Adaptive finite Dünngitter-Elemente höherer Ordnung für elliptische partielle Differentialgleichungen mit variablen Koeffizienten

Stefan Achatz

Institut für Informatik
der Technischen Universität München

Lehrstuhl für numerische Programmierung und
Ingenieuranwendungen in der Informatik

Adaptive finite Dünngitter-Elemente höherer Ordnung für elliptische partielle Differentialgleichungen mit variablen Koeffizienten

Stefan Achatz

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr. Ernst W. Mayr
Prüfer der Dissertation:	
	1. Univ.-Prof. Dr. Christoph Zenger
	2. Univ.-Prof. Dr. Hans–Joachim Bungartz, Universität Stuttgart
	3. Univ.-Prof. Dr. Harry Yserentant, Eberhard-Karls-Universität Tübingen, (schriftliche Beurteilung)

Die Dissertation wurde am 22.01.2003 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 31.03.2003 angenommen.

Zusammenfassung:

Die Methode der Finiten Elemente (FEM) wird zur numerischen Lösung von partiellen Differentialgleichungen bzw. Randwertproblemen vorwiegend im ingenieurstechnischen Bereich verwendet. Die Entwicklung der FEM wurde stets angetrieben von zwei Bestrebungen: Erstens, das Verhältnis von erreichbarer Genauigkeit und dafür zu investierender Anzahl von Freiheitsgraden zu optimieren, und zweitens, die diskreten Gleichungen möglichst effizient zu lösen, bestenfalls mit konstantem Speicher- und Rechenaufwand je Freiheitsgrad.

Dünne Gitter bzw. darüber definierte Ansatzräume bedienen zunächst den ersten Bereich: Aus einer hierarchischen Tensorproduktbasis des Sobolevraums $H^1([0,1]^d)$ werden endliche Teilmengen gebildet, so dass die resultierenden diskreten Funktionenräume das angesprochene Kosten-Nutzen-Verhältnis optimieren. In der H^1-Norm liefern dünne Gitter Fehler von der Ordnung $O(N^{-p})$, wo traditionelle finite Elemente Fehler der Ordnung $O(N^{-p/d})$ produzieren (N ist die Anzahl der Freiheitsgrade, p die Ordnung des Ansatzraums, d die Dimension). Zu den wesentlichen Fortschritten innerhalb des letzten Jahrzehnts gehören die Erweiterung des Dünngitteransatzes auf Basen höherer Ordnung und lokal adaptive Gitter auf Basis von a-posteriori-Fehlerschätzungen.

Bedingt durch den Tensorproduktansatz ist die Dünngittertechnik zunächst nur für einfache Gebiete (Rechtecke, Quader) und einfache Differentialoperatoren (Laplaceoperator) anwendbar. Ein erster Ansatz, für eine Erweiterung auf krummberandete Gebiete und Differentialoperatoren mit variablen Koeffizienten stammt aus dem Jahr 1996 (Dornseifer) und ist auf lineare Ansatzfunktionen beschränkt.

Die vorliegende Arbeit stellt erstmals ein Finite-Element-Verfahren vor, das es erlaubt, elliptische Randwertprobleme zweiter Ordnung mit variablen Koeffizienten über glattberandeten Gebieten mit Hilfe adaptiver dünner Gitter bzw. darüber definierten Ansatzräumen höherer Ordnung zu lösen. Die Konsistenz der hierfür modifizierten Bilinearform wird bewiesen, die Stabilität ausführlich diskutiert. An Hand einiger Beispiele wird gezeigt, dass die numerische Lösung qualitativ das gleiche Fehlerverhalten aufweist, wie es für die Laplacegleichung über dem Quadrat (Würfel) bekannt ist. Ein interessantes Ergebnis ist, dass Dünngitterelemente höherer Ordnung von einer lokalen Gitteradaption profitieren, selbst wenn es glatte Funktionen, also solche ohne Singularitäten, zu approximieren gilt. Des weiteren wird ein Multilevelverfahren vorgestellt, das als Vorkonditionierer für das `BiCGstab`-Verfahren eingesetzt wird und für eine effiziente Lösung des linearen Gleichungssystems sorgt.

Inhaltsverzeichnis

1 Einführung **1**

2 Die Methode der Finiten Elemente **5**
2.1 Grundlagen 5
2.1.1 Schwache Formulierung von Randwertaufgaben 5
2.1.2 Galerkin-Diskretisierung 7
2.2 Finite Elemente 8
2.2.1 Ansatzräume aus finiten Elementen 8
2.2.2 h-, p- und hp-Version der FEM 11
2.2.3 Das lineare Gleichungssystem und seine Lösung 12
2.3 Auf dem Weg zu Hierarchischen Tensorproduktelementen 15

3 Hierarchische Tensorproduktelemente **21**
3.1 Hierarchische Teilraumzerlegung für das Referenzelement 21
3.2 Interpolation mit hierarchischen Basen 24
3.3 Optimale Teilräume und Dünne Gitter 26
3.4 Polynomiale Elemente höherer Ordnung 37
3.5 Lokal adaptive Elemente 44
3.6 Algorithmische Betrachtungen 50
3.6.1 Auswertung bestimmter Integraloperatoren 50
3.6.2 Erweiterte Funktionale 59
3.6.3 Basistransformation 60
3.6.4 Ein diskreter Differentialoperator 61
3.6.5 Transponierte Operatoren 61
3.7 Datenstruktur 62

4 FEM mit hierarchischen Tensorproduktelementen **65**
4.1 FE-Diskretisierung mit hierarchischen Tensorproduktelementen 65
4.2 Berechnung des Matrix-Vektor-Produkts $\boldsymbol{Au}$ 68
4.2.1 Affin-lineare Transformationen und Differentialoperatoren mit konstanten Koeffizienten 70
4.2.2 Erweiterung für Koeffizienten mit Tensorproduktеigenschaft . . 70
4.2.3 Variable Koeffizienten und krumm berandete Elemente 71
4.3 Konvergenzbetrachtungen 77
4.4 Bemerkungen zur Symmetrisierung 88

5 Numerische Ergebnisse **95**
5.1 Diskrete Fehlernormen 95
5.2 Variable Koeffizienten 96
5.2.1 Ein 2D-Beispiel 96
5.2.2 Ein 3D-Beispiel 102
5.2.3 Unstetige Koeffizienten 105
5.3 Krumm berandete Gebiete 107
5.3.1 Potentialströmung um einen Kreiszylinder (2D) 107
5.3.2 Ein 3D-Beispiel 116

6 Ein Mehrgitter-Vorkonditionierer **119**
6.1 Unterraum-Korrektur-Verfahren 119
6.2 Unterraum-Zerlegung für Dünngitterelemente 121
6.3 Kosten für einen Mehrgitterzyklus 125
6.4 Adaptive Elemente und mehrelementige Diskretisierungen 126
6.5 Konvergenzraten in der Praxis 127

7 Zusammenfassung und Ausblick **133**
7.1 Was wurde erreicht? 133
7.2 Anregungen für weitere Arbeiten 134

1 Einführung

Die Methode der Finiten Elemente (FEM) ist eines der gebräuchlichsten Diskretisierungsschemata für partielle Differentialgleichungen bzw. Randwertprobleme über stückweise glatt berandeten Gebieten. Ursprünglich für die numerische Analyse in der Strukturmechanik entwickelt, findet sie heute praktisch in allen Bereichen des Ingenieurwesens und der Physik Verwendung, deren Modelle auf partiellen Differentialgleichungen beruhen.

Die Anfänge der FEM gehen zurück auf das Ende der 1950er Jahre, als Ingenieure nach numerischen Verfahren suchten, um Strukturanalysen für den Flugzeugbau durchzuführen. Grundidee war, das Berechnungsgebiet in kleine, geometrisch einfache Teilgebiete (Dreiecke oder Vierecke) zu zerlegen und die physikalischen Gleichungen bestehend aus Energie- und Kräftebilanzen als Summe über diese sogenannten Elemente zu formulieren. Die physikalischen Eigenschaften eines Elements wurden dabei durch eine kleine Zahl von Freiheitsgraden modelliert. Die FEM war damit zunächst eher ein diskretes Modell für einen physikalischen Sachverhalt.

Der Weg zum Verständnis der FEM als eigenständiges, modellunabhängiges Diskretisierungswerkzeug für partielle Differentialgleichungen wurde in den 60er Jahre beschritten, als die FEM in den Zusammenhang mit klassischen Variations- und Minimierungsverfahren (Ritz-Galerkin-Verfahren) gebracht wurde. Das darauf gegründete mathematische Fundament verhalf der FEM zu einem raschen Aufstieg. Die in ihrer Natur sehr allgemeinen Fehlerabschätzungen waren richtungsweisend für die Entwicklung immer leistungsfähigerer FE-Verfahren. Die wesentliche Erkenntnis ist, dass die globale Approximationsgüte allein durch die lokalen Approximationseigenschaften der Elemente gegeben ist.

Prinzipiell gibt es zwei Möglichkeiten, die Genauigkeit einer Finite-Element-Diskretisierung zu steuern: Zum einen über die Feinheit des Gitters (h-Methode), zum anderen über den Grad der Polynome über den Elementen (p-Methode). Moderne FE-Verfahren kombinieren beide Strategien und arbeiten zudem lokal adaptiv, d.h. Maschenweite und Polynomgrad werden je nach lokaler Regularität der Lösung passend gewählt. Die Verfeinerung wird dabei von a-posteriori-Fehlerschätzern gesteuert. Verfahren, die h-Adaptivität und höhere Ordnung miteinander verbinden, sind allerdings noch mit einigen Nachteilen oder Hindernissen behaftet: Die Implementierung ist auf Grund der komplizierten Datenstrukturen und Algorithmen sehr aufwändig. Die Gitterverfeinerung gestaltet sich für die Viereckselemente, die bei Elementen höherer Ordnung bevorzugt werden, schwierig. Ferner wird der Idealfall für die Konvergenz der FE-Lösung, ein in der Zahl der Freiheitsgrade exponentiell abfallender Fehler, nur er-

reicht, wenn die Lösung des kontinuierlichen Problems hohen Regularitätsansprüchen genügt, etwa wenn sie stückweise analytisch ist. Schließlich fehlen bislang effiziente Löser für das diskrete Gleichungssystem.

Der Entwicklung von Lösern wurde ab Ende der 70er Jahre besondere Aufmerksamkeit geschenkt. Die Notwendigkeit hierzu entstand durch die sich rasch vergrößernden Speicher und damit das Verlangen, immer größere Systeme zu lösen. Aus der heutigen Sicht ist die damals begonnene und heute stärker denn je betriebene Suche nach effizienten Lösern bzw. Vorkonditionierern nicht nur sinnvoll, sondern vielmehr zwingend. Effizient heißt dabei, dass die Rechenzeit (möglichst) linear in der Zahl der Unbekannten skaliert. Die Notwendigkeit ergibt sich aus der Beobachtung, dass Rechenleistung und Speicherkapazität ungefähr gleich schnell wachsen, sie verdoppeln sich etwa alle 18 Monate. Da man stets den gesamten verfügbaren Speicher ausreizen will, kann nur durch eine $O(N)$-Komplexität des Lösers gewährleistet werden, dass die absoluten Rechenzeiten zukünftig nicht ansteigen.

Die Basis von $O(N)$-Lösern bilden in der Regel Mehrgittermethoden, bei denen das Problem auf einer Skala von verschieden feinen, ineinander eingebetteten Gittern gelöst wird. Anfangs waren Mehrgitterverfahren auf eine kleine Klasse von Modellproblemen und auf strukturierte Gitter beschränkt. Im Laufe der Zeit wurden Anpassungen an größere Problemklassen – man spricht in diesem Zusammenhang von robusten Mehrgitterverfahren –, unstrukturierte, insbesondere adaptive Gitter und Diskretisierungen höherer Ordnung entwickelt. Bislang ist es allerdings nicht gelungen, alle Konzepte unter einen Hut zu bringen, also ein für relativ allgemeine Probleme und Diskretisierungen gleichermaßen geeignetes Verfahren zu entwickeln. Ein Ansatzpunkt sind hier die algebraischen Mehrgitterverfahren, die auf die Ausnutzung von problemspezifischen und daher a-priori bekannten Zusammenhängen verzichten und Unterraum-Hierarchien allein auf Grund von algebraischen Eigenschaften des linearen Gleichungssystems bestimmen.

Die Konstruktion von Mehrgitterverfahren erweist sich immer dann als schwierig, wenn durch die Diskretisierung keine natürliche Hierarchie von Teilräumen gegeben ist. Teilraumhierarchien sind automatisch gegeben, wenn hierarchische Basen an Stelle der für klassische FE-Diskretisierungen typischen nodalen Basen eingesetzt werden. Yserentant [54, 55] wählt Mitte der 80er Jahre eine hierarchische Basis für lineare Dreieckselemente und zeigt zunächst, dass die Steifigkeitsmatrix bezüglich dieser Basis eine Konditionszahl hat, die mit kleiner werdender Maschenweite nur unwesentlich steigt. Bank [10, 9] benutzte diese hierarchische Basis für einen Mehrgitter-Löser in seinem Programm `PLTMG`.

In die Zeit, als der Einsatz der hierarchischen Basis für Finite Elemente propagiert wurde, fällt auch die Geburtsstunde der *dünnen Gitter*: Zenger [56] stellt 1991 ein einfaches Diskretisierungs- oder Interpolationsschema vor, basierend auf der (eindimensionalen) hierarchischen Hutbasis und einem Tensorproduktansatz für höhere Dimension. Durch eine „dünne Tensorierung“, bei der von der Hutbasis über einem Gitter der Maschenweite h nur bestimmte Basisfunktionen übernommen werden, wird erreicht, dass die Anzahl N der Freiheitsgrade nur $O(h^{-1} \cdot (\log h^{-1})^{d-1})$ an Stelle von $O(h^{-d})$ beträgt, während der Interpolationsfehler in der L_∞-Norm sich mit $O(h^2 \cdot (\log h^{-1})^{d-1})$

gegenüber $O(h^2)$ für volle Gitter nur unwesentlich verschlechtert. Äquivalente Interpolationsformeln wurden bereits 1963 von Smolyak [42] angegeben, wobei hier die numerische Quadratur hochdimensionaler Funktionen als Anwendungsfeld im Vordergrund stand. Tatsächlich ist es mit Hilfe dünner Gitter möglich, den Fluch der Dimension nahezu zu bannen: Anstelle der sonst beobachteten exponentiellen Abhängigkeit des Interpolationsfehlers von der Anzahl der investierten Gitterpunkte, geht hier die Dimension nur schwach, d.h. als Exponent von logarithmischen Termen, oder sogar überhaupt nicht ein [18, 52].

Innerhalb des vergangenen Jahrzehnts wurden dünne Gitter auf eine Reihe von Modellproblemen erfolgreich angewandt, hierunter die zwei- und dreidimensionale Poissongleichung [14], parabolische Gleichungen [5], die Helmholtzgleichung [6], die biharmonische Gleichung [46], die Stokes-Gleichung [38], gemischte Probleme [25] und elliptische Probleme mit stochastischen Eingabedaten [40], um nur einige Beispiele zu nennen. Zu den wichtigsten Errungenschaften gehört die hierarchische Basis aus Polynomen höheren Grades, mit Hilfe derer Bungartz [17] finite Elemente höherer Ordnung über dünnen Gittern konstruiert. In Abhängigkeit von der Anzahl der investierten Freiheitsgrade bzw. Gitterpunkte ist hier der Fehler bezüglich der Energienorm von der Ordnung $O(N^{-p})$. Zum Vergleich: Bei Vollgitter-Diskretisierungen ergeben sich Fehler der Ordnung $O(N^{-p/d})$, wobei d die Dimension ist.

Ein wesentlicher Bestandteil der Dünngittertechnik ist der Tensorproduktansatz. Er erlaubt, dass die Auswertung der für die Umsetzung der FEM benötigten Integrale auf eindimensionale Integrale zurückgeführt werden können. Dieses unidirektionale Prinzip gewährleistet, dass die FE-Operatoren angewandt auf Dünngitterfunktionen mit konstantem Rechen- und Speicheraufwand je Freiheitsgrad ausgewertet werden können. Allerdings ist das unidirektionale Prinzip nur für Differentialoperatoren mit konstanten Koeffizienten verwendbar. Ein erster Ansatz für eine Erweiterung auf variable Koeffizienten stammt von Dornseifer [22] und ist auf lineare Ansatzfunktionen beschränkt.

Eine weitere Folge des Tensorproduktansatzes ist, dass Dünngitter-Verfahren zunächst nur für Probleme über dem d-dimensionalen Einheitsintervall $[0,1]^d$ definiert sind. Über Abbildungstechniken können Gebiete, die durch eine Transformation aus dem Einheitsintervall hervorgehen, behandelt werden. Kompliziertere Gebiete lassen sich meist durch Blockstrukturierung in Teilgebiete zerlegt werden, die sich dann auf das Einheitsintervall abbilden lassen. Eine dazu gleichwertige Sichtweise besteht darin, das Einheitsintervall bzw. die darüber definierten Ansatzräume als Referenzelement für eine FE-Diskretisierung aufzufassen. Eine genaue Beschreibung dieses *Finite-Element-Verfahrens mit Dünngitterelementen* ist Ziel dieser Arbeit, die wie folgt aufgebaut ist:

Kapitel 2 wiederholt die wichtigsten Konzepte, die der Methode der Finiten Elemente zu Grunde liegen. Hierzu gehören die theoretischen Grundlagen wie die schwache Formulierung von Randwertproblemen und abstrakte Fehlerabschätzungen, die Zusammensetzung von Ansatzräumen aus finiten Elementen, verschiedene Verfeinerungsstrategien und implementationstechnische Gesichtspunkte. Ferner wird im Abschnitt 2.3 versucht, eine Brücke von den traditionellen Finiten Elementen hin zu den hierarchischen Tensorproduktelementen zu schlagen.

Kapitel 3 beschreibt die hierarchischen Tensorproduktelemente oder Dünngitterelemente. Aufbauend auf das Prinzip der hierarchischen Teilraumzerlegung wird das Prinzip der dünnen Gitter als optimierte Zusammensetzung des Ansatzraums aus Teilräumen erläutert. Die Darstellung folgt weitestgehend der Arbeit [17]. Allerdings werden dort nur homogene Randdaten behandelt. In der vorliegenden Arbeit werden erstmals die entsprechenden Fehlerabschätzungen für inhomogene Randdaten vorgestellt und bewiesen. Die Standardalgorithmen für die Auswertung gewisser Integrale über dünnen Gittern werden der Vollständigkeit halber wiederholt und um eine Bemerkung über die Realisierung transponierter Operatoren ergänzt.

In **Kapitel 4** wird beschrieben, wie mit Hilfe von Dünngitterelementen ein effizientes Finite-Element-Verfahren gewonnen wird. Hauptziel und zentrales Thema dieser Arbeit ist die Behandlung von Differentialoperatoren, die (eventuell erst nach der Transformation auf das Einheitsgebiet) variable Koeffizienten besitzen. Es wird eine modifizierte Version der Bilinearform in der schwachen Formulierung des Randwertproblems angegeben, die eine Multiplikation mit der Steifigkeitsmatrix mit konstantem Aufwand je Freiheitsgrad unter Verwendung von Standardalgorithmen erlaubt. Anschließend wird die Konsistenz gezeigt, die Stabilität wird ausführlich diskutiert.

Numerische Experimente sind der Inhalt von **Kapitel 5**. An Hand einer Reihe von Randwertproblemen in 2D und 3D, für die die exakte Lösung bekannt ist, wird die Güte der vorgestellten Diskretisierung herausgearbeitet.

In **Kapitel 6** wird ein Mehrgitter-Vorkonditionierer für lokal adaptive Dünngitterelemente höherer Ordnung vorgestellt, der in Verbindung mit dem `BiCGstab`-Verfahren bei den numerischen Beispielen für gute Konvergenzraten gesorgt hat. Im Gegensatz zu früheren Arbeiten werden hier der Ansatz- und Testraum gleich gewählt.

Abschließend werden in **Kapitel 7** die wichtigsten Ergebnisse zusammengetragen und ein Ausblick auf die zukünftige Arbeit gegeben.

An dieser Stelle möchte ich mich bei allen bedanken, die mich bei der Erstellung dieser Arbeit unterstützt haben. Mein ganz besonderer Dank gilt meinem Doktorvater, Prof. Dr. Christoph Zenger. Seine unzähligen Ideen und Ratschläge haben maßgeblich zum Gelingen dieser Arbeit beigetragen. Nicht zuletzt durch seinen ungebremsten Optimismus hat er mir viel Freude am Forschen vermittelt und sie auch in weniger erfolgreichen Zeiten erhalten. Meinen Kollegen Dr. Christoph Kranz, Max Emans und Dr. Michael Bader danke ich dafür, dass sie meine Arbeit bzw. Teile davon mit großer Sorgfalt Korrektur gelesen haben. Den Kollegen von der Systemadministration, Markus Pögl und Alexander Mors, möchte ich für die rasche Hilfe bei technischen Problemen danken. Mein ehemaliger Kollege Dr. Stefan Schneider hat mir viele hilfreiche Tipps in Sachen dünne Gitter und für die Implementierung der Algorithmen gegeben. Dafür auch ihm ein herzliches Dankeschön! Während meiner Zeit als Doktorand war ich vollbeschäftigter Angestellter der Technischen Universität München. Für die mir dadurch zuteil gewordene finanzielle Unterstützung und technische Ausstattung möchte ich mich bei den verantwortlichen Personen bedanken.

2 Die Methode der Finiten Elemente

Dieses Kapitel bietet einen Überblick über die wesentlichen Konzepte der Methode der Finiten Elemente (FEM). Für eine weitergehende Beschreibung sei auf die einschlägige Literatur verwiesen [13, 20, 32, 47].

2.1 Grundlagen

Die Methode der Finiten Elemente basiert auf der schwachen Formulierung von Randwertproblemen, die im Folgenden für den Fall reiner Dirichlet-Randwerte vorgestellt wird. Ferner werden hinreichende Voraussetzungen für die Existenz einer schwachen Lösung angegeben. Im Anschluss wird auf den Galerkin-Ansatz als natürliche Diskretisierung der schwachen Gleichungen eingegangen und die Abschätzung des Fehlers bezüglich der Energienorm mit dem Lemma von Céa angegeben.

2.1.1 Schwache Formulierung von Randwertaufgaben

Wir betrachten im Folgenden die lineare Randwertaufgabe zweiter Ordnung, gegeben durch die partielle Differentialgleichung

$$-\sum_{i,j=1}^{d} \partial_j(a_{ij}\partial_i u) \;+\; \sum_{i=1}^{d} b_i \partial_i u \;+\; cu \;=\; f \qquad \text{in } \Omega \subset \mathbb{R}^d, \tag{2.1}$$

und der *Dirichlet-Randbedingung*

$$u \;=\; g \quad \text{auf} \quad \Gamma := \partial\Omega. \tag{2.2}$$

Auf andere Randbedingungen, wie etwa die *Neumann-Randbedingung*, bei der die Normalenableitung auf dem Rand vorgeschrieben wird, wird in dieser Arbeit nicht eingegangen. Die Koeffizienten a_{ij}, b_i, c, sowie die rechte Seite f seien Funktionen über Ω, g sei eine Funktion über Γ. Die Differentialgleichung sei ferner vom *elliptischen Typ*, das heißt, es gibt ein $\beta > 0$, so dass

$$\sum_{i,j=1}^{d} a_{ij}(\boldsymbol{x})\xi_i\xi_j \;\geq\; \beta \sum_{i=1}^{d} \xi_i^2 \quad \text{für alle } \boldsymbol{\xi} \in \mathbb{R}^d,\ \boldsymbol{x} \in \Omega. \tag{2.3}$$

Aussagen über die Existenz und Eindeutigkeit einer Lösung $u \in C^2(\Omega) \cap C(\bar{\Omega})$ lassen sich nur unter starken Regularitätsvoraussetzungen an die Koeffizienten und an das Gebiet Ω angeben [24].

Grundlage für die diskrete Lösung des Randwertproblems mit der Methode der Finiten Elemente ist die *schwache* oder *variationelle Formulierung* von (2.1)–(2.2). Sie lautet:

Finde $u \in V := \{w \in H^1(\Omega)$ *mit* $\gamma(w) = g\}$*, so dass für alle* $v \in H_0^1(\Omega)$ *gilt*

$$\mathcal{A}(u, v) \;=\; l(v), \tag{2.4}$$

wobei

$$\begin{aligned} \mathcal{A}(u, v) \;&:=\; \int_\Omega \left\{ \sum_{i,j=1}^{d} a_{ij}\, \partial_i u\, \partial_j v \;+\; \sum_{i=1}^{d} b_i\, \partial_i u\, v \;+\; cuv \right\} d\boldsymbol{x}\ , \\ l(v) \;&:=\; \int_\Omega f v\, d\boldsymbol{x}\ . \end{aligned} \tag{2.5}$$

$H^1(\Omega)$ und $H_0^1(\Omega)$ sind die *Sobolevräume* erster Ordnung von Funktionen mit inhomogenen bzw. homogenen Randwerten. $\gamma : H^1(\Omega) \to L_2(\Gamma)$ bezeichnet den Spuroperator. Eine ausführliche Darstellung der Sobolevräume findet sich in [1]. Mit der Bezeichnung „schwache Formulierung“ wird zum Ausdruck gebracht, dass die Regularitätsvoraussetzungen an die Lösung u geringer sind als bei der klassischen Formulierung (2.1)–(2.2). Zwar ist jede klassische Lösung auch eine schwache Lösung, genügt also der Gleichung (2.4). Andererseits ist eine schwache Lösung nicht notwendig eine Lösung im klassischen Sinne. Schwache Lösungen sind unter Umständen sogar nicht einmal differenzierbar. Allerdings ist eine schwache Lösung in $C^2(\Omega) \cap C(\bar{\Omega})$ auch eine klassische Lösung.

Existenz und Eindeutigkeit der schwachen Lösung

Die folgenden Voraussetzungen sind, sofern sie gleichzeitig erfüllt sind, hinreichend für die Existenz einer eindeutigen Lösung der schwachen Formulierung:

1. Ω ist beschränktes Lipschitz-Gebiet,
2. g liegt im Bild des Spuroperators γ,
3. $a_{ij},\ b_i,\ \sum_{i=1}^{d} \partial_i b_i,\ c \in L_\infty(\Omega),\ f \in L_2(\Omega)\ \ (1 \le i, j \le d)$,
4. (2.3) gilt fast überall,
5. $c - \frac{1}{2} \sum_{i=1}^{d} \partial_i b_i \ge 0$ fast überall in Ω.

Der Beweis erfolgt mit Hilfe des Lemmas von Lax und Milgram, siehe hierzu Kapitel 3.2 in [32].

Die zweite Forderung erlaubt es, das Problem auf ein äquivalentes Problem mit homogenen Randbedingungen zurückzuführen: Liegt g im Bild des Spuroperators, so existiert eine Funktion $\bar{g} \in H^1(\Omega)$ mit $\bar{g}|_\Gamma = g$. Die Funktion $u' := u - \bar{g} \in H_0^1(\Omega)$ genügt der Gleichung

$$\mathcal{A}(u', v) = l'(v) := l(v) - \mathcal{A}(\bar{g}, v) \qquad \text{für alle } v \in H_0^1(\Omega).$$

Umgekehrt ist $u = u' + \bar{g}$ Lösung der ursprünglichen Aufgabe. Vorteil dieser Form ist, dass damit die Asymmetrie in den Funktionenräumen für Lösung u' und Testfunktionen v behoben ist. Wir gehen deshalb im Folgenden von der symmetrischen Form aus, setzen $V := H_0^1(\Omega)$ und lassen die Apostrophe weg.

2.1.2 Galerkin-Diskretisierung

Der im letzten Abschnitt angesprochene Beweis für die Existenz und Eindeutigkeit einer schwachen Lösung ist abstrakter Natur. Er gibt die Lösung nicht an. Diskrete Lösungen erhält man, indem das Problem in einem endlich dimensionalen Teilraum $V_h \subset V$ gelöst wird. Das diskrete Problem lautet:
Bestimme $u_h \in V_h$ so, dass für alle $v \in V_h$ gilt

$$\mathcal{A}(u_h, v) \;=\; l(v). \tag{2.6}$$

Diese Diskretisierung wird nach ihrem Begründer *Galerkin-Diskretisierung* genannt. Darauf aufbauende Verfahren heißen *Galerkin-Verfahren* und unterscheiden sich nur in der Gestaltung des Ansatzraums V_h. Die Lösung u_h heißt *Galerkin-Approximation* von u in V_h.

Von zentraler Bedeutung für Galerkin-Verfahren ist eine einfache Abschätzung des Fehlers $u - u_h$ in der *Energienorm*. Unter der Energienorm wird in dieser Arbeit stets die H^1-Seminorm verstanden, die über $H_0^1(\Omega)$ eine Norm ist und gegeben ist durch

$$\|u\|_E \;:=\; \left(\sum_{i=1}^{d} \left\| \frac{\partial u}{\partial x_i} \right\|_{L_2}^2 \right)^{\frac{1}{2}} . \tag{2.7}$$

Die L_2-Norm $\|\cdot\|_{L_2}$ ist definiert durch

$$\|u\|_{L_2} \;:=\; \left(\int_\Omega |u(\boldsymbol{x})|^2 d\boldsymbol{x} \right)^{\frac{1}{2}} . \tag{2.8}$$

Die eben angesprochene Abschätzung des Fehlers in der Energienorm ist bekannt als das *Lemma von Céa*. Es besagt, dass es eine von V_h unabhängige Konstante $C > 0$ gibt, so dass

$$\|u - u_h\|_E \;\leq\; C \inf_{v \in V_h} \|u - v\|_E \, . \tag{2.9}$$

Für einen Beweis siehe [20], Satz 13.1. Die Abschätzung weist die Galerkin-Approximation abgesehen von dem Faktor C als Bestapproximation in V_h aus. Für die von der Bilinearform $\mathcal{A}$ induzierten Norm zeigt man sogar die echte Bestapproximationseigenschaft. Dazu muss $\mathcal{A}$ allerdings symmetrisch sein. Die Ungleichung (2.9) gilt auch im Falle einer nicht symmetrischen Bilinearform.

Eine konkrete Abschätzung des Fehlers bekommt man, wenn in (2.9) für v der Interpolant von u in V_h eingesetzt wird und a-priori-Abschätzungen für den Interpolationsfehler bekannt sind. Daher wird bei der Konstruktion von Ansatzräume V_h darauf geachtet, dass die Interpolation in V_h möglichst kleine Fehler erzeugt.

2.2 Finite Elemente

Der letzte Abschnitt hat gezeigt, dass die Wahl des Ansatzraums V_h ausschlaggebend ist für die Güte der Galerkin-Approximation u_h. Nach dem Lemma von Céa ist der Fehler $u - u_h$ umso kleiner, je umfangreicher oder feiner V_h ist. Im Folgenden sollen die wesentlichen Prinzipien bei der Konstruktion von Ansatzräumen aufgezeigt werden, wie sie bei der Methode der finiten Elemente zum Einsatz kommen.

2.2.1 Ansatzräume aus finiten Elementen

Die Sobolevräume erster Ordnung enthalten alle stetigen, stückweise polynomialen Funktionen. Finite-Elemente-Verfahren nutzen diese Tatsache für die Konstruktion des Ansatzraums:

Das Berechnungsgebiet Ω wird in geometrische Einheiten $K^{(i)}$, die *Elemente*, zerlegt,

$$\Omega \;=\; \bigcup_{i=1}^{N_E} K^{(i)} \;. \tag{2.10}$$

Die $K^{(i)}$ seien mit Ausnahme gemeinsamer Randpunkte paarweise disjunkt. In 2D sind Zerlegungen in Dreiecke oder konvexe Vierecke die Regel (vergleiche Abbildung 2.1), in 3D verwendet man entsprechend Tetraeder oder Hexaeder. Über den Elementen $K^{(i)}$ werden Funktionenräume $P^{(i)}$ definiert. Meist handelt es sich dabei um Polynomräume. Der Ansatzraum V_h ist dann gegeben durch

$$V_h \;=\; \{u \in C(\Omega),\; u|_{K^{(i)}} \in P^{(i)}\}\,.$$

In jedem Element werden $n^{(i)} = \dim P^{(i)}$ paarweise verschiedene Punkte, die *Knoten*, markiert. Jede Funktion $u \in V_h$ wird dann durch Vorgabe ihrer Funktionswerte an den Knoten eindeutig festgelegt. Einige der Knoten kommen auf dem Rand zu liegen, so dass aneinander grenzende Elemente gemeinsame Knoten besitzen und die dadurch gegebene Interpolationsbedingung die Stetigkeit der Ansatzfunktionen über die Elementgrenzen hinweg garantiert. Damit wird gewährleistet, dass der so erzeugte Ansatzraum V_h ein Teilraum von $V = H^1(\Omega)$ ist. Man spricht in diesem Zusammenhang von einer H^1-*konformen* Diskretisierung.

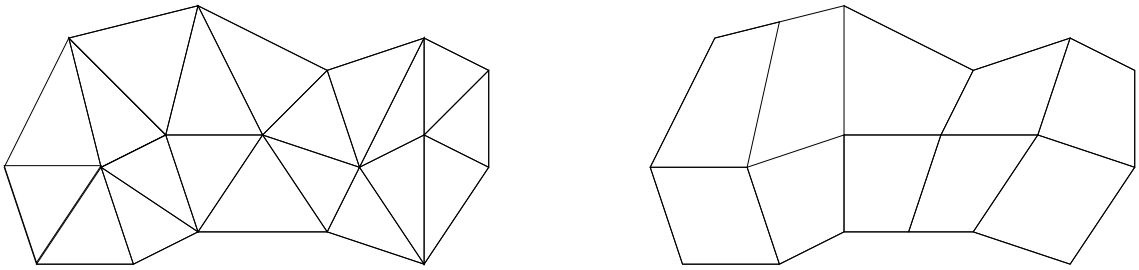


Abbildung 2.1: Zerlegung eines polygonal berandeten Gebiets in Dreiecke (links) und in Vierecke (rechts).

Ein Beispiel: Lineare Dreieckselemente

Standardbeispiel für die Diskretisierung mit finiten Elementen sind die linearen Dreieckselemente. Dazu wird ein polygonal berandetes Gebiet in Dreiecke $K^{(i)}$ aufgeteilt, wobei zwei Dreiecke höchstens eine gemeinsame Seite oder eine gemeinsame Ecke besitzen. Die linke Seite von Abbildung 2.1 gibt ein Beispiel für eine solche *Triangulierung*. Die lokalen Ansatzräume werden definiert durch $P^{(i)} := \{u : K^{(i)} \to \mathbb{R},\ u(x,y) = a^{(i)} + b^{(i)}x + c^{(i)}y\}$. Um die Stetigkeit der dadurch definierten Funktionen über Ω zu erzwingen, werden die $n^{(i)} = 3$ Knoten in die Ecken der Dreiecke gelegt, so dass jeweils alle Dreiecke mit einer gemeinsamen Ecke diese als Knoten besitzen. Da die Einschränkung einer Funktion $u \in P^{(i)}$ auf eine Dreiecksseite bereits durch die Werte an den beiden Endpunkten der Seite festgelegt wird, ist der stetige Übergang an den Dreiecksseiten gewährleistet.

Das Referenzelement

Die Formulierung von Elementtypen geschieht meist über die Definition eines sogenannten *Referenzelements* K, auf das sich die Elemente $K^{(i)}$ der Zerlegung (2.10) durch Transformationen $\psi^{(i)} : K \to K^{(i)}$ zurückführen lassen. Ein Referenzelement wird festgelegt durch einen Definitionsbereich $K \subset \mathbb{R}^d$, das die geometrische Gestalt festlegt, einen darüber definierten Ansatzraum P und eine Menge von Knoten $\boldsymbol{r}_j \in K$, $j = 1, \ldots, \dim P$. Die Elemente $K^{(i)}$ mit ihren lokalen Ansatzräumen $P^{(i)}$ und den Knoten $\boldsymbol{r}_j^{(i)}$ sind dann gegeben durch

$$\begin{aligned} K^{(i)} &= \psi^{(i)}(K), \\ P^{(i)} &= \{v : K^{(i)} \to \mathbb{R},\ v \circ \psi^{(i)} \in P\}, \\ \boldsymbol{r}_j^{(i)} &= \psi^{(i)}(\boldsymbol{r}_j), \quad j = 1, \ldots, \dim P. \end{aligned}$$

Die Funktionen $\phi_j \in P$ mit

$$\phi_j(\boldsymbol{r}_k) = \begin{cases} 1, & \text{falls } j = k, \\ 0, & \text{sonst} \end{cases} \tag{2.11}$$

bilden die *Knotenbasis* oder *nodale Basis* von P. Sie heißen im Zusammenhang mit finiten Elementen auch *Formfunktionen*.

Die nodale Basis des Ansatzraums

Sei $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots$ eine Aufzählung aller Knoten der Finite-Element-Diskretisierung. Jedem Knoten wird dann analog zu (2.11) eine Funktion $\varphi_j : \Omega \to \mathbb{R}$ zugeordnet, definiert durch

$$\varphi_j(\boldsymbol{r}_k) \;=\; \begin{cases} 1, & \text{falls } j = k, \\ 0, & \text{sonst.} \end{cases} \tag{2.12}$$

Die φ_j bilden dann die *nodale Basis* des Ansatzraums V_h. Die Koeffizienten bezüglich dieser Basis sind gerade die Funktionswerte an den Knoten.

Ein wichtiges Merkmal dieser Basis sind die lokalen Träger: Der Träger der Basisfunktion φ_j ist in der Vereinigung aller Elemente enthalten, die den Knoten $\boldsymbol{r}_j$ enthalten:

$$\operatorname{supp} \varphi_j \;\subset \bigcup_{i:\boldsymbol{r}_j \in K^{(i)}} K^{(i)} \tag{2.13}$$

Die Einschränkung von φ_j auf eines dieser Elemente ist dann bis auf die Transformation $\psi^{(i)}$ mit einer der Formfunktionen dieses Elements identisch.

Tensorproduktelemente

Im Folgenden soll eine einfache Klasse von finiten Elementen beschrieben werden, die im Hinblick auf die spätere Darstellung von Interesse sind. Gemeint sind die *Tensorproduktelemente*. Es handelt sich dabei um Rechteckselemente, genauer gesagt sind sie für allgemeine Dimension d über dem d-dimensionalen Würfel $K^{(d)} := [0,1]^d$ definiert. Die höherdimensionalen Elemente gehen dabei aus dem eindimensionalen Element durch Tensorproduktbildung hervor: Sei etwa über $K^{(1)} := [0,1]$ der Raum $P^{(1)} = \operatorname{span}\{\phi_i, 1 \le i \le n\}$ gegeben, dann wird $P^{(d)}$ von den n^d Tensorprodukten

$$\phi_{i_1} \otimes \cdots \otimes \phi_{i_d}, \qquad 1 \le i_j \le n \tag{2.14}$$

aufgespannt. Das Tensorprodukt ist definiert durch

$$\left(\bigotimes_{j=1}^{d} \phi_{i_j} \right)(\boldsymbol{x}) \;=\; \prod_{j=1}^{d} \phi_{i_j}(x_j) \quad \text{für } \boldsymbol{x} = (x_1, \ldots, x_d) \in K^{(d)}. \tag{2.15}$$

Für die univariaten Basisfunktionen ϕ_i werden meist die Lagrange'schen Basispolynome genommen, also

$$\phi_i(x) \;=\; \prod_{\substack{1 \le k \le n \\ k \ne i}} \frac{x - r_k}{r_i - r_k},$$

mit den Knoten $0 = r_1 < r_2 < \cdots < r_n = 1$. Diese werden oft äquidistant gewählt. Die Knoten des d-dimensionalen Elements sind dann durch das d-fache kartesische Produkt $\{r_1, \ldots, r_n\}^d$ gegeben. Ferner vergewissert man sich leicht, dass die Knoten auf dem Rand die Funktion dort eindeutig festlegen: Die Randhyperflächen des d-dimensionalen Elements sind jeweils zu einem $(d-1)$-dimensionalen Element äquivalent. Damit ist der

stetige Übergang an den Elementrändern gesichert und die H^1-Konformität gewährleistet.

Für $n = 2, 3, \ldots$ haben sich die Bezeichnungen *multilineares, multiquadratisches, ... Element* eingebürgert. Abbildung 2.2 zeigt ein biquadratisches Element.

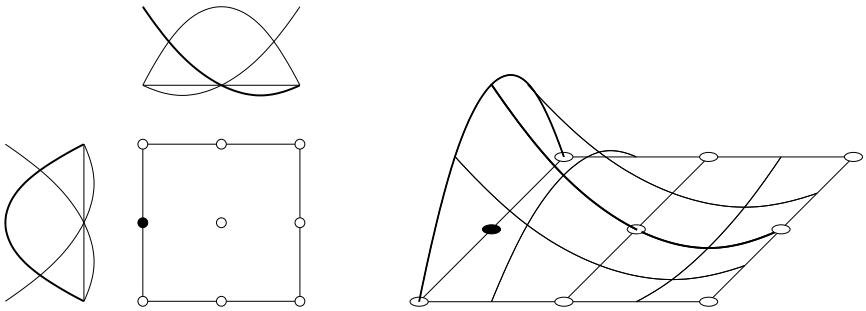


Abbildung 2.2: Das biquadratische Element und eine seiner Formfunktionen

2.2.2 h-, p- und hp-Version der FEM

Eine Finite-Element Berechnung erfolgt in der Regel auf einer Folge (V_h) von sukzessiv verfeinerten Ansatzräumen, die dann abgebrochen wird, wenn die Lösung hinreichend genau ist. Diese Verfeinerung des Ansatzraums kann dabei auf verschiedene Weise geschehen. Die hier angegebenen Aussagen über das Fehlerverhalten sind in [48] genauer ausgeführt.

h-Version

Das klassische Vorgehen besteht darin, die geometrische Zerlegung weiter zu unterteilen, also die Maschenweite h zu verringern, während der Elementtyp der gleiche bleibt. Dies ist die sogenannte *h-Version* der FEM. Für glatte Lösungen u ist bei gleichmäßiger Gitterverfeinerung das typische Konvergenzverhalten in der Anzahl N der Freiheitsgrade gegeben durch

$$\|u - u_h\|_E \;\leq\; C \cdot N^{-p/d}\,. \tag{2.16}$$

Dabei ist p der Polynomgrad und d die Raumdimension. Enthält u jedoch Singularitäten, die auch bei glatten Daten entstehen können (etwa wenn der Gebietsrand einspringende Ecken besitzt), bekommt man

$$\|u - u_h\|_E \;\leq\; C \cdot N^{-\beta/d}. \tag{2.17}$$

Hier ist $\beta := \min(p, \lambda)$, wobei λ von der Glattheit der Lösung abhängt. Die ursprüngliche Konvergenzordnung (2.16) kann aber durch lokale Verfeinerung des Gitters um die Singularität wieder hergestellt werden. Die Verfeinerung findet dabei entweder auf Grund von a-priori bekannten Informationen über die Lösung statt (sogenannte *a-priori-Adaption*, etwa bei Wurzelsingularitäten an einspringenden Ecken) oder auf

Grund von Informationen, die an Hand der diskreten Lösung auf gröberen Gittern gewonnen werden. Diese *a-posteriori-Adaption* ist in den letzten Jahren zu einem der zentralen Forschungsgebiete geworden, da man sich durch die Entwicklung genauer Fehlerschätzer und Adaptionskriterien zum einen eine Automatisierung der Gitterverfeinerung, zum anderen durch die optimale Gittergestaltung eine wesentliche Effizienzsteigerung erhofft [2, 3, 50]. Betrachten wir noch einmal die Abschätzungen (2.16) und (2.17). Charakteristisch für die h-Methode ist, dass das Konvergenzverhalten sich mit zunehmender Dimension d wesentlich verschlechtert. Dieser Umstand wird gewöhnlich als *Fluch der Dimension* bezeichnet.

p-Version

Die *p-Version* [39, 48] verfolgt die Strategie, die geometrische Zerlegung konstant zu lassen und stattdessen den Polynomgrad zu erhöhen. Hier beobachtet man für den Fehler ein asymptotisches Verhalten der Art

$$\|u - u_h\|_E \;\leq\; C \cdot \exp(-\gamma N^{1/\alpha_d}) \tag{2.18}$$

mit $\alpha_1 = 2$, $\alpha_2 = 3$, $\alpha_3 = 5$ und einer Zahl $\gamma > 0$, die von der Glattheit der Lösung abhängt. Für glatte Funktionen ist $\gamma = O(1)$ und die p-Version ist auf Grund des exponentiellen Abfallverhaltens des Fehlers im Vergleich zur h-Version mit seinem polynomialen Verhalten das wesentlich effizientere Verfahren. Liegen jedoch Singularitäten vor, ist γ unter Umständen sehr klein. Vielmehr beobachtet man dann wieder ein polynomiales Fehlerverhalten in N wie in (2.17).

hp-Version

Das hervorragende Approximationsverhalten der p-Version für glatte Lösungen u lässt sich auch für solche mit Singularitäten erhalten, indem p- und h-Methode entsprechend kombiniert werden, also gleichzeitig der Polynomgrad erhöht und die Maschenweite lokal verringert wird. Tatsächlich kann dadurch die Konvergenzordnung (2.18) mit $\gamma = O(1)$ wieder hergestellt werden. Allerdings gestaltet sich eine exakte, der Theorie entsprechende Gitterverfeinerung bei Elementen höherer Ordnung meist sehr schwierig. In der Regel werden deshalb Gitter verwendet, deren Maschenweite mit einem bestimmten Faktor in Richtung von a-priori bekannten Singularitäten abfallen (*graded mesh*). Insbesondere Rand- und Interfacesingularitäten können so erfolgreich behandelt werden [39].

2.2.3 Das lineare Gleichungssystem und seine Lösung

Um die Galerkin-Approximation u_h in einem vorgegebenen Ansatzraum V_h zu berechnen, muss ein lineares Gleichungssystem gelöst werden. Sei hierzu $B_h = \{\varphi_1, \ldots, \varphi_{N_h}\}$ eine Basis von V_h. Dann ist die schwache Formulierung (2.6) des diskreten Problems äquivalent zu

$$\boldsymbol{A}_h \boldsymbol{u}_h \;=\; \boldsymbol{b}_h \tag{2.19}$$

mit der Matrix $\boldsymbol{A}_h \in \mathbb{R}^{N_h \times N_h}$ und dem Vektor $\boldsymbol{b}_h \in \mathbb{R}^{N_h}$. Sie sind gegeben durch die Einträge

$$\begin{aligned} A_{h,ij} &= \mathcal{A}(\varphi_j, \varphi_i), \quad 1 \leq i, j \leq N_h, \\ b_{h,i} &= l(\varphi_i), \qquad\quad 1 \leq i \leq N_h. \end{aligned}$$

Die Matrix $\boldsymbol{A}_h$ wird *Systemmatrix* oder *Steifigkeitsmatrix*, der Vektor $\boldsymbol{b}$ *Lastvektor* genannt. Die Lösung u_h ist dann gegeben durch

$$u_h = \sum_{j=1}^{N_h} u_{h,j} \varphi_j \ ,$$

wobei die $u_{h,j}$ die Koeffizienten des Vektors $\boldsymbol{u}_h$ sind.

Betrachten wir nun das lineare Gleichungssystem, das es zu lösen gilt, wenn finite Elemente für die Galerkin-Approximation verwendet werden. Traditionell wird das Gleichungssystem explizit aufgestellt, d.h. Steifigkeitsmatrix $\boldsymbol{A}$ und Lastvektor $\boldsymbol{b}$ werden berechnet und gespeichert. Seien $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots \boldsymbol{r}_N$ die Knoten der Diskretisierung und $\varphi_1, \varphi_2, \ldots, \varphi_N$ die zugeordneten nodalen Basisfunktionen, siehe Definitionsgleichung (2.12). Folgender Algorithmus ist der Kern eines jeden Finite-Element Programms:

Algorithmus:	*assemble*
Eingabe:	vollständige Information über • die Finite-Element-Diskretisierung • das Randwertproblem
Ausgabe:	Steifigkeitsmatrix $\boldsymbol{A}$ und Lastvektor $\boldsymbol{b}$

Initialisiere $\boldsymbol{A} = \boldsymbol{0}$, $\boldsymbol{b} = \boldsymbol{0}$
Für alle Elemente $K^{(i)}$
  Für alle Knoten $\boldsymbol{r}_j, \boldsymbol{r}_k \in K^{(i)}$
    $A_{jk} := A_{jk} + \mathcal{A}_{K^{(i)}}(\varphi_k|_{K^{(i)}}, \varphi_j|_{K^{(i)}})$
  Für alle Knoten $\boldsymbol{r}_j \in K^{(i)}$
    $b_j := b_j + l_{K^{(i)}}(\varphi_j|_{K^{(i)}})$

Dabei bedeutet $\mathcal{A}_{K^{(i)}}$ und $l_{K^{(i)}}$, dass die Integrale nur über dem Element $K^{(i)}$ ausgewertet werden. Die Auswertung der Integrale erfolgt in der Regel mit Hilfe numerischer Quadraturverfahren, etwa mit der *Gauß-Quadratur* [20, 32]. Die Ordnung des Quadraturverfahrens ist dabei dem maximalen Polynomgrad p im Ansatzraum anzupassen. Der damit verbundene Aufwand steigt mit dem Polynomgrad p rasch an. So ist die Anzahl der Quadraturpunkte für den eindimensionalen Fall von der Ordnung $O(p)$, für den Fall beliebiger Dimension bei Verwendung von Tensorproduktformeln von der Ordnung $O(p^d)$. Damit zu multiplizieren sind die Kosten für die einmalige Auswertung

der Polynome φ_k und φ_i, der Koeffizienten und der Transformationsfunktionen $\psi^{(i)}$ an den Quadraturknoten, wodurch ein Faktor $O(p)$ hinzukommt.

Die Matrix $\boldsymbol{A}$ und der Vektor $\boldsymbol{b}$ werden bei obigem Algorithmus *elementweise* aufgebaut: In $\boldsymbol{A}$ werden die Teilmatrizen, die von den einzelnen Elementen stammen, sukzessive aufaddiert. Diese Teilmatrizen werden in diesem Zusammenhang auch *Elementmatrizen* genannt. Der Vorgang des schrittweisen Zusammensetzens der Gesamtmatrix wird als *Assemblierung* bezeichnet.

Die Elementmatrizen sind im Allgemeinen voll besetzt. Die Anzahl der Spalten und Reihen einer Elementmatrix stimmt mit der Anzahl der Freiheitsgrade des Elements überein. Elementmatrizen zweier benachbarter Elemente überlappen sich in der Gesamtmatrix $\boldsymbol{A}$ dort, wo sie sich auf gemeinsame Knoten der Elemente beziehen.

Aus diesem elementweisen Aufbau folgt, dass die Steifigkeitsmatrix dünn besetzt ist, wenn die Anzahl der Elemente groß ist gegenüber der Anzahl der Knoten je Element. Dies ist bei der h-Version der FEM der Fall. Die Anzahl der Nichtnulleinträge in $\boldsymbol{A}$ ist dort $O(N)$. Im Gegensatz hierzu steht die p-Version, bei der die Systemmatrix entsprechend der Anzahl der Elemente aus wenigen großen, vollbesetzten Blöcken besteht. Hier steigt die Anzahl der nichtverschwindenden Einträge wie N^2. Dieser Umstand ist bei einem Effizienzvergleich von h- und p-Methode von Bedeutung: FE-Programme, die auf die p-Version setzen, verwenden nahezu die gesamte Rechenzeit auf die Assemblierung der Steifigkeitsmatrix.

Neben den strukturellen Eigenschaften der Systemmatrix interessieren im Hinblick auf die Wahl des Gleichungslösers auch die algebraischen Eigenschaften: So folgt aus der Symmetrie und der Elliptizität der Bilinearform $\mathcal{A}(\cdot,\cdot)$ sofort, dass die Steifigkeitsmatrix symmetrisch und positiv definit ist.

Zur Lösung des linearen Gleichungssystems kommen wegen der Größe der Systeme und ihrer tendenziell dünnen Besetzungsmuster in der Regel *iterative* Verfahren zum Einsatz. Hier sind die Verfahren von *Jacobi* und *Gauß/Seidel* sowie das *SOR-Verfahren* zu nennen, im Falle von Symmetrie und Positiv-Definitheit das *konjugierte Gradienten-Verfahren (cg-Verfahren)* [30]. Iterative Verfahren neigen zu einem umso schlechteren Konvergenzverhalten, je größer die Konditionszahl der Systemmatrix ist. Diese steigt jedoch in der Regel mit einer feiner werdenden Zerlegung rasch an, typisch ist etwa ein Ansteigen der Ordnung $O(h^{-2})$. Ähnlich verhält es sich, wenn der Polynomgrad erhöht wird. Damit müsste für eine höhere Genauigkeit eine unverhältnismäßig längere Rechenzeit für die Lösung des Gleichungssystems investiert werden. Abhilfe schaffen geeignete Vorkonditionierer. Auf die Idee, das Problem auf verschiedenen, hierarchisch geordneten Ebenen zu betrachten, bauen die *Multilevel-Verfahren* [29, 49] auf. Multilevel- oder Mehrgitterverfahren zeigen im Idealfall von der Problemgröße unabhängige Konvergenzraten, d.h. der Zeitaufwand für die Lösung des Gleichungssystems steigt linear in der Zahl der Freiheitsgrade.

2.3 Auf dem Weg zu Hierarchischen Tensorproduktelementen

Bei einem Finite-Element-Verfahren entscheiden im Wesentlichen zwei Punkte über den Erfolg. Zuerst sind die durch den Ansatzraum gegebenen Approximationseigenschaften von Bedeutung. Hier wird die erreichbare Genauigkeit der Anzahl der dafür benötigten Freiheitsgrade gegenübergestellt, die ein direktes Maß für den Speicheraufwand ist. Der zweite Punkt betrifft das lineare Gleichungssystem, das sich sodann aus der Galerkin-Diskretisierung eines Randwertproblems ergibt. Auch hier ist der Aufwand zu betrachten, der sich aus dem Speicher- und dem Rechenaufwand für das Aufstellen und Lösen des Gleichungssystems zusammensetzt. Ziel muss es sein, einen Gleichungslöser zu finden, der mit einem in der Zahl der Unbekannten linearen Speicher- und Rechenaufwand zurecht kommt.

An dieser Stelle seien einige Beispiele gegeben, die die Optimierung des Ansatzraums betreffen und die gewisse Parallelen zu den im nächsten Kapitel vorgestellten hierarchischen Tensorproduktelementen aufweisen.

Zunächst werfen wir einen Blick auf die Elemente der sogenannten *Serendipity*-Klasse. Diese gehen aus den bekannten Tensorproduktelementen durch eine Reduktion von Freiheitsgraden hervor, die nur wenig zur Approximation beitragen. Bekanntestes Beispiel ist das quadratische 8-Knoten-Element, das das biquadratische Element beerbt (vergleiche Abbildung 2.2), aber ohne den Mittelpunktsknoten auskommt. Zunächst ist dieser Defekt von der technischen Seite zu betrachten: Die Entfernung des Knotens erfordert für eine sinnvolle Interpolation eine Neudefinition der Formfunktionen. Abbildung 2.3 zeigt das wesentliche Konstruktionsprinzip. Die Formfunktionen zu Knoten, die auf dem Mittelpunkt der Seiten liegen, stimmen entlang der Seite mit der quadratischen Basisfunktion überein, fallen zur gegenüberliegenden Seite aber linear ab. Die Formfunktionen werden aus den Formfunktionen des bilinearen Elements gewonnen, indem die eben definierten Randfunktionen mit dem Gewicht $-1/2$ addiert werden. Dadurch wird erreicht, dass die so definierte Eckformfunktion an der Ecke den Wert 1 besitzt, an den anderen sieben Knoten den Wert 0. Die Konstruktion des kubischen Serendipity-Elements, das aus dem bikubischen Element erneut durch Weglassen der Inneren Knoten entsteht, verläuft analog. Ab dem quartischen Element müssen allerdings neben den Randknoten auch innere Freiheitsgrade mitgenommen werden.

Um dies zu verstehen, sei kurz dargestellt, wie sich der Ansatzraum gegenüber den Tensorproduktelementen verändert. Dies wird am besten in der Standardbasis $\{x^i y^j\}$ der bivariaten Polynome klar. Während die Tensorproduktelemente hier alle Monome $1 \leq i, j \leq q$ umfassen, sparen die Serendipity-Elemente bestimmte davon aus, genau genommen solche, für die die Summe $i + j$ größer als ein festgesetzter Wert ist. Dies steht gerade im Einklang damit, dass Monome $x^i y^j$ bei der Interpolation einer bivariaten Funktionen einen umso höheren Beitrag liefern, je größer $i + j$ ist. Das heißt, es ist sinnvoll nur solche $x^i y^j$ in den Ansatzraum aufzunehmen, für die $i + j \leq q$ für ein festes $q \in \mathbb{N}$ ist. Da Randformfunktionen nach obiger Konstruktion

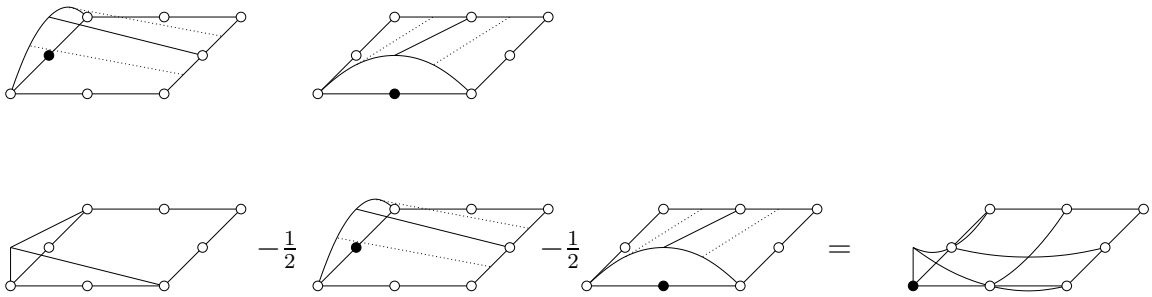


Abbildung 2.3: Zur Konstruktion der Formfunktionen des quadratischen Serendipity-Elements

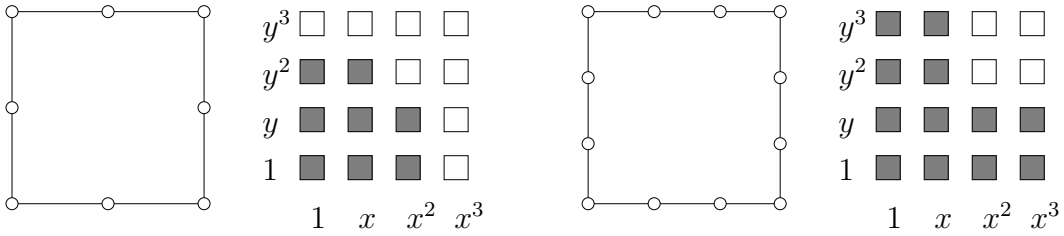


Abbildung 2.4: Quadratisches (links) und kubisches (rechts) Serendipity-Element und der zugeordnete Polynomraum bezüglich der Basis $\{x^i y^j\}$.

Polynome sind, die in der Variable senkrecht zum Rand linear sind, ist dieser Vorsatz für höhere Grade nicht zu halten, ohne dass noch zusätzliche, an inneren Knoten definierte Formfunktionen mit aufgenommen werden. Bei genauerer Betrachtung zeigt sich, dass auch bei den vorgestellten Serendipity-Elementen dieses Prinzip nicht exakt eingehalten wird, vergleiche Abbildung 2.4. So kann mit dem kubischen Element das Monom x^2y^2 nicht dargestellt werden. Dies ist zwar kein unumgängliches Problem – es würde reichen den Mittelknoten hinzuzunehmen und mit einer Formfunktion, die das genannte Monom enthält, zu versehen –, allerdings müssten die Basisfunktionen auf den Rändern mühsam korrigiert werden, damit sie auch am Mittelpunkt den Wert 0 besitzen.

Viel einfacher gestaltet sich die Konstruktion effizienter Elemente, wenn das Prinzip der nodalen Basis aufgegeben wird, also nicht weiter verlangt wird, dass die Koeffizienten in der Darstellung einer Funktion mit den Funktionswerten an den Knoten übereinstimmen. Folgender Vorschlag stammt von Szabó und Babuška [48]. Sie definieren über dem Intervall $K^{(1)} := [-1, 1]$ die eindimensionalen Formfunktionen

$$\begin{aligned}
\phi_1(x) &= \frac{1}{2}(1-x), \\
\phi_2(x) &= \frac{1}{2}(1+x), \\
\phi_i(x) &= \sqrt{\frac{2i-3}{2}} \int_{-1}^{x} L_{i-2}(t)dt \;=\; \sqrt{\frac{1}{4i-6}}(L_{i-1}(t) - L_{i-3}(t)), \quad i = 3, 4, \ldots .
\end{aligned}$$

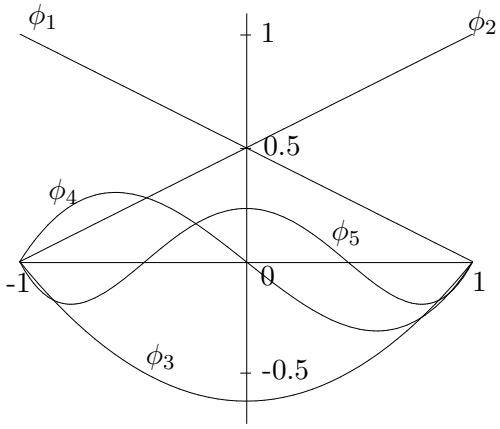


Abbildung 2.5: Hierarchische Basisfunktionen für die polynomialen Elemente nach Szabó und Babuška.

Dabei sind die L_n die bekannten *Legendre Polynome*, definiert durch

$$L_n(t) \;=\; \frac{1}{2^n n!}\frac{d^n}{dt^n}(t^2-1)^n, \quad t \in (-1,1), \quad n = 0,1,2,\ldots.$$

Die Funktion ϕ_1 ist die Formfunktion zum Randpunkt $x = -1$, ϕ_2 zum Randpunkt $x = 1$. Die Funktionen $\phi_i,\ i \geq 3$ lassen sich keinem Punkt sinnvoll zuordnen. Sie erfüllen allerdings $\phi_i(-1) = \phi_i(1) = 0$ und werden deshalb als innere Formfunktionen bezeichnet. ϕ_1 und ϕ_2 sind lineare Polynome, ϕ_i, $i \geq 3$, ist vom Grad $i-1$. In Abbildung 2.5 sind die ersten ϕ_i dargestellt.

Die Formfunktionen für höherdimensionale Elemente über $K^{(d)} = [-1,1]^d$ ergeben sich durch Tensorproduktbildung, vergleiche (2.14–2.15). Dort wird dann zwischen Randformfunktionen und inneren Formfunktionen unterschieden. Erstere zeichnen sich dadurch aus, dass wenigstens einer der Faktoren im Tensorprodukt mit ϕ_1 oder ϕ_2 übereinstimmt. Die inneren Formfunktionen sind nur aus solchen ϕ_i mit $i \geq 3$ zusammengesetzt.

Für konkrete Elemente ist nun eine Auswahl von Basisfunktionen zu treffen, die den lokalen Ansatzraum aufspannen. Die Definition

$$P^{(d,p)} \;:=\; \operatorname{span}\{\phi_{i_1} \times \cdots \times \phi_{i_d}, 1 \leq i_1, \ldots, i_d \leq p+1\}$$

führt auf ein Element, das mit dem Lagrange'schen Tensorproduktelement der Ordnung p gleichwertig ist (vergleiche Seite 10). Anders als beim Serendipity-Element gestaltet sich die Konstruktion reduzierter Ansatzräume hier wesentlich einfacher. Durch die Wahl

$$P^{(d,p)}_{\text{trunc}} \;:=\; \operatorname{span}\{\phi_{i_1} \times \cdots \times \phi_{i_d}, \sum_{j=1}^{d} \max(i_j - 1, 1) \leq p + d - 1\}$$

wird die Idee von einem Ansatzraum verwirklicht, der nur Polynome $x_1^{i_1} \cdots x_d^{i_d}$ mit einem Gesamtgrad $i_1 + \cdots + i_d \leq p + d - 1$ enthält. Der Raum $P^{(d,p)}_{\text{trunc}}$ wird deshalb auch

trunc space genannt, was soviel wie „gestutzter“ Raum bedeutet. Der Kniff mit dem Ausdruck „$\max(i_j - 1, 1)$“ bezweckt lediglich, dass mit jeder Randformfunktion auch ihre Entsprechung auf dem gegenüberliegenden Rand vorhanden ist. Die Stetigkeit auf den Grenzen zwischen zwei Elementen wird dadurch erreicht, dass die Koeffizienten der sich jeweils entsprechenden Randformfunktionen identisch sind.

Szabó und Babuška bezeichnen die ϕ_i als *hierarchische Basisfunktionen*, um damit auszudrücken, dass die Basis des Elements der Ordnung $p+1$ aus der Basis des Elements der Ordnung p hervorgeht, indem eine neue Basisfunktionen hinzugefügt wird, und nicht wie bei der nodalen Basis vollständig neu gebildet werden muss.

Auf Grund der Orthogonalität der Legendrefunktionen bezüglich des L_2-Skalarprodukts,

$$\int_{-1}^{1} L_n(t)L_m(t)dt \;=\; \begin{cases} \dfrac{2}{2n+1}, & \text{falls } n = m \\ 0, & \text{sonst,} \end{cases} \tag{2.20}$$

hat man

$$\int_{-1}^{1} \frac{d\phi_i}{dt}\frac{d\phi_j}{dt}dt \;=\; \delta_{ij}, \quad i,j \geq 3.$$

Die Elementmatrix für die Poissongleichung im Eindimensionalen ist bis auf die die Randformfunktionen betreffenden Einträge die Identität. Die 1D-Massenmatrix besitzt nur auf der Diagonalen und zwei Nebendiagonalen von null verschiedene Einträge. Da die Steifigkeitsmatrix im mehrdimensionalen Fall die Summe von Tensorprodukten aus eindimensionalen Steifigkeits- und Massenmatrizen ist, ist sie ebenfalls dünn besetzt, was sie gegenüber den in der Regel vollbesetzten Elementmatrizen für die nodale Basis gravierend unterscheidet. Dieser Vorteil der hierarchischen Basis ist aber sofort zunichte gemacht, wenn statt des Laplaceoperators ein Differentialoperator mit variablen Koeffizienten ins Spiel kommt. Hier ist im Allgemeinen wieder mit einer vollbesetzten Elementmatrix zu rechnen.[1]

Ein Vorteil der hierarchischen Basis besteht darin, dass die Konditionszahl der Steifigkeitsmatrix mit der Problemgröße essentiell langsamer steigt, als dies bei der nodalen Basis der Fall ist. Bei Yserentant [54, 55] ist nachzulesen, wie sich durch den Einsatz einer hierarchischen Basis für lineare Dreieckselemente mit dem cg-Verfahren optimale Konvergenzraten erzielen lassen, wie sie sonst für Mehrgitterverfahren gelten.

Werfen wir noch kurz einen Blick auf die Zahl der Freiheitsgrade, die man einspart, wenn statt des echten Produktraums der trunc-space verwendet wird. Der Produktraum besitzt eine Komplexität von $O(n^d)$, wenn n die Anzahl der Basisfunktionen des eindimensionalen Elements ist. Für den trunc-space vergewissert man sich leicht, dass die Komplexität hier von der Ordnung $O(n^d/d)$ und damit um den Faktor d

[1] Für Dünngitterelemente ergibt sich ein ähnliches Problem bei variablen Koeffizienten. In Abschnitt 4.2.3 wird eine modifizierte Bilinearform vorgestellt, für die die Multiplikation mit der Steifigkeitsmatrix auf Standardalgorithmen für den Laplace-Operator und Interpolationen zurückgeführt werden kann. Dies könnte auch für die p-Elemente nach Szabó und Babuška von Nutzen sein.

kleiner ist. Für die Elementmatrix bedeutet dies eine Reduktion der Einträge um den Faktor d^2, was sich für $d = 3$ sehr wohl auszahlt.

Der nächste Abschnitt wird zeigen, dass die Idee von einem „abgeschnittenen“ Tensorproduktraum zur Reduzierung der Freiheitsgrade bei gleich bleibender Approximationsgüte hier bei weitem noch nicht ausgereizt ist.

3 Hierarchische Tensorproduktelemente

Dieses Kapitel beschreibt die unter dem Namen *Dünne Gitter* bekannte Diskretisierung von $H^1(K^{(d)})$, $K^{(d)} = [0,1]^d$. Zunächst wird in Abschnitt 3.1 eine Zerlegung des Sobolevraums in eine direkte Summe aus endlich dimensionalen, hierarchisch geordneten Teilräumen vorgenommen. Dieses Prinzip wird im zweiten Abschnitt benutzt, um Finite Elemente über dem Referenzelement $K^{(d)}$ zu konstruieren, bei denen die erzielte Approximationsgenauigkeit im Verhältnis zur Anzahl der Knoten optimiert ist. Diese Idee wird in den nachfolgenden Abschnitten weitergeführt: So werden polynomiale Elemente höherer Ordnung und lokal adaptive Elemente vorgestellt. Die Darstellung folgt im Wesentlichen der von Bungartz [17], neu ist die Berücksichtigung inhomogener Randdaten, wie sie für die Einbettung der Dünngitter-Diskretisierung in ein Finite-Element-Verfahren benötigt werden.

3.1 Hierarchische Teilraumzerlegung für das Referenzelement

Dieser Abschnitt beschreibt die Zerlegung des Sobolevraums $H^1(K^{(d)})$ über dem Referenzelement $K^{(d)} = [0,1]^d$ in eine direkte Summe von endlich dimensionalen, hierarchisch geordneten Teilräumen.

Betrachten wir zunächst den Fall $d = 1$. Die Mengen

$$G_\ell \;=\; \{x_{\ell,i} = i \cdot 2^{-\ell},\; i = 0, 1, \ldots, 2^\ell\}$$

bilden für $l \in \mathbb{N}_0$ eine Folge ineinander eingebetteter, äquidistanter Gitter der Maschenweite $h_\ell = 2^{-\ell}$ über dem Intervall $K^{(1)} = [0,1]$. Mit Hilfe der über $\mathbb{R}$ gegebenen Funktion ϕ,

$$\phi(x) \;=\; \begin{cases} 1 - |x|, & \text{falls } x \in [-1,1], \\ 0, & \text{sonst,} \end{cases}$$

definieren wir die Funktionen $\phi_{\ell,i} : [0,1] \to \mathbb{R}$ $(\ell \in \mathbb{N}_0,\; 0 \le i \le 2^\ell)$ durch

$$\phi_{\ell,i}(x) \;=\; \phi\left(\frac{x - x_{\ell,i}}{h_\ell}\right).$$

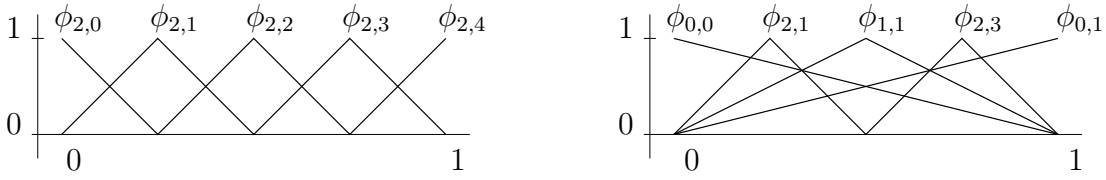


Abbildung 3.1: nodale (links) und hierarchische (rechts) Basis von V_2

Die $\phi_{\ell,i}$ sind die bekannten *Hutfunktionen* und bilden eine *Knotenbasis* oder *nodale* Basis für den Raum V_ℓ der über G_ℓ stückweise linearen Funktionen,

$$V_\ell \;=\; \operatorname{span}\{\phi_{\ell,i},\; 0 \le i \le 2^\ell\}\,. \tag{3.1}$$

Auf der linken Seite von Abbildung 3.1 ist die nodale Basis von V_2 dargestellt.

Wie definieren nun die Indexmenge $\mathfrak{I}_\ell$ ($\ell \in \mathbb{N}_0$) durch

$$\begin{aligned} \mathfrak{I}_0 &:= \{0,1\}, \\ \mathfrak{I}_\ell &:= \{i \in \mathbb{N},\; 0 < i < 2^\ell,\; i \text{ ungerade}\}, \quad i \ge 1 \end{aligned} \tag{3.2}$$

und die Räume

$$W_\ell := \operatorname{span}\{\phi_{\ell,i},\; i \in \mathfrak{I}_\ell\}, \qquad \ell \in \mathbb{N}_0.$$

Es ist

$$\begin{aligned} V_0 &= W_0, \\ V_\ell &= V_{\ell-1} \oplus W_\ell, \qquad \ell \ge 1, \end{aligned} \tag{3.3}$$

und damit

$$V_\ell \;=\; \bigoplus_{0 \le k \le \ell} W_k.$$

Ferner ist

$$B_\ell \;=\; \{\phi_{k,i},\; i \in \mathfrak{I}_k,\; 0 \le k \le \ell\}$$

eine Basis von V_ℓ, die sogenannte *hierarchische Basis*. Die Basen B_ℓ bilden im Gegensatz zur Knotenbasis eine aufsteigende Folge von Mengen, das heißt, beim Wechsel von V_ℓ nach $V_{\ell+1}$ wird die Basis nicht komplett ausgetauscht, es werden lediglich neue Basisfunktionen hinzugefügt. In Abbildung 3.1, rechte Seite, ist die hierarchische Basis von V_2 dargestellt.

Das eben für $d = 1$ Dargelegte lässt sich durch einen Tensorproduktansatz auf beliebige Dimension erweitern. Hierzu verwenden wir die Multiindizes $\boldsymbol{\ell}, \boldsymbol{i} \in \mathbb{N}_0^d$ und definieren die multivariaten Funktionen

$$\phi_{\boldsymbol{\ell},\boldsymbol{i}} \;=\; \phi_{\ell_1,i_1} \otimes \cdots \otimes \phi_{\ell_d,i_d}\;.$$

Das heißt, für $\boldsymbol{x} \in K^{(d)}$ ist

$$\phi_{\boldsymbol{\ell},\boldsymbol{i}}(\boldsymbol{x}) \;=\; \prod_{j=1}^{d} \phi_{\ell_j,i_j}(x_j).$$

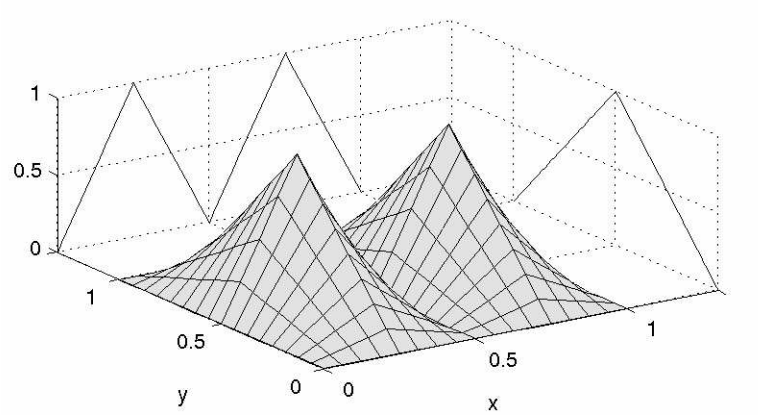


Abbildung 3.2: Die stückweise bilinearen Basisfunktionen $\phi_{(2,1),(1,1)}$ und $\phi_{(2,1),(3,1)}$ als Tensorprodukt von univariaten Basisfunktionen.

Abbildung 3.2 zeigt ein einfaches Beispiel für $d = 2$.

Im Zusammenhang mit Multiindizes vereinbaren wir folgende Schreibweisen:

$$\begin{aligned} 2^{\boldsymbol{\ell}} &= (2^{l_1}, \ldots, 2^{l_d}), \\ \mathbf{0} &= (0, \ldots, 0), \\ \mathbf{1} &= (1, \ldots, 1). \end{aligned}$$

Ferner verwenden wir die Normen

$$\begin{aligned} |\boldsymbol{\ell}|_1 &= \sum_{j=1}^{d} |l_j|, \\ |\boldsymbol{\ell}|_\infty &= \max_{1 \le j \le d} |l_j|, \end{aligned}$$

sowie die Ordnungsrelation

$$\boldsymbol{\ell} \le \boldsymbol{k} \quad :\Longleftrightarrow \quad \ell_j \le k_j \text{ für alle } 1 \le j \le d.$$

Analog zu (3.1) definieren wir nun für $\boldsymbol{\ell} \in \mathbb{N}_0^d$

$$V_{\boldsymbol{\ell}} = \operatorname{span}\{\phi_{\boldsymbol{\ell},\boldsymbol{i}}, \mathbf{0} \le \boldsymbol{i} \le 2^{\boldsymbol{\ell}}\}.$$

Bei den Funktionen $u \in V_{\boldsymbol{\ell}}$ handelt es sich um *stückweise multilineare* Funktionen über dem Rechtecksgitter $G_{\boldsymbol{\ell}} = G_{\ell_1} \times \cdots \times G_{\ell_d}$ mit Maschenweite $h_j = 2^{-l_j}$ in Richtung j. Die $\phi_{\boldsymbol{\ell},\boldsymbol{i}}$ bilden eine nodale Basis von $V_{\boldsymbol{\ell}}$.

Mit Hilfe der Indexmengen

$$\mathcal{I}_{\boldsymbol{\ell}} = \mathcal{I}_{\ell_1} \times \cdots \times \mathcal{I}_{\ell_d}$$

gewinnt man entsprechend der Darstellung für $d = 1$ die hierarchischen Inkremente

$$W_{\boldsymbol{\ell}} = \operatorname{span}\{\phi_{\boldsymbol{\ell},\boldsymbol{i}}, \boldsymbol{i} \in \mathcal{I}_{\boldsymbol{\ell}}\},$$

und damit die Teilraumzerlegung

$$V_{\boldsymbol{\ell}} \;=\; \bigoplus_{\mathbf{0}\le \boldsymbol{k}\le \boldsymbol{\ell}} W_{\boldsymbol{k}}. \tag{3.4}$$

Die hierarchische Basis $B_{\boldsymbol{\ell}}$ von $V_{\boldsymbol{\ell}}$ ist gegeben durch

$$B_{\boldsymbol{\ell}} \;=\; \{\phi_{\boldsymbol{k},\boldsymbol{i}}, \boldsymbol{i} \in \mathfrak{I}_{\boldsymbol{k}}, \mathbf{0} \le \boldsymbol{k} \le \boldsymbol{\ell}\}.$$

Von Bedeutung für die Galerkin-Diskretisierung ist nun, dass die stückweise multilinearen Funktionen dicht in $H^1(K^{(d)})$ liegen:

$$H^1(K^{(d)}) \;=\; \overline{\bigoplus_{\boldsymbol{\ell}\in\mathbb{N}_0^d} W_{\boldsymbol{\ell}}}.$$

Damit haben wir eine *hierarchische Teilraumzerlegung* von $H^1(K^{(d)})$.

3.2 Interpolation mit hierarchischen Basen

Im Folgenden betrachten wir die Interpolation mit hierarchischen Basen. Zunächst zum Fall $d = 1$. Die Standardbasis für die Interpolation ist die Knotenbasis, die Koeffizienten stimmen mit den Funktionswerten an den Gitterpunkten überein:

$$I_\ell(u) \;=\; \sum_{i=0}^{2^\ell} c_{\ell,i} \cdot \phi_{\ell,i}, \qquad c_{\ell,i} \;=\; u(x_{\ell,i}).$$

Bei der Interpolation in der hierarchischen Basis hat man

$$I_\ell(u) \;=\; \sum_{k=0}^{\ell} \sum_{i\in\mathfrak{I}_k} v_{k,i} \cdot \phi_{k,i}$$

wobei nun

$$v_{\ell,i} \;=\; \delta_{\ell,i}(u) \;:=\; \begin{cases} u(x_{\ell,i}), & \text{falls } \ell = 0, \\ u(x_{\ell,i}) - \frac{1}{2}\big(u(x_{\ell,i} - h_\ell) + u(x_{\ell,i} + h_\ell)\big), & \text{falls } \ell \ge 1. \end{cases} \tag{3.5}$$

Für $\ell \ge 1$ ist

$$v_{\ell,i} \;=\; u(x_{\ell,i}) - I_{\ell-1}(u)(x_{\ell,i}),$$

das heißt, $I_\ell(u)$ korrigiert den Interpolanten $I_{\ell-1}$ an den Stellen $x_{\ell,i}$, $i \in \mathfrak{I}_\ell$. Deshalb heißt $v_{\ell,i}$ auch *hierarchischer Überschuss*. Siehe hierzu Abbildung 3.3.

Ist u zweimal stetig differenzierbar, hat man

$$v_{\ell,i} \;\approx\; -\frac{1}{2} h_\ell^2 \cdot u''(x_{\ell,i}) \tag{3.6}$$

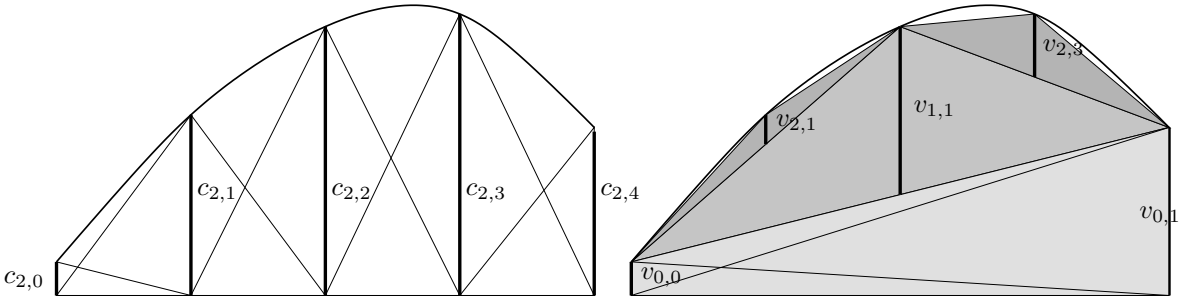


Abbildung 3.3: Interpolation mit nodaler (links) und hierarchischer Basis (rechts)

bzw. die Abschätzung

$$|v_{\ell,i}| \;\leq\; C \cdot 2^{-2\ell-1} \cdot \|u''\|_\infty. \tag{3.7}$$

mit einer von der Maschenweite unabhängigen Konstante $C > 0$. Die Überschüsse fallen damit wie $4^{-\ell}$.

Die multivariate Interpolation in $V_{\boldsymbol{\ell}}$ ist ganz analog zum eindimensionalen Fall gegeben durch

$$I_{\boldsymbol{\ell}}(u) \;=\; \sum_{\mathbf{0}\leq\boldsymbol{k}\leq\boldsymbol{\ell}} \sum_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{k}}} v_{\boldsymbol{k},\boldsymbol{i}} \cdot \phi_{\boldsymbol{k},\boldsymbol{i}}.$$

Die hierarchischen Überschüsse sind nun gegeben durch

$$v_{\boldsymbol{\ell},\boldsymbol{i}} \;=\; \delta_{\boldsymbol{\ell},\boldsymbol{i}}(u) \;:=\; \left(\bigotimes_{j=1}^{d} \delta_{\ell_j,i_j}\right)(u). \tag{3.8}$$

Die Überschüsse kann man wiederum mit Hilfe von Ableitungen von u abschätzen. Für die Überschüsse an den inneren Knoten $\boldsymbol{x}_{\boldsymbol{\ell},\boldsymbol{i}}$, $\boldsymbol{\ell} \geq \mathbf{1}$, hat man in Analogie zu (3.7)

$$|v_{\boldsymbol{\ell},\boldsymbol{i}}| \;\leq\; 2^{-d} \cdot 2^{-2\cdot|\boldsymbol{\ell}|_1} \cdot \left\| \frac{\partial^{2d} u}{\partial x_1^2 \cdots \partial x_d^2} \right\|_\infty. \tag{3.9}$$

Bei der Abschätzung von Überschüssen, die Randpunkten $\boldsymbol{x}_{\boldsymbol{\ell},\boldsymbol{i}}$, $\boldsymbol{\ell} \in \mathbb{N}_0^d \setminus \mathbb{N}^d$, zugeordnet sind, muss man beachten, dass diese nach (3.5) und (3.8) auch nur von der Einschränkung von u auf den Rand abhängen. So hat man für $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_{d'}, 0, \ldots, 0) \in \mathbb{N}^{d'} \times \mathbb{N}_0^{d-d'}$

$$|v_{\boldsymbol{\ell},\boldsymbol{i}}| \;\leq\; 2^{-d'} \cdot 2^{-2\cdot|\boldsymbol{\ell}|_1} \cdot \left\| \frac{\partial^{2d'} u}{\partial x_1^2 \cdots \partial x_d'^2} \right\|_\infty, \tag{3.10}$$

wobei das Supremum der Ableitung nur über dem Rand

$$]0,1[^{d'} \times \{x_{\ell_{d'+1},i_{d'+1}}\} \times \cdots \times \{x_{\ell_d,i_d}\}$$

zu suchen ist (siehe hierzu auch Abbildung 3.4). Insgesamt beobachtet man somit einen Abfall der Überschüsse im Innern und auf dem Rand von der Art

$$|v_{\boldsymbol{\ell},\boldsymbol{i}}| \;=\; O(4^{-|\boldsymbol{\ell}|_1}). \tag{3.11}$$

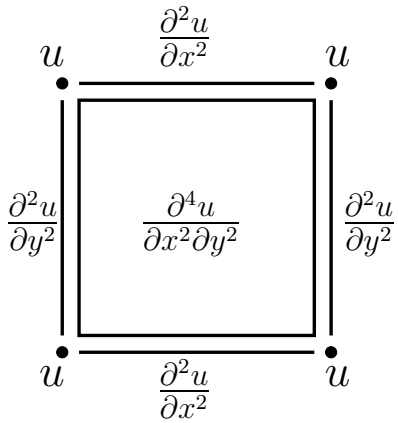


Abbildung 3.4: Die Abschätzung des hierarchischen Überschusses bei der Interpolation von u erfolgt über gemischte Ableitungen von u. Je nach Lage der Überschüsse im Innern, am Rand oder an den Ecken von $K^{(d)}$ (hier $d = 2$) sind andere Ableitungen zu betrachten.

Geeignete Funktionen $u : K^{(d)} \to \mathbb{R}$ für die Interpolation mit hierarchischen Tensorproduktbasen besitzen beschränkte gemischte Ableitungen

$$D^{\boldsymbol{\alpha}} u \;=\; \frac{\partial^{|\alpha|_1} u}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}.$$

Wir definieren dazu die Räume $X^r(K^{(d)})$, $r \in \mathbb{N}_0$, durch

$$X^r(\Omega) \;=\; \{u : \Omega \to \mathbb{R} : D^{\boldsymbol{\alpha}} u \in C^0(\Omega), |\boldsymbol{\alpha}|_\infty \le r\}, \tag{3.12}$$

sowie die Teilräume

$$X_0^r(\Omega) \;=\; \{u \in X^r(\Omega), u|_{\partial\Omega} = 0\}.$$

Sie werden mit den Seminormen

$$\begin{aligned} |u|_{\boldsymbol{\alpha},\infty} &= \|D^{\boldsymbol{\alpha}} u\|_{L_\infty}, \\ |u|_{\boldsymbol{\alpha},2} &= \|D^{\boldsymbol{\alpha}} u\|_{L_2} \end{aligned}$$

versehen ($\boldsymbol{\alpha} \in \mathbb{N}_0^d$, $|\boldsymbol{\alpha}| \le r$). Für die multilineare Interpolation ist wegen (3.9) $X^2(K^{(d)})$ von Interesse.

3.3 Optimale Teilräume und Dünne Gitter

Der für die multilineare Interpolation betrachtete Raum $V_{\boldsymbol{\ell}}$ ist die direkte Summe aller hierarchischen Inkremente $W_{\boldsymbol{k}}$ mit $\mathbf{0} \le \boldsymbol{k} \le \boldsymbol{\ell}$. Bei genauerer Betrachtung ergibt sich, dass sämtliche Räume der Art

$$V_{\mathcal{L}} \;=\; \bigoplus_{\boldsymbol{\ell} \in \mathcal{L}} W_{\boldsymbol{\ell}}, \tag{3.13}$$

mit $\mathcal{L} \subset \mathbb{N}_0^d$ zur Interpolation herangezogen werden können, solange die Bedingung

$$\boldsymbol{\ell} \in \mathcal{L} \quad \Rightarrow \quad \boldsymbol{k} \in \mathcal{L} \text{ für alle } \boldsymbol{k} \le \boldsymbol{\ell} \tag{3.14}$$

erfüllt ist. Sie garantiert, dass in $V_{\mathcal{L}}$ die Abfolge der hierarchischen Inkremente $W_{\boldsymbol{\ell}}$ von den groben zu den feinen Leveln keine Unterbrechung erfährt und damit die Korrekturwirkung der hierarchischen Basis, die ja nur von einem Level zum unmittelbar darauffolgenden Level gegeben ist, gewährleistet ist. Insbesondere interpoliert

$$I_{\mathcal{L}}(u) \;=\; \sum_{\boldsymbol{\ell}\in\mathcal{L}} \sum_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{\ell}}} v_{\boldsymbol{\ell},\boldsymbol{i}} \cdot \phi_{\boldsymbol{\ell},\boldsymbol{i}} \tag{3.15}$$

die Funktion u und stimmt mit ihr an den Punkten von

$$G_{\mathcal{L}} \;:=\; \bigcup_{\boldsymbol{\ell}\in\mathcal{L}} G_{\boldsymbol{\ell}} \tag{3.16}$$

überein.

Der Gestaltungsfreiraum in der Zusammensetzung von $V_{\mathcal{L}}$ wirft nun die Frage auf, welche $\mathcal{L}$ besonders effiziente Ansatzräume $V_{\mathcal{L}}$ liefern. Für die Beurteilung der Effizienz ist die bei der Interpolation in $V_{\mathcal{L}}$ zu erreichende Genauigkeit ins Verhältnis zu dem dazu notwendigen Speicheraufwand zu setzen. Ein Maß für den Speicheraufwand ist die Anzahl $N_{\mathcal{L}}$ der Freiheitsgrade in $V_{\mathcal{L}}$,

$$N_{\mathcal{L}} \;=\; \dim V_{\mathcal{L}} \;=\; \sum_{\boldsymbol{\ell}\in\mathcal{L}} \dim W_{\boldsymbol{\ell}} \;=\; \sum_{\boldsymbol{\ell}\in\mathcal{L}} 2^{|\boldsymbol{\ell}-\mathbf{1}|_1}. \tag{3.17}$$

Der Interpolationsfehler wird wie folgt abgeschätzt:

$$\begin{aligned} \|u - I_{\mathcal{L}}(u)\| \;&=\; \left\| \sum_{\boldsymbol{\ell}\in\mathbb{N}_0^d} \sum_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{\ell}}} v_{\boldsymbol{\ell},\boldsymbol{i}} \cdot \phi_{\boldsymbol{\ell},\boldsymbol{i}} \;-\; \sum_{\boldsymbol{\ell}\in\mathcal{L}} \sum_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{\ell}}} v_{\boldsymbol{\ell},\boldsymbol{i}} \cdot \phi_{\boldsymbol{\ell},\boldsymbol{i}} \right\| \\ &\leq\; \sum_{\boldsymbol{\ell}\in\mathbb{N}_0^d\setminus\mathcal{L}} \left\| \sum_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{\ell}}} v_{\boldsymbol{\ell},\boldsymbol{i}} \cdot \phi_{\boldsymbol{\ell},\boldsymbol{i}} \right\| \end{aligned} \tag{3.18}$$

Die Abschätzungen (3.17)–(3.18) können dazu benutzt werden, um für ein gegebenes $\mathcal{L} \in \mathbb{N}_0^d$ Aussagen über den Interpolationsfehler zu gewinnen, sie taugen jedoch kaum dazu, effiziente Räume $V_{\mathcal{L}}$ zu konstruieren. Eine Konstruktion von $V_{\mathcal{L}}$ muss darin bestehen, ausgehend von $W_{\mathbf{0}}$ unter Beachtung von (3.14) sukzessiv weitere Inkremente $W_{\boldsymbol{\ell}}$ hinzuzunehmen. Bungartz [17] schlägt vor, alle $\boldsymbol{\ell} \in \mathbb{N}_0^d$ aufzunehmen, für die der Quotient

$$\operatorname{cbr}(\boldsymbol{\ell}) \;:=\; \frac{\left\| \sum_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{\ell}}} v_{\boldsymbol{\ell},\boldsymbol{i}} \cdot \phi_{\boldsymbol{\ell},\boldsymbol{i}} \right\|^2}{\dim W_{\boldsymbol{\ell}}} \tag{3.19}$$

größer oder gleich einem beliebigen, aber festen $\sigma > 0$ ist. Er begründet sein Vorgehen mit einem Optimierungsansatz, auf den hier nicht näher eingegangen werden soll.

Der Ausdruck $\operatorname{cbr}(\boldsymbol{\ell})$ gibt im Wesentlichen das lokale *Kosten-Nutzen-Verhältnis* für den Teilraum $W_{\boldsymbol{\ell}}$ wieder, das den von $W_{\boldsymbol{\ell}}$ stammenden Beitrag zum Interpolanten (vergleiche mit (3.18)) ins Verhältnis zu der aufzubringenden Anzahl von Freiheitsgraden setzt.

Homogene Randwerte

Betrachten wir zunächst nur Funktionen $u \in X_0^2(K^{(d)})$, also solche Funktionen, die beschränkte (gemischte) Ableitungen bis zur Ordnung 2 haben und auf dem Rand $\partial\Omega$ verschwinden. Man hat dann $v_{\boldsymbol{\ell},\boldsymbol{i}} = 0$ für $\boldsymbol{\ell} \in \mathbb{N}_0^d \setminus \mathbb{N}^d$. Es genügt somit, $\boldsymbol{\ell} \geq \mathbf{1}$ zu betrachten. (3.9) eingesetzt in (3.19) ergibt

$$\mathrm{cbr}(\boldsymbol{\ell}) \;\approx\; \frac{2^{-2d} \cdot 2^{-4\cdot|l|_1} \cdot \|D^{\mathbf{2}}u\|_\infty}{2^{|\boldsymbol{\ell}|_1} \cdot 2^{-d}} \;=\; 2^{-d} \cdot 2^{-5\cdot|l|_1} \cdot \|D^{\mathbf{2}}u\|_\infty$$

Als Beispiel für eine Ansatzraumoptimierung wird im Folgenden die Optimierung bezüglich der L_∞-Norm geschildert. Wegen $\|\phi_{\boldsymbol{\ell},\boldsymbol{i}}\|_{L_\infty} = 1$ ist

$$\mathrm{cbr}(\boldsymbol{\ell}) \;=\; \frac{\max_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{\ell}}} |v_{\boldsymbol{\ell},\boldsymbol{i}}|^2}{2^{|\boldsymbol{\ell}-\mathbf{1}|_1}}.$$

Legen wir das zu $\boldsymbol{\ell}_n = (n, 1, \ldots, 1)$, $n \in \mathbb{N}$, gehörende lokale Kosten-Nutzen-Verhältnis als Schwellwert σ_n fest, also

$$\sigma_n \;=\; \mathrm{cbr}(\boldsymbol{\ell}_n) \;=\; 2^{-d} \cdot 2^{-5\cdot(n+d-1)} \cdot \|D^{\mathbf{2}}u\|_\infty,$$

so ist nach obiger Darstellung

$$\mathcal{L}_{0,n}^{(1)} \;:=\; \{\boldsymbol{\ell} \in \mathbb{N}^d; |\boldsymbol{\ell}|_1 \leq n + d - 1\} \tag{3.20}$$

die in Bezug auf das lokale Kosten-Nutzen-Verhältnis optimierte Indexmenge. Der Index (1) bezieht sich dabei auf die Norm $|\boldsymbol{\ell}|_1$, die in der Definition von $\mathcal{L}_{0,n}^{(1)}$ die bestimmende Rolle übernimmt.

Bungartz [17] zeigt weiter, dass eine Optimierung bezüglich der L_2-Norm,

$$\|u\|_{L_2} \;=\; \left(\int_{K^{(d)}} u(\boldsymbol{x})^2 d\boldsymbol{x}\right)^{\frac{1}{2}},$$

ebenfalls auf die durch $\mathcal{L}_{0,n}^{(1)}$ gegebenen Räume $V_{0,n}^{(1)} := V_{\mathcal{L}_{0,n}^{(1)}}$ hinausläuft.

Die Optimierung bezüglich der Energienorm,

$$\|u\|_E \;=\; \left(\sum_{j=1}^{d} \left\|\frac{\partial u}{\partial x_j}\right\|_{L_2}^2\right)^{1/2},$$

ergibt als optimale Indexmenge

$$\mathcal{L}_{0,n}^{(E)} \;=\; \left\{\boldsymbol{\ell} \in \mathbb{N}^d; |\boldsymbol{\ell}|_1 - \frac{1}{5}\cdot\log_2\left(\sum_{j=1}^{d} 4^{l_j}\right) \leq (n+d-1) - \frac{1}{5}\cdot\log_2(4^n + 4d - 4)\right\}. \tag{3.21}$$

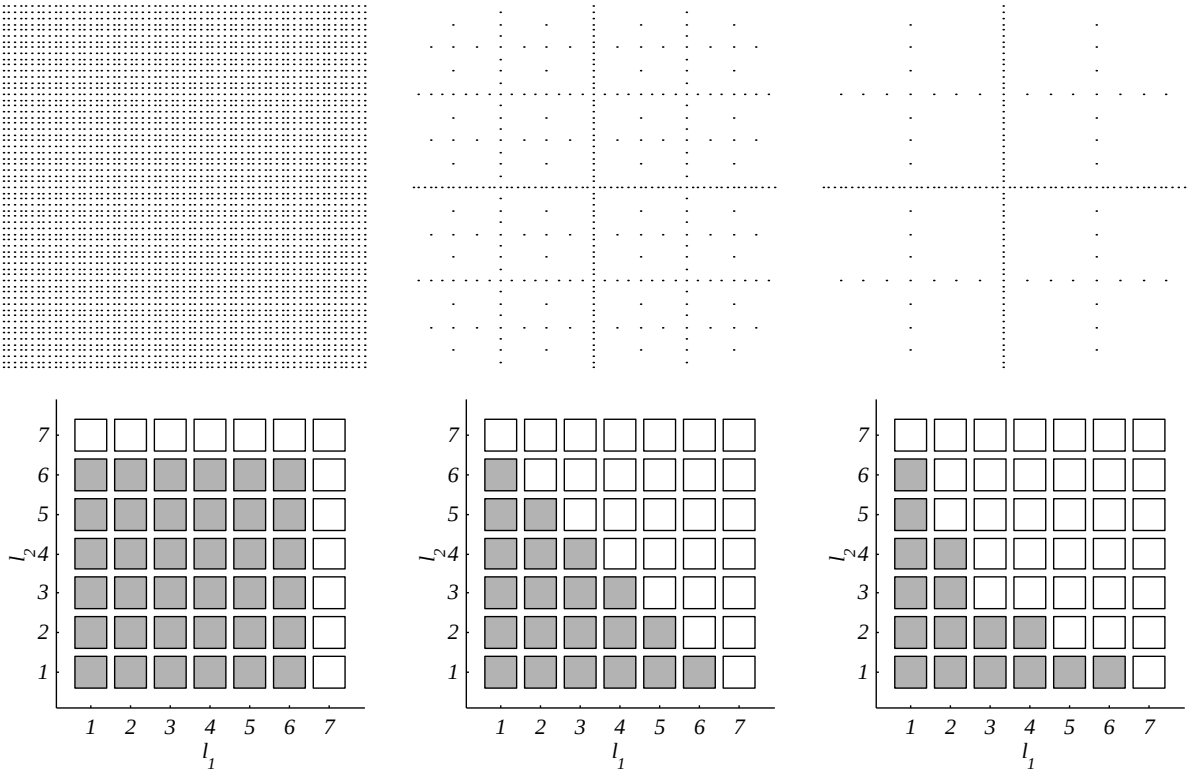


Abbildung 3.5: Oben: Das volle Gitter $G_{0,n}^{(\infty)}$, das L_2-basierte dünne Gitter $G_{0,n}^{(1)}$ und das energiebasierte dünne Gitter $G_{0,n}^{(E)}$ für $n = 6$ und $d = 2$. Unten: Die Indexmengen $\mathfrak{L}_{0,n}^{(\infty)}$, $\mathfrak{L}_{0,n}^{(1)}$ und $\mathfrak{L}_{0,n}^{(E)}$.

Vergleichen wir nun die durch $\mathfrak{L}_{0,n}^{(1)}$ und $\mathfrak{L}_{0,n}^{(E)}$ gegebenen Räume $V_{0,n}^{(1)}$ und $V_{0,n}^{(E)}$ mit dem Raum $V_{0,n}^{(\infty)}$, gegeben durch die Indexmenge

$$\mathfrak{L}_{0,n}^{(\infty)} := \{\boldsymbol{\ell} \in \mathbb{N}^d; |\boldsymbol{\ell}|_\infty \leq n\}. \tag{3.22}$$

$V_{0,n}^{(\infty)}$ ist abgesehen davon, dass er nur Funktionen enthält, die auf dem Rand verschwinden, identisch mit dem bereits bekannten Raum $V_{n\cdot\mathbf{1}}$ multilinearer Funktionen über dem äquidistanten Gitter der Maschenweite $h_n = 2^{-n}$. Die Gitter $G_{0,n}^{(1)}$ und $G_{0,n}^{(E)}$ der Räume $V_{0,n}^{(1)}$ und $V_{0,n}^{(E)}$ sind durch (3.16) gegeben. Beispiele für $d = 2$ sind in Abbildung 3.5 zu sehen. Der kleinste Abstand zweier Gitterpunkte in $G_{0,n}^{(1)}$ und $G_{0,n}^{(E)}$ beträgt wie im Gitter $G_{\infty,n}^{(E)}$ $h_n = 2^{-n}$; dieser wird im Folgenden als die Maschenweite dieser Gitter bezeichnet. Auf Grund ihrer Struktur werden die Gitter $G_{0,n}^{(1)}$ und $G_{0,n}^{(E)}$ als *dünne Gitter* bezeichnet. In diesem Zusammenhang werden die $G_{0,n}^{(\infty)}$ auch *volle Gitter* genannt. Mit Bezug auf die der Optimierung zu Grunde liegenden Norm heißen die $G_{0,n}^{(1)}$ *L_2-basierte dünne Gitter*, die $G_{0,n}^{(E)}$ entsprechend *energiebasierte dünne Gitter*.

Bei einem Vergleich von L_2- und energiebasierten dünnen Gittern fällt auf, dass $G_{0,n}^{(E)} \subset G_{0,n}^{(1)}$. Insbesondere werden bei den energiebasierten Gittern solche Teilräume

$W_{\boldsymbol{\ell}}$ zusätzlich weggelassen, deren Basisfunktionen einen Träger mit ausgeglichenem Verhältnis der Seitenlängen besitzen, also $\ell_1 \approx \cdots \approx \ell_d$. Positiv formuliert bedeutet das, dass Basisfunktionen mit langgestreckten Trägern bevorzugt in $V_{0,n}^{(E)}$ aufgenommen werden. Dies steht im Einklang mit obiger Konstruktionsvorschrift für optimierte Ansatzräume. Es ist nämlich

$$\|\phi_{\boldsymbol{\ell},\boldsymbol{i}}\|_E \;=\; \sqrt{2}\cdot\left(\frac{2}{3}\right)^{(d-1)/2}\cdot 2^{-|\boldsymbol{\ell}|_1/2}\cdot\left(\sum_{j=1}^{d} 4^{l_j}\right)^{1/2}.$$

Das heißt, unter Basisfunktionen mit $|\boldsymbol{\ell}|_1 = const$ liefern diejenigen einen größeren Beitrag zur „Energie“, für die einige der ℓ_j hoch, die anderen niedrig sind.

Die Anzahl der Gitterpunkte in $G_{0,n}^{(\infty/1/E)}$ ist durch (3.17) gegeben und besitzt für $n \to \infty$ das folgende asymptotische Verhalten:

$$\begin{array}{lcll} N_{0,n}^{(\infty)} & = & O(2^{d\cdot n}) & = O(h_n^{-d}), \\ N_{0,n}^{(1)} & = & O(2^n \cdot n^{d-1}) & = O(h_n^{-1}\cdot|\log_2 h_n|^{d-1}), \\ N_{0,n}^{(E)} & = & O(2^n) & = O(h_n^{-1}). \end{array} \tag{3.23}$$

Die Anzahl der Gitterpunkte nimmt bei den vollen Gittern mit jeder Verfeinerung $n \to n+1$ um den Faktor 2^d zu. Bei dreidimensionalen numerischen Simulationen, die einer hohen räumlichen Auflösung bedürfen, um überhaupt sinnvolle Ergebnisse zu liefern – die direkte numerische Simulation (DNS) turbulenter Strömungen ist ein Beispiel – ist dieser rasche Anstieg der Gitterpunktzahlen und der damit verbundene Speicheraufwand das Hauptproblem und letztendlich der begrenzende Faktor für die Berechenbarkeit bestimmter Probleme. Bei den dünnen Gittern verdoppelt sich die Zahl der Gitterpunkte im Wesentlichen mit jeder Verfeinerung. Der Faktor n^{d-1} bei den L_2-basierten Gittern fällt dabei kaum ins Gewicht. Siehe hierzu auch Tabelle 3.1.

Interessant ist nun, dass trotz dieser gravierenden Unterschiede in der Zahl der Gitterpunkte für große n der Interpolationsfehler auf vollen und dünnen Gittern nahezu das gleiche asymptotische Verhalten aufweist: Mit $I_{0,n}^{(\star)}$, $\star \in \{\infty, 1, E\}$, werde der Interpolationsoperator $X_0^2(K^{(d)}) \to V_{0,n}^{(\star)}$ bezeichnet. Für den Interpolationsfehler bezüglich der L_∞-Norm hat man

$$\begin{array}{lll} \|u - I_{0,n}^{(\infty)}(u)\|_{L_\infty} & \le c_1(d)\cdot 2^{-2n}\cdot|u|_{\mathbf{2},\infty} & = O(h_n^2), \\ \|u - I_{0,n}^{(1)}(u)\|_{L_\infty} & \le c_2(d)\cdot 2^{-2n}\cdot n^{d-1}\cdot|u|_{\mathbf{2},\infty}, & = O(h_n^2\cdot|\log_2 h_n|^{d-1}), \end{array} \tag{3.24}$$

für die L_2-Norm ist

$$\begin{array}{lll} \|u - I_{0,n}^{(\infty)}(u)\|_{L_2} & \le c_3(d)\cdot 2^{-2n}\cdot|u|_{\mathbf{2},2} & = O(h_n^2), \\ \|u - I_{0,n}^{(1)}(u)\|_{L_2} & \le c_4(d)\cdot 2^{-2n}\cdot n^{d-1}\cdot|u|_{\mathbf{2},2} & = O(h_n^2\cdot|\log_2 h_n|^{d-1}), \end{array} \tag{3.25}$$

und in der Energienorm ist

$$\begin{array}{lll} \|u - I_{0,n}^{(\infty)}(u)\|_E & \le c_5(d)\cdot 2^{-n}\cdot|u|_{\mathbf{2},\infty} & = O(h_n), \\ \|u - I_{0,n}^{(1)}(u)\|_E & \le c_6(d)\cdot 2^{-n}\cdot|u|_{\mathbf{2},\infty} & = O(h_n), \\ \|u - I_{0,n}^{(E)}(u)\|_E & \le c_7(d)\cdot 2^{-n}\cdot|u|_{\mathbf{2},\infty} & = O(h_n). \end{array} \tag{3.26}$$

n	$\dim V_{0,n}^{(\infty)}$	$\dim V_{0,n}^{(1)}$	$\dim V_{0,n}^{(E)}$
d=2			
5	961	129	81
6	3969	321	177
10	$1.05 \cdot 10^6$	9217	3841
11	$4.19 \cdot 10^6$	20481	7937
20	$1.10 \cdot 10^{12}$	$1.99 \cdot 10^7$	$4.72 \cdot 10^6$
21	$4.40 \cdot 10^{12}$	$4.19 \cdot 10^7$	$9.50 \cdot 10^6$
d=3			
5	29791	351	159
6	250047	1023	399
10	$1.07 \cdot 10^9$	47103	10495
11	$8.58 \cdot 10^9$	114687	22783
20	$1.15 \cdot 10^{18}$	$2.00 \cdot 10^8$	$1.68 \cdot 10^7$
21	$9.22 \cdot 10^{18}$	$4.42 \cdot 10^8$	$3.42 \cdot 10^7$
d=4			
5	923521	769	432
6	15752961	2561	1088
10	$1.10 \cdot 10^{12}$	178177	24321
11	$1.76 \cdot 10^{13}$	471041	56065
20	$1.21 \cdot 10^{24}$	$1.41 \cdot 10^9$	$5.27 \cdot 10^7$
21	$1.93 \cdot 10^{25}$	$3.27 \cdot 10^9$	$1.09 \cdot 10^8$

Tabelle 3.1: Dimension von $V_{0,n}^{(\infty)}$, $V_{0,n}^{(1)}$ und $V_{0,n}^{(E)}$ für verschiedene Werte von n und d.

Um die Gitter in Bezug auf Ihre Effizienz gegenüberzustellen, sei das asymptotische Verhalten der Interpolationsfehler in Abhängigkeit von der Anzahl N der Freiheitsgrade angegeben ($u \in X_0^2(K^{(d)})$):

$$\begin{aligned}
\|u - I_{0,n}^{(\infty)}(u)\|_{L_\infty} &= O(N^{-\frac{2}{d}}) \\
\|u - I_{0,n}^{(1)}(u)\|_{L_\infty} &= O(N^{-2} \cdot |\log_2 N|^{3\cdot(d-1)}) \\
\\
\|u - I_{0,n}^{(\infty)}(u)\|_{L_2} &= O(N^{-\frac{2}{d}}) \\
\|u - I_{0,n}^{(1)}(u)\|_{L_2} &= O(N^{-2} \cdot |\log_2 N|^{3\cdot(d-1)}) \qquad (3.27) \\
\\
\|u - I_{0,n}^{(\infty)}(u)\|_{E} &= O(N^{-\frac{1}{d}}) \\
\|u - I_{0,n}^{(1)}(u)\|_{E} &= O(N^{-1} \cdot |\log_2 N|^{d-1}) \\
\|u - I_{0,n}^{(E)}(u)\|_{E} &= O(N^{-1})
\end{aligned}$$

Hier sieht man nun deutlich, was sich oben bereits abgezeichnet hat: Die Approximationseigenschaften der dünnen Gitter sind denen voller Gitter essentiell überlegen. Der Vorsprung wächst mit steigender Raumdimension. Besondere Beachtung verdienen die energiebasierten Dünngitter. Unabhängig von der Dimension versprechen sie das gleiche Approximationsverhalten $O(N^{-1})$ in der Energienorm. Dies ist umso erfreulicher, als die Energienorm im Zusammenhang mit Randwertaufgaben die natürliche Norm zur Fehlerbemessung darstellt.

Inhomogene Randwerte

Die vorgestellten Ergebnisse beziehen sich bislang auf Funktionen u, die auf dem Rand verschwinden. Dies ist ausreichend, wenn ein Randwertproblem mit homogenen Dirichlet-Randbedingungen über $\Omega = K^{(d)}$ zu lösen ist. Sobald andere Randbedingungen gestellt sind oder das Gebiet mit mehreren Elementen zu diskretisieren ist, was in der Praxis der Regelfall ist, sind die Randpunkte und die zugehörigen Basisfunktionen in den Ansatzraum miteinzubeziehen. Betrachten wir deshalb im Folgenden $u \in X^2(K^{(d)})$.

Prinzipiell kann der Optimierungsansatz für die inneren Punkte ($\boldsymbol{\ell} \in \mathbb{N}^d$) auch auf alle Punkte ($\boldsymbol{\ell} \in \mathbb{N}_0^d$) angewandt werden. Die Abschätzung der Überschüsse am Rand hängt dabei nur von den gemischten zweiten Ableitungen auf diesem Rand ab. Anstelle einer globalen Optimierung, die mit dem Maximum all dieser gemischten Ableitungen arbeitet, empfiehlt sich eine getrennte Betrachtung von Innerem und Rändern.

Wir definieren

$$K_L = \{0\}\,, \quad K_M =]0,1[\,, \quad K_R = \{1\}$$

und haben mit

$$K^{(d)} \;=\; [0,1]^d \;=\; \bigcup_{\boldsymbol{\nu}\in\{L,M,R\}^d} K_{\nu_1} \times \cdots \times K_{\nu_d} \;=:\; \bigcup_{\boldsymbol{\nu}\in\{L,M,R\}^d} K_{\boldsymbol{\nu}} \tag{3.28}$$

eine disjunkte Zerlegung von $K^{(d)}$ in Inneres und Ränder. Die $K_{\boldsymbol{\nu}}$ sind dabei $d_{\boldsymbol{\nu}}$-dimensionale Untermannigfaltigkeiten des $\mathbb{R}^d$, $d_{\boldsymbol{\nu}} = \#\{\nu_j = M\}$. In Abbildung 3.4 auf Seite 26 ist die Zerlegung für $d = 2$ bereits dargestellt worden.

Für ein u, gegeben durch

$$u \;=\; \sum_{\boldsymbol{\ell}\in\mathbb{N}_0^d,\ \boldsymbol{i}\in\mathbb{I}_{\boldsymbol{\ell}}} v_{\boldsymbol{\ell},\boldsymbol{i}} \cdot \phi_{\boldsymbol{\ell},\boldsymbol{i}},$$

definieren wir die Projektionen

$$u_{\boldsymbol{\nu}} \;=\; \sum_{\boldsymbol{x}_{\boldsymbol{\ell},\boldsymbol{i}}\in K_{\boldsymbol{\nu}}} v_{\boldsymbol{\ell},\boldsymbol{i}} \cdot \phi_{\boldsymbol{\ell},\boldsymbol{i}}, \qquad \boldsymbol{\nu} \in \{L,M,R\}^d\,.$$

Nach Konstruktion verschwindet $u_{\boldsymbol{\nu}}$ auf dem Rand von $K_{\boldsymbol{\nu}}$. Damit haben wir die folgende Darstellung von u:

$$u \;=\; \sum_{\boldsymbol{\nu}\in\{L,M,R\}^d} u_{\boldsymbol{\nu}}\,. \tag{3.29}$$

Für den in (3.15) definierten Interpolanten $I_{\mathcal{L}}(u)$, $\mathcal{L} \in \mathbb{N}_0^d$, gilt analog

$$I_{\mathcal{L}}(u) \;=\; I_{\mathcal{L}} \left(\sum_{\boldsymbol{\nu} \in \{L,M,R\}^d} u_{\boldsymbol{\nu}} \right) \;=\; \sum_{\boldsymbol{\nu} \in \{L,M,R\}^d} I_{\mathcal{L}}(u_{\boldsymbol{\nu}}) \; . \tag{3.30}$$

Ein Beispiel für die Zerlegung (3.29) einer bivariaten Funktion ist in Abbildung 3.6 zu sehen.

Für den Fehler bekommt man nun mit (3.29) und (3.30) die Abschätzung

$$\| u - I_{\mathcal{L}}(u) \| \;\leq \sum_{\boldsymbol{\nu} \in \{L,M,R\}^d} \| u_{\boldsymbol{\nu}} - I_{\mathcal{L}}(u_{\boldsymbol{\nu}}) \| \, . \tag{3.31}$$

Der Summand für $\boldsymbol{\nu} = (M, \ldots, M)$ fällt in die Kategorie homogener Randdaten und kann nach (3.27) abgeschätzt werden. Für die restlichen Summanden gilt: $u_{\boldsymbol{\nu}}$ bzw. $I_{\mathcal{L}}(u_{\boldsymbol{\nu}})$ ist Tensorprodukt aus der Funktion $u_{\boldsymbol{\nu}}|_{\overline{K_{\boldsymbol{\nu}}}}$ bzw. $I_{\mathcal{L}}(u_{\boldsymbol{\nu}})|_{\overline{K_{\boldsymbol{\nu}}}}$, die auf der Randhyperfläche $\overline{K_{\boldsymbol{\nu}}}$ lebt, und einer Level-0-Basisfunktion. Der nichttriviale Teil $u_{\boldsymbol{\nu}}|_{\overline{K_{\boldsymbol{\nu}}}}$ bzw. $I_{\mathcal{L}}(u_{\boldsymbol{\nu}})|_{\overline{K_{\boldsymbol{\nu}}}}$ verschwindet nach obiger Konstruktion auf dem Rand seines Definitionsgebiets $\overline{K_{\boldsymbol{\nu}}}$. Damit haben wir die Interpolation bei inhomogenen Randbedingungen auf die Interpolation mit homogenen Randbedingungen zurückgeführt und können obige Ergebnisse wiederverwenden, um folgenden Satz zu beweisen:

Satz 3.1 *Die Indexmengen $\mathcal{L}_n^{(1)}$ und $\mathcal{L}_n^{(E)}$ seien definiert durch*

$$\begin{aligned} \mathcal{L}_n^{(1)} &= \left\{ \boldsymbol{\ell} \in \mathbb{N}_0^d : \exists \boldsymbol{k} \in \mathcal{L}_{0,n}^{(1)} \text{ mit } \boldsymbol{\ell} \leq \boldsymbol{k} \right\}, \\ \mathcal{L}_n^{(E)} &= \left\{ \boldsymbol{\ell} \in \mathbb{N}_0^d : \exists \boldsymbol{k} \in \mathcal{L}_{0,n}^{(E)} \text{ mit } \boldsymbol{\ell} \leq \boldsymbol{k} \right\}. \end{aligned} \tag{3.32}$$

$V_n^{(1)}$ und $V_n^{(E)}$ seien die durch diese Indexmengen und (3.13) definierten Räume stückweise multilinearer Funktionen. Für die Interpolation einer Funktion $u \in X^2([0,1]^d)$ in $V_n^{(1)}$ bzw. $V_n^{(E)}$ gilt dann

$$\begin{aligned} \| u - I_n^{(1)}(u) \|_{L_\infty} &= O(2^{-2n} \cdot n^{d-1}) \\ \| u - I_n^{(1)}(u) \|_{L_2} &= O(2^{-2n} \cdot n^{d-1}) \\ \| u - I_n^{(1)}(u) \|_E &= O(2^{-n} \cdot n^{d-1}) \\ \| u - I_n^{(E)}(u) \|_E &= O(2^{-n}) \end{aligned} \tag{3.33}$$

Beweis: Zunächst wird gezeigt, dass die Einschränkung des durch $\mathcal{L}_n^{(1)}$ bzw. $\mathcal{L}_n^{(E)}$ definierten Gitters $G_n^{(1)}$ bzw. $G_n^{(E)}$ auf den Rand $K_{\boldsymbol{\nu}}$ zu einem niedriger dimensionalen Gitter $G_{0,n}^{(1)}$ kongruent ist bzw. ein niedriger dimensionales Gitter $G_{0,n-1}^{(E)}$ enthält, sofern n groß genug ist.

Zunächst zu L_2-basierten Gittern: Als Rand-Level betrachten wir o.B.d.A. den Level $\boldsymbol{\ell} = (l_1, \ldots, l_{d'}, 0, \ldots, 0) \in \mathbb{N}_0^d$. Er gehört nach Definition (3.32) genau dann zu $\mathcal{L}_n^{(1)}$, wenn $\hat{\boldsymbol{\ell}} := (l_1, \ldots, l_{d'}, 1, \ldots, 1)$ zu $\mathcal{L}_{0,n}^{(1)}$ gehört, also die Ungleichung

$$\sum_{j=1}^{d'} \ell_j + (d - d') \;\leq\; n + d - 1$$

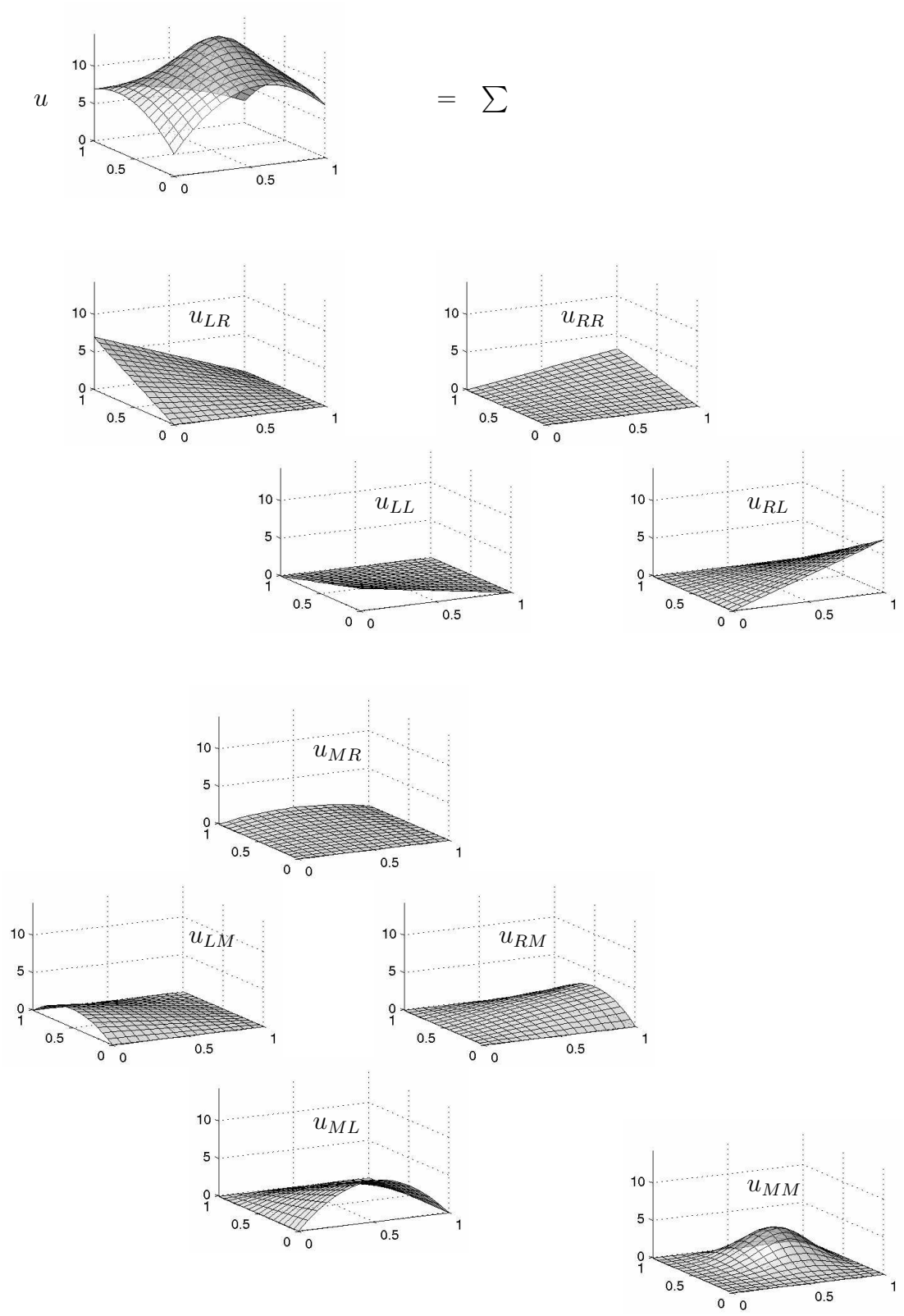


Abbildung 3.6: Zerlegung (3.29) einer Funktion u in ihre (Rand-) Projektionen $u_{\boldsymbol{\nu}}$.

erfüllt ist (vergleiche (3.20)). Für den d'-dimensionalen Level $\boldsymbol{\ell}' := (\ell_1, \ldots, \ell_{d'})$ ist dies gleichbedeutend mit

$$|\boldsymbol{\ell}'|_1 \leq n + d' - 1,$$

was aber gerade die Bestimmungsungleichung (3.20) für $\mathcal{L}_{0,n}^{(1)}$ im Fall von d' Raumdimensionen ist. Die Einschränkung des d-dimensionalen Gitters $G_n^{(1)}$ auf den d'-dimensionalen Rand ist also zu dem d'-dimensionalen Gitter $G_{0,n}^{(1)}$ kongruent.

Betrachten wir nun energiebasierte Gitter. Die Multiindizes $\boldsymbol{\ell}$, $\boldsymbol{\ell}'$ und $\hat{\boldsymbol{\ell}}$ seien wie oben definiert. Der Level $\boldsymbol{\ell}' \in \mathbb{N}^{d'}$ gehöre zu $\mathcal{L}_{0,n-1}^{(E)}$, d.h. nach Definition (3.21) ist

$$|\boldsymbol{\ell}'|_1 - \frac{1}{5}\log_2\left(\sum_{j=1}^{d'} 4^{\ell_j}\right) \;\leq\; (n-1+d'-1) - \frac{1}{5}\log_2(4^{n-1}+4d'-4)\;.$$

Einfache Umformungen ergeben

$$\begin{aligned}|\boldsymbol{\ell}'|_1 + (d-d') - \frac{1}{5}\log_2\left(\sum_{j=1}^{d'} 4^{\ell_j} + 4d - 4d'\right) + \frac{1}{5}\log_2\left(\frac{\sum_{j=1}^{d'} 4^{\ell_j} + 4d - 4d'}{\sum_{j=1}^{d'} 4^{\ell_j}}\right) \;\leq \\ \leq (n-1+d-1) - \frac{1}{5}\log_2(4^n+4d-4) + \frac{1}{5}\log_2\left(\frac{4^n+4d-4}{4^{n-1}+4d'-4}\right)\;. \qquad (3.34)\end{aligned}$$

Die Ungleichung bleibt erfüllt, wenn man den zweiten Logarithmusterm in der ersten Zeile streicht. Ferner ist

$$\frac{1}{5}\log_2\left(\frac{4^n+4d-4}{4^{n-1}+4d'-4}\right) \;\to\; \frac{2}{5} \qquad \text{für } n \to \infty \qquad (3.35)$$

Für n groß genug hat man also

$$|\boldsymbol{\ell}'|_1 + (d-d') - \frac{1}{5}\log_2\left(\sum_{j=1}^{d'} 4^{\ell_j} + 4d - 4d'\right) \;\leq (n-1+d-1) - \frac{1}{5}\log_2(4^n+4d-4) + 1.$$

Das ist die Bestimmungsungleichung für $\hat{\boldsymbol{\ell}} \in \mathcal{L}_n^{(E)}$. Die Einschränkung des d-dimensionalen Gitters $G_n^{(E)}$ auf den Rand enhält also das d'-dimensionale Gitter $G_{0,n-1}^{(E)}$, falls n groß genug ist.

Da nun gezeigt ist, dass die Schnitte $\mathcal{L}_n^{(1/E)} \cap K_{\boldsymbol{\nu}}$ zu niedriger dimensionalen Gittern $G_{\tilde{n}}^{(1/E)}$, $\tilde{n} = O(n)$, oder Obermengen davon kongruent sind und ferner nach Voraussetzung $u_{\boldsymbol{\nu}}|_{\overline{K_{\boldsymbol{\nu}}}} \in X_0^2([0,1]^{d'})$ gilt, lässt sich für die Projektion des Interpolationsfehlers eingeschränkt auf den Rand das folgende asymptotische Verhalten angeben (vergleiche (3.24)–(3.26)):

$$\begin{aligned}\|u_{\boldsymbol{\nu}}|_{\overline{K_{\boldsymbol{\nu}}}} - I_{\mathcal{L}}(u_{\boldsymbol{\nu}})|_{\overline{K_{\boldsymbol{\nu}}}}\|_{L_\infty} &= O(2^{-2n}\cdot n^{d-1}),\\ \|u_{\boldsymbol{\nu}}|_{\overline{K_{\boldsymbol{\nu}}}} - I_{\mathcal{L}}(u_{\boldsymbol{\nu}})|_{\overline{K_{\boldsymbol{\nu}}}}\|_{L_2} &= O(2^{-2n}\cdot n^{d-1}), \qquad (3.36)\\ \|u_{\boldsymbol{\nu}}|_{\overline{K_{\boldsymbol{\nu}}}} - I_{\mathcal{L}}(u_{\boldsymbol{\nu}})|_{\overline{K_{\boldsymbol{\nu}}}}\|_{E} &= O(2^{-n}).\end{aligned}$$

Die Normen beziehen sich auf die d'-dimensionalen Mannigfaltigkeiten $\overline{K_{\boldsymbol{\nu}}}$.

Nun ist

$$u_{\boldsymbol{\nu}} - I_{\mathcal{L}}(u_{\boldsymbol{\nu}}) \;=\; \left(u_{\boldsymbol{\nu}}|_{\overline{K_{\boldsymbol{\nu}}}} - I_{\mathcal{L}}(u_{\boldsymbol{\nu}})|_{\overline{K_{\boldsymbol{\nu}}}}\right) \times \phi' \;=:\; e_{\boldsymbol{\nu}} \times \phi', \tag{3.37}$$

wobei ϕ' ein $(d-d')$-dimensionales Produkt von Level-0-Basisfunktionen ist. Für das Tensorprodukt gilt

$$\begin{aligned} \|e_{\boldsymbol{\nu}} \times \phi'\|_\infty &\leq \|e_{\boldsymbol{\nu}}\|_\infty \cdot \|\phi'\|_\infty, && (3.38)\\ \|e_{\boldsymbol{\nu}} \times \phi'\|_{L_2} &= \|e_{\boldsymbol{\nu}}\|_{L_2} \cdot \|\phi'\|_{L_2}, && (3.39) \end{aligned}$$

und

$$\begin{aligned} \|e_{\boldsymbol{\nu}} \times \phi'\|_E^2 &= \sum_{i=1}^{d'} \|\partial_i e_{\boldsymbol{\nu}}\|_{L_2}^2 \cdot \|\phi'\|_{L_2}^2 \;+\; \sum_{i=d'+1}^{d} \|e_{\boldsymbol{\nu}}\|_{L_2}^2 \cdot \|\partial_i \phi'\|_{L_2}^2 && (3.40)\\ &\leq \|e_{\boldsymbol{\nu}}\|_E^2 \cdot \|\phi'\|_{L_2}^2 \;+\; C \cdot \|e_{\boldsymbol{\nu}}\|_E^2 \cdot \|\phi'\|_E^2 && (3.41) \end{aligned}$$

Dabei wurde die *Poincaré-Friedrichs-Ungleichung* $\|e_{\boldsymbol{\nu}}\|_{L_2} \leq C \cdot \|e_{\boldsymbol{\nu}}\|_E$ benutzt ($e_{\boldsymbol{\nu}}$ verschwindet auf $\partial K_{\boldsymbol{\nu}}$!). Die Konstante C hängt nicht von $e_{\boldsymbol{\nu}}$ ab. Da es sich bei den ϕ' ausschließlich um Level-0-Basisfunktionen handelt, sind die Normen $\|\phi'\|_\star$, $\star \in \{L_\infty, L_2, E\}$, durch eine Konstante nach oben beschränkt. Verfolgen wir mit diesem Wissen die Abschätzungen (3.38)–(3.40) über (3.37) und (3.36) zurück zu (3.31), erhalten wir die im Satz aufgestellten Behauptungen. □

Betrachten wir das Gitter $G_n^{(1/E)}$, das zu der in (3.32) definierten Indexmenge $\mathcal{L}_n^{(1/E)}$ gehört. Es stimmt im Innern von $K^{(d)}$ mit dem Gitter $G_{0,n}^{(1/E)}$ überein und besitzt zusätzlich Randpunkte, die sich aus der Projektion der inneren Punkte auf die Ränder ergeben. Diese Projektionen sind nach dem Beweis von Satz 3.1 in dünnen Gittern gleicher oder nächst größerer Tiefe, jedoch niedrigerer Dimension enthalten. In Abbildung 3.7 ist dies für ein dreidimensionales, L_2-basiertes dünnes Gitter veranschaulicht. Für die Anzahl der Gitterpunkte in $G_n^{(1/E)}$ hat man mit (3.23)

$$\begin{aligned} N_n^{(1)} &= O(2^n \cdot n^{d-1}) + O(2^n \cdot n^{d-2}) + \ldots \;=\; O(2^n \cdot n^{d-1})\,,\\ N_n^{(E)} &= O(2^n) + O(2^n) + \ldots \;=\; O(2^n)\,. \end{aligned} \tag{3.42}$$

Man beachte, dass die Anzahl der Summanden gemäß der Zerlegung (3.28) gleich 3^d ist, also nicht von n abhängt.

Das asymptotische Verhalten in n von sowohl Interpolationsfehler als auch Gitterpunktzahl stimmt für die Gitter $G_{0,n}^{(1/E)}$ und $G_n^{(1/E)}$ überein, vergleiche hierzu (3.24)–(3.26) mit (3.33) und (3.23) mit (3.42). Damit bekommen wir für das asymptotische Verhalten des Interpolationsfehlers in der Anzahl der Gitterpunkte N (vergleiche (3.27)):

$$\begin{aligned} \|u - I_n^{(1)}(u)\|_{L_\infty} &= O(N^{-2} \cdot |\log_2 N|^{3\cdot(d-1)})\\ \|u - I_n^{(1)}(u)\|_{L_2} &= O(N^{-2} \cdot |\log_2 N|^{3\cdot(d-1)})\\ \|u - I_n^{(1)}(u)\|_E &= O(N^{-1} \cdot |\log_2 N|^{d-1})\\ \|u - I_n^{(E)}(u)\|_E &= O(N^{-1}) \end{aligned} \tag{3.43}$$

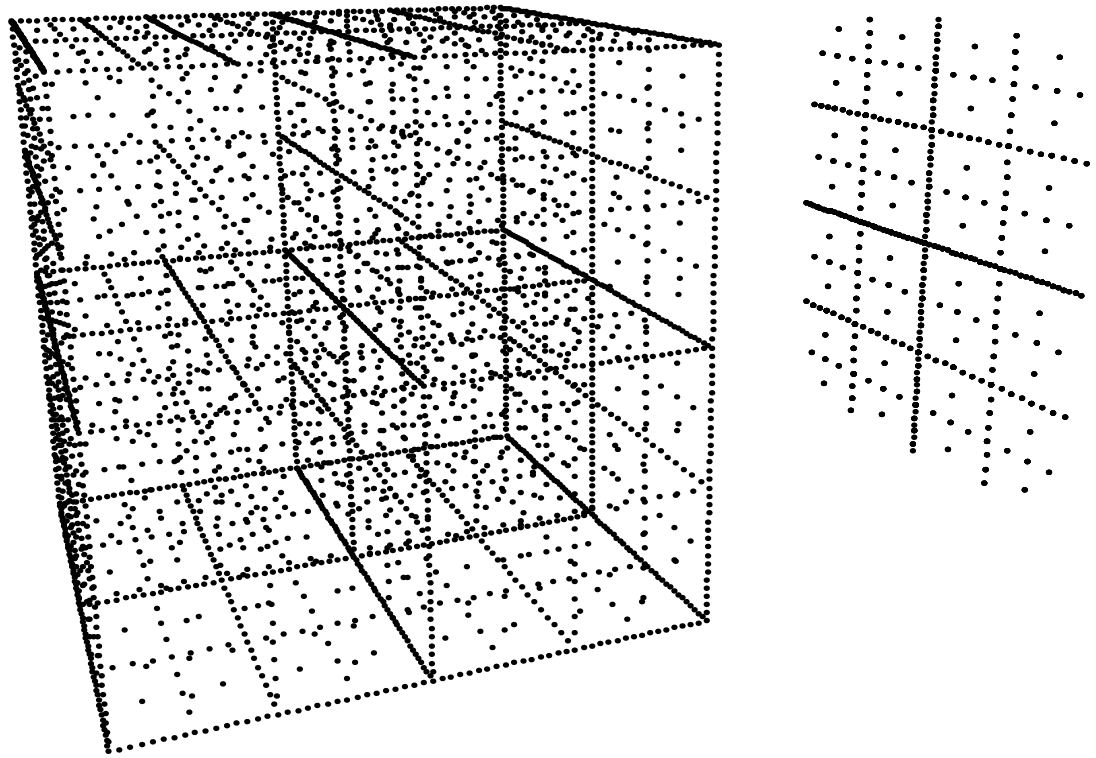


Abbildung 3.7: L_2-basiertes dünnes Gitter in 3D. Die Einschränkung auf die (rechte) Seitenfläche ist ein zweidimensionales, L_2-basiertes dünnes Gitter.

3.4 Polynomiale Elemente höherer Ordnung

Die Idee der hierarchischen Teilraumzerlegung und der hierarchischen Basis wurde in den vorangegangenen Abschnitten für den konkreten Fall der Approximation mit stückweise linearen Funktionen betrachtet. Sie lässt sich in gleicher Weise für die Approximation bzw. Interpolation mit Polynomen höheren Grades formulieren. Betrachten wir dazu das folgende allgemeine Konstruktionsschema.

Über den äquidistanten Gittern G_ℓ der Maschenweite $h_\ell = 2^{-\ell}$ sei eine Folge von Interpolationsformeln I_ℓ gegeben durch

$$I_\ell(u) \;=\; \sum_{i=0}^{2^\ell} u(x_{\ell,i}) \cdot \phi_{\ell,i}$$

mit

$$\phi_{\ell,i}(x_{\ell,i'}) \;=\; \begin{cases} 1, & \text{falls } i = i', \\ 0, & \text{sonst.} \end{cases} \tag{3.44}$$

Die Interpolationen I_ℓ sollen ferner die Bedingung

$$I_\ell I_{\ell-1} \;=\; I_{\ell-1} \quad \forall \ell \geq 1 \tag{3.45}$$

erfüllen. Die Räume

$$V_\ell \;=\; \operatorname{bild} I_\ell \;=\; \operatorname{span}\{\phi_{\ell,i}, 0 \le i \le 2^\ell\}$$

bilden damit eine aufsteigende Folge von diskreten Funktionenräumen.

Die weiteren Ausführungen zur hierarchischen Teilraumzerlegung verlaufen nun analog zu der Darstellung in Abschnitt 3.1. Insbesondere sind die hierarchischen Inkremente gegeben durch

$$W_\ell \;=\; \operatorname{bild}(I_\ell - I_{\ell-1}) \;=\; \operatorname{span}\{\phi_{\ell,i}, i \in \mathfrak{I}_\ell\}\ .$$

Für die Definition von $\mathfrak{I}_\ell$ siehe (3.2). Ferner hat man die hierarchische Basis

$$B_\ell \;=\; \{\phi_{k,i}, i \in \mathfrak{I}_k, 0 \le k \le \ell\}\ . \tag{3.46}$$

Die hierarchischen Überschüsse, also die Koeffizienten $v_{\ell,i}$ bezüglich der hierarchischen Basis, korrigieren den Fehler der Interpolation $I_{\ell-1}(u)$ an der Stelle $x_{\ell,i}$:

$$v_{\ell,i} \;=\; I_\ell(u)(x_{\ell,i}) - I_{\ell-1}(u)(x_{\ell,i}) \;=\; u(x_{\ell,i}) - I_{\ell-1}(u)(x_{\ell,i}).$$

Polynominterpolation nach Lagrange

Der Lagrange-Interpolant einer Funktion u zu einer Menge von n (paarweise verschiedenen) Knoten ist definiert als das Polynom vom Grad $\le n-1$, das an den Knoten mit u übereinstimmt. Der maximale Grad des interpolierenden Polynoms über G_ℓ ist damit 2^ℓ.

Die Basisfunktionen $\phi_{\ell,i}^{(\mathrm{Lagr})}$ sind gegeben durch

$$\phi_{\ell,i}^{(\mathrm{Lagr})}(x) \;=\; \prod_{\substack{0\le j\le 2^\ell \\ j\ne i}} \frac{x - x_{\ell,j}}{x_{\ell,i} - x_{\ell,j}}. \tag{3.47}$$

Die ersten nodalen bzw. hierarchischen Basisfunktionen, die sich aus (3.44) bzw. (3.46) ergeben, sind in Abbildung 3.8 zu sehen.

Zwei Punkte sind anzusprechen: Erstens ist die Wahl äquidistanter Knoten für die Polynominterpolation höheren Grades ungeeignet. Sie lassen die Interpolation bei steigendem Polynomgrad rasch instabil werden, was sich darin äußert, dass der Wertebereich der Basisfunktionen, der optimalerweise zwischen 0 und 1 liegt, unbegrenzt wächst [45, 23]. Diese Schwäche kann durch die Wahl von Tschebyscheff-Knoten

$$x_{\ell,i} \;=\; -\cos\left(i \cdot 2^{-l} \cdot \pi\right), \quad 0 \le i \le 2^\ell,$$

behoben werden [21]. Über die resultierenden Dünngitter-Interpolationsverfahren wird in [12, 36] berichtet.

Der zweite Punkt bleibt auch dann kritisch: Während der Träger der stückweise linearen Basisfunktionen $\phi_{\ell,i}$ auf das Intervall $[x_{\ell,i} - h_\ell, x_{\ell,i} + h_\ell]$ der Breite $2h_\ell$ beschränkt ist, hat man hier für alle Basisfunktionen

$$\operatorname{supp} \phi_{\ell,i}^{(\mathrm{Lagr})} \;=\; [0,1].$$

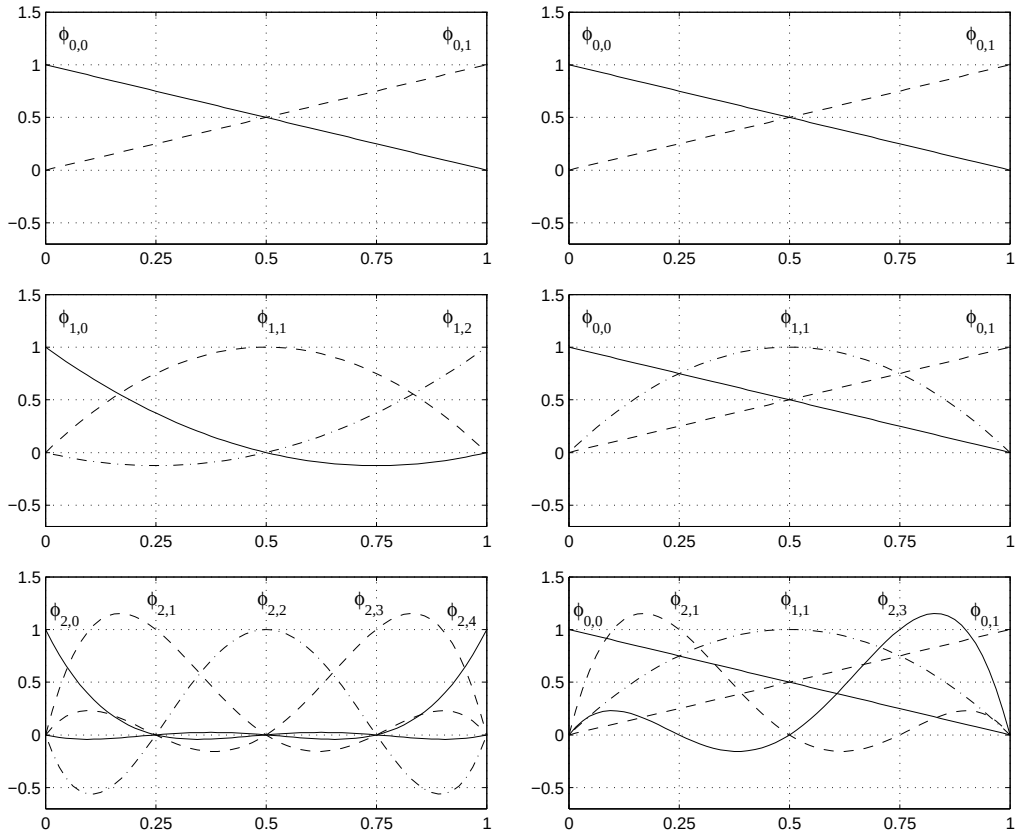


Abbildung 3.8: Nodale (links) und hierarchische (rechts) Basis für die Polynominterpolation nach Lagrange für $\ell = 1, 2, 3$

Vorausschauend auf die Galerkin-Diskretisierung würde dies eine volle Steifigkeitsmatrix $\boldsymbol{A}$ bedeuten, da nun für alle Paare (ϕ, ϕ') von Basisfunktionen der Matrixeintrag $\mathcal{A}(\phi, \phi')$ im Allgemeinen von Null verschieden ist. Für dünne Gitter kann die Multiplikation $\boldsymbol{A} \cdot \boldsymbol{u}$ mit Hilfe von unidirektionalen Algorithmen (Abschnitt 3.6.1) effizient realisiert werden. Voraussetzung hierfür sind ebenfalls lokale Träger, d.h. $\operatorname{supp} \phi_{\ell,i} \subset [x_{\ell,i} - h_\ell, x_{\ell,i} + h_\ell]$.

Hierarchische Polynominterpolation

Anstatt alle Punkte, die zu einem Level $\boldsymbol{\ell}$ gehören, als Knoten zu benutzen und den Interpolanten global, also über dem ganzen Intervall $[0, 1]$ zu definieren, schlägt Bungartz [17] vor, den Interpolanten lokal über den Intervallen

$$J_{\ell,i} := [x_{\ell,i} - h_\ell, x_{\ell,i} + h_\ell], \quad i \in \mathfrak{I}_\ell,$$

zu bestimmen und dabei jeweils den Punkt $x_{\ell,i}$ sowie bestimmte Punkte, die bereits in Gittern der Level $\ell - 1, \ell - 2, \ldots$ vorkommen, als Knoten zu benutzen. Wir wollen dieses Vorgehen im Folgenden präzisieren.

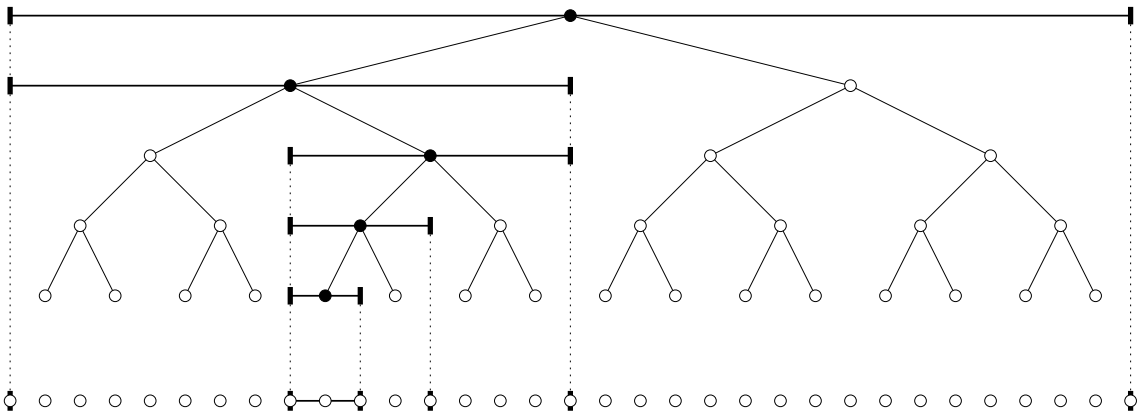


Abbildung 3.9: Anordnung der Punkte von G_ℓ, $\ell = 5$, in einem Binärbaum. Der Punkt $x_{5,9}$ und seine hierarchischen Vorfahren sind durch ausgefüllte Kreise gekennzeichnet. Für sie sind ferner die Intervalle $J_{\ell,i}$ eingezeichnet. Mit ♀ sind die Knoten für die hierarchische Interpolation markiert.

Zunächst werden die inneren Punkte von G_ℓ, wie in Abbildung 3.9 gezeigt, in einem Baum angeordnet. Die k-te Ebene des Baums, $1 \le k \le \ell$, besteht dabei aus allen Punkten $x \in G_k \setminus G_{k-1} = \{x_{k,i}, i \in \mathcal{I}_k\}$. Die Blätter des Baums sind die Punkte $x_{\ell,i}$, $i \in \mathcal{I}_\ell$. Die Kanten des Baums beschreiben lokale Hierarchien. So werden jedem Punkt $x_{\ell,i}$, $i \in \mathcal{I}_\ell$, durch

$$\begin{aligned} \mathcal{S}_1(\ell,i) &:= (\ell+1, 2i-1), \\ \mathcal{S}_2(\ell,i) &:= (\ell+1, 2i+1) \end{aligned} \tag{3.48}$$

die *Söhne* $x_{\mathcal{S}_1(\ell,i)}$ und $x_{\mathcal{S}_2(\ell,i)}$ zugeordnet. Umgekehrt lässt sich jedem Punkt $x_{\ell,i}$, $\ell > 1$, ein *Vaterknoten* $x_{\mathcal{V}(l,i)}$ zuordnen. Die Punkte

$$x_{\mathcal{V}(\ell,i)},\ x_{\mathcal{V}(\mathcal{V}(\ell,i))}, \ldots, x_{\mathcal{V}^{\ell-1}(\ell,i)}$$

heißen dann *hierarchische Vorfahren* von $x_{\ell,i}$.

Für $\ell \in \mathbb{N}$, $i \in \mathcal{I}_\ell$ und $0 \le j \le \ell - 1$ definieren wir

$$K_{\ell,i}^{(j)} := \{x_{\ell,i}\} \cup \partial J_{\ell,i} \cup \partial J_{\mathcal{V}(\ell,i)} \cup \cdots \cup \partial J_{\mathcal{V}^j(\ell,i)},$$

wobei $\partial J_{\ell,i}$ aus den beiden Randpunkten des Intervalls $J_{\ell,i}$ besteht. Aus Abbildung 3.9 leicht ersichtlich ist

$$|K_{\ell,i}^{(j)}| = 3 + j.$$

Der *hierarchische Lagrange-Interpolant* $I_\ell^{(p)}(u)$ über dem Gitter G_ℓ wird nun stückweise über den Blatt-Intervallen $J_{\ell,i}$, $i \in \mathcal{I}_\ell$, definiert: Er ist dort gegeben durch das Polynom vom Grad $\le p$, das an den Punkten in $K_{\ell,i}^{(p-2)}$ mit u übereinstimmt. Zu beachten ist, dass sinnvolle Werte für den Grad p zwischen 2 und $\ell + 1$ liegen. Das heißt umgekehrt, dass die Polynominterpolation mit Grad p erst ab Level $\ell = p - 1$ möglich ist. Insbesondere ist auf Level 0 mit der linearen Interpolation aus den Vorgängerabschnitten zu starten, die in diesem Sinne die „Lücke“ $p = 1$ schließt.

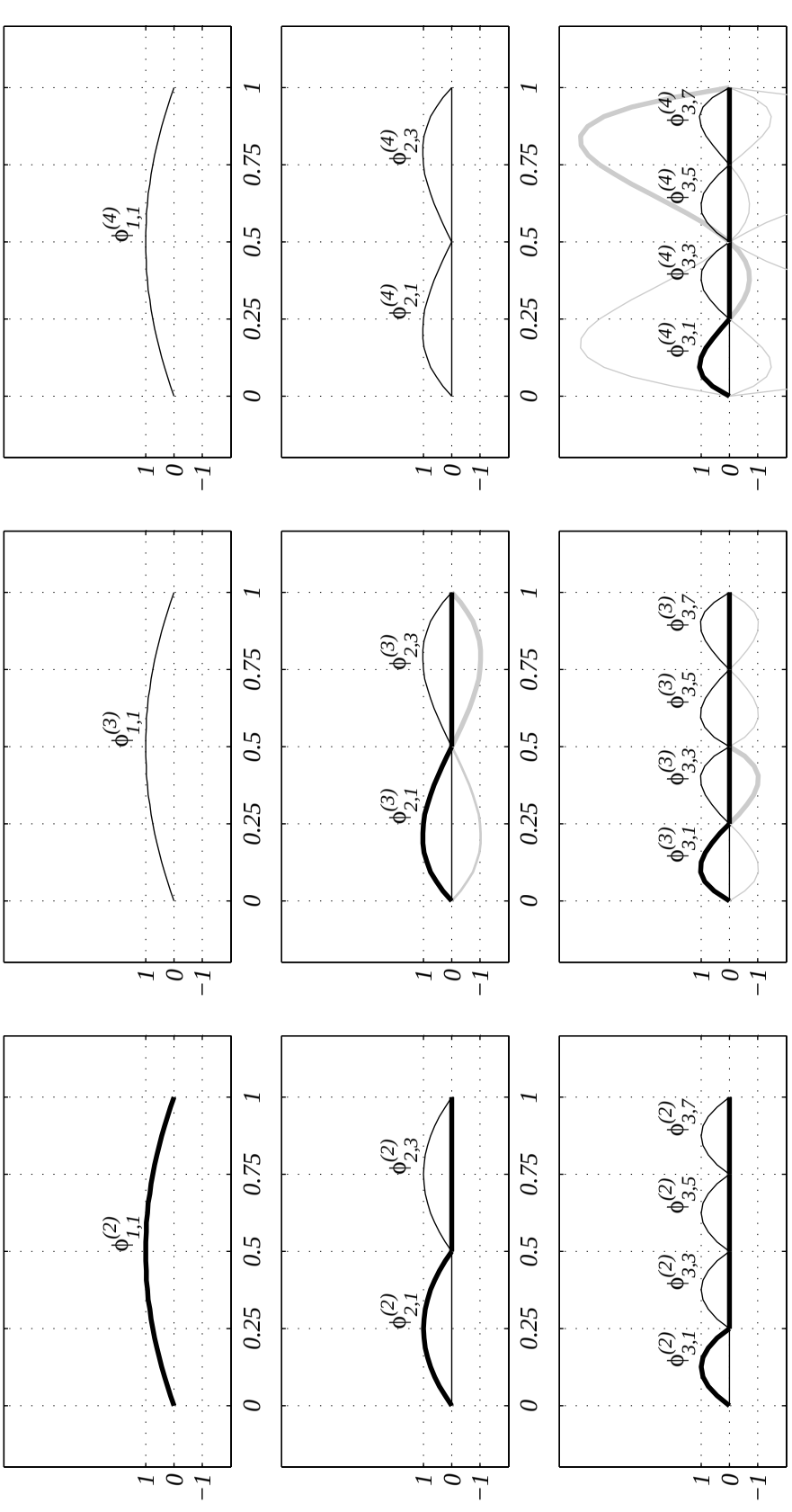


Abbildung 3.10: Hierarchische Basisfunktionen für die hierarchische Lagrange-Interpolation. Für $p = 2$ (links), $p = 3$ (Mitte) und $p = 4$ (rechts) sind die Basisfunktionen der ersten drei Level dargestellt. Zur besseren Unterscheidbarkeit ist jeweils eine Basisfunktion dicker gezeichnet. Die Basisfunktionen sind in schwarz gezeichnet, in grau werden die Polynome über dem Intervall $K^{(p-2)}_{\ell,i}$ fortgesetzt.

Man vergewissert sich leicht, dass (3.45) erfüllt ist. Nach Definition der hierarchischen Interpolation $I_\ell^{(p)}$ und wegen Eigenschaft (3.47) stimmt die Basisfunktion $\phi_{\ell,i}^{(p)}$, $i \in \mathcal{I}_\ell$, über dem Intervall $J_{\ell,i}$ mit dem Polynom vom Grad p überein, das am Punkt $x_{\ell,i}$ den Wert eins besitzt und an den restlichen Punkten von $K_{\ell,i}^{(p-2)}$ verschwindet. Außerhalb von $J_{\ell,i}$ ist $\phi_{\ell,i}^{(p)}$ null, das heißt

$$\operatorname{supp} \phi_{\ell,i}^{(p)} \;=\; J_{\ell,i}.$$

Ferner kann gezeigt werden, dass der Wertebereich für alle hierarchischen Basisfunktionen $\phi_{\ell,i}^{(p)}$, $i \in \mathcal{I}_\ell$ gleichmäßig beschränkt ist. Damit ist gewährleistet, dass bei beschränkten Koeffizienten die zugehörige Funktion im gleichen Maße beschränkt ist, die Darstellung von Funktionen bezüglich dieser Basis also stabil ist. In Abbildung 3.10 sind einige Beispiele für Basisfunktionen zu sehen.

Die Verallgemeinerung der hierarchischen Polynominterpolation auf d Dimensionen erfolgt analog zur stückweise linearen Interpolation durch einen Tensorproduktansatz: Tensorprodukte der univariaten Basisfunktionen spannen die Inkrementräume

$$W_{\boldsymbol{\ell}}^{(\boldsymbol{p})} \;=\; \operatorname{span}\{\phi_{\boldsymbol{\ell},\boldsymbol{i}}^{(\boldsymbol{p})}, \boldsymbol{i} \in \mathcal{I}_{\boldsymbol{\ell}}\}$$

auf. Der Multiindex $\boldsymbol{p}$ erlaubt dabei prinzipiell, den Polynomgrad für jede Raumrichtung individuell zu wählen. Wir beschränken uns auf die isotrope Wahl $\boldsymbol{p} = p \cdot \mathbf{1}$ und setzen $W_{\boldsymbol{\ell}}^{(p)} := W_{\boldsymbol{\ell}}^{(p\cdot\mathbf{1})}$. Erneut kann die Konstruktion von diskreten Funktionenräumen durch eine optimierte Auswahl von Teilräumen $W_{\boldsymbol{\ell}}^{(p)}$ erfolgen.

Bungartz [17] gibt folgende Indexmengen an (vergleiche (3.20), (3.21) und (3.22))

$$\begin{aligned}
\mathcal{L}_{0,n}^{(p,\infty)} \;&:=\; \mathcal{L}_{0,n}^{(\infty)} \;=\; \{\boldsymbol{\ell} \in \mathbb{N}^d; |\boldsymbol{\ell}|_\infty \le n\},\\
\mathcal{L}_{0,n}^{(p,1)} \;&:=\; \mathcal{L}_{0,n}^{(1)} \;=\; \{\boldsymbol{\ell} \in \mathbb{N}^d; |\boldsymbol{\ell}|_1 \le n + d - 1\},\\
\mathcal{L}_{0,n}^{(p,E)} \;&:=\; \left\{\boldsymbol{\ell} \in \mathbb{N}^d; |\boldsymbol{\ell}|_1 - \tfrac{\log_2\left(\sum_{j=1}^d 4^{l_j}\right)}{2p+3} \le (n+d-1) - \tfrac{\log_2(4^n+4d-4)}{2p+3}\right\}.
\end{aligned} \tag{3.49}$$

Die Indexmenge $\mathcal{L}_{0,n}^{(p,1)}$ ist bezüglich des Interpolationsfehlers in der L_∞- und der L_2-Norm optimiert. $\mathcal{L}_{0,n}^{(p,E)}$ ist optimal, wenn der Fehler in der Energienorm gemessen wird. Man beachte, dass $\mathcal{L}_{0,n}^{(p,E)}$ für den Spezialfall $p = 1$ mit $\mathcal{L}_{0,n}^{(E)}$, der Energie-optimierten Indexmenge für die stückweise lineare Interpolation, übereinstimmt.

Für die Interpolation $I^{(p,\star)}(u)$, $\star \in \{\infty, 1, E\}$, $u \in X_0^{p+1}(K^{(d)})$, in den durch

$$V_{0,n}^{(p,\star)} := \bigoplus_{\boldsymbol{\ell} \in \mathcal{L}_{0,n}^{(p,\star)}} W_{\boldsymbol{\ell}}^{(p)}$$

definierten Räumen bekommt man die folgenden Aussagen über das asymptotische

Verhalten des Fehlers in Abhängigkeit von der Zahl der Gitterpunkte [17]:

$$\begin{aligned}
\|u - I_{0,n}^{(p,\infty)}(u)\|_{L_\infty} &= O(N^{-\frac{p+1}{d}}) \\
\|u - I_{0,n}^{(p,1)}(u)\|_{L_\infty} &= O(N^{-(p+1)} \cdot |\log_2 N|^{(p+2)\cdot(d-1)}) \\
\|u - I_{0,n}^{(p,\infty)}(u)\|_{L_2} &= O(N^{-\frac{p+1}{d}}) \\
\|u - I_{0,n}^{(p,1)}(u)\|_{L_2} &= O(N^{-(p+1)} \cdot |\log_2 N|^{(p+2)\cdot(d-1)}) \\
\|u - I_{0,n}^{(p,\infty)}(u)\|_{E} &= O(N^{-\frac{p}{d}}) \\
\|u - I_{0,n}^{(p,1)}(u)\|_{E} &= O(N^{-p} \cdot |\log_2 N|^{p\cdot(d-1)}) \\
\|u - I_{0,n}^{(p,E)}(u)\|_{E} &= O(N^{-p})
\end{aligned} \tag{3.50}$$

Diese Ergebnisse beziehen sich zunächst auf Funktionen, die auf dem Rand verschwinden. Die Argumentation aus Abschnitt 3.3, wie sie dort für die multilineare Interpolation von Funktionen mit nichtverschwindenden Randwerten geführt wurde, gilt analog für die hierarchische Interpolation mit Polynomen höheren Grades. So bekommt man für die aus

$$\mathcal{L}_n^{(p,\star)} = \left\{ \boldsymbol{\ell} \in \mathbb{N}_0^d : \exists \boldsymbol{k} \in \mathcal{L}_{0,n}^{(p,\star)} \text{ mit } \boldsymbol{\ell} \leq \boldsymbol{k} \right\}$$

hervorgehenden Räume $V_n^{(p,\star)}$ und Interpolationsoperatoren $I^{(p,\star)}(u)_n$ sofort das gleiche asymptotische Verhalten (3.50).

Zusammenfassung

An dieser Stelle lohnt eine Zusammenfassung der Ergebnisse aus den vorangegangenen Abschnitten. Mit Hilfe einer Zerlegung von $H^1(K^{(d)})$ in endlich dimensionale, hierarchisch geordnete Teilräume wurde der Grundstein für die effiziente Gestaltung von diskreten Ansatzräumen über dem Referenzelement $K^{(d)} = [0,1]^d$ gelegt. Durch die Optimierung des Kosten-Nutzen-Verhältnisses, also der Approximationsgüte in Relation zur Anzahl der Freiheitsgrade, konnten Räume gefunden werden, deren Approximationseigenschaften denen entsprechender Vollgitter bei weitem überlegen sind. So ergibt sich für den Interpolationsfehler, gemessen in der Energienorm, für feste Raumdimension d und festen Polynomgrad p ein asymptotisches Verhalten von $O(N^{-\frac{p}{d}})$ (Vollgitter) bzw. $O(N^{-p})$ (energiebasiertes Dünngitter), wobei N die Anzahl der Freiheitsgrade ist. Abgesehen von dem sehr erstaunlichen Umstand, dass das Dünngitterverfahren für beliebiges d das gleiche asymptotische Fehlerverhalten an den Tag legt, kann man es als Verfahren höherer Ordnung interpretieren. So entspricht einem Dünngitterverfahren der Ordnung p ein Vollgitterverfahren der Ordnung $d \cdot p$.

3.5 Lokal adaptive Elemente

Die bislang vorgestellten Ansatzräume sind direkte Summen der Inkremente $W_{\boldsymbol{\ell}}^{(p)}$. Die Entscheidung, welche dieser Teilräume in den Ansatzraum konkret aufgenommen werden, wird dabei im Hinblick auf die Approximation einer Funktion u getroffen, von der lediglich bekannt ist, dass gewisse ihrer gemischten partiellen Ableitungen beschränkt sind. Dieses Vorgehen ist sinnvoll, wenn keine weiteren Informationen über die zu approximierende Funktion vorliegen. Man spricht in diesem Zusammenhang von einer *a-priori-Adaption* des Ansatzraums.

Im Gegensatz hierzu steht die *a-posteriori-Adaption*, die aufbauend auf eine bereits gegebene Näherung $\tilde{u} \in V$ und der damit verfügbaren Information über u einen feineren Raum $V' \supset V$ konstruiert. Insbesondere führen lokale Informationen zu einer lokalen Verfeinerung. Für unsere Zwecke bedeutet dies, dass nicht die Inkremente $W_{\boldsymbol{\ell}}^{(p)}$ jeweils als Ganzes in den zu konstruierenden Raum aufgenommen werden, sondern die Entscheidung über die Aufnahme für jede Basisfunktion $\phi_{\boldsymbol{\ell},\boldsymbol{i}}^{(p)}$ im Einzelnen getroffen wird. Wenn u Singularitäten enthält, verschlechtert sich das asymptotische Verhalten des Approximationsfehlers für regelmäßige Gitter wesentlich. Durch gezieltes Einfügen von Knoten in der Umgebung der Singularitäten kann aber ein ähnlich gutes Approximationsverhalten in der Anzahl der Gitterpunkte erreicht werden, wie es bei regelmäßigen Gittern für glatte Funktionen der Fall ist. Dieser für die *hp*-Methode der FEM bekannte Umstand [39, 48] trifft auch für dünne Gitter zu [38].

Bevor ein einfacher Adaptionsalgorithmus für Dünngitterelemente vorgestellt wird, soll eine kleine Auswahl von Arbeiten über lokal adaptive Dünngitter und ihre Anwendung in verschiedenen Bereichen gegeben werden. Zum einen ist hier die numerische Behandlung von partiellen Differentialgleichungen im Allgemeinen [5, 14, 46] zu nennen. Für die Adaption in Kombination mit dualen Fehlerschätzern sind [19, 38] von Interesse. Die Anwendung adaptiver Dünngitter bei der Strömungssimulation ist in [27, 34, 38] beschrieben.

Im Folgenden wird das Prinzip der lokalen Adaption bei dünnen Gittern dargelegt und ein einfacher Adaptionsalgorithmus vorgestellt. Wir definieren hierzu das lokale, auf die Basisfunktion $\phi_{\boldsymbol{\ell},\boldsymbol{i}}$ bezogene Kosten-Nutzen-Verhältnis

$$\operatorname{cbr}(\boldsymbol{\ell},\boldsymbol{i}) \;:=\; \|v_{\boldsymbol{\ell},\boldsymbol{i}}^{(p)} \cdot \phi_{\boldsymbol{\ell},\boldsymbol{i}}^{(p)}\|^2 \;=\; \left(v_{\boldsymbol{\ell},\boldsymbol{i}}^{(p)}\right)^2 \cdot \|\phi_{\boldsymbol{\ell},\boldsymbol{i}}^{(p)}\|^2. \tag{3.51}$$

Im Vergleich zu (3.19) werden die Kosten hier konstant mit 1 bemessen, da es sich um den Beitrag einer einzelnen Basisfunktion handelt. Die Konstruktion eines optimalen Ansatzraums besteht dann analog zu dem in Abschnitt 3.3 Gesagtem darin, alle Basisfunktionen zu sammeln, die die Bedingung

$$\operatorname{cbr}(\boldsymbol{\ell},\boldsymbol{i}) \;\geq\; \sigma \tag{3.52}$$

für einen vorgegebenen Schwellwert $\sigma > 0$ erfüllen.

Dieses Ziel exakt zu erreichen würde allerdings die vollständige Information über die zu approximierende Funktion erfordern. Vielmehr macht man sich die Eigenschaft

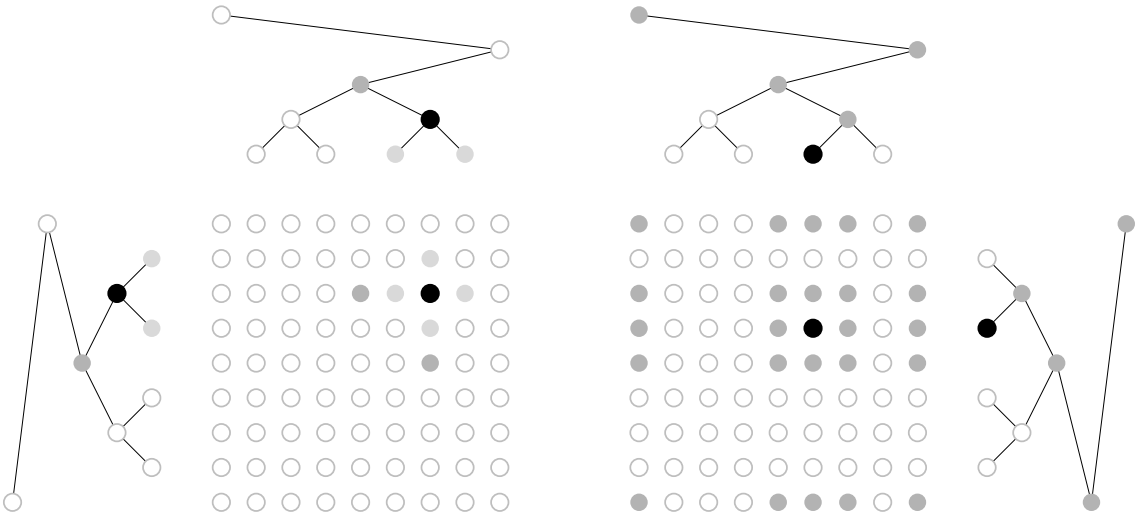


Abbildung 3.11: Links: Väter (grau) und Söhne (hellgrau) eines speziellen Punktes (schwarz). Rechts: Hierarchische Vorfahren (grau) eines speziellen Punktes (schwarz). Sie ergeben zusammen das minimale Gitter, das diesen Punkt enthält.

zu Nutze, dass die Überschüsse und damit auch die Beiträge $\mathrm{cbr}(\boldsymbol{\ell}, \boldsymbol{i})$ für hinreichend glatte Funktionen exponentiell in $|\boldsymbol{\ell}|_1$ fallen, vergleiche (3.11). Ein praktikabler Adaptionsalgorithmus besteht somit darin, ausgehend von einem vorgegebenen Raum schrittweise neue Basisfunktionen höheren Levels hinzuzunehmen und dies solange zu wiederholen, bis von den neuen Basisfunktion keine mehr einen Beitrag größer als σ liefert.

Im Folgenden wollen wir den Algorithmus konkretisieren. Auf Seite 40 wurden die Punkte des eindimensionalen Gitters in einem Binärbaum angeordnet und dadurch die Begriffe *Sohn* und *Vater* eines Knoten definiert. Im Mehrdimensionalen wollen wir diese Begriffe weiter benutzen und nennen einen Knoten Sohn bzw. Vater eines anderen Knotens, wenn die Relation entlang genau einer Raumrichtung gegeben ist. Die Gesamtheit aller Väter, Großväter, usw. ergeben dann die hierarchische Vorfahren. Dies wird in Abbildung 3.11 veranschaulicht. Um die Darstellung einfach zu halten, nehmen wir dabei die Randpunkte als Vorfahren des Punktes $x_{1,1} = \frac{1}{2}$ in die Hierarchie mit auf, etwa $\mathcal{V}(1,1) = (0,1)$ und $\mathcal{V}(0,1) = (0,0)$. Ferner wollen wir einen Knoten des Gitters G als *Blattknoten* bezeichnen, wenn mindestens einer seiner Söhne in G fehlt.

Mit Hilfe dieser Terminologie definieren wir nun den folgenden Algorithmus:

Algorithmus: $adap(G, u, \sigma)$
Eingabe: Gitter G,
Funktion u, gegeben durch die hierarchischen Überschüsse über G,
Schwellwert σ
Ausgabe: verfeinertes Gitter G, u über G

```
wiederhole
    Für alle Blattknoten x_{ℓ,i} von G
        Berechne c := cbr(ℓ, i)
        Falls c > σ
            Füge in G alle Söhne von x_{ℓ,i} ein
    Vervollständige G
    Stelle Funktion u auf G dar
bis keine neuen Punkte eingefügt worden sind
```

Um $\mathrm{cbr}(\boldsymbol{\ell}, \boldsymbol{i})$ nach (3.51) für die Blattknoten zu berechnen, benötigt man neben den hierarchischen Überschüssen die Normen der Basisfunktionen. In [17] werden hierfür folgende Abschätzungen gegeben:

$$\begin{aligned} \left\| \phi^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}} \right\|_{L_\infty} &\leq C_{L_\infty}, \\ \left\| \phi^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}} \right\|_{L_2} &\leq C_{L_2} \cdot 2^{-|\boldsymbol{\ell}|_1/2}, \\ \left\| \phi^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}} \right\|_{E} &\leq C_E \cdot 2^{-|\boldsymbol{\ell}|_1/2} \cdot \left(\sum_{j=1}^{d} 4^{l_j} \right)^{1/2}. \end{aligned} \tag{3.53}$$

Die auftretenden Konstanten hängen nur von d ab, nicht aber von p und $\boldsymbol{\ell}$, so dass wir für den Beitrag $\mathrm{cbr}(\boldsymbol{\ell}, \boldsymbol{i})$ folgende vereinfachte Formel verwenden können:

$$\mathrm{cbr}(\boldsymbol{\ell}, \boldsymbol{i}) := \begin{cases} \left(v^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}} \right)^2 & \text{für die } L_\infty\text{-Norm}, \\ \left(v^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}} \right)^2 \cdot 2^{-|\boldsymbol{\ell}|_1} & \text{für die } L_2\text{-Norm}, \\ \left(v^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}} \right)^2 \cdot 2^{-|\boldsymbol{\ell}|_1} \cdot \sum_{j=1}^{d} 4^{l_j} & \text{für die Energienorm}. \end{cases} \tag{3.54}$$

Die Zeile *„Vervollständige G“* nach dem Einfügen der neuen Knoten bedarf noch einer Erklärung: Analog zur Bedingung (3.14), die es bei der Konstruktion von Ansatzräumen aus den Inkrementen $W^{(p)}_{\boldsymbol{\ell}}$ einzuhalten galt, muss auch bei der lokalen Erweiterung des Ansatzraums darauf geachtet werden, dass keine Unterbrechungen in der Hierarchie auftreten. Das heißt, dass mit jedem Knoten eines Gitters G auch

seine hierarchischen Vorfahren in G enthalten sein müssen. Werden einem hierarchisch vollständigen Gitter neue Knoten hinzugefügt, muss in der Regel im Anschluss daran noch sicher gestellt werden, dass das so verfeinerte Gitter auch alle hierarchischen Vorfahren der neuen Knoten enthält.

Die Notwendigkeit zur Vervollständigung wirft an dieser Stelle die Frage auf, ob die in (3.51) als konstant angesetzten Kosten pro neu hinzugefügter Basisfunktion gerechtfertigt ist. Die Anzahl aller hierarchischen Vorfahren von $x_{\boldsymbol{\ell},\boldsymbol{i}}$ ist gegeben durch (vergleiche Abbildung 3.11)

$$\prod_{j=1}^{d}(\ell_j + 2).$$

Sie ist eine obere Schranke für die Zahl der zusätzlich aufzubringenden Gitterpunkte. Bei unserem Algorithmus ist wenigstens einer der Väter bereits im Ausgangsgitter vorhanden. Damit reduziert sich die obere Schranke auf

$$\prod_{\substack{j=1,\ldots,d\\ j\neq k}} (\ell_j + 2),$$

wenn k die Richtung der Vater-Sohn-Beziehung ist. Da es im Allgemeinen schwierig ist, die Zahl der zusätzlichen Knoten vernünftig abzuschätzen – gerade, wenn mehr als nur ein Punkt auf einmal verfeinert wird und gemeinsame Vorfahren berücksichtigt werden müssten – bleiben wir dabei, die Kosten pro verfeinertem Punkt wie in (3.51) als konstant anzusetzen.

Damit der Algorithmus terminiert, müssen die Beiträge $\mathrm{cbr}(\boldsymbol{\ell},\boldsymbol{i})$ abfallen. Ist $u \in X^{p+1}(K^{(d)})$ (siehe (3.12)), so genügen die hierarchischen Überschüsse der Abschätzung

$$|v^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}}| \;\leq\; C \cdot 2^{-(p+1)\cdot|\boldsymbol{\ell}|_1} \cdot |u|_{(p+1)\cdot\mathbf{1},\infty} \tag{3.55}$$

mit einer Konstanten C, die nur von p und d, aber nicht von $\boldsymbol{\ell}$ oder u abhängt [17]. Damit haben wir zunächst eine globale Aussage über das Verhalten der Überschüsse $v^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}}$, zumal in (3.55) das Supremum der gemischten Ableitungen über ganz Ω eingeht. Für die lokale Adaption benötigen wir allerdings eine lokale Aussage. Hierzu erinnern wir uns, dass der hierarchische Überschuss $v_{\ell,i}$ mit dem Interpolationsfehler $u(x_{\ell,i}) - I_{\ell-1}(u)(x_{\ell,i})$ übereinstimmt. In der Abschätzung (3.55) muss demnach das Supremum nicht global genommen werden, sondern lediglich über dem kleinsten Intervall, das alle Stützstellen von $I_{\ell-1}$ enthält. In der Notation von Abschnitt 3.4 ist das das Intervall $J_{\mathcal{V}^{p-1}(\ell,i)}$. Im d-dimensionalen Fall ist das Supremum entsprechend über dem Quader

$$Q^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}} \;:=\; J_{\mathcal{V}^{p-1}(\ell_1,i_1)} \times \cdots \times J_{\mathcal{V}^{p-1}(\ell_d,i_d)}$$

zu bilden. Man hat dann

$$|v^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}}| \;\leq\; C \cdot 2^{-(p+1)\cdot|\boldsymbol{\ell}|_1} \cdot \left|u\big|_{Q^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}}}\right|_{(p+1)\cdot\mathbf{1},\infty}. \tag{3.56}$$

Für jeden Sohn $(\boldsymbol{\ell}',\boldsymbol{i}')$ von $(\boldsymbol{\ell},\boldsymbol{i})$ ist offensichtlich

$$Q^{(p)}_{\boldsymbol{\ell}',\boldsymbol{i}'} \;\subset\; Q^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}}$$

und damit

$$\left|u|_{Q^{(p)}_{\boldsymbol{\ell}',\boldsymbol{i}'}}\right|_{(p+1)\cdot\mathbf{1},\infty} \;\leq\; \left|u|_{Q^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}}}\right|_{(p+1)\cdot\mathbf{1},\infty}.$$

Wegen $|\boldsymbol{\ell}'|_1 = |\boldsymbol{\ell}| + 1$ hat man dann mit (3.56)

$$|v^{(p)}_{\boldsymbol{\ell}',\boldsymbol{i}'}| \;\leq\; C \cdot 2^{-(p+1)\cdot|\boldsymbol{\ell}'|_1} \cdot \left|u|_{Q^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}}}\right|_{(p+1)\cdot\mathbf{1},\infty} \;\approx\; 2^{-(p+1)} \cdot |v^{(p)}_{\boldsymbol{\ell},\boldsymbol{i}}|. \tag{3.57}$$

Mit (3.54) bekommen wir

$$\frac{\mathrm{cbr}_{L_\infty}(\boldsymbol{\ell},\boldsymbol{i})}{\mathrm{cbr}_{L_\infty}(\boldsymbol{\ell}',\boldsymbol{i}')} \;\approx\; 2^{2(p+1)}, \tag{3.58}$$

$$\frac{\mathrm{cbr}_{L_2/E}(\boldsymbol{\ell},\boldsymbol{i})}{\mathrm{cbr}_{L_2/E}(\boldsymbol{\ell}',\boldsymbol{i}')} \;\approx\; 2^{2(p+1)+1}. \tag{3.59}$$

Mit dem Zeichen $\approx$ wird dem Umstand Rechnung getragen, dass eine echte Abschätzung eine zu (3.56) ähnliche Abschätzung nach unten erfordern würde. Streng genommen ist (3.57) im asymptotischen Grenzfall $|\boldsymbol{\ell}|_1 \to \infty$ zu verstehen, eine Aussage für kleine $|\boldsymbol{\ell}|_1$ lässt sich damit nicht ableiten. Für die Praxis bedeutet dies, dass es letztendlich in der Verantwortung des Benutzers liegt, dem Adaptionsalgorithmus ein bereits hinreichend feines Gitter vorzugeben, damit die Überschüsse an den Blättern und an deren hierarchischen Nachfahren das asymptotische Verhalten (3.57) wiedergeben und der monotone Abfall der Beiträge $\mathrm{cbr}(\boldsymbol{\ell},\boldsymbol{i})$ gewährleistet ist.

Zur Steuerung des Schwellwerts σ

Für eine feste Schranke σ bricht der Algorithmus auf Grund des exponentiellen Abfalls der Beiträge $\mathrm{cbr}(\boldsymbol{\ell},\boldsymbol{i})$ nach einer endlichen Zahl von Verfeinerungsschritten ab. Um die Approximation noch weiter zu führen, kann der Algorithmus mit dem Ergebnis aus dem ersten Durchlauf und einem niedrigeren Wert für σ' erneut gestartet werden. Nach (3.58) und (3.59) empfiehlt es sich $\sigma' = \sigma \cdot 2^{-2(p+1)}$ bzw. $\sigma' = \sigma \cdot 2^{-2(p+1)-1}$ zu wählen. Dieses Vorgehen erzeugt in der Regel eine Folge von Gittern, bei der die Zahl der Gitterpunkte nur langsam zunimmt, weil die Zahl von zu verfeinernden Blattknoten von Schritt zu Schritt tendenziell abnimmt, solange σ festgehalten wird. Da aber bei jeder Verfeinerung die Approximation von u neu berechnet werden muss, ist man an einer möglichst raschen Zunahme von Gitterpunkten interessiert. In der Praxis hat es sich deshalb als günstig erwiesen, σ bereits bei jeder Verfeinerung neu zu wählen. Ein sinnvoller Wert ist dabei der größte Beitrag $\mathrm{cbr}(\boldsymbol{\ell},\boldsymbol{i})$ an den Blattknoten dividiert durch $2^{2(p+1)}$ bzw. $2^{2(p+1)+1}$.

Abbildung 3.12 zeigt den durch die lokale Adaption erreichten Leistungsvorsprung an Hand der Interpolation der Funktion

$$u(x,y) \;=\; \mathrm{Im}(z^{2/3}), \qquad z = 2(x+iy) - 1. \tag{3.60}$$

Durch die Adaption erreicht man ein Konvergenzverhalten bezüglich der Zahl der Gitterpunkte, wie es sonst für glatte Funktionen gelten würde.

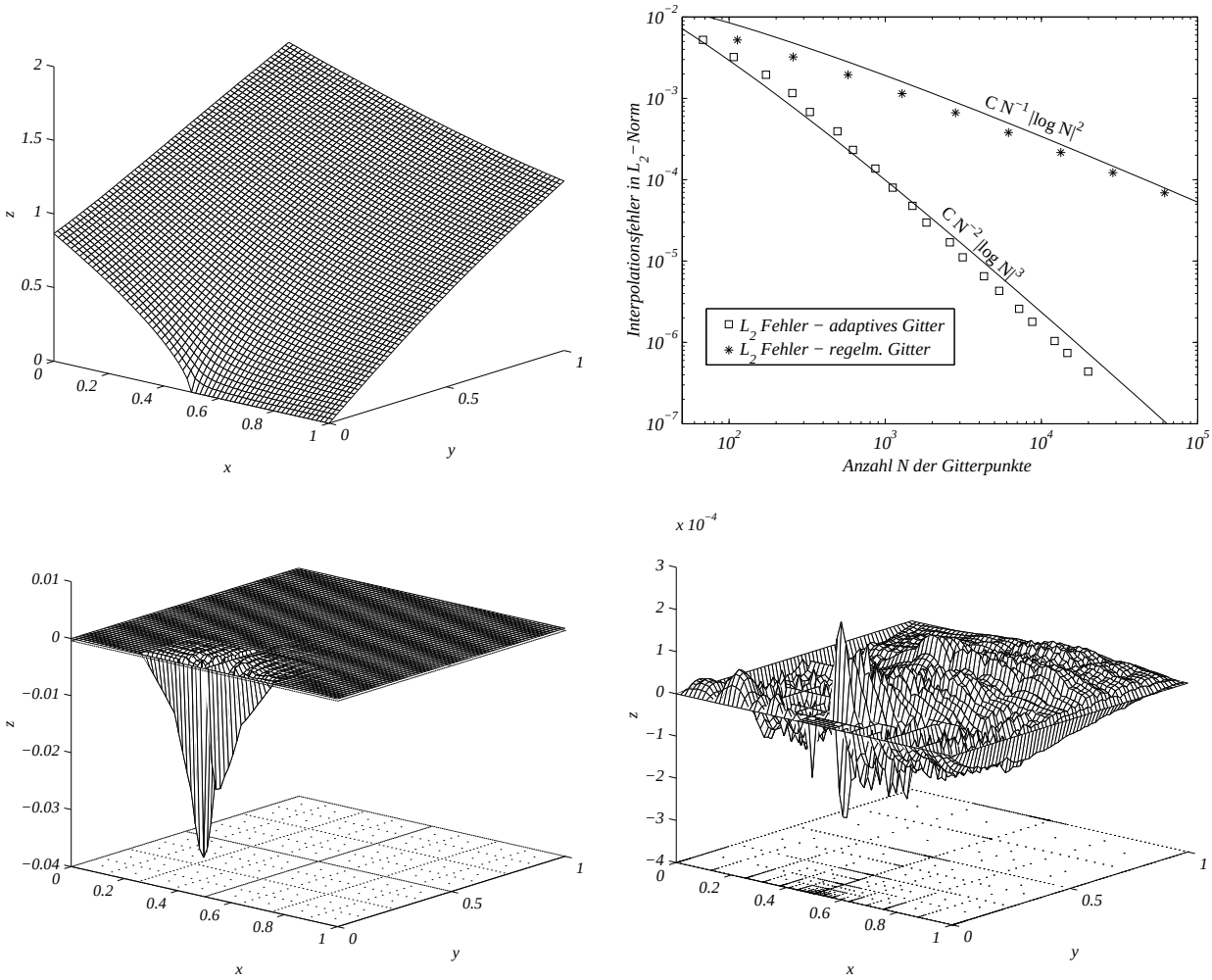


Abbildung 3.12: Links oben: Funktion u mit Wurzelsingularität. Rechts oben: Interpolationsfehler in L_2-Norm bei gleichmäßiger und bei adaptiver (lokaler) Verfeinerung. Unten: Fehler bei regelmäßigem Dünngitter (1281 Gitterpunkte) und bei adaptivem Dünngitter (1114 Gitterpunkte)

3.6 Algorithmische Betrachtungen

In diesem Anschnitt sollen die wichtigsten Algorithmen beschrieben werden, die für die Implementierung des im folgenden Kapitel beschriebenen Finite-Element-Verfahrens von entscheidender Bedeutung sind. Dabei handelt es sich zunächst um Algorithmen zur Auswertung von Integralen, wie sie in der schwachen Formulierung eines Randwertproblems vorkommen. Diese Algorithmen wurden unter dem Stichwort „Unidirektionales Prinzip“ erstmals in [5, 6] beschrieben und später in [15, 17] weiterentwickelt. Wegen der außerordentlichen Bedeutung dieser Algorithmen für ein effizientes FE-Verfahren werden sie in Abschnitt 3.6.1 eingehend beschrieben. Der Abschnitt 3.6.2 zeigt, wie diese Algorithmen für eine erweiterte Klasse von Funktionalen ausgebaut werden können, bei denen noch ein zusätzlicher Faktor im Integral auftaucht. In den Abschnitten 3.6.3 und 3.6.4 wird kurz erläutert, wie der Wechsel von der hierarchischen in die dehierarchische Basis und umgekehrt sowie diskrete Ableitungsoperatoren mit unidirektionalen Algorithmen realisiert werden. Schließlich wird in Abschnitt 3.6.5 dargestellt, wie die Algorithmen zu den transponierten Operatoren zu finden sind.

3.6.1 Auswertung bestimmter Integraloperatoren

Im Folgenden sei G ein dünnes Gitter über $[0,1]^d$ und V ein darüber definierter diskreter Funktionenraum. Zur Identifikation der Basisfunktionen benutzen wir der Einfachheit halber den zugeordneten Gitterpunkt, setzen also $\phi_{\boldsymbol{r}} := \phi_{\boldsymbol{\ell},i}$, wenn $\boldsymbol{r} = \boldsymbol{x}_{\boldsymbol{\ell},i}$.

Gesucht ist ein Algorithmus, der zu einer Funktion $u \in V$, gegeben durch ihre hierarchischen Überschüsse $u_{\boldsymbol{r}}$, $\boldsymbol{r} \in G$, sämtliche Integrale

$$v_{\boldsymbol{r}} \;=\; \int_{[0,1]^d} D^{\boldsymbol{\alpha}} u(\boldsymbol{x})\, D^{\boldsymbol{\beta}} \phi_{\boldsymbol{r}}(\boldsymbol{x})\, d\boldsymbol{x}, \qquad \boldsymbol{r} \in G \tag{3.61}$$

berechnet und dies mit einem Aufwand $O(N)$ erreicht, wenn N die Anzahl der Gitterpunkte ist.

Der eleganteste Ansatz für die Berechnung der $v_{\boldsymbol{r}}$ basiert auf dem sogenannten *unidirektionalen Prinzip*, bei dem das Integral in (3.61) unter Ausnutzung der Tensorprodukteigenschaft der Basisfunktionen auf eindimensionale Integrale zurückgeführt wird. Diese eindimensionalen Integrale können ihrerseits mit jeweils zwei Durchläufen durch die Datenstruktur an allen Knoten ausgewertet werden. Auf den nächsten Seiten wird dies präzisiert. Wir starten mit dem Spezialfall $d = 1$.

Der eindimensionale Fall

Sei u gegeben durch

$$u(x) \;=\; \sum_{r \in G} u_r \cdot \phi_r(x).$$

Die Knoten r seien dabei in einem Baum angeordnet mit den daraus resultierenden Bezeichnungen wie „Nachfahre“ oder „Vorfahre“ eines Knoten (vergleiche Abschnitt 3.4).

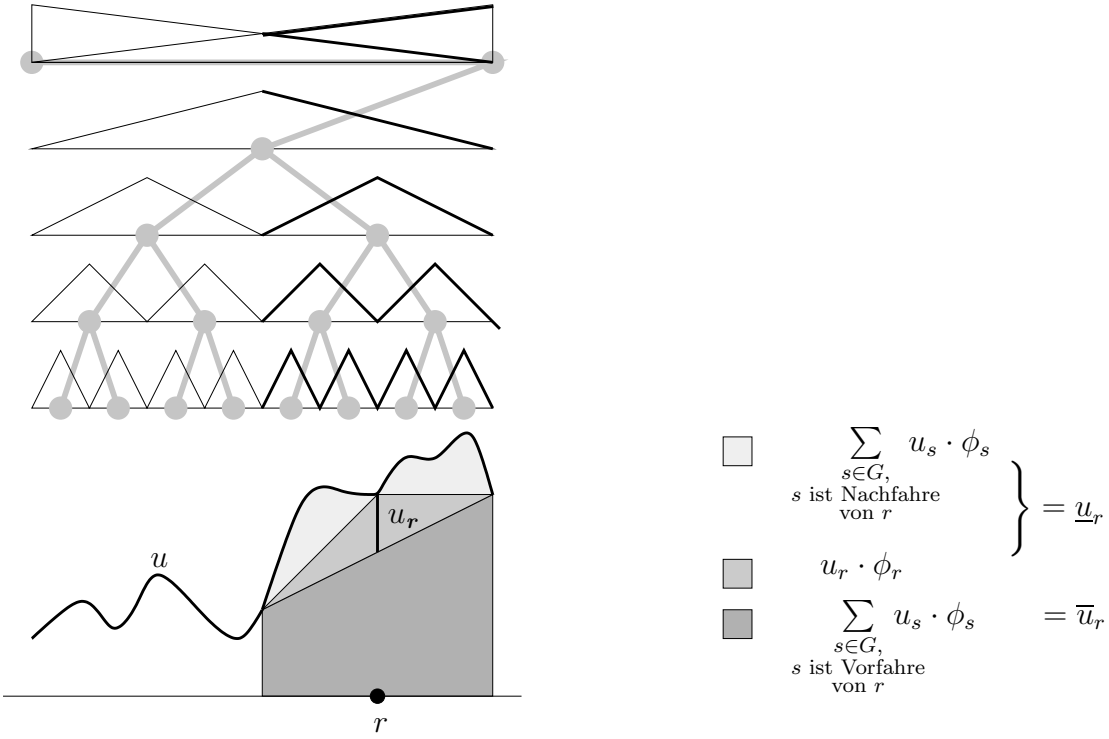


Abbildung 3.13: Zur Zerlegung von $u|_{\operatorname{supp}\phi_r}$ in die Anteile $\overline{u}_r$ und $\underline{u}_r$

Zur Auswertung des Integrals

$$v_{\boldsymbol{r}} \;=\; \langle u, \phi_r \rangle \;:=\; \int_{[0,1]} D^\alpha u(x)\, D^\beta \phi_r(x)\, dx \tag{3.62}$$

genügt es, die Einschränkung von u auf das Intervall $T_r := \operatorname{supp}\phi_r$ zu betrachten. An Hand von Abbildung 3.13 erkennt man sofort, dass für $x \in T_r$ die Darstellung

$$u(x) \;=\; \underbrace{\sum_{\substack{s\in G,\\ s \text{ ist Vorfahre von } r}} u_s \cdot \phi_s(x)}_{=:\,\overline{u}_r(x)} \;+\; u_r \cdot \phi_r(x) \;+\; \underbrace{\sum_{\substack{s\in G,\\ s \text{ ist Nachfahre von } r}} u_s \cdot \phi_s(x)}_{=:\,\underline{u}_r(x)} \tag{3.63}$$

gilt. Damit haben wir die folgende Zerlegung des Integralwerts $v_{\boldsymbol{r}}$:

$$v_{\boldsymbol{r}} \;=\; \langle \overline{u}_r, \phi_r \rangle \;+\; \langle \underline{u}_r, \phi_r \rangle$$

Der Anteil $\overline{u}_r$ lässt sich während eines Baumdurchlaufs von oben nach unten (*top-down*) an allen Knoten r im Gitter rekonstruieren. Dazu startet man für die Randknoten $r \in \{0, 1\}$ und für den Mittelknoten $r = \frac{1}{2}$ mit dem linearen Polynom

$$\overline{u}_r \;=\; u_0 \cdot \phi_0 \;+\; u_1 \cdot \phi_1 \tag{3.64}$$

und nutzt beim weiteren Abstieg, dass für ein gegebenes $\overline{u}_r$ die Funktionen $\overline{u}_{\mathbb{S}_1(r)}$ und $\overline{u}_{\mathbb{S}_2(r)}$ für die Söhne von r durch die folgenden Beziehungen gegeben sind:

$$\begin{aligned} \overline{u}_{\mathbb{S}_1(r)} &= (\overline{u}_r + u_r \cdot \phi_r)\,|_{T_{\mathbb{S}_1(r)}}, \\ \overline{u}_{\mathbb{S}_2(r)} &= (\overline{u}_r + u_r \cdot \phi_r)\,|_{T_{\mathbb{S}_2(r)}}. \end{aligned} \tag{3.65}$$

Die Funktionen $\overline{u}_r$ sind Polynome über T_r und werden im Rechner durch die Koeffizienten bezüglich einer geeigneten Basis dargestellt,

$$\overline{u}_r = \boldsymbol{a}_r^T \boldsymbol{f}_r \tag{3.66}$$

mit dem Koeffizientenvektor $\boldsymbol{a}_r = (a_r^0, \ldots, a_r^p)^T \in \mathbb{R}^{p+1}$ und dem Vektor $\boldsymbol{f}_r = (f_r^0, \ldots, f_r^p)^T$ aus $p+1$ Basispolynomen $f_r^j : T_r \to \mathbb{R}$. Hier ist p der Grad der Polynome. Die triviale Basis $\{f_r^j(x) = x^j\}$ ist zwar vom Knoten r unabhängig, hat aber den Nachteil, dass die Darstellung der $\overline{u}_r$ mit zunehmender Baumtiefe instabil wird. Besser geeignet ist die Basis

$$f_r^j(x) := \frac{1}{j!} \cdot \left(\frac{x-r}{h_r} \right)^j, \qquad r \in (0,1) \tag{3.67}$$

wobei h_r die halbe Länge des Intervalls T_r ist. An den Randpunkten $r \in \{0,1\}$ wird die Basis vom Mittelknoten $r = \frac{1}{2}$ übernommen, also per Definition

$$\boldsymbol{f}_0 = \boldsymbol{f}_1 = \boldsymbol{f}_{\frac{1}{2}}$$

gesetzt. Für verschiedene Knoten r unterscheiden sich die f_r^j im allgemeinen, jedoch lassen sich je zwei Basispolynome f_r^j und f_s^j durch Translation und Streckung bzw. Stauchung ineinander überführen. Konkret hat man für die Einschränkung der Basispolynome in $\boldsymbol{f}_r$ auf die den Söhnen von r zugeordneten Intervallen die Transformationsregeln

$$\begin{aligned} \boldsymbol{f}_r|_{T_{\mathbb{S}_1(r)}} &= \boldsymbol{R}_1 \cdot \boldsymbol{f}_{\mathbb{S}_1(r)}, \\ \boldsymbol{f}_r|_{T_{\mathbb{S}_2(r)}} &= \boldsymbol{R}_2 \cdot \boldsymbol{f}_{\mathbb{S}_2(r)}, \end{aligned}$$

mit Matrizen $\boldsymbol{R}_1$, $\boldsymbol{R}_2 \in \mathbb{R}^{(p+1)\times(p+1)}$, gegeben durch ihre Einträge $(0 \le i, j \le p)$

$$\boldsymbol{R}_{1,ij} = \begin{cases} \frac{(-1)^{i-j}}{(i-j)!} \cdot 2^{-i}, & \text{falls } i \ge j \\ 0, & \text{sonst,} \end{cases}$$

$$\boldsymbol{R}_{2,ij} = \begin{cases} \frac{1}{(i-j)!} \cdot 2^{-i}, & \text{falls } i \ge j \\ 0, & \text{sonst.} \end{cases}$$

Es handelt sich also um untere Dreiecksmatrizen.

Die Basisfunktionen ϕ_r lassen sich, sofern $p \ge 2$ ist, ebenfalls bezüglich $\boldsymbol{f}_r$ darstellen, also

$$\phi_r = \boldsymbol{b}_r^T \boldsymbol{f}_r. \tag{3.68}$$

Hier zahlt sich die spezielle Definition der Basis $\boldsymbol{f}_r$ nach (3.67) aus. Denn bei festem Polynomgrad p im Ansatzraum sind die Basisfunktionen ϕ_r, die auf einem Level $\ell > p-1$ sitzen, bis auf Translationen und Stauchungen Reproduktionen von Basisfunktionen des Levels $\ell = p-1$, vergleiche hierzu Abbildung 3.10 auf Seite 41. Dies hat zur Folge, dass die Koeffizienten $\boldsymbol{b}_r$ bezüglich der Basis $\boldsymbol{f}_r$ ab $\ell > p-1$ die gleichen wie für $\ell = p-1$ sind und somit nur einmal tabelliert werden müssen. Dies gilt auch für stückweise lineare Basisfunktionen. Allerdings sind sie keine Polynome und haben damit keine Darstellung der Art (3.68). Für sie muss entweder die Basis $\boldsymbol{f}_r$ geeignet erweitert werden, oder man denkt sich die Hüte aus linearen Polynomen über dem linken und rechten Halbträger zusammengesetzt.

Damit sind wir nun in der Lage die rekursive Rekonstruktion der $\overline{u}_r$ mit Hilfe ihrer Koeffizienten $\boldsymbol{a}_r$ zu formulieren, wie sie für die Implementierung des Algorithmus benötigt wird. Wir starten für die Randknoten $r \in \{0,1\}$ sowie für den Knoten $r = \frac{1}{2}$ mit

$$\boldsymbol{a}_r \;=\; (0.5 \cdot (u_0 + u_1), 0.5 \cdot (u_1 - u_0), 0, \ldots, 0)^T, \tag{3.69}$$

was dem linearen Polynom (3.64) entspricht. Setzen wir (3.66) und (3.68) in (3.65) ein, so erhalten wir

$$\begin{aligned}
\overline{u}_{\mathbb{S}_1(r)} &= \boldsymbol{a}_{\mathbb{S}_1(r)}^T \boldsymbol{f}_{\mathbb{S}_1(r)} = \boldsymbol{a}_r^T \boldsymbol{R}_1 \boldsymbol{f}_{\mathbb{S}_1(r)} + u_r \cdot \boldsymbol{b}_r^T \boldsymbol{R}_1 \boldsymbol{f}_{\mathbb{S}_1(r)}, \\
\overline{u}_{\mathbb{S}_2(r)} &= \boldsymbol{a}_{\mathbb{S}_2(r)}^T \boldsymbol{f}_{\mathbb{S}_2(r)} = \boldsymbol{a}_r^T \boldsymbol{R}_2 \boldsymbol{f}_{\mathbb{S}_2(r)} + u_r \cdot \boldsymbol{b}_r^T \boldsymbol{R}_2 \boldsymbol{f}_{\mathbb{S}_2(r)},
\end{aligned}$$

und damit für die Koeffizienten die Rekursionsvorschriften

$$\begin{aligned}
\boldsymbol{a}_{\mathbb{S}_1(r)} &= \boldsymbol{R}_1^T \cdot (\boldsymbol{a}_r + u_r \cdot \boldsymbol{b}_r), \\
\boldsymbol{a}_{\mathbb{S}_2(r)} &= \boldsymbol{R}_2^T \cdot (\boldsymbol{a}_r + u_r \cdot \boldsymbol{b}_r).
\end{aligned}$$

Der zu berechnende Wert $\langle \overline{u}_r, \phi_r \rangle$ ergibt sich zu

$$\langle \overline{u}_r, \phi_r \rangle \;=\; \boldsymbol{b}_r^T \boldsymbol{M}_r \boldsymbol{a}_r,$$

wobei $\boldsymbol{M}_r \in \mathbb{R}^{(p+1)\times(p+1)}$ gegeben ist durch

$$\boldsymbol{M}_{r,ij} \;:=\; \langle f_r^j, f_r^i \rangle. \tag{3.70}$$

Kommen wir nun zum Anteil, den $\underline{u}_r$ zu v_r beiträgt. Dieser lässt sich ebenfalls in einem Baumdurchlauf an allen Knoten r auswerten, diesmal allerdings mit einem Datentransfer von den Blättern in Richtung Wurzel (*bottom-up*). Die Rekursionsvorschrift lautet

$$\underline{u}_r \;=\; \begin{cases}
u_r \cdot \phi_r, & \text{falls } r \text{ keine Söhne besitzt,} \\
u_r \cdot \phi_r + \underline{u}_{\mathbb{S}_1(r)} & \text{falls linker Sohn von } r \text{ existiert, nicht aber rechter,} \\
u_r \cdot \phi_r + \underline{u}_{\mathbb{S}_2(r)} & \text{falls rechter Sohn von } r \text{ existiert, nicht aber linker,} \\
u_r \cdot \phi_r + \underline{u}_{\mathbb{S}_1(r)} + \underline{u}_{\mathbb{S}_2(r)} & \text{falls beide Söhne von } r \text{ existieren.}
\end{cases}$$

Mit Hilfe der Koeffizienten $\boldsymbol{b}_r$ von ϕ_r hat man

$$\langle \underline{u}_r, \phi_r \rangle \;=\; u_r \cdot \langle \phi_r, \phi_r \rangle + \langle \underline{u}_{\mathbb{S}_1(r)}, \boldsymbol{b}_r^T \boldsymbol{f}_r \rangle + \langle \underline{u}_{\mathbb{S}_2(r)}, \boldsymbol{b}_r^T \boldsymbol{f}_r \rangle, \tag{3.71}$$

wobei der zweite bzw. der dritte Summand wegfallen, wenn die entsprechenden Söhne von r nicht existieren. Diese Form ist für die Auswertung von $\langle \underline{u}_r, \phi_r \rangle$ noch nicht geeignet, weil die Faktoren in den beiden letzten Bilinearformen auf unterschiedlichen Leveln leben. Statt der Rekursion (3.71) berechnet man an jedem Knoten zunächst den Vektor

$$\boldsymbol{c}_r \;:=\; \langle \underline{u}_r, \boldsymbol{f}_r \rangle.$$

Man erhält ihn aus der Rekursion

$$\begin{aligned}
\boldsymbol{c}_r &= u_r \cdot \langle \phi_r, \boldsymbol{f}_r \rangle \;+\; \langle \underline{u}_{\mathrm{S}_1(r)}, \boldsymbol{f}_r \rangle \;+\; \langle \underline{u}_{\mathrm{S}_2(r)}, \boldsymbol{f}_r \rangle \\
&= u_r \cdot \langle \phi_r, \boldsymbol{f}_r \rangle \;+\; \langle \underline{u}_{\mathrm{S}_1(r)}, \boldsymbol{R}_1 \boldsymbol{f}_{\mathrm{S}_1(r)} \rangle \;+\; \langle \underline{u}_{\mathrm{S}_2(r)}, \boldsymbol{R}_2 \boldsymbol{f}_{\mathrm{S}_2(r)} \rangle \\
&= u_r \cdot \langle \phi_r, \boldsymbol{f}_r \rangle \;+\; \boldsymbol{R}_1 \boldsymbol{c}_{\mathrm{S}_1(r)} \;+\; \boldsymbol{R}_2 \boldsymbol{c}_{\mathrm{S}_2(r)} \\
&= u_r \cdot \boldsymbol{M}_r \boldsymbol{b}_r + \; \boldsymbol{R}_1 \boldsymbol{c}_{\mathrm{S}_1(r)} \;+\; \boldsymbol{R}_2 \boldsymbol{c}_{\mathrm{S}_2(r)}.
\end{aligned}$$

Mit Hilfe von $\boldsymbol{c}_r$ berechnet man dann den gesuchten Wert

$$\langle \underline{u}_r, \phi_r \rangle \;=\; \boldsymbol{b}_r^T \boldsymbol{c}_r.$$

Damit ist der Algorithmus zur Berechnung der v_r aus den u_r vollständig beschrieben. Die beiden Teilalgorithmen *TopDown* und *BottomUp* sind in Abbildung 3.14 noch einmal schematisch dargestellt. Wir fassen zusammen: Die Berechnung erfolgt in zwei Baumdurchläufen, einmal mit Datentransport von oben nach unten, einmal mit Datentransport in der entgegengesetzten Richtung. Jeder Knoten wird also genau zweimal besucht. Je Knoten und Durchlauf sind Matrixmultiplikationen mit $\boldsymbol{M}_r$ und $\boldsymbol{R}_1$ bzw. $\boldsymbol{R}_2$ auszuführen. Da es sich bei diesen Elementarmatrizen um $(p+1)\times(p+1)$- Matrizen handelt, ist die Rechenzeit pro Knoten von der Ordnung p^2. Insgesamt beläuft sich der Rechenaufwand für die Auswertung der Integrale an allen Punkten auf $O(N \cdot p^2)$, wenn N die Anzahl der Gitterpunkte ist. Im Hinblick auf den Speicheraufwand ist folgendes festzustellen: Die Datenvektoren $\boldsymbol{a}_r$ bzw. $\boldsymbol{c}_r$ haben die Länge $p+1$ und werden auf einem Stack gespeichert, dessen maximale Länge mit der Tiefe des Baums übereinstimmt. Der Speicheraufwand für die Hilfsvariablen ist also mit $O(p \cdot \log_2 N)$ anzusetzen, wenn N die Anzahl der Knoten ist und der Baum hinreichend ausbalanciert ist.

Algorithmus: *TopDown*
Eingabe: Gitter G, u_r für alle $r \in G$
Ausgabe: $\overline{v}_r = \langle \overline{u}_r, \phi_r \rangle$ für alle $r \in G$

Für $r \in \{0,1\}$
  Setze $\boldsymbol{a}_r$ wie in (3.69)
  Berechne $\overline{v}_r = \boldsymbol{b}_r^T \boldsymbol{M}_r \boldsymbol{a}_r$
Falls $\frac{1}{2} \in G$
  Setze $\boldsymbol{a}_{\frac{1}{2}}$ wie in (3.69)
  Rufe *Rekursion* für $r = \frac{1}{2}$ mit Datenvektor $\boldsymbol{a}_{\frac{1}{2}}$ auf
Ende

Rekursion Start
  Berechne $\overline{v}_r = \boldsymbol{b}_r^T \boldsymbol{M}_r \boldsymbol{a}_r$
  Falls linker Sohn $\mathcal{S}_1(r)$ existiert
    Berechne $\boldsymbol{a}_{\mathcal{S}_1(r)} = \boldsymbol{R}_1^T \cdot (\boldsymbol{a}_r + u_r \cdot \boldsymbol{b}_r)$
    Rufe *Rekursion* für Knoten $\mathcal{S}_1(r)$
      mit Datenvektor $\boldsymbol{a}_{\mathcal{S}_1(r)}$ auf
  Falls rechter Sohn $\mathcal{S}_2(r)$ existiert
    Berechne $\boldsymbol{a}_{\mathcal{S}_2(r)} = \boldsymbol{R}_2^T \cdot (\boldsymbol{a}_r + u_r \cdot \boldsymbol{b}_r)$
    Rufe *Rekursion* für Knoten $\mathcal{S}_2(r)$
      mit Datenvektor $\boldsymbol{a}_{\mathcal{S}_2(r)}$ auf
Rekursion Ende

Algorithmus: *BottomUp*
Eingabe: Gitter G, u_r für alle $r \in G$
Ausgabe: $\underline{v}_r = \langle \underline{u}_r, \phi_r \rangle$ für alle $r \in G$

Falls $\frac{1}{2} \in G$
  Rufe *Rekursion* für $r = \frac{1}{2}$ auf und
    hole Datenvektor $\boldsymbol{c}_{\frac{1}{2}}$
Für $r \in \{0,1\}$
  Berechne $\underline{v}_r = \boldsymbol{b}_r^T \boldsymbol{c}_r$, wobei $\boldsymbol{c}_0 = \boldsymbol{c}_1 = \boldsymbol{c}_{\frac{1}{2}}$
Ende

Rekursion Start
  Setze $\boldsymbol{c}_r = u_r \cdot \boldsymbol{M}_r \boldsymbol{b}_r$
  Falls linker Sohn $\mathcal{S}_1(r)$ existiert
    Rufe *Rekursion* für Knoten $\mathcal{S}_1(r)$ auf und
      hole Datenvektor $\boldsymbol{c}_{\mathcal{S}_1(r)}$
    Addiere $\boldsymbol{R}_1 \boldsymbol{c}_{\mathcal{S}_1(r)}$ zu $\boldsymbol{c}_r$
  Falls rechter Sohn $\mathcal{S}_2(r)$ existiert
    Rufe *Rekursion* für Knoten $\mathcal{S}_2(r)$ auf und
      hole Datenvektor $\boldsymbol{c}_{\mathcal{S}_2(r)}$
    Addiere $\boldsymbol{R}_2 \boldsymbol{c}_{\mathcal{S}_2(r)}$ zu $\boldsymbol{c}_r$
  Berechne $\underline{v}_r = \boldsymbol{b}_r^T \boldsymbol{c}_r$
Rekursion Ende

Abbildung 3.14: Die Algorithmen „TopDown“ und „BottomUp“

Der Fall beliebiger Dimension

Um nun für beliebige Raumdimension d die Werte

$$v_{\boldsymbol{r}} \;=\; \int_{[0,1]^d} D^{\boldsymbol{\alpha}} u(\boldsymbol{x})\, D^{\boldsymbol{\beta}} \phi_{\boldsymbol{r}}(\boldsymbol{x})\, d\boldsymbol{x}, \qquad\qquad \boldsymbol{r} \in G, \tag{3.72}$$

zu berechnen, wird die Tensorprodukteigenschaft der $\phi_{\boldsymbol{r}}$ ausgenutzt. Damit haben wir

$$v_{\boldsymbol{r}} \;=\; \int\limits_{[0,1]^d} \sum_{\boldsymbol{s}\in G} u_{\boldsymbol{s}} \cdot \phi_{s_1}^{(\alpha_1)}(x_1) \cdots \phi_{s_d}^{(\alpha_d)}(x_d) \;\cdot\; \phi_{r_1}^{(\beta_1)}(x_1) \cdots \phi_{r_d}^{(\beta_d)}(x_d)\, dx_1 \cdots dx_d.$$

Durch Umsortieren erhalten wir

$$\begin{aligned} v_{\boldsymbol{r}} \;=\; \int_{x_1=0}^{1} \sum_{s_1\in G_1} \Bigg(\int_{x_2=0}^{1} \sum_{s_2\in G_2(s_1)} \Bigg(\cdots \int_{x_d=0}^{1} \sum_{s_d\in G_d(s_1,\ldots,s_{d-1})} u_{\boldsymbol{s}} \cdot \phi_{s_d}^{(\alpha_d)}(x_d) \cdot \phi_{r_d}^{(\beta_d)}(x_d)\, dx_d \\ \cdots \Bigg) \cdot \phi_{s_2}^{(\alpha_2)}(x_2) \cdot \phi_{r_2}^{(\beta_2)}(x_2)\, dx_2 \Bigg) \cdot \phi_{s_1}^{(\alpha_1)}(x_1) \cdot \phi_{r_1}^{(\beta_1)}(x_1)\, dx_1, \end{aligned} \tag{3.73}$$

wobei

$$\begin{aligned} G_1 &:= \{s_1 \in [0,1] : \exists s' \in [0,1]^{d-1} \text{ mit } (s_1, s') \in G\}, \\ G_2(s_1) &:= \{s_2 \in [0,1] : \exists s' \in [0,1]^{d-2} \text{ mit } (s_1, s_2, s') \in G\}, \\ &\vdots \\ G_d(s_1,\ldots,s_{d-1}) &:= \{s_d \in [0,1] : (s_1,\ldots,s_{d-1}, s_d) \in G\}. \end{aligned}$$

Angesichts der Darstellung (3.73) liegt die Versuchung nahe, die Integrale von innen nach außen dadurch zu berechnen, dass man den für $d = 1$ formulierten Algorithmus nacheinander entlang der Gitterlinien in den Richtungen x_d, x_{d-1}, …, x_1 anwendet. Der 1D-Algorithmus war aber so ausgelegt, dass er die Werte v_r genau an den Punkten r liefert, wo auch hierarchische Überschüsse u_r vorhanden sind. In (3.73) würde dies nicht ausreichen, da etwa r_d nicht notwendig in $G_d(s_1,\ldots,s_{d-1})$ liegt. Dieses Problem kann behoben werden, indem man nicht wie in (3.73) nur nach Raumdimensionen sortiert, sondern auch die hierarchische Relation zwischen den Knoten $\boldsymbol{r}$ und $\boldsymbol{s}$ berücksichtigt. Dies soll im Folgenden präzisiert werden.

Hierzu definieren wir für je zwei Knoten $\boldsymbol{r}, \boldsymbol{s} \in [0,1]^d$

$$\begin{aligned} \boldsymbol{s} > \boldsymbol{r} \quad &\Longleftrightarrow \quad \boldsymbol{s} \text{ ist Vorfahre von } \boldsymbol{r}, \\ \boldsymbol{s} \le \boldsymbol{r} \quad &\Longleftrightarrow \quad \boldsymbol{s} = \boldsymbol{r} \text{ oder } \boldsymbol{s} \text{ ist Nachfahre von } \boldsymbol{r}. \end{aligned}$$

Zur Erinnerung an die Begriffe „Vorfahre“ und „Nachfahre“ eines Knoten im Mehrdimensionalen siehe Abbildung 3.11 auf Seite 45. Man beachte, dass die Menge $\{\boldsymbol{s} : \boldsymbol{s} > \boldsymbol{r} \text{ oder } \boldsymbol{s} \le \boldsymbol{r}\}$ im Allgemeinen eine echte Teilmenge von G ist.

Die Zerlegung (3.63) lässt sich damit im Mehrdimensionalen analog formulieren. Hier ist für $\boldsymbol{x} \in T_{\boldsymbol{r}}$, $\boldsymbol{r} \in G$

$$u(\boldsymbol{x}) \;=\; \sum_{\boldsymbol{s}>\boldsymbol{r}} u_{\boldsymbol{s}}\phi_{\boldsymbol{s}}(\boldsymbol{x}) \;+\; \sum_{\boldsymbol{s}\leq\boldsymbol{r}} u_{\boldsymbol{s}}\phi_{\boldsymbol{s}}(\boldsymbol{x}). \tag{3.74}$$

Für unsere Zwecke besser geeignet ist eine Zerlegung, bei der nur nach Vor- und Nachfahren entlang der ersten Raumkoordinate getrennt wird:

$$\begin{aligned} u(\boldsymbol{x}) \;=\; u(x_1,x') \;=\; & \sum_{\substack{s_1\in[0,1],\\(s_1,r')\in G,\\ s_1>r_1}} \sum_{\substack{s'\in[0,1]^{d-1},\\(s_1,s')\in G,\\ s'>r' \text{ oder } s'\leq r'}} u_{(s_1,s')}\cdot\phi_{s_1}(x_1)\cdot\phi_{s'}(x') \\ + & \sum_{\substack{s'\in[0,1]^{d-1},\\(r_1,s')\in G,\\ s'>r' \text{ oder } s'\leq r'}} \sum_{\substack{s_1\in[0,1],\\(s_1,s')\in G,\\ s_1\leq r_1}} u_{(s_1,s')}\cdot\phi_{s_1}(x_1)\cdot\phi_{s'}(x') \end{aligned} \tag{3.75}$$

Zur Erläuterung siehe auch Abbildung 3.15. Man vergewissere sich, dass die Reihenfolge der Summation in den Doppelsummen nicht vertauscht werden darf, also insbesondere die Summation über $s_1 > r_1$ außen, die Summation über $s_1 \leq r_1$ innen stehen muss, da sonst nicht alle Knoten berücksichtigt werden, die zu u beitragen.

Setzt man (3.75) in (3.72) ein, erhält man für die erste Doppelsumme den Beitrag

$$\int_{x_1=0}^{1} \sum_{\substack{s_1\in[0,1],\\(s_1,r'),\in G\\ s_1>r_1}} \left\{ \int_{x'\in[0,1]^{d-1}} \sum_{\substack{s'\in[0,1]^{d-1},\\(s_1,s')\in G,\\ s'>r' \text{ oder } s'\leq r'}} u_{(s_1,s')}\cdot\phi_{s'}^{(\alpha')}(x')\cdot\phi_{r'}^{(\beta')}(x')\,dx' \right\} \cdot \phi_{s_1}^{(\alpha_1)}(x_1)\cdot\phi_{r_1}^{(\beta_1)}(x_1)\,dx_1,$$

für die zweite Doppelsumme entsprechend

$$\int_{x'\in[0,1]^{d-1}} \sum_{\substack{s'\in[0,1]^{d-1},\\(r_1,s')\in G,\\ s'>r' \text{ oder } s'\leq r'}} \left\{ \int_{x_1=0}^{1} \sum_{\substack{s_1\in[0,1],\\(s_1,s'),\in G\\ s_1\leq r_1}} u_{(s_1,s')}\cdot\phi_{s_1}^{(\alpha_1)}(x_1)\cdot\phi_{r_1}^{(\beta_1)}(x_1)\,dx_1 \right\} \cdot \phi_{s'}^{(\alpha')}(x')\cdot\phi_{r'}^{(\beta')}(x')\,dx'.$$

Betrachten wir zunächst die Integrale in x_1-Richtung. Diese sind nun von der Art, wie sie mit dem für $d = 1$ beschriebenen Algorithmus ausgewertet werden können. Das heißt insbesondere, dass nur Werte an Knoten r_1 zu berechnen sind, an denen auch Überschüsse (zweite Gleichung) bzw. vorberechnete Werte von Integralen in den anderen $d-1$ Raumrichtungen (erste Gleichung) sitzen. Was die $(d-1)$-dimensionalen Integrale betrifft, so stellt man fest, dass diese die gleiche Form wie das ursprünglich zu berechnende, d-dimensionale Integral haben. Um dies nachzuvollziehen, setze man etwa (3.74) in (3.72) ein.

Damit ist das d-dimensionale Problem zerlegt in ein eindimensionales und ein $(d-1)$-dimensionales Problem. Das eindimensionale Problem kann mit dem oben beschriebenen Algorithmus behandelt werden. Das $(d-1)$-dimensionale Problem hat die gleiche Form, wie das d-dimensionale. Ein Algorithmus, der das d-dimensionale Problem

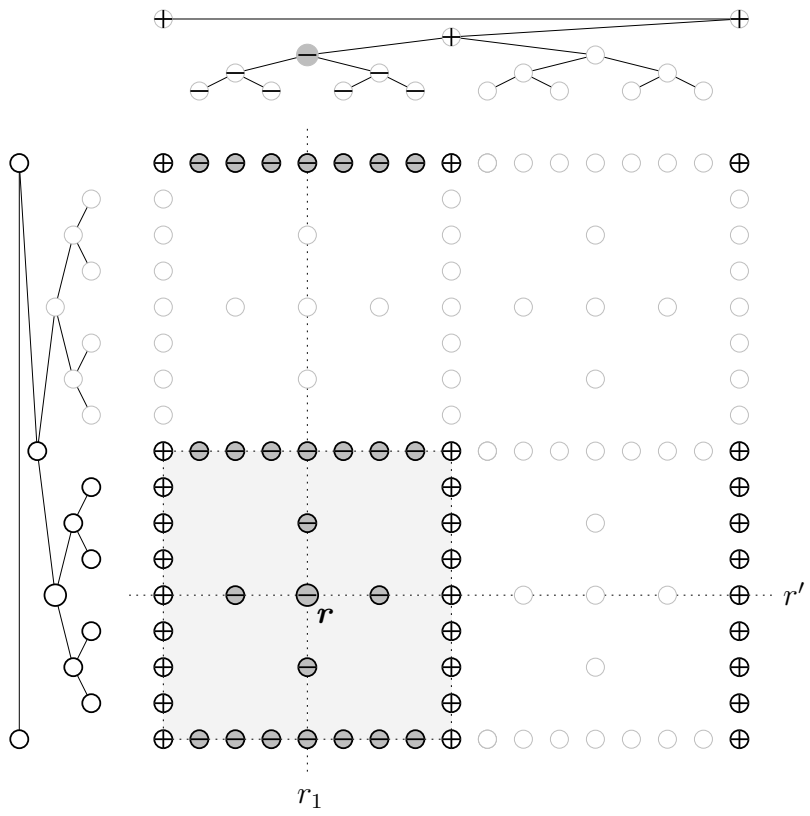


Abbildung 3.15: Zur Zerlegung (3.75): Betrachtet wird ein fester Knoten $\boldsymbol{r} = (r_1, r')$. Im Baum für die x_1-Achse (oben) sind die Knoten $s_1 > r_1$ mit „+“ gekennzeichnet, die Knoten $s_1 \leq r_1$ mit „−“. Im Baum für die x'-Achse sind die Knoten s', für die $s' > r'$ oder $s' \leq r'$ gilt, durch schwarz berandete Kreise ausgewiesen. Alle im Gitter gekennzeichneten Knoten tragen zu u auf dem Träger von $\phi_{\boldsymbol{r}}$ bei. Die erste Doppelsumme in (3.75) läuft über die mit „+“ gekennzeichneten Knoten, die zweite Doppelsumme über die mit „−“ gekennzeichneten.

lösen soll, wird dies also über eine Rekursion in d tun, das Problem also sukzessive auf das $(d-1)-$, $(d-2)$-, usw. -dimensionale Problem zurückführen, bis schließlich $d = 1$ erreicht ist, und der 1D-Algorithmus zum Zuge kommt. In Abbildung 3.16 ist dieser Algorithmus schematisch dargestellt. Das Vorgehen, ein mehrdimensionales Problem auf eindimensionale Problem zurückzuführen, wird als *unidirektionales Prinzip* bezeichnet und ist für das Finite-Elemente-Verfahren mit Dünngitterelementen von grundlegender Bedeutung: Dadurch stehen Algorithmen für die Funktionalauswertung zur Verfügung, die zum einen einfach sind, weil sie im Grunde genommen nur für $d = 1$ formuliert werden müssen, und zum anderen äußerst effizient sind in dem Sinn, dass die Rechenzeit je Gitterpunkt durch eine Konstante beschränkt ist. Insgesamt hat man für die einmalige Auswertung der Integrale an allen Punkten des Gitters einen Rechenaufwand $O(d \cdot N \cdot p^2)$ sowie einen Speicheraufwand $O(d \cdot N)$ für die d Kopien und $O(p \cdot \log_2 N)$ für den Stack.

Algorithmus:	*Unidir*
Eingabe:	Gitter G, $\boldsymbol{u} = (u_{\boldsymbol{r}})_{\boldsymbol{r}\in G}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, aktuelle Integrationsrichtung i
Ausgabe:	Werte $v_{\boldsymbol{r}}$, $\boldsymbol{r} \in G$, in Vektor $\boldsymbol{u}$

Kopiere $\boldsymbol{u}$ nach $\boldsymbol{u}'$
Für alle Gitterlinien in G in Richtung i
  Führe dort *BottomUp* mit $\boldsymbol{u}$ und den Parametern α_i, β_i durch
Falls $i < d$
  Rufe *Unidir* über $\boldsymbol{u}$ mit Integrationsrichtung $i+1$ auf
  Rufe *Unidir* über $\boldsymbol{u}'$ mit Integrationsrichtung $i+1$ auf
Für alle Gitterlinien in G in Richtung i
  Führe dort *TopDown* mit $\boldsymbol{u}'$ und den Parametern α_i, β_i durch
Setze $\boldsymbol{u} := \boldsymbol{u} + \boldsymbol{u}'$

Abbildung 3.16: Der Algorithmus „Unidir“. Vom Hauptprogramm ist er mit Integrationsrichtung $i = 1$ zu starten, die Rekursion über die anderen Raumrichtungen wird vom Algorithmus gesteuert.

3.6.2 Erweiterte Funktionale

Die Algorithmen aus dem vorangegangenen Abschnitt lassen sich auch für Funktionale der Art

$$w_{\boldsymbol{r}} \;=\; \sum_{0\le|\boldsymbol{\alpha}|_1,|\boldsymbol{\beta}|_1\le 1} \int_{[0,1]^d} c_{\boldsymbol{\alpha\beta}}(\boldsymbol{x}) \cdot D^{\boldsymbol{\alpha}} u(\boldsymbol{x}) \cdot D^{\boldsymbol{\beta}} \phi_{\boldsymbol{r}}(\boldsymbol{x})\, d\boldsymbol{x}. \tag{3.76}$$

formulieren, sofern die Funktionen $c_{\boldsymbol{\alpha\beta}}$ Tensorprodukte univariater Funktionen sind, also die Darstellung

$$c_{\boldsymbol{\alpha\beta}}(\boldsymbol{x}) \;=\; c_{\boldsymbol{\alpha\beta},1}(x_1) \cdot \dots \cdot c_{\boldsymbol{\alpha\beta},d}(x_d).$$

besitzen.

Hierzu ist es lediglich nötig, die eindimensionalen Funktionale

$$\langle u, \phi_r \rangle \;:=\; \int_{[0,1]} c_{\boldsymbol{\alpha\beta},k}(x) \cdot D^{\alpha} u(x) \cdot D^{\beta} \phi_r(x)\, dx$$

exakt auszuwerten, vergleiche (3.62). Hierfür werden die elementaren Integrale

$$\boldsymbol{M}_{r,ij} \;=\; \langle f_r^j, f_r^i \rangle \;=\; \int_{[0,1]} c_{\boldsymbol{\alpha\beta},k}(x) \cdot D^{\alpha} f_r^i(x) \cdot D^{\beta} f_r^j(x)\, dx,$$

benötigt, vergleiche (3.70). Sind diese nicht analytisch gegeben, so lassen sie sich numerisch wie folgt approximieren: Man nutzt aus, dass das Produkt $D^{\alpha} f_r^i(x) \cdot D^{\beta} f_r^j(x)$

wiederum ein Polynom ist,

$$D^\alpha f_r^i(x) \cdot D^\beta f_r^j(x) \;=\; \boldsymbol{e}_r^T \boldsymbol{f}_r,$$

wobei die Basis $\boldsymbol{f}_r$ auf der rechten Seite nun Polynome bis zum Grad $2 \cdot p$ enthält. Hat man für jedes r die Werte

$$\boldsymbol{d}_r \;=\; \int_{[0,1]} c_{\boldsymbol{\alpha\beta},k}(x) \cdot \boldsymbol{f}_r \, dx \,, \tag{3.77}$$

bekommt man sofort $\boldsymbol{M}_{r,ij} = \boldsymbol{e}_r^T \boldsymbol{d}_r$. Nun hat das Funktional (3.77) die aus dem letzten Abschnitt bekannte Form (3.62), wobei hier $c_{\boldsymbol{\alpha\beta},k}$ die Rolle von u übernimmt, $\boldsymbol{f}_r$ die von ϕ_r. Die Zahlen $\boldsymbol{d}_r$ lassen sich deshalb wie dort beschrieben durch einen TopDown- und einen BottomUp-Durchlauf berechnen.

Insgesamt hat man also folgendermaßen vorzugehen: In einer Setup-Phase sind einmalig die Elementarmatrizen $\boldsymbol{M}_r$ für jeden Faktor $c_{\boldsymbol{\alpha\beta},k}$, $1 \le k \le d$ zu berechnen. Dies geschieht auf einem eindimensionalen Gitter, das mindestens so fein ist wie die feinste aller Gitterlinien im Gitter G. Die Auswertung der Funktionale 3.76 erfolgt dann mit den unidirektionalen Algorithmus aus dem letzten Abschnitt.

3.6.3 Basistransformation

Der Wechsel von der hierarchischen Basis in die dehierarchische Basis und umgekehrt ist eine der grundlegenden Operationen. So liegen Eingabedaten in der Regel in dehierarchischer, nodaler Form vor und müssen für die bislang vorgestellten Algorithmen *hierarchisiert* werden, also in der hierarchischen Basis dargestellt werden. Umgekehrt sind für die Ausgabe einer Lösungsfunktion nur die Funktionswerte und weniger die hierarchischen Überschüsse an den Knoten von Interesse. Hier muss also *dehierarchisiert* werden.

Betrachten wir zunächst den Fall $d = 1$. Sei r ein Knoten des Gitters. Aus der Zerlegung 3.63 bekommt man für $x = r$ den Zusammenhang

$$u(r) \;=\; \sum_{\substack{s \in G, \\ s \text{ ist Vorfahre} \\ \text{von } r}} u_s \cdot \phi_s(x) \;\; + \; u_r$$

für den Funktionswert $u(r)$ und den hierarchischen Überschuss u_r. Mit Hilfe eines TopDown-Durchlaufs lassen sich damit die hierarchischen Überschüsse in die nodalen Werte bzw. die nodalen Werte in die Überschüsse überführen. Der BottomUp-Durchlauf entfällt wegen der fehlenden Abhängigkeit von den hierarchischen Nachfahren des Knotens.

Für den Fall beliebiger Dimension sind die TopDown-Durchläufe zunächst entlang aller Gitterlinien parallel zur x_1-Achse zu absolvieren, dann entlang aller Gitterlinien parallel zur x_2-Achse usw. Dies entspricht dem Algorithmus *Unidir* (Abbildung 3.16), wobei die BottomUp-Durchläufe hier entfallen und aus diesem Grund der rekursive Algorithmus zu einer einfachen Schleife über die Raumrichtungen entartet.

3.6.4 Ein diskreter Differentialoperator

Die in diesem Kapitel eingeführten Funktionen über dünnen Gittern sind von stückweise polynomialer Gestalt. An den Gitterpunkten sind sie im Allgemeinen nicht differenzierbar. Dort existieren allerdings die linksseitige bzw. rechtsseitige Ableitung. Der folgende diskrete Differentialoperator ist daher für Funktionen u, die über dünnen Gittern gegeben sind, wohldefiniert:

$$(D_i u)(\boldsymbol{x}) \;=\; \begin{cases} \partial_i^{(r)} u(\boldsymbol{x}), & \text{falls } x_i = 0\ , \\ \partial_i^{(l)} u(\boldsymbol{x}), & \text{falls } x_i = 1\ , \\ (\partial_i^{(l)} u(\boldsymbol{x}) + \partial_i^{(r)} u(\boldsymbol{x}))/2, & \text{sonst.} \end{cases}$$

Mit $\partial_i^{(r)}$ und $\partial_i^{(l)}$ sind dabei die rechts- bzw. linksseitige partielle Ableitung in Richtung i bezeichnet. $D_i u$ stimmt an den Punkten, wo u differenzierbar ist, mit der Ableitung $\partial_i u$ überein. An den Gitterpunkten liefert $D_i u$ Approximationen für $\partial_i u^\star$, wenn $u^\star$ eine glatte Funktion ist und u ihr Dünngitterinterpolant.

Um für eine Dünngitterfunktion u an allen Gitterpunkten $\boldsymbol{r}$ den Wert $(D_i u)(\boldsymbol{r})$ zu berechnen, ist wie folgt vorzugehen: Sei u bezüglich der hierarchischen Basis gegeben. Im Fall $d = 1$ hat man analog zu (3.63) die Zerlegung

$$u'(r) \;=\; \overline{u}'_r(r) + \underline{u}'_r(r)\,.$$

In einem TopDown-Durchlauf werden die Werte $\overline{u}'_r(r)$ bestimmt, in einem BottomUp-Durchlauf die Werte $\underline{u}'_r(r)$. Im Falle höherer Dimension ist zuerst entlang aller Gitterlinien parallel zur x_i-Achse der eindimensionale Algorithmus anzuwenden. Anschließend muss entlang der Gitterlinien der verbleibenden Richtungen dehierarchisiert werden. Diese Reihenfolge ist für die Korrektheit des Algorithmus entscheidend: Der BottomUp-Durchlauf beim Differenzieren muss für den richtigen Informationsfluss im Gitter vor den TopDown-Durchläufen des Dehierarchisierens stattfinden.

Der beschriebene Algorithmus liefert lediglich die Werte $(D_i u)(\boldsymbol{r})$ an den Punkten des dünnen Gitters. Es gilt jedoch zu beachten, dass im Allgemeinen $D_i u \neq I_h(D_i u)$ ist. Für glattes $u^\star$ wird jedoch $I_h(D_i I_h(u^\star))$ eine gute Approximation von $\partial_i u^\star$ sein.

3.6.5 Transponierte Operatoren

Manchmal ist es nötig, vom transponierten Operator Gebrauch zu machen. Ist etwa $\boldsymbol{H}$ die Matrix, die den Wechsel von der nodalen in die hierarchische Basis vermittelt, und ist $\boldsymbol{A}$ die Steifigkeitsmatrix bezüglich der hierarchischen Basis, so ist

$$\boldsymbol{A}' \;:=\; \boldsymbol{H}^T \cdot \boldsymbol{A} \cdot \boldsymbol{H}$$

die Steifigkeitsmatrix bezüglich der nodalen Basis. Diese ist für die Formulierung von Mehrgitterverfahren von Bedeutung, da für das nodale System gute Glätter bekannt sind, siehe hierzu auch Kapitel 6. Ist man in der Lage, den Operator $\boldsymbol{H}^T$ durch einen Algorithmus zu realisieren, kann die Multiplikation mit $\boldsymbol{A}'$ entsprechend der obigen Darstellung durch drei Matrix-Vektor-Multiplikationen ersetzt werden.

Die unidirektionalen Algorithmen erlauben unmittelbar die Angabe des Algorithmus für den transponierten Operator: Bei den Baumdurchläufen werden Datenvektoren von der Wurzel zu den Blättern (TopDown) bzw. umgekehrt (BottomUp) transportiert. Diese Vektoren gehen dabei aus den Datenvektoren an den bereits besuchten Knoten und aus den dort gespeicherten Koeffizienten der Eingabe-Funktion durch lineare Operationen hervor. Es handelt sich also um eine sukzessive Multiplikation mit Elementarmatrizen, je eine pro besuchten Knoten. Für die transponierte Operation muss nun – entsprechend der Rechenregeln für Matrizen – lediglich in umgekehrter Reihenfolge mit den transponierten Elementarmatrizen multipliziert werden. Konkret bedeutet dies, dass jeder TopDown-Durchlauf durch einen BottomUp-Durchlauf auszutauschen ist und umgekehrt.

Für das mehrdimensionale unidirektionale Schema (Abbildung 3.16) muss die Regel für das Transponieren eines Produkts von Matrizen erneut beachtet werden: Die Rekursion in den Raumrichtungen ist hier umzukehren. Die Notwendigkeit hierfür ist zunächst durch die besagte Rechenregel gegeben, entspricht aber auch dem Prinzip, dass die TopDown-Durchläufe für den richtigen Datenfluss stets nach den BottomUp-Durchläufen stattzufinden haben. Das heißt, die Umkehr in der Richtungsrekursion ist bereits durch den Austausch von TopDown- und BottomUp-Durchläufen bedingt.

3.7 Datenstruktur

Datenstrukturen für dünne Gitter müssen zum einen so flexibel sein, dass lokal adaptive Gitter gespeichert werden können. Zum anderen müssen sie die Information über die hierarchischen Beziehungen unter den Knoten enthalten. Eine Möglichkeit besteht darin, für jeden Knoten $2d$ Zeiger zu speichern, die auf die beiden Söhne je Raumkoordinate verweisen [5]. Die Konstruktion von Gittern und die Gitterverfeinerung gestalten sich allerdings wegen der Mehrfachverkettung (ein Knoten kann Sohn von mehreren Vätern sein) und der unidirektionalen Verkettung vom Vater zu den Söhnen schwierig. Alternativ bietet sich an, die Koordinate $\boldsymbol{x}$ oder das Multiindex-Paar $(\boldsymbol{\ell}, \boldsymbol{i})$ mittels *Hashtechniken* eindeutig auf den zugehörigen Datenwert abzubilden. Dies hat einige Vorteile: Das Lesen bzw. Schreiben eines Datenwerts zu einem gegebenen Gitterpunkt kann (theoretisch) mit einem $O(1)$-Aufwand geschehen, unabhängig von der Größe des Gitters. Voraussetzung ist allerdings eine effiziente *hash-Funktion*. Die Vater-Sohn-Beziehungen werden über die Koordinaten der Gitterpunkte, also über den hash-Schlüssel, aufgelöst. Dadurch sind insbesondere der Gitteraufbau und die Verfeinerung einfach zu realisieren. Hashtables sind ferner als Bestandteil von Standard-Bibliotheken für viele Programmiersprachen implementiert (z.B. Standard Template Library in C++ [31]). Zum Einsatz von Hash-Techniken im Zusammenhang mit adaptiven Mehrgittermethoden und dünnen Gittern siehe [28].

Die beiden besprochenen Datenstrukturen haben den Nachteil, dass die Daten in der Regel im Speicher verstreut sind und damit eine Cache-orientierte Bearbeitung nicht möglich ist. Ein erster Schritt in Richtung Cache-orientierter Speicherung besteht darin, die Daten in der Reihenfolge abzulegen, wie sie bei einem depth-first

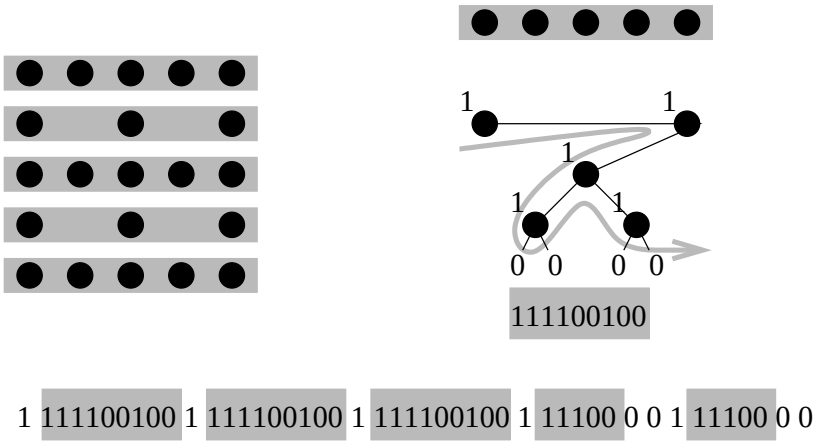


Abbildung 3.17: Zur Speicherung eines Dünngitters mit dimensions-rekursiven linearisierten Binärbäumen.

Durchlauf eines Binärbaums besucht werden. Das Gitter selbst kann dabei mit einem Bit je Knoten kodiert werden (1=Knoten exisitiert, 0=Knoten existiert nicht). Mit Hilfe dimensions-rekursiver linearisierter Binärbäume wird dieses Konzept auf beliebige Dimension ausgeweitet [38]. In Abbildung 3.17 ist dies an einem einfachen Beispiel veranschaulicht. Der Bit-Vektor enthält die vollständige Information über das Gitter. Die grau unterlegten Bits entsprechen den grau unterlegten Gitterlinien im dünnen Gitter.

Allerdings kann man bei dieser Anordnung der Datenwerte nur beim unidirektionalen Durchlauf entlang der am dichtesten gespeicherten Linien von der Datenlokalität profitieren, bezüglich der anderen Richtungen sind die Daten im Speicher verstreut. Um in Abbildung 3.17 die Datenstruktur in y-Richtung zu durchlaufen, müssen die Gitterlinien mühsam aus dem linearisierten Binärbaum extrahiert werden, siehe [38]. In dem Programm, mit dem die in Kapitel 5 vorgestellten numerischen Beispiel gerechnet wurden, wurde daher folgende Alternative implementiert: Durchläufe entlang der linear gespeicherten Linien erfolgen wie gehabt. In den anderen Richtungen wird auf einem Level nicht ein einzelner Gitterpunkt behandelt sondern jeweils alle Gitterpunkte der Hyperebene, die senkrecht zur Durchlaufrichtung und linear, also dicht im Speicher liegt, gleichzeitig. Im Beispiel von Abbildung 3.17 wären dies die Linien parallel zur x-Achse, die bei einem Durchlauf entlang der y-Achse gleichzeitig bearbeitet werden. Dieses Vorgehen hat zwar den Nachteil, dass der Stack sich wesentlich vergrößert, da nun der Datentransport für alle Punkte der Hyperebene gleichzeitig stattfindet. Da sich die Anzahl der Punkte in der Hyperebene mit jedem Abstieg etwa halbiert und die Stackgröße je Gitterpunkt allenfalls linear im Level steigt, bleibt der Speicherbedarf jedoch beschränkt. Interessant ist dieses Vorgehen auch deshalb, weil es die Vektorisierung der elementaren Operationen auf der Hyperebene erlaubt.

4 FEM mit hierarchischen Tensorproduktelementen

Dieses Kapitel erläutert, wie aufbauend auf die hierarchischen Tensorproduktelemente ein effizientes Finite-Elemente-Verfahren definiert werden kann. Zunächst wird auf die Frage eingegangen, was bei der Zusammensetzung des Ansatzraums über einem hinreichend glatt berandeten Gebiet Ω aus Einzelelementen zu beachten ist. Bei näherer Betrachtung des aus der Galerkin-Diskretisierung stammenden linearen Gleichungssystems stellt sich heraus, dass die klassische Trennung in Assemblierung und Lösung des Gleichungssystems die Dünngitterkomplexität zunichte machen würde. Für eine einfache Klasse von Randwertproblemen wird ein effizienter Algorithmus für die Multiplikation der Steifigkeitsmatrix mit einem Vektor angegeben. Um diese Algorithmen für das allgemeine Randwertproblem wiederverwenden zu können, wird ein modifiziertes Galerkin-Verfahren vorgeschlagen. Eine vollständige Konvergenzanalyse mit Hilfe des Lemmas von Strang konnte bisher leider nicht erfolgen. Stattdessen wird eine Beweisskizze vorgestellt und der nicht bewiesene Teil durch numerische Tests untermauert.

4.1 FE-Diskretisierung mit hierarchischen Tensorproduktelementen

Durch die über $K = [0,1]^d$ definierten hierarchischen Tensorproduktelemente ist eine darauf aufbauende Finite-Element-Diskretisierung des Lösungsraums $H_0^1(\Omega)$ bereits weitgehend festgelegt: Das Gebiet $\Omega \subset \mathbb{R}^d$ wird in Rechtecks- (Quader-) Elemente zerlegt,

$$\Omega \;=\; \bigcup_{i=1}^{N_E} K^{(i)}, \qquad K^{(i)} = \psi^{(i)}(K),$$

wobei die Elemente $K^{(i)}$ durch Transformationen $\psi^{(i)}$ aus dem Referenzelement K hervorgehen (vergleiche Abschnitt 2.2.1). Abbildung 4.1 zeigt ein einfaches Beispiel.

Zunächst stehen zwei Möglichkeiten zur Verfügung, die Approximationsgüte des Ansatzraums zu steuern. Die eine besteht darin, den Elementtyp, d.h. den lokalen Ansatzraum, festzulegen und die benötigte Genauigkeit durch eine entsprechend hohe Anzahl N_E von Elementen zu erreichen. Dieses Vorgehen entspricht der h-Version der Finiten Elemente (siehe Abschnitt 2.2.2). Allerdings erweist es sich als wenig sinnvoll,

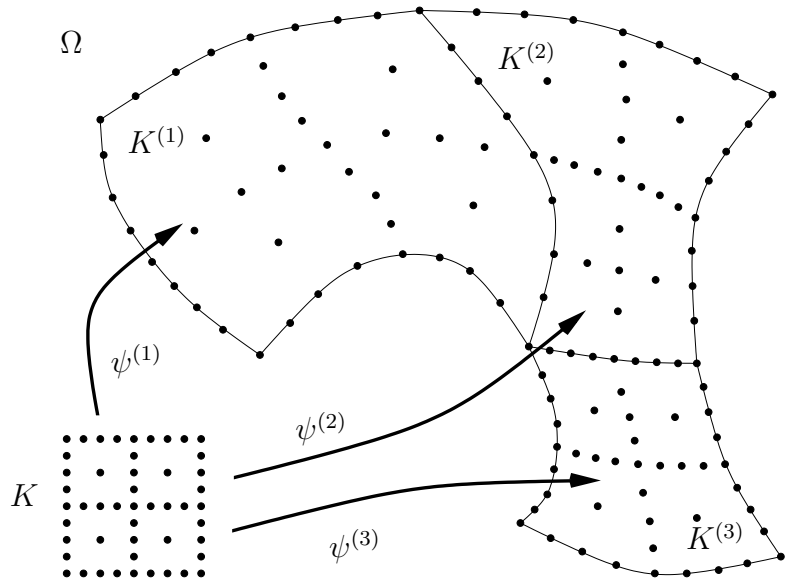


Abbildung 4.1: FE-Diskretisierung mit Dünngitterelementen.

da die Dünngitterkomplexität verloren geht. Schließlich erzeugt man eine Struktur, die einem vollen Gitter ähnelt, d.h. die Anzahl der Freiheitsgrade steigt wie h^{-d}. Die Strategie muss also wie bei der p-Version darin bestehen, ausgehend von einer Zerlegung in wenige Elemente die Approximationsgüte über die Feinheit der lokalen Ansatzräume zu regeln, also die Tiefe der dünnen Gitter über den Elementen hinreichend hoch zu wählen.

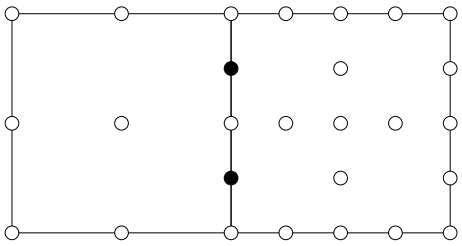
Im Gegensatz zur p-Methode, die erst in Verbindung mit der h-Methode die nötige Flexibilität erhält, um auch singuläre Funktionen durch die lokale Verfeinerung der Zerlegung effizient zu approximieren, ist man bei den hierarchischen Tensorproduktelementen von diesem Zwang entbunden. Die lokale Verfeinerung des Ansatzraums findet hier innerhalb eines Elements statt.

Um mit wenigen Elementen auch krumm berandete Gebiete modellieren zu können, müssen relativ allgemeine Transformationen $\psi^{(i)}$ zugelassen werden. Hier bietet es sich an, die Idee der *isoparametrischen Elemente* aufzugreifen, bei denen die d Komponenten der Funktion $\psi^{(i)}$ durch Funktionen aus dem Ansatzraum des Referenzelements interpoliert werden. Interessant sind in diesem Zusammenhang Transformationen, die aus transfiniter Interpolation hervorgehen. Dabei wird die Funktion $\psi^{(i)}$, die zunächst nur auf dem Rand von K gegeben ist und damit den Rand des Elements $K^{(i)}$ vollständig beschreibt, durch multilineare Interpolation im Innern von K fortgesetzt. Dies ist in hierarchischer Darstellung besonders einfach, da sämtliche hierarchischen Überschüsse im Innern von K null sind. In [22] werden weitere Gittergeneratoren für transformierte Dünngitter diskutiert. Einen allgemeinen Überblick über Gittergeneratoren gibt [33].

Im Hinblick auf eine H^1-konforme Diskretisierung müssen bei der Verfeinerung jedoch zwei Punkte beachtet werden:

- Der (maximale) Polynomgrad p muss für alle Elemente gleich sein, da sonst an der gemeinsamen Grenzlinie (-fläche) unterschiedlich interpoliert werden würde, also im Allgemeinen kein stetiger Übergang gewährleistet ist.

- Die Elemente können unterschiedlich feine Gitter besitzen. Allerdings entstehen dadurch im Allgemeinen *hängende Knoten*:



  Hängende Knoten sind aber in der hierarchischen Darstellung kein wirkliches Problem: Indem der zugeordnete hierarchische Überschuss auf 0 gesetzt wird, ist der C^0-Übergang gesichert. Der Knoten ist damit kein echter Freiheitsgrad mehr.

Damit ist das Finite-Element-Verfahren mit hierarchischen Tensorproduktelementen eindeutig beschrieben. Seine praktische Bedeutung wird aber danach bemessen, mit welchem Aufwand das aus der Diskretisierung der Randwertaufgabe hervorgehende lineare Gleichungssystem gelöst werden kann.

Hier sei kurz wiederholt, was bereits in Abschnitt 2.2.3 über die Struktur der Steifigkeitsmatrix berichtet worden ist: Die Steifigkeitsmatrizen, wie sie bei der h-Methode entstehen, zeichnen sich dadurch aus, dass sie sehr dünn besetzt sind. Die Zahl der nichtverschwindenden Einträge je Zeile ist durch eine Konstante beschränkt unabhängig von der Maschenweite des Gitters. Bei der p-Methode hingegen sind die Teilmatrizen, die zu den einzelnen Elementen gehören, die sogenannten Elementmatrizen, in der Regel voll besetzt. Der Speicheraufwand wächst also quadratisch mit der Zahl der Freiheitsgrade. Die in der Praxis verwendete Knotenzahl je Element ist aus diesem Grund nie höher als 1000. Die p-Methode kann sich nur dadurch gegenüber der h-Methode behaupten, dass sie bei vorgeschriebener Genauigkeit mit essentiell weniger Freiheitsgraden auskommt.

Der Besetztheitsgrad der Systemmatrix bei hierarchischen Tensorproduktelementen ist zwischen den beiden geschilderten Extrema anzusetzen. Die durchschnittliche Anzahl von Einträgen pro Zeile steigt mit der Tiefe des dünnen Gitters an und dies umso schneller, je höher die Raumdimension ist. In Tabelle 4.1 ist dies dokumentiert. Die Anzahl der Nicht-Null-Einträge steigt insbesondere schneller als linear mit der Anzahl der Knoten im Element. Eine Assemblierung und damit explizite Speicherung der Systemmatrix würde also die Dünngitterkomplexität zerstören. Der nächste Abschnitt wird jedoch zeigen, dass die Multiplikation der Systemmatrix mit einem Vektor – zumindest für eine bestimmte Klasse von Differentialoperatoren – mit einem Zeit- und Speicheraufwand bewerkstelligt werden kann, der linear in der Anzahl der Freiheitsgrade skaliert. Die (effiziente) Matrix-Vektor-Multiplikation ist für die Anwendung iterativer Gleichungslöser ausreichend.

	1D			2D			3D		
n	#K	#NNE	ρ	#K	#NNE	ρ	#K	#NNE	ρ
1	1	1	100.0	1	1	100.0	1	1	100.0
2	3	7	77.8	5	21	84.0	7	43	39.7
3	7	27	55.1	17	185	64.0	31	667	26.1
4	15	83	36.9	49	1145	47.7	111	6571	20.3
5	31	227	23.6	129	5897	35.4	351	50107	16.8
6	63	579	14.6	321	27337	26.5	1023	325147	14.1
7	127	1411	8.7	769	119049	20.1			
8	255	3331	5.1	1793	499465	15.5			
9	511	7683	2.9						
10	1023	17411	1.7						

Tabelle 4.1: Anzahl der Nichtnulleinträge (NNE) der Element-Steifigkeitsmatrix für L_2-basierte Dünngitterelemente der Tiefe n. Hier sind nur die Einträge zu Knoten im Innern der Elemente berücksichtigt, in der Spalte #K ist die Anzahl dieser Knoten angegeben. ρ ist der Anteil der Nichtnulleinträge in der Matrix (Angabe in Prozent).

4.2 Berechnung des Matrix-Vektor-Produkts Au

Dieser Abschnitt zeigt, wie die Multiplikation eines Vektors mit der Steifigkeitsmatrix mit konstantem Rechenaufwand je Freiheitsgrad geschehen kann, sofern das Randwertproblem gewisse Voraussetzungen erfüllt.

Sei also ein Randwertproblem in seiner schwachen Formulierung

$$\mathcal{A}(u, v) \;=\; l(v),$$

gegeben mit den in (2.5) definierten Formen $\mathcal{A}$ und l. Sei ferner V_h ein Ansatzraum, der aus einer FE-Diskretisierung über der Zerlegung $\{K^{(i)}\}$ hervorgeht. Ferner sei $\{\phi_k\}$ eine Basis von V_h und u eine beliebige Funktion in V_h gegeben durch

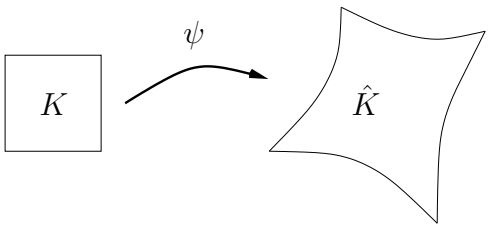
$$u \;=\; \sum u_k \cdot \phi_k.$$

Die Multiplikation des Koeffizientenvektors $\boldsymbol{u} = (u_k)$ mit der Systemmatrix ergibt einen Vektor $\boldsymbol{w} = (w_j)$, dessen Einträge durch

$$w_j \;=\; \mathcal{A}(u, \phi_j)$$

gegeben sind. Analog zum elementweisen Vorgehen bei der Assemblierung der Steifigkeitsmatrix ist es bei der Berechnung von $\boldsymbol{w}$ sinnvoll, zunächst mit $\boldsymbol{w} = \mathbf{0}$ zu starten und in einer Schleife über die Elemente $K^{(i)}$ die von dort stammenden Beiträge in $\boldsymbol{w}$ sukzessive aufzuaddieren. Im Folgenden wird dargestellt, wie diese Beiträge berechnet werden.

Sei $\hat{K}$ ein beliebig herausgegriffenes Element der Zerlegung, $K = [0, 1]^d$ das ihm zugrunde liegende Referenzelement und $\psi : K \to \hat{K}$ die Transformation, vermöge der $\hat{K}$ aus K hervorgeht:



Bei dem Referenzelement K handele es sich um ein hierarchisches Tensorproduktelement. Sei G das Gitter über K. Wir wollen im Folgenden die hierarchischen Basisfunktionen über K durch die Knoten $\boldsymbol{r} \in G$ identifizieren, das heißt $\phi_{\boldsymbol{r}} = \phi_{\boldsymbol{\ell},\boldsymbol{i}}$, falls $\boldsymbol{r} = \boldsymbol{x}_{\boldsymbol{\ell},\boldsymbol{i}}$. Die Basisfunktionen über $\hat{K}$ sind dann gegeben durch

$$\hat{\phi}_{\boldsymbol{r}}(\hat{\boldsymbol{x}}) \;=\; \phi_{\boldsymbol{r}}(\boldsymbol{x}), \qquad \hat{\boldsymbol{x}} = \psi(\boldsymbol{x}),$$

Sei nun $\hat{u}$ eine Funktion aus dem lokalen Ansatzraum über $\hat{K}$ mit der Darstellung

$$\hat{u} \;=\; \sum_{\boldsymbol{r}\in G} u_{\boldsymbol{r}} \hat{\phi}_{\boldsymbol{r}}.$$

Der Beitrag von $\hat{u}$ zum Vektor $\boldsymbol{w}$ ergibt sich dann zu

$$\begin{aligned} w_{\boldsymbol{r}} &= \mathcal{A}(\hat{u}, \hat{\phi}_{\boldsymbol{r}}) = \\ &= \int_{\hat{K}} \left\{ \sum_{i,j=1}^{d} a_{ij}\, \partial_i \hat{u}\, \partial_j \hat{\phi}_{\boldsymbol{r}} \;+\; \sum_{i=1}^{d} b_i\, \partial_i \hat{u}\, \hat{\phi}_{\boldsymbol{r}} \;+\; c\hat{u}\hat{\phi}_{\boldsymbol{r}} \right\} d\hat{\boldsymbol{x}} = \\ &= \int_{K} \left\{ \sum_{i,j=1}^{d} \bar{a}_{ij}\, \partial_i u\, \partial_j \phi_{\boldsymbol{r}} \;+\; \sum_{i=1}^{d} \bar{b}_i\, \partial_i u\, \phi_{\boldsymbol{r}} \;+\; \bar{c} u \phi_{\boldsymbol{r}} \right\} d\boldsymbol{x} \end{aligned} \tag{4.1}$$

mit

$$\begin{aligned} u(\boldsymbol{x}) &:= \hat{u}(\hat{\boldsymbol{x}}) \;=\; \sum_{\boldsymbol{r}\in G} u_{\boldsymbol{r}} \phi_{\boldsymbol{r}}, \\ \bar{a}_{ij}(\boldsymbol{x}) &:= \sum_{1\le k,l\le d} a_{kl}(\hat{\boldsymbol{x}}) \cdot \frac{\partial x_k}{\partial \hat{x}_i} \cdot \frac{\partial x_l}{\partial \hat{x}_j} \cdot |\det J_\psi|, \\ \bar{b}_i(\boldsymbol{x}) &:= \sum_{1\le k\le d} b_k(\hat{\boldsymbol{x}}) \cdot \frac{\partial x_k}{\partial \hat{x}_i} \cdot |\det J_\psi|, \\ \bar{c}(\boldsymbol{x}) &:= c(\hat{\boldsymbol{x}}) \cdot |\det J_\psi|, \qquad\qquad J_\psi := \left(\frac{\partial \psi_i}{\partial x_j} \right)_{ij}. \end{aligned} \tag{4.2}$$

Das auf das Referenzelement transformierte Funktional hat also die gleiche Form wie das ursprüngliche Funktional. Die über $\hat{K}$ definierten Koeffizienten a_{ij}, b_i und c transformieren sich zunächst wie Tensoren zweiter, erster bzw. nullter Stufe durch Multiplikation mit dem metrischen Tensor $\partial x_k / \partial \hat{x}_i$, dessen Einträge in der Praxis aus der Jacobimatrix $J_{\psi^{-1}} = J_\psi^{-1}$ gewonnen werden. Die Determinante der Jacobimatrix J_ψ ist für die Äquivalenz der Integrale $\int_K d\boldsymbol{x}$ und $\int_{\hat{K}} d\hat{\boldsymbol{x}}$ als zusätzlicher Faktor hinzuzunehmen.

4.2.1 Affin-lineare Transformationen und Differentialoperatoren mit konstanten Koeffizienten

Hängen die Koeffizienten a_{ij}, b_i und c nicht vom Ort ab und sind die Transformationen $\psi^{(i)} : K \to K^{(i)}$ linear-affin, das heißt

$$\psi^{(i)}(\boldsymbol{x}) \;=\; \boldsymbol{B}^{(i)}\boldsymbol{x} + \boldsymbol{c}^{(i)}, \qquad \boldsymbol{B}^{(i)} \in \mathbb{R}^{d\times d},\ \boldsymbol{c}^{(i)} \in \mathbb{R}^d,$$

so sind $J_{\psi^{(i)}} = \boldsymbol{B}^{(i)}$ und damit auch die transformierten Koeffizienten $\bar{a}_{ij}$, $\bar{b}_i$ und $\bar{c}$ konstant. Das Integral (4.1) hat also die Form

$$w_{\boldsymbol{r}} \;=\; \sum_{0\le|\boldsymbol{\alpha}|_1,|\boldsymbol{\beta}|_1\le 1} c_{\boldsymbol{\alpha}\boldsymbol{\beta}} \cdot \int_{[0,1]^d} D^{\boldsymbol{\alpha}}u(\boldsymbol{x})\, D^{\boldsymbol{\beta}}\phi_{\boldsymbol{r}}(\boldsymbol{x})\, d\boldsymbol{x}$$

mit Konstanten $c_{\boldsymbol{\alpha}\boldsymbol{\beta}}$.

Für eine effiziente Auswertung der Integrale bietet sich der in Abschnitt 3.6.1 beschriebene unidirektionale Algorithmus an. Der Rechenaufwand für eine Matrix-Vektor-Multiplikation ist nach den dort beschriebenen Eigenschaften des Algorithmus $O(d \cdot N \cdot p^2)$, wenn N die Anzahl der Gitterpunkte ist und p die Ordnung des Ansatzraums.

4.2.2 Erweiterung für Koeffizienten mit Tensorprodukteigenschaft

Sind die Koeffizienten zwar nicht linear, dafür aber – wie die Basisfunktionen $\phi_{\boldsymbol{r}}$ – Tensorprodukte aus univariaten Funktionen, so hat (4.1) die Form

$$w_{\boldsymbol{r}} \;=\; \sum_{0\le|\boldsymbol{\alpha}|_1,|\boldsymbol{\beta}|_1\le 1} \int_{[0,1]^d} c_{\boldsymbol{\alpha}\boldsymbol{\beta}}(\boldsymbol{x}) \cdot D^{\boldsymbol{\alpha}}u(\boldsymbol{x}) \cdot D^{\boldsymbol{\beta}}\phi_{\boldsymbol{r}}(\boldsymbol{x})\, d\boldsymbol{x}.$$

mit

$$c_{\boldsymbol{\alpha}\boldsymbol{\beta}}(\boldsymbol{x}) \;=\; c_{\boldsymbol{\alpha}\boldsymbol{\beta},1}(x_1) \cdot \dots \cdot c_{\boldsymbol{\alpha}\boldsymbol{\beta},d}(x_d).$$

Diese Integrale lassen sich mit Hilfe der erweiterten Algorithmen aus Abschnitt 3.6.2 effizient auswerten.

Der Fall von Koeffizienten mit Tensorproduktstruktur ist allerdings sehr speziell und in der Praxis kaum von Bedeutung. Interessant in diesem Zusammenhang ist, inwiefern glatte Funktionen durch eine endliche Summe von Tensorprodukten so genau approximiert werden können, dass der Abbruchfehler von der gleichen Größenordnung ist, wie der durch die dünnen Gitter verursachte Diskretisierungsfehler. Die Anzahl der dazu benötigten Tensorprodukte darf allerdings nicht zu sehr steigen für $h \to 0$. Die triviale Entwicklung der $c_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ in der hierarchischen Tensorprodukt-Basis würde genauso viel Summanden wie Gitterpunkte erfordern, also zu einer Komplexität $O(N^2)$ führen.

Die Algorithmen wurden hier weniger in der Hoffnung auf ein allgemeines Verfahren formuliert, sondern vielmehr für den Zweck, das im Folgenden vorgestellte, mit einer modifizierten Bilinearform arbeitende Verfahren für eine Reihe von Testfällen mit dem exakten Verfahren vergleichen zu können, siehe die Beispiele in den Abschnitten 5.2.1 und 5.2.2.

4.2.3 Variable Koeffizienten und krumm berandete Elemente

Wenden wir uns nun dem allgemeinen Fall zu, dass variable Koeffizienten vorliegen und die Elemente in der Zerlegung von Ω krumm berandet sind, also durch nichttriviale Transformationen aus dem Referenzelement $K = [0,1]^d$ hervorgehen. Hier gilt es, die Funktionale

$$w_{\boldsymbol{r}} \;=\; \sum_{0 \le |\boldsymbol{\alpha}|_1, |\boldsymbol{\beta}|_1 \le 1} \int_{[0,1]^d} c_{\boldsymbol{\alpha}\boldsymbol{\beta}}(\boldsymbol{x}) \cdot D^{\boldsymbol{\alpha}} u(\boldsymbol{x}) \cdot D^{\boldsymbol{\beta}} \phi_{\boldsymbol{r}}(\boldsymbol{x})\, d\boldsymbol{x}. \tag{4.3}$$

auszuwerten, wobei die $c_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ beliebige (glatte) Funktionen über K sind. Ziel muss es wiederum sein, für ein gegebenes u alle $w_{\boldsymbol{r}}$, $\boldsymbol{r} \in G$, zu berechnen, und dabei mit einem Rechen- und Speicheraufwand von $O(N)$ auszukommen, wenn N die Anzahl der Knoten $\boldsymbol{r} \in G$ ist.

Zunächst einmal müssen die Funktionen $c_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ für eine numerische Behandlung in irgendeiner Form diskretisiert werden. Bei herkömmlichen Finite-Element-Verfahren geschieht dies etwa dadurch, dass die Integrale (4.3) mit Hilfe von Quadraturformeln näherungsweise berechnet werden. Für die $c_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ bedeutet das, dass sie an den Knoten der Quadraturformel und auch nur dort ausgewertet werden. Die Quadraturknoten sind dabei für alle Basisfunktion $\phi_{\boldsymbol{r}}(\boldsymbol{x})$ eines Elements die gleichen, was angesichts der Tatsache, dass bei klassischen Finiten Elementen alle Basisfunktionen denselben Träger – nämlich das Element – besitzen, sinnvoll ist. Damit müssen auch die $c_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ nur einmal ausgewertet werden. Für die Dünngitterelemente mit ihren hierarchischen Basisfunktionen und deren stark variierenden Trägern ist dieses Vorgehen wenig Erfolg versprechend.

Pflaum [37] schlägt vor, die Koeffizienten durch ihre Interpolanten im Ansatzraum V_h zu diskretisieren. Damit wird zum einen erreicht, dass die Dünngitterkomplexität, zumindest was den Speicheraufwand betrifft, erhalten bleibt. Zum anderen ist die Konvergenzanalyse einfach: Für glatte Koeffizienten geht der Interpolationsfehler mit der bekannten Asymptotik in der Zahl der Gitterpunkte gegen null. Dies sichert im vorliegenden Fall Konsistenz und Stabilität der diskreten Operatoren und damit die Konvergenz des Verfahrens. Das große Problem bei dieser Diskretisierung ist, dass bis heute die Frage nach einem effizienten Algorithmus für die Matrix-Vektor-Multiplikation offen ist. Zwar erfüllt für $d = 1$ das in Abschnitt 4.2.2 erläuterte Verfahren seinen Zweck – hier besitzen die univariaten Koeffizienten trivialerweise die Tensorprodukteigenschaft –, und auch für $d = 2$ lässt sich bei konsequenter Fortsetzung der beim eindimensionalen Fall verwendeten Ideen ein effizienter Algorithmus auf Basis des unidirektionalen Prinzips angeben. Doch für $d \ge 3$ blieben bislang alle Anstrengungen ohne Erfolg.

Modifikation der Bilinearform

Die im Folgenden beschriebene Diskretisierung nimmt sich zum Ziel, die bereits vorhandenen unidirektionalen Algorithmen wiederzuverwenden. Hierzu seien die auf das

Referenzelement transformierten Koeffizienten $\bar{a}_{ij}$ und $\bar{b}_i$ in Richtung i partiell differenzierbar. Dann ist

$$\begin{aligned} \bar{a}_{ij} \cdot \partial_i u &= \partial_i(\bar{a}_{ij} \cdot u) - \partial_i \bar{a}_{ij} \cdot u, \\ \bar{b}_i \cdot \partial_i u &= \partial_i(\bar{b}_i \cdot u) - \partial_i \bar{b}_i \cdot u. \end{aligned} \tag{4.4}$$

Für die über dem Referenzelement gegebene Bilinearform

$$\begin{aligned} \mathcal{A}(u,v) &= \int_K \left\{ \sum_{i,j=1}^{d} \bar{a}_{ij}\, \partial_i u\, \partial_j v + \sum_{i=1}^{d} \bar{b}_i\, \partial_i u\, v + \bar{c} u v \right\} d\boldsymbol{x} = \\ &= \int_K \left\{ \sum_{i,j=1}^{d} \big(\partial_i(\bar{a}_{ij} u) - \partial_i \bar{a}_{ij} \cdot u\big) \cdot \partial_j v \right. \\ &\qquad \left. + \sum_{i=1}^{d} \big(\partial_i(\bar{b}_i u) - \partial_i \bar{b}_i \cdot u\big) \cdot v + \bar{c} u \cdot v \right\} d\boldsymbol{x} \end{aligned} \tag{4.5}$$

betrachten wir nun die folgende gitterabhängige Diskretisierung:

$$\begin{aligned} \mathcal{A}_h(u,v) := \int_K \Bigg\{ & \sum_{i,j=1}^{d} \big(\partial_i I_h(\bar{a}_{ij} u) - I_h(\partial_i \bar{a}_{ij} \cdot u)\big) \cdot \partial_j v \\ & + \sum_{i=1}^{d} \big(\partial_i I_h(\bar{b}_i u) - I_h(\partial_i \bar{b}_i \cdot u)\big) \cdot v \\ & + I_h(\bar{c} u) \cdot v \Bigg\} d\boldsymbol{x} \end{aligned} \tag{4.6}$$

Dabei ist I_h die Interpolation im Ansatzraum V_h. Die Produkte $(\bar{a}_{ij} u)$, $(\partial_i \bar{a}_{ij} \cdot u)$, usw. werden im Vergleich zur exakten Bilinearform durch ihre Interpolanten über dem Elementgitter ersetzt werden. An dieser Stelle gilt es zu bemerken, dass die modifizierte Bilinearform $\mathcal{A}_h$ für den Fall konstanter Koeffizienten $\bar{a}_{ij}$, $\bar{b}_i$ und $\bar{c}$ mit $\mathcal{A}$ übereinstimmt.

Effiziente Auswertung der modifizierten Bilinearform

Bevor wir diskutieren, inwiefern $\mathcal{A}_h$ eine geeignete Approximation von $\mathcal{A}$ ist, wollen wir einen Blick auf die Auswertung der Integrale $\mathcal{A}_h(u, \phi_{\boldsymbol{r}})$ werfen. Die Berechnung erfolgt dabei in zwei Schritten. Zuerst sind die Interpolanten $I_h(\bar{a}_{ij} u)$, $I_h(\partial_i a_{ij} \cdot u)$, etc. zu bilden. Das algorithmische Vorgehen hierbei ist einfach: u wird zunächst in die nodale Darstellung überführt, so dass an den Gitterpunkten nicht die hierarchischen Überschüsse sondern die Funktionswerte vorliegen. Danach wird an allen Knoten mit den Funktionswerten von $\bar{a}_{ij}$, $\partial_i \bar{a}_{ij}$, etc. multipliziert und das Ergebnis wieder in die hierarchische Darstellung gebracht. Die Algorithmen für den Basiswechsel sind in Abschnitt 3.6.3 beschrieben.

Der zweite Schritt befasst sich sodann mit der eigentlichen Auswertung der Integrale. Diese haben die Form

$$w_{\boldsymbol{r}} \;=\; \mathcal{A}_h(u, \phi_{\boldsymbol{r}}) \;=\; \sum_{0 \le |\boldsymbol{\alpha}|_1, |\boldsymbol{\beta}|_1 \le 1} \int_{[0,1]^d} D^{\boldsymbol{\alpha}} z_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{x}) \cdot D^{\boldsymbol{\beta}} \phi_{\boldsymbol{r}}(\boldsymbol{x})\, d\boldsymbol{x}.$$

Die $z_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ sind dabei die eben berechneten Interpolanten $I_h(\dots)$, also Funktionen aus dem Raum V_h. Damit liegen die gleichen Voraussetzungen vor, wie wir sie in Abschnitt 4.2.1 hatten. Die Werte $w_{\boldsymbol{r}}$ können unmittelbar mit Hilfe von unidirektionalen Algorithmen berechnet werden. Der Rechenaufwand sowohl für die Interpolation als auch für die Integration steigt jeweils nicht stärker als linear in der Anzahl der Freiheitsgrade. Die Komplexität betreffend haben wir damit das Ziel für eine effiziente Multiplikation mit der Steifigkeitsmatrix erreicht.

Diskrete Ableitungsoperatoren für die Koeffizienten

Um die Bilinearform $\mathcal{A}_h$ in der Form (4.6) verwenden zu können, müssen die Koeffizienten sowie einige ihrer partiellen Ableitungen an den Gitterpunkten vorliegen. Diese Ableitungen sind in der Regel nicht explizit gegeben. Hier stellt sich zunächst die Frage, weshalb der Umweg über die Identitäten (4.4) gemacht worden ist, anstatt die Produkte $(\bar{a}_{ij}\partial_i u)$ und $(\bar{b}_i \partial_i u)$ direkt zu interpolieren. Der Grund liegt darin, dass die Ableitung $\partial_i u$ wegen der stückweisen polynomialen Darstellung von u gerade an den Gitterpunkten nicht eindeutig bestimmt sind und die Interpolation damit nicht wohldefiniert ist. Als sinnvolle Werte für die Ableitung bieten sich zwar die Mittelwerte aus links- und rechtsseitiger Ableitung an. Doch hat dies zur Folge, dass der resultierende Operator einen nicht trivialen Kern besitzt: Für die stückweise lineare Funktion, die an den Knoten abwechselnd die Werte 0 und 1 annimmt, verschwinden die Mittelwerte der einseitigen Ableitungen.

Dornseifer [22] beschreibt eine Diskretisierung für stückweise lineare Ansatzfunktionen, die das eben geschilderte Problem dadurch umgeht, dass $\partial_i u$ mit Hilfe einer Basis aus stückweise konstanten Funktionen dargestellt wird, siehe Abbildung 4.2. Die Multiplikation von $\partial_i u$ mit den Koeffizienten erfolgt dann an den Mittelpunkten der Intervalle, über denen $\partial_i u$ konstant ist. Das Produkt wird wieder als stückweise konstante Funktion dargestellt. Die Integrale lassen sich dann mit einem unidirektionalen Algorithmus auswerten. Folgende Eigenschaften sind für den Erfolg dieser Methode entscheidend: Erstens wird $\partial_i u$ mit Hilfe der stückweise konstanten Basisfunktionen exakt dargestellt. Zweitens gibt es für diese Basis sowohl eine hierarchische als auch eine nodale Version, wobei sich hierarchische und dehierarchische Darstellung einer Funktion ähnlich einfach ineinander überführen lassen, wie dies bei stückweise linearen Funktionen der Fall ist. Für Polynomgrad $p \ge 2$ ist die Angabe einer Basis für den Raum $\{\partial_i u : u \in V_h\}$ mit den genannten Eigenschaften nicht möglich. Um nun die Probleme bei der Behandlung von $\partial_i u$ zu umgehen, wird die Ableitung von u mit Hilfe der Produktregel (4.4) auf den Koeffizienten abgewälzt.

Bleibt zu klären, wie die Ableitungen der Koeffizienten bestimmt werden können. Hier bietet sich an, die Koeffizienten über dem Gitter zu interpolieren und die

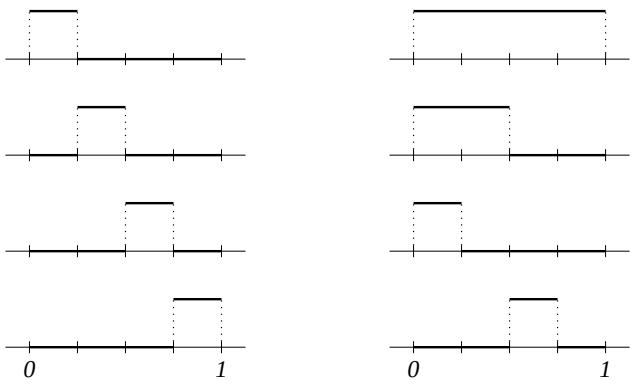


Abbildung 4.2: Nodale (links) und hierarchische (rechts) Basis für stückweise konstante Funktionen, wie sie bei der Differentiation von stückweise linearen Funktionen entstehen.

gewünschten Ableitungen durch die Ableitungen des Interpolanten zu approximieren. Wegen der stückweise polynomialen Gestalt sind die Ableitungen an den Gitterpunkten nicht definiert. Allerdings existieren die einseitige Ableitungen. Wir definieren daher den für stückweise polynomiale Funktionen geeigneten Differentialoperator D_i durch

$$D_i f(\boldsymbol{x}) \;=\; \begin{cases} \partial_i^{(r)} f(\boldsymbol{x}), & \text{falls } x_i = 0\ , \\ \partial_i^{(l)} f(\boldsymbol{x}), & \text{falls } x_i = 1\ , \\ (\partial_i^{(l)} f(\boldsymbol{x}) + \partial_i^{(r)} f(\boldsymbol{x}))/2, & \text{sonst.} \end{cases} \tag{4.7}$$

Mit $\partial_i^{(r)}$ und $\partial_i^{(l)}$ sind dabei die rechts- bzw. linksseitige Ableitung gemeint. Man beachte, dass für die Auswertung der Ableitungen an den Gitterpunkten ein Algorithmus mit $O(N)$-Komplexität existiert. Er ist in Abschnitt 3.6.4 beschrieben.

In der nach (4.6) definierten Bilinearform werden nun folgende Terme ersetzt:

$$\begin{array}{lcl} \partial_i \bar{a}_{ij} & \rightarrow & D_i I_h^{(\hat{p})}(\bar{a}_{ij}), \\ \partial_i \bar{b}_i & \rightarrow & D_i I_h^{(\hat{p})}(\bar{b}_i). \end{array} \tag{4.8}$$

Dabei ist $I_h^{(\hat{p})}$ die Dünngitter-Interpolation mit Polynomen vom Grad $\hat{p}$, wobei $\hat{p}$ nicht notwendigerweise mit der Ordnung p des Ansatzraums, über dem die Bilinearform definiert ist, übereinstimmt. Hinweise für eine sinnvolle Wahl von $\hat{p}$ werden weiter unten gegeben.

Der Übersichtlichkeit halber sei die nach (4.6) modifizierte Bilinearform zusammen

mit den Ersetzungen (4.8) noch einmal angegeben:

$$\begin{aligned}\mathcal{A}_h(u,v) \;=\; \int_K \Bigg\{ & \sum_{i,j=1}^{d} \Big(\partial_i I_h(\bar{a}_{ij} u) \;-\; I_h\big(\, D_i I_h^{(\hat{p})}(\bar{a}_{ij}) \,\cdot\, u \,\big) \Big) \cdot \partial_j v \\ & + \sum_{i=1}^{d} \Big(\partial_i I_h(\bar{b}_i u) \;-\; I_h\big(\, D_i I_h^{(\hat{p})}(\bar{b}_i) \,\cdot\, u \,\big) \Big) \cdot v \\ & + \; I_h(\bar{c} u) \cdot v \Bigg\}\, d\boldsymbol{x}\end{aligned} \tag{4.9}$$

Krumm berandete Elemente

Bislang sind wir davon ausgegangen, dass die auf das Referenzelement transformierten Koeffizienten $\bar{a}_{ij}$, $\bar{b}_i$ und $\bar{c}$ an den Punkten des Gitters vorliegen. Diese Werte sind aus den in physikalischen Koordinaten gegebenen Koeffizienten a_{ij}, b_i und c und der Transformation ψ gemäß (4.2) zu berechnen. Hierzu werden die partiellen Ableitungen von ψ benötigt, die in der Praxis selten explizit gegeben sind. Für eine numerische Approximation gehen wir folgendermaßen vor: Die Komponenten der vektorwertigen Funktion ψ werden gemäß der Idee isoparametrischer Elemente durch Funktionen aus dem Ansatzraum über dem Referenzelement dargestellt bzw. interpoliert [13, 47]. Die Einträge der Jacobimatrix werden durch die partiellen Ableitungen des Interpolanten approximiert:

$$J_{\psi,ij} \;=\; \frac{\partial \psi_i}{\partial x_j} \;\approx\; D_j I_h^{(\tilde{p})}(\psi_i) \;=:\; J_{h,\psi,ij} \tag{4.10}$$

Dabei ist D_j der durch (4.7) definierte Differentialoperator. Die Interpolation der Transformationen ψ_i findet hier mit Polynomen vom Grad $\tilde{p}$ statt, wobei $\tilde{p}$ in Abhängigkeit von der Ordnung p des Ansatzraums gewählt werden kann. Hinweise, wie dies zu geschehen hat, finden sich weiter unten im Abschnitt *Ordnung bei den Interpolationen*.

Die Werte der zu $J_{h,\psi}$ gehörenden, diskreten Jacobideterminante an den Knoten des Gitters erhält man nun unmittelbar aus den dort gegebenen Einträgen von $J_{h,\psi}$. Für die Koeffizienten des metrischen Tensors haben wir

$$\frac{\partial x_k}{\partial \hat{x}_i} \;=\; J_{\psi^{-1},ki} \;=\; (J_\psi^{-1})_{ki} \;\approx\; (J_{h,\psi}^{-1})_{ki} \;=:\; \left[\frac{\partial x_k}{\partial \hat{x}_i}\right]_h .$$

Hier ist für jeden Gitterpunkt die $(d \times d)$-Matrix $J_{h,\psi}$ zu invertieren. Schließlich werden (an den Gitterpunkten) die benötigten Produkte gebildet. In (4.2) werden damit

folgende Ersetzungen vorgenommen:

$$\begin{aligned}
\bar{a}_{ij} &= \sum_{1\le k,l\le d} a_{kl}\cdot\frac{\partial x_k}{\partial \hat{x}_i}\cdot\frac{\partial x_l}{\partial \hat{x}_j}\cdot|\det J_\psi| \quad\to\quad \sum_{1\le k,l\le d} a_{kl}\cdot\left[\frac{\partial x_k}{\partial \hat{x}_i}\right]_h\cdot\left[\frac{\partial x_l}{\partial \hat{x}_j}\right]_h\cdot|\det J_{h,\psi}|\ ,\\
\bar{b}_i &= \sum_{1\le k\le d} b_k\cdot\frac{\partial x_k}{\partial \hat{x}_i}\cdot|\det J_\psi| \quad\to\quad \sum_{1\le k\le d} b_k\cdot\left[\frac{\partial x_k}{\partial \hat{x}_i}\right]_h\cdot|\det J_{h,\psi}|\ ,\\
\bar{c} &= c\cdot|\det J_\psi| \quad\to\quad c\cdot|\det J_{h,\psi}|\ .
\end{aligned} \tag{4.11}$$

Ordnung bei den Interpolationen

Im Zuge der vorgestellten Diskretisierung werden an drei Stellen Interpolationen durchgeführt. In der Reihenfolge ihrer Auswertung betrifft dies die Transformation ψ in (4.10), die (transformierten) Koeffizienten $\bar{a}_{ij}$, $\bar{b}_i$ und $\bar{c}$ in (4.8) und zuletzt die Produkte der (transformierten) Koeffizienten mit der Ansatzfunktion u in (4.6). Die ersten beiden Interpolationen waren nötig, um an den Gitterpunkten Approximationen für die partiellen Ableitungen zu gewinnen. Da bei der Differentiation jeweils eine Ordnung an Genauigkeit verloren geht, ist es sinnvoll, in Abhängigkeit von der Ordnung p des Ansatzraums, die Interpolationen in (4.8) bzw. (4.10) mit Polynomen vom Grad

$$\begin{aligned}
\hat{p} &:= p+1 \qquad \text{bzw.}\\
\tilde{p} &:= p+2
\end{aligned} \tag{4.12}$$

durchzuführen. An Hand der numerischen Beispiele in den Abschnitten 5.2.1 und 5.3.1 wird hierauf näher eingegangen.

Man beachte, dass diese Ordnungserhöhung bei der Interpolation der Koeffizienten und Transformationen nicht zu einer Ordnungserhöhung für das Verfahren führen. Dieses Vorgehen hat etwa den gleichen Stellenwert wie die Wahl einer Quadraturformel höherer Ordnung für die Berechnung der Steifigkeitsmatrix bei herkömmlichen Finite-Element-Verfahren. Ferner ist die Berechnung der Interpolanten bzw. ihrer partiellen Ableitungen in (4.10) und (4.8) für ein gegebenes Gitter einmalig in einer Setup-Phase auszuführen.

Zusammenfassung

Bevor wir uns der Konvergenzanalyse zuwenden, sollen noch einmal die Punkte wiederholt werden, die bei der Formulierung der diskreten Bilinearform $\mathcal{A}_h$ eine wichtige Rolle gespielt haben. Das Hauptziel war ein Algorithmus für die Multiplikation mit der Steifigkeitsmatrix, dessen Rechenaufwand linear in der Problemgröße skaliert. Da solche Algorithmen für den Fall, dass der Differentialoperator nur konstante Koeffizienten enthält, existieren, lag die Idee nahe, diese Algorithmen nach Möglichkeit wiederzuverwenden. Die Voraussetzung hierfür wurde dadurch geschaffen, dass Produkte von

Koeffizienten und Ansatzfunktion durch Interpolation in eine einzelne Dünngitter-Funktion verwandelt wurden und die resultierenden Integrale nunmehr die gleiche Form hatten, wie sie sie bei konstanten Koeffizienten haben.

An dieser Stelle gilt es noch zu bemerken, dass die Diskretisierung in ihrer Formulierung allgemein ist, das heißt, sie kann nicht nur für Dünngitterelemente verwendet werden, sondern für alle bekannten Finite-Element- bzw. Galerkin-Verfahren. Dies könnte überall dort von Interesse sein, wo die Steifigkeitsmatrix etwa für den Laplace-Operator eine sehr einfache Struktur hat, aber bei variablen Koeffizienten beliebig kompliziert wird. Dies trifft zum Beispiel bei Galerkin-Spektralverfahren und den Elementen höherer Ordnung nach Szabó und Babuška zu (vergleiche Abschnitt 2.3).

4.3 Konvergenzbetrachtungen

Das Lemma von Céa besagt, dass die Differenz von schwacher Lösung $u \in H^1(\Omega)$ und Galerkin-Approximation $u_h \in V_h \subset H^1(\Omega)$ der Abschätzung

$$\|u - u_h\|_E \;\leq\; C \cdot \inf_{v_h \in V_h} \|u - v_h\|_E$$

genügt (vergleiche Abschnitt 2.1.2). Eine obere Schranke für das Infimum erhält man, indem man $v_h := I_h(u)$ setzt,

$$\|u - u_h\|_E \;\leq\; C \cdot \|u - I_h(u)\|_E \,.$$

Unter zusätzlichen Annahmen über die Regularität von u erhält man dann konkrete Abschätzungen für den Interpolationsfehler.

Bei Finite-Element-Verfahren benutzt man die Zerlegung des Interpolationsfehlers in eine Summe über die Beiträge von den einzelnen Elementen. Für $e_h := u - I_h(u)$ ist

$$\|e_h\|_E^2 \;=\; \sum_{i=1}^{N_E} \left\|e_h|_{K^{(i)}}\right\|_{K^{(i)},E}^2 .$$

Eine Abschätzung für die einzelnen Summanden erhält man in der Regel durch Transformation auf das Referenzelement und Ausnutzung von Approximationseigenschaften des dort gegebenen Ansatzraums. Sei hierzu K das Referenzelement, $\hat{K}$ eines der Elemente $K^{(i)}$ und ψ die Transformation, durch die $\hat{K}$ aus K hervorgeht. Dann ist

$$\begin{aligned} \left\|e_h|_{\hat{K}}\right\|_{\hat{K},E}^2 \;&=\; \int_{\hat{K}} \nabla_{\hat{\boldsymbol{x}}} e_h^T(\hat{\boldsymbol{x}}) \nabla_{\hat{\boldsymbol{x}}} e_h(\hat{\boldsymbol{x}})\, d\hat{\boldsymbol{x}} \\ &=\; \int_{K} (J_{\psi^{-1}} \nabla_{\boldsymbol{x}} e_h(\boldsymbol{x}))^T (J_{\psi^{-1}} \nabla_{\boldsymbol{x}} e_h(\boldsymbol{x})) \cdot |\det J_\psi(\boldsymbol{x})|\, d\boldsymbol{x} \,. \end{aligned} \tag{4.13}$$

Da bei Verwendung von hierarchischen Tensorproduktelementen die Feinheit des Ansatzraums über die Tiefe der Elementgitter gesteuert wird, die anfangs gewählte Zerlegung $\{K^{(i)}\}$ aber nicht verändert wird, sind die von ψ abhängigen Größen in (4.13)

konstant bezüglich des Diskretisierungsparameters h. Ist die Abbildung ψ nun so beschaffen, dass für alle $\boldsymbol{x} \in K$

$$\begin{aligned} \|J_{\psi^{-1}}(\boldsymbol{x})\| &\leq C_1\,, \\ |\det J_\psi(\boldsymbol{x})| &\leq C_2 \end{aligned} \tag{4.14}$$

mit positiven Konstanten C_1 und C_2 gilt, bekommt man sofort eine Aussage der Art

$$\|e_h|_{\hat{K}}\|_{\hat{K},E} \;\leq\; C \cdot \|e_h \circ \psi\|_{K,E}\;.$$

Das heißt, der Fehler e_h auf dem Element $\hat{K}$ lässt sich direkt durch den entsprechenden Fehler $e_h \circ \psi = u \circ \psi - I_h(u \circ \psi)$ auf dem Referenzelement abschätzen. Für die Interpolation über dem Referenzelement liegen nun a-priori Aussagen über das Fehlerverhalten vor, sofern $u \circ \psi$ hinreichend glatt ist (vergleiche Kapitel 3). Diese übertragen sich nach der eben vorgeführten Ableitung unverändert auf den Gesamtfehler. Man darf also erwarten:

$$\|u - u_h\|_E = O(N^{-p} \cdot |\log_2 N|^{p\cdot(d-1)}) \qquad \text{bzw.} \tag{4.15}$$

$$\|u - u_h\|_E = O(N^{-p}). \tag{4.16}$$

Hier ist N die Anzahl der Freiheitsgrade und p der maximale Polynomgrad im Ansatzraum. (4.15) gilt, wenn die Elemente mit L_2-basierten dünnen Gittern diskretisiert werden, (4.16) bei energiebasierten Gittern.

Die auf das Céa-Lemma gestützten Aussagen setzen allerdings voraus, dass es sich um ein „echtes“ Galerkin-Verfahren handelt, also die Bilinearform für die diskreten Gleichungen die gleiche ist wie für das ursprüngliche Randwertproblem. Die vorangegangenen Abschnitte haben jedoch gezeigt, dass diese Forderung im Allgemeinen nicht aufrecht erhalten werden kann, wenn ein effizientes Verfahren für nichttriviale Probleme gewünscht wird. Deshalb wurde vorgeschlagen, statt der exakten Bilinearform $\mathcal{A}$ in (4.5) die modifizierten, gitterabhängigen Bilinearformen $\mathcal{A}_h$ in (4.6) zu benutzen. Die Ansätze, bei denen die exakte Bilinearform gegen eine (leichter handhabbare) modifizierte Bilinearform ausgetauscht wird, werden in der angelsächsischen Literatur unter dem Stichwort „variational crimes“ gehandelt: So greifen im Prinzip alle Finite-Element-Verfahren für Differentialgleichungen mit variablen Koeffizienten auf Quadraturformeln zurück, wenn es die Integrale der Bilinearform auszuwerten gilt. Wichtigste Stütze für Konvergenzaussagen ist hier das erste Lemma von Strang [13, 47]:

Satz 4.1 (Erstes Lemma von Strang) *Sei $u \in V$ die Lösung von*

$$\mathcal{A}(u, v) \;=\; l(v) \quad \textit{für alle } v \in V.$$

Sei $(V_h)_h$ eine Familie von Ansatzräumen $V_h \subset V$. $\mathcal{A}_h : V_h \times V_h \to \mathbb{R}$ sowie $l_h : V_h \to \mathbb{R}$ seien darüber definierte (Bi-)linearformen, wobei mit einer von h unabhängigen Zahl $\alpha > 0$ gelte

$$\mathcal{A}_h(v, v) \;\geq\; \alpha \cdot \|v\|_E^2. \tag{4.17}$$

Dann gibt es für jedes h genau ein $u_h \in V_h$, so dass

$$\mathcal{A}_h(u_h, v) \;=\; l(v) \quad \textit{für alle } v \in V_h\,,$$

und es gilt

$$\|u - u_h\|_E \;\leq\; C \cdot \inf_{v \in V_h} \left\{ \|u - v\|_E \;+\; \sup_{w \in V_h} \frac{|\mathcal{A}(v,w) - \mathcal{A}_h(v,w)|}{\|w\|_E} \right\} \tag{4.18}$$

Für die Konvergenz $\|u - u_h\|_E \to 0$ genügt es demnach, dass die Bilinearformen $\mathcal{A}_h$ zwei Bedingungen erfüllen. Erstens ist die Elliptizität mit einer von h unabhängigen Elliptizitätskonstante α vorauszusetzen. Man spricht in diesem Zusammenhang auch von *gleichmäßiger Elliptizität*. Die zweite Bedingung betrifft die Konsistenz des gestörten Problems. Für sie ist zu fordern, dass die Differenz von $\mathcal{A}$ und $\mathcal{A}_h$ für $h \to 0$ in bestimmter Weise gegen null konvergiert. Besteht der Unterschied von exakter und gestörter Bilinearform etwa darin, dass $\mathcal{A}_h$ zur Berechnung der Integrale eine Quadraturformel benutzt, so reicht es, dass diese Formel positive Gewichte besitzt und für alle Polynome $\partial_i u \cdot \partial_j v$, $u, v \in V_h$, exakt ist, um mit dem Lemma von Strang für das inexakte Verfahren die gleiche Konvergenzordnung in h nachzuweisen, wie sie für das exakte Galerkin-Verfahren zu erwarten wäre.

An dieser Stelle sei vorweggenommen, dass eine vollständige Analyse für die in Abschnitt 4.2.3 vorgestellte Bilinearform $\mathcal{A}_h$ bislang noch nicht erreicht worden ist. Im Folgenden wird für eine vereinfachte Version der Bilinearform $\mathcal{A}_h$ die Konsistenz gezeigt. Im Anschluss wird unter der Annahme, dass bestimmte Operatoren über dünnen Gittern gleichmäßig stetig sind, die gleichmäßige Elliptizität nachgewiesen. Die Annahme wird durch numerische Experimente gestützt.

Zur Konsistenz der modifizierten Bilinearform

Wir betrachten über $\Omega = [0,1]^d$ die Randwertaufgabe, die durch die Bilinearform

$$\mathcal{A}(u, v) \;=\; \int_\Omega \sum_{1 \leq i,j \leq d} a_{ij}(\boldsymbol{x})\, \partial_i u(\boldsymbol{x})\, \partial_j v(\boldsymbol{x})\, d\boldsymbol{x} \tag{4.19}$$

und homogene Dirichletbedingungen auf dem Rand von Ω gegeben sei. Die rechte Seite wird nicht näher spezifiziert. Die a_{ij} seien partiell differenzierbar. Als Ansatzräume V_h werden die über dünnen Gittern definierten Räume stückweise polynomialer Funktionen, die auf dem Rand verschwinden, herangezogen. $\mathcal{A}$ wird nun durch die Bilinearform

$$\mathcal{A}_h(u, v) \;=\; \int_\Omega \sum_{1 \leq i,j \leq d} \Big\{ \partial_i I_h(a_{ij} u)(\boldsymbol{x}) \;-\; I_h(\partial_i a_{ij} \cdot u)(\boldsymbol{x}) \Big\} \cdot \partial_j v(\boldsymbol{x})\, d\boldsymbol{x} \tag{4.20}$$

approximiert. Diese entspricht der diskreten Bilinearform (4.6). Die Gebietstransformation braucht wegen der speziellen Wahl von Ω nicht berücksichtigt zu werden, Ω bzw. $H_0^1(\Omega)$ wird durch das Referenzelement diskretisiert. Unter diesen Grundvoraussetzungen lässt sich der folgende Satz beweisen:

Satz 4.2 (Konsistenz von $\mathcal{A}_h$) *Ist u die Lösung des kontinuierlichen Problems und gilt*

$$\begin{aligned} \|u - I_h(u)\|_E &\leq \rho(h), & \|a_{ij}u - I_h(a_{ij}u)\|_E &\leq \rho(h), \\ \|u - I_h(u)\|_{L_2} &\leq \rho(h), & \|\partial_i a_{ij} \cdot u - I_h(\partial_i a_{ij} \cdot u)\|_{L_2} &\leq \rho(h) \end{aligned}$$

mit einer nur von h abhängenden, positiven Zahl $\rho(h)$, so ist mit $v := I_h(u)$

$$\frac{|\mathcal{A}(v,w) - \mathcal{A}_h(v,w)|}{\|w\|_E} \leq C \cdot \rho(h) \tag{4.21}$$

für alle $w \in V_h$. Dabei ist C eine von h unabhängige, positive Zahl.

Beweis: Es gilt den Betrag der Differenz

$$\begin{aligned} \mathcal{A}(I_h(u),w) - \mathcal{A}_h(I_h(u),w) = \int_\Omega \sum_{1\leq i,j\leq d} \Big\{ & \partial_i(a_{ij}I_h(u)) - \partial_i a_{ij} \cdot I_h(u) \\ & - \partial_i I_h(a_{ij}I_h(u)) + I_h(\partial_i a_{ij} \cdot I_h(u)) \Big\} \cdot \partial_j w \, d\boldsymbol{x} \end{aligned}$$

abzuschätzen. Da $I_h(u)$ mit u an den Gitterpunkten übereinstimmt, ist

$$\begin{aligned} I_h(a_{ij}I_h(u)) &= I_h(a_{ij}u), \\ I_h(\partial_i a_{ij} \cdot I_h(u)) &= I_h(\partial_i a_{ij} \cdot u). \end{aligned}$$

Mit Hilfe der Dreiecksungleichung für Beträge und der Cauchy-Schwarz-Ungleichung für das L_2-Skalarprodukt bekommen wir

$$\begin{aligned} |\mathcal{A}(I_h(u),w) - \mathcal{A}_h(I_h(u),w)| \leq \sum_{i,j} \big\| & \partial_i(a_{ij}I_h(u)) - \partial_i a_{ij} \cdot I_h(u) \\ & - \partial_i I_h(a_{ij}u) + I_h(\partial_i a_{ij} \cdot u) \big\|_{L_2} \cdot \big\|\partial_j w\big\|_{L_2} \end{aligned}$$

Die Abschätzung lässt sich weiterführen, indem man die triviale Ungleichung $\|\partial_j w\|_{L_2} \leq \|w\|_E$ und die Dreiecksungleichung für die L_2- und für die Energienorm benutzt:

$$\begin{aligned} \ldots &\leq \sum_{i,j} \Big\{ \big\|\partial_i(a_{ij}I_h(u)) - \partial_i I_h(a_{ij}u)\big\|_{L_2} \\ &\qquad + \big\|\partial_i a_{ij} \cdot I_h(u) - I_h(\partial_i a_{ij} \cdot u)\big\|_{L_2} \Big\} \cdot \big\|w\big\|_E \\ &\leq \sum_{i,j} \Big\{ \big\|a_{ij}I_h(u) - I_h(a_{ij}u)\big\|_E + \big\|\partial_i a_{ij} \cdot I_h(u) - I_h(\partial_i a_{ij} \cdot u)\big\|_{L_2} \Big\} \cdot \big\|w\big\|_E \\ &\leq \sum_{i,j} \Big\{ \big\|a_{ij}I_h(u) - a_{ij}u\big\|_E + \big\|a_{ij}u - I_h(a_{ij}u)\big\|_E \\ &\qquad + \big\|\partial_i a_{ij} \cdot I_h(u) - \partial_i a_{ij} \cdot u\big\|_{L_2} + \big\|\partial_i a_{ij} \cdot u - I_h(\partial_i a_{ij} \cdot u)\big\|_{L_2} \Big\} \cdot \big\|w\big\|_E \end{aligned}$$

Die im Folgenden benutzten Konstanten werden der Einfachheit halber alle mit C bezeichnet, wenngleich der Wert jeweils ein anderer sein kann. Mit $C(\dots)$ werden Abhängigkeiten der Konstante angedeutet. Für die Normen in der Klammer erhalten wir unter den im Satz genannten Voraussetzungen die folgenden Abschätzungen:

$$\|a_{ij} I_h(u) - a_{ij} u\|_E \quad \leq \; C(a_{ij}) \cdot \|I_h(u) - u\|_E \; \leq \; C \cdot \rho(h), \tag{4.22}$$

$$\|a_{ij} u \; - \; I_h(a_{ij} u)\|_E \quad \leq \; \rho(h), \tag{4.23}$$

$$\|\partial_i a_{ij} \cdot I_h(u) \; - \; \partial_i a_{ij} \cdot u\|_{L_2} \quad \leq C(\partial_i a_{ij}) \cdot \|I_h(u) - u\|_{L_2} \; \leq \; C \cdot \rho(h), \tag{4.24}$$

$$\|\partial_i a_{ij} \cdot u \; - \; I_h(\partial_i a_{ij} \cdot u)\|_{L_2} \quad \leq \; \rho(h). \tag{4.25}$$

Bei der ersten und der dritten Abschätzung wurde die Stetigkeit der Multiplikationsoperatoren über $H_0^1(\Omega)$ und $L_2(\Omega)$ ausgenutzt. Insgesamt haben wir damit also

$$|\mathcal{A}(I_h(u), w) - \mathcal{A}_h(I_h(u), w)| \quad \leq \; C \cdot \rho(h) \cdot \|w\|_E,$$

was zu beweisen war. □

Die Abschätzung (4.21) ergibt zusammen mit dem Lemma von Strang die Aussage

$$\|u - u_h\|_E \; \leq \; C \cdot \rho(h)$$

für die Lösung u_h des gestörten Problems, wobei $\rho(h)$ im Wesentlichen eine obere Schranke für den Fehler bei der Interpolation der Funktionen u, $a_{ij}u$ und $\partial_i a_{ij}$ ist. Falls also u und die a_{ij} hinreichend glatt sind und die gleichmäßige Elliptizität von $\mathcal{A}_h$ nachgewiesen werden kann, besitzt das Verfahren mit der modifizierten Bilinearform das gleiche Approximationsverhalten wie das (exakte) Galerkin-Verfahren.

Zur gleichmäßigen Elliptizität von $\mathcal{A}_h$

Um gleichmäßige Elliptizität nachzuweisen, wird oft die Elliptizität der exakten Bilinearform und eine erweiterte Konsistenzabschätzung für die gestörte Bilinearform herangezogen. Ist etwa

$$\sup_{v \in V_h} \frac{|\mathcal{A}(v, v) - \mathcal{A}_h(v, v)|}{\|v\|_E^2} \; \leq \; \sigma(h) \tag{4.26}$$

mit $\sigma(h) \to 0$ für $h \to 0$, hat man sofort

$$\begin{aligned} \mathcal{A}_h(v, v) \; &\geq \; \mathcal{A}(v, v) \; - \; |\mathcal{A}(v, v) - \mathcal{A}_h(v, v)| \\ &\geq \; \gamma \cdot \|v\|_E^2 \; - \; \sigma(h) \cdot \|v\|_E^2 \\ &\geq \; \frac{\gamma}{2} \cdot \|v\|_E^2 \, , \end{aligned}$$

für h klein genug. Hier ist γ die Elliptizitätskonstante der exakten Bilinearform. Eine Aussage der Art (4.26) für die Bilinearform (4.20) erhält man, wenn es im Beweis von

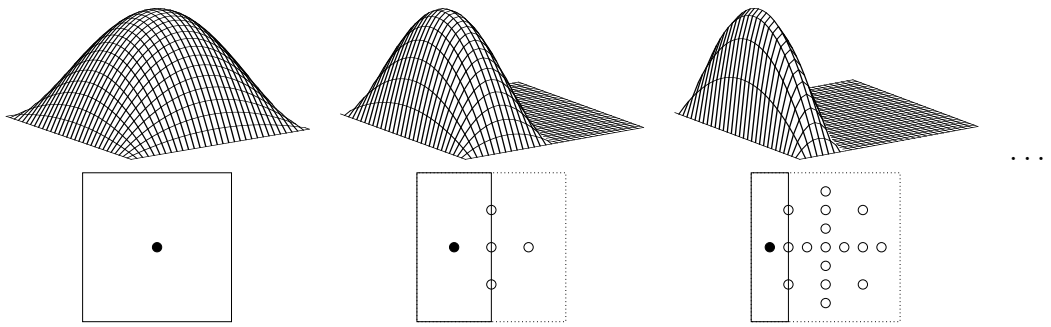


Abbildung 4.3: Die Basisfunktionen $\phi_{\boldsymbol{\ell}_n,\boldsymbol{i}_n} \in V_{0,n}^{(2,1)}$ für $n \in \{1,2,3\}$. Es handelt sich dabei um Basisfunktionen mit dem größtmöglichen Seitenverhältnis des Trägers im jeweiligen Funktionenraum. Sie sind Beispiel für eine Folge von Funktionen, für die der Quotient (4.29) nicht gegen null konvergiert.

Satz 4.2 gelingt, die Teilabschätzungen (4.22)–(4.25) für beliebiges $v \in V_h$ an Stelle von $I_h(u)$ zu wiederholen derart, dass auf der rechten Seite der Faktor $\|v\|_E$ auftritt. Für die Ungleichungen (4.22) und (4.24) ist dabei nichts zu zeigen, da $v = I_h(v)$ gilt und die Normen damit verschwinden. Für die gleichmäßige Stetigkeit der $\mathcal{A}_h$ bliebe also zu zeigen, dass

$$\|a_{ij}v - I_h(a_{ij}v)\|_E \leq \sigma(h) \cdot \|v\|_E, \tag{4.27}$$

$$\|\partial_i a_{ij} \cdot v - I_h(\partial_i a_{ij} \cdot v)\|_{L_2} \leq \sigma(h) \cdot \|v\|_E \tag{4.28}$$

für alle $v \in V_h$ gilt mit $\sigma(h) \to 0$ für $h \to 0$. Setzt man (4.27) und (4.28) in die Beweisführung zu Satz 4.2 ein, erhält man unmittelbar Aussage (4.26).

Für Funktionenräume über dünnen Gittern trifft allerdings bereits die Abschätzung (4.27) nicht zu. Für ein Gegenbeispiel genügt es eine Folge von Funktionen v_n, $n = 1, 2, \ldots$, anzugeben, die aus schrittweise verfeinerten Ansatzräumen V_n stammen und für die die Folge der Quotienten

$$\frac{\|av_n - I_h(av_n)\|_E}{\|v_n\|_E}, \qquad n = 1, 2, \ldots \tag{4.29}$$

nicht gegen null konvergiert. Betrachten wir beispielsweise die Folge der Räume $V_n := V_{0,n}^{(2,1)}$ stückweise quadratischer Funktionen über L_2-basierten dünnen Gittern, die auf dem Rand verschwinden. Für die Funktion $v_n \in V_n$ wählen wir die Basisfunktion $\phi_{\boldsymbol{\ell}_n,\boldsymbol{i}_n}$ mit

$$\boldsymbol{\ell}_n \;=\; (n, 1, \ldots, 1), \qquad \boldsymbol{i}_n \;=\; \mathbf{1}.$$

Sie ist dadurch gekennzeichnet, dass ihr Träger in Richtung x_1 die Länge $2 \cdot 2^{-n}$ besitzt, in den anderen Richtungen die Länge 1. Es handelt sich damit um eine der hierarchischen Basisfunktionen von V_n, für die das Seitenverhältnis des Trägers maximal wird (siehe Abbildung 4.3). Aus der Tatsache, dass im Innern des Trägers von v_n keine wei-

teren Punkte des dünnen Gitters liegen, folgert man rasch, dass für die Interpolation des Produkts av_n die einfache Beziehung

$$I_h(av_n) \;=\; a(x_{\boldsymbol{\ell}_n,\boldsymbol{i}_n}) \cdot v_n$$

gilt. Das heißt, dass von den Funktionswerten $a(\boldsymbol{x})$ über dem Träger von v_n bei der Interpolation nur der Wert am Mittelpunkt des Trägers eingeht. Die folgende Tabelle zeigt, wie sich dieser Informationsverlust bereits für die einfache Funktion a, gegeben durch

$$a(\boldsymbol{x}) \;=\; 1 + \sum_{i=1}^{d} x_i,$$

auswirkt. Sie gibt die Werte für den Quotienten (4.29) in Abhängigkeit von der Dimension d und der Gittertiefe n an.

n	$d = 2$	$d = 3$
1	0.185714	0.221429
2	0.090357	0.137500
3	0.051392	0.092708
4	0.039789	0.077181
5	0.036743	0.072897
6	0.035972	0.071798
7	0.035779	0.071521
8	0.035730	0.071452
9	0.035718	0.071434
10	0.035715	0.071430

Offensichtlich konvergiert die Folge der Quotienten nicht gegen null. Damit haben wir ein einfaches Beispiel gefunden, das die Forderung (4.27) verletzt. Man überzeugt sich leicht, dass dieses Ergebnis nicht auf der speziellen Wahl der Funktionenräume $V_{0,n}^{(2,1)}$ oder der Funktion a beruht, sondern durch die Gestalt des dünnen Gitters und der darüber definierten Basisfunktionen verursacht wird.

Der folgende Satz benutzt einen anderen Ansatz, um die gleichmäßige Elliptizität der Bilinearformen $\mathcal{A}_h$, wie sie in (4.20) definiert sind, zu zeigen:

Satz 4.3 (Gleichmäßige Elliptizität von $\mathcal{A}_h$) *Existiert eine Zerlegung der Koeffizienten a_{ij} in einen konstanten Anteil k_{ij} und einen variablen Anteil g_{ij}, also*

$$a_{ij}(\boldsymbol{x}) \;=\; k_{ij} + g_{ij}(\boldsymbol{x}), \tag{4.30}$$

und gibt es positive, nicht von h abhängende Zahlen B, C, D mit

$$\|I_h(g_{ij}v)\|_E \;<\; B \cdot \|v\|_E \quad \textit{für alle } v \in V_h, \tag{4.31}$$

$$\|I_h(\partial_j g_{ij} \cdot v)\|_{L_2} \;<\; C \cdot \|v\|_E \quad \textit{für alle } v \in V_h, \tag{4.32}$$

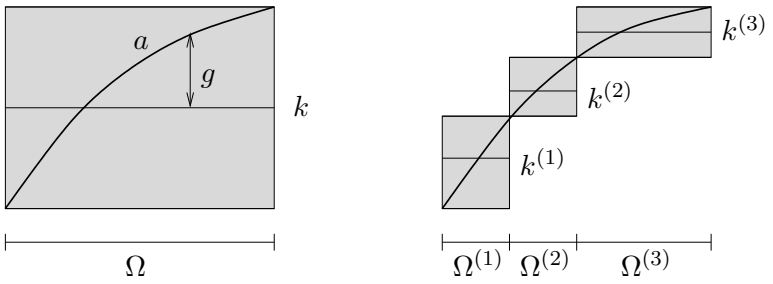
$$(B + C)d^2 \;<\; D \;<\; \lambda_{\min}, \tag{4.33}$$

so sind die Bilinearformen $\mathcal{A}_h$ gleichmäßig elliptisch. $\lambda_{\min}$ ist der kleinste Eigenwert der Matrix $K = (k_{ij})$.

Beweis: Mit Hilfe von Umformungen und Abschätzungen, wie sie bereits im Beweis von Satz 4.2 benutzt wurden, zeigt man

$$\begin{aligned}
\mathcal{A}_h(v,v) &= \sum_{i,j=1}^{d} \int k_{ij}\, \partial_i v\, \partial_j v\, dx \;+\; \sum_{i,j=1}^{d} \int \partial_i v\, \{\partial_j I_h(g_{ij} v) - I_h(\partial_j g_{ij} \cdot v)\}\, dx \\
&\geq \lambda_{\min} \cdot \|v\|_E^2 \;-\; \sum_{i,j=1}^{d} \|\partial_i v\|_{L_2} \cdot \|\partial_j I_h(g_{ij} v) \;-\; I_h(\partial_j g_{ij} \cdot v)\|_{L_2} \\
&\geq \lambda_{\min} \cdot \|v\|_E^2 \;-\; \sum_{i,j=1}^{d} \|v\|_E \cdot \Big(\|I_h(g_{ij} v)\|_E \;+\; \|I_h(\partial_j g_{ij} \cdot v)\|_{L_2} \Big) \\
&\geq (\lambda_{\min} - D) \cdot \|v\|_E^2 \,. \qquad \square
\end{aligned}$$

Nun ist nicht ohne weiteres klar, ob für gegebene Funktionen a_{ij} die Voraussetzungen von Satz 4.3 erfüllt sind. Für den Nachweis der gleichmäßigen Elliptizität wird angenommen, dass von a_{ij} ein konstanter Anteil k_{ij} abgespalten werden kann, der für sich genommen eine so große Elliptizitätskonstante (hier $\lambda_{\min}$) stiftet, dass die als klein angenommene Abweichung g_{ij} nicht mehr zu einem Verlust der Elliptizität führen kann. Dies ist im Allgemeinen nicht der Fall, lässt sich aber einfach dadurch erreichen, dass man das Gebiet Ω in mehrere Elemente unterteilt und die Zerlegung (4.30) individuell für jedes Element vornimmt: Ist die Unterteilung von Ω hinreichend fein und sind die a_{ij} entsprechend glatt, so folgt, dass die g_{ij} klein gegenüber k_{ij} gewählt werden können, siehe folgende Skizze:



Zur gleichmäßigen Stetigkeit des Operators $v \mapsto I_h(gv)$

Um Satz 4.3 anwenden zu können, muss noch sicher gestellt werden, dass die Abschätzungen (4.31) und (4.32) mit gitterunabhängigen Konstanten möglich sind. Sie besagen im wesentlichen, dass die Abbildungen

$$\begin{aligned}
(V_h, \|\cdot\|_E) &\to (V_h, \|\cdot\|_E), \qquad v \mapsto I_h(g_{ij} v)\,, \\
(V_h, \|\cdot\|_E) &\to (V_h, \|\cdot\|_{L_2}), \qquad v \mapsto I_h(g_{ij} v)\,,
\end{aligned}$$

gleichmäßig stetig sind, die Operatornorm also nicht von h abhängt. Es genügt dabei, die erste der beiden Abschätzungen zu zeigen, da mit der Poincaré-Friedrichs-Ungleichung die Beziehung

$$\|I_h(\partial_j g_{ij} \cdot v)\|_{L_2} \;\leq\; c \cdot \|I_h(\partial_j g_{ij} \cdot v)\|_E$$

gilt. Hier ist c eine Konstante, die nur von Ω abhängt.

Handelt es sich bei den V_h um Vollgitterräume oder Standard-Finite-Element-Räume, so kann ein Nachweis der gleichmäßigen Stetigkeit über lokale Abschätzungen geschehen: Seien die Zellen des Vollgitters bzw. der Finite-Element-Triangulierung mit K_i, $i = 1, \ldots, N$, bezeichnet. Dann lässt sich für die Abschätzung von $\|I_h(gu)\|_E^2$ gegen $\|u\|_E^2$ folgendes Schema angeben:

$$\begin{aligned}
\big\|I_h(gu)\big\|_E^2 \;&=\; \sum_{i=1}^{N} \big\|I_h(gu)|_{K_i}\big\|_E^2 \\
&\leq\; \sum_{i=1}^{N} \Big\{ C(\operatorname{grad} g|_{K_i}) \cdot \big\|u|_{K_i}\big\|_{L_2}^2 \;+\; D(g|_{K_i}) \cdot \big\|u|_{K_i}\big\|_E^2 \Big\} \\
&\leq\; \max_i C(\operatorname{grad} g|_{K_i}) \cdot \big\|u\big\|_{L_2}^2 \;+\; \max_i D(g|_{K_i}) \cdot \big\|u\big\|_E^2 \\
&\leq\; C(\operatorname{grad} g,\, g) \cdot \big\|u\big\|_E^2
\end{aligned}$$

Die Konstanten $C(\ldots)$ und $D(\ldots)$ hängen dabei nur von g bzw. von seinen Ableitungen ab, nicht aber von h. Natürlich ist diese Abschätzung erst mit der konkreten Angabe der Konstanten von Wert. Es sollte an dieser Stelle aber nur das Beweisprinzip dargestellt werden.

Wenden wir uns nun den entsprechenden Operatoren über Dünngitterräumen zu. Leider ist ein Nachweis für die gleichmäßige Stetigkeit dieser Operatoren bislang nicht gelungen. Die im Folgenden beschriebenen numerischen Experimente legen jedoch nahe, dass die Operatoren gleichmäßig stetig sind. Hierzu werden die Räume $V_n := V_{0,n}^{(1,1)}$ betrachtet. Zur Erinnerung: Bei den Funktionen $v \in V_n$ handelt es sich um stückweise lineare Funktionen über L_2-basierten Gittern, die auf dem Rand von $\Omega = [0,1]^d$ verschwinden.

Für eine vorgegebene Funktion g kann die Operatornorm der Abbildung $v \mapsto I_n(gv)$, $v \in V_n$, numerisch berechnet werden. Es ist nämlich

$$\sup_{v \in V_n} \frac{\big\|I_n(gv)\big\|_E^2}{\big\|v\big\|_E^2} \;=\; \sup_{\boldsymbol{v} \in \mathbb{R}^{\dim V_n}} \frac{\boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v}}{\boldsymbol{v}^T \boldsymbol{L} \boldsymbol{v}} \;=\; \lambda_{\max}, \tag{4.34}$$

wobei $\lambda_{\max}$ der größte Eigenwert des verallgemeinerten Eigenwertproblems

$$\boldsymbol{A}\boldsymbol{v} \;=\; \lambda \cdot \boldsymbol{L}\boldsymbol{v} \tag{4.35}$$

ist. Die Matrix $\boldsymbol{L}$ ist die Steifigkeitsmatrix des Laplace-Operators bezüglich der hierarchischen Basis von V_n. Die Matrix $\boldsymbol{A}$ ist gegeben durch

$$\boldsymbol{A} \;=\; \boldsymbol{H}^{-T} \cdot \boldsymbol{D}_g \cdot \boldsymbol{H}^T \cdot \boldsymbol{L} \cdot \boldsymbol{H} \cdot \boldsymbol{D}_g \cdot \boldsymbol{H}^{-1}.$$

Hier ist $\boldsymbol{H}$ die Matrix für den Wechsel von der nodalen in die hierarchische Basis. $\boldsymbol{D}_g$ ist die Diagonalmatrix, die auf der Diagonale die Funktionswerte von g trägt.

Tabelle 4.2 zeigt die numerisch berechneten Eigenwerte $\lambda_{\max}$ für die Funktionen

$$g(\boldsymbol{x}) = g_k(\boldsymbol{x}) = \sin(\pi k x_1), \qquad \mathrm{k} \in \{1, 2, 3\}$$

und Dimension $d \in \{1, 2, 3\}$. Man erkennt, dass $\lambda_{\max}$ mit wachsendem n zwar ansteigt. Dennoch legen die Werte die Vermutung nahe, dass es eine obere Schranke für $\lambda_{\max}$ gibt und die diskreten Multiplikationsoperatoren damit gleichmäßig stetig sind.

Die Funktionen g_k wurden übrigens im Hinblick auf einen späteren Beweis gewählt: Die Idee dabei ist, dass es genügt, die gleichmäßige Stetigkeit für alle Funktionen $\sin(\pi k x_1)$ und $\cos(\pi k x_1)$, $k \in \mathbb{N}_0$, zu zeigen. Da die Koordinate x_1 nicht ausgezeichnet ist, hat man damit sofort die gleichmäßige Stetigkeit aller Fouriermoden, die jeweils nur von x_i, $1 \leq i \leq d$, abhängen. Hieraus folgt wiederum, dass auch die Multiplikation mit den allgemeinen Fouriermoden

$$\begin{aligned} s_{\boldsymbol{k}}(\boldsymbol{x}) &= \sin(\pi k_1 x_1) \cdot \ldots \cdot \sin(\pi k_d x_d)\,, \\ c_{\boldsymbol{k}}(\boldsymbol{x}) &= \cos(\pi k_1 x_1) \cdot \ldots \cdot \sin(\pi k_d x_d)\,, \qquad \boldsymbol{k} \in \mathbb{N}_0^d, \end{aligned}$$

gleichmäßig stetig ist. Schließlich ist

$$I_h(s_{\boldsymbol{k}} v) = I_h\Big(sin(\pi k_1 x_1) \cdot I_h\Big(sin(\pi k_2 x_2) \cdot \ldots I_h\Big(sin(\pi k_d x_d) \cdot v\Big) \ldots\Big)\Big)$$

als Produkt gleichmäßig stetiger Operatoren selbst gleichmäßig stetig. Um nun die gleichmäßige Stetigkeit für die Multiplikation mit einer beliebigen glatten Funktion g nachzuweisen, genügt es g als Fourierreihe darzustellen und nachzuprüfen, dass die Fourier-Koeffizienten hinreichend schnell abfallen.

	n	$g(\boldsymbol{x}) = \sin(\pi x_1)$	$g(\boldsymbol{x}) = \sin(2\pi x_1)$	$g(\boldsymbol{x}) = \sin(3\pi x_1)$
$d = 1$	1	1.00000000	0.00000000	1.00000000
	2	1.07313218	1.41421356	2.07313218
	3	1.08675631	1.54992329	2.08992243
	4	1.09610415	1.59759765	2.21151681
	5	1.09880737	1.61000697	2.24354988
	6	1.09948788	1.61313237	2.25164768
	7	1.09965828	1.61391514	2.25367755
	8	1.09970090	1.61411092	2.25418536
	9	1.09971155	1.61415987	2.25431233
$d = 2$	1	1.00000000	0.00000000	1.00000000
	2	1.02772224	1.15977171	1.51227642
	3	1.03474197	1.22545831	1.56485718
	4	1.03864295	1.26086971	1.63469350
	5	1.04114795	1.27264972	1.66893713
	6	1.04204877	1.27690885	1.68107714
	7	1.04233573		1.68494891
	8			1.68621915
$d = 3$	1	1.00000000	0.00000000	1.00000000
	2	1.02129846	1.12306421	1.41881053
	3	1.04129110	1.23821081	1.69170311
	4	1.05227472	1.34598614	1.80471100
	5	1.07021140	1.47580713	1.96063740
	6	1.07352741	1.53247656	2.01504887

Tabelle 4.2: Eigenwerte $\lambda_{\max}$ nach (4.34)–(4.35) für die Normabschätzung bei den diskreten Multiplikationsoperatoren $v \mapsto I_n(gv)$ auf V_n. Dabei ist V_n der Raum stückweise linearer Funktionen über dem L_2-basierten Gitter der Tiefe n mit homogenen Randwerten.

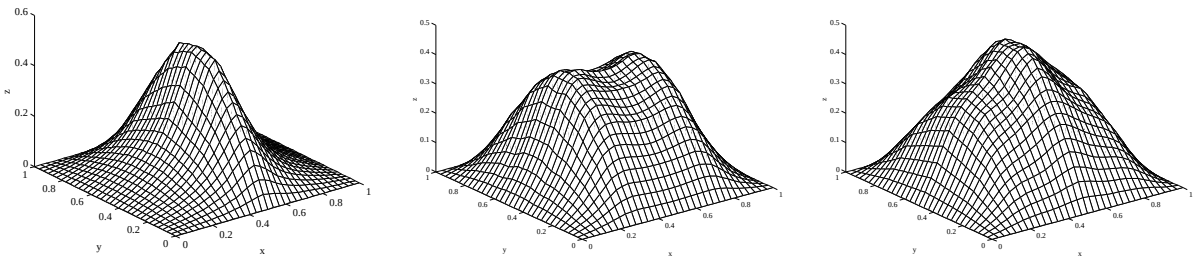


Abbildung 4.4: Eigenfunktionen zu den Eigenwerten aus Tabelle 4.2 ($d = 2,\ n = 5$) für $g(\boldsymbol{x}) = \sin(k\pi x_1)$, $k = 1, 2, 3$

4.4 Bemerkungen zur Symmetrisierung

Die vorgestellte modifizierte Bilinearform $\mathcal{A}_h$ hat einen bedeutenden Nachteil: Wenn die ursprüngliche Bilinearform $\mathcal{A}$ symmetrisch ist, ist die nach (4.6) modifizierte $\mathcal{A}_h$ im Allgemeinen nicht mehr symmetrisch. Dies hat weitreichende Folgen. Zum einen können Löser wie das cg-Verfahren, die Symmetrie voraussetzen, nicht benutzt werden[1]. Zum anderen werden physikalische Eigenschaften oder Prinzipien, die auf der Symmetrie beruhen, vom diskreten Modell unter Umständen nicht mehr exakt wiedergegeben.

Betrachten wir die Bilinearform

$$\mathcal{A}(u,v) \;=\; \int_\Omega \sum_{i=1}^{d} a(\boldsymbol{x})\, \partial_i u\, \partial_i v \, d\boldsymbol{x}, \tag{4.36}$$

wobei $a(\boldsymbol{x}) \geq C > 0$ für alle $\boldsymbol{x} \in \Omega$. Die nach (4.6) modifizierte Form ist gegeben durch

$$\mathcal{A}_h(u,v) \;=\; \int_\Omega \sum_{i=1}^{d} \big(\partial_i I_h(au) - I_h(\partial_i a \cdot u)\big) \cdot \partial_i v \, d\boldsymbol{x}\,. \tag{4.37}$$

Seien $\boldsymbol{u}$ und $\boldsymbol{v}$ die Koeffizientenvektoren von u bzw. v bezüglich der hierarchischen Basis. Dann lautet (4.37) in Matrixschreibweise

$$\mathcal{A}_h(u,v) \;=\; \sum_{i=1}^{d} \boldsymbol{v}^T \boldsymbol{S}_{ii}\, \boldsymbol{H}\boldsymbol{D}_a \boldsymbol{H}^{-1}\boldsymbol{u} \;-\; \boldsymbol{v}^T \boldsymbol{K}_i\, \boldsymbol{H}\boldsymbol{D}_{\partial_i a} \boldsymbol{H}^{-1}\boldsymbol{u}\,.$$

Dabei beschreibt die Matrix $\boldsymbol{H}$ den Wechsel von der nodalen in die hierarchische Basis, $\boldsymbol{D}_f$ ist die Diagonalmatrix, deren Einträge aus den Funktionswerten von $f : \Omega \to \mathbb{R}$ an den Gitterpunkten bestehen. Die Matrizen $\boldsymbol{S}_{ij}$ und $\boldsymbol{K}_i$ repräsentieren die Bilinearformen

$$\boldsymbol{v}^T \boldsymbol{S}_{ij} \boldsymbol{u} \;=\; \int_\Omega \partial_i u\, \partial_j v \, d\boldsymbol{x}\,, \qquad \boldsymbol{v}^T \boldsymbol{K}_i \boldsymbol{u} \;=\; \int_\Omega u\, \partial_i v \, d\boldsymbol{x}\,.$$

Ein einfacher Ansatz, die Bilinearform $\mathcal{A}_h$ in eine symmetrische Variante umzugestalten, ist der folgende: Wegen der Positivität von a lässt sich (4.36) schreiben als

$$\mathcal{A}(u,v) \;=\; \int_\Omega \sum_{i=1}^{d} \sqrt{a}\, \partial_i u \;\cdot\; \sqrt{a}\, \partial_i v \, d\boldsymbol{x}\,.$$

Eine der Form $\mathcal{A}_h$ entsprechende, gitterabhängige Modifikation, die nur die Werte von $\sqrt{a}$ bzw. $\partial_i \sqrt{a}$ an den Gitterpunkten benötigt, lautet

$$\mathcal{A}_h^{\mathrm{sym}}(u,v) \;=\; \int_\Omega \sum_{i=1}^{d} \Big(\partial_i I_h(\sqrt{a}u) - I_h(\partial_i \sqrt{a} \cdot u)\Big) \cdot \Big(\partial_i I_h(\sqrt{a}v) - I_h(\partial_i \sqrt{a} \cdot v)\Big)\, d\boldsymbol{x}\,. \tag{4.38}$$

[1] Tatsächlich zeigt sich das cg-Verfahren für die im nächsten Kapitel vorgestellten numerischen Beispiele trotz der (algebraischen) Unsymmetrie konvergent, ein Hinweis dafür, dass die Bilinearform lediglich „schwach unsymmetrisch“ ist. Hier lohnt eventuell eine weitergehende Untersuchung.

Die Bilinearform $\mathcal{A}_h^{\mathrm{sym}}$ ist offensichtlich symmetrisch. In Matrixschreibweise ist

$$\begin{aligned}\mathcal{A}_h^{\mathrm{sym}}(u,v) \;=\; \boldsymbol{v}^T \cdot \sum_{i=1}^{d} \Big\{ \;& \boldsymbol{H}^{-T}\boldsymbol{D}_{\sqrt{a}}\boldsymbol{H}^T\,\boldsymbol{S}_{ii}\,\boldsymbol{H}\boldsymbol{D}_{\sqrt{a}}\boldsymbol{H}^{-1} \\ & - \boldsymbol{H}^{-T}\boldsymbol{D}_{\sqrt{a}}\boldsymbol{H}^T\,\boldsymbol{K}_i\,\boldsymbol{H}\boldsymbol{D}_{\partial_i\sqrt{a}}\boldsymbol{H}^{-1} \\ & - \boldsymbol{H}^{-T}\boldsymbol{D}_{\partial_i\sqrt{a}}\boldsymbol{H}^T\,\boldsymbol{K}_i^T\,\boldsymbol{H}\boldsymbol{D}_{\sqrt{a}}\boldsymbol{H}^{-1} \\ & + \boldsymbol{H}^{-T}\boldsymbol{D}_{\partial_i\sqrt{a}}\boldsymbol{H}^T\,\boldsymbol{M}\,\boldsymbol{H}\boldsymbol{D}_{\partial_i\sqrt{a}}\boldsymbol{H}^{-1} \Big\} \cdot \boldsymbol{u}\,.\end{aligned}$$

Dabei ist $\boldsymbol{M}$ die Massenmatrix bezüglich der hierarchischen Basis. An der Matrixschreibweise ist abzulesen, wie die Matrix-Vektor-Multiplikation mit der Steifigkeitsmatrix als hintereinander auszuführende Multiplikation mit den Matrizen $\boldsymbol{H}^{-1}$, $\boldsymbol{D}_{\sqrt{a}}$, etc. zu realisieren ist. Neben den Algorithmen für das Hierarchisieren bzw. Dehierarchisieren aus Abschnitt 3.6.3 – hier durch die Matrizen $\boldsymbol{H}$ bzw. $\boldsymbol{H}^{-1}$ wiedergegeben – und den Zwei-Term-Integralen, die bei der Multiplikation mit $\boldsymbol{S}_{ij}$, $\boldsymbol{K}_i$ und $\boldsymbol{M}$ auszuwerten sind (siehe Abschnitt 3.6.1), treten nun auch die transponierten Operatoren $\boldsymbol{H}^T$ bzw. $\boldsymbol{H}^{-T}$ auf. Die Multiplikation mit diesen Operatoren lässt sich nach den Erläuterungen in Abschnitt 3.6.5 ebenfalls mit einem $O(N)$-Aufwand ausführen.

Die Effizienz der Matrix-Vektor-Multiplikation ist also gesichert. Allerdings stellt sich heraus, dass sich die Approximationseigenschaften der mit $\mathcal{A}_h^{\mathrm{sym}}$ berechneten Galerkin-Lösung essentiell verschlechtern. Betrachten wir hierzu das folgende Randwertproblem über $\Omega = [0,1]^d$, $d \in \mathbb{N}$:

$$\begin{aligned} -\nabla\left(\left(1+\sum_{i=1}^{d} x_i\right)^2 \nabla u\right) &= f \qquad \text{auf } \Omega \\ u &= 0 \qquad \text{auf } \partial\Omega. \end{aligned} \tag{4.39}$$

Die rechte Seite f wird so gewählt, dass die Lösung gegeben ist durch

$$u(\boldsymbol{x}) \;=\; \prod_{i=1}^{d} \sin(\pi x_i)\;.$$

Im Folgenden wird mit u_h die Näherungslösung des mit $\mathcal{A}_h^{\mathrm{sym}}$ diskretisierten Problems bezeichnet.

Um einen möglichst umfassenden Überblick über die Approximationsgüte von u_h zu bekommen, wurde das Beispiel für verschiedene Ordnung $p \in \{1,2,3,4\}$, auf vollen und dünnen Gittern, jeweils für $d \in \{1,2,3\}$ gerechnet. Die Abbildungen 4.5 und 4.6 zeigen den Fehler $u - u_h$ gemessen zum einen in der Energienorm, zum anderen in der L_2-Norm. In Tabelle 4.3 ist das beobachtete Konvergenzverhalten dem erwarteten Verhalten gegenübergestellt. Mit dem erwarteten Verhalten ist dabei das für den Interpolationsfehler a-priori bekannte, in (3.50) angegebene Approximationsverhalten gemeint.

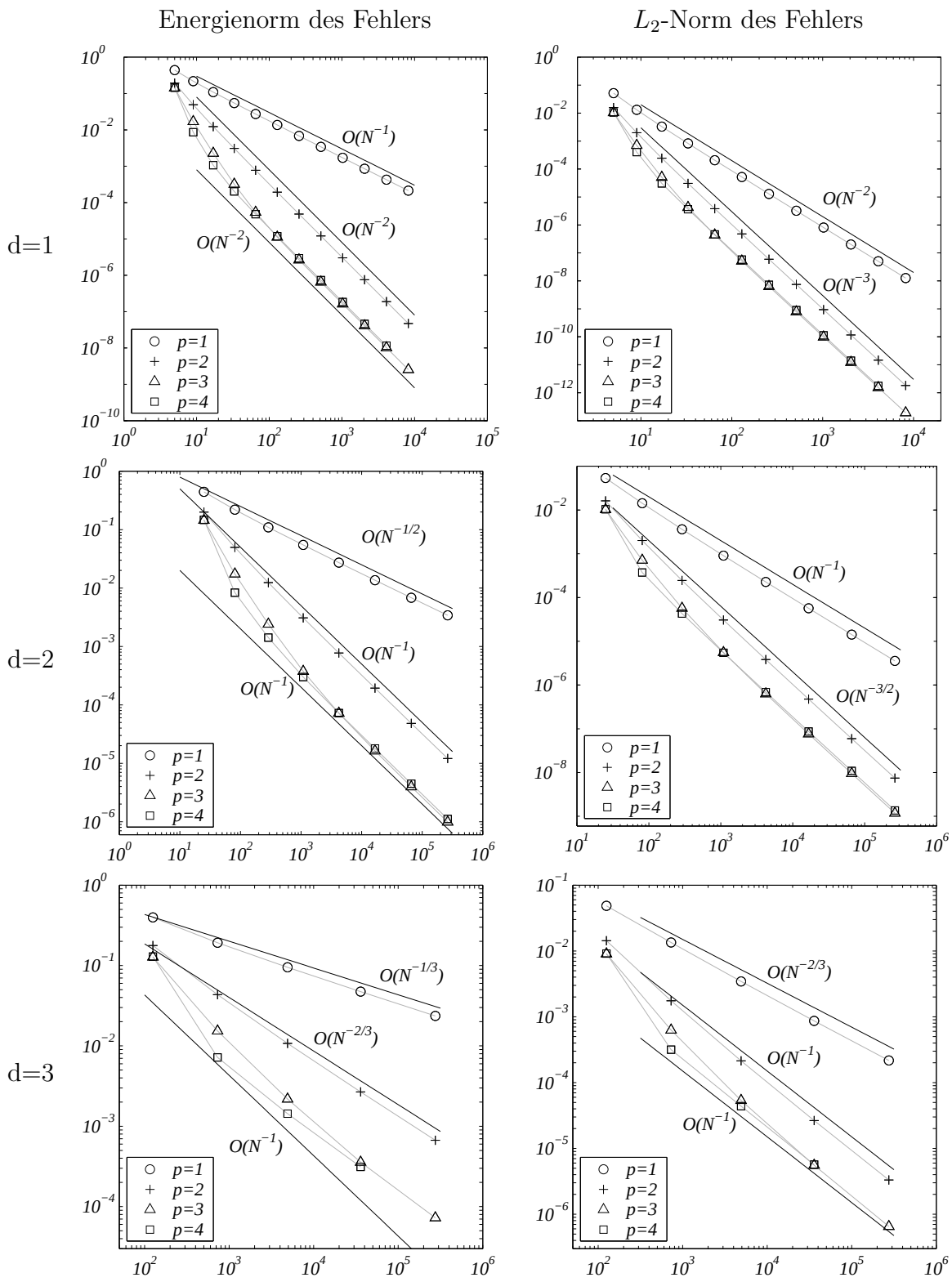


Abbildung 4.5: Symmetrisierte Bilinearform: Energienorm (links) und L_2-Norm (rechts) des Fehlers aufgetragen gegen die Anzahl der Gitterpunkte. Hier wurden volle Gitter verwendet.

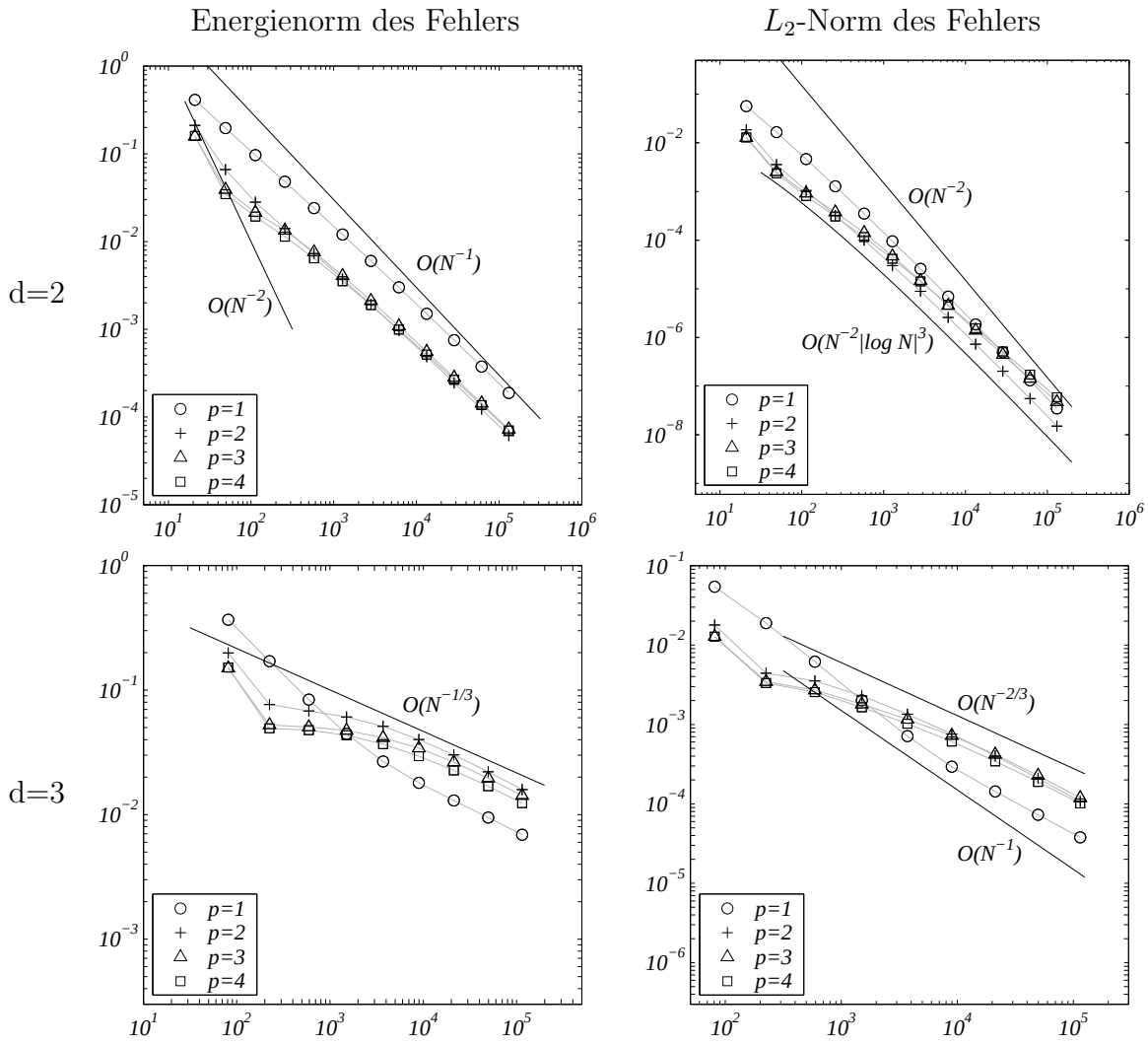


Abbildung 4.6: Symmetrisierte Bilinearform: Energienorm (links) und L_2-Norm (rechts) des Fehlers aufgetragen gegen die Anzahl der Gitterpunkte. Hier wurden dünne L_2-basierte Gitter verwendet.

	Energienorm des Fehlers		L_2-Norm des Fehlers	
	beobachtet	erwartet	beobachtet	erwartet
	d=1			
$p=1$	N^{-1}	N^{-1}	N^{-2}	N^{-2}
2	N^{-2}	N^{-2}	N^{-3}	N^{-3}
3	N^{-2}	N^{-3} !	N^{-3}	N^{-4} !
4	N^{-2}	N^{-4} !	N^{-3}	N^{-5} !
	d=2 Vollgitter			
$p=1$	$N^{-1/2}$	$N^{-1/2}$	N^{-1}	N^{-1}
2	N^{-1}	N^{-1}	$N^{-3/2}$	$N^{-3/2}$
3	N^{-1}	$N^{-3/2}$!	$N^{-3/2}$	N^{-2} !
4	N^{-1}	N^{-2} !	$N^{-3/2}$	$N^{-5/2}$!
	d=3 Vollgitter			
$p=1$	$N^{-1/3}$	$N^{-1/3}$	$N^{-2/3}$	$N^{-2/3}$
2	$N^{-2/3}$	$N^{-2/3}$	N^{-1}	N^{-1}
3	$\approx N^{-2/3}$	N^{-1} !	$\approx N^{-1}$	$N^{-4/3}$!
4	$\approx N^{-2/3}$	$N^{-4/3}$!	$\approx N^{-1}$	$N^{-5/3}$!
	d=2 Dünngitter			
$p=1$	$\approx N^{-1}$	$N^{-1}\lvert\log N\rvert$	$\approx N^{-2}\lvert\log N\rvert^3$	$N^{-2}\lvert\log N\rvert^3$
2	$\approx N^{-1}$	$N^{-2}\lvert\log N\rvert^2$!	$\approx N^{-2}\lvert\log N\rvert^3$	$N^{-3}\lvert\log N\rvert^4$!
3	$\approx N^{-1}$	$N^{-3}\lvert\log N\rvert^3$!	$\approx N^{-2}\lvert\log N\rvert^3$	$N^{-4}\lvert\log N\rvert^5$!
4	$\approx N^{-1}$	$N^{-4}\lvert\log N\rvert^4$!	$\approx N^{-2}\lvert\log N\rvert^3$	$N^{-5}\lvert\log N\rvert^6$!
	d=3 Dünngitter			
$p=1$	$\approx N^{-1/3}$	$N^{-1}\lvert\log N\rvert^2$!	$\approx N^{-2/3}$	$N^{-2}\lvert\log N\rvert^6$!
2	$\approx N^{-1/3}$	$N^{-2}\lvert\log N\rvert^4$!	$\approx N^{-2/3}$	$N^{-3}\lvert\log N\rvert^8$!
3	$\approx N^{-1/3}$	$N^{-3}\lvert\log N\rvert^6$!	$\approx N^{-2/3}$	$N^{-4}\lvert\log N\rvert^{10}$!
4	$\approx N^{-1/3}$	$N^{-4}\lvert\log N\rvert^8$!	$\approx N^{-2/3}$	$N^{-5}\lvert\log N\rvert^{12}$!

Tabelle 4.3: Symmetrisierte Bilinearform: Abweichung der beobachteten Fehler von der erwarteten Konvergenzordnung in der Anzahl N der Gitterpunkte. Die Diskrepanzen sind mit ! gekennzeichnet.

Folgendes ist festzuhalten: Für die Vollgitter-Näherungen stimmt das asymptotische Verhalten des Fehlers für $p \in \{1, 2\}$ mit dem erwarteten Verhalten überein. Für Ansatzräume höherer Ordnung $p > 2$ ergibt sich allerdings entgegen der Erwartung keine weitere Erhöhung der Konvergenzrate. Für dünne Gitter verschärft sich diese Situation noch weiter: Hier ist nur für das zweidimensionale Problem und stückweise lineare Ansatzfunktionen die erwartete Konvergenzrate zu beobachten, für die anderen Testfälle ist das Konvergenzverhalten meist sogar schlechter als bei der entsprechenden Vollgitter-Lösung. Damit ist die hier vorgestellte, symmetrische Bilinearform $\mathcal{A}_h^{\text{sym}}$ leider wertlos.

Das erste Lemma von Strang (Satz 4.1, Abschnitt 4.3) gibt zwar keine untere Schranke für den Fehler an, kann also das Versagen der Bilinearform $\mathcal{A}_h^{\text{sym}}$ nicht begründen. Es gibt allerdings einen Hinweis: Kernstück für die Abschätzung ist der Ausdruck

$$\inf_{v \in V_h} \sup_{w \in V_h} \frac{|\mathcal{A}(v,w) - \mathcal{A}_h(v,w)|}{\|w\|} \ .$$

Während für die Ansatzfunktion (hier v) eine möglichst gutartige Funktion (in der Regel der Interpolant der Lösung) eingesetzt werden kann, um eine obere Schranke zu bekommen, ist unter den Testfunktionen (hier w) diejenige zu berücksichtigen, die die größte Differenz $\mathcal{A}(v,w) - \mathcal{A}_h(v,w)$ erzeugt. Für die nach (4.6) definierte gitterabhängige Bilinearform $\mathcal{A}_h$ war dieser Umstand unproblematisch: Die Modifikation der Bilinearform betrifft dort nur die Ansatzfunktion, nicht aber die Testfunktion (siehe hierzu auch den Beweis von Satz 4.2). Für die symmetrisierte Form $\mathcal{A}_h^{\text{sym}}$ hingegen gilt es nun, die Differenzen der Art

$$\sqrt{a}w \ - \ I_h(\sqrt{a}w) \tag{4.40}$$

abzuschätzen. Für glatte Testfunktionen w ist die Differenz klein, für „nicht-glatte“ w in der Regel groß gegenüber h. Bei dünnen Gitter ist dies z.B. für die Basisfunktionen mit dem größten Seitenverhältnis des Trägers der Fall, wo $I_h(\sqrt{a}w) = \sqrt{a(\boldsymbol{r})} \cdot w$ ist ($\boldsymbol{r}$ ist der Knoten der Basisfunktion w).

5 Numerische Ergebnisse

Durch die Präsentation einiger Testrechnungen soll in diesem Kapitel die Güte der vorgestellten Diskretisierung beleuchtet werden. Hauptkriterium ist dabei, ob der Fehler der Galerkinlösung das gleiche asymptotische Verhalten aufweist wie die Differenz von exakter Lösung und ihrem Interpolanten. Zunächst werden diskrete Fehlernormen eingeführt, die zwar nur Näherungen für ihre kontinuierlichen Entsprechungen darstellen, sich aber leicht numerisch berechnen lassen.

5.1 Diskrete Fehlernormen

Es bezeichne u die exakte Lösung und u_h die diskrete Lösung über dem Gitter G_h. Traditionell interessieren beim Fehler $e_h = u - u_h$ die Normen

$$\|e_h\|_E,\ \|e_h\|_{L_2} \text{ und } \|e_h\|_{L_\infty}. \tag{5.1}$$

Liegt die Lösung u in geschlossener Form vor, können diese Werte berechnet werden. Aus Praktikabilitätsgründen weichen wir jedoch auf diskrete Varianten dieser Normen aus, die zum einen gute Näherungen für die tatsächlichen Fehler ergeben, zum anderen mit den vorhandenen Algorithmen berechnet werden können.

Zu diesem Zweck wird e_h ersetzt durch seinen Interpolanten auf dem nächst feineren Gitter $G_{h/2}$,

$$\tilde{e}_h \ := \ I_{h/2}(e_h) \ = \ I_{h/2}(u) - u_h. \tag{5.2}$$

Das Gitter $G_{h/2}$ geht aus G_h dadurch hervor, dass zu allen Blattknoten sämtliche Sohnknoten eingefügt werden, vergleiche hierzu Abschnitt 3.5. Dieses Vorgehen wird durch den exponentiellen Abfall der Beiträge mit zunehmenden Level gerechtfertigt. So ist zu erwarten, dass der Hauptanteil des Fehlers e_h durch die Beiträge an den Punkten $G_{h/2} \setminus G_h$ erfasst wird.

Für die integralen Normen haben wir nun die Approximationen

$$\|e_h\|_E \ \approx \ \|\tilde{e}_h\|_E \ = \ (\boldsymbol{e}^T \boldsymbol{L}\, \boldsymbol{e})^{\frac{1}{2}}, \tag{5.3}$$

$$\|e_h\|_{L_2} \ \approx \ \|\tilde{e}_h\|_{L_2} \ = \ (\boldsymbol{e}^T \boldsymbol{M}\, \boldsymbol{e})^{\frac{1}{2}}. \tag{5.4}$$

Dabei ist $\boldsymbol{e}$ der Koeffizientenvektor von $\tilde{e}_h$, $\boldsymbol{L}$ ist die Steifigkeitsmatrix des Laplaceoperators und $\boldsymbol{M}$ die Massenmatrix, jeweils über dem Gitter $G_{h/2}$. Für den Fall transformierter Gitter ist noch zu berücksichtigen, dass die exakten diskreten Operatoren $\boldsymbol{L}$ bzw. $\boldsymbol{M}$ durch die modifizierten Operatoren aus Abschnitt 4.2.3 ersetzt werden.

Eine geeignete Näherung für den Fehler in der L_∞-Norm bekommt man, indem man anstelle des Supremums über alle Punkte $\boldsymbol{x} \in \Omega$ nur Punkte des Gitters $G_{h/2}$ betrachtet:

$$\|e_h\|_{L_\infty} \approx \max_{\boldsymbol{x} \in G_{h/2}} |\tilde{e}_h(\boldsymbol{x})|. \tag{5.5}$$

Wenn in der folgenden Darstellung Zahlenwerte genannt oder Diagramme mit Fehlern gezeigt werden, beziehen sich diese auf die hier beschriebenen, diskreten Varianten.

5.2 Variable Koeffizienten

Zunächst betrachten wir partielle Differentialgleichungen über $[0,1]^d$ mit variablen Koeffizienten. Die folgenden Beispiele sollen darüber Aufschluss geben, inwiefern die nach (4.9) modifizierte, gitterabhängige Bilinearform ein brauchbarer Ersatz für die exakte Bilinearform ist. Die Beispiele sind so gewählt, dass eine Auswertung der exakten Bilinearform mit dem Verfahren aus Abschnitt 4.2.2 geschehen kann und damit ein Vergleich von exakt ausgewerteter und modifizierter Bilinearform möglich ist.

5.2.1 Ein 2D-Beispiel

Betrachten wir folgendes Randwertproblem:

Beispiel 5.1

$$-\nabla(C\nabla u) = f \qquad \text{in } \Omega = [0,1]^2, \tag{5.6}$$

$$u(x,y) = \sin(3x)\sin(4y) + \sin(5x)\sin(6y) \qquad \text{auf } \partial\Omega\ , \tag{5.7}$$

mit dem Diffusionskoeffizienten

$$C(x,y) = \begin{pmatrix} 1 + \frac{1}{4}\cos(4x)\cos(5y) & \frac{1}{4}\cos(2x)\cos(2y) \\ \frac{1}{4}\cos(2x)\cos(2y) & 1 + \frac{1}{4}\cos(4x)\cos(5y) \end{pmatrix}.$$

Die rechte Seite f ist so gewählt, dass die Lösung u im Innern von Ω durch den Ausdruck (5.7) gegeben ist.

Das Gebiet $\Omega = [0,1]^2$ bzw. der Raum $H^1(\Omega)$ wird mit einem einzigen Dünngitterelement diskretisiert. Grundlage für die numerischen Berechnungen sind regelmäßige dünne Gitter. Auf lokale Adaption wird wegen der Glattheit der Lösung verzichtet.

Berechnungen mit der modifizierten Bilinearform nach (4.9)

Die Abbildungen 5.1 und 5.2 zeigen die Abhängigkeit des Fehlers $u - u_h$ von der Anzahl der Gitterpunkte. Die Fehler weisen für alle betrachteten Normen ein stabiles, asymptotisches Verhalten auf bis in den Bereich, in dem der Fehler von der Größenordnung der Maschinengenauigkeit $\approx 10^{-16}$ ist. Der Genauigkeitsgewinn durch Erhöhung der

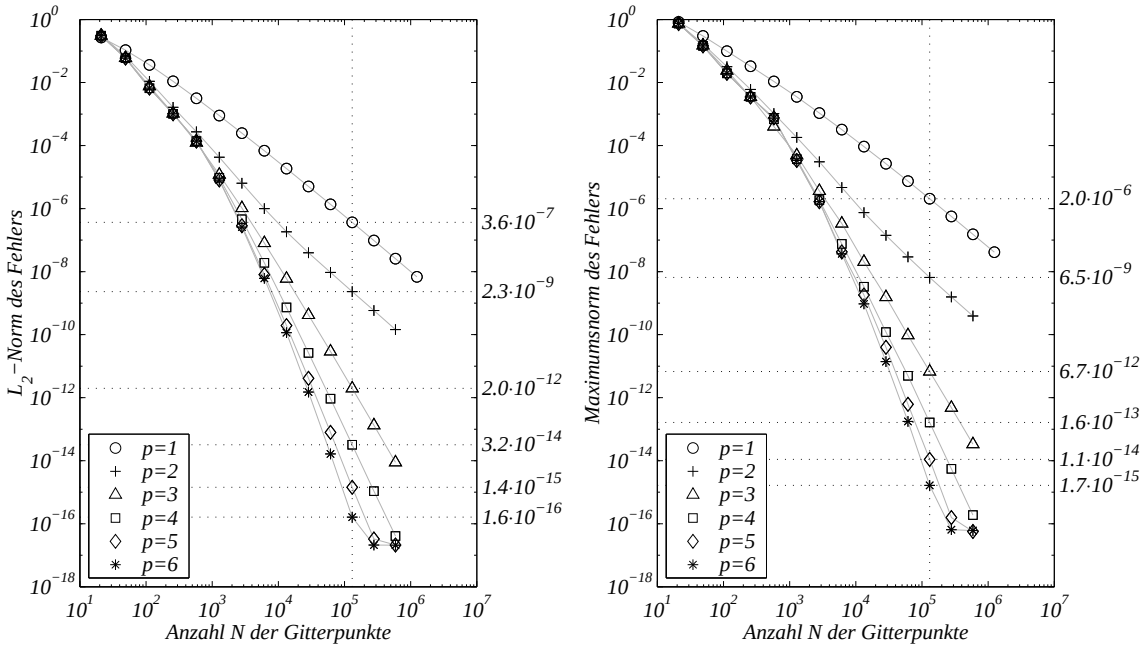


Abbildung 5.1: Beispiel 5.1 – Fehler in L_2-Norm (links) und in L_∞-Norm (rechts) für L_2-basierte Dünngitterräume $V_n^{(p,1)}$

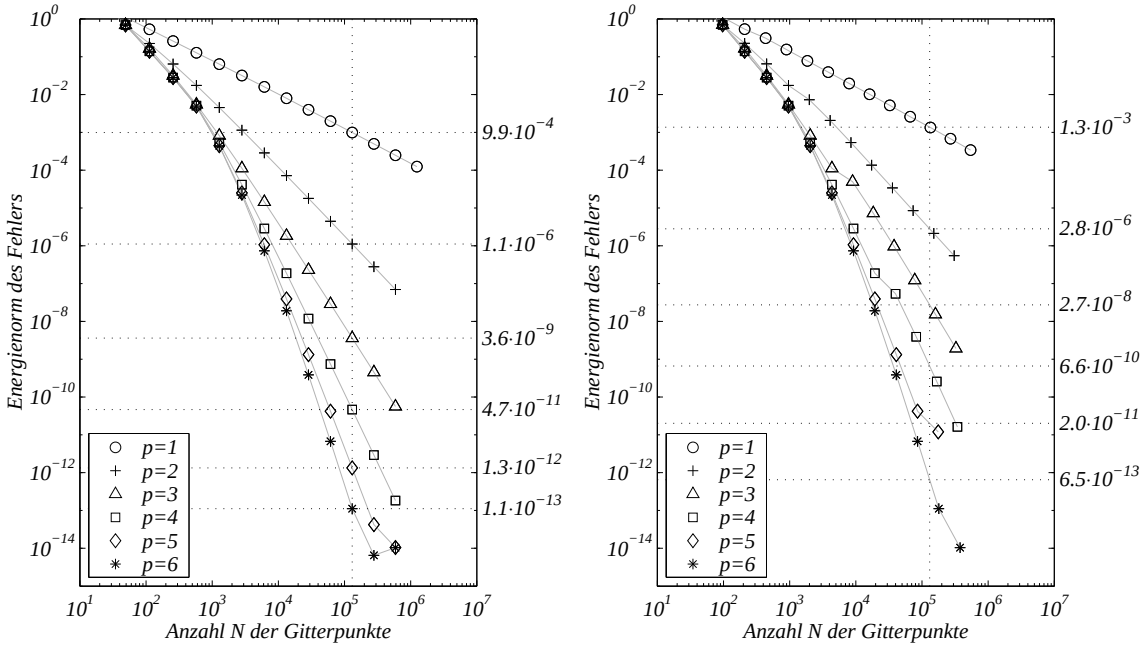


Abbildung 5.2: Beispiel 5.1 – Energienorm des Fehlers für L_2-basierte Dünngitterräume $V_n^{(p,1)}$ (links) und für energiebasierte Dünngitterräume $V_n^{(p,E)}$ (rechts)

Ordnung im Ansatzraum tritt deutlich hervor. Erstaunlich ist, dass der Fehler, gemessen in der Energienorm, für energiebasierte Gitter größer ist als bei L_2-basierten Gittern. Grund ist, dass die für energiebasierte Gitter ausgewiesene Asymptotik (3.50) erst spät einsetzt, zu erkennen etwa an dem „Knick“ in der Fehlerkurve für $p = 4$ bei $N \approx 3 \cdot 10^4$ (Abbildung 5.2, rechts).

In Abbildung 5.4 wird aufgezeigt, inwiefern das nach (3.50) für den Interpolationsfehler vorausgesagte asymptotische Verhalten mit dem Fehler bei der Galerkin-Approximation übereinstimmt. Für $p = 1$ oder $p \geq 3$ ist eine sehr gute Übereinstimmung zu erkennen. Auffällig sind jedoch die mit p stark ansteigenden Konstanten, die als Vorfaktor bei den Asymptotiken auftreten. Im Abschnitt „Vergleich FEM über Rechtecksgitter„ weiter unten wird hierauf näher eingegangen. Für $p = 2$ zeigt der Fehler nicht das erwartete Verhalten. Grund ist, dass die hier gebrachten Ergebnisse auf der nach (4.9) modifizierten Bilinearform beruhen, wobei die Interpolationen in (4.8) entgegen der auf Seite 76 ausgesprochenen Empfehlung mit der Ordnung $\hat{p} = p$, also der Ordnung des Ansatzraums, durchgeführt werden. Erhöht man die Ordnung für die Interpolation um eins, bekommt man wieder das gewünschte asymptotische Verhalten, siehe Abbildung 5.3.

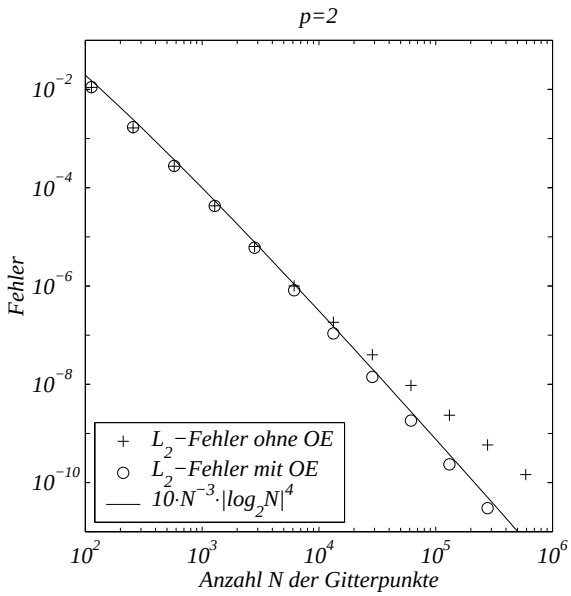


Abbildung 5.3: Beispiel 5.1 – Asymptotisches Verhalten des Fehlers in der L_2-Norm mit und ohne Ordnungserhöhung bei der Interpolation der Koeffizienten in der modifizierten Bilinearform

Vergleich mit exakter Auswertung der Bilinearform

Da die Koeffizientenfunktionen jeweils Summe zweier Tensorprodukte aus univariaten Funktionen sind, kann die Bilinearform $\mathcal{A}$ mit den in Abschnitt 4.2.2 erläuterten Algorithmen exakt ausgewertet werden. Wir wollen diesen Umstand nutzen, um uns ein

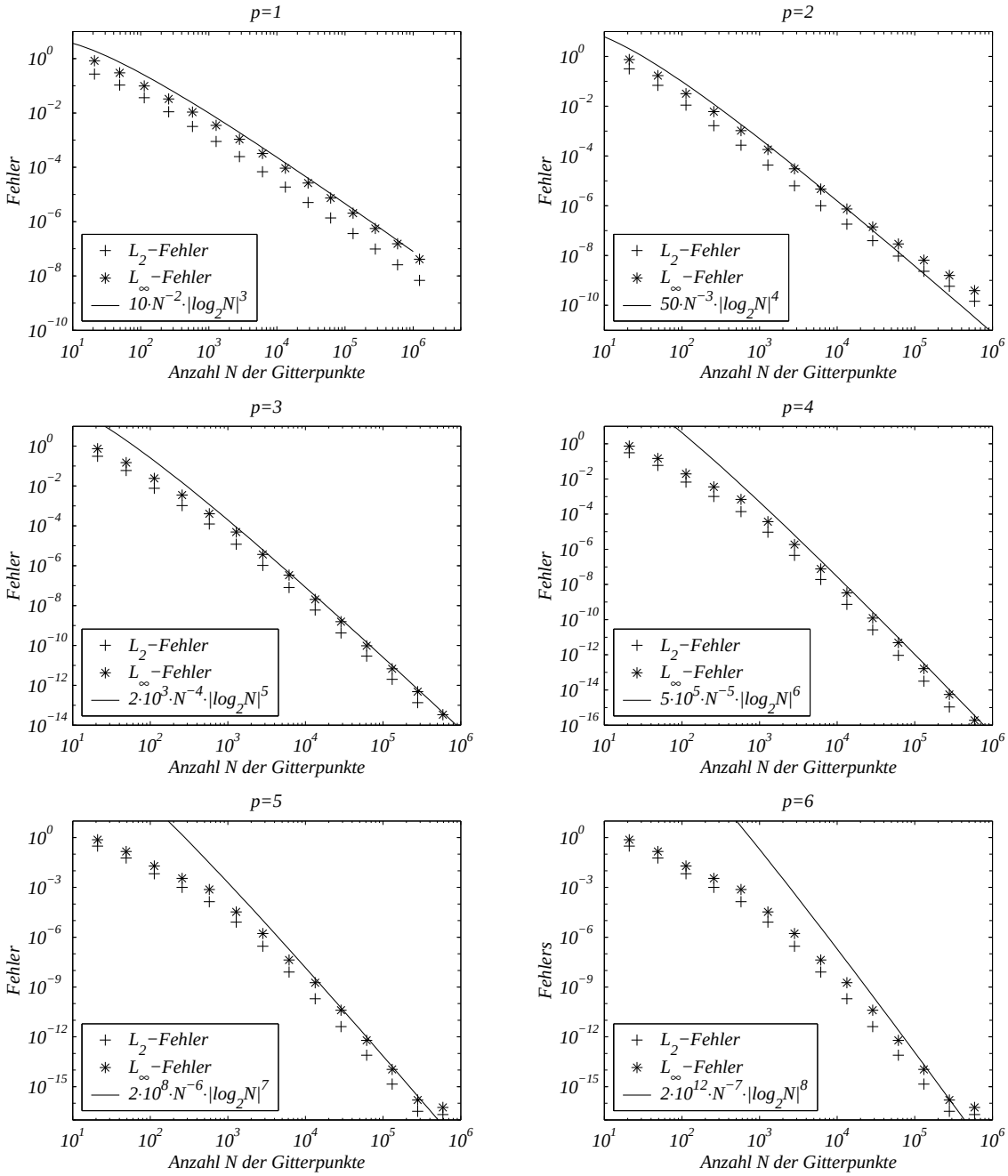


Abbildung 5.4: Beispiel 5.1 – Asymptotisches Verhalten des Fehlers in L_2- und L_∞-Norm im Vergleich mit der a-priori bekannten Asymptotik (3.50) des Interpolationsfehlers

Bild über den durch die Modifikation der Bilinearform verursachten Fehler zu machen. Tabelle 5.1 stellt die Ergebnisse aus der Diskretisierung mit exakter und modifizierter Bilinearform gegenüber. Folgendes ist festzustellen: Für $p \geq 2$ liefern beide Diskretisierungen verschiedene Ergebnisse, wobei die exakt ausgewertete Bilinearform jeweils den kleineren Fehler $\|u - u_h\|_{L_2}$ produziert. Der Quotient

$$q_h := \frac{\|u - u_h^{\text{modif}}\|_{L_2}}{\|u - u_h^{\text{exakt}}\|_{L_2}} \tag{5.8}$$

wächst für kleiner werdende Maschenweite zunächst, bleibt aber nach oben beschränkt durch eine Konstante, die nur von p abhängt und umso größer ist, je größer p ist. Das asymptotische Verhalten des Fehlers ist damit für beide Verfahren das gleiche.

N	$\|u - u_h^{\text{modif}}\|_{L_2}$	$\|u - u_h^{\text{exakt}}\|_{L_2}$	q_h	$\|u - u_h^{\text{modif}}\|_{L_2}$	$\|u - u_h^{\text{exakt}}\|_{L_2}$	q_h
	$p = 1$			$p = 2$		
21	$2.07 \cdot 10^{0}$	$2.07 \cdot 10^{0}$	1.00	$2.52 \cdot 10^{0}$	$2.61 \cdot 10^{0}$	0.96
49	$1.09 \cdot 10^{0}$	$1.08 \cdot 10^{0}$	1.01	$6.87 \cdot 10^{-1}$	$7.14 \cdot 10^{-1}$	0.96
113	$5.30 \cdot 10^{-1}$	$5.17 \cdot 10^{-1}$	1.02	$2.27 \cdot 10^{-1}$	$2.06 \cdot 10^{-1}$	1.10
257	$2.57 \cdot 10^{-1}$	$2.49 \cdot 10^{-1}$	1.03	$6.52 \cdot 10^{-2}$	$5.18 \cdot 10^{-2}$	1.26
577	$1.27 \cdot 10^{-1}$	$1.22 \cdot 10^{-1}$	1.04	$1.77 \cdot 10^{-2}$	$1.31 \cdot 10^{-2}$	1.35
1281	$6.34 \cdot 10^{-2}$	$6.07 \cdot 10^{-2}$	1.04	$4.56 \cdot 10^{-3}$	$3.28 \cdot 10^{-3}$	1.39
2817	$3.17 \cdot 10^{-2}$	$3.03 \cdot 10^{-2}$	1.05	$1.15 \cdot 10^{-3}$	$8.20 \cdot 10^{-4}$	1.40
6145	$1.59 \cdot 10^{-2}$	$1.51 \cdot 10^{-2}$	1.05	$2.89 \cdot 10^{-4}$	$2.05 \cdot 10^{-4}$	1.41
13313	$7.93 \cdot 10^{-3}$	$7.56 \cdot 10^{-3}$	1.05	$7.23 \cdot 10^{-5}$	$5.13 \cdot 10^{-5}$	1.41
28673	$3.97 \cdot 10^{-3}$	$3.78 \cdot 10^{-3}$	1.05	$1.81 \cdot 10^{-5}$	$1.28 \cdot 10^{-5}$	1.41
	$p = 3$			$p = 4$		
21	$2.59 \cdot 10^{0}$	$2.67 \cdot 10^{0}$	0.97	$2.59 \cdot 10^{0}$	$2.68 \cdot 10^{0}$	0.98
49	$6.91 \cdot 10^{-1}$	$7.19 \cdot 10^{-1}$	0.96	$6.90 \cdot 10^{-1}$	$7.19 \cdot 10^{-1}$	0.96
113	$1.64 \cdot 10^{-1}$	$1.38 \cdot 10^{-1}$	1.19	$1.39 \cdot 10^{-1}$	$1.01 \cdot 10^{-1}$	1.38
257	$3.14 \cdot 10^{-2}$	$1.80 \cdot 10^{-2}$	1.75	$2.80 \cdot 10^{-2}$	$1.01 \cdot 10^{-2}$	2.77
577	$5.46 \cdot 10^{-3}$	$2.22 \cdot 10^{-3}$	2.46	$5.07 \cdot 10^{-3}$	$8.31 \cdot 10^{-4}$	6.10
1281	$8.19 \cdot 10^{-4}$	$2.71 \cdot 10^{-4}$	3.02	$5.37 \cdot 10^{-4}$	$5.08 \cdot 10^{-5}$	10.6
2817	$1.12 \cdot 10^{-4}$	$3.36 \cdot 10^{-5}$	3.34	$4.15 \cdot 10^{-5}$	$2.91 \cdot 10^{-6}$	14.3
6145	$1.45 \cdot 10^{-5}$	$4.20 \cdot 10^{-6}$	3.46	$2.87 \cdot 10^{-6}$	$1.77 \cdot 10^{-7}$	16.2
13313	$1.84 \cdot 10^{-6}$	$5.24 \cdot 10^{-7}$	3.51	$1.87 \cdot 10^{-7}$	$1.11 \cdot 10^{-8}$	17.0
28673	$2.31 \cdot 10^{-7}$	$6.55 \cdot 10^{-8}$	3.53			

Tabelle 5.1: Beispiel 5.1 – Vergleich des L_2-Fehlers bei exakt ausgewerteter Bilinearform und bei modifizierter Bilinearform. N ist die Anzahl der Gitterpunkte, q_h ist der in (5.8) definierte Quotient.

Vergleich mit FEM über Rechtecksgitter (h-Version)

Die obigen Testrechnungen reproduzieren für den Fehler gemessen in der L_2-Norm das asymptotische Verhalten, wie es für den Interpolationsfehler in (3.50) vorhergesagt wird. Damit ist das Dünngitterverfahren dem Vollgitterverfahren bzw. der klassischen h-Version der FEM theoretisch überlegen: Für feste Ordnung p hat man für dünne Gitter Fehler der Art $O(N^{-(p+1)}|\log_2 N|^{s(p,d)})$, für volle Gitter $O(N^{-(p+1)/d})$. Damit ist klar, dass ab einer bestimmten Zahl von Gitterpunkten das Dünngitterverfahren den kleineren Fehler liefern muss. Für die praktische Bedeutung des Dünngitterverfahrens ist nun entscheidend, ab welcher Gitterpunktzahl dieser Effekt eintritt. Die Notwendigkeit einer diesbezüglichen Untersuchung ergibt sich außerdem aus dem Umstand, dass die experimentell bestimmten Konstanten C in der asymptotischen Kurve $C \cdot N^{-(p+1)} \cdot |\log_2 N|^s$ mit zunehmender Ordnung p sehr schnell ansteigen, vergleiche Abbildung 5.4.

Zu diesem Zweck wurde das Randwertproblem aus Beispiel 5.1 über äquidistanten Rechtecksgittern mit Lagrange-Tensorproduktelementen berechnet (vergleiche Seite 10). Hierfür kam die an der Universität Heidelberg entwickelte Programmbibliothek `deal.II` zum Einsatz [7, 8]. Abbildung 5.5 zeigt die Ergebnisse dieser Berechnungen im Vergleich mit den Ergebnissen für die Dünngitter-Diskretisierung. Der genannte Vorsprung des Dünngitterverfahrens ab einer bestimmten Anzahl von Git-

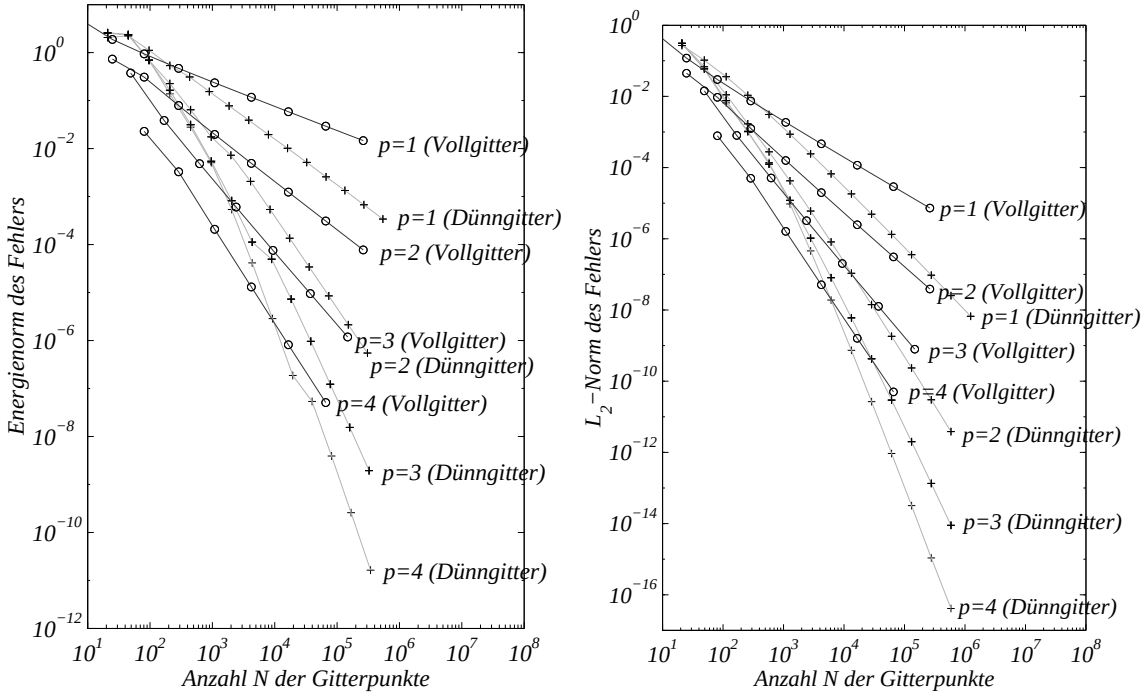


Abbildung 5.5: Beispiel 5.1 – Vergleich von Dünngitterelementen mit FE-Diskretisierung über äquidistantem Rechtecksgitter (Lagrange-Tensorproduktelemente, h-Version der FEM). Links: Fehler in Energienorm (energiebasierte Dünngitter), rechts: Fehler in L_2-Norm (L_2-basierte Dünngitter)

terpunkten lässt sich an Hand der Grafiken ablesen. Der Schwellwert verschiebt sich allerdings mit zunehmender Ordnung nach hinten, für $p = 2$ liegt er bei etwa $5 \cdot 10^2$ Gitterpunkten, für $p = 4$ bei ungefähr 10^4 Gitterpunkten.

Insgesamt ist es weniger angebracht, das Dünngitterverfahren einer bestimmten Ordnung in Konkurrenz zum Vollgitterverfahren der gleichen Ordnung zu betrachten. Vielmehr entspricht einem Dünngitterverfahren der Ordnung p ein Vollgitterverfahren der Ordnung $d \cdot p$, da beide bezüglich des Fehlers in der Energienorm das gleiche asymptotische Verhalten aufweisen, siehe Gleichung (3.50) und den Kommentar in der Zusammenfassung auf Seite 43. Dies ist an Hand der Fehlerkurven auf der linken Seite von Abbildung 5.5 schön nachzuvollziehen: Man vergleiche etwa den Dünngitterfehler zu $p = 1$ mit dem Vollgitterfehler für $p = 2$ oder den Dünngitterfehler bei $p = 2$ mit dem Vollgitterfehler für $p = 4$.

5.2.2 Ein 3D-Beispiel

Wenden wir uns dem folgenden 3D-Beispiel zu:

Beispiel 5.2

$$-\nabla(C\nabla u) = f \qquad \text{in } \Omega = [0,1]^3, \tag{5.9}$$

$$u(x,y) = \cos(5x)\cos(y)\cos(3z) + \cos(3x)\cos(5y)\cos(4z) \qquad \text{auf } \partial\Omega \tag{5.10}$$

mit dem Diffusionskoeffizienten

$$C(x,y,z) = E + \frac{1}{4} \cdot S(x,y,z),$$

$$S(x,y,z) = \begin{pmatrix} \sin(2x)\sin(3y)\sin(4z) & \sin(2x)\sin(y)\sin(5z) & \sin(x)\sin(2y)\sin(2z) \\ \sin(2x)\sin(y)\sin(5z) & \sin(3x)\sin(2y)\sin(3z) & \sin(4x)\sin(3y)\sin(2z) \\ \sin(x)\sin(2y)\sin(2z) & \sin(4x)\sin(3y)\sin(2z) & \sin(3x)\sin(5y)\sin(2z) \end{pmatrix}.$$

E ist die 3×3-Einheitsmatrix. Die rechte Seite f ist so gewählt, dass die Lösung u im Innern von Ω durch den Ausdruck (5.10) gegeben ist.

Das Gebiet $\Omega = [0,1]^3$ bzw. der Raum $H^1(\Omega)$ wird mit einem Dünngitterelement diskretisiert. Um die variablen Koeffizienten zu behandeln, wird die nach (4.9) modifizierte Bilinearform benutzt. Die Interpolationen der Koeffizienten erfolgt mit Ordnung $\hat{p} = p + 1$, wenn p die Ordnung des Ansatzraums ist.

Zur Verwendung kommen zunächst regelmäßige L_2-basierte Gitter. Auf der linken Seite von Abbildung 5.6 ist der L_2-Fehler gegen die Anzahl der Gitterpunkte aufgetragen. Es fällt auf, dass ab Ordnung $p \geq 4$ keine wesentliche Verbesserung mehr durch ein weiteres Erhöhen des Polynomgrades erzielt werden kann. Erst der Einsatz lokal adaptiver dünner Gitter bringt eine Differenzierung des Fehlerverhaltens für Ordnungen $p > 4$ (rechte Seite von Abbildung 5.6). Der Genauigkeitsgewinn, der durch die lokale Gitterverfeinerungen für Ordnungen $p \geq 3$ erzielt wird, ist beträchtlich.

Dies ist umso bemerkenswerter, als es sich bei der Lösung u um eine glatte Funktion handelt und damit zunächst kein Anlass für die Verwendung adaptiver Techniken besteht. Offensichtlich reagieren Ansatzräume höherer Ordnung empfindlicher auf kleine Gitterkorrekturen. Abbildung 5.7 gibt einen Eindruck von den hier verwendeten, a-posteriori erzeugten Gittern.

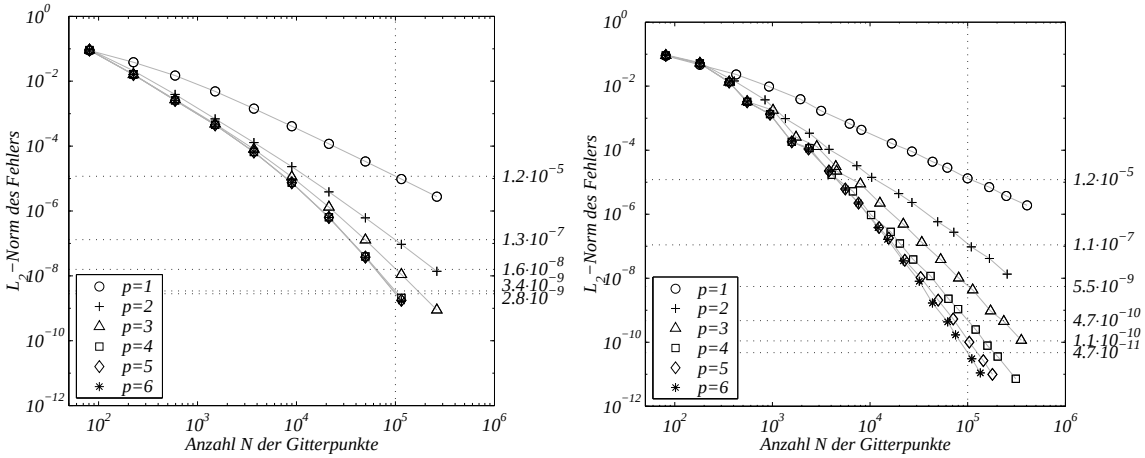


Abbildung 5.6: Beispiel 5.2 – Der Fehler $u - u_h$ gemessen in der L_2-Norm für regelmäßige L_2-basierte dünne Gitter (links) und lokal adaptive L_2-basierte dünne Gitter (rechts)

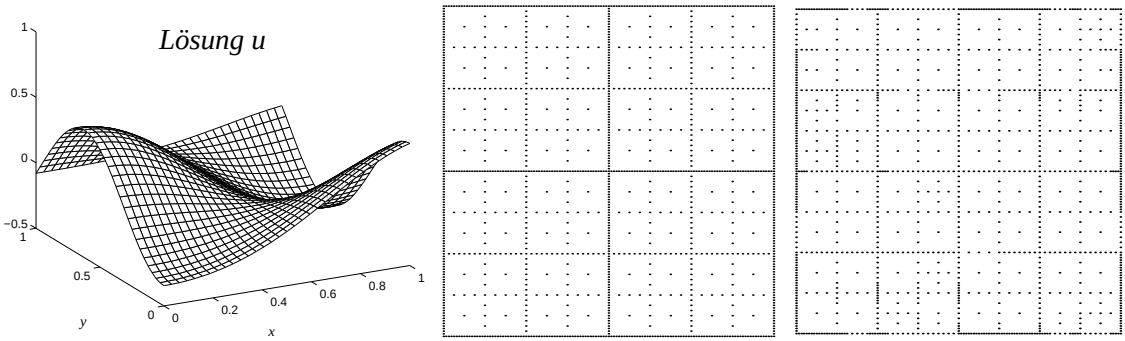


Abbildung 5.7: Beispiel 5.2 – Lösung u, regelmäßiges und adaptives Gitter, jeweils eingeschränkt auf die Ebene $z \equiv 0.5$

Vergleich mit exakt ausgewerteter Bilinearform

Dank der Tensorproduktstruktur der Diffusionskoeffizienten sind wir in der Lage, mit Hilfe der Algorithmen aus Abschnitt 4.2.2 die Bilinearform exakt auszuwerten und damit den durch die Modifikation der Bilinearform bedingten Fehler zu untersuchen. In Tabelle 5.2 sind die Fehler in der L_2-Norm einmal für die modifizierte und einmal für die exakt ausgewertete Bilinearform gegenübergestellt. In beiden Fällen liegen regelmäßige L_2-basierte Gitter zu Grunde. Wie bereits in Beispiel 5.1 festgestellt, ergibt

sich für die exakt ausgewertete Bilinearform erwartungsgemäß der kleinere Fehler, wobei die Differenz zum modifizierten Verfahren umso deutlicher ist, je größer die Ordnung des Ansatzraums ist.

N	$\|u-u_h^{\text{modif}}\|_{L_2}$	$\|u-u_h^{\text{exakt}}\|_{L_2}$	q_h	$\|u-u_h^{\text{modif}}\|_{L_2}$	$\|u-u_h^{\text{exakt}}\|_{L_2}$	q_h
	$p=1$			$p=2$		
81	$8.8691\cdot10^{-2}$	$8.8170\cdot10^{-2}$	1.01	$9.4896\cdot10^{-2}$	$9.3042\cdot10^{-2}$	1.02
225	$3.8245\cdot10^{-2}$	$3.7667\cdot10^{-2}$	1.02	$2.1033\cdot10^{-2}$	$2.1783\cdot10^{-2}$	0.97
593	$1.4822\cdot10^{-2}$	$1.5288\cdot10^{-2}$	0.97	$3.9160\cdot10^{-3}$	$4.4651\cdot10^{-3}$	0.88
1505	$4.79655\cdot10^{-3}$	$5.2205\cdot10^{-3}$	0.92	$6.8828\cdot10^{-4}$	$6.6507\cdot10^{-4}$	1.03
3713	$1.41457\cdot10^{-3}$	$1.6152\cdot10^{-3}$	0.88	$1.2785\cdot10^{-4}$	$9.2687\cdot10^{-5}$	1.38
8961	$4.05402\cdot10^{-4}$	$4.8119\cdot10^{-4}$	0.84	$2.3192\cdot10^{-5}$	$1.3145\cdot10^{-5}$	1.76
21249	$1.15877\cdot10^{-4}$	$1.4147\cdot10^{-4}$	0.82	$3.8662\cdot10^{-6}$	$1.8774\cdot10^{-6}$	2.06
49665	$3.32640\cdot10^{-5}$	$4.1355\cdot10^{-5}$	0.80	$6.1044\cdot10^{-7}$	$2.6621\cdot10^{-7}$	2.29
114689	$9.58371\cdot10^{-6}$	$1.2035\cdot10^{-5}$	0.80	$9.3156\cdot10^{-8}$	$3.7463\cdot10^{-8}$	2.49
	$p=3$			$p=4$		
81	$9.0834\cdot10^{-2}$	$8.9056\cdot10^{-2}$	1.02	$9.0830\cdot10^{-2}$	$8.9060\cdot10^{-2}$	1.02
225	$1.6071\cdot10^{-2}$	$1.7041\cdot10^{-2}$	0.94	$1.6038\cdot10^{-2}$	$1.6949\cdot10^{-2}$	0.95
593	$2.7044\cdot10^{-3}$	$2.8133\cdot10^{-3}$	0.96	$2.5695\cdot10^{-3}$	$2.6696\cdot10^{-3}$	0.96
1505	$4.7874\cdot10^{-4}$	$3.6952\cdot10^{-4}$	1.30	$4.5205\cdot10^{-4}$	$3.0134\cdot10^{-4}$	1.50
3713	$8.0492\cdot10^{-5}$	$3.5894\cdot10^{-5}$	2.24	$6.5904\cdot10^{-5}$	$2.5097\cdot10^{-5}$	2.63
8961	$1.1191\cdot10^{-5}$	$2.6856\cdot10^{-6}$	4.17	$7.4458\cdot10^{-6}$	$1.6002\cdot10^{-6}$	4.65
21249	$1.3111\cdot10^{-6}$	$1.7328\cdot10^{-7}$	7.57	$6.3876\cdot10^{-7}$	$8.1212\cdot10^{-8}$	7.87
49665	$1.2810\cdot10^{-7}$	$1.0475\cdot10^{-8}$	12.2	$3.9337\cdot10^{-8}$	$3.2549\cdot10^{-9}$	12.1
114689	$1.1014\cdot10^{-8}$	$6.1574\cdot10^{-10}$	17.9	$2.0773\cdot10^{-9}$	$1.1586\cdot10^{-10}$	17.9

Tabelle 5.2: Beispiel 5.2 – Vergleich des L_2-Fehlers bei exakt ausgewerteter Bilinearform und bei modifizierter Bilinearform. N ist die Anzahl der Gitterpunkte, q_h ist der in (5.8) definierte Quotient.

Vergleich mit FEM über Tetraedergitter (h-Version)

Wie bereits für das 2D-Beispiel durchgeführt, soll an dieser Stelle ein Vergleich der Dünngittermethode mit einer klassischen Finite-Element-Diskretisierung angestellt werden. Hierzu wurde das Programm `FEMLAB` benutzt[1]. Zur Verwendung kamen Tetraederelemente vom Lagrange-Typ. Abbildung 5.8 zeigt die so erhaltenen Fehlerkurven zusammen mit den Ergebnissen aus den Dünngitter-Rechnungen.

Der Vorteil der Dünngitterelemente gegenüber den Tetraederelementen tritt sehr deutlich hervor. Betrachten wir die Kurven zum Fehler in der L_2-Norm, so ist beispielsweise die Dünngitterdiskretisierung der Ordnung $p=2$ den Tetraederelementen

[1] `FEMLAB`® ist eine FE-Bibliothek für `MATLAB`® und wurde von COMSOL AB, Schweden, entwickelt. Für weitere Informationen sei auf die Homepage `http://www.femlab.com` verwiesen.

der Ordnung $p = 4$ bereits ab ca. 10^4 Gitterpunkten überlegen, eine nach heutigen Maßstäben sehr niedrige Gitterpunktzahl für 3D-Berechnungen.

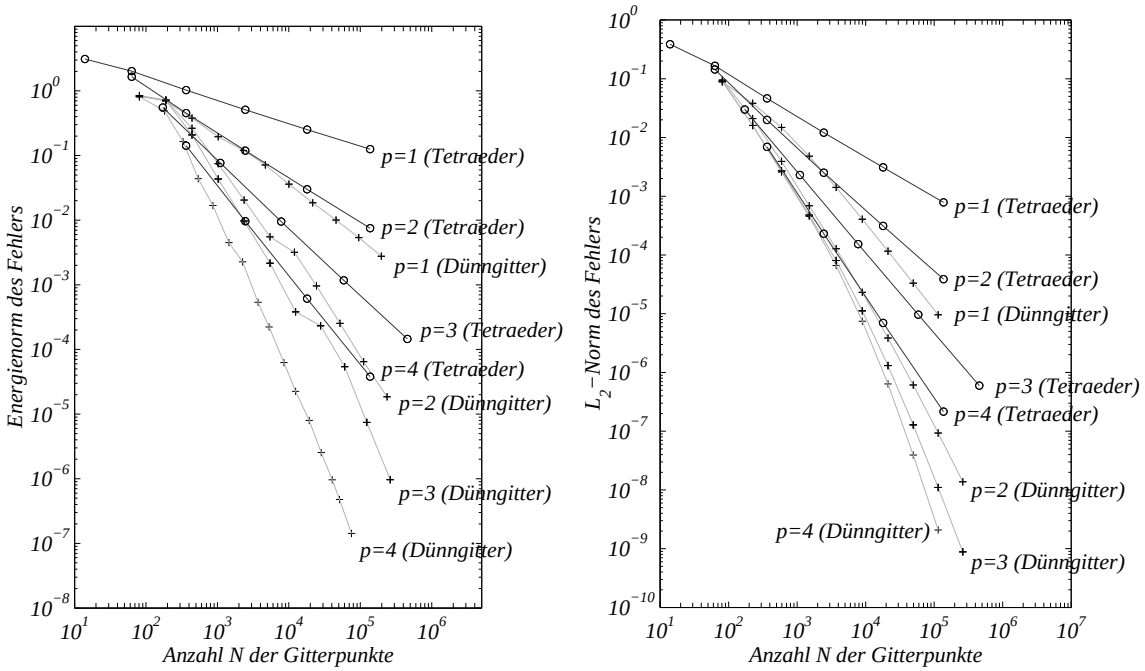


Abbildung 5.8: Beispiel 5.2 – Vergleich von Dünngitterelementen mit FE-Diskretisierung über Tetraedergitter (Lagrange-Elemente, h-Version der FEM). Links: Fehler in Energienorm (energiebasierte Dünngitter), rechts: Fehler in L_2-Norm (L_2-basierte Dünngitter)

5.2.3 Unstetige Koeffizienten

Voraussetzung für die Definition (4.9) der modifizierten Bilinearform $\mathcal{A}_h$ war, dass die Koeffizienten partiell differenzierbar sind. Tatsächlich genügt es, wenn die Differenzierbarkeit im Innern der Elemente gegeben ist und am Rand der Elemente die einseitigen Ableitungen existieren. Somit können Randwertprobleme mit unstetigen Koeffizienten behandelt werden, wenn es gelingt, das Gebiet so in Elemente zu unterteilen, dass die Unstetigkeiten auf den Elementgrenzen zu liegen kommen und die Koeffizienten im Innern der Elemente glatt sind. Betrachten wir hierzu das folgende Beispiel:

Beispiel 5.3 *Die Funktion*

$$u(x,y) \;:=\; u_1(x)u_2(y),$$

$$u_1(x) \;:=\; \begin{cases} \dfrac{2\sinh x}{21\sinh 2}\,, & 0 \le x \le 1\,, \\[2ex] \dfrac{21\sinh x \;-\; 19\sinh(2-x)}{21\sinh 2}\,, & 1 < x \le 2\,, \end{cases}$$

$$u_2(y) \;:=\; \begin{cases} \dfrac{2\sin y}{11\sin 2}\,, & 0 \le y \le 1\,, \\[2ex] \dfrac{11\sin y \;-\; 9\sin(2-y)}{11\sin 2}\,, & 1 < y \le 2\,, \end{cases}$$

genügt der partiellen Differentialgleichung

$$-\nabla(D(x,y)\nabla u(x,y)) \;=\; 0, \quad (x,y) \in \Omega = [0,2]^2\,,$$

wobei D der in Abbildung 5.9 skizzierte, stückweise konstante Diffusionskoeffizient ist. Für das Randwertproblem werden die exakten Werte von u auf dem Rand vorgegeben.

Entsprechend der Gestalt von D wird Ω in vier quadratische Elemente zerlegt. In Abbildung 5.10 werden Ergebnisse aus der Berechnung über energiebasierten, adaptiven Gitter gezeigt. Das feinste Elementgitter wird bei der adaptiven Verfeinerung dort erzeugt, wo die Krümmung der Lösung am größten ist, also dort, wo der Diffusionskoeffizient am kleinsten ist.

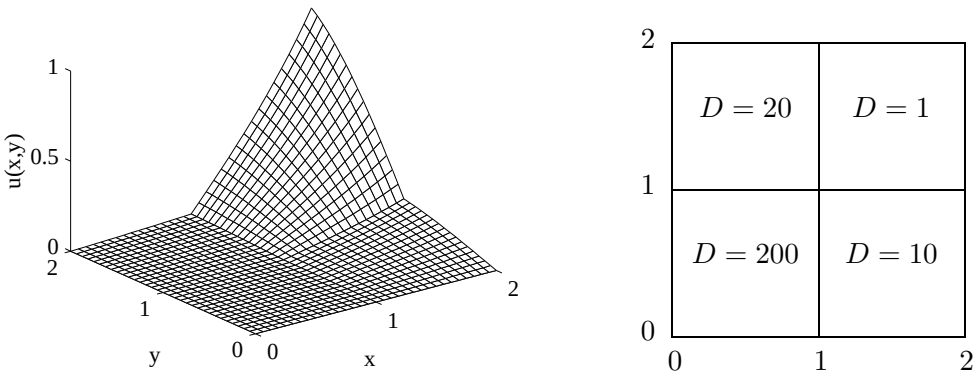


Abbildung 5.9: Beispiel 5.3 – Lösung u (links) und stückweise konstanter Diffusionskoeffizient D (rechts).

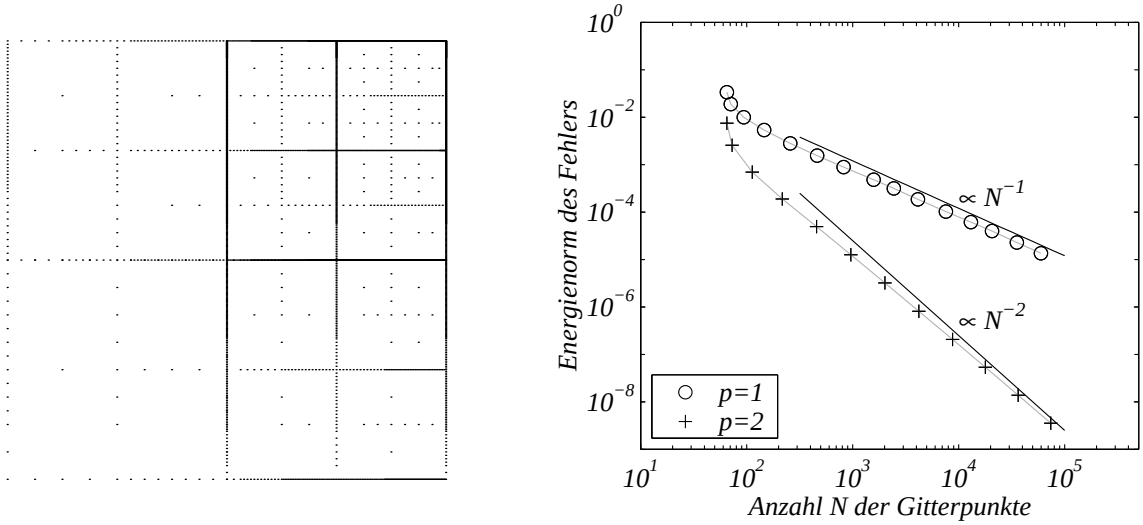


Abbildung 5.10: Beispiel 5.3 – Links: adaptives, energiebasiertes Gitter (3578 Gitterpunkte), rechts: Fehlerkurven für stückweise lineare und stückweise quadratische Ansatzfunktionen

5.3 Krumm berandete Gebiete

Im Folgenden wenden wir uns Beispielen für Randwertprobleme über krumm berandeten Gebieten zu. Die folgenden Berechnungen stützen sich auf die durch (4.9) definierte, modifizierte Bilinearform, wobei die auf das Referenzelement transformierten Koeffizienten gemäß (4.11) gebildet werden.

5.3.1 Potentialströmung um einen Kreiszylinder (2D)

Beispiel 5.4 *Über*

$$\Omega \;=\; [-1,1]^2 \;\cap\; \{(x,y)\in\mathbb{R}^2 :\; x^2+y^2 \geq R^2\}, \qquad R=\frac{1}{4}\,,$$

sei das folgende Randwertproblem definiert:

$$\begin{aligned} -\Delta u &= 0 && \text{in } \Omega, \\ u(x,y) &= y-\frac{R^2 y}{x^2+y^2} && \text{auf } \partial\Omega. \end{aligned} \tag{5.11}$$

Die Lösung $u(x,y)$ ist dann durch den Ausdruck (5.11) für alle $(x,y)\in\Omega$ gegeben.

Dieses Beispiel entstammt einer Anwendung aus der Strömungsmechanik. Stationäre, inkompressible Strömungen, die zudem noch als reibungsfrei angenommen werden, lassen sich durch eine *harmonische* Funktion $\phi : \Omega \to \mathbb{R}$ beschreiben, d.h. ϕ genügt der Laplace-Gleichung [44]. Das Geschwindigkeitsfeld der Strömung ist dann durch den Gradienten von ϕ gegeben. Man bezeichnet ϕ deshalb als *Potential* der Strömung und Strömungen, die sich so beschreiben lassen, als *Potentialströmungen*. Obiges Beispiel beschreibt die Potentialströmung um einen Kreiszylinder mit

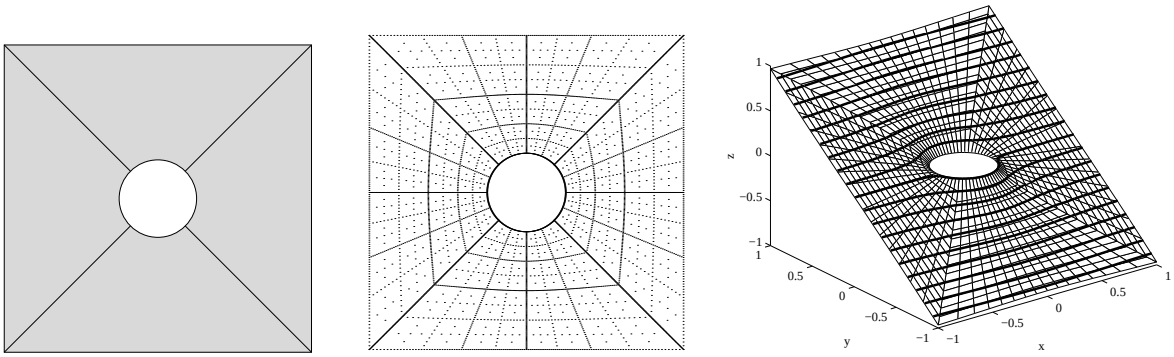


Abbildung 5.11: Beispiel 5.4 – Potentialströmung um einen Kreiszylinder: Zerlegung von Ω in vier Elemente (links), regelmäßige dünne Gitter über den Elementen (Mitte), Stromfunktion u (rechts).

Radius R. Bei der Funktion u handelt es sich allerdings nicht um das eben erwähnte Potential, sondern um die sogenannte *Stromfunktion*, die in 2D ein äquivalentes Gegenstück zum Potential darstellt und ebenfalls der Laplace-Gleichung genügt. Der Name rührt daher, dass die Linien gleichen Funktionswerts parallel zur Strömung verlaufen. Dies gilt im Speziellen für die Begrenzungslinien der Strömung, an denen die Geschwindigkeit konstant 0 ist. Man vergewissere sich, dass dies im obigen Beispiel durch die Dirichlet-Randbedingung auf der Kreislinie $x^2 + y^2 = R^2$ erzwungen wird. Die übrigen Randbedingungen wurden allerdings entsprechend der für freie Ränder bekannten Lösungsfunktion gesetzt, d.h. die Ränder des Quadrats $[-1, 1]^2$ haben hier nicht die Bedeutung von die Strömung begrenzenden Wänden.

Für die numerische Lösung des Randwertproblems wird Ω wie in Abbildung 5.11 gezeigt in vier Elemente zerlegt. Die Transformationen für die Abbildung des Referenzelements auf die Elemente werden durch Vorgabe der Randparametrisierung und transfiniter Interpolation im Innern erzeugt. Die Elemente selbst werden zunächst mit regelmäßigen, L_2-basierten dünnen Gittern versehen. Zur Verwendung kommt die nach (4.9) und (4.11) modifizierte Bilinearform. Ist p die Ordnung des Ansatzraums, so werden die Transformationen entsprechend der Empfehlung (4.12) mit Ordnung $\tilde{p} = p+2$ interpoliert, die transformierten Koeffizienten mit Ordnung $\hat{p} = p + 1$.

In Abbildung 5.12 sind die Fehlerkurven für verschiedene Ordnungen p dokumentiert. Deutlich zu erkennen ist das mit steigender Ordnung besser werdende Approximationsverhalten, wobei der Genauigkeitsgewinn beim Übergang von $p = 1$ auf $p = 2$ wesentlich größer ist als etwa von $p = 4$ auf $p = 5$. Für hohe Ordnungen wird das asymptotische Verhalten offensichtlich erst nach einer beträchtlich längeren Vorlaufzeit angenommen, als dies bei niedrigen Ordnungen der Fall ist. Diese Beobachtung haben wir bereits bei Beispiel 5.2 gemacht. Dort konnte die erwähnte Vorlaufzeit durch die Verwendung adaptiver Gitter erheblich verringert werden. Das gleiche gilt nun auch für das vorliegende Beispiel. In Abbildung 5.13 sind die Fehlerkurven für regelmäßige und adaptive L_2-basierte Gitter gegenübergestellt. Für $p = 1$ produziert das adaptive Gitter keine besseren Ergebnisse als das regelmäßige Gitter. Bemerkenswert ist hinge-

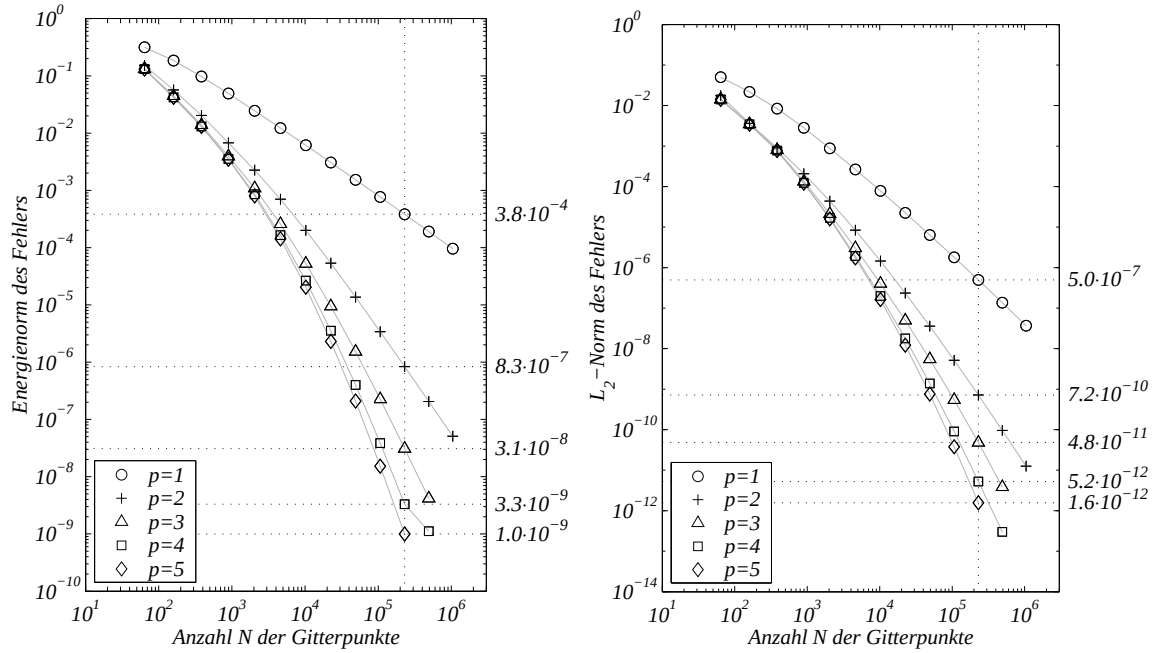


Abbildung 5.12: Beispiel 5.4 – Potentialströmung um einen Kreiszylinder: Fehler gemessen in der Energie-Norm (links) und in der L_2-Norm (rechts) für regelmäßige, L_2-basierte Dünngitterräume $V_n^{(p,1)}$. Die Interpolation der Transformationen fand mit Ordnung $\tilde{p} = p + 2$ statt, die der Koeffizienten mit $\hat{p} = p + 1$.

gen der Vorsprung, der durch die Gitter-Adaption für Ordnungen $p \geq 2$ erreicht wird. Für $p = 4$ etwa ist der Fehler auf dem adaptiven Gitter mit ca. 10^5 Gitterpunkten knapp zwei Zehnerpotenzen kleiner als auf dem vergleichbaren regelmäßigen Gitter. Die Approximationsgüte dünner Gitter wird von der gemischten Ableitung $(p+1)$-ter Ordnung beeinflusst. In der Nähe des Zylinders ist auf Grund der Koordinatentransformation die gemischte Ableitung bezüglich der Polarkoordinaten r und ϕ zu betrachten. Es ist $\partial_r^{p+1}\partial_\phi^{p+1}u = O(r^{-(p+2)})$. Damit ist klar, dass das Gitter in der Umgebung der Zylinderoberfläche umso stärker verfeinert werden muss, je größer die Ordnung p ist. Einen Eindruck von den hier erzeugten lokal adaptiven Gittern gibt Abbildung 5.14.

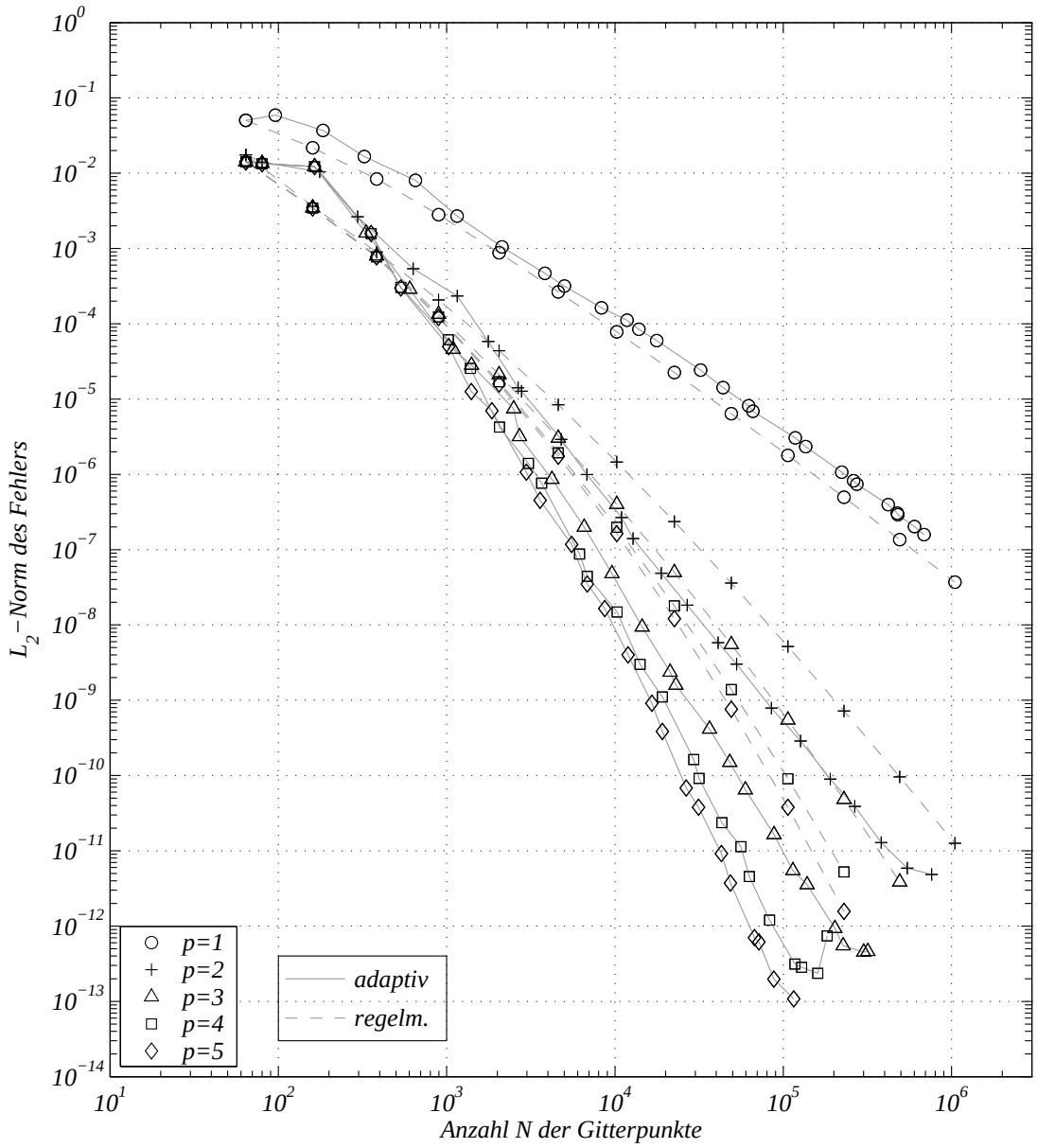


Abbildung 5.13: Beispiel 5.4 – Potentialströmung um einen Kreiszylinder: Vergleich von regelmäßigen und adaptierten L_2-basierten Dünngitterelementen.

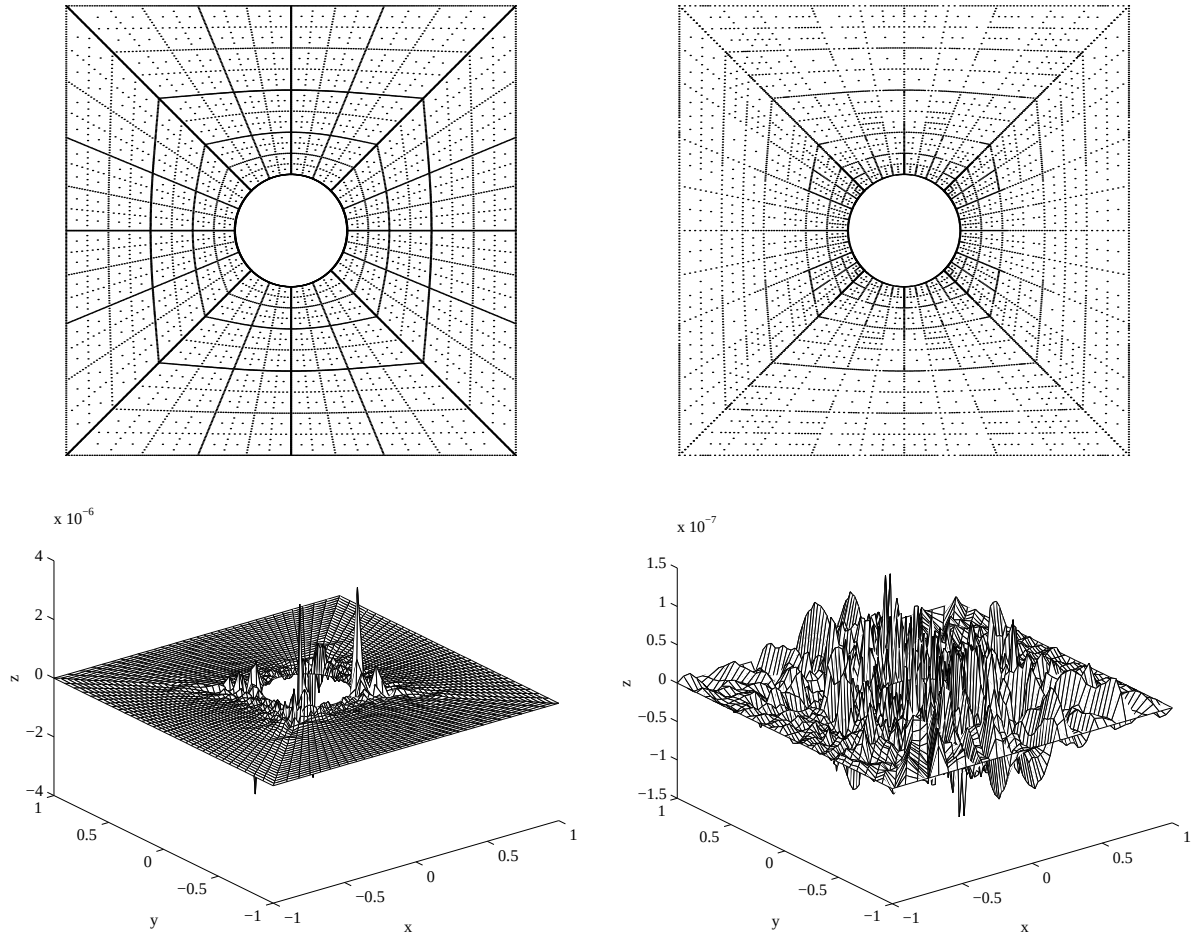


Abbildung 5.14: Beispiel 5.4 – Potentialströmung um einen Kreiszylinder: Oben: regelmäßiges (links) und adaptives (rechts) L_2-basiertes Dünngitter ($p = 4$) mit 10240 bzw. 6850 Gitterpunkten. Unten: Fehler über dem entsprechenden Gitter.

Zur Wahl von $\tilde{p}$ und $\hat{p}$

Die Empfehlung, die Transformationen mit Ordnung $p+2$ und die Koeffizienten mit Ordnung $p+1$ zu interpolieren, wenn p die Ordnung des Ansatzraums ist, basiert auf der Überlegung, dass der jeweilige Interpolant zwei- bzw. einmal differenziert wird, bevor er (an den Gitterpunkten) ausgewertet wird und dadurch in der Regel zwei bzw. eine Ordnung an Genauigkeit verloren geht. Abbildung 5.15 zeigt an Hand des vorliegenden Beispiels, inwiefern sich verschiedene Kombinationen von p, $\hat{p}$ und $\tilde{p}$ auf das Konvergenzverhalten auswirken. Dargestellt ist jeweils der Fehler in der Energienorm für verschiedene Tripel $(p, \hat{p}, \tilde{p})$ und als Vergleichsmaßstab der Interpolationsfehler, gemessen auf dem jeweils nächstfeineren Gitter. Für $p = 1$ und $p = 2$ ist dabei deutlich zu erkennen, wie sich der erwähnte Ordnungsverlust auf die Lösung niederschlägt, wenn $\tilde{p} = \hat{p} = p$ gesetzt wird. Die Erhöhung von $\tilde{p}$ und $\hat{p}$ gleicht diesen wieder aus, so dass der Fehler das gleiche asymptotische Verhalten hat wie der Interpolationsfehler. Für $p \geq 3$ scheint keine Ordnungserhöhung nötig zu sein. Für den L_2-Fehler (Abbildung 5.16) ergibt sich aber auch hier die Notwendigkeit einer Ordnungserhöhung, um die gewünschte Asymptotik zu erzielen. Man beachte, dass die Erhöhung der Ordnung bei den Interpolationen keinen wesentlichen zusätzlichen Aufwand bedeutet: Die erwähnten Interpolationen sind nur einmal pro Triangulierung auszuwerten.

Vergleich mit Standard Lagrange-Dreiecks-Elementen

An dieser Stelle soll ein Vergleich mit der h-Methode der Finiten Elemente gezogen werden. Dafür wurde das Problem 5.4 mit (isoparametrischen) Dreieckselementen vom Lagrange-Typ gerechnet. Zum Einsatz kam die FE-Bibliothek `FEMLAB`. Abbildung 5.17 zeigt eine typische Triangulierung für das Gebiet Ω. Daneben sind die L_2-Fehler für L_2-basierte Dünngitterelemente und für die Dreieckselemente gegenübergestellt. Der Komplexitätsvorsprung der dünnen Gitter tritt dabei deutlich hervor. So liefert das Dünngitterverfahren für $p = 2$ im Vergleich zum Vollgitterverfahren für $p = 3$ bereits ab $N \approx 2 \cdot 10^3$ Gitterpunkte den kleineren Fehler. Das Dünngitterverfahren der Ordnung 1 weist in etwa das gleiche asymptotische Verhalten auf, wie die Dreieckselemente dritter Ordnung: Für letztere sind die Fehler von der Ordnung $O(N^{-(3+1)/2}) = O(N^{-2})$, für die Dünngitterelemente darf man nach (3.27) Fehler der Ordnung $O(N^{-2}|\log N|^3) \approx O(N^{-2})$ erwarten.

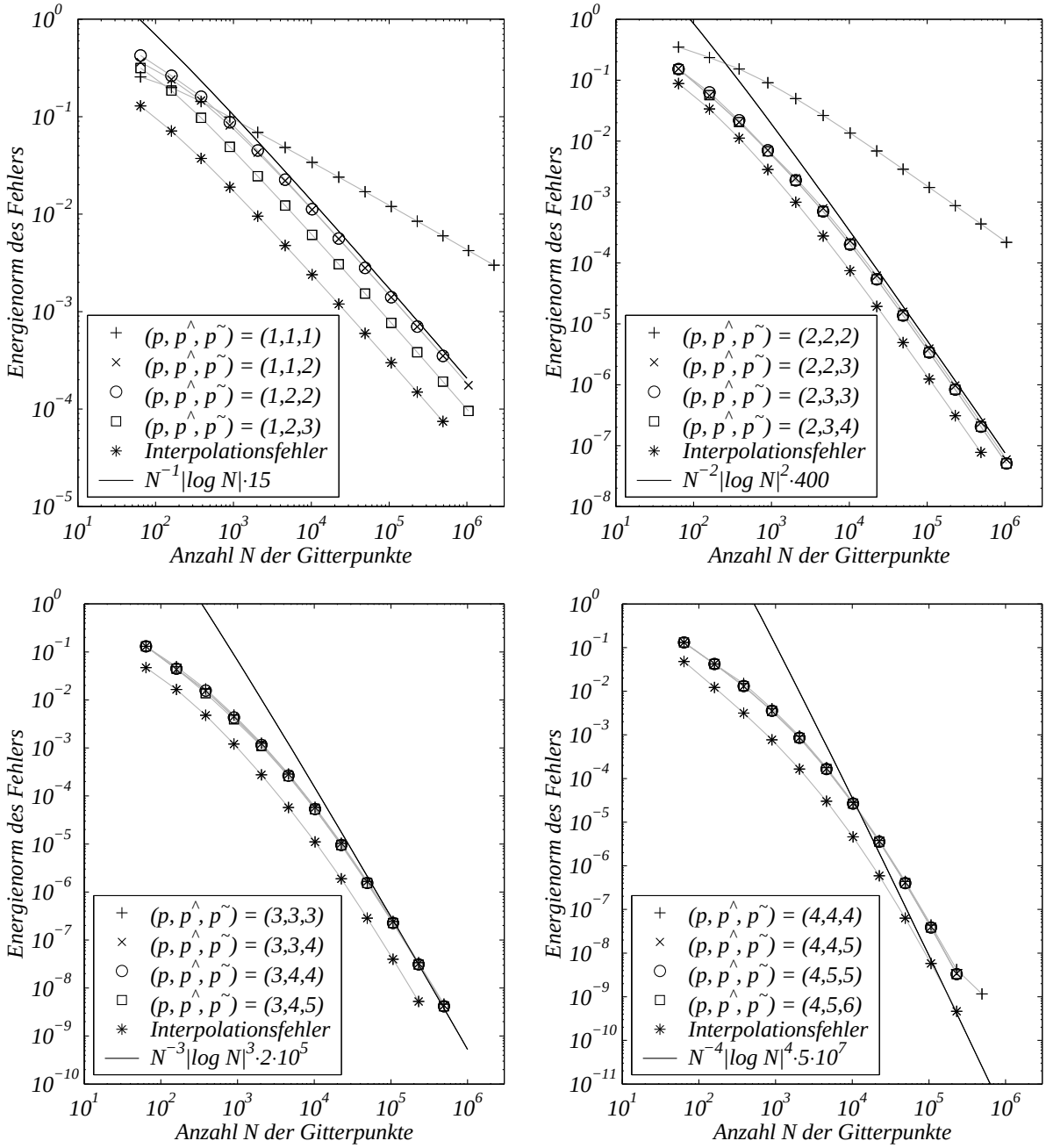


Abbildung 5.15: Beispiel 5.4 – Potentialströmung um einen Kreiszylinder: Fehler gemessen in der Energie-Norm für verschiedene Kombinationen von p, $\hat{p}$ und $\tilde{p}$. Zum Vergleich ist der Interpolationsfehler $\|u - I_h^{(p)} u\|_E$, gemessen auf dem nächstfeineren Gitter, eingetragen.

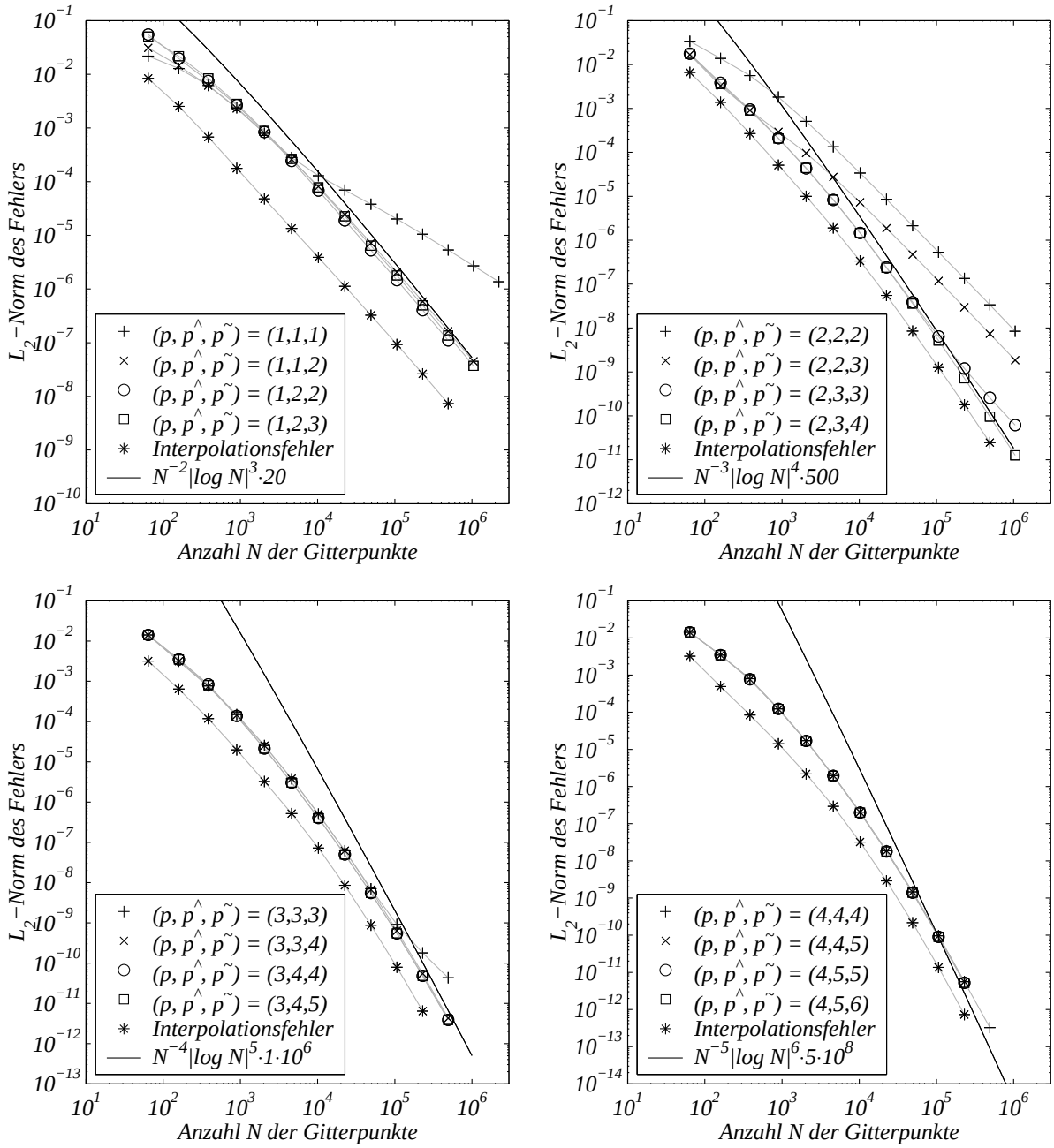


Abbildung 5.16: Beispiel 5.4 – Potentialströmung um einen Kreiszylinder: Fehler gemessen in der L_2-Norm für verschiedene Kombinationen von p, $\hat{p}$ und $\tilde{p}$. Zum Vergleich ist der Interpolationsfehler $\|u - I_h^{(p)} u\|_{L_2}$, gemessen auf dem nächstfeineren Gitter, eingetragen.

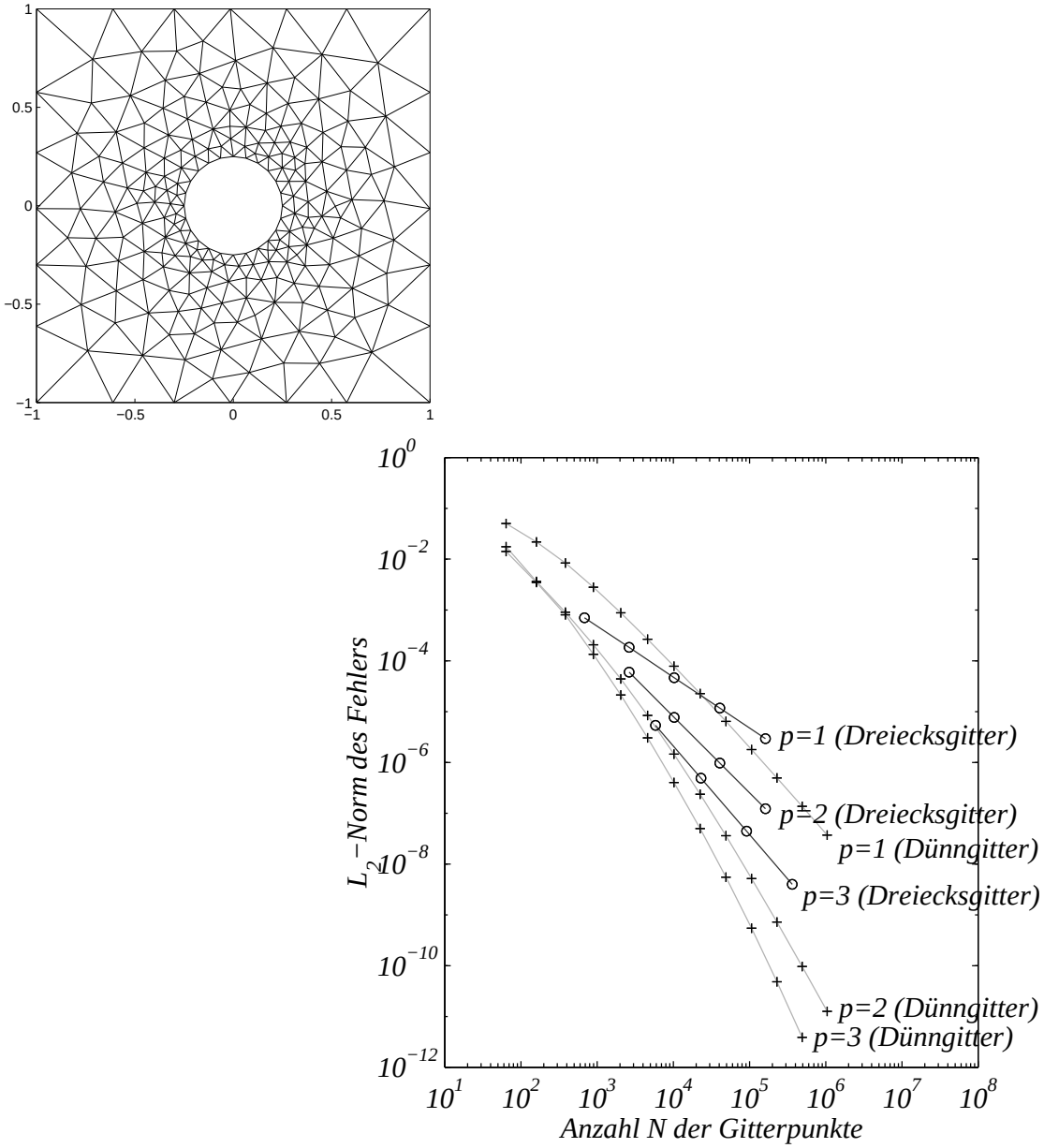


Abbildung 5.17: Beispiel 5.4 – Potentialströmung um einen Kreiszylinder: Vergleich mit Lagrange-Elementen über Dreiecksgitter.

5.3.2 Ein 3D-Beispiel

Beispiel 5.5 *Über*

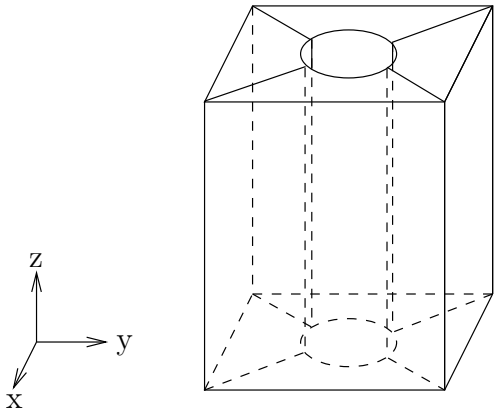
$$\Omega \;=\; [-1,1]^3 \;\cap\; \{(x,y,z) \in \mathbb{R}^2 :\; x^2+y^2 \geq R^2\}, \qquad R = \frac{1}{4}\,,$$

sei das folgende Randwertproblem gegeben:

$$\begin{aligned} -\Delta u &= 0 && \text{in } \Omega, \\ u(x,y,z) &= \frac{1}{\sqrt{x^2+y^2+z^2}} && \text{auf } \partial\Omega. \end{aligned} \tag{5.12}$$

Die Lösung $u(x,y,z)$ im Innern von Ω ist dann ebenfalls durch den Ausdruck (5.12) bestimmt.

Bei dem Gebiet Ω handelt es sich um einen Würfel, aus dem ein Zylinder ausgeschnitten ist. Es wird entsprechend der folgenden Skizze in vier Teilgebiete zerlegt:



Die Teilgebiete werden mit je einem Dünngitterelement diskretisiert. In Abbildung 5.18 sind die Ergebnisse aus den numerischen Berechnungen zusammengetragen. Erneut ist festzustellen, dass für Ordnungen $p \geq 2$ die adaptiven Dünngitter wesentlich bessere Ergebnisse liefern als die regelmäßigen Gitter mit gleicher Knotenanzahl. Das vom Interpolationsfehler a-priori bekannte asymptotische Approximationsverhalten der Art $N^{-(p+1)}|\log N|^{s(p,d)}$ wird allerdings auch von den adaptiven Gittern nicht erreicht. So fallen die L_2-Fehler für $p \geq 3$ nur wie N^{-3}. Trotzdem ist dieses Fehlerverhalten gut: Um das gleiche Abfallverhalten von einem Vollgitterverfahren zu erhalten, müsste dieses mit Elementen 8-ter Ordnung arbeiten ($\frac{8+1}{3} = 3$). Zum Vergleich sind in Abbildung 5.18 auch Berechnungen mit Tetraederelementen erster und zweiter Ordnung aufgenommen (gerechnet mit `FEMLAB`). Im Vergleich zu den Tetraederelementen zweiter Ordnung liefern die regelmäßigen Dünngitter erster Ordnung ab ca. 70.000 Punkten, die adaptiven Dünngitter erster Ordnung bereits ab ca. 15.000 Punkten den kleineren Fehler.

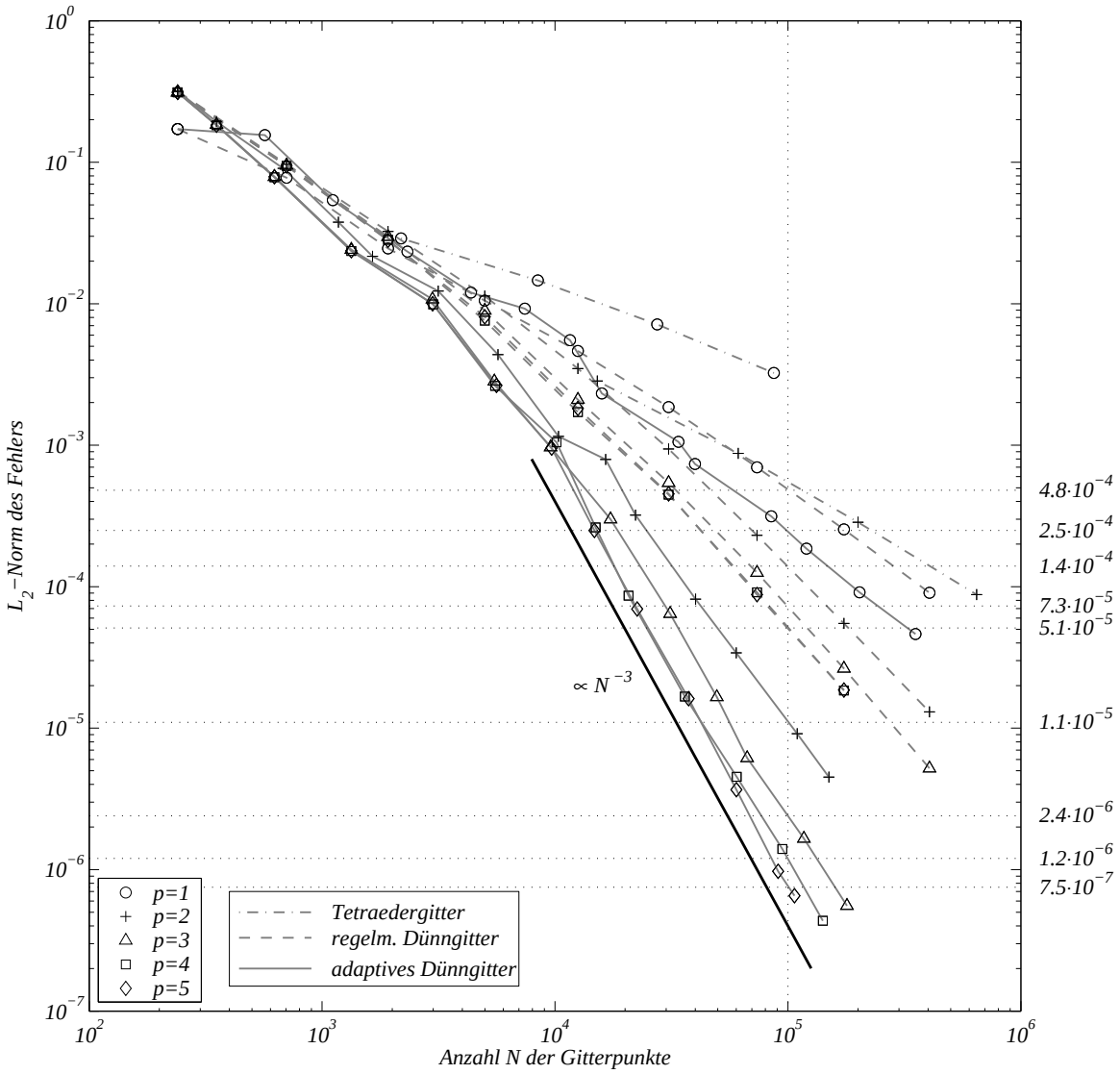


Abbildung 5.18: Beispiel 5.5 – Vergleich von Tetraederelementen, regelmäßigen und adaptiven L_2-basierten Dünngitterelementen.

6 Ein Mehrgitter-Vorkonditionierer

Mehrgitterverfahren [29, 30, 49] beruhen auf der Idee, das Gleichungssystem auf einer Hierarchie von Unterräumen näherungsweise zu lösen. Zur Konstruktion solcher Hierarchien ist die hierarchische Basis ein besonders geeignetes Instrument. Bereits der Basiswechsel von der nodalen in die hierarchische Basis verringert die Konditionszahl der Systemmatrix essentiell [55]. Vorkonditionierer auf der Grundlage von hierarchischen Basen, die zu einer von der Gitterweite unabhängigen Konditionszahl führen, werden 1986 von Yserentant [54] vorgestellt. Griebel und Oswald [26] geben 1994 einen additiven Schwarz-Vorkonditionierer an, der für die Finite-Element-Approximation der Poissongleichung über dünnen Gittern eine optimale, d.h. gitterweitenunabhängige Konvergenzrate aufweist. Für Dünngitterelemente höherer Ordnung stellt Bungartz in [16] einen Mehrgitter-Algorithmus vor. Allerdings werden dort die Testfunktionen nach wie vor als stückweise linear vorausgesetzt, es wird also ein Petrov-Galerkin-Ansatz verfolgt. Dadurch vereinfacht sich zwar der Transport des Residuums zwischen den verschiedenen Gittern, auf das theoretische Fundament der in Ansatz- und Testraum symmetrischen Ritz-Galerkin-Verfahren muss allerdings verzichtet werden.

Das auf den folgenden Seiten vorgestellte Mehrgitterverfahren ist in Ansatz- und Testraum symmetrisch. Als Vorkonditionierer für das `BiCGstab`-Verfahren [11, 51] hat es in den numerischen Beispielen für sehr gute Konvergenzraten gesorgt.

6.1 Unterraum-Korrektur-Verfahren

Das aus der FE-Diskretisierung hervorgehende Gleichungssystem

$$Au \;=\; b \tag{6.1}$$

mit der Lösung $u \in V$ wird auf Grund seiner Größe in der Regel mit Hilfe iterativer Verfahren gelöst. Solche Löser bestimmen ausgehend von einer geschätzten Startlösung $u^{(1)}$ eine Folge von Näherungen $u^{(2)},\ u^{(3)}, \ldots$, die gegen die Lösung u konvergiert.

Den abstrakten Rahmen für eine Reihe von iterativen Lösern bilden die *Unterraum-Korrektur-Verfahren* [53]: Ausgangspunkt ist eine Zerlegung von V in eine nicht notwendigerweise direkte Summe

$$V \;=\; \sum_{j=1}^{J} V_j \tag{6.2}$$

von Unterräumen $V_j \subset V$, $j = 1, \ldots, J$, die durch Abbildungen $P_j : V_j \to V$ in V eingebettet sind. Sei V_j, $j \in \{1, \ldots, J\}$, ein beliebig herausgegriffener Unterraum. Gesucht ist nun eine Korrektur $c_j^{(i)} \in V_j$, so dass bestenfalls

$$u^{(i+1)} \;=\; u^{(i)} + P_j c_j^{(i)} \;=\; u$$

gilt, also

$$AP_j\, c_j^{(i)} \;=\; Au - Au^{(i)} \;=\; f - Au^{(i)} \;=:\; r^{(i)}\,. \tag{6.3}$$

Das Residuum $r^{(i)}$ liegt im Allgemeinen jedoch nicht im Bild von AP_j. Man fordert deshalb lediglich, dass die Projektion von (6.3) auf V_j erfüllt ist:

$$P_j^T AP_j\, c_j^{(i)} \;=\; P_j^T r^{(i)}\,. \tag{6.4}$$

Ist $P_j^T AP_j$ invertierbar, so ist die Korrektur $c_j^{(i)}$ gegeben durch

$$c_j^{(i)} \;=\; (P_j^T AP_j)^{-1}\, P_j^T\, r^{(i)} \;=:\; B_j r^{(i)}\,.$$

Es ist im Allgemeinen $u^{(i+1)} \neq u$, doch man hat immerhin $P_j^T r^{(i+1)} = 0$.

Bei der Kombination der Korrekturen aus den verschiedenen Unterräumen V_j, $j = 1, \ldots, J$, unterscheidet man zwischen zwei Verfahren: Das *additive Verfahren* ist gegeben durch

$$u^{(i+1)} \;=\; u^{(i)} \;+\; \omega \cdot \sum_{j=1}^{J} B_j r^{(i)}$$

mit einem optionalen Dämpfungsfaktor $\omega \in [0, 1]$. Es bearbeitet alle Gitter parallel, d.h. die Korrekturen $B_j r^{(i)}$ werden unabhängig von einander berechnet und anschließend zu einer Gesamtkorrektur addiert. Das *multiplikative Verfahren* hingegen macht für die $(j+1)$-te Korrektur vom Ergebnis aus der j-ten Korrektur Gebrauch. Der Schritt von $u^{(i)}$ auf $u^{(i+1)}$ zerfällt somit in J hintereinander auszuführende Teilschritte:

$$u^{(i+\frac{j}{J})} \;=\; u^{(i+\frac{j-1}{J})} \;+\; B_j r^{(i+\frac{j-1}{J})}, \quad j = 1, \ldots, J.$$

Die Kunst in der Gestaltung effizienter Unterraum-Korrektur-Verfahren besteht nun darin, Unterräume V_j und Einbettungen P_j zu finden, so dass die Korrekturen möglichst wirkungsvoll sind. Die auf den Unterräumen gegebenen Gleichungen (6.4) müssen leicht zu lösen sein, bestenfalls mit einem Aufwand von der Ordnung $O(\dim V_j)$. Falls dies nicht möglich ist, muss wenigstens eine gute Näherung für die Unterraum-Korrektur gefunden werden können.

Ein einfaches Beispiel für ein Unterraum-Korrektur-Verfahren ergibt die triviale Aufspaltung von V in eine direkte Summe eindimensionaler Unterräume V_j. Sie führt auf das (gedämpfte) *Jacobi-Verfahren* (additives Verfahren) bzw. auf das *Gauß-Seidel-Verfahren* (multiplikatives Verfahren) [30]. Wird das Berechnungsgebiet Ω in Teilgebiete Ω_j zerlegt und der Unterraum V_j als der lokale Ansatzraum über Ω_j festgelegt, bekommt man die bekannten *Gebietszerlegungsverfahren* [41].

Klassische Mehrgitterverfahren lassen sich ebenfalls als Unterraum-Korrektur-Verfahren interpretieren. Hier ist $V = V_1 \supset V_2 \supset \cdots \supset V_J$, d.h. die V_j bilden eine Folge von ineinander geschachtelten Unterräumen. Dies ist übrigens ein einfaches Beispiel dafür, dass die Summe der V_j keine direkte Summe ist. Für solche rekursiven Unterraum-Korrektur-Verfahren muss nur die Unterraumgleichung im gröbsten Unterraum V_J exakt gelöst werden. Für die V_j mit $j < J$ genügt es, jeweils nur Fehleranteile in dem Quotientenraum V_j/V_{j+1} zu eliminieren, da ja die Fehleranteile in V_{j+1} von der Korrektur auf dem Level $j+1$ getilgt werden. Verfahren, die dieses leisten, heißen *Glätter*.

6.2 Unterraum-Zerlegung für Dünngitterelemente

Wir betrachten zunächst nur das Referenzelement über $\Omega = [0,1]^d$. Gegeben sei ein Funktionenraum V über einem hierarchisch vollständigen Gitter $G \subset \Omega$. Hierarchisch vollständig bedeutet dabei, dass mit jedem Knoten auch seine hierarchischen Vorfahren im Gitter enthalten sind, vergleiche hierzu Abschnitt 3.5. Als Teilräume V_j kommen zunächst alle Funktionenräume über hierarchisch vollständigen Teilgittern $G_j \subset G$ in Frage. Die Einbettungen $P_j : V_j \to V$ sind bezüglich der hierarchischen Basis besonders einfach zu beschreiben: Die hierarchischen Überschüsse von $P_j v$ stimmen an den Gitterpunkten $\boldsymbol{r} \in G_j$ mit denen von v überein, an den Punkten $\boldsymbol{r} \in G \setminus G_j$ sind sie 0. Dann ist $(P_j v)(\boldsymbol{x}) = v(\boldsymbol{x})$ für alle $\boldsymbol{x} \in \Omega$. Eine explizite Interpolation von Funktionswerten, wie sie beim Gebrauch nodaler Basen für den Übergang von einem groben auf ein feineres Gitter nötig ist, entfällt also.

Betrachten wir nun die Operatoren $A_j := P_j^T A P_j$ über den Teilräumen V_j. Die Multiplikation von A_j mit einem $v \in V_j$ kann dadurch geschehen, dass v zunächst via P_j auf das feine Gitter interpoliert wird, dort mit A multipliziert und das Ergebnis mittels P_j^T auf V_j zurückprojeziert wird. Dieses naive Vorgehen ist allerdings nicht von praktischem Nutzen, würde die Auswertung doch einen Rechenaufwand von $O(\dim V)$ an Stelle der gewünschten $O(\dim V_j)$ erfordern. Der exakte Grobgitter-Operator A_j wird deshalb durch die Steifigkeitsmatrix $\tilde{A}_j$ ersetzt, die aus der Diskretisierung des Randwertproblems über V_j stammt. Mit obiger Forderung nach hierarchisch vollständigen Teilgittern wird gewährleistet, dass das Matrix-Vektor-Produkt $\tilde{A}_j v$ mit den Algorithmen aus Abschnitt 3.6 bzw. Abschnitt 4.2 ausgewertet kann. Der Aufwand für eine Auswertung ist $O(\dim V_j)$. Für den Fall, dass bei der Diskretisierung mit der exakten, d.h. insbesondere gitterunabhängigen Bilinearform gearbeitet wird, gilt sogar $\tilde{A}_j = A_j$. Für die nach (4.9) modifizierte Bilinearform kann man erwarten, dass $\tilde{A}_j$ eine gute Näherung für A_j darstellt, sofern die Koeffizienten der Differentialgleichung hinreichend glatt sind.

Mehrgitterschema für regelmäßige dünne Gitter

Nachdem die Einbettungen P_j und die Grobgitter-Operatoren $\tilde{A}_j$ für beliebige hierarchisch vollständige Teilgitter $G_j \subset G$ beschrieben sind, stellt sich die Frage, welche

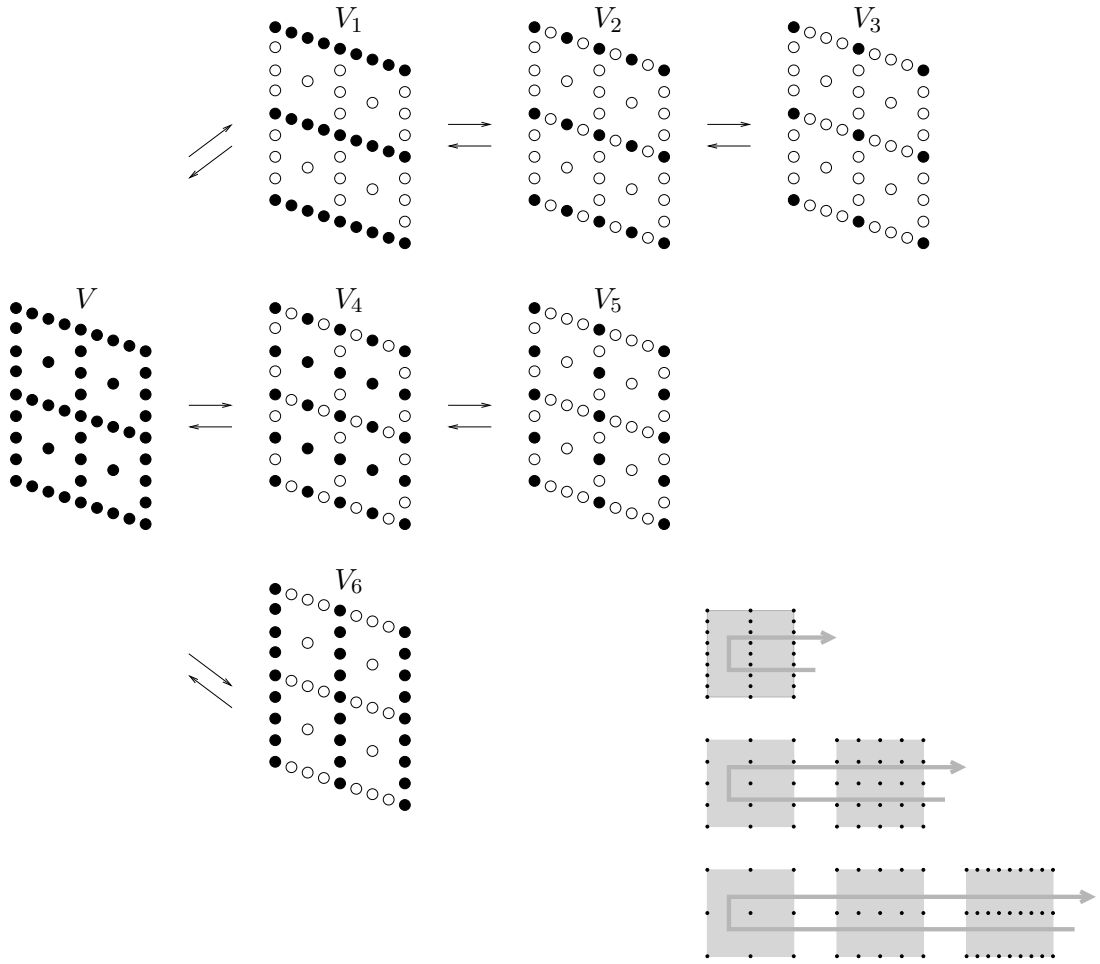


Abbildung 6.1: Zur Unterraum-Korrektur für ein Dünngitterelement: Als Unterräume werden solche mit Vollgitterstruktur ausgewählt; die Pfeile geben die Einbettung der Unterräume und damit die Reihenfolge der Korrekturen an.

Teilgitter und welche Löser für die Teilraumprobleme zu einem effizienten Unterraum-Korrektur-Verfahren führen. In Anlehnung an die Vorarbeiten [26, 16] verwenden wir hier als G_j alle in G enthaltenen Teilgitter mit Vollgitterstruktur, vergleiche Abbildung 6.1. Diese sind teilweise selbst ineinander eingebettet, d.h. es handelt sich hierbei um ein Multi-Level-Splitting des Ansatzraums V. Für zwei ineinander eingebettete Teilräume $V_j \supset V_{j+1}$ genügt es nach der Bemerkung am Ende von Abschnitt 6.1, bei der Lösung der Gleichung über V_j nur Fehleranteile in V_j/V_{j+1} zu eliminieren. Fehleranteile in V_{j+1} werden durch die dort berechnete Korrektur ausgelöscht. Da die Unterräume Vollgitterstruktur besitzen, kann auf dafür bekannte Glätter zurückgegriffen werden.

Der Glätter

Ein für Vollgitterskalen wirkungsvoller und einfach zu implementierender Glätter ist das gedämpfte Jacobi-Verfahren. Hierzu muss die Systemmatrix bezüglich der nodalen, also dehierarchischen Basis betrachtet werden. Ist

$$\boldsymbol{A} \cdot \boldsymbol{u} \;=\; \boldsymbol{b}$$

das Gleichungssystem bezüglich der hierarchischen Basis (in Matrixschreibweise), so lautet das auf die dehierarchische Basis transformierte System

$$\boldsymbol{A}' \cdot \boldsymbol{u}' \;=\; \boldsymbol{b}'$$

mit

$$\begin{aligned} \boldsymbol{A}' &= \boldsymbol{H}^T \boldsymbol{A} \boldsymbol{H}\,, \\ \boldsymbol{u}' &= \boldsymbol{H}^{-1} \boldsymbol{u}\,, \\ \boldsymbol{b}' &= \boldsymbol{H}^T \boldsymbol{b}\,. \end{aligned}$$

Die Matrix $\boldsymbol{H}$ vollzieht dabei den Wechsel von der dehierarchischen in die hierarchische Basis. Die Matrix $\boldsymbol{A'}$ wird nicht explizit gespeichert, die Multiplikation von $\boldsymbol{A'}$ mit einem Vektor wird durch die Multiplikation mit $\boldsymbol{H}$, $\boldsymbol{A}$ und $\boldsymbol{H}^T$ realisiert. Die Algorithmen für die Multiplikation mit $\boldsymbol{H}$, $\boldsymbol{H}^{-1}$ und $\boldsymbol{H}^T$ benötigen wie die Multiplikation mit der Steifigkeitsmatrix $\boldsymbol{A}$ $O(N)$ Rechenoperationen, wenn N die Anzahl der Freiheitsgrade ist. Die Beschreibung der Algorithmen ist in den Abschnitten 3.6.3 und 3.6.5 zu finden. Das gedämpfte Jacobi-Verfahren ist nun gegeben durch die Iterationsvorschrift

$$\boldsymbol{u}'^{(i+1)} \;=\; \boldsymbol{u}'^{(i)} \;+\; \omega \cdot \boldsymbol{D}^{-1} \cdot (\boldsymbol{b}' - \boldsymbol{A}' \cdot \boldsymbol{u}'^{(i)})\,.$$

Hier ist $\boldsymbol{D}$ die Diagonalmatrix, die auf der Diagonalen mit $\boldsymbol{A'}$ übereinstimmt. Der Dämpfungsfaktor wird auf $\omega = 2/3$ gesetzt [30]. Um die Einträge der Diagonalmatrix zu bestimmen, müssen Integrale der Art

$$\int_{[0,1]^d} c_{\boldsymbol{\alpha}\boldsymbol{\beta}}(\boldsymbol{x}) \cdot D^{\boldsymbol{\alpha}} \phi_{\boldsymbol{r}}(\boldsymbol{x}) \cdot D^{\boldsymbol{\beta}} \phi_{\boldsymbol{r}}(\boldsymbol{x})\, d\boldsymbol{x}. \tag{6.5}$$

berechnet werden, wobei $\phi_{\boldsymbol{r}}$ die dem Punkt $\boldsymbol{r}$ zugeordnete nodale Basisfunktion ist. Hierzu werden einige Vereinfachungen vorgenommen:

Zunächst ist festzustellen, dass die nodalen Basisfunktionen höherer Ordnung auf Grund der speziellen Konstruktion der hierarchischen Basis von komplizierterer Gestalt sind als etwa für nodale Lagrange-Elemente, siehe Abbildung 6.2. So besitzt der Träger hier eine Länge von bis zu 2^p Maschenweiten im Gegensatz zu p Maschenweiten beim herkömmlichen Lagrange-Ansatz (p ist der Polynomgrad). Außerhalb der $\pm h$-Umgebung um den zugeordneten Gitterpunkt fällt die Basisfunktion allerdings sehr schnell ab. Lokal betrachtet ähneln die Basisfunktionen für $p > 2$ denjenigen für $p = 2$. Für eine Schätzung der Diagonaleinträge für Dünngitterelemente der Ordnung

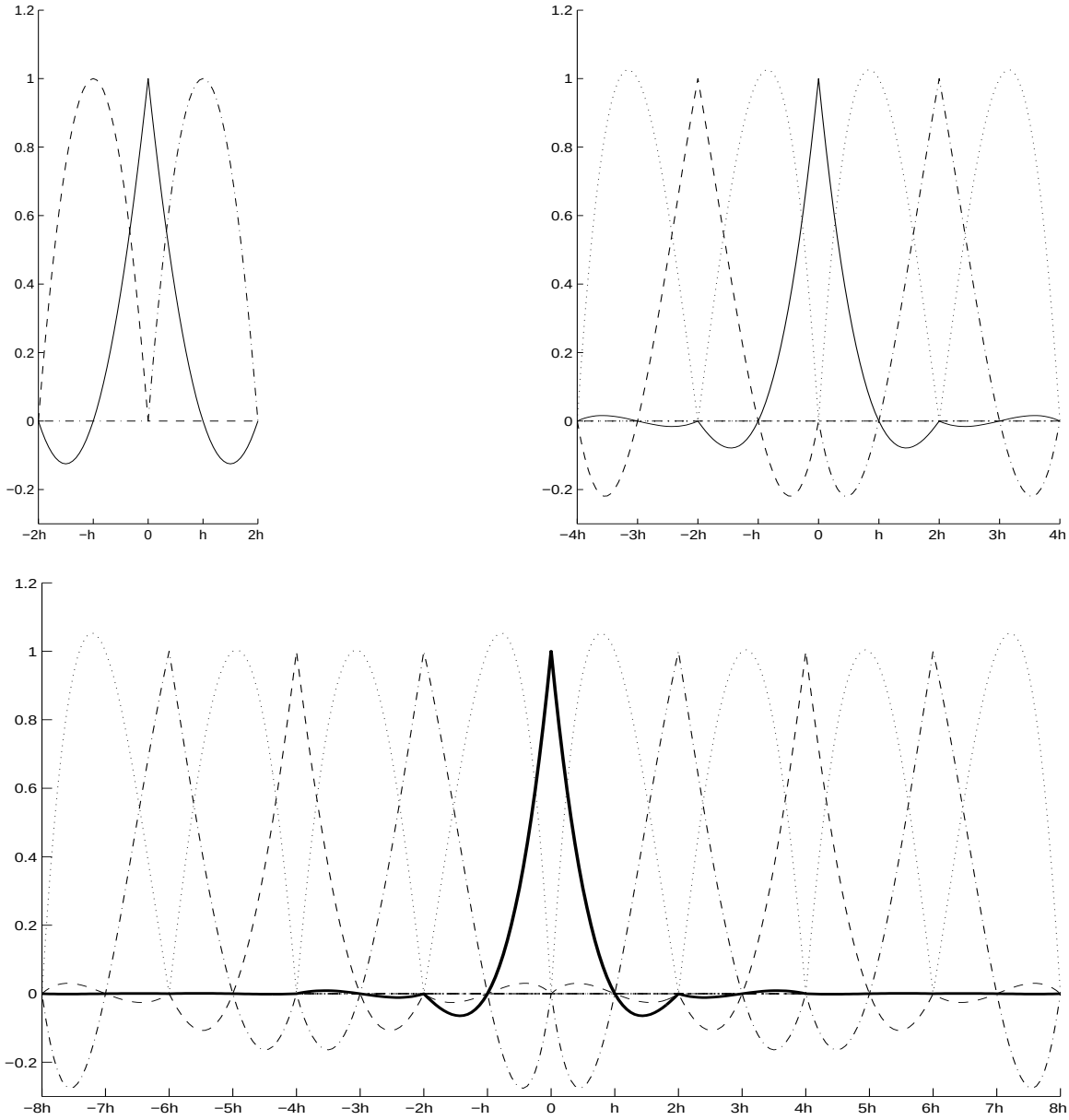


Abbildung 6.2: Die nodalen Basisfunktionen zur hierarchischen Basis der Ordnung 2 (links oben), 3 (rechts oben) und 4 (unten). Der Träger der zum Punkt $x = 0$ gehörenden Basisfunktion der Ordnung 3 (4) erstreckt sich über 8 (16) Maschenweiten.

$p > 2$ werden die Integrale deshalb der Einfachheit halber mit den Basisfunktionen zweiter Ordnung berechnet.

Der lokale Charakter der nodalen Basisfunktionen sowie die Annahme, dass die Koeffizientenfunktionen $c_{\boldsymbol{\alpha\beta}}$ hinreichend glatt sind, rechtfertigen des weiteren, das Integral (6.5) durch den einfacher zu bestimmenden Ausdruck

$$c_{\boldsymbol{\alpha\beta}}(\boldsymbol{r}) \cdot \int_{(\boldsymbol{r}\pm 2\boldsymbol{h})\times(\boldsymbol{r}\pm 2\boldsymbol{h})} D^{\boldsymbol{\alpha}}\phi_{\boldsymbol{r}}(\boldsymbol{x}) \cdot D^{\boldsymbol{\beta}}\phi_{\boldsymbol{r}}(\boldsymbol{x})\, d\boldsymbol{x}. \tag{6.6}$$

zu ersetzen. Mit $(\boldsymbol{r} \pm 2\boldsymbol{h}) \times (\boldsymbol{r} \pm 2\boldsymbol{h})$ ist die Umgebung von ± 2 Maschenweiten um den Gitterpunkt $\boldsymbol{r}$ gemeint. Wegen der Tensorproduktstruktur der Basisfunktionen $\phi_{\boldsymbol{r}}$ lassen sich die Integrale als Produkt eindimensionaler Integrale leicht auswerten.

6.3 Kosten für einen Mehrgitterzyklus

Im Folgenden wird der Rechenaufwand für einen Mehrgitterzyklus abgeschätzt. Als Referenzmaß wird der Aufwand für eine Multiplikation mit der Steifigkeitsmatrix über dem feinsten Gitter, also dem Elementgitter G, angesetzt. Er wird mit WU (work unit) bezeichnet. Ferner wird nur die Arbeit auf den Gittern selbst berücksichtigt, der Datentransport zwischen den Gittern wird vernachlässigt.

Betrachten wir zunächst die Teilgitter G_j, $1 \le j \le J$. Der Rechenaufwand setzt sich dort aus den Glättungen und der Residuumsneuberechnung nach der Korrektur auf dem jeweils nächstgröberen Gitter zusammen. Hierzu sind $(\nu_1+\nu_2+1)$ Multiplikationen mit dem Operator $\tilde{A}_j$ nötig. Dabei ist ν_1 die Zahl der Glättungen vor dem Abstieg auf das nächstgröbere Gitter, ν_2 ist die Zahl der danach stattfindenden Glättungen. Da die Multiplikation mit $\tilde{A}_j$ mit linearem Rechenaufwand in der Anzahl N_j der Gitterpunkte in G_j bewältigt wird, ergibt sich der Rechenaufwand über G_j zu

$$A(G_j) \;=\; \frac{N_j}{N} \cdot (\nu_1 + \nu_2 + 1)\; WU\,. \tag{6.7}$$

Die N_j ergeben in der Summe

$$\sum_{j=1}^{J} N_j \;\approx\; 2^d \cdot N. \tag{6.8}$$

Dies ergibt sich aus dem Zusammenhang zwischen der Vollgitterzerlegung V_j, $1 \le j \le J$, und der hierarchischen Teilraumzerlegung (3.4) in Inkrementräume $W_{\boldsymbol{k}}$, $\mathbf{0} \le \boldsymbol{k} \le \boldsymbol{\ell}$. Dabei entspricht jedem Vollgitterraum V_j genau ein Inkrementraum $W_{\boldsymbol{k}}$. Für die Zahl N_j der Freiheitsgrade im Vollgitterraum gilt nun offensichtlich $N_j \approx 2^d \cdot \dim W_{\boldsymbol{k}}$. Aus $\sum_{\boldsymbol{k}} \dim W_{\boldsymbol{k}} = N$ folgt dann (6.8). Zusammen mit (6.7) ergibt sich damit für den Rechenaufwand summiert über alle Teilgitter G_j

$$\sum_{j=1}^{J} A(G_j) \;\approx\; 2^d \cdot (\nu_1 + \nu_2 + 1)\; WU. \tag{6.9}$$

Nun zum Rechenaufwand auf dem Gitter G. Dort finden keine Glättungen statt. Lediglich das Residuum ist nach der Korrektur auf den (jeweils feinsten) Teilgittern G_j neu zu berechnen, was je einer WU entspricht. Ist n_G die Anzahl dieser G_j, so ergibt sich zusammen mit (6.9) schließlich der Rechenaufwand für einen Mehrgitterzyklus zu

$$\left(2^d \cdot (\nu_1 + \nu_2 + 1) \; + \; n_G\right) \, WU. \tag{6.10}$$

Handelt es sich bei G um ein regelmäßiges L_2-basiertes Dünngitter der Tiefe n, so ist

$$n_G \;=\; \begin{cases} 1, & \text{falls } d = 1, \\ n, & \text{falls } d = 2, \\ \frac{n(n+1)}{2}, & \text{falls } d = 3. \end{cases}$$

Für allgemeine Dimension d hat man

$$n_G \;=\; O(n^{d-1}) = O(|\log N|^{d-1}).$$

6.4 Adaptive Elemente und mehrelementige Diskretisierungen

Für regelmäßige Dünngitter ist die Unterraumskala aus darin enthaltenen Vollgittern in kanonischer Weise durch die hierarchische Teilraumzerlegung (3.4) gegeben. Für adaptive Gitter kann man ebenfalls alle Teilgitter mit nunmehr lokaler Vollgitterstruktur extrahieren: In Abbildung 6.3 ist das lokal um vier Punkte verfeinerte Gitter aus Abbildung 6.1 zu sehen. Für die Unterraumkorrektur sind nun zusätzlich zwei Gitter mit lokaler Vollgitterstruktur zu berücksichtigen, die im Schema von Abbildung 6.1 vor den Gittern V_4 und V_6 einzuordnen sind. Für die Implementierung ist es wichtig, dass mit jedem Gitterpunkt auch seine hierarchischen Vorfahren im Gitter enthalten sind, d.h. das lokale Vollgitter muss entsprechend erweitert werden. Beim Glätten dürfen allerdings nur die (inneren) Punkte der Vollgitterstruktur korrigiert werden.

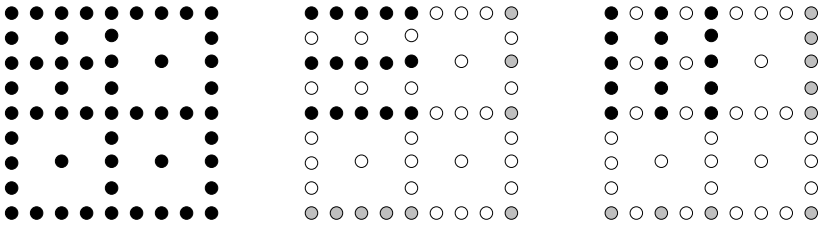
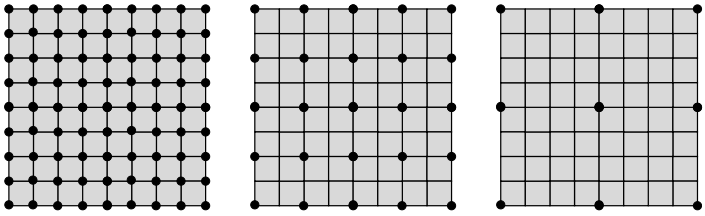


Abbildung 6.3: Adaptives Gitter (links) und lokale Vollgitter. Die grauen Punkte müssen als hierarchische Vorfahren anderer Gitterpunkte ebenfalls im Gitter enthalten sein, werden aber nicht korrigiert.

Ebenfalls weiter gedacht werden muss für mehrelementige Diskretisierungen. Nach der jetzigen Darstellung ist das gröbste Gitter in der Teilraum-Hierarchie durch die Eckpunkte der Elemente gegeben. Für einfache Gebiete, die in wenige Teilgebiete mit

Rechtecksgestalt zerfallen, mag dies völlig ausreichend sein. Ist etwa auf Grund der Geometrie eine sehr feine Zerlegung in Elemente notwendig, müssen weitere Vergröberungsstrategien in Betracht gezogen werden, um die Optimalität des Mehrgitter-Lösers zu gewährleisten. Im Folgenden werden zwei Ideen vorgestellt, die allerdings wegen des hohen Implementierungsaufwands bislang nicht umgesetzt worden sind.

Die einfachste Möglichkeit für eine weitere Vergröberung besteht in der konsequenten Verdopplung der Maschenweite, wie etwa in der folgenden Skizze angedeutet:



Allerdings ist man hier auf einfach zusammenhängende Zerlegungen in $2^{k_1} \times \cdots \times 2^{k_d}$ Elemente angewiesen.

Alternativ kann das Elementnetz in einen Quadtree (Octtree) mit den Elementen als Blattknoten eingebettet werden, siehe Abbildung 6.4. Für eine Unterraumzerlegung bietet sich das folgende Vorgehen an: Die Elemente bzw. ihr Inneres bilden eine Gruppe von Unterräumen, die durch das bereits beschriebene Verfahren unabhängig voneinander korrigiert werden können. Die Trennlinien bilden ihrerseits Unterräume niedriger dimensionaler Struktur, die über die Kantenrelationen des Quadtrees ineinander eingebettet sind. Löcher im Elementnetz können hier einfach realisiert werden: Nicht vorhandene Elemente bedeuten für die Unterraum-Korrektur, dass an dieser Stelle nichts zu tun ist. Für die Beschreibung eines Mehrgitterverfahrens basierend auf einer solchen rekursiven Substrukturierung siehe [4].

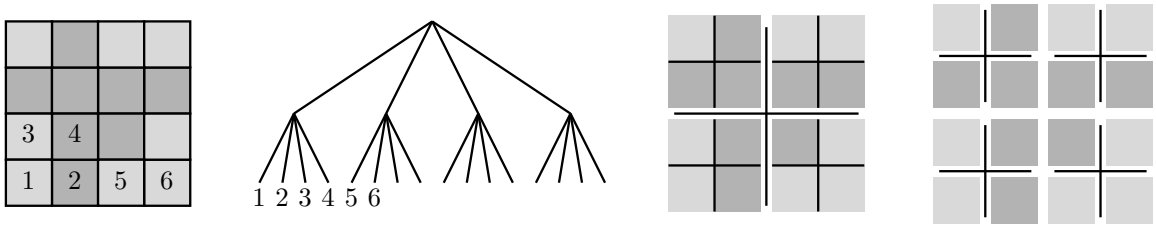


Abbildung 6.4: Einbettung des Elementnetzes in einen Quadtree. Die dunkel schraffierten Quadrate entsprechen Elementen und ergeben zusammen das Berechnungsgebiet. Die hellen Quadrate zeigen Löcher im Elementnetz an.

6.5 Konvergenzraten in der Praxis

In den numerischen Experimenten hat sich das eben vorgestellte Mehrgitterverfahren nur für einfache Probleme als eigenständiger Löser behaupten können, z.B. für Probleme mit konstanten Koeffizienten. Als Vorkonditionierer für das für unsymmetrische

Probleme geeignete `BiCGstab`-Verfahren [11, 51] hat es allerdings bei allen Beispielen aus Kapitel 5 für sehr gute Konvergenzraten gesorgt, wie die nachfolgende Dokumentation bestätigt.

Parameter

Die Parameter sind für die im Folgenden angegebenen bzw. grafisch dargestellten Konvergenzraten jeweils dieselben: Auf jedem Teilgitter wurden drei Vor- und drei Nachglättungsschritte ausgeführt. Für die Abschätzung des Rechenaufwands in (6.10) ist also $\nu_1 = \nu_2 = 3$. Der Dämpfungsfaktor für das Jacobi-Verfahren mit der nach (6.6) vereinfachten Diagonalmatrix ist $\omega = 0.6$.

Messung von Konvergenzrate und Rechenzeit

Bei den im Folgenden dokumentierten Konvergenzraten handelt es sich um Durchschnittswerte für die Reduktion des Residuums. Ist r_0 die euklidische Norm des Startresiduums, r_e das Residuum nach n Iterationsschritten, dann ist die durchschnittliche Reduktionsrate gegeben durch

$$\left(\frac{r_e}{r_0}\right)^{\frac{1}{n}}.$$

Das Endresiduum r_e ist bei den numerischen Experimenten jeweils das letzte, bevor die Näherungslösung akzeptiert worden ist. Für regelmäßige Gitter war das meist nach $n = 3$ bis 4 Schritten der Fall, für adaptive Gitter nach etwa $n = 6$ bis 8 Schritten.

Neben den Konvergenzraten interessieren auch die realen Rechenzeiten. Allerdings ist die Anzahl der Iterationsschritte bis zur Akzeptierung der Näherungslösung von Gitter zu Gitter im Allgemeinen unterschiedlich. Ferner unterliegen die Reduktionsraten für das Residuum Schwankungen. Als ein vernünftiges Maß für die Rechenzeit wurde daher die Zeit genommen, die im Schnitt für die Reduktion des Residuums auf ein Zehntel seines Wertes verbraucht worden ist. Die Rechnungen, auf die sich die angegebenen Zeiten beziehen, sind mit einem mit 800 MHz getakteten Pentium III Prozessor durchgeführt worden.

Ergebnisse

In den Abbildungen 6.5–6.8 sind die Kennzahlen des Mehrgitter-vorkonditionierten `BiCGstab`-Verfahrens für die meisten numerischen Beispiele aus Kapitel 5 grafisch aufbereitet. Links ist jeweils die durchschnittliche Reduktionsrate gegen die Anzahl der Gitterpunkte aufgetragen. Auf der rechten Seite ist die Zeit für die Reduktion des Residuums auf ein Zehntel seiner Größe dargestellt, ebenfalls in Abhängigkeit von der Anzahl der Gitterpunkte.

Insgesamt ist das Verfahren mit Reduktionsraten zwischen 0.01 in den besten Fällen und etwa 0.3 im schlechtesten beobachteten Fall als sehr gut zu bewerten. Der beachtliche Unterschied gibt allerdings Anlass zu einer genaueren Untersuchung, von welchen Faktoren die Reduktionsrate beeinflusst wird:

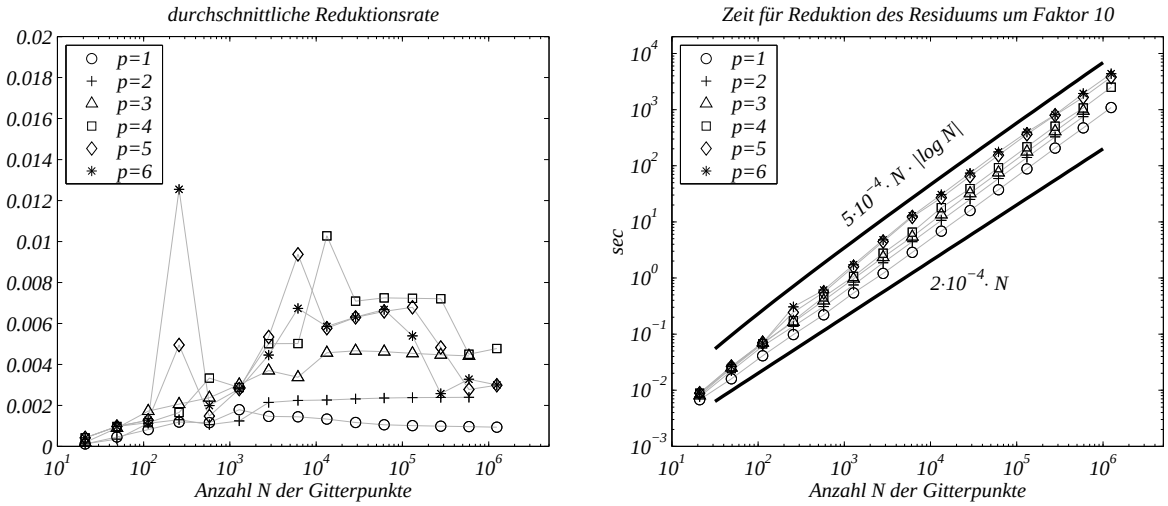


Abbildung 6.5: Beispiel 5.1 (variable Koeffizienten, 2D) – Konvergenzeigenschaften des `BiCGstab`*-Verfahrens mit Mehrgitter-Vorkonditionierung für regelmäßige L_2-basierte Dünngitter*

Niedriger/hoher Polynomgrad im Ansatzraum: Für Ansatzräume aus linearen und quadratischen Polynomen sind die beobachteten Reduktionsraten im Allgemeinen etwas besser und stabiler als für Ansatzräume höherer Ordnung. Mit stabil ist damit gemeint, dass die Rate nur geringen Schwankungen von Gitter zu Gitter unterliegt. Mit zunehmender Ordnung ≥ 3 zeichnet sich ein schwacher Trend zu schlechteren Reduktionsraten ab. Insbesondere sind die Schwankungen zwischen den Gittern hier wesentlich stärker ausgebildet als für niedrige Ordnung.

Dimension: Die Reduktionsrate zeigt sich von der Dimension weitestgehend unabhängig. Bei der Rechenzeit macht sie sich allerdings (leicht) bemerkbar. Nach Formel (6.10) setzt sich der Rechenaufwand aus einem Anteil von den Teilgittern zusammen, der linear in der Anzahl der Gitterpunkte ist, und einem Anteil vom Elementgitter, der wie $O(N \cdot |\log N|^{d-1})$ steigt. An Hand der Abbildungen 6.7 und 6.8, rechts oben, kann man dies nachvollziehen: So überwiegt für grobe Gitter zunächst der $O(N)$-Anteil, für feinere Gitter dagegen der $O(N \cdot |\log N|^{d-1})$-Anteil.

Regelmäßige/adaptive Gitter: Die Reduktionsrate ist für adaptive Gitter schlechter als für regelmäßige Gitter und weist größere Schwankungen zwischen den verschieden feinen Gittern auf. Dies ist auf die unregelmäßige Unterraumhierarchie zurückzuführen.

Fazit

An Hand der Beispiele ist klar geworden, dass das hier vorgestellte Mehrgitterverfahren einen äußerst effizienten Vorkonditionierer darstellt. Die Reduktionsrate bleibt mit zunehmender Feinheit der Gitter nach oben beschränkt. Allerdings muss zugegeben werden, dass es sich bei den Beispielen um gutartige Probleme handelt. Für ein robustes Verfahren, also ein Verfahren, das mit einer größeren Problemklasse von

insbesondere singulär gestörten Problemen zurecht kommt, sind weitere Anpassungen wie problemspezifische Grobgitter nötig. (Das ist ein generelles Problem, das auch bei anderen FE-Diskretisierungen auftritt, also nicht an den dünnen Gittern liegt.) In der hierarchischen Basis und der dadurch einfachen Gestaltung von Gitterskalen steckt hierfür ein großes Potential.

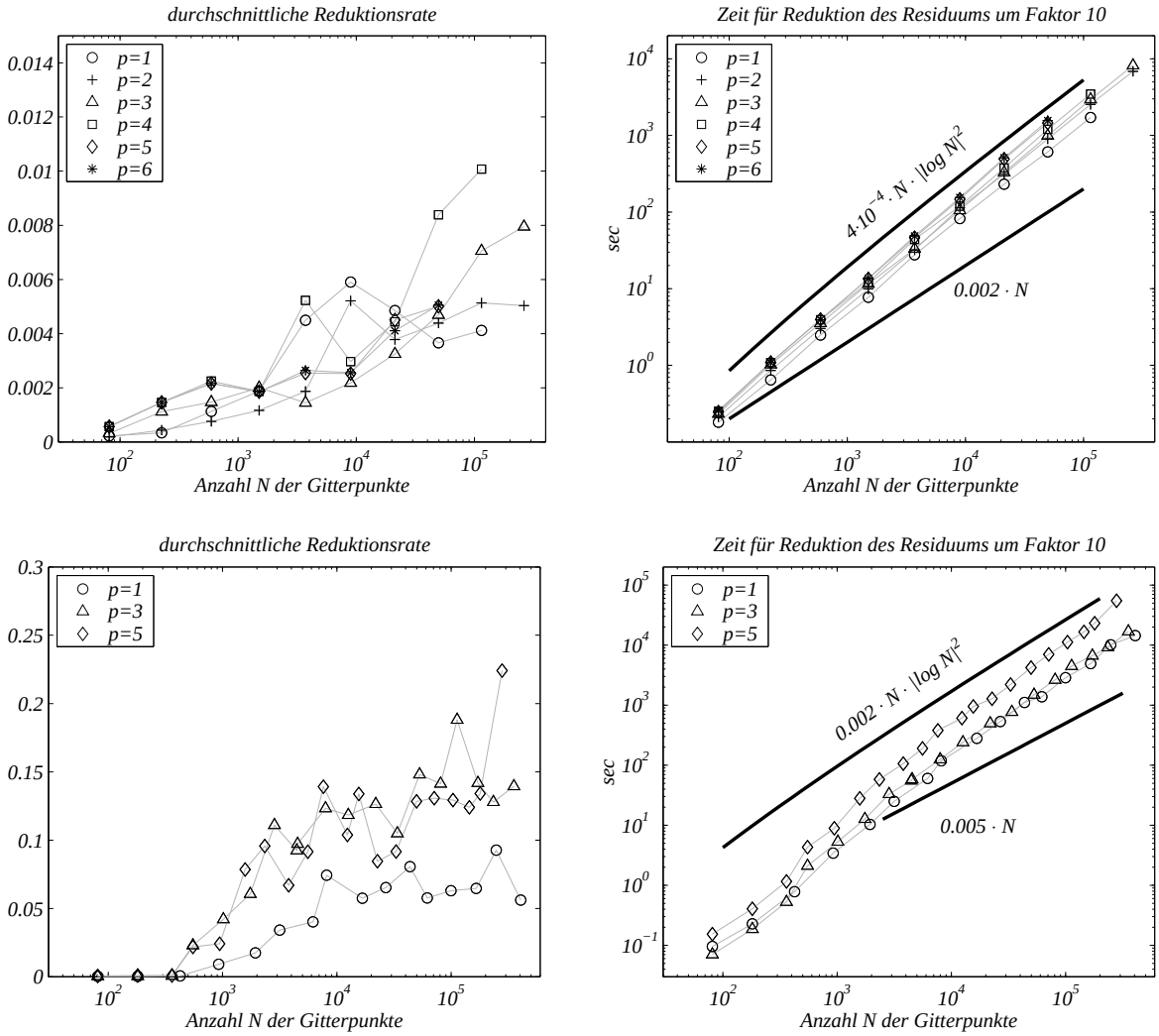


Abbildung 6.6: Beispiel 5.2 (variable Koeffizienten, 3D) – Konvergenzeigenschaften des `BiCGstab`*-Verfahrens mit Mehrgitter-Vorkonditionierung für regelmäßige (oben) und adaptive (unten) L_2-basierte Dünngitter.*

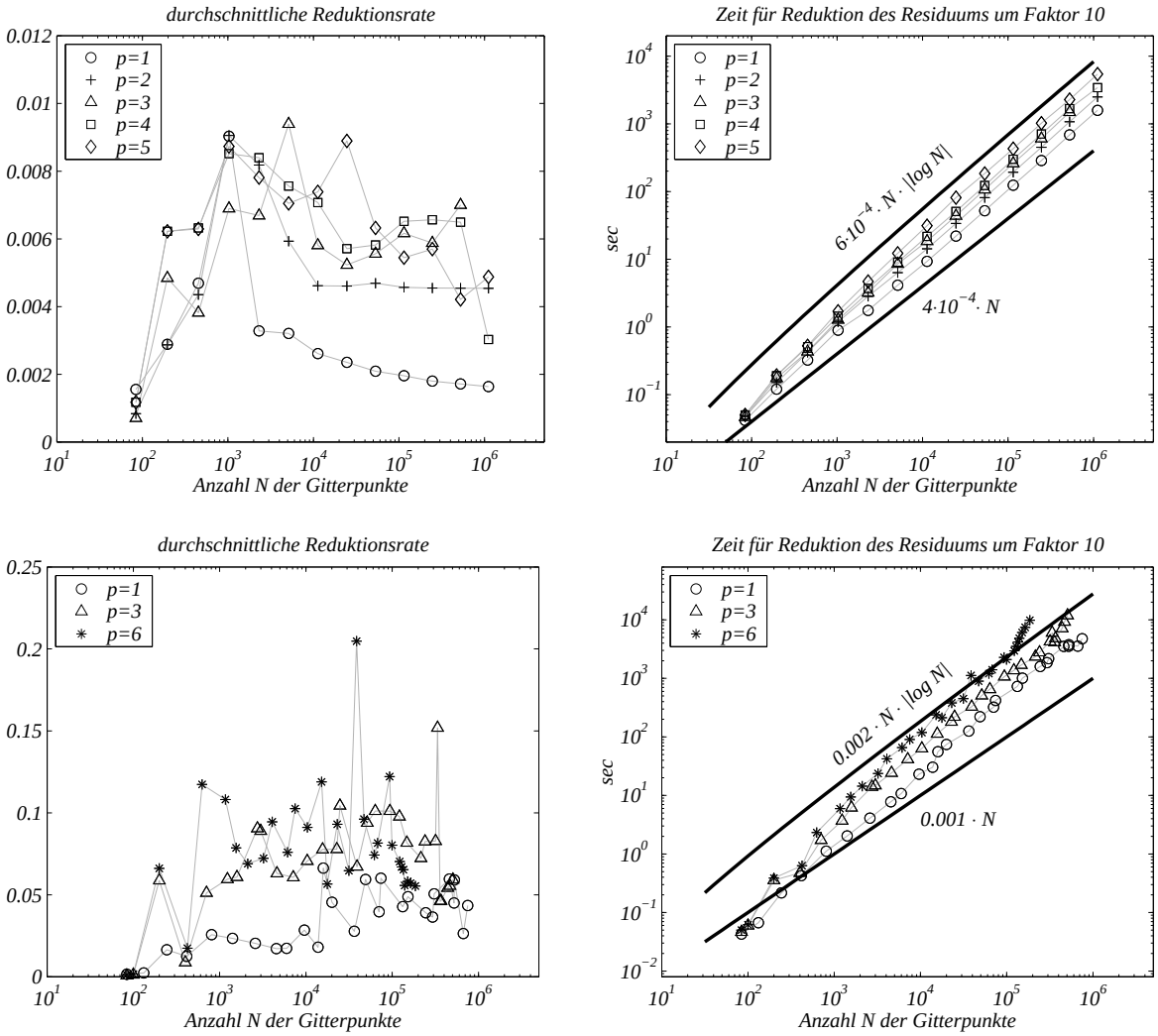


Abbildung 6.7: Beispiel 5.4 (Zylinder, 2D) – Konvergenzeigenschaften des `BiCGstab`*-Verfahrens mit Mehrgitter-Vorkonditionierung für regelmäßige (oben) und adaptive (unten) L_2-basierte Dünngitter.*

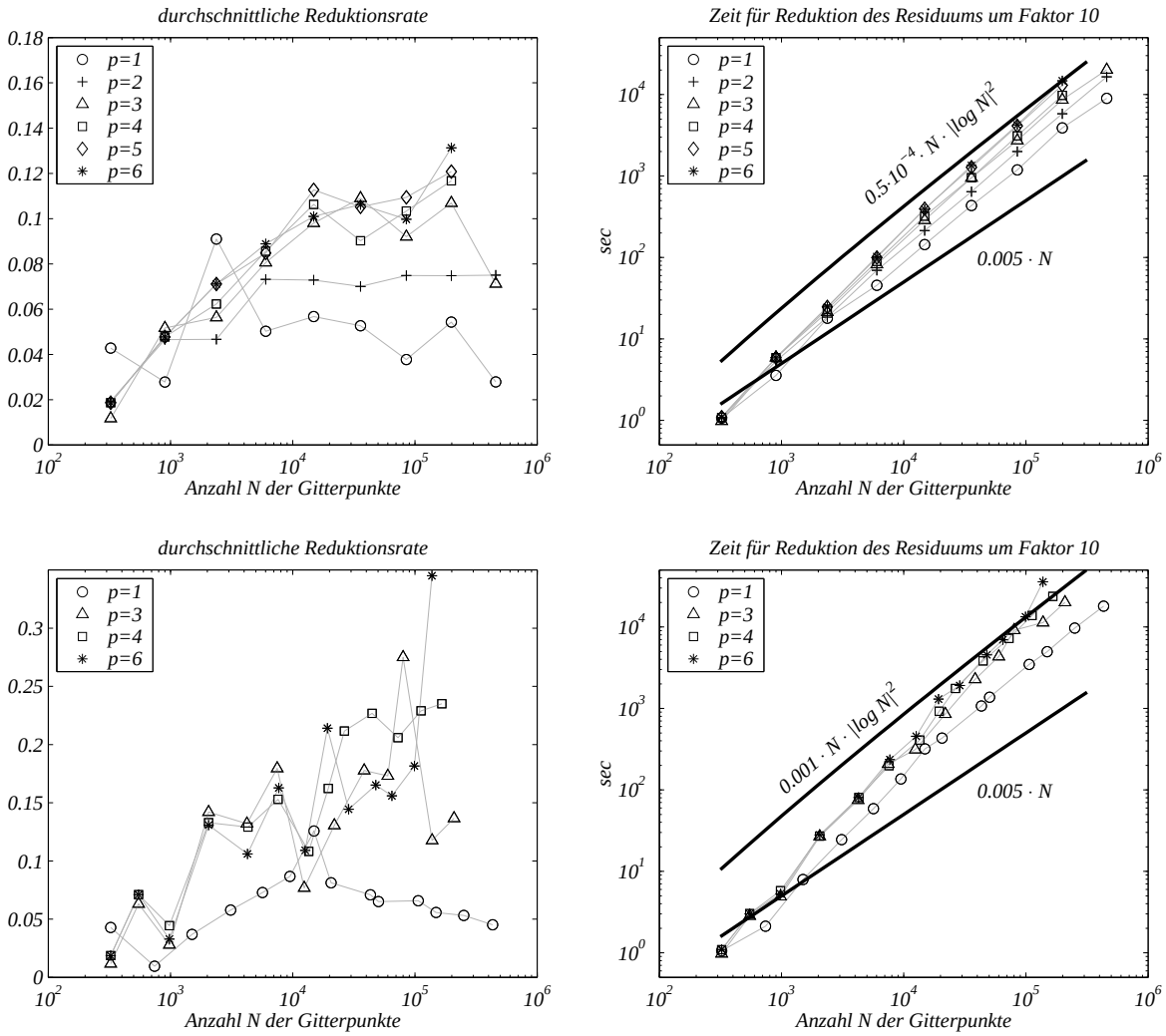


Abbildung 6.8: Beispiel 5.5 (Zylinder, 3D) – Konvergenzeigenschaften des `BiCGstab`*-Verfahrens mit Mehrgitter-Vorkonditionierung für regelmäßige (oben) und adaptive (unten) L_2-basierte Dünngitter.*

7 Zusammenfassung und Ausblick

7.1 Was wurde erreicht?

In der vorliegenden Arbeit wurde ein Finite-Element-Verfahren für elliptische Randwertprobleme zweiter Ordnung mit variablen Koeffizienten vorgestellt, das als Referenzelement isoparametrische, adaptive Dünngitterelemente höherer Ordnung benutzt. In den folgenden Punkten wird kurz zusammengefasst, welche Erweiterungen der Dünngittertechnik, wie sie aus früheren Arbeiten bekannt ist, im Rahmen der vorliegenden Arbeit vorgenommen wurden und welche Ergebnisse damit erzielt wurden:

- Algorithmische Probleme, die sich für die Multiplikation mit der Steifigkeitsmatrix auf Grund von variablen Koeffizienten und Elementtransformationen ergeben, konnten durch eine gitterabhängige Modifikation der Bilinearform in der schwachen Formulierung des Randwertproblems umgangen werden. Mit Hilfe von Standardalgorithmen über dünnen Gittern ist es möglich, die modifizierte Bilinearform effizient auszuwerten: Die Matrix-Vektor-Multiplikation erfolgt mit konstantem, gitterunabhängigem Rechenaufwand je Freiheitsgrad.

- Die für die Analyse mit dem ersten Strang-Lemma benötigte Konsistenz der modifizierten Bilinearform wurde nachgewiesen. Die gleichmäßige Elliptizität wurde auf die gleichmäßige Stetigkeit bestimmter Multiplikationsoperatoren über dünnen Gittern zurückgeführt. Für diese grundlegende Eigenschaft konnte zwar kein Nachweis erfolgen, numerische Experimente untermauern allerdings die Hypothese.

- Die numerischen Berechnungen mit der modifizierten Bilinearform zeigen ein sehr gutes Konvergenzverhalten. In den meisten Fällen stimmt es mit dem für den Interpolationsfehler bereits a-priori bekannten Verhalten überein, siehe hierzu auch den nächsten Punkt. Eines der interessantesten Ergebnisse ist, dass durch Anwendung adaptiver Techniken der für hohe Ordnung meist lange präasymptotische Bereich wesentlich abgekürzt werden kann.

- Für den Fall eines echten Galerkin-Verfahrens, also bei Verwendung der originalen Bilinearform, kann der Approximationsfehler bezüglich der Energienorm mit Hilfe des Lemmas von Céa durch den Interpolationsfehler abgeschätzt werden. Dieser wird in der Regel nach einer Rücktransformation auf das Referenzelement

abgeschätzt. Hierzu wurden die für Dünngitter-Referenzelemente und Funktionen mit homogenen Randdaten bekannten Fehlerabschätzungen auf Funktionen mit nicht verschwindenden Randdaten übertragen.

- Ein Mehrgitterverfahren wurde konstruiert, das im Gegensatz zu früheren Arbeiten symmetrisch in Ansatz- und Testraum ist. Als Vorkonditionierer für das `BiCGstab`-Verfahren hat es für eine Reihe von (gutartigen) Problemen für gute Konvergenzraten gesorgt.

7.2 Anregungen für weitere Arbeiten

Für das theoretische Fundament des vorgestellten FE-Verfahrens muss noch die gleichmäßige Elliptizität für die modifizierte Bilinearform nachgewiesen werden. Ein Ansatz, der das Problem auf die – bislang nur experimentell – bestätigte gleichmäßige Stetigkeit bestimmter Multiplikationsoperatoren zurückführt, wurde in der Arbeit angegeben. Interessant sind in diesem Zusammenhang Wavelet-Basen, die bereits erfolgreich über dünnen Gittern verwendet werden [35, 40, 43]. Hier gelten im Gegensatz zur Polynom-Basis einfache Normäquivalenzen zwischen den dargestellten Funktionen und den zugehörigen Koeffizientenvektoren, die einen leichteren Zugang zu Normabschätzungen bieten.

Ein weiterer Nachteil der modifizierten Bilinearform ist der Verlust der Symmetrie. Eine einfache Symmetrisierung der modifizierten Bilinearform hat nicht zum gewünschten Erfolg geführt, das resultierende Verfahren zeigte für höhere Ordnung und Dimension schlechtere Konvergenzeigenschaften als vergleichbare Vollgitterverfahren. Dabei sollte die Notwendigkeit einer symmetrischen Bilinearform überdacht werden: Die vorgestellte modifizierte Bilinearform ist nur „schwach unsymmetrisch“, für glatte Funktionen bzw. Funktionen, die in gröberen Teilräumen enthalten sind, weicht die Bilinearform nur im Rahmen des Diskretisierungsfehlers von der Transponierten ab. Die Unsymmetrie tritt am stärksten für Basisfunktionen mit langgestrecktem Träger hervor. Als Hinweis dafür, dass die Abweichung von der Symmetrie „gering“ ist, kann die Tatsache gewertet werden, dass das cg-Verfahren für die Probleme in Kapitel 5 trotzdem konvergiert.

Die adaptive Gitterverfeinerung betreffend ist folgendes festzuhalten: Da in der vorgestellten Diskretisierung die Koeffizienten und Transformationen über dem Elementgitter interpoliert werden, muss sich die Adaption nicht nur an der (auf das Referenzelement transformierten) Lösung orientieren, sondern auch an den Koeffizienten und den Transformationen. Dies ist bislang noch nicht berücksichtigt worden. Die Probleme in den numerischen Experimenten von Kapitel 5 sind alle gutartig, d.h. die Koeffizienten der Differentialgleichung sind glatt, die Gebiete bzw. die Transformationen ebenso, so dass hier kein Bedarf für die Adaption bestand.

Um C^0-Übergänge zwischen verschieden feinen Elementen zu erzeugen, wurde vorgeschlagen, Freiheitsgrade auf der gemeinsamen Nahtstelle, die nicht zu allen angrenzenden Elementen gehören, zu eliminieren (siehe Abschnitt 4.1). Aus technischen Gründen

werden also Freiheitsgrade unterschlagen, die für die Approximation eventuell wichtig sind. Auf diesen Umstand lassen sich wahrscheinlich große Fehler in der Umgebung der Elementgrenzen zurückführen, die bei einigen über die Beispiele von Kapitel 5 hinausgehenden Testrechnungen festgestellt wurden. Hier besteht also noch weiterer Bedarf an Untersuchungen, um die Robustheit des Verfahrens zu gewährleisten.

Für eine größere Flexibilität bei der Geometrie-Diskretisierung bieten sich simpliziale Dünngitterelemente in Form von degenerierten Rechteckselementen an, bei denen eine Seite (Fläche) zu einem Punkt zusammengezogen wird. Erreicht werden kann dies innerhalb des bereits vorgestellten isoparametrischen Ansatzes durch entsprechende singuläre Transformationen. Die Freiheitsgrade auf der zu einem Punkt entarteten Seite müssen dabei aneinander gekoppelt werden. Genau genommen sind nur die Eckpunkte gekoppelt, die inneren Freiheitsgrade sind dank der hierarchischen Darstellung lediglich auf null zu setzen, d.h. sie sind keine eigentlichen Freiheitsgrade mehr. Dornseifer [22] hat dieses Vorgehen für einelementige Diskretisierungen bereits erfolgreich angewandt.

Einer der Hauptvorteile der hierarchischen Basis ist, dass Unterraumhierarchien für Mehrgitterverfahren nicht erst konstruiert werden müssen, sondern in natürlicher Weise gegeben sind. Dementsprechend einfach gestaltet sich das in Kapitel 6 vorgestellte Verfahren. Die polynomiale hierarchische Basis nach Bungartz sorgt dafür, dass die für lineare Ansatzfunktionen geeigneten Glätter nahezu unverändert übernommen werden können: Durch die spezielle Konstruktion ist jedem Gitterpunkte – wie im linearen Fall – genau ein Freiheitsgrad zugeordnet, die zugehörigen nodalen Basisfunktionen haben lokalen Charakter und unterscheiden sich für verschiedene Ordnung nicht wesentlich in ihrer Gestalt. Für Elementnetze, die aus vielen Dünngitterpatches bestehen, muss noch ein über die Elementgrenzen hinausgehender Vergröberungsmechanismus gefunden werden. Ein Versuch könnte in einer rekursiven Substrukturierung bestehen, wie sie am Ende von Abschnitt 6.4 kurz angesprochen wurde.

Literaturverzeichnis

[1] Adams, R.A.: *Sobolev Spaces.* Academic Press, New York, 1975.

[2] Ainsworth, M. und T.J. Oden: *A Posteriori Estimation in Finite Element Analysis.* Wiley – Interscience, New York, 2000.

[3] Babuška, I. und B.A. Szabó: *Trends and New Problems in Finite Element Methods.* In: Whiteman, J.R. (Herausgeber): *The Mathematics of Finite Elements and Applications — Highlights 1996.* John Wiley & Sons, 1996.

[4] Bader, M.: *Robuste, parallele Mehrgitterverfahren für die Konvektions-Diffusions-Gleichung.* Doktorarbeit, Technische Universität München, 2001.

[5] Balder, R.: *Adaptive Verfahren für elliptische und parabolische Differentialgleichungen auf dünnen Gittern.* Doktorarbeit, Technische Universität München, 1994.

[6] Balder, R. und C. Zenger: *The solution of multidimensional real Helmholtz equations on sparse grids.* SIAM J. Sci. Comp., 17:631–646, 1996.

[7] Bangerth, W., R. Hartmann und G. Kanschat: `deal.II` *Differential Equations Analysis Library, Technical Reference.* IWR. `http://www.dealii.org`.

[8] Bangerth, W. und G. Kanschat: *Concepts for Object-Oriented Finite Element Software – the* ***`deal.II`*** *library.* Preprint 99-43 (SFB 359), IWR Heidelberg, Oktober 1999.

[9] Bank, R.E.: *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, User's Guide.* Frontiers in Applied Mathematics, 15, 1994.

[10] Bank, R.E., T. Dupont und H. Yserentant: *The hierarchical basis multigrid method.* Numerische Mathematik, 52:427–458, 1988.

[11] Barrett, R., M. Berry, T.F. Chan und et al. : *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, Philadelphia, PA, 1994.

[12] Barthelmann, V., E. Novak und K. Ritter: *High dimensional polynomial interpolation on sparse grids.* Advances in Computational Mathematics, 12:273–288, 2000.

[13] Braess, D.: *Finite Elemente.* Springer, Berlin, 1997.

[14] Bungartz, H.–J.: *Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung.* Doktorarbeit, Technische Universität München, 1992.

[15] Bungartz, H.–J.: *A unidirectional approach for d–dimensional finite element methods of higher order on sparse grids.* In: Manteuffel, T. und S. McCormick (Herausgeber): *Proc. Copper Mountain Conf. on Iterative Methods*, 1996.

[16] Bungartz, H.–J.: *A multigrid algorithm for higher order finite elements on sparse grids.* ETNA, 6:63–77, 1997.

[17] Bungartz, H.–J.: *Finite Elements of Higher Order on Sparse Grids.* Habilitationsschrift, Technische Universität München, 1998.

[18] Bungartz, H.–J. und M. Griebel: *A Note on the Complexity of Solving Poisson's Equation for Spaces of Bounded Mixed Derivatives.* J. Complexity, 15:167–199, 1999.

[19] Bungartz, H.–J. und C. Zenger: *Error Control for Adaptive Sparse Grids.* In: Bulgak, H. und C. Zenger (Herausgeber): *Error Control and Adaptivity in Scientific Computing*, Seiten 125–157. Kluwer Academic Publishers, 1999.

[20] Ciarlet, P.G. und J.L. Lions (Herausgeber): *Handbook of Numerical Analysis, Vol. II.* Elsevier Science Publishers B.V. (North-Holland), 1991.

[21] Deuflhard, P. und A. Hohmann: *Numerische Mathematik I.* Walter de Gruyter, Berlin, New York, 1993.

[22] Dornseifer, T.: *Diskretisierung allgemeiner elliptischer Differentialgleichungen in krummlinigen Koordinatensystemen auf dünnen Gittern.* Doktorarbeit, Technische Universität München, 1997.

[23] Fornberg, B.: *A Practical Guide to Pseudospectral Methods.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1996.

[24] Gilbarg, D. und N.S. Trudinger: *Elliptic Partial Differential Equations of Second Order.* Classics in Mathematics. Springer, Berlin, 2001.

[25] Grădinaru, V.C.: *Whitney Elements on Sparse Grids.* Doktorarbeit, Universität Tübingen, 2002.

[26] Griebel, M. und P. Oswald: *On Additive Schwarz Preconditioners for Sparse grid Discretizations.* Numer. Math., 66:449–464, 1994. auch als Bericht Math/92/7, Institut für angewandte Mathematik, Friedrich-Schiller-Universität Jena, 1992.

[27] Griebel, M. und T. Schiekofer: *An adaptive sparse grid Navier–Stokes solver in 3D based on finite differences.* In: *Proceedings ENUMATH97*, Heidelberg, 1997.

[28] Griebel, M. und G. Zumbusch: *Hash Based Adaptive Parallel Multilevel Methods with Space-Filling Curves.* In: *NIC Symposium 2001, Proceedings*, John von Neumann Institute for Computing, Jülich, 2001.

[29] Hackbusch, W.: *Multigrid Methods and Applications.* Springer Verlag, Berlin, 1985.

[30] Hackbusch, W.: *Iterative Lösung großer schwachbesetzter Gleichungssysteme.* B.G. Teubner, Stuttgart, 1993.

[31] Josuttis, N.M.: *The C++ Standard Library.* Addison Wesley Professional, 1999.

[32] Knabner, P. und L. Angermann: *Numerik partieller Differentialgleichungen.* Springer, Berlin, 2000.

[33] Knupp, P. und S. Steinberg: *Fundamentals of Grid Generation.* CRC Press, Inc., London, 1993.

[34] Koster, F.: *Multiskalen-basierte Finite-Differenzen–Verfahren auf adaptiven dünnen Gittern.* Doktorarbeit, Universität Bonn, 2001.

[35] Nitsche, P.–A.: *Sparse approximation of singularity functions.* Research Report No. 2002–18, Seminar für Angewandte Mathematik, ETH Zürich, 2002.

[36] Novak, E. und K. Ritter: *High dimensional integration of smooth functions over cubes.* Numerische Mathematik, 75:79–97, 1996.

[37] Pflaum, C.: *Diskretisierung elliptischer Differentialgleichungen mit dünnen Gittern.* Doktorarbeit, Technische Universität München, 1996.

[38] Schneider, S. A.: *Adaptive Solution of Elliptic Partial Differential Equations by Hierarchical Tensor Product Finite Elements.* Doktorarbeit, Technische Universität München, 2000.

[39] Schwab, Ch.: *p- and hp-Finite Element Methods, Theory and Applications in Solid and Fluid Mechanics.* Oxford University Press, 1998.

[40] Schwab, Ch. und R.–A. Todor: *Sparse Finite Elements for Elliptic Problems with Stochastic Data.* Research Report No. 2002–5, Seminar für Angewandte Mathematik, ETH Zürich, 2002.

[41] Smith, B.F., E.B. Petter und W.D. Gropp: *Domain Decomposition: Parallel multilevel methods for elliptic partial differential equations.* Cambridge University Press, 1996.

[42] Smolyak, S. A.: *Quadrature and interpolation formulas for Tensor Products of Certain Classes of Functions.* Dokl. Akad. Nauk SSSR, 148:1042–1043, 1963. Russian, Engl. Transl.: Soviet Math. Dokl. 4:240–243, 1963.

[43] Sprengel, F.: *Periodic interpolation and wavelets on sparse grids.* Numerical Algorithms, 17:147–169, 1998.

[44] Spurk: *Strömungslehre.* Springer, 4 Auflage, 1996.

[45] Stoer, J.: *Numerische Mathematik*, Band 1. Springer, Berlin, Heidelberg, 8 Auflage, 1999.

[46] Störtkuhl, T.: *Ein numerisches adaptives Verfahren zur Lösung der biharmonischen Gleichung auf dünnen Gittern.* Doktorarbeit, Technische Universität München, 1995.

[47] Strang, G. und G.J. Fix: *An Analysis of the Finite Element Method.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973.

[48] Szabó, B.A. und I. Babuška: *Finite Element Analysis.* John Wiley & Sons, 1991.

[49] Trottenberg, U., C. Oosterlee und A. Schüller: *Multigrid.* Academic Press, 2001.

[50] Verfürth, R.: *A Review of A Posteriori Error Estimates and Adaptive Mesh–Refinement Techniques.* Teubner–Wiley, Stuttgart, New York, 1995.

[51] Vorst, H. van der: *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems.* SIAM Journal of Scientific Computing, 13(2):631–644, 1992.

[52] Wasilkowski, G. W. und H. Woźniakowski: *Explicit Cost Bounds of Algorithms for Multivariate Tensor Product Problems.* J. Complexity, 11:1–56, 1995.

[53] Xu, J.: *The Method of Subspace Corrections.* Journal of Computational and Applied Mathematics, 128:335–362, 2001.

[54] Yserentant, H.: *Hierarchical bases give conjugate gradient type methods a multigrid speed of convergence.* Applied Mathematics and Computation, 19:347–358, 1986.

[55] Yserentant, H.: *On the multi–level splitting of finite element spaces.* Numerische Mathematik, 49:379–412, 1986.

[56] Zenger, C.: *Sparse Grids.* In: Hackbusch, W. (Herausgeber): *Parallel Algorithms for Partial Differential Equations*, Notes on Numerical Fluid Mechanics 31, Seiten 241–251. Vieweg, Braunschweig, 1991.