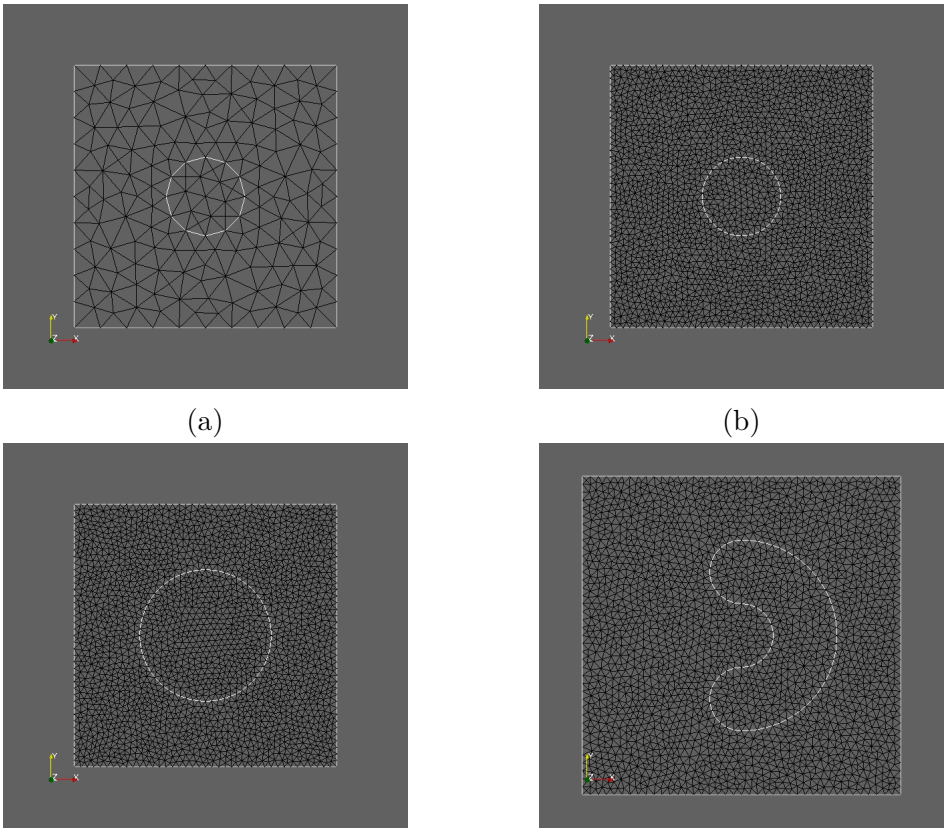
1 Resultate

1.1 ???

l-bfgs springt schnell nahe Lösung: Verwandtschaft zu CG (nocedal + das paper zu konverger
erwähne, dass Gitter Probleme macht; eigentlich Neustart mit Gitter nötig; oder Lameparam
wenn konvergiert, so vllt auch mit lokalen Minima... Gitterunabhängig? Ordnung der Bilinea
plotte l-BFGs vs Gradient mache Langzeit rechnung mit und ohne Line-search bei Pertubati
Programme als Wort, Rechtschreibfehler in anderen Chaptern eventuell bei Zeit die Hinterg

Wie angekündigt möchten wir in diesem abschließenden Abschnitt die Resultate aus unserer Programmierung analysieren. Wir haben mehrere Problemgitter und Kombinationen aus Verfahren getestet, und dabei die Daten aufbereitet. Diese Daten werden auf der beigelegten CD samt dem Programm enthalten sein. Wir haben uns für die Analyse der Verfahren für zwei Testprobleme entschieden. Zum einen soll ein kleiner Kreis zu einem großen Kreis deformiert werden, wobei die Kreismittelpunkte jeweils gleich bleiben. Somit handelt es sich bei diesem Problem um Deformation einer konvexen Form ohne große Translation. Zum anderen soll ein kleiner Kreis zu einer nierenartigen Form deformiert werden. In diesem Problem findet eine verhätlnismäßig starke Translation statt, außerdem ist die Zielform ein nicht konvex, was dieses Problem deutlich schwieriger macht, als das erste. In beiden Fällen sind die Formen in dem Einheitsquadrat im $\mathbb{R}^2$ eingebettet. Diese sind abgebildet 1. Die Parameter, welche bei der Analyse in betracht kommen, sind als Standard auf folgende Werte gesetzt: vllt doch die Namen aus dem Program, mit Tabellenverweis

Gitterfeinheit:	0.1 (normal) , 0.025 (fein)
Lamé-Parameter:	0.0 (min) , 30.0 (max)
Perimeter-Reg.:	0.00001
Funktionswerte für Zustand:	-10.0 (außen) , 100.0 (innen)
Memory-Length:	60
Toleranz für Ausstieg:	0.0008
Backtracking:	5.0 (start_scale), 0.5 (shrinkage), 0.999 (c)

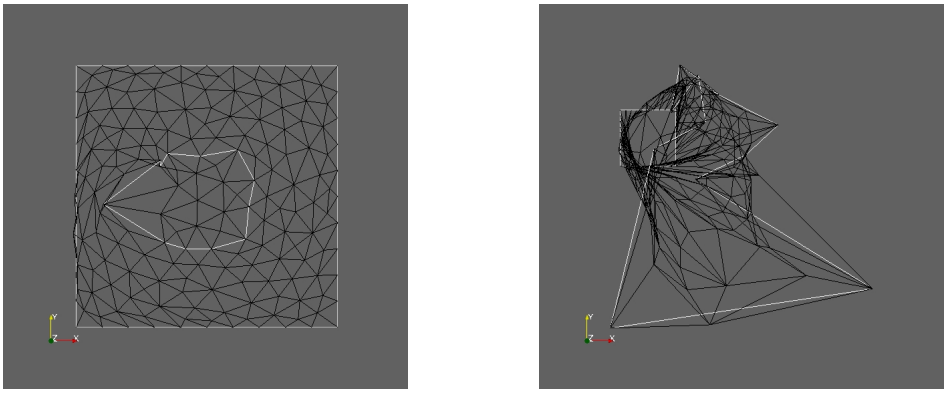


(a) (b)

(c) (d)

Abbildung 1: (a) Ausgangsform kleiner Kreis auf grobem Gitter (b) Ausgangsform kleiner Kreis auf feinem Gitter (c) Zielform großer Kreis auf feinem Gitter (d) Zielform broken Donut auf feinem Gitter

Wir werden sowohl das L-BFGS-Verfahren, als auch das Gradientenverfahren vergleichen. Außerdem wird jeweils die Linesearch an- und ausgeschaltet, sowie alle oben aufgeführten Parameter, bis auf die Funktionswerte die Zustandsgleichung im Zielgitter, den Verbesserungsfaktor c, und den Skalierungsfaktor shrinkage der Backtracking-Linesearch, variiert. Die Verwendung der Memory-Length von 60 bewirkt, dass es sich in allen folgenden Fällen der Verwendung des L-BFGS-Verfahrens eigentlich um ein BFGS-Verfahren handelt, da die gesamte Historie zur Berechnung der Hesseapproximation benutzt wird.
Für das Problem der Deformation zum großen Kreis erhalten wir bei dem groben Gitter bei Verwendung des Gradientenverfahrens ohne Linesearch Konvergenz nach fast 800 Schritten. Schaltet man das Backtracking ein, so erhält



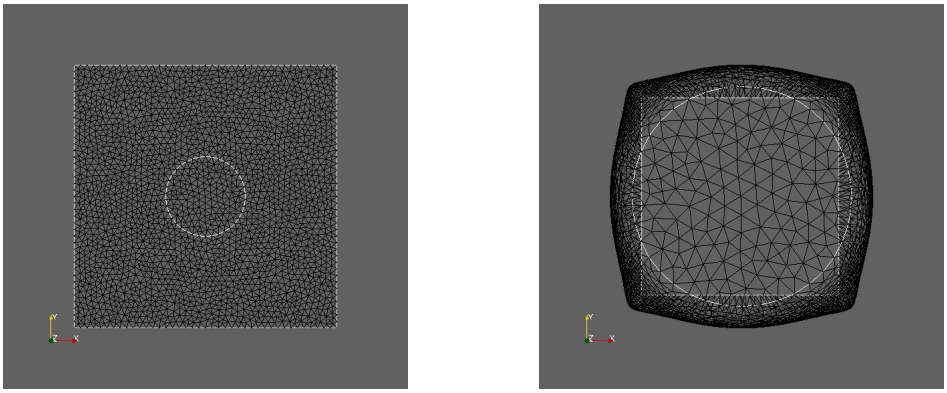
(a) (b)

Abbildung 2: (a) Gitter nach mehrmaligen Schritten unter Verletzung der Curvature Condition und Non-update Schritten (b) der darauf folgende Schritt, Zerstörung des Gitters

man ebenfalls Konvergenz, jedoch schon bei ca. 450 Schritten. Dies legt nahe, das die Schrittweite durch das anfangliche Hochskalieren der Deformation optimaler wird, der Gradient also der Norm nach relativ klein war. Bei Verfeinerung des Gitters um das 4-fache erhalten wir ebenfalls ohne Verwendung der Linesearch beim Gradientenverfahren Konvergenz nach ca. 650 Schritten. Wieder findet durch Verwendung des Backtracking eine Beschleunigung zur Konvergenz statt, diesmal nach 520 Schritten. Erhöht man das anfangliche Hochskalieren vom Faktor 5 auf 20, so erhält man im feinen Gitter Konvergenz schon nach 120 Schritten. Die Geschwindigkeit zur Konvergenz hat sich also im Vergleich zum Gradientenverfahren ohne Backtracking also um das 5-fache gesteigert, wobei sich die erzeugten Gitter bei Ausstieg des Verfahrens mit oder ohne Linesearch und Veränderung des Hochskalierungsfaktors nicht unterscheiden.

Plot: feines gitter; no linesearch; linesearch; highstartscale

Findet das L-BFGS-Verfahren ohne Linesearch Anwendung, so erhält man auf dem groben Gitter keine Konvergenz. Das Verfahren updatet die BFGS-Memory nach mehreren Verletzungen der Curvature Condition nicht, und deformiert das Gebiet schließlich bei Schritt 26 bis zur Unbrauchbarkeit, was in 2 zu sehen ist. Man beachte die starke Entartung der kaum sichtbaren Zellen kurz vor der Zerstörung des Gitters, welche man im Zoom gut erkennt.
Auch bei Verfeinerung des Gitters um das 4-fache erhält man keine Konvergenz des L-BFGS-Verfahrens, sondern erreicht die Zerstörung des Gitters nach



(a) (b)

Abbildung 3: (a) 1. Schritt: Gradientenschritt bei feinem Gitter (b) 2. Schritt: BFGS-Schritt mit Entartung des feinen Gitters

Verletzung der Curvature Condition schon im zweiten Schritt, bei der die Form das Einheitsquadrat verlässt. Man beachte, das dieses mal die Richtung des Schrittes in Richtung Optimum geht, jedoch viel zu lang ist, anders als in **??**, wo zuvor mehrmalige falsche Schritte bei Verletzung der Curvature Condition das Gitter zerstören. Der BFGS-Schritt ist zu sehen in 3. Diese Beobachtung macht erneut deutlich, wie wichtig eine effektive Schrittweitensteuerung bei der Implementierung der Verfahren sind.

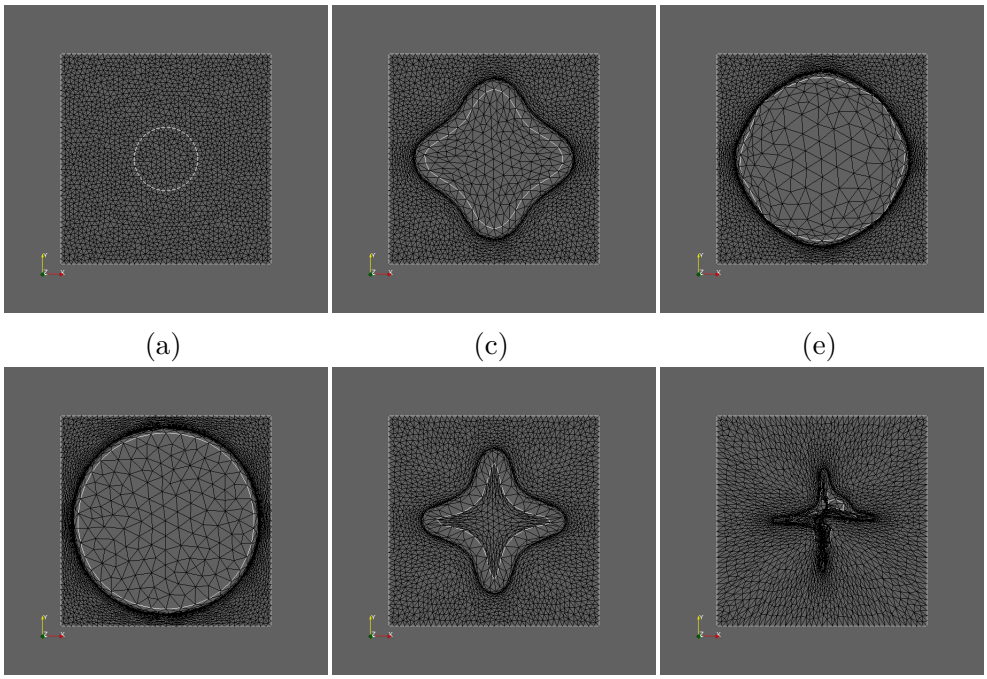
finde raus, warum nur auf der einen Seite kein Text angezeit wird; mache formatierung schön

Wir haben auch versucht, die Gitterzerstörung durch Modifikation der lokal variierenden Lamé-Parameter zu begegnen. Alle bisher gezeigten Gitter hatten die Wahl des minimalen Lamé-Parameters von 0. Um zu beobachten, ob die Verfahren instabiler werden, wenn man konstante Lamé-Parameter wählt, haben wir diese konstant auf 30 gesetzt. Man erhält den Output 4, wobei bei Schritt 5 und 6 die Curvature Condition verletzt sind.

Ein komplett anderes Verhalten stellt sich bei Verwendung der Backtracking-Linesearch ein; in diesem Fall konvergiert das L-BFGS-Verfahren für beide Gitterfeinheiten. Zudem fällt eine erhebliche Steigerung der Konvergenzgeschwindigkeit auf; schon nach 5 Schritten auf dem groben, und nach 6 auf dem feinen Gitter.

konvergenzplot der bfgs und gradientenverfahren mit linesearch

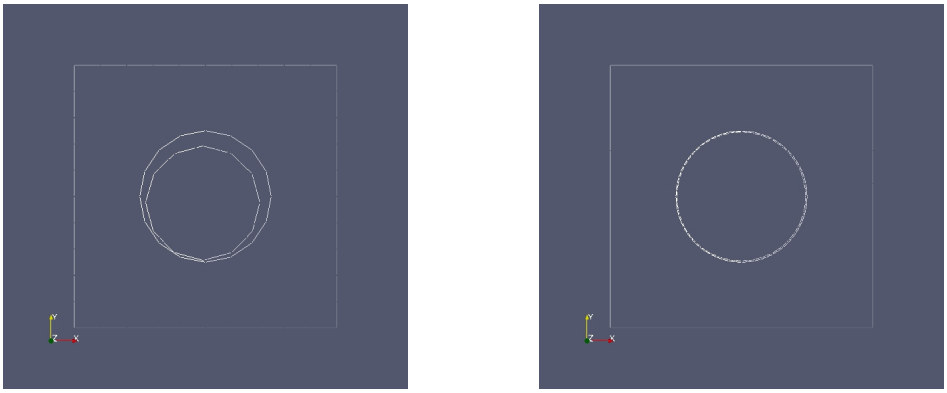
Die bei dem groben Gitter erzeugte optimale Form besitzt noch einen mit bloßem Auge sichtbaren Abstand zur Zielform. In diesem Fall hat unser L-BFGS-Algorithmus bei dem Backtracking maximal herunterskaliert, d.h. kei-



(a) (c) (e)

(b) (d) (f)

Abbildung 4: BFGS-Verfahren ohne Schrittweitensteuerung bei konstanten Lamé-Parametern mit Wert 30; die ersten 6 Schritte



(a) (b)

Abbildung 5: (a) L-BFGS mit Backtracking bei grobem Gitter. Abweichung bei Ausstieg sichtbar. (b) L-BFGS mit Backtracking bei feinem Gitter. Abweichung bei Ausstieg nicht erkennbar.

ne Abstiegsrichtung gefunden und ist ausgestiegen. Damit scheint ein lokales Minimum erreicht. Bei Verwendung des feinen Gitters ist bei selbem Ausstiegskriterium kein Abstand zur Zielform sichtbar, was man in 5 sehen kann.
Wir haben durch Veränderung der Perimeter-Regularisierung untersucht, ob sich durch diese der Abstand zur Zielform erklären lässt. Die Vermutung ist, dass die optimale Form etwas kleiner als die Zielform ist, da die Perimeter-Regularisierung kleinere Formen favorisiert. Die in diesem Fall entstehende optimale Lösung ist in diesem Fall symmetrisch mit gleichem Mittelpunkt wie die Zielform, jedoch besitzt sie fast den selben Abstand wie die optimale Form mit der Perimeterregularisierung. Für die Bilder verweisen wir auf die mitgelieferten Dateien auf der CD.
Kontrollieren wir im Falle des BFGS-Algorithmus auf Veränderung bei Erhöhung des Hochskalierungsfaktors starscale des Backtracking, so stellen wir fest, dass die ersten Schritte stärkere Verbesserungen bringen, sobald sich jedoch eine gewisse Memorygröße, und damit eine Hesseapproximation, aufgebaut haben verschwinden diese Effekte wieder, siehe plot beider Graphen; mit linesearch; fine; high un
Dieses Verhalten sollte auftreten, da BFGS-Schritte mit aufgebauter Hesseapproximation unab
Zudem haben wir einen Vergleich zwischen der vollen BFGS-Methode, welche wir durch die memory_length von 60 erzeugen, und der echten L-BFGS-Methode, mit memory_length von 2 und 3, im Falle des feinen Gitters gezogen. Interessanterweise stellt sich hier kein merklicher Unterschied hinsichtlich Konvergenzgeschwindigkeit in, was man in plot ref erkennen kann. Dies lässt sich

nach unserer Vermutung dadurch erklären, dass für einen korrekten BFGS-Schritt ein Term welcher Ordnung? fehlt. Würde dieser ergänzt, so vermuten wir eine Verbesserung des Algorithmus bei höherer *memory_length*. Außerdem lässt sich in den Diagrammen keine superlineare Konvergenz nachweisen, welche sich bei korrektem Update möglicherweise einstellt.

hier plot; 3 L-BFGS Memlengths im Zielfunktional

Neben der bloßen Bedingung einer Verbesserung im Zielfunktional bei dem Backtracking besteht die Möglichkeit, die Amijo-Bedingung für die Linesearch zu verwenden. Diese haben wir nach [22] implementiert. Verwendet man diese, so konvergiert die L-BFGS-Method nicht mehr, in einigen Fällen konvergiert das Gradientenverfahren. Jedoch würde wir eher behaupten, dass dies *trotz* der Armijo-Bedingung konvergiert. Da diese Bedingung die Verbesserung mit Hilfe eines linearen Modells in der aktuellen Form betrachtet, wäre ein aus der Formoptimierung legitimes anderes Modell vorstellbar, um verbesserte Bedingungen zur Linesearch zu erzeugen. Hierzu hat uns leider die Zeit gefehlt, wir möchten an dieser Stelle jedoch die Relevanz eines solchen Resultats betonen. Bei der Berechnung des Formgradienten in referenz implementierung zeroed haben wir ein Verfahren verwendet, welches in der linearen Elastizitätsgleichung die Punkte, welche nicht Träger des inneren Randes sind, auf Null gesetzt wurden. Schaltet man diese Nullsetzung aus, was durch die Wahl des Parameters zeroed = False möglich ist, so zerstört dies das Verfahren. Sowohl das L-BFGS, als auch das Gradientenverfahren divergieren in allen von uns getesteten Beispielen. Somit tritt des selbe Effekt wie in [21], welcher dort in Abschnitt 5, Fig. 2 zu sehen ist. Würde dieser Effekt durch ein Rundungsrauschen hervorgerufen sein, so vermuten wir, dass dieses weniger drastisch sichtbar würde. Bei uns ist schon nach den ersten Schritten das Gitter bis zur Unbrauchbarkeit entartet.

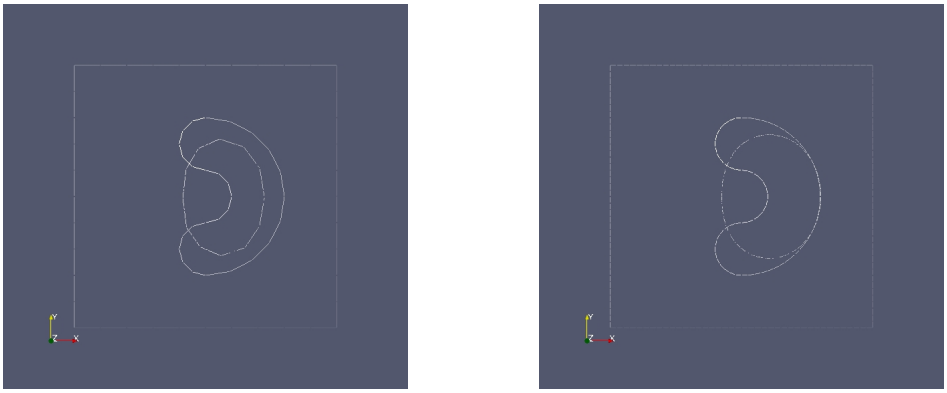
jetzt Donut

Wir haben die oben genannten Beobachtungen auch für das schwierigere Problem der nierenförmigen, nicht konvexen Form gesammelt. Die Ausgangsform, welcher den kleinen Kreis in 1 darstellt, bleibt gleich, wir wechseln also nur die Zielform, die auch in 1 zu sehen ist.

zeige nicht alles in Vollem Detail; Bilder zu Konvergenz; Konvergenzplots

Ausblick und Danksagung an Volker

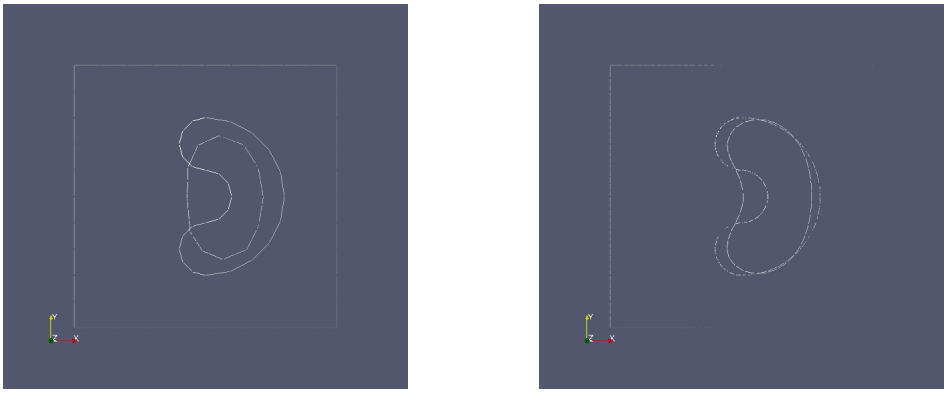
Anders als bei dem einfacheren ersten Problem, erhalten wir für ein Gradientenverfahren ohne Backtracking-Linesearch, in den Fällen sowohl des groben, als auch des feinen Gitters, nach ca. 1200, respektive ca. 600 Schritten zwar Konvergenz, jedoch nicht zum globalen Optimum des Zielgitters. Wir vermu-



(a) (b)

Abbildung 6: (a) Optimale Form und Zielform bei Gradientenverfahren auf grobem Gitter (b) Selbiges auf feinem Gitter

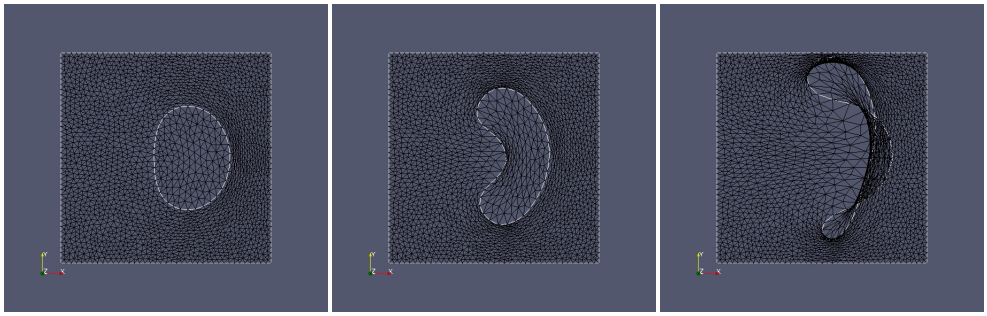
ten, dass es sich hierbei um lokale Minima handelt, die resultierende optimale Form bleibt konvex, trotz des nicht konvexen Zielgitters, siehe Abbildung 6.
Durch das Verwenden der Backtracking-Linesearch schafft es der Algorithmus näher an das globale Optimum zu gelangen. Dies gelingt jedoch ausschließlich bei dem feinen Gitter. Im Falle des groben Gitters erreichen wir auch mit Linesearch die selbe nicht konvexe Form, siehe Abbildung 7, wobei eine leichte Verbesserung zu sehen ist. Dies zeigt, dass das Verfahren stark abhängig ist von der Gitterfeinheit. frag volker dazu, ob das invariant sein sollte. eigentlich nicht mit hesseinformat
Außerdem beobachten wir für das Gradientenverfahren, wie im ersten Problem, eine Skalieren der Konvergenzgeschwindigkeit durch das Hochskalieren der Suchrichtung bei dem Backtracking-Linesearch. Wir sparen uns an dieser Stelle die erneute Angabe eines zu plot von skalierung bei erstem Problem ähnlichen Graphen. Zudem haben wir in diesem Problem die gleichen Feststellungen bei Verwendung der Armijo-Bedingung bei der Linesearch gemacht, wie bei dem ersten Problem zuvor.
jetzt bfgs
Bei Verwendung des L-BFGS-Verfahrens ohne Linesearch bemerken wir, ähnlich wie zu dem ersten Beispiel, dass sowohl beim groben, als auch beim feinen Gitter eine zu 3 ähnliche Divergenz stattfindet, weshalb wir uns Graphiken hierzu sparen. Das Gitter ist nach nur wenigen Schritten, nachdem die Curvature Condition verletzt wurde, bis zur Unkenntlichkeit verformt, wobei das Verfahren auch nicht die lokalen Minima 7 oder 6 erreicht.
Interessanterweise verbessert sich das Verhalten des L-BFGS-Verfahrens oh-



(a) (b)

Abbildung 7: (a) Optimale Form und Zielform bei Gradientenverfahren mit Linesearch auf grobem Gitter (b) Selbiges auf feinem Gitter

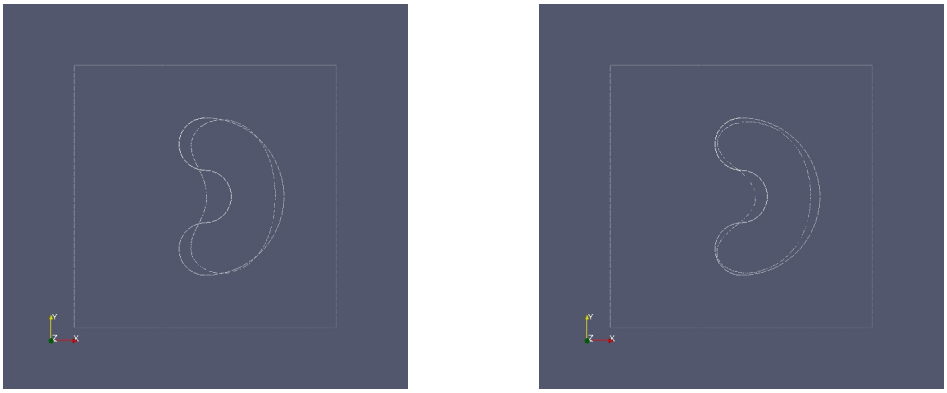
ne Linesearch, wenn man statt lokal variierenden Lamé-Parametern konstante Lamé-Parameter mit Wert 30 wählt. Zuvor divergierte das Verfahren bei dem feinen Gitter nach nur 5 Schritten unter Verletzung der Curvature Condition der letzten 3 Schritte. Bei konstanten Lamé-Parametern findet zwar auch Divergenz statt, jedoch erst nach 14 Schritten, wobei zuvor sogar ein optimales Ergebniss, welches dem Gradientenverfahren mit Linesearch auf dem feinen Gitter nahe kommt. Zu sehen sind ausgewählte Schritte in 8. Man beachte jedoch den Vergleich zu dem Gitter, welches bei dem L-BFGS-Verfahren mit Linesearch und variierenden Lamé-Parametern erzeugt wird, bei dem geringere Entartung der Zellen des Gitters zu beobachten sind. Wir würden an dieser Stelle vorschlagen, ein Kriterium für die Entartung des Gitters, etwa dem Quotienten aus maximalem und minimalem Zellvolumen, zu verwenden, um bei Überschreitung eines festgelegten Entartungsgrades ein neues Gitter mit der aktuellen Form zu initialisieren. Dies ließe sich auch mit lokal variierenden Lamé-Parametern, sowie mit adaptiven Gittermethoden, kombinieren.
Verwendet man zusammen mit dem L-BFGS-Verfahren die Backtracking-Linesearch, so erhalten wir ähnliche Ergebnisse wie in dem ersten Problem. Eine deutliche Erhöhung der Konvergenzgeschwindigkeit, von ca. 900 Schritten des Gradientenverfahrens mit Linesearch und Startskalierung 5, auf 26 Schritte im Falle des L-BFGS mit Linesearch, auf dem feinen Gitter. Auch auf dem groben Gitter ist eine erhebliche Erhöhung der Konvergenzgeschwindigkeit zu beobachten, wobei auch hier lediglich das lokale Minimum des Gradientenverfahrens erreicht wird. Auf dem feinen Gitter wird, ähnlich dem Gradienten-



(a) (b) (c)

Abbildung 8: BFGS-Verfahren bei konstanten Lamé-Parameter von 30 (a) Schritt 6 (b) gute Form bei Schritt 12; man beachte die starke Stauchung/ Streckung der Zellen (c) Zerstörung des Gitters bei Schritt 14

verfahren mit Linesearch, eine sehr gute Lösung erreicht, nahe des globalen Optimums. Damit scheint keine Gitterunabhängigke Konvergenz vorzuliegen. Bilder ref, falls diese verschieden von denen des gradientenverfahrens oben sind, ansonsten au hat das mit der Ordnung des Hessian zu tun? checke, ob konvergenz bei ausstieg, wenn curv Die Curvature Condition wird ab Schritt 17 verletzt, wobei schon in diesem Schritt eine fast vom Ergebniss kaum zu unterscheidene Lösung erzeugt wird. Die Ausstiegsbedingung war lediglich zu stark gewählt. Weiterhin stellen wir fest, dass durch das Weglassen der Perimeter-Regularisierung die Konvergenz im perimeter zero vergleich; mit und ohne linesearch erhalten bleibt. Lässt man zusätzlich die Lamé-Paramter nicht lokal variieren, so bleibt auch hier die Konvergenz erhalten, das Verfahren benötigt jedoch interessanterweise etwas mehr als die doppelte Anzahl an Schritten. Somit scheint in gewissen Fällen die lokal variierende Wahl der Lamé-Parameter die Konvergenzgeschwindigkeit zu erhöhen. Dieser Effekt tritt bei uns nur in Kombination mit dem Weglassen der Perimeter-Regularisierung auf, wobei in allen Fällen ohne Linesearch Divergenz vorliegt.
Außerdem beobachten wir, dass bei einer Memory-Length von 3, sprich bei einem echte Limited-Memory Ansatz, die Anzahl der benötigten Schritte zur Konvergenz von 26 auf 35 auf dem feinen Gitter, und bei dem groben Gitter gleich bleibt. was ist die zugehörige Interpretation? bleibt der Graph ähnlich? wenn ja auf de Vergleich mit mem lengths
konvergenzplot von gradient und bfgs bei linesearch



(a) (b)

Abbildung 9: L-BFGS-Verfahren mit Linesearch (a) Form bei Ausstieg auf grobem Gitter (b) Form bei Ausstieg auf feinem Gitter

plot von bfgs und gradienten im vergleich bei linesearch; Kommentar, das bei guter Linesearc

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] K. Welker, M. Siebenborn. Algorithmic aspects of multigrid methods for optimization in shape spaces. 2017, arXiv: 1611.05272v3.

[4] J. M. Lee. *Introduction to Smooth Manifolds,, Second Edition*. Springer, Graduate Texts in Mathematics, 2013.

[5] H. P. Langtangen, A. Logg. *Solving PDEs in Python - The FEniCS Tutorial Volume I*. Springer, 2017.

[6] M. S. Alnæs, A. Logg. *UFL Specification and User Manual 0.3*. www.fenics.org, 2010.

[7] M. J. D. Powell. Algorithms for nonlinear constraints that use lagrangian functions. *Mathematical Programming 14*, 1976.

[8] K. Burg, H. Haf, F. Wille, A. Meister. *Partielle Differentialgleichungen und funktionalanalytische Grundlagen, 5. Auflage*. Vieweg +Teubner Verlag, Springer Fachmedien, 2010.

[9] 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.

[10] S. Schmidt. Weak and strong form shape hessians and their automatic generation. 2018, SIAM J. Sci. Comput., Vol. 40, No.2, pp. C210-C233.

[11] Volker Schulz. A riemannian view on shape optimization. *Foundations of computational Mathematics*, 14:483-501, 2014.

[12] B. Schweizer. *Partielle Differentialgleichungen - Eine anwendungsorientierte Einführung*. Springer Spektrum, 2013.

[13] Volker Schulz, Martin Siebenborn. Computational comparison of surface metrics for pde constrained shape optimization. *Comput. Methods Appl. Math 2016*, 2016.

[14] Kevin Sturm. *On shape optimization with non-linear partial differential equations*. PhD thesis, Technische Universität Berlin, 2015.

[15] W. Arendt, K. Urban. *Partielle Differenzialgleichungen - Eine Einführung in analytische und numerische Methoden*. Spektrum Akademischer Verlag Heidelberg, 2010.

[16] K. Welker. Suitable Spaces for Shape Optimization. 2017, arXiv: 1702.07579v2.

[17] Kathrin Welker. *Efficient PDE Constrained Shape Optimization in Shape Spaces*. PhD thesis, Universität Trier, 2016.

[18] Volker Schulz, Martin Siebenborn, Kathrin Welker. Towards a lagrange-newton approach for constrained shape optimization. *arXiv: 1405.3266v2*, 2014.

[19] 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.

[20] Volker Schulz, Martin Siebenborn, Kathrin Welker. Structured inverse modeling in parabolic diffusion problems. 2015.

[21] 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.

[22] Jorge Nocedal, Stephen J. Wright. *Numerical Optimization, Second Edition*. Springer, 2006.

[23] M. C. Delfour, J. P. Zolésio. *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization, 2nd ed.* SIAM Advances in Design and Control, 2011.