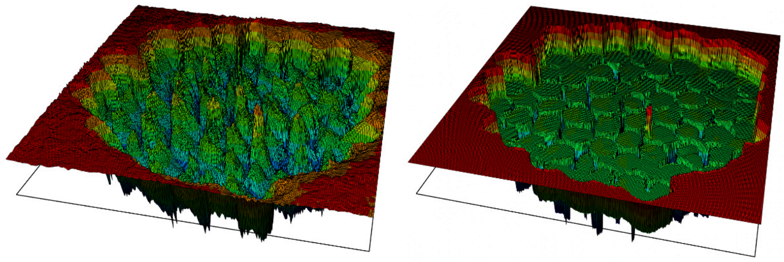
HANA Numerik SEP Projekt

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 Abgabezeitpunkt.
- 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:

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

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

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

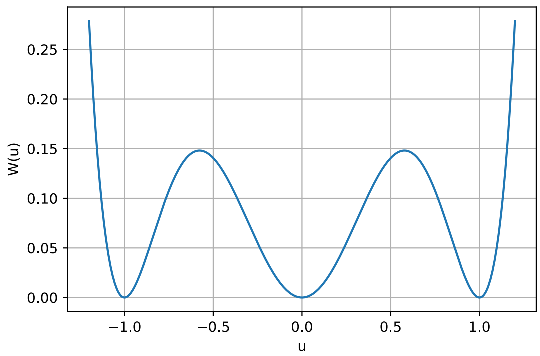


Abbildung 1: Separationsenergie $W(u)$.

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 Energiefunktionals 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 $u(t \to \infty, x)$ zu bestimmen, der das Ginzburg-Landau-Funktional minimiert. Die Lösung der Allen-Cahn-Gleichung löst daher

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

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

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

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 Energiefunktionals in NGSolve können folgende Grössen berechnet werden:

- das Energiefunktional (`Energy`)

$$E(u) \qquad (E : V \to \mathbb{R})$$

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

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

- die zweite Ableitung (`AssembleLinearization`)

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

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-Funktione `u = V.TrialFunction()` definiert werden (vgl. Listing 1).

```
eps = 1e-6
dt = 2.5e-1

a = BilinearForm (V, symmetric=False)
a += Variation((eps/2*grad(u)*grad(u)
               +(u**2*(1-u*u)**2)
               + 0.5/dt*(u-gfuold)**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.pkl` (vgl. Abb. 2 links) soll die Phasenseparation mit Hilfe des Ginzburg-Landau-Funktionals 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 $T = 2$s aus?
2. Studieren Sie den Einfluss der Diffusionskonstante ε. Illustrieren Sie die Abhängigkeit.
3. Erstellen Sie ein Histogram der Grauwerte für das initiale und nach $T = 2$s berechnete Bild. Ist eine Phasenseparation ersichtlich?

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

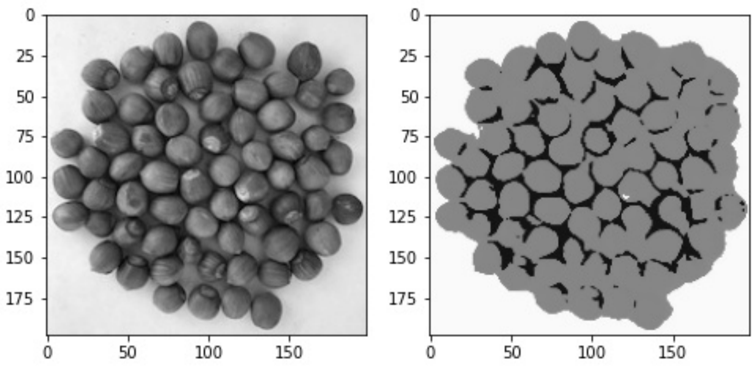


Abbildung 2: Links Ausgangs Daten, rechts Phasenseparation nach $T = 2$s.

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.

```
# parameter initialization
t = Parameter(0)
# set value
t.Set(t.Get()+dt.Get())
# get value
print(t.Get()
```

Listing 2: Definition Paramter

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.

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

Listing 3: Laden der Bilddaten

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

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