Computational methods for Cahn-Hilliard variational inequalities

Dissertation zur Erlangung des Doktorgrades
der Naturwissenschaften (Dr. rer. nat.)
an der Fakultät für Mathematik
der Universität Regensburg

vorgelegt von

Martin Butz

Regensburg, im Dezember 2011.

Promotionsgesuch eingereicht am 22. Dezember 2011.

Die Arbeit wurde angeleitet von Prof. Dr. Harald Garcke.

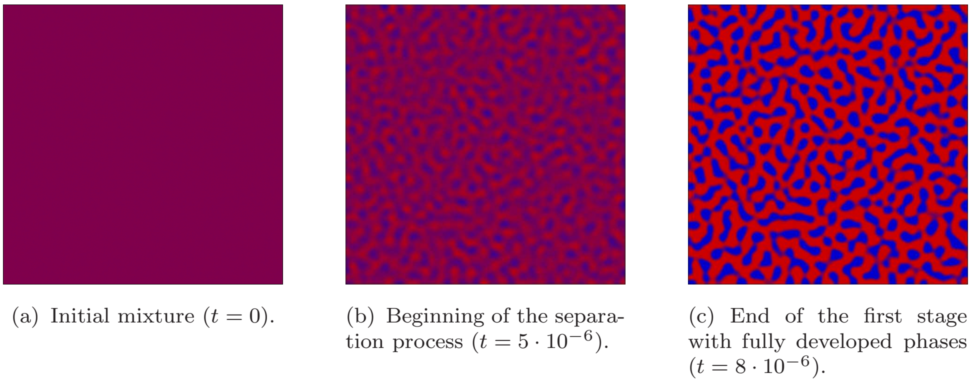
Prüfungsausschuss:	Vorsitzender:	Prof. Dr. Roman Sauer
	1. Gutachter:	Prof. Dr. Harald Garcke
	2. Gutachter:	Prof. Dr. Eberhard Bänsch, Erlangen
	weiterer Prüfer:	Prof. Dr. Georg Dolzmann

Contents

Chapter 1

Introduction

In 1896 Wilhelm Ostwald described the process occurring in binary alloys, where a originally homogeneous mixture forms areas with different phases, see [Ost97, Ost01]. This separation process can be split into three stages. The initially homogeneous mixture develops areas consisting of pure materials and slim areas, called interfaces, in between, where still a mixture of both is present. See Figure 1.1 for a pictorial example of this stage.



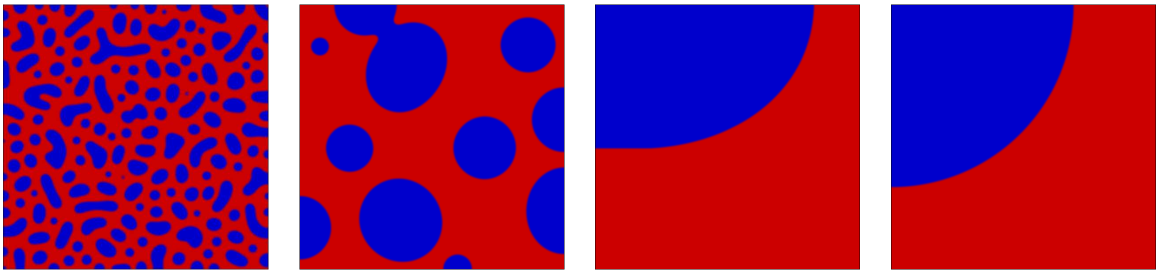
(a) Initial mixture ($t = 0$).

(b) Beginning of the separation process ($t = 5 \cdot 10^{-6}$).

(c) End of the first stage with fully developed phases ($t = 8 \cdot 10^{-6}$).

Figure 1.1: Numerical simulation of the first stage of the phase separation process. Blue is phase A and red phase B, colors in between describe mixtures.

During the next evolutionary phase the regions consisting of the pure phases start to conglomerate, compare Figure 1.2. This happens in such a way that the regions with larger volume are growing at the expense of the smaller ones. These smaller regions eventually disappear. Such a process is termed Ostwald ripening or coarsening. This process continues in the manner of survival of the fattest until only two areas remain with a slim interface layer in between. Finally in the last stage this remaining interfacial layer moves until it reaches a steady state, forming for example a perfect circle, or a quarter-circle in a corner of the surrounding container. This whole phenomenon has been studied by many scientists. Some of the most prominent publications on this topic are the works of Lifshitz and Slyozov [LS61], Wagner [Wag61] and Voorhees [Voo85]. Note that the three stages exhibit different time scales. The first stage is very quick, the second is already quite slow and the evolution in the last stage takes a very long time.



(a) Survival of the fattest stage ($t = 4 \cdot 10^{-5}$).

(b) Survival of the fattest stage ($t = 0.01$).

(c) Only one area of each phase remains ($t = 1.0$).

(d) Numerical steady state ($t = 500$).

Figure 1.2: Numerical simulation of the second and third stage of the phase separation process. Blue is phase A and red phase B, colors in between describe mixtures.

This study of phase separation in binary alloys and hence of interfacial energies lead up to the works of Cahn and Hilliard, who introduced a Ginzburg-Landau model in [CH58]. For this purpose the two phases present are given in terms of concentrations. Since we require a total concentration of one hundred percent at any spatial point, a single variable, denoted by u, is sufficient. A value of 1 at any spatial point x stands for the one phase, a value of -1 for the other, values in between describe mixtures. The Ginzbug-Landau model now uses this quantity to calculate the intrinsic energy of a given state. One part of this energy is given by the free energy. In thermodynamics this quantity describes the amount of work a current configuration can perform, e.g. due to the energy released during demixing of the molecules, and is as such possibly depending on the temperature.

In a similar way to van der Waals [VdW79] the surface energy, i.e. the energy used for the formation of the interfaces separating the two phases or materials, is modeled by adding a gradient term to the free energy. An equilibrium state is hence given by the consideration of a stationary state of the Ginzburg–Landau energy functional $E : H^1(\Omega) \to \mathbb{R}$ given by

$$E(u) = \int\limits_\Omega \frac{\varepsilon\gamma}{2}|\nabla u|^2 + \frac{\gamma}{\varepsilon}\psi(u)\;dx, \tag{1.1}$$

where the free energy is denoted by ψ. The problem we are concerned with starts with a given initial mixture with a given mean composition denoted by $u_m \in (-1, 1)$, i.e. $\int_\Omega u = u_m|\Omega|$. The time evolution should then lead to steady states as large time limits, where any steady state is a minimizer of the above energy functional with the constraint on the mass.

The Cahn–Hilliard equation has found many other applications ranging from the classical aspects in materials science [Gar05] over image processing [DVM02], fluid dynamics [LT98], topology optimization [ZW07], biology [KS08] up to the modeling of mineral growth [KS07] and galaxy structure formation [Tre03]. A general overview on the topic of the Cahn–Hilliard equation and its applications is given by Novick-Cohen in [NC98] as well as in her upcoming book [NCon].

The free energy for temperatures above a critical point can be modeled as a combination of logarithmic terms as discussed in [CH58], resulting in

$$\psi_{log}(u) := \frac{\Theta}{2}\left((1+u)\ln\left(\frac{1+u}{2}\right) + (1-u)\ln\left(\frac{1-u}{2}\right)\right),$$

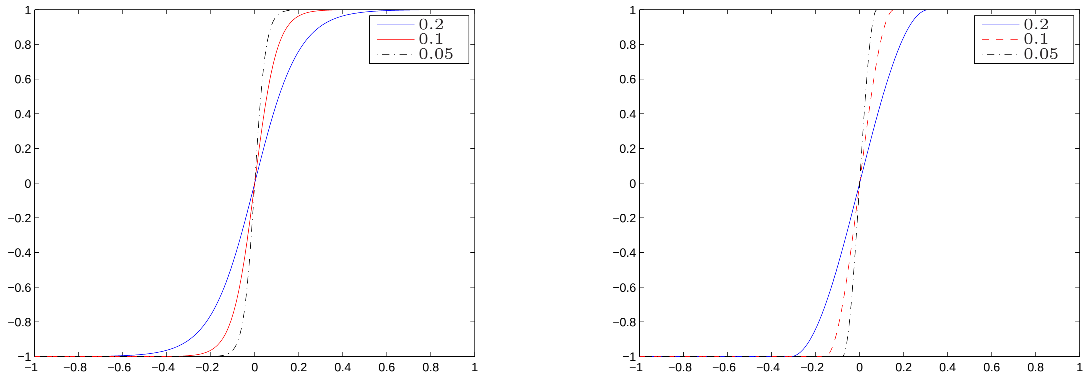
where Θ is the absolute temperature. The thermodynamical justification of the model was derived by Cahn [Cah59], where the equivalence to the self-consistent thermodynamic formalism of Hart [Har59] was shown. A further discussion on the physical background can be found in the article of Gunton and Droz [GD83] and the references therein as well as in Blowey and Elliott [BE91], where a specific derivation of the Cahn-Hilliard model is given. We would also like to point out the book of Ratke and Voorhees [RV01] on growth and coarsening. Quite often, the logarithmic formulation is replaced by a differentiable double well function like, e.g. $\psi(u) := c(1-u^2)^2$, where $c > 0$ is a constant. The interfacial profile of such a free energy, i.e. a cut through the area where u changes from the value 1 to -1, can then be described by means of a *tanh* term with a ε and γ dependent scaling, see, e.g. Section 7.9 of Eck, Garcke and Knabner [EGK08] for a derivation. In Figure 1.3 we show those profiles for different values of ε, where $\gamma = 1$.

The consideration of the so called deep quench limit, i.e. a very rapid cooling of the mixture resulting in temperatures which are very low in comparison to the critical temperature, leads to a non-smooth potential. In this situation the use of an obstacle potential instead of the logarithmic free energy term was introduced by Oono and Puri [OP88]. The mathematical background for this setting is discussed, e.g. by Blowey and Elliott [BE91]. A regularly used choice for the double obstacle potential is given by setting

$$\psi(u) := \begin{cases} \frac{1}{2}(1-u^2) & \text{if } |u| \le 1, \\ +\infty & \text{elsewhere.} \end{cases}$$

Those two variants (with or without differentiability of the free energy) exhibit some distinctive features. The differentiable free energy leads to a system of parabolic partial differential equations, where the interfacial region is diffuse and not bounded to a small area of order ε around the zero level set of u, compare Figure 1.3. The usage of the obstacle potential omits such a feature, since the cosine type profile quickly takes on the values ± 1 without the asymptotic behavior of the *tanh*. This feature is also called sharp diffuse interface. The downside of this approach lies in the resulting system of parabolic partial differential inequalities.

The vast interest in the simulation of problems of this type pushed the development of efficient numerical methods on. There are a large variety of different methods. Here we would like to point out the early development of methods based on a finite element scheme by Blowey and Elliott [BE92], as well as error estimates for linear finite elements, formulated in the multi-component setting by Barrett and Blowey, see [BB97]. Barrett, Nürnberg and Styles discussed in [BNS04] a Gauss-Seidel type method



(a) Logarithmic double well potential ψ_{log}.

(b) Double obstacle potential ψ.

Figure 1.3: Interface profiles for different ε for two choices of ψ.

for a similar problem in the context of void electromigration. Some multigrid methods have also been developed by Kornhuber and Gräser [Kor96, Kor94, GK07, GK09] as well as Kay and Welford [KW06] and Banas and Nürnberg [BN09]. Finally we would like to add the publications by Hintermüller and co-workers [HHT10] and references therein to this list. There the authors use regularization techniques to derive optimality conditions in function space before discretizing the problem. Finally we want to stress that this enumeration of publications is just a small selection.

In the next chapter we present two analytical possibilities to derive the Cahn-Hilliard problem with obstacle potential. Similar to very basic methods, a necessary condition for the minimizers of an energy can be described with the help of the first variation. This method results in a variational inequality due to the constraints imposed on the concentration u. To locate a minimizer this formulation utilizes a mass flux driven by the gradient of the first variation of the energy. Another possibility is the use of a gradient flow structure. Starting out from any initial configuration the evolution always decreases the inherent energy, if possible. This decrease is most profound in direction of the negative gradient of the energy. Since the gradient depends on the used scalar product, the resulting problem also depends on that choice. For the derivation of the Cahn-Hilliard problem the H^{-1} inner product is used.

Both formulations are dependent on another influencing factor, namely the diffusional mobility. Depending on the choice made here different evolution processes are considered. There are corresponding geometric flows, which are also called sharp interface models, to the phase field models. They can be derived by considering the limit $\varepsilon \to 0$ in a certain sense.

The simplest possibility is the usage of a constant mobility. The associated sharp interface problem is the Mullins-Sekerka model, compare Section 2.4. We present both methods for the derivation of the model with constant mobility in Section 2.1 and 2.3. Subsequently we present the problems with non-constant mobility, i.e. the mobility may depend on the concentration u, compare Section 2.5. We discern two separate cases. For one we consider the case of degenerate mobility. The gradient flow formulation requires that the dependency of the mobility on the concentration u is discretized explicitly in time for a well-posed problem. The corresponding sharp interface model is the surface diffusion flow, which we will state briefly in Section 2.6. If we assume the mobility to be non-degenerate, we can formally formulate the Lagrange formulation and state a Lagrange-Newton method for the determination of minimizers.

Finally we give results on the existence and uniqueness of minimizers of the gradient flow formulation with non-constant explicitly discretized mobility in Section 2.7. Those results are applicable to the constant mobility case as well.

In Chapter 3 fully discrete versions of the above methods are derived. We present four different algorithms. In case of constant diffusional mobility we present a projection block sor method (pBSOR), see Algorithm 3.2, as well as the primal-dual active set method (PDAS-II), see Algorithm 3.3. As already described above the non-constant mobility is divided into two separate cases. First the explicit discretization, where we can again formulate a primal-dual active set method (mPDAS-II), see Algorithm 3.5. Then we also discretize the Lagrange-Newton method (LNM) used for the solution of the implicitly discretized mobility, see Algorithm 3.6. Before we close the chapter with existence and uniqueness results for the primal-dual active set algorithms, we address various smaller problems like for example the generation of the meshes, the assembly of the finite element matrices, the selection of the parameter for the primal-dual active set method and the bound on the time step size in the implicitly discretized method.

The only remaining problem is the efficient solution of the arising saddle point problem in all methods. We address this in Chapter 4. First we adapt the Gauss-Seidel type method used for the solution of the variational inequality to derive an iterative method for the solution of the system of equations. Subsequently we give a short introduction on UMFPack, a state of the art direct method, which generates a LU-decomposition of a square system.

Traditionally iterative solvers were superior to the direct methods, which are based on the Gaussian elimination method. Modern methods are able to exploit the sparsity pattern of the system to generate the decomposition very efficiently. In up to two spacial dimensions the matrices are very sparse, due to the usage of linear finite elements and as a result the direct solver is superior to iterative methods.

Only for simulations in three dimensions the advantage begins to shift towards the iterative methods. On the one hand the direct solver requires a growing amount of memory. This is caused by the fact that there are more entries per row in the 3D systems, resulting in an increasing amount of entries generated by so called fill-in. Furthermore, this also results in a quickly increasing computational effort, since more entries have to be calculated. This is where the iterative methods start to catch up. Iterative methods exploit the sparsity too, but are independent of the sparsity pattern, resulting in a slower gain of the computational effort.

Thus we use a Schur complement formulation of the saddle point system to reduce the problem size further and obtain a system we subsequently apply a conjugate gradient method to, compare Chapter 5. It is well known, that those depend heavily on an adequate preconditioning. To this end we adapt a method by Bänsch, Morin and

Nochetto, see [BMN10]. The analysis carried out there is not applicable here, since the used coefficient functions degenerate in our case. However, the basic idea is transferable to formulate a very efficient preconditioning matrix. Due to the missing general results we discuss the necessary spectral conditions on the Schur complement system in one spacial dimension, see Section 5.4.
Finally we show numerical results in Chapter 6. We compare the results obtained by the phase field model to the ones given by the sharp interface models by means of a radially symmetric setting for the Mullins–Sekerka model, see Section 6.1. Starting out with a cylinder in three spacial dimensions, we present simulations with degenerate mobility, related to surface diffusion in Section 6.4.
Next we present results on the efficiency of the presented primal-dual active set methods in two and three spacial dimensions in Section 6.2. In all of these simulations the maximum number of primal-dual active set iterations needed, stays below 10 and thus supports the conjecture that the method is mesh independent.

Chapter 2

The Cahn–Hilliard model

2.1 Variational inequality formulation

This section contains the classical derivation of the Cahn–Hilliard model. We first derive the Cahn–Hilliard equation with a differentiable free energy. Then we will use similar steps for a non-smooth free energy, i.e. the obstacle potential, and obtain the variational inequality formulation.

We consider the above Ginzburg–Landau energy given in (1.1) as a functional on $H^1(\Omega)$, where $\Omega \subset \mathbb{R}^d$ is a bounded domain with Lipschitz boundary, $\gamma > 0$ a parameter related to the interfacial energy density and $\varepsilon > 0$ the parameter controlling the width of the interface. The different approaches are given via the selection of the free energy. As we discussed earlier this results in different interfacial profiles, see Figure 1.3. Those profiles can be derived by means of asymptotic analysis, see e.g. Eck, Garcke and Knabner [EGK08].

The first case we derive will use the differentiable double well potential $\psi(u) = (1-u^2)^2$. We define the first variation of E at a point u in a direction v by

$$\frac{\delta E}{\delta u}(u)(v) := \lim_{\delta \to 0} \frac{E(u + \delta v) - E(u)}{\delta}.$$

For the smooth ψ the first variation of E can easily be calculated and is given by

$$\frac{\delta E}{\delta u}(u)(v) := \int_\Omega \varepsilon\gamma\nabla u \cdot \nabla v + \frac{\gamma}{\varepsilon}\psi'(u)v \, dx.$$

This defines a quantity w, which is called the chemical potential in the context of phase separation, via

$$\int_\Omega wv \, dx := \frac{\delta E}{\delta u}(u)(v) \quad \forall v. \tag{2.1}$$

Starting out from the mass balance law the Cahn–Hilliard equation can now be stated. Therefore we use the mass flux $J := -B\nabla w$, where B is the mobility, and obtain as Cahn, in [Cah61], the evolution equation

$$\partial_t u = -\nabla \cdot J.$$

Since in a closed system there is no mass flux across the boundary the following condition $B\partial_n w = 0$ holds on $\partial\Omega$, where ∂_n denotes the derivative in normal direction on the boundary. The second boundary condition needed for the resulting fourth order problem is given by the natural boundary condition $\partial_n u = 0$. Now taking the mobility to be one and using (2.1) we obtain the Cahn–Hilliard equation as Elliott [Ell89] or Novick-Cohen [NC98]:

$$\partial_t u = \Delta w, \tag{2.2}$$

$$w = -\varepsilon\gamma\Delta u + \frac{\gamma}{\varepsilon}\psi'(u) \tag{2.3}$$

together with the boundary conditions $\partial_n u = \partial_n w = 0$ on $\partial\Omega$.
This approach is valid for smooth free energies ψ. If ψ is not differentiable, as is the case for the double obstacle potential, we introduce the scalar valued indicator function

$$\iota_{[-1,1]}(u) := \begin{cases} 0 & \text{if } u \in [-1,1] \\ +\infty & \text{otherwise.} \end{cases} \tag{2.4}$$

Together with the smooth function $\psi_0(u) := \frac{1}{2}(1-u^2)$ we define the free energy

$$\psi(u) := \psi_0(u) + \iota_{[-1,1]}(u) = \begin{cases} \frac{1}{2}(1-u^2) & \text{if } |u| \leq 1, \\ +\infty & \text{elsewhere} \end{cases} \tag{2.5}$$

introduced by Blowey and Elliott [BE91], where the above approach has to be slightly modified. The energy (1.1) is now given by

$$E(u) = \int\limits_\Omega \frac{\varepsilon\gamma}{2}|\nabla u|^2 + \frac{\gamma}{\varepsilon}\psi_0(u) + \iota_{[-1,1]}(u)\, dx.$$

The calculation of the first variation cannot be done straight forward. Note that the indicator function can only be differentiated in the sense of subdifferentials, see e.g. Evans [Eva10] or Zeidler [Zei85].

Definition 2.1. *Let H be a Hilbert space with inner product $(\cdot,\cdot)_H$ and $f : H \to (-\infty, +\infty]$ be convex and proper. The subdifferential of f at a point $u \in H$ is then given by*

$$\partial f(u) = \{p \in H \mid f(u) - f(z) \geq (p, u-z)_H \text{ for all } z \in H\}.$$

When we consider $\iota_{[-1,1]} : \mathbb{R} \to \mathbb{R}$ as in (2.4), the subdifferential at a point $u \in \mathbb{R}$ is given by

$$\partial\iota_{[-1,1]}(u) = \left\{p \in \mathbb{R} \mid \iota_{[-1,1]}(u) - \iota_{[-1,1]}(z) \geq p(u-z) \text{ for all } z \in \mathbb{R}\right\}.$$

When calculating the first variation of the energy, the first two terms of the energy can be handled as before. To consider the third term, we define the functional $I_{[-1,1]} : L^2(\Omega) \to \mathbb{R}$ by

$$I_{[-1,1]}(u) := \int\limits_\Omega \iota_{[-1,1]}(u(x))\, dx.$$

We denote the inner product on $L^2(\Omega)$ with $(\cdot,\cdot)$. Differentiating $I_{[-1,1]} : L^2(\Omega) \to \mathbb{R}$ in the sense of subdifferentials, we get

$$\begin{aligned}\partial I(u) = &\left\{ \phi \in L^2(\Omega) \mid \int_\Omega \iota_{[-1,1]}(u) - \iota_{[-1,1]}(z)\, dx \geq (\phi, u-z) \text{ for all } z \in L^2(\Omega) \right\} \\ =&\{\phi \in L^2(\Omega) \mid \iota_{[-1,1]}(u(x)) - \iota_{[-1,1]}(z(x)) \geq \phi(x)(u(x)-z(x)) \\ &\qquad \text{almost everywhere, for all } z \in L^2(\Omega)\} \\ =&\left\{\phi \in L^2(\Omega) \mid \phi(x) \in \partial\iota_{[-1,1]}(u(x)) \text{ almost everywhere}\right\}.\end{aligned}$$

When we split the condition in cases depending on u, we obtain that $\mu \in L^2(\Omega)$ is in the subdifferential of I at a point $u \in L^2(\Omega)$ with $|u| \leq 1$ if and only if $\mu(x) \in \partial\iota_{[-1,1]}(u(x))$, i.e.

$$\mu(x) \in \begin{cases} (-\infty, 0] & \text{if } u(x) = -1\,, \\ \{0\} & \text{for } u(x) \in (-1,1)\,, \\ [0,\infty) & \text{if } u(x) = 1\,, \end{cases} \tag{2.6}$$

is fulfilled almost everywhere. This can be rewritten in the following complementarity form

$$\mu = \mu_+ - \mu_-,\ \mu_+ \geq 0,\ \mu_- \geq 0,\ \mu_+(u-1) = 0,\ \mu_-(u+1) = 0, \tag{2.7}$$

which also has to hold almost everywhere and we obtain analogously to (2.2)-(2.3) the equations

$$\partial_t u = \Delta w\,, \tag{2.8}$$

$$w = -\varepsilon\gamma\Delta u + \frac{\gamma}{\varepsilon}(\psi_0'(u) + \mu) \tag{2.9}$$

with $\mu \in \partial I_{[-1,1]}(u)$, $|u| \leq 1$ and zero Neumann boundary conditions for u and w. This formulation can be restated equivalently in a variational inequality formulation, see e.g. Blowey and Elliott [BE91] or Kinderlehrer and Stampacchia [KS80] and Friedman [Fri82] for other obstacle problems, as follows:

$$\partial_t u = \Delta w\,, \tag{2.10}$$

$$(w, \xi - u) \leq \varepsilon\gamma(\nabla u, \nabla(\xi - u)) + \frac{\gamma}{\varepsilon}(\psi_0'(u), \xi - u) \quad \forall\, \xi \in H^1(\Omega), |\xi| \leq 1\,, \tag{2.11}$$

together with $|u| \leq 1$ almost everywhere. This system is the variational inequality formulation of the Cahn-Hilliard model with a Blowey-Elliott potential. It can be shown that a unique solution (u,w) exists to (2.10), (2.11). More precisely the following theorem, see [BE91], is true.

Theorem 2.2. *Assume Ω is convex or $\partial\Omega \in C^{1,1}$, $u_0 \in H^1(\Omega)$ with $|u_0| \leq 1$ and $⨍_\Omega u_0 = m \in (-1,1)$. Then there exists a unique pair (u,w) such that*

$$u \in H^1(0,T;(H^1(\Omega))') \cap L^2(0,T;H^2(\Omega)) \cap L^\infty(0,T;H^1(\Omega))\,,$$

$|u| \leq 1$ a.e. and $w \in L^2(0,T;H^1(\Omega))$ which solves (using the duality pairing $\langle .,. \rangle$ between $H^{-1}(\Omega)$ and $H^1(\Omega)$)

$$\langle \partial_t u, \eta \rangle + (\nabla w, \nabla \eta) = 0 \quad \textit{for all } \eta \in H^1(\Omega) \quad \textit{and } t \in (0,T) \qquad \textit{a.e.}$$

together with the variational inequality (2.11) and $u(0,\cdot) = u_0$.
In particular $\mu = \frac{\varepsilon}{\gamma} w + \varepsilon^2 \Delta u - \psi_0'(u) \in L^2(\Omega_T)$.

2.2 Introduction of the Cahn–Hilliard problem as gradient flow

We first introduce a general gradient flow setting as well as the H^{-1}-space and scalar product that we need. We use an Euler time step scheme for the discretization of the time derivative, where for most terms an implicit approximation will be used. The concave free energy term ψ_0' poses some additional restrictions analytically if taken implicitly. Hence we discuss an explicit variant here as well as an implicit discretization. The time discrete problem features a natural variational structure and is tied to a minimization problem, which penalizes deviations from u^{n-1}. The gradient flow structure enables us then to derive the primal-dual active set method after a time discretization. Consider a vector space $\mathbf{Z}$ and an affine subspace $\mathbf{U} \subset \mathbf{Z}$, i.e. there exists a $\overline{u} \in \mathbf{Z}$ and a linear space $\mathbf{Y} \subset \mathbf{Z}$ such that $\mathbf{U} = \overline{u} + \mathbf{Y}$. We additionally choose an inner product $(\cdot,\cdot)_{\mathbf{Z}}$ on $\mathbf{Z}$, which induces the associated norm $\|\cdot\|_{\mathbf{Z}}$. The gradient of a sufficiently smooth function $E : \mathbf{U} \to \mathbb{R}$ depends on the inner product chosen for $\mathbf{Z}$. As before we define the first variation of E at a point $u \in \mathbf{U}$ in a direction $v \in \mathbf{Y}$ by

$$\frac{\delta E}{\delta u}(u)(v) := \lim_{\delta \to 0} \frac{E(u+\delta v) - E(u)}{\delta}\,.$$

We say that there exists a gradient of E with respect to the inner product $(.,.)_{\mathbf{Z}}$ on $\mathbf{Z}$, which we denote by $\mathrm{grad}_{\mathbf{Z}} E(u)$, if

$$(\mathrm{grad}_{\mathbf{Z}} E(u), v)_{\mathbf{Z}} = \frac{\delta E}{\delta u}(u)(v) \quad \text{holds for all} \quad v \in \mathbf{Y}\,. \tag{2.12}$$

Now the gradient flow of E with respect to the inner product $(.,.)_{\mathbf{Z}}$ is given as

$$\partial_t u(t) = -\mathrm{grad}_{\mathbf{Z}} E(u(t))\,. \tag{2.13}$$

The energy decreases in time due to the inequality

$$\frac{d}{dt} E(u(t)) = (\mathrm{grad}_{\mathbf{Z}} E(u(t)), \partial_t u(t))_{\mathbf{Z}} = -\|\partial_t u\|_{\mathbf{Z}}^2 \leq 0.$$

Remark 2.3. *The gradient flow associated to the L^2 scalar product results in the so called Allen–Cahn equation. In the above notation we set $\mathbf{Z} = L^2(\Omega)$, $\mathbf{U} = \mathbf{Y} = H^1(\Omega)$ and $\overline{u} = 0$ and obtain for the energy E given in (1.1) as Blank, Garcke, Sarbu and Styles in [BGSS09]*

$$\mathit{grad}_{L^2} E(u) = -\varepsilon\gamma\Delta u + \frac{\gamma}{\varepsilon}\psi'(u). \tag{2.14}$$

As mentioned above in the Cahn–Hilliard model the total concentration, i.e. $\int_\Omega u(x)\,dx$ is assumed to be conserved. Denoting by $⨍_\Omega u$ the mean value of a function u, we now define for a given $m \in (-1,1)$ the sets

$$\mathbf{U} := \left\{ u \in H^1(\Omega) \mid ⨍_\Omega u = m \right\}, \quad \mathbf{Y} := \left\{ u \in H^1(\Omega) \mid ⨍_\Omega u = 0 \right\}. \tag{2.15}$$

In addition we introduce $\mathbf{Z} = H^{-1}(\Omega) = \{u' \in (H^1(\Omega))' \mid \langle u', 1 \rangle = 0\}$, i.e. all bounded linear functionals on $H^1(\Omega)$ that vanish on constant functions. Here and in what follows $\langle .,. \rangle$ denotes the dual pairing. On $\mathbf{Z} = H^{-1}(\Omega)$ we define the H^{-1}-inner product for $v_1, v_2 \in \mathbf{Z}$ as

$$(v_1, v_2)_{H^{-1}} := \int_\Omega \nabla(-\Delta)^{-1} v_1 \cdot \nabla(-\Delta)^{-1} v_2 \tag{2.16}$$

where $y = (-\Delta)^{-1} v$ is the weak solution of $-\Delta y = v$ in Ω and $\partial_n y = 0$ on $\partial\Omega$, i.e. $\int_\Omega \nabla y \cdot \nabla \eta = v(\eta)$ for all $\eta \in H^1(\Omega)$. We remark that the solution to this elliptic problem is only defined up to a constant and we always choose y such that $⨍_\Omega y = 0$. The function space $\mathbf{Y}$ is canonically embedded into $\mathbf{Z}$ since $u \in \mathbf{Y}$ can be related to the linear functional $y \mapsto \int_\Omega uy$. For $v_1, v_2 \in \mathbf{Y}$ we obtain using the L^2–inner product, which we denote by $(.,.)$,

$$(v_1, v_2)_{H^{-1}} = (v_1, (-\Delta)^{-1} v_2) = ((-\Delta)^{-1} v_1, v_2)\,.$$

These identities also hold for functions $v_1, v_2 \in L^2(\Omega)$ with mean value zero. To compute the H^{-1}-gradient of E we now need to find $\mathrm{grad}_{H^{-1}} E(u) \in \mathbf{Z}$ such that

$$(v, \mathrm{grad}_{H^{-1}} E(u))_{H^{-1}} = \frac{\delta E}{\delta u}(u)(v) \text{ holds for all } v \in \mathbf{Y}.$$

From the above we obtain $(v, (-\Delta)^{-1}\mathrm{grad}_{H^{-1}} E(u)) = (v, \mathrm{grad}_{L^2} E(u))$ and hence

$$\mathrm{grad}_{H^{-1}} E(u) = (-\Delta)\mathrm{grad}_{L^2} E(u)\,. \tag{2.17}$$

Then, the Cahn-Hilliard equation is given as the H^{-1}-gradient flow of the Ginzburg–Landau energy E. If ψ is smooth we obtain the fourth order parabolic equation

$$\partial_t u = -\mathrm{grad}_{H^{-1}} E(u) = \Delta\left(-\varepsilon\gamma\Delta u + \frac{\gamma}{\varepsilon}\psi'(u)\right) \tag{2.18}$$

or equivalently using the chemical potential w the equation can be rewritten as a system as follows

$$\partial_t u = \Delta w\,, \tag{2.19}$$

$$w = -\varepsilon\gamma\Delta u + \frac{\gamma}{\varepsilon}\psi'(u). \tag{2.20}$$

In addition the boundary conditions $\partial_n u = \partial_n w = 0$ on $\partial\Omega$ have to hold. Let us remark, that in this formulation we do not necessarily have $⨍_\Omega w = 0$, i.e. in general

$w \neq -(-\Delta)^{-1}\partial_t u$ due to the definition of the inverse Laplacian, but both functions only differ by an additive constant.
The gradient flow structure, we used for the modeling of the evolution process, has a natural variational structure for a backward Euler time discretization scheme. Let $\tau > 0$ be the time step width and $t_n := n\tau$ for $n \in \mathbb{N}_0$ the discrete times. We denote $u^n(x) := u(t_n, x)$. The solution u^n of the time discretized gradient flow given by

$$\frac{1}{\tau}\left(u^n - u^{n-1}\right) = -\mathrm{grad}_{\mathbf{Z}} E(u^n) \tag{2.21}$$

can also be computed as the solution of the minimization problem

$$\min_{u\in\mathbf{U}} \left\{ E(u) + \frac{1}{2\tau}\left\|u - u^{n-1}\right\|_{\mathbf{Z}}^2 \right\}, \tag{2.22}$$

i.e. the energy is minimized with an additional term penalizing the deviation of u^{n-1} in the $\mathbf{Z}$–norm. This can easily be seen by calculating the first variation in direction $v \in \mathbf{Y}$ of the energy in (2.22) using the definition (2.12) of $\mathrm{grad}_{\mathbf{Z}}$ we get

$$\begin{aligned}
0 =& \frac{d}{ds}\left(E(u+sv) + \frac{1}{2\tau}\left\|u + sv - u^{n-1}\right\|_{\mathbf{Z}}^2 \right)\Bigg|_{s=0} \\
=& \frac{\delta E}{\delta u}(u)(v) + \frac{1}{\tau}(u - u^{n-1}, v)_{\mathbf{Z}}. \\
=& (\mathrm{grad}_{\mathbf{Z}} E(u), v)_{\mathbf{Z}} + \frac{1}{\tau}(u - u^{n-1}, v)_{\mathbf{Z}}.
\end{aligned}$$

The Cahn–Hilliard problem considered here uses $\mathbf{Z} = H^{-1}(\Omega)$. The spaces $\mathbf{U}$ and $\mathbf{Y}$ are given as in (2.15). The calculation of the H^{-1}–norm requires the solution of a Poisson problem with Neumann boundary conditions and a no-mass condition. We use $v \in \mathbf{Y}$ as an auxiliary variable for the calculation of this norm. Let E be the Ginzburg–Landau energy as in (1.1) and ψ the obstacle potential from (2.5).
The minimization problem (2.22) then reads as follows:

$$\min_{u\in\mathbf{U}} \left\{ \frac{\varepsilon\gamma}{2}\int_\Omega |\nabla u|^2 + \frac{\gamma}{\varepsilon}\int_\Omega \psi_0(u) + \frac{\tau}{2}\int_\Omega |\nabla v|^2 \right\} \tag{2.23}$$

such that

$$\tau\Delta v = u - u^{n-1},\ \partial_n v = 0 \text{ on } \partial\Omega, \tag{2.24}$$

$$⨍_\Omega v = 0, \tag{2.25}$$

$$⨍_\Omega u = m, \tag{2.26}$$

$$|u| \leq 1. \tag{2.27}$$

Note that v now is used for the solution of the Poisson problem hidden in the H^{-1}-norm. The conditions (2.25) and (2.26) are the restrictions resulting from our choice for the spaces $\mathbf{U}$ and $\mathbf{Y}$ given in (2.15). The last restriction (2.27) is due to the usage of the obstacle potential, which would lead to infinite energy for any other u. The above minimization problem has the form of an optimal control problem with control u and state v. We introduce the corresponding Lagrangian

$$\begin{aligned}\mathcal{L}(v,\kappa,u,w,\mu_+,\mu_-) :=& \frac{\varepsilon\gamma}{2}\int_\Omega |\nabla u|^2 + \frac{\gamma}{\varepsilon}\int_\Omega \psi_0(u) + \frac{\tau}{2}\int_\Omega |\nabla v|^2 - \int_\Omega \tau\nabla w\cdot\nabla v\\ &- \int_\Omega (u-u^{n-1})w - \kappa\int_\Omega v - \frac{\gamma}{\varepsilon}\int_\Omega \mu_+(1-u) - \frac{\gamma}{\varepsilon}\int_\Omega \mu_-(1+u)\,,\end{aligned}$$

where $w \in H^1(\Omega)$ is the Lagrange multiplier for the weak formulation of (2.24) and $\kappa \in \mathbb{R}$ for (2.25). The inequality constraints are incorporated with scaled multipliers $\mu_\pm$. The scaling with $\frac{\gamma}{\varepsilon}$ will become important, when we discuss the choice of the parameter $c > 0$ used for the primal-dual active sets, see Section 3.6.3. For more details on the topic of the Lagrangian formulation of constrained minimization problems, see e.g. Nocedal and Wright [NW06] or Geiger and Kanzow [GK02]. Now all constraints are incorporated and can be regained as the first variation of $\mathcal{L}$ with respect to the multipliers, see e.g. Tröltzsch [Trö10] for an introduction to optimal control theory. Note that the variation of w by a constant implies $⨍_\Omega u = m := ⨍_\Omega u^{n-1}$. Hence w is also the Lagrange multiplier for the equality constraint (2.26).

The next step is the derivation of the first order optimality conditions, which leads to the following KKT-system, where (2.28), (2.30) and (2.31) have to be understood in their weak forms:

$$\tau\Delta(w-v) = \kappa \text{ in } \Omega,\ \partial_n w = \partial_n v \text{ on } \partial\Omega\,, \tag{2.28}$$

$$⨍_\Omega v = 0\,, \tag{2.29}$$

$$\frac{1}{\tau}(u-u^{n-1}) = \Delta v \text{ in } \Omega,\ \partial_n v = 0 \text{ on } \partial\Omega\,, \tag{2.30}$$

$$w + \varepsilon\gamma\Delta u - \frac{\gamma}{\varepsilon}\psi_0'(u) - \frac{\gamma}{\varepsilon}\mu = 0 \text{ in } \Omega,\ \partial_n u = 0 \text{ on } \partial\Omega\,, \tag{2.31}$$

$$\mu := \mu_+ - \mu_-,\ \mu_+ \geq 0,\quad \mu_- \geq 0,\ \text{a.e. in } \Omega\,, \tag{2.32}$$

$$\mu_+(u-1) = 0,\ \mu_-(u+1) = 0,\ \text{a.e. in } \Omega\,, \tag{2.33}$$

$$\text{and } |u| \leq 1 \text{ a.e. in } \Omega\,. \tag{2.34}$$

Additionally varying v by a constant only, we get

$$\kappa = 0. \tag{2.35}$$

Thus given (2.28)-(2.29) and (2.35) we obtain $w - ⨍_\Omega w = v$, i.e. v and w only differ by a constant. We can replace v by w in (2.30) and we hence obtain in particular a time discretization of (2.8), (2.9) using the complementary formulation (2.7). The Lagrange multiplier w coincides with the chemical potential, and the scaled Lagrange multiplier μ lies in the subdifferential of $I_{[-1,1]}$. Since the equations (2.28), (2.29) and (2.35) are not needed we omit them in the following.

2.3 Primal–dual active set formulation

Starting from the above KKT-system (2.28)-(2.35), we obtain a reduced first order optimality system due to the simplifications discussed previously. Thus we formulate a primal-dual active set algorithm in a formal way for the following problem.
Find u, w, μ such that

$$\tfrac{1}{\tau}(u - u^{n-1}) = \Delta w \text{ in } \Omega\,,\ \partial_n w = 0 \text{ on } \partial\Omega \tag{2.36}$$

holds together with (2.31)-(2.34).
We now introduce for a $c > 0$ the active sets

$$\begin{aligned} A^+ &= \left\{x \in \Omega \mid u(x) + \tfrac{\mu(x)}{c} > 1\right\},\\ A^- &= \left\{x \in \Omega \mid u(x) + \tfrac{\mu(x)}{c} < -1\right\} \end{aligned}$$

and the inactive set $I := \Omega \setminus (A^+ \cup A^-)$. The conditions (2.32)-(2.34) can be reformulated as

$$\begin{aligned} u(x) &= \pm 1 && \text{if } x \in A^\pm \text{ and}\\ \mu(x) &= 0 && \text{if } x \in I\,. \end{aligned}$$

Formally this leads to the following primal-dual active set strategy employing the primal variable u and the dual variable μ.

Algorithm 2.1 *Formal primal-dual active set algorithm* (PDAS-I)

1. Set $k = 0$, initialize $A_0^\pm$ and define $I_0 = \Omega \setminus (A_0^+ \cup A_0^-)$.
2. Set $u_k = \pm 1$ on $A_k^\pm$ and $\mu_k = 0$ on I_k.
3. Solve the coupled system of PDE's (2.36), (2.31) to obtain u_k on I_k, μ_k on $A_k^+ \cup A_k^-$ and w_k on Ω.
4. Set $A_{k+1}^+ := \left\{x \in \Omega \mid u_k(x) + \tfrac{\mu_k(x)}{c} > 1\right\}$,

   $A_{k+1}^- := \left\{x \in \Omega \mid u_k(x) + \tfrac{\mu_k(x)}{c} < -1\right\}$ and

   $I_{k+1} := \Omega \setminus (A_{k+1}^+ \cup A_{k+1}^-)$.
5. If $A_{k+1}^\pm = A_k^\pm$ stop, otherwise set $k = k+1$ and goto 2.

Remark 2.4. *The above complementarity system* (2.32)-(2.33) *can be reformulated as a semi-smooth equation by*

$$H(u,\mu) := \mu - (\max(0,\mu + c(u-1)) + \min(0,\mu + c(u+1))) = 0. \tag{2.37}$$

A semi-smooth Newton method, based on results from Qi and Sun [QS93], applied in a formal way to (2.36), (2.31) *and* (2.37) *is equivalent to the above primal-dual active set method, see e.g. Hintermüller, Ito and Kunisch [HIK02] for a different context.*

Proof. We start out from (2.37) and checking all cases.
If $\mu + c(u+1) < 0$, then

$$0 = H(u,\mu) = \mu - \mu - c(u+1) = -c(u+1)$$

holds. Thus $u = -1$ and hence $\mu < 0$.
If $\mu + c(u-1) > 0$, then

$$0 = H(u,\mu) = \mu - \mu - c(u-1) = -c(u-1)$$

holds. Thus $u = 1$ and hence $\mu > 0$.
Otherwise we get $\mu = 0$ and obtain

$$c(u-1) \le 0 \text{ and } c(u+1) \ge 0 \quad \Rightarrow \quad |u| \le 1.$$

If we plug the complementarity conditions (2.32)-(2.33) into (2.37) it is obvious that $H(u,\mu) = 0$ holds. □

Remark 2.5. *The iterations in the above algorithm (PDAS-I) are in general not applicable in function space since the iterates μ_k are only measures and not L^2–functions, see Ito and Kunisch [IK03]. In Chapter 3 we derive a fully discretized version of the algorithm and we will show, by means of the equivalence to the semi-smooth Newton method, that local convergence holds.*

2.4 Mullins–Sekerka model as sharp interface limit

For the Cahn–Hilliard problem with non-degenerate constant mobility discussed earlier the solution features a diffuse interfacial region, where the change of the concentration u from 1 to -1 occurs. The shape of the transition of the phases on the interface heavily depends on the used free energy potential. In case of a differentiable potential $\Psi(u) = c \cdot (1-u^2)^2$ the interface omits no sharp boundaries since the intersection can to leading order be described by an adequately scaled *tanh*-term. The obstacle potential considered here admits a so called sharp diffuse interface, i.e. the region where the phase change takes place is a bounded compact subset of Ω. The width of this interfacial region is proportional to $\varepsilon\pi$, i.e. in one space dimension $u(x) \approx \sin(x/\varepsilon)$ on the interface, where $x \in [-\frac{\varepsilon\pi}{2}, \frac{\varepsilon\pi}{2}]$.

For the investigation of the behavior when considering the limit $\varepsilon \to 0$, which in essence is the study of situations with vanishing interfacial thickness, the notion of Γ–limit of the free energy leads to a free boundary problem, namely the Mullins–Sekerka problem (or Hele–Shaw problem) stated below. For further details we refer to Garcke [Gar05] or Pego [Peg89]. A more general discussion of the convergence of a class of phase field models to their corresponding sharp interface limits is given by Caginalp and Chen [CC98].

The sharp interface limit of a differentiable free energy ψ is derived by Chen [Che96], where also exact solutions for some radially symmetric settings are calculated. Since the sharp interface limit of this differentiable free energy coincides with the one obtained for the double obstacle potential, we adapt the notation used there for the formulation of the Mullins–Sekerka model and the technique used there for the computation of those exact solutions we use. The asymptotic analysis for the parabolic obstacle problem we use here is executed in Blowey and Elliott [BE93]. There the phase field model under consideration is given by the following variational inequality problem:

Find u and w such that

$$\begin{aligned} cw_t + \tfrac{l}{2}u_t &= k\Delta w, \\ \big(\sigma_1 \varepsilon u_t - \varepsilon\Delta u + \tfrac{1}{\varepsilon}\psi_0'(u) - \sigma_2 w, \eta - u\big) &\geq 0, \qquad \text{for all } \eta \text{ with } |\eta| \leq 1. \end{aligned}$$

To fit our choice of scaling for the parameters, we set $c = 0$, $k = 1$, $l = 2$, $\sigma_1 = 0$ and $\sigma_2 = \frac{1}{\gamma}$ resulting in:

Find u and w such that

$$\begin{aligned} u_t &= \Delta w, \\ \big(-\varepsilon\gamma\Delta u + \tfrac{\gamma}{\varepsilon}\psi_0'(u) - w, \eta - u\big) &\geq 0, \qquad \text{for all } \eta \text{ with } |\eta| \leq 1. \end{aligned}$$

The corresponding Mullins–Sekerka model obtained is then given by the following free boundary problem, where the free boundary is given by $\Gamma = \partial\{u = 1\} \cap (\Omega \times (0, \infty)) = \bigcup\limits_{t>0} \Gamma_t \times \{t\}$. Note that Γ_t are suitable hypersurfaces in this context. The limit (w, Γ) solves

$$\begin{array}{llll} \Delta w &= 0 & \text{in} \quad \Omega \setminus \Gamma_t, & t > 0, \\ \partial_n w &= 0 & \text{on} \quad \partial\Omega, & t > 0, \\ \sigma\kappa &= w & \text{on} \quad \Gamma_t, & t > 0, \\ 2V &= [\partial_\nu w]_{\Gamma_t} \nu & \text{on} \quad \Gamma_t, & t > 0. \end{array} \tag{2.38}$$

Here ν denotes the normal on Γ_t pointing into the set $\{u \equiv 1\}$, V the normal velocity of the interface and κ the sum of the principal curvatures of Γ_t. By $[.]_{\Gamma_t}$ we denote the jump across the interface. The constant σ is given by the formula

$$\sigma = \gamma \int\limits_{-1}^{1} \sqrt{\frac{\psi_0(u)}{2}}\, du = \gamma \int\limits_{-1}^{1} \frac{1}{2}\sqrt{1 - u^2}\, du = \gamma\frac{\pi}{4},$$

which is a result of the asymptotic analysis.

Remark 2.6. *The different scaling with $\frac{1}{\varepsilon}$ we used in [BBG11] for the free energy term can also be described by the above ansatz by modifying the free energy term to incorporate a γ term by setting $\psi_0(u) = \frac{1}{2\gamma}(1-u^2)$. This results in a changed dependence of the parameter $\sigma = \sqrt{\gamma}\frac{\pi}{4}$ with respect to the order of γ.*

In the following we use a radially symmetric situation with $\Omega = B_1(0)$, where explicit solutions can be calculated. In Stoth, see [Sto96], the analysis for spherical symmetric situations is discussed by means of energy methods. The calculations below are based on results from Chen [Che96], where solutions to radially symmetric settings for an arbitrary number of interfaces are discussed. We use two concentric circles (or spheres) Γ_1 and Γ_2 as depicted in Figure 2.1 as scenario in our numerical experiments.

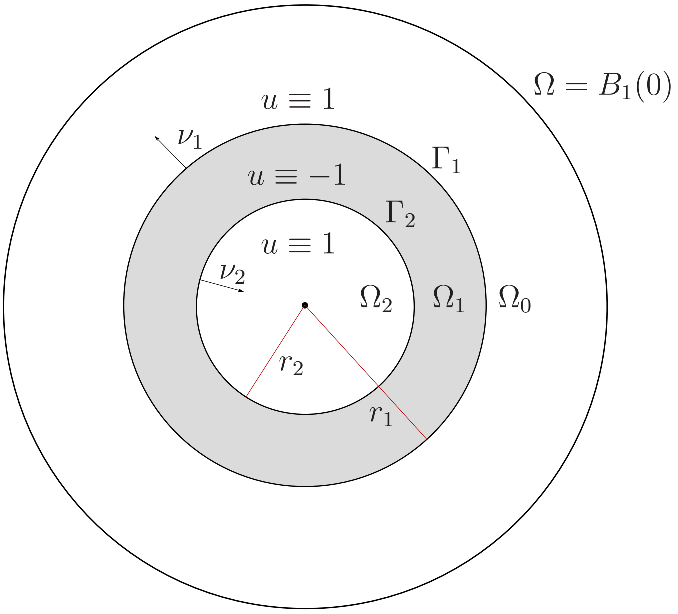


Figure 2.1: Radially symmetric free boundary problem in two space dimensions.

Using (2.38) and the notation used in Figure 2.1 we obtain the following conditions respectively equations for w on the three parts of Ω.

In Ω_0 :

$$\begin{aligned} \Delta w &= 0 && \text{in } \Omega_0, \\ \partial_n w &= 0 && \text{on } \partial\Omega, \\ w &= \sigma\kappa_1 && \text{on } \Gamma_1. \end{aligned}$$

In Ω_1 :

$$\begin{aligned} \Delta w &= 0 && \text{in } \Omega_1, \\ w &= \sigma\kappa_1 && \text{on } \Gamma_1, \\ w &= \sigma\kappa_2 && \text{on } \Gamma_2. \end{aligned}$$

In Ω_2 :

$$\begin{aligned} \Delta w &= 0 && \text{in } \Omega_2, \\ w &= \sigma\kappa_2 && \text{on } \Gamma_2. \end{aligned}$$

The following calculation is now done for two spatial dimensions. A similar calculation can be done in three dimensions with an adequately adapted ansatz function.

Since the curvature κ_i of circles Γ_i is constant, we obtain a solution by setting $w_{|\Omega_2} \equiv \sigma\kappa_2$, $w_{|\Omega_0} \equiv \sigma\kappa_1$ and using the ansatz $w(r) = ln(r)c_1(t) + c_2(t)$ on Ω_1. Use of the boundary conditions on Ω_1 given by

$$\begin{aligned} ln(r_1)c_1(t) + c_2(t) &= w(r_1) = \sigma\kappa_1 \text{ and} \\ ln(r_2)c_1(t) + c_2(t) &= w(r_2) = \sigma\kappa_2 \end{aligned}$$

leads to

$$w(r,t) = \begin{cases} \sigma\kappa_2(t), & \text{if } r \in [0, r_2(t)], \\ \sigma\kappa_1(t) + \sigma(\kappa_2(t) - \kappa_1(t))\frac{ln(r_1(t)) - ln(r)}{ln(r_1(t)) - ln(r_2(t))}, & \text{if } r \in [r_2(t), r_1(t)], \\ \sigma\kappa_1(t), & \text{if } r \in [r_1(t), 1]. \end{cases}$$

Finally with the velocity equation in the above free boundary problem and differentiating w with respect to the variable normal direction, i.e. r, we obtain the evolution of the radii given by the following ODE system

$$\begin{aligned} \dot{r_1}(t) =& V \cdot \nu = \frac{1}{2}\left[\partial_\nu w(r,t)\right]_{\Gamma_1} \\ =& \frac{\sigma}{2}\left(0 - (\kappa_2(t) - \kappa_1(t))\frac{-\frac{1}{r_1(t)}}{ln(r_1(t)) - ln(r_2(t))}\right) \\ =& -\frac{\sigma}{2}\frac{1}{r_1(t)}\left(\frac{1}{r_1(t)} + \frac{1}{r_2(t)}\right)\frac{1}{ln(r_1(t)) - ln(r_2(t))}, \qquad (2.39) \\ \dot{r_2}(t) =& V \cdot \nu = \frac{1}{2}\left[\partial_\nu w(r,t)\right]_{\Gamma_2} \\ =& -\frac{\sigma}{2}\frac{1}{r_2(t)}\left(\frac{1}{r_1(t)} + \frac{1}{r_2(t)}\right)\frac{1}{ln(r_1(t)) - ln(r_2(t))}. \qquad (2.40) \end{aligned}$$

This setting provides an exact solution for the sharp interface model. Hence we can compare the solutions we obtain from the phase field model with varying parameters ε, γ and τ and check for convergence to the sharp interface model. Some simulation examples will be shown in the numerics section, see Section 6.1.

2.5 Non-constant mobility

Up to this point we assumed the diffusional mobility B to be a constant. Together with a scaling argument we used $B \equiv 1$. This section introduces the formulation of the model with non-constant mobility. The assumption of the constant mobility was introduced for ease of handling of the equations in the beginning of the development, but the model derivation by Cahn and Hilliard [CH58, Cah61] used non-constant mobility. As before we consider the Ginzburg-Landau energy functional

$$E(u) = \frac{\varepsilon\gamma}{2}\int\limits_\Omega |\nabla u|^2 + \frac{\gamma}{\varepsilon}\int\limits_\Omega \psi(u) + \frac{1}{\varepsilon}\int\limits_\Omega fu \qquad (2.41)$$

extended by a force term $f \in L^\infty(\Omega)$ like Puri, Binder and Dattagupta [PBD92], similar to the thermal fluctuation term used in [AKK10].
Again we take ψ to be the double obstacle potential given by $\psi(u) = \psi_0(u) + \iota_{[-1,1]}(u)$ as in (2.5). Similar to the preceding discussion the first variation of the energy is again given by means of the subdifferential and we get

$$w := \frac{\delta E}{\delta u} = -\varepsilon\gamma\Delta u + \frac{\gamma}{\varepsilon}\left(\psi_0'(u) + \mu\right) + \frac{1}{\varepsilon} f, \tag{2.42}$$

$$\mu \in \partial I_{[-1,1]}(u). \tag{2.43}$$

The constant diffusional mobility B in the flow equation is replaced by a concentration dependent function $B : \mathbb{R} \to \mathbb{R}_0^+$. Thus the mass flux

$$J = -B(u)\nabla w$$

together with the no-flux Neumann boundary condition $B(u)\partial_n w = 0$ plugged into the evolution equation $\partial_t u = -\nabla \cdot J$ leads, as in Section 2.1, to

$$\partial_t u - \nabla \cdot (B(u)\nabla w) = 0, \tag{2.44}$$

$$w + \varepsilon\gamma\Delta u - \frac{\gamma}{\varepsilon}(\psi_0'(u) + \mu) = \frac{1}{\varepsilon} f, \tag{2.45}$$

$$H(u,\mu) = \mu - \min(0, \mu + c(u+1)) - \max(0, \mu + c(u-1)) = 0. \tag{2.46}$$

Here we already replaced the usual complementarity condition (2.32)-(2.33) with its semi-smooth equivalent, see Remark 2.4. A typical choice for the diffusional mobility would be $B(u) = \max(0, 1-u^2)$ leading to the geometric evolution equations with surface diffusion in the sharp interface limit, see e.g. Cahn, Elliott and Novick-Cohen [CENC96], Taylor and Cahn [TC94], Barrett, Blowey and Garcke [BBG99] as well as the brief discussion in Section 2.6. As stated in these works, this choice is thermodynamically reasonable. In those situations atom movement is confined to the interfacial region and the flow is dominated by interface or surface diffusion [CENC96].
As suggested in [TC94] we use the parameter dependent mobility $B(u) = \frac{1}{\varepsilon} b(u)$ for some function b. This ensures insensitivity to the parameter ε. The asymptotic expansion carried out in [CENC96] (using the unscaled mobility B) shows that the time scale is of type $\varepsilon^2 t$. The modified variant we use results for obvious reasons in a εt time scale as in the Mullins–Sekerka equation for the constant mobility case, see also Pego [Peg89].
Up to this point the properties of b have been formulated somewhat vague. Now we will specify two separate assumptions we pose on the mobility function $B(u) = \frac{1}{\varepsilon} b(u)$ depending on the context.

Assumptions 2.7. *Let the diffusional mobility $b \in C^1(\mathbb{R})$ and $b_{max} \geq b_{min} > 0$ constants such that*

$$b_{max} \geq b(s) \geq b_{min} > 0 \quad \forall s \in \mathbb{R} \tag{2.47}$$

holds.

If b fulfills the above conditions, we also refer to the sitiuation as non-degenerate case with an explicit time discretization scheme. If the implicit time scheme is chosen we require additionally $b \in C^2(\mathbb{R})$ in the gradient flow setting later on, where we need to calculate a second derivative of the Lagrangian function for the formulation of a Newton method.

Assumptions 2.8. *If the diffusional mobility b is degenerate, we require*

$$b \in C([-1,1]),\ b(-1) = b(1) = 0 \text{ and } b(s) > 0 \quad \forall s \in (-1,1). \tag{2.48}$$

Results concerning the existence of solutions of the evolution equation with non-constant mobility have been shown for degenerate b in one dimension by Yin [Yin92]. For higher spatial dimension we refer to the proof and discussion by Elliott and Garcke [EG96] and references therein.

There they showed that in case of a degenerate mobility, i.e. that Assumption 2.8 holds, it is sufficient to solve the partial differential equations only on the set where b does not degenerate. Later, when fully discretizing the problem, we will call this set mobile set M. Due to the degeneracy the chemical potential w as well as the dual variable μ are only uniquely given on this set. Outside of this set w and μ could be chosen almost freely. As a result of the nature of this degenerate problem no uniqueness proof is known.

Elliott and Garcke [EG96] showed existence of solutions for degenerate (and non-degenerate) diffusional mobility together with a smooth free energy potential by approximating the degenerate case by non-degenerate equations. Elliott and Garcke also showed a priori estimates for the deep quench limit, i.e. the obstacle potential case, and thus the existence of a solutions, see Section 4.2 of [EG96].

As before we first discretize the evolution in time by employing an Euler-time discretization and subsequently obtain the elliptic system of partial differential equations

$$\frac{1}{\tau}(u - u^{n-1}) = \nabla \cdot \frac{1}{\varepsilon} b(u^*) \nabla w, \tag{2.49}$$

$$w + \varepsilon\gamma\Delta u - \frac{\gamma}{\varepsilon}(\psi_0'(u^*) + \mu) = \frac{1}{\varepsilon} f, \tag{2.50}$$

$$H(u,\mu) = 0. \tag{2.51}$$

Here u^* stands either for u or u^{n-1} depending on the chosen time discretization for those two terms. Note that this distinction is made due to the arising difficulties if using the implicit form due to either the non-convexity as well as the occurring nonlinearity respectively. We apply the semi-smooth Newton method of Qi and Sun [QS93] and Chen, Nashed and Qi [CNQ01] to solve this system in weak formulation, which is given by the semi-smooth function $F : H^1 \times H^1 \times H^1 \to (H^1)' \times (H^1)' \times (H^1)'$ by

$$F(u,w,\mu)(\varphi,\xi,\zeta) = \begin{pmatrix} \frac{\tau}{\varepsilon}\int_\Omega b(u^*)\nabla w \cdot \nabla\varphi + \int_\Omega u\varphi - \int_\Omega u^{n-1}\varphi \\ \int_\Omega w\xi - \varepsilon\gamma\int_\Omega \nabla u \cdot \nabla\xi - \frac{\gamma}{\varepsilon}\int_\Omega \psi_0'(u^*)\xi - \frac{\gamma}{\varepsilon}\int_\Omega \mu\xi - \frac{1}{\varepsilon}\int_\Omega f\xi \\ \int_\Omega (\mu - \max(0, \mu + c(u-1)) - \min(0, \mu + c(u+1)))\zeta \end{pmatrix}.$$

Thus we need to find a slanting function replacing the derivative. First we replace the terms containing $*$ by introducing some parameters Θ_b, $\Theta_\psi \in \{0,1\}$ to switch between implicit and semi-implicit discretizations resulting in the following formulation of the function

$$F(u,w,\mu)(\varphi,\xi,\zeta) = \\ \begin{pmatrix} \frac{\tau}{\varepsilon}\int_\Omega \big(\Theta_b b(u) + (1-\Theta_b)b(u^{n-1})\big)\nabla w\cdot\nabla\varphi + \int_\Omega (u-u^{n-1})\varphi \\ \int_\Omega (w - \frac{\gamma}{\varepsilon}(\Theta_\psi\psi_0'(u) + (1-\Theta_\psi)\psi_0'(u^{n-1}) + \mu) - \frac{1}{\varepsilon}f)\xi - \varepsilon\gamma\int_\Omega \nabla u\cdot\nabla\xi \\ \int_\Omega (\mu - \max(0,\mu + c(u-1)) - \min(0,\mu + c(u+1)))\zeta \end{pmatrix}.$$

The explicit discretization scheme, i.e. $\Theta_b = 0$ allows for both, the degenerate as well as the non-degenerate case. The implicit scheme with $\Theta_\psi = 1$ is only valid for the non-degenerate case.

The problem is that the set, where the equation for w degenerates, depends on the iterate u_k and thus on the step of the Newton iteration. Hence the new iterate w_{k+1} is given on the set, where $b(u_k)$ is positive. However, to match u_{k+1} we require w_{k+1} on the set, where $b(u_{k+1})$ is positive. As a consequence the resulting formulation would not be well posed.

Using this we can explicitly calculate the derivative, given by the slanting function associated to F, necessary for the semi-smooth Newton method. Differentiating we get the following slanting function, which we denote by DF.

Remark 2.9. *Let F be as above. Furthermore b shall fulfill Assumption 2.7 if $\Theta_b \in \{0,1\}$ or 2.8, if $\Theta_b = 0$. Then a formal derivation results in*

$$DF(u,w,\mu)(\varphi,\xi,\zeta)(\delta u,\delta w,\delta\mu) = \\ \begin{pmatrix} \frac{\tau}{\varepsilon}\int_\Omega (\Theta_b b(u) + (1-\Theta_b)b(u^{n-1}))\nabla\delta w\cdot\nabla\varphi + \frac{\tau}{\varepsilon}\Theta_b\int_\Omega b'(u)\delta u\nabla w\cdot\nabla\varphi + \int_\Omega \delta u\varphi \\ \int_\Omega \delta w\xi - \frac{\gamma}{\varepsilon}\Theta_\psi\int_\Omega \psi_0''(u)\delta u\xi - \varepsilon\gamma\int_\Omega \nabla\delta u\cdot\nabla\xi - \frac{\gamma}{\varepsilon}\int_\Omega \delta\mu\xi \\ -c\int_\Omega \chi_{\{x\in\Omega||u(x)+\frac{\mu(x)}{c}|>1\}}\delta u\zeta + \int_\Omega \chi_{\{x\in\Omega||u(x)+\frac{\mu(x)}{c}|<1\}}\delta\mu\zeta \end{pmatrix}$$

as slanting function for F at (u,w,μ).

Proof. The upper two components of F are classically differentiable and thus can easily be calculated. The third component is not but has already been dealt with before, when we formulated the method with constant mobility, see also [HIK02]. □

Now we have all ingredients to formally write down the semi-smooth algorithm.

Algorithm 2.2 *Semi-smooth Newton method* (SSN-I)

```
1. Set $k = 0$, initialize $u_k$, $w_k$, $\mu_k$.
2. Solve the coupled system of PDE's
       $DF(u_k, w_k, \mu_k)(\varphi, \xi, \zeta)(\delta u_k, \delta w_k, \delta\mu_k) = -F(u_k, w_k, \mu_k)(\varphi, \xi, \zeta).$
3. Set $w_{k+1} = w_k + \delta w_k$, $u_{k+1} = u_k + \delta u_k$ and $\mu_{k+1} = \mu_k + \delta\mu_k$.
4. If $\|\delta w_k\| + \|\delta u_k\| + \|\delta\mu_k\| \leq tol$ stop, otherwise set $k = k + 1$ and goto 2.
```

The simple stopping criterion in step 4 of Algorithm 2.2 can be replaced by a more sophisticated one. The fully discretized version of this algorithm, with $\Theta_b = 0$, is equivalent to a primal-dual active set method derived later, compare Algorithm 2.3 and 3.5. Note that this equivalence can be seen by simply using the splitting into active and inactive sets, necessary for the assembly of the derivative and subtracting most parts of the right hand side from the left side reverting back to an system of equations for u_{k+1}, w_{k+1} and μ_{k+1} in place of the updates.
In case of an implicit discretization of a non-degenerate mobility the primal-dual active set method and the semi-smooth Newton method result in different algorithms, see the brief discussion at the end of Section 2.5.3 or the more extensive presentation of the different discrete methods in Section 3.4.

2.5.1 Gradient flow formulation with non-constant mobility

Earlier in this chapter we reformulated the complementarity problem as a gradient flow. Now we present such a formulation with a non-constant but uniformly positive diffusional mobility. A short discussion of the connections of various sharp interface and diffuse surface motion laws is given by Taylor and Cahn [TC94]. As stated there the non-constant mobility leads to the formulation of the flow with a weighted scalar product. We use $\mathbf{Z} = H^{-1}(\Omega)$ as in in Section 2.2. The weight is given by the diffusional mobility. To simplify the notation we set $\rho = B(u^{n-1})$ or $\rho = B(u)$ depending on the type of time discretization chosen. The inner product on $\mathbf{Z}$ is now given by

$$(v_1, v_2)_{H_\rho^{-1}} := \int_\Omega \nabla(-\nabla \cdot \rho\nabla)^{-1} v_1 \cdot \rho\nabla(-\nabla \cdot \rho\nabla)^{-1} v_2, \tag{2.52}$$

where $y = (-\nabla \cdot \rho\nabla)^{-1} v$ is the weak solution of $-\nabla \cdot \rho\nabla y = v$ in Ω and $\rho\partial_n y = 0$ on $\partial\Omega$. Again as in the definition of the unweighted inner product (2.16) the solution is unique up to a constant, which we fix by $⨍_\Omega y = 0$. Additionally we set

$$\mathbf{U} := \left\{ u \in H^1(\Omega) \mid ⨍_\Omega u = m \right\}, \quad \mathbf{Y} := \left\{ u \in H^1(\Omega) \mid ⨍_\Omega u = 0 \right\} \tag{2.53}$$

and derive for $v_1, v_2 \in \mathbf{Y}$ again with the L^2-inner product $(.,.)$:

$$(v_1, v_2)_{H_\rho^{-1}} = (v_1, (-\nabla \cdot \rho\nabla)^{-1} v_2) = ((-\nabla \cdot \rho\nabla)^{-1} v_1, v_2)\,. \tag{2.54}$$

Thus we obtain for smooth free energy as in Section 2.2 the system

$$\partial_t u = (\nabla \cdot \rho\nabla) w\,, \tag{2.55}$$

$$w = -\varepsilon\gamma\Delta u + \frac{\gamma}{\varepsilon}\psi'(u) + \frac{1}{\varepsilon} f, \tag{2.56}$$

with boundary conditions $\partial_n u = \rho\partial_n w = 0$ on $\partial\Omega$.
Using the obstacle potential ψ from (2.5) together with the Ginzburg–Landau energy (2.41) and the backward Euler time discretization scheme the minimization problem (2.22) reads now as follows:

$$\min_{u \in \mathbf{U}} \left\{ \frac{\varepsilon\gamma}{2} \int_\Omega |\nabla u|^2 + \frac{\gamma}{\varepsilon} \int_\Omega \psi_0(u) + \frac{1}{\varepsilon} \int_\Omega f u + \frac{\tau}{2} \int_\Omega \nabla v \cdot \rho\nabla v \right\} \tag{2.57}$$

such that

$$\tau(\nabla \cdot \rho\nabla) v = u - u^{n-1},\ \rho\partial_n v = 0 \text{ on } \partial\Omega, \tag{2.58}$$

$$⨍_\Omega v = 0, \tag{2.59}$$

$$⨍_\Omega u = m, \tag{2.60}$$

$$|u| \le 1. \tag{2.61}$$

The associated Lagrangian is given as

$$\begin{aligned}\mathcal{L}(v, \kappa, u, w, \mu_+, \mu_-) :=& \frac{\varepsilon\gamma}{2} \int_\Omega |\nabla u|^2 + \frac{\gamma}{\varepsilon} \int_\Omega \psi_0(u) + \frac{1}{\varepsilon} \int_\Omega f u + \frac{\tau}{2} \int_\Omega \nabla v \cdot \rho\nabla v \\ & - \tau \int_\Omega \nabla w \cdot \rho\nabla v - \int_\Omega (u - u^{n-1}) w - \kappa \int_\Omega v \\ & - \frac{\gamma}{\varepsilon} \int_\Omega \mu_+(1-u) - \frac{\gamma}{\varepsilon} \int_\Omega \mu_-(1+u)\,.\end{aligned} \tag{2.62}$$

Before we derive the associated KKT-system we need to select the weighting function ρ, since it influences the derivatives if it depends on u or any of the other implicit variables.

2.5.2 Explicit discretization, $\rho = \frac{1}{\varepsilon}b(u^{n-1})$

The Euler time discretization allows for some choice for each occurring term. Quite often the difficult terms are discretized explicitly leading to a problem, which is more easily solvable than otherwise. Thus we discuss the explicitly discretized diffusional mobility first, i.e. we set $\rho = \frac{1}{\varepsilon}b(u^{n-1})$. We require a uniformly positive diffusional mobility, i.e. Assumption 2.7 holds. Due to the linearity of the problem the semi-smooth Newton method introduced above is equivalent to the primal-dual active set method we derive now. Let $\rho = B(u^{n-1}) = \frac{1}{\varepsilon}b(u^{n-1})$. Note that $b \in C([-1,1])$ is sufficient here, even for the non-degenerate case. It is essential that $|u^{n-1}| \leq 1$ holds, which is obviously true due to the usage of the obstacle potential. Thus no differentiability condition is imposed on b here. We obtain the KKT system

$$\frac{\tau}{\varepsilon}(\nabla \cdot b(u^{n-1})\nabla)(w-v) = \kappa \text{ in } \Omega,\ b(u^{n-1})\partial_n w = b(u^{n-1})\partial_n v \text{ on } \partial\Omega\,, \tag{2.63}$$

$$\fint_\Omega v = 0\,,\ \kappa = 0\,, \tag{2.64}$$

$$(u - u^{n-1}) = \frac{\tau}{\varepsilon}(\nabla \cdot b(u^{n-1})\nabla)v \text{ in } \Omega,\ b(u^{n-1})\partial_n v = 0 \text{ on } \partial\Omega\,, \tag{2.65}$$

$$w + \varepsilon\gamma\Delta u - \frac{\gamma}{\varepsilon}\psi_0'(u) - \frac{\gamma}{\varepsilon}\mu = \frac{1}{\varepsilon}f \text{ in } \Omega,\ \partial_n u = 0 \text{ on } \partial\Omega\,, \tag{2.66}$$

$$\mu := \mu_+ - \mu_-,\ \mu_+ \geq 0, \quad \mu_- \geq 0, \text{ a.e. in } \Omega\,, \tag{2.67}$$

$$\mu_+(u-1) = 0,\ \mu_-(u+1) = 0, \text{ a.e. in } \Omega\,, \tag{2.68}$$

$$\text{and } |u| \leq 1 \text{ a.e. in } \Omega\,. \tag{2.69}$$

Note that again equations (2.63), (2.65) and (2.66) have to be understood in their weak form. Now we continue by repeating the formal steps from Section 2.3. First we can analogously eliminate the variable v by means of (2.63) and (2.64). We again introduce for $c > 0$ the active sets

$$\begin{aligned} A^+ &= \left\{ x \in \Omega \mid u(x) + \frac{\mu(x)}{c} > 1 \right\}, \\ A^- &= \left\{ x \in \Omega \mid u(x) + \frac{\mu(x)}{c} < -1 \right\} \end{aligned}$$

and the inactive set $I := \Omega \backslash (A^+ \cup A^-)$. Finally we get the following formal algorithm for the non-constant mobility with explicit time discretization of the diffusional mobility term.

Algorithm 2.3 *Formal primal-dual active set algorithm with non-constant diffusional mobility* (mPDAS-I)

1. Set $k = 0$, u_0. Initialize $A_0^\pm$ and define $I_0 = \Omega \setminus (A_0^+ \cup A_0^-)$.
2. Set $u_k = \pm 1$ on $A_k^\pm$ and $\mu_k = 0$ on I_k.
3. Solve the coupled system of PDE's (2.65), (2.66) to obtain u_k on I_k, μ_k on $A_k^+ \cup A_k^-$ and w_k on Ω.
4. Set $A_{k+1}^+ := \left\{x \in \Omega \mid u_k(x) + \frac{\mu_k(x)}{c} > 1\right\}$,

   $A_{k+1}^- := \{x \in \Omega \mid u_k(x) + \frac{\mu_k(x)}{c} < -1\}$ and

   $I_{k+1} := \Omega \setminus (A_{k+1}^+ \cup A_{k+1}^-)$.
5. If $A_{k+1}^\pm = A_k^\pm$ stop, otherwise set $k = k + 1$ and goto 2.

Remark 2.10. *If b fulfills Assumption 2.8, i.e. the diffusional mobility is degenerate, the equations (or inequalities) only define w and v as well as μ on the set, where b is positive, compare Elliott and Garcke [EG96]. Thus a formulation of a gradient flow and subsequently of the primal-dual active set method is not possible in the countinuous case. However, in the fully discretized setting we will make use of this property to state a primal-dual active set method, which is also valid for the degenerate case.*

2.5.3 Implicit discretization, $\rho = \frac{1}{\varepsilon} b(u)$

This choice is only reasonable if b is non-degenerate in the sense of Assumption 2.7. The derivation of the KKT system is similar to the explicit case discussed before. The only real difference lies in equation (2.66), the variation of $\mathcal{L}$ in u. In the implicit formulation we obtain

$$w + \varepsilon\gamma\Delta u - \frac{\gamma}{\varepsilon}\psi_0'(u) - \frac{\gamma}{\varepsilon}\mu + \frac{\tau}{2\varepsilon}\nabla v \cdot b'(u)\nabla v - \frac{\tau}{\varepsilon}\nabla w \cdot b'(u)\nabla v = \frac{1}{\varepsilon} f \text{ in } \Omega \tag{2.70}$$

together with the Neumann boundary condition $\partial_n u = 0$ on $\partial\Omega$. Furthermore we have to substitute the equations, where only the dependency of b changes. Hence in place of (2.63) we use

$$\frac{\tau}{\varepsilon}(\nabla \cdot b(u)\nabla)(w - v) = \kappa \text{ in } \Omega,\ b(u)\partial_n w = b(u)\partial_n v \text{ on } \partial\Omega \tag{2.71}$$

and instead of (2.65) we use

$$(u - u^{n-1}) = \frac{\tau}{\varepsilon}(\nabla \cdot b(u)\nabla)v \text{ in } \Omega,\ b(u)\partial_n v = 0 \text{ on } \partial\Omega\,. \tag{2.72}$$

Now we can state a primal-dual active set method analogously to the previous discussion. Note that using the identity $v = w - \fint_\Omega w$ equation (2.70) is reduced to

$$w + \varepsilon\gamma\Delta u - \frac{\gamma}{\varepsilon}\psi_0'(u) - \frac{\gamma}{\varepsilon}\mu - \frac{\tau}{2\varepsilon}\nabla w \cdot b'(u)\nabla w = \frac{1}{\varepsilon}f \text{ in } \Omega,\ \partial_n u = 0 \text{ on } \partial\Omega\,. \qquad (2.73)$$

We get Algorithm 2.3, where the PDE's in step 3 are replaced by (2.72) and (2.73). Due to the nonlinearity of those equations this primal-dual active set method is no longer equivalent to a Newton type method. Note that in comparison to Algorithm 2.2, the only difference is the nonlinear τ and b' dependent term in (2.73). Thus, when considering the limit $\tau \to 0$ both methods are similar and lead to consistent discretizations of the PDE. Due to the larger computational effort of solving a nonlinear problem in each primal-dual active set iteration, this method is probably not competitive and is thus omitted. For a consideration of an implicitly discretized diffusional mobility we present a Lagrange-Newton method, which leads again to an algorithm of a structure, which is similar to the primal-dual active set methods before.

2.5.4 Newton method with implicit discretization

The fully implicit discretization of the model with non-constant mobility leads either to a Newton-type method, where a non-symmetric linear problem for the update remained, or to a primal-dual active set method with an embedded nonlinear problem, discussed above in Section 2.5.3. This paragraph will use a semi-smooth Newton method to calculate critical points of the KKT system associated to the Lagrangian function given in (2.62). Please note that the below discussion is somewhat formal since we omit the discussion of regularity. We expect the arising system to be symmetric, due to the fact that it is a second derivative. Similarly to before, when calculating the first order conditions, we get $w - \fint_\Omega w = v$ and $\kappa = 0$. Thus we eliminate v and κ from the problem formulation. Lets recall the Lagrange function (2.62) we want to find critical points of. For the remainder of this discussion we use a reduced formulation given by

$$\begin{aligned}\mathcal{L}(u, w, \mu_+, \mu_-) :=& \frac{\varepsilon\gamma}{2}\int\limits_\Omega |\nabla u|^2 + \frac{\gamma}{\varepsilon}\int\limits_\Omega \psi_0(u) + \frac{1}{\varepsilon}\int\limits_\Omega fu - \frac{\tau}{2}\int\limits_\Omega |\nabla w|^2 b(u) \\ &- \int\limits_\Omega (u - u^{n-1})w - \frac{\gamma}{\varepsilon}\int\limits_\Omega \mu_+(1-u) - \frac{\gamma}{\varepsilon}\int\limits_\Omega \mu_-(1+u)\end{aligned} \qquad (2.74)$$

together with the restriction on the multipliers $\mu_+, \mu_- \geq 0$. Note that both τ dependent terms have been summed up. We use a Lagrange–Newton method for the calculation of the critical points, compare e.g. Geiger and Kanzow [GK02] or the semi-smooth versions of De Luca, Facchinei and Kanzow [DLFK96] or Facchinei, Fischer and Kanzow [FFK98]. Basically we use a semi–smooth Newton method to solve the KKT problem, where the complementarity condition is replaced by the semi-smooth formulation (2.37). The KKT system associated with (2.74) is given by the derivatives von $\mathcal{L}$ in the directions of ζ_u and ζ_w, given by

$$\varepsilon\gamma \int_\Omega \nabla u \cdot \nabla \zeta_u + \int_\Omega \left(\frac{\gamma}{\varepsilon}(\psi_0'(u) + \mu_+ - \mu_-) - w - \frac{\tau}{2\varepsilon}|\nabla w|^2 b'(u) + \frac{1}{\varepsilon} f \right) \zeta_u = 0,$$
$$-\frac{\tau}{\varepsilon} \int_\Omega b(u) \nabla w \cdot \nabla \zeta_w - \int_\Omega (u - u^{n-1}) \zeta_w = 0$$

together with the complementarity conditions

$$\mu_+ \geq 0,\ (1-u) \geq 0,\ \mu_+(1-u) = 0,$$
$$\mu_- \geq 0,\ (1+u) \geq 0,\ \mu_-(1+u) = 0.$$

The above multiplicators can be replaced by one single variable by setting $\mu := \mu_+ - \mu_-$. We then denote the above directional derivatives by

$$\mathcal{L}_u(u,w,\mu)[\zeta_u] := \varepsilon\gamma \int_\Omega \nabla u \cdot \nabla \zeta_u + \int_\Omega \left(\frac{\gamma}{\varepsilon}(\psi_0'(u) + \mu) - w - \frac{\tau}{2\varepsilon}|\nabla w|^2 b'(u) + \frac{1}{\varepsilon} f \right) \zeta_u,$$
$$\mathcal{L}_w(u,w,\mu)[\zeta_w] := -\frac{\tau}{\varepsilon} \int_\Omega b(u) \nabla w \cdot \nabla \zeta_w - \int_\Omega (u - u^{n-1}) \zeta_w,$$

The complementarity condition can be rephrased, compare Remark 2.4, as

$$\mathcal{L}_\mu(u,w,\mu)[\zeta_\mu] := \int_\Omega (\mu - \min(0, \mu + c(u+1)) - \max(0, \mu + c(u-1)))\, \zeta_\mu.$$

The semi-smooth Newton-method is now applied to the system

$$F(u,w,\mu)[\zeta_u, \zeta_w, \zeta_\mu] := \mathcal{L}_u(u,w,\mu)[\zeta_u] + \mathcal{L}_w(u,w,\mu)[\zeta_w] + \mathcal{L}_\mu(u,w,\mu)[\zeta_\mu] = 0.$$

To obtain a slanting function for this system we define the bilinear forms given by the second derivatives. Calculating the partial derivatives of $\mathcal{L}_u(u,w,\mu)[\zeta_u]$ in the directions δu, δw and $\delta\mu$ we get

$$\mathcal{L}_{u,u}(u,w,\mu)[\zeta_u][\delta u] := \varepsilon\gamma \int_\Omega \nabla \delta u \cdot \nabla \zeta_u + \frac{\gamma}{\varepsilon} \int_\Omega \psi_0''(u) \delta u \zeta_u - \frac{\tau}{2\varepsilon} \int_\Omega |\nabla w|^2 b''(u) \delta u \zeta_u,$$
$$\mathcal{L}_{u,w}(u,w,\mu)[\zeta_u][\delta w] := -\int_\Omega \delta w \zeta_u - \frac{\tau}{\varepsilon} \int_\Omega \nabla w \cdot b'(u) \zeta_u \nabla \delta w,$$
$$\mathcal{L}_{u,\mu}(u,w,\mu)[\zeta_u][\delta\mu] := \frac{\gamma}{\varepsilon} \int_\Omega \delta\mu \zeta_u.$$

The second part of the equation can also be calculated by classical methods resulting in

$$\begin{aligned}
\mathcal{L}_{w,u}(u,w,\mu)[\zeta_w][\delta u] &:= -\frac{\tau}{\varepsilon}\int\limits_\Omega \nabla w\cdot b'(u)\delta u\nabla\zeta_w - \int\limits_\Omega \delta u\zeta_w,\\
\mathcal{L}_{w,w}(u,w,\mu)[\zeta_w][\delta w] &:= -\frac{\tau}{\varepsilon}\int\limits_\Omega \nabla\delta w\cdot b(u)\nabla\zeta_w,\\
\mathcal{L}_{w,\mu}(u,w,\mu)[\zeta_w][\delta\mu] &:= 0.
\end{aligned}$$

For the third part we again can only define a slanting function and not a derivative in the classical sense due to the occurrence of the max and min functions. We introduce the sets

$$\begin{aligned}
A^+ &:= \{x\in\Omega \mid \mu(x)+c(u(x)-1)>0\},\\
A^- &:= \{x\in\Omega \mid \mu(x)+c(u(x)+1)<0\}
\end{aligned}$$

and $I := \Omega\setminus(A^+\cup A^-)$ afresh and get as before in the case with constant mobility:

$$\begin{aligned}
\mathcal{L}_{\mu,u}(u,w,\mu)[\zeta_\mu][\delta u] &:= -c\int\limits_\Omega \chi_{A^+\cup A^-}\delta u\zeta_\mu,\\
\mathcal{L}_{\mu,w}(u,w,\mu)[\zeta_\mu][\delta w] &:= 0,\\
\mathcal{L}_{\mu,\mu}(u,w,\mu)[\zeta_\mu][\delta\mu] &:= \int\limits_\Omega \chi_I\delta\mu\zeta_\mu.
\end{aligned}$$

Thus the update step for the Newton-iteration at a given iteration step $k\geq 0$ with iterate (u_k,w_k,μ_k) is given by

$$\begin{aligned}
-F(u_k,w_k,\mu_k) =& DF(u_k,w_k,\mu_k)(\delta u_k,\delta w_k,\delta\mu_k) \qquad (2.75)\\
:=& \mathcal{L}_{u,u}[\zeta_u][\delta u_k] + \mathcal{L}_{u,w}[\zeta_u][\delta w_k] + \mathcal{L}_{u,\mu}[\zeta_u][\delta\mu_k] + \mathcal{L}_{w,u}[\zeta_w][\delta u_k]\\
&+ \mathcal{L}_{w,w}[\zeta_w][\delta w_k] + \mathcal{L}_{\mu,u}[\zeta_\mu][\delta u_k] + \mathcal{L}_{\mu,\mu}[\zeta_\mu][\delta\mu_k],
\end{aligned}$$

where we omitted the dependency on u_k, w_k, μ_k of the forms $\mathcal{L}_{xx}$ for better readability and used $\delta u_k := u_{k+1} - u_k$ as notation for the update step. Define δw_k and $\delta\mu_k$ analogously. Note that we could reduce the system further, using the split into active and inactive sets motivated from the primal-dual active set methods. Just considering the third row of (2.75) we get u on the inactive and μ on the active set. We execute this in detail in the discrete setting in Section 3.4.

2.6 Surface diffusion

Previously we commented on the connection of the Cahn–Hilliard evolution to a sharp interface model, namely the Mullins–Sekerka model, see Section 2.4. Using formal asymptotic analysis diffuse phase field models can often be associated with a geometric evolution equation. The Cahn–Hilliard problem with degenerate diffusional mobility B, corresponds in this sense to the geometric motion law given by

$$V = -\frac{\pi^2}{16}\Delta_S \kappa. \tag{2.76}$$

Note that V is the normal velocity of a moving surface $\Gamma(t)$, κ the mean curvature of the surface. Furthermore Δ_S denotes the surface Laplacian. For the complete formal asymptotic analysis and derivation of the above motion law, see Cahn, Elliott and Novick-Cohen [CENC96]. Both the Mullins–Sekerka flow and the motion by surface diffusion are volume preserving and perimeter decreasing. The main difference, as stated in Elliott and Garcke [EG97], is that the Mullins–Sekerka flow is non-local, i.e. the velocity in each point on Γ_t depends on data away from this point. Due to the degeneracy of the mobility, the diffusion process is restricted to the interfacial region resulting in a motion by surface diffusion, see Mullins [Mul57].

Since surface diffusion is often used in many applications the development of fast and efficient numerical methods have been of interest. Some numerical methods concerned with the simulation of the phase field model have already been mentioned in the introductory part of this chapter. Additionally there is a variety of numeric methods for the sharp interface model, i.e. geometric evolution equations, see e.g. Barrett, Garcke and Nürnberg [BGN07], Bänsch, Morin and Nochetto [BMN05] or Deckelnick, Dziuk and Elliott [DDE05].

We present some numerical results in Section 6.4 and compare them to the afore mentioned publications.

2.7 Existence and uniqueness of minimizers

Before fully discretizing the model, we will present some results showing existence of minimizers and give conditions for their uniqueness for constant as well as explicitly discretized non-constant but uniformly positive mobility, i.e. we require Assumption 2.7 to hold. Note that simply choosing $\rho \equiv 1$ includes the constant mobility case, discussed in [BBG11].

From now on we consider the choice $\psi_0(u) = \frac{1}{2}(1-u^2)$, $f \in L^\infty(\Omega)$ and show that the KKT system, given either by (2.23)-(2.27) or (2.57)-(2.61) respectively, is solvable. Defining the admissible set

$$\mathbf{U_{ad}} := \left\{ u \in H^1_\rho(\Omega) \mid |u| \le 1, \ \fint_\Omega u = m \right\} \tag{2.77}$$

the minimization problem can be reformulated. For given $u^{n-1} \in \mathbf{U_{ad}}$ find

$$\begin{aligned} \min_{u \in \mathbf{U_{ad}}} E(u) := & \frac{\varepsilon\gamma}{2}\int_\Omega |\nabla u|^2 + \frac{\gamma}{2\varepsilon}\int_\Omega (1-u^2) + \frac{1}{\varepsilon}\int_\Omega fu \\ & + \frac{1}{2\tau}\|\nabla(-\nabla\cdot\rho\nabla)^{-1}(u-u^{n-1})\|^2_{L^2_\rho}. \end{aligned} \tag{2.78}$$

Lemma 2.11. *The minimization problem* (2.78) *has a solution.*

Proof. Since $|u| \leq 1$, we obtain $\int_\Omega \psi_0(u)\ dx = \int_\Omega \frac{1}{2}(1-u^2)\ dx$ is non-negative. Due to $⨍_\Omega (u-m)\ dx = 0$ for all $u \in \mathbf{U_{ad}}$, we can use Poincaré's inequality for functions with mean value zero, see e.g. Adams and Fournier [AF03], and Young's inequality to obtain

$$\begin{aligned}\frac{1}{\varepsilon}\int_\Omega fu\ dx &\geq -\frac{1}{\varepsilon}\|f\|_{L^2}\|u\|_{L^2} \geq -\frac{1}{\varepsilon}\|f\|_{L^2}(C_p\|\nabla u\|_{L^2}+1)\\ &\geq -\frac{1}{\varepsilon}\|f\|_{L^2} - \frac{C_p^2}{\varepsilon^2}\delta\|f\|_{L^2}^2 - \frac{1}{4\delta}\|\nabla u\|_{L^2}^2\\ &= -\frac{1}{\varepsilon}\|f\|_{L^2} - \frac{C_p^2}{\varepsilon^3\gamma}\|f\|_{L^2}^2 - \frac{\varepsilon\gamma}{4}\|\nabla u\|_{L^2}^2,\end{aligned}$$

where $\delta = \frac{1}{\varepsilon\gamma}$ was chosen. Thus the energy is bounded from below and there exists a minimizing sequence $(u_k)_{k\in\mathbb{N}} \subset \mathbf{U_{ad}}$ for E, i.e.

$$E(u_k) \to \inf_{u\in\mathbf{U_{ad}}} E(u) \geq -C \quad \text{for } k \to \infty.$$

Given that $(E(u_k))_{k\in\mathbb{N}}$ is uniformly bounded, we can conclude that $\int_\Omega |\nabla u_k|^2\ dx$ is uniformly bounded and again by the Poincaré inequality we get $(u_k)_{k\in\mathbb{N}}$ is a bounded sequence in $H^1(\Omega)$. Using the fact that bounded sequences in $H^1(\Omega)$ have weakly converging subsequences and applying Rellich's theorem we obtain the existence of a subsequence such that

$$u_{k_j} \rightharpoonup u^* \text{ in } H^1(\Omega),\ u_{k_j} \to u^* \text{ in } L^2(\Omega) \text{ for } j \to \infty. \tag{2.79}$$

Since the terms $\int_\Omega |\nabla u|^2\ dx$ and $\int_\Omega |\nabla(-\nabla\cdot\rho\nabla)^{-1}u|^2\ dx$ are convex, we obtain that they are weakly lower semi-continuous in $H^1(\Omega)$, see e.g. Evans [Eva10]. Since $\int_\Omega \psi_0(u_{k_j})\ dx$ and $\int_\Omega f u_{k_j}$ converge strongly we conclude that u^* is in fact a minimum of E in $\mathbf{U_{ad}}$. □

Depending on the chosen time step size we can show uniqueness of the minimizer. This is the content of the following lemma.

Lemma 2.12. *Let* $0 < \rho \leq \rho_{max}$ *uniformly bounded. The solution of* (2.78) *is unique if* $\tau \in (0, \frac{4\varepsilon^3}{\gamma\rho_{max}})$.

Proof. We obtain uniqueness of the solution from strict convexity of E. The functional E is strictly convex on $\mathbf{U}$ if and only if $F(\eta) := E(\eta + u^{n-1})$ is strictly convex on $\mathbf{Y}$ given by (2.53). Since F is the sum of terms which are constant or linear and of

$$\hat{F}(\eta) := \frac{\varepsilon\gamma}{2}\int_\Omega |\nabla\eta|^2 - \frac{\gamma}{2\varepsilon}\int_\Omega \eta^2 + \frac{1}{2\tau}\|\nabla(-\nabla\cdot\rho\nabla)^{-1}\eta\|_{L^2_\rho}^2, \tag{2.80}$$

it is sufficient to show that $\hat{F}$ is strictly convex on $\mathbf{Y} \setminus \{0\}$. Using the definition of $(\nabla \cdot \rho \nabla)^{-1}$ as well as the Cauchy-Schwarz inequality for the L^2_ρ scalar product together with Young's inequality we obtain for all $\eta \in \mathbf{Y}$

$$
\begin{aligned}
\int\limits_\Omega \eta^2 &= \int\limits_\Omega (\nabla(-\nabla \cdot \rho \nabla)^{-1} \eta) \cdot \rho \nabla \eta \\
&\leq \left(\int\limits_\Omega \nabla(-\nabla \cdot \rho\nabla)^{-1}\eta) \cdot \rho \nabla(-\nabla \cdot \rho \nabla)^{-1}\eta \right)^{1/2} \left(\int\limits_\Omega \nabla \eta \cdot \rho \nabla \eta \right)^{1/2} \\
&\leq \frac{\delta}{2} \int\limits_\Omega \nabla(-\nabla \cdot \rho\nabla)^{-1}\eta) \cdot \rho \nabla(-\nabla \cdot \rho \nabla)^{-1}\eta + \frac{1}{4\delta} \int\limits_\Omega \nabla \eta \cdot \rho \nabla \eta \\
&\leq \frac{\delta}{2} \|\nabla(-\nabla \cdot \rho \nabla)^{-1}\eta\|^2_{L^2_\rho} + \frac{\rho_{max}}{2\delta} \int\limits_\Omega |\nabla \eta|^2.
\end{aligned}
$$

Choosing $\delta = \frac{2\varepsilon}{\tau\gamma}$ we finally get

$$
\hat{F}(\eta) \geq \frac{\gamma}{8\varepsilon^2} \left(4\varepsilon^3 - \gamma \tau \rho_{max}\right) \int\limits_\Omega |\nabla \eta|^2. \tag{2.81}
$$

Since all three terms of $\hat{F}$ are quadratic we can use

$$
(tx + (1-t)y)^2 = tx^2 + (1-t)y^2 - t(1-t)(x-y)^2
$$

to obtain

$$
\hat{F}(t\eta_1 + (1-t)\eta_2) = t\hat{F}(\eta_1) + (1-t)\hat{F}(\eta_2) - \underbrace{t(1-t)}_{>0} \hat{F}(\eta_1 - \eta_2)
$$

for $t \in (0,1)$. Thus if $\tau < \frac{4\varepsilon^3}{\gamma \rho_{max}}$ we obtain strict convexity of $\hat{F}$ and hence the assertion. □

Remark 2.13. *When we consider an explicit discretized free energy term, i.e. $\Theta_\psi = 0$, we don't need the above restriction on the time step, given above due to the concave part given by the free energy potential included in the energy functional.*

We can also show that the solutions we get are the same as for the variational problem formulation.

Lemma 2.14. *A solution $u \in \mathbf{U_{ad}}$ of (2.78) solves the variational inequality*

$$
\gamma\varepsilon \int\limits_\Omega \nabla u \cdot \nabla(\eta - u) - \frac{\gamma}{\varepsilon} \int\limits_\Omega u(\eta - u) + \frac{1}{\varepsilon} \int\limits_\Omega f(\eta - u) + \frac{1}{\tau} \int\limits_\Omega (-\nabla \cdot \rho \nabla)^{-1}(u - u^{n-1})(\eta - u) \geq 0 \tag{2.82}
$$

for all $\eta \in \mathbf{U_{ad}}$.

Proof. We compute the first derivation of (2.78) in a direction $(\eta - u)$ for arbitrary $\eta \in \mathbf{U_{ad}}$ and obtain by (2.54)

$$\begin{aligned}\frac{d}{d\delta}E(u+\delta(\eta-u))\Big|_{\delta=0} &= \gamma\varepsilon\int_\Omega \nabla u\cdot\nabla(\eta-u) - \frac{\gamma}{\varepsilon}\int_\Omega u(\eta-u) + \frac{1}{\varepsilon}\int_\Omega f(\eta-u)\\ &\quad + \frac{1}{\tau}\int_\Omega \nabla(-\nabla\cdot\rho\nabla)^{-1}(u-u^{n-1})\cdot\rho\nabla(-\nabla\cdot\rho\nabla)^{-1}(\eta-u)\\ &= \gamma\varepsilon\int_\Omega \nabla u\cdot\nabla(\eta-u) - \frac{\gamma}{\varepsilon}\int_\Omega u(\eta-u) + \frac{1}{\varepsilon}\int_\Omega f(\eta-u)\\ &\quad + \frac{1}{\tau}\int_\Omega (-\nabla\cdot\rho\nabla)^{-1}(u-u^{n-1})(\eta-u).\end{aligned}$$

If u is a minimizer of (2.78), then the derivative of the functional has to be non-negative in all directions $\eta - u$ with $\eta \in \mathbf{U_{ad}}$ and the assertion follows. □

The following lemma gives the existence of a Lagrange multiplier for the equality constraint $⨍_\Omega u = m$.

Lemma 2.15. *Let $u \in \mathbf{U_{ad}}$ be a solution of the variational inequality* (2.82). *Then there exists a $\lambda \in \mathbb{R}$ such that for all $\eta \in H^1(\Omega)$ with $|\eta| \le 1$ the inequality*

$$\begin{aligned}&\gamma\varepsilon\int_\Omega \nabla u\cdot\nabla(\eta-u) - \frac{\gamma}{\varepsilon}\int_\Omega u(\eta-u) + \frac{1}{\varepsilon}\int_\Omega f(\eta-u)\\ &+\frac{1}{\tau}\int_\Omega (-\nabla\cdot\rho\nabla)^{-1}(u-u^{n-1})(\eta-u) - \lambda\int_\Omega(\eta-u) \ge 0\end{aligned}\tag{2.83}$$

holds.

Proof. We argue similar as in the proof of Proposition 3.3 in Blowey and Elliott [BE91]. Let $g := \frac{2\gamma}{\varepsilon}u - \frac{1}{\tau}(-\nabla\cdot\rho\nabla)^{-1}(u-u^{n-1}) - \frac{1}{\varepsilon}f$. Since the absolute values of u and u^{n-1} are bounded by one we obtain from the theory of elliptic equations and the fact that $f \in L^\infty$ that g is bounded in $L^\infty(\Omega)$. We define for each $\alpha \in \mathbb{R}$ a function $u_\alpha \in K := \{u \in H^1(\Omega) \mid |u| \le 1\}$ such that for all $\eta \in K$

$$\gamma\varepsilon\int_\Omega \nabla u_\alpha\cdot\nabla(\eta-u_\alpha) + \frac{\gamma}{\varepsilon}\int_\Omega u_\alpha(\eta-u_\alpha) - \int_\Omega g(\eta-u_\alpha) - \alpha\int_\Omega(\eta-u_\alpha) \ge 0. \tag{2.84}$$

Using standard theory of variational inequalities we deduce that (2.84) has a unique solution $u_\alpha \in K$, see e.g. Kinderlehrer and Stampacchia [KS80]. We now introduce a function $M : \mathbb{R} \to \mathbb{R}$ by

$$M(\alpha) := ⨍_\Omega u_\alpha\, dx.$$

For all $\eta \in K$ and all $\alpha \in \mathbb{R}$ we have the pointwise inequality

$$(\underbrace{\frac{\gamma}{\varepsilon} - g}_{\leq \frac{\gamma}{\varepsilon} + \|g\|_\infty} - \alpha)\underbrace{(\eta - 1)}_{\leq 0} \geq (\frac{\gamma}{\varepsilon} + \|g\|_\infty - \alpha)(\eta - 1)$$

as well as

$$(\underbrace{\frac{\gamma}{\varepsilon} - g}_{\geq -\frac{\gamma}{\varepsilon} - \|g\|_\infty} - \alpha)\underbrace{(\eta + 1)}_{\geq 0} \geq (-\frac{\gamma}{\varepsilon} - \|g\|_\infty - \alpha)(\eta + 1).$$

Inserting $\eta \equiv \pm 1$ into (2.84) we obtain that $u \equiv 1$ is a solution of (2.84) if $\alpha \geq \frac{\gamma}{\varepsilon} + \|g\|_\infty$ and $u \equiv -1$ is a solution of (2.84) if $\alpha \leq -(\frac{\gamma}{\varepsilon} + \|g\|_\infty)$. Thus $M(\pm(\frac{\gamma}{\varepsilon} + \|g\|_\infty)) = \pm 1$. Similar to [BE91] we show that M is monotone and continuous. Let $\alpha_1, \alpha_2 \in \mathbb{R}$, set $\alpha = \alpha_1$, $\eta = u_{\alpha_2}$ and $\alpha = \alpha_2$, $\eta = u_{\alpha_1}$ in (2.84). Adding the resulting inequalities we obtain

$$\gamma\varepsilon \int_\Omega |\nabla(u_{\alpha_1} - u_{\alpha_2})|^2 + \frac{\gamma}{\varepsilon} \int_\Omega |u_{\alpha_1} - u_{\alpha_2}|^2 \leq |\Omega|(M(\alpha_1) - M(\alpha_2))(\alpha_1 - \alpha_2), \quad (2.85)$$

which shows in particular that M is monotone. From the Cauchy-Schwarz inequality we get additionally the estimate

$$|M(\alpha_1) - M(\alpha_2)|^2 |\Omega| \leq \|u_{\alpha_1} - u_{\alpha_2}\|_{L^2}^2 \overset{(2.85)}{\leq} \frac{\varepsilon}{\gamma}|\Omega| \, |M(\alpha_1) - M(\alpha_2)| \, |\alpha_1 - \alpha_2|.$$

After cancellation we obtain

$$|M(\alpha_1) - M(\alpha_2)| \leq \frac{\varepsilon}{\gamma}|\alpha_1 - \alpha_2|$$

and have shown both properties of M. Using the intermediate value theorem we get the existence of a $\lambda \in \mathbb{R}$ such that $M(\lambda) = m$. We now choose $\eta = u_\lambda$ in (2.82) giving

$$\gamma\varepsilon \int_\Omega \nabla u \cdot \nabla(u - u_\lambda) - \frac{\gamma}{\varepsilon} \int_\Omega u(u - u_\lambda) + \frac{1}{\varepsilon} \int_\Omega f(u - u_\lambda) + \frac{1}{\tau} \int_\Omega (-\nabla \cdot \rho \nabla)^{-1}(u - u^{n-1})(u - u_\lambda) \leq 0$$

and $\eta = u$ in (2.84), where we set $\alpha = \lambda$ and multiply by -1, yielding

$$-\gamma\varepsilon \int_\Omega \nabla u_\lambda \cdot \nabla(u - u_\lambda) - \frac{\gamma}{\varepsilon} \int_\Omega u_\lambda(u - u_\lambda) + \int_\Omega g(u - u_\lambda) + \lambda \int_\Omega (u - u_\lambda) \leq 0.$$

Adding both resulting terms leads to

$$\gamma\varepsilon \int_\Omega |\nabla(u - u_\lambda)|^2 + \frac{\gamma}{\varepsilon} \int_\Omega |u - u_\lambda|^2 \leq 0,$$

where we use the fact that $\int_\Omega (u - u_\lambda) = 0$ due to $⨍_\Omega u = ⨍_\Omega u_\lambda = m$. Hence $u = u_\lambda$. Using this result and the definition of g we conclude from (2.84) that u fulfills (2.83). □

Using regularity theory for obstacle problems we obtain similar as in the proof of Lemma 3.2 in Blowey and Elliott [BE91]

$$u \in W^{2,p}_{loc}(\Omega) \text{ for all } p \in (1,\infty), \quad u \in C^{1,\alpha}(\Omega) \text{ for all } \alpha \in (0,1).$$

Starting out from this point we get the existence of a solution (u, v, w, μ) of the KKT system (2.63)-(2.69), (2.28)-(2.34) respectively, as for other optimization problems with bilateral constraints, see e.g. Tröltzsch [Trö10], by setting

$$\begin{aligned}
v \quad &= -(-\nabla \cdot \rho \nabla)^{-1} \left(\frac{u - u^{n-1}}{\tau} \right), \, w = v + \lambda, \\
\mu_+ \quad &= \varepsilon(\gamma\varepsilon\Delta u + \frac{\gamma}{\varepsilon} u + w)^+ = \varepsilon \max(\gamma\varepsilon\Delta u + \frac{\gamma}{\varepsilon} u + w, 0), \\
\mu_- \quad &= \varepsilon(\gamma\varepsilon\Delta u + \frac{\gamma}{\varepsilon} u + w)^- = \varepsilon \max(-\gamma\varepsilon\Delta u - \frac{\gamma}{\varepsilon} u - w, 0), \\
\mu \quad &= \mu_+ - \mu_-.
\end{aligned}$$

Chapter 3

Discretization

The primal-dual active set or semi-smooth Newton method is not applicable in function space due to the low regularity of the Lagrangian multiplier μ, see Remark 2.5. It is possible to deal with this problem by using regularization methods, see e.g. Hintermüller, Hinze and Tber [HHT10]. However the fully discretized problem does not suffer from this drawback and thus we omit the addition of a regularization term. This chapter discussing the discrete problems is structured as follows. First we will give a brief introduction to the linear finite element spaces used and establish some notation. Then we present the disretization of the variational inequality formulation of the Cahn-Hilliard problem (2.10)-(2.11). Problems of such type can be handled by a projection type successive over-relaxation solver, see e.g. Barrett, Nürnberg and Styles [BNS04] for a similiar problem. We use this established method to test our algorithm, resulting from the discretization of the primal-dual active set method. Following that we present discrete versions of the primal-dual active set method with constant and non-constant diffusional mobility. After we state a discrete version of the Lagrange-Newton method given in Section 2.5.4, we discuss various smaller implementational topics, like the generation of the adaptive meshes, the selection of the parameter c or the initialization of the active sets. Finally, in Section 3.7, we proof existence and uniqueness of solutions of the discrete primal-dual active set methods under certain assumptions.

3.1 Spatial discretization and notation

Let $\Omega \subset \mathbb{R}^d$ be a polyhedral domain. This assumption can be generalized by means of boundary finite elements with curved faces, see e.g. Braess [Bra07]. Let $\{\mathcal{T}_h\}_{h>0}$ be a triangulation of Ω into disjoint open simplices with a maximal element size $h := \max_{T\in\mathcal{T}_h}\{diam(T)\}$. We denote the set of nodes of $\mathcal{T}_h$ by J_h. Let $p_j \in J_h$ be their coordinates. The finite element space of piecewise affine linear, continuous finite elements associated to $\mathcal{T}_h$ is now given as $S_h := \{\varphi \in C^0(\overline{\Omega}) \mid \varphi_{|_T} \in P_1(T) \quad \forall\, T \in \mathcal{T}_h\} \subset H^1(\Omega)$ where we denote by $P_1(T)$ the set of all affine linear functions on T. To each $p_j \in J_h$ we associate the nodal basis function $\chi_j \in S_h$ with the property $\chi_j(p_i) = \delta_{ij}$. For further details on the construction of finite element spaces, see, e.g. Braess [Bra07] or Brenner and Scott [BS08]. We replace the L^2-inner product

$(.,.)$ at some places by a quadrature rule given by the lumped mass inner product $(\eta,\chi)_h = \int_\Omega I_h(\eta\chi)$, where $I_h := C^0(\overline{\Omega}) \to S_h$ is the standard interpolation operator at the nodes. This is equivalent to the use of a quadrature rule to evaluate the integral. Note that this only introduces a further approximation error of roughly the same order as the discretization error, see Thomée [Tho06] for a thorough discussion of the topic. We also use a purely algebraic reformulation of the discretized system to discuss the different numerical solvers. We denote the number of nodes by $N := |J_h|$ and introduce the index set $\boldsymbol{J} := \{1,\dots,N\}$. A finite element function $v \in S_h$ is uniquely given by the algebraic notation $\boldsymbol{v} = (\boldsymbol{v}_i)_{i\in\boldsymbol{J}} \in \mathbb{R}^N$ such that $v = \sum_{i=1}^N \boldsymbol{v}_i\chi_i$, hence $\boldsymbol{v}_i = v(p_i)$. The stiffness and mass matrices are accordingly given by

$$\mathbf{S} := \left((\nabla\chi_j,\nabla\chi_i)\right)_{i,j}\,, \quad \mathbf{M} := \left((\chi_j,\chi_i)_h\right)_{i,j} = diag\left((\chi_i,1)\right)_i\,.$$

At some point when we introduce the algebraic primal-dual active set method later on we need minors of matrices. For a matrix $\mathbf{B}$ and two given index sets $\mathbf{X},\mathbf{Y} \subset \boldsymbol{J}$ we set

$$\mathbf{B}_{\mathbf{XY}} := \left(\mathbf{B}_{i,j}\right)_{i\in\mathbf{X},j\in\mathbf{Y}}\,. \tag{3.1}$$

Occasionally we use $\mathbf{B}_\mathbf{X} := \mathbf{B}_{\mathbf{X},\mathbf{X}}$ as abbreviation. Analogously we define a subvector

$$\boldsymbol{v}_\mathbf{X} = \left(\boldsymbol{v}_i\right)_{i\in\mathbf{X}}\,. \tag{3.2}$$

3.2 Projected block Gauss–Seidel type solver (pBSOR)

For the discretization of the variational inequality (2.10)-(2.11) by a semi-implicit Euler step and piecewise linear finite elements in space, we introduce the space

$$K_h := \{\eta \in S_h \mid |\eta(x)| \le 1 \quad \text{for all} \quad x \in \Omega\} \tag{3.3}$$

similar to Blowey and Elliott [BE92]. This results in the following discrete inequality problem.
For $n = 1,2,3,\dots$ and given $u_h^0 \in K_h$ find $(u_h^n, w_h^n) \in K_h \times S_h$ such that

$$\tfrac{1}{\tau}(u_h^n - u_h^{n-1},\chi)_h + (\nabla w_h^n,\nabla\chi) = 0 \qquad \forall\chi\in S_h, \tag{3.4}$$

$$\varepsilon\gamma(\nabla u_h^n,\nabla(\xi-u_h^n)) - (w_h^n,\xi-u_h^n)_h + \tfrac{\gamma}{\varepsilon}(\Psi_0'(u_h^{n-1}),\xi-u_h^n)_h \ge 0 \qquad \forall\xi\in K_h. \tag{3.5}$$

There is a close connection between variational inequality problems and linear or nonlinear complementarity problems. In Karamardian [Kar71] equivalences for those types of problems are shown. Starting out from these equivalent formulations a lot of numerical solution schemes have been developed. We apply in a similar way to [BNS04] a projected SOR type method, where also a convergence proof is given. The development of iterative methods related to the linear complementarity problem started out with the works of Hildreth [Hil54a] and [Hil54b], who introduced a point Gauss–Seidel iterative scheme together with a suitable projection. The more general scheme attributed to Christopherson has been analyzed and clarified by Cryer [Cry71a] and [Cry71b]. The development leading up to a block partitioned variant of the SOR algorithm, see Cottle, Golub and Sacher [CGS77] and citations within, is comprehensively described by Cottle [Cot79]. Another detailed overview on the topic is given by Harker and Pang [HP90] and the books of Facchinei and Pang [FP03a, FP03b].

In the remainder of this section we will reformulate the above problem in a purely algebraic variational inequality problem and state a projected 2×2-block SOR method. We use a so called hybrid method, where the inner very small problems are solved directly.

Let $\mathbf{K} := \left\{\boldsymbol{v} \in \mathbb{R}^N \mid |\boldsymbol{v}_i| \leq 1 \text{ for all } i \in \boldsymbol{J}\right\}$ be the algebraic equivalent of the set K_h given in (3.3). Reformulating the above variational inequality problem by means of the earlier introduced notation we obtain the following algebraic variational inequality problem.

For $n = 1, 2, 3, \ldots$ and given $\boldsymbol{u}^0 \in \mathbf{K}$ find $(\boldsymbol{u}, \boldsymbol{w}) \in \mathbf{K} \times \mathbb{R}^N$ such that

$$\mathbf{M}\boldsymbol{u} + \tau\mathbf{S}\boldsymbol{w} - \mathbf{M}\boldsymbol{u}^{n-1} = 0, \tag{3.6}$$

$$(\boldsymbol{v} - \boldsymbol{u})^t \left(\varepsilon\gamma\mathbf{S}\boldsymbol{u} - \mathbf{M}\boldsymbol{w} - \tfrac{\gamma}{\varepsilon}\mathbf{M}\boldsymbol{u}^{n-1}\right) \geq 0 \qquad \forall \boldsymbol{v} \in \mathbf{K}. \tag{3.7}$$

The next lemma uses a suitable reordering and combines the equality and inequality into one larger block structured variational inequality.

Lemma 3.1. *Let* $\mathbf{Z} := \left\{\boldsymbol{v} \in \left(\mathbb{R}^2\right)^N \mid |(\boldsymbol{v}_i)_1| \leq 1 \quad \forall i \in \{1, \ldots, N\}\right\}$. *Together with the definitions*

$$\boldsymbol{z} := \left(\begin{pmatrix} \boldsymbol{u}_i \\ \boldsymbol{w}_i \end{pmatrix}\right)_{i=1}^N,$$

$$\boldsymbol{f} := \left(\begin{pmatrix} \frac{\gamma}{\varepsilon}\mathbf{M}_{ii}\boldsymbol{u}_i^{n-1} \\ \mathbf{M}_{ii}\boldsymbol{u}_i^{n-1} \end{pmatrix}\right)_{i=1}^N \quad \textit{and}$$

$$\underline{\mathbf{A}} := \left(\begin{pmatrix} \varepsilon\gamma\mathbf{S}_{ij} & -\mathbf{M}_{ij} \\ \mathbf{M}_{ij} & \tau\mathbf{S}_{ij} \end{pmatrix}\right)_{i,j=1}^N$$

the system (3.6)-(3.7) *is equivalent to the variational inequality*

$$(\boldsymbol{v} - \boldsymbol{z})^t \left(\underline{\mathbf{A}}\boldsymbol{z} - \boldsymbol{f}\right) \geq 0 \quad \forall \boldsymbol{v} \in \mathbf{Z}. \tag{3.8}$$

Proof. Firstly we observe that (3.6) can equivalently be formulated as a variational inequality of the form

$$(\boldsymbol{y} - \boldsymbol{w})^t \left(\mathbf{M}\boldsymbol{u} + \tau\mathbf{S}\boldsymbol{w} - \mathbf{M}\boldsymbol{u}^{n-1}\right) \geq 0 \quad \forall \boldsymbol{y} \in \mathbb{R}^N. \tag{3.9}$$

If (3.6) is fulfilled, then the inequality (3.9) is obviously satisfied for all $\boldsymbol{y} \in \mathbb{R}^N$. Using $\boldsymbol{y} := \boldsymbol{w} + \delta\boldsymbol{e}_i$, where $\boldsymbol{e}_i$ denotes the i-th euclidean unit vector in $\mathbb{R}^N$ for arbitrary $\delta \in \mathbb{R}$ yields the equivalence.

We now combine (3.7) and (3.9) into one larger inequality system

$$\left(\begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{y} \end{pmatrix} - \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{pmatrix}\right)^t \left(\begin{pmatrix} \varepsilon\gamma\mathbf{S} & -\mathbf{M} \\ \mathbf{M} & \tau\mathbf{S} \end{pmatrix}\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{pmatrix} - \begin{pmatrix} \frac{\gamma}{\varepsilon}\mathbf{M}\boldsymbol{u}^{n-1} \\ \mathbf{M}\boldsymbol{u}^{n-1} \end{pmatrix}\right) \geq 0, \tag{3.10}$$

which holds for all test functions $\begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{y} \end{pmatrix} \in \mathbf{K} \times \mathbb{R}^N$.

We now introduce the permutation $\sigma := (1, N+1, 2, N+2, \ldots, N, 2N)$ with the associated orthogonal permutation matrix $\mathbf{P}$. This permutation applied from the left side to a system reorders in such a way that the rows are sorted in the same order as prescribed by σ. Similarly an application from the right side resuts in a reordering of the columns. Using this it is easy to see that (3.10) is equivalent to

$$\Big(\underbrace{\mathbf{P}\begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{y} \end{pmatrix}}_{\in\mathbf{Z}} - \underbrace{\mathbf{P}\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{pmatrix}}_{\boldsymbol{z}}\Big)^t \Big(\underbrace{\mathbf{P}\begin{pmatrix} \varepsilon\gamma\mathbf{S} & -\mathbf{M} \\ \mathbf{M} & \tau\mathbf{S} \end{pmatrix}\mathbf{P}^t}_{\underline{\mathbf{A}}} \underbrace{\mathbf{P}\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{pmatrix}}_{\boldsymbol{z}} - \underbrace{\mathbf{P}\begin{pmatrix} \frac{\gamma}{\varepsilon}\mathbf{M}\boldsymbol{u}^{n-1} \\ \mathbf{M}\boldsymbol{u}^{n-1} \end{pmatrix}}_{\boldsymbol{f}}\Big) \geq 0.$$

Together with the above notation and the fact that $\mathbf{M}$ is diagonal due to the mass lumping this is exactly (3.8). □

Note that the mass lumping, i.e. $\mathbf{M}_{ij} = 0$ if $i \neq j$, also implies the symmetry of the block matrix $\underline{\mathbf{A}}$ with respect to the blocks.

Let $Proj_{[-1,1]}$ denote the projection onto the interval $[-1, 1]$.

When we apply the projected block SOR method by Cea and Glowinski [CG73] as stated in Cottle, Golub and Sacher [CGS77], we get the following algorithm:

Algorithm 3.1 *General projected block SOR* (GPBSOR)

1. Set $k = 1$ and initialize $\boldsymbol{z}^{(0)}$.

2. Solve the unrestricted 2×2-block problem for each vertex $i = 1, \ldots, N$:

$$\underline{\mathbf{A}}_{ii}\tilde{\boldsymbol{z}}_i^{(k)} = \boldsymbol{f}_i - \sum_{j=1}^{i-1} \underline{\mathbf{A}}_{ij}\boldsymbol{z}_j^{(k)} - \sum_{j=i+1}^{N} \underline{\mathbf{A}}_{ij}\boldsymbol{z}_j^{(k-1)}. \tag{3.11}$$

3. Apply projection onto $\mathbf{Z}$:

$$\boldsymbol{z}_i^{(k)} = \begin{pmatrix} Proj_{[-1,1]} & 0 \\ 0 & Id \end{pmatrix} \tilde{\boldsymbol{z}}_i^{(k)}. \tag{3.12}$$

4. If abort criterion is not fulfilled, then set $k = k + 1$ and continue with step 2.

The update step (3.12) above can be replaced by a relaxed scheme for a fixed $\omega \in (0,2)$ given by

$$\boldsymbol{z}_i^{(k)} = \begin{pmatrix} Proj_{[-1,1]} & 0 \\ 0 & Id \end{pmatrix} \left(\omega \tilde{\boldsymbol{z}}_i^{(k)} + (1-\omega)\boldsymbol{z}_i^{(k-1)}\right). \tag{3.13}$$

Note that the 2×2-problem (3.11) can easily be solved directly. Using the fact that only the diagonal blocks of $\underline{\mathbf{A}}$ feature entries on the off-diagonal due to the mass lumping used, we can rewrite (3.11) in the variables $\boldsymbol{u}$ and $\boldsymbol{w}$ to obtain the following equation:

$$\begin{aligned}\begin{pmatrix} \varepsilon\gamma\mathbf{S}_{ii} & -\mathbf{M}_{ii} \\ \mathbf{M}_{ii} & \tau\mathbf{S}_{ii} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_i^{(k)} \\ \boldsymbol{w}_i^{(k)} \end{pmatrix} = \begin{pmatrix} \frac{\gamma}{\varepsilon}\mathbf{M}_{ii}\boldsymbol{u}_i^{n-1} \\ \mathbf{M}_{ii}\boldsymbol{u}_i^{n-1} \end{pmatrix} &- \sum_{j=1}^{i-1} \begin{pmatrix} \varepsilon\gamma\mathbf{S}_{ij} & 0 \\ 0 & \tau\mathbf{S}_{ij} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_i^{(k)} \\ \boldsymbol{w}_i^{(k)} \end{pmatrix} \\ &- \sum_{j=i+1}^{N} \begin{pmatrix} \varepsilon\gamma\mathbf{S}_{ij} & 0 \\ 0 & \tau\mathbf{S}_{ij} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_i^{(k-1)} \\ \boldsymbol{w}_i^{(k-1)} \end{pmatrix}\end{aligned}$$

Taking this into account and using Algorithm 3.1, we get the final fully practical method below.

Algorithm 3.2 *Projected block SOR* (pBSOR)

1. Set $k = 1$ and initialize $\boldsymbol{u}^{(0)} \in \mathbf{K}$ and $\boldsymbol{w}^{(0)} \in \mathbb{R}^N$.
2. For $i = 1, \ldots, N$ calculate the right hand sides

$$\begin{aligned} \boldsymbol{g}_i &:= \mathbf{M}_{ii}\boldsymbol{u}_i^{n-1} - \tau \sum_{j=1}^{i-1} \mathbf{S}_{ij}\boldsymbol{w}_j^{(k)} - \tau \sum_{j=i+1}^{N} \mathbf{S}_{ij}\boldsymbol{w}_j^{(k-1)}, \\ \boldsymbol{h}_i &:= \tfrac{\gamma}{\varepsilon}\mathbf{M}_{ii}\boldsymbol{u}_i^{n-1} - \varepsilon\gamma \sum_{j=1}^{i-1} \mathbf{S}_{ij}\boldsymbol{u}_j^{(k)} - \varepsilon\gamma \sum_{j=i+1}^{N} \mathbf{S}_{ij}\boldsymbol{u}_j^{(k-1)} \end{aligned}$$

   and obtain the new iterates via

$$\boldsymbol{u}_i^{(k)} = Proj_{[-1,1]}\left(\omega \frac{\mathbf{M}_{ii}\boldsymbol{g}_i + \tau\mathbf{S}_{ii}\boldsymbol{h}_i}{\varepsilon\gamma\tau\mathbf{S}_{ii}^2 + \mathbf{M}_{ii}^2} + (1-\omega)\boldsymbol{u}_i^{(k-1)}\right) \tag{3.14}$$

   and

$$\boldsymbol{w}_i^{(k)} = \frac{1}{\tau\mathbf{S}_{ii}}\left(\boldsymbol{g}_i - \mathbf{M}_{ii}\boldsymbol{u}_i^{(k)}\right). \tag{3.15}$$

3. Stop if $\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^{(k-1)}\|_2 < tol$, else set $k = k+1$ and goto 2.

For the derivation of (3.14)-(3.15), we first use (3.12) and the abbreviations for the right hand sides giving

$$\begin{aligned}
\varepsilon\gamma \mathbf{S}_{ii}\tilde{\boldsymbol{u}}_i^{(k)} &\quad - \mathbf{M}_{ii}\tilde{\boldsymbol{w}}_i^{(k)} &&= \boldsymbol{h}_i,\\
\mathbf{M}_{ii}\tilde{\boldsymbol{u}}_i^{(k)} &\quad + \tau\mathbf{S}_{ii}\tilde{\boldsymbol{w}}_i^{(k)} &&= \boldsymbol{g}_i.
\end{aligned}$$

The first equation results in

$$\tilde{\boldsymbol{w}}_i^{(k)} = \frac{1}{\mathbf{M}_{ii}}\left(\varepsilon\gamma\mathbf{S}_{ii}\tilde{\boldsymbol{u}}_i^{(k)} - \boldsymbol{h}_i\right).$$

The second equation after replacing w reads as follows:

$$\mathbf{M}_{ii}^2\tilde{\boldsymbol{u}}_i^{(k)} + \tau\mathbf{S}_{ii}(\varepsilon\gamma\mathbf{S}_{ii}\tilde{\boldsymbol{u}}_i^{(k)} - \boldsymbol{h}_i) = \mathbf{M}_{ii}\boldsymbol{g}_i.$$

Hence we obtain:

$$\tilde{\boldsymbol{u}}_i^{(k)} = \frac{\mathbf{M}_{ii}\boldsymbol{g}_i + \tau\mathbf{S}_{ii}\boldsymbol{h}_i}{\varepsilon\gamma\tau\mathbf{S}_{ii}^2 + \mathbf{M}_{ii}^2}.$$

Finally using the relaxed version of Algorithm 3.1, i.e. we replace (3.12) with (3.13) we get the above Algorithm 3.2.

3.3 Discrete primal-dual active set method (PDAS)

As described in Section 2.2 we consider an implicit as well as an explicit discretization of the term $\Psi_0'(u)$, i.e. we choose $\Psi_0'(u^*)$, where $* \in \{n-1, n\}$. Then, the spatial discretization of (2.36), (2.31)-(2.34) is given as follows:
For $n = 1, 2, 3, \ldots$ and given $u_h^0 \in S_h$ find iteratively $(u_h^n, w_h^n, \mu_h^n) \in S_h \times S_h \times S_h$ such that

$$\frac{1}{\tau}(u_h^n - u_h^{n-1}, \chi)_h + (\nabla w_h^n, \nabla\chi) = 0 \qquad \forall\, \chi \in S_h, \tag{3.16}$$

$$(w_h^n, \chi)_h - \varepsilon\gamma(\nabla u_h^n, \nabla\chi) - \frac{\gamma}{\varepsilon}(\psi_0'(u_h^*), \chi)_h - \frac{\gamma}{\varepsilon}(\mu_h^n, \chi)_h = 0 \qquad \forall\, \chi \in S_h, \tag{3.17}$$

$$\mu_h^n = \mu_{h,+}^n - \mu_{h,-}^n\,, \quad \mu_{h,+}^n \geq 0\,, \quad \mu_{h,-}^n \geq 0\,, \quad |u_h^n| \leq 1, \tag{3.18}$$

$$\mu_{h,+}^n(p_j)(u_h^n(p_j) - 1) \;=\; \mu_{h,-}^n(p_j)(u_h^n(p_j) + 1) = 0 \qquad \forall\, p_j \in J_h. \tag{3.19}$$

Note that (3.19) does in general not imply (2.33) pointwise in all of Ω.
Choosing $\chi \equiv 1$ in (3.16) provides the mass conservation $⨍_\Omega u_h^n = ⨍_\Omega u_h^{n-1} = ⨍_\Omega u_h^0$.
The discretization of (2.32)-(2.34) can also be expressed in terms of sets as we did before in Section 2.3 with the help of active nodes

$$\begin{aligned}
\mathcal{A}_h^{n,+} &= \left\{p_j \in J_h \mid u_h^n(p_j) + \frac{\mu_h^n(p_j)}{c} > \; 1\right\},\\
\mathcal{A}_h^{n,-} &= \left\{p_j \in J_h \mid u_h^n(p_j) + \frac{\mu_h^n(p_j)}{c} < -1\right\}
\end{aligned}$$

for any positive c. Then we define the set of inactive nodes as $\mathcal{I}_h^n = J_h \setminus (\mathcal{A}_h^{n,+} \cup \mathcal{A}_h^{n,-})$ and require

$$\begin{aligned} u_h^n(p_j) &= \pm 1 \qquad \text{if } p_j \in \mathcal{A}_h^{n,\pm}\,, \\ \mu_h^n(p_j) &= 0 \qquad \text{if } p_j \in \mathcal{I}_h^n. \end{aligned} \tag{3.20}$$

In order to compute (u_h^n, w_h^n, μ_h^n) we now choose a discretized version of the primal-dual active set method (PDAS-I), where we iteratively update active sets $\mathcal{A}_{h,k}^{n,\pm}$ for $k = 0, 1, 2, \ldots$. We drop for convenience sometimes the indices n, h. The following discrete version of the primal-dual active set strategy is obtained by using that $\mu_h^n(p_j) = 0$ on $\mathcal{I}_{h,k}^n$ in (3.17). Then (3.17) reduces roughly spoken to a discretized PDE for u_h^n only on an interface given by $\mathcal{I}_{h,k}^n$. For known u_h^n, w_h^n (3.17) determines μ_h^n on the active set. Here one has to use that $(\cdot,\cdot)_h$ is a mass lumped L^2-inner product in order to obtain that in (3.17) unknowns at different nodes only couple through the gradient term, leading to a system split according to active and inactive nodes. For the precise formulation we introduce the notation

$$\tilde{S}_{h,k} := \{\tilde{\chi} \in S_h \,|\, \tilde{\chi}(p_j) = 0 \text{ if } p_j \in \mathcal{A}_{h,k}^{n,+} \cup \mathcal{A}_{h,k}^{n,-}\}.$$

Algorithm 3.3 *Discrete primal-dual active set algorithm* (PDAS-II)

1. Set $k = 0$, initialize $\mathcal{A}_0^{\pm}$ and define $\mathcal{I}_0 = J_h \setminus (\mathcal{A}_0^+ \cup \mathcal{A}_0^-)$.
2. Solve for $(u_k, w_k) \in S_h \times S_h$ the system

$$\begin{aligned} \frac{1}{\tau}(u_k - u_h^{n-1}, \chi)_h + (\nabla w_k, \nabla \chi) &= 0 \qquad \forall\, \chi \in S_h\,, & (3.21)\\ (w_k, \tilde{\chi})_h - \varepsilon\gamma(\nabla u_k, \nabla\tilde{\chi}) - \frac{\gamma}{\varepsilon}(\psi_0'(u_h^*), \tilde{\chi})_h &= 0 \qquad \forall\, \tilde{\chi} \in \tilde{S}_{h,k}\,, & (3.22)\\ u_k(p_j) &= \pm 1 \quad \text{if } p_j \in \mathcal{A}_k^{\pm}\,. & (3.23) \end{aligned}$$

3. Define $\mu_k \in S_h$ via

$$\begin{aligned} \mu_k(p_j)\,(1, \chi_j)_h &= \frac{\varepsilon}{\gamma}(w_k, \chi_j)_h - \varepsilon^2(\nabla u_k, \nabla \chi_j) - (\psi_0'(u_h^*), \chi_j)_h \quad \forall\, p_j \notin \mathcal{I}_k, & (3.24)\\ \mu_k(p_j) &= 0 \qquad \forall\, p_j \in \mathcal{I}_k. & (3.25) \end{aligned}$$

4. Set $\mathcal{A}_{k+1}^+ := \{p_j \in J_h \mid u_k(p_j) + \frac{\mu_k(p_j)}{c} > 1\}$,
   $\mathcal{A}_{k+1}^- := \{p_j \in J_h \mid u_k(p_j) + \frac{\mu_k(p_j)}{c} < -1\}$ and
   $\mathcal{I}_{k+1} := J_h \setminus (\mathcal{A}_{k+1}^+ \cup \mathcal{A}_{k+1}^-)$.
5. If $\mathcal{A}_{k+1}^{\pm} = \mathcal{A}_k^{\pm}$ stop, otherwise set $k = k + 1$ and goto 2.

Recalling the notation given in Section 3.1 and introducing an algebraic notation of the active and inactive sets given as in step 4 of Algorithm 3.3, we use

$$\begin{aligned}
\boldsymbol{A}_k^{n,+} &:= \left\{ i \in \boldsymbol{J} \mid p_i \in \mathcal{A}_{h,k}^{n,+} \right\}, \\
\boldsymbol{A}_k^{n,-} &:= \left\{ i \in \boldsymbol{J} \mid p_i \in \mathcal{A}_{h,k}^{n,-} \right\} \text{ and} \\
\boldsymbol{I}_k^{n} &:= \left\{ i \in \boldsymbol{J} \mid p_i \in \mathcal{I}_{h,k}^{n} \right\} = \boldsymbol{J} \setminus \left(\boldsymbol{A}_k^{n,+} \cup \boldsymbol{A}_k^{n,-} \right).
\end{aligned}$$

At various places the distinction between positive active set and negative active set is not necessary and the abbreviation $\boldsymbol{A}_k^n := \boldsymbol{A}_k^{n,+} \cup \boldsymbol{A}_k^{n,-}$ will be used. Additionally we assume that the vertices are ordered in such a way that the first indices belong to the inactive set. This just simplifies the notation and is no restriction. The ordering could easily be achieved by a simple renumbering or permutation of rows and columns.

Applying these algebraic notations to (PDAS-II) and using the submatrix and subvector notation (3.1) and (3.2), we obtain the algebraic formulation (APDAS) of the primal-dual active set algorithm formulated in the semi-implicit setting, i.e. $u_h^* = u_h^{n-1}$.

Algorithm 3.4 *Algebraic primal-dual active set algorithm* (APDAS)

1. Set $k = 0$, initialize $\boldsymbol{A}_0^{\pm}$, define $\boldsymbol{I}_0 = \boldsymbol{J} \setminus (\boldsymbol{A}_0^{+} \cup \boldsymbol{A}_0^{-})$.

2. Set $\boldsymbol{u}_{\boldsymbol{A}_k^{\pm}} = \pm 1$.

3. Solve for $(\boldsymbol{u}_{\boldsymbol{I}_k}, \boldsymbol{w}_k)$ the system

$$\begin{pmatrix} \tau \mathbf{S} & \begin{matrix} \mathbf{M}_{\boldsymbol{I}_k} \\ 0 \end{matrix} \\ \begin{matrix} \mathbf{M}_{\boldsymbol{I}_k} & 0 \end{matrix} & -\varepsilon\gamma \mathbf{S}_{\boldsymbol{I}_k\boldsymbol{I}_k} \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_k \\ \boldsymbol{u}_{\boldsymbol{I}_k} \end{pmatrix} = \begin{pmatrix} \mathbf{M}\boldsymbol{u}^{n-1} - \mathbf{M}_{\boldsymbol{J}\boldsymbol{A}_k}\boldsymbol{u}_{\boldsymbol{A}_k} \\ -\frac{\gamma}{\varepsilon}\mathbf{M}_{\boldsymbol{I}_k}\boldsymbol{u}_{\boldsymbol{I}_k}^{n-1} + \varepsilon\gamma \mathbf{S}_{\boldsymbol{I}_k\boldsymbol{A}_k}\boldsymbol{u}_{\boldsymbol{A}_k} \end{pmatrix}. \tag{3.26}$$

4. Define $\boldsymbol{\mu}_{\boldsymbol{I}_k}, \boldsymbol{\mu}_{\boldsymbol{A}_k}$ via

$$\boldsymbol{\mu}_{\boldsymbol{A}_k} = \frac{\varepsilon}{\gamma}\boldsymbol{w}_{\boldsymbol{A}_k} - \varepsilon^2 \mathbf{M}_{\boldsymbol{A}_k}^{-1} \left(\mathbf{S}_{\boldsymbol{A}_k\boldsymbol{I}_k}\boldsymbol{u}_{\boldsymbol{I}_k} + \mathbf{S}_{\boldsymbol{A}_k}\boldsymbol{u}_{\boldsymbol{A}_k} \right) + \boldsymbol{u}_{\boldsymbol{A}_k}^{n-1}, \tag{3.27}$$

$$\boldsymbol{\mu}_{\boldsymbol{I}_k} = 0. \tag{3.28}$$

5. Set $\boldsymbol{A}_{k+1}^{+} := \{ i \in \boldsymbol{J} \mid \boldsymbol{u}_{ki} + \frac{\boldsymbol{\mu}_{ki}}{c} > 1 \}$,
   $\boldsymbol{A}_{k+1}^{-} := \{ i \in \boldsymbol{J} \mid \boldsymbol{u}_{ki} + \frac{\boldsymbol{\mu}_{ki}}{c} < -1 \}$ and
   $\boldsymbol{I}_{k+1} := \boldsymbol{J} \setminus (\boldsymbol{A}_{k+1}^{+} \cup \boldsymbol{A}_{k+1}^{-})$.

6. If $\boldsymbol{A}_{k+1}^{\pm} = \boldsymbol{A}_k^{\pm}$ stop, otherwise set $k = k+1$ and goto 2.

This algorithm is used in each time step of the simulation. Note that most steps here are very simple value assignments with the exception of 3., where the solution of (3.26) has to be calculated. We comment on different methods to solve this system of equations in Chapter 4. In Chapter 5 we use the saddle point structure of the system and use a Schur complement decomposition to derive a reduced system, which is then solved by a preconditioned conjugate gradient method. Also we present results concerning the existence and uniqueness of solutions in Section 3.7.

3.4 Primal-dual active set method for non-constant mobility (mPDAS)

Earlier we discussed the extension of the classical Cahn-Hilliard problem by means of a non-constant diffusional mobility, compare Section 2.5. In the following we derive a fully discrete scheme of those problems with non-constant mobility. We restrict ourselves to the case with explicitly discretized mobility due to the fact that the implicit case leads to a non-linear method. Again as for the discretization of the problem with constant mobility we assume $\Omega \subset \mathbb{R}^d$ be a polyhedral domain and piecewise linear finite elements S_h on the triangulation $\{\mathcal{T}_h\}_{h>0}$ as in Section 3.1. To deal with the additional restriction caused by a degenerate mobility we take a subset of the vertices p_j, $j \in J_h$, where changes are possible, and define

$$\mathcal{M}_h^n := \left\{p_j \in J_h \mid \exists T \in \mathcal{T}_h \text{ such that } p_j \in \overline{T}, b(u^n)_{|T} \not\equiv 0\right\} \tag{3.29}$$

as well as the associated subset of the finite element functions

$$V_h^n := \{\chi \in S_h \mid \chi(p_j) = 0 \text{ for all } p_j \in J_h \setminus \mathcal{M}_h^n\}\,. \tag{3.30}$$

Note that the remaining nodes in $J_h \setminus \mathcal{M}_h^n$ remain passive, i.e. no changes in the concentration u_h^n occur there. Extending the fully discrete version of the primal-dual active set algorithm for constant mobility, see Algorithm 3.3, we first replace the definitions of the active sets. Adding the dependency on the mobile set $\mathcal{M}_h^{n-1}$, we define

$$\begin{aligned}
\mathcal{A}_h^{n,+} &= \left\{p_j \in \mathcal{M}_h^{n-1} \mid u_h^n(p_j) + \frac{\mu_h^n(p_j)}{c} > \ 1\right\},\\
\mathcal{A}_h^{n,-} &= \left\{p_j \in \mathcal{M}_h^{n-1} \mid u_h^n(p_j) + \frac{\mu_h^n(p_j)}{c} < -1\right\}
\end{aligned}$$

for any positive c. Note that the added dependency is given due to the explicit discretization of the diffusional mobility term. Then we define the set of inactive nodes as $\mathcal{I}_h^n = \mathcal{M}_h^{n-1} \setminus (\mathcal{A}_h^{n,+} \cup \mathcal{A}_h^{n,-})$ and require

$$\begin{aligned}
u_h^n(p_j) &= \pm 1 \qquad \text{if } p_j \in \mathcal{A}_h^{n,\pm}\,,\\
\mu_h^n(p_j) &= 0 \qquad\ \ \text{if } p_j \in \mathcal{I}_h^n.
\end{aligned} \tag{3.31}$$

Furthermore the immobile nodes omit the condition

$$u_h^n(p_j) = u_h^{n-1}(p_j) \quad \text{if } p_j \in J_h \setminus \mathcal{M}_h^n. \tag{3.32}$$

Plugging these into the algorithm we get the following new algorithm for one time step with non-constant diffusional mobility.

Algorithm 3.5 *Primal–dual active set algorithm with explicit non-constant mobility* (mPDAS-II)

1. Set $k = 0$, initialize $\mathcal{M}_h^{n-1}$ as well as $\mathcal{A}_0^\pm$ and define $\mathcal{I}_0$.
2. Solve for $(u_k, w_k) \in S_h \times V_h^{n-1}$ the system

$$\begin{aligned} (u_k - u_h^{n-1}, \chi)_h + \frac{\tau}{\varepsilon}(\nabla w_k, b(u^{n-1})\nabla\chi) &= 0 & \forall\, \chi \in S_h\,, \quad (3.33)\\ (w_k, \tilde\chi)_h - \varepsilon\gamma(\nabla u_k, \nabla\tilde\chi) - \frac{\gamma}{\varepsilon}(\psi_0'(u_h^*), \tilde\chi)_h &= \frac{1}{\varepsilon}(f, \tilde\chi)_h & \forall\, \tilde\chi \in \tilde S_{h,k}\,, \quad (3.34)\\ u_k(p_j) &= \pm 1 & \text{if } p_j \in \mathcal{A}_k^\pm\,. \quad (3.35) \end{aligned}$$

3. Define $\mu_k \in V_h^{n-1}$ via

$$\begin{aligned} \mu_k(p_j)\,(1, \chi_j)_h = {} & \frac{\varepsilon}{\gamma}(w_k, \chi_j)_h - \varepsilon^2(\nabla u_k, \nabla\chi_j) \\ & - (\psi_0'(u_h^*), \chi_j)_h - \frac{1}{\gamma}(f, \chi_j)_h \qquad \forall\, p_j \in \mathcal{A}_k^\pm, \quad (3.36)\\ \mu_k(p_j) = {} & 0 \qquad \forall\, p_j \in \mathcal{I}_k. \quad (3.37) \end{aligned}$$

4. Set $\mathcal{A}_{k+1}^+ := \{p_j \in \mathcal{M}_h^{n-1} \mid u_k(p_j) + \frac{\mu_k(p_j)}{c} > 1\}$,
   $\mathcal{A}_{k+1}^- := \{p_j \in \mathcal{M}_h^{n-1} \mid u_k(p_j) + \frac{\mu_k(p_j)}{c} < -1\}$ and
   $\mathcal{I}_{k+1} := \mathcal{M}_h^{n-1} \setminus (\mathcal{A}_{k+1}^+ \cup \mathcal{A}_{k+1}^-)$.
5. If $\mathcal{A}_{k+1}^\pm = \mathcal{A}_k^\pm$ stop, otherwise set $k = k + 1$ and goto 2.

Note that (3.33) can also be split into two parts. If $\chi \in S_h \setminus V_h^{n-1}$ the equation is reduced to $(u_k - u^{n-1}, \chi)_h = 0$, leading to a smaller system. When we rewrite the system consisting of (3.33) and (3.34) in algebraic notation, we have to define a modified stiffness matrix given by

$$\mathbf{S}_b^{n-1} := \big((\nabla\chi_j, b(u^{n-1})\nabla\chi_i)\big)_{i,j}\,, \quad i, j \in \boldsymbol{M}^{n-1}, \tag{3.38}$$

where $\boldsymbol{M}^{n-1} \subset \boldsymbol{J}$ denotes the set of indices corresponding to $\mathcal{M}_h^{n-1}$ similarly to the sets introduced in Section 3.1. For some remarks on the actual implementation of the system assembly process see Section 3.6.6. Again we assume that the vertices are ordered in such a way that the inactive vertices are numbered before the active ones.

Using (3.38) we get

$$\begin{pmatrix} \frac{\tau}{\varepsilon}\mathbf{S}_b^{n-1} & & \mathbf{M}_{\boldsymbol{I}_k} \\ & & 0 \\ \mathbf{M}_{\boldsymbol{I}_k} & 0 & -\varepsilon\gamma\mathbf{S}_{\boldsymbol{I}_k\boldsymbol{I}_k} \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_k \\ \boldsymbol{u}_{\boldsymbol{I}_k} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{\boldsymbol{M}^{n-1}}\boldsymbol{u}^{n-1}_{\boldsymbol{M}^{n-1}} - \mathbf{M}_{\boldsymbol{M}^{n-1}\boldsymbol{A}_k}\boldsymbol{u}_{\boldsymbol{A}_k} \\ \frac{1}{\varepsilon}\mathbf{M}_{\boldsymbol{I}_k}\boldsymbol{f}_{\boldsymbol{I}_k} - \frac{\gamma}{\varepsilon}\mathbf{M}_{\boldsymbol{I}_k}\boldsymbol{u}^{n-1}_{\boldsymbol{I}_k} + \varepsilon\gamma\mathbf{S}_{\boldsymbol{I}_k\boldsymbol{A}_k}\boldsymbol{u}_{\boldsymbol{A}_k} \end{pmatrix}. \tag{3.39}$$

Note that the vector $\boldsymbol{w}_k \in \mathbb{R}^{|\boldsymbol{M}^{n-1}|}$ as well as the the matrices have to be understood in the sense that they only operate on the mobile vertices given by $\boldsymbol{M}^{n-1}$. The corresponding algebraic notation of Algorithm 3.5 could now be formulated analogously to the derivation of Algorithm 3.4.

3.5 Lagrange-Newton method with non-degenerate mobility (LNM)

Finally we present the fully implicit discrete Lagrange-Newton method with non-degenerate diffusional mobility given in Section 2.5.4. Please note that we require Assumption 2.7. Again we discretize with linear finite elements and use the mass lumped inner product in place of the L^2 product, compare Section 3.1. Rewriting the Newton-update formula (2.75), we get the following weak problem by testing with $(\zeta_u, 0, 0)$, $(0, \zeta_w, 0)$ and $(0, 0, \zeta_\mu)$.
Find $(\delta u_k, \delta w_k, \delta\mu_k) \in S_h \times S_h \times S_h$ such that

$$\begin{aligned}
\varepsilon\gamma(\nabla\delta u_k, \nabla\zeta_u) &- \frac{\gamma}{\varepsilon}(\delta u_k, \zeta_u)_h - \frac{\tau}{2\varepsilon}(|\nabla w_k|^2 b''(u_k)\delta u_k, \zeta_u)_h \\
&- (\delta w_k, \zeta_u)_h - \frac{\tau}{\varepsilon}(\nabla w_k, b'(u_k)\zeta_u\nabla\delta w_k) + \frac{\gamma}{\varepsilon}(\delta\mu_k, \zeta_u)_h \\
&= -\varepsilon\gamma(\nabla u_k, \nabla\zeta_u)_h + \frac{\gamma}{\varepsilon}(u_k, \zeta_u)_h + (w_k, \zeta_u)_h \\
&\quad + \frac{\tau}{\varepsilon}(\nabla w_k, b'(u_k)\zeta_u\nabla w_k) - \frac{\gamma}{\varepsilon}(\mu_k, \zeta_u)_h - \frac{1}{\varepsilon}(f, \zeta_u)_h \qquad \forall\zeta_u \in S_h, \\
-\frac{\tau}{\varepsilon}(\nabla w_k, b'(u_k)\delta u_k\nabla\zeta_w) &- (\delta u_k, \zeta_w)_h - \frac{\tau}{\varepsilon}(\nabla\delta w_k, b(u_k)\nabla\zeta_w) \\
&= \frac{\tau}{\varepsilon}(\nabla w_k, b(u_k)\nabla\zeta_w) + (u_k - u^{n-1}, \zeta_w)_h \qquad \forall\zeta_w \in S_h, \\
-c(\chi_{A^+\cup A^-}\delta u_k, \zeta_\mu)_h &+ (\chi_I\delta\mu_k, \zeta_\mu)_h \\
&= (\mu_k - \min(0, \mu_k + c(u_k+1)) - \max(0, \mu_k + c(u_k-1)), \zeta_\mu)_h \quad \forall\zeta_\mu \in S_h.
\end{aligned}$$

Note that the third equation uses the earlier definitions of the active and inactive sets for the formulation of a slanting function. It can again be split and we obtain

$$\begin{aligned}
(\delta u_k, \zeta_\mu)_h &= (\pm 1 - u_k, \zeta_\mu)_h & \forall\zeta_\mu \in S_h \setminus \tilde{S}_{h,k}, \\
(\delta\mu_k, \zeta_\mu)_h &= (-\mu_k, \zeta_\mu)_h & \forall\zeta_\mu \in \tilde{S}_{h,k}.
\end{aligned}$$

Due to the linearity of the equations we can easily replace the update steps, i.e. set $\delta u_k = u_{k+1} - u_k$ and so on, and eliminate most of the right hand side terms by simply rearranging the terms, to obtain a formulation similar to the primal-dual active set methods. We end up with

$$\begin{aligned}
\varepsilon\gamma(\nabla u_{k+1},\nabla\zeta_u) - \frac{\gamma}{\varepsilon}(u_{k+1},\zeta_u)_h - \frac{\tau}{2\varepsilon}(|\nabla w_k|^2 b''(u_k)u_{k+1},\zeta_u)_h&\\
- (w_{k+1},\zeta_u)_h - \frac{\tau}{\varepsilon}(\nabla w_k, b'(u_k)\zeta_u\nabla w_{k+1}) + \frac{\gamma}{\varepsilon}(\mu_{k+1},\zeta_u)_h&\\
= - \frac{\tau}{2\varepsilon}(|\nabla w_k|^2 b''(u_k)u_k,\zeta_u)_h - \frac{1}{\varepsilon}(f,\zeta_u)_h & \qquad \forall\zeta_u\in S_h, \quad (3.40)\\
-\frac{\tau}{\varepsilon}(\nabla w_k, b'(u_k)u_{k+1}\nabla\zeta_w) - (u_{k+1},\zeta_w)_h - \frac{\tau}{\varepsilon}(\nabla w_{k+1}, b(u_k)\nabla\zeta_w)&\\
= - \frac{\tau}{\varepsilon}(\nabla w_k, b'(u_k)u_k\nabla\zeta_w) - (u^{n-1},\zeta_w)_h & \qquad \forall\zeta_w\in S_h, \quad (3.41)
\end{aligned}$$

as well as $u_{k+1}(p_j) = \pm 1$ for all $p_j \in \mathcal{A}^{\pm}_{k+1}$ and $\mu_k(p_j) = 0$ for all $p_j \in \mathcal{I}_k$.

Remark 3.2. *The above discrete problem* (3.40)-(3.41) *has some additional terms in comparison to the standard discretization of the Cahn-Hilliard PDE. More precisely these are the terms depending on b' and b''. However all of them are multiplied by τ and hence we have again a consistent discretization of the partial differential equation.*

The remaining part of this section gives the necessary notations we need for stating the above method in algebraic notation. Extending the already introduced notation from Section 3.1, we define the matrices

$$\begin{aligned}
\mathbf{S}_b^{(k)} &:= \left((\nabla\chi_j, b(u^{(k)})\nabla\chi_i)\right)_{i,j},\\
\mathbf{S}_{b'}^{(k)} &:= \left((\nabla\chi_j, \chi_i b'(u^{(k)})\nabla w_k)\right)_{i,j} \text{ and}\\
\mathbf{M}_{b''}^{(k)} &:= \left((|\nabla w_k|^2 b''(u_k)\chi_j, \chi_i)\right)_{i,j}.
\end{aligned}$$

Using this we rephrase the semi-smooth Newton method in a similar style to the primal-dual active set methods given earlier. We obtain the method stated in Algorithm 3.6 below.

This algorithm again consists of almost only simple assignments and matrix vector multiplications with the exception of step 3., where the solution to (3.42) has to be calculated. The structure of the problem here is similar as before. We can also see that in case of a constant diffusional mobility the standard method is recovered again. The system of equations (3.42) is also symmetric due to the symmetry of the matrices $\mathbf{S}$, $\mathbf{S}_b$, $\mathbf{M}$ and $\mathbf{M}_{b''}$. Thus the same methods as before can be applied to solve this system, see Chapter 4 and 5.

Algorithm 3.6 *Lagrange-Newton method for implicit non-degenerate mobility* (LNM)

1. Set $k = 0$, initialize $\boldsymbol{A}_0^{\pm}$, define $\boldsymbol{I}_0 = \boldsymbol{J} \setminus (\boldsymbol{A}_0^+ \cup \boldsymbol{A}_0^-)$.
2. Set $\boldsymbol{u}_{\boldsymbol{A}_k^{\pm}} = \pm 1$.
3. Solve for $(\boldsymbol{u}_{\boldsymbol{I}_k}, \boldsymbol{w}_k)$ the system

$$\begin{pmatrix} \frac{\tau}{\varepsilon}\mathbf{S}_b & (\frac{\tau}{\varepsilon}\mathbf{S}_{b'}^{(k)} + \mathbf{M})_{\boldsymbol{J}\boldsymbol{I}_k} \\ (\frac{\tau}{\varepsilon}\mathbf{S}_{b'}^{(k)^T} + \mathbf{M})_{\boldsymbol{I}_k\boldsymbol{J}} & -(\varepsilon\gamma\mathbf{S} - \frac{\gamma}{\varepsilon}\mathbf{M} - \frac{\tau}{2\varepsilon}\mathbf{M}_{b''}^{(k)})_{\boldsymbol{I}_k\boldsymbol{I}_k} \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_k \\ \boldsymbol{u}_{\boldsymbol{I}_k} \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{M}\boldsymbol{u}^{n-1} - \mathbf{M}_{\boldsymbol{J}\boldsymbol{A}_k}\boldsymbol{u}_{\boldsymbol{A}_k} + \frac{\tau}{\varepsilon}\mathbf{S}_{b'}^{(k)}\boldsymbol{u}_{k-1,\boldsymbol{I}_k} \\ (\varepsilon\gamma\mathbf{S} - \frac{\gamma}{\varepsilon}\mathbf{M} - \frac{\tau}{2\varepsilon}\mathbf{M}_{b''}^{(k)})_{\boldsymbol{I}_k\boldsymbol{A}_k}\boldsymbol{u}_{\boldsymbol{A}_k} + (\frac{\tau}{2\varepsilon}\mathbf{M}_{b''}^{(k)}\boldsymbol{u}_{k-1})_{\boldsymbol{I}_k} + \frac{1}{\varepsilon}\mathbf{M}_{\boldsymbol{I}_k}\boldsymbol{f}_{\boldsymbol{I}_k} \end{pmatrix}. \tag{3.42}$$

4. Define $\boldsymbol{\mu}_{\boldsymbol{I}_k}, \boldsymbol{\mu}_{\boldsymbol{A}_k}$ via

$$\boldsymbol{\mu}_{\boldsymbol{A}_k} = \mathrm{M}_{\boldsymbol{A}_k}^{-1}\Big((\tfrac{\tau}{\gamma}\mathbf{S}_{b'}^{(k)} + \tfrac{\varepsilon}{\gamma}\mathbf{M})\boldsymbol{w}_k - (\varepsilon^2\mathbf{S} - \mathbf{M} - \tfrac{\tau}{2\gamma}\mathrm{M}_{b''}^{(k)})\boldsymbol{u}_k - \tfrac{\tau}{2\gamma}\mathrm{M}_{b''}^{(k)}\boldsymbol{u}_{k-1} - \tfrac{1}{\gamma}\mathrm{M}\boldsymbol{f}\Big)_{\boldsymbol{A}_k},$$
$$\boldsymbol{\mu}_{\boldsymbol{I}_k} = 0.$$

5. Set $\boldsymbol{A}_{k+1}^+ := \{i \in \boldsymbol{J} \mid \boldsymbol{u}_{ki} + \frac{\boldsymbol{\mu}_{ki}}{c} > 1\}$,
   $\boldsymbol{A}_{k+1}^- := \{i \in \boldsymbol{J} \mid \boldsymbol{u}_{ki} + \frac{\boldsymbol{\mu}_{ki}}{c} < -1\}$ and
   $\boldsymbol{I}_{k+1} := \boldsymbol{J} \setminus (\boldsymbol{A}_{k+1}^+ \cup \boldsymbol{A}_{k+1}^-)$.
6. If $\boldsymbol{A}_{k+1}^{\pm} = \boldsymbol{A}_k^{\pm}$ stop, otherwise set $k = k + 1$ and goto 2.

3.6 Various tasks on the discrete level

There are still some details open before the algorithms introduced here can be implemented. Below we discuss those things and point out restrictions for the selection of the various parameters.

3.6.1 The adaptive grid

For all the simulations presented here the finite element toolbox ALBERTA by Schmidt and Siebert [SS05] was used for mesh generation, the assembly of the matrices and administration. To generate the adaptive meshes we used the mesh adaption strategy of Barrett, Nürnberg, Styles [BNS04]. Experiments showed that it is essential to ensure that at least eight vertices are present across the interfaces to avoid mesh effects like anisotropy, see also Blowey and Elliott [BE93] or [BE94]. We hence refine on the

interface down to a level where eight vertices are present and coarsen in the areas where the concentration u is constant. For given parameter ε this results in an upper bound $h_{fine} \leq \varepsilon\frac{\pi}{9}$, where h_{fine} is the refinement level on the interface. We remark here that for the potential ψ_0 in (2.5) the interfacial thickness is given by $\varepsilon\pi$, compare [BE94]. Since we want to avoid too coarse meshes we additionally define $h_{coarse} := 10 \cdot h_{fine}$ and choose a tolerance $tol = 10^{-6}$. Afterwards the mesh adaption is done in the following way: For each element $T \in \mathcal{T}^h$ calculate the indicator $\eta_T := |\underset{x \in T}{min} |u(x)| - 1|$. Then, a triangle is marked for refinement if it, or one of its adjacent elements, satisfies $\eta_T > tol \cdot 10^{-1}$ and if $h_T > h_{fine}$. A triangle is marked for coarsening if it satisfies $\eta_T < tol \cdot 10^{-3}$ and $h_T < h_{coarse}$. This is repeated until no further refinements or coarsenings are made.
When using adaptive meshes for the primal-dual active set algorithm a choice has to be made for every newly created vertex if it should belong to the active or inactive set. Due to the close link to Newton methods the selection of initial data for the iteration process is a key ingredient for a fast and efficient method. Thus the selection of good initial active and inactive sets is crucial.

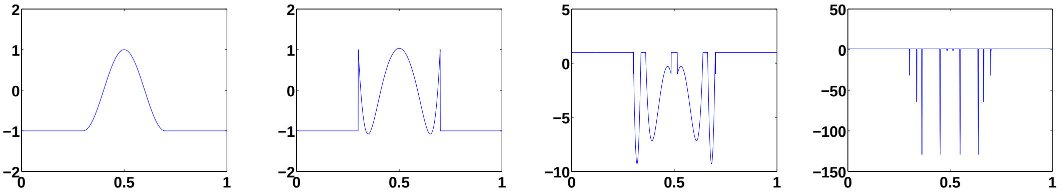
3.6.2 Initialization of the active sets

As mentioned previously the application of a PDAS-method to the interface evolution has the advantage that the good initialization due to the information from the previous time step leads to a large speedup. At the first time step $n = 1$ the active set $\mathcal{A}_0^{n,\pm}$ is initialized using the given initial data u_h^0. Since in the limit the active sets describe the sets where u is strictly active a good approximation of $\mathcal{A}_0^{1,\pm}$ is given by the active set of u_h^0. Hence we choose $\mathcal{A}_0^{1,\pm} = \{p_j \in S^h \mid |u_h^0(p_j) \mp 1| \leq 10^{-8}\}$.
For time steps $n \geq 2$ we can exploit in addition $\mu_{h_{n-1}}^{n-1}$. Due to possible grid changes from time step $n-1$ to time step n one may have to apply additionally the standard interpolation I_{h_n} to the new grid S^{h_n}, i.e. with $u_{-1} := I_{h_n} u_{h_{n-1}}^{n-1}$ and $\mu_{-1} := I_{h_n} \mu_{h_{n-1}}^{n-1}$ initialize the active set $\mathcal{A}_0^{n,\pm}$ as in step 4 of Algorithm 3.3.
However a less time consuming method is to initialize the active set in the following way, which is applied in this work: if an edge between two positive or two negative active vertices is bisected, the new vertex is set active and otherwise the new vertex is set inactive. This is sufficient, since it is only an initial guess for the sets, and does not need additional effort to determine the sets on each node anew.

3.6.3 Choosing of the parameter c

To determine the active sets we have to choose the parameter $c > 0$. In the unilateral case the selection of $c > 0$ has no influence on the iterates after the first iteration and can be chosen arbitrary, see Hintermüller, Ito and Kunisch [HIK02]. However this is no longer true in the case of bilateral bounds. This is also discussed for similar obstacle problems in the paper by Blank, Garcke, Sarbu and Styles, see [BGSS09]. If c is chosen too small we observed cases in which the iterates oscillated and the algorithm did not converge. Figure 3.1 shows the values of u at various PDAS iterations in one time step of a simulation in one space dimension with $h = \frac{1}{512}$, $\tau = 10^{-5}$, $\pi\varepsilon = 0.2$ and $c = 0.01$.

In the eighth iteration the algorithm breaks down because all vertices are in the active set and the system no longer exhibits a valid solution, compare Remark 3.5. Redoing the simulation with $c = 0.2$ fixed the problem and after two iterations the time step was completed with only marginal changes to u since the initial data was close to a stationary solution. The same phenomenon was observed in higher space dimensions. An heuristic approach showed that it is sufficient to ensure that no vertex can change from the positive active set $\mathcal{A}^+$ to the negative active set $\mathcal{A}^-$ and vice versa in one iteration. This can be achieved by selecting a PDAS parameter c large enough, depending on the magnitude of the Lagrange multiplier μ.



u_h^0 u_h^3 u_h^5 u_h^7

Figure 3.1: Oscillations in 1D if c is too small.

Remark 3.3. *Note that our scaling of the Lagrange multiplier μ with $\frac{\gamma}{\varepsilon}$ becomes important at this point. With this scaling we obtain a μ with a magnitude of roughly 1. In some early iterations especially when starting with absolutely random initial data this might not always be the case but there the magnitude of μ depends heavily on the mesh size of the underlying grid. In all situations where the interfaces are well developed μ typically takes values of roughly ± 1.1 independently of γ and ε.*

In all the simulations a value of $c = 10$ was sufficient when the interfaces were already well developed and adequate initial guesses for the active sets were known. Therefore, if not mentioned otherwise $c = 10$ is chosen in the calculation. In simulations with distortions or jumps in the concentration u larger values depending on the mesh size were necessary. Choosing the parameter c larger had no percievable influence on the simulation.

3.6.4 Some remarks on the time step width

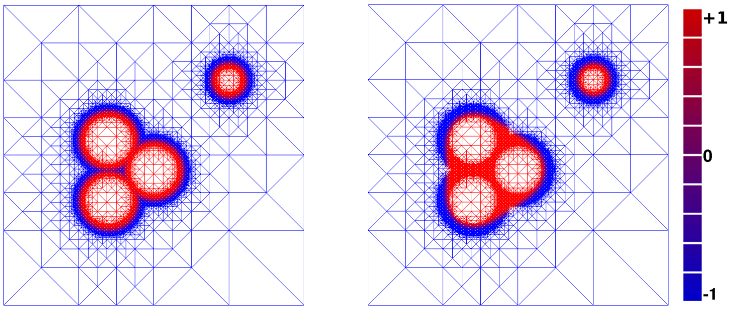
Considering the semi-implicitly time discretized problem, we show in Lemma 3.4 in the following section that the PDAS iteration is well posed for all time step sizes. All simulations on a suitably fine equidistant grid to resolve the interface adequately converged after very few PDAS iterations. Naturally the approximation error gets larger for large time steps and the related sharp interface problem is no longer approximated nicely. This behavior can be seen, e.g. in the simulations comparing the numerical solutions to the sharp interface solutions, see Section 6.1.2.

In Lemma 2.11 and Lemma 2.12 we have proven the existence and uniqueness of a minimizer of the fully implicit time discrete scheme with a restriction on the time step width. Naturally the same effects as above occur on the adaptive grid. However in almost all of our simulations no problems occurred for sensible large time steps even above the upper bound, where we could show well-posedness of the problem. However for comparatively large time steps significantly more primal-dual active set iterations were necessary. To showcase this behavior we use constant mobility, i.e. $\rho \equiv 1$ and set $\gamma = 1$. As initial data we use data similar to the initial data used in Figure 3.2 below. However here we use an equidistant mesh to avoid any influence of the adaptivity on this test. As long as the time step size is below the bound where our proof works, the iteration numbers stay low. Simulations using a time step size larger than that converge somewhat slower and sometimes no convergence is reached for comparatively large time steps, see Table 3.1.

ε	vertices	$\tau_{max} = \frac{4\varepsilon^2}{\gamma}$	PDAS-iterations for various time step sizes τ						
			10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	$5 \cdot 10^{-4}$	10^{-3}
0.04	16641	$6.4 \cdot 10^{-5}$	2	2	3	5	6	14	—
0.02	66049	$1.6 \cdot 10^{-5}$	3	4	5	6	12	—	—
0.01	263169	$4 \cdot 10^{-6}$	4	5	7	7	32	54	53

Table 3.1: Implicit method for different time step sizes.

Another problem arises when the grid used is adaptively generated by the method described in Section 3.6.1. For large time steps this leads to mesh anisotropy effects due to the fact that the interface moves beyond the finely resolved part, further decreasing the accuracy of the time evolution process, compare Figure 3.2, where the solution after a comparatively large time step on an adaptive mesh is shown. Please note that this behavior occurs only when we use the implicit time discretization of the free energy term.

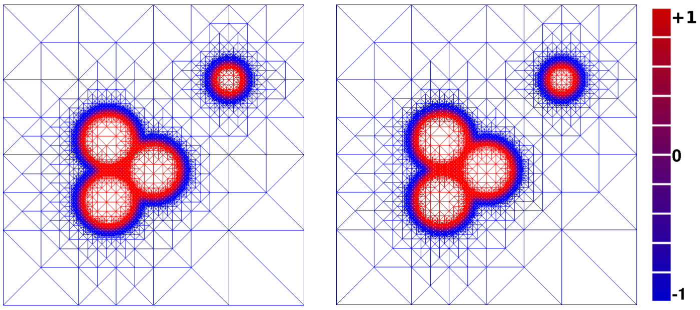


Data at beginning of the time step.

Solution after one time step ($\tau = 10^{-4}$).

Figure 3.2: Solution u of the implicit method after one time step. Here the time step $\tau = 10^{-4}$ leads to errors due to the adaptive mesh and fast moving interface.

The explicit method allows for large time steps, but the evolution is slowed by a pinning effect, which also explains the slowed evolution in the comparative study we did in Section 6.1.2. We repeated the time step shown in Figure 3.2 with the explicitly discretized free energy term and show the results for the same time step size as well as for an extremely large time step size, see Figure 3.3.



Solution after one time step ($\tau = 10^{-4}$).

Solution after one large time step ($\tau = 10^{4}$).

Figure 3.3: Explicit time steps with initial data as in Figure 3.2.

The problem with the generated meshes for the implicit method occurs due to the fact that the explicit algorithm controls the adaptivity of the mesh with information of u^{n-1} at the old time step. To circumvent this problem an implicit method could be used, where the mesh also resolves u^n equally good. Even if we might use larger time steps without the problems arising in the explicit method, this additional iteration leads to a higher computational effort for a single time step. In addition to the effort for adapting the mesh all matrices have to be adjusted or assembled anew. Since the time step size necessary for this phenomenon to appear is quite large, we are content with the explicit method.

3.6.5 Remark on the tolerance for the saddle point system solver

We would like to point out that the tuning of the tolerance used for the solution of the saddle point problem becomes very important with respect to the convergence of the primal-dual active set iteration. If the tolerance allowed for the solution of saddle point problem is too large, convergence for the primal-dual active set iteration is not possible. A closely related topic, namely the control of the tolerances prescribed for an inexact preconditioned conjugate gradient method, is extensively discussed by Golub and co-workers [GY99], where some algorithm for the control of the tolerances of the nested iterations there is given. Some similar problems occur when we use a Schur complement type solver in conjunction with an iterative solver, compare Chapter 5.

The simulations using the direct solver for the solution of the saddle point problem done during the course of this project, exhibited no problems at all, which was of course the expected behavior. The application of an iterative solver on the other hand with a tolerance, which is too large, leads to a diverging primal-dual active set iteration.

Some speedup was gained by the use of a heuristic decreasing of the tolerance for each primal-dual active set iteration, i.e. we started out in the first iteration with a tolerance of 10^{-5} for the first iteration and decreased it with each iteration by the factor of 0.1 down to 10^{-10}.

3.6.6 System matrix assembly with non-constant mobility

The assembly of the stiffness matrix with non-constant mobility can reuse values calculated for the assembly of the standard stiffness matrix. We show this by testing (3.33) with χ_j and the following short calculation

$$\begin{aligned}
(\nabla w_k, b(u^{n-1})\nabla\chi_i) &= \sum_{j\in\mathcal{M}_h^{n-1}} \boldsymbol{w}_{k,j}(\nabla\chi_j, b(u^{n-1})\nabla\chi_i) \\
&= \sum_{j\in\mathcal{M}_h^{n-1}} \boldsymbol{w}_{k,j} \sum_{T\in\mathcal{T}_h} \int_T \nabla\chi_j \cdot b(u^{n-1})\nabla\chi_i \\
&= \sum_{j\in\mathcal{M}_h^{n-1}} \boldsymbol{w}_{k,j} \sum_{T\in\mathcal{T}_h} \frac{1}{|T|} \int_T \nabla\chi_j \cdot \nabla\chi_i \int_T b(u^{n-1}),
\end{aligned}$$

where we used the fact that $\nabla\chi_j \cdot \nabla\chi_i$ is constant for piecewise linear nodal basis functions χ_i, $\chi_j \in S_h$ on each element. We don't need to evaluate the integral again, because the scalar product of the basis functions gradients have already been calculated for the assembly of the discrete Laplacian with constant mobility necessary for the u term. We denote the local element stiffness matrix by $\mathbf{s}_T$, which is given by

$$\mathbf{s}_T = \left(\int_T \nabla\chi_j \cdot \nabla\chi_i \right)_{i,j} \quad \text{for all } i,j \text{ such that } p_i, p_j \in \overline{T}.$$

The indices i, j have to be interpreted correctly. To be formally correct we would require the introduction of a mapping from local indices to global indices, which we omit for a shorter presentation. For more details on the assembly of the matrices by element we refer to standard finite element literature like Braess [Bra07] or Brenner and Scott [BS08]. Finally we obtain

$$\left(\mathbf{S}_b^{n-1}\right)_{i,j} = \sum_{T\in\mathcal{T}_h} (\mathbf{s}_T)_{i,j} \frac{1}{|T|} \int_T b(u^{n-1})$$

for the generation of the stiffness matrix with non-constant mobility.

Thus it is sufficient to evaluate the integral over b on each element for the creation of the modified system matrix. If the value of the integral is zero, then nothing is added. Additionally we gain the information on the mobile set $\mathcal{M}_h^{n-1}$, i.e. the set where the mobility function b is positive, simply by storing the information, which vertices are updated by the above formula. For the actual calculation of $\int_T b(u)$ we use a quadrature formula, which is exact for polynomials of degree two. Since u is linear on each simplex T this is sufficient for the chosen mobility $b(u) = (1 - u^2)$.

The implicitly discretized mobility lead to the Lagrange-Newton method. Analogously to before, we get

$$(\mathbf{S}_b^{(k)})_{i,j} = \sum_{T\in\mathcal{T}_h} (\mathbf{s}_T)_{i,j} \frac{1}{|T|} \int_T b(u^{(k)}).$$

The calculation of the entries of $\mathbf{M}_{b''}^{(k)}$ can be done in a very similar way. The fact that ∇w_k is constant on each element and a simple reformulation gives

$$\begin{aligned}(|\nabla w_k|^2 b''(u_k)\chi_j, \chi_i) &= \sum_{T\in\mathcal{T}_h} \int_T |\nabla w_k|^2 b''(u_k)\chi_j\chi_i \\ &= \sum_{T\in\mathcal{T}_h} |\nabla w_{k|_T}|^2 \int_T b''(u_k)\chi_j\chi_i.\end{aligned}$$

Additionally the factor on each element given by the ∇w term can easily be calculated with the help of the local stiffness matrix. Using local numbering again and $\nabla w_k = \sum_i \nabla \boldsymbol{w}_{k,i}\nabla\chi_i$, we get

$$\begin{aligned}|\nabla w_{k|_T}|^2 =& \boldsymbol{w}_{k,j}\nabla\left(\sum_i \boldsymbol{w}_{k,i}\chi_i\right)\cdot\nabla\left(\sum_j \boldsymbol{w}_{k,j}\chi_j\right) \\ =& \sum_{i,j} \boldsymbol{w}_{k,i}\boldsymbol{w}_{k,j}\nabla\chi_i\cdot\nabla\chi_j = \frac{1}{|T|}\boldsymbol{w}_{k|_T}^T(\mathbf{s}_T)\boldsymbol{w}_{k|_T}.\end{aligned}$$

Thus for the calculation we can also use the already calculated values of the local element stiffness matrix. The remaining integral is again evaluated by a quadrature formula. Depending on b'' the integral is either evaluated only approximately or the number of quadrature points and thus the degree of integrands the formula is exact for has to be adjusted. If b'' is constant the latter can also be evaluated by means of the element mass matrix.

The remaining matrix $\mathbf{S}_{b'}^{(k)}$ is also assembled by element, again we reduce the formula by using precalculated parts. Replacing the finite element function by its representation in basis functions again, we obtain

$$\begin{aligned}(\nabla\chi_j, \chi_i b'(u^{(k)})\nabla w_k) &= \sum_{T\in\mathcal{T}_h} \frac{1}{|T|}\int_T \nabla\chi_j\cdot\nabla\left(\sum_l \boldsymbol{w}_{k,l}\chi_l\right)\int_T b'(u^{(k)})\chi_i \\ &= \sum_{T\in\mathcal{T}_h} \frac{1}{|T|}\int_T (\mathbf{s}_T\boldsymbol{w}_{k|_T})_j \int_T b'(u^{(k)})\chi_i.\end{aligned}$$

Analogously to before we calculate $\int_T b'(u^{(k)})\chi_i$ by means of a quadrature formula.

3.7 Existence and uniqueness of solutions

Ahead of discussing practical solution methods of the discrete saddle point system arising in (mPDAS-II), we present existence and uniqueness results for the corresponding system of equations (3.33)-(3.35). The diffusional mobility given by $\rho = \frac{1}{\varepsilon}b(u^{n-1})$ is required to fulfill either Assumption 2.7 or Assumption 2.8. The results below are formulated for the primal-dual active set method with degenerate explicit non-constant diffusional mobility. Note that the non-degenerate case is included, due to the fact that $V_h^{n-1} = S_h$ and $\mathcal{M}_h^{n-1} = J_h$ then holds. Again as before those results are also true for (PDAS-II) by setting the diffisional mobility to $\rho \equiv 1$, i.e. $b(u) \equiv \varepsilon$, and the force term $f \equiv 0$. Note that the assertion of a non-empty inactive set is crucial.

Lemma 3.4. *For all $u_h^{n-1} \in S_h$ and $\mathcal{A}_k^{\pm}$ there exists a unique solution $(u_k, w_k) \in S_h \times V_h^{n-1}$ of (3.33)-(3.35) with $* = (n-1)$, i.e. the semi-implicit case, provided that $\mathcal{I}_k = \mathcal{M}_h^{n-1} \setminus (\mathcal{A}_k^{+} \cup \mathcal{A}_k^{+}) \neq \emptyset$.*

Proof. The idea of this proof is to consider the discretized version of the minimization problem (2.57) under the constraints (2.58) and $u = \pm 1$ on $\mathcal{A}_k^{\pm}$ and to use ideas similar to the existence proof in Lemma 2.11. Hence, we define $S_{h,m} := \{\chi \in S_h \mid ⨍_\Omega \chi = m\}$, where $m := ⨍_\Omega u_h^{n-1}$,

$$S_h^I := \{u \in S_h \mid u(p_j) = \pm 1 \text{ if } p_j \in \mathcal{A}_k^{\pm} \text{ and } u(p_j) = u^{n-1}(p_j) \text{ if } p_j \in J_h \setminus \mathcal{M}_h^{n-1}\}\,,$$

and $S_{h,m}^I := S_h^I \cap S_{h,m}$. Since $\mathcal{I}_k \neq \emptyset$ we conclude $S_{h,m}^I \neq \emptyset$. Additionally we set $V_{h,m}^{n-1} := \{\chi \in V_h^{n-1} \mid ⨍_{\mathcal{M}_h^{n-1}} \chi = m\}$, where V_h^{n-1} is the set of finite element functions, whose support intersects with the one of the mobility function ρ, see (3.30).
Similar to the continuous weighted Laplacian given in (2.52), the discrete inverse weighted Laplacian $(-\nabla \cdot \rho \nabla)_h^{-1} : V_{h,0}^{n-1} \to V_{h,0}^{n-1}$, $\eta^h \mapsto (-\nabla \cdot \rho\nabla)_h^{-1}\eta^h$ is defined via

$$(\nabla((-\nabla \cdot \rho\nabla)_h^{-1}\eta^h), \rho\nabla\chi) = (\eta^h, \chi)_h \quad \text{for all} \quad \chi \in S_{h,0}\,. \tag{3.43}$$

The linear equation (3.43) has a unique solution, since the homogeneous problem only has the trivial solution and $V_{h,0}^{n-1}$ is finite dimensional. We define $u_k \in S_{h,m}^I$ as the solution of the minimization problem

$$\begin{aligned} \min_{\eta \in S_{h,m}^I} \Big\{ &\frac{1}{2\tau}(\nabla(-\nabla \cdot \rho\nabla)_h^{-1}(\eta - u_h^{n-1}), \rho\nabla(-\nabla \cdot \rho\nabla)_h^{-1}(\eta - u_h^{n-1})) \\ &+ \frac{\gamma\varepsilon}{2}(\nabla\eta, \nabla\eta) + \frac{\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \eta)_h + \frac{1}{\varepsilon}(f, \eta)_h \Big\} \end{aligned} \tag{3.44}$$

which exists uniquely since the Poincaré inequality for functions with mean value zero, similar as in the proof of Lemma 2.11, implies coerciveness. Computing the first variation of the minimization problem (3.44) gives for the solution $u_k \in S_{h,m}^I$

$$\begin{aligned} 0 = &\frac{1}{\tau}(\nabla(-\nabla \cdot \rho\nabla)_h^{-1}(u_k - u_h^{n-1}), \rho\nabla(-\nabla \cdot \rho\nabla)_h^{-1}\tilde{\chi}) + \gamma\varepsilon(\nabla u_k, \nabla\tilde{\chi}) \\ &+ \frac{\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \tilde{\chi})_h + \frac{1}{\varepsilon}(f, \tilde{\chi})_h \end{aligned} \tag{3.45}$$

for all $\tilde{\chi} \in \tilde{S}_{h,k}$ with $⨍_\Omega \tilde{\chi} = 0$. Now we define $w_k \in V_h^{n-1}$ as

$$w_k = -(-\nabla \cdot \rho\nabla)_h^{-1}\left(\frac{u_k - u_h^{n-1}}{\tau}\right) + \lambda_k, \tag{3.46}$$

where $\lambda_k \in \mathbb{R}$ is uniquely given with the help of any nodal basis function $\chi_j \in S_h$ with $p_j \in \mathcal{I}_k$ by

$$\begin{aligned}\lambda_k = \bigg\{ &\frac{1}{\tau}((-\nabla\cdot\rho\nabla)_h^{-1}(u_k - u_h^{n-1}), \chi_j)_h + \gamma\varepsilon(\nabla u_k, \nabla\chi_j) \\ &+ \frac{\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \chi_j)_h - \frac{1}{\varepsilon}(f, \chi_j)_h \bigg\} / (1, \chi_j)_h . \end{aligned} \tag{3.47}$$

Using the definition of the discrete inverse weighted Laplacian, see (3.43), and the fact that $\fint_\Omega u_k = \fint_\Omega u^{n-1}$ now gives that (3.33) holds. Furthermore (3.45), (3.43) and (3.46) imply that (3.34) holds for all $\tilde{\chi} \in \tilde{S}_{h,k}$ with $\fint_\Omega \tilde{\chi} = 0$. For $\tilde{\chi} \in \tilde{S}_{h,k}$ which do not satisfy the integral constraint $\fint_\Omega \tilde{\chi} = 0$ we set $\hat{\chi} := \tilde{\chi} - \alpha\chi_j$ with $p_j \in I_k$ and $\alpha \in \mathbb{R}$ such that $\int_\Omega \hat{\chi} = 0$. We use this choice of $\hat{\chi}$ as a test function in (3.45). We then obtain that (3.34) holds for all $\tilde{\chi} \in \tilde{S}_{h,k}$ due to the following caluculation:

$$\begin{aligned} 0 &\overset{(3.43)}{=} \frac{1}{\tau}((-\nabla\cdot\rho\nabla)_h^{-1}(u_k - u_h^{n-1}), \tilde{\chi} - \alpha\chi_j)_h + \gamma\varepsilon(\nabla u_k, \nabla(\tilde{\chi} - \alpha\chi_j)) \\ &\quad + \frac{\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \tilde{\chi} - \alpha\chi_j)_h + \frac{1}{\varepsilon}(f, \tilde{\chi} - \alpha\chi_j)_h \\ &\overset{(3.47)}{=} \frac{1}{\tau}((-\nabla\cdot\rho\nabla)_h^{-1}(u_k - u_h^{n-1}), \tilde{\chi})_h + \gamma\varepsilon(\nabla u_k, \nabla\tilde{\chi}) + \frac{\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \tilde{\chi})_h \\ &\quad + \frac{1}{\varepsilon}(f, \tilde{\chi})_h - \alpha\lambda_k(1, \chi_j)_h \\ &= \frac{1}{\tau}((-\nabla\cdot\rho\nabla)_h^{-1}(u_k - u_h^{n-1}), \tilde{\chi})_h + \gamma\varepsilon(\nabla u_k, \nabla\tilde{\chi}) + \frac{\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \tilde{\chi})_h \\ &\quad + \frac{1}{\varepsilon}(f, \tilde{\chi})_h - \lambda_k(1, \tilde{\chi})_h \\ &\overset{(3.46)}{=} -(w_k, \tilde{\chi})_h + \gamma\varepsilon(\nabla u_k, \nabla\tilde{\chi}) + \frac{\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \tilde{\chi})_h + \frac{1}{\varepsilon}(f, \tilde{\chi})_h . \end{aligned}$$

Hence (3.33)-(3.35) has a solution.
It remains to prove uniqueness. Let us assume that (3.33)-(3.35) has two solutions $(u_{k,1}, w_{k,1}), (u_{k,2}, w_{k,2}) \in S_h \times V_h^{n-1}$. Then we obtain for the differences $v = u_{k,1} - u_{k,2}$, $z = w_{k,1} - w_{k,2}$ by testing (3.33) with z and (3.34) with v for $(u_{k,1}, w_{k,2})$ and $(u_{k,2}, w_{k,2})$ after taking differences:

$$(v, z)_h + \tau\|\nabla z\|_{L^2_\rho}^2 - (z, v)_h + \gamma\varepsilon\|\nabla v\|_{L^2}^2 = 0\,.$$

Since $\fint_\Omega u_{k,1} = \fint_\Omega u_{k,2} = \fint_\Omega u^{n-1}$ we obtain $v \equiv 0$ in Ω and hence $u_{k,1} = u_{k,2}$. The identities (3.33), (3.34) imply that necessarily the identities (3.46) and (3.47) have to hold. This implies that also w_k is unique. We remark that this uniqueness result also implies that the definition of λ_k in (3.47) does not depend on j. □

Now $\mu_k \in V_h^{n-1}$ is uniquely defined by (3.36), (3.37) and hence taking Lemma 3.4 into account we obtain that a unique solution of the linear equation system used in (mPDAS-II), i.e. (3.33)-(3.37), exists.

Remark 3.5.

1. *We require the condition* $\mathcal{I}_k = \mathcal{M}_h^{n-1} \setminus (\mathcal{A}_k^+ \cup \mathcal{A}_k^-) \neq \emptyset$*, which guarantees that there is a* $u \in S_h$ *such that* $⨍_\Omega u = m$*. Otherwise* (3.33) *is not solvable.*

2. *In order to solve* (3.33)-(3.37) *the main computational effort is to solve the system* (3.33)*,* (3.34) *which has a specific structure. The discretized elliptic equation* (3.33) *for* w *is defined on the whole of* Ω *whereas the elliptic equation* (3.34) *is defined only on the inactive set. The two equations are coupled in a way which leads to an overall symmetric system which will be used later when we propose numerical algorithms.*

The discretization of the complementarity condition (2.67)-(2.69) can also be formulated with the help of the semi-smooth function $H(u, \mu) = \mu - (\max(0, \mu + c(u-1)) + \min(0, \mu + c(u+1))$, see Remark 2.4, as a nonlinear equation

$$H(u_h^n(p_j), \mu_h^n(p_j)) = 0 \qquad \forall\ p_j \in \mathcal{M}_h^{n-1}\,. \tag{3.48}$$

Using the approach of Hintermüller, Ito and Kunisch, see [HIK02], we can interpret (mPDAS-II) as a semi-smooth Newton method for the system (3.33), (3.34), (3.48) and we obtain the following local convergence result for the semi-implicit discretization.

Theorem 3.6. *Let* $(u, w, \mu) \in S_h \times V_h^{n-1} \times V_h^{n-1}$ *be a solution of* (3.33)*,* (3.34)*,* (3.48) *with* $* = (n-1)$ *such that* $\{p_j \in \mathcal{M}_h^{n-1} \mid |u(p_j)| < 1\} \neq \emptyset$*. Then the semi-smooth Newton method for* (3.33)*,* (3.34)*,* (3.48) *and hence (mPDAS-II) converges in a neighborhood of* (u, w, μ)*.*

Proof. Showing the existence of a solution to (3.33), (3.34), (3.48) is equivalent to the problem of finding a zero of the mapping

$$G : S_h \times V_h^{n-1} \times V_h^{n-1} \to S_h \times V_h^{n-1} \times V_h^{n-1}$$

where for $(u, w, \mu) \in S_h \times V_h^{n-1} \times V_h^{n-1}$ we define $G = (G_1, G_2, G_3)$ via

$$\begin{aligned}
&(G_1(u, w, \mu), \chi)_h := (u - u_h^{n-1}, \chi)_h + \tau(\nabla w, \rho\nabla\chi)\,,\\
&(G_2(u, w, \mu), \chi)_h := (w, \chi)_h - \gamma\varepsilon(\nabla u, \nabla\chi) - \frac{\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \chi)_h - \frac{\gamma}{\varepsilon}(\mu, \chi)_h - \frac{1}{\varepsilon}(f, \chi)_h\,,\\
&(G_3(u, w, \mu), \chi)_h := (H(u, \mu), \chi)_h\,.
\end{aligned}$$

The min-max-function $H(u, \mu)$, see (2.37), is slantly differentiable and a slanting function is given by $DH(u, \mu) = (0, 1)$ if $|u + \frac{\mu}{c}| \leq 1$ and $DH(u, \mu) = (-c, 0)$ otherwise, (see [GK07]). As a consequence G is slantly differentiable. Moreover similar as in [GK07] we can derive that the primal-dual active set method (mPDAS-II) is equivalent to a semi-smooth Newton method for G. We now get local convergence of (mPDAS-II) if we can show that the slanting function of G is invertible in a neighborhood of (u, w, μ) and the inverses are uniformly bounded, see [CNQ01, HIK02].

The semi-smooth derivative (slanting function) of G is invertible at $(\hat{u}, \hat{w}, \hat{\mu}) \in S_h \times V_h^{n-1} \times V_h^{n-1}$ if and only if we can show injectivity, i.e. that the solution $(\overline{u}, \overline{w}, \overline{\mu}) \in S_h \times V_h^{n-1} \times V_h^{n-1}$ of the following linear system

$$(\overline{u}, \chi)_h + \tau(\nabla \overline{w}, \rho \nabla \chi) = 0,\ \forall \chi \in S_h\,, \tag{3.49}$$

$$(\overline{w}, \chi)_h - \gamma\varepsilon(\nabla \overline{u}, \nabla \chi) - \frac{\gamma}{\varepsilon}(\overline{\mu}, \chi)_h = 0,\ \forall \chi \in V_h^{n-1}\,, \tag{3.50}$$

$$\overline{u}(p_j) = 0 \quad \text{if} \quad p_j \in \hat{\mathcal{A}} := \left\{ p_j \in \mathcal{M}_h^{n-1} \mid \left| \hat{u}(p_j) + \frac{\hat{\mu}(p_j)}{c} \right| > 1 \right\}\,, \tag{3.51}$$

$$\overline{\mu}(p_j) = 0 \quad \text{if} \quad p_j \in \hat{\mathcal{I}} := \mathcal{M}^{n-1} \setminus \hat{\mathcal{A}}\,, \tag{3.52}$$

is unique. Testing (3.49) with $\overline{w}$, (3.50) with $\overline{u}$ and using $(\overline{\mu}, \overline{u})_h = 0$ we obtain

$$\tau(\nabla \overline{w}, \rho \nabla \overline{w}) + \gamma\varepsilon(\nabla \overline{u}, \nabla \overline{u}) = 0\,. \tag{3.53}$$

This implies that $\overline{u}$ and $\overline{w}$ are constant. Then (3.49) gives $\overline{u} \equiv 0$. Using the fact that there exists a $p_j \in \mathcal{M}_h^{n-1}$ with $|u(p_j)| < 1$ and $\mu(p_j) = 0$ we can guarantee that $|\hat{u}(p_j) + \frac{\hat{\mu}(p_j)}{c}| \leq 1$ for $(\hat{u}, \hat{w}, \hat{\mu})$ in a ball around (u, w, μ). Hence, $\hat{\mathcal{I}} \neq \emptyset$ for $(\hat{u}, \hat{w}, \hat{\mu})$ out of this neighborhood. Testing in (3.50) with χ_j where $p_j \in J_h$ implies $\overline{w} \equiv 0$ and finally (3.52) and (3.50) yield $\overline{\mu} \equiv 0$.

The semi-smooth derivatives only differ if the active and inactive sets change. Since only a finite number of different choices of these sets are possible we obtain that the inverses are uniformly bounded for all $(\hat{u}, \hat{w}, \hat{\mu})$ with a non-vanishing inactive set $\hat{\mathcal{I}}$. Since we can find an open neighborhood of (u, w, μ), where the condition $\hat{\mathcal{I}} \neq \emptyset$ holds, we proved the theorem. □

Remark 3.7. *Let* (u, w, μ) *be a solution to* (3.33)*,* (3.34)*,* (3.48)*. The proof of Theorem 3.6 requires a neighborhood of* (u, w, μ)*, where the active sets do not vanish. This can limit the size of the neighborhood in which local convergence can be guaranteed. However in numerical simulations the mesh size always has to be chosen such that at least eight points lie across the interface. Hence the above mentioned condition never led to any problems in practice.*

Again as in the continuous setting we can proof results for the implicitly discretized free energy only if we impose a constraint on the time step width.

Corollary 3.8. *Theorem 3.6 holds also for the implicit discretization, i.e.* $* = n$*, if* $\tau < \frac{4\varepsilon^3}{\gamma \rho_{max}}$.

Proof. The proof follows along the lines of the proof of Theorem 3.6 if one can show injectivity. Together with $\psi_0'(u) = -u$, equation (3.50) changes to

$$(\overline{w}, \chi)_h - \gamma\varepsilon(\nabla \overline{u}, \nabla \chi) - \frac{\gamma}{\varepsilon}(\overline{\mu}, \chi)_h + \frac{\gamma}{\varepsilon}(\overline{u}, \chi)_h = 0$$

The same testing as above leads to $\tau(\nabla \overline{w}, \rho \nabla \overline{w}) + \gamma\varepsilon(\nabla \overline{u}, \nabla \overline{u}) - \frac{\gamma}{\varepsilon}(\overline{u}, \overline{u})_h = 0$ and testing (3.49) with $\overline{u}$ yields $(\overline{u}, \overline{u})_h = -\tau(\rho \nabla \overline{w}, \nabla \overline{u})$. We use Cauchy-Schwarz and Young's inequality similar to the proof of Lemma 2.12, and obtain $\nabla \overline{u} = 0$, if $\tau < \frac{4\varepsilon^3}{\gamma \rho_{max}}$. Subsequently we get $\nabla \overline{w} = 0$. We can now argue as in the proof of Theorem 3.6 that $\overline{u} = \overline{w} = 0$, which implies injectivity. □

Chapter 4

Solvers for the saddle point problem

The previous chapter introduced in detail the discretized primal-dual active set method applied to the Cahn–Hilliard gradient flow with obstacle potential. Most of the steps in Algorithm 3.4 are very simple value assignments and pose no further trouble. The only remaining challenge is the efficient solution of the arising saddle point structured system (3.26), which reads as

$$\begin{pmatrix} \begin{array}{cc} \multicolumn{2}{c}{\tau\mathbf{S}} \\ \mathbf{M}_{\boldsymbol{I}_k} & 0 \end{array} & \begin{array}{c} \mathbf{M}_{\boldsymbol{I}_k} \\ 0 \\ -\varepsilon\gamma\mathbf{S}_{\boldsymbol{I}_k\boldsymbol{I}_k} \end{array} \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_k \\ \boldsymbol{u}_{\boldsymbol{I}_k} \end{pmatrix} = \begin{pmatrix} \mathbf{M}\boldsymbol{u}^{n-1} - \mathbf{M}_{\boldsymbol{J}\boldsymbol{A}_k}\boldsymbol{u}_{\boldsymbol{A}_k} \\ -\frac{\gamma}{\varepsilon}\mathbf{M}_{\boldsymbol{I}_k}\boldsymbol{u}^{n-1}_{\boldsymbol{I}_k} + \varepsilon\gamma\mathbf{S}_{\boldsymbol{I}_k\boldsymbol{A}_k}\boldsymbol{u}_{\boldsymbol{A}_k} \end{pmatrix}. \tag{4.1}$$

The models using non-constant diffusional mobility like Algorithm 3.5 and even Algorithm 3.6 have a very similar structure. Hence all we describe below pertaining the constant mobility case is applicable to the other situations as well.

In this chapter we present a selection of methods to tackle this linear algebra problem. We also give some remarks on the effects the selection of this interior solver has on the primal-dual active set method. A thorough overview on the numerical treatment of saddle point problems is given by Benzi, Golub and Liesen [BGL05]. Firstly we will present an adapted over-relaxation Gauß–Seidel type scheme, used previously to solve the variational inequality problem, see Section 3.2. Instead of a projection a distinction between two kinds of arising 2 × 2-blocks (on the active respective inactive sets) will be necessary.

The discretization with linear finite element functions and structured meshes gives rise to sparse matrices with very few non-zero entries, since they reflect the connection of the vertices by edges. For calculations in upto two space dimensions the current state of the art is the use of efficient direct solution methods. Those methods depend on a good preordering scheme. For sensible simulations in higher dimensions the use of direct solvers is heavily depending on the underlying hardware structure since the arising system is still sparse, but the fill in takes its toll and a large amount of memory for storage has to be provided.

There is also a variety of other methods for this type of problem. Kornhuber and Gräser use methods based on a monotone multigrid approach for the solution of the Cahn–Hilliard variational problem, see e.g. [GK07] or [GK09]. Welford and Kay implemented a multigrid method for the solution of the Cahn–Hilliard problem with a logarithmic potential and present a result pertaining the mesh independence, see [KW06]. A block multigrid solver was developed by Banas and Nürnberg, see [BN09].

In the next chapter a preconditioned Schur complement approach will be presented, where we can use a conjugate gradient method on a suitable Hilbert space.

4.1 Block Gauss–Seidel type solver (BSOR)

The saddle point problem (4.1) has a structure akin to the variational inequality formulation (3.6)-(3.7). Here in this situation the space $\mathbf{K}$ for the test functions omits no constraints and hence results in an equality. However to apply the block successive over-relaxation method from before we have to augment the system by the condition on the active set. Thus we start out with

$$\begin{pmatrix} \tau\mathbf{S}_{\boldsymbol{I}_k\boldsymbol{I}_k} & \tau\mathbf{S}_{\boldsymbol{I}_k\boldsymbol{A}_k} & \mathbf{M}_{\boldsymbol{I}_k} & 0 \\ \tau\mathbf{S}_{\boldsymbol{A}_k\boldsymbol{I}_k} & \tau\mathbf{S}_{\boldsymbol{A}_k\boldsymbol{A}_k} & 0 & \mathbf{M}_{\boldsymbol{A}_k} \\ \mathbf{M}_{\boldsymbol{I}_k} & 0 & -\gamma\varepsilon\mathbf{S}_{\boldsymbol{I}_k\boldsymbol{I}_k} & 0 \\ 0 & 0 & 0 & \mathbf{Id}_{\boldsymbol{A}_k} \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_{\boldsymbol{I}_k} \\ \boldsymbol{w}_{\boldsymbol{A}_k} \\ \boldsymbol{u}_{\boldsymbol{I}_k} \\ \boldsymbol{u}_{\boldsymbol{A}_k} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{\boldsymbol{I}_k}\boldsymbol{u}^{n-1}_{\boldsymbol{I}_k} \\ \mathbf{M}_{\boldsymbol{A}_k}\boldsymbol{u}^{n-1}_{\boldsymbol{A}_k} \\ -\frac{\gamma}{\varepsilon}\mathbf{M}_{\boldsymbol{I}_k}\boldsymbol{u}^{n-1}_{\boldsymbol{I}_k} \\ \pm\mathbf{1}_{\boldsymbol{A}_k} \end{pmatrix}. \tag{4.2}$$

Applying the same reordering of the blocks to write this system in $(\boldsymbol{u}_i, \boldsymbol{w}_i)^t$ we get three different types of blocks in the system matrix, namely

$$\underline{\mathbf{A}}_{ij} = \begin{cases} \begin{pmatrix} \tau\mathbf{S}_{ij} & -\mathbf{M}_{ij} \\ -\mathbf{M}_{ij} & -\gamma\varepsilon\mathbf{S}_{ij} \end{pmatrix}, & \text{if } i \in \boldsymbol{I}_k \text{ and } j \in \boldsymbol{I}_k, \\ \begin{pmatrix} \tau\mathbf{S}_{ij} & -\mathbf{M}_{ij} \\ 0 & -\delta_{ij} \end{pmatrix}, & \text{if } i \in \boldsymbol{A}_k \text{ and } j \in \boldsymbol{A}_k, \\ \begin{pmatrix} \tau\mathbf{S}_{ij} & 0 \\ 0 & 0 \end{pmatrix}, & \text{otherwise.} \end{cases}$$

Now we can apply the general block SOR Algorithm 3.1 from before, where $\mathbf{Z}$ is the whole space. Hence the projection is no longer necessary and can be omitted. We will still make a distinction between inactive and active vertices for the explicit solution formula on the vertices to save computation time.

Algorithm 4.1 *Block SOR solver for PDAS saddle point system* (PDAS-BSOR)

```
1. Set $l = 1$ and $\boldsymbol{u}_{\boldsymbol{A}_k^\pm} = \pm\boldsymbol{1}$, initialize $\boldsymbol{u}_{\boldsymbol{I}_0}^{(0)}$ and $\boldsymbol{w}^{(0)}$
2. For $i = 1, \dots, N$ calculate the right hand sides
       $\boldsymbol{g}_i := \mathbf{M}_{ii}\boldsymbol{u}_i^{n-1} + \tau \sum_{j=1}^{i-1} \mathbf{S}_{ij}\boldsymbol{w}_j^{(l)} + \tau \sum_{j=i+1}^{N} \mathbf{S}_{ij}\boldsymbol{w}_j^{(l-1)},$
       $\boldsymbol{h}_i := \frac{\gamma}{\varepsilon}\mathbf{M}_{ii}\boldsymbol{u}_i^{n-1} + \gamma\varepsilon \sum_{j=1}^{i-1} \mathbf{S}_{ij}\boldsymbol{u}_j^{(l)} + \gamma\varepsilon \sum_{j=i+1}^{N} \mathbf{S}_{ij}\boldsymbol{u}_j^{(l-1)}$
   and obtain the new iterates for the two cases. If $i \in \boldsymbol{A}_k$ we just need to update
       $\boldsymbol{w}_i^{(l)} = \frac{1}{\tau\mathbf{S}_{ii}}\left(\boldsymbol{g}_i - \mathbf{M}_{ii}\boldsymbol{u}_i^{(l)}\right)$   (4.3)
   otherwise it suffices to set
       $\boldsymbol{u}_i^{(l)} = \frac{\mathbf{M}_{ii}\boldsymbol{g}_i + \tau\mathbf{S}_{ii}\boldsymbol{h}_i}{\gamma\varepsilon\tau\mathbf{S}_{ii}^2 + \mathbf{M}_{ii}^2}$ and $\boldsymbol{w}_i^{(l)} = \frac{1}{\tau\mathbf{S}_{ii}}\left(\boldsymbol{g}_i - \mathbf{M}_{ii}\boldsymbol{u}_i^{(l)}\right).$   (4.4)
3. Stop if $\|\boldsymbol{u}^{(l)} - \boldsymbol{u}^{(l-1)}\|_2 < tol$, else set $l = l + 1$ and goto 2.
```

Note that $\boldsymbol{h}_i$ needs to be computed only if $i \in \boldsymbol{I}_k$ in step 2.

4.2 Direct multi-frontal solver UMFPack (UMF)

In many large scale applications with sparse structured systems the use of iterative solvers is not always the most efficient method. Depending on the structure and size of the system as well as the amount of available fast storage space for the factorized matrix direct solution techniques can be very competitive. Especially in lower space dimensions, i.e. one or two, the arising system (4.1) comprises of very few entries per column and hence the application of sophisticated sparse symmetric direct solvers is superior to the use of iterative methods. Similar experiences have been recorded in alike situations, see for example Janna, Comerlati and Gambolati [JCG09] for a comparative study in the case of elastic structural problems.

Basically all direct solution methods are based on some kind of LU factorization of the system matrix. The historical example therefore would be the Gauss–Algorithm taught in almost every elemental linear algebra or numerics course. The dependence of the algorithm on the ordering of the system nodes can easily be seen. A rudimentary method to adress this is given by pivoting strategies, like for example column pivoting. There is a large variety of different direct solver packages available using a similar method. In the last decades the pivoting strategies have become very advanced and result in a huge gain in efficiency concerning the factorization algorithm. For a list of available solvers we would like to refer the reader to the listing by Davis, see [Dav06b]. An evaluation of such solvers for symmetric systems was done by Gould, Hu and Scott [GSH07] and in the case of unsymmetric systems by Gupta, see [Gup02].

For our application we decided to use the software package UMFPack written by Timothy Davis [Dav04a], where an unsymmetric multi-frontal method for sparse LU factorization is realized, see Davis and Duff [DD99, Dav04b, DD97]. The used multi-frontal method introduced by Duff and Reid [DR83], which is a generalization of the frontal method of Irons [Iro70], gains its speed up from the fact that the reordering, necessary for the pivoting, is not applied directly to the matrix but aggregated into one frontal matrix. We would like to refer the reader to the works of Liu [Liu92], where a comprehensive presentation of the method is given, as well as to the book of Davis on the direct solution methods for sparse systems [Dav06a].

We want to point out the applicability of this software package to indefinite systems. The most important point is the use of a mixture of 1×1 and 2×2 pivots, see Duff, Reid, Munksgaard and Nielsen [DRMN79]. Essentially this deals with occurring zero pivots. This property will be very important later, when we use this package for the Schur complement solver.

In this work we applied this package to solve the whole saddle point system (4.1) at once by reassembling the system in each primal-dual active set iteration and factorizing this updated system. Despite this overhead the direct solver is competitive, which can be seen in the chapter on numerical experiments.

Chapter 5

Schur complement type solvers

The numerical experiments we present in Chapter 6 show that the iterative solver based on the block successive over-relaxation method is not competitive in comparison to the primal-dual active set method together with the direct method, i.e. UMFPack, in two spatial dimensions. However, especially in simulations with three spatial dimensions, the direct solver requires huge amounts of memory for the storage of the factorized system due to fill-in and subsequently the efficiency is vastly decreased. The primal-dual active set method replaces the variational inequality problem by an iterative process, where only linear systems of equations have to be solved. Just using the block SOR method here does not lead to better results than the pBSOR method, but we now can use stronger iterative methods, like for example just a conjugate gradient or a multigrid method.

The symmetry and the saddle point structure of the system of equations (4.1), i.e. (3.26), and those given in the non-constant mobility case (3.39) and (3.42) allow for a further reduction of the problem by means of a Schur complement decomposition. The resulting system is again symmetric and positive definite and can thus be solved using a conjugate gradient method. The convergence speed depends heavily on a good preconditioning of the system. Due to the special structure of our problem we can adapt the ideas of Bänsch, Morin and Nochetto, see [BMN10]. The theoretical results there are not directly applicable due to the degeneracy of one of the operator parts, caused by the fact that it is given by the Laplacian on the inactive set only. We illustrate this point in the discussion in one spatial dimension, see Section 5.4.

Ahead of applying the Schur complement decomposition, we introduce a simple reformulation by shifting the variables by a constant in such a way that they carry no longer any mass. Due to this reformulation we can easily invert the blocks of (4.1) separately and subsequently eliminate one of the unknowns. Note that the following results are formulated for the constant mobility case only, but are analogously applicable to the other formulations with non-constant mobility as well.

We begin by restating the saddle point problem of (PDAS-II), i.e. the finite element formulation of the system of equations corresponding to (4.1), used in step 3 of Algorithm 3.3:

Determine $(u_k, w_k) \in S_h \times S_h$ for given u_h^{n-1} and $\mathcal{A}_k^{\pm}$ such that

$$\frac{1}{\tau}(u_k - u_h^{n-1}, \chi)_h + (\nabla w_k, \nabla \chi) = 0 \qquad \forall\, \chi \in S_h\,, \tag{3.21}$$

$$(w_k, \tilde{\chi})_h - \varepsilon\gamma(\nabla u_k, \nabla \tilde{\chi}) - \frac{\gamma}{\varepsilon}(\psi_0'(u_h^*), \tilde{\chi})_h = 0 \qquad \forall\, \tilde{\chi} \in \tilde{S}_{h,k}\,, \tag{3.22}$$

$$u_k(p_j) = \pm 1 \quad \text{if } p_j \in \mathcal{A}_k^{\pm} \tag{3.23}$$

holds.

5.1 Reformulation of the saddle point system

The Schur complement formulation of the saddle point problem given by equations (3.21) and (3.22) requires the inverse of the Laplacian with Neumann boundary conditions in equation (3.21). It is well known that additional conditions are necessary to obtain a unique solution, see e.g. Bochev and Lehoucq [BL05].

In our context the natural condition is given by the subspace defined by the mass restrictions we already imposed on u and w. To obtain a more unified notation we make a change in variables to obtain a system stated in variables that carry no mass, i.e. variable w is replaced by $v := w - \fint_\Omega w$. Note that this quantity has already occurred naturally in the derivation of the gradient flow formulation. Something similar will be done to the variable u to ensure that both variables, which the Schur complement system is formulated in, are given on the same subspace. Even without such a transformation of the problem, a Schur complement decomposition would still be possible, but the Laplacian operator on the upper left block would map from one space with a mass restriction to one with another mass restriction and thus lead to complicated expressions.

Ahead of replacing the variables, we need some additional notation.

5.1.1 Notation

We introduce characteristic finite element functions $1_{\mathcal{A}_k}, 1_{\mathcal{I}_k} \in S_h$ for the active and inactive sets and a signed function $1_{\mathcal{A}_k^{\pm}} \in S_h$ fulfilling

$$1_{\mathcal{A}_k}(p_j) = \begin{cases} 1 & \text{if } p_j \in \mathcal{A}_k, \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad 1_{\mathcal{I}_k}(p_j) = \begin{cases} 1 & \text{if } p_j \in \mathcal{I}_k, \\ 0 & \text{otherwise} \end{cases}$$

$$1_{\mathcal{A}_k^{\pm}}(p_j) = \begin{cases} \pm 1 & \text{if } p_j \in \mathcal{A}_k^{\pm}, \\ 0 & \text{otherwise.} \end{cases}$$

Additionally we denote the restriction of u_k onto the active set by $u_{\mathcal{A}_k} := I_h(u_k 1_{\mathcal{A}_k})$, where $I_h : C^0(\overline{\Omega}) \to S_h$ denotes the standard interpolation operator at the nodes as in Section 3.1. Analogously we define $u_{\mathcal{I}_k} := I_h(u_k 1_{\mathcal{I}_k})$. As a result we can define the mass $m := (u_h^{n-1}, 1)_h$, the active mass $m_{\mathcal{A}_k} := (u_{\mathcal{A}_k}, 1)_h = (1_{\mathcal{A}_k^\pm}, 1)_h$ due to (3.23), the inactive mass $m_{\mathcal{I}_k} := m - m_{\mathcal{A}_k} = (u_h^{n-1} - 1_{\mathcal{A}_k^\pm}, 1)_h$ as well as the constant $\overline{m}_k := \frac{m_{\mathcal{I}_k}}{(1_{\mathcal{I}_k}, 1)_h}$. Having all these constants we can define an admissible concentration distribution d_k by fixing the active part and distributing the remaining mass equally across the inactive set, i.e.

$$d_k := 1_{\mathcal{A}_k^\pm} + \overline{m}_k 1_{\mathcal{I}_k}. \tag{5.1}$$

The splitting of Ω into active and inactive sets and the fact that the second equation is only used on a subset of Ω, makes it very convenient to use not only the standard finite element space S_h, as introduced in Section 3.1, but also some other spaces. For any given subset $\omega \subset \Omega$ we define

$$S_\omega := \{\chi \in S_h \mid \chi(p_j) = 0 \text{ if } p_j \in \Omega \setminus \omega\}. \tag{5.2}$$

Additionally we use the mass constrained finite element space, similar to the definition used in Section 3.7 to derive existence and uniqueness results, by setting

$$S_{\omega,m} := \{\chi \in S_\omega \mid (\chi, 1)_h = m\}. \tag{5.3}$$

Later the subset ω will be the active or inactive set as necessary. Hence the space of test functions $\tilde{S}_{h,k}$, used in (3.22), is also denoted by $S_{\mathcal{I}_k}$, i.e. the space of the finite element functions with support on the inactive set.

5.1.2 Introduction of the mass free variables

Previously we already had a formulation with a mass free chemical potential $w - \fint_\Omega w$. We revert back to this formulation in v replacing w_k by using a constant $\overline{w}_k := \frac{(w_k,1)_h}{(1,1)_h}$ and setting

$$v_k := w_k - \overline{w}_k 1. \tag{5.4}$$

Note that this way v_k is orthogonal to the kernel of the Laplacian, see Section 5.1.3. Testing (3.21) with $\chi \equiv 1$ results in a restriction on the mass of the variable u_k. Furthermore a part of it is already fixed by the condition on the active sets (3.23), namely this quantity is given by the active mass $m_{\mathcal{A}_k}$. Thus we obtain a restriction on the mass of $u_{\mathcal{I}_k}$, which is given by the constant $m_{\mathcal{I}_k}$ defined above.

Equation (3.22) is only used to determine u_k on the inactive set. With the aim of mass free variables in mind, we add a constant to u_k such that the resulting restriction is given by a 0–mass condition on the inactive set. We replace the variable u_k in the system of equations by defining the new variable $z_k \in S_h$ by

$$z_k := u_k - \overline{m}_k 1_\Omega.$$

For the actual reformulation we split the finite element function z_k into two parts, one on the active set, the other on the inactive set by multiplication with the characteristic functions $1_{\mathcal{A}_k}$ and $1_{\mathcal{I}_k}$. In the following we use

$$z_k = z_{\mathcal{A}_k} + z_{\mathcal{I}_k},$$

where $z_{\mathcal{A}_k} := I_h(z_k 1_{\mathcal{A}_k}) \in S_{\mathcal{A}_k}$ and $z_{\mathcal{I}_k} := I_h(z_k 1_{\mathcal{I}_k}) \in S_{\mathcal{I}_k}$.
Due to the definition of the constant $\overline{m}_k$, we attain a new formulation of the problem in the variable $z_{\mathcal{I}_k} \in S_{\mathcal{I}_k,0}$ with the desired property, i.e. $(z_k, 1_{\mathcal{I}_k})_h = 0$. This will be a key property when we split the operator later on.
Utilizing definition (5.1) as well as the information on the active set due to (3.23), namely $z_{\mathcal{A}_k} = 1_{\mathcal{A}_k^\pm} - \overline{m}_k 1_{\mathcal{A}_k}$, we can rewrite the relation between u_k and z_k with the help of the admissible distribution d_k and obtain

$$\begin{aligned} u_k &= z_k + \overline{m}_k 1_\Omega = z_{\mathcal{I}_k} + z_{\mathcal{A}_k} + \overline{m}_k(1_{\mathcal{I}_k} + 1_{\mathcal{A}_k}) \\ &= z_{\mathcal{I}_k} + (u_{\mathcal{A}_k} - \overline{m}_k 1_{\mathcal{A}_k}) + \overline{m}_k(1_{\mathcal{I}_k} + 1_{\mathcal{A}_k}) \\ &= z_{\mathcal{I}_k} + (1_{\mathcal{A}_k^\pm} + \overline{m}_k 1_{\mathcal{I}_k}) \\ &= z_{\mathcal{I}_k} + d_k. \end{aligned}$$

Since u_k, z_k and d_k are piecewise linear functions, $\nabla u_k = \nabla z_{\mathcal{I}_k} + \nabla d_k$ holds.
Plugging both variable transformations in the first equation of (PDAS-II), i.e. (3.21), we obtain an equivalent formulation by the following short calculation:

$$\begin{aligned} 0 &= \tau(\nabla w_k, \nabla \chi) + (u_k - u_h^{n-1}, \chi)_h \\ &= \tau(\nabla(v_k + \overline{w}_k 1_\Omega), \nabla \chi) + (z_{\mathcal{I}_k} + d_k - u_h^{n-1}, \chi)_h \\ &= \tau(\nabla v_k, \nabla \chi) + (z_{\mathcal{I}_k}, \chi)_h - (u_h^{n-1} - d_k, \chi)_h. \end{aligned} \tag{5.5}$$

Ahead of transforming (3.22) we introduce the parameter $\Theta_\psi \in \{0, 1\}$ for the distinction between the implicit ($\Theta_\psi = 1$) and semi-implicit ($\Theta_\psi = 0$) time discretization and replace the term previously marked by an asterisk by two terms and obtain

$$(w_k, \tilde{\chi})_h - \varepsilon\gamma(\nabla u_k, \nabla \tilde{\chi}) - \frac{\Theta_\psi \gamma}{\varepsilon}(\psi_0'(u_k), \tilde{\chi})_h = \frac{(1-\Theta_\psi)\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \tilde{\chi})_h \qquad \forall\, \tilde{\chi} \in S_{\mathcal{I}_k}.$$

Finally we replace w_k and u_k with the new variables and reorder the terms, giving

$$\begin{aligned} 0 &= (w_k, \tilde{\chi})_h - \varepsilon\gamma(\nabla u_k, \nabla \tilde{\chi}) - \frac{\Theta_\psi \gamma}{\varepsilon}(\psi_0'(u_k), \tilde{\chi})_h - \frac{(1-\Theta_\psi)\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \tilde{\chi})_h \\ &= (v_k + \overline{w}_k 1, \tilde{\chi})_h - \varepsilon\gamma(\nabla(z_{\mathcal{I}_k} + d_k), \nabla \tilde{\chi}) - \frac{\Theta_\psi \gamma}{\varepsilon}(\psi_0'(z_{\mathcal{I}_k} + d_k), \tilde{\chi})_h \\ &\quad - \frac{(1-\Theta_\psi)\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \tilde{\chi})_h \\ &= (v_k, \tilde{\chi})_h - \varepsilon\gamma(\nabla z_{\mathcal{I}_k}, \nabla \tilde{\chi}) - \frac{\Theta_\psi \gamma}{\varepsilon}(\psi_0'(z_{\mathcal{I}_k} + d_k), \tilde{\chi})_h \\ &\quad - \frac{(1-\Theta_\psi)\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \tilde{\chi})_h + \overline{w}_k(1, \tilde{\chi})_h - \varepsilon\gamma(\nabla d_k, \nabla \tilde{\chi}). \end{aligned} \tag{5.6}$$

The new variables introduced in (5.4) and (5.1) satisfy the desired restrictions on the mass by construction. We recapitulate the essential key properties in the following lemma.

Lemma 5.1. *The above variables pertain the following conditions:*

1. $(z_{\mathcal{I}_k}, 1)_h = 0 \quad \Longleftrightarrow \quad (u_{\mathcal{I}_k}, 1)_h = m_{\mathcal{I}_k}$.
2. $(v_k, 1)_h = 0$.
3. $(u_h^{n-1} - d_k, 1)_h = 0$.
4. *If* $\psi_0(u) = \frac{1}{2}(1-u^2)$ *then*

$$\overline{w}_k = \frac{(v_k, 1_{\mathcal{I}_k})_h - \varepsilon\gamma(\nabla(z_{\mathcal{I}_k} - d_k), \nabla 1_{\mathcal{I}_k}) + \frac{(1-\Theta_\psi)\gamma}{\varepsilon}(u_{\mathcal{I}_k}^{n-1}, 1_{\mathcal{I}_k})_h}{(1_{\mathcal{I}_k}, 1_{\mathcal{I}_k})_h} + \frac{\Theta_\psi \gamma \overline{m}_k}{\varepsilon}.$$

Proof. 1. This equivalence follows directly from the linearity of the scalar product used in the following short computation:

$$(z_{\mathcal{I}_k}, 1)_h = (u_k - \overline{m}_k 1, 1_{\mathcal{I}_k})_h = (u_k, 1_{\mathcal{I}_k})_h - \overline{m}_k (1, 1_{\mathcal{I}_k})_h = (u_{\mathcal{I}_k}, 1)_h - m_{\mathcal{I}_k}.$$

2. Recalling the definition of v_k given by (5.4), we obtain

$$(v_k, 1)_h = (w_k, 1)_h - \frac{(w_k, 1)_h}{(1,1)_h}(1,1)_h = 0.$$

3. The way the variable d_k was designed, some terms in (5.5) vanish, when we test the equation with 1. Utilizing the definition of $\overline{m}_k$ we obtain

$$\begin{aligned}
(u_h^{n-1} - d_k, 1)_h &= (u_h^{n-1} - (1_{\mathcal{A}_k^\pm} + \overline{m}_k 1_{\mathcal{I}_k}), 1)_h \\
&= (u_h^{n-1} - 1_{\mathcal{A}_k^\pm}, 1)_h - \overline{m}_k (1_{\mathcal{I}_k}, 1)_h \\
&= (u_h^{n-1} - 1_{\mathcal{A}_k^\pm}, 1)_h - \frac{m_{\mathcal{I}_k}}{(1_{\mathcal{I}_k}, 1)_h}(1_{\mathcal{I}_k}, 1)_h \\
&= (u_h^{n-1} - 1_{\mathcal{A}_k^\pm}, 1)_h - m_{\mathcal{I}_k} \\
&= (u_h^{n-1} - 1_{\mathcal{A}_k^\pm}, 1)_h - (u_h^{n-1} - 1_{\mathcal{A}_k^\pm}, 1)_h = 0.
\end{aligned}$$

4. The last assertion follows by testing (5.5) with $\tilde{\chi} = 1_{\mathcal{I}_k}$ and using $\psi_0(u) = \frac{1}{2}(1-u^2)$, which leads to

$$\begin{aligned}
0 &= (v_k, 1_{\mathcal{I}_k})_h - \varepsilon\gamma(\nabla z_{\mathcal{I}_k}, \nabla 1_{\mathcal{I}_k}) + \frac{\Theta_\psi \gamma}{\varepsilon}(u_{\mathcal{I}_k}, 1_{\mathcal{I}_k})_h \\
&\quad + \frac{(1-\Theta_\psi)\gamma}{\varepsilon}(u_{\mathcal{I}_k}^{n-1}, 1_{\mathcal{I}_k})_h + \overline{w}_k (1_{\mathcal{I}_k}, 1_{\mathcal{I}_k})_h - \varepsilon\gamma(\nabla \overline{u}_k, \nabla 1_{\mathcal{I}_k}) \\
&= (v_k, 1_{\mathcal{I}_k})_h - \varepsilon\gamma(\nabla(z_{\mathcal{I}_k} - d_k), \nabla 1_{\mathcal{I}_k}) + \frac{\Theta_\psi \gamma \overline{m}_k}{\varepsilon}(1_{\mathcal{I}_k}, 1_{\mathcal{I}_k})_h \\
&\quad + \frac{(1-\Theta_\psi)\gamma}{\varepsilon}(u_{\mathcal{I}_k}^{n-1}, 1_{\mathcal{I}_k})_h + \overline{w}_k (1_{\mathcal{I}_k}, 1_{\mathcal{I}_k})_h.
\end{aligned}$$

□

Finally we formulate the primal-dual active set algorithm in the new variables. We replace the system of equations used in step 2 of (PDAS-II) by the equivalent equations given by (5.5) and (5.6). Please note that we use an additional step to calculate u_k and w_k from the new variables before we continue in the algorithm. This additional computational effort is not necessary, but allows the use of the same implementation as for the earlier given algorithm.

Algorithm 5.1 *Mass free primal-dual active set algorithm* (mfPDAS)

1. Set $k = 0$, initialize $\mathcal{A}_0^{\pm}$ and define $\mathcal{I}_0 = J_h \setminus (\mathcal{A}_0^{+} \cup \mathcal{A}_0^{-})$.
2. Solve for $(z_{\mathcal{I}_k}, v_k, \overline{w}_k) \in S_{\mathcal{I}_k,0} \times S_{h,0} \times \mathbb{R}$
$$\tau(\nabla v_k, \nabla\chi) + (z_{\mathcal{I}_k}, \chi)_h = (u_h^{n-1} - d_k, \chi) \qquad \forall\, \chi \in S_h\,, \tag{5.7}$$
$$\begin{aligned}(v_k,\tilde{\chi})_h - \varepsilon\gamma(\nabla z_{\mathcal{I}_k}, \nabla\tilde{\chi}) - \frac{\Theta_\psi\gamma}{\varepsilon}(\psi_0'(z_{\mathcal{I}_k} + d_k), \tilde{\chi})_h \\ = \frac{(1-\Theta_\psi)\gamma}{\varepsilon}(\psi_0'(u_h^{n-1}), \tilde{\chi})_h + \varepsilon\gamma(\nabla d_k, \nabla\tilde{\chi}) - \overline{w}_k(1,\tilde{\chi})_h \qquad \forall\, \tilde{\chi} \in S_{\mathcal{I}_k},\end{aligned} \tag{5.8}$$
$$\overline{w}_k = \frac{(v_k, 1_{\mathcal{I}_k})_h - \varepsilon\gamma(\nabla(z_{\mathcal{I}_k} - d_k), \nabla 1_{\mathcal{I}_k}) + \frac{(1-\Theta_\psi)\gamma}{\varepsilon}(u_{\mathcal{I}_k}^{n-1}, 1_{\mathcal{I}_k})_h}{(1_{\mathcal{I}_k}, 1_{\mathcal{I}_k})_h} + \frac{\Theta_\psi\gamma\overline{m}_k}{\varepsilon}. \tag{5.9}$$
3. Set $u_k = z_{\mathcal{I}_k} + d_k$ and $w_k = v_k + \overline{w}_k 1$.
4. Define $\mu_k \in S_h$ via
$$\begin{aligned}(\mu_k, \overline{\chi})_h =& \frac{\varepsilon}{\gamma}(w_k, \overline{\chi})_h - \varepsilon^2(\nabla u_k, \nabla\overline{\chi}) \\ &- \Theta_\psi(\psi_0'(u_k), \overline{\chi})_h - (1-\Theta_\psi)(\psi_0'(u_h^{n-1}), \overline{\chi})_h \qquad \forall\, \overline{\chi} \in S_{\mathcal{A}_k},\end{aligned} \tag{5.10}$$
$$\mu_k(p_j) = 0 \qquad \forall\, p_j \in \mathcal{I}_k. \tag{5.11}$$
5. Set $\mathcal{A}_{k+1}^{+} := \{p_j \in J_h \mid u_k(p_j) + \frac{\mu_k(p_j)}{c} > 1\}$,
$\mathcal{A}_{k+1}^{-} := \{p_j \in J_h \mid u_k(p_j) + \frac{\mu_k(p_j)}{c} < -1\}$ and
$\mathcal{I}_{k+1} := J_h \setminus (\mathcal{A}_{k+1}^{+} \cup \mathcal{A}_{k+1}^{-})$.
6. If $\mathcal{A}_{k+1}^{\pm} = \mathcal{A}_k^{\pm}$ stop, otherwise set $k = k + 1$ and goto 2.

Before further discussing the Schur complement formulation of the saddle point problem, we rephrase the above algorithm again by means of an algebraic notation. The associated matrix representation of the Laplacian is not given by the stiffness matrix only, but also uses the mass matrix. We denote the discrete Laplacian by $\mathbf{L} := \mathbf{M}^{-1}\mathbf{S}$ and set $\boldsymbol{e} := (1, 1, \ldots, 1)^T \in \mathbb{R}^{|\boldsymbol{J}|}$. Note that the vector $\boldsymbol{e}_{\boldsymbol{I}_k}$ has to be understood, as before, as the restriction of the vector onto the index set $\boldsymbol{I}_k$. Furthermore with $\boldsymbol{e}_{\boldsymbol{A}_k^{\pm}} \in \mathbb{R}^{|\boldsymbol{A}_k|}$ we denote the vector consisting of ±1 depending on which active set the vertex is associated with. Additionally we use the following abbreviations

$$\begin{aligned}
\mathbf{C} :=& \mathbf{L}_{\boldsymbol{I}_k} - \frac{\Theta_\psi}{\varepsilon^2}\mathbf{Id}_{\boldsymbol{I}_k} \in \mathbb{R}^{|\boldsymbol{I}_k|\times|\boldsymbol{I}_k|},\\
\mathbf{E} :=& \begin{pmatrix} \mathbf{Id}_{\boldsymbol{I}_k} \\ 0 \end{pmatrix} \in \mathbb{R}^{|\boldsymbol{I}_k|\times|\boldsymbol{J}|},\\
\boldsymbol{g} :=& \left(\boldsymbol{u}^{n-1} - \boldsymbol{d}_k\right) \in \mathbb{R}^{|\boldsymbol{J}|},\\
\boldsymbol{h} :=& \varepsilon\gamma\mathbf{M}^{-1}_{\boldsymbol{I}_k}\left(\mathbf{S}\boldsymbol{d}\right)_{\boldsymbol{I}_k} - \frac{(1-\Theta_\psi)\gamma}{\varepsilon}\boldsymbol{u}^{n-1}_{\boldsymbol{I}_k} = \varepsilon\gamma\mathbf{E}^T\mathbf{L}\boldsymbol{d} - \frac{(1-\Theta_\psi)\gamma}{\varepsilon}\boldsymbol{u}^{n-1}_{\boldsymbol{I}_k} \in \mathbb{R}^{|\boldsymbol{I}_k|}.
\end{aligned}$$

Applying the algebraic notation to (mfPDAS) and multiplying the equations with $\mathbf{M}^{-1}$ and $\mathbf{M}^{-1}_{\boldsymbol{I}_k}$ respectively, we obtain the algebraic formulation (mfAPDAS) of the primal-dual active set algorithm, see Algorithm 5.2.

Algorithm 5.2 *Mass free algebraic primal-dual active set algorithm* (mfAPDAS)

1. Set $k = 0$, initialize $\boldsymbol{A}^{\pm}_0$, define $\boldsymbol{I}_0 = \boldsymbol{J} \setminus (\boldsymbol{A}^+_0 \cup \boldsymbol{A}^-_0)$.

2. Calculate $\overline{m}_k = \frac{\boldsymbol{e}^t\left(\mathbf{M}\boldsymbol{u}^{n-1} - \mathbf{M}_{\boldsymbol{J}\boldsymbol{A}_k}\boldsymbol{e}_{\boldsymbol{A}^{\pm}_k}\right)}{\boldsymbol{e}^t_{\boldsymbol{I}_k}\mathbf{M}_{\boldsymbol{I}_k}\boldsymbol{e}_{\boldsymbol{I}_k}}$.

3. Solve for $(\boldsymbol{z}_{\boldsymbol{I}_k}, \boldsymbol{v}_k, \overline{w}_k)$ with $\boldsymbol{e}^t_{\boldsymbol{I}_k}\mathbf{M}_{\boldsymbol{I}_k}\boldsymbol{z}_{\boldsymbol{I}_k} = \boldsymbol{e}^t\mathbf{M}\boldsymbol{v} = 0$ the system
$$\begin{pmatrix} \tau\mathbf{L} & \mathbf{E} \\ \mathbf{E}^t & -\varepsilon\gamma\mathbf{C} \end{pmatrix}\begin{pmatrix} \boldsymbol{v}_k \\ \boldsymbol{z}_{\boldsymbol{I}_k} \end{pmatrix} = \begin{pmatrix} \boldsymbol{g} \\ \boldsymbol{h} + (\overline{w}_k - \overline{m}_k\frac{\Theta_\psi\gamma}{\varepsilon})\boldsymbol{e}_{\boldsymbol{I}_k} \end{pmatrix} \tag{5.12}$$
together with
$$\overline{w}_k = \frac{\boldsymbol{e}^t_{\boldsymbol{I}_k}\mathbf{M}_{\boldsymbol{I}_k}\left(\boldsymbol{v}_{\boldsymbol{I}_k} + \frac{(1-\Theta_\psi)\gamma}{\varepsilon}\boldsymbol{u}^{n-1}_{\boldsymbol{I}_k}\right) + \varepsilon\gamma\boldsymbol{e}^t_{\boldsymbol{I}_k}\left(\mathbf{S}(\boldsymbol{z}_{\boldsymbol{I}_k} + \boldsymbol{d}_k)\right)_{\boldsymbol{I}_k} + \frac{\Theta_\psi\gamma\overline{m}_k}{\varepsilon}}{\boldsymbol{e}^t_{\boldsymbol{I}_k}\mathbf{M}_{\boldsymbol{I}_k}\boldsymbol{e}_{\boldsymbol{I}_k}}. \tag{5.13}$$

4. Set $\boldsymbol{u}_{\boldsymbol{A}_k} = \pm 1$, $\boldsymbol{u}_{\boldsymbol{I}_k} = \boldsymbol{z}_{\boldsymbol{I}_k} + \overline{m}_k\boldsymbol{e}_{\boldsymbol{I}_k}$ and $\boldsymbol{w}_k = \boldsymbol{v}_k + \overline{w}_k\boldsymbol{e}$.

5. Define $\boldsymbol{\mu}_{\boldsymbol{I}_k}, \boldsymbol{\mu}_{\boldsymbol{A}_k}$ via
$$\boldsymbol{\mu}_{\boldsymbol{A}_k} = \frac{\varepsilon}{\gamma}\boldsymbol{w}_{\boldsymbol{A}_k} - \varepsilon^2\mathbf{M}^{-1}_{\boldsymbol{A}_k}\left(\mathbf{S}_{\boldsymbol{A}_k\boldsymbol{J}}\boldsymbol{u}_k\right) + (1-\Theta_\psi)\boldsymbol{u}^{n-1}_{\boldsymbol{A}_k} + \Theta_\psi\boldsymbol{u}_{\boldsymbol{A}_k}, \tag{5.14}$$
$$\boldsymbol{\mu}_{\boldsymbol{I}_k} = 0. \tag{5.15}$$

6. Set $\boldsymbol{A}^+_{k+1} := \{i \in \boldsymbol{J} \mid \boldsymbol{u}_{ki} + \frac{\boldsymbol{\mu}_{ki}}{c} > 1\}$,
   $\boldsymbol{A}^-_{k+1} := \{i \in \boldsymbol{J} \mid \boldsymbol{u}_{ki} + \frac{\boldsymbol{\mu}_{ki}}{c} < -1\}$ and
   $\boldsymbol{I}_{k+1} := \boldsymbol{J} \setminus (\boldsymbol{A}^+_{k+1} \cup \boldsymbol{A}^-_{k+1})$.

7. If $\boldsymbol{A}^{\pm}_{k+1} = \boldsymbol{A}^{\pm}_k$ stop, otherwise set $k = k+1$ and goto 2.

Remark 5.2. *Note that the mass matrix* $\mathbf{M}$ *is a regular diagonal matrix due to the projected scalar product* $(\cdot,\cdot)_h$ *we use. This scalar product has its algebraic equivalent* $\cdot_m$ *defined by* $\boldsymbol{x} \cdot_m \boldsymbol{y} := \boldsymbol{x} \cdot \mathbf{M}\boldsymbol{y} = \sum_{j \in \boldsymbol{J}} \boldsymbol{x}_j \mathbf{M}_{jj} \boldsymbol{y}_j$.

Also equation (5.13), defining $\overline{w}_k$, can be evaluated on one arbitrary inactive vertex only. A few experiments showed no difference in comparison to the original formulation using the average over all inactive vertices, which might be slightly more stable.
Similarly to the original formulation (4.1), the system of equations (5.12) has saddle point structure. Also the system is self-adjoint with respect to the scalar product $\cdot_m$. The difference is given by the 0-mass condition on the variable $\boldsymbol{v}_k$. On the corresponding subspace the upper left block $\mathbf{L}$ can be inverted. This is the concern of the following section. Using this we can subsequently eliminate the variable $\boldsymbol{v}_k$ from the system.
At some point in the following discussion it is more convenient to use a symmetric formulation of the saddle point problem (5.12). We obtain it by multiplication of the system with the square root of the mass matrix and inserting a product of this root of the mass matrix with its inverse on the right side of the matrix. We obtain the following equivalent system, which is now symmetric with respect to the l_2 scalar product:

$$\begin{pmatrix} \tau\hat{\mathbf{L}} & \mathbf{E} \\ \mathbf{E}^t & -\varepsilon\gamma(\hat{\mathbf{L}} - \frac{\Theta_\psi}{\varepsilon^2}\mathbf{Id})_{\boldsymbol{I}} \end{pmatrix} \begin{pmatrix} \mathbf{M}^{\frac{1}{2}}\boldsymbol{v}_k \\ \mathbf{M}_{\boldsymbol{I}}^{\frac{1}{2}}\boldsymbol{z}_{\boldsymbol{I}_k} \end{pmatrix} = \begin{pmatrix} \mathbf{M}^{\frac{1}{2}}\boldsymbol{g} \\ \mathbf{M}\boldsymbol{I}^{\frac{1}{2}}(\boldsymbol{h} + (\overline{w}_k - \overline{m}_k\frac{\Theta_\psi\gamma}{\varepsilon})\boldsymbol{e}_{\boldsymbol{I}_k}) \end{pmatrix}, \quad (5.16)$$

where we denote the rephrased matrix representation of the discrete Laplacian by $\hat{\mathbf{L}} := \mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}$.

5.1.3 Inverse Laplacian with Neumann boundary conditions

With the aim of splitting the system of equations (5.12), we have to discuss how to solve the Laplace problem in equation (5.7). More precisely we need to take a closer look at the upper row of (5.12). The Neumann boundary conditions require an additional constraint to ensure uniqueness of the solution. That is why the assembled stiffness matrix $\mathbf{L}$ is only positive semi-definite. This can easily be verified since $\ker(\mathbf{L}) = \operatorname{span}\{\boldsymbol{e}\}$.
Since we already imposed the 0-mass condition on both of our variables we will deal with this problem by restricting the operator to the subspace $\mathbb{U} := \{\boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{J}|} \mid \boldsymbol{x} \cdot_m \boldsymbol{e} = 0\}$.

Remark 5.3. *As shown in Lemma 5.1 the mass of the vector* $\boldsymbol{g}$ *vanishes, i.e.* $\boldsymbol{e} \cdot_m \boldsymbol{g} = 0$. *The same is true for* $\boldsymbol{e} \cdot_m \mathbf{E}\boldsymbol{z}_{\boldsymbol{I}} = 0$, *namely we get that* $\boldsymbol{g}$, $\mathbf{E}\boldsymbol{z}_{\boldsymbol{I}} \in \operatorname{span}\{\boldsymbol{e}\}^{\perp_m}$.

Remark 5.4. *The discrete Laplacian* $\mathbf{L}$ *is self-adjoint with respect to* $\cdot_m$. *Since* $\mathbf{M}$ *is a positive definite diagonal matrix and* $\mathbf{S}$ *maps all constant vectors to* 0, *we get* $\ker(\mathbf{L}) = \ker(\mathbf{S}) = \mathit{span}(\boldsymbol{e}) = \mathbb{U}^{\perp_m}$ *as well as* $\operatorname{im}(\mathbf{L}) = (\ker(\mathbf{L}))^{\perp_m} = (\mathit{span}(\boldsymbol{e}))^{\perp_m}$.

Similar statements are true for the symmetric version $\hat{\mathbf{L}}$, where the corresponding subspace is given by $\hat{\mathbb{U}} := \{\boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{J}|} \mid \boldsymbol{x} \cdot \mathbf{M}^{-\frac{1}{2}}\boldsymbol{e} = 0\}$. There is also a very close connection between both operators $\mathbf{L}$ and $\hat{\mathbf{L}}$. When we use $\mathbf{Y} := \mathbf{M}^{\frac{1}{2}}$ and $\mathbf{Z} := \mathbf{M}^{\frac{1}{2}}\mathbf{S}$, then $\mathbf{L} = \mathbf{Y}\mathbf{Z}$ and $\hat{\mathbf{L}} = \mathbf{Z}\mathbf{Y}$. Hence both matrices have exactly the same eigenvalues and the eigenvectors are given by multiplication with the inverse of $\mathbf{Y}$, see Lemma 5.5 below.

Lemma 5.5. *Let* $\mathbf{Y}, \mathbf{Z} : H \to H$ *be linear operators on a Hilbert space* H*. Additionally we assume that* $\mathbf{Y}$ *is invertible and denote the point spectrum of an operator by* $\sigma(\mathbf{Y})$*.*

1. *Then* $\sigma(\mathbf{YZ}) = \sigma(\mathbf{ZY})$*.*
2. *If* $\boldsymbol{v} \in H$ *is an eigenvector of* $\mathbf{YZ}$*, then* $\mathbf{Y}^{-1}\boldsymbol{v}$ *is an eigenvector of* $\mathbf{ZY}$*.*

Proof. We prove both assertions simultaneously. Let $\boldsymbol{v}$ be an eigenvector of $\mathbf{YZ}$ with eigenvalue λ. Then we get:

$$\mathbf{ZY}(\mathbf{Y}^{-1}\boldsymbol{v}) = (\mathbf{Y}^{-1}\mathbf{Y})\mathbf{Z}(\mathbf{Y}^{-1}\mathbf{Y})\boldsymbol{v} = \mathbf{Y}^{-1}\mathbf{YZ}\boldsymbol{v} = \mathbf{Y}^{-1}\lambda\boldsymbol{v} = \lambda\mathbf{Y}^{-1}\boldsymbol{v}.$$ □

Since $\hat{\mathbf{L}}$ is symmetric positive semi-definite, the singular value decomposition is given by the principal axis transformation. Thus we get the singular value decomposition of $\mathbf{L}$ also. We now consider the mapping $\mathbf{L} : \mathbb{U} \to \mathbb{U}$ and aim to define an inverse mapping $\mathbf{L}^{-1} : \mathbb{U} \to \mathbb{U}$.

Let $\lambda_1 \geq \ldots \geq \lambda_{n-1} > \lambda_n = 0$ be the eigenvalues of $\mathbf{L}$ and $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1}, \overline{\boldsymbol{e}})$ an orthonormal system of eigenvectors, where $\overline{\boldsymbol{e}} := \boldsymbol{e}/\|\boldsymbol{e}\|$. Then the pseudo inverse

$$\begin{aligned}\mathbf{L}^{\dagger} :=& \mathbf{V}\Sigma\mathbf{V}^t = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1}, \boldsymbol{e}) \begin{pmatrix} \lambda_1^{-1} & & & & \\ & \lambda_2^{-1} & & & \\ & & \ddots & & \\ & & & \lambda_{n-1}^{-1} & \\ & & & & 0 \end{pmatrix} (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1}, \boldsymbol{e})^t \\ =& \sum_{i=1}^{n-1} \lambda_i^{-1} \boldsymbol{v}_i \boldsymbol{v}_i^t\end{aligned}$$

is a representation of the inverse mapping $\mathbf{L}^{-1}$. Note that for arbitrary right hand side $\boldsymbol{b}$ the representation $\mathbf{L}^{\dagger}\boldsymbol{b} = \sum_{i=1}^{n-1} \beta_i \lambda_i^{-1} \boldsymbol{v}_i \in \mathbb{U}$ holds, where $\beta_i := \boldsymbol{v}_i^t \boldsymbol{b}$. Thus the eigenvectors of $\mathbf{L}^{\dagger}$ are given by $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1}, \overline{\boldsymbol{e}})$ with eigenvalues $\lambda_1^{-1}, \ldots, \lambda_{n-1}^{-1}$ and 0. There is a variety of possibilities to solve this problem numerically and a good overview is given by Bochev and Lehoucq [BL05]. To apply a direct method successfully without modifications to a semi-definite system the solver must be able to handle zero pivots. Since UMFPack has zero pivot strategies implemented we can use it as solver for this problem, compare Section 4.2. The returned solution $\boldsymbol{y}$ may still contain some parts which lie in the kernel. To ensure a well defined solution we will remove these by means of the projection

$$\boldsymbol{v} = \Pi_{\Omega}\boldsymbol{y} := \boldsymbol{y} - \frac{\boldsymbol{e} \cdot_m \boldsymbol{y}}{\boldsymbol{e} \cdot_m \boldsymbol{e}}\boldsymbol{e}. \tag{5.17}$$

Hence $\boldsymbol{v} \in \operatorname{span}\{\boldsymbol{e}\}^{\perp_m} = \mathbb{U}$. In the same way it is possible to apply a conjugate gradient method as well as a multigrid solver which works fine even for positive semi-definite systems if the right hand side and the initial guess fulfill the constraint and are orthogonal to the kernel of the system matrix with respect to the scalar product, compare Remark 5.3.

5.2 Schur complement solver (SC)

To solve the system of equations (5.12) we formulate the Schur complement in the variable $\boldsymbol{z_I}$. Omitting the subscript k the first row is transformed to

$$\boldsymbol{v} = \frac{1}{\tau}\mathbf{L}^{-1}(\boldsymbol{g} - \mathbf{E}\boldsymbol{z_I}) = \frac{1}{\tau}\mathbf{L}^{-1}\boldsymbol{g} - \frac{1}{\tau}\mathbf{L}^{-1}\mathbf{E}\boldsymbol{z_I},$$

where $\mathbf{L}^{-1}$ denotes the inverse operator of $\mathbf{L}$ with respect to the 0-mass constraint, see Section 5.1.3. It is essential here that $\boldsymbol{g}$ as well as $\boldsymbol{z_I}$ don't posses any mass. Otherwise the second transformation would not be well defined.
Using the above for $\boldsymbol{v}$ and reordering the terms in the second row of (5.12) yields

$$\left(\frac{1}{\tau}\mathbf{E}^t\mathbf{L}^{-1}\mathbf{E} + \varepsilon\gamma\mathbf{C}\right)\boldsymbol{z_I} = \frac{1}{\tau}\mathbf{E}^t\mathbf{L}^{-1}\boldsymbol{g} - \boldsymbol{h} - (\overline{w} - \overline{m}\frac{\Theta_\psi\gamma}{\varepsilon})\boldsymbol{e_I}. \tag{5.18}$$

The preconditioning by Bänsch, Morin and Nochetto, see [BMN10], is based on the idea that both parts of the operator, i.e. $\frac{1}{\tau}\mathbf{E}^t\mathbf{L}^{-1}\mathbf{E}$ and $\varepsilon\gamma\mathbf{C}$, are similar. This similarity is expressed by a spectral condition, compare Section 5.3. The efficiency of the preconditioning depends on this relation between them. Since $\mathbf{L}$ and $\mathbf{C}$ are independent of ε, γ and τ, at least for $\Theta_\psi = 0$, we multiply equation (5.18) by $\sqrt{\frac{\tau}{\varepsilon\gamma}}$, resulting in the Schur complement equation

$$\underbrace{\left(\frac{1}{\sqrt{\tau\varepsilon\gamma}}\mathbf{E}^t\mathbf{L}^{-1}\mathbf{E} + \sqrt{\tau\varepsilon\gamma}\mathbf{C}\right)}_{=:\mathbf{F}}\boldsymbol{z_I} = \underbrace{\frac{1}{\sqrt{\tau\varepsilon\gamma}}\mathbf{E}^t\mathbf{L}^{-1}\boldsymbol{g} - \sqrt{\frac{\tau}{\varepsilon\gamma}}\boldsymbol{h} - \sqrt{\frac{\tau}{\varepsilon\gamma}}(\overline{w} - \overline{m}\frac{\Theta_\psi\gamma}{\varepsilon})\boldsymbol{e_I}.}_{=:\boldsymbol{f}} \tag{5.19}$$

Recalling the previously defined subspace $\mathbb{U} = \left\{\boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{J}|} \mid \boldsymbol{x} \cdot_m \boldsymbol{e} = 0\right\}$ and the construction of $\boldsymbol{z_I}$, we want to stress that (5.19) has to be solved with regards to the 0-mass constraint on the inactive set for $\boldsymbol{z_I}$, i.e. $\boldsymbol{z_I} \cdot \mathbf{M}_{\boldsymbol{I}}\boldsymbol{e_I} = 0$.
The operator $\mathbf{F}$, as in (5.19), is symmetric, with respect to the $\cdot_m$ scalar product, and positive definite and can as such be solved by a CG method. We proof this assertion after introducing some notation, see Lemma 5.11. Analogously to the discussion of the Laplacian on the whole space, we define the space $\mathbb{H} := \{\boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{I}|} \mid \mathbf{E}\boldsymbol{x} \cdot_m \boldsymbol{e} = 0\}$. The CG iteration will be carried out on this Hilbert space. Therefore we apply a projected CG method and define the orthogonal projection

$$\Pi : \mathbb{R}^{|\boldsymbol{I}|} \to \mathbb{H}, \quad \boldsymbol{y} \mapsto \boldsymbol{y} - \frac{\boldsymbol{y}^t\mathbf{M}_{\boldsymbol{I}}\boldsymbol{e_I}}{\boldsymbol{e_I}^t\mathbf{M}_{\boldsymbol{I}}\boldsymbol{e_I}}\boldsymbol{e_I}. \tag{5.20}$$

Lemma 5.6. *The above projection given by* (5.20) *is the algebraic equivalent of the mapping* $y \mapsto y - \frac{(y,1_{\mathcal{I}})_h}{(1_{\mathcal{I}},1_{\mathcal{I}})_h}1_{\mathcal{I}}$. *This projection has the following properties:*

1. $\Pi^* = \Pi$, *i.e.* $\boldsymbol{y} \cdot_m \Pi\boldsymbol{z} = \Pi\boldsymbol{y} \cdot_m \boldsymbol{z}$ *for all* $\boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^{|\boldsymbol{I}|}$.

2. $\Pi\boldsymbol{y} = \boldsymbol{y}$ *for all* $\boldsymbol{y} \in \mathbb{H}$.

3. $\Pi \boldsymbol{e_I} = 0$.

Proof. The mapping Π is a orthogonal projection with respect to the $\cdot_m$ product and thus has these properties. We can verify them in the following way.

1. Rewriting the mapping in matrix form, we get

$$\Pi(\boldsymbol{y}) = \left(\mathbf{Id}_{\boldsymbol{I}} - \frac{\boldsymbol{e_I}\boldsymbol{e_I}^t\mathbf{M}_{\boldsymbol{I}}}{\boldsymbol{e_I}^t\mathbf{M}_{\boldsymbol{I}}\boldsymbol{e_I}} \right) \boldsymbol{y},$$

   which is a symmetric matrix with respect to the $\cdot_m$ inner product.

2. Let $\boldsymbol{y} \in \mathbb{H}$ be given. Then the assertion follows from $\boldsymbol{y}^t\mathbf{M}_{\boldsymbol{I}}\boldsymbol{e_I} = \mathbf{E}\boldsymbol{y} \cdot_m \boldsymbol{e} = 0$.

3. Simple calculation yields

$$\Pi \boldsymbol{e_I} = \boldsymbol{e_I} - 1\boldsymbol{e_I} = 0.$$

□

The second property is important to enable the incorporation of the projection in front of the variable $\boldsymbol{z_I}$, transforming $\mathbf{F}\boldsymbol{z_I} = \boldsymbol{f}$ to

$$\Pi\mathbf{F}\Pi\boldsymbol{z_I} = \Pi\boldsymbol{f}. \tag{5.21}$$

We want to remark that the term containing $\overline{w}$ and $\overline{m}$ is canceled out by the projection. To derive the Algorithm we can treat the projection as a kind of preconditioning and apply the algorithm for the preconditioned CG method, see e.g. Greenbaum [Gre97] or Meister [Mei05]. Setting $\mathbf{P} = \Pi^*\Pi = \Pi$ we obtain the following iteration in each step.

Algorithm 5.3 *Projected conjugate gradient method* (PCG)

```
1. Choose $\boldsymbol{z}_0 \in \mathbb{H}$ and set $\boldsymbol{r}_0 := \boldsymbol{f} - \mathbf{F}\boldsymbol{z}_0$.
2. $\boldsymbol{d}_0 := \mathbf{P}\boldsymbol{r}_0$, $\alpha_0 := (\boldsymbol{r}_0, \boldsymbol{d}_0)_M$.
3. For $m = 0, \dots, n-1$:
       If $\alpha_m <$ tolerance $\to$ STOP.
       $\boldsymbol{v}_m := \mathbf{F}\boldsymbol{d}_m$, $\beta_m := \frac{\alpha_m}{(\boldsymbol{v}_m, \boldsymbol{d}_m)_M}$.
       $\boldsymbol{z}_{m+1} := \boldsymbol{z}_m + \beta_m\boldsymbol{d}_m$.
       $\boldsymbol{r}_{m+1} := \boldsymbol{r}_m - \beta_m\boldsymbol{v}_m$.
       If $\|\boldsymbol{r}_{m+1}\| <$ tolerance $\to$ STOP.
       $\boldsymbol{p}_{m+1} := \mathbf{P}\boldsymbol{r}_{m+1}$, $\alpha_{m+1} := (\boldsymbol{r}_{m+1}, \boldsymbol{p}_{m+1})_M$.
       $\boldsymbol{d}_{m+1} := \boldsymbol{p}_{m+1} + \frac{\alpha_{m+1}}{\alpha_m}\boldsymbol{d}_m$.
4. Return the $\boldsymbol{z}_{m+1}$.
```

Remark 5.7. *After obtaining the solution $\boldsymbol{z_I}$ we use Lemma 5.1 and compute*

$$\overline{w} = \frac{\boldsymbol{e}_{\boldsymbol{I}}^t \left(\mathbf{M}_{\boldsymbol{I}}(\boldsymbol{v}_{\boldsymbol{I}} + \frac{(1-\Theta_\psi)\gamma}{\varepsilon} \boldsymbol{u}_{\boldsymbol{I}}^{n-1}) - \varepsilon\gamma(\mathbf{S}_{\boldsymbol{IJ}}\boldsymbol{u}) \right)}{\boldsymbol{e}_{\boldsymbol{I}}^t \mathbf{M}_{\boldsymbol{I}} \boldsymbol{e}_{\boldsymbol{I}}} + \frac{\Theta_\psi \gamma \overline{m}}{\varepsilon}.$$

As a final step we compute a solution to $\tau\mathbf{L}\boldsymbol{v} = \mathbf{M}\left(\boldsymbol{g} - \mathbf{E}\boldsymbol{z_I}\right)$. With $\boldsymbol{z_I}$, $\boldsymbol{v}$ and $\overline{w}$ we solved the saddle point system (3.26)*.*

The above formulation made use of the symmetry of the discrete Laplacian with respect to the M-inner product. It is again easily transferable to a symmetric system with respect to the l_2-product as before. Multiplying (5.19) with $\mathbf{M}_{\boldsymbol{I}}^{\frac{1}{2}}$ as well as adding the factor $\mathbf{M}_{\boldsymbol{I}}^{\frac{1}{2}}$ to the variable $\boldsymbol{z_I}$ the Schur complement would then be given by

$$\left(\frac{1}{\sqrt{\tau\varepsilon\gamma}} \mathbf{E}^t (\mathbf{M}^{-\frac{1}{2}} \mathbf{S} \mathbf{M}^{-\frac{1}{2}})^{-1} \mathbf{E} + \sqrt{\tau\varepsilon\gamma} \mathbf{M}_{\boldsymbol{I}}^{-\frac{1}{2}} \mathbf{S}_{\boldsymbol{I}} \mathbf{M}_{\boldsymbol{I}}^{-\frac{1}{2}} - \frac{\Theta_\psi \sqrt{\tau\varepsilon\gamma}}{\varepsilon^2} \mathbf{Id}_{\boldsymbol{I}_k} \right) (\mathbf{M}_{\boldsymbol{I}}^{\frac{1}{2}} \boldsymbol{z_I}) = \tilde{\boldsymbol{f}}.$$

Note that the variables are again scaled with the square root of the mass matrix. Note that the subspace $\mathbb{H}$ has to be modified together with the projection Π. Additionally we would like to point out that the resulting conjugate gradient method with the l_2 product gives the same iterates than the Algorithm 5.3. This becomes apparent, when the mass matrix $\mathbf{M}$ used in the scalar product is also split by means of its root and incorporated into the variables.
Furthermore the associated preconditioning, which we derive in the next section, is essentially the same as for (5.19), again with the exception of an additional factor $\mathbf{M}_{\boldsymbol{I}}^{-\frac{1}{2}}$, which appears on the left as well as on the right side. Overall this results in a symmetric preconditioning with respect to the l_2 product.

5.3 Preconditioning of the Schur complement system (PrecSC)

Let us recall the system (5.19) we have to solve, which is given by

$$\mathbf{F} = \frac{1}{\sqrt{\tau\gamma\varepsilon}} \mathbf{E}^t \mathbf{L}^{-1} \mathbf{E} + \sqrt{\tau\gamma\varepsilon}\mathbf{C} =: \mathcal{S}^{-1} + \mathcal{T} \tag{5.22}$$

defined on $\mathbb{U}$. Using $\sigma := \sqrt{\tau\gamma\varepsilon}$, the two parts of the operator $\mathbf{F}$ are denoted by

$$\mathcal{S}^{-1} := \sigma^{-1}\mathbf{E}^t\mathbf{L}^{-1}\mathbf{E} = \sigma^{-1}\left(\mathbf{L}^{-1}\right)_{\boldsymbol{II}} \quad \text{and} \quad \mathcal{T} := \sigma\mathbf{C} = \sigma\left(\mathbf{L}_{\boldsymbol{II}} - \frac{\Theta_\psi}{\varepsilon^2}\mathbf{Id}_{\boldsymbol{II}}\right).$$

It is a well known fact that iterative solvers greatly benefit from an adequate preconditioning. A preconditioner for a saddle point system, which is alike to (3.26), is constructed by Bänsch, Morin and Nochetto, see [BMN10]. The theory there poses conditions on the spectra of the operators $\mathcal{S}$ and $\mathcal{T}$, compare Theorem 5.16.
For a better understanding of the operator $\mathcal{S}$ we require some additional notation and preliminary results. We use the representation of $\mathcal{S}^{-1}$ as a block of the inverse Laplacian. There are two essential properties of the Laplacian matrix which we need.

Lemma 5.8. *For every non-empty index set $\alpha \subsetneq \boldsymbol{J}$ the sub-matrix $\mathbf{L}_{\alpha\alpha}$ is positive definite.*

Proof. This fact follows from the observation that $\mathbf{L}_{\alpha\alpha}$ is equivalent to the Laplacian where the vertices not belonging to the index set α are Dirichlet boundary nodes and as such is positive definite. □

Furthermore the Schur complement of Hermitian matrices has some definiteness property inherited from the original matrix. In case of the Laplacian the following is true:

Lemma 5.9. *For each index set $\emptyset \neq \alpha \subsetneq \boldsymbol{J}$ and $\beta := \boldsymbol{J} \setminus \alpha$ the Schur complement*

$$\mathbf{L}/\mathbf{L}_{\alpha\alpha} := \mathbf{L}_{\beta\beta} - \mathbf{L}_{\beta\alpha} \left(\mathbf{L}_{\alpha\alpha}\right)^{-1} \mathbf{L}_{\alpha\beta}$$

is positive semi-definite.

Proof. This assertion follows directly from the previous Lemma and Theorem 1.12, see Zhang [Zha05]. □

The inverse of a block partitioned matrix omits a distinct structure similar to the solution formula of 2×2 matrices. To this end let $X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}$ be a partitioned regular matrix. Under the assumption that X, X_{11} and X_{22} are regular we can give an explicit formula for the inverse, namely

$$X^{-1} = \begin{pmatrix} (X/X_{11})^{-1} & -X_{11}X_{12}(X/X_{11})^{-1} \\ -X_{22}^{-1}X_{21}(X/X_{22})^{-1} & (X/X_{22})^{-1} \end{pmatrix},$$

where X/X_{11} and X/X_{22} denote the Schur complements analogously to the definition in Lemma 5.9. Similar formulae have been derived for various types of generalized inverses, see e.g. Ben-Israel and Greville [BIG04], Rhode [Rho65] or Groß [Gro00]. Redefining the Schur complement by $X/X_{22} = X_{11} - X_{12}X_{22}^{\dagger}X_{21}$, where we extend the definition by using the pseudo inverse, the pseudo inverse of the discrete Laplacian can be written as

$$\mathbf{L}^{\dagger} = \begin{pmatrix} \mathbf{L}_{\boldsymbol{AA}}^{\dagger} + \mathbf{L}_{\boldsymbol{AA}}^{\dagger}\mathbf{L}_{\boldsymbol{AI}}(\mathbf{L}/\mathbf{L}_{\boldsymbol{AA}})^{\dagger}\mathbf{L}_{\boldsymbol{IA}}\mathbf{L}_{\boldsymbol{AA}}^{\dagger} & -\mathbf{L}_{\boldsymbol{AA}}^{\dagger}\mathbf{L}_{\boldsymbol{AI}}(\mathbf{L}/\mathbf{L}_{\boldsymbol{AA}})^{\dagger} \\ -(\mathbf{L}/\mathbf{L}_{\boldsymbol{AA}})^{\dagger}\mathbf{L}_{\boldsymbol{IA}}\mathbf{L}_{\boldsymbol{AA}}^{\dagger} & (\mathbf{L}/\mathbf{L}_{\boldsymbol{AA}})^{\dagger} \end{pmatrix}. \tag{5.23}$$

Hence we can associate $\mathcal{S}$ with a Schur complement quotient of the Laplacian and obtain

$$\mathcal{S} = \sigma\mathbf{L}/\mathbf{L}_{\boldsymbol{AA}} = \sigma\left(\mathbf{L}_{\boldsymbol{II}} - \mathbf{L}_{\boldsymbol{IA}}\mathbf{L}_{\boldsymbol{AA}}^{-1}\mathbf{L}_{\boldsymbol{AI}}\right). \tag{5.24}$$

Scroggs and Odell, see [SO66], constructed the pseudo inverse of a matrix via the Jordan canonical form, retaining the spectral property that if μ is an eigenvalue corresponding to the eigenvector v, then μ^{-1} is an eigenvalue of the pseudo inverse corresponding to v. Their definition and the classical definition by Moore, see [Moo20], and Penrose, see [Pen55], coincide for symmetric matrices. Thus the eigenvalues of $\mathcal{S}$ are given by the modified Laplacian given in (5.24).

At this point a useful link between $\mathcal{S}$ and $\mathcal{T}$ is presented in the following lemma.

Lemma 5.10. *Let $\boldsymbol{I} \neq \emptyset$, $\Theta_\psi = 0$, $\mathcal{S}$ and $\mathcal{T}$ as above. Then*

$$0 \leq (\mathcal{S}w, w) \leq (\mathcal{T}w, w) \quad \forall w \in \mathbb{R}^{|\boldsymbol{I}|}.$$

Proof. If $\boldsymbol{A} = \emptyset$, the assertion is trivial, since $\mathcal{S} = \mathcal{T} = \sigma\mathbf{L}$. Let $w \in \mathbb{R}^{|\boldsymbol{I}|}$ arbitrary. By means of the representation (5.24) and the symmetry of $\mathbf{L}$ we obtain

$$\begin{aligned}(\mathcal{S}w, w) &= \sigma(\mathbf{L}_{\boldsymbol{II}}w, w) - \sigma(\mathbf{L}_{\boldsymbol{IA}}(\mathbf{L}_{\boldsymbol{AA}})^{-1}\mathbf{L}_{\boldsymbol{AI}}w, w)\\ &= (\mathcal{T}w, w) - \sigma((\mathbf{L}_{\boldsymbol{AA}})^{-1}\mathbf{L}_{\boldsymbol{AI}}w, \mathbf{L}_{\boldsymbol{AI}}w) \leq (\mathcal{T}w, w),\end{aligned}$$

where we used that $\mathbf{L}_{\boldsymbol{AA}}$ is positive definite, see Lemma 5.8. Due to Lemma 5.9 the operator $\mathcal{S}$ is positive semi-definite and thus the assertion follows. □

The use of the semi-implicit discretization of the free energy, i.e. setting $\Theta_\psi = 1$, allows for no such general estimate. The additional term, can also be understood as a spectral shift applied to $\mathbf{L}^{-1}$.

Lemma 5.11. *Let $\boldsymbol{I} \neq \emptyset$ and $\Theta_\psi = 0$. The system* $\mathbf{F}$*, given in* (5.19)*, is positive definite on* $\mathbb{U}$*.*

Proof. We show that $\mathbf{F}$ is positive definite due to the fact that it is composed of a positive definite and positive semi-definite part.
As discussed before the first part of the operator, i.e. $\mathbf{E}^t\mathbf{L}^{-1}\mathbf{E}$, has a one dimensional kernel spanned by the constant vector. In Section 5.1.3 we used the restriction unto $\mathbb{U}$ to define the inverse Laplacian. Further restricting the positive definite operator $\mathbf{L}^{-1}$ by means of the extension $\mathbf{E}$ and associated restriction $\mathbf{E}^t$ the remainder is also a positive definite operator. The second part, given by $\mathbf{C}$, is obviously positive definite in case of an explicit discretization of the free energy term, i.e. $\Theta_\psi = 0$, due to Lemma 5.8. □

In case of an implicit discretization, i.e. $\Theta_\psi = 1$, we require the additional condition on the smallest eigenvalue $\lambda_{min}(\mathbf{L}_{\boldsymbol{II}}) > \frac{1}{\varepsilon^2}$ to ensure that $\mathcal{T}$ remains positive semi-definite. This can also be understood as a restriction on the shape of the inactive set. In the following we give a short motivation on how this restriction can be seen.
Considering one spatial dimension we calculate the occurring eigenvalues of the restricted Laplacian below, see Section 5.4. When we introduced the adaptive mesh in Section 3.6.1, we stated that a minimum number of eight vertices across the interface is required to avoid any mesh anisotropy effects. Since the diameter of the interface is given by $\varepsilon\pi$, we get $\pi\varepsilon = 9h$, where h is the mesh width. Denoting the size of the largest inactive set in vertices by n, we get the following approximation by means of the discontinued sum representation of cos:

$$\lambda_{min}(\mathbf{L}_{\boldsymbol{II}}) = \frac{2}{h^2}\left(1 - \cos\left(\frac{\pi}{n+1}\right)\right) \approx \frac{2}{h^2}\frac{\pi^2}{2(n+1)^2} = \frac{\pi^2}{h^2(n+1)^2} = \frac{81}{(n+1)^2}\frac{1}{\varepsilon^2}.$$

If the largest inactive set only includes eight vertices, i.e. $n \leq 8$, we have the desired property. Additionally the final iteration uses the system with the projection onto the space with mean value zero, further improving the above estimate.

5.3.1 Eigenvalue estimates for the Laplacian

Zhang discusses various eigenvalue estimates for Schur complements, see [Zha05]. In this section the results required later will be given. We denote the eigenvalues of a matrix by $\lambda_i(\mathbf{L})$ and assume the ordering $\lambda_1(\mathbf{L}) \geq \lambda_2(\mathbf{L}) \geq \ldots \geq \lambda_{n-1}(\mathbf{L}) > \lambda_n(\mathbf{L}) = 0$. Recalling the construction of $\mathbf{L}^{-1}$ via $\mathbf{L}^{\dagger}$, we get

$$\lambda_j(\mathbf{L}^{-1}) = \lambda_j(\mathbf{L}^{\dagger}) = \lambda_{n-j}(\mathbf{L})^{-1} \quad \text{for } j = 1, \ldots, n-1 \tag{5.25}$$

together with $\lambda_n(\mathbf{L}^{-1}) = \lambda_n(\mathbf{L}) = 0$.

Theorem 5.12. *For every non-empty index set $\boldsymbol{I} \subsetneq \boldsymbol{J} = \{1, \ldots, n\}$ the eigenvalues of the sub-matrix $\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}$ satisfy*

$$\lambda_i(\mathbf{L}) \geq \lambda_i(\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}) \geq \lambda_{i+n-|\boldsymbol{I}|}(\mathbf{L}) \quad \textit{for } i = 1, 2, \ldots, |\boldsymbol{I}|.$$

Proof. This Theorem by Zhang, see [Zha05], is an application of the Cauchy eigenvalue interlacing theorem. A proof is given by Lancaster and Tismentsky, see [LT85]. □

Transferring these results to the inverse of the Laplacian, we get an interleaving property which will be useful later.

Corollary 5.13. *For every non-empty index set $\boldsymbol{I} \subsetneq \boldsymbol{J}$ the eigenvalues of the inverse of the sub-matrix $\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}$ satisfy*

$$\lambda_{i-1}(\mathbf{L}^{-1}) \geq \lambda_i(\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}^{-1}) \geq \lambda_{i+n-1-|\boldsymbol{I}|}(\mathbf{L}^{-1}) \quad \textit{for } i = 2, \ldots, |\boldsymbol{I}|$$

as well as

$$\lambda_1(\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}^{-1}) \geq \lambda_{n-|\boldsymbol{I}|}(\mathbf{L}^{-1}).$$

Proof. Due to Lemma 5.8 the operator $\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}$ is positive definite

$$\lambda_j(\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}) = \big(\lambda_{|\boldsymbol{I}|-j+1}(\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}^{-1})\big)^{-1} \quad \text{holds for } j = 1, \ldots, |\boldsymbol{I}|. \tag{5.26}$$

Since $|\boldsymbol{I}| < n$, we obtain from (5.25) that

$$\lambda_j(\mathbf{L}) = \big(\lambda_{n-j}(\mathbf{L}^{-1})\big)^{-1} \quad \text{for } j = 1, \ldots, n-1.$$

Using these identities together with Theorem 5.12 we get

$$\big(\lambda_{n-j}(\mathbf{L}^{-1})\big)^{-1} = \lambda_j(\mathbf{L}) \geq \lambda_j(\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}) = \big(\lambda_{|\boldsymbol{I}|-j+1}(\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}^{-1})\big)^{-1} \quad \text{for } j = 1, \ldots, |\boldsymbol{I}|.$$

Inverting this inequality and applying the substitution $i = |\boldsymbol{I}| - j + 1$ we obtain

$$\lambda_i(\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}^{-1}) \geq \lambda_{i+n-|\boldsymbol{I}|-1}(\mathbf{L}^{-1}) \quad \text{for } i = 1, 2, \ldots, |\boldsymbol{I}|.$$

The second estimate is derived analogously by using

$$\big(\lambda_{|\boldsymbol{I}|-j+1}(\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}^{-1})\big)^{-1} = \lambda_j(\mathbf{L}_{\boldsymbol{I}\boldsymbol{I}}) \geq \lambda_{j+n-|\boldsymbol{I}|}(\mathbf{L}) = \big(\lambda_{n-(j+n-|\boldsymbol{I}|)}(\mathbf{L}^{-1})\big)^{-1}$$

for $j = 1, \ldots, |\boldsymbol{I}| - 1$. The assertion then follows again with $i = |\boldsymbol{I}| - j + 1$. □

We use a similar interlacing theorem concerning pseudo inverse of hermitian matrices and their minors for the estimation of the eigenvalues of $\mathcal{S}$. Note that the fact that we deal with a pseudo inverse becomes helpful since a similar theorem regarding just the Schur complement of a hermitian matrix does not hold, see Zhang [Zha05] for a counter example.

Theorem 5.14. *Let* $\mathbf{H}$ *be an hermitian square matrix of size* n *and* $\mathbf{A}$ *be an arbitrary square sub-matrix of* $\mathbf{H}$ *of size* k. *Using the Moore–Penrose pseudo inverse the eigenvalues satisfy*

$$\lambda_i(\mathbf{H}^\dagger) \geq \lambda_i\left((\mathbf{H}/\mathbf{A})^\dagger\right) \geq \lambda_{i+k}(\mathbf{H}^\dagger) \quad i = 1, 2, \ldots, n-k.$$

Proof. See Theorem 2.2 in [Zha05]. □

Thus setting $\mathbf{H} = \hat{\mathbf{L}}$ and $\mathbf{A} = \hat{\mathbf{L}}_{\boldsymbol{AA}}$ and using the afore mentioned relationship between eigenvalues of a matrix and its pseudo inverse discussed by Scroggs and Odell, see [SO66], we obtain an interleaving property for $\hat{\mathbf{L}}$. Since the eigenvalues are exactly the same as those of $\mathbf{L}$, compare Lemma 5.5, we simultaneously obtain the interleaving property

$$\lambda_i(\mathbf{L}^\dagger) \geq \lambda_i\left(\mathcal{S}^\dagger\right) \geq \lambda_{i+n-|\boldsymbol{I}|}(\mathbf{L}^\dagger) \quad i = 1, 2, \ldots, |\boldsymbol{I}|.$$

Note that Corollary 5.13 gives a very similar property for the operator $\mathcal{T}^{-1}$ in the semi-implicit case with $\Theta_\psi = 0$.

5.3.2 Symmetric preconditioning

Bänsch, Morin and Nochetto introduced a symmetric preconditioning of problems similar to $\mathbf{F} = \mathcal{S}^{-1} + \mathcal{T}$. The preconditioner $\mathbf{P}_\mathcal{S} := (\mathbf{Id} + \mathcal{S})^2\mathcal{S}^{-1}$ is based on the fact that if both parts of $\mathbf{F}$ are roughly the same, i.e. $\mathcal{T} \approx \mathcal{S}$, we can conclude that

$$\mathbf{F} = \mathcal{S}^{-1} + \mathcal{T} = \mathcal{S}^{-1}\left(\mathbf{Id} + \mathcal{S}\mathcal{T}\right) \approx \mathcal{S}^{-1}\left(\mathbf{Id} + \mathcal{S}^2\right) \approx \mathcal{S}^{-1}\left(\mathbf{Id} + \mathcal{S}\right)^2 = \mathbf{P}_\mathcal{S}.$$

In the semi-implicit case, i.e. $\Theta_\psi = 0$, the order of the operators with respect to the mesh size Δx, the time step size τ and to the parameters γ and ε is roughly the same (note that they correspond to a Laplacian with Neumann resp. Dirichlet boundary data). In this case the symmetric preconditioner works really well and a good speed up is obtained. When we consider the implicit case $\mathcal{T}$ additionally carries a spectral shift and the estimates, as we show below, do not necessarily hold. Despite the lack of general theoretical results the method works very well and the numerical results we present in Section 6.3, seem to be mesh independent.

As before we do not consider the operators $\mathcal{S}$ and $\mathcal{T}$ with a projection already included, but instead, add the projection operator afterwards by combining it with the obtained preconditioning to one final preconditioner.

The condition of the preconditioned system is estimated with the help of the following elementary lemma stated in Lemma 3.2 [BMN10].

Lemma 5.15. *Let A, B be symmetric positive definite operators in a Hilbert space H. If there exist two positive constants C_1, C_2 such that*

$$C_1(Aw,w) \leq (Bw,w) \leq C_2(Aw,w) \quad \textit{for all } w \in H,$$

then the condition number $cond(A^{-\frac{1}{2}}BA^{-\frac{1}{2}})$ with respect to the H-norm is bounded by $\frac{C_2}{C_1}$.

Using this, the above preconditioning gives the following result by Bänsch, Morin and Nochetto, see Theorem 4.1 in [BMN10].

Theorem 5.16. *Let $\mathcal{S}$ and $\mathcal{T}$ be self-adjoint positive definite operators on $\mathbb{U}$, and $0 < \lambda \leq \Lambda$ constants such that*

$$\left(\mathcal{S}^{-1}w,w\right) + (\mathcal{S}w,w) \geq \lambda\left(\left(\mathcal{S}^{-1}w,w\right) + (\mathcal{T}w,w)\right), \tag{5.27}$$
$$\left(\mathcal{S}^{-1}w,w\right) + (\mathcal{S}w,w) \leq \Lambda\left(\left(\mathcal{S}^{-1}w,w\right) + (\mathcal{T}w,w)\right) \tag{5.28}$$

holds for all $w \in \mathbb{U}$. Then

$$cond\left(\mathcal{S}^{-\frac{1}{2}}\mathbf{F}\mathcal{S}^{-\frac{1}{2}}\right) \leq \frac{2\Lambda}{\lambda}.$$

The required inequalities, stated as assumptions in Theorem 5.16 above, hold due to the following lemma, which is a weaker result than that given by Bänsch, Morin and Nochetto. They don't use the eigenvalues of the operator, but those of the coefficient functions, which might degenerate in our setting.

Lemma 5.17. *Let $\mathcal{S}$ and $\mathcal{T}$ be as in* (5.22)*. Then $\mathcal{S}$ and $\mathcal{T}$ are self-adjoint positive definite operators on $\mathbb{U}$ and there exist constants $0 < \lambda \leq \Lambda$ such that* (5.27) *and* (5.28) *hold for all $w \in \mathbb{U}$.*

Proof. Let λ_S and Λ_S be the smallest and largest eigenvalue of $\mathcal{S}$ on $\mathbb{U}$ and λ_T and Λ_T those of $\mathcal{T}$. Then, we get:

$$\left(\mathcal{S}^{-1}w,w\right) + (\mathcal{S}w,w) \geq \frac{1}{\Lambda_S}(w,w) + \lambda_S(w,w) = \left(\frac{1}{\Lambda_S} + \lambda_S\right)(w,w) = \frac{1+\lambda_S\Lambda_S}{\Lambda_S}(w,w),$$
$$\left(\mathcal{S}^{-1}w,w\right) + (\mathcal{T}w,w) \leq \left(\frac{1}{\lambda_S} + \Lambda_T\right)(w,w) = \frac{1+\lambda_S\Lambda_T}{\lambda_S}(w,w).$$

Putting both inequalities together (5.27) follows with

$$\lambda := \frac{1+\lambda_S\Lambda_S}{1+\lambda_S\Lambda_T}\cdot\frac{\lambda_S}{\Lambda_S}.$$

Omitting the additional $\mathcal{S}^{-1}$ term, we can also choose $\lambda := \frac{\lambda_S}{\Lambda_T}$ by a similar argument. To obtain (5.28) we use Lemma 5.10 and set $\Lambda = 1$. □

Remark 5.18. *Theorem 5.16 shows that cond*$(\mathbf{P}_{\mathcal{S}}{}^{-1/2}(\mathcal{T}+\mathcal{S}^{-1})\mathbf{P}_{\mathcal{S}}{}^{-1/2}) \leq \frac{2\Lambda}{\lambda}$ *holds due to Lemma 5.17. We would like to remark that the estimate involving the largest and smallest eigenvalue are too crude to obtain a satisfying theoretical result. Using the variant with* $\Lambda = 1$*, we get*

$$2\frac{\Lambda}{\lambda} = 2\frac{\Lambda_T}{\lambda_S} = 2\frac{\Lambda_S}{\lambda_S}\frac{\Lambda_T}{\Lambda_S} \leq 2\, cond(\mathcal{S}).$$

However, recalling the definition of $\mathcal{S}$ *and* $\mathcal{T}$*, we immediately see that both carry the same factor* σ*. The parameters* ε*,* γ *as well as the time step size* τ *only influence both operators via* σ*. Thus we can expect independence of the algorithm of those quantities, compare the experiments in Section 6.3.*

An easy reformulation allows for the utilization of $\mathcal{S}^{-1}$ in place of $\mathcal{S}$ for the preconditioning, which we prefer in our case. The preconditioner we get is the same since

$$\mathbf{P}_{\mathcal{S}} = (\mathbf{Id}+\mathcal{S})^2\mathcal{S}^{-1} = (\mathbf{Id}+\mathcal{S})(\mathcal{S}^{-1}+\mathbf{Id}) = (\mathcal{S}^{-1}+\mathbf{Id})(\mathbf{Id}+\mathcal{S}) = (\mathcal{S}^{-1}+\mathbf{Id})^2\mathcal{S}.$$

The fact that the basic idea hinges on $\mathcal{S} \approx \mathcal{T}$ makes the use of $\mathcal{T}$ for the construction of the preconditioning equally sensible. For the solution of (5.22) it is more convenient, because $\mathcal{S}$ involves the solution of the Laplacian on the whole domain Ω, whereas for

$$\mathbf{P}_{\mathcal{T}} := (\mathbf{Id}+\mathcal{T})^2\mathcal{T}^{-1} = (\mathbf{Id}+\mathcal{T}^{-1})^2\mathcal{T} \tag{5.29}$$

it suffices to solve $\mathbf{L}_{\boldsymbol{II}}$, which is essentially the Laplacian on the inactive set. For situations with fully developed interfaces this is effectively one dimension smaller than the whole domain. Extending the theory of [BMN10] for this change is quite straight forward. Exchanging the places of $\mathcal{S}$ and $\mathcal{T}$ we now require the spectral equivalence of $\mathcal{S}^{-1}$ and $\mathcal{T}$. Note that both operators omit interleaving properties with respect to the inverse Laplacian on the whole domain pointing to a strong similarity. Analogously to Theorem 4.1 [BMN10] the following is true:

Theorem 5.19. *Let* $\mathcal{S}$ *and* $\mathcal{T}$ *be as above self-adjoint positive definite operators on* $\mathbb{U}$*, and* $0 < \lambda < \Lambda$ *constants such that*

$$\left(\mathcal{T}^{-1}w, w\right) + (\mathcal{T}w, w) \geq \lambda\left(\left(\mathcal{S}^{-1}w, w\right) + (\mathcal{T}w, w)\right), \tag{5.30}$$

$$\left(\mathcal{T}^{-1}w, w\right) + (\mathcal{T}w, w) \leq \Lambda\left(\left(\mathcal{S}^{-1}w, w\right) + (\mathcal{T}w, w)\right) \tag{5.31}$$

hold for all $w \in \mathbb{U}$*. Then*

$$cond\left(\mathbf{P}_{\mathcal{T}}{}^{-\frac{1}{2}}\mathbf{F}\mathbf{P}_{\mathcal{T}}{}^{-\frac{1}{2}}\right) \leq \frac{2\Lambda}{\lambda}.$$

Proof. We repeat the steps of the proof given in [BMN10] with the slight change of the used operators. In place of $\mathcal{S}$ and $\mathcal{T}$ we now need to use the inverse operators. Due to the following short computation

$$\begin{aligned}\mathbf{P}_{\mathcal{T}}{}^{-\frac{1}{2}}(\mathcal{S}^{-1}+\mathcal{T})\mathbf{P}_{\mathcal{T}}{}^{-\frac{1}{2}} =& (\mathbf{Id}+\mathcal{T}^{-1})^{-1}\mathcal{T}^{-\frac{1}{2}}(\mathcal{S}^{-1}+\mathcal{T})\mathcal{T}^{-\frac{1}{2}}(\mathbf{Id}+\mathcal{T}^{-1})^{-1}\\ &= \left[(\mathbf{Id}+\mathcal{T}^{-1})^2\right]^{-\frac{1}{2}}(\mathcal{T}^{-\frac{1}{2}}\mathcal{S}^{-1}\mathcal{T}^{-\frac{1}{2}}+\mathbf{Id})\left[(\mathbf{Id}+\mathcal{T}^{-1})^2\right]^{-\frac{1}{2}},\end{aligned}$$

the boundedness of the preconditioned system, namely $cond(\mathbf{P}_{\mathcal{T}}{}^{-\frac{1}{2}}(\mathcal{S}^{-1} + \mathcal{T})\mathbf{P}_{\mathcal{T}}{}^{-\frac{1}{2}})$ follows by proving the existence of constants such that

$$\frac{1}{2\Lambda}\left((\mathbf{Id} + \mathcal{T}^{-1})^2 v, v\right) \leq (v,v) + \left(\mathcal{T}^{-\frac{1}{2}}\mathcal{S}^{-1}\mathcal{T}^{-\frac{1}{2}}v, v\right) \leq \frac{1}{\lambda}\left((\mathbf{Id} + \mathcal{T}^{-1})^2 v, v\right)$$

holds for all $v \in \mathbb{U}$. Setting $w := T^{-\frac{1}{2}}v$ and using (5.30) we get

$$\begin{aligned}\left((\mathbf{Id} + \mathcal{T}^{-1})^2 v, v\right) =&(v,v) + 2(\mathcal{T}^{-1}v, v) + (\mathcal{T}^{-1}v, \mathcal{T}^{-1}v) \geq (v,v) + (\mathcal{T}^{-1}v, \mathcal{T}^{-1}v)\\ =&(\mathcal{T}w, w) + (\mathcal{T}^{-1}w, w) \geq \lambda\left(\left(\mathcal{S}^{-1}w, w\right) + (\mathcal{T}w, w)\right)\\ =&\lambda\left(\left(\mathbf{Id} + \mathcal{T}^{-\frac{1}{2}}\mathcal{S}^{-1}\mathcal{T}^{-\frac{1}{2}}\right)v, v\right).\end{aligned}$$

Similarly using (5.31) together with the elementary inequality $2(a,b) \leq (a,a) + (b,b)$ for arbitrary $a, b \in \mathbb{U}$, we get the final estimate

$$\begin{aligned}\left((\mathbf{Id} + \mathcal{T}^{-1})^2 v, v\right) =&(v,v) + 2(\mathcal{T}^{-1}v, v) + (\mathcal{T}^{-1}v, \mathcal{T}^{-1}v) \leq 2(v,v) + 2(\mathcal{T}^{-1}v, \mathcal{T}^{-1}v)\\ =&2\left((\mathcal{T}w, w) + (\mathcal{T}^{-1}w, w)\right) \leq 2\Lambda\left(\left(\mathcal{S}^{-1}w, w\right) + (\mathcal{T}w, w)\right)\\ =&2\Lambda\left(\left(\mathbf{Id} + \mathcal{T}^{-\frac{1}{2}}\mathcal{S}^{-1}\mathcal{T}^{-\frac{1}{2}}\right)v, v\right).\end{aligned}$$

□

To finalize the theory we require constants λ and Λ fulfilling the above conditions. One possibility to find such constants is again the use of the smallest and largest eigenvalues for $\mathcal{S}^{-1}$ and $\mathcal{T}^{-1}$ analogously to the proof of Theorem 5.16. Similar to the estimate needed for the preconditioning involving $\mathcal{S}$ earlier, this resulting estimate is not very strong.
To gain a better understanding we discuss the eigenvalues and eigenvectors of $\mathcal{S}$ and $\mathcal{T}$ in one spatial dimension, see Section 5.4 below.

5.3.3 Implementation of the symmetric preconditioning

To sum up the preconditioning given by $\mathbf{P}_{\mathcal{T}} = (\mathbf{Id} + \mathcal{T}^{-1})^2\mathcal{T}$, we give a short comment on the practical implementation. Additionally we combine it with the projection onto $\mathbb{H}$, as in (5.20), and obtain

$$\mathbf{P}_L := \mathbf{P}_{\mathcal{T}}{}^{-\frac{1}{2}}\Pi = \mathcal{T}^{-\frac{1}{2}}(\mathbf{Id} + \mathcal{T}^{-1})^{-1}\Pi \tag{5.32}$$

as the left preconditioning matrix. When deriving the preconditioned conjugate gradient method the relevant system we need is given by:

$$\mathbf{P} := \mathbf{P}_L^t \cdot \mathbf{P}_L = \Pi\mathbf{P}_{\mathcal{T}}{}^{-\frac{1}{2}}\mathbf{P}_{\mathcal{T}}{}^{-\frac{1}{2}}\Pi = \Pi(\mathbf{Id} + \mathcal{T}^{-1})^{-1}\mathcal{T}^{-1}(\mathbf{Id} + \mathcal{T}^{-1})^{-1}\Pi. \tag{5.33}$$

Hence we obtain:

$$\mathbf{P} = \Pi(\sigma\mathbf{C} + \mathbf{Id})^{-1}(\sigma\mathbf{C})(\sigma\mathbf{C} + \mathbf{Id})^{-1}\Pi. \tag{5.34}$$

To actually apply this preconditioning, i.e. calculate $\mathbf{P}\boldsymbol{y} = \boldsymbol{x}$ for given $\boldsymbol{y}$, we use the following step by step algorithm:

```
Algorithm 5.4  Symmetric Preconditioning                                  (PREC)

  1. Set $\boldsymbol{y}^{(1)} = \Pi\boldsymbol{y}$.
  2. Solve $(\sigma\mathbf{C} + \mathbf{Id}_{\boldsymbol{I}})\boldsymbol{y}^{(2)} = \boldsymbol{y}^{(1)}$.
  3. Set $\boldsymbol{y}^{(3)} = \sigma\mathbf{C}\boldsymbol{y}^{(2)}$.
  4. Solve $(\sigma\mathbf{C} + \mathbf{Id}_{\boldsymbol{I}})\boldsymbol{y}^{(4)} = \boldsymbol{y}^{(3)}$.
  5. Return $\Pi\boldsymbol{y}^{(4)}$.
```

The application of the preconditioner requires the solution of $\sigma\mathbf{C}+\mathbf{Id}$ for each iteration. There is a variety of fast solvers for this kind of equation system. In our case we used the direct solver package UMFPack in most cases. There are two reasons for this. Firstly the system doesn't change during one PDAS iteration and the once generated factorization can be reused in every iteration of the conjugate gradient method. Hence the computational effort for each iteration besides the first one, is optimal. The second reason is the fact that the system of equations defining the preconditioning is given on the interface only, which results in a comparatively small system and can be computed very fast by means of a direct solver.
For situations, where we don't want to use the direct solver, we also implemented a conjugate gradient method as solver for $\sigma\mathbf{C} + \mathbf{Id}$. This was necessary for simulations in three space dimensions with fine grids due to memory restrictions. The convergence of this conjugate gradient method is very fast and in all studied test cases up to eight iterations were sufficient.

5.4 Spectral comparison in one space dimension

We consider an equidistant mesh on $\Omega = [0, 1]$ with N vertices, i.e. with $h = \frac{1}{N-1}$ the nodes are given by $x_i = h \cdot i$ for all $i \in \{0, \ldots, N-1\}$. Using linear Finite Elements the discrete Laplacian $\mathbf{L}$ on Ω is given by

$$\mathbf{L} = \mathbf{M}^{-1}\mathbf{S} = \frac{1}{h^2}\begin{pmatrix} 2 & -2 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & 0 \\ 0 & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -2 & 2 \end{pmatrix}.$$

The primal-dual active set method now splits Ω into three subsets, where for our consideration we can recombine both active sets and hence omit the sign distinction in this section. We need to consider the spectral properties of both operator parts $\mathcal{S}$ and $\mathcal{T}$. The spectral shift added to the operator $\mathcal{T}$ if the free energy is discretized implicitly, is not essential for the derivation of the eigenvalues and eigenvectors. Thus we set $\Theta_\psi = 0$ initially. Without loss of generality we drop the scalar factor σ in both operators $\mathcal{S}$ and $\mathcal{T}$. Hence we use

$$\mathcal{S} = \mathbf{L}_{\boldsymbol{II}} - \mathbf{L}_{\boldsymbol{IA}}\mathbf{L}_{\boldsymbol{AA}}^{-1}\mathbf{L}_{\boldsymbol{AI}} \text{ and } \mathcal{T} = \mathbf{L}_{\boldsymbol{II}}.$$

Note that this representation is derived from the observation that $\mathcal{S}$ is essentially a block of the inverse Laplacian and can as such be represented by a suitable Schur complement, see (5.24). Hence it suffices to show the spectral similarity of $\mathbf{L}_{\boldsymbol{II}}$ and $\mathbf{L}_{\boldsymbol{II}} - \mathbf{L}_{\boldsymbol{IA}}\mathbf{L}_{\boldsymbol{AA}}^{-1}\mathbf{L}_{\boldsymbol{AI}}$.

Additionally we remark that $\mathcal{S}$ has to be understood as operator on a mass free space again. Later, we show that $\mathbf{L}_{\boldsymbol{II}} - \mathbf{L}_{\boldsymbol{IA}}\mathbf{L}_{\boldsymbol{AA}}^{-1}\mathbf{L}_{\boldsymbol{AI}}$ has the characteristics of a suitable Laplacian with Neumann boundary conditions and as such omits an eigenvalue of 0 for the constant eigenvector.

5.4.1 Sub-matrices of the discrete Laplacian

For any non-empty set of indices $\boldsymbol{I}$ the sub-matrix $\mathbf{L}_{\boldsymbol{II}}$ is given as a diagonal block matrix

$$\mathbf{L}_{\boldsymbol{II}} = \frac{1}{h^2}\begin{pmatrix} \overline{\mathbf{L}}_L^{(n_1)} & & & & 0 \\ & \overline{\mathbf{L}}_M^{(n_2)} & & & \\ & & \ddots & & \\ & & & \overline{\mathbf{L}}_M^{(n_{k-1})} & \\ 0 & & & & \overline{\mathbf{L}}_R^{(n_k)} \end{pmatrix},$$

where n_i denotes the block sizes consisting of connected inactive vertices. Hence the amount k of diagonal blocks is also the number of disjoint inactive sets. Note that we don't necessarily require $n_1 \neq 0$ or $n_k \neq 0$. If $n_1 = 0$ the left boundary of Ω belongs to the active set. If the right boundary is part of the active set $n_k = 0$ holds. If they are positive the inactive set touches the boundaries. The three different block types consist of the left and right border block

$$\overline{\mathbf{L}}_L^{(n)} = \begin{pmatrix} 2 & -2 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & 0 \\ 0 & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{pmatrix}, \; \overline{\mathbf{L}}_R^{(n)} = \begin{pmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & 0 \\ 0 & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -2 & 2 \end{pmatrix}$$

and the middle blocks

$$\overline{\mathbf{L}}_M^{(n)} = \begin{pmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & 0 \\ 0 & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{pmatrix}.$$

Remark 5.20. *The eigenvalues and their eigenvectors of $\mathbf{L}_{\boldsymbol{II}}$ are obviously given by the eigenvalues of each of the blocks making up the matrix. The blocks themselves are again discrete Laplacian matrices on the smaller interfacial areas. Those interior interfaces omit a Dirichlet boundary condition where they touch active sets.*

Those eigenvalues and eigenvectors can be explicitly calculated as shown in the following lemma.

Lemma 5.21. *For fixed but arbitrary $n \in \mathbb{N}$ the above small block Laplace type matrices omit the following eigenvalues and eigenvectors using homogeneous boundary conditions for $k = 1, \ldots, n$:*

$$\begin{aligned} \mu_M^{(n,k)} &= 2\left(1 - \cos(\tfrac{k}{n+1}\pi)\right), & v_M^{(n,k)} &= \left(sin(\tfrac{k\pi i}{n+1})\right)_{i=1}^n, \\ \mu_L^{(n,k)} &= 2\left(1 - \cos(\tfrac{2k-1}{2n}\pi)\right), & v_L^{(n,k)} &= \left(sin(\tfrac{(2k-1)\pi i}{2n})\right)_{i=1}^n, \\ \mu_R^{(n,k)} &= 2\left(1 - \cos(\tfrac{2k-1}{2n}\pi)\right), & v_R^{(n,k)} &= \left(sin(\tfrac{(2k-1)\pi(n+1-i)}{2n})\right)_{i=1}^n. \end{aligned}$$

Proof. The eigenvalues and eigenvectors can be derived from the continuous formulation. It suffices to simply multiply the eigenvectors with the block matrices to obtain the assertion. Note that the factor in the eigenvalues is just 2 and not $\frac{2}{h^2}$ due to the definition of the blocks. We will briefly show the calculations for $\overline{\mathbf{L}}_M^{(n)}$ and $v_M^{(n,k)}$ for fixed but arbitrary n and k. With the exception of the first and last entry, i.e. for $i = 2, \ldots, n-1$, $\left(\overline{\mathbf{L}}_M^{(n)} v_M^{(n,k)}\right)_i$ is given by:

$$\begin{aligned} -v_{M,i-1}^{(n,k)} + 2v_{M,i}^{(n,k)} - v_{M,i+1}^{(n,k)} &= -\sin\left(\tfrac{k\pi}{n+1}(i-1)\right) + 2\sin\left(\tfrac{k\pi}{n+1}i\right) - \sin\left(\tfrac{k\pi}{n+1}(i+1)\right) \\ &= 2\sin\left(\tfrac{k\pi}{n+1}i\right) - 2\sin\left(\tfrac{k\pi}{n+1}i\right)\cos\left(\tfrac{k\pi}{n+1}\right) \\ &= 2\left(1 - \cos\left(\tfrac{k\pi}{n+1}\right)\right)\sin\left(\tfrac{k\pi}{n+1}i\right) = \mu_M^{(n,k)} v_{M,i}^{(n,k)}. \end{aligned}$$

The first and last row can be calculated analogously, since the additional terms are 0. □

Hence we can explicitly calculate the smallest and largest eigenvalues of the Laplacian restricted to the inactive set.

Theorem 5.22. *Let $\emptyset \neq \boldsymbol{I} \subsetneq \boldsymbol{J}$ with $n_{\boldsymbol{I}} := |\boldsymbol{I}|$. We denote the size of the smallest consecutive block of interior elements by n_m and the largest by n_M. Without loss of generality, we assume that the only boundary vertices belong to the left bound and denote the size of this block by n_L. The smallest eigenvalue of $\mathbf{L}_{\boldsymbol{II}}$, and thus of $\mathcal{T}$, is given by*

$$\mu^{(1)}(\mathbf{L}_{\boldsymbol{II}}) = \begin{cases} \frac{2}{h^2}(1-\cos(\frac{1}{2n_L}\pi)) & \textit{if } 2n_l - n_m \geq 1 \\ \frac{2}{h^2}(1-\cos(\frac{1}{n_m+1}\pi)) & \textit{otherwise.} \end{cases} \tag{5.35}$$

Similarly we get the largest eigenvalue

$$\mu^{(n_{\boldsymbol{I}})}(\mathbf{L}_{\boldsymbol{II}}) = \begin{cases} \frac{2}{h^2}(1-\cos(\frac{2n_L-1}{2n_L}\pi)) & \textit{if } 2n_l - n_m \geq 1 \\ \frac{2}{h^2}(1-\cos(\frac{n_m}{n_m+1}\pi)) & \textit{otherwise.} \end{cases} \tag{5.36}$$

Proof. Note that the discrete eigenvalues of each block are increasing in k, i.e. $\mu^{(n,1)}$ denotes the smallest and $\mu^{(n,n)}$ the largest eigenvalue for a fixed n. Furthermore the smallest eigenvalues are descending as a function in n, i.e. $\mu^{(n+1,1)} \leq \mu^{(n,1)}$. The largest eigenvalues are ordered ascending with respect to n, i.e. $\mu^{(n,n)} \leq \mu^{(n+1,n+1)}$. Hence the smallest eigenvalue of the operator $\mathbf{L}_{\boldsymbol{II}}$ is given by $min(\mu_M^{(n_m,1)}, \mu_L^{(n_L,1)})$. Recall that the earlier calculated smallest eigenvalues of the blocks are given by

$$\begin{aligned} \mu_M^{(n_m,1)} &= 2\left(1-\cos(\tfrac{1}{n_m+1}\pi)\right), \\ \mu_L^{(n_L,1)} &= 2\left(1-\cos(\tfrac{1}{2n_L}\pi)\right). \end{aligned}$$

The smallest eigenvalue is given by

$$\begin{aligned} \mu^{(1)}(\mathbf{L}_{\boldsymbol{II}}) &= h^{-2} min(\mu_M^{(n_m,1)}, \mu_L^{(n_L,1)}) \\ &= \tfrac{2}{h^2}(1-\cos(\pi \cdot min(\tfrac{n_m}{n_m+1}, \tfrac{2n_L-1}{2n_L}))). \end{aligned} \tag{5.37}$$

Using the monotonicity of cos on $(0,\pi)$ and the estimate $\frac{1}{2n_L} \geq \frac{1}{n_m+1} \Leftrightarrow 2n_L - n_m \geq 1$, we can write the minimum in an explicit formula with two cases and obtain (5.35). With similar calculations for the largest eigenvalue we obtain

$$\begin{aligned} \mu^{(n_I)}(\mathbf{L}_{\boldsymbol{II}}) &= h^{-2} max(\mu_M^{(n_M,n_M)}, \mu_L^{(n_L,1)}) \\ &= \tfrac{2}{h^2}(1-\cos(\pi \cdot min(\tfrac{n_M}{n_M+1}, \tfrac{2n_L-1}{2n_L}))) \qquad (5.38) \\ &= \begin{cases} \frac{2}{h^2}(1-\cos(\frac{2n_L-1}{n_L}\pi)) & \text{if } 2n_L - n_M \geq 1 \\ \frac{2}{h^2}(1-\cos(\frac{n_M}{n_M+1}\pi)) & \text{otherwise,} \end{cases} \qquad (5.39) \end{aligned}$$

where we used $\frac{2n_L-1}{2n_L} \geq \frac{n_m}{n_m+1} \Leftrightarrow 2n_L - n_m \geq 1$ completing the proof. □

5.4.2 Schur complement representation of $\mathcal{S}$

Recalling the definition of $\mathcal{S}$, we can see that it is a modification applied to $\mathbf{L}_{\boldsymbol{II}}$, which involves the inverse of $\mathbf{L}_{\boldsymbol{AA}}$. Naturally $\mathbf{L}_{\boldsymbol{AA}}$ has a similar structure than $\mathbf{L}_{\boldsymbol{II}}$. To get the inverse of such a matrix, we need the inverse of each of the blocks. Using Gauss elimination we derive an explicit formula for them. Note that for these calculations it is convenient to split the discrete Laplacian into mass matrix and stiffness matrix, do the calculations on the stiffness matrix and then incorporate the diagonal mass matrix again. Below we just state the results. To verify the statement, the block and its inverse can easily be multiplied and the result is the identity. We obtain for the inverses of the boundary blocks

$$\left(\overline{\mathbf{L}}_L^{(n)}\right)^{-1} = \begin{pmatrix} \frac{n}{2} & n-1 & n-2 & \cdots & 2 & 1 \\ \frac{n-1}{2} & n-1 & n-2 & \cdots & 2 & 1 \\ \frac{n-2}{2} & n-2 & n-2 & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 1 & 2 & \cdots & \cdots & 2 & 1 \\ \frac{1}{2} & 1 & \cdots & \cdots & 1 & 1 \end{pmatrix},$$

$$\left(\overline{\mathbf{L}}_R^{(n)}\right)^{-1} = \begin{pmatrix} 1 & 1 & \cdots & \cdots & 1 & \frac{1}{2} \\ 1 & 2 & \cdots & \cdots & 2 & 1 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & n-2 & n-2 & \frac{n-2}{2} \\ 1 & 2 & \cdots & n-2 & n-1 & \frac{n-1}{2} \\ 1 & 2 & \cdots & n-2 & n-1 & \frac{n}{2} \end{pmatrix}$$

and the middle blocks

$$\left(\overline{\mathbf{L}}_M^{(n)}\right)^{-1} = \frac{1}{n+1} \begin{pmatrix} n & n-1 & n-2 & \cdots & 3 & 2 & 1 \\ n-1 & 2n-2 & 2n-4 & \cdots & 6 & 4 & 2 \\ n-2 & 2n-4 & 3n-3 & \cdots & 9 & 6 & 3 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 3 & 6 & 9 & \cdots & 3n-3 & 2n-4 & n-2 \\ 2 & 4 & 6 & \cdots & 2n-4 & 2n-2 & n-1 \\ 1 & 2 & 3 & \cdots & n-2 & n-1 & n \end{pmatrix}.$$

It can easily be seen that the matrices below are the inverse matrices of the Laplacian block type matrices by multiplication of the blocks.

The matrices of type $\mathbf{L}_{\boldsymbol{IA}}$ and $\mathbf{L}_{\boldsymbol{AI}}$ consist of negative standard euclidean basis vectors. Later on we will be interested in $\mathbf{L}_{\boldsymbol{IA}}(\mathbf{L}_{\boldsymbol{AA}})^{-1}\mathbf{L}_{\boldsymbol{AI}}$. Hence it suffices to study $-\mathbf{L}_{\boldsymbol{IA}}$ and $-\mathbf{L}_{\boldsymbol{AI}}$ respectively. In the following we need a more detailed notation for the size of the sets the mesh is split into. We denote the existing blocks with $n_{\boldsymbol{I}_1}$, $n_{\boldsymbol{A}_1}$, $n_{\boldsymbol{I}_2}$, $n_{\boldsymbol{A}_2}\dots$. Again we do not require $n_{\boldsymbol{I}_1} > 0$. Let $n_{\boldsymbol{I}} = \sum_i n_{\boldsymbol{I}_i}$ and $n_{\boldsymbol{A}} = \sum_i n_{\boldsymbol{A}_i}$. Considering now the whole discrete Laplacian we split it into blocks like

$$h^2\mathbf{L} = \begin{array}{cc} \left(\begin{array}{c|c|c|cc} \boldsymbol{II} & \boldsymbol{IA} & 0 & 0 & \cdots \\ \hline \boldsymbol{AI} & \boldsymbol{AA} & \boldsymbol{AI} & 0 & \\ \hline 0 & \boldsymbol{IA} & \boldsymbol{II} & \boldsymbol{IA} & \\ \hline 0 & 0 & \boldsymbol{AI} & \boldsymbol{AA} & \\ \vdots & & & & \ddots \end{array}\right) & \begin{array}{c} n_{\boldsymbol{I}_1} \\ n_{\boldsymbol{A}_1} \\ n_{\boldsymbol{I}_2} \\ n_{\boldsymbol{A}_2} \\ \vdots \end{array} \\ \begin{array}{ccccc} n_{\boldsymbol{I}_1} & n_{\boldsymbol{A}_1} & n_{\boldsymbol{I}_2} & n_{\boldsymbol{A}_2} & \cdots \end{array} & \end{array}$$

and thus using $e_i \in \mathbb{R}^{n_{\boldsymbol{I}}}$ we obtain the following representation

$$-h^2\mathbf{L}_{\boldsymbol{AI}} = \left(\begin{array}{cccccccccccccc} 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & & \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \\ \vdots & & \vdots & \vdots & 0 & \vdots & & \vdots & \vdots & \vdots & & \\ \vdots & & \vdots & \vdots & 1 & \vdots & & \vdots & 0 & \vdots & & \\ \vdots & & \vdots & \vdots & 0 & \vdots & & \vdots & 1 & \vdots & & \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & 0 & \vdots & & \cdots \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & 0 & & \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & 1 & & \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & 0 & & \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & 0 & & \\ \multicolumn{3}{c}{\underbrace{0 \quad \cdots \quad 0}_{n_{\boldsymbol{I}_1}}} & 0 & 0 & \multicolumn{3}{c}{\underbrace{0 \quad \cdots \quad 0}_{n_{\boldsymbol{I}_2}}} & 0 & 0 & & \end{array}\right)$$

$$= \begin{pmatrix} 0 & \ldots & 0 & e_1 & e_{n_{\boldsymbol{A}_1}} & 0 & \ldots & 0 & e_{n_{\boldsymbol{A}_1}+1} & e_{n_{\boldsymbol{A}_1}+n_{\boldsymbol{A}_2}} & \ldots \end{pmatrix}.$$

Thus building $\mathbf{L}_{\boldsymbol{IA}}X\mathbf{L}_{\boldsymbol{AI}}$ for an arbitrary matrix $X \in \mathbb{R}^{n_{\boldsymbol{A}}\times n_{\boldsymbol{A}}}$ we get

$$h^2\mathbf{L}_{\boldsymbol{IA}}X\mathbf{L}_{\boldsymbol{AI}} = \left(\begin{array}{c|cc|c|c|c} \mathbf{0}_{n_{\boldsymbol{I}_1}-1} & & & & & \\ \hline & X_{1,1} & X_{1,n_{\boldsymbol{A}_1}} & & X_{1,n_{\boldsymbol{A}_1}+n_{\boldsymbol{A}_2}} & \\ & X_{n_{\boldsymbol{A}_1},1} & X_{n_{\boldsymbol{A}_1},n_{\boldsymbol{A}_1}} & & X_{n_{\boldsymbol{A}_1},n_{\boldsymbol{A}_1}+n_{\boldsymbol{A}_2}} & \\ \hline & & & \mathbf{0}_{n_{\boldsymbol{I}_2}-2} & & \\ \hline & X_{1,n_{\boldsymbol{A}_1}+n_{\boldsymbol{A}_2}} & X_{n_{\boldsymbol{A}_1},n_{\boldsymbol{A}_1}+n_{\boldsymbol{A}_2}} & & X_{n_{\boldsymbol{A}_1}+n_{\boldsymbol{A}_2},n_{\boldsymbol{A}_1}+n_{\boldsymbol{A}_2}} & \\ & & & & & \ddots \end{array}\right),$$

where $\mathbf{0}_k$ denotes a square zero matrix of size k. When we now employ the shape of $\mathbf{L}_{\boldsymbol{AA}}$ as block diagonal matrix we find that all the offdiagonal blocks only contain zeros and hence

$$h^2\mathbf{L}_{\boldsymbol{IA}}\mathbf{L}_{\boldsymbol{AA}}^{-1}\mathbf{L}_{\boldsymbol{AI}} = \begin{pmatrix} \mathbf{0}_{n_{\boldsymbol{I}_1}-1} & & & & & \\ & \boxed{\begin{matrix} \frac{n_{\boldsymbol{A}_1}}{n_{\boldsymbol{A}_1}+1} & \frac{1}{n_{\boldsymbol{A}_1}+1} \\ \frac{1}{n_{\boldsymbol{A}_1}+1} & \frac{n_{\boldsymbol{A}_1}}{n_{\boldsymbol{A}_1}+1} \end{matrix}} & & & & \\ & & \mathbf{0}_{n_{\boldsymbol{I}_2}-2} & & & \\ & & & \boxed{\begin{matrix} \frac{n_{\boldsymbol{A}_2}}{n_{\boldsymbol{A}_2}+1} & \frac{1}{n_{\boldsymbol{A}_2}+1} \\ \frac{1}{n_{\boldsymbol{A}_2}+1} & \frac{n_{\boldsymbol{A}_2}}{n_{\boldsymbol{A}_1}+1} \end{matrix}} & & \\ & & & & \ddots & \\ & & & & & \boxed{1} \end{pmatrix} \tag{5.40}$$

Finally using the block representation of $\mathbf{L}_{\boldsymbol{AA}}$ we obtain a tridiagonal representation of $\mathcal{S}$. For example for arbitrary $n_{\boldsymbol{I}_i}$, $n_{\boldsymbol{A}_i}$, $i = 1, 2$ we get

$$\mathbf{L}_{\boldsymbol{II}} - \mathbf{L}_{\boldsymbol{IA}}\mathbf{L}_{\boldsymbol{AA}}^{-1}\mathbf{L}_{\boldsymbol{AI}} = h^{-2} \left(\begin{array}{ccccc|ccccc} 2 & -2 & & & & & & & & \\ -1 & 2 & -1 & & & & & & & \\ & \ddots & \ddots & \ddots & & & & & & \\ & & -1 & 2 & -1 & & & & & \\ & & & -1 & 1+\frac{1}{n_{\boldsymbol{A}_1}+1} & -\frac{1}{n_{\boldsymbol{A}_1}+1} & & & & \\ \hline & & & & -\frac{1}{n_{\boldsymbol{A}_1}+1} & 1+\frac{1}{n_{\boldsymbol{A}_1}+1} & -1 & & & \\ & & & & & -1 & 2 & -1 & & \\ & & & & & & \ddots & \ddots & \ddots & \\ & & & & & & & -1 & 2 & -1 \\ & & & & & & & & -1 & 1 \end{array}\right).$$

5.4.3 The Laplacian of a weighted path graph

We continue by studying a quite simple situation, where no boundary interfaces are present with three active and two inactive sets in between, i.e. $n_{\boldsymbol{I}_1} = 0$, $n_{\boldsymbol{A}_1}, n_{\boldsymbol{A}_3} > 0$ arbitrary, $n_{\boldsymbol{I}_2}$ and $n_{\boldsymbol{I}_3}$ determine the block sizes below and finally $n_{\boldsymbol{A}_2}$, describing the size of the active set in between the two interfaces, will be denoted by n resulting in

$$h^2\mathcal{S} = \left(\begin{array}{ccccc|ccccc} 1 & -1 & & & & & & & & \\ -1 & 2 & -1 & & & & & & & \\ & \ddots & \ddots & \ddots & & & & & & \\ & & -1 & 2 & -1 & & & & & \\ & & & -1 & 1+\frac{1}{n+1} & -\frac{1}{n+1} & & & & \\ \hline & & & & -\frac{1}{n+1} & 1+\frac{1}{n+1} & -1 & & & \\ & & & & & -1 & 2 & -1 & & \\ & & & & & & \ddots & \ddots & \ddots & \\ & & & & & & & -1 & 2 & -1 \\ & & & & & & & & -1 & 1 \end{array}\right). \tag{5.41}$$

If $n = 0$, i.e. if no interior active set is present, the system reduces to the stiffness matrix of a discrete Laplacian operator with Neumann boundary conditions multiplied by h.

There are three different possibilities to view the matrix given in (5.41). We start out with the graph theoretical point of view. The following brief introduction of the used terminology is kept alongside of the survey article of Mohar, see [Moh91]. In this sense a graph $G = (V, E, W)$ is a set of vertices $V = \{v_i\}$ and edges E in between. We additionally assign a weight $\omega_{ij} = \omega_{ji} \geq 0$ to the edge connecting the vertices v_i and v_j. The degree of vertex $v_i \in V$ is given by $d(v_i) = \sum_j \omega_{ij}$. The Laplacian matrix of G is then given by $Q(G) = D(G) - A(G)$, where $A(G) := (\omega_{ij})_{i,j}$ denotes the adjacency matrix and $D(G) := diag(d(v_i))$.

To derive (5.41) as Laplacian matrix of a graph, we use the given finite element triangulation, cut out the active sets and reconnect them by a weighted edge across the removed interior active sets, compare Figure 5.1. Using this graph, where almost all weights are set to 1 with the exception of the longer edge, which is set to $\frac{1}{n+1}$, we get $Q(G) = h^2\mathcal{S}$. Additionally, as an immediate result, we obtain that $Q(G)$ is positive semi definite and $\ker(Q(G)) = span(\boldsymbol{e})$, see e.g. Mohar [Moh91].

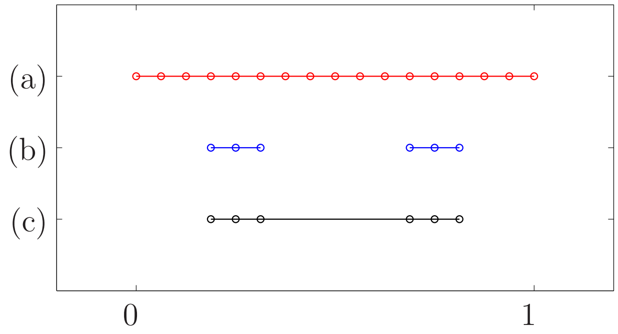


Figure 5.1: (a) Finite element mesh, (b) Inactive set, (c) Weighted graph G related to $\mathcal{S}$.

This point of view is very useful when calculating the eigenvalues and eigenvectors, especially for the case $n = 0$. Starting out from a ring graph R_m with m vertices, which essentially consists of a polygon with m edges, where the eigenvectors are given by

$$x^{(m,k)} = \left(\sin(\tfrac{2\pi ki}{m})\right)_{i=1}^{m} \text{ and } y^{(m,k)} = \left(\cos(\tfrac{2\pi ki}{m})\right)_{i=1}^{m}$$

for $1 \leq k \leq \frac{m}{2}$ and their corresponding eigenvalues $2\left(1 - \cos(\frac{2\pi k}{m})\right)$. The Laplacian of the path graph P_m omits the same eigenvalues as R_{2m}, which can be verified gluing together two copies of P_m and a splitting argument, see e.g. Spielman [Spi09].

Lemma 5.23. *The eigenvectors of* (5.41) *with* $n = 0$ *are given by*

$$z^{(N,k)} := \sin(\tfrac{\pi k}{N}) \left(x_i^{(2N,k)}\right)_{i=1}^{N} + (1 + \cos(\tfrac{\pi k}{N})) \left(y^{(2N,k)}\right)_{i=1}^{N} \qquad \textit{for } 0 \leq k \leq N-1,$$

where N denotes the sum of all inactive sets or equivalently the size of the system. The corresponding eigenvalues are given by the eigenvalues on the ring graph

$$\mu^{(N,k)} := 2\left(1 - \cos(\tfrac{\pi k}{N})\right).$$

Proof. Making the ansatz $z^{(N,k)} = a \cdot x^{(2N,k)} + b \cdot y^{(2N,k)}$ and using the condition that $z_i^{(N,k)} = z_{2N+1-i}^{(N,k)}$ has to hold for $i = 1, \dots, N$. We get

$$\begin{aligned} a \cdot x_i^{(2N,k)} + b \cdot y_i^{(2N,k)} &= a \cdot x_i^{(2N,k)} + b \cdot y_i^{(2N,k)} \\ &= a \cdot \sin(\tfrac{2\pi k}{N}(2N+1-i)) + b \cdot \cos(\tfrac{2\pi k}{N}(2N+1-i)). \end{aligned}$$

Using the periodicity of sin and cos as well as the trigonometric addition formulas we get

$$\begin{aligned} a \cdot x_i^{(2N,k)} &+ b \cdot y_i^{(2N,k)} = \\ &= a \cdot \sin(\tfrac{2\pi k}{N}(1-i)) + b \cdot \cos(\tfrac{2\pi k}{N}(1-i)) \\ &= a \left(\sin(\tfrac{2\pi k}{N}) \cos(\tfrac{2\pi k}{N} i) - \sin(\tfrac{2\pi k}{N} i) \cos(\tfrac{2\pi k}{N})\right) \\ &\quad + b \left(\cos(\tfrac{2\pi k}{N}) \sin(\tfrac{2\pi k}{N} i) - \cos(\tfrac{2\pi k}{N} i) \sin(\tfrac{2\pi k}{N})\right) \\ &= \left(b \cdot \sin(\tfrac{2\pi k}{N}) - a \cdot \cos(\tfrac{2\pi k}{N})\right) \sin(\tfrac{2\pi k}{N} i) + \left(a \cdot \sin(\tfrac{2\pi k}{N}) + b \cdot \cos(\tfrac{2\pi k}{N})\right) \cos(\tfrac{2\pi k}{N} i). \end{aligned}$$

Hence we obtain an 2×2 system for the determination of a and b. With the solution $a = \sin(\frac{2\pi k}{N})$ and $b = 1 + \cos(\frac{2\pi k}{N})$ the assertion is shown. □

In case of $n > 0$, this construction heavily depends on the size of the sets, or position of the active sets, and thus gives no general formula. The construction of the weighted graph suggests to consider the graph without the weighted edge, reconnection the inactive sets, first and then use existing results to incorporate the missing edge. This gives some control over the eigenvalues due to an interlacing theorem, which holds here, compare Theorem 3.2 [Moh91]. The basic idea to this result is the idea that the Laplacian of the new graph is given as a rank-one update of the Laplacian of the graph without the edge.

5.4.4 Weighted Laplacian on an equidistant mesh

Considering a weighted Laplacian operator on an equidistant mesh enables us to show a relation to the theory given by Bänsch, Morin and Nochetto in [BMN10]. The operator $\mathcal{S}$ can be expressed as the discretization of a weighted Laplacian, i.e.

$$\mathcal{S} = \frac{\sigma}{h} \left((a(x) \nabla \phi_j, \nabla \phi_i)\right)_{i,j \in \boldsymbol{I}},$$

where a is a piecewise constant function. For a given mesh in one dimension with vertices x_i, $i \in \{1, \dots, n\}$, we assume $a(x) \equiv a_i \in \mathbb{R}$ if $x \in [x_{i-1}, x_i)$, where a_i is the inverse of the distance to the next inactive vertex, i.e. $a \hat{=} (1 + \text{Distance in active vertices})^{-1}$, see Figure 5.2 for an example.

The main problem here is caused by the fact that a is zero, at the boundaries. Thus the positivity of the coefficient function a, which is required in [BMN10] to estimate the constants λ or Λ, does not hold here. The handling of the operator $\mathcal{T}$ is much simpler, since we can set $b \equiv 1$.

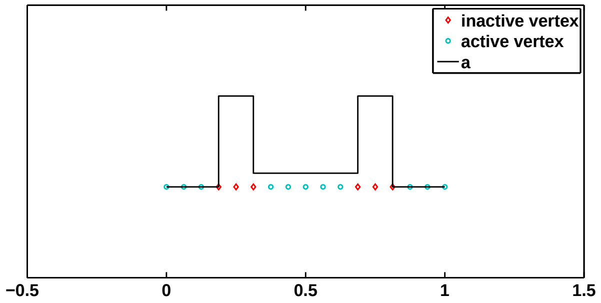


Figure 5.2: Example mesh and coefficient function associated to $\mathcal{S}$.

However, if we assume that the boundary vertices are inactive, the coefficient function is bounded such that $\frac{1}{1+n} \leq a(x) \leq 1$, where n denotes the size of the largest active set. Then Corollary 4.2 from [BMN10] shows that the condition number of the preconditioned system is bounded by $2(n+1)$. The numerical results we present in Section 6.3.5, suggest that the preconditioning works even better than suggested by this result.

5.4.5 $\mathcal{S}$ as rank-one update of the stiffness matrix

Starting out from the Laplacian matrix of the graph given by the inactive set only, we obtain the corresponding Laplacian matrix

$$Q(I) = \left(\begin{array}{ccccc|ccccc} 1 & -1 & & & & & & & & \\ -1 & 2 & -1 & & & & & & & \\ & \ddots & \ddots & \ddots & & & & & & \\ & & -1 & 2 & -1 & & & & & \\ & & & -1 & 1 & & & & & \\ \hline & & & & & 1 & -1 & & & \\ & & & & & -1 & 2 & -1 & & \\ & & & & & & \ddots & \ddots & \ddots & \\ & & & & & & & -1 & 2 & -1 \\ & & & & & & & & -1 & 1 \end{array}\right). \tag{5.42}$$

Using the results from Section 5.4.3 for $n = 0$ with the appropriate size of the system, we already know the spectrum as well as the eigenvectors of this matrix, or more precisely the spectrum and eigenvalues of its blocks. Adding a weighted edge to the graph can be expressed as a rank-one update with $u := (0, \ldots, 0, \sqrt{\frac{1}{n+1}}, -\sqrt{\frac{1}{n+1}}, 0, \ldots, 0)^t$, i.e.

the update of the matrix is given by

$$uu^t = \begin{pmatrix} \ddots & & & \\ & \frac{1}{n+1} & -\frac{1}{n+1} & \\ & -\frac{1}{n+1} & \frac{1}{n+1} & \\ & & & \ddots \end{pmatrix}.$$

After normalizing the vector we get

$$Q(I) = h^2\mathcal{S} - \frac{2}{n+1}vv^t,$$

where $v = (0,\dots,0,\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}},0,\dots,0)^t$. Note that $Q(I)$ can also be motivated as the stiffness matrix associated to the triangulation given by the reconnected inactive sets, i.e. the one depicted in Figure 5.1.(c).

Bunch, Nielsen and Sorensen extended results of Golub and stated an algorithm for the calculation of the eigensystem of rank-one updated matrices, see [BNS78].

Again, as before, we get no general result. Just calculating the eigenvalues of the updated system requires the solution of a nonlinear equation. We consider a special symmetric case, consisting of two inactive sets of the same size, similar to the situation depicted in Figure 5.1. Using inactive sets of size three and adopting the denotaions of [BNS78], we get

$$B = Q(I) = \left(\begin{array}{ccc|ccc} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 1 & & & \\ \hline & & & 1 & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{array}\right), \quad b = \frac{2}{n+1}\begin{pmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0 \end{pmatrix},$$

such that $h^2\mathcal{S} = B + \frac{2}{n+1}bb^t$. By means of Lemma 5.23, we obtain the eigenvectors and values of the two blocks. For each of the 3×3-blocks we have

$$z^{(3,1)} = \frac{1}{\sqrt{3}}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad z^{(3,2)} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \text{ and } z^{(3,3)} = \frac{1}{\sqrt{6}}\begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix},$$

together with the eigenvalues

$$\mu^{(3,1)} = 0, \mu^{(3,2)} = 1 \text{ and } \mu^{(3,3)} = 3.$$

Let

$$Q := \begin{pmatrix} & 0 & & 0 & & 0 \\ z^{(3,1)} & 0 & z^{(3,2)} & 0 & z^{(3,3)} & 0 \\ & 0 & & 0 & & 0 \\ 0 & & 0 & & 0 & \\ 0 & z^{(3,1)} & 0 & z^{(3,2)} & 0 & z^{(3,3)} \\ 0 & & 0 & & 0 & \end{pmatrix} \text{ and } z := Q^t b = \begin{pmatrix} \frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{6}} \\ -\frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{\sqrt{12}} \\ -\frac{1}{\sqrt{12}} \end{pmatrix},$$

then $B + \frac{2}{n+1}bb^t = Q(D + \frac{2}{n+1}zz^t)Q^t$, where D is a diagonal matrix, whose entries are given by the eigenvalues. Since we have duplicate eigenvalues, we can apply the deflation method and reduce the problem with the help of an elementary reflector

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}.$$

For the first and third eigenvalue we can use H as is. To handle the second eigenvalue we use H^t. After applying this reflexions and permuting the columns of Q, we get

$$\overline{Q} := \begin{pmatrix} \frac{1}{\sqrt{6}} & \frac{1}{2} & \frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{6}} & \frac{1}{2} & -\frac{1}{\sqrt{12}} \\ \frac{1}{\sqrt{6}} & 0 & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & 0 & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{2} & \frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{6}} & -\frac{1}{2} & -\frac{1}{\sqrt{12}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{2} & \frac{1}{\sqrt{12}} & \frac{1}{\sqrt{6}} & \frac{1}{2} & \frac{1}{\sqrt{12}} \\ \frac{1}{\sqrt{6}} & 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & 0 & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & \frac{1}{2} & \frac{1}{\sqrt{12}} & \frac{1}{\sqrt{6}} & -\frac{1}{2} & \frac{1}{\sqrt{12}} \end{pmatrix} \text{ and } \overline{z} := \overline{Q}^t b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} \end{pmatrix}.$$

At this point we already know three of the eigenvalues and eigenvectors given by the first three columns of $\overline{Q}$ and the original eigenvalues $\mu_S^{(n,1)} = 0$, $\mu_S^{(n,3)} = 1$ and $\mu_S^{(n,5)} = 3$, which are independent of the size of the active set n. The remaining problem is to compute the eigensystem of

$$\begin{pmatrix} 0 & & \\ & 1 & \\ & & 3 \end{pmatrix} + \frac{2}{n+1} \begin{pmatrix} -\frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} \end{pmatrix}^t. \tag{5.43}$$

The remaining eigenvalues are given as the roots of

$$f_n(\lambda) := 1 + \frac{2}{n+1} \left(\frac{\frac{1}{3}}{0-\lambda} + \frac{\frac{1}{2}}{1-\lambda} + \frac{\frac{1}{6}}{3-\lambda} \right) \tag{5.44}$$

or equivalently as solutions to the cubic equation

$$(n+1)\lambda^3 + (-4n-6)\lambda^2 + (3n+9)\lambda - 2 = 0.$$

Since we are only interested in the case with $n \geq 1$, we get three real solutions by means of a solution formula for cubic equations. We omit the general explicit fomula, since it is quite lengthy. What we obtain is that the eigenvalues are monotonically decreasing for increasing $n \geq 1$. Additionally there is an interleaving property discussed by Buch, Nielsen and Sorensen, such that we have also $\mu_S^{(n,6)} \geq 3$, $\mu_S^{(n,4)} \geq 1$ and $\mu_S^{(n,2)} \geq 0$. Thus we have basically two extreme cases given by $n = 1$ and $n = \infty$. We get

$n = 1$: $\mu_S^{(1,6)} = 3.2469$, $\mu_S^{(1,4)} = 1.5549$ and $\mu_S^{(1,2)} = 0.1980$ and the coresponding eigenvectors

$$v_S^{(1,6)} = \begin{pmatrix} -0.23 \\ 0.52 \\ -0.41 \\ 0.41 \\ -0.52 \\ 0.23 \end{pmatrix}, \; v_S^{(1,4)} = \begin{pmatrix} -0.41 \\ 0.23 \\ 0.52 \\ -0.52 \\ -0.23 \\ 0.41 \end{pmatrix} \text{ and } v_S^{(1,2)} = \begin{pmatrix} -0.52 \\ -0.41 \\ -0.23 \\ 0.23 \\ 0.41 \\ 0.52 \end{pmatrix}.$$

$n = 100$: $\mu_S^{(100,6)} = 3.0066$, $\mu_S^{(100,4)} = 1.0198$ and $\mu_S^{(100,2)} = 0.0128$ and the coresponding eigenvectors

$$v_S^{(100,6)} = \begin{pmatrix} 0.28 \\ -0.57 \\ 0.29 \\ -0.29 \\ 0.57 \\ -0.28 \end{pmatrix}, \; v_S^{(100,4)} = \begin{pmatrix} -0.49 \\ 0.00 \\ 0.50 \\ -0.50 \\ -0.00 \\ 0.49 \end{pmatrix} \text{ and } v_S^{(100,2)} = \begin{pmatrix} -0.41 \\ -0.40 \\ -0.39 \\ 0.39 \\ 0.40 \\ 0.41 \end{pmatrix}.$$

$n \to \infty$: $\mu_S^{(\infty,6)} = 3$, $\mu_S^{(\infty,4)} = 1$ and $\mu_S^{(\infty,2)} = 0$ and the coresponding eigenvectors

$$v_S^{(\infty,6)} = \begin{pmatrix} -1 \\ 2 \\ -1 \\ 1 \\ -2 \\ 1 \end{pmatrix}, \quad v_S^{(\infty,4)} = \begin{pmatrix} -1 \\ 0 \\ 1 \\ -1 \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad v_S^{(\infty,2)} = \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

Thus we have determined all eigenvalues and eigenvectors of $\mathcal{S}$. To obtain the desired constants λ and Λ for the preconditioning, we require also those of $\mathcal{T}$. They are of course independent of n and are given by $\mu_T^{(1)} = \mu_T^{(2)} = 2 - \sqrt{2}$, $\mu_T^{(3)} = \mu_T^{(4)} = 2$ and $\mu_T^{(5)} = \mu_T^{(6)} = 2 + \sqrt{2}$, see Lemma 5.21, together with

$$v_T^{(1)} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 1 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \; v_T^{(2)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 1 \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \; v_T^{(3)} = \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$v_T^{(4)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}, \; v_T^{(5)} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -1 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ and } v_T^{(6)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ -1 \\ \frac{1}{\sqrt{2}} \end{pmatrix}.$$

The extremal case, where $n = \infty$, no longer admits a constant $\lambda > 0$ such that $(\mathcal{S}w, w) \geq \lambda(\mathcal{T}w, w)$. Just setting $w = v_S^{(\infty,2)}$ gives $(\mathcal{S}w, w) = 0$, whereas $(\mathcal{T}w, w) = 4$. Using the above, we get the following representation for the eigenvectors of $\mathcal{S}$:

$$\begin{aligned} v_S^{(n,1)} &= \frac{2-\sqrt{2}}{4}(v_T^{(1)} + v_T^{(2)}) + \frac{2+\sqrt{2}}{4}(v_T^{(5)} + v_T^{(6)}), \\ v_S^{(n,3)} &= v_T^{(3)} - v_T^{(4)}, \\ v_S^{(n,5)} &= (1-\sqrt{2})(v_T^{(1)} + v_T^{(2)}) + (1+\sqrt{2})(v_T^{(5)} + v_T^{(6)}). \end{aligned}$$

However, the eigenvectors of $\mathcal{S}$, which are depending on n, require all eigenvectors of $\mathcal{T}$ in their representation, thus the best estimate we can use to this end, is $\lambda = \frac{\mu_S^{(n,6)}}{\mu_T^{(1)}}$, compare also Lemma 5.17. Similar results can be obtained for the preconditioning with $\mathcal{T}$ in place of $\mathcal{S}$.

We can improve on this estimate a bit, with the help of the $\mathcal{S}^{-1}$ term occurring in (5.27). We require the following basic estimate.

Lemma 5.24. *Let $c > 0$ be a constant. Then*

$$\left(\frac{1}{x} + x\right) \geq \lambda \left(\frac{1}{x} + c\right), \tag{5.45}$$

with $\lambda := \frac{2\left(\sqrt{1+c^2}-1\right)}{c^2}$, holds for arbitrary $x \in (0, \infty)$.

Proof. We consider the real valued function

$$f(x) := \frac{\frac{1}{x} + x}{\frac{1}{x} + c} = \frac{1 + x^2}{1 + cx}$$

and show that its minimal value is bounded by λ from below. The derivatives of f are given by:

$$\begin{aligned} f'(x) =& \frac{cx^2 + 2x - c}{(1 + cx)^2}, \\ f''(x) =& \frac{2c^2 + 2}{(1 + cx)^3}. \end{aligned}$$

We obtain two critical points of f', namely $x_{1,2} = \frac{1}{c}\left(-1 \pm \sqrt{1 + c^2}\right)$. It suffices to consider x_1 since $x_2 < 0$. Since $f''(x_1) > 0$, the minimal value of f is given at x_1 and we obtain:

$$\begin{aligned} f(x_1) =& \frac{1 + \frac{1}{c^2}\left(-1 + \sqrt{1 + c^2}\right)^2}{1 + \frac{c}{c}\left(-1 + \sqrt{1 + c^2}\right)} \\ =& \frac{2\left(\sqrt{1 + c^2} - 1\right)}{c^2}. \end{aligned}$$

□

Let v be an arbitrary eigenvector of $\mathcal{S}$ with eigenvalue μ. Furthermore we denote the largest eigenvalue of $\mathcal{T}$ by $\overline{\mu}$ such that $\mathcal{T}v \leq \overline{\mu}v$ holds. Then we obtain the following estimate due to Lemma 5.24:

$$\begin{aligned} (\mathcal{S}v, v) + (\mathcal{S}^{-1}v, v) &= \left(\mu + \frac{1}{\mu}\right)(v, v) \\ &\geq \lambda \left(\overline{\mu} + \frac{1}{\mu}\right) \\ &\geq \lambda \left((\mathcal{T}v, v) + (\mathcal{S}^{-1}v, v)\right). \end{aligned}$$

Since this holds for all eigenvectors of $\mathcal{S}$, we get (5.27). And finally obtain

$$cond(\mathbf{P}_{\mathcal{S}}{}^{-1/2}(\mathcal{T}+\mathcal{S}^{-1})\mathbf{P}_{\mathcal{S}}{}^{-1/2}) \leq \frac{2\Lambda}{\lambda} = \frac{\overline{\mu}^2}{\sqrt{1+\overline{\mu}^2}-1}, \tag{5.46}$$

where $\overline{\mu} = \frac{\sigma}{h^2}4$ due to Theorem 5.12. With $\sigma = \sqrt{\varepsilon\gamma\tau}$ and the assumption that $\varepsilon \approx \frac{9h}{\pi}$ this leads to a h independent bound if $\tau \sim h^3$.

Chapter 6

Numerical results

Finally we report on various numerical results. First we study the convergence of the phase field model with constant mobility to the corresponding sharp interface model, i.e. the Mullins–Sekerka model. Additionally we use this setting, where exact solutions are known, to study the quality of approximation of the implicit and semi-implicit time discretization scheme with respect to the time step size.

The evolution with constant mobility separates nearly homogeneous mixtures very quickly. Thus we start out with a constant mixture with a stochastic disturbance. During the initial phase all of Ω is inactive and hence one primal-dual active set iteration is sufficient. Initially, when the first active vertices start to appear the necessary amount of iterations is slightly higher than in later stages, but remained below 10 in all our experiments with sensible initial data for the active and inactive sets.

Since the initial phase of the above process is not very important for the study of the method, we use a setting where the interface consists of 4 circles. There all interesting situations appear, namely convex and concave interface sections and the appearance and disappearance of inactive and active sets. Here we compare iteration counts and runtime of our method, see Section 6.2.

Following that, we extend this configuration to three spatial dimensions and finally present a comparison of the various implemented linear Algebra solvers.

We close the presentation of the constant mobility simulations with an extensive study of the preconditioning described in the previous chapter.

In Section 6.4 we present some results for the model with non-constant diffusional mobility and compare the results of the degenerate method to the results of Barrett, Blowey and Garcke, see [BBG99], and Bänsch, Morin and Nochetto, see [BMN05].

6.1 Radially symmetric situations

Let $\Omega = B_1(0) \subset \mathbb{R}^2$ and the initial data shall describe two concentric circles around 0 with $r_1 = 0.3$ and $r_2 = 0.15$. The time evolution results in a shrinking of both radii until the smaller one vanishes at time $t_c = 1.85 \cdot 10^{-3}$. We obtain an exact solution to the Mullins–Sekerka problem, compare Section 2.4, by solving the system of ordinary differential equations (2.39)-(2.40) with a numerical method. Figure 6.1 shows the progression of the sum of the radii over time.

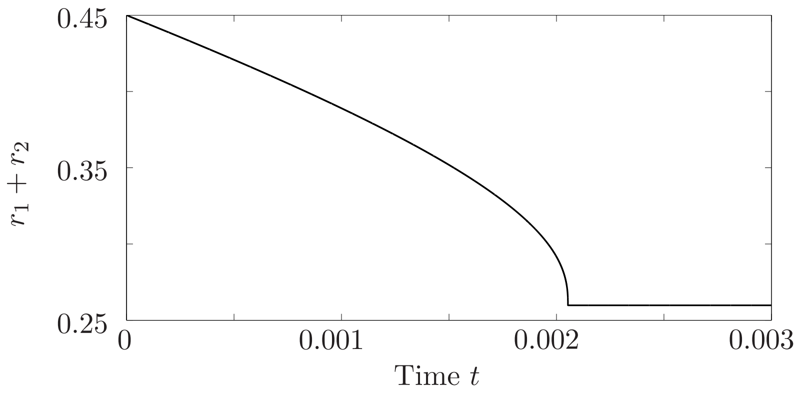


Figure 6.1: Evolution of the radii of two concentric circles given via the Mullins–Sekerka Problem in two spatial dimensions.

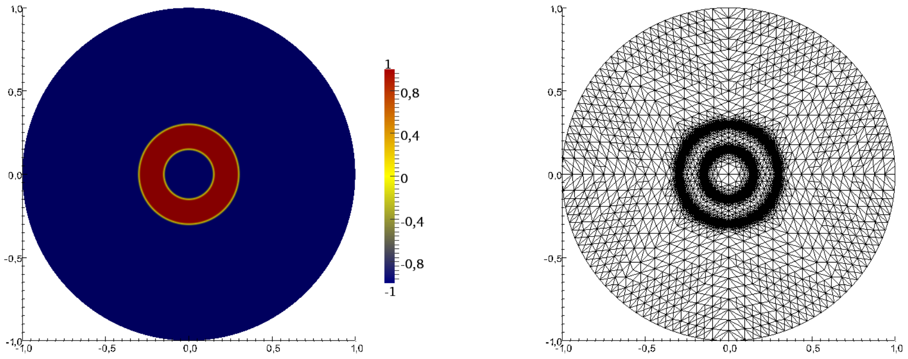


Figure 6.2: Initial data and mesh for $\gamma = 1$, $\varepsilon = \frac{0.025}{\pi}$ in two space dimensions.

For our simulations we choose the initial data in a way that the zero level sets coincide with the desired circles or spheres and the diffuse interfaces with a width of $\varepsilon\pi$ are already present, see Figure 6.2.

At some point in time the smaller one of the circles vanishes, resulting in a singularity in the sharp interface model. In each simulation there is a deviation from this exact critical time t_c, where the smaller of the circles should vanish, depending on the underlying mesh, the time step size and obviously the parameters ε and γ. When the zero level set of the smaller circle has vanished, the concentration changes for a few more time steps, where the remaining mass of the smaller circle is transported to the outer area Ω_0. After that the configuration remains constant. Figure 6.3 and Figure 6.4 show the values of u and the underlying mesh of such an evolution process with $\gamma = 1.0$, $\varepsilon = \frac{0.00625}{\pi}$ and $\tau = 10^{-7}$ for the implicit and semi-implicit discretization scheme. Note that in the semi-implicit simulation the smaller circle vanishes later in time than in the more precise implicit case, which is discussed later in this section.

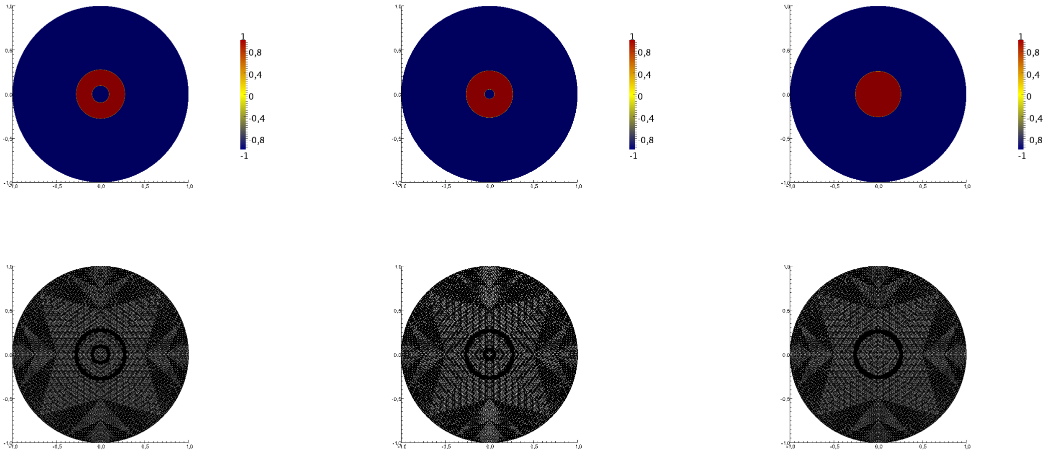


Figure 6.3: Evolution of two concentric circles with the fully implicit time discretization scheme at times $t = 1.2 \cdot 10^{-3}$, $1.8 \cdot 10^{-3}$ and $2.1 \cdot 10^{-3}$.

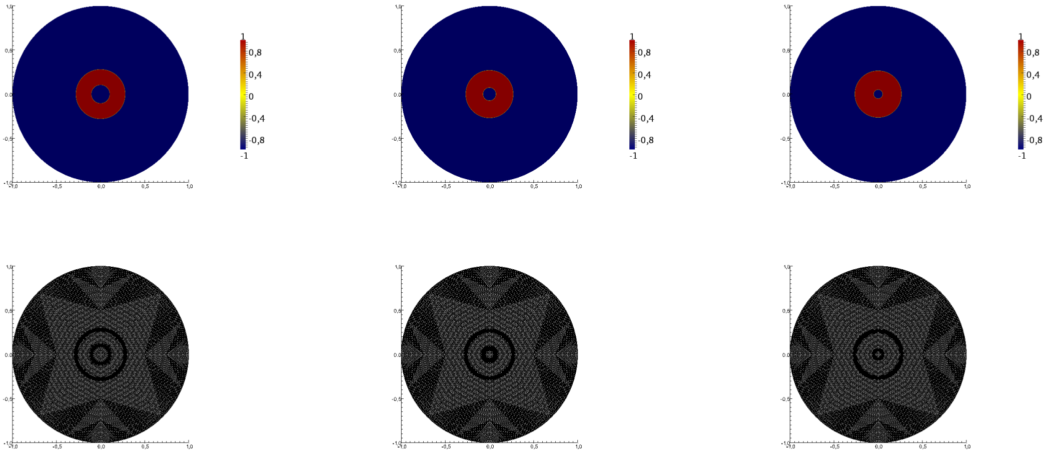


Figure 6.4: Evolution of two concentric circles with the semi-implicit time discretization scheme at times $t = 1.2 \cdot 10^{-3}$, $1.8 \cdot 10^{-3}$ and $2.1 \cdot 10^{-3}$.

6.1.1 Spatial approximation error in mesh coupled to the interface width

In this first experiment one can see the expected behavior, that the spatial approximation error influences the exactness of the evolution process. As general setup we use a fixed very small time step size of $\tau = 10^{-8}$, surface tension of $\gamma = 1.0$ and a mesh width coupled to the varying interfacial parameter ε. Below we present the graph showing the results for the fully implicit time discretization scheme. The same experiment conducted with the semi-implicit time discretization essentially leads to the same graphic and is thus omitted. In Figure 6.5 we show the evolution of the sum of the radii of the circles given by the zero level sets of the concentration u for simulations on an adaptive grid. One can see that the general behavior is correct for all of the simulations. The velocity of the movement is dependent on the curvature of the interfaces and speeds up the smaller the circles get. Also the monotone convergence of the model with respect to the parameter ε can be observed nicely.

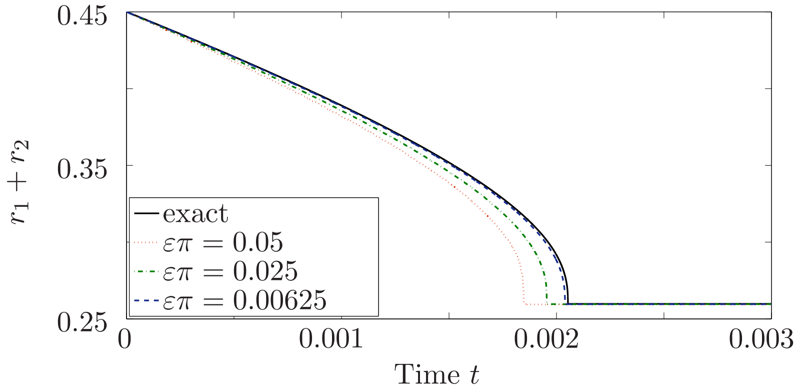


Figure 6.5: Influence of ε on the quality of the approximation.

6.1.2 Comparison of implicit and semi-implicit time discretization

Naturally the implicit and semi-implicit Euler time discretization of the free energy term $\Psi(u)$ omits a different approximation quality. We fix $\gamma = 1.0$ and $\varepsilon = \frac{0.025}{\pi}$, such that the interface has an approximate width of 0.025. The figures below show again the sum of the level set radii versus the time. Figure 6.6 uses the semi-implicit discretization, whereas Figure 6.7 uses the fully implicit model. In Figure 6.6 we can see that the approximation is very crude for larger time steps and very small time steps are necessary to capture the evolution of the sharp interface model. The implicit discretization omits an almost perfect approximation even for large time steps, if it converges. There are some problems for large time step sizes, as we already discussed in Section 3.6.4.

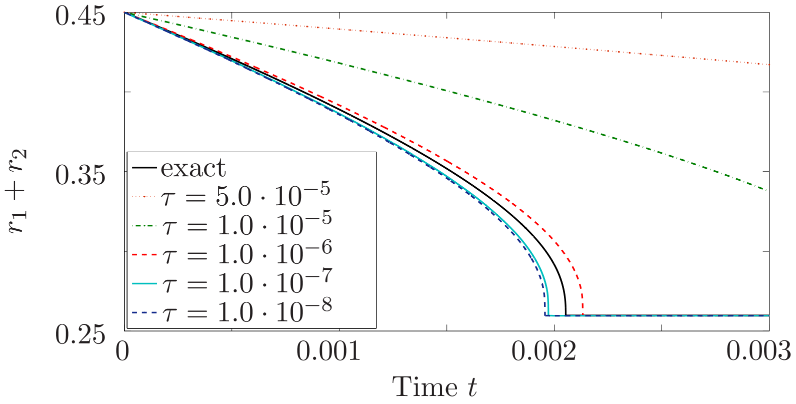


Figure 6.6: Progression of the sum of the radii with semi-implicit discretization and $\varepsilon = \frac{0.025}{\pi}$ for different time step sizes on an adaptive mesh.

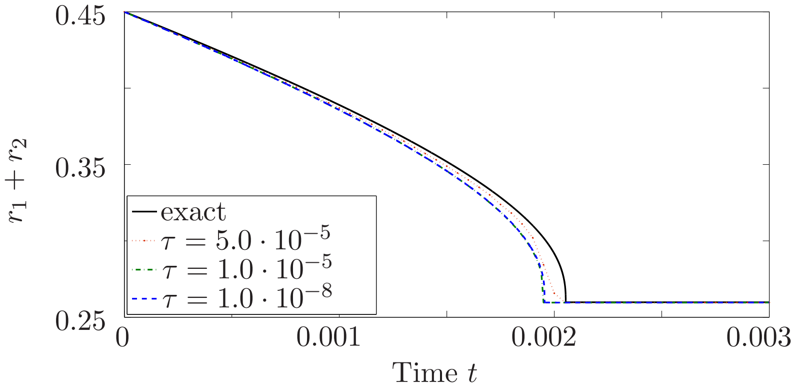


Figure 6.7: Progression of the sum of the radii with implicit discretization and $\varepsilon = \frac{0.025}{\pi}$ for different time step sizes on an adaptive mesh.

6.1.3 Influence of γ on the evolution speed

The sharp interface model suggests, that the surface tension parameter γ influences the velocity of the zero level sets movement, see Section 2.4. The analysis there suggests a linear dependency. We now use an adaptive mesh, for a fixed parameter $\varepsilon = \frac{1}{160\pi} = \frac{0.00625}{\pi}$ and a time step size $\tau = 10^{-6}$ together with an implicit scheme. Figure 6.8 shows the exact solutions given by the Mullins-Sekerka model and those given by the phase field model for three parameter γ. We can see that for $\gamma = 1.0$ the solution given by the simulation approximates the sharp interface almost exactly. The approximate solution using a smaller parameter γ is a little bit too quick but still tracks the evolution quite nicely. Increasing the size of the parameter γ leads to a slightly too fast simulation.

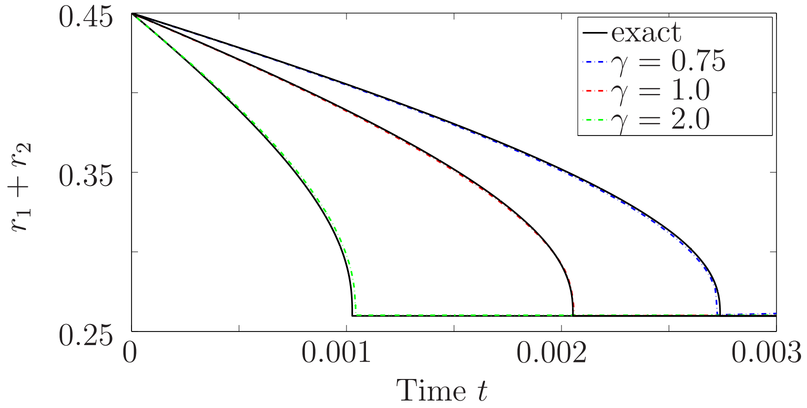


Figure 6.8: Influence of γ on the evolution process, with $\varepsilon = \frac{1}{160\pi}$ and $\tau = 10^{-6}$ on an adaptive mesh.

6.2 The primal-dual active set method with constant mobility

The following simulations show the efficiency of the primal-dual active set method applied to the Cahn–Hilliard problem with constant diffusional mobility, i.e. Algorithm 3.3. We study different initial concentration distributions, describing either a nearly homogeneous mixture or a configuration with well developed interfaces, to apply the numerical methods described in this work, too.

6.2.1 Randomly perturbed initial data

In applications one often has to consider initial data which are a random perturbation of an equally distributed concentration u. Therefore we give results on an equally distributed mass on $\Omega = (0,1)^2$ with a stochastic distortion. We define $u_0(x) := 0.5 \cdot \sigma(x) + 0.2$, where $\sigma : \Omega \to [-1,1]$ denotes a random number generator. Consequently there is no pure phase initially, i.e. all vertices are inactive, and the resulting equation system (3.21)-(3.23) is as large as possible. For this simulation we used the implicit discretization and a uniform mesh with $h = 0.00552$, $\tau = 10^{-5}$, $T_{end} = 0.005$, $\pi\varepsilon = 0.05$ and $c = 10$. In each time step where a new active set emerges, we observe large values in the Lagrangian multiplier μ, however $max\,|\mu| \leq 20$. Figure 6.9 shows u, w and μ after 0, 5, 50 and 500 time steps. Already after 5 time steps the phase separation can be clearly seen.

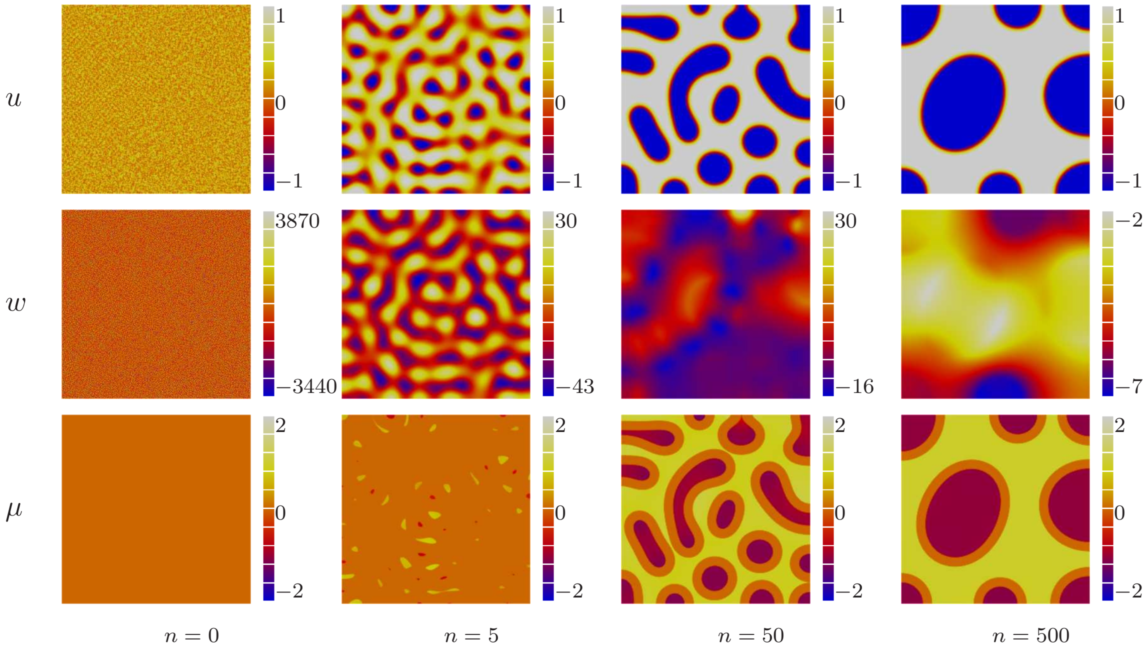


Figure 6.9: 2D simulation with random initial data.

In Figure 6.10 we see that in the early stage of this simulation one PDAS iteration is sufficient since there is no active set present and we just have to solve a linear system. After that a larger number of iterations is neccessary because there are quite a few topological changes and a huge amount of vertices changes from inactive to active. However there have never been more than 10 PDAS-iterations necessary. Afterwards when the interfaces are well developed an average amount of 2-3 iterations is sufficient.

As we described in the introduction, the next phase of the evolution process is dominated by the so called survival of the fattest stage, where already well developed interfaces are present.

6.2.2 Circular interfaces in two spatial dimensions

In this example we choose initial data with a concave as well as a convex section of the interface. The initial data on $\Omega = (0, 1)^2$ consist of four circular interfaces of width $\varepsilon\pi$. The centres and radii are chosen in such a way that three of circles intersect and one is detached. The values ± 1 are connected by a sine profile which is given as the lowest order term in an asymptotic expansion of the Cahn–Hilliard variational inequality, see

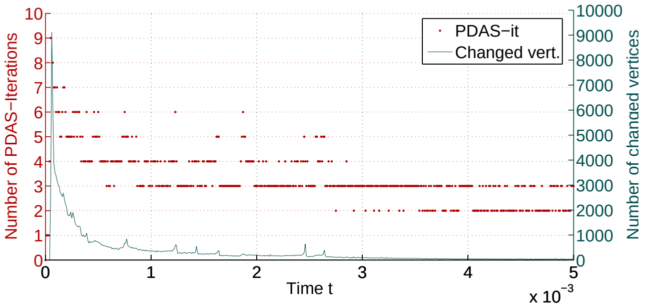


Figure 6.10: PDAS-iterations and vertices changing sets per time step for random initial data.

e.g. Blowey and Elliott [BE94]. In Figure 6.11 we show the inital data for two different interface width. The initial active sets show a value of 0 on each inactive vertex and a positive resp. negative value on each active vertex.

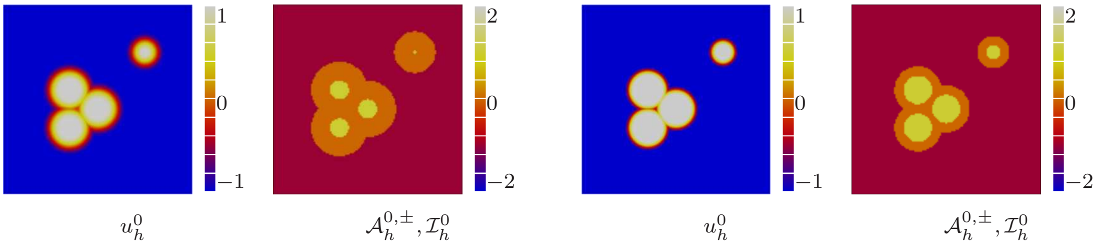


Figure 6.11: Initial data for $\varepsilon = \frac{0.1}{\pi}$ (left) and $\varepsilon = \frac{0.05}{\pi}$ (right).

We set $T_{end} = 0.02$ for all the following simulations. In Figure 6.12 and Figure 6.13 the evolution of u, w and μ in time is plotted. Here we used a semi-implicit discretization with an adaptive mesh with $h_{fine} = 0.01$ for $\varepsilon = \frac{0.1}{\pi}$ and $h_{fine} = 0.005$ for $\varepsilon = \frac{0.05}{\pi}$ respectively, the time step $\tau = 10^{-5}$. Simulations with equidistant mesh give the same results. The columns from left to right show the values of u, w, μ and the mesh after 5, 50, 100 and 200 time steps.

In our numerical experiments we use the projected block sor-method (pBSOR) with overrelaxation using $\omega = 1.3$ for comparison with the PDAS-method. As stopping criteria $\|u_k - u_{k-1}\|_2 \leq 10^{-7}$ and a maximum of 50000 iterations is used.

Table 6.1 shows that for a small number of vertices the pBSOR algorithm is still fast but with an increasing number of vertices its performance quickly deteriorates. Here the columns showing CPU times for the primal-dual active set method were generated with the direct method (UMF) as linear algebra solver for the saddle point system, since this is the fastest one of the implemented methods, compare Section 6.2.4. Using the corresponding BSOR-method in combination with the PDAS-method, the resulting solver is even a bit slower for large time steps. The direct solver on the other hand lowers the runtime considerably. Moreover we see that there is nearly no difference

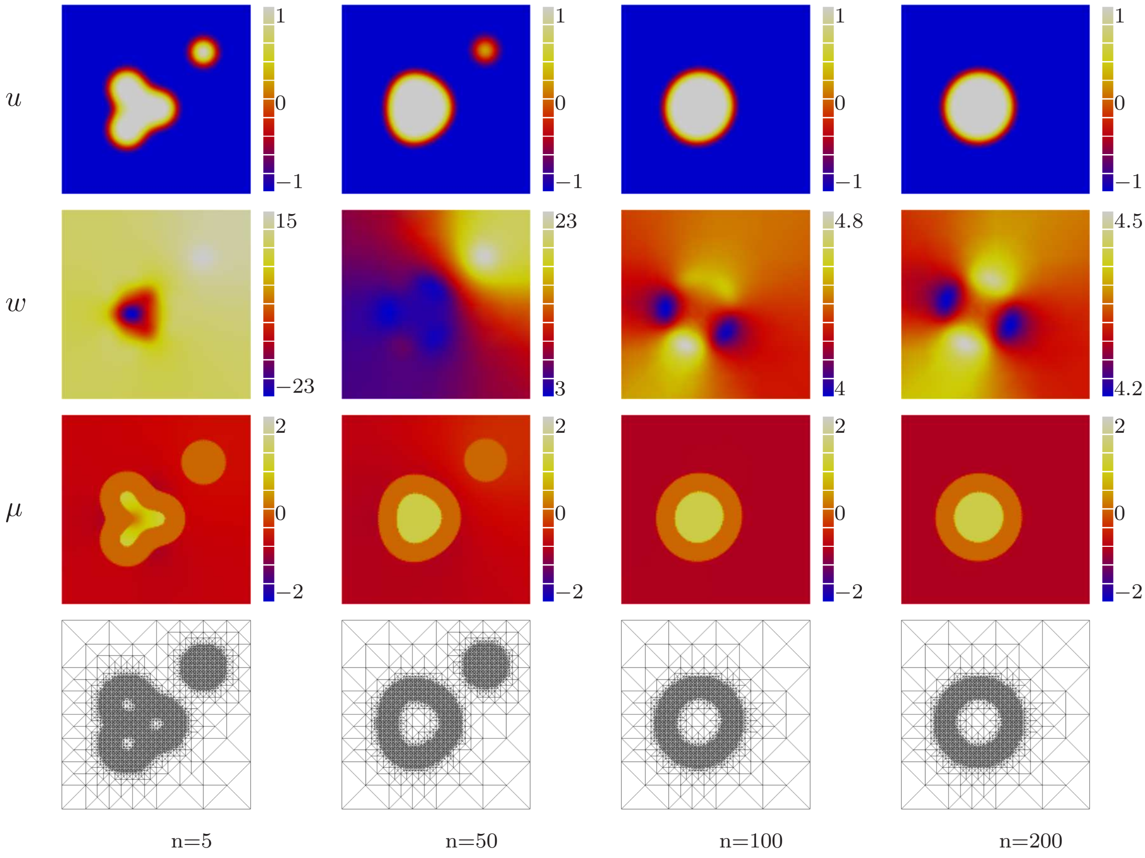


Figure 6.12: Time evolution of the example with four circular interfaces with $\varepsilon = \frac{0.1}{\pi}$.

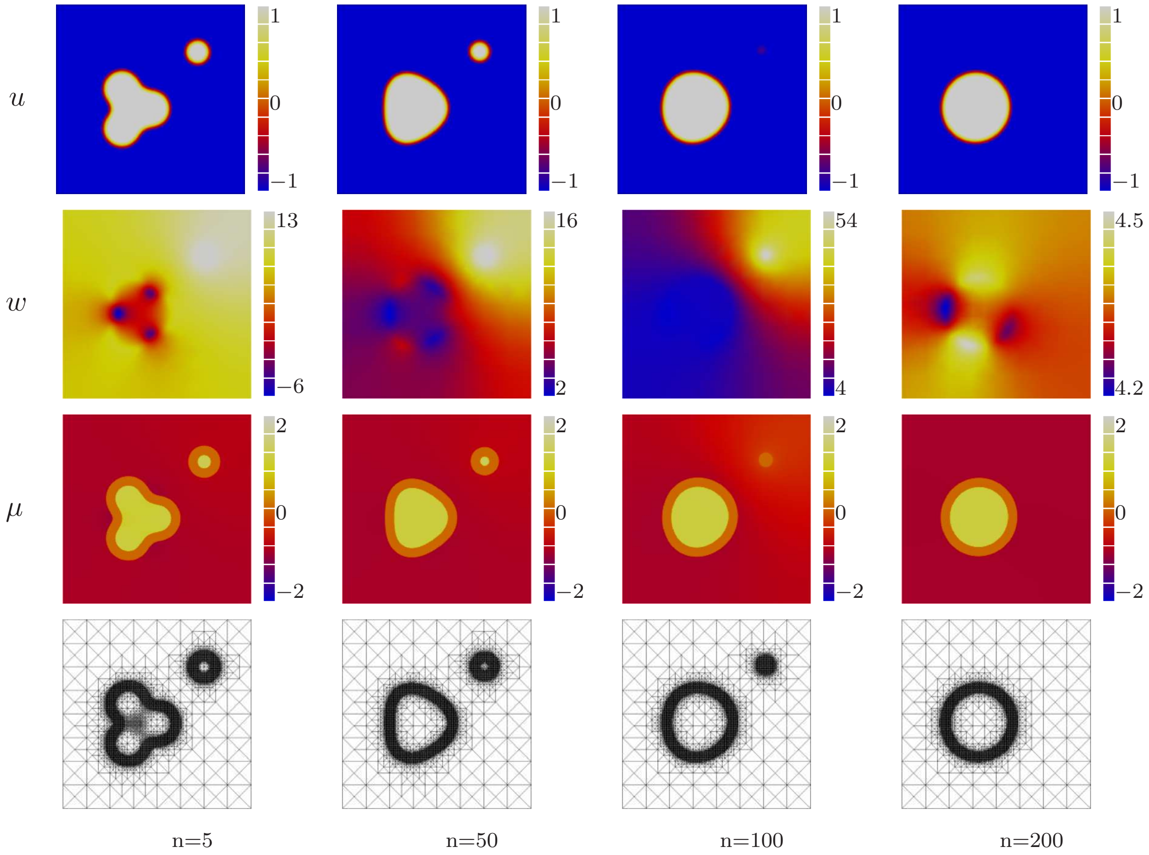


Figure 6.13: Time evolution of example with four circular interfaces with $\varepsilon = \frac{0.05}{\pi}$.

$\varepsilon\pi$	h	τ	CPU–Time in seconds			PDAS-iterations		
	(N)			semi-impl.	impl.			
			pBSOR	PDAS	PDAS	total	average	max
0.2	0.02210	10^{-4}	57.1	10.7	11.2	74	3.5	5
	(4225)	10^{-5}	270.9	29.7	63.6	450	2.2	4
		10^{-6}	703.3	195.0	202.6	2958	1.5	3
	0.01105	10^{-4}	1071.6	69.4	39.5	100	4.7	7
	(16641)	10^{-5}	5522.9	203.3	199.2	577	2.8	4
		10^{-6}	13506.2	1353.6	1325.6	3795	1.9	3
	adaptive	10^{-4}	5.2	2.9	2.9	72	3.4	5
	(≈ 2000)	10^{-5}	23.1	17.6	17.5	447	2.2	4
		10^{-6}	70.2	117.9	117.8	2968	1.5	3
0.1	0.01105	10^{-4}	1374.6	17.9	17.8	70	3.3	6
	(16641)	10^{-5}	4179.5	103.4	105.6	409	2.0	5
		10^{-6}	10111.0	793.1	727.6	2922	1.5	4
	0.00552	10^{-4}	—	130.3	134.5	91	4.3	10
	(66049)	10^{-5}	—	750.7	754.5	524	2.6	6
		10^{-6}	181285.1	4905.4	4813.2	3362	1.7	5
	adaptive	10^{-4}	45.1	5.1	5.1	69	3.3	7
	(≈ 3600)	10^{-5}	74.5	27.7	28.2	403	2.0	4
		10^{-6}	390.2	198.7	194.0	2897	1.4	3
0.05	0.00552	10^{-4}	11145.0	126.6	—	88	4.2	7
	(66049)	10^{-5}	72715.0	592.0	597.3	497	2.4	6
		10^{-6}	192554.4	3911.3	4013.2	3275	1.7	5
	adaptive	10^{-4}	737.1	13.6	—	85	4.0	7
	(≈ 7000)	10^{-5}	602.3	76.8	73.4	503	2.5	6
		10^{-6}	1478.2	467.6	478.1	3260	1.6	5

Table 6.1: CPU–Runtimes and iteration counts for the example with four circular interfaces.

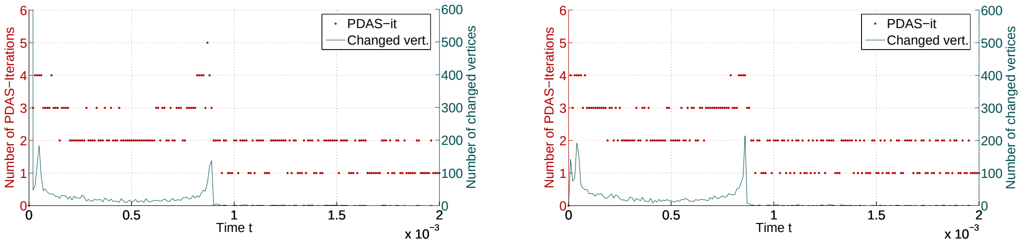
in CPU-time between the semi-implicit and the implicit discretization. The severe restriction on the time step for the implicit case as stated in Lemma 2.12 has not been observed. Only for $\varepsilon = \frac{0.05}{\pi}$ the choice $\tau = 10^{-4}$ failed even for very large parameter $c = 10^{10}$.

When we compare the runtimes used on the fixed mesh with 16641 vertices we notice that the simulations with $\varepsilon = \frac{0.2}{\pi}$ used up almost double the time of the one with $\varepsilon = \frac{0.1}{\pi}$. The reason lies in the size of the inactive set, which is roughly spoken the interface with width $\varepsilon\pi$. Hence for $\varepsilon = \frac{0.2}{\pi}$ the system (3.21)-(3.22), which has to be solved, is of larger dimension.

In addition in Table 6.1 the total, the averaged and the maximal number of PDAS-iterations are listed for the semi-implicit discretization. The numbers for the implicit discretization are nearly the same except for the failures and hence not listed. The average number of PDAS-iterations depend more on the time step than on the mesh.

This is an expected behavior since when we use larger time steps the active sets change on a bigger scale than with smaller time steps. In most of the above simulations the maximum number of iterations was needed in the first time step. The reason is that the mean curvature of the interface is high in the beginning of the time evolution, resulting in fast movement of the interface region. Even taking a rather large time step, like for example $\tau = 10^{-4}$ for $\varepsilon = \frac{0.05}{\pi}$, the maximum number of necessary iterations per time step keeps low and never exceeded 11. The averaged numbers of iterations are much smaller since the time evolution of the interface becomes slow for larger t, resulting in only one or two PDAS-iterations.

In Figure 6.14 we plot the time against the number of used PDAS-iterations per time step as well as against the number of changed vertices per time step for the above simulation with $\varepsilon = \frac{0.1}{\pi}$, $\tau = 10^{-5}$, in the semi-implicit and implicit case for an adaptive mesh with $h_{fine} = 0.00552$.



semi-implicit PDAS

implicit PDAS

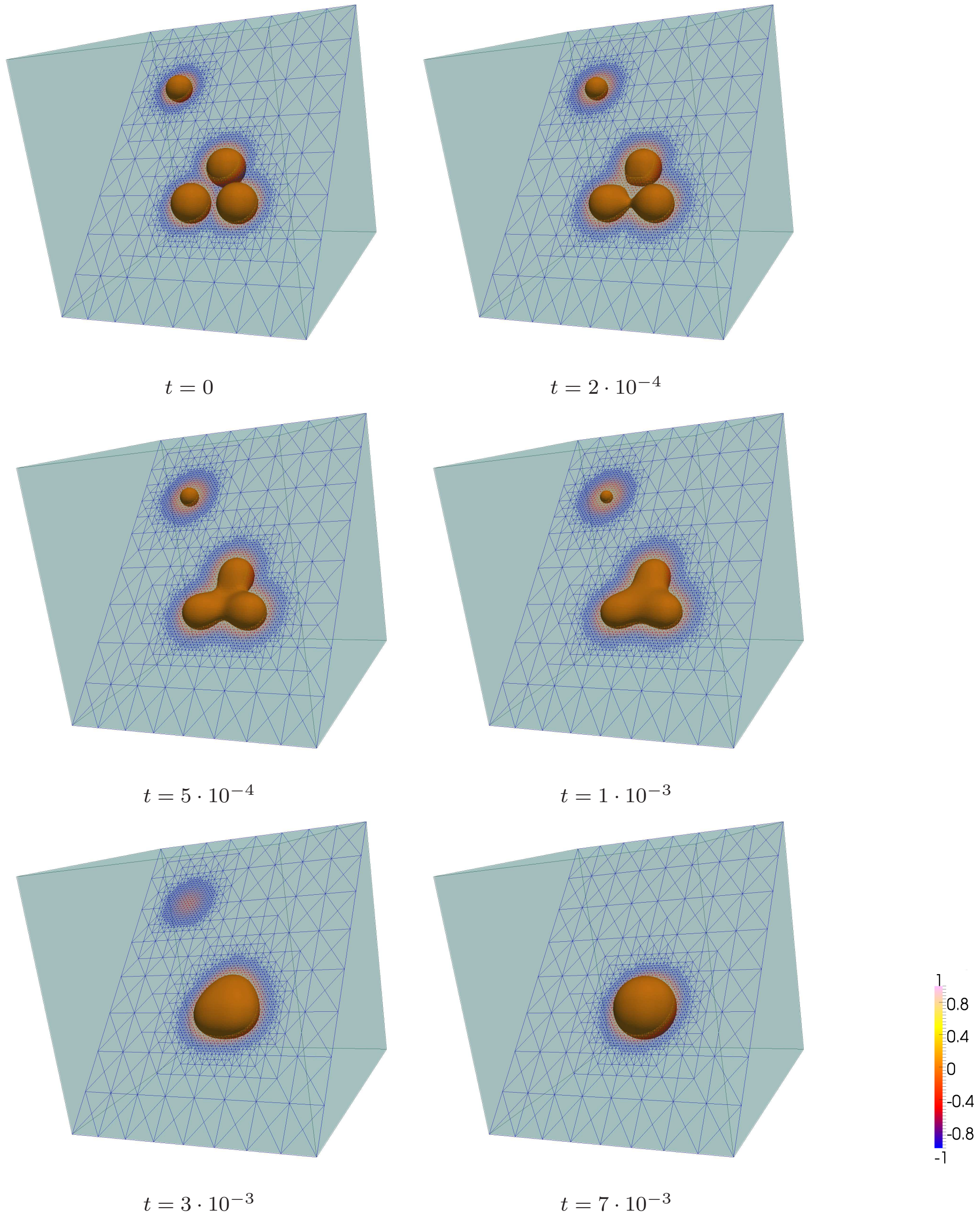
Figure 6.14: PDAS-iterations and vertices changing sets per time step.

In the first few time steps the evolution smoothens the interfaces and the concave part is moving quickly. These two facts result in an increased number of neccessary PDAS-iterations. After that typically two to four iterations are sufficient. The steps where we only need two iterations are optimal in a way that if there is any change in the active set we need at least these two iterations. Only when there are no changes in the sets just one iteration is sufficient. What we can observe in these plots is the expected rise in iteration numbers when there is a big change in the active set. The second peak is due to the disappearance of the bubble in the upper right quadrant. If we use an equidistant mesh for the above example, the results and numbers of PDAS-iterations stay nearly the same, although in the adaptive case we have to adapt the starting active set due to a grid change in time, see Section 3.6.1.

However, instead of roughly 10 minutes CPU-time for an equidistant grid only 76 seconds CPU-time is needed in the adaptive case to determine a solution up to $T = 0.02$.

6.2.3 Spherical interfaces in three spatial dimensions

Finally we give an example in 3D. Therefore we expand the example with circular interfaces above, see Section 6.2.2, to initial data consisting of four balls in $\Omega = (0,1)^3$. Figure 6.15 shows the zero–level sets of u of such a simulation with $\tau = 10^{-5}$, $\pi\varepsilon = 0.1$ and $c = 10$ after 0, 20, 50, 100, 300 and 700 time steps on an adaptive mesh with the semi-implicit primal-dual active set solver. For these examples in three dimensions we use the conjugate gradient method with a simple diagonal preconditioning for the solution of the saddle point problem, i.e. we apply Algorithm 5.3, where $\mathbf{F}$ and $\boldsymbol{f}$ are given by the full saddle point system (4.1) and $\mathbf{P}$ by its diagonal entries.


$t = 0$

$t = 2 \cdot 10^{-4}$

$t = 5 \cdot 10^{-4}$

$t = 1 \cdot 10^{-3}$

$t = 3 \cdot 10^{-3}$

$t = 7 \cdot 10^{-3}$

Figure 6.15: Zero–level sets of a 3D simulation with 4 spheres as initial data on an adaptive mesh for $\pi\varepsilon = 0.1$ and a slice, showing the mesh.

The simulation up to $T_{end} = 0.015$, i.e. 1500 time steps, where a coupled system corresponding to roughly 96000 grid points has to be solved, took 1.2 hours with a total of 2537 PDAS-iterations. This is only five percent of the computation time used by the pBSOR method which took 23.5 hours. Additional speed up -which is not possible for the pBSOR-method- can be obtained by a different linear algebra solver. Even for this three dimensional problem with the topological changes a maximal number of only five PDAS-iterations in each time step is sufficient for the simulation. In Figure 6.16 we plot the time against the number of primal-dual active set iterations as well as the number of vertices changing from one set to another. The highest number of iterations occured again in the first time step, where the initial data for the active and inactive sets had to be guessed and hence a larger number of vertices needs to change sets. Also the second peak, where a larger number of vertices changes occurs at the point in time, where the three connected spheres merge completely.

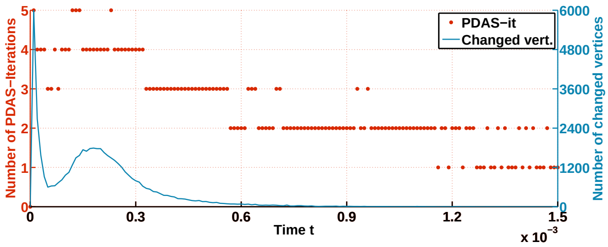


Figure 6.16: PDAS-iterations and vertices changing sets per time step for four spherical interfaces as initial data with $\varepsilon\pi = 0.1$ on an adaptive mesh.

Computation times of simulations using different meshes and time step sizes are shown in Table 6.2. There we stop the simulations earlier since the evolution process slows down considerably. This is expected since this final stage adheres to a slower time scale as we discussed in the introduction. We set $T_{end} = 0.0015$ and hence omit the time steps, where almost no changes occur. For the generation of the table we only used the diagnoally preconditioned conjugate gradient method as solver for the sadde point problem. The effects of the use of other linear algebra solvers is studied in the next subsection.

$\varepsilon\pi$	h	τ	CPU–Time in seconds		PDAS-iterations		
	(N)		pBSOR	PDAS-CG	total	average	max
0.2	0.01969	10^{-4}	10927.1	558.6	72	4.5	6
	(170081)	10^{-5}	58112.6	2216.1	382	2.5	5
		10^{-6}	375175.1	10874.9	3260	2.2	5
	adaptive	10^{-4}	928.6	156.9	69	4.6	7
	($\approx$ 33000)	10^{-5}	4231.1	581.2	378	2.5	5
		10^{-6}	18334.5	3126.1	2091	1.4	4
0.1	adaptive	10^{-4}	4186.5	196.9	63	3.9	8
	($\approx$ 96000)	10^{-5}	17277.7	584.3	377	2.5	5
		10^{-6}	54251.9	2575.4	2790	1.8	4

Table 6.2: CPU–Runtimes and iteration counts for the example with four spherical interfaces.

6.2.4 Study of the different interior linear algebra solvers

Up to this point we just compared the runtimes for the projected block sor solver to the primal-dual active set method with the direct solver for two spacial dimensions or a conjugate gradient method applied directly to the full saddle point problem, in case of three spatial dimensions. Now we will use the first two time steps of the four circles resp. spheres setting to compare the different linear algebra solvers. We restrict the investigation to $\gamma = 1.0$ and use equidistant as well as adaptive meshes. Furthermore we fix the time step size to match the mesh size, i.e. we set $\tau \approx \frac{4\varepsilon^2}{1000\gamma}$.

Both Schur complement formulations require the solution of the discrete Laplacian in each iteration. In two spatial dimensions we use the direct solver UMFPack for this. This is very efficient, since the mesh does not change during one time step and thus the factorization can be reused. In three dimensions a conjugate gradient method is used instead. At this point a further improvement on the performance of the outer conjugate gradient method could be obtained by the use of a stronger method, i.e. a geometric multigrid method, which is not available to us in ALBERTA.

The times given in Tables 6.3 and 6.4 show that in two spatial dimensions the direct solver is very strong. Later in the simulation, when the evolution slows down, the iterative methods gain an advantage in comparison to the direct solver, whose effort is independent of the quality of the initial guess for the solution. Comparing the classical (pBSOR) solver to the primal-dual active set methods, we can see, that with the exception of very crude approximations with large ε the efficiency of the pBSOR method deteriorates quickly and for the simulation with $\varepsilon\pi = 0.025$ no convergence is obtained within 50000 iterations. The number of iterations required could eventually be decreased with the help of a smaller time step size, but then the number of time steps required increases too. The primal-dual active set method in conjunction with the BSOR method is even slower, however a reformulation allows for stronger methods.

If we consider the simulations in three dimensions the direct method is no longer competitive due to the huge amount of fill in. The simulations with $\varepsilon\pi \geq 0.05$ required more than 8 Gigabytes of memory. Nevertheless again the possibility to use good linear algebra solvers for the system of equations instead of an inequality allows for a significant speed up.

The performance of the Schur complement solver depends heavily on the solver used for the Laplacian required in each iteration. In upto two spatial dimensions this can be done very efficiently with the direct solver, since it is suffcent to create the LU-decomposition only once in each time step. After that a simple forward-backward sweep generates the solution. In three dimensions this option is no longer viable and we use a standard conjugate gradient method. This is obviously not good enough to beat the conjugate gradient method applied directly to the whole saddle point problem and more sophisticated methods are required to improve the results.

	$\varepsilon\pi$	τ	vertices	pBSOR	PDAS-it.	BSOR	UMF	CG	SC	PrecSC
2D	0.2	$2 \cdot 10^{-4}$	1012	0.09, 0.07	4, 4	0.13, 0.11	0.07, 0.07	0.03, 0.03	0.16, 0.15	0.07, 0.07
	0.1	$4 \cdot 10^{-5}$	2228	0.52, 0.54	5, 4	1.07, 1.02	0.21, 0.15	0.07, 0.07	0.75, 0.54	0.18, 0.15
	0.05	$1 \cdot 10^{-5}$	4852	1.89, 2.23	5, 4	4.52, 5.37	0.51, 0.36	0.37, 0.24	2.34, 1.53	0.42, 0.29
	0.025	$2 \cdot 10^{-6}$	10457	9.71, 7.71	5, 5	27.64, 27.21	1.05, 0.97	0.81, 0.79	5.19, 3.69	1.07, 0.84
	0.01	$4 \cdot 10^{-7}$	33310	140.22, 111.43	6, 6	875.40, 863.90	4.22, 4.08	5.19, 4.80	10.23, 8.63	3.55, 3.33
	0.005	$1 \cdot 10^{-7}$	75227	674.97, 524.99	6, 6	1587.14, 1391.22	9.78, 9.65	18.25, 15.76	23.98, 18.93	13.79, 11.28
	0.001	$4 \cdot 10^{-9}$	890787	—	7, 7	—	262.51, 267.53	404.48, 318.87	385.04, 307.12	156.18, 128.22
3D	0.2	$2 \cdot 10^{-4}$	33210	40.50, 25.61	7, 4	193.70, 70.92	296.11, 188.12	12.29, 6.46	96.04, 52.30	65.33, 44.34
	0.1	$4 \cdot 10^{-5}$	95924	345.01, 268.02	7, 4	1507.14, 888.27	1642.12, 1313.91	41.06, 41.79	161.14, 124.11	109.49, 83.23
	0.05	$1 \cdot 10^{-5}$	347204	1511.40, 1180.20	6, 4	3226.92, 2289.67	—	149.05, 72.05	968.84, 665.69	547.82, 368.27
	0.025	$2 \cdot 10^{-6}$	1334773	9151.11, 7914.22	5, 4	28935.04, 14273.33	—	544.34, 168.88	3515.14, 3157.97	1642.41, 1432.7

Table 6.3: CPU–Runtimes in seconds for various linear algebra solvers of the fist two time steps of the example with four circular/spherical interfaces on an adaptive mesh.

	$\varepsilon\pi$	τ	vertices	pBSOR	PDAS-it.	BSOR	UMF	CG	SC	PrecSC
2D	0.2	$2 \cdot 10^{-4}$	4225	5.89, 4.79	5, 5	12.21, 10.67	0.35, 0.33	0.38, 0.36	0.96, 0.94	0.43, 0.39
	0.1	$4 \cdot 10^{-5}$	16641	97.18, 87.88	4, 5	140.72, 158.34	1.12, 1.34	2.03, 2.26	3.46, 4.18	1.26, 1.35
	0.05	$1 \cdot 10^{-5}$	66049	—	6, 5	—	7.67, 6.32	17.37, 16.90	29.58, 22.66	6.56, 4.50
	0.025	$2 \cdot 10^{-6}$	263169	—	5, 6	—	34.34, 41.04	107.89, 101.88	100.06, 97.34	32.57, 26.50
3D	0.2	$2 \cdot 10^{-4}$	170081	1222.54, 934.12	6, 4	4403.61, 3152.11	—	63.98, 45.27	467.01, 385.54	222.85, 143.67
	0.1	$4 \cdot 10^{-5}$	1335489	—	5, 6	—	—	645.48, 847.88	4121.11, 4099.15	1935.80, 1602.10

Table 6.4: CPU–Runtimes in seconds for various linear algebra solvers of the fist two time steps of the example with four circular/spherical interfaces on an equidistant mesh.

6.3 Performance of the preconditioned Schur complement solver

In this section we test the efficiency of the preconditioning with some numerical experiments in two different settings, which we have already used before. Both are given on $\Omega = [0,1]^d$, where $d = 2, 3$, and we set $\gamma = 1.0$. The parameter c, which controls the primal-dual active set strategy, is set to a fixed value of 10. The implemented boundaries for the iteration counts are set to 20 for the primal-dual active set iterations and to 1000 for the conjugate gradient method iterations. We use the following two configurations:

1. The radially symmetric setting, where the interface describe two concentric circles around the point $(0.5, 0.5)$ and in $\mathbb{R}^3$ two concentric spheres around the point $(0.5, 0.5, 0.5)$ respectively, describes an easy situation, where exact solutions to the sharp interface model are known. We choose the radii $r = 0.05$ and $R = 0.15$. The Mullins-Sekerka model, which is the corresponding sharp interface model to our phase field approach, see Section 2.4, predicts, that at time $T_e = 1.21 \cdot 10^{-4}$ the smaller circle has vanished and the remaining system remains stable. Hence we set the final time $T = 2 \cdot 10^{-4}$ to allow for the deviation caused by the explicit approximation. This setting is similar to the one used in Section 6.1, see also Figures 6.2-6.4.

2. In the second setting, the interface consists of four circular shapes as described in Section 6.2.2, see Figure 6.11 for a depiction. The final time, where the state is nearly steady, is given by $T = 1.2 \cdot 10^{-3}$.

In each PDAS iteration we initially calculate a solution u_{exact} by means of the direct solver UMFPack and use $\|u - u_{exact}\|_2 < tol_{cg} := 10^{-7}$ as abort criterion for the projected conjugate gradient methods with and without preconditioning. This is not a useful condition in practice, but it provides us with a good environment to compare the unpreconditioned and preconditioned solver independently.

We want to stress that in each iteration of the conjugate gradient method a Laplace-type problem has to be solved, when we calculate the application of $\mathbf{F}$. Our experiments show the method, solving this subproblem, must be able to reach a small tolerance in comparison to tol_{cg}. This is similar to the required tolerance for the solver of the saddle point problem to obtain a converging primal-dual active set iteration, see Section 3.6.5. When we use the direct solver UMFPack this is no restriction. When we use the standard multigrid solver from ALBERTA we need to prescribe a tolerance of at least 10^{-10} to obtain a converging CG iteration. If the tolerance is set adequately the method is independent of the type of solver we use. For the generation of the tables in this section we used the direct solver. As we discussed earlier, the advantage is given by the very good re-usability of the generated factorizations.

The same is true for the part of the program that takes care of the solution of the subproblem, when applying the preconditioner, given by $\mathbf{Id} + \sigma\mathbf{C}$. Thus we use the direct solver again, where possible.

6.3.1 Comparison type 1 - adaptive mesh, fixed $\tau \approx \Delta x \cdot 10^{-2}$ and variable interface width $\varepsilon\pi$

With this first simulation series we will test the behavior of the CG solver with respect to a shrinking parameter ε. Note that $\varepsilon\pi$ is approximately the interface width. The mesh is generated by an adaptive algorithm, which refines the mesh on the interface to a level that ensures the existence of at least eight vertices across the interface and coarsens outside of the interface region. For decreasing parameter ε the mesh size Δx_{min} is reduced by the same factor and the time step size is adjusted accordingly.

$\varepsilon\pi$	τ	avg	Unpreconditioned SC				Preconditioned SC			
		vertices	total	max	avg	CPU	total	max	avg	CPU
0.2	$2 \cdot 10^{-4}$	1974	no convergence				382	12	10	7
0.1	$1 \cdot 10^{-4}$	3724	3808	84	56	21	583	12	9	13
0.05	$5 \cdot 10^{-5}$	7191	9615	108	64	89	1187	11	8	52
0.025	$2.5 \cdot 10^{-5}$	14627	20811	127	72	369	2139	11	8	200
0.0125	$1.25 \cdot 10^{-5}$	30565	39977	140	73	1605	3496	10	7	833

Table 6.5: Comparison type 1 - 4 circles setting.

Table 6.5 shows that for decreasing ε the maximal as well as the average number of iterations increases steadily in the unpreconditioned case, whereas with the preconditioner in place, they stay low and are even decreasing. Also we can observe the quadruplication of the CPU time needed when we divide the parameter ε by two for the preconditioned as well as the unpreconditioned method. This is an expected behavior, because the number of vertices, i.e. the dimension of the equation system, and the number of time steps are both doubled.

$\varepsilon\pi$	τ	avg	Unpreconditioned SC				Preconditioned SC			
		vertices	total	max	avg	CPU	total	max	avg	CPU
0.2	$2 \cdot 10^{-5}$	1203	517	35	26	1.7	159	10	8	1.7
0.1	$1 \cdot 10^{-5}$	2391	1346	44	34	9	307	10	8	8
0.05	$5 \cdot 10^{-6}$	5020	3484	54	41	42	639	10	8	34
0.025	$2.5 \cdot 10^{-6}$	10501	10432	68	53	235	1465	9	8	164
0.0125	$1.25 \cdot 10^{-6}$	22927	26752	71	64	1362	2920	8	7	789

Table 6.6: Comparison type 1 - Radially symmetric setting.

In Table 6.6 the same behavior can be observed. Here we can see additionally that for decreasing ε the shortening of the used CPU time with the use of the preconditioning increases more and more.

6.3.2 Comparison type 2 - equidistant mesh, $\tau = 10^{-6}$ and $\varepsilon = \frac{0.1}{\pi}$ fixed

Now we want to study the behavior of the solver with respect to a varying mesh size. Therefore we fix the other parameters τ and ε and carry out the simulation for a changing number of global refinements of the initial macro mesh. The mesh we use here is not an adaptive mesh, but an equidistant mesh generated by a number of bisection steps given by the number of global refinements.

glob.	vertices	Δx	PDAS	Unpreconditioned SC				Preconditioned SC			
ref.			it.	total	max	avg	CPU	total	max	avg	CPU
11	4225	0.02	1640	24349	24	15	242	20503	18	13	276
13	16641	0.01	2060	39530	34	20	1750	25472	19	13	1767
15	66049	0.005	2385	79524	61	34	14590	32157	20	13	9758

Table 6.7: Comparison type 2 - 4 circles setting.

glob.	vertices	Δx	PDAS	Unpreconditioned SC				Preconditioned SC			
ref.			it.	total	max	avg	CPU	total	max	avg	CPU
10	2113	0.03	218	1765	11	9	12	1449	8	7	12
11	4225	0.02	220	2216	13	11	30	1598	9	8	30
12	8321	0.015	245	3103	18	13	82	1764	10	8	78
13	16641	0.01	253	4416	24	18	227	1888	10	8	192
14	33025	0.007	273	6511	34	24	660	2067	10	8	512

Table 6.8: Comparison type 2 - Radially symmetric setting.

Table 6.7 and Table 6.8 show that there is only a slight dependency of the preconditioned algorithm with respect to the mesh. We want to note that the bisection of the mesh effectively doubles the amount of inactive vertices across the interface. Therefore the number of vertices contained in the interface, i.e. the width of the interface measures in the number of vertices, has almost no influence on the condition of the preconditioned system matrix. In the radially symmetric simulation the situation is not as bad as for the 4 circles case but essentially this case is a one dimensional problem.

6.3.3 Comparison type 3 - $\tau = 10^{-5}$, adaptive mesh and variable interface width $\varepsilon\pi$

Here we can see that the additional computational effort used for the preconditioning is too large to achieve a faster simulation for relatively large ε. As we have seen before the number of CG iterations doesn't increase for the preconditioned system in comparison to the unpreconditioned method, hence the computational costs stay at an adequate level and for $\varepsilon\pi > 0.05$ we can observe an increasing speed up. Comparing the CPU times of Table 6.9 with the corresponding rows of Table 6.1 we can see that even the unpreconditioned SC method is superior to the pBSOR method. However since the examples are still in two spacial dimensions the direct solver remains the fastest method.

$\varepsilon\pi$	avg.	PDAS	Unpreconditioned SC				Preconditioned SC			
	vertices	it.	total	max	avg	CPU	total	max	avg	CPU
0.2	2043	260	5513	31	22	36	2993	15	12	44
0.1	3329	285	9459	51	34	81	2924	16	11	72
0.05	6509	346	17362	82	51	250	3413	16	10	168
0.025	13922	390	25584	116	66	720	3329	13	9	410
0.0125	30262	428	30955	138	73	1884	2859	10	7	951

Table 6.9: Comparison type 3 - 4 circles setting.

6.3.4 Preconditioning with implicitly discretized free energy

All simulations above were done with an explicit time discretization of the free energy term. To show the efficiency of the preconditioning also in case of an implicit discretization, we repeat comparison type 1 with the implicit method. Table 6.10 shows the results for the 4 circles setting, whereas Table 6.11 concerns the radially symmetric data.

$\varepsilon\pi$	τ	PDAS	Unpreconditioned SC				Preconditioned SC			
		it.	total	max	avg	CPU	total	max	avg	CPU
0.2	$2 \cdot 10^{-4}$	29	no convergence				210	11	8	1.1
0.1	$4 \cdot 10^{-5}$	100	1862	48	19	8.5	461	10	5	6.4
0.05	$1 \cdot 10^{-5}$	345	5024	55	15	50.5	1285	11	4	47.1
0.025	$2 \cdot 10^{-6}$	1536	13060	59	9	345.7	4140	11	3	281.5

Table 6.10: Comparison type 1 - 4 circles setting with implicit free energy.

In Table 6.10 we used smaller time step sizes to avoid a strong influence of the adaptive mesh due to the fast movement of the interface, compare Section 3.6.4. The results are not as strong as in the explicit case, which is caused by the smaller time step sizes. The unpreconditioned method deteriorates for larger time steps, as we have seen before. The preconditioned system on the other hand suffer no such drawback, as we will see below.

The radially symmetric simulation with $\varepsilon = \frac{0.025}{\pi}$ and the same time step as in the explicit test, resulted in one time step, where the primal–dual active set did not converge, since one vertex oscillated between active and inactive set. After aborting the time step due to the maximum iteration count and continuing in the simulation, the next time step smoothed out the problem and the simulation finished without any more problems. Simply using a smaller time step size fixed this problem. Here the preconditioning still works, but the number of iterations is no longer independent of the refinement. However changing the time step size has no discernible influence on the required number of iterations per time step for the solution of the preconditioned system. This is illustrated by the additional simulations with $\varepsilon = \frac{0.025}{\pi}$ using different time step sizes.

$\varepsilon\pi$	τ	avg	Unpreconditioned SC				Preconditioned SC			
	vertices		total	max	avg	CPU	total	max	avg	CPU
0.2	$1 \cdot 10^{-5}$	1203	553	37	30	1.8	185	12	10	1.6
0.1	$5 \cdot 10^{-6}$	2391	1441	54	42	8.0	366	13	11	6.9
0.05	$5 \cdot 10^{-6}$	5020	3976	69	52	47.4	849	14	12	32.2
0.025	$2.5 \cdot 10^{-6}$	10501	13591	115	82	296.3	2218	17	14	157.3
0.025	$1 \cdot 10^{-6}$	10501	21265	88	62	498.9	4203	16	13	306.1
0.025	$5 \cdot 10^{-7}$	10501	31094	69	50	786.6	7522	17	13	560.9

Table 6.11: Comparison type 1 - Radially symmetric setting with implicit free energy.

6.3.5 Condition numbers

Finally to round out the numerical study of the preconditioning, we study two configurations and calculate the condition number and spectral radius of the Schur complement operator with an explicit discretization of the free energy. First we consider the example with four circular interfaces in two spacial dimensions, i.e. $\Omega = [0, 1]^2$. We set $\varepsilon = \frac{0.1}{\pi}$, $\gamma = 1.0$, $\tau = 4 \cdot 10^{-5}$ and $c = 10$ and use the adaptive mesh, which results in approximately 2216 vertices. Table 6.12 shows the values for some selected time steps. We can see that the largest eigenvalue of the preconditioned system is always exactly one and the condition number is very close to one.

As a second example, we use initial data, where the interface describes two circles of the same radius $r = 0.1$ in $\Omega = [0, 1]^2$. We set $\varepsilon = \frac{0.1}{\pi}$, $\gamma = 1.0$ and $\tau = 4 \cdot 10^{-5}$ as well as $c = 10$. Furthermore we use an adaptive mesh resulting in roughly 3300 vertices. Table 6.13 shows the condition numbers and spectral radii of the unpreconditioned and preconditioned Schur complement operator at the second primal-dual active set iteration of the first time step for different distances d between the two shapes. We

Time step	PDAS iteration	$cond(SC)$	$\rho(SC)$	$cond(PrecSC)$	$\rho(PrecSC)$
First	1	113.4317	73277	1.0003	1.0
	2	117.0272	71473	1.0007	1.0
	3	4299.7184	63588	1.1347	1.0
	4	299.7184	68164	1.0089	1.0
Second	1	116.4975	73293	1.0031	1.0

Table 6.12: Condition numbers and spectral radii of the (un-)preconditioned Schur complement operator for the four-circles setting in two dimensions.

would like to point out, that the inactive sets are connected in the last case with $d = 0.08$. Similar values have been observed for the other time and iteration steps.

d	$cond(SC)$	$\rho(SC)$	$cond(PrecSC)$	$\rho(PrecSC)$
0.65	88.8369	294250	1.0006	1.0
0.35	88.8369	294250	1.0005	1.0
0.08	88.8371	294250	1.0005	1.0

Table 6.13: Condition numbers and spectral radii of the (un-)preconditioned Schur complement operator for two circular interfaces with variing distance.

6.4 Surface diffusion

We restrict ourselves to simulations, where the diffusional mobility is discretized explicitly in time. Subsequently the movement of the mobile set is confined to a layer of one additional vertex along the boundary of the mobile set. This can also be understood as a restriction on the velocity of the surface movement or similarly as an upper bound on the time step size, compare [BBG99].

Let $\Omega = [0,4] \times [0,1]^{d-1}$, where $d = 2, 3$. We use initial data, where the zero–level set describes a dumbbell, see Figure 6.17. We set $\varepsilon = \frac{0.1}{\pi}$, $\gamma = 1.0$, $c = 100$ for all simulations below. Additionally we fix the diameter of the handle at 0.04 and vary the time step size as well as the type of time discretization used for the free energy term. Furthermore we use the adaptive mesh, resulting in roughly 14000 vertices in two spatial dimensions and two million vertices in three. The same experiment employing an equidistant mesh lead to similar results. The diffusional mobility $b(u) := \max(1-u^2, 0)$ is discretized explicitly in time, i.e. we use the corresponding primal-dual active set method (mPDAS-II), where we use either the direct solver or a conjugate gradient method for the solution of the saddle point system in two and three dimensions respectively.

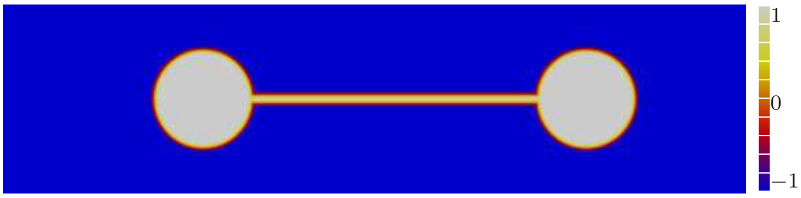


Figure 6.17: Initial data describing a dumbbell in two spatial dimensions for $\varepsilon = \frac{0.1}{\pi}$.

The evolution process governed by surface diffusion, leads to a pinch-off in finite time for initial configurations like the dumbbell given here, compare e.g. Bänsch, Morin and Nochetto, see [BMN05].

In Figure 6.18 the different time steps of the evolution using an explicit discretization of the free energy term are depicted in each row from left to right. Each row shows the evolution process for a different time step size. The same simulations with an implicit discretization of the free energy term leads to a better approximation for larger time step sizes than the explicit version, see Figure 6.19, where the number of droplets in the numerical steady state varies. Note that the first two pictures of the upper row are missing due to the time step size of $\tau = 10^{-5}$, i.e. the depicted image is the state after one time step.

The number of necessary primal-dual active set method iterations stayed below 6 for the explicit discretization. In case of the largest time step, i.e. $\tau = 10^{-5}$, an average of four iterations was sufficient. However the implicit discretization behaves differently in the first time step, mostly due to the significant change which occurs there. For the time step size $\tau = 10^{-7}$, and smaller sizes, the number of primal-dual active set iterations is the same as with the explicit discretization.

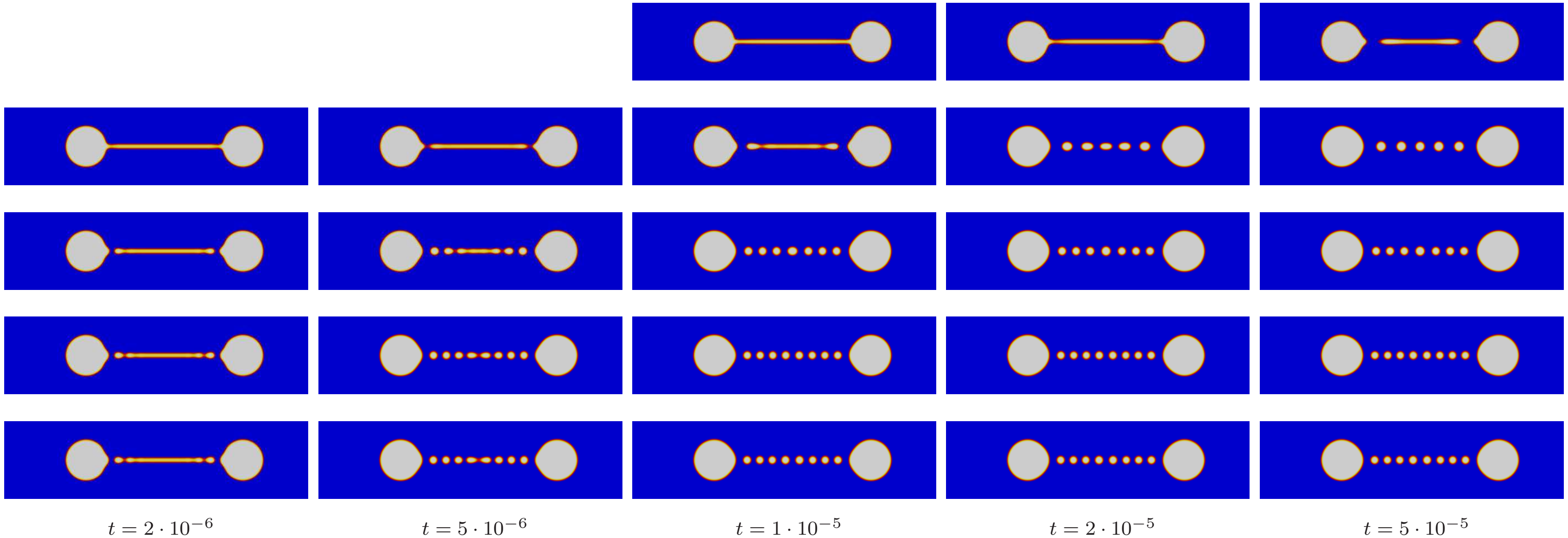


Figure 6.18: Evolution of a two dimensional dumbbell using an explicit time discretization of the free energy for different time step sizes $\tau = 10^{-5}$, 10^{-6}, 10^{-7}, 10^{-8} and 10^{-9} from top to bottom row.

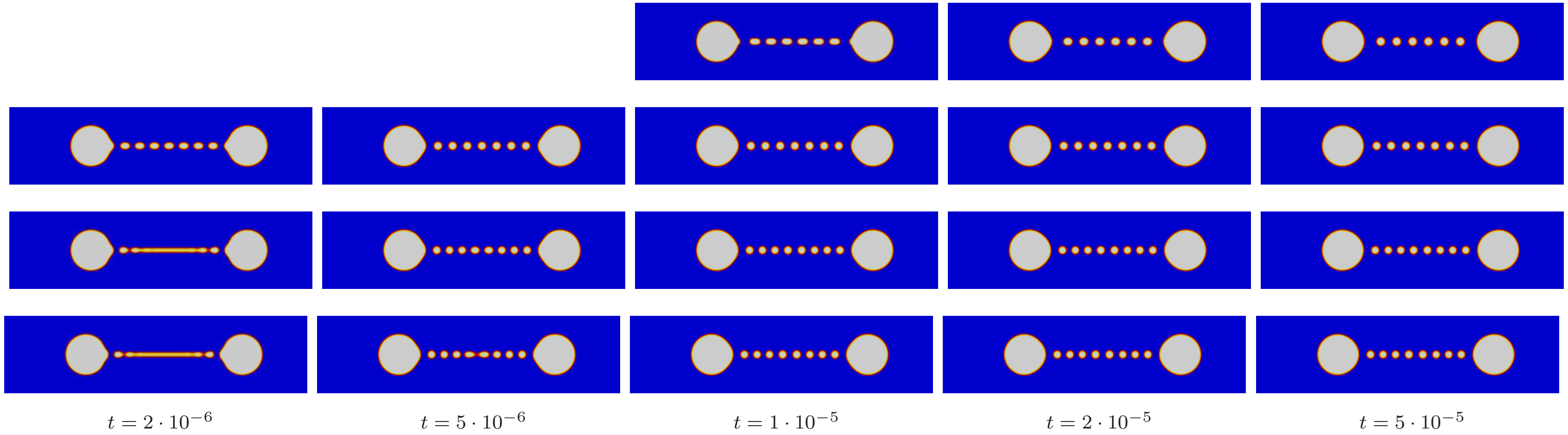
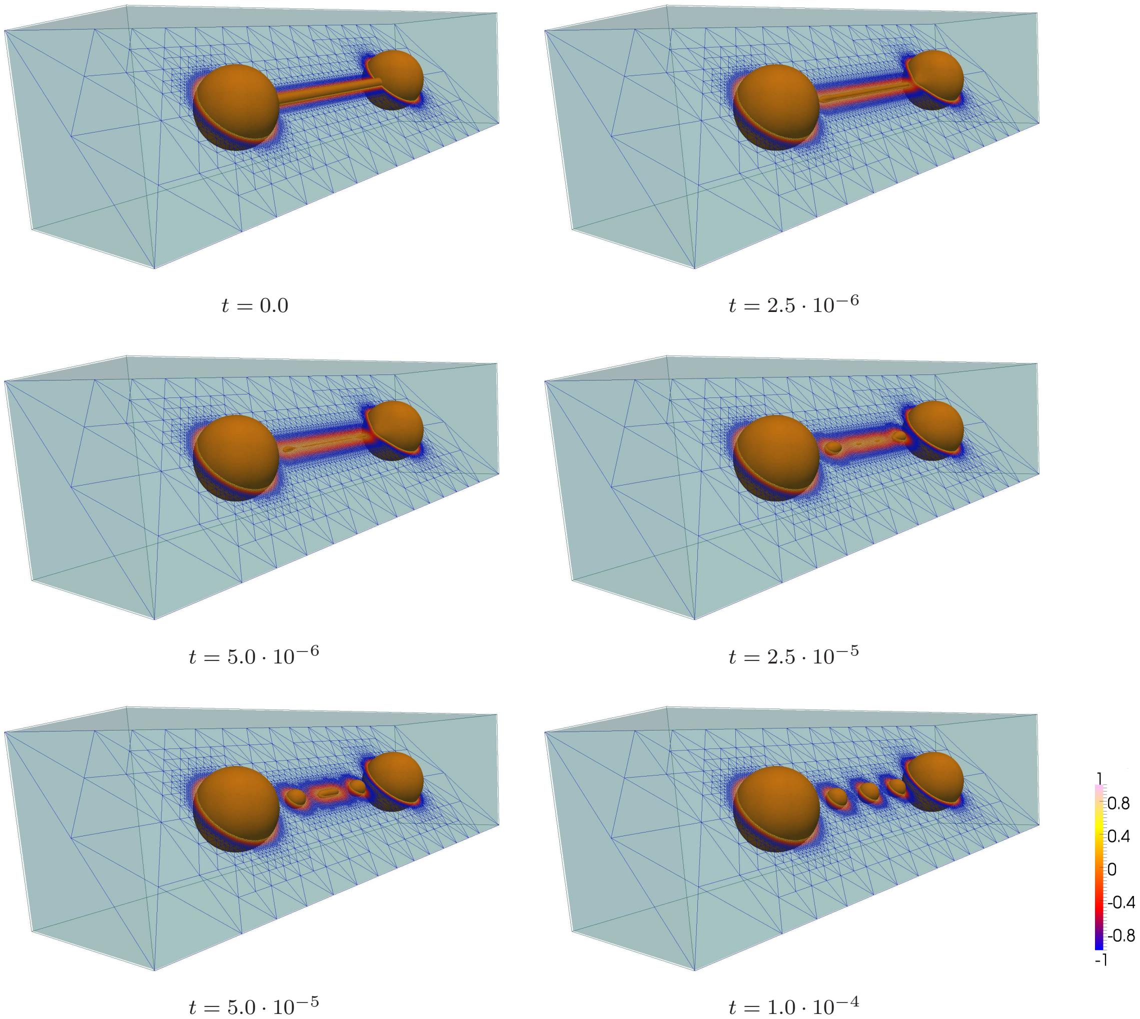


Figure 6.19: Evolution of a two dimensional dumbbell using an implicit time discretization of the free energy for different time step sizes $\tau = 10^{-5}$, 10^{-6}, 10^{-7} and 10^{-8} from top to bottom row.

The simulation with $\tau = 10^{-6}$ requires 27 and with $\tau = 10^{-5}$ 121 iterations in the first time step. After that 5 primal-dual active set iterations per time step suffice. This is probably related to the uniqueness result we have, where we require a bound on the time step size such that $\tau \in (0, 4\varepsilon^4) \approx (0, 4 \cdot 10^{-6})$, see Lemma 2.12 or Corollary 3.8. The behavior does not continually worsen, for example, 38 iterations are sufficient for a time step size of $\tau = 5 \cdot 10^{-5}$.
Extending the above result to three spatial dimensions we get similar results. We use the same parameters as before, with the exception of the the diameter of the dumbbells handle, which was set to 0.08 leading to better visible results. Figure 6.20 shows the evolution of the 0–level set of the simulation with $\tau = 5 \cdot 10^{-7}$. Again the discretization of the free energy with an implicit method, leads to the correct steady state with a larger time step size in comparison to the explicit method, compare Figure 6.21, showing the numerically steady states for different time step sizes for explicit and



$t = 0.0$

$t = 2.5 \cdot 10^{-6}$

$t = 5.0 \cdot 10^{-6}$

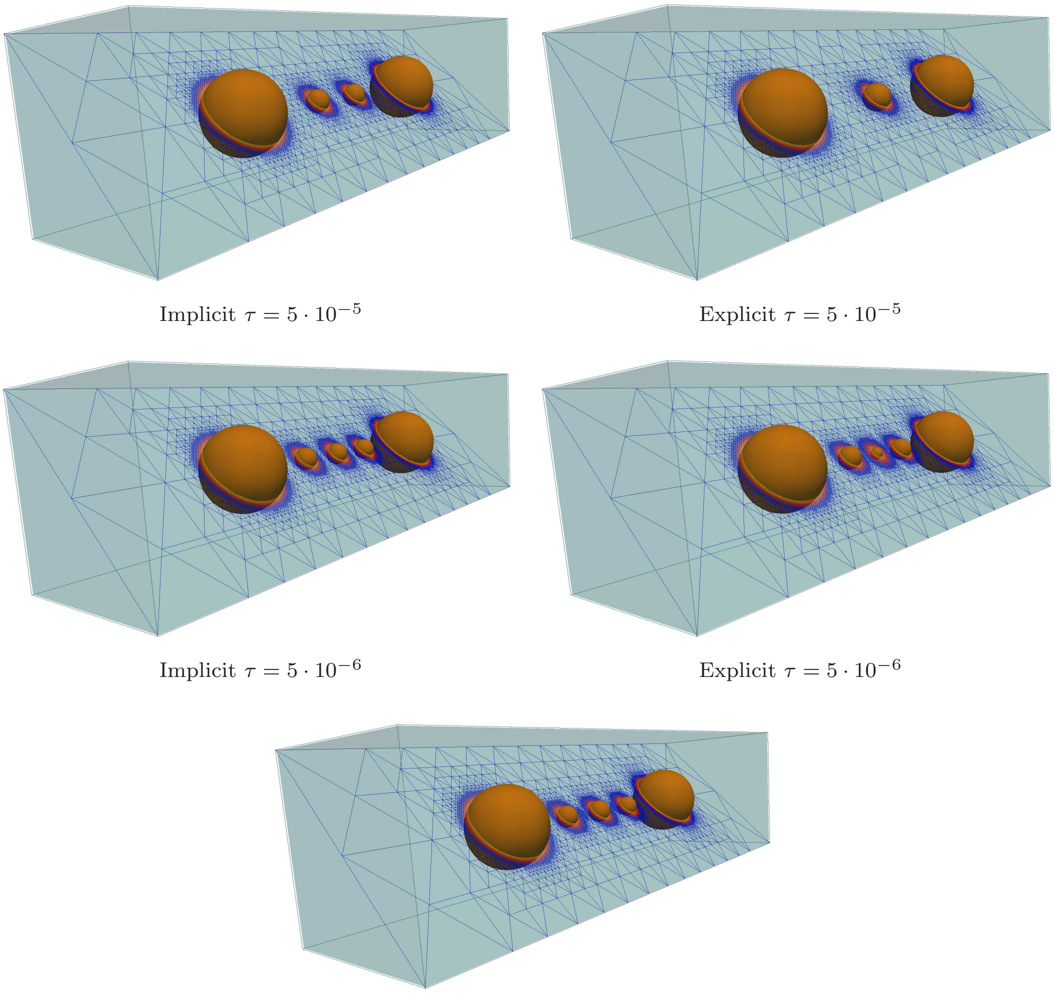
$t = 2.5 \cdot 10^{-5}$

$t = 5.0 \cdot 10^{-5}$

$t = 1.0 \cdot 10^{-4}$

Figure 6.20: Evolution of a three dimensional dumbbell using an implicit time discretization of the free energy with a time step size of $\tau = 5 \cdot 10^{-7}$.

implicit discretization of the free energy term. The existence and uniqueness theory for solutions with an explicitly discretized free energy require no restriction of the time step size, but from this experimental point of view the error of approximation of the model or more precisely the differences in the final states for larger time steps require a bound on the time step size here too.



Implicit $\tau = 5 \cdot 10^{-5}$

Explicit $\tau = 5 \cdot 10^{-5}$

Implicit $\tau = 5 \cdot 10^{-6}$

Explicit $\tau = 5 \cdot 10^{-6}$

Implicit $\tau = 5 \cdot 10^{-7}$

Figure 6.21: Final states of a three dimensional dumbbell with explicit and implicit time discretization of the free energy with different time step sizes.

Acknowledgments

My deepest gratitude is to my advisor, Prof. Dr. Harald Garcke, who has always given me his constant support and ready advice.

I am grateful for the opportunity to work with Prof. Dr. Luise Blank, who supported this thesis with insightful ideas and constructive comments.

The joint publication with Prof. Dr. Luise Blank and Prof. Dr. Harald Garcke helped develop this thesis very much.

I am indebted to Prof. Dr. Eberhard Bänsch for his comments and ideas on the topic of preconditioning and showing me the fun in math.

I am also grateful to my colleague, Dr. Daniel Depner, for the constant proof-reading and all the interesting discussions we have.

I would like to acknowledge Dr. Vanessa Styles, Dr. Lavinia Sarbu and Dr. Hassan Farshbaf-Shaker for our uncountable conversations on various topics.

Finally, I would like to thank all my friends and family for putting up with me during this time.

List of Figures

List of Algorithms

Bibliography

[AF03] Robert A. Adams and John J.F. Fournier, *Sobolev spaces*, Pure and Applied Mathematics, vol. 140, Elsevier, 2003.

[AKK10] Dimitra C. Antonopoulou, Georgia D. Karali, and George T. Kossioris, *Asymptotics for a generalized Cahn-Hilliard equation with forcing terms*, Discrete Contin. Dyn. Syst. A **30** (2010), 1037–1054.

[BB97] John W. Barrett and James F. Blowey, *Finite element approximation of a model for phase separation of a multi-component alloy with non-smooth free energy*, Numer. Math. **77** (1997), 1–34.

[BBG99] John W. Barrett, James F. Blowey, and Harald Garcke, *Finite element approximation of the Cahn-Hilliard equation with degenerate mobility*, SIAM J. Num. Anal. **37** (1999), no. 1, 286–318.

[BBG11] Luise Blank, Martin Butz, and Harald Garcke, *Solving the Cahn-Hilliard variational inequality with a semi-smooth Newton method*, ESAIM: COCV **17** (2011), 931 – 954.

[BE91] James F. Blowey and Charles M. Elliott, *The Cahn-Hilliard gradient theory for phase separation with non-smooth free energy part 1*, European J. Appl. Math. **2** (1991), 233–280.

[BE92] ______, *The Cahn-Hilliard gradient theory for phase separation with non-smooth free energy part 2*, European J. Appl. Math. **3** (1992), 147–179.

[BE93] ______, *Curvature dependent phase boundary motion and parabolic double obstacle problems*, Degenerate diffusions (Wei-Ming Ni, ed.), The IMA volumes in mathematics and its applications, vol. 47, Springer NY, 1993, pp. 19–60.

[BE94] ______, *A phase field model with a double obstacle potential*, Motion by mean curvature (G. Buttazzo and A. Visintin, eds.), de Gruyter, 1994, pp. 1–22.

[BGL05] Michele Benzi, Gene H. Golub, and Jörg Liesen, *Numerical solution of saddle point problems*, Acta Numer. **14** (2005), 1–137.

[BGN07] John W. Barrett, Harald Garcke, and Robert Nürnberg, *A parametric finite element method for fourth order geometric evolution equations*, J. Comput. Phys. **222** (2007), no. 1, 441–467.

[BGSS09] Luise Blank, Harald Garcke, Lavinia Sarbu, and Vanessa Styles, *Primal-dual active set methods for Allen-Cahn variational inequalities with non-local constraints*, Preprint SPP1253-09-01 (2009), 1–29, to appear in Numer. Methods Partial Differential Equations.

[BIG04] Adi Ben-Israel and Thomas N.E. Greville, *Generalized Inverses Theory and Application*, 2nd ed., CMS Books, Springer, 2004.

[BL05] Pavel Bochev and Rich B. Lehoucq, *On the finite element solution of the pure Neumann problem*, SIAM Rev. **47** (2005), no. 1, 50–66.

[BMN05] Eberhard Bänsch, Pedro Morin, and Ricardo H. Nochetto, *A finite element method for surface diffusion: the parametric case*, J. Comput. Phys. **203** (2005), no. 1, 321–343.

[BMN10] Eberhard Bänsch, Pedro Morin, and Ricardo H. Nochetto, *Preconditioning a class of fourth order problems by operator splitting*, Numer. Math. (2010), 1–32.

[BN09] L'ubomír Banas and Robert Nürnberg, *A multigrid method for the Cahn-Hilliard equation with obstacle potential*, Applied Mathematics and Computation **213** (2009), no. 2, 290–303.

[BNS78] James R. Bunch, Christopher P. Nielsen, and Danny C. Sorensen, *Rank-one modification of the symmetric eigenproblem*, Numerische Mathematik **31** (1978), 31–48.

[BNS04] John W. Barrett, Robert Nürnberg, and Vanessa Styles, *Finite element approximation of a phase field model for void electromigration*, SIAM J. Numer. Anal. **42** (2004), no. 2, 738–772.

[Bra07] Dietrich Braess, *Finite elements: theory, fast solvers, and applications in solid mechanics*, 3rd ed., Cambridge Univ. Press, 2007.

[BS08] Susanne C. Brenner and L. Ridgway Scott, *The mathematical theory of finite element methods*, 3rd ed., Texts in applied mathematics, no. 15, Springer, NY, 2008.

[Cah59] John W. Cahn, *Free energy of a nonuniform system. II. Thermodynamic basis*, J. Chem. Phys. **30** (1959), no. 5, 1121–1124.

[Cah61] ———, *On spinodal decomposition*, Acta Metallurgica **9** (1961), no. 9, 795–801.

[CC98] Gunduz Caginalp and Xinfu Chen, *Convergence of the phase field model to its sharp interface limits*, European J. Appl. Math. **9** (1998), 417–445.

[CENC96] John W. Cahn, Charles M. Elliott, and Amy Novick-Cohen, *The Cahn-Hilliard equation with a concentration dependent mobility: motion by minus the laplacian of the mean curvature*, European J. Appl. Math. **7** (1996), no. 3, 287–301.

[CG73] Jean Céa and Roland Glowinski, *Sur des méthodes d'optimasion par relaxation*, RAIRO R-3 (1973), 5–32.

[CGS77] Richard W. Cottle, Gene H. Golub, and Richard S. Sacher, *On the solution of large, structured linear complementarity problems: The block partitioned case*, Applied Mathematics and Optimization **4** (1977), 347–363.

[CH58] John W. Cahn and John E. Hilliard, *Free energy of a nonuniform system. I. Interfacial energy*, J. Chem. Phys. **28** (1958), no. 2, 258–267.

[Che96] Xinfu Chen, *Global asymptotic limit of solutions of the Cahn-Hilliard equation*, J. Differential Geom. **44** (1996), no. 2, 262–311.

[CNQ01] Xiaojun Chen, Zuhair Nashed, and Liqun Qi, *Smoothing methods and semismooth methods for nondifferentiable operator equations*, SIAM J. Numer. Anal. **38** (2001), no. 4, 1200–1216.

[Cot79] Richard Cottle, *Numerical methods for complementarity problems in engineering and applied science*, Computing Methods in Applied Sciences and Engineering, 1977, I (R. Glowinski, J. Lions, and Iria Laboria, eds.), Lecture Notes in Mathematics, vol. 704, Springer Berlin / Heidelberg, 1979, pp. 37–52.

[Cry71a] Colin W. Cryer, *The method of Christopherson for solving free boundary problems for infinite journal bearings by means of finite differences*, Math. Comp. **25** (1971), no. 115, 435–443.

[Cry71b] ______, *The solution of a quadratic programming problem using systematic overrelaxation*, SIAM J. Control **9** (1971), no. 3, 385–392.

[Dav04a] Timothy A. Davis, *Algorithm 832: UMFpack v4.3—an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Softw. **30** (2004), no. 2, 196–199.

[Dav04b] ______, *A column pre-ordering strategy for the unsymmetric-pattern multifrontal method*, ACM Trans. Math. Softw. **30** (2004), 165–195.

[Dav06a] ______, *Direct Methods for Sparse Linear Systems*, Fundamentals of Algorithms, SIAM, 2006.

[Dav06b] ______, *Summary of available software for sparse direct methods*, 2006.

[DD97] Timothy A. Davis and Iain S. Duff, *An unsymmetric-pattern multifrontal method for sparse LU factorization*, SIAM J. Matrix Anal. Appl. **18** (1997), no. 1, 140–158.

[DD99] ______, *A combined unifrontal/multifrontal method for unsymmetric sparse matrices*, ACM Trans. Math. Softw. **25** (1999), no. 1, 1–20.

[DDE05] Klaus Deckelnick, Gerhard Dziuk, and Charles M. Elliott, *Computation of geometric partial differential equations and mean curvature flow*, Acta Numerica **14** (2005), 139–232.

[DLFK96] Tecla De Luca, Francisco Facchinei, and Christian Kanzow, *A semismooth equation approach to the solution of nonlinear complementarity problems*, Mathematical Programming **75** (1996), 407–439.

[DR83] Iain S. Duff and John K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Softw. **9** (1983), no. 3, 302–325.

[DRMN79] Iain S. Duff, John K. Reid, N. Munksgaard, and Hans B. Nielsen, *Direct solution of sets of linear equations whose matrix is sparse, symmetric and indefinite*, IMA J Appl Math **23** (1979), no. 2, 235–250.

[DVM02] Italo C. Dolcetta, Stefano F. Vita, and Riccardo March, *Area preserving curve shortening flows: From phase separation to image processing*, Interfaces and Free Boundaries **4** (2002), 325–434.

[EG96] Charles M. Elliott and Harald Garcke, *On the Cahn-Hilliard equation with degenerate mobility*, SIAM J. Math. Anal. **27** (1996), no. 2, 404–423.

[EG97] ______, *Existence results for diffusive surface motion laws*, Adv. Math. Sci. Appl. **7** (1997), no. 1, 465–488.

[EGK08] Christof Eck, Harald Garcke, and Peter Knabner, *Mathematische Modellierung*, Springer, 2008.

[Ell89] Charles M. Elliott, *The Cahn-Hilliard model for the kinetics of phase separation*, Int. Ser. of Numerical Math. (José-Francisco Rodrigues, ed.), vol. 88, Birkhäuser Basel, 1989, pp. 35–73.

[Eva10] Lawrence C. Evans, *Partial differential equations*, 2. ed. ed., Graduate studies in mathematics, vol. 19, American Math. Soc., Providence, RI, 2010.

[FFK98] Francisco Facchinei, Andreas Fischer, and Christian Kanzow, *Regularity properties of a semismooth reformulation of variational inequalities*, SIAM J. Optim. **8** (1998), no. 3, 850–869.

[FP03a] Francisco Facchinei and Jong-Shi Pang, *Finite-dimensional variational inequalities and complementarity problems volume I*, Springer Series in Operations Research, Springer New York, 2003.

[FP03b] ______, *Finite-dimensional variational inequalities and complementarity problems volume II*, Springer Series in Operations Research, Springer New York, 2003.

[Fri82] Avner Friedman, *Variational principles and free-boundary problems*, Pure and Applied Mathematics, John Wiley & Sons Inc., New York, 1982.

[Gar05] Harald Garcke, *Mechanical effects in the Cahn–Hilliard model: A review on mathematical results*, Mathematical Methods and Models in Phase Transitions (Alain Miranville, ed.), Nova Science Publishers, New York, 2005.

[GD83] James D. Gunton and Michel Droz, *Introduction to the theory of metastable and unstable states*, Lecture Notes in Physics, no. 183, Springer, 1983.

[GK02] Carl Geiger and Christian Kanzow, *Theorie und numerik restringierter optimierungsaufgaben*, Springer, 2002.

[GK07] Carsten Gräser and Ralf Kornhuber, *On preconditioned Uzawa-type iterations for a saddle point problem with inequality constraints*, Domain Decomposition Methods in Science and Engineering XVI (O. D. Widlund and D. Keyes, eds.), LNCSE, vol. 55, Springer, 2007, pp. 91–102.

[GK09] ______, *Nonsmooth Newton methods for set-valued saddle point problems*, SIAM J. Numer. Anal. **47** (2009), 1251–1273.

[Gre97] Anne Greenbaum, *Iterative methods for solving linear systems*, Frontiers in applied mathematics, vol. 17, SIAM, 1997.

[Gro00] Jürgen Groß, *The Moore-Penrose inverse of a partitioned nonnegative definite matrix*, Linear Algebra and its applications **321** (2000), no. 1–3, 113–121.

[GSH07] Nicholas I.M. Gould, Jennifer A. Scott, and Yifan Hu, *A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations*, ACM Trans. Math. Softw. **33** (2007), no. 2, 23.

[Gup02] Anshul Gupta, *Recent advances in direct methods for solving unsymmetric sparse systems of linear equations*, ACM Trans. Math. Softw. **28** (2002), no. 3, 301–324.

[GY99] Gene H. Golub and Quiang Ye, *Inexact preconditioned conjugate gradient method with inner-outer iteration*, SIAM J. Sci. Comput. **21** (1999), no. 4, 1305–1320.

[Har59] Edward W. Hart, *Thermodynamics of inhomogeneous systems*, Phys. Rev. **113** (1959), no. 2, 412–416.

[HHT10] Michael Hintermüller, Michael Hinze, and Mohammed H. Tber, *An adaptive finite element Moreau-Yoshida-based solver for a non-smooth Cahn-Hilliard problem*, Hamburger Beiträge zur Angewandten Mathematik 2010-02 (to appear in OMS) (2010), 34.

[HIK02] Michael Hintermüller, Kazufumi Ito, and Karl Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim. **13** (2002), no. 3, 865–888.

[Hil54a] Clifford Hildreth, *Point estimates of ordinates of concave functions*, J. Amer. Statist. Assoc. **49** (1954), no. 267, 598–619.

[Hil54b] ______, *A quadratic programming procedure*, Naval Res. Logist. Quart. **4** (1954), no. 1, 79–85.

[HP90] Patrick T. Harker and Jong-Shi Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Mathematical Programming **48** (1990), 161–220.

[IK03] Kazufumi Ito and Karl Kunisch, *Semi-smooth Newton methods for variational inequalities of the first kind*, ESAIM: Mathematical Modelling and Numerical Analysis **37** (2003), no. 1, 41–62.

[Iro70] Bruce M. Irons, *A frontal solution program for finite element analysis*, Int. J. Numer. Methods Eng. **2** (1970), no. 1, 5–32.

[JCG09] Carlo Janna, Andrea Comerlati, and Giuseppe Gambolati, *A comparison of projective and direct solvers for finite elements in elastostatics*, Adv. Eng. Softw. **40** (2009), no. 8, 675–685.

[Kar71] Stepan Karamardian, *Generalized complementarity problem*, J. Optim. Theory Appl. **8** (1971), 161–168.

[Kor94] Ralf Kornhuber, *Monotone multigrid methods for variational inequalities I*, Numer. Math. **69** (1994), 167–184.

[Kor96] ______, *Monotone multigrid methods for variational inequalities II*, Numer. Math. **72** (1996), 481–499.

[KS80] David Kinderlehrer and Guido Stampacchia, *An introduction to variational inequalities and their applications*, Pure and Applied Mathematics, vol. 88, Academic Press, 1980.

[KS07] Ellen Kuhl and Daniel Schmid, *Computational modeling of mineral unmixing and growth*, Computational Mechanics **39** (2007), 439–451.

[KS08] Evgeniy Khain and Leonard M. Sander, *Generalized Cahn-Hilliard equation for biological applications*, Phys. Rev. E **77** (2008), no. 5, 051129.

[KW06] David Kay and Richard Welford, *A multigrid finite element solver for the Cahn-Hilliard equation*, J. Comput. Phys. **212** (2006), 288–304.

[Liu92] Joseph W.H. Liu, *The multifrontal method for sparse matrix solution: Theory and practice*, SIAM Review **34** (1992), no. 1, 82–109.

[LS61] Il'ya M. Lifshitz and Vasili V. Slyozov, *The kinetics of precipitation from supersaturated solid solutions*, J. Phys. Chem. Solids **19** (1961), no. 1-2, 35–50.

[LT85] Peter Lancaster and Miron Tismentsky, *The Theory of Matrices: Second Edition with Applications*, Academic Press, 1985.

[LT98] John S. Lowengrub and L. Truskinovsky, *Quasi-incompressible Cahn-Hilliard fluids and topological transitions*, R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci **454** (1998), no. 1987, 2617–2654.

[Mei05] Andreas Meister, *Numerik linearer Gleichungssysteme*, 2nd ed., Vieweg, 2005.

[Moh91] Bojan Mohar, *The Laplacian spectrum of graphs*, Graph theory combinatorics and applications **2** (1991), no. 6, 871–898.

[Moo20] E. H. Moore, *On the reciprocal of the general algebraic matrix*, Bulletin of the American Mathematical Society **26** (1920), no. 9, 394–395.

[Mul57] William W. Mullins, *Theory of thermal grooving*, J. Appl. Phys. **28** (1957), 333–339.

[NC98] Amy Novick-Cohen, *The Cahn-Hilliard quation: mathematical and modeling perspectives*, Adv. Math. Sci. Appl. **8** (1998), no. 2, 965–985.

[NCon] ______, *The Cahn-Hilliard equation: From backwards diffusion to surface diffusion*, Cambridge University Press, in preparation.

[NW06] Jorge Nocedal and Stephen J. Wright, *Numerical optimization*, 2nd ed., Springer series in operations research and financial engineering, Springer, 2006.

[OP88] Yoshitsugu Oono and Sanjay Puri, *Study of phase-separation dynamics by use of cell dynamical systems. i. modeling*, Phys. Rev. A **38** (1988), no. 1, 434–453.

[Ost97] Wilhelm Ostwald, *Lehrbuch der allgemeinen Chemie*, vol. 2, Engelmann, 1897.

[Ost01] ______, *Beziehungen zwischen Oberfächenspannung und Löslichkeit*, Z. Phys. Chem. **37** (1901), 385.

[PBD92] Sanjay Puri, Kurt Binder, and Sushanta Dattagupta, *Dynamical scaling in anisotropic phase-separating systems in a gravitational field*, Phys. Rev. B **46** (1992), no. 1, 98–105.

[Peg89] Robert L. Pego, *Front migration in the nonlinear Cahn-Hillard equation*, Proc. Roy. Soc. London Ser. A **422** (1989), no. 1863, 261–278.

[Pen55] Roger Penrose, *A generalized inverse for matrices*, Mathematical Proceedings of the Cambridge Philosophical Society **51** (1955), no. 3, 406–413.

[QS93] Liqun Qi and Jie Sun, *A nonsmooth version of Newton's method*, Math. Program. **58** (1993), no. 3, 353–367.

[Rho65] Charles A. Rhode, *Generalized inverses of partitioned matrices*, SIAM **13** (1965), no. 4, 1033–1035.

[RV01] Lorenz Ratke and Peter W. Voorhees, *Growth and coarsening*, Engineering Materials, Springer Verlag, Heidelberg, 2001.

[SO66] James E. Scroggs and Patrick L. Odell, *An alternate definition of a pseudoinverse of a matrix*, SIAM J. Appl. Math. **14** (1966), no. 4, 796–810.

[Spi09] Dan Spielman, *Spectral graph theory*, Lecture notes, published online at `http://www.cs.yale.edu/homes/spielman/561/lect02-09.pdf`, 2009.

[SS05] Alfred Schmidt and Kunibert G. Siebert, *Design of adaptive finite element software: The finite element toolbox ALBERTA*, Lecture Notes in Computational Science and Engineering, no. 42, Springer Berlin Heidelberg, 2005.

[Sto96] Barbara E.E. Stoth, *Convergence of the Cahn-Hilliard equation to the Mullins-Sekerka problem in spherical symmetry*, J. Differential Equations **125** (1996), no. 1, 154–183.

[TC94] Jean E. Taylor and John W. Cahn, *Linking anisotropic sharp and diffuse surface moption laws via gradient flows*, J. Statist. Phys. **77** (1994), no. 1/2, 183–197.

[Tho06] Vidar Thomée, *Galerkin finite element methods for parabolic problems*, 2nd ed., Springer Series in Computational Mathematics, no. 25, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Tre03] Scott Tremaine, *On the origin of irregular structure in Saturn's rings*, The Astronomical Journal **125** (2003), no. 2, 894.

[Trö10] Fredi Tröltzsch, *Optimal control of partial differential equations, theory, methods and applications*, Graduate Studies in Mathematics, vol. 112, AMS, 2010.

[VdW79] Johannes D. Van der Waals, *The thermodynamic theory of capillarity under the hypothesis of a continuous variation of density*, J. Statist. Phys. **20** (1979), no. 2, 200–244.

[Voo85] Peter W. Voorhees, *The theory of Ostwald ripening*, J. Statist. Phys. **38** (1985), no. 1-2, 231–252.

[Wag61] Carl Wagner, *Theorie der Alterung von Niederschlägen durch Umlösen (Ostwald-Reifung)*, Z. Elektrochemie **65** (1961), 581–591.

[Yin92] Jingxue Yin, *On the existence of nonnegative continuous solutions of the Cahn-Hilliard equation*, J. Differential Equations (1992), no. 2, 310–327.

[Zei85] Eberhard Zeidler, *Nonlinear functional analysis and its applications*, vol. III, Springer, New York, 1985.

[Zha05] Fuzhen Zhang, *The Schur complement and its applications*, Springer, NY, 2005.

[ZW07] Shiwei Zhou and Michael Y. Wang, *Multimaterial structural topology optimization with a generalized Cahn-Hilliard model of multiphase transition*, Structural and Multidisciplinary Optimization **33** (2007), 89–111.