



School of Engineering

IAMP Institute of Applied
Mathematics and Physics

HANA Numerik SEP Projekt

Mathematische Bildverarbeitung

Autoren

Cyril Gabriele
Philip Kehl

Betreuung

Dr. Simon Iwan Stingelin

Date

05.01.2024

Inhaltsverzeichnis

1 Aufgabe 1	1
1.1 Berechnung der ersten Variation	1
1.2 Berechnung der zweiten Variation	2
1.3 Vergleich der ersten Variation mit dem impliziten Eulerverfahren	3
2 Aufgabe 2	4
2.1 Lösung für $\epsilon = 10^{-6}$ nach $T = 2s$	5
2.2 Lösung mit der von Hand berechneten Variationen	6
2.3 Einfluss der Diffusionskonstante ϵ	6
2.4 Histogram der Grauwertstufen	7
3 Aufgabe 3	8
3.1 Zeitabhängige Diffusionskonstante ϵ	8
3.2 Beschleunigen des Verfahrens mit variablem Δt	8
4 Weitere Ansätze zur Verbesserung	9
4.1 Thresholding statt mathematische Bildverarbeitung	9
4.2 Weniger Iterationen und Thresholding	10

1 Aufgabe 1

In der Aufgabenstellung (Anhang: Aufgabestellung) wurde folgender Auftrag für Aufgabe 1 definiert:

Berechnen Sie die erste und zweite Variation des Funktional E^{IE} (Anhang: Aufgabestellung (3)). Zeigen Sie, dass die erste Variation mit dem impliziten Euler-Verfahren übereinstimmt. Um die erste und zweite Variation zu berechnen, wurde die Theorie aus dem Skript zum Thema Variationsrechnung verwendet. Nachfolgend sind die von Hand ausgeführten Berechnungen zu sehen (siehe Abbildung 1 und Abbildung 2). Des Weiteren wurde gezeigt, dass die ersten Variation mit dem impliziten Eulerverfahren übereinstimmt, dazu wurde Vorwissen aus der Numerikvorlesung verwendet.

1.1 Berechnung der ersten Variation

Im Folgenden sind die analytischen Berechnungen der ersten Variation dargestellt (siehe Abbildung 1).

$$E^{IE}(u) = \int_{\Omega} \frac{\varepsilon_1}{2} |\nabla u|^2 + \underbrace{u^2(1-u^2)^2}_{\omega(u)} + \frac{1}{2\Delta t} |u - u^{(n)}|^2 dx, \quad (3)$$

Gef.:

- Funktional $E^{IE}(u)$
- Raum $H^1(\Omega) \rightarrow \mathbb{R}$

Ges.:

- 1. Variation

$$\begin{aligned} E^{IE}(u) &= \int_{\Omega} \frac{\varepsilon_1}{2} |\nabla u|^2 + W(u) + \frac{1}{2\Delta t} |u - u^{(n)}|^2 dx \\ \Rightarrow \frac{d}{d\varepsilon_i} E^{IE}(u + \varepsilon_i v) &\Big|_{\varepsilon_i=0} \\ &= \frac{d}{d\varepsilon_i} \int_{\Omega} \frac{\varepsilon_1}{2} |\nabla u + \varepsilon_i \nabla v|^2 + W(u + \varepsilon_i v) + \frac{1}{2\Delta t} |(u + \varepsilon_i v) - u^{(n)}|^2 dx \Big|_{\varepsilon_i=0} \\ &= \frac{d}{d\varepsilon_i} \int_{\Omega} \underbrace{\frac{\varepsilon_1}{2} |\nabla u + \varepsilon_i \nabla v|^2}_{\textcircled{1}} + \underbrace{W(u + \varepsilon_i v)}_{\textcircled{2}} + \underbrace{\frac{1}{2\Delta t} |(u + \varepsilon_i v) - u^{(n)}|^2}_{\textcircled{3}} dx \Big|_{\varepsilon_i=0} \end{aligned}$$

Betragsfunktion kann beim Ableiten, wegen Potenz mit geradem Exponenten "ignoriert" werden?

$$\begin{aligned}
\textcircled{1} &= \frac{d}{d\varepsilon_1} \int_{\Omega} \frac{\varepsilon_1}{2} \cdot (\nabla u + \varepsilon_1 \nabla v)^2 dx \Big|_{\varepsilon_1=0} \\
&= \int_{\Omega} \frac{\varepsilon_1}{2} \cdot 2(\nabla u + \varepsilon_1 \nabla v) \cdot \nabla v dx \Big|_{\varepsilon_1=0} \\
&= \int_{\Omega} \varepsilon_1 (\nabla u + \varepsilon_1 \nabla v) \cdot \nabla v dx \Big|_{\varepsilon_1=0} \\
\textcircled{2} &= \frac{d}{d\varepsilon_1} \int_{\Omega} \underbrace{(u + \varepsilon_1 v)^2}_{f(\varepsilon)} \underbrace{(1 - (u + \varepsilon_1 v)^2)^2}_{g(\varepsilon)} dx \Big|_{\varepsilon_1=0} \\
&= \int_{\Omega} f'(\varepsilon_1) \cdot g(\varepsilon_1) + f(\varepsilon_1) \cdot g'(\varepsilon_1) dx \Big|_{\varepsilon_1=0} \\
&= \int_{\Omega} 2(u + \varepsilon_1 v) \cdot v (1 - (u + \varepsilon_1 v)^2)^2 - 4(u + \varepsilon_1 v)^2(u + \varepsilon_1 v) (1 - (u + \varepsilon_1 v)^2) \cdot v dx \Big|_{\varepsilon_1=0} \\
\textcircled{3} &= \frac{d}{d\varepsilon_1} \int_{\Omega} \frac{1}{2\Delta t} ((u + \varepsilon_1 v) - u^{(u)})^2 dx \Big|_{\varepsilon_1=0} \\
&= \frac{1}{2\Delta t} \int_{\Omega} 2((u + \varepsilon_1 v) - u^{(u)}) \cdot v dx \Big|_{\varepsilon_1=0} \\
&= \frac{1}{\Delta t} \int_{\Omega} ((u + \varepsilon_1 v) - u^{(u)}) \cdot v dx \Big|_{\varepsilon_1=0}
\end{aligned}$$

1. Variation

$$\begin{aligned}
&\int_{\Omega} \textcircled{1} + \textcircled{2} + \textcircled{3} dx \Big|_{\varepsilon_1=0} \\
&= \int_{\Omega} \varepsilon_1 (\nabla u + \varepsilon_1 \nabla v) \cdot \nabla v + 2(u + \varepsilon_1 v) \cdot v (1 - (u + \varepsilon_1 v)^2)^2 - 4(u + \varepsilon_1 v)^2(u + \varepsilon_1 v) (1 - (u + \varepsilon_1 v)^2) \cdot v + \frac{1}{\Delta t} ((u + \varepsilon_1 v) - u^{(u)}) \cdot v dx \Big|_{\varepsilon_1=0} \\
&\Rightarrow \text{unks der Bedingung: } \varepsilon_1 = 0 \\
&\int_{\Omega} \varepsilon_1 \nabla u \cdot \nabla v + 2uv(1-u^2)^2 - 4u^3(1-u^2)v + \frac{1}{\Delta t} (u - u^{(u)}) v dx \quad (1. \text{ Variation})
\end{aligned}$$

Abbildung 1: Lösungsweg der ersten Variation des Funktionalen E^{IE} .

Die resultierende erste Variation ist die notwendige Bedingung für das Minimierungsproblem. Um die hinreichende Bedingung zu erhalten, musste die zweite Variation berechnet werden (siehe Abbildung ??).

1.2 Berechnung der zweiten Variation

Analog zur ersten Variation wurde die zweite Variation berechnet. Dabei ist die zweite Variation im Grunde die Ableitung der ersten Variation nach ε_2 . Zusätzlich wurde F' (erste Variation) mit einer Funktion $\varepsilon_2 \cdot w(x)$ gestört. Dieses $w(x)$ ist nicht zu verwechseln mit dem Funktional der Separationsenergie $W(u)$ (siehe Anhang: Aufgabenstellung). Zudem ist zu beachten, dass das ε_2 der Störfunktion ein anderes als jenes ε_1 des Funktionalen ist.

1. Variation gegeben als:

$$F^1 = \int_{\Omega} \varepsilon_1 \nabla u \cdot \nabla v + 2uv(1-u^2)^2 - 4u^3(1-u^2)v + \frac{1}{\Delta t} (u - u^{(n)}) \cdot v \, dx$$

Gesucht: 2. Variation: $\frac{d}{d\varepsilon_2} F^1(u + \varepsilon_2 w) \Big|_{\varepsilon=0}$

$$\frac{d}{d\varepsilon_2} \left. \int_{\Omega} \underbrace{\varepsilon_1 (\nabla u \cdot \varepsilon_2 \nabla w) \cdot \nabla v}_{\textcircled{1}} + \underbrace{2(u + \varepsilon_2 w)v(1 - (u + \varepsilon_2 w)^2)^2}_{\textcircled{2}} - \underbrace{4(u + \varepsilon_2 w)^3(1 - (u + \varepsilon_2 w)^2)v}_{\textcircled{3}} + \underbrace{\frac{1}{\Delta t} ((u + \varepsilon_2 w) - u^{(n)}) \cdot v}_{\textcircled{4}} \, dx \right|_{\varepsilon=0}$$

$$\textcircled{1} \quad \frac{d}{d\varepsilon_2} \int_{\Omega} \varepsilon_1 (\nabla u \cdot \varepsilon_2 \nabla w) \cdot \nabla v \, dx$$

$$= \int_{\Omega} \varepsilon_1 \nabla w \cdot \nabla v \, dx$$

$$\textcircled{2} \quad 2 \frac{d}{d\varepsilon_2} \int_{\Omega} \underbrace{(u + \varepsilon_2 w)}_{f(x)} \underbrace{(1 - (u + \varepsilon_2 w)^2)^2}_{g(x)} \, dx$$

$$= \int_{\Omega} 2vw(1 - 5(u + \varepsilon_2 w)^2)(1 - (u + \varepsilon_2 w)^2) \, dx$$

$$\textcircled{3} \quad \frac{d}{d\varepsilon_2} \int_{\Omega} -4(u + \varepsilon_2 w)^3(1 - (u + \varepsilon_2 w)^2) \cdot v \, dx$$

$$= \int_{\Omega} 4vw(u + \varepsilon_2 w)^2(5u^2 + 10uw\varepsilon_2 + 5w^2\varepsilon_2^2 - 3) \, dx$$

$$\textcircled{4} \quad \frac{d}{d\varepsilon_2} \int_{\Omega} \frac{1}{\Delta t} ((u + \varepsilon_2 w) - u^{(n)}) \cdot v \, dx$$

2. Variation

$$\int_{\Omega} \textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{4} \, dx \Big|_{\varepsilon_2=0}$$

$$= \int_{\Omega} \varepsilon_1 \nabla w \cdot \nabla v + 2vw(1 - 5(u + \varepsilon_2 w)^2)(1 - (u + \varepsilon_2 w)^2) + 4vw(u + \varepsilon_2 w)^2(5u^2 + 10uw\varepsilon_2 + 5w^2\varepsilon_2^2 - 3) + \frac{1}{\Delta t} w \cdot v \, dx \Big|_{\varepsilon_2=0}$$

\Rightarrow Vektors der Bedingung: $\varepsilon_2=0$

$$\int_{\Omega} \varepsilon_1 \nabla w \cdot \nabla v + 2vw(1 - su^2)(1 - u^2) + 4vwu^2(su^2 - 3) + \frac{1}{\Delta t} w \cdot v \, dx \quad (\text{2. Variation})$$

Abbildung 2: Lösungsweg der zweiten Variation des Funktionalen E^{IE} .

Somit wurde auch die hinreichende Bedingung analytisch berechnet (siehe Abbildung 2).

1.3 Vergleich der ersten Variation mit dem impliziten Eulerverfahren

Um zu zeigen, dass die beiden Ausdrücke $Mu_{n+1} = Mu_n + \Delta t \delta E(u_n)$ und $\delta E^{IE}(u_{n+1}) = 0$ äquivalent sind, wird im folgenden die Herleitung des impliziten Eulerverfahrens dargestellt

und anschliessend mit der ersten Variation des diskretisierten Funktionals aus Unterabschnitt 1.1 verglichen.

Definition des impliziten Eulerverfahrens:

$$M(u_{n+1}, v) = M(u_n, v) + \Delta t \cdot \delta E(u_n) \quad \text{für alle } v$$

Die Definition der Bilinearform lautet:

$$M(u, v) = \int_{\Omega} u v \, dx$$

Implizit Euler umstellen und Bilinearform $M(u, v)$ anwenden, erste Variation von $E(u)$ einsetzen:

$$0 = M(u_n, v) - M(u_{n+1}, v) + \Delta t \cdot \delta E(u_n)$$

$$0 = \int_{\Omega} (u_n - u_{n+1}) v \, dx + \Delta t \cdot \int_{\Omega} \varepsilon \cdot \nabla u \cdot \nabla v + W'(u) v \, dx$$

Umschreiben und Integrale zusammennehmen:

$$0 = \int_{\Omega} \varepsilon \cdot \nabla u \cdot \nabla v + W'(u) v + \frac{1}{\Delta t} (u_n - u_{n+1}) v \, dx$$

Nach dem Einsetzen von $W'(u)$ wird ersichtlich, dass dieser Ausdruck ^d_{entsc}^h ist zu $\delta E^I(u)$:

$$\int_{\Omega} \varepsilon \cdot \nabla u \cdot \nabla v + 2uv(1-u^2)^2 - 4u^3(1-u^2)v + \frac{1}{\Delta t} (u_n - u_{n+1}) v \, dx = 0$$

$$\delta E^I(u) = \int_{\Omega} \varepsilon \nabla u \nabla v + 2uv(1-u^2)^2 - 4u^3(1-u^2)v + \frac{1}{\Delta t} (u - u^{(n)}) \cdot v \, dx = 0$$

Abbildung 3: Vergleich der ersten Variation mit dem impliziten Eulerverfahren.

2 Aufgabe 2

In der Aufgabenstellung (Anhang: Aufgabestellung) wurde folgender Auftrag für Aufgabe 2 definiert:

Basierend auf den Bilddaten img.pkl (vgl. Abb. 2 links) soll die Phasenseparation mit Hilfe des Ginzburg-Landau-Funktional angewandt werden. Implementieren Sie daher eine zeitabhängige Berechnung der Lösung, in der Sie in jedem Zeitschritt das nichtlineare Problem (2) mit der Newton-Iteration lösen.

1. Wie sieht die Lösung für $\epsilon = 10^{-6}$ nach $T = 2s$ aus?
2. Studieren Sie den Einfluss der Diffusionskonstante. Illustrieren Sie die Abhängigkeit.

3. Erstellen Sie ein Histogram der Grauwerte für das initiale und nach $T = 2s$ berechnete Bild. Ist eine Phasenseparation ersichtlich?
 Benutzen Sie eine FEM Diskretisierung 2. Ordnung (order = 2).

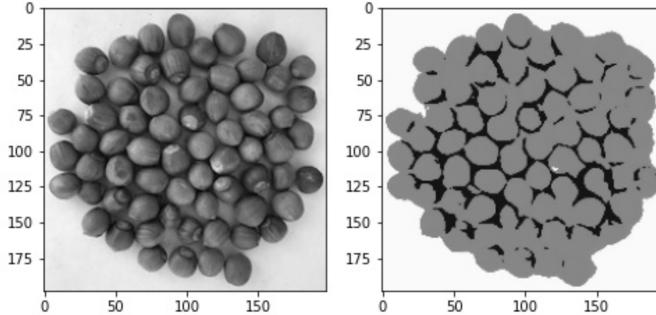


Abbildung 4: Links Ausgangs Daten, rechts Phasenseparation nach $T = 2s$.

Der für dieses Projekt erstellte Python respektive NGSolve Code ist im Anhang zu finden. Des Weiteren wurden die Codesegmente in den Jupyter Notebooks kommentiert und Interpretationen zu den Resultaten verfasst.

2.1 Lösung für $\epsilon = 10^{-6}$ nach $T = 2s$

Nach zwei Sekunden ist in Abbildung 6 eine eindeutige Phasentrennung gegenüber dem Ausgangsbild (Abbildung 5) ersichtlich.

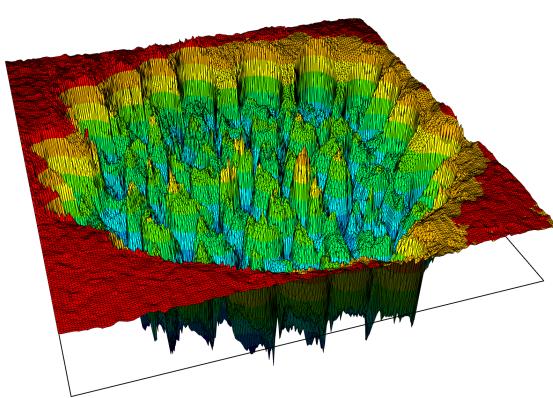


Abbildung 5: Deformiertes Mesh vor der Phasentrennung

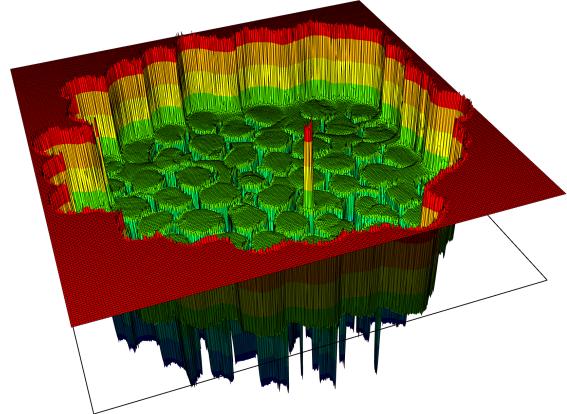


Abbildung 6: Deformiertes Mesh mit eindeutiger Phasentrennung nach 2 Sekunden

Für diese erste Berechnung wurden nicht die von Hand gerechneten Variationen aus Aufgabe 1 verwendet. Stattdessen wurde die erste und zweite Variation berechnet mit Hilfe des NGSolves Code aus der Aufgabestellung berechnet. Der Python Code ist im Anhang ersichtlich.
 Um die Lösung wieder als Graustufenbild auszugeben, muss die Gridfunction (gfu) für alle Pixel im ursprünglichen Bild ausgewertet werden. Abbildung 7 zeigt das umgewandelte Bild, die in der dritten Aufgabe erwähnte helle Stelle ist gut zu erkennen.

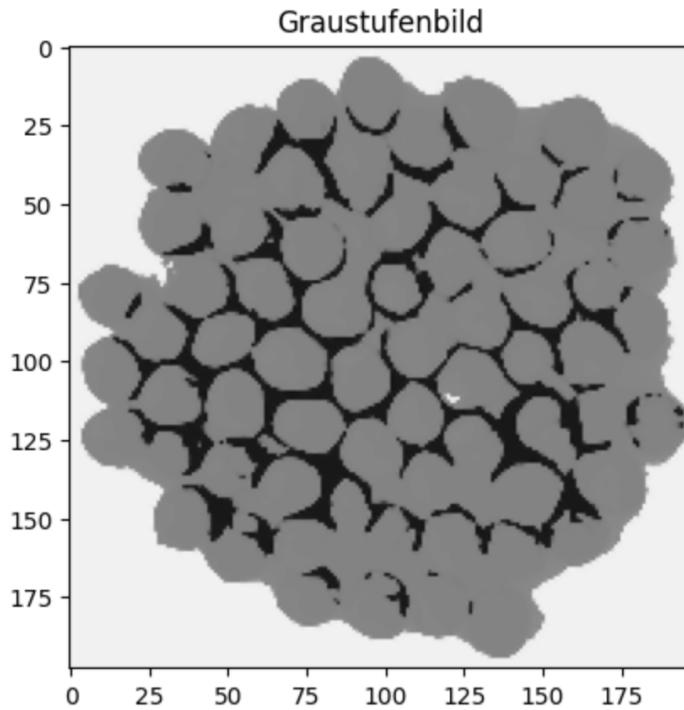


Abbildung 7: Phasengetrenntes Bild, in Graustufen umgewandelt.

2.2 Lösung mit der von Hand berechneten Variationen

In Abschnitt 1 wurden die erste und zweite Variation des Energiefunktional von Hand berechnet. Um die Handrechnung und die Lösung mit NGSolve zu vergleichen, wurde die Phasentrennung auch für die von Hand berechneten Variationen durchgeführt. Die Ergebnisse unterscheiden sich nicht. Der Sourcecode dazu ist im Anhang zu finden.

2.3 Einfluss der Diffusionskonstante ϵ

Um den Einfluss der Diffusionskonstante sauber darzustellen, wurde das Funktional auf ein 1D Problem angewendet. Für die lineare Funktion $f(x) = x$ (siehe Abbildung 8) wurde für verschiedene Werte von ϵ eine Phasentrennung durchgeführt.

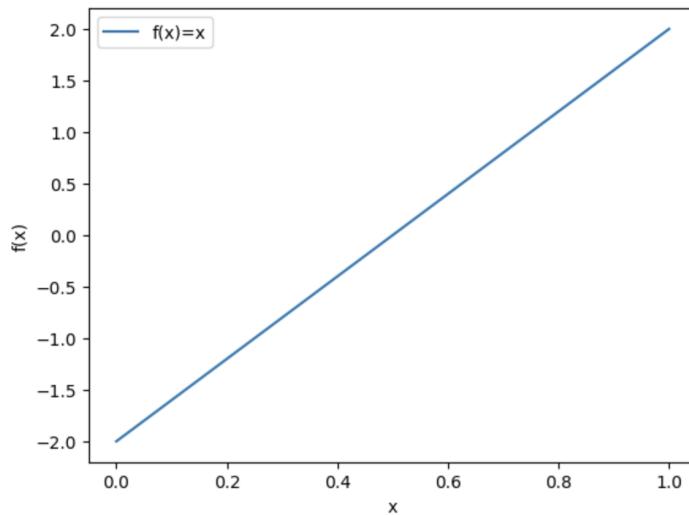


Abbildung 8: 1D Problem $f(x) = x$.

In Abbildung 9 ist gut zu erkennen, dass für grössere ϵ steilere Gradienten stärker gemieden werden und die Werteverteilung nach der Phasentrennung weniger schnelle Änderungen aufweist.

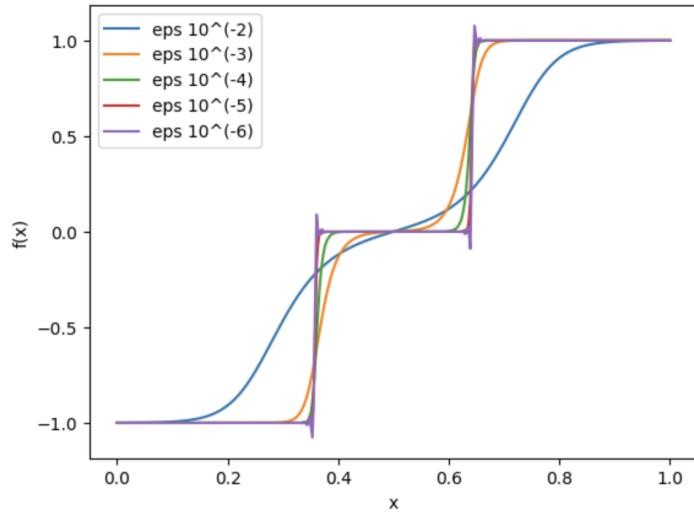


Abbildung 9: Phasentrennung des 1D Problems für verschiedene Werte von ϵ

2.4 Histogramm der Grauwertstufen

Ein weiterer interessanter geforderter Vergleich war das Histogramm der Graustufen vor beziehungsweise nach dem Lösen. Dies ist in Abbildung 10 illustriert.

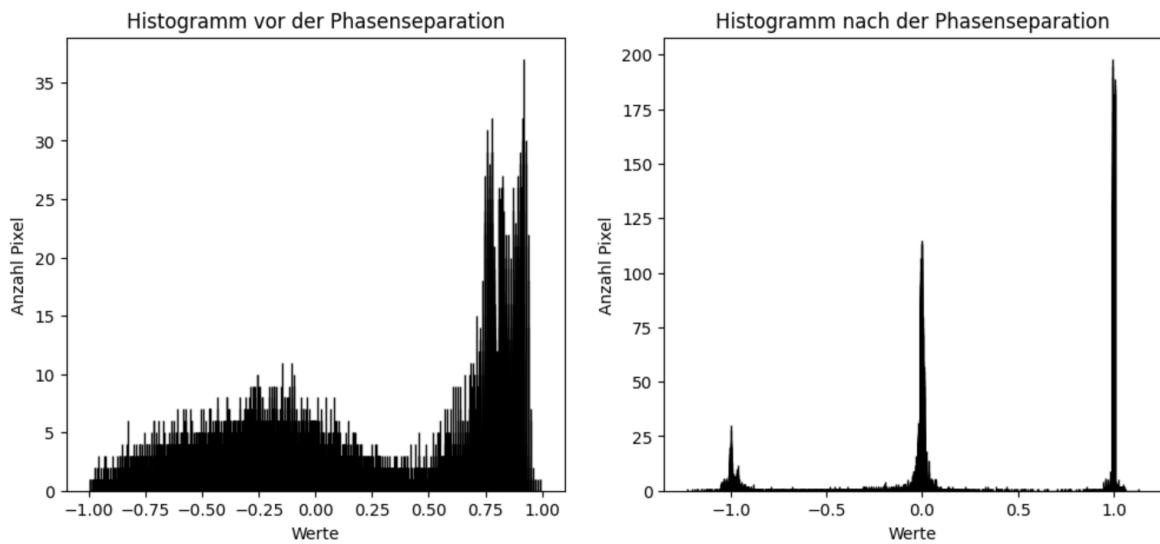


Abbildung 10: Histogramm der Graustufen vor und nach der Phasenseparation.

Dabei kann man klar beobachten, dass bereits eine klare Phasenseparation ersichtlich ist. Die Pixel werden sinngemäss auf -1, 0, oder 1 gezogen. Im Ausgangsbild sind die Graustufenwerte noch nicht separiert was sich auch im Histogramm bemerkbar macht.

3 Aufgabe 3

3.1 Zeitabhängige Diffusionskonstante ϵ

Um die in Abbildung 7 ersichtliche störende weisse Stelle zu eliminieren, kann mit einer zeitabhängigen Diffusionskonstante ϵ gerechnet werden. Der Gedanke hierbei ist, dass starke Gradienten zu Beginn des Verfahrens mit einem eher hohen ϵ stärker zu vermeiden, damit sich am Anfang nicht schon zu starke Trennungen abzeichnen. Da es reicht, das ϵ nur in den ersten Zeitschritten etwas höher zu wählen, wurde folgende exponentiell abfallende Funktion verwendet:

$$\epsilon(t) = 5 \times 10^{-6} \cdot e^{-5t} + 5 \times 10^{-7} \quad (1)$$

Abbildung 11 zeigt das Resultat der Phasenseparation nach 2 Sekunden mit zeitabhängigem ϵ . Die weisse Stelle ist verschwunden, jedoch zeigen sich auch andere Bereiche des Bilds weniger stark separiert im Vergleich zu Abbildung 7

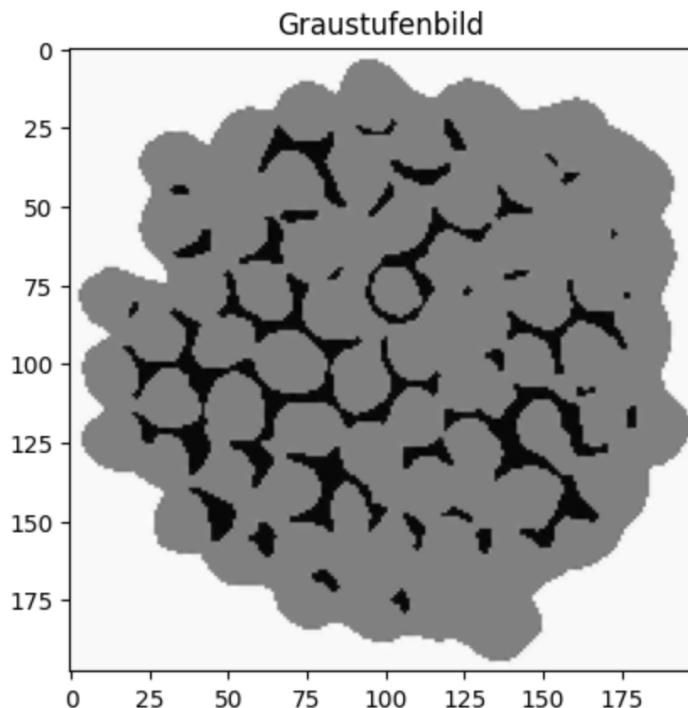


Abbildung 11: Resultat der Phasenseparation mit zeitabhängigem ϵ .

3.2 Beschleunigen des Verfahrens mit variablem Δt

Da sich das Bild mit zunehmender Zeit t pro einzelnen Zeitschritt Δt nicht mehr so stark verändert, kann das Δt nach jedem Zeitschritt etwas grösser gewählt werden um Rechnezeit zu sparen.

Die Zeitmessungen in Tabelle 1 wurden mit dem Apple M2 Prozessor durchgeführt.

konstantes $\Delta t = 0.25\text{s}$	variables Δt , +0.1s pro Iteration
68.105 Sekunden	42.479 Sekunden

Tabelle 1: Vergleich der Rechenzeiten für verschiedene Δt

4 Weitere Ansätze zur Verbesserung

In diesem Kapitel sollen Ideen vorgestellt werden, welche die Berechnung des Problems weiter optimieren.

4.1 Thresholding statt mathematische Bildverarbeitung

Thresholding kann verwendet werden, um Graustufen eindeutig voneinander zu trennen. Es können verschiedene Bereiche definiert werden, und alle Werte innerhalb eines Bereichs werden dann auf -1, 0 oder 1 gesetzt. Die Bereichsgrenzen liegen bei den Threshold-Werten. Angewendet auf dieses Problem mit Thresholds bei -0.5 und 0.5, ergibt sich das in Abbildung 12 dargestellte Resultat.

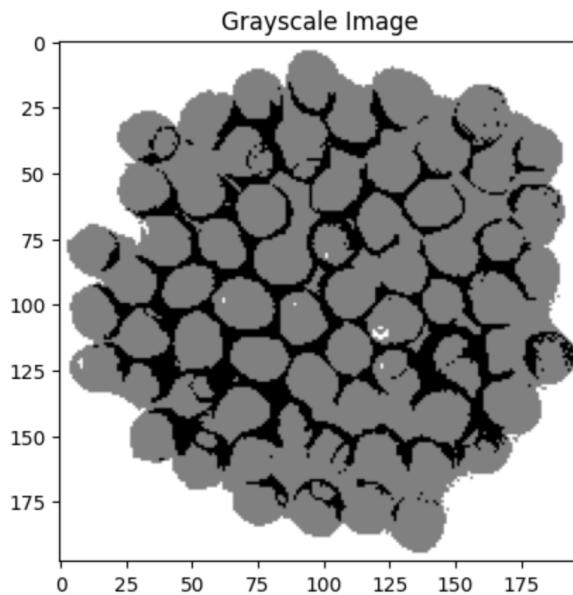


Abbildung 12: Separiertes Bild mit Thresholds bei -0.5 und 0.5

Es ist wie erwartet eine deutliche Trennung in drei Graustufen zu sehen, jedoch gibt es viele störende Stellen bei denen keine saubere Phasentrennung vorliegt. Um dieses Problem zu beheben, kann das Bild zuerst mit einem Gaußschen Unschärfefilter bereinigt werden bevor die Thresholds angewandt werden (Abbildung 13). Das resultierende Bild kommt nahe an das Resultat aus Abschnitt 2 und Abschnitt 3 heran, gewisse Details sind jedoch weniger gut ersichtlich.

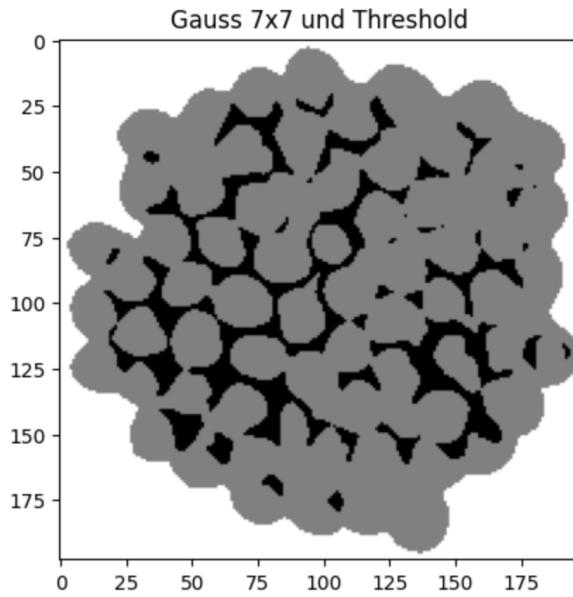


Abbildung 13: Separiertes Bild mit Gaußschem 7×7 -Unschärfefilter und Thresholds bei -0.5 und 0.5

4.2 Weniger Iterationen und Thresholding

Um die Methoden der mathematischen- mit denjenigen der klassischen Bildverarbeitung zu kombinieren, wurde das Energiefunktional nur für einen Zeitschritt (0.25 anstelle von 2 Sekunden) auf das Bild angewandt und anschliessend mit Thresholds ganz getrennt. Die Rechenzeit für diese Methode beträgt 8.554 Sekunden, was eine deutliche Steigerung gegenüber der rein mathematischen Bildverarbeitung (siehe Tabelle 1) darstellt.

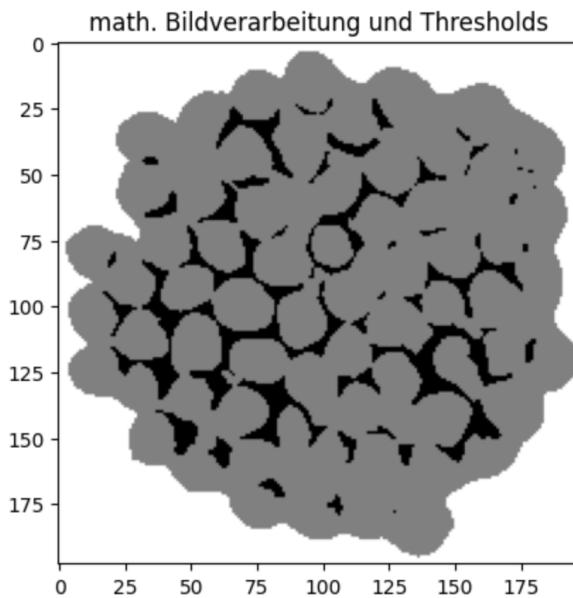


Abbildung 14: 5 Iterationen math. Bildverarbeitung und Thresholds bei -0.5 und 0.5

Anhang

Quellcode

Der im Zuge dieser Arbeit entwickelte Code ist in dem folgenden GitHub-Repository verfügbar. Das Repo ist privat, die Autoren können bei Bedarf jedoch den Zugriff freigeben. .
GitHub Repo: HANA Gruppe 2

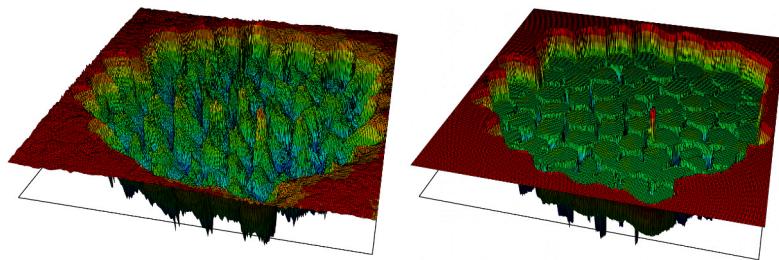
weitere Anhänge

- Aufgabenstellung
- Jupyter Notebook: 1D Problem Eps Vergleich (1D zur Veranschaulichung versch. Epsilon)
- Jupyter Notebook: Aufgabe2.1 (Variation mit NGSolve)
- Jupyter Notebook: Aufgabe2.2 (Variation analytisch)
- Jupyter Notebook: Aufgabe3 (Variables eps und dt)
- Jupyter Notebook: Aufgabe3 mit Threshold (weniger Iterationen, dafür Threshold)

HANA Numerik SEP Projekt

Simon Stingelin (stiw@zhaw.ch)

23. November 2023



Gruppe: Cyril Gabriele, Philip Kehl

Thema: Mathematische Bildverarbeitung

Lerninhalt:

- Berechnen der schwachen Gleichung und zweiten Variation eines Funktionals.
- Anwenden der Methode der finiten Elemente auf ein Problem aus der Bildverarbeitung.
- Lösen eines zeitabhängigen nichtlinearen Randwert Problems.

Abgabe:

- Die Abgabe erfolgt als Kurzbericht in Form eines **PDF**-Dokuments über die **moodle** Abgabe bis zum in der **moodle** Abgabe definierten spätesten Abgabetermin.
- Die numerischen Resultate sind im Bericht dokumentiert und können mit lauffähigen Skripts nachvollzogen werden.

Bewertung:

Note 1	Note 3	Note 4	Note 4.5	Note 5	Note 5.5	Note 6
Keine Abgabe	Nicht bewertbar: die Aufgaben wurden nicht ernsthaft bearbeitet.	Die Aufgaben wurden knapp bearbeitet und sind nur im Ansatz dokumentiert.	Die Aufgaben wurden halbwegs bearbeitet und sind sehr knapp Ansatz dokumentiert.	Die Aufgaben wurden bearbeitet und dokumentiert.	Die Aufgaben wurden detailliert bearbeitet und dokumentiert.	Die Aufgaben wurden detailliert bearbeitet und dokumentiert. Es wurden noch zusätzliche Aspekte bearbeitet.

1 Aufgabestellung

Die *Allen-Cahn-Gleichung* ist eine von mehreren Gleichungen, die zur Modellierung von Phasentrennungen von Mehrkomponenten-Legierungen verwendet werden. Der zeitabhängige lokale Zustand des inhomogenen Materials wird mit Hilfe eines Phasenfeldes $u(t, x)$ beschrieben. In einer Legierung kann die Funktion u die Konzentration eines von mehreren Bestandteilen am Ort x darstellen. Physikalische Überlegungen, die auf Landau zurückgehen, ergeben das folgende *Ginzburg-Landau-Funktional* für die freie Energie des Systems:

$$\begin{aligned} E : H^1(\Omega) &\rightarrow \mathbb{R} \\ u \mapsto E(u) &= \int_{\Omega} \frac{\varepsilon}{2} |\nabla u|^2 + W(u) \, dx. \end{aligned} \tag{1}$$

Dabei ist Ω ein begrenzter Bereich im \mathbb{R}^2 . Symmetrieargumente für homogene Materialien implizieren, dass die Expansion von $W(u)$ in Bezug auf u nur aus geraden Potenzen besteht. Wir wählen

$$W(u) = u^2 (1 - u^2)^2.$$

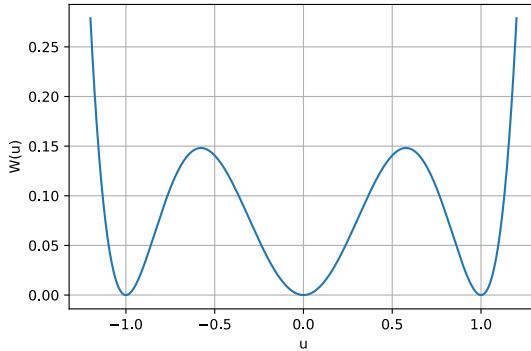


Abbildung 1: Separationsenergie $W(u)$.

Die Legierung wird sich dann schließlich in Regionen aufteilen, in denen entweder $u = -1$, $u = 0$ oder $u = 1$ ist. Diese Regionen sind von diffusen Grenzflächen umgeben, deren mittlere Breite selbst in der Physik der kondensierten Materie von Interesse ist. Der Summand $|\nabla u|^2$ trägt den räumlichen Schwankungen (Fluktuationen) im Medium Rechnung. Die beiden Teile des Energiefunktional konkurrieren:

- $W(u)$ will u in eines der drei Minima ($-1, 0$ und 1) des nichtlinearen Terms $u^2 (1 - u^2)^2$ ziehen, wodurch die Phasen getrennt werden,
- während $|\nabla u|^2$ steile Gradienten nicht bevorzugt.

Unser Ziel ist es, die zeitliche Entwicklung eines gegebenen Anfangsphasenfeldes $u_0(x) = u(t = 0, x)$ zum stationären Zustand $u(t \rightarrow \infty, x)$ zu bestimmen, der das Ginzburg-Landau-Funktional minimiert. Die Lösung der Allen-Cahn-Gleichung löst daher

$$\int_{\Omega} \frac{\partial u}{\partial t} v \, dx = -\delta E(u)v \quad \forall v \in H^1(\Omega). \tag{2}$$

1.1 Input zur Numerik

Wir verwenden die Funktion `Variation`, um das Energieminimierungsproblem zu definieren und es mit einer *impliziten Euler-Diskretisierung* zu kombinieren:

$$Mu^{(n+1)} - Mu^{(n)} = -\Delta t \delta E(u^{(n+1)})$$

was wir wiederum als ein nichtlineares Minimierungsproblem interpretieren können, mit der Energie

$$E^{IE}(u) = \int_{\Omega} \frac{\varepsilon}{2} |\nabla u|^2 + u^2(1-u^2)^2 + \frac{1}{2\Delta t} |u - u^{(n)}|^2 dx, \quad (3)$$

wobei $u^{(n)}$ die Lösung aus dem letzten Zeitschritt sei.

In jedem Zeitschritt benutzen wir das Newton-Verfahren (vgl. [Algorithmus 11.1](#)), um die nichtlineare Gleichung (2) zu lösen. Mit Hilfe der symbolischen Beschreibung des Energiefunktional in `NGSolve` können folgende Größen berechnet werden:

- das Energiefunktional (`Energy`)

$$E(u) \quad (E : V \rightarrow \mathbb{R})$$

- die Gateau-Ableitung für ein gegebenes u (`Apply`)

$$A(u)(v) = E'(u)(v) \quad (A(u) : V \rightarrow \mathbb{R})$$

- die zweite Ableitung (`AssembleLinearization`)

$$(\delta A)(w)(u, v) \quad (\delta A(w) : V \times V \rightarrow \mathbb{R})$$

Definieren Sie die Bilinear mit Hilfe einer zusätzlichen `GridFunction` für die Lösung zum Zeitpunkt t . Die Lösung zum Zeitpunkt $t + \Delta t$ muss mit Hilfe der Proxy-Funktion `u = V.TrialFunction()` definiert werden (vgl. Listing 1).

```

1  eps = 1e-6
2  dt = 2.5e-1
3
4  a = BilinearForm (V, symmetric=False)
5  a += Variation((eps/2*grad(u)*grad(u)
6                  +(u**2*(1-u*u)**2)
7                  + 0.5/dt*(u-gfu.old)**2)*dx)
```

Listing 1: Definition der Bilinearfunktion

2 Aufgaben

Aufgabe 1

Berechnen Sie die erste und zweite Variation des Funktionals E^{IE} (3). Zeigen Sie, dass die erste Variation mit dem impliziten Euler-Verfahren übereinstimmt.

Aufgabe 2

Basierend auf den Bilddaten `img.pk1` (vgl. Abb. 2 links) soll die Phasenseparation mit Hilfe des Ginzburg-Landau-Funktionalen angewandt werden. Implementieren Sie daher eine zeitabhängige Berechnung der Lösung, in der Sie in jedem Zeitschritt das nichtlineare Problem (2) mit der Newton-Iteration lösen.

1. Wie sieht die Lösung für $\varepsilon = 10^{-6}$ nach $T = 2\text{s}$ aus?
2. Studieren Sie den Einfluss der Diffusionskonstante ε . Illustrieren Sie die Abhängigkeit.
3. Erstellen Sie ein Histogramm der Grauwerte für das initiale und nach $T = 2\text{s}$ berechnete Bild. Ist eine Phasenseparation ersichtlich?

Benutzen Sie eine FEM Diskretisierung 2. Ordnung (`order = 2`).

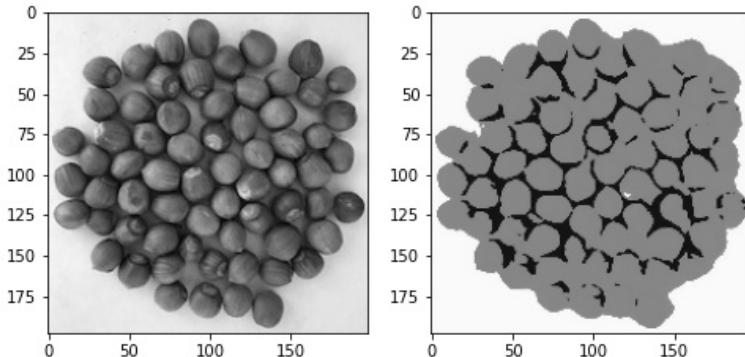


Abbildung 2: Links Ausgangs Daten, rechts Phasenseparation nach $T = 2\text{s}$.

Aufgabe 3

1. In der Abbildung 2 ist im rechten Bild, der Phasenseparation zu sehen, dass es eine störende helle Stelle gibt. Versuchen Sie diesen mit Hilfe einer zeitabhängigen Diffusionskonstante zu eliminieren.
2. Beschleunigen Sie zu dem Ihre Verfahren, in dem Sie das Δt immer grösser werden lassen.

Sie können dazu sowohl die Zeit t , wie auch den Zeitschritt dt und die Diffusionskonstante eps als Parameter definieren (vgl. Listing 2). Versuchen Sie das Ergebnis zu optimieren.

```

1 # parameter initialization
2 t = Parameter(0)
3 # set value
4 t.Set(t.Get()+dt.Get())
5 # get value
6 print(t.Get())

```

Listing 2: Definition Parameter

A Bemerkungen

Die Daten können einfach gemäss Listing 3 geladen werden. Achten Sie darauf, dass die Darstellung der Bilder den Ursprung oben links haben und die y -Achse nach unten zeigt. Die Daten können mit `VoxelCoefficient` als algebraischer Ausdruck benutzt werden. dabei übergeben wir die y -Achse vertauschten Daten, um einen direkten Vergleich mit dem Bild zu haben. Für das Erstellen des strukturierten Mesh sei auf das Listing 4 verwiesen. Mit `gfu.Set(cfImg)` kann u initialisiert werden.

```

1 import pickle
2 with open('img.pkl','rb') as fp:
3     img = pickle.load(fp)
4 plt.imshow(img,interpolation='none', cmap='gray')

```

Listing 3: Laden der Bilddaten

```

1 mesh = MakeStructured2DMesh(nx = np.int64((img.shape[0]-1)/1),
2                             ny=np.int64((img.shape[1]-1)/1),
3                             quads=False)
4 cfImg = VoxelCoefficient((0,0),(1,1),
5                           img[np.arange(img.shape[0]-1,-1,-1)])
6 Draw(cfImg,mesh,'cfImg')

```

Listing 4: Definition Mesh und VoxelCoefficient Funktion für die Bilddaten.