

# 1 BFGS-Algorithmus in der Formoptimierung

## 1.1 Shape Spaces und Formgradienten

habe ich sicher das Kapitel zu  $\mathbb{R}^n$  BFGS? falls nicht korrigiere text soll ich 2 dimensional od  
statt jedes mal die voraussetzunge zu überprüfen, also zsmhgd, beschr,... nutze lieber vllt ein

Notation: einmal Gradient  $H$ , dann später  $U$ ... mache das einheitlich! Nachdem wir in die Grundlagen der Formoptimierung und des BFGS-Algorithmus im endlich-dimensionalen eingeführt haben, möchten wir nun in diesem Abschnitt beide Bereiche zusammenführen und die endlich-dimensionalen BFGS-Updates in den Kontext von Formen abstrahieren. Dies ist nicht ohne weiteres möglich, da a priori nicht klar ist, wie Gradienten von Formen definiert werden sollen, insbesondere deshalb, weil wir uns in unendlich-dimensionalen Shape-space befinden. Erschwert wird dies weiter dadurch, dass, wie eingangs bemerkt, keine natürliche Vektorraumstruktur auf dem betrachteten Shape-space vorhanden ist. Ziel dieses Abschnittes wird es sein zunächst die Wahl eine geeigneten Metrik im Sinne der Riemannschen Geometrie zu wählen, wodurch die Definition eines Gradienten überhaupt erst möglich wird. Im Abschnitt über BFGS-Methoden im  $\mathbb{R}^n$  haben wir zur Darstellung solcher Gradienten stets stillschweigend die euklidische Metrik vorausgesetzt. Diese lässt sich offensichtlich nicht einfach im Shape-setting verwenden. Anschließend übertragen wir den BFGS-Algorithmus in das Shape-setting. In diesem Abschnitt halten wir uns vorallem an [10], sowie an [14] und [17].

Zunächst definieren wir, was wir unter dem Raum aller Formen verstehen, vgl. [10]. Hierzu bleiben wir in zwei Dimensionen, da hier schon die wesentlichen Elemente und Zusammenhänge klar werden. Prinzipiell ist ein betrachten von höherdimensionalen Objekten auch möglich, sofern die zugrundeliegende Topologie der Formen beachtet wird.

**Definition 1** (Shape-space für den  $\mathbb{R}^2$ ). Bezeichne mit  $\text{Emb}(S^1, \mathbb{R}^2)$  die Menge aller  $C^\infty$  Einbettungen von  $S^1$  in den  $\mathbb{R}^2$ , und mit  $\text{Diff}(S^1)$  die Menge aller Diffeomorphismen von  $S^1$  in sich selber. Dann heißt der Quotientenraum

$$B_e(S^1, \mathbb{R}^2) := \text{Emb}(S^1, \mathbb{R}^2) / \text{Diff}(S^1)$$

Shape-space für den  $\mathbb{R}^2$ .

Man sieht, dass die Elemente von  $B_e(S^1, \mathbb{R}^2)$  Äquivalenzklassen sind. In ihnen sind jeweils unter anderem Umparametrisierungen der selben geschlossenen  $C^\infty$ -Kurven  $c : \mathbb{R} \rightarrow \mathbb{R}^2$  enthalten. Das bedeutet, dass ein Punkt in

## 1 BFGS-Algorithmus in der Formoptimierung

$B_e(S^1, \mathbb{R}^2)$  als eine geschlossene, geometrische Kurve im  $\mathbb{R}^2$  interpretiert werden kann. Betrachtet man nun eine über eine beschränkte Menge  $\Omega \subset \mathbb{R}^2$  mit Lipschitz-Rand definierte Hold-all-Domain, so lassen sich Ränder  $\partial\Omega_2$  von kompakten, beschränkten, zusammenhängenden Mengen  $\Omega_2 \subset \Omega$  mit  $C^\infty$ -Rand genau mit solchen geschlossenen Kurven identifizieren. Zudem gilt außerdem, dass  $B_e(S^1, \mathbb{R}^2)$  eine Mannigfaltigkeit bildet, siehe [10], was wesentlich von der Glattheit der Einbettungen abhängt.

Wir fahren fort mit unserer Konstruktion, und geben hier eine Darstellung des Tangentialbündels auf  $B_e(S^1, \mathbb{R}^2)$ . Da wir hier explizit die Strukturen des  $\mathbb{R}^2$  und  $B_e(S^1, \mathbb{R}^2)$  für die Darstellung der Tangentialräume verwenden, handelt es sich bei unserer Definition um zu den üblichen Tangentialräumen der Differentialgeometrie isomorphe Objekte. Für eine Definition der klassischen Tangentialräume, sowie eine tiefgreifende Einführung in die Differentialgeometrie, empfehlen wir [3], Kapitel 3.

**Definition 2** (Tangentialbündel). Sei  $B_e(S^1, \mathbb{R}^2)$  der Shape-space für den  $\mathbb{R}^2$ . Betrachte einen Repräsentant  $c : S^1 \rightarrow \mathbb{R}^2$  eines Punktes in  $B_e(S^1, \mathbb{R}^2)$ , sowie das zugehörige äußere Normalenvektorfeld  $n$  der mit  $c$  identifizierten Form  $\partial\Omega_2$ . Dann gilt für den *Tangentialraum*  $T_c B_e$

$$T_c B_e \cong \{h : h = \alpha n \text{ für } \alpha \in C^\infty(\Omega_2, \mathbb{R})\}.$$

Diese Darstellung gilt, da die Quotientenstruktur von  $B_e$  die Isomorphie der Immersionen von  $S^1$  nach  $\mathbb{R}^2$  mit den  $C^\infty$ -Funktionen von  $S^1$  nach  $\mathbb{R}^2$  vererbt, für Details siehe [14], Kapitel 3. Somit ermöglicht diese isomorphe Darstellung es uns, Tangentialvektoren  $v \in T_c B_e$  mit Hilfe von  $C^\infty(\Omega_2, \mathbb{R})$ -Funktionen zu beschreiben. Das nutzen wir aus, um auf  $B_e$  eine geeignete riemannsche Metrik zu definieren.

### Diagramm ??

Es gibt verschieden Möglichkeiten riemannsche Metriken auf dem Raum  $B_e$  zu definieren, je nachdem auf welche künftige Anwendung man abzieht. Laut [14] scheint eine Sobolev-Metrik, definiert mittels des Laplace-Beltrami-Operators, natürlich zu erscheinen. Der Nachteil ist, das erheblicher Rechenaufwand beim Ermitteln des Gradienten aus einer Formableitung entsteht. Aus diesem Grund führen die Autoren von [10] eine riemannsche Metrik auf Grundlage der sogenannten Dirichlet-zu-Neumann Abbildung ein, weshalb wir uns Ausführungen zu Sobolev-Metriken an dieser Stelle sparen, und verweisen für diese auf [10] und [14]. Im Folgenden führen wir eine sogenannte Stekolov-Poincaré-Metrik ein, und verweisen für die hierzu nötigen Grundlagen bei Differentialgleichun-

## 1 BFGS-Algorithmus in der Formoptimierung

gen und Sobolev-Slobodeckij-Räumen auf **REF SLOBODECKIJ UND PDE**. Zunächst definieren wir die nötigen Abbildungen, vgl. [10].

**Definition 3** (Verallgemeinerte Spurabbildung). Sei  $\Omega \subset \mathbb{R}^n$  eine beschränktes, offenes Gebiet. Sei  $\Omega_2 \subset \Omega$  zusammenhängend, offen und habe einen Lipschitzrand, und  $\partial\Omega_2$  sei eine Form in diesem Gebiet. Dann heißt die Abbildung

$$\gamma : H_0^1(\Omega, \mathbb{R}^n) \rightarrow H^{1/2}(\Omega, \mathbb{R}^n) \times H^{-1/2}(\Omega, \mathbb{R}^n)$$

$$U \mapsto \begin{pmatrix} \gamma_0 U \\ \gamma_1 U \end{pmatrix} := \begin{pmatrix} U|_{\partial\Omega_2} \\ \partial_n U|_{\partial\Omega_2} \end{pmatrix},$$

verallgemeinerte Spurabbildung, wobei  $\partial_n U|_{\partial\Omega_2}$  die Ableitung von  $U$  auf der Form  $\partial\Omega_2$  in Richtung des äußeren Normalenvektorfeldes  $n$  ist. **vielleicht einfach die Standard n**

Wir erinnern an dieser Stelle, dass diese Abbildung wohldefiniert ist, siehe **REFERENZ SPURSÄTZE**. Nun geben wir uns eine symmetrische, koerzive Bilinearform  $a : H^1(\Omega, \mathbb{R}^n) \times H^1(\Omega, \mathbb{R}^n) \rightarrow \mathbb{R}$  vor. Diese werden wir später gemeinsam mit der zu definierenden Riemannschen Metrik verwenden, um aus gegebenen Formableitungen Gradienten zu konstruieren. Gegeben einer solchen Bilinearform  $a$  definieren wir die zugehörigen Lösungsoperatoren zu folgendem Variationsproblem.

**Notation: alpha oder u ???!!! n den folgenden Definitionen**

**Definition 4** (Lösungsoperatoren). Sei  $\Omega \subset \mathbb{R}^n$  offen, zusammenhängend, beschränkt, und sei  $\Omega_2 \subset \Omega$  offen, zusammenhängend und habe einen  $C^\infty$ -Rand  $\partial\Omega_2$ . Weiterhin sei  $a : H^1(\Omega, \mathbb{R}^n) \times H^1(\Omega, \mathbb{R}^n) \rightarrow \mathbb{R}$  eine koerzive, symmetrische und **stetige** Bilinearform. Dann heißt der Operator

$$E_N : H^{-1/2}(\partial\Omega_2, \mathbb{R}^n) \rightarrow H_0^1(\Omega, \mathbb{R}^n)$$

$$u \mapsto U,$$

Neumann-Lösungsoperator, wobei  $U$  die Lösung des Problems

$$a(U, V) = \int_{\partial\Omega_2} u^T(\gamma_0 V) ds \quad \forall V \in H_0^1(\Omega, \mathbb{R}^n)$$

ist, wobei  $u^T(\gamma_0 V)$  eine duale Paarung meint. Außerdem heißt der Operator

$$E_D : H^{1/2}(\partial\Omega_2, \mathbb{R}^n) \rightarrow H_0^1(\Omega, \mathbb{R}^n)$$

$$u \mapsto U,$$

## 1 BFGS-Algorithmus in der Formoptimierung

*Dirichlet-Lösungsoperator*, wobei  $U$  die Lösung des Problems

$$\begin{aligned} a(U, V) &= 0 \\ \text{unter } U|_{\partial\Omega_2} &= u \end{aligned} \quad \forall V \in H_0^1(\Omega, \mathbb{R}^n),$$

ist.

Wir fahren fort und definieren die projizierten Neumann-zu-Dirichlet und Dirichlet-zu-Neumann-Operatoren, welche wir zur Konstruktion der Riemann'schen Metrik benötigen, siehe [10].

sind die Namen richtig?, soll ich den unnötigen Operator nicht definieren?

**Definition 5** (projizierte Randwertoperatoren). Seien die Voraussetzungen von 3 und 4 gegeben. Dann heißt der Operator

$$\begin{aligned} S^p : H^{-1/2}(\partial\Omega_2) &\rightarrow H^{-1/2}(\partial\Omega_2, \mathbb{R}^n) \rightarrow H_0^1(\Omega, \mathbb{R}^n) \rightarrow H^{1/2}(\partial\Omega_2, \mathbb{R}^n) \rightarrow H^{1/2}(\partial\Omega_2) \\ \alpha &\mapsto n^T[\gamma_0 \circ E_N(\alpha \cdot n)] \end{aligned}$$

*projizierter Neumann-zu-Dirichlet-Operator*, wobei  $n^T$  eine duale Paarung meint.

ist es richtig, wie linearformen und normale abbildungen miteinander verkettet werden. gibt es

Der Operator

$$\begin{aligned} T^p : H^{1/2}(\partial\Omega_2) &\rightarrow H^{1/2}(\partial\Omega_2, \mathbb{R}^n) \rightarrow H_0^1(\Omega, \mathbb{R}^n) \rightarrow H^{-1/2}(\partial\Omega_2, \mathbb{R}^n) \rightarrow H^{-1/2}(\partial\Omega_2) \\ \alpha &\mapsto n^T[\gamma_1 \circ E_D(\alpha \cdot n)] \end{aligned}$$

heißt *projizierter Dirichlet-zu-Neumann-Operator*.

Die hier definierten Operatoren sind nicht die klassischen Dirichlet-zu-Neumann-Operatoren, welche auch *Stekolov-Poincaré-Operatoren* genannt werden, da diese nicht in projizierter Variante definiert werden. Wir haben diese so definiert, da wir die Identifikation der Tangentialvektoren, für welche die Riemann'sche Metrik definiert wird, mit Skalarfeldern  $\alpha$  auf dem Rand  $\partial\Omega_2$  nach Auswertung über äußere Normalenvektorfelder ausnutzen. Zur Erinnerung, siehe 2. Was die Operatoren im wesentlichen tun, ist für gegebene Dirichlet- bzw. Neumann-Bedingungen die Lösung des durch die Bilinearform  $a$  definierten Problems zu finden, und die entsprechenden anderen, automatisch konsistenten, Neumann- bzw. Dirichlet-Bedingungen zurückzugeben. Das heißt, für gegebene Dirichlet-Bedingungen liefert der projizierte Dirichlet-zu-Neumann-Operator die Neumann-Bedingungen, welche die selbe Lösung erzeugt, und vice versa.

## 1 BFGS-Algorithmus in der Formoptimierung

Die klassischen Stekolov-Poincaré-Operatoren sind zueinander invers und vererben die Koerzitivität, Stetigkeit und Symmetrie in dualer Paarung, aufgrund der Koerzitivität und siehe die Quellen bei [10], Def. 3.1. Die projizierten Operatoren besitzen weiterhin alle genannten Eigenschaften, bis auf die des Inversen. Um Rechenaufwand einzusparen, und Glättungseigenschaften auszunutzen, fällt die Wahl des Operators zur Definition des Skalarproduktes auf den Tangentialräumen auf  $(S^p)^{-1}$ , für Details, unter anderem zu spektraler Äquivalenz der Kandidaten, siehe [10] und die dort genannten Quellen. Wir kommen nun zu unserem Ziel, eine Riemann'sche Metrik auf dem Shape-space für den  $\mathbb{R}^2$  zu definieren, vgl. [10].

Zitat Schulz und Jahr, for the credit :)

**Definition 6** (Stekolov-Poincaré-Metrik). Seien die Voraussetzungen wie in 5. Dann heißt das Skalarprodukt

$$g^S : H^{1/2}(\partial\Omega_2) \times H^{1/2}(\partial\Omega_2) \rightarrow \mathbb{R}$$
$$(\alpha, \beta) \mapsto \langle \alpha, (S^p)^{-1}\beta \rangle = \int_{\partial\Omega_2} \alpha(s) \cdot ((S^p)^{-1}\beta)(s) ds,$$

*Stekolov-Poincaré-Metrik*, wobei die Produkte auf der rechten Seite als duale Paarungen zu verstehen sind.

Das so definierte Skalarprodukt ist auf allgemeinen Sobolev-Slobodeckij-Räumen definiert, da Lösungen der Zustandsgleichung bei Auswertung auf dem Rand in der Regel Ordnung 1/2 besitzen, siehe Referenz auf PDE kapitel. Diese Räume enthalten, wie wir in der Einführung zu Differentialgleichungen gezeigt haben, die  $C^\infty$ -Funktionen auf dem Rand. Somit ist das Skalarprodukt für Tangentialvektoren aller Formen aus  $B_e(S^1, \mathbb{R}^2)$  wohldefiniert, da wir erneut Tangentialvektoren mit den Koeffizientenfeldern der äußeren Normalenvektorfelder identifizieren. Für den Beweis, dass  $g^S$  eine Riemann'sche Metrik bildet, und  $(B_e(S^1, \mathbb{R}^2), g^S)$  somit eine Riemann'sche Mannigfaltigkeit ist, siehe zitiere Welker.

Wir kommen nun zu dem Zusammenhang des in zitiere shape calc definierten Formkalküls und der soeben konstruierten Riemann'schen Mannigfaltigkeit  $(B_e(S^1, \mathbb{R}^2), g^S)$ . Für die in dieser Arbeit genutzten Algorithmen ist die Gewinnung von Gradienten, welche wir Generierung von Deformationen nutzen wollen, aus der Formableitung eines Zielfunktional von zentraler Bedeutung. Das Skalarprodukt  $g^S$  bietet genau hierzu das geeignete Mittel.

Definition von Formen mit Omega im shape kapitel, hier mit partialOmega2, irgendwie komisch

## 1 BFGS-Algorithmus in der Formoptimierung

**Definition 7** (Formgradient). Seien die Voraussetzungen wie in 3 und 4. Sei  $\mathcal{J}$  ein formdifferenzierbares Formfunktional und die zugehörige Formableitung  $D\mathcal{J}$ . Betrachte die zu  $c \in B_e(S^1, \mathbb{R}^2)$  gehörige Form  $\partial\Omega_2$ , mit Tangentialraum  $T_c B_e$ . Dann heißt der Tangentialvektor  $h \in T_c B_e$  *Formgradient von  $\mathcal{J}$  in  $c$* , falls seine Darstellung  $\alpha_h$  als Skalarvektorfeld, gemäß 2,

stimmt hier für alle Cinfy  $V$ ? eigentlich H10, aber das ersteres dicht liegt und gS bzw a stetig

$$g^S(\alpha_h, \alpha_V) = D\mathcal{J}(\partial\Omega_2)[V] \quad \forall V \in C^\infty(\Omega, \mathbb{R}^2) \quad (1.1)$$

erfüllt, wobei  $\alpha_V = \langle V|_{\partial\Omega_2}, n \rangle$  das Koeffizientenfeld des äußeren Normalenanteils  $n$  von  $V$  auf dem Rand  $\partial\Omega_2$  ist.

Die hier gemachte Definition ist als eine duale Repräsentation der Formableitung  $D\mathcal{J}$  in dem Skalarprodukt  $g^S$  zu sehen. Damit folgen wir nicht der typischen Definition eines gewöhnlichen Gradienten, wie diese oft in der elementaren Analysis stattfindet, sondern betrachten diese in Anlehnung an den Riesz'schen Darstellungssatzes. Der Aufwand bis zur Definition des Formgradienten macht auch deutlich, wie wesentlich die Art des Gradienten vom zugrunde liegenden Skalarprodukt abhängt, ganz im Gegenteil zur Formableitung, für welche kein zugrunde liegendes Skalarprodukt benötigt wird. Somit ergeben sich für die selbe Formableitung, je nachdem welche Metrik man für den Shape-space verwendet, andere Gradienten und somit andere numerische Verfahren. Für eine Auswahl weiterer Alternativen, siehe [14], Kapitel 3.2.

Es fällt weiterhin auf, dass die hier geforderte Gleichung 1.1 nicht in dualer Paarung mit Skalarfeldern, welche in die Ableitung  $D\mathcal{J}$  einfließen, sondern in deren Koeffizientenfeldern der Normalenkomponente auf der Form berücksichtigt werden. Dies lässt sich mit dem Hadamard'schen Darstellungssatz begründen:

$$\begin{aligned} D\mathcal{J}(\partial\Omega_2)[V] &= \int_{\partial\Omega_2} f(s) \langle V(s), n(s) \rangle ds \\ &= \int_{\partial\Omega_2} f(s) \alpha_V(s) \langle n(s), n(s) \rangle ds \\ &= \int_{\partial\Omega_2} f(s) \alpha_V(s) ds \\ &= (f, \alpha_V)_{\mathcal{L}^2(\partial\Omega_2)} \end{aligned}$$

Damit wird klar, dass Vektorfelder  $V \in C^\infty(\Omega, \mathbb{R}^2)$  mit gleichen Werten in äußerer Normalenrichtung  $n$  auf  $\partial\Omega_2$  auch den selben Wert bei Ableitung er-

## 1 BFGS-Algorithmus in der Formoptimierung

halten, und somit lediglich die Koeffizientenfelder  $\alpha$  relevant sind. Die Identifikation mit diesen ist auch nötig, da wir das Skalarprodukt  $g^S$  für gerade diese Funktionen definiert haben, und nicht bezüglich der echten Tangentialvektoren, welche Vektorfelder sind.

Erklärung, warum das alles ok ist mit dem Koeffizientenfeld, und das Cinfy R2 nicht so stark

Mit diesem Zusammenhang lässt sich auch leicht der Bezug zu auf ganz  $\Omega$  definierten Vektorfeldern  $U, V$ , und der Bilinearform  $a$ , welche  $g^S$  zu Grunde liegt, beschreiben. Wir formulieren die Zusammenhänge aus [10] für unsere Situation und Notation als Satz um, und geben einen Beweis.

vielleicht  $h = \text{grad } J$  nenne?

**Theorem 8** (Formgradienten und Deformationsfelder). Seien die Voraussetzungen wie in 7. Sei  $a$  eine koerzive, beschränkte, symmetrische Bilinearform, und  $g^S$  die zugehörige Stekolov-Poincaré-Metrik. Weiterhin sei  $h$  der zu einer Form  $\partial\Omega_2$  und Formfunktional  $\mathcal{J}$  gehörige Formgradient. Dann existiert ein zu  $h$  gehöriges Vektorfeld  $H \in H_0^1(\Omega, \mathbb{R}^n)$ , so dass gilt

$$g^S(\alpha_h, \alpha_V) = D\mathcal{J}(\partial\Omega_2)[V] = a(H, V) \quad \forall C^\infty(\Omega, \mathbb{R}^2) \quad (1.2)$$

und, mit Notation aus 3,

$$(\gamma_0 \circ H)^T n = \alpha_h.$$

sicher das hier  $\Omega = \Omega_2$  sein kann?? muss ich nicht zuerst eine Hold-all-domain definieren

*Beweis.* Seien die Voraussetzungen wie oben. Nach der zuvor geführten Rechnung gilt

$$g^S(\alpha_h, \alpha_V) = (f, \alpha_V)_{L^2(\partial\Omega_2)},$$

wobei  $f$  aus der Randdarstellung von  $D\mathcal{J}$  mittels Hadamard'schem Darstellungssatz stammt. Verwendet man die Definition 6 von  $g^S$ , so erhält man

$$\int_{\partial\Omega_2} \alpha_h(S^p)^{-1}(\alpha_V) ds = \int_{\partial\Omega_2} f \alpha_V ds$$

für beliebige Vektorfelder  $V \in C^\infty(\Omega, \mathbb{R}^2)$ . Wegen der Invertierbarkeit von  $S^p$ , welche wegen **Stetigkeit** und Koerzivität der Bilinearform  $a$  und Regularität des Randes **mit dem Lemma von Lax-Milgram, sicher auch für Neumann??, REFERENZ**

## 1 BFGS-Algorithmus in der Formoptimierung

gilt, sowie der Symmetrie von  $g^S$ , folgt mit dem Fundamentallemma der Variationsrechnung

$$S^p(f) = \alpha_h.$$

Nun folgt mit der Definition von  $S^p$ , siehe 4, und mit der Stetigkeit und Symmetrie von  $a$ , dass ein  $H \in H_0^1(\Omega, \mathbb{R}^2)$  **checke ob wirklich auf Omega, oder nur Omega2** existiert, so dass gilt

$$a(H, V) = \int_{\partial\Omega_2} f \alpha_V ds \quad \forall V \in C^\infty(\Omega, \mathbb{R}^2),$$

woraus insgesamt die erste Gleichung **vllt ref? label?** folgt. Die zweite Gleichung folgt direkt aus der Definition des projizierten Operators  $S^p$ , siehe 5, und des Neumann-Lösungsoperators  $E_N$ , denn

$$\alpha_h = S^p(f) = n^T(\gamma_0 \circ E_N(f \cdot n)) = n^T(\gamma_0 \circ H)$$

□

Mit diesem Resultat haben wir nun mehrere Möglichkeiten, aus gegebener Formableitung  $D\mathcal{J}$  eine Deformation einer Form  $\partial\Omega_2$  zu erzeugen. Zum einen besteht die Möglichkeit, auf dem Rand  $\partial\Omega_2$  zu agieren, indem man ein Normalenvektorfeld  $h = \alpha_h \cdot n$  auf diesem mittels des Variationsproblems 1.2 auf Basis der linken Gleichheit erzeugt. Zu beachten ist, dass die so erzeugte Deformation lediglich auf dem Rand definiert ist, womit dann in der Anwendung das Gitter im Inneren und Äußeren nicht bewegt wird. Aus diesem Grund bezeichnen wir diese Art der Deformationsgewinnung *Randformulierung*. Die andere Variante ist, eine Deformation auf dem gesamten Gebiet  $\Omega$  zu gewinnen, indem man das Variationsproblem 1.2 auf Basis der rechten Gleichung mittels der Bilinearform  $a$  zu erzeugen. Im Gegensatz zur ersten Variante können hier potentiell alle Punkte des Gitters verschoben werden, trotz dessen, dass lediglich die Randpunkte zur Änderung des Zielfunktionals beitragen. Diese Variante bezeichnen wir als *Volumenformulierung*, angelehnt an die Arten der Darstellung der Formableitung **REFERENZ Kapitel**. Beide Varianten verwenden als zur Erzeugung des Gradienten die Ableitungsinformation, welche physikalische als Kraft interpretierbar ist. Wählt man als Beispiel für die Bilinearform  $a$  die lineare Elastizitätsgleichung, so wird diese Interpretation noch deutlicher. Genau diese Bilinearform werden wir in der später **wirklich später oder wann? lage anschauen, wo Definiere ich mal diese lineare Elastizität? vllt**



## 1 BFGS-Algorithmus in der Formoptimierung

folgenden praktischen Implementierung wählen, um in Volumenformulierung Formgradienten zu erzeugen.

Wir erwähnen an dieser Stelle noch, dass die durch das Variationsproblem 1.2 erzeugte Deformationen  $h$  im Allgemeinen nicht in  $C^\infty(\Omega, \mathbb{R}^2)$  sind, siehe [10]. Dies liegt daran, dass im Allgemeinen je nach Wahl der Bilinearform  $a$ , welche den Operator  $S^p$  und somit das Skalarprodukt  $g^S$  erzeugt, sowie der rechten Seite der Gleichung und der Regularität des Gebietes auf der diese definiert sind, lediglich eine Lösung in  $H_0^1(\Omega, \mathbb{R}^2)$  existiert. Aus diesem Grunde ist nach unserer Definition 2  $h$  nicht immer Element des Tangentialraums  $T_c B_e$ , was Probleme theoretischer Natur macht. Unter anderem deshalb besteht an dieser Stelle der Bedarf eines allgemeineren Formraumes. Beispielsweise wäre eine Möglichkeit, Formen mit  $H^{1/2}$ -Rändern zuzulassen, was in [14], Kapitel 7, kurz erläutert wird. Hierbei entstehen allerdings neue Schwierigkeiten theoretischer Natur, welche noch nicht völlig erforscht sind. Für eine Einführung in diese Thematik und zugehörige Konzepte verweisen wir auf [13].

Da wir nun mit 1.2 verstanden haben, wie Formgradienten mit Hilfe von Metriken  $g^S$  und Bilinearformen  $a$  erzeugt werden können, möchten wir für dieses ein Beispiel angeben. Die Wahl einer solchen Bilinearform  $a$ , und damit die einer Metrik  $g^S$ , ist in der Tat nicht trivial, und mehrere Möglichkeiten mit unterschiedlichen Eigenschaften bestehen hier. Die Wahl hängt unter anderem davon ab, welche **Ordnung der Hesse-Operator des betrachteten Problems hat.** Eine mögliche Wahl hierzu, wäre die zur schwachen Formulierung der *biharmonischen Gleichung* gehörende Bilinearform **soll ich die konkret angeben? vllt mit Quelle?**. Wir werden für die Gewinnung von Formgradienten im Nachfolgenden die lineare Elastizitätsgleichung verwenden, welche wir nun einführen, wobei wir uns an [10] und [17] halten. **auf beliebigem gebiet Omega oder auf (0,1)^2? und Form dazu definieren**

**Definition 9** (Lineare Elastizitätsgleichung). Betrachte das Gebiet  $(0, 1)^2 \subset \mathbb{R}^2$ . Sei  $f_{elas} : (0, 1)^2 \rightarrow \mathbb{R}$  eine Funktion. Dann heißt die Differentialgleichung

$$\begin{aligned} \operatorname{div}(\sigma) &= f_{elas} \quad \text{in } (0, 1)^2 \\ U &= 0 \quad \text{auf } \partial([0, 1]^2) \end{aligned} \tag{1.3}$$

**mit,**

$$\begin{aligned} \sigma &:= \lambda_{elas} \operatorname{Tr}(\epsilon) I + 2\mu_{elas} \epsilon \\ \epsilon &:= \frac{1}{2}(\nabla U + \nabla U^T), \end{aligned}$$

## 1 BFGS-Algorithmus in der Formoptimierung

lineare Elastizitätsgleichung in starker Formulierung, wobei  $\text{Tr}(\epsilon)$  die Spur der Matrix  $\epsilon$  ist, und  $\lambda_{elas} \in \mathbb{R}$ , sowie  $\mu_{elas} \in [0, \infty)$ , die sogenannten *Lamé-Parameter* sind.  $\sigma$  wird als Dehnungs-,  $\epsilon$  als Spannungstensor bezeichnet.

brauche ich das mit dem young modul und poisson zahl unbedingt? oder sollte ich vollständig

Die Lösung  $U : \Omega \rightarrow \mathbb{R}^2$  in dieser Differentialgleichung lässt sich als physikalische

Deformation auf dem Gebiet  $\Omega$  auffassen, wobei  $f_{elas}$  sich als physikalische Kraft interpretiert v

LOKAL VARIIERENDE LAME-PARAMETER SIND IMPLEMENTIERT. SOLL ICH DIES

Die Lamé-Parameter  $\lambda_{elas}, \epsilon_{elas}$  besitzen hier keine physikalische Bedeutung, jedoch lassen haben diese Einfluss auf die Gitterdeformation, indem sie indirekt die Schrittweite steuern. Sie sind mit Hilfe des sogenannten *Young'schen Elastizitätsmoduls*  $E$  und der *Poissonzahl*  $\nu$  durch

$$\lambda_{elas} = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}, \quad \epsilon_{elas} = \frac{E}{2(1 + \nu)}$$

darstellbar. Das Elastizitätsmodul  $E$  lässt sich als Steifigkeit des Materials interpretieren. Die Poissonzahl  $\nu$  gibt an, in welchem Verhältniss sich das einen Gitterpunkt umgebende Gitter ausdehnt, falls dieser Gitterpunkt in eine Richtung verschoben wird. Insgesamt lassen also die Lamé-Parameter durch ihre Wahl die Möglichkeit einer Art Schrittweitensteuerung zu. Im weiteren Verlauf dieser Arbeit wählen wir  $\lambda_{elas} = 0$ .

Möchte man nun mit Hilfe der linearen Elastizitätsgleichung auf Basis von Theorem 1.2 einen Formgradienten bestimmen, so besitzt man zwei Möglichkeiten. Zum einen lässt sich der Gradient mittels der Randformulierung der Formableitung  $D\mathcal{J}$  bilden. Hierzu wird eine weitere Dirichlet-Randbedingung bei der lineare Elastizitätsgleichung 1.2, welche den Formgradienten bezüglich einer Sobolevmetrik auf der Form  $\partial\Omega_2$  repräsentiert gebildet, und  $f_{elas}$  wird auch 0 gesetzt. Für Details hierzu, siehe [10]. Als andere Variante lässt sich die Volumenformulierung der Formableitung  $D\mathcal{J}$  verwenden. Hierzu assembliert man bei der linearen Elastizitätsgleichung 1.3 die rechte Seite  $f_{elas}$  als den Volumenanteil  $D\mathcal{J}_{target}$  referenz auf Volumenform anderes Kapitel. Die Perimeter-Regularisierung, welche nur von der Form  $\partial\Omega_2$  abhängt, wird mit Hilfe einer zusätzlichen von-Neumann-Randbedingung der Form

$$\frac{\partial U}{\partial n} = f^{surf} \quad \text{auf } \partial\Omega_2$$

berücksichtigt. Formuliert man nun die so erhaltene lineare Elastizitätsgleichung in schwacher Formulierung um und verwendet als rechte Seite zitiere anders Kapitel Form

## 1 BFGS-Algorithmus in der Formoptimierung

so erhält man

$$a_{elas}(U, V) := \int_{(0,1)^2} \sigma(U) : \epsilon(V) \, dx = D\mathcal{J}(\Omega_2)[V] \quad \forall V \in H_0^1((0,1)^2, \mathbb{R}^2). \quad (1.4)$$

1.4 ermöglicht nun bei relativ leichter Assemblierung die Berechnung eines Formgradienten durch lösen einer aus der linearen Elastizitätsgleichung erzeugten Variationsproblems, wobei wir die Rechtfertigung hierfür wie bei der Herleitung erwähnt in 1.2 finden. Wir werden für die praktische Implementierung 1.4 verwenden, wobei die Randformulierung prinzipiell Alternative zur Verfügung stände.

vllt. Bereiche trennen, evtl unterabschnitte?

## 1.2 Quasi-Newton-Verfahren

definiere inear, quad, und superlineare konvergenz, dazu noch bilder von typischen konvergenz

Nun besitzen wir alle theoretischen Hintergründe aus dem Bereich der Formoptimierung, um auf Ableitungen basierte Optimierungsverfahren einzuführen.

Im Rahmen dieser Arbeit konzentrieren wir uns auf das sogenannte *Limited-Memory-Broyden-Fletcher-Goldfarb-Shanno-Verfahren* (*L-BFGS-Verfahren*). Dieses Verfahren ist ein sogenanntes *Quasi-Newton-Verfahren*, welche wir im Folgenden nach [18] erläutern möchten. Bevor wir dies tun, geben wir noch Definitionen von Konvergenzgeschwindigkeiten numerischer Optimierungsmethoden an, vgl. [18], Appendix A2.

schaue, ob nicht problem mit formuliert werden muss, denke so ist

**Definition 10** (Konvergenzraten). Betrachte ein Minimierungsproblem, wobei  $x^*$  eine optimale Lösung ist. Sei  $\{x_n\}_{n \in \mathbb{N}}$  eine Folge mit  $x_n \rightarrow x^*$  in geeignetem Sinne. Dann heißt die Folge

- i) *linear konvergent*, falls ein  $c \in (0, 1)$  existiert, mit

$$\frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|} \leq c \quad \text{für } n \text{ hinreichend groß}, \quad (1.5)$$

- ii) *superlinear konvergent*, falls gilt

$$\lim_{n \rightarrow \infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|} = 0, \quad (1.6)$$

- iii) *quadratisch konvergent*, falls ein  $c \in (0, 1)$  existiert, mit

$$\frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^2} \leq c \quad \text{für } n \text{ hinreichend groß}. \quad (1.7)$$

## 1 BFGS-Algorithmus in der Formoptimierung

erklärung + evtl diagramme, damit man weiß, was typisch ist Übergang besser, auch zu new

Bevor wir die Quasi-Newton- und Newton-Verfahren in Shape spaces diskutieren, beginnen wir mit der theoretisch gesicherten Betrachtung der Verfahren in endlich dimensionalen Vektorräumen. Das klassische Newton-Verfahren versucht, im Gegensatz zu einem Gradientenverfahren, auch Informationen zweiter Ordnung des Zielfunktional in Form eines Hesseoperators  $\text{Hess}\mathcal{J}$  bei der Bildung eines Schrittes miteinzubeziehen. Verwendet die übliche Notation im endlich-dimensionalen Fall, so geschieht dies über das Lösen des Problems

$$\text{Hess}\mathcal{J}(x_k)\Delta x = -\nabla\mathcal{J}(x_k). \quad (1.8)$$

gefällt noch nicht ganz Verwendet man dieses Verfahren zur Lösung eines endlich-dimensionalen Optimierungsproblems mit geeigneter Schrittweitensteuerung, so besitzt das volle Newton-Verfahren bei geeigneten Voraussetzungen quadratische Konvergenz, siehe [18]. Quasi-Newton-Verfahren ergeben sich nun aus dem Newton-Ansatz, indem man statt der Hessematrix  $\text{Hess}\mathcal{J}(x_k)$  eine Approximation  $B_k$  für diese verwendet. Hierzu gibt es eine Vielzahl von Möglichkeiten. Genannt seien hier unter anderem die *Symmetric Rank 1 (SR1)* Updateformel und das *DFP-Verfahren*, siehe etwa [18]. Die BFGS-Update-Formeln für die Hessematrix und ihre Inverse, falls existent, sind im endlich-dimensionalen Fall wie folgt definiert, vgl. [18], 6.1.

**Definition 11** (BFGS-Updates (endl. Dim.)). soll ich vllt noch dein endl. dim. Optimierungsp  
Sei  $B_k$  die Approximation der Hessematrix  $\text{Hess}\mathcal{J}(x_k)$ , weiterhin seien

$$\begin{aligned} s_k &:= x_{k+1} - x_k \\ y_k &:= \nabla\mathcal{J}(x_{k+1}) - \nabla\mathcal{J}(x_k) \\ \rho_k &:= \frac{1}{y_k^T s_k}. \end{aligned}$$

Dann ist der *BFGS-Update* definiert durch

$$B_{k+1} := B_k - \frac{B_k(s_k s_k^T)B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \quad (1.9)$$

Das Inverse der Matrix  $B_{k+1}$  besitzt die Updateformel

$$B_{k+1}^{-1} := (I - \rho_k s_k y_k^T) B_k^{-1} (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T. \quad (1.10)$$

In der Definition ist nicht klar, was die natürliche Wahl für ein  $B_0$ , beziehungsweise  $B_0^{-1}$  ist. Hier gibt es mehrere Möglichkeiten, beispielsweise kann

## 1 BFGS-Algorithmus in der Formoptimierung

man eine mittels automatischem Differenzieren gewonnene Hesseapproximation als Start  $B_0$  wählen. Alternativ kann man die skalierte Identität  $\alpha I$  mit  $\alpha \in [0, \infty)$  wählen, um eine positiv definite Startmatrix zu erhalten.

Diese Formeln sind sogenannte *symmetric rank 2 Updates*, da diese eine symmetrische Approximation der Hessematrix mit Updates vom Rang 2 sind. Leitet man die Updates auf abstraktem Wege her, so bietet sich die Möglichkeit an, zuerst  $B_{k+1}^{-1}$  zu definieren, und mit Hilfe der sogenannten Sherman–Morrison–Woodbury Formel den Update für  $B_{k+1}$  aus  $B_{k+1}^{-1}$  herzuleiten. Wir geben an dieser Stelle genau diese abstrakte definierende Eigenschaft der BFGS-Approximation  $B_{k+1}^{-1}$  an, vgl. [18], da wir in dieser eine mögliche Chance zur Verallgemeinerung einer Hesseapproximation im Shape Space sehen.

**Theorem 12** (Definierende Eigenschaft des BFGS-Updates). Es gelte die Notation aus 11. Betrachte das Optimierungsproblem

$$\begin{aligned} \min_H & \|H - B_k^{-1}\|_{W\text{Frob}} \\ \text{s.t. } & H^T = H \text{ und } Hy_k = s_k, \end{aligned} \quad (1.11)$$

wobei  $B_k^{-1}$  die in 11 definierte Inverse des  $k$ 'ten BFGS-Updates ist, und  $\|\cdot\|_{W\text{Frob}}$  die gewichtete Frobeniusnorm ist, siehe [18], 6.1. Dann ist der BFGS-Update der Inversen  $B_{k+1}^{-1}$  die eindeutige Lösung des Problems 1.11.

Wie man sieht, lässt sich die Inverse der BFGS-Approximation der Hessematrix als Lösung eines beschränkten Optimierungsproblems für Matrizen definieren. Die einschränkenden Bedingungen entstehen hierbei auf natürliche Weise. Zum einen beschreibt die erste Nebenbedingung lediglich die Forderung einer symmetrischen Approximation, die zweite Nebenbedingung lässt sich, unter Verwendung der eingeführten Notation aus 11, äquivalent umformen zu

$$B_k(x_{k+1} - x_k) = \nabla \mathcal{J}(x_{k+1}) - \nabla \mathcal{J}(x_k), \quad (1.12)$$

was bekannt ist als *Sekantengleichung*. Diese ergibt sich als natürliche Forderung, was man direkt sieht, wenn man  $B_k$  durch die eigentliche Hessematrix  $\text{Hess}\mathcal{J}(x_k)$  ersetzt. rly? :D Um Existenz einer Inversen zu gewährleisten ist als Bedingung hinreichend, dass die Matrix  $B_k$  positiv definit ist. Setzt man dies voraus, so ergibt sich aus der Sekantengleichung als notwendige Bedingung

$$s_k^T y_k = s_k^T B_k s_k > 0. \quad (1.13)$$

Diese sogenannte *curvature condition* muss also notwendigerweise erfüllt sein, wenn man positive Definitheit von  $B_k$  erhofft. Andererseits führt diese auch dazu, dass ein BFGS-Update-Schritt, angewandt auf eine positiv definite Matrix

## 1 BFGS-Algorithmus in der Formoptimierung

$B_k$  oder ihr Inverses  $B_k^{-1}$ , erneut eine positiv definite Matrix erzeugt. Trivialerweise ist die bei konvexen Problemen die curvature condition immer erfüllt. Im nicht-konvexen Fall ist es trotzdem möglich mit Hilfe von Line-search-Techniken, etwa der Wolfe oder starken Wolfe-Bedingungen, vgl. [18], die curvature condition zu erfüllen. Zu der Konvergenzgeschwindigkeit des BFGS-Verfahrens in endlicher Dimension lässt sich sagen, dass unter Voraussetzungen, wie etwa Lipschitz-stetige Hessematrizen, superlineare Konvergenz vorhanden ist, siehe [18], Theorem 6.6. Sind die genannten Bedingungen nicht erfüllt, so ist die Konvergenz des BFGS-Verfahrens nicht gewährleistet. Es fällt auf, dass man zur Berechnung eines BFGS-Updates die volle  $n \times n$ -Matrix  $B_k$  speichern muss, da diese in der Regel keine dünn-besetztheit aufweisen, siehe [18], 7.2. Da dies vor allem im Fall großer  $n$ , etwa bei feinen, höherdimensionalen Gittern in der Formoptimierung, beträchtliche Kosten verursacht, bedient man sich der sogenannten *Limited Memory* Variante der Verfahren. Hierzu wird eine Approximation an  $B_k$  erzeugt, indem man eine vorgegebene Anzahl  $l$  der letzten Vektoren, welche bei Erzeugung der letzten Updates verwendet wurden, benutzt und speichert. Dies spart Speicherkapazität auf dem Rechner. Die formale Definition der approximativen Update-Schritte findet sich in [18], 7.(19), welche wir hier nicht angeben, da wir diese, ohne die BFGS-Approximationen  $B_k$  zuvor auf Shape spaces verallgemeinert zu haben, sowieso nicht verallgemeinern können. Stattdessen verwenden wir die sogenannte [?], welche wir später, siehe cite später, auf den Formfall unter Vermeidung der Operatoren  $B_k$  verallgemeinern können.

Wir führen nun in analoger Weise zu dem klassischen Newton Verfahren von zuvor ein Lagrange-Newton-Verfahren in Falle der Formoptimierung ein, um anschließend mögliche Verallgemeinerungen des BFGS-Verfahrens auf den Formfall zu diskutieren. Da wir in Formräumen keine kanonische Vektorraumstruktur besitzen, benötigt man zur Verallgemeinerung des Newton-Verfahrens auf diese zusätzliche Techniken. Die Autoren von [15] liefern dafür einen Zugang unter Ausnutzung der Mannigfaltigkeitsstruktur, welche wir bereits zuvor in dieser Arbeit zur Konstruktion eines Formgradienten eingeführt haben. Hierzu konstruieren die Autoren auf Grundlage des Raums aller Formen eine neue Mannigfaltigkeit, welche im wesentlichen ein Produkt aus Bündeln von den Formen  $\partial\Omega_2$  abhängiger Hilberträume, wie in referenz als Beispiel wie in Kapitel zu Formableit und dem Shape space selber sind, siehe [15], unter Remark 1. Nun verwenden die Autoren Tangentialvektoren auf eben dieser Mannigfaltigkeit, um Richtungen zu gewinnen, mit derer Hilfe mittels der Exponentialabbildung auf dem Raum der Formen ein Optimierungsschritt definiert werden kann. Formal lässt

## 1 BFGS-Algorithmus in der Formoptimierung

sich dies wie folgt notieren, wobei  $\zeta = (h_y, h_{\Omega_2}, h_p)$  ein Tangentialvektor der oben genannten Mannigfaltigkeit, von der Form analog zu **zitieren Lagrange-Funktion**, ist:

i) Löse das Newton-Problem

$$\text{Hess}\mathcal{L}(\zeta_k)\Delta\zeta = -\nabla\mathcal{L}(\zeta_k) \quad (1.14)$$

ii) Berechne den Schritt mit einer Schrittlänge  $\alpha_k$

$$\zeta_{k+1} := \exp_{\zeta_k}(\alpha_k\Delta\zeta). \quad (1.15)$$

Wir haben das Lagrange-Newton-Verfahren im Shape-space ausformuliert, um auf die offenen Schwierigkeiten und Fragestellungen der Übertragung des Quasi-Newton-Ansatzes aus **...** aufmerksam zu machen. Da es sich bei 1.14 um eine Operatorgleichung handelt, anders als bei 1.8, welches ein LGS darstellt, ist nicht klar, welche Arten von Updates eine sinnvolle Verallgemeinerung der Updates **referenz bfgs updates** im endlich-dimensionalen Fall für die Approximation des Hesseoperators im Shape Space sind. **dies müsste ja auch von der Metrik abhängen, oder?**

**man könnte versuchen in analogie zu der Konstruktion von bfgs und dfp aus nocoal vorgehen**

**man muss auch die curvature Bedingung für bfgs in shape metriken übersetzen richtig? da in**

Aufgrund der genannten Schwierigkeiten werden wir uns vorerst mit der Übertragung der 2-Schleifen-L-BFGS Rekursion auf den Formfall begnügen. Hierzu haben die Autoren von [10] den Algorithmus aus [18], 7.4, unter Verwendung des Zusammenhangs 1.2 und der dort vorkommenden Metrik  $g^S$ , sowie der Bilinearform  $a$ , angepasst. Zu beachten ist, dass hier die Tangentialvektoren Elemente verschiedener Tangentialräumen sind, weshalb diese noch zuerst in die richtigen Räume transportiert werden müssen. Für die differentialgeometrische Definition der Transporte  $\mathcal{T}$ , siehe etwa [3] oder [14], 2.3.1. Der klassische Algorithmus entsteht, wenn man statt der Metrik  $g^S$  im endlich-dimensionalen Fall das euklidische Skalarprodukt verwendet, wobei keine Transporte von Tangentialvektoren anfallen. Es folgt die Definition nach [10], Kapitel 4.

**notation hj lieber zu grad (partial omega j) ???** **achtung: länge l oder länge m oder was genau**

**referenz zu den verwendeten Randauswertungen gamma aus vorigem Kapitel**

**Definition 13** (2-Schleifen-L-BFGS Rekursion im Formfall). Seien die Voraussetzungen wie aus 1.2. Seien  $h_j \in T_{\text{partial}\Omega_j}B_e$  Formgradienten der Formen  $\partial\Omega_j$  mit ihren Darstellungen als Koeffizientenvektorfelder  $\alpha_{h_j} \in H^{1/2}(\partial\Omega_j)$

## 1 BFGS-Algorithmus in der Formoptimierung

nach 2, sowie den zugehörigen Darstellungen  $H_j \in H_0^1(\Omega, \mathbb{R}^2)$  auf dem ganzen Gebiet, nach 1.2. Weiterhin seien  $S_j \in H_0^1(\Omega, \mathbb{R}^2)$  Deformationsvektorfelder und  $Y_j := H_{j+1} - \mathcal{T}_{S_j} H_j \in H_0^1(\Omega, \mathbb{R}^2)$  die entlang der Deformation  $\mathcal{J}_{S_j}$  transportierte Differenz der Gradienten. Sei  $m \in \mathbb{N}$  die Anzahl der gespeicherten Gradienten und Deformationen. Dann ist der *2-Schleifen-L-BFGS Rekursion im Formfall* gegeben durch

hier lieber alphas statt gamma verwenden?? also die definition wie wir sie zuvor gemacht haben

format: einrücken bei code wie in python | stimmt das mit den Vorzeichen hier: hatt ja den Faktor

$$\begin{aligned}
 \rho_j &\leftarrow g^S((\gamma_0 Y_j)^T n, (\gamma_0 S_j)^T n)^{-1} = a(Y_j, S_j)^{-1} \\
 q &\leftarrow H_j \\
 \textbf{for } i &= j-1, \dots, j-m \textbf{ do} \\
 S_i &\leftarrow \mathcal{T}_q S_i \\
 Y_i &\leftarrow \mathcal{T}_q Y_i \\
 \alpha_i &\leftarrow \rho_i g^S((\gamma_0 S_i)^T n, (\gamma_0 q)^T n) = \rho_i a(S_i, q) \\
 q &\leftarrow q - \alpha_i Y_i \textbf{end for} \\
 q &\leftarrow \frac{g^S((\gamma_0 Y_{j-1})^T n, (\gamma_0 S_{j-1})^T n)}{g^S((\gamma_0 Y_{j-1})^T n, (\gamma_0 Y_{j-1})^T n)} q = \frac{a(Y_{j-1}, S_{j-1})}{a(Y_{j-1}, Y_{j-1})} q \\
 \textbf{for } i &= j-m, \dots, j-1 \textbf{ do} \\
 \beta_i &\leftarrow \rho_i g^S((\gamma_0 Y_i)^T n, (\gamma_0 q)^T n) = \rho_i a(Y_i, q) \\
 q &\leftarrow q + (\alpha_i - \beta_i) S_i \\
 \textbf{end do.}
 \end{aligned}$$

Dann lässt sich das erzeugte Vektorfeld als Deformation  $S_j := q$  verwenden.

vorzeichen!

mache noch die hier benutzte vereinfachung der Transporte, rede kurz über transportgleichungen

Erklärung



## Literatur

- [1] M. Genzen, A. Staab, Prof. E. Emmrich. Sobolew-Slobodeckij-Räume - die Theorie der gebrochenen Sobolew-Räume, Technische Universität Berlin. 2014.
- [2] G. Geymonat. Trace Theorems for Sobolev Spaces on Lipschitz Domains. Necessary Conditions. 2007.
- [3] J. M. Lee. *Introduction to Smooth Manifolds,, Second Edition*. Springer, Graduate Texts in Mathematics, 2013.
- [4] H. P. Langtangen, A. Logg. *Solving PDEs in Python - The FEniCS Tutorial Volume I*. Springer, 2017.
- [5] K. Burg, H. Haf, F. Wille, A. Meister. *Partielle Differentialgleichungen und funktionalanalytische Grundlagen, 5. Auflage*. Vieweg +Teubner Verlag, Springer Fachmedien, 2010.
- [6] B. Zhong P.A. Sherar, C.P.Thompson, B. Xu. An optimization method based on b-spline shape functions & the knot insertion algorithm. *Proceedings of the World Congress on Engineering*, II, 2007.
- [7] S. Schmidt. Weak and strong form shape Hessians and their automatic generation. 2018, SIAM J. Sci. Comput., Vol. 40, No.2, pp. C210-C233.
- [8] Volker Schulz. A riemannian view on shape optimization. *Foundations of computational Mathematics*, 14:483-501, 2014.
- [9] B. Schweizer. *Partielle Differentialgleichungen - Eine anwendungsorientierte Einführung*. Springer Spektrum, 2013.
- [10] Volker Schulz, Martin Siebenborn. Computational comparison of surface metrics for pde constrained shape optimization. *Comput. Methods Appl. Math 2016*, 2016.
- [11] Kevin Sturm. *On shape optimization with non-linear partial differential equations*. PhD thesis, Technische Universität Berlin, 2015.
- [12] W. Arendt, K. Urban. *Partielle Differenzialgleichungen - Eine Einführung in analytische und numerische Methoden*. Spektrum Akademischer Verlag Heidelberg, 2010.

## LITERATUR

- [13] K. Welker. Suitable Spaces for Shape Optimization. 2017, arXiv: 1702.07579v2.
- [14] Kathrin Welker. *Efficient PDE Constrained Shape Optimization in Shape Spaces*. PhD thesis, Universität Trier, 2016.
- [15] Volker Schulz, Martin Siebenborn, Kathrin Welker. Towards a lagrange-newton approach for constrained shape optimization. *arXiv: 1405.3266v2*, 2014.
- [16] Volker Schulz, Martin Siebenborn, Kathrin Welker. Pde constrained shape optimization as optimization on shape manifolds. *Geometric Science of Information, Lecture Notes in Computer Science*, 9389:pp. 499–508, 2015.
- [17] Volker Schulz, Martin Siebenborn, Kathrin Welker. Efficient pde constrained shape optimization based on steklov-poincaré-type metrics. *SIAM J. OPTIM.*, Vol. 26, No. 4, pp. 2800-2819, 2016.
- [18] Jorge Nocedal, Stephen J. Wright. *Numerical Optimization, Second Edition*. Springer, 2006.
- [19] M. C. Delfour, J. P. Zolésio. *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization, 2nd ed.* SIAM Advances in Design and Control, 2011.