

Projektionslose Approximation des harmonischen Wärmeflusses für S^2 -wertige Vektorfelder

Bachelorarbeit im Studiengang
Bachelor of Science Mathematik
an der Universität Tübingen

vorgelegt von
Benjamin Kiefer
3870596
ben93_kiefer@yahoo.de

betreut durch
Prof. Dr. Christian Lubich

2017

Inhaltsverzeichnis

1	Einführung	1
1.1	Idee	2
2	Theoretische Grundlagen	4
2.1	Finite Elemente-Räume	4
2.2	Differenzenquotienten	6
2.3	Diskrete Einschränkung	6
3	Harmonischer Wärmefluss	8
3.1	Eigenschaften der schwachen Formulierung	8
3.2	Algorithmus zum harmonischen Wärmefluss	11
4	Numerische Implementation	14
4.1	Aufstellung des Gleichungssystems	15
4.2	Auswertung	18
5	Matlab-Anlagen	21

1 Einführung

Der harmonische Wärmefluss ist das typische Beispiel einer parabolischen partiellen Differentialgleichung. Zusammen mit der geometrischen Einschränkung, dass die Lösung des Wärmeflusses nur Werte auf der Einheitssphäre annehmen darf, ist diese Art der nichtlinearen Differentialgleichung besonders interessant. Wir wollen im Folgenden ein Verfahren entwickeln, welches stabil und konvergent ist und nur mit linearen Gleichungssystemen auskommt und anschließend die numerische Implementierung vornehmen. Hierbei lehnt sich diese Arbeit an das Paper [Bartels, 2016] an.

Für die Herleitung des harmonischen Wärmeflusses betrachten wir zunächst reellwertige harmonische Funktionen. Diese sind Extremstellen des Dirichlet-Funktional

$$I(u) = \frac{1}{2} \int_{\Omega} |\nabla u|^2 dx$$

über alle Vektorfelder $u : \Omega \rightarrow \mathbb{R}^m$ mit der Einheitssphärenbeschränkung $|u(x)| = 1$ für fast alle $x \in \Omega \subset \mathbb{R}^d$ mit der Dirichlet-Randbedingung $u|_{\Gamma_D} = u_D$. Mittels der Euler-Lagrange-Gleichungen für das Dirichlet-Funktional bekommen wir eine Charakterisierung der Extremstellen desselbigen:

$$\begin{aligned} \Delta u &= |\nabla u|^2 u, & u|_{\Gamma_D} &= u_D, \\ |u|^2 &= 1, & \partial_\nu u|_{\Gamma_N} &= 0, \end{aligned}$$

wobei $\Gamma_N = \partial\Omega \setminus \Gamma_D$ und $\partial_\nu = \nabla u \cdot \nu$ die äußere Normalenableitung auf $\partial\Omega$ ist. Durch Gradientenflüsse finden wir Lösungen dieser Euler-Lagrange-Gleichungen. Wir beschränken uns auf den einfachsten Fall des L^2 -Gradientenflusses, welcher folgendermaßen gegeben ist:

$$\begin{aligned} \partial_t u - \Delta u &= |\nabla u|^2 u, & u(t, \cdot)|_{\Gamma_D} &= u_D, & u(0) &= u_0. \\ |u|^2 &= 1, & \partial_\nu u(t, \cdot)|_{\Gamma_N} &= 0, \end{aligned}$$

Diese zeitabhängige partielle Differentialgleichung heißt harmonischer Wärmefluss. Für die schwache Formulierung dieses Problems existiert eine Lösung und für $d = 2$ ist die Eindeutigkeit dieser bekannt.

Die Einheitssphärenbeschränkung erlaubt nur Lösungen auf der Einheitssphäre. In diversen anderen Anwendungen verlangt man jedoch Einschränkungen auf andere Zielmannigfaltigkeiten. Diese werden durch eine Funktion $g : \mathbb{R}^m \rightarrow \mathbb{R}$ mit $g(u(x)) = 0$ für fast alle $x \in \Omega$ für zeitunabhängige Lösungen u oder durch $g(u(t, x)) = 0$ für fast alle $(t, x) \in [0, T] \times \Omega$ für zeitabhängige u gekennzeichnet. Wir schreiben dafür kurz $g(u) = 0$.

Die heuristische Herangehensweise ist, die Einheitssphärenbeschränkung an den Knoten der zugrundeliegenden Triangulierung zu fordern. Dieser Ansatz benötigt die schwach-spitze Eigenschaft der zugrundeliegenden Triangulierung ([Bartels et al., 2015]). Hier wollen wir eine Implementierung des Wärmeflusses vornehmen, die ohne Projektionsschritt, also ohne die Einheitssphärenbeschränkung an den Knoten auskommt und werden zeigen, dass der zusätzliche Fehler, der ohne diese Projektion entsteht, durch die Schrittweite kontrolliert wird und unabhängig von der Anzahl der Schritte beschränkt bleibt. Dabei implementieren wir das Problem für $d = 2$ und $m = 3$.

Zunächst skizzieren wir die Idee anhand der Zeitdiskretisierung des Wärmeflusses und zeigen damit gleich, dass die Approximationslösung die Einschränkung im Grenzwert erfüllt und formulieren danach das theoretische Fundament.

1.1 Idee

Der Einfachheit halber sei $\Gamma_D = \emptyset$. Aufgrund der Einheitslängen-Beschränkung gilt, dass $\partial_t u \cdot u = 0$ und falls u^{k-1} eine Approximation an $u(t_{k-1})$ und v^k an $\partial_t u(t_k)$ ist, fordern wir die Orthogonalitätsbeziehung

$$v^k \cdot u^{k-1} = 0.$$

Damit berechnen wir für ein bereits gegebenes u^{k-1} das eindeutige $v^k \in H^1(\Omega; \mathbb{R}^m)$ mit $v^k \cdot u^{k-1} = 0$ fast überall in Ω und

$$(v^k, w) + (\nabla[u^{k-1} + \tau v^k], \nabla w) = (|\nabla u^k|^2 u^{k-1}, w) = 0 \quad (1)$$

für alle $w \in H^1(\Omega; \mathbb{R}^m)$ mit $u^{k-1} \cdot w = 0$ fast überall in Ω , wobei $\tau > 0$ eine Zeitschrittweite und (\cdot, \cdot) das L^2 -Skalarprodukt ist. Diese schwache Formulierung erhalten wir direkt mit der Variationsrechnung. Anschließend iterieren wir

$$u^k = u^{k-1} + \tau v^k.$$

Nun erfüllt u^k im Allgemeinen nicht die Einheitssphärenbeschränkung. Im Folgenden zeigen wir jedoch, dass diese Beschränkung im Grenzwert $\tau \rightarrow 0$ erfüllt ist. Es folgt mit der punktweisen Orthogonalität von u^{k-1} und v^k für alle $l \geq 1$

$$\begin{aligned} |u^l|^2 &= |(u^l)^2| = |(u^{l-1} + \tau v^l)^2| = |(u^{l-1})^2 + 2u^{l-1}\tau v^l + \tau^2(v^l)^2| = |(u^{l-1})^2 + \tau^2(v^l)^2| \\ &= |u^{l-1}|^2 + \tau^2|v^l|^2. \end{aligned}$$

Induktiv folgt somit

$$|u^l|^2 = 1 + \tau^2 \sum_{k=1}^l |v^k|^2$$

mit der Anfangsbedingung $|u^0| = 1$ fast überall in Ω . Nehmen wir die L^1 -Norm auf beiden Seiten, dann folgt mit der Dreiecksungleichung

$$\| |u^l|^2 - 1 \|_{L^1(\Omega)} = \left\| \tau^2 \sum_{k=1}^l |v^k|^2 \right\|_{L^1(\Omega)} \leq \tau^2 \sum_{k=1}^l \| (v^k)^2 \|_{L^1(\Omega)} = \tau^2 \sum_{k=1}^l \| v^k \|_{L^2(\Omega)}^2. \quad (2)$$

Nun wählen wir als Testfunktion $w = v^k$ in der schwachen Formulierung (1) und mit der Binomischen Formel $2(a+b)a = 2a^2 + 2ba = a^2 + a^2 + 2ab + b^2 - b^2 = a^2 + (a+b)^2 - b^2$ folgt, wie wir später genauer sehen werden,

$$\|v^k\|^2 + \frac{\tau}{2} \|\nabla v^k\|^2 + \frac{1}{2\tau} (\|\nabla u^k\|^2 - \|\nabla u^{k-1}\|^2) = 0.$$

Nun folgt nach Multiplikation mit τ und Summation über $k = 1, 2, \dots, l$

$$\tau \sum_{k=1}^l \|v^k\|^2 + \frac{\tau^2}{2} \sum_{k=1}^l \|\nabla v^k\|^2 + \frac{1}{2} \sum_{k=1}^l (\|\nabla u^k\|^2 - \|\nabla u^{k-1}\|^2) = 0.$$

Die letzte Summe ist eine Teleskopsumme und nach Umstellen erhalten wir

$$\tau \sum_{k=1}^l \|v^k\|^2 = \frac{1}{2} \|\nabla u^0\|^2 - \frac{\tau^2}{2} \sum_{k=1}^l \|\nabla v^k\|^2 - \frac{1}{2} \|\nabla u^l\|^2.$$

Setzen wir dies in (2) ein, folgt

$$\begin{aligned} \| |u^l|^2 - 1 \|_{L^1(\Omega)} &\leq \tau^2 \sum_{k=1}^l \|v^k\|_{L^2(\Omega)}^2 \\ &= \tau \left(\frac{1}{2} \|\nabla u^0\|^2 - \frac{\tau^2}{2} \sum_{k=1}^l \|\nabla v^k\|^2 - \frac{1}{2} \|\nabla u^l\|^2 \right) \leq \tau \frac{1}{2} \|\nabla u^0\|^2 = \tau I(u^0), \end{aligned}$$

wobei hier I wieder das Dirichlet-Funktional ist. Für $\tau \rightarrow 0$ ist die Einheitssphärenbeschränkung erfüllt und der Approximationsfehler ist unabhängig von der Anzahl der Zeitschritte beschränkt. In der folgenden Grafik ist das numerische Schema illustriert.

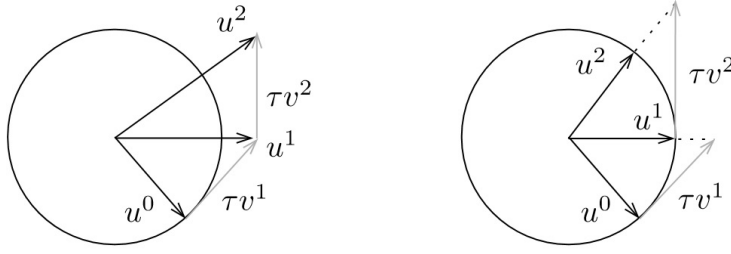


ABBILDUNG 1: (aus [Bartels, 2016]) Im linken Plot sieht man, dass die Einheitskugelbeschränkung verletzt ist. Der Fehler ist jedoch unabhängig von der Anzahl der Zeitschritte beschränkt und wird durch die Schrittweite kontrolliert. Im rechten Plot wird nach jeder Iteration ein Projektionsschritt vorgenommen, damit die Einschränkung in jedem Schritt erfüllt ist.

2 Theoretische Grundlagen

2.1 Finite Elemente-Räume

Wir betrachten das beschränkte, polyedrische Lipschitz-Gebiet (beschränktes Gebiet mit Lipschitz-Rand $\Omega \subset \mathbb{R}^d$, wobei hier $d = 2, 3$ zugelassen wird). Weiterhin sei \mathcal{T}_h eine reguläre Triangulierung von Ω in Dreiecke bzw. Tetraeder. Regulär bedeutet, dass es eine Konstante $c > 0$ gibt, sodass

$$\sup_{h>0} \sup_{T \in \mathcal{T}_h} h_T / \rho_T \leq c,$$

wobei h_T der Durchmesser eines finites Elements $T \in \mathcal{T}_h$ (maximaler Abstand zweier Punkte) ist und ρ_T den Innenkreisradius eines finiten Elements T bezeichnet. Anschaulich bedeutet Regularität, dass kein finites Element beliebig flach sein darf (s. Abbildung 3). Für ein nichtleeres, abgeschlossenes Randstück $\Gamma_D \subset \partial\Omega$ fordern wir, dass es genau bei den Eckpunkten bzw. Eckkanten der finiten Elemente aufhört. Die Menge der Knoten der Triangulierung bezeichnen wir mit \mathcal{N}_h und die Basisfunktionen mit $(\varphi_z \in \mathcal{S}^1(\mathcal{T}_h) : z \in \mathcal{N}_h)$ des Raums

$$\mathcal{S}^1(\mathcal{T}_h) = \{w_h \in C(\bar{\Omega}) : w_h|_T \text{ affin für alle } T \in \mathcal{T}_h\}$$

aller stetigen und elementweise affine Funktionen. Für $\Gamma_D \subset \partial\Omega$ bezeichnen wir mit $H_D^1(\Omega) = \{v \in H^1 : v|_D = 0\}$ den Sobolevraum, dessen Funktionen auf dem Rand verschwinden. Wir setzen schließlich

$$\mathcal{S}_D^1(\mathcal{T}_h) = \mathcal{S}^1(\mathcal{T}_h) \cap H_D^1(\Omega).$$

Der Knoten-Interpolationsoperator $\mathcal{I}_h : C(\bar{\Omega}) \rightarrow \mathcal{S}^1(\mathcal{T}_h)$ ist für eine Funktion $v \in C(\bar{\Omega})$ definiert durch $\mathcal{I}_h v = \sum_{z \in \mathcal{N}_h} v(z) \varphi_z$. Außerdem sei h bis auf eine Konstante eine obere

Schranke für die maximale Gitterweite: $\max_{T \in \mathcal{T}_h} h_T \leq ch$. Mit dieser Notation betrachten wir im Folgenden immer eine Folge von Triangulierungen $(\mathcal{T}_h)_{h>0}$, wobei $h \rightarrow 0$ gilt. Die kleinste Gitterweite in einer bestehenden Triangulierung bezeichnen wir mit $h_{\min} = \min_{T \in \mathcal{T}_h} h_T$. Der Durchmesser des Trägers $w_z = \text{supp } \varphi_z = \overline{\{x \in \Omega : \varphi_z(x) \neq 0\}}$ einer Basisfunktion $\varphi_z \in \mathcal{S}^1(\mathcal{T}_h)$ für alle z nennen wir h_z .

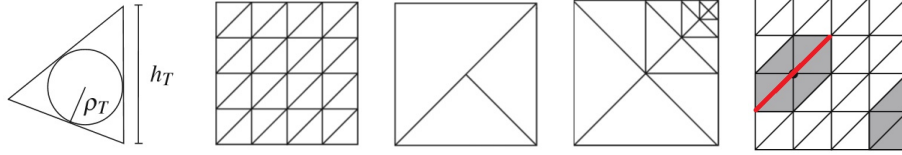


ABBILDUNG 2: (aus [Bartels et al., 2015]) Von links nach rechts sind zu sehen: Ein finites Element mit Durchmesser h_T und Inkreisradius ρ_T , eine regelkonforme Triangulierung, eine nicht regelkonforme Triangulierung, eine regelkonforme lokal-verfeinerte Triangulierung und der Träger (grau) einer Basisfunktion in einer Triangulierung und dessen Durchmesser (dicke Linie).

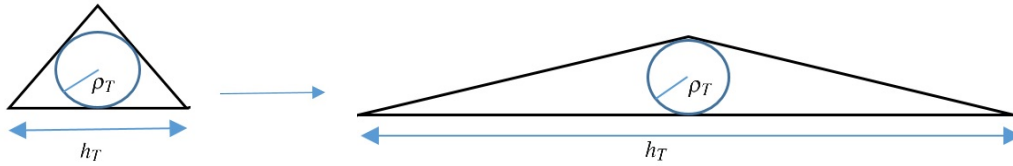


ABBILDUNG 3: Links ist ein finites Element mit Durchmesser h_T und Inkreisradius ρ_T , welches nach einer Verfeinerung einen größeren Durchmesser h_T , aber einen gleichen Inkreisradius ρ_T besitzt. Falls dies sukzessive durchgeführt wird, ist diese Art der Triangulierung nicht regulär.

Für spätere Abschätzungen benötigen wir folgendes

Lemma 1 (Normäquivalenz). *Für Konstanten c, C gilt für alle $v_h \in \mathcal{S}^1(\mathcal{T}_h)$ und für alle $1 \leq r < \infty$ die Normäquivalenz*

$$c \|v_h\|_{L^r(\Omega)}^r \leq \sum_{z \in \mathcal{N}_h} h_z^d |v_h(z)|^r \leq C \|v_h\|_{L^r(\Omega)}^r$$

Beweis: $\|\cdot\|_{L^r(\Omega)}$ ist eine Norm auf $\mathcal{S}^1(\mathcal{T}_h)$. Weiterhin ist $(\sum_{z \in \mathcal{N}_h} h_z^d |v_h(z)|^r)^{1/r}$ eine Norm:

Die positive Definitheit ist klar. Sei nun $\alpha \in \mathbb{R}$. Dann folgt mit

$$\left(\sum_{z \in \mathcal{N}_h} h_z^d |\alpha \cdot v_h(z)|^r \right)^{1/r} = \left(\sum_{z \in \mathcal{N}_h} h_z^d \alpha^r |v_h(z)|^r \right)^{1/r} = \alpha \left(\sum_{z \in \mathcal{N}_h} h_z^d |v_h(z)|^r \right)^{1/r}$$

sofort die absolute Homogenität. Die Dreiecksungleichung folgt direkt aus der Minkowski-Ungleichung für Folgen, womit der Ausdruck eine Norm ist. Da wir uns auf dem endlich-dimensionalen Raum $\mathcal{S}^1(\mathcal{T}_h)$ befinden, sind die beiden Normen äquivalent zueinander und die Behauptung folgt. \square

2.2 Differenzenquotienten

Da wir später in der Diskretisierung der Wärmeleitungsgleichung Zeitableitungen approximieren, benötigen wir Differenzenquotienten. Wir benutzen den Rückwärtsdifferenzenquotient: Für eine Folge von Funktionen oder Zahlen $(a^k)_{k=0,\dots,K}$, wobei die Werte zum jeweiligen Zeitschritt $t_k = k \cdot \tau$ mit der Zeitschrittweite τ assoziiert sind, bezeichnen wir den Rückwärtsdifferenzenquotient mit

$$d_t a^k = \frac{1}{\tau} (a^k - a^{k-1}).$$

2.3 Diskrete Einschränkung

In diesem Unterkapitel zeigen wir, dass, falls die Einheitskugelbeschränkung an den Knoten gilt, sie auch in Häufungspunkten im Grenzwert $(h, \tau) \rightarrow 0$ erfüllt ist.

Lemma 2 (Approximation der Einheitssphärenbeschränkung).

Falls $(u_h)_{h>0}$ eine beschränkte Folge in $H^1(\Omega; \mathbb{R}^m)$ mit $u_h \in \mathcal{S}^1(\mathcal{T}_h)^m$ für alle $h > 0$ ist, $u_h \rightarrow u$ in $L^2(\Omega; \mathbb{R}^m)$ für $h \rightarrow 0$ für ein $u \in H^1(\Omega; \mathbb{R}^m)$ gilt und falls

$$\|\mathcal{I}_h[|u_h|^2] - 1\|_{L^1(\Omega)} \rightarrow 0$$

für $h \rightarrow 0$, dann gilt, dass $|u|^2 = 1$ fast überall in Ω .

Beweis: Wir wenden die Dreiecksungleichung zwei Mal an und erhalten

$$\left\| |u|^2 - 1 \right\|_{L^1(\Omega)} \leq \left\| |u|^2 - |u_h|^2 \right\|_{L^1(\Omega)} + \left\| |u_h|^2 - \mathcal{I}_h[|u_h|^2] \right\|_{L^1(\Omega)} + \left\| \mathcal{I}_h[|u_h|^2] - 1 \right\|_{L^1(\Omega)}.$$

Nach Voraussetzung konvergiert der letzte Term für $h \rightarrow 0$ gegen 0. Der erste Term konvergiert auch gegen 0, da mit der dritten binomischen Formel und der Hölder-Ungleichung mit $p = q = 2$ folgt:

$$\left\| |u|^2 - |u_h|^2 \right\|_{L^1(\Omega)} \leq \| |u| - |u_h| \|_{L^2(\Omega)} \cdot \| |u| + |u_h| \|_{L^2(\Omega)}.$$

Nach Voraussetzung konvergiert der erste Term auf der rechten Seite für $h \rightarrow 0$ gegen 0 und somit der Term auf der linken Seite. Für den zweiten Term von oben benutzen wir wieder die Hölder-Ungleichung mit $p = q = 2$ und erhalten

$$\begin{aligned}
\left\| |u_h|^2 - \mathcal{I}_h \left[|u_h|^2 \right] \right\|_{L^1(\Omega)} &\leq \|1\|_{L^2(\Omega)} \cdot \left\| |u_h|^2 - \mathcal{I}_h \left[|u_h|^2 \right] \right\|_{L^2(\Omega)} \\
&= \left(\int_{\Omega} 1^2 dx \right)^{1/2} \left\| |u_h|^2 - \mathcal{I}_h \left[|u_h|^2 \right] \right\|_{L^2(\Omega)} \\
&\leq \left(\sup_{x \in \Omega} 1 \cdot C_1 \cdot h_T^d \right)^{1/2} \left\| |u_h|^2 - \mathcal{I}_h \left[|u_h|^2 \right] \right\|_{L^2(\Omega)} \\
&= C_2 \cdot h_T^{d/2} \left\| |u_h|^2 - \mathcal{I}_h \left[|u_h|^2 \right] \right\|_{L^2(\Omega)}.
\end{aligned}$$

Für die nächste Abschätzung benötigen wir eine Interpolation-Fehlerabschätzung. Diese ist aus [Bartels et al., 2015], wo auch der Beweis zu finden ist:

Hilfssatz 1 (Interpolationsabschätzung). *Sei T ein finites Element mit Durchmesser h_T , welches die Regularitätsanforderungen von oben erfüllt. Dann gilt für die lineare Interpolation \mathcal{I}_h folgende Abschätzung für eine Konstante C :*

$$\|v - \mathcal{I}_h v\|_{L^2(T)} \leq C \cdot h_T^2 \cdot \|D^2 v\|_{L^2(T)}$$

für alle $v \in H^2(\Omega; \mathbb{R}^m)$.

Mit diesem folgt

$$C_2 \cdot h_T^{d/2} \left\| |u_h|^2 - \mathcal{I}_h \left[|u_h|^2 \right] \right\|_{L^2(T)} \leq C_3 \cdot h_T^{d/2} h_T^2 \left\| D^2 |u_h|^2 \right\|_{L^2(T)} \leq C_4 \cdot h_T^2 \|\nabla u_h\|_{L^2(T)}^2.$$

Nun summieren wir über alle $T \in \mathcal{T}_h$ und nutzen die Eigenschaft $h_T \leq ch$ für ein geeignetes c und erhalten folgende Abschätzung für den zweiten Term über Ω

$$\left\| |u_h|^2 - \mathcal{I}_h \left[|u_h|^2 \right] \right\|_{L^1(\Omega)} \leq C_5 \cdot h^2 \|\nabla u_h\|_{L^2(\Omega)}^2.$$

Für $h \rightarrow 0$ verschwindet auch dieser Term, womit insgesamt die Behauptung folgt. \square

Bemerkung 1. *Diese Aussage kann auch für allgemeine Einschränkungen $g(u) = 0$ gezeigt werden. Dann benötigt man allerdings quasiuniforme Triangulierungen oder die Schrittwerteneinschränkung $h \leq ch_{\min}^{d/2-1} \log(h_{\min}^{-1})^{-1/2}$ ([Bartels, 2016]).*

3 Harmonischer Wärmefluss

Das Problem des harmonischen Wärmeflusses auf der Einheitssphäre ist gegeben durch

$$\begin{aligned} \partial_t u - \Delta u &= |\nabla u|^2 u, & u(t, \cdot)|_{\Gamma_D} &= u_D, & u(0) &= u_0, \\ |u|^2 &= 1, & \partial_\nu u(t, \cdot)|_{\Gamma_N} &= 0. \end{aligned}$$

3.1 Eigenschaften der schwachen Formulierung

Die schwache Formulierung erhalten wir mit den Methoden der Variationsrechnung: Multipliziere dazu die Differentialgleichung mit $w \in H_D^1(\Omega; \mathbb{R}^m)$ und integriere über Ω und erhalte

$$\int_{\Omega} (\partial_t u) \cdot w \, dx - \int_{\Omega} (\Delta u) \cdot w \, dx = \int_{\Omega} |\nabla u|^2 \cdot u \cdot w \, dx.$$

Mit der Greenschen Formel erhalten wir für den zweiten Term der linken Seite mit dem äußeren Normalenvektor n

$$\int_{\Omega} (\Delta u) \cdot w \, dx = \int_{\Gamma} (\nabla u) \cdot w \cdot n \, dx - \int_{\Omega} \nabla u \cdot \nabla w \, dx.$$

Da $w \in H_D^1(\Omega)$, verschwindet w auf $\Gamma_D \subset \Gamma$ und das Integral wird 0. Auf Γ_N haben wir die Bedingung, dass $\partial_\nu u|_{\Gamma_N} = \nabla u \cdot \nu|_{\Gamma_N} = 0$ und das Integral fällt auch dafür weg. Daraus folgt insgesamt die schwache Formulierung

$$(\partial_t u, w) + (\nabla u, \nabla w) = (|\nabla u|^2 u, w),$$

für fast alle $t \in [0, T]$ und alle $w \in H_D^1(\Omega; \mathbb{R}^m)$ mit $|u|^2 = 1$, $u(0) = u_0$ und den Randbedingungen $u(t, \cdot)|_{\Gamma_D} = u_D$, $\partial_\nu u(t, \cdot)|_{\Gamma_N} = 0$. Für diese schwache Formulierung existiert eine Lösung nach folgendem Satz, für dessen Beweis wir nur auf [Bartels et al., 2015] verweisen.

Satz 3 (Existenz). *Für ein gegebenes $u_0 \in H^1(\Omega; \mathbb{R}^m)$ mit $|u_0(x)| = 1$ für fast alle $x \in \Omega$ existiert ein $u \in H^1([0, T]; L^2(\Omega; \mathbb{R}^m)) \cap L^\infty([0, T]; H^1(\Omega; \mathbb{R}^m))$ mit $|u(t, x)| = 1$ für fast alle $(t, x) \in [0, T] \times \Omega$, $u(0, \cdot) = u_0$ und*

$$(\partial_t u, w) + (\nabla u, \nabla w) = (|\nabla u|^2 u, w)$$

für fast alle $t \in [0, T]$ und alle $w \in H_D^1(\Omega; \mathbb{R}^m) \cap L^\infty(\Omega; \mathbb{R}^m)$.

Bemerkung 2. (i) *Die Eindeutigkeit einer solchen Lösung für $d = 2$ ist für Lösungen, die das Dirichlet-Funktional in der Zeit nicht ansteigen lassen, in [Freire, 1995] und für Lösungen, die nur bestimmtes Wachstum beim Funktional zulassen, in [Rupflin, 2010] bewiesen.*

(ii) Lösungen der Wärmeleitungsgleichung können in endlich vielen Schritten Singularitäten entwickeln, wie es in diesem Beispiel aus [Chang et al., 1992], welches wir später implementieren, der Fall ist:

Sei $\Omega = (-1/2, 1/2)^2$, $\Gamma_D = \partial\Omega$, $\Gamma_N = \emptyset$, $u_D = u_0|_{\Gamma_D}$ mit u_0 definiert durch

$$u_0(x) = \frac{1}{|x|} (x_1 \sin(\phi(2|x|)), x_2 \sin(\phi(2|x|)), |x| \cos \phi(2|x|))^T$$

für $x = (x_1, x_2) \in \Omega$ und $\phi(s) = (3\pi/2) \min\{s^2, 1\}$.

Singulär bedeutet hier, dass ein $T_c > 0$ existiert, sodass $\lim_{t \rightarrow T_c} \|\nabla u(t)\|_{L^\infty(\Omega)} = \infty$.

Wie in der Eingangsideoe beschrieben, fordern wir eine Orthogonalitätsrelation, wodurch die rechte Seite der Differentialgleichung wegfällt. Dass es sich dabei um eine äquivalente Problemstellung handelt, zeigen wir für $m = 3$ nun in folgendem

Satz 4 (Äquivalente Formulierungen).

Für eine Funktion $u \in H^1([0, T]; L^2(\Omega; \mathbb{R}^3)) \cap L^\infty([0, T]; H^1(\Omega; \mathbb{R}^3))$ mit $|u(t, x)| = 1$ für fast alle $(t, x) \in [0, T] \times \Omega$ sind folgende Aussagen äquivalent:

(i) Für fast alle $t \in [0, T]$ und alle $w \in H_D^1(\Omega; \mathbb{R}^3) \cap L^\infty(\Omega; \mathbb{R}^3)$ gilt

$$(\partial_t u, w) + (\nabla u, \nabla w) = (|\nabla u|^2 u, w).$$

(ii) Für fast alle $t \in [0, T]$ und alle $w \in H_D^1(\Omega; \mathbb{R}^3)$ mit $w(x) \cdot u(t, x) = 0$ für fast alle $x \in \Omega$ gilt

$$(\partial_t u, w) + (\nabla u, \nabla w) = 0.$$

(iii) Für fast alle $t \in [0, T]$ und alle $\phi \in H_D^1(\Omega; \mathbb{R}^3) \cap L^\infty(\Omega; \mathbb{R}^3)$ gilt

$$(\partial_t u, \phi) - (\nabla u, \nabla[u \times (u \times \phi)]) = 0.$$

Beweis: (i) \rightarrow (ii)

Sei (i) erfüllt ist und gelte weiterhin für alle $w \in H_D^1(\Omega; \mathbb{R}^3) \cap L^\infty(\Omega; \mathbb{R}^3)$ $w(x) \cdot u(t, x) = 0$ für fast alle x . Dann folgt sofort mit (i)

$$(\partial_t u, w) + (\nabla u, \nabla w) = (|\nabla u|^2 u, w) = 0.$$

Die Aussage für alle $w \in H_D^1(\Omega; \mathbb{R}^3)$ folgt mit folgenden abgeschnittenen Funktionen w_n :

$$w_n(x) = \begin{cases} w(x) & \text{für } \|w(x)\| \leq n \\ n & \text{für } \|w(x)\| > n \end{cases}$$

Für diese gilt, dass $w_n \in H_D^1(\Omega; \mathbb{R}^3) \cap L^\infty(\Omega; \mathbb{R}^3)$ für alle n und $w(x) = \lim_{n \rightarrow \infty} w_n(x)$. Daraus folgt für alle $w \in H_D^1(\Omega; \mathbb{R}^3)$ mit der Linearität des Differentialoperators und des Skalarprodukts:

$$\begin{aligned} (\partial_t u, w) + (\nabla u, \nabla w) &= (\partial_t u, \lim_{n \rightarrow \infty} w_n(x)) + (\nabla u, \nabla \lim_{n \rightarrow \infty} w_n(x)) \\ &= \lim_{n \rightarrow \infty} ((\partial_t u, w_n) + (\nabla u, \nabla w_n)) = 0. \end{aligned}$$

(ii) \rightarrow (iii)

Aussage (ii) gelte für alle $w \in H_D^1(\Omega; \mathbb{R}^3)$. Dann gilt diese auch für das Kreuzprodukt $w = u \times (u \times \phi) \in H_D^1(\Omega; \mathbb{R}^3)$ für fast alle t , denn mit der Graßmann-Identität $a \times (b \times c) = b(a \cdot c) - c(a \cdot b)$ für $a, b, c \in \mathbb{R}^3$ folgt

$$u \times (u \times \phi) = u(u \cdot \phi) - \phi(u \cdot u) = u(u \cdot \phi) - \phi|u|^2 = u(u \cdot \phi) - \phi$$

und somit auch

$$w \cdot u = (u(u \cdot \phi) - \phi) \cdot u = u \cdot \phi - u \cdot \phi = 0.$$

Damit folgt insgesamt aus (ii)

$$\begin{aligned} 0 &= (\partial_t u, w) + (\nabla u, \nabla w) = (\partial_t u, u(u \cdot \phi) - \phi) + (\nabla u, \nabla[u \times (u \times \phi)]) \\ &= (\partial_t u, u(u \cdot \phi)) - (\partial_t u, \phi) + (\nabla u, \nabla[u \times (u \times \phi)]). \end{aligned}$$

Umgestellt folgt dann mit $\partial_t u \cdot u = 0$, dass

$$(\partial_t u, \phi) - (\nabla u, \nabla[u \times (u \times \phi)]) = (\partial_t u, u(u \cdot \phi)) = 0.$$

(iii) \rightarrow (i)

Wir wählen wie oben $w = u \times (u \times \phi) = u(u \cdot \phi) - \phi$ und erhalten, dass

$$\begin{aligned} (\partial_t u, \phi) - (\nabla u, \nabla[u \times (u \times \phi)]) &= 0 = (|\nabla u|^2 u, \phi) - (|\nabla u|^2 u, \phi) \\ &= (|\nabla u|^2 u, \phi) - (|\nabla u|^2 u, u(u \cdot \phi)) \\ &= (|\nabla u|^2 u, -(u(u \cdot \phi) - \phi)) \end{aligned}$$

für alle $w \in H_D^1(\Omega; \mathbb{R}^3) \cap L^\infty(\Omega; \mathbb{R}^3)$ äquivalent ist zu

$$(\partial_t u, \phi) - (\nabla u, \nabla w) = -(|\nabla u|^2 u, w).$$

Wieder folgt mit $\partial_t u \cdot u = 0$

$$\begin{aligned} -(\partial_t u, \phi) + (\nabla u, \nabla w) &= (\partial_t u, u(u \cdot \phi)) - (\partial_t u, \phi) + (\nabla u, \nabla w) \\ &= (\partial_t u, w) + (\nabla u, \nabla w) = (|\nabla u|^2 u, w) \end{aligned}$$

und Aussage (i) gilt, womit alle Aussagen äquivalent sind. □

3.2 Algorithmus zum harmonischen Wärmefluss

Nun betrachten wir einen Algorithmus für die zweite Formulierung, da bei dieser die für uns aufwändiger zu berechnende rechte Seite der ersten Formulierung wegfällt:

$$(\partial_t u, w) + (\nabla u, \nabla w) = 0, |u|^2 = 1, u(0) = u_0$$

für fast alle $t \in [0, T]$ und alle $w \in H_D^1(\Omega; \mathbb{R}^m)$ mit $w(x) \cdot u(x, t) = 0$ für fast alle $x \in \Omega$, wobei $|u|^2 = 1$ mit den Randbedingungen

$$u(t, \cdot)|_{\Gamma_D} = u_D, \quad \partial_\nu u(t, \cdot)|_{\Gamma_N} = 0.$$

Dies ist äquivalent zu unserem Ausgangsproblem, wo die rechte Seite nicht null ist.

In folgendem Algorithmus wird eine Approximationslösung iterativ bestimmt.

Algorithmus 1 (Harmonischer Wärmefluss auf Einheitssphäre).

Sei $u^0 \in H^1(\Omega; \mathbb{R}^m)$ eine gegebene Anfangsfunktion mit $u_0|_{\Gamma_D} = u_D$ und $|u_0| = 1$ in Ω .

Definiere nun $u_h^0 \in \mathcal{S}^1(\mathcal{T}_h)^m$ durch $u_h^0(z) = u_0(z)$ für alle $z \in \mathcal{N}_h$ und setze $k = 1$.

(1) Berechne $v_h^k \in \mathcal{S}^1(\mathcal{T}_h)^m$, sodass gilt

$$v_h^k(z) \cdot u_h^{k-1}(z) = 0$$

für alle $z \in \mathcal{N}_h$ und

$$(v_h^k, w_h) + (\nabla[u_h^{k-1} + \tau v_h^k], \nabla w_h) = 0$$

für alle $w_h \in \mathcal{S}^1(\mathcal{T}_h)^m$ mit $w_h(z) \cdot u_h^{k-1}(z) = 0$ für alle $z \in \mathcal{N}_h$.

(2) Setze $u_h^k = u_h^{k-1} + \tau v_h^k$ und beende, falls $k \geq K = \lceil T/\tau \rceil$.

Bemerkung 3. (i) Man kann in diesen Algorithmus einen Projektionsschritt einbauen. Dann benötigt man allerdings für die Stabilität der Iteration, dass \mathcal{T}_h schwach spitz (jede Summe zweier Winkel, welche gegenüber der gemeinsamen Kante zweier finiter Elemente liegen, ist beschränkt durch π und jede Kante auf dem Rand durch $\pi/2$) oder man muss die Schrittweite einschränkung $\tau = \mathcal{O}(h^2)$ fordern (Vergleiche [Bartels, 2016]).

(ii) *Bedingungslose Konvergenz dieses Algorithmus wird für $d = 3$ in [Bartels et al., 2015] bewiesen.*

Wir werden im Folgenden die Stabilität dieses Algorithmus beweisen.

Proposition 1 (Stabilitätsabschätzung). *Seien $(u_h^k)_{k=0,\dots,K} \subset \mathcal{S}^1(\mathcal{T}_h)^m$ die Iterierten des obigen Algorithmus. Es gilt $v_h^k = d_t u_h^k$ für $k = 1, \dots, K$ und für $L = 1, \dots, K$ folgt*

$$I(u_h^L) + \tau \sum_{k=1}^L \|v_h^k\|^2 + \frac{\tau^2}{2} \sum_{k=1}^L \|\nabla v_h^k\|^2 = I(u_h^0),$$

$$\|\mathcal{I}_h [|u_h^L|^2] - 1\|_{L^1(\Omega)} \leq c\tau I(u_h^0).$$

Beweis: Für ein gegebenes $u_h^{k-1} \in \mathcal{S}^1(\mathcal{T}_h)^m$ ist die Menge

$$\mathcal{F}_h[u_h^{k-1}] = \{w_h \in \mathcal{S}^1(\mathcal{T}_h)^m : w_h(z) \cdot g'(u_h^{k-1}(z)) = 0 \quad \forall z \in \mathcal{N}_h\}$$

ein abgeschlossener Unterraum von $\mathcal{S}^1(\mathcal{T}_h)^m$. Da dies ein abgeschlossener Unterraum des Hilbertraums $\mathcal{S}^1(\mathcal{T}_h)^m$ ist, ist auch dieser ein Hilbertraum und es folgt sofort mit dem Lax-Milgram-Lemma die Existenz einer eindeutigen Lösung v_h^k in jeder Iteration.

Setze nun die Testfunktion $w_h = v_h^k = d_t u_h^k$ in die Gleichung des Algorithmus ein:

$$(v_h^k, v_h^k) + (\nabla[u_h^{k-1} + \tau v_h^k], \nabla v_h^k) = \|v_h^k\|^2 + (\nabla[u_h^{k-1} + \tau v_h^k], \nabla v_h^k) = 0.$$

Für den zweiten Term verwenden wir die Binomialgleichung $(a+b)a = \frac{1}{2}(a^2 + (a+b)^2 - b^2)$, wobei $a = \tau \nabla v_h^k$ und $b = \nabla u_h^{k-1}$, und beachten $u_h^k = u_h^{k-1} + \tau v_h^k$:

$$\begin{aligned} (\nabla[u_h^{k-1} + \tau v_h^k], \nabla v_h^k) &= \int_{\Omega} \nabla[u_h^{k-1} + \tau v_h^k] \cdot \nabla v_h^k \, dx = \frac{1}{\tau} \int_{\Omega} (\nabla u_h^{k-1} + \tau \nabla v_h^k) \tau \nabla v_h^k \, dx \\ &= \frac{1}{2\tau} \int_{\Omega} \tau^2 (\nabla v_h^k)^2 + (\tau \nabla v_h^k + \nabla u_h^{k-1})^2 - (\nabla u_h^{k-1})^2 \, dx \\ &= \frac{\tau}{2} \|\nabla v_h^k\|^2 + \frac{1}{2\tau} \|\nabla u_h^k\|^2 - \frac{1}{2\tau} \|\nabla u_h^{k-1}\|^2. \end{aligned}$$

Ingesamt folgt damit

$$\|v_h^k\|^2 + \frac{1}{2\tau} \left(\|\nabla u_h^k\|^2 - \|\nabla u_h^{k-1}\|^2 \right) + \frac{\tau}{2} \|\nabla v_h^k\|^2 = 0.$$

Wir multiplizieren mit τ und summieren über $k = 1, \dots, L$ und erhalten

$$\tau \sum_{k=1}^L \|v_h^k\|^2 + \frac{1}{2} \sum_{k=1}^L \left(\|\nabla u_h^k\|^2 - \|\nabla u_h^{k-1}\|^2 \right) + \frac{\tau^2}{2} \sum_{k=1}^L \|\nabla v_h^k\|^2 = 0.$$

Der mittlere Term ist eine Teleskopsumme und nach Umstellen erhalten wir

$$\frac{1}{2} \|\nabla u_h^L\|^2 + \tau \sum_{k=1}^L \|v_h^k\|^2 + \frac{\tau^2}{2} \sum_{k=1}^L \|\nabla v_h^k\|^2 = \frac{1}{2} \|\nabla u_h^0\|^2 = I(u_h^0).$$

Für die zweite Ungleichung gilt wegen der punkweisen Orthogonalität von u_h^{k-1} zu v_h^k

$$\left| u_h^k(z) \right|^2 = \left| u_h^{k-1}(z) + \tau v_h^k(z) \right|^2 = \left| u_h^{k-1}(z) \right|^2 + \tau^2 \left| v_h^k(z) \right|^2$$

und induktiv folgt mit $|u_h^0(z)| = 1$

$$\left| u_h^L(z) \right|^2 - 1 = \tau^2 \sum_{k=1}^L \left| v_h^k(z) \right|^2.$$

Aufgrund der Normäquivalenz aus Lemma 1 folgt damit

$$\begin{aligned} c_1 \left\| \mathcal{I}_h \left[|u_h^L|^2 \right] - 1 \right\|_{L^1(\Omega)} &\leq \sum_{z \in \mathcal{N}_h} h_z^d \left| |u_h^L(z)|^2 - 1 \right| \leq \tau^2 \sum_{k=1}^L \sum_{z \in \mathcal{N}_h} h_z^d \left| v_h^k(z) \right|^2 \\ &\leq C_1 \tau^2 \sum_{k=1}^L \left\| v_h^k \right\|_{L^2(\Omega)}^2. \end{aligned}$$

Mit der Poincaré-Ungleichung $\|v_h^k\|_{L^2(\Omega)} \leq c \|\nabla v_h^k\|_{L^2(\Omega)}$ für $v \in H_0^1(\Omega)$ und der Stabilitätsgleichung folgt

$$\left\| \mathcal{I}_h \left[|u_h^L|^2 \right] - 1 \right\|_{L^1(\Omega)} \leq c \tau^2 \sum_{k=1}^L \left\| \nabla v_h^k \right\|_{L^2(\Omega)}^2 \leq c \tau \|\nabla u_h^0\|^2 = c \tau I(u_h^0),$$

womit die Proposition bewiesen ist. □

4 Numerische Implementation

In diesem Kapitel betrachten wir an einem Beispiel die Implementation des projektionsfreien harmonischen Wärmeflusses für $d = 2$ in Matlab.

Beispiel 1. Sei $d = 2$, das Zeitintervall $T = [0, 2]$ über dem Gebiet $\Omega = (-1/2, 1/2)^2$ mit $\Gamma_D = \partial\Omega$ und $\Gamma_N = \emptyset$, wobei wir Vektorfelder u betrachten mit $u : \Omega \rightarrow \mathbb{R}^3$ und $|u|^2 = 1$ mit der Startfunktion

$$u_0(x) = \frac{1}{|x|} (x_1 \sin \phi(2|x|), x_2 \sin \phi(2|x|), |x| \cos \phi(2|x|))^T$$

für $x = (x_1, x_2) \in \Omega$ und $\phi(s) = (3\pi/2) \min\{s^2, -1\}$.

Für die Implementierung verwenden wir eine reguläre und gleichmäßige Triangulierung mit Schrittweite 2^{-n} mit $n \in \mathbb{N}_0$. Wir lassen dabei quadratische Gebiete der Länge 2^n mit $n \in \mathbb{N}$ und beliebigen Mittelpunkt $(x, y) \in \mathbb{R}^2$ zu. Dies realisiert *Gebietsdefinition*. Die Funktion *Verfeinerung* erstellt dabei eine Triangulierung mit zulässiger Schrittweite. Dabei zählen wir die Knoten von unten beginnend jede Zeile von links aufwärts bis wir am rechten oberen Knoten angelangt sind. Die Knotenkoordinaten speichern wir im Vektor *Knoten*.

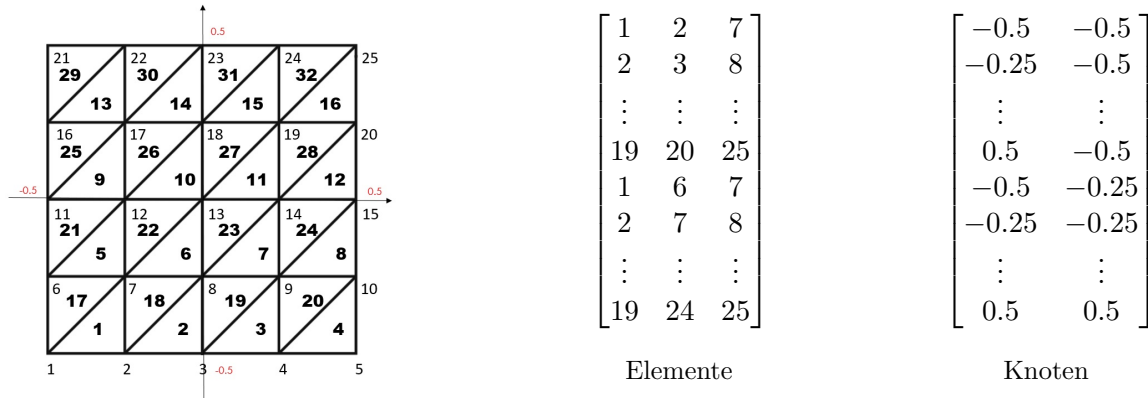


ABBILDUNG 4: In dünner Schrift ist die Nummerierung der Knoten gekennzeichnet, in dicker die der Elemente. Die Vektoren daneben korrespondieren dabei zu den Elementen bzw. zu den Koordinaten der Knoten. Die Nummerierung der Knoten wird durch die Zeilenzahl erreicht. Jede Zeile des Elementvektor korrespondiert zu einem finiten Element mit den jeweiligen Eckpunkten, welche mit der zugehörigen Zeile des Knotenvektors korrespondieren.

Die Elemente teilen wir in zwei Untermengen ein. Die erste Untermenge enthält die Knoten, deren 90-Grad-Eck nach unten rechts zeigt, die zweite Untermenge enthält die Knoten, deren 90-Grad-Eck nach oben links zeigt. Zuerst zählen wir - wie bei den Knoten - die

Elemente der ersten Untermenge von unten links beginnend zeilenweise nach oben. Das gleiche Prinzip wenden wir danach auf die zweite Untergruppe an. Siehe Abbildung 4.

4.1 Aufstellung des Gleichungssystems

Für die Aufstellung des Gleichungssystems beachten wir, dass wir nur lineare Basisfunktionen $(\varphi_z \in \mathcal{S}^1(\mathcal{T}_h) : z \in \mathcal{N}_h)$ mit $\mathcal{S}^1(\mathcal{T}_h) = \{w_h \in C(\bar{\Omega}) : w_h|_T \text{ affin für alle } T \in \mathcal{T}_h\}$ benutzen. Damit können wir u_h^{k-1} , v_h^k und w_h aus $\mathcal{S}^1(\mathcal{T}_h)^3$ in der i-ten Vektorkomponente als Linearkombination darstellen:

$$(u_h^{k-1})_i = \sum_{z \in \mathcal{N}_h} \mu_z \varphi_z, (v_h^k)_i = \sum_{z \in \mathcal{N}_h} \nu_z \varphi_z, (w_h)_i = \sum_{z \in \mathcal{N}_h} \omega_z \varphi_z,$$

wobei $\mu_z = (u_h^{k-1}(z))_i$, $\nu_z = (v_h^k(z))_i$ und $\omega_z = (w_h(z))_i$ für alle $z \in \mathcal{N}_h$ sind. Dann folgt mit der Linearität der Bilinearform und der zu implementierenden Gleichung im k-ten Zeitschritt und der i-ten Komponente für alle w_h

$$\begin{aligned} 0 &= (v_h^k, w_h) + (\nabla[u_h^{k-1} + \tau v_h^k], \nabla w_h) = \\ &= \left(\sum_{z \in \mathcal{N}_h} \nu_z \varphi_z, \sum_{n \in \mathcal{N}_h} \omega_n \varphi_n \right) + \left(\sum_{m \in \mathcal{N}_h} \mu_m \nabla \varphi_m, \sum_{n \in \mathcal{N}_h} \omega_n \nabla \varphi_n \right) + \left(\tau \sum_{z \in \mathcal{N}_h} \nu_z \nabla \varphi_z, \sum_{n \in \mathcal{N}_h} \omega_n \nabla \varphi_n \right) = \\ &= \sum_{z \in \mathcal{N}_h} \sum_{n \in \mathcal{N}_h} \nu_z \omega_n (\varphi_z, \varphi_n) + \sum_{m \in \mathcal{N}_h} \sum_{n \in \mathcal{N}_h} \mu_m \omega_n (\nabla \varphi_m, \nabla \varphi_n) + \tau \sum_{z \in \mathcal{N}_h} \sum_{n \in \mathcal{N}_h} \nu_z \omega_n (\nabla \varphi_z, \nabla \varphi_n). \end{aligned}$$

In Matrixschreibweise ergibt das

$$\omega^\top M \nu + \omega^\top S \mu + \tau \omega^\top S \nu = 0 \Leftrightarrow \omega^\top (M + \tau S) \nu = -\omega^\top S \mu,$$

wobei M die Massematrix, S die Steifigkeitsmatrix und $\omega = (\omega_{n_j})_{j=1}^L$, $\mu = (\mu_{m_j})_{j=1}^L$ und $\nu = (\nu_{z_j})_{j=1}^L$ die Koeffizientenvektoren der Länge $L = \#\mathcal{N}_h$ sind.

Somit lösen wir die Gleichung für jeden Vektoreintrag und fügen die Lösung anschließend zusammen.

Das heißt, im Folgenden entwickeln wir die Steifigkeitsmatrix S und Massematrix M für die i-te Komponente der Lösung u . Die ij-Komponente von S ist

$$(S)_{ij} = (\nabla \varphi_i, \nabla \varphi_j) = \int_{\Omega} \nabla \varphi_i \nabla \varphi_j dx = \sum_e \int_e \nabla \varphi_i \nabla \varphi_j dx,$$

wobei über alle finiten Elemente e summiert wird. Letzteres Integral über einem Dreieck e bzw. f lässt sich leicht mit den in Abbildung 5 zu sehenden Basisfunktionen auf diesem

berechnen:

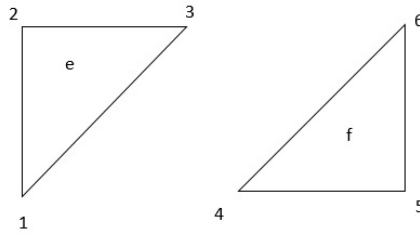


ABBILDUNG 5: $\varphi_1 = 1 - \frac{y}{h}$, $\varphi_2 = \frac{y}{h} - \frac{x}{h}$, $\varphi_3 = \frac{x}{h}$, $\varphi_4 = 1 - \frac{x}{h}$, $\varphi_5 = \frac{x}{h} - \frac{y}{h}$, $\varphi_6 = \frac{y}{h}$

Das Differential der jeweiligen Basisfunktionen ergibt sich aufgrund der Linearität sofort:

$$\begin{aligned} \nabla\varphi_1 &= \begin{pmatrix} 0 \\ -1/h \end{pmatrix}, \quad \nabla\varphi_2 = \begin{pmatrix} -1/h \\ 1/h \end{pmatrix}, \quad \nabla\varphi_3 = \begin{pmatrix} 1/h \\ 0 \end{pmatrix}, \\ \nabla\varphi_4 &= \begin{pmatrix} -1/h \\ 0 \end{pmatrix}, \quad \nabla\varphi_5 = \begin{pmatrix} 1/h \\ -1/h \end{pmatrix}, \quad \nabla\varphi_6 = \begin{pmatrix} 0 \\ 1/h \end{pmatrix}. \end{aligned}$$

Damit folgt

$$\int_e \nabla\varphi_1 \nabla\varphi_1 d(x,y) = \int_e \begin{pmatrix} 0 \\ -1/h \end{pmatrix}^\top \cdot \begin{pmatrix} 0 \\ -1/h \end{pmatrix} dx = 1/h^2 \cdot \int_e 1 \, dx = 1/h^2 \cdot h^2/2 = 1/2.$$

Aus Symmetriegründen bzw. analoger Rechnung folgt

$$\begin{aligned} \int \nabla\varphi_j \nabla\varphi_j &= 1/2 \quad \text{für } j = 3, 4, 6, & \int \nabla\varphi_j \nabla\varphi_j &= 1 \quad \text{für } j = 2, 5, \\ \int \nabla\varphi_i \nabla\varphi_j &= 0 \quad \text{für } i = 1, 4, \quad j = 3, 6, & \int \nabla\varphi_i \nabla\varphi_j &= -1/2 \quad \text{für } i, j \text{ sonst.} \end{aligned}$$

Nach Summation über alle Elemente ergibt sich somit die Steifigkeitsmatrix. Für die Massematrix berechnen wir einmal beispielhaft einen Matrixeintrag, wobei wir uns wieder auf ein finites Element beschränken und mit Summation den globalen Eintrag erhalten. Für die restlichen Einträge geht man analog vor:

$$\int_e \varphi_1 \varphi_1 d(x,y) = \int_0^h \int_0^y \left(1 - \frac{y}{h}\right)^2 dx dy = \int_0^h \left(1 - \frac{y}{h}\right)^2 \cdot y \, dy = \int_0^h y - \frac{2y^2}{h} + \frac{y^3}{h^2} dy = \frac{h^2}{12}.$$

Somit folgt wieder analog und aus Symmetriegründen

$$\int \varphi_j \varphi_j = \frac{h^2}{12} \quad \text{für } j = 2, 3, 4, 5, 6, \quad \int \varphi_i \varphi_j = \frac{h^2}{24} \quad \text{für } i \neq j.$$

Mit Summation über alle Dreiecke erhalten wir die Massematrix.

Da wir die Lösung in jeder Vektorkomponente haben wollen, führen wir die Matrizen SSS und MMM ein, welche die gestreckte Steifigkeits- bzw. Massematrix sind:

$$S = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \rightarrow SSS = \begin{bmatrix} a_{11} & 0 & 0 & a_{12} & 0 & 0 \\ 0 & a_{11} & 0 & 0 & a_{12} & 0 \\ 0 & 0 & a_{11} & 0 & 0 & a_{12} \\ a_{21} & 0 & 0 & a_{22} & 0 & 0 \\ 0 & a_{21} & 0 & 0 & a_{22} & 0 \\ 0 & 0 & a_{21} & 0 & 0 & a_{22} \end{bmatrix}$$

Mit diesen Matrizen behandeln wir obiges Gleichungssystem analog.

Weil wir außerdem die Orthogonalitätsbedingung fordern, erweitern wir das Gleichungssystem um die Gleichung $B \cdot V = 0$, wobei B die folgende Form hat

$$B = \begin{bmatrix} \mu_{z_1}^\top & 0 & & \\ 0 & \mu_{z_2}^\top & 0 & \\ & 0 & \ddots & 0 \\ & & 0 & \mu_{z_L}^\top \end{bmatrix},$$

sodass $B \cdot V = [\mu_{z_1} \cdot v_{z_1}, \dots, \mu_{z_L} \cdot v_{z_L}]^\top$. Für die Lösung dieses Gleichungssystems gilt dann

$$B \cdot V = 0, \quad \omega^\top (MMM + \tau SSS) \nu = -\omega^\top SSS \mu \quad \text{für alle } \omega \in \ker B$$

genau dann, wenn ein $\Lambda \in \mathbb{R}^L$ existiert, sodass

$$\begin{bmatrix} MMM + \tau SSS & B^\top \\ B & 0 \end{bmatrix} \begin{bmatrix} V \\ \Lambda \end{bmatrix} = \begin{bmatrix} -SSS \mu \\ 0 \end{bmatrix}.$$

Nach Lösen dieses Gleichungssystems im k -ten Zeitschritt setzen wir dann den Koeffizientenvektor μ^k der Lösung u_h^k durch

$$\mu^k = \mu^{k-1} + \tau V.$$

4.2 Auswertung

Die Lösung der harmonischen Wärmeleitungsgleichung kann Singularitäten entwickeln und obiges Beispiel aus [Chang et al., 1992] führt zu solch einer Situation und stellt damit ein Worst-Case-Szenario dar. Wir werten den Fehler für $h = 2^{-\ell}$ für $\ell = 4, 5, 6, 7$ und mit $\tau = 2^{-\ell}$ für $\ell = 6, 7, 8, 9$ aus. Dazu betrachten wir den Fehler $\|\mathcal{I}_h|u_h^k|^2 - 1\|_{L^1(\Omega)}$ in Abhängigkeit des Zeitpunkts $k = 1, \dots, K$ und der Schrittweite h . Da wir nur lineare Lösungsfunktionen betrachten nehmen wir als zweidimensionale Quadraturformel die Eckpunktregel, welche lineare Funktionen exakt integriert ([Vogt, 2006]):

$$\begin{aligned} \|\mathcal{I}_h|u_h^k|^2 - 1\|_{L^1(\Omega)} &= \sum_e \int_e |\mathcal{I}_h|u_h^k|^2 - 1| dx \\ &= \sum_e F(e) \frac{1}{3} \left[(\mathcal{I}_h|u_h^k|^2(e_1) - 1) + (\mathcal{I}_h|u_h^k|^2(e_2) - 1) + (\mathcal{I}_h|u_h^k|^2(e_3) - 1) \right] \\ &= \sum_e F(e) \frac{1}{3} \left[(|u_h^k(e_1)|^2 - 1) + (|u_h^k(e_2)|^2 - 1) + (|u_h^k(e_3)|^2 - 1) \right], \end{aligned}$$

wobei $F(e)$ der Flächeninhalt des Dreiecks e mit zugehörigen Ecken/Knoten e_1, e_2, e_3 und dementsprechend $u_h^k(e_i)$ der dreidimensionale Lösungsvektor im i -ten Knoten ist. In folgenden Plots sieht man die Implementierung in Matlab, deren Code im Anhang zu finden ist.

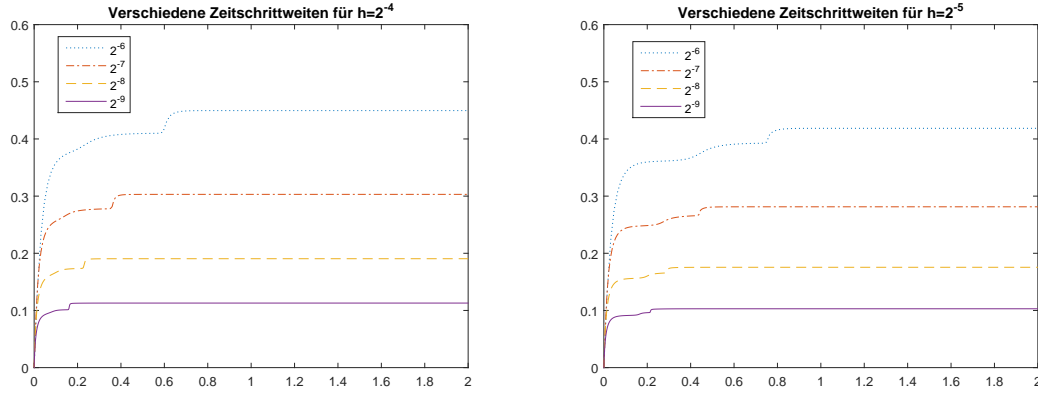


ABBILDUNG 6: Schrittweite $h = 2^{-4}$ bzw. $h = 2^{-5}$ für $\tau = 2^{-\ell}$ mit $\ell = 5, 6, 7, 8$

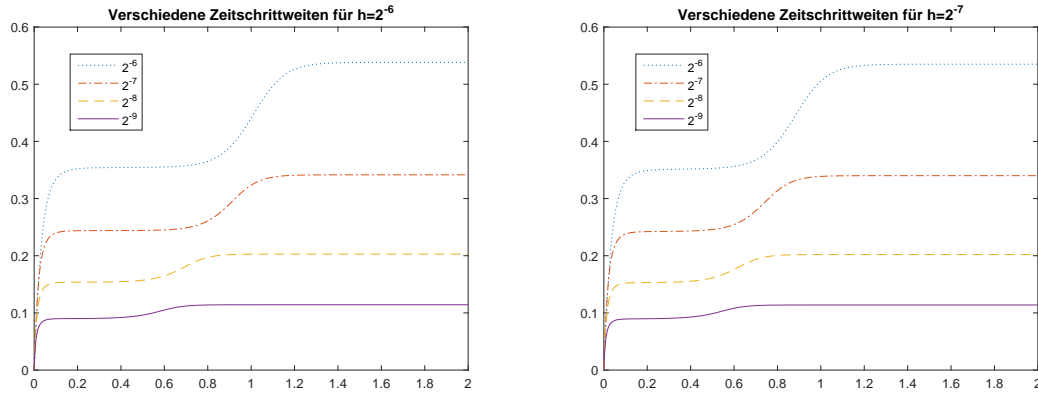


ABBILDUNG 7: Schrittweite $h = 2^{-6}$ bzw. $h = 2^{-7}$ für $\tau = 2^{-\ell}$ mit $\ell = 5, 6, 7, 8$

Man erkennt, dass der Fehler zu Beginn rapide ansteigt, bis er dann, wenn der Wärmefluss stationär wird, nach einem weiteren kurzen Anstieg konstant, also wie in Proposition 1 bewiesen, beschränkt bleibt. Allerdings zeigen die Plots nicht die vollständige lineare Konvergenz wie in Proposition 1 bewiesen. In [Bartels, 2016] wird vermutet, dass dies im Zusammenhang mit der globalen Eigenschaft der Abschätzung zu tun hat, jedoch wird nicht näher darauf eingegangen. Neben der annähernd linearen Konvergenz bewirkt die Schrittweiten senkung eine Glättung der Fehlerkurven in Sprungbereichen. Der Approximationsfehler mag im ersten Moment groß erscheinen, was allerdings die hohe Anfangsenergie $I(u_0) \approx 72,16$ des Ausgangsbeispiels widerspiegelt. Für $h = 2^{-6}, 2^{-7}$ ist ein stärkerer zweiter Anstieg zu erkennen, was vermutlich an der zunehmenden Rundungsfehlerakkumulation liegt, welche bei den kleineren Schrittweiten noch keine größere Rolle spielt.

Literaturverzeichnis

- [Bartels, 2016] Bartels, S. (2016). Projection-free approximation of geometrically constrained partial differential equations. *Mathematics of Computation*, 85(299):1033–1049.
- [Bartels et al., 2015] Bartels, S. et al. (2015). *Numerical methods for nonlinear partial differential equations*, volume 47. Springer.
- [Chang et al., 1992] Chang, K.-C., Ding, W. Y., Ye, R., et al. (1992). Finite-time blow-up of the heat flow of harmonic maps from surfaces. *Journal of Differential Geometry*, 36(2):507–515.
- [Freire, 1995] Freire, A. (1995). Uniqueness for the harmonic map flow from surfaces to general targets. *Commentarii Mathematici Helvetici*, 70(1):310–338.
- [Rupflin, 2010] Rupflin, M. (2010). *Harmonic map flow and variants*. PhD thesis, ETH Zurich.
- [Vogt, 2006] Vogt, W. (2006). Adaptive verfahren zur numerischen quadratur und kubatur. *Preprint No. M*, 1(06).

5 Matlab-Anlagen

Im Folgenden ist die Matlab-Implementation der verschiedenen Funktionen zu finden. Hierbei wurde aufgrund der Übersichtlichkeit auf Kommentare verzichtet. Chronologisch sind folgende Funktionen: Gebietsdefinition, Verfeinerung, Steifigkeitsmatrix, Massematrix, Waermeffluss. Anschließend ist noch die Skript-Datei Beispiel_Singulaer, welche die obigen Plots erzeugt.

```
function [ Knoten,Elemente ] = Gebietsdefinition( Mittelpunkt,Groesse )

if Groesse < 0 | mod(log2(Groesse),1)~=0
    msg = 'Nur Zahlen der Form 2^n, wobei n aus N_0 ist';
    error(msg)
else
    halbiert=0.5*Groesse;
    Knoten(1,1) = Mittelpunkt(1)-halbiert;
    Knoten(1,2) = Mittelpunkt(2)-halbiert;
    Knoten(2,1) = Mittelpunkt(1)+halbiert;
    Knoten(2,2) = Mittelpunkt(2)-halbiert;
    Knoten(3,1) = Mittelpunkt(1)+halbiert;
    Knoten(3,2) = Mittelpunkt(2)+halbiert;
    Knoten(4,1) = Mittelpunkt(1)-halbiert;
    Knoten(4,2) = Mittelpunkt(2)+halbiert;

    Elemente = [1,2,3;1,3,4];

end
end
```

```
function [ KnotenNew,ElementeNew ] = Verfeinerung( Knoten,Elemente,...
                                                Schrittweite )

if Schrittweite < 0 | mod(log2(Schrittweite),1)~=0
    msg = 'Bitte Zahlen der Form 2^-n, wobei n aus N_0 ist';
    error(msg)
end

Gebietslaenge=Knoten(2,1)-Knoten(1,1);
NeueElementeTest=(Gebietslaenge/Schrittweite);

if NeueElementeTest==1
    KnotenNew=Knoten;
    ElementeNew=Elemente;
    return;
elseif NeueElementeTest<1
    msg = 'Schrittweite unzuessaessig (zu gross)';
    error(msg)
end
```

```
KnotenanzahlNew=(Gebietslaenge/Schrittweite+1)^2;
Laenge=sqrt(KnotenanzahlNew);
KnotenNew=zeros(KnotenanzahlNew,2);
k=0;
h=0;
j=1;
while j<KnotenanzahlNew

    if h+Knoten(1,1)>Knoten(length(Knoten)-1,1)
        k=k+Schrittweite;
        h=0;
    end
    KnotenNew(j,1)=Knoten(1,1)+h;
    KnotenNew(j,2)=Knoten(1,2)+k;
    h=h+Schrittweite;
    j=j+1;
end
KnotenNew(j,:)=Knoten(length(Knoten)-1,:);

j=1;
h=1;

ElementeNew=zeros(NeueElementeTest,3);

while j<=KnotenanzahlNew-Laenge
    if 0~=mod(j,Laenge)
        ElementeNew(h,1)=j;
        ElementeNew(h,2)=j+1;
        ElementeNew(h,3)=j+1+Laenge;
        j=j+1;
        h=h+1;
    else
        j=j+1;
    end
end
j=1;

while j<=KnotenanzahlNew-Laenge
    if 0~=mod(j,Laenge)
        ElementeNew(h,1)=j;
        ElementeNew(h,2)=j+Laenge;
        ElementeNew(h,3)=j+1+Laenge;
        j=j+1;
        h=h+1;
    else
        j=j+1;
    end
end

end

function [ S ] = Steifigkeitsmatrix( Knoten,Elemente )
```

```
S=zeros(length(Knoten),length(Knoten));
Schrittweite=Knoten(2,1)-Knoten(1,1);
Flaeche=Schrittweite^2/2;
for j=1:length(Elemente(:,1))
    S(Elemente(j,1),Elemente(j,1))=S(Elemente(j,1),Elemente(j,1))...
        +1/Schrittweite^2*Flaeche;
    S(Elemente(j,1),Elemente(j,2))=S(Elemente(j,1),Elemente(j,2))...
        -1/Schrittweite^2*Flaeche;
    S(Elemente(j,2),Elemente(j,2))=S(Elemente(j,2),Elemente(j,2))...
        +2/Schrittweite^2*Flaeche;
    S(Elemente(j,2),Elemente(j,3))=S(Elemente(j,2),Elemente(j,3))...
        -1/Schrittweite^2*Flaeche;
    S(Elemente(j,3),Elemente(j,3))=S(Elemente(j,3),Elemente(j,3))...
        +1/Schrittweite^2*Flaeche;
end

Transpo=S';
for j=1:length(S)
    Transpo(j,j)=0;
end
S=S+Transpo;
```

```
function [ M ] = Massematrix( coorKnoten,coorElem )
```

```
M=zeros(length(coorKnoten),length(coorKnoten));
Schrittweite=coorKnoten(2,1)-coorKnoten(1,1);
for j=1:length(coorElem(:,1))
    M(coorElem(j,1),coorElem(j,1))=M(coorElem(j,1),coorElem(j,1))...
        +Schrittweite^2/12;
    M(coorElem(j,1),coorElem(j,2))=M(coorElem(j,1),coorElem(j,2))...
        +Schrittweite^2/24;
    M(coorElem(j,1),coorElem(j,3))=M(coorElem(j,1),coorElem(j,3))...
        +Schrittweite^2/24;
    M(coorElem(j,2),coorElem(j,2))=M(coorElem(j,2),coorElem(j,2))...
        +Schrittweite^2/12;
    M(coorElem(j,2),coorElem(j,3))=M(coorElem(j,2),coorElem(j,3))...
        +Schrittweite^2/24;
    M(coorElem(j,3),coorElem(j,3))=M(coorElem(j,3),coorElem(j,3))...
        +Schrittweite^2/12;
end

Transpo=M';
for j=1:length(M)
    Transpo(j,j)=0;
end
M=M+Transpo;
end
```

```
function [u,Fehler]= Waermefluss(h,tau,T,Mittelpunkt,Laenge)
```

```
[Knoten,Elemente] = Gebietsdefinition(Mittelpunkt,Laenge);
```

```

tau = 2^(-tau);
K = ceil(T/tau);
h=2^(-h);

[Knoten,Elemente]=Verfeinerung(Knoten,Elemente,h);

nC = size(Knoten,1);

s=Steifigkeitsmatrix(Knoten,Elemente);
m=Massematrix(Knoten,Elemente);
SSS = sparse(3*nC,3*nC); MMM = sparse(3*nC,3*nC);

for k = 1 : 3
    idx = k:3:3*nC; SSS(idx,idx) = s; MMM(idx,idx) = m;
end

u = zeros(3*nC,1);

for j = 1:nC
    u(3*j-[2,1,0]) = u_0(Knoten(j,:));
end

Fehler=zeros(K+1,1);
for j=1:length(Elemente)
    Fehler(1)=Fehler(1)+(1/2)*(h^2)*(1/3)*...
        (abs(norm(u(3*Elemente(j,1)-[2,1,0]))^2-1)...
        +abs(norm(u(3*Elemente(j,2)-[2,1,0]))^2-1)...
        +abs(norm(u(3*Elemente(j,3)-[2,1,0]))^2-1));
end

for k = 1:K
    B = sparse(nC,3*nC);
    for j = 1:nC
        B(j,3*j-[2,1,0]) = u(3*j-[2,1,0]);
    end
    X = [MMM+tau*SSS,B';B,sparse(nC,nC)];
    b = [-SSS*u;zeros(nC,1)];
    x = X\b;
    v = x(1:3*nC);
    u = u+tau*v;

    for j=1:length(Elemente)
        Fehler(k+1)=Fehler(k+1)+...
            (abs(norm(u(3*Elemente(j,1)-[2,1,0]))^2-1)...
            +abs(norm(u(3*Elemente(j,2)-[2,1,0]))^2-1)...
            +abs(norm(u(3*Elemente(j,3)-[2,1,0]))^2-1));
    end
    Fehler(k+1)=Fehler(k+1)*(h^2)/2*(1/3);

```

```
end

function val = u_0(x)
if x==0
    val=[0,0,1];
else
    d = size(x,2);
    x = [x,zeros(1,3-d)];
    r = norm(x); phi = (3*pi/2)*min(4*r^2,1);
    val=(1/r)*[x(1)*sin(phi),x(2)*sin(phi),r*cos(phi)]';
end

clear all
warning('off','MATLAB:legend:IgnoringExtraEntries');
Plotfenster=1;
Linienvektor={' ','-.-','--','-' };
Beschriftung=zeros(4,1);
Legendenzaehler=0;
for h=4:1:7
    Linienart=1;

    for tau=6:1:9
        Legendenzaehler=Legendenzaehler+1;
        if Linienart>4
            Linienart=1;
        end
        Beschriftung(Legendenzaehler)=tau;
        [~,Fehler]=Waermefluss(h,tau,2,[0,0],1);
        Zeitschritte=zeros(length(Fehler),1);

        for k=0:length(Fehler)-1
            Zeitschritte(k+1)=k*(1/2^(tau));
        end

        figure(Plotfenster);
        plot(Zeitschritte,Fehler,Linienvektor{Linienart})
        legend(sprintf('2^{-%d}','2^{-%d}','2^{-%d}','2^{-%d}',
            Beschriftung(1)),...
            sprintf('2^{-%d}','2^{-%d}','2^{-%d}','2^{-%d}',
            Beschriftung(2)),...
            sprintf('2^{-%d}','2^{-%d}','2^{-%d}','2^{-%d}',
            Beschriftung(3)),...
            sprintf('2^{-%d}','2^{-%d}','2^{-%d}','2^{-%d}',
            Beschriftung(4)));

        title(sprintf('Verschiedene Zeitschrittweiten fuer h=2^{-%d}',h))
        axis([0 2 0 0.6])
        hold on

        Linienart=Linienart+1;
    end
    Plotfenster=Plotfenster+1;
end
```

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht.

.....
Ort, Datum

.....
Unterschrift des Verfassers