

---

# **Gebietszerlegungsmethoden mit spektralen Ansätzen für elliptische Differentialgleichungen**

---

**Fakultät für Mathematik an der  
Universität Duisburg-Essen  
zur Erlangung des akademischen Grades**

**Dr. rer. nat.**

**vorgelegte Dissertation**

**von**

**Zhu Zhai**

Referent: Prof. Dr. Wilhelm Heinrichs

Koreferent: Prof. Dr. Fleurianne Bertrand

Tag der mündlichen Prüfung: 22.12.2022

# DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT  
DUISBURG  
ESSEN

*Offen im Denken*

ub | universitäts  
bibliothek

Diese Dissertation wird via DuEPublico, dem Dokumenten- und Publikationsserver der Universität Duisburg-Essen, zur Verfügung gestellt und liegt auch als Print-Version vor.

**DOI:** 10.17185/duepublico/78272

**URN:** urn:nbn:de:hbz:465-20230420-105720-2

Alle Rechte vorbehalten.

# Abstract

This PhD thesis deals with the numerical solving of elliptic differential equations by domain decompositions. For the solution, the Dirichlet-Neumann method and Neumann-Neumann method are applied to different domains in separate numerical tests. The numerical method carried out is the spectral collocation method.

Domains decomposed into triangles and quadrilaterals are studied first and Poisson equations are solved on them respectively. To find a solution, Dirichlet-Neumann method and Neumann-Neumann method are performed in separate numerical tests using spectral collocation method, which utilizes Fekete points on triangles and Gauss-Lobatto-Legendre points on quadrilaterals as collocation points. The focus of this work is on domain decomposition, domains are decomposed into triangles and local problems are solved using the Neumann-Neumann method in combination with global  $h$ - and  $p$ -refinement and adaptive  $p$ -refinement, while Fekete points as used as collocation points.

To validate the methods they are applied to solve four convection-diffusion equations with a steady state, where the exact solutions are known.

We find out for singular-disturbed problems, such as the convection-diffusion equations considered, that adaptive  $p$ -refinement requires fewer collocation points and also produces comparable qualitative solutions compared to the studied manual  $h$ - and  $p$ -refinement.



# Zusammenfassung

Diese Dissertation befasst sich mit der numerischen Lösung elliptischer Differentialgleichungen durch Gebietszerlegungen. Zur Lösung wird die Dirichlet-Neumann Methode und Neumann-Neumann Methode in getrennten numerischen Tests auf verschiedene Gebiete angewendet. Das ausgeführte numerische Verfahren ist die spektrale Kollokationsmethode.

Es werden zuerst Gebiete untersucht, die in Dreiecke und Vierecke zerlegt sind und darauf Poisson-Gleichungen gelöst. Zur Lösung werden die Dirichlet-Neumann Methode und Neumann-Neumann Methode in separaten numerischen Tests unter Verwendung spektraler Kollokationsmethode durchgeführt, welche Fekete-Punkte auf Dreiecken und Gauß-Lobatto-Legendre-Punkte auf Vierecken als Kollokationspunkte verwendet.

Der Fokus dieser Arbeit liegt auf den Gebietszerlegungen. Die Gebiete werden in Dreiecke zerlegt und lokale Probleme werden mit der Neumann-Neumann Methode in Kombination mit globaler  $h$ - und  $p$ -Verfeinerung sowie adaptiven  $p$ -Verfeinerung gelöst, während Fekete-Punkte als Kollokationspunkte verwendet werden.

Zur Validierung der Methoden werden diese für die Lösung von vier Konvektions-Diffusionsgleichungen eingesetzt, wobei die exakten Lösungen bekannt sind.

Wir finden heraus, dass für singular-gestörte Probleme, wie die betrachteten Konvektions-Diffusionsgleichungen, adaptive  $p$ -Verfeinerung weniger Kollokationspunkte benötigen und auch im Vergleich zur untersuchten globalen  $h$ - und  $p$ -Verfeinerung vergleichbare qualitative Lösungen erzeugen.



# Danksagung

*An dieser Stelle möchte ich allen Menschen danken, die mich bei der Anfertigung meiner Dissertation unterstützt haben.*

*Mein besonderer Dank gilt Herrn Prof. Dr. Heinrichs, der mir die Möglichkeit gegeben hat, diese Arbeit unter seiner Leitung durchzuführen. Außerdem möchte ich mich für die Betreuung und Unterstützung bei der Umsetzung der gesamten Arbeit bedanken. Die vielen interessanten wissenschaftlichen und nicht wissenschaftlichen Gespräche werde ich in guter Erinnerung behalten. Und etliche Cappuccinos später (viele auf Kosten von Ihnen - auch dafür bedanke ich mich) wurde die Dissertation fertig.*

*Ein großes Dankeschön geht an Frau Prof. Dr. Fleurianne Bertrand. Danke, dass du dich mit großem Interesse bereit erklärt hast meine Arbeit als Zweitgutachterin zu lesen und für den kritischen Diskurs.*

*Außerdem möchte ich mich bei Aleksandra und Petra bedanken, die mich nicht nur seit meinem ersten Semester an der Uni begleitet haben, sondern auch inspiriert haben. Danke, dass ihr immer ein offenes Ohr für mich hattet!*

*Was wäre die Promotionszeit ohne gute Freunde an der Uni? Danke, dass durch euch die Zeit viel lustiger war.*

*Meiner Familie und Freunden danke ich für ihre Ermutigungen und Zusprüche während der Arbeit an dieser Dissertation. Ich danke meinen Eltern, die mich immer auf meinem Weg unterstützt und mir alles ermöglicht haben.*



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Gebietszerlegungsmethode</b>	<b>3</b>
2.1	Überlappende Zerlegung . . . . .	6
2.2	Nicht-überlappende Zerlegung . . . . .	7
2.2.1	Dirichlet-Neumann Methode . . . . .	8
2.2.2	Neumann-Neumann Methode . . . . .	9
<b>3</b>	<b>Allgemeine Spektralmethoden</b>	<b>11</b>
3.1	Wahl der Ansatzfunktionen . . . . .	13
3.1.1	Fourier-Reihe . . . . .	14
3.1.2	Chebyshev-Polynome . . . . .	14
3.1.3	Legendre-Polynome . . . . .	17
3.2	Wahl der Testfunktionen . . . . .	19
3.2.1	Kollokation . . . . .	20
3.3	Spektrale Approximation . . . . .	21
3.3.1	Gauß-Lobatto-Quadratur und Interpolation . . . . .	21
3.3.2	Gauß-Lobatto-Legendre-Punkte und Legendre-Interpolation . . . . .	22
3.3.3	Legendre-Polynome in 2D . . . . .	24
3.3.4	2D Gauß-Lobatto-Legendre-Punkte . . . . .	25
<b>4</b>	<b>Dreiecksbasierte Spektralmethode</b>	<b>27</b>
4.1	Dubiner-Polynome . . . . .	27
4.1.1	Ableitungen der Dubiner-Polynomen . . . . .	33
4.2	Fekete-Punkte . . . . .	36
4.3	Spektrale Differentiation auf dem Standarddreieck $T_0$ . . . . .	39
4.4	Formulierung auf einem allgemeinen Dreieck $T_k$ . . . . .	41
4.5	Interface Permutationen . . . . .	43
<b>5</b>	<b>Numerische Analysis der dreiecksbasierten Spektralmethode</b>	<b>45</b>
5.1	Approximationsgüte von Fekete-Punkten auf Dreiecken . . . . .	46
5.2	Konvergenzüberlegungen zur Kollokationsmethode . . . . .	47

5.3	Effektivität von Dirichlet-Neumann und Neumann-Neumann Methoden	49
5.3.1	Dirichlet-Neumann Iteration	50
5.3.2	Neumann-Neumann Iteration	52
<b>6</b>	<b>Numerische Ergebnisse für die Dirichlet-Neumann Iteration</b>	<b>55</b>
6.1	Numerischer Test 1	56
6.2	Numerischer Test 2	60
6.3	Numerischer Test 3	65
6.4	Numerischer Test 4	69
<b>7</b>	<b>Numerische Ergebnisse für die Neumann-Neumann Iteration</b>	<b>75</b>
7.1	Vergleich von Dirichlet-Neumann und Neumann-Neumann Iteration	76
7.1.1	Numerischer Test 1	76
7.1.2	Numerischer Test 2	80
7.1.3	Numerischer Test 3	83
7.1.4	Numerischer Test 4	87
7.2	Betrachtung von $h$ - und $p$ -Verfeinerung	90
7.2.1	Klassische Poisson-Gleichung	92
7.2.2	Konvektions-Diffusionsgleichung	99
7.2.2.1	Moderater Boundary-Layer	103
7.2.2.2	Verschärfter Boundary-Layer	107
7.2.2.3	Eckenlayer	111
7.2.2.4	Hut-Layer	113
7.3	Adaptive $p$ -Verfeinerung	119
7.3.1	Spektrale Erweiterungskoeffizienten	120
7.3.2	Koeffizienten entlang der Spur des Spektrums	122
7.3.3	Adaptive $p$ -Verfeinerung der Konvektion-Diffusionsgleichung	122
7.3.3.1	Moderate Boundary-Layer	123
7.3.3.2	Verschärfter Boundary-Layer	126
7.3.3.3	Eckenlayer	130
7.3.3.4	Hut-Layer	133
<b>8</b>	<b>Fazit und Ausblick</b>	<b>143</b>
<b>A</b>	<b>Dubiner-Polynome</b>	<b>147</b>
A.1	Dubiner-Basis	147
A.2	Ableitungen der Dubiner-Polynome	148
<b>B</b>	<b>Fekete-Punkte</b>	<b>149</b>

# Kapitel 1

## Einleitung

In der Ingenieurwissenschaft spielen partielle Differentialgleichungen eine wichtige Rolle, um Modelle aus Natur und Technik zu behandeln. Die durchgeführten numerischen Simulationen sollen dabei effizient implementiert werden. Deshalb forschen wir nach Methoden, die speichereffizient und schnell unter Berücksichtigung von Hardware-Einstellungen sind. In den letzten Jahrzehnten lag ein großes Interesse an Methoden zur Gebietszerlegung vor (zum Beispiel [GK02]). Gebietszerlegung bezieht sich auf Teilungstechniken zum Lösen partieller Differentialgleichungen durch iteratives Lösen von Teilproblemen, die auf kleineren Teilgebieten definiert sind. Zu den Hauptvorteilen gehören die Verbesserung der Parallelisierbarkeit, die lokalisierte Behandlung komplexer und unregelmäßiger Geometrien und Singularitäten. Zusätzlich reduziert die Gebietszerlegung in einigen Fällen den Rechenaufwand der zugrunde liegenden Lösungsmethode [CM<sup>+</sup>94]. Spektralmethoden liefern dabei gute Konvergenzeigenschaften. Numerische Experimente zeigen die erwartete hohe spektrale Genauigkeit für glatte Lösungen [Tre00]. Die effiziente Lösung elliptischer Gleichungen ist hier von zentralem Interesse. Auf Tensorgebieten werden dabei die Gauß-Lobatto-Punkte zum Lösen der Differentialgleichungen verwendet. Auf Dreiecken, die keine Tensorgebiete sind, werden entweder durch Transformation die Gauß-Lobatto-Punkte verwendet oder die sogenannten Fekete-Punkte, die fast optimale Interpolationseigenschaften liefern [BTW01], [PR04], [Hei98].

Aktuellere Forschungen zeigen, dass Gebietszerlegungen mit Kollokationsmethoden in 3D für einen L-förmigen Körper gute Genauigkeiten aufweisen [CLT16]. Es gibt viele physikalische Probleme auf komplizierten Geometrien im Ingenieurbereich, die mit Hilfe von verschiedenen Ansätzen der Gebietszerlegungsmethode gelöst werden. Hierzu verweisen wir auf folgende aktuelle Forschungsergebnisse: [BMN19], [RL18], [AG16].

In dieser Arbeit werden Gebietszerlegungsmethoden mit Dirichlet-Neumann und

Neumann-Neumann Methoden für elliptische Differentialgleichungen betrachtet. Dabei verwenden wir als numerisches Verfahren die spektrale Kollokationsmethode. Wir verwenden auf Dreiecksgebieten als Kollokationspunkte die Fekete-Punkte und auf Tensorgebieten die Gauß-Lobatto-Punkte und möchten herausfinden, welche Methode im Hinblick auf Anzahl der Iterationsschritte und Größe der Fehler auf den Gebieten effizienter ist. Zusätzlich vergleichen wir unsere Ergebnisse mit denen aus [HL01]. Dort wird mit Chebyshev-Gauß-Lobatto-Punkten die Dirichlet-Neumann Methode verwendet. Ebenfalls untersuchen wir die Effizienz der Neumann-Neumann Methode in Kombination mit  $h$ - und  $p$ -Verfeinerung sowie adaptiven  $p$ -Verfeinerung anhand von Poisson-Gleichung und singular-gestörte Probleme wie Konvektion-Diffusionsgleichungen.

Die Arbeit ist wie folgt aufgebaut:

Kapitel 2 beschreibt zwei Arten der Gebietszerlegungsmethode und geht dabei detaillierter auf zwei Methoden der nicht-überlappenden Zerlegung ein. Kapitel 3 stellt die Grundidee der Spektralmethoden vor und präsentiert die Kollokationspunkte, die wir für die viereckigen Gebiete benötigen. In Kapitel 4 gehen wir auf einige Besonderheiten der Dreiecksgebiete ein. Hierbei stellen wir eine Polynombasis und eine Reihe von Kollokationspunkten vor, die beide für Dreieckselemente geeignet sind. Wir leiten ab, wie die spektrale Differentiation auf einem beliebigen Dreieck durchgeführt wird. Außerdem erläutern wir, wie wir das Problem benachbarter Dreiecke, die die gleichen Interfacepunkte teilen, lösen werden. Kapitel 5 beleuchtet den theoretischen Hintergrund der verwendeten Verfahren. Kapitel 6 zeigt die numerischen Ergebnisse der Dirichlet-Neumann Methode verschiedener Gebiete angewandt auf das Poisson-Problem und Kapitel 7 präsentiert den Hauptbestandteil dieser Arbeit. Dabei wenden wir die Neumann-Neumann Methode auf das Poisson-Problem und auf komplexere Konvektions-Diffusionsgleichungen an und kombinieren diese mit globaler  $h$ - und  $p$ -Verfeinerung und adaptiven  $p$ -Verfeinerung. Das letzte Kapitel, Kapitel 8, umfasst das Fazit der numerischen Experimente und gibt einen Ausblick, worauf in Zukunft aufgebaut werden könnte.

# Kapitel 2

## Gebietszerlegungsmethode

Die Idee hinter der vorgestellten Gebietszerlegungsmethode ist, ein Randwertproblem auf dem Integrationsgebiet  $\Omega$  durch Teilgebiete  $\Omega_i$  zu zerlegen, anstatt es auf dem gesamten Integrationsgebiet zu diskretisieren, anschließend das resultierende Gleichungssystem aufzustellen und mit einem geeigneten Verfahren zu lösen. Sie ermöglichen die Neuformulierung eines Randwertproblems, welche meistens so groß sind, dass sie schwer numerisch zu berechnen sind, auf dem zu berechneten Gebiets in Teilgebiete, was zu einer reduzierten Größe führt [QV99]. Mit Hilfe von Gebietszerlegung kann die Lösung eines großen linearen Systems in kleinere Probleme unterteilt werden, deren Lösungen können verwendet werden, um eine Vorkonditionierung oder einen Löser für das Gleichungssystem zu erzeugen, das sich aus der Diskretisierung der partiellen Differentialgleichung auf dem gesamten Gebiet ergibt [Smi96]. Häufig wird die Methode bei komplizierten Geometrien oder Strömungsprobleme mit unregelmäßigen Geometrien verwenden.

In diesem Kapitel befassen wir uns exemplarisch mit zwei Gebietszerlegungsarten, der überlappenden und nicht-überlappenden Zerlegung. Im ersten Abschnitt wird die überlappende Zerlegung und im Folgeabschnitt die nicht-überlappende Zerlegung kurz vorgestellt. Weiterhin werden bei der nicht-überlappenden Gebietszerlegungsmethode die Verfahren Dirichlet-Neumann und Neumann-Neumann Iterationen präsentiert, wobei wir innerhalb dieser Arbeit einen Fokus auf die Neumann-Neumann Iteration legen.

Die allgemeine lineare Differentialgleichung zweiter Ordnung in zwei Variablen lautet

$$a(x, y) \frac{\partial^2}{\partial x^2} u + 2b(x, y) \frac{\partial^2}{\partial x \partial y} u + c(x, y) \frac{\partial^2}{\partial y^2} u + d(x, y) \frac{\partial}{\partial x} u + e(x, y) \frac{\partial}{\partial y} u + f(x, y) u + g(x, y) = 0.$$

Sie heißt elliptisch in  $(x, y)$ , falls  $a(x, y)c(x, y) - b^2(x, y) > 0$ .

Das allgemeine elliptische Randwertproblem mit Dirichlet-Randbedingung hat die Form

$$\begin{aligned}\mathcal{L}u &= f \quad \text{auf } \Omega \subset \mathbb{R}^2, \\ u &= 0 \quad \text{auf } \partial\Omega,\end{aligned}$$

wobei  $\mathcal{L}$  ein elliptischer, linearer Operator ist.

Ein Spezialfall des soeben beschriebenen Problems ist die Poisson-Gleichung mit homogener Dirichlet-Randbedingung:

$$\begin{aligned}-\Delta u &= f \quad \text{auf } \Omega \subset \mathbb{R}^2, \\ u &= 0 \quad \text{auf } \partial\Omega.\end{aligned}$$

Die Poisson-Gleichung ist eine PDE zweiter Ordnung und kann mit einer Dirichlet- oder Neumann-Randbedingung versehen werden. Die Poisson-Gleichung spielt eine fundamentale Rolle in der Elektrostatik, Astrophysik und Quantenchemie.

Sei  $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}_0^n$  ein Multiindex mit der Konvention  $|\alpha| = \alpha_1 + \dots + \alpha_n$ ,  $x^\alpha = x_1^{\alpha_1} \cdot \dots \cdot x_n^{\alpha_n}$  und  $D^\alpha u = \left( \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \dots \frac{\partial^{\alpha_n}}{\partial x_n^{\alpha_n}} \right) u$ . Sei im Folgenden  $\Omega \subset \mathbb{R}^n$  offen und beschränkt.

**Definition 2.0.1.** Eine Funktion  $u \in L^1_{loc}(\Omega)$  besitzt eine  $\alpha$ -te schwache Ableitung in  $\Omega$ , wenn es eine Funktion  $v \in L^1_{loc}(\Omega)$  gibt, sodass

$$\int_{\Omega} u D^\alpha \phi \, dx = (-1)^{|\alpha|} \int_{\Omega} v \phi \, dx \quad \text{für alle } \phi \in C_0^\infty(\Omega).$$

Die Funktion  $v \in L^1_{loc}(\Omega)$  ist eindeutig bestimmt und wir schreiben  $v = D^\alpha u$ .

**Definition 2.0.2.** Für  $1 \leq p \leq \infty$  und  $k \in \mathbb{N}$  heißt

$$W^{k,p}(\Omega) = \{u \in L^p(\Omega) \mid D^\alpha u \in L^p(\Omega) \quad \text{für alle } |\alpha| \leq k\}$$

Sobolev-Raum.

**Bemerkung 2.0.1.** Betrachten wir den Spezialfall  $H^1(\Omega) := W^{1,2}(\Omega)$ , so ist dieser Sobolev-Raum ausgestattet mit der Norm

$$\|u\|_{H^1} = \left( \|u\|_{L^2}^2 + \sum_{i=1}^n \|D_i u\|_{L^2}^2 \right)^{\frac{1}{2}}$$

---

und dem Skalarprodukt

$$\langle u, v \rangle_{H^1} = \langle u, v \rangle_{L^2} + \sum_{i=1}^n \langle D_i u, D_i v \rangle_{L^2}$$

ein Hilbertraum, wobei  $D_i = D^{\alpha_i}$  mit  $\alpha_i = (0, \dots, 0, \underbrace{1}_{i\text{-te Stelle}}, 0, \dots, 0)$ .

Sei

$$H_0^1(\Omega) = \{u \in H^1(\Omega) \mid u = 0 \text{ auf } \partial\Omega\}$$

der Unterraum aller Funktionen des Sobolev-Raumes  $H^1(\Omega)$ , die am Rand  $\partial\Omega$  verschwindet, wobei  $u = 0$  auf  $\partial\Omega$  im Sinne des Spursatzes (s. [QV08]) zu interpretieren ist.  $H_0^1(\Omega)$  ist auch ein Hilbertraum.

Die schwache Formulierung des Poisson-Problems mit homogenen Dirichlet-Randbedingungen lässt sich nach Einführung der oben genannten Funktionenräume folgendermaßen definieren:

Finde  $u \in H_0^1(\Omega)$ , sodass

$$\int_{\Omega} (\nabla u, \nabla v) \, d\Omega = \int_{\Omega} f v \, d\Omega \quad \text{für alle } v \in H_0^1(\Omega).$$

Wir werden in unseren numerischen Tests in Kapitel 6 und 7 gut gestellte Probleme behandeln, das heißt, die Anforderungen an Existenz und Eindeutigkeit der Lösung, sowie die stetige Abhängigkeit der Lösung von den Daten sind erfüllt. Mit Hilfe vom Satz von Lax-Milgram lässt sich Existenz und Eindeutigkeit von schwachen Lösungen für elliptische Probleme beweisen.

**Lemma 2.0.1** (Satz von Lax-Milgram). *Es sei  $H$  ein Hilbertraum und  $a : H \times H \rightarrow \mathbb{R}$  eine Bilinearform. Es sei  $a$  stetig ( $a(x, y) \leq c_0 \|x\| \|y\|$ ) und koerzitiv ( $a(x, x) \geq c_1 \|x\|^2$ ). Außerdem sei  $F \in H'$ . Dann gibt es eine eindeutige Lösung des Problems: Finde  $u \in H$ , sodass*

$$a(v, u) = F(v) \text{ für alle } v \in H.$$

**Beweis.** Die Beweisidee ist in [Bre11] nachzulesen.

## 2.1 Überlappende Zerlegung

Wir beschäftigen uns in diesem Kapitel mit dem zuvor eingeführten Poisson-Problem mit homogener Dirichlet-Randbedingung:

$$\begin{aligned} -\Delta u &= f & \text{auf } \Omega \subset \mathbb{R}^2, \\ u &= 0 & \text{auf } \partial\Omega. \end{aligned}$$

Wir betrachten eine Zerlegung von  $\Omega = \Omega_1 \cup \Omega_2$  mit Überlappung:

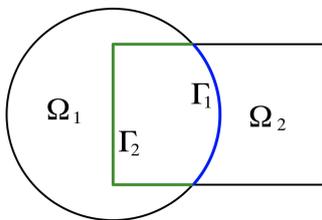


Abbildung 2.1: Gebietszerlegung mit Überlappung

Wir bezeichnen den Teil des Randes der Teilgebiete  $\Omega_1$  und  $\Omega_2$ , der nicht Teil des eigentlichen Randes  $\partial\Omega$  ist, als künstlichen inneren Rand. In Abbildung 2.1 sehen wir, dass  $\Gamma_1$  bzw.  $\Gamma_2$  der künstliche innere Rand des Teilgebiets  $\Omega_1$  bzw.  $\Omega_2$  ist. Um die Poisson-Gleichung auf einem Kreis und einem Rechteck separat zu lösen, schlug Schwarz ein iteratives Verfahren vor, um die approximierte Lösung auf dem gesamten Gebiet  $\Omega = \Omega_1 \cup \Omega_2$  zu finden. In [Cai03] wird das Verfahren ausführlich behandelt. Sei  $u_i^n$  die approximierte Lösung auf dem Teilgebiet  $\Omega_i$  und  $f_i$  die Beschränkung von  $f$  auf  $\Omega_i$ . Wir starten mit  $u_0$  und iterieren für  $n = 1, 2, \dots$ , um sukzessiv bessere Näherungslösungen  $u_1, u_2, \dots$  auf  $\Omega$ , zu finden. Bei jeder Iteration lösen wir zuerst die auf den Kreis  $\Omega_1$  beschränkte Poisson-Gleichung unter Verwendung der Lösung der vorherigen Iteration von  $\Omega_2$  auf dem künstlichen inneren Rand  $\Gamma_1$ :

$$\begin{aligned} -\Delta u_1^n &= f_1 & \text{auf } \Omega_1, \\ u_1^n &= 0 & \text{auf } \partial\Omega_1 \setminus \Gamma_1, \\ u_1^n &= u_2^{n-1} & \text{auf } \Gamma_1. \end{aligned}$$

Danach lösen wir die Poisson-Gleichung innerhalb des Rechtecks  $\Omega_2$  durch die Verwendung der Lösung  $u_1^n$  auf dem künstlichen inneren Rand  $\Gamma_2$ :

$$\begin{aligned} -\Delta u_2^n &= f_2 && \text{auf } \Omega_2, \\ u_2^n &= 0 && \text{auf } \partial\Omega_2 \setminus \Gamma_2, \\ u_2^n &= u_1^n && \text{auf } \Gamma_2. \end{aligned} \tag{2.1}$$

Die zwei lokalen Poisson-Gleichungen in  $\Omega_1$  und  $\Omega_2$  sind wie folgt zusammengekoppelt: die künstliche Dirichlet-Bedingung am inneren Rand  $\Gamma_1$  vom Teilgebiet  $\Omega_1$  wird von Teilgebiet  $\Omega_2$  in Form von  $u_2^{n-1}|_{\Gamma_1}$  bereitgestellt und andersherum.  $u_2^{n-1}|_{\Gamma_1}$  und  $u_1^n|_{\Gamma_2}$  ändern sich von Iteration zu Iteration, während  $u_n$  gegen die analytische Lösung konvergiert. Daher müssen die beiden Poisson-Gleichungen in jedem Iterationsschritt die künstlichen Dirichlet-Bedingungen auf  $\Gamma_1$  und  $\Gamma_2$  durch Austausch von Daten aktualisieren. Wir müssen beachten, dass die klassische alternierende Schwarz-Methode sequenziell ist, was bedeutet, dass die zwei Poisson-Lösungen innerhalb jeder Iteration in einer festgelegten Reihenfolge ausgeführt werden müssen. Natürlich können wir auch das Rechteck als  $\Omega_1$  und den Kreis als  $\Omega_2$  wählen, ohne dass es bemerkbare Auswirkungen auf die Konvergenz hat. Wir unterscheiden bei der Schwarz-Methode zwei Varianten: die multiplikative Schwarz-Methode und die additive Schwarz-Methode. Die obige Iteration ist die multiplikative Schwarz-Methode und für die additive verwenden wir statt (2.1) dann  $u_2^n = u_1^{n-1}$  auf  $\Gamma_2$ . Den Schwarz-Algorithmus für spektrale Methoden kann der Leser in [Zan87] nachlesen.

## 2.2 Nicht-überlappende Zerlegung

Zur Beschreibung von nicht-überlappenden Zerlegungen teilen wir das Gebiet  $\Omega = \Omega_1 \cup \Omega_2$  in zwei disjunkte Teilgebiete  $\Omega_1$  und  $\Omega_2$  auf.

Wie im vorherigen Abschnitt betrachten wir das Poisson-Problem mit homogener Dirichletbedingung auf  $\Omega$ .

Angenommen  $u$  sei eine Lösung des Poisson-Problems. Damit lassen sich  $u|_{\Omega_i} =: u_i$  für  $i = 1, 2$  definieren, wobei  $u_i$  die Lösungen von Teilproblemen

$$\begin{aligned} -\Delta u_i &= f && \text{auf } \Omega_i, \\ u_i &= 0 && \text{auf } \partial\Omega_i \setminus \Gamma, \end{aligned}$$

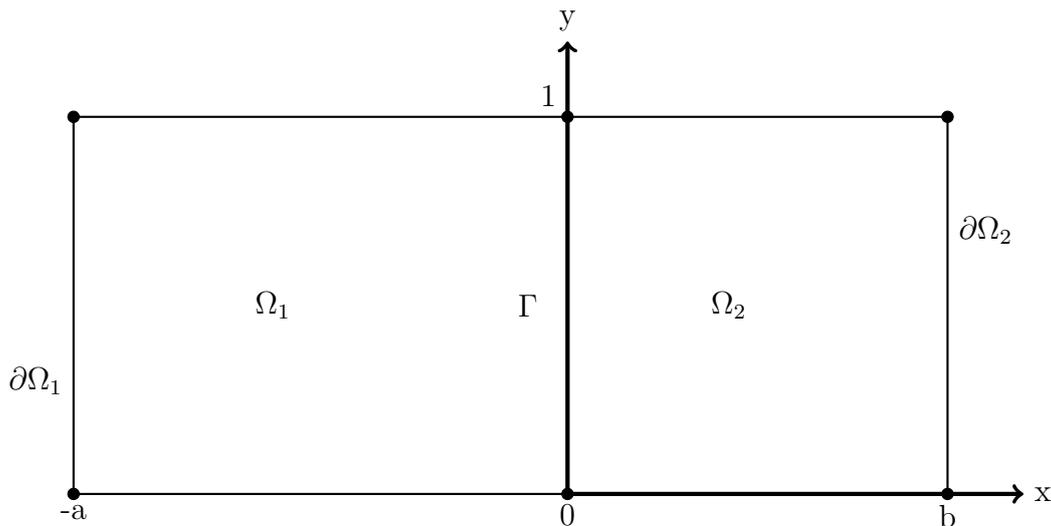


Abbildung 2.2: Gebietszerlegung in zwei nicht-überlappende Teilgebiete

unter Hinzunahme von Interface-Bedingungen

$$\begin{aligned} u_1 &= u_2 && \text{auf } \Gamma, \\ \frac{\partial u_1}{\partial n_1} &= -\frac{\partial u_2}{\partial n_2} && \text{auf } \Gamma \end{aligned}$$

ist.

Somit lässt sich das auf  $\Omega$  definierte Poisson-Problem auf Teilgebiete unter Hinzunahme von Interface-Bedingungen lösen und anschließend zu einer Lösung auf ganz  $\Omega$  vereinigen. Andersherum gesehen bilden Lösungen auf den Teilgebieten unter Hinzunahme der Interface-Bedingungen Lösungen auf ganz  $\Omega$ . Diese Tatsache wird in den folgenden Abschnitten als Dirichlet-Neumann und Neumann-Neumann Methode diskutiert.

### 2.2.1 Dirichlet-Neumann Methode

Eine Methode, um Lösungen auf nicht-überlappende Teilgebiete zu berechnen, ist die Dirichlet-Neumann Methode. Der grundlegende Dirichlet-Neumann Algorithmus besteht aus zwei Schritten, die den zwei Teilgebieten  $\Omega_i, i = 1, 2$ , nach Abbildung 2.2, entsprechen. Wir lösen bei der Anfangsannahme  $u_\Gamma^0$  ein Dirichlet-Problem in  $\Omega_1$  mit Dirichlet Daten  $u_\Gamma^0$  auf  $\Gamma$  und dann ein gemischtes Dirichlet-Neumann-Problem auf  $\Omega_2$  mit einer Neumann-Bedingung auf  $\Gamma$ , welches durch die im vorherigen Schritt erhaltene Lösung und mit Dirichlet-Bedingungen auf dem Rest von  $\partial\Omega_2$  bestimmt wird. Die neue Iteration  $u_\Gamma^1$  wird als Spur der Lösung in  $\Omega_2$  oder allgemeiner als lineare Kombination dieser Spur und  $u_\Gamma^0$  unter Verwendung eines geeigneten gewähl-

ten Relaxationsparameters  $\theta$  gewählt. Wir iterieren die Lösungen auf dem Interface  $u_\Gamma^1, u_\Gamma^2, \dots$  und sobald die Werte auf dem Interface auf vorher definierter Weise konvergieren, lösen wir die Teilgebiete, um eine Lösung zu erhalten.

Die Iteration  $n$ , für  $n \geq 0$ , besteht aus zwei Teilschritten, wobei  $u_\Gamma^0$  vorgegeben wird [BW86]:

1. Löse das Dirichlet-Problem auf  $\Omega_1$ , benutze  $u_\Gamma^n$  als Dirichlet Wert auf  $\Gamma$ :

$$\begin{aligned} -\Delta u_1^{n+1} &= f && \text{auf } \Omega_1, \\ u_1^{n+1} &= 0 && \text{auf } \partial\Omega_1 \setminus \Gamma, \\ u_1^{n+1} &= u_\Gamma^n && \text{auf } \Gamma. \end{aligned} \tag{2.2}$$

2. Löse das Neumann-Problem auf  $\Omega_2$  durch Anpassen der Normalenableitungen mit  $u_1^{n+1}$ :

$$\begin{aligned} -\Delta u_2^{n+1} &= f && \text{auf } \Omega_2, \\ u_2^{n+1} &= 0 && \text{auf } \partial\Omega_2 \setminus \Gamma, \\ \frac{\partial u_2^{n+1}}{\partial n_2} &= -\frac{\partial u_1^{n+1}}{\partial n_1} && \text{auf } \Gamma. \end{aligned} \tag{2.3}$$

Passe das Interface an:

$$u_\Gamma^{n+1} = (1 - \theta)u_\Gamma^n + \theta u_2^{n+1} \quad \text{auf } \Gamma, \quad 0 < \theta \leq 1.$$

Die Wahl des Relaxationsparameter  $\theta$  für das Interface hat Einfluss auf die Konvergenz des Verfahrens bezüglich Geschwindigkeit und Konvergenzverhalten. Diese Schritte werden wiederholt, bis  $u_\Gamma$  zu einer genügend glatten Lösung  $u$  auf  $\Gamma$  führt.

### 2.2.2 Neumann-Neumann Methode

Eine weitere Methode, um Lösungen auf nicht-überlappende Teilgebiete zu berechnen, ist die Neumann-Neumann Methode. Analog zur Dirichlet-Neumann Methode wollen wir den grundlegende Neumann-Neumann Algorithmus beschreiben: Wir beginnen mit der Anfangsannahme  $u_\Gamma^0$  und lösen Dirichlet-Probleme auf jedem  $\Omega_i$ ,  $i = 1, 2$ , nach Abbildung 2.2, mit Daten  $u_\Gamma^0$  auf  $\Gamma$  und danach ein Problem auf jedem Teilgebiet mit Neumann Daten auf  $\Gamma$ , welche als Differenz der Normalenableitungen der Lösungen der zwei Dirichlet-Probleme gewählt wird. Die Werte auf  $\Gamma$  der Lösungen dieser Neumann-Probleme werden dann verwendet, um  $u_\Gamma^0$  vom Anfang

zu korrigieren und die neue iterierte  $u_\Gamma^1$  zu bestimmen.

Für  $n \geq 0$  lösen wir das Problem wie folgt, wobei  $u_\Gamma^0$  vorgegeben wird [TW06]:

1. Löse das Dirichlet-Problem auf  $\Omega_i, i = 1, 2$ :

$$\begin{aligned} -\Delta u_i^{n+1} &= f && \text{auf } \Omega_i, \\ u_i^{n+1} &= 0 && \text{auf } \partial\Omega_i \setminus \Gamma, \\ u_i^{n+1} &= u_\Gamma^n && \text{auf } \Gamma. \end{aligned} \tag{2.4}$$

2. Löse das Neumann-Problem zur Verbesserung  $\psi_i^{n+1}$  auf  $\Omega_i, i = 1, 2$ :

$$\begin{aligned} -\Delta \psi_i^{n+1} &= 0 && \text{auf } \Omega_i, \\ \psi_i^{n+1} &= 0 && \text{auf } \partial\Omega_i \setminus \Gamma, \\ \frac{\partial \psi_i^{n+1}}{\partial n_i} &= \frac{\partial u_1^{n+1}}{\partial n_1} + \frac{\partial u_2^{n+1}}{\partial n_2} && \text{auf } \Gamma. \end{aligned} \tag{2.5}$$

3. Ergänze die Dirichlet Spur :

$$u_\Gamma^{n+1} = u_\Gamma^n - \theta(\psi_1^{n+1} + \psi_2^{n+1}) \quad \text{auf } \Gamma, \quad 0 < \theta \leq 1.$$

Die Iterationsschritte werden wiederholt, bis  $u_\Gamma$  zu einer genügend glatten Lösung  $u$  auf  $\Gamma$  führt.

# Kapitel 3

## Allgemeine Spektralmethoden

Differentialgleichungen sind wichtig bei der Behandlung vieler mathematischer, physikalischer, biologischer oder chemischer Fragestellungen. In der Physik werden zum Beispiel stationäre Temperaturverteilungen durch partielle Differentialgleichungen beschrieben. Diese haben die Form

$$\mathcal{L}u = f \text{ in } \Omega \subset \mathbb{R}^n \quad (3.1)$$

mit Randbedingungen

$$\mathcal{B}u = 0 \text{ auf } \partial\Omega, \quad (3.2)$$

wobei  $\mathcal{L}$  und  $\mathcal{B}$  lineare Differentialoperatoren sind.

Häufig lassen sich analytische Lösungen nur unter großem Aufwand oder gar nicht herleiten, weswegen numerische Lösungsverfahren zum Einsatz kommen. Mit Hilfe numerischer Verfahren kann die exakte Lösung zwar in den meisten Fällen nicht berechnet werden, aber sie lässt sich hinreichend genau approximieren, und für die meisten praktischen Anwendungen genügt eine Annäherung an die Lösung. Zwei bekannte Verfahren, mit denen eine approximierte Lösung von Differentialgleichung ermittelt werden können, sind die Finite-Differenzen-Methode (FDM) und Finite-Element-Methode (FEM). Die Vorteile der FDM sind die einfache Herleitung und leichte Implementierung für einfache Geometrien. Bei komplexen Geometrien sind sie jedoch weniger geeignet, da sie in ihrer Grundform auf rechteckige Formen beschränkt sind, eine Erweiterung auf komplexeren Geometrien bedeutet zusätzlichen Aufwand. Der Vorteil der FEM besteht darin, dass sie leicht bei beliebigen Geometrien angewendet werden kann, da sie in endlich viele Teilgebiete aufgeteilt werden, die als finite Elemente bezeichnet werden. Der Nachteil bei der FEM ist, dass sie für kleine Polynomgrade eine geringe Konvergenzrate aufweist und daher große, lineare Gleichungssysteme, wenn auch meist schwach besetzt, gelöst werden müssen, um

eine akzeptable Genauigkeit der Lösung zu erzielen.

Während des 20. Jahrhunderts wurde eine weitere Klasse neuer Methoden entwickelt, um Differentialgleichungen der Form (3.1) zu lösen. Diese werden als Spektralmethoden bezeichnet. Dabei ist die Wahl der Ansatzfunktionen (auch als Approximationsfunktionen bezeichnet) und die Wahl der Testfunktionen (auch als Gewichtsfunktionen bekannt) der Hauptbestandteil der Spektralmethoden [CHQZ10]. Die drei häufigsten Arten von Spektralmethoden unterscheiden sich durch die Auswahl der Testfunktionen, nämlich die Galerkin-, die Kollokations- und die Tau-Methode, welche im Abschnitt 3.2 erläutert werden. Die Galerkin-Methode wurde erstmals 1954 von Silbermann für die meteorologische Modellierung angewendet [Sil54]. 1969/1970 wurde die Galerkin-Methoden jedoch erst für hochauflösende Berechnungen nichtlinearer Probleme praktisch, nachdem Orszag [Ors69], [Ors72] und Eliassen, Machenhauer und Rasmussen [EMR70] Transformationsmethoden zur Bewertung der Faltungssummen entwickelten. Die Tau-Methode wurde erstmals 1938 von Lanczos [Lan38] entwickelt und 1971 durch weitere Ansätze von Orszag [Ors71] weiterentwickelt. 1972 wurde die Kollokationsmethode, auch Pseudospektrale Methode genannt, von Kreiss und Olinger [KO72] und Orszag [Ors72] entwickelt.

Spektralmethoden bieten für analytische Funktionen eine spektrale Konvergenz, sie konvergieren schneller als  $\frac{1}{N}$ , wobei  $N$  die Dimension der Basis der Ansatz-Funktionen ist. Das Problem besteht darin, dass die zu lösenden linearen Gleichungssysteme durch vollbesetzte Matrizen dargestellt werden. Außerdem können sie nicht auf komplexe Geometrien angewendet werden. 1984 wurde von Patera die Spektral-Element-Methode (SEM) eingeführt. Diese kombiniert die Finite-Element-Methode (FEM) mit der Genauigkeit der Spektralmethode [Pat84].

Spektralmethoden basieren auf der Methode der gewichteten Residuen (MWR). Für eine detaillierte Analyse der MWR wird der Leser auf [FS66] verwiesen.

Die Idee besteht darin, die unbekannte Funktion  $u$  aus (3.1) durch eine Linearkombination von Ansatz- oder Basisfunktion  $\phi_k(x)$

$$\tilde{u}(x) = \sum_{k=0}^N u_k \phi_k(x) \tag{3.3}$$

zu approximieren, wobei  $u_k$  unbekannte, zu bestimmende Koeffizienten sind und  $\tilde{u}$  eine Näherungslösung von (3.1) sein soll. Dementsprechend kann das Residuum  $R$

gegeben durch

$$R = \mathcal{L}\tilde{u} - f$$

berechnet werden. Da  $\tilde{u}$  sich von der Lösung  $u$  unterscheidet und die Differentialgleichung nicht exakt löst, verschwindet das Residuum  $R$  nicht für alle  $x \in \Omega$ . Ziel ist es  $u_k$  so zu bestimmen, dass sich die Funktion  $\tilde{u}$  der exakten Lösung  $u$  bestmöglich annähert. Zu diesem Zweck wählen wir im nächsten Schritt geeignete Testfunktionen  $\omega_k(x)$ ,  $k = 0, \dots, N$ , sodass die Residuumsfunktion  $R(x; u_0, \dots, u_N) = \mathcal{L}\tilde{u} - f$  bezüglich den Koeffizienten  $\{u_k\}$  minimiert wird. Das Residuum soll mit den Testfunktionen  $\omega_k(x)$  multipliziert und über das Gebiet integriert gleich Null sein:

$$\int_{\Omega} \omega_k(x) \cdot R \, dx = 0 \quad \text{für alle } k = 0, \dots, N.$$

Oder mit dem Skalarprodukt  $\langle f, g \rangle \equiv \int_{\Omega} f \cdot g \, dx$  ausgedrückt:

$$\langle \omega_k, R \rangle = 0 \quad \text{für alle } k = 0, \dots, N. \quad (3.4)$$

Dies bedeutet, dass das Residuum zu allen Testfunktionen  $\omega_k$  orthogonal sein muss. Spektralmethoden unterscheiden sich hauptsächlich in der Wahl der Ansatzfunktionen von Finite-Element- und Finite-Differenzen-Methoden. Es werden Ansatzfunktionen gewählt, die global glatt sind. Typischerweise sind diese die Tensorprodukte der Eigenfunktionen des singulären Sturm-Liouville-Problems. Im Fall der Finite-Element-Methoden, wird das Gebiet in kleine Elemente geteilt und die Ansatzfunktion approximieren die Funktion über jedes dieser Elemente. Die Ansatzfunktionen sind lokal glatt. Die Ansatzfunktionen bei Finite-Differenzen-Methoden sind ebenfalls lokal [CHQ<sup>+</sup>12].

Wir werden in den Abschnitten 3.1 und 3.2 bekannte Ansatz- und Testfunktionen betrachten, die bei den Spektralmethoden eine wichtige Rolle spielen. In Abschnitt 3.3 stellen wir die Gauß-Lobatto-Quadratur vor und führen die GLL-Punkte und Legendre Interpolation ein. Die Überlegungen in diesem Kapitel stammen aus [QV08] und [BM92].

## 3.1 Wahl der Ansatzfunktionen

Die Wahl der Ansatzfunktionen ist abhängig von dem zu lösenden Problem. In den meisten Fällen sind die Ansatzfunktionen auf dem gesamten Gebiet  $\Omega$  glatte Funktionen. Es gibt viele Möglichkeiten Ansatzfunktionen zu wählen, insbesondere werden

trigonometrische (Fourier) Funktionen sowie Chebyshev- und Legendre-Polynome verwendet. Im nächsten Abschnitt betrachten wir Entwicklungen glatter Funktionen als Reihen, die auf orthogonale Basen zurückzuführen sind, wie die Fourier-Reihe, Chebyshev- und Legendre-Polynome.

### 3.1.1 Fourier-Reihe

Fourier-Reihen sind eine gute Wahl zur Diskretisierung von Funktionen mit periodischen Randbedingungen. Sei  $u(x)$  eine periodische, integrierbare Funktion für  $x \in [0, 2\pi]$ , dann ist die Fourier Reihe dieser Funktion durch

$$u(x) = \sum_{k=-\infty}^{\infty} u_k e^{ikx}$$

gegeben, wobei

$$u_k = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-ikx} dx$$

gilt. Die sich so ergebenden Ansatzfunktionen sind durch

$$\phi_k(x) = e^{ikx}$$

gegeben.

Die Fourier-Reihe einer glatten Funktion konvergiert mit zunehmendem  $k$  schnell, da die Größe des Koeffizienten  $|u_k|$  exponentiell abnimmt. Dieses Verhalten wird als spektrale Konvergenz bezeichnet und tritt nur bei genügend glatten Funktionen auf.

### 3.1.2 Chebyshev-Polynome

Bei nicht-periodischen Problemen werden Ansatz-Funktionen benutzt, die auf orthogonalen Polynomen basieren. Gottlieb und Orszag haben bewiesen, dass die Eigenfunktionen des singulären Sturm-Liouville-Problem, welche auf  $(-1, 1)$  definiert sind, angewandt als Ansatz-Funktionen ebenfalls spektrale Genauigkeit liefern [GO77].

Das Sturm-Liouville-Eigenwertproblem hat die Form

$$(pu')' + (q + \lambda\omega)u = 0,$$

wobei  $p, q, \omega$  reellwertige, stetige Funktionen sind. Das Sturm-Liouville-Problem heißt singulär, wenn  $p$  am Rand verschwindet. Die einzigen Polynome, die als Ei-

genfunktionen eines singulären Sturm-Liouville-Problems auftreten, sind die Jacobi-Polynome.

**Definition 3.1.1.** Die Jacobi-Polynome  $P_n^{(\alpha,\beta)}$  sind für  $x \in [-1, 1]$ ,  $\alpha, \beta > -1$  und  $n \in \mathbb{N}_0$  definiert als

$$P_n^{(\alpha,\beta)}(x) = \frac{\Gamma(\alpha + n + 1)}{n! \Gamma(\alpha + \beta + n + 1)} \sum_{m=0}^n \binom{n}{m} \frac{\Gamma(\alpha + \beta + n + m + 1)}{\Gamma(\alpha + m + 1)} \left(\frac{x-1}{2}\right)^m.$$

Für  $n \in \mathbb{N}$  gilt:  $\Gamma(n) = (n-1)!$ .

**Bemerkung 3.1.1.** Die Jacobi-Polynome erfüllen die Rekursion

$$\begin{aligned} P_0^{(\alpha,\beta)}(x) &= 1, \\ P_1^{(\alpha,\beta)}(x) &= \frac{1}{2}((\alpha - \beta) + (\alpha + \beta + 2)x), \\ a_n^1 P_{n+1}^{(\alpha,\beta)}(x) &= a_n^2 P_n^{(\alpha,\beta)} - a_n^3 P_{n-1}^{(\alpha,\beta)} \end{aligned}$$

mit den Koeffizienten

$$\begin{aligned} a_n^1 &= 2(n+1)(n+\alpha+\beta+1)(2n-\alpha+\beta), \\ a_n^2 &= (2n+\alpha+\beta+1)(\alpha^2+\beta^2) - x \frac{\Gamma(2n+\alpha+\beta+3)}{\Gamma(2n+\alpha+\beta)}, \\ a_n^3 &= 2(n+\alpha)(n+\beta)(2n+\alpha+\beta+2). \end{aligned}$$

**Lemma 3.1.1.** Die Jacobi-Polynome vom Grad höchstens  $n$  mit festem  $\alpha, \beta$  bilden eine orthogonale Basis von

$$\mathcal{P}_n([-1, 1]) = \left\{ \sum_{k=0}^n a_k x^k \mid a_k \in \mathbb{R}, x \in [-1, 1] \right\}$$

bezüglich des gewichteten Skalarproduktes

$$\langle P_m^{(\alpha,\beta)}, P_n^{(\alpha,\beta)} \rangle = \int_{-1}^1 \omega(x) P_m^{(\alpha,\beta)}(x) P_n^{(\alpha,\beta)}(x) dx$$

mit der Gewichtsfunktion  $\omega(x) = (1-x)^\alpha (1+x)^\beta$ . Außerdem gilt

$$\langle P_n^{(\alpha,\beta)}, P_n^{(\alpha,\beta)} \rangle = \frac{2^{\alpha+\beta+1} \Gamma(n+\alpha+1) \Gamma(n+\beta+1)}{(2n+\alpha+\beta+1) n! \Gamma(n+\alpha+\beta+1)}.$$

**Beweis.** Der Beweis ist in [Gau04] nachzulesen.

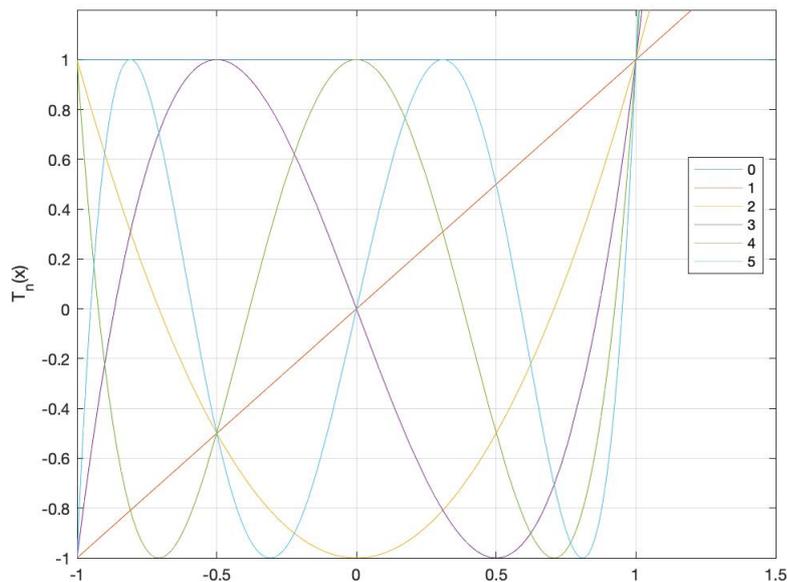


Abbildung 3.1: Chebyshev-Polynome  $T_n$  für  $n = 0, \dots, 5$

Spezialfälle der Jacobi-Polynome sind die Chebyshev- und Legendre-Polynome. Die Legendre-Polynome werden im nächsten Abschnitt ausführlich behandelt.

Für  $\alpha = \beta = -\frac{1}{2}$  und  $\omega = \frac{1}{\sqrt{1-x^2}}$  erhalten wir die Chebyshev-Polynome

$$T_n(x) = P_n^{(-\frac{1}{2}, -\frac{1}{2})}(x).$$

Die Chebyshev-Polynome sind für  $|x| \leq 1$  definiert als

$$T_n(x) = \cos(n \cdot \arccos(x)), \quad n = 0, 1, 2, \dots$$

Die ersten Polynome sind

$$T_0(x) = 1,$$

$$T_1(x) = x,$$

$$T_2(x) = 2x^2 - 1,$$

$$T_3(x) = 4x^3 - 3x,$$

$$T_4(x) = 8x^4 - 8x^2 + 1,$$

$$T_5(x) = 16x^5 - 20x^3 + 5x.$$

Die rekursive Formel lautet

$$T_{n+1} = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1.$$

Die Funktion  $u$  aus (3.1) wird approximiert durch

$$\tilde{u}(x) = \sum_{n=0}^N u_n T_n(x),$$

wobei  $u_n$  die Chebyshev-Koeffizienten sind. Diese sind gegeben durch

$$u_n = \frac{2}{\pi c_n} \int_{-1}^1 u(x) T_n(x) \frac{1}{\sqrt{1-x^2}} dx,$$

wobei

$$c_n = \begin{cases} 2 & \text{für } n = 0, \\ 1 & \text{für } n \geq 1. \end{cases}$$

Zur späteren Berechnung und Approximation müssen die Ansatzfunktionen an diskreten Punkten ausgewertet werden, eine häufige Punkteverteilung speziell für die Chebyshev-Polynome sind die Gauß-Lobatto-Punkte

$$x_j = \cos\left(\frac{\pi j}{N}\right), \quad j = 0, \dots, N.$$

Die Chebyshev-Polynome besitzen folgende wichtige Eigenschaften:

- Für die Randbedingungen gilt:  $T_n(1) = 1$  und  $T_n(-1) = (-1)^n$ .
- Die Werte der Chebyshev-Polynome an den Gauß-Lobatto Knoten sind

$$T_n(x_j) = \cos\left(\frac{n\pi j}{N}\right), \quad j, n = 0, \dots, N.$$

### 3.1.3 Legendre-Polynome

Wie zuvor erwähnt sind die Legendre-Polynome ein Spezialfall der Jacobi-Polynome. Für  $\alpha = \beta = 0$  und  $\omega = 1$  erhalten wir die Legendre-Polynome

$$L_n(x) = P_n^{(0,0)}(x).$$

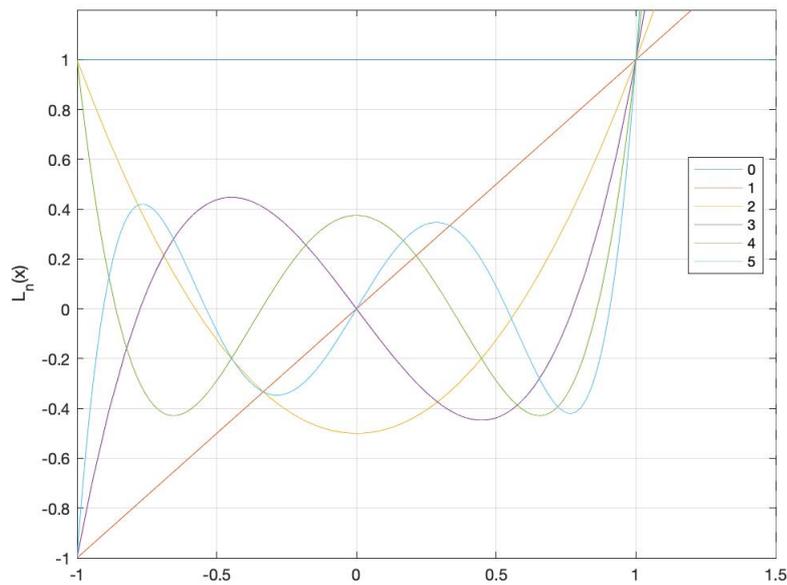


Abbildung 3.2: Legendre-Polynome  $L_n$  für  $n = 0, \dots, 5$

Unter Verwendung des  $L^2$ - Skalarproduktes und Norm

$$\langle u, v \rangle = \int_{-1}^1 u(x)v(x) dx, \quad \|u\|_0 = \left( \int_{-1}^1 u^2(x) dx \right)^{\frac{1}{2}}$$

haben wir

$$\langle L_n, L_m \rangle = \frac{2}{2n+1} \delta_{nm} . \tag{3.5}$$

Die ersten Legendre-Polynome lauten:

$$\begin{aligned} L_0(x) &= 1, \\ L_1(x) &= x, \\ L_2(x) &= \frac{1}{2}(3x^2 - 1), \\ L_3(x) &= \frac{1}{2}(5x^3 - 3x), \\ L_4(x) &= \frac{1}{2}(35x^4 - 30x^2 + 3), \\ L_5(x) &= \frac{1}{2}(63x^5 - 70x^3 + 15x). \end{aligned}$$

Die rekursive Formel ist gegeben durch

$$L_{n+1}(x) = \frac{2n+1}{n+1}xL_n(x) - \frac{n}{n+1}L_{n-1}(x), \quad n \geq 1.$$

Die Funktion  $u$  aus (3.1) wird approximiert durch

$$\tilde{u}(x) = \sum_{n=0}^N u_n L_n(x),$$

wobei  $u_n$  die Legendre-Koeffizienten sind. Diese sind gegeben durch

$$u_n = \left(n + \frac{1}{2}\right) \int_{-1}^1 u(x) L_n(x) dx .$$

Die Erweiterung auf höher-dimensionale Probleme für viereckige Spektralmethoden ist unkompliziert, denn ein Rechteck kann als kartesisches Produkt zweier Intervalle angesehen werden. Deshalb sind die Ansatzfunktionen durch das Produkt der eindimensionalen orthogonalen Funktionen gegeben. Dieses Vorgehen lässt sich jedoch nicht auf dreieckige Elemente anwenden, daher werden im Abschnitt 4.1 die Dubiner-Polynome vorgestellt, die eine orthogonal Basis auf dem Dreieck bilden und auch auf den Jacobi-Polynomen basieren.

## 3.2 Wahl der Testfunktionen

Die Auswahl der Testfunktionen ist entscheidend für die Spektralmethode, da abhängig von der Auswahl der Testfunktionen verschiedene Diskretisierungsmethoden unterschieden werden. Drei der bekanntesten und am häufigsten verwendeten Testfunktionen erzeugen die Diskretisierungsmethoden Galerkin-Methode, Tau-Methode und Kollokationsmethode.

Bei der Galerkin-Methode werden für Ansatzfunktionen und Testfunktionen die gleichen Basen verwendet, dies bedeutet  $\phi_k = \omega_k$  und die Ansatzfunktionen erfüllen die gegebenen Randbedingungen. Die Grundidee der Tau-Methode ist dieselbe wie die der Galerkin-Methode. Bei der Tau-Methode sind die Testfunktionen und die meisten Ansatzfunktionen gleich, jedoch erfüllen die Ansatzfunktionen nicht zwingend die Randbedingungen, diese werden durch Einsatz zusätzlicher Gleichungen erzwungen.

Im Speziellen wird in dieser Arbeit die Methode der Kollokation verwendet, welche im folgenden Abschnitt gesondert betrachtet wird.

### 3.2.1 Kollokation

Für die Kollokationsmethode, oder auch pseudospektrale Methode genannt, werden im ersten Schritt die Kollokationspunkte  $x_k$ ,  $k = 0, \dots, N$ , definiert. Bei vier-eckigen Elementen können dies zum Beispiel die Gauß-Lobatto-Legendre (GLL)-Punkte sein. Da die GLL-Punkte nicht auf dreieckigen Elementen angewendet werden können, werden wir die später in Abschnitt 4.2 eingeführten Fekete-Punkte im Zusammenhang mit Dreiecken verwenden. Die Testfunktionen  $\omega_k$  sind hierbei Dirac-Delta-Funktionen definiert unter Verwendung der zuvor gewählten Kollokationspunkte:

$$\omega_k = \delta(x - x_k),$$

wobei  $(x_k)$  die Kollokationspunkte sind.

Im Fall  $\phi_k = \omega_k$  gilt zur Minimierung der Residuumsfunktion aus (3.4) für alle  $k = 0, \dots, N$ :

$$\begin{aligned} \langle \omega_k, R \rangle &= 0 \\ \Leftrightarrow \langle \delta(x - x_k), R \rangle &= 0 \\ \Leftrightarrow R(x_k) &= 0 \\ \Leftrightarrow \mathcal{L}\tilde{u}(x_k) &= f(x_k) \\ \Leftrightarrow \sum_{n=0}^N \mathcal{L}\phi_n(x_k)u_n &= f(x_k). \end{aligned}$$

Aus den Randbedingung (3.2) folgt,

$$\begin{aligned} \mathcal{B}\tilde{u} &= 0 \\ \Rightarrow \mathcal{B} \sum_{n=0}^N u_n \phi_n(x) &= 0 \\ \sum_{n=0}^N u_n \mathcal{B}\phi_n(x) &= 0 \\ \Rightarrow \langle \omega_k, \sum_{n=0}^N u_n \mathcal{B}\phi_n(x) \rangle &= 0 \\ \langle \delta(x - x_k), \sum_{n=0}^N u_n \mathcal{B}\phi_n(x) \rangle &= 0 \\ \Rightarrow \sum_{n=0}^N u_n \mathcal{B}\phi_n(x_k) &= 0 \end{aligned}$$

für  $k$  mit  $x_k \in \partial\Omega$ .

Damit ergeben sich  $N + 1$  Bedingungen für  $N + 1$  Unbekannte:

$$\sum_{n=0}^N \mathcal{L}\phi_n(x_k)u_n = f(x_k), \quad x_k \in \Omega \setminus \partial\Omega \quad (3.6)$$

$$\sum_{n=0}^N \mathcal{B}\phi_n(x_k)u_n = 0, \quad x_k \in \partial\Omega. \quad (3.7)$$

### 3.3 Spektrale Approximation

Zur Konstruktion von Quadraturformeln mit hoher Genauigkeit können die Nullstellen und die Extrema der Legendre-Polynome oder anderer Familien orthogonaler Polynome verwendet werden. Als Basis können die Gauß und Gauß-Lobatto-Quadratur Formeln verwendet werden [BM97]. Spektralmethoden verwenden häufig für viereckige Elemente einen Knotenansatz, der auf dem Tensorprodukt eindimensionaler Gauß-Lobatto-Legendre-Punkte (GLL) basiert. Hierbei können wir die Tensorprodukte von Gauß-Lobatto-Legendre-Punkten als Interpolations- und Integrationsknoten verwenden. Die Vorteile sind die hervorragenden Interpolations- und Integrationsseigenschaften. Wir werden auf unseren Versuchsgebieten sowohl auf Dreiecken, als auch auf Vierecken rechnen. In Kapitel 4 werden wir detaillierter auf die dreiecksbasierte Spektralmethode eingehen. Auf Vierecken verwenden wir die GLL-Punkte und Legendre-Polynome.

#### 3.3.1 Gauß-Lobatto-Quadratur und Interpolation

Zu den häufigsten Arten der Gauß-Quadratur gehören Gauß-, Gauß-Radau- und Gauß-Lobatto-Quadratur. Der Unterschied zwischen diesen drei Arten der Quadratur beruht auf der Wahl der Quadraturpunkte. Bei der Gauß-Quadratur werden ausschließlich Quadraturpunkte im Innern des Intervalls verwendet. Hierzu sei erwähnt, dass zusätzliche Variationen existieren, welche auch die Randpunkte einschließen, diese sind aber unter Umständen weniger genau. Wird ein einziger Randpunkt, meistens  $x = -1$  (auf dem Referenzintervall  $[-1, 1]$ ), verwendet, sprechen wir von Gauß-Radau-Quadratur. Werden beide Randpunkte,  $x = \pm 1$ , verwendet, heißt das Verfahren Gauß-Lobatto-Quadratur.

Die Quadraturformeln für die Integration von Polynomen, die an den Legendre-Quadraturpunkten gegeben sind, können direkt aus den unten abgeleiteten Formeln durch Setzen von  $\alpha = 0$  erhalten werden [HG07].

Passende Knoten  $x_j$  und Gewichten  $\omega_j$  sollen derart bestimmt werden, sodass

$$\int_{-1}^1 p(x)\omega(x) dx = \sum_{j=0}^N p(x_j)\omega_j \quad (3.8)$$

gilt, wobei  $\mathcal{P}_N$  den Raum aller Polynome von Grad kleiner gleich  $N$  bezeichnet. Die Knoten sind  $-1 = x_0, x_1, \dots, x_{N-1}, x_N = 1$ , wobei die inneren Punkte  $x_1, \dots, x_{N-1}$  die Nullstellen der ersten Ableitung des Polynoms

$$P_N^{(\alpha)}(x) = \frac{\Gamma(\alpha + 1)\Gamma(n + 2\alpha + 1)}{\Gamma(2\alpha + 1)\Gamma(n + \alpha + 1)} P_N^{(\alpha, \alpha)}(x) \quad (3.9)$$

sind, wobei  $P_N^{(\alpha, \alpha)}(x)$  das zuvor definierte Jacobi-Polynom und  $\Gamma(n)$  die Gammafunktion darstellt.

Die Gewichte  $\omega_j$  sind gegeben durch

$$\omega_j = \begin{cases} (\alpha + 1) \prod_{N,j}^{(\alpha)} & j = 0, N \\ \prod_{N,j}^{(\alpha)} & j \in \{1, \dots, N - 1\} \end{cases} ,$$

wobei

$$\prod_{N,j}^{(\alpha)} = 2^{2\alpha+1} \frac{\Gamma^2(\alpha + 1)\Gamma(N + 2\alpha + 1)}{NN!(N + 2\alpha + 1)\Gamma^2(2\alpha + 1)} \left[ P_N^{(\alpha)}(x_j) \right]^{-2} .$$

Die Gauß-Lobatto-Formel ist für alle Polynome  $p(x) \in \mathcal{P}_{2N-1}$  genau. Die Quadraturknoten sind nicht explizit gegeben, sie werden durch Auffinden der Nullstellen des Polynoms in (3.9) berechnet.

### 3.3.2 Gauß-Lobatto-Legendre-Punkte und Legendre-Interpolation

Die Gauß-Lobatto-Legendre-Punkte  $x_j$  werden definiert durch die Nullstellen des Polynoms

$$q(x) = (1 - x^2) \frac{dL_N}{dx}(x),$$

wobei  $L_N(x)$  das Legendre-Polynom von Grad  $N$  ist.

Die Gewichte  $\omega_j$  sind gegeben durch

$$\omega_j = \frac{2}{N(N+1)} [L_N(x_j)]^{-2}, \quad j = 0, \dots, N. \quad (3.10)$$

Die diskrete Approximation zum Skalarprodukt von  $L^2(-1, 1)$  und die dazugehörige Norm lauten

$$\begin{aligned} \langle u, v \rangle_N &= \frac{2}{N(N+1)} \sum_{j=0}^N \frac{1}{L_N^2(x_j)} u(x_j) v(x_j), \\ \|v\|_N &= \left( \frac{2}{N(N+1)} \sum_{j=0}^N \frac{v^2(x_j)}{L_N^2(x_j)} \right)^{\frac{1}{2}}, \end{aligned}$$

wobei  $\{x_j\}$  die GLL-Punkte sind.

Wegen (3.5) und (3.8) gelten

$$\|L_k\|_N = \|L_k\|_0, \quad k = 0, \dots, N-1, \quad (3.11)$$

$$\|L_N\|_N = \left( \frac{2}{N} \right)^{\frac{1}{2}} = \left( 2 + \frac{1}{N} \right)^{\frac{1}{2}} \|L_N\|_0. \quad (3.12)$$

Es folgt

$$\|u_N\|_0 \leq \|u_N\|_N \leq \left( 2 + \frac{1}{N} \right)^{\frac{1}{2}} \|u_N\|_0 \quad \forall u_N \in \mathcal{P}_N. \quad (3.13)$$

Nun führen wir für jede stetige Funktion  $u$  die Interpolierende  $\mathcal{I}_N u \in \mathcal{P}_N$  ein, welche mit  $u$  an den  $N+1$  GLL-Punkten  $\{x_j\}_{j=0, \dots, N}$  übereinstimmt.

Es ist

$$\mathcal{I}_N u(x) = \sum_{k=0}^N u_k^* L_k(x) \quad (3.14)$$

mit

$$u_k^* = \frac{2k+1}{N(N+1)} \sum_{j=0}^N u(x_j) L_k(x_j) \frac{1}{L_N^2(x_j)}, \quad k = 0, \dots, N-1 \quad (3.15)$$

$$u_N^* = \frac{1}{N+1} \sum_{j=0}^N u(x_j) \frac{1}{L_N(x_j)}. \quad (3.16)$$

(3.15) und (3.16) sind bekannt als diskrete Legendre-Transformation. Für jede stetige Funktion  $u$  können wir die Legendre pseudospektrale Ableitung  $\partial_n u$  definieren:

$$\partial_n u = \frac{d}{dx}(\mathcal{I}_N u),$$

wobei  $\mathcal{I}_N u$  wie in (3.14) ist.

$\mathcal{I}_N u(x)$  kann dargestellt werden als

$$\mathcal{I}_N u(x) = \sum_{j=0}^N u(x_j) \psi_j(x)$$

mit der Lagrange Funktion

$$\psi_j(x) = -\frac{1}{N(N+1)} \frac{(1-x^2)L'_N(x)}{(x-x_j)L_N(x)}, \quad j = 0, \dots, N \ .$$

$\psi_j(x)$  ist ein Polynom von Grad  $N$  und erfüllt  $\psi_j(x_k) = \delta_{jk}$ .

Die Legendre pseudospektrale Differentiationsmatrix  $D_N$  ist gegeben durch

$$(D_N)_{ij} = \begin{cases} \frac{1}{x_i - x_j} \frac{L_N(x_i)}{L_N(x_j)} & i \neq j, \\ 0 & 1 \leq i = j \leq N - 1, \\ \frac{N(N+1)}{4} & i = j = 0, \\ -\frac{N(N+1)}{4} & i = j = N. \end{cases}$$

Der Interpolationsfehler kann angenähert werden durch

$$\|u - \mathcal{I}_N u\|_{L_2} \leq CN^{-s} \|u\|_{H^s} \ ,$$

wobei  $C$  eine positive Konstante und  $H^s$  für  $s \geq 1$ , dabei ist  $H^s$  ein Sobolev-Raum.

### 3.3.3 Legendre-Polynome in 2D

In diesem Abschnitt werden wir die vorherigen Definitionen und Ergebnisse auf das zweidimensionale Gebiet  $\Omega = (-1, 1)^2$  erweitern. Wir bezeichnen für jedes  $N \in \mathbb{N}$  den Raum der Polynome von Grad kleiner gleich  $N$  als  $\mathcal{Q}_N$  in Abhängigkeit von jeder Variable  $x_i, i = 1, 2$ . Außerdem bezeichnen wir mit  $\mathcal{Q}_N^0$  den Unterraum derjenigen Polynome, die auf dem Rand  $\partial\Omega$  verschwinden.

Wir führen den Raum

$$L^2(\Omega) = \left\{ u : \Omega \rightarrow \mathbb{R} \text{ messbar} \mid \int_{\Omega} u^2(x) dx < \infty \right\}$$

ein, der mit folgendem Skalarprodukt und Norm ausgestattet ist

$$\langle u, v \rangle = \int_{\Omega} u(x)v(x) dx, \quad \|u\|_0 = \left( \int_{\Omega} u^2(x) dx \right)^{\frac{1}{2}}.$$

Zudem definieren wir die Gewichtsfunktionen

$$\omega(x) = (1 - x_1^2)^{-\frac{1}{2}}(1 - x_2^2)^{-\frac{1}{2}}$$

und

$$L_{\omega}^2(\Omega) = \left\{ u : \Omega \rightarrow \mathbb{R} \text{ messbar} \mid \int_{\Omega} u^2(x)\omega(x) dx < \infty \right\}$$

als gewichteten Lebesgue-Raum mit

$$\langle u, v \rangle_{\omega} = \int_{\Omega} u(x)v(x)\omega(x) dx, \quad \|u\|_{0,\omega} = \left( \int_{\Omega} u^2(x)\omega(x) dx \right)^{\frac{1}{2}}.$$

Wir werden den Fall für die Legendre-Polynome betrachten. Für den Chebyshev Fall verweisen wir auf [QV08].

Die zweidimensionale Legendre-Polynome sind definiert als Tensorprodukt der eindimensionalen Legendre-Polynome

$$L_{km}(x, y) = L_k(x)L_m(y), \quad k, m \in \mathbb{N}.$$

Aus (3.5) folgt

$$\langle L_{km}, L_{kn} \rangle = \frac{2}{2k+1} \frac{2}{2m+1} \delta_{kl} \delta_{nm}. \quad (3.17)$$

### 3.3.4 2D Gauß-Lobatto-Legendre-Punkte

Die 2D GLL-Punkte und Gewichte sind gegeben durch

$$x_{ij} = (x_i, x_j) \in \mathbb{R}^2, \quad \omega_{ij} = \omega_i \omega_j, \quad i, j = 0, \dots, N, \quad N \geq 1,$$

wobei  $\{x_k\}_{k=0,\dots,N}$  die Nullstellen von  $(1-t^2)L'_N(t)$  sind und  $\{\omega_k\}_{k=0,\dots,N}$  wie in (3.10). Das dazugehörige diskrete Skalarprodukt lautet

$$\langle u, v \rangle_N = \sum_{i,j=0}^N u(x_{ij})v(x_{ij})\omega_{ij}, \quad u, v \in C^0(\bar{\Omega}).$$

Wir können aus (3.11)-(3.13) folgern, dass folgende Ungleichung für alle  $u_N \in \mathcal{Q}_N$  gilt:

$$\|u_N\|_0 \leq \|u_N\|_N \leq \left(2 + \frac{1}{N}\right) \|u_N\|_0.$$

Für jede stetige Funktion  $u$  gilt für den Interpolanten  $\mathcal{I}_N u \in \mathcal{Q}_N$  auf den Punkten  $\{x_{ij}\}$ :

$$\mathcal{I}_N u(x) = \sum_{i,j=0}^N u_{ij}^* L_{ij}(x)$$

mit

$$u_{ij}^* = \frac{1}{\|L_{ij}\|_N^2} \langle u, L_{ij} \rangle_N,$$

wobei  $\|v\|_N^2 = \langle v, v \rangle_N$ . Der Interpolationsfehler ist dann gegeben durch

$$\|u - \mathcal{I}_N u\|_{L^2} \leq CN^{-s} \|u\|_{H^s},$$

falls  $u \in H^s(\Omega)$  für  $s \geq 2$  und für  $C$  ist eine positive Konstante.

# Kapitel 4

## Dreiecksbasierte Spektralmethode

Die Wahl der Kollokationspunkte auf dem Referenzgebiet beeinflusst die Effizienz und Effektivität der Spektralmethode. Im Fall klassischer vierecksbasierter Spektralmethoden können wir die Tensorprodukte von Gauß-Lobatto-Legendre-Punkten als Interpolations- und Integrationsknoten verwenden.

Dies hat hervorragende Interpolations- und Integrationseigenschaften zum Vorteil. Für Dreiecksgebiete, also nicht Tensorgebieten, ist dieser Ansatz nicht mehr möglich. Wir wollen dennoch eine ähnliche Effizienzsteigerung für Dreiecksgebiete ermöglichen. Dubiner präsentierte eine orthogonale Polynomialbasis, welche auf die konkrete Anwendung im Spektralverfahren ausgerichtet war. Diese Basis ist ein Spezialfall, der bereits von Prorior und Koornwinder früher konstruiert worden war. Wir verwenden die Dubiner-Basis als orthogonale Basis von Polynomen. Taylor, Wingate und Vincent konstruierten sogenannte Fekete-Punkte im Dreieck. Diese weisen fast optimale Interpolationseigenschaften auf, weshalb wir sie als Kollokationspunkte verwenden [TWV00].

In Abschnitt 4.1 werden die Dubiner-Polynome und in Abschnitt 4.2 die Fekete-Punkte vorgestellt. In Abschnitt 4.3 und 4.4 betrachten wir die spektralen Differentiationsmatrizen unter Anwendung der Dubiner-Polynome und Fekete-Punkten. Im letzten Abschnitt präsentieren wir eine Herangehensweise, wenn benachbarte Dreiecke sich das gleiche Interface teilen.

### 4.1 Dubiner-Polynome

In der vierecksbasierten Spektralmethode werden Tensorprodukte von eindimensionalen Basisfunktionen eingesetzt, um gute Orthogonalitätseigenschaften und eine geringe Matrix-Kondition zu erhalten. Dies ist jedoch auf Dreiecksgebieten nicht analog umsetzbar. Wie wir im Weiteren betrachten werden, bieten Dubiner-Polynome im

Fall von orthogonalen Polynomialbasen auf Dreiecken die gewünschten Eigenschaften. Die Dubiner-Polynome bzw. Proriol-Koornwinder-Dubiner-Polynome (PKD-Polynome) lassen sich mit Hilfe eines zusammenfallenden Koordinatensystems (collapsed coordinate system), durch das ein Dreieck als Quadrat mit identischen Ecken aufgefasst wird, aus den eindimensionalen Jacobi-Polynomen  $P_n^{\alpha,\beta}$  konstruieren. Sie bilden eine hierarchische Basis bestehend aus Polynomen in den Dreieckskoordinaten, die bezüglich des  $L^2$ -Skalarproduktes orthogonal sind. Zur Konstruktion betrachten wir das Standarddreieck

$$T_0 = \{(x, y) \in \mathbb{R}^2 \mid -1 \leq x, y; x + y \leq 0\}$$

und das Standardquadrat

$$Q_0 = \{(x', y') \in \mathbb{R}^2 \mid -1 \leq x', y' \leq 1\}.$$

Wir definieren eine Abbildung  $\phi$  zwischen dem Standarddreieck  $T_0$  und dem Standardquadrat  $Q_0$ :

$$\phi : Q_0 \rightarrow T_0 : \begin{pmatrix} x' \\ y' \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(1+x')(1-y') - 1 \\ y' \end{pmatrix}$$

mit der zugehörigen inversen Abbildung:

$$\phi^{-1} : T_0 \rightarrow Q_0 : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 2\frac{1+x}{1-y} - 1 \\ y \end{pmatrix}.$$

Die Zuordnung bildet  $T_0 \setminus \{(-1, 1)\}$  bijektiv auf  $Q_0 \setminus \{[-1, 1] \times \{1\}\}$  ab.

Die inverse Abbildung bildet das durch  $T'_0 = T_0 \setminus \{(-1, 1)\}$  gegebene nach oben offene Dreieck auf das nach oben offene Quadrat  $Q'_0 = Q_0 \setminus \{(x', 1) \mid -1 \leq x' \leq 1\}$  ab.

Die Funktion  $\phi$  verursacht ein Kollabieren der oberen Kante von  $Q_0$  auf den singulären Eckpunkt  $(-1, 1)$  des Dreiecks  $T_0$ . Das Koordinatensystem  $(x, y)$  wird auch zusammenfallendes Koordinatensystem (*collapsed coordinate system*) genannt.

Seien  $p, q : [-1, 1] \rightarrow \mathbb{R}$  Polynome in einer Variable, wobei  $p$  vom Grad  $m$  und  $q$

vom Grad  $n$  ist. Dann ist

$$g_{mn} : T'_0 \rightarrow \mathbb{R},$$

$$g_{mn}(x, y) = p \left( 2 \frac{1+x}{1-y} - 1 \right) \left( \frac{1-y}{2} \right)^m q(y)$$

zu einem Polynom in den Variablen  $x$  und  $y$  auf ganz  $T_0$  fortsetzbar.  
 $g_{mn}$  lässt sich in  $T'_0$  als Produkt zweier eindimensionaler Polynome

$$g_{mn}(x, y) = g_m^1(x'(x, y)) \cdot g_n^2(y'(x, y))$$

schreiben mit

$$g_m^1(x') = p(x')$$

und

$$g_n^2(y') = \left( \frac{1-y'}{2} \right)^n q(y').$$

Durch die Wahl der Jacobi-Polynome  $m$ -ten und  $n$ -ten Grades,

$$p(x') = P_m^{(0,0)}(x')$$

und

$$q(y') = P_n^{(2m+1,0)}(y')$$

erhalten wir die Prorjol-Koornwinder-Dubiner-Polynome oder kurz PKD-Polynome bezüglich den Gewichtsfunktionen

$$\omega(x) = 1$$

bzw.

$$\omega(x) = (1-x)^{2m+1}.$$

**Definition 4.1.1.** Die PKD-Polynome sind gegeben durch

$$g_{mn}(x, y) = P_m^{(0,0)} \left( 2 \frac{1+x}{1-y} - 1 \right) \left( \frac{1-y}{2} \right)^m P_n^{(2m+1,0)}(y), \quad m, n, \in \mathbb{N}_0,$$

wobei  $P_k^{(\alpha,\beta)}(\zeta)$  das  $k$ -te Jacobi-Polynom bezüglich der Gewichtsfunktion  $(1-\zeta)^\alpha(1+\zeta)^\beta$  bezeichnet.

Die Indizes  $m$  und  $n$  geben den maximalen Polynomgrad in  $x$  und  $y$  an.  $g_{mn}$  ist ein

Polynom in  $x, y$  von höchst Grad  $m + n$ .

**Lemma 4.1.1.** *Die Polynome  $g_{mn}$  sind bezüglich des  $L^2$ -Skalarproduktes auf dem Dreieck  $T_0$  orthogonal, das  $L^2$ -Skalarprodukt ist in diesem Fall gegeben durch*

$$(g, \tilde{g}) = \int_{T_0} g(x, y) \tilde{g}(x, y) dx dy, \quad g, \tilde{g} \in L^2(T_0),$$

und es gilt

$$(g_{mn}, g_{mn}) = \frac{2}{(2m+1)(n+m+1)}. \quad (4.1)$$

Weiterhin können wir für ein festes  $N \in \mathbb{N}_0$  eine orthogonale Basis  $\mathcal{B}_N$  von

$$\mathcal{P}_N := \mathcal{P}_N(T_0) = \text{span}\{x^i y^j \mid 0 \leq i, j; i + j \leq N\}$$

bilden.  $\mathcal{P}_N$  hat die Dimension  $d := d_N = \frac{1}{2}(N+1)(N+2)$  und wird von

$$\mathcal{B}_N = \{g_{mn} \mid 0 \leq m, n; m + n \leq N\} \subset \mathcal{P}_N$$

aufgespannt.

**Beweis.** Die Funktion  $\phi$  hat die Jacobi-Determinante  $\det(D_\phi(x', y')) = \frac{1-y'}{2}$ .

Es folgt, dass

$$\begin{aligned} (g_{mn}, g_{kl}) &= \int_{T_0} g_{mn}(x, y) g_{kl}(x, y) dx dy \\ &= \int_{Q_0} P_m^{(0,0)}(x') P_k^{(0,0)}(x') \left(\frac{1-y'}{2}\right)^{m+k+1} P_n^{(2m+1,0)}(y') P_l^{(2k+1,0)}(y') dx' dy' \\ &= \int_{-1}^1 P_m^{(0,0)}(x') P_k^{(0,0)}(x') dx' \cdot \int_{-1}^1 \left(\frac{1-y'}{2}\right)^{m+k+1} P_n^{(2m+1,0)}(y') P_l^{(2k+1,0)}(y') dy'. \end{aligned}$$

Aufgrund der Orthogonalität der Legendre-Polynome für  $m \neq k$  und der Jacobi-Polynome für  $n \neq l$  folgt:  $(g_{mn}, g_{kl}) = 0$ .

Für  $m = k$  oder  $n = l$  erhalten wir

$$(g_{mn}, g_{mn}) = \int_{-1}^1 (P_m^{(0,0)}(x'))^2 dx' \cdot \int_{-1}^1 \left(\frac{1-y'}{2}\right)^{2m+1} (P_n^{(2m+1,0)}(y'))^2 dy'.$$

Aus

$$\begin{aligned} & \int_{-1}^1 (1-\zeta)^\alpha (1+\zeta)^\beta P_m^{(\alpha,\beta)}(\zeta) P_n^{(\alpha,\beta)}(\zeta) d\zeta \\ &= \frac{2^{\alpha+\beta+1}}{2n+\alpha+\beta+1} \cdot \frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{\Gamma(n+\alpha+\beta+1)n!} \delta_{nm} \end{aligned}$$

folgt

$$\int_{-1}^1 P_m^{(0,0)}(x) P_m^{(0,0)}(x) dx = \frac{2}{(2m+1)} \cdot \frac{\Gamma(n+1)\Gamma(n+1)}{\Gamma(n+1)n!} \stackrel{\Gamma(n+1)=n!}{=} \frac{2}{2m+1}$$

und wir erhalten

$$\begin{aligned} (g_{mn}, g_{mn}) &= \frac{2}{(2m+1)} \cdot \frac{1}{2^{2m+1}} \cdot \frac{2^{2m+1}}{2n+2m+2} \cdot \frac{\Gamma(n+2m+2)\Gamma(n+1)}{\Gamma(n+2m+2)n!} \\ &= \frac{2}{(2m+1)(n+m+1)}. \end{aligned}$$

Somit haben wir den ersten Teil des Lemmas gezeigt. Wir zeigen nun, dass  $\mathcal{B}_N$  eine orthogonale Basis von  $\mathcal{P}_N(T_0)$  ist.

Wir betrachten die Linearkombination

$$\sum_{m,n} \alpha_{mn} g_{mn} = 0.$$

Für beliebige  $(k, l) \in \mathbb{N}_0^2$  ist

$$0 = \left( \sum_{m,n} \alpha_{mn} g_{mn}, g_{kl} \right)_{L^2(T_0)} = \sum_{m,n} \alpha_{mn} (g_{mn}, g_{kl})_{L^2(T_0)} = \alpha_{kl} \|g_{kl}\|_{L^2(T_0)}.$$

Daraus folgt, dass  $\alpha_{kl} = 0$ . Somit ist die lineare Unabhängigkeit gezeigt. Bleibt noch zu zeigen, dass  $g_{mn} \in \mathcal{P}_N(T_0)$  für alle  $m, n \in \mathbb{N}_0$  und  $m+n \leq N$ .

Wir schreiben

$$P_m^{(0,0)}(x) = \sum_{l=0}^m \alpha_l x^l$$

und haben

$$\begin{aligned}
 g_{mn}(x, y) &= \sum_{l=0}^m \alpha_l \left( 2 \frac{1+x}{1-y} - 1 \right)^l \left( \frac{1-y}{2} \right)^m P_n^{(2m+1,0)}(y) \\
 &= \sum_{l=0}^m \alpha_l \left[ \sum_{\tau=0}^l 2^\tau (-1)^{l-\tau} \binom{\tau}{l} \left( \frac{1+x}{1-y} \right)^\tau \right] \left( \frac{1-y}{2} \right)^m P_n^{(2m+1,0)}(y) \\
 &= \sum_{l=0}^m \sum_{\tau=0}^l \alpha_l (-1)^{l-\tau} \binom{\tau}{l} (1+x)^\tau \left( \frac{1-y}{2} \right)^{m-\tau} P_n^{(2m+1,0)}(y) \quad .
 \end{aligned}$$

Das Jacobi-Polynom  $P_n^{(2m+1,0)}$  hat den höchsten Grad  $n$ , daraus folgt dass  $g_{mn}$  den höchsten Grad  $\tau + (m - \tau) + n = m + n \leq N$  hat. Also ist  $g_{mn} \in \mathcal{P}_N(T_0)$ .

Da  $|\mathcal{B}_N| = \dim \mathcal{P}_N(T_0) = \frac{1}{2}(N+1)(N+2) = d$ , folgt, dass  $\mathcal{B}_N$  ist eine orthogonale Basis von  $\mathcal{P}_N(T_0)$ . □

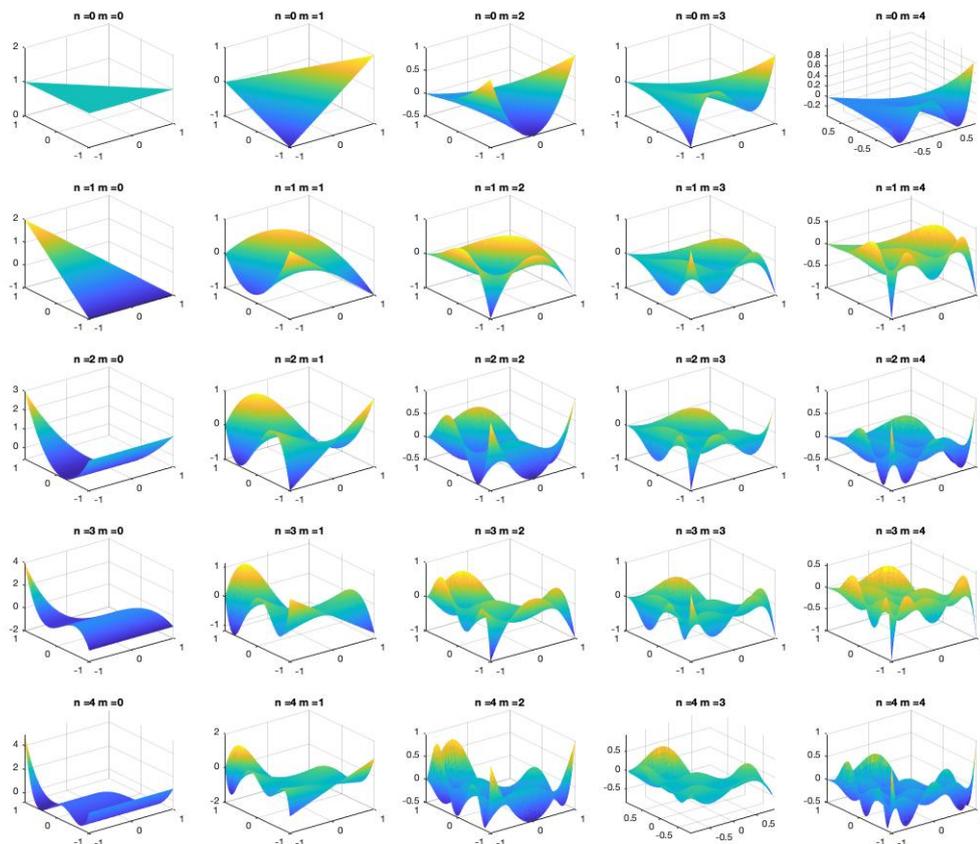


Abbildung 4.1: Dubiner-Polynome auf dem Referenzdreieck

Für eine einfachere Indizierung der Elemente von  $\mathcal{B}_N$  führen wir für ein festes  $N \in \mathbb{N}_0$  folgende Bijektion ein:

$$\begin{aligned} \iota : \{(m, n) | 0 \leq m, n \leq N\} &\rightarrow \{1 \leq k \leq d\} \\ (m, n) &\mapsto k = \frac{1}{2}(m + n + 1)(m + n + 2) - n. \end{aligned}$$

Mit  $g_k(x, y) = g_{mn}(x, y)$ , wobei  $k = \iota(m, n)$ , erhalten wir die Dubiner-Basis vom Grad  $\leq N$ :

$$\mathcal{B}_N = \{g_k | 0 \leq k \leq d\} \subseteq \mathcal{P}_N(T_0). \quad (4.2)$$

Durch die Indizierung der Dubiner-Polynome haben wir den Vorteil, dass die Basisfunktionen nach dem Polynomgrad sortiert sind, also

$$\deg g_k \leq \deg g_m \quad \text{für } k < m.$$

### 4.1.1 Ableitungen der Dubiner-Polynomen

Mit Hilfe der Differentiationsregel für die Jacobi-Polynome

$$\frac{d^k}{dz^k} P_n^{(\alpha, \beta)}(z) = \frac{\Gamma(\alpha + \beta + n + 1 + k)}{2^k \Gamma(\alpha + \beta + n + 1)} P_{n-k}^{(\alpha+k, \beta+k)}(z)$$

lässt sich die Ableitungen der Dubiner-Polynome bestimmen.

Da wir nur die erste Ableitung benötigen, folgt aus der obigen Gleichung für  $\alpha \in \mathbb{N}_0$  und  $\beta = 0$ :

$$\frac{d}{dz} P_n^{(\alpha, 0)}(z) \stackrel{\Gamma(n) = (n-1)!}{=} \frac{(\alpha + n + 1 + 1 - 1)!}{2(\alpha + n + 1 - 1)!} P_{n-1}^{(\alpha+1, 1)}(z) = \frac{\alpha + n + 1}{2} P_{n-1}^{(\alpha+1, 1)}(z).$$

Daraus folgt für die partielle Ableitung nach  $x$  für  $m \neq 0$

$$\frac{\partial}{\partial x} P_m^{(0, 0)} \left( 2 \frac{1+x}{1-y} - 1 \right) = \frac{m+1}{2} P_{m-1}^{(1, 1)} \left( 2 \frac{1+x}{1-y} - 1 \right) \frac{2}{1-y}.$$

Wir setzen

$$P_{-1}^{(\alpha, \beta)}(z) = 0.$$

Wir erhalten die erste Ableitung der Dubiner-Polynome in  $x$ -Richtung

$$\frac{\partial}{\partial x} g_{mn}(x, y) = \frac{m+1}{2} P_{m-1}^{(1,1)} \left( 2 \frac{1+x}{1-y} - 1 \right) \left( \frac{1-y}{2} \right)^{m-1} P_n^{(2m+1,0)}(y).$$

Für die partielle Ableitung nach  $y$  für  $m, n \neq 0$  erhalten wir:

$$\begin{aligned} \frac{\partial}{\partial y} P_m^{(0,0)} \left( 2 \frac{1+x}{1-y} - 1 \right) &= (m+1) \frac{1+x}{(1-y)^2} P_{m-1}^{(1,1)} \left( 2 \frac{1+x}{1-y} - 1 \right) \\ \frac{\partial}{\partial y} \left( \frac{1-y}{2} \right)^m &= -\frac{m}{2} \left( \frac{1-y}{2} \right)^{m-1} \\ \frac{\partial}{\partial y} P_n^{(2m+1,0)}(y) &= \frac{2m+n+2}{2} P_{n-1}^{(2m+2,1)}(y). \end{aligned}$$

Sei

$$\begin{aligned} h_1(x, y) &= \left[ \frac{\partial}{\partial y} P_m^{(0,0)} \left( 2 \frac{1+x}{1-y} - 1 \right) \right] \left( \frac{1-y}{2} \right)^m P_n^{(2m+1,0)}(y) \\ &= \frac{m+1}{2^m} (1+x)(1-y)^{m-2} P_{m-1}^{(1,1)} \left( 2 \frac{1+x}{1-y} - 1 \right) P_n^{(2m+1,0)}(y), \\ h_2(x, y) &= P_m^{(0,0)} \left( 2 \frac{1+x}{1-y} - 1 \right) \left[ \frac{\partial}{\partial y} \left( \frac{1-y}{2} \right)^m \right] P_n^{(2m+1,0)}(y) \\ &= -\frac{m}{2^m} (1-y)^{m-1} P_m^{(0,0)} \left( 2 \frac{1+x}{1-y} - 1 \right) P_n^{(2m+1,0)}(y) \end{aligned}$$

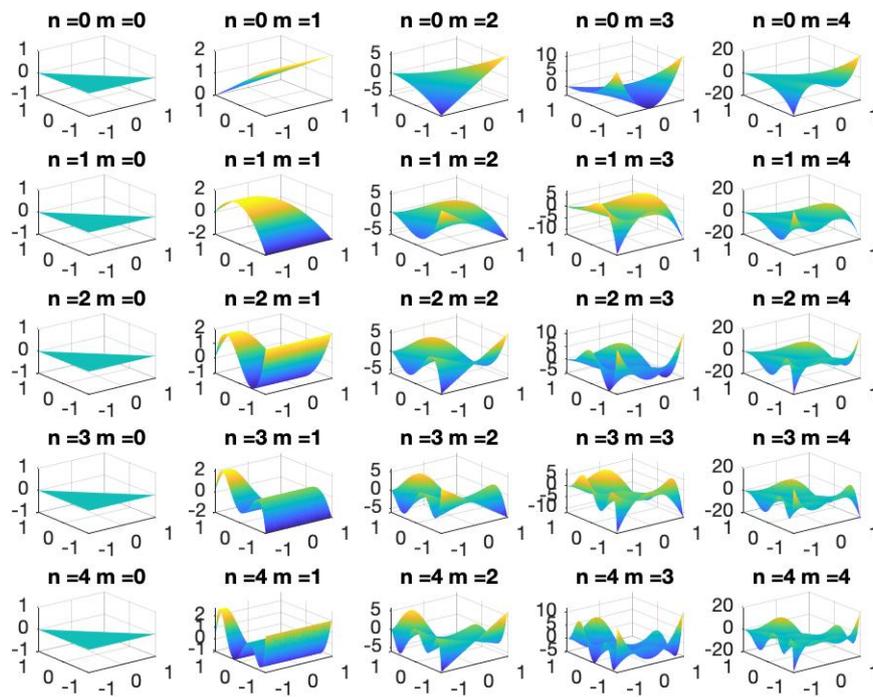
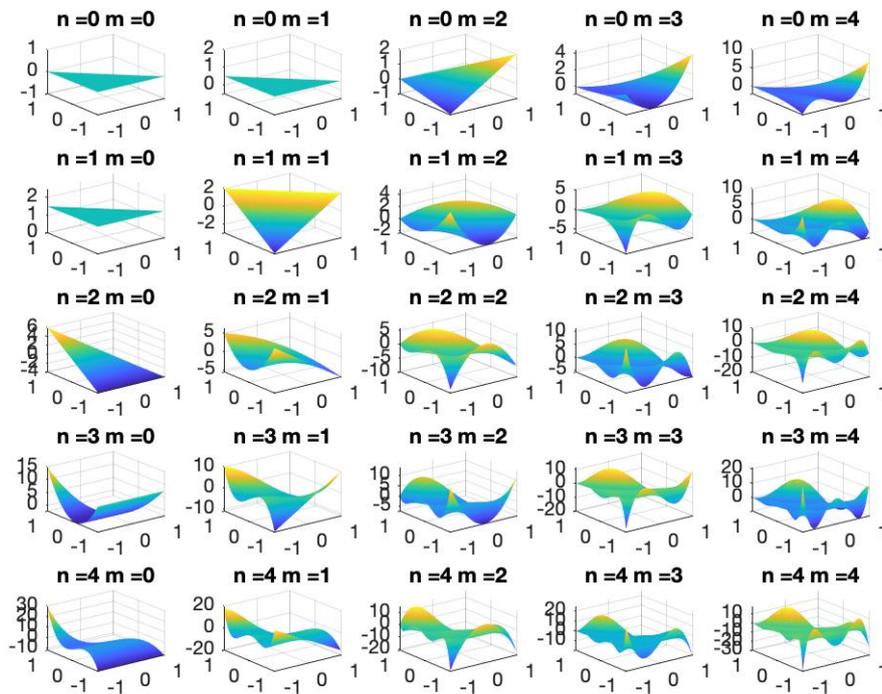
und

$$\begin{aligned} h_3(x, y) &= P_m^{(0,0)} \left( 2 \frac{1+x}{1-y} - 1 \right) \left( \frac{1-y}{2} \right)^m \left[ \frac{\partial}{\partial y} P_n^{(2m+1,0)}(y) \right] \\ &= \frac{2m+n+2}{2^{m+1}} (1-y)^m P_m^{(0,0)} \left( 2 \frac{1+x}{1-y} - 1 \right) P_{n-1}^{(2m+2,1)}(y). \end{aligned}$$

Dann ist die Summe dieser drei Funktionen die partielle Ableitung der Dubiner-Polynome nach  $y$  :

$$\frac{\partial}{\partial y} g_{mn}(x, y) = h_1(x, y) + h_2(x, y) + h_3(x, y).$$

Die Ableitungen der Dubiner-Polynome sind im Anhang dargestellt.

Abbildung 4.2: Dubiner-Polynome nach  $x$  abgeleitet auf dem ReferenzdreieckAbbildung 4.3: Dubiner-Polynome nach  $y$  abgeleitet auf dem Referenzdreieck

## 4.2 Fekete-Punkte

Folgende Überlegungen stammen aus [TWV00].

Die Gauß-Lobatto Quadraturpunkte werden aufgrund der genauen Polynominterpolationseigenschaften hoher Ordnung auf Tensorprodukt-Gebieten verwendet. Auf Nicht-Tensorprodukt-Gebieten, wie Dreiecke, funktioniert dies nicht. Da Fekete-Punkte als Gauß-Lobatto-Punkte auf Linien und im  $d$ -dimensionalen Würfel bekannt sind, sind Fekete-Punkte eine mögliche Verallgemeinerung von Gauß-Lobatto-Punkten für das Dreieck. Fekete-Punkte sind gut geeignete Interpolationspunkte. Im nächsten Abschnitt schauen wir uns diese Eigenschaften an.

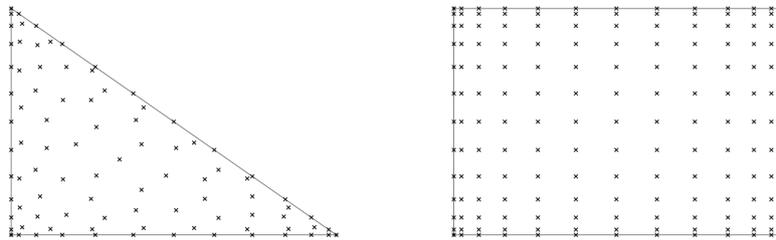


Abbildung 4.4:  $N=12$  links: Fekete-Punkte rechts: GLL-Punkte [DPR08]

Sei  $N \in \mathbb{N}$  und  $\mathcal{P}_N := \mathcal{P}_N(T_0)$  der Raum der Polynome von Grad kleiner gleich  $N$  auf dem Gebiet  $T_0$  und  $d := \dim \mathcal{P}_N = \frac{1}{2}(N+1)(N+2)$ . Sei  $f$  eine beliebige Funktion gegeben und eine Punktmenge  $\{\zeta_i\}$ , so dass eine eindeutige Funktion  $\mathfrak{g} \in \mathcal{P}_N$  existiert, welche folgende Eigenschaft erfüllt:

$$\mathfrak{g}(\zeta_i) = f(\zeta_i), \quad \forall 1 \leq i \leq d.$$

Wir bezeichnen  $\mathfrak{g}$  als Interpolationspolynom mit

$$\mathfrak{g}(\zeta) = (\mathcal{I}_N f)(\zeta).$$

Wir wählen eine Funktion  $h \in \mathcal{P}_N$ , welche  $f$  am besten in der Maximumsnorm approximiert, wobei die Maximumsnorm gegeben ist durch

$$\|f\| = \max_{\zeta \in T_0} |f(\zeta)|.$$

Diese Funktion ist nicht unbedingt  $\mathcal{I}_N(f)$ , aber  $h = \mathcal{I}_N(h)$ , da  $h \in \mathcal{P}_N$  und die Eindeutigkeit der Interpolation. Daraus folgt

$$\|f - \mathcal{I}_N(f)\| = \|f - h + \mathcal{I}_N(h) - \mathcal{I}_N(f)\| \quad (4.3)$$

$$\leq \|f - h\| + \|\mathcal{I}_N\| \|h - f\| \quad (4.4)$$

$$= (1 + \|\mathcal{I}_N\|) \|f - h\|, \quad (4.5)$$

wobei hier folgende Operatornorm impliziert wird:

$$\|\mathcal{I}_N\| = \max_{\|f\|=1} \|\mathcal{I}_N(f)\|.$$

$\|\mathcal{I}_N\|$  wird die Lebesgue-Konstante  $\Lambda$  genannt. Die Lebesgue-Konstante gibt an, wie gut die Approximation  $\mathcal{I}_N(f)$  in Bezug auf das beste approximierende Polynom zu  $f$  ist.

Wir wissen, dass Punkte mit gleichem Abstand eine Lebesgue-Konstante haben, die exponentiell wächst. Bei einer Menge von Punkten mit der kleinstmöglichen Lebesgue-Konstante in Abhängigkeit vom gewählten Gebiet und dem gewählten Funktionsraum sind diese Punkte die optimalen Interpolationspunkte. Deshalb werden sie die Lebesgue-Punkte genannt.

Über Lebesgue-Punkte höher als eine Dimension ist fast nichts bekannt, nicht einmal eine numerische Methode, um diese zu berechnen. Chen und Babuška haben theoretisch und numerisch gezeigt, dass die Fekete-Punkte in zwei Dimensionen nahezu optimal sind [CB95]. Daraus folgt, dass die Fekete-Punkte eine sinnvolle Alternative zu den Lebesgue-Punkten sind. Bevor wir sie definieren können, führen wir eine verallgemeinerte Form der Vandermonde-Matrix ein.

Für eine beliebige Basis  $\{g_i\} \subset \mathcal{P}_N$ ,  $i = 1, \dots, d$  und einer gegebenen Mengen von Punkten  $\{\zeta_i\}$ ,  $i = 1, \dots, d$  nennen wir  $V(\zeta_1, \dots, \zeta_d) = (g_j(\zeta_i))_{ij} \in \mathbb{R}^{d \times d}$  die verallgemeinerte Vandermonde-Matrix.

**Definition 4.2.1.** Sei  $\{g_i\}$ ,  $i = 1, \dots, d$ , eine Basis von  $\mathcal{P}_N$ . Dann heißen  $\{\zeta_i\}$ ,  $i = 1, \dots, d$ , Fekete-Punkte auf  $T_0$ , falls sie die Determinante von  $V$  maximieren:

$$\max_{\zeta_i \in T_0} |V(\zeta_1, \dots, \zeta_d)|.$$

Fekete-Punkte sind unabhängig von der Wahl der Basis, da jede Änderung der Basis lediglich die Determinante mit einer Konstante unabhängig von den Punkten multipliziert. Bos hat bewiesen, dass die Fekete-Punkte auf dem Quadrat einzigartig und durch das Tensorprodukt der Gauß-Lobatto-Punkten gegeben sind [BTW01]. Es gibt einige Ergebnisse für Fekete-Punkte auf einem Dreieck. Wenn wir annehmen, dass die Vandermonde-Matrix nicht singulär ist, dann gibt es eine maximale Anzahl von Punkten, die auf dem Rand liegen können. Bos vermutete, dass Fekete-Punkte auf dem Dreieck die maximale Anzahl von Punkten am Rand haben werden [Bos91]. Unter dieser Annahme bewies er, dass die Randpunkte die eindimensionalen Gauß-Lobatto-Punkte sind. Diese Vermutung wurde in [BTW01] numerisch bestätigt.

Die Lagrangsche Polynome  $L_i \in \mathcal{P}_N$  lassen sich durch die Punkte im Dreieck wie folgt auswerten:

$$L_i(\zeta) = \frac{|V(\zeta_1, \dots, \zeta_{i-1}, \zeta, \zeta_{i+1}, \dots, \zeta_d)|}{|V(\zeta_1, \dots, \zeta_d)|},$$

wobei  $V$  die Vandermonde-Matrix ist.

**Lemma 4.2.1.** *Für die Fekete-Punkte erreicht das Polynom  $\sum_{i=1}^d L_i^2(\zeta)$  sein Maximalwert von 1 und nur an den Punkten  $\zeta_i$  selbst.*

**Beweis.** *Der Beweis ist in [BTW01] nachzulesen.*

Wenn wir  $L_i(\zeta)$  an den Fekete-Punkten auswerten, erhalten wir eine obere Grenze für die Lebesgue-Konstante:

$$\Lambda_N = \max_{\zeta \in T_0} \sum_{i=1}^d |L_i(\zeta)| \leq d = \frac{1}{2}(N+1)(N+2).$$

Die Lebesgue-Konstante ist in Wirklichkeit kleiner. In [TW99] und [TWV00] wurde eine Methode zur Berechnung von Fekete-Punkten im Dreieck entwickelt. Unter Verwendung des Algorithmus für den steilsten Anstieg wurde die Determinante der verallgemeinerten Vandermonde-Matrix maximiert. Die numerischen Ergebnisse zeigten, dass die Grenze für die Lebesgue-Konstante im Dreieck  $C\sqrt{d}$  ist.

In Tabelle 4.1 sind die Lebesgue-Konstanten für die Fekete-Punkte dargestellt. Abbildung 4.5 zeigt die Fekete-Punkte auf dem Standarddreieck  $T_0$  für die Polynomgrade  $N = 6, 9, 12$ . Die dazugehörigen kartesischen Koordinaten sind im Anhang nachzulesen. An jeder Kante befinden sich  $N+1$  Punkte, die wie die Gauß-Lobatto-Punkte verteilt sind und die übrigen  $\frac{1}{2}(N-1)(N-2)$  Punkte liegen innerhalb des Dreiecks  $T_0$ .

Zusammenfassend lassen sich folgende Vorteile der Fekete-Punkte nennen:  
 Einmal sind Fekete-Punkte GLL-Punkte für den Würfel, dann sind Fekete-Punkte auf dem Dreieck GLL-Punkte an den Rändern und die Lagrange-Polynome erreichen an den Fekete-Punkten den Maximalwert 1 [DPR08].

$N$	6	9	12	15	18
$\Lambda$	4,17	6,80	9,60	9,97	13,5

Tabelle 4.1: Lebesgue-Konstante  $\Lambda_N$  der Fekete-Punkte in Abhängigkeit vom Polynomgrad  $N$  [Tay00]

### 4.3 Spektrale Differentiation auf dem Standarddreieck $T_0$

Seien für  $1 \leq i, j \leq d$ ,

$$V = V(\zeta_1, \dots, \zeta_d) = (g_j(\zeta_i))_{ij} \in \mathbb{R}^{d \times d}$$

die allgemeine Vandermonde-Matrix auf  $T_0$ , wobei  $\zeta_i$  die Fekete-Punkte und  $g_j$  die Dubiner-Polynome sind und seien

$$L_i \in \mathcal{P}_N(T_0), L_i(\zeta_j) = \delta_{ij}$$

die Lagrangschen Polynome von Grad  $\leq N$  auf den Fekete Punkten.

Sei  $v$  ein beliebiges Element aus  $\mathcal{P}_N(T_0)$ . Wir können  $v$  sowohl als Linearkombination

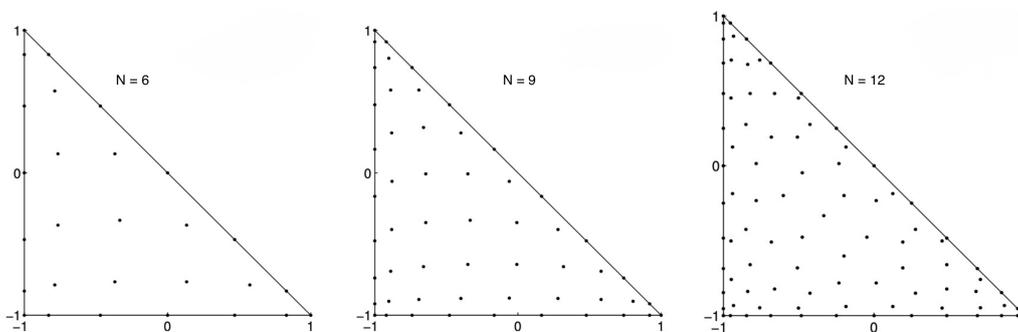


Abbildung 4.5: Fekete Punkte auf  $T_0$

aus den Lagrangschen Polynomen als auch aus den Dubiner Basen  $\mathcal{B}_N$  schreiben:

$$v(x, y) = \sum_{i=1}^d v_i L_i(x, y) = \sum_{k=1}^d \hat{v}_k g_k(x, y).$$

Es gilt [KS05]

$$\hat{v} = V^{-1}v, \tag{4.6}$$

wobei  $v = (v_1, \dots, v_d)^T$  und  $\hat{v} = (\hat{v}_1, \dots, \hat{v}_d)^T$ .

Wir erhalten die erste spektrale Ableitung  $D_x$  und  $D_y$  auf dem Dreieck  $T_0$ , indem wir die Ableitung der Dubiner Polynome bestimmen und dann mit Hilfe von (4.6) transformieren [Hei06].

$$D_x = U_x V^{-1} \text{ und } D_y = U_y V^{-1},$$

wobei

$$U_x = \left( \frac{\partial}{\partial x} g_j(\zeta_i) \right)_{ij} \text{ und } U_y = \left( \frac{\partial}{\partial y} g_j(\zeta_i) \right)_{ij}$$

für  $1 \leq i, j \leq d$ .

Zur Diskretisierung der Rand- und Interface-Bedingungen benötigen wir die Normalenableitungen auf den Kanten  $E_1, E_2$  und  $E_3$  des Standarddreiecks  $T_0$ , welche durch

$$D_n^{E_1} = D_x + D_y \tag{4.7}$$

$$D_n^{E_2} = -D_x \tag{4.8}$$

$$D_n^{E_3} = -D_y \tag{4.9}$$

gegeben sind. Hierbei wählen wir die Normalenrichtung, die aus dem Dreieck hinaus zeigt. Deshalb haben wir in (4.8) und (4.9) ein negatives Vorzeichen.

Die partiellen Ableitungen zweiter Ordnung in x- und y-Richtung erhalten wir, indem wir die jeweilige Richtungsableitung erster Ordnung quadrieren bzw. mehrfach anwenden [Wel97]:

$$D_{xx} = (D_x)^2 \text{ und } D_{yy} = (D_y)^2. \tag{4.10}$$

Die diskrete  $L^2$ -Norm für eine Funktion  $u$  auf einem Dreieck  $T$  inklusive Kollokati-

onspunkte  $x_1, x_2, \dots, x_d \in T$  ist wie folgt definiert:

$$\|u\|_{2,dis,T} = \left( \frac{1}{d} \sum_{i=1}^d u(x_i)^2 \right)^{1/2},$$

wobei  $x_1, x_2, \dots, x_d \in T$ ,  $d$  die Anzahl der gegebenen Punkte. Analog gilt dies für Rechtecke  $R$  anstatt Dreiecke  $T$ . Seien  $T_1$  und  $T_2$  benachbarte Dreiecke einer Triangulierung,  $\Omega = T_1 \cup T_2$ , wobei  $x_1, x_2, \dots, x_k \in T_1$  und  $y_1, y_2, \dots, y_l \in T_2$  Kollokationspunkte in  $T_1$  und  $T_2$  und  $v$  eine Funktion auf  $\Omega$ . Dann gilt

$$\|v\|_{2,dis,\Omega} = \left( \frac{1}{k+l} \left( \sum_{i=1}^k v(x_i)^2 + \sum_{j=1}^l v(y_j)^2 \right) \right)^{1/2}.$$

## 4.4 Formulierung auf einem allgemeinen Dreieck $T_k$

Die Berechnung der zuvor eingeführten Ableitungsmatrizen auf einem beliebigen Dreieck  $T_k$  mit den Eckpunkten  $P_i = (\hat{x}_i, \hat{y}_i)$  für  $i = 1, 2, 3$  soll dadurch vereinfacht werden, dass die Berechnung auf das Standarddreieck  $T_0$  zurückgeführt wird.

Hierfür definieren wir eine Bijektion durch  $\chi : T_0 \rightarrow T_k$  mit

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} r_1 + r_2x + r_3y \\ s_1 + s_2x + s_3y \end{pmatrix}. \quad (4.11)$$

Die inverse Abbildung  $\chi^{-1} : T_k \rightarrow T_0$  ist gegeben durch

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{r_2s_3 - r_3s_2} \begin{pmatrix} r_1s_3 - r_3s_1 + s_3\hat{x} - r_3\hat{y} \\ r_2s_1 - r_1s_2 - s_2\hat{x} + r_2\hat{y} \end{pmatrix}.$$

Für die Eckpunkte  $P_i = (\hat{x}_i, \hat{y}_i)$  für  $i = 1, 2, 3$  des Dreiecks  $T_k$  gilt

$$\begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \hat{x}_2 + \hat{x}_3 \\ \hat{x}_2 - \hat{x}_1 \\ \hat{x}_3 - \hat{x}_1 \end{pmatrix}$$

und

$$\begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \hat{y}_2 + \hat{y}_3 \\ \hat{y}_2 - \hat{y}_1 \\ \hat{y}_3 - \hat{y}_1 \end{pmatrix}.$$

Die Jacobi-Determinante von  $\chi$  lautet:

$$|J| = \begin{vmatrix} \frac{\partial \hat{x}}{\partial x} & \frac{\partial \hat{x}}{\partial y} \\ \frac{\partial \hat{y}}{\partial x} & \frac{\partial \hat{y}}{\partial y} \end{vmatrix} = r_2 s_3 - s_2 r_3. \quad (4.12)$$

Die spektralen Ableitungen erster Ordnung von  $T_k$  lauten

$$D_{\hat{x}} = \frac{1}{|J|} (s_3 D_x - s_2 D_y)$$

und

$$D_{\hat{y}} = \frac{1}{|J|} (-r_3 D_x + r_2 D_y).$$

Die spektralen Ableitungen zweiter Ordnung sehen nach (4.10) wie folgt aus:

$$D_{\hat{x}\hat{x}} = \left( \frac{1}{|J|} (s_3 D_x - s_2 D_y) \right)^2$$

und

$$D_{\hat{y}\hat{y}} = \left( \frac{1}{|J|} (-r_3 D_x + r_2 D_y) \right)^2.$$

Um die Normalenableitungen zu bestimmen, müssen wir zuerst die Normalenvektoren der Kanten von  $T_k$  bestimmen. Die Eckpunkte des allgemeinen Dreiecks  $T_k$  sind nach (4.11)

$$\hat{P}_1 = \begin{pmatrix} r_1 - r_2 - r_3 \\ s_1 - s_2 - s_3 \end{pmatrix}, \hat{P}_2 = \begin{pmatrix} r_1 + r_2 - r_3 \\ s_1 + s_2 - s_3 \end{pmatrix}, \hat{P}_3 = \begin{pmatrix} r_1 - r_2 + r_3 \\ s_1 - s_2 + s_3 \end{pmatrix}.$$

Daraus folgt für die Seiten des Dreiecks  $T_k$

$$\hat{E}_1 = \hat{P}_3 - \hat{P}_2 = 2 \begin{pmatrix} -r_2 + r_3 \\ -s_2 + s_3 \end{pmatrix}, \hat{E}_2 = \hat{P}_1 - \hat{P}_3 = 2 \begin{pmatrix} -r_3 \\ -s_3 \end{pmatrix}, \hat{E}_3 = \hat{P}_2 - \hat{P}_1 = 2 \begin{pmatrix} r_2 \\ s_2 \end{pmatrix}.$$

Die Normalenvektoren, die aus dem Dreieck  $T_k$  hinaus zeigen, lauten

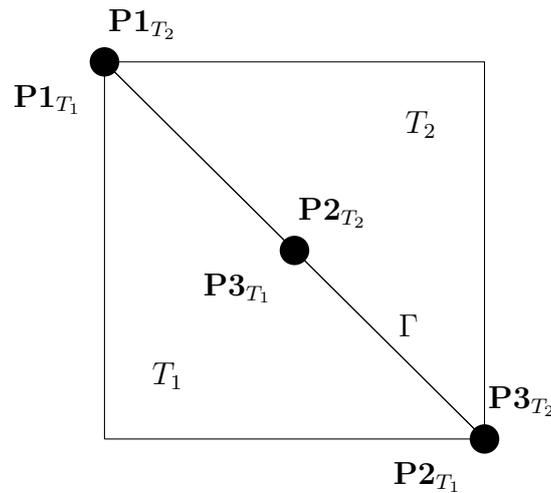
$$\vec{n}_1 = \begin{pmatrix} -s_2 + s_3 \\ r_2 - r_3 \end{pmatrix}, \vec{n}_2 = \begin{pmatrix} -s_3 \\ r_3 \end{pmatrix}, \vec{n}_3 = \begin{pmatrix} s_2 \\ -r_2 \end{pmatrix}. \quad (4.13)$$

Wir erhalten die Normalenableitungen

$$\begin{aligned} D_n^{\hat{E}_1} &= (-s_2 + s_3)D_{\hat{x}} + (r_2 - r_3)D_{\hat{y}} , \\ D_n^{\hat{E}_2} &= -s_3D_{\hat{x}} + r_3D_{\hat{y}} , \\ D_n^{\hat{E}_3} &= s_2D_{\hat{x}} - r_2D_{\hat{y}} . \end{aligned}$$

## 4.5 Interface Permutationen

In beiden zu betrachteten Fällen, der Dirichlet-Neumann und der Neumann-Neumann Iteration, werden Bedingungen auf den Interfaces der Dreiecke gefordert. Die Bedingungen zeichnen sich durchs Gleichsetzen oder Zuordnen von expliziten Werten an Knotenpunkten der angrenzenden Gebiete aus. Während der Implementierung ist die Sortierung der Punkte zu beachten, damit beim Gleichsetzen der Werte eine korrekte Zuordnung stattfindet. Zur Veranschaulichung betrachten wir das folgende Problem:



Hier treffen die zwei Dreiecke  $T_1$  und  $T_2$  aufeinander. Entlang des gemeinsamen Interfaces  $\Gamma$  befinden sich jeweils drei Punkte assoziiert mit  $T_1$  also  $\mathbf{P1}_{T_1}$ ,  $\mathbf{P2}_{T_1}$ ,  $\mathbf{P3}_{T_1}$  und drei entsprechende Punkte assoziiert mit  $T_2$  demnach  $\mathbf{P1}_{T_2}$ ,  $\mathbf{P2}_{T_2}$ ,  $\mathbf{P3}_{T_2}$ . Im globalen Kontext betrachtet bilden beide Gruppen von Punkten dieselben drei Punkte ab, sie sind jedoch lokal, aus der Sicht der jeweiligen Dreiecke, verschieden nummeriert. Verwenden wir diese Punkte als Kollokationspunkte und forcieren eine Gleichheitsbedingung, fordern wir Folgendes

$$u_{T_1}(\mathbf{x}_i) = u_{T_2}(\mathbf{x}_i), \quad \forall \mathbf{x}_i \text{ Kollokationspunkt auf } \Gamma.$$

Daraus ergeben sich für unser Beispiel drei Bedingungen

$$\begin{aligned} u_{T_1}(\mathbf{P1}_{T_1}) &= u_{T_2}(\mathbf{P1}_{T_2}), \\ u_{T_1}(\mathbf{P3}_{T_1}) &= u_{T_2}(\mathbf{P2}_{T_2}), \\ u_{T_1}(\mathbf{P2}_{T_1}) &= u_{T_2}(\mathbf{P3}_{T_2}). \end{aligned} \tag{4.14}$$

Aus lokaler Sicht werden Punkte immer im selben Schema gespeichert. Eine Möglichkeit, die Punkte auf dem Interface  $\Gamma$  zu speichern, wäre als Vektor der Form

$$\begin{pmatrix} u_{T_i}(\mathbf{P1}_{T_i}) \\ u_{T_i}(\mathbf{P2}_{T_i}) \\ u_{T_i}(\mathbf{P3}_{T_i}) \end{pmatrix}, i = 1, 2.$$

Um ein Gleichsetzen der Vektoren bezüglich  $T_1$  und  $T_2$  zu ermöglichen, verwenden wir eine zusätzliche Permutationsmatrix, die eine richtige Zuordnung ermöglicht.

$$\begin{pmatrix} u_{T_1}(\mathbf{P1}_{T_1}) \\ u_{T_1}(\mathbf{P2}_{T_1}) \\ u_{T_1}(\mathbf{P3}_{T_1}) \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{\text{Permutation}} \begin{pmatrix} u_{T_2}(\mathbf{P1}_{T_2}) \\ u_{T_2}(\mathbf{P2}_{T_2}) \\ u_{T_2}(\mathbf{P3}_{T_2}) \end{pmatrix} \Leftrightarrow \begin{pmatrix} u_{T_1}(\mathbf{P1}_{T_1}) \\ u_{T_1}(\mathbf{P2}_{T_1}) \\ u_{T_1}(\mathbf{P3}_{T_1}) \end{pmatrix} = \begin{pmatrix} u_{T_2}(\mathbf{P1}_{T_2}) \\ u_{T_2}(\mathbf{P3}_{T_2}) \\ u_{T_2}(\mathbf{P2}_{T_2}) \end{pmatrix}.$$

Somit können wir mit Hilfe der Permutationsmatrix, die im selben Schema gespeicherten, Werte verwenden und gleichzeitig die Bedingung in (4.14) erfüllen. Eine solche Permutationsmatrix muss im Zweifel für jedes Interface einmalig erstellt werden. Dieses Beispiel ist die einfachste Form, durch Anwachsen der Kollokationspunkten auf dem Interface, als Resultat von Verwendung Polynomen höherer Ordnung, vergrößert sich die Dimension des Vektors, sowie die der assoziierten Permutationsmatrix.

# Kapitel 5

## Numerische Analysis der dreiecksbasierten Spektralmethode

Das Verfahren, welches wir in dieser Arbeit hergeleitet haben, stützt sich im Grunde auf drei Teilüberlegungen, wenn wir die Lösungseigenschaft dessen untersuchen möchten. Dabei lassen sich die drei verschiedenen Gebiete unabhängig und teils isoliert voneinander betrachten. Folgende Überlegung hilft zur strukturierten Betrachtung des Verfahrens:

- In der Hierarchie der angewandten Teilverfahren stehen die Gebietszerlegungsmethoden, Dirichlet-Neumann und Neumann-Neumann an erster Stelle. Hier besteht die Frage der Lösbarkeit des Verfahrens unter der Voraussetzung von vorhandenen Lösungen auf den Teilgebieten  $\Omega_i$ , siehe Abschnitt 2.2 zur näheren Beschreibung. Setzen wir solche Lösungen voraus, dann lassen sich die gewählten Gebietszerlegungsmethoden bewerten. Die Überlegungen dazu wollen wir in Abschnitt 5.3 vornehmen.
- An zweiter Stelle der Überlegung steht die Lösbarkeit der lokalen Probleme auf jedem Teilgebiet  $\Omega_i$  unter Verwendung der Kollokationsmethode. Auf diese Überlegung wird in Abschnitt 5.2 eingegangen.
- Zuletzt ist die Kollokationsmethode in ihrer Güte abhängig von den verwendeten Kollokationspunkten, sprich in dem untersuchten Fall, den Fekete-Punkten, was in Abschnitt 5.1 erläutert wird.

Eigenschaften des Verfahrens lassen sich daher auf diese drei Teilüberlegungen reduzieren, welche in den folgenden Abschnitten erläutert werden, wobei jeweils nur Überlegungen angegeben werden, da eine all umfassende Analysis des Verfahrens den Rahmen dieser auf Anwendung ausgelegten Dissertation sprengen würde.

## 5.1 Approximationsgüte von Fekete-Punkten auf Dreiecken

Die Approximationsgüte von Fekete-Punkten wurde in Bezug auf ihre Interpolations-  
eigenschaft und Quadratureigenschaft in [TW99] besprochen.

In [MP89] wird ein Ansatz zum Beweis von Konvergenz im Zusammenhang mit  
Spektralmethoden gegeben, jedoch lassen sich diese Ergebnisse nicht in der Form  
verallgemeinern, dass dieser Ansatz auch für Fekete-Punkte gelingt. Ein grundlegen-  
des Problem besteht in der Tatsache, dass im Fall von dreiecksbasierten Methoden  
keine ausreichende Formulierung zur Struktur und Verteilung der Fekete-Punkte  
im Dreieck besteht, siehe [TWW00]. Auch zum heutigen Zeitpunkt fehlen daher es-  
sentielle Resultate, um sich dem Thema zu nähern. Nichtsdestotrotz wurde zum  
Beispiel durch numerische Berechnungen in [TW99] eine ausreichende Eigenschaft  
numerisch gezeigt. Um abseits einer rigorosen Konvergenzbetrachtung zu dreiecks-  
basierten Fekete-Punkten trotzdem die Approximationsgüte zu besprechen, wollen  
wir ein paar grundlegende Fehlerabschätzungen geben.

Sei  $f$  eine zu approximierende analytische Funktion und  $h \in \mathcal{P}_N$  die polynomia-  
le Bestapproximation, wobei in diesem Fall  $N = \dim \mathcal{P}_N$  und

$$\mathcal{P}_N = \text{span}\{x^m y^n, m + n \leq d\}$$

gilt und  $d$  den maximalen Polynomgrad angibt. Im Falle von Dreiecken gilt

$$N = (d + 1)(d + 2)/2.$$

Wie zuvor schon in (4.3) gezeigt, gilt

$$\|f - \mathcal{I}_N(f)\| \leq (1 + \|\mathcal{I}_N\|)\|f - h\|,$$

wobei  $\|\mathcal{I}_N\| \leq N \leq Cd^2$  mit einer Konstanten  $C > 0$ . Eine ähnliche Approximation  
lässt sich auch für die von uns genutzten diskreten Ableitungen herleiten:

$$\begin{aligned} \|\nabla f - \nabla \mathcal{I}_N(f)\| &= \|\nabla f - \nabla h + \nabla \mathcal{I}_N(h) - \nabla \mathcal{I}_N(f)\| \\ &\leq \|\nabla f - \nabla h\| + \|\nabla \mathcal{I}_N(f) - \nabla \mathcal{I}_N(h)\| \\ &\leq (1 + Cd^2 \|\mathcal{I}_N\|)\|f - h\|_{W_1^\infty}. \end{aligned}$$

Hierbei stellt  $\|\cdot\|_{W_1^\infty}$  die Sobolev-Norm dar und es wurde  $\|\nabla\| < Cd^2$  genutzt, was bei Einschränkung auf Polynome von Grad  $d$  gilt.

Zuletzt betrachten wir die Eigenschaften von Fekete-Punkten bezüglich numerischer Quadratur. Sei hier  $Q_N(\cdot)$  die numerische Quadratur unter Anwendung der Fekete-Punkte. Zusätzlich ist  $Q_N(h)$  exakt für  $h \in \mathcal{P}_N$ . Daher gilt:

$$\begin{aligned} \left\| \int f \, dA - Q_N(f) \right\| &= \left\| \int f \, dA - \int h \, dA + Q_N(h) - Q_N(f) \right\| \\ &\leq \left\| \int (f - h) \, dA \right\| + \|Q_N(f) - Q_N(h)\| \\ &\leq (A + \|Q_N\|)\|f - h\|, \end{aligned}$$

wobei  $A$  die Fläche des Dreiecks darstellt. Zusätzlich lässt sich  $\|Q_N\| \leq AN < Cd^2$  herleiten.

Aus den hier hergeleiteten Fehlerschranken lässt sich für die Approximation durch Fekete-Punkte, der Differentiation und Quadratur eine uniforme und exponentielle Konvergenz herleiten.

## 5.2 Konvergenzüberlegungen zur Kollokationsmethode

Die Kollokationsmethode schreibt einem Polynom Funktionswerte eines Differentialoperators an einer Menge von Kollokationspunkten zu und versucht damit die gesuchte Funktion einer Differentialgleichung zu approximieren.

Die Herleitung dieses Verfahrens wurde schon in (3.6) und (3.7) für einen elliptischen Operator  $\mathcal{L}$  und Randoperator  $\mathcal{B}$  formuliert. Im Speziellen interessieren uns in dieser Arbeit die Approximationen, welche durch die Dirichlet-Neumann bzw. Neumann-Neumann Verfahren entstehen. Die Probleme sind jeweils auf Dreiecken formuliert und stellen entweder klassische Dirichlet Randwertprobleme dar, oder gemischte Randwertprobleme mit Dirichlet- und Neumannbedingungen. Die entstehenden Probleme sind in (2.2) und (2.3) für die Dirichlet-Neumann Iteration gegeben und in (2.4) und (2.5) für die Neumann-Neumann Iteration. Letztlich lassen sich die von uns betrachteten Kollokationsprobleme auf zwei Typen reduzieren. Dazu sei  $T \in \mathcal{T}_h$  und  $\Gamma_T = \partial T \setminus (\partial T \cap \partial\Omega)$  der Rand eines Dreiecks  $T$ , welcher im Inneren des Gebietes  $\Omega$  liegt.

1. Reines Dirichlet-Problem: Gesucht ist  $u_T \in \mathcal{P}_N(T)$  für  $T \in \mathcal{T}_h$ , so dass

$$\begin{aligned} \mathcal{L}u_T(x_k) &= f(x_k) & x_k \in T \cap (\Omega \setminus \partial\Omega) \\ u_T(x_k) &= g(x_k) & x_k \in \partial T \end{aligned}$$

für alle  $x_k$  in der Menge der Fekete-Punkte passend zur Dimension von  $\mathcal{P}_N$  des Dreiecks  $T$ . Hierbei ist die Funktion  $g$  bestimmt durch den von der Differentialgleichung vorgeschriebenen Dirichlet-Randbedingung auf  $\partial\Omega$  und von Dirichlet-Bedingungen der Gebietszerlegungsmethode auf  $\Gamma_T$ .

2. Gemischtes Randwert-Problem: Gesucht ist  $u_T \in \mathcal{P}_N(T)$  für  $T \in \mathcal{T}_h$ , so dass

$$\begin{aligned} \mathcal{L}u_T(x_k) &= f(x_k) & x_k \in T \cap (\Omega \setminus \partial\Omega) \\ u_T(x_k) &= g(x_k) & x_k \in \partial T \cap \partial\Omega \\ \frac{\partial u_T}{\partial n}(x_k) &= h(x_k) & x_k \in \Gamma_T \end{aligned}$$

für alle  $x_k$  in der Menge der Fekete-Punkte passend zur Dimension von  $\mathcal{P}_N$  des Dreiecks  $T$  und  $\Gamma$  dem Interface der Dirichlet-Neumann Iteration.

Zur näheren Betrachtung untersuchen wir das reine Dirichlet-Problem. Der vorgestellte Ansatz funktioniert mit wenigen Abänderungen auch für gemischte Randwertprobleme und folgt der Herleitung aus [CHQZ10].

Im ersten Schritt wird das folgende Skalarprodukt in Abhängigkeit der verwendeten Kollokationspunkte  $x_k$  mit passenden Gewichten  $w_k$  definiert:

$$(u, v)_N = \sum_k u(x_k)v(x_k)w_k .$$

Die Gewichte  $w_k$  sollen so gewählt werden, dass folgendes gilt:

$$(u, v)_N = (u, v) \quad \forall u, v, \text{ so dass } uv \in \mathcal{P}_{2N-1}(T). \quad (5.1)$$

Im Falle von Gauß-Punkten, bzw. Gauß ähnlichen Punkten im Intervall oder Tensorprodukt des Intervalls ergeben sich diese Punkte relativ einfach. Dies lässt sich aber nicht allzu einfach auf Dreiecken verallgemeinern, mehr dazu in [TW99]. Angenommen, solche Punkte lassen sich finden, dann ist diese Eigenschaft auch erfüllbar, was sich auf einfache Resultate der Quadratureigenschaften von Gauß-Punkten stützt. Für die von uns betrachteten Kollokationspunkte, die Fekete-Punkte, gilt im allge-

meinen höchstens

$$(u, v)_N = (u, v) \quad \forall u, v, \text{ so dass } uv \in \mathcal{P}_N(T), \quad (5.2)$$

vgl. [GT06], [PRPZ06], [LTB17]. Würde (5.1) gelten, wäre es möglich den Kollokationsansatz in eine schwache Formulierung zu überführen, ähnlich eines Galerkin-Ansatzes. Im Anschluss lassen sich klassische Resultate vergleichbar der Resultate auf dem Gebiet der FEM verwenden, um weitere Eigenschaften herzuleiten und zu beweisen. Weitere Schritte lassen sich in [CHQZ10] nachlesen. Wie zuvor beschrieben, lässt sich dies nicht für Fekete-Punkte aufgrund von (5.2) umsetzen, wodurch sich der klassische Zugang nicht ergibt und es uns nicht möglich war analytische Ergebnisse für Fekete-Punkte herzuleiten.

## 5.3 Effektivität von Dirichlet-Neumann und Neumann-Neumann Methoden

Betrachten wir das schon in Abb. 2.2 gegebene vereinfachte Problem, bestehend aus zwei Gebieten  $\Omega_1$  und  $\Omega_2$ , folgendermaßen nach einer Sortierung der Freiheitsgrade bezüglich  $\Omega_1$  und  $\Omega_2$  sowie dem Interface  $\Gamma$ , so lässt sich das Gleichungssystem  $Au = f$  wie folgt beschreiben:

$$A = \begin{pmatrix} A_{II}^{(1)} & 0 & A_{I\Gamma}^{(1)} \\ 0 & A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(1)} & A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma} \end{pmatrix}, u = \begin{pmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_\Gamma \end{pmatrix}, f = \begin{pmatrix} f_I^{(1)} \\ f_I^{(2)} \\ f_\Gamma \end{pmatrix}.$$

Hierbei beschreibt  $u_I^{(1)}$  die Freiheitsgrade exklusiv assoziiert bezüglich  $\Omega_1$ ,  $u_I^{(2)}$  die Freiheitsgrade exklusiv assoziiert bezüglich  $\Omega_2$  und  $u_\Gamma$  die Freiheitsgrade auf dem Interface  $\Gamma$ . Hier und im Folgenden stützen wir uns auf die Herleitung aus [TW06]. Analog dazu verhalten sich die Blockmatrizen in  $A$  bzw. die Aufteilung in  $f$ . Der hier beschriebene Vorgang bezieht sich ausschließlich auf eine Umsortierung der Freiheitsgrade und hat keinen Einfluss auf sonstige Eigenschaften des Gleichungssystems. Die jeweiligen einzelnen Komponenten lassen sich aus den lokalen Gleichungssystemen entnehmen.

Die Matrix  $A$  lässt sich mit Hilfe der  $LR$ -Zerlegung so zerlegen:

$$A = LR = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ A_{\Gamma I}^{(1)} A_{II}^{(1)-1} & A_{\Gamma I}^{(2)} A_{II}^{(2)-1} & I \end{pmatrix} \begin{pmatrix} A_{II}^{(1)} & 0 & A_{\Gamma I}^{(1)} \\ 0 & A_{II}^{(2)} & A_{\Gamma I}^{(2)} \\ 0 & 0 & S \end{pmatrix}$$

und das Problem  $Au = f$  reduziert sich nach einer Multiplikation mit  $L^{-1}$  zu

$$\begin{pmatrix} A_{II}^{(1)} & 0 & A_{\Gamma I}^{(1)} \\ 0 & A_{II}^{(2)} & A_{\Gamma I}^{(2)} \\ 0 & 0 & S \end{pmatrix} u = \begin{pmatrix} f_I^{(1)} \\ f_I^{(2)} \\ g_\Gamma \end{pmatrix},$$

wobei

$$\begin{aligned} S &= A_{\Gamma\Gamma} - A_{\Gamma I}^{(1)} A_{II}^{(1)-1} A_{\Gamma I}^{(1)} - A_{\Gamma I}^{(2)} A_{II}^{(2)-1} A_{\Gamma I}^{(2)} =: S^{(1)} + S^{(2)}, \\ g_\Gamma &= (f_\Gamma^{(1)} - A_{\Gamma I}^{(1)} A_{II}^{(1)-1} f_I^{(1)}) + (f_\Gamma^{(2)} - A_{\Gamma I}^{(2)} A_{II}^{(2)-1} f_I^{(2)}) =: g_\Gamma^{(1)} + g_\Gamma^{(2)}. \end{aligned}$$

Mit den jeweiligen lokalen Anteilen bezüglich  $\Omega_1$  und  $\Omega_2$  ergibt sich:

$$\begin{aligned} S^{(1)} &= A_{\Gamma\Gamma}^{(1)} - A_{\Gamma I}^{(1)} A_{II}^{(1)-1} A_{\Gamma I}^{(1)}, & S^{(2)} &= A_{\Gamma\Gamma}^{(2)} - A_{\Gamma I}^{(2)} A_{II}^{(2)-1} A_{\Gamma I}^{(2)}, \\ g_\Gamma^{(1)} &= f_\Gamma^{(1)} - A_{\Gamma I}^{(1)} A_{II}^{(1)-1} f_I^{(1)}, & g_\Gamma^{(2)} &= f_\Gamma^{(2)} - A_{\Gamma I}^{(2)} A_{II}^{(2)-1} f_I^{(2)}. \end{aligned}$$

Das Gleichungssystem lässt sich nun in zwei Schritten lösen: zuerst lösen wir ein reduziertes Gleichungssystem bezüglich des Schur-Komplement

$$Su_\Gamma = g_\Gamma$$

und im Anschluss ein Update der inneren Freiheitsgrade

$$\begin{aligned} u_I^{(1)} &= A_{II}^{(1)-1} (f_I^{(1)} - A_{\Gamma I}^{(1)} u_\Gamma), \\ u_I^{(2)} &= A_{II}^{(2)-1} (f_I^{(2)} - A_{\Gamma I}^{(2)} u_\Gamma). \end{aligned}$$

### 5.3.1 Dirichlet-Neumann Iteration

Unter Verwendung der gleichen Schreibweisen lässt sich der Dirichlet-Neumann Algorithmus, beschrieben in (2.2) und (2.3), zusammenfassen durch die folgende Formeln, wobei das Dirichlet Problem beschrieben in (2.2) äquivalent ist zu (5.3) und

das Neumann Problem (2.3) äquivalent zu (5.4) ist:

$$A_{II}^{(1)} v_1^{n+\frac{1}{2}} + A_{I\Gamma}^{(1)} u_\Gamma^n = f_I^{(1)}, \quad (5.3)$$

$$\begin{pmatrix} A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(2)} \end{pmatrix} \begin{pmatrix} w_2^{n+1} \\ \tilde{u}_\Gamma^{n+1} \end{pmatrix} = \begin{pmatrix} f_I^{(2)} \\ f_\Gamma^{(2)} - \lambda_\Gamma^{n+\frac{1}{2}} \end{pmatrix}, \quad (5.4)$$

$$u_\Gamma^{n+1} = \theta \tilde{u}_\Gamma^{n+1} + (1 - \theta) u_\Gamma^n, \quad (5.5)$$

mit

$$v_1 = u_I^{(1)}, \quad w_2 = u_I^{(2)} \quad \text{und} \quad \lambda_\Gamma^{n+\frac{1}{2}} = A_{\Gamma I}^{(1)} v_1^{n+\frac{1}{2}} + A_{\Gamma\Gamma}^{(1)} u_\Gamma^n - f_\Gamma^{(1)}.$$

Unter Verwendung von (5.3) gilt

$$v_1^{n+\frac{1}{2}} = \left( A_{II}^{(1)} \right)^{(-1)} \left( f_I^{(1)} - A_{I\Gamma}^{(1)} u_\Gamma^n \right).$$

Daher ist

$$\begin{aligned} \lambda_\Gamma^{n+\frac{1}{2}} &= A_{\Gamma I}^{(1)} v_1^{n+\frac{1}{2}} + A_{\Gamma\Gamma}^{(1)} u_\Gamma^n - f_\Gamma^{(1)} \\ &= - \left( g_\Gamma^{(1)} - S^{(1)} u_\Gamma^n \right). \end{aligned}$$

Unter Verwendung des Schur-Komplementes auf (5.4) gilt

$$S^{(2)} \tilde{u}_\Gamma^{n+1} = g_\Gamma^{(2)} - \lambda_\Gamma^{n+\frac{1}{2}} \quad \Leftrightarrow \quad \tilde{u}_\Gamma^{n+1} = (S^{(2)})^{-1} (g_\Gamma^{(2)} - \lambda_\Gamma^{n+\frac{1}{2}}).$$

Eingesetzt in (5.5) ergibt sich

$$S^{(2)} (u_\Gamma^{n+1} - u_\Gamma^n) = \theta (g_\Gamma - S u_\Gamma^n), \quad (5.6)$$

welches auf eine klassische Richardson-Iteration führt mit

$$u_\Gamma^{n+1} = (I - \theta (S^{(2)})^{-1} S) u_\Gamma^n + \theta (S^{(2)})^{-1} g_\Gamma.$$

Hierbei agiert  $(S^{(2)})^{-1}$  als Vorkonditionierer und es ist leicht erkennbar, dass das Dirichlet-Neumann Verfahren somit als vorkonditionierte Richardson-Iteration verstanden werden kann.

Für den vorkonditionierten Operator gilt

$$S^{(2)-1} S = I + S^{(2)-1} S^{(1)}.$$

In [TW06] wird eine gleichmäßige Schranke für die Konditionsnummer des vorkonditionierten Operators hergeleitet. Dies verwendet die Existenz einer stabilen, diskreten harmonischen, Finite-Element-Erweiterung einer Lösung auf dem Interface  $\Gamma$  in das Innere der zerlegten Gebiete. Damit gilt

$$\kappa(S^{(2)-1}S^{(1)}) \leq \frac{C_1\tilde{C}_2}{C_2\tilde{C}_1},$$

bzw.

$$\kappa(S^{(2)-1}S) \leq C, \tag{5.7}$$

wobei die Konstante  $C > 0$  unabhängig der Dimension des Kollokationraumes ist. Infolgedessen ist die Methode optimal im Sinne der Unabhängigkeit von Konvergenzrate und Größe des zu lösenden Systems.

### 5.3.2 Neumann-Neumann Iteration

Wie zuvor im Fall der Dirichlet-Neumann Iteration, Formeln (5.3), (5.4) und (5.5) lassen sich vergleichbare Gleichungen zur Neumann-Neumann Iteration herleiten:

$$\begin{aligned} A_{II}^{(i)}v_i^{n+\frac{1}{2}} + A_{I\Gamma}^{(i)}u_\Gamma^n &= f_I^{(i)}, & i = 1, 2, \\ \begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{pmatrix} \begin{pmatrix} w_i^{n+1} \\ \eta_\Gamma^{n+1} \end{pmatrix} &= \begin{pmatrix} 0 \\ r_\Gamma \end{pmatrix}, & i = 1, 2, \\ u_\Gamma^{n+1} &= u_\Gamma^n - \theta(\eta_1^{n+1} + \eta_2^{n+1}), \end{aligned}$$

wobei das Residuum  $r_\Gamma$  gegeben ist durch

$$r_\Gamma = (A_{\Gamma I}^{(1)}v_1^{n+\frac{1}{2}} + A_{\Gamma\Gamma}^{(1)}u_\Gamma^n - f_\Gamma^{(1)}) + (A_{\Gamma I}^{(2)}v_2^{n+\frac{1}{2}} + A_{\Gamma\Gamma}^{(2)}u_\Gamma^n - f_\Gamma^{(2)}).$$

Analog zu (5.6) gilt im Falle der Neumann-Neumann Iteration:

$$u_\Gamma^{n+1} - u_\Gamma^n = \theta(S^{(1)-1} + S^{(2)-1})(g_\Gamma - Su_\Gamma^n).$$

Hierbei lässt sich einfach ablesen, dass die Neumann-Neumann Iteration als Vorkonditionierer zum Schur-Komplement verstanden werden kann, wobei der Vorkonditionierer durch  $S^{(1)-1} + S^{(2)-1}$  gegeben ist.

Das bedeutet für die vorkonditionierte Matrix:

$$(S^{(1)^{-1}} + S^{(2)^{-1}})S = (S^{(1)^{-1}} + S^{(2)^{-1}})(S^{(1)} + S^{(2)}) = S^{(2)^{-1}}S^{(1)} + 2I + (S^{(2)^{-1}}S^{(1)})^{-1}. \quad (5.8)$$

Damit reduziert sich die Frage nach der Kondition der vorkonditionierten Matrix in (5.8) auf die Frage nach der Kondition von  $S^{(2)^{-1}}S^{(1)}$ , welche schon im vorangegangenen Abschnitt zur Dirichlet-Neumann Iteration abgehandelt wurde, somit ist auch die Konditionszahl im Falle der Neumann-Neumann Iteration gleichmäßig beschränkt, vgl. (5.7).

Experimentelle Ergebnisse zur Konditionszahl einer Neumann-Neumann Iteration als Schur-Komplement Vorkonditionierer im Zusammenhang von Fekete-Punkten und der TSEM (Dreiecksbasierter Spektraler Element Methode) lassen sich in [PRPZ06] finden. Anzumerken ist, dass die in der zitierten Arbeit betrachteten Methode der TSEM unterschiedlich zu der von uns betrachteten Kollokation ist.



# Kapitel 6

## Numerische Ergebnisse für die Dirichlet-Neumann Iteration

Im folgenden betrachten wir die Poisson-Gleichung mit Dirichlet-Randbedingungen

$$\Delta u = f \quad \text{in } \Omega, \tag{6.1}$$

$$u = g \quad \text{auf } \partial\Omega. \tag{6.2}$$

Wir wenden für die Gebietszerlegungen die Patching-Methode und die Dirichlet-Neumann Iteration an, um unsere betrachteten Differentialgleichungen auf verschiedenen Gebieten zu lösen. Patching-Methoden basieren auf dem spektralen Kollokationsansatz, dabei wird das Gebiet  $\Omega$  in Teilgebiete  $\Omega_i$  aufgeteilt. Dann wird die Differentialgleichung an allen internen Kollokationspunkten zusammengestellt, während die Kollokationspunkte auf dem Interface verwendet werden, um die Stetigkeit der Lösung und ihrer ersten Ableitungen zu erzwingen [Qua87]. Auf dem Interface zwischen den zwei Teilgebieten werden iterativ Informationen ausgetauscht, bis Stetigkeit genügend genau erreicht ist. Aus der einen Richtung werden Dirichlet Information übertragen und aus der anderen Richtung Neumann Information. Wir wählen einen festen Relaxationsparameter  $\theta$  und iterieren solange, bis der Fehler auf dem Interface kleiner als ein gewünschter Wert ist. In [HL01] wird die Dirichlet-Neumann Iteration mit dem numerischen Verfahren Chebyshev-Kollokationsmethode in Kombination mit Chebyshev-Gauß-Lobatto-Punkte (CGL) angewendet, dabei werden die Punkte durch Transformation vom Viereck auf ein Dreieck übertragen, zudem bleiben die Anzahl der Punkte bei der Transformation erhalten. Die Anzahl der Kollokationspunkte auf dem Dreieck für die Fekete-Punkte lassen sich durch  $(N + 1)(N + 2)/2$  berechnen und die GLL-Punkte sowie die CGL-Punkte auf dem Viereck durch  $(N + 1)(N + 1)$ . In unseren numerischen Tests untersuchen wir die gleichen Gebiete wie in [HL01], damit wir im Anschluss vergleichen können, welche

Punkte bei der Gebietszerlegung effizienter sind. Wir verwenden auf den Vierecken die GLL-Punkte und auf Dreiecken die Fekete-Punkte.

## 6.1 Numerischer Test 1

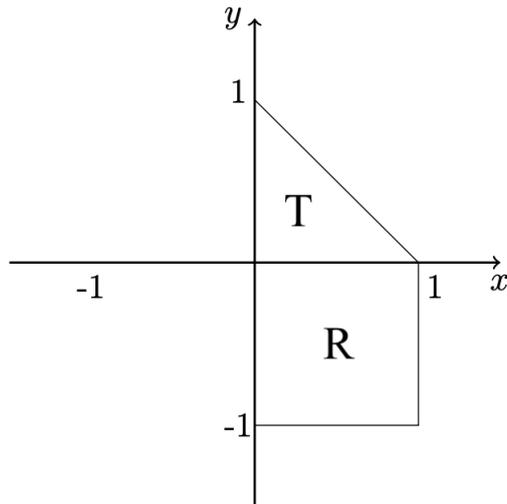


Abbildung 6.1: Gebiet  $\Omega$

Im ersten numerischen Test betrachten wir ein Gebiet  $\Omega = T \cup R$ , das in ein Dreieck

$$T = \{(x, y) \in \mathbb{R}^2 \mid 0 < x, y; x + y < 1\},$$

und Rechteck

$$R = \{(x, y) \in \mathbb{R}^2 \mid 0 < x < 1; -1 < y < 0\}$$

mit dem Interface  $\Gamma = (0, 1) \times \{0\}$  aufgeteilt ist.

Die Anfangsbedingungen sind  $u_1^0 = u_2^0 = 0$  auf  $\Omega$  und  $u_1^1 = g$  auf  $\Gamma$ .

Der Algorithmus zur Lösung des Problems lautet wie folgt:

Löse erst auf dem Dreieck

$$\begin{aligned} \Delta u_1^{n+1} &= f && \text{auf } T, \\ u_1^{n+1} &= g && \text{auf } \partial T \setminus \Gamma, \\ u_1^{n+1} &= (1 - \theta)u_1^n + \theta u_2^n && \text{auf } \Gamma \end{aligned}$$

und im Anschluss auf dem Rechteck

$$\begin{aligned}\Delta u_2^{n+1} &= f && \text{auf } R, \\ u_2^{n+1} &= g && \text{auf } \partial R \setminus \Gamma, \\ \frac{\partial u_2^{n+1}}{\partial y} &= \frac{\partial u_1^{n+1}}{\partial y} && \text{auf } \Gamma.\end{aligned}$$

Als Beispiel wenden wir den Algorithmus auf die Lösung

$$u(x, y) = \sin(\pi x) \sin\left(\pi y + \frac{\pi}{4}\right) \quad (6.3)$$

an. Dadurch ergibt sich aus (6.1) und (6.2)

$$\begin{aligned}f &= -2\pi^2 \sin(\pi x) \sin\left(\pi y + \frac{\pi}{4}\right), \\ g &= \sin(\pi x) \sin\left(\pi y + \frac{\pi}{4}\right).\end{aligned}$$

Wir wählen den Relaxationsparameter  $\theta = \frac{1}{2}$  und iterieren in unserem Beispiel solange, bis der Fehler auf dem Interface kleiner als  $1e^{-12}$  ist. Für  $N = 6, 18$ , wobei  $N$  den höchsten Grad der Polynome angibt, sind die folgende Kollokationspunkte für das Gebiet  $\Omega$  unten dargestellt:

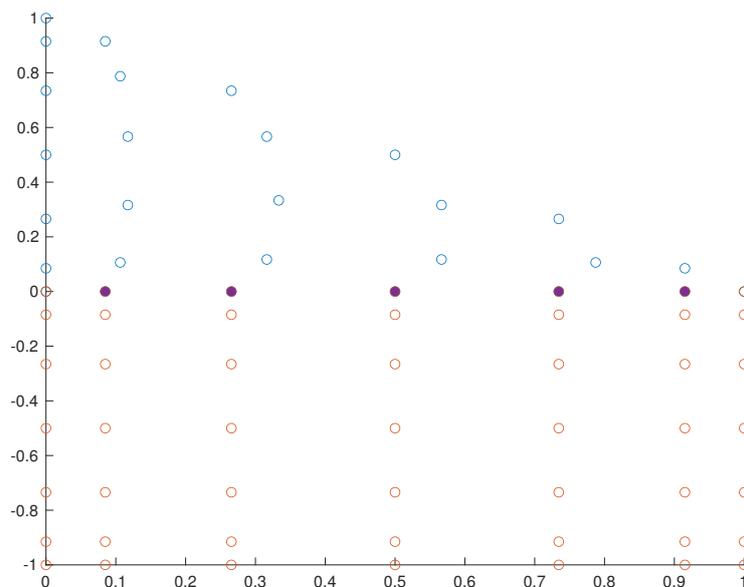


Abbildung 6.2: Kollokationspunkte  $N = 6$

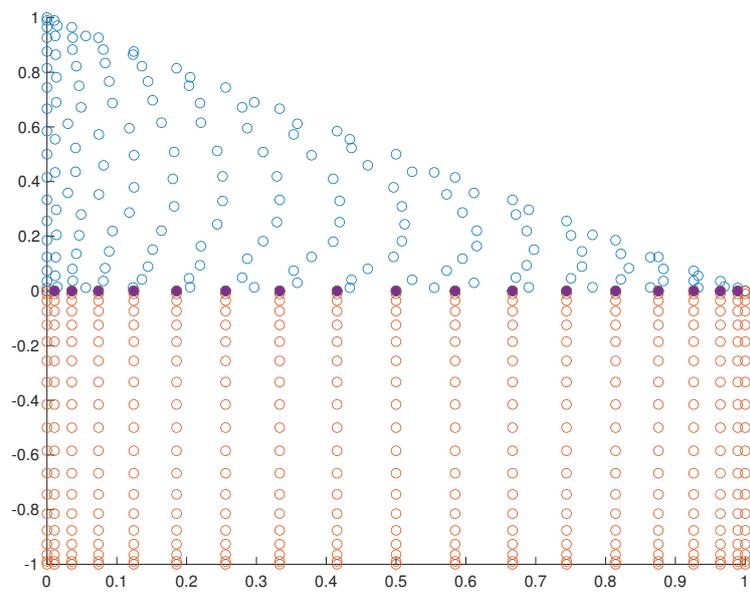


Abbildung 6.3: Kollokationspunkte  $N = 18$

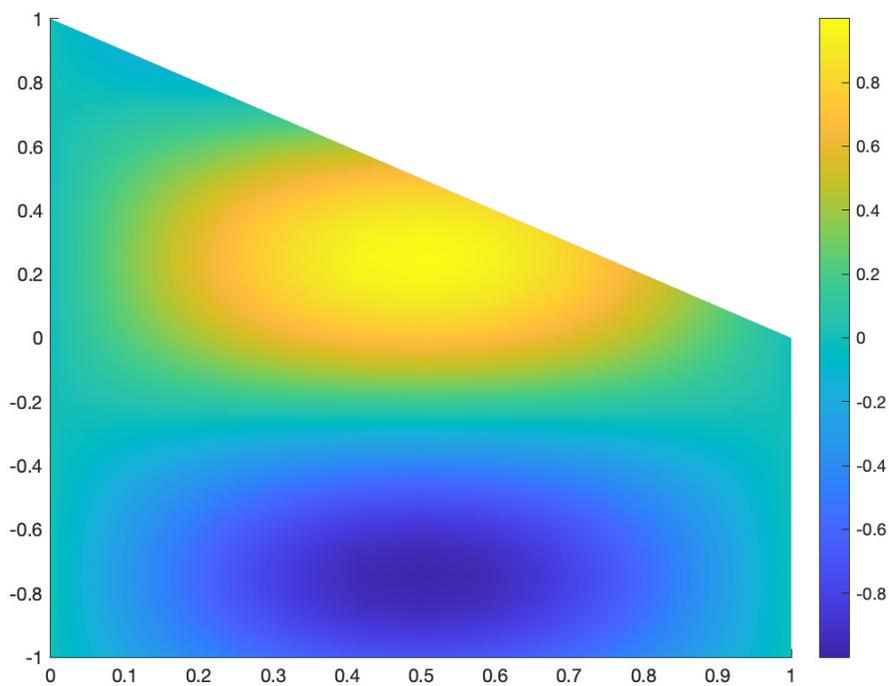
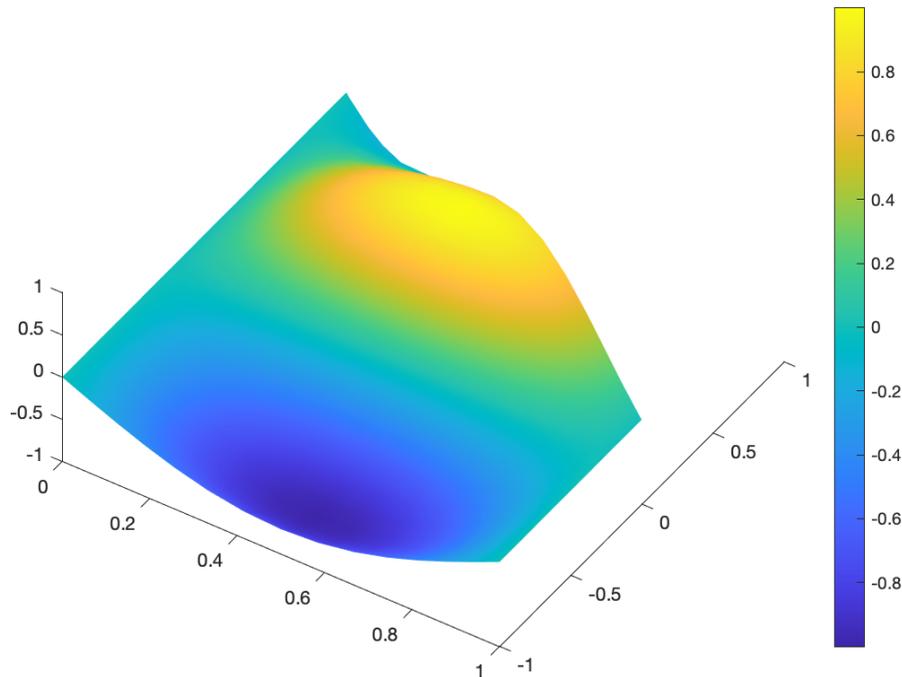


Abbildung 6.4: Numerische Lösung in  $\Omega$  mit (6.3) für  $N = 18$

Abbildung 6.5: Numerische Lösung in  $\Omega$  mit (6.3) für  $N = 18$ 

Der diskrete  $L_2$ -Fehler auf dem Dreieck zwischen der exakten Lösung  $u$  und unser Approximation  $u_a$  wird berechnet durch

$$T = \frac{\|u_a - u\|_2}{\frac{1}{2}(N+1)(N+2)}$$

und auf dem Viereck

$$R = \frac{\|u_a - u\|_2}{(N+1)(N+1)},$$

wobei  $\|v\|_2 = \sqrt{\sum_{x_i \in \text{Kollokationspunkt}} v(x_i)^2}$ .

Wenden wir die Dirichlet-Neumann Iteration mit (6.3) auf  $\Omega$  an, erhalten wir die folgenden Ergebnisse:

N	Koll.punkte	It	$T$	$R$
6	77	25	4.42e-04	2.47e-04
12	260	29	2.84e-09	1.68e-09
18	551	33	1.37e-15	3.87e-15

Tabelle 6.1:  $\text{tol}=1e^{-12}$  und  $\theta = \frac{1}{2}$

Für den Vergleich mit den Ergebnissen aus [HL01] müssen wir die Anzahl der Kollokationspunkte berücksichtigen. In unserem Verfahren haben wir für Polynomgrad 6 genau 77 Kollokationspunkte, in [HL01] liefert das Verfahren für  $N = 8$  mit CGL-Punkten 162 Kollokationspunkte und einen Fehler von  $2.42e^{-05}$  auf dem Dreieck und  $1.64e^{-05}$  auf dem Viereck. Vergleichen wir dies, ist unsere Methode trotz der höheren Anzahl an Iterationen im Vorteil, da wir weniger Kollokationspunkte bei der Implementierung benötigen und die Fehler auf den Gebieten sich nur um  $1e^{-01}$  unterscheiden. Hier liefert der höchste Polynomgrad kleinere Fehler auf den Gebieten als in [HL01] mit  $N = 16$ .

## 6.2 Numerischer Test 2

Beim zweiten Test besteht das Gebiet  $\Omega_1$  aus dem  $\Omega$  aus dem ersten Test mit einem hinzugefügten Dreieck  $T_L$ . Wir beginnen hier die Iterationen auf dem Dreieck  $T_R$  mit den Interfacegrenzen  $\Gamma_1 = (0, 1) \times \{0\}$ ,  $\Gamma_2 = \{0\} \times (0, 1)$  und  $\Gamma = \Gamma_1 \cup \Gamma_2$  und lösen im Anschluss auf  $R$  und  $T_L$ .

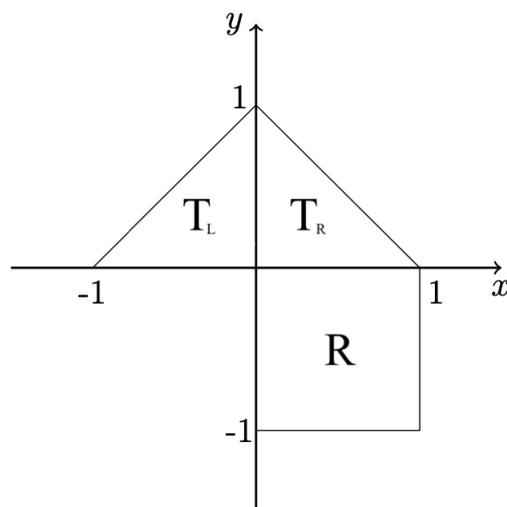


Abbildung 6.6: Gebiet  $\Omega_1$

Die Iteration auf dem Dreieck erfolgt

$$\begin{aligned}\Delta u_1^{n+1} &= f && \text{auf } T_R \\ u_1^{n+1} &= g && \text{auf } \partial T_R \setminus \Gamma, \\ u_1^{n+1} &= (1 - \theta)u_1^n + \theta u_2^n && \text{auf } \Gamma_1, \\ u_1^{n+1} &= (1 - \theta)u_1^n + \theta u_3^n && \text{auf } \Gamma_2,\end{aligned}$$

danach auf dem Rechteck

$$\begin{aligned}\Delta u_2^{n+1} &= f && \text{auf } R, \\ u_2^{n+1} &= g && \text{auf } \partial R \setminus \Gamma_1, \\ \frac{\partial u_2^{n+1}}{\partial y} &= \frac{\partial u_1^{n+1}}{\partial y} && \text{auf } \Gamma_1\end{aligned}$$

und im Anschluss auf dem anderen Dreieck

$$\begin{aligned}\Delta u_3^{n+1} &= f && \text{auf } T_L, \\ u_3^{n+1} &= g && \text{auf } \partial T_L \setminus \Gamma_2, \\ \frac{\partial u_3^{n+1}}{\partial x} &= \frac{\partial u_1^{n+1}}{\partial x} && \text{auf } \Gamma_2.\end{aligned}$$

Wir wenden den Algorithmus auf die Lösung

$$u(x, y) = \sin\left(x\pi + \frac{\pi}{4}\right) \sin\left(\pi y + \frac{\pi}{4}\right) \quad (6.4)$$

an. Dann ist wegen (6.1) und (6.2)

$$\begin{aligned}f &= -2\pi^2 \sin\left(x\pi + \frac{\pi}{4}\right) \sin\left(\pi y + \frac{\pi}{4}\right), \\ g &= \sin\left(x\pi + \frac{\pi}{4}\right) \sin\left(\pi y + \frac{\pi}{4}\right).\end{aligned}$$

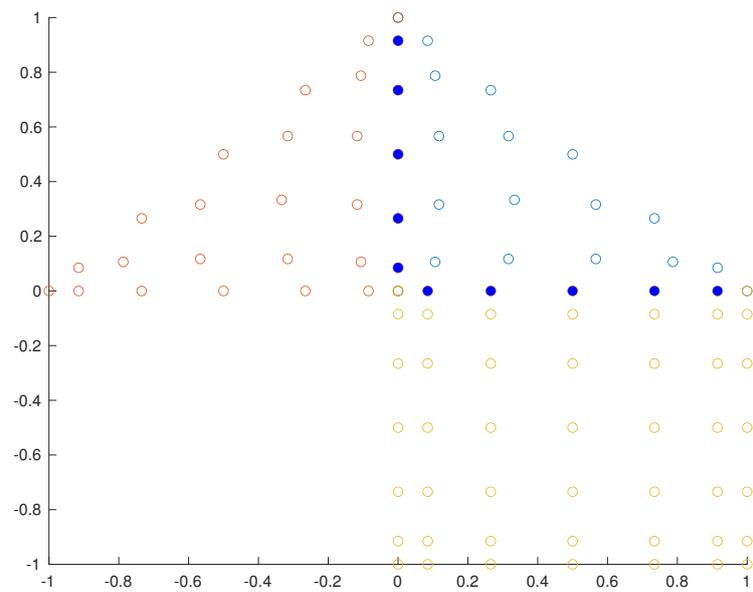


Abbildung 6.7: Kollokationspunkte  $N = 6$

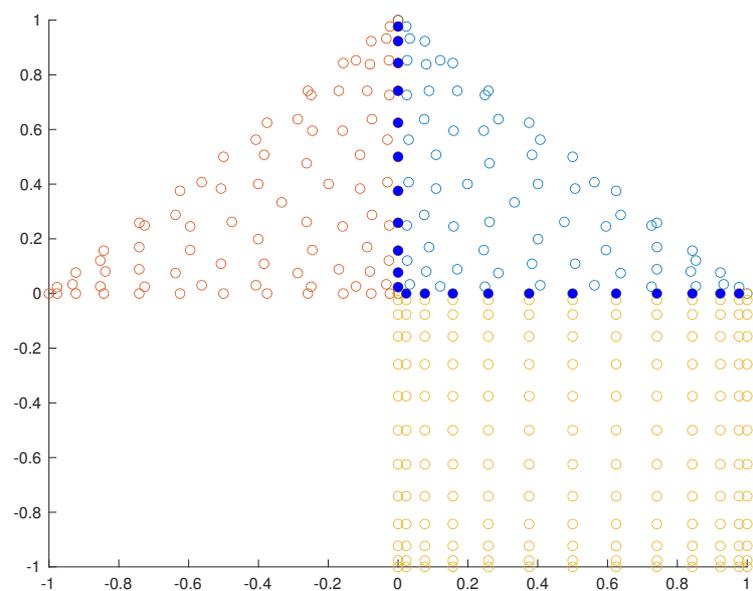
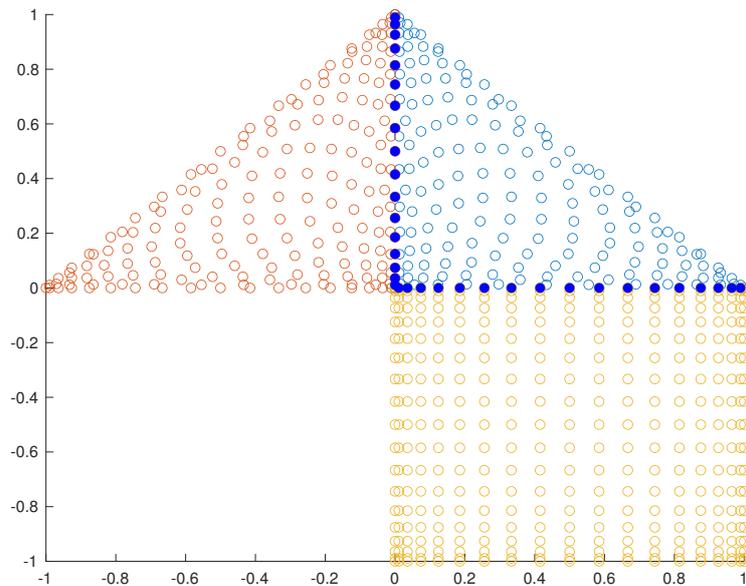


Abbildung 6.8: Kollokationspunkte  $N = 12$

Abbildung 6.9: Kollokationspunkte  $N = 18$ 

Der Relaxationsparameter wird mit  $\theta = 0.65$  gesetzt und wir iterieren hier, bis der Fehler auf dem Interface kleiner als  $1e^{-12}$  ist.

Der numerische Test liefert folgende Ergebnisse:

N	Koll.punkte	It	$T_R$	$T_L$	$R$
6	105	51	5.64e-04	4.75e-04	1.67e-04
12	351	56	3.24e-09	3.28e-09	3.59e-10
18	741	58	7.14e-15	2.78e-15	7.45e-15

Tabelle 6.2:  $\text{tol}=1e^{-12}$  und  $\theta = 0.65$ 

In [HL01] wird für  $N = 16$  insgesamt 867 Kollokationspunkte benötigt und liefert Fehler von  $1.78e^{-13}$  auf dem Dreieck  $T$ ,  $3.70e^{-14}$  auf dem Viereck  $R$  und  $1.83e^{-13}$  auf dem Dreieck  $T_1$ . Dies vergleichen wir mit unserem Fall  $N = 18$  und stellen fest, dass zwar unser Verfahren leicht mehr Iterationen benötigt, jedoch weniger Kollokationspunkte verwendet und bessere Ergebnisse liefert. Werfen wir einen Blick auf die numerischen Ergebnisse in [HL01], so sehen wir, dass für kleinere Polynomgrade beide Verfahren ähnliche Fehler auf den Gebieten haben.

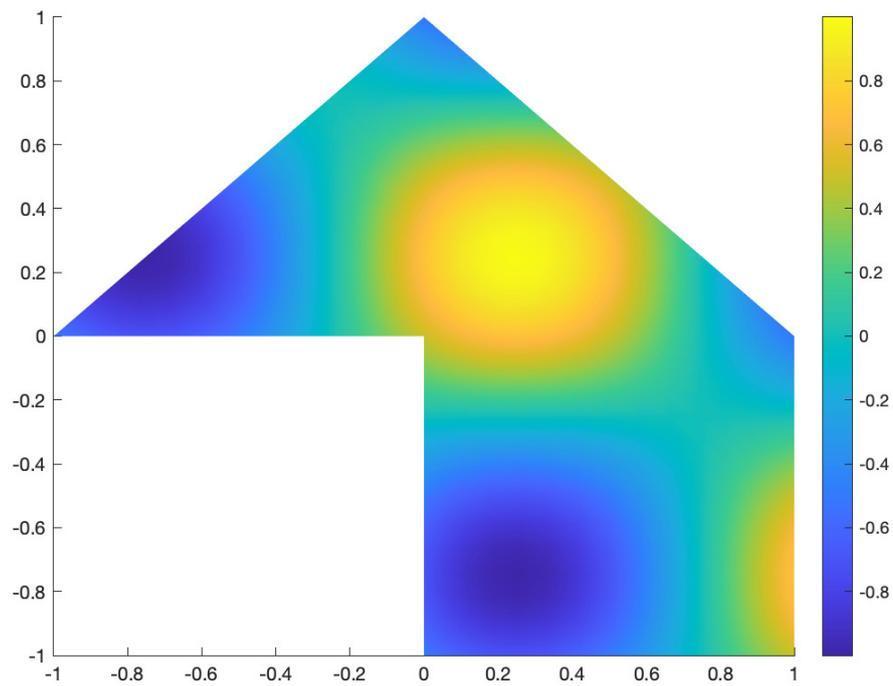


Abbildung 6.10: Numerische Lösung in  $\Omega_1$  mit (6.4) für  $N = 18$

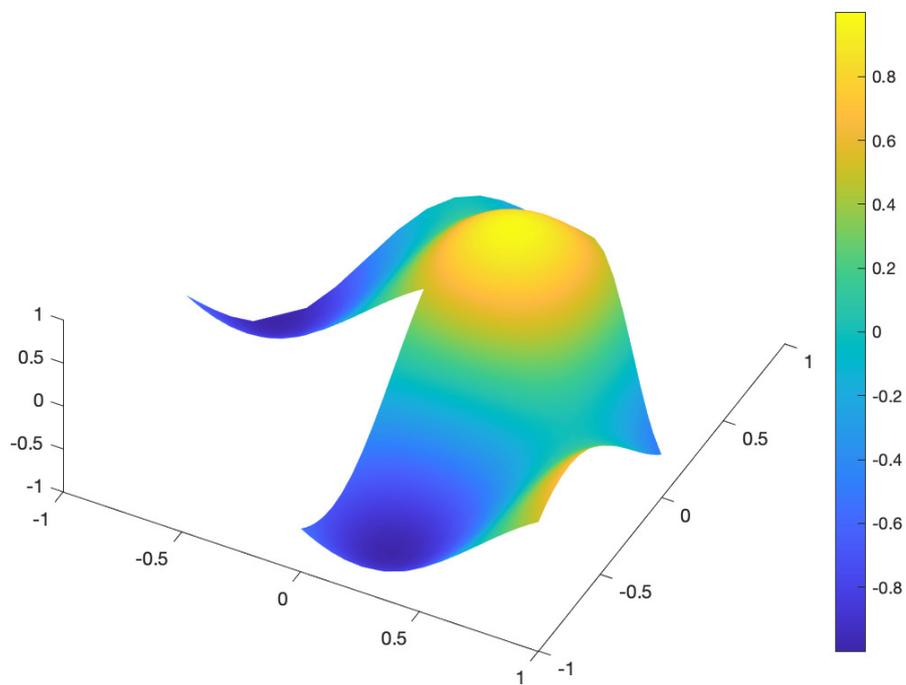


Abbildung 6.11: Numerische Lösung in  $\Omega_1$  mit (6.4) für  $N = 18$

## 6.3 Numerischer Test 3

In unserem dritten numerischen Test betrachten wir eine „Windmühle“, die aus folgenden Geometrien besteht  $\Omega_2 = R \cup T_O \cup T_U \cup T_L \cup T_R$ . Wir beginnen die Iterationen auf dem Dreieck  $R$  mit den Interfacegrenzen

$$\begin{aligned}\Gamma_1 &= (-0.5, 0.5) \times \{0.5\}, \\ \Gamma_2 &= \{0.5\} \times (-0.5, 0.5), \\ \Gamma_3 &= (-0.5, 0.5) \times \{-0.5\}, \\ \Gamma_4 &= \{-0.5\} \times (-0.5, 0.5), \\ \Gamma &= \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4,\end{aligned}$$

und lösen im Anschluss auf  $T_O, T_R, T_U$  und  $T_L$ . Wir betrachten die Lösung

$$u(x, y) = \sin\left(3x\pi + \frac{\pi}{4}\right) \sin\left(3\pi y + \frac{\pi}{4}\right) \quad (6.5)$$

und erhalten aufgrund von (6.1) und (6.2)

$$\begin{aligned}f &= -18\pi^2 \sin\left(3x\pi + \frac{\pi}{4}\right) \sin\left(3\pi y + \frac{\pi}{4}\right), \\ g &= \sin\left(3x\pi + \frac{\pi}{4}\right) \sin\left(3\pi y + \frac{\pi}{4}\right).\end{aligned}$$

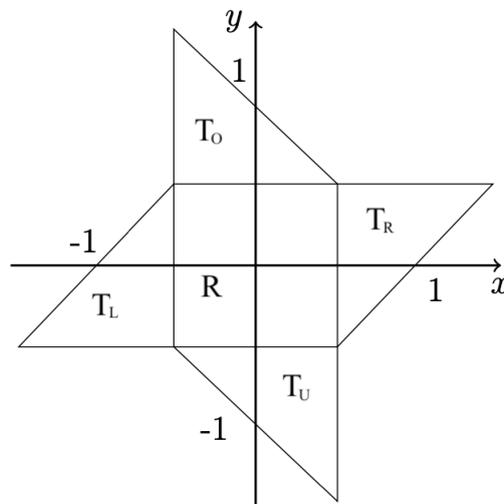


Abbildung 6.12: Gebiet  $\Omega_2$

Die Iteration auf dem Rechteck erfolgt

$$\begin{aligned}
 \Delta u_1^{n+1} &= f && \text{auf } R, \\
 u_1^{n+1} &= g && \text{auf } \partial R \setminus \Gamma, \\
 u_1^{n+1} &= (1 - \theta)u_1^n + \theta u_2^n && \text{auf } \Gamma_1, \\
 u_1^{n+1} &= (1 - \theta)u_1^n + \theta u_3^n && \text{auf } \Gamma_2, \\
 u_1^{n+1} &= (1 - \theta)u_1^n + \theta u_4^n && \text{auf } \Gamma_3, \\
 u_1^{n+1} &= (1 - \theta)u_1^n + \theta u_5^n && \text{auf } \Gamma_4,
 \end{aligned}$$

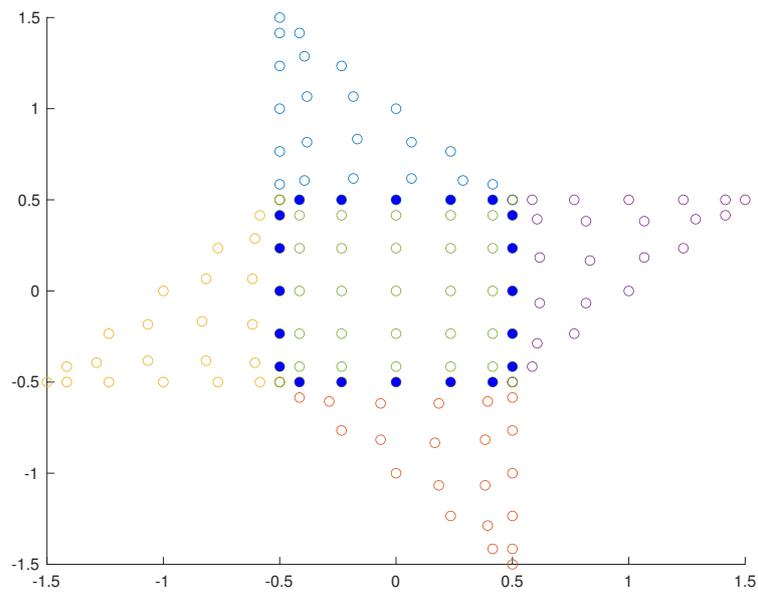
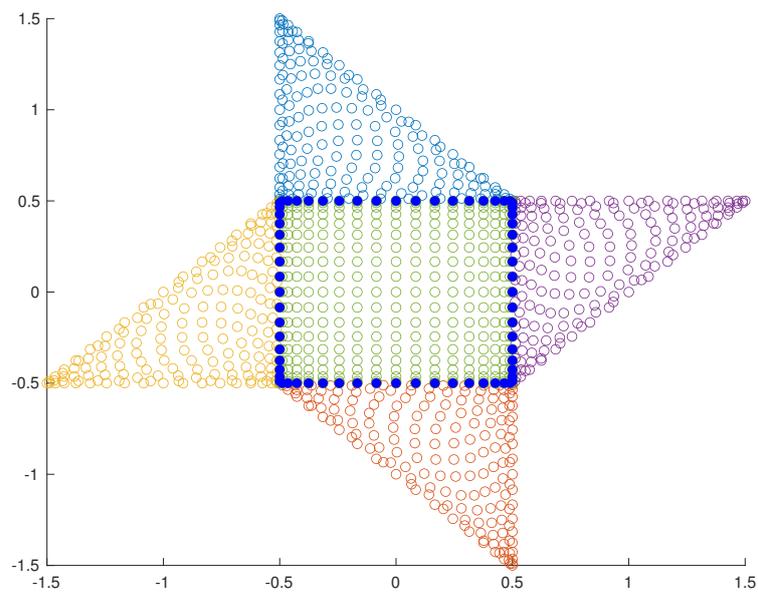
danach auf den Dreiecken

$$\begin{aligned}
 \Delta u_2^{n+1} &= f && \text{auf } T_O, \\
 u_2^{n+1} &= g && \text{auf } \partial T_O \setminus \Gamma_1, \\
 \frac{\partial u_2^{n+1}}{\partial y} &= \frac{\partial u_1^{n+1}}{\partial y} && \text{auf } \Gamma_1,
 \end{aligned}$$

$$\begin{aligned}
 \Delta u_3^{n+1} &= f && \text{auf } T_R, \\
 u_3^{n+1} &= g && \text{auf } \partial T_R \setminus \Gamma_2, \\
 \frac{\partial u_3^{n+1}}{\partial x} &= \frac{\partial u_1^{n+1}}{\partial x} && \text{auf } \Gamma_2,
 \end{aligned}$$

$$\begin{aligned}
 \Delta u_4^{n+1} &= f && \text{auf } T_U, \\
 u_4^{n+1} &= g && \text{auf } \partial T_U \setminus \Gamma_3, \\
 \frac{\partial u_4^{n+1}}{\partial y} &= \frac{\partial u_1^{n+1}}{\partial y} && \text{auf } \Gamma_3,
 \end{aligned}$$

$$\begin{aligned}
 \Delta u_5^{n+1} &= f && \text{auf } T_L, \\
 u_5^{n+1} &= g && \text{auf } \partial T_L \setminus \Gamma_4, \\
 \frac{\partial u_5^{n+1}}{\partial x} &= \frac{\partial u_1^{n+1}}{\partial x} && \text{auf } \Gamma_4.
 \end{aligned}$$

Abbildung 6.13: Kollokationspunkte  $N = 6$ Abbildung 6.14: Kollokationspunkte  $N = 18$ 

Wir wählen den Relaxationsparameter  $\theta = 0.26$  und brechen die Iteration ab, sobald der Fehler kleiner als  $1e^{-11}$  ist.

Das angewandte numerische Verfahren liefert die Ergebnisse:

N	Koll.punkte	It	$T_O$	$T_U$	$T_L$	$T_R$	R
6	161	83	1.07e-01	8.94e-02	6.91e-02	1.01e-01	1.10e-01
12	533	77	1.19e-03	7.15e-04	1.65e-03	1.41e-03	1.60e-03
18	1121	85	7.09e-07	7.60e-07	1.15e-06	1.00e-06	1.17e-06

Tabelle 6.3:  $\text{tol}=1e^{-11}$  und  $\theta = 0.26$

Unser Verfahren mit Fekete-Punkten benötigt nicht nur deutlich weniger Kollokationspunkte im Vergleich zum Verfahren mit CGL-Punkte und liefert für  $N = 18$  bessere Ergebnisse als in [HL01] für  $N = 16$ . Die Anzahl der Iterationen sind bei beiden Verfahren sehr ähnlich.

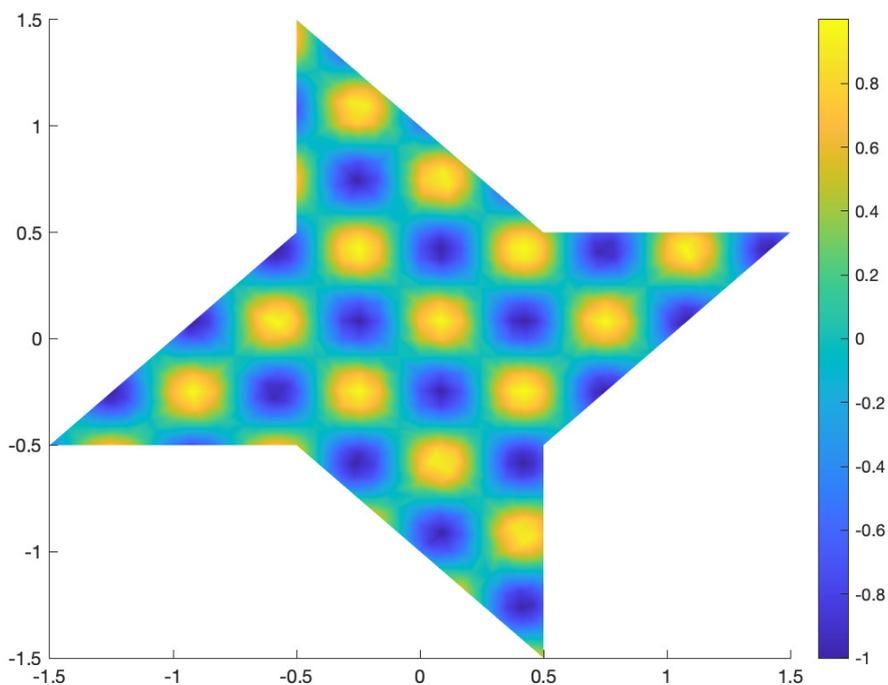
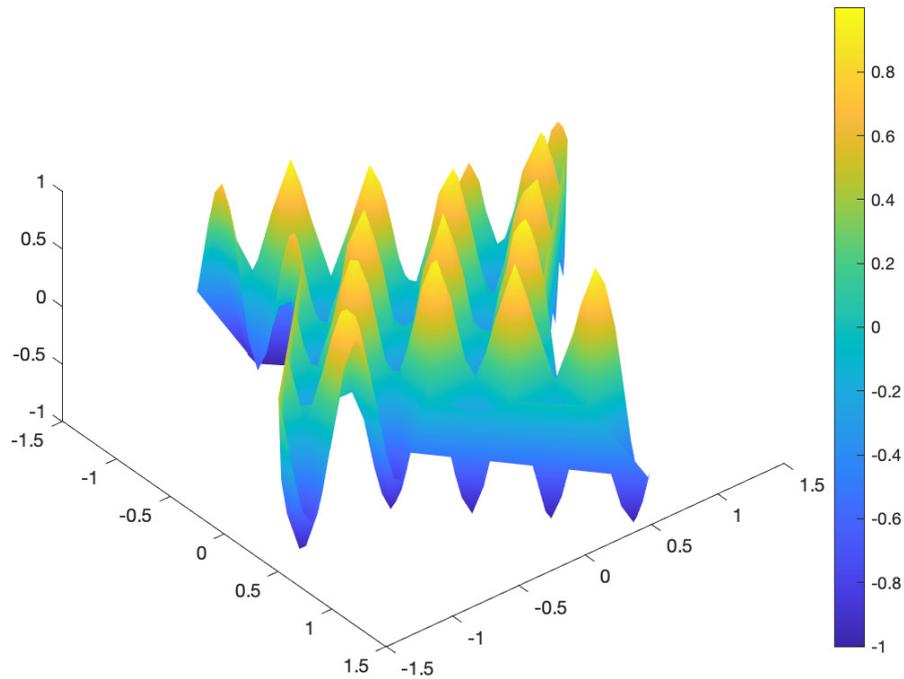
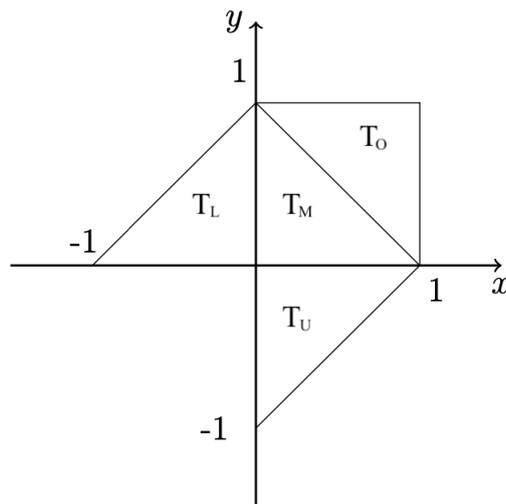


Abbildung 6.15: Numerische Lösung in  $\Omega_2$  mit (6.5) für  $N = 18$

Abbildung 6.16: Numerische Lösung in  $\Omega_2$  mit (6.5) für  $N = 18$ 

## 6.4 Numerischer Test 4

Abbildung 6.17: Gebiet  $\Omega_3$ 

Zum Schluss untersuchen wir ein Gebiet  $\Omega_3 = T_O \cup T_U \cup T_L \cup T_M$ , das nur aus Dreiecken besteht. Wir wählen (6.5) als Lösung. Wir nehmen den festen Relaxati-

onsparameter  $\theta = 0.38$  und iterieren, bis der Fehler auf allen Interfaces kleiner als  $1e^{-12}$  ist. Da wir nur Dreiecke haben, verwenden wir hier nur Fekete-Punkte. Wir lösen zuerst auf  $T_M$  mit den Interfaces

$$\begin{aligned}\Gamma_1 &= \{0\} \times (0, 1), \\ \Gamma_2 &= \{(x, y) \in \mathbb{R}^2 \mid x, y > 0 \text{ und } x + y = 1\}, \\ \Gamma_3 &= (0, 1) \times \{0\}, \\ \Gamma &= \Gamma_1 \cup \Gamma_2 \cup \Gamma_3\end{aligned}$$

und im Anschluss auf  $T_O, T_U$  und  $T_L$ . Wir betrachten die Lösung

$$u(x, y) = \sin\left(3x\pi + \frac{\pi}{4}\right) \sin\left(3\pi y + \frac{\pi}{4}\right)$$

und erhalten aufgrund von (6.1) und (6.2) wie im vorherigen Test

$$\begin{aligned}f &= -18\pi^2 \sin\left(3x\pi + \frac{\pi}{4}\right) \sin\left(3\pi y + \frac{\pi}{4}\right), \\ g &= \sin\left(3x\pi + \frac{\pi}{4}\right) \sin\left(3\pi y + \frac{\pi}{4}\right).\end{aligned}$$

Die Iteration auf  $T_M$  erfolgt

$$\begin{aligned}\Delta u_1^{n+1} &= f && \text{auf } T_M, \\ u_1^{n+1} &= g && \text{auf } \partial T_M \setminus \Gamma, \\ u_1^{n+1} &= (1 - \theta)u_1^n + \theta u_2^n && \text{auf } \Gamma_1, \\ u_1^{n+1} &= (1 - \theta)u_1^n + \theta u_3^n && \text{auf } \Gamma_2, \\ u_1^{n+1} &= (1 - \theta)u_1^n + \theta u_4^n && \text{auf } \Gamma_3,\end{aligned}$$

danach auf den verbleibenden Dreiecken

$$\begin{aligned}\Delta u_2^{n+1} &= f && \text{auf } T_L, \\ u_2^{n+1} &= g && \text{auf } \partial T_L \setminus \Gamma_1, \\ \frac{\partial u_2^{n+1}}{\partial x} &= \frac{\partial u_1^{n+1}}{\partial x} && \text{auf } \Gamma_1,\end{aligned}$$

$$\begin{aligned}\Delta u_3^{n+1} &= f && \text{auf } T_O, \\ u_3^{n+1} &= g && \text{auf } \partial T_O \setminus \Gamma_2, \\ \frac{\partial u_3^{n+1}}{\partial n} &= \frac{\partial u_1^{n+1}}{\partial n} && \text{auf } \Gamma_2,\end{aligned}$$

$$\begin{aligned}\Delta u_4^{n+1} &= f && \text{auf } T_U, \\ u_4^{n+1} &= g && \text{auf } \partial T_U \setminus \Gamma_3, \\ \frac{\partial u_4^{n+1}}{\partial y} &= \frac{\partial u_1^{n+1}}{\partial y} && \text{auf } \Gamma_3,\end{aligned}$$

wobei  $\partial \cdot / \partial n$  die Ableitung in Normalenrichtung ist und  $n$  den Normalenvektor von  $T_M$  darstellt.

Auf den einzelnen Dreiecken liefert die Dirichlet-Neumann Iteration mit Feketen-Punkten die Fehler:

N	Koll.punkte	It	$T_O$	$T_U$	$T_L$	$T_M$
6	112	60	3.27e-01	1.21e-01	1.21e-01	3.93e-01
12	364	53	4.37e-03	3.27e-03	3.27e-03	7.23e-03
18	760	70	1.33e-06	1.89e-06	1.89e-06	3.22e-06

Tabelle 6.4: tol=1e-12 und  $\theta = 0.38$

In [HL01] wurden für  $N = 8$  genau 324 und  $N = 16$  insgesamt 1156 Kollokationspunkte zur Berechnung verwendet. Zum Vergleich haben wir nur 112 Punkte für  $N = 6$  und 760 für  $N = 18$  verwendet. In diesem Beispiel benötigen die Fekete-Punkte, trotz leicht mehr Iterationen, auf den Dreiecken deutlich weniger Rechenaufwand als die verwendeten CGL-Punkten in [HL01]. Die Fehler auf den Gebieten sind in unserem Verfahren ein wenig besser.

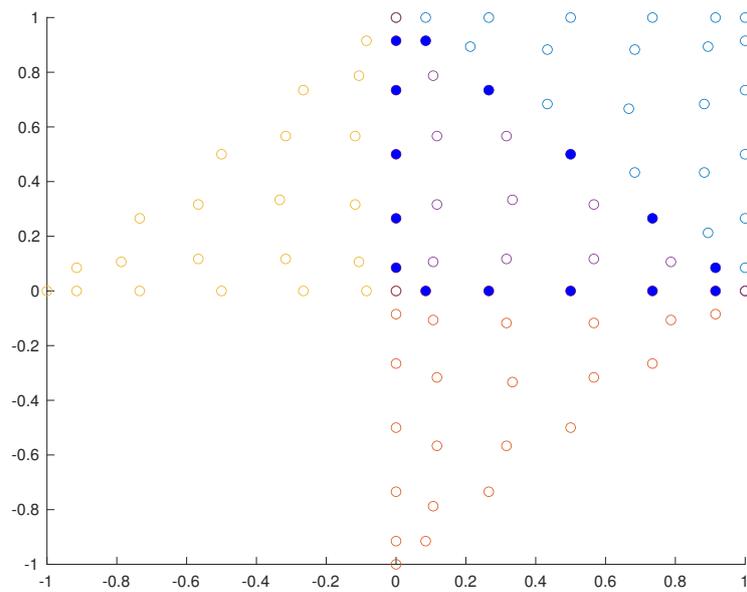


Abbildung 6.18: Kollokationspunkte  $N = 6$

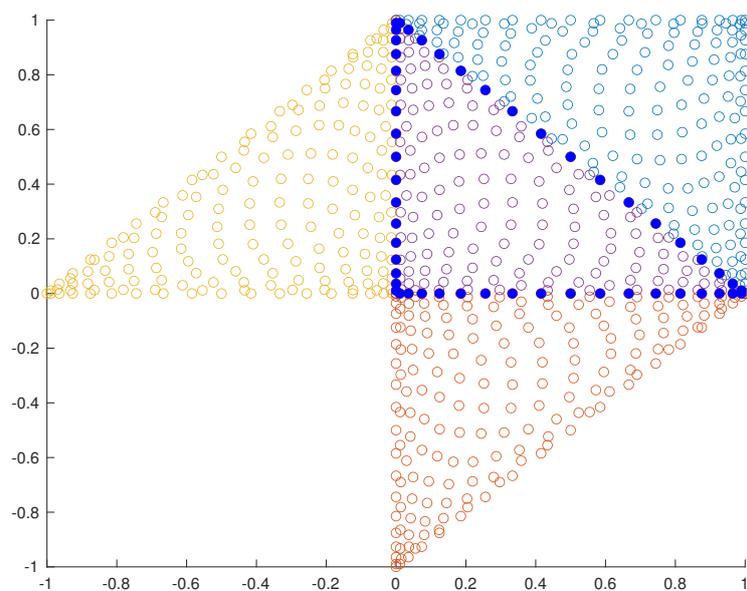
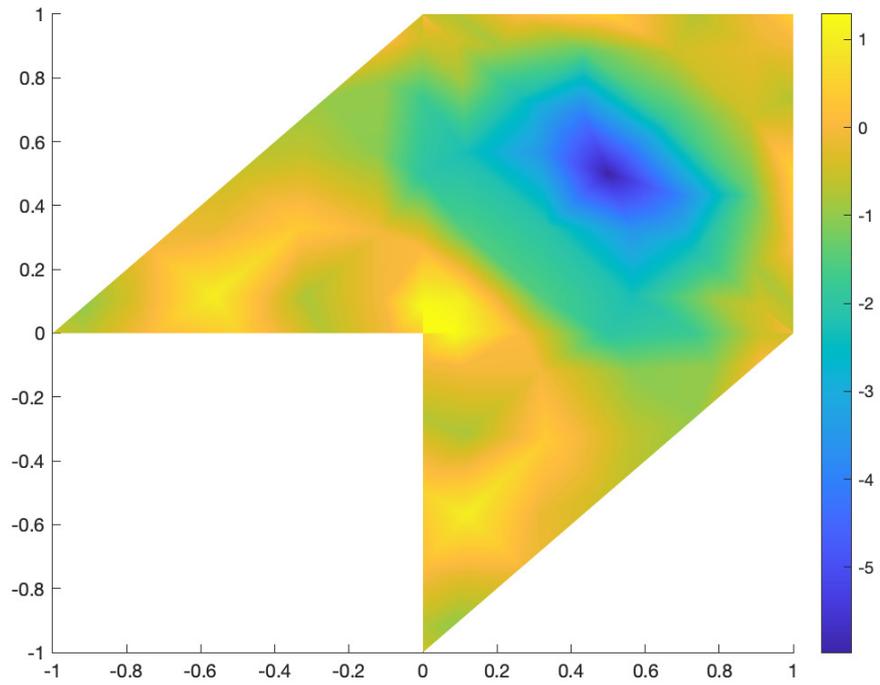
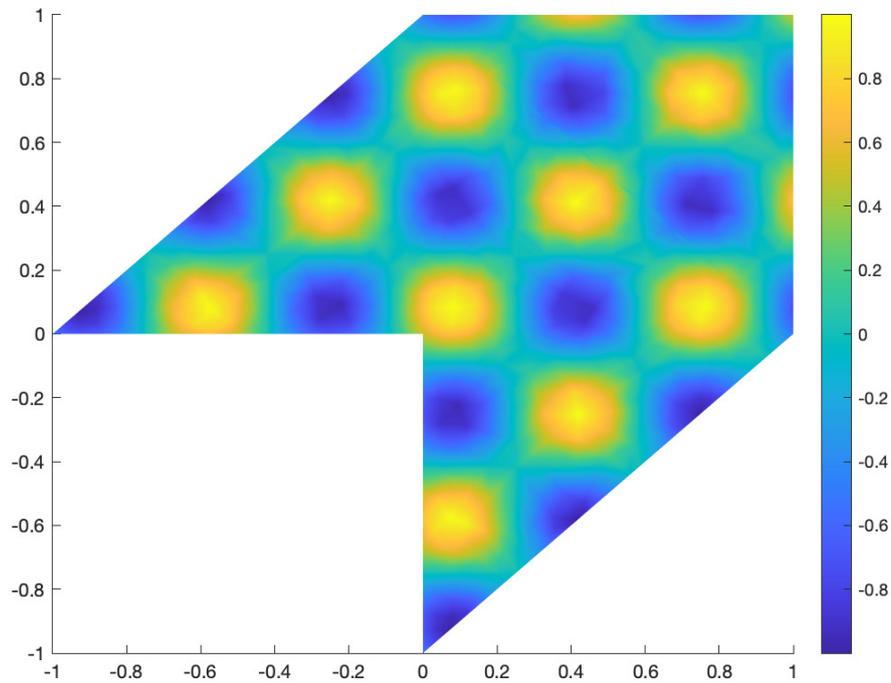


Abbildung 6.19: Kollokationspunkte  $N = 18$

Abbildung 6.20: Numerische Lösung in  $\Omega_3$  mit (6.5) für  $N = 6$ Abbildung 6.21: Numerische Lösung in  $\Omega_3$  mit (6.5) für  $N = 18$

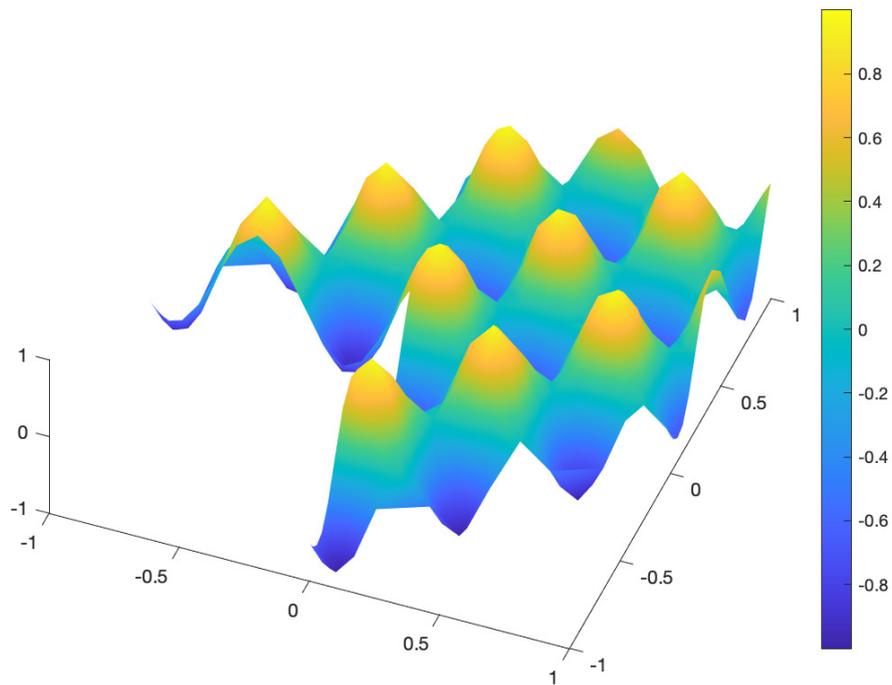


Abbildung 6.22: Numerische Lösung in  $\Omega_3$  mit (6.5) für  $N = 18$

# Kapitel 7

## Numerische Ergebnisse für die Neumann-Neumann Iteration

Hauptbestandteil unserer Untersuchung ist die Effektivität der Neumann-Neumann Iteration unter Berücksichtigung von Fekete-Punkten im Bezug auf die klassische Poisson-Gleichung und komplexere Konvektion-Diffusionsgleichungen. Dabei gliedert sich das Kapitel in drei Teile.

Im ersten Teil werden die numerischen Tests des vorigen Kapitels aufgegriffen und unter Verwendung der Neumann-Neumann Iteration in Kombination mit Fekete-Punkten gelöst. Mit den gewonnenen Ergebnissen können wir einen Vergleich zwischen der Dirichlet-Neumann Iteration und Neumann-Neumann Iteration herstellen. Im zweiten Teil untersuchen wir die Möglichkeiten der globalen  $h$ - und  $p$ -Verfeinerung im Bezug auf die Neumann-Neumann Iteration, anhand von einfachen Poisson-Problemen, aber auch komplexeren Konvektion-Diffusionsgleichungen. Wird die Elementgröße  $h$  verringert, während die Polynomordnung unverändert bleibt, wird dies als  $h$ -Verfeinerung bezeichnet. In der Regel wird der Fehler durch eine Verfeinerung von  $h$  kleiner, wobei dieser zusätzlich von der Glattheit der Lösung und in  $h$  exponentiell von der Polynomordnung abhängig ist. Wird die Genauigkeit durch Erhöhen der Polynomordnung  $p$  verbessert, wird dies  $p$ -Verfeinerung genannt. Diese wird üblich mit spektraler Konvergenz verbunden. Spektral impliziert, dass der numerische Fehler schneller abnimmt als jede Potenz von  $1/p$ , wobei  $p$  die Ordnung der Polynomentwicklung ist. Werden beide Verfeinerungsstrategien kombiniert, erhalten wir die  $hp$ -Verfeinerung [Poz].

Im letzten Teil behandeln wir komplexere Konvektion-Diffusionsgleichungen unter dem Gebrauch der Neumann-Neumann Iteration und der adaptiven  $p$ -Verfeinerung nur an den Stellen, wo der Fehler am größten geschätzt wird.

## 7.1 Vergleich von Dirichlet-Neumann und Neumann-Neumann Iteration

In diesem Abschnitt werden die Problemstellungen des zurückliegenden Kapitels aufgegriffen und der durch die Neumann-Neumann Iteration resultierende Algorithmus angegeben. Das jeweilige Problem wird mit Hilfe der Neumann-Neumann Iteration unter Verwendung von Fekete-Punkten für verschiedene Polynomgrade gelöst. Dadurch stellen wir eine Vergleichbarkeit zum vorigen Kapitel, der Dirichlet-Neumann Iteration, her und können quantitative Aussagen über die Verwendung beider Algorithmen treffen.

### 7.1.1 Numerischer Test 1

Unter Berücksichtigung der Problemstellung in Kapitel 6, dem vorgegebenen Gebiet  $\Omega = R \cup T$  in Abschnitt 6.1 und denselben Fekete-Punkten in Abhängigkeit vom verwendeten Polynomgrad, exemplarisch skizziert in Abbildung 6.2 und 6.3, wenden wir den Neumann-Neumann Algorithmus aus Abschnitt 2.2.2 an.

Als Startwerte der Neumann-Neumann Iteration verwenden wir  $u_\Gamma^0 = \mathbf{0}$  und den Parameter  $\theta = 0.1$ .

Die Iteration im  $n$ -ten Schritt lautet wie folgt:

1. Lösen auf jedem Teilgebiet ein Dirichlet-Problem:

$$\begin{aligned} \Delta u_1^{n+1} &= f & \text{auf } R & & \Delta u_2^{n+1} &= f & \text{auf } T \\ u_1^{n+1} &= g & \text{auf } \partial R \setminus \Gamma & \text{ und } & u_2^{n+1} &= g & \text{auf } \partial T \setminus \Gamma \\ u_1^{n+1} &= u_\Gamma^n & \text{auf } \Gamma & & u_2^{n+1} &= u_\Gamma^n & \text{auf } \Gamma \end{aligned} .$$

2. Löse auf jedem Teilgebiet, hier  $T$  und  $R$ , ein Neumann-Problem:

$$\begin{aligned} \Delta \psi_1^{n+1} &= 0 & \text{auf } R & & \Delta \psi_2^{n+1} &= 0 & \text{auf } T \\ \psi_1^{n+1} &= 0 & \text{auf } \partial R \setminus \Gamma & \text{ und } & \psi_2^{n+1} &= 0 & \text{auf } \partial T \setminus \Gamma \\ \frac{\partial \psi_1^{n+1}}{\partial n_1} &= \frac{\partial u_1^{n+1}}{\partial n_1} + \frac{\partial u_2^{n+1}}{\partial n_2} & \text{auf } \Gamma & & \frac{\partial \psi_2^{n+1}}{\partial n_2} &= \frac{\partial u_1^{n+1}}{\partial n_1} + \frac{\partial u_2^{n+1}}{\partial n_2} & \text{auf } \Gamma \end{aligned} .$$

3. Prüfe das Abbruchkriterium:

$$\|(\psi_1^{n+1} + \psi_2^{n+1})|_\Gamma\|_{L_1(\Gamma)} \leq \varepsilon_\Gamma.$$

Falls das Abbruchkriterium erfüllt ist, wird die Iteration beendet, andernfalls wird weiter iteriert.

4. Update  $u_\Gamma^{n+1}$ :

$$u_\Gamma^{n+1} = u_\Gamma^n - \theta(\psi_1^{n+1} + \psi_2^{n+1}) \text{ auf } \Gamma .$$

Wir setzen  $n \rightarrow n + 1$  und gehen zu Schritt 1.

Unter Verwendung der obengenannten Iteration resultieren folgende Ergebnisse:

$\varepsilon_\Gamma$	N	Kollokationspunkte	It	Fehler $T$	Fehler $R$
$1e^{-6}$	6	77	30	4.42e-04	2.47e-04
	12	260	30	3.17e-09	1.75e-09
	18	551	31	1.65e-09	9.62e-10
$1e^{-8}$	6	77	39	4.42e-04	2.47e-04
	12	260	39	2.82e-09	1.67e-09
	18	551	40	1.58e-11	9.18e-12
$1e^{-10}$	6	77	48	4.42e-04	2.47e-04
	12	260	48	2.84e-09	1.68e-09
	18	551	49	1.51e-13	8.78e-14
$1e^{-12}$	6	77	57	4.42e-04	2.47e-04
	12	260	57	2.84e-09	1.68e-09
	18	551	62	2.75e-15	2.16e-15

Der Parameter  $\varepsilon_\Gamma$  nimmt direkten Einfluss auf die Abbruchbedingung der Neumann-Neumann Iteration. Wir bemerken, dass ein niedriges  $\varepsilon_\Gamma$  nur dann nötig ist, falls auch der zu erwartende globale Fehler auf  $T$  und  $R$  von der gleichen Größenordnung ist. Es ist nicht nötig, ein besonders kleines  $\varepsilon_\Gamma$  zu wählen, um moderate Ergebnisse zu erhalten. Wir beobachten, dass  $\varepsilon_\Gamma$  eine obere Grenze für den Fehler auf den Teilgebieten darstellt und somit klein genug gewählt werden muss, falls genauere Lösungen erforderlich sind.

Außerdem erkennen wir, dass die Anzahl der Iterationen über  $N$  fast konstant sind und sich zwischen  $N = 6$  und  $N = 18$  nur um wenige Iterationen unterscheidet, wobei der Fehler auf den Teilgebieten doch stark unterschiedlich ist und mit  $N = 18$  erheblich bessere Ergebnisse erzielt werden können.

Insgesamt verzeichnen wir einen sehr schönen exponentiellen Abfall des Fehlers in Abhängigkeit des Polynomgrades, wie wir es von Spektralmethoden, wie dem Kollokationsverfahren, erwarten würden.

Vergleichen wir die Werte in diesem Abschnitt mit den Ergebnissen in 6.1, entnehmen wir, dass gleiche Größenordnungen für die Fehler in den Teilgebieten erreicht

werden konnten. Die Dirichlet-Neumann Iteration benötigt dabei nur ungefähr die Hälfte der Iterationen, um das Abbruchkriterium der Dirichlet-Neumann Iteration zu erfüllen. Dies liegt an den verschiedenen Abbruchkriterien der Dirichlet-Neumann Iteration und Neumann-Neumann Iteration und die Anzahl der Iterationsschritte in Abhängigkeit von den gewählten Relaxationsparametern  $\theta$ . Eine Optimierung des Parameters für die Neumann-Neumann Iteration kann die Anzahl der Iterationsschritte bei gleichwertigen Ergebnissen verringern. Zum Schluss stellen wir fest, dass beide Methoden Lösungen mit der gleichen Fehlergröße produzieren und in der Anzahl der Iterationen zwar von ca. 30 Iterationen zu ca. 60 Iterationen auseinanderliegen, jedoch dieser Unterschied auf dem benutzten Laptop zur Lösung kaum Unterschiede in der Rechenzeit darstellt.

Vergleichen wir die in dieser Arbeit gewonnenen Werte mit den Ergebnissen von [HL01], so stellen wir fest, dass durch die dynamische Bestimmung des Relaxationsparameter in [HL01] eine geringere Anzahl von Iterationen zum Abbruch nötig waren (ca. 17), dabei aber ein leicht schlechteres Ergebnis folgt. Der Mehraufwand der Berechnung des dynamischen Parameter führt auch zu zusätzlichen Kosten, was wir hier nicht berücksichtigen können. Desweiteren wurde in [HL01] auch der Fall  $N = 32$  betrachtet, welcher aber leicht schlechtere Ergebnisse liefert, als zu dem hier untersuchten Fall für  $N = 18$ .

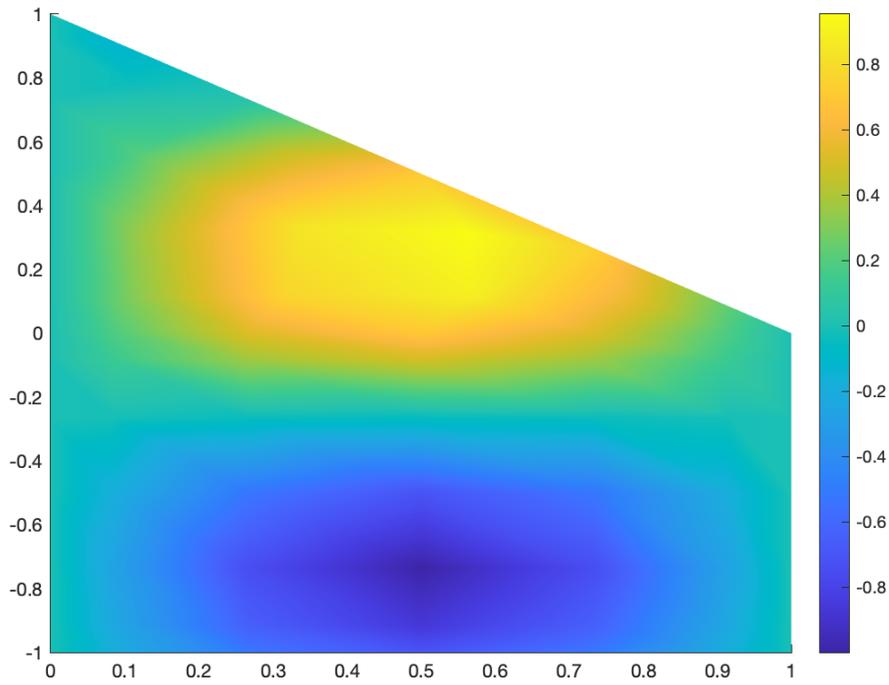


Abbildung 7.1: Lösung:  $\varepsilon_\Gamma = 1e^{-12}$ ,  $N = 6$

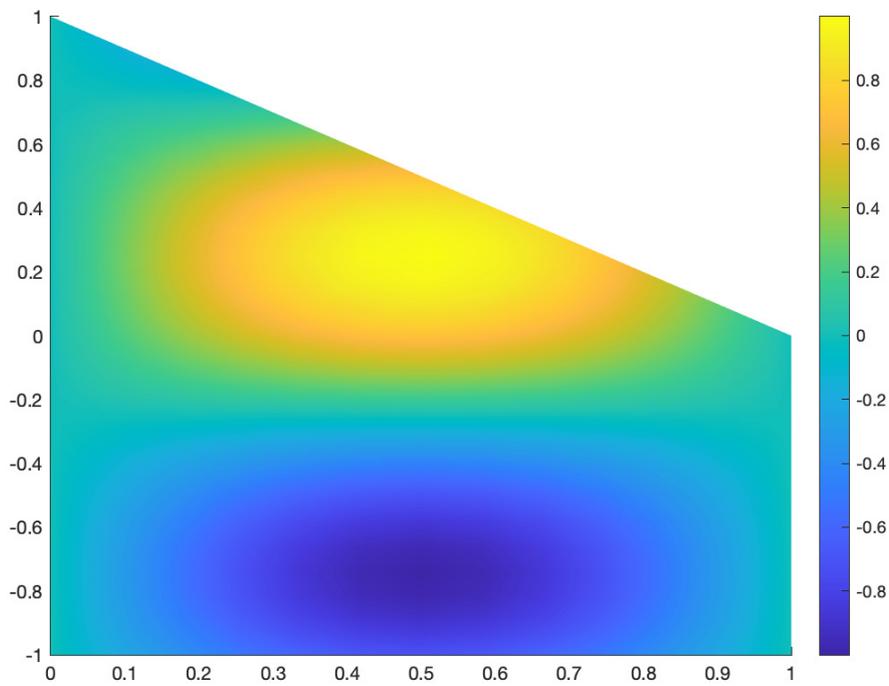


Abbildung 7.2: Lösung:  $\varepsilon_\Gamma = 1e^{-12}$ ,  $N = 18$

## 7.1.2 Numerischer Test 2

In diesem numerischen Test betrachten wir das Gebiet  $\Omega_1 = R \cup T_L \cup T_R$  aus dem Abschnitt 6.2 und verwenden für den Neumann-Neumann Algorithmus dieselben Fekete-Punkte, exemplarisch skizziert in Abbildung 6.7 und 6.9.

Als Startwerte der Neumann-Neumann Iteration verwenden wir  $u_\Gamma^0 = \mathbf{0}$  und den Parameter  $\theta = 0.1$ .

Die Iteration im  $n$ -ten Schritt lautet wie folgt:

1. Lösen auf jedem Teilgebiet ein Dirichlet-Problem:

$$\begin{aligned} \Delta u_1^{n+1} &= f & \text{auf } R & & \Delta u_2^{n+1} &= f & \text{auf } T_R \\ u_1^{n+1} &= g & \text{auf } \partial R \setminus \Gamma_1 & & u_2^{n+1} &= g & \text{auf } \partial T_R \setminus \{\Gamma_1 \cup \Gamma_2\} \\ u_1^{n+1} &= u_{\Gamma_1}^n & \text{auf } \Gamma_1 & & u_2^{n+1} &= u_{\Gamma_1}^n & \text{auf } \Gamma_1 \\ & & & & u_2^{n+1} &= u_{\Gamma_2}^n & \text{auf } \Gamma_2 \end{aligned} ,$$

$$\text{und} \quad \begin{aligned} \Delta u_3^{n+1} &= f & \text{auf } T_L \\ u_3^{n+1} &= g & \text{auf } \partial T_L \setminus \Gamma_2 \\ u_3^{n+1} &= u_{\Gamma_2}^n & \text{auf } \Gamma_2 \end{aligned} .$$

2. Löse auf jedem Teilgebiet, hier  $T_L$ ,  $T_R$  und  $R$ , ein Neumann-Problem:

$$\begin{aligned} \Delta \psi_1^{n+1} &= 0 & \text{auf } R & & \Delta \psi_2^{n+1} &= 0 & \text{auf } T_R \\ \psi_1^{n+1} &= 0 & \text{auf } \partial R \setminus \Gamma_1 & & \psi_2^{n+1} &= 0 & \text{auf } \partial T_R \setminus \{\Gamma_1 \cup \Gamma_2\} \\ \frac{\partial \psi_1^{n+1}}{\partial y} &= \frac{\partial u_1^{n+1}}{\partial y} - \frac{\partial u_2^{n+1}}{\partial y} & \text{auf } \Gamma_1 & & \frac{\partial \psi_2^{n+1}}{\partial y} &= \frac{\partial u_1^{n+1}}{\partial y} - \frac{\partial u_2^{n+1}}{\partial y} & \text{auf } \Gamma_1 \\ & & & & \frac{\partial \psi_2^{n+1}}{\partial x} &= \frac{\partial u_1^{n+1}}{\partial x} - \frac{\partial u_3^{n+1}}{\partial x} & \text{auf } \Gamma_2 \end{aligned} ,$$

$$\text{und} \quad \begin{aligned} \Delta \psi_3^{n+1} &= 0 & \text{auf } T_L \\ \psi_3^{n+1} &= 0 & \text{auf } \partial T_L \setminus \Gamma_2 \\ \frac{\partial \psi_3^{n+1}}{\partial x} &= \frac{\partial u_3^{n+1}}{\partial x} - \frac{\partial u_2^{n+1}}{\partial x} & \text{auf } \Gamma_2 \end{aligned} .$$

3. Prüfe das Abbruchkriterium:

$$\begin{aligned} \|(\psi_1^{n+1} + \psi_2^{n+1})|_{\Gamma_1}\|_{L_1(\Gamma_1)} &\leq \varepsilon_\Gamma, \\ \|(\psi_2^{n+1} + \psi_3^{n+1})|_{\Gamma_2}\|_{L_1(\Gamma_2)} &\leq \varepsilon_\Gamma. \end{aligned}$$

Falls das Abbruchkriterium erfüllt ist, wird die Iteration beendet, andernfalls wird weiter iteriert.

4. Update  $u_{\Gamma}^{n+1}$ :

$$\begin{aligned} u_{\Gamma_1}^{n+1} &= u_{\Gamma_1}^n - \theta(\psi_1^{n+1} + \psi_2^{n+1}) && \text{auf } \Gamma_1, \\ u_{\Gamma_2}^{n+1} &= u_{\Gamma_2}^n - \theta(\psi_2^{n+1} + \psi_3^{n+1}) && \text{auf } \Gamma_2. \end{aligned}$$

Wir setzen  $n \rightarrow n + 1$  und gehen zu Schritt 1.

Unter Verwendung der angegebenen Iteration führen wir folgende Ergebnisse auf:

$\varepsilon_{\Gamma}$	N	Kollokationspunkte	It	Fehler $T_R$	Fehler $T_L$	Fehler $R$
$1e^{-6}$	6	105	29	5.64e-04	4.75e-04	1.67e-04
	12	351	30	4.79e-09	4.51e-09	9.49e-10
	18	741	32	1.77e-09	1.49e-09	5.22e-10
$1e^{-8}$	6	105	38	5.64e-04	4.75e-04	1.67e-04
	12	351	40	3.23e-09	3.28e-09	3.55e-10
	18	741	41	2.13e-11	1.85e-11	6.09e-12
$1e^{-10}$	6	105	47	5.64e-04	4.75e-04	1.67e-04
	12	351	49	3.24e-09	3.28e-09	3.59e-10
	18	741	51	1.68e-13	1.54e-13	4.91e-14
$1e^{-12}$	6	105	57	5.64e-04	4.75e-04	1.67e-04
	12	351	57	3.24e-09	3.28e-09	3.59e-10
	18	741	2600	7.65e-15	5.28e-15	1.04e-14

Wie im vorigen numerischen Test bildet der Parameter  $\varepsilon_{\Gamma}$ , wie zu erwarten, eine gewisse obere Schranke bezüglich des globalen Fehlers auf den Abschnitten. Unter Verwendung eines hohen Polynomgrades sollte  $\varepsilon_{\Gamma}$  klein genug sein, um die ganze Möglichkeit der Konvergenzgüte auszuschöpfen, siehe  $N = 18$ , welche im Fall von  $\varepsilon_{\Gamma} = 1e^{-6}$  eine Fehlergenauigkeit in der verwendeten Norm von  $\approx 1e^{-9}$  aufweist und im Fall von  $\varepsilon_{\Gamma} = 1e^{-12}$  annähernd Maschinengenauigkeit mit  $\approx 1e^{-15}$  erreicht. Bei kleinem Polynomgrad genügt auch ein geringes Abbruchkriterium. Die Anzahl der Iterationen ist fast unabhängig von  $N$  und zeigt eine Abhängigkeit von  $\varepsilon_{\Gamma}$ . Im letzten untersuchten Fall  $N = 18$  und  $\varepsilon = 1e^{-12}$  sehen wir eine stark steigende Zahl an Iterationen (2600), bei näherer Betrachtung iteriert hier der Fehler auf den Interfaces zwischen  $1e^{-10}$  und  $1e^{-11}$  bis schließlich die Bedingung erfüllt werden konnte. Ein möglicher Grund ist, dass die Lösung annähernd Maschinengenauigkeit erreicht und es hier zu Rundungsfehlern kommen kann. Letztlich stellen wir auch in diesem Beispiel eine exponentielle Konvergenz in Abhängigkeit von  $N$  fest.

Vergleichen wir die erreichten Werte für  $\varepsilon = 1e^{-12}$  mit den Werten in Kapitel 6.2,

erkennen wir kaum Unterschiede der Methoden, wenn wir die unüblich hohe Anzahl an Iterationen im Fall  $N = 18$  vernachlässigen. Ebenso stellen wir die Ergebnisse mit den Werten in [HL01] gegenüber. In [HL01] wurde das Problem mit festen Relaxationsparameter betrachtet und wir sehen, dass in diesem Fall die betrachteten Fälle  $N = 16$  und  $N = 18$  Fehler auf den Gebieten in der Größenordnung von  $10^{-13}$  bei 46 und 87 Iterationen für die jeweiligen Polynomgrade produzieren. In diesem Fall stellen wir ähnliche Ergebnisse für  $\varepsilon = 1e^{-10}$  fest und im Fall  $\varepsilon = 1e^{-12}$  erreicht sogar die Konvergenz eine höhere Größenordnung ( $\approx 10^{-15}$ ).

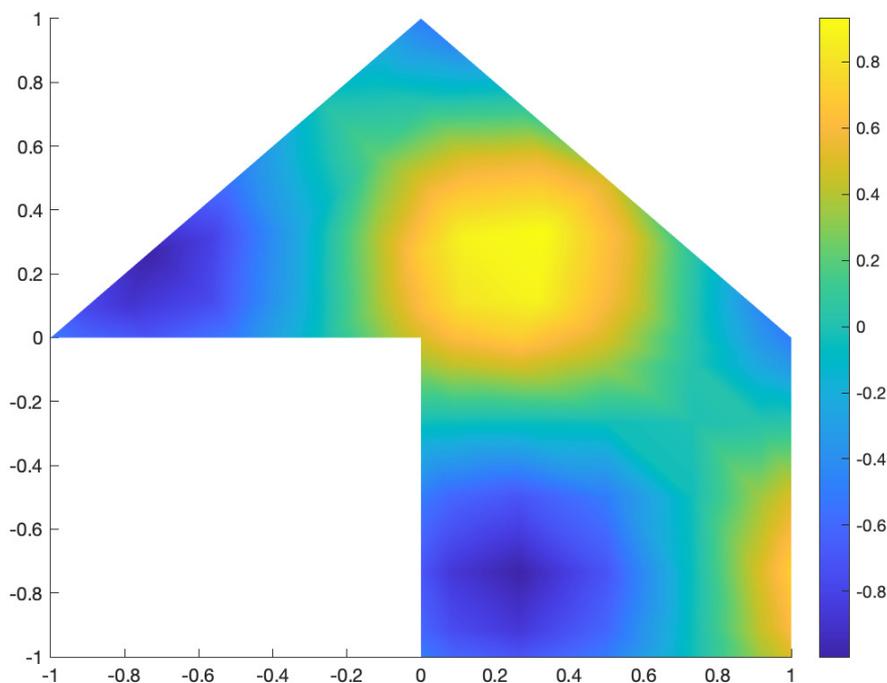


Abbildung 7.3: Lösung:  $\varepsilon_{\Gamma} = 1e^{-12}$ ,  $N = 6$

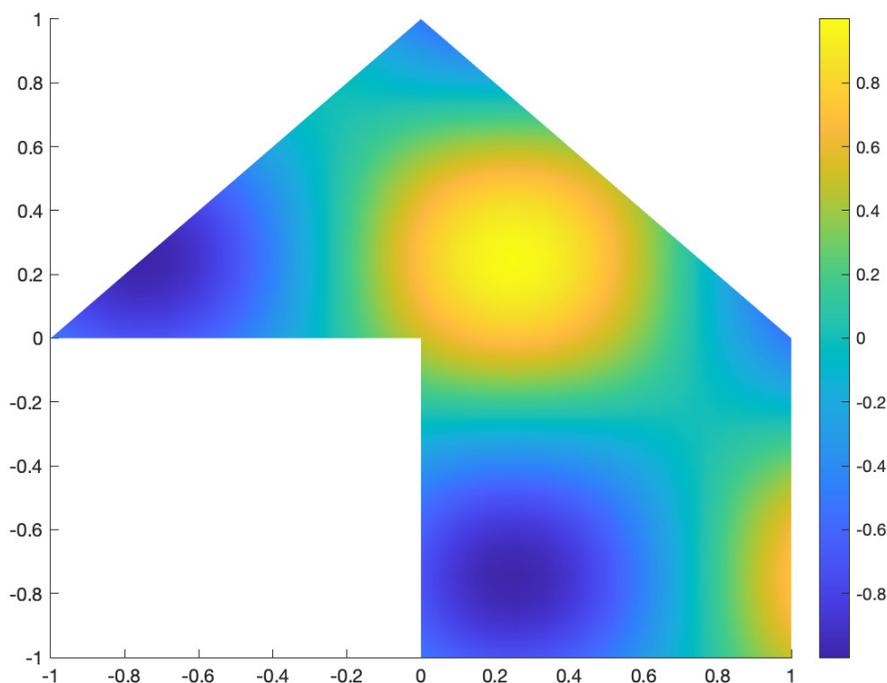


Abbildung 7.4: Lösung:  $\varepsilon_\Gamma = 1e^{-12}$ ,  $N = 18$

### 7.1.3 Numerischer Test 3

Wir lösen das Problem auf dem Gebiet  $\Omega_2 = R \cup T_O \cup T_U \cup T_L \cup T_R$  aus dem Abschnitt 6.3. Dabei verwenden wir dieselben Fekete-Punkte, skizziert in Abbildung 6.13 und 6.14.

Als Startwerte der Neumann-Neumann Iteration verwenden wir  $u_\Gamma^0 = \mathbf{0}$  und den Parameter  $\theta = 0.1$ .

Die Iteration im  $n$ -ten Schritt lautet wie folgt:

1. Lösen auf jedem Teilgebiet ein Dirichlet-Problem:

$$\begin{aligned}
 \Delta u_1^{n+1} &= f && \text{auf } R \\
 u_1^{n+1} &= g && \text{auf } \partial R \setminus \{\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4\} \\
 u_1^{n+1} &= u_{\Gamma_1}^n && \text{auf } \Gamma_1 \\
 u_1^{n+1} &= u_{\Gamma_2}^n && \text{auf } \Gamma_2 \\
 u_1^{n+1} &= u_{\Gamma_3}^n && \text{auf } \Gamma_3 \\
 u_1^{n+1} &= u_{\Gamma_4}^n && \text{auf } \Gamma_4
 \end{aligned}
 \quad \text{und}$$

$$\begin{aligned}
 \Delta u_2^{n+1} &= f && \text{auf } T_O \\
 u_2^{n+1} &= g && \text{auf } \partial T_O \setminus \Gamma_1 \\
 u_2^{n+1} &= u_{\Gamma_1}^n && \text{auf } \Gamma_1 \\
 \Delta u_3^{n+1} &= f && \text{auf } T_R \\
 u_3^{n+1} &= g && \text{auf } \partial T_R \setminus \Gamma_2 \\
 u_3^{n+1} &= u_{\Gamma_2}^n && \text{auf } \Gamma_2
 \end{aligned}
 ,$$

$$\begin{aligned} \Delta u_4^{n+1} &= f & \text{auf } T_U & , & \Delta u_5^{n+1} &= f & \text{auf } T_L \\ u_4^{n+1} &= g & \text{auf } \partial T_U \setminus \Gamma_3 & , & u_5^{n+1} &= g & \text{auf } \partial T_L \setminus \Gamma_4 \\ u_4^{n+1} &= u_{\Gamma_3}^n & \text{auf } \Gamma_3 & , & u_5^{n+1} &= u_{\Gamma_4}^n & \text{auf } \Gamma_4 \end{aligned}$$

2. Löse auf jedem Teilgebiet ein Neumann-Problem:

$$\begin{aligned} \Delta \psi_1^{n+1} &= 0 & \text{auf } R \\ \psi_1^{n+1} &= 0 & \text{auf } \partial R \setminus \{\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4\} \\ \frac{\partial \psi_1^{n+1}}{\partial y} &= \frac{\partial u_1^{n+1}}{\partial y} - \frac{\partial u_2^{n+1}}{\partial y} & \text{auf } \Gamma_1 \\ \frac{\partial \psi_1^{n+1}}{\partial x} &= \frac{\partial u_1^{n+1}}{\partial x} - \frac{\partial u_3^{n+1}}{\partial x} & \text{auf } \Gamma_2 \\ \frac{\partial \psi_1^{n+1}}{\partial y} &= \frac{\partial u_1^{n+1}}{\partial y} - \frac{\partial u_4^{n+1}}{\partial y} & \text{auf } \Gamma_3 \\ \frac{\partial \psi_1^{n+1}}{\partial x} &= \frac{\partial u_1^{n+1}}{\partial x} - \frac{\partial u_5^{n+1}}{\partial x} & \text{auf } \Gamma_4 \end{aligned} \quad \text{und}$$

$$\begin{aligned} \Delta \psi_2^{n+1} &= 0 & \text{auf } T_O & , & \Delta \psi_3^{n+1} &= 0 & \text{auf } T_R \\ \psi_2^{n+1} &= 0 & \text{auf } \partial T_O \setminus \Gamma_1 & , & \psi_3^{n+1} &= 0 & \text{auf } \partial T_R \setminus \Gamma_2 \\ \frac{\partial \psi_2^{n+1}}{\partial y} &= \frac{\partial u_2^{n+1}}{\partial y} - \frac{\partial u_1^{n+1}}{\partial y} & \text{auf } \Gamma_1 & , & \frac{\partial \psi_3^{n+1}}{\partial x} &= \frac{\partial u_3^{n+1}}{\partial x} - \frac{\partial u_1^{n+1}}{\partial x} & \text{auf } \Gamma_2 \end{aligned}$$

$$\begin{aligned} \Delta \psi_4^{n+1} &= 0 & \text{auf } T_U & , & \Delta \psi_5^{n+1} &= 0 & \text{auf } T_L \\ \psi_4^{n+1} &= 0 & \text{auf } \partial T_U \setminus \Gamma_3 & , & \psi_5^{n+1} &= 0 & \text{auf } \partial T_L \setminus \Gamma_4 \\ \frac{\partial \psi_4^{n+1}}{\partial y} &= \frac{\partial u_4^{n+1}}{\partial y} - \frac{\partial u_1^{n+1}}{\partial y} & \text{auf } \Gamma_3 & , & \frac{\partial \psi_5^{n+1}}{\partial x} &= \frac{\partial u_5^{n+1}}{\partial x} - \frac{\partial u_1^{n+1}}{\partial x} & \text{auf } \Gamma_4 \end{aligned}$$

3. Prüfe das Abbruchkriterium:

$$\begin{aligned} \|(\psi_1^{n+1} + \psi_2^{n+1})|_{\Gamma_1}\|_{l_1(\Gamma_1)} &\leq \varepsilon_\Gamma, \\ \|(\psi_1^{n+1} + \psi_3^{n+1})|_{\Gamma_2}\|_{l_1(\Gamma_2)} &\leq \varepsilon_\Gamma, \\ \|(\psi_1^{n+1} + \psi_4^{n+1})|_{\Gamma_3}\|_{l_1(\Gamma_3)} &\leq \varepsilon_\Gamma, \\ \|(\psi_1^{n+1} + \psi_5^{n+1})|_{\Gamma_4}\|_{l_1(\Gamma_4)} &\leq \varepsilon_\Gamma. \end{aligned}$$

Falls das Abbruchkriterium erfüllt ist, wird die Iteration beendet, andernfalls wird weiter iteriert.

4. Update  $u_\Gamma^{n+1}$ :

$$\begin{aligned} u_{\Gamma_1}^{n+1} &= u_{\Gamma_1}^n - \theta(\psi_1^{n+1} + \psi_2^{n+1}) & \text{auf } \Gamma_1, \\ u_{\Gamma_2}^{n+1} &= u_{\Gamma_2}^n - \theta(\psi_1^{n+1} + \psi_3^{n+1}) & \text{auf } \Gamma_2, \\ u_{\Gamma_3}^{n+1} &= u_{\Gamma_3}^n - \theta(\psi_1^{n+1} + \psi_4^{n+1}) & \text{auf } \Gamma_3, \\ u_{\Gamma_4}^{n+1} &= u_{\Gamma_4}^n - \theta(\psi_1^{n+1} + \psi_5^{n+1}) & \text{auf } \Gamma_4. \end{aligned}$$

Wir setzen  $n \rightarrow n + 1$  und gehen zu Schritt 1.

Unter Verwendung der gelisteten Iteration erfassen wir folgende Ergebnisse:

$\varepsilon_\Gamma$	N	Koll.punkte	It	$T_O$	$T_U$	$T_L$	$T_R$	R
$1e^{-4}$	6	161	37	1.07e-01	8.94e-02	6.91e-02	1.01e-01	1.10e-01
	12	533	37	1.19e-03	7.15e-04	1.65e-03	1.41e-03	1.60e-03
	18	1121	40	8.14e-07	6.31e-07	1.22e-06	9.21e-07	1.16e-06
$1e^{-6}$	6	161	53	1.07e-01	8.94e-02	6.91e-02	1.01e-01	1.10e-01
	12	533	54	1.19e-03	7.15e-04	1.65e-03	1.41e-03	1.60e-03
	18	1121	57	7.10e-07	7.58e-07	1.15e-06	1.00e-06	1.17e-06
$1e^{-8}$	6	161	69	1.07e-01	8.94e-02	6.91e-02	1.01e-01	1.10e-01
	12	533	71	1.19e-03	7.15e-04	1.65e-03	1.41e-03	1.60e-03
	18	1121	74	7.09e-07	7.60e-07	1.15e-06	1.00e-06	1.17e-06
$1e^{-10}$	6	161	86	1.07e-01	8.94e-02	6.91e-02	1.01e-01	1.10e-01
	12	533	88	1.19e-03	7.15e-04	1.65e-03	1.41e-03	1.60e-03
	18	1121	92	7.09e-07	7.60e-07	1.15e-06	1.00e-06	1.17e-06
$1e^{-11}$	18	1121	103	7.09e-07	7.60e-07	1.15e-06	1.00e-06	1.17e-06

Im dritten Test beobachten wir, aufgrund der komplexeren Geometrie des Problems, größere Approximationsfehler für alle  $\varepsilon_\Gamma$  im Vergleich zu den vorangegangenen betrachteten Tests. Da für  $N = 18$  die besten Approximationen im Fehlerbereich von  $\approx 10^{-7}$  auf allen Gebieten liegen, haben verschiedene Werte von  $\varepsilon_\Gamma$  kaum Einfluss und schon mit  $\varepsilon_\Gamma = 1e^{-4}$  werden annähernd die besten Werte erreicht. Die Anzahl der Iterationen ist über  $N$  fast konstant und variiert nur mit  $\varepsilon_\Gamma$ .

Ziehen wir die Werte mit den Werten in Abschnitt 6.3 zum Vergleich heran, so finden wir schnell heraus, dass bei  $\varepsilon_\Gamma = 1e^{-4}$  Werte in der gleichen Größenordnung zu den Werten in Abschnitt 6.3 erreicht werden. Diese werden sogar mit der Hälfte der Iterationen erzielt. Im weiteren Vergleich mit [HL01] schneidet die hier verwendete Neumann-Neumann Iteration auch besser ab. In [HL01] wurde für  $N = 16$  ein Fehler in der Größenordnung von  $\approx 10^{-5}$  bei 72 Iterationen erreicht. Ziehen wir den Vergleich mit  $\varepsilon_\Gamma = 1e^{-4}$  und  $N = 18$ , so benötigt die hier untersuchte Methode nur 40 Iterationen und erzeugt Fehlergrößen der Ordnung  $\approx 10^{-7}$ .

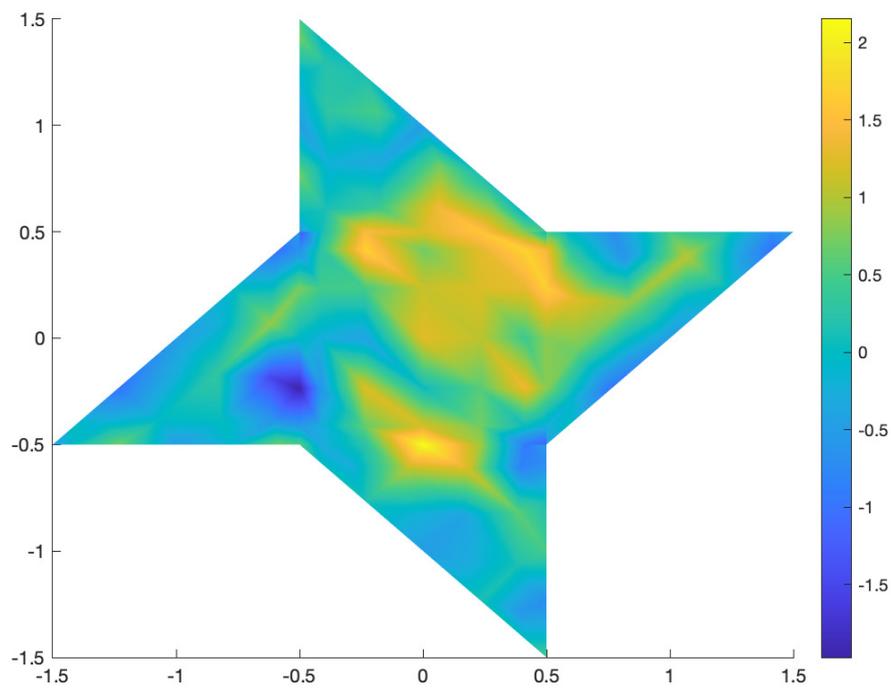


Abbildung 7.5: Lösung:  $\varepsilon_\Gamma = 1e^{-11}$ ,  $N = 6$

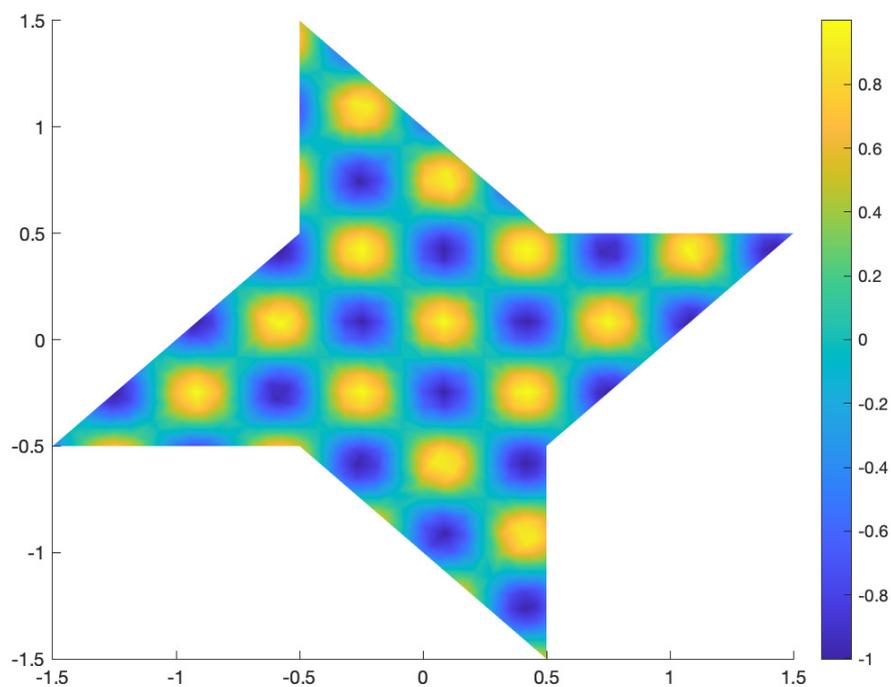


Abbildung 7.6: Lösung:  $\varepsilon_\Gamma = 1e^{-11}$ ,  $N = 18$

### 7.1.4 Numerischer Test 4

In unserem letzten Beispiel betrachten wir das Gebiet  $\Omega_3 = T_O \cup T_U \cup T_L \cup T_M$  aus dem Abschnitt 6.4. Wir benutzen dieselben Fekete-Punkte, skizziert in Abbildung 6.18 und 6.19.

Als Startwerte der Neumann-Neumann Iteration verwenden wir  $u_\Gamma^0 = \mathbf{0}$  und den Parameter  $\theta = 0.1$ .

Die Iteration im  $n$ -ten Schritt lautet wie folgt:

1. Lösen auf jedem Teilgebiet ein Dirichlet-Problem:

$$\begin{aligned} \Delta u_1^{n+1} &= f & \text{auf } T_M \\ u_1^{n+1} &= g & \text{auf } \partial T_M \setminus \{\Gamma_1 \cup \Gamma_2 \cup \Gamma_3\} \\ u_1^{n+1} &= u_{\Gamma_1}^n & \text{auf } \Gamma_1 \\ u_1^{n+1} &= u_{\Gamma_2}^n & \text{auf } \Gamma_2 \\ u_1^{n+1} &= u_{\Gamma_3}^n & \text{auf } \Gamma_3 \end{aligned} \quad , \quad \begin{aligned} \Delta u_2^{n+1} &= f & \text{auf } T_L \\ u_2^{n+1} &= g & \text{auf } \partial T_L \setminus \Gamma_1 \\ u_2^{n+1} &= u_{\Gamma_1}^n & \text{auf } \Gamma_1 \end{aligned}$$

$$\begin{aligned} \Delta u_3^{n+1} &= f & \text{auf } T_O \\ u_3^{n+1} &= g & \text{auf } \partial T_O \setminus \Gamma_2 \\ u_3^{n+1} &= u_{\Gamma_2}^n & \text{auf } \Gamma_2 \end{aligned} \quad , \quad \begin{aligned} \Delta u_4^{n+1} &= f & \text{auf } T_U \\ u_4^{n+1} &= g & \text{auf } \partial T_U \setminus \Gamma_3 \\ u_4^{n+1} &= u_{\Gamma_3}^n & \text{auf } \Gamma_3 \end{aligned}$$

2. Löse auf jedem Teilgebiet ein Neumann-Problem:

$$\begin{aligned} \Delta \psi_1^{n+1} &= 0 & \text{auf } T_M \\ \psi_1^{n+1} &= 0 & \text{auf } \partial T_M \setminus \{\Gamma_1 \cup \Gamma_2 \cup \Gamma_3\} \\ \frac{\partial \psi_1^{n+1}}{\partial x} &= \frac{\partial u_1^{n+1}}{\partial x} - \frac{\partial u_2^{n+1}}{\partial x} & \text{auf } \Gamma_1 \\ \frac{\partial \psi_1^{n+1}}{\partial y} &= \frac{\partial u_1^{n+1}}{\partial y} - \frac{\partial u_3^{n+1}}{\partial y} & \text{auf } \Gamma_2 \\ \frac{\partial \psi_1^{n+1}}{\partial n} &= \frac{\partial u_1^{n+1}}{\partial n} - \frac{\partial u_4^{n+1}}{\partial n} & \text{auf } \Gamma_3 \end{aligned} \quad , \quad \begin{aligned} \Delta \psi_2^{n+1} &= 0 & \text{auf } T_L \\ \psi_2^{n+1} &= 0 & \text{auf } \partial T_L \setminus \Gamma_1 \\ \frac{\partial \psi_2^{n+1}}{\partial x} &= \frac{\partial u_2^{n+1}}{\partial x} - \frac{\partial u_1^{n+1}}{\partial x} & \text{auf } \Gamma_1 \end{aligned}$$

$$\begin{aligned} \Delta \psi_3^{n+1} &= 0 & \text{auf } T_O \\ \psi_3^{n+1} &= 0 & \text{auf } \partial T_O \setminus \Gamma_2 \\ \frac{\partial \psi_3^{n+1}}{\partial n} &= \frac{\partial u_3^{n+1}}{\partial n} - \frac{\partial u_1^{n+1}}{\partial n} & \text{auf } \Gamma_2 \end{aligned} \quad , \quad \begin{aligned} \Delta \psi_4^{n+1} &= 0 & \text{auf } T_U \\ \psi_4^{n+1} &= 0 & \text{auf } \partial T_U \setminus \Gamma_3 \\ \frac{\partial \psi_4^{n+1}}{\partial y} &= \frac{\partial u_4^{n+1}}{\partial y} - \frac{\partial u_1^{n+1}}{\partial y} & \text{auf } \Gamma_3 \end{aligned}$$

wobei  $\frac{\partial}{\partial n}$  der Ableitung in Richtung des Einheitsnormalenvektor entlang des Interfaces  $\Gamma_2$  ist.

3. Prüfe das Abbruchkriterium:

$$\begin{aligned} \|(\psi_1^{n+1} + \psi_2^{n+1})|_{\Gamma_1}\|_{l_1(\Gamma_1)} &\leq \varepsilon_\Gamma, \\ \|(\psi_1^{n+1} + \psi_3^{n+1})|_{\Gamma_2}\|_{l_1(\Gamma_2)} &\leq \varepsilon_\Gamma, \\ \|(\psi_1^{n+1} + \psi_4^{n+1})|_{\Gamma_3}\|_{l_1(\Gamma_3)} &\leq \varepsilon_\Gamma. \end{aligned}$$

Falls das Abbruchkriterium erfüllt ist, wird die Iteration beendet, andernfalls wird weiter iteriert.

4. Update  $u_{\Gamma}^{n+1}$ :

$$\begin{aligned} u_{\Gamma_1}^{n+1} &= u_{\Gamma_1}^n - \theta(\psi_1^{n+1} + \psi_2^{n+1}) && \text{auf } \Gamma_1, \\ u_{\Gamma_2}^{n+1} &= u_{\Gamma_2}^n - \theta(\psi_1^{n+1} + \psi_3^{n+1}) && \text{auf } \Gamma_2, \\ u_{\Gamma_3}^{n+1} &= u_{\Gamma_3}^n - \theta(\psi_1^{n+1} + \psi_4^{n+1}) && \text{auf } \Gamma_3. \end{aligned}$$

Wir setzen  $n \rightarrow n + 1$  und gehen zu Schritt 1.

Unter Verwendung der obigen Iteration tragen wir folgende Ergebnisse zusammen:

$\varepsilon_{\Gamma}$	N	Koll.punkte	It	$T_O$	$T_U$	$T_L$	$T_M$
$1e^{-4}$	6	112	47	3.27e-01	1.21e-01	1.21e-01	3.93e-01
	12	364	48	4.37e-03	3.27e-03	3.27e-03	7.23e-03
	18	760	54	1.46e-06	2.01e-06	2.01e-06	3.51e-06
$1e^{-6}$	6	112	68	3.27e-01	1.21e-01	1.21e-01	3.93e-01
	12	364	70	4.37e-03	3.27e-03	3.27e-03	7.23e-03
	18	760	78	1.33e-06	1.89e-06	1.89e-06	3.23e-06
$1e^{-8}$	6	112	88	3.27e-01	1.21e-01	1.21e-01	3.93e-01
	12	364	92	4.37e-03	3.27e-03	3.27e-03	7.23e-03
	18	760	102	1.33e-06	1.89e-06	1.89e-06	3.22e-06
$1e^{-10}$	6	112	109	3.27e-01	1.21e-01	1.21e-01	3.93e-01
	12	364	114	4.37e-03	3.27e-03	3.27e-03	7.23e-03
	18	760	126	1.33e-06	1.89e-06	1.89e-06	3.22e-06
$1e^{-12}$	6	112	130	3.27e-01	1.21e-01	1.21e-01	3.93e-01
	12	364	137	4.37e-03	3.27e-03	3.27e-03	7.23e-03
	18	760	157	1.33e-06	1.89e-06	1.89e-06	3.22e-06

In unserem letzten numerischen Test nehmen wir wahr, dass wir größere Approximationsfehler für alle  $\varepsilon_{\Gamma}$  haben. Dies liegt an der komplexeren Geometrie des Problems. Für  $N = 18$  haben wir auf allen Gebieten die besten Approximationen im Fehlerbereich von  $\approx 10^{-6}$ . Deshalb haben verschiedene Werte von  $\varepsilon_{\Gamma}$  wenig Einwirkung und schon mit  $\varepsilon_{\Gamma} = 1e^{-4}$  werden annähernd die besten Werte erreicht. Die Anzahl der Iterationen ist über  $N$  fast konstant und variiert nur mit  $\varepsilon_{\Gamma}$ .

Vergleichen wir die Werte mit den Werten in Abschnitt 6.4, so stellen wir fest, dass bei  $\varepsilon_{\Gamma} = 1e^{-4}$  Werte in der gleichen Größenordnung zu den Werten in Abschnitt 6.4 erreicht werden. Wir benötigen hier deutlich weniger Iterationen.

Im weiteren Vergleich mit [HL01] schneidet die hier verwendete Neumann-Neumann Iteration vergleichbar ab. In [HL01] wurde für  $N = 16$  ein Fehler in der Größenordnung von  $\approx 10^{-5}$  bei 46 Iterationen erzielt. Stellen wir die Werte mit  $\varepsilon_{\Gamma} = 1e^{-4}$  und  $N = 18$  nebeneinander, erkennen wir, dass unser Verfahren zwar 54 Iterationen benötigt, aber Fehlergrößen der Ordnung  $\approx 10^{-6}$  erzeugt.

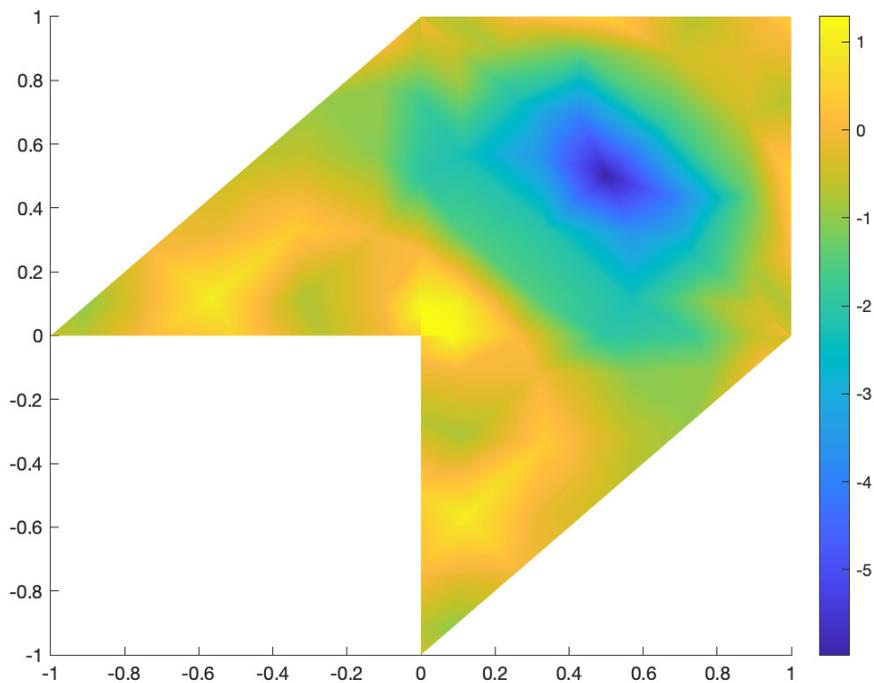


Abbildung 7.7: Lösung:  $\varepsilon_{\Gamma} = 1e^{-12}$ ,  $N = 6$

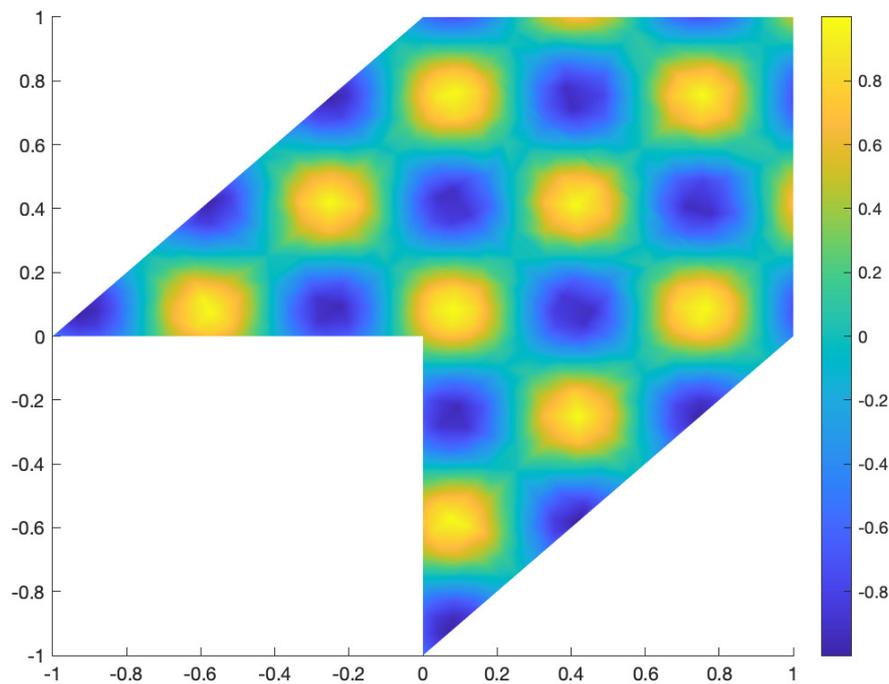


Abbildung 7.8: Lösung:  $\varepsilon_{\Gamma} = 1e^{-12}$ ,  $N = 18$

## 7.2 Betrachtung von $h$ - und $p$ -Verfeinerung

In diesem Abschnitt betrachten wir, neben der im letzten Kapitel untersuchten  $p$ -Verfeinerung, das heißt die Verwendung unterschiedlicher und höherer Polynomgrade, auch die  $h$ -Verfeinerung. Unter  $h$ -Verfeinerung wird der Einsatz von kleiner werdenden Dreiecken unter Verwendung des gleichen Gebietes  $\Omega$  verstanden. Hier ist  $h$  die Größenordnung der längsten Seite aller Dreiecke, die  $\Omega$  ausmachen. Wird  $h$  verkleinert, so werden mehr kleinere Dreiecke benötigt, um  $\Omega$  zu füllen. Ein höherer Polynomgrad sowie mehr Dreiecke erhöhen den Rechenaufwand. Beide Verfahren verfolgen das Ziel, den Approximationsfehler zu verringern, jedoch ist es meist ungewiss, welches Verfahren effizienter ist. Daher untersuchen wir in diesem Abschnitt verschiedene Fälle und verfeinern in  $h$  und  $p$ .

Im Allgemeinen beschäftigen wir uns in diesem Abschnitt mit quadratischen oder rechteckigen Gebieten  $\Omega$ . Wir zerlegen diese gleichmäßig entlang der  $x$ - und  $y$ -Achse in Dreiecke. Beispiele zu diesen Zerlegungen werden an den jeweiligen Stellen aufgegriffen und abgebildet.

Um im Folgenden die Algorithmen besser beschreiben zu können, führen wir folgende Notation ein:

Wir betrachten die Zerlegung eines Gebietes

$$\Omega = \cup_{T_i \in \mathcal{T}_h} T_i, \quad \mathcal{T}_h = \{T_1, T_2, \dots, T_n\}$$

in  $n$  Dreiecke  $T_i \in \mathcal{T}_h$ . Hierbei nehmen wir eine reguläre Triangulierung an, siehe dazu [QV08]. Jedes Dreieck  $T_i \in \mathcal{T}_h$  hat drei Kanten  $E_i^1, E_i^2$  und  $E_i^3$ , wobei eine Kante entweder bis auf einzelne Punkte in  $\Omega$  liegt oder komplett auf dem Rand  $\partial\Omega$ , daher unterscheiden wir in verschiedene Mengen von Kanten.

$$\begin{aligned} \mathcal{E}_h &\equiv \text{Menge aller Kanten} \\ \mathcal{E}_h^{\partial\Omega} &\equiv \text{Menge aller Kanten auf } \partial\Omega \\ \mathcal{E}_h^\circ &= \mathcal{E}_h \setminus \mathcal{E}_h^{\partial\Omega} \equiv \text{Menge aller inneren Kanten} \end{aligned}$$

Diese drei Mengen wollen wir auch im Bezug auf einzelne Dreieck  $T_i \in \mathcal{T}_h$  definieren.

$$\begin{aligned} \mathcal{E}_h(T_i) &\equiv \text{Menge der Kanten von } T_i \\ \mathcal{E}_h^{\partial\Omega}(T_i) &\equiv \text{Menge der Kanten } E \in \mathcal{E}_h(T_i) \text{ von } T_i \text{ mit } E \cap \partial\Omega = E \\ \mathcal{E}_h^\circ(T_i) &= \mathcal{E}_h(T_i) \setminus \mathcal{E}_h^{\partial\Omega}(T_i) \equiv \text{Menge aller inneren Kanten von } T_i \end{aligned}$$

Im Weiteren werden Elemente  $\Gamma_i \in \mathcal{E}_h^\circ = \{\Gamma_1, \Gamma_2, \dots, \Gamma_m\}$  Interfaces genannt. Funktionen  $u$ , die entlang des Interfaces  $\Gamma_i$  definiert sind, bezeichnen wir mit  $u_{\Gamma_i}$ . Jedes  $\Gamma_i$  hat eine eindeutige äußere Einheitsnormale  $n_i$ , welche orthogonal entlang des Interfaces  $\Gamma_i$  ist.

Zuletzt werden wir die Eckpunkte eines Dreiecks kategorisieren. Ein Dreieck  $T_i \in \mathcal{T}_h$  hat immer drei Eckpunkte  $p_i^1, p_i^2$  und  $p_i^3$ . Für einen Eckpunkt gibt es zwei Möglichkeiten, entweder liegt ein Eckpunkt auf  $\partial\Omega$  oder im inneren von  $\Omega$ . Wir definieren dazu die folgenden allgemeinen Mengen und auf einzelne Dreiecke  $T_i \in \mathcal{T}_h$  bezogene Mengen:

$$\begin{aligned} \mathcal{N}_h &\equiv \text{Menge alle Eckpunkte von allen Dreiecken von } \mathcal{T}_h \\ \mathcal{N}_h(T_i) &= \{p_i^1, p_i^2, p_i^3 \in \mathcal{N}_h\} \equiv \text{Menge der Eckpunkte von } T_i \\ \mathcal{N}_h^{\partial\Omega}(T_i) &= \mathcal{N}_h(T_i) \cap \partial\Omega \equiv \text{Menge der Eckpunkte von } T_i \text{ auf } \partial\Omega \\ \mathcal{N}_h^\circ(T_i) &= \mathcal{N}_h(T_i) \setminus \mathcal{N}_h^{\partial\Omega}(T_i) \equiv \text{Menge der Eckpunkte von } T_i \text{ im inneren von } \Omega \end{aligned}$$

Zur Formulierung der Algorithmen werden über die Mengen von Kanten  $\mathcal{E}_h^{\partial\Omega}(T_i)$  und  $\mathcal{E}_h^\circ(T_i)$  Bedingungen gefordert. Sollte für ein Dreieck  $\mathcal{E}_h^{\partial\Omega}(T_i) = \emptyset$  oder  $\mathcal{E}_h^\circ(T_i) = \emptyset$  gelten, so wird diese Bedingung nicht für das Dreieck gefordert. Dasselbe gilt für

Bedingungen über die Mengen  $\mathcal{N}_h^{\partial\Omega}(T_i)$  und  $\mathcal{N}_h^\circ(T_i)$ .

### 7.2.1 Klassische Poisson-Gleichung

Wir betrachten die klassische Poisson-Gleichung auf dem Gebiet  $\Omega = [-2, 2] \times [-1, 1]$  gegeben durch

$$-\Delta u = f \text{ in } \Omega, \quad (7.1)$$

$$u = g \text{ auf } \partial\Omega, \quad (7.2)$$

mit

$$f = -2\pi^2 \sin(x\pi) \sin\left(\pi y + \frac{\pi}{4}\right), \quad g = \sin(x\pi) \sin\left(\pi y + \frac{\pi}{4}\right).$$

Die analytische Lösung zu diesem Problem ist durch

$$u(x, y) = \sin(x\pi) \sin\left(\pi y + \frac{\pi}{4}\right) \quad (7.3)$$

gegeben.

Mit Hilfe der eingeführten Notation nehmen wir eine Zerlegung von  $\Omega$  in Dreiecke  $\mathcal{T}_h$  an und geben einen Startwert

$$\forall \Gamma_i \in \mathcal{E}_h^\circ : \quad u_{\Gamma_i}^{(0)} = 0$$

vor. Damit können wir die Neumann-Neumann Iteration wie folgt beschreiben.

Sei  $n = 0$ :

1. Auf jedem  $T_i \in \mathcal{T}_h$  wird das folgendes Dirichlet-Problem gelöst:

$$\begin{aligned} & -\Delta u_{T_i}^{(n)} = f && \text{in } T_i, \\ \forall E \in \mathcal{E}_h^{\partial\Omega}(T_i) : & u_{T_i}^{(n)} = g && \text{auf } E, \\ \forall \Gamma_i \in \mathcal{E}_h^\circ(T_i) : & u_{T_i}^{(n)} = u_{\Gamma_i}^{(n)} && \text{auf } \Gamma_i \setminus \mathcal{N}_h^\circ(T_i), \\ \forall p_i^j \in \mathcal{N}_h^\circ(T_i) : & -\Delta u_{T_i}^{(n)}(p_i^j) = f(p_i^j). \end{aligned}$$

2. Auf jedem  $T_i \in \mathcal{T}_h$  wird das folgendes Neumann-Problem gelöst:

$$\begin{aligned}
 \forall E \in \mathcal{E}_h^{\partial\Omega}(T_i) : \quad & -\Delta\psi_{T_i}^{(n)} = 0 && \text{in } T_i, \\
 & \psi_{T_i}^{(n)} = 0 && \text{auf } E, \\
 \forall \Gamma_i \in \mathcal{E}_h^\circ(T_i) : \quad & \psi_{T_i}^{(n)} = \frac{\partial u_{T_i}^{(n)}}{\partial n_i} - \frac{\partial u_{T_j}^{(n)}}{\partial n_i} && \text{auf } \Gamma_i \setminus \mathcal{N}_h^\circ(T_i), \\
 \forall p_i^j \in \mathcal{N}_h^\circ(T_i) : \quad & -\Delta\psi_{T_i}^{(n)}(p_i^j) = 0.
 \end{aligned}$$

Hier ist  $T_j$  das an  $\Gamma_i$  angrenzende Dreieck mit  $T_i \cap T_j = \Gamma_i$ .

3. Prüfe das Abbruchkriterium:

$$\forall \Gamma_i \in \mathcal{E}_h^\circ : \quad \|(\psi_{T_i}^{(n)} + \psi_{T_j}^{(n)})|_{\Gamma_i}\|_{L_1(\Gamma_i)} \leq \varepsilon_\Gamma \quad , \text{ wobei } T_i, T_j \in \mathcal{T}_h : T_i \cap T_j = \Gamma_i.$$

Sollte das Abbruchkriterium auf allen  $\Gamma_i$  erfüllt sein, dann kann die Iteration abgebrochen werden.

4. Update  $u_{\Gamma_i}^{(n+1)}$ :

$$\forall \Gamma_i \in \mathcal{E}_h^\circ : \quad u_{\Gamma_i}^{(n+1)} = u_{\Gamma_i}^{(n)} - \theta \left( \psi_{T_i}^{(n)} + \psi_{T_j}^{(n)} \right) \quad , \text{ wobei } T_i, T_j \in \mathcal{T}_h : T_i \cap T_j = \Gamma_i.$$

Setze  $n \rightarrow n + 1$  und gehe wieder zu Schritt 1.

Die Neumann-Neumann Iteration wird mit  $\theta = 0.1$  angewandt.

Die Abbildungen 7.9 und 7.10 zeigen das gröbste Gitter, bestehend aus 8 Dreiecken, und das zweit feinste Gitter, bestehend aus 32 Dreiecken. Ebenfalls sind die Kollokationspunkte unter Verwendung von Fekete-Punkten auf den Abbildungen illustriert. In der zweiten Abbildung 7.10 erkennen wir die starke Zunahme von Kollokationspunkten durch die Verwendung von mehr Dreiecken.

Die Abbildungen 7.11 und 7.12 bilden die unterschiedlichen numerischen Lösungen für die Fälle  $nT = 8, N = 6$  und  $nT = 32, N = 12$ . Dabei gibt  $nT$  die Anzahl der Dreiecke an. Wir nehmen insbesondere gut die zusätzliche Approximationsordnung im Fall  $nT = 32, N = 12$  wahr. Der erste Fall in Abbildung 7.11 zeigt noch die klar erkennbaren Kanten, der eigentlich glatten Lösung, besonders in den Ecken des unterliegenden Gitters. Im Fall  $nT = 32, N = 12$  bemerken wir eine sehr glatte Lösung, die sich augenscheinlich nicht von einer analytischen Lösung unterscheidet.

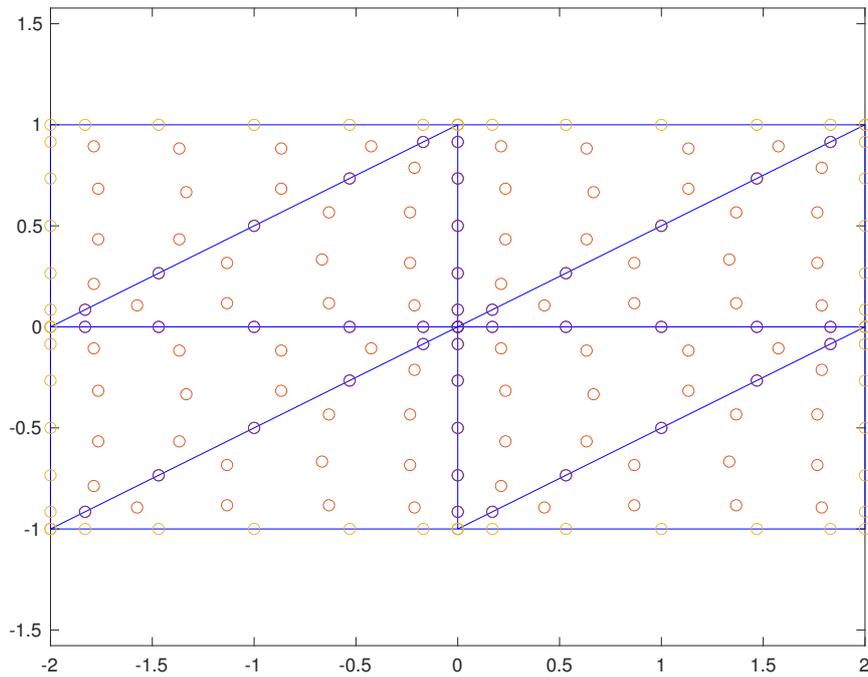


Abbildung 7.9: Gitter mit Kollokationspunkten:  $nT = 8$ ,  $N = 6$

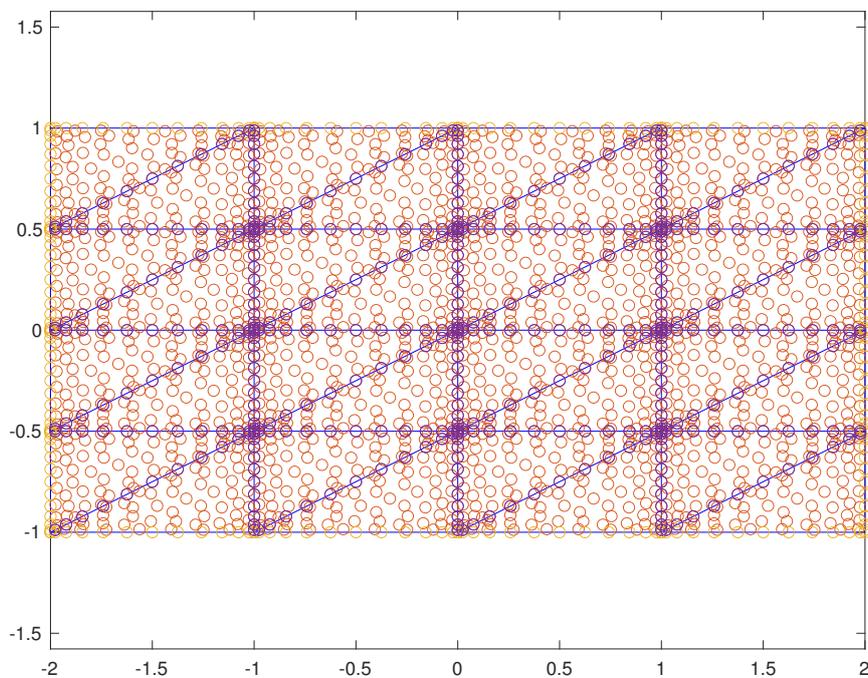


Abbildung 7.10: Gitter mit Kollokationspunkten:  $nT = 32$ ,  $N = 12$

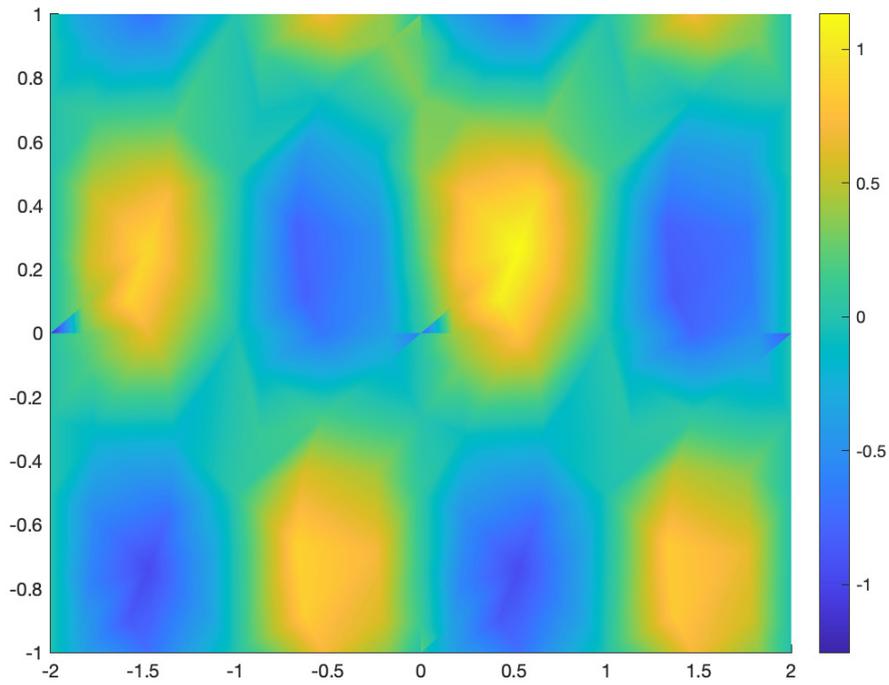


Abbildung 7.11: Lösung:  $nT = 8$ ,  $N = 6$

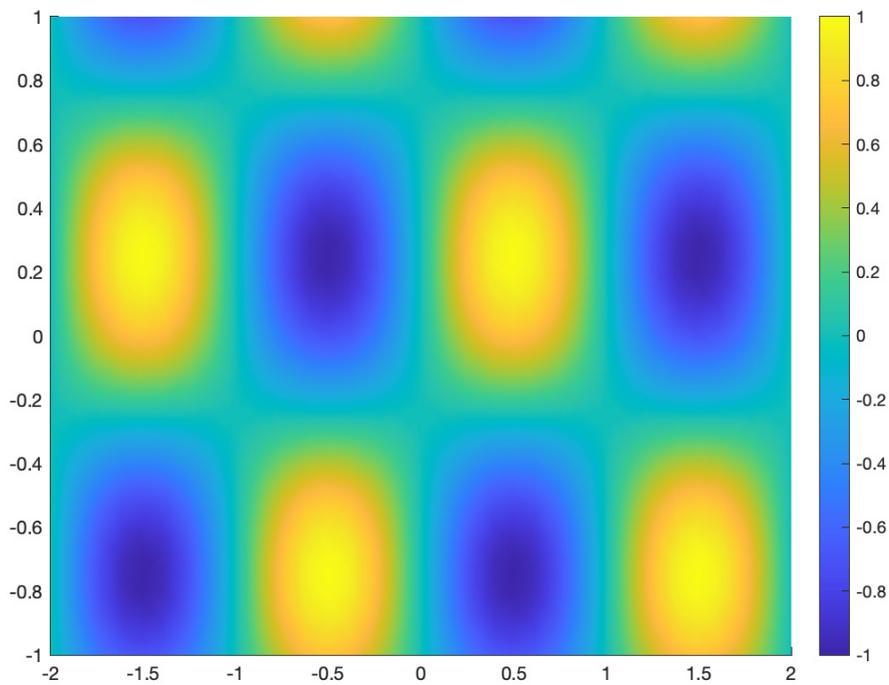


Abbildung 7.12: Lösung:  $nT = 32$ ,  $N = 12$

Tabelle 7.1: Ergebnisse klassische Poisson-Gleichung

$\varepsilon_T$	$nT$	N	Kollokationspunkte	Fehler	It	Zeit (sek)
$1e^{-06}$	8	6	224	1.01e-02	306	0.67
		9	440	2.65e-04	366	1.08
		12	728	8.17e-07	466	2.08
		15	1088	2.65e-08	536	4.01
	18	6	504	9.13e-04	407	1.82
		9	990	9.28e-06	574	3.80
		12	1638	2.11e-08	750	7.77
		15	2448	1.88e-08	875	14.38
	32	6	896	1.67e-04	376	2.85
		9	1760	3.27e-07	492	5.21
		12	2912	2.73e-08	519	8.72
		15	4352	2.52e-08	566	17.07
	50	6	1400	4.78e-05	497	5.42
		9	2750	4.98e-08	734	11.55
		12	4550	1.29e-08	833	21.74
		15	6800	1.15e-08	990	47.69
$1^{-08}$	8	6	224	1.01e-02	432	0.87
		9	440	2.65e-04	542	1.56
		12	728	8.39e-07	707	3.02
		15	1088	2.03e-09	820	5.81
	18	6	504	9.13e-04	574	2.38
		9	990	9.29e-06	796	4.53
		12	1638	3.42e-09	1050	10.16
		15	2448	1.88e-10	1219	20.32
	32	6	896	1.67e-04	681	4.60
		9	1760	3.25e-07	924	9.60
		12	2912	3.75e-10	1047	17.11
		15	4352	2.82e-10	1186	35.08
	50	6	1400	4.78e-05	880	9.27
		9	2750	2.76e-08	1323	20.35
		12	4550	2.71e-10	1235	33.45
		15	6800	3.02e-10	1649	80.59
$1e^{-10}$	8	6	224	1.01e-02	558	1.16
		9	440	2.65e-04	719	2.17
		12	728	8.40e-07	947	4.39

$\varepsilon_\Gamma$	$nT$	$N$	Kollokationspunkte	Fehler	It	Zeit (sek)
		15	1088	1.91e-09	1105	8.88
	18	6	504	9.13e-04	752	3.35
		9	990	9.29e-06	1032	6.26
		12	1638	3.48e-09	1359	15.08
		15	2448	1.12e-11	1565	29.88
	32	6	896	1.67e-04	1028	8.31
		9	1760	3.25e-07	1367	15.70
		12	2912	1.87e-10	1610	29.75
		15	4352	2.80e-12	1826	57.43
	50	6	1400	4.78e-05	1380	16.30
		9	2750	2.75e-08	1970	33.93
		12	4550	1.42e-11	1981	62.43
		15	6800	3.19e-12	2555	125.45
$1e^{-12}$	8	6	224	1.01e-02	684	1.42
		9	440	2.65e-04	897	2.70
		12	728	8.40e-07	1190	5.15
		15	1088	1.91e-09	1389	9.78
	18	6	504	9.13e-04	947	4.02
		9	990	9.29e-06	1290	7.64
		12	1638	3.48e-09	1680	16.73
		15	2448	1.03e-11	1914	31.93
	32	6	896	1.67e-04	1377	9.85
		9	1760	3.25e-07	1811	21.26
		12	2912	1.88e-10	2174	41.20
		15	4352	7.74e-14	2467	75.85
	50	6	1400	4.78e-05	1902	21.75
		9	2750	2.74e-08	2627	43.99
		12	4550	1.32e-11	2802	86.35
		15	6800	3.24e-14	3471	175.31

Die numerischen Versuche untersuchten die Auswirkung von verschiedenen Abbruchkriterien  $\varepsilon_\Gamma$ , Anzahl an Dreiecken und unterschiedliche Polynomgrade. Der Einfluss von  $\varepsilon_\Gamma$  ist wenig überraschend. Je kleiner  $\varepsilon_\Gamma$  gewählt wird, desto mehr Iterationen werden benötigt. Vergleichen wir die Fälle kleinster Ordnung, hier  $nT = 8$ ,  $N = 6$ , für  $\varepsilon_\Gamma = 1e^{-06}$  und  $\varepsilon_\Gamma = 1e^{-12}$ , stellen wir fest, dass es keine Auswirkung auf den Fehler hat, aber sehr wohl auf die Anzahl der Iterationen, die im Fall  $\varepsilon_\Gamma = 1e^{-12}$

doppelt so hoch sind wie für  $\varepsilon_\Gamma = 1e^{-06}$  (vgl. Tabelle 7.1). Somit schlussfolgern wir, dass es nur lohnenswert ist, kleine  $\varepsilon_\Gamma$  zu wählen, wenn wir davon ausgehen können, dass der Fehler auf den Interfaces dominieren könnte und bessere Approximationen möglich sind. Einen solchen Fall sehen wir für  $\varepsilon_\Gamma = 1e^{-06}$  und  $\varepsilon_\Gamma = 1e^{-12}$  in Verbindung mit  $nT = 32$ ,  $N = 15$ . Hier liegt der Fehler für  $\varepsilon_\Gamma = 1e^{-06}$  bei  $\approx 10^{-08}$ , der Fehler auf den Interfaces dominiert den globalen Fehler. Für  $\varepsilon_\Gamma = 1e^{-12}$  bemerken wir einen Fehler von  $\approx 10^{-14}$ , hier ist der Interface-Fehler nicht mehr ausschlaggebend und es kann annähernd Maschinengenauigkeit erreicht werden.

Der Einfluss der Anzahl der Dreiecke wird exemplarisch am Datensatz von  $\varepsilon = 1e^{-10}$  diskutiert. Das hier betrachtete Problem hat eine im ganzen Gebiet  $\Omega$  gleichmäßige Lösung, somit erwarten wir bei einer gleichmäßigen Verfeinerung auch einen zum Aufwand, den Kollokationspunkten, gleichmäßig fallenden Fehler. Diese Erwartung wird auch durch die Tests unterstützt. Das Abfallen des Fehlers wird aber vom Polynomgrad beeinflusst. Wir beobachten, dass ab einer bestimmten Größenordnung der Abfall des Fehlers geringer wird. Bei näherer Betrachtung in Abbildung 7.13 sehen wir einen Abfall des Fehlers mit Zunahme der Kollokationspunkte in direkter Abhängigkeit der Anzahl der Dreiecke. Ebenfalls erkennen wir gut, dass ein Zusammenspiel von  $h$ - und  $p$ -Verfeinerung benötigt wird, um weiter einen schnellen Abfall des Fehlers zu garantieren.

Zuletzt befassen wir uns mit dem Einfluss des Polynomgrades. Dazu prüfen wir als Beispiel den Fall  $\varepsilon = 1e^{-10}$  und  $nT = 18$ . Hier stellen wir einen exponentiellen Abfall des Fehlers in Abhängigkeit zum Polynomgrad fest. Dies deckt sich ebenfalls mit den Beobachtungen im vorherigen Kapitel, indem der Unterschied des Fehlers über den verschiedenen Polynomgraden beobachtet wurde.

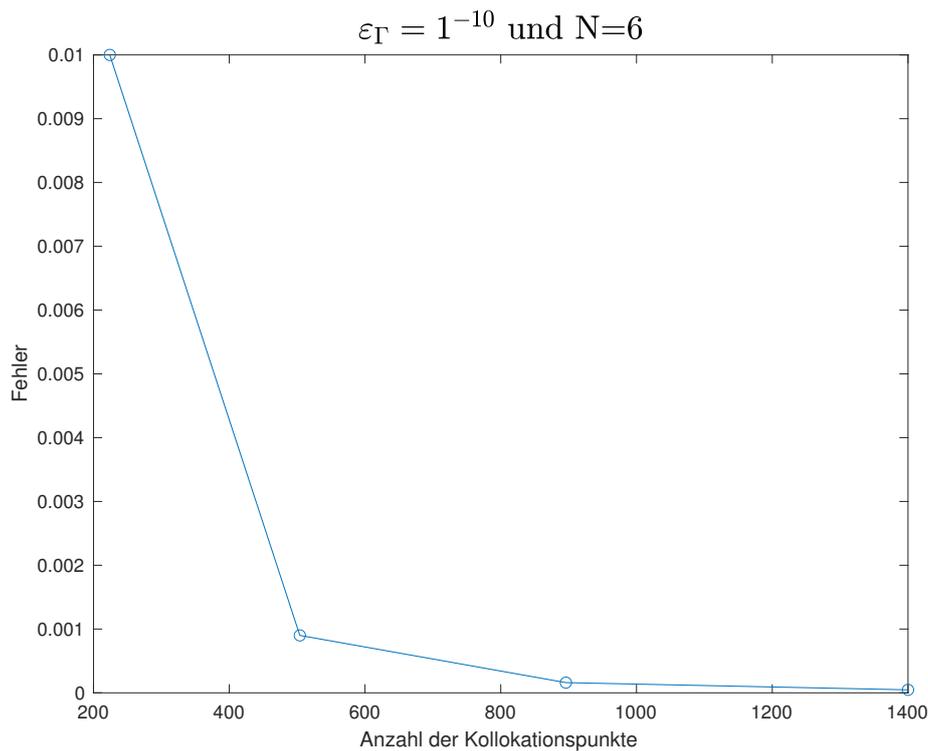


Abbildung 7.13: Konvergenzplot

## 7.2.2 Konvektions-Diffusionsgleichung

In diesem Abschnitt präsentieren wir die numerischen Ergebnisse für die Neumann-Neumann-Iteration angewandt auf die Konvektions-Diffusionsgleichung, welche wichtig für die Physik und die Transportphänomene ist.

Beim Transport physikalischer Größen wie Wärme oder Stoffkonzentration überlagern sich üblicherweise zwei Phänomene: Diffusion und Konvektion. Stofftransportphänomene spielen bei allen chemischen Prozessen eine wichtige Rolle. Auch bei thermischen Prozessen, die einer chemischen Reaktion oft vorausgehen, oder an sie anschließen, wie zum Beispiel das Auflösen fester Stoffe, Kondensieren oder Kristallisieren, sind diese oft von zentraler Bedeutung.

Bei der Diffusion wandern die kleinsten Teilchen eines Stoffs (Moleküle, Atome, Ionen) durch einen anderen, ruhenden Stoff hindurch. Diese werden in der Regel durch die ungeordnete Wärmebewegung der Teilchen angetrieben. Ein Beispiel für Diffusion ist der Gasaustausch zwischen Lungenbläschen und Blut, welcher bei der Lungenatmung erfolgt.

Bei der Konvektion werden ganze Stoffbezirke in einem strömenden Medium gegeneinander verschoben. Wie bei der Wärmeübertragung kann die Strömung durch Einwirkung äußerer Kräfte, zum Beispiel Rühren, oder innerer Kräfte, wie Dichte-

oder Druckunterschiede, entstehen. Im ersten Fall sprechen wir von einer erzwungenen und im zweiten Fall von einer natürlichen Konvektion. Ein Beispiel für den Stofftransport durch erzwungene Konvektion ist das Auflösen von Kaffeepulver in der Tasse bei gleichzeitigem Umrühren. Für den Stofftransport durch natürliche Konvektion ist das Abführen von Wasserdampf beim Trocknen von Wäsche an der Luft ein Beispiel [Chr09].

Treten Konvektion und Diffusion gleichzeitig auf, wie etwa bei verschiedenen Modellen der Fluidströmung, so lautet die Gleichung:

$$\begin{aligned} -\varepsilon\Delta u + b\nabla u + cu &= f \quad \text{in } \Omega \in \mathbb{R}^n \text{ offen,} \\ u &= g \quad \text{auf } \partial\Omega, \end{aligned} \tag{7.4}$$

wobei  $-\varepsilon\Delta u$  der Diffusionsanteil,  $b\nabla u$  der Konvektionsanteil und  $\varepsilon$  ein kleiner positiver Störungsparameter ist.  $b, c$  und  $f$  sind gegebene Funktionen. Für  $0 < \varepsilon \ll 1$  dominiert der Konvektionsanteil.

Bevor wir numerische Tests durchführen, betrachten wir ein Beispiel von einer linearen Konvektions-Diffusionsgleichung, um zu sehen, wie sich die Lösung verhält, wenn  $\varepsilon \rightarrow 0$  geht.

**Beispiel 7.2.1.** *Die Randwertaufgabe*

$$-\varepsilon u'' + u' = 1 \quad \text{in } \Omega = (0, 1), \quad u(0) = u(1) = 0 \tag{7.5}$$

hat die exakte Lösung

$$u(x, \varepsilon) = -\frac{e^{-\frac{1-x}{\varepsilon}} - e^{-\frac{1}{\varepsilon}}}{1 - e^{-\frac{1}{\varepsilon}}} + x.$$

Es ist

$$\lim_{\varepsilon \rightarrow 0} u(x, \varepsilon) = x.$$

Außerdem gilt für ein  $t \in [0, 1)$ :

$$\lim_{\varepsilon \rightarrow 0} \lim_{x \rightarrow t} u(x, \varepsilon) = t = \lim_{x \rightarrow t} \lim_{\varepsilon \rightarrow 0} u(x, \varepsilon).$$

Wegen der Randbedingung  $u(1) = 0$  ändert sich am rechten Rand dynamisch die Kurve, wo die Ableitung von  $u(x, \varepsilon)$  schnell sinkt:

$$0 = \lim_{\varepsilon \rightarrow 0} \lim_{x \rightarrow 1} u(x, \varepsilon) \neq \lim_{x \rightarrow 1} \lim_{\varepsilon \rightarrow 0} u(x, \varepsilon) = 1.$$

Regionen, in denen dies geschieht, werden als *layer* bezeichnet. Singulär-gestörte Probleme sind typischerweise durch das Vorhandensein solcher *layer* gekennzeichnet.

Wir nennen diese auch *singulär-gestörte Randwertprobleme*.

Die genaue Definition von *singulär-gestörten* Problemen ist gegeben durch

**Definition 7.2.1.** Sei  $B$  ein Funktionenraum ausgestattet mit der Norm  $\|\cdot\|_B$ . Sei  $D \subset \mathbb{R}^d$  ein Gebiet. Wir nennen die stetige Funktion  $u : D \rightarrow B, \varepsilon \mapsto u(\varepsilon)$  *regulär* für  $\varepsilon \rightarrow \varepsilon^* \in \partial D$ , falls eine Funktion  $u^* \in B$  existiert, sodass

$$\lim_{\varepsilon \rightarrow \varepsilon^*} \|u_\varepsilon - u^*\|_B = 0,$$

andernfalls nennen wir  $u_\varepsilon$  *singulär* für  $\varepsilon \rightarrow \varepsilon^*$ .

Sei  $(\mathcal{P}_\varepsilon)$  ein Problem mit einer Lösung  $u_\varepsilon \in B$  für alle  $\varepsilon \in D$ . Wir nennen  $(\mathcal{P}_\varepsilon)$  **singulär-gestört** für  $\varepsilon \rightarrow \varepsilon^* \in \partial D$  in der Norm  $\|\cdot\|_B$ , falls  $u$  *singulär* für  $\varepsilon \rightarrow \varepsilon^*$  ist. [Lin09]

**Bemerkung 7.2.1.** Die obige Definition ist von der Norm abhängig. Zum Beispiel ist unser betrachtetes Randwertproblem (7.5) *singulär-gestört* in der  $C^0([0, 1])$ -Norm,

denn es gilt  $u_\varepsilon \rightarrow u$  *punktweise* in  $[0, 1]$ , wobei  $u(x) = \begin{cases} x, & x \in [0, 1) \\ 0, & x = 1 \end{cases}$ .

Da aber  $u_\varepsilon$  für alle  $\varepsilon > 0$  *stetig* ist,  $u$  aber nicht, kann  $u_\varepsilon$  nicht *gleichmäßig* auf  $[0, 1]$  konvergieren. Es ist aber nicht *singulär-gestört* in der  $L_2$ -Norm, denn es existiert eine Funktion  $u^* : x \mapsto x$  mit  $\|u_\varepsilon - u^*\|_2 = \mathcal{O}(\varepsilon^{\frac{1}{2}})$ .

**Beweis.**

$$\begin{aligned} & \int_0^1 |u(x, \varepsilon) - x|^2 dx \\ &= \int_0^1 \left| -\frac{e^{-\frac{1-x}{\varepsilon}} - e^{-\frac{1}{\varepsilon}}}{1 - e^{-\frac{1}{\varepsilon}}} + x - x \right|^2 dx \\ &= \int_0^1 \left[ \frac{e^{-\frac{2-2x}{\varepsilon}} - 2e^{-\frac{2-x}{\varepsilon}} + e^{-\frac{2}{\varepsilon}}}{(1 - e^{-\frac{1}{\varepsilon}})^2} \right] dx \\ &= \frac{1}{(1 - e^{-\frac{1}{\varepsilon}})^2} \int_0^1 e^{-\frac{2-2x}{\varepsilon}} - 2e^{-\frac{2-x}{\varepsilon}} + e^{-\frac{2}{\varepsilon}} dx \\ &= \frac{e^{-\frac{2}{\varepsilon}}}{(1 - e^{-\frac{1}{\varepsilon}})^2} \int_0^1 e^{\frac{2x}{\varepsilon}} - 2e^{\frac{x}{\varepsilon}} + 1 dx \\ &= \frac{e^{-\frac{2}{\varepsilon}}}{(1 - e^{-\frac{1}{\varepsilon}})^2} \left[ \frac{\varepsilon}{2} e^{\frac{2x}{\varepsilon}} - 2\varepsilon e^{\frac{x}{\varepsilon}} + x \right]_0^1 \\ &= \frac{e^{-\frac{2}{\varepsilon}}}{(1 - e^{-\frac{1}{\varepsilon}})^2} \left[ \frac{\varepsilon}{2} e^{\frac{2}{\varepsilon}} - 2\varepsilon e^{\frac{1}{\varepsilon}} + 1 + \frac{3}{2}\varepsilon \right] \\ &= \frac{\frac{\varepsilon}{2} - 2\varepsilon e^{-\frac{1}{\varepsilon}} + e^{-\frac{2}{\varepsilon}} + \frac{3}{2}\varepsilon e^{-\frac{2}{\varepsilon}}}{(1 - e^{-\frac{1}{\varepsilon}})^2} =: c(\varepsilon) \in \mathcal{O}(\varepsilon), \end{aligned}$$

denn

$$\begin{aligned}
 & \left| c(\varepsilon) - \frac{\varepsilon}{2} \right| \\
 &= \left| c(\varepsilon) - \frac{\varepsilon (1 - e^{-\frac{1}{\varepsilon}})^2}{2 (1 - e^{-\frac{1}{\varepsilon}})^2} \right| \\
 &= \left| \frac{1}{(1 - e^{-\frac{1}{\varepsilon}})^2} \cdot \left[ \frac{\varepsilon}{2} - 2\varepsilon e^{-\frac{1}{\varepsilon}} + e^{-\frac{2}{\varepsilon}} + \frac{3}{2}\varepsilon e^{-\frac{2}{\varepsilon}} - \frac{\varepsilon}{2}(1 - 2e^{-\frac{1}{\varepsilon}} + e^{-\frac{2}{\varepsilon}}) \right] \right| \\
 &= \left| \underbrace{\frac{1}{(1 - e^{-\frac{1}{\varepsilon}})^2}}_{\rightarrow 1} \cdot \left[ \underbrace{-\varepsilon e^{-\frac{1}{\varepsilon}}}_{\rightarrow 0} + \underbrace{e^{-\frac{2}{\varepsilon}}}_{\rightarrow 0} + \underbrace{\varepsilon e^{-\frac{2}{\varepsilon}}}_{\rightarrow 0} \right] \right| \xrightarrow{\varepsilon \rightarrow 0} 0
 \end{aligned}$$

und somit

$$\begin{aligned}
 & \Rightarrow \int_0^1 |u(x, \varepsilon) - x|^2 dx \in \mathcal{O}(\varepsilon) \\
 & \Rightarrow \|u(x, \varepsilon) - x\|_2 = \left( \int_0^1 |u(x, \varepsilon) - x|^2 dx \right)^{\frac{1}{2}} \in \mathcal{O}(\varepsilon^{\frac{1}{2}}).
 \end{aligned}$$

□

Wir beabsichtigen nun die Neumann-Neumann Iteration für die Konvektions-Diffusionsgleichung mit Hilfe der vorangegangenen Notation zu beschreiben. Hierbei nehmen wir, wie zuvor, eine Zerlegung von  $\Omega$  in Dreiecke  $\mathcal{T}_h$  an und geben einen Startwert

$$\forall \Gamma_i \in \mathcal{E}_h^\circ : \quad u_{\Gamma_i}^{(0)} = 0$$

vor. Damit können wir die Neumann-Neumann Iteration wie folgt beschreiben.

Sei  $n = 0$ :

1. Auf jedem  $T_i \in \mathcal{T}_h$  wird das folgendes Dirichlet-Problem gelöst:

$$\begin{aligned}
 & -\varepsilon \Delta u_{T_i}^{(n)} + b \nabla u_{T_i}^{(n)} + c u_{T_i}^{(n)} = f \text{ in } T_i, \\
 \forall E \in \mathcal{E}_h^{\partial\Omega}(T_i) : & \quad u_{T_i}^{(n)} = g \text{ auf } E, \\
 \forall \Gamma_i \in \mathcal{E}_h^\circ(T_i) : & \quad u_{T_i}^{(n)} = u_{\Gamma_i}^{(n)} \text{ auf } \Gamma_i \setminus \mathcal{N}_h^\circ(T_i), \\
 \forall p_i^j \in \mathcal{N}_h^\circ(T_i) : & \quad -\varepsilon \Delta u_{T_i}^{(n)}(p_i^j) + b \nabla u_{T_i}^{(n)}(p_i^j) + c u_{T_i}^{(n)}(p_i^j) = f(p_i^j).
 \end{aligned}$$

2. Auf jedem  $T_i \in \mathcal{T}_h$  wird das folgendes Neumann-Problem gelöst:

$$\begin{aligned} \forall E \in \mathcal{E}_h^{\partial\Omega}(T_i) : \quad & -\varepsilon \Delta \psi_{T_i}^{(n)} + b \nabla \psi_{T_i}^{(n)} + c \psi_{T_i}^{(n)} = 0 \text{ in } T_i, \\ & \psi_{T_i}^{(n)} = 0 \text{ auf } E, \\ \forall \Gamma_i \in \mathcal{E}_h^\circ(T_i) : \quad & \psi_{T_i}^{(n)} = \frac{\partial u_{T_i}^{(n)}}{\partial n_i} - \frac{\partial u_{T_j}^{(n)}}{\partial n_i} \text{ auf } \Gamma_i \setminus \mathcal{N}_h^\circ(T_i), \\ \forall p_i^j \in \mathcal{N}_h^\circ(T_i) : \quad & -\varepsilon \Delta \psi_{T_i}^{(n)}(p_i^j) + b \nabla \psi_{T_i}^{(n)}(p_i^j) + c \psi_{T_i}^{(n)}(p_i^j) = 0. \end{aligned}$$

Hier ist  $T_j$  das an  $\Gamma_i$  angrenzende Dreieck mit  $T_i \cap T_j = \Gamma_i$ .

3. Prüfe das Abbruchkriterium:

$$\forall \Gamma_i \in \mathcal{E}_h^\circ : \quad \|(\psi_{T_i}^{(n)} + \psi_{T_j}^{(n)})|_{\Gamma_i}\|_{l_1(\Gamma_i)} \leq \varepsilon_\Gamma, \text{ wobei } T_i, T_j \in \mathcal{T}_h : T_i \cap T_j = \Gamma_i.$$

Sollte das Abbruchkriterium auf alle  $\Gamma_i$  erfüllt sein, dann kann die Iteration abgebrochen werden.

4. Update  $u_{\Gamma_i}^{(n+1)}$ :

$$\forall \Gamma_i \in \mathcal{E}_h^\circ : \quad u_{\Gamma_i}^{(n+1)} = u_{\Gamma_i}^{(n)} - \theta \left( \psi_{T_i}^{(n)} + \psi_{T_j}^{(n)} \right), \text{ wobei } T_i, T_j \in \mathcal{T}_h : T_i \cap T_j = \Gamma_i.$$

Setze  $n = n + 1$  und gehe wieder zu Schritt 1.

### 7.2.2.1 Moderater Boundary-Layer

Wir planen jetzt singular-gestörte Probleme nach der Definition im vorherigen Kapitel numerisch mit Hilfe der Neumann-Neumann-Iteration zu lösen. Wir verwenden dazu Gleichung (7.4).

Als erstes Beispiel betrachten wir folgendes Problem:

$$u(x, y) = \exp\left(-\frac{(x-1)(y-1)}{\varepsilon}\right), \quad (x, y) \in \Omega$$

mit

$$b = (1, 1)^t, \quad c = 1, \quad \varepsilon = 0.1$$

und der resultierenden rechten Seite

$$f(x, y) = -\frac{1}{\varepsilon}(x(x-1) - \varepsilon + y(y-1))u(x, y).$$

Die Lösung  $u$  ist auf  $\Omega$  nah bei 0 und steigt am rechten Rand auf 1 an.

Wir verwenden hier  $\theta = 0.1$  als Parameter für die Iteration.

Die Abbildungen 7.14 und 7.15 zeigen die unterschiedlichen numerischen Lösungen für die Fälle  $nT = 8, N = 6$  und  $nT = 32, N = 12$ . Optisch sehen beide numerischen Lösungen qualitativ identisch aus, jedoch unterscheiden sich diese bei näherer Betrachtung des Fehlers in Tabelle 7.2. In unseren numerischen Versuchen prüfen wir die Auswirkung verschiedener Abbruchkriterien  $\varepsilon_\Gamma$ , Anzahl an Dreiecken und Polynomgrade.

Je größer  $\varepsilon_\Gamma$  gewählt wird, desto weniger Iterationen werden benötigt, um das Abbruchkriterium zu erfüllen. Bei Betrachtung der Fälle kleinster Ordnung, hier  $nT = 8, N = 6$ , für  $\varepsilon_\Gamma = 1e^{-06}$  und  $\varepsilon_\Gamma = 1e^{-12}$ , stellen wir fest, dass es keine Auswirkung auf den Fehler hat, aber auf die Anzahl der Iterationen, die im Fall  $\varepsilon_\Gamma = 1e^{-12}$  doppelt so hoch sind wie im Fall  $\varepsilon_\Gamma = 1e^{-06}$ . Daraus folgern wir, dass es nur lohnenswert ist, kleine  $\varepsilon_\Gamma$  zu wählen, wenn wir annehmen können, dass der Fehler auf den Interfaces dominieren könnte und bessere Approximationen möglich sind. Einen solchen Fall sehen wir für  $\varepsilon_\Gamma = 1e^{-06}$  und  $\varepsilon_\Gamma = 1e^{-12}$  in Verbindung mit  $nT = 32, N = 15$ . Hier liegt der Fehler für  $\varepsilon_\Gamma = 1e^{-06}$  bei  $\approx 10^{-10}$ , der Fehler auf den Interfaces dominiert den globalen Fehler. Für  $\varepsilon_\Gamma = 1e^{-12}$  bemerken wir einen Fehler von  $\approx 10^{-16}$ , hier ist der Interface Fehler nicht mehr bestimmend und es kann annähernd Maschinengenauigkeit erreicht werden.

Der Einfluss der Anzahl der Dreiecke wird beispielhaft am Datensatz von  $\varepsilon = 1e^{-10}$  diskutiert. Wir beobachten einen exponentiellen Abfall des Fehlers mit Zunahme der Kollokationspunkte in direkter Abhängigkeit von der Anzahl der Dreiecke. Zum Schluss wollen wir den Wirkung des Polynomgrades untersuchen. Dazu prüfen wir exemplarisch den Fall  $\varepsilon = 1e^{-10}$  und  $nT = 18$ . Hier stellen wir ein exponentiellen Abfall des Fehlers in Abhängigkeit zum Polynomgrad fest.

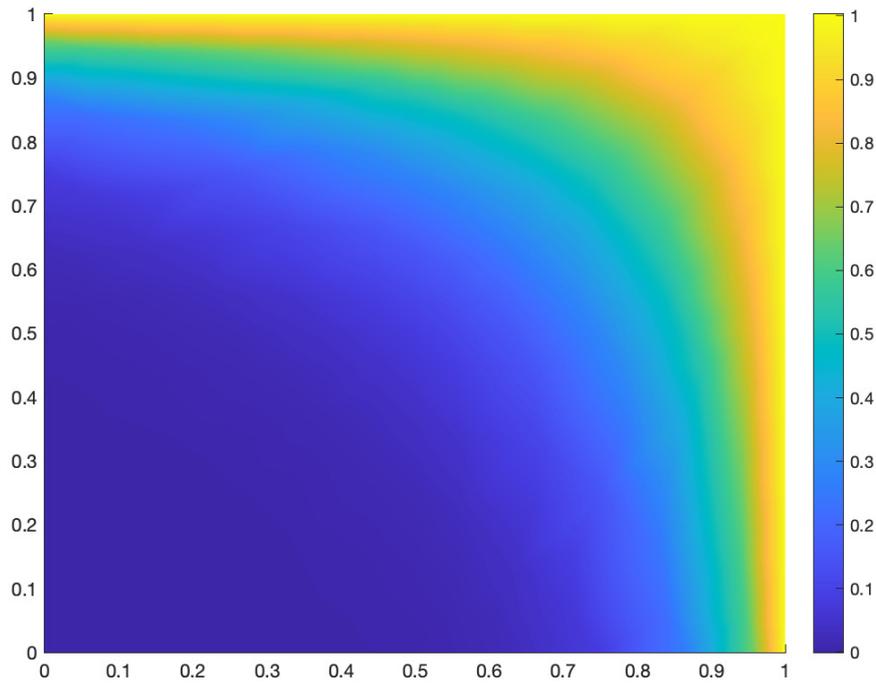


Abbildung 7.14: Lösung:  $nT = 8$ ,  $N = 6$

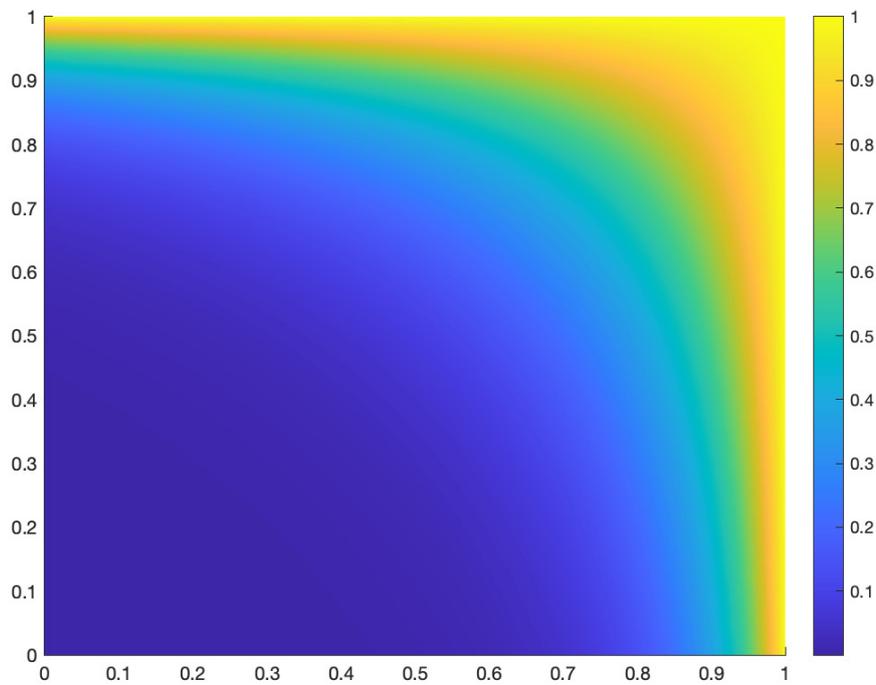


Abbildung 7.15: Lösung  $nT = 32$ ,  $N = 12$

Tabelle 7.2: Ergebnisse moderater Boundary-Layer

$\varepsilon_\Gamma$	$nT$	N	Kollokationspunkte	Fehler	It	Zeit (sek)
$1e^{-06}$	8	6	224	4.52e-06	141	0.33
		9	440	5.65e-09	205	0.61
		12	728	6.96e-10	367	1.69
		15	1088	5.00e-10	417	3.12
	18	6	504	1.71e-07	223	1.05
		9	990	3.91e-10	313	2.09
		12	1638	3.02e-10	526	5.10
		15	2448	2.14e-10	614	10.29
	32	6	896	1.76e-08	344	2.69
		9	1760	2.29e-10	457	5.02
		12	2912	1.94e-10	695	11.48
		15	4352	1.36e-10	803	24.02
	50	6	1400	3.08e-09	485	5.55
		9	2750	2.00e-10	624	9.93
		12	4550	1.51e-10	899	23.98
		15	6800	1.01e-10	1028	49.58
$1e^{-08}$	8	6	224	4.52e-06	189	0.41
		9	440	5.81e-09	298	0.86
		12	728	1.36e-11	520	2.25
		15	1088	5.05e-12	607	4.46
	18	6	504	1.71e-07	293	1.38
		9	990	6.23e-11	437	2.90
		12	1638	2.96e-12	725	6.75
		15	2448	2.12e-12	857	14.14
	32	6	896	1.75e-08	457	3.34
		9	1760	4.03e-12	619	6.35
		12	2912	1.90e-12	950	15.93
		15	4352	1.33e-12	1107	32.92
	50	6	1400	2.96e-09	652	7.00
		9	2750	1.96e-12	845	13.16
		12	4550	1.50e-12	1227	33.11
		15	6800	1.00e-12	1409	67.25
$1e^{-10}$	8	6	224	4.52e-06	246	0.52
		9	440	5.81e-09	391	1.13
		12	728	9.48e-12	674	3.28

$\varepsilon_\Gamma$	$nT$	N	Kollokationspunkte	Fehler	It	Zeit (sek)
		15	1088	5.11e-14	798	6.02
	18	6	504	1.71e-07	373	1.77
		9	990	6.12e-11	555	3.68
		12	1638	4.77e-14	923	9.17
		15	2448	2.14e-14	1099	19.80
	32	6	896	1.75e-08	570	4.76
		9	1760	3.45e-12	773	9.47
		12	2912	1.90e-14	1204	21.89
		15	4352	1.33e-14	1409	43.99
	50	6	1400	2.96e-09	819	9.91
		9	2750	3.83e-13	1064	18.29
		12	4550	1.49e-14	1555	42.08
		15	6800	1.00e-14	1789	83.98
$1e^{-12}$	8	6	224	4.52e-06	302	0.65
		9	440	5.81e-09	485	1.77
		12	728	9.45e-12	829	4.19
		15	1088	7.84e-15	991	7.29
	18	6	504	1.71e-07	459	2.26
		9	990	6.12e-11	667	4.73
		12	1638	3.43e-14	1126	12.56
		15	2448	2.21e-16	1341	23.68
	32	6	896	1.75e-08	685	5.77
		9	1760	3.45e-12	912	10.74
		12	2912	7.83e-16	1461	25.18
		15	4352	1.37e-16	1709	53.08
	50	6	1400	2.96e-09	988	12.82
		9	2750	3.85e-13	1281	23.25
		12	4550	1.45e-16	1887	50.14
		15	6800	9.77e-17	2171	103.57

### 7.2.2.2 Verschärfter Boundary-Layer

In diesem Beispiel betrachten wir wieder folgendes Problem:

$$u(x, y) = \exp\left(-\frac{(x-1)(y-1)}{\varepsilon}\right), \quad (x, y) \in \Omega$$

mit

$$b = (1, 1)^t, \quad c = 1,$$

und der resultierenden rechten Seite

$$f(x, y) = -\frac{1}{\varepsilon}(x(x-1) - \varepsilon + y(y-1))u(x, y).$$

Aber diesmal wählen wir

$$\varepsilon = 0.01,$$

damit haben wir eine stärkere singuläre Störung. Die Lösung  $u$  ist nah bei 0 und steigt am rechten und oberen Rand sehr schnell auf 1 an.

Wir verwenden hier  $\theta = 0.01$  als Parameter für die Iteration.

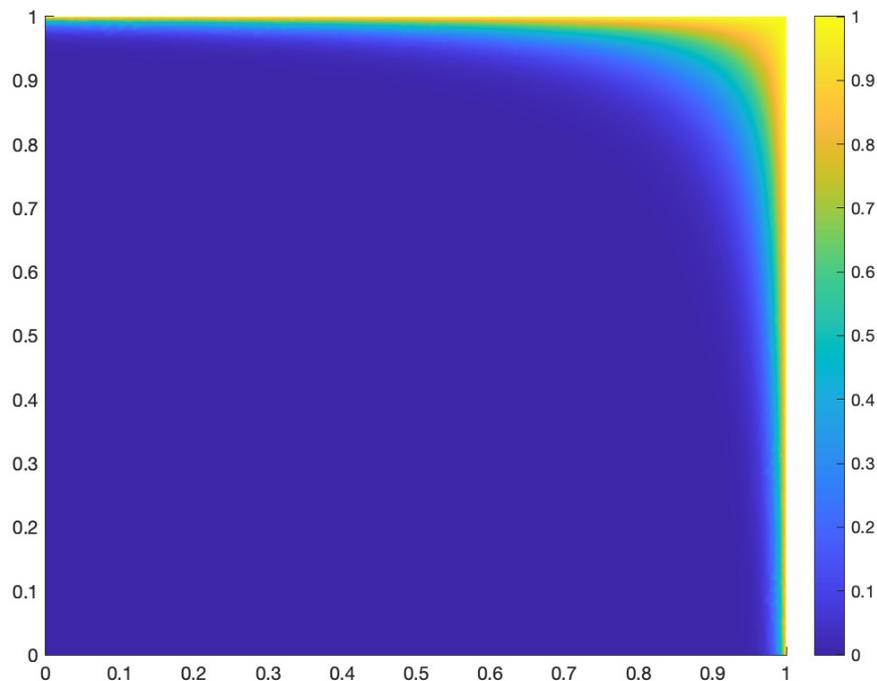
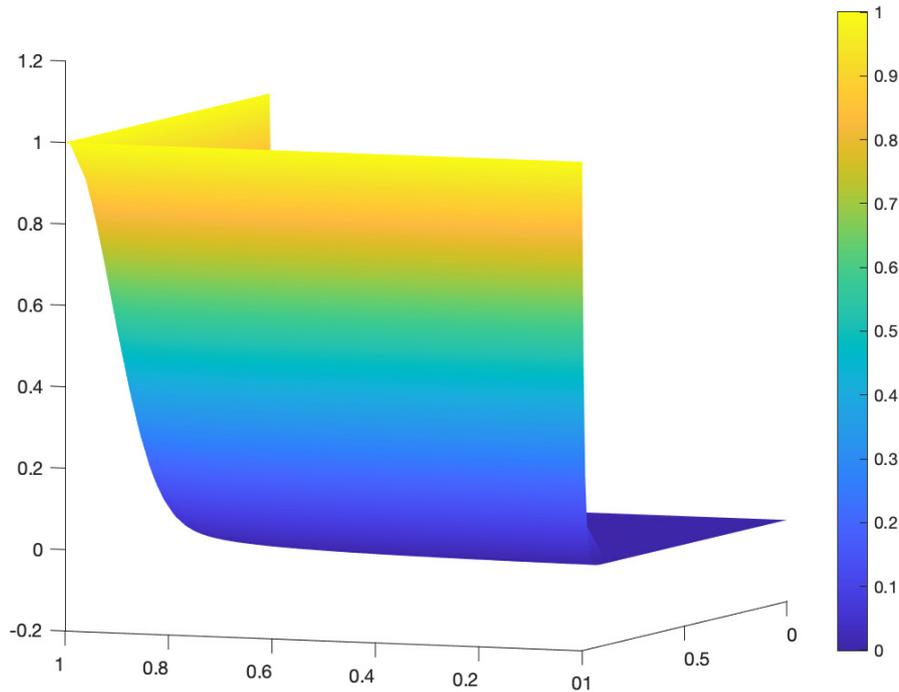


Abbildung 7.16: Lösung:  $nT = 50$ ,  $N = 18$

Abbildung 7.17: Lösung  $nT = 50$ ,  $N = 18$ 

Die Abbildungen 7.16 und 7.17 zeigen die numerischen Lösungen aus verschiedenen Perspektiven für den Fall  $nT = 50$ ,  $N = 18$ . Wir untersuchen die Auswirkung von verschiedenen Abbruchkriterien  $\varepsilon_\Gamma$ , Anzahl an Dreiecken und unterschiedliche Polynomgrade. Je größer wir  $\varepsilon_\Gamma$  wählen, desto weniger Iterationen werden benötigt. Vergleichen wir die Fälle kleinster Ordnung, hier  $nT = 18$ ,  $N = 12$ , für  $\varepsilon_\Gamma = 1e^{-04}$  und  $\varepsilon_\Gamma = 1e^{-12}$ , sehen wir, dass es keine Auswirkung auf den Fehler hat, aber auf die Anzahl der Iterationen, die im Fall  $\varepsilon_\Gamma = 1e^{-12}$  mehr als dreimal so hoch sind wie für  $\varepsilon_\Gamma = 1e^{-04}$  (vgl. Tabelle 7.3). Wir ziehen daraus den Schluss, dass es nur lohnenswert ist, kleine  $\varepsilon_\Gamma$  zu wählen, wenn wir annehmen können, dass der Fehler auf den Interfaces dominieren könnte und bessere Approximationen möglich sind. Dies bemerken wir für  $\varepsilon_\Gamma = 1e^{-04}$  und  $\varepsilon_\Gamma = 1e^{-12}$  in Verbindung mit  $nT = 50$ ,  $N = 18$ . Hier liegt der Fehler für  $\varepsilon_\Gamma = 1e^{-04}$  bei  $\approx 10^{-09}$ , der Fehler auf den Interfaces dominiert den globalen Fehler. Für  $\varepsilon_\Gamma = 1e^{-12}$  beobachten wir einen Fehler von  $\approx 10^{-12}$ , hier ist der Interface Fehler nicht mehr gravierend und es kann annähernd Maschinengenauigkeit erreicht werden.

Der Einfluss der Anzahl der Dreiecke wird exemplarisch am Datensatz von  $\varepsilon = 1e^{-12}$  diskutiert. Wir beobachten einen Abfall des Fehlers mit Zunahme der Kollokationspunkte in direkter Abhängigkeit der Anzahl der Dreiecke. Schließlich begutachten wir noch den Einfluss des Polynomgrades. Hierfür prüfen wir exemplarisch den

Fall  $\varepsilon = 1e^{-12}$  und  $nT = 18$ . Wir stellen ein exponentiellen Abfall des Fehlers in Abhängigkeit zum Polynomgrad fest.

Tabelle 7.3: Ergebnisse verschärfter Boundary-Layer

$\varepsilon_T$	$nT$	N	Kollokationspunkte	Fehler	It	Zeit (sek)
$1e^{-04}$	18	12	1638	1.66e-06	850	8.98
		15	2448	3.01e-08	913	15.71
		18	3420	3.85e-09	1187	33.02
	32	12	2912	1.45e-07	973	18.75
		15	4352	4.10e-09	937	29.23
		18	6080	3.66e-09	992	51.70
	50	12	4550	2.21e-08	991	30.84
		15	6800	2.32e-09	995	48.35
		18	9500	1.06e-09	1317	103.51
$1e^{-08}$	18	12	1638	1.66e-06	1829	19.18
		15	2448	3.06e-08	2318	39.92
		18	3420	1.57e-09	2981	81.76
	32	12	2912	1.46e-07	2257	40.29
		15	4352	1.75e-09	2699	82.40
		18	6080	6.65e-11	3217	159.70
	50	12	4550	2.27e-08	2438	75.72
		15	6800	1.64e-10	2898	143.62
		18	9500	4.33e-12	3583	280.52
$1e^{-12}$	18	12	1638	1.66e-06	2864	30.57
		15	2448	3.06e-08	3707	64.51
		18	3420	1.57e-09	4792	139.01
	32	12	2912	1.46e-07	3697	74.33
		15	4352	1.75e-09	4511	146.00
		18	6080	6.65e-11	5399	290.05
	50	12	4550	2.27e-08	4032	138.84
		15	6800	1.64e-10	4996	266.62
		18	9500	4.32e-12	6015	479.64

## 7.2.2.3 Eckenlayer

In diesem Beispiel untersuchen wir ein Eckenlayer-Problem:

$$u(x, y) = \exp\left(-\frac{x^6 + y^6}{\varepsilon}\right), \quad (x, y) \in \Omega$$

mit

$$b = \left(\frac{1}{4} - x, \frac{1}{4} - y\right)^t, \quad c = 1, \quad \varepsilon = 0.005$$

und der resultierenden rechten Seite

$$f(x, y) = \left[ -36(x^{10} + y^{10}) - \frac{3}{2}(x^5 + y^5) + 6(x^6 + y^6) + (30(x^4 + y^4) + 1)\varepsilon \right] \frac{u}{\varepsilon}.$$

Die Lösung  $u$  ist größtenteils auf  $\Omega$  nah bei 0 und steigt zur Ecke  $(0, 0)$  sehr steil auf.

Wir verwenden die folgenden Parameter in der Iteration

$$\varepsilon_{\Gamma} = 1e^{-05}, \quad \theta = 0.01.$$

Die Abbildungen 7.18 und 7.18 zeigen die numerischen Lösungen aus verschiedenen Perspektiven für den Fall  $nT = 32$ ,  $N = 18$ . In diesem Fall untersuchen wir die Tragweite der verschiedenen Anzahl an Dreiecken und unterschiedlichen Polynomgrade.

Wir erkennen auch in diesem Fall ähnliche Muster, wie in dem vorangegangenen Fällen des Boundary-Layers, dennoch können zwei Punkte herausgestellt werden. Zum einen konvergiert das Problem nicht zureichend auf den Interfaces, weshalb  $\varepsilon_{\Gamma} = 1e^{-05}$  als einziger Fall betrachtet wurde, zum anderen wurde eine reduzierte Anzahl an Dreiecken betrachtet. Wir bemerken, dass für  $nT = 32$  der Fehler nur minimal abnimmt, was auf das relativ große  $\varepsilon_{\Gamma}$  zurückzuführen ist. Im Allgemeinen beobachten wir einen Abfall des Fehlers mit Zunahme der Kollokationspunkte in direkter Abhängigkeit der Anzahl der Dreiecke (vgl. Tabelle 7.4). Ebenso stellen wir einen exponentiellen Abfall des Fehlers in Abhängigkeit zum Polynomgrad fest.

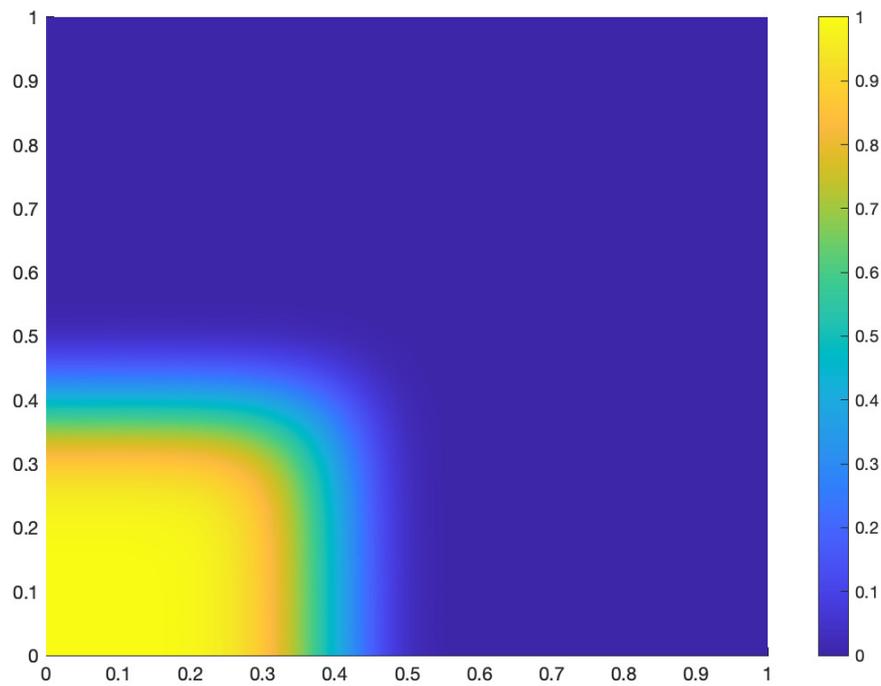


Abbildung 7.18: Lösung:  $nT = 32$ ,  $N = 18$

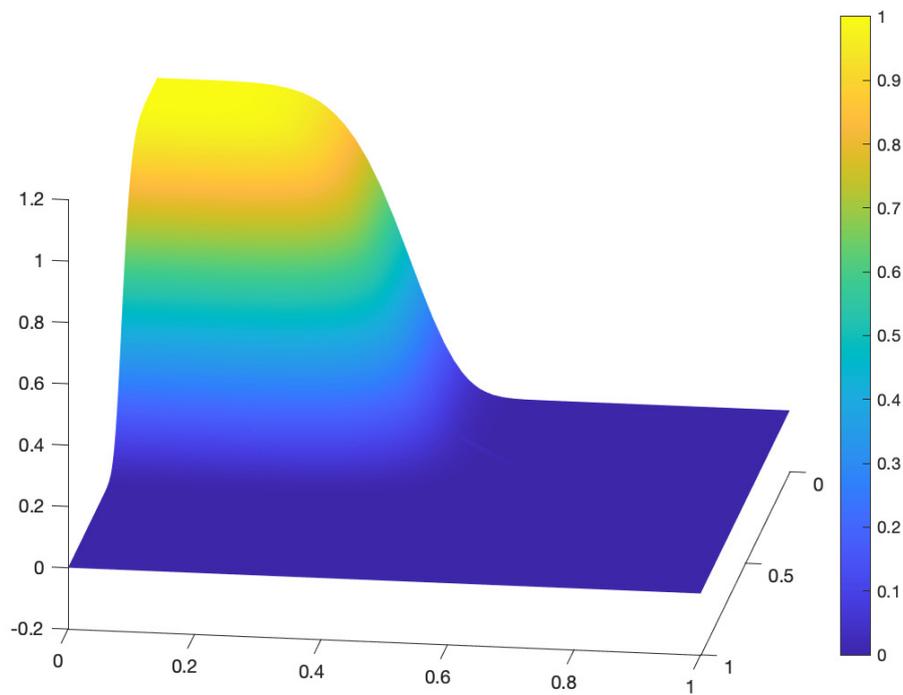


Abbildung 7.19: Lösung  $nT = 32$ ,  $N = 18$

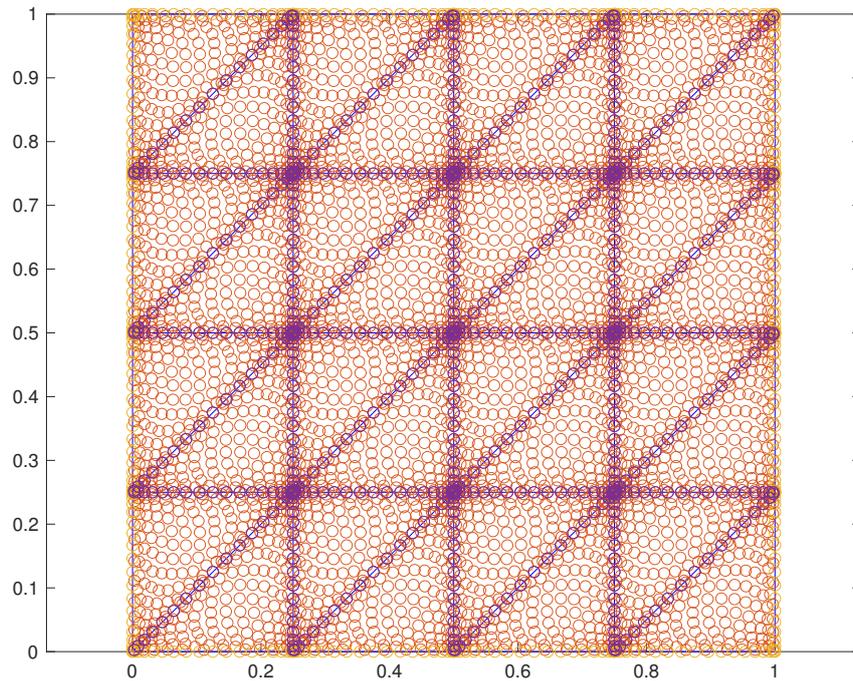
Abbildung 7.20: Gitter  $nT = 32$ ,  $N = 18$ 

Tabelle 7.4: Ergebnisse Ecken-Layer

$nT$	$N$	Kollokationspunkte	Fehler	It	Zeit (sek)
8	12	728	3.23e-07	855	3.47
	15	1088	1.93e-08	637	4.90
	18	1520	3.01e-09	1308	15.96
18	12	1638	4.88e-08	2469	25.83
	15	2448	1.85e-09	3099	51.43
	18	3420	8.35e-10	3553	98.75
32	12	2912	9.76e-10	3193	60.89
	15	4352	5.86e-10	3906	118.87
	18	6080	4.08e-10	4408	228.41

#### 7.2.2.4 Hut-Layer

Als letztes Beispiel betrachten wir einen Hutlayer. Im Inneren des Gebietes ist eine kreisrunde Region, innerhalb dieser kreisrunden Region besteht ein starker Anstieg inklusive eines Plateaus, außerhalb ist die Lösung beinahe konstant, dadurch bildet sich eine fast unstetige Kante zwischen den Bereichen innerhalb und außerhalb der

Kreisregion. An den Rändern ist die Lösung  $u$  konstant  $-1$  und steigt zur Mitte hin auf  $1$ .

Unser Gleichung sieht wie folgt aus:

$$\begin{aligned} -\varepsilon \Delta u + u &= f \text{ in } \Omega \\ u &= g \text{ auf } \partial\Omega \end{aligned}$$

Wir wählen

$$u(x, y) = \tanh\left(\frac{\frac{1}{16} - (x - 0.5)^2 - (y - 0.5)^2}{\varepsilon}\right), \quad (x, y) \in \Omega$$

mit  $\varepsilon = 0.01$  und resultierender rechter Seite

$$f(x, y) = 4 + u(x, y) - 4(u(x, y))^2 + \frac{8}{\varepsilon} \left( (x - 0.5)^2 + (y - 0.5)^2 \right) (u(x, y) - (u(x, y))^3).$$

Wir verwenden hier  $\theta = 0.01$  als Parameter für die Iteration.

Die Abbildungen 7.21 und 7.24 zeigen die numerischen Lösungen für die Fälle  $nT = 8, N = 15$  und  $nT = 162, N = 15$ . Im Fall  $nT = 8, N = 15$  erkennen wir deutlich, wie unscharf die Umrandung des Hut-Layers in Abbildung 7.21 ist. Es lässt sich nur der Kreis des Hut-Layers vermuten. Wir kommen zu dem Schluss, dass wir entsprechend mehr Dreiecke oder höhere Polynomgrade für eine bessere Approximation der Lösung benötigen. Wir überprüfen hier die Auswirkung von verschiedenen Abbruchkriterien  $\varepsilon_\Gamma$ , Anzahl an Dreiecken und unterschiedlichen Polynomgraden. Je größer wir  $\varepsilon_\Gamma$  wählen, desto weniger Iterationen werden benötigt. Vergleichen wir die Fälle kleinster Ordnung, hier  $nT = 8, N = 6$ , für  $\varepsilon_\Gamma = 1e^{-04}$  und  $\varepsilon_\Gamma = 1e^{-12}$ , sehen wir, dass es keine Auswirkung auf den Fehler hat, jedoch auf die Anzahl der Iterationen, die im Fall  $\varepsilon_\Gamma = 1e^{-12}$  mehr als dreimal so hoch sind wie für  $\varepsilon_\Gamma = 1e^{-04}$  (vgl. Tabelle 7.5). Deshalb schlussfolgern wir, dass es nur effizient ist, kleine  $\varepsilon_\Gamma$  zu wählen, wenn wir annehmen können, dass der Fehler auf den Interfaces dominieren könnte und bessere Approximationen möglich sind. Wir betrachten den Fall  $\varepsilon_\Gamma = 1e^{-04}$  und  $\varepsilon_\Gamma = 1e^{-12}$  in Verbindung mit  $nT = 50, N = 15$ , hier liegt der Fehler für  $\varepsilon_\Gamma = 1e^{-04}$  bei  $\approx 10^{-05}$  und für  $\varepsilon_\Gamma = 1e^{-12}$  liegt der Fehler ebenfalls bei  $\approx 10^{-05}$ . Die Verringerung von  $\varepsilon_\Gamma$  hat wenig Einfluss auf den Gesamtfehler, da die Approximationsgüte schon fast erreicht ist.

Der Einfluss der Anzahl der Dreiecke wird beispielhaft am Datensatz von  $\varepsilon = 1e^{-12}$  diskutiert. Wir beobachten einen Abfall des Fehlers mit Zunahme der Kollokationspunkte in direkter Abhängigkeit der Anzahl der Dreiecke. Schließlich begutachten

wir noch den Einfluss des Polynomgrades. Hierfür prüfen wir exemplarisch den Fall  $\varepsilon = 1^{-12}$  und  $nT = 18$ . Wir stellen ein exponentiellen Abfall des Fehlers in Abhängigkeit zum Polynomgrad fest.

Tabelle 7.5: Ergebnisse Hut-Layer

$\varepsilon_\Gamma$	$nT$	N	Kollokationspunkte	Fehler	It	Zeit (sek)	
$1e^{-04}$	8	6	224	8.39e-03	81	0.23	
		9	440	2.63e-03	197	0.66	
		12	728	2.08e-03	285	1.47	
		15	1088	5.41e-04	350	2.85	
	18	6	504	2.91e-03	169	0.83	
		9	990	9.01e-04	222	1.64	
		12	1638	4.03e-04	329	3.38	
		15	2448	6.09e-05	368	6.69	
	32	6	896	5.87e-04	178	1.52	
		9	1760	5.99e-04	311	3.65	
		12	2912	9.22e-05	364	6.77	
		15	4352	4.48e-05	429	13.19	
	50	6	1400	5.12e-04	222	2.74	
		9	2750	3.85e-04	385	6.46	
		12	4550	4.29e-05	434	13.04	
		15	6800	1.23e-05	494	24.21	
	$1e^{-08}$	8	6	224	8.39e-03	190	0.44
			9	440	2.63e-03	381	1.15
			12	728	2.08e-03	560	2.65
			15	1088	5.41e-04	694	5.16
18		6	504	2.91e-03	319	1.50	
		9	990	9.01e-04	459	3.12	
		12	1638	4.03e-04	671	6.39	
		15	2448	6.10e-05	789	13.44	
32		6	896	5.87e-04	375	2.94	
		9	1760	5.99e-04	611	6.77	
		12	2912	9.23e-05	749	12.51	
		15	4352	4.48e-05	906	27.31	
50		6	1400	5.12e-04	479	5.48	
		9	2750	3.85e-04	762	12.86	
		12	4550	4.29e-05	955	26.90	

$\varepsilon_T$	$nT$	N	Kollokationspunkte	Fehler	It	Zeit (sek)
		15	6800	1.23e-05	1061	50.44
$1e^{-10}$	8	6	224	8.39e-03	244	0.58
		9	440	2.63e-03	474	1.63
		12	728	2.08e-03	697	3.52
		15	1088	5.41e-04	865	6.64
	18	6	504	2.91e-03	394	1.98
		9	990	9.01e-04	579	4.02
		12	1638	4.03e-04	842	9.61
		15	2448	6.10e-05	1001	17.57
	32	6	896	5.87e-04	477	4.14
		9	1760	5.99e-04	761	9.66
		12	2912	9.23e-05	948	19.58
		15	4352	4.48e-05	1151	36.15
	50	6	1400	5.12e-04	614	8.63
		9	2750	3.85e-04	952	18.90
		12	4550	4.29e-05	1217	40.34
		15	6800	1.23e-05	1378	74.58
$1e^{-12}$	8	6	224	8.39e-03	299	0.72
		9	440	2.63e-03	566	1.78
		12	728	2.08e-03	839	3.97
		15	1088	5.41e-04	1038	7.67
	18	6	504	2.91e-03	469	2.40
		9	990	9.01e-04	698	4.91
		12	1638	4.03e-04	1017	11.26
		15	2448	6.10e-05	1216	20.63
	32	6	896	5.87e-04	578	4.68
		9	1760	5.99e-04	913	11.01
		12	2912	9.23e-05	1155	23.06
		15	4352	4.48e-05	1402	43.18
	50	6	1400	5.12e-04	748	8.92
		9	2750	3.85e-04	1144	20.52
		12	4550	4.29e-05	1482	45.73
		15	6800	1.23e-05	1693	88.05
	162	15	22032	2.57e-07	3524	567.36

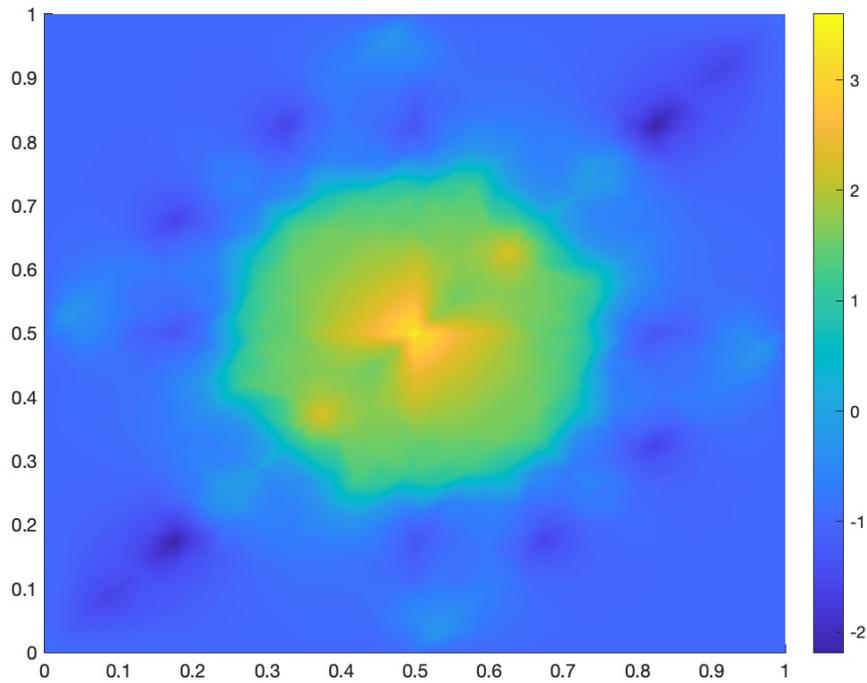


Abbildung 7.21: Lösung obige Ansicht:  $nT = 8$ ,  $N = 15$

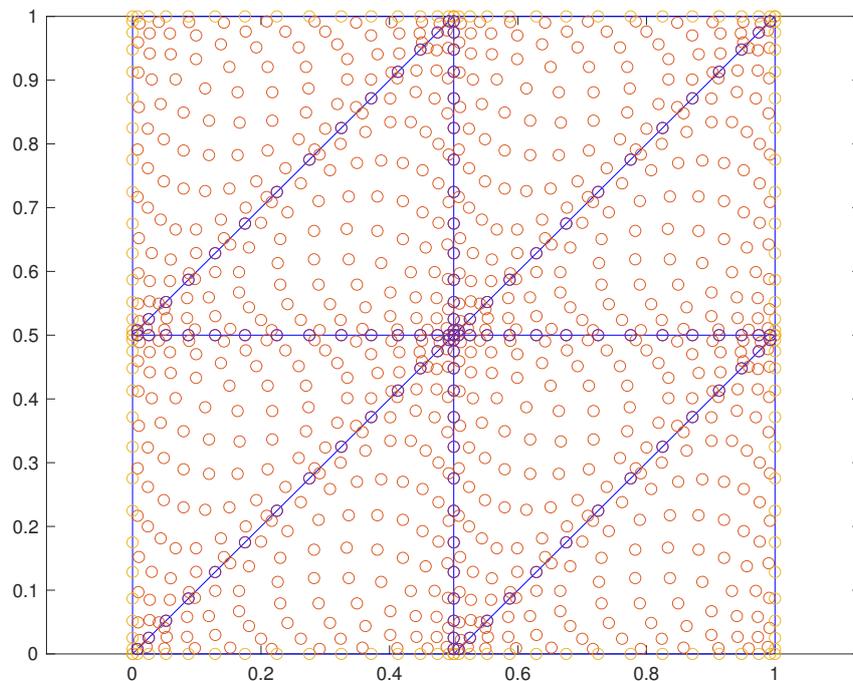


Abbildung 7.22: Gitter mit Kollokationspunkten für  $nT = 8$  und  $N = 15$

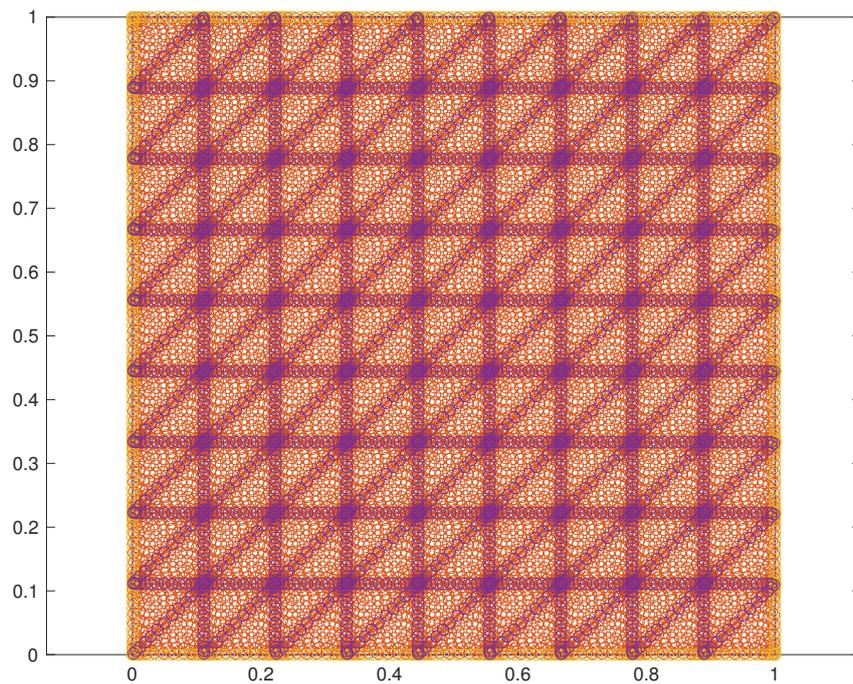


Abbildung 7.23: Gitter mit Kollokationspunkten für  $nT = 162$  und  $N = 15$

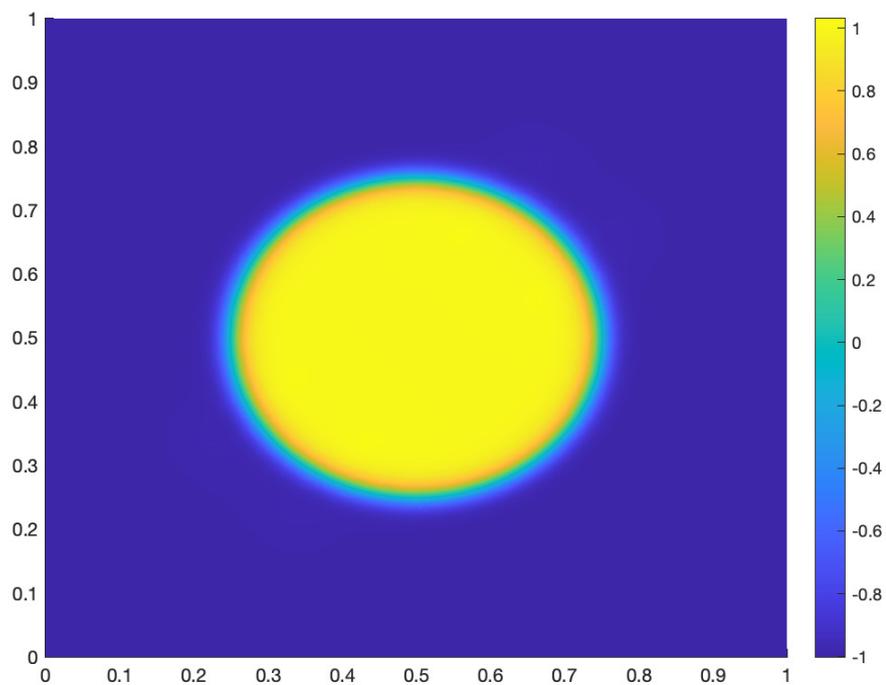
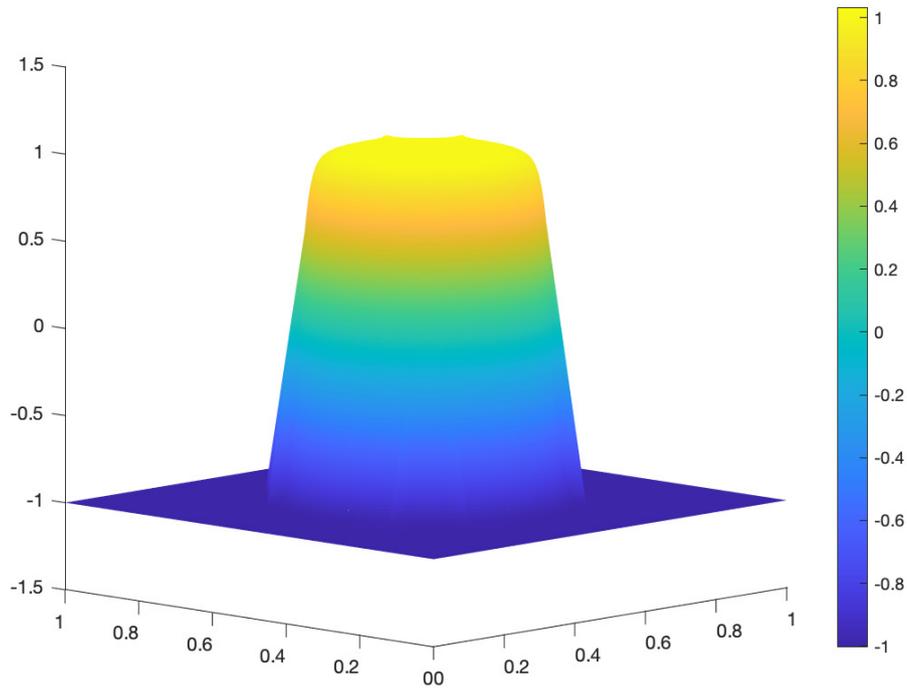


Abbildung 7.24: Lösung obige Ansicht:  $nT = 162$ ,  $N = 15$

Abbildung 7.25: Lösung seitliche Ansicht:  $nT = 162$ ,  $N = 15$ 

### 7.3 Adaptive $p$ -Verfeinerung

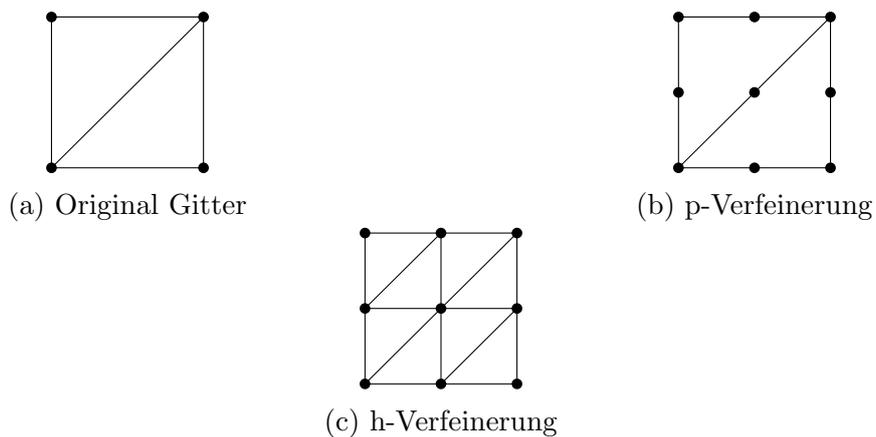


Abbildung 7.26: Gitter-Verfeinerungsmethoden

Wir haben die Gebiete aus 7.2.2, auf denen die Lösungen berechnet werden, durch Triangulierungen diskretisiert. Der Diskretisierungsfehler ist abhängig von der Feinheit der Diskretisierung, Feinheit bezüglich der Größe der Dreiecke sowie der Höhe

des Polynomgrades. Wir sehen anhand unserer numerischen Ergebnisse, dass bei zu grobem Gitter oder zu niedrigem Polynomgrad der Fehler größer ist. Wir müssen, um den Diskretisierungsfehler zu verringern, das Gitter verfeinern oder den Polynomgrad erhöhen. Jedoch müssen wir berücksichtigen, dass mit zunehmender Feinheit der Diskretisierung auch der Rechenaufwand steigt. Deshalb wollen wir den Polynomgrad nur dort erhöhen, wo der Fehler lokal am größten ist bzw. das Gitter nur dort verfeinern.

Um den lokalen Fehler zu schätzen, wird ein Fehlerindikator für jedes Element bestimmt. Beruhend auf diesem geschätzten lokalen Fehler werden die zu verfeinern- den Elemente ausgewählt. Die Menge der ausgewählten Elemente wird durch eine Markierungsstrategie bestimmt. Wir werden als Markierungsstrategie das Dörfler-Marking Kriterium verwenden. Dieses besagt, dass für einen gegebenen Fehlerindikator  $\eta(T)$  für alle  $T \in \mathcal{T}$  der Triangulierung  $\mathcal{T}$  und einem Markierungsparameter  $0 < \theta \leq 1$ , eine Menge  $\mathcal{M} \subseteq \mathcal{T}$  optimalerweise folgende Bedingung erfüllen soll [Dör96]:

$$\min_{\mathcal{M} \subseteq \mathcal{T}} \#\mathcal{M} : \quad \theta \sum_{T \in \mathcal{T}} \eta(T)^2 \leq \sum_{T \in \mathcal{M}} \eta(T)^2. \quad (7.6)$$

Nachdem die Elemente ausgewählt wurden, werden diese verfeinert.

### 7.3.1 Spektrale Erweiterungskoeffizienten

Wie zuvor beschrieben fehlt nun noch ein Fehlerindikator, dazu führen wir das folgende Kriterium ein. Dieses Kriterium schätzt den Approximationsfehler für jedes Element unter direkter Ausnutzung der polynomialen Basis höherer Ordnung [Mav92, Mav89].

Gegeben sei eine glatte Funktion  $u$  auf dem Standarddreieck  $T_0$ .

Die Entwicklung von  $u$  nach den Dubiner-Polynomen ist gegeben durch

$$u(x, y) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} a_{mn} g_{mn}(x, y).$$

Die Erweiterungskoeffizienten auf  $T_0$  sind gegeben durch

$$a_{mn} = \frac{1}{\gamma_{mn}} \int_{T_0} u(x, y) g_{mn}(x, y) dx dy$$

oder bezogen auf ein beliebiges Dreieck  $T_k \in \mathcal{T}_h$

$$a_{mn}^{(k)} = \frac{1}{\gamma_{mn}} \int_{T_k} u(x, y) g_{mn}(x, y) |J| dx dy$$

mit  $\gamma_{mn} = \langle g_{mn}, g_{mn} \rangle$  aus (4.1) und  $|J|$  aus (4.12).

Im ersten Schritt werden die zweidimensionalen Spekralkoeffizienten in ein eindimensionales transformiert:

$$\bar{a}_p = |a_{pp}| + \sum_{i=0}^{p-1} (|a_{ip}| + |a_{pi}|), \quad 0 \leq p \leq N.$$

Wir setzen  $a_{ij} = 0$ , falls  $i + j > N$ , da  $g_{ij}$  in diesem Fall nicht definiert ist.

In [Mav89] wird angeführt, dass der geschätzte Fehler  $\epsilon_{est}$  in der  $L^2$ -Norm durch

$$\epsilon_{est} \approx \left( a_N^2 + \sum_{n=N+1}^{\infty} \tilde{a}_n^2 \right)^{\frac{1}{2}}$$

gegeben ist. Da die Koeffizienten  $\tilde{a}_n$  für  $n \geq N + 1$  unbekannt sind, müssen diese aus den gegebenen Koeffizienten extrapoliert werden. Die Ergebnisse aus [Mav89] zeigen, dass das Spektrum für große  $n$  exponentiell abfällt. Daher bestimmen wir eine geeignetere Methode für das Spektrum  $\bar{a}_p$ :

$$a(n) \sim ce^{-\sigma n}.$$

Die Konstanten  $c$  and  $\sigma$  lassen sich leicht durch eine lineare Regression über das Spektrum  $\bar{a}_p$  berechnen. Für eine effiziente Berechnung hat es sich als ausreichend erwiesen, nur die letzten vier Punkte des Spektrums zu verwenden. Schließlich können wir entscheiden, ob ein Element verfeinert werden muss, wenn es die folgende Ungleichung nicht erfüllt:

$$\epsilon_{est} \approx \left( a(p)^2 + \int_{p+1}^{\infty} a(n)^2 dn \right)^{\frac{1}{2}} \leq \varepsilon \tag{7.7}$$

für eine gewisse Toleranz des geschätzten Fehlers  $\varepsilon$ .

Falls das Spektrum nicht zerfällt, das heißt  $\sigma \leq 0$ , ist das Element ebenfalls für die Verfeinerung markiert. Die Analysis dieses Kriteriums [Mav89] weist auf, dass dies ein effektives Verfeinerungskriterium unter der Einschränkung  $N \geq 6$  ist.

### 7.3.2 Koeffizienten entlang der Spur des Spektrums

Dieses Kriterium ist eine Vereinfachung der Methode aus dem vorigen Abschnitt. In [Hen99] wird vorgeschlagen, entlang der Spur des Spektrums zu summieren. Für eine gegebene Toleranz des geschätzten Fehlers  $\varepsilon$  ist das Verfeinerungskriterium gegeben durch

$$|a_{NN}| + \sum_{i=0}^{N-1} (|a_{iN}| + |a_{Ni}|) \leq \varepsilon$$

unter Verwendung der im vorigen Abschnitt eingeführten Erweiterungskoeffizienten. Im Fall der Dubiner-Polynome reduziert sich das Kriterium auf

$$|a_{0N}| + |a_{N0}| \leq \varepsilon.$$

Dieses Kriterium ist einfacher anzuwenden und Henderson hat gezeigt, dass es fast identische Ergebnisse liefert wie das Kriterium aus Abschnitt 7.3.1.

### 7.3.3 Adaptive $p$ -Verfeinerung der Kovektion-Diffusionsgleichung

In diesem Abschnitt wird für eine Triangulierung  $\mathcal{T}_h$  und für  $T \in \mathcal{T}_h$  mit einer Approximation  $u_T = (u_T^{(1)}, \dots, u_T^{(n)})$  an den Punkten  $x_T = (x_T^{(1)}, \dots, x_T^{(n)})$ ,  $x_T^{(j)} \in T, j \in \{1, \dots, n\}$  der Fehler der Approximation zu der exakten Lösung  $u_{\text{Exakt}}$  wie folgt definiert:

$$T \in \mathcal{T}_h : \quad E_T = \frac{1}{n_T} \sum_{i=1}^n \left( u_T^{(i)} - u_{\text{Exakt}}(x_T^{(i)}) \right)^2,$$

$$\text{Error} = \sqrt{\sum_{T \in \mathcal{T}_h} E_T},$$

wobei  $E_T$  den Fehler auf einem Dreieck  $T \in \mathcal{T}_h$  und Error den gesamten Fehler auf  $\Omega$  bezeichnet.

Als Verfeinerungskriterium wird das zuvor eingeführte Kriterium aus dem Abschnitt 7.3.2 angewandt. Hier findet die Betrachtung der Koeffizienten der Dubiner-Polynome statt, jedoch verwenden wir die Dörfler-Strategie (7.6) zur Auswahl der zu verfeinernden Dreiecke, anstatt ein festes  $\varepsilon$  als Schranke zur Verfeinerung vorzugeben. Als Folge werden die Koeffizienten entlang der Spur des Spektrums bestimmt und im Anschluss alle Dreiecke zur  $p$ -Verfeinerung markiert, welche den größten Anteil am gesamten aufsummierten Spektrum besitzen, frei nach der Dörfler-Strategie.

### 7.3.3.1 Moderate Boundary-Layer

In diesem Fall betrachten wir die gleiche Differentialgleichung wie in Abschnitt 7.2.2.1. Wir untersuchen dieses Problem speziell für  $nT = 32$  Dreiecke und starten mit Polynomgrad  $N = 6$ . Wir verwenden folgende Parameter in der Neumann-Neumann Iteration  $\varepsilon_{\Gamma} = 1e^{-09}$  und  $\theta = 0.1$ . Das Dörfler-Kriterium wird angewandt mit  $\theta_{\text{Dörfler}} = 0.9$ .

Die Triangulierung sowie die Kollokationspunkte zum Start (Level 1) sind in Abbildung 7.27 veranschaulicht.

Nach zwei  $p$ -Verfeinerungen erkennen wir die Tendenz der  $p$ -Verfeinerung entlang des oberen und rechten Randes. Werfen wir einen Blick auf die Approximation der Lösung auf Level 15 (Abbildung 7.30), stellen wir schnell fest, dass dies durchaus sinnvoll ist, da gerade entlang dieser Ränder die Lösung von Null verschieden ist. Dasselbe Bild zeichnet sich ebenfalls auf Level 10 (Abbildung 7.29) ab, hier werden Dreiecke des unteren linken Quadranten verfeinert, da diese nach der Verfeinerung des oberen und rechten Randes einen größeren Einfluss auf den Fehler nehmen und infolgedessen auch verfeinert werden.

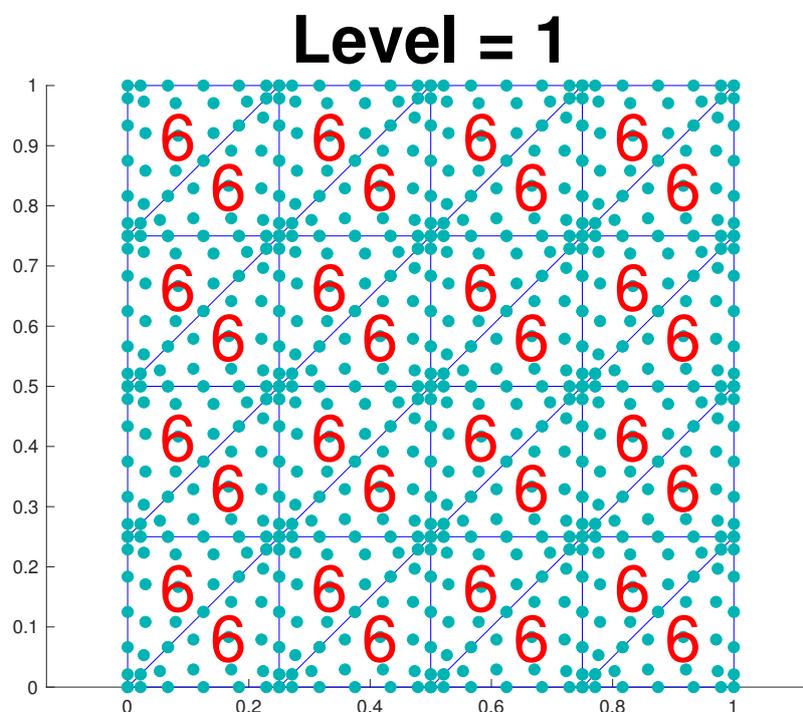


Abbildung 7.27:  $nT = 32$ , Polynomgrad entsprechend der roten Zahlen

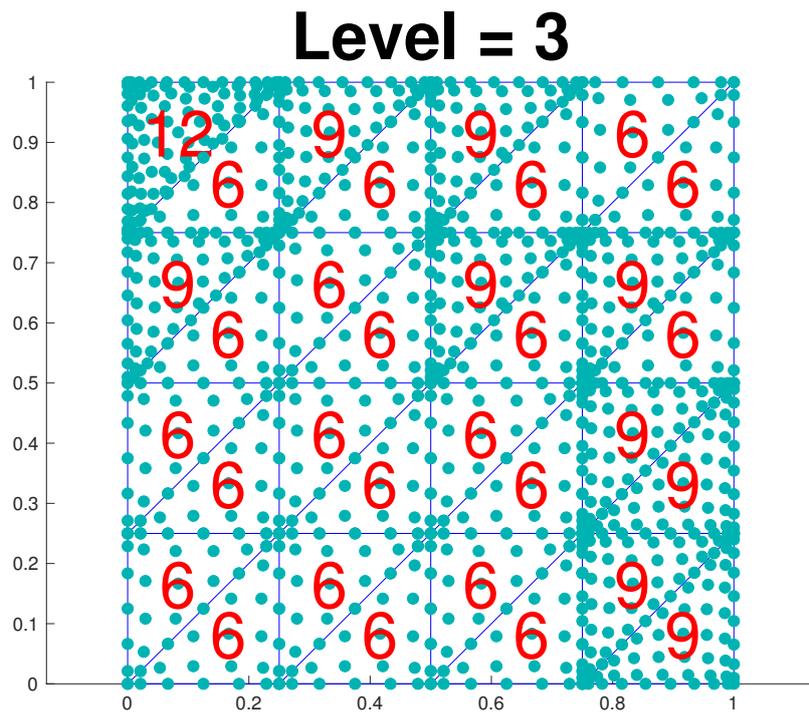


Abbildung 7.28:  $nT = 32$ , Polynomgrad entsprechend der roten Zahlen

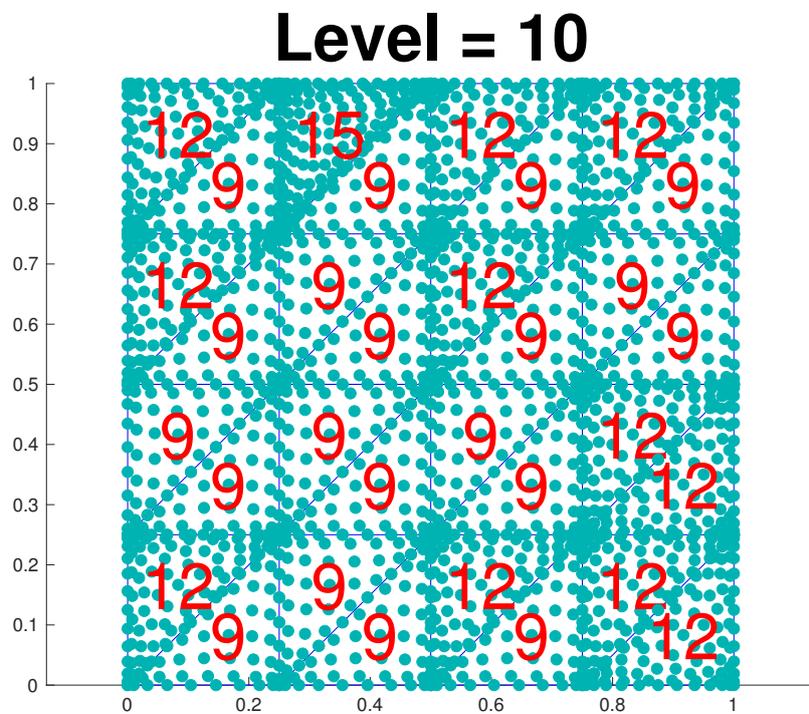


Abbildung 7.29:  $nT = 32$ , Polynomgrad entsprechend der roten Zahlen

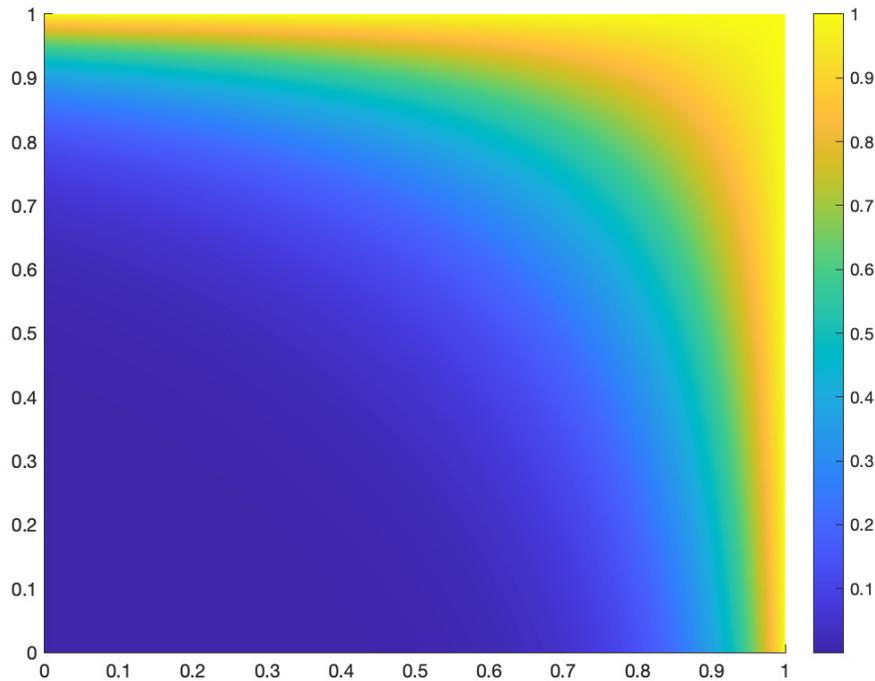


Abbildung 7.30: Approximation der Lösung auf Level 15

Untersuchen wir den Fehler in Abhängigkeit des Verfeinerungslevels und den resultierenden Kollokationspunkten, erkennen wir einen stark abfallenden Fehler mit steigender Anzahl der Kollokationspunkte, vorangetrieben durch einen höheren Polynomgrad, wie schon in den vorangegangenen Abschnitten. Die Ergebnisse sind in Tabelle 7.6 dargestellt. Wird zusätzlich eine LogLog-Darstellung angewandt, können wir unter Einsatz einer  $p$ -Verfeinerung einen exponentiellen Abfall des Fehlers feststellen (Abb. 7.31).

Tabelle 7.6: Ergebnisse moderater Boundary-Layer

Level	Kollokationspunkte	Fehler	Iterationen
1	896	8.88e-05	514
2	1004	6.89e-05	269
3	1202	5.83e-05	353
4	1400	4.16e-05	417
5	1580	1.22e-05	340
6	1715	3.65e-06	295
7	1796	9.77e-07	364
8	1913	9.15e-07	301
9	2084	9.33e-08	270

Level	Kollokationspunkte	Fehler	Iterationen
10	2237	3.18e-08	117
11	2426	3.42e-08	146
12	2570	2.21e-08	171
13	2750	1.13e-08	86
14	2939	2.48e-09	92
15	3011	1.32e-09	18

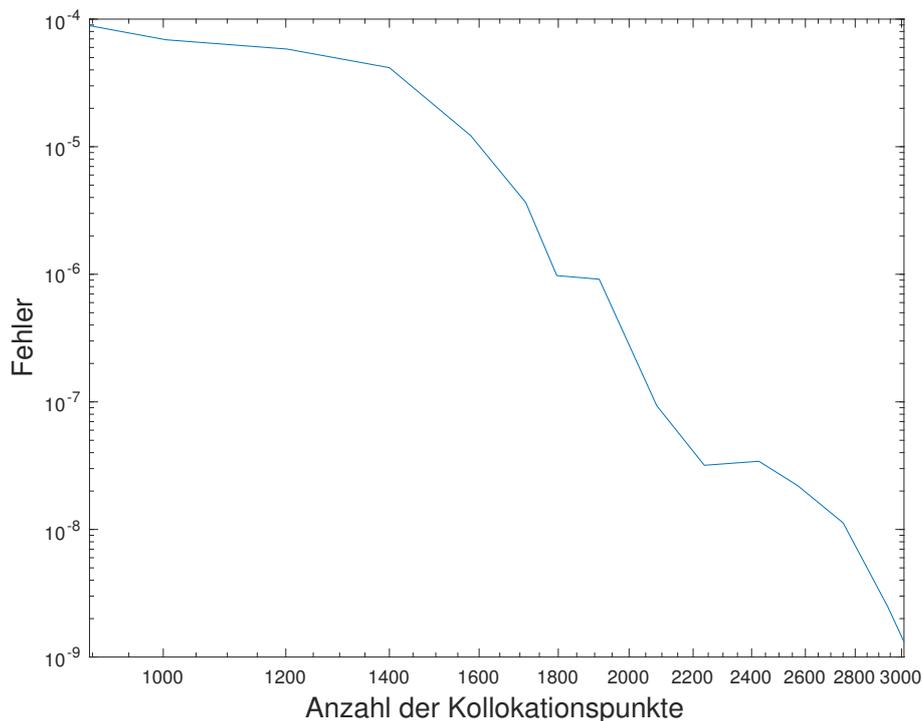


Abbildung 7.31: Konvergenzkurve

### 7.3.3.2 Verschärfter Boundary-Layer

Im Fall des verschärften Boundary-Layer Problems betrachten wir die gleiche Differentialgleichung wie in Abschnitt 7.2.2.2. Wir befassen uns hierbei speziell mit dem Fall  $nT = 32$  Dreiecke und starten mit uniformem Polynomgrad  $N = 12$ . Wir verwenden folgende Parameter in der Neumann-Neumann Iteration  $\varepsilon_{\Gamma} = 1e^{-09}$  und  $\theta = 0.01$ . Das Dörfler-Kriterium wird angewandt mit  $\theta_{\text{Dörfler}} = 0.9$ . Die Triangulierung sowie die Kollokationspunkte zum Start (Level 1) sind in Abb. 7.32 dargestellt. Wir beobachten eine Tendenz der  $p$ -Verfeinerung entlang des oberen und rechten Randes. Dies lässt sich an der Approximation der Lösung auf Level 15 (Abbildung 7.35) erklären, da gerade entlang dieser Ränder die Lösung von Null verschieden

ist. Dasselbe Bild zeichnet sich auch auf Level 10 (Abbildung 7.33) ab. Hier werden Dreiecke des oberen linken und unteren rechten Quadranten verfeinert, da diese nach der Verfeinerung des oberen und rechten Randes einen größeren Einfluss auf den Fehler nehmen und infolgedessen auch verfeinert werden.

Untersuchen wir den Fehler in Abhängigkeit des Verfeinerungslevels und den daraus resultierenden Kollokationspunkten, nehmen wir einen abfallenden Fehler mit steigender Anzahl der Kollokationspunkte, vorangetrieben durch einen höheren Polynomgrad, wahr. Die Ergebnisse sind in Tabelle 7.7 ausgeführt. Wird zusätzlich eine LogLog-Darstellung angewandt, können wir unter Einsatz einer  $p$ -Verfeinerung einen Abfall des Fehlers feststellen (Abb. 7.36).

Tabelle 7.7: Ergebnisse verschärfter Boudary-Layer

Level	Kollokationspunkte	Fehler	Iterationen
1	2912	2.41e-03	2593
2	2912	2.25e-03	1886
3	3002	1.82e-03	1760
4	3101	8.87e-04	2029
5	3191	1.45e-04	2032
6	3245	1.44e-04	1554
7	3353	1.35e-04	2713
8	3443	7.31e-05	2016
9	3542	5.38e-05	3032
10	3695	1.28e-05	2568
11	3749	9.68e-06	1823
12	3794	8.63e-06	1592
13	3884	3.80e-06	1944
14	3884	3.80e-06	0
15	3884	3.80e-06	0

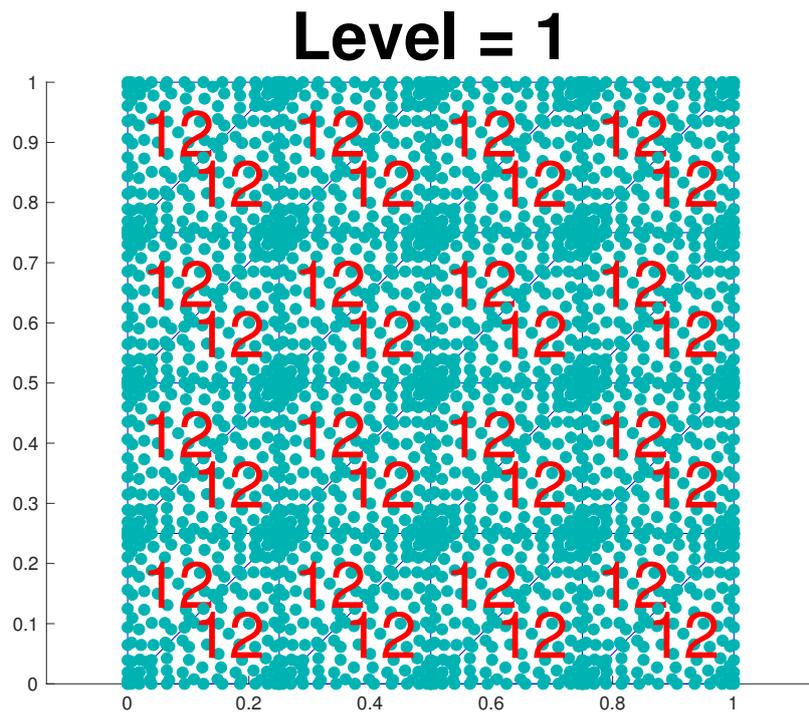


Abbildung 7.32:  $nT = 32$ , Polynomgrad entsprechend der roten Zahlen

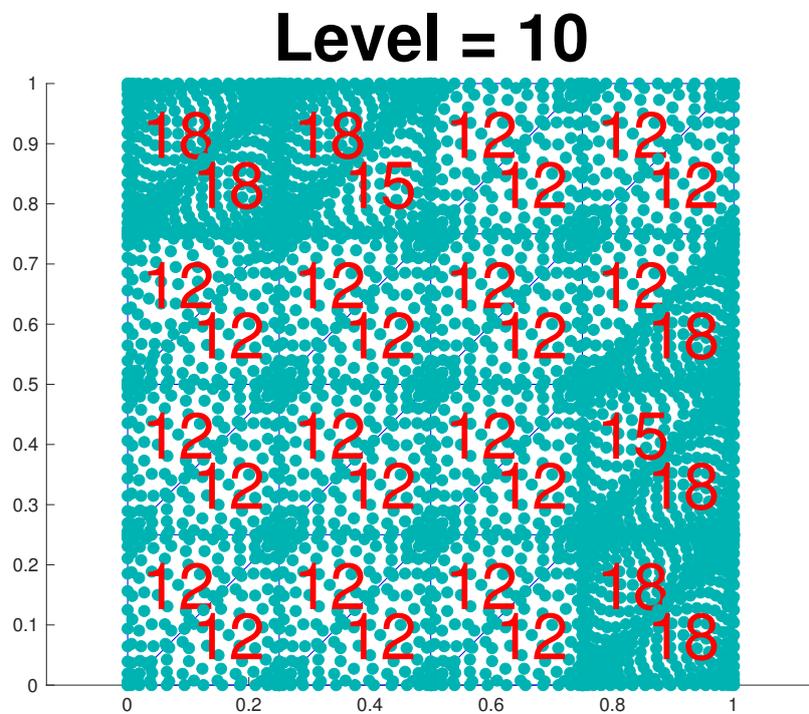


Abbildung 7.33:  $nT = 32$ , Polynomgrad entsprechend der roten Zahlen

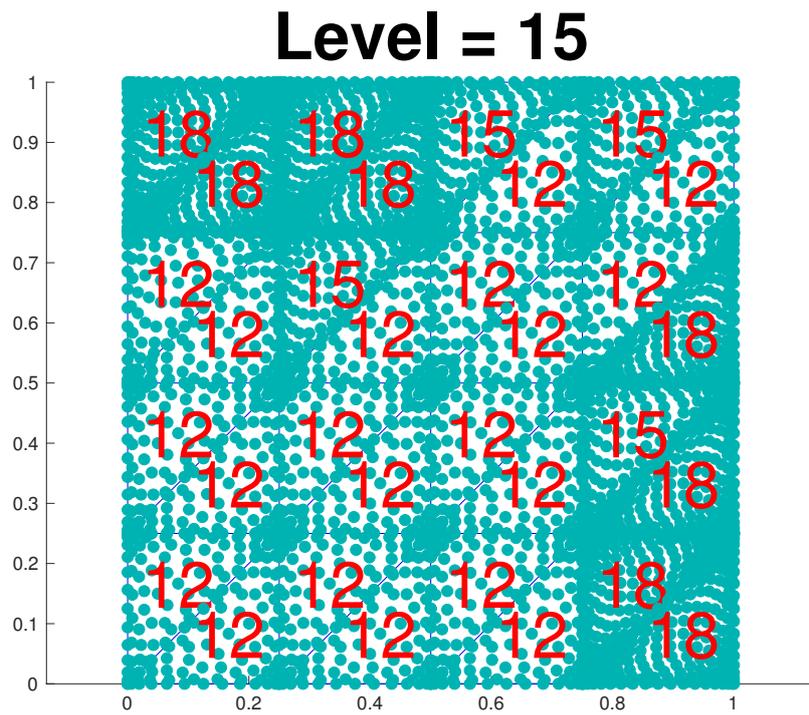
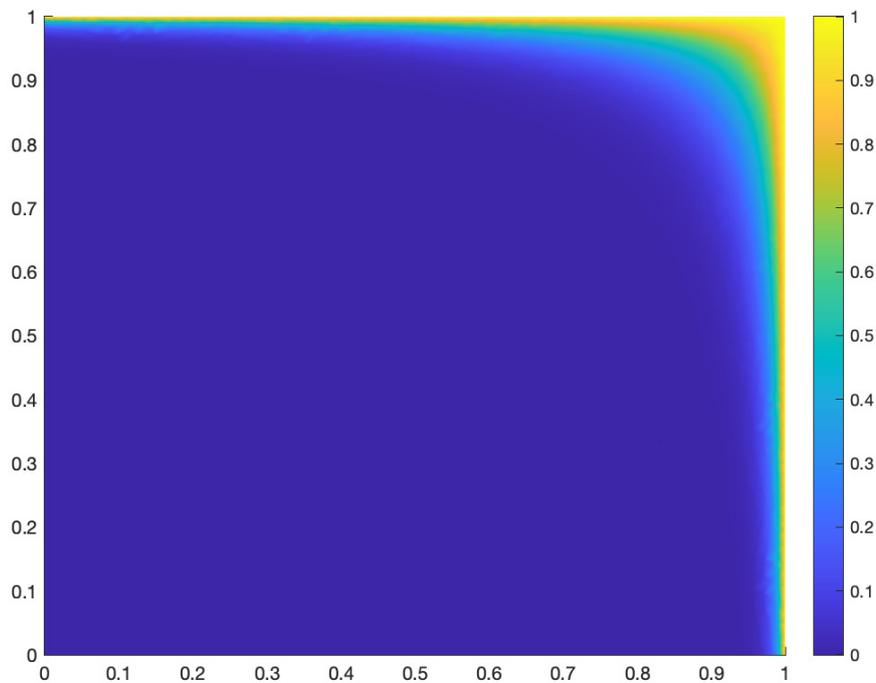
Abbildung 7.34:  $nT = 32$ , Polynomgrad entsprechend der roten Zahlen

Abbildung 7.35: Approximation der Lösung auf Level 15

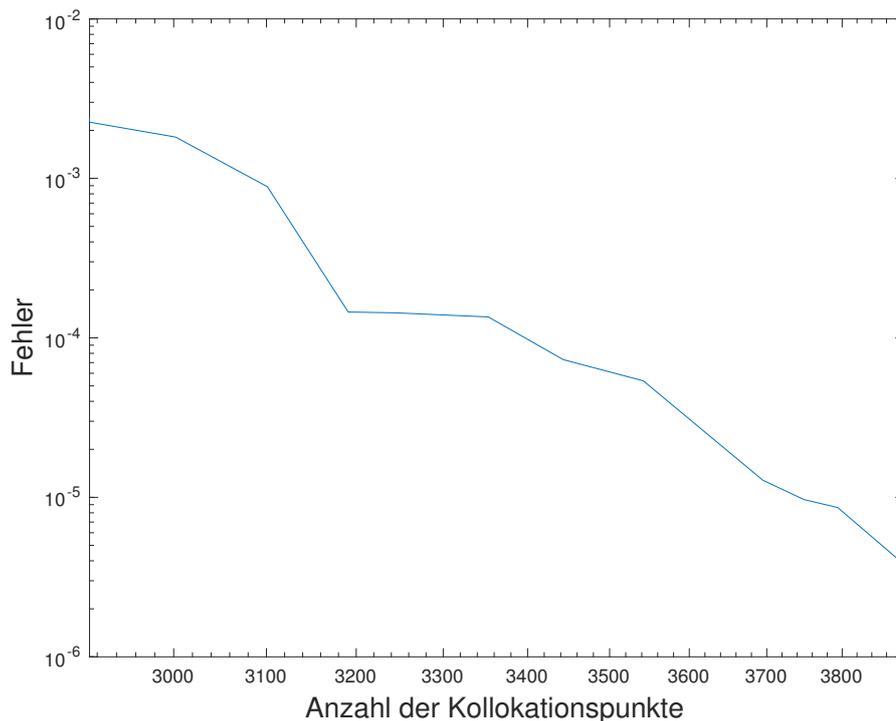
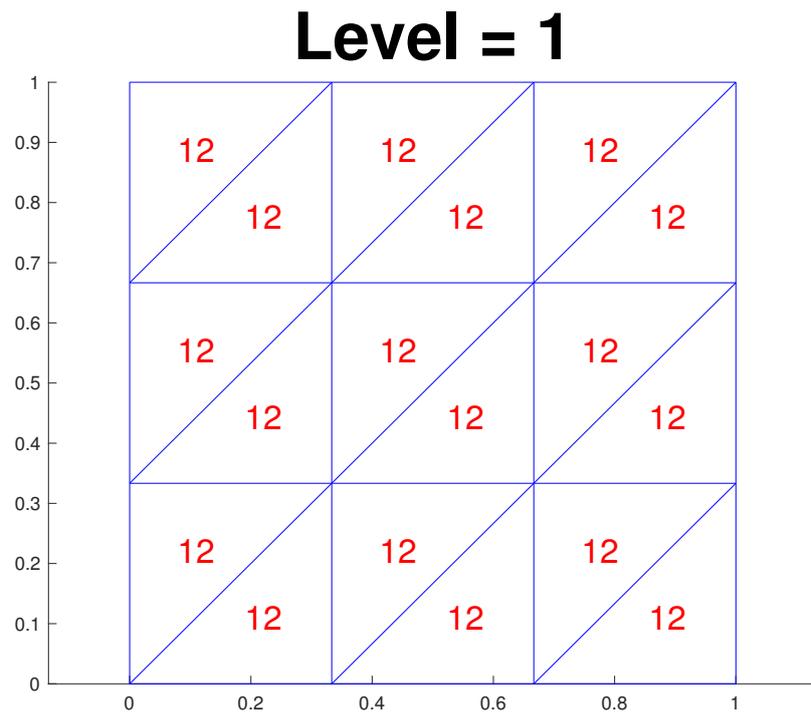
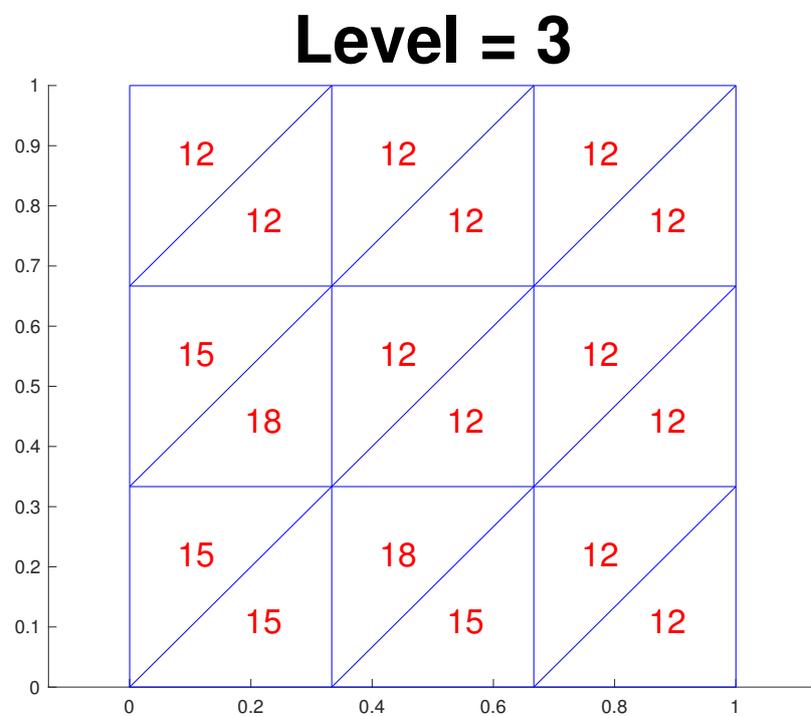


Abbildung 7.36: Loglog Plot des Fehlers

### 7.3.3.3 Eckenlayer

In diesem Abschnitt arbeiten wir mit dem betrachteten Modell aus 7.2.2.3. Wir verwenden folgende Parameter in der Neumann-Neumann Iteration  $\varepsilon_{\Gamma} = 1e^{-06}$ ,  $\theta = 0.01$  und starten auf  $nT = 18$  Dreiecke mit Polynomgrad  $N = 12$ . Das Dörfler-Kriterium wird angewandt mit  $\theta_{\text{Dörfler}} = 0.8$ .

Wir erkennen eine Verfeinerung entlang der unteren linken Ecke. Nach 4 Verfeinerungsschritten stoppt das Verfahren, da wir den maximalen Polynomgrad 18 erreicht haben und die kritische Ecke nicht mehr weiter verfeinert werden kann. Auch hier erkennen wir eine Abhängigkeit des Fehlers der Verfeinerungslevels und den resultierenden Kollokationspunkten. Der Fehler wird mit steigender Anzahl der Kollokationspunkte kleiner. Die Ergebnisse sind in Tabelle 7.8 ausgeführt.

Abbildung 7.37:  $nT = 18$ , Polynomgrad entsprechend der roten ZahlenAbbildung 7.38:  $nT = 18$ , Polynomgrad entsprechend der roten Zahlen

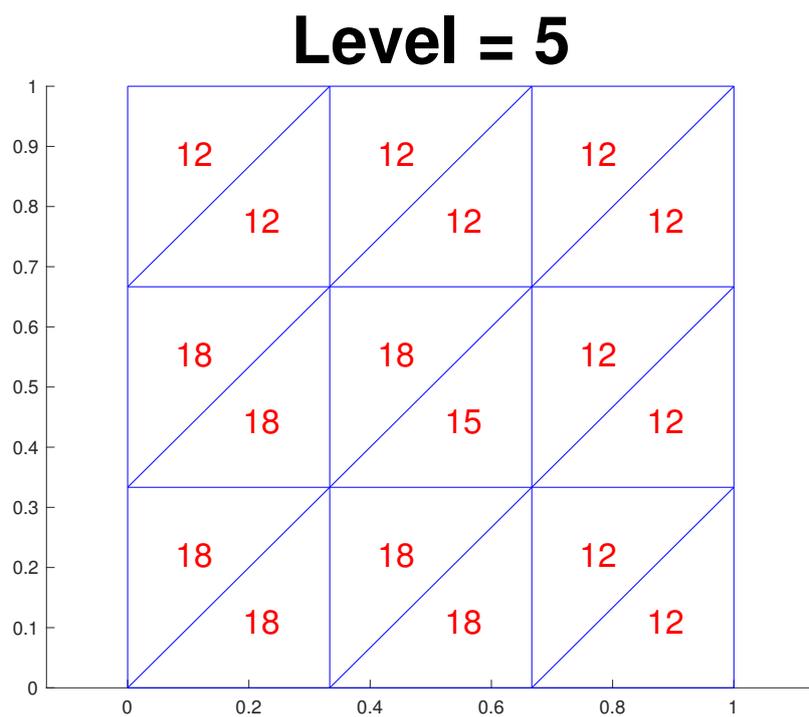


Abbildung 7.39:  $nT = 18$ , Polynomgrad entsprechend der roten Zahlen

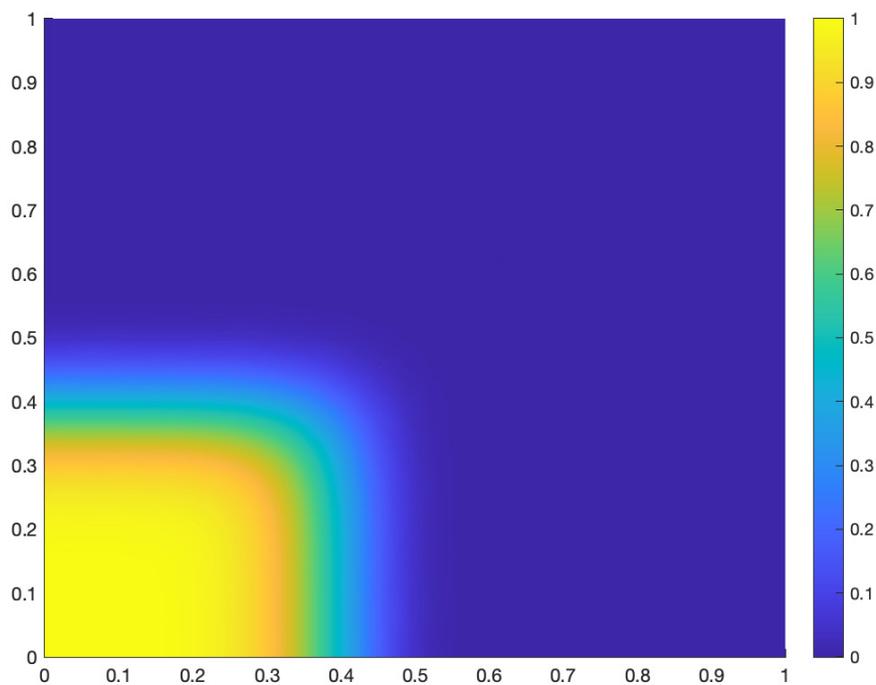


Abbildung 7.40:  $nT = 18$ , Lösung auf Level 5

Tabelle 7.8: Ergebnisse Ecken-Layer

Level	Kollokationspunkte	Fehler	Iterationen
1	1638	4.24e-08	2745
2	1818	2.09e-08	11855
3	2016	2.37e-08	24672
4	2214	1.38e-09	1258
5	2376	2.04e-09	3061

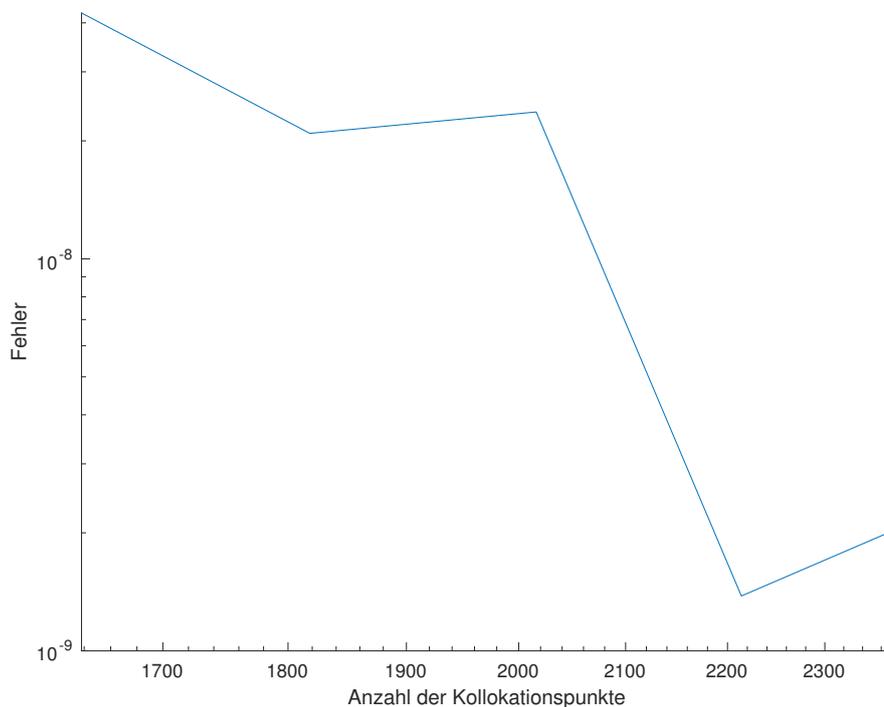


Abbildung 7.41: Konvergenzkurve

### 7.3.3.4 Hut-Layer

Im Falle des Hut-Layer Problems, beschrieben in 7.2.2.4, starten wir auf allen  $nT = 450$  Dreiecken mit Polynomgrad  $N = 9$ . Die hohe Menge an Dreiecken ist nötig, um den Rand des Hut-Layers genügend gut aufzulösen. Für die Neumann-Neumann Iteration, beschrieben in 7.2.2, gilt  $\varepsilon_{\Gamma} = 0.01$  und  $\theta = 0.1$ . Das Dörfler-Kriterium wird angewandt mit  $\theta_{\text{Dörfler}} = 0.95$ .

Die Abbildungen 7.42 und 7.43 stellen die uniforme Verteilung der Polynomgrade zum Start und die dazugehörige Lösung dar. Zwar wird der Übergang der Hut-Layer ansatzweise richtig abgebildet, jedoch weist der Hut-Layer nicht die richtige Größe

auf und wir können auch tiefe Gräben am Interface entdecken.

Abbildung 7.44 präsentiert die Verteilung der Polynomgrade nach 5  $p$ -Verfeinerungen. Eine Verfeinerung findet nur direkt am Interface statt. Auf Level 15 sehen wir die Verteilung der  $p$ -Verfeinerung in Abbildung 7.45 und der zugehörige Plot der Lösung in Abbildung 7.46 und 7.47. Die Lösung im letzten Schritt entspricht der erwarteten Lösung, der Sprung am Interface des Hut-Layers wird exzellent approximiert, was Abbildung 7.46 gut darstellt.

Trotz der Schwierigkeit dieses Problems, nehmen wir einen abfallenden Fehler mit steigender Anzahl der Kollokationspunkte wahr (vgl. Tabelle 7.9). Dazu wollen wir anmerken, dass in unser Routine der maximale Polynomgrad eines Dreieck bei  $N = 18$  liegt und eine Verfeinerung darüber hinaus nicht stattfindet. Diese Auswirkungen können in unseren Ergebnissen erkannt werden, der Fehler sowie die Kollokationspunkte stagnieren, da Dreiecke, die bereits den höchsten Polynomgrad erreicht haben, verfeinert werden sollen, was aber nicht möglich ist. Wird zusätzlich eine LogLog-Darstellung angewandt, können wir unter Einsatz einer  $p$ -Verfeinerung einen Abfall des Fehlers feststellen (Abb. 7.48).

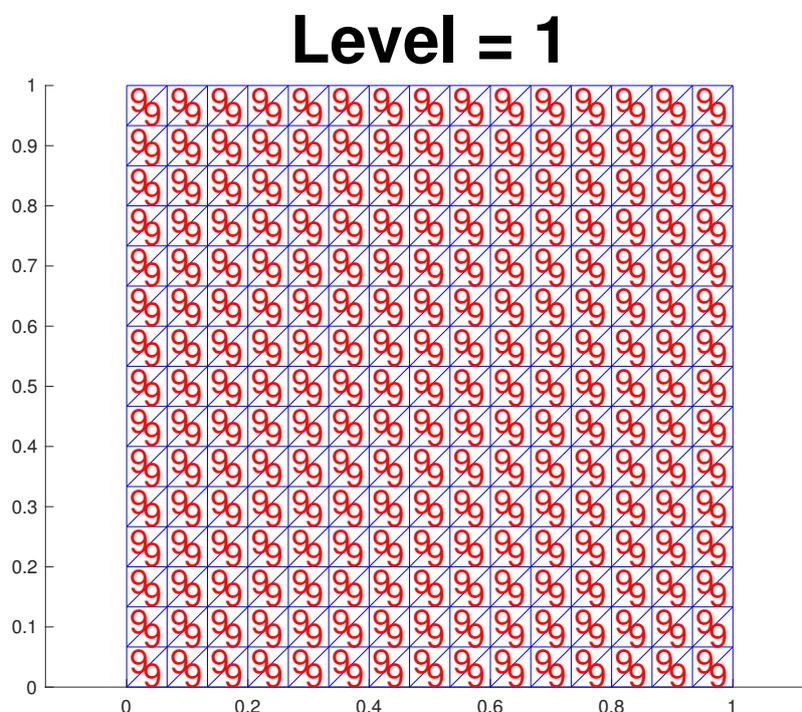


Abbildung 7.42: Gitter und Polynomgrad  $N = 9$  zum Start

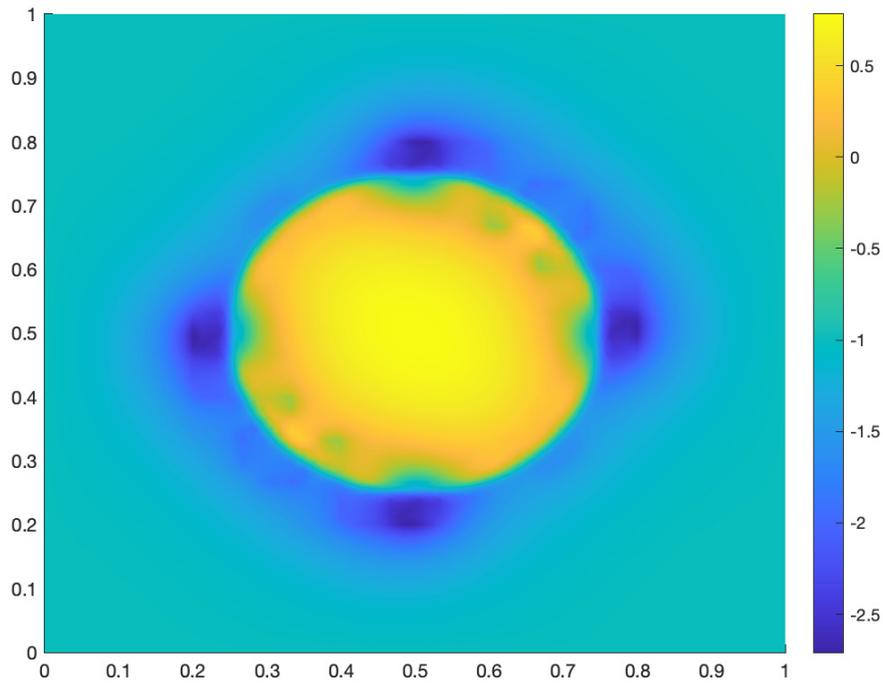


Abbildung 7.43: Lösung obige Ansicht: Level 1

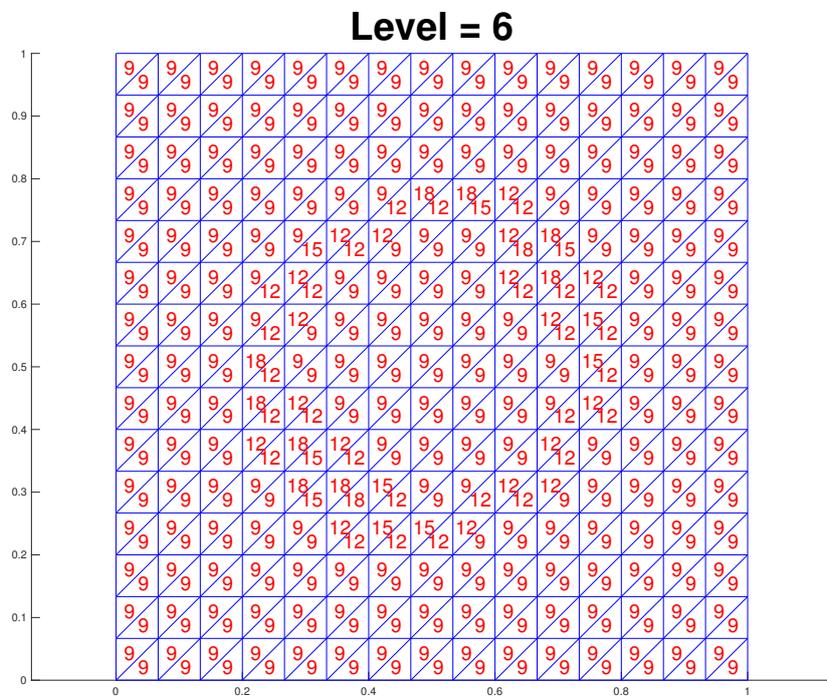


Abbildung 7.44: Gitter und Verteilung der Polynomgrad nach 5  $p$ -Verfeinerungen



Abbildung 7.45: Gitter und Verteilung der Polynomgrade nach 14  $p$ -Verfeinerungen

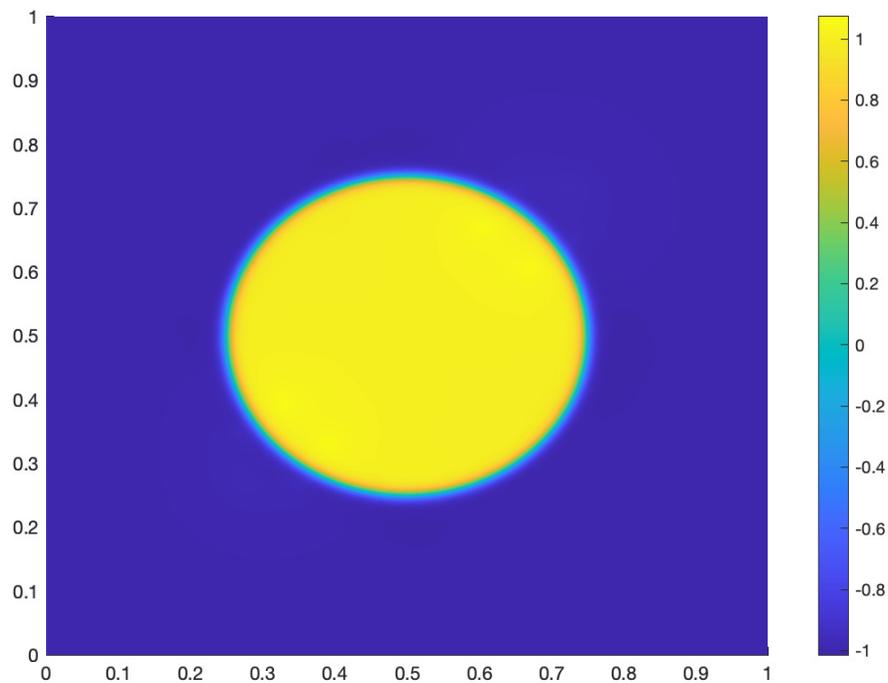


Abbildung 7.46: Lösung obige Ansicht: Level 15

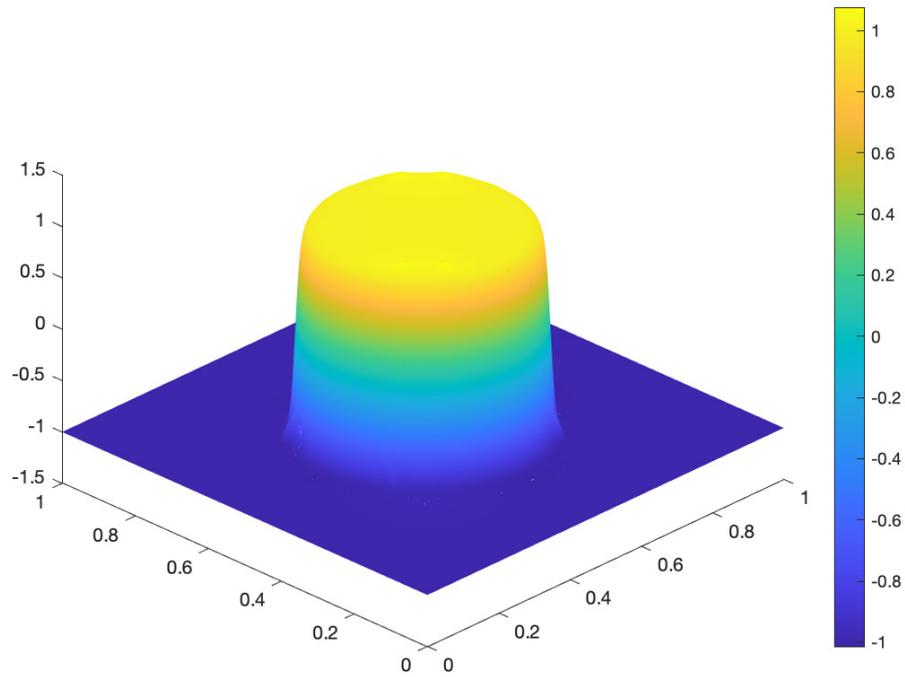


Abbildung 7.47: Lösung seitliche Ansicht: Level 15

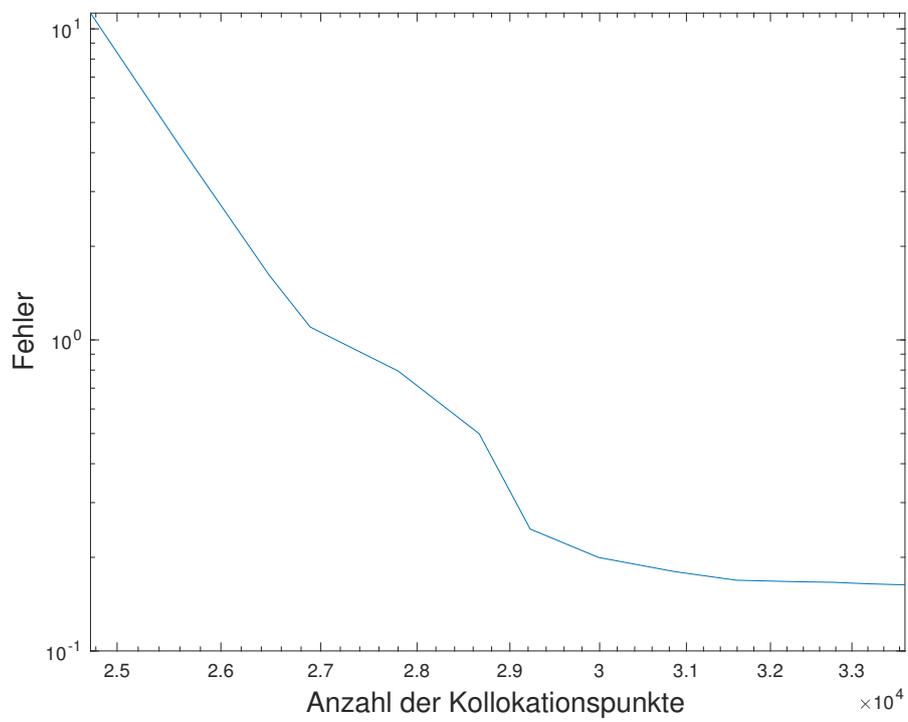


Abbildung 7.48: Konvergenzkurve

Tabelle 7.9: Ergebnisse Hut-Layer

Level	Kollokationspunkte	Fehler	Iterationen
1	24750	1.12e+01	429
2	25614	4.13e+00	377
3	26478	1.62e+00	297
4	26892	1.10e+00	212
5	27801	7.94e-01	140
6	28665	4.99e-01	136
7	29223	2.47e-01	134
8	29988	2.00e-01	57
9	30861	1.80e-01	52
10	31590	1.69e-01	32
11	32283	1.67e-01	9
12	32769	1.66e-01	7
13	33165	1.65e-01	10
14	33534	1.63e-01	8
15	33669	1.63e-01	2

Vergrößern wir den Parameter  $\varepsilon$  der Hut-Gleichung aus Kapitel 7.2.2.4 auf  $\varepsilon = 0.05$ , so ist die Lösung des Problems eine glattere Lösung. Dies lässt sich besser approximieren, da der Übergang des Hut-Layers flacher ist, siehe Abbildung 7.51.

Durch die glatter zu erwartende Lösung lässt sich auf der einen Seite mit erheblich weniger Dreiecke rechnen ( $nT = 32$ ) und auf der anderen Seite eine niedrigere Grenze des Abbruchkriteriums auf den Interfaces ( $\varepsilon_\Gamma = 10^{-05}$ ) anwenden. Entgegen dieser nicht zutragenden Änderungen, lassen sich dank der glatteren Lösung in 7.10 geringe Fehler bei einer viel geringeren Anzahl an Kollokationspunkten feststellen. Des Weiteren bemerken wir zwar in Tabelle 7.10 einen kleineren und abfallenden Fehler im Vergleich zum zuvor untersuchten Hut-Layer, jedoch fällt dieser Fehler angesichts der glatteren Lösung trotz weiterer Verfeinerung nur langsam. Dieses lässt sich mit der Wahl von  $\varepsilon_\Gamma$  begründen. Dieser Parameter bestimmt die Grenze der Neumann-Neumann Iteration auf den Interfaces. Zusätzlich lässt sich die nicht gänzlich symmetrische Verfeinerung mit der Anwendung des Dörfler-Kriteriums begründen, dabei wird, wie zuvor beschrieben, eine prozentual abhängige Menge von Dreiecken gemessen am gesamten Fehler zur Verfeinerung ausgewählt. Diese Auswahl führt bei dieser geringen Anzahl von Dreiecken und geringen Anzahl von verfeinerten Dreiecken pro Iteration zu nicht symmetrischen Mustern, wie sie auf Abbildung 7.50 zu sehen sind, dabei fällt auf, dass beim Auftreten von nicht Symmetrie

in fast allen Fällen nur eine Verfeinerungsstufe liegt. Im Fall des glatten Hut-Layers wurde  $\varepsilon_\Gamma = 10^{-5}$  und  $\theta = 0.1$  gewählt, damit stellt dies auch indirekt eine Obergrenze der Minimierung des Fehlers auf den Interfaces dar. Das Dörfler-Kriterium wird angewandt mit  $\theta_{\text{Dörfler}} = 0.97$ .

In unseren Tests lies sich  $\varepsilon_\Gamma$  nicht kleiner wählen, da sonst die Neumann-Neumann Iteration nicht konvergierte, somit wird der Fehler in Tabelle 7.10 durch den Interfacefehler dominiert und eine bessere Approximation ist nicht möglich.

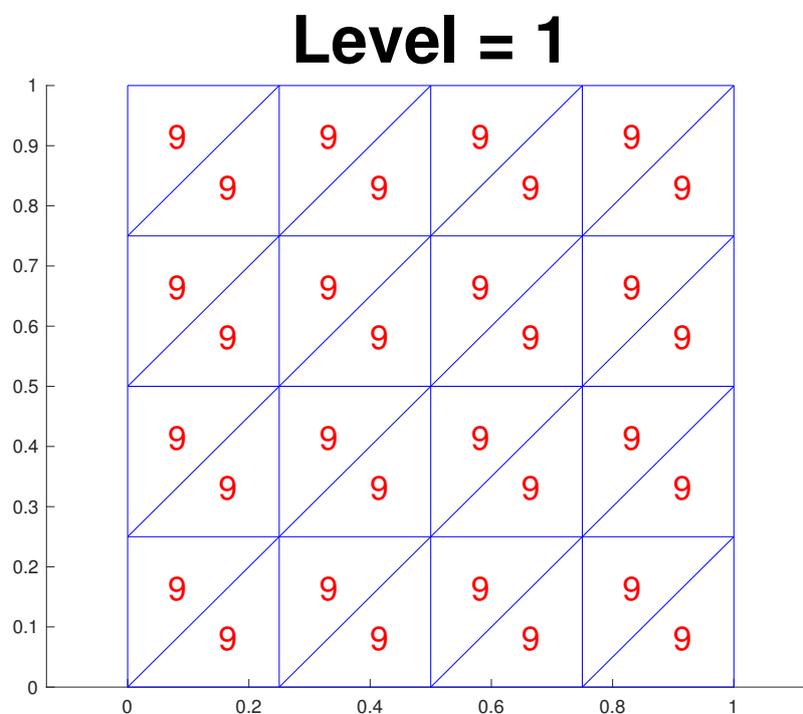


Abbildung 7.49: Gitter und Verteilung der Polynomgrade zu Beginn

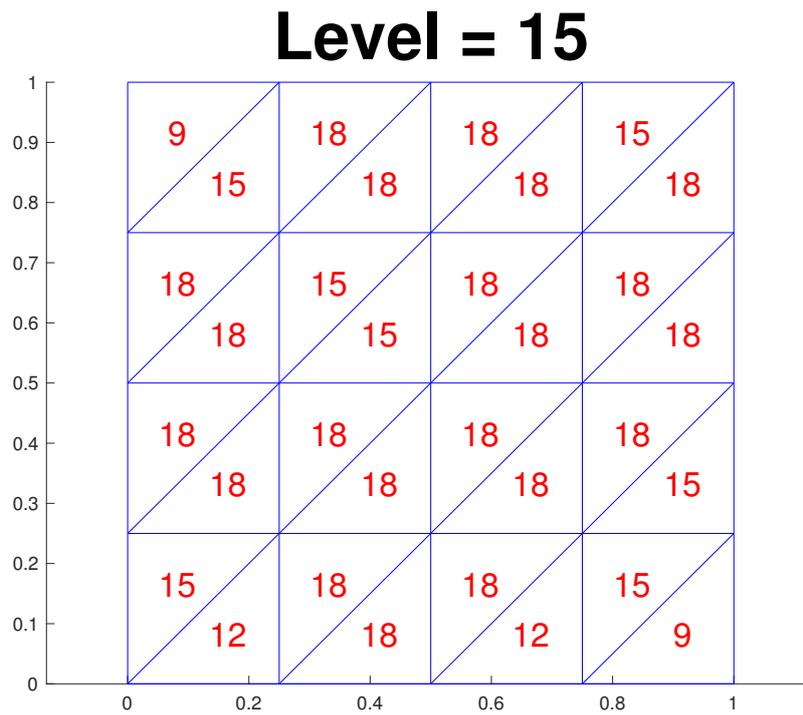


Abbildung 7.50: Gitter und Verteilung der Polynomgrade nach 14  $p$ -Verfeinerungen

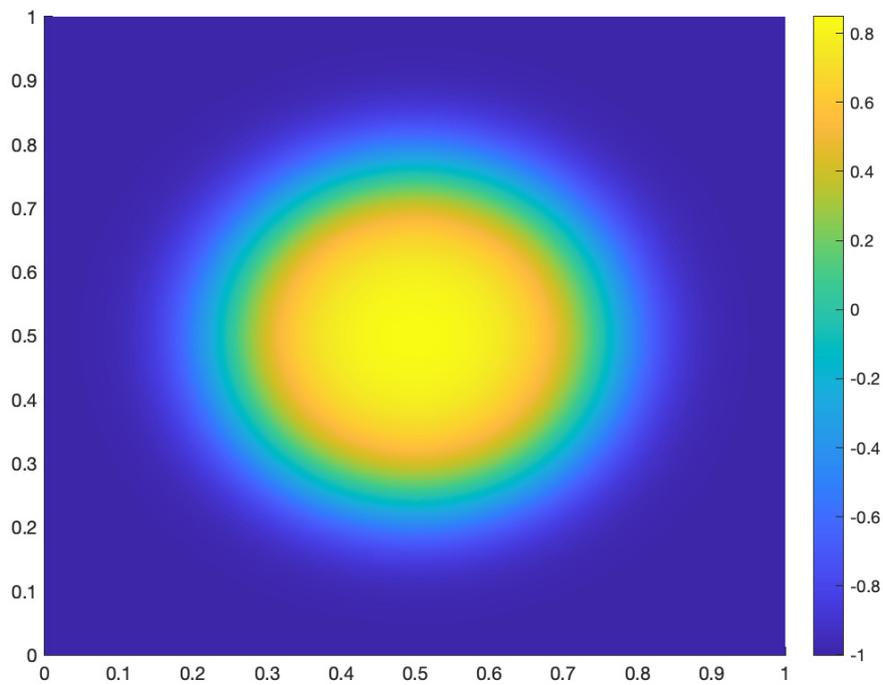


Abbildung 7.51: Lösung: Level 15

Tabelle 7.10: Ergebnisse des glatten Hut-Layer

Level	Kollokationspunkte	Fehler
1	1760	4.52e-02
2	1904	2.21e-02
3	2300	1.67e-02
4	2498	8.36e-03
5	2813	2.75e-03
6	3110	3.86e-04
7	3488	3.03e-04
8	3686	2.59e-04
9	4001	2.14e-04
10	4046	2.08e-04
11	4325	1.73e-04
12	4568	1.48e-04
13	4730	1.27e-04
14	5027	1.04e-04
15	5234	9.65e-05



# Kapitel 8

## Fazit und Ausblick

Diese Arbeit behandelte thematisch die Kombination von Gebietszerlegungsmethoden und spektralen Kollokationsmethoden. Zu Beginn wurde in Kapitel 2 verschiedene Arten der Gebietszerlegungsmethoden diskutiert und im Speziellen zwei Gebietszerlegungsmethoden vorgestellt, die Dirichlet-Neumann Iteration und die Neumann-Neumann Iteration, welche zur Gruppe der nicht-überlappenden Zerlegungen gehören und in der Arbeit Anwendungen fanden.

Im Anschluss wurde im Abschnitt 3.2.1 die spektrale Kollokationsmethode eingeführt. Dazu wurden verschiedene Klassen von Polynomen beschrieben und als Folge die Dubiner-Polynome in Kapitel 4.1 definiert, die zur Herleitung der Fekete-Punkte benötigt wurden. Diese Fekete-Punkte stellten unter anderem die Grundlagen der verwendeten Kollokationsmethode auf Dreiecken da. Zusätzlich wurden auch in Kapitel 3.3 weitere Möglichkeiten für Kollokationspunkte diskutiert, welche unter anderem Anwendung in Kapitel 6.1 finden, die zur Kollokation auf rechteckigen Gebieten verwendet werden.

In Kapitel 5 wurde in jeweils drei Untersektionen die drei Hauptbestandteile, der in dieser Arbeit untersuchten Methode, analytisch betrachtet und diskutiert.

Anschließend wurde die Kombination der eingeführten Kollokationsmethode und den zuvor im Abschnitt 2.2.1 und 2.2.2 beschriebenen Dirichlet-Neumann Iteration und Neumann-Neumann Iteration untersucht. Dazu wurden verschiedene numerische Probleme aufgegriffen, die bereits in [HL01] mit anderen Kollokationspunkten untersucht worden sind.

Wir konnten in unseren numerischen Tests mit der Dirichlet-Neumann Methode erkennen, dass unser Verfahren mit Fekete-Punkten als Kollokationspunkten auf Dreiecken ähnliche oder gar bessere Ergebnisse liefert als in [HL01]. Dadurch, dass wir weniger Kollokationspunkte benötigen, entstehen kleinere Gleichungssysteme, welche zu einer Reduktion des Rechenaufwandes beitragen können.

Durch Anwenden der Neumann-Neumann Methode auf die Serie von Modellproble-

men aus Kapitel 6 bemerken wir, dass sich der Fehler ähnlich zu den der vorherigen Tests mittels Dirichlet-Neumann Methode verhält. Im Vergleich zu den Ergebnissen aus [HL01], stellen wir jedoch bessere Ergebnisse fest.

Zusätzlich bestand der Fokus dieser Arbeit auf die Kombination der Neumann-Neumann Iteration und der spektralen Kollokationsmethode angewandt auf einer Auswahl von Konvektions-Diffusionsproblemen, dazu wurden rechteckige Gebiete  $\Omega$  in gleichmäßige Dreiecke zerlegt und Fekete-Punkte als Kollokationspunkte verwendet. Innerhalb der verschiedenen numerischen Tests wurden globale Verfeinerungen in  $h$  (Größe der Dreiecke) und  $p$  (Polynomgrad) untersucht, dies geschah zur Feststellung der Effizienz der unterschiedlichen Verfeinerungsmöglichkeiten bezüglich der Verringerung des Approximationsfehlers.

Neben der klassischen Poisson-Gleichung wurden vier verschiedene Konvektions-Diffusionsgleichungen, die zu den singular-gestörten Problemen zählen, untersucht. Wir stellen einen exponentiellen Abfall des Fehlers bei festem  $h$  (Anzahl der Dreiecke) mit der Zunahme des Polynomgrades fest und einen Abfall des Fehlers entsprechend des festgehaltenen Polynomgrades bei einer  $h$ -Verfeinerung.

Aufgrund dieser Ergebnisse folgern wir, dass sowohl die Verfeinerung des Gitters als auch die Erhöhung des Polynomgrades die gewünschten Auswirkungen unter Verwendung der vorstellten Methode liefern.

Der Nachteil einer gleichmäßigen  $h$ - oder  $p$ -Verfeinerung stellt die stark ansteigende Anzahl an Rechenaufwand dar, im Speziellen bei den von uns betrachteten Konvektions-Diffusionsproblemen, welche nur lokale Ereignisse modellieren (zum Beispiel Boundary-Layer). Bei dieser Art von Problem würde eine lokale Verfeinerung womöglich ausreichen, dies würde weniger zusätzlichen Rechenaufwand bedeuten. Aufbauend auf dieser Überlegung, wurde in Kapitel 7.3 der Algorithmus um  $p$ -Adaptivität erweitert. Im Kern soll der Polynomgrad nur dort erhöht werden, wo der Fehler lokal am größten geschätzt wird. Dazu verwenden wir, das in Abschnitt 7.3.2 eingeführte unter dem Namen „Koeffizienten entlang der Spur des Spektrums“ bekannte, als Fehlerindikator und das Dörfler-Marking Kriterium, um die Menge der zu verfeinernden Elemente zu bestimmen.

Die numerischen Tests der vier Konvektions-Diffusionsgleichungen zeigen, dass wir zwar durch die adaptive  $p$ -Verfeinerung deutlich weniger Kollokationspunkte benötigen, um vergleichbare Fehlergrößen zu replizieren und besonders im Fall des moderaten Boundary-Layers, verschärften Boundary-Layers und Eckenlayers, jedoch die Anzahl der Iterationen höher sind, dies kann aber durch Optimierung der Schrittweite der Neumann-Neumann Iteration verbessert werden, was nicht Teil dieser Arbeit war. Im Fall des Hut-Layers zeigt sich, dass die adaptive  $p$ -Verfeinerung aufgrund

---

der Limitierung der Fekete-Punkte, nicht die volle Effizienz entfalten kann. Die Limitierung besteht in dem maximalen Polynomgrad eines Dreiecks bei  $N = 18$ , eine Verfeinerung darüber hinaus kann nicht stattfinden, wodurch es nicht möglich war den Polynomgrad der betroffenen Dreiecke weiter zu verfeinern. Zusammenfassend kann gesagt werden, dass bei den untersuchten singular-gestörten Problemen, die durch das Vorhandensein von Layern gekennzeichnet sind, adaptive  $p$ -Verfeinerung einen positiven Einfluss auf die berechneten Fehler unter Berücksichtigung der Anzahl der Kollokationspunkte hat und den gewünschten Mehrwert liefert.

Aufbauend auf dieser Arbeit verbleiben noch weitere interessante Fragestellungen, die in Zukunft betrachtet werden können. Zum Einen wurde in dieser Arbeit ein Fehlerindikator verwendet, welcher zwar als Maß des Fehlers verstanden werden kann, aber kein mathematisch rigorosen Fehlerschätzer darstellt. Das Anwenden von problemspezifischen, beweisbaren Fehlerschätzern könnte zum einen zu bessere und effizienteren Verfeinerung führen und garantierte Konvergenzraten sichern. Zusätzlich wurde in der Arbeit eine globale und uniforme  $h$  und  $p$ -Verfeinerung sowie eine adaptive  $p$ -Verfeinerung untersucht. Diese Untersuchungen lassen sich auch auf adaptive  $hp$  und  $h$ -Verfeinerungen ausweiten, welche bessere Ergebnisse erwarten lassen.

Das Herleiten zusätzlicher Fekete-Punkte, welche den Nutzen von Lösungen höheren Polynomgrades ermöglichen (über Grad 18), könnte auch zu besseren Ergebnissen in der  $p$ -Adaptivität führen und zusätzliche Verfeinerungsstufen ermöglichen.

Des Weiteren lassen sich noch andere physikalische Problemstellungen betrachten, wie Navier-Stokes oder auch zeitabhängige Probleme. Zuletzt sparte die in Kapitel 5 analytische Betrachtung wichtige Aspekte aus und zeigte die fehlende mathematische Theorie hinsichtlich eines Kollokationsansatzes auf Dreiecken durch Fekete-Punkte, hierzu besteht auch heute noch Forschungsbedarf, um nicht nur experimentelle Konvergenzeigenschaften zu zeigen, sondern auch gestützt durch theoretische Ergebnisse der Mathematik.



# Anhang A

## Dubiner-Polynome

Wir präsentieren die Dubiner-Polynome vom Grad  $\leq 3$  und ihre Ableitungen nach  $x$  und  $y$ . Dabei ist  $g_k \hat{=} g_{mn}$  für  $k = \iota(m, n)$  wie in (4.2).

### A.1 Dubiner-Basis

Die Dubiner-Polynome sind eine orthogonale Basis  $\mathcal{B}_N$  des Raumes  $\mathcal{P}_N(T_0)$ . Es ist  $\mathcal{B}_0 = \{g_1\}$ ,  $\mathcal{B}_1 = \{g_1, g_2, g_3\}$ ,  $\mathcal{B}_2 = \{g_1, \dots, g_6\}$  und  $\mathcal{B}_3 = \{g_1, \dots, g_{10}\}$  :

$$g_1(x, y) = 1$$

$$g_2(x, y) = \frac{3}{2}y + \frac{1}{2}$$

$$g_3(x, y) = x + \frac{1}{2}y + \frac{1}{2}$$

$$g_4(x, y) = \frac{5}{2}y^2 + y - \frac{1}{2}$$

$$g_5(x, y) = \frac{5}{2}xy + \frac{5}{4}y^2 + \frac{3}{2}x + 2y + \frac{3}{4}$$

$$g_6(x, y) = \frac{3}{2}x^2 + \frac{3}{2}xy + \frac{1}{4}y^2 + \frac{3}{2}x + y + \frac{1}{4}$$

$$g_7(x, y) = \frac{35}{8}y^3 + \frac{15}{8}y^2 - \frac{15}{8}y - \frac{3}{8}$$

$$g_8(x, y) = \frac{21}{4}xy^2 + \frac{21}{8}y^3 + \frac{9}{2}xy + \frac{39}{8}y^2 + \frac{1}{4}x + \frac{19}{8}y + \frac{1}{8}$$

$$g_9(x, y) = \frac{21}{4}x^2y + \frac{21}{4}xy^2 + \frac{7}{8}y^3 + \frac{15}{4}x^2 + 9xy + \frac{33}{8}y^2 + \frac{15}{4}x + \frac{27}{8}y + \frac{5}{8}$$

$$g_{10}(x, y) = \frac{5}{2}x^3 + \frac{15}{4}x^2y + \frac{3}{2}xy^2 + \frac{1}{8}y^3 + \frac{15}{4}x^2 + \frac{9}{2}xy + \frac{9}{8}y^2 + \frac{3}{2}x + \frac{9}{8}y + \frac{1}{8}$$

## A.2 Ableitungen der Dubiner-Polynome

Ableitungen in  $x$  Richtung:

$$\begin{aligned} \frac{\partial}{\partial x} g_1(x, y) &= 0 \\ \frac{\partial}{\partial x} g_2(x, y) &= 0 \\ \frac{\partial}{\partial x} g_3(x, y) &= 1 \\ \frac{\partial}{\partial x} g_4(x, y) &= 0 \\ \frac{\partial}{\partial x} g_5(x, y) &= \frac{5}{2}y + \frac{3}{2} \\ \frac{\partial}{\partial x} g_6(x, y) &= 3x + \frac{3}{2}y + \frac{3}{2} \\ \frac{\partial}{\partial x} g_7(x, y) &= 0 \\ \frac{\partial}{\partial x} g_8(x, y) &= \frac{21}{4}y^2 + \frac{9}{2}y + \frac{1}{4} \\ \frac{\partial}{\partial x} g_9(x, y) &= \frac{21}{2}xy + \frac{21}{4}y^2 + \frac{15}{2}x + 9y + \frac{15}{4} \\ \frac{\partial}{\partial x} g_{10}(x, y) &= \frac{15}{2}x^2 + \frac{15}{2}xy + \frac{3}{2}y^2 + \frac{15}{2}x + \frac{9}{2}y + \frac{3}{2} \end{aligned}$$

Ableitungen in  $y$  Richtung:

$$\begin{aligned} \frac{\partial}{\partial y} g_1(x, y) &= 0 \\ \frac{\partial}{\partial y} g_2(x, y) &= \frac{3}{2} \\ \frac{\partial}{\partial y} g_3(x, y) &= \frac{1}{2} \\ \frac{\partial}{\partial y} g_4(x, y) &= 5y + 1 \\ \frac{\partial}{\partial y} g_5(x, y) &= \frac{5}{2}x + \frac{5}{2}y + 2 \\ \frac{\partial}{\partial y} g_6(x, y) &= \frac{3}{2}x + \frac{1}{2}y + 1 \\ \frac{\partial}{\partial y} g_7(x, y) &= \frac{105}{8}y^2 + \frac{15}{4}y - \frac{15}{8} \\ \frac{\partial}{\partial y} g_8(x, y) &= \frac{21}{2}xy + \frac{63}{8}y^2 + \frac{9}{2}x + \frac{39}{4}y + \frac{19}{8} \\ \frac{\partial}{\partial y} g_9(x, y) &= \frac{21}{4}x^2 + \frac{21}{2}xy + \frac{21}{8}y^2 + 9x + \frac{33}{4}y + \frac{27}{8} \\ \frac{\partial}{\partial y} g_{10}(x, y) &= \frac{15}{4}x^2 + 3xy + \frac{3}{8}y^2 + \frac{9}{2}x + \frac{9}{4}y + \frac{9}{8} \end{aligned}$$

# Anhang B

## Fekete-Punkte

Wir stellen die Fekete-Punkte in kartesischen Koordinaten auf dem Referenzdreieck  $T_0$  für  $N \in \{3, 6, 9, 12\}$  vor [TWV00].

N=3:

x	y	x	y
-1.0000000000	-1.0000000000	0.4472135954	-0.4472135954
-0.4472135954	-1.0000000000	-0.3333333333	-0.3333333333
0.4472135954	-1.0000000000	-1.0000000000	0.4472135954
1.0000000000	-1.0000000000	-0.4472135954	0.4472135954
-1.0000000000	-0.4472135954	-1.0000000000	1.0000000000

N=6:

x	y	x	y
-1.0000000000	-1.0000000000	0.4688697196	-0.4688697196
-0.8302291554	-1.0000000000	0.1330985740	-0.3674604082
-0.4688697196	-1.0000000000	-0.7656381658	-0.3674604082
0.0000000000	-1.0000000000	-0.3333333333	-0.3333333333
0.4688697196	-1.0000000000	-1.0000000000	0.0000000000
0.8302291554	-1.0000000000	0.0000000000	0.0000000000
1.0000000000	-1.0000000000	-0.7656381658	0.1330985740
-1.0000000000	-0.8302291554	-0.3674604082	0.1330985740
0.8302291554	-0.8302291554	-1.0000000000	0.4688697196
-0.7873290632	-0.7873290632	-0.4688697196	0.4688697196
0.5746581264	-0.7873290632	-0.7873290632	0.5746581264
-0.3674604082	-0.7656381658	-1.0000000000	0.8302291554
0.1330985740	-0.7656381658	-0.8302291554	0.8302291554
-1.0000000000	-0.4688697196	-1.0000000000	1.0000000000

N=9:

x	y	x	y
-1.0000000000	-1.0000000000	0.4779256080	-0.4779256080
-0.9195339860	-1.0000000000	-0.8822870242	-0.3979515780
-0.7387741816	-1.0000000000	0.2802386022	-0.3979515780
-0.4779256080	-1.0000000000	-0.6431324824	-0.3495130200
-0.1652794130	-1.0000000000	-0.0073544976	-0.3495130200
0.1652794130	-1.0000000000	-0.3333333333	-0.3333333333
0.4779256080	-1.0000000000	0.1652794130	-0.1652794130
0.7387741816	-1.0000000000	-1.0000000000	-0.1652794130
0.9195339860	-1.0000000000	-0.0600824712	-0.0600824712
1.0000000000	-1.0000000000	-0.8798350576	-0.0600824712
-1.0000000000	-0.9195339860	-0.3495130200	-0.0073544976
0.9195339860	-0.9195339860	-0.6431324824	-0.0073544976
-0.9021308608	-0.9021308608	-1.0000000000	0.1652794130
0.8042617216	-0.9021308608	-0.1652794130	0.1652794130
-0.6912196112	-0.8896483842	-0.8822870242	0.2802386022
0.5808679954	-0.8896483842	-0.3979515780	0.2802386022
-0.3979515780	-0.8822870242	-0.6591363598	0.3182727196
0.2802386022	-0.8822870242	-0.4779256080	0.4779256080
-0.0600824712	-0.8798350576	-1.0000000000	0.4779256080
-1.0000000000	-0.7387741816	-0.8896483842	0.5808679954
0.7387741816	-0.7387741816	-0.6912196112	0.5808679954
-0.8896483842	-0.6912196112	-1.0000000000	0.7387741816
0.5808679954	-0.6912196112	-0.7387741816	0.7387741816
-0.6591363598	-0.6591363598	-0.9021308608	0.8042617216
0.3182727196	-0.6591363598	-1.0000000000	0.9195339860
-0.3495130200	-0.6431324824	-0.9195339860	0.9195339860
-0.0073544976	-0.6431324824	-1.0000000000	1.0000000000
-1.0000000000	-0.4779256080		

---

N=12:

x	y	x	y
-1.0000000000	-1.0000000000	-1.0000000000	-0.4829098210
-0.9533098466	-1.0000000000	0.4829098210	-0.4829098210
-0.8463475646	-1.0000000000	-0.4763189598	-0.4763189598
-0.6861884690	-1.0000000000	-0.0473620804	-0.4763189598
-0.4829098210	-1.0000000000	-0.8502497696	-0.4250356576
-0.2492869302	-1.0000000000	0.2752854272	-0.4250356576
0.0000000000	-1.0000000000	-0.3333333333	-0.3333333333
0.2492869302	-1.0000000000	-1.0000000000	-0.2492869302
0.4829098210	-1.0000000000	0.2492869302	-0.2492869302
0.6861884690	-1.0000000000	0.0145023904	-0.2324962484
0.8463475646	-1.0000000000	-0.7820061420	-0.2324962484
0.9533098466	-1.0000000000	-0.6022233048	-0.1988883476
1.0000000000	-1.0000000000	-0.1988883476	-0.1988883476
-1.0000000000	-0.9533098466	-0.9395364342	-0.1856301448
0.9533098466	-0.9533098466	0.1251665790	-0.1856301448
-0.5021440620	-0.9499754770	-0.4763189598	-0.0473620804
0.4521195390	-0.9499754770	-1.0000000000	0.0000000000
-0.7586347292	-0.9474709564	0.0000000000	0.0000000000
0.7061056856	-0.9474709564	-0.2324962484	0.0145023904
-0.1856301448	-0.9395364342	-0.7820061420	0.0145023904
0.1251665790	-0.9395364342	-0.9395364342	0.1251665790
-0.9326048528	-0.9326048528	-0.1856301448	0.1251665790
0.8652097056	-0.9326048528	-0.6818331042	0.1909695082
-0.4250356576	-0.8502497696	-0.5091364040	0.1909695082
0.2752854272	-0.8502497696	-1.0000000000	0.2492869302
-1.0000000000	-0.8463475646	-0.2492869302	0.2492869302
0.8463475646	-0.8463475646	-0.4250356576	0.2752854272
-0.8385226450	-0.8385226450	-0.8502497696	0.2752854272
0.6770452900	-0.8385226450	-0.9499754770	0.4521195390
-0.6605731084	-0.8225925648	-0.5021440620	0.4521195390
0.4831656732	-0.8225925648	-1.0000000000	0.4829098210
-0.2324962484	-0.7820061420	-0.4829098210	0.4829098210
0.0145023904	-0.7820061420	-0.8225925648	0.4831656732
-0.9474709564	-0.7586347292	-0.6605731084	0.4831656732
0.7061056856	-0.7586347292	-0.8385226450	0.6770452900

x	y	x	y
-1.0000000000	-0.6861884690	-1.0000000000	0.6861884690
0.6861884690	-0.6861884690	-0.6861884690	0.6861884690
0.1909695082	-0.6818331042	-0.9474709564	0.7061056856
-0.5091364040	-0.6818331042	-0.7586347292	0.7061056856
0.4831656732	-0.6605731084	-1.0000000000	0.8463475646
-0.8225925648	-0.6605731084	-0.8463475646	0.8463475646
-0.1988883476	-0.6022233048	-0.9326048528	0.8652097056
0.1909695082	-0.5091364040	-1.0000000000	0.9533098466
-0.6818331042	-0.5091364040	-0.9533098466	0.9533098466
-0.9499754770	-0.5021440620	-1.0000000000	1.0000000000
0.4521195390	-0.5021440620		

# Literaturverzeichnis

- [AG16] A. V. Astaneh and M. N. Guddati. A two-level domain decomposition method with accurate interface conditions for the helmholtz problem. *International Journal for Numerical Methods in Engineering*, 107(1):74–90, 2016.
- [BM92] C. Bernardi and Y. Maday. *Approximations spectrales de problemes aux limites elliptiques*, volume 10. Springer, 1992.
- [BM97] C. Bernardi and Y. Maday. Spectral methods. *Handbook of numerical analysis*, 5:209–485, 1997.
- [BMN19] S. Badia, A. F. Martín, and H. Nguyen. Physics-based balancing domain decomposition by constraints for multi-material problems. *Journal of Scientific Computing*, 79(2):718–747, 2019.
- [Bos91] L. Bos. On certain configurations of points in  $\mathbb{R}^n$  which are unisolvent for polynomial interpolation. *Journal of approximation theory*, 64(3):271–280, 1991.
- [Bre11] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*, volume 2. Springer, 2011.
- [BTW01] L. Bos, M. Taylor, and B. Wingate. Tensor product gauss-lobatto points are fekete points for the cube. *Mathematics of Computation*, 70(236):1543–1547, 2001.
- [BW86] P. E. Bjørstad and O. B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM Journal on Numerical Analysis*, 23(6):1097–1120, 1986.
- [Cai03] X. Cai. Overlapping domain decomposition methods. In *Advanced Topics in Computational Partial Differential Equations*, pages 57–95. Springer, 2003.

- [CB95] Q. Chen and I. Babuška. Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle. *Computer Methods in Applied Mechanics and Engineering*, 128(3):405–417, 1995.
- [CHQ<sup>+</sup>12] C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, et al. *Spectral methods in fluid dynamics*. Springer Science & Business Media, 2012.
- [CHQZ10] C. G. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral methods: Fundamentals in single domains*. Springer, 2010.
- [Chr09] D. S. Christen. *Praxiswissen der chemischen Verfahrenstechnik: Handbuch für Chemiker und Verfahreningenieure*. Springer-Verlag, 2009.
- [CLT16] S.-S. Chen, B.-W. Li, and X.-Y. Tian. Chebyshev collocation spectral domain decomposition method for coupled conductive and radiative heat transfer in a 3d l-shaped enclosure. *Numerical Heat Transfer, Part B: Fundamentals*, 70(3):215–232, 2016.
- [CM<sup>+</sup>94] T. F. Chan, T. P. Mathew, et al. Domain decomposition algorithms. *Acta numerica*, 3(1):61–143, 1994.
- [Dör96] W. Dörfler. A convergent adaptive algorithm for poisson’s equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, 1996.
- [DPR08] V. Dolean, R. Pasquetti, and F. Rapetti. p-multigrid for fekete spectral element method. In *Domain Decomposition Methods in Science and Engineering XVII*, pages 485–492. Springer, 2008.
- [EMR70] E. Eliassen, B. Machenhauer, and E. Rasmussen. *On a numerical method for integration of the hydrodynamical equations with a spectral representation of the horizontal fields*. Kobenhavns Universitet, Institut for Teoretisk Meteorologi, 1970.
- [FS66] B. Finlayson and L.E. Scriven. The method of weighted residuals - a review. *Appl. Mech. Rev.*, 19:735–748, 01 1966.
- [Gau04] W. Gautschi. *Orthogonal polynomials: Computation and approximation*. clarendon, 2004.
- [GK02] E. Giladi and H.B. Keller. Space-time domain decomposition for parabolic problems. *Numerische Mathematik*, 93(2):279–313, 2002.

- [GO77] D. Gottlieb and S. A. Orszag. *Numerical analysis of spectral methods: theory and applications*, volume 26. Siam, 1977.
- [GT06] F. X. Giraldo and M. A. Taylor. A diagonal-mass-matrix triangular-spectral-element method based on cubature points. *Journal of Engineering Mathematics*, 56(3):307–322, 2006.
- [Hei98] W. Heinrichs. Spectral collocation on triangular elements. *Journal of Computational Physics*, 145(2):743–757, 1998.
- [Hei06] W. Heinrichs. An adaptive least-squares spectral collocation method with triangular elements for the incompressible navier–stokes equations. *Journal of engineering mathematics*, 56(3):337–350, 2006.
- [Hen99] R. D. Henderson. Adaptive spectral element methods for turbulence and transition. In *High-order methods for computational physics*, pages 225–324. Springer, 1999.
- [HG07] J. S. Hesthaven and D. Gottlieb, S. and Gottlieb. *Spectral methods for time-dependent problems*, volume 21. Cambridge University Press, 2007.
- [HL01] W. Heinrichs and B. I. Loch. Spectral schemes on triangular elements. *Journal of Computational Physics*, 173(1):279–301, 2001.
- [KO72] H.-O. Kreiss and J. Oliger. Comparison of accurate methods for the integration of hyperbolic equations. *Tellus*, 24(3):199–215, 1972.
- [KS05] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp methods for computational fluid dynamics*. Oxford University Press, 2005.
- [Lan38] C. Lanczos. Trigonometric interpolation of empirical and analytical functions. *Studies in Applied Mathematics*, 17(1-4):123–199, 1938.
- [Lin09] T. Linß. *Layer-adapted meshes for reaction-convection-diffusion problems*. Springer, 2009.
- [LTB17] Y. Liu, T. Teng, J. and Xu, and J. Badal. Higher-order triangular spectral element method with optimized cubature points for seismic wavefield modeling. *Journal of Computational Physics*, 336:458–480, 2017.
- [Mav89] C. Mavriplis. *Nonconforming discretizations and a posteriori error estimators for adaptive spectral element techniques*. PhD thesis, Massachusetts Institute of Technology, 1989.

- [Mav92] C. Mavriplis. Adaptive mesh strategies for the spectral element method. Technical report, Institute for Computer Applications in Science and Engineering, Hampton VA, 1992.
- [MP89] Y. Maday and A. T. Patera. Spectral element methods for the incompressible navier-stokes equations. *IN: State-of-the-art surveys on computational mechanics (A90-47176 21-64)*. New York, pages 71–143, 1989.
- [Ors69] S. A. Orszag. Numerical methods for the simulation of turbulence. *The Physics of Fluids*, 12(12):II–250, 1969.
- [Ors71] S. A. Orszag. Accurate solution of the orr–sommerfeld stability equation. *Journal of Fluid Mechanics*, 50(4):689–703, 1971.
- [Ors72] S. A. Orszag. Comparison of pseudospectral and spectral approximation. *Studies in Applied Mathematics*, 51(3):253–259, 1972.
- [Pat84] A. T. Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of computational Physics*, 54(3):468–488, 1984.
- [Poz] C. Pozrikidis. Introduction to finite and spectral element methods using matlab. 2005. *Chapman & amp.*
- [PR04] R. Pasquetti and F. Rapetti. Spectral element methods on triangles and quadrilaterals: comparisons and applications. *Journal of Computational Physics*, 198(1):349–362, 2004.
- [PRPZ06] R. Pasquetti, F. Rapetti, L. Pavarino, and E. Zampieri. Neumann–neumann–schur complement methods for fekete spectral elements. *Journal of engineering mathematics*, 56(3):323–335, 2006.
- [Qua87] A. Quarteroni. Domain decomposition techniques using spectral methods. *Calcolo*, 24(2):141–177, 1987.
- [QV99] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Number BOOK. Oxford University Press, 1999.
- [QV08] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*, volume 23. Springer Science & Business Media, 2008.
- [RL18] E. M. Rudoy and N.P. Lazarev. Domain decomposition technique for a model of an elastic body reinforced by a timoshenko’s beam. *Journal of Computational and Applied Mathematics*, 334:18–26, 2018.

- [Sil54] I. Silberman. Planetary waves in the atmosphere. *Journal of Meteorology*, 11(1):27–34, 1954.
- [Smi96] BF Smith. Domain decomposition: parallel multilevel methods for elliptic partial differential equations/smith bf, bjordstad pe, gropp w, 1996.
- [Tre00] L. N. Trefethen. *Spectral methods in MATLAB*. SIAM, 2000.
- [TW99] M. A. Taylor and B. A. Wingate. Fekete collocation points for triangular spectral elements. *SIAM Journal of Numerical Analysis*, 1999.
- [TW06] A. Toselli and O. Widlund. *Domain decomposition methods-algorithms and theory*, volume 34. Springer Science & Business Media, 2006.
- [TWW00] M. A. Taylor, B. A. Wingate, and R. E. Vincent. An algorithm for computing fekete points in the triangle. *SIAM Journal on Numerical Analysis*, 38(5):1707–1720, 2000.
- [Wel97] B. D. Welfert. Generation of pseudospectral differentiation matrices i. *SIAM Journal on Numerical Analysis*, 34(4):1640–1657, 1997.
- [Zan87] P. Zanolli. Domain decomposition algorithms for spectral methods. *Calcolo*, 24(3-4):201–240, 1987.