

**Advancing Front Gittergenerierung und
a priori Fehlerabschätzungen
für elliptische Randwertprobleme
mit Singularitäten**

Von der Fakultät Mathematik der Universität Stuttgart
zur Erlangung der Würde eines Doktors der
Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Wolfgang Hoffmann aus Hockenheim

Hauptberichter:	Prof. Dr. K. Höllig
Mitberichter:	Prof. Dr. G. Wittum
Mitberichter:	Prof. Dr. A. H. Schatz
Tag der Einreichung:	16.12.1999
Tag der mündlichen Prüfung:	14.12.2000

Mathematisches Institut A der Universität Stuttgart

Inhaltsverzeichnis

Zusammenfassung und Ausblick	8
1 Einleitung	11
2 Gittergenerierungsverfahren	16
2.1 Geometrische Definitionen	18
2.2 Delaunay Triangulierung	22
2.3 Quadtree-Methode	25
2.4 Advancing Front Methode	29
3 Gebietsdefinitionen und Datenstrukturen	34
3.1 Gebiete in Parameterdarstellung	34
3.2 Gebiete in polygonaler Darstellung	40
3.3 Datenstrukturen	41
4 Die Advancing Front Methode des Programmbaukastens <i>UG</i>	44
4.1 Listenkonzept des Gittergenerators	47
4.1.1 Unabhängige Frontlisten	48
4.1.2 Frontlisten	50
4.1.3 Frontkomponenten	51
4.2 Erzeugung von Knoten auf dem Rand	51
4.3 Erzeugung von Knoten und Dreieckselementen im Rechengebiet	53
4.4 Kontrollfunktionen	56
4.4.1 Kontrollfunktion NearerNeighbor	58
4.4.2 Kontrollfunktion CutNeighbor	58
4.4.3 Kontrollfunktion InCircleOrTriangle	59
4.4.4 Kontrollfunktion FrontLineIntersection	60
4.4.5 Aktualisierung der Frontlisten	62
4.5 Verbesserung der Gitterqualität	66
4.5.1 Laplace-Glättung	66
4.5.2 Modifizierte Laplace-Glättung	67
4.6 Anwendungen des Gittergenerators	69

4.6.1	Beispiel Nordseeküste	69
4.6.2	Weitere Beispiele	72
5	Einführung des Modellproblems	80
5.1	Sobolev Räume und Normen	80
5.2	Formulierung des Randwertproblems	82
5.3	Eigenschaften der Lösungen des Dirichlet-Problems	84
5.4	Approximationssätze und grundlegende Definitionen	86
5.5	Bedeutung des relativen Fehlers	90
6	Eine neue <i>a priori</i> Abschätzung des relativen Fehlers des Gradienten in der L_2-Norm	93
6.1	Klassische Fehlerabschätzung in der Energienorm	93
6.2	Standardabschätzung des Fehlers für verfeinerte Gitter	95
6.3	Beweis einer <i>a priori</i> Abschätzung des relativen Fehlers des Gra- dienten in der L_2 -Norm	97
7	Eine neue <i>a priori</i> Abschätzung des relativen Fehlers in der L_∞-Norm	102
7.1	Bekannte Abschätzungen in der L_∞ -Norm	102
7.2	Vorbereitende Lemmata	103
7.3	Diskussion einer neuen Abschätzung des relativen Fehlers in der L_∞ -Norm	109
7.4	Beweis der Theoreme zur Abschätzung der relativen Fehler	112
8	Numerische Experimente	119
8.1	Das Randwertproblem der numerischen Experimente	119
8.2	Algorithmus zur Konstruktion eines geometrischen Gitters	121
8.3	Funktionsklassen	124
8.4	Vorbemerkungen zu den numerischen Experimenten	129
8.5	Numerisches Experiment 1	131
8.6	Numerisches Experiment 2	139
8.7	Numerisches Experiment 3	146
8.8	Numerisches Experiment 4	153
8.9	Numerisches Experiment 5	159

Abbildungsverzeichnis

2.1	Beispiele von Triangulierungen ohne und mit Steiner Punkten . . .	21
2.2	Triangulierung einer Punktmenge mit Voronoi-Polygonen	23
2.3	Punktmenge mit konvexer Hülle	23
2.4	Voronoi-Polygon exemplarisch für einen Punkt	23
2.5	Quadtree einer Punktmenge mit zugehörigem Diagramm	27
2.6	Quadtree eines polygonal berandeten Gebietes	28
2.7	Anpassen des Quadrees an die Geometrie	28
2.8	Rechengitter nach der Quadtree-Methode	28
2.9	Advancing Front Methode nach Tracy	31
2.10	Geometrie einer Advancing Front	33
2.11	Geometrie mit erster aktiver Front	33
2.12	Geometrie mit aktualisierter aktiver Front nach dem Einfügen ei- niger Dreiecke	33
3.1	Parameterdarstellung eines Rechengebietes	35
3.2	Parametrisierung eines Quadrats	38
3.3	Parametrisierung eines Kreises	39
3.4	Polygongebiet mit inneren Rändern	40
4.1	Advancing Front Algorithmus des Programmbaukastens <i>UG</i> . . .	45
4.2	Datenstruktur des Gittergenerators	47
4.3	Zusammenhang zwischen Geometrie und Frontlisten	49
4.4	Erzeugung von Randknoten	52
4.5	Erzeugung eines Knotens im Rechengebiet	55
4.6	Kontrollfunktionen	57
4.7	Bestimmung eines neuen Knotens im Rechengebiet durch die Funk- tion <i>NearerNeighbor</i>	59
4.8	Bestimmung eines neuen Knotens im Rechengebiet durch die Funk- tion <i>CutNeighbor</i>	59
4.9	Bestimmung eines neuen Knotens im Rechengebiet durch die Funk- tion <i>InCircleOrTriangle</i>	60

4.10	Bestimmung eines neuen Knotens im Rechengebiet durch die Funktion <code>FrontLineIntersection</code>	61
4.11	Aktive Front vor und nach der Erzeugung eines Dreiecks mit neuem Knoten	63
4.12	Aktualisierung der aktiven Front	65
4.13	Aktualisierung der aktiven Front	66
4.14	Laplace-Glättung	67
4.15	Gebiet Nordseeküste	70
4.16	Funktion $G(s)$	70
4.17	Triangulierung der Nordseeküste	71
4.18	Gitter in der Umgebung der Insel Helgoland	71
4.19	Advancing Front mit 1000 Elementen	74
4.20	Advancing Front mit 2000 Elementen	74
4.21	Advancing Front mit 3000 Elementen	75
4.22	Advancing Front mit 6200 Elementen	75
4.23	Advancing Front mit 6490 Elementen	76
4.24	Vollständige Triangulierung des Nordseeküstengebiets	76
4.25	Triangulierung eines Salzstocks I	77
4.26	Triangulierung eines Salzstocks II	77
4.27	Triangulierung des Wolfgangsee-Gebietes I	78
4.28	Triangulierung des Wolfgangsee-Gebietes II	78
4.29	Grob-gitter eines Prandtl-Reuß-Plastizität-Problems	79
4.30	Adaptiv verfeinertes Gitter eines Prandtl-Reuß-Plastizität-Problems	79
5.1	Rechengebiete des Modellproblems	82
5.2	Veranschaulichung von Definition 5.4.1	86
5.3	Gitterparameter h_d und ρ	88
7.1	Fortsetzung von u und u_h am Rand des Gebietes Ω	104
8.1	Rechengebiete der numerischen Experimente	121
8.2	Geometrisches Gitter	122
8.3	Konstruktion eines geometrischen Gitters	123
8.4	Experiment 1: Lösung u des Problems (8.23)	132
8.5	Experiment 1: Lösung u des Problems (8.23) als Höhenliniendarstellung	132
8.6	Experiment 1: Lösung u des Problems (8.23)	133
8.7	Experiment 1: Lösung u des Problems (8.23) als Höhenliniendarstellung	133
8.8	Experiment 2: Lösung u des Problems (8.26)	140

8.9	Experiment 2: Lösung u des Problems (8.26) als Höhenliniendarstellung	140
8.10	Experiment 2: Lösung u des Problems (8.26)	141
8.11	Experiment 2: Lösung u des Problems (8.26) als Höhenliniendarstellung	141
8.12	Experiment 3: Lösung u des Problems (8.31)	147
8.13	Experiment 3: Lösung u des Problems (8.31) als Höhenliniendarstellung	147
8.14	Experiment 3: Lösung u des Problems (8.31)	148
8.15	Experiment 3: Lösung u des Problems (8.31) als Höhenliniendarstellung	148
8.16	Experiment 4: Lösung u des Problems (8.34)	154
8.17	Experiment 4: Lösung u des Problems (8.34) als Höhenliniendarstellung	154
8.18	Experiment 5: Lösung u des Problems (8.41)	160
8.19	Experiment 5: Lösung u des Problems (8.41) als Höhenliniendarstellung	160
8.20	Experiment 5: Lösung u des Problems (8.41)	161
8.21	Experiment 5: Lösung u des Problems (8.41) als Höhenliniendarstellung	161

Tabellenverzeichnis

8.1	Numerisches Experiment 1: Konvergenzordnungen der relativen Fehler des Gradienten in der $L_2(\Omega)$ -Norm, uniforme Verfeinerung	134
8.2	Numerisches Experiment 1: Konvergenzordnungen der relativen Fehler der Funktion in der $L_\infty(\tau)$ -Norm, geometrische Verfeinerung	136
8.3	Numerisches Experiment 1: Konvergenzordnungen der relativen Fehler des Gradienten in der $L_\infty(\tau)$ -Norm, geometrische Verfeinerung	137
8.4	Numerisches Experiment 1: Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm, uniforme Verfeinerung	138
8.5	Numerisches Experiment 2: Konvergenzordnungen der relativen Fehler des Gradienten in der $L_2(\Omega)$ -Norm, uniforme Verfeinerung	142
8.6	Numerisches Experiment 2: Konvergenzordnungen der relativen Fehler der Funktion in der $L_\infty(\tau)$ -Norm, geometrische Verfeinerung	143
8.7	Numerisches Experiment 2: Konvergenzordnungen der relativen Fehler des Gradienten in der $L_\infty(\tau)$ -Norm, geometrische Verfeinerung	144
8.8	Numerisches Experiment 2: Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm, uniforme Verfeinerung	145
8.9	Numerisches Experiment 3: Konvergenzordnungen der relativen Fehler des Gradienten in der $L_2(\Omega)$ -Norm, uniforme Verfeinerung	149
8.10	Numerisches Experiment 3: Konvergenzordnungen der relativen Fehler der Funktion in der $L_\infty(\tau)$ -Norm, geometrische Verfeinerung	150
8.11	Numerisches Experiment 3: Konvergenzordnungen der relativen Fehler des Gradienten in der $L_\infty(\tau)$ -Norm, geometrische Verfeinerung	151
8.12	Numerisches Experiment 3: Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm, uniforme Verfeinerung	152
8.13	Numerisches Experiment 4: Spannungsintensitätsfaktoren	155
8.14	Numerisches Experiment 4: Konvergenzordnungen der relativen Fehler in der $L_2(\Omega)$ -Norm, uniforme Verfeinerung	156

8.15	Numerisches Experiment 4: Konvergenzordnungen der relativen Fehler der Funktion in der $L_\infty(\tau)$ -Norm, geometrische Verfeinerung	157
8.16	Numerisches Experiment 4: Konvergenzordnungen der relativen Fehler in der $L_\infty(\tau)$ -Norm, geometrische Verfeinerung	157
8.17	Numerisches Experiment 4: Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm, uniforme Verfeinerung	158
8.18	Numerisches Experiment 5: Spannungsintensitätsfaktoren	162
8.19	Numerisches Experiment 5: Konvergenzordnungen der relativen Fehler in der $L_2(\Omega)$ -Norm, uniforme Verfeinerung	163
8.20	Numerisches Experiment 5: Konvergenzordnungen der relativen Fehler der Funktion in der L_∞ -Norm, geometrische Verfeinerung .	164
8.21	Numerisches Experiment 5: Konvergenzordnungen der relativen Fehler des Gradienten in der L_∞ -Norm, geometrische Verfeinerung	165
8.22	Numerisches Experiment 5: Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm, uniforme Verfeinerung	166

Zusammenfassung und Ausblick

Das Ziel dieser Arbeit ist die Darstellung eines neu entwickelten Advancing Front Gittergenerators für ebene, zweidimensionale komplexe Gebiete mit vielen inneren Rändern und *a priori* Fehlerabschätzungen von Lösungen und deren Gradienten von elliptischen Randwertproblemen mit Singularitäten. Die Lösungen der elliptischen Randwertprobleme werden mit der Methode der finiten Elemente bestimmt, wobei lineare Lagrange-Elemente verwendet werden. Die Untersuchung der elliptischen Randwertproblemen mit Singularitäten erfolgt auf einem Gebiet mit einer einspringenden Ecke.

Es wird ein Gittergenerator vorgestellt, der im Programmpaket *UG* [11] implementiert worden ist und für eine große Klasse von Anwendungen adaptive, unstrukturierte Dreiecksgitter erzeugt. Die hier vorgestellte Advancing Front Methode ermöglicht die Auflösung der Ränder von Rechengebieten in einer vom Benutzer vorgegebenen Feinheit durch benutzerspezifizierte Knotendichtefunktionen. Desweiteren kann ebenso durch die Vorgabe einer Knotendichtefunktionen das Gitter im Innern des Rechengebietes lokal angepaßt werden. Die Erzeugung von Knoten im Rechengebiet wird durch vier Kontrollfunktionen gesteuert. Auf der Grundlage hierarchisch aufgebauter Datenstrukturen erfolgt eine effiziente Verwaltung der Advancing Front und es ist die Konstruktion von Rechengittern mit Dreiecken fast optimaler Qualität möglich.

Außerdem können mit dem Gittergenerator geometrische Gitter erzeugt werden. Geometrische Gitter werden zur Bestimmung von Lösungen elliptischer Randwertprobleme gebraucht, bei denen Singularitäten auftreten. Zur Auflösung einer Singularität wäre ein unendlich feines Gitter ideal, doch ist dies nicht durchführbar. Deshalb ist es von großer Bedeutung, wie verfeinert und wie und wann mit der Verfeinerung des Rechengitters aufgehört werden soll. Durch eine neue Definition des relativen Fehlers kann eine Verfeinerungsvorschrift zur Konstruktion von Rechengittern zur Bestimmung von Lösungen elliptischer Randwertprobleme mit Singularitäten mit optimaler Konvergenzordnung angegeben werden, um den Aufwand an Rechenzeit und Speicherkapazität effizient einzusetzen.

Es wird eine *a priori* Fehlerabschätzung bewiesen, aufgrund der der relative Fehler des Gradienten auf einem uniform verfeinerten Gitter optimale Konvergenzordnung besitzt, obwohl die bekannten Abschätzungen ein lokal verfeinertes Gitter fordern.

Desweiteren wird ein Kriterium angegeben, bis zu welchem Abstand ρ zur Singularität verfeinert werden muß, um optimale Konvergenzordnung des relativen Fehlers der Funktion und deren Gradienten in der L_∞ -Norm zu erzielen. In klassischen *a priori* Fehlerabschätzungen in der $L_\infty(\Omega)$ -Norm mit Rechengebiet $\Omega \subset \mathbb{R}^2$ wird gezeigt, daß der Gesamtfehler der Funktion und des Gradienten auf angepaßten Gittern optimale Konvergenzordnung besitzt. Lokale Aussagen sind in der $L_\infty(\Omega)$ -Norm nicht möglich. In dieser Arbeit wird bewiesen, daß durch Verwenden eines geometrischen Gitters lokale Abschätzungen in der L_∞ -Norm des relativen Fehlers der Funktion und deren Gradienten mit optimaler Konvergenzordnung möglich sind. Der Unterschied der hier vorgestellten Fehlerabschätzung zu den bekannten Methoden liegt darin, daß z.B. bei Halbierung der Maschenweite eines geometrischen Gitters der relative Fehler der Funktion auf *jedem* Dreieck des Gitters innerhalb des Gültigkeitsbereichs der Theorie geviertelt wird, während dies in der $L_\infty(\Omega)$ -Norm nur auf den maximalen Fehler der Gesamtlösung zutrifft.

Die Verfeinerungsvorschrift des Rechengitters zur Auflösung der Singularität und Bestimmen der Konvergenzordnung des relativen Fehlers in der L_∞ -Norm wird aus Überlegungen der Approximationstheorie motiviert. Der eindimensionale Fall birgt schon die Konstruktionsvorschrift und die Beweise sind mit elementaren Methoden durchführbar. Allerdings ist die Erweiterung der Beweise auf den zweidimensionalen Fall sehr kompliziert, da hier Dualitätsargumente und gewichtete Normen, wie sie z.B. in der Arbeit von Nitsche [75] eingeführt wurden, verwendet werden müssen. Außerdem werden Beweistechniken für lokale Abschätzungen, die keine Standardtechniken sind, herangezogen. Die grundlegenden, hier für lokale Abschätzungen verwendeten und erweiterten Beweistechniken wurden in einer Serie von Artikeln von Schatz und Wahlbin (siehe die Arbeiten [91], [92], [93], [94], [95], [96]) entwickelt.

Die Abschätzungen der relativen Fehler zeigen außerdem keine Abhängigkeit von der Stärke der Singularität und sind robust gegenüber Änderungen der rechten Seite.

Die hier vorgestellten numerischen Experimente zeigen, daß die Konvergenzordnungen der relativen Fehler optimal sind, und belegen, daß die Konstanten in den Abschätzungen der relativen Fehler vernünftigerweise klein sind. Die numerischen Experimente zeigen den Einfluß des "pollution"-Effekts und ermöglichen eine Abschätzung der oberen Gültigkeitsgrenze der hier vorgestellten Abschätzungen der relativen Fehler.

Die hier vorgestellte Arbeit eröffnet vielfältige Perspektiven. Bei der Entwicklung des Gittergenerators ist eine Datenstruktur entstanden, die auch die Erzeugung von Tetraedergittern mit wenig Aufwand ermöglicht. Zur Erzeugung von Hexaedergittern ist die Datenstruktur ebenfalls geeignet, doch liegen in der Konstruktion von Hexaedergittern noch Probleme, die durch die Datenstruktur nur bedingt gelöst werden können. Hier besteht Forschungsbedarf und es ist sicherlich lohnenswert, das hier vorgestellte Konzept hierarchischer Listen auf drei Dimensionen zu erweitern. Die Datenstruktur ermöglicht ebenfalls einen leichten Übergang zur Parallelisierung und somit wäre eine Weiterentwicklung der in dieser Arbeit eingeführten Advancing Front Methode in diese Richtung wünschenswert.

Die hier vorgestellte Theorie zur *a priori* Fehlerabschätzung der relativen Fehler von Lösungen und deren Gradienten von elliptischen Randwertproblemen mit Singularitäten ermöglicht eine Vielzahl weiterführender Arbeiten. Es ist sicherlich lohnenswert, die hier vorgestellten *a priori*-Fehlerabschätzungen mit dem Gittergenerator zu koppeln. Durch eine *a posteriori*-Betrachtung könnten Singularitäten detektiert und danach in einer Umgebung der Singularität geometrische Gitter erzeugt werden. Aus den hier vorgestellten Theoremen könnten Abschätzungen von Spannungsintensitätsfaktoren vorgenommen werden. Es sind Untersuchungen von degenerierten elliptischen Randwertproblemen oder Problemen mit springenden Koeffizienten vorzunehmen und der Einfluß geometrischer Gitter auf die Fehler zu untersuchen. Eine weitere Anwendungsmöglichkeit besteht sicherlich in Abschätzungen von Green's-Funktionen elliptischer Randwertprobleme. Außerdem sollte untersucht werden, ob aus den hier vorgestellten Fehlerabschätzungen ein Verfahren zur Abschätzung der Potenz einer Singularität entwickelt werden kann.

Kapitel 1

Einleitung

Die automatische Erzeugung von Rechengittern zur Lösung partieller Differentialgleichungen mit finiten Elementen oder finiten Volumen Verfahren ist auch in zwei Raumdimensionen mit großen Schwierigkeiten verbunden. Insbesondere die Vernetzung komplexer Geometrien mit vielen inneren Rändern ist nach wie vor ein großes Problem, wenn dabei Elemente optimaler Qualität erzeugt werden sollen. Außerdem werden für die meisten Problemklassen des wissenschaftlichen Rechnens adaptive Gitter verlangt.

Erstes Ziel dieser Arbeit ist es, einen Gittergenerator zur Erzeugung von Dreiecksgittern auf komplexen zweidimensionalen Gebieten mit vielen inneren Rändern und möglichst optimaler Elementqualität für eine große Klasse von Anwendungen zu entwickeln. Optimale Dreiecke sind dadurch gekennzeichnet, daß sie keine spitzen oder stumpfen Winkel enthalten. Besonders Dreiecke mit stumpfen Winkeln können die Diskretisierungseigenschaften finiter Elemente- oder finiter Volumen-Verfahren stark beeinträchtigen. In einer Arbeit von Babuška und Aziz [3] wird gezeigt, daß der Interpolationsfehler ansteigt, falls der maximale Winkel α eines Dreiecks nahe bei $\alpha \approx \pi$ liegt. Enthält ein Dreieck einen sehr kleinen Winkel α , so wird der Interpolationsfehler nach [3] zwar nicht ansteigen, doch wird im Artikel von Fried [38] gezeigt, daß die Konditionszahl der Steifigkeitsmatrix sich wie $1/\sin \alpha$, mit α als kleinstem Winkel eines Dreiecks, verhält. Damit wird deutlich, wie wichtig die Elementqualität eines Rechengitters ist.

Es gibt eine Vielzahl verschiedener Techniken, um Dreiecks- oder Vierecksnetze für unterschiedlichste Anwendungen zu erzeugen. Beispiele hierfür sind Voronoi-Diagramme bzw. Delaunay-Methoden, Quadtree-Algorithmen, blockstrukturierte Gittergeneratoren oder Advancing Front Methoden. Eine wichtige Frage war, welche Methode für die Problemklassen des Programmpakets *UG* [11] am besten geeignet ist. Deshalb wurden die wichtigsten Gittergenerierungsverfahren sorgfältig

analysiert und als geeignetster Kandidat die Advancing Front Methode gefunden. Aufbauend auf der Advancing Front Methode von Peraire et. al. [80] und Paolini und Verdi [79] wurde eine neue Advancing Front Methode entwickelt, mit der es möglich ist, komplizierte Gebiete mit vielen inneren Rändern automatisch und adaptiv zu vernetzen. Die automatische Kontrolle der Elementqualität dieser Methode gewährleistet bereits bei der Erzeugung des Gitters fast optimale Qualität der Dreieckselemente. Eine Besonderheit des hier vorgestellten Gittergenerators liegt im Unterschied zu den beiden in [80] und [79] vorgestellten Verfahren in den Auswahlkriterien zur Erzeugung neuer Punkte und Dreiecke im Rechengebiet und in der Datenstruktur zur Steuerung des Advancing Front Algorithmus. Aufgrund der besonderen Datenstruktur ist ein Gittergenerator entstanden, der, im Gegensatz zu den meisten anderen Vernetzern, mit nur wenigen Kontrollfunktionen und ohne Hintergrundnetz adaptive Dreiecks-Gitter erzeugen kann. Die hier vorgestellte Advancing Front Methode kann aufgrund der sehr allgemein gehaltenen Datenstruktur auf 3 Raumdimensionen zur Erzeugung von Tetraedernetzen mit wenig Aufwand erweitert werden. Prinzipiell besteht die Möglichkeit, die hier beschriebene Advancing Front Methode zur Erzeugung von Hexadernetzen zu verwenden. Die Erzeugung von Hexadernetzen ist allerdings kompliziert und aufwendig und es ist zu erwarten, daß die in dieser Arbeit diskutierte Advancing Front Technik nur mit Einschränkungen an die möglichen Orientierungen von Flächen im dreidimensionalen Raum angewendet werden kann. Außerdem ist der Gittergenerator so konzipiert, daß er inhärent die Möglichkeit zur Parallelisierung besitzt.

Das zweite Ziel dieser Arbeit ist die Untersuchung relativer Fehler in singular gestörten elliptischen Randwertaufgaben, deren Lösungen und deren Ableitungen ein potenzabhängiges Verhalten zeigen. Singularitäten der Lösungen oder Singularitäten der Ableitungen der Lösungen partieller Differentialgleichungen verursachen erhebliche numerische Schwierigkeiten. Die Probleme liegen nicht nur in der Approximation der Singularität, sondern auch in der durch sie hervorgerufenen Verschlechterung des Fehlers in weiter entfernt liegenden Regionen des Rechengebietes. Dieses Verhalten des Fehlers ist selbst dann zu beobachten, wenn die Lösung in weiter entfernt liegenden Regionen des Rechengebietes glatt ist. Dieser Störeffekt ist wohlbekannt und wird als "pollution effect" bezeichnet. Eine Methode zur Behebung dieser Schwierigkeiten ist die Verwendung lokal angepaßter Gitter, insbesondere in der Umgebung der Singularität. Die Verfeinerung kann entweder vor dem eigentlichen Lösungsprozeß, d. h. *a priori*, oder adaptiv während des Lösungsprozeß, d. h. *a posteriori*, vorgenommen werden. Die Kon-

struktion von *a priori* Verfeinerungsstrategien hängt von der *a priori* Kenntnis der Singularität und von den Eigenschaften der Lösung ab. In dieser Arbeit werden Klassen von Funktionen betrachtet, deren Ableitungen proportional zu einer Potenz des Abstandes zu einem Punkt, einer Linie oder Oberfläche begrenzt sind. Diese Funktionen werden als potenzabhängige Funktionen und die Singularitäten als potenzabhängige Singularitäten bezeichnet. Die betrachteten Funktionsklassen spielen in den nachfolgend aufgeführten praktischen Problemen eine wichtige Rolle, z. B. in elliptischen Randwertproblemen auf Gebieten mit einspringenden Ecken oder auf Polyedern (siehe hierzu z. B. [48]). Weitere Beispiele sind Lösungen elliptischer Randwertaufgaben mit springenden Koeffizienten in 2 Raumdimensionen, siehe z. B. Kellog [57], die Bestimmung der Greens-Funktion elliptischer Randwertprobleme oder degenerierte elliptische Randwertprobleme, siehe z. B. in French [37], Khoromskij [43], und Marini und Pietra [66]. Eine andere wichtige Problemklasse stellen nichtlineare elliptische Randwertaufgaben, wie sie z. B. in Tolksdorf [107] beschrieben werden, dar. In vielen Fällen ist bekannt, daß die Lösung potenzabhängiges Verhalten zeigt, die Exponenten sind allerdings unbestimmt. Ein spezieller Fall ist der degenerierte Laplace-Operator, der z. B. in [29] studiert wird. Dort wird auch ein Verfahren zur Bestimmung der Exponenten vorgestellt.

In dieser Arbeit wird in erster Linie die Effektivität einer Verfeinerungsstrategie auf die relativen Fehler der Funktion und ihres Gradienten bei Approximation der Lösung von elliptischen Randwertaufgaben mit der Methode der finiten Elemente untersucht. Die zugrundegelegte Verfeinerungsstrategie beruht auf dem oft benutzten geometrischen Gitter, das in dieser Arbeit mit nur zwei Parametern definiert werden kann. Die hier vorgestellten geometrischen Gitter zeigen einige neue und interessante Eigenschaften und eignen sich als optimale Gitter zur Untersuchung verschiedener relativer Fehler von Lösungen nahe bei potenzabhängigen Singularitäten. Die hier vorgestellten geometrischen Gitter können sehr effektiv in Kombination mit einem *a posteriori* Fehlerschätzer und einer adaptiven Verfeinerung benutzt werden, um *a priori* Gitter zur Auflösung von Singularitäten zu konstruieren. Motiviert wird diese Vorgehensweise durch Überlegungen aus der Approximationstheorie. In drei neuen Theoremen wird bewiesen, daß bei wohldefinierten Annahmen für die Lösung elliptischer Randwertprobleme der relative Fehler des Gradienten in der L_2 -Norm auf uniform verfeinerten Gittern und die relativen Fehler der Lösungsfunktion und des zugehörigen Gradienten auf geometrisch verfeinerten Gittern in der lokalen L_∞ -Norm optimale Konvergenzordnung besitzen. Die Beweise hierzu werden in [53] veröffentlicht.

Die Arbeit ist folgendermaßen gegliedert:

In Kapitel 2 werden die grundlegenden geometrische Definitionen eingeführt und die Gründe aufgezeigt, warum eine Advancing Front Technik zur Gittererzeugung für die Problemklassen des Programmpakets *UG* [11] am besten geeignet ist.

Die hier verwendeten Methoden zur Beschreibung von Rechengebieten und elementare Datenstrukturen werden in Kapitel 3.3 vorgestellt. Dieses Kapitel eignet sich zum Nachschlagen der im darauf folgenden Kapitel beschriebenen Listenkonzepte.

Kapitel 4 spielt eine zentrale Rolle in der Beschreibung des Gittergenerators. Es wird im ersten Abschnitt ein allgemeines Listenkonzept zur Verwaltung der Advancing Front vorgestellt, das in dieser Form nach Kenntnis des Autors noch nicht zur Gittergenerierung verwendet worden ist.

Es folgt in den Abschnitten 4.2 und 4.3 die Beschreibung der Erzeugung von Randknoten und von Knoten und Dreiecken im Innern des Rechengebietes. Die Position der Randknoten und die Größe der Dreiecke im Rechengebiet können vom Benutzer durch die Vorgabe einer Maschenweitekontrollfunktion auf einfache Weise definiert werden.

Daran anknüpfend werden die vier Kontrollfunktionen zur Konstruktion von Elementen optimaler Qualität vorgestellt und die Aktualisierung der aktiven Front nach Erzeugung eines Elements beschrieben. Die 4 Kontrollfunktionen in Zusammenspiel mit dem Listenkonzept sind essentiell zur Erzeugung von Dreiecksgittern mit Dreiecken optimaler Güte auf komplexen Gebieten mit vielen inneren Rändern. Die Qualität der Elemente kann nach der Gittererzeugung durch die wohlbekannt Laplace-Glättung (Abschnitt 4.5.1) im Falle konvexer Geometrien verbessert werden. Bei nichtkonvexen Gebieten versagt diese Methode und wird durch die in Abschnitt 4.5.2 vorgestellte modifizierte Laplace-Glättung ersetzt. Die modifizierte Laplace-Glättung verhindert, daß der zu verschiebende Punkt nach der Verschiebung außerhalb des Rechengebietes liegt.

In Abschnitt 4.6 wird anhand der Triangulierung der Nordseeküste demonstriert, wie durch Verwenden einer Maschenweitekontrollfunktion adaptive Gitter erzeugt werden. Daran anschließend werden Anwendungen des Gittergenerators auf kom-

plexe Geometrien zur Konstruktion von Grobgittern für Mehrgittermethoden gezeigt.

In Kapitel 5 werden die mathematischen Grundlagen und das Modellproblem der *a priori*-Fehlerabschätzung vorgestellt. Die verwendeten Sobolev-Räume sind auf dem Raum L_p aufgebaut (Abschnitt 5.1). Das den Untersuchungen zugrundegelegte Dirichlet-Randwertproblem wird in Abschnitt 5.2 beschrieben. In den Abschnitten 5.3 und 5.4 werden Regularitätseigenschaften des Dirichlet-Problems und Approximationseigenschaften stückweise linearer finiter Elemente beschrieben. Es folgt eine Diskussion des relativen Fehlers in Abschnitt 5.5.

Das neue Theorem zur *a priori*-Abschätzung des relativen Fehlers des Gradienten in der L_2 -Norm wird in Kapitel 6 vorgestellt. Eine Diskussion klassischer Fehlerabschätzungen in $L_2(\Omega)$ zeigt, daß optimale Konvergenz nur auf lokal verfeinerten Gittern möglich ist (Abschnitt 6.1 und 6.2). In Abschnitt 6.3 folgt die Formulierung und der Beweis der *a priori*-Abschätzung des relativen Fehlers des Gradienten in der L_2 -Norm. Es wird gezeigt, daß der relative Fehler des Gradienten optimale Konvergenzordnung auf uniform verfeinerten Gittern besitzt.

Das Kapitel 7 enthält die neue *a priori*-Fehlerabschätzung in der lokalen L_∞ -Norm des relativen Fehlers der Funktion und des Gradienten. In Abschnitt 7.1 werden bekannte Fehlerabschätzungen in der L_∞ -Norm dargestellt und in Abschnitt 7.2 die zum Beweis der folgenden *a priori* Fehlerabschätzungen notwendigen Hilfssätze eingeführt. Es folgt in Abschnitt 7.3 die Formulierung und Diskussion der *a priori*-Fehlerabschätzung in der lokalen L_∞ -Norm und in Abschnitt 7.4 folgen die Beweise.

Numerischen Experimente zur Überprüfung der Optimalität der Fehlerabschätzungen werden in Kapitel 8 vorgestellt. Abschnitt 8.1 beschreibt das den Experimenten zugrundegelegte Randwertproblem und stellt die verwendeten Singularitätenfunktionen vor. Es folgt ein Algorithmus zur Konstruktion des hier verwendeten geometrischen Gitters (Abschnitt 8.2) und es werden Funktionenklassen in Abschnitt 8.3 vorgestellt, die die notwendigen Annahmen der Theoreme in Kapitel 6 und Kapitel 7 erfüllen. In den Vorbemerkungen zu den numerischen Experimenten (Abschnitt 8.4) werden die Parameter, mit denen die Experimente gesteuert werden, erläutert. Die numerischen Experimente werden in den Abschnitten 8.5 bis 8.9 beschrieben.

Kapitel 2

Gittergenerierungsverfahren

Dieses Kapitel erläutert die Grundlagen der Gittergenerierungsverfahren, die zur Entwicklung des in dieser Arbeit vorgestellten Gittergenerators untersucht worden sind. In einer kurzen Einleitung werden die Problemklassen des Programmpakets *UG* und die untersuchten Gittergenerierungsmethoden vorgestellt und gezeigt, warum strukturierte Gitter nicht verwendet werden können. Daran anknüpfend werden im ersten Abschnitt die zur Gittergenerierung notwendigen geometrischen Objekte vorgestellt. Danach folgt in Abschnitt 2.2 eine Diskussion und Bewertung der Delaunay-Triangulierung. Im dritten Abschnitt wird die Quadtree-Methode zur Gittererzeugung untersucht und aufgezeigt, warum diese Methode bei komplexen Gebieten mit vielen inneren Rändern nur bedingt anwendbar ist. Im letzten Abschnitt wird die Advancing Front Technik anhand zweier Beispiele erläutert und mit der hier entwickelten neuen Advancing Front Technik verglichen.

In der numerischen Behandlung partieller Differentialgleichungen ist die Generierung geeigneter Rechengitter eine kritische Aufgabe. Besondere Probleme bereitet die Vernetzung komplexer Gebiete mit vielen inneren Rändern. Es ist ein noch ungelöstes Problem, optimale Qualität der Elemente eines Rechengitters für ein komplexes Gebiet mit vielen inneren Rändern zu erzielen. Benutzt der Anwender Diskretisierungsmethoden wie finite Elemente oder finite Volumen, wird die Approximationseigenschaft des Verfahrens entscheidend von den Winkeln der Elemente beeinflusst. Ein wichtiges Qualitätsmaß eines Gitters mit Dreieckselementen liegt in der Vermeidung von spitzwinkligen bzw. stumpfwinkligen Dreiecken. Der hier vorgestellte Gittergenerator ist in der Lage, für die untersuchten, in Kapitel 4.6 vorgestellten Beispiele, Dreiecke mit Winkeln α im Bereich $20^\circ < \alpha < 132^\circ$ zu erzeugen.

Ziel und Ergebnis dieser Arbeit ist die Entwicklung eines Gittergenerators für das Programmpaket *UG* (*UG* als Abkürzung von "unstrukturierte Gitter") [11],

der diesen Forderungen entspricht. Dieser neu entwickelte Gittergenerator ist für die unterschiedlichsten Anwendungen der Problemklassenbibliotheken von *UG* geeignet. Dazu gehören Anwendungen in den Bereichen Diffusion, Konvektion-Diffusion mit finiten Elementen und finiten Volumen, lineare Elastizität, inkompressible Navier-Stokes Gleichungen, dichtegetriebene Strömungen und Zweiphasenströmungen in porösen Medien, Euler und kompressible Navier-Stokes Gleichungen, lineare und nichtlineare Strukturmechanik und Bingham-Fluide.

Eine zentrale Frage ist, welches Verfahren zur Gittergenerierung verwendet wird. Grundsätzlich kann zwischen strukturierten und unstrukturierten Gittern unterschieden werden. Die nachfolgend aufgeführten Methoden stellen die wichtigsten Gittergenerierungsverfahren dar. Es existieren weitere, speziell an die betrachteten Problem angepasste Verfahren, die an dieser Stelle jedoch nicht betrachtet werden. Folgende Methoden zur Gittergenerierung stehen prinzipiell zur Verfügung:

- strukturierte Gittergenerierung, z. B.
 - blockstrukturierte Verfahren
 - elliptische Gittergenerierung
- unstrukturierte Gittergenerierung, z. B.
 - Delaunay-Verfahren
 - Advancing-Front-Technik
 - Quadtree-Verfahren

Die geeignetsten Verfahren stellen das Delaunay-, das Quadtree-Verfahren und die Advancing-Front-Technik dar. Deshalb werden diese Methoden in den nachfolgenden Kapiteln genauer beschrieben und erläutert, warum die Advancing-Front-Technik am besten geeignet ist. Als Ergänzung zur noch folgenden Darstellung der verschiedenen Gittergenerierungsverfahren wird auf die Übersichtsartikel [35], [51], [54], [90], [99], [102], [104] und [106] verwiesen.

Prinzipiell werden Rechengitter in zwei Hauptgruppen unterteilt, in *strukturierte* und *unstrukturierte* Gitter.

Ein strukturiertes Gitter entsteht dann, wenn die Gitterpunkte durch eine affine Abbildung in \mathbb{Z}^n eingebettet werden können. Hier wird der Fall $n = 2$ betrachtet.

Die Knoten des Gitters besitzen eine isomorphe lokale Nachbarschaft mit Ausnahme der Randknoten. In einem *unstrukturierten* Gitter sind die lokalen Nachbarschaftsverhältnisse willkürlich variierend.

In beiden Fällen bestehen die Elemente aus Polytopen, die typischerweise Dreiecke oder Vierecke sind.

Strukturierte Gitter werden im allgemeinen zur Vernetzung einfacher Geometrien verwendet. Ihr großer Vorteil besteht darin, daß sie wenig Arbeitsspeicher benötigen, da die Koordinaten während des Erzeugens der Elemente berechnet werden können und nicht dauerhaft abgespeichert werden müssen. Außerdem können strukturierte Gitter mit geringem Rechenaufwand erzeugt werden und es ist eine direkte Kontrolle der Elementgröße möglich. Strukturierte Gitter haben den numerischen Vorteil, daß die meisten Löser auf diesen Gittern besonders effizient arbeiten. Diese Verfahren zur Erzeugung von Gittern versagen jedoch bei der Vernetzung komplizierter Geometrien und können deshalb für viele realistische Anwendungen entweder gar nicht oder nur sehr unbefriedigend eingesetzt werden. Es gibt verschiedene Techniken, um das Anwendungsspektrum strukturierter Gitter zu erweitern. Die am meisten verwendeten Techniken zur Gittererzeugung für komplexe Gebiete sind algebraische Methoden, speziell angepaßte konforme Abbildungen oder numerische Methoden, wie das Lösen von Differentialgleichungen. Dieser generelle Mangel strukturierter Gittererzeugung ist bis heute trotz intensiven Forschens noch nicht befriedigend gelöst. Eine Einführung in diese Problematik ist z.B. in [23], [56], [60] und [105] zu finden.

Der Trend weist deutlich in Richtung unstrukturierter Gittergenerierung [63], [104], [106]. Mit unstrukturierten Gittern ist es möglich, komplizierte Gebiete aufzulösen und adaptiv zu vernetzen. Ziel dabei ist, mit einem Verfahren möglichst viele Problemklassen zu erfassen.

2.1 Geometrische Definitionen

In diesem Abschnitt werden die zum Verständnis der in den folgenden Abschnitten vorgestellten Gittergenerierungsverfahren notwendigen geometrischen Objekte in Anlehnung an [81] eingeführt. Die geometrischen Objekte, auf die sich die Definitionen beziehen, bestehen aus Punkten des zweidimensionalen euklidischen Raums \mathbb{R}^2 mit der bekannten euklidischen Metrik $(x^2 + y^2)^{\frac{1}{2}}$, $x, y \in \mathbb{R}^2$. Außerdem werden die charakteristischen Eigenschaften der hier verwendeten geometri-

schen Randbeschreibungen vorgestellt. Die geometrischen Randbeschreibungen werden in 4 Klassen eingeteilt.

Definition 2.1.1 (Verbindungsstrecke)

Gegeben seien zwei Elemente $q_1, q_2 \in \mathbb{R}^2$. Dann ist die Verbindungsstrecke von $q_1, q_2 \in \mathbb{R}^2$ durch den folgenden Ausdruck

$$\alpha q_1 + (1 - \alpha) q_2 \quad \text{mit } \alpha \in \mathbb{R}, \quad 0 \leq \alpha \leq 1$$

gegeben und wird als *Segment* bezeichnet. Ein Segment wird in der Form $\overline{q_1 q_2}$ als ungeordnetes Paar geschrieben.

Definition 2.1.2 (Konvexe Menge)

Eine Teilmenge $D \subset \mathbb{R}^2$ ist konvex, falls für jede beliebige Kombination von 2 Elementen $q_1 \in \mathbb{R}^2$ und $q_2 \in \mathbb{R}^2$ das Segment $\overline{q_1 q_2}$ vollständig in D enthalten ist, d.h. $\overline{q_1 q_2} \subset D$.

Definition 2.1.3 (Konvexe Hülle)

Die *konvexe Hülle* $C(D)$ der Menge $D \subset \mathbb{R}^2$ ist die Menge aller reellen Linearkombinationen

$$\begin{aligned} q &= \alpha_1 q_1 + \dots + \alpha_n q_n, \\ \alpha_1, \dots, \alpha_n &\text{ reell,} \\ q_1, \dots, q_n &\in D \end{aligned}$$

mit

$$\sum_{k=1}^n \alpha_k = 1, \quad \alpha_k \geq 0$$

Definition 2.1.4 (Polygon)

Ein Polygon in \mathbb{R}^2 besteht aus einer endlichen Menge von Segmenten, wobei die beiden äußersten Endpunkte von genau zwei Segmenten berührt werden und keine Teilmengen der Segmente gleich sind. Die Segmente eines Polygons werden als Kanten und deren äußerste Endpunkte als Vertices oder Knoten bezeichnet. Die Anzahl der Vertices und Kanten eines Polygons ist identisch.

Ein Polygon wird als *einfach* bezeichnet, falls sich nur Paare aufeinanderfolgender Kanten einen Knoten teilen. Ein einfaches Polygon teilt die Ebene in zwei disjunkte Teilgebiete, in das innere (begrenzte) und das äußere (unbegrenzte) Gebiet, das durch das Polygon getrennt wird. Ein einfaches Polygon ist konvex, falls das innere Gebiet eine konvexe Menge ist.

Definition 2.1.5 (Ebener Graph)

Ein Graph $G = (V, K)$ mit der Vertexmenge V und der Kantenmenge K ist eben, falls er in eine Ebene ohne Überschneidungen eingebettet werden kann. Die Einbettung ebener Verbindungsstrecken des Graphen G wird ebene Teilaufspaltung genannt. Wird mit v, e, f die Anzahl der Vertices, Kanten und Teilgebiete bezeichnet, so sind diese drei Parameter über die Beziehung [16] (Euler's Formel)

$$v - e + f = 2$$

miteinander verknüpft.

Definition 2.1.6 (Triangulierung)

Eine ebene Teilaufspaltung ist eine Triangulierung, falls alle ihre beschränkten Teilgebiete Dreiecke sind. Eine Triangulierung einer endlichen Menge S von Punkten ist ein ebener Graph in S mit maximaler Anzahl von Kanten.

Eine äquivalente Definition einer Triangulierung lautet: Eine Triangulierung von S entsteht durch die Verbindung der Punkte von S durch sich nicht überschneidende Verbindungsstrecken. Jedes Teilgebiet im Innern der konvexen Hülle von S ist ein Dreieck.

Definition 2.1.7 (Steiner-Triangulierung, Steiner-Punkte)

Bei einer *Steiner-Triangulierung* einer Punktmenge in der Ebene können zusätzlich Punkte hinzugefügt werden. Die zur Menge der ursprünglich vorhandenen Punkte hinzugefügten Punkte werden als *Steiner-Punkte* bezeichnet.

Bemerkung 2.1.1 (Steiner-Triangulierung)

Bei einer *Steiner-Triangulierung* (siehe Definition 2.1.7) einer Punktmenge S in der Ebene können Punkte außerhalb der konvexen Hülle von S hinzugefügt werden. Die vorhandene Punktmenge ist folglich eine Teilmenge der Knoten des Gitters und der Rand des Gebietes ist zwar konvex, kann jedoch größer sein, als die konvexe Hülle. Es ist möglich, jede Punktmenge einer Ebene ohne Steiner-Punkte zu vernetzen. Die Hinzunahme von Steiner-Punkten ist deshalb nur dann sinnvoll, wenn man bestimmte Qualitätskriterien erfüllen will, wie z.B. die Beschränkung der maximalen und minimalen Winkel der Elemente des Gitters. Ein typisches Problem, bei dem keine Steiner-Punkte erlaubt sind, besteht in der Triangulierung einer Punktmenge bei gleichzeitiger Minimierung des maximalen Winkels [12]. Ein Beispiel einer Triangulierung mit Steiner-Punkten ist die Vernetzung eines Polygons mit n Knoten mit der Bedingung, daß die Winkel der Dreiecke kleiner als 90° sind [32].

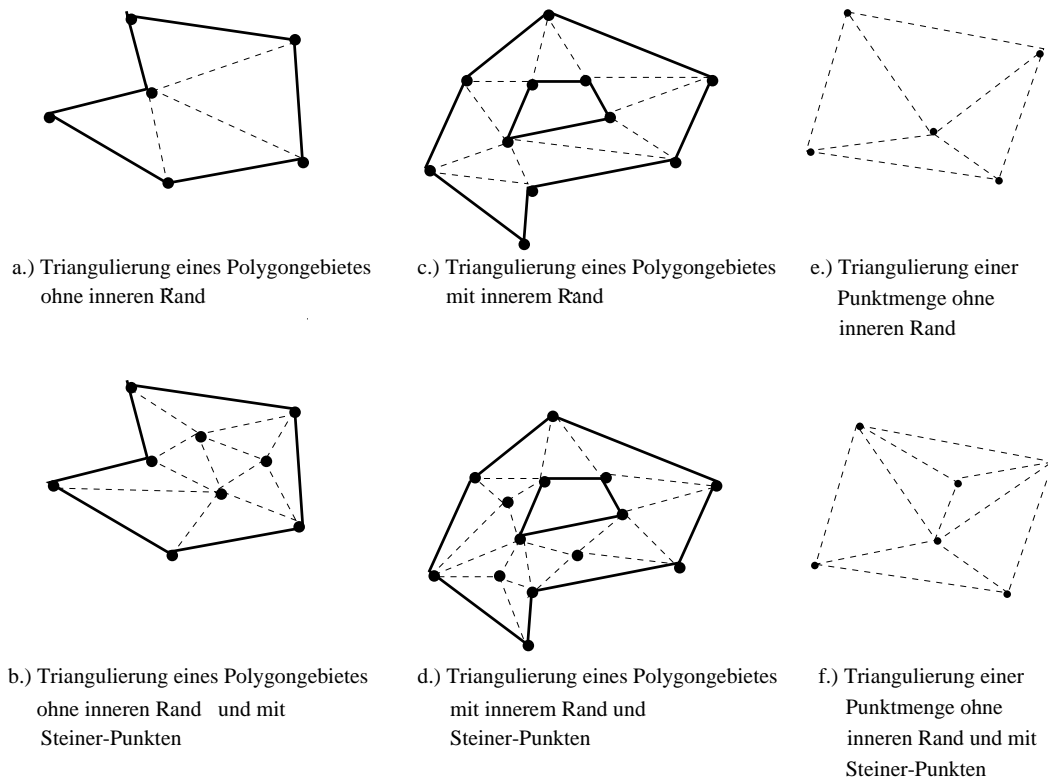


Abbildung 2.1: Triangulierungen ohne und mit Steiner Punkten eines Polygons, eines Polygons mit inneren Rändern und einer Punktmenge

Die Randbeschreibung der 4 in dieser Arbeit verwendeten Gebiete ist eingeteilt in:

1. *Einfaches Polygonebiet*: Das Rechengebiet ist eine ebene Fläche, deren Ränder durch einfache geschlossene polygonale Kurven beschrieben werden, siehe hierzu die Beispiele in Abbildung 2.1a und in Abbildung 2.1b.
2. *Polygonales Gebiet mit innerem Rand*: Dieses unterscheidet sich von dem zuvor vorgestellten Polygonebiet dadurch, daß der Rand aus mehreren disjunkten Jordan-Kurven besteht. In den Abbildung 2.1c und 2.1d ist ein Polygonebiet mit einem inneren Rand mit und ohne Steiner-Punkte dargestellt.
3. *Punktmenge*: Die Eingabe für den Gittergenerator besteht in diesem Fall aus einer Punktmenge in der Ebene. Die Punktmenge stellt die Punkte der Triangulation dar und der Rand des Gitters ist die konvexe Hülle. Durch die Hinzunahme von Steiner-Punkten werden bestimmte Qualitätskriterien des Gitters (siehe Definition 2.1.7 und die Bemerkung dazu) erfüllt. Eine Punktmenge ohne und mit Steiner-Punkten ist in den Abbildungen 2.1e und 2.1f zu finden.
4. *Parameterdarstellung*: Der Rand eines Gebietes wird durch Parameterfunktionen dargestellt. Die Punkte der Kurven mit den Koordinaten $(x, y) \in \mathbb{R}^2$

werden in der Form $x = \phi(\lambda) = x(\lambda)$, $y = \psi(\lambda) = y(\lambda)$, $\lambda \in [\varepsilon_1, \varepsilon_2]$ mit der endlichen Teilmenge $[\varepsilon_1, \varepsilon_2] \subset \mathbb{R}$ dargestellt.

2.2 Delaunay Triangulierung

In diesem Kapitel wird die Delaunay-Triangulierung diskutiert und gezeigt, warum dieses Verfahren bei komplexen Gebieten mit inneren Rändern nur unzureichende Gitter erzeugen kann.

Eine Delaunay-Triangulierung bezeichnet ein Verfahren, bei dem aus einer gegebenen Punktmenge S in der Ebene ein Rechengitter mit Dreieckselementen erzeugt wird. Als Nebenbedingung wird dabei die Summe der kleinsten Winkel der Dreiecke maximiert. Der polygonale Rand der Punktmenge besteht aus der konvexen Hülle von S . Die Zerlegung einer durch Polygone berandeten Fläche in Dreiecke geht auf Dirichlet zurück [28]. Die Zerlegung einer Ebene in Gebiete gleicher nächster Nachbarn wird als Voronoi-Diagramm [110] bezeichnet. In einer Arbeit von Delaunay aus dem Jahre 1934 [26] wird gezeigt, daß der zum Voronoi-Diagramm duale Graph eine Triangulierung darstellt. Eine detaillierte Diskussion dieses Verfahrens sowie einige Bemerkungen zu Algorithmen zur Konstruktion von Voronoi-Diagrammen sind anschaulich in [81] dargestellt. Eine Übersicht über Delaunay-Triangulierungen und Voronoi-Diagramme vermitteln die Artikel von Aurenhammer [2] und Fortune [36].

Das Delaunay-Verfahren ist auf n Raumdimensionen erweiterbar [113]. Zur Veranschaulichung genügt es allerdings, die Konstruktion von Dreiecksgittern in der Ebene zu betrachten. Im folgenden wird deshalb die Gittergenerierung für zweidimensionale ebene Geometrien diskutiert.

Zur Konstruktion eines Rechengitters wird die Delaunay-Triangulation als das ebene duale Problem eines Voronoi-Diagramms aufgefaßt. Ein Voronoi-Diagramm entsteht aus der Einteilung einer Ebene in polygonale Zellen. Jedem Punkt $P_m \in S$ wird eine polygonal berandete Fläche zugewiesen, innerhalb der jeder Punkt einen kleineren Abstand zu $P_m \in S$ als zu jedem anderen Punkt $P_k \in S$ besitzt. Diese Unterteilung der zu vernetzenden Geometrie führt auf nichtüberlappende, konvexe Polygone.

Wird mit V_i das i -te Polygon bezeichnet, so bildet die Gesamtheit aller Polygone V_i das Voronoi-Diagramm V und wird formal geschrieben als:

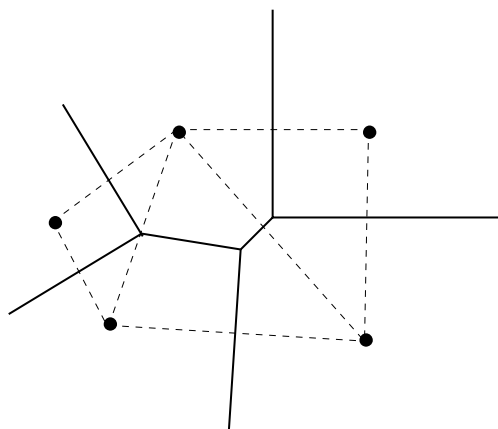


Abbildung 2.2: Triangulierung einer Punktmenge mit Voronoi-Polygonen

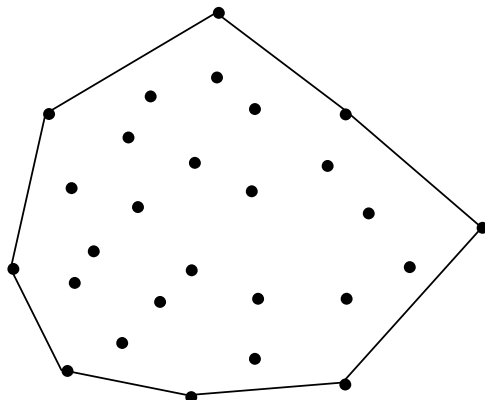


Abbildung 2.3: Punktmenge mit konvexer Hülle

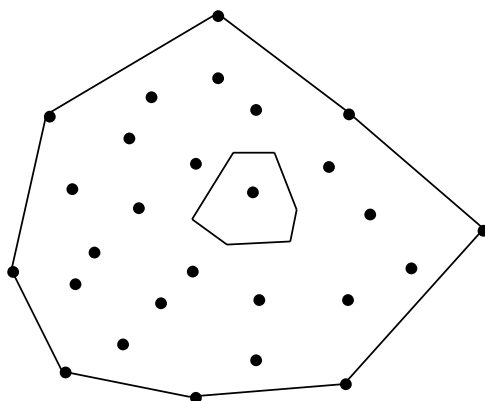


Abbildung 2.4: Voronoi-Polygon exemplarisch für einen Punkt

$$V = \{\cup V_i, i = 1, \dots, N\}$$

$$\text{mit } V_i = \{x \in \mathbb{R}^2 : |x - P_i| < |x - P_k| \quad \forall k \neq i\}$$

Die geometrische Konstruktion des Voronoi-Diagramms im zweidimensionalen Fall erfolgt durch die Konstruktion der Mittelsenkrechten auf der Verbindungslinie zweier benachbarter Punkte der Menge S . Die Verbindungen der Schnittpunkte der Mittelsenkrechten ergeben das Voronoi-Diagramm (siehe hierzu die Abbildungen 2.2, 2.3 und 2.4).

Die Dreieckselemente des Rechengitters entstehen durch die Verbindungsgeraden (gestrichelte Linie in Abbildung 2.2) aller Punktpaare, die sich eine Polygonseite teilen. Ein Voronoi-Polygon für einen Punkt wird exemplarisch in Abbildung 2.3 dargestellt. Der Rand des Rechengebietes ist durch die konvexe Hülle der Punktmenge S gegeben (Abbildung 2.4).

Es gibt mittlerweile eine Vielzahl von Algorithmen, die auf der Delaunay/Voronoi-Technik beruhen, z.B. [17], [36], [44], [100], [101], [103], [113], [115], [121]. Das am häufigsten benutzte Verfahren ist das Verfahren von Bowyer [17]. Es ist einfach zu implementieren und sowohl in 2d als auch in 3d einsetzbar. Ein weiteres Verfahren, das auch für nicht-konvexe Gebiete geeignet ist, wird in [64] beschrieben.

Die Delaunay-Triangulierung zeigt allerdings Schwächen bei der Vernetzung komplexer nichtkonvexer Gebiete mit inneren Rändern. Im allgemeinen können Ränder mittels der Delaunay-Triangulierung nur unzureichend aufgelöst werden. Es gibt verschiedene Ansätze, dieses Problem zu umgehen. Von Baker [8] wird vorgeschlagen, den Rand des Gebietes abzuspalten und nur das verbleibende Innere mit der Delaunay-Methode zu vernetzen. Der Rand wird anschließend stufenförmig aufgelöst. Eine weitere Möglichkeit besteht darin, zuerst ein vorläufiges Gitter zu erzeugen und danach mit einer wohldefinierten Anzahl von Transformationen die Elemente zu restaurieren, die außerhalb des Rechengebietes liegen. Hinweise zu den vorgestellten Qualitätsverbesserungsverfahren sowie Einblicke in andere Methoden sind in [40], [41], [42] [114] und [42] zu finden.

Ein weiterer Nachteil des Delaunay-Verfahrens liegt darin, daß aufbauend auf einer vorher bestimmten Punktmenge die Dreiecke erzeugt werden. Es können Dreieckselemente mit spitzen oder stumpfen Winkels entstehen, da a priori kaum Aussagen über die Güte der Punkteverteilung möglich sind. Es gibt jedoch Strategien,

um die Gitterqualität durch Hinzufügen neuer Punkte (Steiner-Punkte) zu verbessern. Die einfachste Methode besteht darin, neue Punkte innerhalb des Umkreises von großen Dreiecken bzw. Dreiecken mit spitzen oder stumpfen Winkeln einzufügen [55]. Eine modifizierte Form dieser Vorgehensweise ist in [84] zu finden. Dort wird eine Dichtekontrollfunktion für die zu vernetzenden Punkte entwickelt, die sicherstellt, daß die Punkte nur auf einem vom Anwender vorgegebenen Umkreis liegen können. Weatherill [116] kontrolliert die Dichtekontrollfunktion über den Abstand der Punkte untereinander. Die Dichtefunktion wird dort durch die Punkteverteilung am Rand definiert und in das Innere des zu vernetzenden Gebietes fortgesetzt.

Desweiteren ist es mittels des Delaunay-Verfahrens nicht einfach, Elemente mit lokal vorgegebener Maschenweite zu erzeugen. Dreiecke, die z.B. bei Strömungsberechnungen eine Kante mit einer vorher definierten Länge haben sollen, sind nur schwer zu erzeugen. Gittergeneratoren, die auf dem Delaunay-Verfahren basieren, müssen im allgemeinen für das jeweils betrachtete Problem angepaßt werden und verlieren damit einen Teil ihrer Flexibilität. So wird z.B. in der Arbeit von Mavriplis [68] ein Gittergenerator vorgestellt, mit dem auf einer vorgegebenen Punktmenge mit lokalen konformen Abbildungen Elementkanten in die gewünschte Richtung ausgerichtet werden sollen. Weitere Vorschläge zur Steuerung der Maschenweite sind in [67] zu finden.

Zusammenfassend folgt aus der Diskussion des Delaunay-Verfahrens, daß sich diese Methode nur bedingt zur Erzeugung von Rechengittern für die in Kapitel 2 beschriebenen Problemklassen eignet.

2.3 Quadtree-Methode

Eine weitere effiziente Möglichkeit, Gitter zu erzeugen, stellt die nachfolgend beschriebene Quadtree-Methode dar. Zum besseren Verständnis der Quadtree-Methode wird die formale Definition eines Quadtrees vorgestellt. Anschließend folgt eine detaillierte Beschreibung der Gittergenerierung mit dem Quadtree-Verfahren und es wird verdeutlicht, daß diese Methode nur unzureichend für komplexe Gebiete mit beliebigen inneren Rändern geeignet ist. Auf Anwendungen des Quadtree-Verfahrens wird durch entsprechende Literatur hingewiesen.

Ein Quadtree wird folgendermaßen definiert:

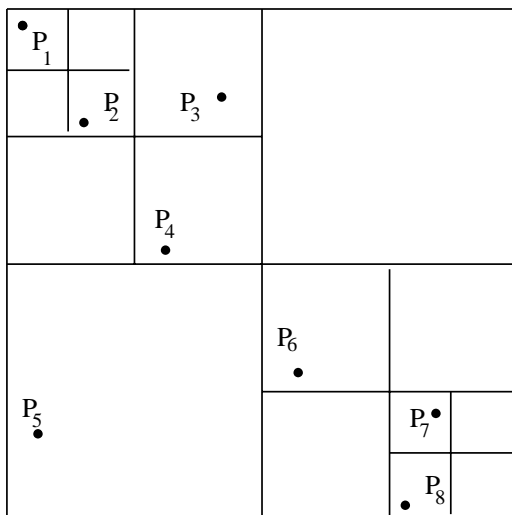
Definition 2.3.1 (Quadtree [88], [89])

Ein *Quadtree* ist eine rekursive Einteilung eines ebenen Gebietes in achsenparallele Quadrate. Das Quadrat, das das ganze Gebiet beinhaltet, wird als *Wurzel* bezeichnet. Ein Quadrat kann in 4 Quadrate durch Teilung mittels Verbindungsgeraden in horizontaler und vertikaler Richtung aufgespalten werden. Die Verbindungsgeraden verlaufen durch den Mittelpunkt des Quadrates. Die 4 neu entstandenen Quadrate werden als *Söhne*, das Quadrat aus dem sie entstanden sind, als *Vater* bezeichnet. Ein Quadrat eines Quadrates ist ein *Blatt*, wenn es keine Söhne hat.

Bemerkung 2.3.1 (Quadtree)

Ein Quadtree einer Punktmenge ist in Abbildung 2.5 dargestellt. Die Zusammenfassung aller Quadrate ergibt einen Baum mit kleiner werdenden Quadraten bei zunehmender Tiefe. Durch Angabe einer Balancierungsbedingung entsteht ein *balancierter Quadtree*. Die bekannten Arten von balancierten Bäumen lassen sich auf zwei Grundtypen zurückführen. Werden die Höhen der Unterbäume balanciert, so entsteht ein *höhenbalancierter Quadtree*. Wird die Anzahl der Knoten balanciert, ergibt sich ein *gewichtsbalancierter Quadtree*. Mittels Balancierungsbedingungen läßt sich die Qualität der Gittererzeugung verbessern. Ein Beispiel eines balancierten Quadrates wird in der Arbeit von Bern et. al. [13] gegeben. Detailliertere Erklärungen mit Komplexitätsabschätzungen sind z.B. in [69] zu finden.

Eine Form der Gittererzeugung durch einen Quadtree ist in der Arbeit von Shephard et.al [120] zu finden. Nach der in dieser Arbeit vorgestellte Methode wird ein beliebiges Gebiet in Quadrate verschiedener Größe eingeteilt. Die Auflösung des Randes erfolgt über Randdichtekontrollfunktionen. Durch die Vorgabe der Tiefe des Quadrates wird die Elementgröße im Innern des Rechengebietes gesteuert. Die Netzgenerierung erfolgt in den folgenden Schritten. Als erstes werden die Quadrate, die einen Teil des Randes des Rechengebietes enthalten, bis zur vorgegebenen Feinheit unterteilt. Danach werden im Innern des Rechengebietes die Quadrate weiter unterteilt. Bei dieser Unterteilung dürfen sich die Kanten aneinandergrenzender Quadrate um höchstens die doppelte Kantenlänge unterscheiden. Lokale Verfeinerungen werden durch vorgegebene Verfeinerungspunkte gesteuert. Die Quadrate, die einen Teil des Randes einschließen, werden so abgeschnitten, daß kein Teilrand über das Rechengebiet reicht. Die Knoten des Rechengitters im Innern entstehen aus den Eckpunkten der Quadrate und die Knoten auf dem Rand aus dem Schnitt der Quadratseiten mit dem Gebietsrand. Durch Einfügen der Diagonalen in die Quadrate des Quadrates entstehen Dreieckselemente. Die einzelnen



Quadtree einer Punktmenge

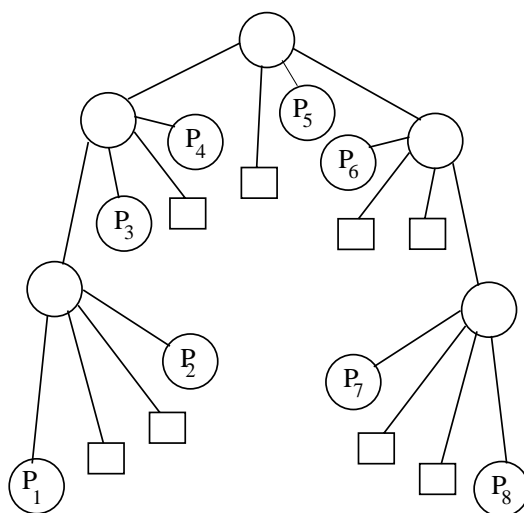


Diagramm eines Quadtrees

Abbildung 2.5: Quadtree einer Punktmenge

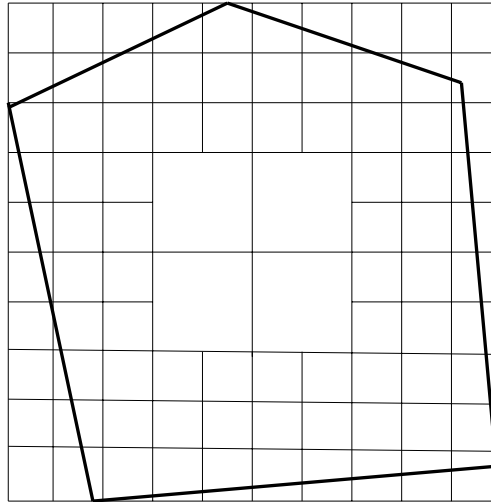


Abbildung 2.6: Quadtree eines polygonal berandeten Gebietes

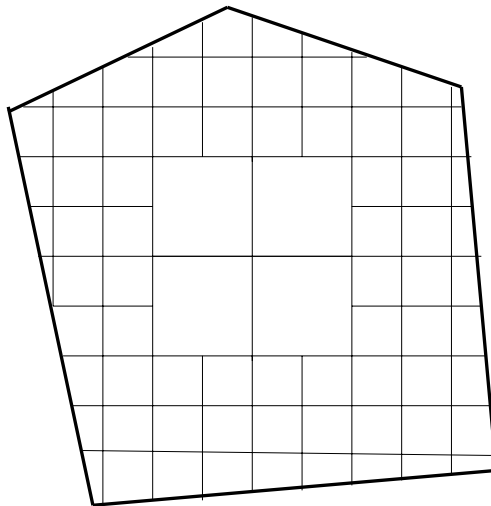


Abbildung 2.7: Anpassen des Quadtrees an die Geometrie

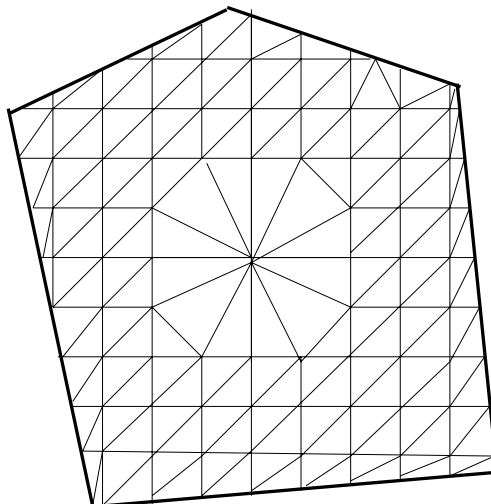


Abbildung 2.8: Rechengitter nach der Quadtree-Methode

Schritte der Gittergenerierung werden in den Abbildung 2.6, 2.7 und 2.8 verdeutlicht. In Abbildung 2.6 wird der Quadtree eines polygonal berandeten Gebietes dargestellt. Abbildung 2.7 zeigt den an die Geometrie angepaßten Quadtree und Abbildung 2.8 das endgültige Rechengitter.

Eine verbesserte Version des oben beschriebenen Verfahrens von Shephard et.al [120] ist in [7] dargestellt und zeigt, wie die Auflösung des Randes verbessert werden kann. Eine theoretische Arbeit von Bern et. al. [13], gibt Regeln an, wie spitze Winkel bei den Dreiecken des Gitters vermieden werden. Außerdem sind dort auch Abschätzungen über die Anzahl der entstehenden Dreiecke zu finden.

Quadtree-Verfahren sind gut geeignet, um Rechengitter zu erzeugen, die kleine Elemente zur Auflösung des Randes komplexer Geometrien brauchen, aber in einem bestimmten Abstand vom Rand möglichst grob werden sollen. Ein weiterer Vorteil liegt darin, daß der Algorithmus in ganzzahliger Arithmetik programmiert werden kann und damit Gleitkommaoperationen nur im Innern kleiner Quadrate, in denen die Geometrie des Gebietes abgelegt ist, notwendig sind. Ein Nachteil liegt in der Repräsentation der Randelemente und der Auflösung innerer Ränder. Die Randelemente zeigen bei komplexen Gebieten schlechte Winkelverhältnisse. Aufgrund der Konstruktionsvorschrift eines Quadtreees ist es schwierig, innere Ränder hinreichend genau aufzulösen. Außerdem gibt es ausgezeichnete Richtungen, d.h. es werden bevorzugt Elemente parallel zur x bzw. y -Richtung erzeugt [102]. Die Qualität der Randelemente kann mit Glättungsmethoden verbessert werden, doch ist das Problem der ausgezeichneten Richtungen noch nicht befriedigend gelöst [13]. Ferner ist es schwierig, adaptive Netze zu erzeugen, da die Konstruktion eines Quadtreees nur eine begrenzte Unterteilung der Quadrate zuläßt. Vorschläge zur Behebung dieses Problems sind in [14] zu finden.

Quadtree-Methoden eignen sich nur bedingt zur Gittererzeugung der in Kapitel 2 aufgeführten Problemklassen.

2.4 Advancing Front Methode

In diesem Abschnitt wird das Prinzip der Advancing Front Methode erläutert und anhand der Arbeiten von Tracy [108] und Peraire et. al. [80] zwei Varianten dieser Gittergenerierungsmethode vorgestellt. Der in dieser Arbeit entwickelte Gittergenerator basiert teilweise auf den in [80] verwendeten Techniken und verwendet auch Methoden von Paolini und Verdi [79]. Die Arbeit von Paolini und Verdi [79]

stellt eine Verbesserung der Methode von Peraire et. al. dar, verwendet jedoch auch ein Hintergrundnetz zur adaptiven Gittererzeugung. Durch die Diskussion der Verfahren von Tracy und Peraire et. al. wird deutlich, daß der hier entwickelte Gittergenerator durch die Einführung einer hierarchischen Datenstruktur komplexe Gebiete mit beliebigen inneren Rändern adaptiv ohne Hintergrundnetz mit Dreieckselementen optimaler Qualität vernetzen kann.

Die Advancing Front Methode ist eine Gittergenerierungstechnik, mit der gleichzeitig Knoten und Elemente erzeugt werden können. Als *Advancing Front* wird eine Menge von Kanten, die sich nur an ihren Endpunkten berühren, bezeichnet. Der Rand einer Geometrie wird durch Knoten in Verbindungsstrecken unterteilt. Die Länge der Verbindungsstrecken wird durch die gewünschte Maschenweite der Elemente definiert und ergeben ein geschlossenes Polygon, das als *aktive Front* bezeichnet wird. Die Elemente werden entlang der aktiven Front erzeugt und weisen in das Innere des zu vernetzenden Gebietes. Nach jeder Generierung eines Elementes wird die aktive Front aktualisiert und enthält nur noch die Verbindungsstrecken, die zur weiteren Gittererzeugung verwendet werden.

Die Dichte der Gitterknoten und somit die Maschenweite werden durch entsprechende Parameter vom Benutzer spezifiziert. Durch die Besonderheit, zuerst den Rand einer Geometrie aufzulösen und danach das Gitter in das Innere fortzusetzen, ist diese Methode sehr gut zur Vernetzung komplexer Gebiet mit inneren Rändern geeignet.

Eine der ersten Arbeiten zur Advancing Front Methode ist in dem Artikel von Tracy [108] (Abbildung 2.9) zu finden. Dort werden zuerst Knoten auf dem Rand des Gebietes erzeugt. Diese Knoten bilden die erste aktive Front. Ein Dreieck wird aus den Knoten gebildet, deren Scheitelwinkel kleiner als 90° sind (Fall a. in Abbildung 2.9). Bilden die Knoten der aktiven Front einen Scheitelwinkel zwischen 90° und 180° , wird ein Dreieck durch die Erzeugung eines neuen Knoten gebildet (Fall b. in Abbildung 2.9). Dieses Verfahren wird solange fortgesetzt, bis nur noch drei Knoten in der aktiven Front sind und daraus das letzte Dreieck erzeugt. Anschließend wird die Qualität der Elemente durch Nachglättungsschritte verbessert. Das Verfahren nach Tracy ist jedoch nur auf konvexe Gebiete anwendbar. Nichtkonvexe Gebiete werden manuell vor der Vernetzung in konvexe Gebiete durch geeignete Gebietsaufteilung eingeteilt. Während der Gittererzeugung nach Tracy [108] ist keine Kontrolle der Güte der Dreiecke möglich.

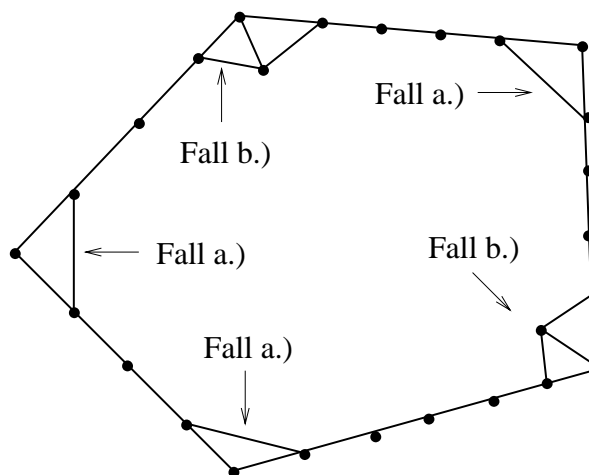


Abbildung 2.9: Advancing Front Methode nach Tracy

Im Artikel von Peraire et. al. [80] wird ein Gittergenerator beschrieben, der Dreieckselemente an beliebigen Kanten der aktiven Front erzeugen kann. Ein Dreieck ist durch eine Kante der aktiven Front sowie durch einen Punkt im Innern des Rechengebietes gegeben. Der Punkt im Innern des Rechengebietes wird entweder neu erzeugt oder es wird ein bereits vorhandener Knoten der aktiven Front zur Konstruktion eines Dreiecks verwendet. Diese wichtige Erweiterung der Advancing Front Technik ermöglicht die Konstruktion fast optimaler Dreiecke. Die Dichte der Dreiecke wird durch ein Hintergrundnetz gesteuert. Das Hintergrundnetz überdeckt das gesamte Rechengebiet und wird zur Interpolation der lokalen Maschenweite benutzt. In Abbildung 2.10 wird eine zu vernetzende Geometrie dargestellt. Abbildung 2.11 zeigt die Approximation der Geometrie durch die erste initialisierende Advancing Front. Der Benutzer kann durch Vorgabe einer lokalen Maschenweite den Rand der zu vernetzenden Geometrie beliebig fein auflösen. Eine fortschreitende aktive Front eines Gitters, das bereits Dreiecke enthält, ist in Abbildung 2.12 dargestellt.

In der Arbeit von Paolini und Verdi [79], die sich eng an Peraire et.al. [80] orientiert, wird eine Advancing Front Methode vorgestellt, in der die lokalen Maschenweiten und die Gestalt der Dreieckselemente des Gitters ebenfalls durch ein Hintergrundnetz gesteuert wird. Der Unterschied zu Peraire et.al. [80] besteht darin, daß auch komplizierte Gebiete mit inneren Rändern vernetzt werden können. In den Arbeiten von Peraire et. al. [80] und Paolini und Verdi [79] ist eine Kontrolle der Gitterqualität zwar möglich, doch findet die eigentliche Verbesserung der Gestalt der Dreieckselemente in Nachglättungsschritten statt.

Es gibt bereits Ansätze, siehe [27], in denen eine Kontrolle der Elementqualität während der Gittererzeugung untersucht werden. Das Prinzip der Qualitätsverbesserung in [27] besteht in einer Netzglättung entlang der aktiven Front. Die Methode in [27] ist allerdings nur auf konvexe Gebiete anwendbar. Eine Erweiterung des Verfahrens nach [27] auf Viereckselemente und nichtkonvexe Gebiete stellt die Arbeit von Xie et.al. [119] dar, stößt jedoch auf Schwierigkeiten bei Gebieten mit inneren Rändern.

Der Gittergenerator, der für das Programmpaket *UG* [11] entwickelt worden ist, beruht teilweise auf den Arbeiten von Peraire et. al. [80] und Paolini und Verdi [79], verwendet aber kein Hintergrundnetz und kann bereits während der Gittererzeugung die Qualität der Elemente auch auf komplexen Gebieten mit inneren Rändern verbessern. Durch die Vorgabe einer vom Benutzer frei wählbaren Dichtekontrollfunktion zur Steuerung der Maschenweite der Elemente können adaptive Gitter einfach erzeugt werden. Ein weiterer wichtiger Unterschied zu den Verfahren in [27], [80], [79] und [119] sind die im noch folgenden Kapitel 3.3 beschriebenen Datenstrukturen. Die dort beschriebenen linearen Listen sind hierarchisch angeordnet und erlauben eine optimale Kontrolle des Verlaufs der aktiven Front. Das in Kapitel 3.3 eingeführte lineare Listenkonzept ermöglicht die Vernetzung komplexer Gebiete mit vielen inneren Rändern mit Dreiecken fast optimaler Güte. Nachglättungsschritte können auf ein Minimum beschränkt werden.

Als Beispiel einer Anwendung der Advancing Front Methode im Bereich Strömungsmechanik wird auf die Arbeit von Barth und Jespersen [10] hingewiesen. Dort wurde diese Technik in modifizierter Form zur Generierung eines Netzes zur Berechnung der Umströmung eines Tragflügelprofils angewandt.

Die Weiterentwicklung der Advancing Front Technik für die Erzeugung von Viereckselementen ist etwas aufwendiger und wird in [21] und [122] beschrieben. Eine der Advancing Front ähnliche Methode stellt die Arbeit von [15] dar. Dort wird mit einer Methode, die als *paving* bezeichnet wird, ein beliebiges zweidimensionales, ebenes Gebiet mit Viereckselementen vernetzt. Das Prinzip besteht in einem iterativen Prozeß, beginnend vom Rand des Gebietes entlang der aktiven Front schichtweise Elemente erzeugt werden.

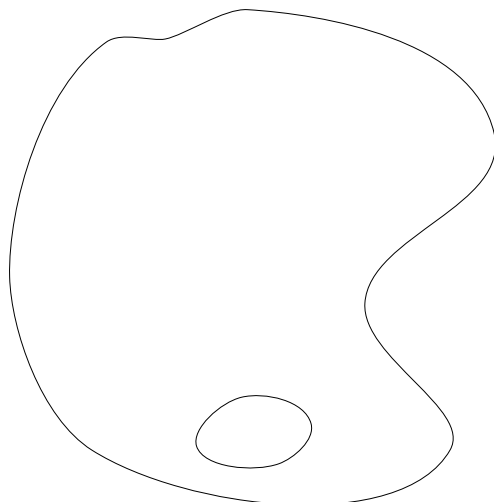


Abbildung 2.10: Geometrie einer Advancing Front

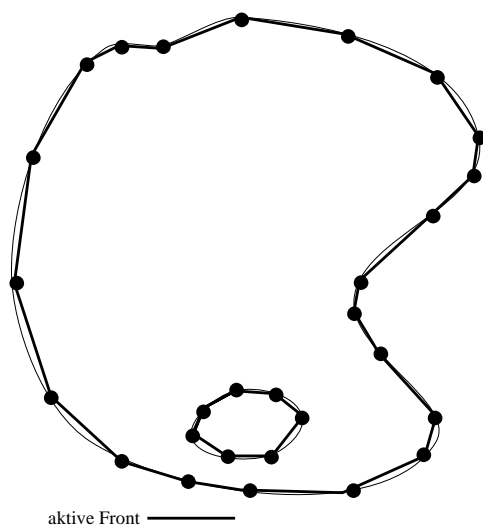


Abbildung 2.11: Geometrie mit erster aktiver Front

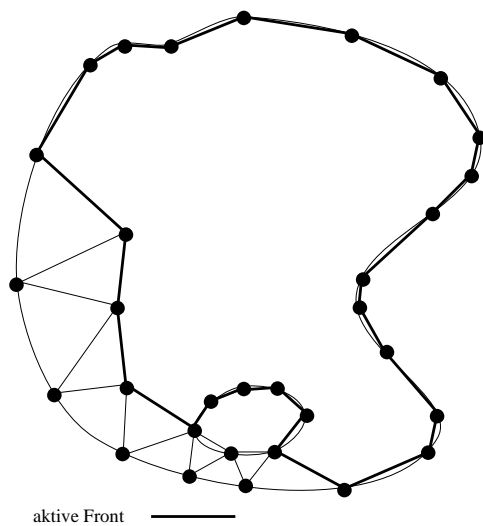


Abbildung 2.12: Geometrie mit aktualisierter aktiver Front nach dem Einfügen einiger Dreiecke

Kapitel 3

Gebietsdefinitionen und Datenstrukturen

In diesem Kapitel wird die Gebietsbeschreibung des Programmpakets *UG* [11] vorgestellt. Zweidimensionale Geometrien können im Programm *UG* in Parameterform oder durch Polygone beschrieben werden. Im ersten Abschnitt wird die Darstellung von Gebieten in parametrisierter Form beschrieben und am Beispiel eines eilipsoidförmigen Gebietes mit inneren Rändern veranschaulicht. Es folgt im anschließenden Abschnitt die Darstellung eines polygonal berandeten Gebietes. Der dritte Abschnitt behandelt die zum Verständnis des Listenkonzepts des Gittergenerators notwendigen Definitionen der Operationen mit linearen Listen und ist an [78] orientiert.

3.1 Gebiete in Parameterdarstellung

Ein Beispiel für die Parameterdarstellung eines Gebietes $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ mit inneren Rändern ist in Abbildung 3.1 dargestellt. Der Rand $\partial\Omega$ des Gebietes $\Omega \subset \mathbb{R}^2$ ist die Vereinigung der Randsegmente $\partial\Omega_i$, $i = 1, 2, \dots, 6$. Durch die Einteilung in 6 Randsegmente wird das Rechteckgebiet in die 3 Teilgebiete Ω_i , $i = 1, 2, 3$ zerlegt. Teilgebiet Ω_1 enthält außerdem noch einen inneren Rand. Die Randsegmente werden durch nichtlineare Abbildungen von einem geeignet gewählten Parameterraum in den \mathbb{R}^2 beschrieben. Die Abbildungen müssen so gewählt werden, daß die Randsegmente zusammenhängend sind. Als Beispiel werden die Parameterfunktionen zur Beschreibung des Gebietes in Abbildung 3.1 angegeben. Dabei bezeichnen a die große Halbachse, b die kleine Halbachse der großen Ellipse, a' , b' die große und kleine Halbachse der kleinen Ellipse. Der Koordinatenursprung

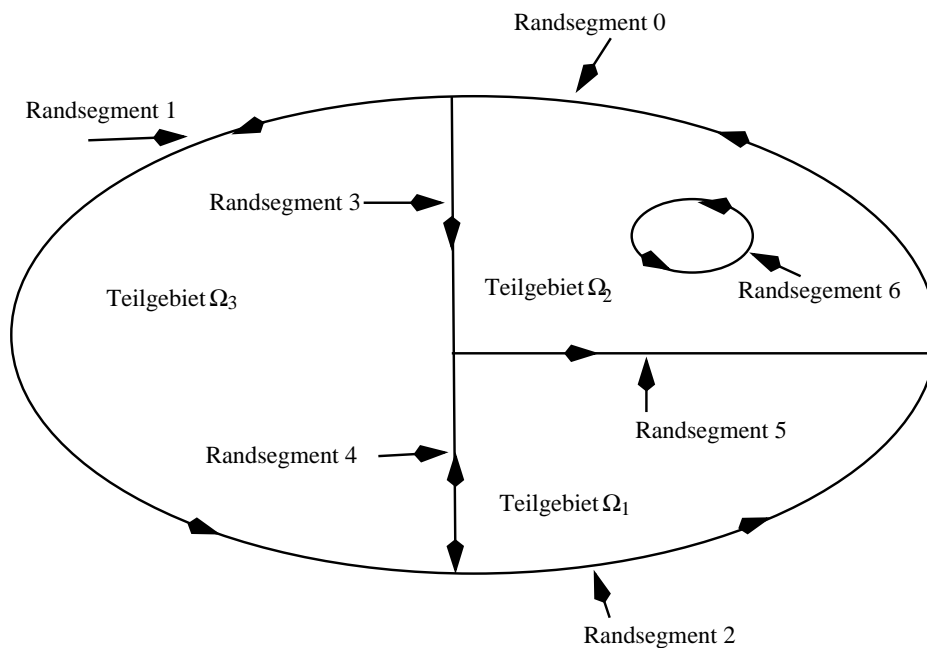


Abbildung 3.1: Parameterdarstellung eines Gebietes mit inneren Rändern und mehreren Teilgebieten

liegt in der Mitte des Gebietes in Abbildung 3.1.

Die Parametrisierung eines Ellipsenbogens erfolgt über:

$$\begin{aligned} x &= a \cos \lambda \\ y &= b \sin \lambda, \quad \lambda \in [\varepsilon_1, \varepsilon_2], \quad \varepsilon_1, \varepsilon_2 \in \mathbb{R} \end{aligned} \quad (3.1)$$

Die Ränder eines Gebietes werden als Randsegmente bezeichnet. Die einzelnen Randsegmente der Ellipse unterscheiden sich nur durch den Definitionsbereich von λ . Für die Randsegmente 0, 1 und 2 wird Gleichung 3.2 mit folgenden Parametern verwendet:

$$\text{Randsegment 0: } \lambda \in [0, \frac{\pi}{2}]$$

$$\text{Randsegment 1: } \lambda \in [\frac{\pi}{2}, \frac{3}{2}\pi]$$

$$\text{Randsegment 2: } \lambda \in [\frac{3}{2}\pi, 2\pi]$$

Die Beschreibung der Randsegmente 3, 4 und 5 erfolgt durch:

$$\text{Randsegment 3: } \lambda \in [0, 1]$$

$$x = 0$$

$$y = b\lambda$$

Randsegment 4: $\lambda \in [-1, 0]$

$$x = 0$$

$$y = b\lambda$$

Randsegment 5: $\lambda \in [0, 1]$

$$x = a\lambda$$

$$y = 0$$

Randsegment 6: $\lambda \in [0, 2\pi]$

$$x = x' + a' \cos \lambda$$

$$y = y' + b' \sin \lambda$$

Zur Unterscheidung von innerem und äußerem Gebiet wird jedem Randsegment eines Gebietes ein Durchlaufsinne zugeordnet. Dadurch kann entschieden werden, ob ein Punkt links oder rechts eines Randsegments liegt und je nach Durchlaufsinne entschieden werden, ob ein Punkt im Gebietsinnern liegt.

Die Konstruktion des Randes eines Rechengebietes in einer Form, die das Programmpaket *UG* verarbeiten kann, erfolgt in mehreren Schritten. Dazu wird folgende Definition benötigt:

Definition 3.1.1 (Parameter *res*, polygonales Raster)

Der Parameter *res* gibt an, in wieviele Polygonstücke ein Randsegment eingeteilt wird. Die daraus entstehende Menge von Polygonen wird als polygonales Raster eines Randsegments bezeichnet.

Bemerkung 3.1.1

Ein Randsegment des Randes einer zu vernetzenden Geometrie wird in einer vom Benutzer definierten Feinheit mittels des Parameters *res* in ein polygonales Raster eingeteilt. Das polygonale Raster stellt eine diskrete Form der Parameterfunktion des Randsegments dar. Je feiner der Parameter *res* gewählt wird, desto besser werden die Randsegmente der zu vernetzenden Geometrie im Programmpaket *UG* dargestellt. Das polygonale Raster wird in einer geeigneten Datenstruktur abgespeichert und enthält alle Geometrieinformationen zur Bestimmung von Punkten auf dem Rand des Rechengebietes. Der Parameter *res* wird nur zur Berechnung der Länge der Polygonzüge des polygonalen Rasters (siehe Gleichung (3.4)) benötigt und ist unabhängig von der gewählten Maschenweite *h*. So ist z.B. in Abbildung 3.3 der Parameter *res* = 100 und das polygonale Raster enthält 100 Polygonstücke. Mit Vorgabe der Maschenweite *h* = 0.05 besteht die erste aktive Front jedoch nur aus 20 Punkten und 20 Polygonstücken.

Es folgt die Beschreibung der Konstruktion des polygonalen Rasters einer Kurve in Parameterform in \mathbb{R}^2 . Gegeben sei eine Kurve $f(x, y)$ in Parameterform mit $x = x(\lambda), y = y(\lambda)$. Im ersten Schritt wird die Bogenlänge s der Kurve $f(x, y) = (f_1(\lambda), f_2(\lambda))$ im endlichen Intervall $[\lambda_1, \lambda_2], \lambda_1, \lambda_2 \in \mathbb{R}$ berechnet. Die Bogenlänge s_{param} der Funktion f in Parameterfunktion folgt aus der Berechnung der Bogenlänge s_{kart} von f in kartesischen Koordinaten und ist gegeben durch:

$$s_{kart} = \int_{x_1}^{x_2} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \quad (3.2)$$

Mit der Bezeichnungsweise

$$\begin{aligned} \frac{dx(\lambda)}{d\lambda} &= x'_\lambda \\ \frac{dy(\lambda)}{d\lambda} &= y'_\lambda \end{aligned} \quad (3.3)$$

wird die Bogenlänge folgendermaßen berechnet:

$$s_{param} = \int_{\lambda_1}^{\lambda_2} \sqrt{1 + \frac{y'^2_\lambda}{x'^2_\lambda} x'^2_\lambda} d\lambda = \int_{\lambda_1}^{\lambda_2} \sqrt{x'^2_\lambda + y'^2_\lambda} d\lambda$$

Die Länge m der Polygonzüge folgt aus:

$$m = \frac{s_{param}}{res} \quad (3.4)$$

Die Veranschaulichung des Parameters res erfolgt durch zwei Beispiele.

Beispiel 3.1.1

Als einfaches Beispiel eines Gebietes in Parameterdarstellung wird das Einheitsquadrat betrachtet (Abbildung 3.2). Das Einheitsquadrat wird durch 4 Randsegmente parametrisiert. Die Parameterfunktionen sind in Abbildung 3.2 definiert. Da die Randsegmente als Geraden vorliegen, kann der Parameter $res = 1$ gewählt werden, ohne an Genauigkeit der Randauflösung zu verlieren. Die diskrete Darstellung eines Randsegments ist eine Polygonseite und das polygonale Raster besteht aus 4 Polygonseiten. Durch die Vorgabe der Maschenweite $h = \frac{1}{6}$ wird die erste aktive Front der Advancing Front bestimmt (vergl. Abbildung 3.2).

Beispiel 3.1.2

Eine Parameterdarstellung des Einheitskreises ist in Abbildung 3.3 definiert. Zur Beschreibung des Einheitskreises genügt die Angabe einer Parameterfunktion. In

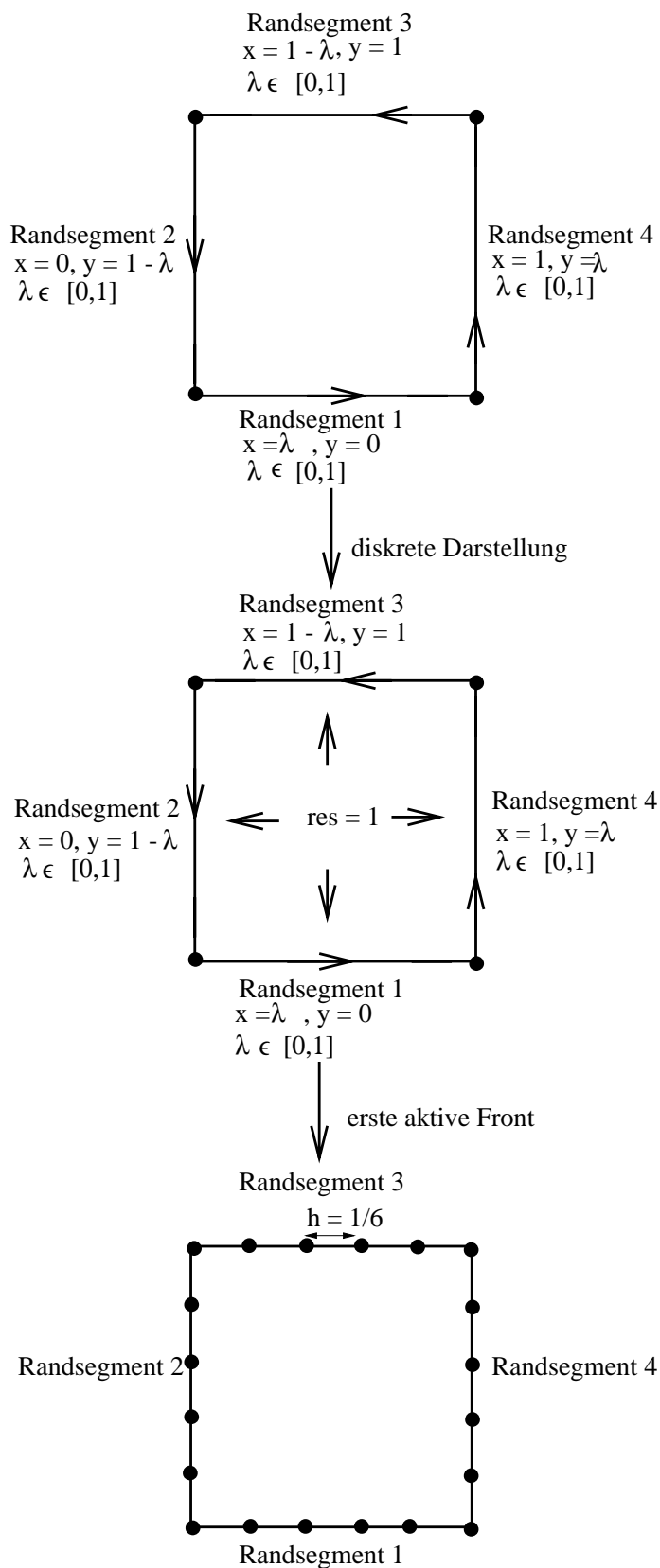


Abbildung 3.2: Beispiel 1: Parametrisierung eines Quadrats

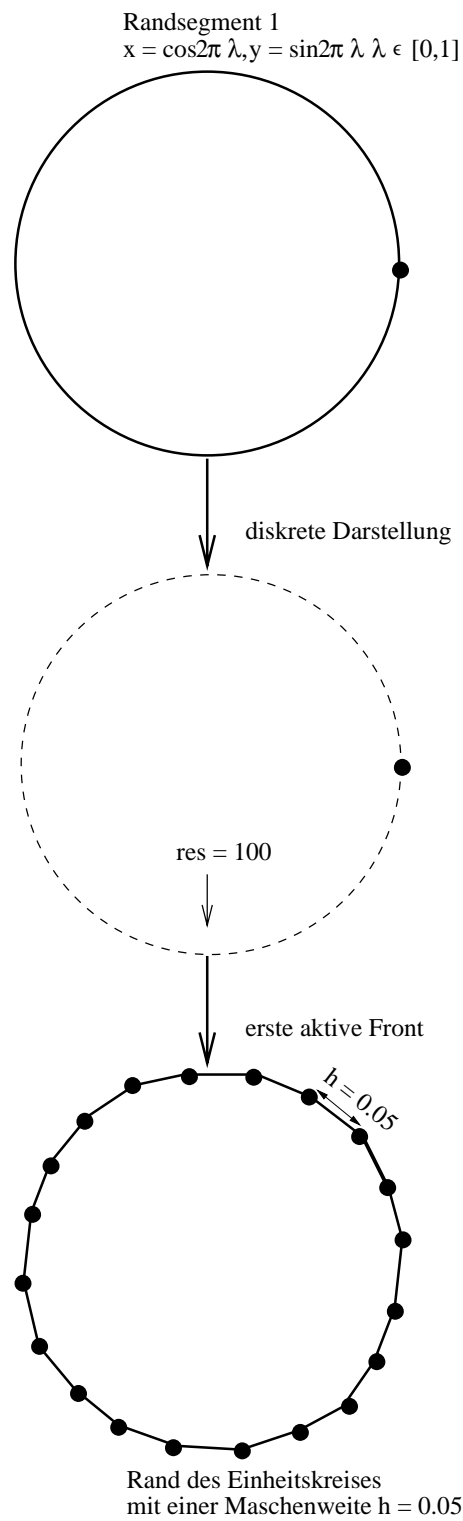


Abbildung 3.3: Beispiel 2: Parametrisierung eines Kreises

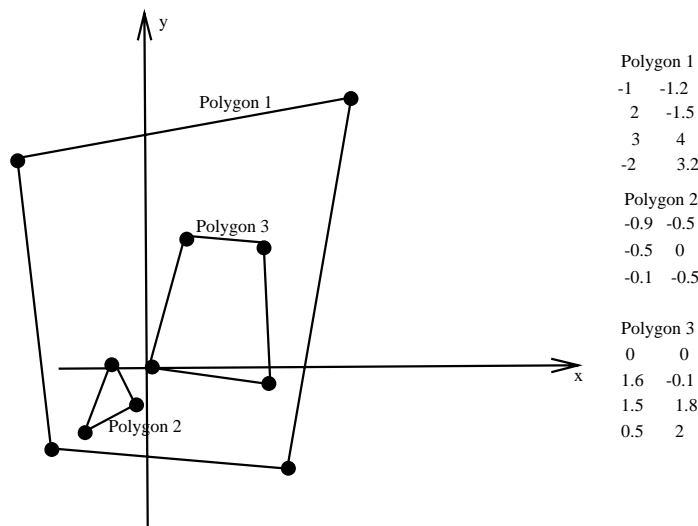


Abbildung 3.4: Polygonebiet mit inneren Rändern

diesem Beispiel wird der Parameter $res = 100$ gewählt und somit der Einheitskreis durch 100 Polygonseiten approximiert (Abbildung 3.3). In diesem Beispiel wird eine Maschenweite von $h = 0.05$ gewählt und damit ergibt sich eine erste aktive Front, die aus 20 Punkten und 20 Kanten besteht (Abbildung 3.3).

Prinzipiell können beliebige Geometrien durch Parameterfunktionen beschrieben werden. In der Praxis ist dies jedoch schwierig, da es nicht einfach ist, für beliebige Gebiete geeignete Parametrisierungen zu finden. In *UG* wurden daher Entwicklungen begonnen, Geometrien, die durch CAD-Programme modelliert werden, über eine Schnittstelle zu CAD-Programmen einzulesen.

3.2 Gebiete in polygonaler Darstellung

Die Eingabe eines Polygons in *UG* geschieht über ein ASCII-File, in dem die Punkte des Polygons in kartesischen Koordinaten abgelegt sind. Ein Beispiel hierzu mit den zugehörigen kartesischen Koordinaten ist in Abbildung 3.4 zu finden. Auch hier ist wiederum ein Umlaufsinn einzuführen, um eindeutig das Rechengebiet zu definieren.

3.3 Datenstrukturen

Ein wichtiger Punkt in der Erzeugung unstrukturierter Gitter ist die Wahl einer geeigneten Datenstruktur. Für den in dieser Arbeit entwickelten Gittergenerator werden die Konzepte linearer einfach verketteter Listen, linear doppelt verketteter Listen und zyklischer Listen verwendet. Zum leichteren Verständnis der speziellen Datenstrukturen des Gittergenerators werden nachfolgend einige grundlegende Begriffe und Erläuterungen zu den verwendeten Listenkonzepten in Anlehnung an [78] eingeführt.

Definition 3.3.1 (ADT)

Ein *abstrakter Datentyp (ADT)* besteht aus einer oder mehreren Mengen von Objekten und darauf definierten Operationen. In einem ADT werden Daten und Operationen mit den Daten als Einheit zusammengefaßt.

Definition 3.3.2 (Datenstruktur)

Eine *Datenstruktur* bezeichnet die Realisierung eines abstrakten Datentyps (ADT) mit Mitteln einer Programmiersprache.

Definition 3.3.3 (Lineare Liste)

Eine *lineare Liste* besteht ist eine endliche Folge von Elementen eines gegebenen Grundtyps, bei der nicht alle Elemente voneinander verschieden sein müssen.

Bemerkung 3.3.1 (Lineare Liste)

Das Konzept der linearen Liste ist sehr verwandt mit der mathematischen Darstellung einer endlichen Folge. Analog zu den Gliedern der Folge $a_n, n \in \mathbb{N}$ wird auch für die Glieder einer linearen Liste die gleiche Sprechweise verwendet. Oft wird anstelle von Gliedern einer linearen Liste auch die Bezeichnung Elemente einer linearen Liste verwendet. Es ist möglich, ein Element in einer linearen Liste an einer vorgegebenen Stelle einzufügen oder zu entfernen und aus zwei Listen von Elementen durch *Hintereinanderhängen (Verketteten)* eine neue Liste zu erzeugen. Ferner wird die leere Folge explizit zugelassen. Eine lineare Liste L , die n Elemente enthält, wird durch $L = \langle a_1, \dots, a_n \rangle$, die leere Liste durch das Symbol $\langle \rangle$ dargestellt.

Definition 3.3.4 (Grundtyp)

Ein Grundtyp einer linearen verketteten Liste besteht aus mindestens 2 Komponenten. Eine Komponente wird als Schlüsselkomponente bezeichnet. Mindestens eine weitere Komponente enthält die zu verarbeitende Information. Ein Grundtyp hat den folgenden formalen Aufbau:

typeGrundtyp = **record**

key : integer ;

inf : {infotype} {weitere Einträge}

end

Es werden nachfolgend die wichtigsten Operationen mit linearen Listen vorgestellt.

Einfügen (e, p, L): Wird ein neues Element e des entsprechenden Grundtyps an der Stelle p der Liste L eingefügt, so rücken alle Elemente ab Position p um eine Stelle nach rechts. Aus der ursprünglichen Liste $L = \langle a_1, \dots, a_n \rangle$ entsteht die neue Liste $L' = \langle a_1, \dots, a_{p-1}, e, a_p, \dots, a_n \rangle$ mit $1 \leq p \leq n$. Mit $L = \langle \rangle$ und $p = 1$ ergibt das Einfügen von e die Liste $L' = \langle e \rangle$. Ist $p = n + 1$, so ergibt sich $L' = \langle a_1, \dots, a_n, e \rangle$.

Entfernen (p, L): Das Entfernen bzw. Löschen eines Elementes e an der Position p einer Liste $L = \langle a_1, \dots, a_{p-1}, a_p, a_{p+1}, \dots, a_n \rangle$ liefert als Ergebnis $L' = \langle a_1, \dots, a_{p-1}, a_{p+1}, \dots, a_n \rangle$, mit $1 \leq p \leq n$.

Suchen (e, L): Mit dieser Operation wird die Position des Elementes e in der Liste L bestimmt, sofern e ein Element von L ist. Kommt e mehrfach in L vor, so wird die am weitesten links liegende Position ermittelt.

Zugriff (a_p, L): Mit dieser Operation wird das Element a_p an der p -ten Position der Liste $L = \langle a_1, \dots, a_n \rangle$ mit $1 \leq p \leq n$ bestimmt.

Leere Liste (L): Es wird geprüft, ob die Liste L die leere Liste ist.

Verketten (L_1, L_2): Gegeben seien zwei Listen $L_1 = \langle a_1, \dots, a_n \rangle$ und $L_2 = \langle b_1, \dots, b_m \rangle$. Das Aneinanderfügen bzw. Verketten beider Listen nach der Vorschrift $L_1 + L_2$ erzeugt die Liste $L = \langle a_1, \dots, a_n, b_1, \dots, b_m \rangle$.

Teilen (L_1, L_2): Eine Liste $L = \langle a_1, \dots, a_m, a_{m+1}, \dots, a_n \rangle$ bestehe aus n Elementen und soll in zwei Listen mit m und $n - m$, $n > m$ Elemente aufgeteilt werden. Es entstehen die Listen $L_1 = \langle a_1, \dots, a_m \rangle$ und $L_2 = \langle a_{m+1}, \dots, a_n \rangle$. Soll ein Element von L_1 ebenfalls in L_2 enthalten sein, so wird dieses verdoppelt und an die

erste Position in L_2 gesetzt. Als Beispiel wird das Listenelement a_m aus L_1 auch in L_2 eingefügt. Als Ergebnis dieser Teilung ergibt sich $L_2 = \langle a_m, a_{m+1}, \dots, b_n \rangle$

Die vielfältigen Implementierungen linearer Listen können in folgende Klassen eingeteilt werden.

Werden die Listenelemente in einem zusammenhängenden Speicherbereich derart abgelegt, daß über eine Adressrechnung auf das i -te Element zugreiffen werden kann, ergibt sich eine *sequentiell gespeicherte lineare Liste*.

Können die Listenelemente in beliebigen Speicherzellen abgelegt und durch Verweise miteinander verknüpft werden, so handelt es sich um eine *verkettete gespeicherte lineare Liste*.

Sind alle Listenelemente durch Verweise auf die Nachbarelemente miteinander verknüpft, d.h. Listenelement a_m besitzt einen Verweis auf Listenelement a_{m+1} und a_{m-1} , liegt eine *doppelt verkettete Liste* vor.

Eine doppelt verkettete Liste wird zu einer *zyklisch doppelt verketteten Liste*, wenn Anfang und Ende der Liste miteinander durch einen Verweis verknüpft werden.

Die für den hier entwickelten Gittergenerator verwendeten linearen Listen werden als verkettete und zyklisch verkettete Listen abgespeichert. Die häufigsten Operationen sind Einfügen und Löschen von Elementen. Diese Vorgänge sind mit verketteten bzw. zyklisch verketteten abgespeicherten Listen besonders effizient auszuführen. Für sequentiell gespeicherte Listen sind diese Operationen sehr aufwendig, weil große Speicherbereiche verschoben werden müssen.

Detaillierte Ausführungen zu den vorgestellten Operationen sowie weitere nützliche Datenstrukturen sind z.B. in [62], [78] oder [98] zu finden.

Kapitel 4

Die Advancing Front Methode des Programmbaukastens *UG*

In diesem Kapitel wird die neue Advancing Front Methode und deren Implementierung im Programmbaukasten *UG* beschrieben. Die hier vorgestellte Advancing Front Methode orientiert sich teilweise an den Arbeiten von Peraire et. al. [80] et. al. und Paolini und Verdi [79]. Im Gegensatz zur Vorgehensweise in der Arbeit von Peraire et.al. [80] wird hier kein Hintergrundnetz zur adaptiven Gittererzeugung benötigt. Die Verbesserung der Advancing Front Technik liegt im Vergleich mit den Methoden in Peraire et. al. [80] und Paolini und Verdi [79] in der Steuerung der Advancing Front, in den Auswahlkriterien zur Erzeugung eines Dreiecks im Rechengebiet und in den Datenstrukturen.

Zuerst wird der prinzipielle Aufbau der Advancing Front Methode von *UG* zusammengefaßt und in Abbildung 4.1 veranschaulicht. Damit soll ein prinzipielles Verständnis der hier entwickelten Methode ohne Detailkenntnisee der Datenstrukturen und Kontrollfunktionen ermöglicht werden. Danach werden im ersten Abschnitt und in den zugehörigen Unterabschnitten die verwendeten Datenstrukturen detailliert beschrieben. In Abschnitt 4.2 wird gezeigt, wie Knoten auf den Rändern eines Gebietes erzeugt werden. Im folgenden Abschnitt 4.3 erfolgt die Beschreibung der Konstruktion von Knoten und Elementen im Rechengebiet. Die 4 Kontrollfunktionen zur Kontrolle der Qualität eines neu zu erzeugenden Dreiecks werden in Abschnitt 4.4 vorgestellt und in Unterabschnitt 4.4.5 die Aktualisierung der Advancing Front diskutiert. In Abschnitt 4.5 wird gezeigt, wie die Gitterqualität nach der Gittergenerierung durch Laplace-Glättung (Unterabschnitt 4.5.1) im Falle konvexer Gebiete und durch eine modifizierte Laplace-Glättung (Unterabschnitt 4.5.2) bei nicht konvexen Gebieten verbessert werden kann. Im letzten Abschnitt 4.6 dieses Kapitels werden Anwendungen des Gittergenerators

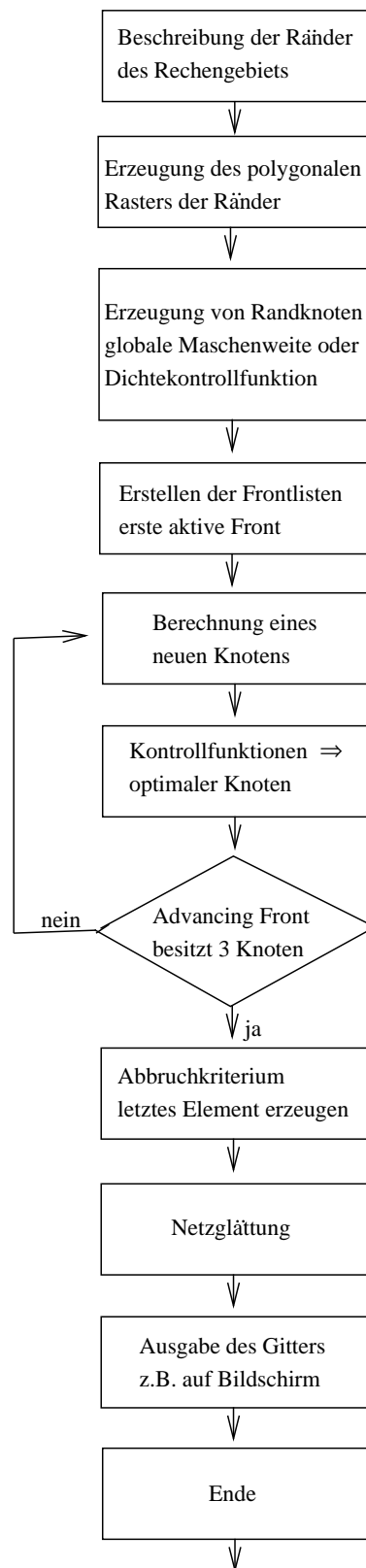


Abbildung 4.1: Advancing Front Algorithmus des Programmabaukastens UG

vorgestellt und anhand der Vernetzung der Nordseeküste (Unterabschnitt 4.6.1) gezeigt, wie durch Verwenden einer Dichtekontrollfunktion adaptive Gitter erzeugt werden können.

Zusammenfassung der Advancing Front Methode:

Die prinzipielle Vorgehensweise zur Realisierung der hier entwickelten Advancing Front Methode wird folgendermaßen zusammengefaßt. Im ersten Schritt erfolgt die Beschreibung der Ränder der Rechengebiete in Parameterform oder als Polygone. Jedem Teilgebiet wird eine Teilgebietsnummer und jedem Randsegment ein Durchlaufsinne zugeordnet. Anschließend werden Punkte auf den Rändern nach einem vorgegebenen Kriterium, z. B. einer Dichtekontrollfunktion, erzeugt. Die Randpunkte werden in Form von hierarchisch verknüpften linearen Listen zur ersten aktiven Front zusammengefaßt. Die Randpunkte werden als Frontkomponenten (FC = front components) in Form von linearen doppelt verketteten Listen abgespeichert. Frontkomponenten beinhalten neben geometrischen Informationen auch Daten zur Weiterverarbeitung im Programm *UG*. Alle Frontkomponenten, die sich auf dem gleichen inneren oder äußeren Rand befinden werden in einer weiteren linearen Liste, die als Frontliste (front list = FL) bezeichnet wird, zusammengefaßt (vergl. Abbildung 4.2). Genau eine Frontliste ist einem Randsegment zugeordnet. Die Frontlisten werden wiederum als Elemente unabhängiger Frontlisten (independent frontlist = IFL) abgespeichert (vergl. Abbildung 4.2). Die Organisation der Datenstrukturen ermöglicht einen leichten Übergang zur Parallelisierung. Basierend auf der ersten aktiven Front erfolgt die Erzeugung eines Punktes im Gebietsinnern. Die Auswahl der Kante der aktiven Front und der damit verknüpften Punkte erfolgt nach einem Kanten- oder Winkelkriterium. Kontrollfunktionen ermöglichen die Bestimmung der optimalen Position des neuen Punktes. Es wird ein Dreieck erzeugt und in die Datenstruktur von *UG* eingebaut. Danach wird die Advancing Front aktualisiert und der Prozess solange fortgesetzt, bis nur noch 3 Punkte in der Advancing Front vorhanden sind. In diesem Fall wird daraus das letzte Dreieck gebildet und die Gittergenerierung beendet. Durch Glättungsverfahren nach Beendigung der Netzerzeugung wird die Gitterqualität verbessert.

Der Advancing Front Algorithmus des Programmbaukastens *UG* wird in Abbildung 4.1 graphisch veranschaulicht. Die Komplexität des Algorithmus konnte von $O(n\sqrt{n})$ durch die Konstruktion eines Quadtrees zur Beschleunigung der Suchvorgänge in den Kontrollfunktionen im Rahmen einer Diplomarbeit [34] auf optimale Komplexität $O(n \log n)$ verbessert werden.

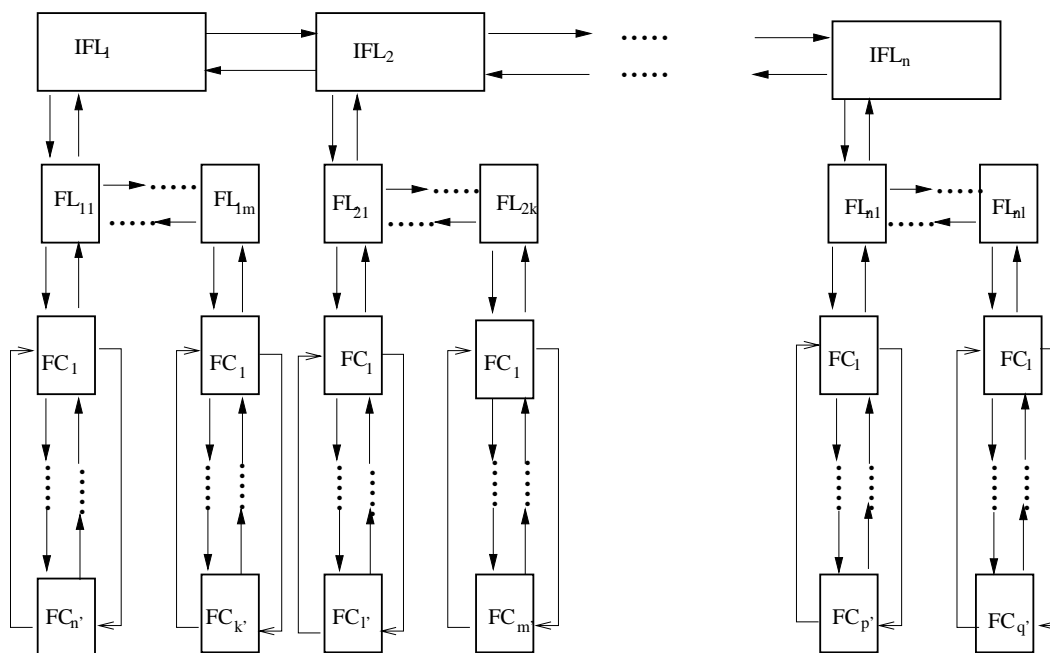


Abbildung 4.2: Datenstruktur des Gittergenerators

4.1 Das Listenkonzept des Gittergenerators

In diesem Abschnitt werden die linearen Listenkonzepte im Überblick erläutert und ein prinzipielles Verständnis der Datenstrukturen gegeben. In den nachfolgenden Unterabschnitten werden die Datenstrukturen ausführlich dargestellt.

Die Datenstruktur zur Steuerung des Advancing Front Algorithmus besteht aus drei hierarchisch miteinander verknüpften Grundtypen linearer Listen. Diese Grundtypen wurden in der Zusammenfassung (Kapitel 4) als unabhängige Frontlisten (independent frontlist = IFL), Frontlisten (front list = FL) und Frontkomponenten (FC = front components) eingeführt. Es gilt die Beziehung:

$$FC \in FL \in IFL$$

Bei den unabhängigen Frontlisten und den Frontlisten handelt sich um lineare doppelt verkettete Listen. Die Frontkomponenten sind als zyklische lineare doppelt verkettete Listen implementiert. Die Bereitstellung des RechenSpeichers der Listen und ihrer Komponenten erfolgt dynamisch und ist deshalb sehr flexibel und nur durch die Rechnerkapazität begrenzt.

Die Datenstrukturen und ihre Verknüpfungen werden in Abbildung 4.2 dargestellt. Die Verbindung zwischen den unabhängigen Frontlisten (IFL) wird über jeweils einen Zeiger zum Vorgänger bzw. Nachfolger hergestellt. Ansonsten beste-

hen keine weiteren Abhängigkeiten (Abbildung 4.2). Jede unabhängige Frontliste besteht aus einer oder mehreren Frontlisten (FL) (Abbildung 4.2). Eine Frontliste repräsentiert einen Rand eines Teilgebietes der zu vernetzenden Geometrie. Die Einträge einer Frontliste werden als Frontkomponenten (FC) (Abbildung 4.2) bezeichnet und beinhalten die zur Gittergenerierung notwendigen Daten, wie z.B. Koordinaten, Zeiger zu Strukturen des Programms *UG*, etc.. Die Frontkomponenten werden zur Berechnung von Knoten im Innern des Rechengebietes und zur Gitterkontrolle während des Generierungsvorgangs herangezogen.

Die Zuordnung der Frontlisten zur Geometrie werden in Abbildung 4.3 verdeutlicht. In Abbildung 4.3 (linke Hälfte) wird ein Gebiet Ω mit Loch, das nur aus einem Teilgebiet besteht, dargestellt. Dem Gebiet Ω wird eine unabhängige Frontliste zugewiesen. Die unabhängige Frontliste besteht wegen des inneren Lochs aus zwei Frontlisten. Jedem Randsegment wird genau eine Frontliste mit den zugehörigen Frontkomponenten zugeordnet. Die Anzahl der Frontkomponenten wird durch die vorgegebene Maschenweite auf dem Rand bestimmt.

In der rechten Hälfte von Abbildung 4.3 besteht das Gebiet Ω aus den Teilgebieten Ω_1 und Ω_2 . Dem Teilgebiet Ω_1 wird die unabhängige Frontliste IFL_1 zugewiesen. Die unabhängige Frontliste IFL_1 enthält die Frontlisten FL_1 und FL_2 . Die Frontlisten FL_1 und FL_2 sind mit den Rändern 1 und 3 verknüpft. Analog ergibt sich für Teilgebiet Ω_2 die unabhängige Frontliste IFL_2 mit den Frontlisten FL_2 und FL_4 als Einträge. Die Frontlisten enthalten ebenfalls Frontkomponenten. Weiterhin existiert eine doppelte Verzeigerung der unabhängigen Frontliste IFL_1 und IFL_2 (vergl. Abbildung 4.3).

Während der Gittergenerierung kann die Advancing Front so verlaufen, daß Teile der aktiven Front abgespalten werden. Daraus ergeben sich neue unabhängige Frontlisten, die wiederum Frontlisten und Frontkomponenten enthalten. Die unabhängigen Frontlisten werden nacheinander und einzeln abgearbeitet.

4.1.1 Das Konzept der unabhängigen Frontliste

Eine unabhängige Frontliste (IFL) enthält als Elemente weitere lineare doppelt verkettete Listen vom Typ Frontlist. Bei Beginn der Vernetzung können bereits mehrere unabhängige Frontlisten vorhanden sein, die dann unabhängig voneinander zur Erzeugung von Elementen des Rechengitters abgearbeitet werden. Durch diesen hierarchischen Aufbau der Listen, insbesondere die Einführung der unabhängigen Frontlisten, besteht die Möglichkeit, die unabhängigen Frontlisten auf

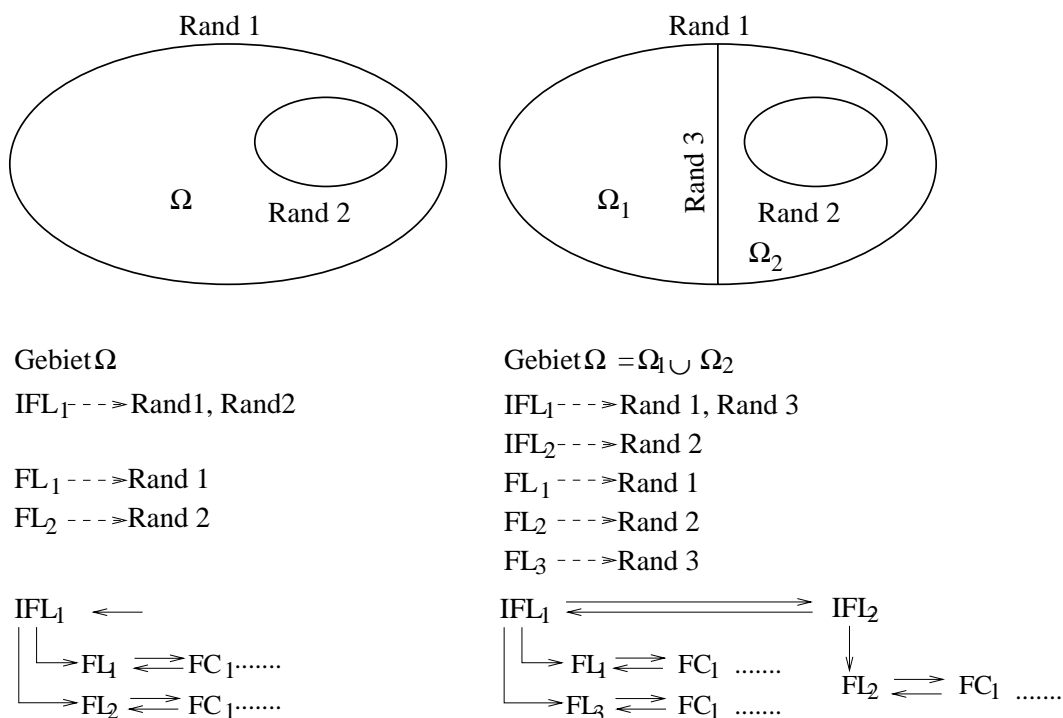


Abbildung 4.3: Zusammenhang zwischen Geometrie und Frontlisten

jeweils verschiedenen Prozessoren abzuarbeiten und somit zu parallelisieren. Eine unabhängige Frontliste (IFL) hat den folgenden Aufbau:

```

struct indepfrontlist {
    struct indepfrontlist *succifl  Pointer auf den Nachfolger
                                   in der doppelt verketteten
                                   Liste
    struct indepfrontlist *predifl  Pointer auf den Vorgänger
                                   in der doppelt verketteten
                                   Liste
    struct frontlist *startfl       Pointer auf den Anfang
                                   einer Frontliste
    struct frontlist *lastfl        Pointer auf das Ende
                                   einer Frontliste
    int nFrontlist                  Anzahl der Frontlisten
                                   in einer IFL
}

```

Die Erzeugung einer unabhängigen Frontliste geschieht mit der Funktion `CreateIndepFrontList`, die als Input nur einen Zeiger auf den Anfang der Gitterverwaltung des Programmpakets `UG` benötigt und als Ausgabe einen Zeiger auf eine unabhängige Frontliste zurückgibt. Die neu erzeugte unabhängige Frontliste wird an die erste Stelle der unabhängigen Frontlisten eingefügt. Mit der Funktion

`DisposeIndepFrontList`, die als Eingabe einen Zeiger auf die zu löschende unabhängige Frontliste erhält, wird die unabhängige Frontliste gelöscht, die Zeiger richtig gesetzt und der Speicher frei gegeben.

4.1.2 Das Konzept der Frontlisten

Der Datentyp `Frontlist (FL)` besteht aus linearen, doppelt verketteten, Listen, die aus den Frontkomponenten (FC) bestehen. Die Frontlisten stellen die eigentliche `Advancing Front` dar. Da jedoch bei mehreren Teilgebieten auch mehrere Frontlisten entstehen können, werden diese als Teilmenge der im vorigen Abschnitt eingeführten unabhängigen Frontlisten (IFL) verwaltet. Dadurch wird die Steuerung der `Advancing Front` einfach und ermöglicht eine Kontrolle der Qualitätseigenschaften der entstehenden Dreiecke des Rechengitters möglich. Eine Frontliste besteht aus:

```

struct frontlist {
    frontlist *succfl           Zeiger auf den Vorgänger
                               in der doppelt verketteten
                               Liste
    frontlist *prefl           Zeiger auf den Nachfolger
                               in der doppelt verketteten
                               Liste
    struct indepfrontlist *myIFL Zeiger zu der zugehörigen
                               IFL
    INT orientation            Orientierung der FL
                               mathematisch positiv oder
                               negativ
    INT SubdomainID           Nummer des zugehörigen Teil-
                               gebietes
    frontcomp *startfc        Zeiger auf den Anfang
                               der Frontkomponentenliste
    frontcomp *lastfc         Zeiger auf das Ende
                               der Frontkomponentenliste
    INT nFrontComp            Anzahl der Frontkomponenten
                               der Frontliste
}

```

Eine neue Frontliste wird mit der Funktion `CreateFrontList` erzeugt. Die Funktion `CreateFrontList` erhält als Eingabe einen Zeiger des Gitters, auf dem ein Knoten bzw. ein Element erzeugt werden soll. Die neue Frontliste wird an die erste Stelle der Liste der Frontlisten eingefügt und die Frontlisten werden aktualisiert. Mit der Funktion `DisposeFrontList` kann eine Frontlist mit ihren Elementen gelöscht und der Speicher freigegeben werden.

4.1.3 Das Konzept der Frontkomponenten

Die Frontkomponenten (FC) sind eine Teilmenge der Frontlisten und enthalten die für die Erzeugung der Elemente des Rechengitters wichtigen Informationen, wie z.B. kartesische Koordinaten von Punkten. Sie stellen die unterste Ebene der Listenhierarchie dar und werden verwendet, um Knoten, Kanten und Elemente zu erzeugen. Der Aufbau einer Frontkomponente sieht folgendermaßen aus:

```

struct frontcomp {
  struct frontcomp *succfc  Zeiger auf den Vorgänger
                           ein der doppelt verketteten
                           Liste
  struct frontcomp *predfc  Zeiger auf den Nachfolger
                           in der doppelt verketteten
                           Liste
  struct frontlist *myFl    Zeiger auf die zugehörige Front-
                           liste
}

```

Die Frontkomponenten werden mit der Funktion `CreateFrontComp` erzeugt. Dazu benötigt diese Funktion einen Zeiger auf die zugehörige Frontliste. Danach wird die neue Frontkomponente an den Anfang der Frontkomponentenliste eingefügt. Beim Entfernen einer Frontkomponente mittels der Funktion `DisposeFrontComp` werden Zeiger umgehängt und der Speicherbereich der zu löschenden Frontkomponente freigegeben.

4.2 Erzeugung von Knoten auf dem Rand

Die Erzeugung von Knoten auf den Rändern des zu vernetzenden Gebiets wird dargestellt und gezeigt, wie daraus die erste aktive Front erzeugt wird.

Die Kurven zur Beschreibung der Ränder des Gebietes Ω und der inneren Teilgebiete $\partial\Omega_i, i = 1, 2, \dots$ sind als Folge benachbarter Strecken bzw. Segmente in Parameterform oder als Polygon gegeben. Jedes Segment ist eindeutig durch die Angabe der zwei Endpunkte, die als Vertex oder Knoten bezeichnet werden, die Angabe eines Index zur Identifizierung und durch Festlegung des Durchlaufsinns definiert. Die kartesischen Koordinaten des neu zu erzeugenden Punktes werden aufgrund der vorgegebenen Maschenweite $h(x, y)$ berechnet. Ist die vorgegebene Maschenweite größer als das kleinste Segment einer Kurve des Gebietes Ω , wird die Geometrie nicht hinreichend genau aufgelöst. Der Benutzer kann nach einer

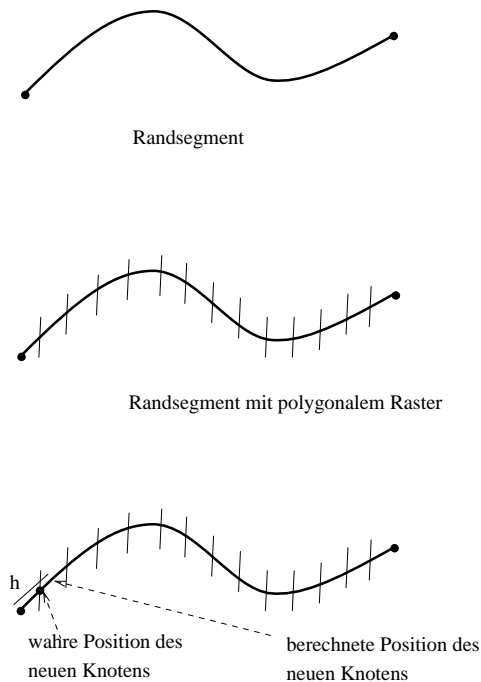


Abbildung 4.4: Erzeugung von Randknoten

Abfrage entscheiden, ob die Maschenweite automatisch so gewählt wird, daß auch dieser Teil der Geometrie aufgelöst wird.

Die Vorgabe einer konstanten Maschenweite $h(x,y)$ ermöglicht eine gleichmäßige Verteilung der Knoten auf den Randsegmenten. Durch Verwenden einer Randdichtefunktion sind beliebige Anordnungen der Punkte auf den Randsegmenten möglich. Die Knoten auf den Randsegmenten werden entsprechend dem zugeordneten polygonalen Raster positioniert. Ist das Randsegment gekrümmt, wird die Maschenweite h als Bogenlänge auf dem gekrümmten Rand bestimmt. Der Punkt des polygonalen Rasters des Randsegments, der der gewünschten Position des neuen Knotens am nächsten liegt, ergibt den wahren geometrischen Ort des neuen Knotens (siehe Abbildung 4.4). Mit der Bestimmung der Maschenweite als Bogenlänge auf gekrümmten Randsegmenten wird die Lage der neuen Knoten an die vorgegebene Geometrie angepaßt. Eine Veranschaulichung der Erzeugung eines neuen Knoten wird in Abbildung 4.4 gezeigt. Nach der Generierung aller Randknoten werden die Knoten und Kanten der Segmente zu der ersten aktiven Advancing Front zusammengefaßt und es kann mit der Generierung von Elementen im Inneren des Rechengebietes begonnen werden.

4.3 Erzeugung von Knoten und Dreieckselementen im Rechengebiet

In diesem Abschnitt wird gezeigt, wie aus der ersten aktiven Front der Advancing Front Elemente im Gebietsinnern erzeugt werden. Es stehen zwei Kriterien zur Auswahl der Basis des neu zu erzeugenden Dreiecks zur Verfügung. Außerdem kann der Anwender vorgeben, ob gleichseitige oder gleichschenklige Dreiecke erzeugt werden sollen, vorausgesetzt die Geometrie läßt dies zu.

Zur Erzeugung eines Knotens im Rechengebiet wird eine Kante der Advancing Front, die als Basis des neu zu erzeugenden Dreiecks verwendet wird, entweder nach dem *Winkelkriterium* oder dem *Kantenkriterium* ausgewählt. Wird das *Winkelkriterium* gewählt, werden alle Winkel, die die Kanten der Advancing Front bilden, bestimmt und mit dem Sortierverfahren *Quicksort* (erstmalig 1962 von Hoare [52] veröffentlicht) nach aufsteigenden Winkeln sortiert. Bei der Wahl des *Kantenkriteriums* werden die Kanten der Advancing Front nach aufsteigender Länge ebenfalls mit Quicksort sortiert.

Als erstes wird das *Winkelkriterium* diskutiert und im Anschluß gezeigt, daß die Bestimmung eines neuen Knotens bei Wahl des Kantenkriteriums analog vorgenommen werden kann. Jeder Winkel der aktiven Front wird von zwei Kanten der aktiven Front gebildet. Die kleinere Kante wird als Basis des neu zu erzeugenden Dreiecks gewählt. Aus den Koordinaten der Endknoten der Basiskante wird die Position des neuen Knotens berechnet. Durch Kontrollfunktionen werden die Fälle erkannt, bei denen kein gleichseitiges bzw. gleichschenkliges Dreieck möglich ist und Dreiecke erzeugt, die an die Geometrie oder die aktive Front der Advancing Front angepaßt sind.

Es folgt die Beschreibung der Generierung gleichseitiger Dreiecke (siehe auch Abbildung 4.5). Mit $\vec{r}_1 = (x_1, y_1)$, $\vec{r}_2 = (x_2, y_2)$ werden die Ortsvektoren zu den beiden Endknoten der ausgewählten Kante bezeichnet (Abbildung 4.5). Der Einheitsdifferenzvektor $\Delta\vec{r}$ und der dazu senkrechte Vektor $\Delta\vec{r}^\perp$ aus \vec{r}_1 und \vec{r}_2 ist gegeben durch:

$$\Delta\vec{r} = \frac{\vec{r}_1 - \vec{r}_2}{|\vec{r}_1 - \vec{r}_2|} = \frac{(x_1 - x_2, y_1 - y_2)}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}} \quad (4.1)$$

$$\Delta\vec{r}^\perp = \frac{(y_1 - y_2, x_2 - x_1)}{\sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2}} \quad (4.2)$$

Die Höhe h des gleichseitigen Dreiecks beträgt:

$$h = |\vec{r}_1 - \vec{r}_2| \frac{\sqrt{3}}{2}$$

Die endgültige Position des neuen Knotens P_3 mit den Koordinaten (x_3, y_3) ergibt sich aus:

$$(x_3, y_3) = \vec{r}_2 + \frac{\vec{r}_1 - \vec{r}_2}{2} + \Delta\vec{r}^\perp \cdot h \quad (4.3)$$

Die Erzeugung des neuen Knotens P_3 mit den Koordinaten (x_3, y_3) wird in Abbildung 4.5 illustriert.

Wird die Option für gleichschenklige Dreiecke gewählt, besteht die Möglichkeit, eine globale Maschenweite h vorzugeben oder durch eine vom Benutzer vorgegebene Maschenweitestrukturfunktion, die auch als Dichtefunktion bezeichnet wird, adaptive Netze zu erzeugen. Die Maschenweite des Dreiecks wird als dessen Höhe aufgefaßt.

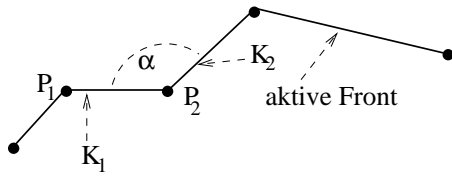
Analog zur Bestimmung des neuen Punktes bei der Generierung gleichseitiger Dreiecke wird der Einheitsdifferenzvektor $\Delta\vec{r}/|\Delta\vec{r}|$ gemäß Gleichung (4.1) bestimmt. Bei Vorgabe einer globalen Maschenweite h_{global} wird die Position des neuen Knotens durch den folgenden Ausdruck berechnet:

$$(x_3, y_3) = \frac{\vec{r}_1 - \vec{r}_2}{2} + \frac{\Delta\vec{r}^\perp}{|\Delta\vec{r}^\perp|} \cdot h_{global} \quad (4.4)$$

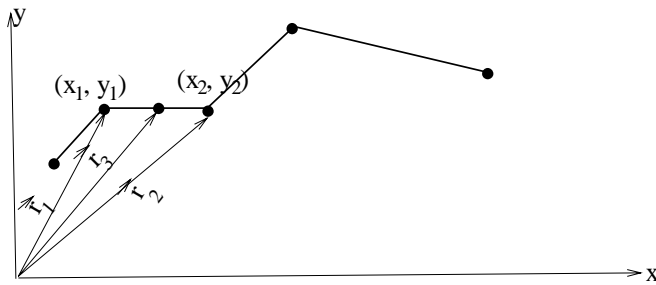
Die Maschenweite h kann als Funktion des Ortes mit $h(x, y)$ durch eine vom Anwender bestimmte Maschenweitestrukturfunktion definiert werden. In diesem Fall wird vor der Erzeugung des neuen Punktes der Vektor \vec{r}_s des Schwerpunktes eines gleichseitigen Hilfsdreiecks berechnet (vergl. Abbildung 4.5) und an dieser Stelle die lokale Maschenweite bestimmt:

$$\begin{aligned} \vec{r}_s &= \frac{1}{3} (\vec{r}_1 + \vec{r}_2 + \tilde{\vec{r}}_3) \\ \text{mit} \\ \tilde{\vec{r}}_3 &= \vec{r}_2 + \frac{\vec{r}_1 - \vec{r}_2}{2} + (\vec{r}_1 - \vec{r}_2)^\perp \cdot \sqrt{\frac{3}{4}} \end{aligned}$$

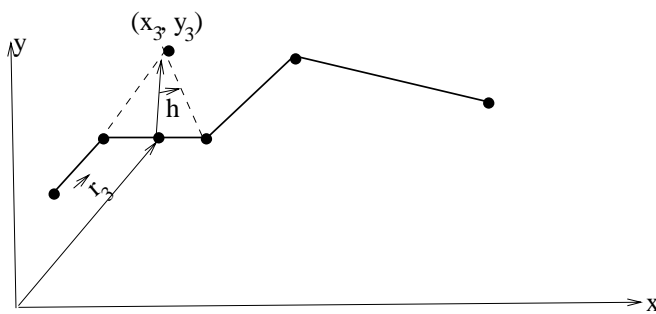
Mit den Koordinaten (x_s, y_s) des Vektors \vec{r}_s wird aus der Maschenweitestrukturfunktion die lokale Maschenweite $h(x_s, y_s)$ bestimmt. Die lokale Maschenweite $h(x_s, y_s)$ ist



$K_1, K_2 =$ Verbindungskanten
 $\alpha =$ kleinster Winkel



$\vec{r}_1, \vec{r}_2 =$ Ortsvektoren zu P_1 und P_2
 $\vec{r}_3 = (\vec{r}_1 + \vec{r}_2) / 2$



$$\vec{h} = \vec{r}_3 / 2 + (\vec{r}_1 - \vec{r}_2) \frac{\vec{r}_3^\perp}{|\vec{r}_3|} h$$

Abbildung 4.5: Erzeugung eines Knotens im Rechengebiet nach dem Winkelkriterium

die Höhe des neu zu erzeugenden Dreiecks und die Position des neuen Knotens mit den Koordinaten (x_3, y_3) folgt aus:

$$(x_3, y_3) = \vec{r}_2 + \frac{\vec{r}_1 - \vec{r}_2}{2} + \Delta\vec{r}^\perp \cdot h(x_s, y_s) \quad (4.5)$$

Bei Anwendung des *Kantenkriteriums* ist die Basis des neu zu erzeugenden Dreiecks die kleinste Kante der aktiven Front. Wiederum kann die Option zur Erzeugung eines gleichseitigen oder gleichschenkligen Dreiecks mit konstanter oder ortsabhängiger Maschenweite gewählt werden.

Die Position des neuen Knotens mit den Koordinaten (x_3, y_3) im gleichseitigen Dreieck erfolgt nach Gleichung (4.3) und somit analog zur Bestimmung des neuen Knotens bei Wahl des Winkelkriteriums.

Mit der Vorgabe, das neue Dreieck gleichschenklig zu konstruieren, können homogene und adaptive Gitter erzeugt werden. Bei Vorgabe einer globalen Maschenweite h_{global} ergibt sich die Position des neuen Knotens nach Gleichung (4.4). Wird eine Dichtekontrollfunktion zur Bestimmung der lokalen Maschenweite $h(x_s, y_s)$ verwendet, folgen die Koordinaten des neuen Punktes aus Gleichung (4.5).

Das Kantenkriterium ist dem Winkelkriterium bei der Vernetzung komplexer Gebiete mit inneren Rändern deutlich überlegen, da bei Verwendung des Winkelkriteriums spitzwinklige Dreiecke entstehen können. Das Winkelkriterium zeigt allerdings bei der Triangulierung einfacher Gebietes im Vergleich zum Kantenkriterium deutliche Qualitätsvorteile.

4.4 Kontrollfunktionen

In diesem Abschnitt werden die vier Kontrollfunktionen zur Steuerung der Advancing Front und zur Kontrolle der Qualität der Elemente vorgestellt und die prinzipiellen Eigenschaften zusammenfassend erläutert. In den nachfolgenden Unterabschnitten folgt eine genaue Beschreibung der Kontrollfunktionen.

Zusammenfassung der Eigenschaften der Kontrollfunktionen:

Nachdem ein vorläufig neuer Knoten nach dem Kanten- oder Winkelkriterium (vergl. Abschnitt 4.3) ausgewählt wurde, wird mit der Kontrollfunktion *Nearer-Neighbor* geprüft, ob ein bereits vorhandener Knoten der aktiven Front zur Drei-

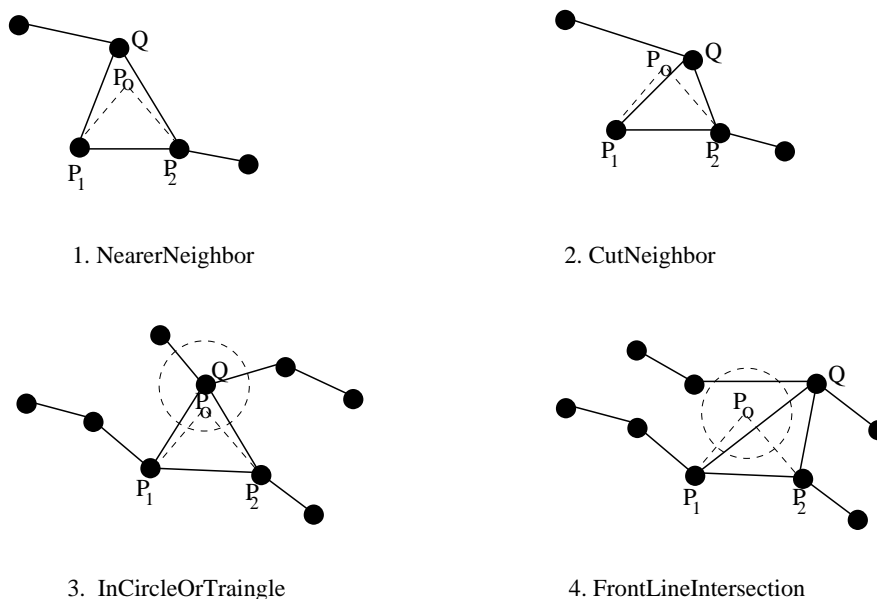


Abbildung 4.6: Kontrollfunktionen

eckserzeugung verwendet werden kann (siehe Abbildung 4.6 1.). Die gestrichelten Linien in Abbildung 4.6 geben die Kanten des vorläufigen Wunschdreiecks, die offenen Kreise die Lage des vorläufig neuen Knotens an. Findet die Kontrollfunktion *NearerNeighbor* einen geeigneten Knoten, werden rekursiv die Funktionen *InCircleOrTriangle* und *FrontLineIntersection* aufgerufen und geprüft, ob sich noch weitere Knoten bzw. Kanten der Advancing Front innerhalb des Dreiecks $P_1 P_2 Q$ befinden (siehe Abbildung 4.6). Werden keine weiteren Knoten oder Kanten gefunden, entsteht das neue Dreieck $P_1 P_2 Q$ und der vorläufig neue Knoten P wird verworfen. Wird durch die Kontrollfunktion *NearerNeighbor* kein geeigneter Knoten gefunden, erfolgt der Aufruf der Kontrollfunktion *CutNeighbor*.

Die Kontrollfunktion *CutNeighbor* prüft durch rekursives Aufrufen der Kontrollfunktionen *InCircleOrTriangle* und *FrontLineIntersection*, ob die Kanten des neu zu erzeugenden Dreiecks im Rechengebietes liegen. Ist dies der Fall und befinden sich keine weiteren Knoten oder Kanten innerhalb des neu zu erzeugenden Dreiecks, entsteht das Dreieck $P_1 P_2 Q$ (Abbildung 4.6 2.).

Die Kontrollfunktion *InCircleOrTriangle* stellt fest, ob ein Knoten der aktiven Front innerhalb des in Abbildung 4.6.3 dargestellten Suchgebietes befindet. Falls keine Kanten durch das Suchgebiet verlaufen, entsteht das Dreieck $P_1 P_2 Q$ (Abbildung 4.6.3). Die aktive Front wird aufgespalten und es entsteht eine neue Frontliste.

Mit Kontrollfunktion *FrontLineIntersection* werden die Kanten, die das Suchgebiet in Abbildung 4.6.4 schneiden, bestimmt und ein Dreieck mit dem Knoten der Kante mit kürzestem Abstand zur Basis des neu zu erzeugenden Dreiecks gebildet. Nach Abbildung 4.6.4 entsteht das Dreieck P_1P_2Q . Die aktive Front wird aufgespalten und es entsteht eine neue Frontliste.

Falls die Kontrollfunktionen keine geeigneten Knoten der aktiven Front finden, wird der neue Knoten P erzeugt. Der neue Knoten mit den zugehörigen Kanten wird in die aktive Front aufgenommen. Die Kanten, die durch die Konstruktion des neuen Dreiecks nicht mehr zur aktiven Front gehören, werden aus den Frontlisten gelöscht und die Advancing Front aktualisiert.

4.4.1 Kontrollfunktion *NearerNeighbor*

Die Kontrollfunktion *NearerNeighbor* prüft, ob die benachbarte linke oder rechte Kante der aktiven Front als eine der Kanten zur Dreiecksgenerierung verwendet werden kann.

Der Verlauf der aktiven Front der Advancing Front wird in Abbildung 4.7 dargestellt. Der unschraffierte Bereich stellt das innere Rechengebiet dar, der schraffierte Bereich bezeichnet das äußere Gebiet. Die gestrichelt gezeichneten Linien geben die Gestalt des vorläufig neuen, nach den Kriterien in Abschnitt 4.3 berechneten Dreiecks an. Zur Dreieckserzeugung wird der Knoten der aktiven Front mit kürzestem Abstand zum vorläufig neuen Knoten P (Abbildung 4.7) verwendet.

Das bestmögliche Dreieck entsteht in diesem Fall durch die Verwendung einer bereits vorhandenen Kante der aktiven Front (durchgezogene Linie in Abbildung 4.7). Das neue Dreieck besteht aus den Knoten P_1P_2Q der aktiven Front. Die Aktualisierung der Advancing Front geschieht durch *Einfügen* der neuen Kante und *Entfernen* der Basiskante des neuen Dreiecks (vergl. Abschnitt 3.3).

4.4.2 Kontrollfunktion *CutNeighbor*

Mit der Kontrollfunktion *CutNeighbor* wird geprüft, ob die neuen Kanten, die bei der Erzeugung des Punktes P entstehen, außerhalb des Rechengebietes liegen (siehe Abbildung 4.8). In diesem Fall schneiden die Kanten des vorläufig neuen Dreiecks P_1P_2Q die aktive Front. Das Prüfkriterium wird durch das Skalarprodukt

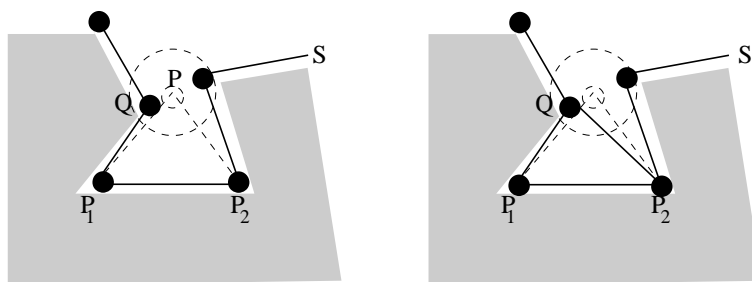


Abbildung 4.7: Bestimmung eines neuen Knotens im Rechengebiet durch die Funktion *NearerNeighbor*

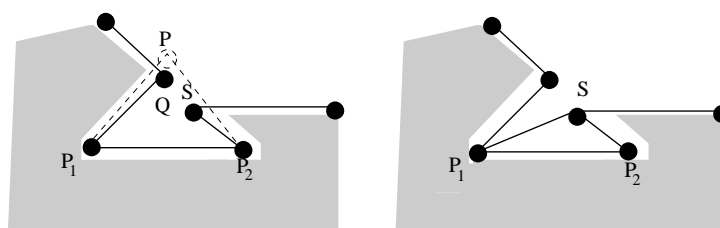


Abbildung 4.8: Bestimmung eines neuen Knotens im Rechengebiet durch die Funktion *CutNeighbor*

der Verbindungsvektoren zu den Knoten P_1 und P (Position des vorläufig neuen Knotens) bzw. P_2 und P definiert. Wird mit \vec{r}_{P_1Q} der Differenzvektor von Knoten P_1 zum Knoten Q der aktiven Front, mit \vec{r}_{P_1P} der Differenzvektor von Knoten P_1 zum vorläufigen Knoten P und mit $\vec{r}_{P_1P}^\perp$ der zu \vec{r}_{P_1P} senkrechte Vektor, der in das Gebietsinnere zeigt, bezeichnet, lautet die Prüfbedingung:

$$\vec{r}_{P_1Q} \cdot \vec{r}_{P_1P}^\perp < 0 \Rightarrow P \text{ befindet sich außerhalb des Rechengebietes} \quad (4.6)$$

Die Lage des Knotens S (Abbildung 4.8) wird analog geprüft. Schneiden zwei Kanten des vorläufig neuen Dreiecks die aktive Front, so wird der Knoten mit dem kleinsten Abstand zur Basis P_1P_2 zur Dreieckskonstruktion verwendet. Mit den Bezeichnungen, die in Abbildung 4.8 verwendet werden, entsteht hier das neue Dreieck P_1P_2Q .

Die Aktualisierung der Advancing Front erfolgt durch *Entfernen* der Kante $\overline{P_1Q}$ und *Einfügen* der Kante $\overline{P_2Q}$ in die aktive Front (vergl. Abschnitt 3.3).

4.4.3 Kontrollfunktion *InCircleOrTriangle*

Die Kontrollfunktion *InCircleOrTriangle* bestimmt die Menge der Knoten der aktiven Front, die sich innerhalb des in Abbildung 4.9 gestrichelt dargestellten Suchgebietes befinden. Das Suchgebiet besteht aus der Überlagerung der Fläche

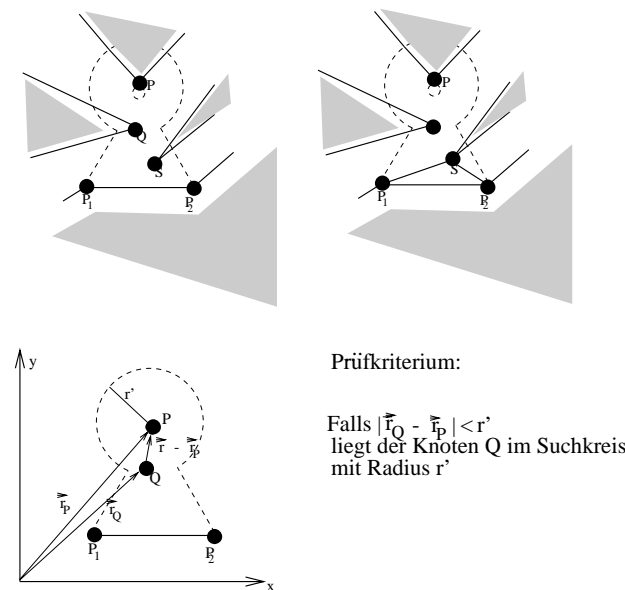


Abbildung 4.9: Bestimmung eines neuen Knotens im Rechengebiet durch die Funktion *InCircleOrTriangle*

des vorläufig neuen Dreiecks und einem Kreis mit den Koordinaten des vorläufig neuen Knotens P als Kreismittelpunkt (gestrichelte Figur in Abbildung 4.9). Der Radius r' bezeichnet den Durchmesser des Suchkreises. Die Bestimmung der Knoten, die innerhalb des durch den Suchkreis abgeschnittenen Dreiecks liegen, (siehe Abbildung 4.9) erfolgt nach dem Kriterium (4.6) der Kontrollfunktion *CutNeighbor*. Die Knoten innerhalb des Kreises mit Radius r' werden durch Längenvergleich von Differenzvektoren und r' gefunden. Werden mit \vec{r}_P und \vec{r}_Q die Ortsvektoren zu den Knoten P und Q und mit $\vec{r}_Q - \vec{r}_P$ der Differenzvektor von Knoten P und Q bezeichnet, liegt ein Knoten im Suchkreis, falls gilt:

$$|\vec{r}_Q - \vec{r}_P| < r'$$

Zur Dreieckserzeugung wird aus den gefundenen Knoten der Knoten mit minimalen Abstand zur Basis $(\overline{P_1P_2})$ des neu zu erzeugenden Dreiecks verwendet.

Die Advancing Front wird durch die in Abschnitt 3.3 beschriebenen Operationen *Entfernen* und *Einfügen* aktualisiert.

4.4.4 Kontrollfunktion *FrontLineIntersection*

Mit der Kontrollfunktion *FrontLineIntersection* wird geprüft, ob eine Kante der aktiven Front das vorläufige neue Dreieck (gestrichelt dargestellt in Abbildung

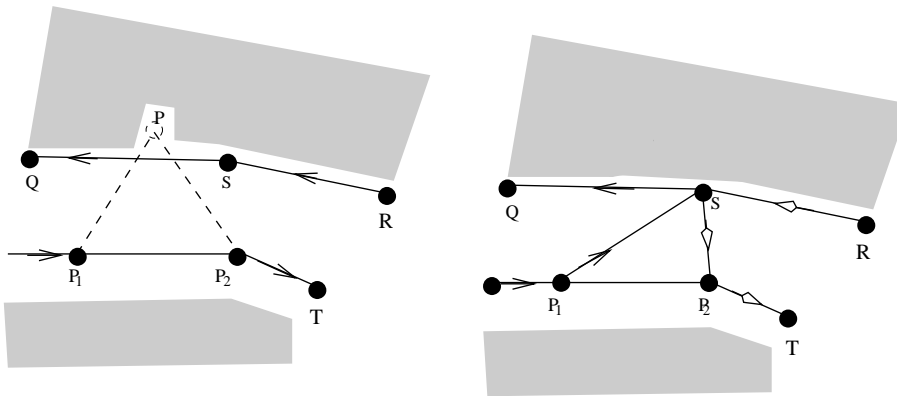


Abbildung 4.10: Bestimmung eines neuen Knotens im Rechengebiet durch die Funktion `FrontLineIntersection`

4.10) schneidet. Die Verbindungskanten und die dazugehörigen Knoten der aktiven Front, die das Geradenstück vom Mittelpunkt der Kante $\overline{P_1P_2}$ schneiden, werden in einer temporären Liste gespeichert. Der Knoten aus dieser Liste mit dem kürzesten Abstand zur Kante $\overline{P_1P_2}$ ergibt den fehlenden Knoten des neu zu erzeugenden Dreiecks. Das neue Dreieck besteht aus den Knoten $P_1P_2P_3$ und den zugehörigen Kanten (Abbildung 4.10).

Durch die Konstruktion des Dreiecks $P_1P_2P_3$ wird die aktive Front aufgespalten. Es entsteht zusätzlich zur existierenden aktiven Front eine weitere aktive Front. Der Verlauf der aktiven Fronten ist in Abbildung 4.10 dargestellt. Die erste aktive Front verläuft von P_1 über S nach Q , die zweite aktive Front verläuft von R über S nach P_2 und T . Die zwei aktiven Fronten werden als zwei unabhängige Frontlisten verwaltet und die entsprechenden Frontlisten in die jeweilige unabhängige Frontliste eingetragen. Eine genaue Beschreibung der Aktualisierung der Listen wird im nachfolgenden Unterabschnitt 4.4.5 gegeben.

4.4.5 Aktualisierung der Frontlisten

In diesem Abschnitt wird die Aktualisierung der aktiven Front nach der Generierung eines neuen Dreiecks beschrieben. Die Kontrollfunktionen (siehe hierzu die Abschnitte 4.4.1 bis 4.4.4) geben die Kriterien an, ob ein neuer Knoten oder ein bereits vorhandener Knoten der aktiven Front zur Dreieckskonstruktion verwendet wird. Die Aktualisierung der aktiven Front hängt von den Ergebnissen der Kontrollfunktionen ab und wird in zwei Fälle unterteilt.

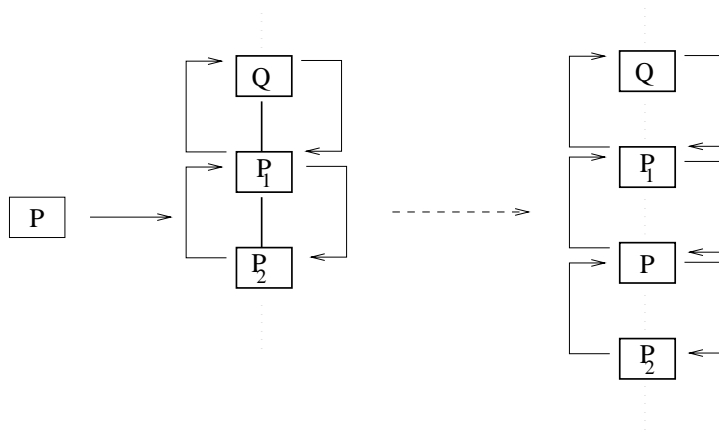
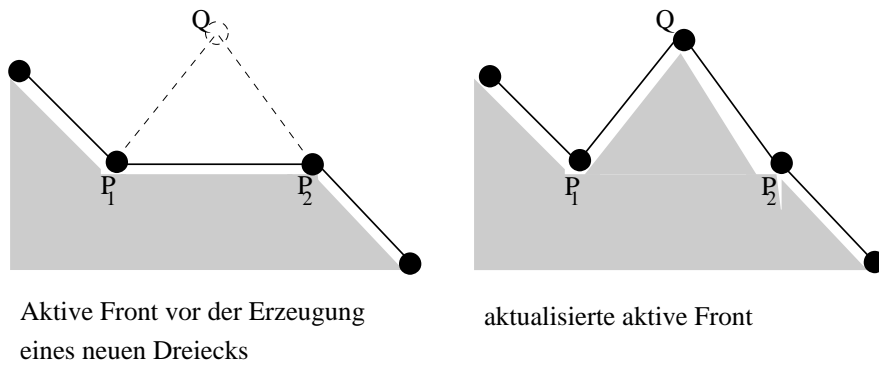
1. Fall: Dreieckserzeugung mit einem neuen Knoten

Zur Dreieckskonstruktion wird der berechnete Knoten P verwendet (siehe Abbildung 4.11). Aus der aktiven Front wird die aktuelle Kante P_1P_2 entfernt und zwei neue Kanten P_1P und PP_2 hinzugefügt. Die Anzahl der Frontkomponenten erhöht sich um 1, während Anzahl der unabhängigen Frontlisten und Frontlisten gleich bleibt.

Der Verlauf der aktiven Front nach der Aktualisierung ist in Abbildung 4.11 dargestellt. Die Verzeigerung der Frontkomponenten der aktiven Front ist aufgrund des linearen zyklischen Listenkonzepts und der dynamischen Speicherverwaltung nicht aufwendig. In Abbildung 4.11 wird die Verzeigerung der Frontkomponenten vor und nach der Erzeugung eines neuen Dreiecks veranschaulicht. Die Aktualisierung besteht in der Bereitstellung von Speicherplatz für den neuen Knoten P und dem Einfügen des Knotens P in die Liste der Frontkomponenten. Das Einfügen geschieht durch das Setzen eines Zeigers von Knoten P_1 zu Knoten P und umgekehrt sowie von Knoten P zu Knoten P_2 und umgekehrt (doppelte Verkettung). Die Knoten nach P_1 rücken eine Position nach hinten.

2. Fall: Dreieckserzeugung mit einem Knoten der aktiven Front

Es wird die Vorgehensweise zur Aktualisierung der aktiven Front bei Verwendung eines Knotens aus der aktiven Front behandelt und gezeigt, wie die Verzeigerung der Frontlisten vorgenommen wird. Die Aktualisierung der aktiven Front ist bei den Kontrollfunktionen *NearerNeighbor* und *CutNeighbor* identisch. Deshalb wird in der weiteren Diskussion nur die Kontrollfunktion *NearerNeighbor* betrachtet. Erfolgt die Dreieckserzeugung aufgrund der Ergebnisse der Kontrollfunktion *InCircleOrTriangle* oder *FrontLineIntersection*, kann sich neben der Anzahl der Knoten und Kanten auch die Anzahl der unabhängigen Frontlisten (IFL)



Verzweigung einer neuen Frontkomponente der aktiven Front

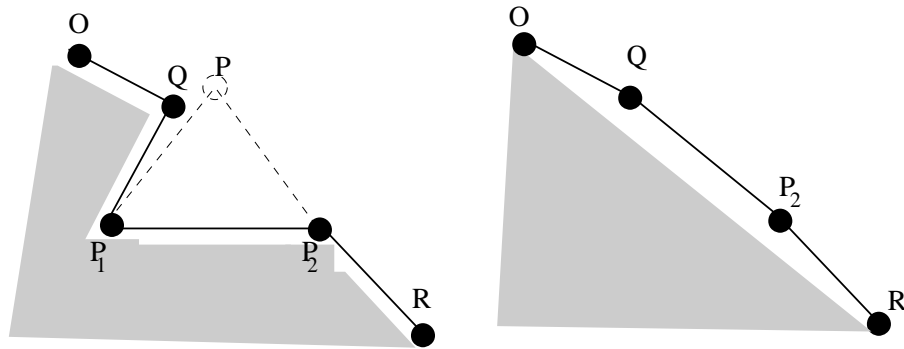
Abbildung 4.11: Aktive Front vor und nach der Erzeugung eines Dreiecks mit neuem Knoten

und Frontlisten ändern. Die Datenstrukturen ermöglichen jedoch auch eine einfache Behandlung dieser Fälle. Die Aktualisierung der aktiven Front für die Kontrollfunktionen *InCircleOrTriangle* und *FrontLineIntersection* sind gleich und deshalb wird im folgenden nur der Fall für die Kontrollfunktion *FrontLineIntersection* betrachtet.

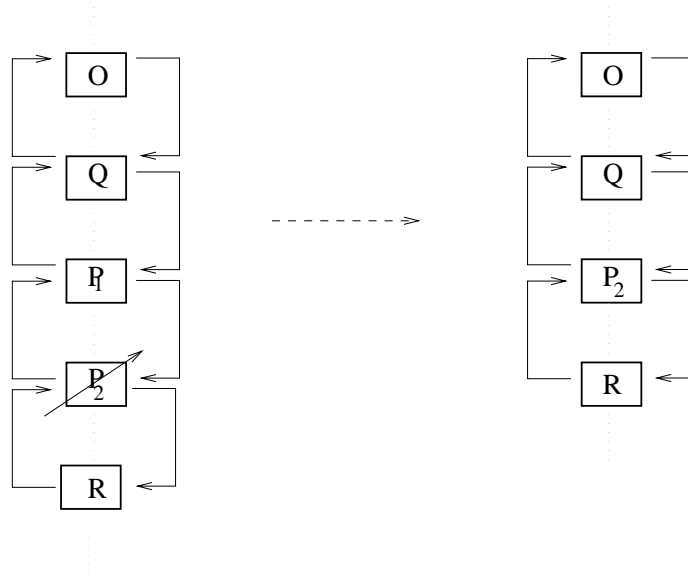
Zuerst wird die Aktualisierung der aktiven Front nach der Erzeugung eines neuen Knotens durch die Kontrollfunktion *NearerNeighbor* diskutiert. Die Lage der aktiven Front ist in Abbildung 4.12 dargestellt. Die Aktualisierung der aktiven Front besteht aus dem Entfernen des Knotens P_1 und den Kanten P_1Q und P_1P_2 . Eine neue Kante QP_2 wird erzeugt und die Anzahl der Frontkomponenten wird um 1 verringert. Der Kante QP_2 wird Speicherplatz zugewiesen und der Speicherplatz für den Knoten P_1 freigegeben (dynamische Speicherverwaltung). Die neue Verzeigerung der Frontkomponenten ist in Abbildung 4.12 demonstriert.

Die Aktualisierung der aktiven Front nach der Erzeugung eines neuen Dreiecks durch die Kontrollfunktion *FrontLineIntersection* wird in Abbildung 4.13 veranschaulicht. Aus der aktiven Front wird die Kante P_1P_2 entfernt und zusätzlich zur vorhandenen unabhängigen Frontliste IFL_1 und Frontliste FL_1 die neue unabhängige Frontliste IFL_2 erzeugt. Die Frontliste FL_1 wird geteilt und es entsteht die neue Frontliste FL_2 , die der unabhängigen Frontliste IFL_2 zugeordnet wird. Die unabhängige Frontliste IFL_1 enthält dann die Frontliste FL_1 und die Knoten P_1, P'_1, Q', \dots . Die neu erzeugte unabhängige Frontliste IFL_2 besteht aus der Frontliste FL_1 und den Knoten $S', P'_2, P'_1, P_2, S, \dots$. Der Knoten P'_1 wird hier verdoppelt und in beiden Frontlisten verwaltet. Jede der beiden unabhängigen Frontlisten IFL_1 und IFL_2 können unabhängig voneinander und nacheinander zur Dreieckerzeugung verwendet werden.

Bei der Gittergenerierung kann eine aktive Front so verlaufen, daß Teile davon aufeinandertreffen und Frontlisten verschmelzen. In diesen Fällen verringert sich die Anzahl der unabhängigen Frontlisten und Frontlisten. Aufgrund der dynamischen Speicherverwaltung ist auch in diesen Fällen die Aktualisierung der aktiven Front problemlos möglich.

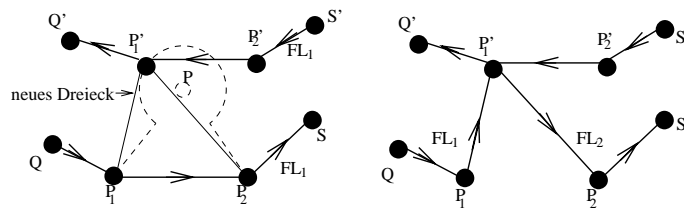


Aktive Front vor und nach der Erzeugung eines neuen Dreiecks durch die Kontrollfunktion *NearerNeighbor*

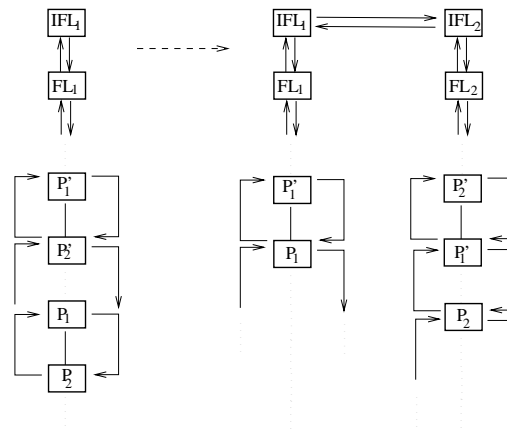


Verzeigerung der Frontkomponenten vor und nach der Dreieckszeugung

Abbildung 4.12: Aktualisierung der aktiven Front nach der Dreieckszeugung durch die Kontrollfunktion *NearerNeighbor*



Aktive Front vor und nach der Erzeugung eines neuen Dreiecks durch die Funktion *FrontLineIntersection*



Verzögerung der Frontkomponenten vor und nach der Dreieckserzeugung

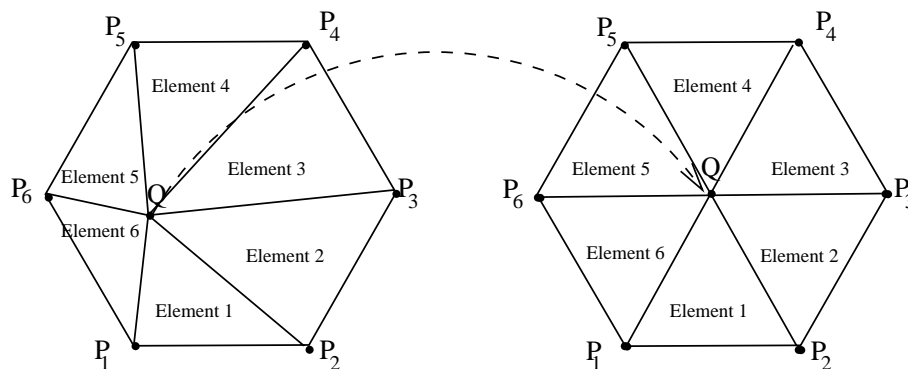
Abbildung 4.13: Aktualisierung der aktiven Front nach der Dreieckserzeugung durch die Kontrollfunktion *FrontlineIntersection*

4.5 Verbesserung der Gitterqualität

In diesem Abschnitt werden die Methoden zur Verbesserung der Gitterqualität nach der Gittererzeugung vorgestellt. Verwendet wird die bekannte Laplace-Glättung, die bereits in früheren Arbeiten zur Netzglättung eingesetzt worden ist (vergl. z. B. [24], [65]). Außerdem wird eine Modifikation der Laplace-Glättung für nichtkonvexe Gebiete vorgestellt.

4.5.1 Laplace-Glättung

Durch die Laplace-Glättung wird ein Knoten eines Rechengitters von seiner momentanen Position zum Schwerpunkt der Fläche der ihn umgebenden Dreiecke verschoben. Die Laplace-Glättung eignet sich sehr gut zur Glättung konvexer Gebiete und kann im allgemeinen nicht auf nichtkonvexe Gebiete übertragen werden. Da alle Knoten, auch Randknoten, verschoben werden, können bei Gebieten mit einspringenden Ecken nach der Verschiebung des Knotens und aktualisieren der aktiven Front auch Kanten ausserhalb des Rechengebietes liegen. Dieser Man-

Abbildung 4.14: Verschiebung des Knotens Q durch Laplace-Glättung

gel wird durch die modifizierte Laplace-Glättung, die im nachfolgenden Unterabschnitt beschrieben wird, behoben.

Durch iteratives Anwenden der Laplace-Glättung auf alle Knoten wird die Anzahl von Dreiecken mit spitzen Winkeln verringert. Der iterative Prozess wird durch eine vom Benutzer vorgegebene Anzahl von Glättungsschritten gesteuert.

Wird mit \vec{r}_i der Ortsvektor zum Knoten P_i , mit \vec{r}_Q der Ortsvektor zum zu verschiebenden Knoten Q und mit n die Anzahl der Dreiecke, die den Knoten Q umgeben, bezeichnet, ergibt sich für die Verschiebung des Knotens Q (siehe Abbildung 4.14) der Vektor \vec{r}_Q nach Gleichung:

$$\vec{r}_Q = \frac{1}{n} \sum_{i=1}^n \vec{r}_i$$

In Abbildung 4.14 wird die Wirkungsweise der Laplace-Glättung veranschaulicht.

4.5.2 Modifizierte Laplace-Glättung

Die modifizierte Laplace-Glättung garantiert, daß auch bei nichtkonvexen Gebieten alle Knoten des Rechengitters nach einer Verschiebung innerhalb des Rechengebietes liegen. Die Vorgehensweise besteht darin, ein Funktional $F(x,y)$ zu minimieren. Das Funktional $F(x,y)$ faßt die Summe der Abstandskquadrate der Strecken von Knoten Q zu den umgebenden Knoten P_i und die Abstandskquadrate der Strecken des Knoten Q zu den umgebenden Kanten zusammen. Das Minimum von $F(x,y)$ ergibt die Koordinaten des Knotens nach der Verschiebung. Dieser Vorgang wird nachfolgend präzisiert.

Der Ortsvektor zum verschiebenden Knoten Q wird mit \vec{r}_Q mit den Koordina-

ten (x, y) bezeichnet. Ferner sind die Ortsvektoren und Koordinaten zu den Knoten P_i und P_{i+1} durch die Ortsvektoren \vec{P}_i, \vec{P}_{i+1} und den Koordinaten $(x_i, y_i), (x_{i+1}, y_{i+1})$ gegeben. Die modifizierte Laplace-Glättung minimiert das durch den folgenden Ausdruck definierte Funktional $F(x, y)$:

$$F(x, y) = \sum_{i=0}^n \sqrt{(x - x_i)^2 + (y - y_i)^2} + (G_i)^{-2}$$

Die Funktion G_i beschreibt den Abstand des zu verschiebenden Knotens Q zu einer Kante, die durch 2 Knoten mit den Koordinaten (x_i, y_i) und (x_{i+1}, y_{i+1}) (benachbarte Knoten) definiert ist. Die Funktion G_i ist definiert als:

$$G_i = \frac{Ax + By + C}{\sqrt{A^2 + B^2}}$$

Die Koeffizienten A, B, C sind gegeben durch:

$$A = y_{i+1} - y_i$$

$$B = -(x_{i+1} - x_i)$$

$$C = -Ax_i + By_i$$

Die Koordinaten des zu verschiebenden Knotens werden durch die Minimierung des Funktionals $F(x, y)$ bestimmt. Die numerische Bestimmung des Minimum des Funktionals $F(x, y)$ erfolgt durch ein Newton-Verfahren. Die Berechnung der neuen Lage des zu verschiebenden Knotens als Minimum von $F(x, y)$ garantiert, daß der Knoten innerhalb des Rechengebietes liegt. Eine neuere Arbeit zur Netzglättung nichtkonvexer Gebiete ist in [59] zu finden. In [59] werden im Unterschied zu der hier vorgestellten Methode die Knoten gewichtet verschoben und dadurch eine anisotrope Glättung ermöglicht. Die Gewichtung der zu verschiebenden Knoten muß allerdings vom Benutzer vorgegeben werden. Anwendung findet die Methode in [59] z. B. bei der Berechnung von Ladungstransporten in Halbleitern. Ein weiteres interessantes Glättungsverfahren, das für Rechengitter, die aus Dreiecks-, Vierecks- oder gemischt aus Dreiecks- und Viereckselementen bestehen, anwendbar ist, wird in [22] vorgestellt.

4.6 Anwendungen des Gittergenerators

In diesem Kapitel werden Beispiele von Rechengittern auf einfachen und komplexen Gebieten mit vielen inneren Rändern vorgestellt. Die adaptive Anpassung des Gitters an die vorgegebene Geometrie durch eine Gittermaschenweite- bzw. Dichtekontrollfunktion wird anhand der Triangulierung der Nordseeküste in Unterabschnitt 4.6.1 demonstriert. Bei der Diskussion des Beispiels Nordseeküste wird ebenfalls die Wirkungsweise des Advancing Front durch eine Darstellung der sukzessiven Dreieckerzeugung in das Gebietsinnere veranschaulicht. Ein weiteres Beispiel in Unterabschnitt 4.6.2 stammt aus der Elastizitätstheorie und zeigt ein Gitter, wie es als Grobgitter für eine Mehrgittermethode eines Prandtl-Reuß-Plastizitätsproblems verwendet worden ist [118]. Ein weiteres Beispiel eines Grobgitters wird anhand der Vernetzung des Wolfgangsees gezeigt. Als letztes Beispiel in Unterabschnitt 4.6.2 wird die Triangulierung eines Schnittes durch einen Salzstock dargestellt.

4.6.1 Beispiel Nordseeküste

Als Beispiel einer komplexen Geometrie mit vielen inneren Rändern wird die Triangulierung der Nordseeküste in der Gegend der deutschen Bucht betrachtet. Das Gebiet enthält als innere Ränder die west-, ost- und nordfriesischen Inseln sowie Helgoland (siehe Abbildung 4.15). Alle Längenmaße sind auf den dimensionslosen Wert 1 skaliert. Die Geometrie muß hinreichend genau im Bereich der Inseln aufgelöst werden, d.h., jede Insel ist durch mindestens ein Dreieck darzustellen. Das ganze Gebiet mit einer konstanten Maschenweite zu triangulieren bedeutet, sehr viele Elemente auch in den Raumbereichen, in denen eine feine Auflösung nicht erforderlich ist, zu erzeugen. Durch Vorgabe einer Dichtekontrollfunktion kann ein optimal an die Geometrie angepaßtes Gitter erzeugt werden. Die Dichtekontrollfunktion wird nachfolgend erläutert.

Mit $D(x,y)$ wird die Dichtekontrollfunktion zur Bestimmung der lokalen Maschenweite am Ort (x,y) des Rechengebietes Nordseeküste bezeichnet und folgendermaßen definiert:

$$D(x,y) = (H - h) * G(s) + h \quad (4.7)$$

Die Funktion $G(S)$ in Gleichung 4.7 ist eine Funktion des senkrechten Abstand s eines Punktes (x,y) des Gebietes Nordseeküste zum rechten Rand der Nordseeküste (Abbildung 4.15). $D(s)$ beschreibt das stetige Anwachsen der lokalen Maschenweite vom kleinsten Wert h zum maximalen Wert H . Die Funktion $G(s)$

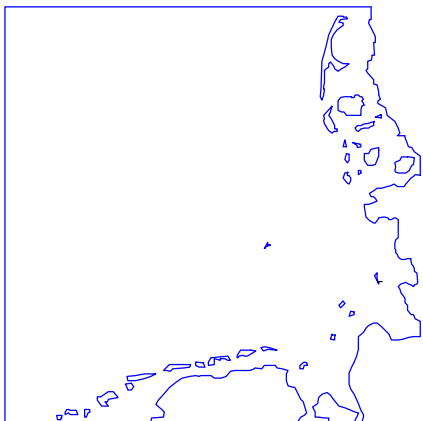
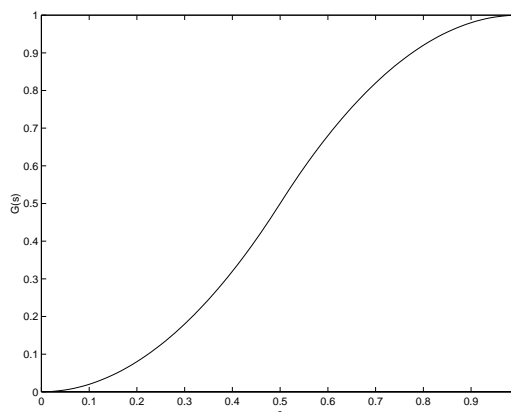


Abbildung 4.15: Gebiet Nordseeküste

Abbildung 4.16: Funktion $G(s)$

wird definiert als:

$$G(s) = \begin{cases} 2s^2 & \text{für } 0 \leq s < \frac{1}{2}, \\ -2(s-1)^2 + 1 & \text{für } \frac{1}{2} \leq s \leq 1 \end{cases} \quad (4.8)$$

Zur Bestimmung des Eingabeparameter s der Funktion $G(s)$ wird eine Kante der aktiven Front nach dem Winkel- oder Kantenkriterium ausgewählt und der Schwerpunkt des gleichseitigen Dreiecks mit Seitenlänge der ausgewählten Kantenlänge berechnet. Anschließend wird der Abstand s des Schwerpunkts zum rechten Rand des Nordseeküstengebietes bestimmt. Beträgt dieser Abstand s zum rechten Rand z. B. $s = \frac{1}{2}$, wird dem neu zu erzeugenden Element die Maschenweite $(H + h)/2$ zugeordnet. Mit Abstand $s = 1$ ist der linke Rand erreicht und die Maschenweite der Elemente beträgt H . In Abbildung 4.16 ist der Verlauf der Funktion $G(s)$ dargestellt.

Als weiteres Beispiel wird in den Abbildungen 4.19 bis 4.24 veranschaulicht, wie die Advancing Front in das Gebietsinnere des Nordseeküstengebietes wandert. Als grobe Maschenweite wird hier $H = 0.1$ und als feine Maschenweite $h = 0.01$ gewählt. Das vollständige Gitter besteht aus 3548 Knoten und 6509 Elementen. Alle Winkel α der Dreiecke des Gitters befinden sich im Intervall $20^\circ < \alpha < 113^\circ$. Zur Auswahl einer Kante der aktiven Front wurde die kleinste Kante der aktiven Front (Kantenkriterium, vergl. Kapitel 4.3) verwendet.

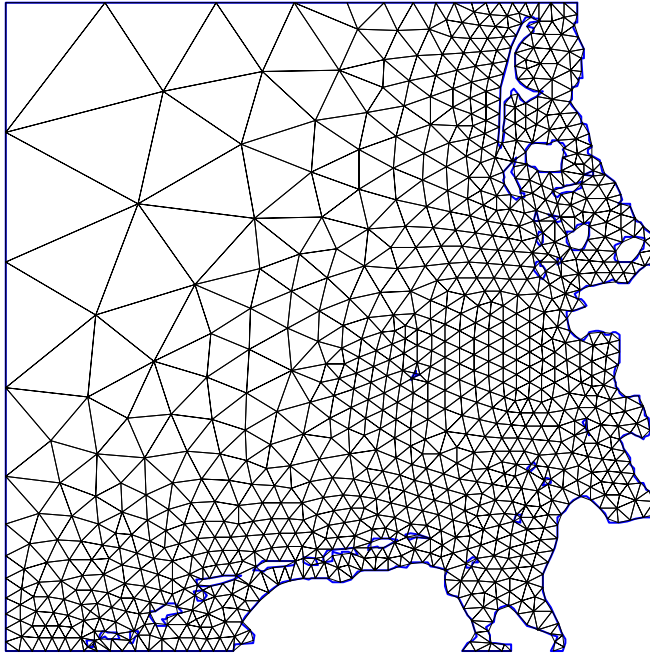


Abbildung 4.17: Triangulierung der Nordseeküste mit grober Maschenweite $H = 0.2$ und feiner Maschenweite $h = 0.02$

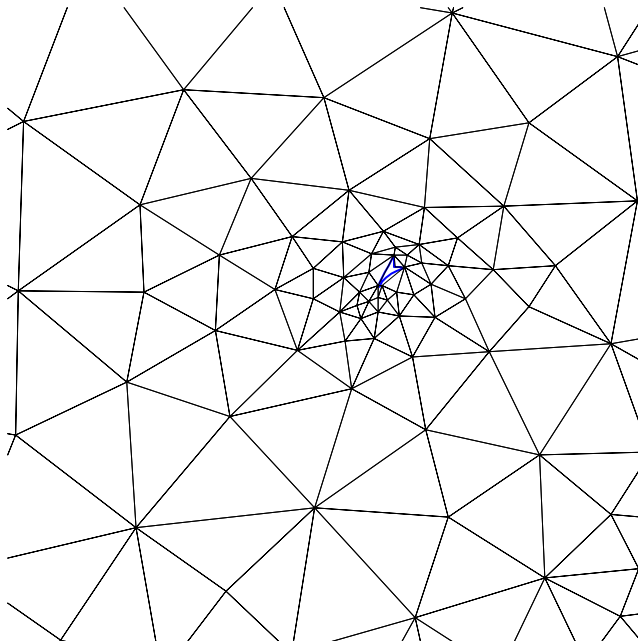


Abbildung 4.18: Gitter in der Umgebung der Insel Helgoland

Eine Triangulierung der Nordseeküste unter Verwendung der Dichtekontrollfunktion $D(x,y)$ ist in Abbildung 4.17 dargestellt. Es wurde eine grobe Maschenweite $H = 0.2$ und eine feine Maschenweite $h = 0.02$ verwendet. Das mit diesen Parametern entstandene Gitter besteht aus 1013 Knoten und 1747 Elementen und stellt ein grobes Gitter für Mehrgitterverfahren mit hinreichend genauer Auflösung der Geometrie dar. Die Winkel α der Dreiecke liegen im Bereich $20^\circ < \alpha < 120^\circ$. Die Dichtekontrollfunktion ist so konzipiert, daß die Maschenweite der Dreiecke zum linken Rand anwächst, aber um die Insel Helgoland wiederum feiner wird. Ein Ausschnitt des Gitters in einer Umgebung der Insel Helgoland ist in Abbildung 4.18 dargestellt.

4.6.2 Weitere Beispiele

In diesem Kapitel werden weitere Anwendungen des Gittergenerators vorgestellt. Das komplizierte Gebiet eines Salzstocks wurde im Rahmen des Forschungsvorhabens *Schnelles Grundwasserprogramm* [33] zur Berechnung dichtegetriebener Strömungen trianguliert. Die Geometrie (siehe Abbildungen 4.25 und 4.26) besteht aus 49 inneren Rändern, die teilweise sehr dicht aneinander anschließen. Die horizontale Ausdehnung der Geometrien in den Abbildungen 4.25 und 4.26 beträgt 300 m, die vertikale Ausdehnung beträgt 70 m. Abbildung 4.25 wurde mit einer konstanten Maschenweite $h = 10m$, Abbildung mit einer konstanten Maschenweite $h = 30m$ vernetzt. Die damit entstandenen Gitter konnten weder mit einem kommerziellen noch mit einem anderen im universitären Bereich verbreiteten Gittergenerator erzeugt werden. Die Winkel α der Dreiecke in den Abbildungen 4.25 und 4.26 liegen in beiden Gittern im Bereich $20^\circ < \alpha < 120^\circ$.

Eine weiteres Beispiel ist die Erzeugung eines Grobgitters für das Wolfgangsee-Gebiet. In Abbildung 4.28 ist ein Gitter mit 113 Knoten und 169 Elementen dargestellt. Die Winkel α der Dreiecke des Gitters befinden sich im Bereich $61^\circ < \alpha < 128^\circ$. Eine Verbesserung der Randauflösung durch eine feinere Triangulierung ist in Abbildung 4.28 dargestellt. Die Winkel der Dreiecke liegen hier im Intervall $54^\circ < \alpha < 131^\circ$.

Eine Anwendung des Gittergenerators zur Erzeugung eines Grobgitters für eine Mehrgittermethode zur Berechnung der Prandtl-Reuß-Plastizität einer ebenen Platte [118] mit einer finiten Elemente Diskretisierung ist in den Abbildungen 4.29 und 4.30 illustriert. Die Abbildungen 4.29 und 4.30 zeigen nur ein Viertel der Geometrie, da Symmetrieeigenschaften ausgenutzt werden können. Als

Grobgitter des Mehrgitterverfahrens wird das Gitter in Abbildung 4.29 benutzt. In Abbildung 4.30 ist das durch einen Fehlerindikator nach [118] adaptiv verfeinerte Gitter dargestellt.

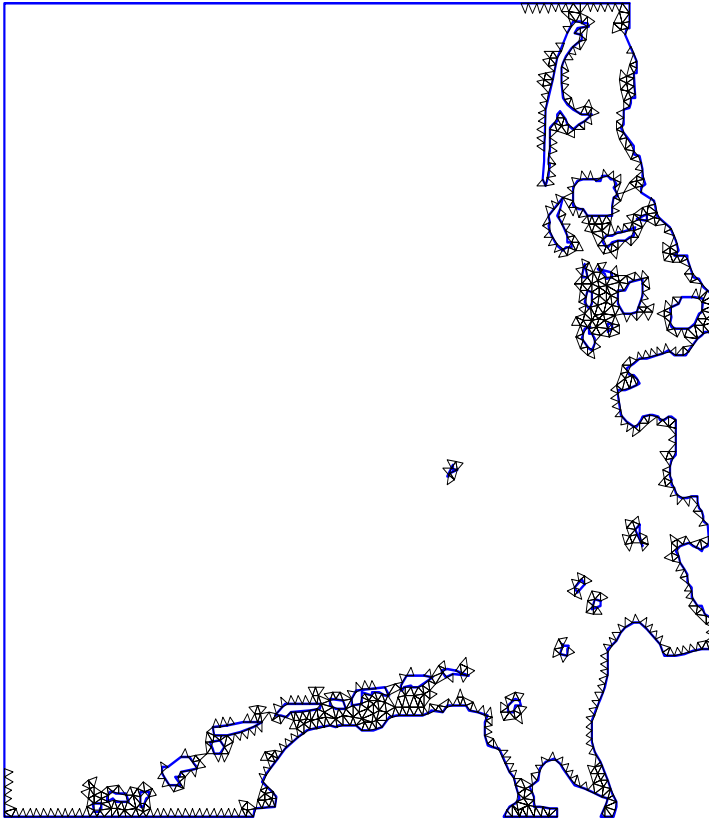


Abbildung 4.19: Advancing Front mit 1200 Knoten und 1000 Elementen

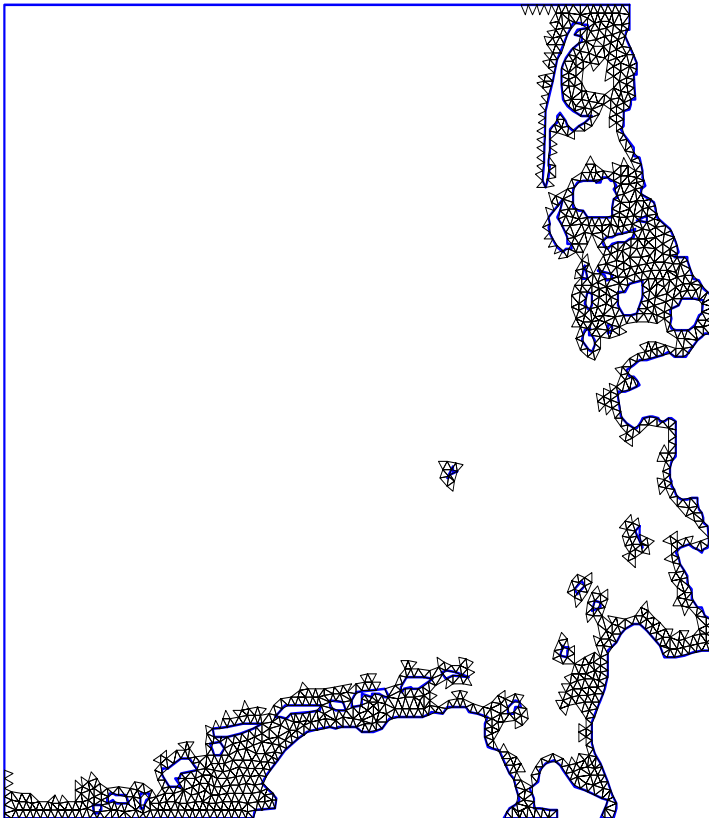


Abbildung 4.20: Advancing Front mit 1558 Knoten und 2000 Elementen

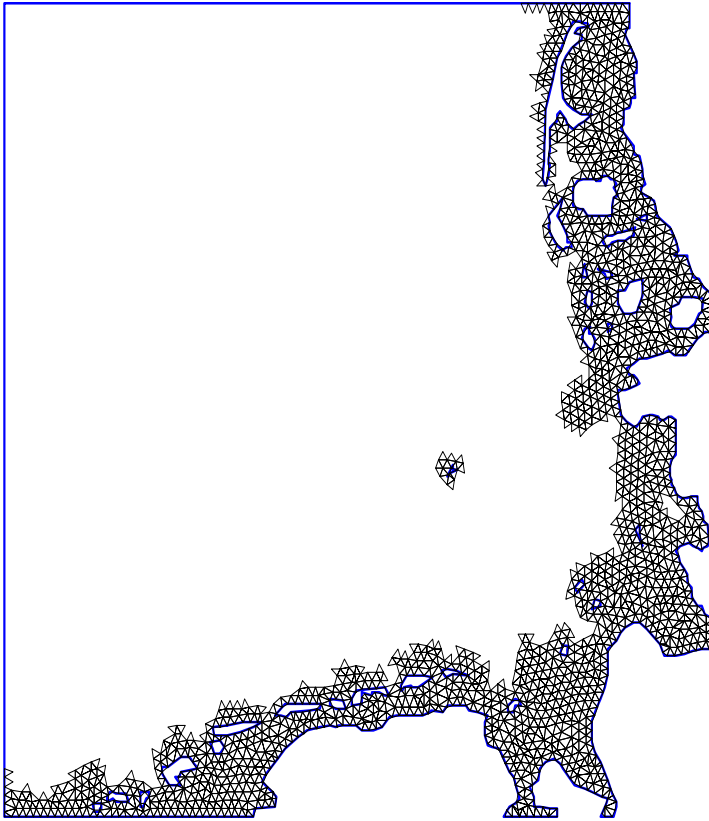


Abbildung 4.21: Advancing Front mit 1970 Knoten und 3000 Elementen

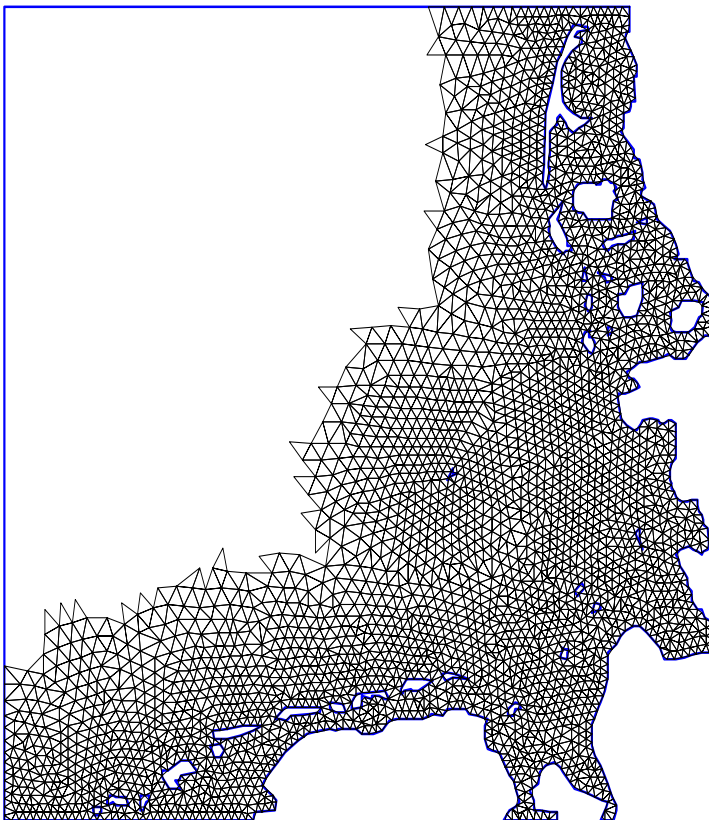


Abbildung 4.22: Advancing Front mit 3436 Knoten und 6200 Elementen

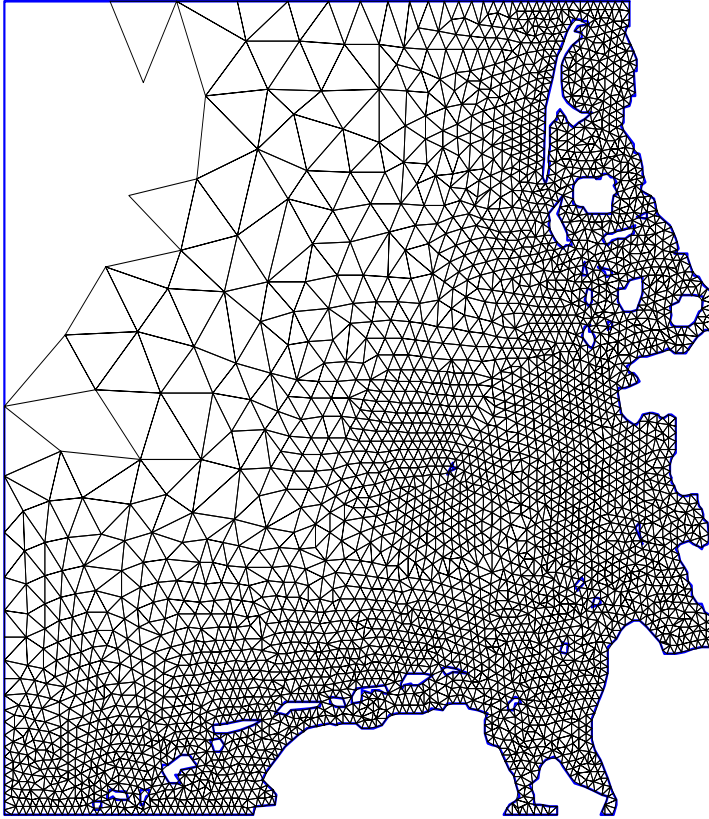


Abbildung 4.23: Advancing Front mit 3546 Knoten und 6490 Elementen

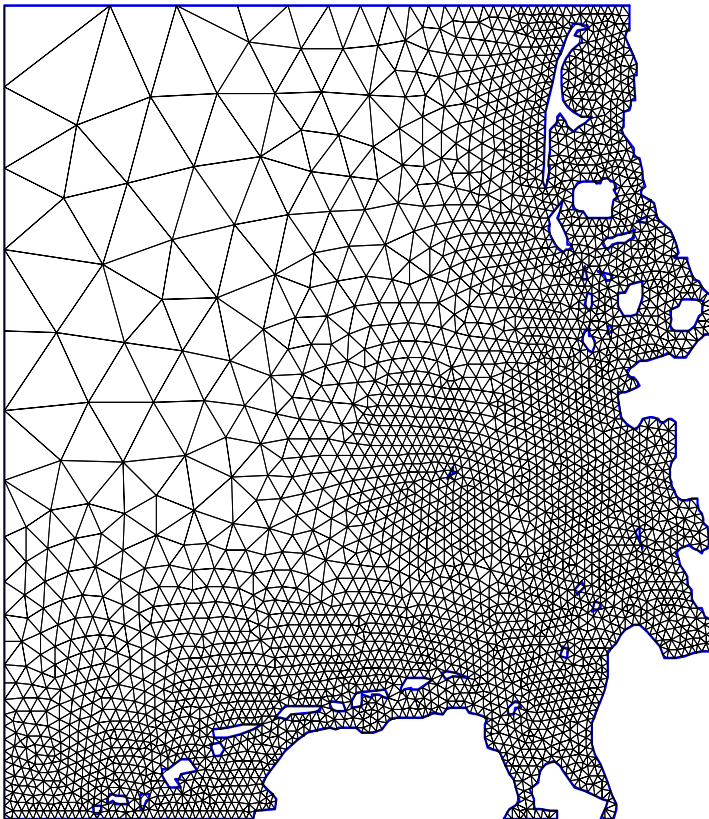


Abbildung 4.24: Vollständige Triangulierung des Nord-seeküstengebiets mit 3548 Knoten und 6509 Elementen

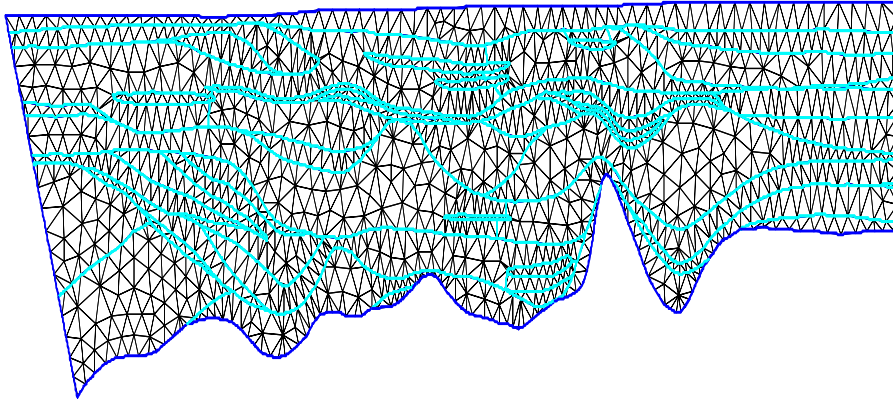


Abbildung 4.25: Triangulierung eines Salzstocks mit einer Maschenweite von $h_{konst} = 10m$

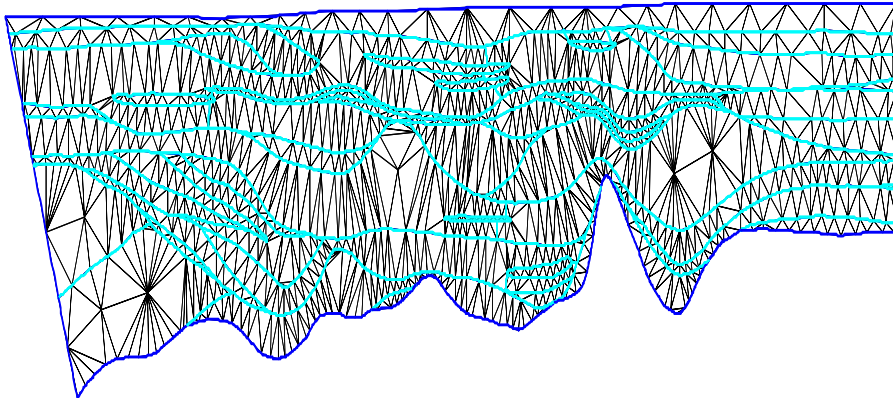


Abbildung 4.26: Triangulierung eines Salzstocks mit einer Maschenweite von $h_{konst} = 30m$

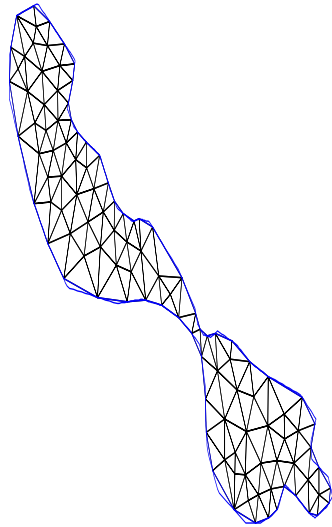


Abbildung 4.27: Grobgitter des Wolfgangsees mit 113 Knoten und 169 Elementen

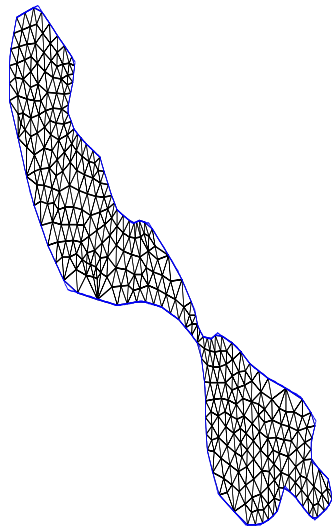


Abbildung 4.28: Grobgitter des Wolfgangsees mit 383 Knoten und 657 Elementen

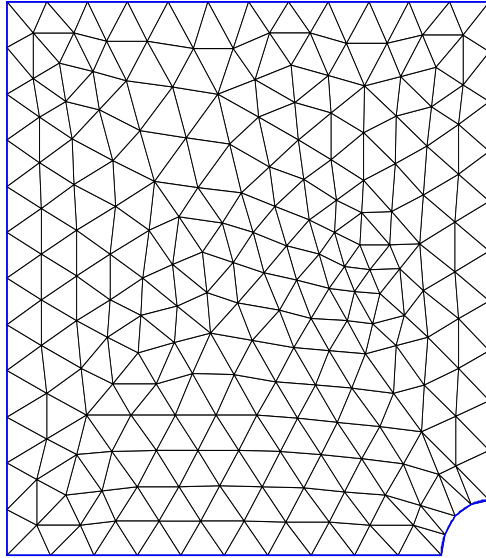


Abbildung 4.29: Grobgitter eines Prandtl-Reuß-Plastizität-Problems

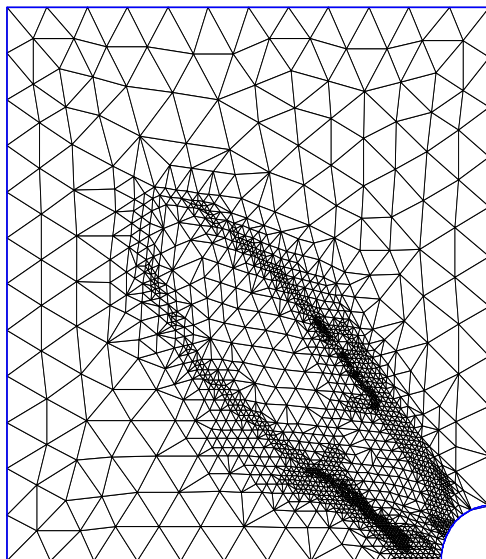


Abbildung 4.30: Adaptiv verfeinertes Gitter eines Prandtl-Reuß-Plastizität-Problems

Kapitel 5

Einführung des Modellproblems

In diesem Kapitel werden die mathematischen Grundlagen der *a priori* Fehlerabschätzungen der hier untersuchten elliptischen Randwertprobleme mit Singularitäten eingeführt und das zugrundeliegende Modellproblem erläutert.

Im ersten Abschnitt werden die verwendeten Sobolev-Räume und Normen eingeführt. Im anschließenden Abschnitt 5.2 folgt die Formulierung des elliptischen Randwertproblems, das als Dirichlet-Problem mit homogenen Randbedingungen auf einem zweidimensionalen Gebiet $\Omega \subset \mathbb{R}^2$ mit einspringender Ecke gegeben ist, sowie Regularitätseigenschaften der Lösung. In Abschnitt 5.3 werden Annahmen über die Lösung und deren Gradienten vorgestellt. Die in Abschnitt 5.4 vorgestellte Approximationstheorie gilt für stückweise lineare finite Elemente und folgt aus dem Bramble-Hilbert Lemma. In Abschnitt 5.5 wird der relative Fehler definiert und dessen Bedeutung diskutiert.

5.1 Sobolev Räume und Normen

In diesem Abschnitt wird die partielle Ableitung im \mathbb{R}^n in der Multiindex-Schreibweise eingeführt und es folgen die Definitionen der in dieser Arbeit verwendeten Sobolev-Räume $W_p^m(\Omega)$ und $\overset{\circ}{W}_p^m$ und Normen.

Eine partielle Ableitung im \mathbb{R}^n mit $\mathbf{x} \in \mathbb{R}^n$ wird durch den Ausdruck

$$D^\gamma u = \frac{\partial^{|\gamma|} u}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_n}} \quad (5.1)$$

definiert. Der Multiindex γ ist ein Vektor von nichtnegativen ganzen Zahlen, $\gamma = (\gamma_1, \dots, \gamma_n)$, $n \in \mathbb{N}$ mit dem Betrag $|\gamma| = \gamma_1 + \dots + \gamma_n$.

Die hier verwendeten Sobolev-Räume W_p^m werden auf dem Funktionenraum $L_p(\Omega)$ aufgebaut.

Definition 5.1.1

Gegeben sei ein offenes und beschränktes Gebiet $\Omega \subset \mathbb{R}^n$ und $1 \leq p \leq \infty$. Mit $L_p(\Omega)$ wird der übliche Lebesgue-Raum bezeichnet, auf dem für $1 \leq p < \infty$ die Norm

$$L_p(\Omega) := \left\{ u : \Omega \rightarrow \mathbb{R}^n : \|u\|_{L_p(\Omega)} = \left(\int_{\Omega} |u|^p \, dx \right)^{\frac{1}{p}} < \infty \right\} \quad (5.2)$$

und für $p = \infty$ die Norm

$$\|u\|_{L_{\infty}(\Omega)} = \left\{ u : \Omega \rightarrow \mathbb{R} : \|u\|_{L_{\infty}(\Omega)} = \operatorname{ess\,sup}_{x \in \Omega} |u(x)| \right\} \quad (5.3)$$

definiert ist.

Definition 5.1.2

Gegeben sei $m \geq 1$, $m \in \mathbb{N}$ und es gelte $1 \leq p \leq \infty$. Der Sobolev-Raum $W_p^m(\Omega)$ ist definiert durch:

$$W_p^m(\Omega) = \{ U : \Omega \rightarrow \mathbb{R}^n : D^{\gamma}u \in L_p(\Omega), |\gamma| \leq m \} \quad (5.4)$$

Die zugehörige Norm lautet:

$$\|u\|_{W_p^m(\Omega)} = \begin{cases} \left(\sum_{|\gamma| \leq m} \|D^{\gamma}u\|_{L_p(\Omega)}^p \right)^{\frac{1}{p}} & \text{für } 1 \leq p < \infty \\ \max_{|\gamma| \leq m} \|D^{\gamma}u\|_{L_{\infty}(\Omega)} & \text{für } p = \infty \end{cases} \quad (5.5)$$

Ferner wird der Sobolev-Raum $\overset{\circ}{W}_p^m$ eingeführt:

$$\overset{\circ}{W}_p^m(\Omega) = \{ u : u \in W_p^m(\Omega), D^{\gamma}u = 0 \text{ auf } \partial\Omega, |\gamma| \leq m - 1 \} \quad (5.6)$$

Die zugehörige Sobolev-Semi-Norm für $1 \leq p < \infty$ lautet:

$$|u|_{W_p^m(\Omega)} = \left(\sum_{|\gamma|=m} \|D^{\gamma}u\|_{L_p(\Omega)}^p \right)^{\frac{1}{p}} \quad (5.7)$$

Für $p = \infty$ ist die Sobolev-Semi-Norm gegeben durch:

$$|u|_{W_{\infty}^m(\Omega)} = \max_{|\gamma|=m} \|D^{\gamma}f\|_{L_{\infty}(\Omega)} \quad (5.8)$$

Die Länge der j -ten partiellen Ableitungen von u wird folgendermaßen definiert:

$$|\nabla^{(j)}u(x)| = \left(\sum_{|\gamma|=j} |D^{\gamma}u(x)|^2 \right)^{\frac{1}{2}} \quad (5.9)$$

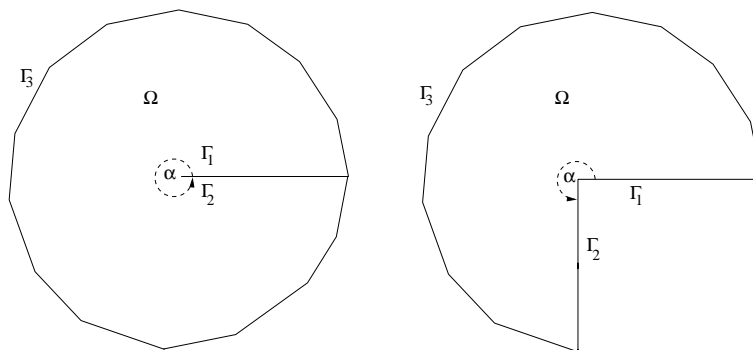


Abbildung 5.1: Beispiele für Rechengebiete Ω mit Rand $\Gamma = \Gamma_1 + \Gamma_2 + \Gamma_3$ des Dirichlet-Problems (5.11)

Für jede offene Teilmenge G von Ω ist

$$\left\| \nabla^{(j)} u(x) \right\| = \max_{u(x) \in \bar{G}} \left| \nabla^{(j)} u(x) \right|$$

die Maximumnorm der j -ten partiellen Ableitungen.

5.2 Formulierung des Randwertproblems

Es wird das für die folgenden a priori Fehlerabschätzungen zugrundeliegende Dirichlet-Randwertproblem und Eigenschaften der Lösung beschrieben.

Sei $\Omega \subset \mathbb{R}^2$ ein polygonales Gebiet mit einem inneren Winkel $\pi < \alpha \leq 2\pi$ im Ursprung des Koordinatensystems. Das Gebiet $\Omega \subset \mathbb{R}^2$ wird mit den Polarkoordinaten (r, Θ) folgendermaßen definiert:

$$\Omega = \{(r, \Theta) : 0 \leq r < 1, 0 < \Theta < \alpha\}$$

Ferner bezeichne $\partial\Omega = \Gamma_1 + \Gamma_2 + \Gamma_3$ den Rand des Gebietes Ω (siehe Abbildung 8.1). Weiter wird $\beta = \frac{\pi}{\alpha}$ gesetzt. Dabei gilt:

$$\frac{1}{2} \leq \beta < 1$$

Das betrachtete Dirichlet-Problem ist gegeben durch:

$$\begin{aligned} -\Delta u &= f & \text{in } \Omega \\ u &= g & \text{auf } \partial\Omega \end{aligned} \tag{5.10}$$

Hierbei ist f eine gegebene Funktion, die z.B. in L_2 enthalten ist. Die Regularitätseigenschaften der Lösung des Problems (5.11) werden im noch folgenden Lemma

5.13 spezifiziert. In einigen Betrachtungen wird $g = 0$ gewählt.

Durch partielle Integration folgt, daß das zugehörigen Variationsproblem in der Form

$$D(u, v) = \int_{\Omega} \nabla u \cdot \nabla v dx = \int_{\Omega} f v dx = (f, v) \quad \forall v \in \overset{\circ}{W}_2^1(\Omega) \quad (5.11)$$

dargestellt werden kann und aus dem Lax-Milgram-Theorem folgt, daß das Variationsproblem 5.11 eine eindeutig bestimmte Lösung $u \in \overset{\circ}{W}_2^1(\Omega)$ für $g = 0$ besitzt.

Mit $S_h(\Omega)$ wird der endlichdimensionale Raum der stetigen Funktionen auf $\bar{\Omega}$ bezeichnet, deren Restriktionen auf jedes Dreieck $\tau \in T(\tau)$ aus stückweise stetigen linearen Funktionen besteht. Ferner ist $\overset{\circ}{S}_h(\Omega)$ der Unterraum der Funktionen von $S_h(\Omega)$, die auf dem Rand $\partial\Omega$ verschwinden. Es gilt $S_h(\Omega) \subset W_{\infty}^1$.

Mit $u_h \in S_h$ wird die approximierte Lösung von (5.11) bezeichnet und u_h ist die eindeutige Lösung von:

$$D(u_h, \chi) = (f, \chi) \quad \text{für alle } \chi \in \overset{\circ}{S}_h(\Omega) \quad (5.12)$$

Außerdem interpoliert u_h die Funktion g auf dem Rand Γ mit $g = 0$ auf Γ_1 und Γ_2 .

Die Regularität der Lösung u wird durch das nachfolgende Lemma 5.2.1 gegeben, das in ähnlicher Form in [112], Kapitel 3, Abschätzung 12.20 zu finden ist und auf den Arbeiten von Merigot [70], [71], [72], [73] basiert.

Lemma 5.2.1 *Gegeben sei $f \in L_p$, $1 < p < \frac{2}{2-\beta}$. Dann ist die schwache Lösung von (5.11) in $W_p^2(\Omega)$ enthalten und es gilt:*

$$\|u\|_{W_p^2(\Omega)} \leq c_p \|f\|_{L_p(\Omega)} \quad (5.13)$$

Für ein konvexes Gebiet kann p als $p = 2$ in Lemma 5.2.1 gewählt werden. Bei einem nichtkonvexen Gebiet ist diese Beziehung nicht mehr erfüllt und es ergibt sich eine schwächere Regularitätsaussage. Weitere Eigenschaften der Funktion u werden im Verlauf der Arbeit an entsprechender Stelle hinzugefügt.

Es ist wohlbekannt (siehe [45], [47], [57], [61]), daß in einer Umgebung des Koordinatenursprungs die Lösung u eine Entwicklung der folgenden Form besitzt:

$$u = \sum_{j=1}^N a_j r^{j\beta} \sin j\beta\Theta + w(r, \Theta) \quad w \in W_2^2, N \in \mathbb{N} \quad (5.14)$$

Dabei sind (r, Θ) Polarkoordinaten und $w(r, \Theta)$ ist beliebig glatt bei hinreichend groß gewähltem N . Die Koeffizienten a_j in (5.14) werden als Spannungsintensitätsfaktoren bezeichnet und spielen in der Bruchmechanik eine wichtige Rolle. Aus der Vielzahl von Veröffentlichungen soll exemplarisch auf die Artikel [20], [39], [50], [87] sowie auf [49] und die dort aufgeführte Literatur verwiesen werden. Die Spannungsintensitätsfaktoren a_j werden hier durch eine Methode, die als "trigonometric fitting" bekannt ist, berechnet. Erstmals wurden die Spannungsintensitätsfaktoren nach der Methode "trigonometric fitting" in [85] und [86] vorgestellt. Hier werden die Spannungsintensitätsfaktoren in Anlehnung an [95] nach folgendem Ausdruck berechnet:

$$a_j = \frac{2}{\alpha} \int_0^\alpha g(\Theta) \sin(j\beta\Theta) d\Theta \quad (5.15)$$

Mit $j = 1, 2, \dots$ wird die j -te Singularitätenfunktion \mathcal{S}_j definiert als:

$$\mathcal{S}_j = r^{j\beta} \sin j\beta\Theta \quad (5.16)$$

Für jeden Wert des Parameters $0 < h < \frac{1}{2}$ sei $T(\tau)$ eine Triangulierung des Gebietes Ω_h , wobei Ω_h das Gebiet Ω mit einer Menge von Dreiecken $\tau \in T(\tau)$ approximiert, deren Randknoten auf Γ liegen. Weiter wird angenommen, daß die Dreiecke nicht degeneriert sind. Somit gibt es von h und τ unabhängige Konstanten $0 < m_1 < m_2$, so daß für jedes Element τ ein $h(\tau)$, das als Durchmesser des Dreiecks bezeichnet wird, existiert. Dann ist τ in einem Kreis mit Radius $m_1 h(\tau)$ und ebenfalls in einem Kreis mit Radius $m_2 h(\tau)$ enthalten.

5.3 Eigenschaften der Lösungen des Dirichlet-Problems

Im folgenden Abschnitt werden die Annahmen zum Verhalten der Lösung und ihres Gradienten zu den Beweisen in Kapitel 6 und 7 vorgestellt. Die wichtigste Annahme besteht darin, daß die erste Singularitätenfunktion dominant ist.

Folgende Annahmen zur Lösung und deren Gradienten des Randwertproblems (5.11) gelten:

Es gibt positive Konstanten c_3, c_4 und $r_0 \leq 1$, die unabhängig von u und β für $\frac{1}{2} \leq \beta < 1$ sind, sodaß für $0 \leq r \leq r_0$ und $0 \leq \Theta \leq \alpha$ die Abschätzungen

$$c_3 |a_1| r^{\beta-1} \leq |\nabla u| \quad (5.17)$$

und

$$\left| \nabla^{(2)} u \right| \leq c_4 |a_1| r^{\beta-2} \quad (5.18)$$

gelten. Diese Abschätzungen drücken aus, daß der erste Term in der Entwicklung (5.14) der dominante Term für $r \leq r_0$ ist. Beispielsweise erfüllen die Funktionen $u = a_1 \mathcal{S}_1$ folgende Ungleichungen:

$$\beta |a_1| r^{\beta-1} \leq |\nabla u| \quad (5.19)$$

$$\text{und} \quad (5.20)$$

$$\left| \nabla^{(2)} u \right| \leq 2\beta |\beta - 1| |a_1| r^{\beta-2}$$

mit $c_3 = \frac{1}{2}$ und $c_4 = 1$, wobei die Konstanten unabhängig von β für $\frac{1}{2} \leq \beta < 1$ sind.

Allgemein gilt für die j -te Singularitätenfunktion:

$$j\beta r^{j\beta-1} = \left| \nabla^{(1)} \mathcal{S}_j \right| \quad (5.21)$$

$$\text{und} \quad (5.22)$$

$$\left| \nabla^{(2)} \mathcal{S}_j \right| \leq 2j\beta |j\beta - 1| r^{j\beta-2}$$

Zur Abschätzung des punktweisen relativen Fehlers der Funktion auf einem Element τ wird angenommen, daß folgende Ungleichung für jeden Punkt $(x, y) \in \tau$ gilt:

$$c |a_1| r^\beta \leq |u(x, y)| \quad (5.23)$$

Ein Beispiel von Funktionenklassen, die die Bedingungen (5.17) und (5.18) erfüllen, stellt die wichtige Klasse harmonischer Funktionen dar. In Kapitel 8.3 wird bewiesen, daß harmonische Funktionen und weitere Funktionenklassen die Ungleichungen (5.17) und (5.18) erfüllen.

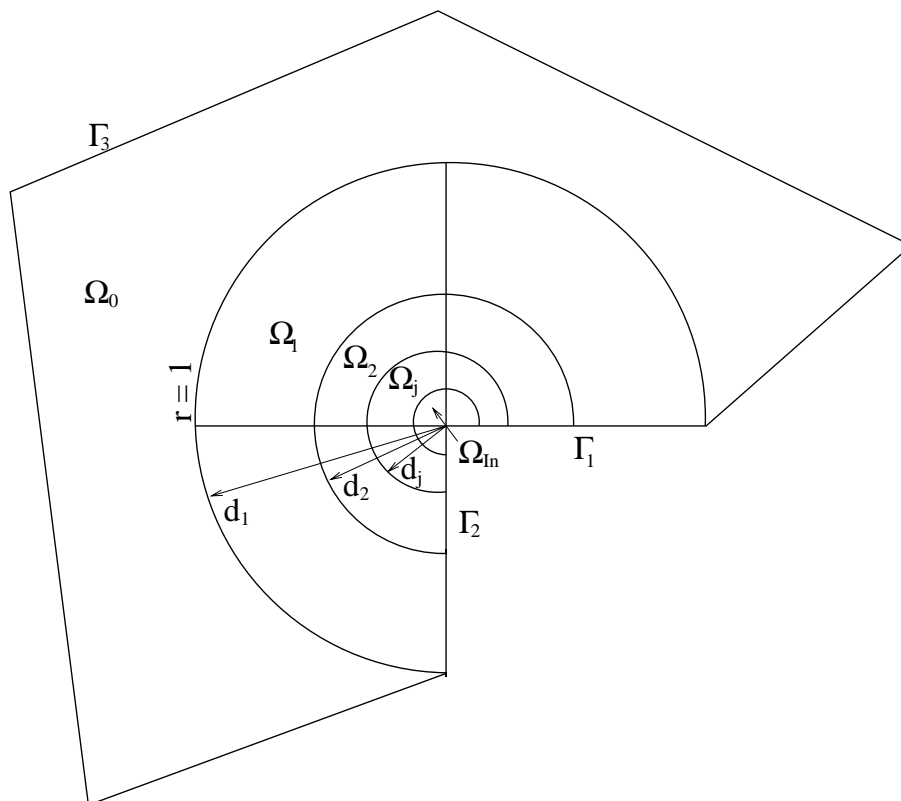


Abbildung 5.2: Rechengebiet Ω mit den Teilgebieten Ω_j zur Veranschaulichung von Definition 5.4.1

5.4 Approximationssätze und grundlegende Definitionen

Die zu den Beweisen der Theoreme 6.3.1, 7.3.1 und 7.3.2 benötigten Approximationseigenschaften stückweise linearer Funktionen werden hier in Form zweier Lemmata vorgestellt. Es folgen einige Notationen zur Charakterisierung von Teilgebieten Ω_j des Rechengebietes Ω , die ebenfalls in den oben genannten Beweisen verwendet werden. Ferner wird das hier verwendete geometrische Gitter definiert.

Aufgrund der Annahme, daß für die Lösung des Dirichlet-Problems (5.11) $u \in W_2^1(\tau)$ gilt, ist u stetig und die Interpolierende von u ist definiert und wird mit u_I bezeichnet. Die hier vorgestellten Approximationsabschätzungen basieren auf dem Bramble-Hilbert Lemma [18] und einen Skalierungsargument. Eine ausführliche Diskussion des Bramble-Hilbert Lemmas ist auch in [25] zu finden. Die hier verwendeten, nachfolgenden Lemmata sind ein Spezialfall des Theorems (4.4.4) aus [19]. Lemma 5.4.2 folgt auch aus [112], Kapitel 3, Abschätzung 12.21 mit $\frac{1}{p} + \frac{1}{q} = 1$.

Lemma 5.4.1 Gegeben sei $u \in W_p^2(\tau)$ und $1 \leq p \leq \infty$. Dann existiert für jedes Element τ mit der Maschenweite h_τ eine Konstante c , so daß gilt:

$$|u - u_I|_{W_p^1(\tau)} + h_\tau^{-1} \|u - u_I\|_{L_p(\tau)} \leq ch_\tau |u|_{W_p^2(\tau)} \quad (5.24)$$

Die Konstante c ist unabhängig von u und h .

Lemma 5.4.2 Gegeben sei u mit den gleichen Eigenschaften wie in Lemma 5.4.1. Ferner gelte $1 \leq p \leq 2$. Dann gibt es für jedes Element τ eine Interpolierende u_I derart, daß

$$|u - u_I|_{W_2^1(\tau)} \leq c_p h_\tau^{1-2\left(\frac{1}{p}-\frac{1}{2}\right)} |u|_{W_p^2(\tau)} \quad (5.25)$$

gilt. Hierbei ist die Konstante c_p unabhängig von u und h .

Zur Beschreibung und zum Beweis bekannter Fehlerabschätzungen, z. B. in der L_2 -Norm, sowie für den Beweis des Theorems zum relativen Fehler des Gradienten in der L_2 -Norm sowie der Theoreme zu den relativen Fehlern des Gradienten und der Funktion in der L_∞ -Norm werden die folgenden Bezeichnungen benötigt.

Definition 5.4.1

Gegeben sei $d_j = 2^{-j}$, $j = 1, 2, \dots, J$. Folgende Bezeichnungen werden eingeführt und in Bemerkung 5.4.2 erläutert:

$$\Omega_j = \{(x, y) \in \Omega : 2^{-j} < r < 2^{-j-1}\} \quad (5.26)$$

$$\Omega_{In} = \{(x, y) \in \Omega : r < 2^{-J}\} \quad J = \left[\ln_2 \left(\frac{1}{\rho} \right) \right] + 1 \quad (5.27)$$

$$\Omega_0 = \Omega \setminus \left(\left(\bigcup_{j=1}^J \bar{\Omega}_j \right) \cup \Omega_{In} \right)$$

$$\Omega'_0 = \Omega_0 \cup \Omega_1$$

$$\Omega'_j = \Omega_{j-1} \cup \bar{\Omega}_j \cup \Omega_{j+1}$$

$$\Omega''_j = \Omega_{j-2} \cup \bar{\Omega}_{j-1} \cup \bar{\Omega}_j \cup \bar{\Omega}_{j+1} \cup \Omega_{j+2}$$

$$\Omega_{In'} = \Omega_{In} \cup \Omega_j$$

Das hier verwendete geometrisches Gitter wird nachfolgend definiert. Zum leichteren Verständnis erfolgen zuvor einige Erläuterungen.

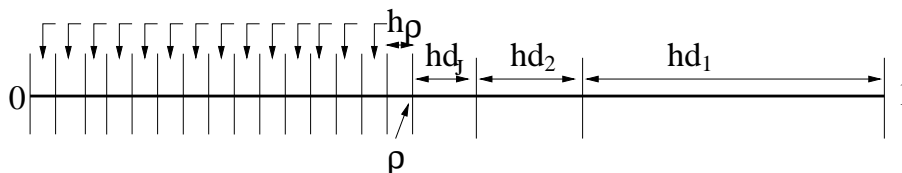


Abbildung 5.3: Graphische Darstellung der Gitterparameter h_d und ρ eines eindimensionalen geometrischen Gitters

Bemerkung 5.4.1 (zu Definition 5.4.1)

Die Maschenweite h bezeichnet die Maschenweite des Rechengitters im Abstand $d \leq 1$ vom Ursprung und h_d bezeichnet die Maschenweite des Elementes τ_d im Abstand $d > 0$ vom Ursprung. Desweiteren ist $0 < \rho < 1$ die Distanz zum Ursprung, ab der keine weitere Verfeinerung mehr vorgenommen wird und das Gitter uniform ist. Durch die Wahl von ρ wird der Index J festgelegt. Index J sagt aus, wie oft die Maschenweite h bis zum Abstand ρ halbiert werden muß und wird als ganzzahliger Anteil des folgenden Ausdrucks berechnet:

$$2^{-J} = \rho \quad \implies \quad J = \left\lceil \log_2 \frac{1}{\rho} \right\rceil + 1$$

In dieser Arbeit wird $\rho = h^\gamma$, $\gamma > 1$ gewählt und damit ergibt sich für J :

$$J = \gamma \left\lceil \log_2 \frac{1}{h} \right\rceil + 1 \quad (5.28)$$

$$J = c \ln \frac{1}{h} \quad (5.29)$$

Die Maschenweiten der Dreiecke $\tau \subset \Omega$ ergeben sich folgendermaßen:

$$\text{Falls} \quad (5.30)$$

$$\tau \subset \Omega_0, \quad \text{dann ist } h_\tau = h$$

$$\tau \subset \Omega_j, \quad \text{dann ist } h_\tau = h d_j \quad (5.31)$$

$$\tau \subset \Omega_{In}, \quad \text{dann ist } h_\tau = h \rho \quad (5.32)$$

Die Bedeutung der in dieser Bemerkung aufgeführten Größen wird in Abbildung 5.3 für den eindimensionalen Fall illustriert.

Definition 5.4.2 (Geometrisches Gitter)

Ein geometrisches Gitter ist ein Rechengitter, das von zwei Parametern h und ρ abhängt und dessen Maschenweite h_d im Abstand $0 \leq d \leq 1$ zum Ursprung bzw.

zur Singularität, nach folgendem Ausdruck berechnet wird:

$$h_d = \begin{cases} h & \text{für } d \leq 1 \\ hd & \text{für } \rho \leq d < 1 \\ h\rho & \text{für } 0 \leq d \leq \rho \end{cases} \quad (5.33)$$

5.5 Bedeutung des relativen Fehlers

In diesem Abschnitt wird eine spezielle Betrachtungsweise relativer Fehler motiviert, mit der *a priori*-Aussagen über lokale Eigenschaften der Fehlers in der L_2 - und L_∞ -Norm vorgenommen werden können.

Es gibt viele Möglichkeiten, Fehler der Lösungen oder der Gradienten der Lösungen elliptischer Randwertaufgaben zu bestimmen. Oft benutzte Fehlerabschätzungen sind z. B. die globale Energienorm, die globale L_2 -Norm oder die auf das gesamte Rechengebiet bezogene L_∞ -Norm. Alle diese Normen ergeben jedoch nur eine Aussage über das globale Verhalten des Fehlers und beziehen sich auf das gesamte Rechengebiet. Lokale Aussagen sind mit diesen Normen nur schwer zu erhalten. Die ersten Arbeiten zu L_∞ -Abschätzungen sind die Artikel von Natterer [74] und Scott [97], in denen Abschätzungen auf unregelmäßigen, quasi-uniformen Gittern in 2 Raumdimensionen der Art

$$\|u - u_h\|_{L_\infty(\Omega)} \leq ch \left(\log \frac{1}{h} \right) \inf_{\chi \in \mathcal{S}_h} \|u - \chi\|_{W_\infty^1(\Omega)}$$

für stückweise lineare finite Elemente und

$$\|u - u_h\|_{W_\infty^1(\Omega)} \leq c \inf_{\chi \in \mathcal{S}_h} \|u - \chi\|_{W_\infty^1(\Omega)} \quad (5.34)$$

für quadratische oder höherpolynomiale finite Elemente hergeleitet werden. In [83] wird auch für stückweise lineare finite Elemente auf konvexen Gebieten ein Abschätzung der Form 5.34 bewiesen. In den Arbeiten von Nitsche [75], [76] wurde die Methode der gewichteten Normen eingeführt, die in den nachfolgenden Artikeln [82] und [92] weiter entwickelt wurde.

Lokale Abschätzungen, z. B. auf einem Teilgebiet oder auf einzelnen Elementen des Rechengebietes sind bei *a priori*-Abschätzungen nur schwer zu erhalten. Gerade diese Informationen sind oft von großem praktischem Interesse. Durch eine besondere Betrachtungsweise relativer Fehler ist es möglich, lokale Abschätzungen des Fehlers der Lösung sowie des Gradienten der Lösung zu finden. Der relative Fehler kann auf verschiedene Weise definiert werden. So wird z. B. in [123] zur *a posteriori*-Fehlerschätzung eines linearen Elastizitätsproblems der relative Fehler in der Energie- bzw. L_2 -Norm bezogen auf das gesamte Rechengebiet unter der Voraussetzung, daß der Nenner ungleich Null ist, definiert als:

$$\frac{\|u - u_h\|_{L_2(\Omega)}}{\|u\|_{L_2(\Omega)}} \quad \text{und} \quad \frac{\|\nabla(u - u_h)\|_{L_2(\Omega)}}{\|\nabla u\|_{L_2(\Omega)}}$$

Weitere Arbeiten, in denen relative Fehler in der Energie- bzw. L_2 -Norm in ähnlicher Form wie in [123] verwendet werden, sind z.B. in [5], [31], [30], [9] oder [58] zu finden. In [6] werden relative Fehler zur Beschreibung von Punkten, an denen Superkonvergenz zu beobachten ist, verwendet. In den zuvor aufgeführten Arbeiten wird der relative Fehler allerdings immer in Zusammenhang mit *a posteriori*-Fehlerschätzern und adaptiven Gitterverfeinerungsstrategien betrachtet. Es gibt eine Vielzahl von Arbeiten im Umfeld der adaptiven Gittergenerierung und *a posteriori*-Fehlerschätzung. Die hier zitierten Artikel stellen nur einen kleinen Teil des Literaturspektrums vor. Deshalb wird als ergänzender und weiterführender Literaturhinweis auf die Übersicht [109] und die dort zitierte Literatur verwiesen.

Mit den in dieser Arbeit vorliegenden Erfahrungen gelingt es, einen neuen Fehlerbegriff zu definieren, der gleichmäßige *a priori*-Abschätzungen erlaubt. Der hier vorgestellte relative Fehler wird unter der Voraussetzung, daß der Nenner von Null verschieden ist, folgendermaßen definiert:

$$\left\| \frac{u - u_h}{u} \right\| \quad \text{und} \quad \left\| \frac{\nabla(u - u_h)}{|\nabla u|} \right\| \quad (5.35)$$

Die Quotienten in (5.35) sind als punktweise Division zweier Funktionen zu verstehen.

Der relative Fehler nach der Definition in (5.35) kann als Betrag des Fehlers einer gegebenen Größe pro Einheit dieser Größe interpretiert werden. Aufbauend auf (5.35) ist es möglich, verschiedene relative Fehler zu definieren. Ein Beispiel der Funktion und ihres Gradienten in der L_2 -Norm sieht folgendermaßen aus:

$$\left\| \frac{\nabla(u - u_h)}{|\nabla u|} \right\|_{L_2(\Omega_0)} \quad \text{oder} \quad \left\| \frac{u - u_h}{u} \right\|_{L_2(\Omega_0)} \quad (5.36)$$

Eine weitere Betrachtungsweise stellt der folgende Ausdruck dar:

$$\left\| \frac{\nabla(u - u_h)}{|\nabla u|} \right\|_{L_\infty(\Omega_0)} \quad \text{oder} \quad \left\| \frac{u - u_h}{u} \right\|_{L_\infty(\Omega_0)} \quad (5.37)$$

Ω_0 bezeichnet in (5.36) und (5.37) ein beliebiges Teilgebiet des Rechengebietes Ω . Die in (5.36) und (5.37) eingeführten relativen Fehler werden hier zur Untersuchung elliptischer Randwertaufgaben auf Gebieten mit einspringenden Ecken benutzt, bei denen die Ableitungen der Lösungen in einer Umgebung einer Singularität unbegrenzt anwachsen. Das Ziel ist, die in (5.36) und (5.37) definierten relativen Fehler abzuschätzen und Rechengitter zu finden, auf denen sie von optimaler Ordnung bei Verwendung stückweise linearer Funktionen in der finiten

Elemente Methode sind. Die Konstruktion des Gitters wird durch Überlegungen aus der Approximationstheorie geleitet und geht direkt in die Konvergenzbeweise des Kapitels 7 ein.

Kapitel 6

Eine neue *a priori* Abschätzung des relativen Fehlers des Gradienten in der L_2 -Norm

Dieses Kapitel ist in 3 Abschnitte unterteilt. Im ersten Abschnitt wird eine klassische Fehlerabschätzung in der Energie- bzw. L_2 -Norm des singular gestörten Randwertproblems (5.11) gegeben und gezeigt, daß mit uniformen Gittern der Fehler der Funktion die nichtoptimale Konvergenzordnung $O(h^{2\beta-\varepsilon})$ und der Fehler des zugehörigen Gradienten die nichtoptimale Konvergenzordnung $O(h^{\beta-\varepsilon})$ besitzt. Eine Diskussion der Arbeit von Babuška und Rosenzweig [4] (Abschnitt 6.2) zeigt, wie durch lokale Verfeinerung des Gitters fast optimale Konvergenzordnung des Fehlers der Funktion und ihres Gradienten erreicht werden kann. Durch eine neue Betrachtungsweise des relativen Fehlers des Gradienten in Kapitel 6.3 und der Voraussetzung, daß die erste Singularitätenfunktion das Verhalten der Lösung bestimmt, wird bewiesen, daß der relative Fehler des Gradienten auf uniform verfeinerten Gittern fast optimale Konvergenz besitzt.

6.1 Klassische Fehlerabschätzung in der Energienorm

In diesem Abschnitt wird gezeigt, daß auf einem Gebiet mit einspringender Ecke mit uniform verfeinerten Gittern der Fehler der Funktion eine Konvergenzordnung von $O(h^{2\beta-\varepsilon})$ und der Gradient eine Konvergenzordnung von $O(h^{\beta-\varepsilon})$ in der L_2 -Norm aufweist.

Dazu sind ein paar Vorbemerkungen notwendig.

Vorbemerkungen zu Lemma 6.1.1:

Sei $f \in L_2(\Omega)$ mit Ω aus Kapitel 5.2 und seien $u \in \overset{\circ}{W}_2^1$, $\varepsilon > 0$ und $u_h \in \overset{\circ}{S}_h$,

die Lösungen der Gleichungen

$$D(u, v) = (f, v) \quad \forall v \in \overset{\circ}{W}_2^1(\Omega)$$

und

$$D(u_h, \Phi) = (f, \Phi) \quad \forall \Phi \in \overset{\circ}{S}_h(\Omega)$$

erfüllen. Für den Fehler $e = u - u_h$ folgt wegen der Galerkinorthogonalität:

$$D(e, \Phi) = 0 \quad \text{für alle } \Phi \in \overset{\circ}{S}_h(\Omega) \quad (6.1)$$

Lemma 6.1.1 *Gegeben sei $\varepsilon > 0$ und $p = \frac{2}{2-\beta+\varepsilon}$. Dann gibt es eine von h, u und u_h unabhängige Konstante $c(\varepsilon)$, so daß gilt:*

$$\|\nabla e\|_{L_2(\Omega)} \leq ch^{\beta-\varepsilon} |u|_{W_p^2(\Omega)} \quad (6.2)$$

Weiter gilt:

$$\|e\|_{L_2(\Omega)} \leq ch^{2\beta-\varepsilon} |u|_{W_p^2(\Omega)} \quad (6.3)$$

Beweis:

Aus der Definition des Fehlers e , verwenden der Schwarz'schen Ungleichung und aus (6.1) folgt:

$$\begin{aligned} \|\nabla e\|_{L_2(\Omega)}^2 &= D(e, e) = D(e, u - u_h) = D(e, u - u_I) \\ &\leq \|\nabla e\|_{L_2(\Omega)} \|\nabla(u - u_I)\|_{L_2(\Omega)} \end{aligned}$$

Dabei ist u_I die Interpolierende von u . Unter Verwendung von Standardaussagen der Approximationstheorie für die Interpolierende (Lemma 5.4.2) folgt für ein beliebiges $\varepsilon > 0$:

$$\|\nabla e\|_{L_2(\Omega)} \leq c_\varepsilon h^{\beta-\varepsilon} \|u\|_{W_p^2(\Omega)} \quad \text{für } p = \frac{2}{2-\beta+\varepsilon} \quad (6.4)$$

Damit ist Ungleichung (6.2) bewiesen.

Die Abschätzung von $\|e\|_{L_2(\Omega)}$ erfolgt mit dem Dualitätsargument von Aubin-Nitsche.

Gegeben sei z , das die Bedingung

$$D(v, z) = (e, v) \quad \text{für alle } v \in \overset{\circ}{W}_2^1(\Omega) \quad (6.5)$$

erfüllt. Aus (6.5), (6.4), (6.1) und der Schwarz'schen Ungleichung folgt:

$$\|e\|_{L_2(\Omega)}^2 = (e, e) = D(e, z - z_I) \leq \|\nabla e\|_{L_2(\Omega)} \|\nabla(z - z_I)\|_{L_2(\Omega)} \quad (6.6)$$

Mit 5.2.1 ergibt sich:

$$\|z\|_{W_p^2(\Omega)} \leq c_p \|e\|_{L_p(\Omega)} \quad \text{für alle } 1 < p < \frac{2}{2-\beta}$$

Unter Verwendung obiger Aussage und den Lemmata 5.4.1 und 5.4.2 für die Interpolierende z_I und mit $p = \frac{2}{2-\beta+\varepsilon}$ folgt:

$$\begin{aligned} \|e\|_{L_2(\Omega)}^2 &\leq ch^{\beta-\varepsilon} \|\nabla e\|_{L_2(\Omega)} \|z\|_{W_p^2(\Omega)} \leq ch^{\beta-\varepsilon} \|\nabla e\|_{L_2(\Omega)} \|e\|_{L_p(\Omega)} \\ &\leq ch^{\beta-\varepsilon} \|\nabla e\|_{L_2(\Omega)} \|e\|_{L_2(\Omega)} \end{aligned}$$

Aus (6.4) folgt weiter, daß

$$\|e\|_{L_2(\Omega)} \leq ch^{2(\beta-\varepsilon)} \|u\|_{W_p^2(\Omega)} \quad (6.7)$$

und mit Ungleichung (6.7) ist der Beweis von Lemma 6.1.1 abgeschlossen. \square

Die Abschätzung der Fehler für die Funktion und des Gradienten in der H^1 -Norm bzw. in der L_2 -Norm zeigt somit keine optimale Ordnung.

6.2 Standardabschätzung des Fehlers für verfeinerte Gitter

Anhand der Arbeit von Babuška und Rosenzweig [4] wird gezeigt, wie durch lokale Verfeinerung des Gitters der Fehler der Lösung und deren Gradienten eines elliptischen Randwertproblems mit einer einspringenden Ecke optimale Konvergenzordnung in $W_2^1(\Omega)$ erzielt werden kann.

Babuška und Rosenzweig betrachten ein kreisförmiges Teilgebiet Ω_j des Rechengebietes Ω mit $d_{j-1} \leq r \leq d_j$, auf dem die Maschenweite mit h_{d_j} bezeichnet wird. Weiter wird in [4] angenommen, daß für u die Ungleichung

$$|\nabla^{(2)} u| \leq cr^{\beta-2}$$

gilt. Die Wahl des Gitters folgt aus einer aus der Approximationstheorie motivierten Überlegung. Aus Lemma 5.4.1, der Berechnung der Sobolev-Seminorm (5.7) mit Integrationsgrenzen d_j und d_{j-1} , wobei $d_j = 2^{-j}$ gilt, sowie geeigneten Abschätzungen (vergl. Abbildung 5.2) ergibt sich für den Interpolationsfehler des Gradienten

$$\begin{aligned} \|\nabla(u - u_I)\|_{L_2(\Omega_j)} &\leq ch_{d_j}|u|_{W_2^2(\Omega_j)} \\ &\leq ch_{d_j}d_j^{\beta-1} \end{aligned}$$

Der Fehler im Abstand d_j zur Singularität ist dann von optimaler Ordnung, wenn

$$h_{d_j}d_j^{\beta-1} = h \quad \text{oder} \quad h_{d_j} = hd_j^{1-\beta}$$

gewählt wird. Im letzten Teilgebiet mit der Maschenweite h_{I_n} wird h_{I_n} folgendermaßen gewählt:

$$h_{I_n}^\beta = h \quad \text{oder} \quad h_{I_n} = h^{\frac{1}{\beta}}$$

Damit ergibt sich mit $p = \frac{2}{2-\beta+\varepsilon}$:

$$\|\nabla(u - u_I)\|_{L_2(\Omega_m)} \leq ch_{I_n}^{\beta-\varepsilon}|u|_{W_p^2(\Omega)}$$

Unter der Voraussetzung, daß die Graduierung des Gitters in der oben aufgeführten Art vorgenommen wird, beweisen Babuška und Rosenzweig in [4], daß für jedes gegebene $\varepsilon > 0$

$$\|\nabla(u - u_h)\|_{L_2(\Omega)} \leq c_\varepsilon h^{1-\varepsilon}$$

gilt. Weiterhin folgt nach [4] aus einem Aubin-Nitsche Dualitätsargument:

$$\|u - u_h\|_{L_2(\Omega)} \leq c_\varepsilon h^{2-\varepsilon}$$

Babuška und Rosenzweig zeigen in [4], daß mit der vorgegebenen Verfeinerungsregel $h_{d_j} = hd_j^{1-\beta}$ für den Fehler des Gradienten und der Funktion eine fast optimale Fehlerordnung erzielt werden kann.

Als nächstes wird ein spezieller Fall einer lokalen Abschätzung in der Energienorm in Lemma 6.2.1 vorgestellt. Lemma 6.2.1 wird im noch folgenden Beweis von Theorem 6.3.1 benötigt. Der Beweis von Lemma 6.2.1 ist in der Arbeit von Schatz und Wahlbin [93] (Lemma 3.1) zu finden.

Lemma 6.2.1 ([93]) *Gegeben sei ein beliebiges $j = 0, 1, \dots$ und es werde angenommen, daß ein quasiuniformes Gitter mit der lokalen Maschenweite h auf Ω_j vorliege. Dann gilt:*

$$\begin{aligned} \|\nabla e\|_{L_2(\Omega_j)} \leq c & \left(\|\nabla(u - u_I)\|_{L_2(\Omega'_j)} \right. \\ & \left. + d_j^{-1} \|u - u_I\|_{L_2(\Omega'_j)} + d_j^{-1} \|u - u_h\|_{L_2(\Omega'_j)} \right) \end{aligned} \quad (6.8)$$

6.3 Formulierung und Beweis einer neuen *a priori* Abschätzung des relativen Fehlers des Gradienten in der L_2 -Norm

In diesem Abschnitt wird eine neue Betrachtungsweise des relativen Fehlers des Gradienten vorgestellt und gezeigt daß in der L_2 -Norm eine lokale *a priori* Abschätzung mit fast optimaler Konvergenzordnung auf uniform verfeinerten Gitter möglich ist. Voraussetzung hierzu ist, daß die erste Singularitätenfunktion das Verhalten der Lösung bestimmt. Zum Beweis optimaler Konvergenzordnung genügt es, nur die Quasiuniformität des Gitters mit der Maschenweite h zu fordern.

Theorem 6.3.1 *Gegeben sei eine quasiuniformes Gitter mit der Maschenweite h . Außerdem sei angenommen, daß (5.17) und (5.18) gelte. Dann existiert für ein beliebiges $\varepsilon > 0$ eine Konstante c_ε , so daß*

$$\left\| \frac{\nabla(u - u_h)}{\nabla u} \right\|_{L_2(\Omega)} \leq c_\varepsilon h^{1-\varepsilon} \quad \text{für alle} \quad \frac{1}{2} \leq \beta < 1$$

gilt.

Beweis:

Es wird nochmal hervorgehoben, daß aus den Annahmen (5.17) und (5.18) folgt, daß die erste Singularitätenfunktion das Verhalten der Lösung in jedem Punkt des Rechengebietes Ω bestimmt. Außerdem gilt i.a., falls $a_1 \neq 0$ ist, daß ein solches Verhalten zumindest in einer Umgebung um den Ursprung erwartet werden kann.

Die L_2 -Norm des relativen Fehlers wird durch die gewichtete L_2 -Norm unter Verwendung von (5.17)

$$\left\| \frac{\nabla e}{\nabla u} \right\|_{L_2(\Omega)} \leq c \frac{1}{|a_1|} \left\| r^{1-\beta} \nabla e \right\|_{L_2(\Omega)} \quad (6.9)$$

abgeschätzt. Die Anwendung der Dreiecksungleichung führt auf:

$$\left\| r^{1-\beta} \nabla e \right\|_{L_2(\Omega)} \leq \left\| r^{1-\beta} \nabla e \right\|_{L_2(\Omega_{In})} + \sum_{j=0}^J \left\| r^{1-\beta} \nabla e \right\|_{L_2(\Omega_j)} \quad (6.10)$$

Für den ersten Ausdruck auf der rechten Seite von (6.10) folgt aufgrund der Beziehung

$$r \leq ch \quad \text{in } \Omega_{In}$$

der Ausdruck

$$\left\| r^{1-\beta} \nabla e \right\|_{L_2(\Omega_{In})} \leq ch^{1-\beta} \|\nabla e\|_{L_2(\Omega_{In})} \leq ch^{1-\beta} \|\nabla e\|_{L_2(\Omega)}$$

Weiter folgt mit Lemma 6.1.1, Ungleichung (6.4):

$$\left\| r^{1-\beta} \nabla e \right\|_{L_2(\Omega_{In})} \leq c_\varepsilon h^{1-\varepsilon} \|u\|_{W_p^2(\Omega)} \quad \text{für } p = \frac{2}{2-\beta+\varepsilon} \quad (6.11)$$

Mittels einer direkten Rechnung unter Verwendung von (5.18) und wegen $(\beta-2)p > -2$ ergibt sich:

$$\|u\|_{W_p^2(\Omega)} \leq |a_1| c_4 \left(\int_{\Omega} r^{(\beta-2)p} dx dy \right)^{\frac{1}{p}} < c_\varepsilon |a_1| \quad (6.12)$$

Die Kombination von (6.11) und (6.12) ergibt:

$$\left\| r^{1-\beta} \nabla e \right\|_{L_2(\Omega_{In})} \leq c_\varepsilon |a_1| h^{1-\varepsilon} \quad (6.13)$$

Für jedes $j = 0, \dots, J$ ergibt sich unter Verwendung von Lemma 6.2.1 und $d_j = 2^{-j}$ (vergleiche Definition 5.4.1 in Kapitel 5, Abschnitt 5.4):

$$\begin{aligned} \|\nabla e\|_{L_2(\Omega_j)} &\leq c \left(\|\nabla(u - u_I)\|_{L_2(\Omega'_j)} \right. \\ &\quad \left. + d_j^{-1} \|u - u_I\|_{L_2(\Omega'_j)} + d_j^{-1} \|e\|_{L_2(\Omega'_j)} \right) \\ &\leq c \left(\|\nabla(u - u_I)\|_{L_2(\Omega'_j)} + d_j^{-1} \|u - u_h\|_{L_2(\Omega'_j)} \right. \\ &\quad \left. + d_j^{-1+\beta} \|e\|_{L_q(\Omega'_j)} \right) \end{aligned} \quad (6.14)$$

Der Ausdruck

$$d_j^{-1} \|e\|_{L_2(\Omega'_j)} \quad (6.15)$$

wird durch den folgenden Zwischenschritt in die Form

$$d_j^{-1+\beta} \|e\|_{L_q(\Omega'_j)}$$

übergeführt. Es gilt $q = \frac{2}{1-\beta}$. Die Anwendung der Hölder'schen Ungleichung auf Gleichung (6.15) mit $p' = \frac{1}{1-\beta}$ und $q' = \frac{1}{\beta}$ ergibt die folgende Abschätzung:

$$\begin{aligned} \|e\|_{L_2(\Omega'_j)} &= \left(\int_{\Omega'_j} (e)^2 \, dx \, dy \right)^{\frac{1}{2}} \leq \left(\int_{\Omega'_j} dx \, dy \right)^{\frac{\beta}{2}} \left(\int_{\Omega'_j} (e)^{\frac{2}{1-\beta}} \, dx \, dy \right)^{\frac{1-\beta}{2}} \\ &\leq c d_j^\beta \|e\|_{L_q(\Omega'_j)} \end{aligned}$$

Einsetzen in Gleichung (6.15) bzw. (6.14) ergibt das gewünschte Ergebnis. Schließlich folgt unter Verwendung von (6.14):

$$\begin{aligned} \left\| r^{1-\beta} \nabla e \right\|_{L_2(\Omega_j)} &\leq c d_j^{1-\beta} \|\nabla e\|_{L_2(\Omega_j)} \\ &\leq c \left[d_j^{1-\beta} \left(\|\nabla(u - u_I)\|_{L_2(\Omega'_j)} \right. \right. \\ &\quad \left. \left. + d_j^{-1} \|u - u_I\|_{L_2(\Omega'_j)} + \|e\|_{L_q(\Omega'_j)} \right) \right] \end{aligned} \quad (6.16)$$

Im nächsten Schritt werden die ersten zwei Ausdrücke von (6.16) abgeschätzt. Unter Verwendung von Lemma 5.4.1 für die Interpolierende und wegen $d_j \geq ch$ ergibt sich:

$$\|\nabla(u - u_I)\|_{L_2(\Omega'_j)} + d_j^{-1} \|u - u_I\|_{L_2(\Omega'_j)} \leq ch |u|_{W_2^2(\Omega'_j)} \quad (6.17)$$

Eine direkte Rechnung unter Beachtung von (5.18) führt auf:

$$|u|_{W_2^2(\Omega'_j)} \leq c |a_1| \left(\int_{\frac{1}{8}d_j}^{8d_j} r^{2(\beta-2)} \, dx \, dy \right)^{\frac{1}{2}} \leq c d_j^{\beta-1} |a_1|$$

Dieses Ergebnis wird in (6.17) eingesetzt und es ergibt sich für jedes $j = 0, 1, \dots, J$

$$d_j^{1-\beta} \left(\|\nabla(u - u_I)\|_{L_2(\Omega'_j)} + d_j^{-1} \|u - u_I\|_{L_2(\Omega'_j)} \right) \leq c |a_1| h \quad (6.18)$$

Zur Abschätzung des Ausdrucks $\|e\|_{L_q(\Omega'_j)}$ wird das folgende Lemma benötigt:

Lemma 6.3.1 Sei $\frac{2}{\beta} < s < \infty$. Dann gibt es eine Konstante c_s , die unabhängig von β ist, so daß gilt:

$$\|e\|_{L_s(\Omega)} \leq c_s h^{\beta - \varepsilon + \frac{2}{s}} \quad (6.19)$$

Beweis:

Für $\frac{1}{s} + \frac{1}{s'} = 1$ ist:

$$\|e\|_{L_s(\Omega)} = \sup_{\|z\|_{L_{s'}(\Omega)}=1} (e, z)$$

Für $z \in L_{s'}$ erfülle Ψ die Gleichung:

$$\begin{aligned} -\Delta \Psi &= z && \text{auf } \Omega \\ \Psi &= 0 && \text{auf } \partial\Omega \end{aligned}$$

Daraus folgt:

$$\begin{aligned} (e, z) &= (e, -\Delta \Psi) = (\nabla e, \nabla \Psi) = (\nabla e, \nabla (\Psi - \Psi_I)) \\ &\leq \|\nabla e\|_{L_2(\Omega)} \|\nabla (\Psi - \Psi_I)\|_{L_2(\Omega)} \end{aligned} \quad (6.20)$$

Es ist $p = \frac{2}{2-\beta+\varepsilon}$ und $z \in L_{s'}(\Omega)$ mit $1 < s' < \frac{2}{2-\beta}$ und $\frac{1}{s'} = 1 - \frac{1}{s}$, so daß mit Lemma 5.4.2 folgt:

$$\|\nabla e\|_{L_2(\Omega)} \|\nabla (\Psi - \Psi_I)\|_{L_2(\Omega)} \leq c(\varepsilon, s) h^{\beta-\varepsilon+\frac{2}{s}} \|u\|_{W_p^2(\Omega)} \|\Psi\|_{W_{s'}^2(\Omega)}$$

Damit ergibt sich:

$$\|e\|_{L_s(\Omega)} \leq c_3 h^{\beta-\varepsilon+\frac{2}{s}} |u|_{W_p^2(\Omega)} \quad (6.21)$$

Damit ist Lemma 6.3.1 bewiesen. \square

Nachfolgend wird die Abschätzung der rechten Seite von (6.16) weitergeführt. Es gilt $q \leq s$ und anstelle von $q = \frac{2}{1-\beta}$ wird s in der folgenden Form verwendet:

$$s = \begin{cases} \frac{2}{1-\beta}, & \text{falls } \frac{1}{2} + \varepsilon \leq \beta < 1 \\ \frac{2}{1-\beta-\varepsilon}, & \text{falls } \frac{1}{2} \leq \beta < \frac{1}{2} + \varepsilon \end{cases}$$

Weiter ergibt sich für s' der Ausdruck $1 < s' < \frac{2}{2-\beta}$. Mit (6.21) und (6.12) und $p = \frac{2}{2-\beta+\varepsilon}$ folgt:

$$\|e\|_{L_q(\Omega)} \leq \|e\|_{L_s(\Omega)} \leq c_\varepsilon h^{\beta-\varepsilon} h^{1-\beta-\varepsilon} |u|_{W_p^2} \quad (6.22)$$

$$\leq c(\varepsilon) h^{1-2\varepsilon} |a_1| \quad (6.23)$$

Das Einsetzen von (6.23) und (6.18) in (6.16) führt auf:

$$\left\| r^{1-\beta} \nabla u \right\|_{L_2(\Omega_j)} \leq c_\varepsilon (h + h^{1-2\varepsilon}) |a_1|$$

Summieren über $j = 0, 1, \dots, J$ unter Beachtung von $J = c \ln \frac{1}{h}$ (Gleichung (5.29), Seite 88) ergibt:

$$\sum_{j=0}^J \left\| r^{1-\beta} u \right\|_{L_2(\Omega'_j)} \leq c_\varepsilon h^{1-\varepsilon} |a_1| \quad (6.24)$$

Durch Einfügen von (6.24) und (6.13) in (6.10) wird der Beweis von Theorem 6.3.1 abgeschlossen. □

Kapitel 7

Eine neue *a priori* Abschätzung des relativen Fehlers in der L_∞ -Norm

Dieses Kapitel enthält eine neue punktweise Abschätzung des relativen Fehlers der Funktion und deren Gradienten in der L_∞ -Norm des in Kapitel 5 eingeführten elliptischen Randwertproblems. In Abschnitt 7.1 werden die zum Verständnis der Theoreme 7.3.1 und 7.3.2 notwendigen bekannten Abschätzungen der Fehler in der L_∞ -Norm dargestellt. Daran anknüpfend folgen in Abschnitt 7.2 die im Beweis der Theoreme 7.3.1 und 7.3.2 verwendeten Hilfssätze, die in dieser Darstellung in ähnlicher Form in [93] zu finden sind. In Abschnitt 7.3 werden die neuen Theoreme formuliert und ihre Bedeutung diskutiert. Anschließend folgen die Beweise in Abschnitt 7.4.

Es gelten analog zum Beweis von Theorem 6.3.1 die Abschätzungen (5.17) und (5.18).

7.1 Bekannte Abschätzungen in der L_∞ -Norm

In diesem Abschnitt werden zur besseren Einordnung der in Abschnitt 7.3 vorgestellten Theoreme bekannte Fehlerabschätzungen in der L_∞ -Norm diskutiert.

Aus den bekannten Abschätzungen der Fehler für den Gradienten und für die Funktion in der L_∞ -Norm ist ersichtlich, daß für Probleme mit potenzabhängigen Singularitäten ohne lokale Gitterverfeinerung der Fehler in einer Umgebung der Singularität keine optimale Ordnung aufweist. Mit Verfeinerungsregeln, die vom Abstand zur Singularität abhängen, kann eine fast optimale Genauigkeit für finite Elemente mit stückweise linearen Ansatzfunktionen erzielt werden.

Als erstes wird der Fehler auf einem unverfeinertem Gitter betrachtet. Aus einer Arbeit von Schatz und Wahlbin [93] folgt für ein polygonales Gebiet der Fehler für die Funktion

$$|(u - u_h)| \leq c \begin{cases} h^{2\beta-\varepsilon} d^{-\beta} & \text{falls } ch \leq d \\ h^{\beta-\varepsilon} & \text{falls } d \leq ch \end{cases} \quad (7.1)$$

und es ist klar, daß für diesen Fall kein optimaler Fehler erhalten werden kann.

Der Fehler zeigt ein verbessertes Verhalten, wenn ein verfeinertes Gitter verwendet wird. In einer Arbeit von Schatz und Wahlbin [94] wird gezeigt, daß mit einer Verfeinerung nach der Regel

$$\begin{aligned} h_{d_j} &= h d_j^{1-\frac{\beta}{2}} & \text{für } d_j \geq ch^{\frac{2}{\beta}} \\ h_{I_n} &= h^{\frac{2}{\beta}} & \text{für } d \leq ch^{\frac{2}{\beta}} \end{aligned}$$

fast optimale Konvergenz erzielt werden kann. Es ergibt sich für den Fehler der Funktion:

$$\|u - u_h\|_{L_\infty(\Omega)} \leq ch^{2-\varepsilon}$$

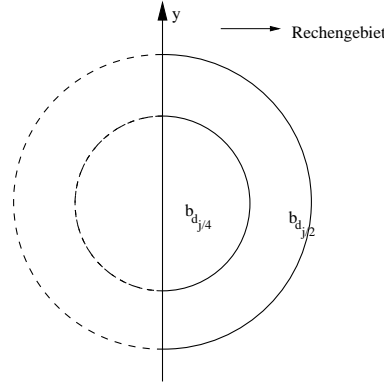
7.2 Vorbereitende Lemmata

In diesem Abschnitt werden die in den Beweisen der Theoreme 7.3.1 und 7.3.2 notwendigen Hilfssätze vorgestellt. Lemma 7.2 und Lemma 7.4 werden aus der Arbeit von Schatz und Wahlbin [93] abgeleitet und bewiesen.

Lemma 7.2.1 *Gegeben sei $\hat{\Omega}_j = \Omega_{I_n} \cup_{k=j}^J \bar{\Omega}_k$. Außerdem sei auf Ω_j das Gitter quasiuniform mit der Maschenweite hd_j und auf Ω_{I_n} quasiuniform mit der Maschenweite $h\varrho$. Dann gilt:*

$$\begin{aligned} \|\nabla e\|_{L_2(\hat{\Omega}_j)} &\leq c \left(\|\nabla(u - u_I)\|_{L_2(\hat{\Omega}_{j-1})} \right. \\ &\quad \left. + d_j^{-1} \|u - u_I\|_{L_2(\hat{\Omega}_{j-1})} + d_j^{-1} \|u - u_h\|_{L_2(\hat{\Omega}_j)} \right) \end{aligned} \quad (7.2)$$

Der Beweis von Lemma 7.2 folgt analog zu dem Beweis in [93] unter Berücksichtigung des Abstandes d_j zwischen den Teilgebieten $\hat{\Omega}_j$ und $\hat{\Omega}_{j-1}$. Zur Abschätzung der Normen $\|u - u_h\|_{L_\infty(\tau)}$ und $\|\nabla(u - u_h)\|_{L_\infty(\tau)}$ für $\tau \subset \Omega_j$ werden lokale *a priori* Abschätzungen auf den kreisförmigen Gebieten Ω_j benötigt. Zu Lemma 7.2.2 wird nur eine Beweisskizze gegeben. Die Formulierung und die Beweise dieses Lemmas sind an den Ergebnissen in den Arbeiten von [93] orientiert.

Abbildung 7.1: Fortsetzung von u und u_h am Rand des Gebietes Ω

Lemma 7.2.2 Gegeben sei $\Omega_j = \{(x,y) \in \bar{\Omega}_j : 2^{-j} < r < 2^{-j-1}\}$ und $\Omega'_j = \Omega_{j-1} \cup \bar{\Omega}_j \cup \Omega_{j+1}$ und ein quasiuniformes Gitter mit der lokalen Maschenweite h auf Ω'_j . Ferner gelte, daß u und u_h die Bedingungen (5.11) und (5.12) erfülle. Dann gibt es eine Konstante c , unabhängig von h , u , u_h und j , so daß

$$\|e\|_{L^\infty(\Omega_j)} \leq c \left(\ln \left(\frac{1}{h} \right) \|u - u_I\|_{L^\infty(\Omega'_j)} + d_j^{-1} \|e\|_{L_2(\Omega'_j)} \right) \quad (7.3)$$

und desweiteren

$$\begin{aligned} \|\nabla e\|_{L^\infty(\Omega_j)} \leq c \left(\|\nabla(u - u_I)\|_{L^\infty(\Omega'_j)} \right. \\ \left. + d_j^{-1} \|u - u_I\|_{L^\infty(\Omega'_j)} + d_j^{-2} \|e\|_{L_2(\Omega'_j)} \right) \end{aligned} \quad (7.4)$$

gilt.

Beweis:

Der Beweis von Lemma 7.2.2 erfolgt nach den Ideen in [93].

Das Gebiet Ω_j kann mit einer festgelegten endlichen Anzahl von Kugeln $B_{\frac{d_j}{4}}$ mit Radius $\frac{d_j}{4}$, deren Mittelpunkte in den Punkten $P_i, i = 1, \dots, n$ liegen, überdeckt werden. Dabei liegen die konzentrischen Kugeln $B_{\frac{d_j}{2}}(P_i)$ vollständig in Ω'_j bzw. die zugehörigen auf dem Rand liegenden Halbkugeln $b_{\frac{d_j}{4}}(P_k) k = 1, \dots, M$ sind so angeordnet, daß deren konzentrische Halbkugeln $b_{\frac{d_j}{2}}(P_k)$ ebenfalls vollständig in Ω'_j liegen. Damit gezeigt werden kann, daß die Ungleichungen (7.3) und (7.4) gelten, muß gezeigt werden, daß für jedes innere Gebiet $B_{\frac{d_j}{4}}(P_i) = B_{\frac{d_j}{4}}$ bzw. für jede Halbkugel auf dem Rand $b_{\frac{d_j}{4}}(P_k) = b_{\frac{d_j}{4}}$ gilt. Durch Ersetzen der inneren Gebiete Ω_j durch $B_{\frac{d_j}{4}}$ und Ω'_j durch $B_{\frac{d_j}{2}}$ in (7.3) und (7.4) wird das Problem bis auf

die Randhalbkugel auf die bekannte Lösung in [93] zurückgeführt. Die noch verbleibende Abschätzung für die Halbkugeln mit $b_{\frac{d_j}{4}}$ am Rande des Gebietes erfolgt durch das Ersetzen von Ω_j durch $b_{\frac{d_j}{4}}$ bzw. von Ω'_j durch $b_{\frac{d_j}{2}}$ und wird auf den Fall der Gebiete, die im Innern liegen, durch die folgende Überlegung reduziert.

Es wird eine Halbkugel $b_{\frac{d_j}{4}}$ betrachtet, deren geradliniger Anteil am Rand in y -Richtung ausgerichtet ist (s. Abbildung 7.1). Die Funktionen u bzw. u_h werden auf die Kugeln $B_{\frac{d_j}{2}}$ und $B_{\frac{d_j}{4}}$ (s. Abbildung 7.1) durch $u(x,y) = -u(-x,y)$ und $u_h(x,y) = -u_h(-x,y)$ für $(-x,y) \in b_{\frac{d_j}{2}}$. Das Gitter wird durch Spiegelung an der y -Achse in den Außenbereich ausgedehnt. Außerdem wird der stückweise lineare Unterraum aller $B_{\frac{d_j}{2}}$ so erweitert, daß dieser alle stückweise linearen Funktionen auf $B_{\frac{d_j}{2}}$ enthält, wobei vorausgesetzt wird, daß nicht die Funktionen enthalten sind, die auf dem Rand verschwinden. Dann folgt für den gespiegelten Ausdruck $u - u_h$:

$$(\nabla e, \nabla \Phi) = 0 \quad \text{für alle } \Phi \in \overset{\circ}{S}_h \left(B_{\frac{d_j}{2}} \right)$$

Somit gelten die Abschätzungen (7.3) und (7.4) auch auf den Gebieten $B_{\frac{d_j}{4}}$ und $B_{\frac{d_j}{2}}$. Aus den bekannten Eigenschaften von ungeraden Funktionen folgt weiter:

$$\begin{aligned} \|e\|_{L_\infty \left(b_{\frac{d_j}{4}} \right)} &\leq \|e\|_{L_\infty \left(B_{\frac{d_j}{4}} \right)} \leq c \left(\ln \left(\frac{1}{h} \right) \|u - u_I\|_{L_\infty \left(B_{\frac{d_j}{2}} \right)} \right. \\ &\quad \left. + d_j^{-1} \|e\|_{L_2 \left(B_{\frac{d_j}{2}} \right)} \right) \\ &\leq c \left(\ln \left(\frac{1}{h} \right) \|u - u_I\|_{L_\infty \left(b_{\frac{d_j}{2}} \right)} \right. \\ &\quad \left. + d_j^{-1} \|u - u_I\|_{L_2 \left(B_{\frac{d_j}{2}} \right)} \right) \end{aligned}$$

Diese Abschätzung vervollständigt den Beweis von Lemma 7.2.2, Abschätzung (7.3) für die Halbkugeln am Rand des Gebietes. Der noch ausstehende Beweis für (7.4) in Lemma 7.2.2 folgt in der gleichen Weise. \square

Als nächstes wird eine *a priori* Abschätzung einer gewichteten L_2 -Norm vorgestellt. Es gelte $0 < r < 1$. Desweiteren sei die Gewichtsfunktion σ_j gegeben durch:

$$\sigma_j = (r + d_j)^{-1-\beta} \tag{7.5}$$

Ferner erfülle Ψ für $z \in L_2(\Omega)$ folgende Gleichung:

$$\begin{aligned} -\Delta \Psi &= \sigma_j z & \text{in } \Omega \\ \Psi &= 0 & \text{auf } \partial\Omega \end{aligned}$$

Lemma 7.2.3 *Gegeben sei $\varepsilon > 0$, dann existiert eine Konstante c_ε , so daß für $1 < p = \frac{2}{2-\beta+\varepsilon}$ gilt:*

$$\left\| \nabla^{(2)} \Psi \right\|_{L_p(\Omega)} \leq c_\varepsilon d_j^{-2\beta+\varepsilon} \|z\|_{L_2(\Omega)} \quad (7.6)$$

Beweis:

Mit Lemma 5.2.1 ergibt sich:

$$\left\| \nabla^{(2)} \Psi \right\|_{L_p(\Omega)} \leq c_\varepsilon \left\| \sigma_j z \right\|_{L_p(\Omega)}$$

Die Anwendung der Hölder'schen Ungleichung führt auf

$$\left\| \sigma_j z \right\|_{L_p(\Omega)} = \left(\int_{\Omega} \sigma_j^p |z|^p \, dx \right)^{\frac{1}{p}} \leq \left(\int_{\Omega} \sigma_j^{pq} \, dx \right)^{\frac{1}{pq}} \|z\|_{L_2(\Omega)}$$

mit

$$\frac{1}{q} + \frac{1}{\frac{2}{p}} = 1 \quad \text{bzw. } pq = \frac{2p}{2-p}$$

Es wird die Hilfsvariable $R = r + d_j$ eingeführt. Eine direkte Rechnung führt auf:

$$\begin{aligned} \left(\int_{\Omega} (\sigma_j)^{pq} \right)^{\frac{1}{pq}} &\leq \left(\int_0^{2\pi} \int_0^\infty (r + d_j)^{-(1+\beta)pq} r \, dr \, d\Theta \right)^{\frac{1}{pq}} \\ &\leq c \left(\int_0^\infty R^{-(1+\beta)pq} R \, dr \right)^{\frac{1}{pq}} \\ &\leq c \left(d_j^{-(1+\beta)pq+2} \right)^{\frac{1}{pq}} \\ &\leq c d_j^{-2\beta+\varepsilon} \end{aligned}$$

Damit ist die Behauptung bewiesen. □

Lemma 7.2.4 *Gegeben sei $\varepsilon > 0$. Ferner sei $k \geq j$. Dann existiert eine Konstante c_ε , so daß gilt:*

$$\left\| \nabla^{(2)} \Psi \right\|_{L_2(\Omega_k)} \leq c_\varepsilon d_k^{-1} d_j^{-\beta - \varepsilon} \|z\|_{L_2(\Omega)} \quad (7.7)$$

Beweis:

Die nachfolgende Abschätzung (7.8) folgt aus inneren Regularitätseigenschaften und Abschätzungen bis zum Rand in der gleichen Weise wie der Beweis von Lemma 7.2.2 durch Spiegelung innerer Abschätzungen entlang des Randes. Diese Vorgehensweise wird z.B. in [92] und [93] beschrieben und es ergibt sich:

$$\left\| \nabla^{(2)} \Psi \right\|_{L_2(\Omega_k)} \leq \left\| \sigma_j z \right\|_{L_2(\Omega'_k)} + d_k^{-1} \left\| \nabla \Psi \right\|_{L_2(\Omega'_k)} \quad (7.8)$$

Weiter gilt:

$$\left\| \sigma_j z \right\|_{L_2(\Omega'_k)} \leq d_k^{-1 - \beta} \|z\|_{L_2(\Omega)}$$

Für $q' = 1 + \varepsilon$ und $\frac{1}{p'} + \frac{1}{q'} = 1$ ergibt sich:

$$\left\| \nabla \Psi \right\|_{L_2(\Omega)}^2 = (\sigma_j z, \Psi) \leq \left\| \sigma_j z \right\|_{L_{q'}(\Omega)} \left\| \Psi \right\|_{L_{p'}(\Omega)}$$

Die Anwendung von Sobolev's Ungleichung für Funktionen in $\overset{\circ}{W}_2^1(\Omega)$ führt auf:

$$\left\| \Psi \right\|_{L_{p'}(\Omega)} \leq c_\varepsilon \left\| \nabla \Psi \right\|_{L_2(\Omega)}$$

Unter Verwendung der Hölder'schen Ungleichung folgt:

$$\begin{aligned} \left\| \sigma_j z \right\|_{L_{1+\varepsilon}} &\leq \left(\int_{\Omega} \sigma_j^{2\left(\frac{1+\varepsilon}{1-\varepsilon}\right)} \right)^{\frac{1-\varepsilon}{2(1+\varepsilon)}} \|z\|_{L_2(\Omega)} \\ &\leq c_\varepsilon d_j^{-\beta - \varepsilon} \|z\|_{L_2(\Omega)} \\ &\leq c_\varepsilon h^{-\varepsilon} d_j^{-\beta} \|z\|_{L_2(\Omega)} \end{aligned}$$

Einsetzen all dieser Ungleichungen in (7.8) ergibt das gewünschte Ergebnis (7.7).
□

Lemma 7.2.5 *Gegeben sei $\varepsilon > 0$, Ferner sei $k < j$. Dann existiert eine Konstante c_ε , so daß gilt:*

$$\left\| \nabla^{(2)} \Psi \right\|_{L_2(\Omega_k)} \leq c_\varepsilon d_j^{-\varepsilon} d_k^{-1-\beta} \|z\|_{L_2(\Omega)} \quad (7.9)$$

Beweis:

Es gelte $q = \frac{2}{\beta}$. Aus dem Beweis von Lemma 7.2.4 ergibt sich:

$$\left\| \nabla^{(2)} \Psi \right\|_{L_2(\Omega_k)} \leq c \left\| \sigma_j z \right\|_{L_2(\Omega'_k)} + d_k^{-1-\beta} \|\Psi\|_{L_q(\Omega'_k)}$$

Analog zu Lemma 7.2.4 folgt:

$$\left\| \sigma_j z \right\|_{L_2(\Omega'_k)} \leq c d_k^{-1-\beta} \|z\|_{L_2(\Omega)}$$

Weiterhin gilt mit $\zeta = \frac{2}{2-\beta}$:

$$\|\Psi\|_{L_{\frac{2}{\beta}}(\Omega)} = \sup_{\|\eta\|_{L_\zeta(\Omega)}=1} (\Psi, \eta)$$

Für jedes η läßt sich ein $w \in \overset{\circ}{W}_2^1(\Omega)$ finden, so daß gilt:

$$D(w, v) = (\eta, v) \quad \text{für alle } v \in \overset{\circ}{W}_2^1(\Omega)$$

Dann folgt weiter:

$$\begin{aligned} |(\Psi, \eta)| &= |D(\Psi, w)| = \\ &= |(\sigma_j z, w)| = \left| \left((r + d_j)^{-1} z, (r + d_j)^{-\beta} w \right) \right| \\ &\leq \left\| (r + d_j)^{-1} z \right\|_{L_1(\Omega)} \left\| (r + d_j)^{-\beta} w \right\|_{L_\infty(\Omega)} \\ &\leq c d_j^{-\varepsilon} \|z\|_{L_2(\Omega)} \|w\|_{C^{\beta-\varepsilon}(\Omega)} \end{aligned}$$

Dabei ist $C^{\beta-\varepsilon}(\Omega)$, $0 < \beta - \varepsilon < 1$ der bekannte Hölder-Raum (siehe z.B. [1]).

Aus einer Sobolev-Ungleichung ([1], Lemma 5.17) ergibt sich für $p = \frac{2}{2-\beta+\varepsilon}$ und Lemma 5.2.1 mit $\beta < 1$:

$$\|w\|_{C^{\beta-\varepsilon}(\Omega)} \leq c_\varepsilon \|w\|_{W_p^2(\Omega)} \leq c_\varepsilon \|\eta\|_{L_p(\Omega)}$$

Die Zusammenfassung der Ungleichungen führt auf:

$$\|\Psi\|_{L_{\frac{2}{\beta}}(\Omega)} \leq c_\varepsilon d_j^{-\varepsilon} \|\eta\|_{L_{\frac{2}{2-\beta}}(\Omega)} \leq c_\varepsilon d_j^{-\varepsilon}$$

Schließlich folgt:

$$\left\| \nabla^{(2)} \Psi \right\|_{L_2(\Omega_k)} \leq c_\varepsilon d_j^{-\varepsilon} d_k^{-1-\beta} \|z\|_{L_2(\Omega)}$$

Damit ist der Beweis vollständig und Lemma 7.2.5 ist damit bewiesen. \square

7.3 Formulierung und Diskussion einer neuen Abschätzung des relativen Fehlers des Gradienten und der Funktion in der L_∞ -Norm

In diesem Abschnitt werden die neuen Theoreme zur Abschätzung des relativen Fehlers der Funktion und deren Gradienten formuliert. In Form von Fallunterscheidungen kann der Gültigkeitsbereich der Theoreme abgeschätzt werden. Die Beweise zu den Theoremen werden im anschließenden Abschnitt gesondert dargestellt.

Der relative Fehler des Gradienten wird punktweise durch folgendes Theorem bestimmt:

Theorem 7.3.1 *Gegeben seien u und u_h und es gelte (5.11) und (5.12). Außerdem erfülle u die Bedingung (5.18) auf τ und (5.17) auf ganz Ω . Weiterhin sei ein geometrisch verfeinertes Gitter mit $\rho = h^\gamma$, für ein beliebiges $\gamma \geq 0$, das der Bedingung (5.33) genügt, gegeben. Dann gibt es für ein beliebiges $\varepsilon > 0$ eine Konstante $c(\varepsilon)$, die unabhängig von h und d ist, so daß für $d \leq \text{dist}(\tau, \{0\}) \leq 2d$ gilt:*

$$\left\| \frac{\nabla(u - u_h)}{|\nabla u|} \right\|_{L_\infty(\tau)} \leq c_\varepsilon \left(h + \left(\frac{h\rho}{d} \right)^{2\beta} \right) h^{-\varepsilon} \quad \text{für } \rho \leq d \quad (7.10)$$

Ferner gelte:

$$\left\| \frac{\nabla(u - u_h)}{|\nabla u|} \right\|_{L_\infty(\tau)} \leq c_\varepsilon \left(\frac{h\rho}{d} + \left(\frac{h\rho}{d} \right)^{2\beta} \right) h^{-\varepsilon} \quad \text{für } ch\rho \leq d \leq \rho \quad (7.11)$$

Die Abschätzungen (7.10) und (7.11) zeigen, daß der relative Fehler aus zwei Anteilen besteht. Die ersten Ausdrücke auf der rechten Seite von (7.10) und (7.11) sind Interpolationsfehler. Bei Ungleichung (7.10) wurde das Gitter geometrisch verfeinert so daß dieser Term von fast optimaler Ordnung $ch^{1-\varepsilon}$ ist. Im Bereich

$ch\rho \leq d \leq \rho$ wird nicht mehr verfeinert und das Gitter ist quasiuniform. Somit besitzt der erste Ausdruck in Ungleichung (7.11) keine optimale Genauigkeit mehr.

Der zweite Term der rechten Seite der Abschätzungen (7.10) und (7.11) wird durch den "pollution"-Effekt hervorgerufen. Erstmals wurde der Fehler in der Methode der finiten Elemente als die Summe eines lokalen Interpolationsfehlers und einem globaleren Fehler, der als "pollution" bezeichnet, wird in der Arbeit von Nitsche und Schatz [77], sowie in den Artikel von Schatz und Wahlbin [92], [96], beschrieben. Weitere Arbeiten hierzu sind in [111] zu finden. Der Fehler, hervorgerufen durch den "pollution"-Effekt, entsteht in den hier betrachteten Problemen hauptsächlich durch die Singularität und der in der Umgebung der Singularität fehlenden Glattheit der Lösung.

Nachfolgend wird das Verhalten des relativen Fehlers im Bereich, in dem das Gitter geometrisch verfeinert wurde und im Bereich, in dem das Gitter quasiuniform ist, beschrieben.

Fall 7.3.1

Mit vorgegebenem Verfeinerungsabstand ρ folgt aus Ungleichung (7.10) fast optimale Konvergenz bis zum Abstand:

$$\rho < \rho h^{1-\frac{1}{2\beta}} \leq d \leq 1$$

Der Einfluß des "pollution"-Effekts ist in diesem Intervall nicht signifikant. Im Bereich $\rho \leq d \leq \rho h^{1-\frac{1}{2\beta}}$ ist der "pollution"-Effekt dominant und es ergibt sich eine Konvergenzrate von:

$$\left(\frac{h\rho}{d}\right)^{2\beta} h^{-\varepsilon}$$

Mit der Annahme, daß der erste Spannungsintensitätsfaktor das Verhalten der Lösung bestimmt, ist der relative Fehler unabhängig von β und robust gegenüber den Spannungsintensitätsfaktoren und somit der rechten Seite f .

Fall 7.3.2

$$ch\rho \leq d \leq \rho, \frac{1}{2} < \beta \leq 1.$$

In diesem Fall dominiert der Interpolationsfehler, ist jedoch nicht von optimaler Ordnung. Aus Theorem 7.3.1 folgt hierfür, daß bei einer Halbierung des Abstandes zur Singularität der relative Fehler des Gradienten um den Faktor 2 ansteigen

wird.

Für den relativen Fehler der Funktion gilt das folgende Theorem:

Theorem 7.3.2 *Gegeben seien u und u_h und es gelte (5.11) und (5.12). Außerdem erfülle u die Bedingung (5.23) auf τ und die Bedingung (5.17) auf ganz Ω . Weiterhin sei ein geometrisch verfeinertes Gitter mit $\rho = h^\gamma$, für ein beliebiges $\gamma \geq 0$, das der Bedingung (5.33) genügt, gegeben. Dann gibt es für ein beliebiges $\varepsilon > 0$ eine Konstante $c(\varepsilon)$, die unabhängig von h und d ist, so daß für $d \leq \text{dist}(\tau, \{0\}) \leq 2d$ gilt:*

$$\left\| \frac{u - u_h}{u} \right\|_{L_\infty(\tau)} \leq c_\varepsilon \left(h^2 + \left(\frac{h\rho}{d} \right)^{2\beta} \right) h^{-\varepsilon} \quad \text{für } \rho \leq d \quad (7.12)$$

Weiter gelte:

$$\left\| \frac{u - u_h}{u} \right\|_{L_\infty(\tau)} \leq c_\varepsilon \left(\left(\frac{h\rho}{d} \right)^2 + \left(\frac{h\rho}{d} \right)^{2\beta} \right) h^{-\varepsilon} \quad \text{für } ch\rho \leq d \leq \rho \quad (7.13)$$

Analog zu Theorem 7.3.1 sind die ersten beiden Ausdrücke in Ungleichung (7.12) und Ungleichung (7.13) die Interpolationsfehler und die zweiten Terme stammen vom "pollution"-Effekt. Der relative Fehler der Funktion zeigt nach Ungleichung (7.12) mit einer geometrischen Verfeinerung die fast optimale Ordnung $O(h^{2-\varepsilon})$, während im Gültigkeitsbereich von Ungleichung (7.13) kein optimales Verhalten vorliegt. Analog zu Theorem 7.3.1 sind die ersten Ausdrücke in Ungleichung (7.12) und Ungleichung (7.13) die Interpolationsfehler. Die geometrische Verfeinerung bewirkt in Ungleichung (7.11) fast optimale Genauigkeit der Ordnung $ch^{2-\varepsilon}$, während in Ungleichung (7.12) keine optimale Genauigkeit vorliegt.

In den nachfolgenden Fallunterscheidungen werden die relativen Fehler im Bereich der geometrischen Verfeinerung und im Bereich der quasiuniformen Vernetzung diskutiert.

Fall 7.3.3

Ist der Verfeinerungsabstand ρ vorgegeben, so folgt aus Ungleichung (7.12), daß fast optimale Konvergenz bis zum Abstand

$$\rho < \rho h^{1-\frac{1}{\beta}} \leq d \leq 1 \quad (7.14)$$

zu erwarten ist.

Für $\rho \leq d \leq \rho h^{1-\frac{1}{\beta}}$ dominiert der "pollution"-Effekt und die Konvergenzrate beträgt:

$$\left(\frac{h\rho}{d}\right)^{2\beta} h^{-\varepsilon}$$

Der relative Fehler der Funktion ist robust gegenüber Änderungen der Spannungsintensitätsfaktoren und folglich unabhängig von der rechten Seite.

Fall 7.3.4

$$ch\rho \leq d \leq \rho, \frac{1}{2} \leq \beta < 1.$$

Hier spielt der "pollution"-Fehler die entscheidende Rolle. Der relative Fehler besitzt in diesem Bereich die nicht optimale Konvergenz $O(h^{2\beta})$, steigt jedoch bei Halbierung der Maschenweite um den Faktor 2 an. Auch hier ist der relative Fehler unter der Annahme, daß der erste Spannungsintensitätsfaktor das Verhalten der Lösung bestimmt, robust gegenüber Änderungen von Spannungsintensitätsfaktoren und somit unabhängig von der rechten Seite f .

7.4 Beweis der Theoreme zur Abschätzung des relativen Fehlers des Gradienten und der Funktion in der L_∞ -Norm

Die Theoreme 7.3.1 und 7.3.2 werden in diesem Abschnitt bewiesen. Es ist ausreichend, in Theorem 7.3.1 die Aussage (7.10) und in Theorem 7.3.2 die Aussage (7.12) zu beweisen. Die Beweise von (7.11) und (7.13) können mit unwesentlichen Änderungen analog vorgenommen werden.

Die Theoreme 7.3.1 und 7.3.2 werden in 2 Schritten bewiesen. In den ersten beiden Abschätzungen werden 7.10 und 7.12 so reduziert, daß eine Fehlerabschätzung für die gleiche gewichtete L_2 -Norm durch Modifikation einer Abschätzung aus [94] möglich ist. Im zweiten Schritt werden diese gewichteten Normen abgeschätzt.

Beweis der Theoreme 7.3.1 und 7.3.2:

In diesem Beweis ist ε eine positive reelle Zahl, die in den Beweisschritten nicht immer gleich bleibt, jedoch beliebig klein gewählt werden kann und unabhängig von u , h und u_h ist.

Schritt 1:

Unter Verwendung von (7.3) aus Lemma 7.2.2 ergibt sich für jedes $0 \leq j \leq J$ mit $d_j \geq \rho$ der Ausdruck

$$\begin{aligned} \|\nabla e\|_{L^\infty(\Omega_j)} &\leq c \left(\|\nabla(u - u_I)\|_{L^\infty(\Omega'_j)} \right. \\ &\quad \left. + d_j^{-1} \|u - u_I\|_{L^\infty(\Omega'_j)} + d_j^{-2} \|e\|_{L_2(\Omega'_j)} \right) \end{aligned} \quad (7.15)$$

Mit Lemma 5.4.1, der lokalen Maschenweite hd_j auf Ω_j , der Annahme (5.17) und einer direkten Rechnung folgt:

$$\begin{aligned} \|\nabla(u - u_I)\|_{L^\infty(\Omega'_j)} + d_j^{-1} \|u - u_I\|_{L^\infty(\Omega'_j)} &\leq chd_j |u|_{W_\infty^2(\Omega'')} \quad (7.16) \\ &\leq c|a_1|hd_j d_j^{\beta-2} \\ &\leq c|a_1|hd_j^{\beta-1} \end{aligned}$$

Einsetzen von (7.16) in (7.15) und nochmaliger Verwendung von (5.17) ergibt für den relativen Fehler:

$$\left\| \frac{\nabla(u - u_h)}{|\nabla u|} \right\|_{L^\infty(\tau)} \leq c \left(h^{1-\varepsilon} + \frac{d_j^{-1-\beta}}{|a_1|} \|u - u_h\|_{L_2(\Omega'_j)} \right) \quad (7.17)$$

Die Ungleichung (7.10) folgt, falls der Ausdruck

$$\frac{d_j^{-1-\beta}}{|a_1|} \|u - u_h\|_{L_2(\Omega'_j)} \leq c_\varepsilon \left[\left(\frac{h\rho}{d_j} \right)^{2\beta-\varepsilon} + h^{2-\varepsilon} \right] \quad (7.18)$$

bewiesen ist.

In der gleichen Form folgt mit (7.3) aus Lemma 7.2.2:

$$\|u - u_h\|_{L^\infty(\Omega_j)} \leq c \left(\ln \left(\frac{1}{h} \right) \|u - u_h\|_{L^\infty(\Omega'_j)} + d_j^{-1} \|u - u_h\|_{L_2(\Omega'_j)} \right) \quad (7.19)$$

Es wird vorausgesetzt, daß auf einem Dreieck $\tau \subset \Omega$ die Bedingung (5.23) gilt. Mittels Interpolationsabschätzungen, eingesetzt in (7.19) ergibt sich:

$$\begin{aligned} \left\| \frac{u - u_h}{u} \right\|_{L^\infty(\tau)} &\leq c \left(\ln \left(\frac{1}{h} \right) \frac{(h^2 d_j^2 d_j^{\beta-2})}{d_j^\beta} + \frac{d_j^{-1} \|u - u_h\|_{L_2(\Omega'_j)}}{|a_1| d_j^\beta} \right) \\ &\leq c \left(h^{2-\varepsilon} + \frac{d_j^{-1-\beta}}{|a_1|} \|u - u_h\|_{L_2(\Omega'_j)} \right) \end{aligned} \quad (7.20)$$

Auch hier gilt, daß aus dem Beweis von Ungleichung (7.18) die Behauptung (7.12) folgt.

Schritt 2:

Als nächstes wird die gewichtete Funktion

$$\sigma_j = \left(\frac{1}{r + d_j} \right)^{1+\beta} = (r + d_j)^{-(1+\beta)} \quad (7.21)$$

betrachtet. der Beweis von (7.18) folgt, falls bewiesen werden kann, daß

$$\frac{1}{|a_1|} \|\sigma_j e\|_{L_2(\Omega)} \leq c_\varepsilon \left[\left(\frac{h\rho}{d_j} \right)^{2\beta-\varepsilon} + h^{2-\varepsilon} \right] \quad \text{für } \rho \leq d_j \quad (7.22)$$

gilt. Zum Beweis von (7.22) wird das Dualitätsargument von Aubin-Nitsche verwendet und es folgt:

$$\|\sigma_j e\|_{L_2(\Omega)} = \sup_{\substack{z \in C_0^\infty(\Omega) \\ \|z\|_{L_2(\Omega)}=1}} (\sigma_j e, z) \quad (7.23)$$

Für jedes z mit $\|z\|_{L_2(\Omega)}=1$ sei Ψ die eindeutige schwache Lösung in $\overset{\circ}{W}_2^1(\Omega)$ der Gleichung:

$$\begin{aligned} -\Delta \Psi &= \sigma_j z & \text{in } \Omega \\ \Psi &= 0 & \text{auf } \partial\Omega \end{aligned}$$

In der schwachen Formulierung erfüllt Ψ die Bedingung:

$$(v, \sigma_j z) = D(v, \Psi) \quad \text{für alle } v \in \overset{\circ}{W}_2^1(\Omega) \quad (7.24)$$

Somit ergibt sich aus (6.10) für jedes z mit $v = e$:

$$(\sigma_j e, z) = (e, \sigma_j z) = D(e, \Psi - \Psi_I) \quad (7.25)$$

Sei $\hat{\Omega} = \Omega_{In} \cup (\cup_{k=j}^J \bar{\Omega}_k)$, dann gilt:

$$\begin{aligned} |D(e, \Psi - \Psi_I)| &\leq \|\nabla e\|_{L_2(\hat{\Omega})} \|\nabla(\Psi - \Psi_I)\|_{L_2(\hat{\Omega})} \\ &\quad + \sum_{k=0}^{j-1} \|\nabla e\|_{L_2(\Omega_k)} \|\nabla(\Psi - \Psi_i)\|_{L_2(\Omega_k)} \\ &= I_1 \quad + \quad I_2 \end{aligned} \quad (7.26)$$

Als nächstes wird mit der Abschätzung von I_1 begonnen. Die Anwendung von (6.8) auf $\hat{\Omega}$ führt auf:

$$\begin{aligned} \|\nabla e\|_{L_2(\hat{\Omega}_j)} &\leq c \left(\|\nabla(u - u_I)\|_{L_2(\hat{\Omega}_{j-1})} + d_j^{-1} \|u - u_I\|_{L_2(\hat{\Omega}_{j-1})} \right. \\ &\quad \left. + d_j^{-1} \|e\|_{L_2(\hat{\Omega}_j)} \right) \end{aligned} \quad (7.27)$$

Hierbei ist $\hat{\Omega}_{j-1} = \hat{\Omega}_j \cup \bar{\Omega}_{j-1}$. Es gilt:

$$\begin{aligned} \|\nabla(u - u_I)\|_{L_2(\hat{\Omega}_{j-1})} &\leq \|\nabla(u - u_I)\|_{L_2(\Omega_m)} \\ &\quad + \sum_{k=j-1}^J \|\nabla(u - u_I)\|_{L_2(\Omega_k)} \end{aligned} \quad (7.28)$$

Nochmalige Anwendung von Lemma 5.4.1 und 5.4.2 führt für $p = \frac{2}{2-\beta+\varepsilon}$ und $\Omega'_{In} = \Omega_{In} \cup \bar{\Omega}_J$ auf:

$$\|\nabla(u - u_I)\|_{L_2(\Omega'_{In})} \leq c_\varepsilon (h\rho)^{\beta-\varepsilon} |u|_{W_p^2(\Omega)} \leq c_\varepsilon |a_1| (h\rho)^{\beta-\varepsilon} \quad (7.29)$$

Es gilt $u \in W_2^2(\Omega \setminus \Omega_{In})$, so daß sich bei Anwendung von Lemma 5.2.1 und Lemma 5.4.1 ergibt:

$$\|\nabla(u - u_I)\|_{L_2(\Omega_k)} \leq ch d_k |u|_{W_2^2(\Omega'_k)} \leq ch |a_1| d_k^\beta \quad (7.30)$$

Die Kombination von (7.29) und (7.30) führt auf:

$$\begin{aligned} \|\nabla(u - u_I)\|_{L_2(\hat{\Omega}_{j-1})} &\leq c_\varepsilon |a_1| \left((h\rho)^{\beta-\varepsilon} + h \sum_{k=j-1}^J d_k^\beta \right) \\ &\leq c_\varepsilon |a_1| \left((h\rho)^{\beta-\varepsilon} + h d_j^\beta \right) \end{aligned} \quad (7.31)$$

In gleicher Weise folgt:

$$d_j^{-1} \|u - u_I\|_{L_2(\hat{\Omega}_{j-1})} \leq c_\varepsilon |a_1| \left((h\rho)^{\beta-\varepsilon} + h d_j^\beta \right) \quad (7.32)$$

Einsetzen von (7.31) und (7.32) in (7.27) liefert:

$$\|\nabla e\|_{L_2(\hat{\Omega}_j)} \leq c_\varepsilon |a_1| \left((h\rho)^{\beta-\varepsilon} + h d_j \right) + c d_j^{-1} \|e\|_{L_2(\hat{\Omega}_{j-1})} \quad (7.33)$$

Im folgenden wird der Ausdruck $\|\nabla(\Psi - \Psi_I)\|_{L_2(\hat{\Omega}_{j-1})}$ abgeschätzt:

$$\begin{aligned} \|\nabla(\Psi - \Psi_I)\|_{L_2(\hat{\Omega}_{j-1})} &\leq \|\nabla(\Psi - \Psi_I)\|_{L_2(\Omega_m)} \\ &\quad + \sum_{k=j-1}^J \|\nabla(\Psi - \Psi_I)\|_{L_2(\Omega_k)} \end{aligned} \quad (7.34)$$

Weiterhin ergibt sich bei Verwendung der Lemmata 5.4.1, 5.4.2 und 7.2.3 für $p = \frac{2}{2-\beta+\varepsilon}$:

$$\begin{aligned} \|\nabla(\Psi - \Psi_I)\|_{L_2(\Omega_{In})} &\leq c_\varepsilon (h\rho)^{\beta-\varepsilon} |\Psi|_{W_p^2(\Omega)} \\ &\leq c_\varepsilon \frac{(h\rho)^{\beta-\varepsilon}}{d_j^{2\beta-\varepsilon}} \|z\|_{L_2(\Omega)} \end{aligned} \quad (7.35)$$

Für jedes $k \geq j - 1$ ergibt sich aus Lemma 7.2.3:

$$\|\nabla(\Psi - \Psi_I)\|_{L_2(\Omega_k)} \leq chd_k |\Psi|_{W_2^2(\Omega'_k)} \leq chd_k d_k^{-1} d_j^{-\beta-\varepsilon} \quad (7.36)$$

Summieren über $j - 1 \leq k \leq J$ führt auf:

$$\|\nabla(\Psi - \Psi_I)\|_{L_2(\hat{\Omega}_{j-1})} \leq c_\varepsilon \left[\frac{(h\rho)^{\beta-\varepsilon}}{d_j^{2\beta-\varepsilon}} + h^{1-\varepsilon} d_j^{-\beta-\varepsilon} \right] \quad (7.37)$$

Bemerkung 7.4.1

Nach Voraussetzung gilt $d_j \leq \rho \leq h^\gamma$ für ein beliebiges positives γ . Somit ist $d_j^{-\varepsilon} \leq h^{-\gamma\varepsilon}$, mit beliebig kleinem ε .

Unter Beachtung von Bemerkung 7.4.1 folgt weiter:

$$\begin{aligned} \|\nabla e\|_{L_2(\hat{\Omega}_j)} \|\nabla(\Psi - \Psi_I)\|_{L_2(\hat{\Omega}_j)} &\leq c_\varepsilon \left[|a_1| \left((h\rho)^{\beta-\varepsilon} + h^{1-\varepsilon} d_j^\beta \right) \right. \\ &\quad \left. + d_j^{-1} \|e\|_{L_2(\hat{\Omega}_j)} \right] \left[\frac{(h\rho)^{\beta-\varepsilon}}{d_j^{2(\beta-\varepsilon)}} + h^{1-\varepsilon} d_j^{-\beta-\varepsilon} \right] \\ &\leq c_\varepsilon |a_1| \left[\left(\frac{h\rho}{d_j} \right)^{2(\beta-\varepsilon)} + h^{(2-2\varepsilon)} \right. \\ &\quad \left. + h^{1-\varepsilon} \left(\frac{h\rho}{d_j} \right)^{\beta-\varepsilon} \right] \\ &\quad + c_\varepsilon d_j^{-1-\beta} \|e\|_{L_2(\hat{\Omega}_j)} \left(\left(\frac{h\rho}{d_j} \right)^{\beta-\varepsilon} + h^{1-\varepsilon} d_j^{-\varepsilon} \right) \end{aligned} \quad (7.38)$$

Falls gilt

$$\left(\frac{h\rho}{d_j} \right)^{\beta-\varepsilon} \leq h^{1-\varepsilon}$$

folgt:

$$h^{1-\varepsilon} \left(\frac{h\rho}{d_j} \right)^{\beta-\varepsilon} \leq h^{2-2\varepsilon}$$

Ebenfalls ergibt sich, falls

$$h^{1-\varepsilon} \leq \frac{h\rho}{d_j}$$

die Beziehung

$$h^{1-\varepsilon} \leq \left(\frac{h\rho}{d_j} \right)^{\beta-\varepsilon}$$

Obige Abschätzung führt auf:

$$h^{1-\varepsilon} \left(\frac{h\rho}{d_j} \right)^{\beta-\varepsilon} \leq \left(\frac{h\rho}{d_j} \right)^{2(\beta-\varepsilon)}$$

Somit kann (7.38) durch

$$I_1 = \|\nabla e\|_{L_2(\hat{\Omega}_j)} \|\nabla(\Psi - \Psi_I)\|_{L_2(\hat{\Omega}_j)} \leq c_\varepsilon |a_1| \left[\left(\frac{h\rho}{d_j} \right)^{2(\beta-\varepsilon)} + h^{2(\beta-\varepsilon)} \right] + c_\varepsilon h^{\beta-\varepsilon} \|\sigma_j e\|_{L_2(\Omega)} \quad (7.39)$$

Zur Bestimmung von (7.39) wurde außerdem die Beziehung

$$\left(\frac{h\rho}{d_j} \right)^{\beta-\varepsilon} + h^{1-\varepsilon} d_j^{-\varepsilon} \leq h^{\beta-\varepsilon}$$

für ein hinreichend kleines ε verwendet (siehe dazu Bemerkung 7.4.1).

Als nächstes wird eine Abschätzung des Ausdrucks I_2 gegeben. Die Größe I_2 ist gegeben durch:

$$I_2 = \sum_{k=0}^{j-1} \|\nabla e\|_{L_2(\Omega_k)} \|\nabla(\Psi - \Psi_I)\|_{L_2(\Omega_k)} \quad (7.40)$$

Mittels Lemma 7.2.3 folgt:

$$\begin{aligned} \|\nabla e\|_{L_2(\Omega_k)} &\leq c \left(h d_k |u|_{W_2^2(\Omega'_k)} + d_k^{-1} \|e\|_{L_2(\Omega'_k)} \right) \\ &\leq c \left(|a_1| h d_k^\beta + d_k^{-1} \|e\|_{L_2(\Omega'_k)} \right) \end{aligned}$$

Analog zur Vorgehensweise in (7.36), aber diesmal unter Verwendung von Lemma 7.2.5 folgt für I_2 :

$$\begin{aligned} I_2 &\leq c \sum_{k=0}^{j-1} \left(|a_1| h d_k^\beta + d_k^{-1} \|e\|_{L_2(\Omega'_k)} \right) (h^{1-\varepsilon} d_k) (d_k^{-1-\beta}) \\ &\leq c \sum_{k=0}^{j-1} \left(|a_1| h^{2-\varepsilon} + h^{1-\varepsilon} d_k^{-1-\beta} \right) \|e\|_{L_2(\Omega'_k)} \end{aligned}$$

Aufgrund der Beziehung $j \leq J \leq c \ln\left(\frac{1}{h}\right)$ und mit einem geeignet gewählten ε kann Ungleichung (7.40) in

$$I_2 \leq c |a_1| h^{2-\varepsilon} + h^{1-\varepsilon} \|\sigma_j e\|_{L_2(\Omega)} \quad (7.41)$$

übergeführt werden.

In Abschätzung (7.26) wird (7.41) und (7.39) eingesetzt. Weiter wird das Ergebnis aus (7.27) und danach (7.23) verwendet. Daraus ergibt sich die Ungleichung:

$$\frac{1}{|a_1|} \|\sigma_j e\|_{L_2(\Omega)} \leq c_\varepsilon h^{-\varepsilon} \left(\left(\frac{h\rho}{d_j} \right)^{2\beta} + h^2 \right) + c_\varepsilon h^{\beta-\varepsilon} \frac{1}{|a_1|} \|\sigma_j e\|_{L_2(\Omega)}$$

Ungleichung (7.22) folgt durch die Wahl eines hinreichend kleinen ε und einer Maschenweite h , die ebenfalls geeignet klein gewählt wird. Dadurch ergibt sich

$$c_\varepsilon h^{\beta-\varepsilon} \leq \frac{1}{2}$$

und unter Anwendung eines "kick back argument" wird der Beweis vollständig.

□

Kapitel 8

Numerische Experimente

In diesem Kapitel werden die numerischen Experimente zu den Theoremen 6.3.1, 7.3.1 und 7.3.2 beschrieben. In Abschnitt 8.1 wird das Randwertproblem der numerischen Experimente vorgestellt. Es folgt in Abschnitt 8.2 ein Algorithmus zur Konstruktion eines geometrischen Gitters. In Abschnitt 8.3 werden die in den numerischen Experimenten verwendeten Funktionenklassen vorgestellt und bewiesen, daß deren Eigenschaften durch die erste Singularitätenfunktion bestimmt wird und sie somit die Voraussetzungen der Theoreme 6.3.1, 7.3.1 und 7.3.2 erfüllen. Daran anschließend werden in den Abschnitten 8.5 bis 8.9 die numerischen Experimente vorgestellt. Die Rechnungen werden jeweils auf dem Gebiet einer geschlitzten Kreisscheibe ($\alpha = 2\pi$, $\beta = \frac{1}{2}$) und einem Kreissegment ($\alpha = \frac{3}{2\pi}$, $\beta = \frac{2}{3}$) auf unverfeinerten und geometrisch verfeinerten Gittern vorgenommen.

Eine wichtige Voraussetzung zur Durchführung der numerischen Experimente ist, daß die 1. Singularitätenfunktion das Verhalten der Lösung bestimmt und die Annahmen (5.17) und (5.18) erfüllt sind. Unter diesen Voraussetzungen zeigen die Ergebnisse der numerischen Experimente die Optimalität der Theoreme 6.3.1, 7.3.1 und 7.3.2.

8.1 Das Randwertproblem der numerischen Experimente

In diesem Abschnitt wird das elliptische Randwertproblem der numerischen Experimente vorgestellt.

Die numerischen Experimente werden mit der Poisson-Gleichung auf einem stück-

weise glatten, ebenen Gebiet $\Omega \subset \mathbb{R}^2$ mit $\pi < \alpha \leq 2\pi$ mit einer nichtkonvexen einspringenden Ecke, die im Ursprung des Koordinatensystems liegt, durchgeführt. Mit den Polarkoordinaten (r, Θ) ist das Gebiet Ω durch

$$\Omega = \{(r, \Theta) : 0 < r < 1, 0 < \Theta < \alpha\} \quad (8.1)$$

beschrieben. Der Rand Γ des Gebietes ist mit den Teilränder Γ_1, Γ_2 und Γ_3 definiert durch $\Gamma = \Gamma_1 + \Gamma_2 + \Gamma_3$. Die Rechnungen werden auf den Gebieten mit $\alpha = 2\pi, \beta = \frac{1}{2}$ und $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$ (siehe Abbildung 8.1) durchgeführt.

Das Dirichlet-Problem lautet:

$$-\Delta u = f \quad (8.2)$$

$$u = 0 \quad \text{auf } \Gamma_1 \quad \text{und } \Gamma_2 \quad (8.3)$$

$$u = g \neq 0 \quad \text{auf } \Gamma_3$$

Die Entwicklung der Lösung u in einer Umgebung des Ursprungs $\{0\}$ ergibt, falls $f \in C^\infty$:

$$u = \sum_{j=1}^N a_j r^{j\beta} \sin j\beta\Theta + w(r, \Theta) \quad (8.4)$$

Die Koeffizienten a_j werden als Spannungsintensitätsfaktoren bezeichnet. Die Spannungsintensitätsfaktoren a_j sind durch die rechte Seite f der Gleichung (5.11) und den dazugehörigen Randwerten g festgelegt. Spannungsintensitätsfaktoren wurden bereits in Kapitel 5.2 eingeführt und werden deshalb hier nicht weiter erläutert (vergl. Gleichung (5.15) in Kapitel 5.2 und die Kommentare dazu). Der Restterm $w(r, \Theta)$ in Gleichung (8.4) ist beliebig glatt, falls N hinreichend groß gewählt wird. Die genaue Herleitung von (8.4) ist z.B. in [46] dargestellt.

Für $\beta = \frac{\pi}{\alpha}$ gilt:

$$\frac{1}{2} \leq \beta < 1 \quad (8.5)$$

Mit $j = 1, 2, \dots, N$ wird die j -te Singularitätenfunktion definiert durch:

$$S_j = r^{j\beta} \sin j\beta\Theta \quad (8.6)$$

Für den Fall $f = 0$ im Gebiet Ω , d.h. daß u eine harmonische Funktion auf Ω

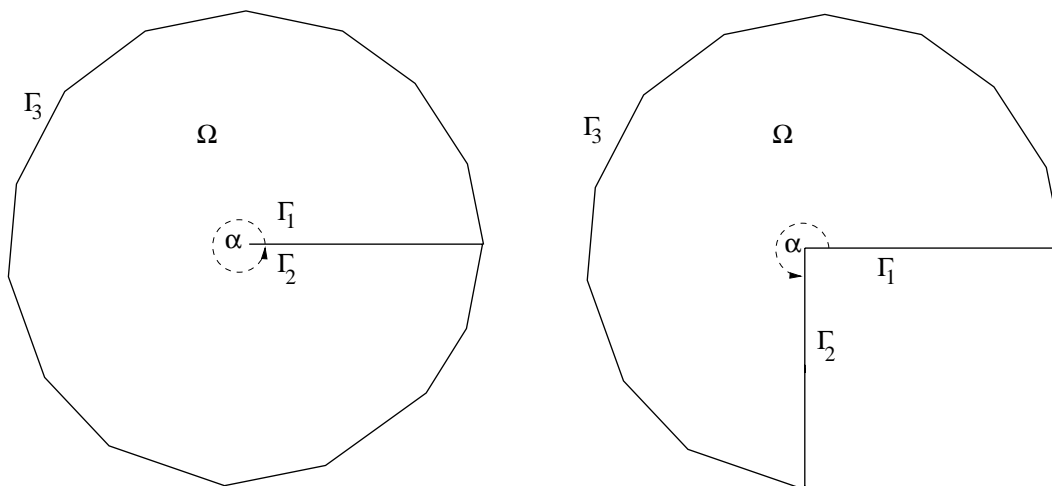


Abbildung 8.1: Rechengebiete der numerischen Experimente

ist, erhält Gleichung (8.4) die Darstellung

$$u = \sum_{j=1}^{\infty} a_j \mathcal{S}_j \quad (8.7)$$

In diesem Fall sind die Spannungsintensitätsfaktoren a_j die Fourierkoeffizienten und werden als

$$a_j = \frac{2}{\alpha} \int_0^{\alpha} g(\Theta) \sin(j\beta\Theta) d\Theta \quad (8.8)$$

mit $g(0) = g(\alpha) = 0$ definiert.

8.2 Algorithmus zur Konstruktion eines geometrischen Gitters

In diesem Abschnitt wird der Algorithmus zur Erzeugung eines geometrischen Gitters, der im Programmpaket *UG* [11] implementiert worden ist, beschrieben.

Ein geometrisches Gitter ist in dieser Arbeit ein Gitter, dessen Maschenweite h mit $d_j = 2^{-j}$, $j = 0, 1, 2, \dots, J$ nach folgendem Ausdruck berechnet wird (vergl. auch Definition 5.4.2 auf Seite 89):

$$h_{d_j} = \begin{cases} h & \text{für } d \leq 1 \\ hd_j & \text{für } \rho \leq d \leq 1 \\ h\rho & \text{für } 0 \leq d \leq \rho \end{cases} \quad (8.9)$$

Mit l wird die Länge des Randsegmentes, mit ρ der Verfeinerungsabstand und mit N die Anzahl der Randknoten auf Γ_3 (siehe Abbildung 8.1) bezeichnet. Die

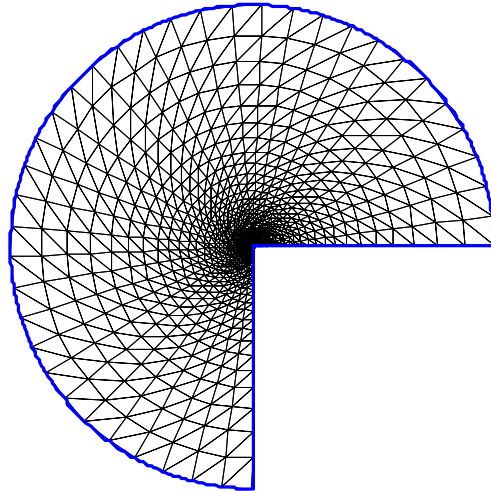


Abbildung 8.2: Geometrisches Gitter mit der Anfangsmaschenweite $h = \frac{1}{8}$ und $\rho = h^5$

Größe N_F gibt die Anzahl der Knoten entlang des Radius r bis zum Verfeinerungsabstand ρ an (siehe Abbildung 8.3). Die Anzahl N_F der Gitterknoten bis zum Verfeinerungsabstand ρ wird durch

$$N_F = \frac{(1-h)^J}{h\rho} = \frac{1}{h}$$

vom Benutzer vorgegeben. Ferner wird der Radius r_j zum j -ten Punkt einer radialen Linie des Gitters folgendermaßen bezeichnet (siehe Abbildung 8.3) :

$$r_j = (1-h)^j, \quad j = 0, 1, 2, \dots, J$$

Die Maschenweite h_{d_j} ist mit r_j folgendermaßen verknüpft:

$$h_{d_j} = r_j - r_{j+1} \quad j = 0, 1, 2, \dots, J$$

Die Maschenweite der Elemente des geometrischen Gitters werden bis zum Verfeinerungsabstand ρ nach Vorschrift (8.9) verfeinert. Im Bereich zwischen Verfeinerungsabstand ρ und Ursprung bleibt die Maschenweite konstant und es gilt $h_\rho = h\rho$. Der Gitterparameter ρ wird zur Berechnung der Anzahl der Punkte J entlang einer radialen Linie herangezogen. Für J gilt:

$$(1-h)^J = \rho$$

$$J = \left\lceil \frac{\ln \rho}{\ln(1-h)} \right\rceil + 1$$

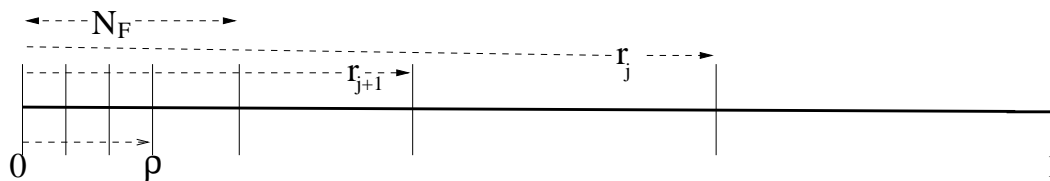


Abbildung 8.3: Konstruktion des geometrischen Gitters entlang einer radialen Linie

Die Konstruktion des geometrischen Gitters erfolgt in zwei Schritten. Als erstes werden die Knoten berechnet und in einer Matrix abgespeichert. Im zweiten Schritt werden aus den Knoten reguläre Dreieckselemente erzeugt.

Der Algorithmus ist folgendermaßen aufgebaut:

Algorithmus 8.2.1

1. Konstruktion radialer Linien mit $N = \left[\frac{l}{h} \right] + 1$
2. Berechnung der Knoten des geometrischen Gitters bei Radius r_j

for $m = 0, 1, 2, \dots, N$

for $j = 0, 1, 2, \dots, J$

$$x(m, j) = (1 - h)^j \cdot \cos(m \cdot \Theta)$$

$$y(m, j) = (1 - h)^j \cdot \sin(m \cdot \Theta)$$

for $m = 0, \dots, N$

for $k = 1, \dots, N_F$

$$x(m, k + J) = (1 - h)^J \left(1 - \frac{k}{N_F} \right) \cos(m \cdot \Theta)$$

$$y(m, k + J) = (1 - h)^J \left(1 - \frac{k}{N_F} \right) \sin(m \cdot \Theta)$$

Ein geometrisches Gitters mit Maschenweite $h = 0.125$ auf dem Rand des Gebietes und einem Verfeinerungsabstand $\rho = h^5 = 3.05 \cdot 10^{-5}$ ist in Abb. 8.2 dargestellt.

8.3 Funktionenklassen

In diesem Abschnitt werden anhand von 3 Beispielen Funktionsklassen eingeführt, die die Bedingungen (5.17) und (5.18) (Kapitel 5 Abschnitt 5.3) erfüllen. Die Eigenschaften dieser Funktionsklassen werden von der ersten Singularitätenfunktion bestimmt.

Beispiel 8.3.1

Im folgenden werden Lösungen u der Form (8.7), d.h. $f = 0$ in Ω und $g \neq 0$ auf Γ_3 mit Koeffizienten a_j gemäß (8.10) betrachtet. Als Klasse von Funktionen $u \in C(\bar{\Omega})$ mit $g \in C(\bar{\Gamma}_3)$ sollen die Funktionen betrachtet werden, für die $g \geq 0$ ohne Restriktion für die maximalen Werte von g , gilt. Da u eine harmonische Funktion in Ω ist, folgt aus dem Maximum-Prinzip, daß $u \geq 0$ ist. Somit besteht die Klasse der betrachteten Funktionen aus den nicht-negativen harmonischen Funktionen auf Ω mit stetigen Randbedingungen.

Es wird gezeigt, daß für diese Klasse von Funktionen Abschätzungen der Form (5.17) und (5.18) gelten. Für $r \leq r_0$ soll noch einmal betont werden, daß r_0 unabhängig von $g(\Theta)$ ist. Da $\frac{1}{2} \leq \beta < 1$ gewählt wird, gilt weiter, daß r_0 unabhängig von allen Funktionen dieser Klasse ist. Ein Problem besteht in der Glattheit der Funktionen g auf dem Rand Γ . Ist g nur stetig, so liefert die finite Elemente Methode i.a. nur eine unzureichende Aproximation von u nicht nur nahe $r = 0$ sondern auf ganz $\bar{\Omega}$, unabhängig davon, ob das Gitter in der Nähe des Ursprungs verfeinert wurde.

Nachfolgend wird gezeigt, daß die Funktionen dieser Klasse die Bedingung (5.17) erfüllen.

Nach Gleichung (5.15) (Kapitel 5.2) gilt für ein beliebiges $j = 1, 2, \dots, N$

$$a_j = \frac{2}{\alpha} \int_0^\alpha g(\Theta) \sin(j\beta\Theta) d\Theta \quad (8.10)$$

Die Erweiterung des Ausdrucks mit $\sin(\beta\Theta)$ ergibt:

$$a_j = \frac{2}{\alpha} \int_0^\alpha g(\Theta) \sin(\beta\Theta) \frac{\sin(j\beta\Theta)}{\sin(\beta\Theta)} d\Theta \quad (8.11)$$

Es gilt $g \geq 0$ und $\sin(\beta\Theta) \geq 0$ und damit folgt

$$b_j = \sup_{0 \leq \Theta \leq \alpha} \left| \frac{\sin(j\beta\Theta)}{\sin(\beta\Theta)} \right| \leq j \quad (8.12)$$

Die Kombination von (8.10) und (8.11) führt auf:

$$|a_j| \leq j \frac{2}{\alpha} \int_0^\alpha g(\Theta) \sin(\beta\Theta) d\Theta$$

Mit

$$|a_1| = \frac{2}{\alpha} \int_0^\alpha g(\Theta) \sin(\beta\Theta) d\Theta$$

ergibt sich

$$|a_j| \leq j|a_1| \tag{8.13}$$

Verwenden von (8.12), (8.13), (5.17), (5.18), der Dreiecksungleichung und Eigenschaften der geometrischen Reihe führt auf:

$$|\nabla u| \geq |a_1| |\nabla \mathcal{S}_1| - \sum_{j=2}^{\infty} |a_j| |\nabla \mathcal{S}_j| \tag{8.14}$$

$$\geq |a_1| \beta r^{\beta-1} - \sum_{j=2}^{\infty} |a_1| j |\nabla \mathcal{S}_j|$$

$$\geq |a_1| \beta r^{\beta-1} - |a_1| \sum_{j=2}^{\infty} j^2 \beta r^{j\beta-1}$$

$$\geq |a_1| \beta r^{\beta-1} \left(1 - \sum_{j=2}^{\infty} j^2 (r^{j-1})^\beta \right)$$

$$\geq |a_1| \beta r^{\beta-1} \left(1 - \sum_{k=0}^{\infty} (k+2)^2 r^{k\beta} \right)$$

$$\geq |a_1| \beta r^{\beta-1} \left(1 - r^\beta \sum_{k=0}^{\infty} 2(k+1)(k+2) r^{k\beta} \right)$$

es folgt mit $r^\beta = t$ und $\beta = \frac{1}{2}$:

$$2 \sum_{k=0}^{\infty} (k+1)(k+2) t^k = 2 \frac{d^2}{dt^2} \left(\frac{1}{1-t} \right) = \frac{4}{(1-r^\beta)^3}$$

$$\geq \frac{1}{2} |a_1| r^{\beta-1} \left(1 - \frac{4r^{\frac{1}{2}}}{(1-r^{\frac{1}{2}})^3} \right)$$

Für jede positive Konstante $L > 1$ kann ein $0 \leq r_0 < 1$ so gewählt werden, sodaß

$$1 - \frac{4r_0^{\frac{1}{2}}}{(1-r_0^{\frac{1}{2}})^3} \geq \frac{1}{L} \tag{8.15}$$

erfüllt ist. Dann gilt (5.17) für alle $u(r, \Theta) \in C(\Omega)$ mit $g(\Gamma_3) \geq 0$ und $\beta \geq \frac{1}{2}$ mit $c_3 = \frac{1}{2L}$ und damit ist gezeigt, daß für diese Klasse von Funktionen Abschätzung (5.17) gültig ist.

Der Beweis, daß die Funktionen dieser Klasse auch Bedingung (5.18) erfüllen, wird in ähnlicher Form unter Verwendung von (5.23), der Dreiecksungleichung und der Bedingung $0 < r_0 < r$ folgendermaßen geführt:

$$\begin{aligned} |\nabla^{(2)}u| &\leq \sum_{j=1}^{\infty} |a_j| |\nabla^{(2)}S_j| \\ &\leq |a_1| r^{\beta-2} \sum_{j=2}^{\infty} j^3 \beta^2 (r^\beta)^{j-1} \\ &\leq |a_1| r^{\beta-2} \sum_{j=2}^{\infty} j^3 \left(r_0^{\frac{1}{2}}\right)^{j-1} \end{aligned} \quad (8.16)$$

Hier wird

$$c_4 = \sum_{j=2}^{\infty} j^3 \left(r_0^{\frac{1}{2}}\right)^{j-1} \quad (8.17)$$

gewählt. Die Reihe in (8.17) konvergiert und ist unabhängig von u und β . Damit ist Bedingung (5.18) erfüllt.

Beispiel 8.3.2

Als nächstes wird eine Teilmenge der Funktionen des vorigen Beispiels betrachtet. Für die Funktionen dieser Klasse gilt $g \in C^2(\bar{\Gamma}_3)$ mit $g''(\Theta) \leq 0$ und $g(0) = g(\alpha) = 0$. In diesem Fall ist der Spannungsintensitätsfaktor nach zweimaliger partieller Integration gegeben durch:

$$a_j = \frac{1}{j^2 \beta^2} \frac{2}{\alpha} \int_0^\alpha g''(\Theta) \sin(j\beta\Theta) d\Theta$$

Für b_j gilt analog zum vorigen Beispiel

$$b_j = \left| \frac{\sin(j\beta\Theta)}{\sin(\beta\Theta)} \right|_{C(\Gamma_3)} \leq j$$

mit $\frac{1}{2} \leq \beta < 1$.

Für den Spannungsintensitätsfaktor folgt analog zur Herleitung von (8.13):

$$|a_j| \leq \frac{1}{j^2 \beta^2} b_j |a_1| \leq \frac{4}{j} |a_1| \quad (8.18)$$

Der Nachweis, daß die in diesem Beispiel betrachteten Funktionen die Bedingung (5.17) erfüllen, wird in der gleichen Weise wie in Beispiel 8.3.1 geführt. Wird anstelle von (8.13) der Ausdruck (8.18) verwendet, so ergibt sich:

$$\begin{aligned}
 |\nabla^{(1)}u| &\geq |a_1|\beta r^{\beta-1} - \sum_{j=2}^{\infty} \frac{4}{j} |a_1| j \beta r^{j\beta-1} \\
 &\geq |a_1|\beta r^{\beta-1} \left(1 - \frac{1}{\beta^2} r^\beta \sum_{k=0}^{\infty} r^{k\beta} \right) \\
 &\geq \frac{|a_1|r^{\beta-1}}{2} \left(1 - \frac{4r_0^\beta}{(1-r_0^\beta)^3} \right)
 \end{aligned} \tag{8.19}$$

Wiederum kann für jede positive Konstante $L > 1$ ein $0 \leq r_0 < 1$ so gewählt werden, daß

$$1 + \frac{4r_0^\beta}{(1-r_0^\beta)^3} \leq \frac{1}{L} \tag{8.20}$$

erfüllt ist. Dann gilt (5.17) für alle $u(r, \Theta) \in C(\Omega)$ mit $g \in C^2(\Omega)$, $g''(\Theta) \leq 0$, $g(0) = g(\alpha) = 0$, $\beta \geq \frac{1}{2}$, mit $c_3 = \frac{1}{2L}$ und Bedingung (5.17) ist erfüllt.

Bemerkung: Für diese Klasse von Funktionen ist r_0 größer als im vorigen Beispiel.

Der Nachweis, daß die in diesem Beispiel betrachteten Klasse von Funktionen der Bedingung (5.18) genügt, folgt analog zu Beispiel 8.3.1 aus:

$$|\nabla^{(2)}u| \leq \sum_{j=1}^{\infty} |a_j| |\nabla^{(2)}\mathcal{S}_j|$$

Auf die Rechnung wird hier verzichtet und nur die Konstante c_4 angegeben.

$$c_4 = \sum_{j=2}^{\infty} j \left(r_0^{\frac{1}{2}} \right)^{j-1} \tag{8.21}$$

Die Reihe in (8.21) ist konvergent, unabhängig von u und β und damit ist gezeigt, daß diese Klasse von Funktionen die Regularitätseigenschaft (5.18) besitzt.

Beispiel 8.3.3

Im diesem Beispiel werden stärkere Regularitätsbedingungen an die Klasse der zu untersuchenden Funktionen gestellt. Es wird gezeigt, daß die Koeffizienten

$a_j, j = 2, 3, \dots$ der in der Entwicklung der Lösung u nach (8.4) mit steigenden Laufvariablen j ab einem geeigneten j vernachlässigbar klein werden.

Gegeben sei die Teilmenge der positiven harmonischen Funktionen in $C^2(\bar{\Omega}/\{0\})$ mit $u|_{\Gamma_3} \in C^4$. Dann gilt:

$$|a_j| \leq \frac{16}{j^3} |a_1|, \quad j = 2, 3, \dots \quad (8.22)$$

Der Beweis erfolgt folgendermaßen:

Zu zeigen ist, daß $g''(0) = g''(\alpha) = 0$ gilt. Mit dem Laplace-Operator in Kugelkoordinaten ergibt sich:

$$\Delta u = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \Theta^2} = 0$$

Desweiteren ergibt sich

$$\frac{\partial^2 u}{\partial r^2} \equiv \frac{\partial u}{\partial r} \equiv 0$$

da auf den Teilrändern Γ_1 und Γ_2 die Funktion u verschwindet. Schließlich ist

$$\frac{\partial^2 u}{\partial \Theta^2} = 0$$

an den Stellen $r = 1$ und $\Theta = 0, \alpha$ und somit ergibt sich $g''(0) = g''(\alpha) = 0$.

Viermalige partielle Integration von (8.10) führt auf den Ausdruck:

$$a_j = \frac{1}{j^4 \beta^4} \frac{2}{\alpha} \int_0^\alpha g^{(IV)}(\Theta) \sin(j\beta\Theta) d\Theta$$

Schließlich folgt (8.22) in der selben Art und Weise wie in den vorher gezeigten zwei Beispielen.

Die Konstanten c_3 und c_4 zum Nachweis, daß die in diesem Beispiel betrachteten Funktionen die Bedingungen 5.17 und 5.18 erfüllen, werden in der gleichen Art und Weise wie in den Beispielen 8.3.1 und 8.3.2 berechnet.

8.4 Vorbemerkungen zu den numerischen Experimenten

In diesem Abschnitt folgt eine kurze Darstellung der in den numerischen Experimenten betrachteten Fehler und einige vorbereitende Erläuterungen zur Durchführung der numerischen Experimente.

Es werden fünf numerische Experimente zum Nachweis der fast optimalen Konvergenzordnungen der relativen Fehler der Theoreme 6.3.1, 7.3.1 und 7.3.2 vorgestellt. In jedem numerischen Experiment wird die Konvergenzordnung der relativen Fehler in der $L_2(\Omega)$ -Norm auf uniform verfeinerten Gittern bestimmt und gezeigt, daß die fast optimale Konvergenzordnung $O(h^{1-\varepsilon})$ erreicht wird. Zur Einordnung und Abgrenzung von Standardaussagen zur Konvergenzordnung des Fehlers wird ebenfalls die Konvergenzordnung des Fehlers in der $L_2(\Omega)$ -Norm auf uniform verfeinerten Gittern bestimmt. Die Experimente zeigen zu erwartende nichtoptimale Konvergenzordnung.

Auf geometrisch verfeinerten Gittern werden die Konvergenzordnungen der relativen Fehler der Funktion und des Gradienten in der $L_\infty(\tau)$ -Norm bestimmt und gezeigt, daß die fast optimalen Konvergenzordnungen erreicht werden. Zusätzlich werden zum Vergleich auf uniform verfeinerten Gittern die Fehler der Funktion und des Gradienten die nichtoptimalen Konvergenzordnungen in der $L_\infty(\Omega)$ -Norm bestimmt.

Der Fehler in der $L_\infty(\Omega)$ -Norm wird als maximaler Fehler von $\|u - u_h\|_{L_\infty(\Omega)}$ auf Gittern unterschiedlicher Maschenweiten h gemessen und es zeigt sich, daß aus den Rechnungen die nach [93] zu erwartende Konvergenzordnung $O(h^\beta)$ folgt. Die Betrachtung des Fehlers in $L_\infty(\Omega)$ dient zum Vergleich der Konvergenzordnungen, die sich aus Rechnungen geometrisch verfeinerter Gitter ergeben. Die Untersuchungen in der $L_2(\Omega)$ -Norm erfolgen für den relativen Fehler des Gradienten $\|\frac{\nabla(u-u_h)}{|\nabla u} \|_{L_2(\Omega)}$ und werden auf uniform verfeinerten Gittern durchgeführt. Es ist bemerkenswert, daß im Gegensatz zu den klassischen Fehlerabschätzungen in der Energienorm (siehe Lemma 6.1.1) durch die Betrachtung des relativen Fehlers in der $L_2(\Omega)$ -Norm auf uniform verfeinerten Gittern optimale Konvergenz erzielt wird.

Die Berechnung der relativen Fehler nach den Theoreme 7.3.1 und 7.3.2 erfolgt durch Bestimmung des maximalen relativen Fehlers der Funktion und des Gra-

dienten in der $L_\infty(\tau)$ -Norm. Dabei ist der relative Fehler als punktweise Division zweier Funktionen zu verstehen. Aus den Theoremen 7.3.1 und 7.3.2 folgt, daß fast optimale Konvergenz für beide Fehler innerhalb gewisser Schranken möglich ist. Deshalb wird der Radius r_0 eingeführt, der ausdrückt, bis zu welcher Entfernung zur Singularität fast optimale Konvergenzordnungen erzielt werden können. Fast optimale Konvergenzordnungen sind somit nur im Intervall $\rho < d \leq r_0 \leq 1$ für den relativen Fehler des Gradienten und im Intervall $\rho < \rho h^{1-\frac{1}{\beta}} < d \leq r_0 \leq 1$ für den relativen Fehler der Funktion zu erwarten. Die Begründung hierfür ist darin zu finden, daß die Entwicklung (8.4) der Lösung nicht auf dem gesamten Gebiet Ω gültig ist, sondern nur innerhalb einer Umgebung um den Ursprung mit Radius $r \leq r_0$. Aus den in den nachfolgenden numerischen Experimenten berechneten relativen Fehlern des Gradienten und der Funktion kann r_0 abgeschätzt werden und liegt bei $r_0 \leq 0.125$ für den relativen Fehler des Gradienten und bei $r_0 \leq 0.25$ für den relativen Fehler der Funktion.

Außerdem beeinflussen "pollution"-Effekte die Lösung im Innern des Gebietes und bewirken ein Anwachsen des Fehlers. Diese Effekte können so dominant sein, daß die erste Singularitätenfunktion nicht mehr das Verhalten der Lösung bestimmt und die optimale Konvergenzordnung verloren geht.

In den Auswertungen der verschiedenen numerischen Experimente wird der maximale relative Fehler des Gradienten im Intervall $\rho \leq d \leq 0.125$ und der maximale relative Fehler der Funktion im Intervall $\rho < \rho h^{1-\frac{1}{\beta}} < d \leq 0.25$ betrachtet. Desweiteren beinhaltet die Berechnung des relativen Fehlers eine weitere prinzipielle Schwierigkeit. Die Lösung einer elliptischen partiellen Differentialgleichung kann im Rechengebiet zu Null werden und als Konsequenz geht der relative Fehler gegen Unendlich. Um dies zu vermeiden, wird deshalb der maximale Fehler der Funktion $|u - u_h|_{max}$ und der maximale Wert der Lösung $|u|_{max}$ entlang des Winkels Θ bei gegebener Maschenweite h bestimmt und der Quotient $\frac{|u - u_h|_{max}}{|u|_{max}}$ berechnet.

8.5 Numerisches Experiment 1

In diesem numerischen Experiment wird das Randwertproblem so gestellt, daß die Lösung des Randwertproblems durch die 1. Singularitätenfunktion im gesamten Rechengebiet bestimmt wird und keine Störeinflüsse anderer Terme der Entwicklung der Lösung auftreten. Das untersuchte Dirichlet-Problem ist gegeben durch:

$$\begin{aligned} -\Delta u &= 0 & \text{in } \Omega \\ u &= g & \text{auf } \partial\Omega \end{aligned} \quad (8.23)$$

Die Funktion g wird so gewählt, daß die Lösung von (8.23) die folgende Gestalt Form besitzt:

$$u(r, \Theta) = |a_1| \mathcal{S}_1$$

In (8.23) bezeichnet $|a_1|$ den ersten Spannungsintensitätsfaktor und \mathcal{S}_1 die erste Singularitätenfunktion. Der Gradient der Lösung von (8.23) erfüllt die in Kapitel 5 Abschnitt 5.3 dargestellten Voraussetzungen (5.17) und (5.18). In diesem numerischen Experiment haben die Voraussetzungen (5.17) und (5.18) die folgende Gestalt:

$$\begin{aligned} \beta |a_1| r^{\beta-1} &\leq |\nabla u| \\ \left| \nabla^{(2)} u \right| &\leq 2\beta |\beta - 1| |a_1| r^{\beta-2} \end{aligned}$$

Damit sind alle Bedingungen der Theoreme 6.3.1, 7.3.1 und 7.3.2 gegeben.

In den hier durchgeführten numerischen Experimenten wird für den Spannungsintensitätsfaktor $a_1 = 10$ gewählt. Ferner gilt $u \in C^2(\Omega) \cup C^0(\partial\Omega)$, $\partial\Omega = \Gamma_1 + \Gamma_2 + \Gamma_3$ (siehe Abbildung 5.1 in Kapitel 5) mit $g = 0$ auf Γ_1 und Γ_2 und $g \neq 0$ auf Γ_3 .

Die exakte Lösung des Randwertproblems auf dem Gebiet der geschlitzten Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$ ist gegeben durch:

$$\begin{aligned} u &= 10r^{\frac{1}{2}} \sin \frac{1}{2}\Theta \\ g &= 10 \sin \frac{1}{2}\Theta \end{aligned} \quad (8.24)$$

Der Graph der Lösung u in kartesischen Koordinaten auf dem Gebiet der geschlitzten Kreisscheibe ist in Abbildung 8.4 und als Höhenliniendarstellung in

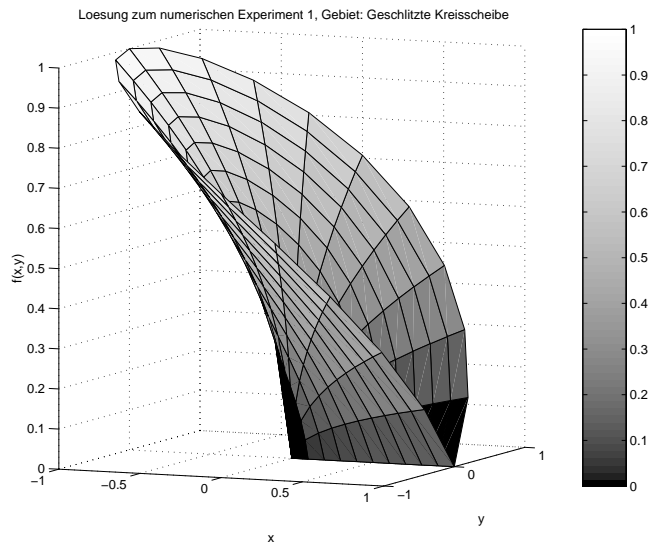


Abbildung 8.4: Experiment 1: Lösung u des Problems (8.23), Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$

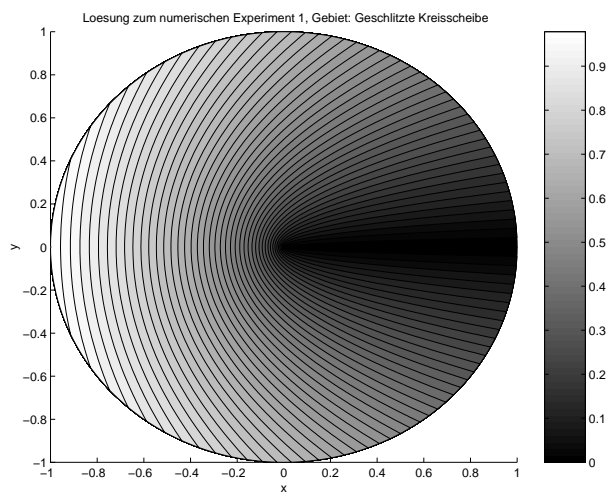


Abbildung 8.5: Experiment 1: Lösung u des Problems (8.23) als Höhenliniendarstellung, Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$

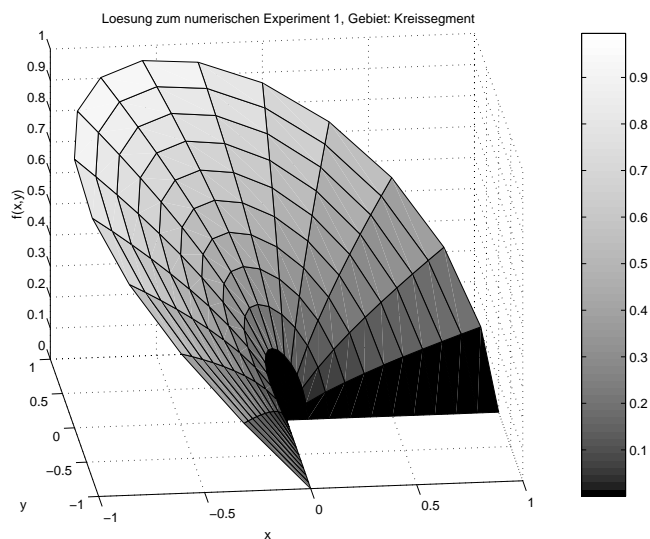


Abbildung 8.6: Experiment 1: Lösung u des Problems (8.23), Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi$, $\beta = \frac{2}{3}$

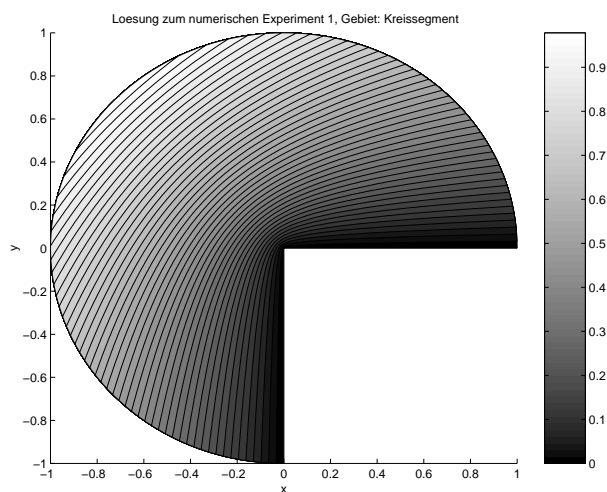


Abbildung 8.7: Experiment 1: Lösung u des Problems (8.23) als Höhenliniendarstellung, Gebiet: Geschnittene Kreisscheibe mit $\alpha = \frac{3}{2}\pi$, $\beta = \frac{2}{3}$

Numerisches Experiment 1			
Relativer Fehler des Gradienten in $L_2(\Omega)$			
Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi, \beta = \frac{1}{2}$			
h	# Knoten	relativer Fehler in $L_2(\Omega)$	Konvergenzordnung
0.1	600	0.1399	
0.05	2480	$0.7251 \cdot 10^{-1}$	0.95
0.025	9920	$0.3719 \cdot 10^{-1}$	0.96
0.0125	40000	$0.1899 \cdot 10^{-1}$	0.97
Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$			
h	# Knoten	relativer Fehler in $L_2(\Omega)$	Konvergenzordnung
0.1	440	0.1103	
0.05	1840	$0.5685 \cdot 10^{-1}$	0.96
0.025	7520	$0.2887 \cdot 10^{-1}$	0.99
0.0125	30080	$0.1460 \cdot 10^{-2}$	0.98

Tabelle 8.1: Konvergenzordnung der relativen Fehler des Gradienten des Problems $-\Delta u = 0$ auf Ω , $g = \sin \beta \Theta$ mit der Lösung $u = 10r^\beta \sin(\beta \Theta)$ in der $L_2(\Omega)$ -Norm bei uniformer Verfeinerung

Abbildung 8.5 dargestellt.

Weiter ergibt sich als Lösung des Kreissegment mit $\alpha = \frac{3}{2}\pi$:

$$\begin{aligned}
 u &= 10r^{\frac{2}{3}} \sin \frac{2}{3} \Theta & (8.25) \\
 u &= 10 \sin \frac{2}{3} \Theta
 \end{aligned}$$

Die Lösung u (8.25) wird in den Abbildungen 8.6 und 8.7 dargestellt.

Relativer Fehler in $L_2(\Omega)$

Aus Theorem 6.3.1 folgt, daß der relative Fehler des Gradienten auf einem uniformen Gitter mit der Maschenweite h die Konvergenzordnung $O(h)$ besitzt. In Tabelle 8.1 wird der berechnete relative Fehler des Gradienten bei verschiedenen Maschenweiten h dargestellt. Der relative Fehler des Gradienten gibt an, wie groß der prozentuale Fehler der Energie im betrachteten Rechengebiet ist. Bei einer Maschenweite von $h = 0.1$ beträgt der prozentuale Fehler ca. 14% der Gesamtenergie. Die Konvergenzordnung erreicht für diese Rechnung mit 0.95 nicht ganz den optimalen Wert für $O(h)$, da ein Gitter mit dieser Maschenweite noch nicht

in den asymptotischen Bereich führt. Die Rechnungen auf Gittern, bei denen die Maschenweite jeweils halbiert wurde, zeigen jedoch die in Theorem 6.3.1 zu erwartenden fast optimalen Konvergenzordnungen von $O(h^{1-\varepsilon})$ (vergl. Tabelle 8.1).

Relativer Fehler in $L_\infty(\tau)$

Die numerischen Experimente zur Bestätigung der fast optimalen Konvergenzordnung der relativen Fehler der Funktion und ihres Gradienten werden auf den gleichen Gebieten wie in Unterabschnitt 8.5 mit den Maschenweiten $h = 0.1$ und $h = 0.05$ am Rand der Gebiete durchgeführt. Es werden die relativen Fehler auf geometrischen Gittern mit Maschenweiten von $h = 0.1$ und $h = 0.05$ mit Verfeinerungsabständen von $\rho = h^2$ bzw. $\rho = h^3$ durchgeführt.

In Tabelle 8.2 werden die Konvergenzordnungen des relativen Fehlers der Funktion dargestellt. Für beide Gebiete werden die nach Theorem 7.3.2 zu erwartende fast optimale Konvergenzordnungen $O(h^{2-\varepsilon})$, $\varepsilon > 0$ erreicht. Die Konvergenzordnung des relativen Fehlers der Funktion auf dem Kreissegment ist im Vergleich zu der Konvergenzordnung auf der geschlitzten Kreisscheibe geringfügig kleiner. Die geringfügigen Abweichungen werden durch den "pollution"-Effekt verursacht. In Tabelle 8.3 werden die Ergebnisse der Berechnungen des relativen Fehlers des Gradienten auf geometrisch verfeinerten Gittern zur Bestätigung von Theorem 7.3.1 vorgestellt. Die Konvergenzordnungen der numerischen Experimente zeigen eine gute Übereinstimmung der zu erwartenden Konvergenzordnung $O(h^{1-\varepsilon})$, $\varepsilon > 0$. Es ist bemerkenswert, daß der relative Fehler bereits bei einer Maschenweite von $h = 0.1$ im Promillebereich liegt. Auf uniformen Gittern besitzt nach einer Arbeit von Schatz und Wahlbin [93] der Fehler der Funktion in der $L_\infty(\Omega)$ -Norm die nichtoptimale Konvergenzordnung $O(h^\beta)$. In weiteren Rechnungen auf uniformen Gittern wird dieses bekannte Resultat mit dem relativen Fehler der Funktion auf uniformen Gittern in Tabelle 8.4 dargestellt. Es ist ersichtlich, daß erst bei kleinen Maschenweiten die Konvergenzordnung $O(h^\beta)$ erzielt werden kann. Erst durch eine Maschenweite von $h = 0.125$ und einem Gitter mit 40000 Knoten kann die Konvergenzordnung $O(h^\beta)$ erzielt werden kann. Der Fehler beträgt in diesem Fall $0.239 \cdot 10^{-1}$ und ist größer als die Werte der relativen Fehler der Funktion auf geometrisch verfeinerten Gittern.

Numerisches Experiment 1				
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	3120	$0.190 \cdot 10^{-2}$	1.90
0.05	$0.25 \cdot 10^{-2}$	16864	$0.510 \cdot 10^{-3}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	4440	$0.280 \cdot 10^{-2}$	2.03
0.05	$0.125 \cdot 10^{-3}$	24056	$0.685 \cdot 10^{-3}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	2288	$0.180 \cdot 10^{-2}$	1.86
0.05	$0.25 \cdot 10^{-2}$	12512	$0.495 \cdot 10^{-3}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	3256	$0.258 \cdot 10^{-2}$	1.87
0.05	$0.125 \cdot 10^{-3}$	17848	$0.7065 \cdot 10^{-3}$	

Tabelle 8.2: Konvergenzordnungen der relativen Fehler der Funktion des numerischen Experiments 1 mit $\rho h^{1-\frac{1}{\beta}} \leq d \leq 0.25$ bei geometrischer Verfeinerung

Numerisches Experiment 1				
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	3120	$0.363 \cdot 10^{-1}$	1.05
0.05	$0.25 \cdot 10^{-2}$	16864	$0.180 \cdot 10^{-1}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	4440	$0.370 \cdot 10^{-1}$	1.09
0.05	$0.125 \cdot 10^{-3}$	24056	$0.180 \cdot 10^{-1}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	2288	$0.247 \cdot 10^{-1}$	1.04
0.05	$0.25 \cdot 10^{-2}$	12512	$0.120 \cdot 10^{-1}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	3256	$0.248 \cdot 10^{-1}$	1.06
0.05	$0.125 \cdot 10^{-3}$	17848	$0.119 \cdot 10^{-1}$	

Tabelle 8.3: Konvergenzordnungen der relativen Fehler des Gradienten des numerischen Experiments 1 mit $\rho \leq d \leq 0.125$ bei geometrischer Verfeinerung

Numerisches Experiment 1			
Fehler in der $L_\infty(\Omega)$ -Norm			
<i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>			
h	Anzahl Knoten	Fehler in $L_\infty(\Omega)$	Konvergenzordnung
0.1	600	0.324	0.43
0.05	2480	0.241	
0.025	9920	0.174	0.47
0.0125	40000	0.124	0.49
Fehler in der $L_\infty(\Omega)$ -Norm			
<i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>			
h	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	600	$0.919 \cdot 10^{-1}$	0.63
0.05	2480	$0.594 \cdot 10^{-1}$	
0.025	9920	$0.378 \cdot 10^{-1}$	0.65
0.0125	40000	$0.239 \cdot 10^{-1}$	0.66

Tabelle 8.4: Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm bei uniformer Verfeinerung

8.6 Numerisches Experiment 2

Die numerischen Untersuchungen werden mit Funktionen aus der Funktionenklasse 8.3.2 vorgenommen. Für diese Funktionen sind die Spannungsintensitätsfaktoren durch $|a_j| \leq \frac{4}{j}|a_1|, j = 2, 3, \dots$ gegeben. Damit ist gewährleistet, daß die 1. Singularitätenfunktion der größte Term in (8.4) ist und das Verhalten der Lösung bestimmt. Außerdem gelten die Ungleichungen (5.17) und (5.18). Die Randwertfunktion g zeigt im gesamten Definitionsbereich ein konvexes Krümmungsverhalten, so daß die Forderung, daß $g''(\Theta)$ entweder nur positive oder nur negative Werte annehmen darf, gegeben ist. Somit sind alle Voraussetzungen für die Gültigkeit der Theoreme 6.3.1, 7.3.1 und 7.3.2 erfüllt. Den Rechnungen wird als Lösung u die Reihe (8.4) bis $j = 11$ zugrundegelegt. Die Beiträge der nachfolgenden Reihenglieder können als vernachlässigbar klein im Vergleich zu $|a_1|$ angesehen werden.

Das Randwertproblem ist gegeben durch:

$$\begin{aligned} -\Delta u &= 0 & \text{in } \Omega \\ u &= g & \text{auf } \partial\Omega \end{aligned} \quad (8.26)$$

Die exakte Lösung u und die Randwerte g für die geschlitzte Kreisscheibe mit $\alpha = 2\pi, \beta = \frac{1}{2}$ sind gegeben durch:

$$u = \sum_{j=1}^{11} \frac{10}{j} r^{\frac{1}{2}j} \sin \frac{1}{2}j\Theta \quad (8.27)$$

$$g = \sum_{j=1}^{11} \frac{10}{j} \sin \frac{1}{2}j\Theta \quad (8.28)$$

Eine Darstellung der Funktion (8.26) ist in Abbildung 8.8 und in Abbildung 8.9 als Höhenliniendarstellung zu finden.

Das Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$ besitzt die exakte Lösung und die Randwertfunktion g in der Form:

$$u = \sum_{j=1}^{11} \frac{10}{j} r^{\frac{2}{3}j} \sin \frac{2}{3}j\Theta \quad (8.29)$$

$$g = \sum_{j=1}^{11} \frac{10}{j} \sin \frac{2}{3}j\Theta \quad (8.30)$$

In Abbildung 8.10 wird die Funktion (8.26) dargestellt. Abbildung 8.11 zeigt eine Höhenliniendarstellung der Funktion (8.26).

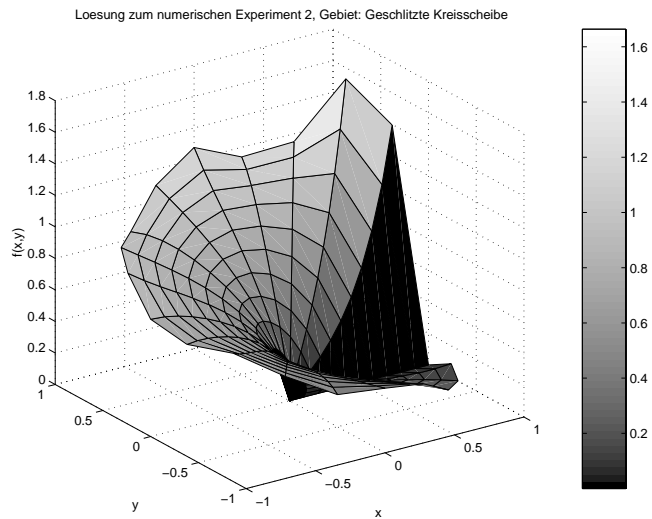


Abbildung 8.8: Experiment 2: Lösung u des Problems (8.26), Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$

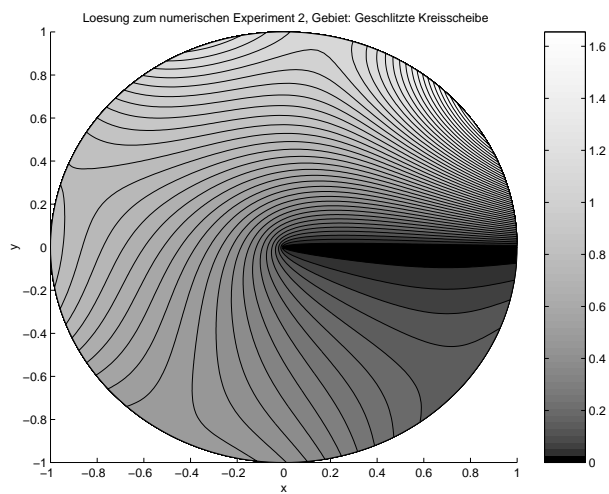


Abbildung 8.9: Experiment 2: Lösung u des Problems (8.26) als Höhenliniendarstellung, Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$

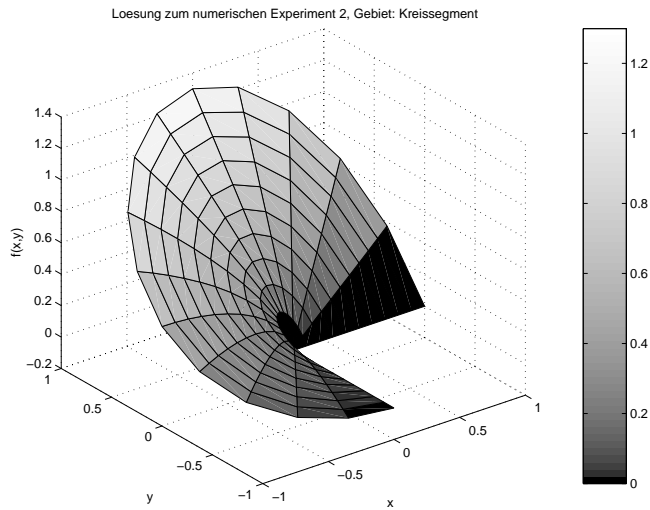


Abbildung 8.10: Experiment 2: Lösung u des Problems (8.26),
Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi$, $\beta = \frac{2}{3}$

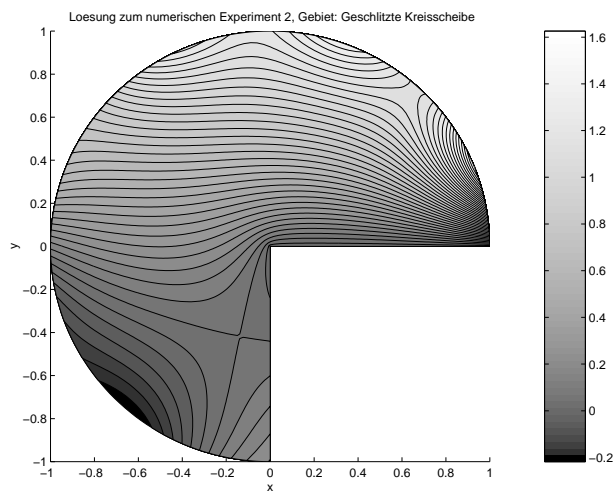


Abbildung 8.11: Experiment 2: Lösung u des Problems (8.26)
als Höhenliniendarstellung, Gebiet: Geschlitzte Kreisscheibe mit
 $\alpha = \frac{3}{2}\pi$, $\beta = \frac{2}{3}$

Numerisches Experiment 2			
Relativer Fehler des Gradienten in $L_2(\Omega)$			
Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi, \beta = \frac{1}{2}$			
h	# Knoten	Fehler in $L_2(\Omega)$	Konvergenzordnung
0.128	288	0.261	
0.064	1344	0.1418	0.88
0.032	5880	$0.816 \cdot 10^{-1}$	0.81
0.016	24304	$0.458 \cdot 10^{-1}$	0.83
0.008	97216	$0.255 \cdot 10^{-1}$	0.85
Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$			
h	# Knoten	Fehler in $L_2(\Omega)$	Konvergenzordnung
0.128	216	0.2095	
0.064	1008	$0.1078 \cdot 10^{-1}$	0.96
0.032	4320	$0.570 \cdot 10^{-1}$	0.92
0.016	18104	$0.2964 \cdot 10^{-1}$	0.94
0.008	72912	$0.1539 \cdot 10^{-1}$	0.95

Tabelle 8.5: Konvergenzordnung der relativen Fehler des Gradienten des Problems $-\Delta u = 0$ auf Ω , $g = \sum_{j=1}^{11} \frac{10}{j} \sin \frac{1}{2} j \Theta$ mit der Lösung $u = 10r^\beta \sin(\frac{1}{2}\Theta)$ in der $L_2(\Omega)$ -Norm bei uniformer Verfeinerung

Relativer Fehler in $L_2(\Omega)$

Die berechneten Konvergenzordnungen des relativen Fehlers des Gradienten in der $L_2(\Omega)$ -Norm auf unverfeinerten Gittern mit der Maschenweite h sind in Tabelle 8.5 dargestellt und zeigen die nach Theorem 6.3.1 zu erwartende optimale Ordnung $O(h^{1-\varepsilon})$, $\varepsilon > 0$. Eine Verkleinerung der Maschenweite h bewirkt eine stetig verbesserte Konvergenzordnung. Es ist bemerkenswert, wie wenig sich die relativen Fehler der Rechnungen auf den verschiedenen Gebieten unterscheiden. Der relative Fehler der Funktion auf der geschlitzten Kreisscheibe zeigt im Vergleich zum relativen Fehler der Funktion auf dem Kreissegment eine kleinere Konvergenzordnung. Die Singularität der geschlitzten Kreisscheibe ist aufgrund des Exponenten β stärker als auf dem Kreissegment und deshalb ist bei gleicher Maschenweite die Konvergenzordnung des relativen Fehlers des Gradienten der geschlitzten Kreisscheibe kleiner als im Falle des Kreissegments. Anhand der Konvergenzraten in Tabelle 8.5 sind auch "pollution"-Effekte zu sehen.

Relativer Fehler in L_∞

Die numerischen Experimente zur Bestätigung der Theoreme 7.3.1 und 7.3.2 werden auf geometrischen Gittern mit den Maschenweiten $h = 0.1$ und $h = 0.05$ bei

Numerisches Experiment 2				
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	3120	$0.204 \cdot 10^{-2}$	1.94
0.05	$0.25 \cdot 10^{-2}$	16864	$0.532 \cdot 10^{-3}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	4440	$0.281 \cdot 10^{-2}$	2.0
0.05	$0.125 \cdot 10^{-3}$	24056	$0.700 \cdot 10^{-3}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	2288	$0.184 \cdot 10^{-2}$	1.9
0.05	$0.25 \cdot 10^{-2}$	12512	$0.495 \cdot 10^{-3}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	3256	$0.268 \cdot 10^{-2}$	1.97
0.05	$0.125 \cdot 10^{-3}$	17848	$0.685 \cdot 10^{-3}$	

Tabelle 8.6: Konvergenzordnungen der relativen Fehler der Funktion des numerischen Experiments 2 mit $\rho h^{1-\frac{1}{\beta}} \leq d \leq 0.25$ bei geometrischer Verfeinerung

Verfeinerungsabständen $\rho = h^2$ bzw. $\rho = h^3$ durchgeführt. In Tabelle 8.6 werden die relativen Fehler der Funktion dargestellt. Die berechneten Konvergenzordnungen stimmen gut mit der aus Theorem 7.3.2 zu erwartenden fast optimalen Konvergenzordnung $O(h^{2-\varepsilon})$, $\varepsilon > 0$ überein. Es ist bemerkenswert, wie klein der relative Fehler auf beiden Gebieten ist. Die Konvergenzordnungen des relativen Fehlers des Gradienten werden auf den gleichen geometrischen Gittern, die auch zur Bestimmung des relativen Fehlers der Funktion verwendet worden sind, vorgenommen und in Tabelle 8.8 dargestellt. Die berechneten Konvergenzordnungen bestätigen die nach Theorem 7.3.1 zu erwartenden fast optimalen Konvergenzordnungen $O(h^{1-\varepsilon})$, $\varepsilon > 0$.

Numerisches Experiment 2				
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm				
<i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	3120	$0.495 \cdot 10^{-1}$	1.04
0.05	$0.25 \cdot 10^{-2}$	16864	$0.240 \cdot 10^{-1}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm				
<i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	2288	$0.281 \cdot 10^{-1}$	1.13
0.05	$0.25 \cdot 10^{-2}$	12512	$0.128 \cdot 10^{-1}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm				
<i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	3256	$0.243 \cdot 10^{-1}$	1.04
0.05	$0.125 \cdot 10^{-3}$	17848	$0.118 \cdot 10^{-1}$	

Tabelle 8.7: Konvergenzordnungen der relativen Fehler des Gradienten des numerischen Experiments 2 mit $\rho \leq d \leq 0.125$ bei geometrischer Verfeinerung

In Tabelle 8.8 werden die nichtoptimalen Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm auf unverfeinerten Gittern vorgestellt. Nach [93] ist eine Konvergenzordnung von $O(h^{\beta-\varepsilon})$, $\varepsilon > 0$ zu erwarten. Die Rechnungen bestätigen die Aussagen nach [93]. Im Vergleich zum relativen Fehler der Funktion (Tabelle 8.7) sind die Werte des Fehlers in Tabelle 8.8 deutlich höher.

Numerisches Experiment 2			
Fehler in der $L_\infty(\Omega)$ -Norm			
<i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi$, $\beta = \frac{1}{2}$</i>			
h	Anzahl Knoten	Fehler in $L_\infty(\Omega)$	Konvergenzordnung
0.128	288	0.3893	0.47
0.064	1344	0.2817	
0.032	5880	0.1996	0.49
0.016	24304	0.1408	0.5
0.008	97216	0.1	0.49
Fehler in der $L_\infty(\Omega)$ -Norm			
<i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi$, $\beta = \frac{2}{3}$</i>			
h	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.128	216	0.1234	0.72
0.064	1008	$0.746 \cdot 10^{-1}$	
0.032	4320	$0.457 \cdot 10^{-1}$	0.70
0.016	18104	$0.283 \cdot 10^{-1}$	0.69
0.008	72912	$0.1786 \cdot 10^{-1}$	0.66

Tabelle 8.8: Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm bei uniformer Verfeinerung

8.7 Numerisches Experiment 3

In diesem Experiment wird die Funktionenklasse aus Beispiel 8.3.3 untersucht. In Beispiel 8.3.3 wird aufgezeigt, daß bei entsprechenden Regularitätseigenschaften der Lösung die Spannungsintensitätsfaktoren durch $|a_j| = \frac{16}{j^3}|a_1|$ gegeben sind. Die Voraussetzung, daß die 1. Singularitätenfunktion das Lösungsverhalten bestimmt, ist damit erfüllt. Außerdem wurde in Beispiel 8.3.3 gezeigt, daß für Funktionen dieser Funktionenklasse die Ungleichungen (5.17) und (5.18) gelten. Die Randwertfunktion g zeigt ebenfalls die geforderte Bedingung $g''(\Theta) \geq 0$. Damit sind die Voraussetzungen für die Theoreme 6.3.1, 7.3.1 und 7.3.2 gegeben. Den Rechnungen wird als Lösung u die Reihe (8.4) bis zum 11. Glied zugrundegelegt. Die Verwendung der Reihe (8.4) bis zum 11. Glied ist ausreichend, da die Beiträge der folgenden Glieder bezüglich $|a_1|$ vernachlässigbar klein werden.

Das untersuchte Randwertproblem lautet:

$$\begin{aligned} -\Delta u &= 0 & \text{in } \Omega \\ u &= g & \text{auf } \partial\Omega \end{aligned} \quad (8.31)$$

Als Lösung für die geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$ wird der folgende Ausdruck verwendet:

$$\begin{aligned} u &= \sum_{j=1}^{11} \frac{16}{j^3} r^{\frac{1}{2}j} \sin \frac{1}{2}j\Theta \\ g &= \sum_{j=1}^{11} \frac{16}{j^3} \sin \frac{1}{2}j\Theta \end{aligned} \quad (8.32)$$

Die Funktion u aus (8.31) ist zur Veranschaulichung in den Abbildungen 8.12 und 8.13 dargestellt. Analog ergibt sich für das Kreissegment mit $\alpha = \frac{3}{2}\pi$, $\beta = \frac{2}{3}$ als Lösung des Randwertproblems:

$$\begin{aligned} u &= \sum_{j=1}^{11} \frac{16}{j^3} r^{\frac{2}{3}j} \sin \frac{2}{3}j\Theta \\ g &= \sum_{j=1}^{11} \frac{16}{j^3} \sin \frac{2}{3}j\Theta \end{aligned} \quad (8.33)$$

Die Darstellung der Funktion u aus (8.33) wird in den Abbildungen 8.14 und 8.15 gegeben.

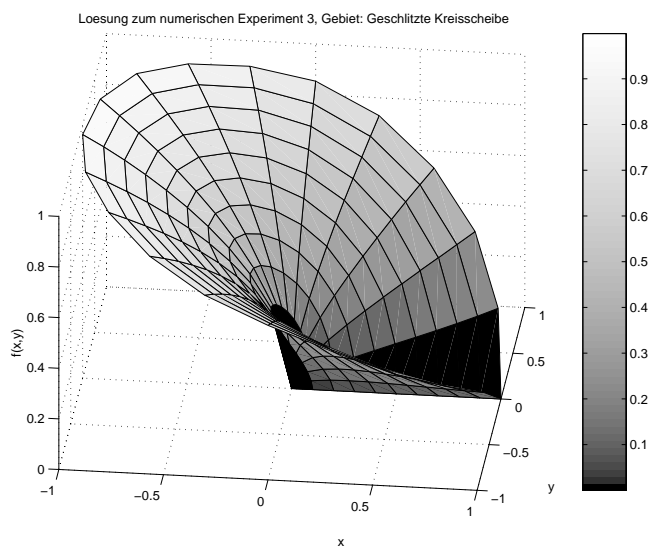


Abbildung 8.12: Experiment 3: Lösung u des Problems (8.31),
Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$

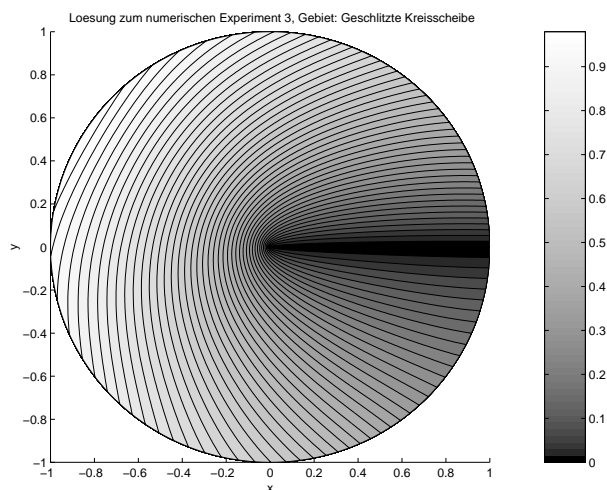


Abbildung 8.13: Experiment 3: Lösung u des Problems (8.31)
als Höhenliniendarstellung, Gebiet: Geschlitzte Kreisscheibe mit
 $\alpha = 2\pi$, $\beta = \frac{1}{2}$

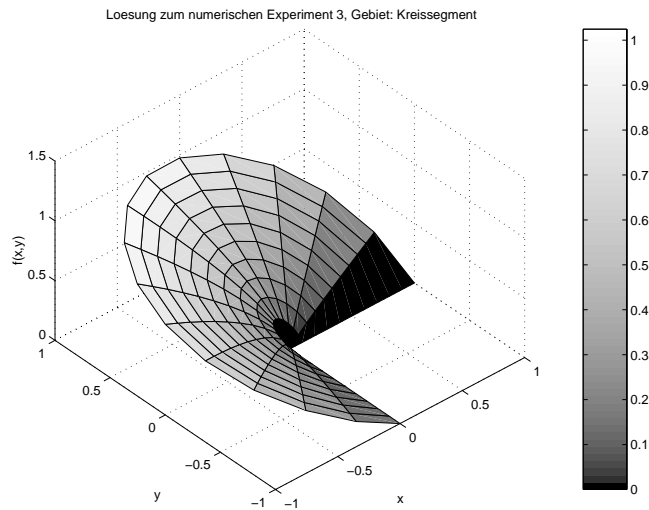


Abbildung 8.14: Experiment 3: Lösung u des Problems (8.31),
Gebiet: Kreissegment mit $\alpha = \frac{3}{2\pi}$, $\beta = \frac{2}{3}$

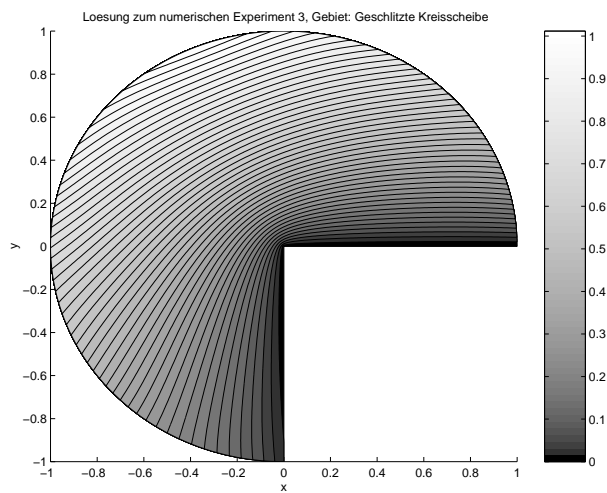


Abbildung 8.15: Experiment 3: Lösung u des Problems (8.31)
als Höhenliniendarstellung, Gebiet: Geschlitzte Kreisscheibe mit
 $\alpha = \frac{3}{2\pi}$, $\beta = \frac{2}{3}$

Numerisches Experiment 3			
Relativer Fehler des Gradienten in $L_2(\Omega)$			
Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi, \beta = \frac{1}{2}$			
h	# Knoten	Fehler in $L_2(\Omega)$	Konvergenzordnung
0.1	600	0.1429	
0.05	2480	$0.7401 \cdot 10^{-1}$	0.95
0.025	9920	$0.3794 \cdot 10^{-1}$	0.96
0.0125	40000	$0.1936 \cdot 10^{-1}$	0.97
Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$			
h	# Knoten	Fehler in $L_2(\Omega)$	Konvergenzordnung
0.1	440	0.1122	
0.05	1840	$0.5787 \cdot 10^{-1}$	0.96
0.025	7520	$0.2939 \cdot 10^{-1}$	0.98
0.0125	30080	$0.1487 \cdot 10^{-1}$	0.98

Tabelle 8.9: Konvergenzordnungen der relativen Fehler des Gradienten des Problems $-\Delta u = 0$ auf Ω , $g = \sum_{j=1}^{11} \frac{16}{j^3} \sin \frac{1}{2} j \Theta$ mit der Lösung $u = \sum_{j=1}^{11} \frac{16}{j^3} r^{\beta j} \sin(\beta j \Theta)$ in der $L_2(\Omega)$ -Norm bei uniformer Verfeinerung

Relativer Fehler in $L_2(\Omega)$

Die berechneten Werte für den relativen Fehler des Gradienten in $L_2(\Omega)$ auf unverfeinerten Gittern sind in Tabelle 8.9 dargestellt. Die numerisch ermittelten Werte für den relativen Fehler des Gradienten wird bei Halbierung der Maschenweite um den Faktor 2 kleiner und zeigt das nach Theorem 6.3.1 zu erwartende optimale Konvergenzverhalten von $O(h^{1-\varepsilon})$, $\varepsilon > 0$. Bei Maschenweite $h = 0.1$ beträgt die Konvergenzordnung 0.95 und liegt damit geringfügig unter dem zu erwarteten Wert von $O(h^{1-\varepsilon})$, $\varepsilon > 0$. Es ist jedoch aus den Werten in Tabelle 8.9 ersichtlich, daß mit kleiner werdender Maschenweite und der damit verbundenen Annäherung an den asymptotischen Bereich fast optimales Konvergenzverhalten erzielt wird. Auch in diesem numerischen Experiment unterscheiden sich die relativen Fehler des Gradienten bei gleicher Maschenweite aber unterschiedlichem Rechengebiet nur gering. Es ist ebenfalls bemerkenswert, daß sich die Werte der relativen Fehler des Gradienten dieses numerischen Experimentes von denen in Experiment 2 und Experiment 1 berechneten bei kleinen Maschenweiten nur gering unterscheiden.

Numerisches Experiment 3				
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	3120	$0.214 \cdot 10^{-2}$	2.03
0.05	$0.25 \cdot 10^{-2}$	16864	$0.522 \cdot 10^{-3}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	4440	$0.281 \cdot 10^{-2}$	1.98
0.05	$0.125 \cdot 10^{-3}$	24056	$0.710 \cdot 10^{-3}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	2288	$0.184 \cdot 10^{-2}$	1.9
0.05	$0.25 \cdot 10^{-2}$	12512	$0.495 \cdot 10^{-3}$	
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	3256	$0.260 \cdot 10^{-2}$	1.9
0.05	$0.125 \cdot 10^{-3}$	17848	$0.702 \cdot 10^{-3}$	

Tabelle 8.10: Konvergenzordnungen der relativen Fehler der Funktion des numerischen Experiments 3 mit $\rho h^{1-\frac{1}{\beta}} \leq d \leq 0.25$ bei geometrischer Verfeinerung

Relativer Fehler in L_∞

In den Tabellen 8.10 und 8.11 werden die Ergebnisse der numerischen Experimente zur Bestätigung der Theoreme 7.3.1 und 7.3.2 dargestellt. Die Berechnungen der Konvergenzordnungen wurde auf geometrisch verfeinerten Gittern mit den Maschenweiten $h = 0.1$ und $h = 0.05$ mit Verfeinerungsabständen $\rho = h^2$ bzw. $\rho = h^3$ durchgeführt. Die berechneten Konvergenzordnungen des relativen Fehlers der Funktion stimmen gut mit den fast optimalen Konvergenzordnungen $O(h^{2-\varepsilon})$, $\varepsilon > 0$ aus Theorem 7.3.1 überein (siehe Tabelle 8.10). Der Einfluß der "pollution" auf die Konvergenzordnung ist in den Rechnungen auf dem Kreissegment zu sehen. Die Konvergenzraten erreichen hier den Wert $O(h^{1.9})$ und liegen

Numerisches Experiment 3				
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	3120	$0.372 \cdot 10^{-1}$	1.08
0.05	$0.25 \cdot 10^{-2}$	16864	$0.176 \cdot 10^{-1}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	4440	$0.385 \cdot 10^{-1}$	1.01
0.05	$0.125 \cdot 10^{-3}$	24056	$0.192 \cdot 10^{-1}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	2288	$0.385 \cdot 10^{-1}$	1.01
0.05	$0.25 \cdot 10^{-2}$	12512	$0.192 \cdot 10^{-1}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	3256	$0.393 \cdot 10^{-1}$	1.04
0.05	$0.125 \cdot 10^{-3}$	17848	$0.191 \cdot 10^{-1}$	

Tabelle 8.11: Konvergenzordnungen der relativen Fehler des Gradienten des numerischen Experiments 3 mit $\rho \leq d \leq 0.125$ bei geometrischer Verfeinerung

damit geringfügig unter der fast optimalen Rate. Tabelle 8.11 zeigt die berechneten Konvergenzordnungen des relativen Fehlers des Gradienten. Die Konvergenzordnungen zeigen eine gute Übereinstimmung mit dem aus Theorem 7.3.2 zu erwartenden fast optimalen Wert $O(h^{1-\varepsilon})$, $\varepsilon > 0$. Zum Vergleich der Konvergenzraten auf unverfeinerten Gittern werden die Fehler in der $L_\infty(\Omega)$ -Norm in Tabelle 8.12 vorgestellt. Nach [93] beträgt die Konvergenzordnung $O(h^\beta)$. Die Werte in Tabelle 8.12 zeigen die nach [93] zu erwartende Konvergenzordnung.

Numerisches Experiment 3			
Fehler in der $L_\infty(\Omega)$ -Norm			
<i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>			
h	Anzahl Knoten	Fehler in $L_\infty(\Omega)$	Konvergenzordnung
0.1	600	0.324	0.43
0.05	2480	0.241	
0.025	9920	0.174	0.47
0.0125	40000	0.124	0.49
Fehler in der $L_\infty(\Omega)$ -Norm			
<i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>			
h	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	600	$0.9194 \cdot 10^{-1}$	0.63
0.05	2480	$0.5943 \cdot 10^{-1}$	
0.025	9920	$0.3782 \cdot 10^{-1}$	0.65
0.0125	40000	$0.2391 \cdot 10^{-1}$	0.66

Tabelle 8.12: Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm bei uniformer Verfeinerung

8.8 Numerisches Experiment 4

In diesem Experiment wird das Torsionsproblem als Beispiel für ein Randwertproblem mit rechter Seite $f \neq 0$ und verschwindenden Randwerten untersucht. Das Randwertproblem ist in der folgenden Form gegeben:

$$\begin{aligned} -\Delta u &= -20 && \text{in } \Omega \\ u &= 0 && \text{auf } \partial\Omega \end{aligned} \quad (8.34)$$

Die Lösung u wird in den Rechnungen nach folgender Reihe bestimmt:

$$u(r, \Theta) = \sum_{j=1}^{11} a_j S_j \quad (8.35)$$

In (8.35) bezeichnet a_j den j -ten Spannungsintensitätsfaktor und S_j die j -te Singularitätenfunktion. Die Berechnung der Spannungsintensitätsfaktoren a_j setzt die Kenntnis der Funktion $g(\Theta)$ voraus. Die Funktion $g(\Theta)$ kann aus einer Lösung des Randwertproblems (8.34) ermittelt werden. Eine Lösung von (8.34) für die geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$ kann folgendermaßen dargestellt werden:

$$u = 10y^2 + v \quad (8.36)$$

Dabei erfüllt die Funktion v die Bedingung:

$$\Delta v = 0 \quad \text{und } v(r, \Theta) = -10 \sin^2 \Theta \quad \text{auf } \Gamma_3 \quad (8.37)$$

Aus der eindeutigen Lösbarkeit von (8.34) folgt, daß v dadurch eindeutig festgelegt ist.

Der Graph der Lösung von (8.34) ist in den Abbildungen 8.16 und 8.17 dargestellt.

Aus (8.37) ergibt sich $g(\Theta) = v$. Damit können die Spannungsintensitätsfaktoren der Lösung (8.35) aus der Beziehung

$$a_j = \frac{2}{\alpha} \int_0^\alpha g(\Theta) \sin(j\beta\Theta) d\Theta \quad (8.38)$$

mit $g(0) = g(\alpha) = 0$ berechnet werden.

Die Lösung $u(r, \Theta)$ des Problems 8.34 besitzt die Form:

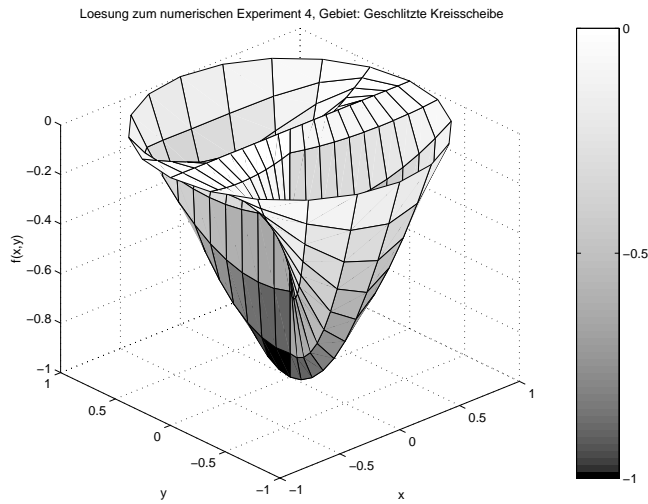


Abbildung 8.16: Experiment 4: Lösung u des Problems (8.34),
Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$

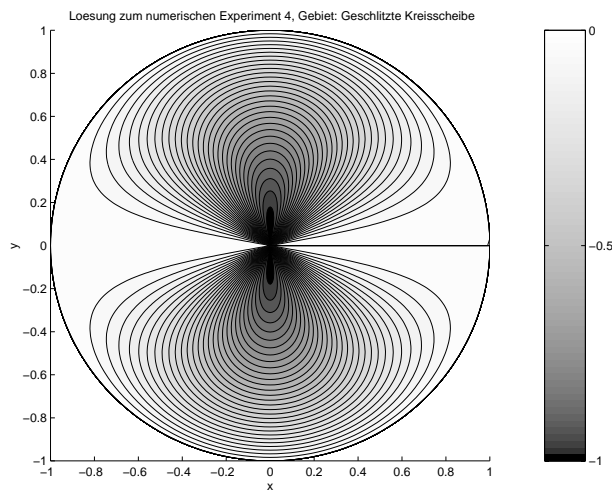


Abbildung 8.17: Experiment 4: Lösung u des Problems (8.34)
als Höhenliniendarstellung, Gebiet: Geschlitzte Kreisscheibe mit
 $\alpha = 2\pi$, $\beta = \frac{1}{2}$

Numerisches Experiment 4		
Spannungsintensitätsfaktoren		
$a_1 = -6.791$	$a_7 = 0.441$	$a_{13} = 0.051$
$a_3 = -4.850$	$a_9 = 0.174$	$a_{15} = 0.032$
$a_5 = 2.264$	$a_{11} = 0.080$	

Tabelle 8.13: Spannungsintensitätsfaktoren des Torsionsproblems $-\Delta u = -20$ in Ω und $u = 0$ auf $\partial\Omega$ mit $a_j = 0$ für geradzahliges j

$$u(r, \Theta) = \sum_{j=1}^{11} r^{\frac{1}{2}j} \sin\left(\frac{1}{2}j\Theta\right) \quad (8.39)$$

Die Rechnungen werden auf der geschlitzten Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$ durchgeführt, da für dieses Gebiet die Lösung von (8.34) bekannt ist. Für das Kreissegment mit $\alpha = \frac{3}{2}\pi$, $\beta = \frac{2}{3}$ ist es nicht möglich, eine Randwertfunktion $g(\Theta)$ mit $g(0) = g(\Theta) = 0$ zu finden. Folglich können die Spannungsintensitätsfaktoren zur Bestimmung der Lösung (8.4) nicht berechnet werden und die Untersuchungen beschränken sich auf das Gebiet der geschlitzten Kreisscheibe.

Die den Rechnungen zugrundegelegte Lösung von (8.34) besteht aus den ersten 16 Gliedern der Reihe (8.4), da die nachfolgenden Glieder nur noch unwesentlich zum Verhalten der Lösung beitragen. Die Spannungsintensitätsfaktoren a_j werden durch den folgenden Ausdruck berechnet:

$$a_j = -\frac{10}{\pi} \int_0^{2\pi} \sin^2 \Theta \cdot \sin \frac{1}{2}j\Theta d\Theta \quad (8.40)$$

Die berechneten Spannungsintensitätsfaktoren a_j werden nachfolgend aufgelistet, um zu zeigen, daß der erste Spannungsintensitätsfaktor und damit verbunden die erste Singularitätenfunktion maßgeblich die Lösung von (8.34) bestimmt. Für geradzahliges j in (8.4) gilt $a_j = 0$, $j = 2, 4, \dots, 16$. In Tabelle 8.13 sind die Werte der Spannungsintensitätsfaktoren, die von 0 verschieden sind, aufgelistet.

Relativer Fehler in $L_2(\Omega)$

Der berechnete relative Fehler des Gradienten auf unverfeinerten Gittern ist in Tabelle 8.14 dargestellt. Nach Theorem 6.3.1 ist eine fast optimale Konvergenzordnung $O(h^{1-\varepsilon})$, $\varepsilon > 0$ zu erwarten. Die berechneten Konvergenzordnungen zeigen erst bei einer Maschenweite von $h = 0.0125$ das erwartete Verhalten. Ei-

Numerisches Experiment 4			
Relativer Fehler des Gradienten in $L_2(\Omega)$			
Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi, \beta = \frac{1}{2}$			
h	# Knoten	Fehler in $L_2(\Omega)$	Konvergenzordnung
0.1	600	0.3187	
0.05	2480	0.1737	0.88
0.025	9920	$0.9872 \cdot 10^{-1}$	0.82
0.0125	40000	$0.5261 \cdot 10^{-1}$	0.91

Tabelle 8.14: Konvergenzordnungen der relativen Fehler des Torsion-Problems $-\Delta u = -20$ mit der Lösung $u = -10r^2 \sin^2 \Theta + f$, $\Delta f = 0$ auf Ω , $u = 0$ auf $\partial\Omega$ in der $L_2(\Omega)$ -Norm bei uniformer Verfeinerung

ne mögliche Erklärung liefert der "pollution"-Effekt, der hier dominiert und das Konvergenzverhalten des Torsionsproblems beeinflusst. Erst bei kleinen Maschenweiten kann die "pollution" kontrolliert werden und die Konvergenzrate strebt gegen $O(h^{1-\varepsilon})$, $\varepsilon > 0$. Eine weitere Störquelle stellt die Entwicklung der Lösung in eine Reihe nach (8.39) dar. Durch Überlagerung von Funktionen der Reihe (8.39) ist es möglich, daß die erste Singularitätenfunktion während des Lösungsprozeß nicht immer der größte Term ist. Somit kann die Annahme in Theorem 6.3.1, daß die erste Singularitätenfunktion das Lösungsverhalten bestimmt, verletzt werden. Die Konvergenzrate erreicht in diesem Fall nicht die fast optimale Ordnung.

Relativer Fehler in L_∞

Die auf geometrisch verfeinerten Gittern berechneten relativen Fehler der Funktion und des Gradienten zur Verifizierung der Theoreme 7.3.1 und 7.3.2 sind in den Tabellen 8.15 und 8.16 aufgelistet.

Nach Theorem 7.3.2 zeigt der relative Fehler der Funktion eine fast optimale Konvergenzordnung von $O(h^{2-\varepsilon})$, $\varepsilon > 0$. Der Vergleich mit den Werten in Tabelle 8.15 zeigt eine gute Übereinstimmung der berechneten Konvergenzordnungen mit den Aussagen von Theorem 7.3.2. Die berechneten Konvergenzordnungen des relativen Fehlers des Gradienten sind Tabelle 8.16 aufgelistet. Die nach Theorem 7.3.1 erwartete fast optimale Konvergenzordnung von $O(h^{1-\varepsilon})$, $\varepsilon > 0$ wird durch die numerischen Experimente bestätigt.

Numerisches Experiment 4				
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	3120	$0.43 \cdot 10^{-1}$	2.03
0.05	$0.25 \cdot 10^{-2}$	16864	$0.105 \cdot 10^{-1}$	
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	4440	$0.5 \cdot 10^{-1}$	2.04
0.05	$0.125 \cdot 10^{-3}$	24056	$0.121 \cdot 10^{-1}$	

Tabelle 8.15: Konvergenzordnungen der relativen Fehler der Funktion des numerischen Experiments 4 mit $\rho h^{1-\frac{1}{\beta}} \leq d \leq 0.25$ bei geometrischer Verfeinerung

Numerisches Experiment 4				
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	3120	$0.452 \cdot 10^{-1}$	0.98
0.05	$0.25 \cdot 10^{-2}$	16864	$0.229 \cdot 10^{-1}$	
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	4440	$0.448 \cdot 10^{-1}$	0.97
0.05	$0.125 \cdot 10^{-3}$	24056	$0.228 \cdot 10^{-1}$	

Tabelle 8.16: Konvergenzordnungen der relativen Fehler des Gradienten des numerischen Experiments 4 mit $\rho \leq d \leq 0.125$ bei geometrischer Verfeinerung

Numerisches Experiment 4			
Fehler in der $L_\infty(\Omega)$ -Norm			
<i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>			
h	Anzahl Knoten	Fehler in $L_\infty(\Omega)$	Konvergenzordnung
0.1	600	0.225	0.46
0.05	2480	0.164	
0.025	9920	0.118	0.48
0.0125	40000	0.0844	0.49

Tabelle 8.17: Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm bei uniformer Verfeinerung

In Tabelle 8.17 werden die berechneten Konvergenzordnungen in der $L_\infty(\Omega)$ -Norm auf unverfeinerten Gittern dargestellt. Die nach [93] zu erwartende Konvergenzordnung des Fehlers von $O(h^\beta)$ wird durch die numerischen Experimente bestätigt. Im Vergleich zum relativen Fehler der Funktion ist der Fehler der Funktion selbst bei kleinen Maschenweiten deutlich größer.

8.9 Numerisches Experiment 5

Als weiteres Beispiel eines Randwertproblems mit einer rechten Seite $f \neq 0$ und verschwindenden Randwerten wird das Problem

$$\begin{aligned} -\Delta u &= 10r^2 && \text{in } \Omega \\ u &= 0 && \text{auf } \partial\Omega \end{aligned} \quad (8.41)$$

betrachtet. Eine Lösung des Problems (8.41) für die geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$ mit einer geeigneten Funktion v zur Erfüllung der Randbedingungen kann dargestellt werden als:

$$u = -\frac{1}{2}r^2 \sin^2 \Theta \cos^2 \Theta + v \quad \text{mit } \Delta v = 0 \quad (8.42)$$

Die Funktion v besitzt die Form $v = \sin^2 2\Theta$ auf dem Rand. Wegen der eindeutigen Lösbarkeit des Problems (8.41) folgt, daß deshalb, daß v eindeutig festgelegt ist.

In den Abbildungen 8.18 und 8.19 wird die Lösungsfunktion u des Problems 8.41 auf der geschlitzten Kreisscheibe veranschaulicht. Aus (8.42) folgt die Funktion $g(\Theta) = v$ zur Berechnung der Spannungsintensitätsfaktoren a_j . Somit werden die zugehörigen Fourier-Koeffizienten für die Bestimmung der Koeffizienten a_j durch das folgende Integral berechnet:

$$a_j = \frac{10}{8\pi} \int_0^{2\pi} \sin^2 2\Theta \cdot \sin \frac{1}{2} j\Theta \, d\Theta \quad (8.43)$$

Die Spannungsintensitätsfaktoren für die ersten 16 Glieder der Reihe (8.4) sind in Tabelle 8.18 aufgelistet. Für geradzahliges j sind die Koeffizienten gleich 0. Aus Tabelle 8.18 ist ersichtlich, daß der Term a_1 am größten ist und somit die erste Singularitätenfunktion das Lösungsverhalten bestimmt. Die Spannungsintensitätsfaktoren werden wie $\frac{1}{j}$ kleiner und somit sind ebenfalls die Bedingungen (5.17) und (5.18) erfüllt. Für die Lösung von (8.41) auf einem Kreissegment mit $\alpha = \frac{3}{2}\pi$, $\beta = \frac{2}{3}$ gilt ebenfalls (8.42) mit einer geeignet gewählten Funktion v zur Bestimmung von $g(\Theta)$ in (8.8). Das Fourier-Integral zur Berechnung der Koeffizienten von (8.4) ist gegeben durch:

$$a_j = \frac{20}{3\pi} \int_0^{\frac{3}{2}\pi} \sin^2 2\Theta \cdot \sin \frac{2}{3} j\Theta \, d\Theta \quad (8.44)$$

Die ersten 16 Spannungsintensitätsfaktoren sind in Tabelle 8.18 dargestellt. Hier sind ebenfalls die Spannungsintensitätsfaktoren mit geradzahligem Index j gleich

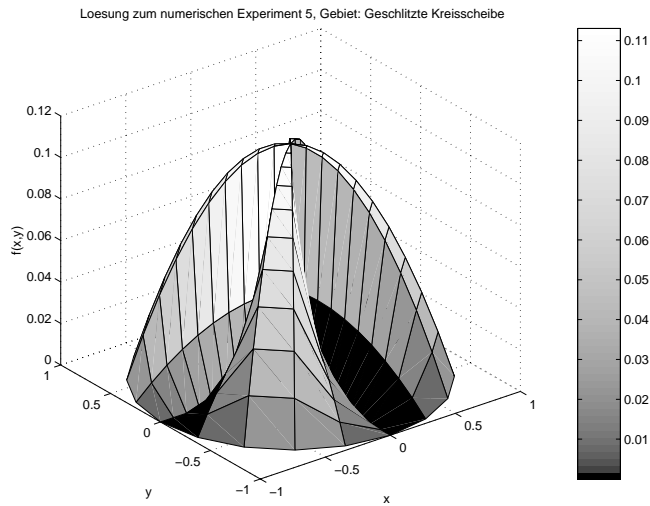


Abbildung 8.18: Experiment 5: Lösung u des Problems (8.41), Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$)

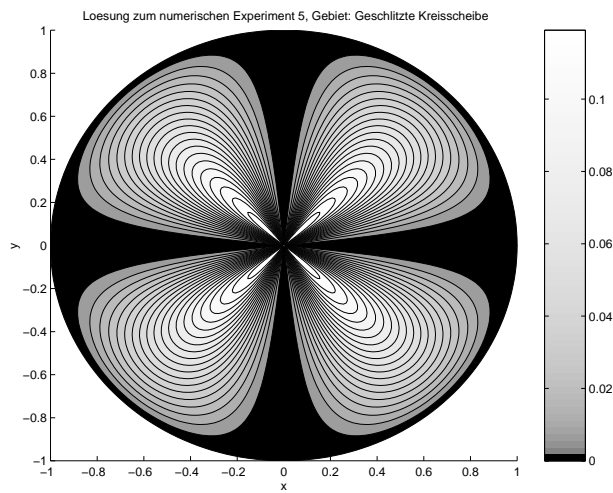


Abbildung 8.19: Experiment 5: Lösung u des Problems (8.41) als Höhenliniendarstellung, Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi$, $\beta = \frac{1}{2}$)

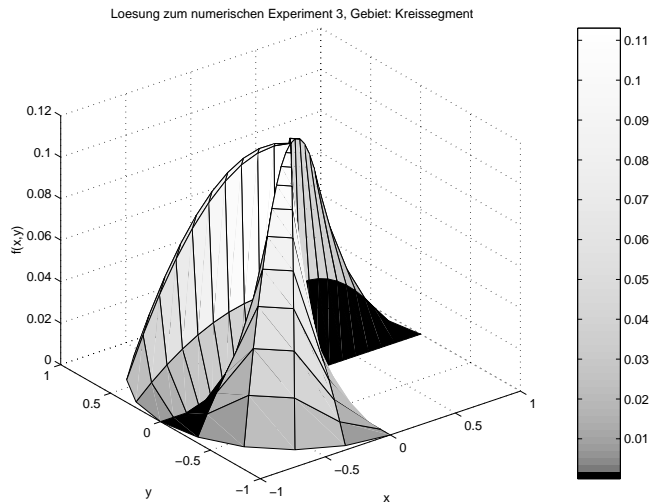


Abbildung 8.20: Experiment 5: Lösung u des Problems (8.41),
Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi$, $\beta = \frac{2}{3}$

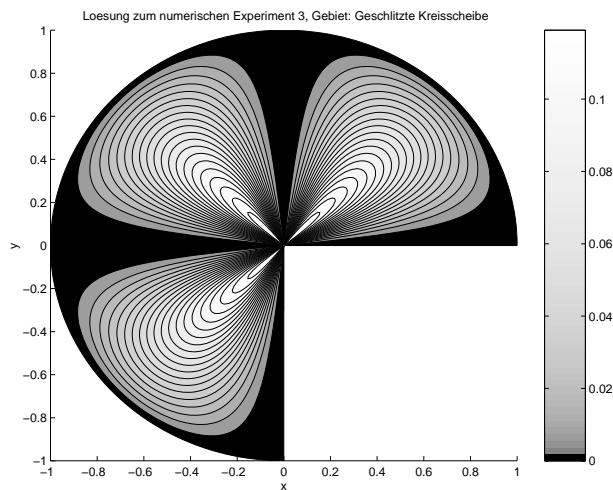


Abbildung 8.21: Experiment 5: Lösung u des Problems (8.41)
als Höhenliniendarstellung, Gebiet: Geschlitzte Kreisscheibe mit
 $\alpha = \frac{3}{2}\pi$, $\beta = \frac{2}{3}$

Numerisches Experiment 5		
Spannungsintensitätsfaktoren		
<i>Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>		
$a_1 = 0.810$	$a_7 = 0.485$	$a_{13} = -0.037$
$a_3 = 0.310$	$a_9 = -0.333$	$a_{15} = -0.021$
$a_5 = 0.261$	$a_{11} = -0.081$	
<i>Kreissegment, $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>		
$a_1 = 0.819$	$a_7 = -0.315$	$a_{13} = -0.017$
$a_3 = 0.354$	$a_9 = -0.071$	$a_{15} = -0.010$
$a_5 = 0.520$	$a_{11} = -0.031$	

Tabelle 8.18: Spannungsintensitätsfaktoren des Randwertproblems $-\Delta u = 10r^2$ in Ω und $u = 0$ auf $\partial\Omega$, $j = 0$, falls j geradzahlig

0. Der erste Spannungsintensitätsfaktor a_1 ist am größten und somit wird die Lösung durch die erste Singularitätenfunktion bestimmt. Da die Spannungsintensitätsfaktoren mindestens mit $\frac{1}{j}$ modulo einer Konstanten kleiner werden, sind ebenfalls die Bedingungen (5.17) und (5.18) erfüllt. In den Abbildungen 8.20 und 8.21 wird die Lösungsfunktion von (8.41) auf einem Kreissegment dargestellt.

Relativer Fehler in $L_2(\Omega)$

Die berechneten Konvergenzraten der relativen Fehler des Gradienten auf unverfeinerten Gittern sind in Tabelle 8.19 dargestellt. Nach den Aussagen des Theorems 6.3.1 ist eine fast optimale Konvergenzordnung $O(h^{1-\varepsilon})$, $\varepsilon > 0$ zu erwarten. Die berechneten Konvergenzordnungen erreichen auf der geschlitzten Kreisscheibe erst bei einer Maschenweiten von $h = 0.0125$ einen fast optimalen Wert. Die Konvergenzraten auf dem Kreissegment bleiben unter der nach Theorem 6.3.1 vorhergesagten Ordnung. In diesem Fall wird das Lösungsverhalten nicht nur vom ersten Spannungsintensitätsfaktor bestimmt und damit eine wichtige Annahme zur Gültigkeit von Theorem 6.3.1 nicht ganz erfüllt. Ein Vergleich der Spannungsintensitätsfaktoren in Tabelle 8.18 zeigt, daß der Spannungsintensitätsfaktor a_5 einen im Vergleich zu a_1 zu großen Wert besitzt. Als Konsequenz ist auf diesem Gebiet die Konvergenzrate nicht von optimaler Ordnung.

Numerisches Experiment 5			
Relativer Fehler des Gradienten in $L_2(\Omega)$			
Gebiet: Geschlitzte Kreisscheibe mit $\alpha = 2\pi, \beta = \frac{1}{2}$			
h	# Knoten	Fehler in $L_2(\Omega)$	Konvergenzordnung
0.1	600	0.3179	0.76
0.05	2480	0.1882	
0.025	9920	0.1161	0.7
0.0125	40000	0.5909	0.97
Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$			
h	# Knoten	Fehler in $L_2(\Omega)$	Konvergenzordnung
0.1	440	0.2215	0.54
0.05	1840	0.5121	
0.025	7520	$0.8112 \cdot 10^{-1}$	0.91
0.0125	30080	$0.4439 \cdot 10^{-1}$	0.87

Tabelle 8.19: Konvergenzordnungen der relativen Fehler des Gradienten des Problems $-\Delta u = 10r^2$ mit der Lösung $u = -\frac{1}{2}r^2 \sin^2 \Theta \cos^2 \Theta + v$ mit $\Delta v = 0$ in Ω und $u = 0$ auf $\partial\Omega$ in der $L_2(\Omega)$ -Norm bei uniformer Verfeinerung

Relativer Fehler in L_∞

In den Tabellen 8.20 und 8.21 werden die auf geometrischen Gitter berechneten Konvergenzordnungen der relativen Fehler der Funktion und des Gradienten dargestellt.

Nach Theorem 7.3.2 beträgt die Konvergenzordnung des relativen Fehlers der Funktion $O(h^{2-\varepsilon})$, $\varepsilon > 0$. Die berechneten Werte nach Tabelle 8.20 stimmen gut mit den theoretisch vorgesagten Werten überein. In Tabelle 8.21 werden die auf geometrisch verfeinerten Gittern berechneten Konvergenzordnungen der relativen Fehler des Gradienten vorgestellt. Die nach Theorem 7.3.1 zu erwartende Konvergenzordnung $O(h^{1-\varepsilon})$, $\varepsilon > 0$ wird durch die numerischen Experimente bestätigt. Zum Vergleich der Konvergenzordnungen auf unverfeinerten Gittern wird der Fehler in der $L_\infty(\Omega)$ -Norm berechnet. Nach [93] ist eine Konvergenzordnung $O(h^\beta)$ zu erwarten. Bei einer Maschenweite von $h = 0.0125$ erreichen die berechneten Konvergenzordnung die nach [93] zu erwartenden Werte.

Numerisches Experiment 5				
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	3120	$0.118 \cdot 10^{-1}$	1.87
0.05	$0.25 \cdot 10^{-2}$	16864	$0.323 \cdot 10^{-2}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	4440	$0.134 \cdot 10^{-2}$	1.98
0.05	$0.125 \cdot 10^{-3}$	24056	$0.340 \cdot 10^{-3}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	2288	$0.994 \cdot 10^{-3}$	2.0
0.05	$0.25 \cdot 10^{-2}$	12512	$0.248 \cdot 10^{-3}$	
Relativer Fehler der Funktion in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	3256	$0.176 \cdot 10^{-2}$	1.95
0.05	$0.125 \cdot 10^{-3}$	17848	$0.455 \cdot 10^{-3}$	

Tabelle 8.20: Konvergenzordnungen der relativen Fehler der Funktion des numerischen Experiments 5 mit $\rho h^{1-\frac{1}{\beta}} \leq d \leq 0.25$ bei geometrischer Verfeinerung

Numerisches Experiment 5;				
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	3120	$0.371 \cdot 10^{-1}$	1.04
0.05	$0.25 \cdot 10^{-2}$	16864	$0.181 \cdot 10^{-1}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	4440	$0.356 \cdot 10^{-1}$	1.05
0.05	$0.125 \cdot 10^{-3}$	24056	$0.172 \cdot 10^{-2}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^2$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	0.01	2288	$0.243 \cdot 10^{-1}$	1.03
0.05	$0.25 \cdot 10^{-2}$	12512	$0.119 \cdot 10^{-1}$	
Relativer Fehler des Gradienten in der $L_\infty(\tau)$ -Norm <i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>				
h	$\rho = h^3$	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	$0.1 \cdot 10^{-3}$	3256	$0.240 \cdot 10^{-1}$	0.98
0.05	$0.125 \cdot 10^{-3}$	17848	$0.122 \cdot 10^{-1}$	

Tabelle 8.21: Konvergenzordnungen der relativen Fehler des Gradienten des numerischen Experiments 5 mit $\rho \leq d \leq 0.125$ bei geometrischer Verfeinerung

Numerisches Experiment 5			
Fehler in der $L_\infty(\Omega)$ -Norm			
<i>Gebiet: Geschlitzte Kreisscheibe, $\alpha = 2\pi, \beta = \frac{1}{2}$</i>			
h	Anzahl Knoten	Fehler in $L_\infty(\Omega)$	Konvergenzordnung
0.1	600	0.225	0.46
0.05	2480	0.164	
0.025	9920	0.118	0.48
0.0125	40000	0.0844	0.49
Fehler in der $L_\infty(\Omega)$ -Norm			
<i>Gebiet: Kreissegment mit $\alpha = \frac{3}{2}\pi, \beta = \frac{2}{3}$</i>			
h	Anzahl Knoten	rel. Fehler	Konvergenzordnung
0.1	600	$0.543 \cdot 10^{-1}$	0.71
0.05	2480	$0.332 \cdot 10^{-1}$	
0.025	9920	$0.202 \cdot 10^{-1}$	0.71
0.0125	40000	$0.125 \cdot 10^{-1}$	0.69

Tabelle 8.22: Konvergenzordnungen des Fehlers in der $L_\infty(\Omega)$ -Norm bei uniformer Verfeinerung

Literaturverzeichnis

- [1] ADAMS, R.A.: *Sobolev Spaces*. Academic Press, 1978.
- [2] AURENHAMMER, F.: *Voronoi diagrams - a survey of a fundamental geometric data structure*. ACM Computing Surveys, 23:345 – 405, 1991.
- [3] BABUŠKA, I. und A.K. AZIZ: *On the angle condition in the finite element method*. SIAM J. Numer. Anal., 13:214 – 226, 1976.
- [4] BABUŠKA, I. und M. ROSENZWEIG: *A finite element scheme for domains with corners*. Numer. Math., 20:1 – 21, 1972.
- [5] BABUŠKA, I. und B. SZABO: *On the rate of convergence of the finite element method*. Int. J. Numer. Meth. Eng., 18:323 – 341, 1982.
- [6] BABUŠKA, I. und B. SZABO: *Trends and new problems in finite element methods*, Seiten 35 – 54. In: WHITEMAN, J.R. [117], 1997.
- [7] BÄHMANN, P. L., S. L. WITTCHEN, M. S. SHEPHARD, K. R. GRICE und M. A. YERRY: *Robust geometrically-based automatic two-dimensional mesh generation*. The Comp. Journ., 24.2:162 – 167, 1981.
- [8] BAKER, T. J.: *Mesh generation by a sequence of transformations*. Proc. AIAA 8th CFD, Honolulu, 1987.
- [9] BANK, R.E. und A. WEISER: *Some a posteriori error estimators for elliptic partial differential equations*. Math. Comp., 44:283 – 301, 1985.
- [10] BARTH, T. J. und D. C. JESPERSEN: *The design and application of upwind schemes on unstructured meshes*. Proc. AIAA 27th Aerospace sciences meeting, Reno, 1989.
- [11] BASTIAN, P., K. BIRKEN, K. JOHANNSEN, N. NEUSS S. LANG, H. RENZ-REICHERT und C. WIENERS: *UG - A flexible software toolbox for solving partial differential equations*. Computing and visualization in science, 1:793 – 800, 1997.

- [12] BERN, M. und D. EPPSTEIN: *Polynomial-size nonobtuse triangulation of polygons*. Proc. 7th ACM Symp. Comp. Geometry, Seiten 342 – 350, 1991.
- [13] BERN, M., D. EPPSTEIN und J. R. GILBERT: *Probably good mesh generation*. Proc. 31st IEEE Symp. Foundations of Computer Science, 1990.
- [14] BHATTACHARYA, M., S. KUMAR, C.V. RAMAKRISHNAN und S. RAMAKRISHNAN: *An integrated approach for automated generation of two/three dimensional finite element grids using spatial occupancy enumeration and Delaunay triangulation*. Int. J. Numer. Meth. Eng., 34:1035 – 1050, 1992.
- [15] BLACKER, T. D. und M. B. STEPHENSON: *Paving: a new approach to automated quadrilateral mesh generation*. Int. J. Numer. Meth. Eng., 32:811 – 847, 1991.
- [16] BOLLOBAS, B.: *Graph theory, An introductory course*. New York, Springer, 1979.
- [17] BOWYER, A.: *Computing Dirichlet tessellations*. Comput. J., 24:162 – 166, 1981.
- [18] BRAMBLE, J. H. und S. R. HILBERT: *Estimation of linear functionals on Sobolev spaces with applications to Fourier transforms and spline interpolation*. Siam. J. Numer. Anal., 7:113 – 124, 1970.
- [19] BRENNER, S.C. und L.R. SCOTT: *The mathematical theory of finite element methods*. Springer, New York, 1994.
- [20] BRENNER, S.C. und L.-Y. SUNG: *Multigrid methods for the computation of singular solutions and stress intensity factors II: crack singularities*. Numer. Math., 37.3:623 – 643, 1997.
- [21] CANAAN, S.A., S. SAIGAL und M.L. STATENAND S.J. OWEN: *Advancing front quadrilateral meshing using triangle transformations*. Proceedings, 7th Meshing Roundtable, Dearborn, MI, Oct. 1998.
- [22] CANAAN, S.A., M.L. STATEN und J.R. TRISTANO: *An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral and quad-dominant meshes*. Proceedings, 7th Meshing Roundtable, Dearborn, MI, Oct. 1998.
- [23] CASTILLO, J.E.: *Mathematical aspects of numerical grid generation*. SIAM, 1991.

- [24] CAVENDISH, J. C.: *Automatic triangulation of arbitrary planar domains for the FE-method*. Int. J. Numer. Meth. Eng., 87:679 – 696, 1974.
- [25] CIARLET, P.G.: *The finite element method for elliptic problems*. North Holland, 1978.
- [26] DELAUNAY, B.: *Sur la sphère vide*. Bull. Acad. Sci USSR(VII), Classe Sci. Mat. Nat., Seiten 793 – 800, 1934.
- [27] DELJOUÏE-RAKHSANDEH, K.: *An approach to the generation of triangular grids possessing few obtuse triangles*. Int. J. Numer. Meth. Eng., 29:1299 – 1321, 1990.
- [28] DIRICHLET, G. T.: *Über die Reduktion der positiven quadratischen Formen mit drei unterbestimmten ganzen Zahlen*. Z. Reine Angew. Math., 40.3:209 – 227, 1850.
- [29] DOBROWOLSKI, M.: *Numerical approximation of elliptic interface and corner problems*. Habilitationsschrift, Rheinische Friedrich-Wilhelms-Universität Bonn, 1981.
- [30] DÖRFLER, W.: *A convergent adaptive algorithm for Poisson's equation*. SIAM J. Numer. Anal., 33:1106 – 1124, 1996.
- [31] DURAN, R., M. A. MUSCHIETTE und R. RODRIGUEZ: *On the asymptotic exactness of error estimators for linear triangular finite elements*. Numer. Math., 59:107 – 127, 1991.
- [32] EDELSBRUNNER, H., T.S. TAN und R. WAUPOTITSCH: *A polynomial time algorithm for the minmax angle triangulation*. Proc. 5th ACM Symp. Comp. Geometry, Seiten 44 – 52, 1990.
- [33] E. FEIN, ED.: *d3f - Ein Programmpaket zur Modellierung von Dichteströmungen*. Technischer Report, GRS mbH, Theodor-Heus-Str. 4, 38122 Braunschweig, GRS - 139, 1998.
- [34] FEUCHTER, D.: *Optimierung eines Gittergenerators für Mehrgitterverfahren*. Diplomarbeit, Institut für Computeranwendungen 3 (ICA 3), Universität Stuttgart, 1995.
- [35] FORMAGGIO, G.: *Unstructured mesh generation and adaption*. ZAMM, 76, S1:65 – 68, 1996.

- [36] FORTUNE, S.: *A sweepline algorithm for Voronoi diagrams*. Technischer Bericht, AT & T Bell Laboratory Report, Murray Hill, NJ, 1985.
- [37] FRENCH, D. A.: *The finite element method for a degenerate elliptic problem*. SIAM J. Numer. Anal., 24.4:788 – 815, 1987.
- [38] FRIED, I.: *Condition of finite element matrices generated from nonuniform meshes*. AIAA, 10:219 – 221, 1972.
- [39] GALLAGHER, R.H.: *Survey and evaluation of the finite element method in linear fracture mechanics analysis*. in: Computational fracture mechanics, Second national congress on pressure vessels and piping, ASME, New York, 1975.
- [40] GEORGE, P., F. HECHT und E. SALTEL: *Automatic 3d mesh generator with prescribed meshed boundaries*. IEEE Transactions on magnetics, 26.2:771 – 774, 1990.
- [41] GEORGE, P., F. HECHT und E. SALTEL: *Automatic mesh generator with specified boundary*. Comp. Meth. Appl. Mech. Eng., 92:269 – 288, 1991.
- [42] GEORGE, P. und F. HERMELINE: *Delaunay's mesh of a convex polyhedron in dimension d. Application to arbitrary polyhedra*. Int. J. Numer. Meth. Eng., 33:975 – 995, 1992.
- [43] GRAHAM, I. G., M.J. HAGGER und B.N. KHOROMSKIJ: *Locally refined finite element approximation of a model degenerate elliptic equation*. Technischer Bericht, Preprint, January, University of Bath, UK, 1999.
- [44] GREEN, P. J. und R. SIBSON: *Computing Dirichlet tessellations in the plane*. Comp. J., 21.2:168 – 173, 1978.
- [45] GRISVARD, P.: *Behaviour of the solution of an elliptic boundary value problem in a polygonal or polyhedral domain*, Seiten 3 – 363. Academic Press, New York, 1976.
- [46] GRISVARD, P.: *Behaviour of the solution of an elliptic boundary value problem in a polygonal or polyhedral domain*. Academic Press, New York, 207 - 274, 1976.
- [47] GRISVARD, P.: *Elliptic problems in nonsmooth domains*. Pitman, London, 1985.

- [48] GRISVARD, P.: *Singularities in boundary value problems*. Masson, Paris, Springer, Heidelberg, 1992.
- [49] GRISVARD, P., W. WENDLAND und J.R. WHITEMAN (Herausgeber): *Lecture notes in mathematics, 1121*. Springer, Berlin, 1985.
- [50] HELLEN, T.K.: *Numerical methods in fracture mechanics*. Applied science publishers, Barking, England, 1980.
- [51] HENSHAW, W. D.: *Automatic grid generation*. Acta Numerica, Seiten 121 – 148, 1996.
- [52] HOARE, C. A. R.: *Quicksort*. Comput. J., 5:10 – 15, 1962.
- [53] HOFFMANN, W., A.H. SCHATZ, L.B. WAHLBIN und G. WITTUM: *The analysis of some local pointwise a priore error estimates for elliptic problems*. to be published.
- [54] HO-LE, K.: *Finite element mesh generation methods: a review and classification*. Computer-Aided Design, 20:271 – 38, 1988.
- [55] HOLMES, D.G. und D. D. SNYDER: *The generation of unstructured triangular meshes using Delaunay triangulation*. In: *Proc. 2nd Int. Conf. Numer. Grid Generation in Comp. Fluid Mech, Miami, Florida*, Seiten 137 – 149. Pineridge Press, Swansea, 1988.
- [56] J. F. THOMPSON, ED.: *Numerical grid generation: Foundations and applications*. North-Holland, 1985.
- [57] KELLOG, R.B.: *Higher order singularities for interface problems*, Seiten 589 – 602. Academic Press, New York, 1972.
- [58] KELLY, D.W., J.P. DE, S.R. GAGO und O.C. ZIENKIEWICZ: *A posteriori error analysis and adaptive process in the finite element method: Part I - error analysis*. Int. J. Numer. Meth. Eng., 19:1593 – 1619, 1983.
- [59] KHAMAYSEH, A. und A. KUPRAT: *Anisotropic smoothing and solution adaption for unstructured grids*. Int. J. Numer. Meth. Eng., 39:3163 – 3174, 1996.
- [60] KNUPP, P. und S. STEINBERG: *Fundamentals of grid generation*. CRC Press, North Holland, 1993.

- [61] KONDRATEV, V.A.: *Boundary value problems for elliptic equations in domains with conical or angular points*. Trans. Moscow Math. Soc., 16:227 – 313, 1967.
- [62] LÖHNER, R.: *Some useful data structures for the generation of unstructured grids*. Int. J. Numer. Meth. Eng., 38:123 – 135, 1985.
- [63] LÖHNER, R.: *Finite elements in CFD; what lies ahead*. Int. J. Numer. Meth. Eng., 24:1741 – 1756, 1987.
- [64] LO, S. H.: *Delaunay triangulation of non-convex planar domains*. Int. J. Numer. Meth. Eng., 28:2695 – 2707, 1989.
- [65] MARCUM, D. L. und N. P. WEATHERILL: *Unstructured grid generation using iterative point insertion and local reconnection*. AIAA Journal, 33.9:1619 – 1625, 1995.
- [66] MARINI, D. und P. PIETRA: *Mixed finite element approximation of a degenerate elliptic problem*. Numer. Math., 71:225 – 236, 1995.
- [67] MAVRIPILIS, D. J.: *Adaptive mesh generation for viscous flows using Delaunay Triangulation*. J. Comp. Physics, 90:271 – 291, 1990.
- [68] MAVRIPILIS, D. J.: *Unstructured and adaptive mesh generation for high Reynolds number viscous flows*. In: *Numerical grid generation in computational fluid dynamics and related fields, Proceedings of the Third Conference on numerical grid generation in computational fluid dynamics and related fields, Barcelona, Spain, 3 - 7 June, 1991*, Seiten 79 – 91. Elsevier, North-Holland, 1991.
- [69] MEHLHORN, K.: *Effiziente Algorithmen, Band 1*. Teubner, Stuttgart, 1988.
- [70] MERIGOT, M.: *Règularité des dérivées de la solution du problème de Dirichlet dans un secteur plan*. C.R. Acad. Sci. Paris, 273:356 – 359, 1971.
- [71] MERIGOT, M.: *Règularité des dérivées de la solution du problème de Dirichlet dans un secteur plan*. Matematiche, 28:1 – 36, 1972.
- [72] MERIGOT, M.: *Etude de la problème $\Delta u = f$ dans une polygone plan: Inègalités à priori*. Boll. U.M.I, 4:577 – 597, 1974.
- [73] MERIGOT, M.: *Potentiels et régularité L_p pour les problèmes elliptiques dans un secteur plan*. C.R. Acad. Sci. Paris, 278:1487 – 1489, 1974.

- [74] NATTERER, F.: *Über die punktweise Konvergenz finiter Elemente*. Numer. Math., 25:67 – 77, 1975.
- [75] NITSCHKE, J.A.: *L_∞ convergence of finite element approximation*. Proc. Sec. Conf. on finite elements, Rennes, France, 1975.
- [76] NITSCHKE, J.A.: *L_∞ convergence of finite element approximation*. Mathematical aspects of finite element methods, Lecture Notes in Math., Springer, 606:261 – 274, 1977.
- [77] NITSCHKE, J.A. und A.H. SCHATZ: *Interior estimates for Ritz-Galerkin methods*. Comp. Math., 28:937 – 958, 1974.
- [78] OTTMANN, T. und P. WIDMAYER: *Algorithmen und Datenstrukturen*. BI Mannheim, 1990.
- [79] PAOLINI, M. und C. VERDI: *An automatic triangular mesh generator for planar domains*. Technischer Report, SE.A.G. Via San Fermo, I-Pavia.
- [80] PERAIRE, J., M. VAHDATI, K. MORGAN und O.C. ZIENKIEWICZ: *Adaptive remeshing for compressible flow computations*. J. Comp. Physics, 72:449 – 466, 1987.
- [81] PREPARATA, F. P. und M. I. SHAMOS: *Computational Geometry*. Springer Berlin, 1988.
- [82] RANNACHER, R.: *Zur L_∞ -Konvergenz linearer finiter Elemente beim Dirichlet Problem*. Math. Z., 149:69 – 77, 1976.
- [83] RANNACHER, R. und R. SCOTT: *Some optimal error estimates for piecewise linear finite element approximations*. Math. Comp., 38:437 – 535, 1982.
- [84] REBAY, S.: *Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm*. J. Comp. Physics, 106:125 – 138, 1993.
- [85] ROSSER, J.B.: *Calculation of potential in a sector*. Technischer Bericht, MRC report 1535, Madison, WI, 1975.
- [86] ROSSER, J.B.: *Harmonic functions on region with reentrant corners*. Technischer Bericht, MRC report 1796, Madison, WI, 1977.
- [87] RÖSSLE, A. und A. SÄNDIG: *Stress singularities in bonded elastic materials*, Seiten 405 – 416. In: WHITEMAN, J.R. [117], 1997.

- [88] SAMET, H.: *The quadtree and related hierarchical data structures*. Computing Surveys, 16:188 – 260, 1984.
- [89] SAMET, H.: *The Design and Analysis of Spatial Data structures*. Addison-Wesley, 1990.
- [90] SAPIDIS, N. und R. PERUCCHIO: *Advanced techniques for automatic finite element meshing from solid models*. Computer-Aided Design, 21:248 – 253, 1989.
- [91] SCHATZ, A. H.: *Pointwise error estimates and asymptotic error expansion inequalities for the finite element method on irregular grids: Part I. Global estimates*. Math. Comp., 67:877 – 899, 1998.
- [92] SCHATZ, A. H. und L.B. WAHLBIN: *Interior maximum norm estimates for finite element method*. Math. Comp., 31:414 – 442, 1977.
- [93] SCHATZ, A. H. und L.B. WAHLBIN: *Maximum norm estimates in the finite element method on plane polygonal domains. Part 1*. Math. Comp., 32:73 – 109, 1978.
- [94] SCHATZ, A. H. und L.B. WAHLBIN: *Maximum norm estimates in the finite element method on plane polygonal domains. Part 2, refinements*. Math. Comp., 33:465 – 492, 1979.
- [95] SCHATZ, A. H. und L.B. WAHLBIN: *On a local asymptotic error estimate in finite elements and its use: numerical examples*. in: R. Vichnevetsky und R.S. Stepleman, (Herausgeber), Advances in computer methods for partial differential equations, IV, IMACS, New Brunswick, NJ, 1981.
- [96] SCHATZ, A. H. und L.B. WAHLBIN: *Interior maximum norm estimates for the finite element method, part 2*. Math. Comp., 64:907 – 928, 1995.
- [97] SCOTT, R.: *Optimal L_∞ estimates for the finite element method on irregular grids*. Math. Comp., 30:681 – 697, 1976.
- [98] SEDGEWICK, R.: *Algorithms*. Addison-Wesley, 1988.
- [99] SHEPHARD, M. S.: *Approaches to the automatic generation and control of finite element meshes*. Appl. Mech. Rev., 41:169 – 185, 1988.
- [100] SLOAN, S. W.: *A fast algorithm for generating constrained Delaunay triangulations*. Computers and Structures, 47.3:441 – 450, 1993.

- [101] SLOAN, S. W. und G. T. HOULSBY: *An implementation of Watson's algorithm for computing 2D Delaunay triangulations*. Advances in Eng. Software, 6/4:192 – 197, 1984.
- [102] SRINIVASAN, V., L. R. NACKMAN, J-M. TANG und S. N. MESHKAT: *Automatic mesh generation using the symmetric axis transformation of polygonal domains*. Technischer Bericht, Comp. Science, IBM Research Division, Tech. Rept. RC 16132, Yorktown Heights, NY, 1990.
- [103] TANEMURA, M., T. OGAWA und N. OGITA: *A new algorithm for Delaunay triangulation*. JCP, 52:191 – 207, 1983.
- [104] THACKER, W. C.: *A brief review of techniques for generating irregular computational grids*. Int. J. Numer. Meth. Eng., 15:1335 – 1341, 1980.
- [105] THOMPSON, J. F., Z. U. A. WARSI und C. W. MASTIN: *Numerical grid generation: foundations and applications*. Kluwer Academic Publishers, Dordrecht, the Netherlands, 1987.
- [106] THOMPSON, J. F. und N. P. WEATHERILL: *Aspects of numerical grid generation: current science and art*. Technischer Bericht, AIAA, AIAA paper 93-3539, 1993.
- [107] TOLKSDORF, P.: *On quasilinear boundary value problems in domains with corners*. Technischer Bericht, Universität Bonn, SFP 72 Preprint no. 396, 1980.
- [108] TRACY, F. T.: *Graphical pre- and post-processor for two-dimensional finite element method programs*. Computer Graphics, 11.2:8 – 12, 1977.
- [109] VERFÜRTH, R.: *A review of a posteriori error estimation and adaptive mesh-refinement techniques*. Wiley & Sons, Chichester, Teubner, Stuttgart, 1996.
- [110] VORONOI, G.: *Nouvelles applications des parametres continu à la theorie des formes quadratiques. Deuxième Mémoire: Recherches sur les paralléloèdres primitifs*. Z. Reine Angew. Math., 134:198 – 287, 1908.
- [111] WAHLBIN, L. B.: *On the sharpness of certain local estimates for H^1 projections into finite element spaces: influence of a reentrant corner*. Math. Comp., 42:1 – 8, 1984.
- [112] WAHLBIN, L. B.: *Local behavior in finite element methods*, Seiten 353 – 522. Elsevier Science Publishers B.V., North-Holland, 1991.

- [113] WATSON, D. F.: *Computing the n-dimensional Delaunay Tessellation with applications to Voronoi polytopes*. Comput. J., 24.2:167 – 172, 1981.
- [114] WEATHERILL, N. P.: *The Delaunay triangulation in CFD*. Computers and Mathematics, 24.5/6:129 – 150, 1981.
- [115] WEATHERILL, N. P.: *A method for generating irregular computational grids in multiply connected planar domains*. Int. J. Numer. Meth. Fluids, 8, 1988.
- [116] WEATHERILL, N. P. und O. HASSAN: *Efficient three-dimensional grid generation using the Delaunay triangulation*. Proc. 1st European Comp. Fluid Dynamics Conf., Brussels, Sept. 1992, 1992.
- [117] WHITEMAN, J.R. (Herausgeber): *The mathematics of finite elements and application*. Wiley, New York, 1997.
- [118] WIENERS, C.: *Multigrid methods for Prandtl-Reuß plasticity*. eingereicht bei Numer. Lin. Alg. Appl., 1998.
- [119] XIE, G., R.M. CLOUGH und J.A.H. RAMÄKERS: *Graded mesh generation and transformation*. Int. J. Numer. Meth. Eng., 29:1299 – 1321, 1994.
- [120] YERRY, M.A. und M. S. SHEPHARD: *A modified quadtree approach to finite element mesh generation*. IEEE Computer Graphics and Applications, 3:39 – 46, 1982.
- [121] ZHOU, J.-M.: *Computing constraint triangulation and Delaunay triangulation: a new algorithm*. IEEE Transactions on Magnetics, 26.2:694 – 697, 1990.
- [122] ZHOU, J.Z., O. C. ZIENKIEWICZ, E. HINTON und J. WU: *A new approach to the development of automatic quadrilateral mesh generation*. Int. J. Numer. Meth. Eng., 32:849 – 866, 1991.
- [123] ZIENKIEWICZ, O.C. und J.Z. ZHOU: *A simple error estimator and adaptive procedure for practical engineering analysis*. Int. J. Numer. Meth. Eng., 24:337 – 357, 1987.