$$

which is clearly true.

Now, for the inductive step, assume that the statement is true for n, and we will show it is true for $n+1$. The explicit scheme is given by:

$$\begin{aligned} |\underline{X}_j^{n+1}| &= \left| \frac{2\Delta t}{q_j^n + q_{j+1}^n} \left(\frac{\underline{X}_{j+1}^n}{q_{j+1}^n} - \frac{\underline{X}_j^n}{q_{j+1}^n} - \frac{\underline{X}_j^n}{q_j^n} + \frac{\underline{X}_{j-1}^n}{q_j^n} + \underline{X}_j^n \right) \right|, \\ &= \left| \underline{X}_j^n \left(1 - \frac{2\Delta t}{q_j^n q_{j+1}^n} \right) + \frac{2\Delta t}{q_j^n q_{j+1}^n} \left(\frac{\underline{X}_{j+1}^n}{q_{j+1}^n} + \frac{\underline{X}_{j-1}^n}{q_j^n} \right) \right|, \end{aligned}$$

by the triangle inequality,

$$\leq |\underline{X}_j^n| \left| 1 - \frac{2\Delta t}{q_j^n q_{j+1}^n} \right| + \left| \frac{2\Delta t}{q_j^n q_{j+1}^n} \right| \left| \frac{\underline{X}_{j+1}^n}{q_{j+1}^n} + \frac{\underline{X}_{j-1}^n}{q_j^n} \right|,$$

if we let

$$|\underline{X}_i^n| = \max_{j\in[1,N]} |\underline{X}_j^n|,$$

then:

$$\leq |\underline{X}_i^n| \left(\left| 1 - \frac{2\Delta t}{q_j^n q_{j+1}^n} \right| + \left| \frac{2\Delta t}{q_j^n q_{j+1}^n} \right| \right).$$

Since we have $\frac{2\Delta t}{q_j^n q_{j+1}^n} > 0$, and if we have $1 - \frac{2\Delta t}{q_j^n q_{j+1}^n} \geq 0$, then:

$$= |\underline{X}_i^n| \left(1 - \frac{2\Delta t}{q_j^n q_{j+1}^n} + \frac{2\Delta t}{q_j^n q_{j+1}^n} \right),$$

$$= |\underline{X}_i^n|.$$

As this is true for $\forall j$, we have

$$\max_{j\in[1,N]} |\underline{X}_j^{n+1}| \leq \max_{j\in[1,N]} |\underline{X}_j^n|.$$

However we only have this if $1 - \frac{2\Delta t}{q_j^n q_{j+1}^n} \geq 0$, which is equivalent to $\frac{q_j^n q_{j+1}^n}{2} > \Delta t$. This is our bound on Δt.

Thus we have shown that if the statement is true for n, it is true for $n+1$, and it is also true for the base case $n = 0$, and thus is true for all n. □

It is of interest that the bound on Δt depends on n, and will get smaller with time if $q_j^n q_{j+1}^n$ gets smaller. This is why the instabilities tend to occur after some time, for example in fig. 2.

3.6.2 Semi-Implicit Scheme with $f = 0$

Theorem: Maximum Principle for the Parametric Semi Implicit Scheme with $f = 0$. *We have, for boundary conditions $\underline{X}_b$, that:*

$$\max_{j\in[1,N]} |\underline{X}_j^n| \leq \max_{j\in[1,N]} \left(|\underline{X}_j^0|, |\underline{X}_b|\right).$$

Proof. We will prove this by induction. The base case is:

$$\max_{j\in[1,N]} |\underline{X}_j^0| \leq \max_{j\in[1,N]} \left(|\underline{X}_j^0|, |\underline{X}_b|\right),$$

which is clearly true.

Now, for the inductive step, assume that the statement is true for n, and we will show it is true for $n+1$. The semi-implicit scheme is given by:

$$|\underline{X}_j^n| = \left| \underline{X}_j^{n+1}\left(1 + \frac{2\Delta t}{q_{j+1}^n q_j^n}\right) - \underline{X}_{j+1}^{n+1}\frac{2\Delta t}{q_{j+1}^n\left(q_j^n + q_{j+1}^n\right)} - \underline{X}_{j-1}^{n+1}\frac{2\Delta t}{q_j^n\left(q_j^n + q_{j+1}^n\right)} \right|$$

By the lower triangle inequality,

$$\geq \left| |\underline{X}_j^{n+1}| \left|1 + \frac{2\Delta t}{q_{j+1}^n q_j^n}\right| - \left| \underline{X}_{j+1}^{n+1}\frac{2\Delta t}{q_{j+1}^n\left(q_j^n + q_{j+1}^n\right)} + \underline{X}_{j-1}^{n+1}\frac{2\Delta t}{q_j^n\left(q_j^n + q_{j+1}^n\right)} \right| \right|,$$

and then by the triangle inequality,

$$\geq \left| |\underline{X}_j^{n+1}| \left|1 + \frac{2\Delta t}{q_{j+1}^n q_j^n}\right| - \left| \underline{X}_{j+1}^{n+1}\frac{2\Delta t}{q_{j+1}^n\left(q_j^n + q_{j+1}^n\right)}\right| - \left| \underline{X}_{j-1}^{n+1}\frac{2\Delta t}{q_j^n\left(q_j^n + q_{j+1}^n\right)} \right| \right|.$$

If we let

$$\left|\underline{X}_i^{n+1}\right| = \max_{j\in[1,N]} \left|\underline{X}_j^{n+1}\right|,$$

then:

$$\geq \left| |\underline{X}_i^{n+1}| \left|1 + \frac{2\Delta t}{q_{j+1}^n q_j^n}\right| - \left| \underline{X}_i^{n+1}\frac{2\Delta t}{q_{j+1}^n\left(q_j^n + q_{j+1}^n\right)}\right| - \left| \underline{X}_i^{n+1}\frac{2\Delta t}{q_j^n\left(q_j^n + q_{j+1}^n\right)} \right| \right|.$$

Since $\frac{2\Delta t}{q_{j+1}^n + q_j^n} \geq 0$, we are left with:

$$= |\underline{X}_i^{n+1}|$$

As this is true for $\forall j$, we have

$$\max_{j\in[1,N]} |\underline{X}_j^{n+1}| \leq \max_{j\in[1,N]} |\underline{X}_j^n|,$$

with no bounds on Δt.
Thus we have shown that if the statement is true for n, it is true for $n+1$, and it is also true for the base case $n = 0$, and thus is true for all n. □

4 The Phase Field Approach

4.1 Deriving the Allen-Cahn Equation - Calculus of Variations

We wish to derive the Allen-Cahn equation as it is an approximation to the mean curvature flow. This is all referenced in [3]. In our quest to derive the equation using calculus of variations, we start with the energy functional:

$$J(u) = \int_\Omega \left[\frac{\epsilon}{2}|\nabla u|^2 + \frac{1}{\epsilon}w(u) + C_w f(u)\right] dx, \forall \epsilon > 0,$$

here $w(u)$ is the free energy, double well or double obstacle. These are defined later on, however $w(u)$ is intended to drive negative values of u to -1, and positive values of u to 1; the $|\nabla u|^2$ term has a smoothing effect on the solution, diffusing large gradients in u, and $C_w f(u)$ is the forcing. Thus if the boundaries of Ω are given by $u = -1$, and the interior of the curve is given by $u = 1$, we can intuitively see that area covered by the interior of the curve will shrink, as u is smoothed, until $u = -1$ everywhere. This movement follows mean curvature flow.

We want to minimise $J(u)$ over some F. If we have homogenous Neumann boundary conditions, then $F = \{u \in H^1(\Omega)$ s.t. $\frac{\partial u}{\partial \underline{n}} = 0$ where $\underline{n}$ is the normal to the boundary$\}$. If we have Dirichlet boundary conditions, then $F = \{u \in H^1(\Omega)$ s.t. $u|_{\partial\Omega} = 0\}$.

Note that for some function $h \in F$, we have, for $y_0 \in F$, the direction derivative of $J(u)$ in direction h at point y_0 is:

$$\lim_{\epsilon\to 0} \frac{J(y_o + \epsilon h) - J(y_0)}{\epsilon}.$$

If we let $V(\epsilon) = J(y_0 + \epsilon h)$, so $V(0) = J(y_0)$, which gives:

$$V'(0) = \lim_{\epsilon\to 0} \frac{V(\epsilon) - V(0)}{\epsilon} = \lim_{\epsilon\to 0} \frac{J(y_o + \epsilon h) - J(y_0)}{\epsilon}.$$

So if J has a minimum at y_0, then $V'(0) = 0 (= J'(y_0))$. This means that $J(y_0) \leq J(y)$. If $y = y_0 + \epsilon h$:

$$J(y_0) \leq J(y_0 + \epsilon h), \forall \epsilon.$$

Or equivalently:

$$V(0) \leq V(\epsilon), \forall \epsilon,$$

which implies $\epsilon = 0$ is the minimum of V, ie $V'(0) = 0$

4.1.1 Calculus of Variations

Let $v \in F$, and $h \in \Re$, then:

$$V(h) = E(u+hv) = \int_\Omega \left[\frac{\epsilon}{2} |\nabla(u+hv)|^2 + \frac{1}{\epsilon} w(u+hv) + C_w f(u+hv) \right] dx.$$

And so:

$$\begin{aligned} V'(0) = \lim_{h\to 0} \quad & \frac{1}{h} \int_\Omega \left[\frac{\epsilon}{2} |\nabla(u+hv)|^2 + \frac{1}{\epsilon} w(u+hv) \right. \\ & \left. - \frac{\epsilon}{2} |\nabla u|^2 - \frac{1}{\epsilon} w(u) + C_w f(u+hv-u) \right] dx, \\ = \lim_{h\to 0} \quad & \frac{1}{h} \int_\Omega \left[\frac{\epsilon}{2} |\nabla u|^2 + \frac{h^2 \epsilon}{2} |\nabla v|^2 + h\epsilon \nabla v \nabla u \right. \\ & \left. - \frac{\epsilon}{2} |\nabla u|^2 + \frac{1}{\epsilon} w(u+hv) - \frac{1}{\epsilon} w(u) + C_w f h v \right] dx. \end{aligned}$$

If this is sufficiently smooth, then:

$$\begin{aligned} V'(0) &= \int_\Omega \left[\lim_{h\to 0} \frac{1}{h} \left(\frac{h^2 \epsilon}{2} |\nabla v|^2 + h\epsilon \nabla v \nabla u \right) + \frac{1}{\epsilon} w'(u) v + C_w f v \right] dx, \\ &= \int_\Omega \left[\epsilon \nabla v \nabla u + \frac{1}{\epsilon} w'(u) v + C_w f v \right] dx. \end{aligned}$$

Using Greens Theorem on the first term in the integral gives:

$$V'(0) = \int_\Omega -\epsilon \triangle uv + \frac{1}{\epsilon} w'(u) v dx + \int_{\partial\Omega} \frac{\partial u}{\partial \underline{n}} v dx.$$

For homogenous Neumann boundary conditions we have $\frac{\partial u}{\partial \underline{n}} = 0$, and for Dirichlet boundary conditions we have $v|_{\partial\Omega} = 0$. Note also that we want $V'(0) = 0$, so:

$$0 = \int_\Omega \left[-\epsilon \triangle u + \frac{1}{\epsilon} w'(u) + C_w f \right] v dx.$$

By the Fundamental Theorem of Calculus of Variations, we have:

$$0 = -\epsilon \triangle u + \frac{1}{\epsilon} w'(u) + C_w f. \tag{8}$$

The Allen-Cahn equation is obtained by adding ϵu_t to the left hand side, such that the steady stable solution of the Allen-Cahn equation satisfies (8).

$$\epsilon u_t = \epsilon \triangle u - \frac{1}{\epsilon} w'(u) + C_w f.$$

4.2 The Allen-Cahn Equation

As we previously calculated in the calculus of variations section, the Allen-Cahn equation with free energy $w(u)$, and forcing term f, is:

$$u_t - \triangle u + \frac{1}{\epsilon^2} w'(u) + \frac{C_w}{\epsilon} f = 0,$$

where $C_w = \int_{-1}^{1} \frac{1}{2}\sqrt{2w(r)}dr$, taken from [3], is a constant that depends on $w(u)$. Here we either have the Dirichlet boundary condition (setting $u = -1$ on the boundary), or homogenous Neumann boundary conditions (where $\frac{\partial u}{\partial \underline{n}} = 0$ on the boundary, and $\underline{n}$ is the normal), along with the initial condition $u(x, 0) = u^0(x)$. This approach splits the region Ω into two phase spaces, $u = 1$ and $u = -1$, with a phase transition region connecting the two phase spaces. The curve we are evolving with mean curvature flow is contained with in the transition region.

4.3 Double Well Free Energy

We define the double well free energy to be $w(u) = \frac{1}{4}\left(u^2 - 1\right)^2$, shown in fig. 8. This means $C_w = \int_{-1}^{1} \frac{1}{2}\sqrt{2w(r)}dr = \frac{\sqrt{2}}{3}$.

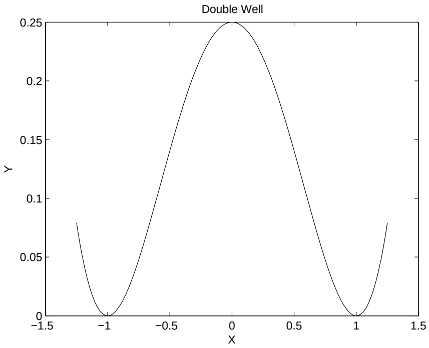

Figure 8: This is the graph of the double well free energy, $w(u) = \frac{1}{4}\left(u^2 - 1\right)^2$.

Given that $w'(u) = u^3 - u$, approximating u_t by $u_t = \frac{u^{n+1} - u^n}{\Delta t}$, and letting θ_1 and $\theta_2 = n + 1$ or n, we have:

$$\frac{u^{n+1} - u^n}{\Delta t} - \triangle u^{\theta_1} + \frac{1}{\epsilon^2}\left(u^{n^3} - u^{\theta_2}\right) + \frac{C_w}{\epsilon} f = 0.$$

First we will test with $\rho \in V = \{u \in H^1 \text{ s.t. } u|_{\partial\Omega} = 0\}$ if Dirichlet boundary conditions, and $\rho \in V = H^1$ if homogenous Neumann boundary conditions, and integrate over Ω.

$$\begin{aligned}\int_\Omega u^{n+1}\rho dx &= \int_\Omega u^n\rho dx + \Delta t\int_\Omega \triangle u^{\theta_1}\rho dx - \frac{\Delta t}{\epsilon^2}\int_\Omega (u^n)^3\rho dx \\ &+\frac{\Delta t}{\epsilon^2}\int_\Omega u^{\theta_2}\rho dx - \frac{\Delta t C_w}{\epsilon}\int_\Omega f\rho dx, \forall \rho \in V.\end{aligned}$$

Integrate the second term by parts, and cancel the boundary term using either the Neumann or Dirichlet boundary conditions.

$$\begin{aligned}\int_\Omega u^{n+1}\rho dx &= \int_\Omega u^n\rho dx - \Delta t\int_\Omega \nabla u^{\theta_1}\nabla\rho dx - \frac{\Delta t}{\epsilon^2}\int_\Omega (u^n)^3\rho dx \\ &+\frac{\Delta t}{\epsilon^2}\int_\Omega u^{\theta_2}\rho dx - \frac{\Delta t C_w}{\epsilon}\int_\Omega f\rho dx, \forall \rho \in V.\end{aligned}$$

4.3.1 Homogenous Neumann Boundary Conditions

Now consider a finite element triangulation of Ω, made up of an $N \times N$ grid of points, as shown in fig. 9.

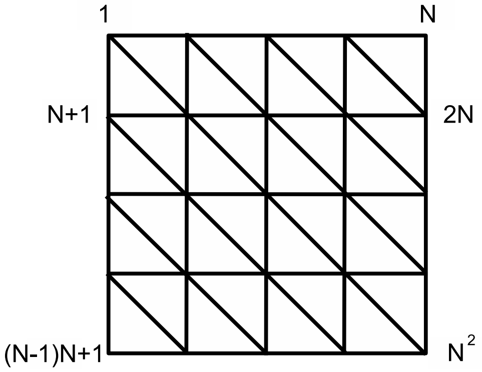

Figure 9: Finite element triangulation of Ω

Discretize into the space $V_h = \{$ Functions that are in $H^1(\Omega)$ and piecewise linear over each triangle $\}$.

$$\begin{aligned}\int_\Omega u_h^{n+1}\rho_h dx &= \int_\Omega u_h^n\rho_h dx - \Delta t\int_\Omega \nabla u_h^{\theta_1}\nabla\rho_h dx - \frac{\Delta t}{\epsilon^2}\int_\Omega (u_h^n)^3\rho_h dx \\ &+\frac{\Delta t}{\epsilon^2}\int_\Omega u_h^{\theta_2}\rho_h dx - \frac{\Delta t C_w}{\epsilon}\int_\Omega f_h\rho_h dx, \forall \rho_h \in V_h.\end{aligned}$$

Given the basis for V_h is given by $\{\phi_j\}_{j=1}^{N^2}$, we have $u_h^{\theta_k} = \sum_{i=1}^{N^2} u_i^{\theta_k}(t)\phi_i(x)$ and $\rho_h = \phi_j$ for $j = 1, \ldots, N^2$. Substituting this in:

$$\begin{aligned}\int_\Omega \sum_{i=1}^{N^2} u_i^{n+1}\phi_i\phi_j dx &= \int_\Omega \sum_{i=1}^{N^2} u_i^n \phi_i\phi_j dx - \Delta t \int_\Omega \sum_{i=1}^{N^2} u_i^{\theta_1}\nabla\phi_i\nabla\phi_j dx - \frac{\Delta t}{\epsilon^2}\int_\Omega \sum_{i=1}^{N^2}(u_i^3)^n\phi_i\phi_j dx \\ &\quad + \frac{\Delta t}{\epsilon^2}\int_\Omega \sum_{i=1}^{N^2} u_i^{\theta_2}\phi_i\phi_j dx - \frac{\Delta t C_w}{\epsilon}\int_\Omega f_h\phi_j dx, \forall j \in [1, \ldots, N^2].\end{aligned}$$

By mass lumping (5) we have:

$$M\underline{U}^{n+1} = M\underline{U}^n - \Delta t A\underline{U}^{\theta_1} - \frac{\Delta t}{\epsilon^2}M(\underline{U}^3)^n + \frac{\Delta t}{\epsilon^2}M\underline{U}^{\theta_2} - \frac{\Delta t C_w}{\epsilon}M\underline{F},$$

where M is the mass matrix, which is diagonal and has entries $\left(\int_\Omega \phi_j\right)_{j=1}^{N^2}$. A is the stiffness matrix, given by $A := \left(\int_\Omega \nabla\phi_i\nabla\phi_j\right)_{i,j=1}^{N^2}$. $\underline{U}^\theta$ is the vector defined as $\underline{U}^{\theta_k} := \left(u_i^{\theta_k}\right)_{i=1}^{N^2}$. Finally $\underline{F}$ is the vector defined as $\underline{F} := (f(x_i))_{i=1}^{N^2}$.

Taking $\theta_1 = n+1$ and $\theta_2 = n$ gives the semi-implicit scheme:

$$(M + \Delta t A)\,\underline{U}^{n+1} = \left(1 + \frac{\Delta t}{\epsilon^2}\right)M\underline{U}^n - \frac{\Delta t}{\epsilon^2}M(\underline{U}^3)^n - \frac{\Delta t C_w}{\epsilon}M\underline{F}.$$

4.3.2 Dirichlet Boundary Conditions

Now consider a finite element triangulation of Ω, made up of an $N \times N$ grid of points, as shown in fig. 9.

Discretize into the space $V_h = \{u_h \in H^1(\Omega)$ s.t u_h is piecewise linear over each triangle, and $u_h|_{\partial\Omega} = -1\}$.

$$\begin{aligned}\int_\Omega u_h^{n+1}\rho_h dx &= \int_\Omega u_h^n\rho_h dx - \Delta t\int_\Omega \nabla u_h^{\theta_1}\nabla\rho_h dx - \frac{\Delta t}{\epsilon^2}\int_\Omega (u_h^n)^3\rho_h dx \\ &\quad + \frac{\Delta t}{\epsilon^2}\int_\Omega u_h^{\theta_2}\rho_h dx - \frac{\Delta t C_w}{\epsilon}\int_\Omega f_h\rho_h dx, \forall\rho_h \in V_h.\end{aligned}$$

If we let the set $Y :=$ { nodes on $\partial\Omega$}, and the set $X :=$ { nodes not on $\partial\Omega$}, then the basis for V_h is given by $\{\phi_j\}_{j\in X}$, since $\rho = 0$ on $\partial\Omega$. We have $u_h^\theta = \sum_{i=1}^{N^2} u_i^\theta(t)\phi_i(x)$, $f_h = \sum_{i=1}^{N^2} f_i(x)\phi_i(x)$ and $\rho_h = \phi_j$ for $j \in X$. Substituting this in, and considering mass lumping (5), we can derive the semi implicit scheme, with $\theta_1 = n+1$ and $\theta_2 = n$, as:

$$\sum_{i\in X} u_i^{n+1}\left(\int_\Omega \phi_j dx + \Delta t \int_\Omega \nabla\phi_j \nabla\phi_i dx\right) =$$
$$\left(1+\frac{\Delta t}{\epsilon^2}\right)\sum_{i\in X} u_i^n \int_\Omega \phi_j dx - \frac{\Delta t}{\epsilon^2}\sum_{i\in X}(u_i^n)^3 \int_\Omega \phi_j dx + \frac{\Delta t C_w}{\epsilon}\int_\Omega \sum_{i\in X} f_i \phi_j dx$$
$$-\sum_{i\in Y} u_i^{n+1}\left(\int_\Omega \phi_j dx + \Delta t \int_\Omega \nabla\phi_j \nabla\phi_i dx\right), \forall j \in X.$$

As before take M to be the mass matrix, which is diagonal and has entries $\left(\int_\Omega \phi_j\right)_{j\in X}$. A is the stiffness matrix, given by $A := \left(\int_\Omega \nabla\phi_i \nabla\phi_j\right)_{i,j\in X}$. $\underline{U}^\theta$ is the vector defined as $\underline{U}^\theta := \left(u_i^\theta\right)_{i\in X}$. We will also take the matrix $B := \left(\int_\Omega \phi_j dx + \sum_{i\in Y}\Delta t \int_\Omega \nabla\phi_j\nabla\phi_i dx\right)_{j\in X} = \Delta t$ if $i \in Y\backslash\{1, N, (N-1)N+1, N^2\}$ and 0 otherwise. Finally $\underline{F}$ is the vector defined as $\underline{F} := (f_i)_{i\in X}$. Then we have the semi-implicit scheme:

$$(M+\Delta t A)\,\underline{U}^{n+1} = \left(1+\frac{\Delta t}{\epsilon^2}\right)M\underline{U}^n - \frac{\Delta t}{\epsilon^2}M(\underline{U}^3)^n - \frac{\Delta t C_w}{\epsilon}M\underline{F} + B.$$

4.4 Double Obstacle Free Energy

We now consider the double obstacle free energy. The reason for this is that the values of u will be unable to leave the range $[-1,1]$. This gives rise to a "sharp" diffuse interface region, where $|u| < 1$, as opposed to the double well potential, where the interface region is not well defined. The energy is given by:

$$w(u) = \begin{cases} \frac{1}{2}(1-u^2) & \text{if } -1 \le \phi \le 1 \\ \infty & \text{otherwise} \end{cases},$$

which looks like fig. 10. Thus we now have $C_w = \frac{\pi}{4}$. Note that the sub derivative [3] of the free energy is:

$$w'(u) = \begin{cases} -u & \text{if } |u| < 1 \\ [0,\infty) & \text{if } u = 1 \\ (-\infty, 0] & \text{if } u = -1 \end{cases}.$$

First we must note that if $u = -1$, then $|\rho| \le 1$ and we have $(\rho - u) \ge 0$. Hence:

$$\int_\Omega (-\infty, 0](\rho - u)dx \le 0 \text{ for } u = -1.$$

This means which ever value you take from $(-\infty, 0]$, we still have $(-\infty, 0](\rho - u) \le 0$, and thus the integral is also ≤ 0.

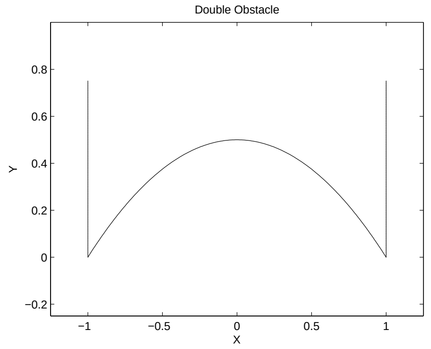

Figure 10: This is the graph of the double obstacle free energy.

We also note also that:

$$\int_\Omega [0,\infty)(\rho-u)dx \leq 0 \text{ for } u=1.$$

By similar logic to the above, $(\rho-u)\leq 0$, implying that $[0,\infty)(\rho-u)\geq 0$, and hence this integral is also ≤ 0.

Now to discretize our Allen-Cahn equation; first we will test with $\rho-u$, where $\rho \in V = \{u \in H^1(\Omega) \text{ s.t. } u|_{\partial\Omega} = -1\}$ if Dirichlet boundary conditions, and $\rho \in V = H^1(\Omega)$ if homogenous Neumann boundary conditions. Then integrate over Ω.

$$\begin{aligned} & \int_\Omega u_t(\rho-u)dx - \int_\Omega \triangle u(\rho-u)dx \\ & +\frac{1}{\epsilon^2}\int_\Omega -\underbrace{u(\rho-u)}_{|u|<1} + \underbrace{(-\infty,1](\rho-u)}_{u=-1} + \underbrace{[-1,\infty)(\rho-u)}_{u=1}\,dx \\ = & \frac{C_w}{\epsilon}\int_\Omega f(\rho-u)dx, \end{aligned}$$

which, by the previously shown facts, is equivalent to:

$$\int_\Omega u_t(\rho-u)dx - \int_\Omega \triangle u(\rho-u)dx + \frac{1}{\epsilon^2}\int_\Omega u(\rho-u)dx \geq \frac{C_w}{\epsilon}\int_\Omega f(\rho-u)dx.$$

Integrate the second term by parts, and cancel the boundary term using either the Neumann or Dirichlet boundary condition, thus:

$$\int_\Omega u_t(\rho-u)dx + \int_\Omega \nabla u \nabla(\rho-u)dx + \frac{1}{\epsilon^2}\int_\Omega u(\rho-u)dx \geq \frac{C_w}{\epsilon}\int_\Omega f(\rho-u)dx.$$

4.4.1 Homogenous Neumann Boundary Conditions

Now consider a finite element triangulation of Ω, made up of an $N \times N$ grid of points, as shown in fig. 9.

Discretize into the space $V_h = \{$ Functions that are piecewise linear over each triangle $\}$.

$$\int_\Omega u_{h_t}(\rho_h - u_h)dx + \int_\Omega \nabla u_h \nabla(\rho_h - u_h)dx + \frac{1}{\epsilon^2}\int_\Omega u_h(\rho_h - u_h)dx \geq \frac{C_w}{\epsilon}\int_\Omega f_h(\rho_h - u_h)dx, \forall \rho_h \in V_h.$$

Given the basis for V_h is given by $\{\phi_j\}_{j=1}^{N^2}$, we have $u_h = \sum_{i=1}^{N^2} u_i(t)\phi_i(x)$, $f_h = \sum_{i=1}^{N^2} f_i(x)\phi_i(x)$ and $\rho_h = u_h + \phi_j$ for $j = 1 \ldots N^2$. Substituting this in:

$$\begin{aligned}\sum_{i=1}^{N^2} u_{i_t}\int_\Omega \phi_i\phi_j dx \quad &+ \quad \sum_{i=1}^{N^2}\int_\Omega \nabla\phi_i\nabla\phi_j dx \\ &+ \quad \frac{1}{\epsilon^2}\sum_{i=1}^{N^2} u_i \int_\Omega \phi_i\phi_j dx \geq \frac{C_w}{\epsilon}\int_\Omega \sum_{i=1}^{N^2} f_i\phi_i\phi_j dx, \forall j \in [1, \ldots, N^2].\end{aligned}$$

As $u_t = \frac{u^{n+\frac{1}{2}} - u^n}{\Delta t}$, and considering mass lumping (5), we have the first part of the semi implicit scheme:

$$\left(M + \Delta t A - \frac{\Delta t}{\epsilon^2}M\right)\underline{U}^{n+\frac{1}{2}} = M\left(\underline{U}^n + \frac{C_w \Delta t}{\epsilon}\underline{F}\right),$$

where M is the mass matrix, which is diagonal and has entries $\left(\int_\Omega \phi_j\right)_{j=1}^{N^2}$. A is the stiffness matrix, given by $A := \left(\int_\Omega \nabla\phi_i\nabla\phi_j\right)_{i,j=1}^{N^2}$, $\underline{U}^n$ is the vector defined as $\underline{U}^n := (u_i^n)_{i=1}^{N^2}$. Finally $\underline{F}$ is the vector defined as $\underline{F} := (f(x_i))_{i=1}^{N^2}$.

We must then project $U^{n+\frac{1}{2}}$ to give U^{n+1} using the projection $\mathcal{P}u^{n+\frac{1}{2}} \to u^{n+1}$ such that:

$$\mathcal{P}u = \begin{cases} 1 & \text{if } u > 1 \\ -1 & \text{if } u < -1 \\ u & \text{if } |u| \leq 1 \end{cases}.$$

4.4.2 Dirichlet Boundary Conditions

Now consider a finite element triangulation of Ω, made up of an $N \times N$ grid of points, as shown in fig. 9.

Discretize into the space $V_h = \{u_h \in H^1(\Omega)$ s.t u_h is piecewise linear over each triangle, and $u_h|_{\partial\Omega} = -1\}$.

$$\begin{aligned}\int_\Omega u_{h_t}(\rho_h - u_h)dx \quad &+ \quad \int_\Omega \nabla u_h \nabla(\rho_h - u_h)dx \\ &+ \quad \frac{1}{\epsilon^2}\int_\Omega u_h(\rho_h - u_h)dx \geq \frac{C_w}{\epsilon}\int_\Omega f_h(\rho_h - u_h)dx, \forall \rho_h \in V_h.\end{aligned}$$

If we let the set $Y := \{$ nodes on $\partial\Omega\}$, and the set $X := \{$ nodes not on $\partial\Omega\}$, then the basis for V_h is given by $\{\phi_j\}_{j\in X}$. We have $u_h = \sum_{i=1}^{N^2} u_i(t)\phi_i(x)$, $f_h = \sum_{i=1}^{N^2} f_i(x)\phi_i(x)$ and $\rho_h = \phi_j$ for $j \in X$. Substituting this in, along with $u_t = \frac{u^{n+\frac{1}{2}} - u^n}{\Delta t}$, and considering mass lumping (5), we can derive the semi implicit scheme:

$$\left(\left(1 - \frac{\Delta t}{\epsilon^2}\right) M + \Delta t A\right) U^{n+\frac{1}{2}} = M\left(U^n + \frac{C_w \Delta t}{\epsilon} F\right) - BU^{n+1}.$$

We take M is the mass matrix, which is diagonal and has entries $\left(\int_\Omega \phi_j\right)_{j\in X}$. A is the stiffness matrix, given by $A := \left(\int_\Omega \nabla\phi_i \nabla\phi_j\right)_{i,j\in X}$. $\underline{U}^\theta$ is the vector defined as $\underline{U}^\theta := \left(u_i^\theta\right)_{i\in X}$. We will also take the matrix B, defined as:

$$B := \left(\int_\Omega \phi_j dx + \sum_{i\in Y} \Delta t \int_\Omega \nabla\phi_j \nabla\phi_i dx\right)_{j\in X} = \begin{cases} \Delta t & \text{if } i \in Y \backslash \{1, N, (N-1)N+1, N^2\} \\ 0 & \text{if otherwise} \end{cases}.$$

Finally $\underline{F}$ is the vector defined as $\underline{F} := (f(x_i))_{i\in X}$. Then we have:

Finally we must project $U^{n+\frac{1}{2}}$ to give U^{n+1} using the projection $\mathcal{P}U^{n+\frac{1}{2}} \rightarrow U^{n+1}$ such that:

$$\mathcal{P}u = \begin{cases} 1 & \text{if } u > 1 \\ -1 & \text{if } u < -1 \\ u & \text{if } |u| \leq 1 \end{cases}.$$

5 Solving a Linear System with Gauss-Seidel

5.1 The Gauss-Seidel Algorithm

The Gauss-Seidel algorithm is an iterative method that approximates the solution of the linear system $Ax = b$. Here A is an $N \times N$ matrix, where each element is given by a_{ij}; x and b are vectors of length N, whose elements are given by x_i and b_i, respectively. Given and initial guess x_0 (which is a vector of length N with elements given by X_{0_i}), a maximum number of loops, *loops*, and a minimum tolerance, *tol*. The Gauss-Seidel algorithm was used for the double well free energy. The double obstacle free energy required a projection $\mathcal{P}U$. Thus the Projected Gauss-Seidel algorithm was used.

Algorithm 2 Gauss-Seidel

```
This algorithm solves the linear system $A\underline{x} = \underline{b}$ with initial guess $\underline{x}_0$. Algorithm taken from [1].
for $k = 1$ to $loops$ do
    for $i = 1 \dots N$ do
        Set: $x_i = \frac{-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{N} a_{ij}x_{0_j}}{a_{ii}}$
    end for
    if $\|\underline{x} - \underline{x}_0\| < tol$ then
        Return $\underline{x}$
    end if
end for
```

Algorithm 3 Projected Gauss-Seidel

```
This algorithm solves the linear system $A\underline{x} = \underline{b}$ with initial guess $\underline{x}_0$, using a projection $\mathcal{P}\underline{x_i}$ within each iteration. Algorithm taken from [1] and then modified.
for $k = 1$ to $loops$ do
    for $i = 1 \dots N$ do
        Set: $x_i = \frac{-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{N} a_{ij}x_{0_j}}{a_{ii}}$
        Project $x_i = \mathcal{P}x_i$
    end for
    if $\|\underline{x} - \underline{x}_0\| < tol$ then
        Return $\underline{x}$
    end if
end for
```

5.2 How the Guass-Seidel Algorithm is used

In many of the finite element methods described previously, in particular in the phase field approach, we are inverting either $(M + \Delta t A)$, or $\big(M + \Delta t A - \frac{\Delta t}{\epsilon^2} M\big)$, both of which are either $N^2 \times N^2$ or $(N-2)^2 \times (N-2)^2$ matrices (depending of the boundary conditions). This means that if we directly store them, we will be using an extremely large amount of memory to store the matrix, most of whose elements are zero.

So instead of storing them all, we use the iterative method defined in the above algorithms, which only requires knowledge of the non zero values of a_{ij}, thus allowing us to use a much larger value of N.

6 Numerical Results for the Phase Field Approach

In this section we present some numerical results for the phase field approach.

6.1 The Results

The phase field approach is significantly more powerful than the parametric approach. It is able to deal with a topology change (discussed later), does not require parametrization, and has no problem when the curve gets extremely small. The program takes parameters of N, the number of nodes, h, the distance between each node, Δt, the time step, and ϵ, the parameter in the Allen-Cahn equation. The initial values of u, and the forcing f are also initialised at the start of the program. In all but one of our results we use the double obstacle potential energy.

Fig. 11 shows a circle evolving over over time with double obstacle free energy, similar to fig. 3. Note how initially there is a very stark contrast between the inside and outside of the curve. This is due to the initial condition setting u to be either -1 or 1. As the model progresses, the transition segment smooths, and then begins to shrink to a point. Compare fig. 11, the double obstacle free energy approach, with fig. 12, the same initial conditions and program parameters, but with double well free energy. Note how the phase transition region is thicker with the double well scheme, and thinner with the double obstacle scheme.

Fig. 13 shows a circle evolving over over time, with the same forcing as fig. 5. The evolution between the two curves is very similar. This shows that the Allen-Cahn equation is a good way of modelling the evolution of the curve.

6.2 Topology Change

When the solution generated by the parametric approach self intersected, the solution would stop behaving as we wished. The phase field approach, on the other hand, behaves exactly as required when the line intersects.

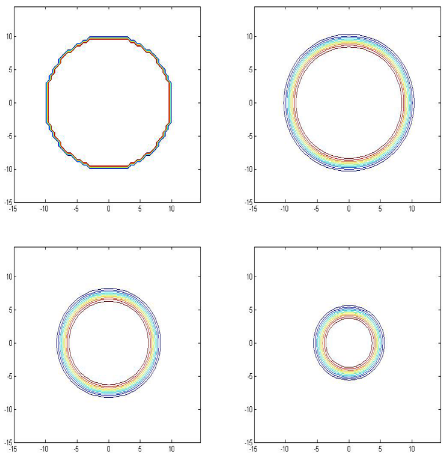

Figure 11: This is the double obstacle dirichlet formulation of the Allen-Carhn equation with no forcing. The curve evolves from left to right and top to bottom.

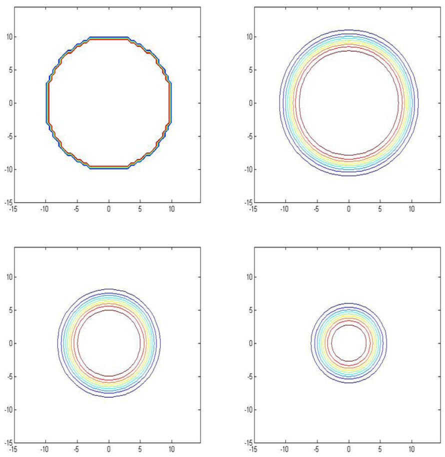

Figure 12: This is the double well dirichlet formulation of the Allen-Carhn equation with no forcing. The curve evolves from left to right and top to bottom.

Using the same forcing term f as for the parametric test results in fig. 7, the results are found in fig. 14. Here we can see that as the lines draw closer together, they split into two separate systems, which then evolve independently.

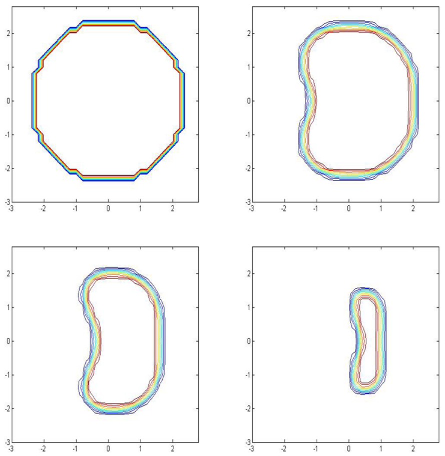

Figure 13: This is the double obstacle dirichlet formulation of the Allen-Carhn equation, with the same forcing as fig. 5. The curve evolves from left to right and top to bottom.

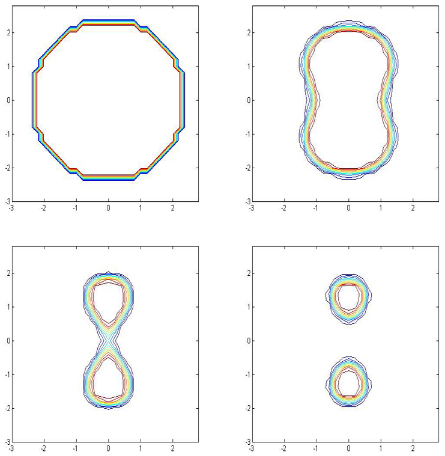

Figure 14: This is a double obstacle free energy scheme. Top left: The initial starting shape (a circle). Top right: The forcing term begins to push the left and right of the shape together. Bottom left: The curve splits into two separate shapes, creating a topology change that cannot happen using the parametric approach Bottom right: The two shapes evolve independently.

6.3 The Speed Test

As with the parametric approach, we conduct a speed test, with the radius ranging from 0.75 to 0.25. To extract an estimate of the radius from the smooth phase transition, the mean radius of all u_j such that $-0.5 \leq u_j \leq 0.5$ was found. The results can be seen in the following Table 2.

Table 2: The error in the speed test for the phase field approach.

Approximate Radius	Error
0.75	0.0007
0.5	0.0001
0.25	0.001

7 Conclusion

Two numerical approaches to approximate the evolution of curves in $\Re^2$, under the velocity $\underline{V} = -\kappa\underline{\hat{n}} + f\underline{\hat{n}}$, have been presented. We first used a parametric approach to model the curve, and from there approximated the evolution with a finite element method. The second method split the plane into two phases, one of $u = -1$, and one of $u = 1$. Between these there was smooth transition layer, which approximates the curve. This then evolved under the influence of the Allen-Cahn equation, which was approximated numerically with a finite element method.

The parametric approach had the advantage of speed. This is due to modelling only the curve, which was parametrised on an interval $[0, 2\pi)$, meaning that we had only N nodes (compared to the phase fields N^2 nodes). The shortfalls of this approach were, however, large. The biggest problem was that the curve could not deal with a topology change. When opposite edges of the curve touch each other, they pass through.

The phase field approach was much slower than the parametric approach. This is because the whole of Ω was simulated, which contained N^2 nodes. Contrary to the parametric approach, the phase field approach was able to self intersect. When opposite edges of the curve touch each other, the curve would split into two separate curves, which would then evolve independently.

For both of these approaches, we derived both an explicit, and a semi-implicit finite element method. Using the maximum principle, it was shown that the explicit approach had a bound on the time interval, while the semi-implicit had no such bound.

So, in conclusion, the main advantage of the parametric approach was speed. The phase field approach was slower, but able to cope with more complicated situations, for example topology changes.

References

[1] R. L. Burden & J. D. Faires, (2001) Numerical analysis, *Brooks Cole.*

[2] B. Flannery, W. Press, S. Teukolsky & W. Vetterling, (1992), Numerical Recipes in C, *Cambridge University Press*, ISBN: 0-521-43108-5.

[3] K. Deckelnick, G. Dziuk, Gerhard and C.M. Elliott, (2005), Computation of geometric partial differential equations and mean curvature flow, *Acta Numerica*, **14**, 139–232.