
EIN VERFAHREN ZUR BESTIMMUNG DES
BRECHUNGSINDEXES AUFGRUND VON
LAUFZEITMESSUNGEN AKUSTISCHER
SIGNALE

Dissertation

zur Erlangung des Grades des

Doktors der Naturwissenschaften

der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

vorgelegt von

Udo Schröder

Saarbrücken

Mai 2015

Tag des Kolloquiums:

7. August 2015

Dekan:

Univ.-Prof. Dr. Markus Bläser

Vorsitzender:

Univ.-Prof. Dr. Matthias Hein

Berichterstatter:

Univ.-Prof. Dr. Thomas Schuster

Akademischer Mitarbeiter:

Univ.-Prof. Dr. Dr. h.c. mult. Alfred K. Louis

Dr. Frederik Heber

Zusammenfassung

Im Jahre 1988 zeigte S.J. Norton, dass, unter bestimmten Voraussetzungen und a-priori-Informationen, das Problem der Rekonstruktion eines Vektorfelds v aus Laufzeitmessungen akustischer Signale möglich ist. Das zugrunde liegende Modell dafür ist die Doppler-Transformation

$$Dv(x, \xi) = \int_{\gamma_{x,\xi}} \frac{dl}{c(y) + \langle v(y), \omega(y) \rangle}.$$

Dabei ist c die Schallgeschwindigkeit und ω der Tangentialvektor entlang der Kurve $\gamma_{x,\xi} : [\tau, 0] \rightarrow M \subset \mathbb{R}^2$, wobei das Signal am Punkt $\gamma_{x,\xi}(0) = x$ in Richtung $\dot{\gamma}_{x,\xi}(0) = \xi$ ausgesandt wird.

Entgegen Nortons Annahme, dass sich die Signale entlang von Geraden ausbreiten, sind die Wege durch den Brechungsindex $\tilde{n} := \frac{c_0}{c}$ bestimmt. Nur im Fall $\tilde{n} \equiv 1$, also in homogenen Medien, trifft Nortons Annahme zu. Im inhomogenen Fall werden die Signale jedoch gebrochen und bewegen sich gemäß des Fermatschen Prinzips entlang von geodätischen Kurven, welche durch die zugrunde liegende Riemannsche Metrik bestimmt sind. Die Metrik wiederum ist durch den Brechungsindex definiert. Eine Vernachlässigung des Brechungsindex in der Rekonstruktion führt also zu einem Modellierungs- und somit zu einem Rekonstruktionsfehler.

Um diese Fehler zu vermeiden soll der Brechungsindex aus den Laufzeitdaten und dem Zusammenhang

$$R\tilde{n}(x, \xi) = \int_{\gamma_{x,\xi}} \tilde{n}(y) dl = \int_{\tau}^0 \tilde{n}(\gamma_{x,\xi}(t)) \|\dot{\gamma}_{x,\xi}(t)\|_2 dt$$

rekonstruiert werden. Dies ist ein nichtlineares inverses Problem, da die Integrationswege vom Integranden abhängen. Insbesondere ist die Frage nach der Existenz und Eindeutigkeit einer Lösung noch nicht umfassend geklärt.

Ziel dieser Arbeit ist es einen Rekonstruktionsalgorithmus zu entwickeln um aus Laufzeitmessungen von akustischen Signalen am Rand von M den Brechungsindex zu rekonstruieren. Hierzu nutzen wir einen lokalen Tikhonov-Ansatz um das Rekonstruktionsproblem stabil lösen zu können. Der Nichtlinearität begegnen wir mit einem adaptiven, iterativen Abstiegsverfahren. Das neu entwickelte Verfahren wird anschließend an diversen Beispielen getestet.

Abstract

In 1988 S.J. Norton showed that the problem to reconstruct a vector field v from time-of-flight (TOF) measurements over a bounded region $M \subset \mathbb{R}^2$ has a unique solution under certain conditions and a priori information. The underlying model is the Doppler transformation

$$Dv(x, \xi) = \int_{\gamma_{x,\xi}} \frac{dl}{c(y) + \langle v(y), \omega(y) \rangle}$$

where c is the velocity of sound and ω is the tangential vector along the curve $\gamma_{x,\xi} : [\tau, 0] \rightarrow M$, $\gamma_{x,\xi}(0) = x$, $\dot{\gamma}_{x,\xi}(0) = \xi$ which describes the path of a signal emitted from x in direction of ξ .

Norton assumed that emitted signals move along straight lines through the object. This corresponds to the situation where the refractive index $\tilde{n} := \frac{c_0}{c}$ is equal to 1. In case of an inhomogeneous medium the ultrasound beams are deflected and, according to Fermat's principle, travel along geodesic curves of a Riemannian metric which is associated with \tilde{n} . The simplified assumption that ultrasound signals move along straight lines therefore leads to reconstruction errors.

The first step to solve the problem is to compute the refractive index from TOF measurements

$$R\tilde{n}(x, \xi) = \int_{\gamma_{x,\xi}^{\tilde{n}}} \tilde{n}(y) dl = \int_{\tau}^0 \tilde{n}(\gamma_{x,\xi}^{\tilde{n}}(t)) \|\dot{\gamma}_{x,\xi}^{\tilde{n}}(t)\|_2 dt,$$

where $\gamma_{x,\xi}^{\tilde{n}}$ denotes the geodesics associated with \tilde{n} . This is a nonlinear, inverse problem whereas the integration paths depend particularly on the integrand. Even existence and uniqueness of solutions are not entirely clarified.

The objective of this thesis is therefore the development of reconstruction algorithms to compute the refractive index in M from TOF measurements at the boundary of M taking physical circumstances like deflection into account. We use a local Tikhonov regularization scheme for solving stably the reconstruction problem. An adaptive iterative descent method is applied to find a minimum in each step. The method's outcome is both, the refractive index \tilde{n} as well as the associated geodesics. Furthermore the algorithm is tested at several examples.

Danksagung

Ich möchte mich an dieser Stelle bei allen bedanken, die mich bei der Erstellung dieser Arbeit sehr unterstützt haben.

Als erstes möchte ich Univ.-Prof. Dr. Thomas Schuster danken. Bei der Ausrichtung des Promotionsthemas hat er mir viele Freiheiten gelassen, so dass ich meinen eigenen Weg gehen konnte. Dabei haben seine Anregungen und kritischen Kommentare maßgeblich zum Gelingen dieser Arbeit beigetragen. Nicht zuletzt wurde er auch durch private Gespräche zu einem wertvollen und freundschaftlichen Mentor.

Ich danke Dr. Frank Schöpfer für seine fruchtbaren Anmerkungen und Ideen in der frühen Phase meiner Promotion, welche mich schließlich die Richtung einschlagen ließen, auf der ich mich derzeit befinde.

Des Weiteren danke ich der Arbeitsgruppe für das angenehme Arbeitsklima und die schönen Momente. Es war mir eine Freude mit ihnen gemeinsam durch diese Zeit zu gehen. Insbesondere danke ich auch Dr. Frederik Heber für seine Zeit, die er für die Korrekturlesung opferte.

Schließlich bedanke ich mich bei meinem Kollegen und mittlerweile guten Freund M.Sc. Jens Tepe. Die vielen fachlichen, wie auch privaten Gespräche haben mich immer wieder aufs neue motiviert. Durch ihn ist meine Promotionszeit im positiven Sinne viel zu schnell vorüber gegangen.

Abschließend möchte ich meiner Frau Marina danken, die mich immer immer wieder aufgebaut hat, wenn es mal nicht so gut lief. Dazu hat zum Ende hin auch meine Tochter Paulina Olga Numerica beigetragen, auf die ich unheimlich stolz bin.

Udo Schröder

Saarbrücken, 5. Mai 2015

Inhaltsverzeichnis

Einleitung	1
1. Problemstellung und Motivation	5
1.1. Vektorfeldtomographie	6
1.2. Euklidische und Riemannsche Metriken	7
1.3. Das inverse Problem der Seismik und das Randsteifigkeitsproblem	9
2. Modellierung und analytische Resultate	13
2.1. Analytische Grundlagen	13
2.2. Modellierung	18
2.2.1. Modellierung mit Geradenstücken als Wege	24
2.2.2. Modellierung mit geodätischen Kurven als Wege	25
2.3. Existenz und Eindeutigkeit	29
3. Geodätische Kurven und das Vorwärtsproblem	37
3.1. Charakteristikenmethode	38
3.1.1. Numerische Umsetzung	42
3.2. Berechnung der Geodäten als Kürzeste	53
3.2.1. Ein Verfahren für das Vorwärtsproblem	57
3.2.2. Numerische Umsetzung	62
4. Regularisierung und Rekonstruktion	71
4.1. Herleitung des Abstiegsverfahrens	75
4.2. Konvergenzanalyse	78
4.3. Diskretisierung und Implementierung	80
4.4. Numerische Ergebnisse	88
5. Zusammenfassung und Ausblick	111
A. Quellcode	115

Einleitung

Verfahren zur zerstörungsfreien Bildgebung, das heißt, eine Darstellung von Gegebenheiten im Inneren von Objekten ohne diese aufzubrechen und somit in der Regel beschädigen zu müssen, spielen in vielen Bereichen in Industrie, Medizin und Forschung eine große Rolle. Dabei sind diverse Eigenschaften innerhalb von gegebenen Objekten von Interesse, die es optisch darzustellen gilt, wie z.B.

- der Dichte und Struktur von Materialien,
- der Stärke und Ausbreitung von elektrischen Feldern oder
- den Fluss von Flüssigkeiten und Gasen.

Die Frage, was genau rekonstruiert werden soll kommt aus der konkreten Fragestellung einer Anwendung. Konkrete Ziele können zum Beispiel

- die optische Darstellung von Gewebedichten in Lebewesen zur Tumorerkennung,
- die Rekonstruktion des Erdinneren zur Auffindung von Rohstoffen,
- die Bestimmung der Gasströmungen in Hochöfen oder
- die Detektion von Materialschäden in Faserverbundstoffen

sein. Einer Rekonstruktion geht eine Messung voraus. Da die Bildgebung zerstörungsfrei sein soll, muss man sich mit indirekten Messungen zufrieden geben, das heißt Messungen sind nur außerhalb des Zielobjektes möglich. So kann man als einfaches Beispiel Röntgenstrahlen ausgehend von einem Sender durch ein Objekt leiten und deren Abschwächung im Objekt an Empfängern (Detektoren) nach dem Austritt messen (vgl. Abbildung 0.1). Eine andere Möglichkeit wäre einen schwachen nuklearen Zerfall im Objekt anzuregen und die Richtung zu detektieren, aus der dieser Zerfall kam.

Letzten Endes entscheidet aber die ausgehende Fragestellung, ob die Art des Messvorgangs mehr oder minder praktikabel ist, denn in irgend einer Art und Weise muss die Messung durch die zu rekonstruierenden Eigenschaften beeinflusst werden. So eignet sich zum Beispiel die Röntgenstrahlung hervorragend zur Bestimmung von Gewebedichten, da die Strahlung umso stärker abgeschwächt wird, je höher die Dichte des durchlaufenden Gewebes ist. Ist man aber beispielsweise an Strömungsfeldern von Flüssigkeiten oder Gasen interessiert, so ist Röntgenstrahlung denkbar ungeeignet, um diese zu messen, da diese hochenergetische Strahlung nahezu nicht vom Strömungsfeld beeinflusst wird.

Die vorliegende Arbeit beschäftigt sich mit Messungen auf der Basis von niedrigerenergeti-

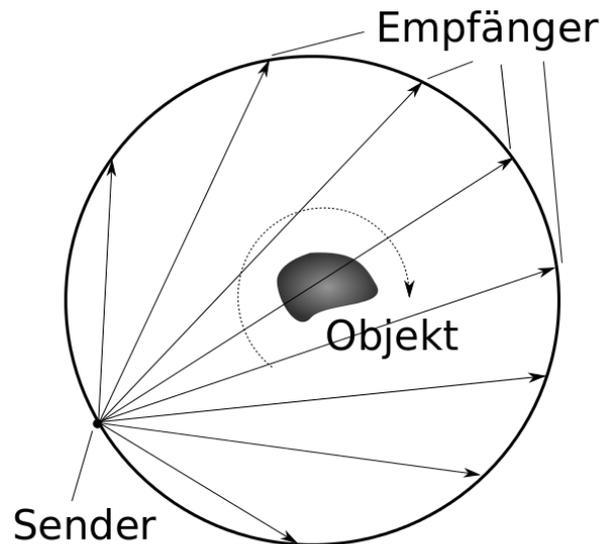


Abbildung 0.1.: Schematische Darstellung einer zerstörungsfreien Prüfung (Fächerstrahlgeometrie). Am Sender werden Signale abgesondert, die das Objekt durchdringen. An den Empfängern werden die Signale aufgenommen. Um mehr Daten zu erhalten wird das Objekt oder die Messvorrichtung gedreht.

schen Strahlungen und Wellen, welche beispielsweise bei der Ultraschalltomographie vorliegen. Messungen durch Ultraschallsignale haben gegenüber CT-Messungen einige Vorteile, so wird in das Messobjekt durch den Messvorgang nahezu keine degenerative Energie abgegeben, zudem können mit diesem Messmedium auch Strömungsfelder gemessen und gegebenenfalls rekonstruiert werden. Der Grund dafür ist, dass die Ultraschallsignale durch Strömungsfelder im Messobjekt mittransportiert werden. Weitere Effekte, die auftreten, sind, dass zum einen die Signale an Grenzflächen gebrochen werden und zum anderen, dass deren Weg durch das Objekt gebeugt wird, also nicht mehr als Gerade modelliert werden kann, sofern das Material inhomogen ist. Die Effekte der Brechung und Beugung sind durch die örtliche Schallgeschwindigkeit determiniert.

Verfahren zur Rekonstruktion von Strömungsfeldern aufgrund von Ultraschallmessungen wurden bereits vorgeschlagen, untersucht und implementiert. Allen ist jedoch gemein, dass entweder die Kenntniss der Schallgeschwindigkeit vorausgesetzt oder deren Einfluss auf die Wege der Signale vernachlässigt wurde. Insbesondere haben Pfitzenreiter und Schuster [2011] ein Verfahren hergeleitet um Vektorfelder zu rekonstruieren unter Berücksichtigung der Schallgeschwindigkeit. Diese wurde zwar auch hier als bekannt vorausgesetzt, jedoch wurde ein Vorschlag gemacht, die Schallgeschwindigkeit selbst iterativ aus den Messungen zu rekonstruieren.

Dieser Idee soll hier nachgegangen werden. Insbesondere wollen wir zunächst in Kapitel 1 die Notwendigkeit der Berücksichtigung der Schallgeschwindigkeit motivieren und das

Problem formulieren, sowie einige Vorarbeiten betrachten. Dazu wird ein mathematischer Zusammenhang bzw. ein mathematisches Modell benötigt, welches die gemessenen Daten (Wirkung, hier: Laufzeit der Signale) mit den ursächlichen Eigenschaften des Körpers (Ursache oder Parameter, hier: örtliche Schallgeschwindigkeit) in einen sinnvollen Zusammenhang bringt. Dieses Modell wird aus physikalischen Zusammenhängen und Gesetzen hergeleitet und führt auf eine Integralgleichung mit Anfangs- und Randbedingungen. Wir werden sehen, dass dazu eine äquivalente Formulierung als Transportgleichung besteht.

Dabei ergeben sich jedoch einige Schwierigkeiten. So ist jedes physikalische Modell eine Vereinfachung mit dem Ziel, die für den beobachteten Effekt relevanten Kenngrößen zu isolieren. Die Wahl des Modells bestimmt sich aus der gewünschten Genauigkeit, der Genauigkeit der Messdaten und dem erforderlichen Rechenaufwand bei der Lösung der Modellgleichungen. Sobald eine Modellierung festgelegt ist und ergeben sich automatisch Modellierungsfehler, die der Diskrepanz zwischen Modell und Wirklichkeit geschuldet sind. Diese Fehler können später nicht mehr "herausgerechnet" werden.

Allgemein beschreibe das mathematische Modell A nun die Wirkung der inneren Struktur des Objekts x , welche es zu rekonstruieren gilt, auf die Signale beim Durchgang. Gemessen werden dann die Daten y . Es gilt also der Zusammenhang

$$x \xrightarrow{A} y.$$

Ziel ist es nun, aus den gegebenen Messdaten y die Ursache x zu rekonstruieren

$$y \xrightarrow{A^{-1}} x,$$

also mathematisch den Operator A zu invertieren. Es stellt sich die Frage, ob und wie das möglich ist. Das Problem wird zudem noch verschärft, da in der Praxis keine genauen Messungen gemacht werden können. Statt der tatsächlichen Wirkung y werden fehlerbesetzte Daten y^ε gemessen. Angenommen der Operator A könnte invertiert werden, so würde man allerdings nicht die Ursache x rekonstruieren, sondern eine Ursache x^ε , welche die Messungen y^ε bewirkt hätte. Doch wie viel haben x und x^ε miteinander gemein? Oder anders gefragt: Wenn der Messfehler $\|y^\varepsilon - y\|$ verringert wird, verringert sich dann auch der Abstand von x zu x^ε ?

Wir werden im 2. Kapitel den mathematisch-physikalischen Zusammenhang zwischen Messdaten und Ursache herleiten und den obigen Fragen nachgehen. Dazu werden zunächst grundlegende Begriffe aus der Differentialgeometrie und Funktionalanalysis eingeführt, welche für die folgenden Betrachtungen notwendig sind. Das Ausgangsproblem wird mathematisch als Lösung einer Operatorgleichung modelliert. Ein wesentlicher Aspekt ist hierbei die Wege der zu messenden Signale zu verfolgen, welche als geodätische Kurven aufgefasst werden können. Anschließend werden wir Eigenschaften des Operators nachweisen und die Gleichung auf Existenz und Eindeutigkeit einer Lösung hin untersuchen. Das 3. Kapitel beschäftigt sich mit der Bestimmung der Laufwege der Signale. Da zur Bestimmung der Schallgeschwindigkeit ein iteratives Verfahren hergeleitet werden soll, ist es essentiell, dass diese effizient und hinreichend genau bestimmt werden können. Daher werden zwei Verfahren vorgestellt um diese numerisch zu berechnen. Das erste Verfahren basiert auf der Lösung eines Systems von gewöhnlichen Differentialgleichungen und

das andere auf der Lösung eines mehrdimensionalen Minimierungsproblems. Abgeleitet werden diese von zwei unterschiedlichen Messvorgängen, welche aus praktischen Gründen naheliegend sind.

Im letzten Kapitel wird dann ein Verfahren hergeleitet um die Ausgangsfrage numerisch beantworten zu können, also die Bestimmung der Schallgeschwindigkeit aufgrund von Laufzeitmessungen akustischer Signale. Hier werden uns jedoch noch einige Probleme begegnen, mit denen wir uns auseinander setzen werden. Unter anderem handelt es sich bei dem abgeleiteten mathematischen Modell aus Kapitel 2 um ein nichtlineares, so dass Standardverfahren zur Lösung von Inversen Problemen nicht direkt zur Anwendung kommen können. Daher wird ein Verfahren hergeleitet, welches lineare Teilprobleme löst und adaptiv die Nichtlinearität des Ausgangsproblems berücksichtigt. Dieses Vorgehen wurde gewählt, um nicht die natürliche Struktur der physikalischen Zusammenhänge zu vernachlässigen. Aus diesem Grund wird also erst sehr spät im Verfahren diskretisiert.

Schließlich werden wir das Verfahren an diversen Beispielrechnungen testen und können dann die numerischen mit den theoretischen Ergebnissen vergleichen, sowie Stärken und Schwächen des Verfahrens aufzeigen.

Kapitel 1

Problemstellung und Motivation

In vielen industriellen, physikalischen, medizinischen oder geophysischen Anwendungen ist man an der Rekonstruktion von Geschwindigkeitsfeldern aus indirekten (Laufzeit-) Messungen interessiert. Mögliche Bereiche wären

- die Bestimmung von Gasbewegungen in Hochöfen (Industrie),
- die Rekonstruktion des Blutflusses in Lebewesen (Medizin oder Biologie),
- die Bestimmung von Flüssen unterhalb der Erdoberfläche (Geologie),
- das Überwachen und Simulieren von Meeresströmungen in nicht direkt zugänglichen Bereichen (Ozeanographie),
- die Berechnung von Windgeschwindigkeiten (Meteorologie) oder
- die Rekonstruktion von Geschwindigkeitsfeldern von Plasma (Plasmaphysik).

Maßgeblich für die Anwendung ist, dass kein direkter Zugang zum Objekt besteht (zum Beispiel ist es nicht möglich Messgeräte direkt in einem Hochofen zu platzieren aufgrund der hohen Temperaturen, welche das Gerät zerstören würden) oder nicht erwünscht ist (Minimierung von invasiven Eingriffen in der Medizin). Zur Messung werden in der Regel Schallsignale (Ultraschall) verwendet, welche durch das Objekt gesendet werden. Denkbar sind aber auch andere schwach energetische Strahlen bzw. Wellen. Sender und Empfänger können dann außerhalb des Objekts positioniert werden.

Physikalisch sind Schallsignale zur Bestimmung von Geschwindigkeitsfeldern prädestiniert, da diese durch das zu rekonstruierende Vektorfeld beeinflusst, also abgebremst oder beschleunigt, werden. Gemessen werden die Laufzeiten der Signale (*Time-of-Flight*, TOF), also die Zeit, die die Signale vom Sender bis zum Empfänger benötigen. Aus diesen Daten soll dann das Strömungsfeld rekonstruiert werden.

Wir wollen kurz ein einfaches Modell beschreiben: Ein Schwimmer soll von einem Punkt A auf direktem Wege, also auf gerader Linie, zu einem Punkt B schwimmen (vgl. Abbildung 1.1). Dabei wird er von einer homogenen Strömung beeinflusst. Existiert keine Strömung, so wird er nach einer gewissen Zeit sein Ziel erreichen. Hat er allerdings eine Gegenströmung, so wird er verlangsamt. Kommt die Strömung von der Seite, so wird er entweder anteilig beschleunigt oder gebremst, je nachdem wie stark die Strömung seiner Bewegung entgegenwirkt oder diese ihn anschiebt, da er seine Bewegung so anpasst, dass er sich auf einer geraden Linie bewegt. Bei einer Strömung von hinten würde er entspre-

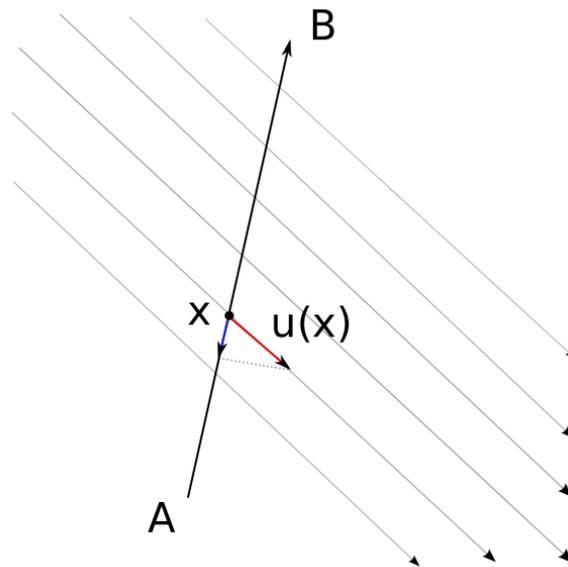


Abbildung 1.1.: Ein Schwimmer bewegt sich von A nach B und wird von einer Strömung u beeinflusst (graue Pfeile). Im Punkt x wirkt die Strömung $u(x)$ (roter Pfeil), so dass sich eine Geschwindigkeitsverringerung für den Schwimmer ergibt (blauer Pfeil), welche als Orthogonalprojektion von $u(x)$ auf die Gerade durch A und B aufgefasst werden kann (gepunktete Strecke).

chend beschleunigt.

Ist nun bekannt, wie lange der Schwimmer ohne Strömung gebraucht hätte, könnte man hoffen aus der Differenz der Zeiten mit und ohne Strömung das durchschnittliche Strömungsfeld auf der Strecke zu rekonstruieren. Nimmt man nun hinreichend viele Punkte zwischen denen sich der Schwimmer bewegt hinzu, so ist zu erwarten das gesamte Strömungsfeld rekonstruieren zu können.

Ein anderes Prinzip, aufgrund dessen man sinnvolle Messungen vornehmen könnte, ist der Doppler-Effekt. Eine Bewegung eines Schallsignals durch ein nichttriviales Vektorfeld führt zu einer Änderung seiner Wellenlänge, welche man messen und mit der Ausgangswellenlänge vergleichen kann.

Beide Prinzipien führen allerdings auf im Wesentlichen das gleiche mathematische Modell, welches durch die Doppler-Transformation repräsentiert und im Kapitel 2 ausführlich erläutert wird.

1.1. Vektorfeldtomographie

Das Problem der Rekonstruktion von Vektorfeldern aufgrund von Laufzeitmessungen wurde von Norton [1988] behandelt. Norton zeigt, dass unter gewissen Annahmen (Divergenzfreiheit des Vektorfelds, euklidische Geometrie) eine Rekonstruktion in zweidimensionalen

Gebieten möglich ist. Unter anderem auf Grundlage dieser Ergebnisse wurden die Erkenntnisse zur Vektortomographie erweitert, wie zum Beispiel

- die Bestimmung von Vektorfeldern in höheren Dimensionen (Lade u. a. [2005], Katsevich und Schuster [2013]),
- die Entwicklung von numerischen Verfahren zur Rekonstruktion von Vektorfeldern aus TOF-Messungen bei euklidischen Geometrien (Wernsdörfer [1993], Osman und Prince [1998]),
- die Rekonstruktion von Vektorfeldern in nicht-euklidischen Metriken (Sharafutdinov [1994])
- und deren numerische Berechnung (Schuster u. a. [2008], Pfitzenreiter und Schuster [2011], Svetov u. a. [2014]).

Insbesondere beschäftigt sich diese Arbeit mit dem letzten genannten Punkt, also der Berechnung von Rekonstruktionen in nicht-euklidischen Geometrien. Tatsächlich geht Norton in seinen Betrachtungen von einer euklidischen Geometrie im Versuchsobjekt aus. Diese Annahme muss allerdings zu Fehlern führen. Wie bereits erwähnt, werden Ultraschallsignale für die Bestimmung von Geschwindigkeitsfeldern deshalb genommen, da die Felder die Signale beeinflussen und somit Laufzeitabweichungen gemessen werden können. Dies wäre zum Beispiel bei Röntgenstrahlen nicht der Fall, wodurch diese auch ungeeignet wären um Vektorfelder zu rekonstruieren. Unglücklicherweise beeinflusst aber auch das Material, durch welches sich das Signal bewegt, ebenfalls dessen Geschwindigkeit. Grundsätzlich bewegen sich Schallsignale gemäß des *Fermatschen Prinzips*, welches besagt, dass die Signale den lokal energetisch günstigsten Weg nehmen, also sich nur in homogenen Medien entlang von Strahlen bewegen.

1.2. Euklidische und Riemannsche Metriken

In homogenen Medien, wie zum Beispiel Luft oder Wasser, bewegen sich Schallsignale entlang von Geraden. Im Allgemeinen muss man jedoch zwei Phänomene berücksichtigen, nämlich ob das Medium

- homogen oder inhomogen bzw.
- isotrop oder anisotrop

ist. Bei homogenen Medien ist die Schallgeschwindigkeit c in jedem Punkt des Gebiets identisch, in inhomogenen Medien kann diese variieren. Bei isotropen Stoffen hat die Richtung, aus der das Signal kommt, keinen Einfluss auf seine Geschwindigkeit, bei anisotropen ist dies jedoch der Fall. Das heißt, dass bei isotropen Medien die Laufzeit von A nach B die gleiche ist, wie von B nach A , bei anisotropen ist dies im Allgemeinen nicht der Fall. Die vorliegende Arbeit beschäftigt sich ausschließlich mit dem inhomogenen, isotropen Fall. Hier kann man einen *metrischen Tensor* g_{ij} auf dem Gebiet durch einen quadra-

tisch antiproportionalen Zusammenhang zur örtlich abhängigen Schallgeschwindigkeit c in jedem Punkt x des Gebiets definieren

$$g_{ij}(x) \propto c^{-2}(x)\delta_{ij},$$

von der wir fordern, dass sie im Gebiet glatt ist. Der metrische Tensor definiert nun wiederum eine *Riemannsche Metrik*, so dass die Wege der Schallsignale durch das Fermatsche Prinzip bestimmt sind, nämlich mathematisch durch das Lösen eines Variationsproblems oder der geodätischen Differentialgleichung. Im ersten Fall wird ein Extremum der Laufzeit über alle möglichen Wege und im zweiten Fall werden lokal kürzeste Kurven bezüglich der Riemannschen Metrik bestimmt. Gilt $c(x) \equiv 1$, so liegt eine Euklidische Metrik zugrunde und die kürzesten Kurven, welche zwei Punkte verbinden, sind Strecken.

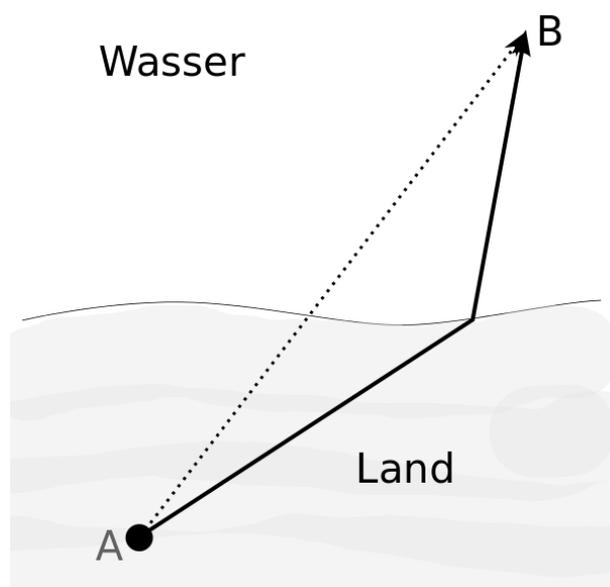


Abbildung 1.2.: Einfluss des Mediums auf die Geschwindigkeit. Eine Person möchte möglichst schnell von A nach B . Der direkte Weg (gestrichelter Pfeil) ist dabei nicht der schnellste, da eine Fortbewegung an Land schneller ist als im Wasser. Der schwarze Pfeil beschreibt einen schnelleren Weg.

Gemäß des Beispiels in Abbildung 1.2 bedeutet dies, dass sich auf der Verbindung von einem Punkt A zu einem Punkt B das Medium ändern würde. Die kürzeste Verbindung wäre dann nicht mehr unbedingt eine gerade Verbindung von A nach B , sondern zum Beispiel eine abknickende Kurve. Befindet sich der Punkt A beispielsweise auf einem Strand und der Punkt B im Wasser, so wäre es vorteilhaft für eine Person sich möglichst lange auf dem Land zu bewegen, da dort seine Geschwindigkeit höher wäre als im Wasser, sofern keine Strömung vorliegt.

In vielen tomographischen Anwendungen kann man davon ausgehen, dass die Inhomogenität des Mediums vernachlässigt werden kann. Dies ist zum Beispiel bei Tomographiearten der Fall, die Röntgenstrahlung einsetzen, denn diese werden in ihrer Bewegung kaum vom

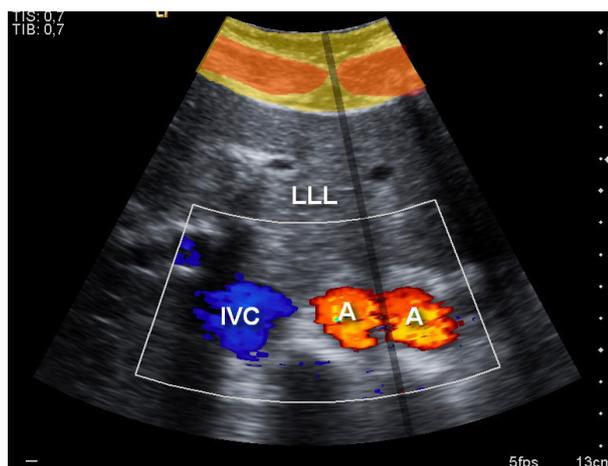


Abbildung 1.3.: Rekonstruktion zweier Aorten (A) aus Ultraschallmessungen. Das Artefakt entsteht wegen einer unterschiedlichen Schallgeschwindigkeit in Fett- und Muskelgewebe. Aus: Wikipedia, http://commons.wikimedia.org/wiki/File:Aorta_duplication_artifact_131206105958250c.jpg, aufgerufen am 16.01.2015.

Material beeinträchtigt. Bei Schallwellen (oder auch optischen Wellen) führt diese Annahme jedoch zu Modellfehlern, wie man der Abbildung 1.3 entnehmen kann. Hier wurden zwei Aorten in einem menschlichen Körper rekonstruiert, obwohl die gescannte Person nur eine besitzt.

Die Ursachen für diese Fehlrekonstruktion sind die Phänomene der Brechung und Beugung, die sich direkt aus dem Fermatschen Prinzip ableiten lassen und die bei der Rekonstruktion aus den Messdaten nicht berücksichtigt wurden. In Abbildung 1.4 ist ein solcher Messfehler schematisch dargestellt. Das blaue Dreieck wird bei der Vernachlässigung von Beugungseffekten nicht am ursprünglichen Ort detektiert, sondern als gestricheltes Dreieck an einer anderen Stelle. Dieses ist zudem verzerrt.

Die örtlichen Schallgeschwindigkeiten, welche letzten Endes die Bewegungskurven der Signale bestimmen, sind im Allgemeinen jedoch nicht bekannt. Will man also die Wege der Signale in dem Modell und der Rekonstruktion berücksichtigen, so müssen diese Kurven zunächst bestimmt werden.

1.3. Das inverse Problem der Seismik und das Randsteifigkeitsproblem

Es stellt sich nun die Frage, ob es überhaupt möglich ist aus Laufzeitmessungen akustischer Signale die zugrunde liegende Metrik, also den metrischen Tensor, des Gebiets eindeutig zu bestimmen. Dieses Problem ist nicht neu, tauchte erstmals in der Geophysik auf und ist als *inverses Problem der Seismik* bekannt. Dieses lautet:

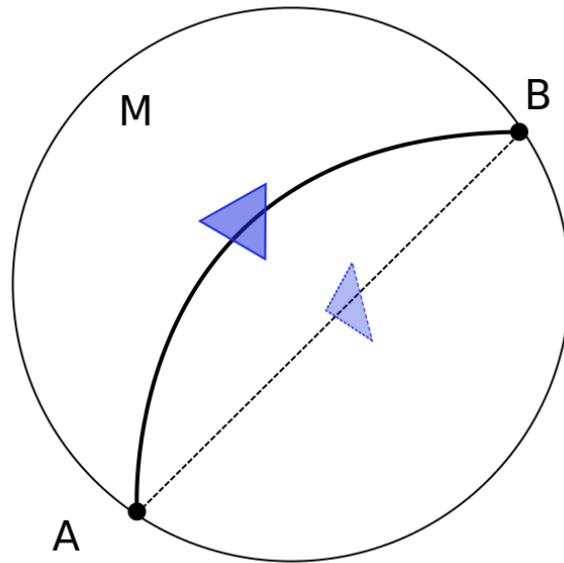


Abbildung 1.4.: Schematische Darstellung eines Messfehlers durch den Einfluss von Beugungen. Die durchgehende Kurve beschreibt den tatsächlichen Weg eines Signals von A nach B , die gestrichelte Linie den angenommenen Weg. Das blaue Dreieck wird daher bei der Vernachlässigung von Beugungseffekten am falschen Ort und verzerrt rekonstruiert (gestricheltes Dreieck).

Kann man die innere Struktur der Erde aus der Kenntnis der Laufzeiten akustischer Signale (seismische Wellen) von einem Ort der Oberfläche zum anderen bestimmen?

Diese Frage wurde von Herglotz [1905], Wiechert und Zoeppritz [1907] zu Beginn des letzten Jahrhunderts gestellt. Herglotz näherte sich der Lösung des Problems indem er annahm, dass die Erde ein Kugel ist und eine radialsymmetrische Struktur besitzt, das heißt die Schallgeschwindigkeit c lässt sich darstellen als

$$c(x) = f(\|x\|_2) \text{ für } x \in B := \{y \in \mathbb{R}^3 \mid \|y\|_2 \leq 1\}.$$

Er konnte nun eine Formel für c aus den Laufzeiten herleiten. Romanov [1967] bewies dann, dass radialsymmetrische Metriken über die Integrale ihrer Geodäten (also gerade die Laufzeiten) bestimmt sind.

Leider wurde recht schnell ersichtlich, dass allgemeine Metriken nicht eindeutig aufgrund von Laufzeitmessungen rekonstruierbar sind. Gibt man sich eine Metrik vor, die die Geodäten und somit die Laufzeiten zwischen allen Randpunkten des Gebiets (das so genannte Sinogramm) erzeugt, so kann man sich eine neue Metrik definieren, welche das gleiche Sinogramm besitzt. Hierzu schaltet man der Ausgangsmetrik einen Diffeomorphismus vor, welcher auf dem Rand des Gebiets die identische Abbildung ist. Ein Diffeomorphismus ist eine bijektive, stetig differenzierbare Abbildung, deren Umkehrabbildung auch stetig

differenzierbar ist.

Es ergeben sich nun folgende Fragen:

- Unter welchen Voraussetzungen ist die Metrik eindeutig aus Laufzeitmessungen auf dem Rand bestimmt? In diesem Fall kann es nur einen dieser Diffeomorphismen geben, nämlich die Identität. (Inverses Problem der Seismik)
- Existiert für alle Metriken, die das gleiche Sinogramm erzeugen, ein Diffeomorphismus, der diese ineinander überführt? Welche Bedingungen an das Gebiet und die Metriken müssen dafür gelten? (Randsteifigkeitsproblem)

Bezüglich der ersten Fragestellung wurde entsprechend nach Eigenschaften gesucht, welche die Eindeutigkeit der Rekonstruktion sicher stellen. Im isotropen, zweidimensionalen Fall bewies Mukhometov [1975] und [1977], dass einfache Metriken durch ihre Sinogramme eindeutig bestimmt sind. Einfache Metriken sind dadurch charakterisiert, dass zwischen je zwei Punkten im Gebiet genau eine verbindende Geodäte existiert. Wenige Jahre später erweiterten Mukhometov [1981] und Romanov [1978] die Ergebnisse unabhängig voneinander auf den dreidimensionalen Fall.

Die Linearisierung des inversen Problems der Seismik führt schließlich zum integralgeometrischen Problem für Tensorfelder (*integral geometry problem for tensor fields*), wobei hier wiederum davon ausgegangen wird, dass sich die Signale gerade durch den Raum bewegen und allgemeine Tensorfelder rekonstruiert werden sollen. Eine Übersicht darüber findet man z. B. bei Sharafutdinov [1994] oder Paternain u. a. [2013].

Die zweite, allgemeinere Frage ist auch als Randsteifigkeitsproblem (*boundary rigidity problem*) bekannt. Anders formuliert lautet die Frage (vergleiche Lassas u. a. [2003]):

Gegeben seien zwei verschiedene Riemannsche Metriken g^1 und g^2 über eine Mannigfaltigkeit M , welche die gleichen Sinogramme liefern, d. h. für je zwei Punkte a und b auf dem Rand der Mannigfaltigkeit sei der Abstand beider Punkte in beiden Metriken gleich, also $d_{g^1}(a, b) = d_{g^2}(a, b)$. Impliziert dies die Existenz eines Diffeomorphismus $\varphi : M \rightarrow M$, so dass $\varphi^*g^2 = g^1$ gilt?

Falls ja, so nennen wir das Problem *randsteif* (boundary rigid). Leider gibt es bisher zu dieser Frage nur wenige allgemeine Antworten. Im zweidimensionalen Fall wurden Ergebnisse für einfache Mannigfaltigkeiten mit konstanten positiven (Michel [1981]) und negativen (Croke [1990], Otal [1990]) Gauß-Krümmungen erzielt. Pestov und Uhlmann [2005] gelang es, einen Beweis für einfache, zweidimensionale Riemannsche Mannigfaltigkeiten zu führen. Unter bestimmten Annahmen, wie dass beide Metriken schon hinreichend dicht beieinander liegen, wurden Aussagen von Croke u. a. [2000] und Stefanov und Uhlmann [1998] gemacht. Weitere Teilerfolge zu speziellen Metriken wurden von Gromov u. a. [1983] (flache Metriken) und Croke [1991, 2004] (SGM-Metriken) erzielt. Eine semiglobale Aussage wurde von Lassas u. a. [2003] getroffen.

Kapitel 2

Modellierung und analytische Resultate

Wir betrachten in diesem Kapitel das folgende Ausgangsproblem

Problemstellung 2.0.1

Gegeben sei ein kompaktes und konvexes Gebiet $M \subset \mathbb{R}^2$. Wie schnell bewegen sich akustische Signale im Inneren des Gebiets, wenn man deren Laufzeiten nur am Rand messen kann?

und möchten zu dessen Lösung ein mathematisches Modell herleiten, welches das Problem und die Zusammenhänge zwischen Messdaten (TOF-Messungen) und den beeinflussenden Größen (Geometrie, metrischer Tensor) hinreichend gut beschreibt.

2.1. Analytische Grundlagen

Zur Modellierung des Problems benötigen wir zunächst einige für uns wesentliche Begriffe und Ergebnisse aus der Differentialgeometrie, wobei eine gewisse Kenntniss der elementaren Grundbegriffe vorausgesetzt wird. Gute Einführungen zu diesem Thema bieten beispielsweise die Bücher von Do Carmo [1976], Jänich [1992] und Kühnel [2013].

Definition 2.1.1 (Riemannsche Metrik)

Es sei M eine n -dimensionale, differenzierbare Mannigfaltigkeit. Eine *Riemannsche Metrik* auf M ist eine differenzierbare Abbildung $M \ni x \rightarrow g_x = g(x)$, so dass

$$g_x : T_x M \times T_x M \rightarrow \mathbb{R}$$

eine positiv definite, symmetrische Bilinearform auf dem Tangentialraum $T_x M$ auf M ist. Das heißt für $\xi, \eta \in T_x M$ gilt

1. $g_x(\xi, \eta) = g_x(\eta, \xi)$,
2. $g_x(\xi, \xi) > 0$, falls $\xi \neq 0$ und
3. für differenzierbare Vektorfelder $X, Y : M \rightarrow TM$ ist $x \rightarrow g_x(X_x, Y_x)$ eine differenzierbare Abbildung.

Hierbei bezeichnet $T_x M$ den Tangentialraum von M in x und TM das Tangentialbündel von M .

Die dritte Bedingung aus Definition 2.1.1 ist äquivalent zu der Forderung, dass die Koeffizientenfunktionen $g_{ij}(x)$ in jeder lokalen Darstellung (das heißt in jeder Karte)

$$g_x = \sum_{i,j=1}^n g_{ij}(x) \cdot dx^i|_x \otimes dx^j|_x \quad (2.1)$$

differenzierbar sind. Man nennt die Koeffizientenfunktionen $(g_{ij})_{ij}$ auch *Metrischen Tensor*.

Beispiel 2.1.2

1. Die Abbildung $g_0 : M \subset \mathbb{R}^n \rightarrow \mathbb{R}$ mit

$$g_0(\eta, \xi) := \eta^T (g_0)_{ij} \xi,$$

wobei der metrische Tensor

$$(g_0)_{ij} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

die Einheitsmatrix ist, ist eine Riemannsche Metrik und heißt *Euklidische Metrik*. $g_0(\cdot, \cdot) = \langle \cdot, \cdot \rangle$ ist somit das Standardskalarprodukt.

2. Für eine differenzierbare Funktion $\tilde{n} : M \subset \mathbb{R}^2 \rightarrow \mathbb{R}_{>0}$ ist die Abbildung $g : M \rightarrow \mathbb{R}$, bestimmt über den metrischen Tensor

$$(g(x))_{ij} = \begin{pmatrix} \tilde{n}^2(x) & 0 \\ 0 & \tilde{n}^2(x) \end{pmatrix},$$

eine Riemannsche Metrik.

In Zukunft wollen wir, sofern es der Übersichtlichkeit dient und eindeutig ist, die Einsteinsche Summenkonvention verwenden, das heißt über doppelt auftauchende Indizes wird summiert und das Summenzeichen weggelassen. Die Darstellung (2.1) hätte dann die Gestalt

$$g_x = g_{ij}(x) \cdot dx^i|_x \otimes dx^j|_x.$$

Jetzt können wir den Begriff der Riemannschen Mannigfaltigkeit definieren.

Definition 2.1.3 (Riemannsche Mannigfaltigkeit)

Es sei M eine n -dimensionale, differenzierbare Mannigfaltigkeit und g eine Riemannsche Metrik auf M . Das Tupel (M, g) heißt dann *Riemannsche Mannigfaltigkeit*.

Beispiel 2.1.4 (Euklidischer Raum)

Das Tupel (\mathbb{R}^n, g_0) , wobei g_0 die Euklidische Metrik aus Beispiel 2.1.2 ist, ist eine Riemannsche Mannigfaltigkeit und heißt *Euklidischer Raum*.

Das Ausgangsproblem werden wir für den Fall $M \subset \mathbb{R}^2$ modellieren. Der Einfachheit halber beziehen wir uns also in den folgenden Definitionen auf den reellen zweidimensionalen Vektorraum. Viele Begriffe lassen sich aber auch mit einigen Hilfsmitteln auf allgemeinere Mannigfaltigkeiten erweitern.

Als nächstes möchten wir den Begriff der Kurve einführen, mit denen wir die Laufwege der akustischen Signale modellieren werden.

Definition 2.1.5 (Kurven und Tangenten)

Es sei $I \subset \mathbb{R}$ ein reelles Intervall. Eine Abbildung $\gamma : \mathbb{R} \rightarrow \mathbb{R}^2$ heißt *parametrisierte Kurve*, falls γ eine stetig differenzierbare Abbildung ist. Ist γ zusätzlich eine Immersion, das heißt

$$\dot{\gamma}(t) := \frac{d\gamma}{dt}(t) \neq 0$$

für alle $t \in I$, so heißt γ *reguläre parametrisierte Kurve*.

Für $t_0 \in I$ heißt

$$\dot{\gamma}(t_0) = \left. \frac{d\gamma}{dt} \right|_{t=t_0}$$

Tangentenvektor von γ in t_0 . Die Gerade

$$L(t_0) = \{\gamma(t_0) + s\dot{\gamma}(t_0) : s \in \mathbb{R}\}$$

heißt *Tangente* an γ in t_0 .

Nun können wir für Kurven deren Längen beziehungsweise Laufzeiten definieren, welche maßgeblich über die zugrunde liegende Riemannsche Metrik bestimmt sind.

Definition 2.1.6 (Kurvenlänge, Laufzeit)

Es sei $M \subset \mathbb{R}^2$ und (M, g) eine Riemannsche Mannigfaltigkeit. Des Weiteren sei eine Kurve $\gamma : I \rightarrow M$ gegeben mit $I = [\tau_0, \tau_1]$. Dann ist die *Länge* (bzw. *Laufzeit*) T von γ gegeben über

$$\begin{aligned} T(\gamma) &= \int_{\gamma} ds \\ &= \int_{\tau_0}^{\tau_1} \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt \end{aligned}$$

Beispiel 2.1.7

1. Für den Euklidischen Raum $(\mathbb{R}^2, g_0 = \langle \cdot, \cdot \rangle)$ ist die Länge einer Kurve $\gamma : [\tau_0, \tau_1] \rightarrow \mathbb{R}^2$ gegeben durch

$$\begin{aligned} T(\gamma) &= \int_{\gamma} ds \\ &= \int_{\tau_0}^{\tau_1} \sqrt{g_0(\dot{\gamma}(t), \dot{\gamma}(t))} dt \\ &= \int_{\tau_0}^{\tau_1} \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle} dt \\ &= \int_{\tau_0}^{\tau_1} \|\dot{\gamma}(t)\| dt. \end{aligned}$$

2. Für die Metrik $g(x) = \tilde{n}^2(x) \delta_{ij} dx^i \otimes dx^j$ aus Beispiel 2.1.2 ist die Länge der Kurve $\gamma : [\tau_0, \tau_1] \rightarrow \mathbb{R}^2$ gegeben durch

$$\begin{aligned} T(\gamma) &= \int_{\gamma} ds \\ &= \int_{\tau_0}^{\tau_1} \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt \\ &= \int_{\tau_0}^{\tau_1} \sqrt{\tilde{n}^2(\gamma(t)) \langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle} dt \\ &= \int_{\tau_0}^{\tau_1} \tilde{n}(\gamma(t)) \|\dot{\gamma}(t)\| dt. \end{aligned}$$

Ist eine Kurve $\gamma : [\tau_0, \tau_1] \rightarrow M$ regulär, so ändert sich ihre Länge nicht unter regulären (orientierungserhaltenden) Parametertransformationen, das heißt für bijektive und stetige Abbildungen $\varphi : [\sigma_0, \sigma_1] \rightarrow [\tau_0, \tau_1]$ mit $\dot{\varphi} > 0$ gilt

$$\begin{aligned} T(\gamma \circ \varphi) &= \int_{\gamma \circ \varphi} ds \\ &= \int_{\sigma_0}^{\sigma_1} \sqrt{g_{\gamma(\varphi(t))}(\dot{\gamma}(\varphi(t))\dot{\varphi}(t), \dot{\gamma}(\varphi(t))\dot{\varphi}(t))} dt \\ &= \int_{\sigma_0}^{\sigma_1} \sqrt{g_{\gamma(\varphi(t))}(\dot{\gamma}(\varphi(t)), \dot{\gamma}(\varphi(t)))} \dot{\varphi}(t) dt \\ &= \int_{\tau_0}^{\tau_1} \sqrt{g_{\gamma(\tilde{t})}(\dot{\gamma}(\tilde{t}), \dot{\gamma}(\tilde{t}))} d\tilde{t} \\ &= T(\gamma) \end{aligned}$$

mit der Substitution $\tilde{t} = \varphi(t)$ bzw. $d\tilde{t} = \dot{\varphi}(t) dt$ und da g eine Bilinearform ist.

Das bedeutet, dass wir reguläre Kurven γ so transformieren können, dass deren Tangentenvektor immer normiert ist, d. h. $\|\dot{\gamma}\| = 1$. Man nennt solche Kurven auch *nach Bogenlänge parametrisiert*. Insbesondere kann man somit auch das Intervall, auf dem eine Kurve definiert ist, immer auf das Intervall $[0, \tau]$ transformieren.

Da nun ein Längenbegriff bezüglich der Metrik eingeführt wurde, liegt es nahe auch den Begriff der Distanz zu definieren.

Definition 2.1.8 (Distanz)

Es sei (M, g) eine Riemannsche Mannigfaltigkeit und $x, y \in M$. Die *Distanz* zwischen x und y ist definiert als

$$d(x, y) := \inf_{\substack{\gamma \in C^\infty(I, M) \\ \gamma(0)=x, \gamma(\tau)=y}} T(\gamma).$$

Wird das Infimum für eine Kurve $\tilde{\gamma}$ in M angenommen, so heißt $\tilde{\gamma}$ *Kürzeste* von x nach y .

Nun betrachten wir noch eine besondere Klasse von Kurven, die Geodäten oder auch geodätische Kurven. Diese sind durch eine Differentialgleichung charakterisiert.

Definition 2.1.9 (Geodäten)

Es sei (M, g) eine Riemannsche Mannigfaltigkeit. Die Kurve $\gamma : I \rightarrow M$ heißt *geodätische Kurve* oder *Geodäte*, falls sie die geodätische Differentialgleichung

$$\ddot{\gamma}^i + \Gamma_{jk}^i(\gamma)\dot{\gamma}^j\dot{\gamma}^k = 0 \quad (2.2)$$

erfüllt. Hierbei sind die Christoffelsymbole Γ_{jk}^i für $x \in M$ gegeben über

$$\Gamma_{jk}^i(x) := \frac{1}{2}g^{ip}(x) \left(\frac{\partial g_{jp}}{\partial x^k}(x) + \frac{\partial g_{kp}}{\partial x^j}(x) - \frac{\partial g_{jk}}{\partial x^p}(x) \right), \quad (2.3)$$

wobei g^{ip} gerade die Inverse des metrischen Tensors g_{ip} sei.

Die Christoffelsymbole hängen nur vom metrischen Tensor ab. Dieser vererbt deswegen auch seine Symmetrieeigenschaften (vgl. Definition 2.1.1) direkt auf die Christoffelsymbole, das heißt es gilt

$$\Gamma_{jk}^i(x) = \Gamma_{kj}^i(x).$$

Gibt man nun Anfangswerte vor, also einen Punkt $\gamma(0) = x \in M$ und eine Richtung $\dot{\gamma}(0) = \xi \in T_x M$, so liefert der Satz von Picard-Lindelöf, dass die geodätische Differentialgleichung (2.2) eindeutig lösbar ist (siehe z. B. Heuser [2009]). Für solch eine Geodäte γ schreiben wir auch $\gamma = \gamma_{x,\xi}$. Um klar zu stellen bezüglich welcher Metrik g die Geodäte definiert ist, schreiben wir auch $\gamma = \gamma_{x,\xi}^g$.

Eine Kurve, die die geodätische Differentialgleichung 2.2 erfüllt, hat insbesondere ein konstantes Geschwindigkeitsvektorfeld entlang derselben. Die Richtungsableitung von γ in Richtung $\dot{\gamma}$ verschwindet also in jedem Punkt. geodätische Kurven kann man also auch darüber charakterisieren, dass sie lokal die kürzeste Verbindung von Punkten in der Mannigfaltigkeit darstellen. Das heißt Kürzeste sind immer Geodäten, da sie insbesondere auch lokal Kürzeste sind. Die Umkehrung gilt nur unter zusätzlichen Voraussetzungen an die Riemannsche Mannigfaltigkeit.

Lemma 2.1.10 (Existenz von Kürzesten)

Es sei $M \subset \mathbb{R}^2$ kompakt und einfach zusammenhängend, das heißt jede geschlossene Kurve lässt sich stetig zu einem Punkt zusammen ziehen (M hat also keine Löcher), und (M, g) eine Riemannsche Mannigfaltigkeit. Für Punkte $x, y \in M$ existiert dann eine geodätische Kurve $\gamma : I = [0, \tau] \rightarrow M$ mit $\gamma(0) = x$ und $\gamma(\tau) = y$, welche Kürzeste ist, das heißt

$$T(\gamma) = d(x, y).$$

Beweis. Das Lemma folgt aus dem Satz von Hopf-Rinow und ist z. B. bei [Jost, 2008, Korollar 1.5.1] zu finden. Insbesondere sind einfache, kompakte Teilmengen des \mathbb{R}^2 metrisch vollständig. \square

Beispiel 2.1.11 (Geodäten)

1. Im Euklidischen Raum (\mathbb{R}^2, g_0) sind alle Geraden Geodäten und umgekehrt. Insbesondere Strecken zwischen zwei Punkten auch Kürzeste.

2. Auf der Sphäre $S^2 := \{x \in \mathbb{R}^3 : \|x\| = 1\}$ mit der von der Euklidischen Metrik des \mathbb{R}^3 induzierten Riemannschen Metrik sind beispielsweise alle Großkreise (das sind Kreise, die entstehen, wenn man die Sphäre mit einer Ebene schneidet, die durch den Ursprung geht) Geodäten. Wählt man zwei verschiedene, nicht gegenüber liegende Punkte $A, B \in S^2$, so ist die Schnittebene zur Erzeugung der Geodäten durch die drei Punkte $\{0, A, B\}$ eindeutig festgelegt. Es entstehen nun genau zwei geodätische Kurven von A nach B , wobei nur eine auch Kürzeste ist (vgl. Abbildung 2.1).

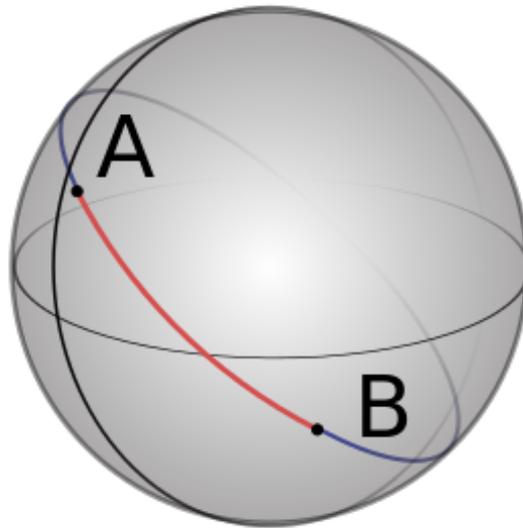


Abbildung 2.1.: Beispiele für geodätische Kurven auf der Sphäre S^2 . Die Punkte A und B werden durch zwei verschiedene geodätische Kurven verbunden, wobei die rote Kürzeste ist, die blaue jedoch nicht.

Abschließend wollen wir noch den Begriff der einfachen Riemannschen Mannigfaltigkeit definieren, welche einen für die Praxis sinnvollen Verlauf der Kürzesten fordert.

Definition 2.1.12 (Einfache Riemannsche Mannigfaltigkeiten)

Sei $M \subset \mathbb{R}^2$ kompakt und (M, g) eine Riemannsche Mannigfaltigkeit mit Rand ∂M . Dann heißt (M, g) *einfach*, falls es zu jedem Punktepaar $x, y \in M$ genau eine (endliche) Kürzeste $\gamma : [0, \tau] \rightarrow M$ gibt mit $\gamma(0) = x$ und $\gamma(\tau) = y$, welche komplett in M verläuft.

2.2. Modellierung

Zur Modellierung des Problems 2.0.1 legen wir uns auf die Einheitskreisscheibe als Teilmenge des zweidimensionalen reellen Raumes fest, das heißt $M := \{x \in \mathbb{R}^2 : \|x\| \leq 1\}$. Der Rand $\partial M = \{x \in \mathbb{R}^2 : \|x\| = 1\}$ ist dann der Einheitskreis und glatt. Außerdem

setzen wir voraus, dass die zu rekonstruierende Riemannsche Mannigfaltigkeit einfach sei. Es werden nun Signale jeweils vom Rand des Gebiets ∂M gesendet und deren Laufzeiten gemessen.

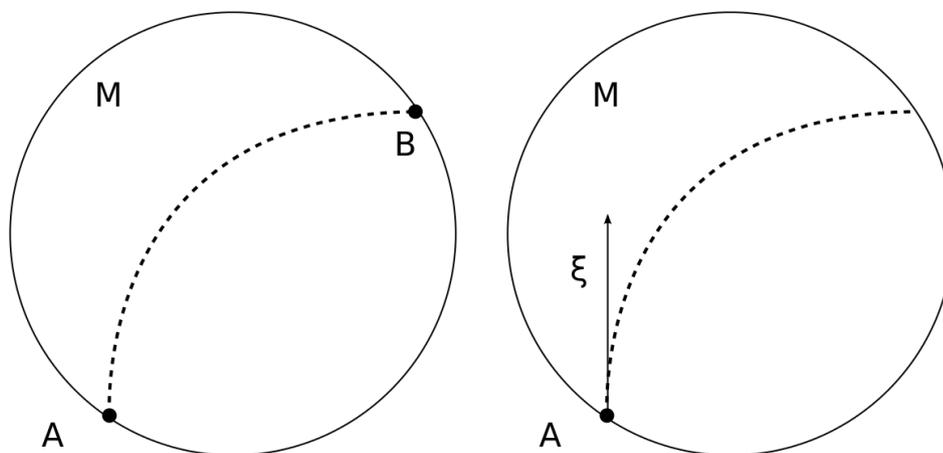


Abbildung 2.2.: Schematische Darstellung der beiden möglichen Messmethoden. *Links:* Messung der Signallaufzeit zwischen den Punkten A und B . *Rechts:* Messung der Signallaufzeit bei Aussendung eines Signals vom Punkt A in Richtung ξ bis zum Auftreffen auf dem Rand.

Jedoch muss festgelegt werden, wie gemessen werden soll, denn es ergeben sich offensichtlich zwei Möglichkeiten (vgl. Abbildung 2.2).

Problemstellung 2.2.1 (Messungen)

Es sind grundsätzlich zwei Messmethoden naheliegend:

1. *Es werden zwei Punkte A und B auf dem Rand ∂M fest gewählt. Vom Punkt A wird eine akustische Welle kreisförmig ausgesandt und es wird die Zeit gestoppt, sobald am Punkt B ein Signal eintrifft.*
2. *Es wird ein Punkt A auf dem Rand ∂M und eine Richtung $\xi \in S^1 := \{x \in \mathbb{R}^2 : \|x\| = 1\}$ fixiert. Nun wird ein Signal vom Punkt A in Richtung ξ geschickt und die Zeit wird gestoppt, sobald das Signal auf den Rand ∂M trifft.*

Die erste Möglichkeit berücksichtigt den Wellencharakter von akustischen Signalen, die zweite Messmethode eher den Teilchencharakter. Wie wir später sehen werden, sind die Messungen gemäß Lemma 2.2.7 mathematisch aber äquivalent.

Als nächstes modellieren wir den metrischen Tensor für inhomogene Medien. Hierfür sei $c : M \rightarrow \mathbb{R}^+$ die Schallgeschwindigkeit in M . Ist c_0 eine (konstante) Referenzschallgeschwindigkeit, zum Beispiel die Schallgeschwindigkeit in Luft oder Wasser, so kann man den Brechungsindex $\tilde{n}_0 : M \rightarrow \mathbb{R}^+$ definieren durch

$$\tilde{n}_0(x) := \frac{c_0}{c(x)}$$

für $x \in M$. Fordern wir $c \in C^\infty(M, \mathbb{R}^+)$, so gilt ebenfalls $\tilde{n}_0 \in C^\infty(M, \mathbb{R}^+)$. Nun können wir eine Metrik \tilde{g} definieren durch

$$\tilde{g}(x) = \tilde{g}_{ij}(x) dx^i dx^j = \tilde{n}_0^2(x) ((dx^1)^2 + (dx^2)^2).$$

Hierbei ist

$$\tilde{g}_{ij}(x) = \tilde{n}_0^2(x) \delta_{ij}$$

der metrische Tensor, wobei abermals die Einsteinsche Summenkonvention verwendet wurde. δ_{ij} ist das Kronecker-Symbol, das heißt $\delta_{ij} = 1$, falls $i = j$ und $\delta_{ij} = 0$, falls $i \neq j$. Normiert man der Einfachheit halber die Referenzschallgeschwindigkeit, setzt also $c_0 = 1$, so erhält man unsere zukünftig betrachtete Messgeometrie.

Lemma 2.2.2 (Messgeometrie)

Es sei $M := \{x \in \mathbb{R}^2 : \|x\| \leq 1\} \subset \mathbb{R}^2$ die Einheitskreisscheibe und $g : M \rightarrow \mathbb{R}$ definiert durch

$$g(x) = g_{ij}(x) dx^i dx^j = \tilde{n}^2(x) ((dx^1)^2 + (dx^2)^2)$$

mit metrischem Tensor

$$g_{ij}(x) = \tilde{n}^2(x) \delta_{ij} = \frac{1}{c^2(x)} \delta_{ij},$$

wobei $\tilde{n} = \frac{1}{c} : M \rightarrow \mathbb{R}^+$ eine positive, differenzierbare Abbildung sei. Dann ist (M, g) eine Riemannsche Mannigfaltigkeit.

Beweis. $M \subset \mathbb{R}^2$ kann M kanonisch in den \mathbb{R}^2 eingebettet werden. Als lokale Karte kann also die Identität gewählt werden und insbesondere sind alle Kartenübergänge differenzierbar. Der metrische Tensor g erfüllt alle Punkte der Definition 2.1.1, da er offensichtlich symmetrisch ($g_{ij} = g_{ji}$), differenzierbar (da die Komponentenfunktionen n differenzierbar sind) und positiv definit (da $\tilde{n} > 0$) ist. \square

In diesem Fall entspricht also der metrische Tensor gerade einer Diagonalmatrix mit gleichen Einträgen auf der Diagonalen. Wir untersuchen jetzt noch die geodätischen Kurven auf ihre stetige Abhängigkeit von der rechten Seite bezüglich unserer Messgeometrie. Zunächst bringen wir hierfür die geodätische Differentialgleichung (2.2)

$$\ddot{\gamma}^i + \Gamma_{jk}^i(\gamma) \dot{\gamma}^j \dot{\gamma}^k = 0$$

mit Anfangswerten $\gamma(0) = x \in M$ und $\dot{\gamma}(0) = \xi \in T_x M$ auf ein System erster Ordnung. Es sei daher

$$\begin{aligned} z_1(t) &:= \gamma_1(t) \\ z_2(t) &:= \gamma_2(t) \\ z_3(t) &:= \dot{z}_1(t) = \dot{\gamma}_1(t) \\ z_4(t) &:= \dot{z}_2(t) = \dot{\gamma}_2(t). \end{aligned}$$

Wir erhalten somit das System erster Ordnung mit $\bar{z}(t) := (z_1(t), z_2(t))^T$

$$\begin{cases} \dot{z}_1(t) & = & z_3(t) \\ \dot{z}_2(t) & = & z_4(t) \\ \dot{z}_3(t) & = & -\Gamma_{ij}^1(\bar{z}(t))z_1^i(t)z_1^j(t) \\ \dot{z}_4(t) & = & -\Gamma_{ij}^2(\bar{z}(t))z_2^i(t)z_2^j(t) \\ \bar{z}(0) & = & x \\ (z_3(0), z_4(0))^T & = & \xi \end{cases} \quad (2.4)$$

Dieses können wir weiter auflösen.

Lemma 2.2.3 (Christoffelsymbole)

Bezüglich unserer Messgeometrie 2.2.2 gilt für die Christoffelsymbole

$$(\Gamma_{ij}^1(x))_{i,j=1}^2 = \tilde{n}^{-1}(x) \begin{pmatrix} \frac{\partial \tilde{n}(x)}{\partial x_1} & \frac{\partial \tilde{n}(x)}{\partial x_2} \\ \frac{\partial \tilde{n}(x)}{\partial x_2} & -\frac{\partial \tilde{n}(x)}{\partial x_1} \end{pmatrix} \quad (2.5)$$

und

$$(\Gamma_{ij}^2(x))_{i,j=1}^2 = \tilde{n}^{-1}(x) \begin{pmatrix} -\frac{\partial \tilde{n}(x)}{\partial x_2} & \frac{\partial \tilde{n}(x)}{\partial x_1} \\ \frac{\partial \tilde{n}(x)}{\partial x_1} & \frac{\partial \tilde{n}(x)}{\partial x_2} \end{pmatrix} \quad (2.6)$$

für alle $x \in M$.

Beweis. Bezüglich des metrischen Tensors

$$g_{ij}(x) = \tilde{n}^2(x)\delta_{ij}$$

und seiner Inversen

$$g^{ij}(x) = g_{ij}^{-1}(x) = \tilde{n}^{-2}(x)\delta_{ij}$$

erhalten wir aus der Definition der Christoffelsymbole (2.3) für $i, j, k = 1, 2$ und $x \in M$

$$\begin{aligned} \Gamma_{ij}^k(x) &= \frac{1}{2}g^{kl}(x) (\partial_i g_{jl}(x) + \partial_j g_{il}(x) - \partial_l g_{ij}(x)) \\ &= \frac{1}{2}g^{k1}(x) (\partial_i g_{j1}(x) + \partial_j g_{i1}(x) - \partial_1 g_{ij}(x)) + \frac{1}{2}g^{k2}(x) (\partial_i g_{j2}(x) + \partial_j g_{i2}(x) - \partial_2 g_{ij}(x)). \end{aligned}$$

Es gilt für $i, j, k = 1, 2$ und $x \in M$

$$\partial_i g_{jl}(x) = \frac{\partial \tilde{n}^2}{\partial x_i}(x)\delta_{jl} = 2\tilde{n}(x)\frac{\partial \tilde{n}}{\partial x_i}(x)\delta_{jl} = 2\tilde{n}(x)\partial_i \tilde{n}(x)\delta_{jl}$$

und wir erhalten (2.5) aus

$$\Gamma_{ij}^1(x) = \frac{1}{2}\tilde{n}^{-2}(x) (\partial_i g_{j1}(x) + \partial_j g_{i1}(x) - \partial_1 g_{ij}(x)),$$

also

$$\begin{aligned}\Gamma_{11}^1(x) &= \frac{1}{2}\tilde{n}^{-2}(x) (\partial_1 g_{11}(x) + \partial_1 g_{11}(x) - \partial_1 g_{11}(x)) = \tilde{n}^{-1}(x) \partial_1 \tilde{n}(x) \\ \Gamma_{12}^1(x) &= \Gamma_{21}^1(x) = \frac{1}{2}\tilde{n}^{-2}(x) (\partial_1 g_{21}(x) + \partial_2 g_{11}(x) - \partial_1 g_{12}(x)) = \tilde{n}^{-1}(x) \partial_2 \tilde{n}(x) \text{ und} \\ \Gamma_{22}^1(x) &= \frac{1}{2}\tilde{n}^{-2}(x) (\partial_2 g_{21}(x) + \partial_2 g_{21}(x) - \partial_1 g_{22}(x)) = -\tilde{n}^{-1}(x) \partial_1 \tilde{n}(x).\end{aligned}$$

Gleichung (2.6) folgt analog. \square

Die stetige Abhängigkeit der Geodäten von der Metrik und den Anfangsdaten folgt nun aus dem folgenden Satz.

Satz 2.2.4 (Stetige Abhängigkeit bei gewöhnlichen Differentialgleichungen)

Es sei $\eta : I \rightarrow M$ auf einem kompakten Intervall $I \subset \mathbb{R}$ die Lösung des Anfangswertproblems

$$\begin{cases} \dot{\eta}(t) &= f(t, \eta(t)) \text{ für } t \in I \\ \eta(t_0) &= p \end{cases}$$

Für $\alpha > 0$ bezeichne

$$S_\alpha := \{(t, x) : t \in I, \|x - \eta(t)\| \leq \alpha\}$$

die α -Umgebung der Kurve. Wenn ein $\alpha > 0$ existiert, so dass die rechte Seite $f(t, x)$ in S_α stetig ist und einer Lipschitzbedingung

$$\|f(t, x_1(t)) - f(t, x_2(t))\| \leq L \|x_1(t) - x_2(t)\| \quad (2.7)$$

genügt, dann hängt die Lösung η stetig vom Anfangswert und von der rechten Seite f ab. Das heißt, für alle $\varepsilon > 0$ existiert ein $\delta > 0$, so dass jede Lösung $\theta(t)$ eines gestörten Anfangswertproblems

$$\begin{cases} \dot{\theta}(t) &= g(t, \theta(t)) \text{ für } t \in I \\ \theta(t_0) &= \tilde{p} \end{cases}$$

mit einem in S_α stetigem g und

$$\|g(t, \eta(t)) - f(t, \eta(t))\| < \delta \text{ und } \|p - \tilde{p}\| < \delta$$

in ganz I existiert und der Ungleichung

$$\|\theta(t) - \eta(t)\| < \varepsilon$$

genügt.

Beweis. Ein Beweis ist in Walter [1990], Kapitel III, Satz VI, zu finden. \square

Es ist nun klar, welche Voraussetzungen unsere Metrik g erfüllen muss, damit wir eine stetige Abhängigkeit erhalten. Dies halten wir im folgenden Korollar fest.

Korollar 2.2.5 (Stetige Abhängigkeit der Geodäten vom Brechungsindex)

Es sei $\tilde{n} \in C^2(M)$. Dann sind die geodätischen Kurven und deren ersten Ableitungen stetig von den Anfangswerten und dem Brechungsindex \tilde{n} abhängig.

Beweis. Indem wir (2.5) und (2.6) einsetzen, erhalten wir aus den Gleichungen (2.4) die rechte Seite f

$$\begin{aligned} f_1(t, z(t)) &= z_3(t), \\ f_2(t, z(t)) &= z_4(t), \\ f_3(t, z(t)) &= -\tilde{n}^{-1}(\bar{z}(t)) \left[\bar{z}^T(t) \begin{pmatrix} \frac{\partial \tilde{n}(x)}{\partial x_1} & \frac{\partial \tilde{n}(x)}{\partial x_2} \\ \frac{\partial \tilde{n}(x)}{\partial x_2} & -\frac{\partial \tilde{n}(x)}{\partial x_1} \end{pmatrix} \bar{z}(t) \right] \text{ und} \\ f_4(t, z(t)) &= -\tilde{n}^{-1}(\bar{z}(t)) \left[\bar{z}^T(t) \begin{pmatrix} -\frac{\partial \tilde{n}(x)}{\partial x_2} & \frac{\partial \tilde{n}(x)}{\partial x_1} \\ \frac{\partial \tilde{n}(x)}{\partial x_1} & \frac{\partial \tilde{n}(x)}{\partial x_2} \end{pmatrix} \bar{z}(t) \right], \end{aligned}$$

wobei $\bar{z}(t) := (z_1(t), z_2(t))^T$ für $t \in I$ sei. Da \tilde{n} zweimal stetig differenzierbar ist, ist f insbesondere stetig. Wir zeigen nun, dass die rechte Seite die Lipschitzbedingung (2.7) aus Satz 2.2.4 erfüllt. Gesucht ist also eine Lipschitzkonstante $L > 0$, so dass

$$\frac{\|f(t, x_1(t)) - f(t, x_2(t))\|}{\|x_1(t) - x_2(t)\|} \leq L \quad (2.8)$$

für alle $(t, x_1), (t, x_2) \in S_\alpha$ gilt. Da f sogar stetig differenzierbar ist, finden wir nach dem Mittelwertsatz eine Zwischenstelle $(t, z) \in S_\alpha$ mit

$$\frac{\|f(t, x_1(t)) - f(t, x_2(t))\|}{\|x_1(t) - x_2(t)\|} = f'(t, z(t)).$$

Es gilt für die partiellen Ableitungen von f an der Stelle $z := z(t)$ und $\bar{z} := \bar{z}(t) = (z_1(t), z_2(t))^T$, $t \in I$

$$\begin{aligned} \partial_1 f_1(z) &= \partial_2 f_1(z) = \partial_4 f_1(z) = 0 \text{ und } \partial_3 f_1(z) = 1, \\ \partial_1 f_2(z) &= \partial_2 f_2(z) = \partial_3 f_2(z) = 0 \text{ und } \partial_4 f_2(z) = 1, \end{aligned}$$

sowie

$$\partial_3 f_3(z) = \partial_4 f_3(z) = \partial_3 f_4(z) = \partial_4 f_4(z) = 0$$

und

$$\begin{aligned} \partial_1 f_3(z) &= \partial_1 \tilde{n}(\bar{z}) \tilde{n}^{-2}(\bar{z}) \left[\bar{z}^T \begin{pmatrix} \frac{\partial \tilde{n}(x)}{\partial x_1} & \frac{\partial \tilde{n}(x)}{\partial x_2} \\ \frac{\partial \tilde{n}(x)}{\partial x_2} & -\frac{\partial \tilde{n}(x)}{\partial x_1} \end{pmatrix} \bar{z}(t) \right] \\ &\quad - \tilde{n}^{-1}(\bar{z}) \left[z_1^2 \frac{\partial^2 \tilde{n}}{\partial x_1^2}(\bar{z}) + 2z_1 \frac{\partial \tilde{n}}{\partial x_1}(\bar{z}) + 2z_1 z_2 \frac{\partial^2 \tilde{n}}{\partial x_1 \partial x_2}(\bar{z}) + 2z_2 \frac{\partial \tilde{n}}{\partial x_2}(\bar{z}) \right]. \end{aligned} \quad (2.9)$$

Die partiellen Ableitungen $\partial_1 f_4(z)$, $\partial_2 f_3(z)$ und $\partial_2 f_4(z)$ ergeben sich ähnlich wie (2.9). Insbesondere ist aber $\sup_{x \in M} \|\nabla_x f(t, x)\|$ beschränkt, da $\tilde{n} \in C^2(M)$ und $\tilde{n} > 0$, so dass für die Wahl $L := \sup_{x \in M} \|\nabla_x f(t, x)\|$ die Lipschitz-Bedingung (2.8) erfüllt ist und die Aussage des Korollars unmittelbar aus Satz 2.2.4 folgt. \square

Zur Modellierung der Wege der akustischen Signale nutzen wir ein physikalisches Phänomen aus, das Fermatsche Prinzip (vgl. Scheck [2006] oder Demtröder [2005]).

Axiom 2.2.6 (Das Fermatsche Prinzip)

Ein (optisches, mechanisches oder akustisches Wellen-) Signal, das sich in einem Medium von einem Punkt zum anderen bewegt, wählt dabei immer den Weg, bei dem die Laufzeit bei kleinen Variationen lokal stationär, also minimal ist. Das bedeutet gerade, dass die Beschleunigung der Wegkurve entlang des Weges verschwindet, was wiederum bedeutet, dass diese Kurven gerade die geodätische Differentialgleichung erfüllen (vgl. auch die Bemerkungen nach Definition 2.1.9). Diese Signale bewegen sich also immer entlang geodätischer Kurven.

Werden die Kurven linearisiert, so gelangt man zum Gebiet der geometrischen Optik. Insbesondere kann man dann unmittelbar aus dem Fermatschen Prinzip das Snelliussche Brechungsgesetz und das Reflexionsgesetz herleiten. Wir gehen allerdings für unsere weiteren theoretischen Überlegungen von glatten Metriken und Kurven aus. In homogenen Medien besagt das Fermatsche Prinzip gerade, dass sich die Signale entlang von Geraden ausbreiten.

Wir wollen nun ein mathematisches Modell herleiten, welches einen Zusammenhang zwischen den Signalen, also deren Wegen (bestimmt durch die zugrunde liegende Metrik) und deren Laufzeit, mit der Metrik herstellen. Hierfür orientieren wir uns zunächst am Vorgehen von Norton [1988], welcher das Problem der Vektorfeldtomographie behandelt.

2.2.1. Modellierung mit Geradenstücken als Wege

Wir betrachten die Einheitskreisscheibe $M \subset \mathbb{R}^2$ mit Rand ∂M , in welchem ein Vektorfeld $v : M \rightarrow \mathbb{R}^2$ definiert sei. Gegeben ist nun eine Laufzeitmessung zwischen den Randpunkten $a, b \in \partial M$ und wir nehmen an, dass sich die akustischen Signale auf Geraden bewegen. Die Laufzeit des Signals ist dann gegeben über

$$T(\gamma) = \int_a^b \frac{dl}{c(x) + \langle v(x), \omega \rangle},$$

wobei dl gerade ein Längenelement auf der Gerade und ω der Tangentenvektor entlang der Geraden sei, also $dl = \sqrt{(dx^1)^2 + (dx^2)^2}$. Physikalisch ist diese Formel sinnvoll, da sie sowohl die jeweilige Schallgeschwindigkeit, als auch die Geschwindigkeitsänderung, die durch das Vektorfeld u in Transportrichtung ω gegeben ist, berücksichtigt. Führt man eine Taylorentwicklung um $v = 0$ durch, so erhält man

$$T(\gamma) \approx \int_a^b \frac{dl}{c(x)} - \int_a^b \frac{\langle v(x), \omega \rangle}{c^2(x)} dl, \quad (2.10)$$

wobei man annimmt, dass die Norm des Vektorfelds $\|v\|$ im Verhältnis zur örtlichen Schallgeschwindigkeit klein ist. Nimmt man nun an, dass der umgekehrte Weg, also von

b nach a , auf derselben Geraden verläuft, welche nur in umgekehrter Richtung $\omega^- = -\omega$ durchlaufen wird (3. Axiom der geometrischen Optik), so erhält man

$$T(\gamma^-) \approx \int_b^a \frac{dl}{c(x)} - \int_b^a \frac{\langle v(x), \omega^- \rangle}{c^2(x)} dl = \int_a^b \frac{dl}{c(x)} + \int_a^b \frac{\langle v(x), \omega \rangle}{c^2(x)} dl. \quad (2.11)$$

Addiert und subtrahiert man nun jeweils die Messungen (2.10) von a nach b und (2.11) von b nach a , so erhält man

$$T(\gamma) + T(\gamma^-) \approx 2 \int_a^b \frac{dl}{c(x)} \quad \text{und} \quad (2.12)$$

$$T(\gamma) - T(\gamma^-) \approx -2 \int_a^b \frac{\langle v(x), \omega \rangle}{c^2(x)} dl. \quad (2.13)$$

Hat man also die Messdaten in beide Richtungen, so kann man hieraus ein zweistufiges Verfahren konstruieren. Im ersten Schritt rekonstruiert man die Schallgeschwindigkeit c aus Gleichung (2.12) und im zweiten Schritt dann das Vektorfeld v aus Gleichung (2.13). In seinem Paper leitet Norton dann die Divergenz und Rotation von v her. Die Annahme, dass sich die Signale auf Geraden ausbreiten führt, neben der Tailorentwicklung in (2.10) und (2.11) unter der Annahme, dass $\frac{\|v\|}{c}$ klein ist, zu einem Modellierungsfehler.

Ist $\tau := \|b - a\|$ und $\gamma : [0, \tau] \rightarrow M$ die Verbindungsstrecke von a und b , also

$$\gamma(t) := a + \frac{t}{\tau}(b - a) = \frac{\tau - t}{\tau}a + \frac{t}{\tau}b,$$

so ergibt sich aus Gleichung (2.13)

$$\begin{aligned} \frac{1}{2} (T(\gamma) + T(\gamma^-)) &\approx \int_a^b \frac{dl}{c(x)} \\ &= \int_0^\tau c^{-1}(\gamma(t)) \|\dot{\gamma}(t)\| dt \\ &= \int_0^\tau \tilde{n}(\gamma(t)) dt \\ &= \int_\gamma ds \end{aligned}$$

mit $ds = \sqrt{\tilde{n}^2(x)((dx^1)^2 + (dx^2)^2)}$ und der Eigenschaft $\|\dot{\gamma}(t)\| = \left\| \frac{b-a}{\|b-a\|} \right\| = 1$ gerade unsere Messgeometrie aus Lemma 2.2.2. Das heißt, wenn die Metrik zu Geradenstücken als Laufwege führen, so stimmt die Modellierung von Norton mit unserer Messgeometrie 2.2.2 überein.

2.2.2. Modellierung mit geodätischen Kurven als Wege

Wir gehen nun davon aus, dass sich die akustischen Signale nicht mehr entlang von Geraden bewegen, sondern deren Wege gemäß dem Fermatschen Prinzip 2.2.6 bestimmt sind.

Ist die örtliche Schallgeschwindigkeit $c : M \rightarrow \mathbb{R}_{>0}$ also nicht konstant, so sind die Wege gekrümmt und folgen geodätischen Kurven, welche durch die geodätische Differentialgleichung 2.1.9 bestimmt sind.

Lemma 2.2.7 (Äquivalenz der Messungen)

Es sei (M, g) unsere Messgeometrie aus Lemma 2.2.2 und zusätzlich einfach. Dann sind die beiden möglichen Messungen aus Problemstellung 2.2.1 äquivalent.

Beweis. Gegeben seien zwei Randpunkte $a, b \in \partial M$. Da (M, g) einfach ist, existiert genau eine Kürzeste $\gamma : [\tau_-, \tau_+] \rightarrow M$, welche die beiden Punkte verbindet, also $\gamma(\tau_-) = a$ und $\gamma(\tau_+) = b$. Entlang dieser Kurve bewegen sich die akustischen Signale. Betrachtet man nun die geodätische Differentialgleichung (2.2)

$$\ddot{\beta}^i + \Gamma_{jk}^i(\beta)\dot{\beta}^j\dot{\beta}^k = 0$$

mit den Anfangswerten

$$\beta(\tau_-) := \gamma(\tau_-) = a$$

und

$$\xi := \dot{\beta}(\tau_-) := \dot{\gamma}(\tau_-),$$

so ist diese lösbar und liefert eine Geodäte $\beta : [\tau_-, \infty) \rightarrow \mathbb{R}^2$ mit $\beta|_{[\tau_-, \tau_+]} = \gamma$ und $\beta(\tau_+) = \gamma(\tau_+) = b$.

Gibt man umgekehrt einen Punkt $a \in \partial M$ vor, so gibt es zu jedem Punkt $a \neq b \in \partial M$ eine Richtung $\xi \in T_a M$, so dass die Lösung der geodätischen Differentialgleichung $\beta : [\tau_-, \infty) \rightarrow \mathbb{R}^2$ gerade $\beta(\tau_+) = b$ erfüllt für ein kleinstes

$$\tau_+ := \min \{ \tau \in (\tau_-, \infty) : \beta(\tau) \cap \partial M \neq \emptyset \},$$

da die Lösung der Differentialgleichung stetig von den Anfangswerten abhängt. Da M konvex ist, liegt die Kurve $\gamma := \beta|_{[\tau_-, \tau_+]}$ ganz in M und ist Kürzeste von a nach b . \square

Für das Lemma ist eine essentielle Voraussetzung, dass die Messgeometrie einfach ist. Andernfalls würde die Lösung der geodätischen Differentialgleichung nicht unbedingt zu einer Kürzesten führen. Im Folgenden beschränken wir daher unsere Betrachtungen auf einfache Metriken.

Definition 2.2.8 (Laufzeitabbildung)

Es sei (M, g) unsere Messgeometrie 2.2.2. Dann heißt

$$u : T^0 M \rightarrow \mathbb{R}^+, (x, \xi) \mapsto \int_{\tau_-(x, \xi)}^0 c^{-1}(\gamma_{x, \xi}(t)) dt$$

Laufzeitabbildung, wobei

$$T^0 M := \{(x, \xi) \in TM : \xi \neq 0\}$$

und $TM = \{(x, \xi) | x \in M, \xi \in T_x M\}$ das Tangentialbündel sei. $\gamma_{x, \xi} : \mathbb{R} \rightarrow M$ sei gerade die nach Bogenlänge parametrisierte Lösung der geodätischen Differentialgleichung mit Anfangswerten $\gamma(0) := x$ und $\dot{\gamma}(0) := \xi$. $\tau_-(x, \xi)$ sei gerade der Zeitpunkt, an dem der Rand ∂M von der Geodäte berührt wird, also

$$\tau_- := \max \{ \tau \in (-\infty, 0] : \gamma_{x, \xi}(\tau) \cap \partial M \neq \emptyset \}.$$

Definition 2.2.8 ist zweckmäßig, sofern x ein innerer Punkt von M ist. Andernfalls muss zur Erfassung der Laufzeitdaten die Richtung der Signale berücksichtigt werden, denn sinnvolle Messungen liegen nur vor, wenn auch wirklich ein Signal durch M gewandert ist.

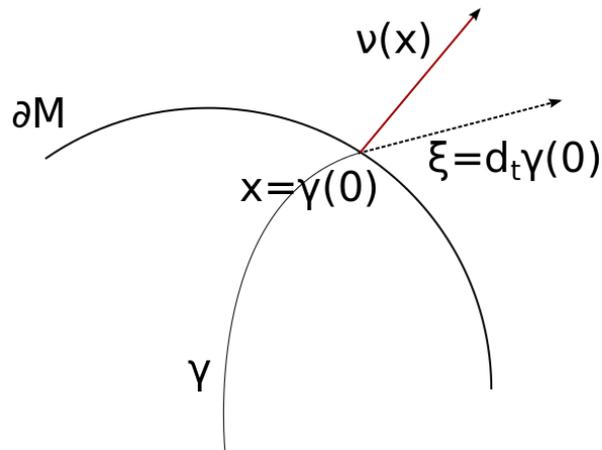


Abbildung 2.3.: Messdaten bezüglich der äußeren Normale. Im Punkt $x \in \partial M$ gilt $\langle \xi, \nu(x) \rangle \geq 0$, wobei $\nu(x)$ die äußere Normale (roter Pfeil) an ∂M und ξ der Richtungsvektor (gestrichelter Pfeil) in x ist. Es liegt hier also ein “Outflow” vor mit Messdaten $T(\gamma) = \int_{\gamma_{\tilde{n}}} \tilde{n}(y) dl \geq 0$. Im Falle eines “Inflows”, also $\langle \xi, \nu(x) \rangle < 0$, läge die Geodäte γ außerhalb des Gebiets M und es wäre $T(\gamma) = 0$.

Ist $x \in \partial M$ ein Randpunkt und $\xi \in T^0 M$ eine Richtung, in die gemessen werden soll, so liegen ausschließlich Messdaten vor, sofern ξ aus das Gebiet heraus zeigt. Das heißt falls $\langle \xi, \nu(x) \rangle \geq 0$, wobei $\nu(x) \in S^1$ die äußere Normale an M in x sei (vgl. Abbildung 2.3).

Definition 2.2.9 (Inflow, Outflow)

Es sei $\Omega M := \{(x, \xi) \in TM : \|\xi\|_g^2 = g_{ij}(x)\xi^i \xi^j = 1\}$. Dann heißt

$$\partial_+ \Omega M := \{(x, \xi) \in \Omega M : x \in \partial M, \langle \xi, \nu(x) \rangle \geq 0\}$$

Outflow und

$$\partial_- \Omega M := \{(x, \xi) \in \Omega M : x \in \partial M, \langle \xi, \nu(x) \rangle < 0\}$$

Inflow, wobei $\nu(x)$ die äußere Normale an M in x sei. Es ist

$$\partial \Omega M := \partial_+ \Omega M \cup \partial_- \Omega M.$$

Für die Aufnahme von Messdaten, welche modellierungsbedingt nur am Rand ∂M vorliegen, muss also unterschieden werden, ob sich die Signale durch das Zielobjekt bewegt haben (Outflow) oder nicht (Inflow). Nichttriviale TOF-Messungen liegen nur im ersteren Fall vor.

Definition 2.2.10 (Messdaten)

Die Abbildung $u^{\text{mess}} \in L^2(\partial\Omega M, \mathbb{R}^+)$ mit

$$u^{\text{mess}}(x, \xi) = \begin{cases} 0 & \text{falls } (x, \xi) \in \partial_- \Omega M \\ \int_{\gamma_{x,\xi}^{\tilde{n}}} \tilde{n}(y) dl & \text{falls } (x, \xi) \in \partial_+ \Omega M \end{cases}, \quad (2.14)$$

wobei $\gamma_{x,\xi}^{\tilde{n}}$ die Lösung der geodätischen Differentialgleichung 2.2 sei bezüglich der durch $\tilde{n} \in C^\infty(M)$ induzierten Metrik und Anfangswerten $\gamma(0) = x$ und $\dot{\gamma}(0) = \xi$, heißt *Messdaten*.

Die Kurven $\gamma_{x,\xi}^{\tilde{n}}$, also die Integrationswege, selbst sind zunächst nicht bekannt, sondern indirekt über die geodätische Differentialgleichung bestimmt. Der Vorwärtsoperator, also die Abbildung, welche einem Brechungsindex \tilde{n} eine Laufzeitabbildung 2.2.8 zuordnet, ist folgendermaßen gegeben.

Definition 2.2.11 (Vorwärtsoperator)

Der *Vorwärtsoperator* $R : L^2(M) \cap C^\infty(M) \rightarrow L^2(\partial\Omega M)$ ist gegeben über

$$R(\tilde{n})(x, \xi) := \int_{\gamma_{x,\xi}^{\tilde{n}}} \tilde{n}(z) dl, \quad \forall (x, \xi) \in \partial\Omega M$$

wobei $\gamma_{x,\xi}^{\tilde{n}}$ die Lösung der geodätischen Differentialgleichung 2.2 sei bzgl. \tilde{n} und Anfangswerten $\gamma(0) = x$ und $\dot{\gamma}(0) = \xi$. Zu einem $a \in C^\infty(M)$ ist der *linearisierte Vorwärtsoperator* $R_a : L^2(M) \rightarrow L^2(\partial\Omega M)$ gegeben durch

$$R_a(\tilde{n})(x, \xi) := \int_{\gamma_{x,\xi}^a} \tilde{n}(z) dl, \quad \forall (x, \xi) \in \partial\Omega M$$

wobei $\gamma_{x,\xi}^a$ die Lösung der geodätischen Differentialgleichung 2.2 sei bzgl. a und Anfangswerten $\gamma(0) = x$ und $\dot{\gamma}(0) = \xi$. Ist $a = \tilde{n}$, so gilt offenbar

$$R_{\tilde{n}}(\tilde{n}) = R(\tilde{n})$$

für $\tilde{n} \in L^2(M) \cap C^\infty(M)$.

Wir können nun das Problem wie folgt definieren.

Problemstellung 2.2.12 (Inverses Problem)

Gegeben seien zu einem differenzierbaren Brechungsindex \tilde{n} , welcher eine einfache Riemannsche Mannigfaltigkeit (M, g) induziert, Laufzeitmessungen akustischer Signale

$u^{\text{mess}} \in L^2(\partial\Omega M, \mathbb{R}^+)$ am Rand von M . Ist es möglich aus diesen Messungen den zugrunde liegenden Brechungsindex zu rekonstruieren? Oder mit anderen Worten, existiert eine eindeutige Lösung $a \in C^\infty(M)$ der Operatorgleichung

$$R(a) = u^{\text{mess}}$$

und falls ja, wie kann man diese berechnen oder approximieren?

Bemerkung 2.2.13

1. Die Schwierigkeit liegt natürlich darin, dass wir Messungen nur auf dem Rand vornehmen können, aber eine Funktion auf ganz M rekonstruieren wollen. Es ist also nicht offensichtlich, dass das Problem überhaupt lösbar ist.
2. Der Zusammenhang der Messdaten mit der zu rekonstruierenden Funktion ist nicht-linear, was wir später durch eine andere, äquivalente Darstellung noch besser sehen werden. Daher versagen die Standardrekonstruktionstechniken, so dass wir einen neuen Algorithmus zur Berechnung des Brechungsindex entwickeln müssen.
3. Das Problem ist schlecht gestellt (siehe Kirsch und Rieder [2014]), so dass für die Rekonstruktion Regularisierungstechniken zum Einsatz kommen müssen. Diese werden wir in Form einer Tikhonov-Regularisierung berücksichtigen.

Wir wollen im Folgenden versuchen, Antworten auf das obige Problem zu geben. Im nächsten Abschnitt beschäftigen wir uns daher mit der Existenz und Eindeutigkeit des Problems (Punkt 1 der Bemerkung) und in den folgenden Kapiteln mit der Rekonstruktion des Brechungsindex (Punkte 2 und 3 der Bemerkung).

2.3. Existenz und Eindeutigkeit

Zunächst wollen wir sicherstellen, dass der Vorwärtsoperator und dessen Linearisierung stetig sind, also dass kleine Änderungen des Brechungsindex nur zu kleinen Änderungen der Laufzeit führen. Dazu parametrisieren wir die Geodäten etwas anders als bisher. Ausgehend von einer vorgegebenen Richtung $\xi \in S^1$, lassen wir die Geodäten nun außerhalb des Gebiets entlang einer Geraden starten. Für $s \in \mathbb{R}$ und $\xi \in S^1$ sei also $\gamma_{\xi,s} : [\tau_-, 0] \rightarrow \mathbb{R}^2$ gegeben als Lösung der geodätischen Differentialgleichung (2.2) mit Anfangswerten

$$\gamma_{\xi,s}(0) = r\xi + s\xi^\perp \text{ und } \dot{\gamma}_{\xi,s}(0) = \xi$$

mit $r > \frac{1}{2} \text{diam}(M) = 1$ (vergleiche Abbildung 2.4). Hierbei sei $\tau_- < 0$ der eindeutige Parameter, bei dem die Geodäte $\gamma_{\xi,s}$ zum zweiten Mal den Rand schneidet. Unter der Annahme der Schnittpunktfreiheit der Geodäten, können wir nun eine Phasenfunktion definieren, welche einem Punkt $y \in M$ und einer Richtung $\xi \in S^1$ einen Startpunkt einer Geodäten zuordnet, die durch y verläuft (vgl. auch das Vorgehen von Pfitzenreiter und Schuster [2011]).

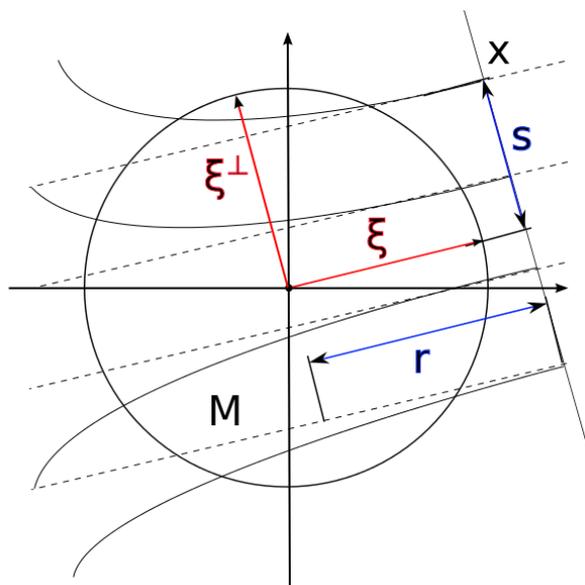


Abbildung 2.4.: Schematische Darstellung der Parametrisierung bzgl. Punkten auf einer Geraden. Bei festgelegter Richtung $\xi \in S^1$ und Normalen $\xi^\perp \in S^1$ (rote Pfeile) kann ein Punkt $x \in \mathbb{R}^2$ durch die Parameter $r, s \in \mathbb{R}$ (blau) über $x = r\xi + s\xi^\perp$ beschrieben werden. Die schwarzen Kurven stellen Geodäten, ausgehend von der Geraden $r\xi + \text{span } \xi^\perp$ in Richtung ξ dar.

Lemma 2.3.1 (Phasenfunktion)

Die Metrik g sei so beschaffen, dass sich für $\xi \in S^1$ und $s, s' \in \mathbb{R}$ mit $s \neq s'$ die Geodäten $\gamma_{\xi,s}$ und $\gamma_{\xi,s'}$ nicht schneiden. Dann existiert eine Phasenfunktion

$$\Phi : M \times S^1 \rightarrow \mathbb{R}, (y, \xi) \mapsto \Phi(y, \xi),$$

so dass die Geodäte $\gamma_{\xi,s}$ implizit durch

$$\Phi(y, \xi) = s \iff y = \gamma_{\xi,s}(t), \quad s \in \mathbb{R}$$

eindeutig festgelegt ist.

Beweis. Diese Abbildung Φ ist wohldefiniert, sofern die Matrix $(\partial_t \gamma_{(x(s), \xi)}(t), \partial_s \gamma_{(x(s), \xi)}(t))$ regulär für festes $\xi \in S^1$ ist. Gemäß Guillemin und Sternberg [1990, Proposition 5.2] ist dies der Fall, wenn sich die Geodäten nicht schneiden, was allerdings vorausgesetzt wurde. \square

Man kann die Phasenfunktion auch als geodätische Projektion auffassen, da sie bei fester Richtung ξ einem Punkt $y \in M$ in gewisser Weise einen Randpunkt $x \in \partial M$ zuordnet, so dass die in x in Richtung ξ definierte Geodäte durch y verläuft. Den Randpunkt $x \in \partial M$ erhält man, indem man der Geodäten nicht bis zur Geraden $r\xi + \text{span } \xi^\perp$, sondern nur bis zum Rand ∂M folgt, also

$$x = \gamma_{\xi,s}(\tau_0),$$

wobei $\tau_0 < 0$ der eindeutige Parameter sei, bei dem die Geodäte $\gamma_{\xi,s}$ zum ersten Mal den Rand schneidet.

Hierfür definieren wir zu gegebener Richtung $\xi \in S^1$ die Menge

$$\Lambda_\xi := \{x = \gamma_{\xi,s}(\tau_0) \in \partial M : s \in \mathbb{R}, \tau_0 = \max\{\gamma_{s,\xi}^{-1}(\gamma_{s,\xi} \cap \partial M)\}\}$$

und schränken Φ auf Λ_ξ ein. Wir erhalten die Abbildung $\Psi_\xi : \Lambda_\xi \rightarrow \mathbb{R}$, gegeben durch

$$\Psi_\xi(x) = \Phi(x, \xi).$$

Diese Abbildung ist nun invertierbar. Für einen Punkt $y \in M$ und eine Richtung $\xi \in S^1$ erhalten wir also den zugehörigen Randpunkt $x \in \partial M$ (vergleiche Abbildung 2.5) durch

$$x = \Psi_\xi^{-1}(\Phi(y, \xi)).$$

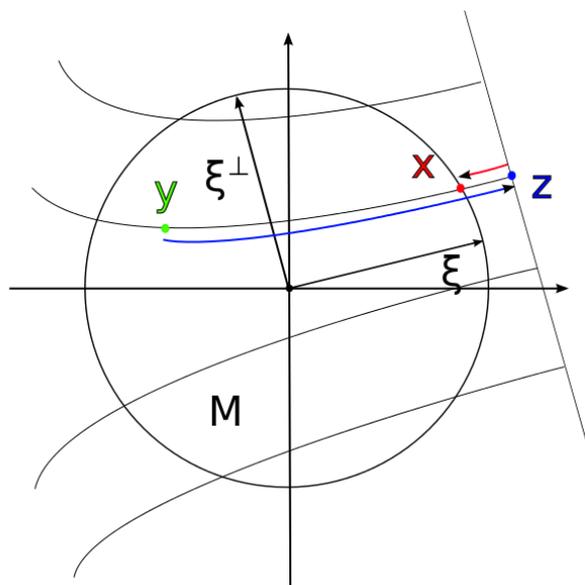


Abbildung 2.5.: Schematische Darstellung der geodätischen Projektion. Der grüne Punkt $y \in M$ wird zunächst durch Φ (blauer Pfeil) auf $z = \Phi(y, \xi)\xi^\perp + r\xi$ (blauer Punkt) abgebildet. Anschließend wird er durch die Abbildung Ψ_ξ^{-1} (roter Pfeil) auf den Rand ∂M projiziert $x = \Psi_\xi^{-1}(z)$ (roter Punkt).

Definition 2.3.2 (Geodätische Projektion)

Die Abbildung $\Pi : TM \times S^1 \rightarrow \partial M$, gegeben durch

$$(y, \xi) \mapsto \Pi(y, \xi) := \Psi_\xi^{-1}(\Phi(y, \xi))$$

heißt *geodätische Projektion* auf den Rand von M .

Bei der Röntgenstrahltomographie fällt der Begriff der geodätischen mit der Euklidischen Projektion zusammen. Für unsere Betrachtungen spielt sie jedoch eine zentrale Rolle, insbesondere taucht sie bei der Rückprojektion in Kapitel 4 auf und muss geeignet approximiert werden.

Beispiel 2.3.3 (Phasenfunktion und geod. Projektion in der eukl. Metrik)

Für die Euklidische Metrik g_0 ist die Phasenfunktion gegeben über

$$\Phi(y, \xi) = \langle y, \xi^\perp \rangle.$$

Die Geodäten des Euklidischen Raumes sind Geraden und können für Anfangswerte $(x = r\xi + s\xi^\perp, \xi) \in M \times S^1$ für $s \in \mathbb{R}$, $r > 1$ über

$$\gamma_{\xi,s}(t) = r\xi + s\xi^\perp + t\xi = s\xi^\perp + (r+t)\xi$$

parametrisiert werden. Es gilt dann für die Phasenfunktion

$$\Phi(\gamma_{\xi,s}(t), \xi) = \langle s\xi^\perp + (r+t)\xi, \xi^\perp \rangle = s$$

für alle $t \in \mathbb{R}$. Zu $\xi \in S^1$ ist

$$\Lambda_\xi = \{x \in \partial M : \langle x, \xi \rangle > 0\}$$

und für $y = \tilde{r}\xi + \tilde{s}\xi^\perp \in M$ mit $\tilde{r}, \tilde{s} \in \mathbb{R}$ ist die geodätische Projektion $\Pi(y, \xi)$ gegeben über

$$\Pi(y, \xi) = \Psi_\xi^{-1}(\Phi(y, \xi)) = \Psi_\xi(\tilde{s}) = \bar{r}\xi + \tilde{s}\xi^\perp$$

mit einem geeignetem $\bar{r} > 0$. Es gilt dann

$$\langle \Phi(y, \xi), \xi \rangle = \langle \bar{r}\xi + \tilde{s}\xi^\perp, \xi \rangle = \bar{r} > 0,$$

also $\Pi(y, \xi) \in \Lambda_\xi$.

Der linearisierte Vorwärtsoperator R_a ist nun gegeben über

$$R_a \tilde{n}(x, \xi) = \int_{\gamma_{x,\xi}^a} \tilde{n}(z) dl = \int_{\Pi(z,\xi)=x} \tilde{n}(z) dl = \int_{\mathbb{R}^2} \tilde{n}(z) \delta(x - \Pi(z, \xi)) dl,$$

wobei δ die Delta-Distribution sei. Nun können wir die Stetigkeit des linearisierten Vorwärtsoperators nachweisen.

Satz 2.3.4 (Stetigkeit des linearisierten Vorwärtsoperators)

Sei $a \in C^\infty(M)$. Der linearisierte Vorwärtsoperator $R_a : L^2(M) \rightarrow L^2(\partial\Omega M)$ mit

$$R_a \tilde{n}(x, \xi) := \int_{\gamma_{x,\xi}^a} \tilde{n}(z) dl,$$

wobei $\gamma_{x,\xi}^a$ die Lösung der geodätischen Differentialgleichung 2.2 sei bzgl. a und Anfangswerten $\gamma_{x,\xi}^a(0) = x$ und $\dot{\gamma}_{x,\xi}^a(0) = \xi$, ist stetig.

Beweis. Für $(x, \xi) \in \partial\Omega M$ und der Geodäten $\gamma_{x,\xi}^a$ bezüglich a folgt mit der Cauchy-Schwarz-Ungleichung

$$\begin{aligned} |R_a(\tilde{n})(x, \xi)|^2 &= \left| \int_{\gamma_{x,\xi}^a} \tilde{n}(z) dl \right|^2 \\ &\leq \int_{\gamma_{x,\xi}^a} 1^2 dl \cdot \int_{\gamma_{x,\xi}^a} |\tilde{n}(z)|^2 dl \\ &= L(\gamma_{x,\xi}^a) \int_{\gamma_{x,\xi}^a} |\tilde{n}(z)|^2 dl, \end{aligned}$$

wobei $C := L(\gamma_{(x,\xi)}) < \infty$ aus der Einfachheit von (M, g) folgt. Ebenfalls folgt daraus die Abschätzung

$$\begin{aligned} \|R_a(\tilde{n})\|_{L^2(\partial\Omega M)}^2 &= \int_{\partial\Omega M} \left| \int_{\gamma_{x,\xi}} \tilde{n}(z) dl \right|^2 d(x, \xi) \\ &\leq C \int_{\partial\Omega M} \int_{\gamma_{x,\xi}} |\tilde{n}(z)|^2 dl d(x, \xi) \\ &= C \int_{S^1} \int_{\|\xi\|_g=1} \int_{\mathbb{R}^2} |\tilde{n}(z)|^2 \delta(\Pi(z, \xi) - x) dl dx d\xi \\ &= C \int_{S^1} \int_M |\tilde{n}(y)|^2 dy d\xi \\ &= \tilde{C} \|\tilde{n}\|_{L^2(M)}^2 \end{aligned}$$

mit $\tilde{C} = 2\pi C$. □

Aufbauend darauf können wir die Stetigkeit auch für den allgemeinen Vorwärtsoperator zeigen.

Satz 2.3.5 (Stetigkeit des Vorwärtsoperators)

Der Vorwärtsoperator $R : L^2(M) \cap C^\infty(M) \rightarrow L^2(\partial\Omega M)$ mit

$$R(\tilde{n}) := \int_{\gamma_{x,\xi}^{\tilde{n}}} \tilde{n}(z) dl,$$

wobei $\gamma_{x,\xi}^{\tilde{n}}$ die Lösung der geodätischen Differentialgleichung 2.2 sei bzgl. \tilde{n} und Anfangswerten $\gamma_{\tilde{n}}(0) = x$ und $\dot{\gamma}_{\tilde{n}}(0) = \xi$, ist stetig.

Beweis. Es gilt für $a, b \in L^2(M) \cap C^\infty(M)$

$$\begin{aligned} &\|R(a) - R(b)\|_{L^2(\partial\Omega M)} \\ &= \|R(a) - R_b(a) + R_b(a) - R(b)\|_{L^2(\partial\Omega M)} \\ &\leq \|R(a) - R_b(a)\|_{L^2(\partial\Omega M)} + \|R_b(a - b)\|_{L^2(\partial\Omega M)}. \end{aligned}$$

Da R_b ein linearer, stetiger Operator ist, lässt sich der rechte Summand abschätzen über

$$\|R_b(a - b)\|_{L^2(\partial\Omega M)} \leq \|R_b\|_{L^2(M) \rightarrow L^2(\partial\Omega M)} \|a - b\|_{L^2(M)}.$$

Für den linken Summanden folgt, sofern $\gamma_{x,\xi}^a$ und $\gamma_{x,\xi}^b$ jeweils die auf $[0, 1]$ parametrisierten Lösungen der geodätischen Differentialgleichungen bezüglich a und b seien,

$$\begin{aligned} & \|R(a) - R_b(a)\|_{L^2(\partial\Omega M)} \\ = & \left\| \int_{\gamma_{x,\xi}^a} a(z) dl - \int_{\gamma_{x,\xi}^b} a(z) dl \right\|_{L^2(\partial\Omega M)} \\ \leq & \left\| \int_0^1 a(\gamma_{x,\xi}^a(t)) \|\dot{\gamma}_{x,\xi}^a(t)\| dt - \int_0^1 a(\gamma_{x,\xi}^b(t)) \|\dot{\gamma}_{x,\xi}^b(t)\| dt \right\|_{L^2(\partial\Omega M)} \\ \leq & \left\| \sup_{y \in M} a(y) \int_0^1 \|\dot{\gamma}_{x,\xi}^a(t) - \dot{\gamma}_{x,\xi}^b(t)\| dt \right\|_{L^2(\partial\Omega M)}. \end{aligned}$$

Für $\|a - b\|_{L^2(M)} < \delta$ für ein hinreichend kleines $\delta > 0$, gilt nun gemäß Korollar 2.2.5 $\|\dot{\gamma}_{x,\xi}^a(t) - \dot{\gamma}_{x,\xi}^b(t)\| < \varepsilon$ für ein $\varepsilon > 0$ und es folgt

$$\begin{aligned} \|R(a) - R(b)\|_{L^2(\partial\Omega M)} & \leq \|R(a) - R_b(a)\|_{L^2(\partial\Omega M)} + \|R_b(a - b)\|_{L^2(\partial\Omega M)} \\ & \leq C_1 \min\{\varepsilon, \delta\} + C_2 \varepsilon \\ & \leq (C_1 + C_2) \varepsilon \end{aligned}$$

mit Konstanten $C_1, C_2 \in \mathbb{R}$. □

Es stellt sich nun die Frage, ob sich aus der Kenntniss der Messdaten u^{mess} der zugrunde liegende Brechungsindex aus dem Vorwärtsoperator 2.2.11 rekonstruieren lässt. Eine Antwort darauf gibt Sharafutdinov [2004] im linearisierten Fall.

Satz 2.3.6 (Existenz und Eindeutigkeit beim linearen inversen Problem)

Es sei $a \in C^\infty(M)$ ein Brechungsindex und dieser induziere eine einfache Riemannsche Metrik. Gegeben sei der linearisierte Vorwärtsoperator $R_a : L^2(M) \cap C^2(M) \rightarrow L^2(\partial\Omega M)$,

$$R_a(\tilde{n}) := \int_{\gamma_a(x,\xi)} \tilde{n}(z) dl,$$

mit entsprechenden Lösungen $\gamma_{x,\xi}^a$ der geodätischen Differentialgleichung (2.2) bezüglich a . Des Weiteren seien die Messdaten u^{mess} aus Definition 2.2.10 gegeben. Dann besitzt die Operatorgleichung

$$u^{\text{mess}} = R_a \tilde{n}$$

genau eine Lösung $\tilde{n} \in L^2(M) \cap C^2(M)$.

Beweis. Den Beweis findet man in Sharafutdinov [2004, Beweis zum Theorem 1.1.1]. □

Aber auch im allgemeinen, nichtlinearen Fall ist die Existenz einer Lösung gesichert, sofern diese hinreichend regulär ist.

Satz 2.3.7 (Existenz und Eindeutigkeit beim nichtlinearen inversen Problem)

Es sei $\tilde{n} \in C^4(M)$ ein Brechungsindex und induziere eine einfache Riemannsche Metrik. Basierend auf \tilde{n} seien die Messdaten u^{mess} gegeben. Dann besitzt die Operatorgleichung

$$u^{\text{mess}} = R(\tilde{n})$$

genau eine Lösung.

Beweis. Der Satz wird bei Sharafutdinov [2004, Theorem 1.2.1] bewiesen. □

Wir haben also die Existenz und Eindeutigkeit einer Lösung des inversen Problems gezeigt. Allerdings wurde das Ergebnis mit starken Voraussetzungen erkaufte, wie zum Beispiel die Einfachheit der Mannigfaltigkeit, die Schnittfreiheit der Geodäten und die Glattheit des metrischen Tensors. Schwächt man diese Forderungen ab oder betrachtet die Fragestellung auf höheren Dimensionen, so gelangt man automatisch zum *boundary rigidity problem* (Randsteifigkeits-Problem), welches nach wie vor im aktuellen Fokus der Forschung steht (vgl. Kapitel 1). Auf allgemeinen Riemannschen Mannigfaltigkeiten ist diese Frage also keineswegs einfach zu beantworten (siehe beispielsweise Croke [1991]).

In der Praxis wird man viele theoretische Voraussetzungen nicht erfüllen können. Trotzdem können wir mit dem Wissen, dass Lösungen unter gewissen Bedingungen existieren, ein Rekonstruktionsverfahren entwickeln und hoffen, dass dieses auch für allgemeine Probleme funktioniert. Daher wenden wir dieses auf viele Beispiele an und können uns so hoffentlich von der Brauchbarkeit des Verfahrens überzeugen.

Das Rekonstruktionsverfahren, welches wir im 4. Kapitel vorstellen wollen, erfordert eine effiziente Lösung des Vorwärtsproblems und somit wollen wir uns diesem im nächsten Kapitel widmen.

Kapitel 3

Geodätische Kurven und das Vorwärtsproblem

In diesem Kapitel werden wir zwei numerische Verfahren herleiten, um das folgende Problem zu lösen.

Problemstellung 3.0.8 (Vorwärtsproblem)

Gegeben sei eine einfache Riemannsche Mannigfaltigkeit (M, g) . Zu

- *einem Randpunkt $x \in \partial M$ mit zugehöriger Richtung $\xi \in T_x M$ oder*
- *zwei Randpunkten $a, b \in \partial M$*

berechne die durchlaufende Geodäte und deren Laufzeit.

Wie im vorherigen Kapitel gezeigt, basieren diese Ansätze auf naheliegende Messarten (Problemstellung 2.2.1). Ziel ist es Verfahren zu entwickeln um das Vorwärtsproblem hinreichend genau und sehr effektiv lösen zu können. Denn zum einen wollen wir basierend auf den Algorithmen zur Lösung des Vorwärtsproblems simulierte Messdaten erzeugen, zum anderen erfordert das in Kapitel 4 entwickelte iterative Verfahren eine wiederholte Lösung des Vorwärtsproblems.

Wir werden zunächst das erste Problem betrachten und die Berechnung der Geodäten auf das Lösen eines Systems von gewöhnlichen Differentialgleichungen mit Anfangswerten zurückführen. Im zweiten Abschnitt dieses Kapitels werden wir uns der zweiten Messmethode widmen, hierfür jedoch einen anderen Lösungsansatz entwickeln. Dieser führt letzten Endes auf das Lösen eines Minimierungsproblems.

Wir werden sehen, dass beide Verfahren sowohl Vor- als auch Nachteile haben. So kann es bei ersteren zu “cold spots” kommen, das sind Bereiche im Gebiet M , die nicht oder nur von sehr wenigen Geodäten durchlaufen werden. Ebenso können die berechneten Geodäten sehr starke Krümmungen aufweisen, was das Rekonstruktionsverfahren aus Kapitel 4 instabil machen kann. Auf der anderen Seite ist dieses Verfahren, welches auf der Charakteristikenmethode basiert, äußerst schnell, so dass viele Geodäten in kurzer Zeit berechnet werden können.

Das zweite Verfahren wiederum benötigt vergleichsweise viel Rechenzeit, erzwingt aber eine gute “Ausleuchtung” des Gebiets. Allerdings müssen zusätzliche, recht starke Annahmen gemacht werden, um die Geodäten durch das Lösen eines Minimierungsproblems berechnen zu können. Je besser die Geodäten durch dieses Verfahren approximiert werden sollen, desto höherdimensional ist das zu lösende Optimierungsproblem, so dass eine gute Approximation mit einer sehr großen Rechenzeit einher geht.

3.1. Charakteristikenmethode

Wir betrachten zunächst die erste Methode, bei der ein Randpunkt und eine Richtung gegeben sind. Ziel dieses Abschnitts ist es, die Charakteristikenmethode zur Bestimmung der Geodäten und deren Laufzeit bei gegebenem Brechungsindex, herzuleiten. Hierfür definieren wir zunächst den geodätischen Fluss.

Definition 3.1.1 (Geodätischer Fluss)

Es sei (M, g) eine Riemannsche Mannigfaltigkeit und $u : T^0M \rightarrow \mathbb{R}^+$ die Laufzeitabbildung aus Definition 2.2.8, also

$$u : T^0M \rightarrow \mathbb{R}^+, (x, \xi) \mapsto \int_{\tau_-(x, \xi)}^0 \tilde{n}(\gamma_{x, \xi}(t)) dt,$$

wobei

$$\tau_- := \max \{ \tau \in (-\infty, 0] : \gamma_{x, \xi}(\tau) \cap \partial M \neq \emptyset \}.$$

der Zeitpunkt des ersten Schnittpunkts mit dem Rand ∂M sei und $\gamma_{x, \xi}$ die Lösung der geodätischen Differentialgleichung 2.2 mit $\gamma_{x, \xi}(0) = x$ und $\dot{\gamma}_{x, \xi} = \xi$ sei. Dann ist der *geodätische Fluss* $Hu : T^0M \rightarrow \mathbb{R}$ definiert als

$$Hu(x, \xi) := \xi^i \frac{\partial u}{\partial x^i} - \Gamma_{jk}^i(x) \xi^j \xi^k \frac{\partial u}{\partial \xi^i} \text{ für alle } (x, \xi) \in T^0M$$

Hierbei seien Γ_{jk}^i die Christoffel-Symbole aus (2.3).

Es ergibt sich nun ein fundamentaler Zusammenhang zwischen dem geodätischen Fluss und dem Brechungsindex bezüglich unserer Messgeometrie.

Satz 3.1.2 (Transportgleichung)

Es sei die Messgeometrie aus Lemma 2.2.2 gegeben, das heißt der metrische Tensor ist gegeben über $g_{ij}(x) = \tilde{n}^2(x) \delta_{ij}$. Dann gilt für alle $(x, \xi) \in T^0M$ der Zusammenhang

$$Hu(x, \xi) = \tilde{n}(x).$$

Beweis. Der Beweis orientiert sich an den Ausführungen von Sharafutdinov [1994]. Sei $(x, \xi) \in T^0M$ und $\gamma = \gamma_{x, \xi} : I = [\tau_-(x, \xi), 0]$ die Geodäte mit $\gamma(0) = x$ und $\dot{\gamma}(0) = \xi$. $\tau_-(x, \xi)$ sei der Zeitpunkt des ersten Schnittpunkts mit dem Rand von M . Für $t_0 \in I$ gilt der Zusammenhang $\gamma_{\gamma(t_0), \dot{\gamma}(t_0)}(t) = \gamma(t + t_0)$ für alle $t \in J := \{s \in I : s + t_0 \in I\}$, denn $\gamma(0 + t_0) = \gamma_{\gamma(t_0), \dot{\gamma}(t_0)}(0) = \gamma(t_0)$ und $\dot{\gamma}(0 + t_0) = \dot{\gamma}_{\gamma(t_0), \dot{\gamma}(t_0)}(0) = \dot{\gamma}(t_0)$. Es gilt dann

$$u(\gamma(t_0), \dot{\gamma}(t_0)) = \int_{\tau_-(x, \xi)}^{t_0} \tilde{n}(\gamma(t)) \|\dot{\gamma}(t)\| dt.$$

Leiten wir die Gleichung nun an der Stelle $t_0 = 0$ ab, so erhalten wir

$$\frac{\partial u}{\partial x^i} \dot{\gamma}^i + \frac{\partial u}{\partial \xi^i} \ddot{\gamma}^i = \tilde{n}(\gamma(0)) \|\dot{\gamma}(0)\|.$$

Unter der Ausnutzung von $\dot{\gamma}(0) = \xi$ und der geodätischen Differentialgleichung (2.2) $\ddot{\gamma} + \Gamma_{jk}^i \dot{\gamma}^j \dot{\gamma}^k = 0$ erhalten wir

$$\tilde{n}(x) = \xi^i \frac{\partial u}{\partial x^i} - \Gamma_{jk}^i(x) \xi^j \xi^k \frac{\partial u}{\partial \xi^i} = Hu(x, \xi).$$

□

Bemerkung 3.1.3

Der Differentialoperator

$$H = \xi^i \frac{\partial}{\partial x^i} - \Gamma_{jk}^i \xi^j \xi^k \frac{\partial}{\partial \xi^i}$$

wird als *geodätisches Vektorfeld* bezeichnet und hat eine geometrische Bedeutung. Ist $G(t; \cdot, \cdot) : TM \rightarrow TM$ der durch

$$G'(t; x, \xi) = H(G(t; x, \xi))$$

definierte zu G gehörende (geodätische) Fluss, so gilt

$$G(t; x, \xi) = (\gamma_{x,\xi}(t), \dot{\gamma}_{x,\xi}(t))$$

für die Geodäten $\gamma_{x,\xi}$.

Wir wollen nun H konkret für unsere Metrik g aus Lemma 2.2.2 angeben.

Korollar 3.1.4

Es sei $M := \{x \in \mathbb{R}^2 : \|x\| \leq 1\} \subset \mathbb{R}^2$ die Einheitskreisscheibe und $g : M \rightarrow \mathbb{R}$ definiert durch

$$g(x) = g_{ij}(x) dx^i dx^j = \tilde{n}^2(x) ((dx^1)^2 + (dx^2)^2)$$

mit metrischem Tensor

$$g_{ij}(x) = \tilde{n}^2(x) \delta_{ij},$$

wobei $\tilde{n} : M \rightarrow \mathbb{R}^+$ eine positive, differenzierbare Abbildung sei. Dann gilt

$$Hu(x, \xi) = \xi^i \frac{\partial u}{\partial x^i}(x, \xi) + \tilde{n}^{-1}(x) \left(\frac{\partial \tilde{n}}{\partial x^i}(x) \|\xi\|^2 - 2\xi^i \langle \xi, \nabla \tilde{n}(x) \rangle \right) \frac{\partial u}{\partial \xi^i}(x, \xi) = \tilde{n}(x) \quad (3.1)$$

für alle $(x, \xi) \in T^0M$.

Beweis. Sei $(x, \xi) \in T^0M$. Für $i = 1$ gilt für die Christoffel-Symbole (vergleiche Lemma 2.2.3)

$$\left(\Gamma_{ij}^1(x) \right)_{i,j=1}^2 = \tilde{n}^{-1}(x) \begin{pmatrix} \frac{\partial \tilde{n}(x)}{\partial x_1} & \frac{\partial \tilde{n}(x)}{\partial x_2} \\ \frac{\partial \tilde{n}(x)}{\partial x_2} & -\frac{\partial \tilde{n}(x)}{\partial x_1} \end{pmatrix}.$$

Es gilt dann für die partiellen Ableitungen

$$\begin{aligned}
\Gamma_{jk}^1(x)\xi^j\xi^k &= \tilde{n}^{-1}(x) \begin{pmatrix} \xi_1 & \xi_2 \end{pmatrix} \begin{pmatrix} \frac{\partial\tilde{n}(x)}{\partial x_1} & \frac{\partial\tilde{n}(x)}{\partial x_2} \\ \frac{\partial\tilde{n}(x)}{\partial x_2} & -\frac{\partial\tilde{n}(x)}{\partial x_1} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} \\
&= \tilde{n}^{-1}(x) \left(\frac{\partial\tilde{n}(x)}{\partial x_1}\xi_1^2 + 2\frac{\partial\tilde{n}(x)}{\partial x_2}\xi_1\xi_2 - \frac{\partial\tilde{n}(x)}{\partial x_1}\xi_2^2 \right) \\
&= \tilde{n}^{-1}(x) \left(2\frac{\partial\tilde{n}(x)}{\partial x_1}\xi_1^2 + 2\frac{\partial\tilde{n}(x)}{\partial x_2}\xi_1\xi_2 - \frac{\partial\tilde{n}(x)}{\partial x_1}\|\xi\|^2 \right) \\
&= \tilde{n}^{-1}(x) \left(2\xi_1 \left[\xi_1 \frac{\partial\tilde{n}(x)}{\partial x_1} + \xi_2 \frac{\partial\tilde{n}(x)}{\partial x_2} \right] - \frac{\partial\tilde{n}(x)}{\partial x_1}\|\xi\|^2 \right) \\
&= \tilde{n}^{-1}(x) \left(2\xi_1 \langle \xi, \nabla\tilde{n}(x) \rangle - \frac{\partial\tilde{n}(x)}{\partial x_1}\|\xi\|^2 \right)
\end{aligned}$$

Analog folgt

$$\Gamma_{jk}^2(x)\xi^j\xi^k = \tilde{n}^{-1}(x) \left(2\xi_2 \langle \xi, \nabla\tilde{n}(x) \rangle - \frac{\partial\tilde{n}(x)}{\partial x_2}\|\xi\|^2 \right)$$

und somit die Behauptung. \square

Bemerkung 3.1.5

1. Durch den Satz 3.1.2 wurde das Lösen der Operatorgleichung aus der Problemstellung 2.2.12 in eine Parameteridentifikation bei der Transportgleichung mit Randwerten (2.14) überführt. Man kann an der Darstellung im Korollar 3.1.4 sehr schön erkennen, dass das Problem nichtlinear ist, da der Brechungsindex \tilde{n} sowohl als Quellterm, als auch als nichtlinearer Parameter auftritt.
2. In der obigen Darstellung der Transportgleichung wurde der Fokus auf den Brechungsindex \tilde{n} gelegt. Geht man allerdings von der Schallgeschwindigkeit $c = \tilde{n}^{-1}$ aus, so erhält man mit der Identität $\frac{\partial\tilde{n}}{\partial x_i}(x) = -c^{-2}(x)\frac{\partial c}{\partial x_i}(x)$ die leicht andere Darstellung der Transportgleichung

$$Hu(x, \xi) = \xi^i \frac{\partial u}{\partial x^i}(x, \xi) - c^{-1}(x) \left(2\frac{\partial c}{\partial x^i}(x)\|\xi\|^2 - 4\xi^i \langle \xi, \nabla c(x) \rangle \right) \frac{\partial u}{\partial \xi^i}(x, \xi) = c^{-1}(x).$$

Beispiel 3.1.6

In homogenen Medien ist $\tilde{n} \equiv 1$ und daher

$$Hu(x, \xi) = \xi^i \frac{\partial u}{\partial x^i} = \langle \xi, \nabla u \rangle = 1,$$

was gerade der Advektions- oder auch Konvektionsgleichung entspricht.

Wir wollen nun aufbauend auf der Transportgleichung (3.1) ein Verfahren herleiten, um die Geodäten zu berechnen. In der Bemerkung 3.1.3 wurde bereits auf einen Zusammenhang zwischen Geodäten und Transportgleichung hingewiesen, den wir nun weiter verfolgen

wollen. Hierzu bedienen wir uns der Charakteristikenmethode.

Es sei $\text{Graph}(u) := \{(x, \xi, u(x, \xi)) \mid (x, \xi) \in TM\}$ der Graph von u und

$$(x(t), \xi(t), z(t)) := (x(t), \xi(t), u(x(t), \xi(t)))$$

eine Kurve auf diesem mit $t \in I$, so dass die Kurve ganz in $\text{Graph}(u)$ verläuft. Differenzieren wir nun nach t , so erhalten wir

$$\left(\dot{x}(t), \dot{\xi}(t), \dot{z}(t) \right) = \left(\dot{x}(t), \dot{\xi}(t), \frac{\partial u(x(t), \xi(t))}{\partial x^i} \dot{x}^i(t) + \frac{\partial u(x(t), \xi(t))}{\partial \xi^i} \dot{\xi}^i(t) \right).$$

Dann ist $(\nabla_x u(x(t), \xi(t)), \nabla_\xi u(x(t), \xi(t)), -1)$ ein Normalenvektor der Kurve, denn es gilt

$$\begin{aligned} & \begin{pmatrix} \nabla_x u(x(t), \xi(t)) \\ \nabla_\xi u(x(t), \xi(t)) \\ -1 \end{pmatrix} \cdot \left(\dot{x}(t), \dot{\xi}(t), \frac{\partial u(x(t), \xi(t))}{\partial x^i} \dot{x}^i(t) + \frac{\partial u(x(t), \xi(t))}{\partial \xi^i} \dot{\xi}^i(t) \right) \\ &= \nabla_x u(x(t), \xi(t)) \cdot \dot{x}(t) + \nabla_\xi u(x(t), \xi(t)) \cdot \dot{\xi}(t) - \frac{\partial u(x(t), \xi(t))}{\partial x^i} \dot{x}^i(t) \\ & \quad - \frac{\partial u(x(t), \xi(t))}{\partial \xi^i} \dot{\xi}^i(t) \\ &= 0. \end{aligned}$$

Die Transportgleichung $Hu = \tilde{n}$ liefert dann, dass $\left(\xi(x), (-\Gamma_{jk}^i(x(t))\xi^j(t)\xi^k(t))_i, \tilde{n}(x(t)) \right)$ ein Tangentenvektor ist, da

$$\begin{aligned} & \begin{pmatrix} \nabla_x u(x(t), \xi(t)) \\ \nabla_\xi u(x(t), \xi(t)) \\ -1 \end{pmatrix} \cdot \left(\xi(x), (-\Gamma_{jk}^i(x(t))\xi^j(t)\xi^k(t))_i, \tilde{n}(x(t)) \right) \\ &= \xi \cdot \nabla_x u(x(t), \xi(t)) - \Gamma_{jk}^i(x(t))\xi^i(t)\xi^j(t)\xi^k(t) \frac{\partial u(x(t), \xi(t))}{\partial \xi^i} - \tilde{n}(x(t)) \\ &= Hu(x(t), \xi(t)) - \tilde{n}(x(t)) \\ &= 0 \end{aligned}$$

gilt. Wählen wir nun einen Randpunkt $x_0 \in \partial M$ und eine Richtung $\xi_0 \in T_{x_0}M$, so erhalten wir nach dem Gleichsetzen der Tangentenvektoren das System erster Ordnung

$$(AWP) \begin{cases} \dot{x}_i(t) = \xi_i(t) & \text{für } t \in I, i = 1, 2 \\ \dot{\xi}_i(t) = -\Gamma_{jk}^i(x(t))\xi^j(t)\xi^k(t) & \text{für } t \in I, i = 1, 2 \\ \dot{z}(t) = \tilde{n}(x(t)) & \text{für } t \in I \\ x(0) = x_0 \\ \xi(0) = \xi_0 \\ z(0) = 0 \end{cases} \quad (3.2)$$

als Anfangswertproblem.

Die Lösung ist eine Charakteristik, also eine Kurve im Graph von u , welcher als Vereinigung aller Charakteristiken aufgefasst werden kann. Die Charakteristiken von H sind aber die Geodäten. Letzten Endes werden also die Geodäten vom Rand zurück verfolgt, was genau der Idee der Messung entspricht. Wir wollen nun noch die Menge der Geodäten zu einem gegebenem Brechungsindex definieren.

Definition 3.1.7 (Menge der Geodäten)

Es sei $a \in C^\infty(M)$ ein Brechungsindex und

$$\mathcal{G}_a := \left\{ \gamma_{x,\xi}^a : [\tau_{x,\xi}^-, 0] \rightarrow M : \gamma_{x,\xi}^a \text{ ist Lösung der geodätischen DGL 2.2 bzgl. } a, \right. \\ \left. \gamma_{x,\xi}^a(0) = x, \dot{\gamma}_{x,\xi}^a(0) = \xi \forall (x, \xi) \in \partial\Omega M \right\}$$

die *Menge der Geodäten* bezüglich a . Sofern klar ist auf welchen Brechungsindex wir uns beziehen, so schreiben wir auch $\mathcal{G} = \mathcal{G}_a$.

3.1.1. Numerische Umsetzung

Um die Geodäten zu berechnen, muss das Anfangswertproblem (3.2) gelöst werden. Im folgenden Abschnitt wird in Lemma 3.2.8 gezeigt, dass Geodäten C^∞ -Abbildungen sind. Demzufolge bietet es sich an ein numerisches Verfahren zu wählen, welches eine hohe Ordnung besitzt. Gleichzeitig soll sichergestellt werden, dass der Zeitaufwand für die Berechnung möglichst gering ist, da das inverse Problem iterativ gelöst werden soll. Demzufolge muss der Vorwärtsoperator mehrfach ausgewertet werden, was in diesem Fall bedeutet, dass die Geodäten und deren Laufzeiten zu gegebenem Brechungsindex \tilde{n}_k bestimmt werden müssen.

Wir rechnen auf der Einheitskreisscheibe $M = \{x \in \mathbb{R}^2 : \|x\| \leq 1\}$ verschiedene Beispiele. Diese wurden in MATLAB implementiert. Zur Berechnung benutzen wir einen Intel Xeon E5-2650 mit 2.0 GHz (max. 2.8 GHz) und 384 GB Arbeitsspeicher. Sofern wir den Code parallelisiert haben, nutzen wir zwölf Kerne.

Zur Lösung des Anfangswertproblems (3.2) benutzen wir eine Schrittweitensteuerung auf Basis der Extrapolationsverfahren. Es wurde das Runge-Kutta-Verfahren 4. Ordnung verwendet. Als Sicherheitsfaktor für die Schrittweitensteuerung wurde $\rho = 0.9$ gewählt, die Anfangsschrittweite ist $h_0 = 0.025$ und die Fehlerschranke für das Extrapolationsverfahren wurde auf $\varepsilon = 10^{-6}$ gesetzt. Wir fassen die Parameter als $\kappa := (\rho, h_0, \varepsilon)$ zusammen.

Für $N_x, N_\xi \in \mathbb{N}$ seien

$$X_{N_x} := \left\{ x_i = \begin{pmatrix} \sin(2\pi \frac{i-1}{N_x}) \\ \cos(2\pi \frac{i-1}{N_x}) \end{pmatrix} : i = 1, \dots, N_x \right\}$$

die Menge der Aufpunkte und

$$\Xi_i^{N_\xi} := \left\{ \xi_i^j = \begin{pmatrix} \sin \left(2\pi \left(\frac{i-1}{N_x} + \frac{j-1}{N_\xi} - \frac{1}{4} \right) \right) \\ \cos \left(2\pi \left(\frac{i-1}{N_x} + \frac{j-1}{N_\xi} - \frac{1}{4} \right) \right) \end{pmatrix} : j = 1, \dots, N_\xi \right\}$$

j	$(\xi_1^j)_1$	$(\xi_1^j)_2$	$T(\gamma_{x_1, \xi_1^j})$
1	-1.000000	0.000000	0.000000
2	-0.982973	0.183750	0.367499
3	-0.932472	0.361242	0.722483
4	-0.850217	0.526432	1.052864
5	-0.739009	0.673696	1.347391
6	-0.602635	0.798017	1.596034
7	-0.445738	0.895163	1.684093
8	-0.273663	0.961826	2.015086
9	-0.092268	0.995734	1.993324
10	0.092268	0.995734	1.991468
11	0.273663	0.961826	1.985405
12	0.445738	0.895163	1.790327
13	0.602635	0.798017	1.596034
14	0.739009	0.673696	1.347391
15	0.850217	0.526432	1.052864
16	0.932472	0.361242	0.722483
17	0.982973	0.183750	0.367499

Tabelle 3.1.: Die approximierten Laufzeiten vom Punkt $x_1 = (0, 1)^T$ in die Richtungen ξ_1^j , $j = 1, \dots, 17$ für das Beispiel “Peaks”.

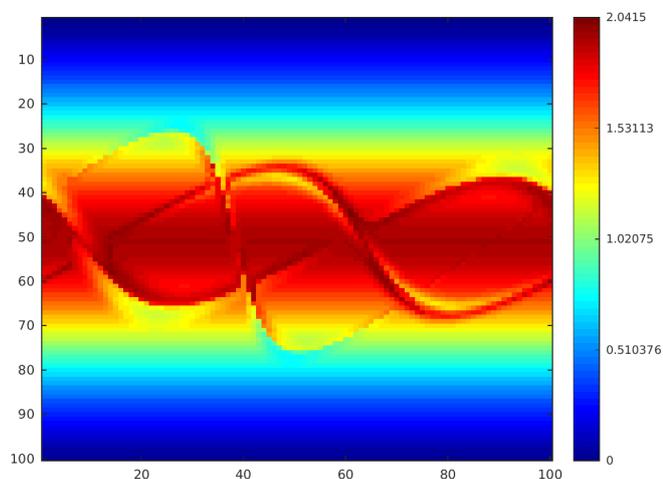


Abbildung 3.1.: Sinogramm “Peaks” für die Parameter $\kappa := (0.9, 0.025, 10^{-6})$, $N_x = 100$ und $N_\xi = 100$. Auf der x-Achse sind die Aufpunkte $x_i \in X_{N_x}$ und auf der y-Achse die Richtungen $\xi_i^j \in \Xi_i^{N_\xi}$ abgetragen. Der Wert im Plot ist gerade die Laufzeit $T(\gamma_{x_i, \xi_i^j})$ der approximierenden Geodäten für $i = 1, \dots, N_x$, $j = 1, \dots, N_\xi$.

die Menge der Richtungen für $i = 1, \dots, N_x$. Dann ist ein Datensatz $\bar{\Gamma}_{N_x, N_\xi}^\kappa$ zum Diskretisierungsparameter $\kappa \in \mathbb{R}^3$ gegeben durch

$$\bar{\Gamma}_{N_x, N_\xi}^\kappa := \left\{ \gamma_{x_i, \xi_i^j} : \gamma_{x_i, \xi_i^j} \text{ ist Näherungslösung des AWP (3.2),} \right. \\ \left. x_i \in X_{N_x}, \xi_i^j \in \Xi_i^{N_\xi}, i = 1, \dots, N_x, j = 1, \dots, N_\xi \right\}.$$

In den folgenden Beispielen wurden für drei Aufpunkte, also $N_x = 3$, $x_1 = (0, 1)^T$, $x_2 = \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)^T$ und $x_3 = \left(-\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)^T$, jeweils $N_\xi = 17$ äquidistant verteilte Richtungen $\Xi_i^{N_\xi}$, $i = 1, 2, 3$, konstruiert. Durch die Definition von $\Xi_i^{N_\xi}$ ist sichergestellt, dass alle dadurch erzeugte Geodäten vom Typ *Outflow* sind. Die approximierenden Geodäten $\bar{\Gamma}_{N_x, N_\xi}^\kappa$ und deren Laufzeiten $T(\gamma_{x_i, \xi_i^j})$, $i = 1, 2, 3$, $j = 1, \dots, 17$, wurden berechnet.

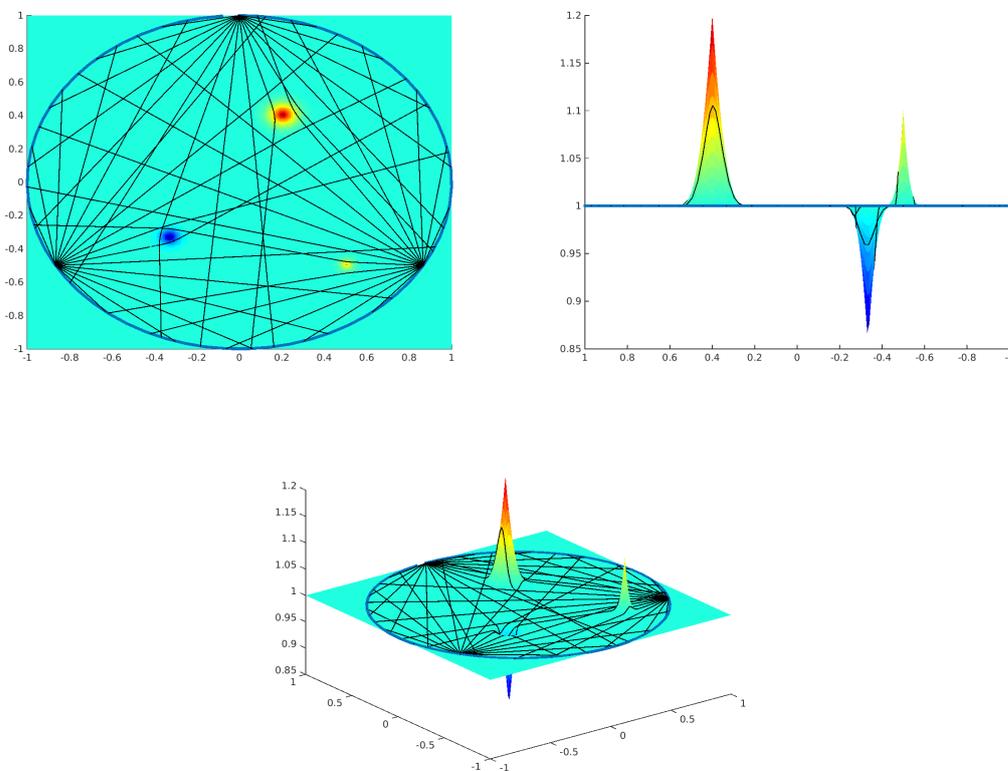


Abbildung 3.2.: Schallgeschwindigkeit c des Beispiels “Peaks” mit approximierenden Geodäten $\bar{\Gamma}_{N_x, N_\xi}^\kappa$ für $N_x = 3$, $N_\xi = 17$ und $\kappa := (0.9, 0.025, 10^{-6})$.

Beispiel 3.1.8 (Peaks)

In diesem Beispiel betrachten wir eine Funktion, die fast überall identisch Eins ist und an wenigen Stellen sehr steile Peaks hat. Wir modellieren die Schallgeschwindigkeit als

$$c(x) = 1 + \sum_{i=1}^3 \varphi_i(x) \chi_i(x)$$

mit

$$\varphi_i(x) = \theta_i e^{-\frac{1}{r_i - \|x - p_i\|}}$$

und

$$\chi_i(x) = \begin{cases} 1 & \text{falls } \|x - p_i\| \leq r_i \\ 0 & \text{sonst} \end{cases}$$

für gegebene Zentren $p_i \in M$, Radien $r_i > 0$ und Amplituden $\theta_i \in (-1, \infty)$, $i = 1, 2, 3$. Für die Beispielrechnung wählen wir als Parameter die Zentren

$$p_1 = \begin{pmatrix} \frac{1}{5} \\ \frac{2}{5} \end{pmatrix}, p_2 = \begin{pmatrix} -\frac{1}{3} \\ -\frac{1}{3} \end{pmatrix}, p_3 = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$$

die Radien

$$r_1 = \frac{1}{4}, r_2 = \frac{1}{5}, r_3 = \frac{1}{6}$$

und die Amplituden

$$\theta_1 = \frac{1}{5}, \theta_2 = -\frac{3}{20}, \theta_3 = \frac{1}{10}.$$

Wir erhalten die Tabelle 3.1 mit den berechneten Laufzeiten für den Aufpunkt x_1 . Bei der Euklidischen Metrik entspricht die Laufzeit eines Signals gerade der Länge der Strecke vom Aufpunkt bis zum Rand. Im Gegensatz dazu geht bei diesem Verfahren sowohl die Schallgeschwindigkeit ein, als auch die Länge der Kurve, so dass die Daten nur bedingt verglichen werden können.

Die Gesamtheit aller Messdaten kann in einem Sinogramm dargestellt werden (Abbildung 3.1). Hier wurde allerdings ein größerer Datensatz berechnet, nämlich $\bar{\Gamma}_{100,100}^{\kappa}$, also 10.000 Geodäten. Dabei sind auf der x-Achse die Aufpunkte $x_i \in X_{N_x}$ und auf der y-Achse die Richtungen $\xi_i^j \in \Xi_i^{N_\xi}$ abgetragen. Der Wert im Plot ist gerade die Laufzeit $T(\gamma_{x_i, \xi_i^j})$ der approximierenden Geodäten für $i = 1, \dots, 100$, $j = 1, \dots, 100$. Im Sinogramm erkennt man deutlich den Einfluss der “Peaks” auf die Laufzeiten.

Abbildung 3.2 zeigt die Schallgeschwindigkeit und die approximierenden Geodäten aus verschiedenen Blickwinkeln, dabei sind die Plots alle gleich skaliert. Bei der Draufsicht kann man sehr schön erkennen, wie die Geodäten durch die “Peaks” beeinflusst werden. Es kommt aber nicht zu “cold spots”, also Bereichen mit wenigen oder keinen Geodäten, die diese durchlaufen. Das kann man gut am “Peak” in $(0.2, 0.4)$ beobachten. Die Geodäten, ausgehend vom oberen Randpunkt $x_1 = (0, 1)^T$ werden durch den “Peak” abgelenkt, so dass hinter diesem (vom Punkt x_1 aus gesehen) keine Geodäten mehr verlaufen. Jedoch füllen die Geodäten, ausgehend von den anderen beiden Punkten x_2 und x_3 , diesen Bereich wieder auf, so dass die Schallgeschwindigkeit in diesem Bereich durchaus erfasst wird.

Die Rechenzeit für den Datensatz $\bar{\Gamma}_{3,17}^{\kappa}$ betrug etwa zweieinhalb Sekunden und für den Datensatz $\bar{\Gamma}_{100,100}^{\kappa}$ etwa zwei Minuten.

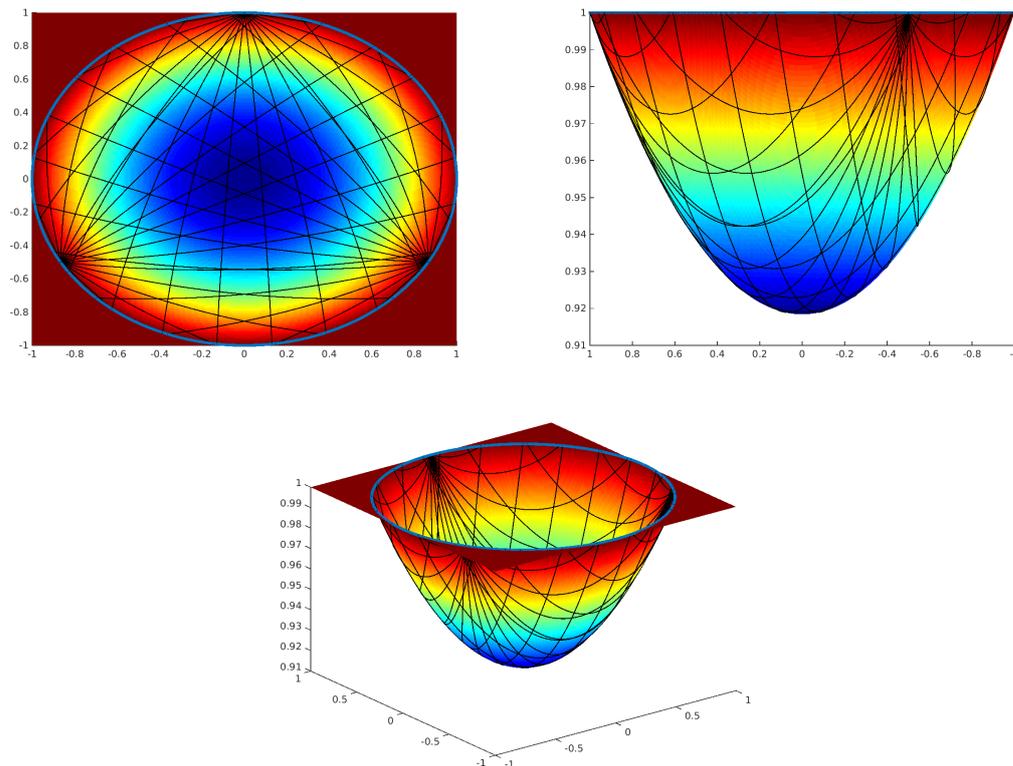


Abbildung 3.3.: Schallgeschwindigkeit c des Beispiels “Negative Krümmung” mit approximierenden Geodäten $\bar{\Gamma}_{N_x, N_\xi}^\kappa$ für $N_x = 3$, $N_\xi = 17$ und $\kappa := (0.9, 0.025, 10^{-6})$.

Beispiel 3.1.9 (Konstante Krümmung)

Wir betrachten hier sowohl eine konstant negative, als auch konstant positive Krümmung. Diese erhält man für die Schallgeschwindigkeit

$$c(x) = 1 + \varphi_\pm(x)\chi_M(x)$$

mit

$$\varphi_-(x) = \frac{(R^2 + a^2\|x\|^2)^2 - (R^2 + a^2)^2}{4R^2}$$

bei negativer und

$$\varphi_+(x) = \frac{(R^2 - a^2\|x\|^2)^2 - (R^2 - a^2)^2}{4R^2}$$

bei positiver Krümmung und charakteristischer Funktion

$$\chi_M(x) = \begin{cases} 1 & \text{falls } \|x\| \leq 1 \\ 0 & \text{sonst} \end{cases}$$

für Parameter $a^2 < R^2$. Je größer man $|a|$ im Verhältnis zu $|R|$ wählt, desto gekrümmter ist die Oberfläche.

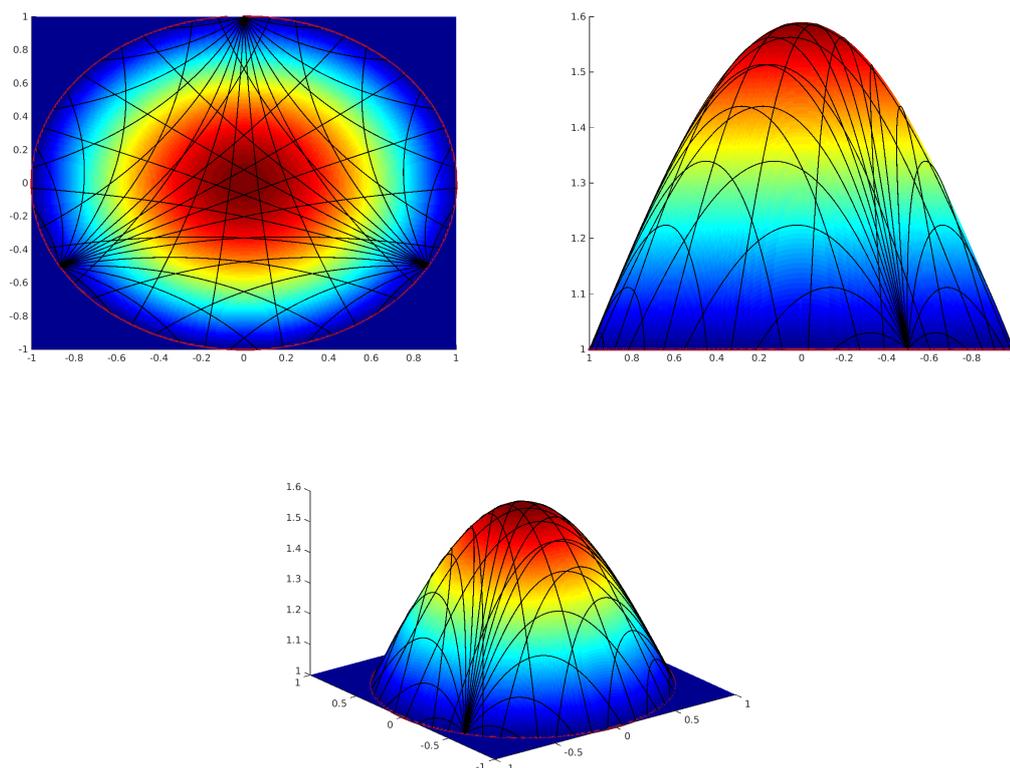


Abbildung 3.4.: Schallgeschwindigkeit c des Beispiels “Positive Krümmung” mit approximierenden Geodäten $\bar{\Gamma}_{N_x, N_\xi}^\kappa$ für $N_x = 3$, $N_\xi = 17$ und $\kappa := (0.9, 0.025, 10^{-6})$.

1. Negative Krümmung

Wir betrachten die negative Krümmung

$$\varphi_-(x) = \frac{(R^2 + a^2\|x\|^2)^2 - (R^2 + a^2)^2}{4R^2}$$

mit Parametern $a = 0.4$ und $R = 2$.

2. Positive Krümmung

Es ist die konstant positive Krümmung

$$\varphi_+(x) = \frac{(R^2 - a^2\|x\|^2)^2 - (R^2 - a^2)^2}{4R^2}$$

mit Parametern $a = 1.2$ und $R = 2$ gegeben.

Abbildung 3.3 zeigt die Schallgeschwindigkeit mit negativer und Abbildung 3.4 die Schallgeschwindigkeit der positiven Krümmung und die zugehörigen approximierenden Geodäten aus verschiedenen Blickwinkeln.

Bei der positiven Krümmung fällt dabei auf, dass im Zentrum ein “cold spot” entsteht. Alle Geodäten, bis auf die, welche als Geraden direkt durch das Zentrum laufen, werden von diesem weggekrümmt. Dies hat zur Folge, dass die Schallgeschwindigkeit am Rand von M viel stärker in die Laufzeiten eingeht, als im Zentrum von M . Dies wird sich später bei der Rekonstruktion der Schallgeschwindigkeit in Kapitel 4 als Nachteil erweisen. Man könnte das Problem eventuell beheben, indem mehr Geodäten in Richtung des Ursprungs gesandt werden, also die Richtungen nicht äquidistant gewählt werden. Dafür muss allerdings im Vorfeld bekannt sein, wo “cold spots” auftreten. Das Auftreten der “Cold Spots” ist umso größer, je stärker die Krümmung ausfällt.

Im Gegensatz dazu werden bei der negativen Krümmung die Geodäten zum Zentrum hin gebeugt, so dass die Schallgeschwindigkeit am Rand relativ gesehen weniger Einfluss auf die Laufzeiten hat.

j	$(\xi_1^j)_1$	$(\xi_1^j)_2$	$T_-(\gamma_{x_1, \xi_1^j})$	$T_+(\gamma_{x_1, \xi_1^j})$
1	-1.000000	0.000000	0.000000	0.000000
2	-0.982973	0.183750	0.368183	0.363734
3	-0.932472	0.361242	0.876552	0.481308
4	-0.850217	0.526432	1.331947	0.520296
5	-0.739009	0.673696	1.636298	0.631906
6	-0.602635	0.798017	1.845714	0.775191
7	-0.445738	0.895163	1.985453	0.952636
8	-0.273663	0.961826	2.071123	1.176581
9	-0.092268	0.995734	2.111895	1.426461
10	0.092268	0.995734	2.111895	1.426461
11	0.273663	0.961826	2.071123	1.176581
12	0.445738	0.895163	1.985453	0.952636
13	0.602635	0.798017	1.845714	0.775191
14	0.739009	0.673696	1.636298	0.631906
15	0.850217	0.526432	1.331947	0.520296
16	0.932472	0.361242	0.876552	0.481308
17	0.982973	0.183750	0.368183	0.363734

Tabelle 3.2.: Die approximierten Laufzeiten vom Punkt $x_1 = (0, 1)^T$ in die Richtungen ξ_1^j , $j = 1, \dots, 17$ für die Beispiele “Negative und positive Krümmung”.

In Tabelle 3.2 sind die Laufzeiten vom Punkt x_1 aufgetragen und zwar sowohl für die negative, als auch für die positive Krümmung. Dabei fällt zum einen auf, dass bei positiver Krümmung die Laufzeiten geringer sind als bei der negativen, zum anderen ist die Tabelle symmetrisch um $j = 9$ und $j = 10$. Beide Phänomene sind zu erwarten, da zum einen die positive Krümmung zu einer höheren Schallgeschwindigkeit führt, also zu kürzeren Laufzeiten. Zum anderen werden die Signale in äquidistante Richtungen gesendet, so dass an $\text{span}\{x_1\}$ gespiegelte Richtungen die gleiche Laufzeit haben sollten. Das ist hier der Fall, denn $\xi_1^j + \xi_1^{19-j} \in \text{span}\{x_1\}$ für $j = 2, \dots, 17$.

In Abbildung 3.5 sind die Sinogramme für die Datensätze $\bar{\Gamma}_{100,100,\pm}^\kappa$ geplottet. Dabei be-

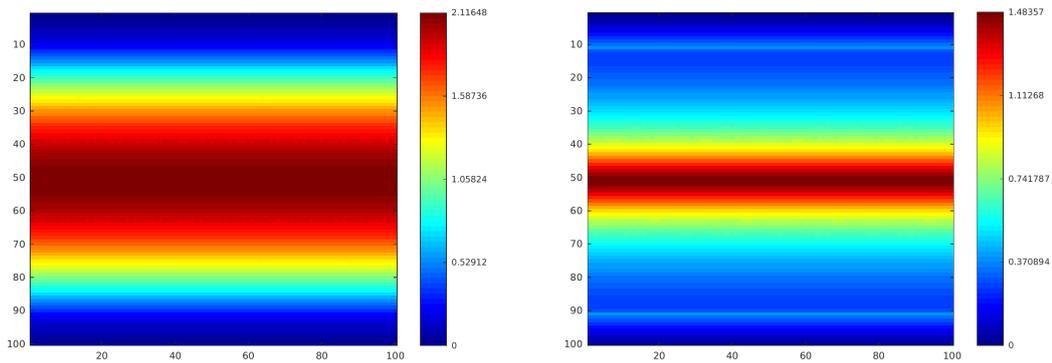


Abbildung 3.5.: Sinogramme “Negative und positive Krümmung” für die Parameter $\kappa := (0.9, 0.025, 10^{-6})$, $N_x = 100$ und $N_\xi = 100$. Auf der x-Achse sind die Aufpunkte $x_i \in X_{N_x}$ und auf der y-Achse die Richtungen $\xi_i^j \in \Xi_i^{N_\xi}$ abgetragen. Der Wert im Plot ist gerade die Laufzeit $T_\pm(\gamma_{x_i, \xi_i^j})$ der approximierenden Geodäten für $i = 1, \dots, N_x$, $j = 1, \dots, N_\xi$

sitzt das Sinogramm für die negative Krümmung im Vergleich zur positiven Krümmung einen relativ breiten Bereich mit konstant hoher Laufzeit, wohingegen bei der positiven Krümmung der Bereich in den kurzen Laufzeiten stärker ausgeprägt ist. Der Grund dafür ist, dass beispielsweise bei der negativen Krümmung die Geodäten durch die Ablenkung der Kurven hin zum Ursprung länger in M gehalten werden. Im Gegensatz dazu werden bei der positiven Krümmung die Geodäten zum Rand hin gekrümmt, so dass sie insgesamt schneller den Rand erreichen.

Die Rechenzeit für den Datensatz mit negativer Krümmung $\bar{\Gamma}_{3,17,-}^\kappa$ betrug etwa drei Sekunden und für den Datensatz mit positiver Krümmung $\bar{\Gamma}_{3,17}^\kappa$ etwa zweieinhalb Sekunden. Zur Berechnung der Sinogramme $\bar{\Gamma}_{100,100,\pm}^\kappa$ wurden etwa etwa zweieinhalb beziehungsweise zwei Minuten benötigt.

Beispiel 3.1.10 (Unstetige Schallgeschwindigkeit)

Wir betrachten die unstetige Funktion

$$c(x) = \begin{cases} 1.5 & \text{falls } \|x\| < 1 \text{ und } x_2 > 0 \\ 1 & \text{sonst} \end{cases}.$$

Abbildung 3.6 zeigt klar, dass das Verfahren nicht für unstetige Metriken funktioniert. Der Sprung wird vollständig ignoriert. Der Grund dafür ist, dass die Ableitungen immer konstant sind und somit die gleichen Geodäten erzeugt werden, wie bei der Euklidischen Metrik. Die Unstetigkeit widerspricht auch der Annahme von C^∞ -Schallgeschwindigkeiten.

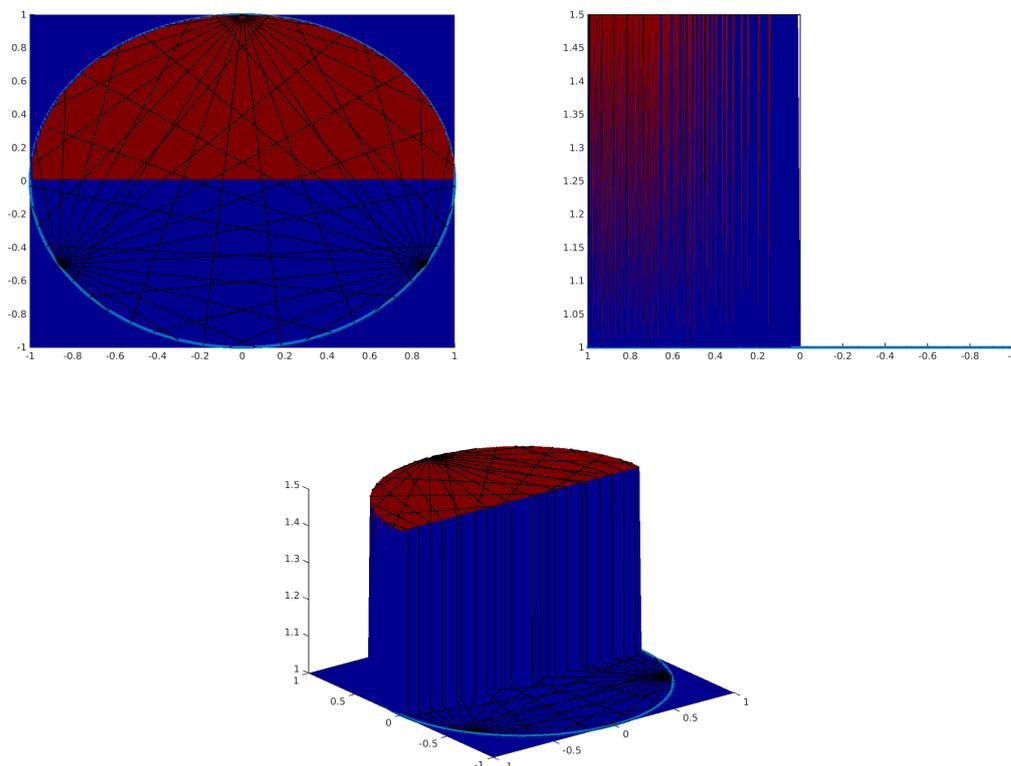


Abbildung 3.6.: Schallgeschwindigkeit c des Beispiels “Unstetig” mit approximierenden Geodäten $\bar{\Gamma}_{N_x, N_\xi}^\kappa$ für $N_x = 3$, $N_\xi = 17$ und $\kappa := (0.9, 0.025, 10^{-6})$.

Beispiel 3.1.11 (Welle mit Peaks)

Wir betrachten hier eine ähnliche Schallgeschwindigkeit wie in Beispiel 3.1.8, jedoch erzeugen wir um die Peaks noch eine halbe Sinuswelle. Dies soll ein grundsätzliches Problem der Modellierung beheben.

Wir sind nämlich in der Modellierung von einer glatten Schallgeschwindigkeit ausgegangen. In der Praxis ist das aber oftmals nicht der Fall. Beispielsweise wird man bei der medizinischen Anwendung von einer unstetigen Schallgeschwindigkeit ausgehen müssen, zum Beispiel beim Übergang von Fett- zu Knochengewebe. Das vorherige Beispiel 3.1.10 hat gezeigt, dass der Algorithmus nicht für unstetige Funktionen funktioniert. Daher soll in diesem Beispiel eine Knochenwand, die zum Beispiel ein Schädel sein könnte, geglättet werden. Idealerweise durchdringen die Geodäten den Knochen und werden von den inneren Peaks beeinflusst.

Die Schallgeschwindigkeit ist also gegeben über

$$c(x) = 1 + \sum_{i=1}^3 \varphi_i(x) \chi_i(x) + \bar{\varphi}(x) \bar{\chi}(x)$$

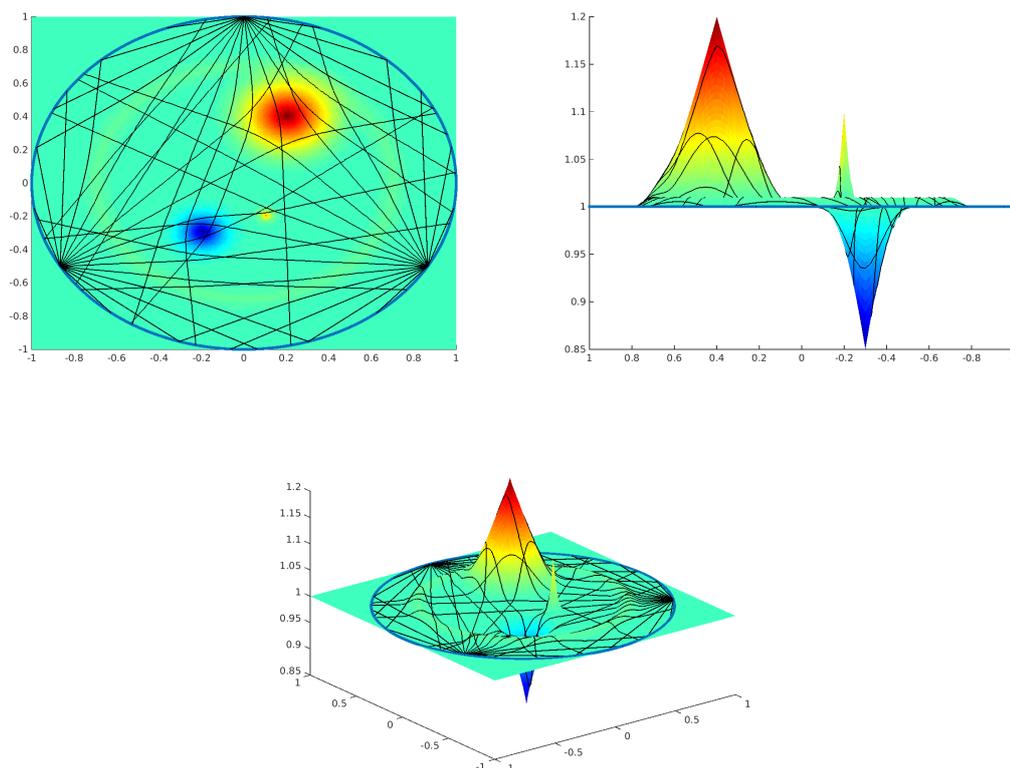


Abbildung 3.7.: Schallgeschwindigkeit c des Beispiels “Welle mit Peaks” mit approximierenden Geodäten $\bar{\Gamma}_{N_x, N_\xi}^\kappa$ für $N_x = 3$, $N_\xi = 17$ und $\kappa := (0.9, 0.025, 10^{-6})$.

mit $\varphi_i(x)$ und $\chi_i(x)$ wie in Beispiel 3.1.8. Weiter sei

$$\bar{\varphi}(x) = \bar{\theta} \cos\left(\pi \frac{\|x - \bar{p}\|}{\bar{r}}\right)$$

und

$$\bar{\chi}(x) = \begin{cases} 1 & \text{falls } 1 - 4\bar{r} \leq \|x - \bar{p}\| \leq 1 - 2\bar{r} \\ 0 & \text{sonst} \end{cases}$$

für gegebene Zentren $p_1, p_2, p_3, \bar{p} \in M$, Radien $r_1, r_2, r_3 > 0$, $\bar{r} \in (0, 0.5)$ und Amplituden $\theta_1, \theta_2, \theta_3, \bar{\theta} \in (-1, \infty)$.

Für die Beispielrechnung wählen wir als Parameter die Zentren

$$p_1 = \begin{pmatrix} \frac{1}{3} \\ \frac{3}{5} \end{pmatrix}, p_2 = \begin{pmatrix} -\frac{1}{3} \\ -\frac{3}{10} \end{pmatrix}, p_3 = \begin{pmatrix} \frac{1}{10} \\ -\frac{1}{5} \end{pmatrix}, \bar{p} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

die Radien

$$r_1 = \frac{1}{4}, r_2 = \frac{1}{5}, r_3 = \frac{1}{6}, \bar{r} = \frac{1}{10}$$

und die Amplituden

$$\theta_1 = \frac{1}{5}, \theta_2 = -\frac{3}{20}, \theta_3 = \frac{1}{10}, \bar{\theta} = \frac{1}{200}.$$

Abbildung 3.7 zeigt die Schallgeschwindigkeit und die approximierenden Geodäten aus

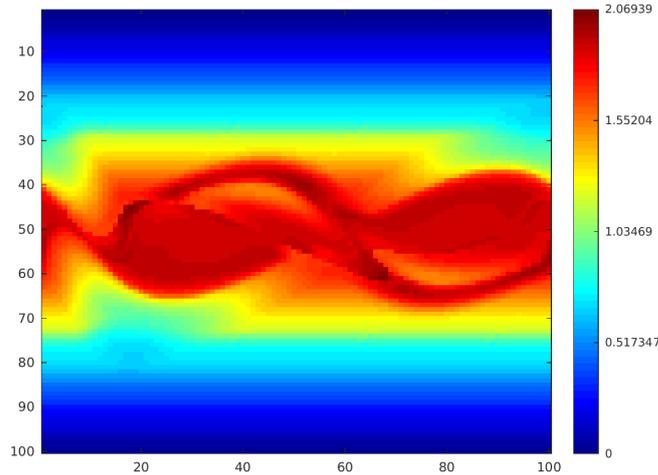


Abbildung 3.8.: Sinogramm “Welle mit Peaks” für die Parameter $\kappa := (0.9, 0.025, 10^{-6})$, $N_x = 100$ und $N_\xi = 100$. Auf der x-Achse sind die Aufpunkte $x_i \in X_{N_x}$ und auf der y-Achse die Richtungen $\xi_i^j \in \Xi_i^{N_\xi}$ abgetragen. Der Wert im Plot ist gerade die Laufzeit $T(\gamma_{x_i, \xi_i^j})$ der approximierenden Geodäten für $i = 1, \dots, N_x, j = 1, \dots, N_\xi$.

verschiedenen Blickwinkeln. Der grün-gelbliche Kreis, welcher die Peaks umgibt, stellt dabei den Knochen dar. Die Amplitude θ der umgebenden Welle ist gerade noch so groß, dass relativ viele Geodäten in das innere Gebiet eindringen können. Wählt man die Amplitude größer, so wird der größte Teil der Geodäten “reflektiert”. Das spiegelt sehr gut das Verhalten der Ultraschalltomographie in der Praxis wider, denn beispielsweise Materialien wie Knochen können von Ultraschallsignalen zwar durchdrungen werden, werden aber zum größten Teil reflektiert, so dass keine brauchbaren Messergebnisse hinter diesen erzeugt werden können.

Abbildung 3.8 zeigt das Sinogramm zu den Parametern $\kappa := (0.9, 0.025, 10^{-6})$, $N_x = 100$ und $N_\xi = 100$. Die Rechenzeit für den Datensatz $\bar{\Gamma}_{3,17}^\kappa$ betrug etwa zwanzig Sekunden und für den Datensatz $\bar{\Gamma}_{100,100}^\kappa$ knapp 14 Minuten.

Die Beispiele haben gezeigt, dass zumindest stetige Schallgeschwindigkeiten gut durch das Lösen des Anfangswertproblems 3.2 detektiert werden. In speziellen Fällen kann es dabei zu “cold spots” kommen, also Gebieten, die nur verhältnismäßig wenig von Geodäten durchlaufen werden, da sie vor dem Erreichen weggekrümmt werden. Außerdem

haben wir gesehen, dass unstetige Schallgeschwindigkeiten nicht erfasst werden. Durch eine Glättung der Sprungstellen erhält man zwar wieder Ergebnisse, diese sind aber nur brauchbar, sofern der Sprung nicht zu groß ist. Insgesamt kann man also sagen, dass dieses Verfahren gut für glatte Metriken funktioniert, allerdings unter der Voraussetzung, dass die Krümmungen nicht allzu stark sind, andernfalls treten “cold spots” auf.

Die Berechnung der Geodäten ist insgesamt sehr effektiv und kann sehr gut parallel ablaufen, da jede Geodäte unabhängig von den anderen durch Lösen des Anfangswertproblems 3.2 bestimmt werden kann. Hier muss aber erwähnt werden, dass die Berechnungsdauer für die einzelnen Geodäten sehr stark schwanken kann. Die Berechnung einer sehr kurzen oder einer wenig gebeugten Geodäte ist sehr effektiv. Im Gegensatz dazu nimmt die Berechnung einer langen und stark gekrümmten Geodäte vergleichsweise viel Rechenzeit in Anspruch.

Demzufolge ist die zunächst erhoffte Laufzeitbeschleunigung durch die Einbeziehung einer Grafikkarte in die Berechnungen nur von mäßigem Erfolg gekrönt worden. Die Idee war, in mehreren hundert Prozessen die Berechnung zu parallelisieren. Da GPUs aber nur wenige (beziehungsweise eine) Steuereinheiten besitzen, kann erst ein neuer Satz Geodäten berechnet werden, wenn alle alten Berechnungen abgeschlossen worden sind. Dadurch entstehen Wartezeiten und es hat sich herausgestellt, dass eine Rechnung auf schnellen CPUs derzeit sinnvoller ist.

3.2. Berechnung der Geodäten als Kürzeste

Wir wollen nun geodätische Kurven als kürzeste Verbindungen zwischen zwei Punkten interpretieren und daraus ein Verfahren herleiten um diese zu berechnen. Die Grundidee dazu stammt von Frank Schöpfer. Das Verfahren, welches angewendet werden soll, stellt allerdings weitere Anforderungen an die Metrik, da es nur für eine bestimmte Klasse von Kurven funktioniert. Allerdings können wir für die folgenden Betrachtungen auf die Einfachheit der Mannigfaltigkeit verzichten.

Um zu betonen, dass wir uns auf diese Klasse an Geodäten einschränken, werden wir teilweise neue Begriffe (wie zum Beispiel gleichförmig, approximierende Kürzeste, Interpolierende) definieren, welche so allerdings nicht in der allgemeinen Literatur bekannt sind.

Definition 3.2.1 (Stückweise reguläre Kurven)

Eine Kurve $\gamma : I := [0, \tau] \rightarrow M$ heißt *stückweise regulär*, falls γ stetig und nur an endlich vielen Stellen $\{t_i\}_{i=1}^{N+1}$ nicht regulär ist, genauer falls $\gamma|_{(t_i, t_{i+1})} \in C^\infty$ ist und die Ableitungen von γ auf $[t_i, t_{i+1}]$ stetig fortsetzbar sind für alle $i = 1, \dots, N$, wobei wir $t_1 = 0$ und $t_{N+1} = \tau$ setzen. Man nennt die $\{t_i\}_{i=1}^{N+1}$ auch *Ecken von γ* .

Wir möchten nun den Begriff der gleichförmigen Kurven einführen. Diese können vergleichsweise einfach approximiert werden, Sie zeichnen sich dadurch aus, dass ihr Geschwindigkeitsvektor immer Anteile in die Vorwärtsrichtung besitzt. Dies führt unter anderem dazu, dass sie keine Schleifen und Selbstschnitte bilden (vergleiche Abbildung 3.9).

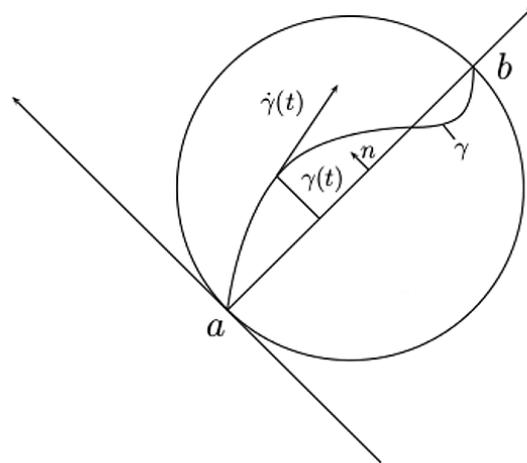


Abbildung 3.9.: Schematische Darstellung einer gleichförmigen Kurve.

Definition 3.2.2 (Gleichförmige Kurven)

Sei $M \subset \mathbb{R}^2$ kompakt und einfach zusammenhängend und $a, b \in \partial M$. Dann heißt eine stückweise reguläre Kurve $\gamma : [0, \tau] \rightarrow M$ von a nach b (d.h. $\gamma(0) = a$, $\gamma(\tau) = b$) *gleichförmig* (bezüglich a und b), falls

$$\langle \dot{\gamma}(t), b - a \rangle > 0$$

für alle $t \in (t_j, t_{j+1})$, $j = 1, \dots, N$ und auf den Ecken $(t_i)_{i=1}^{N+1}$ von γ

$$\left\langle \lim_{t \nearrow t_j} \dot{\gamma}(t), b - a \right\rangle > 0$$

sowie

$$\left\langle \lim_{t \searrow t_j} \dot{\gamma}(t), b - a \right\rangle > 0$$

gilt. Dabei bedeutet $t \nearrow t_j$, dass sich t von links nähert, also $t \mapsto t_j$ und $t < t_j$, beziehungsweise gilt für $t \searrow t_j$, dass sich t von rechts nähert, also $t \mapsto t_j$ und $t > t_j$.

Definition 3.2.3 (Gleichförmige Mannigfaltigkeit)

Wir definieren die Klasse der stückweisen regulären und gleichförmigen Kurven von a nach b durch

$$K := K(a, b) := \{ \gamma \in C(I, M) : \gamma \text{ ist eine stückweise reguläre und gleichförmige Kurve bzgl. } a \text{ und } b \}.$$

Eine Mannigfaltigkeit (M, g) nennen wir *gleichförmig*, sofern zwischen zwei beliebigen Punkten $a, b \in M$ Kürzeste existieren und diese gleichförmig sind.

Durch das folgende Lemma erhält man nun eine praktische Darstellungsmöglichkeit für gleichförmige Kurven.

Lemma 3.2.4 (Charakterisierung gleichförmiger Kurven)

Seien $M \subset \mathbb{R}^2$ kompakt und konvex, $a, b \in \partial M$ und $\gamma : [0, \tau] \rightarrow M$ eine stückweise reguläre Kurve mit Ecken $\{t_i\}_{i=1}^{N+1}$. Dann gilt:

γ ist genau dann gleichförmig bezüglich a und b , wenn ein stückweiser Diffeomorphismus $\phi : [0, 1] \rightarrow [0, \tau]$ mit $\phi(0) = 0$ und $\phi(1) = \tau$, welcher orientierungserhaltend ist, das heißt $\dot{\phi}(t) > 0$ für alle $t \in [0, 1] \setminus \{\phi^{-1}(t_i)\}_{i=1}^{N+1}$, und eine stückweise reguläre Abbildung $s : [0, \tau] \rightarrow \mathbb{R}$ mit $s(0) = 0 = s(1)$ existieren, so dass

$$\gamma(\phi(t)) = a + t(b - a) + s(t)\xi$$

für alle $t \in [0, \tau]$ gilt. Hierbei bezeichnet ξ eine festgelegte Einheitsnormale auf $b - a$, also $\xi \perp (b - a)$ und $\|\xi\| = 1$. Der stückweise Diffeomorphismus ϕ und die stückweise reguläre Abbildung s sind höchstens an den Ecken $\{\phi^{-1}(t_i)\}_{i=1}^{N+1}$ nicht differenzierbar.

Beweis. Sei zunächst γ eine gleichförmige und reguläre Kurve. Wir definieren die Abbildung

$$\phi^{-1} : [0, \tau] \rightarrow [0, 1]; t \mapsto \frac{\langle \gamma(t) - a, b - a \rangle}{\|b - a\|^2}.$$

Dann ist $\phi^{-1} \in C^\infty([0, \tau], [0, 1])$ und es gilt $\phi^{-1}(0) = 0$, $\phi^{-1}(\tau) = 1$ sowie

$$\dot{\phi}^{-1}(t) = \frac{\langle \dot{\gamma}(t), b - a \rangle}{\|b - a\|^2} > 0,$$

da γ als gleichförmig vorausgesetzt wurde. Insbesondere folgt daraus, dass die Umkehrabbildung $\phi : [0, 1] \rightarrow [0, \tau]$ existiert und es gilt $\phi(0) = 0$, sowie $\phi(1) = \tau$. Da $\dot{\phi}^{-1} > 0$ auf $[0, \tau]$ ist, folgt mit dem Satz von der Umkehrabbildung, dass ϕ eine $C^\infty([0, 1], [0, \tau])$ -Abbildung ist. Insbesondere ist ϕ somit ein Diffeomorphismus und $\gamma \circ \phi$ eine orientierungserhaltende Umparametrisierung von γ . Wir definieren nun

$$s : [0, 1] \rightarrow \mathbb{R}, t \mapsto \langle \gamma(\phi(t)) - a, \xi \rangle,$$

wobei ξ eine eindeutig festgelegte Normale auf $b - a$ sei. Dann gilt

$$s(0) = \langle \gamma(\phi(0)) - a, \xi \rangle = \langle 0, \xi \rangle = 0$$

sowie

$$s(1) = \langle \gamma(\phi(1)) - a, \xi \rangle = \langle b - a, \xi \rangle = 0.$$

s ist offenbar eine C^∞ -Funktion auf $[0, 1]$. Sei $t \in [0, 1]$, dann gilt

$$t = \phi^{-1}(\phi(t)) = \frac{\langle \gamma(\phi(t)) - a, b - a \rangle}{\|b - a\|^2},$$

also

$$t\|b - a\| = \left\langle \gamma(\phi(t)) - a, \frac{b - a}{\|b - a\|} \right\rangle.$$

Da $\left\{ \frac{b-a}{\|b-a\|}, \xi \right\}$ eine Orthonormalbasis des \mathbb{R}^2 ist, gilt mit

$$s(t) = \langle \gamma(\phi(t)) - a, \xi \rangle$$

die Darstellung

$$\begin{aligned} \gamma(\phi(t)) - a &= \left\langle \gamma(\phi(t)) - a, \frac{b-a}{\|b-a\|} \right\rangle \frac{b-a}{\|b-a\|} + \langle \gamma(\phi(t)) - a, \xi \rangle \xi \\ &= t \|b-a\| \frac{b-a}{\|b-a\|} + s(t) \xi, \end{aligned}$$

also

$$\gamma(\phi(t)) = a + t(b-a) + s(t)\xi.$$

Sei nun γ eine stückweise reguläre und gleichförmige Kurve, also an endlich vielen Stellen $\{t_i\}_{i=1}^{N+1}$ nicht differenzierbar aber stetig. Dann ist ϕ^{-1} , wie oben definiert, stetig und $\phi^{-1}|_{(t_i, t_{i+1})}$ ist stetig differenzierbar für $i = 1, \dots, N$. Darüber hinaus gilt $\dot{\phi}^{-1}|_{(t_i, t_{i+1})} > 0$ für $i = 0, \dots, n$. Nach dem Satz über die Umkehrfunktion existieren nun die Umkehrabbildungen

$$\left(\phi^{-1}|_{(t_i, t_{i+1})} \right)^{-1} = \phi|_{(\phi^{-1}(t_i), \phi^{-1}(t_{i+1}))}$$

auf den einzelnen Teilintervallen, welche stetig ineinander übergehen müssen. Somit ist ϕ ein stückweiser Diffeomorphismus mit den Ecken $\{\phi^{-1}(t_i)\}_{i=1}^{N+1}$. Für den Rest des Beweises geht man analog vor, wobei s jedoch im Allgemeinen nur stückweise regulär ist.

Für die Rückrichtung existiere nun ein stückweise regulärer und orientierungserhaltender Diffeomorphismus $\phi : [0, 1] \rightarrow [0, \tau]$ sowie eine stückweise reguläre Abbildung $s : [0, 1] \rightarrow \mathbb{R}$, so dass γ die Darstellung

$$\gamma(\phi(t)) = a + t(b-a) + s(t)\xi$$

für alle $t \in [0, 1]$ besitzt. Sei $t \in \{r \in [0, \tau] : \gamma \text{ ist in } r \text{ differenzierbar}\}$. Für ϕ existiert eine lokale, differenzierbare Umkehrabbildung ϕ^{-1} um r und es gilt

$$\dot{\gamma}(t) = \dot{\gamma}(\phi(\phi^{-1}(t))) = \dot{\phi}^{-1}(t)(b-a) + \dot{\phi}^{-1}(t)\dot{s}(\phi^{-1}(t))\xi.$$

Hieraus folgt

$$\langle \dot{\gamma}, b-a \rangle = \dot{\phi}^{-1}(t)\|b-a\|^2 + \dot{\phi}^{-1}(t)\dot{s}(\phi^{-1}(t))\langle \xi, b-a \rangle > 0,$$

da ϕ orientierungserhaltend ist und $\xi \perp (b-a)$ gilt. □

Bemerkung 3.2.5

Aus dem obigen Lemma folgt insbesondere, dass bei gleichförmigen Kurven eine Umparametrisierung existiert, so dass man die Kurve als Funktion bzgl. der Achsen $\text{span}\{b-a\}$ und $\text{span}\{\xi\}$ auffassen kann.

Bei nicht zu komplizierten Metriken entspricht das auch der Anschauung. Man lässt also unter anderem keine Schleifen, Reflexionen und Selbstschnitte zu. Die Geodäte darf nicht "zurück laufen".

3.2.1. Ein Verfahren für das Vorwärtsproblem

Sofern die Metrik es zulässt, dass die Geodäten zwischen zwei Punkten der Riemannschen Mannigfaltigkeit gleichförmig sind, bietet sich das folgende Verfahren an (vergleiche Abbildung 3.10). Hierfür seien eine gleichförmige, konvexe Mannigfaltigkeit (M, g) und Punkte $a, b \in M$ gegeben. Berechnet werden soll eine gleichförmige Kürzeste γ , welche die beiden Punkte a und b verbindet.

Gemäß Lemma 3.2.4 existiert nun für γ (nach Umparametrisierung) die Darstellung

$$\gamma(t) = a + t(b - a) + s(t)\xi,$$

mit $t \in [0, 1]$ und ξ sei eine eindeutig festgelegte Normale auf $b - a$. Nun unterteilen wir die Gerade \overline{ab} gleichmäßig in $N \in \mathbb{N}$ Teilstücke und interpolieren γ linear. Das heißt, mit

$$\begin{aligned} \gamma_k &:= a + t_k(b - a) + s_k\xi, \\ t_k &:= \frac{k - 1}{N} \end{aligned}$$

für $k = 1, \dots, N + 1$ und den sinnvollen Festlegungen $s_1 = s_{N+1} = 0$, $s_k := s(t_k)$ für $k = 2, \dots, N$ und $\gamma_k \in M$ ist der Weg zwischen γ_k und γ_{k+1} beschrieben durch

$$L(\gamma_k, \gamma_{k+1}, s) : [0, 1] \rightarrow M$$

mit

$$\begin{aligned} L(\gamma_k, \gamma_{k+1}, s)(t) &:= (1 - t)\gamma_k + t\gamma_{k+1} \\ &= \gamma_k + t(\gamma_{k+1} - \gamma_k) \\ &= a + t_k(b - a) + s_k\xi + t \left(\frac{1}{N}(b - a) + (s_{k+1} - s_k)\xi \right) \\ &= a + (k - 1)\omega + s_k\xi + t(\omega + (s_{k+1} - s_k)\xi), \end{aligned}$$

wobei $s = (s_1, \dots, s_{N+1})^\tau = (0, s_2, \dots, s_N, 0)^\tau \in \mathbb{R}^{N+1}$ und $\omega := \frac{b-a}{N}$. Die *approximierende Kürzeste* γ_s von γ ist dann gegeben durch

$$\gamma_s(a, b) : [0, N] \rightarrow M; t \mapsto \begin{cases} L(\gamma_1, \gamma_2, s)(t), & \text{für } t \in [0, 1] \\ \vdots & \vdots \\ L(\gamma_k, \gamma_{k+1}, s)(t - k + 1), & \text{für } t \in [k - 1, k] \\ \vdots & \vdots \\ L(\gamma_N, \gamma_{N+1}, s)(t - N + 1), & \text{für } t \in [N - 1, N] \end{cases}$$

Lemma 3.2.6

Es sei (M, g) eine konvexe Mannigfaltigkeit. Seien $a, b \in M$ und $\gamma \in K(a, b)$. Dann gilt für die approximative Kürzeste γ_s , wie oben konstruiert (insbesondere gelte $\gamma_k \in M$ für alle $k = 1, \dots, N$ und $s_1 = s_{N+1} = 0$):

$$\gamma_s(a, b) \in K(a, b).$$

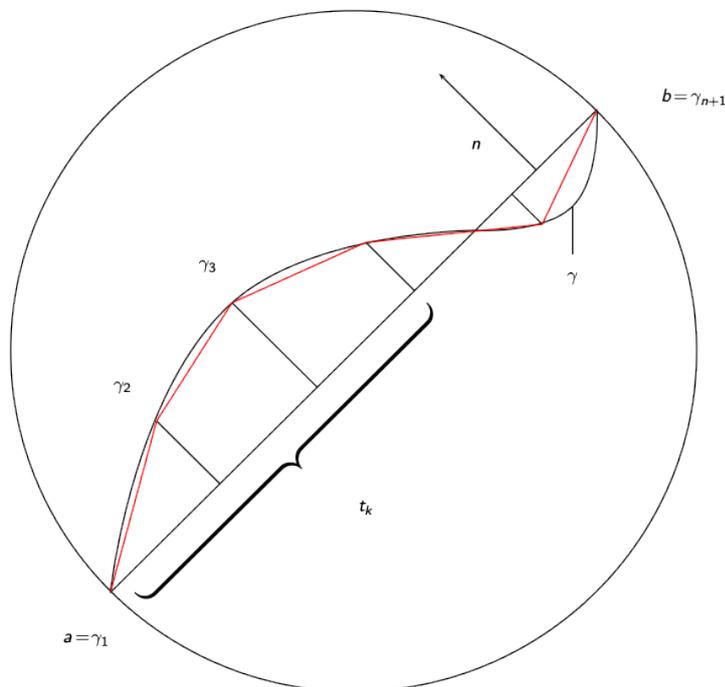


Abbildung 3.10.: Schematische Darstellung einer approximierenden Kürzesten. Die schwarze Kurve sei die Geodäte γ von a nach b . Die rote Linie sei die Approximation γ_s an γ . n ist die Normale auf $b - a$.

Beweis. γ_s ist nach Konstruktion stückweise linear und stetig, also eine stückweise C^∞ -Kurve und liegt aufgrund der Bedingungen an s vollständig in M . Wegen $s_1 = s_{N+1} = 0$ gilt $\gamma_s(0) = a$ und $\gamma_s(N) = b$. Es bleibt zu zeigen, dass γ_s gleichförmig ist. Sei $t \in [0, N]$, dann existiert ein $k \in \{1, \dots, N\}$, so dass $t \in (k-1, k)$. Ist $\gamma_s(t)$ eine Ecke, so kann man die Ableitung darauf links- und rechtsseitig fortsetzen, da γ_s stückweise linear ist, und die folgende Argumentation bleibt gültig. Dann gilt

$$\begin{aligned}
 \langle \dot{\gamma}_s(t), b - a \rangle &= \langle \dot{L}(\gamma_k, \gamma_{k+1}, s)(t - k + 1), b - a \rangle \\
 &= \langle \omega + (s_{k+1} - s_k)\xi, b - a \rangle \\
 &= \frac{1}{N} \|b - a\|^2 + (s_{k+1} - s_k) \langle \xi, b - a \rangle \\
 &= \frac{1}{N} \|b - a\|^2 > 0.
 \end{aligned}$$

□

Lemma 3.2.7 (Laufzeit der approximierenden Kürzesten)

Es seien $a, b \in \partial M$ und γ_s die approximierende Kürzeste. Dann gilt für deren Laufzeit $T(a, b, \gamma_s)$ von a nach b

$$T(a, b, \gamma_s) = \sum_{k=1}^N \|\gamma_{k+1} - \gamma_k\|_2 \int_0^1 \tilde{n}(L(\gamma_k, \gamma_{k+1}, s)(t)) dt.$$

Beweis. Es gilt für $k = 1, \dots, N$

$$\begin{aligned} \omega + (s_{k+1} - s_k)\xi &= \frac{1}{N}(b - a) + (s_{k+1} - s_k)\xi \\ &= \frac{k}{N}(b - a) - \frac{k-1}{N}(b - a) + (s_{k+1} - s_k)\xi \\ &= a + t_{k+1}(b - a) + s_{k+1}\xi - a - t_k(b - a) - s_k\xi \\ &= \gamma_{k+1} - \gamma_k. \end{aligned}$$

Es folgt damit

$$\begin{aligned} T(a, b, \gamma_s) &= \int_{\gamma_s} \tilde{n}(x) \sqrt{dx^i \otimes dx^j(x, x)} ds_x \\ &= \sum_{k=1}^N \int_{L(\gamma_k, \gamma_{k+1}, s)} \tilde{n}(x) \sqrt{dx^i \otimes dx^j(x, x)} ds_x \\ &= \sum_{k=1}^N \int_0^1 \tilde{n}(L(\gamma_k, \gamma_{k+1}, s)(t)) \|\dot{L}(\gamma_k, \gamma_{k+1}, s)(t)\|_2 dt \\ &= \sum_{k=1}^N \|\omega + (s_{k+1} - s_k)\xi\|_2 \int_0^1 \tilde{n}(L(\gamma_k, \gamma_{k+1}, s)(t)) dt \\ &= \sum_{k=1}^N \|\gamma_{k+1} - \gamma_k\|_2 \int_0^1 \tilde{n}(L(\gamma_k, \gamma_{k+1}, s)(t)) dt \end{aligned}$$

□

Das folgende Lemma besagt nun, dass eine Minimierung über die Laufzeit aller (approximierenden) Kurven aus $K(a, b)$ bereits mit der Laufzeit einer Kürzesten übereinstimmt.

Lemma 3.2.8 (Kürzeste sind glatt)

Seien (M, g) eine gleichförmige, konvexe Mannigfaltigkeit und $\gamma \in C^\infty(I, M)$ eine Kürzeste zwischen den Punkten a und b in M . Dann gilt

$$d(a, b) = \inf_{\sigma \in C^\infty(I, M) \cap K(a, b)} T(a, b, \sigma) = \inf_{\sigma \in K(a, b)} T(a, b, \sigma).$$

Beweis. Dass die linke Seite größer ist als die rechte ist offensichtlich. Die andere Ungleichung folgt aus Do Carmo [1976, Kapitel 4, Proposition 2], denn die Proposition besagt, dass $\gamma = \inf_{\sigma \in K(a,b)} T(a, b, \sigma)$ für eine Geodäte $\gamma \in C^\infty(I, M)$ angenommen wird, also insbesondere $\gamma \in C^\infty(I, M) \cap K(a, b)$ gilt. \square

Es stellt sich nun noch die Frage, ob die Minimierung über $s \in \mathbb{R}^{N+1}$ eine gute Approximation an die Geodäte liefert.

Definition 3.2.9 (Approximierende und Interpolierende)

Es seien (M, g) eine gleichförmige, konvexe Mannigfaltigkeit und $a, b \in M$. Für $m \in \mathbb{N}$ ist

$$\Delta_m(a, b) := \{p_m \in K(a, b) : \exists 0 = t_1 < t_2 < \dots < t_{m+1} = \tau : \\ p_m|_{[t_k, t_{k+1}]} \text{ ist linear } \forall k = 1, \dots, m\}$$

die Menge der *linearen, gleichförmigen Approximierenden von a nach b der Ordnung m*. Ist $\gamma \in K(a, b)$, so heißt $p_m \in \Delta_m(a, b)$ mit der Zerlegung $t_1 < \dots < t_{m+1}$ *lineare, gleichförmige Interpolierende an γ vom Grad m*, falls

$$p_m(t_i) = \gamma(t_i)$$

für alle $i = 1, \dots, m+1$ gilt. Es gilt mit der obigen Notation für $t \in [t_k, t_{k+1}]$

$$p_m(t) = L(\gamma_{k_m}, \gamma_{k_m+1}, s_m)(m(t - t_{k_m})),$$

wobei $s_m := (p_m(t_i))_{i=1}^{m+1} \in \mathbb{R}^{m+1}$ sei.

Lemma 3.2.10 (Approximationseigenschaft der Interpolierenden)

Es sei (M, g) eine gleichförmig konvexe Mannigfaltigkeit, $a, b \in M$ und $\gamma \in K(a, b)$. $p_m \in \Delta_m(a, b)$ sei eine lineare, gleichförmige Interpolierende von γ . Dann gilt

$$\lim_{m \rightarrow \infty} |T(a, b, p_m) - T(a, b, \gamma)| = 0$$

Beweis. Der Beweis erfolgt unter Ausnutzung des Satzes über die majorisierte Konvergenz. Hierfür definieren wir die Abbildungen

$$f : [0, \tau] \rightarrow \mathbb{R} \quad , \quad t \mapsto \tilde{n}(\gamma(t)) \|\dot{\gamma}(t)\|_2 \\ f_m : [0, \tau] \rightarrow \mathbb{R} \quad , \quad t \mapsto \tilde{n}(p_m(t)) \|\dot{p}_m(t)\|_2.$$

Dann gilt $\lim_{m \rightarrow \infty} f_m(t) = f(t)$ für alle $t \in [0, \tau]$, da \tilde{n} und $\|\cdot\|_2$ stetig sind und punktweise $\lim_{m \rightarrow \infty} \dot{p}_m(t) = \dot{\gamma}(t)$ für $t \in [0, \tau]$ gilt. Dies sieht man folgendermaßen ein: Für festes $m \in \mathbb{N}$ und $t \in [t_{k_m}, t_{k_m+1}]$ gilt (gemäß der Notation von Definition 3.2.9)

$$\begin{aligned} \dot{p}_m(t) &= m \dot{L}(\gamma_{k_m}, \gamma_{k_m+1}, s_m)(m(t - t_{k_m})) \\ &= m (\gamma_{k_m+1} - \gamma_{k_m}) \\ &= \frac{\gamma(t_{k_m} + \frac{1}{m}) - \gamma(t_{k_m})}{\frac{1}{m}}. \end{aligned}$$

Für $m \rightarrow \infty$ gilt sowohl $t_{k_m} \rightarrow t$, als auch $\dot{p}_m(t) \rightarrow \dot{\gamma}(t)$. Des Weiteren ist $f_m(t)$ beschränkt, da M kompakt, \tilde{n} stetig und \dot{p}_m gegen $\dot{\gamma}$ konvergiert, also insbesondere beschränkt ist. Hieraus folgt dann

$$\begin{aligned} \lim_{m \rightarrow \infty} \int_0^\tau \tilde{n}(p_m(t)) \|\dot{\gamma}(t)\|_2 dt &= \lim_{m \rightarrow \infty} \int_0^\tau f_m(t) dt \\ &= \int_0^\tau f(t) dt \\ &= \int_0^\tau \tilde{n}(\gamma(t)) \|\dot{\gamma}(t)\|_2 dt \end{aligned}$$

□

Satz 3.2.11 (Konvergenzsatz für Interpolierende)

Es sei (M, g) eine gleichförmig konvexe Mannigfaltigkeit, $a, b \in M$ und $m \in \mathbb{N}$. Definiere $\beta_m := \inf_{p_m \in \Delta_m} T(a, b, p_m)$. Dann gilt

$$\beta_m \xrightarrow{m \rightarrow \infty} d(a, b).$$

Das heißt, die Interpolierenden konvergieren gegen eine Kürzeste.

Beweis. Sei $\varepsilon > 0$. Dann existiert ein $\gamma \in K(a, b)$, so dass

$$d(a, b) \leq T(a, b, \gamma) < d(a, b) + \varepsilon.$$

Zu $m \in \mathbb{N}$ sei $p_m : [0, \tau] \rightarrow M$ eine lineare, gleichförmige Interpolierende zu γ . Gemäß Lemma 3.2.10 gilt

$$\int_0^\tau \tilde{n}(p_m(t)) \|\dot{p}_m(t)\|_2 dt \xrightarrow{m \rightarrow \infty} \int_0^\tau \tilde{n}(\gamma(t)) \|\dot{\gamma}(t)\|_2 dt.$$

Also existiert ein $m_0 \in \mathbb{N}$, so dass

$$\left| \int_0^\tau \tilde{n}(p_m(t)) \|\dot{p}_m(t)\|_2 dt - \int_0^\tau \tilde{n}(\gamma(t)) \|\dot{\gamma}(t)\|_2 dt \right| < \varepsilon$$

für alle $m \geq m_0$ gilt. Mit der obigen Definition für β_m und Lemma 3.2.8 folgt nun

$$\begin{aligned} d(a, b) &\leq \beta_m = \inf_{p_m \in \Delta_m} T(a, b, p_m) \\ &\leq \beta_{m_0} := \int_0^\tau \tilde{n}(p_{m_0}(t)) \|\dot{p}_{m_0}(t)\|_2 dt \\ &\leq \int_0^\tau \tilde{n}(\gamma(t)) \|\dot{\gamma}(t)\|_2 dt + \varepsilon \\ &< d(a, b) + 2\varepsilon. \end{aligned}$$

Da $\varepsilon > 0$ beliebig war, folgt die Behauptung. □

Bemerkung 3.2.12

Satz 3.2.11 sagt aus, dass in der Menge der linearen Approximationen eine Kürzeste angenähert werden kann im Sinne der Distanz. Für den Beweis war es nicht notwendig, dass die Riemannsche Mannigfaltigkeit einfach ist.

3.2.2. Numerische Umsetzung

Effektiv ergibt sich aus dem vorangehenden Abschnitt unmittelbar ein numerisches Verfahren, um die Geodäten zu approximieren. Denn gemäß Lemma 3.2.6 und Satz 3.2.11 lässt sich das Vorwärtsproblem nun in ein $(m - 1)$ -dimensionales Optimierungsproblem mit Nebenbedingungen umformulieren.

$$\begin{aligned}
\min_{p_m \in \Delta_m(a,b)} T(a, b, p_m) &= \min_{p_m \in \Delta_m(a,b)} \sum_{k=1}^m \|p_m(t_{k+1}) - p_m(t_k)\|_2 \times \\
&\quad \int_0^1 \tilde{n}(L(p_m(t_k), p_m(t_{k+1}), s)(t)) dt \\
&= \min_{s \in \mathbb{R}^{m+1}} \sum_{k=1}^m \|p_{m,k+1} - p_{m,k}\|_2 \times \\
&\quad \int_0^1 \tilde{n}(L(p_{m,k}, p_{m,k+1}, s)(t)) dt \tag{3.3}
\end{aligned}$$

unter den Nebenbedingungen

$$\begin{aligned}
p_{m,1} &:= p_m(t_1) = a, \\
p_{m,m+1} &:= p_m(t_{m+1}) = b, \text{ und} \\
p_{m,k} &:= a + t_k(b - a) + s_k \xi = p_m(t_k) \in M
\end{aligned}$$

für alle $k = 2, \dots, m$.

Das Argument der Zielfunktion ist dann eine Approximierende an eine Kürzeste von a nach b . Da p_m für festes $m \in \mathbb{N}$ durch $s \in \mathbb{R}^{m+1}$ bestimmt ist, schreiben wir auch $p_m = p_m(s)$. Insbesondere ist also über $s \in \mathbb{R}^{m+1}$ zu minimieren, wobei Anfangs- und Endpunkt bereits festgelegt sind. Für $N_a, N_b \in \mathbb{N}$ seien

$$A_{N_a} := \left\{ a_i = \begin{pmatrix} \sin(2\pi \frac{i-1}{N_a}) \\ \cos(2\pi \frac{i-1}{N_a}) \end{pmatrix} : i = 1, \dots, N_a \right\}$$

die Menge der Aufpunkte und

$$B_i^{N_b} := \left\{ b_i^j = \begin{pmatrix} \sin \left(2\pi \left(\frac{i-1}{N_a} + \frac{j-1}{N_b} \right) \right) \\ \cos \left(2\pi \left(\frac{i-1}{N_a} + \frac{j-1}{N_b} \right) \right) \end{pmatrix} : j = 1, \dots, N_b \right\}$$

die Menge der Zielpunkte für $i = 1, \dots, N_a$. Dann ist ein Datensatz $\tilde{\Gamma}_{N_a, N_b}^m$ zum Diskretisierungsparameter $m \in \mathbb{N}$ gegeben durch

$$\begin{aligned}
\tilde{\Gamma}_{N_a, N_b}^m &:= \left\{ p_m(a_i, b_i^j) \in \Delta_m(a_i, b_i^j) : p_m \text{ ist Minimierer von (3.3)}, \right. \\
&\quad \left. a_i \in A_{N_a}, b_i^j \in B_i^{N_b}, i = 1, \dots, N_a, j = 1, \dots, N_b \right\}.
\end{aligned}$$

In den folgenden Beispielen wurden für drei Aufpunkte, also $N_a = 3$, $a_1 = (0, 1)^T$,

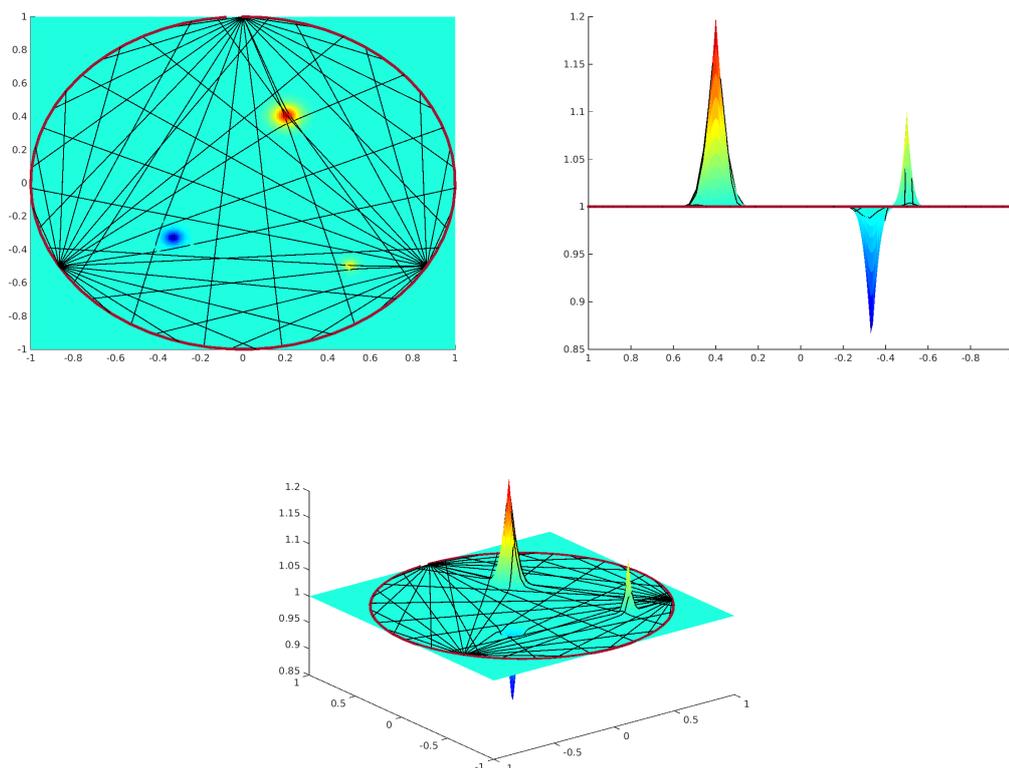


Abbildung 3.11.: Schallgeschwindigkeit c des Beispiels “Peaks” mit approximierenden Geodäten $\tilde{\Gamma}_{N_a, N_b}^m$ für $N_a = 3$, $N_b = 17$ und $m = 50$.

$a_2 = \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)^T$ und $a_3 = \left(-\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)^T$ jeweils $N_b = 17$ äquidistant auf dem Rand ∂M verteilte Punkte $B_i^{N_b}$, $i = 1, 2, 3$ konstruiert und die approximierenden Geodäten $\tilde{\Gamma}_{N_a, N_b}^m$ und deren Laufzeiten $T(a_i, b_i^j)$, $i = 1, 2, 3$, $j = 1, \dots, 17$ bezüglich dieser Punkte berechnet. In dieser Rechnung wurde $m = 50$ gewählt, um eine hinreichend gute Abdeckung des Gebiets zu erhalten. Wird m zu klein gewählt, so werden örtlich kleine Variationen in der Schallgeschwindigkeit nicht mehr durch die Geodäten detektiert. Ein sehr feine Diskretisierung der Geodäten führt allerdings zu einer erheblichen Rechenzeiterhöhung, da m gerade die Dimension des Optimierungsproblems angibt.

Als Minimierer wurde die MATLAB-Funktion `fminunc` genutzt. Dieser verwendet das Quasi-Newton-Verfahren, wobei die Ableitungen über das Differenzenverfahren approximiert werden. Für das Update der Hesse-Matrix wurde BFGS verwendet (siehe unter anderem Shanno [1970]). Die Optimierung wurde beendet, sobald die Differenz zweier aufeinander folgender Funktionswerte unter 10^{-6} oder der Abstand zweier aufeinander folgende Iterierten unter 10^{-6} fiel.

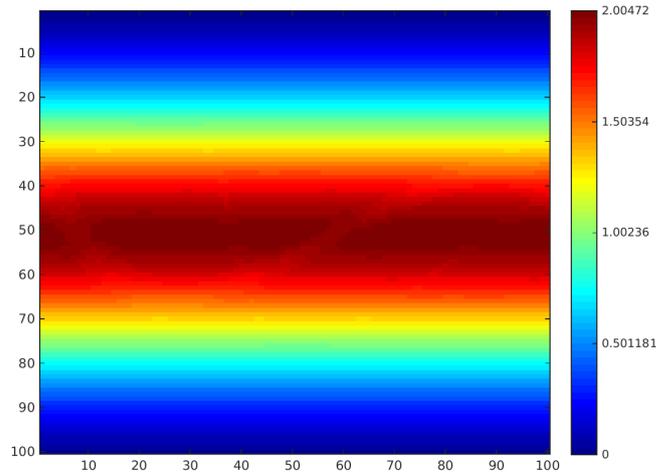


Abbildung 3.12.: Sinogramm “Peaks” für die Parameter $m = 50$, $N_a = 100$, $N_b = 100$. Auf der x-Achse sind die Aufpunkte $a_i \in A_{N_a}$ und auf der y-Achse die Randpunkte $b_i^j \in B_i^{N_b}$ abgetragen. Der Wert im Plot ist gerade die Laufzeit $T(a_i, b_i^j)$ zwischen den Punkten, $i = 1, \dots, N_a$, $j = 1, \dots, N_b$.

Beispiel 3.2.13 (Peaks)

Wir betrachten die gleiche Schallgeschwindigkeit wie in Beispiel 3.1.8, also

$$c(x) = 1 + \sum_{i=1}^3 \varphi_i(x) \chi_i(x)$$

mit den dort angegebenen Funktionen und Parametern.

Abbildung 3.11 zeigt die Schallgeschwindigkeit und die approximierenden Geodäten aus verschiedenen Blickwinkeln. Im Gegensatz zur ersten Methode sind hier die Zielpunkte auf dem Rand fest vorgegeben, so dass eine gleichmäßigere Ausleuchtung des Gebietes vorliegt. Es fällt auf, dass ein Durchlaufen der Peaks erzwungen wird, wobei bei der ersten Methode die Geodäten stark gekrümmt wurden. Das Verhalten kann man sehr gut am Peak in $(0.2, 0.4)^T$ erkennen.

Abbildung 3.12 zeigt das Sinogramm zu den Parametern $m = 50$, $N_a = 100$, $N_b = 100$. Im Vergleich zum Sinogramm der ersten Methode (Abbildung 3.1), welches auf der gleichen Anzahl an berechneten Geodäten basiert, liefert das Minimierungsverfahren mit festen Randpunkten ein gleichmäßigeres Sinogramm. Die Randbereiche oben und unten (jeweils ca. 20 Richtungen bzw. Zielpunkte) stimmen bei beiden Verfahren überein, was auch zu erwarten ist, da dort die Geodäten Geradenstücke sind und somit die Laufzeit bei beiden Verfahren identisch sein muss. Im zentralen Bereich liefert das Lösen des Anfangswertproblems 3.2 allerdings stärkere Laufzeitschwankungen. Wie bereits erwähnt, ist dies dadurch zu erklären, dass bei der ersteren Methode die Geodäten durch die Peaks abgelenkt werden und somit erhebliche Abweichungen von Geraden entstehen können. Das ist bei dem

j	$(b_1^j)_1$	$(b_1^j)_2$	$T_-(a_1, b_1^j)$	$T_+(a_1, b_1^j)$
1	0.000000	1.000000	0.000000	0.000000
2	0.361242	0.932472	0.368132	0.362058
3	0.673696	0.739009	0.727353	0.683659
4	0.895163	0.445738	1.068225	0.943351
5	0.995734	0.092268	1.380363	1.140576
6	0.961826	-0.273663	1.652256	1.284419
7	0.798017	-0.602635	1.871601	1.384439
8	0.526432	-0.850217	2.026360	1.447620
9	0.183750	-0.982973	2.106594	1.478220
10	-0.183750	-0.982973	2.106594	1.478220
11	-0.526432	-0.850217	2.026360	1.447620
12	-0.798017	-0.602635	1.871601	1.384439
13	-0.961826	-0.273663	1.652256	1.284419
14	-0.995734	0.092268	1.380363	1.140576
15	-0.895163	0.445738	1.068225	0.943351
16	-0.673696	0.739009	0.727353	0.683659
17	-0.361242	0.932472	0.368132	0.362058

Tabelle 3.3.: Die approximierten Laufzeiten vom Punkt $a_1 = (0, 1)^T$ zu den Punkten b_1^j , $j = 1, \dots, 17$ für das Beispiel “Negative und positive Krümmung”.

Minimierungsansatz nicht möglich, da zum einen der Zielpunkt erreicht werden muss und zum anderen die Approximierenden gleichförmig sind. Somit sind zurück laufende Geodäten nicht möglich und durch die Minimierung tendenziell eine Kurve zu erwarten, die nicht allzu stark von einer Geraden abweicht.

Die Rechenzeit für den Datensatz $\tilde{\Gamma}_{3,17}^{50}$ betrug etwa vier Minuten und für den Datensatz $\tilde{\Gamma}_{100,100}^{50}$ etwa neuneinhalb Stunden. Dies ist erheblich mehr als bei der ersten Methode, nämlich um den Faktor 96 beziehungsweise 285 größer. Selbst bei einer Vergrößerung der Diskretisierung der Geodäten, also einer Verkleinerung von m , ist das Minimierungsverfahren erheblich langsamer als das Lösen des Anfangswertproblems, also der ersten Methode.

Beispiel 3.2.14 (Konstante Krümmung)

Wir betrachten das Beispiel 3.1.9 mit konstanter Krümmung, also

$$c(x) = 1 + \varphi_{\pm}(x)\chi_M(x)$$

mit den dort angegebenen Funktionen $\varphi_{\pm} : M \rightarrow \mathbb{R}$.

Die Laufzeiten $T_{\pm}(a_1, b_1^j)$ von a_1 nach b_1^j für $j = 1, \dots, 17$ sind im Vergleich in Tabelle 3.3 dargestellt. Dabei kann man die gleichen Phänomene wie in Beispiel 3.1.9 beobachten, jedoch in etwas weniger starker Ausprägung. Bei der negativen Krümmung sind die Laufzeiten der zweiten Methode immer geringer als bei der ersten. Bei der positiven Krümmung verhält es sich genau umgekehrt.

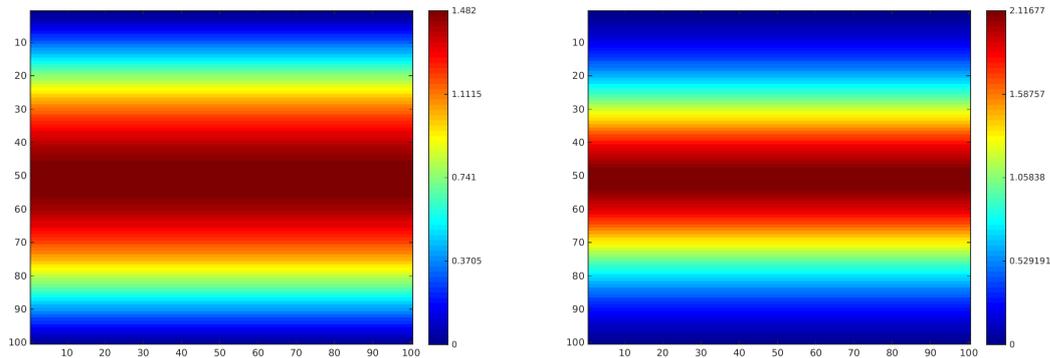


Abbildung 3.13.: Sinogramm “Negative und positive Krümmung” für die Parameter $m = 50$, $N_a = 100$, $N_b = 100$. Auf der x-Achse sind die Aufpunkte $a_i \in A_{N_a}$ und auf der y-Achse die Randpunkte $b_i^j \in B_i^{N_b}$ abgetragen. Der Wert im Plot ist gerade die Laufzeit $T_{\pm}(a_i, b_i^j)$ zwischen den Punkten a_i und b_i^j für $i = 1, \dots, N_a$, $j = 1, \dots, N_b$

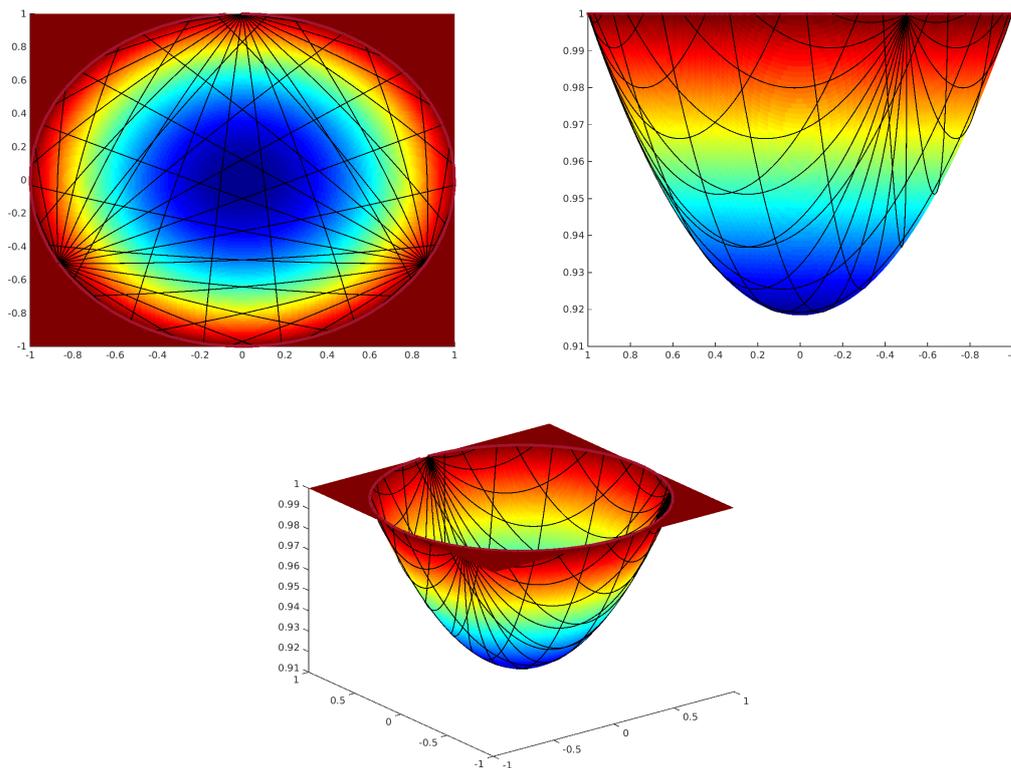


Abbildung 3.14.: Schallgeschwindigkeit c des Beispiels “Negative Krümmung” mit approximierenden Geodäten $\tilde{\Gamma}_{N_a, N_b}^m$ für $N_a = 3$, $N_b = 17$ und $m = 50$.

Das Verhalten kann man auch sehr schön an den Sinogrammen für einen größeren Datensatz sehen (siehe Abbildung 3.13). Im Vergleich zur ersten Methode (Abbildung 3.5) sind die Sinogramme hier deutlich dichter beieinander.

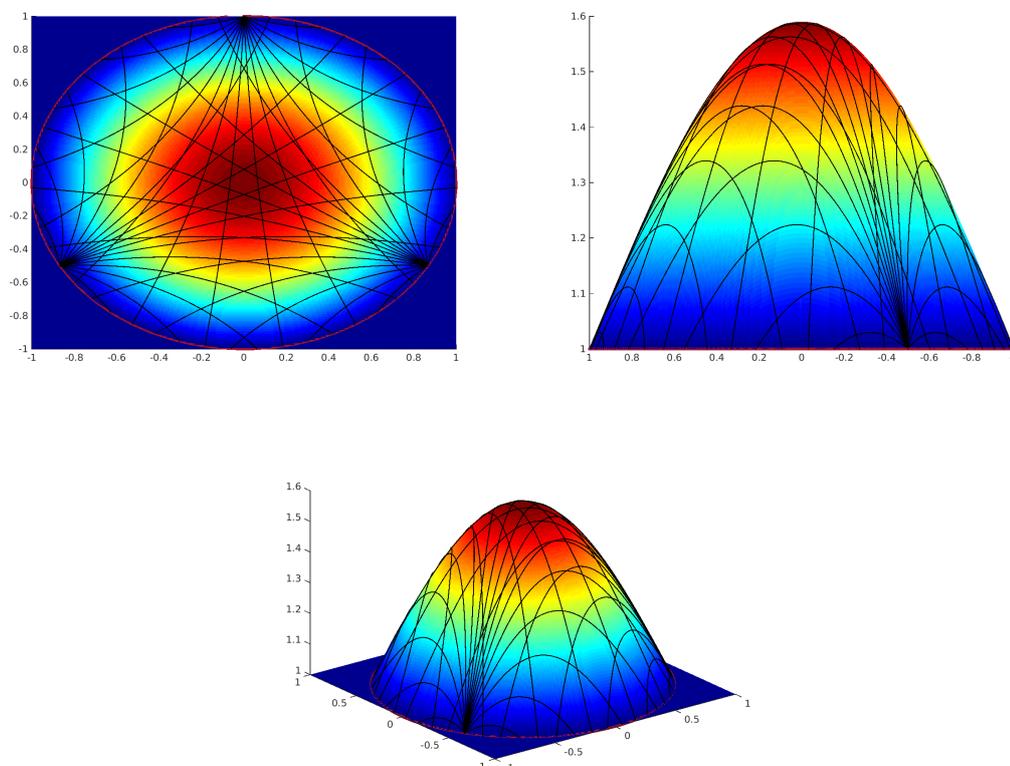


Abbildung 3.15.: Schallgeschwindigkeit c des Beispiels “Positive Krümmung” mit approximierenden Geodäten $\tilde{\Gamma}_{N_a, N_b}^m$ für $N_a = 3$, $N_b = 17$ und $m = 50$.

Den Grund dafür sieht man, wenn man sich den tatsächlichen Verlauf der berechneten Geodäten anschaut (Abbildung 3.14 für die negative und Abbildung 3.15 für die positive Krümmung). Das ganze Gebiet wird hier in beiden Fällen sehr gut ausgeleuchtet, die Geodäten sind relativ nah an Geradenstücken. Somit ergeben sich nur geringe Laufzeitverlängerungen oder -verkürzungen durch die Krümmung der berechneten Wege. Allerdings muss man annehmen, dass evtl. Wege erzwungen werden, die so in der Praxis nicht auftreten würde, wie zum Beispiel das direkte durchlaufen von Senken oder Höhen. Dafür treten bei diesem Verfahren keine “cold spots” auf.

Abbildung 3.14 zeigt die Schallgeschwindigkeit und die approximierenden Geodäten aus verschiedenen Blickwinkeln und Abbildung 3.13 das Sinogramm zu den Parametern $m = 50$, $N_a = 100$, $N_b = 100$.

Die Rechenzeit für den Datensatz $\tilde{\Gamma}_{3,17,-}^{50}$ betrug etwa sechs Minuten und für den Datensatz $\tilde{\Gamma}_{3,17,+}^{50}$ etwa fünfeinhalb Minuten. Die Berechnung der Sinogramme $\tilde{\Gamma}_{100,100,\pm}^{50}$ hat jeweils etwa zwölf Stunden veranschlagt, was im Vergleich zur ersten Methode eine längere

Rechenzeit um den Faktor 120 bis 132 beziehungsweise 288 bis 360 bedeutet.

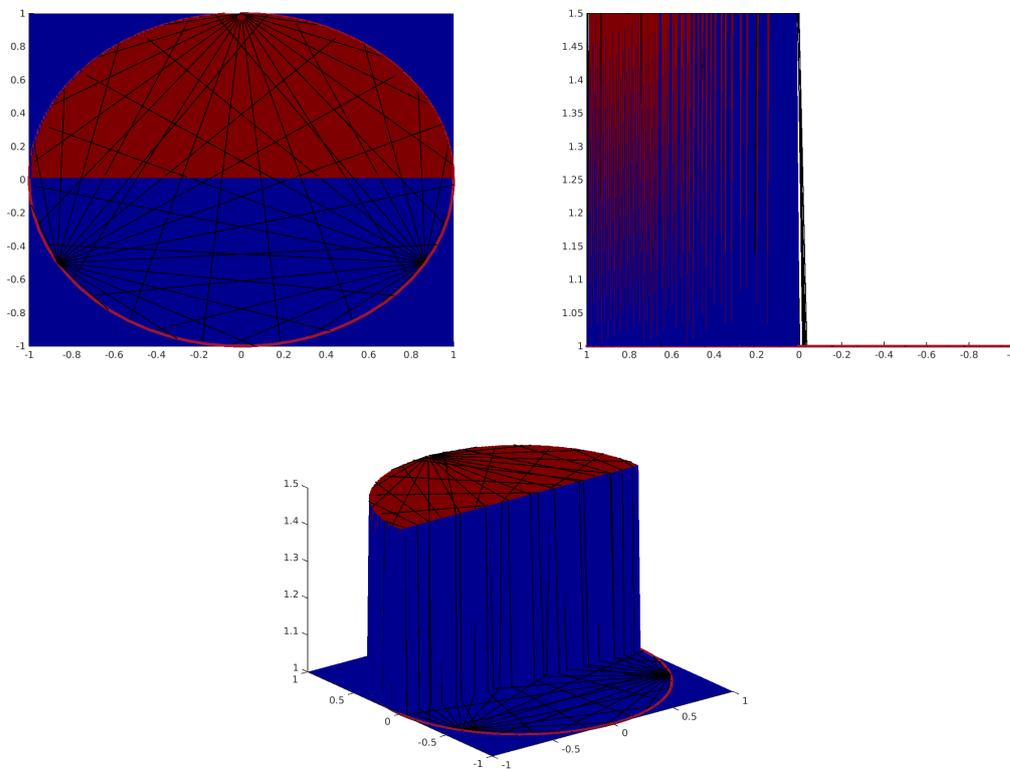


Abbildung 3.16.: Schallgeschwindigkeit c des Beispiels “Unstetig” mit approximierenden Geodäten $\tilde{\Gamma}_{N_a, N_b}^m$ für $N_a = 3$, $N_b = 17$ und $m = 50$.

Beispiel 3.2.15 (Unstetige Schallgeschwindigkeit)

Für die unstetige Funktion

$$c(x) = \begin{cases} 1.5 & \text{falls } \|x\| < 1 \text{ und } x_2 > 0 \\ 1 & \text{sonst} \end{cases}.$$

funktioniert dieses Verfahren nicht. Der Sprung wird in der Optimierung nicht berücksichtigt, die berechneten Kürzesten sind identisch mit denen bei der Euklidischen Metrik, was man in Abbildung 3.16 erkennen kann.

Das Problem liegt hier ebenfalls, wie bei der ersten Methode, an der Berechnung der Ableitungen. Diese sind immer konstant Null und demzufolge wird die Schallgeschwindigkeit als konstant angenommen. Es erfolgt somit keine Optimierung. Indem die Schallgeschwindigkeit geglättet und hinreichend fein in den Geodäten diskretisiert wird, könnte das Problem jedoch entschärft werden.

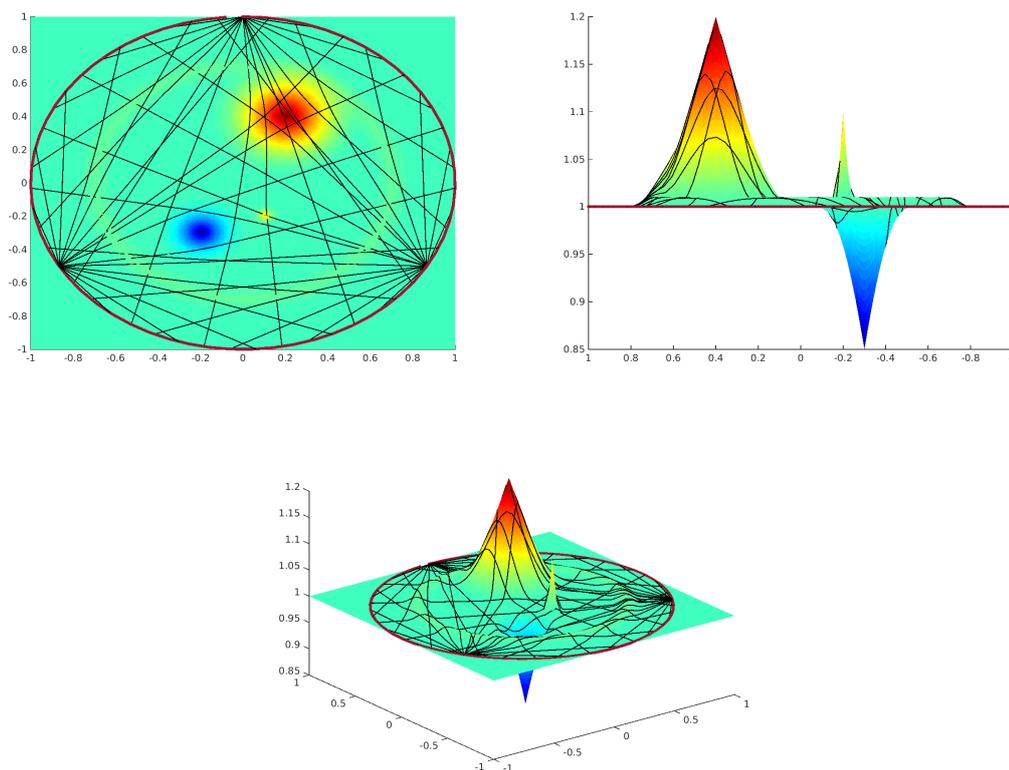


Abbildung 3.17.: Schallgeschwindigkeit c des Beispiels “Welle mit Peaks” mit approximierenden Geodäten $\tilde{\Gamma}_{N_a, N_b}^m$ für $N_a = 3$, $N_b = 17$ und $m = 50$.

Beispiel 3.2.16 (Welle mit Peaks)

In Anlehnung an Beispiel 3.1.11 sei die Schallgeschwindigkeit c gegeben durch

$$c(x) = 1 + \sum_{i=1}^3 \varphi_i(x) \chi_i(x) + \bar{\varphi}(x) \bar{\chi}(x)$$

mit den entsprechenden, dort angegebenen Funktionen und Parametern. In Abbildung 3.17 ist die Schallgeschwindigkeit mit den berechneten Geodäten abgebildet. Es treten hier ähnliche Effekte auf, wie im Beispiel “Peaks”. Die Geodäten werden im Vergleich zur ersten Methode (vergleiche Abbildung 3.7) insgesamt weniger stark gekrümmt und die Peaks direkter durchlaufen. Dies führt zu dem Sinogramm 3.18, welches für $N_a = 100$, $N_b = 100$ aufgestellt wurde. Wie zu erwarten ist dies im Vergleich zur ersten Methode deutlich homogener (vergleiche Abbildung 3.8).

Die Rechenzeit für den Datensatz $\tilde{\Gamma}_{3,17}^{50}$ betrug etwa sechs Minuten und für den Datensatz $\tilde{\Gamma}_{100,100}^{50}$ etwa 18 Stunden, was abermals deutlich länger ist als bei der ersten Methode.

Bemerkung 3.2.17

Abschließend lässt sich sagen, dass beide Methoden zur Berechnung der geodätischen Kurven und deren Laufzeiten gut funktionieren. Insbesondere reicht für die numerische

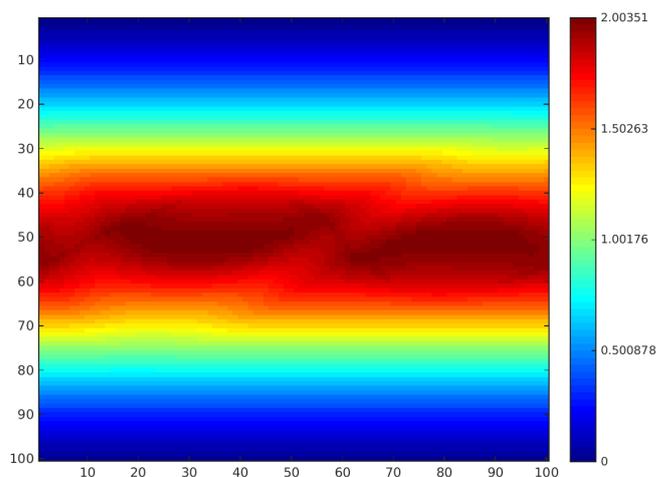


Abbildung 3.18.: Sinogramm “Welle mit Peaks” für die Parameter $m = 50$, $N_a = 100$, $N_b = 100$. Auf der x-Achse sind die Aufpunkte $a_i \in A_{N_a}$ und auf der y-Achse die Randpunkte $b_i^j \in B_i^{N_b}$ abgetragen. Der Wert im Plot ist gerade die Laufzeit $T(a_i, b_i^j)$ zwischen den Punkten, $i = 1, \dots, N_a$, $j = 1, \dots, N_b$

Berechnung eine stetige Schallgeschwindigkeit aus, bei unstetigen Schallgeschwindigkeiten versagen jedoch beide Algorithmen.

Die erste Methode, also das Lösen des Anfangswertproblems, ist deutlich schneller (etwa um den Faktor 100) als die zweite, welche auf dem Lösen eines Minimierungsproblems basiert. Auf der anderen Seite erkaufte man sich die Geschwindigkeit durch das Auftreten von “cold spots”, also Bereichen, die gar nicht oder nur geringfügig von Geodäten durchlaufen werden. Dies wird sich auch in den Rekonstruktionen im nächsten Kapitel bemerkbar machen.

Man kann vermuten, dass die zweite Methode zu besseren Ergebnissen führt, sofern ungünstige Gegebenheiten vorliegen, wie beispielsweise beim Beispiel 3.1.11, wo das interessante Gebiet von einem stark brechenden beziehungsweise beugenden Bereich umgeben ist.

Allerdings wird die zweite Methode zum Lösen des Inversen Problems nicht weiter verfolgt. Durch die obigen Tests hat sich ergeben, dass eine mit Methode 1 vergleichbare Rekonstruktion derzeit voraussichtlich eine Rechenzeit von zwei Monaten benötigen würde. Weitere Verbesserungen im Algorithmus oder der Hardware bezüglich der Rechenzeit sind also wünschenswert, so dass auch, basierend auf der zweiten Methode, nichttriviale numerische Experimente gemacht werden können.

Eine Portierung der Algorithmen auf eine Grafikkarte würde sich im Vergleich zur ersten Methode wahrscheinlich eher lohnen. Da alle Geodäten im gleichen Maße diskretisiert sind (im Gegensatz zur ersten Methode, bei der eine Schrittweitensteuerung eingesetzt wird und somit die Schrittweite massiv variieren kann), sollte die Berechnung der einzelnen Geodäten etwa gleich lange dauern.

Kapitel 4

Regularisierung und Rekonstruktion

In diesem Kapitel beschäftigen wir uns mit einer Lösung des inversen Problems 2.2.12. Es seien also Daten $u^{\text{mess}} \in L^2(\partial\Omega M, \mathbb{R}^+)$ von Laufzeitmessungen am Rand von M gemäß Definition 2.2.10 gegeben. Ziel ist es den zugrunde liegenden Brechungsindex $\tilde{n} : M \rightarrow \mathbb{R}_+$ aus diesen Daten zu rekonstruieren, wobei der Zusammenhang

$$R(\tilde{n}) = u^{\text{mess}}$$

auf $\partial\Omega M$ gelte und $R : L^2(M) \cap C^\infty(M) \rightarrow Y := L^2(\partial\Omega M)$ der Vorwärtsoperator (siehe Definition 2.2.11), definiert durch

$$R(\tilde{n})(x, \xi) := \int_{\gamma_{\tilde{n}}(x, \xi)} \tilde{n}(z) dl, \quad \forall (x, \xi) \in \partial\Omega M,$$

sei.

Zur Lösung des Problems wählen wir die Tikhonov-Regularisierung. Einen guten Überblick dazu erhält man in den Monographien von Louis [1989], Rieder [2003] und Schuster u. a. [2012]. Dazu definieren wir uns das folgende Funktional.

Definition 4.0.18 (Tikhonov-Funktional)

Das Funktional $J_\alpha : X := L^2(M) \cap C^\infty(M) \rightarrow \mathbb{R}$, gegeben durch

$$J_\alpha(\tilde{n}) = \frac{1}{2} \|R(\tilde{n}) - u^{\text{mess}}\|_Y^2 + \alpha \Sigma(\tilde{n})$$

mit einem gegebenen Funktional $\Sigma : X \rightarrow \mathbb{R}_{>0}$ und $\alpha > 0$, heißt *Tikhonov-Funktional*. Den Ausdruck

$$\frac{1}{2} \|R(\tilde{n}) - u^{\text{mess}}\|_Y^2$$

bezeichnen wir als *Datenterm* und das Funktional Σ als *Straf- oder Regularisierungsterm*. Der Parameter α heißt *Regularisierungsparameter*.

Ziel ist es das Tikhonov-Funktional zu minimieren. Den Strafterm legen wir für unsere weiteren Ausführungen fest als

$$\Sigma := \Sigma_p : L^p(M) \rightarrow \mathbb{R}_{>0}, \tilde{n} \mapsto \frac{1}{p} \|\tilde{n} - 1\|_{L^p(M)}^p$$

für ein $p \in (1, \infty)$. Das so definierte Σ bestraft also Abweichungen von der euklidischen Metrik und zwar in der p -Norm. Durch die Wahl von p wird es möglich sein gewisse Eigenschaften der Lösung zu steuern. Damit das Tikhonov-Funktional mit der obigen Festsetzung von Σ wohldefiniert ist, muss natürlich sichergestellt sein, dass sowohl Daten- als auch Strafterm sinnvoll ausgewertet werden können. Deshalb setzen wir voraus, dass es mindestens einen Punkt $a \in X$ gibt, so dass $\Sigma(a) < \infty$ ist.

Das Tikhonov-Funktional J_α soll nun minimiert werden

$$\left. \begin{aligned} J_\alpha(\tilde{n}) = \frac{1}{2} \|R(\tilde{n}) - u^{\text{mess}}\|_Y^2 + \frac{\alpha}{p} \|\tilde{n} - 1\|_{L^p(M)}^p \rightarrow \min \\ \text{unter der Nebenbedingung } \tilde{n} \in X \cap L^p(M) =: \mathcal{D}(J) \end{aligned} \right\} \quad (4.1)$$

Die Wahl des Regularisierungsparameters α bestimmt dabei, wie die Lösung aussieht. Wird α sehr klein gewählt, so muss der Datenterm im Verhältnis zum Strafterm klein sein, also die gemessenen Laufzeiten mit den durch die approximierte Metrik erzeugten übereinstimmen. Dabei kann die approximierte Metrik aber sehr stark von der Ausgangsmetrik abweichen (bei der wir davon ausgehen, dass sie nahe bei der euklidischen Metrik ist). Wählt man das α größer, so wird eine Lösung nahe der euklidischen Metrik favorisiert; stärkere Abweichungen in den Laufzeiten werden zugelassen.

Die Formulierung des Problems als Minimierungsproblem für das Tikhonov-Funktional hat einige angenehme Eigenschaften.

Satz 4.0.19 (Existenz einer Lösung)

Für alle $\alpha > 0$ und $u^{\text{mess}} \in Y$ existiert eine Lösung $\tilde{n}_\alpha \in \mathcal{D}(J)$, die das Minimierungsproblem (4.1) löst.

Beweis. Einen Beweis findet man beispielsweise bei Schuster u. a. [2012, Proposition 4.1] oder Hofmann u. a. [2007]. □

Bei nichtlinearen Problemen ist es im Allgemeinen nicht möglich globale Eindeutigkeitsaussagen machen zu können. Wir müssen uns also mit der Existenz zufrieden geben. Der Vollständigkeit halber wollen wir noch eine wichtige Stabilitätsaussage angeben.

Satz 4.0.20 (Stabilität der regularisierten Lösungen)

Zu gegebenem $\alpha > 0$ sind die Minimierer \tilde{n}_α von (4.1) stabil bezüglich der Daten u^{mess} , das heißt konvergiert eine Datenfolge $\{u_k^{\text{mess}}\}$ gegen u^{mess} , so existiert unter der zugehörigen Folge $\{\tilde{n}_k\}$, als Lösung des Minimierungsproblems

$$\frac{1}{2} \|R(\tilde{n}) - u_k^{\text{mess}}\|_Y^2 + \frac{\alpha}{p} \|\tilde{n} - 1\|_{L^p(M)}^p \rightarrow \min, \text{ unter der Nebenbedingung } \tilde{n} \in \mathcal{D}(J)$$

eine Teilfolge $\{x_{k_s}\}$, welche zumindest schwach gegen den Minimierer von (4.1) konvergiert.

Beweis. Bei [Schuster u. a., 2012, Proposition 4.2] wird ein Beweis gegeben. □

Unter anderem folgt aus dem Satz, dass wir eine Konvergenz zum Minimierer des Ausgangsproblems erwarten können, sofern die Messfehler kleiner werden.

Möchten wir nun eine Lösung des Minimierungsproblems (4.1) bestimmen, so kommen wir mit den üblichen Mitteln leider nicht weiter. Will man ein Abstiegsverfahren ansetzen, so muss die Fréchet-Ableitung des Funktionals bestimmt werden. Unter anderem müsste dann die Ableitung $R'(\tilde{n}) : X \rightarrow Y$ von R berechnet werden. Es müsste dann gelten

$$0 = \lim_{\|h\|_X \rightarrow 0} \frac{\|R(\tilde{n} + h) - R(\tilde{n}) - R'(\tilde{n}) \cdot h\|_Y}{\|h\|_X}.$$

Fassen wir R als Abbildung in zwei Variablen auf, also

$$\tilde{R} : X \times X \rightarrow Y,$$

definiert durch

$$\tilde{R}(a, \tilde{n})(x, \xi) := R_a(\tilde{n})(x, \xi) = \int_{\gamma_{x,\xi}^a} \tilde{n}(z) dl, \quad \forall (x, \xi) \in \partial\Omega M,$$

wobei $\gamma_{x,\xi}^a$ die Lösung der geodätischen Differentialgleichung 2.2 sei bzgl. a und Anfangswerten $\gamma_{x,\xi}^a(0) = x$ und $\dot{\gamma}_{x,\xi}^a(0) = \xi$, so erhält man, sofern \tilde{R} bezüglich der einzelnen Variablen Fréchet-differenzierbar ist

$$\tilde{R}'(a, \tilde{n})(h_a, h_{\tilde{n}}) = \tilde{R}_a(a, \tilde{n}) \cdot h_a + \tilde{R}_{\tilde{n}}(a, \tilde{n}) \cdot h_{\tilde{n}},$$

wobei $\tilde{R}_a(a, \tilde{n}) = D_a R(a, \tilde{n})$ und $\tilde{R}_{\tilde{n}}(a, \tilde{n}) = D_{\tilde{n}} R(a, \tilde{n})$ die Fréchet-Ableitungen nach der jeweiligen Variablen im Index seien. Hierbei ist der rechte Summand

$$\tilde{R}_{\tilde{n}}(a, \tilde{n}) \cdot h_{\tilde{n}} = D_{\tilde{n}} \tilde{R}(a, \tilde{n}) \cdot h_{\tilde{n}} = D_{\tilde{n}} \int_{\gamma_{x,\xi}^a} \tilde{n} dl \cdot h_{\tilde{n}}$$

durchaus noch analytisch bestimmbar, da hier nach dem Integranden, also ein linearer Operator, abgeleitet werden muss. Der linke Summand

$$\tilde{R}_a(a, \tilde{n}) \cdot h_a = D_a \tilde{R}(a, \tilde{n}) \cdot h_a = D_a \int_{\gamma_{x,\xi}^a} \tilde{n} dl \cdot h_a$$

ist allerdings nicht ohne Weiteres angebbbar, denn es muss erklärt werden wie sich die Integrationswege bei kleinen Änderungen des Brechungsindex ändern.

Als Ausweg führen wir eine Linearisierung des Tikhonov-Funktional ein.

Definition 4.0.21 (Linearisiertes Tikhonov-Funktional)

Es sei $a \in C^\infty(M)$ ein Brechungsindex. Das Funktional $J_\alpha^a : \mathcal{D}(J) \cap \mathcal{D}(\Sigma) \rightarrow \mathbb{R}$, gegeben durch

$$J_\alpha^a(\tilde{n}) = \frac{1}{2} \|R_a(\tilde{n}) - u^{\text{mess}}\|_Y^2 + \alpha \Sigma(\tilde{n})$$

mit einem gegebenen Funktional $\Sigma : X \rightarrow \mathbb{R}_{>0}$ und $\alpha > 0$, heißt *linearisiertes Tikhonov-Funktional*. Hierbei ist R_a der linearisierte Vorwärtsoperator aus Definition 2.2.11.

Das linearisierte Tikhonov-Funktional ist nun vergleichsweise einfach zu minimieren. Allerdings würde eine Minimierung zu einer festen Menge an Geodäten \mathcal{G} (aus Definition 3.1.7) gerade nicht den Einfluss des Brechungsindex auf dessen Laufwege berücksichtigen. Demzufolge muss noch eine Kopplung stattfinden. Diese soll durch eine adaptive Neuberechnung der Geodäten erfolgen, was wir im folgenden Algorithmus festhalten wollen.

Algorithmus 4.0.22 (Iteratives, adaptives Minimierungsverfahren)

Eingabe: Brechungsindex \tilde{n}_0 , Regularisierungsparameter $\alpha = (\alpha_k)_{k=1,2,\dots} > 0$.

- (S0) Berechne zum Brechungsindex \tilde{n}_0 die Menge der Geodäten $\mathcal{G}_0 = \mathcal{G}_{\tilde{n}_0}$ und das linearisierte Tikhonov-Funktional $J_{\alpha_0}^0(\tilde{n}_0) = J_{\alpha}^{\tilde{n}_0}(\tilde{n}_0)$.
- (S1) Wiederhole für $k = 1, 2, \dots$

- (a) Minimiere das linearisierte Tikhonov-Funktional, also berechne

$$\tilde{n}_k := \arg \min J_{\alpha}^{\tilde{n}_{k-1}}(\tilde{n}).$$

- (b) Berechne die neuen Geodäten $\mathcal{G}_k = \mathcal{G}_{\tilde{n}_k}$ und werte das Tikhonov-Funktional $J_{\alpha}^k(\tilde{n}_k) = J_{\alpha}^{\tilde{n}_k}(\tilde{n}_k)$ aus.

Ausgabe: Brechungsindex \tilde{n}_k , Geodäten \mathcal{G}_k

Bemerkung 4.0.23

- Für die Eingabe muss ein Startwert für den Brechungsindex \tilde{n}_0 vorgegeben werden. Hier können Zusatzinformationen einfließen. Liegen keine vor, so ist $\tilde{n}_0 \equiv 1$ eine gute Wahl, da wir an der 1-Minimum-Norm-Lösung interessiert sind.
- Um das linearisierte Tikhonov-Funktional auswerten zu können, müssen die Integrationswege bekannt sein. Diese werden in jedem Schritt neu berechnet, deshalb ist das Verfahren auch adaptiv.
- Die Berechnung des Minimierers $\tilde{n}_k := \arg \min J_{\alpha}^{\tilde{n}_{k-1}}(\tilde{n})$ kann sehr aufwändig sein, außerdem ändert sich in jedem Schritt das Tikhonov-Funktional. Man erhält jedoch die Möglichkeit in jedem Schritt den Regularisierungsparameter α_k neu zu wählen.
- Schritt (S1b) wird ausgeführt, indem das Vorwärtsproblem mit einem Verfahren aus Kapitel 3 gelöst wird.
- Als Ausgabe erhält man sowohl eine Näherung an den Brechungsindex \tilde{n}_k , als auch einen Datensatz von Geodäten \mathcal{G}_k . Ist man eigentlich an der Rekonstruktion eines Vektorfeldes interessiert und die Bestimmung des Brechungsindex ein zu lösendes Hilfsproblem, so kann die Kenntniss der Integrationswege in die Rekonstruktion des Vektorfeldes einfließen.

4.1. Herleitung des Abstiegsverfahrens

Im Schritt (S1a) des Algorithmus 4.0.22 soll das Minimum des linearisierten Tikhonov-Funktional (4.0.21) berechnet werden. Wir wollen dazu ein Abstiegsverfahren benutzen, welches auf dem Landweber-Verfahren basiert, in Anlehnung an Schuster u. a. [2012, Kapitel 6 und 7]. Zu einem gegebenem Brechungsindex \tilde{n}_k ist ein Abstiegsverfahren gegeben über

$$\begin{aligned}\tilde{n}_k^0 &= \tilde{n}_k \\ \tilde{n}_k^{l+1} &= \tilde{n}_k^l - \mu_l \left(A_k^*(A_k \tilde{n}_k^l - u^{\text{mess}}) + \frac{\alpha_k}{p} \partial \left(\|\tilde{n}_k^l - 1\|_{L^p(M)}^p \right) \right), \quad l = 0, 1, 2, \dots \\ \tilde{n}_{k+1} &= \tilde{n}_k^\infty\end{aligned}$$

mit $A_k := R_{\tilde{n}_k}$, Regularisierungsparameter $\alpha_k > 0$ und Abstiegsschrittweiten $\mu_l > 0$, $l = 0, 1, \dots$. Die Ableitung der Norm ist gegeben durch

$$\partial \left(\|\tilde{n}_k^l - 1\|_{L^p(M)}^p \right) = p \|\tilde{n}_k^l - 1\|_{L^p(M)}^{p-2} \cdot \tilde{n}_k^l.$$

für $\tilde{n}_k^l \in \mathcal{D}(J)$. Es ergibt sich daraus der folgende Algorithmus als Landweber-Verfahren.

Algorithmus 4.1.1 (Abstiegsverfahren für das lin. Tikhonov-Funktional)

Eingabe: Brechungsindex \tilde{n}_k , Regularisierungsparameter $\alpha_k > 0$.

(S0) Definiere den Operator $R_k = R_{\tilde{n}_k}$ und seine Adjungierte $R_k^* = R_{\tilde{n}_k}^*$.

(S1) Setze $\tilde{n}_k^0 = \tilde{n}_k$ und iteriere für $l = 0, 1, 2, \dots$

(a) Berechne das Residuum $r^{l-1} = R_k \tilde{n}_k^{l-1} - u^{\text{mess}}$ und stoppe, falls $\|r^{l-1}\|_Y = 0$.

(b) Werte den adjungierten Operator aus $r_*^{l-1} = R_k^* r^{l-1}$.

(c) Bestimme die neue Suchrichtung $\Delta^{l-1} = r_*^{l-1} + \alpha_k \|\tilde{n}_k^{l-1} - 1\|_{L^p(M)}^{p-2} \cdot \tilde{n}_k^{l-1}$.

(d) Wähle eine Abstiegsschrittweite $\mu^{l-1} > 0$ und führe einen Update-Schritt durch

$$\tilde{n}_k^l = \tilde{n}_k^{l-1} - \mu^{l-1} \Delta^{l-1}$$

(S2) Setze $\tilde{n}_{k+1} = \tilde{n}_k^l$.

Ausgabe: Brechungsindex \tilde{n}_{k+1} .

Bemerkung 4.1.2

- Damit der Hauptalgorithmus 4.0.22 konvergiert, ist es nicht notwendig, dass der neue Brechungsindex \tilde{n}_{k+1} (auf dessen Grundlage die neuen Geodäten berechnet werden) sehr genau bestimmt wird. Es zeigt sich sogar, dass dies hinderlich für eine Konvergenz sein kann. Der Grund dafür liegt in der Natur des Problems, also der Schlechtgestellttheit. Nimmt man beispielsweise ein sehr inhomogenes Sinogramm als Messdaten an und beginnt den Hauptalgorithmus mit der euklidischen Metrik,

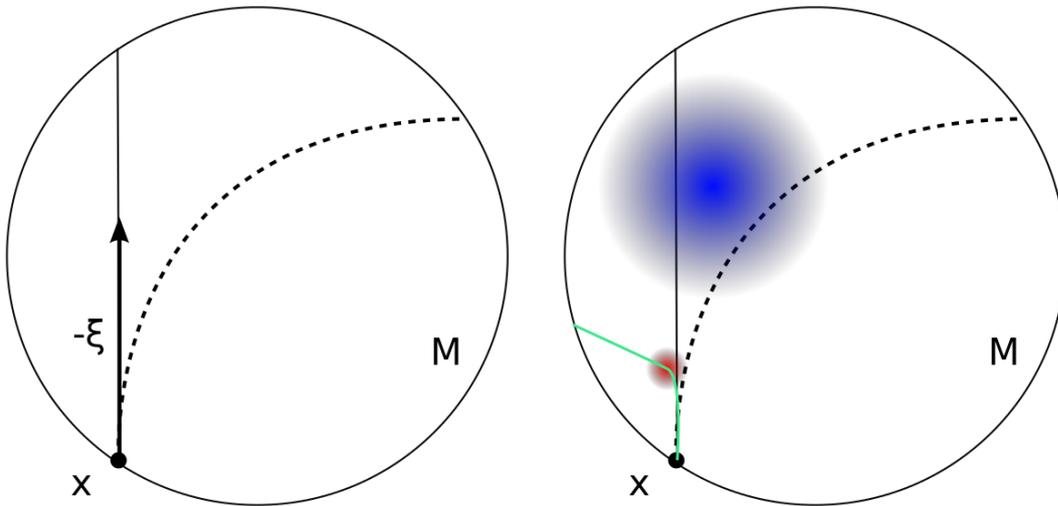


Abbildung 4.1.: Schematische Darstellung des Problems zur exakten Lösung des Hilfsproblems. *Links*: Ausgehend von einer konstanten negativen Krümmung als Lösung, ergäbe sich die gestrichelte Linie als Geodäte vom Punkt x in Richtung ξ . Der Algorithmus 4.0.22 geht im ersten Iterationsschritt von der euklidischen Metrik aus, so dass sich als erste Geodäte ein Geradenstück ergibt (durchgehende Linie). *Rechts*: Da eine konstante negative Krümmung vorliegt, wird der Algorithmus 4.1.1 die Metrik so anpassen, dass die Laufzeit der durchgehenden Linie insgesamt verkürzt wird (roter Spot: Erhöhung der Schallgeschwindigkeit, blauer Spot: Verringerung der Schallgeschwindigkeit). Im nächsten Iterationsschritt werden die Geodäten neu berechnet und es ergibt sich die grüne Kurve. Diese hat sich der echten Lösung nicht angenähert.

so wird der obige Algorithmus einen Brechungsindex liefern, der die Laufzeiten gut approximiert, aber nur bezüglich der euklidischen Wege. Das kann dazu führen, dass der berechnete Brechungsindex \tilde{n}_{k+1} sehr viele hochfrequente Anteile aufweist. Im nächsten Iterationsschritt würden dann zu diesem neuen hochfrequenten Brechungsindex ein Datensatz von Geodäten \mathcal{G}_{k+1} berechnet werden. Diese würden sehr große Krümmungen aufweisen, was wiederum zu erheblich längeren Kurven führen würde. Der Algorithmus 4.1.1 würde dann wieder durch Anpassung des Brechungsindex einen noch stärker oszillierenden \tilde{n}_{k+2} hervorbringen usw. (vergleiche Abbildung 4.1).

Diesem Problem kann man entgegenwirken, indem entweder der Regularisierungsparameter hinreichend groß gewählt wird, oder indem nur wenige Schritte durchgeführt werden. In der Praxis haben sich ein oder zwei Schritte als sinnvoll heraus gestellt.

- Die Auswertung des linearisierten Vorwärtsoperators in Schritt (S1a) ist schnell durchführbar und hängt im Wesentlichen nur von der Diskretisierung der Geodäten

und der Auswertung des Brechungsindex ab.

- Die Berechnung des adjungierten Operators in Schritt (S1b) ist sehr aufwändig. Die Schwierigkeit liegt bei der Interpolation der Geodäten. Im Gegensatz zu einer Modellierung mit Geraden als Wege, ist es schwierig herauszufinden welche Geodäte durch einen gegebenen Punkt $y \in M$, ausgesendet aus der Richtung $\xi \in T\partial M$, verläuft. Im Folgenden wollen wir näher darauf eingehen.
- Der Algorithmus funktioniert für die Wahl $p > 1$. Man kann ihn aber auch erweitern auf den Fall $p = 1$, was wir mit dem Threshold-Algorithmus verwirklichen werden.
- Die Abstiegschrittweite $\mu^l > 0$ aus Schritt (S1d) kann beispielsweise durch einen *line-search-Algorithmus* bestimmt werden. Wie aber bereits angemerkt ist eine exakte Lösung nicht unbedingt erwünscht, so dass eine empirische Wahl eventuell geeigneter erscheint.

Satz 4.1.3 (Konvergenz des Abstiegsverfahrens)

Bei einer geeigneten Wahl der Abstiegschrittweite $\mu^l > 0$, $l = 0, 1, \dots$ bricht der Algorithmus 4.1.1 entweder nach einer endlichen Anzahl an Iterationen mit der 1-Minimum-Norm-Lösung ab oder die Iterierten konvergieren gegen diese.

Beweis. Ein Beweis ist bei Schuster u. a. [2012, Theorem 6.6] zu finden. Dort ist auch eine geeignete Wahl für μ^l im Algorithmus 6.1 angegeben. \square

Wir wollen nun den adjungierten Operator $R_{\tilde{n}_k}^*$ etwas näher betrachten.

Satz 4.1.4 (Adjungierter Operator)

Es sei $a \in X$ ein Brechungsindex. Dann gilt für den adjungierten Operator $R_a^* : Y \rightarrow X$

$$R_a^* \omega(y) = \int_{S^1} \omega(\Pi(y, \xi), \xi) |\det \nabla_y \Pi(y, \xi)| d\xi \text{ für alle } y \in M,$$

wobei $\Pi : M \rightarrow \partial M$ die geodätische Projektion aus Definition 2.3.2 ist.

Beweis. Es seien $\tilde{n} \in X$ und $\omega \in Y$. Dann gilt mit dem Transformationssatz

$$\begin{aligned} \langle R_a \tilde{n}(x, \xi), \omega(x, \xi) \rangle_Y &= \int_{\partial M} \int_{S^1} R_a \tilde{n}(x, \xi) \omega(x, \xi) d\xi dx \\ &= \int_{S^1} \int_{\partial M} \int_{\gamma_{x, \xi}} \tilde{n}(y) \omega(x, \xi) dy dx d\xi \\ &= \int_{S^1} \int_{\mathbb{R}^2} \tilde{n}(y) \omega(\Pi(y, \xi), \xi) |\det \nabla_y \Pi(y, \xi)| dy d\xi \\ &= \int_{\mathbb{R}^2} \tilde{n}(y) \int_{S^1} \omega(\Pi(y, \xi), \xi) |\det \nabla_y \Pi(y, \xi)| dy d\xi \\ &= \int_{\mathbb{R}^2} \tilde{n}(y) R_a^* \omega(y) dy \\ &= \langle \tilde{n}(y), R_a^* \omega(y) \rangle_X \end{aligned}$$

mit der Transformation $x = \Pi(y, \xi)$. \square

4.2. Konvergenzanalyse

Wir wollen nun untersuchen, ob unser Verfahren konvergiert. Hierzu definieren wir Algorithmus 4.1.1 als Operator $T : X \rightarrow X$ über die Zuordnung

$$\tilde{n} \mapsto \arg \min_{a \in X} J_{\alpha}^{\tilde{n}}(a) = \arg \min_{a \in X} \left\{ \frac{1}{2} \|R_{\tilde{n}}a - u^{\text{mess}}\|_Y^2 + \frac{\alpha}{p} \|a - 1\|_X^p \right\}$$

und den Algorithmus 4.0.22 als Iterationsvorschrift

$$\tilde{n}_{k+1} := T(\tilde{n}_k) = \arg \min_{a \in X} J_{\alpha}^{\tilde{n}_k}(a) \quad (4.2)$$

bei gegebenem Startwert $t_0 \in X$.

Satz 4.2.1 (Fixpunkt)

Es sei $T : x \rightarrow X$ eine kontrahierende Abbildung. Dann konvergiert der Algorithmus 4.0.22 gegen genau einen Fixpunkt $\tilde{n}^* \in X$, das heißt es gilt

$$\tilde{n}^* = T(\tilde{n}^*).$$

Beweis. Da $T : X \rightarrow X$ eine kontrahierende Selbstabbildung zwischen vollständigen metrischen Räumen ist, folgt die Behauptung aus dem Fixpunktsatz von Banach. \square

Korollar 4.2.2 (Minimaleigenschaft)

Es sei $\tilde{n}^* \in X$ Fixpunkt von T , also $\tilde{n}^* = \lim_{k \rightarrow \infty} \tilde{n}_k$. Dann gilt

$$J_{\alpha}(\tilde{n}^*) \leq J_{\alpha}^{\tilde{n}^*}(a)$$

für alle $a \in X$.

Beweis. Die Ungleichung folgt aus

$$\tilde{n}^* = T(\tilde{n}^*) = \arg \min_{a \in X} J_{\alpha}^{\tilde{n}^*}(a),$$

also

$$J_{\alpha}^{\tilde{n}^*}(\tilde{n}^*) = \min_{b \in X} J_{\alpha}^{\tilde{n}^*}(b) \leq J_{\alpha}^{\tilde{n}^*}(a)$$

für alle $a \in X$. \square

Bemerkung 4.2.3

Leider gilt im Allgemeinen nicht

$$J_{\alpha}(\tilde{n}^*) \leq J_{\alpha}^a(\tilde{n}^*) \text{ für alle } a \in X, \quad (4.3)$$

wie bereits in Bemerkung 4.1.2 deutlich gemacht wurde. Denn für ein $a \in X$ mit $J_{\alpha}^a(\tilde{n}^*) < J_{\alpha}^{\tilde{n}^*}(\tilde{n}^*)$, würde die Iteration

$$\begin{aligned} \bar{n}_0 &:= a \\ \bar{n}_{k+1} &:= T(\bar{n}_k), \quad k = 0, 1, \dots \end{aligned}$$

zwar gegen den selben Fixpunkt $\lim_{k \rightarrow \infty} \bar{n}_k =: \bar{n}^* = \tilde{n}^*$ konvergieren, da dieser eindeutig ist, jedoch wäre \tilde{n}^* nicht unbedingt ein Minimum von J_{α} . Es muss also zumindest in der Nähe des Fixpunktes die Eigenschaft (4.3) gefordert werden.

Wir wollen nun die Frage beantworten, ob der Algorithmus 4.0.22 wirklich ein Abstiegsverfahren ist. Dazu führen wir ein Surrogatfunktional (siehe Daubechies u. a. [2004] und Lange u. a. [2000]) ein.

Definition 4.2.4 (Surrogatfunktional)

Es sei $F : X \rightarrow \mathbb{R}$ ein Funktional. Dann heißt $G : X \times X \rightarrow \mathbb{R}$ Surrogatfunktional von F , falls

- i) $G(x, a) \geq F(x)$ für alle $(x, a) \in X \times X$,
- ii) $G(a, a) = F(a)$ für alle $a \in X$

gilt.

Wenn wir $\tilde{J}_\alpha(\tilde{n}, a) := J_\alpha^{\tilde{n}}a$ für $\tilde{n}, a \in X$ setzen, dann erfüllt \tilde{J}_α unmittelbar die Eigenschaft ii) aus Definition 4.2.4 bezüglich J_α , denn

$$\tilde{J}_\alpha(\tilde{n}, \tilde{n}) = J_\alpha^{\tilde{n}}\tilde{n} = J_\alpha(\tilde{n}).$$

Außerdem wissen wir aufgrund von Korollar 4.2.2, dass im Fixpunkt $\tilde{n}^* \in X$ aus Satz 4.2.1 die Eigenschaft i) aus Definition 4.2.4 erfüllt ist. Fordern wir diese Eigenschaft und zusätzlich (4.3) in einer Umgebung von \tilde{n}^* , so erhalten wir die folgende Aussage. \tilde{J}_α beschreibt den Algorithmus 4.0.22 durch die Vorschrift

$$\tilde{n}_{k+1} := \arg \min_{a \in X} \tilde{J}_\alpha(\tilde{n}_k, a)$$

für $k = 0, 1, \dots$

Satz 4.2.5 (Abstiegsverfahren)

Es sei $\{\tilde{n}_k\}_{k=0,1,\dots}$ die Folge der Iterierten aus Algorithmus 4.0.22. Es existiere ein $k^* \in \mathbb{N}$ mit

- i) $J_\alpha(\tilde{n}_k) \leq J_\alpha^{\tilde{n}_k}(a)$ und
- ii) $J_\alpha(\tilde{n}_k) \leq J_\alpha^a(\tilde{n}_k)$

für alle $k \geq k^*$ und $a \in X$. Dann sind für $k > k^*$ das Tikhonov- und das Surrogatfunktional monoton fallend, das heißt es gilt sowohl

$$J_\alpha(\tilde{n}_{k+1}) \leq J_\alpha(\tilde{n}_k),$$

als auch

$$\tilde{J}_\alpha(\tilde{n}_{k+1}, \tilde{n}_{k+2}) \leq \tilde{J}_\alpha(\tilde{n}_k, \tilde{n}_{k+1})$$

für $k \geq k^*$.

Beweis. Für $k \geq k^*$ ist \tilde{J}_α nach Voraussetzung ein Surrogatfunktional von J_α und es gilt

$$J_\alpha(\tilde{n}_{k+1}) + (J_\alpha^{\tilde{n}_k}(\tilde{n}_{k+1}) - J_\alpha(\tilde{n}_{k+1})) = \tilde{J}_\alpha(\tilde{n}_k, \tilde{n}_{k+1}) \leq \tilde{J}_\alpha(\tilde{n}_k, \tilde{n}_k) = J_\alpha(\tilde{n}_k),$$

da \tilde{n}_{k+1} ein Minimierer von $\tilde{J}(\cdot, \tilde{n}_k)$ ist. Weiter gilt wegen ii)

$$J_\alpha^{\tilde{n}_k}(\tilde{n}_{k+1}) \geq J_\alpha(\tilde{n}_{k+1}),$$

also $J_\alpha(\tilde{n}_{k+1}) \leq J_\alpha(\tilde{n}_k)$. Für das Surrogatfunktional \tilde{J}_α gilt

$$\tilde{J}_\alpha(\tilde{n}_{k+1}, \tilde{n}_{k+2}) \leq J_\alpha(\tilde{n}_{k+1}) \leq J_\alpha(\tilde{n}_{k+1}) + (J_\alpha^{\tilde{n}_k}(\tilde{n}_{k+1}) - J_\alpha(\tilde{n}_{k+1})) = \tilde{J}_\alpha(\tilde{n}_k, \tilde{n}_{k+1})$$

wegen $J_\alpha^{\tilde{n}_k}(\tilde{n}_{k+1}) - J_\alpha(\tilde{n}_{k+1}) \geq 0$. □

Wir haben also gezeigt, dass Algorithmus 4.0.22 ein Abstiegsverfahren ist, sofern wir schon hinreichend dicht am Fixpunkt sind.

Bemerkung 4.2.6

- Um die Voraussetzungen des Satzes 4.2.5 sicherstellen zu können, muss die Startiererte \tilde{n}_0 hinreichend dicht am Minimum sein. Erst dann können wir einen monotonen Abstieg des Surrogat- und des Tikhonov-Funktionalen erwarten.
- In jedem Schritt des iterativen, adaptiven Minimierungsverfahren 4.0.22 wird ein neues Minimierungsproblem aufgestellt und mit dem Algorithmus 4.1.1 gelöst. Gerade zu Beginn ist allerdings nicht sichergestellt, dass $J_\alpha(\tilde{n}_{k+1}) \leq J_\alpha(\tilde{n}_k)$ gilt. Um diese Eigenschaft aber zu einem gewissen Grad zu erzwingen, könnte man den Algorithmus 4.1.1 modifizieren. Erfüllt die neue Iterierte \tilde{n}_{k+1} , welche durch wenige Iterationsschritte berechnet wurde $J_\alpha(\tilde{n}_{k+1}) > J_\alpha(\tilde{n}_k)$, so kann der Relaxationsparameter μ verkleinert und die Schritte aus Algorithmus 4.1.1 wiederholt werden. Dies würde auch dem Problem der Instabilität aus Bemerkung 4.1.2 entgegenwirken. Allerdings müsste dafür immer wieder der adjungierte Operator ausgewertet werden, was sehr teuer ist.

4.3. Diskretisierung und Implementierung

Wir wollen nun konkret die Algorithmen 4.1.1 und 4.0.22 umsetzen. Hierfür wollen wir zunächst unser Gebiet diskretisieren.

Definition 4.3.1 (Diskretisierung)

Es sei $\tilde{n} : \mathbb{R}^2 \rightarrow \mathbb{R}$ eine stetige Abbildung. Zu einem Parameter $h > 0$ definieren wir das äquidistante Gitter

$$E^h := \{(r_i, s_j) \in \mathbb{R}^2 : r_i = ih, s_j = jh, \forall i, j \in \mathbb{Z}\} \subset \mathbb{R}^2$$

und ein Element $e_{ij}^h \in \mathbb{R}^2$ über

$$e_{ij}^h := \{(r, s) \in \mathbb{R}^2 : r_i \leq r < r_{i+1}, s_j \leq s < s_{j+1}, (r_i, s_j), (r_{i+1}, s_{j+1}) \in E^h\}$$

für $i, j \in \mathbb{Z}$. Auf dem Gitter definieren wir nun die Gitterwerte $F^h(\tilde{n})$ von \tilde{n} durch

$$F^h(\tilde{n}) := \{\tilde{n}_{ij} = \tilde{n}(r_i, s_j) : (r_i, s_j) \in E^h\}$$

und darauf die *bilineare Ansatzfunktion* $\varphi_{ij} : \mathbb{R}^2 \rightarrow \mathbb{R}$ durch

$$\varphi_{ij}(r, s) = \begin{cases} F_{ij}^h(\tilde{n}) + \frac{r-r_i}{h} (F_{i+1,j}^h(\tilde{n}) - F_{i,j}^h(\tilde{n})) \\ + \frac{s-s_j}{h} (F_{i,j+1}^h(\tilde{n}) - F_{i,j}^h(\tilde{n})) \\ + \frac{(r-r_i)(s-s_j)}{h^2} (F_{i+1,j+1}^h(\tilde{n}) - F_{i+1,j}^h(\tilde{n}) - F_{i,j+1}^h(\tilde{n}) + F_{i,j}^h(\tilde{n})) & \text{für } (r, s) \in e_{ij}^h \\ 0 & \text{sonst} \end{cases}$$

Die *Interpolierende* $\tilde{n}^h : \mathbb{R}^2 \rightarrow \mathbb{R}$ von \tilde{n} ist dann gegeben über

$$\tilde{n}^h(r, s) := \sum_{i,j \in \mathbb{Z}} \varphi_{ij}(r, s).$$

Die Menge aller Interpolierenden nennen wir *Ansatzraum*

$$V^h := \left\{ \tilde{n}^h(r, s) := \sum_{i,j \in \mathbb{Z}} \varphi_{ij}(r, s) : \tilde{n} \in X \right\}.$$

Lemma 4.3.2 (Eigenschaften der Interpolierenden)

Sei $\tilde{n} : \mathbb{R}^2 \rightarrow \mathbb{R}$ und seine Interpolierende $\tilde{n}^h : \mathbb{R}^2 \rightarrow \mathbb{R}$. Dann gilt

- a) $\tilde{n}(r_i, s_j) = \tilde{n}^h(r_i, s_j)$ für alle $(r_i, s_j) \in E^h$ und
- b) \tilde{n}^h ist stetig und auf jedem Element im Inneren differenzierbar.

Beweis. Es gilt für $(r_i, s_j) \in E^h$

$$\tilde{n}^h(r_i, s_j) = \sum_{k,l \in \mathbb{Z}} \varphi_{kl}(r_i, s_j) = \varphi_{ij}(r_i, s_j) = F_{ij}^h(\tilde{n}) = \tilde{n}(r_i, s_j).$$

Des Weiteren ist \tilde{n}^h eingeschränkt auf eine Kante eines Elements e_{ij}^h linear und eindeutig durch die Funktionswerte an den Knoten festgelegt. Nach Konstruktion stimmen diese mit dem Nachbarlement überein, denn beispielsweise gilt für $(r, s) \in e_{ij}^h$

$$\begin{aligned} \lim_{(r,s) \rightarrow (r_{i+1}, s_{j+1})} \tilde{n}^h(r, s) &= \lim_{(r,s) \rightarrow (r_{i+1}, s_{j+1})} \sum_{k,l \in \mathbb{Z}} \varphi_{kl}(r, s) \\ &= \lim_{(r,s) \rightarrow (r_{i+1}, s_{j+1})} \left[F_{ij}^h(\tilde{n}) \right. \\ &\quad + \frac{r-r_i}{h} (F_{i+1,j}^h(\tilde{n}) - F_{i,j}^h(\tilde{n})) + \frac{s-s_j}{h} (F_{i,j+1}^h(\tilde{n}) - F_{i,j}^h(\tilde{n})) \\ &\quad \left. + \frac{(r-r_i)(s-s_j)}{h^2} (F_{i+1,j+1}^h(\tilde{n}) - F_{i+1,j}^h(\tilde{n}) - F_{i,j+1}^h(\tilde{n}) + F_{i,j}^h(\tilde{n})) \right] \\ &= F_{ij}^h(\tilde{n}) + (F_{i+1,j}^h(\tilde{n}) - F_{i,j}^h(\tilde{n})) + (F_{i,j+1}^h(\tilde{n}) - F_{i,j}^h(\tilde{n})) \\ &\quad + (F_{i+1,j+1}^h(\tilde{n}) - F_{i+1,j}^h(\tilde{n}) - F_{i,j+1}^h(\tilde{n}) + F_{i,j}^h(\tilde{n})) \\ &= F_{i+1,j+1}^h(\tilde{n}) \\ &= \tilde{n}^h(r_{i+1}, s_{j+1}) \end{aligned}$$

wegen $\frac{r_{i+1}-r_i}{h} = \frac{(i+1)h-ih}{h} = 1$ und somit ist \tilde{n}^h stetig. Im Inneren eines Elements e_{ij}^h ist \tilde{n}^h ein Polynom, also differenzierbar. \square

Bemerkung 4.3.3

- Die Interpolierende \tilde{n}^h ist eindeutig durch die Knotenwerte festgelegt.
- $h > 0$ heißt *Diskretisierungsparameter*. Für die numerischen Berechnungen werden wir uns später auf die Einheitskreisscheibe M zurückziehen und dieses Gebiet durch eine Teilmenge des Gitters E^h überdecken. Je kleiner h ist, desto mehr Punkte enthält diese Teilmenge, sie wächst wie $(1/h)^2$.
- Es sind durchaus auch andere Diskretisierungen und Ansatzfunktionen denkbar. Beispielsweise könnte man das Gebiet durch Dreieckselemente diskretisieren und darauf Hütchenfunktionen definieren. Jedoch hat die obige Diskretisierung einen entscheidenden Vorteil. Zu einem gegebenen Punkt $(r, s) \in M$ ist es sehr einfach zu entscheiden in welchem Element e_{ij}^h dieser liegt. Ein Nachteil ist, dass komplizierte Geometrien nicht so gut approximiert werden können.
- Für die numerische Umsetzung ist es wichtig, dass \tilde{n}^h überall differenzierbar ist. Um dies zu erreichen wird die Ableitung zu den Rändern eines Elements e_{ij}^h eindeutig fortgesetzt.

Wir betrachten nun die Einheitskreisscheibe M und beschränken das zu diskretisierende Gebiet auf das Quadrat $Q := [-1, 1] \times [-1, 1] \supset M$. Hierzu wählen wir den Diskretisierungsparameter $n \in \mathbb{N}$ und setzen $h := 1/n$. Das Gitter verkleinern wir dann auf $\tilde{E}^n := E^h \cap Q$ und es gilt $|\tilde{E}^n| = (2n + 1)^2$, sowie

$$|M \cap \tilde{E}^n| = \left| \left\{ (r_i, s_j) \in \tilde{E}^n : \|(r_i, s_j)\| \leq 1 \right\} \right| \approx \frac{\pi}{4} |\tilde{E}^n| = \frac{\pi}{4} (2n + 1)^2,$$

so dass die Anzahl der Freiheitsgrade von $\tilde{n}^n := \tilde{n}^h$ wie $\mathcal{O}(n^2)$ wächst, selbst wenn man alle Gitterwerte außerhalb von M auf Eins setzt.

Betrachten wir nun die Messdaten u^{mess} . In der Praxis liegen nur unvollständige Messdaten vor, u^{mess} ist also nur auf einer endlichen Teilmenge $P \subset \partial\Omega M$ bekannt. Diese Menge wollen wir festlegen wie in Abschnitt 3.1.1, also zu $N_x, N_\xi \in \mathbb{N}$ seien

$$X_{N_x} := \left\{ x_i = \begin{pmatrix} \sin\left(2\pi \frac{i-1}{N_x}\right) \\ \cos\left(2\pi \frac{i-1}{N_x}\right) \end{pmatrix} : i = 1, \dots, N_x \right\}$$

die Menge der Aufpunkte und

$$\Xi_i^{N_\xi} := \left\{ \xi_i^j = \begin{pmatrix} \sin\left(2\pi \left(\frac{i-1}{N_x} + \frac{j-1}{N_\xi} - \frac{1}{4}\right)\right) \\ \cos\left(2\pi \left(\frac{i-1}{N_x} + \frac{j-1}{N_\xi} - \frac{1}{4}\right)\right) \end{pmatrix} : j = 1, \dots, N_\xi \right\}$$

die Menge der Richtungen für $i = 1, \dots, N_x$. Übertragen wir diese Situation auf das Minimierungsproblem 4.1, so erhalten wir mit der Festsetzung $N := (N_x, N_\xi)$ und

$$P := P^N := \left\{ (x_i, \xi_i^j) : x_i \in X_{N_x}, \xi_i^j \in \Xi_i^{N_\xi}, i = 1, \dots, N_x, j = 1, \dots, N_\xi \right\}$$

das diskretisierte Tikhonov-Funktional $J_\alpha^{n,N} := J_\alpha^{n,P} : V^n \rightarrow \mathbb{R}$, gegeben durch

$$\begin{aligned} J_\alpha^{n,N}(\tilde{n}^k) &= \frac{1}{2} \|R^\kappa(\tilde{n}^k) - u^{\text{mess}}\|_{L^2(P)}^2 + \alpha \Sigma(\tilde{n}^k) \\ &= \frac{1}{2} \sum_{(x,\xi) \in P^N} |R^\kappa(\tilde{n}^k)(x,\xi) - u^{\text{mess}}(x,\xi)|^2 + \frac{\alpha}{p} \sum_{(r_i,s_j) \in \tilde{E}^n} |\tilde{n}^k(r_i,s_j) - 1|^p, \end{aligned}$$

wobei R^κ berücksichtigt, dass \tilde{n}^k nur stetig und ein mal differenzierbar ist. R^κ bestimmt also die Integralkurven als approximative Lösung des Anfangswertproblems 3.2 zum Parameter $\kappa = (\rho, h_0, \varepsilon) \in \mathbb{R}^3$.

Einen Datensatz Geodäten bezeichnen wir als

$$\begin{aligned} \bar{\Gamma}_N^\kappa &:= \left\{ \gamma_{x_i, \xi_i^j} : \gamma_{x_i, \xi_i^j} \text{ ist Näherungslösung des AWP (3.2)}, \right. \\ &\quad \left. x_i \in X_{N_x}, \xi_i^j \in \Xi_i^{N_\xi}, i = 1, \dots, N_x, j = 1, \dots, N_\xi \right\}. \end{aligned}$$

Wir wollen nun den adjungierten Operator R_k^* im k -ten Iterationsschritt, gegeben durch

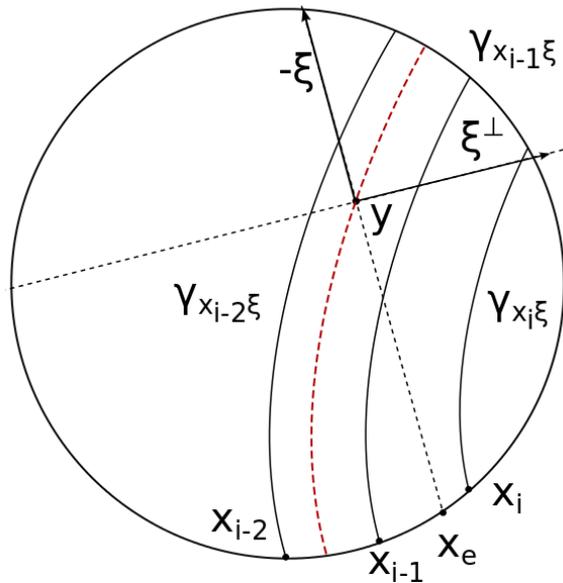


Abbildung 4.2.: Schematische Darstellung von Algorithmus 4.3.4.

$$R_k^* \omega(y) = \int_{S^1} \omega(\Pi(y, \xi), \xi) |\det \nabla_y \Pi(y, \xi)| d\xi \text{ für alle } y \in M,$$

approximieren, wobei der Brechungsindex in die geodätische Projektion $\Pi = \Pi(\tilde{n}_k)$ ein- geht. Für dessen Berechnung im Schritt (S1b) des Algorithmus 4.1.1 muss also zu einem gegebenen Punkt $y \in M$ über alle Geodäten integriert werden, die diesen Punkt durch- laufen. Das Vorgehen ist also ähnlich wie bei der Computertomographie, also der Model- lierung mit geradlinigen Strahlen. Es ergeben sich jedoch zwei gravierende Unterschiede.

1. Die Determinante des Gradienten der Projektion $|\det \nabla_y \Pi(y, \xi)|$ ist bei der Modellierung mit Strahlen identisch Eins. In unserer Modellierung ist die Abbildung jedoch nicht explizit bekannt.
2. Bei der Modellierung mit Strahlen ist es sehr einfach zu entscheiden, welche Strahlen durch einen gegebenen Punkt $y \in M$ laufen, bzw. bei endlichen Daten kann man leicht zwischen zwei vorbeilaufenden Strahlen interpolieren. Der Grund dafür ist, dass die geodätische Projektion in diesem Fall eine einfache euklidische Projektion auf den Kreis ist.

In unserem Fall ist die Projektion Π nicht explizit gegeben, so dass zu $y \in M$ nicht unmittelbar ein Randpunkt $x \in \partial M$ lokalisiert werden kann.

Da wir grundsätzlich von einer Schallgeschwindigkeit ausgehen, die nahe der Referenzschallgeschwindigkeit liegt, können wir davon ausgehen, dass

$$|\det \nabla_y \Pi(y, \xi)| \approx 1$$

gilt. Es bleibt das Problem der Unkenntnis der geodätischen Projektion Π , also der Frage danach, wie zu einem gegebenem Punkt $y \in M$ und einer Richtung $\xi \in S^1$ ein Randpunkt bestimmt werden kann, so dass ausgehend von diesem eine Geodäte in Richtung ξ den Punkt y durchläuft. Hierfür wählen wir den folgenden Algorithmus.

Algorithmus 4.3.4 (Berechnung des adjungierten Operators)

Eingabe: Datensatz Geodäten $\bar{\Gamma}_N^\kappa$ und deren Werte $\omega(\bar{\Gamma}_N^\kappa)$, Punkt $y \in M$.

(S0) Setze $0 = \omega \in \mathbb{R}^{N_\xi}$.

(S1) Iteriere für $j = 1, 2, \dots, N_\xi$

- (a) Fixiere eine Richtung $\xi^j \in \Xi_1^{N_\xi}$ und bestimme dessen Senkrechte $(\xi^j)^\perp$.
- (b) Projiziere den Punkt y auf den Rand in Richtung ξ^j auf den Randpunkt $x_e \in \partial M$ und wähle den Aufpunkt $x_i \in X_{N_x}$, der am dichtesten zu x_e liegt.
- (c) Falls $y \in \gamma_{x_i, \xi^j}$, setze $\omega_j = \omega(\gamma_{x_i, \xi^j})$ gehe zu Schritt (S1).
- (d) Falls γ_{x_i, ξ^j} links an y bezüglich $(\xi^j)^\perp$ vorbei läuft, so überprüfe,
 - i) ob γ_{x_{i+1}, ξ^j} auch links an y bezüglich $(\xi^j)^\perp$ vorbei läuft. Falls ja, so setze $i = i + 1$ und gehe zu (S1d).
 - ii) Andernfalls interpoliere ω_j zwischen $\omega(\gamma_{x_i, \xi^j})$ und $\omega(\gamma_{x_{i+1}, \xi^j})$ und gehe zu (S1).
- (e) Falls γ_{x_i, ξ^j} rechts an y bezüglich $(\xi^j)^\perp$ vorbei läuft, so überprüfe,
 - i) ob γ_{x_{i-1}, ξ^j} auch rechts an y bezüglich $(\xi^j)^\perp$ vorbei läuft. Falls ja, so setze $i = i - 1$ und gehe zu (S1e).
 - ii) Andernfalls interpoliere ω_j zwischen $\omega(\gamma_{x_i, \xi^j})$ und $\omega(\gamma_{x_{i-1}, \xi^j})$ und gehe zu (S1).

(S2) Approximiere den adjungierten Operator durch Berechnung des Integrals

$$R_k^*(y) \approx \frac{2\pi}{N_\xi} \sum_{j=1}^{N_\xi} \omega_j.$$

Ausgabe: Adjungierter Operator $R_k^*(y)$.

Der Algorithmus funktioniert also folgendermaßen (vergleiche Abbildung 4.2): Zunächst wird der Punkt y euklidisch auf einen Randpunkt $x_e \in \partial M$ in Richtung ξ projiziert, wobei nach und nach alle Richtungen durchlaufen werden. Anschließend wird der zu x_e dichteste Punkt x_i im Datensatz P^N fixiert. Die zugehörige Geodäte $\gamma_{x_i, \xi}$ verläuft nun rechts von y bezüglich der affinen Hyperebene $y + \text{span}(\xi^\perp)$.

Der Algorithmus betrachtet nun die weiter links liegende Geodäte $\gamma_{x_{i-1}, \xi}$ und entscheidet, ob diese links oder rechts von y verläuft. In diesem Beispiel verläuft sie abermals rechts von y , so dass i abermals dekrementiert wird. Schließlich liegt y genau zwischen den Geodäten, die von den Aufpunkten x_{i-1} und x_{i-2} ausgehen. Die rot gestrichelte Linie stellt dabei die interpolierte Geodäte dar, die nicht im Datensatz vorhanden ist. An dieser sind wir aber auch nicht interessiert sind, sondern ausschließlich an der interpolierten Laufzeit.

Bemerkung 4.3.5

- Liegt ein Punkt $y \in M$ sehr dicht am Rand ∂M , so kann es vorkommen, dass keine zwei Geodäten existieren, so dass der Punkt zwischen diesen liegt. In diesem Fall kann man die Laufzeit der nächstgelegenen Geodäten mit 0 interpolieren.
- Es ist wichtig festzulegen worauf sich “links oder rechts liegen von” bezieht. Im CT-Fall ist das trivial, hier allerdings nicht. Die Wahl, sich auf die affine Hyperebene $y + \text{span}(\xi^\perp)$ zu beziehen, ist naheliegend, aber muss keinesfalls die besten Ergebnisse hervorbringen.
- Beim In- bzw. Dekrementieren der Aufpunkte muss modulo N_x gerechnet werden.
- Die Interpolation der Laufzeit (bei Kenntnis der beiden umschließenden Geodäten) kann auf verschiedene Arten geschehen. Hier wurden die Laufzeiten linear gewichtet bezüglich der Entfernung von y zu den Schnittpunkten $\gamma_{x_i, \xi^j} \cap (y + \text{span}(\xi^\perp))$ und $\gamma_{x_{i\pm 1}, \xi^j} \cap (y + \text{span}(\xi^\perp))$.

Schließlich wollen wir uns noch der Auswertung der Ableitung des Regularisierungsterms ((S1c) aus Algorithmus 4.1.1) widmen. Für $p > 1$ gilt

$$\partial \left(\|\tilde{n}_k^l - 1\|_{L^2(M)}^p \right) = p \|\tilde{n}_k^l - 1\|_{L^p(M)}^{p-2} \cdot \tilde{n}_k^l,$$

was auch unmittelbar auf die Diskretisierung übertragen werden kann. Wir wollen den Algorithmus aber für den Fall $p = 1$ erweitern. Das Problem hierbei ist, dass in diesem Fall die Norm nicht mehr differenzierbar ist. Daher verwenden wir einen Threshold-Algorithmus, welcher bei Daubechies u. a. [2004] vorgeschlagen wird.

Hierfür ist es notwendig, dass die zu minimierende Funktion einen kompakten Träger hat. Daher setzen wir $\tilde{m} := \tilde{n} - 1$ und erhalten

$$\begin{aligned} J_\alpha^k(\tilde{n}) &= \frac{1}{2} \|R_k(\tilde{n}) - u^{\text{mess}}\|_Y^2 + \alpha \|\tilde{n} - 1\|_{L^1(M)} \\ &= \frac{1}{2} \|R_k(\tilde{m} + 1) - u^{\text{mess}}\|_Y^2 + \alpha \|\tilde{m}\|_{L^1(M)} \\ &= \frac{1}{2} \|R_k(\tilde{m}) - (u^{\text{mess}} - R_k 1)\|_Y^2 + \alpha \|\tilde{m}\|_{L^1(M)} \\ &=: K_\alpha^k(\tilde{m}). \end{aligned}$$

Es gilt dann $\min J_\alpha^k(\tilde{n}) = \min K_\alpha^k(\tilde{m})$. Ist $\{\varphi_i\}_{i \in \bar{I}}$ eine Orthogonalbasis von X , so gilt

$$K_\alpha^k(\tilde{m}) = \frac{1}{2} \|R_k(\tilde{m}) - \tilde{u}^{\text{mess}}\|_Y^2 + \alpha \sum_{i \in \bar{I}} |\langle \tilde{m}, \varphi_i \rangle| \quad (4.4)$$

mit der Setzung $\tilde{u}^{\text{mess}} := u^{\text{mess}} - R_k 1$. Es gilt der folgende Satz

Satz 4.3.6 (Konvergenz des Threshold-Algorithmus)

Es sei $\|R_k\| < 1$ und $\tilde{m}^0 \in X$. Dann konvergiert der Threshold-Algorithmus, gegeben durch

$$\tilde{m}^l = \mathcal{S}_\alpha(\tilde{m}^{l-1} - R_k^*(R_k \tilde{m}^{l-1} - u^{\text{mess}})), \quad l = 1, 2, \dots$$

gegen einen Minimierer von (4.4). Dabei ist der Operator \mathcal{S}_α gegeben durch

$$\mathcal{S}_\alpha(g) = \sum_{i \in \bar{I}} S_\alpha(\langle g, \varphi_i \rangle) \varphi_i$$

mit der Threshold-Funktion

$$S_\alpha(r) = \begin{cases} r + \frac{\alpha}{2} & , \text{ falls } r \leq -\frac{\alpha}{2} \\ 0 & , \text{ falls } |r| < \frac{\alpha}{2} \\ r - \frac{\alpha}{2} & , \text{ falls } r \geq \frac{\alpha}{2} \end{cases}.$$

Beweis. Ein Beweis ist bei Daubechies u. a. [2004] zu finden. □

Führen wir noch einen Dämpfungsparameter $\mu > 0$ ein, so erhalten wir aus der jetzigen Iterierten \tilde{n}_k^l die neue Iterierte \tilde{n}_k^{l+1} durch die Vorschrift

$$\begin{aligned} \tilde{n}_k^{l+1} &= \tilde{m}_k^l + 1 \\ &= 1 + \mathcal{S}_\alpha(\tilde{m}_k^l - \mu R_k^*(R_k \tilde{m}_k^l - \tilde{u}^{\text{mess}})) \\ &= 1 + \mathcal{S}_\alpha(\tilde{n}_k^l - 1 - \mu R_k^*(R_k \tilde{n}_k^l - R_k 1 - u^{\text{mess}} + R_k 1)) \\ &= 1 + \mathcal{S}_\alpha(\tilde{n}_k^l - 1 - \mu R_k^*(R_k \tilde{n}_k^l - u^{\text{mess}})). \end{aligned} \quad (4.5)$$

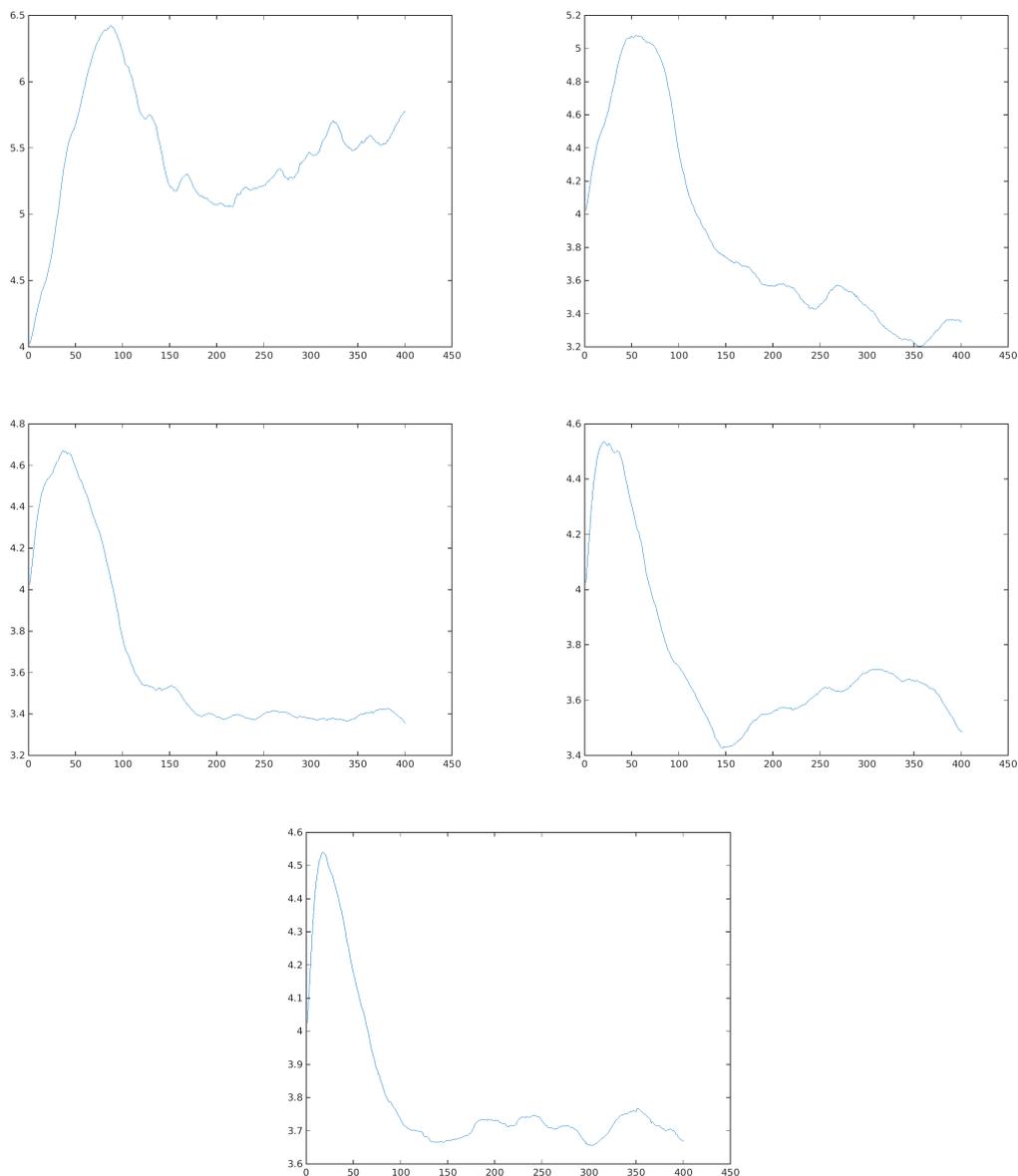


Abbildung 4.3.: Zielfunktionale $J_{\alpha}^{n,N,k}(\tilde{n}_{k+1})$ zu verschiedenen Regularisierungsparametern α . Auf der y-Achse ist jeweils der Wert des k -ten Zielfunktionalen and der Stelle \tilde{n}_{k+1} ausgewertet. Auf der x-Achse ist der Iterationsschritt abgetragen. *Oben links:* $\alpha_1 = 0.3$, *Oben rechts:* $\alpha_2 = 0.9$, *Mitte links:* $\alpha_3 = 1.5$, *Mitte rechts:* $\alpha_4 = 2.1$, *Unten:* $\alpha_5 = 2.7$.

4.4. Numerische Ergebnisse

Das iterative, adaptive Regularisierungsverfahren 4.0.22 haben wir in Matlab implementiert. Zur Lösung der auftretenden linearen Optimierungsprobleme wurde Algorithmus 4.0.22 umgesetzt und zur Berechnung der auftauchenden adjungierten Operatoren wurde Algorithmus 4.3.4 implementiert. Der Quellcode ist im Anhang zu finden. Ziel ist es die theoretischen Überlegungen mit den praktischen Ergebnissen zu vergleichen und die berechneten Lösungen auf Parametervariationen zu untersuchen. Zur Berechnung benutzen wir einen Intel Xeon E5-2650 mit 2.0 GHz (max. 2.8 GHz) und 384 GB Arbeitsspeicher. Sofern wir den Code parallelisiert haben, nutzen wir 30 Kerne.

Die Berechnung der Geodäten erfolgt nun ausschließlich mit der Charakteristikenmethode aus Abschnitt 3.1, da das in Abschnitt 3.2 vorgeschlagene Verfahren (Modellierung der Geodäten als Kürzeste) etwa um den Faktor 100 langsamer ist. Pro Iterationsschritt muss jedoch das Vorwärtsproblem zumindest einmalig gelöst werden, so dass der Faktor direkt auf das Rekonstruktionsverfahren durchgereicht wird.

Da jede Geodäte unabhängig von jeder anderen berechnet werden kann, schlägt der verstärkte Einsatz von mehr Rechenkernen unmittelbar auf die Gesamtrechenzeit durch. Ein ähnliches Verhalten ist bei der Auswertung der adjungierten Operatoren zu beobachten, denn hier kann sowohl bezüglich der Punkte, als auch bezüglich der Richtungen parallelisiert werden. Numerische Experimente hatten ergeben, dass eine Parallelisierung in den Richtungen Geschwindigkeitsvorteile bringt. Somit lassen sich glücklicherweise die rechenintensivsten Schritte ausgezeichnet parallelisieren und somit Rekonstruktionen in einer akzeptablen Laufzeit erzeugen.

Das Zielfunktional $J_\alpha^{n,N,k}$ ist im k -ten Schritt für $p \geq 1$ gegeben über

$$\begin{aligned} J_\alpha^{n,N,k}(\tilde{n}) &= \frac{1}{2} \|R_k^\kappa(\tilde{n}) - u^{\text{mess}}\|_{L^2(P)}^2 + \alpha \Sigma(\tilde{n}) \\ &= \frac{1}{2} \sum_{(x,\xi) \in P^N} |R_k^\kappa(\tilde{n}^k)(x, \xi) - u^{\text{mess}}(x, \xi)|^2 + \frac{\alpha}{p} \sum_{(r_i, s_j) \in \tilde{E}^n} |\tilde{n}(r_i, s_j) - 1|^p, \end{aligned}$$

wobei n die Diskretisierungsstufe im Gebiet M und N den Diskretisierungsgrad der Messdaten \mathcal{G} beschreibt. Da die zu rekonstruierenden Schallgeschwindigkeiten $c : M \rightarrow \mathbb{R}^+$ in unseren Beispielen bekannt sind, lässt sich die Abweichung der Rekonstruktion zum Original im k -ten Schritt berechnen über

$$\mathcal{K}_k := \left(\sum_{(r,s) \in \tilde{E}^n} |c_k(r, s) - c(r, s)|^p \right)^{\frac{1}{p}},$$

wobei $c_k(r, s) := \tilde{n}_k^{-1}(r, s)$ die rekonstruierte Schallgeschwindigkeit in den Punkten $(r, s) \in M$ sei.

Beispiel 4.4.1 (Regularisierungsparameter)

Im ersten Beispiel untersuchen wir die Konvergenz in Hinblick auf die Wahl des Regularisierungsparameters $\alpha > 0$ bei fester Norm $p = 2$. Wir wählen in jedem Schritt $\alpha_k = \bar{\alpha}$

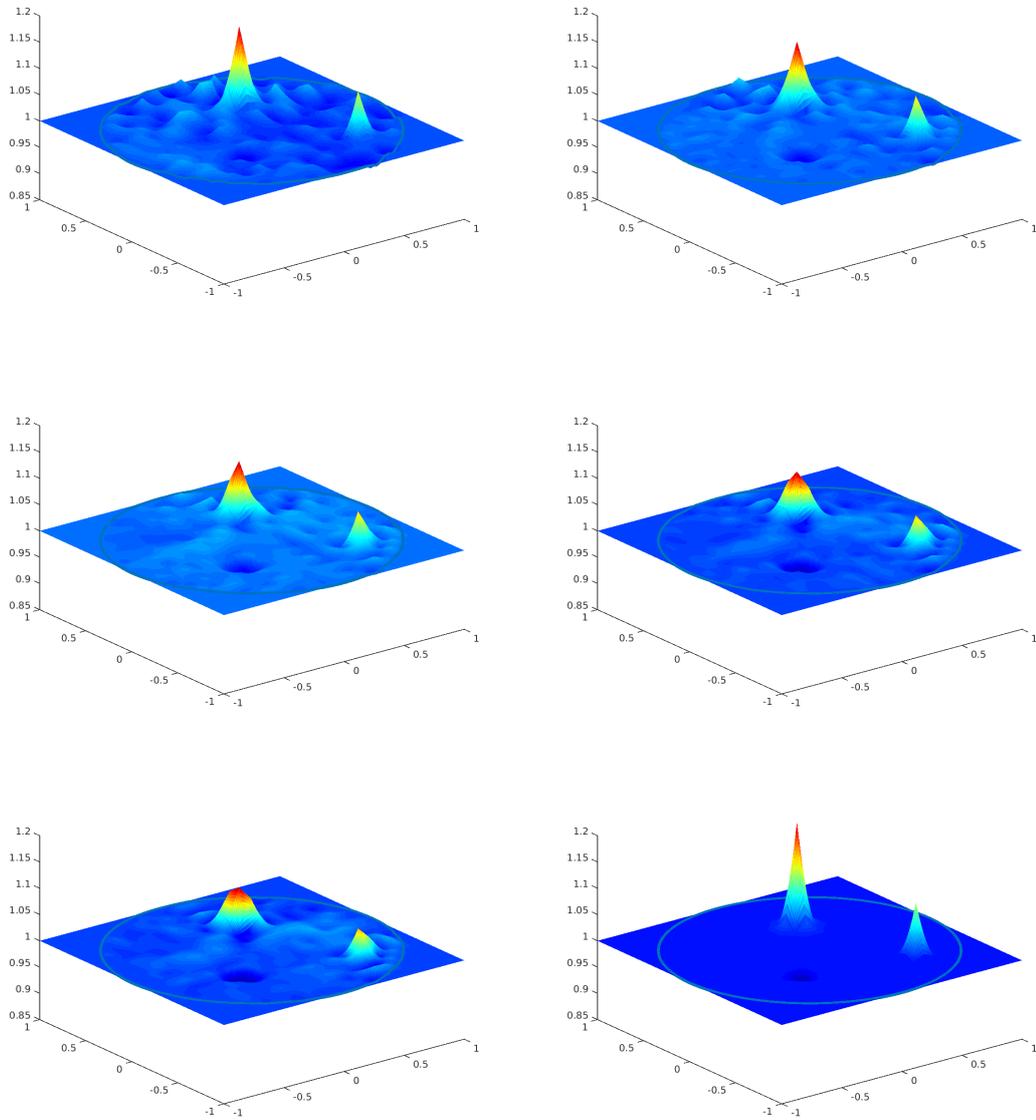


Abbildung 4.4.: Schallgeschwindigkeiten aus der Vogelperspektive. *Oben links:* $\alpha_1 = 0.3$, *Oben rechts:* $\alpha_2 = 0.9$, *Mitte links:* $\alpha_3 = 1.5$, *Mitte rechts:* $\alpha_4 = 2.1$, *Unten links:* $\alpha_5 = 2.7$, *Unten rechts:* zu rekonstruierende Schallgeschwindigkeit.

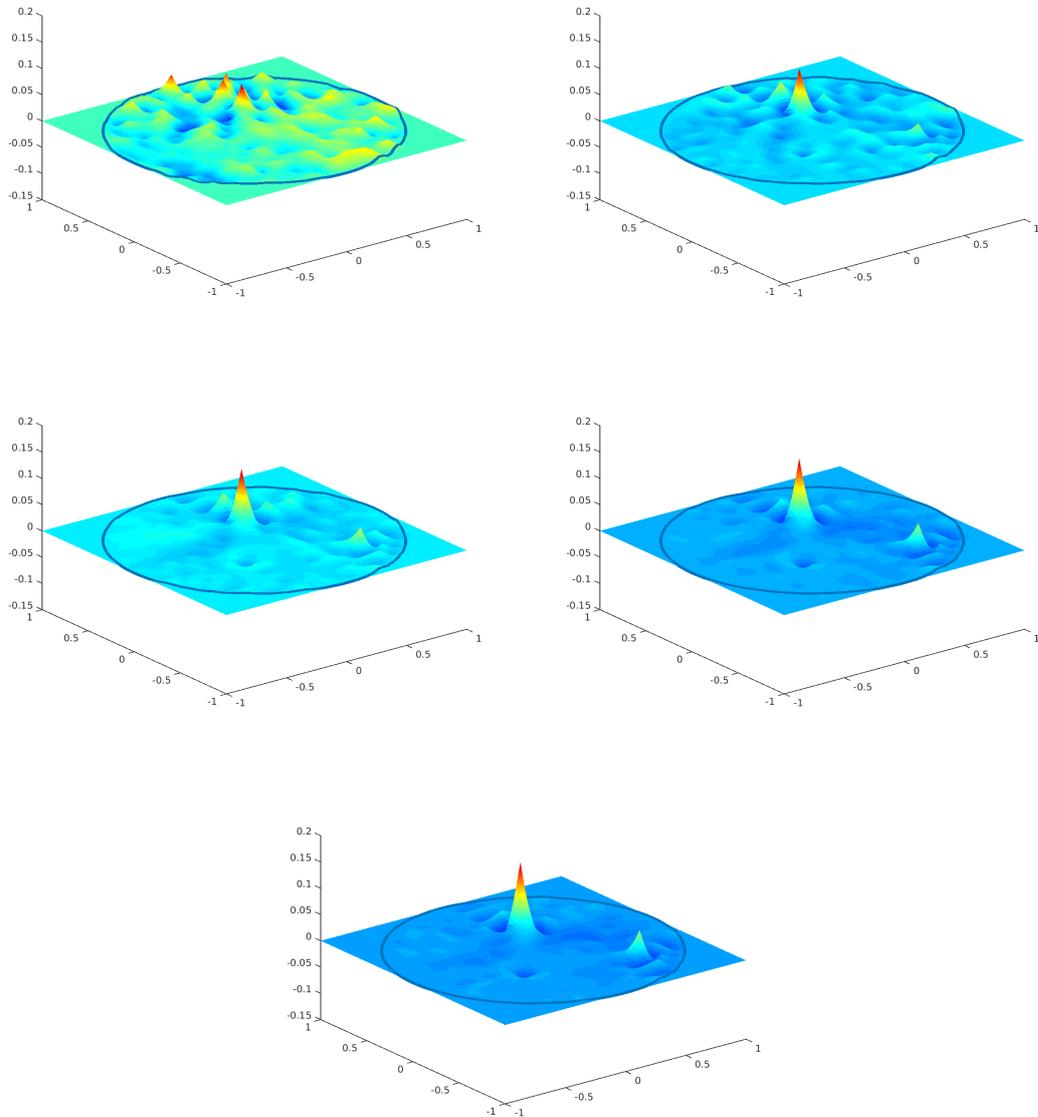


Abbildung 4.5.: Differenz der originalen und rekonstruierten Schallgeschwindigkeiten aus der Vogelperspektive. *Oben links:* $\alpha_1 = 0.3$, *Oben rechts:* $\alpha_2 = 0.9$, *Mitte links:* $\alpha_3 = 1.5$, *Mitte rechts:* $\alpha_4 = 2.1$, *Unten:* $\alpha_5 = 2.7$.

konstant. Als Messdaten betrachten wir die Metrik ‐Peaks‐, also

$$c(x) = 1 + \sum_{i=1}^3 \varphi_i(x) \chi_i(x)$$

mit

$$\varphi_i(x) = \theta_i e^{-\frac{1}{r_i - \|x - p_i\|}}$$

und

$$\chi_i(x) = \begin{cases} 1 & \text{falls } \|x - p_i\| \leq r_i \\ 0 & \text{sonst} \end{cases}$$

für gegebene Zentren $p_i \in M$, Radien $r_i > 0$ und Amplituden $\theta_i \in (-1, \infty)$, $i = 1, 2, 3$. Wie in Beispiel 3.1.8 wählen wir als Parameter die Zentren

$$p_1 = \begin{pmatrix} \frac{1}{5} \\ \frac{3}{5} \end{pmatrix}, p_2 = \begin{pmatrix} -\frac{1}{3} \\ -\frac{1}{3} \end{pmatrix}, p_3 = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$$

die Radien

$$r_1 = \frac{1}{4}, r_2 = \frac{1}{5}, r_3 = \frac{1}{6}$$

und die Amplituden

$$\theta_1 = \frac{1}{5}, \theta_2 = -\frac{3}{20}, \theta_3 = \frac{1}{10}.$$

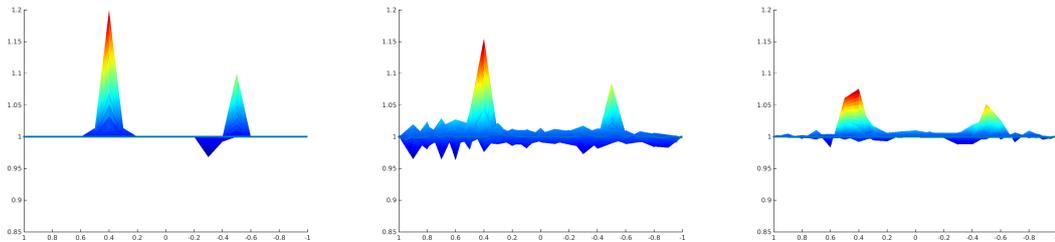


Abbildung 4.6.: Original- und rekonstruierte Schallgeschwindigkeiten von der Seite. *Links:* Original, *Mitte:* $\alpha_1 = 0.3$ *Rechts:* $\alpha_5 = 2.7$.

Pro Iterationsschritt wählen wir $N_x = 100$ Aufpunkte und $N_\xi = 100$ Richtungen pro Aufpunkt, so dass wir in jedem Schritt $|\bar{\Gamma}_N| = |\bar{\Gamma}_{N_x, N_\xi, k}^\beta| = 10000$ Geodäten berechnen mit dem Parameter $\beta = (0.9, 0.05, 10^{-3})$ (vergleiche Abschnitt 3.1.1). Die Berechnung eines vollen Datensatzes Γ_N dauert etwa eine halbe Minute, die Auswertung des adjungierten Operators R_k^* dauert über eine Minute. Ein Satz Geodäten benötigt dabei etwa 13 MB Speicher.

Gerechnet wurde für die Regularisierungsparameter $\alpha_i = 0.3 + 0.6i$, $i = 1, \dots, 5$. Es wurde in jedem Iterationsschritt genau ein Landweber-Schritt (Algorithmus 4.1.1) durchgeführt mit Dämpfungsparameter $\mu = 0.01$. Das Gebiet wurde mit dem Parameter $n = 10$ diskretisiert, so dass insgesamt 346 Freiheitsgrade vorlagen. Pro Regularisierungsparameter dauerte eine Rechnung mit insgesamt 400 Iterationsschritten relativ konstant 15 Stunden.

Abbildung 4.3 zeigt die entsprechenden Zielfunktionale $J_\alpha^{n,N,k}(\tilde{n}_{k+1})$ zu den verschiedenen Regularisierungsparametern α_i . Man erkennt bei allen Funktionalen zunächst einen Anstieg und bei allen, bis auf dem ersten, einen Abfall bis zu einem lokalen Minimum, welches unterhalb des Startwertes liegt. Anhand dieser Kurven wurden Iterationspunkte k_i^* , $i = 1, \dots, 5$ gewählt, bei denen ein lokales Minimum erreicht wird:

$$k_1^* = 216, k_2^* = 246, k_3^* = 240, k_4^* = 147 \text{ und } k_5^* = 303.$$

Zu diesen Zeitpunkten sind die rekonstruierten Schallgeschwindigkeiten in Abbildung 4.4 abgetragen, wobei der Plot unten rechts die Ausgangsschallgeschwindigkeit c ist. Abbildung 4.5 zeigt jeweils die Differenz zu c . Dabei fällt auf, dass je geringer der Regularisierungsparameter α gewählt wurde, desto höher die Schwankungen in den Rekonstruktionen sind. Außerdem werden bei allen Regularisierungsparametern die "Peaks" gut detektiert, deren Ausmaß jedoch umso schlechter, je größer der Regularisierungsparameter ist. So hat im Punkt $(0.2, 0.4)$ die Ausgangsschallgeschwindigkeit einen Wert von $c(0.2, 0.4) = 1.2$ und die schwach regularisierte Rekonstruktion einen Wert von $c_1(0.2, 0.4) \approx 1.15$, jedoch die stark regularisierte Lösung nur einen Wert von $c_5(0.2, 0.4) \approx 1.08$ (vergleiche Abbildung 4.6).

Dies war jedoch zu erwarten, denn je größer α ist, desto stärker wird der Strafterm im Verhältnis zum Datenterm beim Tikhonov-Funktional gewichtet. Ein großes α sollte also eine geringe Abweichung der Rekonstruktion zur euklidischen Metrik erzwingen. Genau das ist zu beobachten.

Der schwache Ausschlag der Schallgeschwindigkeit im Punkt $(-\frac{1}{3}, -\frac{1}{3})$ wird erst bei einem hohem Regularisierungsparameter detektiert. Ist dieser zu niedrig, so geht der Ausschlag an dieser Stelle im "Rauschen" unter.

Beispiel 4.4.2 (p -Normen im Regularisierungsterm)

Wir betrachten wie im vorherigen Beispiel die Schallgeschwindigkeit "Peaks", nun jedoch mit Fokus auf die p -Norm im Regularisierungsterm. Es wurden die Normen

$$p_1 = 1, p_2 = 1.1, p_3 = 2, p_4 = 4 \text{ und } p_5 = 10$$

untersucht. Im Falle $p_1 = 1$ wurde das Threshold-Verfahren 4.5 benutzt, so dass hier der Regularisierungsparameter $\alpha = 0.01$, in den anderen Fällen jedoch $\alpha = 0.2$, gewählt wurde. Pro Iterationsschritt wählen wir $N_x = 80$ Aufpunkte und $N_\xi = 80$ Richtungen pro Aufpunkt, so dass in jedem Schritt $|\bar{\Gamma}_N| = |\bar{\Gamma}_{N_x, N_\xi, k}^\beta| = 6400$ Geodäten mit dem Parameter $\beta = (0.9, 0.05, 10^{-3})$ berechnet werden.

Die Berechnung eines vollen Datensatzes Γ_N dauert etwa 70 Sekunden, die Auswertung

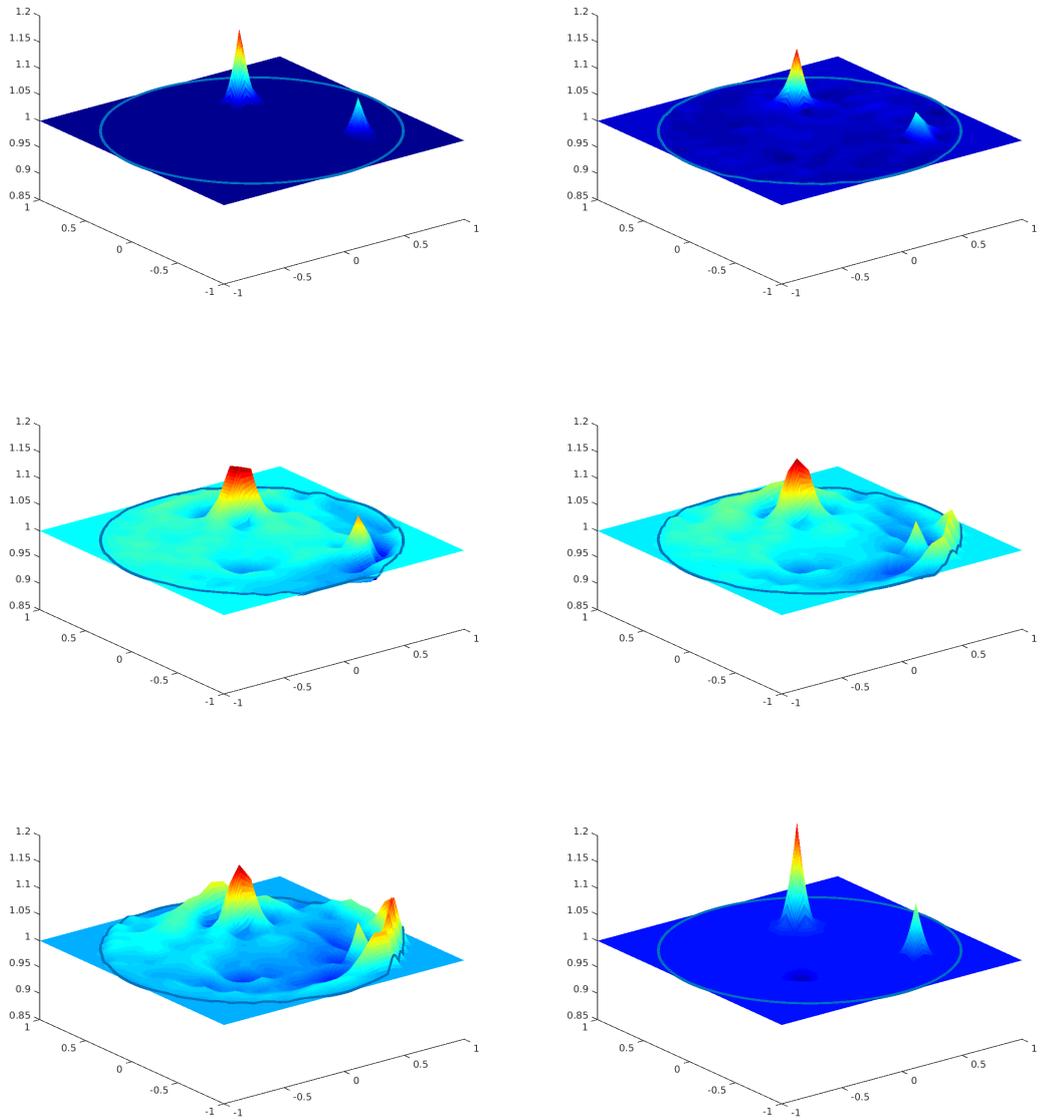


Abbildung 4.7.: Schallgeschwindigkeiten aus der Vogelperspektive. *Oben links:* $p_1 = 1$, *Oben rechts:* $p_2 = 1.1$, *Mitte links:* $p_3 = 2$, *Mitte rechts:* $p_4 = 4$, *Unten links:* $p_5 = 10$, *Unten rechts:* zu rekonstruierende Schallgeschwindigkeit.

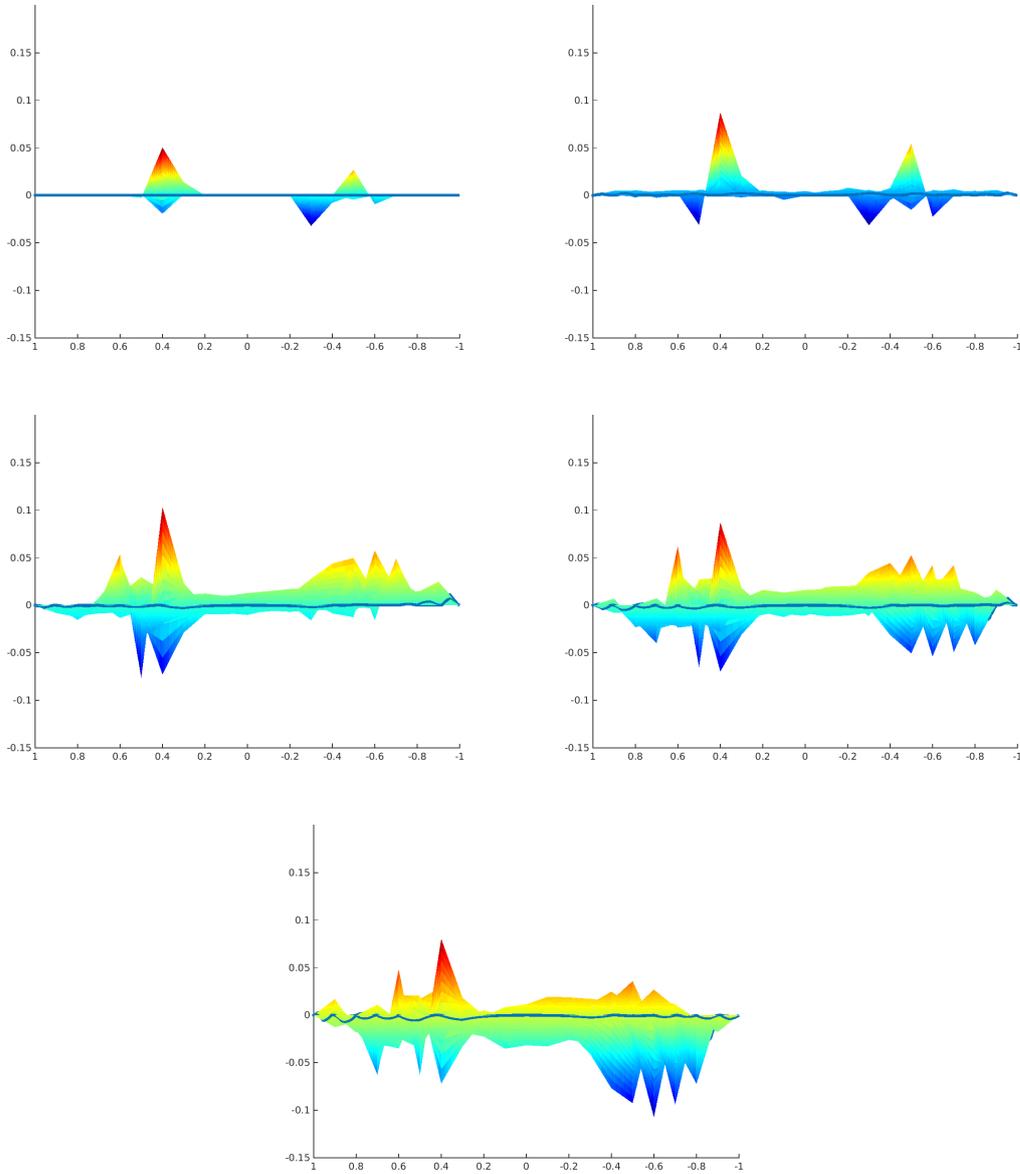


Abbildung 4.8.: Differenz der originalen und rekonstruierten Schallgeschwindigkeiten von der Seite. *Oben links:* $p_1 = 1$, *Oben rechts:* $p_2 = 1.1$, *Mitte links:* $p_3 = 2$, *Mitte rechts:* $p_4 = 4$, *Unten:* $p_5 = 10$.

des adjungierten Operators R_k^* dauert etwa 20 Sekunden. Ein Satz Geodäten benötigt dabei etwa 12 MB Speicher. Es wurde in jedem Iterationsschritt genau ein Landweber-Schritt durchgeführt mit Dämpfungsparameter $\mu = 0.05$. Das Gebiet wurde mit dem Parameter $n = 10$ diskretisiert, so dass insgesamt 346 Freiheitsgrade vorlagen.

Auch hier wurde wieder anhand der Zielfunktionale $J_\alpha^{n,N,k}(\tilde{n}_{k+1})$ optimale k_i^* , $i = 1, \dots, 5$, gewählt. Es ergeben sich die folgenden Werte:

$$k_1^* = 42, k_2^* = 50, k_3^* = 8, k_8^* = 10 \text{ und } k_5^* = 11.$$

Für $k > k^*$ wurden die Abweichungen wieder größer. Die Rekonstruktionen sind in Abbildung 4.7 geplottet im Vergleich zur Ausgangsschallgeschwindigkeit. Man erkennt hier deutliche Unterschiede in den Ergebnissen. Wie zu erwarten war, liefern die 1- und 1.1-Normen die besten Ergebnisse. Insbesondere bleibt hier der überwiegende Teil der Schallgeschwindigkeit konstant. Aber schon die 2-Norm erzeugt kleine Schwankungen, wo hingegen die 4- und 10-Normen sehr große Schwankungen im Gebiet liefern.

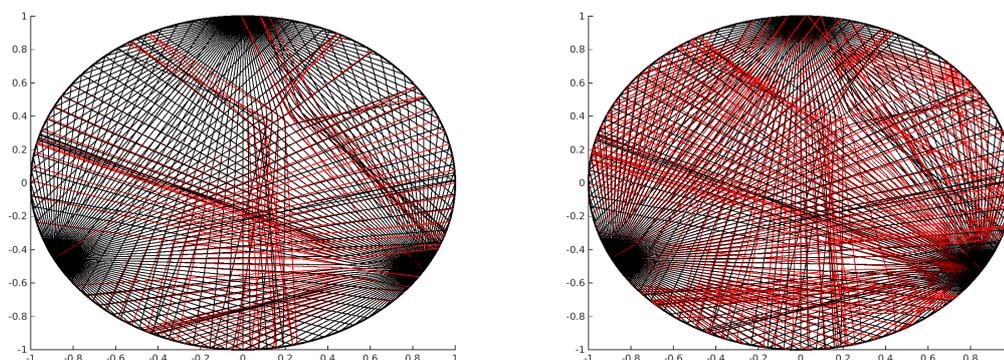


Abbildung 4.9.: Verlauf der Geodäten. Die schwarzen Kurven entsprechen den originalen Messwegen, die roten sind die Rekonstruktionen. *Links:* $p_1 = 1$, $k^* = 42$
Rechts: $p_5 = 10$, $k^* = 11$.

Dies erkennt man insbesondere an Abbildung 4.8 bei dem die Differenz der originalen und jeweils rekonstruierten Schallgeschwindigkeit von der Seite geplottet wurde. Die Peaks werden bei allen Parametern zwar gut detektiert und nachgebildet (maximaler Fehler liegt bei 0.1), jedoch erkennt man bei höherer Norm stärkere Abweichungen von der euklidischen Metrik. Insbesondere sind die Fehler teilweise genauso groß wie die korrekt detektierten Peaks.

Die zum Teil sehr großen Abweichungen, insbesondere im Randbereich spiegeln sich auch in dem Verlauf der Geodäten wieder. In Abbildung 4.9 sind die originalen Messwege, anhand derer die Laufzeitdaten aufgenommen wurden, in schwarz geplottet. Darunter wurden in rot die zum jeweiligen Zeitpunkt k^* zur Rekonstruktion berechneten Geodäten abgebildet. Die linke Grafik zeigt den Fall $p_1 = 1$ und die rechte $p_5 = 10$. Im Falle $p_1 = 1$ sind fast alle roten Geodäten von den schwarzen überdeckt, das heißt, dass Messdaten und Rekonstruktion zusammen fallen. Im anderen Extremfall $p_5 = 10$ nehmen die Geodäten

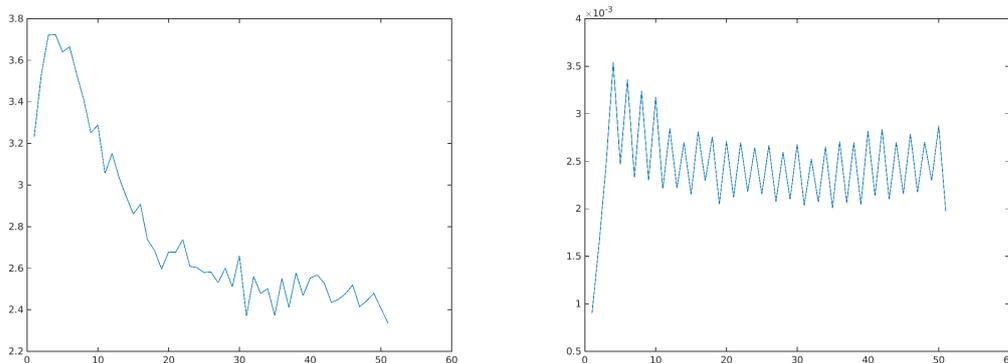


Abbildung 4.10.: *Links*: Wert des Zielfunktional $J_\alpha^{n,N,k}(\tilde{n})$ in Abhängigkeit zum Iterationsschritt k , *Rechts*: Fehler in den Knoten \mathcal{K}_{k^*} in Abhängigkeit zum Iterationsschritt k .

zur Rekonstruktion zum Teil erheblich andere Wege.

Bei der Rekonstruktion mit Straftermen $p_2 = 1.1$ tritt auch ein interessanter Effekt auf, den wir kurz erläutern wollen. In Abbildung 4.10 ist zum einen der Wert des Zielfunktional in Abhängigkeit des Iterationsschritts geplottet und zum anderen der Knotenfehler

$$\mathcal{K}_k = \left(\sum_{(r,s) \in \tilde{E}^n} |c_k(r,s) - c(r,s)|^{1.1} \right)^{\frac{1}{1.1}},$$

also die Differenz von Rekonstruktion c_k und Ausgangsschallgeschwindigkeit c , gemessen in der 1.1-Norm. Wie man sehr schön erkennen kann, liegt ein tendenzieller Abstieg in den linearisierten Tikhonov-Funktionalen vor. Auf der anderen Seite korreliert das nicht mit dem Abstand zwischen Rekonstruktion und Original. Dort ist eher eine Oszillation zu erkennen. Der Grund für dieses Verhalten wird wahrscheinlich daran liegen, dass wir die Abstiegsschrittweite μ_k nicht adaptiv anpassen. So wird das lokale Minimum wahrscheinlich übersprungen und im nächsten Schritt wieder korrigiert, was zu den Schwankungen im Knotenfehler führt. Auf der anderen Seite führt das Verfahren insgesamt zu einer besseren Approximation der zur Rekonstruktion gehörenden Geodäten an die Messgeodäten, so dass die Funktionale insgesamt kleiner werden.

Als nächstes betrachten wir das Beispiel “konstante positive Krümmung”, wobei die Schallgeschwindigkeit gegeben ist über

$$c(x) = 1 + \varphi_\pm(x)\chi_M(x)$$

mit

$$\varphi_+(x) = \frac{(R^2 - a^2\|x\|^2)^2 - (R^2 - a^2)^2}{4R^2}$$

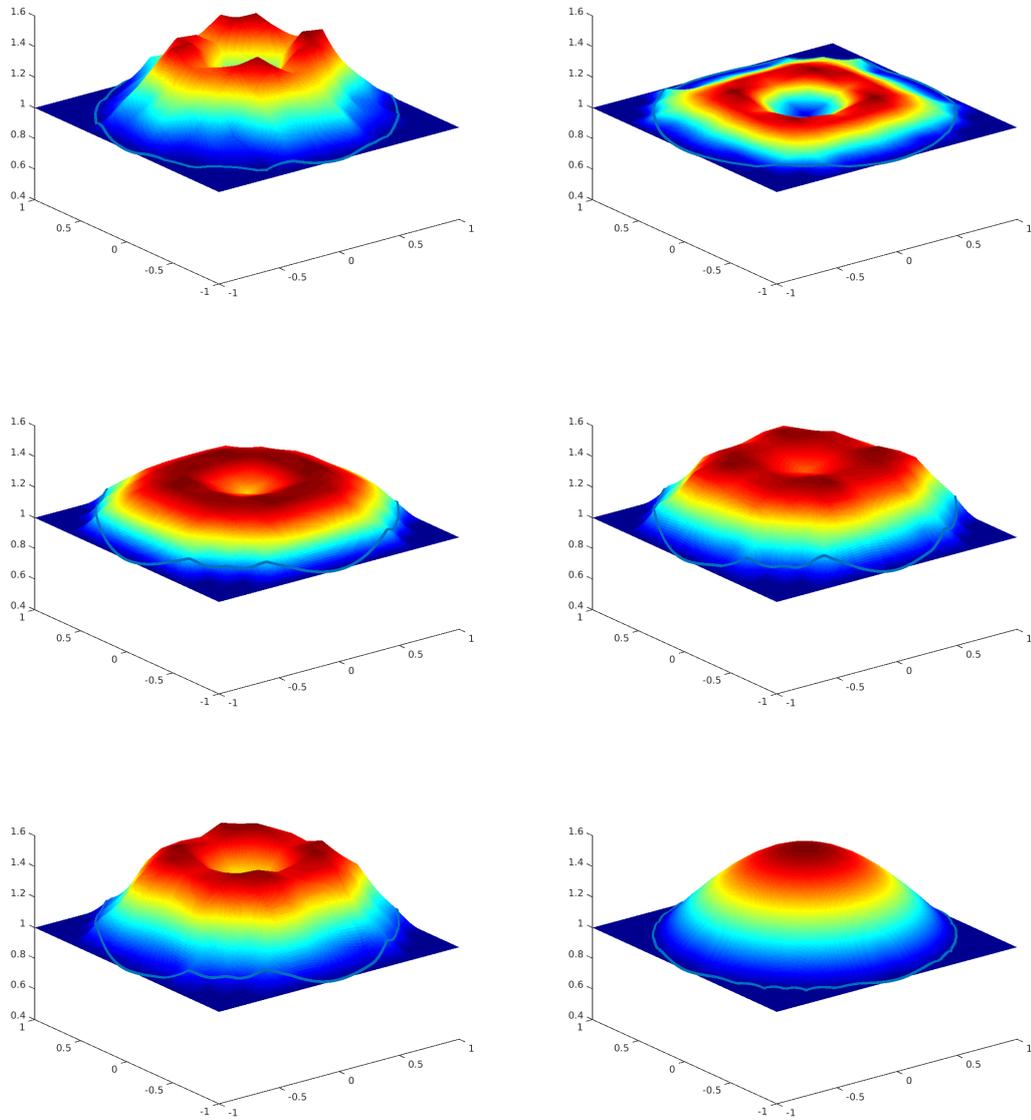


Abbildung 4.11.: Schallgeschwindigkeiten aus der Vogelperspektive. *Oben links:* $p_1 = 1$, *Oben rechts:* $p_2 = 1.1$, *Mitte links:* $p_3 = 2$, *Mitte rechts:* $p_4 = 4$, *Unten links:* $p_5 = 10$, *Unten rechts:* zu rekonstruierende Schallgeschwindigkeit.

mit Parametern $a = 1.2$ und $R = 2$. Es wurden wieder die Normen

$$p_1 = 1, p_2 = 1.1, p_3 = 2, p_4 = 4 \text{ und } p_5 = 10$$

untersucht. Im Falle $p_1 = 1$ wurde der Threshold-Algorithmus 4.5 benutzt, so dass hier der Regularisierungsparameter $\alpha = 0.01$, in den anderen Fällen jedoch $\alpha = 0.4$ gewählt wurde. Pro Iterationsschritt wählen wir $N_x = 48$ Aufpunkte und $N_\xi = 48$ Richtungen pro Aufpunkt, so dass wir in jedem Schritt $|\bar{\Gamma}_N| = |\bar{\Gamma}_{N_x, N_\xi, k}^\beta| = 2304$ Geodäten mit dem Parameter $\beta = (0.9, 0.05, 10^{-3})$ berechnen. Die Erstellung eines vollen Datensatzes Γ_N und die Auswertung des adjungierten Operators R_k^* dauern jeweils circa sechs Sekunden. Ein Satz Geodäten benötigt dabei etwa zwei MB Speicher. Es wurde in jedem Iterationsschritt genau ein Landweber-Schritt durchgeführt mit Dämpfungsparameter $\mu = 0.05$. Das Gebiet wurde mit dem Parameter $n = 6$ diskretisiert, so dass insgesamt 126 Freiheitsgrade vorlagen.

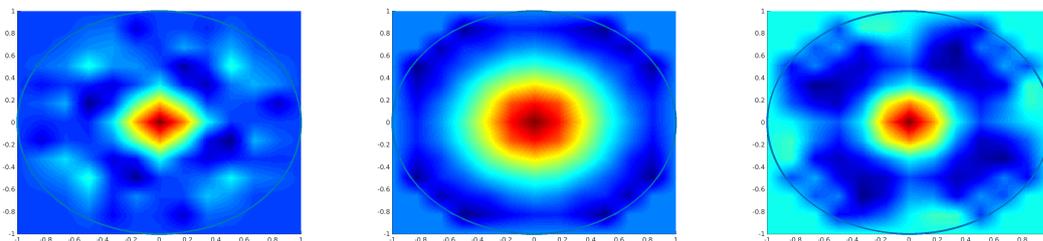


Abbildung 4.12.: Differenz der originalen und rekonstruierten Schallgeschwindigkeiten als Draufsicht. *Links:* $p_1 = 1$, *Mitte:* $p_3 = 2$, *Rechts:* $p_5 = 10$.

Auch hier wurde wieder anhand der Zielfunktionale $J_\alpha^{n,N,k}(\tilde{n}_{k+1})$ die optimalen k_i^* , $i = 1, \dots, 5$, gewählt. Es ergeben sich die folgenden Werte:

$$k_1^* = 50, k_2^* = 10, k_3^* = 20, k_4^* = 28 \text{ und } k_5^* = 20.$$

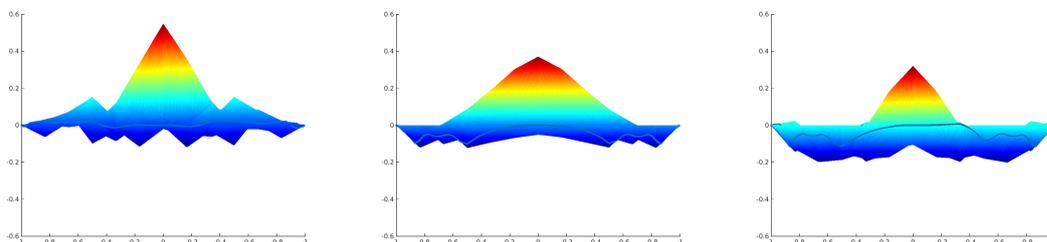


Abbildung 4.13.: Differenz der originalen und rekonstruierten Schallgeschwindigkeiten von der Seite. *Links:* $p_1 = 1$, *Mitte:* $p_3 = 2$, *Rechts:* $p_5 = 10$.

In Abbildung 4.11 sind die Rekonstruktionen zusammen mit der originalen Schallgeschwindigkeit geplottet. Im Vergleich zum vorherigen Beispiel “Peaks” ergibt sich hier ein etwas anderes Bild bezüglich der p -Normen im Strafterm. Eine höhere Norm lässt insgesamt eine stärkere Abweichung im gesamten Gebiet zu. Man erkennt deutlich, dass sich hier das Problem der “cold spots” in allen Rekonstruktionen auswirkt. Die Schallgeschwindigkeit im Ursprung wird nicht sehr gut rekonstruiert, am besten noch im Falle $p_5 = 10$.

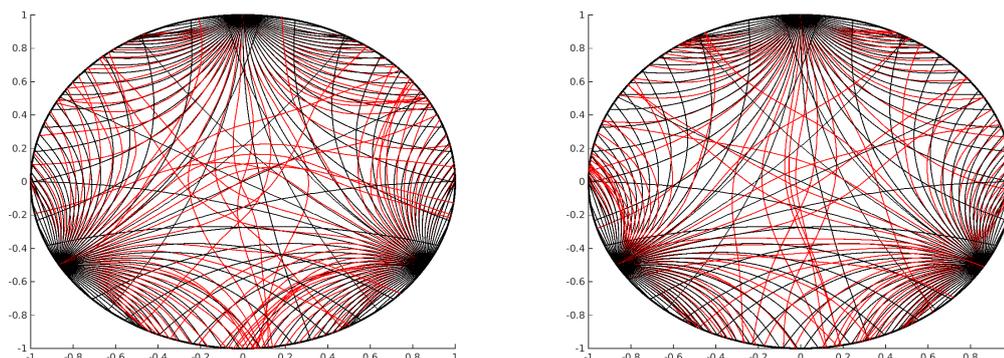


Abbildung 4.14.: Verlauf der Geodäten. Die schwarzen Kurven entsprechen den originalen Messwegen, die roten sind die Rekonstruktionen. *Links:* $p_1 = 1$, $k_1^* = 50$ *Rechts:* $p_5 = 10$, $k_5^* = 20$.

Dies erkennt man sehr gut an den Fehlerplots 4.12 und 4.13. Hier werden die Rekonstruktionen c_i für $p_1 = 1$, $p_3 = 2$ und $p_5 = 10$ als Draufsicht, beziehungsweise von der Seite, in Differenz zur Ausgangsschallgeschwindigkeit c gezeigt. Man erkennt deutlich, dass das Zentrum am besten für p_5 rekonstruiert wird, obwohl hier nach wie vor die Auswirkungen des “cold spots” zu erkennen sind. Allerdings wird dafür die Schallgeschwindigkeit in unmittelbarer Nähe zum Ursprung zu hoch angesetzt. Bei der Norm $p_3 = 2$ werden die Randgebiete im Vergleich deutlich besser rekonstruiert als bei den anderen Fällen. Bei der Norm $p_1 = 1$ wird der Ursprung nahezu als Eins rekonstruiert, außerdem sind starke Schwankungen in den Randbereichen zu erkennen.

In Abbildung 4.14 sind einige originale und rekonstruierte Geodäten (rote Kurven) geplottet. Im Vergleich erkennt man, dass die Geodäten im Fall p_1 , insbesondere im Zentrum, sehr starke Krümmungen aufweisen. Die rekonstruierten Geodäten im Fall p_5 folgen gerade im Bereich des Ursprungs schon deutlich besser dem Verlauf der Messgeodäten (schwarze Kurven).

Beispiel 4.4.3 (Rauschen in den Messdaten)

Wir wollen nun den Algorithmus dahingehend untersuchen, wie sich ein zufälliges, gleichverteiltes Rauschen auf den Messdaten auf die Rekonstruktion auswirkt. Dazu bestimmen wir zunächst anhand des Beispiels “Welle mit Peaks” mit der in Beispiel 3.1.11 aus Kapitel 3 angegebenen Schallgeschwindigkeit und den zugehörigen Parametern sehr genaue

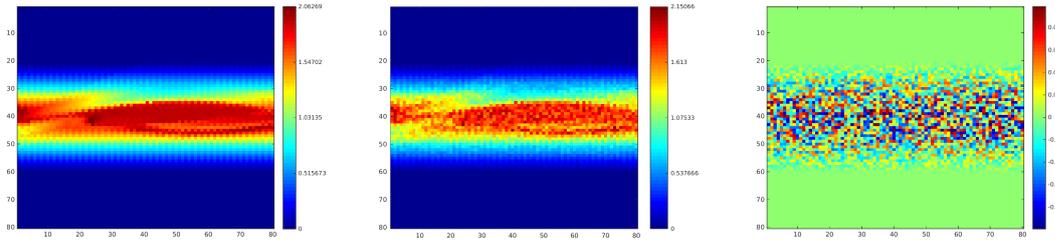


Abbildung 4.15.: Sinogramme zum Experiment “Welle mit Peaks” für $N_x = N_\xi = 80$.
Links: bei exakten Messdaten, *Mitte*: mit 10%igem Rauschen, *Rechts*:
 Differenz der exakten und gestörten Daten.

Messdaten. Die Schallgeschwindigkeit ist also gegeben über

$$c(x) = 1 + \sum_{i=1}^3 \varphi_i(x) \chi_i(x) + \bar{\varphi}(x) \bar{\chi}(x).$$

Anschließend stören wir die Messdaten durch zufällig erzeugte, gleichmäßig verteilte Zufallszahlen im Bereich -10% bis $+10\%$. Die erzeugten Sinogramme und deren Differenz sind in Abbildung 4.15 zu sehen.

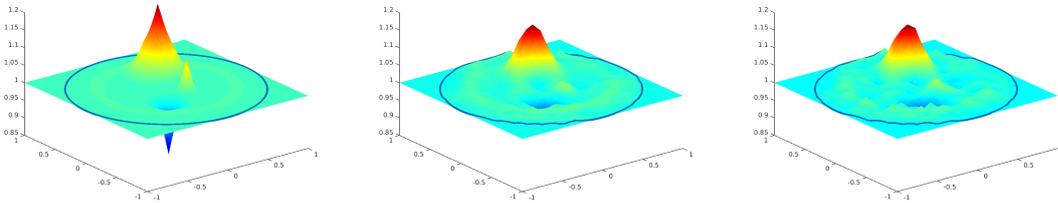


Abbildung 4.16.: Schallgeschwindigkeiten aus der Vogelperspektive. *Links*: Originale Schallgeschwindigkeit c , *Mitte*: Rekonstruierte Schallgeschwindigkeit c_{50} ohne Rauschen, *Rechts*: Rekonstruierte Schallgeschwindigkeit \tilde{c}_{50} .

Die Rekonstruktionsparameter sind $\mu = 0.02$, $\alpha = 1.5$ und $p = 2$. Die Anzahl der berechneten Geodäten ist $80 \cdot 80 = 6400$ pro Iterationsschritt und die Auflösung des Gebiets ist durch $n = 10$ vorgegeben. Wir berechnen einmal eine Rekonstruktion c_k ohne Störung und eine Rekonstruktion \tilde{c}_k mit Störung der Messdaten für jeweils $k = 50$.

Abbildung 4.16 zeigt dabei links das Original, in der Mitte die Rekonstruktion mit Rauschen und rechts die ohne Rauschen nach 50 Iterationsschritten. Man erkennt, dass trotz Messfehlern im Bereich von 20% die Peaks sehr gut detektiert werden. Allerdings wird die Welle nur schlecht rekonstruiert, was man sehr gut an Abbildung 4.17 sieht. Hier hat das Rauschen einen zu großen Einfluss auf die Rekonstruktion. Der grünliche Ring, der die Peaks umgibt, wird, im Gegensatz zur Rekonstruktion ohne Messfehlern, nur bruchstückhaft detektiert.

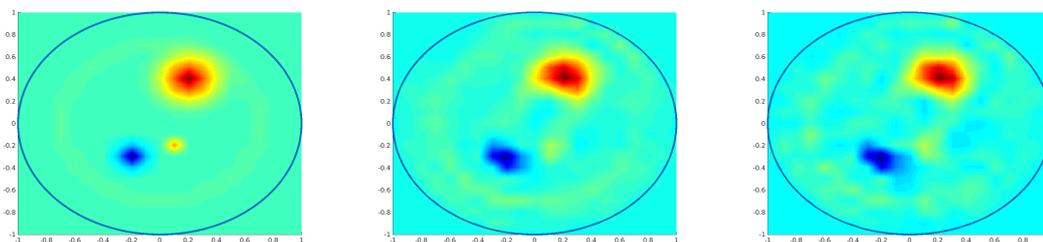


Abbildung 4.17.: Schallgeschwindigkeiten aus der Draufsicht. *Links*: Originale Schallgeschwindigkeit c , *Mitte*: Rekonstruierte Schallgeschwindigkeit c_{50} ohne Rauschen, *Rechts*: Rekonstruierte Schallgeschwindigkeit \tilde{c}_{50} .

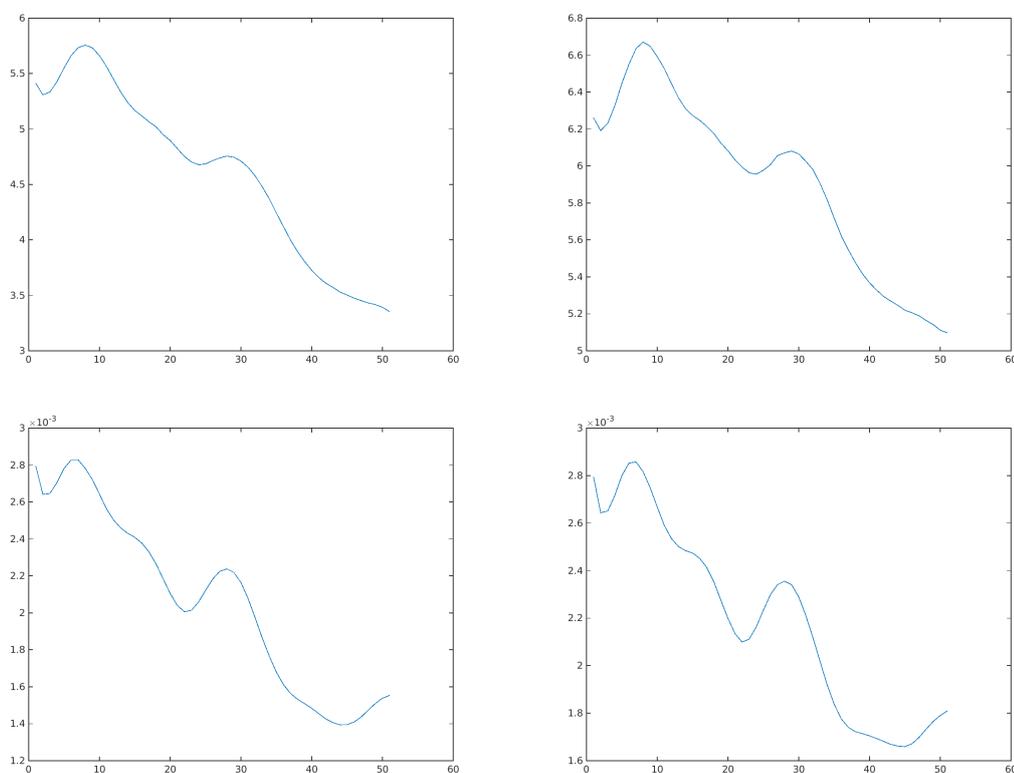


Abbildung 4.18.: Konvergenz des Funktionals beziehungsweise des Knotenfehlers, geplottet gegen den Iterationsschritt. *Oben links*: Funktional bei exakten Messdaten, *Oben rechts*: Funktional bei gestörten Messdaten, *Unten links*: Knotenwerte bei exakten Messdaten, *Unten rechts*: Knotenwerte bei gestörten Messdaten.

Trotzdem liegt insgesamt bei beiden Messdatensätzen eine Konvergenz, sowohl bezüglich des Funktionals $J_\alpha^{n,N,k}$, als auch bezüglich der akkumulierten Knotenabweichung \mathcal{K}_k , vor (vergleiche Abbildung 4.18). Der Fehler in den Knotenwerten ist bei beiden Datensätzen zu Beginn natürlich gleich. Bei exakten Messdaten stellt sich aber eine viel schnellere Konvergenz ein, als bei den gestörten Daten. Etwas anderes wäre auch nicht zu erwarten gewesen, da das Iterationsverfahren gegen eine Schallgeschwindigkeit konvergieren sollte, die die gestörten Daten erzeugt, und nicht die exakten Messdaten.

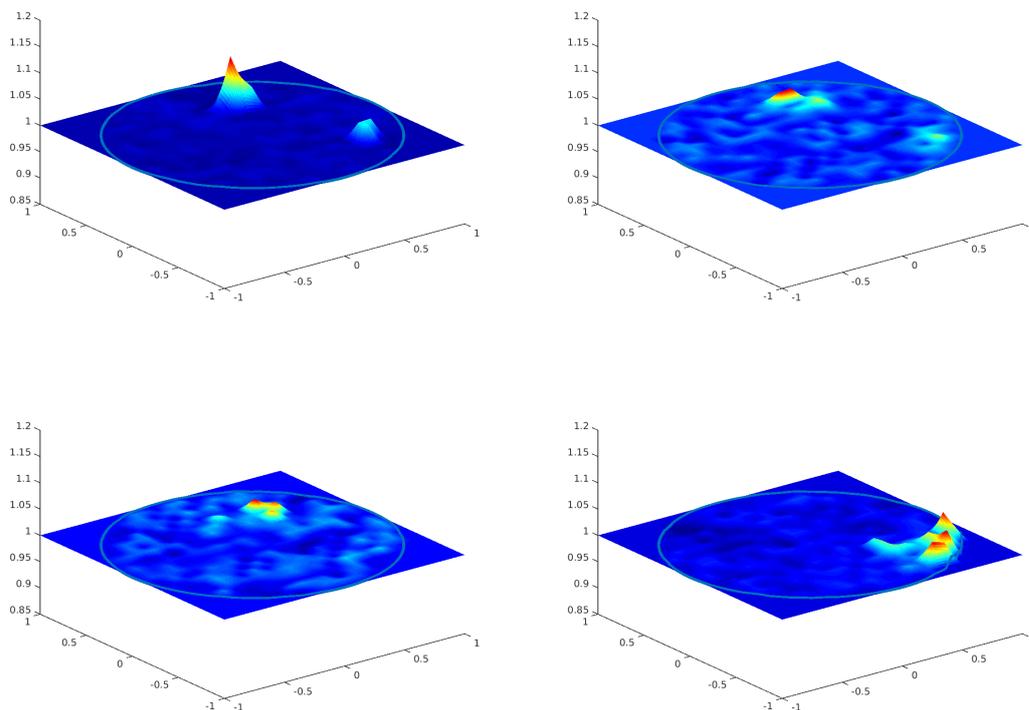


Abbildung 4.19.: Rekonstruierte Schallgeschwindigkeiten bei wenig Messdaten aus der Vogelperspektive. *Oben links*: Datensatz 1 mit $N_x = N_\xi = 60$, *Oben rechts*: Datensatz 2 mit $N_x = N_\xi = 30$, *Unten links*: Datensatz 3 mit $N_x = N_\xi = 20$, *Unten rechts*: Datensatz 4 mit $N_x = N_\xi = 12$.

Beispiel 4.4.4 (Wenig Daten)

Als nächstes Beispiel betrachten wir die Auswirkungen von wenig Messdaten auf die Rekonstruktionen. Wir betrachten hierfür das Beispiel 3.1.8 “Peaks” aus Kapitel 3 mit der dort angegebenen Schallgeschwindigkeit und Parametern.

Gerechnet haben wir zu den Parametern $p = 1.1$ für die Norm, $\alpha = 0.2$ als Regularisierungsparameter und $\mu = 0.03$ als Abstiegschrittweite. Das Gebiet wurde mit $n = 10$ diskretisiert und es wurden für vier Datensätze

1. $N_x = N_\xi = 60$,
2. $N_x = N_\xi = 30$,
3. $N_x = N_\xi = 20$ und
4. $N_x = N_\xi = 15$

Rekonstruktionen erstellt. Bei jeder Rechnung wurden $k = 20$ Iterationsschritte durchgeführt.

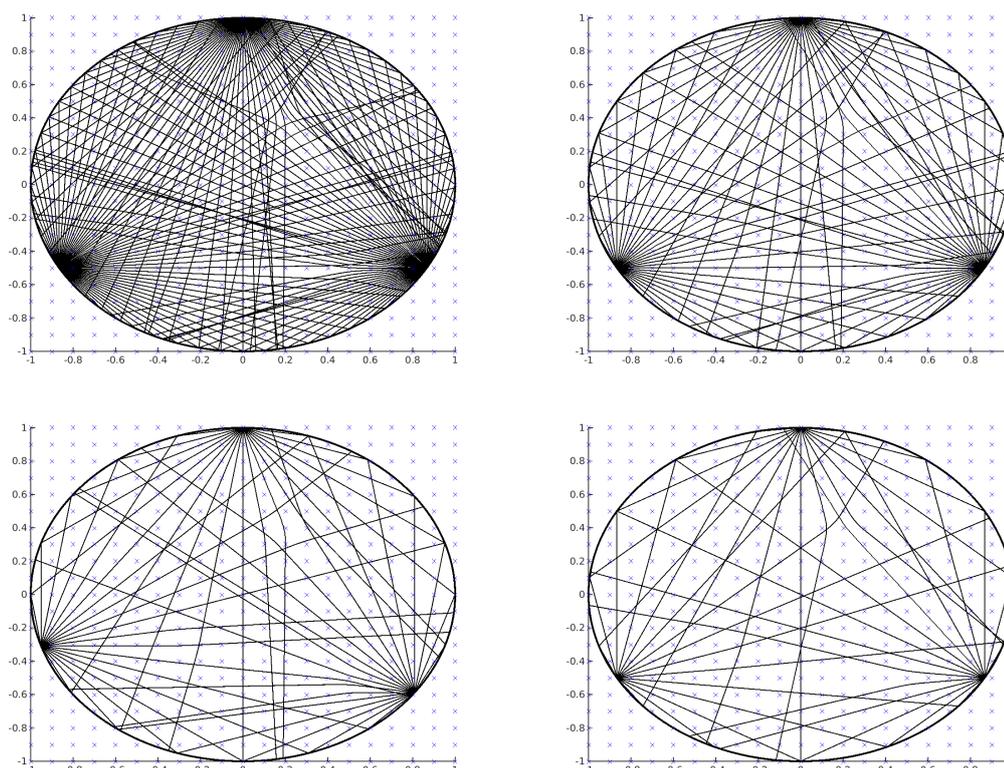


Abbildung 4.20.: Plot der von den drei Punkten $x_1 = (0, 1)^T$, $x_2 = \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)^T$ und $x_3 = \left(-\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)^T$ ausgehenden Geodäten, welche die Messdaten erzeugen (schwarz), und Rekonstruktionpunkte (blau). *Oben links*: Datensatz 1 mit $N_x = N_\xi = 60$, *Oben rechts*: Datensatz 2 mit $N_x = N_\xi = 30$, *Unten links*: Datensatz 3 mit $N_x = N_\xi = 20$, *Unten rechts*: Datensatz 4 mit $N_x = N_\xi = 12$.

Das Ergebnis dieser Rechnung ist in Abbildung 4.19 zu sehen. Dabei ist zu erkennen, dass bei einem Datensatz von $60 \cdot 60 = 3600$ Laufzeiten (Datensatz 1), die beiden positiven Peaks gut lokalisiert werden. Bei der Hälfte der Daten (Datensatz 2) lässt sich der kleinere Ausschlag nur noch erahnen. Wird die Anzahl der Messdaten weiter reduziert auf

$30 \cdot 30 = 900$ Laufzeiten (Datensatz 3), so wird nur noch der höchste Peak lokalisiert, aber schon in seiner Quantität nicht mehr gut angenähert, wo hingegen der zweite Peak bei $(0.5, -0.5)^T$ nicht mehr detektiert wird. Bei noch weniger Messdaten (Datensatz 4) hat die Rekonstruktion keine Gemeinsamkeiten mehr mit der Ausgangsschallgeschwindigkeit.

Anzahl der Messdaten	3600	900	400	225
Rechenzeit für die Geodäten	11 Sek	3 Sek	1,5 Sek	1,2 Sek
Rechenzeit zur Auswertung von R_k^*	25 Sek	7 Sek	4 Sek	2 Sek
Größe eines Datensatzes	4,6 MB	1,2 MB	0,5 MB	0,3 MB
Speicherbedarf für R_k^*	3,45 kB	3,45 kB	3,45 kB	3,45 kB
Gesamtlaufzeit	13,6 Min	4,54 Min	2,83 Min	2,01 Min

Tabelle 4.1.: Ungefähre Rechenzeiten und Speicherbedarf in einem Iterationsschritt für verschieden große Datensätze.

Denn aufgrund der geringen Anzahl an Messdaten können auch nur wenig Geodäten für die Rekonstruktion erzeugt werden, so dass der adjungierte Operator in einem Iterationsschritt gemäß Algorithmus 4.3.4 nicht mehr hinreichend genau ausgewertet werden kann. Kommen dazu noch Beugungseffekte, so kann man davon ausgehen, dass es viele Bereiche gibt, durch die keine Geodäten verlaufen. Somit müssen zur Rekonstruktion in Punkten in diesen Bereichen die Laufzeiten von weit entfernten Geodäten in die Auswertung von R_k^* einfließen (vergleiche Abbildung 4.20, zur besseren Übersicht wurden nur Geodäten, von jeweils drei Aufpunkten ausgehend, dargestellt).

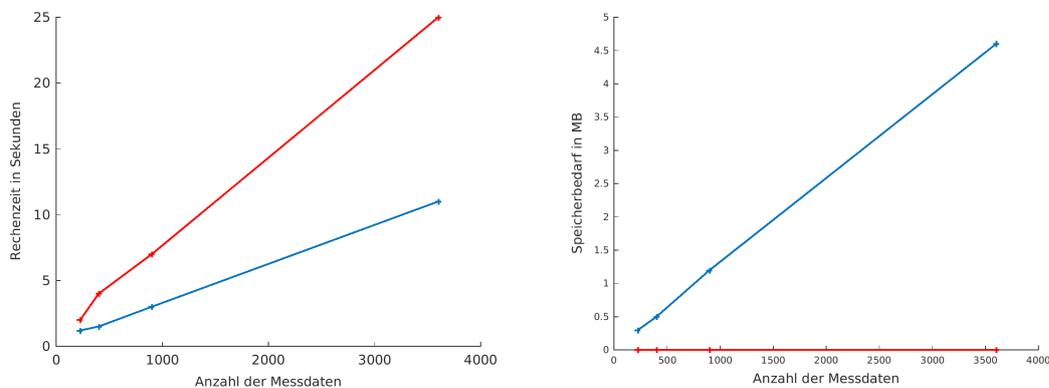


Abbildung 4.21.: Ungefähre Rechenzeiten (*links*) und Speicherbedarf (*rechts*) in einem Iterationsschritt für verschieden große Datensätze. Die rote Kurve bezieht sich auf die benötigte Rechenzeit / den benötigten Speicherbedarf zur Erzeugung eines Datensatzes an Geodäten, die blaue bezieht sich entsprechend auf eine einmalige Auswertung / Speicherung des adjungierten Operators.

Vergleicht man die Rechenzeiten für die Rekonstruktionen (siehe Tabelle 4.1), so sieht

man, dass sowohl die Zeiten für die Auswertung des adjungierten Operators, als auch für die Berechnung eines vollen Datensatzes an Geodäten, nahezu linear steigt. Das gleiche gilt für die Gesamtrechenzeit und für den Speicherbedarf für die Geodäten. Da die Auflösung n des Gebiets konstant ist, gilt dies auch für den Speicherbedarf des adjungierten Operators. Dieser Zusammenhang wurde in Abbildung 4.21 grafisch dargestellt.

Insgesamt muss man also abwägen zwischen der gewünschten Genauigkeit der Rekonstruktion und der Rechenzeit, die für diese in Kauf genommen werden muss.

Beispiel 4.4.5 (Erhöhung der inneren Iterationsschritte)

Wir wollen nun die Auswirkungen der Anzahl der inneren Iterationen auf die Konvergenz des Verfahrens untersuchen. Wie bereits in Bemerkung 4.1.2 angemerkt, stellt sich die Frage wie viele Landweber-Schritte in Algorithmus 4.1.1 sinnvoll sind. Klar sollte sein, dass zu viele Schritte zu einer Divergenz führen können.

Wir haben für Beispiel “Peaks” mit den Parametern $p = 2$ für die Norm, $\alpha = 0.5$ als Regularisierungsparameter und $\mu = 0.03$ als Abstiegsschrittweite gerechnet. Das Gebiet wurde mit $n = 10$ diskretisiert und es wurde ein Datensatz von $60 \times 60 = 3600$ Geodäten für die Rekonstruktion zu Grunde gelegt.

Nun wurde allerdings nicht nur eine Landweber-Iteration in Algorithmus 4.1.1 durchgeführt, sondern jeweils $l_{max}^1 = 2$ oder $l_{max}^2 = 5$. Das heißt, zu einer gegebenen, berechneten Schallgeschwindigkeit c_k mit zugehörigem Datensatz Geodäten, wurde die Schallgeschwindigkeit bezüglich des linearisierten Tikhonov-Funktional $c_{k+1} := \arg \min J_{\alpha}^{c_k}(c)$ optimiert.

Interessant sind die Rekonstruktionen der Schallgeschwindigkeit für die Iterationsschritte $k = 3$ bis $k = 5$, welche im Vergleich in Abbildung 4.22 dargestellt sind. Im Iterationsschritt $k = 3$ sind die Peaks bei fünf Landweber-Schritten quantitativ besser detektiert, als bei nur zwei inneren Iterationen. Jedoch sieht man bei den (äußeren) Iterationsschritten $k = 4$ und $k = 5$ bereits, dass das Verfahren für $l_{max}^2 = 5$ divergiert, wohingegen im Fall $l_{max}^1 = 2$ eine Konvergenz zu erwarten ist.

In den Abbildungen 4.23 und 4.24 ist dazu der Verlauf einiger, im entsprechenden k -ten Iterationsschritt berechneter, Geodäten geplottet. Man sieht, dass bereits im dritten Iterationsschritt im Fall $l_{max}^2 = 5$ (Abbildung 4.24) eine starke Abweichung der Geodäten von Geraden vorliegt. Diese Abweichung verstärkt sich dann noch. Insbesondere ändert sich der Verlauf der Geodäten im ersten und vierten Quadranten beim Übergang von $k = 4$ auf $k = 5$ sehr stark. Diese große Änderung basiert auf einer stärkeren Optimierung der Schallgeschwindigkeit im k -ten Schritt bezüglich der Wege im $(k - 1)$ -ten Schritt durch vergleichsweise viele Landweber-Schritte. Diese “falschen” Messwege führen dann im nächsten Schritt zu einer Verschlechterung der Approximation der Schallgeschwindigkeit. Im Gegensatz dazu bleiben die Wege bei wenigen Landweber-Schritten (Abbildung 4.23) relativ stabil.

Davon abgesehen erfordert jeder weitere Landweber-Schritt eine Auswertung des adjungierten Operators. Eine Erhöhung der Anzahl der Landweber-Schritte ist also nur sinnvoll, wenn die Auswertung des adjungierten Operators im Verhältnis zur Berechnung der Geodäten relativ billig ist (zum Beispiel bei grober Diskretisierung des Gebiets, also kleinem n , und großer Anzahl an Messdaten, also großem N_x bzw. N_{ξ}).

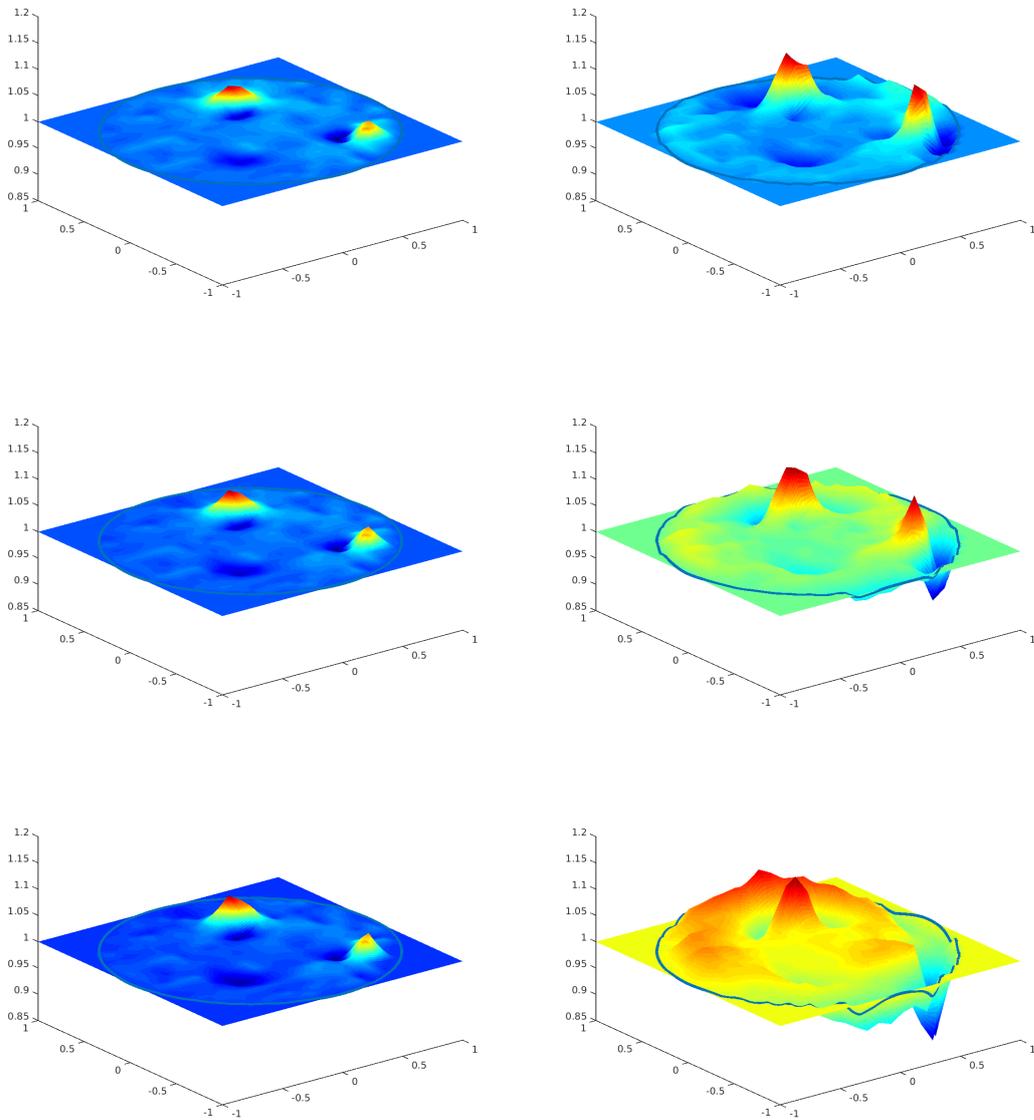


Abbildung 4.22.: Rekonstruierte Schallgeschwindigkeiten bei zwei (*links*) und fünf (*rechts*) inneren Iterationen. *Oben*: $k = 3$, *Mitte*: $k = 4$, *Unten*: $k = 5$.

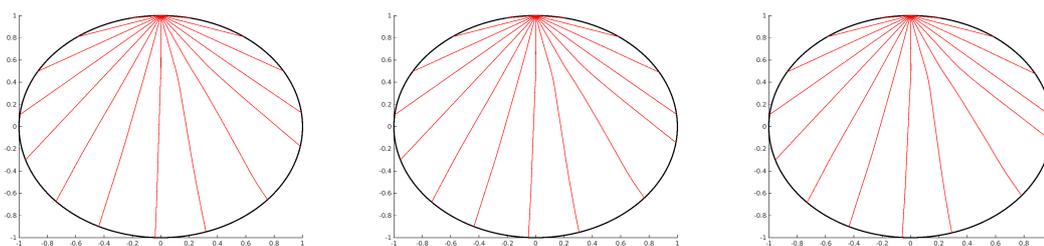


Abbildung 4.23.: Approximierte Geodäten bei zwei inneren Iterationen. *Links: $k = 3$, Mitte: $k = 4$, Rechts: $k = 5$.*

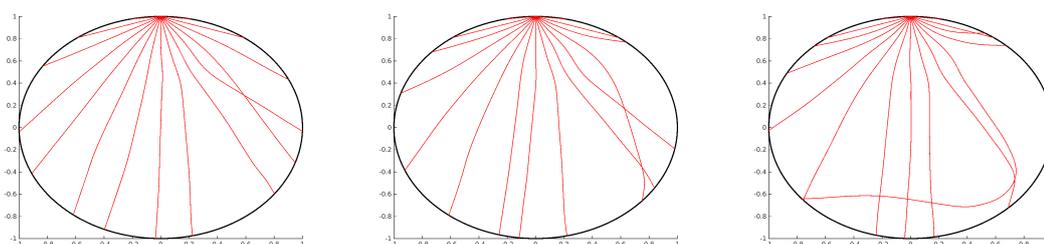


Abbildung 4.24.: Approximierte Geodäten bei fünf inneren Iterationen. *Links: $k = 3$, Mitte: $k = 4$, Rechts: $k = 5$.*

Beispiel 4.4.6 (Adaptive Anpassung der Schrittweite)

Als letztes Beispiel führen wir noch eine kleine Verbesserung im Algorithmus ein und testen diese. Unter anderem haben wir im vorhergehenden Beispiel gesehen, dass eine zu starke Anpassung der Schallgeschwindigkeit im k -ten Schritt zur Divergenz des Verfahrens führen kann, aufgrund der dann “fehlerhaften” Neuberechnung der Geodäten. Diesem Problem kann man begegnen, indem entweder nur wenige Landweber-Schritte gemacht werden oder indem der Relaxationsparameter μ sehr klein gewählt wird. Der Nachteil einer kleinen Schrittweite liegt allerdings in einer sehr langsamen Konvergenz des Verfahrens.

Die Idee ist nun die Schrittweite adaptiv anzupassen. Dazu setzen wir zunächst eine Schrittweite $\mu_0 > 0$ fest und führen einen Iterationsschritt durch. Anschließend vergleichen wir die Tikhonov-Funktionale, denn wünschenswert wäre ein monotoner Abstieg, zumindest, wenn wir uns schon in der Nähe der Lösung befinden. Ist $\tilde{n}_{k+1}^0 = \tilde{n}_{k+1}(\mu_0)$ die neue Iterierte und gilt

$$J_\alpha^k(\tilde{n}_{k+1}^0) < J_\alpha^{k-1}(\tilde{n}_k), \quad (4.6)$$

so rechnen wir einfach weiter. Andernfalls wird in Schritt (S1d) aus Algorithmus 4.1.1 die Abstiegschrittweite verkleinert

$$\mu_1 = \epsilon \mu_0, \quad \epsilon \in (0, 1)$$

und eine neue Approximation $\tilde{n}_{k+1}^1 = \tilde{n}_{k+1}(\mu_1)$ bestimmt. Anschließend kann wieder mittels (4.6) überprüft werden, ob ein Abstieg vorliegt. Falls nicht, so wird wieder

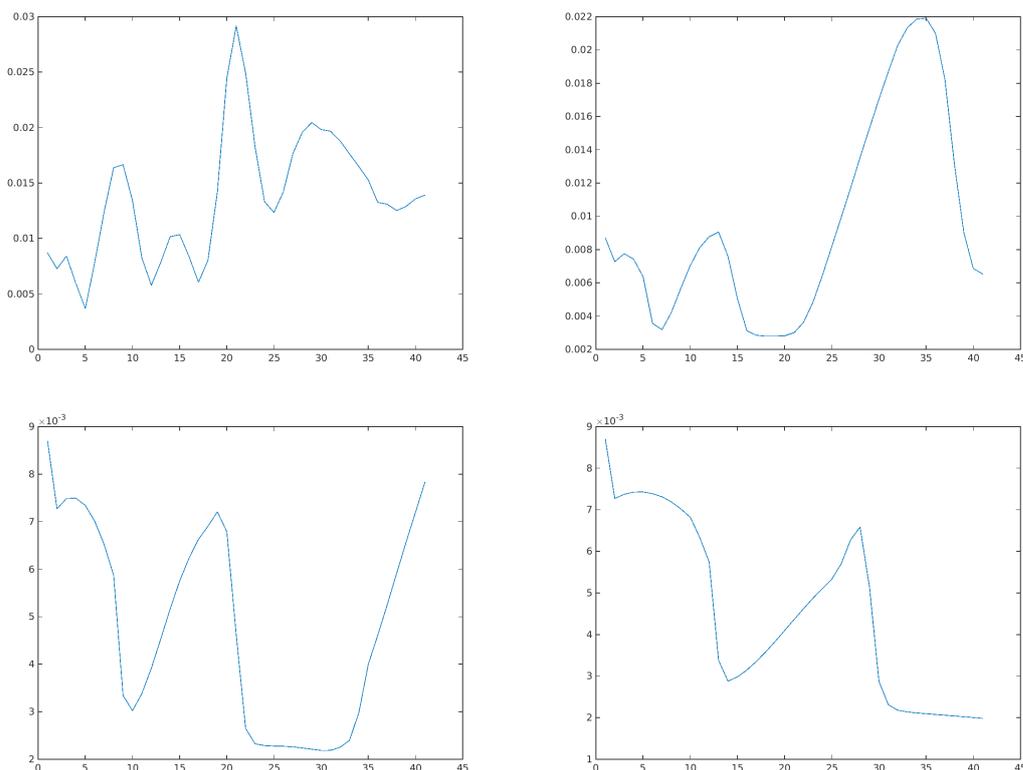


Abbildung 4.25.: Konvergenz des Knotenfehlers, geplottet gegen den Iterationsschritt, zu verschiedenen maximalen Verkleinerungen ν_i , $i = 0, 1, 2, 3$. *Oben links:* $i = 0$, *Oben rechts:* $i = 1$, *Unten links:* $i = 2$, *Unten rechts:* $i = 3$.

verkleinert $\mu_2 = \epsilon\mu_1 = \epsilon^2\mu_0$ und so weiter bis entweder Ungleichung 4.6 erfüllt wird oder eine maximale Verkleinerungsstufe $\nu \in \mathbb{N}$ erreicht wird.

Denn wie wir an den vorhergehenden Beispielen gesehen haben, ist es oftmals in der Praxis so, dass zunächst keine Verringerung des Tikhonov-Funktional vorliegt und erst später in den Konvergenzbereich vorgestoßen wird. Demzufolge ist es sinnvoll die Anzahl der Verkleinerungen von μ_0 zu beschränken, und somit gerade zu Beginn der Rekonstruktion einen Anstieg des Funktional zuzulassen, damit überhaupt Suchrichtungen gefunden werden können.

Der Vorteil der adaptiven Anpassung der Schrittweite ist, dass gegebenenfalls große Schrittweiten zugelassen werden. Der Nachteil des Verfahrens liegt allerdings in einer eventuell häufigen Neuberechnung der Geodäten. Es eignet sich also eher, wenn das Gebiet fein aufgelöst werden soll (großes n), aber nur wenige Messdaten vorliegen.

Testen wollen wir die adaptive Anpassung der Schrittweite an der Schallgeschwindigkeit aus Beispiel 3.1.9 “Konstante negative Krümmung”. Es liegen $20 \times 20 = 400$ Messdaten vor und aus diesen soll die Approximation an c für $n = 7$ berechnet werden. Als Rekonstruktionsparameter wählen wir $\alpha = 0.1$, $p = 2$ und führen $n = 40$ Iterationsschritte durch. Zum Vergleich rechnen wir mit den maximalen Verkleinerungen $\nu_i = i$ für

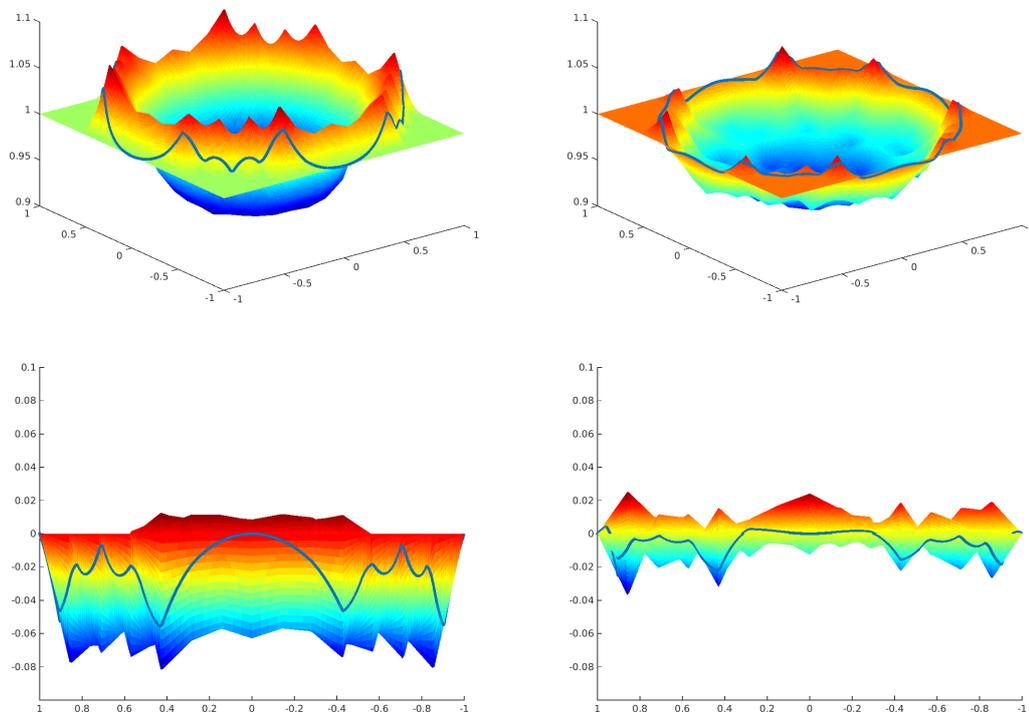


Abbildung 4.26.: Rekonstruierte Schallgeschwindigkeiten c_i und Abweichungen $c - c_i$ zur originalen Schallgeschwindigkeit c zu verschiedenen maximalen Verkleinerungen ν_i , $i = 0, 2$. *Oben links*: c_0 aus der Vogelperspektive, *Oben rechts*: c_2 aus der Vogelperspektive, *Unten links*: Abweichung $c - c_0$ von der Seite, *Unten rechts*: Abweichung $c - c_2$ von der Seite.

Maximale Verkleinerungen i	$\nu_0 = 0$	$\nu_1 = 1$	$\nu_2 = 2$	$\nu_3 = 3$
Optimaler Abbruchindex k^*	5	20	30	40
Kleinst mögliche Schrittweite μ_{ν_i}	$60 \cdot 10^{-3}$	$3.6 \cdot 10^{-3}$	$0.216 \cdot 10^{-3}$	$0.01296 \cdot 10^{-3}$
Rechenzeit bis k^*	2.43 Min	9.86 Min	15,65 Min	22.14 Min

Tabelle 4.2.: Kleinste Schrittweiten und ungefähre Rechenzeiten bis zum jeweiligen optimalen Abbruchindex k^* bei verschiedenen maximalen Anpassungen der Schrittweite.

$i = 0, 1, 2, 3$. Die Startschrittweite ist immer $\mu_0 = 0.06$.

Abbildung 4.25 zeigt den jeweiligen Knotenfehler der Rechnungen. Zunächst fällt auf, dass alle Plots ähnlich verlaufen, für größeres i jedoch in x -Richtung gestreckt wird. Dies ist auf die kleineren Schrittweiten zurückzuführen.

Je größer die maximale Verkleinerung ν_i ist, desto geringer ist die Abweichung in den Knotenwerten der Rekonstruktionen von der Ausgangsschallgeschwindigkeit c beim optimalen Abbruchindex k_i^* . Für $i = 0$ ist beispielsweise $k^* = 5$ und die Abweichung beträgt ungefähr $\mathcal{K}_5(c_0) \approx 4 \cdot 10^{-3}$. Im Fall $i = 2$ ist $k^* = 30$ und es ist $\mathcal{K}_{30}(c_2) \approx 2.2 \cdot 10^{-3}$, also deutlich kleiner. Unter anderem liegt das an einer deutlich schlechteren Rekonstruktion bei ν_0 im Randbereich als bei ν_2 (siehe Abbildung 4.26).

Die besseren Rekonstruktionen erkaufte man sich allerdings mit deutlich höheren Rechenzeiten, was man Tabelle 4.2 entnehmen kann. Auf der anderen Seite ist die adaptive Schrittweitenanpassung jedoch deutlich schneller als eine Rechnung mit sehr kleinen Schrittweiten von Beginn an. Eine Rechnung ohne Schrittweitenanpassung mit Schrittweite $\mu_2 = 0.216 \cdot 10^{-3}$ benötigt ca. 86.92 Minuten bis zum Erreichen des Minimums, wobei der optimale Abbruchindex hier $k^* = 69$ ist. Im Vergleich zur Rekonstruktion c_2 ist das mehr als dreimal so lang.

Man kann also sagen, dass eine adaptive Anpassung der Schrittweite zu einer schnelleren Berechnung der Rekonstruktionen führen kann im Vergleich zu einer festen, kleinen Wahl der Schrittweite.

Kapitel 5

Zusammenfassung und Ausblick

Ausgehend vom Problem der Vektorfeldtomographie, haben wir aufgezeigt, dass eine Modellierung von Ultraschallmessungen unter Vernachlässigung der Schallgeschwindigkeit, beziehungsweise unter Vernachlässigung des Einflusses der Schallgeschwindigkeit auf die Signalwege, zu Rekonstruktionsfehlern führen. Wir haben daher ein mathematisches Modell hergeleitet, welches diesen Einfluss berücksichtigt.

Ziel war es, ein numerisches Verfahren abzuleiten, welches die Schallgeschwindigkeit aus Laufzeitmessungen rekonstruiert. Dazu haben wir zwei Methoden aufgezeigt, mit denen die Wege der Signale zu einer gegebenen Schallgeschwindigkeit berechnet werden können. Beide Verfahren wurden implementiert und ausgiebig an Beispielen getestet.

Als nächsten Schritt wurde darauf aufbauend ein iteratives, adaptives Verfahren abgeleitet, welches letzten Endes die Schallgeschwindigkeit rekonstruieren kann. Die numerischen Experimente dazu sind sehr vielversprechend. Insbesondere hat sich gezeigt, dass eine geeignete Wahl der Straftermnorm im Tikhonov-Funktional bei entsprechenden Problemen zu besseren Lösungen führen kann. Der Grund dafür ist, dass sich durch verschiedene Normen gewünschte Eigenschaften der Rekonstruktion forcieren lassen.

Das hergeleitete Rekonstruktionsverfahren liefert jedoch nicht nur eine Approximation an die Schallgeschwindigkeit, sondern auch einen kompletten Datensatz \mathcal{G} an Kurven, welche die Laufwege der Signale approximieren. Ausgehend von der Absicht ein Vektorfeld zu rekonstruieren, kann man nun sowohl die Kenntniss der Schallgeschwindigkeit, als auch der Integralkurven, in die Rekonstruktion von Vektorfeldern einfließen lassen. Ausgehend von der Gleichungen 2.13 aus Kapitel 2

$$g(a, b) := T(\gamma) - T(\gamma^-) \approx -2 \int_a^b \frac{\langle v(x), \omega \rangle}{c^2(x)} dl,$$

lässt sich nun relativ einfach ein Verfahren zur Rekonstruktion des Integranden ableiten. Die Schallgeschwindigkeit c ist bekannt und sogar die Wege γ (beziehungsweise γ^-) von a nach b (oder umgekehrt) können aufgrund der Kenntnis der approximierten Geodäten \mathcal{G} interpoliert werden.

Insbesondere würde sich das ART-Verfahren dazu anbieten. Dabei wird nach der Kollokationsmethode ein lineares Gleichungssystem erzeugt, welches dann mit dem Kaczmarz-Methode gelöst wird. Ausgehend von einem Satz Messdaten $(g_i)_{i \in I}$ soll der Integrand

$f := -2 \frac{\langle v(x), \omega \rangle}{c^2(x)}$ aus den Gleichungen

$$g_i = \int_{\gamma_i} f(x) dl$$

für $i \in I$ bestimmt werden. Dazu wird zunächst der Integrand mittels Punktallokation (also einer konstanten Pixelbasis S_j , $j \in J$) diskretisiert. Schließlich wird die Matrix aufgebaut, indem die Längen der Kurven γ_i im Pixel S_j berechnet werden, also $a_{ij} = |\gamma_i \cap S_j|$ für $i \in I$ und $j \in J$. Nach dem Lösen des entstandenen Gleichungssystems und somit der Kenntnis des Integranden f kann dann das Vektorfeld v rekonstruiert werden. Dadurch, dass die Schallgeschwindigkeit und die Geodäten iterativ bestimmt werden, ist auch die Entwicklung eines Algorithmusses denkbar, der das ART-Verfahren in den adaptiven Algorithmus 4.0.22 aus Kapitel 4 einbaut. Der Vorteil wäre, dass gegebenenfalls auf die Taylorentwicklung 2.10 aus Kapitel 2 verzichtet werden kann und direkt von dem Zusammenhang

$$T(\gamma) = \int_{\gamma} \frac{dl}{c(x) + \langle v(x), \omega \rangle},$$

ein Rekonstruktionsverfahren abgeleitet werden könnte. Dafür könnte man folgendermaßen vorgehen:

Algorithmus 5.0.7 (Adaptives Verfahren zur Berechnung von Vektorfeldern)

Eingabe: Schallgeschwindigkeit c_0 , Vektorfeld v_0 , Laufzeiten g

(S0) Wiederhole für $k = 0, 1, 2, \dots$

(a) Berechne eine neue Schallgeschwindigkeit durch Minimierung

$$c_{k+1} := \arg \min_{c \in X} \frac{1}{2} \left\| \int_{\gamma, \cdot} \frac{dl}{c(x) + \langle v_k(x), \omega \rangle} - g(\cdot, \cdot) \right\|_Y^2 + \alpha \Sigma(c, v_k).$$

(b) Bestimme ein neues Vektorfelds mit dem ART-Verfahren F , also

$$v_{k+1} := F(c_{k+1}, v_k).$$

Ausgabe: Schallgeschwindigkeit c_k , Geodäten \mathcal{G}_k , Vektorfeld v_k .

Das adaptive, iterative Optimierungsverfahren zur Berechnung der Schallgeschwindigkeit aus Laufzeitmessungen akustischer Signale liefert zwar gute Ergebnisse, allerdings kann das Gebiet M bisher nur relativ niedrig aufgelöst werden. Damit möglichst auch zwischen zwei Rekonstruktionspunkten zumindest eine Geodäte verläuft müssen bei einer höheren Auflösung mehr Daten gemessen und berücksichtigt werden. Dies erfordert natürlich dann auch in jedem Schritt die Berechnung von mehr geodätischen Kurven. Auf der anderen Seite erhöht sich der Rechenaufwand auch enorm durch die Auswertung des adjungierten Operators R^* , da von jedem Punkt zunächst die geodätische Projektion approximiert werden muss.

Man könnte den Algorithmus also noch dahingehend verbessern, dass dieser auf eine Rechnung auf GPUs optimiert wird. Dazu möchten wir zwei Vorschläge machen:

1. *Vorsortierung der Datensätze*

Da GPUs im Vergleich zu den vorhandenen Recheneinheiten nur wenige Steuereinheiten besitzen, kommt es derzeit zu langen Wartezeiten. Wird nämlich beispielsweise zur Berechnung der Geodäten ein Datensatz an Punkten und Richtungen (x_i, ξ_i^j) , $i \in I$, $j \in J$ an die GPU übergeben, so ist es wahrscheinlich, dass einige Geodäten sehr schnell berechnet werden (wenn sie beispielsweise nur sehr kurz sind oder durch einen sehr glatten Bereich gehen), und wiederum andere sehr viel Rechenzeit benötigen (wenn sie durch ein Gebiet gehen, was sehr starke Schwankungen in der Schallgeschwindigkeit aufweisen). Aufgrund der geringen Anzahl an Steuereinheiten, warten dann viele Recheneinheiten auf die Beendigung der Rechnungen von wenigen anderen. Denn erst wenn der komplette Datensatz berechnet wurde, kann ein neuer geladen und initialisiert werden.

Glücklicherweise hat man einige Informationen darüber, wie schnell sich die Geodäten berechnen lassen, denn man kann davon ausgehen, dass Geodäten, die nur kurz das Gebiet durchlaufen, also falls $\langle x, \xi \rangle \approx 0$ gilt, schnell bestimmt werden können. Zusätzlich wäre denkbar die Berechnungszeiten aus dem vorherigen Iterationsschritt mit in die Umsortierung einfließen zu lassen. Dies erfordert allerdings einen höheren Speicherbedarf und es ist nicht klar, ob der Aufwand der Umsortierung den Gesamtaufwand aufwiegt.

2. *Approximation der geodätischen Projektion*

Ein Großteil des Rechenaufwands entsteht bei der Berechnung des adjungierten Operators R^* . Und dabei wiederum, in der Ausführung von Algorithmus 4.3.4, bei Berechnung der geodätischen Projektionen.

Die Idee ist nun, in der Annahme, dass die geodätische Projektion $\Pi : TM \times S^1 \rightarrow \partial M$ eine glatte Abbildung ist und somit gut approximiert werden kann. Man könnte also Π als Abbildung vom \mathbb{R}^4 in den \mathbb{R}^2 auffassen und diese durch Polynomfunktionen annähern. Diese dann auszuwerten, sollte sehr einfach und effektiv sein.

Insgesamt verspricht sich der Autor durch diese Maßnahmen zum einen die Möglichkeit einer feineren Diskretisierung im Gebiet und zum anderen die Möglichkeit der effizienten Umsetzung der Methode zur Berechnung der Geodäten als Minimierungsproblem in den Rekonstruktionsalgorithmus.

Anhang A

Quellcode

Alle Programme wurden in Matlab verfasst. Es sind nur die wesentlichen Programmteile angegeben, also ohne den Code für Textausgaben, Plots und die Speicherung von Daten.

Skript - Rekonstruktion des Brechungsindex

```
1  %*-----MATLAB-Skript-S_Transport-----
2  % Berechnet Rekonstruktionen von Metriken aufgrund von
3  % Laufzeitmessungen akustischer Signale
4  % im Einheitskreis. Es koennen diverse Beispiele für
5  % Metriken gewählt werden.
6  % Udo Schroeder, Saarland University, 04.02.2014
7  %*-----
8  zeitTotalStart      = tic;
9  %-----Parameter fuer die Messwerte-----
10 rauschen            = 0;          %Die Messdaten werden mit einem
    Rauschen gestört
11 anzAufpunkte        = 100;
12 anzRichtungen       = 200;
13 anzMessdaten        = struct('anzAufpunkte', anzAufpunkte,
14 'anzRichtungen', anzRichtungen);
15 %-----Parameterwahl fuer die Rekonstruktion-----
16 % Berechnung der Geodaeten (Schrittweitensteuerung)
17 options(1)           = 1/20;      %Ausgangsschrittweite
18 options(2)           = 1e-3;      %Fehlertoleranz
19 options(3)           = 0.90;      %Sicherheitsfaktor
20 % Tikhonov-Regularisierung
21 maxIterUpGeodaeten  = 400;        %Anzahl der Neuberechnungen
    %der Geodaeten, maximale
22 %Iterationsschritte
23 alpha                = 0.1;       %Regularisierungsparameter
24 p                    = 2;         %Norm des
    %Regularisierungsterms
25
26
27
28 % Abstiegsverfahren
```

```

29 maxIterUpMetrik      = 0;           %Abstiegsschritte, also
30                               %maximale Updates der
31                               %Schallgeschwindigkeit bei
32                               %festen Geodaeten
33 mu                   = 0.05;       %Relaxationsparameter
34 % Optionale Parameter zur Verbesserung der Konvergenz
35 maxVerkleinerungen  = 0;           %Verkleinere den
36                               %Relaxationsparameter,
37                               %falls  $J_{k+1} > J_k$ 
38 % Diskretisierung
39 ansatz               = 'linear';   %Ansatzfunktionen:
40                               %linear, FEM
41 diskParameter       = 1/10;       %Gitternetzweite
42 %-----Modellprobleme-----
43 %1. 'eukl':          Euklidische Metrik
44 %2. 'spitzen':      Peaks
45 %3. 'kreis':        Hohler Kreis mit Innenleben
46 %4. 'positiv':      Konstante Krümmung (positiv)
47 %5. 'negativ':      Konstante Krümmung (negativ)
48 %6. 'unstetig':    Unstetige Metrik
49 %7. 'welle':        Starke Schwankungen
50 modellproblem       = 'eukl';
51 %-----Parallelrechnung-----
52 needNewWorkers      = (isempty(gcf('nocreate')));
53 if needNewWorkers
54     parpool('LocalProfile1', 30);
55 end
56 %-----Erzeugung der Messdaten-----
57 [messdatenGeodaeten, messdatenLaufzeit, originalKnoten] =
    ErzMessdaten(modellproblem, anzMessdaten, konti, ansatz,
    diskParameter, rechnungsID, plotparameter, verbose);
58 if (rauschen)
59     for i = 1:size(messdatenLaufzeit,1)
60         if messdatenLaufzeit(i,5) ~= 0
61             messdatenLaufzeit(i,5) = messdatenLaufzeit(i,5) +
                rauschen*(rand - 0.5);
62         end
63     end
64 end
65 %-----Initialisierung-----
66 [gitter, knoten]    = InitGitter(ansatz, diskParameter,
    verbose);
67 normalGitter        = ErzNormalesGitter(ansatz, gitter);
68 AppSchallgeschwindigkeit = @(x) BerAppSchallgeschwindigkeit(x,
    ansatz, gitter, knoten);

```

```

69 appDatenGeodaeten      = BerGeodaeten(anzAufpunkte,
    anzRichtungen, AppSchallgeschwindigkeit, options, verbose);
70 appDatenLaufzeit      = BerLaufzeit(appDatenGeodaeten, @(x)
    1./AppSchallgeschwindigkeit(x), verbose);
71 diffDatenLaufzeit(:,5) = appDatenLaufzeit(:,5) -
    messdatenLaufzeit(:,5);
72 funktional(1)         = norm(diffDatenLaufzeit(:,5)) + alpha*
    sum(sum(power(abs(knoten-1), p)))^(1/p);
73 fehlerKnoten          = zeros(1, maxIterUpGeodaeten + 1);
74 groesseKnoten        = size(knoten, 1);
75 abwKnoten             = originalKnoten - knoten;
76 switch ansatz
77     case 'linear'
78         for j = 1:groesseKnoten
79             fehlerKnoten(1) = fehlerKnoten(1) + power(sum(sum(abs(
                abwKnoten(:,j).^p))),1/p)/groesseKnoten^2;
80         end
81     case 'FEM'
82         fehlerKnoten(1) = power(sum(sum(abs(abwKnoten(:,j).^p))),1/
                p)/groesseKnoten;
83 end
84 %-----Rekonstruktion fuer p > 1-----
85 if p ~= 1
86     for l = 1:maxIterUpGeodaeten
87         zeitDurchlaufStart = tic;
88         knoten_alt          = knoten;
89         adjWerte            = BerAdjRadon(normalGitter,
                appDatenGeodaeten, diffDatenLaufzeit, anzMessdaten,
                verbose);
90         [pNormAbwMetrik, adjWerte] = BerAbwMetrik(ansatz, knoten,
                p, adjWerte, diskParameter);
91         ableitung            = -adjWerte + alpha*
                pNormAbwMetrik;
92         ableitung_neu        = ableitung;
93         for k = 1:maxIterUpMetrik %Update der Metrik ohne Anpassung
                der Geodaeten
94             knoten            = knoten - mu*ableitung;
95             AppSchallgeschwindigkeit = @(x)
                BerAppSchallgeschwindigkeit(x, ansatz, gitter,
                knoten);
96             appDatenLaufzeit  = BerLaufzeit(
                appDatenGeodaeten, @(x) 1./AppSchallgeschwindigkeit(
                x), verbose);
97             diffDatenLaufzeit(:,5) = appDatenLaufzeit(:,5) -
                messdatenLaufzeit(:,5);

```

```

98         adjWerte                = BerAdjRadon(normalGitter,
          appDatenGeodaeten, diffDatenLaufzeit, anzMessdaten,
          verbose);
99     [pNormAbwMetrik, adjWerte] = BerAbwMetrik(ansatz,
          knoten, p, adjWerte, diskParameter);
100    ableitung                    = -adjWerte + alpha*
          pNormAbwMetrik;
101    ableitung_neu                = ableitung_neu + ableitung
          ;
102    end
103    ableitung                    = ableitung_neu/(
          maxIterUpMetrik + 1);
104    normAbleitung(1)              = norm(ableitung);
105    knoten                       = knoten_alt - mu*ableitung;
106
107    AppSchallgeschwindigkeit     = @(x)
          BerAppSchallgeschwindigkeit(x, ansatz, gitter, knoten);
108    appDatenGeodaeten           = BerGeodaeten(anzAufpunkte,
          anzRichtungen, AppSchallgeschwindigkeit, options,
          verbose);
109    appDatenLaufzeit            = BerLaufzeit(appDatenGeodaeten
          , @(x) 1./AppSchallgeschwindigkeit(x), verbose);
110    diffDatenLaufzeit(:,5)      = appDatenLaufzeit(:,5) -
          messdatenLaufzeit(:,5);
111    funktional(1+1)              = norm(diffDatenLaufzeit(:,5))
          + alpha*sum(sum(power(abs(knoten-1), p)))^(1/p);
112    mu_neu                      = mu;
113    anzVerkleinerungen          = 0;
114    while funktional(1+1) - funktional(1) > 0
115        anzVerkleinerungen = anzVerkleinerungen + 1;
116        if anzVerkleinerungen > maxVerkleinerungen
117            break;
118        end
119        mu_neu                = mu_neu/2;
120        knoten                 = knoten_alt - mu_neu*
          ableitung;
121    AppSchallgeschwindigkeit     = @(x)
          BerAppSchallgeschwindigkeit(x, ansatz, gitter,
          knoten);
122    appDatenGeodaeten           = BerGeodaeten(anzAufpunkte
          , anzRichtungen, AppSchallgeschwindigkeit, options,
          verbose);
123    appDatenLaufzeit            = BerLaufzeit(
          appDatenGeodaeten, @(x) 1./AppSchallgeschwindigkeit(
          x), verbose);

```

```

124         pNormAbwMetrik           = BerAbwMetrik(ansatz ,
125             knoten , p, knoten , diskParameter);
126         diffDatenLaufzeit(:,5)    = appDatenLaufzeit(:,5) -
            messdatenLaufzeit(:,5);
127         funktional(l+1)           = norm(diffDatenLaufzeit
            (:,5)) + alpha*sum(sum(power(abs(knoten-1), p)))^(1/
            p);
128     end
129     abwKnoten                     = originalKnoten - knoten;
130     switch ansatz
131     case 'linear'
132         for j = 1:groesseKnoten
133             fehlerKnoten(l+1) = fehlerKnoten(l+1) + power(
134                 sum(sum(abs(abwKnoten(:,j).^p))),1/p)/
135                 groesseKnoten^2;
136         end
137     case 'FEM'
138         fehlerKnoten(l+1) = power(sum(sum(abs(abwKnoten(:,j)
139            ).^p))),1/p)/groesseKnoten;
140     end
141     zeitDurchlauf = toc(zeitDurchlaufStart);
142 end
143 %-----Rekonstruktion fuer p = 1-----
144 else
145     for l = 1:maxIterUpGeodaeten
146         zeitDurchlaufStart = tic;
147         knoten_alt         = knoten;
148         adjWerte           = BerAdjRadon(normalGitter ,
149             appDatenGeodaeten , diffDatenLaufzeit , anzMessdaten ,
150             verbose);
151         [pNormAbwMetrik , adjWerte] = BerAbwMetrik(ansatz , knoten ,
152             p, adjWerte , diskParameter);
153         knoten              = Threshold(knoten_alt +
154             adjWerte*mu - 1, alpha) + 1;
155         AppSchallgeschwindigkeit = @(x)
156             BerAppSchallgeschwindigkeit(x, ansatz , gitter , knoten);
157         appDatenGeodaeten    = BerGeodaeten(anzAufpunkte ,
158             anzRichtungen , AppSchallgeschwindigkeit , options ,
159             verbose);
160         appDatenLaufzeit     = BerLaufzeit(appDatenGeodaeten
161             , @(x) 1./AppSchallgeschwindigkeit(x) , verbose);
162         diffDatenLaufzeit(:,5) = appDatenLaufzeit(:,5) -
163             messdatenLaufzeit(:,5);
164         funktional(l+1)      = norm(diffDatenLaufzeit(:,5))
165             + alpha*sum(sum(abs(knoten-1)));

```

```

152     for k = 1:maxIterUpMetrik %Update der Metrik ohne Anpassung
153         der Geodaeten
154             knoten = Threshold(knoten_alt +
155                 adjWerte*mu - 1, alpha) + 1;
156             AppSchallgeschwindigkeit = @(x)
157                 BerAppSchallgeschwindigkeit(x, ansatz, gitter,
158                 knoten);
159             appDatenLaufzeit = BerLaufzeit(
160                 appDatenGeodaeten, @(x) 1./AppSchallgeschwindigkeit(
161                 x), verbose);
162             diffDatenLaufzeit(:,5) = appDatenLaufzeit(:,5) -
163                 messdatenLaufzeit(:,5);
164             adjWerte = BerAdjRadon(normalGitter,
165                 appDatenGeodaeten, diffDatenLaufzeit, anzMessdaten,
166                 verbose);
167             [pNormAbwMetrik, adjWerte] = BerAbwMetrik(ansatz,
168                 knoten, p, adjWerte, diskParameter);
169             knoten_alt = knoten;
170             funktional(l+1) = norm(diffDatenLaufzeit
171                 (:,5)) + alpha*sum(sum(abs(knoten-1)))
172         end
173         normAbleitung(l) = norm(adjWerte);
174         mu_neu = mu;
175         anzVerkleinerungen = 0;
176         while funktional(l+1) - funktional(l) >= 0
177             anzVerkleinerungen = anzVerkleinerungen + 1;
178             if anzVerkleinerungen > maxVerkleinerungen
179                 break;
180             end
181             mu_neu = mu_neu/2;
182             knoten = Threshold(knoten_alt +
183                 mu_neu*adjWerte - 1, alpha) + 1;
184             AppSchallgeschwindigkeit = @(x)
185                 BerAppSchallgeschwindigkeit(x, ansatz, gitter,
186                 knoten);
187             appDatenGeodaeten = BerGeodaeten(anzAufpunkte
188                 , anzRichtungen, AppSchallgeschwindigkeit, options,
189                 verbose);
190             appDatenLaufzeit = BerLaufzeit(
191                 appDatenGeodaeten, @(x) 1./AppSchallgeschwindigkeit(
192                 x), verbose);
193             pNormAbwMetrik = BerAbwMetrik(ansatz,
194                 knoten, 1, knoten, diskParameter);
195             diffDatenLaufzeit(:,5) = appDatenLaufzeit(:,5) -
196                 messdatenLaufzeit(:,5);

```

```

177         funktional(l+1) = norm(diffDatenLaufzeit
178             (:,5)) + alpha*sum(sum(abs(knoten-1)));
179     end
180     abwKnoten = originalKnoten - knoten;
181     switch ansatz
182     case 'linear'
183         for j = 1:groesseKnoten
184             fehlerKnoten(l+1) = fehlerKnoten(l+1) + power(
185                 sum(sum(abs(abwKnoten(:,j).^p))),1/p)/
186                 groesseKnoten^2;
187         end
188     case 'FEM'
189         fehlerKnoten(l+1) = power(sum(sum(abs(abwKnoten(:,j)
190            ).^p))),1/p)/groesseKnoten;
191     end
192     zeitDurchlauf = toc(zeitDurchlaufStart);
193 end
194 end
195 zeitTotal = toc(zeitTotalStart);
196 %-----Terminator-----
197 if needNewWorkers
198     delete(gcp('nocreate'))
199 end
200 fprintf('\nProgramm korrekt beendet.\n\n');

```

Gitterinitialisierung

```

1 function [gitter, knoten] = InitGitter(ansatz, diskParameter,
2     verbose, Schallgeschwindigkeit)
3 %InitGitter initialisiert ein Gitter auf dem die
4 %Schallgeschwindigkeit approximiert werden soll.
5 %ansatz: 'linear' lineare Interpolation, äquidistantes Gitter;
6 %'FEM' lineare Ansatzfunktionen auf Triangulierung
7 %knoten: Schallgeschwindigkeit(gitter)
8
9 if nargin == 3
10     Schallgeschwindigkeit = @(x) ones(size(x,1),1);
11 end
12 switch ansatz
13     case 'linear'
14         [xRichtung, yRichtung] = meshgrid(-1:diskParameter:1);
15         gitter = [xRichtung, yRichtung];
16         knoten = ones(size(xRichtung,1));
17         for i = 1:size(gitter,1)
18             for j = 1:size(gitter,1)

```

```

18         if norm([gitter(j,i) gitter(j,size(gitter,2)/2+i)])
19             < 1
20             knoten(j,i) = Schallgeschwindigkeit([gitter(j,i
21                 ) gitter(j,size(gitter,2)/2+i)]);
22         else
23             knoten(j,i) = 1;
24         end
25     end
26 case 'FEM'
27     geomKreis = [1; 0; 0; 1];
28     decDiskrGeomKreis = decsg(geomKreis);
29     diskGeomKreis = initmesh(decDiskrGeomKreis, 'hmax',
30         diskParameter, 'init', 'off');
31     gitter = DelaunayTri(diskGeomKreis');
32     knoten = Schallgeschwindigkeit(gitter.X);
33 otherwise
34     gitter = 0;
35     knoten = 0;
36     warning('Unerlaubter Parameter <ansatz>!');
37 end
38 end

```

Auswertung der Schallgeschwindigkeit

```

1 function [c, c_x, c_y] = BerAppSchallgeschwindigkeit(punkt, ansatz,
2     gitter, knoten)
3 %BerAppSchallgeschwindigkeit wertet die approximierete
4 %Schallgeschwindigkeit an einem Punkt, ggf. mit den zugehörigen
5 %partiellen Ableitungen, aus. Die approximierete
6 %Schallgeschwindigkeit ist zunächst über die Art des Ansatzes
7 %bestimmt. Darüber hinaus gehen das Gitter (bzw. die
8 %Triangulierung) und die Knotenwerte auf den Gitterpunkten mit
9 %ein.
10 if nargin == 1
11     ansatz = 'linear';
12     [xRichtung, yRichtung] = meshgrid(-1:0.25:1);
13     gitter = [xRichtung, yRichtung];
14     knoten = ones(size(xRichtung,1));
15 end
16 switch ansatz
17     case 'linear'
18         if nargin > 1
19             diff = abs(gitter(1,2) - gitter(1,1))*1e-8;
20             xdiff = punkt(:,1) + diff;

```

```

21         ydiff = punkt(:,2) + diff;
22         ausw = InterpLinear([punkt; xdiff punkt(:,2); punkt
23             (:,1) ydiff], knoten);
24         c_x = (ausw(length(ausw)/3 + 1:length(ausw)*2/3) -
25             ausw(1:length(ausw)/3))/diff;
26         c_y = (ausw(length(ausw)*2/3 + 1:length(ausw)) -
27             ausw(1:length(ausw)/3))/diff;
28         c = ausw(1:length(ausw)/3);
29     else
30         c = InterpLinear(punkt, knoten);
31     end
32 case 'FEM'
33     [dreieck, BCS] = pointLocation(gitter, punkt); % BCS =
34         Baryzentrische Koordinaten
35     n = size(dreieck,1);
36     c = zeros(n,1);
37     if nargout > 1
38         c_x = zeros(n,1);
39         c_y = zeros(n,1);
40         for i = 1:size(dreieck,1)
41             if isnan(dreieck(i))
42                 c(i) = 1;
43                 c_x(i) = 0;
44                 c_y(i) = 0;
45             else
46                 c(i) = BCS(i,:)*knoten(gitter(dreieck(i),:))
47                     /(BCS(i,1) + BCS(i,2) + BCS(i,3));
48                 %Berechnung der partiellen Ableitungen. Die
49                 %Ebene des Dreiecks ist über die drei
50                 %Knotenpunkte eindeutig bestimmt. Es gibt
51                 %dann für jeden Punkt eine Parameter-
52                 %darstellung. Diese wird dann abgeleitet.
53                 z21 = knoten(gitter(dreieck(i),2)) - knoten
54                     (gitter(dreieck(i),1));
55                 z31 = knoten(gitter(dreieck(i),3)) - knoten
56                     (gitter(dreieck(i),1));
57                 x21 = gitter.X(gitter(dreieck(i),2),1) -
58                     gitter.X(gitter(dreieck(i),1),1);
59                 x31 = gitter.X(gitter(dreieck(i),3),1) -
60                     gitter.X(gitter(dreieck(i),1),1);
61                 y21 = gitter.X(gitter(dreieck(i),2),2) -
62                     gitter.X(gitter(dreieck(i),1),2);
63                 y31 = gitter.X(gitter(dreieck(i),3),2) -
64                     gitter.X(gitter(dreieck(i),1),2);
65                 divisor = y21*x31 - x21*y31;
66                 c_x(i) = (z31*y21 - z21*y31)/divisor;

```

```

56             c_y(i) = (z21*x31 - z31*x21)/divisor;
57         end
58     end
59     c_x = c_x';
60     c_y = c_y';
61     else
62         for i = 1:size(dreieck,1)
63             if isnan(dreieck(i))
64                 c(i) = 1;
65             else
66                 c(i) = BCS(i,:)*knoten(gitter(dreieck(i),:))/(
                    BCS(i,1) + BCS(i,2) + BCS(i,3));
67             end
68         end
69     end
70     c = c';
71     otherwise
72         warning('Unerlaubter Parameter <ansatz>!');
73 end
74 end

```

Lineare Interpolation auf dem Gitter

```

1 function interpWerte = InterpLinear(punkte, knoten)
2 %InterpLinear interpoliert die Punkte im Gebiet (gitter) mit
3 %zugehoerigen Knotenwerten.
4 n = size(knoten,1);
5 temp = (punkte(:,1) + 1)*(n-1)/2 + 1; %steps from
   left grid to u0
6 numLeft = floor(temp); %number of the detector point
   to the left of u0
7 numLeft(numLeft <= 1) = 1;
8 numLeft(numLeft >= n) = n;
9 numRight = numLeft + 1; %number of the detector point
   to the right of u0
10 numRight(numLeft >= n-1) = n;
11
12 distLeft = 1 - temp + numLeft; %distance from the left detector
   point to u0
13 distRight = 1 - distLeft; %distance from the right detector point
   to u0
14
15 temp = (punkte(:,2) + 1)*(n-1)/2 + 1; %do the same for
   vertical detector points
16 numBelow = floor(temp);
17 numBelow(numBelow <= 1) = 1;

```

```

18 numBelow(numBelow >= n) = n;
19 numAbove      = numBelow + 1;
20 numAbove(numBelow >= n-1) = n;
21
22 distBelow      = 1 - temp + numBelow;
23 distAbove      = 1 - distBelow;
24 m              = size(punkte,1);
25 interpWerte = zeros(m,1);
26
27 for i = 1:m
28     interpWerte(i) = distLeft(i)*(distAbove(i)*knoten(numAbove(i),
29         numLeft(i)) ...
30         + distBelow(i)*knoten(numBelow(i),numLeft(i)))
31         + distRight(i)*(distAbove(i)*knoten(numAbove(i)
32         ,numRight(i)) ...
33         + distBelow(i)*knoten(numBelow(i),numRight(i)))
34         ;
35 end
36 end

```

Umsortierung des Gitters

```

1 function normalGitter = ErzNormalesGitter(ansatz, gitter)
2 %ErzNormalesGitter sortiert das Gitter, so dass es in einer
3 %nx2-Matrix steht
4 switch ansatz
5     case 'FEM'
6         normalGitter = gitter.X;
7     case 'linear'
8         [n,m]          = size(gitter);
9         normalGitter = reshape(gitter,n*m/2,2);
10 end
11 end

```

Berechnung der Geodäten

```

1 function datenGeodaeten = BerGeodaeten(anzAufpunkte, anzRichtungen,
2     Schallgeschwindigkeit, options, verbose)
3 %BerGeodaeten berechnet auf der Einheitskreisscheibe zur
4 %Schallgeschwindigkeit auf anz_aufpunkte Aufpunkten
5 %(aequidistant) mit jeweils anz_richtungen Richtungen Geodaeten
6 %und deren Laufzeit zur gegebenen Schallgeschwindigkeit. Die
7 %Ausgabe erfolgt in einem datensatz[Punkt, Richtung, Laufzeit].
8 %options: [Schrittweite SWS, Fehlertoleranz, Sicherheitsfaktor]
9

```

```

9  anzGeodaeten = anzAufpunkte*anzRichtungen;
10
11  if nargin == 3
12      options = [0.05, 1e-8, 0.9];
13  end
14
15  %Parameter fuer die Schrittweitensteuerung
16  h = options(1);      %Ausgangsschrittweite
17  eps = options(2);   %Fehlertoleranz
18  rho = options(3);   %Sicherheitsfaktor
19
20  %Datenaufnahme
21  datenGeodaeten = cell(anzAufpunkte, anzRichtungen); %Cell-Array zur
    Speicherung der Wege; anzAufpunkte x anzRichtungen
22  parfor nrAufpunkt = 1:anzAufpunkte %PARALLEL COMPUTING
23      %Erzeugung der Anfangsaufpunkte
24      u_0 = zeros(1,5); %[Punkt, Richtung, Laufzeit]
25      winkelAufpunkt = (nrAufpunkt - 1)*2*pi/anzAufpunkte;
26      u_0(1:2) = [sin(winkelAufpunkt), cos(winkelAufpunkt)];
27      for nrRichtung = 1:anzRichtungen
28          %Erzeugung der Anfangsrichtungen
29          winkelRichtung = (nrRichtung - 1)*2*pi/anzRichtungen;
30          u_0(3:4) = [sin(winkelRichtung), cos(winkelRichtung)
    ];
31          if dot(u_0(1:2),u_0(3:4)) > 0 %*[sin(winkelRichtung); cos(
    winkelRichtung)] > 0 %Outflow wird zugelassen; Inflow
    nicht
32              zulaessigeRichtung = 1;
33          else
34              zulaessigeRichtung = 0;
35          end
36          datenGeodaeten{nrAufpunkt, nrRichtung} = u_0;
37          %Initialisierung der Schrittweitensteuerung
38          u_alt = u_0;
39          u_neu = u_0;
40          diff = 0;      %Abbruchkriterium
41          h_akt = h;
42          l = 1;
43          %Schrittweitensteuerung
44          while(zulaessigeRichtung)
45              k = 0;
46              while(1)
47                  if (k>0); %Festlegung der neuen Schrittweite
48                      h_akt = h_akt * (rho * eps / diff)^(1/5);
49                  else
50                      h_akt = h;

```

```

51         end
52         k = k + 1;
53         %2 Schritte mit h/2
54         v = CompStepRK4Transport(u_alt, -0.5*h_akt,
55             Schallgeschwindigkeit);
56         u_neu = CompStepRK4Transport(v, -0.5*h_akt,
57             Schallgeschwindigkeit);
58         normU_neu = norm(u_neu(1:2));
59         if (normU_neu - 1 >= 0) %Ueberprueft, ob der neu
60             berechnete Punkt noch im Gebiet liegt
61             break;
62         end
63         %1 Schritt mit h
64         v = CompStepRK4Transport(u_alt, -h_akt,
65             Schallgeschwindigkeit);
66         %Fehlerschätzer
67         diff = norm(v - u_neu)/15;
68         if (diff <= eps)
69             break;
70         end
71         end
72         if (normU_neu - 1 >= 0) %Ueberprueft, ob der neu
73             berechnete Punkt noch im Gebiet liegt
74             break;
75         end
76         l = l + 1;
77         u_alt = u_neu;
78         datenGeodaeten{nrAufpunkt, nrRichtung} = [
79             datenGeodaeten{nrAufpunkt, nrRichtung}; u_neu];
80     end
81     if (zulaessigeRichtung) %Letzter Schritt wird so
82         durchgefuehrt, dass die Kurve auf dem Rand endet
83         produkt = dot(u_alt(1:2), u_alt(3:4));
84         norm_richt_sq = power(norm(u_alt(3:4)), 2);
85         norm_punkt_sq = power(norm(u_alt(1:2)), 2);
86         h_akt = -min(-produkt/norm_richt_sq + sqrt(
87             power((produkt/norm_richt_sq), 2) - (norm_punkt_sq - 1)/
88             norm_richt_sq), -produkt/norm_richt_sq - sqrt(
89             power((produkt/norm_richt_sq), 2) - (norm_punkt_sq - 1)/
90             norm_richt_sq));
91         u_neu = u_alt - h_akt*RechteSeiteTransport(
92             u_alt, Schallgeschwindigkeit);
93         datenGeodaeten{nrAufpunkt, nrRichtung} = [
94             datenGeodaeten{nrAufpunkt, nrRichtung}; real(u_neu)
95         ];
96     end

```

```

83     end
84 end
85 end

```

Ausführung eines Runge-Kutta-Schritts

```

1 function y = CompStepRK4Transport(y0, h, Schallgeschwindigkeit)
2 %CompStepRK4Transport berechnet einen Iterationsschritt mit Hilfe
3 %des klassischen Runge-Kutta-Verfahrens 4. Ordnung für das
4 %Transportproblem zu gegebener Schallgeschwindigkeit.
5 %Die Startstelle ist hier belanglos.
6 %   Eingabe: y0 - Startwert, Schallgeschwindigkeit - diese liegt
7 %   dem Problem zugrunde, h - Schrittweite.
8 %Ausgabe: y - Vektor mit den dazugehörigen Werten.
9
10 k1 = RechteSeiteTransport(y0, Schallgeschwindigkeit);
11 k2 = RechteSeiteTransport(y0 + 0.5*h*k1, Schallgeschwindigkeit);
12 k3 = RechteSeiteTransport(y0 + 0.5*h*k2, Schallgeschwindigkeit);
13 k4 = RechteSeiteTransport(y0 + h*k3, Schallgeschwindigkeit);
14 y = y0 + h * (k1 + 2*(k2 + k3) + k4)/6;
15 end

```

Auswertung der rechten Seite

```

1 function v = RechteSeiteTransport(u, Schallgeschwindigkeit)
2 %RechteSeite wertet die rechte Seite der ODE zur
3 %Schallgeschwindigkeit einmalig aus.
4
5 if norm(u(1:2)) < 0.95
6     [c, c_x, c_y] = Schallgeschwindigkeit(u(1:2));
7 else
8     c = 1;
9     c_x = 0;
10    c_y = 0;
11 end
12 v(1) = u(3);
13 v(2) = u(4);
14 norm_xi = power(u(3),2) + power(u(4),2);
15 prod_xi_c = u(3)*c_x + u(4)*c_y;
16 v(3) = -(2*c_x*norm_xi - 4*u(3)*prod_xi_c)/c;
17 v(4) = -(2*c_y*norm_xi - 4*u(4)*prod_xi_c)/c;
18 v(5) = norm([v(1) v(2)]) / c;
19 end

```

Auswertung des Operators R_k

```

1 function [laufzeit] = BerLaufzeit(datenGeodaeten, Brechungsindex,
   verbose)
2 %BerLaufzeit berechnet zu einem gegebenen Datensatz und zu
3 %gegebenem Brechungsindex (M -> R) die Laufzeiten entlang der
4 %durch den Datensatz gegebenen Wege. Die Ausgabe ist ein Array mit
5 %[Aufpunkt, Aufrichtung, Laufzeit].
6 %Achtung: Brechungsindex = Schallgeschwindigkeit^(-1)!
7 %Die Laufzeit wird mit der Trapezregel approximiert.
8
9 if nargin == 1
10     Brechungsindex = @(x) 1; %Euklidische Metrik
11     verbose        = 0;
12 end
13 n                = numel(datenGeodaeten);
14 laufzeit        = zeros(n,5);
15 for i = 1:n
16     laufzeit(i,1:4) = datenGeodaeten{i}(1,1:4);
17     m                = size(datenGeodaeten{i},1);
18     metrik_wert      = Brechungsindex(datenGeodaeten{i}(1:m-1,1:2));
19     for j = 1:m-1
20         laufzeit(i,5) = laufzeit(i,5) + norm(datenGeodaeten{i}(j
   +1,1:2) - datenGeodaeten{i}(j,1:2)) *(metrik_wert(j));
21     end
22 end
23 end

```

Auswertung des Operators R_k^*

```

1 function adjWerte = BerAdjRadon(punkte, datenGeodaeten,
   datenLaufzeit, parameter, verbose)
2 %BerAdjRadon berechnet die adjungierte Radon-Transformation in
3 %gegebenen Punkten.
4 %Die Punkte [punkte] müssen zunächst geodaetisch auf
5 %einen Punkt x auf dem Rand des Gebiets projiziert werden, so
6 %dass eine Geodaete von x in Richtung xi durch y laeuft. Dann
7 %kann die Laufzeit aus datenLaufzeit interpoliert werden.
8
9 radius          = 0.95;
10 anzRichtungen  = parameter.anzRichtungen; %diskrete Richtungen des
   Kreises
11 anzGitterpunkte = size(punkte, 1);
12 anzAufpunkte   = parameter.anzAufpunkte;
13 adjWerte       = zeros(anzGitterpunkte, 1);
14 aufpunkte      = datenLaufzeit(1:anzAufpunkte, 1:2);
15 if (verbose) parfor_progress(anzRichtungen); end
16 parfor nrXi = 1:anzRichtungen %Diskretisierung der S^1

```

```

17     nrDichtesterPunkt_alt = 1;
18     index_alt            = 1;
19     xi                   = datenGeodaeten{1,nrXi}(1,3:4); %
    Ausgangsrichtung der Geodaeten, ebenfalls Richtung der
    ersten Projektion auf den Rand
20     xiPerp               = [-xi(2) xi(1)]; %Senkrechte zur
    Richtung, normiert
21     integrand            = zeros(anzGitterpunkte,1);
22     for nrGitterpunkt = 1:anzGitterpunkte
23         gitterpunkt = punkte(nrGitterpunkt,:);
24         if gitterpunkt(1)^2 + gitterpunkt(2)^2 < radius %Punkt muss
    echt im Inneren liegen
25             randpunkt = ProjLinRand(gitterpunkt, xi); %
    eukl. Projektion von y auf den Rand in Richtung xi
26             nrDichtesterPunkt = FindeDichtAufpunkt(randpunkt,
    aufpunkte);
27             index = FindeIndexUebertritt(
    datenGeodaeten{nrDichtesterPunkt, nrXi}, gitterpunkt
    , xi);
28             if index == -1 %Die Geodaete laeuft durch y,
    Interpolation nicht notwendig!
29                 nrDichtesterPunkt_alt = nrDichtesterPunkt;
30                 integrand(nrGitterpunkt) = datenLaufzeit(
    nrDichtesterPunkt,5);
31             else %Die Geodaete laeuft nicht durch y
32                 vIndexPerp = [-datenGeodaeten{
    nrDichtesterPunkt, nrXi}(index,4) datenGeodaeten
    {nrDichtesterPunkt, nrXi}(index,3)];
33                 cosBeta_neu = dot(gitterpunkt - datenGeodaeten{
    nrDichtesterPunkt, nrXi}(index,1:2), vIndexPerp)
    ;
34                 cosBeta = cosBeta_neu;
35                 while cosBeta*cosBeta_neu >= 0
36                     %Ueberpruefe ob man links oder rechts vom Punkt
    vorbei gelaufen ist, danach Update des
    Randpunktes
37                     if cosBeta_neu > 0
38                         nrDichtesterPunkt_alt = nrDichtesterPunkt;
39                         if nrDichtesterPunkt == 1
40                             nrDichtesterPunkt = anzAufpunkte;
41                         else
42                             nrDichtesterPunkt = nrDichtesterPunkt -
    1;
43                         end
44                     elseif cosBeta_neu < 0
45                         nrDichtesterPunkt_alt = nrDichtesterPunkt;

```

```

46         if nrDichtesterPunkt == anzAufpunkte
47             nrDichtesterPunkt = 1;
48         else
49             nrDichtesterPunkt = nrDichtesterPunkt +
50                 1;
51         end
52     else %Geodaete verlauft durch y, keine neue
53         Geodaete untersuchen, sondern direkt
54         auswerten
55         nrDichtesterPunkt_alt = nrDichtesterPunkt;
56         index_alt = index;
57         break;
58     end
59     index = FindeIndexUebertritt(datenGeodaeten{
60         nrDichtesterPunkt, nrXi}, gitterpunkt, xi);
61     %Berechne ob man links oder rechts am Punkt
62     vorbei gelaufen ist
63     vIndexPerp = [-datenGeodaeten{
64         nrDichtesterPunkt, nrXi}(index,4)
65         datenGeodaeten{nrDichtesterPunkt, nrXi}(
66         index,3)];
67     cosBeta = cosBeta_neu;
68     cosBeta_neu = dot(gitterpunkt -
69         datenGeodaeten{nrDichtesterPunkt, nrXi}(
70         index,1:2), vIndexPerp);
71     end
72     %Bestimmung der Schnitte mit xiPerp von den an y
73     vorbeilaufenden Geodaeten
74     A = [-datenGeodaeten{nrDichtesterPunkt_alt, nrXi}(
75         index_alt,3:4)' xiPerp'];
76     B = [-datenGeodaeten{nrDichtesterPunkt, nrXi}(
77         index,3:4)' xiPerp'];
78     a = (gitterpunkt - datenGeodaeten{
79         nrDichtesterPunkt_alt, nrXi}(index_alt, 1:2))';
80     b = (gitterpunkt - datenGeodaeten{nrDichtesterPunkt
81         , nrXi}(index, 1:2))';
82     s1 = A\a;
83     s2 = B\b;
84     t1 = abs(s1(1));
85     t2 = abs(s2(1));
86     %Interpolation zwischen den Geodaeten
87     integrand(nrGitterpunkt) = (t1*datenLaufzeit((nrXi
88         -1)*anzAufpunkte + nrDichtesterPunkt,5) + t2*
89         datenLaufzeit((nrXi -1)*anzAufpunkte +
90         nrDichtesterPunkt_alt,5))/(t1+t2);

```

```

74         end
75     end
76 end
77     adjWerte = adjWerte + integrand;
78 end
79 adjWerte = 2*pi*adjWerte/anzRichtungen; %Skalierung für Integration
    , Trapezregel
80 end

```

Euklidische Projektion auf den Rand

```

1 function randpunkte = ProjLinRand(punkte, richtung)
2 %ProjLinRand projiziert Punkte [punkte] im Inneren der
3 %Einheitskreisscheibe auf dessen Rand in Richtung
4 %[richtung]. Die Richtung sei normiert.
5
6 n = size(punkte);
7 randpunkte = zeros(n);
8 for i = 1:n(1)
9     a0 = power(punkte(i,1),2) + power(punkte(i,2),2) -
10         1;
11     a1 = 2*dot(punkte(i,:),richtung);
12     b = a1/2;
13     diskriminante = power(b,2) - a0;
14     randpunkte(i,:) = punkte(i,:) + (-b + sqrt(diskriminante))*
15         richtung;
16 end
17 end

```

Bestimmung des dichtesten Punktes auf dem Rand

```

1 function nrAufpunkt = FindeDichtAufpunkt(randpunkt, aufpunkte)
2 %FindeDichtAufpunkt gibt die Nummer des dichtesten Punktes auf
3 %dem Rand zurück
4
5 n = size(aufpunkte, 1);
6 alpha = 2*pi - mod(atan2(randpunkt(2), randpunkt(1)),2*pi);
7 nrAufpunkt = mod(round(alpha*n/(2*pi) + 1), n) + 1;
8 end

```

Bestimmung Geodätenabschnitts, der von einer Geraden geschnitten wird

```

1 function index = FindeIndexUebertritt(geodaete, punkt, richtung)
2 %FindeIndexUebertritt bestimmt den Index i, ab dem ein
3 %Uebertritt ueber die Gerade punkt + k*richtung_senkrecht

```

```

4 %stattfindet. Die Kurve geodaete ist stückweise linear, also
5 %gespeichert ueber die einzelnen Eckpunkte x_i.
6
7 if size(geodaete,1) == 1 %Die Geodaete ist leer.
8     index = 1;
9     return;
10 end
11
12 cosAlpha_neu = dot(geodaete(1,1:2) - punkt, richtung);
13 for index = 1:size(geodaete,1)
14     cosAlpha      = cosAlpha_neu;
15     cosAlpha_neu = dot(geodaete(index,1:2) - punkt, richtung);
16     if cosAlpha*cosAlpha_neu < 0
17         index = index - 1;
18         return;
19     end
20 end
21 end

```

Berechnung der p-Norm der Ableitung

```

1 function [pNormAbwMetrik, adjWerte] = BerAbwMetrik(ansatz, knoten,
2     p, adjWerte)
3 %BerAbwMetrik bringt die Ableitung in eine andere Darstellung.
4 %Es wird die p-Norm^(p-1) bestimmt.
5 groesseKnoten = size(knoten, 1);
6 if ansatz == 'linear'
7     adjWerte = reshape(adjWerte, groesseKnoten, groesseKnoten);
8 end
9 pNormAbwMetrik = power(abs(knoten - 1), p-1).*sign(knoten - 1);
10 end

```

Anwendung der Threshold-Abbildung

```

1 function knoten_neu = Threshold(knoten, omega)
2 %Threshold berechnet die Threshold-Abbildung der Knotenwerte.
3 %Hier gilt p = 1.
4
5 omegahalb = omega/2;
6 n = size(knoten, 2);
7 knoten_neu = zeros(size(knoten));
8 for i = 1:n
9     knoten_neu(:,i) = Tres(knoten(:,i), omegahalb);
10 end
11 end
12

```

```

13 function s = Tres(knotenspalte, omegahalb)
14 %Auswertung der Threshold-Funktion
15
16 s = (knotenspalte - omegahalb).*(knotenspalte > omegahalb) + (
    knotenspalte + omegahalb).*(knotenspalte < -omegahalb);
17 end

```

Skript - Erzeugung von Messdaten als Minimierungsproblem

```

1  %*-----MATLAB-Skript-S_ErzeugeMessdaten-----
2  %Zu gegebener Schallgeschwindigkeit werden geodaetische Kurven
3  %mittels Minimierung berechnet. Es wird deren Laufzeit
4  %abgespeichert.
5  % Udo Schroeder, Saarland University, 04.02.2014
6  %*-----
7  %-----Parallelrechnung-----
8  needNewWorkers = (isempty(gcp('nocreate')));
9  if needNewWorkers
10     parpool('LocalProfile1',30)
11 end
12 %-----Zeitmessung-----
13 tic
14 %-----Parameterwahl-----
15 modellproblem = 'spitzen'; %Wahl des Modellproblems
16 anzAufpunkte = 100; %Anzahl der Aufpunkte und
    Datensaeetze
17 anzZielpunkte = 100; %Anzahl der Punkte auf dem Kreis zu
    denen Geodaeten berechnet werden sollen (aequidistant verteilt)
18 n = 50; %Diskretisierungsparameter der
    Geodaeten
19 stab = 0.0; %Regularisierungsparameter fuer die
    Geodaeten
20 %Parameter fuer den Optimierer fminunc
21 options = optimset('Diagnostics', 'off', 'LargeScale', 'off',
    'GradObj', 'off', 'Display', 'off', 'MaxFunEvals', 300*n, '
    TolFun', 1e-6, 'TolX', 1e-6);
22 %-----Initialisierung-----
23 s = zeros(1, n - 1); %Optimierungsvektor
24 punktLaufzeiten = zeros(anzZielpunkte,3 ); %Datenvektor
25 switch modellproblem
26     case 'eukl'
27         c = @(x) SchallEukl(x);
28     case 'spitzen'
29         c = @(x) SchallPeaks(x);
30     case 'kreis'
31         c = @(x) SchallKreis(x);

```

```

32     case 'negativ'
33         c = @(x) SchallKruemmung(x, 'neg', [2, 0.4]);
34     case 'positiv'
35         c = @(x) SchallKruemmung(x, 'pos', [2, 1.2]);
36     case 'unstetig'
37         c = @(x) SchallUnstetig(x);
38     otherwise
39         display('Fehler: □Anderes □Modellproblem □wählen!');
40 end
41 %-----Schallgeschwindigkeiten-----
42 for k = 1:anzAufpunkte
43     alpha                = (k-1)/anzAufpunkte;
44         %Winkel des Aufpunkts
45     aufpunkt            = [sin(alpha*2*pi), cos(alpha*2*pi)
46         ]; %Parametrisierung des Aufpunkts
47     punktLaufzeiten(1,1:2) = aufpunkt;
48 %-----Berechnung der Geodäten (Parallelrechnung)
49     -----
50     parfor p = 2:anzZielpunkte %Durchlaufen des Kreises
51         beta                = (p-1)/anzZielpunkte; %Winkel des
52             Zielpunktes b
53         zielpunkt          = [sin(beta*2*pi + 2*alpha*pi), cos
54             (beta*2*pi + 2*alpha*pi)];
55             %Parametrisierung des
56             Zielpunkts
57         Zielfunktion        = @(s) AppLaufzeit(aufpunkt,
58             zielpunkt, n, c, s) + stab * norm(s)^2;
59             %Zu minimierende Zielfunktion
60         [x, laufzeit]      = fminunc(Zielfunktion, s, options)
61             ; %Aufruf des Minimierers fminunc
62         geodaeten(p,:)     = x;
63         punktLaufzeiten(p,:) = [zielpunkt laufzeit];
64     end
65 end
66 ende = toc;
67 %-----Terminator-----
68 if needNewWorkers
69     delete(gcp('nocreate'))
70 end
71 fprintf('Laufzeit: □%f □Minuten\n', ende/60);

```

Berechnung der approximierten Laufzeit

```

1 function laufzeit = AppLaufzeit(a, b, n, c, s)
2 %AppLaufzeit berechnet die approximierte Laufzeit eines Signals von
   a nach

```

```

3 %b bezueglich einem gegebenen Brechungsindex n = c^(-1) und dem
4 %Abweichungsvektor s. Die Diskretisierungsstufe ist durch n gegeben
5
6 laufzeit          = 0;                               %Initialisierung
7   der Laufzeit
8 bminusa          = b - a;                             %Berechnet b - a
9 schrittweite      = 1 / n;                             %Schrittweite
10 ortho            = OrthoVec2D(bminusa, 0.001);        %Orthogonale zu b -
11   a
12 %-----Einbettung von s vom R^(n-1) in den R^(n+1)-----
13 for i = n:-1:2
14     s(i) = s(i-1);
15 end
16 s(1) = 0;
17 s(n+1) = 0;
18 %-----
19 X      = 0:0.5:1;   %Schrittweite für die Trapezregel zur
20   Berechnung der Integrale
21 for k = 1:n
22     valc      = @(tau) clokak(tau, bminusa, k, schrittweite,
23   ortho, s, a, c); %Parametrisierung von c^(-1)
24     Y          = valc(X);
25     laufzeit   = laufzeit + sqrt(norm(bminusa)^2 * schrittweite^2
26   + (s(k + 1) - s(k))^2) * trapz(X, Y);
27 end
28 end

```

Auswerten der linearisierten Kurve

```

1 function valc = clokak(t, bminusa, k, h, ortho, s, a, c)
2 %clokak parametrisiert den Weg von Gamma_k zu Gamma_k+1 und setzt
3 %ihn in c^(-1) ein.
4 n = numel(t);
5 valc = zeros(1,n);
6 bminusah = bminusa * h;
7 for i = 1:n
8     valc(i) = 1./c(a + k * bminusah + s(k) * ortho + t(i) * (
9     bminusah + (s(k+1) - s(k)) * ortho));
10 end
11 end

```

Erzeugung von Messdaten als Lösung des AWP

```

1 function [messdatenGeodaeten, messdatenLaufzeit, originalKnoten] =
2   ErzMessdaten(modellproblem, anzMessdaten, konti, ansatz,
3   diskParameter)

```

```

2 %ErzMessDaten erzeugt die Messdaten in relativ hoher Genauigkeit.
3 %Input:
4 %modellproblem: Gibt das Modellproblem an, für welches Messdaten
5 %berechnet werden sollen.
6 %anzMessDaten: Anzahl der Aufpunkte und Anzahl der Richtungen.
7 %konti, ansatz, diskParameter: Kontinuierliche Metrik oder
8 %diskretisierte Metrik mit ansatz und Parameter
9 %Output: Geodaeten, Laufzeiten und Knotenwerte
10 %-----Modellprobleme-----
11 %1. 'eukl':      Euklidische Metrik
12 %2. 'spitzen':  Peaks (zum Testen der l1-Metrik)
13 %3. 'kreis':    Hohler Kreis mit Innenleben
14 %4. 'positiv':  Konstante Krümmung (positiv)
15 %5. 'negativ':  Konstante Krümmung (negativ)
16 %6. 'welle':    Starke Schwankungen (fast ueberall ungleich 1)
17 %7. 'unstetig': Unstetige Metrik (Shepp-Logan)
18 %8. 'zufall':   Zufällige Metrik
19 %Parameter fuer die Schrittweitensteuerung für die Erzeugung der
    Messdaten
20 optionsDaten(1)    = 1/40;      %Ausgangsschrittweite 1/40
21 optionsDaten(2)    = 1e-6;      %Fehlertoleranz 1e-6
22 optionsDaten(3)    = 0.90;      %Sicherheitsfaktor 0.90
23 switch modellproblem
24     case 'eukl'
25         Schallgeschwindigkeit = @(x) SchallEukl(x);
26     case 'spitzen'
27         Schallgeschwindigkeit = @(x) SchallPeaks(x);
28     case 'kreis'
29         Schallgeschwindigkeit = @(x) SchallKreis(x);
30     case 'negativ'
31         Schallgeschwindigkeit = @(x) SchallKruemmung(x, 'neg',
32             [2, 0.4]);
33     case 'positiv'
34         Schallgeschwindigkeit = @(x) SchallKruemmung(x, 'pos',
35             [2, 1.2]);
36     case 'unstetig'
37         Schallgeschwindigkeit = @(x) SchallUnstetig(x);
38     otherwise
39         display('Fehler: □Anderes □Modellproblem □wählen!');
40 end
41 if konti
42     [gitter, originalKnoten] = InitGitter(ansatz, diskParameter
43         , ~konti, Schallgeschwindigkeit);
44 else
45     [gitter, originalKnoten] = InitGitter(ansatz, diskParameter
46         , ~konti, Schallgeschwindigkeit);

```

```

43     Schallgeschwindigkeit      = @(x) BerAppSchallgeschwindigkeit(
        x, ansatz, gitter, originalKnoten);
44 end
45 messdatenGeodaeten  = BerGeodaeten(anzMessdaten.anzAufpunkte,
        anzMessdaten.anzRichtungen, Schallgeschwindigkeit, optionsDaten)
        ;
46 Metrik              = @(x) 1./Schallgeschwindigkeit(x);
47 messdatenLaufzeit   = BerLaufzeit(messdatenGeodaeten, Metrik,
        verbose);
48 end
49
50 function [c, c_x, c_y] = SchallEukl(x)
51 %Euklidische Metrik
52 n = size(x, 1);
53 c = ones(n, 1);
54 if nargin > 1
55     c_x = 0*c;
56     c_y = c_x;
57 end
58 end
59
60 function [c, c_x, c_y] = SchallPeaks(x)
61 %Konstant 1 mit lokalen Peaks mit positivem oder negativem
        Ausschlag.
62 n          = size(x,1);
63 c_x        = zeros(n,1);
64 c_y        = zeros(n,1);
65 punktTheta1 = zeros(n,1);
66 punktTheta2 = zeros(n,1);
67 punktTheta3 = zeros(n,1);
68 %Parameter
69 theta1      = 1/4;
70 theta2      = 1/5;
71 theta3      = 1/6;
72 zentrumTheta1 = [0.2 0.4];
73 zentrumTheta2 = [-1/3 -1/3];
74 zentrumTheta3 = [0.5 -0.5];
75 skalaTheta1  = 0.2;
76 skalaTheta2  = -0.15;
77 skalaTheta3  = 0.1;
78 for k = 1:n
79     punktTheta1(k) = norm(x(k,:) - zentrumTheta1);
80     punktTheta2(k) = norm(x(k,:) - zentrumTheta2);
81     punktTheta3(k) = norm(x(k,:) - zentrumTheta3);
82 end
83 diffPunktTheta1 = theta1 - punktTheta1;

```

```

84 diffPunktTheta2 = theta2 - punktTheta2;
85 diffPunktTheta3 = theta3 - punktTheta3;
86 cTheta1 = exp(-1./(abs(theta1 - punktTheta1)));
87 cTheta2 = exp(-1./(abs(theta2 - punktTheta2)));
88 cTheta3 = exp(-1./(abs(theta3 - punktTheta3)));
89 c = skalaTheta1*exp(1./theta1).*cTheta1.*(diffPunktTheta1 > 0) +
    ...
90     skalaTheta2*exp(1./theta2).*cTheta2.*(diffPunktTheta2 > 0) +
    ...
91     skalaTheta3*exp(1./theta3).*cTheta3.*(diffPunktTheta3 > 0) + 1;
92 %Bestimmung der partiellen Ableitungen
93 if nargout > 1
94     b_imZentrum = (punktTheta1 < 1e-8) | (punktTheta2 < 1e-8) | (
        punktTheta3 < 1e-8);
95     c_x(b_imZentrum) = 0;
96     c_y(b_imZentrum) = 0;
97     b_nichtZentrum = ~b_imZentrum;
98     divTheta1 = punktTheta1(b_nichtZentrum).*diffPunktTheta1(
        b_nichtZentrum).^2;
99     divTheta2 = punktTheta2(b_nichtZentrum).*diffPunktTheta2(
        b_nichtZentrum).^2;
100    divTheta3 = punktTheta3(b_nichtZentrum).*diffPunktTheta3(
        b_nichtZentrum).^2;
101    if any(b_nichtZentrum)
102        c_x(b_nichtZentrum) = (-(x(b_nichtZentrum,1) -
            zentrumTheta1(1))./divTheta1.*(diffPunktTheta1(
                b_nichtZentrum) > 0) ...
103                -(x(b_nichtZentrum,1) - zentrumTheta2(1))./
                    divTheta2.*(diffPunktTheta2(
                        b_nichtZentrum) > 0) ...
104                -(x(b_nichtZentrum,1) - zentrumTheta3(1))./
                    divTheta3.*(diffPunktTheta3(
                        b_nichtZentrum) > 0)).*(c(b_nichtZentrum)
                        -1);
105        c_y(b_nichtZentrum) = (-(x(b_nichtZentrum,2) -
            zentrumTheta1(2))./divTheta1.*(diffPunktTheta1(
                b_nichtZentrum) > 0) ...
106                -(x(b_nichtZentrum,2) - zentrumTheta2(2))./
                    divTheta2.*(diffPunktTheta2(
                        b_nichtZentrum) > 0) ...
107                -(x(b_nichtZentrum,2) - zentrumTheta3(2))./
                    divTheta3.*(diffPunktTheta3(
                        b_nichtZentrum) > 0)).*(c(b_nichtZentrum)
                        -1);
108    end
109 end

```

```

110 end
111
112 function [c, c_x, c_y] = SchallKreis(x)
113 %Erzeugt eine Cosinus-Welle um [0, 0]. Inneren befinden sich Peaks.
114 n          = size(x,1);
115 c_x        = zeros(n,1);
116 c_y        = zeros(n,1);
117 punktCosinus = zeros(n,1);
118 punktTheta1 = zeros(n,1);
119 punktTheta2 = zeros(n,1);
120 punktTheta3 = zeros(n,1);
121 %Parameter
122 %Cosinuswelle (es wird jeweils die vorletzte Welle genommen)
123 dicke      = 1/10;
124 zentrumCosinus = [0 0];
125 amplitude  = 0.005;
126 %Peaks
127 theta1     = 1/2;
128 theta2     = 1/3;
129 theta3     = 1/6;
130 zentrumTheta1 = [0.2 0.4];
131 zentrumTheta2 = [-0.2 -0.3];
132 zentrumTheta3 = [0.1 -0.2];
133 skalaTheta1  = 0.2;
134 skalaTheta2  = -0.15;
135 skalaTheta3  = 0.10;
136 for k = 1:n
137     punktCosinus(k) = norm(x(k,:) - zentrumCosinus);
138     punktTheta1(k) = norm(x(k,:) - zentrumTheta1);
139     punktTheta2(k) = norm(x(k,:) - zentrumTheta2);
140     punktTheta3(k) = norm(x(k,:) - zentrumTheta3);
141 end
142 diffPunktTheta1 = theta1 - punktTheta1;
143 diffPunktTheta2 = theta2 - punktTheta2;
144 diffPunktTheta3 = theta3 - punktTheta3;
145 cTheta1 = exp(-1./(abs(theta1 - punktTheta1)));
146 cTheta2 = exp(-1./(abs(theta2 - punktTheta2)));
147 cTheta3 = exp(-1./(abs(theta3 - punktTheta3)));
148 c = -amplitude*(cos(pi*punktCosinus/dicke) - 1).*(punktCosinus < 1
149     - 2*dicke).*(1 - 4*dicke < punktCosinus) + ...
150     skalaTheta1*exp(1./theta1).*cTheta1.*(diffPunktTheta1 > 0) +
151     ...
152     skalaTheta2*exp(1./theta2).*cTheta2.*(diffPunktTheta2 > 0) +
153     ...
154     skalaTheta3*exp(1./theta3).*cTheta3.*(diffPunktTheta3 > 0) + 1;
155 %Bestimmung der partiellen Ableitungen

```

```

153 if nargout > 1
154     b_imZentrum = (punktTheta1 < 1e-6) | (punktTheta2 < 1e-6) | (
        punktTheta3 < 1e-6) | (punktCosinus < 1e-6);
155     c_x(b_imZentrum) = 0;
156     c_y(b_imZentrum) = 0;
157     b_nichtZentrum = ~b_imZentrum;
158     divTheta1 = punktTheta1(b_nichtZentrum).*diffPunktTheta1(
        b_nichtZentrum).^2;
159
160     divTheta2 = punktTheta2(b_nichtZentrum).*diffPunktTheta2(
        b_nichtZentrum).^2;
161     divTheta3 = punktTheta3(b_nichtZentrum).*diffPunktTheta3(
        b_nichtZentrum).^2;
162     if any(b_nichtZentrum)
163         c_x(b_nichtZentrum) = amplitude*(x(b_nichtZentrum,1) -
            zentrumCosinus(1))*pi/dicke./punktCosinus(b_nichtZentrum
            )*.
164             sin(pi*punktCosinus(b_nichtZentrum)/dicke)
                .*(punktCosinus(b_nichtZentrum) < 1 - 2*
                dicke).
165             (1 - 4*dicke < punktCosinus(b_nichtZentrum)
                ) + ...
166             (-(x(b_nichtZentrum,1) - zentrumTheta1(1))
                ./divTheta1.*(diffPunktTheta1(
                b_nichtZentrum) > 0).*(abs(skalaTheta1)
                > 0)) ...
167             -(x(b_nichtZentrum,1) - zentrumTheta2(1))./
                divTheta2.*(diffPunktTheta2(
                b_nichtZentrum) > 0).*(abs(skalaTheta2)
                > 0)) ...
168             -(x(b_nichtZentrum,1) - zentrumTheta3(1))./
                divTheta3.*(diffPunktTheta3(
                b_nichtZentrum) > 0).*(abs(skalaTheta3)
                > 0)).* ...
169             (c(b_nichtZentrum) - 1);
170         c_y(b_nichtZentrum) = amplitude*(x(b_nichtZentrum,2) -
            zentrumCosinus(2))*pi/dicke./punktCosinus(b_nichtZentrum
            )*.
171             sin(pi*punktCosinus(b_nichtZentrum)/dicke)
                .*(punktCosinus(b_nichtZentrum) < 1 - 2*
                dicke).
172             (1 - 4*dicke < punktCosinus(b_nichtZentrum)
                ) + ...
173             (-(x(b_nichtZentrum,2) - zentrumTheta1(2))
                ./divTheta1.*(diffPunktTheta1(
                b_nichtZentrum) > 0).*(abs(skalaTheta1)

```

```

> 0) ...
174     -(x(b_nichtZentrum,2) - zentrumTheta2(2))./
        divTheta2.*(diffPunktTheta2(
            b_nichtZentrum) > 0).*(abs(skalaTheta2)
            > 0) ...
175     -(x(b_nichtZentrum,2) - zentrumTheta3(2))./
        divTheta3.*(diffPunktTheta3(
            b_nichtZentrum) > 0).*(abs(skalaTheta3)
            > 0)).* ...
176     (c(b_nichtZentrum)-1);
177     end
178 end
179 end
180
181 function [c, c_x, c_y] = SchallKruemmung(Punkte, kruemmung,
        parameterSchall)
182 %SchallKruemmung gibt eine Schallgeschwindigkeit mit positiver
183 %oder negativer Kruemmung an den Orten A(i,:) zurueck,
184 %zusaetzlich die partiellen Ableitungen.
185 %Parameter: parameterSchall = [R, a] (a^2 < R^2).
186 %Ausgabe: c - Schallgeschwindigkeit, c_x, c_y Ableitungen in
187 %Richtung x und y.
188 if nargin == 1 %Euklidische Metrik
189     kruemmung = 'pos';
190     R = 2.;
191     a = 0.;
192 end
193
194 R = parameterSchall(1);
195 a = parameterSchall(2);
196
197 if kruemmung == 'neg'
198     rr      = zeros(numel(Punkte(:,1)),1);
199     rr(:)   = Punkte(:,1).^2 + Punkte(:,2).^2;
200     RR      = R*R;
201     aa      = a*a;
202     RRR     = RR + aa.*rr;
203     c       = RRR.^2 ./ (4*RR) + (1-(RR + aa)^2/(4*RR));
204     %Bestimmung der partiellen Ableitungen
205     if nargin > 1
206         c_x = aa*RRR.*Punkte(:,1) ./ RR;
207         c_y = aa*RRR.*Punkte(:,2) ./ RR;
208     end
209 elseif kruemmung == 'pos'
210     rr      = zeros(numel(Punkte(:,1)),1);
211     rr(:)   = Punkte(:,1).^2 + Punkte(:,2).^2;

```

```

212     RR      = R*R;
213     aa      = a*a;
214     RRR     = RR - aa.*rr;
215     c       = RRR.^2 ./ (4*RR) + (1-(RR - aa)^2/(4*RR));
216     %Bestimmung der partiellen Ableitungen
217     if nargout > 1
218         c_x = -aa*RRR.*Punkte(:,1) ./ RR;
219         c_y = -aa*RRR.*Punkte(:,2) ./ RR;
220     end
221 else
222     fprintf('\nBitte <<pos>> oder <<neg>> als Parameter übergeben.\n');
223     return;
224 end
225 for i = 1:size(Punkte,1)
226     if (norm(Punkte(i,:))>1)
227         c(i) = 1;
228         c_x(i) = 0;
229         c_y(i) = 0;
230     end
231 end
232 end
233
234 function [c, c_x, c_y] = SchallUnstetig(Punkte)
235 %SchallUnstetig gibt eine unstetige Schallgeschwindigkeit an
236 %den Orten Punkte(i,:) zurück, zusätzlich die partiellen
237 %Ableitungen.
238 %Ausgabe: c - Schallgeschwindigkeit, c_x, c_y Ableitungen
239 %in Richtung x und y.
240 n = size(Punkte,1);
241 c = ones(n,1);
242 for i = 1:n
243     if (Punkte(i,2) > 0) && (norm(Punkte(i,:)) < 1)
244         c(i) = 1.5;
245     end
246 end
247 c_x = zeros(n,1);
248 c_y = c_x;
249 end

```


Literaturverzeichnis

- [Anikonov und Romanov 1998] ANIKONOV, Yu E. ; ROMANOV, VG: On uniqueness of determination of a form of first degree by its integrals along geodesics. In: *JOURNAL OF INVERSE AND ILL POSED PROBLEMS* 5 (1998), S. 487–490
- [Beylkin 1984] BEYLKIN, Gregory: The inversion problem and applications of the generalized radon transform. In: *Communications on Pure and Applied Mathematics* 37 (1984), Nr. 5, S. 579–599. – URL <http://dx.doi.org/10.1002/cpa.3160370503>. – ISSN 1097-0312
- [Croke u. a. 2000] CROKE, Christopher ; DAIRBEKOV, Nurlan ; SHARAFUTDINOV, Vladimir: Local boundary rigidity of a compact Riemannian manifold with curvature bounded above. In: *Transactions of the American Mathematical Society* 352 (2000), Nr. 9, S. 3937–3956
- [Croke 1990] CROKE, Christopher B.: Rigidity for surfaces of non-positive curvature. In: *Commentarii Mathematici Helvetici* 65 (1990), Nr. 1, S. 150–169. – URL <http://link.springer.com/content/pdf/10.1007/BF02566599.pdf>
- [Croke 1991] CROKE, Christopher B.: Rigidity and the distance between boundary points. In: *J. Differential Geom* 33 (1991), Nr. 2, S. 445–464. – URL <http://intlpress.cn/journals/JDG/archive/pdf/1991/33-2-445.pdf>
- [Croke 2004] CROKE, Christopher B.: Rigidity theorems in Riemannian geometry. In: *Geometric methods in inverse problems and PDE control*. Springer, 2004, S. 47–72. – URL <http://web-hans.math.upenn.edu/~ccroke/dvi-papers/rigidity.pdf>
- [Daubechies u. a. 2004] DAUBECHIES, Ingrid ; DEFRISE, Michel ; DE MOL, Christine: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. In: *Communications on pure and applied mathematics* 57 (2004), Nr. 11, S. 1413–1457
- [Demtröder 2005] DEMTRÖDER, Wolfgang: *Experimentalphysik*. Springer, 2005
- [Do Carmo 1976] DO CARMO, Manfredo P.: *Differential geometry of curves and surfaces*. Bd. 2. Prentice-Hall Englewood Cliffs, 1976
- [Greenleaf u. a. 2003] GREENLEAF, Allan ; LASSAS, Matti ; UHLMANN, Gunther: On nonuniqueness for Calderon’s inverse problem. In: *Mathematical Research Letters* 10 (2003), Nr. 5/6, S. 685–694. – URL <http://www.rni.helsinki.fi/~mjl/nonuniqu.pdf>
- [Gromov u. a. 1983] GROMOV, Mikhael u. a.: Filling riemannian manifolds. In: *Journal of Differential Geometry* 18 (1983), Nr. 1, S. 1–147

- [Guillemin und Sternberg 1990] GUILLEMIN, Victor ; STERNBERG, Shlomo: *Geometric asymptotics*. American Mathematical Soc., 1990 (14)
- [Herglotz 1905] HERGLOTZ, G.: Über die Elastizität der Erde bei Berücksichtigung ihrer variablen Dichte. In: *Zeitschr. für Math. Phys* 52 (1905), S. 275–299
- [Heuser 2009] HEUSER, Harro: *Gewöhnliche Differentialgleichungen: Einführung in Lehre und Gebrauch*. Springer, 2009
- [Hinze u. a. 2009] HINZE, Michael ; PINNAU, Rene ; ULBRICH, Michael ; ULBRICH, Stefan: *Optimization with PDE constraints, volume 23 of Mathematical Modelling: Theory and Applications*. 2009
- [Hofmann u. a. 2007] HOFMANN, Bernd ; KALTENBACHER, Barbara ; POESCHL, Christiane ; SCHERZER, Otmar: A convergence rates result for Tikhonov regularization in Banach spaces with non-smooth operators. In: *Inverse Problems* 23 (2007), Nr. 3, S. 987
- [Jänich 1992] JÄNICH, Klaus: *Vektoranalysis*. Bd. 5. Springer, 1992
- [Jost 2008] JOST, Jürgen: *Riemannian geometry and geometric analysis*. Bd. 42005. Springer, 2008
- [Katsevich und Schuster 2013] KATSEVICH, A. ; SCHUSTER, T.: An exact inversion formula for cone beam vector tomography. In: *Inverse Problems* (2013)
- [Kühnel 2013] KÜHNEL, Wolfgang: *Differentialgeometrie: Kurven-Flächen-Mannigfaltigkeiten*. Springer DE, 2013
- [Kirsch und Rieder 2014] KIRSCH, Andreas ; RIEDER, Andreas: Seismic Tomography is Locally Ill-Posed. In: *Inverse Problems* 30 125001 (2014). – URL http://www.math.kit.edu/ianm3/~rieder/seite/publikationen/media/note_on_ill-posedness.pdf
- [Kurylev u. a. 2010] KURYLEV, Yaroslav ; LASSAS, Matti ; UHLMANN, Gunther: Rigidity of broken geodesic flow and inverse problems. In: *American journal of mathematics* (2010). – URL <http://arxiv.org/pdf/math/0703518>
- [Lade u. a. 2005] LADE, S. J. ; PAGANIN, David ; MORGAN, M. J.: 3-D Vector tomography of Doppler-transformed fields by filtered-backprojection. In: *Optics communications* 253 (2005), Nr. 4, S. 382–391
- [Lange u. a. 2000] LANGE, Kenneth ; HUNTER, David R. ; YANG, Ilsoon: Optimization transfer using surrogate objective functions. In: *Journal of computational and graphical statistics* 9 (2000), Nr. 1, S. 1–20
- [Lassas u. a. 2011] LASSAS, Matti ; SALO, Mikko ; UHLMANN, Gunther: Wave Phenomena. In: *Handbook of Mathematical Methods in Imaging* (2011), S. 867–909. – URL <http://arxiv.org/pdf/math-ph/0703059>

- [Lassas u. a. 2003] LASSAS, Matti ; SHARAFUTDINOV, Vladimir ; UHLMANN, Gunther: Semiglobal boundary rigidity for Riemannian metrics. In: *Mathematische Annalen* 325 (2003), Nr. 4, S. 767–793. – URL <http://www.math.nsc.ru/~sharafutdinov/files/articles/semiglobal.pdf>
- [Louis 1989] LOUIS, A. K.: Inverse und schlecht gestellte Probleme. In: *Teubner, Stuttgart* (1989)
- [Louis 1996] LOUIS, A. K.: Approximate inverse for linear and some nonlinear problems. In: *Inverse problems* 12 (1996), Nr. 2, S. 175
- [Michel 1981] MICHEL, René: Sur la rigidité imposée par la longueur des géodésiques. In: *Inventiones mathematicae* 65 (1981), Nr. 1, S. 71–83
- [Mukhometov 1975] MUKHOMETOV, R. G.: Inverse kinematic problem of seismic on the plane. In: *Math. Problems of Geophysics. Akad. Nauk. SSSR, Sibirsk. Otdel., Vychisl. Tsentr, Novosibirsk* 6 (1975), S. 243–252
- [Mukhometov 1977] MUKHOMETOV, R. G.: The problem of reconstructing a two-dimensional Riemannian metric and integral geometry. In: *Soviet Math. Dokl.* 18 (1977), S. 32–35
- [Mukhometov 1981] MUKHOMETOV, R. G.: A problem of reconstructing a Riemannian metric. In: *Siberian Mathematical Journal* 22 (1981), Nr. 3, S. 420–433
- [Norton 1988] NORTON, Stephen J.: Tomographic reconstruction of 2-D vector fields: application to flow imaging. In: *Geophysical Journal International* 97 (1988), Nr. 1, S. 161–168. – URL <http://gji.oxfordjournals.org/content/97/1/161.full.pdf>
- [Osman und Prince 1998] OSMAN, Nael F. ; PRINCE, Jerry L.: 3D vector tomography on bounded domains. In: *Inverse problems* 14 (1998), Nr. 1, S. 185
- [Otal 1990] OTAL, Jean-Pierre: Sur les longueurs des géodésiques d’une métrique à courbure négative dans le disque. In: *Commentarii Mathematici Helvetici* 65 (1990), Nr. 1, S. 334–347. – URL <http://link.springer.com/content/pdf/10.1007%2FBF02566611.pdf>
- [Paternain u. a. 2013] PATERNAIN, Gabriel P. ; SALO, Mikko ; UHLMANN, Gunther: Tensor tomography: progress and challenges. In: *arXiv preprint arXiv:1303.6114* (2013). – URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.309.9074&rep=rep1&type=pdf>
- [Pestov und Uhlmann 2004] PESTOV, Leonid ; UHLMANN, Gunther: On characterization of the range and inversion formulas for the geodesic X-ray transform. In: *International Mathematics Research Notices* 2004 (2004), Nr. 80, S. 4331–4347. – URL <http://imrn.oxfordjournals.org/content/2004/80/4331.full.pdf>
- [Pestov und Uhlmann 2005] PESTOV, Leonid ; UHLMANN, Gunther: Two dimensional compact simple Riemannian manifolds are boundary distance rigid. In: *Annals of mathematics* (2005), S. 1093–1110. – URL <http://www.jstor.org/stable/3597325>
- [Pfitzenreiter und Schuster 2011] PFITZENREITER, Tim ; SCHUSTER, Thomas: Tomographic reconstruction of the curl and divergence of 2D vector fields taking refractions into account. In: *SIAM Journal on Imaging Sciences* 4 (2011), Nr. 1, S. 40–56

- [Rieder 2003] RIEDER, Andreas: *Keine Probleme mit Inversen Problemen*. Bd. 1. Vieweg Wiesbaden, 2003
- [Romanov 1967] ROMANOV, Vladimir G.: Reconstructing a function by means of integrals along a family of curves. In: *Sibirsk. Mat. Zh* 8 (1967), Nr. 5, S. 1206–1208
- [Romanov 1978] ROMANOV, Vladimir G.: Integral Geometry on Geodesics of Isotropic Riemannian Metric. In: *DOKLADY AKADEMII NAUK SSSR* 241 (1978), Nr. 2, S. 290–293
- [Romanov 1987] ROMANOV, Vladimir G.: *Inverse problems of mathematical physics*. BRILL, 1987
- [Romanov 1999] ROMANOV, Vladimir G.: A stability estimate in the inverse problem of determining the speed of sound. In: *Sibirskii Matematicheskii Zhurnal* 40 (1999), Nr. 6, S. 1323–1338. – URL <http://link.springer.com/content/pdf/10.1007/BF02677536.pdf>
- [Salo u. a. 2011] SALO, Mikko ; UHLMANN, Gunther u. a.: The attenuated ray transform on simple surfaces. In: *J. Diff. Geom* 88 (2011), Nr. 1, S. 161–187. – URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.309.9372&rep=rep1&type=pdf>
- [Scheck 2006] SCHECK, Florian: *Theoretische Physik 3*. Springer, 2006. – URL <http://link.springer.com/book/10.1007%2F3-540-29478-3>
- [Schuster u. a. 2012] SCHUSTER, Thomas ; KALTENBACHER, Barbara ; HOFMANN, Bernd ; KAZIMIERSKI, Kamil S.: *Regularization methods in Banach spaces*. Bd. 10. Walter de Gruyter, 2012
- [Schuster u. a. 2008] SCHUSTER, Thomas ; THEIS, David ; LOUIS, Alfred K.: A reconstruction approach for imaging in 3D cone beam vector field tomography. In: *Journal of Biomedical Imaging* 2008 (2008), S. 12
- [Shanno 1970] SHANNO, David F.: Conditioning of quasi-Newton methods for function minimization. In: *Mathematics of computation* 24 (1970), Nr. 111, S. 647–656
- [Sharafutdinov u. a. 2000] SHARAFUTDINOV, Vladimir ; UHLMANN, Gunther u. a.: On deformation boundary rigidity and spectral rigidity of Riemannian surfaces with no focal points. In: *Journal of Differential Geometry* 56 (2000), Nr. 1, S. 93–110. – URL <ftp://ftp.math.ethz.ch/EMIS/journals/NYJM/nyjm/jdg/p/2000/56-1-3.pdf>
- [Sharafutdinov 1994] SHARAFUTDINOV, Vladimir A.: *Integral geometry of tensor fields*. Bd. 1. Walter de Gruyter, 1994. – URL <http://www.math.nsc.ru/~sharafutdinov/files/book.pdf>
- [Sharafutdinov 2004] SHARAFUTDINOV, Vladimir A.: Ray transform on Riemannian manifolds. In: *New analytic and geometric methods in inverse problems*. Springer, 2004, S. 187–259
- [Stefanov und Uhlmann 1998] STEFANOV, Plamen ; UHLMANN, Gunther: Rigidity for metrics with the same lengths of geodesics. In: *Mathematical Research Letters* 5 (1998), S. 83–96

- [Stefanov und Uhlmann 2005a] STEFANOV, Plamen ; UHLMANN, Gunther: Boundary rigidity and stability for generic simple metrics. In: *Journal of the American Mathematical Society* 18 (2005), Nr. 4, S. 975–1003. – URL <http://www.ams.org/jams/2005-18-04/S0894-0347-05-00494-7/S0894-0347-05-00494-7.pdf>
- [Stefanov und Uhlmann 2005b] STEFANOV, Plamen ; UHLMANN, Gunther: Recent progress on the boundary rigidity problem. In: *Electronic research announcements of the American Mathematical Society* 11 (2005), Nr. 8, S. 64–70. – URL <http://www.ams.org/era/2005-11-08/S1079-6762-05-00148-4/S1079-6762-05-00148-4.pdf>
- [Svetov u. a. 2014] SVETOV, I.E. ; DEREVTSOV, E.Yu. ; VOLKOV, Yu.S. ; SCHUSTER, T.: A numerical solver based on B-splines for 2D vector field tomography in a refracting medium. In: *Mathematics and Computers in Simulation* 97 (2014), Nr. 0, S. 207 – 223. – URL <http://www.sciencedirect.com/science/article/pii/S0378475413002346>. – ISSN 0378-4754
- [Walter 1990] WALTER, Wolfgang: *Gewöhnliche Differentialgleichungen*. Bd. 7. Springer Berlin, 1990
- [Wernsdorfer 1993] WERNSDÖRFER, A: Complete reconstruction of three-dimensional vector fields. In: *Proc. ECAPT 93 (Karlsruhe)* (1993)
- [Wiechert und Zoeppritz 1907] WIECHERT, Emil ; ZOEPPRITZ, Karl: Über Erdbebenwellen. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1907 (1907), S. 415