

BACHELOR THESIS

---

**Numerical methods for the implementation  
of the Cahn-Hilliard equation in one  
dimension and a dynamic boundary  
condition in two dimensions**

---

*Author:*  
Zimo SIBBING

*Supervisors:*  
Prof.dr.ir. C.R. KLEIJN  
Dr.ir. F.J. VERMOLEN  
M.Sc. H.G. PIMPALGAONKAR

28/01/2015

Delft University of Technology  
Faculty of Applied Sciences  
Department of Chemical Engineering  
Transport Phenomena group



# Abstract

This project can be divided into two parts. The goal of the first part is to numerically implement the Cahn-Hilliard equation in one dimension both explicitly and implicitly. This will be done using Matlab. The goal of the second part is to validate the coupled Cahn-Hilliard-Navier-Stokes equation and the dynamic boundary condition for moving contact lines of (Carlson et al., 2011, p. 9) by considering a two-dimensional spreading droplet case. This will be done using the CFD software OpenFOAM.

In Chapter 1, the theory of positive and negative diffusion, including the normal diffusion equation and the Cahn-Hilliard equation, are discussed. Some background is given regarding the thermodynamics of the Cahn-Hilliard equation and its steady-state solution. After that, the theory of the coupled Cahn-Hilliard-Navier-Stokes equation, the dynamic boundary condition for moving contact lines and the case which is implemented in OpenFOAM, are discussed.

In Chapters 2 and 3, the diffusion equation and the Cahn-Hilliard equation are implemented in one dimension, using the Euler Forward scheme. In implementing the Cahn-Hilliard equation, two different discretizations are used, of which only one gives the desired results. Next, an extensive stability analysis is done, using a linearization of the Cahn-Hilliard equation as well as numerical experiments. The stability condition is increasingly severe with increasing interface width. Regarding the results of the evolving interface, a qualitative analysis is done which discusses three subjects: the deviation of the solutions with the steady-state solution, the interface width for different parameters and grid sizes and the interface overshoot, which is an unphysical appearance.

In Chapter 4, two semi-implicit methods and one implicit iterative method, are discussed. The implementations of the two semi-implicit methods, Implicit-Explicit (ImEx) and Modified Furihata, are successful and their stability conditions are better than the stability condition of the Euler Forward scheme, for most interface widths. The results regarding the evolving interface are nearly identical to the results of the Euler Forward scheme, therefore the qualitative analysis is also similar. The implicit iterative method, which involves the use of the Gâteaux derivative, has not been successfully implemented, even though two different discretizations are used. The results regarding the evolving interface are behaving in a positive diffusive way, which results in a flattening interface with time.

	Explicit	Semi-implicit	Implicit
Euler Forward	X		
Implicit-Explicit (ImEx)		X	
Modified Furihata		X	
Gâteaux derivative			X

Table 0.1: An overview of all used discretizations and whether they are explicit, semi-implicit or implicit.

In Chapter 5, the coupled Cahn-Hilliard-Navier-Stokes equation and a dynamic boundary condition for moving contact lines are used to model a spreading droplet on a flat surface. The implemented model is validated using different cases in which the steady-state contact angle and the friction factor of the surface varies. Next, parametric studies are done regarding the interface width, the surface tension and the ratio of the surface tension and the friction factor. The conclusions are that the modeled system differs too much from the system in literature to make an absolute comparison but, qualitatively, the model behaves as expected.

## List of symbols

Symbol	Description	Unit
$a_{ij}$	Matrix coefficient	(-)
$A, B_1, B_2, K$	Matrices	(-)
$c(x, t)$	Concentration	(mol m <sup>-3</sup> )
$c_{steady}$	Steady-state concentration	(mol m <sup>-3</sup> )
$c_i^n$	Spatially and temporally discretized concentration	(-)
$c_p^n$	Discretized concentration in iteration $p$	(-)
$C$	Thermal diffusivity	(m <sup>2</sup> s <sup>-1</sup> )
$D$	Diffusivity	(m <sup>2</sup> s <sup>-1</sup> )
$D_w$	Friction factor	(kg m <sup>-1</sup> s <sup>-1</sup> )
$f$	OpenFOAM boundary condition constant	(-)
$F$	Scaled Gibbs free energy	(-)
$F_{liquid-gas}$	Force along liquid-gas surface	(N)
$F_{liquid-solid}$	Force along liquid-solid surface	(N)
$F_s$	Force due to surface tension	(N)
$F_{solid-gas}$	Force along solid-gas surface	(N)
$g$	Gibbs free energy density	(J m <sup>-3</sup> )
$g$	Gravity constant	(m s <sup>-2</sup> )
$G$	OpenFOAM boundary condition constant	(-)
$G_k^N$	Discretized chemical potential	(-)
$h$	Enthalpy density	(J m <sup>-3</sup> )
$h$	Discretized time step size	(-)
$H$	Enthalpy	(J)
$I$	Identity matrix	(-)
$J$	Concentration diffusion flux	(mol m <sup>-1</sup> s <sup>-1</sup> )
$L$	Dimensionless domain length	(-)
$L$	Initial droplet radius	(m)
$M$	Mobility of particles	(m <sup>2</sup> s <sup>-1</sup> )
$N$	Grid size	(-)
$p$	Pressure	(N m <sup>-2</sup> )
$p, q, r$	Discretization constants	(-)
$r$	Spreading radius	(m)
$r_0$	Initial spreading radius	(m)
$R$	Dimensionless spreading radius	(-)
$s$	Entropy density	(J K <sup>-1</sup> m <sup>-3</sup> )
$s, y$	Gâteaux derivative parameter	(-)
$t$	Time	(s)
$T$	Temperature	(K)
$T_B, T_I$	OpenFOAM boundary condition parameter	(-)
$u$	Velocity of fluid	(m s <sup>-1</sup> )
$\vec{u}$	Convection velocity	(m s <sup>-1</sup> )
$u_i^j$	Discretized diffusion temperature	(-)
$U(x, t)$	Diffusion temperature	(-)
$U_k^n$	Discretized concentration	(-)
$x$	1-dimensional location	(m)

Symbol	Description	Unit
$\alpha, \beta$	OpenFOAM boundary condition constants	(-)
$\Delta S$	Entropy difference	(J K <sup>-1</sup> )
$\Delta G$	Gibbs free energy difference	(J)
$\Delta p$	Pressure difference	(N m <sup>-2</sup> )
$\Delta t$	Discretization time step size	(-)
$\Delta t_{Euler}$	Euler Forward stable time step	(-)
$\Delta t_{Exp}$	Experimentally stable time step	(-)
$\Delta t_{ImEx}$	ImEx stable time step	(-)
$\Delta t_{MF}$	Modified Furihata stable time step	(-)
$\Delta t_{Theory}$	Theoretically stable time step	(-)
$\Delta x$	Discretization grid step size	(-)
$\epsilon$	Phenomological parameter	(m)
$\theta$	Dynamic contact angle	(radial)
$\theta_s$	Steady-state contact angle	(radial)
$\kappa$	Phenomological parameter	(m <sup>2</sup> )
$\lambda$	Eigenvalue of a matrix	(-)
$\mu$	Chemical potential	(J mol <sup>-1</sup> )
$\mu_i^n$	Discretized chemical potential	(-)
$\rho$	Density	(kg m <sup>-3</sup> )
$\rho_l$	Density of droplet liquid	(kg m <sup>-3</sup> )
$\sigma$	Surface tension	(J m <sup>-2</sup> )
$\tau$	Characteristic time constant	(-)
$\tau$	Dimensionless time	(-)
$\vec{\tau}$	Stress tensor	(N m <sup>-2</sup> )
$\phi$	Concentration	(mol m <sup>-3</sup> )
$\phi_B^n, \phi_I^n$	Discretized concentration	(-)
$\psi$	Tolerance in Gâteaux derivative algorithm	(-)

# Contents

<b>Abstract</b>	<b>ii</b>
<b>List of symbols</b>	<b>iii</b>
<b>1 Theory</b>	<b>2</b>
1.1 Negative diffusion and the Cahn-Hilliard equation . . . . .	2
1.2 Thermodynamics of the Cahn-Hilliard equation . . . . .	4
1.3 The Cahn-Hilliard equation in steady-state . . . . .	4
1.4 The coupled Cahn-Hilliard-Navier-Stokes equation . . . . .	5
1.5 Dynamic boundary condition of a spreading droplet . . . . .	6
1.6 Scope of the project . . . . .	7
<b>2 Numerical methods for the diffusion equation</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 The explicit method of Euler Forward in 1D . . . . .	8
2.3 The implicit method of Euler Backward in 1D . . . . .	11
<b>3 Modeling the Cahn-Hilliard equation with the Euler Forward discretization</b>	<b>15</b>
3.1 Discretizations . . . . .	15
3.2 Boundary and initial conditions . . . . .	16
3.3 Experimental and theoretical stability conditions . . . . .	16
3.4 Results . . . . .	21
3.5 Quantitative analysis of the results . . . . .	23
<b>4 Implicit methods for modeling the Cahn-Hilliard equation</b>	<b>28</b>
4.1 The semi-implicit scheme of Implicit-Explicit (ImEx) . . . . .	28
4.2 The semi-implicit method of Modified Furihata . . . . .	32
4.3 The implicit iterative method involving the Gâteaux derivative . . . . .	36
<b>5 The spreading droplet</b>	<b>42</b>
5.1 Implementation in OpenFOAM . . . . .	42
5.2 Validation of the implementation . . . . .	42
5.3 Parametric study of the dynamic boundary condition . . . . .	46
<b>6 Recommendations</b>	<b>50</b>
<b>References</b>	<b>51</b>



# 1 Theory

## 1.1 Negative diffusion and the Cahn-Hilliard equation

Consider a system consisting of two fluids that are uniformly mixed. Will the fluids stay mixed or will they separate as in the case of oil and water, for example? A simple experiment might give the answer, but so does the thermodynamics of diffusion. The physics of diffusion can be divided into positive and negative diffusion. Positive diffusion takes place against the direction of the concentration gradient, and diminishes the gradient in this way, while negative diffusion takes place in the same direction. The equilibrium state for positive diffusion is a uniform mixture whereas for negative diffusion, the equilibrium state is a two-phase system separated by a fluid interface. These two cases are considered separately, where the dominant transport method is the positive diffusion in one case and the negative diffusion in the other.

In the case with the positive diffusion only, Fick's diffusion law is considered, in combination with the continuity equation. Fick's first law of diffusion is:

$$J = -D\nabla c; \quad (1.1)$$

where  $c$  is the concentration,  $D$  is the diffusivity and  $J$  is the concentration diffusion flux. The continuity equation is:

$$\frac{\partial c}{\partial t} + \nabla \cdot \vec{J} = 0; \quad (1.2)$$

The combination of the first law of diffusion and the continuity equation for fluids results in Fick's second law of diffusion:

$$\frac{\partial c}{\partial t} = D\nabla^2 c; \quad (1.3)$$

From Equation (1.3), one expects that a uniform homogenous mixture is always in steady-state: in the absence of a spacial concentration gradient, no concentration change will occur with time. However, in the case where phase separation occurs, the diffusion is against the concentration gradient, which is not in accordance with Equation (1.3). Hence, the concentration gradient is not the driving force for diffusion: another force must be at work. In the negative diffusion case, the driving force is the gradient of the chemical potential, according to (Cahn and Hilliard, 1958). A new expression for Equation (1.1) can be derived:

$$J = -M\nabla\mu; \quad (1.4)$$

where  $M$  is the mobility of particles (analogous to the diffusivity  $D$ ) and  $\mu$  is the chemical potential. Using this new driving force with the continuity equation, an alternate and more generic form of Fick's second law is found. This equation is known as the Cahn-Hilliard equation by (Cahn and Hilliard, 1958):

$$\frac{\partial c}{\partial t} = M\nabla^2\mu; \quad (1.5)$$

According to (Schroeder, 1999), the definition of the chemical potential follows from the Gibbs free energy density as:

$$\mu = \frac{\partial g}{\partial c}; \quad (1.6)$$

where  $g$  is the Gibbs free energy density and  $c$  is the concentration. The energy density of this two-phase system, which is a function of the concentration, is also derived by Cahn and Hilliard, in (Cahn and Hilliard, 1958):

$$F = \int_V [f(c) + \kappa|\nabla c|^2]dV; \quad (1.7)$$

Here,  $F$  is a scaled Gibbs free energy,  $f(c)$  is the scaled free energy density due to the contributions of both homogenous phases,  $\kappa|\nabla c|^2$  is the free energy density due to the concentration gradient at the interface, or the interface energy, and  $\kappa = \epsilon^2$  is a phenomenological parameter which scales with the interface width. The function  $f(c)$  has the form of a double well potential, according to (Lee et al., 2014, p. 218). A double well potential can be presented by Equation (1.8), where  $c$  is a scaled concentration. In Figure 1.1, this double well potential is shown.

$$f(c) = \frac{(c^2 - 1)^2}{4}; \quad (1.8)$$

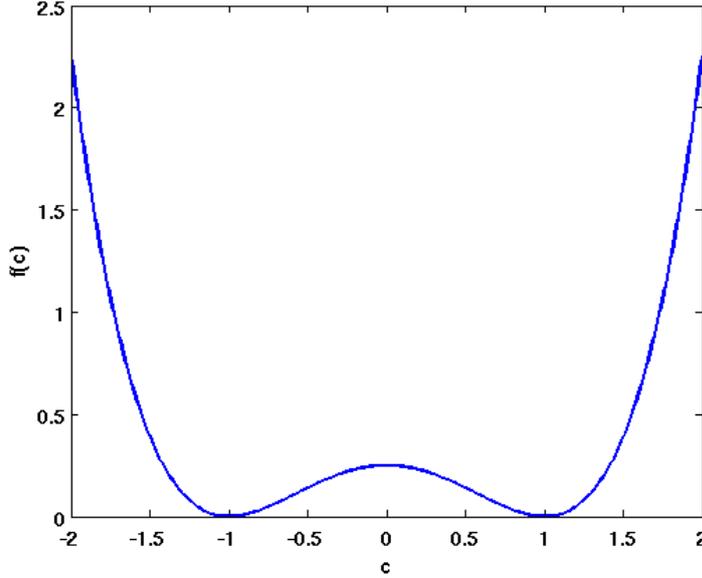


Figure 1.1: The double well potential function  $f(c) = \frac{(c^2-1)^2}{4}$ . The minima at  $c = -1$  and  $c = 1$  represent the two phases of a system with negative diffusion.

Substituting  $\kappa = \epsilon^2$  in Equation (1.7) results in the Gibbs free energy density for the two-phase system:

$$g(c) = f(c) + \epsilon^2|\nabla c|^2; \quad (1.9)$$

The chemical potential can be calculated substituting Equation (1.9) in Equation (1.6):

$$\begin{aligned} \mu &= \frac{\partial g}{\partial c} = \frac{\partial f(c)}{\partial c} + \frac{\partial}{\partial c}(\epsilon^2|\nabla c|^2) \\ &= \frac{\partial}{\partial c} \left( \frac{(c^2 - 1)^2}{4} \right) + \epsilon^2 \nabla^2 c \\ &= c^3 - c + \epsilon^2 \nabla^2 c; \end{aligned} \quad (1.10)$$

With the new expression for the chemical potential in Equation (1.10), Equation (1.5) can be rewritten as:

$$\frac{\partial c}{\partial t} = M \nabla^2 \mu = M \nabla^2 (c^3 - c + \epsilon^2 \nabla^2 c); \quad (1.11)$$

Equation (1.11) is the expression that will be referred to as the Cahn-Hilliard equation for the rest of this report. It is a fourth-order, nonlinear partial differential equation.

## 1.2 Thermodynamics of the Cahn-Hilliard equation

In Equation (1.9), it was stated that the Gibbs free energy density can be expressed as:

$$\begin{aligned} g(c) &= f(c) + \epsilon^2 |\nabla c|^2 \\ &= \frac{(c^2 - 1)^2}{4} + \epsilon^2 |\nabla c|^2; \end{aligned} \quad (1.12)$$

According to (Schroeder, 1999), the Gibbs free energy density is also defined as:

$$g = h - sT; \quad (1.13)$$

where  $h$  is the enthalpy density,  $T$  the temperature and  $s$  the entropy density. According to the second law of thermodynamics, the entropy of a closed system always increases. Applying this to Equation (1.13) implies that the Gibbs free energy always decreases in a closed system.

$$(\Delta S)_{H,T} \geq 0 \Leftrightarrow (\Delta G)_{H,T} \leq 0; \quad (1.14)$$

By the same reasoning, any system moves to an equilibrium with a maximum entropy and a minimum Gibbs free energy. Minimizing the Gibbs free energy as in Equation (1.12) yields the steady-state solution for this system.

For Equation (1.12) to be minimized, both contributions need to be minimized. The homogenous phases term,  $f(c) = \frac{(c^2-1)^2}{4}$ , is minimum at  $c = 1$  and  $c = -1$  as is shown in Figure 1.1. These minima represent the Gibbs free energy of the two homogenous phases of a mixture, separated by the fluid interface. The interface energy term,  $\epsilon^2 |\nabla c|^2$ , is minimum when  $|\nabla c|$  is, which is the case when the interface is as smooth as possible. The actual concentration of the phases in the mixture cannot exceed the concentration range of  $c \in [-1, 1]$  since that would imply a concentration of over 100% for one of the phases, which is not in accordance with mass conservation.

## 1.3 The Cahn-Hilliard equation in steady-state

In the system that is used for the implementation of the one-dimensional Cahn-Hilliard equation, Equation (1.11) converges to a steady-state solution. The initial condition of the one-dimensional system is a centered step-function, which implies that the two phases of the fluid are initially fully separated by a sharp interface. The boundary conditions are defined such that the concentration gradient and the chemical potential are zero on both boundaries. This also implies mass conservation for both phases. The derivation of the steady-state solution is outside the scope of this report. The steady-state solution, according to (Lee et al., 2014, p. 221), is:

$$c_{steady}(x) = \tanh\left(\frac{x}{\epsilon\sqrt{2}}\right); \quad (1.15)$$

The initial condition and the steady-state solution are both in Figure 1.2. The two phases of the system are indeed at  $c = -1$  and  $c = 1$  and the interface is located at  $x = 0$ . The interface is relocated at  $x = 1/2$  later (without loss of generality) but in Equation (1.15) and in Figure 1.2 it is kept at  $x = 0$ .

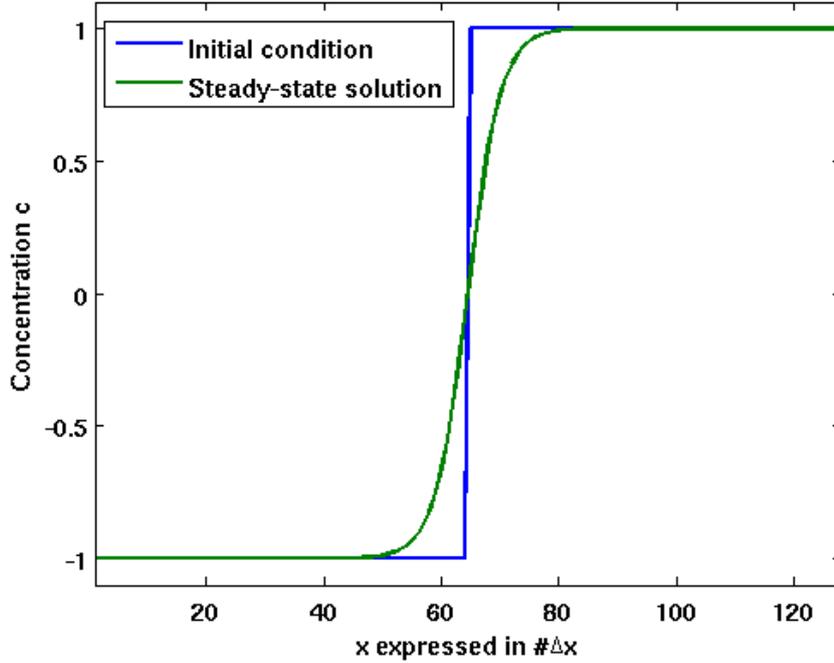


Figure 1.2: Starting with a given initial condition (for example, a discretized step function), the steady-state solution for the Cahn-Hilliard equation evolves with time.

#### 1.4 The coupled Cahn-Hilliard-Navier-Stokes equation

In Sections 1.1 - 1.3, a system in which the dominant way of mass transfer is only diffusion, was considered. In this section, a system in which not only diffusion but also convection is dominant, is considered. The system becomes dynamic, hence, apart from the Cahn-Hilliard equation, the Navier-Stokes equation must be used to describe the combination of diffusion and convection.

The continuity Equation (1.2) is rewritten, with a convection term, as:

$$\frac{\partial c}{\partial t} + \nabla \cdot (\vec{u}c) = M\nabla^2\mu; \quad (1.16)$$

where  $(\vec{u}c)$  is the convection flux,  $\vec{u}$  is the velocity field of the fluid and all other parameters are as described before. The Navier-Stokes equation, that will be used in combination with the Cahn-Hilliard equation, is:

$$\rho \left( \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} \right) = -\nabla p + \nabla \cdot \vec{\tau} + \vec{F}_s + \rho g; \quad (1.17)$$

where  $\vec{F}_s$  is the force due to surface tension, which is added because of the interfacial stresses at the fluid interface in the binary fluid. The interfacial stresses are treated here as locally active body forces. In Equation (1.17),  $\vec{\tau}$  is the stress tensor. The interfacial tension is given by (Kim, 2004):

$$\vec{F}_s = c\nabla\mu; \quad (1.18)$$

In (1.18), the chemical potential from Equation (1.10), that is also used in the Cahn-Hilliard equation, can be substituted.

## 1.5 Dynamic boundary condition of a spreading droplet

In this section, a two-dimensional dynamic convection-diffusion case, in which the coupled Cahn-Hilliard-Navier-Stokes equation is applied, is considered.

An initially spherical fluid droplet sits on a flat surface. See Figure 1.3 for this situation as modeled in OpenFOAM. With time, the droplet will spread out over the surface until it reaches a steady-state situation. The spreading radius and the dynamic contact angle are determined by the interfacial tensions along the interfaces between the fluid and the gas phase, between the gas phase and the solid surface and between the solid surface and the fluid. Young's law states, in the case of equilibrium, that:

$$F_{solid-gas} - F_{liquid-solid} = F_{liquid-gas} \cos(\theta); \quad (1.19)$$

See Figure 1.4 for the force balance between these surface tensions. In steady-state, there is a fixed spreading radius and a fixed contact angle for each combination of surface tensions between these three phases. However, under dynamic non-equilibrium conditions, the dynamic contact angle depends on the velocity of the contact line. While modeling this case, apart from the coupled equation as mentioned before, a dynamic boundary condition is used, as described in (Carlson et al., 2011, p. 9):

$$D_w \left( \frac{\partial \phi}{\partial t} + u \nabla \phi \right) = -\nabla \phi - \frac{\phi(\phi - 1) \cos(\theta)}{\epsilon \sqrt{2}}; \quad (1.20)$$

with  $\phi$  the phasefield parameter, analogous to the concentration  $c$ ,  $u$  the velocity of the fluid at the solid surface,  $\theta$  the dynamic contact angle,  $D_w$  the friction factor of the surface and  $\epsilon$  a phenomenological parameter.

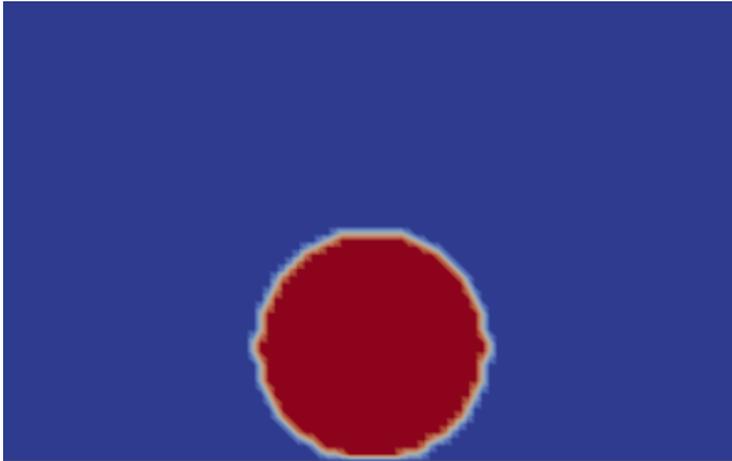


Figure 1.3: The initial condition of the droplet as implemented in OpenFOAM.

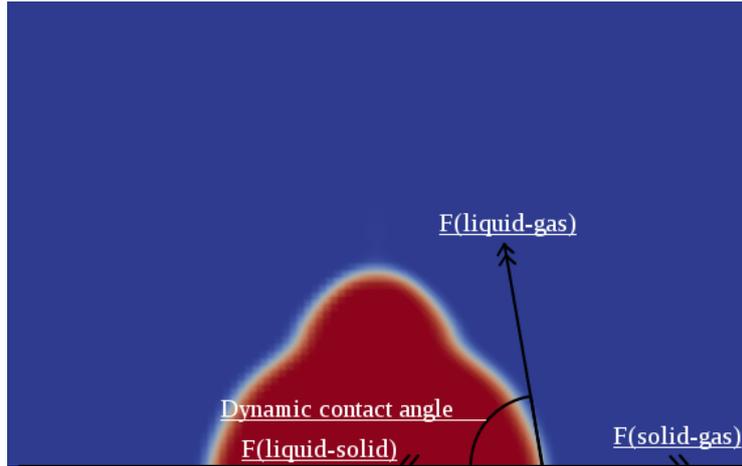


Figure 1.4: The droplet at  $t > 0$  as implemented in OpenFOAM. The dynamic contact angle  $\theta$  is determined by  $F_{solid-gas}$ ,  $F_{liquid-solid}$  and  $F_{liquid-gas}$ .

## 1.6 Scope of the project

In this project, the focus is on the implementation of the negative diffusion case mostly. However, to get familiar with the numerical methods concerning this implementation, positive diffusion is implemented first. Fick's second law of diffusion (1.3) is numerically implemented in one dimension using two temporal discretization schemes, the explicit Euler Forward scheme and the implicit Euler Backward scheme. Partially using the results of these implementations and partially applying other discretizations schemes, the Cahn-Hilliard Equation (1.11) for negative diffusion is numerically implemented after that. The used temporal discretization schemes are Euler Forward, two semi-implicit schemes that are developed on the way and a fully implicit iterative scheme involving the Gâteaux derivative. After discussing several of advantages and disadvantages of the various schemes, some conclusions are drawn regarding which one can be considered the best.

Finally, the implementation of the dynamic boundary condition in OpenFOAM and the validation of the results, are done. Some parametric studies are done and the results are analyzed.

## 2 Numerical methods for the diffusion equation

### 2.1 Introduction

The ordinary diffusion equation is a partial differential equation used for describing simple gradient heat or mass diffusion as a function of space and time. In this chapter, the heat variant of the diffusion equation is examined as the physics of the concentration variant is the same. First, the diffusion equation is numerically discretized in time with the Euler Forward discretization scheme, along with its initial and boundary conditions. Next, a stability analysis that uses the Gershgorin theorem from (Vuik et al., 2006, p. 106) is done to derive the stability criteria for the numerical discretization. Finally, the Euler Backward discretization scheme is implemented and both schemes, including stability conditions, are compared. All numerical schemes are implemented in Matlab.

The diffusion equation in one dimension involves the temperature  $U(x, t)$  and the thermal diffusivity  $C$ :

$$\frac{\partial U(x, t)}{\partial t} = C \frac{\partial^2 U(x, t)}{\partial x^2}; \quad (2.1)$$

with  $t \geq 0$  and  $x \in [0, 1]$ . From now on, the diffusivity is chosen to be  $C = 1$  for convenience. The boundary conditions are Dirichlet boundary conditions:  $U(0, t) = 1$  and  $U(1, t) = 0$ . These two conditions represent a hot wall at  $x = 0$  and a cold wall at  $x = 1$ . The initial condition is  $U(x, 0) = 0$  for  $0 < x \leq 1$ .

### 2.2 The explicit method of Euler Forward in 1D

The first method that is used is Euler Forward. This is an explicit scheme, which means that every next time step is calculated using only the information of the previous time step(s). The spatial second order derivative in Equation (2.1) is discretized using a central difference scheme. See Figure 2.1 for a schematic view of the Euler Forward scheme.

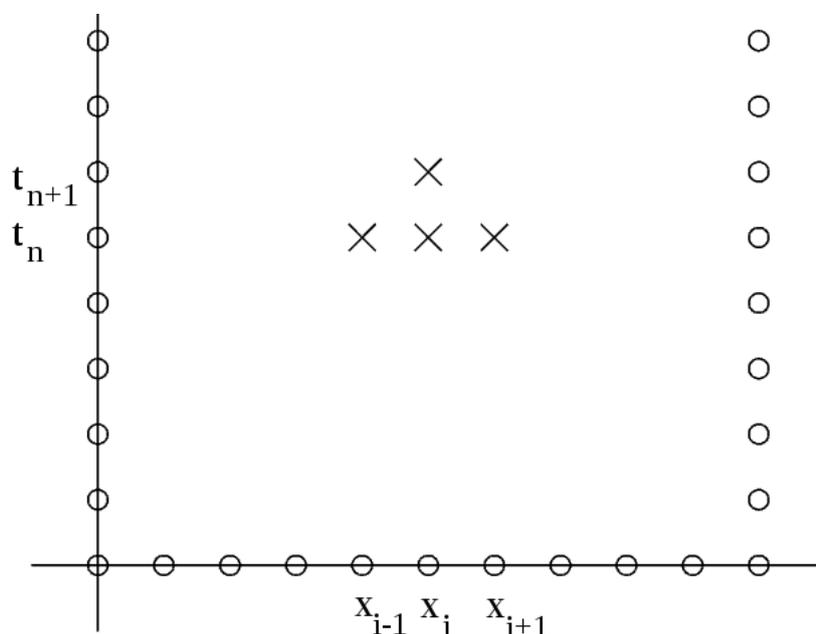


Figure 2.1: The dependence of the Euler Forward discretization scheme is as shown: for every time level  $n$  and location  $i$ ,  $u_i^{n+1}$  depends on  $u_{i-1}^n$ ,  $u_i^n$  and  $u_{i+1}^n$ .

The discretization of a variable is denoted by  $u_i^n$ , where  $n$  represents the time discretization and  $i$  the space discretization. This notation is used for the rest of this chapter, unless specified otherwise. Using the Euler Forward discretization scheme for the time and the central difference scheme for space, the equation becomes:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{(\Delta x)^2}; \quad (2.2)$$

The allowed values of  $\Delta x$  and  $\Delta t$  depend on the stability analysis of Subsection 2.2.1. The discretized Dirichlet boundary conditions are:  $u_1^n = 1$  and  $u_N^n = 0$  where  $i = N$  corresponds to the boundary at  $x = 1$ . The discretized initial condition is  $u_1^1 = 1$  and for all other  $i$ ,  $u_i^1 = 0$ .

### 2.2.1 The stability condition

For convenience, the Euler Forward discretization is rewritten as a system of  $N$  equations, one for every  $i$ :

$$\frac{\bar{u}^{n+1} - \bar{u}^n}{\Delta t} = A\bar{u}^n + \frac{\bar{f}}{(\Delta x)^2}; \quad (2.3)$$

The matrix  $A$  contains all the coefficients of  $u_i^n$  for  $i = 1 \dots N$  and the terms  $\bar{u}^n$  and  $\bar{u}^{n+1}$  on both sides of the equation represent vectors for all  $i$ . The vector  $\bar{f}$  represents the boundary conditions: it contains  $u_1^n = 1$  and  $u_i^n = 0$  for  $i = 2 \dots N$ .

$$A = \frac{1}{(\Delta x)^2} \begin{pmatrix} -2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -2 \end{pmatrix}; \quad (2.4)$$

This matrix representation allows the use of the *Gershgorin theorem* from (Vuik et al., 2006, p. 106), which involves the use of the eigenvalues of the matrix. The Gershgorin theorem is used to evaluate the stability of the system of equations.

#### The Gershgorin theorem

*The eigenvalues of a matrix  $A$  lie within a union of circles, given by:*

$$|\lambda - a_{ii}| \leq \sum_{j=1, j \neq i}^N |a_{ij}| \text{ with } \lambda \in \mathbb{C}; \quad (2.5)$$

where  $a_{ii}$  are the main diagonal terms of a matrix and  $a_{ij}$  are the nondiagonal terms.

Since any symmetrical matrix like  $A$  has only real eigenvalues, the union of circles simplifies to an equation for real  $\lambda$ . After applying the Gershgorin theorem to Equation (2.4), the inequality becomes:

$$\forall i = 1 \dots N : \left| \lambda + \frac{2}{(\Delta x)^2} \right| \leq \left| \frac{1}{(\Delta x)^2} \right| + \left| \frac{1}{(\Delta x)^2} \right|; \quad (2.6)$$

This can be further simplified to:

$$\begin{aligned} \forall i = 1 \dots N : \frac{-4}{(\Delta x)^2} &\leq \lambda \leq 0 \\ \Rightarrow \forall i = 1 \dots N : \frac{4}{(\Delta x)^2} &\geq |\lambda|; \end{aligned} \quad (2.7)$$

In the first line of equation (2.7),  $\lambda$  is negative, which implies a stable solution, and  $\lambda$  is limited from below, which will lead to the stability condition in the end.

From (Vuik et al., 2006, p. 68), it is known that the characteristic amplification factor for the Euler Forward method is given by  $Q(h\lambda) = 1 + h\lambda$ , where  $h = \Delta t$ . Any stable system can't have an amplification factor greater than 1 in absolute value, since any error would then also amplify over time with a factor greater than 1 in absolute value. Thus, for a stable numerical discretization,  $|Q(h\lambda)| < 1$  is needed. The characteristic stability condition can be derived as follows, with the assumption of negative, real  $\lambda$ .

$$\begin{aligned}
Q(h\lambda) &= |1 + h\lambda| < 1 \\
&\Leftrightarrow -1 < 1 + h\lambda < 1 \\
&\Leftrightarrow -2 < h\lambda < 0 \\
&\Leftrightarrow 2 > h|\lambda| > 0 \\
&\Leftrightarrow 2 > h|\lambda| \\
&\Leftrightarrow h < \frac{2}{|\lambda|};
\end{aligned} \tag{2.8}$$

So  $\Delta t < \frac{2}{|\lambda|}$  is needed for a stable solution. Equations (2.7) and (2.8) are two different conditions for  $\lambda$ , which can be combined to conclude the stability condition in terms of  $\Delta t$  as a function of  $\Delta x$ :

$$\begin{aligned}
\frac{2}{\Delta t} &> \frac{4}{(\Delta x)^2} \\
\Rightarrow \Delta t &< \frac{(\Delta x)^2}{2};
\end{aligned} \tag{2.9}$$

According to the stability condition in Equation (2.9) the choice for  $\Delta t$  depends on the square of the grid size. Using  $\Delta x = 1/100$ , the time step is chosen as  $\Delta t = 1/25,000$ . In Table 2.1, the relation between  $\Delta x$  and  $\Delta t$  is shown for other values of  $\Delta x$ .

Grid size $\Delta x$	Maximum $\Delta t$
$1/10$	$1/200$
$1/20$	$1/800$
$1/50$	$1/5,000$
$1/100$	$1/20,000$
$1/200$	$1/80,000$
$1/500$	$1/500,000$
$1/1000$	$1/2,000,000$

Table 2.1: The relation between the chosen grid size  $\Delta x$  and the maximum  $\Delta t$ , which follows from the stability analysis, is quadratic for the Euler Forward scheme. A finer grid increases  $\Delta t$  and so the simulation time very rapidly.

## 2.2.2 Results

See Figure 2.2 for the results of the Euler Forward discretized diffusion equation. As expected, the temperature of the system increases with time but is strictly decreasing for increasing  $x$  at all times. Under steady-state condition, the diffusion results in a temperature profile that is a straight line from  $(0, 1)$  to  $(1, 0)$ . The steady-state solution is also shown in Figure 2.2.

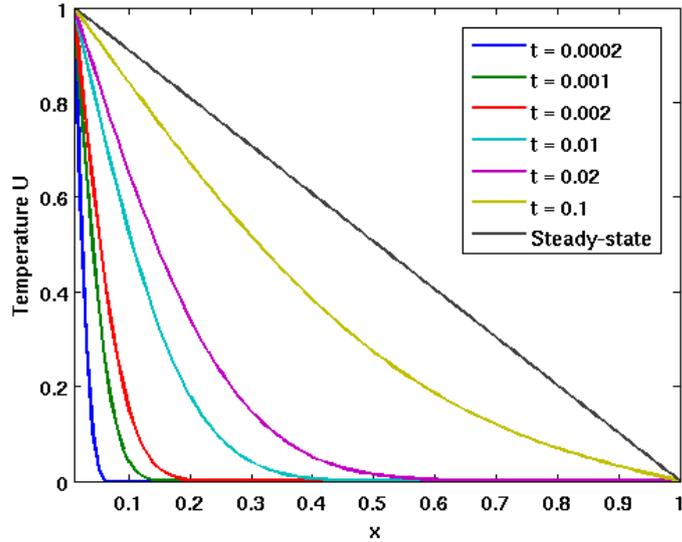


Figure 2.2: The diffusion equation as discretized with the explicit scheme of Euler Forward. Due to the stability condition and the grid with  $\Delta x = 1/100$ , the time step is  $\Delta t = 1/25,000$ .

Judging from Subsections 2.2.1 and 2.2.2, some advantages and disadvantages of this discretization scheme can be stated. As mentioned before, the Euler Forward scheme is an explicit numerical scheme, making use of information only from the previous time step. This means that the implementation is straightforward, which is an advantage. A disadvantage, that also follows from the fact that Euler Forward is an explicit method, is that the stability condition puts severe restrictions on the time step size. In Equation (2.9) and in Table 2.1 it is shown that  $\Delta t$  is proportional to  $(\Delta x)^2$ , which means that a decreasing grid size results in an even more rapidly decreasing time step and a rapidly increasing simulation processing time. Especially for more complex functions, such as the Cahn-Hilliard equation, this can be a problem.

In the next section, an implicit scheme, namely the Euler Backward scheme, is considered. In comparison to Euler Forward, the discretization and the stability condition are different, which leads to other advantages and also disadvantages.

### 2.3 The implicit method of Euler Backward in 1D

Euler Backward is an implicit method, which means that, unlike with an explicit method, every next time step is calculated making use of that same time step. Any time derivative still contains terms of the current and the next time step, but spatial derivatives are expressed in terms of the next time step only. See Figure 2.3 for the dependence scheme. Generally, the stability criteria of implicit schemes are a lot easier to satisfy, the schemes are not as stiff as the explicit ones.

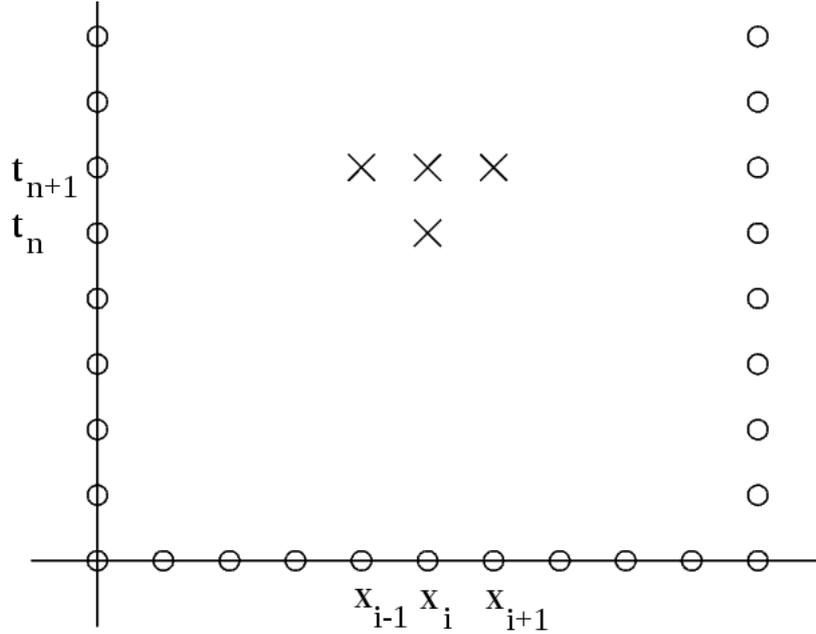


Figure 2.3: The dependance of the Euler Backward discretization scheme is the opposite of the Euler Forward scheme:  $u_{i-1}^{n+1}$ ,  $u_i^{n+1}$  and  $u_{i+1}^{n+1}$  depend on each other and on  $u_i^n$ , for all  $n$  and  $i$ .

The discretization of the diffusion equation, using the Euler Backward scheme, is:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}}{(\Delta x)^2}; \quad (2.10)$$

The discretization of the Dirichlet boundary conditions and initial condition are the same as with the Euler Forward scheme. The discretized boundary conditions are:  $u_1^n = 1$  and  $u_N^n = 0$ . The discretized initial condition is  $u_1^1 = 1$  and  $u_i^1 = 0$  for  $i = 2 \dots N$ .

As can be seen in Figure 2.3 and in the discretization, three different terms involving  $n + 1$  are used, terms which are all unknown at time step  $n$ . Calculating these three terms involves solving a system of equations for the spatial discretization. When the grid size is  $N$ , this results in a system of  $N$  equations which can be put in a matrix of size  $N$  by  $N$ . The discretization becomes:

$$\frac{\bar{u}^{n+1} - \bar{u}^n}{\Delta t} = K\bar{u}^{n+1} + \frac{\bar{f}}{(\Delta x)^2}; \quad (2.11)$$

where  $K$  is the  $N$  by  $N$  matrix and all terms of  $\bar{u}^n$  and  $\bar{u}^{n+1}$  are vectors of length  $N$ . The matrix  $K$  is similar to the matrix  $A$  from Equation (2.4), the vector  $\bar{f}$  containing the boundary conditions is also similar to  $\bar{f}$  from Equation (2.4). All terms of  $\bar{u}^{n+1}$  are now moved to one side of the equation:

$$\bar{u}^n = (I - \Delta t \cdot K)\bar{u}^{n+1} - \frac{\Delta t \bar{f}}{(\Delta x)^2}; \quad (2.12)$$

Equation (2.12) can be rewritten as:

$$\bar{u}^{n+1} = (I - \Delta t \cdot K)^{-1} \left( \bar{u}^n + \frac{\Delta t \bar{f}}{(\Delta x)^2} \right); \quad (2.13)$$

Here, the matrix  $I$  is an identity matrix of size  $N$  by  $N$ . The expression in Equation (2.13) gives  $\bar{u}^{n+1}$  in terms of  $\bar{u}^n$ , involving an inverse matrix. This is the final expression, which is implemented in Matlab.

Figure 2.4 shows a plot created with the Euler Backward scheme. The grid that has been used is the same as with the Euler Forward scheme, so  $\Delta x = 1/100$ . The time step was set at  $\Delta t = 1/500$ . Why the time step can be much bigger than with the Euler Forward scheme, is explained in Subsection 2.3.1.

### 2.3.1 Stability condition

In this stability analysis, the method to obtain the stability condition of the Euler Backward scheme is the same as the one that was used to obtain the stability condition of the Euler Forward scheme. First, the Gershgorin theorem is applied.

$$|\lambda - a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ij}|; \quad (2.14)$$

The matrix  $K$  is the same as the matrix  $A$  from Equation (2.4), so the result of applying the Gershgorin theorem is also the same:

$$\forall i = 1 \dots N : \frac{4}{(\Delta x)^2} \geq |\lambda|; \quad (2.15)$$

According to (Vuik et al., 2006, p. 68), the characteristic amplification factor for Euler Backward is  $Q(h\lambda) = \frac{1}{1-h\lambda}$  where  $h$  is  $\Delta t$ . As before, the discretization scheme requires  $|Q(h\lambda)| < 1$  to be stable:

$$\begin{aligned} Q(h\lambda) &= \frac{1}{1-h\lambda} < 1 \\ &\Leftrightarrow |1-h\lambda| > 1 \\ &\Leftrightarrow 1-h\lambda > 1 \vee h\lambda - 1 > 1 \\ &\Leftrightarrow h\lambda < 0 \vee h\lambda > 2 \\ &\Leftrightarrow h|\lambda| > 0; \end{aligned} \quad (2.16)$$

In the last step it is used that  $\lambda$  is negative so  $h\lambda > 2$  drops out. Combining Equation (2.15) and Equation (2.16) is actually not even necessary anymore, the conclusion just follows from Equation (2.16):

$$\Delta t |\lambda| > 0; \quad (2.17)$$

This inequality is easily satisfied: any positive  $\Delta t$  will suffice. The limiting factor on  $\Delta t$  is the characteristic time constant of the system, over which the actual physical changes take place. This time constant is defined as follows, according to (van den Akker and Mudde, 2008):

$$\tau = \frac{L^2}{\pi D}; \quad (2.18)$$

Here,  $\tau$  is the characteristic time constant,  $L$  is the length of the domain and  $D$  is the diffusivity. All parameters are used dimensionless. Filling in the values for  $L$  and  $D$  as used here, in Equation (2.18), results in  $\tau = 0.32$ . All major changes occur in the time interval  $t = [0, \tau]$ . To accurately resolve these changes,  $\Delta t \ll \tau$  is needed.

### 2.3.2 Results

In plotting Figure 2.4, the stability condition of Subsection 2.3.1 resulted in  $\Delta t = 1/500$ . From a stability point of view, the chosen time step can be as high as  $\Delta t = 1/10$  or  $\Delta t = 1/5$ , although the accuracy of the solution decreases significantly. Figure 2.5 shows the inaccuracy of a large time step.

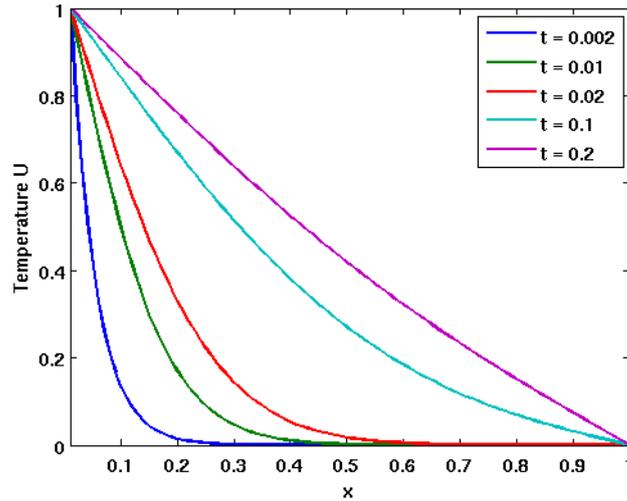


Figure 2.4: The Euler Backward implicit scheme makes large time steps possible. This plot is identical to the plot in Figure 2.2 but the used time step is  $\Delta t = 1/500$ , which is a factor 50 larger.

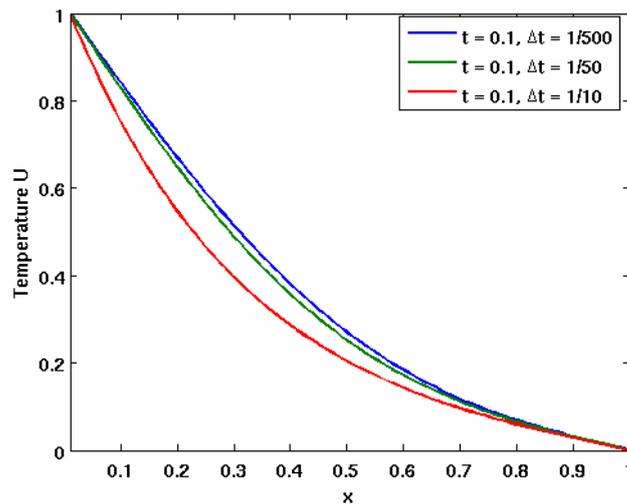


Figure 2.5: Using the Euler Backward scheme,  $\Delta t$  can be chosen as large as desired. Although the simulation runs extremely fast with  $\Delta t$  very large, the accuracy decreases significantly. This is still a limiting factor in choosing  $\Delta t$ .

Comparing Figure 2.4 and Figure 2.2, one can conclude that, for fine enough  $\Delta t$  and  $\Delta x$ , both discretization schemes have the same results. The differences are in the stability conditions. A great advantage of the Euler Backward scheme is that the time step can be chosen a lot larger than while using the Euler Forward scheme, due to the fact that Euler Backward is an unconditionally stable scheme. The time step is bounded though, since a very large time step results in inaccuracies and the loss of desired data. A disadvantage, in comparison to the Euler Forward scheme, is the fact that Euler Backward is slightly harder to implement in Matlab.

### 3 Modeling the Cahn-Hilliard equation with the Euler Forward discretization

In this chapter, the application of the Euler Forward discretization scheme to the one-dimensional Cahn-Hilliard equation is discussed. Furthermore, the discretization involves central difference schemes for the second and fourth order spatial derivatives. The boundary conditions are applied as described in literature and a step-function is chosen as an initial condition. Next, a stability analysis is used to show what the relation between grid sizes and critically stable time step sizes is. This theoretical stability condition is compared to the stability condition, obtained from numerical experiments. At last, the results of the implementation are discussed quantitatively.

#### 3.1 Discretizations

The Cahn-Hilliard equation in three dimensions has been derived in Chapter 1. Derived from Equation (1.11), the one dimensional version can be written as:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 \mu}{\partial x^2} = D \frac{\partial^2}{\partial x^2} \left[ c^3 - c - \epsilon^2 \frac{\partial^2 c}{\partial x^2} \right]; \quad (3.1)$$

where  $D$  is the diffusivity (before, this was the mobility of the particles  $M$ ),  $\epsilon^2$  is a phenomenological parameter which scales with the width of the interface and  $\mu$  is the chemical potential. Applying the Euler Forward scheme to Equation (3.1), using  $D = 1$ , yields the following discretization:

$$\begin{aligned} \frac{c_i^{n+1} - c_i^n}{\Delta t} &= \frac{\mu_{i-1}^n - 2\mu_i^n + \mu_{i+1}^n}{(\Delta x)^2} \\ &= \frac{(c_{i-1}^n)^3 - 2(c_i^n)^3 + (c_{i+1}^n)^3}{(\Delta x)^2} - \frac{c_{i-1}^n - 2c_i^n + c_{i+1}^n}{(\Delta x)^2} \\ &\quad - \epsilon^2 \left( \frac{c_{i-2}^n - 4c_{i-1}^n + 6c_i^n - 4c_{i+1}^n + c_{i+2}^n}{(\Delta x)^4} \right); \end{aligned} \quad (3.2)$$

In Equation (3.2), the term  $\frac{\partial^2(c^3)}{\partial x^2}$  from Equation (3.1) is discretized directly. An alternative is to rewrite Equation (3.1) so that the second derivative is evaluated and, after that, discretize the result. Equation (3.1) is rewritten accordingly as:

$$\frac{\partial c}{\partial t} = \frac{\partial^2}{\partial x^2} \left[ c^3 - c - \epsilon^2 \frac{\partial^2 c}{\partial x^2} \right] = 6c \left( \frac{\partial c}{\partial x} \right)^2 + (3c^2 - 1) \frac{\partial^2 c}{\partial x^2} - \epsilon^2 \frac{\partial^4 c}{\partial x^4}; \quad (3.3)$$

The discretization of Equation (3.3) is as follows:

$$\begin{aligned} \frac{c_i^{n+1} - c_i^n}{\Delta t} &= 6c_i^n \left( \frac{c_{i+1}^n - c_{i-1}^n}{2\Delta x} \right)^2 + ((3c_i^n)^2 - 1) \left( \frac{c_{i-1}^n - 2c_i^n + c_{i+1}^n}{(\Delta x)^2} \right) \\ &\quad - \epsilon^2 \left( \frac{c_{i-2}^n - 4c_{i-1}^n + 6c_i^n - 4c_{i+1}^n + c_{i+2}^n}{(\Delta x)^4} \right); \end{aligned} \quad (3.4)$$

As will be shown later, the implementation of this alternative doesn't give desired results. This might be caused by the terms  $6c \left( \frac{\partial c}{\partial x} \right)^2$  or  $(3c^2 - 1) \frac{\partial^2 c}{\partial x^2}$  in Equation (3.3). From the diffusive term  $(3c^2 - 1) \frac{\partial^2 c}{\partial x^2}$ , it can be seen that negative diffusion only occurs for  $|c| < \frac{\sqrt{3}}{3}$ , any  $c$  outside this range results in positive diffusion. If the behavior of  $c$  is as expected, though,  $c$  is bounded in the range of  $-1 \leq c \leq 1$ , which overlaps with the range of  $|c| > \frac{\sqrt{3}}{3}$ . Therefore, the discretization of Equation (3.2), and not of Equation (3.4), is used throughout the rest of this chapter.

## 3.2 Boundary and initial conditions

The boundary conditions are defined as has been mentioned in Section 1.3: no concentration gradient or chemical potential on both boundaries. Physically, this means that the in- and outflow of the system equals zero, or, in other words: it is a closed system so mass conservation is satisfied. Mathematically, this means that the boundary conditions are Neumann boundary conditions, involving the derivative of both  $c$  and  $\mu$  at both boundaries. It makes sense that four boundary conditions are used, since the Cahn-Hilliard equation is a fourth order differential equation. The boundaries of the (one-dimensional) system are located at  $x = 0$  and  $x = 1$ .

$$\frac{\partial c}{\partial x}(0, t) = 0 \quad \frac{\partial c}{\partial x}(1, t) = 0 \quad \frac{\partial \mu}{\partial x}(0, t) = 0 \quad \frac{\partial \mu}{\partial x}(1, t) = 0; \quad (3.5)$$

As stated in Section 1.3, the initial condition is a step-function. This initial condition is chosen because its results can be compared with an analytical solution.

$$c^{t=0} = \begin{cases} -1 & \text{for } 0 \leq x \leq \frac{1}{2} \\ 1 & \text{for } \frac{1}{2} < x \leq 1 \end{cases} ; \quad (3.6)$$

The fluid interface that arises due to the Cahn-Hilliard equation, is located at  $x = \frac{1}{2}$ , as is the case with the initial condition. In terms of concentration, this interface is located at  $c = 0$ .

### 3.2.1 Discretized boundary and initial conditions

The boundary conditions of the Cahn-Hilliard equation are discretized in a different way than in Chapter 2.2, since they are Neumann boundary conditions instead of Dirichlet. The discretizations are as follows:

$$\begin{aligned} \frac{c_1^n - c_{-1}^n}{2\Delta x} = 0 &\Rightarrow c_{-1}^n = c_1^n; \\ \frac{c_{N+1}^n - c_{N-1}^n}{2\Delta x} = 0 &\Rightarrow c_{N+1}^n = c_{N-1}^n; \\ \frac{\mu_1^n - \mu_{-1}^n}{2\Delta x} = 0 &\Rightarrow \mu_{-1}^n = \mu_1^n; \\ \frac{\mu_{N+1}^n - \mu_{N-1}^n}{2\Delta x} = 0 &\Rightarrow \mu_{N+1}^n = \mu_{N-1}^n; \end{aligned} \quad (3.7)$$

Because of these discretizations, the ghost points at  $i = -1$  and at  $i = N + 1$  can be substituted with known values. The discretization of the initial condition is:

$$c^{t=0} = \begin{cases} -1 & \text{for } 0 \leq i \leq \frac{1}{2}N \\ 1 & \text{for } \frac{1}{2}N + 1 \leq i \leq N \end{cases} ; \quad (3.8)$$

## 3.3 Experimental and theoretical stability conditions

Since the Cahn-Hilliard equation is a nonlinear fourth order differential equation, the stability condition is a lot harder to satisfy than the stability condition of the diffusion equation, Equation (2.9). Because the stability condition is also harder to derive, a combination of an experimental trial-and-error approach and a stability analysis, similar to the one in Subsection 2.2.1, is used. The experimental approach leads to an experimental numerical stability condition and the Gershgorin theorem approach leads to the theoretical stability condition. According to (Vuik et al., 2006), the Gershgorin theorem can only be applied to a linear system of equations, while a nonlinear differential equation can not be presented as a linear system of equations. Therefore, the Cahn-Hilliard equation is linearized in Subsection 3.3.1. The stability analysis, regarding the linearized Cahn-Hilliard equation, is done in Subsections 3.3.2 and 3.3.3. Next, the theoretical stability condition is compared to the experimental numerical stability in Subsection 3.3.4.

### 3.3.1 The linearized Cahn-Hilliard equation

Because of the nonlinearity of the Cahn-Hilliard equation, the Gershgorin theorem is not applicable to it. The Cahn-Hilliard equation is therefore linearized and the Gershgorin theorem is applied to the linearization. The linearization only concerns the second derivative of the nonlinear term  $c^3$ .

For the linearization, the following representation of Equation (3.1) is used:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x} \left[ (3c^2 - 1) \frac{\partial c}{\partial x} \right] - \epsilon^2 \frac{\partial^4 c}{\partial x^4}; \quad (3.9)$$

Next, the fact that  $3c^2 \in [0, 3]$  is used, it follows from the fact that  $c \in [-1, 1]$ . Taking the two bounds of the range of  $3c^2$  gives an estimate of what the range of the entire differential equation is.

$$\frac{\partial c}{\partial t} = \begin{cases} \frac{\partial}{\partial x} \left[ -1 \frac{\partial c}{\partial x} \right] - \epsilon^2 \frac{\partial^4 c}{\partial x^4} & \text{if } 3c^2 = 0 \\ \frac{\partial}{\partial x} \left[ 2 \frac{\partial c}{\partial x} \right] - \epsilon^2 \frac{\partial^4 c}{\partial x^4} & \text{if } 3c^2 = 3 \end{cases}; \quad (3.10)$$

Which is rewritten as:

$$\frac{\partial c}{\partial t} = \begin{cases} -1 \frac{\partial^2 c}{\partial x^2} - \epsilon^2 \frac{\partial^4 c}{\partial x^4} & \text{if } 3c^2 = 0 \\ 2 \frac{\partial^2 c}{\partial x^2} - \epsilon^2 \frac{\partial^4 c}{\partial x^4} & \text{if } 3c^2 = 3 \end{cases}; \quad (3.11)$$

Equation (3.11) can now be written as a system of equations, using a second derivative operator  $A$ :

$$\begin{aligned} \frac{\partial c}{\partial t} &= -Ac - \epsilon^2 A^2 c & \vee & \frac{\partial c}{\partial t} = 2Ac - \epsilon^2 A^2 c \\ \Rightarrow \frac{\partial c}{\partial t} &= (-A - \epsilon^2 A^2)c & \vee & \frac{\partial c}{\partial t} = (2A - \epsilon^2 A^2)c \\ \Rightarrow \frac{\partial c}{\partial t} &= B_1 c & \vee & \frac{\partial c}{\partial t} = B_2 c \end{aligned} \quad (3.12)$$

with:

$$A = \frac{1}{(\Delta x)^2} \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -1 \end{pmatrix}; \quad (3.13)$$

and:

$$B_1 = \frac{1}{(\Delta x)^2} \times \begin{pmatrix} \frac{-2\epsilon^2}{(\Delta x)^2} + 1 & \frac{3\epsilon^2}{(\Delta x)^2} - 1 & \frac{-\epsilon^2}{(\Delta x)^2} & 0 & \dots & 0 & 0 & 0 \\ \frac{3\epsilon^2}{(\Delta x)^2} - 1 & \frac{-6\epsilon^2}{(\Delta x)^2} + 2 & \frac{4\epsilon^2}{(\Delta x)^2} - 1 & \frac{-\epsilon^2}{(\Delta x)^2} & 0 & \dots & 0 & 0 \\ \frac{-\epsilon^2}{(\Delta x)^2} & \frac{4\epsilon^2}{(\Delta x)^2} - 1 & \frac{-6\epsilon^2}{(\Delta x)^2} + 2 & \frac{4\epsilon^2}{(\Delta x)^2} - 1 & \frac{-\epsilon^2}{(\Delta x)^2} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \\ 0 & \dots & 0 & \frac{-\epsilon^2}{(\Delta x)^2} & \frac{4\epsilon^2}{(\Delta x)^2} - 1 & \frac{-6\epsilon^2}{(\Delta x)^2} + 2 & \frac{4\epsilon^2}{(\Delta x)^2} - 1 & \frac{-\epsilon^2}{(\Delta x)^2} \\ 0 & 0 & \dots & 0 & \frac{-\epsilon^2}{(\Delta x)^2} & \frac{4\epsilon^2}{(\Delta x)^2} - 1 & \frac{-6\epsilon^2}{(\Delta x)^2} + 2 & \frac{3\epsilon^2}{(\Delta x)^2} - 1 \\ 0 & 0 & 0 & \dots & 0 & \frac{-\epsilon^2}{(\Delta x)^2} & \frac{3\epsilon^2}{(\Delta x)^2} - 1 & \frac{-2\epsilon^2}{(\Delta x)^2} + 1 \end{pmatrix}; \quad (3.14)$$

and:

$$B_2 = \frac{1}{(\Delta x)^2} \times \begin{pmatrix} \frac{-2\epsilon^2}{(\Delta x)^2} - 2 & \frac{3\epsilon^2}{(\Delta x)^2} + 2 & \frac{-\epsilon^2}{(\Delta x)^2} & 0 & \dots & 0 & 0 & 0 \\ \frac{3\epsilon^2}{(\Delta x)^2} + 2 & \frac{-6\epsilon^2}{(\Delta x)^2} - 4 & \frac{4\epsilon^2}{(\Delta x)^2} + 2 & \frac{-\epsilon^2}{(\Delta x)^2} & 0 & \dots & 0 & 0 \\ \frac{-\epsilon^2}{(\Delta x)^2} & \frac{4\epsilon^2}{(\Delta x)^2} + 2 & \frac{-6\epsilon^2}{(\Delta x)^2} - 4 & \frac{4\epsilon^2}{(\Delta x)^2} + 2 & \frac{-\epsilon^2}{(\Delta x)^2} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \\ 0 & \dots & 0 & \frac{-\epsilon^2}{(\Delta x)^2} & \frac{4\epsilon^2}{(\Delta x)^2} + 2 & \frac{-6\epsilon^2}{(\Delta x)^2} - 4 & \frac{4\epsilon^2}{(\Delta x)^2} + 2 & \frac{-\epsilon^2}{(\Delta x)^2} \\ 0 & 0 & \dots & 0 & \frac{-\epsilon^2}{(\Delta x)^2} & \frac{4\epsilon^2}{(\Delta x)^2} + 2 & \frac{-6\epsilon^2}{(\Delta x)^2} - 4 & \frac{3\epsilon^2}{(\Delta x)^2} + 2 \\ 0 & 0 & 0 & \dots & 0 & \frac{-\epsilon^2}{(\Delta x)^2} & \frac{3\epsilon^2}{(\Delta x)^2} + 2 & \frac{-2\epsilon^2}{(\Delta x)^2} - 2 \end{pmatrix}; \quad (3.15)$$

To the representation in Equation (3.12), the Gershgorin theorem can be applied.

### 3.3.2 Applying the Gershgorin theorem

Applying the Gershgorin theorem helps to find the eigenvalues of  $B_1$  and  $B_2$ , which are needed in the end to find the stability conditions. For the complete Gershgorin theorem, see Equation (2.5). First, the derivation of the eigenvalues is done using the matrix  $B_1$ . The matrix  $B_2$  will be used later, separately. Filling in the Gershgorin theorem for the middle rows of  $B_1$  yields:

$$\left| \lambda - \left( \frac{-6\epsilon^2}{(\Delta x)^2} + 2 \right) \frac{1}{(\Delta x)^2} \right| \leq \left| \left( \frac{8\epsilon^2}{(\Delta x)^2} - 2 \right) \frac{1}{(\Delta x)^2} \right| + \frac{2\epsilon^2}{(\Delta x)^4}; \quad (3.16)$$

This equation has to be divided into three cases. These cases, and the results of Equation (3.16) in these cases, are:

$$\begin{aligned} \frac{\epsilon}{\Delta x} < \frac{1}{2} &\Rightarrow |\lambda| \leq \frac{4}{(\Delta x)^2} - \frac{12\epsilon^2}{(\Delta x)^4} \\ \frac{1}{2} < \frac{\epsilon}{\Delta x} < \frac{1}{\sqrt{3}} &\Rightarrow |\lambda| \leq \frac{4}{(\Delta x)^2} \\ \frac{1}{\sqrt{3}} < \frac{\epsilon}{\Delta x} &\Rightarrow |\lambda| \leq \frac{16\epsilon^2}{(\Delta x)^4} - \frac{4}{(\Delta x)^2} \end{aligned} \quad (3.17)$$

Next, the eigenvalues of  $B_2$  are derived by filling in the Gershgorin theorem again.

$$\left| \lambda - \left( \frac{-6\epsilon^2}{(\Delta x)^2} - 4 \right) \frac{1}{(\Delta x)^2} \right| \leq \left| \left( \frac{8\epsilon^2}{(\Delta x)^2} + 4 \right) \frac{1}{(\Delta x)^2} \right| + \frac{2\epsilon^2}{(\Delta x)^4}; \quad (3.18)$$

Here, no separate cases have to be dealt with. Rewriting Equation (3.18) yields:

$$|\lambda| \leq \frac{8}{(\Delta x)^2} + \frac{16\epsilon^2}{(\Delta x)^4}; \quad (3.19)$$

After combining the inequalities of Equation (3.17) with the inequality of Equation (3.19), the only remaining inequality is:

$$\forall \frac{\epsilon}{\Delta x} : |\lambda| \leq \frac{8}{(\Delta x)^2} + \frac{16\epsilon^2}{(\Delta x)^4}; \quad (3.20)$$

Equation (3.20) is the final and only condition on  $\lambda$ .

### 3.3.3 The stability condition on $\Delta t$

As has been discussed in Subsection 2.2.1 (and can be found in Equation (2.8)), the discretization of Euler Forward prescribes that  $\Delta t < \frac{2}{|\lambda|}$  holds, which can be rewritten as  $|\lambda| < \frac{2}{\Delta t}$ . Combining inequality

(3.20) with the Euler Forward inequality, results in:

$$\Delta t < \frac{(\Delta x)^2}{4 + \frac{8\epsilon^2}{(\Delta x)^2}}; \quad (3.21)$$

Equation (3.21) concludes with a useful stability condition for  $\Delta t$  in terms of  $\Delta x$ . However, literature on the Cahn-Hilliard equation states that  $\Delta t \propto (\Delta x)^4$ , instead of a relation like in Equation (3.21). For  $\epsilon \gg \Delta x$ , Equation (3.21) converges to a relation like  $\Delta t \propto (\Delta x)^4$  indeed. Validation of this hypothesis comes from the actual numerical experiments.

Equation (3.21) is just the stability condition for a linearized Cahn-Hilliard equation, not the original one. In the next subsection, the results of Equation (3.21) and the experimental data on stability are compared.

### 3.3.4 Comparing the linearized Cahn-Hilliard equation stability condition with the experimental data on stability

Concerning the numerical experiments, the definition that is used for a stable solution is as follows: the solution of the discretization still oscillates after  $t = 5,000\Delta t$  but doesn't oscillate anymore after  $t = 10,000\Delta t$ . This is not the best definition for stability but it is consistent though.

See Table 3.1 for the comparison of the numerical experiments and the results of Equation (3.21). The values of all  $\Delta t_{Exp}$  and  $\Delta t_{Theory}$  are close. This can be seen even better from the ratio  $\frac{\Delta t_{Exp}}{\Delta t_{Theory}}$ , also in Table 3.1. Because the data in Table 3.1 are so close, there is no need to plot both the theoretical data and the experimental data. A plot of only the experimental data is shown in Figure 3.1.

$\epsilon = 0.25\Delta x$				$\epsilon = 0.5\Delta x$			
$\Delta x$	$\Delta t_{Exp}$	$\Delta t_{Theory}$	$\frac{\Delta t_{Exp}}{\Delta t_{Theory}}$	$\Delta x$	$\Delta t_{Exp}$	$\Delta t_{Theory}$	$\frac{\Delta t_{Exp}}{\Delta t_{Theory}}$
$1/32$	$1/4,572$	$1/4,608$	1.0079	$1/32$	$1/6,055$	$1/6,144$	1.0147
$1/64$	$1/18,400$	$1/18,432$	1.0017	$1/64$	$1/24,510$	$1/24,576$	1.0027
$1/128$	$1/73,720$	$1/73,728$	1.0001	$1/128$	$1/98,280$	$1/98,304$	1.0002
$\epsilon = \Delta x$				$\epsilon = 2\Delta x$			
$\Delta x$	$\Delta t_{Exp}$	$\Delta t_{Theory}$	$\frac{\Delta t_{Exp}}{\Delta t_{Theory}}$	$\Delta x$	$\Delta t_{Exp}$	$\Delta t_{Theory}$	$\frac{\Delta t_{Exp}}{\Delta t_{Theory}}$
$1/32$	$1/12,040$	$1/12,288$	1.0206	$1/32$	$1/36,150$	$1/36,864$	1.0198
$1/64$	$1/48,930$	$1/49,152$	1.0045	$1/64$	$1/147,000$	$1/147,456$	1.0031
$1/128$	$1/196,600$	$1/196,608$	1.0000	$1/128$	$1/589,800$	$1/589,824$	1.0000
$\epsilon = 4\Delta x$				$\epsilon = 8\Delta x$			
$\Delta x$	$\Delta t_{Exp}$	$\Delta t_{Theory}$	$\frac{\Delta t_{Exp}}{\Delta t_{Theory}}$	$\Delta x$	$\Delta t_{Exp}$	$\Delta t_{Theory}$	$\frac{\Delta t_{Exp}}{\Delta t_{Theory}}$
$1/32$	$1/133,100$	$1/135,168$	1.0155	$1/32$	$1/523,000$	$1/528,384$	1.0103
$1/64$	$1/538,800$	$1/540,672$	1.0035	$1/64$	$1/2,107,000$	$1/2,113,536$	1.0031
$1/128$	$1/2,163,000$	$1/2,162,688$	0.9986	$1/128$	$1/8,451,000$	$1/8,454,144$	1.0004

Table 3.1: The stability conditions for the Euler Forward implementation, as obtained experimentally, are compared to the theoretical stability condition, obtained from the stability analysis of the linearized Cahn-Hilliard equation. For all different  $\frac{\epsilon}{\Delta x}$  and  $\Delta x$ , the values of  $\Delta t_{Exp}$  and  $\Delta t_{Theory}$  are very close, which is shown by the ratio  $\frac{\Delta t_{Exp}}{\Delta t_{Theory}}$ .

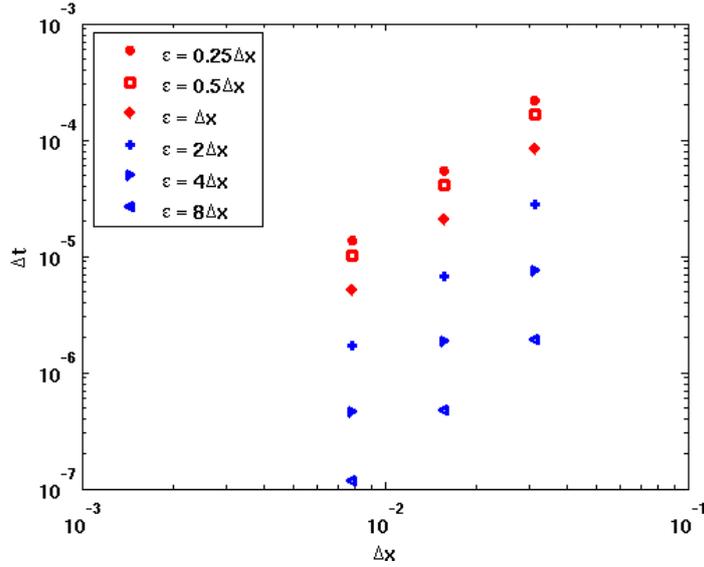


Figure 3.1: The relation between  $\Delta t$  and  $\Delta x$  for several values of  $\epsilon$  due to the stability condition obtained from experimental data. For constant  $\frac{\epsilon}{\Delta x}$ , increasing  $\Delta x$  by a factor 2 results in increasing  $\Delta t$  by a factor 4.

From Figure 3.1 and Figure 3.2, it can be concluded that  $\Delta t \propto (\Delta x)^2$  for all  $\epsilon$  and that, for any constant  $\Delta x$ , a larger  $\epsilon$  results in a smaller maximum  $\Delta t$ . It can also be concluded that, for given  $\Delta x$  and for at least the values of  $\frac{\epsilon}{\Delta x} \in [0.25, 8]$ , the theoretical stability condition is a very good approximation. Therefore, the theoretical stability condition can be used as a predictor for  $\Delta t$  if one is implementing the Cahn-Hilliard equation in Euler Forward with a particular  $\Delta x$  and  $\frac{\epsilon}{\Delta x} \in [0.25, 8]$ . More specifically, for increasing  $\Delta x$ , the ratio in Table 3.1 converges to 1. Extrapolating this result, this stability condition predictor might hold for larger  $\Delta x$  than  $\Delta x = 1/128$ .

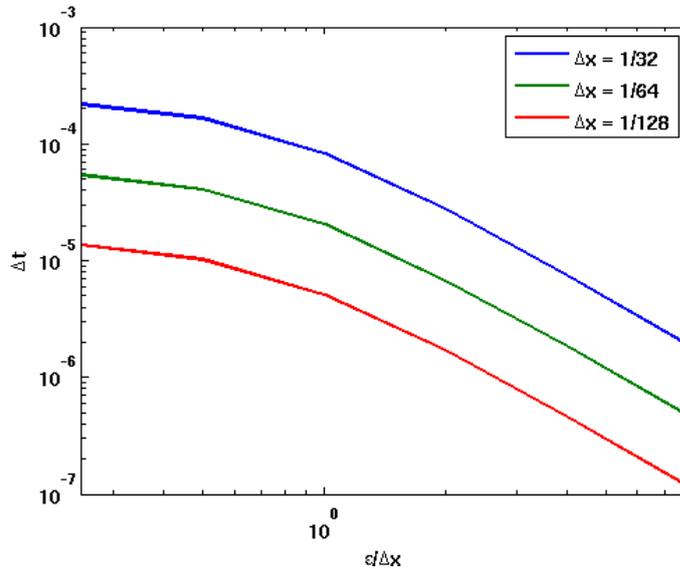


Figure 3.2: For each value of  $\Delta x$ , the relation between  $\frac{\epsilon}{\Delta x}$  and  $\Delta t$  follows a similar curve. Comparing two curves, a factor 4 increase in  $\Delta t$  results in a factor 2 increase in  $\Delta x$ .

### 3.4 Results

In this section, the results of the evolution of concentration with time are presented, using the discretization of the Cahn-Hilliard equation of Equation (3.2). In the implementations that are used to obtain the results, for all plots with different values of  $\epsilon$ , the time step size is a bit smaller than the critical time step sizes of Table 3.1. The grid size is  $\Delta x = 1/128$  for all plots.

Figure 3.3 shows the plot of the concentration with  $\epsilon = 1/128$ . The evolution of the interface, shown by different plots which are indicated in the legend, can't be seen as good as desired since, with this particular  $\epsilon$ , the interface is quite thin. As expected though, the plots converge to the steady-state solution with time.

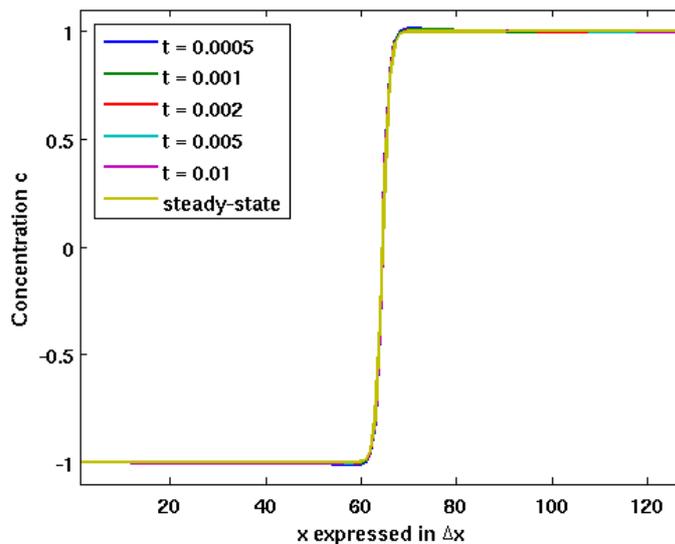


Figure 3.3: The Cahn-Hilliard equation implemented with the Euler Forward discretization scheme, using  $\Delta x = 1/128$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/200.000$ .

Figure 3.4 shows the plot of the concentration with  $\epsilon = 4/128$ . The evolution of the interface can be seen more clearly, compared to Figure 3.3: lower values of  $t$  consequently show steeper interfaces than higher values of  $t$ . Obviously, the steady-state solution shows the least steep interface.

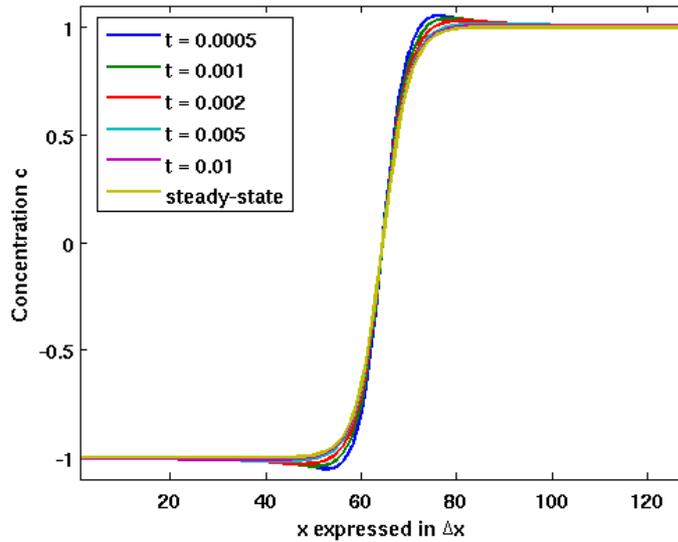


Figure 3.4: The Cahn-Hilliard equation implemented with the Euler Forward discretization scheme, using  $\Delta x = 1/128$ ,  $\epsilon = 4/128$  and  $\Delta t = 1/2,200,000$ . The maxima and minima near the interface are discussed in Subsection 3.5.3.

Figure 3.5 shows only the interface of the plot with  $\epsilon = 0.25/128$ . The solutions are consequently even steeper than with  $\epsilon = 1/128$ . Therefore, even though the plot uses only the near-interface data points, the evolution of the interface is impossible to see.

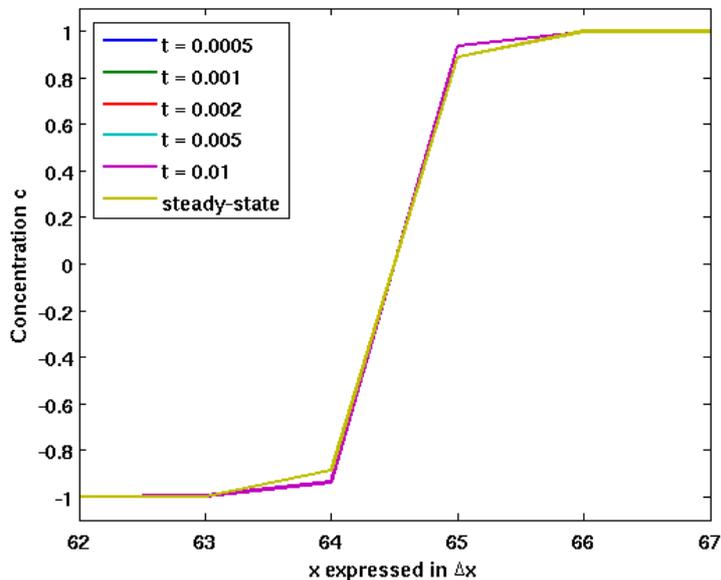


Figure 3.5: The Cahn-Hilliard equation implemented with the Euler Forward discretization scheme, using  $\Delta x = 1/128$ ,  $\epsilon = 0.25/128$  and  $\Delta t = 1/80,000$ .

In Section 3.1, two different discretizations of the Euler Forward scheme were presented. The plots in this section were made using the discretization of Equation (3.2) until here. In Figure 3.6, it is shown why the discretization of Equation (3.4) doesn't give the desired results. For increasing  $t$ , the solution

doesn't converge to the steady-state solution, instead it converges to the steady-state solution with a magnitude of approximately  $\sqrt{3}$  higher, exceeding  $c = 1$  and  $c = -1$  on almost the entire domain already for  $t = 0.01$ . It seems that the integral over all  $c$  is the same for all  $t$ , though, which is as expected. For lower values of  $\epsilon$ , the solutions converge to an even higher magnitude of the steady-state solution. For higher values of  $\epsilon$ , though, the solutions seem to converge to the desired steady-state solution. Unfortunately, this was not investigated further since the discretization of Equation (3.2) already gives desired results.

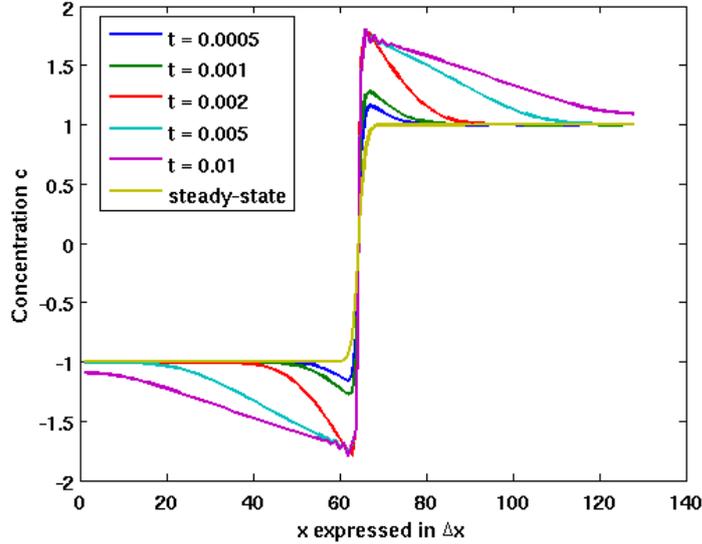


Figure 3.6: The Cahn-Hilliard equation implemented with the Euler Forward discretization scheme as in Equation (3.4), using  $\Delta x = 1/128$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/400.000$ . The solutions diverge for increasing  $t$ , which is not in accordance with the expectations and the theory of Section 1.2.

## 3.5 Quantitative analysis of the results

### 3.5.1 Steady-state deviation

As discussed in Section 3.4, the interface evolution always converges to the steady-state solution. How fast the plots converge, can be seen best in a plot of the steady-state deviation, which is defined as:

$$\text{Steady-state deviation } (x, t) = |c_{\text{steady}}(x) - c(x, t)|; \quad (3.22)$$

Here,  $c_{\text{steady}}(x)$  is from Equation (1.15). Figure 3.7 shows the steady-state deviation of the case with  $\epsilon = 1/128$ .

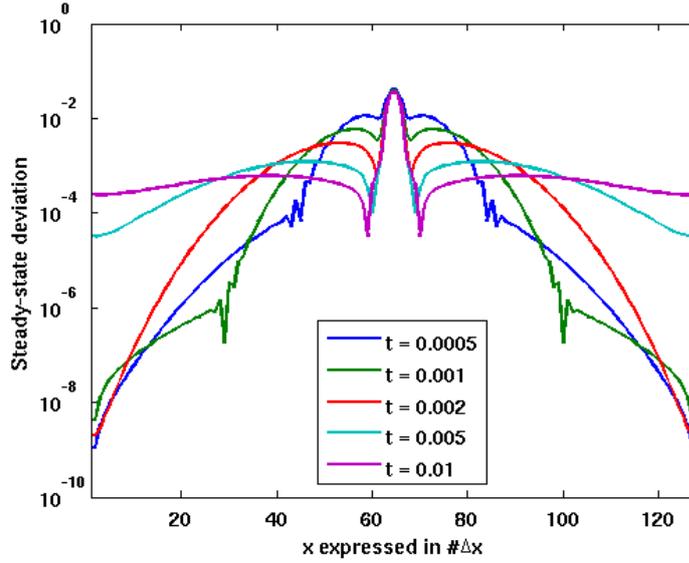


Figure 3.7: The steady-state deviation is defined as the absolute difference between the solution at time  $t$  and the steady-state solution. The deviation as shown here, is of the implementation with  $\Delta x = 1/128$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/200,000$ . The maximum at  $x = 0.5$  of the solution for  $t = 0.01$  has a value of 0.03548.

In Figure 3.7, the steady-state deviation has a clear maximum which is constant with time at  $x = 0.5$ . This is an unfortunate result: it means that the implementation doesn't converge to the steady-state solution at this specific location. In the vicinity of the boundaries, the steady-state deviation decreases with time, which does show the convergence of the solutions. The grid size is an important factor for the steady-state deviation: while keeping  $\epsilon$  constant, a finer grid makes the deviation at  $x = 0.5$  have a lower maximum although it is still a maximum that doesn't change with time. In Figure 3.8, this is shown for  $\Delta x = 1/256$  and  $\epsilon = 1/128$ .

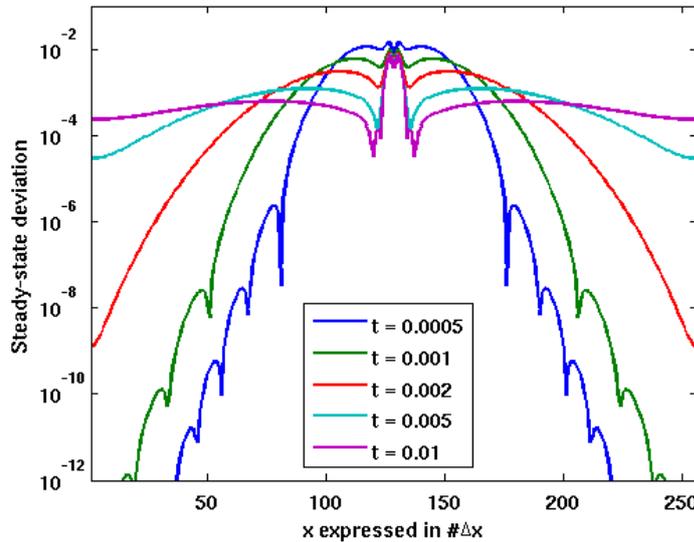


Figure 3.8: The steady-state deviation of the implementation with  $\Delta x = 1/256$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/2,500,000$ . The maximum at  $x = 0.5$  is lower than with  $\Delta x = 1/128$  (0.007425).

### 3.5.2 90%-interval

In Section 3.4, the steepness of the plots and the width of the interface was discussed qualitatively. In this subsection, the interface width and the slope is discussed quantitatively. The 90%-interval is a good quantity to define concerning the analysis of the interface width and so the steepness of the interface. The 90%-interval is defined as the interval where  $-0.9 \leq c \leq 0.9$  holds. Since the experimental data consists of discretized points, the 90%-interval is also of a discretized form. Concluding from these results, the 90%-interval increases with increasing  $\epsilon$ , which follows from the fact that  $\epsilon$  scales with the interface width and vice versa. Table 3.2 shows the width of the 90%-intervals for different  $\epsilon$  and  $\Delta x$ .

Table 3.2: The width of the 90%-interval is proportional to  $\epsilon$ . With a finer grid, one can determine the 90%-interval more precisely, as can be seen with  $\epsilon = 1/32$  and  $\epsilon = 1/16$ .

$\Delta x$	$\epsilon$	Interval width	$\Delta x$	$\epsilon$	Interval width	$\Delta x$	$\epsilon$	Interval width
$1/32$	$1/32$	$3/32$	$1/64$	$1/64$	$3/64$	$1/128$	$1/128$	$3/128$
$1/32$	$1/16$	$7/32$	$1/64$	$1/32$	$7/64$	$1/128$	$1/64$	$7/128$
			$1/64$	$1/16$	$15/64$	$1/128$	$1/32$	$15/128$
						$1/128$	$1/16$	$33/128$

### 3.5.3 Interface overshoot

In Figures 3.3 and 3.4, close to the interface, all plots except the steady-state solution have maxima and minima that are larger than  $c = 1$  and smaller than  $c = -1$ , respectively. These extreme values will be referred to as overshoot: the solutions exceed their maximum and minimum allowed values. According to Section 1.2, this is against the mass conservation law. Therefore, it can cause errors in the results. The overshoot is easily seen in Figure 3.4, but not in Figure 3.3. Therefore, Figure 3.9 shows a close-up of the latter figure.

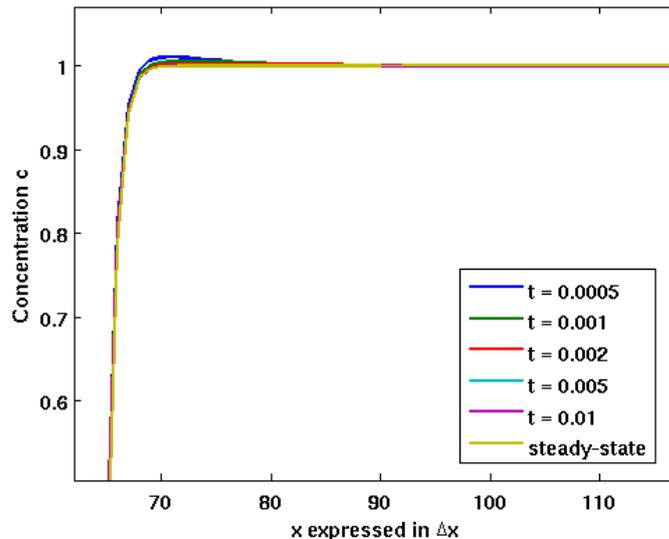


Figure 3.9: The close-up of Figure 3.3 shows the overshoot near the interface. It also shows that the overshoot decreases with time.

The overshoot is defined for  $x > 0.5$ , the domain  $x < 0.5$  is not taken into account because the interface is symmetrical:

$$\text{Overshoot}(x, t) = c(x, t) - 1, \quad x > 0.5; \quad (3.23)$$

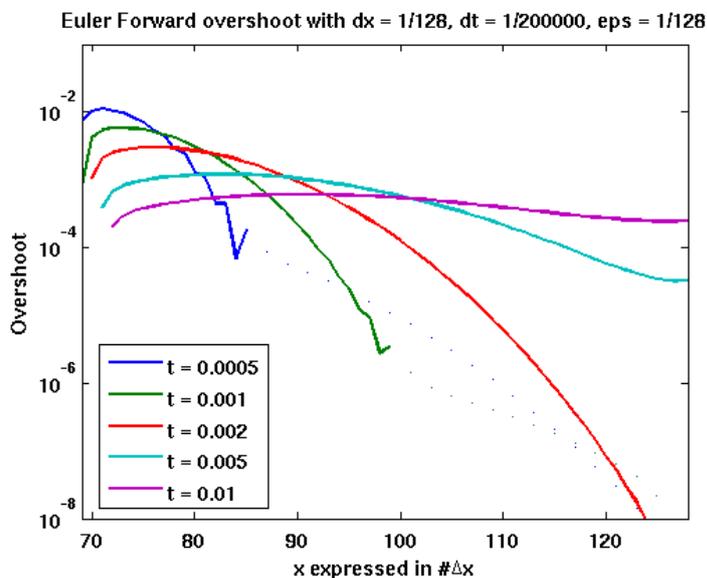


Figure 3.10: The interface overshoot from Figure 3.3 and the close-up Figure 3.9, with  $\Delta x = 1/128$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/200,000$ , using the domain  $x > 0.5$ . The maximum overshoot of the plot of  $t = 0.0005$  is 0.01118.

Figure 3.10 shows that the overshoot has a global maximum slightly to the right of  $x = 0.5$ . It also shows that, for the lowest two values of  $t$ , the plot partially consists of dots instead of a continuous line. This indicates that those solutions oscillate around  $c = 1$  (the logarithmic scale can't handle negative values). Although the maximum overshoot dampens with time, it increases with increasing  $\epsilon$ . Figure 3.11 shows this for  $\epsilon = 4/128$ .

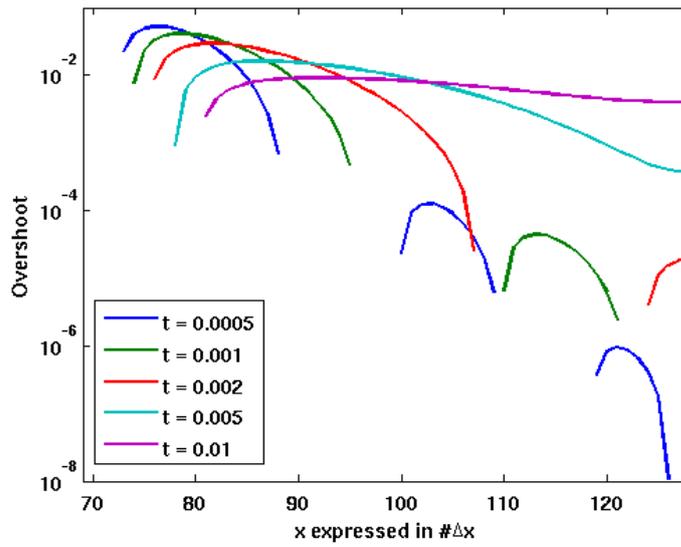


Figure 3.11: The interface overshoot is shown for the implementation with  $\Delta x = 1/128$ ,  $\epsilon = 4/128$  and  $\Delta t = 1/2.200.000$ . The maxima are significantly higher than with  $\epsilon = 1/128$  in Figure 3.10. The maximum overshoot of the plot of  $t = 0.0005$  is 0.05288.

## 4 Implicit methods for modeling the Cahn-Hilliard equation

In previous chapters, it has been mentioned that the Cahn-Hilliard equation is a nonlinear fourth-order partial differential equation. In implicit implementing, the nonlinearity causes a lot of difficulties. For instance, the method that has been discussed in Section 2.3, the Euler Backward scheme, is not suitable for discretizing a nonlinear differential equation, according to (Vuik et al., 2006, p. 66-67). If one is to implement the Cahn-Hilliard equation implicitly, iterative methods, meant to solve nonlinear differential equations, have to be used. In this chapter, the implementation of two different semi-implicit discretization schemes is described, both of which are developed during this research. Next, a fully implicit, iterative discretization scheme, involving the use of the Gâteaux derivative, is discussed. All advantages and disadvantages are discussed accordingly.

### 4.1 The semi-implicit scheme of Implicit-Explicit (ImEx)

The first semi-implicit discretization scheme that is used, is the Implicit-Explicit (ImEx) scheme. The ImEx scheme uses a combination of an explicit and an implicit discretization. The nonlinear part of the Cahn-Hilliard equation is discretized using Euler Forward, and the linear part is discretized using Euler Backward. The method has a combination of advantages and disadvantages of both methods. First, the stability condition that has to be satisfied is less severe than with the Euler Forward method, but more severe than with the Euler Backward method. Second, the processing time is partially from a matrix calculation and partially from an iteration over the grid points. Later, the quantitative results of this method, regarding the steady-state deviation, the 90%-interval and the interface overshoot, are discussed.

#### 4.1.1 ImEx discretization

The discretization of the ImEx scheme in Equation (4.1) is similar to the Euler Backward discretization with the notable difference that the nonlinear part  $(c_i^n)^3$  is discretized using an Euler Forward scheme.

$$\begin{aligned} \frac{c_i^{n+1} - c_i^n}{\Delta t} &= \frac{\mu_{i-1}^{(n,n+1)} - 2\mu_i^{(n,n+1)} + \mu_{i+1}^{(n,n+1)}}{(\Delta x)^2} \\ &= \frac{(c_{i-1}^n)^3 - 2(c_i^n)^3 + (c_{i+1}^n)^3}{(\Delta x)^2} - \frac{c_{i-1}^{n+1} - 2c_i^{n+1} + c_{i+1}^{n+1}}{(\Delta x)^2} \\ &\quad - \epsilon^2 \left( \frac{c_{i-2}^{n+1} - 4c_{i-1}^{n+1} + 6c_i^{n+1} - 4c_{i+1}^{n+1} + c_{i+2}^{n+1}}{(\Delta x)^4} \right); \end{aligned} \quad (4.1)$$

To convert the discretization in Equation (4.1) to a form, which can be implemented in Matlab, all terms involving time step  $n + 1$  are collected together and all remaining terms, involving time step  $n$  as well:

$$\begin{aligned} &\frac{c_i^{n+1}}{\Delta t} + \frac{c_{i-1}^{n+1} - 2c_i^{n+1} + c_{i+1}^{n+1}}{(\Delta x)^2} + \epsilon^2 \left( \frac{c_{i-2}^{n+1} - 4c_{i-1}^{n+1} + 6c_i^{n+1} - 4c_{i+1}^{n+1} + c_{i+2}^{n+1}}{(\Delta x)^4} \right) \\ &= \frac{c_i^n}{\Delta t} + \frac{(c_{i-1}^n)^3 - 2(c_i^n)^3 + (c_{i+1}^n)^3}{(\Delta x)^2}; \end{aligned} \quad (4.2)$$

Equation (4.2) can be rewritten as a system of equations, using a matrix  $A$  and a vector  $B$ :

$$\begin{aligned} (I + \Delta t \cdot A) c^{n+1} &= c^n + \Delta t \cdot B(c^n) \\ \Rightarrow c^{n+1} &= (I + \Delta t \cdot A)^{-1} (c^n + \Delta t \cdot B(c^n)); \end{aligned} \quad (4.3)$$

with:

$$A = \frac{1}{(\Delta x)^2} \times \begin{pmatrix} \frac{2\epsilon^2}{(\Delta x)^2} - 1 & \frac{-3\epsilon^2}{(\Delta x)^2} + 1 & \frac{\epsilon^2}{(\Delta x)^2} & 0 & \dots & 0 & 0 & 0 \\ \frac{-3\epsilon^2}{(\Delta x)^2} + 1 & \frac{6\epsilon^2}{(\Delta x)^2} - 2 & \frac{-4\epsilon^2}{(\Delta x)^2} + 1 & \frac{\epsilon^2}{(\Delta x)^2} & 0 & \dots & 0 & 0 \\ \frac{\epsilon^2}{(\Delta x)^2} & \frac{-4\epsilon^2}{(\Delta x)^2} + 1 & \frac{6\epsilon^2}{(\Delta x)^2} - 2 & \frac{-4\epsilon^2}{(\Delta x)^2} + 1 & \frac{\epsilon^2}{(\Delta x)^2} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \\ 0 & \dots & 0 & \frac{\epsilon^2}{(\Delta x)^2} & \frac{-4\epsilon^2}{(\Delta x)^2} + 1 & \frac{6\epsilon^2}{(\Delta x)^2} - 2 & \frac{-4\epsilon^2}{(\Delta x)^2} + 1 & \frac{\epsilon^2}{(\Delta x)^2} \\ 0 & 0 & \dots & 0 & \frac{\epsilon^2}{(\Delta x)^2} & \frac{-4\epsilon^2}{(\Delta x)^2} + 1 & \frac{6\epsilon^2}{(\Delta x)^2} - 2 & \frac{-3\epsilon^2}{(\Delta x)^2} + 1 \\ 0 & 0 & 0 & \dots & 0 & \frac{\epsilon^2}{(\Delta x)^2} & \frac{-3\epsilon^2}{(\Delta x)^2} + 1 & \frac{2\epsilon^2}{(\Delta x)^2} - 1 \end{pmatrix}; \quad (4.4)$$

and:

$$B(c^n) = \frac{1}{(\Delta x)^2} \begin{pmatrix} -(c_1^n)^3 + (c_2^n)^3 \\ \vdots \\ (c_{j-2}^n)^3 - 2(c_{j-1}^n)^3 + (c_j^n)^3 \\ (c_{j-1}^n)^3 - 2(c_j^n)^3 + (c_{j+1}^n)^3 \\ (c_j^n)^3 - 2(c_{j+1}^n)^3 + (c_{j+2}^n)^3 \\ \vdots \\ (c_{N-1}^n)^3 - (c_N^n)^3 \end{pmatrix}; \quad (4.5)$$

Equation (4.3) is suitable for implementation in Matlab. The boundary conditions, as described in Section 3.2, are already in  $A$  and  $B(c^n)$  and the discretized initial condition is as before:

$$c^{t=0} = \begin{cases} -1 & \text{for } 0 \leq i \leq \frac{1}{2}N \\ 1 & \text{for } \frac{1}{2}N + 1 \leq i \leq N \end{cases}; \quad (4.6)$$

This concludes the discretization of the ImEx scheme.

#### 4.1.2 stability condition

In this section, the stability condition of the ImEx discretization scheme is obtained, applying a trial-and-error approach on the numerical experiments. As mentioned before, the expectation, concerning the maximum  $\Delta t$  that follows from the stability, is that it is somewhere between the Euler Forward maximum and Euler Backward maximum. Since the Euler Backward stability condition actually states that any solution using a positive time step is stable, it is expected that the stability condition of the ImEx scheme concludes with a maximum  $\Delta t$  higher than the Euler Forward maximum. The experimental results for the stability of the ImEx scheme are in Table 4.1, together with the results of the Euler Forward scheme from Table 3.1.

$\epsilon = 0.25\Delta x$				$\epsilon = 0.5\Delta x$			
$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{ImEx}$	$\frac{\Delta t_{Euler}}{\Delta t_{ImEx}}$	$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{ImEx}$	$\frac{\Delta t_{Euler}}{\Delta t_{ImEx}}$
$1/32$	$1/4,572$	$1/7,630$	1.669	$1/32$	$1/6,055$	$1/6,101$	1.008
$1/64$	$1/18,400$	$1/30,670$	1.667	$1/64$	$1/24,510$	$1/24,560$	1.002
$1/128$	$1/73,720$	$1/122,900$	1.667	$1/128$	$1/98,280$	$1/98,320$	1.000
$\epsilon = \Delta x$				$\epsilon = 2\Delta x$			
$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{ImEx}$	$\frac{\Delta t_{Euler}}{\Delta t_{ImEx}}$	$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{ImEx}$	$\frac{\Delta t_{Euler}}{\Delta t_{ImEx}}$
$1/32$	$1/12,040$	$1/2,025$	0.168	$1/32$	$1/36,150$	$1/470$	0.0130
$1/64$	$1/48,930$	$1/8,200$	0.168	$1/64$	$1/147,000$	$1/2,015$	0.0137
$1/128$	$1/196,600$	$1/32,710$	0.166	$1/128$	$1/589,800$	$1/8,180$	0.0139
$\epsilon = 4\Delta x$				$\epsilon = 8\Delta x$			
$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{ImEx}$	$\frac{\Delta t_{Euler}}{\Delta t_{ImEx}}$	$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{ImEx}$	$\frac{\Delta t_{Euler}}{\Delta t_{ImEx}}$
$1/32$	$1/133,100$	$1/84,2$	$6.33 \cdot 10^{-4}$	$1/32$	$1/523,000$	$1/9.90$	$1.89 \cdot 10^{-5}$
$1/64$	$1/538,800$	$1/466$	$8.65 \cdot 10^{-4}$	$1/64$	$1/2,107,000$	$1/84.0$	$3.99 \cdot 10^{-5}$
$1/128$	$1/2,163,000$	$1/2,010$	$9.29 \cdot 10^{-4}$	$1/128$	$1/8,451,000$	$1/466$	$5.51 \cdot 10^{-5}$

Table 4.1: The experimental data on stability for the ImEx scheme, compared to the data for the Euler Forward scheme. The maximum  $\Delta t$  of the ImEx scheme decreases with increasing  $\epsilon$ , which is exactly opposite to the maximum  $\Delta t$  of the Euler Forward scheme.

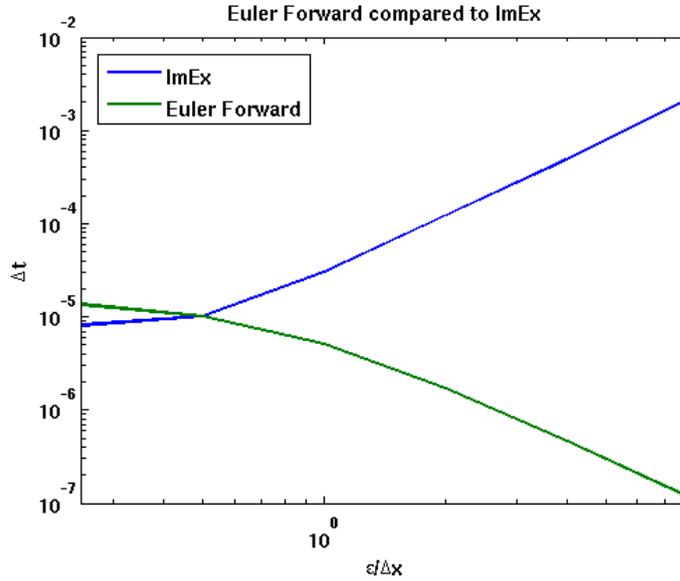


Figure 4.1: The maximum  $\Delta t$  of the ImEx scheme due to the stability condition, for  $\Delta x = 1/128$ , is shown. It decreases with increasing  $\epsilon$ , which is exactly opposite to the maximum  $\Delta t$  of the Euler Forward scheme. Only for  $\epsilon = 0.25\Delta x$  and  $\epsilon = 0.5\Delta x$ , the stability conditions for the Euler Forward scheme are favorable, for all other  $\epsilon$ , the stability condition of the ImEx scheme is.

From the data in Table 4.1, it can be concluded that for most values of  $\epsilon$ , the stability condition is less severe than the stability condition of the Euler Forward scheme. This validates the hypothesis

regarding the relation of these stability conditions. However, an interesting fact is that the maximum  $\Delta t$  due to the stability condition of the ImEx scheme increases with increasing  $\epsilon$  while the maximum  $\Delta t$  of the Euler Forward scheme decreases with increasing  $\epsilon$ . Figure 4.1 indicates that the stability condition of the ImEx scheme scales with  $(\frac{\epsilon}{\Delta x})^2$  instead of with  $(\frac{\Delta x}{\epsilon})^2$ , as was the case with the stability condition of the Euler Forward scheme in Equation (3.21).

### 4.1.3 Results

In this subsection, the evolution of the interface with time, using the ImEx discretization scheme, is discussed. Figure 4.2 shows the result for  $\frac{\epsilon}{\Delta x} = 1$ , the plot is very similar to the plot of the evolution, making use of the Euler Forward scheme, in Figure 3.3. Other values of  $\frac{\epsilon}{\Delta x}$  also result in plots similar to the ones made with the Euler Forward scheme, therefore they are not shown.

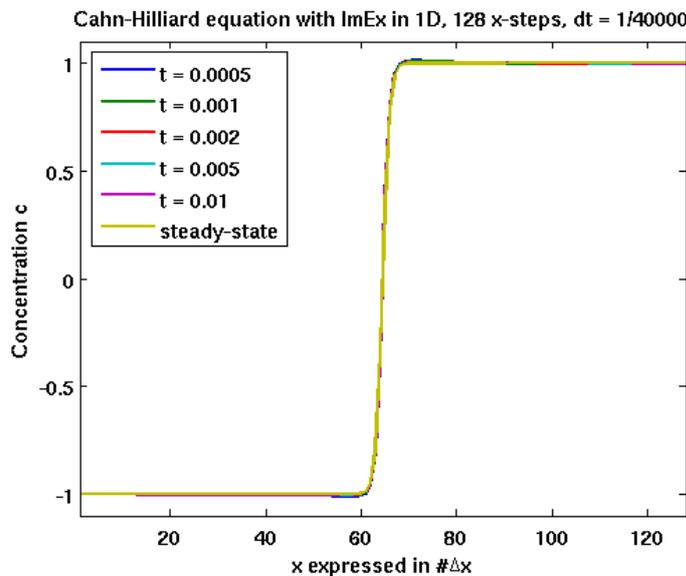


Figure 4.2: The Cahn-Hilliard equation implemented with the ImEx discretization scheme, using  $\Delta x = 1/128$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/40,000$ . This plot is very similar to the plot, using the Euler Forward scheme, in Figure 3.3.

### 4.1.4 Quantitative analysis of the results

In Subsection 4.1.3, it was concluded that the evolution of the interface using either the ImEx scheme or the Euler Forward scheme, results in the same plots. The steady-state deviation therefore also shows quantitatively similar behavior for both discretization schemes, it is shown in Figure 4.2 for  $\frac{\epsilon}{\Delta x} = 1$ . Apart from some oscillations in the vertical range smaller than approximately  $10^{-5}$ , Figure 4.2 is identical to Figure 3.7.

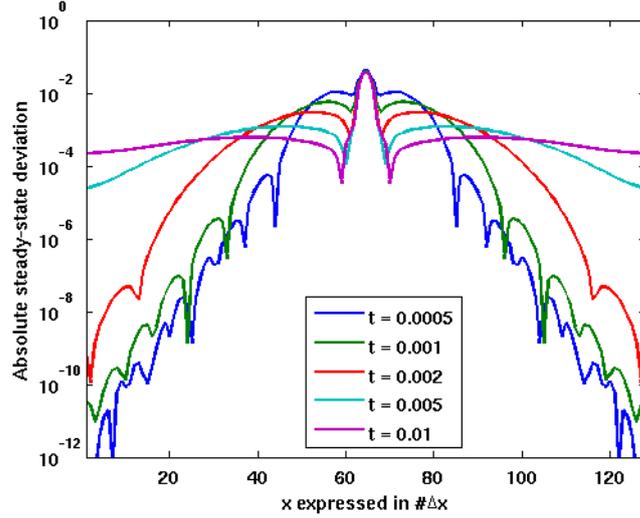


Figure 4.3: The steady-state deviation of the ImEx scheme, using  $\Delta x = 1/128$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/40,000$ . The maximum at  $x = 0.5$  of the solution for  $t = 0.01$  has a value of 0.03547. A great similarity with Figure 3.7 occurs: the ImEx scheme and the Euler Forward scheme have similar results but a different stability condition.

The analyses of the overshoot and the 90%-interval are also similar, using either one of both discretization schemes, hence, they are not repeated. It can be concluded that the developed ImEx discretization scheme is a scheme that gives the same quantitative results as the Euler Forward scheme but has a stability condition that is entirely different and, for most  $\epsilon$ , favorable.

## 4.2 The semi-implicit method of Modified Furihata

The second semi-implicit discretization scheme that is used, is a modified form of the one suggested by Furihata in (Furihata, 2000, p. 1-2). Furihata suggests a fully implicit method, which is impossible to implement in Matlab due to the nonlinear term in the Cahn-Hilliard equation. The modification is applied so that the discretization scheme becomes semi-implicit, as was the case with the ImEx scheme.

### 4.2.1 Furihata discretization

The original Furihata discretization is in Equation (4.7) and (4.8). All terms in Equation (4.8) are implicit or semi-implicit, including the nonlinear part (the term with constant  $r$ ).

$$\frac{U_k^{(n+1)} - U_k^{(n)}}{\Delta t} = \frac{1}{(\Delta x)^2} \left( \left( \frac{\partial G}{\partial U} \right)_{k-1}^{(n+1/2)} - 2 \left( \frac{\partial G}{\partial U} \right)_k^{(n+1/2)} + \left( \frac{\partial G}{\partial U} \right)_{k+1}^{(n+1/2)} \right); \quad (4.7)$$

with  $k = 0, 1, \dots, N$  and  $n = 0, 1, 2, \dots$  and:

$$\begin{aligned} \left( \frac{\partial G}{\partial U} \right)_k^{(n+1/2)} &= p \left( \frac{U_k^{(n+1)} + U_k^{(n)}}{2} \right) \\ &+ r \left( \frac{(U_k^{(n+1)})^3 + (U_k^{(n+1)})^2 U_k^{(n)} + U_k^{(n+1)} (U_k^{(n)})^2 + (U_k^{(n)})^3}{4} \right) \\ &+ \frac{q}{(\Delta x)^2} \left( \frac{U_{k-1}^{(n+1)} + U_{k-1}^{(n)} - 2(U_k^{(n+1)} + U_k^{(n)}) + U_{k+1}^{(n+1)} + U_{k+1}^{(n)}}{2} \right); \end{aligned} \quad (4.8)$$

Here,  $U$  is the concentration and  $G$  the chemical potential, which is the notation that is also used by Furihata in his publication. It is used here as well for convenience and for comparison. Furihata uses three constants, being  $p$ ,  $r$  and  $q$ , which are chosen, later, analogous to the constants in the Euler Forward or ImEx discretization scheme. Equation (4.8) is modified next so that the nonlinear term is explicit.

#### 4.2.2 Modified Furihata discretization

The discretization of Modified Furihata is:

$$\frac{U_k^{(n+1)} - U_k^{(n)}}{\Delta t} = \frac{1}{(\Delta x)^2} \left( \left( \frac{\partial G}{\partial U} \right)_{k-1}^{(n+1/2)} - 2 \left( \frac{\partial G}{\partial U} \right)_k^{(n+1/2)} + \left( \frac{\partial G}{\partial U} \right)_{k+1}^{(n+1/2)} \right); \quad (4.9)$$

with  $k = 0, 1, \dots, N$  and  $n = 0, 1, 2, \dots$  and:

$$\begin{aligned} \left( \frac{\partial G}{\partial U} \right)_k^{(n+1/2)} &= p \left( \frac{U_k^{(n+1)} + U_k^{(n)}}{2} \right) + r (U_k^{(n)})^3 \\ &+ \frac{q}{(\Delta x)^2} \left( \frac{U_{k-1}^{(n+1)} + U_{k-1}^{(n)} - 2(U_k^{(n+1)} + U_k^{(n)}) + U_{k+1}^{(n+1)} + U_{k+1}^{(n)}}{2} \right); \end{aligned} \quad (4.10)$$

The term with  $r$  is a lot simpler after this modification. It is no longer the mean of four different explicit, implicit and semi-implicit terms, but instead, it is just one explicit term. Equations (4.9) and (4.10) can be written as a system of equations:

$$\begin{aligned} (I - \Delta t \cdot A) U^{(n+1)} &= (I + \Delta t \cdot A) U^{(n)} + \Delta t \cdot B(U^{(n)}) \\ \Rightarrow U^{(n+1)} &= (I - \Delta t \cdot A)^{-1} \left( (I + \Delta t \cdot A) U^{(n)} + \Delta t \cdot B(U^{(n)}) \right); \end{aligned} \quad (4.11)$$

with:

$$A = \frac{1}{(\Delta x)^2} \times \begin{pmatrix} \frac{q}{(\Delta x)^2} - \frac{p}{2} & \frac{-3}{2(\Delta x)^2} + \frac{p}{2} & \frac{q}{2(\Delta x)^2} & 0 & \dots & 0 & 0 & 0 \\ \frac{-3}{2(\Delta x)^2} + \frac{p}{2} & \frac{3q}{(\Delta x)^2} - p & \frac{-2q}{(\Delta x)^2} + \frac{p}{2} & \frac{q}{2(\Delta x)^2} & 0 & \dots & 0 & 0 \\ \frac{q}{2(\Delta x)^2} & \frac{-2q}{(\Delta x)^2} + \frac{p}{2} & \frac{3q}{(\Delta x)^2} - p & \frac{-2q}{(\Delta x)^2} + \frac{p}{2} & \frac{q}{2(\Delta x)^2} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \\ 0 & \dots & 0 & \frac{q}{2(\Delta x)^2} & \frac{-2q}{(\Delta x)^2} + \frac{p}{2} & \frac{3q}{(\Delta x)^2} - p & \frac{-2q}{(\Delta x)^2} + \frac{p}{2} & \frac{q}{2(\Delta x)^2} \\ 0 & 0 & \dots & 0 & \frac{q}{2(\Delta x)^2} & \frac{-2q}{(\Delta x)^2} + \frac{p}{2} & \frac{3q}{(\Delta x)^2} - p & \frac{-3}{2(\Delta x)^2} + \frac{p}{2} \\ 0 & 0 & 0 & \dots & 0 & \frac{q}{2(\Delta x)^2} & \frac{-3}{2(\Delta x)^2} + \frac{p}{2} & \frac{q}{(\Delta x)^2} - \frac{p}{2} \end{pmatrix}; \quad (4.12)$$

and:

$$B(U^{(n)}) = \frac{r}{(\Delta x)^2} \begin{pmatrix} -(U_1^{(n)})^3 + (U_2^{(n)})^3 \\ \vdots \\ (U_{j-2}^{(n)})^3 - 2(U_{j-1}^{(n)})^3 + (U_j^{(n)})^3 \\ (U_{j-1}^{(n)})^3 - 2(U_j^{(n)})^3 + (U_{j+1}^{(n)})^3 \\ (U_j^{(n)})^3 - 2(U_{j+1}^{(n)})^3 + (U_{j+2}^{(n)})^3 \\ \vdots \\ (U_{N-1}^{(n)})^3 - (U_N^{(n)})^3 \end{pmatrix}; \quad (4.13)$$

with  $q = -\epsilon^2$ ,  $p = -1$  and  $r = 1$ . Equation (4.11) is suitable for implementation in Matlab. The boundary conditions are already in  $A$  and  $B(c^n)$  and the discretized initial condition is as before:

$$c^{t=0} = \begin{cases} -1 & \text{for } 0 \leq i \leq \frac{1}{2}N \\ 1 & \text{for } \frac{1}{2}N + 1 \leq i \leq N \end{cases} ; \quad (4.14)$$

This concludes the discretization of the Modified Furihata scheme.

### 4.2.3 Stability condition

As in previous sections, a trial-and-error approach is applied to the numerical experiments to obtain the stability conditions. Since the Modified Furihata discretization scheme is semi-implicit, like the ImEx scheme, the stability condition is expected to be less severe than with the Euler Forward scheme. However, at this point there is no hypothesis regarding the relation between the stability conditions of the ImEx and Modified Furihata schemes. Table 4.2 shows the results of the trial-and-error approach, using the Modified Furihata scheme. The relation between the maximum values of  $\Delta t$  for different  $\epsilon$  is again totally different from the increasing or decreasing  $\Delta t$  which were discussed in previous sections. Using the Modified Furihata scheme, the stability condition is relatively constant for most of the considered  $\epsilon$ . Changes in  $\Delta t_{MF}$  are only observed at relatively large  $\epsilon$ . This is shown, together with the stability conditions of the the Euler Forward scheme and the ImEx scheme, in Figure 4.4.

$\epsilon = 0.25\Delta x$				$\epsilon = 0.5\Delta x$			
$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{MF}$	$\frac{\Delta t_{Euler}}{\Delta t_{MF}}$	$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{MF}$	$\frac{\Delta t_{Euler}}{\Delta t_{MF}}$
$1/32$	$1/4,572$	$1/6,094$	1.333	$1/32$	$1/6,055$	$1/6,084$	1.005
$1/64$	$1/18,400$	$1/24,540$	1.334	$1/64$	$1/24,510$	$1/24,540$	1.001
$1/128$	$1/73,720$	$1/98,290$	1.333	$1/128$	$1/98,280$	$1/98,300$	1.000
$\epsilon = \Delta x$				$\epsilon = 2\Delta x$			
$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{MF}$	$\frac{\Delta t_{Euler}}{\Delta t_{MF}}$	$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{MF}$	$\frac{\Delta t_{Euler}}{\Delta t_{MF}}$
$1/32$	$1/12,040$	$1/6,080$	0.505	$1/32$	$1/36,150$	$1/6,010$	0.166
$1/64$	$1/48,930$	$1/24,550$	0.502	$1/64$	$1/147,000$	$1/24,600$	0.167
$1/128$	$1/196,600$	$1/98,400$	0.501	$1/128$	$1/589,800$	$1/98,500$	0.167
$\epsilon = 4\Delta x$				$\epsilon = 8\Delta x$			
$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{MF}$	$\frac{\Delta t_{Euler}}{\Delta t_{MF}}$	$\Delta x$	$\Delta t_{Euler}$	$\Delta t_{MF}$	$\frac{\Delta t_{Euler}}{\Delta t_{MF}}$
$1/32$	$1/133,100$	$1/5,570$	$4.18 \cdot 10^{-2}$	$1/32$	$1/523,000$	$1/2,570$	$4.91 \cdot 10^{-3}$
$1/64$	$1/538,800$	$1/24,280$	$4.51 \cdot 10^{-2}$	$1/64$	$1/2,107,000$	$1/21,650$	$1.03 \cdot 10^{-2}$
$1/128$	$1/2,163,000$	$1/97,950$	$4.53 \cdot 10^{-2}$	$1/128$	$1/8,451,000$	$1/89,500$	$1.06 \cdot 10^{-2}$

Table 4.2: The experimental data on stability for the Modified Furihata scheme, compared to the data for the Euler Forward scheme. The stability condition of the Modified Furihata scheme is relatively constant for different values of  $\epsilon$ . It is also less severe compared to the stability condition of Euler Forward, for most  $\epsilon$ .

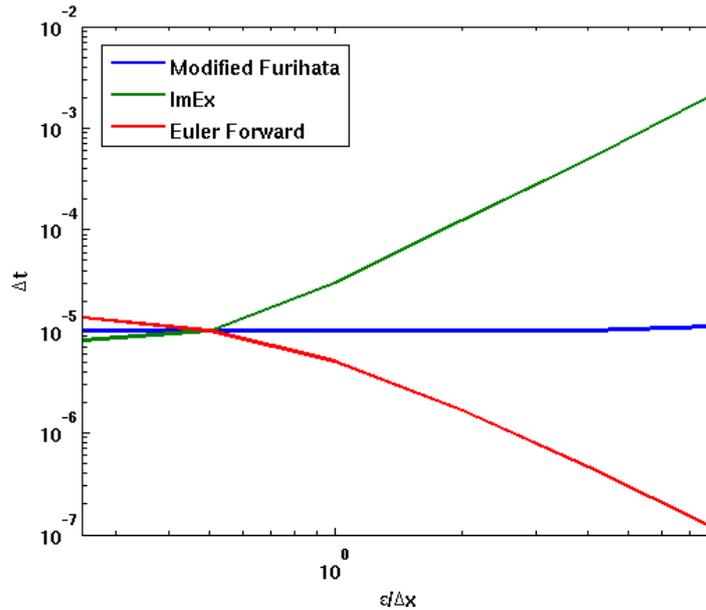


Figure 4.4: The stability conditions of the three used discretization schemes, are all totally different. The Modified Furihata scheme has a relatively constant stability condition while the ImEx and Euler Forward schemes have increasing and decreasing  $\Delta t$  for increasing  $\epsilon$ , respectively.

#### 4.2.4 Results

In this subsection, the evolution of the interface with time, using the Modified Furihata scheme, is discussed. Figure 4.5 shows the result using  $\frac{\epsilon}{\Delta x} = 1$ , a figure which has great similarity with Figure 3.3 of the evolving interface, using Euler Forward scheme and with 4.2, using the ImEx scheme. Since plots with other values of  $\frac{\epsilon}{\Delta x}$  also show great similarity with plots made using the other discretization schemes, they are not shown here.

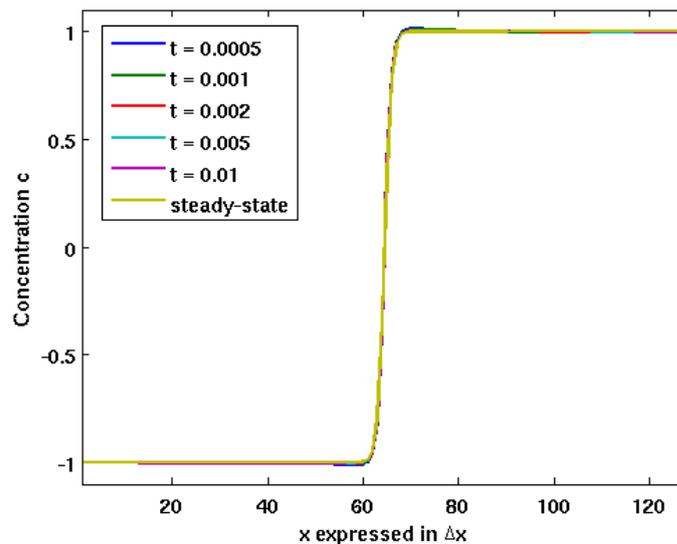


Figure 4.5: The evolving interface, implemented with the Modified Furihata scheme, using  $\Delta x = 1/128$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/110,000$ , gives the same result as the Euler Forward scheme did in Figure 3.3.

### 4.2.5 Quantitative analysis of the results

In Subsection 4.2.4, it was concluded that the results, concerning the evolving interface, are the same for all three discretizations, used in this chapter. Therefore, the steady-state deviation for different  $\epsilon$ , as well as the interface overshoot and the 90%-interval are also the same for all three discretizations. In Figure 4.6, this is shown for the steady-state deviation of the Modified Furihata scheme, using  $\frac{\epsilon}{\Delta x} = 1$ .

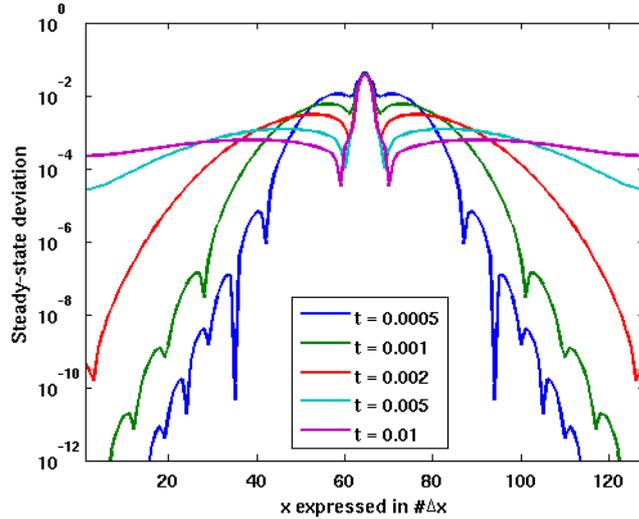


Figure 4.6: The steady-state deviation of the Modified Furihata scheme is very similar to the steady-state deviation in Figure 3.7, using the Euler Forward or in Figure 4.3, using ImEx scheme. In this graph,  $\Delta x = 1/128$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/110,000$  are used and the maximum at  $x = 0.5$  of the solution for  $t = 0.01$  has a value of 0.03548.

## 4.3 The implicit iterative method involving the Gâteaux derivative

In this section, the implicit iterative method involving the Gâteaux derivative, is discussed. This method is fully implicit, which is a step forward in stability compared to the methods that have been used in previous sections. Another difference with previous methods is that this method uses iterations within each time step, making it ideal to use for the implementation of a nonlinear differential equation, such as the Cahn-Hilliard equation. The function of the Gâteaux derivative itself is to linearize the Cahn-Hilliard equation so implicit implementing is possible. The Gâteaux derivative method is closely related to the Newton-Raphson method, in the sense that it uses a similar derivative formulation and it requires an initial guess for every time step with which the algorithm can calculate the next iteration. With that iteration, another iteration can be calculated and so on. When successive iterations have satisfied a convergence criterion, the algorithm continues with the next time step and repeats the iterations algorithm.

### 4.3.1 Algorithm of the Gâteaux derivative method

The Gâteaux derivative is used as formulated in (Dugger and Lambert, 2013). The algorithm, which is used to implement the Cahn-Hilliard equation, states the following about the Gâteaux derivative of the Cahn-Hilliard equation:

$$\frac{\partial}{\partial y}[F(c^{n+1} + sy, c^n)]_{y=0} = -F(c^{n+1}, c^n); \quad (4.15)$$

Here,  $y$  is the parameter with respect to which the derivative is taken, in the Gâteaux derivative,  $F(c^{n+1}, c^n)$  is the Cahn-Hilliard equation in a representation with all terms on one side:

$$F(c^{n+1}, c^n) = \frac{c^{n+1} - c^n}{\Delta t} - \frac{\partial^2}{\partial x^2} \left[ (c^{n+1})^3 - c^{n+1} - \epsilon^2 \frac{\partial^2 c^{n+1}}{\partial x^2} \right] = 0; \quad (4.16)$$

and  $s$  is defined as:

$$s = c_{p+1}^{n+1} - c_p^{n+1}; \quad (4.17)$$

Here,  $p+1$  is the indicator for the next iteration. Since information about iteration  $p+1$  is not at hand in iteration  $p$ ,  $s$  is unknown, initially. In equations (4.15), (4.16) and (4.17), the discretization with respect to  $n$  is already implemented. The discretization with respect to  $x$  is implemented later. Substituting Equation (4.16) in Equation (4.15) gives:

$$\begin{aligned} & \frac{\partial}{\partial y} \left[ \frac{c^{n+1} + sy - c^n}{\Delta t} - \frac{\partial^2}{\partial x^2} \left[ (c^{n+1} + sy)^3 - (c^{n+1} + sy) - \epsilon^2 \frac{\partial^2 (c^{n+1} + sy)}{\partial x^2} \right] \right]_{y=0} \\ &= -\frac{c^{n+1} - c^n}{\Delta t} + \frac{\partial^2}{\partial x^2} \left[ (c^{n+1})^3 - c^{n+1} - \epsilon^2 \frac{\partial^2 c^{n+1}}{\partial x^2} \right]; \end{aligned} \quad (4.18)$$

In Equation (4.18),  $c$  nor  $s$  is  $y$ -dependent, by the definition of the Gâteaux derivative. The derivatives with respect to  $y$  and with respect to  $x$  can be swapped so that  $\frac{\partial}{\partial y}$  can be evaluated:

$$\left[ \frac{s}{\Delta t} - \frac{\partial^2}{\partial x^2} \left[ 3s(c^{n+1} + sy)^2 - s - \epsilon^2 \frac{\partial^2 s}{\partial x^2} \right] \right]_{y=0} = -\frac{c^{n+1} - c^n}{\Delta t} + \frac{\partial^2}{\partial x^2} \left[ (c^{n+1})^3 - c^{n+1} - \epsilon^2 \frac{\partial^2 c^{n+1}}{\partial x^2} \right]; \quad (4.19)$$

Substituting  $y = 0$ :

$$\frac{s}{\Delta t} - \frac{\partial^2}{\partial x^2} \left[ 3s(c^{n+1})^2 - s - \epsilon^2 \frac{\partial^2 s}{\partial x^2} \right] = -\frac{c^{n+1} - c^n}{\Delta t} + \frac{\partial^2}{\partial x^2} \left[ (c^{n+1})^3 - c^{n+1} - \epsilon^2 \frac{\partial^2 c^{n+1}}{\partial x^2} \right]; \quad (4.20)$$

By the definition of the Gâteaux derivative, Equation (4.20) is linear in terms of  $s$ . A discretization for  $x$  can now be applied so that  $s$  can be calculated. As has been discussed in Section 3.1, two different ways of discretizing  $\frac{\partial^2}{\partial x^2}$  can be applied. Either the central difference discretization is applied directly to  $\frac{\partial^2}{\partial x^2}$  or the second derivative is evaluated before discretizing a new expression of Equation (4.20). Here, both discretization methods are applied and conclusions are drawn accordingly.

### 4.3.2 Direct central difference discretization

Equation (4.20) can be discretized directly, using an indicator  $i$  for the  $x$ -discretization:

$$\begin{aligned} & \frac{s}{\Delta t} - \left[ (3(c^{n+1})^2 - 1) \left( \frac{s_{i-1} - 2s_i + s_{i+1}}{(\Delta x)^2} \right) - \epsilon^2 \left( \frac{c_{i-2}^{n+1} - 4c_{i-1}^{n+1} + 6c_i^{n+1} - 4c_{i+1}^{n+1} + c_{i+2}^{n+1}}{(\Delta x)^4} \right) \right] \\ &= -\frac{c^{n+1} - c^n}{\Delta t} + \left[ \frac{(c_{i-1}^{n+1})^3 - 2(c_i^{n+1})^3 + (c_{i+1}^{n+1})^3}{(\Delta x)^2} - \frac{c_{i-1}^{n+1} - 2c_i^{n+1} + c_{i+1}^{n+1}}{(\Delta x)^2} \right] \\ & \quad + \left[ -\epsilon^2 \left( \frac{c_{i-2}^{n+1} - 4c_{i-1}^{n+1} + 6c_i^{n+1} - 4c_{i+1}^{n+1} + c_{i+2}^{n+1}}{(\Delta x)^4} \right) \right]; \end{aligned} \quad (4.21)$$

Equation (4.21) can be written as a system of equations for all  $s_i$ , the right side is a constant in terms of  $s_i$ :

$$\begin{aligned} & \frac{s}{\Delta t} - A \cdot s = -\frac{c^{n+1} - c^n}{\Delta t} + B(c^{n+1}) \\ \Leftrightarrow & \left( \frac{1}{\Delta t} I - A \right) \cdot s = -\frac{c^{n+1} - c^n}{\Delta t} + B(c^{n+1}) \\ \Leftrightarrow & s = -\left( \frac{1}{\Delta t} I - A \right)^{-1} \left( \frac{c^{n+1} - c^n}{\Delta t} + B(c^{n+1}) \right); \end{aligned} \quad (4.22)$$

with:

$$A = \frac{1}{(\Delta x)^2} \times \begin{pmatrix} \frac{-2\epsilon^2}{(\Delta x)^2} - (3(c^{n+1})^2 - 1) & \frac{3\epsilon^2}{(\Delta x)^2} + (3(c^{n+1})^2 - 1) & \frac{-\epsilon^2}{(\Delta x)^2} & 0 & \dots \\ \frac{3\epsilon^2}{(\Delta x)^2} + (3(c^{n+1})^2 - 1) & \frac{-6\epsilon^2}{(\Delta x)^2} - (6(c^{n+1})^2 - 2) & \frac{4\epsilon^2}{(\Delta x)^2} + (3(c^{n+1})^2 - 1) & \frac{-\epsilon^2}{(\Delta x)^2} & \ddots \\ \frac{-\epsilon^2}{(\Delta x)^2} & \frac{4\epsilon^2}{(\Delta x)^2} + (3(c^{n+1})^2 - 1) & \frac{-6\epsilon^2}{(\Delta x)^2} - (6(c^{n+1})^2 - 2) & \frac{4\epsilon^2}{(\Delta x)^2} + (3(c^{n+1})^2 - 1) & \ddots \\ \vdots & \frac{-\epsilon^2}{(\Delta x)^2} & \frac{4\epsilon^2}{(\Delta x)^2} + (3(c^{n+1})^2 - 1) & \frac{-6\epsilon^2}{(\Delta x)^2} - 2(3(c^{n+1})^2 - 1) & \ddots \\ 0 & \ddots & \frac{-\epsilon^2}{(\Delta x)^2} & \frac{4\epsilon^2}{(\Delta x)^2} + (3(c^{n+1})^2 - 1) & \ddots \\ 0 & 0 & \ddots & \frac{-\epsilon^2}{(\Delta x)^2} & \ddots \\ 0 & 0 & 0 & \ddots & \ddots \end{pmatrix}; \quad (4.23)$$

Here, only the upperleft corner of  $A$  is shown, due to its size. The vector  $B(c^{n+1})$  is defined as:

$$B(c^{n+1}) = \frac{1}{(\Delta x)^2} \times \begin{pmatrix} -(c_1^{n+1})^3 + (c_2^{n+1})^3 - (c_1^{n+1} + c_2^{n+1}) - \epsilon^2 \left( \frac{2c_1^{n+1} - 3c_2^{n+1} + c_3^{n+1}}{(\Delta x)^2} \right) \\ \vdots \\ (c_{i-2}^{n+1})^3 - 2(c_{i-1}^{n+1})^3 + (c_i^{n+1})^3 - (c_{i-2}^{n+1} - 2c_{i-1}^{n+1} + c_i^{n+1}) - \epsilon^2 \left( \frac{c_{i-3}^{n+1} - 4c_{i-2}^{n+1} + 6c_{i-1}^{n+1} - 4c_i^{n+1} + c_{i+1}^{n+1}}{(\Delta x)^2} \right) \\ (c_{i-1}^{n+1})^3 - 2(c_i^{n+1})^3 + (c_{i+1}^{n+1})^3 - (c_{i-1}^{n+1} - 2c_i^{n+1} + c_{i+1}^{n+1}) - \epsilon^2 \left( \frac{c_{i-2}^{n+1} - 4c_{i-1}^{n+1} + 6c_i^{n+1} - 4c_{i+1}^{n+1} + c_{i+2}^{n+1}}{(\Delta x)^2} \right) \\ (c_i^{n+1})^3 - 2(c_{i+1}^{n+1})^3 + (c_{i+2}^{n+1})^3 - (c_i^{n+1} - 2c_{i+1}^{n+1} + c_{i+2}^{n+1}) - \epsilon^2 \left( \frac{c_{i-1}^{n+1} - 4c_i^{n+1} + 6c_{i+1}^{n+1} - 4c_{i+2}^{n+1} + c_{i+3}^{n+1}}{(\Delta x)^2} \right) \\ \vdots \\ (c_{N-1}^{n+1})^3 - (c_N^{n+1})^3 - (c_{N-1}^{n+1} - c_N^{n+1}) - \epsilon^2 \left( \frac{c_{N-2}^{n+1} - 3c_{N-1}^{n+1} + 2c_N^{n+1}}{(\Delta x)^2} \right) \end{pmatrix}; \quad (4.24)$$

Substituting a rewritten version of the definition of  $s$  from Equation (4.17) in Equation (4.22):

$$c_{p+1}^{n+1} = c_p^{n+1} + s = c_p^{n+1} - \left( \frac{1}{\Delta t} I - A \right)^{-1} \left( \frac{c^{n+1} - c^n}{\Delta t} + B(c^{n+1}) \right); \quad (4.25)$$

With Equation (4.25), the algorithm is completed for one iteration. The number of iterations that is used is determined by how fast the algorithm converges to a solution within one time step. The stop condition for an iteration involves the root mean square and some tolerance  $\psi$ :

$$\sqrt{\sum_{i=1}^N \frac{1}{N} (c_{p+1}^{n+1} - c_p^{n+1})^2} < \psi; \quad (4.26)$$

The value of  $\psi$ , which is important for the accuracy of the solution, is given later. If the stop condition is satisfied, the algorithm moves on to the next iteration. For every iteration, an initial condition,  $c_{p=0}^{n+1}$ , is needed, which is chosen to be the result of the previous time step  $c^n$ . This concludes the discretization and the algorithm.

### 4.3.3 Indirect central difference discretization

Equation (4.20) can be rewritten as:

$$\begin{aligned} & \frac{s}{\Delta t} - \left[ 6 \left( \left( \frac{\partial c^{n+1}}{\partial x} \right)^2 + c^{n+1} \frac{\partial^2 c^{n+1}}{\partial x^2} \right) s + 12 \left( c^{n+1} \frac{\partial c^{n+1}}{\partial x} \right) \frac{\partial s}{\partial x} \right] \\ & + \left[ (3(c^{n+1})^2 - 1) \frac{\partial^2 s}{\partial x^2} - \epsilon^2 \frac{\partial^4 s}{\partial x^4} \right] \\ & = -\frac{c^{n+1} - c^n}{\Delta t} + \left[ 6c^{n+1} \left( \frac{\partial c^{n+1}}{\partial x} \right)^2 + ((3c^{n+1})^2 - 1) \frac{\partial^2 c^{n+1}}{\partial x^2} - \epsilon^2 \frac{\partial^4 c^{n+1}}{\partial x^4} \right]; \end{aligned} \quad (4.27)$$

To Equation (4.27), the  $x$ -discretization, using an indicator  $i$ , can be applied:

$$\begin{aligned} & \frac{s_i}{\Delta t} - \left[ 6 \left( \left( \frac{c_i^{n+1} - c_{i-1}^{n+1}}{\Delta x} \right)^2 - c_i^{n+1} \left( \frac{c_{i-1}^{n+1} - 2c_i^{n+1} + c_{i+1}^{n+1}}{(\Delta x)^2} \right) \right) s_i \right] \\ & - \left[ 12c_i^{n+1} \left( \frac{c_i^{n+1} - c_{i-1}^{n+1}}{\Delta x} \right) \left( \frac{s_i - s_{i-1}}{\Delta x} \right) + (3(c_i^{n+1})^2 - 1) \left( \frac{s_{i-1} - 2s_i + s_{i+1}}{(\Delta x)^2} \right) \right] \\ & - \left[ -\epsilon^2 \left( \frac{s_{i-2} - 4s_{i-1} + 6s_i - 4s_{i+1} + s_{i+2}}{(\Delta x)^4} \right) \right] \\ & = -\frac{c_i^{n+1} - c_i^n}{\Delta t} + \left[ 6c_i^{n+1} \left( \frac{c_i^{n+1} - c_{i-1}^{n+1}}{\Delta x} \right)^2 + (3(c_i^{n+1})^2 - 1) \left( \frac{c_{i-1}^{n+1} - 2c_i^{n+1} + c_{i+1}^{n+1}}{(\Delta x)^2} \right) \right] \\ & + \left[ -\epsilon^2 \left( \frac{c_{i-2}^{n+1} - 4c_{i-1}^{n+1} + 6c_i^{n+1} - 4c_{i+1}^{n+1} + c_{i+2}^{n+1}}{(\Delta x)^4} \right) \right]; \end{aligned} \quad (4.28)$$

Equation (4.28) can again be written as a system of equations for all  $s_i$ , like in Equation (4.22):

$$s = -\left( \frac{1}{\Delta t} I - A \right)^{-1} \left( \frac{c^{n+1} - c^n}{\Delta t} + B(c^{n+1}) \right); \quad (4.29)$$

Here,  $A$  and  $B(c^{n+1})$  are analogous to Equation (4.23) and Equation (4.24), respectively, but with the matrix coefficients that follow from Equation (4.28). Again, substituting the rewritten version of the definition of  $s$  from Equation (4.17) in Equation (4.29):

$$c_{p+1}^{n+1} = c_p^{n+1} + s = c_p^{n+1} - \left( \frac{1}{\Delta t} I - A \right)^{-1} \left( \frac{c^{n+1} - c^n}{\Delta t} + B(c^{n+1}) \right); \quad (4.30)$$

With Equation (4.30), the algorithm is completed again. The stop condition is the same as in (4.26) and the initial condition for the next iteration is  $c_{p=0}^{n+1} = c^n$ . This concludes the discretization and the algorithm.

### 4.3.4 Results

In this section, the results of implementing the discretizations of Subsections 4.3.2 and 4.3.3 are discussed. Other than the results for the ImEx and Modified Furihata schemes in Sections 4.1 and 4.2, respectively, the results for the evolving interface are not as desired here. In Figures 4.7 and 4.8, the evolving interface is shown for the discretizations in Subsection 4.3.2 and 4.3.3, respectively.

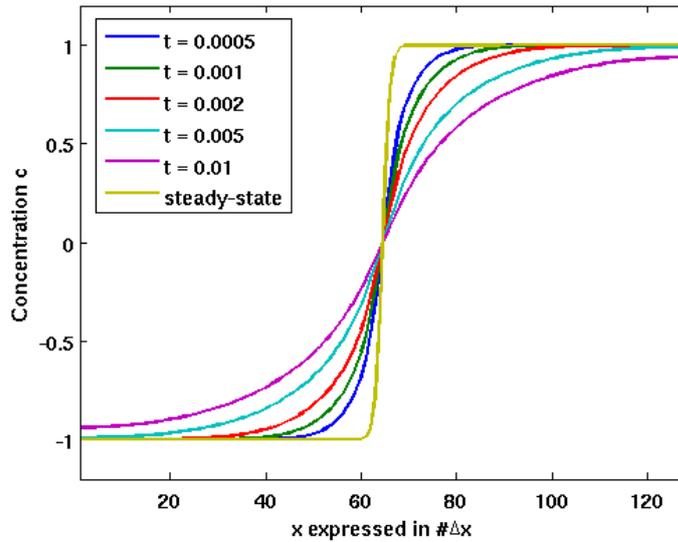


Figure 4.7: The Cahn-Hilliard equation as implemented using the discretization from Subsection 4.3.2,  $\Delta x = 1/128$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/30,000$  and the stop condition is  $\psi = 1/1,000$ . The solutions are not as desired since they dampen with time and don't converge to the theoretical steady-state solution.

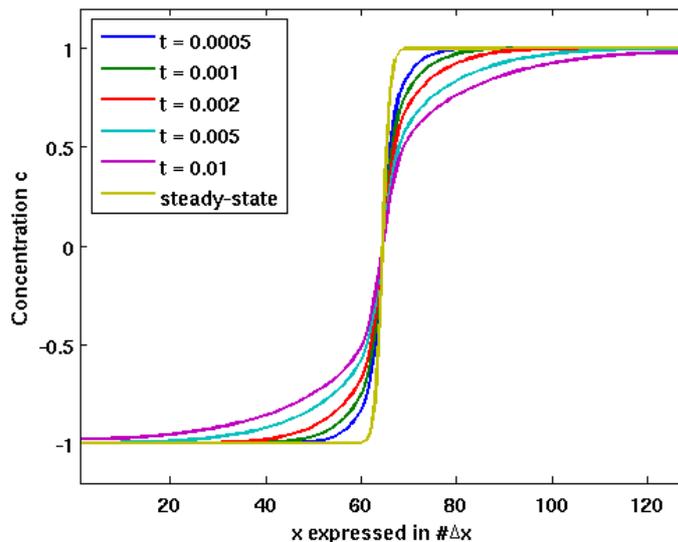


Figure 4.8: The Cahn-Hilliard equation as implemented using the discretization from Subsection 4.3.3,  $\Delta x = 1/128$ ,  $\epsilon = 1/128$  and  $\Delta t = 1/30,000$  and the stop condition is  $\psi = 1/10,000$ . The solutions dampen slower than in Figure 4.7 but they still do. The solutions don't converge to the steady-state solution.

#### 4.3.5 Qualitative analysis of the Gâteaux derivative method

In Subsection 4.3.4, it is shown that the used iterative method involving the Gâteaux derivative didn't work for two different discretizations. In both cases, as can be seen in Figures 4.7 and 4.8, the solutions dampen with time and don't converge to the theoretical steady-state solution. Instead, the solutions converge to the line  $c = 0$ , they behave like positive diffusion solutions. Figure 4.9 shows this for the discretization in Section 4.3.2 for several larger values of  $t$ .

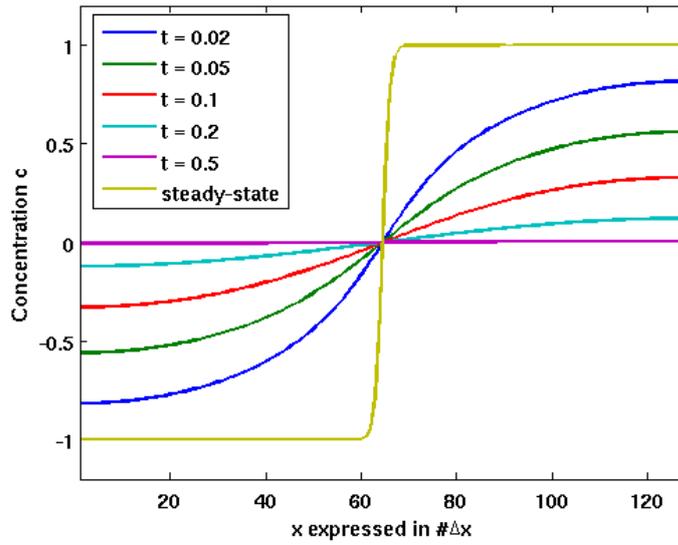


Figure 4.9: The Cahn-Hilliard equation as implemented using the same discretization and parameter values as used for Figure 4.7. The large values of  $t$  show convergence to the line  $c = 0$ .

It can be concluded that both the direct and the indirect discretizations from Subsections 4.3.2 and 4.3.3, respectively, do not give desired results. Extensive research and debugging has been done to find out why. Using the indirect discretization, making a specific term zero, might give an answer to the question which term causes the positive diffusion. This has been done with, for example, the terms  $6 \left( \left( \frac{\partial c^{n+1}}{\partial x} \right)^2 + c^{n+1} \frac{\partial^2 c^{n+1}}{\partial x^2} \right) = 0$ ,  $12 \left( c^{n+1} \frac{\partial c^{n+1}}{\partial x} \right) = 0$ ,  $(3(c^{n+1})^2 - 1) = 0$  and all possible combinations of making several terms zero and several nonzero. Unfortunately, the precise reason for the positive diffusive behavior was not found. It is presumed that negative diffusive behavior only occurs in a specific range of  $c$ , as has been discussed in Section 3.1. Only for some  $c$ , the second order derivative of  $c$  (or of  $s$ , as is the case with the Gâteaux derivative method) has a minus sign. Therefore, a mistake with a minus sign can be the reason that the direction of the diffusion is wrong.

## 5 The spreading droplet

In this chapter, the implementation, validation and the results of different parametric studies on the dynamic boundary condition for a moving contact line, are discussed. This is done using the spreading droplet case, which is a 2-dimensional convection-diffusion problem that is solved using the coupled Cahn-Hilliard-Navier-Stokes equation. The theory regarding the dynamic boundary condition and the spreading droplet, has been discussed briefly in theory Section 1.5. First, the implementation of the boundary condition withing the OpenFOAM framework, is discussed. Next, a validation study is done in which the implementation is compared to the expected behavior of a known system from literature. Finally, parametric studies are done, in which the behavior of the spreading droplet is considered with varying surface tension, droplet interface width and viscosity.

The dynamic boundary condition is stated in Equation (1.20). For convenience, it is repeated here:

$$D_w \left( \frac{\partial \phi}{\partial t} + u \nabla \phi \right) = -\nabla \phi - \frac{\phi(\phi - 1) \cos(\theta)}{\epsilon \sqrt{2}}; \quad (5.1)$$

Here,  $\phi$  is analogous to the concentration  $c$ ,  $u$  is the velocity of the fluid at the solid surface,  $\theta$  is the dynamic contact angle,  $D_w$  is analogous to a friction factor at the contact line between the fluid and the solid and  $\epsilon$  is a phenomological parameter.

### 5.1 Implementation in OpenFOAM

This section is removed intentionally due to publishing reasons of my supervisor M.Sc. H.G. PIMPALGAONKAR.

### 5.2 Validation of the implementation

In Section 5.1, the dynamic boundary condition was discretized so that the implementation in OpenFOAM is possible. In this section, the case that is implemented is discussed more thoroughly than before and the results are qualitatively compared to the data from (Bird et al., 2008).

The initial condition of the spreading droplet was already shown in Figure 1.3. In Figure 5.1, the spreading droplet is shown for  $t > 0$  s. The droplet sits on a solid surface along which it is spreading. On the left and on the right side of the figure, two solid walls are placed. The upper limit of the computational domain is modeled as atmosphere. The spreading radius,  $r$ , is an important indication of the status of the process of spreading. It is defined as the distance along the liquid-solid contact line from the center of the droplet to the liquid-gas interface, where  $\phi = 0$ , on the left or the right side, as shown in Figure 5.1. The rate of spreading is determined by  $D_w$  from Equation (5.1), by the contact angle in steady-state,  $\theta_s$ , and by the surface tension between the liquid and the gas,  $\sigma$ . The spreading radius as a function of time is used as a qualitative analysis function for the droplet spreading during the rest of this chapter.

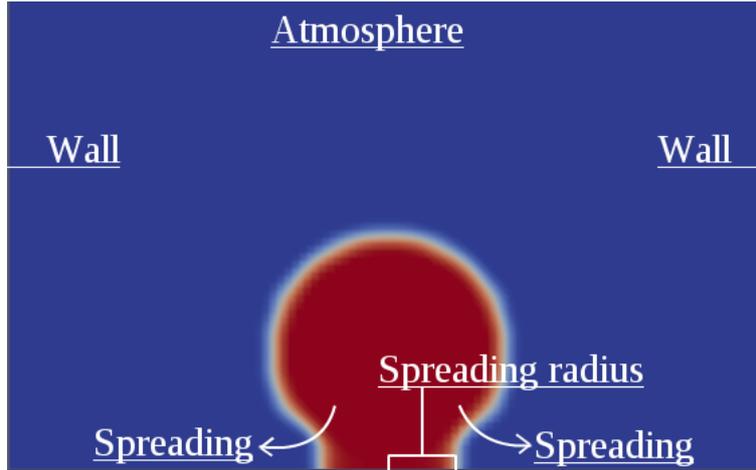


Figure 5.1: The spreading of a droplet with initial radius  $L = 7.8 \cdot 10^{-4}$  m and initial contact angle  $\theta = 180.0^\circ$  as modeled using OpenFOAM with  $D_w = 2.0 \text{ kg m}^{-1}\text{s}^{-1}$ ,  $\epsilon = 0.6 \cdot 10^{-4}$  m,  $\theta_s = 3.0^\circ$ ,  $\sigma = 0.07 \text{ J m}^{-2}$ . The spreading radius here, at  $t = 2.5 \cdot 10^{-4}$  s, is  $r = 4.47 \cdot 10^{-4}$  m. The spreading continues in the direction of the arrows.

The actual experimental data is available from (Bird et al., 2008) and, initially, it was planned to use this data to compare with the results obtained here, rather than discussing the results qualitatively. However, it was realized very late that the system being modeled here is not the same as the system in the literature. One of the reasons for this is that the spherical droplet in the initial condition could never have been the initial condition of the experiments, the droplet was deposited on the surface by, for example, a pipette. This is not implemented here. More efforts are needed to be able to compare the results of this research with the literature data, hence, the analyses here are strictly qualitative.

Figures 5.2, 5.3, 5.4 show the spreading radii for the validation cases with  $\theta_s = 3.0^\circ$ ,  $\theta_s = 43.0^\circ$  and  $\theta_s = 117.0^\circ$ , respectively. In each individual plot, solutions with various values of  $D_w$  are shown. For all figures, the spreading radius has a strange offset on the vertical axis, making it around  $r = 0.2 \cdot 10^{-3}$  m initially. It would make sense if the spreading radius would start from  $r = 0$ , since this is the value of the spreading radius in the initial condition, but it doesn't.

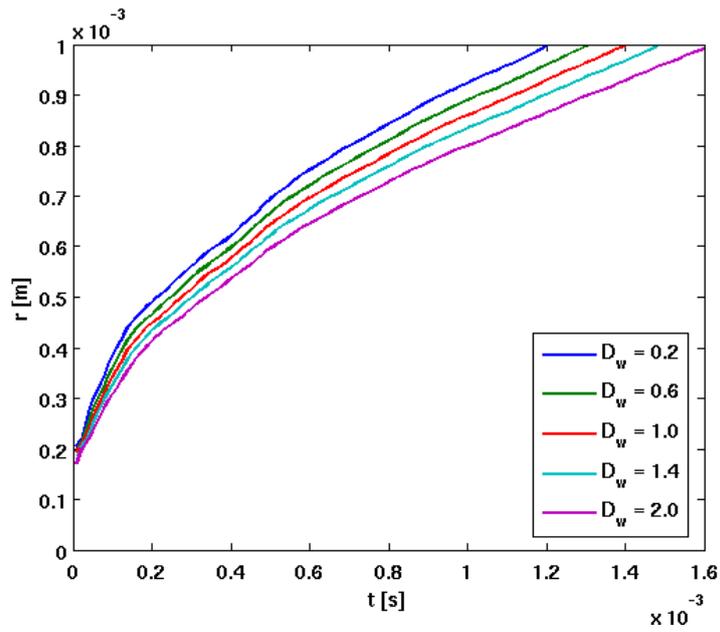


Figure 5.2: The spreading radius as a function of time for  $\theta_s = 3.0^\circ$ ,  $\epsilon = 6.0 \cdot 10^{-5}$  m,  $\sigma = 0.07$  J m $^{-2}$  and various values of  $D_w$ . The offset of  $r = 0.2 \cdot 10^{-3}$  m on the left does not belong there. However, it doesn't stop the qualitative analysis.

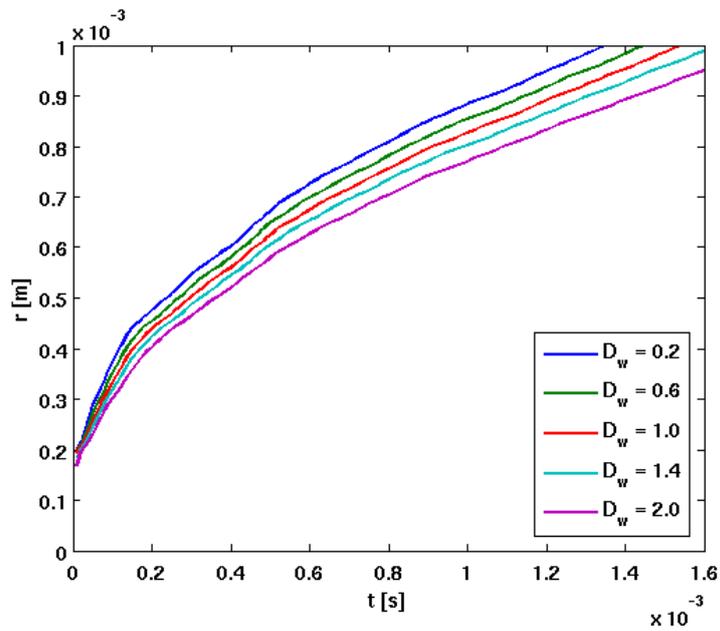


Figure 5.3: The spreading radius as a function of time for  $\theta_s = 43.0^\circ$ ,  $\epsilon = 6.0 \cdot 10^{-5}$  m,  $\sigma = 0.07$  J m $^{-2}$  and various values of  $D_w$ .

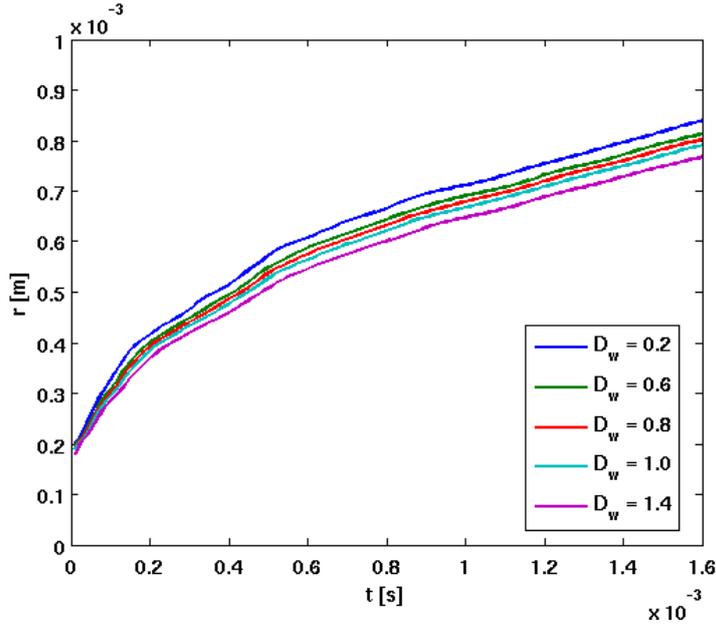


Figure 5.4: The spreading radius as a function of time for  $\theta_s = 117.0^\circ$ ,  $\epsilon = 6.0 \cdot 10^{-5}$  m,  $\sigma = 0.07$  J m $^{-2}$  and various values of  $D_w$ .

As explained in the introduction of this chapter,  $D_w$  is analogous to the friction factor at the contact line between the fluid and the solid. A higher value of  $D_w$  implies a higher friction at this contact line, which causes a slower spreading rate of the droplet. This is indeed observed in Figures 5.2 - 5.4. Also, all solution lines approximately follow the same curve, similar to a square root curve. In Figure 5.5, the relation between a nondimensionalized spreading radius and a nondimensionalized time is shown on a logarithmic scale for all  $\theta_s$ . The nondimensionalized spreading radius and time are as suggested by (Bird et al., 2008):

$$R = \frac{r - r_0}{L}; \quad (5.2)$$

where  $R$  is the nondimensionalized spreading radius,  $r_0$  is the initial spreading radius (the offset) and  $L$  is the initial radius of the droplet.

$$\tau = \frac{t}{\sqrt{\frac{(\rho_l L^3)}{\sigma}}}; \quad (5.3)$$

where  $\tau$  is the nondimensionalized time and  $\rho_l$  is the density of the liquid in the droplet.

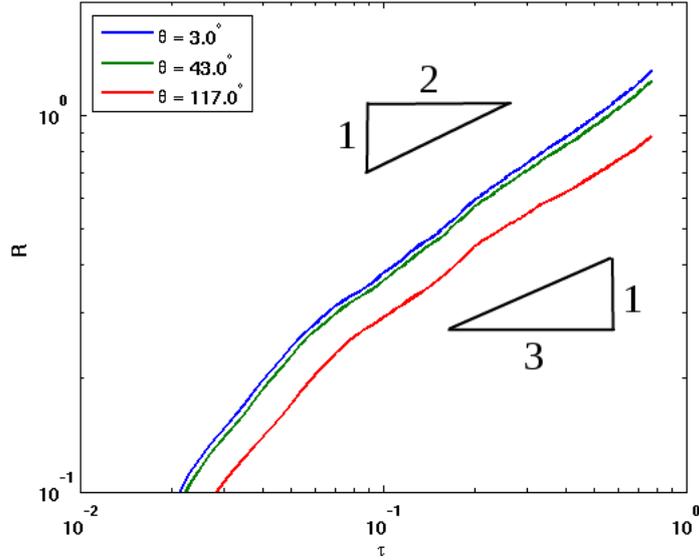


Figure 5.5: The nondimensionalized spreading radius as a function of the nondimensionalized time for  $\epsilon = 6.0 \cdot 10^{-5}$  m,  $D_w = 1.0$  kg m<sup>-1</sup>s<sup>-1</sup>,  $\sigma = 0.07$  J m<sup>-2</sup> and various values of  $\theta_s$ . The solutions should show approximately  $R = c \cdot \tau^{0.5}$  for  $\theta_s = 3.0^\circ$  and  $\theta_s = 43.0^\circ$  and should show  $R = c \cdot \tau^{0.33}$  for  $\theta_s = 117.0^\circ$ .

The slopes of the normalized spreading radius as a function of the normalized time, in Figure 5.5, should confirm the hypothesis of the square root behavior of the spreading rate with time, according to (Bird et al., 2008). However, the slopes of the plots in Figure 5.5, as in Table 5.1, are not as close as desired for  $\theta_s = 3.0^\circ$  and  $\theta_s = 43.0^\circ$ .

$\theta$	Slope of $R(\tau)$	Hypothetical slope of $R(\tau)$
$3.0^\circ$	0.42	0.5
$43.0^\circ$	0.41	0.5
$117.0^\circ$	0.35	0.33

Table 5.1: The slopes of  $R(\tau)$  in Figure 5.5 compared to the hypothetical slopes, according to (Bird et al., 2008), are not as close as desired for  $\theta_s = 3.0^\circ$  and  $\theta_s = 43.0^\circ$ . The slopes from Figure 5.5 are calculated using an exponential fit of the form  $y = a \cdot x^b$ .

### 5.3 Parametric study of the dynamic boundary condition

In this section, the results of parametric studies, which are performed on the spreading rate, using the dynamic boundary condition implementation in OpenFOAM, are discussed. These studies are performed to test a few hypotheses regarding modeling the surface heterogeneities. The tests concern the effect of the interface width,  $\epsilon$ , the surface tension,  $\sigma$  and the ratio  $\frac{\sigma}{D_w}$ . Because of the offset in the figures in previous chapter and the partially failed hypothesis regarding the square root behavior of Figure 5.5, the qualitative relations between the solutions for the varying parameters are considered, rather than the absolute values of the solutions.

First, the effect of the surface tension on the spreading rate is considered. Figure 5.6 shows the spreading radius as a function of time for varying  $\sigma$ . Increasing  $\sigma$  results in an increasing spreading radius, which is explained by the fact that the difference in pressure  $\Delta p$  between the fluid and the gas is proportional to  $\sigma$  and the spreading radius is proportional to  $\Delta p$ . In other words, a higher surface tension results in a higher  $\Delta p$ , which causes a greater force to be applied to the contact line of the

droplet, which results in faster spreading. Applying an exponential fit to the plots of Figure 5.6, results in powers of 0.58 – 0.60 for all  $\sigma$ . This is again not the desired square root behavior.

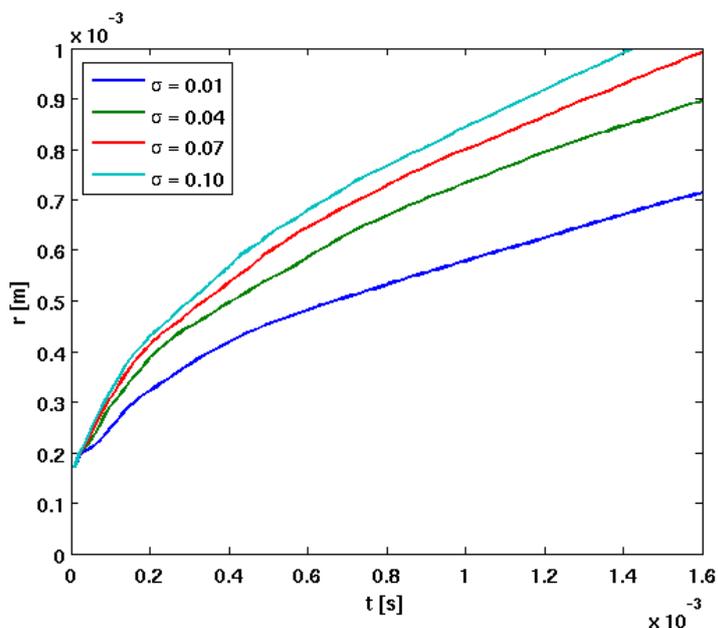


Figure 5.6: The spreading radius for  $\theta_s = 3.0^\circ$ ,  $\epsilon = 6.0 \cdot 10^{-5}$  m,  $D_w = 2.0 \text{ kg m}^{-1}\text{s}^{-1}$  and various values of  $\sigma$ . Due to the proportionality between the pressure difference, between the fluid and the gas, and the surface tension, the spreading radius increases with increasing  $\sigma$ .

Next, the effect of the interface width on the spreading rate is considered. Figure 5.7 shows the spreading radius as a function of time for varying  $\epsilon$ . The solutions for different  $\epsilon$  are considered to be the same, with a certain inaccuracy. This is explained by the fact that  $\epsilon$  only determines the interface width, which has no direct influence on the spreading rate. The difference in interface width can be seen best in Figure 5.8: in Figure 5.8a,  $\epsilon = 0.6 \cdot 10^{-5}$ , while in Figure 5.8b,  $\epsilon = 1.8 \cdot 10^{-5}$ . The transition between the red fluid and the blue gas, indicating the liquid-gas interface, is three times as wide for  $\epsilon = 1.8 \cdot 10^{-5}$  as for  $\epsilon = 0.6 \cdot 10^{-5}$ .

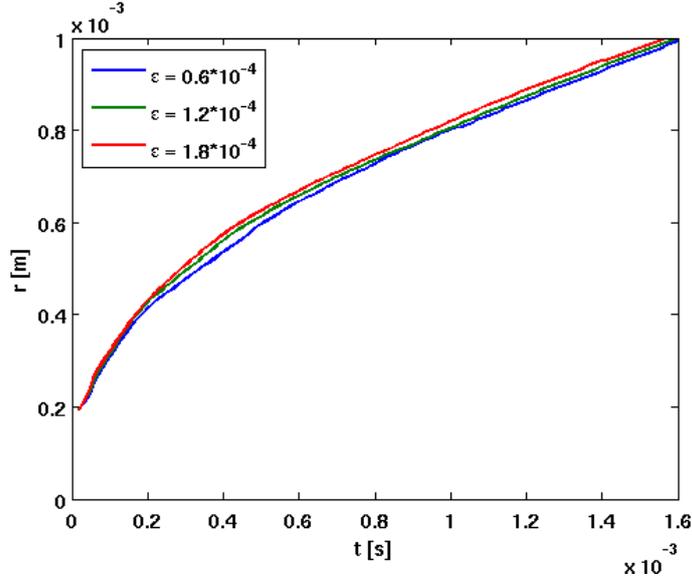


Figure 5.7: The spreading radius for  $\theta_s = 3.0^\circ$ ,  $\sigma = 0.07 \text{ J m}^{-2}$ ,  $D_w = 2.0 \text{ kg m}^{-1}\text{s}^{-1}$  and various values of  $\epsilon$ . The parameter  $\epsilon$  determines the interface width, which has no influence on the spreading radius. Therefore, the solutions are the same, with a certain inaccuracy.

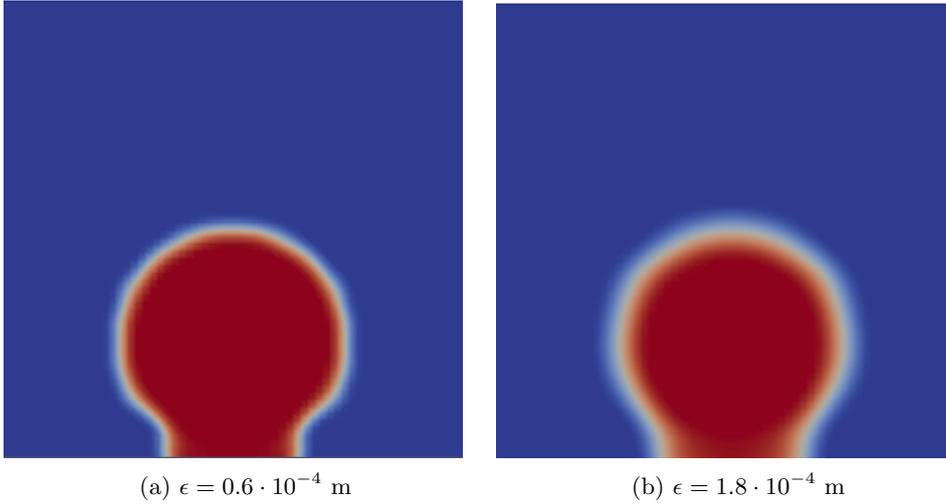


Figure 5.8: Two spreading droplets with  $D_w = 2.0 \text{ kg m}^{-1}\text{s}^{-1}$ ,  $\theta_s = 3.0^\circ$  and  $\sigma = 0.07 \text{ J m}^{-2}$  at  $t = 2.5 \cdot 10^{-4} \text{ s}$ . In Figure 5.8a,  $\epsilon = 0.6 \cdot 10^{-4} \text{ m}$  and in Figure 5.8b,  $\epsilon = 1.8 \cdot 10^{-4} \text{ m}$ . The interface in Figure 5.8b is three times wider than the interface in Figure 5.8a.

Finally, the effect of  $\frac{\sigma}{D_w}$  on the spreading rate, is discussed. According to (Bird et al., 2008), for a given droplet, the dynamics of the droplet spreading is determined by  $\frac{\sigma}{D_w}$ . To test this, two different cases were simulated. In these cases,  $\sigma$  as well as  $D_w$  was increased by a factor of four, hence,  $\frac{\sigma}{D_w}$  was constant. Figure 5.9 shows the spreading radius as a function of time for this test. Again, the solutions are considered to be the same, with a certain inaccuracy. This happens because increasing  $\sigma$  would lead to an increase in applied force to the contact line of the droplet, which causes a higher spreading rate. However, proportionally increasing  $D_w$  should cancel this effect because this increases the friction at the contact line.

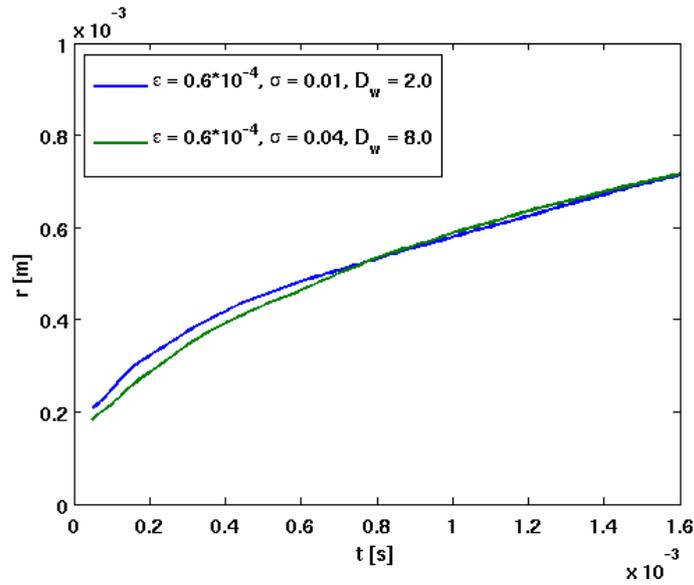


Figure 5.9: The spreading radius for  $\theta_s = 3.0^\circ$ ,  $\epsilon = 6.0 \cdot 10^{-5}$  m and a constant value of  $\frac{\sigma}{D_w} = 5.0 \cdot 10^{-3}$   $\text{m s}^{-1}$ . The solutions are considered to be the same, with a certain inaccuracy, because increasing  $\sigma$  and  $D_w$  proportionally results in their effects to cancel.

The parametric study of varying  $\epsilon$ ,  $\sigma$ , and constant  $\frac{\sigma}{D_w}$  was succesful: all solutions behaved as expected, let aside the square root behavior that is not satisfied.

## 6 Recommendations

With the knowledge of the results of this research, several recommendations can be made concerning future research. These recommendations focus on the topics of Chapters 4 and 5.

Concerning the semi-implicit discretizations that were discussed in Chapter 4, no theoretical stability analysis was done. If this is done in future research, one can find the relation between the stability of the ImEx scheme, the Modified Furihata scheme and the Euler Forward scheme for a significantly larger range of  $\epsilon$  and for different grid sizes.

Concerning the iterative Gâteaux derivative method, obviously, some future research is possible. Apart from finding a decisive answer to what causes the positive diffusive behavior, other implicit iterative methods such as the Newton-Raphson method, can be implemented and the results can be compared. Also, the Gâteaux derivative method can be used to implement a different nonlinear partial differential equation than the Cahn-Hilliard equation. From such a research, interesting results might be obtained. Finally, a stability analysis regarding the Gâteaux derivative method can be done to learn how quantitatively valuable this method really is.

Concerning the dynamic boundary condition implementations, several improvements of the spreading droplet model can be made so that the model fits better with the experimental data from (Bird et al., 2008). First, the droplet depositing device that was used in the experiments, was not modeled here. Modeling this could be done by placing a horizontal wall on the top of the droplet from which the droplet is deposited. Second, the definition of  $D_w$  that was used here is different from the definition in the experiments. This can be improved. Third, the offset of  $r = 0.2 \cdot 10^{-3}$  can be removed. Not just in the plots, but more specifically in the model in OpenFOAM. This might lead to a more accurate spreading radius. Finally, the grid size used here is different from the implementation by (Carlson et al., 2011), this was realized very late and can be improved in future research.

## References

- James C. Bird, Shreyas Mandre, and Howard A. Stone. Short-Time Dynamics of Partial Wetting. *Physical Review Letters*, 100, 2008.
- John W. Cahn and John E. Hilliard. Free Energy of a Nonuniform System. I. Interfacial Free Energy. *Journal of Chemical Physics*, 28, 1958.
- A. Carlson, M. Do-Quang, and G. Amberg. Dissipation in rapid dynamic wetting. *Journal of Fluid Mechanics*, 682, 2011.
- Daniel Dugger and Peter J. Lambert. The 1913 paper of René Gâteaux, upon which the modern-day influence function is based. *Journal of Economic Inequality*, 12:149–152, 2013.
- Daisuke Furihata. A stable and conservative finite difference scheme for the Cahn-Hilliard equation. *Numerische Mathematik*, 87, 2000.
- Junseok Kim. A continuous surface tension force formulation for diffuse-interface models. *Journal of Computational Physics*, 204, 2004.
- Dongsun Lee, Joo-Youl Huh, Darae Jeong, Jaemin Shin, Ana Yun, and Junseok Kim. Physical, mathematical, and numerical derivations of the Cahn-Hilliard equation. *Computational Materials Science*, 81:216–225, 2014.
- Daniel V. Schroeder. *An Introduction to Thermal Physics*. Addison-Wesley, 1999.
- Harrie van den Akker and Rob Mudde. *Fysische transportverschijnselen*. VSSD, 2008.
- C. Vuik, P. van Beek, F. Vermolen, and J. van Kan. *Numerieke Methoden voor Differentiaalvergelijkingen*. VSSD, 2006.