

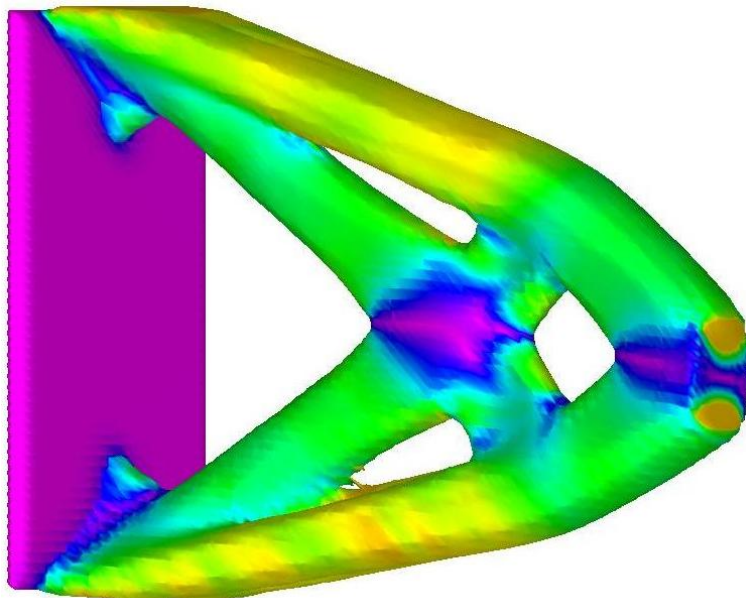


UNIVERSITÄT DUISBURG-ESSEN, CAMPUS DUISBURG
FACHBEREICH MATHEMATIK

Levelsetverfahren in der Shapeoptimierung

Diplomarbeit
von
Martin Pach

Duisburg, September 2005



Inhaltsverzeichnis

1	Einleitung	5
2	Problembeschreibung	8
2.1	Linearisierte Elastizität	8
2.2	Shapefunktion	9
3	Shape Sensitivity Analysis	11
3.1	Shapefunktionen und Shapeableitungen	11
3.2	Erster Ordnung Semiableitungen	16
3.3	Struktursatz	18
3.4	Shapeableitung für eine Klasse von Shapefunktionen	20
4	Levelsetmethoden	30
4.1	Levelsetgleichung	31
4.2	Extension Velocity	32
4.3	Reinitialisierung	33
4.4	Shapeableitung in Levelsetformulierung	34
5	Shapegradient und Regularisierung	36
5.1	Shapegradient	36
5.2	Regularisierung	37
6	Diskretisierung	39
6.1	Diskretisierung der Elastizität	39
6.2	Diskretisierung der Shapeableitung	40
7	Optimierungsverfahren	44
7.1	Schrittweitensteuerung	44
7.2	Optimierungsalgorithmen	45
8	Numerische Ergebnisse	47
8.1	Kurzer Kranarm	47
8.2	Langer Kranarm	57
8.3	Würfel	62
9	Schlussbemerkung	64
	Literatur	67

1 Einleitung

Der Begriff Shape-Optimierung beschreibt jene Art von Optimierungsproblemen, bei welchen die zu Optimierende Variable nicht etwa eine Menge von Parametern oder Funktionen, sondern die Form und Struktur eines geometrischen Objektes ist. Diese Art von Optimierung findet man in einem breiten Spektrum von Problemen. Angefangen bei der Suche nach der optimalen Form oder Oberflächenstruktur der Tragflächen eines Flugzeuges über Bildregistrierung [25] bis hin zur Optimierung von elastischen Strukturen. Die mathematischen Grundlagen für diesen Themenbereich werden unter dem Begriff *Shape analysis* zusammengefasst. Die Terminologie *Shape analysis* wurde in unterschiedlichen Zusammenhängen eingeführt. Zum einen in der klassischen Mechanik und zum anderen in der Theorie der partiellen Differentialgleichungen.

In der klassischen Mechanik umfasst die *Shape analysis* viele teilweise auch speziell an das Problem angepasste Methoden bei der Theorie der Gestaltung von Balken, Platten, Muschelschalen, Brückenbögen und Stützbalken. Dabei ist das Ziel die Minimierung eines Funktionals, welches z. B. die von einer auf den betrachteten Körper einwirkenden Kraft verrichtete Arbeit beschreibt. Dies geschieht durch Bestimmung der charakteristischen formgebenden Parameter des Körpers, wie z.B. Dicke oder Breite.

Ein Beispiel dafür ist die Optimierung der Dicke einer Muschelschale [21]. Hierbei ist der Parameter, der die Dicke beschreibt, auf einem 2-dimensionalen Gebiet verteilt. Wenn die Dicke an einer Stelle Null wird, entstehen Löcher und induzieren topologische Änderungen in dem zugrundeliegenden 2-dimensionalen Gebiet. Dies führt in den Bereich der 'topologischen Optimierung', bei der es z.B. um das Finden der optimalen Anzahl von Löchern in Strukturen oder das Auftreten von Mikrostrukturen geht. Diese Art von Problemen wurden vor allem mit Methoden aus der Homogenisierungstheorie und Γ -Konvergenz untersucht.

In der Theorie der partiellen Differentialgleichungen bezeichnet man mit *Shape analysis* die Sensibilität der Lösung eines Randwertproblems bezüglich der Geometrie oder Variation des Gebietes, auf der das partielle Differentialgleichungssystem definiert ist. Die Analysis aus diesem Themenbereich wird als *Shape sensitivity analysis* bezeichnet und wurde schon weitgehend entwickelt (siehe [30] [33] [37]). Für eine Einführung verweisen wir auf die neuere Literatur von Sokolowski und Zolesio [38] und von Delfour und Zolesio [24]. Die *Shape sensitivity analysis* ist eine sehr allgemeine Methode und kann auf eine große Klasse von Shape-Optimierungsproblemen angewandt werden. Die zugrundeliegende Idee ist die Variation eines Funktionals $J(\Omega)$ zu berechnen, wenn das Gebiet Ω in Richtung einer Transformation (Diffeomorphismus) gestört wird. Wobei diese Transformation durch Geschwindigkeitsfelder in Normalenrichtung des Randes von Ω erzeugt wird. Unter geeigneten Voraussetzungen kann die Stetigkeit von $J(\Omega)$, ja sogar Existenz von Ableitungen erster und zweiter Ordnung, gezeigt werden. Die Ableitung von $J(\Omega)$ bezüglich Ω werden als *Shapeableitungen* bezeichnet. Diese wollen wir nutzen, um mit einem Gradientenverfahren ein Shape-Optimierungsproblem numerisch zu lösen.

Wir widmen uns in dieser Arbeit der Shapeoptimierung von elastischen Strukturen. Dies ist ein sehr wichtiges und populäres Arbeitsfeld. Ganz wesentlich für den Optimierungsprozess sind drei Sachen.

- (i) Erstens ist es notwendig eine flexible Methode zur Beschreibung von Gebieten oder Körpern zur Verfügung zu haben.
- (ii) Zweitens die oben angesprochene Theorie der *Shape sensitivity analysis* zur Generierung von Abstiegsrichtungen in Form von Deformationen.
- (iii) Und drittens ist es wichtig einen effizienten und akkuraten Elastizitätslöser zur Verfügung zu haben, um die elastischen Eigenschaften des zu optimierenden Körpers zu beschreiben und um den Zeitaufwand des Optimierungsprozesses gering zu halten.

Wir wollen an dieser Stelle kurz auf schon existierende Arbeiten zu diesem Thema eingehen, welche zu den drei oben angesprochen Punkten gleiche oder ähnliche Methoden benutzt haben.

Den ersten Punkt berücksichtigend wollen wir uns der Levelsetmethoden bedienen, welche von Osher und Sethian [32] entwickelt wurden, um Bewegung und Evolution von Flächen zu beschreiben. Hierbei werden Flächen als Isolevel einer sogenannten Levelsetfunktion beschrieben. Levelsetmethoden haben sich zu den mit am erfolgreichsten Methoden in der Bildverarbeitung entwickelt und bieten sehr robuste, akkurate und effektive Verfahren zur Berechnung der Evolution von Flächen.

In der Arbeit von Sethian und Wiegmann [35] werden Levelsetmethoden für die Shapeoptimierung von 2-dimensionalen elastischen Strukturen benutzt. Die Form des Körpers wurde dabei durch einen freien Rand beschrieben, welcher selbst auf einem fixierten Gitter mit der 'Immersed Interface method' dargestellt wird. Die Belastbarkeit der Struktur wurde dort durch eine eigens für dieses Verfahren entwickelte Methode verbessert, die auf einer der 'Von Mises' equivalenten Spannung basiert. Die in dieser Arbeit erzeugte Gebietsvariation basierte aber nicht auf der *Shape sensitivity Analysis*.

Die Arbeit von Allaire, Jouve und Toader [3] greift sowohl auf die Levelsetmethoden als auch auf die *Shape sensitivity analysis* zurück, um ein Gradientenverfahren zu implementieren, das dann in gewissem Sinne gegen eine Lösung konvergiert. In dieser Arbeit zeigt sich der Vorteil der *Shape sensitivity analysis*, da hier Shapeableitungen von Funktionalen unterschiedlichen Types behandelt werden. Dort wird eine ganze Reihe von interessanten Ergebnissen (teilweise sogar 3-dimensional) präsentiert. Der Optimierungsprozess startet mit einem Initialgebiet Ω , das in einem Konstruktionsgebiet D enthalten ist und deformiert Ω iterativ gemäß der berechneten Shapeableitung. Dabei muß nach jedem Iterationsschritt die Elastizitätsgleichung für das neue Gebiet gelöst werden. Die Elastizitätsgleichung wird dort auf dem ganzen Konstruktionsgebiet gelöst, indem man das Gebiet $D \setminus \Omega$ mit einem 'weichen' Ersatzmaterial auffüllt. Jedoch läßt diese Arbeit noch Verbesserungen im Bezug auf den verwendeten Elastizitätslöser und vorallem auf die Generierung von Abstiegsrichtungen zu.

Einen Schritt in Richtung eines theoretischen Konzepts zur allgemeinen Berechnung von Shapeableitungen und Abstiegsrichtungen machen die Arbeit von Burger [11] und die Arbeit von Burger und Osher [12]. Hier werden durch Wahl verschiedener Skalarprodukte auf

dem Rand des betrachteten Gebietes und der so erzeugten Abstiegsrichtungen unterschiedliche Gradientenflüsse definiert. Weiter wird die Möglichkeit vorgestellt, die Shapeableitungen, die die *Shape sensitivity analysis* liefert, in den Kontext der Levelsetfunktionen zu übertragen. Dies ist ein Aspekt der auch in unserer Arbeit eine zentrale Rolle gespielt hat. Wir haben die Absicht den Optimierungsprozess komplett in eine Levelsetformulierung zu übersetzen.

Damit kommen wir jetzt zu einem Überblick über den Inhalt unserer Arbeit und den Unterschieden zu den oben angesprochenen.

- In Kapitel 2 und 3 wird das betrachtete Optimierungsproblem beschrieben. Nach einem kurzen Resümee der Elastizitätstheorie und der Formulierung des Optimierungsproblems in Kapitel 2 folgt in Kapitel 3 ein Überblick der theoretischen Grundlagen der *Shape Sensitivity Analysis*.
- Die Levelsetformulierung des Optimierungsprozesses wird in Kapitel 4 motiviert. Nach einer Einführung der gängigen Levelsetmethoden wird die Beziehung zwischen der Variation einer Levelsetfunktion ϕ und der geometrischen Veränderung des zugehörigen Gebietes $\Omega = [\phi < 0]$ durch Geschwindigkeitsfelder aufgezeigt. Damit ist es dann möglich die gewünschte Levelsetformulierung des Optimierungsprozesses herzuleiten.
- In Kapitel 5 wird aus den Ergebnissen aus Kapitel 4 eine allgemeine Definition eines Shapegradienten bereitgestellt. Hier sind unsere Überlegungen eine Verallgemeinerung der Betrachtungen aus [11], da uns die Levelsetformulierung eine Fülle von neuen Möglichkeiten bietet, Abstiegsrichtungen zu generieren.
- In Kapitel 6 wird die von uns verwendete Diskretisierung dargelegt. Diese basiert auf der sogenannten CFE-Diskretisierung (Composite Finite Elements), welche eine gute Auflösung eines Gebietsrandes auch bei einer recht geringer Gittertiefe gewährleistet. Der hierzu von Liehr und Rumpf [29] entwickelte Elastizitätslöser ermöglicht es uns nur auf dem tatsächlich betrachteten Gebiet die Elastizitätsgleichung zu lösen. Im Gegensatz zu dem Löser in der Arbeit von Allaire, Jouve und Toader [3]. Die diskrete Darstellung der Shapeableitung und des Shapegradienten bezüglich der CFE-Diskretisierung ist Thema im zweiten Teil von Kapitel 6.
- In Kapitel 7 wollen wir das Gradientenverfahren beschreiben, das wir benutzen. Wir haben uns für ein Newton-Verfahren mit Schrittweitensteuerung entschieden.
- Die numerischen Ergebnisse dieser Arbeit werden dann in Kapitel 8 vorgestellt und in Kapitel 9 diskutiert.

2 Problembeschreibung

2.1 Linearisierte Elastizität

Zuerst wollen wir kurz die Resultate der Elastizitätstheorie zusammentragen. Beginnen wir also mit der Beschreibung der Problemstellung.

Sei $\Omega \in \mathbb{R}^3$ ein beschränktes Gebiet ohne innere Ränder, welches einen Körper, bestehend aus einem isotropen elastischen Material, darstellen soll. In unserem Modellproblem wirken nun auf diesen Körper zum einen Volumenkräfte, bezeichnet mit f , und zum anderen Oberflächenkräfte, bezeichnet mit g , gemäß dem Hook'schen Gesetz ein. Der Rand von Ω bestehe aus zwei disjunkten Teilen $\partial\Omega = \Gamma_N \cup \Gamma_D$ mit Dirichletrandbedingungen auf Γ_D und Neumannrandbedingungen auf Γ_N . Die Dirichletfunktion setzen wir der Einfachheit halber konstant gleich null. Gesucht ist die Deformation $\theta : \Omega \rightarrow \mathbb{R}^3$, die dieser Körper erfährt.

Betrachtet man das Verschiebungsfeld $u : \Omega \rightarrow \mathbb{R}^3$, definiert durch $\theta = \mathbb{I} + u$, dann erhält man den linearisierten Verzerrungstensor $\epsilon(u)$

$$\epsilon(u) := \frac{\nabla u + \nabla u^T}{2}$$

Die mechanische Spannung \mathcal{S} wird durch den Elastizitätstensor \mathcal{C} beschrieben :

$$\mathcal{S} = \mathcal{C}(\epsilon(u)) = \lambda \operatorname{tr}(\epsilon(u)) \mathbb{I} + 2\mu \epsilon(u)$$

mit den Lamé-Koeffizienten λ und μ als materialabhängigen skalarwertigen Konstanten. Für (3 x 3)-Tensoren \mathcal{A} und \mathcal{B} verwenden wir das Skalarprodukt

$$\mathcal{A} : \mathcal{B} = \sum_{i,j=1}^3 \mathcal{A}_{ij} \mathcal{B}_{ij} = \operatorname{tr} [\mathcal{A}^T \mathcal{B}]$$

Die Minimierung elastischen Energie

$$E(v) = \frac{1}{2} \int_{\Omega} \epsilon(v) : \mathcal{C}(\epsilon(v)) - f \cdot v \, dx - \int_{\partial\Omega} g \cdot v \, d\nu \quad (2.1)$$

mit $f \in L^2(\Omega)^3$ und $g \in L^2(\partial\Omega)^3$ über dem Hilbertraum

$$H_{\Gamma_D}^1(\Omega) := \{v \in H^{1,2}(\Omega)^3 : v|_{\Gamma_D} = 0\}$$

liefert das Minimum u

$$E(u) = \min_{v \in H_{\Gamma_D}^1(\Omega)} E(v)$$

welches den mechanischen Gleichgewichtszustand beschreibt. Die zugehörigen Euler-Lagrange Gleichungen sind im folgenden Differentialgleichungssystem für u beschrieben

$$(P_u) \quad \begin{cases} -\operatorname{div}(\mathcal{C}(\epsilon(u))) & = f & \text{in } \Omega \\ u & = 0 & \text{on } \Gamma_D \\ \mathcal{C}(\epsilon(u)) \cdot n & = g & \text{on } \Gamma_N \end{cases}$$

Für den Beweis der Existenz einer Lösung u verweisen wir auf [22].

2.2 Shapefunktion

Die von uns betrachtete Funktion sieht wie folgt aus.

$$J(\Omega) = \int_{\Omega} f u \, dx + \int_{\Gamma_N} g u \, d\nu = \int_{\Omega} \mathcal{C}(\epsilon(u)) : \epsilon(u) \, dx \quad (2.2)$$

Wobei u die Lösung zu (P_u) ist, f und g wie oben die Volumenkraft und Oberflächenkraft beschreiben. $J(\Omega)$ kann als die vom Körper Ω in Form von Spannungsenergie aufgenommene Energie interpretiert werden. Das Optimierungsproblem lässt sich nun schreiben als Minimierungsproblem von $J(\Omega)$ bezüglich Ω :

$$(P_{\Omega}) \quad \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega)$$

Hierbei beschreibt \mathcal{O}_{ad} eine Menge von zulässigen Gebieten. An dieser Stelle sei \mathcal{O}_{ad} wie folgt gegeben :

$$\mathcal{O}_{ad} = \{\Omega \subset D, |\Omega| = \mathcal{V}\}$$

$D \subset \mathbb{R}^3$ ist ein beschränktes Gebiet, welches wir als Konstruktionsgebiet bezeichnen und \mathcal{V} sei ein fixiertes Volumen.

Da P_u für alle Gebiete $\Omega \in \mathcal{O}_{ad}$ lösbar sein muss, müssen die Größen f und g auf dem ganzen Konstruktionsgebiet D gegeben sein. Wir wählen aus diesem Grund $f \in L^2(D)^d$ und $g \in H^{1,2}(D)^d$.

Der Beweis der Existenz einer Lösung für Probleme solcher Art ist nichttrivial und soll in dieser Arbeit nicht weiter behandelt werden. Es soll jedoch der Vollständigkeit halber ein wohldefiniertes Minimierungsproblem formuliert werden, welches unsere weiteren Betrachtungen motiviert.

Obiges Minimierungsproblem ist so noch nicht wohlgestellt und es existiert kein globales Minimum. Um die Existenz von Minima zu gewährleisten ist eine Modifikation von $J(\Omega)$ nötig. Sei $\mathcal{P}(\Omega) = \mathcal{H}(\partial\Omega)$ das $(d-1)$ -dimensionale Hausdorffmaß des Randes von Ω . Setzt man

$$J'(\Omega) = J(\Omega) + l \mathcal{P}(\Omega)$$

mit $l \in \mathbb{R}$, $l > 0$ einem positiven Lagrangemultiplikator, dann besitzt das Minimierungsproblem

$$(P'_{\Omega}) \quad \min_{\Omega \in \mathcal{O}_{ad}} J'(\Omega)$$

mindestens eine optimale Lösung $\Omega_{opt} \in \mathcal{O}_{ad}$. Wir verweisen auf die Arbeit von Ambrosio und Buttazzo [5]. Der Term $l \mathcal{P}(\Omega)$ unterdrückt die Bildung von Mikrostrukturen, welche bei dieser Art von Problemen typisch ist.

Aus verschiedenen Gründen können wir von vornherein nicht erwarten das Shapeoptimierungsproblem exakt zu lösen, d.h. die globale Lösung Ω_{opt} zu finden. Zum einen betrachten wir nur eine numerische Approximation des Problems mit beschränkter geometrischer

Auflösung der Strukturen. Zum anderen wollen wir das Shape-Optimierungsproblem mit einem Gradientenverfahren lösen. Dieses ist sehr anfällig dafür lokale Minima statt globaler zu finden. Deshalb schränken wir unsere Betrachtungen auf folgende Restriktionsmenge ein :

$$\mathcal{O}_{ad} = \{\Omega \subset D, \partial\Omega \text{ Lipschitz-stetig}\}$$

Zusätzlich stellen wir Bedingungen an den Rand des Konstruktionsgebiets D

$$\partial D = \partial D_D \cup \partial D_N$$

Dabei seien Dirichlerandwerte auf ∂D_D und Neumannrandwerte auf ∂D_N gegeben. Weiter sei ∂D_0 eine Teimenge von ∂D mit $\mathcal{H}(\partial\Omega) \neq 0$, die einen fixierten Rand beschreibt. Für die Gebiete aus der Menge \mathcal{O}_{ad} gelten dann die Bedingungen

$$\partial\Omega \cap \partial D_D \neq \emptyset \quad (2.3)$$

$$\partial\Omega \cap \partial D_0 \neq \emptyset \quad (2.4)$$

Damit lautet unser Optimierungsproblem

$$(P_\Omega) \quad \min_{\Omega \in \mathcal{O}_{ad}} J(\Omega)$$

$$\mathcal{O}_{ad} = \{\Omega \subset D : \partial\Omega \text{ Lipschitz-stetig und erfüllt (2.3) und (2.4)}\}$$

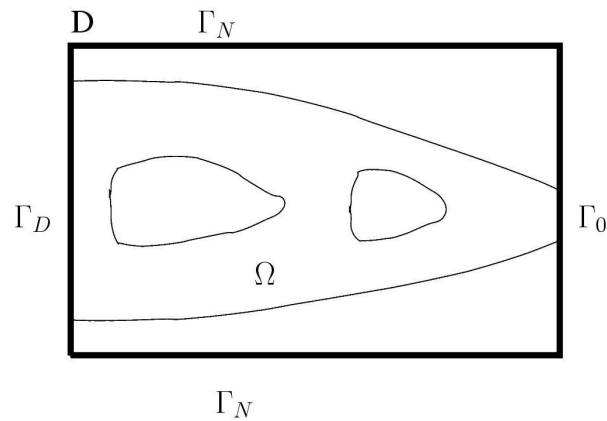


Abbildung 2.1: Konstruktionsgebiet D und Konfiguration der Randbedingungen in 2D

3 Shape Sensitivity Analysis

Die *Shape Sensitivity Analysis* bildet die theoretische Grundlage unseres Optimierungsverfahrens, deshalb wollen wir dieses Kapitel einer kurzen Einführung in dieses Gebiet widmen. Wir werden die für uns wesentlichen Resultate zusammentragen. Eine ausführliche Diskussion über den Inhalt dieses Kapitels findet sich in dem Buch von Zolesio und Delfour [24].

Die Sensitive Analysis führt auf der Basis von Transformationen des Gebietes Ω gemäß der 'Velocity-Method', sogenannte Shapeableitungen ein, welche es uns dann erlaubt Transformationen als Abstiegsrichtungen zu identifizieren. Das heißt, wir finden Transformationen $T(\Omega) = \tilde{\Omega} \subset D$ für die gilt :

$$J(\tilde{\Omega}) < J(\Omega)$$

Somit bietet sich also die Möglichkeit an ein Gradientenverfahren anzuwenden, um unser Ausgangsproblem (P_Ω) zu lösen. Leider ist ein Gradientenverfahren in der Hinsicht anfällig, lokale statt globale Minima zu finden und darum müssen wir die Tatsache akzeptieren auf diese Art der Optimierung nur lokale Minima zu erhalten. Wir werden später sehen wie sich die Wahl eines Initialgebietes auf das von unserem Optimierungsalgorithmus gefundene 'Optimum' auswirkt.

3.1 Shapefunktionen und Shapeableitungen

In diesem Abschnitt wollen wir die Definitionen der Semi-Differenzierbarkeit von Shapefunktionen, anhand der Definitionen von Gâteaux- und Hadamard-Semi-ableitungen in topologischen Vektorräumen, motivieren. Wir werden hierzu den von Vektorfeldern erzeugten Fluss, und die dadurch erzeugte Transformation eines Gebietes, als natürliches Gerüst übernehmen, um Semi-ableitungen für Shapefunktionen zu definieren. Es gibt zwei Varianten Transformationen durch Vektorfelder zu erzeugen, nämlich einmal die Methode der 'Störung der Identität' und die 'Velocity Method'. Die erste Variante, die Methode der 'Störung der Identität', wird dabei der Gâteaux-Semi-ableitung entsprechen während die 'Velocity Method' der Hadamard-Semi-ableitung entspricht.

Zunächst rufen wir uns noch einmal die Definition von Gâteaux- und Hadamard-Semi-ableitungen in topologischen Vektorräumen ins Gedächtnis.

Definition 3.1 (Semi-ableitungen)

Sei f eine reelwertige Funktion, definiert in einer Umgebung U eines Punktes x_0 aus einem topologischen Vektorraum E .

- (i) Wir sagen, dass f eine Gâteaux-Semi-ableitung im Punkt $x \in U$ in Richtung $v \in E$ besitzt, falls folgender Limes existiert :

$$\lim_{\varepsilon \searrow 0} \frac{f(x + \varepsilon v) - f(x)}{\varepsilon}$$

wenn der Grenzwert existiert, dann bezeichnen wir ihn mit $df(x;v)$.

- (ii) Wir sagen, dass f eine Hadamard-Semiablenitung im Punkt $x \in U$ in Richtung $v \in E$ besitzt, falls folgender Limes existiert :

$$\lim_{\varepsilon \searrow 0} \lim_{w \rightarrow v} \frac{f(x + \varepsilon w) - f(x)}{\varepsilon}$$

wenn dieser Grenzwert existiert, dann bezeichnen wir ihn mit $d_H(x;v)$.

Es ist klar, dass $df(x;v)$ existiert und $df(x;v) = d_H(x;v)$ gilt, falls die Semiablenitung $d_H(x;v)$ existiert.

Für Shapefunktionen wollen wir äquivalente Semiablenitungen definieren. Aber zunächst einmal geben wir unsere Definition einer Shapefunktion an.

Definition 3.2 (Shapefunktion)

Sei $D \neq \emptyset$ eine nichtleere Untermenge von \mathbb{R}^d und bezeichne $\mathcal{K}(\Omega) = \{\Omega : \Omega \subset D\}$ die Menge aller Untermengen von D . D ist unser Konstruktionsgebiet. Eine Shapefunktion ist nun eine Abbildung

$$J : \mathcal{A} \rightarrow E$$

von einer Familie \mathcal{A} von zulässigen Gebieten aus $\mathcal{K}(\Omega)$ in einen topologischen Raum E , so dass für jeden Homeomorphismus T von $\bar{\Omega}$ mit der Eigenschaft $T(\Omega) = \Omega$ für alle Elemente $\Omega \in \mathcal{A}$ gilt $J(\Omega) = J(T(\Omega))$.

Das Konstruktionsgebiet D kann, wie in unserem Fall, eine physikalische oder mechanische Einschränkung repräsentieren, oder sogar eine Untermanigfaltigkeit des \mathbb{R}^d sein. Im allgemeinen kann D so groß und der Rand von D so glatt wie nötig gewählt werden. Im unbeschränkten Fall ist D gleich dem \mathbb{R}^d .

Die Menge \mathcal{A} besitzt im allgemeinen keine Vektorraumstruktur. Trotzdem ist es möglich Differenzenquotienten und deren Grenzwerte, entlang ein-dimensionaler Richtungen um ein Element $\Omega \in \mathcal{A}$, zu betrachten. Damit ist eine Variation Ω 's mit folgender Methode gemeint.

Sei $V \in C^0(\mathbb{R}^d, \mathbb{R}^d)$ ein Vektorfeld. Dann definieren wir die folgende Abbildung :

$$T(s, x) := x + sV(x) \quad \forall x \in \mathbb{R}^d, \quad s \geq 0$$

Für s hinreichend klein ist die Abbildung

$$T(s, \cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d, \quad x \mapsto x + sV(x) \quad \forall x \in \mathbb{R}^d, \quad s \geq 0$$

ein lokaler C^1 -Diffeomorphismus, denn es gilt :

$$(\mathbb{I} + sV) \in C^1(\mathbb{R}^d, \mathbb{R}^d) \quad \text{und} \quad \lim_{s \searrow 0} \det(\mathbb{I} + sV) = 1$$

Daraus folgt, dass es für jedes beschränkte Gebiet Ω ein $s_0 > 0$ gibt, so dass $T_{s_0}(\cdot)$ Ω diffeomorph auf $\Omega_{s_0} = T_{s_0}(\Omega)$ abbildet. Dabei ist $T_s(\Omega)$ folgendermaßen definiert

$$T_s(\Omega) := \{T_s(x) : \forall x \in \Omega\}$$

wobei $T_s(x) := T(s, x) \quad \forall x \in \Omega$ gilt.

Diese Methode bezeichnen wir als Störung der Identität. Es ist uns nun möglich eine Semiableitung $dJ(\Omega; V)$ von J , an der Stelle Ω in Richtung V , als den Grenzwerte (falls dieser existiert) eines Differenzenquotienten zu definieren.

Dazu konstruieren wir uns eine Abbildung $T_s : \Omega \mapsto \Omega_s$ gemäß Wir definieren nun die Semiableitung $dJ(\Omega; V)$ wie folgt :

$$\lim_{s \searrow 0} \frac{J(T_s(\Omega)) - J(\Omega)}{s} =: dJ(\Omega; V)$$

Betrachtet man Vektorfelder V aus einem topologischen Vektorraum, so lässt sich die Stetigkeit und Linearität der Abbildung

$$V \mapsto dJ(\Omega; V) := \lim_{s \searrow 0} \frac{J(T_s(\Omega)) - J(\Omega)}{s}$$

untersuchen und die Resultate sind analog zu denen in Banachräumen. Jedoch unterliegt diese Methode insofern Einschränkungen, da Variationen von Ω nur entlang der Verbindungslinien zwischen $\mathbb{I} + V$ und der Identität \mathbb{I} auftreten. Falls D eine Untermanigfaltigkeit des \mathbb{R}^d mit nichtverschwindender Krümmung ist, ist sie generell nicht anwendbar. So wie die Hadamard-Semiableitung eine Verallgemeinerung der Gâteaux-Semiableitung darstellt, wollen wir jetzt eine allgemeinere Methode zur Variation von Gebieten einführen. Sei $V \in C^0(\mathbb{R}^d, \mathbb{R}^d)$ wieder ein Vektorfeld. Dann betrachten wir jetzt den von V erzeugten lokalen Fluss, also die Abbildung $x : \mathbb{R}^+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ mit folgenden Eigenschaften :

$$\begin{aligned} \dot{x}(s, \bar{x}) &= V(x(s, \bar{x})) & \forall s \geq 0 \\ x(0, \bar{x}) &= \bar{x} & \forall \bar{x} \in \mathbb{R}^d \end{aligned}$$

Damit können wir uns jetzt eine Abbildung $T_s : \mathbb{R}^d \rightarrow \mathbb{R}^d$ definieren mit der wir Gebiete Ω variieren.

$$T_s(\bar{x}) := x(s, \bar{x}) \quad \forall s \geq 0 \quad \forall \bar{x} \in \mathbb{R}^d$$

Hierbei ist die Geschwindigkeit des Punktes $x(s, \bar{x})$ zum Zeitpunkt s abhängig von $V(x(s, \bar{x}))$ und nicht von $V(\bar{x})$. Dies ist der Grundgedanke der 'Velocity-method'. Diese Konstruktion lässt sich leicht auf zeitabhängige Vektorfelder ausweiten.

Sei $V \in C^0(\mathbb{R}^+ \times \mathbb{R}^d, \mathbb{R}^d)$ ein zeitabhängiges Vektorfeld. Dann betrachten wir den von V erzeugten Fluss gemäß :

$$\begin{aligned} \dot{x}(s, \bar{x}) &= V(s, x(s, \bar{x})) & \forall s \geq 0 \\ x(0, \bar{x}) &= \bar{x} & \forall \bar{x} \in \mathbb{R}^d \end{aligned}$$

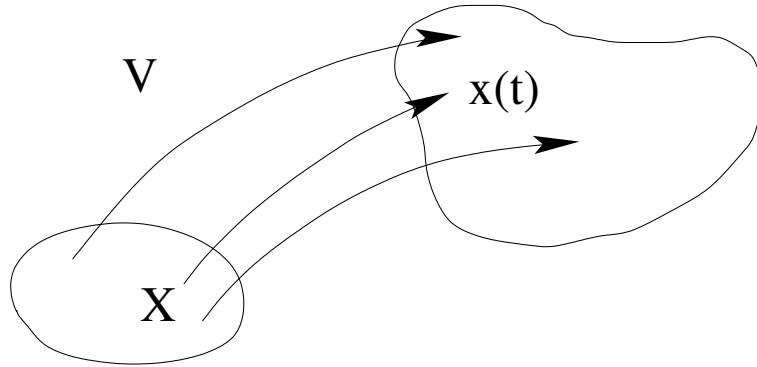


Abbildung 3.1: Transport von Ω durch das Geschwindigkeitsfeld V

Wie oben definieren wir nun die Abbildung $T : \mathbb{R}_0^+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$

$$T(s, \bar{x}) := x(s, \bar{x}) \quad \forall s \geq 0 \quad \forall \bar{x} \in \mathbb{R}^d \quad (3.1)$$

Die obige Funktion $x : \mathbb{R}_0^+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ ist die Lösung einer gewöhnlichen Differentialgleichung. Es ist leicht zu zeigen, dass die Lösung stetig von den Anfangswerten abhängt. Die Funktion $T : \mathbb{R}_0^+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ ist somit stetig.

Analog zu obiger Definition wird die Semiableitung $d_H J(\Omega; V)$ der Shapefunktion J in Richtung des zeitabhängigen Vektorfeldes V als der Grenzwert, falls dieser existiert, eines Differenzenquotienten definiert

$$d_H J(\Omega; V) := \lim_{s \searrow 0} \frac{J(T(s, \Omega)) - J(\Omega)}{s}$$

wobei $T(s, \Omega) = \{ T(s, \bar{x}) : \bar{x} \in \Omega \}$ ist.

Wir zeigen jetzt, dass die 'Velocity-method' eine Verallgemeinerung der Methode der 'Störung der Identität' ist. Somit lassen sich die Gâteaux-Semiableitung mit der 'Störung der Identität' und die Hadamard-Semiableitung mit der 'Velocity-method' in Bezug bringen.

Um diesen Vergleich zu motivieren, konstruieren wir nun ein zeitabhängiges Vektorfeld $V : \mathbb{R}_0^+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, welches wir einem konstanten Vektorfeld $\tilde{V} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ zuordnen können. Und zwar in der Weise, dass die Transformation, aus dem zeitabhängigen Vektorfeld durch die 'Velocity-method' erzeugt, mit der Transformation übereinstimmt, die durch Störung der Identität mit dem konstanten Vektorfeld \tilde{V} erzeugt wird. Betrachten wir also die Transformation

$$T(s, \bar{x}) = x(s) = \bar{x} + s\tilde{V}(\bar{x}) \quad \forall \bar{x} \in \mathbb{R}^d, \quad s \geq 0$$

Das Vektorfeld V muss so gewählt werden, dass die Funktion x die Lösung zu folgender Differentiallösung ist.

$$\begin{aligned} \dot{x}(s) &= V(s, x(s)) \quad \forall s \geq 0 \\ x(0) &= \bar{x} \end{aligned}$$

Es ist leicht zu sehen, dass die Lösung wie folgt aussieht :

$$V(s, \bar{x}) := \tilde{V}(T^{-1}(s, \bar{x})) \quad \forall \bar{x} \in \mathbb{R}^d, \quad s \geq 0$$

Unter geeigneten Voraussetzungen an Stetigkeit und Differenzierbarkeit (und s genügend klein) lässt sich feststellen :

$$\begin{aligned} V(0, x) &= \tilde{V} & \forall x \in \mathbb{R}^d \\ \frac{\partial V}{\partial s}(s, x) \Big|_{s=0} &= -[D\tilde{V}(x)]\tilde{V}(x) & \forall x \in \mathbb{R}^d \end{aligned}$$

Dabei ist $D\tilde{V}(x)$ die Jacobimatrix von \tilde{V} im Punkt x . Aus den beiden Gleichungen oben erkennt man, dass die Punkte aus Ω gleichzeitig von dem Vektorfeld $V(0, x) = \tilde{V}(x)$ und dem Beschleunigungsfeld $\dot{V}(0, x) = -[D\tilde{V}(x)]\tilde{V}(x)$ beeinflusst werden. Die beiden vorgestellten Methoden werden bei geeigneten Bedingungen (z.B. D ist keine wirkliche Untermanigfaltigkeit des \mathbb{R}^d) die gleichen Semiableitungen erster Ordnung produzieren. Semiableitungen zweiter Ordnung würden sich durch einen Beschleunigungsterm unterscheiden, welcher bei der Methode der Störung der Identität auftritt. Ableitungen zweiter Ordnung wollen wir hier jedoch nicht behandeln.

Die bisher aufgezeigten Parallelen zwischen Hadamard- und Gâteaux-Semiableitungen auf der einen Seite und den Semiableitungen, erzeugt durch die 'Velocity-method' und Störung der Identität, auf der anderen Seite, sollen nun durch folgenden Satz konkretisiert werden. Er bringt die Hadamard-Semiableitung mit der Semiableitung, erzeugt durch die 'Velocity-method', einer reelwertigen Funktion $f : \mathbb{R}^d \rightarrow \mathbb{R}$ in Beziehung.

Satz 3.3 (Hadamard-Semiableitung und 'Velocity-method')

Sei $f : N_x \rightarrow \mathbb{R}$ eine reelwertige Funktion definiert in einer Umgebung N_x eines Punktes $x \in \mathbb{R}^d$. Dann ist f Hadamard semidifferenzierbar im Punkt x in Richtung eines Vektors $v \in \mathbb{R}^d$ genau dann, wenn es ein $\tau > 0$ gibt, so dass für alle zeitabhängigen Vektorfelder $V : [0, \tau] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, die den folgenden Bedingungen genügen

$$(i) \quad V(\cdot, x) \in C([0, \tau]; \mathbb{R}^d) \quad \forall x \in \mathbb{R}^d$$

$$(ii) \quad \exists c > 0, \text{ so dass } \forall x, y \in \mathbb{R}^d \text{ gilt } \|V(\cdot, y) - V(\cdot, x)\|_{C([0, \tau]; \mathbb{R}^d)} \leq c \|y - x\|$$

(iii) der Grenzwert

$$d_H f(x, V) := \lim_{s \searrow 0} \frac{f(T(V)(s, x)) - f(x)}{s}$$

existiert, hierbei ist $T(V)(s, x) := \bar{x}(s)$ die Lösung der Differentialgleichung

$$\begin{aligned} \dot{\bar{x}}(s) &= V(s, \bar{x}(s)) & \forall s \geq 0 \\ \bar{x}(0) &= x \end{aligned}$$

gilt, falls $V(0,x) = v$ ist, dann folgt daraus

$$d_H f(x, V) := d_H f(x, \tilde{V}) \quad \text{wobei} \quad \tilde{V}(s, x) = v \quad \forall s \geq 0$$

Für den Beweis zu diesem Satz verweisen wir auf [24]. Satz 3.3 motiviert uns also die Hadamard-Semiablenitung für Shapefunktionen zu definieren. Im nächsten Abschnitt tragen wir die hier gesammelten Ergebnisse zusammen.

3.2 Erster Ordnung Semiablenitungen

Zusätzlich zu den Bedingungen im obigen Satz, ist im beschränkten Fall $D \subset \mathbb{R}^d$, noch zu gewährleisten, dass transformierte Gebiete noch in D enthalten sind. Dies erreichen wir durch die Forderung

$$\forall x \in \partial D, \forall s \in [0, \tau] \text{ gilt } V(s, x) \in TR_{\partial D}(x)$$

an die betrachteten Vektorfelder. Hierbei bezeichnet $TR_{\partial D}(x)$ den Tangentialraum von ∂D im Punkt $x \in \partial D$. Der Rand ∂D unseres Konstruktionsgebietes D muss dafür genügen glatt sein. In dem von uns betrachteten Optimierungsproblem stellt eine solche Forderung keine Einschränkung dar. Die im beschränkten Fall geforderten Bedingungen an die Vektorfelder lauten also :

$$V(., x) \in C([0, \tau]; \mathbb{R}^d) \quad \forall x \in \mathbb{R}^d \quad (3.2)$$

$$\exists c > 0, \text{ so dass } \forall x, y \in \mathbb{R}^d \text{ gilt } \|V(., y) - V(., x)\|_{C([0, \tau]; \mathbb{R}^d)} \leq c \|y - x\| \quad (3.3)$$

$$\forall x \in \partial D, \forall s \in [0, \tau] \text{ gilt } V(s, x) \in TR_{\partial D}(x) \quad (3.4)$$

Die von einem, die Bedingungen (3.2) bis (3.4) erfüllenden, Vektorfeld V erzeugte Transformation

$$T_s(V)(\Omega) := T(V)(s, \Omega) \quad (3.5)$$

bildet Gebiete $\Omega \in D$ auf Gebiete $\Omega_s(V) \in D$ ab.

Definition 3.4 (Semiablenitung und Ableitung von Shapefunktionen)

Sei \mathcal{V} ein topologischer Untervektorraum von $\text{Lip}(D, \mathbb{R}^d)$ und sei J eine reelwertige Shapefunktion.

- (i) Wenn V ein zeitabhängiges Vektorfeld ist, welches den Bedingungen (3.2) und (3.3) genügt, dann sagen wir J besitzt eine Euler-Semiablenitung bei Ω in Richtung V , falls der folgende Grenzwert existiert und endlich ist

$$dJ(\Omega; V) := \lim_{s \searrow 0} \frac{J(\Omega_s(V)) - J(\Omega)}{s}$$

(ii) Für ein $\tilde{V} \in \text{Lip}(D, \mathbb{R}^d)$ und das zeitabhängiges Vektorfeld V

$$V(s, x) = \tilde{V}(x) \quad \forall s \in [0, \tau], \quad \forall x \in D$$

benutzen wir die Notation $dJ(\Omega; V)$ oder einfach $dJ(\Omega; \tilde{V})$.

(iii) Sei $\tilde{V} \in \mathcal{V}$, dann sagen wir, J hat eine Hadamard Semiableitung bei Ω in Richtung \tilde{V} bezüglich \mathcal{V} , falls für alle zeitabhängigen Vektorfelder V , für die erstens (3.2) und (3.3), zweitens $V(s, \cdot) \in \mathcal{V}$ und drittens $V(0, x) = \tilde{V}(x) =: V(0)$ gilt, der Grenzwert $dJ(\Omega; V)$ existiert und nur von $V(0)$ abhängt.

In diesem Fall wird die Semiableitung mit $d_H J(\Omega; \tilde{V})$ bezeichnet und natürlich gilt

$$d_H J(\Omega; V(0)) = dJ(\Omega; V(0))$$

(iv) J heißt differentierbar bei Ω in \mathcal{V} , falls J in jede Richtung $\tilde{V} \in \mathcal{V}$ eine Euler-Semiableitung besitzt, und die Abbildung

$$dJ(\Omega; \cdot) : \mathcal{V} \rightarrow \mathbb{R} \tag{3.6}$$

linear und stetig, d. h. ein Element aus dem Dualraum \mathcal{V}' von \mathcal{V} ist.

Die Definition der Euler-Semiableitung ist sehr allgemein. Zum Beispiel ist sie leicht erweiterbar auf Shapefunktionen, die auf Untermanigfaltigkeiten des \mathbb{R}^d definiert sind. Sie schließt Fälle ein, in denen $dJ(\Omega; V)$ nicht nur von $V(0)$ abhängen kann, sondern auch von $V(s)$ in einer kleinen Umgebung von $s = 0$. Wenn $dJ(\Omega; V)$ nur von $V(0)$ abhängt, dann kann die Analysis auf stationäre Vektorfelder beschränkt werden, da dann die Hadamard-Semiableitungen existieren. Die von uns betrachtete Shapefunktion liegt in einer Klasse für die dieses gilt. Wir verweisen dazu auf Kapitel 8 in [24].

Zunächst ein einfaches Beispiel für eine Shapeableitung.

Beispiel 3.5

Sei $V \in C_0^\infty(\mathbb{R}^d, \mathbb{R}^d)$ und sei der Rand von Ω von der Klasse C^1 . Die Shapefunktion J sei wie folgt definiert :

$$J(\Omega) = \int_{\Omega} dx$$

Es ist $\Omega_s = T_s(V)(\Omega)$, wobei $T_s(V)(\cdot)$ Ω diffeomorph auf Ω_s abbildet. Dann gilt :

$$J(\Omega_s) = \int_{\Omega_s} dx = \int_{\Omega} |\det DT_s(V)(x)| dx$$

$DT_s(V)(x)$ ist die Jacobi-Matrix von $T_s(V)(\cdot)$ bezüglich x . Dann gilt weiter

$$\begin{aligned} dJ(\Omega; V) &= \left. \frac{\partial J(\Omega_s)}{\partial s} \right|_{s=0} = \frac{\partial}{\partial s} \int_{\Omega} |\det DT_s(V)(x)| dx \Big|_{s=0} \\ &= \int_{\Omega} \left. \frac{\partial |\det DT_s(V)(x)|}{\partial s} \right|_{s=0} dx \end{aligned}$$

Betrachten wir die Taylorentwicklung bezüglich s der Abbildung $T_s(V)(\cdot)$:

$$\begin{aligned} T_s(V)(x) &= T_0(V)(x) + \left. \frac{\partial}{\partial s} T_s(V)(x) \right|_{s=0} s + o(s^2) \\ &= \mathbb{I} + V(x) s + o(s^2) \end{aligned}$$

Dann gilt für die Jacobi-Matrix von $T_s(V)(x)$:

$$DT_s(V)(x) = \mathbb{I} + DV(x) s + o(s^2)$$

Einsetzen in die Formel für die Shapeableitung ergibt:

$$\begin{aligned} dJ(\Omega_s) \Big|_{s=0} &= \int_{\Omega} \left. \frac{\partial}{\partial s} \det DT_s(V)(x) \right|_{s=0} dx \\ &= \int_{\Omega} \left. \frac{\partial}{\partial s} \det(\mathbb{I} + DV(x) s + o(s^2)) \right|_{s=0} dx \\ &= \int_{\Omega} \text{tr}(DV(x)) dx \\ &= \int_{\partial\Omega} V(x) \cdot \vec{n} d\nu \end{aligned}$$

Man sieht, dass nur der Normalenanteil von V auf dem Rand von Ω in die Shapeableitung eingeht.

3.3 Struktursatz

Angelehnt an die Diskussion am Ende des vorherigen Abschnitts, beschränken wir uns jetzt auf stationäre Vektorfelder $V \in \mathcal{V}$, wobei \mathcal{V} ein topologischer Untervektorraum von $\text{Lip}(\mathbb{R}^d, \mathbb{R}^d)$ sei. Der Einfachheit halber betrachten wir an dieser Stelle den unbeschränkten Fall $D = \mathbb{R}^d$. Der beschränkte Fall liefert ähnliche Resultate, ist aber technisch aufwendiger. Für Details verweisen wir auf [23].

Die Shapeableitung hängt natürlich wesentlich von der Wahl des topologischen Untervektorraumes von $\text{Lip}(\mathbb{R}^d, \mathbb{R}^d)$ ab. Der Anschaulichkeit halber arbeiten wir mit glatten Vektorfeldern. Sei also $\mathcal{V} = \mathcal{D}(\mathbb{R}^d, \mathbb{R}^d)$ der Raum der unendlich oft differenzierbaren Funktionen mit kompaktem Träger. Damit sind die Bedingungen (3.2) und (3.3) für alle $V \in \mathcal{V}$ erfüllt.

Definition 3.6

Sei J eine reelwertige Shapefunktion und sei $\Omega \subset \mathbb{R}^d$ eine Untermenge des \mathbb{R}^d .

- (i) Wir sagen, die Shapefunktion J ist shapedifferentierbar bei Ω , falls sie bei Ω in alle Richtungen $V \in \mathcal{V}$ differentierbar ist.
- (ii) Die Abbildung in (3.6) definiert eine Vektordistribution $\frac{dJ}{d\Omega}$ in $\mathcal{D}'(\mathbb{R}^d, \mathbb{R}^d)$, für welche wir folgende Notation einführen :

$$\langle \frac{\partial J}{\partial \Omega}, V \rangle := dJ(\Omega; V)$$

Hierbei bedeutet $\langle \frac{\partial J}{\partial \Omega}, \cdot \rangle$ die Anwendung von $\frac{\partial J}{\partial \Omega}$ auf ein Element $V \in \mathcal{D}(\mathbb{R}^d, \mathbb{R}^d)$.

Im Weiteren wollen wir die Eigenschaften von $\langle \frac{\partial J}{\partial \Omega}, \cdot \rangle$ untersuchen. Hierzu dient folgendes vorbereitende Lemma.

Lemma 3.7

Die in (3.1) definierte Abbildung $T(s, x)$ ist für ein $V \in \mathcal{V} = \mathcal{D}(\mathbb{R}^d, \mathbb{R}^d)$ ein C^1 -Diffeomorphismus.

Beweis

Zunächst ist das zu einem $V \in \mathcal{V}$ gehörige dynamische System F autonom, denn es gilt $F(s, x) = F(x) := V(x)$. Desweiteren ist F linear beschränkt, denn es gilt

$$\|F(s, x)\| \leq L \|x\| + \|F(0)\|$$

wobei L die Lipschitz-Konstante von V ist. Somit ist für jedes Intervall $[0, a]$, $a > 0$ die Abbildung $T(s, x)$ mit $s \in [0, a]$ definiert, da die einzelnen Integralkurven auf ganz $[0, a]$ definiert sind. Nach dem Satz von der Diffeomorphie der Zustandsabbildung ist $T_s(V)(x) = T(s, x)$ $s \in [0, a]$, $T_s(V)(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ ein Diffeomorphismus.

Wir verweisen für Details auf [28]. □

Der folgende Satz gibt Aufschluss über die Gestalt von $\langle \frac{\partial J}{\partial \Omega}, \cdot \rangle$.

Satz 3.8 (Struktursatz)

Sei J eine reelwertige Shapefunktion. J sei shapedifferentierbar für ein Untermenge Ω des \mathbb{R}^d .

Dann ist der Träger von $\langle \frac{\partial J}{\partial \Omega}, \cdot \rangle$ in $\partial\Omega$ enthalten.

Beweis

Sei $V \in C_0^\infty(\mathbb{R}^d, \mathbb{R}^d)$ mit $V = 0$ auf $\partial\Omega$. Zu zeigen ist, dass $T_s(V)(\Omega) = \Omega$ ist.

Sei $x_0 \in \partial\Omega$. Für die Trajektorie $\bar{x}(s, x_0)$ von x_0 gilt:

$$\begin{aligned} \dot{\bar{x}}(s) &= V(\bar{x}(s)) \quad \forall s \geq 0 \\ \bar{x}(0) &= x_0 \\ \dot{\bar{x}}(0) &= V(\bar{x}(0)) = 0 \end{aligned}$$

Es ist leicht zu sehen, dass die eindeutig definierte Lösung die Form $\bar{x}(s, x_0) = x_0$ hat. Das heisst also, der Rand $\partial\Omega$ von Ω wird auf sich selbst abgebildet.

$$T_s(V)(\partial\Omega) = \partial\Omega$$

Der Rand von Ω teilt den \mathbb{R}^d in zwei disjunkte Zusammenhangskomponenten. Nämlich in $\text{int}(\Omega)$ und $\mathbb{R}^d \setminus \bar{\Omega}$. Da $T_s(V)(\cdot)$ als Diffeomorphismus stetig ist, gilt dies auch für die Bilder der beiden Zusammenhangskomponenten. Da stetige Abbildungen beschränkte Mengen auf beschränkte Mengen abbilden, muss $T_s(V)(\text{int}(\Omega)) = \text{int}(\Omega)$ gelten. Somit gilt also

$$\begin{aligned} T_s(V)(\Omega) = \Omega \quad \forall s \geq 0 &\quad \Rightarrow \quad J(T_s(V)(\Omega)) = J(\Omega) \quad \forall s \geq 0 \\ \Rightarrow \quad dJ(\Omega; V) &= 0 \end{aligned}$$

Es gilt also $\langle \frac{\partial J}{\partial \Omega}, V \rangle = 0$ für alle $V \in \mathcal{V}$ mit $V|_{\partial\Omega} = 0$.

Bemerkung 3.9

Die Aussage des obigen Satzes lässt sich auf Vektorfelder $V \in C_0^\infty(\mathbb{R}^d, \mathbb{R}^d)$ ohne Normalenanteil übertragen.

$$V \cdot \vec{n} = 0 \quad \text{auf } \partial\Omega \quad \Rightarrow \quad dJ(\Omega; V) = 0$$

Man kann zeigen, dass der Rand von Ω auf sich selbst abgebildet wird. Dann folgt aus obigen Argumenten die Behauptung.

3.4 Shapeableitung für eine Klasse von Shapefunktionen

In diesem Abschnitt wollen wir die Shapeableitung für die von uns betrachtete Shapefunktion berechnen. Die folgenden Lemmata bereiten die Berechnung der Shapeableitung der Shapefunktion aus (2.2) vor.

Lemma 3.10

Sei $V \in \mathcal{D}(\mathbb{R}^d, \mathbb{R}^d)$ und der Rand von Ω sei von der Klasse C^1 . Zusätzlich sei eine Funktion $\vartheta \in C^1(\mathbb{R}^d, \mathbb{R}^d)$ gegeben. Die Shapefunktion

$$J(\Omega) = \int_{\Omega} \vartheta(x) dx$$

ist differenzierbar bei Ω und es gilt:

$$\langle \frac{\partial J}{\partial \Omega}, V \rangle = \int_{\Omega} \text{div}(V(x)\vartheta(x)) dx = \int_{\partial\Omega} V(x) \cdot \vec{n}(x)\vartheta(x) d\nu$$

Beweis

Die Abbildung $T_s(\mathbf{V})(\cdot)$ ist ein C^1 -Diffeomorphismus und bildet Ω diffeomorph auf $\Omega_s := \{T_s(v)(x) : x \in \mathbb{R}^d\}$ ab.

$$J(\Omega_s) = \int_{\Omega_s} \vartheta(x) dx = \int_{\Omega} \vartheta(T_s^{-1}(V)(x)) |\det DT_s(V)(x)| dx$$

$DT_s(V)(x)$ ist die Jacobi-Matrix von $T_s(\mathbf{V})(\cdot)$ bezüglich x . Dann gilt weiter

$$\begin{aligned} \left\langle \frac{\partial J}{\partial \Omega}, V \right\rangle &= \left. \frac{\partial J(\Omega_s)}{\partial s} \right|_{s=0} = \left. \frac{\partial}{\partial s} \int_{\Omega} \vartheta(T_s(V)(x)) |\det DT_s(V)(x)| dx \right|_{s=0} \\ &= \int_{\Omega} \left. \frac{\partial}{\partial s} (\vartheta(T_s(V)(x)) |\det DT_s(V)(x)|) \right|_{s=0} dx \\ &= \int_{\Omega} \left. \frac{\partial}{\partial s} (\vartheta(T_s(V)(x))) \right|_{s=0} |\det DT_0(V)(x)| \\ &\quad + \vartheta(T_0(V)(x)) \left. \frac{\partial}{\partial s} (|\det DT_s(V)(x)|) \right|_{s=0} dx \\ &= \int_{\Omega} \left. \frac{\partial}{\partial s} (\vartheta(T_s(V)(x))) \right|_{s=0} \\ &\quad + \vartheta(x) \left. \frac{\partial}{\partial s} (|\det DT_s(V)(x)|) \right|_{s=0} dx \end{aligned}$$

Für die Taylorentwicklung der Abbildung $T_s(\mathbf{V})(\cdot)$ bezüglich s gilt :

$$\begin{aligned} T_s(V)(x) &= T_0(V)(x) + \left. \frac{\partial}{\partial s} T_s(V)(x) \right|_{s=0} s + o(s^2) \\ &= \mathbb{I} + V(x) s + \sigma(s^2) \end{aligned}$$

Für die Jacobi-Matrix von $T_s(\mathbf{V})(x)$ gilt dann:

$$DT_s(V)(x) = \mathbb{I} + DV(x) s + \sigma(s^2)$$

Dann gilt für den zweiten Term im Integral:

$$\left. \frac{\partial}{\partial s} (\det DT_s(V)(x)) \right|_{s=0} = \text{tr}(DV(x)) = \text{div}(V(x))$$

Für die Shapeableitung gilt dann :

$$\begin{aligned}
\left\langle \frac{\partial J}{\partial \Omega}, \cdot \right\rangle &= \int_{\Omega} \frac{\partial}{\partial s} (\vartheta(T_s(V)(x))) \Big|_{s=0} + \vartheta(x) \operatorname{div}(V(x)) dx \\
&= \int_{\Omega} \nabla \vartheta(x) \frac{\partial}{\partial s} (T_s(V)(x)) \Big|_{s=0} + \vartheta(x) \operatorname{div}(V(x)) dx \\
&= \int_{\Omega} \nabla \vartheta(x) V(x) + \vartheta(x) \operatorname{div}(V(x)) dx \\
&= \int_{\Omega} \operatorname{div}(V(x)\vartheta(x)) dx \\
&= \int_{\partial\Omega} V(x) \cdot \vec{n}(x) \vartheta(x) d\nu
\end{aligned}$$

□

Nun folgt ein Lemma, welches später dazu dienen wird Neumannrandwerte in die Shapefunktion einzubauen.

Lemma 3.11

Sei wieder $V \in \mathcal{D}(\mathbb{R}^d, \mathbb{R}^d)$ und der Rand von Ω sei von der Klasse C^1 . Zusätzlich sei eine Funktion $\vartheta \in C^2(\mathbb{R}^d, \mathbb{R}^d)$ gegeben. Die Shapefunktion

$$J(\Omega) = \int_{\partial\Omega} \vartheta(x) dx$$

ist differenzierbar bei Ω und es gilt:

$$\left\langle \frac{\partial J}{\partial \Omega}, V \right\rangle = \int_{\partial\Omega} V(x) \cdot \vec{n}(x) \left(\frac{\partial \vartheta}{\partial n}(x) + h(x)\vartheta(x) \right) d\nu$$

Hierbei ist $h(x)$ die mittlere Krümmung der Fläche $\partial\Omega$. Den Beweis dieses Lemmas wollen wir an dieser Stelle nicht herleiten.

Bemerkung 3.12 Aus den Lemmata (3.10) und (3.11) lassen sich leicht Terme zur Volumen- oder Oberflächenbeschränkung herleiten. So gilt

$$O(\Omega) = \int_{\Omega} dx \quad \Rightarrow \quad dO(\Omega; V) = \int_{\partial\Omega} V(x) \cdot n(x) d\nu$$

Zur Oberflächenbeschränkung benutzt man:

$$P(\Omega) = \int_{\partial\Omega} dx \quad \Rightarrow \quad dP(\Omega; V) = \int_{\partial\Omega} V(x) \cdot n(x) h(x) d\nu$$

Nun fügen wir noch zwei Lemmata ein, die uns das Rechnen mit dem symmetrischen Tensor \mathcal{C} erleichtern werden.

Lemma 3.13

Für den symmetrischen Tensor \mathcal{C} aus Kapitel 2 gilt mit hinreichender Differenzierbarkeit folgende Identität:

$$\int_{\Omega} \mathcal{C} : \epsilon(\varphi) dx = \int_{\Omega} \operatorname{div}(\mathcal{C}) \cdot \varphi dx + \int_{\partial\Omega} \mathcal{C} \cdot n \cdot \varphi d\nu$$

Beweis

Im Folgenden benutzen wir die Einsteinsche Summenkonvention.

$$\begin{aligned} \int_{\Omega} \mathcal{C} : \epsilon(\varphi) dx &= \frac{1}{2} \int_{\Omega} \mathcal{C}_{ij}(\varphi_{ij} + \varphi_{ji}) dx \\ &= -\frac{1}{2} \int_{\Omega} \mathcal{C}_{ij,j} \varphi_i + \mathcal{C}_{ij,i} \varphi_j dx + \frac{1}{2} \int_{\partial\Omega} \mathcal{C}_{ij}(\varphi_i n_j + \varphi_j n_i) d\nu \\ &= - \int_{\Omega} \operatorname{div}(\mathcal{C}) \cdot \varphi dx + \int_{\partial\Omega} \varphi \cdot \mathcal{C} \cdot n d\nu \end{aligned}$$

□

Lemma 3.14

Für den symmetrischen Tensor

$$\mathcal{C}(\epsilon(u)) = \lambda \operatorname{tr}(\epsilon(u)) \mathbb{I} + 2\mu \epsilon(u)$$

gilt, wegen der Symmetrie von $\epsilon(u)$, folgende Identität:

$$\mathcal{C}(\epsilon(u)) : \epsilon(\varphi) = \mathcal{C}(\epsilon(\varphi)) : \epsilon(u)$$

Beweis

$$\begin{aligned} \mathcal{C}(\epsilon(u)) : \epsilon(\varphi) &= \lambda \operatorname{tr}(\epsilon(u)) \mathbb{I} : \epsilon(\varphi) + 2\mu \epsilon(u) : \epsilon(\varphi) \\ &= \lambda \operatorname{tr}(\epsilon(u)) \operatorname{tr}(\epsilon(\varphi)) + 2\mu \operatorname{tr}(\epsilon(u) \cdot \epsilon(\varphi)) \\ &= \lambda \operatorname{tr}(\epsilon(\varphi)) \mathbb{I} : \epsilon(u) + 2\mu \epsilon(\varphi) : \epsilon(u) \\ &= \mathcal{C}(\epsilon(\varphi)) : \epsilon(u) \end{aligned}$$

□

Nun wollen wir die Shapeableitung der Shapefunktion aus (2.2) angeben.

Satz 3.15

Sei $\Omega \in \mathbb{R}^3$ ein glatt berandetes Gebiet und $V \in \mathcal{D}(\mathbb{R}^3, \mathbb{R}^3)$. Für die Funktionen f und g fordern wir $f \in H^{1,2}(\Omega)^3$ und $g \in H^{2,2}(\Omega)^3$. Zusätzlich fordern wir für die Lösung u von (\mathcal{P}_u) $u \in H^{2,2}(\Omega)^3$. Dann ist die Shapeableitung der Shapefunktion J aus (2.2) gegeben durch :

$$\begin{aligned} \left\langle \frac{\partial J}{\partial \Omega}, V \right\rangle &= \int_{\Gamma_N} \left(2 \left[\frac{\partial(g \cdot u)}{\partial n} + hg \cdot u + f \cdot u \right] - \mathcal{C}\epsilon(u) : \epsilon(u) \right) V \cdot \vec{n} \, d\nu \\ &\quad + \int_{\Gamma_D} (\mathcal{C}\epsilon(u) : \epsilon(u)) V \cdot \vec{n} \, d\nu \end{aligned}$$

Beweis

Wir folgen dem Beweis aus [3] und greifen die Notation auf. Sei eine allgemeine Shapefunktion J gegeben

$$J(\Omega) = \int_{\Omega} j(x, u(x)) \, dx + \int_{\Gamma_N} l(x, u(x)) \, d\nu$$

mit differenzierbaren Funktionen $j : \mathbb{R} \times \mathbb{R}^3 \rightarrow \mathbb{R}$ und $l : \mathbb{R} \times \mathbb{R}^3 \rightarrow \mathbb{R}$. Die folgende Vorgehensweise ist bekannt aus der Kontrolltheorie. Das partielle Differentialgleichungssystem wird als Nebenbedingung in einem Minimierungsproblem berücksichtigt. Man konstruiert eine Lagrangefunktion von der Gestalt

$$\mathcal{L}(\Omega, \varphi, \psi) = F(\Omega, \varphi) + dE(\Omega, \varphi; \psi) \quad \varphi, \psi \in H_D^1(\Omega)$$

dabei ist F wie folgt definiert

$$F(\Omega, \varphi) = \int_{\Omega} j(x, u(x)) \, dx + \int_{\Gamma_N} l(x, u(x)) \, d\nu$$

und $dE(\Omega, \varphi; \psi)$ beschreibt die Variation der zu (\mathcal{P}_u) gehörenden elastischen Energie aus (2.1) in Richtung ψ an der Stelle φ . Diese Methode ist hier nicht direkt anwendbar, da der betrachtete Funktionenraum $H_D^1(\Omega)$ von Ω abhängt.

Um diese Abhängigkeit zu umgehen, wird die Lagrangefunktion etwas modifiziert. Dies

geschieht durch addieren eines vom Problem (\mathcal{P}_u) abhängigen Terms.

$$\begin{aligned} \mathcal{L}(\Omega, \varphi, \psi) &= \int_{\Omega} j(x, u(x)) dx + \int_{\Gamma_N} l(x, u(x)) d\nu \\ &+ \underbrace{\int_{\Omega} \mathcal{C}\epsilon(\varphi) : \epsilon(\psi) - \psi \cdot f dx - \int_{\Gamma_N} \psi \cdot g d\nu}_{dE(\Omega, \varphi; \psi)} \\ &- \underbrace{\int_{\Gamma_D} \psi \cdot \mathcal{C}\epsilon(\varphi) \cdot \vec{n} + \varphi \cdot \mathcal{C}\epsilon(\psi) \cdot \vec{n} d\nu}_{\text{Zusatzterm}} \quad \varphi, \psi \in H^{1,2}(\mathbb{R}^3; \mathbb{R}^3) \end{aligned}$$

Betrachten wir den stationären Punkt von $\mathcal{L}(\Omega, \varphi, \psi)$, den wir mit (u, p) bezeichnen, dann ist u die Lösung zum Problem (\mathcal{P}_u) und p die Lösung zum adjungierten Problem (\mathcal{P}_p) . Denn für den stationären Punkt (u, p) gilt einmal

$$\begin{aligned} \left\langle \frac{\partial \mathcal{L}}{\partial \psi}(\Omega, u, p), \phi \right\rangle &= \int_{\Omega} \phi (\operatorname{div}(\mathcal{C}\epsilon(u)) + f) dx \\ &+ \int_{\Gamma_N} \phi ((\mathcal{C}\epsilon(u)) \cdot \vec{n} - g) d\nu - \int_{\Gamma_D} u \cdot (\mathcal{C}\epsilon(\phi)) \cdot \vec{n} d\nu \end{aligned}$$

Hierbei wurde Lemma (3.13) auf den Term $\mathcal{C}\epsilon(u) : \epsilon(\phi)$ angewendet. Wählt man zuerst $\phi \in C_0^\infty(\Omega)$, dann folgt mit dem Fundamentallema der Variationsrechnung

$$-\operatorname{div}(\mathcal{C}\epsilon(u)) = f \quad \text{in } \Omega$$

wählt man dann $\phi \in H_D^1(\Omega)$ so folgt

$$\mathcal{C}\epsilon(u) \cdot \vec{n} = g \quad \text{auf } \Gamma_N$$

und abschließend variiert man $\mathcal{C}\epsilon(\phi) \cdot \vec{n}$ auf Γ_D und erhält die Dirichletrandbedingung $u = 0$ auf Γ_D . Somit ist u die Lösung von (\mathcal{P}_u) .

Auf der anderen Seite gilt

$$\begin{aligned} \left\langle \frac{\partial \mathcal{L}}{\partial \varphi}(\Omega, u, p), \phi \right\rangle &= \int_{\Omega} \frac{\partial j}{\partial y}(x, u(x)) \cdot \phi dx + \int_{\partial \Omega} \frac{\partial l}{\partial y}(x, u(x)) \cdot \phi d\nu \\ &+ \int_{\Omega} \mathcal{C}\epsilon(\phi) : \epsilon(p) dx \\ &- \int_{\Gamma_D} (p \cdot \mathcal{C}\epsilon(\phi) \cdot \vec{n} + \phi \cdot \mathcal{C}\epsilon(p) \cdot \vec{n}) d\nu \end{aligned}$$

Es gilt mit Verwendung von Lemma (3.13) und (3.14)

$$\begin{aligned} \int_{\Omega} \mathcal{C}\epsilon(\phi) : \epsilon(p) \, dx &= \int_{\Omega} \mathcal{C}\epsilon(p) : \epsilon(\phi) \, dx \\ &= - \int_{\Omega} \operatorname{div}(\mathcal{C}\epsilon(p)) \cdot \phi \, dx + \int_{\partial\Omega} \phi \cdot \mathcal{C}\epsilon(p) \cdot \vec{n} \, d\nu \end{aligned}$$

Damit ergibt sich

$$\begin{aligned} \left\langle \frac{\partial \mathcal{L}}{\partial \varphi}(\Omega, u, p), \phi \right\rangle &= \int_{\Omega} \frac{\partial j}{\partial y}(x, u(x)) \cdot \phi - \operatorname{div}(\mathcal{C}\epsilon(p)) \cdot \phi \, dx \\ &\quad + \int_{\Gamma_N} \phi \cdot \mathcal{C}\epsilon(p) \cdot \vec{n} + \frac{\partial l}{\partial y}(x, u(x)) \cdot \phi \, d\nu \\ &\quad - \int_{\Gamma_D} p \cdot \mathcal{C}\epsilon(\phi) \cdot \vec{n} \, d\nu \end{aligned}$$

Mit der gleichen Argumentation wie oben erhalten wir

$$(P_p) \quad \begin{cases} -\operatorname{div}(\mathcal{C}(\epsilon(p))) &= -\frac{\partial j}{\partial y}(x, u(x)) & \text{in } \Omega \\ p &= 0 & \text{on } \Gamma_D \\ \mathcal{C}(\epsilon(p)) \cdot n &= -\frac{\partial l}{\partial y}(x, u(x)) & \text{on } \Gamma_N \end{cases}$$

Dies ist ein wohldefiniertes Randwertproblem für den adjungierten Zustand p . Weiter gilt für die Shapefunktion $J(\Omega)$

$$J(\Omega) = \mathcal{L}(\Omega, u, p)$$

da die Bedingungen $dE(\Omega, u; p) = 0$ (u ist Minimum der Energie E) und $u = p = 0$ auf Γ_D gelten. Die Shapeableitung der Shapefunktion $J(\Omega)$ in Richtung eines Vektorfeldes $V \in \mathcal{D}(\mathbb{R}^3, \mathbb{R}^3)$ ist nun gegeben in der Form

$$\left\langle \frac{\partial J}{\partial \Omega}, V \right\rangle = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u, p)$$

Die Funktionen u und p sind dabei die Lösungen zu $(\mathcal{P}_u(\Omega))$ und $(\mathcal{P}_p(\Omega))$ und somit konstante Größen. Diese Darstellung ist eines der zentralen Ergebnisse der Sensitivity Analysis. Die Rechtfertigung der obigen Gleichung würde den Rahmen dieser Arbeit sprengen, darum verweisen wir auf die Arbeit von C ea [20] und Kapitel 9 aus [24].

Wir k onnen nun Lemma (3.10) und (3.11) benutzen um die Shapeableitung f ur J zu be-

rechnen

$$\begin{aligned}
\left\langle \frac{\partial J}{\partial \Omega}, V \right\rangle &= \int_{\partial \Omega} (j(x, u(x)) + \mathcal{C}(\epsilon(u)) : \epsilon(p) - p \cdot f) V \cdot \vec{n} \, d\nu \\
&+ \int_{\Gamma_N} \left(\frac{\partial l(x, u(x))}{\partial \vec{n}} + h l(x, u(x)) \right) V \cdot \vec{n} \, d\nu \\
&- \int_{\Gamma_N} \left(\frac{\partial (g \cdot p)}{\partial \vec{n}} + h g \cdot p \right) V \cdot \vec{n} \, d\nu \\
&- \int_{\Gamma_D} \left(\frac{\partial w}{\partial \vec{n}} + h w \right) V \cdot \vec{n} \, d\nu
\end{aligned}$$

mit $w = (u \cdot \mathcal{C}(\epsilon(p)) \cdot \vec{n} + p \cdot \mathcal{C}(\epsilon(u)) \cdot \vec{n})$.

Wir zeigen in einer Nebenrechnung, dass für w auf Γ_D gilt :

$$\left(\frac{\partial w}{\partial \vec{n}} + h w \right) = 2 \mathcal{C}(\epsilon(u)) : \epsilon(p)$$

Es gilt $u = p = 0$ auf Γ_D , d.h. :

$$\nabla u_i = a_i \vec{n} \quad \nabla p_i = b_i \vec{n} \quad a_i, b_i \in \mathbb{R} \quad (3.7)$$

Wir notieren $\vec{a} = (a_i)_i$ und $\vec{b} = (b_i)_i$. Dann gilt

$$\begin{aligned}
\mathcal{C}(\epsilon(u)) : \epsilon(p) &= (2\mu \epsilon(u) + \lambda \operatorname{tr}(\epsilon(u)) \mathbb{I}) : \epsilon(p) \\
&= 2\mu \epsilon(u) : \epsilon(p) + \lambda (\operatorname{tr}(\epsilon(u)) \mathbb{I}) : \epsilon(p) \\
&= 2\mu \operatorname{tr} \left[\frac{\nabla u + \nabla u^T}{2} \cdot \frac{\nabla p + \nabla p^T}{2} \right] + \lambda \operatorname{tr}(\epsilon(u)) \operatorname{tr}(\epsilon(p)) \\
&= \frac{\mu}{2} \underbrace{\sum_{i=1}^3 \nabla u_i \cdot \nabla p_i}_{\nabla u \cdot \nabla p^T} + \frac{\mu}{2} \underbrace{\sum_{i=1}^3 (u_{1,i}, u_{2,i}, u_{3,i}) \cdot (p_{1,i}, p_{2,i}, p_{3,i})}_{\nabla u^T \cdot \nabla p} \\
&\quad + \frac{\mu}{2} \underbrace{\sum_{i=1}^3 \nabla u_i \cdot (p_{1,i}, p_{2,i}, p_{3,i})}_{\nabla u \cdot \nabla p} + \frac{\mu}{2} \underbrace{\sum_{i=1}^3 (u_{1,i}, u_{2,i}, u_{3,i}) \cdot \nabla p_i}_{\nabla u^T \cdot \nabla p^T} \\
&\quad + \lambda \operatorname{tr}(\nabla u) \operatorname{tr}(\nabla p)
\end{aligned}$$

Benutzt man Gleichung (3.7) dann erhält man :

$$\begin{aligned} \mathcal{C}(\epsilon(u)) : \epsilon(p) &= \frac{\mu}{2} \left[\underbrace{\vec{a} \cdot \vec{b} \underbrace{\vec{n} \cdot \vec{n}}_{=1}} + \underbrace{\vec{a} \cdot \vec{b} (n_1^2 + n_2^2 + n_3^2)}_{=1} + \vec{a} \cdot \vec{n} \vec{b} \cdot \vec{n} + \vec{a} \cdot \vec{n} \vec{b} \cdot \vec{n} \right] \\ &+ \lambda \vec{a} \cdot \vec{n} \vec{b} \cdot \vec{n} \\ &= \mu \vec{a} \cdot \vec{b} + (\mu + \lambda)(\vec{a} \cdot \vec{n} \vec{b} \cdot \vec{n}) \end{aligned}$$

Betrachten wir jetzt $(\frac{\partial w}{\partial \vec{n}} + h w)$. Zunächst einmal gilt wegen $u = p = 0$ auf Γ_D :

$$\begin{aligned} \left(\frac{\partial w}{\partial \vec{n}} + h w \right) &= \frac{\partial w}{\partial \vec{n}} = \nabla (u \cdot \mathcal{C}(\epsilon(p)) \cdot \vec{n}) \cdot \vec{n} \\ &+ \nabla (p \cdot \mathcal{C}(\epsilon(u)) \cdot \vec{n}) \cdot \vec{n} \end{aligned}$$

Für den ersten Term gilt :

$$\begin{aligned} \nabla (u \cdot \mathcal{C}(\epsilon(p)) \cdot \vec{n}) \cdot \vec{n} &= (\nabla u^T \mathcal{C}(\epsilon(p)) \cdot \vec{n}) \cdot \vec{n} + \underbrace{u \cdot \nabla (\mathcal{C}(\epsilon(p))) \vec{n}}_{=0} \cdot \vec{n} \\ &= \left(\nabla u^T \left(2\mu \left(\frac{\nabla p + \nabla p^T}{2} \right) + \lambda \operatorname{tr}(\nabla p) \mathbb{I} \right) \cdot \vec{n} \right) \cdot \vec{n} \\ &= \left(\nabla u^T \left(\mu \vec{b} + \mu (\vec{b} \cdot \vec{n}) \vec{n} + \lambda \underbrace{\operatorname{tr}(\nabla p) \cdot \vec{n}}_{= \vec{b} \cdot \vec{n}} \right) \right) \cdot \vec{n} \\ &= \left(\mu (\vec{a} \cdot \vec{b}) \vec{n} + \mu (\vec{a} \cdot \vec{n}) (\vec{b} \cdot \vec{n}) \vec{n} + \lambda (\vec{a} \cdot \vec{n}) (\vec{b} \cdot \vec{n}) \vec{n} \right) \cdot \vec{n} \\ &= \mu \vec{a} \cdot \vec{b} + (\mu + \lambda)(\vec{a} \cdot \vec{n} \vec{b} \cdot \vec{n}) \end{aligned}$$

durch vertauschen von \vec{a} und \vec{b} gilt für den zweiten Term selbiges.

Damit gilt also tatsächlich :

$$\left(\frac{\partial w}{\partial \vec{n}} + h w \right) = 2 \mathcal{C}(\epsilon(u)) : \epsilon(p) \quad \text{auf } \Gamma_D$$

Dann gilt weiter für $dJ(\Omega; V)$:

$$\begin{aligned}
dJ(\Omega; V) &= \int_{\Gamma_N} (j(x, u(x)) + \mathcal{C}(\epsilon(u)) : \epsilon(p) - p \cdot f) V \cdot \vec{n} \, d\nu \\
&+ \int_{\Gamma_N} \left(\frac{\partial l(x, u(x))}{\partial \vec{n}} + hl(x, u(x)) \right) V \cdot \vec{n} \, d\nu \\
&- \int_{\Gamma_N} \left(\frac{\partial (g \cdot p)}{\partial \vec{n}} + hg \cdot p \right) V \cdot \vec{n} \, d\nu \\
&- \int_{\Gamma_D} (-j(x, u(x)) + \mathcal{C}(\epsilon(u)) : \epsilon(p) + p \cdot f) V \cdot \vec{n} \, d\nu
\end{aligned}$$

Wählt man jetzt j und l wie folgt :

$$j(x, u(x)) = f \cdot u \quad l(x, u(x)) = g \cdot u$$

Dann gilt für die Lösung p von (\mathcal{P}_p) und die Lösung u von (\mathcal{P}_u)

$$p = -u$$

Einsetzen liefert dann die Aussage :

$$\begin{aligned}
\left\langle \frac{\partial J}{\partial \Omega}, V \right\rangle &= \int_{\Gamma_N} \left(2 \left[\frac{\partial (g \cdot u)}{\partial \vec{n}} + hg \cdot u + f \cdot u \right] - \mathcal{C}\epsilon(u) : \epsilon(u) \right) V \cdot \vec{n} \, d\nu \\
&+ \int_{\Gamma_D} (\mathcal{C}\epsilon(u) : \epsilon(u)) V \cdot \vec{n} \, d\nu
\end{aligned}$$

□

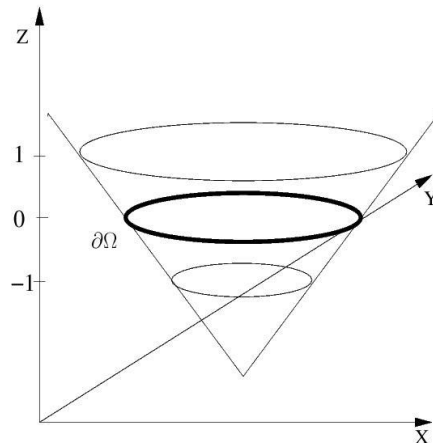


Abbildung 4.1: Levelsetfunktion und Isolevel Null als Rand für Ω

4 Levelsetmethoden

Ziel dieses Abschnittes soll es sein, den Optimierungsprozess in einem Levelset basierten Szenario zu formulieren.

Levelsetmethoden wurden zuerst von Osher und Sethian [32] eingeführt, als einfache Möglichkeit die Evolution von Kurven oder Flächen zu beschreiben. Die zentrale Idee ist recht einfach. Sei $\Gamma \in \mathbb{R}^d$ das betrachtete Interface(Hyperfläche) mit Codimension 1, für $d = 3$ eine zwei-dimensionale Fläche, welches ein Gebiet $\Omega \subset \mathbb{R}^d$ berandet. Γ läßt sich nun in folgender Weise als ein Isolevel einer höherdimensionalen, glatten (mindestens lipschitz-stetigen) Funktion, der sogenannten Levelsetfunktion, $\phi(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ identifizieren :

$$\begin{cases} \phi(x) = 0 & \Leftrightarrow x \in \partial\Omega = \Gamma \\ \phi(x) < 0 & \Leftrightarrow x \in \Omega \\ \phi(x) > 0 & \Leftrightarrow x \notin \Omega \end{cases}$$

Die Variation des Interfaces in der Zeit, wird nun implizit durch die Variation der Levelsetfunktion in der Zeit ausgedrückt. Dieses Vorgehen ist leicht auf Probleme in höheren dimensionen anwendbar. Ein weiterer großer Vorteil ist das sehr einfache Verarbeiten von aufeinandertreffenden und verschmelzenden Interface-rändern. Mittlerweile sind Levelsetmethoden gut ausgearbeitet und bieten sehr robuste, akkurate und effektive Verfahren zur Berechnung der Evolution von Flächen. Die für uns relevanten Methoden wollen wir in diesem Kapitel kurz vorstellen.

4.1 Levelsetgleichung

Sei $\Omega \subset D \subset \mathbb{R}^3$ ein gegebenes Gebiet mit glattem Rand, welches in obiger Weise durch die Levelsetfunktion $\phi \in C_0^\infty(\Omega)$ beschrieben wird.

$$\begin{aligned}\partial\Omega &= \{x \in D \mid \phi(x) = 0\} &=: [\phi = 0] \\ \Omega &= \{x \in D \mid \phi(x) < 0\} &=: [\phi < 0]\end{aligned}$$

Mit dieser Darstellung lassen sich die geometrischen Größen \vec{n} , die äußere Einheitsnormale, und h , die mittlere Krümmung, der von $\partial\Omega$ beschriebenen Fläche durch die Levelsetfunktion berechnen.

$$\vec{n} = \frac{\nabla\phi}{\|\nabla\phi\|} \quad h = \operatorname{div}(\vec{n})$$

Die Evolution von Flächen und Gebieten wird nun durch eine zeitabhängige Levelsetfunktion erreicht.

$$\begin{aligned}\partial\Omega_{(t)} &= \{x \in D \mid \phi(t, x) = 0\} &=: [\phi_{(t)} = 0] \\ \Omega_{(t)} &= \{x \in D \mid \phi(t, x) < 0\} &=: [\phi_{(t)} < 0]\end{aligned}$$

Setzt man voraus, dass sich die einzelnen Isolevel von ϕ mit einer Normalengeschwindigkeit $v(x) \in C^0(D)$ bewegen, d.h. für die Trajektorie $x(t)$ eines Partikels x_0 auf einem Isolevel gilt

$$\dot{x}(t) = v(x(t)) \vec{n}(x(t))$$

dann lässt sich folgende Hamilton-Jacobi-Gleichung verifizieren:

$$\begin{aligned}\frac{\partial}{\partial t} (\phi(x(t))) &= 0 \\ \Leftrightarrow \phi(t, x(t)) + \nabla \phi(t, x(t)) \cdot \dot{x}(t) &= 0 \\ \Leftrightarrow \phi(t, x(t)) + v(x(t)) \|\nabla \phi(t, x(t))\| &= 0\end{aligned}$$

Obige Gleichung wird auch Levelsetgleichung genannt. Sie macht die Beziehung zwischen geometrischer Variation des Randes $\partial\Omega$ und der zeitlichen Variation der Levelsetfunktion deutlich. Hat man eine Variation des Randes, beschrieben durch die Normalengeschwindigkeit, gegeben, dann liefert die Levelsetfunktion die zeitliche Veränderung der Levelsetfunktion. Umgekehrt ist nicht sofort klar welches Geschwindigkeitsfeld eine Variation der Levelsetfunktion erzeugt. Diese Fragestellung ist Gegenstand der in Abschnitt 4.4 geführten Diskussion, da unser Ziel ist die *ShapeSensitivityAnalysis* aus Kapitel 3 von der Betrachtung von Vektorfeldern zu lösen und vollständig in eine Levelsetformulierung zu übersetzen. Vorweg genommen können wir sagen, dass das Ergebniss aus Abschnitt 4.4 die Vermutung bestätigt, die sich aus der Levelsetgleichung ergibt.

$$v(x(t)) = \frac{\dot{\phi}(t, x(t))}{\|\nabla \phi(t, x(t))\|}$$

Hat man eine Variation der Levelsetfunktion gegeben, dann gibt die Levelsetgleichung das Geschwindigkeitsfeld der geometrischen Veränderung von $\partial\Omega$ an.

4.2 Extension Velocity

Wie im obigen Abschnitt erläutert, sind Levelset-methoden gut geeignet, um die Evolution von Hyperflächen (Isolevel) unter einem gegebenen Geschwindigkeitsfeld zu beschreiben. Aus der Levelsetgleichung geht aber hervor, dass diese Geschwindigkeitsfeld für alle Isolevel, also auf dem ganzen betrachteten Gebiet D , definiert sein muss. Im allgemeinen ist man nur an der Evolution eines Isolevels, wie in unserem Fall dem Null-Isolevel, interessiert. So kann es sein, dass bei bestimmten Problemen ein Geschwindigkeitsfeld V zunächst nur auf dem relevanten Isolevel definiert werden kann. Um die Stabilität des Levelsetverfahrens zu gewährleisten, ist es notwendig eine angemessene Fortsetzung der Geschwindigkeit V_{ext} von dem betrachteten Isolevel auf die umgebenden Isolevel zu erzeugen. Die Fortsetzung muss dabei nicht auf ganz D berechnet werden, es genügt eine kleine Umgebung (narrow-band).

Wie ist nun eine angemessene Fortsetzung zu wählen ? Eine wünschenswerte Eigenschaft von V_{ext} wäre die Erhaltung einer Signed-Distance Funktion.

Wählt man V_{ext} so, dass folgende Gleichung gilt

$$\nabla V_{ext} \cdot \nabla \phi = 0 \quad (4.1)$$

Setzt man also V in Normalenrichtung konstant fort, dann ist einfach zu zeigen, dass die Gleichung

$$\begin{aligned} \frac{\partial \|\nabla \phi\|^2}{\partial t} &= \frac{\partial}{\partial t} (\nabla \phi \cdot \nabla \phi) = 2 \nabla \phi \cdot \frac{\partial}{\partial t} (\nabla \phi) \\ &= -2 \nabla \phi \cdot \nabla V_{ext} \|\nabla \phi\| - 2 \nabla \phi \cdot \nabla (\|\nabla \phi\| V_{ext}) \end{aligned} \quad (4.2)$$

gilt. Ist ϕ nun eine Signed-Distance Funktion, gilt also

$$\|\nabla \phi\| = 1$$

dann ist die rechte Seite in obiger Gleichung gleich Null, denn für den ersten Term gilt wegen Gleichung (4.1)

$$-2 \nabla \phi \cdot \nabla V_{ext} \|\nabla \phi\| = 0.$$

Für den zweiten Term gilt

$$-2 \nabla \phi \cdot \nabla (\|\nabla \phi\| V_{ext}) = -2 \nabla \phi \cdot [\nabla (\|\nabla \phi\|) V_{ext} + \|\nabla \phi\| \nabla V_{ext}] .$$

Das heißt ϕ bleibt zumindest für kleine Zeitschritte nahe einer signierte Distanzfunktion. Um diese Fortsetzung im diskreten Fall zu konstruieren benutzen wir die Fast-Marching-Method, welche wir weiter unten kurz erläutern wollen.

4.3 Reinitialisierung

Unter Reinitialisierung versteht man das Ersetzen einer Levelsetfunktion ϕ durch eine Levelsetfunktion ϕ_{new} , welche eine signifizierte Distanzfunktion ist und mit ϕ auf dem betrachteten Isolevel übereinstimmt.

$$\phi_{new} = \phi \quad \text{auf } [\phi = 0] \quad \|\nabla\phi_{new}\| = 1$$

Diese Reinitialisierung wird in unserem späteren Optimierungsalgorithmus regelmäßig durchgeführt, da eine signifizierte Distanzfunktion eine bessere Kondition zur Berechnung, der von ihr beeinflussten Größen hat. So ist zum Beispiel die mittlere Krümmung h der impliziten Fläche $[\phi = 0]$, wie weiter oben beschrieben, gegen durch

$$h = \operatorname{div} \left(\frac{\nabla\phi}{\|\nabla\phi\|} \right)$$

hierbei sieht man, dass es für $\|\nabla\phi\| \ll 1$ zu nicht unerheblichen Rundungsfehlern kommen kann. Im nächsten Abschnitt werden wir sehen, dass auch die Berechnung der Abstiegsrichtung von der Norm des Gradienten der Levelsetfunktion abhängig ist, wie am Ende von Abschnitt 4.1 angedeutet. Das heißt die Norm des Gradienten von ϕ beeinflusst unmittelbar die Berechnung der Shapeableitung. Deshalb ist es notwendig mit einer Levelsetfunktion zu arbeiten, welche 'nahe' einer signifizierten Distanzfunktion ist.

Die Reinitialisierung kann auf mehreren Wegen bewerkstelligt werden. Wir wollen auch hierbei die Fast-Marching-Method benutzen um die signifizierte Distanzfunktion ϕ_{new} zu erzeugen. Dazu lösen wir die Eikonalgleichung

$$\|\nabla T\| = 1$$

auf beiden Seiten der Fläche $[\phi = 0]$, mit den Randwertbedingungen $T = 0$ auf $[\phi = 0]$. Die allgemeine Formulierung der Eikonalgleichung lautet

$$\|\nabla T\| F = 1$$

und bietet eine Alternative zur Levelsetgleichung, um die Evolution von Flächen zu schreiben. Im Unterschied zur Levelsetgleichung, welche ein Initialwertproblem beschreibt, beschreibt die Eikonalgleichung ein Randwertproblem. Die Eikonalgleichung stellt die Bedingung $F > 0$ ($F < 0$) an das Geschwindigkeitsfeld F (da F stetig sein muß), das heißt ein betrachtetes Isolevel bewegt sich entweder nach 'außen' oder nach 'innen'. Sie ist daher nicht so flexibel wie die Levelsetgleichung, ist aber numerisch sehr effizient lösbar.

4.4 Shapeableitung in Levelsetformulierung

Ziel dieses Abschnittes soll es sein, den Optimierungsprozess in einem Levelsetbasierten Setting zu formulieren. Dazu wollen wir eine Brücke schlagen zu den Methoden der Sensitive Analysis wie im vorherigen Kapitel beschrieben, dann stehen uns die Ergebnisse dieser Arbeit zur Verfügung. Wir beschränken unsere Betrachtungen auf den Funktionenraum $C_0^\infty(D)$.

Zuerst soll geklärt werden, wie Variationen des von uns betrachteten Gebiet erzeugt werden. Sei also die Levelsetfunktion ϕ gegeben, welche implizit unser Gebiet Ω beschreibt :

$$\begin{aligned}\partial\Omega &= \{x \in D \mid \phi(x) = 0\} &=: [\phi = 0] \\ \Omega &= \{x \in D \mid \phi(x) < 0\} &=: [\phi < 0]\end{aligned}$$

Die Variation des Gebietes Ω wird durch Variation der Levelsetfunktion ϕ erreicht. Sei dazu $\psi \in C_0^\infty(D)$ und $s \in \mathbb{R}$. Die Funktion

$$\phi_s(x) = \phi_0(x) + s \psi(x)$$

bezeichne die in Richtung ψ variierte Levelsetfunktion. Analog definieren wir uns die Variation Ω_s des Gebiets Ω

$$\begin{aligned}\partial\Omega_s &= \{x \in D \mid \phi_s(x) = 0\} &=: [\phi_s = 0] \\ \Omega_s &= \{x \in D \mid \phi_s(x) < 0\} &=: [\phi_s < 0]\end{aligned}$$

Die Variation eines Gebiets Ω wurde in Kapitel 3 mit der 'Velocity-Method' konstruiert.

$$\Omega_s = T_s(V)(\Omega) \quad s \in \mathbb{R}$$

wobei $T_s(V)(\Omega)$ die in Gleichung (3.5) definierte Abbildung ist. Ziel ist es nun die Variation des Gebietes Ω , erzeugt durch die Störung der Levelsetfunktion ϕ , in der Form zu schreiben, die der 'Velocity-Method' entspricht. Dann stehen uns die Ergebnisse der Sensitive Analysis zur Verfügung. Das Problem ist also ein Vektorfeld V zu finden, so das die Gleichung gilt:

$$[\phi_s < 0] = T_s(V)([\phi_0 < 0]) \quad s \in \mathbb{R} \tag{4.3}$$

Betrachten wir also die Variation der Levelsetfunktion :

$$\phi_s(x) = \phi_0(x) + s \psi(x)$$

mit $s \in \mathbb{R}^+$ und $\psi \in C^2(D, \mathbb{R})$. Die Idee ist jetzt die Bewegung der einzelnen Isolevel von ϕ_s zu analysieren um ein Geschwindigkeitsfeld zu erkennen. Wir schauen uns zunächst die Evolution des Isolevels $[\phi = 0]$ an. Die Ergebnisse lassen sich dann analog auf die anderen Isolevel übertragen.

Sei also $\bar{x} \in [\phi = 0]$ ein Partikel auf dem Isolevel $[\phi = 0]$. Gesucht ist die Trajektorie $x(t)$, entlang sich dieses Partikel bewegt.

$$\begin{aligned}\phi_s(x(s)) &= \phi(x(s)) + s \psi(x(s)) = 0 \quad \forall s > 0 \\ x(0) &= \bar{x}\end{aligned}$$

Wir leiten obige Gleichung nach s ab und erhalten :

$$\begin{aligned}\frac{\partial}{\partial s} [\phi(x(s)) + s \cdot \psi(x(s))] &= 0 \\ \Leftrightarrow \nabla \phi(x(s)) \cdot \dot{\phi}(s) + \psi(x(s)) + s \cdot \nabla \psi(x(s)) &= 0 \\ \Leftrightarrow [\nabla \phi + s \cdot \nabla \psi] \cdot \dot{x}(s) + \psi(x(s)) &= 0\end{aligned}$$

Offensichtlich gilt für Lösungen $x(s)$:

$$\dot{x}(s) = -\psi(x(s)) \frac{\nabla \phi + s \cdot \nabla \psi}{\|\nabla \phi + s \cdot \nabla \psi\|^2} + c(s) [\nabla \phi + s \cdot \nabla \psi]^\perp$$

Für eine beliebige skalare Funktion c . Somit finden wir zeitabhängige Vektorfelder V , so dass die zugehörige Abbildung $T_s(V)(\cdot)$ der Gleichung (4.3) genügt.

$$V(s, x) := -\psi(x) \frac{\nabla \phi + s \cdot \nabla \psi}{\|\nabla \phi + s \cdot \nabla \psi\|^2} + c(s) [\nabla \phi + s \cdot \nabla \psi]^\perp$$

Der Struktursatz 3.8 sagt aus, dass die Shapeableitung nur von $V(0, x)$ auf $[\phi = 0]$ abhängig ist. Weiter folgt aus Bemerkung 3.9, dass der tangentielle Anteil von $V(0, x)$ für die Shapeableitung nicht relevant ist. Somit gilt für die Shapeableitung in Richtung des oben definierten zeitabhängigen Vektorfeldes :

$$dJ(\Omega, V) = dJ(\Omega; \bar{V})$$

wobei

$$\bar{V}(s, x) := -\psi(x) \frac{\nabla \phi}{\|\nabla \phi\|^2} = -\frac{\psi(x)}{\|\nabla \phi\|} \vec{n} \quad (4.4)$$

ein stationäres Vektorfeld ist ($\bar{V} \in C^\infty(D)$), welches durch $\psi(x)$ eindeutig definiert wird. Durch die Beziehung in Gleichung (4.4) lässt sich nun die Shapeableitung in Levelsetformulierung schreiben.

$$J(\phi) := J(\Omega)$$

$$dJ(\phi; \psi) := dJ(\Omega; \bar{V})$$

Dabei hängt $dJ(\phi; \psi)$ nur von $\psi|_{[\phi=0]}$ ab. Damit ist möglich die Vektordistribution aus Definition 3.6 mit einem Objekt aus dem Dualraum $C_0^\infty([\phi = 0])'$ von $C_0^\infty([\phi = 0])$ zu identifizieren.

$$\left\langle \frac{\partial J(\phi)}{\partial \phi}; \psi \right\rangle := dJ(\phi; \psi) = dJ(\Omega; \bar{V}) := \left\langle \frac{\partial J(\Omega)}{\partial \Omega}; \bar{V} \right\rangle$$

Das Funktional $\left\langle \frac{\partial J(\phi)}{\partial \phi}; \cdot \right\rangle \in C_0^\infty([\phi = 0])'$ beschreibt in obiger Art und Weise die Shapeableitung der Shapefunktion $J(\Phi)$ in Richtungen $\psi \in C_0^\infty(D)$.

5 Shapegradient und Regularisierung

5.1 Shapegradient

Unser Optimierungsalgorithmus soll auf einem Gradientenfluss basieren. Dazu ist es notwendig zu klären, wie ein Gradient einer Shapefunktion aussieht.

Definition 5.1 (Shapegradient)

Sei $J(\phi)$ eine shapedifferenzierbare Shapefunktion aus der in Satz 3.15 betrachteten Klasse. Desweiteren seien der Funktionenraum \mathbb{F} mit einem Skalarprodukt $\langle \cdot, \cdot \rangle_M$ gegeben. Dann wird der Shapegradient, den wir im Weiteren mit $grad$ bezeichnen, bezüglich $\langle \cdot, \cdot \rangle_M$ in \mathbb{F} durch folgende Gleichung in \mathbb{F} definiert, falls diese eine Lösung besitzt :

$$\langle grad, \psi \rangle_M = \left\langle \frac{\partial J(\phi)}{\partial \phi}; \psi \right\rangle \quad \forall \psi \in \mathbb{F} \quad (5.1)$$

Die Lösbarkeit der obigen Gleichung ist gewährleistet, wenn $\langle \cdot, \cdot \rangle_M$ eine beschränkte und koerzitive Bilinearform ist, \mathbb{F} mit $\langle \cdot, \cdot \rangle_M$ ein Hilbertraum ist und $\left\langle \frac{\partial J(\phi)}{\partial \phi}; \cdot \right\rangle$ ein lineares, beschränktes Funktional auf \mathbb{F} ist. $grad$ ist dann selbst eine Funktion aus dem Raum \mathbb{F} . Für die Shapefunktion aus (2.2) ist $\left\langle \frac{\partial J(\phi)}{\partial \phi}; \cdot \right\rangle$ linear und beschränkt.

Dies ist eine sehr allgemeine Definition, da der zu wählende Funktionenraum \mathbb{F} sowohl auf dem Rand von Ω [$\phi = 0$], als auch auf dem ganzen Konstruktionsgebiet D definiert sein kann. Da $\left\langle \frac{\partial J(\phi)}{\partial \phi}; \cdot \right\rangle$ ein Objekt mit Träger auf dem Isolevel [$\phi = 0$] ist, liegt es zunächst nahe einen Funktionenraum mit zugehörigem Skalarprodukt auf diesem Isolevel zu definieren, um einen Gradienten zu definieren. Diese Vorgehen wurde von uns zuerst benutzt, um Abstiegsrichtungen zu berechnen.

Methode 5.2 (1. Variante)

Für \mathbb{F} haben wir den Hilbertraum $L^2([\phi = 0])$ mit dem Skalarprodukt $\langle \cdot, \cdot \rangle_{L^2([\phi=0])}$ gewählt. Somit lautet die Gleichung 5.1 :

$$\langle grad, \psi \rangle_{L^2([\phi=0])} = \left\langle \frac{\partial J(\phi)}{\partial \phi}; \psi \right\rangle = \langle \alpha, \psi \rangle_{L^2([\phi=0])} \quad \forall \psi \in L^2([\phi = 0]) \quad (5.2)$$

dabei ist α die Funktion aus Satz 3.15 mit $\frac{1}{\|\nabla\phi\|}$ multipliziert.

$$\alpha = \begin{cases} \left(2 \left[\frac{\partial(g \cdot u)}{\partial n} + hg \cdot u + f \cdot u \right] - C\epsilon(u) : \epsilon(u) \right) \frac{1}{\|\nabla\phi\|} & \text{auf } \Gamma_N \\ (C\epsilon(u) : \epsilon(u)) \frac{1}{\|\nabla\phi\|} & \text{auf } \Gamma_D \end{cases}$$

Die Lösung der Gleichung (5.1) ist dann gegeben durch :

$$grad = \alpha \quad (5.3)$$

Damit haben wir eine Updatefunktion auf $[\phi = 0]$ definiert. Um eine Updatefunktion auf ganz D zu erhalten, wenden wir die Methode an, welche in Abschnitt 4.2 vorgestellt wurde. Dabei wird $grad$ einfach in Normalenrichtung konstant fortgesetzt und wir so erhalten eine Funktion

$$\phi_{update} \quad \text{mit} \quad \phi_{update} = grad \quad \text{auf} \quad [\phi = 0]$$

Die Variation der Levelsetfunktion gestaltet sich dann wie folgt :

$$\phi_{new} = \phi_{old} - \Delta t \phi_{update}$$

Auf die Berechnung einer geeigneten Schrittweite Δt wird in Kapitel 8 genauer eingegangen.

Die numerische Implementierung dieses Verfahrens hat gezeigt, dass diese Methode jedoch einige Nachteile aufweist. Und zwar hat sich gezeigt, dass dieser Shapegradient auf dem Isolevel $[\phi = 0]$ hohen Schwankungen unterliegt. Mit Hilfe einer Metrik mit glättenden Eigenschaften, z.B.

$$\langle \varphi, \psi \rangle_M := \int_{[\phi=0]} \varphi \psi + \frac{\sigma^2}{2} \nabla \varphi \cdot \nabla \psi \, d\nu \quad \sigma \in \mathbb{R} \quad (5.4)$$

können regularisierte Shapegradienten auf dem Isolevel erzeugt werden. Dieses Vorgehen funktioniert, falls der Rand von Ω $[\phi = 0]$ glatt genug ist und durch die Diskretisierung gut approximiert wird. Während des Optimierungsprozesses tritt aber immer wieder die Situation ein, dass Gebietsränder aufeinander treffen. Dabei können die neu entstandenen Ränder stark verrauscht sein. Dann ist die glättende Eigenschaft der Metrik aber nur noch gewährleistet, wenn mit einer hinreichend feinen Auflösung gearbeitet wird. Eine feine Diskretisierung lässt jedoch den Rechenaufwand des Elastizitätslösers stark ansteigen, was den Optimierungsprozess enorm verlangsamt.

5.2 Regularisierung

Aus oben genannten Gründen wollen wir eine weitere Methode vorstellen, eine Abstiegsrichtung zu berechnen.

Methode 5.3 (2. Variante)

Wir wählen uns jetzt für \mathbb{F} den Hilbertraum $H_0^{1,2}(D)$ mit dem Skalarprodukt $\langle \cdot, \cdot \rangle_M$ aus Gleichung (5.4) für ein geeignetes $\sigma > 0$. Die Gleichung (5.1) lautet dann :

$$\int_D grad \psi + \frac{\sigma^2}{2} \nabla grad \cdot \nabla \psi \, dx = \left\langle \frac{\partial J(\phi)}{\partial \phi}; \psi \right\rangle \quad \forall \psi \in H_0^{1,2}(D)$$

Die linke Seite ist eine beschränkte und koerzitive Bilinearform, daraus folgt mit dem Satz von Lax-Milgram die Existenz einer Lösung $grad \in H_0^{1,2}(D)$. Diese ist, im Gegensatz

zu dem Shapegradienten aus Methode 5.2 , nicht nur auf dem Isolevel $[\phi = 0]$ definiert, sondern auf ganz D . Daher entfällt das Fortsetzen in Normalenrichtung. Der Update der Levelsetfunktion erfolgt analog :

$$\phi_{new} = \phi_{old} - \Delta t \text{ grad}$$

Die Diskussion der Schrittweitenbestimmung erfolgt in Kapitel 7.

6 Diskretisierung

In diesem Kapitel wollen wir kurz auf die Diskretisierung eingehen, die benutzt haben. Dabei beschreiben wir die grundlegenden Aspekte der Ortsdiskretisierung und wie wir anhand dieser eine diskrete Shapeableitung konstruiert haben.

6.1 Diskretisierung der Elastizität

Eine naive Vorgehensweise eine Diskretisierung eines Gebietes Ω mit komplizierten Rändern zu erzeugen, ist ein uniformes Gitter so weit zu verfeinern, dass eine gewisse Genauigkeit erreicht ist. Diese Methode führt jedoch zu einem raschen Anwachsen der Freiheitsgrade und somit zu hohem Rechen- und Speicheraufwand. Auch bei Verwendung von adaptiven Diskretisierungsmethoden ist dieser Nachteil von Belang.

Wir benutzen hier eine andere Methode, um komplizierte Ränder angemessen zu approximieren, nämlich die Diskretisierung mit den sogenannten Composite Finite Elements (CFE). Der Zweck dieser Diskretisierung beinhaltet mehr als nur die Auflösung von Gebietsrändern, sie dient in erster Linie der Konstruktion eines Finite Element Raumes, welcher an ein Differentialgleichungsproblem mit springenden Koeffizienten angepasst ist. Damit ist es dann z.B. möglich die Elastizitätsgleichung für inhomogene Körper zu lösen. Für eine Einführung verweisen wir auf [26] [27] [29] [41]. Für uns sind aber nur die Auflösungseigenschaften von Interesse. Im Folgenden wollen wir deshalb kurz erklären wie die Ortsdiskretisierung aussieht.

Die Basis der Diskretisierung bildet ein uniformes Gitter aus kubischen (in 2D aus quadratischen) Elementen. In diese Elemente wird ein Tetraeder-Gitter eingebettet, so dass jedes kubische Element in 6 (in 2D in 2) sogenannte Standard Tetraeder unterteilt wird. So erhält man eine Zerlegung z.B. des Einheitswürfels $[0, 1]^3$ in Simplicies, deren Inneres paarweise disjunkt ist. Sei $I \subset \mathbb{N}$ eine endliche Indexmenge. Dann gilt

$$[0, 1]^3 = \bigcup_{i \in I} T_i \quad T_i \cap T_j \neq \emptyset \Leftrightarrow T_i \cup T_j \text{ ist ein Untersimplex von } T_i \text{ und } T_j$$

und $\mathcal{M} := \{T_i : i \in I\}$ heißt zulässiges simpliciales Gitter von $[0, 1]^3$. Die Menge der Eckpunkte, im Folgenden Knotenmenge genannt, bezeichnen wir mit \mathcal{N} . Der betrachtete diskrete Lösungsraum

$$\mathcal{S} := \{\varphi \in C^0([0, 1]^3) : \varphi|_T \text{ affin linear für } T \in \mathcal{M}\}$$

ist der Raum der stetigen, stückweise affin linearen Funktionen mit der üblichen Lagrange-Basis $\{\varphi_x\}_{x \in \mathcal{N}}$.

$$\varphi_x(y) = \delta_{xy} \quad \forall x, y \in \mathcal{N}$$

Ist nun $[\phi < 0] \subset [0, 1]^3$ das aufzulösende Gebiet, so wird durch folgende Verfeinerungsmethode der Rand von Ω approximiert. Schneidet $\partial\Omega$ ein kubisches Element, so werden

anhand der Schnittpunkte von $\partial\Omega$ mit den Kanten der standard Tetraeder diese standard Tetraeder so in Untertetraeder zerlegt, dass die Eckpunkte der Untertetraeder entweder obige Schnittpunkte oder Knoten aus \mathcal{N} sind. Der Fall, dass $\partial\Omega$ einen Knoten aus \mathcal{N} schneidet wird durch eine leichte Modifikation von Ω ausgeschlossen. Damit bleibt das Gitter regulär. Durch diesen Verfeinerungsschritt des simplizialen Gitters wird eine Approximation von Ω aus Simplicizes erzeugt, welche wir mit $\tilde{\Omega}$ bezeichnen wollen. Die Menge dieser Simplicizes benennen wir mit $\tilde{\mathcal{M}}$. Es ist also $\tilde{\mathcal{M}}$ ein zulässiges und reguläres simpliziales Gitter von $\tilde{\Omega}$. Der Rand von Ω wird durch eine Triangulierung aus Facetten von Untertetraedern approximiert. Die Menge dieser Facetten bezeichnen wir mit F . Bei dem Verfeinerungsschritt entstehen keine neuen Freiheitsgrade. Es werden lediglich die Träger der Basisfunktionen φ_x $x \in \mathcal{N}$ modifiziert, deren Träger $\partial\Omega$ schneidet. Und zwar werden diese Basisfunktionen einfach 'abgeschnitten'

$$\tilde{\varphi}_x = \chi_{\tilde{\Omega}} \varphi_x$$

hierbei ist $\chi_{\tilde{\Omega}}$ die charakteristische Funktion von $\tilde{\Omega}$. Die beschriebene Vorgehensweise wird

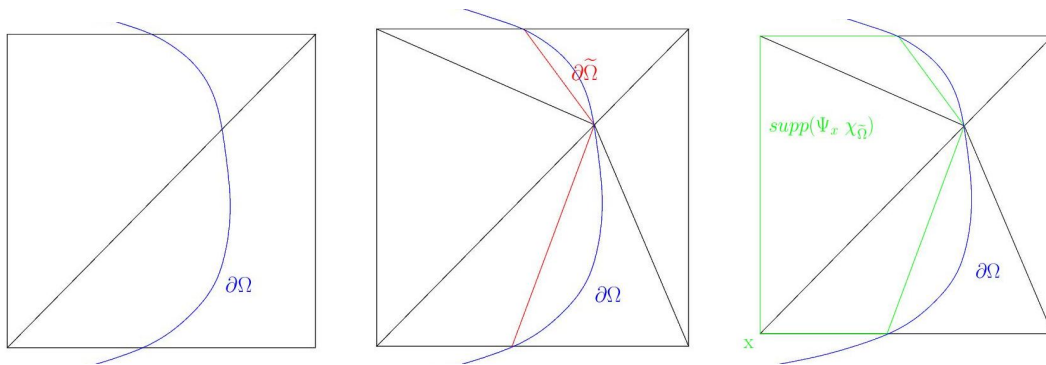


Abbildung 6.1: Die obigen Abbildungen zeigen den Verfeinerungsschritt in 2D. links: Standardzerlegung eines Würfелеlements; mitte: Verfeinerung und Triangulierung des Randes; rechts: Modifikation des Trägers von Ψ_x

von Florian Liehr in seiner Diplomarbeit [29] ausführlich diskutiert. Zum Lösen des resultierenden Gleichungssystems wird dort eine Mehrgitterverfahren herangezogen. Der implementierte Löser wird von uns benutzt, um die Elastizitätsgleichung zu lösen.

6.2 Diskretisierung der Shapeableitung

Im Folgenden werden wir mit Großbuchstaben die diskreten Funktionen bezeichnen. So ist zum Beispiel

$$\Phi(x) = \sum_{x \in \mathcal{N}} \Phi_x \cdot \Psi_x(x) \quad \vec{\Phi} = (\Phi_x)_{x \in \mathcal{N}}$$

dabei ist $\vec{\Phi}$ der Knotenvektor aus den zu den Basisfunktionen gehörenden Koeffizienten. Der diskrete Shapegradient wird mit Grad gekennzeichnet.

Die kontinuierliche Shapefunktion aus (2.2) hat die Gestalt

$$J(\Omega) = \int_{\Omega} f u \, dx + \int_{\Gamma_N} g u \, d\nu = \int_{\Omega} \mathcal{C}(\epsilon(u)) : \epsilon(u) \, dx \quad (6.1)$$

Dann definieren wir die diskrete Shapefunktion $\hat{J}(\Phi)$ indem wir einfach die diskreten Größen einsetzen

$$\tilde{J}(\Phi) := \sum_{T \in \tilde{\mathcal{M}}} \int_T \zeta(\epsilon(U)) : \epsilon(U) \, dx$$

dabei ist $\zeta(\epsilon(U)) : \epsilon(U)$ konstant auf den Simplizes $T \in \tilde{\mathcal{M}}$, da U stückweise affin linear ist. Analog definieren wir die Anwendung des diskreten Funktionals $\langle \frac{dJ(\phi)}{d\phi}, \cdot \rangle$ auf eine Funktion ψ durch einsetzen der diskreten Funktionen

$$\begin{aligned} \left\langle \frac{\partial \tilde{J}(\Phi)}{\partial \Phi}; \Psi \right\rangle &:= \sum_{f \in F, f \subseteq \Gamma_N} \int_f \left(2 \left[\frac{\partial(G \cdot U)}{\partial N} + HG \cdot U + F \cdot U \right] - \zeta(\epsilon(U)) : \epsilon(U) \right) \frac{-\Psi}{\|\nabla \Phi\|} \, d\nu \\ &+ \sum_{f \in F, f \subseteq \Gamma_D} \int_f \zeta(\epsilon(U)) : \epsilon(U) \frac{-\Psi}{\|\nabla \Phi\|} \, d\nu \end{aligned}$$

Da $G = 0$ auf den freien Rändern ist vereinfacht sich die Gleichung zu

$$\begin{aligned} \left\langle \frac{\partial \tilde{J}(\Phi)}{\partial \Phi}; \Psi \right\rangle &:= \sum_{f \in F, f \subseteq \Gamma_N} \int_f (2F \cdot U - \zeta(\epsilon(U)) : \epsilon(U)) \frac{-\Psi}{\|\nabla \Phi\|} \, d\nu \\ &+ \sum_{f \in F, f \subseteq \Gamma_D} \int_f \zeta(\epsilon(U)) : \epsilon(U) \frac{-\Psi}{\|\nabla \Phi\|} \, d\nu \\ &= \sum_{f \in F, f \subseteq \Gamma_N} \int_f \underbrace{2F \cdot U \frac{-\Psi}{\|\nabla \Phi\|}}_{\text{Polynom 3. Grades}} - \underbrace{\zeta(\epsilon(U)) : \epsilon(U) \frac{-\Psi}{\|\nabla \Phi\|}}_{\text{Polynom 1. Grades}} \, d\nu \\ &+ \sum_{f \in F, f \subseteq \Gamma_D} \int_f \underbrace{\zeta(\epsilon(U)) : \epsilon(U) \frac{-\Psi}{\|\nabla \Phi\|}}_{\text{Polynom 1. Grades}} \, d\nu \end{aligned}$$

Für die Integration der Polynome auf den Dreiecken werden exakte simpliziale Quadraturformeln benutzt.

Nun können wir die diskreten Shapegradienten zu den Methoden (5.2) und (5.3) berechnen.

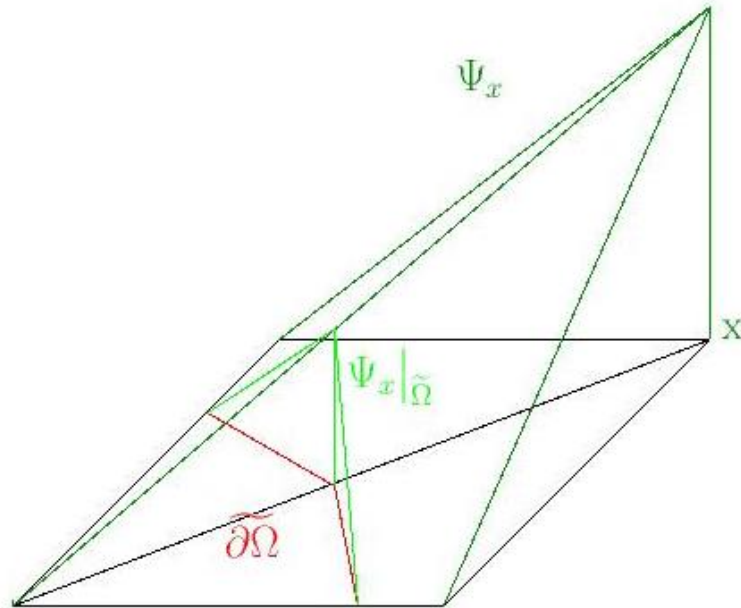


Abbildung 6.2: Die Abbildung zeigt die Spur (hellgrün) auf dem Rand (rot) einer randnahen Basisfunktion Ψ_x (dunkelgrün)

Die diskrete Gleichung für den Gradienten grad aus Methode (5.3) sieht dann wie folgt aus:

$$\int_D \text{Grad } \Psi_x + \frac{\sigma^2}{2} \nabla \text{Grad} \cdot \nabla \Psi_x dx = \left\langle \frac{\partial \tilde{J}(\Phi)}{\partial \Phi}; \Psi_x \right\rangle \quad \forall x \in \mathcal{N}$$

Hierbei ist wieder Ψ_x die Basisfunktion am Knoten x . Setzt man für G die Summe $\sum_{x \in \mathcal{N}} \mathcal{G}_x \Psi_x$ ein, dann erhält man folgendes Gleichungssystem.

$$\left(M + \frac{\sigma^2}{2} L \right) \cdot \vec{Grad} = \vec{R}$$

Dabei ist

$$M = (M_{ij})_{ij} \quad \text{mit} \quad M_{ij} = \int_D \Psi_y \Psi_x dx \quad x, y \in \mathcal{N}$$

die sogenannte Massmatrix und

$$L = (L_{ij})_{ij} \quad \text{mit} \quad L_{ij} = \int_D \nabla \Psi_y \cdot \nabla \Psi_x dx \quad x, y \in \mathcal{N}$$

die Steifigkeitsmatrix. $\vec{G} = (G_x)_x$ bezeichnet den Koeffizientenvektor der Funktion G und die rechte Seite des Gleichungssystems ist der Vektor

$$\vec{R} = \left(\left\langle \frac{\partial \tilde{J}(\Phi)}{\partial \Phi}; \Psi_x \right\rangle \right)_x \quad x \in \mathcal{N}$$

Die Lösung dieser Gleichung erfordert nur eine Modifikation der lokalen Matrizen und wird mit dem selben Löser gelöst den wir für die Elastizitätsgleichung benutzen.

Die Diskretisierung des Shapegradienten aus Methode 5.2 geschieht durch Projektion der Funktion α aus Gleichung (5.3) in den Finite-Elemente Raum auf dem Isolevel $[\Phi = 0]$, der wie folgt aussieht. Die Knotenmenge \mathcal{N}_S ist die Menge der Eckpunkte aller Facetten $f \in F$, die eine Triangulierung des Randes $\partial\Omega$ bilden. Der diskrete Lösungsraum

$$\mathcal{S}_S := \{ \varphi \in C^0([\Phi = 0]) : \varphi|_f \text{ affin linear für } f \in F \}$$

ist der Raum der stetigen, stückweise affin linearen Funktionen mit der üblichen Lagrange-Basis $\{\varphi_x\}_{x \in \mathcal{N}_S}$.

$$\varphi_x(y) = \delta_{xy} \quad \forall x, y \in \mathcal{N}_S$$

Die Projektion der Funktion α hat die Form

$$Grad = \sum_{x \in \mathcal{N}_S} \langle \alpha, \varphi_x \rangle_{L^2[\Phi=0]} \varphi_x$$

Dabei sind wieder Integrale von Polynomen auf den einzelnen Facetten zu berechnen. Diese werden wie oben erwähnt mit der entsprechenden simplizialen Quadraturformel berechnet. Damit haben wir eine Updatefunktion $Grad$ auf dem Rand $[\Phi = 0]$ gegeben. Diese setzen wir dann mit der Fast-Marching-Method in Normalenrichtung fort. Für Details zur Implementierung diese Verfahrens verweisen wir auf [34]. So erhalten wir eine Updatefunktion $\widetilde{Grad} \in C^0(D)$.

Es sei hier bemerkt, dass durch diverse Tests keine Konsistenz für einen der beiden Shapegradienten nachgewiesen werden konnte. Jedoch gibt es auch in der bis jetzt existierenden Literatur zu diesem Thema keinen Konsistenznachweis für die Shapeableitung der von uns betrachteten Shapefunktion.

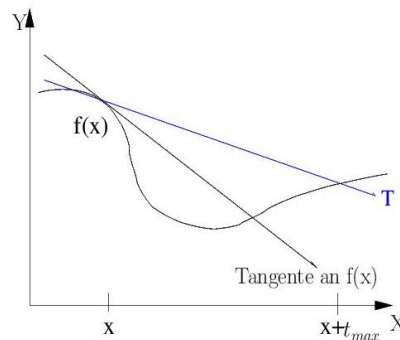


Abbildung 7.1: Schrittweitenbestimmung mit Armijo-Regel

7 Optimierungsverfahren

Nun wollen wir die zu den beiden konstruierten Shapegradienten gehörenden Optimierungsalgorithmen vorstellen. Wie schon Anfangs erwähnt wollen wir Gradientenverfahren benutzen um die Shapefunktion (2.2) zu minimieren. Genauer gesagt benutzen wir ein Newton-Verfahren mit Armijo-Schrittweitensteuerung. Dazu beschreiben wir kurz die Schrittweitenbestimmung mit der Armijo-Regel im \mathbb{R}^n .

7.1 Schrittweitensteuerung

Sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$ eine differenzierbare Funktion und sei $p \in \mathbb{R}^n$ eine Abstiegsrichtung mit $\nabla f \cdot p < 0$. Zusätzlich seien zwei Konstanten γ und β $0 < \gamma, \beta < 1$ gegeben. Dann ist eine Schrittweite $\tau > 0$ akzeptabel falls sie der Bedingung genügt :

$$f(x + \tau p) - f(x) < \gamma \tau \nabla f \cdot p \quad (7.1)$$

Anschaulich gesagt liegt $f(x + \tau p)$ unter der ‘‘nach oben gedrehten Tangente’’ $T = \{f(x) + t \nabla f \cdot p \mid t > 0\}$. Ist Bedingung (7.1) nicht erfüllt, dann wird eine neue Schrittweite τ_{neu} getestet.

$$\tau_{neu} = \beta \tau$$

Dieser Prozess liefert nach endlich vielen Schritten eine akzeptable Schrittweite τ . Diese Schrittweitenbestimmung ist für unseren Optimierungsprozess ausreichend. Für die Shapefunktion J lautet die Bedingung (7.1) für τ :

$$J(\Phi + \tau grad) - J(\Phi) < \gamma \tau < \frac{\partial J(\Phi)}{\partial \Phi}; grad > \quad (7.2)$$

Als Parameter haben wir $\gamma = 0.7$ und $\beta = 0.5$ gesetzt.

Damit kommen wir jetzt zur Beschreibung der Algorithmen.

7.2 Optimierungsalgorithmen

Zuerst stellen wir den Optimierungsalgorithmus vor, der Methode (5.2) aufgreift.

Algorithmus 7.1

- (i) Initialisierung der Levelsetfunktion Φ als signifizierte Distanzfunktion. Festlegen der fixierten Ränder und der Neumann- und Dirichletrandbedingungen. Die Anfangsschrittweite τ wird abhängig von der Gittertiefe gewählt. Bei Gittertiefe 6 wird sie 2 gesetzt.
- (ii) Lösen der Elastizitätsgleichung (\mathcal{P}_u) mit dem Mehrgitterlöser aus [29] .
- (iii) Berechnung des Shapegradienten $grad$ gemäß Methode (5.2) .
- (iv) Fortsetzen des Shapegradienten mit der Fast-Marching-Method aus Abschnitt 4.4 .
- (v) a) Update der Levelsetfunktion Φ

$$\Phi_{neu} = \Phi_{alt} + \tau grad$$
 - b) Lösen der Elastizitätsgleichung für Φ_{neu} .
 - c) Testen der Armijo-Regel.
 - d) Falls die Armijo-Regel nicht erfüllt ist wird τ verringert und Schritt (iv) wiederholt.
- (vi) Falls die Bedingung $\frac{1}{3} < \|\nabla\Phi\| < 3$ auf dem Rand $[\Phi = 0]$ verletzt ist, wird die Levelsetfunktion Φ , wie in Abschnitt 4.3 beschrieben, reinitialisiert. Dann muß die Elastizitätsgleichung erneut gelöst werden.
- (vii) Zurück zu Schritt (iii).

Dies ist eine rekursive Prozedur bei der die Schrittweite τ sukzessiv verringert wird, bis keine spürbare Variation des Gebietes mehr erfolgt. Dann kann man den Optimierungsprozess als beendet betrachten. Der zweite Optimierungsalgorithmus basiert auf Methode (5.3). Er unterscheidet sich nur geringfügig von Algorithmus (7.2).

Algorithmus 7.2

- (i) Initialisierung der Levelsetfunktion Φ als signifizierte Distanzfunktion. Festlegen der fixierten Ränder und der Neumann- und Dirichletrandbedingungen. Die Anfangsschrittweite τ wird abhängig von der Gittertiefe gewählt. Bei Gittertiefe 6 wird sie 2 gesetzt.
- (ii) Lösen der Elastizitätsgleichung (\mathcal{P}_u) mit dem Mehrgitterlöser aus [29] .

(iii) Berechnung des Shapegradienten $grad$ gemäß Methode (5.3) .

(iv) a) Update der Levelsetfunktion Φ

$$\Phi_{neu} = \Phi_{alt} + \tau grad$$

b) Lösen der Elastizitätsgleichung für Φ_{neu} .

c) Testen der Armijo-Regel.

d) Falls die Armijo-Regel nicht erfüllt ist wird τ verringert und Schritt (iv) wiederholt.

(v) Falls die Bedingung $\frac{1}{3} < \|\nabla\Phi\| < 3$ auf dem Rand $[\Phi = 0]$ verletzt ist, wird die Levelsetfunktion Φ , wie in Abschnitt 4.3 beschrieben, reinitialisiert. Dann muß die Elastizitätsgleichung erneut gelöst werden.

(vi) Zurück zu Schritt (iii).

8 Numerische Ergebnisse

Beginnen wollen wir die Präsentation unserer Ergebnisse mit der Modellierung eines kurzen Kranarmes. Dies ist motiviert durch die existierenden Ergebnisse in den Arbeiten [35] und [40]. Die dort betriebene Shapeoptimierung ist allerdings auf den 2-dimensionalen Fall beschränkt.

Im weiteren Verlauf sind die Lamé-Koeffizienten $\lambda = 70$ und $\mu = 40$ fest gewählt.

Der Optimierungsprozess ist beendet, wenn die Armijo-Regel keinen weiteren Iterationsschritt zuläßt und die Schrittweite nur noch kleiner wird.

8.1 Kurzer Kranarm

Initialgebiete sind ein Quader (Dimensionen $0.3 \times 0.3 \times 1$) ohne innere Löcher und ein Quader mit einem inneren Loch. Für beide gelten dieselben Randbedingungen; homogene Dirichlet-Randwerte auf der linken Seite und Neumann-Randwerte auf der rechten Seite, wie in Abbildung 8.1 dargestellt. Die Neumann-Randwerte sollen eine Kraft von einer Gewichtseinheit simulieren. Der Lagrangemultiplikator l für den Volumenterm ist 1,5 gewählt. An diesem Beispiel wollen wir die beiden Optimierungsverfahren gegenüberstellen und das Verhalten bei unterschiedlichen Initialgebieten analysieren.

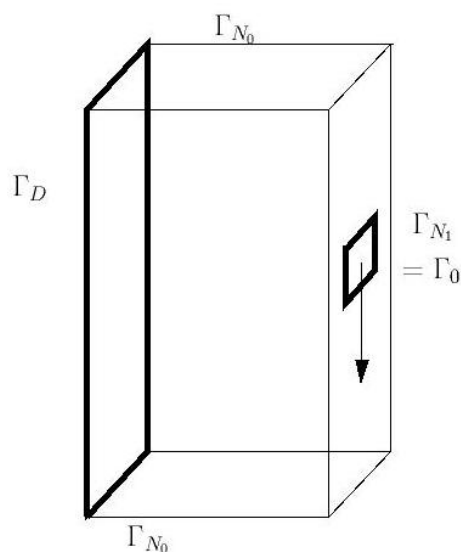
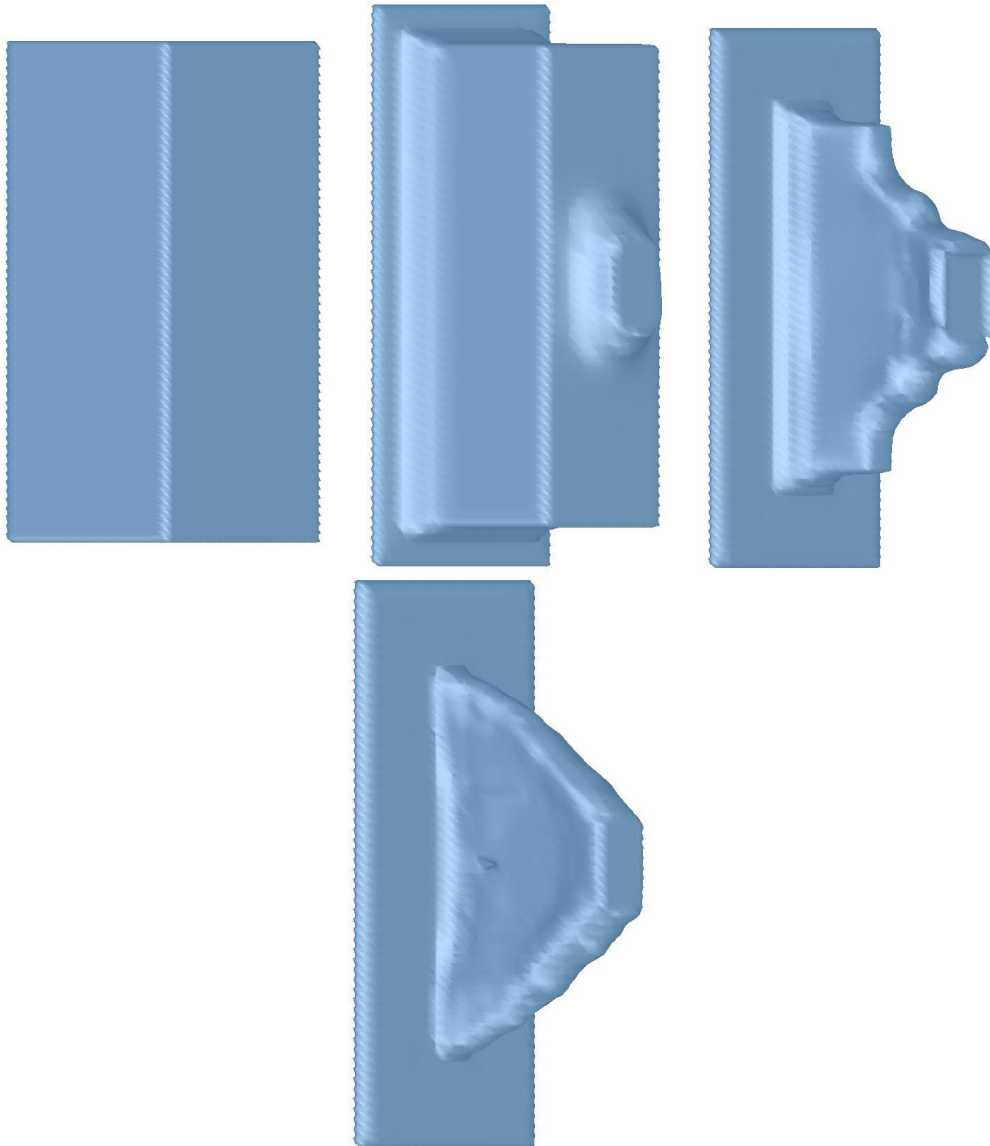


Abbildung 8.1: Konfiguration der Randwerte: Γ_0 fixierter Rand; Γ_{N_1} inhomogene Neumann-Randwerte; Γ_{N_0} homogene Neumann-Randwerte; Γ_D homogene Dirichlet-Randwerte

Algorithmus 7.1 (Gradient auf dem Isolevel) für ein Initialgebiet ohne Löcher**Abbildung 8.2:** Initialgebiet und Iterationen 2, 5 und 7

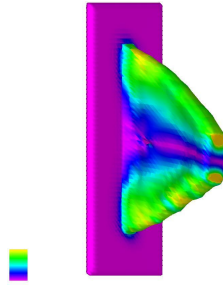


Abbildung 8.3: Die Deformation des resultierenden Körpers mit farblicher Visualisierung der 'Von Mises'-Spannung. (lila = 0 , gelb = 0.7)

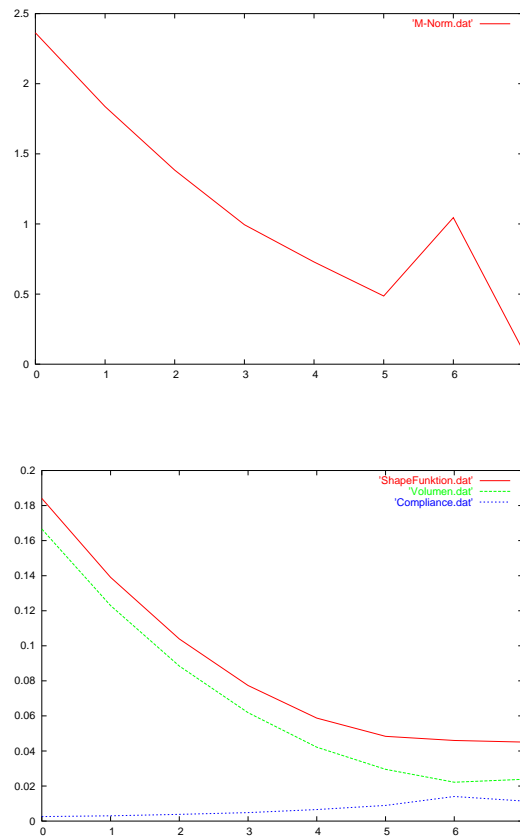
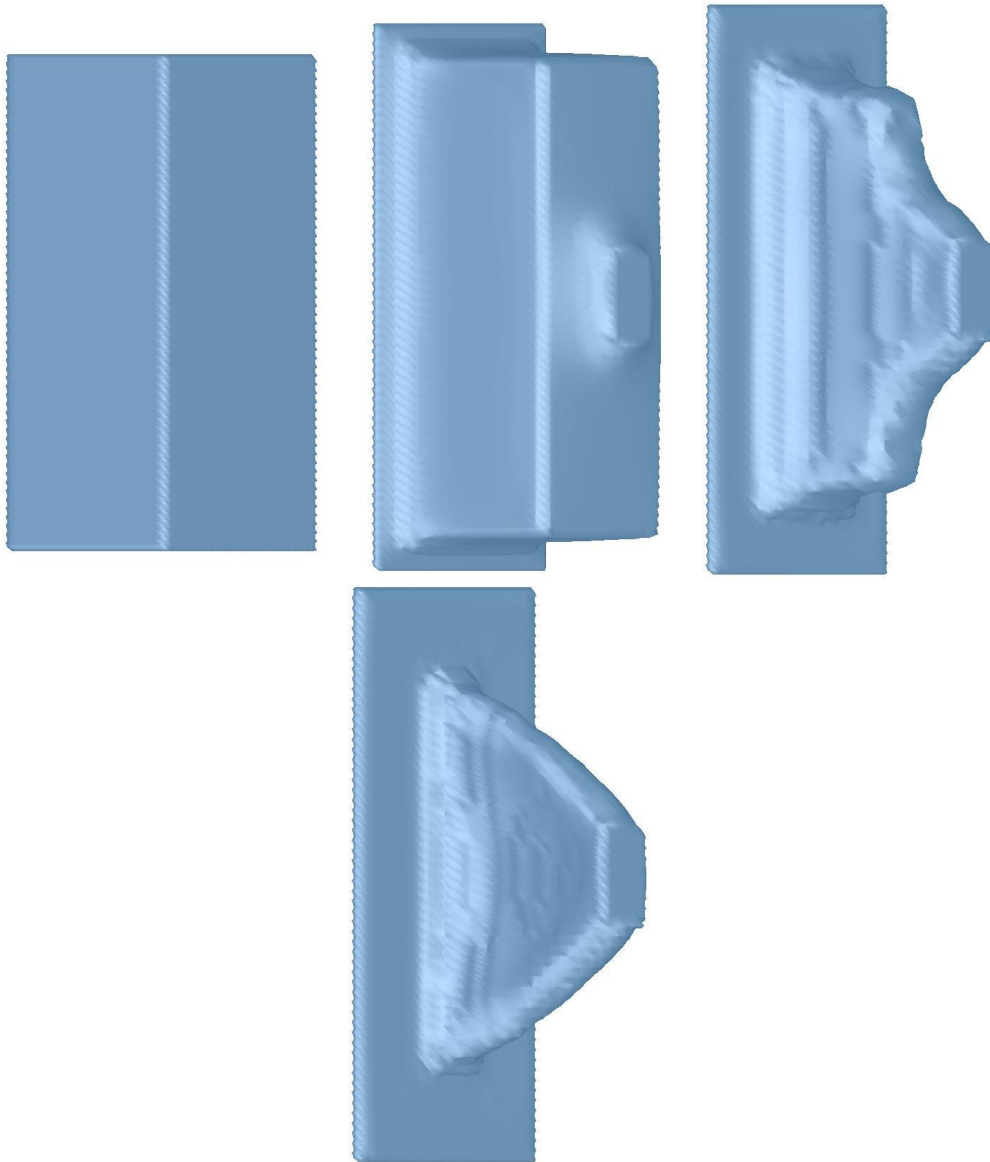


Abbildung 8.4: oben: L^2 -Norm des Gradienten auf dem Interface; unten: Verlauf der Shapefunktion (rot) , des Volumens (grün) und der Spannungsenergie (blau)

Algorithmus 7.2 (regularisierter Gradient) für Initialgebiet ohne Löcher**Abbildung 8.5:** Iterationen 0,2, 5 und 8

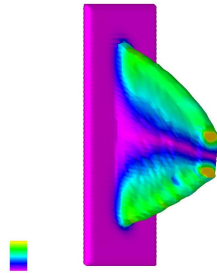


Abbildung 8.6: Die Deformation des resultierenden Körpers mit farblicher Visualisierung der 'Von Mises'-Spannung.(lila = 0 , gelb = 0.7)

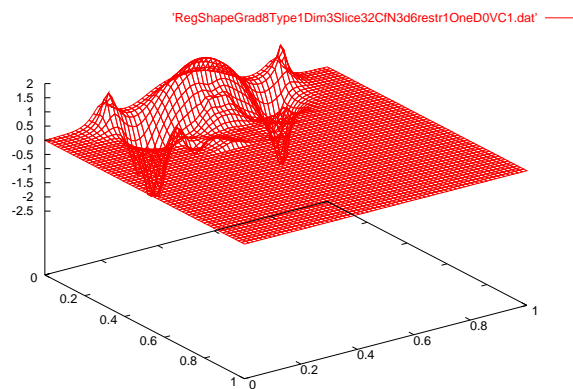


Abbildung 8.7: Querschnitt (x-, z-Achse) des regularisierten Gradienten zum Iterationsgebiet 5

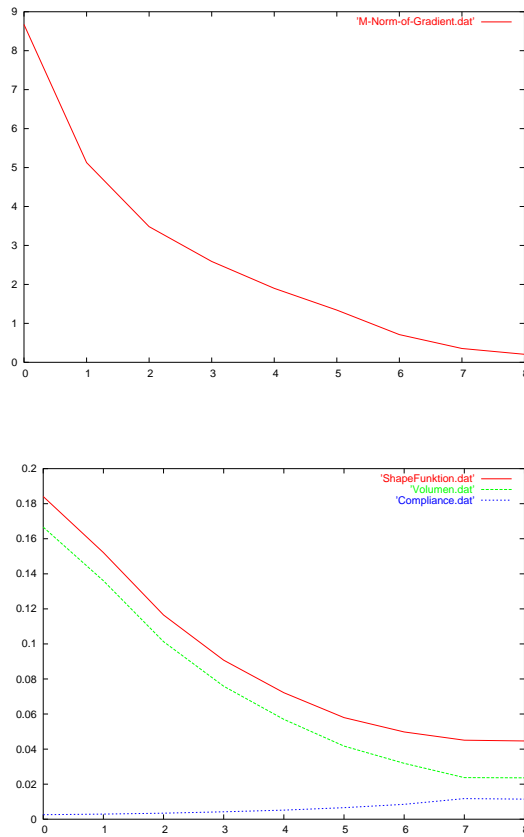


Abbildung 8.8: oben: Norm des regularisierten Gradienten; unten: Verlauf der Shapefunktion (rot), des Volumens (grün) und der Spannungsenergie (blau)

Algorithmus 7.1 (Gradient auf dem Isolevel) für ein Initialgebiet mit einem Loch

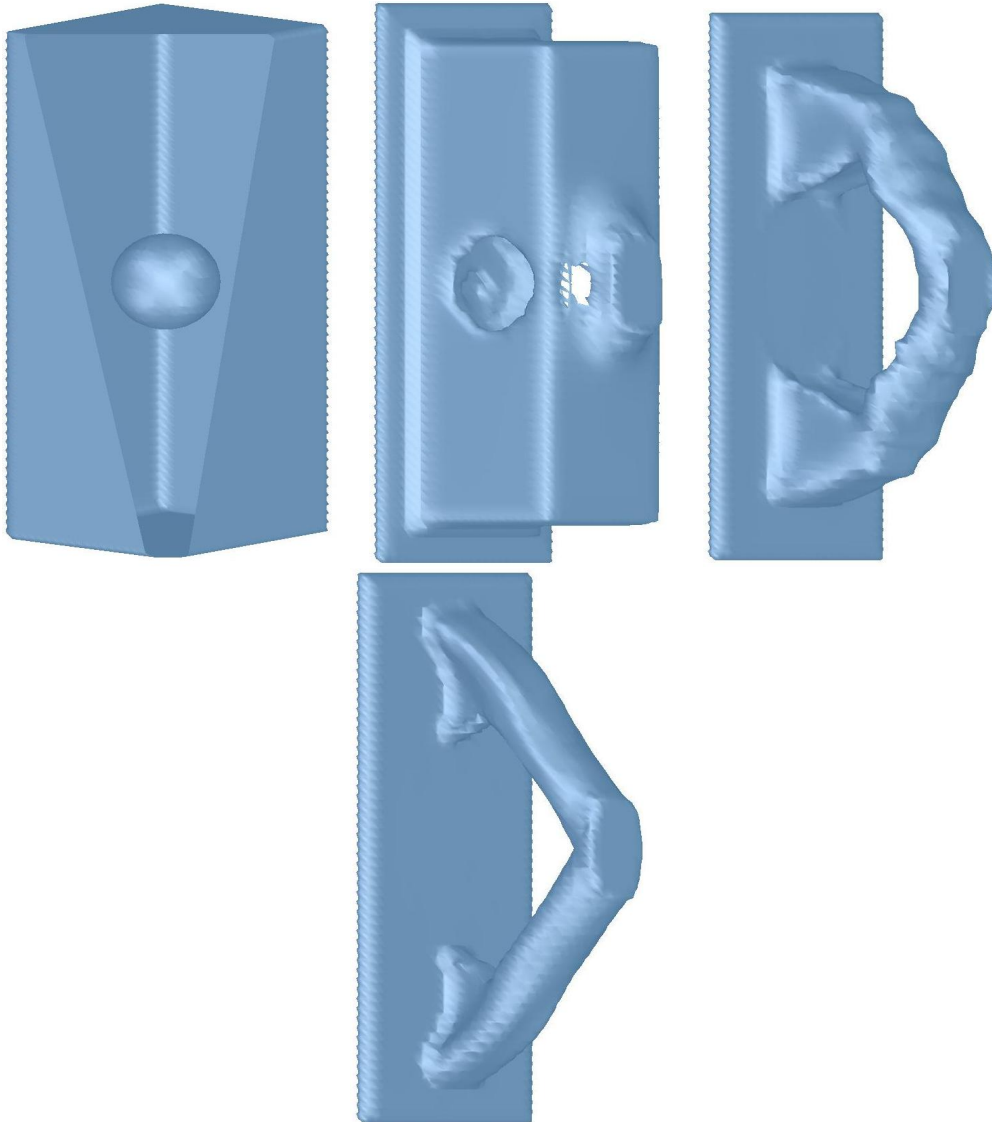


Abbildung 8.9: Iterationen 0, 2, 5 und 11

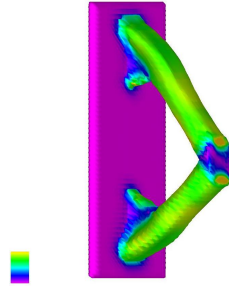


Abbildung 8.10: Die Deformation des resultierenden Körpers mit farblicher Visualisierung der 'Von Mises'-Spannung. (lila = 0 , gelb = 0.7)

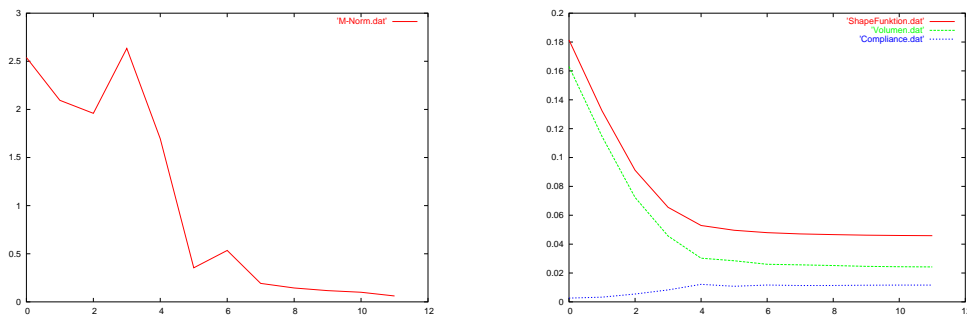


Abbildung 8.11: oben: L^2 -Norm des Gradienten; unten: Verlauf der Shapefunktion (rot), des Volumens (grün) und der Spannungsenergie (blau). Die Ausschläge der Norm sind zurückzuführen auf das 'Aufbrechen' der Struktur, wobei die Struktur lokal sehr dünn werden kann.

Algorithmus 7.2 (regularisierter Gradient) für ein Initialgebiet mit einem Loch

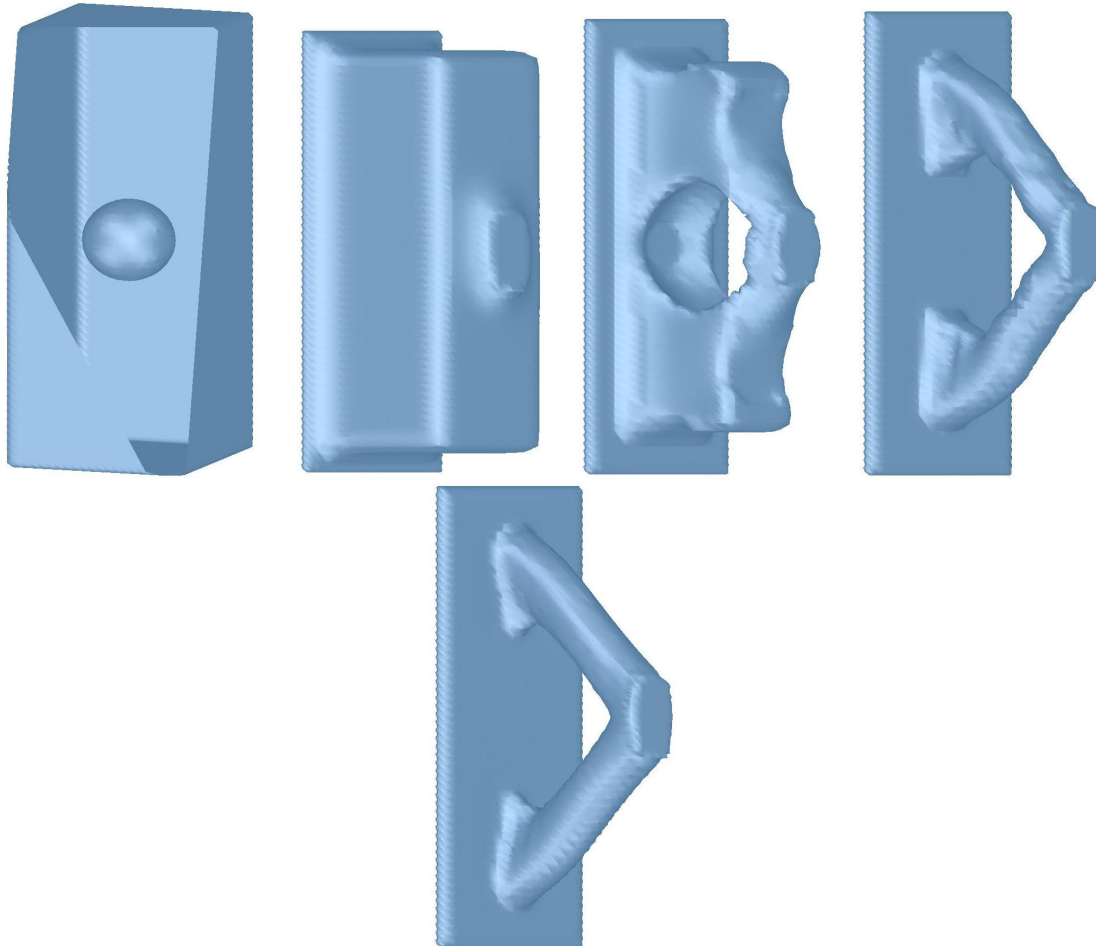


Abbildung 8.12: Initialgebiet und Iterationsgebiete 2, 5, 10 und 16

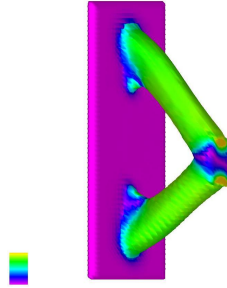


Abbildung 8.13: Die Deformation des resultierenden Körpers mit farblicher Visualisierung der 'Von Mises'-Spannung. (lila = 0 , gelb = 0.7)

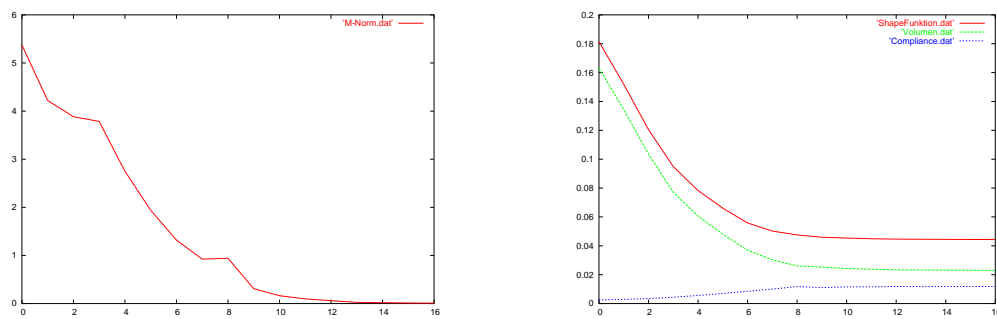


Abbildung 8.14: links: M-Norm des regularisierten Gradienten; rechts: Verlauf der Shapefunktion (rot) , des Volumens (gruen) und der Spannungsenergie (blau)

8.2 Langer Kranarm

Initialgebiet ist ein perforierter Quader (Dimensionen $1 \times 0.3 \times 0.6$) mit den Randbedingungen gemäß Abbildung 8.15 .

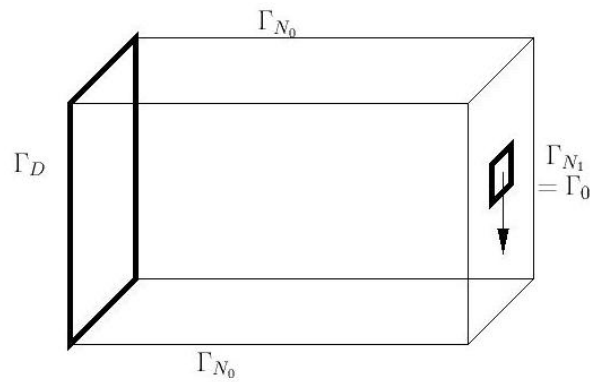


Abbildung 8.15: Konfiguration der Randwerte: Γ_0 fixierter Rand; Γ_{N_1} inhomogene Neumann-Randwerte; Γ_{N_0} homogene Neumann-Randwerte; Γ_D homogene Dirichlet-Randwerte

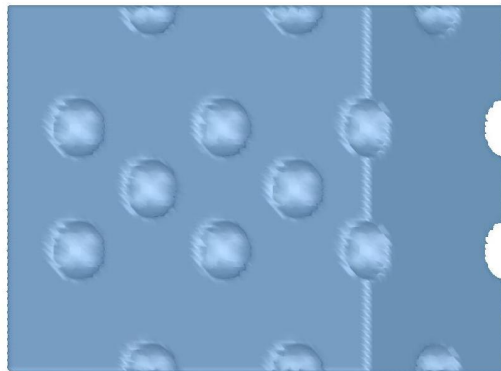


Abbildung 8.16: Initialgebiet

Algorithmus 7.2 (regularisierter Gradient)**Konfiguration 1**

Die Neumann-Randwerte repräsentieren eine Kraft von 0.3 Gewichtseinheiten . Der Lagrangemultiplikator l für den Volumenterm ist 0.2 gewählt.

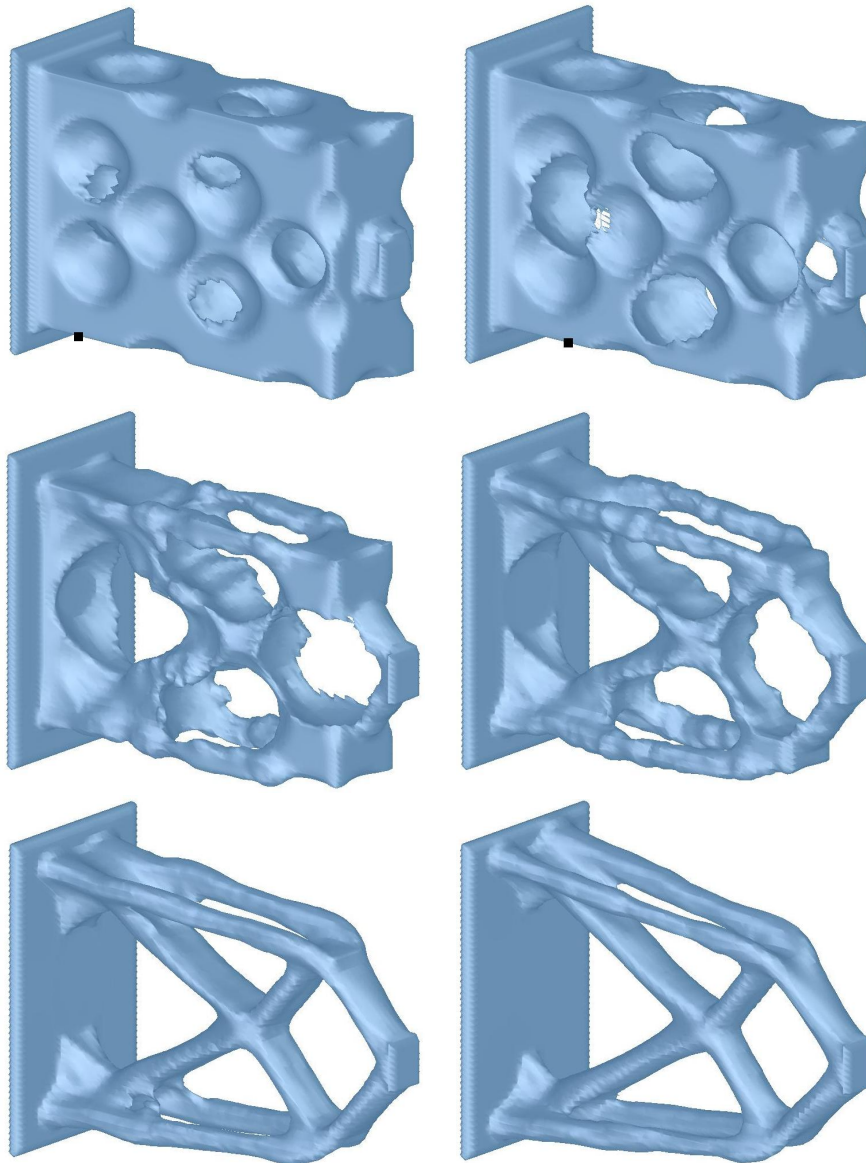


Abbildung 8.17: Iterationsgebiete 2, 3, 5, 7, 10 und 14

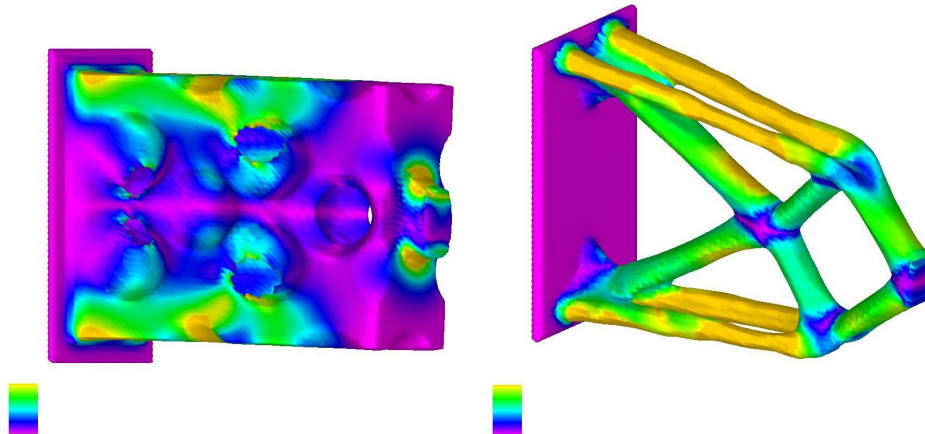


Abbildung 8.18: Die Deformation des 2-ten Iterations-Struktur und der resultierenden Struktur mit farblicher Visualisierung der 'Von Mises'-Spannung. (lila = 0, gelb = 0.5)

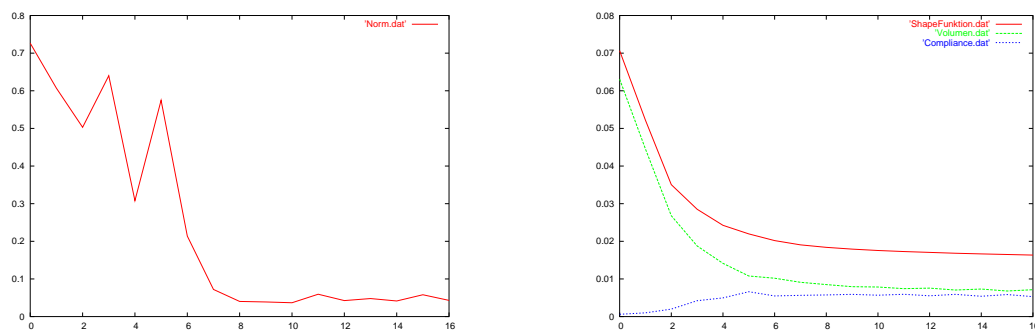


Abbildung 8.19: oben: Norm des regularisierten Gradienten; unten: Verlauf der Shapefunktion (rot), des Volumens (grün) und der Compliance-Energie (blau). Die Ausschläge der Norm sind zurückzuführen auf das 'Aufbrechen' der Struktur, wobei die Struktur lokal sehr dünn sein kann.

Algorithmus 7.2 (regularisierter Gradient)**Konfiguration 2**

Die Neumann-Randwerte repräsentieren eine Kraft von 0.6 Gewichtseinheiten . Der Lagrangemultiplikator l für den Volumenterm ist 0.2 gewählt.

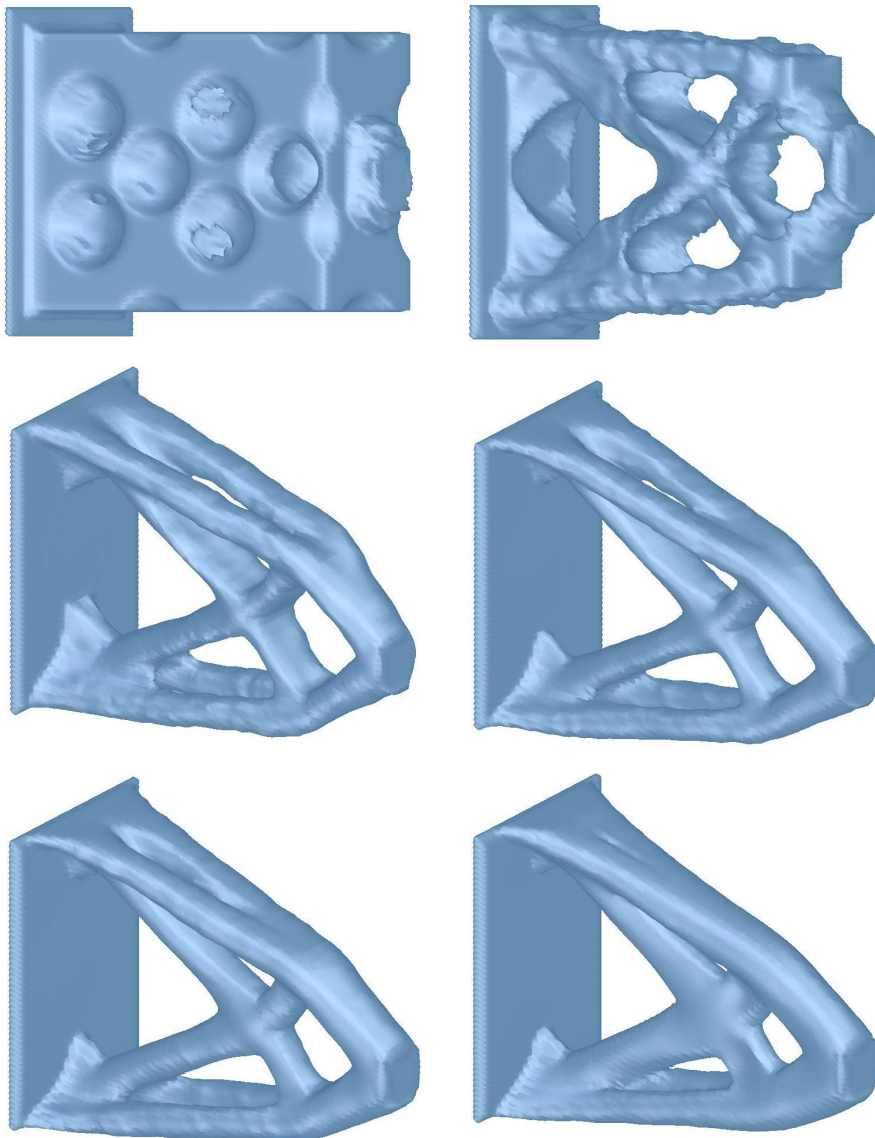


Abbildung 8.20: Iterationsgebiete 2, 5, 10, 15, 20 und 36

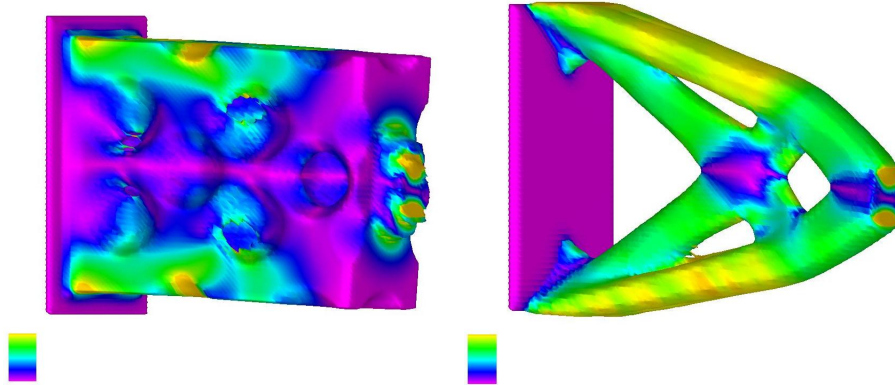


Abbildung 8.21: Die Deformation des 2-ten Iterations-Struktur und der resultierenden Struktur mit farblicher Visualisierung der 'Von Mises'-Spannung. (lila = 0, gelb = 0.5)

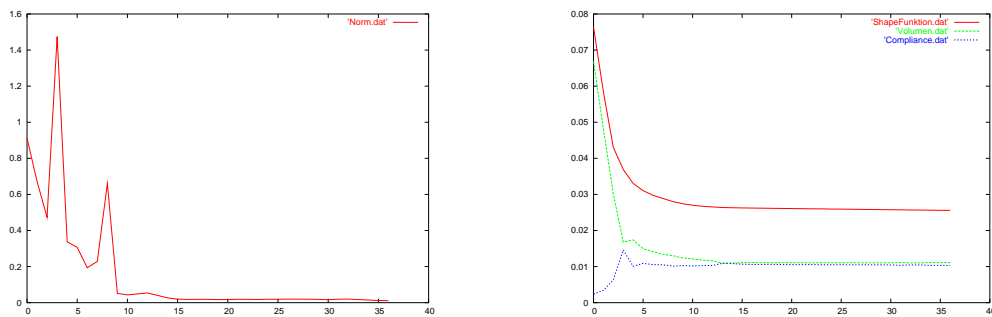


Abbildung 8.22: oben: Norm des regularisierten Gradienten; unten: Verlauf der Shapefunktion (rot), des Volumens (grün) und der Compliance-Energie (blau). Die Ausschläge der Norm sind zurückzuführen auf das 'Aufbrechen' der Struktur, wobei die Struktur lokal sehr dünn sein kann.

8.3 Würfel

Initialgebiet ist ein perforierter Würfel (Dimensionen $1 \times 1 \times 1$) mit den Randbedingungen gemäß Abbildung 8.15 . Die Neumann-Randwerte repräsentieren eine Kraft von 8 Gewichtseinheiten . Der Lagrangemultiplikator l für den Volumenterm ist 1.2 gewählt.

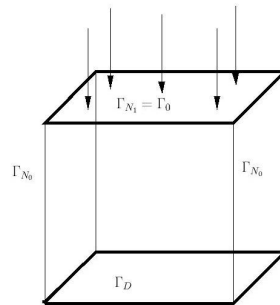


Abbildung 8.23: Konfiguration der Randwerte: Γ_0 fixierter Rand; Γ_{N_1} inhomogene Neumann-Randwerte; Γ_{N_0} homogene Neumann-Randwerte; Γ_D homogene Dirichlet-Randwerte

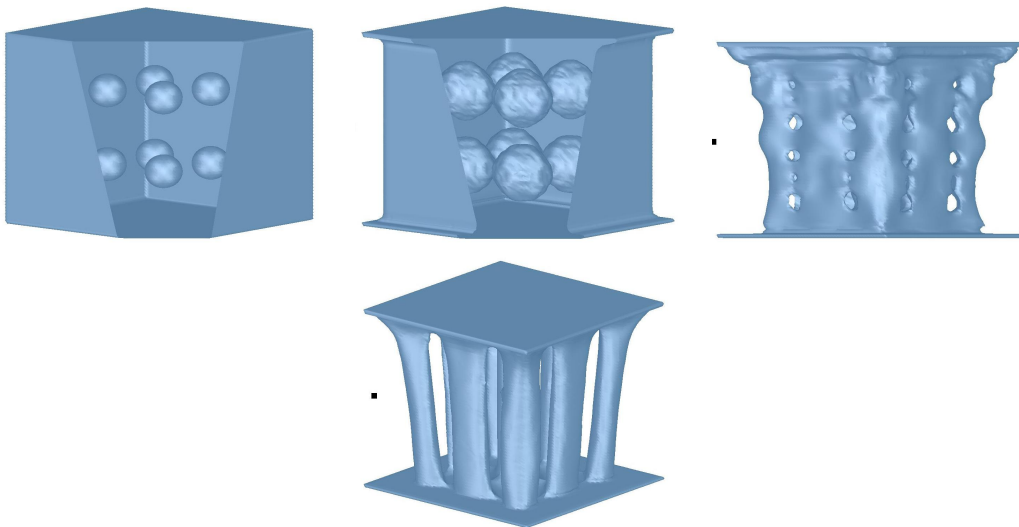


Abbildung 8.24: Initialgebiet und Iterationen 2, 5 und 10

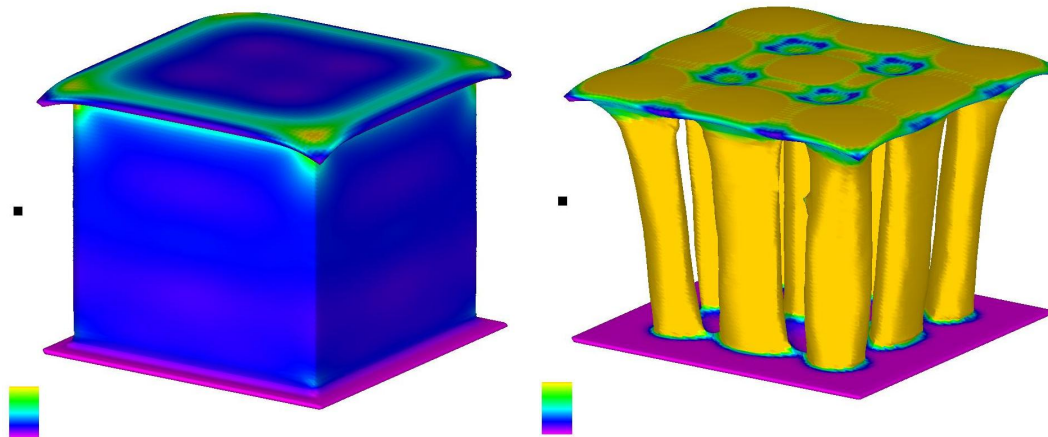


Abbildung 8.25: Die Deformation der 2-ten Iterationsstruktur und der resultierenden Struktur mit farblicher Visualisierung der 'Von Mises'-Spannung. (lila = 0, gelb = 0.5)

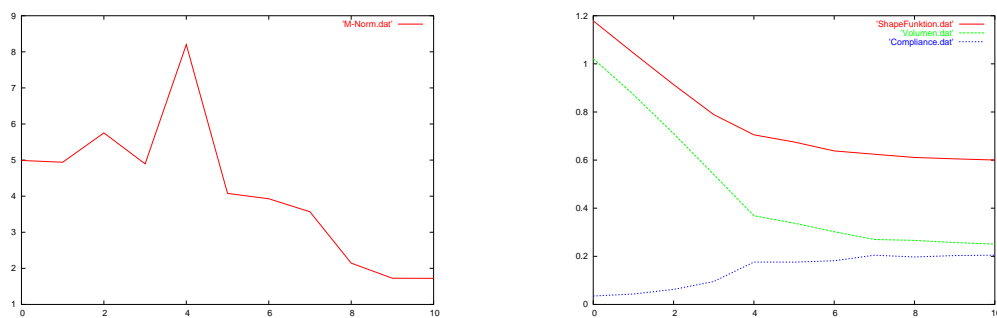


Abbildung 8.26: links: Norm des regularisierten Gradienten; rechts: Verlauf der Shapefunktion (rot), des Volumens (gruen) und der Spannungsenergie (blau)

9 Schlussbemerkung

Wie die numerischen Ergebnisse zeigen ist der Shape-Optimierungsprozess sehr sensibel. Das Beispiel des kuzen Kranarmes zeigt den Einfluss der Wahl eines Initialisierungsgebietes auf das Ergebnis, welches unser Optimierungsalgorithmus liefert.

Viel größeren Einfluß auf das Verhalten des Optimierungsalgorithmuses haben aber die Konfiguration der wirkenden Kräfte und des Lagrangemultiplikators l für den Volumenterm. Bei der Wahl eines zu großen Lagrangemultiplikators kann die Struktur des Gebietes lokal so fein (teilweise im Bereich einer Gitterweite) werden, dass sie von der Diskretisierung nicht mehr genügend aufgelöst wird. Dann bekommt die Matrix der diskreten Elastizitätsgleichung eine komplizierte Struktur und der Löser liefert kein Ergebnis mehr. Anders herum bewirkt die Wahl einer zu großen Kraft kaum Änderungen des Initialgebietes oder ein Verdicken der Struktur.

Ein 'Herantasten' an eine optimale Struktur und einer guten Konfiguration der Parameter durch Rechnen auf einem gröberem Gitter ist eine Möglichkeit die oben erwähnten Effekte zu vermeiden. Dies ist aber nicht immer möglich. Das Beispiel des langen Kranarmes ist hierfür ein Beispiel. Die Komplexität der Struktur ist auf Gitterlevel 5 nicht zu erreichen. Es wäre hilfreich vorab eine Aussage über die Topologie zu haben, welche zu erwarten ist. Dazu könnte man sich der Methoden der Topologischen Optimierung bedienen.

Bemerkenswert ist das Ergebnis für den langen Kranarm aus Abschnitt 8.2 mit Konfiguration 2. Der Algorithmus schafft es die entstandenen Tragarme wieder zu verschmelzen.

Abschließend lassen die Ergebnisse des Optimierungsprozesses folgende allgemeine Interpretation zu. Im Verlauf des Algorithmus nimmt mit zunehmender Herausbildung einer Struktur die aufgenommene Spannung zu, was die Graphen des Spannungsverlaufs belegen. Es findet aber eine gleichmäßige Verteilung der Spannung statt, wie in den Visualisierungen der Deformationen und Spannungen zu sehen ist.

Die Möglichkeiten mit weiteren Skalarprodukten andere Abstiegsrichtungen zu erzeugen sind noch nicht ausgeschöpft. Man könnte zum Beispiel das Skalarprodukt aus Methode 5.3 ersetzen durch eine geometrische Regularisierung :

$$\langle \varphi, \psi \rangle_M := \int_{\Omega} \|\nabla \phi\| (\varphi \psi + \frac{\sigma^2}{2} P(\nabla \varphi) \cdot \nabla \psi) dx$$

Wobei P die Projektion auf den Tangentialraum ist.

Aus zeitlichen Gründen haben wir dieses nicht mehr in Angriff genommen.

Danksagung

Bedanken möchte ich mich bei Herrn Prof. Dr. Martin Rumpf für die sehr gute Betreuung bei der Anfertigung dieser Arbeit. Desweiteren bedanke ich mich bei Florian Liehr, Mark Droske, Martin Lenz und Nadine Olischläger, die mir bei Fragen zur Implementierung stets zur Verfügung standen. Besonders die Diskussionen mit Mark Droske waren vor allem in der Anfangsphase dieser Arbeit sehr hilfreich.

Literatur

- [1] B. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *J. Comput. Phys.*, 148:2–22, 1999.
- [2] G. Allaire. *Shape optimization by the homogenization method*. Springer-Verlag, New York, 2001.
- [3] G. Allaire, F. Jouve, and A. Toader. Structural optimization using sensitivity analysis and a level-set method. *J. Comput. Phys.*, 194:363–393, 2004.
- [4] G. Allaire, F. Jouve, and A. M. Toader. A level-set method for shape optimization. *C. R. Acad. Sci. Paris Sér. I*, 334:1125–1130, 2002.
- [5] L. Amrosio and G. Buttazzo. An optimal design problem with perimeter penalization. *Calc. Var.*, 1:55–69, 1993.
- [6] D. Az and G. Buttazzo. Some remarks on the optimal design of periodically reinforced structures. *Math. Mod. Num. Anal.*, 23(1):53–61, 1989.
- [7] M. Bendsoe and O. Sigmund. *Topology Optimization. Theory, Methods, and Applications*. Springer, New York, 2003.
- [8] G. Bouchitté and G. Buttazzo. Characterization of optimal shapes and masses through Monge-Kantorovich equation. *J. Eur. Math. Soc.*, 3:139–168, 2001.
- [9] D. Bucur. An Existence Result for Optimal Obstacles. *J. Func. Anal.*, 162:96–119, 1999.
- [10] D. Bucur, G. Buttazzo, and N. Varchon. On the Problem of optimal cutting. *SIAM J. Optim.*, 13(1):157–167, 2002.
- [11] M. Burger. A framework for the construction of level set methods for shape optimization and reconstruction. *Interfaces and Free Boundaries*, 5:301–329, 2003.
- [12] M. Burger and S. Osher. A Survey on Level Set Methods for Inverse Problems and Optimal Design. Under consideration for publication in *Euro. J. Appl. Math.*, 2004.
- [13] G. Buttazzo. Some Relaxation Problems in Optimal Control Theory. *J. Math. Anal. Appl.*, 125:273–287, 1987.
- [14] G. Buttazzo. On the existence of minimizing domains for some shape optimization problems. volume 3, pages 51–64. ESAIM, 1998.
- [15] G. Buttazzo, M. E. Drakhlin, L. Freddi, and E. Stepanov. Homogenization of Optimal Control Problems for Functional Differential Equations. *J. Optim. Appl.*, 93(1):103–119, 1997.

-
- [16] G. Buttazzo and P. Guasoni. Shape Optimization Problems over Classes of Convex Domains. *J. Conv. Anal.*, 4(2):343–351, 1997.
- [17] G. Buttazzo and G. D. Maso. Shape Optimization for Dirichlet Problems: Relaxed Formulation and Optimality Conditions. *Appl. Math. Optim.*, 23:17–49, 1991.
- [18] G. Buttazzo and G. D. Maso. An Existence Result for a Class of Shape Optimization Problems. *Arch. Rational Mech. Anal.*, 122:183–195, 1993.
- [19] G. Buttazzo and P. Trebeschi. The role of monotonicity in some shape optimization problems. *mn*, 122:183–195, 1993.
- [20] J. C ea. Conception optimale ou identification de formes, calcul rapide de la d riv e directionnelle de la fonction co t. *Math. Model. Num. Anal.*, 20:371–402, 1986.
- [21] Chapelle and Bathe. *The Finite Element Analysis of Shells - Fundamentals*. Springer.
- [22] P. G. Ciarlet. *Mathematical Elasticity, Volume I: Three-Dimensional Elasticity*. North-Holland, Amsterdam, 1988.
- [23] M. C. Delfour and J.-P. Zol sio. Structure of shape derivatives for nonsmooth domains. *J. Funct. Anal.*, 104:1–33, 1992.
- [24] M. C. Delfour and J.-P. Zol sio. *Shapes and Geometries. Analysis, Differential Calculus, and Optimization*. Siam, Philadelphia, 2001.
- [25] M. Droske and W. Ring. A Mumford-Shap level set approach for geometric image registration. July 2004.
- [26] W. Hackbush and S. A. Sauter. Composite finite element approach for the approximation of PDEs on domains with complicated micro-structures. *Numer. Math.*, 75 no.4:447–472, 1997.
- [27] W. Hackbush and S. A. Sauter. New finite element approach for problems containing small geometric details. *Arch. Math.*, 34:105–107, 1998.
- [28] K. K onigsberger. *Analysis 2*. Springer, Berlin, 2002.
- [29] F. Liehr. Ein effizienter L oser f ur elastische Mikrostrukturen. Master’s thesis, Universit t Duisburg-Essen, 2004.
- [30] F. Murat and S. Simon. Etudes de probl mes d’optimal design. volume 41, pages 54–62. Springer, 2000.
- [31] S. Osher and R. P. Fedkiw. Level Set Methods : An Overview and Some Recent Results. *J. Comput. Phys.*, 169:463–502, 2001.

-
- [32] S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed. Algorithms based on the Hamilton-Jacobi formulation. *J. Comput. Phys.*, 79:12, 1988.
- [33] O. Pironeau. *Optimal shape design for elliptic systems*. Springer-Verlag, New York, 1984.
- [34] J. Sethian. *Level-Set Methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision and materials science*. Cambridge University Press, 1999.
- [35] J. Sethian and A. Wiegmann. Structural boundary design via level-set and immersed interface methods. *J. Comp. Phys.*, 163:489–528, 2000.
- [36] J. A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge Univ. Press, 1996.
- [37] J. Simon. Differentiation with respect to the domain in boundary value problems. *Num. Funct. Anal. Optimz.*, 2:649–687, 1980.
- [38] J. Sokolowski and J.-P. Zolésio. *Introduction to Shape Optimization*. Springer-Verlag, Berlin, 1992.
- [39] M. Wang, X. Wang, and D. Guo. A level-set method for structural topology optimization. *Comput. Methods Appl. Mech. Engrg.*, 192:227–246, 2003.
- [40] M. Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Comp. Methods Appl. Mech. Enggrg*, 192:227–246, 2003.
- [41] R. Warnke. *Schnelle Löser für elliptische Randwertprobleme mit springenden Koeffizienten*. PhD thesis, Zürich, 2003. Dissertation.

Erklärung

Hiermit versichere ich, dass ich diese Diplomarbeit selbständig und nur unter Gebrauch der angegebenen Hilfsmittel und Quellen angefertigt habe.

Duisburg, Oktober 2005