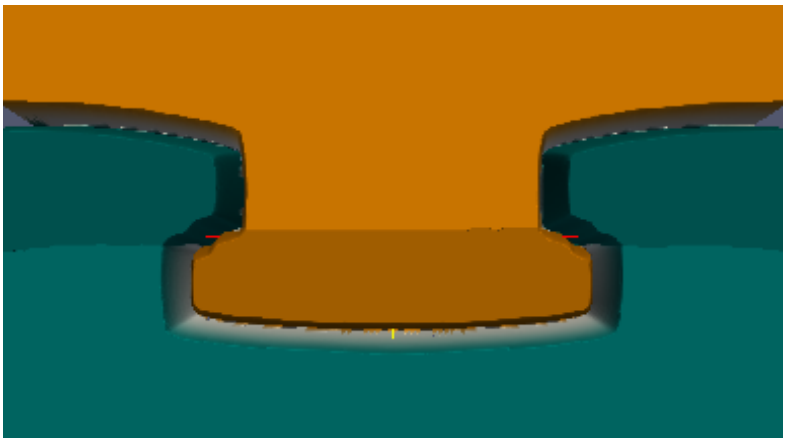
Master Thesis

Diffusion Induced Anisotropic Grain Boundary Motion



by Lisa Julia Nebel

Matriculation number: 4362086

First corrector: Prof. Dr. Ralf Kornhuber
Second corrector: Dr. Carsten Gräser
Fachbereich Mathematik und Informatik, Freie Universität Berlin

29th of November 2013

Für Burkhard Nebel.

Contents

1 Introduction

Diffusion induced anisotropic grain boundary motion (DIAGM) treats the motion of several grains that is caused by the diffusion of solute along the grain boundaries. The boundaries move according to the L^2-gradient-flow of the Ginzburg–Landau-Energy, containing an anisotropy function, an "asymmetric norm". As explained in [BP, p.537], the effects of the anisotropy function are highly relevant when examining crystalline structures, for instance when steaming solute onto a metal plate with differently oriented grains inside.
The model using the eucledian norm within the Ginzburg–Landau-Energy, is referred to as isotropic and the corresponding phase field model is studied in [GS] and in [KKSG]. The anisotropic phase field model and its numerical solution are treated here.
In the first part of this thesis, properties of anisotropy functions together with some examples are covered, similar to [GKS,T]. Then, the phase field model arising from the L^2-gradient-flow of the Ginzburg–Landau-Energy is presented together with the differential equations describing the movement of the solute.
In the second part the equations are discretized in time using Rothe's method. The resulting variational equations for the phase field function and the solute function are equivalent to minimization problems. In order to prove existence and uniqueness of the solution of the minimization problem for the phase field function, theorems from convex analysis are used. Then we discretize the equations in space and prove the existence and uniqueness of the solution of the minimization problem for the solute function.
The next part of this thesis covers the algebraic formulation of the discretized minimization problems and numerical methods to solve them. The minimization problem we obtain from the L^2-gradient-flow involves simplex constraints and a nonlinear part reusulting from the anisotropy function. To solve this minimization problem numerically, special subspace correction methods for nonlinear problems can be used. In particular we choose the truncated nonsmooth newton mulitgrid method (TNNMG) from [Gr], where the actual problem is just solved on the finest grid and on the coarser grids, a linearization is considered instead.
In the last part of this thesis, numerical experiments in 2D and 3D with several initial functions, different anisotropy functions and different numbers of grains are shown. Finally we compare the performance of the TNNMG method for the isotropic model with the performance for the anisotropic model.

At this point I would like to thank Carsten Gräser for his patience and his helpful support.

2 Modelling of DIAGM

In this chapter we will discuss some properties of anisotropy functions. We will need the results later for the examination of the Ginzburg–Landau-Energy.

2.1 Minkowski norms

A Minkowski Norm is not a norm in the classical sense, it is an "asymmetric norm". In contrast to the eucledian, symmetric norm, an "asymmetric norm" is not onehomogeneous.

Definition 2.1.1 (Minkowski Norm)**.** *A continuous function $\beta : \mathbb{R}^d \to [0, +\infty[$ is called a Minkowski Norm if :*

i) γ *is positively homogeneous of degree one, i.e.* $\gamma(\lambda x) = \lambda\gamma(x), \forall\lambda \geq 0$

ii) $\gamma(v + w) \leq \gamma(v) + \gamma(w)$

[BP, p.541]

In this paper we will work with special Minkowski Norms, namely anisotropy functions. Only admissible anisotropy functions will be considered.

Definition 2.1.2 (admissible)**.** *An anisotropy function $\gamma : \mathbb{R}^d \to \mathbb{R}$ is called admissible if:*

i) γ *is a Minkowski Norm*

ii) $\gamma \in C^2(\mathbb{R}^d \setminus \{0\})$, $\gamma(x) > 0$ *for* $x \in \mathbb{R}^d \setminus \{0\}$

iii) $\exists\gamma_0 > 0$ *such that* $\langle\gamma''(x)y, y\rangle \geq \gamma_0|y|^2$ $\forall x, y \in \mathbb{R}^d$, $|x| = 1$, $\langle x, y\rangle = 0$

[DDE, p.52] [GKS,T, p.3]

As γ does not have to be onehomogeneous, for some $\lambda \in \mathbb{R}$ and some $x \in \mathbb{R}^d$ the following may hold: $\gamma(-\lambda x) \neq \gamma(\lambda x)$. This property can later be seen in Example 2.1.4, the twisted l^r-anisotropy, in Example 2.1.5, the weighted anisotropy and in Example 2.1.7, the anisotropy function introduced by Kobayashi.
Later, in Lemma 2.1.10 we will see that the property iii) from Definition 2.1.2 actually holds for all $x, y \in \mathbb{R}^d, x \neq 0$, but it is enough to require it for $|x| = 1$, $\langle x, y\rangle = 0$.
Here and throughout this thesis, $\langle\cdot\,,\cdot\rangle$ denotes the eucledian scalar product and $|\cdot|$ denotes the eucledian norm.
Anisotropy functions can be visualized using Franck diagrams $\mathcal{F}$ and Wulff shapes $\mathcal{W}$, the unit balls of the function and its dual. They are given by

$$\mathcal{F} = \left\{x \in \mathbb{R}^d \,|\, \gamma(x) \leq 1\right\} \quad \text{and} \quad \mathcal{W} = \left\{x \in \mathbb{R}^d \,|\, \gamma^*(x) \leq 1\right\}$$

where

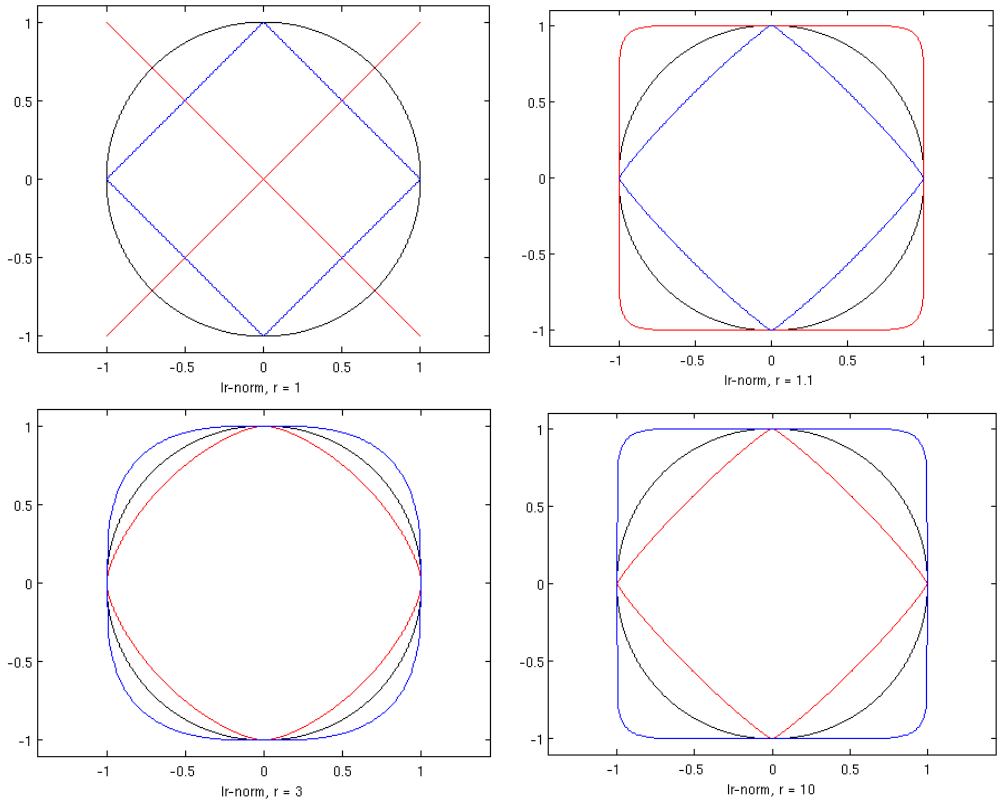
$$\gamma^*(x) = \sup_{y \in \mathbb{R}^d \setminus \{0\}} \frac{\langle y, x \rangle}{\gamma(y)}$$

is the dual of $\gamma(x)$.

We will now look at some examples of anisotropy functions. In all figures, the eucledian unit circle is painted in black.

Example 2.1.3 (Eucledian Norm)**.** $\gamma = |.|$
As already mentioned in the introduction, in this case our model is called **isotropic** or **Diffusion Induced Grain Boundary Motion**. $\mathcal{W}$ and $\mathcal{F}$ are both equal to the eucledian unit ball.

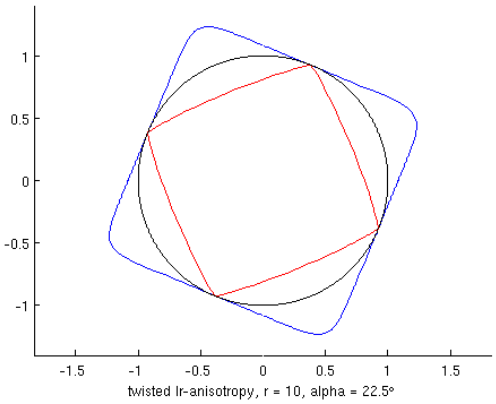
Example 2.1.4 (discrete l^r-norm)**.** $\gamma_r(x) = \left(\sum_{k=1}^d |x_k|^r \right)^{\frac{1}{r}}$
As $r \to \infty$, $\mathcal{F}$ turns into a cube and $\mathcal{W}$ turns into a diamond. $r = \infty$ corresponds to the uniform norm. As $r \searrow 1$, it is exactly the other way round, so $\mathcal{F}$ turns into a diamond and $\mathcal{W}$ turns into a cube.
The discrete l^r-norm is only admissible only for $r > 1$, because for $r < 1$, the triangle inequality is not fulfilled and for $r = 1$, γ_1 is not in $C^2(\mathbb{R}^d \setminus \{0\})$, as can be easily seen in the graph for $r = 1$.

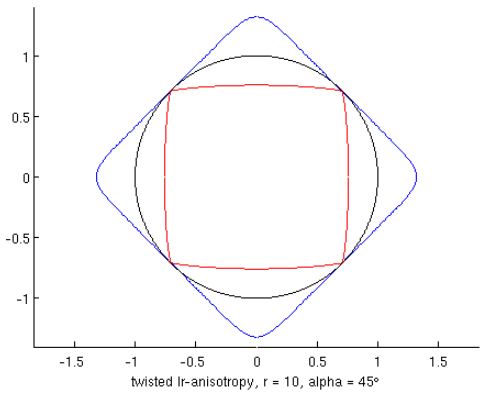


We can also twist the norm using a turning matrix M. Then $\gamma_{rM}(x) = \left(\sum_{k=1}^d |\widetilde{x}_k|^r \right)^{\frac{1}{r}}$,

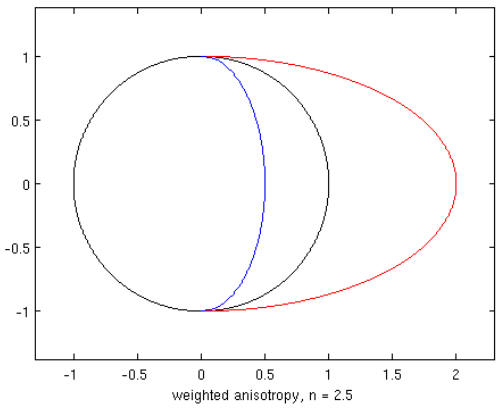
where $\widetilde{x} = Mx$. In the graphs below, M was chosen to turn the vector x through the angle α, so

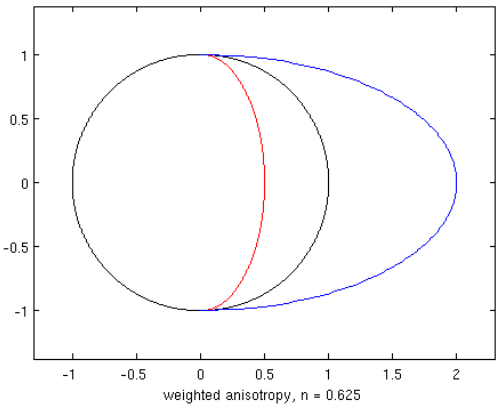
$$M = \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix}.$$



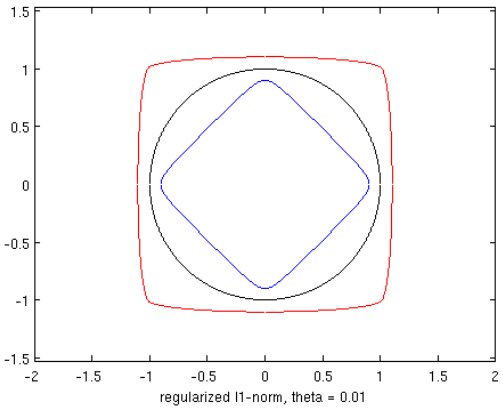


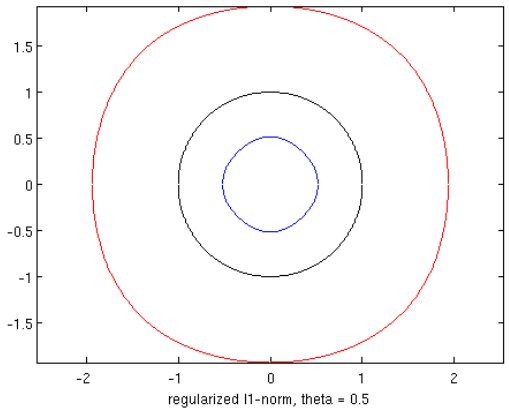
Example 2.1.5 (weighted anisotropy)**.** $\gamma_n(x) = \sqrt{\big(n + (n-1)\,\mathrm{sign}(x_1)\big)x_1^2 + \sum_{k=2}^{d} x_k^2}$
For $n \to \infty$, $\mathcal{F}$ gets very flat on the side where $x_1 > 0$ and $\mathcal{W}$ bulges out until $\sqrt{2n-1}$. For $n \searrow 0.5$, it is exactly the other way round, so $\mathcal{W}$ gets very flat on the side where $x_1 > 0$ and $\mathcal{F}$ bulges out until $\sqrt{\frac{1}{2n-1}}$. On the half where $x_1 < 0$, $\mathcal{F}$ and $\mathcal{W}$ are equal to the eucledian unit circle, as $\gamma_n(x)$ is equal to the eucledian norm for $x_1 < 0$.
It is easy to see that this anisotropy function is only admissible for $n > 0.5$, because for $n = 0.5$, γ_n is not in $C^2(\mathbb{R}^d \backslash \{0\}$ and for $n < 0.5$, the triangle inequality is not fulfilled.





Example 2.1.6 (regularized l^1-norm)**.** $\gamma_\theta(x) = \sum_{i=1}^{d} \sqrt{x_i^2 + \theta|x|^2}$
This norm is a modification of the l^1-norm. We can see that the plots for a small θ resemble the plots of the l^r-norm for $r = 1 + \epsilon$. γ_θ is admissible for all $\theta > 0$.





Example 2.1.7 (anisotropy function introduced by Kobayashi from [GKS,T, p.6])**.**

$$\gamma_{k,a} : \mathbb{R}^2 \to \mathbb{R}, \quad \gamma_{k,a}(x) = \begin{cases} (1 + a\, cos(k\beta(x)))\, |x| & \text{for } x \neq 0 \\ 0 & \text{for } x = 0 \end{cases}$$

where $k \geq 1$ and $\beta(x)$ is given as the angle between the vector x and the positve x_1-axis, so we have:

$$\beta(x) = \begin{cases} arctan\left(\frac{x_2}{x_1}\right) & \text{for } x_1, x_2 > 0 \\ \pi - arctan\left(\frac{x_2}{x_1}\right) & \text{for } x_1 < 0, x_2 > 0 \\ arctan\left(\frac{x_2}{x_1}\right) - \pi & \text{for } x_1, x_2 < 0 \\ 2\pi - arctan\left(\frac{x_2}{x_1}\right) & \text{for } x_1 > 0, x_2 < 0 \end{cases}$$

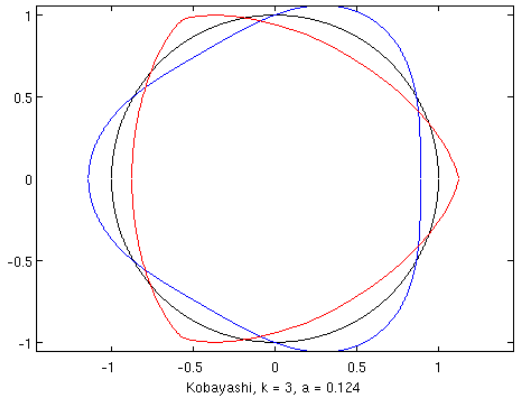
This anisotropy function is only admissible for $-\frac{1}{k^2-1} < a < \frac{1}{k^2-1}$ [BR, p.1142]. In the graphs, one can see that the triangle inequality is not satisfied if $a \geq \frac{1}{k^2-1}$ or $a \leq -\frac{1}{k^2-1}$. To check that $\gamma_{k,a}$ is not admissible, we will show that its Hessian is only positive definite if $-\frac{1}{k^2-1} a < \frac{1}{k^2-1}$. The Hessian of the anisotropy function $\gamma_{k,a} =: \gamma$ is given by

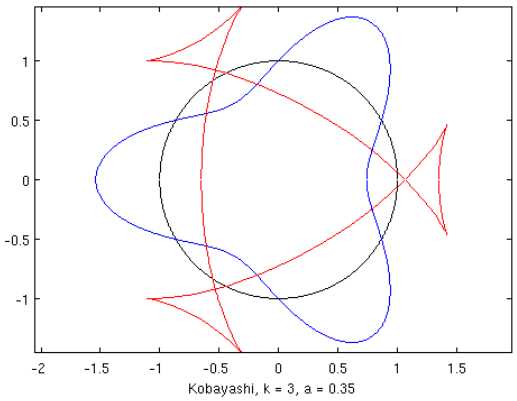
$$\gamma''(x) = \begin{pmatrix} \gamma_{x_1x_1} & \gamma_{x_1x_2} \\ \gamma_{x_1x_2} & \gamma_{x_2x_2} \end{pmatrix} = \begin{pmatrix} -x_2^2 & x_1x_2 \\ x_1x_2 & -x_1^2 \end{pmatrix} \frac{a(k^2-1)cos(k\beta(x)) - 1}{(x_1^2 + x_2^2)^{\frac{3}{2}}}$$

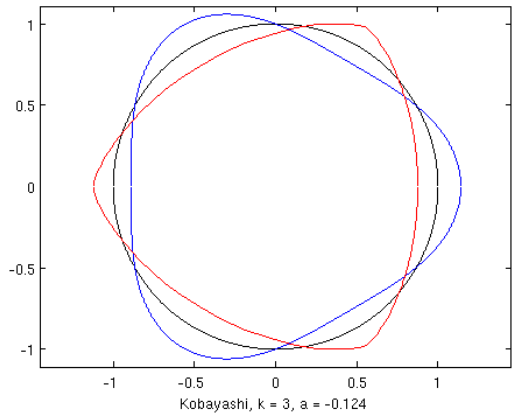
We want to determine when $\langle \gamma''(x)y, y \rangle > 0$ for all $y \neq 0$, so we look at

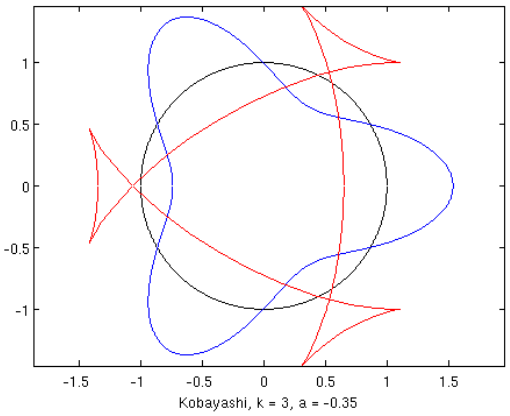
$$\langle \gamma''(x)y, y \rangle = -(x_1y_1 - x_2y_2)^2 \quad \frac{a(k^2-1)cos(k\beta(x)) - 1}{(x_1^2 + x_2^2)^{\frac{3}{2}}} > 0$$

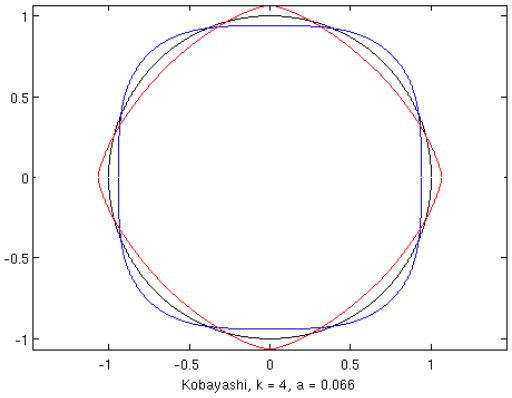
$$\Leftrightarrow a(k^2-1)cos(k\beta(x)) - 1 < 0.$$

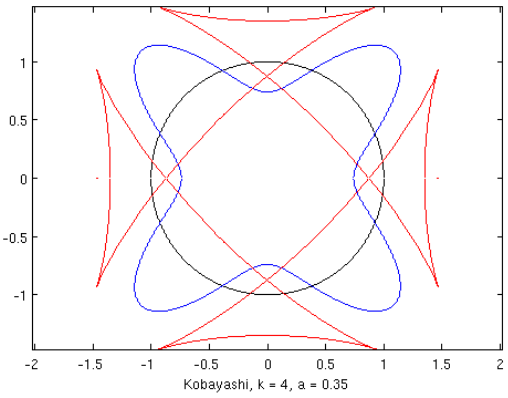

Kobayashi, k = 3, a = 0.124


Kobayashi, k = 3, a = 0.35


Kobayashi, k = 3, a = -0.124


Kobayashi, k = 3, a = -0.35


Kobayashi, k = 4, a = 0.066


Kobayashi, k = 4, a = 0.35

If $k = 1$, this inequality holds for all a.
For $k > 1$, we know that $cos(k\beta(x))$ is bounded by 1 and -1. Then we obtain

$$a(k^2-1) > -1 \Leftrightarrow a > \frac{-1}{k^2-1} \quad \text{and} \quad a(k^2-1) < 1 \Leftrightarrow a < \frac{1}{k^2-1}.$$

In the following we want to examine some properties of admissible anisotropy functions:

Lemma 2.1.8. *From Definition 2.1.1 ii) we get* $\forall x \in \mathbb{R}^d \backslash \{0\}$ *and* $\forall \lambda > 0$*:*

$$\langle \gamma'(x), x \rangle = \gamma(x), \quad \gamma''(x)\, x = 0 \in \mathbb{R}^d \tag{2.1.1}$$

$$\gamma'(\lambda x) = \gamma'(x), \quad \gamma''(\lambda x) = \frac{1}{\lambda}\gamma''(x) \tag{2.1.2}$$

[DDE, p.52]

Proof. Differentiate Definition 2.1.1 ii) with respect to λ to obtain (2.1.1) and with respect to x to obtain (2.1.2). □

Looking at (2.1.1) we see that x is an eigenvector of $\gamma''(x)$ to the eigenvalue 0, so it lies within the kernel of $\gamma''(x)$.

In the following we will show that the square γ^2 of an admissible anisotropy function is strictly convex. Later on, this property will be used to show existence and uniqueness of the solutions of several minimization problems. We will even show strong convexity in the following Theorem.

Theorem 2.1.9 (strong convexity)**.** *Let* $\gamma : \mathbb{R}^d \to [0, +\infty)$ *be an admissible anisotropy function. Then* γ^2 *is strongly convex, so*

$$\exists c > 0 : \ \gamma^2(x) - \gamma^2(y) \geq \langle (\gamma^2)'(y), x - y \rangle + c|x - y|^2. \tag{2.1.3}$$

To give a proof of Theorem 2.1.9, we need the following Lemmata 2.1.10 to 2.1.15.

Lemma 2.1.10.

$$\langle \gamma''(x)y, y \rangle \geq 0, \ \forall x, y \in \mathbb{R}^d, x \neq 0$$

Proof. From (2.1.2) we see that the matrix $\gamma''(x)$ does not depend on the length of the vector x (as long as $x \neq 0$).
Then from Definition 2.1.2 iii) we see that $\exists \gamma_0 > 0$ such that $\langle \gamma''(x)y, y \rangle \geq \gamma_0 |y|^2$ $\forall x, y \in \mathbb{R}^d$, $\langle x, y \rangle = 0$.
Now choose $y = a\, x + x^\perp$, where $a \in \mathbb{R}$ and $\langle x, x^\perp \rangle = 0$. We obtain (using (2.1.1) and that γ'' is symmetric, as it is a Hessian matrix)

$$\langle \gamma''(x)y, y \rangle = a\langle \gamma''(x)x, a\, x + x^\perp \rangle + \langle \gamma''(x)x^\perp, a\, x + x^\perp \rangle =$$

$$= 0 + \langle \gamma''(x)x^\perp, x^\perp \rangle + \langle x^\perp, \gamma''(x)\, a\, x \rangle = \langle \gamma''(x)x^\perp, x^\perp \rangle \geq 0.$$

□

Lemma 2.1.11 (continuous differentialbility)**.**

$$\gamma^2 \in C^1(\mathbb{R}^d) \quad \text{and} \quad \gamma^2 \in C^2(\mathbb{R}^d \backslash \{0\})$$

[GKS,T, p.3]

Proof. $(\gamma^2)''(x) = 2\gamma(x)\,\gamma''(x) + 2\gamma'(x)^T\gamma'(x)$ is continuous $\forall x \neq 0$, as $\gamma \in C^2(\mathbb{R}^d \backslash \{0\})$ and $\gamma \in C(\mathbb{R}^d)$.
$(\gamma^2(x))'$ is also continuous $\forall x \neq 0$. Now, we still have to show that $(\gamma^2(x))'$ is continuous at $x = 0$. We get $(\gamma^2(0))' = 0$ when we look at the difference quotient at $x = 0$:

$$\frac{\gamma^2(hx) - \gamma^2(0)}{h} = \frac{h^2\gamma^2(x) - 0}{h} = h\gamma^2(x) \underset{h \searrow 0}{\longrightarrow} 0$$

Also $(\gamma^2(x))' \underset{x \to 0}{\longrightarrow} 0$, as

$$|(\gamma^2(x))'| = 2|\gamma'(x)\gamma(x)| = 2|\gamma'(x)\gamma'(x) \cdot x| \leq 2|\gamma'(x) \cdot \gamma'(x)|\,|x| = C|x|$$

□

Lemma 2.1.12 (positive definiteness)**.** *The Hessian $(\gamma^2)''(x)$ is positive definite $\forall x \neq 0$.*
[GKS,T, p.4]

Proof. Take $y = a\,x + x^\perp$, where $\langle x, x^\perp \rangle = 0$.

Case 1: $x^\perp = 0$. Then we have:

$$\langle (\gamma^2)''(x)y, y\rangle = \big\langle \big(2\gamma(x)\,\gamma''(x) + 2\gamma'(x)^T\gamma'(x)\big)a\,x, a\,x\big\rangle =$$

$$= 2a^2\gamma(x)\langle \gamma''(x)x, x\rangle + 2a^2\langle \gamma'(x)x, \gamma'(x)x\rangle \overset{(2.1.1)}{=} 0 + 2a^2|\gamma'(x)x| > 0$$

Case 2: $x^\perp \neq 0$. Then we have:

$$\langle (\gamma^2)''(x)y, y\rangle = 2\langle \gamma(x)\gamma''(x)(ax + x^\perp), ax + x^\perp\rangle + 2\langle \gamma'(x)y, \gamma'(x)y\rangle >$$

$$> 2\gamma(x)\langle \gamma''(x)x^\perp, x^\perp\rangle \geq 2\gamma_0\gamma(x)\,|x^\perp|^2 > 0$$

□

So for $\forall x \neq 0$, the Hessian $(\gamma^2)''(x) \in R^{d\times d}$ is symmetric and positive definite. The Hessian is also bounded from above and below.

Lemma 2.1.13 (boundedness)**.** *There are constants $\mu, L > 0$ such that*

$$\mu|y|^2 \leq \langle (\gamma^2)''(x)y, y\rangle \leq L|y|^2 \quad \textit{holds for all } x, y \in \mathbb{R}^d, x \neq 0.$$

[GKS,T, p.4]

Proof. Considering that $(x, y) \to \langle(\gamma^2)''(x)y, y\rangle$ is continuous on a compact set $S \subset \mathcal{R}^d$, we see that the constants

$$L = \sup_{x,y\in S} \langle(\gamma^2)''(x)y, y\rangle \text{ and } \mu = \inf_{x,y\in S} \langle(\gamma^2)''(x)y, y\rangle$$

fulfill the inequality. □

Lemma 2.1.14 (Lipschitz continuity)**.** *For the gradient* $(\gamma^2)' : \mathbb{R}^d \to \mathbb{R}^d$

$$|(\gamma^2)'(x) - (\gamma^2)'(y)| \leq L\,|x - y|$$

holds for L from Lemma 2.1.13 [GKS,T, p.4].

Proof. From Lemma 2.1.11 we know that $(\gamma^2)'(x)$ is continuous $\forall x \neq 0$. Therefore we need to consider two different cases:

Case 1: $0 \notin [x, y]$, where $[x, y]$ is the line connecting x and y
As $(\gamma^2)''(z)$ is bounded from above by L and $(\gamma^2)'(z)$ is continuous $\forall z \in [x, y]$, the fundamental theorem of calculus yields the Lipschitz continuity immedtiately.

Case 2: $0 \in [x, y]$
First we want to look at the case $[0, y]$. On $[\delta x, y]$, Lipschitz continuity holds, and as we let $\delta \to 0$ we obtain Lipschitz continuity on $[0, y]$. Then consider:

$$|(\gamma^2)'(x) + (\gamma^2)'(0) - (\gamma^2)'(0) - (\gamma^2)'(y)| \leq L(|y - 0| + |0 - x|) = L|x - y|$$

We can transform $|x| + |y| = |x - y|$ as $0 \in [x, y]$, so x and y just have opposite directions.

□

Lemma 2.1.15 (strong monotonicity)**.** *For the gradient* $(\gamma^2)' : \mathbb{R}^d \to \mathbb{R}^d$

$$\langle(\gamma^2)'(y) - (\gamma^2)'(x), y - x\rangle \geq \mu\,|y - x|^2 \;\; \forall x, y \in \mathbb{R}^d$$

holds for μ from Lemma 2.1.13 [GKS,T, p.5].

Proof. Like in Lemma 2.1.14, we have to consider the cases $0 \in [x, y]$ and $0 \notin [x, y]$.

Case 1: $0 \notin [x, y]$
From Lemma 2.1.13 we know that $(\gamma^2)''(x)$ is bounded from below $\forall x \neq 0$, so the fundamental theorem of calculus yields

$$|(\gamma^2)'(y) - (\gamma^2)'(x)| \geq \mu|y - x| \Leftrightarrow \langle(\gamma^2)'(y) - (\gamma^2)'(x), y - x\rangle \geq \mu\,|y - x|^2.$$

Case 2: $0 \in [x, y]$

Again we look at the case $[0, y]$. On $[\delta x, y]$ strong monotonicity holds, and as we let $\delta \to 0$, we obtain strong monotonicity on $[0, y]$. As $0 \in [x, y]$, x and y just have opposite directions, so we have $x = \alpha y$ for some $\alpha \leq 0$. Then we get

$$\big\langle\langle(\gamma^2)'(y) - (\gamma^2)'(x), y - x\big\rangle = \big\langle(\gamma^2)'(y) - (\gamma^2)'(0) + (\gamma^2)'(0) - (\gamma^2)'(x), y - x\big\rangle =$$

$$= \big\langle(\gamma^2)'(y) - (\gamma^2)'(0), \big(1 - \frac{1}{\alpha}\big)y - 0\big\rangle + \big\langle(\gamma^2)'(0) - (\gamma^2)'(x), (1-\alpha)x - 0\big\rangle \geq$$

$$\geq \mu\Big(1 - \frac{1}{\alpha}\Big)|y|^2 + \mu(1-\alpha)|x|^2 = \mu\Big(|y - \frac{1}{\alpha}y||y| + |x - \alpha x||x|\Big) = \mu|y - x|^2.$$

□

Now we will use Lemmata 2.1.10 to 2.1.15 to give the proof of Theorem 2.1.9.

Proof. (of Theorem 2.1.9)
We will show that (2.1.3) holds for $c = \frac{\mu}{2}$ with μ from Lemma 2.1.13.
Suppose

$$\gamma^2(x) - \gamma^2(y) < \langle(\gamma^2)'(y), x - y\rangle + \frac{\mu}{2}|x - y|^2$$

holds for some $\mu > 0$. Then we also have:

$$\gamma^2(y) - \gamma^2(x) < \langle(\gamma^2)'(x), y - x\rangle + \frac{\mu}{2}|x - y|^2$$

Adding these two inequalities, we get:

$$\langle(\gamma^2)'(y) - (\gamma^2)'(x), y - x\rangle < \mu\,|y - x|^2 \;\; \forall x, y \in \mathbb{R}^d$$

in contradiction to Lemma 2.1.15. □

Strict convexity follows directly from strong convexity:

Remark 2.1.16 (strict convexity)**.** It follows from Theorem 2.1.9 that:

$$\gamma^2(\lambda x + (1-\lambda)y) < \lambda\gamma^2(x) + (1-\lambda)\gamma^2(y),\ \forall\, 0 < \lambda < 1$$

Proof. Suppose we have

$$\gamma^2(\lambda x + (1-\lambda)y) \geq \lambda\gamma^2(x) + (1-\lambda)\gamma^2(y).$$

Then it follows

$$\langle(\gamma^2)'(y), x - y\rangle = \lim_{\lambda\to 0}\frac{\gamma^2(\lambda x + (1-\lambda)y) - \gamma^2(y)}{\lambda} > \lim_{\lambda\to 0}\frac{\lambda\gamma^2(x) - \lambda\gamma^2(y)}{\lambda} = \gamma^2(x) - \gamma^2(y)$$

in contradiction to 2.1.9. □

2.2 The phase field model

In our model, we consider the motion of several grains contained in a Lipschitz domain $\Omega \subset \mathbb{R}^d$, caused by the diffusion of solute along the grain boundaries. The grain boundaries are not directly connected, but they are separated by a gap, the so-called **interfacial region**. Looking at N different grains, each one is represented by a phase field function $\varphi_i : \mathbb{R}^d \times [0,T] \to \mathbb{R}$ for $i = 1, ..., N$. As we have to distinguish between the alloyed and the non-alloyed state, we represent the alloyed twin of φ_i by φ_{i+N}.
Since each φ_i represents the fraction of unalloyed and respectively alloyed grain present in each point of Ω, all $\varphi_i, i = 1, ..., 2N$ add up to one and each $\varphi_i \geq 0$. So $\varphi : \Omega \times [0,T] \to G$, where G is the Gibbs–Simplex in $\mathbb{R}^{2N}$

$$G := \left\{ x \in \mathbb{R}^{2N} \;\middle|\; \sum_{i=1}^{2N} x_i = 1,\ x_i \geq 0 \right\}. \tag{2.2.1}$$

The solute is represented by a dimesionless concentration $u : \Omega \times [0,T] \to \mathbb{R}$.
The inner energy of the phase field model is given by the following free energy, the so-called Ginzburg–Landau-Energy

$$E(\varphi, u) = \int_\Omega \underbrace{\frac{1}{2\epsilon} u^2}_{\substack{\text{chemical} \\ \text{energy}}} + \underbrace{\frac{\epsilon}{2} \sum_{i=1}^{N} \gamma_i \left(\nabla \left(\varphi_i + \varphi_{i+N} \right) \right)^2}_{\text{interfacial energy}} + \underbrace{\frac{W(\varphi)}{\epsilon}}_{\substack{\text{potential} \\ \text{energy}}} + \underbrace{g(u) \sum_{i=1}^{N} \varphi_i}_{\text{interaction energy}} \; dx. \tag{2.2.2}$$

This energy was first presented by H. Garcke and V. Styles [GS, pp.3-4], where a more general form of the free energy is discussed. Here, γ_i is an anisotropy function for each phase φ_i, ϵ is the thickness of the interfacial region and $W : \mathbb{R}^{2N} \to \mathbb{R}$ is a double well potential.
Looking at the special case, where each γ_i equals the eucledian norm, we obtain the already mentioned isotropic phase field model like in [KKSG, p.2].
In the following $(\cdot, \cdot)$ will denote the L^2-scalar-product in Ω.
As stated before, the movement of the grains φ is given by the scaled L^2-gradient-flow. Before we can calculate it, we need to distinguish a solution space for φ in order to determine suitable test functions. We want $\varphi \in L^2(\Omega)^{2N}$ and $\varphi_i + \varphi_{i+N} \in H^1(\Omega)$, so we define

$$X := \{ v : \Omega \subset \mathbb{R}^d \to \mathbb{R}^{2N} \mid v \in L^2(\Omega)^{2N}, v_i + v_{i+N} \in H^1(\Omega),\ \forall\, 0 \leq i \leq N \}$$

$$\text{with } (v,w)_X = \sum_{i=1}^{2N} (v_i, w_i)_{L^2(\Omega)} + \sum_{i=1}^{N} (\nabla(v_i + v_{i+N}), \nabla(w_i + w_{i+N}))_{L^2(\Omega)} =$$

$$= (v,w) + (\nabla \widetilde{v}, \nabla \widetilde{w})$$

where $\widetilde{v} = (v_i + v_{i+N})_{i=1}^N \in \mathbb{R}^N$. Then the admissible set for φ is given by

$$\mathcal{G} := \{ v \in X \mid v(x) \in G \text{ for almost all } x \in \Omega \}. \tag{2.2.3}$$

Now we are ready to calculate the L^2-gradient-flow of the Ginzburg–Landau-Energy (2.2.2). The L^2-gradient-flow is given by

$$\epsilon(\varphi_t, v) = -\frac{\partial}{\partial\varphi}E(\varphi)(v), \ \forall v \in \mathcal{G}$$

$$\Leftrightarrow \epsilon(\varphi_t, v) = -\sum_{i=1}^{N} \epsilon(\gamma_i(\nabla\widetilde{\varphi}_i)\gamma_i'(\nabla\widetilde{\varphi}_i), \nabla v_i) - \frac{1}{\epsilon}(W'(\varphi), v) - (g(u)\cdot e, v), \ \forall v \in \mathcal{G} \quad (2.2.4)$$

where $e = (1, ..., 1, 0, ..., 0) \in \mathbb{R}^{2N}$. We obtain an equation of Allen–Cahn type.
The actual parabolic Allen–Cahn equation (Allen and Cahn 1979) is given by

$$\epsilon\varphi_t - \epsilon\nabla\varphi + \frac{1}{\epsilon}W'(\varphi) = 0.$$

This equation results, if we evaluate the L^2-gradient-flow of our energy (2.2.2), but use $\gamma_i(.) = |.|$, $\forall i$ and completely leave out the motion of the solute.
We choose W to be the classical obstacle potential, so

$$W(\varphi) = \begin{cases} \sum_{i<j}\varphi_i\varphi_j = \frac{1}{2}\sum_{i=1}^{2N}\varphi_i - \varphi_i^2 = \frac{1}{2}\big(1 - \sum_{i=1}^{2N}\varphi_i^2\big) & \text{for } \varphi \in G \\ \infty & \text{for } \varphi \notin G. \end{cases}$$

Using the characteristic function of $\mathcal{G}$, the admissible set for φ (2.2.3)

$$\chi_{\mathcal{G}}(\varphi) = \begin{cases} 0 & \text{if } \varphi \in \mathcal{G} \\ \infty & \text{if } \varphi \notin \mathcal{G} \end{cases} \quad (2.2.5)$$

we get the equivalence

$$W(\varphi) = \chi_{\mathcal{G}}(\varphi) + \sum_{i<j}\varphi_i\varphi_j.$$

For the subdifferential of $\chi_{\mathcal{G}}$ we have

$$(\partial\chi_{\mathcal{G}}(\varphi), . - \varphi) = \begin{cases} 0 & \text{for } \varphi \in \mathcal{G} \\ \infty & \text{for } \varphi \notin \mathcal{G}. \end{cases} \quad (2.2.6)$$

Then the equation (2.2.4) transforms to the following inequality:

$$\epsilon(\varphi_t, v-\varphi) \geq -\sum_{i=1}^{N} \epsilon(\gamma_i(\nabla\widetilde{\varphi}_i)\gamma_i'(\nabla\widetilde{\varphi}_i), \nabla\widetilde{v_i} - \nabla\widetilde{\varphi}_i) + \frac{1}{\epsilon}(\varphi, v-\varphi) - (g(u)\cdot e, v-\varphi), \ \forall v \in \mathcal{G}. \quad (2.2.7)$$

Naturally, the boundary conditions for the phase field function φ are given by [KKSG, p.2]

$$\nabla\varphi \cdot \nu = 0 \quad \text{on } \partial\Omega. \quad (2.2.8)$$

The solute u fulfills the kinetic equation from [GS, p.4]

$$\beta\epsilon u_t = \nabla \cdot (D(\varphi)\nabla w) \quad \text{in } \Omega \quad (2.2.9)$$

and the Newton flux boundary conditions from [KKSG, p.2]

$$D(\varphi)\frac{\partial w}{\partial \nu} = \alpha D(\varphi)(1-u) \quad \text{on } \partial\Omega \tag{2.2.10}$$

where $\beta > 0$ and $\alpha \gg 0$ are coefficients and

$$w = \epsilon \frac{\partial}{\partial u} E(\varphi, u) = u + \epsilon g'(u) \sum_{i=1}^{N} \varphi_i$$

is the scaled chemical potential. As the diffusion of the solute along the grain boundaries can only occur inside the interfacial region, i.e. where all $\varphi_i \neq 1$, the equations (2.2.9) and (2.2.10) contain the so-called diffusivity

$$D(\varphi) = \prod_{i=1}^{2N} (1-\varphi_i).$$

From (2.2.9) and (2.2.10) we obtain the weak formulation

$$\beta\epsilon(u_t, v) + (D(\varphi)\nabla w, \nabla v) + \alpha(D(\varphi)(u-1), v)_{\partial\Omega} = 0, \ \ \forall v \in H^1(\Omega) \tag{2.2.11}$$

Looking at (2.2.11) or (2.2.9) it is even more obvious that the solute can only flow inside of the interfacial region, i.e. where $D(\varphi) \neq 0$, because outside of the interfacial region, they reduce to

$$u_t = 0.$$

Thus, the domain Ω is divided into two parts: the inactive set or interfacial region

$$\Omega_I(\varphi) = \{x \in \Omega \mid D(\varphi) > 0\} \tag{2.2.12}$$

with thickness ϵ and the active set

$$\Omega_A(\varphi) = \Omega \backslash \Omega_I(\varphi), \tag{2.2.13}$$

where the constraints from the Gibbs Simplex (2.2.1) hold.
The Newton flux boundary conditions (2.2.10) only hold on $\partial\Omega_I \backslash int(\Omega)$, the green part of the boundary, as it can be seen in Figure 1.

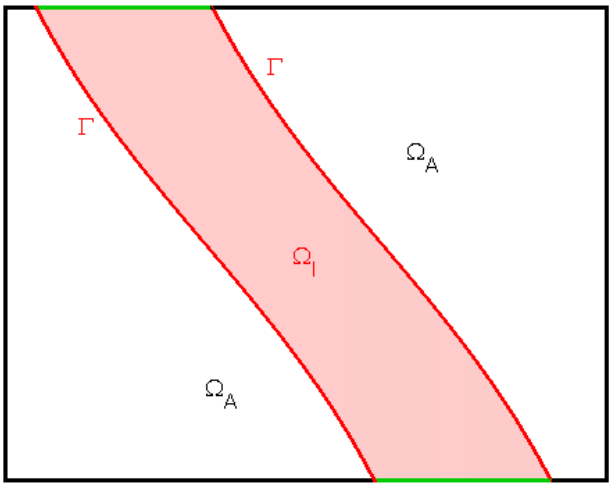


Figure 1: separation of Ω

3 Discretization in time and space

In this section we want to prepare the algebraic and numerical solutions of our problems by discretizing them in time and space.
In the first part we discretize the variational problems (2.2.7) and (2.2.11) in time using Rothe's Method. We obtain two discretized variational problems that are equivalent to minimization problems. Then, we prove existence and uniqueness of the solution of the minimization problem for the phase field function. It is not clear if the minimization problem for the solute function has a unique solution in the continuous space.
After the discretization in time, we discretize the minimization problems in space. Now, we can show that the minimization problem for the solute function has a unique solution in the discrete space.
The formulation in the discrete space will then be used in the next chapter to gain an algebraic formulation and a numerical solution of the discretized mimization problems.

3.1 Discretization in time (Rothe's Method)

We want to solve the problems (2.2.7) and (2.2.11) on the interval $[0,T]$, so we choose a small timestep $\triangle t$ and split $[0,T]$ into intervals $[t_k, t_{k+1}]$ with $t_k := k\triangle t$ and get the following approximations for the time derivatives:

$$\frac{u^{k+1}-u^k}{\triangle t} = u_t \quad \text{and} \quad \frac{\varphi^{k+1}-\varphi^k}{\triangle t} = \varphi_t$$

Using an implicit Euler method, the variational inequality (2.2.7) transforms to

$$\left(\frac{\epsilon}{\triangle t} - \frac{1}{\epsilon}\right)(\varphi^{k+1}, v-\varphi^{k+1}) + \sum_{i=1}^{N} \epsilon(\gamma_i(\nabla\widetilde{\varphi}_i^{k+1})\gamma_i'(\nabla\widetilde{\varphi}_i^{k+1}), \nabla\widetilde{v_i} - \nabla\widetilde{\varphi}_i^{k+1}) \geq$$

$$\frac{\epsilon}{\triangle t}(\varphi^k, v-\varphi^k) - (g(u^k)\cdot\vec{e}, v-\varphi^k) \ \forall v \in \mathcal{G}.$$

According to [ET, p.37], this variational inequality is equivalent to the minimization problem

$$\varphi^{k+1} \in \mathcal{G}\colon\ J_0(\varphi^{k+1}) \leq J_0(v) \ \forall v \in \mathcal{G} \tag{3.1.1}$$

with $J_0(\varphi^{k+1}) =$

$$= \int_\Omega \left(\frac{\epsilon}{2\triangle t} - \frac{1}{2\epsilon}\right)\left(\varphi^{k+1}\right)^2 + \sum_{i=1}^{N}\frac{\epsilon}{2}\gamma_i\left(\nabla\widetilde{\varphi}_i^{k+1}\right)^2 - \frac{\epsilon}{\triangle t}\varphi^{k+1}\varphi^k + g(u^k)\sum_{i=1}^{N}\varphi_i^{k+1}\ dx.$$

Now we want to determine if this minimization problem has a unique solution. To formulate the following uniqueness theorem, we need to define lower-semi-continuous and proper.

Definition 3.1.1 (lower-semi-continuous, proper)**.** *Let V be a Hilbertspace.*
A function $f : V \to \mathbb{R} \cup \{+\infty\}$ is called lower-semi-continuous if $\forall v_n \to v$ in V

$$\liminf_{n\to\infty} f(v_n) := \lim_{n\to\infty} \big(\inf_{m\geq n} f(v_n)\big) \geq f(v)$$

A function f is called proper if $f \not\equiv +\infty$ and $f(v) \neq -\infty \ \forall v \in H$ [ET, p. 8]

Theorem 3.1.2. *Let V be a Hilbertspace, $\mathcal{C}$ a closed convex subset of V and $J_0 : \mathcal{C} \to \mathbb{R}$. Then there exists a unique $u \in \mathcal{C}$ that solves the minimization problem*

$$u \in V: \ J_0(u) \leq J_0(v) \ \forall v \in V$$

if:

i) J_0 is strictly convex, lower-semi-continuous and proper

ii) J_0 is coercive over $\mathcal{C}$, i.e. $\lim_{||u||\to\infty} J_0(u) = \infty$

[ET, p.35]

Now we want to apply Theorem 3.1.2 to our minimization problem (3.1.1), but it is already clear that the functional J_0 is in general not convex, because the term $\frac{\epsilon}{2\triangle t} - \frac{1}{2\epsilon}$ is negative if $\triangle t < \epsilon^2$. This is a very pessimistic timestep constraint, as, on the one hand, in reality the interfacial region is very narrow and we therefore want to use a fairly small ϵ, but on the other hand $\triangle t$ should not be chosen that small. We can either accept the restriction and use a tiny $\triangle t < \epsilon^2$ or we can avoid this timestep constraint using a semi-implicit Euler scheme instead. This means we take the term associated with $-\frac{1}{2\epsilon}$ with φ^k instead of φ^{k+1}. Using this, the variational inequality (2.2.7) transforms to

$$\frac{\epsilon}{\triangle t}(\varphi^{k+1}, v - \varphi^{k+1}) + \sum_{i=1}^{N} \epsilon(\gamma_i(\nabla\widetilde{\varphi}_i^{k+1})\gamma_i'(\nabla\widetilde{\varphi}_i^{k+1}), \nabla\widetilde{v}_i - \nabla\widetilde{\varphi}_i^{k+1}) \geq$$

$$\geq \left(\frac{\epsilon}{\triangle t} + \frac{1}{\epsilon}\right)(\varphi^k, v - \varphi^{k+1}) - (g(u^k)\cdot e, v - \varphi^{k+1}) \quad \forall v \in \mathcal{G}. \tag{3.1.2}$$

Now we have an equivalence to the minimization problem

$$\varphi^{k+1} \in \mathcal{G}: \ J_0(\varphi^{k+1}) \leq J_0(v) \ \forall v \in \mathcal{G} \tag{3.1.3}$$

$$\text{with} \quad J_0(\varphi^{k+1}) = \int_\Omega \frac{\epsilon}{2\triangle t}\big(\varphi^{k+1}\big)^2 + \sum_{i=1}^{N} \frac{\epsilon}{2}\gamma_i\big(\nabla\widetilde{\varphi}_i^{k+1}\big)^2 - \left(\frac{\epsilon}{\triangle t} + \frac{1}{\epsilon}\right)\varphi^{k+1}\varphi^k + g(u^k)\sum_{i=1}^{N}\varphi_i^{k+1}\ dx.$$

In the following we assume that the interaction function g depends linearly on the solute. Now we can prove existence and uniqueness of the solution of the minimization problem (3.1.3).

Theorem 3.1.3. *The minimization problem* (3.1.3) *has a unique solution in* $\mathcal{G}$ *if each* γ_i *is admissible and* g *is linear.*

Proof. As it is clear that $\mathcal{G}$ (2.2.3) is a closed convex subset of X, we need to check i) and ii) of Theorem 3.1.2.

- J_0 is strictly convex, as the quadratic together with the linear parts are strictly convex and the part containing the anisotropy function is also strictly convex. Look at:

$$\int_\Omega \sum_{i=1}^N \frac{\epsilon}{2}\gamma_i^2\big(\lambda\nabla\widetilde{u}_i+(1-\lambda)\nabla\widetilde{v}_i\big) \overset{\gamma_i^2 \text{ strictly convex}}{\leq} \lambda\int_\Omega \sum_{i=1}^N \frac{\epsilon}{2}\gamma_i^2\big(\nabla\widetilde{u}_i\big)+(1-\lambda)\int_\Omega \sum_{i=1}^N \frac{\epsilon}{2}\gamma_i^2\big(\nabla\widetilde{v}_i\big) \quad \forall\lambda\in(0,1)$$

- J_0 is lower-semi-continuous. Take any convergent sequence $v_n \to v \in \mathbb{R}^{2N}$. We need to treat two different cases:

  Case 1: $v \in \mathcal{G} \Leftrightarrow J_0(v) < \infty$
  As the sequence is convergent, we either have $\liminf_{n\to\infty} J_0(v_n) = \infty$ or $\liminf_{n\to\infty} J_0(v_n) = J_0(v)$.
  So we automatically obtain: $\liminf_{n\to\infty} J_0(v_n) \geq J_0(v)$

  Case 2: $v \notin \mathcal{G} \Leftrightarrow J_0(v) = \infty$
  Look at: $\lim_{n\to\infty}\big(\inf_{m\geq n} f(v_n)\big)$. As the sequence is convergent and $\mathcal{G}$ is closed, there is some n' with: $v_{n'} \notin \mathcal{G}$, so $\inf_{m\geq n'} J_0(v_{n'}) = \infty$. Then we obtain $\liminf_{n\to\infty} J_0(v_n) = \infty = J_0(v)$

- J_0 is proper, as $-\infty < J_0(v) \leq \infty\ \forall v \in \mathbb{R}^{2N}$ and $\exists v \in \mathbb{R}^{2N} : J_0(v) < \infty$

- J_0 is coercive, because $\exists\,\alpha$: $J_0(v) \geq \alpha||v||_X^2 = \alpha(v,v)_X$

  Look at: $\displaystyle\int_\Omega \sum_{i=1}^N \frac{\epsilon}{2}\gamma_i^2\big(\nabla\widetilde{v}_i\big) = \sum_{i=1}^N \frac{\epsilon}{2}\int_\Omega \gamma_i^2\big(\nabla\widetilde{v}_i\big)-\gamma_i^2(0) \overset{\text{Thm 2.1.9}}{\geq}$

$$\geq \sum_{i=1}^N \frac{\epsilon}{2}\int_\Omega \big\langle(\gamma_i^2)'(0), \nabla\widetilde{v}_i-0\big\rangle+\frac{\mu}{2}|\nabla\widetilde{v}_i-0|^2 \overset{(\gamma_i^2)'(0)=0,\ \text{see } 2.1.11}{=} \sum_{i=1}^N \frac{\epsilon\mu}{4}(\nabla\widetilde{v}_i, \nabla\widetilde{v}_i) = \frac{\epsilon\mu}{4}(\nabla v_i, \nabla v_i)$$

  and at: $\displaystyle\int_\Omega g(u)\sum_{i=1}^N v_i - \Big(\frac{\epsilon}{\triangle t}+\frac{1}{\epsilon}\Big)v\varphi^k = \Big(g(u)\cdot e - \big(\frac{\epsilon}{\triangle t}+\frac{1}{\epsilon}\big)\varphi^k, v\Big) =: -l(v) \geq$

$$\geq -||l||_{L^2(\Omega)'}||v||_{L^2(\Omega)} = c_1||v||_{L^2(\Omega)}$$

  Then: $\displaystyle J_0(v) = \int_\Omega \frac{\epsilon}{2\triangle t}v^2 + \sum_{i=1}^N \frac{\epsilon}{2}\gamma_i^2\big(\nabla\widetilde{v}_i\big) - \left(\frac{\epsilon}{\triangle t}+\frac{1}{\epsilon}\right)v\varphi^k + g(u)\sum_{i=1}^N u_i \geq$

$$\geq \frac{\epsilon}{2\triangle t}\Big(v,v\Big) + \frac{\epsilon\mu}{4}\sum_{i=1}^N\Big(\nabla v_i, \nabla v_i\Big) - l(v) \geq c_2||v||_X^2 + c_2||v||_{L^2(\Omega)}$$

$$\geq \alpha||v||_X^2 + c \quad \text{for } v\to\infty$$

□

Until now, we have only used $\mathcal{G}$ as the domain of the functional J_0. We can extend the domain to the whole space X though by adding the characteristic function of the set $\mathcal{G}$ (2.2.5) to the functional J_0 and obtain

$$J = J_0 + \chi_{\mathcal{G}}. \tag{3.1.4}$$

However, in contrast to J_0, J is not differentiable on its whole domain. Just like $\partial\chi_{\mathcal{G}}$ (2.2.6) is set valued outside of $\mathcal{G}$, we get

$$\partial J(v) = \begin{cases} \nabla J_0 & \text{for } \varphi \in \mathcal{G} \\ [0, \infty) & \text{for } \varphi \notin \mathcal{G}. \end{cases}$$

Using the functional J, the minimization problem (3.1.3) transfers to

$$\varphi^{k+1} \in X : \; J(\varphi^{k+1}) \leq J(v) \quad \forall v \in X \tag{3.1.5}$$

and the variational inequality (3.1.2) transfers to

$$\frac{\epsilon}{\triangle t}(\varphi^{k+1}, v - \varphi^{k+1}) + \sum_{i=1}^{N} \epsilon(\gamma_i(\nabla\widetilde{\varphi}_i^{k+1})\gamma_i'(\nabla\widetilde{\varphi}_i^{k+1}), \nabla\widetilde{v}_i - \nabla\widetilde{\varphi}_i^{k+1}) + \chi_{\mathcal{G}}(v) - \chi_{\mathcal{G}}(\varphi^{k+1}) \geq$$

$$\geq \left(\frac{\epsilon}{\triangle t} + \frac{1}{\epsilon}\right)(\varphi^k, v - \varphi^{k+1}) - (g(u^k) \cdot e, v - \varphi^{k+1}) \quad \forall v \in X. \tag{3.1.6}$$

Now we are ready to apply an implicit Euler method to the variational equality (2.2.11) of the solute. We end up with

$$\frac{\beta\epsilon}{\triangle t}(u^{k+1}, v) + (D(\varphi^k)\nabla w^{k+1}, \nabla v) + \alpha(D(\varphi^k)(u^{k+1} - 1), v)_{\partial\Omega} = \frac{\beta\epsilon}{\triangle t}(u^k, v), \;\; \forall v \in H^1(\Omega). \tag{3.1.7}$$

Usually, for a variational equality of the form

$$v : \;\; a(u, v) = l(u), \;\; \forall u$$

which we have in (3.1.7) we can use the Lax–Milgram-Lemma to determine if it has a unique solution.

Theorem 3.1.4 (Lax–Milgram-Lemma)**.** *Let V be a Hilbertspace, $a(.,.)$ a V-elliptic bilinear form and $l \in V'$ a functional in the dual space of V. Then the variational equality*

$$u \in V : \;\; a(u, v) = l(v), \;\; \forall v \in V$$

has a unique solution in V.

We can see that the variational equality (3.1.7) does not have a unique solution in $H^1(\Omega)$, as the left side, the bilinear part, is not $H^1(\Omega)$-elliptic and not even $H^1(\Omega_I)$-elliptic. If we look at $D(\varphi^k)$, we see that

$$\lim_{\varphi_i^k \to 1} D(\varphi^k) = 0, \text{ for some } i \in \{1, ..., 2N\}.$$

Therefore, we cannot find a constant $c > 0$ such that: $a(v, v) \geq c||v||^2_{H^1(\Omega)}$.
However, the variational equality (3.1.7) has a unique solution in a discrete setting inside the inactive part Ω_I (2.2.12), because the bounded domain Ω then consists of finitely many points. The discretization in space will be done in the next chapter.

Before we discretize (3.1.2) and (3.1.7) in space though, we take a look at a simplified 2D-model, which is presented in [GS, p.6] and [KKSG, p.4] and works for very thin specimens (for instance a very fine copper plate). There, it is assumed that the solute does not have to flow into the interfacial region but is present immediately, so

$$u^k = \begin{cases} 1 & \forall p \in \Omega, \text{ where } \varphi_i^k(p) > 0 \\ 0 & \text{else} \end{cases}$$

substitutes the variational equations for the solute, especially (3.1.7). The equations for the phase field function stay the same.

3.2 Finite element discretization

Similar to [GS, p.10], we let Ω_h be a polyhedral approximation of Ω. Before we can discretize Ω_h, we need to define what a conforming triangulation is.

Definition 3.2.1 (triangulation)**.** *A finite set $\mathcal{T} \subset 2^\Omega$ is called a triangulation of $\Omega_h \subset \mathbb{R}^d$ if each $\tau \in \mathcal{T}$ is a nonempty, d-dimensional open simplex and*

$$\overline{\Omega} = \bigcup_{\tau \in \mathcal{T}} \overline{\tau}, \qquad \tau_1 \neq \tau_2 \Rightarrow \tau_1 \cap \tau_2 = \emptyset \qquad \forall \tau_1, \tau_2 \in \mathcal{T}.$$

We will use the term face for faces of arbitrary dimension smaller or equal to $dim(\Omega_h)$ and denote them by $\mathcal{F}_{dim}$. The set of all nodes in $\mathcal{T}$ is denoted by $\mathcal{N} = \mathcal{F}_0$ and the set of all edges is denoted by $\mathcal{E} = \mathcal{F}_1$. [Gr, p.23]

Definition 3.2.2 (conforming)**.** *A triangulation is called conforming, if*

$$\forall \tau_i \neq \tau_j \in \mathcal{T}_j : \overline{\tau_i} \cap \overline{\tau_j} = \begin{cases} \emptyset & \text{either empty or} \\ f \in \mathcal{F}_{dim}, \text{ where } f \in \overline{\tau_i} \text{ and } f \in \overline{\tau_j} & \text{a common face.} \end{cases}$$

[Gr, p.23]

Now let $\mathcal{T}_j$ be a conforming triangulation of Ω_h with the mesh size

$$h_j := \max_{\tau \in \mathcal{T}_j} \operatorname{diam}(\tau), \tag{3.2.1}$$

faces $\mathcal{F}_{j,dim}$, edges $\mathcal{E}_j$ and nodes $\mathcal{N}_j$. The examples in the last chapter of this thesis will only treat $\Omega \subset \mathbb{R}^2$ and $\Omega \subset \mathbb{R}^3$.
On the triangulation, we will use the finite element space

$$S_j := \left\{ \chi \in C^0(\Omega_h) \;\middle|\; \chi \text{ is linear on each } \tau \in \mathcal{T}_j \right\} \tag{3.2.2}$$

together with the nodal basis

$$\Lambda^{(j)} = \left\{ \lambda_i \mid p_i \in \mathcal{N}_j \right\}, \quad \lambda_i(p_k) = \delta_{ik}. \tag{3.2.3}$$

In analogon to the admissible set for φ in the continuous setting (2.2.3), we define the discrete equivalent

$$\mathcal{G}_j := \left\{ v \in (S_j)^{2N} \;\middle|\; \sum_{i=1}^{2N} v_i = 1, v_i \geq 0 \quad \forall\ i = 1, ..., 2N \right\} = \mathcal{G}_j \cap (S_j)^{2N}. \tag{3.2.4}$$

The space for the phase field function $\varphi_{\mathcal{T}_j}^{k+1} \in (S_j)^{2N}$ is now given by $\mathcal{G}_j$. We see that $(S_j)^{2N} \subset X$, as $(S_j)^{2N}$ contains piecewise linear and continuous functions, so $\varphi_{\mathcal{T}_j}^{k+1} \in L^2(\Omega)$ and $(\varphi_{\mathcal{T}_j}^{k+1})_i + (\varphi_{\mathcal{T}_j}^{k+1})_{i+N} \in H^1(\Omega)$. If we are given the phase field function $\varphi_{\mathcal{T}_j}^{k}$ and the solute function $u_{\mathcal{T}_j}^{k}$ for a time $t_k := k\triangle t$, then we can determine $\varphi_{\mathcal{T}_j}^{k+1}$ by solving

$$\begin{aligned} &\frac{\epsilon}{\triangle t}\Big(\varphi_{\mathcal{T}_j}^{k+1}, v - \varphi_{\mathcal{T}_j}^{k+1}\Big) + \sum_{i=1}^{N} \epsilon\Big(\gamma_i\big((\nabla\widetilde{\varphi}_{\mathcal{T}_j}^{k+1})_i\big)\gamma_i'\big((\nabla\widetilde{\varphi}_{\mathcal{T}_j}^{k+1})_i\big), \nabla\widetilde{v}_i - (\nabla\widetilde{\varphi}_{\mathcal{T}_j}^{k+1})_i\Big) \geq \\ &\geq \left(\frac{\epsilon}{\triangle t} + \frac{1}{\epsilon}\right)\Big(\varphi_{\mathcal{T}_j}^{k}, v - \varphi_{\mathcal{T}_j}^{k+1}\Big) - \Big(g(u_{\mathcal{T}_j}^{k}) \cdot e, v - \varphi_{\mathcal{T}_j}^{k+1}\Big) \quad \forall v \in \mathcal{G}_j \end{aligned} \tag{3.2.5}$$

or the equivalent minimization problem (where J_0 is given just like in (3.1.3)):

$$\varphi_j^{k+1} \in \mathcal{G}_j\colon\ J_0(\varphi_{\mathcal{T}_j}^{k+1}) \leq J_0(v)\ \ \forall v \in \mathcal{G}_j. \tag{3.2.6}$$

Again, we have to determine if (3.2.5) or (3.2.6) respectively have a unique solution. We can apply Theroem 3.1.2, just like in the section before, the only difference is that now the Hilbertspace is given by the finite element space $(S_j)^{2N}$.

The space for the solute is given by S_j and we can obtain $u_{\mathcal{T}_j}^{k+1}$ for given $\varphi_{\mathcal{T}_j}^{k+1}$ and $u_{\mathcal{T}_j}^{k}$ by solving the discrete equivalent to (3.1.7)

$$\frac{\beta\epsilon}{\triangle t}\big(u_{\mathcal{T}_j}^{k+1} - u_{\mathcal{T}_j}^{k}, v\big) + \big(D(\varphi_{\mathcal{T}_j}^{k+1})\nabla w_{\mathcal{T}_j}^{k+1}, \nabla v\big) + \alpha\big(D(\varphi_{\mathcal{T}_j}^{k+1})(u_{\mathcal{T}_j}^{k+1} - 1), v\big)_{\partial\Omega} = 0,\ \forall v \in S_j \tag{3.2.7}$$

with

$$w_{\mathcal{T}_j}^{k+1} = u_{\mathcal{T}_j}^{k+1} + \epsilon g'(u_{\mathcal{T}_j}^{k}) \sum_{i=1}^{N} (\varphi_{\mathcal{T}_j}^{k+1})_i.$$

Now we can show that the equation (3.2.7) has a unique solution, because we can apply the Lax–Milgram-Lemma 3.1.4.

Theorem 3.2.3. *Let $(\Omega_h)_I$ be the discrete equivalent of Ω_I (2.2.12) and $\mathcal{T}((\Omega_h)_I)$ be a conforming triangulation of $(\Omega_h)_I$. Then, for a linear g, the variational equality (3.2.7) has a unique solution in*

$$(S_j)_I = \left\{ \chi \in C^0\big((\Omega_h)_I\big) \mid \chi \text{ is linear on each } \tau \in \mathcal{T}\big((\Omega_h)_I\big) \right\}.$$

Proof. We want to apply the Lax–Milgram-Lemma 3.1.4 to the variational equality (3.2.7) limited to $(\Omega_h)_I$. (3.2.7) takes the form $a(u,.) = l(.)$, where $a(.,.)$ is bilinear and l is from the dual space of $(S_j)_I$, so we still need to show that $a(.,.)$ is $(S_j)_I$-elliptic, i.e there exists $c > 0$ such that

$$\frac{\beta\epsilon}{\triangle t}\big(v,v\big) + \Big(D(\varphi_{\mathcal{T}_j}^{k+1})\nabla v, \nabla v\Big) + \alpha\big(D(\varphi_{\mathcal{T}_j}^{k+1})v, v\big)_{\partial(\Omega_I)\setminus \operatorname{int}(\Omega)} \geq c||v||_{S_j}^2$$

for $\varphi_{\mathcal{T}_j}^{k+1} \in \mathcal{G}_j$ and for all $v \in (S_j)_I$.
Now we are in a discrete setting, so we can find a $c_1 > 0$ such that $D(\varphi_j^{k+1}) > c_1$ inside $(\Omega_h)_I$, so

$$\big(D(\varphi_{\mathcal{T}_j}^{k+1})\nabla v, \nabla v\big) \geq c_1(\nabla v, \nabla v).$$

As $v \in S_j$, we can estimate the integral on the boundary from below by

$$\alpha\big(D(\varphi_{\mathcal{T}_j}^{k+1})v, v\big)_{\partial(\Omega_I)\setminus \operatorname{int}(\Omega)} \geq c_1 c_2 (v,v).$$

Then we get

$$\frac{\beta\epsilon}{\triangle t}\big(v,v\big) + \Big(D(\varphi_{\mathcal{T}_j}^{k+1})\nabla v, \nabla v\Big) + \alpha\big(D(\varphi_{\mathcal{T}_j}^{k+1})v, v\big)_{\partial(\Omega_I)\setminus \operatorname{int}(\Omega)} \geq$$

$$\geq \frac{\beta\epsilon}{\triangle t}(v,v) + (c_1 + c_1 c_2)(\nabla v, \nabla v) \geq c||v||_{S_j}^2$$

□

Using the discretizations in time and space we just constructed, we can give an algebraic formulation of our problems (3.2.5) or (3.2.6) respectively and (3.1.7) and solve them numerically in the next section.

4 Algebraic solutions of the discretized spatial problems

In this section we will derive the algebraic and accordingly numerical solutions of our variational problems. For simplification, we will now denote the discrete phase field function at the time t_{k+1} by φ instead of $\varphi_{\mathcal{T}_j}^{k+1}$, the discrete solute function by u instead of $u_{\mathcal{T}_j}^{k+1}$ and call the ν-th iterates within a numerical scheme φ^ν and u^ν.

In this section, a basic Gauss–Seidel-algorithm to solve our minimization problems together with its algebraic formulation is presented first. Afterwards, a multigrid algorithm with a much better convergence rate is considered. However, the multigrid algorithm for the phase field function involves a significantly higher effort, because the minimization problem (3.1.5) is simplex-constrained and nonlinear. In the implementation, we used an algorithm that avoids the high effort but still shows optimal convergence, namely the **truncated nonsmooth newton multigrid method (TNNMG)**. Using this algorithm, the actual problem is just solved on the finest grid and on the coarser grids, a linearization is considered instead. This method will be presented at the end of this section.

4.1 Polyhedral Gauss–Seidel-method

In order to obtain φ we will solve the discrete version of the minimization problem (3.1.5)

$$\varphi \in (S_j)^{2N} : J(\varphi) \leq J(v), \quad \forall v \in (S_j)^{2N} \tag{4.1.1}$$

or equivalently the discrete version of (3.1.3)

$$\varphi \in \mathcal{G}_j : J_0(\varphi) \leq J_0(v), \quad \forall v \in \mathcal{G}_j \tag{4.1.2}$$

where J is of the form $J(\varphi) = J_0(\varphi) + \chi_{\mathcal{G}_j}(\varphi)$, like in (3.1.4).
As stated before, (4.1.1) and (4.1.2) are equivalent to solving a variational inequality of the form

$$\varphi \in \mathcal{G}_j : \langle \nabla J_0(\varphi), v - \varphi \rangle \geq 0, \quad \forall v \in \mathcal{G}_j. \tag{4.1.3}$$

In the anisotropic case, ∇J_0 is nonlinear, only in the isotropic case ∇J_0 is linear, thus we can rewrite (4.1.3) and get

$$\varphi \in \mathcal{G}_j : a(\varphi, v - \varphi) \geq l(v - \varphi), \quad \forall v \in \mathcal{G}_j.$$

If we did not have to solve the minimization problem (4.1.2) within the convex set $\mathcal{G}_j$ but within the whole space $(S_j)^{2N}$, we could choose for instance a normal Gauss–Seidel-method to find the minimum φ of J_0:
One iteration of the Gauss–Seidel-method is given by:

1. initialize $w_0 = \varphi^\nu$

2. for $n = 1, ..., \#\mathcal{N}_j$:
   $\{$ for $i = 1, ..., 2N$:

$$\{ \quad \text{set } corr_{n,i} = \underset{v \in span\{\lambda_n e_i\}}{\operatorname{argmin}} J_0(w_{n-1} + v)$$

$$\text{update } w_{n-1} = w_{n-1} + corr_{n,i} \quad \} \tag{4.1.4}$$

   set $w_n = w_{n-1}\}$

3. set $\varphi^{\nu+1} = w_{\#\mathcal{N}_j}$

where $e_i \in \mathbb{R}^{2N}$ is the $i-$th eucledian unit vector in $\mathbb{R}^{2N}$ and $\lambda_n \in \Lambda^{(j)}$ is the nodal basis function at the node p_n.
In the isotropic case, the functional J_0 is given by $\frac{1}{2}a(.,.) - l(.)$, so we can calculate the $n, i-$th correction by

$$corr_{n,i} = \frac{l(\lambda_n e_i) - a(w_{n-1}, \lambda_n e_i)}{a(\lambda_n e_i, \lambda_n e_i)} \lambda_n e_i.$$

In the anisotropic case, we cannot calculate $corr_{n,i}$ directly, because ∇J_0 is nonlinear. We have to use a bisection method to find $corr_{n,i} = z\, e_i \lambda p_n$ that satisfies

$$\langle \nabla J_0(w_{n-1} + z\lambda_n e_i), \lambda_n e_i \rangle = 0. \tag{4.1.5}$$

For convergence, it suffieces to find $z' \in (\epsilon', 1]z$ close enough to the correct z for a fixed $\epsilon' \in (0, 1]$, as explained in [Gr, p.55]. This bisection can be regarded as an inexact variant of the Gauss–Seidel-algorithm presented above. Using the inexact version, the computation of the correction in step 2 is substituted by

$$\text{find} \quad corr_{n,i} \in (\epsilon', 1] \underset{v \in span\{\lambda_n e_i\}}{\operatorname{argmin}} J_0(w_{n-1} + v). \tag{4.1.6}$$

We have to solve our problem inside the convex set $\mathcal{G}_j$ though, so we need to use other search directions than the eucledian ones. In [GSa,P] it is proven that for a minimization problem of our kind, we obtain global convergence if we use the the edges of the convex set as search directions.
The edges of the convex set G (2.2.1) are given by

$$\mathcal{E} := \{E_{m(i,j)} \mid E_{m(i,j)} = e_i - e_j \in \mathbb{R}^{2N}\}. \tag{4.1.7}$$

and they can be enumerated by the bijective mapping

$$m : \big\{(i,j) \mid i, j \in \{1, ..., 2N\}, i < j\big\} \to \big\{1, ..., M\big\}$$

where $\#\mathcal{E} = M = \frac{1}{2}2N(2N-1) = N(2N-1)$.
Using these edges as search directions we obtain a polyhedral or polygonal Gauss–Seidel-method [KKSG, p.5]:

1. initialize $w_0 = \varphi^\nu$
2. for $n = 1, ..., \#\mathcal{N}_j$:
   $\{$ for $m = 1, ..., M$:

$$\Big\{ \quad \text{set } corr_{n,m} = \underset{\substack{v \in span\{\lambda_n E_m\} \\ w+v \in \mathcal{G}_j}}{\operatorname{argmin}} J_0(w_{n-1} + v)$$

$$\text{update } w_{n-1} = w_{n-1} + corr_{n,m} \qquad \Big\} \tag{4.1.8}$$

   set $w_n = w_{n-1}\Big\}$
3. set $\varphi^{\nu+1} = w_{\#\mathcal{N}_j}$

The $n, m-$th correction can be calculated by

$$corr_{n,m} = z\lambda_n E_m. \tag{4.1.9}$$

In the istropic case, the factor z is given by

$$z = max\left\{-w_{n-1}(\lambda_n)_i, min\left\{\frac{l(\lambda_n E_m) - a(w_{n-1}, \lambda_n E_m)}{a(\lambda_n E_m, \lambda_n E_m)}, w_{n-1}(\lambda_n)_j\right\}\right\}.$$

In the anisotropic case, again, we have to use a bisection method that solves

$$\langle \nabla J_0(w_{n-1} + z\lambda_n E_m), \lambda_n E_m \rangle + \partial \chi_{[-w_{n-1}(\lambda_n)_i, w_{n-1}(\lambda_n)_j]}(z) \ni 0.$$

Like before, for convergence, it is enough to find a $z' \in (\epsilon', 1]z$ for a fixed $\epsilon' \in (0, 1]$. Again, this bisection can be regarded as an inexact version of the polyhedral Gauss–Seidel-method and, analogously to (4.1.6), when using the inexact version, the computation of the correction in step 2 is substituted by

$$\text{find} \quad corr_{n,m} \in (\epsilon', 1] \underset{\substack{v \in span\{\lambda_n E_m\} \\ w+v \in \mathcal{G}_j}}{\operatorname{argmin}} J_0(w_{n-1} + v). \tag{4.1.10}$$

In the next theorem, we will show that the polyhedral Gauss–Seidel-method (4.1.8) using the search directions

$$\{\lambda_n E_m \mid \lambda_n \in \Lambda^{(j)}, E_m \in \mathcal{E}\} \tag{4.1.11}$$

converges.

Theorem 4.1.1. *The algorithm* (4.1.8) *converges to the unique solution of* (4.1.1) *or* (4.1.2) *respectively.*

Proof. As explained in Theorem 2.2 in [GSa,P, p.6], the algorihm (4.1.8) converges if for each intermediate iterate w_n that is not a minimizer of the functional J, there is a direction $\lambda_n E_m$ such that the energy decreases in this direction, i.e. $\exists z \in \mathbb{R} : J(w_n + z\lambda_n E_m) < J(w_n)$. The edges used in the algorithm (4.1.8) given by (4.1.11) fulfill the requirement for convergence, by Lemma 5.1 in [GSa,P, p.17] or Corollary 5.1. in [GSa,P, p.20].
Convergence of the inexact version of the polyhedral Gauss-Seidel-method using the search directions (4.1.11) follows from Theorem 4.1 in [GSa,P, p.16]. □

In both the anisotropic and the isotropic case, the convex set is marked by a lower $\underline{\Psi}_{n,m} = -w_{n-1}(\lambda_n)_i$ and an upper $\overline{\Psi}_{n,m} = w_{n-1}(\lambda_n)_j$ obstacle for each node p_n and each direction $m(i,j)$, to ensure that each component of φ is nonnegative.
It is easy to see that the sum of the components of the corrections at each node p_n fulfills

$$\sum_{l=1}^{2N} \left(\sum_{m=1}^{M} corr_{n,m} \right)_l = 0.$$

Thus, the sum of the components of an iterate φ^ν cannot change. As we want to stay inside the discrete admissible set $\mathcal{G}_j$ (3.2.4), the components of the initial function φ^0 we use for a Gauss–Seidel-step, also have to sum up to one.
The polyhedral Gauss–Seidel-alogrithm (4.1.8) is referred to as an **edge-oriented fine grid smoother**, as we are solving on a fine grid and the oscillations of the error will be smoothed out much faster than the error will be reduced.

Now we want to take a look at the convergence rate of the algorithm (4.1.8). Applied to a quadratic minimization problem with box constraints

$$v \in K : J_0^K(v) \leq J_0^K(w), \quad \forall w \in K \tag{4.1.12}$$

where the box $K = \prod_{i=1}^{d} [\underline{\Psi}_i, \overline{\Psi}_i]$ is independent of the current iterate and the functional $J_0^K = \frac{1}{2}a(.,.) - l(.)$ contains a bilinear form $a(.,.)$, the convergence rate is given by [GK, p.8]

$$||\varphi^{\nu+1} - \varphi||^2 \leq \left(1 - Ch_j^2\right) ||\varphi^\nu - \varphi||^2 \tag{4.1.13}$$

depending on the mesh size h_j (3.2.1) and on a constant $C > 0$ independent of h_j. $||.||$ denotes the energy norm associated with the bilinear form $a(.,.)$. As our problem contains the more complicated simplex constraints $\chi_{\mathcal{G}_j}$ and the nonquadratic anisotropy function, we have to expect an even worse convergence rate.

4.2 Algebraic formulation

Now we want to take a look at the algebraic system we obtain when using the nodal basis (3.2.3). The coefficient vector for φ is given by $\vec{\varphi} \in (\mathbb{R}^{2N})^{\#\mathcal{N}_j}$, with $\vec{\varphi}_k \in \mathbb{R}^{2N}$, for

$k = 1, ..., \#\mathcal{N}_j$.
The minimization problem (4.1.2) transforms to

$$\vec{\varphi} \in G^{\#\mathcal{N}_j} : \epsilon NL(\vec{\varphi}) + \frac{\epsilon}{2\triangle t}\Big\langle M\vec{\varphi}, \vec{\varphi}\Big\rangle - \Big\langle \vec{b}, \vec{\varphi}\Big\rangle \leq$$

$$\leq \epsilon NL(\vec{v}) + \frac{\epsilon}{2\triangle t}\Big\langle M\vec{v}, \vec{v}\Big\rangle - \Big\langle \vec{b}, \vec{v}\Big\rangle, \quad \forall\ \vec{v} \in G^{\#\mathcal{N}_j}. \tag{4.2.1}$$

The variational equality (4.1.3) transforms to

$$\epsilon\langle \nabla NL(\vec{\varphi}), \vec{v} - \vec{\varphi}\rangle + \frac{\epsilon}{\triangle t}\langle M\vec{\varphi}, \vec{v} - \vec{\varphi}\rangle \leq \langle \vec{b}, \vec{v} - \vec{\varphi}\rangle, \quad \forall \vec{v} \in (\mathbb{R}^{2N})^{\#\mathcal{N}_j}. \tag{4.2.2}$$

In (4.2.1) and (4.2.2), the mass matrix $M \in \big(\mathbb{R}^{2N\times 2N}\big)^{\#\mathcal{N}_j \times \#\mathcal{N}_j}$ is given by

$$M_{i,j} = Id \int_\Omega \lambda_i \lambda_j \in \mathbb{R}^{2N\times 2N}$$

and the discrete equivalent to the nonlinearity resulting from the anisotropic part is given by

$$NL(\vec{\varphi}) = \frac{1}{2}\sum_{l=1}^{\#\mathcal{T}_j}\sum_{i=1}^{N} \gamma_i^2(D_{i,l}\vec{\varphi})\ \mid \tau_l \mid \tag{4.2.3}$$

with $D_{l,i}\vec{\varphi} := \big(\nabla\varphi_i + \nabla\varphi_{i+N}\big)_{|\tau_l} = \sum_{k=1}^{\#\mathcal{N}_j}\big(\vec{\varphi}_i + \vec{\varphi}_{i+1}\big)_k\big(\nabla\lambda_{k|\tau_l}\big) \in \mathbb{R}^d$. $\vec{b}$ is the vector corresponding to l.
Using the notation above, we denote the algebraic equivalent of the functional J_0 by

$$\vec{J}_0(\vec{\varphi}) = \epsilon NL(\vec{\varphi}) + \frac{\epsilon}{2\triangle t}\Big\langle M\vec{\varphi}, \vec{\varphi}\Big\rangle - \Big\langle \vec{b}, \vec{\varphi}\Big\rangle. \tag{4.2.4}$$

In the isotropic case, the nonlinearity (4.2.3) turns into a linearity and can be written as $NL(\vec{\varphi}) = \frac{1}{2}\langle S\vec{\varphi}, \vec{\varphi}\rangle$ with the stiffness matrix

$$S_{i,j} = E\int_\Omega \nabla\lambda_i \nabla\lambda_j$$

where $E \in \mathbb{R}^{2N\times 2N}, E_{k,k} = 1, E_{k+N,k} = 1, E_{k,k+N} = 1$.
We can solve (4.2.1) or (4.2.2) respectively using for instance the polyhedral Gauss–Seidel-method (4.1.8) from the last section.

In order to obtain u we will solve the discrete version of the variational equality (3.1.7) of the form

$$u \in S_j\colon \quad a(u,u) = l(u), \quad \forall u \in S_j. \tag{4.2.5}$$

Using the nodal basis (3.2.3) $\Lambda^{(j)}$ with (4.2.5), we obtain the following discrete variational problem

$$\vec{u} \in (S_j)^{\#\mathcal{N}_j}\colon \quad a(\vec{u},\vec{u}) = l(\vec{u}), \quad \forall \vec{u} \in (S_j)^{\#\mathcal{N}_j}$$

$$\Leftrightarrow \Big(\frac{\beta\epsilon}{\triangle t}M + S + \alpha B\Big)\vec{u} = \vec{b}$$

where $\vec{u}$ is the coefficient vector for the solute u, M is the mass matrix, $S = (D(\varphi^{k+1})\nabla\lambda_i, \nabla\lambda_l)_{i,l=1,...,\#\mathcal{N}_j}$ is the stiffness matrix, $B = \big((D(\varphi^{k+1})\nabla\lambda_i, \nabla\lambda_l)_{\partial\Omega_h}\big)_{i,l}$ is the matrix resulting from the integral on the boundary and $\vec{b}$ is the discrete version of the functional l. Here, a normal Gauss–Seidel-method can be used, because there are no constraints and the problem is linear.

In the following, when we describe algorithms, we will mostly present just the version that has been identified with a nodal basis, because transforming the algorithm always works analogously.

Until now, we only have a fairly slow convergence rate given by (4.1.13). Therefore, in the next section, we will take a look at more sufficient methods to solve our problems than just a Gauss–Seidel-method on a fine grid, namely multigrid methods.

4.3 Multilevel relaxation methods for nonlinear problems

The main aspect about multigrid or subspace correction methods is that we do not just solve our problem on the finest grid, but also include coarser ones. Later, we will present the multigrid algorithm we used in the implementation, the **truncated nonsmooth newton mulitgrid method (TNNMG)**, which was developed in [Gr].

In order to apply a multigrid method, we need the to define what a refinement and a grid hierarchy are.

Definition 4.3.1 (refinement)**.** *For $\mathcal{T}_1$ and $\mathcal{T}_2$ being triangulations of Ω_h, we call $\mathcal{T}_2$ a refinement of $\mathcal{T}_1$, if*

$$\forall\tau \in \mathcal{T}_1 : \{\tau' \in \mathcal{T}_2 \;\big|\; \tau' \cap \tau \neq \emptyset\} \text{ is a triangulation of } \tau.$$

[Gr, p.23]

Like in [GK, p.3], we assume that $\mathcal{T}_0$ is shape regular, so it consists of a finite, small number of non-degenerate simplices. To preserve the shape regularity, we refine uniformely, hence according to [GK, p.4] we divide each $\tau \in \mathcal{T}_{k-1}$ into four congruent subtriangles in 2D and into eight sub-tetrahedra in 3D to obtain $\mathcal{T}_k$.

Definition 4.3.2 (hierarchy)**.** *A tuple $(\mathcal{T}_0, ...\mathcal{T}_j)$ is called a grid or refinement hierarchy of Ω_h if $\mathcal{T}_0$ is a conforming triangulation of Ω_h and if each $\mathcal{T}_i$, $i = 1, ..., j$ is a refinement of $\mathcal{T}_{i-1}$. [Gr, p.24]*

From this point on, let $\mathcal{T}_j$ be a refined conforming triangulation resulting from j successive refinements from a coarse conforming triangulation $\mathcal{T}_0$ of Ω_h. Concerning the mesh size (3.2.1), we see that

$$h_j = \mathcal{O}(2^{-j}).$$

We want to solve our problem not within the whole space S_j but split S_j into subspaces

$$S_j = V_0 + ... + V_j$$

of lower dimension $dim(S_j) > dim(V_k) =: n_k$ and solve in all V_k consecutively. The refinement hierarchy already proposes a subspace hierarchy together with corresponding nodal bases

$$V_k = \left\{\chi \in C^0(\Omega_h) \;\middle|\; \chi \text{ is linear on each } T \in \mathcal{T}_k\right\} = S_k \quad \text{and} \quad \Lambda^{(k)} = \left\{\lambda_i^{(k)} \mid p_i \in \mathcal{N}_k\right\}. \tag{4.3.1}$$

A natural embedding of $V_k \subset V_{k+1}$ corresponding to the identity is given by transfer operators $W^{(k)} \in \mathbb{R}^{n_{k+1} \times n_k}$ [Gr, p.70] that fulfill

$$\lambda_i^{(k)} = \sum_{m=1}^{n_{k+1}} (W^{(k)})_{mi}\, \lambda_m^{(k+1)}$$

and represent the connection of the basis functions on different levels with each other. As we are using linear finite elements, the operator $W^{(k)}$ has to interpolate along the edges of the simplices, hence $W^{(k)}$ is given by

$$(W^{(k)})_{mi} = \begin{cases} 1 & \text{for } p_m = p_i \\ \frac{1}{2} & \text{for } p_m \text{ is a neighbor of } p_i, p_m \neq p_i \\ 0 & \text{else} \end{cases} \tag{4.3.2}$$

for $p_m \in \mathcal{N}_{k+1}$, $p_i \in \mathcal{N}_k \subset \mathcal{N}_{k+1}$ [Kh, p.64].
For $k' < k$ we have [Gr, p.70]

$$\lambda_i^{(k')} = \sum_{m=1}^{n_{k+1}} (W^{(k)} \cdots W^{(k')})_{mi} \lambda_m^{(k+1)}.$$

For two representations $v^{(k+1)}$ and $v^{(k)}$ with coefficient vectors $\vec{v}^{(k+1)}$ and $\vec{v}^{(k)}$ of the same function on two different levels the transfer operators $W^{(k)} : V_k \to V_{k+1}$ satisfy

$$\vec{v}^{(k+1)} = W^{(k)} \vec{v}^{(k)}.$$

One step of a multigrid method will apply a smoothing step in each subspace. A general multigrid algorithm for some iterate v^ν looks like:

1. initialize $w_{-1} = v^\nu$

2. for $k = 0, ..., j$:

$$\{ \quad corr_k \in V_k \; : corr_k = \underset{v \in V_k}{\operatorname{argmin}} J_{|V_k}(w_{k-1} + v) \tag{4.3.3}$$

$$\text{set } w_k = w_{k-1} + corr_k \quad \}$$

3. set $v^{\nu+1} = w_j$

The minimization problems on the subspaces V_k can be solved using for instance a Gauss–Seidel-method like in the section before.

To obtain the phase field function φ, we can solve (4.1.2) using the multilevel nodal basis $\cup_{i=1,...,j}\Lambda^{(i)}$ [KKSG, p.4] that covers all levels. We get the following algorithm.

1. initialize $w_{0,0} = \varphi^{\nu}$

2. for $k = 0, ..., j$:
   { for $n = 1, ..., dim(\Lambda^{(k)})$:
   { for $m(i,j) = 1, ..., M$:

$$\{ \quad \text{set} \quad corr_{k,n,m} = \underset{\substack{v \in span\left\{\lambda_n^{(k)} E_m\right\} \\ w_{k,n-1}+v \in \mathcal{G}_j}}{\operatorname{argmin}} J(w_{k,n-1}+v)$$

$$\text{update} \qquad w_{k,n-1} = w_{k,n-1} + corr_{k,n,m} \qquad \} \tag{4.3.4}$$

   set $w_{k,n} = w_{k,n-1}$ }
   if $k < j$: set $w_{k+1,1} = w_{k,dim(\Lambda^{(k)})}$ }

3. set $\varphi^{\nu+1} = w_{j,dim(\Lambda^{(j)})}$

Then $w_{k,n}$ is the iterate on level k after the corrections at the node p_n.
When identifying the algorithms (4.3.3) or (4.3.4) with a nodal basis, we have to use the transfer matrices (4.3.2) to add the correction to the intermediate iterate in step 2. If J is linear, we can calculate the algebraic version $\vec{J}$ once on the finest grid and then use the transfer matrices $W^{(k)}$ to consecutively transfer $\vec{J}$ to the coarser levels. However, this does not work if the functional J is nonlinear, like in the anisotropic case, and we have to completely recalculate the nonlinear parts of $\vec{J}$ on each level.
In the simpler case (4.1.12) with box constraints and a quadratic functional, the convergence rate of the algorithm (4.3.4) is given by the following Theorem. Again, as we consider a more complicated problem with simplex constraints $\chi_{\mathcal{G}_j}$ and a nonlinear anisotropy function, we have to expect a worse convergence rate.

Theorem 4.3.3. *Let $j' = \sum_{i=1}^{j} \#\mathcal{N}_i$ and $\cup_{i=1,...,j}\Lambda^{(i)} = span\{\lambda_1, ..., \lambda_{j'}\}$. Assume we have a decomposition*

$$v = \sum_{l=1}^{j'} v_l \quad \text{with} \quad v_l \in span\{\lambda_l\} \quad \text{that satisfies} \quad \sum_{l=1}^{j'} ||v_l||^2 \leq C_0 ||v||^2$$

with a constant $C_0 > 0$ for any $v \in span\{\lambda_1, ..., \lambda_{j'}\}$.
Moreover assume that $\exists\ C_1 > 0$ such that

$$\sum_{i,j=1}^{j'} |a(v_i, w_j)| \leq C_1 \Big(\sum_{i=1}^{j'} ||v_i||^2 \Big)^{\frac{1}{2}} \Big(\sum_{j=1}^{j'} ||w_j||^2 \Big)^{\frac{1}{2}}$$

holds for all $w_j \in span\{\lambda_j\}$ *and* $v_i \in span\{\lambda_i\}$. *Then the iterates* $v^{\nu+1}$ *from* (4.3.4) *satisfy the error estimate*

$$||v^{\nu+1} - v||^2 \leq \left(1 - \frac{1}{C_0(1-C_1)^2}\right)||v^{\nu} - v||^2 \tag{4.3.5}$$

where v *is the correct solution in the solution space and* $||.||$ *denotes the energy norm associated with the bilinear form* $a(.,.)$ *of the functional* J. *[GK, p.7]*

Using the Cauchy–Schwarz-inequality, we see that $C_1 \leq j'$ [GK, p.8]. We cannot find an upper bound for C_0 this easy though. Therefore, the convergence of this algorithm depends a lot on the choice of the subspaces V_k. For a quadratic functional, it is clear that we get the fastest convergence if we choose $a(.,.)$-orthogonal subspaces V_k, because then we obtain a Conjugate–Gradient-method. As a coarse grid basis function is almost $a(.,.)$-orthogonal to a fine grid basis function, the multilevel nodal basis $\cup_{i=1,\dots,j}\Lambda^{(i)}$ is a much better choice than only solving on the finest grid. This is the reason why the convergence result (4.3.5) is better than (4.1.13).
Actually, we can obtain (4.1.13) when using $C_0 = \mathcal{O}(h_j^{-2})$ and $C_1 = \mathcal{O}(1)$ in equation (4.3.5) [GK, p.8].

However, the better convergence result of algorithm (4.3.4) is accompanied by a higher effort. We can compute the correction $corr_{k,n,m}$ just like in (4.1.9), but it is very costly to evaluate the exact obstacles $(\underline{\Psi}^{(k)})_{n,m(i,j)}$ and $(\overline{\Psi}^{(k)})_{n,m(i,j)}$ and completely recalculate the nonlinearity resulting from the anisotropic part on each level for each minimization step.
To avoid recalculating the nonlinarity completely on each level k, the truncated nonsmooth newton mulitgrid method can be used, where the actual equation is only solved on the finest space and a linearized version is solved on the coarse spaces.
The exact obstacles $(\underline{\Psi}^{(k)})_{n,m(i,j)}$ and $(\overline{\Psi}^{(k)})_{n,m(i,j)}$ for the coarse corrections at node n on level k can be calculated by

$$(\overline{\Psi}^{(k)})_{n,m(i,l)} = \max\big\{ - w_{k,n-1}(\lambda_r^{(j)})_l\big(\lambda_n^{(k)}(p_r)\big)^{-1} \mid \forall p_r \in \mathcal{N}_j \cap \operatorname{supp}\lambda_n^{(k)}\big\}$$

$$\text{and } (\underline{\Psi}^{(k)})_{n,m(i,l)} = \min\big\{w_{k,n-1}(\lambda_r^{(j)})_i\big(\lambda_n^{(k)}(p_r)\big)^{-1} \mid \forall p_r \in \mathcal{N}_j \cap \operatorname{supp}\lambda_n^{(k)}\big\}.$$

When computing these coarse obstacles one has to go up to the fine level j to compute the defect obstacles. Therefore, the exact obstacles can only be used at the price of a higher complexity, namely $\mathcal{O}(n_j)$ evaluations of the basis functions at each level k.
An idea to avoid the high coplexity is to a priorily construct a convex set $K_{k,n-1}$ on the level k using the finest obstacles from level j. This convex set $K_{k,n-1}$ must be completely contained in the exact shifted convex set $\mathcal{G}_j - w_{k,n-1} \supset K_{k,n-1}$. Then, we look for a correction $corr_{k,n,m} \in V_k \cap K_{k,n-1}$. However, this idea leads to a slow convergence, as $K_{k,n-1}$ might be much smaller than the exact coarse obstacle so we can probably only apply very small coarse corrections.
In the TNNMG algorithm, the decomposition of Ω into an active Ω_A (2.2.13) and an

inactive set Ω_I (2.2.12) is used to avoid an unefficient calculation of the obstacles.

For the solute function u, we can use a multigrid method just like (4.3.3), because the variational equality (4.2.5) is completely linear and does not involve any constraints.

4.4 Truncated Nonsmooth Newton Multigrid Method

Now we will present the algorithm actually used to solve our minimization problems (4.1.1) or (4.1.2) respectively.
In the first part of this section, we will develop the idea of the TNNMG method for the simpler, box constrained, linear problem (4.1.12) like in [Gr] and then transfer the algorithm to our specific, simplex constrained, nonlinear minimziation problem (4.1.1) or (4.1.2) respectively.

4.4.1 Motivation of the Truncated Nonsmooth Newton Multigrid Method

In this section we look at the minimization problem (4.1.12)

$$\vec{v} \in K : \vec{J}_0^K(\vec{v}) \leq \vec{J}_0^K(\vec{w}), \quad \forall \vec{w} \in K$$

with the discrete functional $\vec{J}_0^K = \frac{1}{2}A\langle .,. \rangle - \langle b, . \rangle : \mathbb{R}^d \to \mathbb{R}$, where A and $\vec{b}$ are the discrete versions of the bilinear form $a(.,.)$ and $l(.)$. The constraints are given by the box $K = \prod_{i=1}^d [\underline{\Psi}_i, \overline{\Psi}_i]$. Equivalently, we can look at

$$\vec{v} \in \mathbb{R}^d : \vec{J}_0^K(\vec{v}) + \chi_K =: \vec{J}^K(\vec{v}) \leq \vec{J}^K(\vec{w}), \quad \forall \vec{w} \in \mathbb{R}^d.$$

We want to develop a multigrid method that solves the problem exactly on the finest grid $\mathcal{T}_j$ and solves only a linearization on the coarser grids.
Applying this idea to (4.1.12), we obtain a the following algorithm.

1. initialize $\vec{w}_0 = \vec{v}^\nu$
2. for $n = 1, ..., \#\mathcal{N}_j$:
$$\{ \quad \vec{corr}_n = \underset{\substack{\vec{u} \in span\{e_n\} \\ \vec{w}_{n-1} + \vec{u} \in K}}{\operatorname{argmin}} \vec{J}_0^K(\vec{w}_{n-1} + \vec{u}) \qquad (4.4.1)$$
$$\text{set } \vec{w}_n = \vec{w}_{n-1} + \vec{corr}_n \quad \}$$
3. set $\vec{w} = \vec{w}_{\#\mathcal{N}_j} + \vec{corr}_{coarse}$, where $\vec{corr}_{coarse} \in K - \vec{w}_{\#\mathcal{N}_j}$ and $\vec{J}_0^K(\vec{w}_{\#\mathcal{N}_j} + \vec{corr}_{coarse}) \leq \vec{J}_0^K(\vec{w}_{\#\mathcal{N}_j})$
4. set $\vec{v}^{\nu+1} = \vec{w}$

In step 3, we don't require $\vec{w}_n + \vec{corr}_{coarse}$ to be the minimum of $\vec{J}_0^K$ in $K - \vec{w}_n$, we only require a decrease of the energy when adding $\vec{corr}_{coarse}$ to the iterate.

Step 3, the correction on the coarse space, will be derived as a modification of the Gauss–Seidel-method. A Gauss–Seidel-step for (4.1.12) is given by

$$\vec{v}^{\nu+1} = \vec{v}^{\nu} + \big(D + L + \partial\chi_K\big)^{-1}(\vec{b} - A\vec{v}^{\nu})$$

where the matrix $A = D+L+R$ is decomposed into diagonal, lower left and upper right part. This Gauss–Seidel-step corresponds directly to step 2 and can also be written as a correction operator

$$\mathcal{F}_{GS}(\vec{v}^{\nu}) = (D + L + \partial\chi_K)^{-1}(\vec{b} - R\vec{v}^{\nu}) - \vec{v}^{\nu}. \tag{4.4.2}$$

The Gauss–Seidel-method fulfills the following Theorem.

Theorem 4.4.1. *$\mathcal{F}_{GS}$ (4.4.2) is a well-defined, Lipschitz–continuous operator. It fulfills the so-called monotonicity assumption*

$$\mathcal{F}_{GS}(\vec{v}^{\nu}) = 0 \quad \Leftrightarrow \quad \nabla\vec{J}^{K}(\vec{v}^{\nu}) = 0 \tag{4.4.3}$$

and the so-called convergence assumption

$$\vec{v}^{\nu} \underset{\nu\to\infty}{\to} \vec{v} \;\Rightarrow\; \vec{v} \text{ is a minimizer of } \vec{J}^{K}, \tag{4.4.4}$$

so a sequence given by

$$\vec{v}^{\nu+1} = \vec{v}^{\nu} + \mathcal{F}_{GS}(\vec{v}^{\nu})$$

converges to the minimum of $\vec{J}^{K}$. [Gr, p.52,53]

Considering Theorem 4.4.1, we can regard an application of the correction operator $\mathcal{F}_{GS}$ (4.4.2) as a fixed point iteration of $Id + \mathcal{F}_{GS}$, hence it suggests itself to apply Newton-type methods to find the root of $\mathcal{F}_{GS}$. We will use this idea to compute the correction in step 3 from algorithm (4.4.1) by

$$\vec{corr}_{coarse} = -\big("\partial\mathcal{F}_{GS}(\vec{v}^{\nu})"\big)^{-1}\mathcal{F}_{GS}(\vec{v}^{\nu}). \tag{4.4.5}$$

However, $\mathcal{F}_{GS}$ is not differentiable everywhere, as it contains $\partial\chi_K$, which is set-valued outside of K. Newton-type methods require the functional to be differentiable and therefore, like in [Gr, p.58], we will compute a generalized linearization of the operator instead.
Before that, we want to reformulate $\mathcal{F}_{GS}$ to be able to compute a derivative using a chain rule. From (4.4.2), elementary calculations yield

$$\mathcal{F}_{GS}(\vec{v}^{\nu}) = (D + \partial\chi_K)^{-1}\big(\vec{b} - (R+L)\vec{v}^{\nu} - L\mathcal{F}_{GS}(\vec{v}^{\nu})\big) - \vec{v}^{\nu}.$$

Then, assuming a chain rule we get the generalized linearization [Gr, p.60]

$$\partial\mathcal{F}_{GS}(\vec{v}^{\nu}) := \partial\big((D+\partial\chi_K)^{-1}\big)\big(\vec{b}-(R+L)\vec{v}^{\nu}-L\mathcal{F}_{GS}(\vec{v}^{\nu})\big)\big(-(R+L)-L\partial\mathcal{F}_{GS}(\vec{v}^{\nu})\big)-Id. \tag{4.4.6}$$

We already know that $\big((D+\partial\chi_K)^{-1}\big)$ is not differentiable, because $\partial\chi_K$ is set valued outside of K. As a substitute, we define the generalized linearizations $\mu_i : \mathbb{R} \to \mathbb{R}$ (plotted in Figure 2) by

$$\mu_i(x) := \big(A_{ii}(.) + \partial\chi_{[\underline{\Psi}_i,\overline{\Psi}_i]}(.)\big)^{-1}(x) = \begin{cases} A_{ii}^{-1}x & \text{if } A_{ii}^{-1}x \in [\underline{\Psi}_i, \overline{\Psi}_i] \\ \underline{\Psi}_i & \text{if } A_{ii}^{-1}x \leq \underline{\Psi}_i \\ \overline{\Psi}_i & \text{if } A_{ii}^{-1}x \geq \overline{\Psi}_i. \end{cases}$$

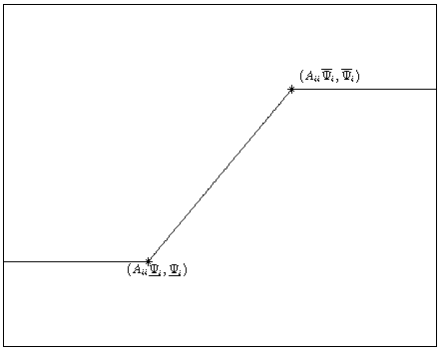


Figure 2: μ_i

The linearizations of μ_i are then given by [Gr, p.59]

$$\partial\mu_i(x) = \begin{cases} A_{ii}^{-1} & \text{if } A_{ii}^{-1}x \in [\underline{\Psi}_i, \overline{\Psi}_i] \\ 0 & \text{else.} \end{cases}$$

It is easy to see that μ_i is the i-th component of $(D+\partial\chi_K)^{-1}$, so we can use all μ_i, like in [Gr, p.60], to rewrite the generalized linearization (4.4.6) of $\mathcal{F}_{GS}$. We obtain

$$\partial\mathcal{F}_{GS}(\vec{v}^{\nu}) = \operatorname*{diag}_{i=1,\dots,\#\mathcal{N}_j}(\partial\mu_i(\vec{r}_i))\big(-(R+L) - L\partial\mathcal{F}_{GS}(\vec{v}^{\nu})\big) - Id \tag{4.4.7}$$

where $\vec{r}_i = \big(\vec{b} - (R+L)\vec{v}^{\nu} - L\mathcal{F}_{GS}(\vec{v}^{\nu})\big)_i$. We see that the matrix $\operatorname{diag}(\partial\mu_i(\vec{r}_i))$ is singular, as some rows and columns only contain zeros. These are exactly the rows and columns corresponding to the nodes where the obstacle K is touched, i.e. all nodes p_i, where $\mu_i(r_i)$ is equal to $\underline{\Psi}_i$ or $\overline{\Psi}_i$. This part of Ω_h is called active, as there the constraints are active, while the rest of Ω_h is called inactive. We can truncate these active rows and columns of the matrix $\operatorname{diag}(\partial\mu_i(\vec{r}_i))$ using the notation of truncated matrices.

Definition 4.4.2 (truncated)**.** *Let $I, J \in \mathbb{N}$ be index sets, $M \in \mathbb{R}^{n\times m}$ a matrix and $\vec{v} \in \mathbb{R}^n$ a vector. The truncated matrix $M_{I,J}$ and the truncated vector $\vec{v}_I$ are given by*

$$(M_{I,J})_{i,j} = \begin{cases} M_{i,j} & \textit{if } i \in I \textit{ and } j \in J \\ 0, & \textit{else} \end{cases}$$

$$\text{and} \quad (\vec{v}_I)_i = \begin{cases} \vec{v}_i, & \text{if } i \in I \\ 0, & \text{else} \end{cases}$$

We define the abbrevation $M_{I,I} =: M_I$.
[Gr, p.59]

Generally, truncated matrices are singular, but we can still invert them using the so-called Moore-Penrose pseudoinverse.

Definition 4.4.3 (Moore–Penrose pseudoinverse)**.** *Let* $M \in \mathbb{R}^{m\times n}$. *The Moore-Penrose pseudoinverse* $M^+ \in \mathbb{R}^{n\times m}$ *is then defined by*

$$M^+ := \lim_{\epsilon\to 0} \big(M^T M + \epsilon\, Id\big)^{-1} M^T = \lim_{\epsilon\to 0} M^T \big(MM^T + \epsilon\, Id\big)^{-1}.$$

The limits exist even for singular, especially truncated matrices. The pseudoinverse can also be characterized by

$$MM^+M = M, \quad M^+MM^+ = M^+, \quad \big(M^+M\big)^T = MM^+, \quad \big(MM^+\big)^T = M^+M.$$

If $M \in \mathbb{R}^{n\times n}$ *is regular, then the pseudoinverse equals the regular inverse* M^{-1}. *Moreover*

$$M^{++} = M, \qquad \big(M^T\big)^+ = \big(M^+\big)^T, \qquad \big(\lambda M\big)^+ = \frac{1}{\lambda} M^+ \quad \forall \lambda > 0.$$

[Gr, p.148,149]

Now, using the inactive set for a vector $\vec{v}$

$$\mathcal{I}(\vec{v}) = \{i \mid \partial\chi_{[\underline{\Psi}_i, \overline{\Psi}_i]}\big(\vec{v}_i\big) \text{ is single-valued}\}, \tag{4.4.8}$$

containing all inactive nodes, the representaion $\mu_i(\vec{r}_i) = \vec{v}_i^{\,\nu} + (\mathcal{F}_{GS}(\vec{v}^{\,\nu}))_i$ and the Moore–Penrose pseudoinverse, we can rewrite (4.4.7) like in [Gr, p.60] and obtain

$$\partial\mathcal{F}_{GS}(\vec{v}^{\,\nu}) = D^+_{\mathcal{I}(\vec{v}^{\,\nu}+\mathcal{F}_{GS}(\vec{v}^{\,\nu}))}\big(-(R+L) - L\partial\mathcal{F}_{GS}(\vec{v}^{\,\nu})\big) - Id.$$

Elementary calculations yield the linearization of the operator $\mathcal{F}_{GS}$ (4.4.2) [Gr, p.60]

$$\partial\mathcal{F}_{GS}(\vec{v}^{\,\nu}) = -\big(D+L\big)^+_{\mathcal{I}(\vec{v}^{\,\nu}+\mathcal{F}_{GS}(\vec{v}^{\,\nu}))} R_{\mathcal{I}(\vec{v}^{\,\nu}+\mathcal{F}_{GS}(\vec{v}^{\,\nu}))} - Id. \tag{4.4.9}$$

Now we can use the Newton-type method (4.4.5) to find $\vec{corr}_{coarse}$ in step 3 of algorithm (4.4.1), as explained in the following Theorem.

Theorem 4.4.4. *The iteration resulting from algorithm* (4.4.1) *can be rewritten as*

$$\begin{aligned} \vec{v}^{\,\nu+\frac{1}{2}} &= \vec{v}^{\,\nu} + \mathcal{F}_{GS}(\vec{v}^{\,\nu}) \\ \vec{v}^{\,\nu+1} &= \vec{v}^{\,\nu+\frac{1}{2}} + \mathcal{C}(\vec{v}^{\,\nu+\frac{1}{2}}). \end{aligned} \tag{4.4.10}$$

The correction operator $\mathcal{C}$ *is given by the newton step* (4.4.5) *and can be written as*

$$\mathcal{C}(\vec{v}) := A^+_{\mathcal{I}(\vec{v})}(b - A\vec{v})_{\mathcal{I}(\vec{v})}. \tag{4.4.11}$$

[Gr, p.61]

We see that the correction operator $\mathcal{C}$ (4.4.11) can be equivalently expressed by

$$\mathcal{C}(\vec{v}) = -\big((\vec{J}^K)''\big)^+_{\mathcal{I}(\vec{v})}\big((\vec{J}^K)'(\vec{v})\big)_{\mathcal{I}(\vec{v})} \tag{4.4.12}$$

using the truncated Hessian and the truncated gradient of the functional $\vec{J}^K$. As constructed, $\mathcal{C}$ corresponds to a Newton-step in the subspace $\mathcal{I}(\vec{v})$. $\vec{J}^K$ is smooth in this subspace, because $\mathcal{I}(\vec{v})$ contains all inactive nodes. The linear subproblem (4.4.11) we have to solve in each iteration step on the coarse levels, has a nice structure, as the matrix $-\big(\vec{J}^K\big)''_{\mathcal{I}(\vec{v})}$ is symmetric and positive semidefinite. Its kernel is given by the Eucledian unit vectors e_i with $i \notin \mathcal{I}(\vec{v})$ (4.4.8), as we truncated these rows and columns. The matrix $-\big(\vec{J}^K\big)''_{\mathcal{I}(\vec{v})}$ is even positive definite on its range $\text{span}\{e_i \mid i \in \mathcal{I}(\vec{v})\}$.

Now we developed the desired algorithm with the help of the box-constrained minimization problem (4.1.12). In the following we want to take a look on how to solve the truncated algebraic system we obtain form the correction operator $\mathcal{C}$ (4.4.11).
We want to solve the system

$$H_{\mathcal{I}(\vec{v})}\vec{c} = \vec{d}_{\mathcal{I}(\vec{v})}, \tag{4.4.13}$$

where $-\big(\vec{J}^K\big)'' =: H$, $\big(\vec{J}^K\big)'(\vec{v}) =: \vec{d}$ and $\vec{corr}_{coarse} =: \vec{c}$. A general multigrid algorithm for the nontruncated system $H\vec{c} = d$ is given by [Gr, p.70]:

1. initialize $H_j = H$, $\vec{r}_j = \vec{d} - H_j\vec{c}^{\nu}$
2. for $k = j, ..., 0$:

$$\begin{aligned} \{ \quad & \vec{v}_k = (B_k)^{-1}\vec{r}_k \\ \text{if } k > 0 \text{ set: } & \vec{r}_{k-1} = (W^{(k-1)})^T\big(\vec{r}_k - H_k\vec{v}_k\big) \\ \text{and } & H_{k-1} = (W^{(k-1)})^T H_k W^{(k-1)} \quad \} \end{aligned}$$

3. set $\vec{c}^{\nu+1} = \vec{c}^{\nu} + \sum_{k=0}^{j}\vec{v}_k$

Here, B_k is an approximation of H_k, that is easy to invert. For instance, we can choose B_k equal to the diagonal or the diagonal and the lower left part of H_k. In the algorithm above we use the so-called residual $\vec{r}_j = \vec{d} - H_j\vec{c}^{\nu}$ and the transfer operators $W^{(k)}$ (4.3.2).
When transferring this algorithm to the truncated system (4.4.13), we have to apply small modifications, like in [Gr, p.71]. We get:

1. initialize $\widetilde{H}_j = H_{\mathcal{I}(\vec{v})}$, $\vec{r}_j = \vec{d}_{\mathcal{I}(\vec{v})} - \widetilde{H}_j\vec{c}^{\nu}$
2. for $k = j, ..., 0$:

$$\begin{aligned} \{ \ & \vec{v}_k = (\widetilde{B}_k)^+\vec{r}_k \\ \text{if } k > 0 \text{ set: } & \vec{r}_{k-1} = (W^{(k-1)})^T\big(\vec{r}_k - \widetilde{H}_k\vec{v}_k\big) \\ \text{and } & \widetilde{H}_{k-1} = \big(W^{(k-1)}\big)^T\widetilde{H}_k W^{(k-1)} \ \} \end{aligned} \tag{4.4.14}$$

3. set $\vec{c}^{\nu+1} = \vec{c}^{\nu} + \sum_{k=0}^{j} \vec{v}_k$

As we are using truncated matrices, generally $\widetilde{B}_k$ cannot be inverted, as it might be singular, so instead, we use the Moore–Penrose pseudoinverse $\widetilde{B}_k^*$.

Before we transfer the TNNMG algorithm to the simplex-constrained case in the next section, we give a general convergence result for an iteration of type (4.4.10) in the following Lemma.

Lemma 4.4.5. *Let $\vec{J}$ be a strictly convex, lower-semi-continuous, coercive and proper functional. Let the correction operator $\mathcal{F}$ fulfill the monotonicity assumption* (4.4.3) *and the convergence assumption* (4.4.4) *and let the second correction operator $\mathcal{C}$ fulfill the monotonicity assumption* (4.4.3)*. Then the sequence $\vec{v}^{\nu}$ generated by*

$$\vec{v}^{\nu+\frac{1}{2}} = \vec{v}^{\nu} + \mathcal{F}(\vec{v}^{\nu})$$

$$\vec{v}^{\nu+1} = \vec{v}^{\nu+\frac{1}{2}} + \mathcal{C}(\vec{v}^{\nu+\frac{1}{2}})$$

converges to the minimum of $\vec{J}$ for all $\vec{v}^0$ with $\vec{J}(\vec{v}^0) < \infty$. [Gr, p.54]

We know from Lemma 4.4.1 that the correction operator $\mathcal{F}_{GS}$ fulfills (4.4.3) and (4.4.4). However, $\mathcal{C}$ (4.1.9) might not fulfill (4.4.3) and in fact, the energy can actually be increased in this step. To avoid an energy increase, it is common in Newton theory to apply damping to the correction operator and replace $\mathcal{C}(\vec{v}^{\nu+\frac{1}{2}})$ by $\rho\ \mathcal{C}(\vec{v}^{\nu+\frac{1}{2}})$. When using

$$\rho = \rho_{argmin} := \underset{z}{\operatorname{argmin}}\ J^K(\vec{v}^{\nu+\frac{1}{2}} + z\,\mathcal{C}(\vec{v}^{\nu+\frac{1}{2}})),$$

then the damping parameter ρ is automatically bounded from above, according to [Gr, p.65], by

$$\min\Big\{ \min_{i\in\mathcal{I}_+} \frac{\overline{\Psi}_i - \vec{v}_i^{\nu+\frac{1}{2}}}{\mathcal{C}(\vec{v}^{\nu+\frac{1}{2}})_i}, \min_{i\in\mathcal{I}_-} \frac{\underline{\Psi}_i - \vec{v}_i^{\nu+\frac{1}{2}}}{\mathcal{C}(\vec{v}^{\nu+\frac{1}{2}})_i} \Big\}$$

with $\mathcal{I}_+ = \{i \,|\, \mathcal{C}(\vec{v}^{\nu+\frac{1}{2}})_i > 0\}$ and $\mathcal{I}_- = \{i \,|\, \mathcal{C}(\vec{v}^{\nu+\frac{1}{2}})_i < 0\}$. This restriction on ρ can lead to arbitrarily small damping parameters causing slow convergence. To avoid this effect, we will orthogonally project the correction $\mathcal{C}(\vec{v}^{\nu+\frac{1}{2}})$ into the shifted convex set $K - \vec{v}^{\nu+\frac{1}{2}}$ on the finest level, before we use damping. The projection of a vector $\vec{w}$ into $K - \vec{v}^{\nu+\frac{1}{2}}$ is given by

$$\Big(P_{\left(K-\vec{\varphi}^{\nu+\frac{1}{2}}\right)}\vec{w}\Big)_i := \max\big\{ \min\{\vec{w}_i, \overline{\Psi}_i - \vec{v}_i^{\nu+\frac{1}{2}}\}, \underline{\Psi}_i - \vec{v}_i^{\nu+\frac{1}{2}}\big\}. \tag{4.4.15}$$

Using this projection together with a damping parameter, we obtain the following Theorem direcly from Lemma 4.4.5.

Theorem 4.4.6. *Let the functional $\vec{J}$ be given as $\vec{J}^K = \vec{J}_0^K + \chi_K$ and let $\vec{J}_0$ be strictly convex, coercive, lower-semi-continuous and proper. Let K be a d-dimensional simplex in $\mathbb{R}^d$ and let the first iterate $\vec{v}^0 \in K$. Then the sequence generated by:*

$$\vec{v}^{\nu+\frac{1}{2}} = \vec{v}^{\nu} + \mathcal{F}_{GS}(\vec{v}^{\nu})$$

$$\vec{v}^{\nu+1} = \vec{v}^{\nu+\frac{1}{2}} + \rho^{\nu} P^{\nu}\big(\mathcal{C}(\vec{v}^{\nu+\frac{1}{2}}) + \vec{\epsilon}^{\nu}\big)$$

with the correction operator $\mathcal{C}$ (4.4.11), the projection $P^{\nu} = P_{\left(K-\vec{v}^{\nu+\frac{1}{2}}\right)}$ (4.4.15) and the damping parameter ρ^{ν} chosen such that

$$J\Big(\vec{v}^{\nu+\frac{1}{2}} + \rho^{\nu} P^{\nu}\big(\mathcal{C}(\vec{v}^{\nu+\frac{1}{2}}) + \epsilon^{\nu}\big)\Big) \leq J\Big(\vec{v}^{\nu+\frac{1}{2}}\Big),$$

converges to the unique solution of (4.1.12) for every error $\vec{\epsilon}^{\nu}$. [Gr, p.65]

The convergent multigrid method in Theorem 4.4.6 and its extension to W- and V-cycles is called **truncated nonsmooth newton multigrid method (TNNMG)**. In the next section we will transfer this algorithm to our minimization problem (4.1.1) or (4.1.2) respectively.

4.4.2 Truncated Nonsmooth Newton Multigrid Method for the simplex-constrained problem

In this section, we will transfer the TNNMG method to our problem. In contrast to the box constrained, linear problem (4.1.12), the problem (4.1.1) or (4.1.2) respectively involves simplex-constraints and a nonlinearity resulting from the anisotropy function. Nevertheless, we can use the TNNMG algorithm, yet with slightly different operators $\mathcal{F}_{GS}$ and $\mathcal{C}$ and a slightly different coarse correction space. Like in [GSa,P, p.2], we get the following amended algorithm

1. initialize $\vec{w}_0 = \vec{\varphi}^{\nu}$
2. for $n = 1, ..., \#\mathcal{N}_j$:
   { for $m = 1, ..., M$:

$$\{ \quad \vec{corr}_{n,m} = \underset{\substack{\vec{u} \in span\{e_n \otimes E_m\} \\ \vec{w}_{n-1}+\vec{u} \in G^{\#\mathcal{N}_j}}}{\operatorname{argmin}} \vec{J}_0(\vec{w}_{n-1} + \vec{u}) \tag{4.4.16}$$

   update $\vec{w}_{n-1} = \vec{w}_{n-1} + \vec{corr}_{n,m}$ }

   set $\vec{w}_n = \vec{w}_{n-1}$ }
3. set $\vec{w} = \vec{w}_{\#\mathcal{N}_j} + \vec{corr}_{coarse}$, where $\vec{corr}_{coarse} \in W(\vec{w}_{\#\mathcal{N}_j})$ and $\vec{J}_0(\vec{w}_{\#\mathcal{N}_j} + \vec{corr}_{coarse}) \leq \vec{J}_0(\vec{w}_{\#\mathcal{N}_j})$
4. set $\vec{\varphi}^{\nu+1} = \vec{w}$

where e_n is the n-th eucledian unit vector in $\mathbb{R}^{\#\mathcal{N}_j}$ and $E_m \in \mathcal{E}$, for $m = 1, ...M$, are the edges of the Gibbs–Simplex G (2.2.1).
First we will look at step 2. In the algorithm (4.4.16), the polyhedral Gauss–Seidel-method (4.1.8) is used instead of the linear Gauss–Seidel-method $\mathcal{F}_{GS}$ (4.4.2). In the anisotropic case, (4.1.8) involves the bisection method (4.1.5) which can be regarded as an inexact version of the regular polyhedral Gauss–Seidel-method. As proven in Theorem 4.1.1, the polyhedral Gauss–Seidel-method and also its inexact variant are convergent.

In the following we will look at step 3 of the algorihm (4.4.16). First, we need to determine the coarse correction space W, that depends on the iterate $\vec{w}_{\#\mathcal{N}_j}$. We want to exclude all nodes, where the current iterate lies in an edge of the Gibbs–Simplex G (2.2.1). At these nodes, we want to use each edge E_m of G, where the current iterate $\vec{w}_{\#\mathcal{N}_j}$ is inside the relative interior of the edge E_m. It is easy to check that the nonadmissble or active edges of an iterate $\vec{w} \in (\mathbb{R}^{2N})^{\#\mathcal{N}_j}$ at the node p_n are given like in [GSa,T, p.3] by

$$\mathcal{A}_n(\vec{w}_n) = \{E_{m(i,j)} \mid \vec{w}_{n,i} = 0 \text{ or } \vec{w}_{n,j} = 0\}. \tag{4.4.17}$$

The coarse correction space for an iterate $\vec{w}$ inside the convex set $G^{\#\mathcal{N}_j}$ is then given by [GSa,T, p.3]

$$W(\vec{w}) = \prod_{i=1}^{n_k} W_i(\vec{w}_i), \qquad W_i(\vec{w}_i) = \operatorname{span}\{E_m \otimes e_i \in \mathcal{E} \otimes \mathbb{R}^{\#\mathcal{N}_j} \mid E_m \notin \mathcal{A}_i(\vec{w}_i)\}.$$

In the extreme case, where $\vec{w}_i$ lies in a corner of the Gibbs-Simplex at the node p_i, we see that all edges are nonadmissible, so $\mathcal{A}_i(\vec{w}_i) = \mathcal{E}$ and then $W_i(\vec{w}_i) = \operatorname{span}\{\emptyset\} = \{0\}$.
Now we will look at the correction operator $\mathcal{C}$. As we want to use the generalized form of $\mathcal{C}$ (4.4.12), the functional J has to be smooth inside the coarse correction space $W(\vec{w}_{\#\mathcal{N}_j})$. This is fulfilled, so $J''(\vec{w}_{\#\mathcal{N}_j})$ and $J'(\vec{w}_{\#\mathcal{N}_j})$ can be calculated in $W(\vec{w}_{\#\mathcal{N}_j})$. Analogously to the truncation in (4.4.12), we use the notation

$$\langle H_{\vec{w}}\vec{x}, \vec{y}\rangle := \begin{cases} \langle -J''\vec{x}, \vec{y}\rangle & \text{if } \vec{x}, \vec{y} \in W(\vec{w}) \\ 0 & \text{if } \vec{x} \perp W(\vec{w}) \text{ or } \vec{y} \perp W(\vec{w}), \end{cases}$$

$$\langle \vec{d}_{\vec{w}}, \vec{x}\rangle = \begin{cases} \langle J'(\vec{w}), \vec{x}\rangle & \text{if } \vec{x}, \in W(\vec{w}) \\ 0 & \text{else} \end{cases}$$

and obtain the correction operator

$$\mathcal{C}(\vec{w}) = H^+_{\vec{w}}\vec{d}_{\vec{w}}. \tag{4.4.18}$$

Now, like in [GSa,T, p.3], this yields the following linearized problem

$$\vec{c} \in W(\vec{w}): \quad H_{\vec{w}}\vec{c} = \vec{d}_{\vec{w}}. \tag{4.4.19}$$

This linearized problem has a unique solution in $W(\vec{w})$, however, as explained in [GSa,T,

p.3], it is not directly suited for the application of common multigrid methods, as the spaces $W_i(\vec{w}_i)$ change for varying i. Therefore, we expand the system (4.4.18) to the admissible edges in $W(\vec{w})$ and obtain

$$\vec{c}^* \in W^*(\vec{w}): \quad H^*_{\vec{w}}\vec{c}^* = \vec{d}^*_{\vec{w}} \tag{4.4.20}$$

where the expanded coarse space is given by

$$W^*(\vec{w}) = \prod_{i=1}^{\#\mathcal{N}_j} W_i^*(\vec{w}_i), \quad W_i^*(\vec{w}_i) = \{\vec{x} \in \mathbb{R}^M \mid \vec{x}_m = 0 \text{ if } E_m \in \mathcal{A}_i(\vec{w}_i)\} \tag{4.4.21}$$

and $\vec{c}^*, \vec{d}^*_{\vec{w}} \in (\mathbb{R}^M)^{\#\mathcal{N}_j}$, $H^*_{\vec{w}} \in (\mathbb{R}^{M\times M})^{\#\mathcal{N}_j \times \#\mathcal{N}_j}$.
Like in [GSa,T, p.3], we define the surjective mapping

$$\mathcal{M} : \mathbb{R}^{M\times \#\mathcal{N}_j} \to \mathbb{R}^{2N\times \#\mathcal{N}_j}, (\mathcal{M}\vec{v}^*)_{k,i} = \sum_{j=1}^{2N} v_{k,j} E_{m(i,j)}$$

such that $\mathcal{M}\vec{d}^*_{\vec{w}} = \vec{d}_{\vec{w}}$ and $H^*_{\vec{w}} = \mathcal{M}^T H_{\vec{w}} \mathcal{M}$.
The mapping $\mathcal{M}$ is especially surjective from the expanded space $W^*(\vec{w})$ (4.4.21) to $W(\vec{w})$. In [GSa,T, p.3] it is proven, that the expanded system (4.4.20) has a solution $\vec{c}^*$ in the extended space $W^*(\vec{w})$ and that the application of $\mathcal{M}$ yields the unique solution $\vec{c} = \mathcal{M}\vec{c}^* \in W(\vec{w})$ of (4.4.19).
We do not have to compute the matrix $H_{\vec{w}}$ and $d_{\vec{w}}$ to solve (4.4.19), we can directly compute $H^*_{\vec{w}}$ and $d^*_{\vec{w}}$ like in [GSa,T, p.3] and solve the expanded system (4.4.20). Now we can apply a multigrid method for truncated systems to (4.4.20), like it was done in the last chapter, for instance (4.4.14).

Remark 4.4.7. *As explained in [Gr, p.63], when J is nonquadratic, solving the linearized problem* (4.4.19) *is not equivalent to a Newton-step, in contrast to the linearized problem* (4.4.13) *in the box-constrained case. Nevertheless, we will use the matrices defined like above, because we still obtain convergence of the algorithm* (4.4.16).

Now, analogously to the convergence result in the box-constrained case in Theorem 4.4.6, we obtain the following theorem for the simplex-constrained case.

Theorem 4.4.8. *Let $\vec{J}_0$ be given like in* (4.2.4). *Then the sequence φ^ν generated by the algorithm* (4.4.16)

$$\vec{\varphi}^{\nu+\frac{1}{2}} = \vec{\varphi}^{\nu} + \mathcal{F}(\vec{\varphi}^{\nu})$$
$$\vec{\varphi}^{\nu+1} = \vec{\varphi}^{\nu+\frac{1}{2}} + \rho^\nu P^\nu\big(\mathcal{C}(\vec{\varphi}^{\nu+\frac{1}{2}}) + \vec{\epsilon}^{\nu}\big)$$

with the polyhedral Gauss–Seidel method $\mathcal{F}$ (4.1.8), *the correction operator $\mathcal{C}$* (4.4.18), *the orthogonal projection P^ν into the Gibbs–Simplex for each node and the damping parameter ρ^ν chosen such that*

$$\vec{J}\Big(\vec{\varphi}^{\nu+\frac{1}{2}} + \rho^\nu P^\nu\big(\mathcal{C}(\vec{\varphi}^{\nu+\frac{1}{2}}) + \epsilon^\nu\big)\Big) \le \vec{J}\Big(\vec{\varphi}^{\nu+\frac{1}{2}}\Big),$$

converges to the unique solution of

$$\vec{\varphi} \in G^{\#\mathcal{N}_j} : \ \vec{J}_0(\vec{\varphi}) \leq \vec{J}_0(\vec{v}) \ \ \forall \vec{v} \in G^{\#\mathcal{N}_j}$$

for every error $\vec{\epsilon}^{\nu}$.

Proof. This theorem follows directly from Lemma 4.4.5 or Theorem 4.1 in [Gr, p.54] respectively. □

Now we have obtained the convergent TNNMG algorithm (4.4.16) adapted to our simplex-constrained nonlinear problem, that is used for the calculation of the examples in the very last chapter of this thesis.

In the next section, we will refine our grid adaptively, as the structure of our problem with a very large active set Ω_h is compared to the inactive set Ω_I, demands an adaptive calculation.

4.5 Adaptivity

In order to solve our problem adaptively, we have to refine our grid locally instead of globally, thus the concept of a leaf grid is needed.

Definition 4.5.1 (leaf grid)**.** *Let* $\mathcal{T}_0, ..., \mathcal{T}_j$ *be a refinement hierarchy where* $\mathcal{T}_0$ *is a triangulation of* Ω_h *yet* $\mathcal{T}_i$ *contains only local refinements, so for* $i > 0$ *the triangulation* $\mathcal{T}_i$ *does not need to cover* Ω_h *completely. Then the leaf grid is given by*

$$\mathcal{T} = \mathcal{L}(\mathcal{T}_0, ..., \mathcal{T}_j) = \mathcal{T}_j \ \cup \bigcup_{i=0}^{j-1} \left\{ \tau \in \mathcal{T}_i \mid \tau \cap \tau' = \emptyset \ \forall \, \tau' \in \mathcal{T}_{i+1} \right\}$$

[Gr, p.24]

On a leaf grid the finite element space is given by

$$S(\mathcal{T}) := \{ v \in C(\Omega_h) \ \big| \ v_{|\tau} \text{ is affine linear } \forall \tau \in \mathcal{T} \} \subset H^1(\Omega_h). \tag{4.5.1}$$

If the leaf grid is a conforming triangulation, then $S(\mathcal{T})$ is just the regular finite element space S_j (3.2.2).
We want to preserve shape regularity, therefore we only refine according to our refinement rule. Then, generally, the leaf grid is nonconforming, as we can see for instance in Figure 3, and so-called hanging nodes appear.

Definition 4.5.2 (hanging node)**.** *Let* $\mathcal{T}$ *be a triangulation of* Ω*. A node* $p \in \mathcal{N}(\mathcal{T})$ *is called a hanging node if there is an element* $\tau \in \mathcal{T}$ *with* $p \in \overline{\tau}$ *but* p *is not a vertex of* τ*. The set of all hanging nodes is denoted by* $\mathcal{H}(\mathcal{T}) \subset \mathcal{N}(\mathcal{T})$*. [Gr, p.25]*

In the implementation, we only allow one hanging node per edge.
On a nonconforming grid, it is not clear how to choose a nodal basis Λ. One possibility is to use so-called nonconforming basis functions.

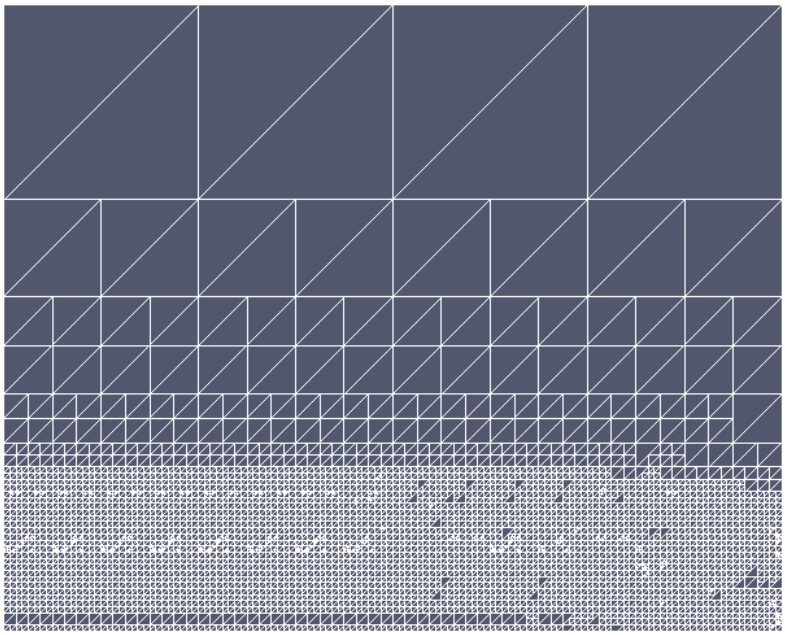


Figure 3: leaf grid

Definition 4.5.3 (nonconforming basis function)**.** *Let $\mathcal{T}$ be a triangulation of Ω. A nonconforming basis function $\overline{\lambda}_i \in L^2(\Omega)$ at the point $p_i \in \mathcal{N}(\mathcal{T})$ is defined as:*
For each $\tau \in \mathcal{T}$ there is a linear representation

$$\overline{\lambda}_{p_{|\tau}} = \mu_{p,\tau} \in C(\overline{\tau})$$

using the regular basis function $\mu_{p,\tau}(q) = \delta_{pq}$ for all points $q \in \tau$. [Gr, p.25]

Using a conforming grid, the nonconforming basis functions are equal to the conforming basis function defined like in (3.2.3) or (4.3.1).
We can represent each nonconforming basis function at a hanging node by a linear combination of nonconforming basis functions at non-hanging nodes, as stated in the following Lemma.

Lemma 4.5.4. *Let $\mathcal{T} = \mathcal{L}(\mathcal{T}_0, ..., \mathcal{T}_j)$ be the leaf grid for a grid hierarchy on Ω_h. Then for all hanging nodes $q \in \mathcal{H}(\mathcal{T})$ there are coefficients a_{qp} where $p \in \mathcal{N}(\mathcal{T})\backslash\mathcal{H}(\mathcal{T})$ is not a hanging node, such that*

$$v(q) = \sum_{p \in \mathcal{N}(\mathcal{T})\backslash\mathcal{H}(\mathcal{T})} a_{qp} v(p), \quad \forall\, v \in S(\mathcal{T}).$$

[Gr, p.27]

Using Lemma 4.5.4, we can gain a representation of the nodal basis on a nonconforming grid where we just use basis functions at non-hanging nodes.

Theorem 4.5.5. *Let $\mathcal{T} = \mathcal{L}(\mathcal{T}_0, ..., \mathcal{T}_j)$ be the leaf grid of a grid hierarchy on Ω_h. Then a basis of $\mathcal{S}(\mathcal{T})$ is given by*

$$\mathcal{B}(\mathcal{T}) := \left\{ \lambda_p = \overline{\lambda}_p + \sum_{q \in \mathcal{H}(\mathcal{T})} a_{qp} \overline{\lambda}_q \;\middle|\; p \in \mathcal{N}(\mathcal{T}) \backslash \mathcal{H}(\mathcal{T}) \right\}$$

[Gr, p.26]

From Theorem 4.5.5, we see that a hanging node is not a degree of freedom, in contrast to a non-hanging node.
As we now have a leaf grid concept, we can refine the grid locally. To do this, we need error estimators that indicate where we have to perform a local refinement.

4.6 Hierarchical error estimators

In this section we will develop a hierarchical error estimator based on the extension of our finite element space $S(\mathcal{T})$ (4.5.1) on the leaf grid by an incremental space V. In the implementation we use this error estimator only for the phase field function and then calculate with the adapted grid for both the phase field function and the solute function.

For the phase field function, we want to solve the minimization problem (3.1.3) or (3.1.5) respectively, so we consider the arccording variational inequalities (3.1.2) or (3.1.6).
We want to derive a posteriori error estimates for the error $||\varphi - \varphi_S||_X$, where $\varphi \in X$ is the correct solution and $\varphi_S \in S(\mathcal{T})$ is the solution in the finite element space $S(\mathcal{T})$. From (3.1.2) we obtain, using the decomposition $\varphi = \varphi_S + e$, the defect problem for the error $e \in X$:

$$e \in (\mathcal{G}_j - \varphi_S) : \langle \nabla J_0(\varphi_S + e), v - e \rangle \geq 0, \quad \forall v \in (\mathcal{G}_j - \varphi_S) \tag{4.6.1}$$

where the error is from the shifted space $(\mathcal{G}_j - \varphi_S)$ and the nonlinear ∇J_0 also is shifted by φ_S. In the isotropic case, the defect problem (4.6.1) transforms to [GKS,H, p.2]

$$e \in (\mathcal{G}_j - \varphi_S) : a(e, v - e) \geq l(v - e) - a(\varphi_S, v - e), \quad \forall v \in (\mathcal{G}_j - \varphi_S).$$

To get an approximation of (4.6.1), we choose a space $V \subset X$ such that $V \cap S(\mathcal{T}) = \{0\}$, and consider the extension space: $Q = V \oplus S(\mathcal{T})$. In the implementation, we resemble a complete refinement of the finest triangulation $\mathcal{T}_j$ by using the corresponding basis functions. Then the extension space V is given by

$$V = \sum_{i=1}^{\#\text{ edges in } \mathcal{T}} V_i \tag{4.6.2}$$

where $V_i = \text{span}\{\lambda_i \,|\, p_i \text{ is the center of an edge of } \mathcal{T}\}$, containing only high-frequent functions. Now we can compute the error on each edge of $\mathcal{T}$. In analogon to $\mathcal{G}_j \subset S_j^{2N}$

(3.2.4), the admissible set for φ_Q in Q is given by

$$\mathcal{G}_Q := \left\{ v \in Q^{2N} \;\middle|\; \sum_{i=1}^{2N} v_i = 1, v_i \geq 0 \quad \forall\ i = 1, ..., 2N \right\} \supset \mathcal{G}_j.$$

For the error $e_Q = \varphi_Q - \varphi_{S(\mathcal{T})}$ in Q, given by the difference of the solution in $S(\mathcal{T})$ and in Q, we obtain the discretized defect problem

$$e_Q \in (\mathcal{G}_Q - \varphi_{S(\mathcal{T})}) : \langle \nabla J_0(\varphi_{S(\mathcal{T})} + e_Q), v - e_Q \rangle \geq 0, \quad \forall v \in (\mathcal{G}_Q - \varphi_{S(\mathcal{T})}).$$

The error e_Q can be decomposed into a part $e_V \in V$ and a part $e_{S(\mathcal{T})} \in S(\mathcal{T})$ such that $e_Q = e_V + e_{S(\mathcal{T})}$. As explained in [GKS,H, p.3], the exact evaluation of e_Q is too costly, so we plainly assume that the low-frequent part of the error $e_{S(\mathcal{T})}$ can be disregarded and it is enough to estimate the error e_V in V (4.6.2) containing high-frequent functions. Then we obtain the localized defect problem

$$e_V \in (\mathcal{G}_V - \varphi_{S(\mathcal{T})}) : \langle \nabla J_0(\varphi_{S(\mathcal{T})} + e_V), v - e_V \rangle \geq 0, \quad \forall v \in (\mathcal{G}_V - \varphi_{S(\mathcal{T})}), \tag{4.6.3}$$

where $\mathcal{G}_V$ is given analogously to $\mathcal{G}_Q$.
As V is the sum of the one-dimensional subspaces V_i, we can decompose the error in V into $e_V = \sum_{i=1}^{\#\text{ edges in }\mathcal{T}} e_i$, with $e_i \in V_i$. Using this decomposition, the localized defect problem (4.6.3) uncouples into several problems

$$e_i \in (\mathcal{G}_{V_i} - \varphi_{S(\mathcal{T})}) : \langle \nabla J_0(\varphi_{S(\mathcal{T})} + e_i), v - e_i \rangle \geq 0, \quad \forall v \in (\mathcal{G}V_i - \varphi_{S(\mathcal{T})})$$

where $\mathcal{G}_{V_i}$ is defined analogously to $\mathcal{G}_Q$.
We solve these localized problems using a fixed number of Gauss-Seidel-steps. Then the local error on each edge in $\mathcal{T}$ is given by

$$\eta_i = ||e_i||. \tag{4.6.4}$$

while the global error is given by

$$\eta = \sqrt{\sum_{i=1}^{\#\text{ edges in }\mathcal{T}} \eta_i^2}.$$

To measure the error in (4.6.4), we use the norm introduced by the bilinear form $a(.,.)$ from the functional J in the isotropic case. Therefore, to get a reasonable error estimation also in the anisotropic case, it is important to choose an anisotropy function, whose unit circle approximately corresponds to the eucledian unit circle. All examples presented in the first chapter, except the regularized l^1-norm from Example 2.1.6 with a large θ, fulfill this condition.
To obtain a finer grid, we mark the edges corresponding to a certain percentage ω of the error η, i.e. for an ordering $\eta_1 \geq \eta_2 \geq .. \geq \eta_{\#\text{ edges in }\mathcal{T}}$ we consecutively mark edges until we reach k such that $\sqrt{\sum_{i=1}^{k} \eta_i^2} \geq \omega\ \eta$. Then all elements bordering the marked edges are refined.

In the next section we will look at some numerical experiments using the anisotropy functions that were presented in the first chapter.

5 Numerical experiments

In this section we will present experiments using different anisotropy functions from the first chapter and then compare the performance of the TNNMG method in the isotropic case with the performance in the anisotropic case. The implementation for the numerical tests was done using the DUNE framework [D]. The following DUNE modules were used:

- digm (Revision 1632): This module contains the implementation of the isotropic and the anisotropic phase field model.
- dune-common (Revision 7395), dune-geometry (Revision 447), dune-grid (Revision 8892), dune-localfunctions (Revision 1199), dune-istl (Revision 1873): These core modules were used for shape functions, matrix and vector classes, grid interfaces, reference elements and quadratures [Gr, p.133] [D] [D, GG I] [D, GG II] [D, PC].
- dune-fufem (Revision 12382): This discretization module was used for handling grid functions and finite element spaces, for assembling matrices and vectors, and for the implementation of the hierarchical error estimator [Gr, p.133].
- dune-solvers (Revision 12389): This module was used for the handling of multigrid transfer operators and as an infrastructure for algebraic solvers [Gr, p.133].
- dune-subgrid (Revision 419): This module was used for the handling of adaptive grids when applying Rothe's method [Gr, p.133] [GSa,D].
- dune-tnnmg (Revision 12397): The TNNMG method is implemented in this module.
- phase-field (Revision 1171): Some anisotropy functions used in the following examples were implemented in this module.

5.1 Phenomenology

In all examples, the domain in 2D is given by a square $\Omega = [-1,-1] \times [-1,1]$ and in 3D by a cube $\Omega = [-1,-1] \times [-1,1] \times [-1,1]$. In the examples in 2D, the phases are colored in green, orange and red while the alloyed parts are colored in dark green and in dark orange for 2 phases and in light green, light orange and light red for 3 phases.
In each example, we give the parameters used the initial function and some representative timesteps.

Example 5.1.1 (3D example with the eucledian norm (Example 2.1.3), Figure 4)**.**

- $d = 3$, $N = 2$, $\gamma_\theta(x) = |x|$, $\theta = 0.01$
- $g(u) = 30\,u$, $\alpha = 100$, $\beta = 1$
- $\epsilon = 0.02$, $\tau = 0.05$

In the initial funciton, the alloyed part of phase 2 is in a circle with radius 0.5 inside the interface with thickness 0.02, surronded by the alloyed part of phase 1.
In the 3d plots, the parts of the cube where $0.5 < \varphi_1 + \varphi_3 < 1$ are shown. The interfacial region is painted in beige and the phase $\varphi_1 + \varphi_3$ is painted in red.

Example 5.1.2 (2D example with 3 phases and the eucledian norm (Example 2.1.3), Figure 5)**.**

- $d = 2$, $N = 2$, $\gamma_\theta(x) = |x|$, $\theta = 0.01$
- $g(u) = 20\,u$, $\alpha = 100$, $\beta = 1$
- $\epsilon = 0.01$, $\tau = 0.05$

In the plot of the initial function, each alloyed twin to phase i is denoted by i' and the numbers inside of the circle denote the slope of the line that separates two phases at that point. We see that the interface looks like the unit circle of the eucledian norm.

Example 5.1.3 (regularized l^1-norm (Example 2.1.6), Figure 6)**.**

- $d = 2$, $N = 2$, $\gamma_\theta(x) = \sum_{k=1}^{2} \sqrt{x_k^2 + \theta|x|^2}$, $\theta = 0.01$
- $\epsilon = 0.02$, $\tau = 0.05$
- $g(u) = 30\,u$, $\alpha = 100$, $\beta = 1$

In this example we can clearly recognize the structure of the Wulff shape of the regularized l^1-norm with $\theta = 0.01$. It is especially interesting to look at timestep 18, where the interface looks like stairs, yet in timestep 48, the interface is completely strainght again. These "stairs" will also occur in further anisotropic examples.

Example 5.1.4 (l^r-norm (Example 2.1.4), Figure 7)**.**

- $d = 2$, $N = 2$, $\gamma_r(x) = \left(\sum_{k=1}^{d} |x_k|^r\right)^{\frac{1}{r}}$, $r = 10$
- $\epsilon = 0.02$, $\tau = 0.05$
- $g(u) = 30\,u$, $\alpha = 100$, $\beta = 1$

We can clearly recognize the structure of the Wulff shape.

Example 5.1.5 (twisted l^r-anisotropy (Example 2.1.4), Figure 8)**.**

- $d = 2$, $N = 2$, $\gamma_{r,M_i}(x) = \left(\sum_{k=1}^{d} |M_i x_k|^r\right)^{\frac{1}{r}}$, $r = 10$, $M_1 = Id$, $M_2 = \begin{pmatrix} cos(45°) & -sin(45°) \\ sin(45°) & cos(45°) \end{pmatrix}$
- $\epsilon = 0.02$, $\tau = 0.05$
- $g(u) = 30\,u$, $\alpha = 100$, $\beta = 1$

In this example we can see a mixture of the Wulff shape from the twisted and from the untwisted norm. When looking at the interface, the parts belonging to the twisted anisotropy and the parts belonging to the untwisted norm alternate. Again, in timestep 29, the interface resembles stairs.

Example 5.1.6 (anisotropy function by Kobayashi (Example 2.1.7), Figure 9)**.**

- $d = 2$, $N = 2$, $\gamma_{a,k}(x) \begin{cases} (1 + a\, cos(k\beta(x)))\, |x| & \text{for } x \neq 0 \\ 0 & \text{for } x = 0 \end{cases}$
- $\epsilon = 0.02$, $\tau = 0.05$
- $g(u) = 30\, u$, $\alpha = 100$, $\beta = 1$

In this example, we cannot distinguish the Wulff shape as easy as in the other cases. The reason is that the ansotropy by Kobayashi resembles the eucledian unit circle the most in comparison to the other anisotropy functions considered here.

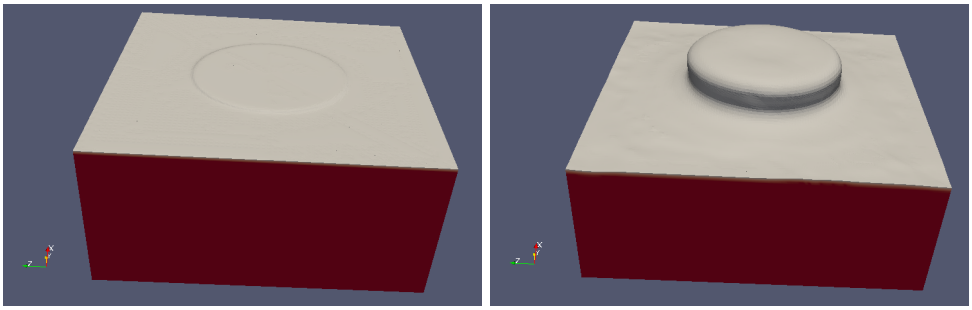


Figure 4: 3D example: timestep 0, timestep 8

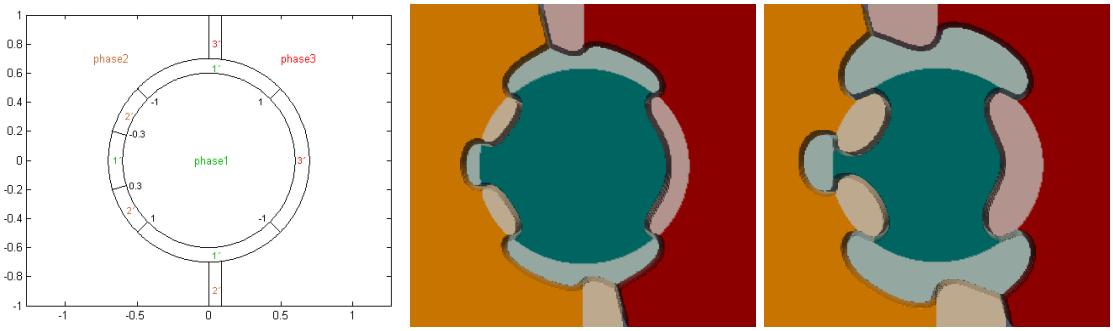


Figure 5: 3 phases: initial function, timesteps 40 and 69

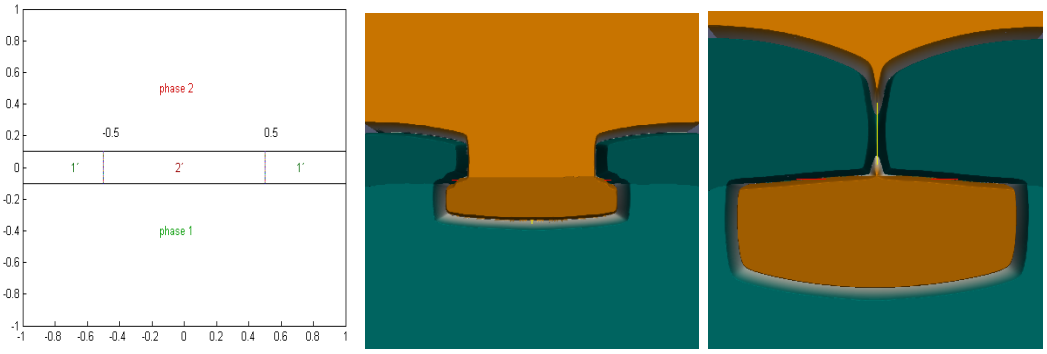


Figure 6: regularized l^1-norm: initial function, timesteps 18 and 49

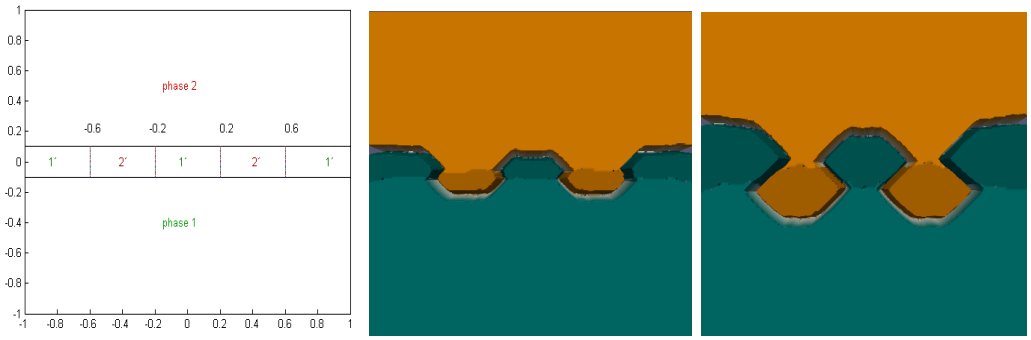


Figure 7: l^r-norm: initial function, timesteps 9 and 19

Figure 8: twisted l^r-anisotropy: initial function, timesteps 18 and 29

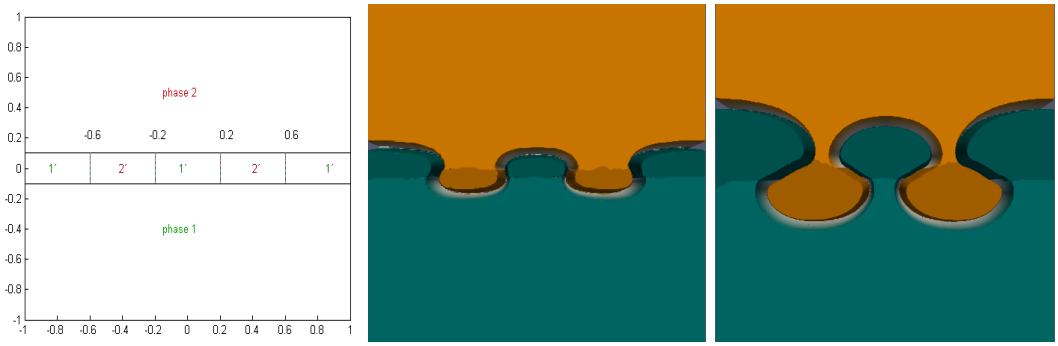


Figure 9: anisotropy by Kobayashi: initial function, timesteps 9 and 23

5.2 Errors and convergence

In this section we will take a look at the convergence of the TNNMG algorithm and get a comparison between the isotropic case and the different anisotropic cases. In all plots, we used an example with the following parameters and the initial function that can be seen in Figure 6:

- $d = 2$, $N = 2$, $\epsilon = 0.02$, $\tau = 0.05$
- $g(u) = 30\,u$, $\alpha = 100$, $\beta = 1$

In Figure 10, we can see the error over the number of unknowns during the adaptive refinement process of timestep 1. Timestep 1 was chosen, because there the function from the anterior timestep is not an initial input but was calculated in the timestep before. In the two plots at the very top using the eucledian norm, once the code for the isotropic case and then the anisotropic case, where the stiffness matrix is treated like a nonlinearity, were used. The plots look almost the same, this tells us that the TNNMG algorithm is very efficient especially for anisotropic problems.
Looking at the anisotropy functions, it sticks out that for the regularized $l1$-norm much less nodes are needed to achieve an error 0.05, compared to the other anisotropy functions. Looking at the l^r-norm, a lot nodes are needed to get just a small reduction of the error.
In all cases though, we can see a logarithmic decay of the error.
In Figure 11, we can see the convergence rate plotted over the unknowns for the eucledian norm. The more unknowns we have, the worse the convergence rate gets. However, when solving with the code for the anisotropic case, the convergence is near one already for much less nodes compared to the isotropic case.
The jump in the beginning when solving with the anisotropic code can be explained by the little number of nodes the refinement process is started with.
Finally, we see that the TNNMG method is a very powerful solver for nonlinear minimization problems, especially anisotropic problems.

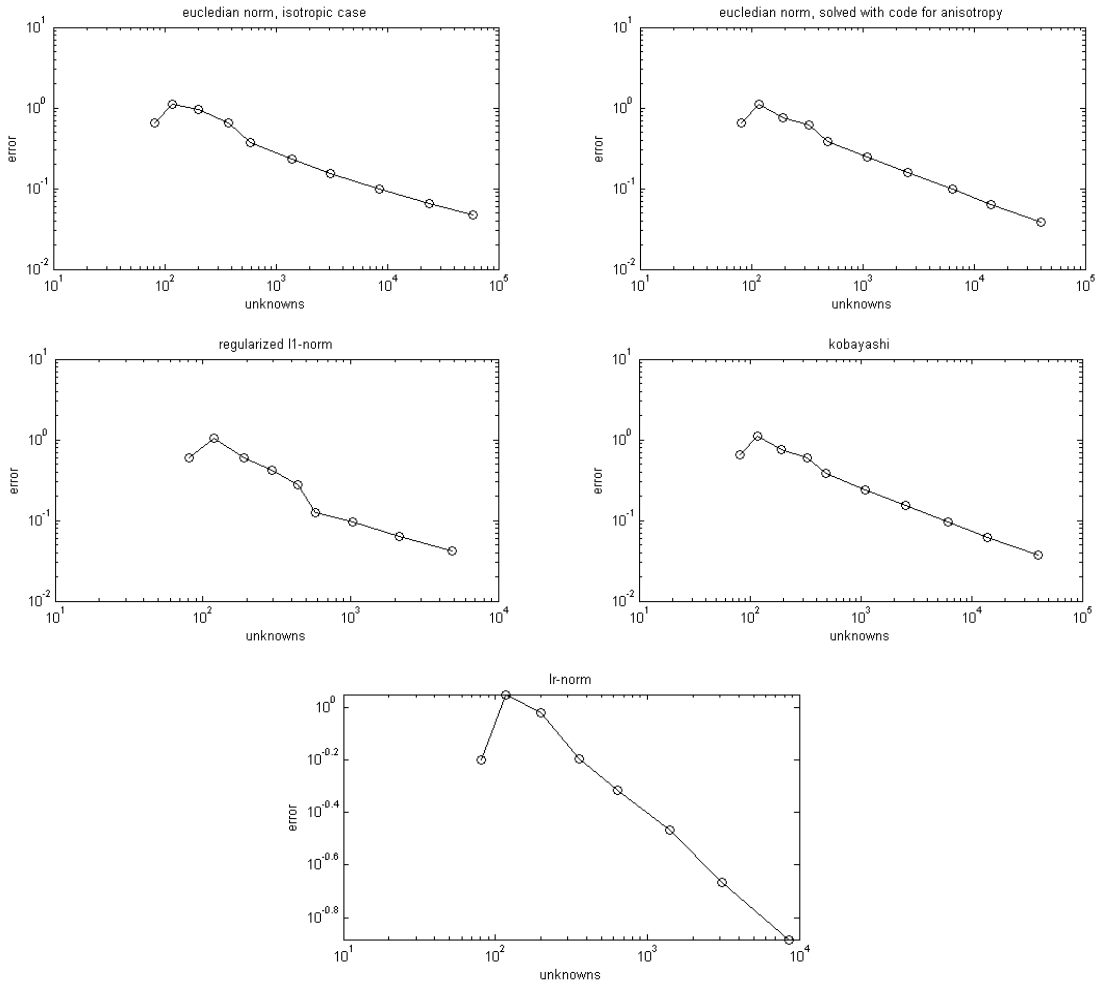


Figure 10: error over unknowns in timestep 1

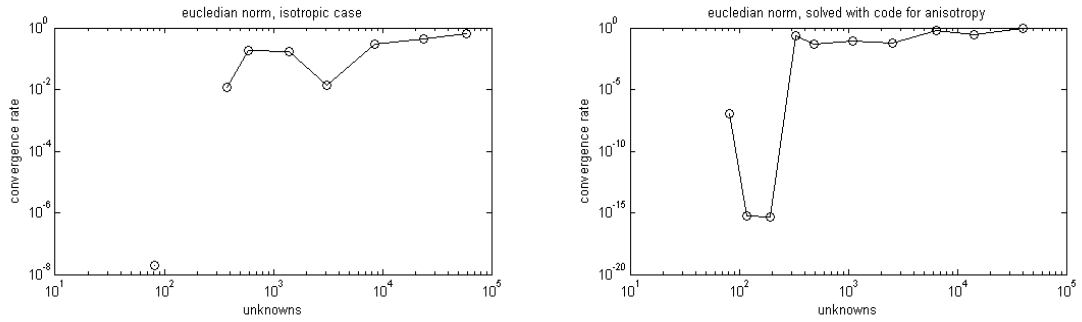


Figure 11: convergence rate over unknowns in timestep 1

References

[BP] G. Bellentini, M. Paolini: Anisotropic motion by mean curvature in the context of Finsler geometry. Hokkaido Mathematical Journal, 25:537-566, 1996.

[BR] Eric Burman, Jacques Rapaz: Existence of solutions to an anisotropic phase-field model. Mathematical Methods in the Applied Sciences, 26(13):1137-1160, 2003.

[DDE] Klaus Deckelnick, Gernard Dziuk, Charles M. Elliot: Computation of geometric partial differential equations and mean curvature flow. Acta Numerica, 14:139–232, 2005.

[D] DUNE: DUNE Automatic Documentation, http://www.dune-project.org/ (29th of November 2013)

[DSO] Dune-Solvers, http://numerik.mi.fu-berlin.de/dune/dune-solvers (29th of November 2013)

[DSU] Dune-Subgrid, http://numerik.mi.fu-berlin.de/dune-subgrid/ (29th of November 2013)

[D, GG I] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, O. Sander. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part I: Abstract Framework. Computing, 82(2-3), 2008, pp. 103-119.

[D, GG II] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, M. Ohlberger, O. Sander. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part II: Implementation and Tests in DUNE. Computing, 82(2-3), 2008, pp. 121-138.

[D, PC] M. Blatt, P. Bastian. The Iterative Solver Template Library. In B. Kåström, E. Elmroth, J. Dongarra and J. Wasniewski, Applied Parallel Computing. State of the Art in Scientific Computing. Volume 4699 of Lecture Notes in Scientific Computing, pages 666-675. Springer, 2007.

[ET] Ivar Ekeland, Roger Témam: Convex Analysis and Variational Problems. Society for Industrial and Applied Mathematics, 1999.

[Gr] Carsten Gräser: Convex Minimization and Phase Field Models. PhD Thesis, Freie Universität Berlin, 2011.

[GK] Carsten Gräser, Ralf Kornhuber: Multigrid nethods for obstacle problems. Journal of Computational Mathematics, 27(1):1-44, 2009.

[GKS,T] Carsten Gräser, Ralf Kornhuber, Uli Sack: Time discretizations of anisotropic Allen-Cahn equations. IMA Journal of Numerical Analysis, 33(4):1226-1244, 2013.

[GKS,H] Carsten Gräser, Ralf Kornhuber, Uli Sack: On hierarchical error estimators for time-discretized phase field models. In G. Kreiss, P.Lotstedt, A. Malqvist and M. Neytcheva, editors, Proceedings of ENUMATH 2009, pages 397-406, Springer, 2010.

[GSa,D] Carsten Gräser, Oliver Sander: The dune-subgrid module and some applications. Computing 86:269-290, 2009.

[GSa,P] Carsten Gräser, Oliver Sander: Polyhedral Gauß-Seidel converges. Journal of Numerical Mathematics, 2013. To appear.

[GSa,T] Carsten Gräser, Oliver Sander: Truncated Monsmooth Newton Multigrid Methods for simplex constrained minimization problems. In preparation.

[GS] Harald Garcke, Vanessa Styles: Bi-directional diffusion induced grain boundary motion with triple junctions. Interfaces and Free Boundaries, 6(3):271-294, 2004.

[KKSG] Ralf Kornhuber, Rolf Krause, Vanessa Styles, Carsten Gräser: Multigrid computations of diffusion induced grain boundary motion. In preparation.

[Kh] Ralf Kornhuber: Mehrgittermethoden für lineare Variationsprobleme, Vorlesungsmaterialien Sommersemester 2002. Freie Universität Berlin, 2002.

Declaration of originality

I hereby confirm that I have written the accompanying thesis by myself, without contributions from any sources other than those cited in the text and acknowledgements. This applies also to all graphics, drawings, maps and images included in the thesis.

Place, Date

Lisa Julia Nebel