Phasenfeld-Optimierung

Vinh-An Pham

28. November 2020

Inhaltsverzeichnis

1 Ankündigung 1

2 Einführung 1

3 Minimierungsverfahren 2

4 Abstiegsverfahren 4

5 Auswertung 5

6 Aufgabe 5

1 Ankündigung

Die nächsten vereinbarten Treffen sind am Montag, den 30. November (optional) und am darauf folgenden Montag, den 07. Dezember.

2 Einführung

Themen in dieser Vorlesung ist die Beschreibung von Phasenfeldern und die Optimierung.

Sei $v\colon \Omega \to [-1, 1]$ eine genügend oft differenzierbare Funktion. Es soll auf dem Material $M \subseteq \Omega$ den Wert $v|_M = 1$ und außerhalb davon den Wert $v|_{\Omega\setminus M} = -1$ sein. Solch einer Funktion kann es aufgrund der Differenzierbarkeit nicht existieren. Daher soll der Fehlertoleranz $\varepsilon \in \mathbb{R}^+$ eingeführt werden, sodass unter den Definitionen

$$M_\varepsilon := \{x \in M : \mathrm{Dist}(\partial M, x) < \varepsilon\} \text{ und } (\Omega\ M)_\varepsilon := \{x \in \Omega \setminus M : \mathrm{Dist}(\partial M, x)\} \tag{1}$$

der Abstand zum Rand möglichst klein, aber nicht Null sein darf. Nur so kann die differenzierbare Funktion v auf M_ε den Wert 1 und auf $(\Omega \setminus M)_\varepsilon$ den Wert -1 haben und da zwischen $[-1, 1]$ annehmen.

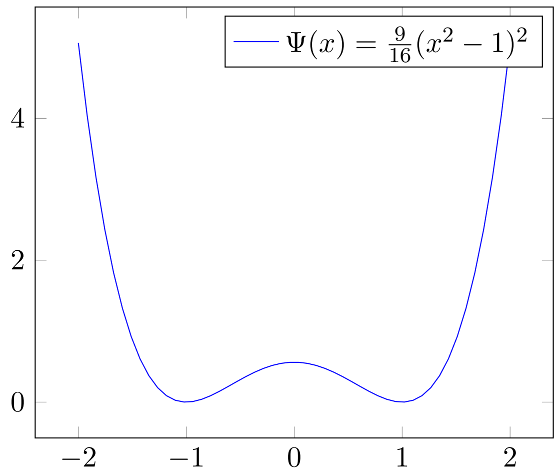
Die folgende Energiefunktion J soll minimiert werden:

$$J(v) := \int_\Omega \frac{1}{2} \left(\varepsilon |\nabla v|^2 + \frac{1}{\varepsilon} \Psi(v) \right) \mathrm{d}x \tag{2}$$

Dabei ist die definierte Funktion

$$\Psi \colon \mathbb{R} \ni x \mapsto \frac{9}{16}(x^2 - 1)^2 \in \mathbb{R} \tag{3}$$

die Doppel-Welle-Potential-Funktion (double wave potential) sein. Es hat die Nullstellen bei -1 und 1. Dieser wird als Strafterm für die Minimierung der Funktion J gebraucht, damit bei geringer Abweichung zur Minimierer der Fehler möglichst groß wird.



Dabei kann für den Übergang $\varepsilon \to 0$ die Funktion J gegen den Perimeterfunktion (punktweise) konvergieren, sodass die Grenzfunktion auf $M \setminus \partial M$ den Wert 1 hat und auf $(\Omega \setminus M) \setminus \partial M$ den Wert -1.

Beispiel 2.1 (Phasenübergänge):
Es gibt unterschiedliche Lösungen, die abhängig von der Wahl der Randwerte sind.

3 Minimierungsverfahren

In der Praxis bei FENICS hilft für den einfachen Ansatz das Newton Verfahren. Dies funktioniert sogar für nicht-lineare Probleme. Zum Beispiel kann die Energiefunktion 2 darauf getestet werden. Für die Shape-Optimierung soll ein bestimmtes Verfahren angewendet werden, welches auf Gradientenabstieg basiert. Das ist der Anfang.

Intuitiv seien v_0 der Startwert und $\tau \in \mathbb{R}^+$ die Steuerkonstante. Darauf wird anhand des Gradientenabstiegs eine Folge rekursiv definiert durch:

$$\forall k \in \mathbb{N} : v_{k+1} = v_k - \tau \cdot g_k \text{ mit } g_k := \nabla J(v_k) \tag{4}$$

Die Frage ist, was $\nabla J(v_k)$ zu verstehen ist. Nach Analysis II enthält dieser Richtungsableitungen $\varphi \neq 0$

$$\mathrm{D}J(v)(\varphi) = \frac{\mathrm{d}}{\mathrm{d}t} J(v + t \cdot \varphi)\Big|_{t=0} \tag{5}$$

Nach Fenics gelten die Befehle, wobei φ die Testfunktion ist:

$$\text{derivative}(J, v) \tag{6}$$

$$\text{derivative}(J, v, \varphi) \tag{7}$$

In der Praxis wird Fenics sämtliche Basisvektoren im endlich-dimensionalen Fall durchtesten und berechnen. Wichtig ist, dass der erste Befehl keinen Namen ausgibt. Dies sind lineare Funktionale. Der Ansatz

$$\langle g, \varphi \rangle \overset{!}{=} \mathrm{D}J(v)(\varphi) \tag{8}$$

wobei $\langle ., . \rangle$ das Skalarprodukt ist, in diesem Fall:

$$\langle g, \varphi \rangle = \int g \cdot \varphi \mathrm{d}x = \mathrm{D}J(v)(\varphi) \tag{9}$$

führt zum Befehl von Fenics:

$$\text{solve}(g * \varphi * \mathrm{d}x == \text{derivative}(J, v, \varphi)) \tag{10}$$

Weil alle Basisvektoren durchgetestet werden müssen resultiert sich ein lineares Gleichungssystem. Zusammen gilt in 4:

$$v_{k+1} = v_k - \tau \cdot g_k \Leftrightarrow g_k = -\frac{v_{k+1} - v_k}{\tau} \tag{11}$$

Oben angewendet gilt:

$$\int \frac{v_{k+1} - v_k}{\tau} \cdot \varphi \mathrm{d}x = -\mathrm{D}J(v)(\varphi) \tag{12}$$

Problematisch ist, was in v eingesetzt werden soll, v_k für ein explizites Gleichungssystem oder v_{k+1} für ein implizites Gleichungssystem. Einfacher ist die erste Variante für den Praxis. Um Fenics anwenden zu können, gibt es zwei Fälle:

1. Gleichungssystem ist linear:
   Dann ist es von technisch rechnerischen Aufwand her, alles auf eine Seite umzustellen, sodass die andere Seite also Null ergibt. Fenics berechnet lineare Probleme schnelle als allgemeinere Probleme.

2. Gleichungssystem ist nicht linear:
   Dann kann das Gleichungsystem stehen lassen.

Mit dieser ersten Variante wird mit Linien-Suche numerisch gelöst.

4 Abstiegsverfahren

Eindimensionaler Fall: Sei j eine Funktion definiert durch, sofern v und g bekannt ist:

$$j\colon \mathbb{R} \ni t \mapsto j(t) := J(v + t \cdot g) \in \mathbb{R} \tag{13}$$

Die Ableitung ist sicherlich $j'(t) = \mathrm{D}(v+t\cdot g)(g)$. Die Funktion j wird durch ein Polynom zweiten Grades p approximiert. Dafür sind drei Informationen mindestens nötig für die Modellierung:

$$p(0) = j(0) \text{ und } p'(0) = j'(0) \text{ und } p(T) = j(T) \tag{14}$$

Hier muss $T \neq 0$ gefordert werden. Somit hat das Polynom die Darstellung:

$$p(t) = j(0) + j'(0) \cdot t + \frac{j(T) - j(0) - j'(0)}{T^2} \tag{15}$$

Dieses Polynom besitzt eine Minimalstelle bei:

$$\hat{t} := -\frac{j'(0) \cdot T^2}{2(j(T) - j'(0) \cdot T - j(0))} \tag{16}$$

Ein kurzer Überlick auf 13 liefert:

$$j(t) = J(v + t \cdot g) \tag{17}$$
$$j(0) = J(v) \tag{18}$$
$$j'(0) = \mathrm{D}J(v)(g) \tag{19}$$
$$j(T) = J(v + T \cdot g) \tag{20}$$

Was jetzt bekannt ist, dass $\hat{t}$ die Abstiegsrichtung ist. Was ist aber mit der Schrittweite? Es kann sein, dass ein zu großer Schritt vorbei an der Minimalstelle führt.

$$j(\hat{t}) = J(v - \hat{t} \cdot g) \overset{!}{\leq} J(v) \tag{21}$$

Damit lässt sich durch mehrfaches Testen schnell lösen: Sei β eine Zahl mit $0 < \beta < 1$ gewählt. Teste für eine wachsende Zahl $n \in \mathbb{N}$, dass es gilt:

$$J(v - \beta^n \cdot \hat{t} \cdot g) \leq J(v) \tag{22}$$

Alternativ lässt sich mit der Steigungsfunktion für eine Zahl μ mit $0 < 2\mu < 1$ für eine gesuchte Zahl $n \in \mathbb{N}^+$ lösen:

$$J(v - \hat{t} \cdot g) \leq J(v) + \mu^n \cdot \hat{t} \cdot \mathrm{D}J(v)(g) \tag{23}$$

5 Auswertung

Wie wird eigentlich $J(v)$ berechnet? Nach FENICS wird über Folgendes gegeben:

$$\text{assemble}(J) \tag{24}$$

Dies liefert eine Fließkommazahl, wenn J keine Test- oder Trial-Funktion enthält. Es lässt sich Beispielsweise darstellen:

$$v = \text{Function}(V)j = \text{assemble}(v * v * dx) \tag{25}$$

Was verbirgt sich hinter diesem Auswertungsbefehl? Wenn φ eine Testfunktion ist und v_t eine Trial-Funktion ist kommt bei

1. assemble $(v * \varphi * \mathrm{d}x)$ ein Vektor
2. assemble $(v_t * \varphi * \mathrm{d}x)$ eine Matrix

Nach Fenics werden Basisvektoren zur Berechnung herangezogen, je nach dem welche Dimensionsgröße es ist. Die Ergebnisse werden im ersten Fall in einem Vektor und im zweiten Fall in einer Matrix eingeschrieben.

Eine wesentlich billigere Methode ist:

$$\text{assemble}(\text{derivative}(J, v)) \tag{26}$$

Dieser Befehl liefert ein Vektor, eine Art Gradient von J. Es lässt sich lösen beispielsweise:

$$b = \text{Function}(V) \tag{27}$$
$$\text{assemble}(\text{derivative}(J, v), \text{tensor} = b.\text{vector}()) \tag{28}$$

Die Koordinaten von der Funktion b bilden daraus im hinteren Teil ein Vektor.

6 Aufgabe

Grob formuliert: Teste mit Randbedingungen Ihrer Wahl mit Newton und Gradienten-Abstieg. Die Python-Bibliothek SCIPY liefert mit dem Befehl

$$\text{scipy.minimize} \tag{29}$$

die Lösung für die Minimierung und ist eine Hilfe.